Neuronske mreže i duboko učenje

Mašinsko učenje 2020/21.

Matematički fakultet
Univerzitet u Beogradu

Neuronske mreže

- Očekujemo da se neuronske mreže dobro ponašaju (daju bolje rezultate od ostalih modela) kada imamo veliku količinu podataka u sirovoj formi

Neuronske mreže

- Očekujemo da se neuronske mreže dobro ponašaju (daju bolje rezultate od ostalih modela) kada imamo veliku količinu podataka u sirovoj formi
- To znači da nisu već prerađeni tako da su eksperti iz njih izdvojili podatke koje misle da su od značaja

Neuronske mreže

- Očekujemo da se neuronske mreže dobro ponašaju (daju bolje rezultate od ostalih modela) kada imamo veliku količinu podataka u sirovoj formi
- To znači da nisu već prerađeni tako da su eksperti iz njih izdvojili podatke koje misle da su od značaja
- To mogu biti signali, slike, zvuk, pa i tekst, u sirovom obliku, bez da su na osnovu njih izdvojeni neki atributi

Neuronske mreže

- Očekujemo da se neuronske mreže dobro ponašaju (daju bolje rezultate od ostalih modela) kada imamo veliku količinu podataka u sirovoj formi
- To znači da nisu već prerađeni tako da su eksperti iz njih izdvojili podatke koje misle da su od značaja
- To mogu biti signali, slike, zvuk, pa i tekst, u sirovom obliku, bez da su na osnovu njih izdvojeni neki atributi
- Ključna snaga neuronskih mreža je da same uoče sta je to u signalu koji obrađuju zapravo relevantno za reăvanje problema

Neuronske mreže

- Očekujemo da se neuronske mreže dobro ponašaju (daju bolje rezultate od ostalih modela) kada imamo veliku količinu podataka u sirovoj formi
- To znači da nisu već prerađeni tako da su eksperti iz njih izdvojili podatke koje misle da su od značaja
- To mogu biti signali, slike, zvuk, pa i tekst, u sirovom obliku, bez da su na osnovu njih izdvojeni neki atributi
- Ključna snaga neuronskih mreža je da same uoče sta je to u signalu koji obrađuju zapravo relevantno za reăvanje problema
- Na primer, za obradu teksta, može nam delovati da je bitno da znamo npr. frekvencije nekih reči, ali neuronske mreže mogu da uoče da su bitni neki drugi atributi koje možda ne možemo nikako intuitivno interpretirati

Neuronske mreže

- Kod tabelarnih podataka, već postoje neki atributi koji su organizovani po kolonama

Neuronske mreže

- Kod tabelarnih podataka, već postoje neki atributi koji su organizovani po kolonama
- Ako imamo male količine tabelarnih podataka, nema razloga da verujemo da će neuronska mreža dati dobre rezultate i često su tada bolji izbor ansambli poput slučajnih šuma

Neuronske mreže

- Kod tabelarnih podataka, već postoje neki atributi koji su organizovani po kolonama
- Ako imamo male količine tabelarnih podataka, nema razloga da verujemo da će neuronska mreža dati dobre rezultate i često su tada bolji izbor ansambli poput slučajnih šuma
- Ako smo između (imamo male količine sirovih podataka ili velike količine tabelarnih podataka), teško je bilo šta reći o izboru modela

Neuronske mreže

- Postoje različite vrste neuronskih mreža

Neuronske mreže

- Postoje različite vrste neuronskih mreža
- Osnovnu varijantu predstavljaju *potpuno povezane neuronske mreže*

Neuronske mreže

- Postoje različite vrste neuronskih mreža
- Osnovnu varijantu predstavljaju *potpuno povezane neuronske mreže*
- U obradi slika i drugih vrsta signala, pa i teksta, vrlo su popularne *konvolutivne neuronske mreže*

Neuronske mreže

- Postoje različite vrste neuronskih mreža
- Osnovnu varijantu predstavljaju *potpuno povezane neuronske mreže*
- U obradi slika i drugih vrsta signala, pa i teksta, vrlo su popularne *konvolutivne neuronske mreže*
- Za obradu podataka nalik nizovima promenljive dužine, najčešće se koriste *rekurentne neuronske mreže*

Neuronske mreže

- Postoje različite vrste neuronskih mreža
- Osnovnu varijantu predstavljaju *potpuno povezane neuronske mreže*
- U obradi slika i drugih vrsta signala, pa i teksta, vrlo su popularne *konvolutivne neuronske mreže*
- Za obradu podataka nalik nizovima promenljive dužine, najčešće se koriste *rekurentne neuronske mreže*
- Za obradu podataka koji se mogu predstaviti stablima koriste se *rekurzivne neuronske mreže*

Neuronske mreže

- Postoje različite vrste neuronskih mreža
- Osnovnu varijantu predstavljaju *potpuno povezane neuronske mreže*
- U obradi slika i drugih vrsta signala, pa i teksta, vrlo su popularne *konvolutivne neuronske mreže*
- Za obradu podataka nalik nizovima promenljive dužine, najčešće se koriste *rekurentne neuronske mreže*
- Za obradu podataka koji se mogu predstaviti stablima koriste se *rekurzivne neuronske mreže*
- Za obradu podataka koji se predstavljaju grafovima koriste se *grafovske neuronske mreže*

Pregled

Potpuno povezane neuronske mreže

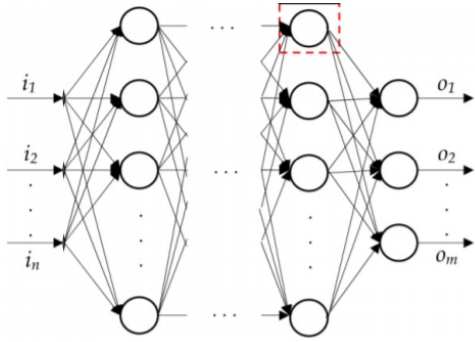
Konvolutivne neuronske mreže

Rekurentne neuronske mreže

Praktične tehnike i napredni koncepti

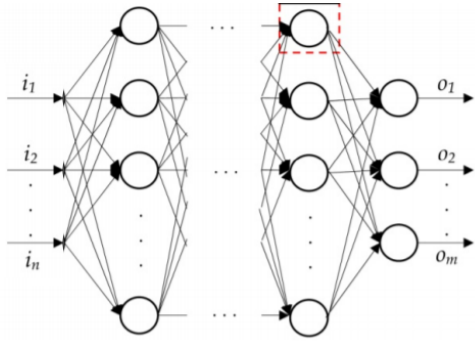
Potpuno povezane neuronske mreže

- Kružiće nazivamo *neuronima* ili *jedinicama*



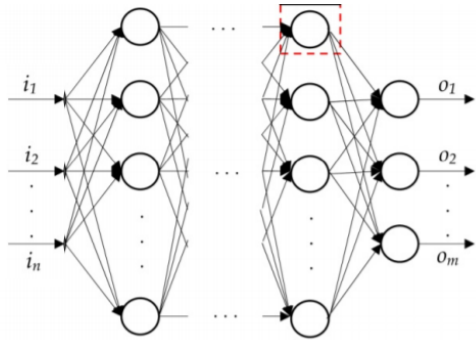
Potpuno povezane neuronske mreže

- Kružiće nazivamo *neuronima* ili *jedinicama*
- Svaki neuron sprovodi jednu jednostavnu nelinearnu transformaciju (izračunava jednostavnu funkciju)



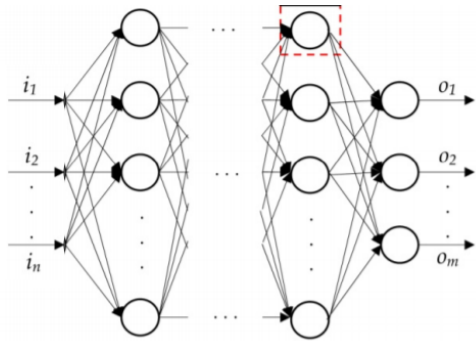
Potpuno povezane neuronske mreže

- Kružiće nazivamo *neuronima* ili *jedinicama*
- Svaki neuron sprovodi jednu jednostavnu nelinearnu transformaciju (izračunava jednostavnu funkciju)
- Udružujući veliki broj ovakvih neurona u jednu mrežu, očekujemo da su u stanju da urade vrlo kompleksne transformacije



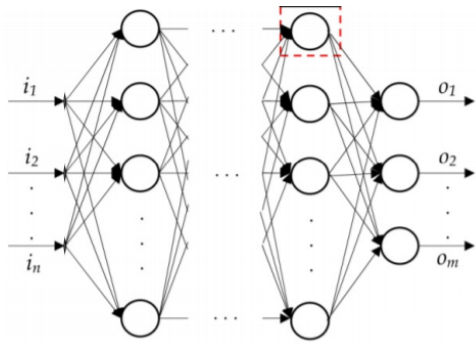
Potpuno povezane neuronske mreže

- Ulaz $i_1, \ldots, i_n$ predstavlja vektor atributa kakve smo imali i u drugim modelima
- Ulazni vektor možemo posmatrati kao argument funkcija koje se izračunavaju u neuronima
- Na kraju se nalaze izlazne jedinice $o_1, \ldots, o_m$ koje daju izlaz mreže



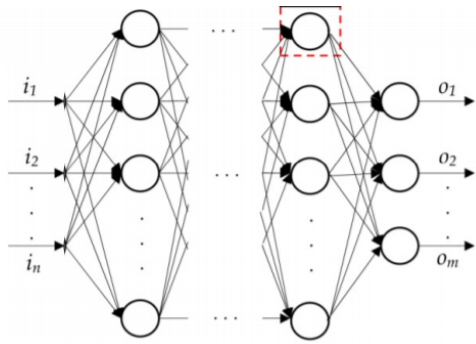
Potpuno povezane neuronske mreže

- Neuroni su organizovani u slojeve - skupove neurona koji nisu povezani između sebe



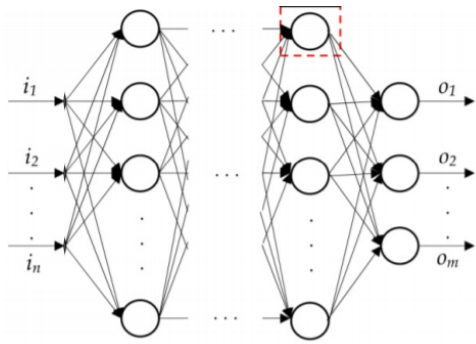
Potpuno povezane neuronske mreže

- Neuroni su organizovani u slojeve - skupove neurona koji nisu povezani između sebe
- Svi slojevi osim poslednjeg se nazivaju *skrivenim slojevima*



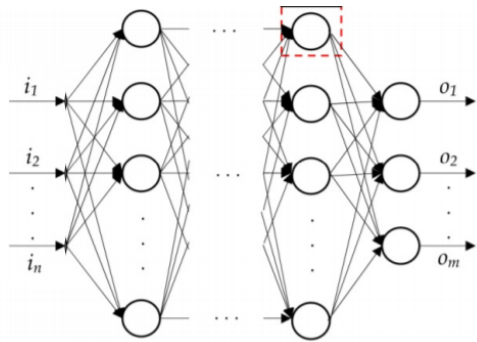
Potpuno povezane neuronske mreže

- Neuroni su organizovani u slojeve - skupove neurona koji nisu povezani između sebe
- Svi slojevi osim poslednjeg se nazivaju *skrivenim slojevima*
- Svaki neuron u jednom sloju izračunava vrednost koju prosleđuje neuronima u narednom sloju



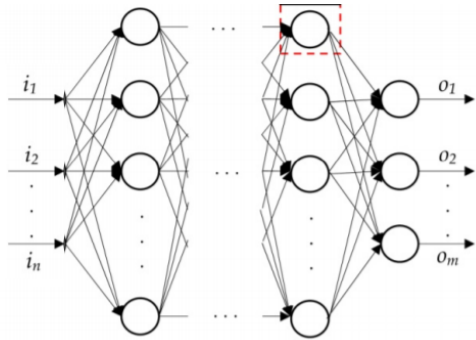
Potpuno povezane neuronske mreže

- Neuroni su organizovani u slojeve - skupove neurona koji nisu povezani između sebe
- Svi slojevi osim poslednjeg se nazivaju *skrivenim slojevima*
- Svaki neuron u jednom sloju izračunava vrednost koju prosleđuje neuronima u narednom sloju
- Potpuna povezanost se odnosi na to da je svaki neuron jednog sloja povezan sa svakim neuronom sledećeg sloja, odnosno da svaki neuron jednog sloja prosleđuje svoj izlaz svakom neuronu sledećeg sloja



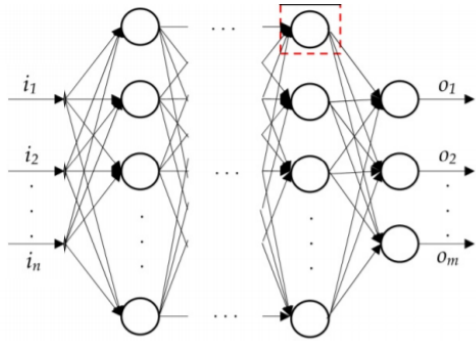
Potpuno povezane neuronske mreže

- Vezu između neurona smo predstavi šematski



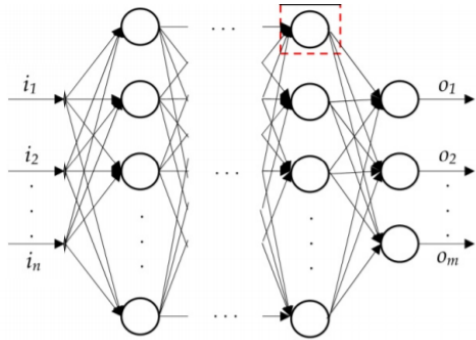
Potpuno povezane neuronske mreže

- Vezu između neurona smo predstavi šematski
- Da vidimo sada šta ta veza predstavlja matematički



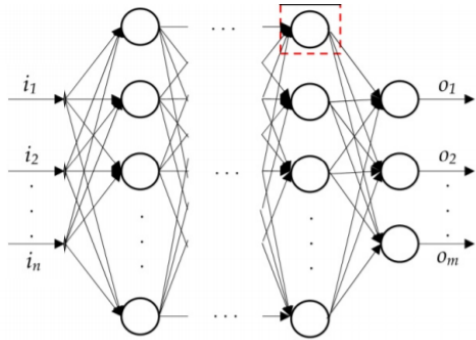
Potpuno povezane neuronske mreže

- Vezu između neurona smo predstavi šematski
- Da vidimo sada šta ta veza predstavlja matematički
- Strelice između neurona označavaju *komponovanje* funkcija

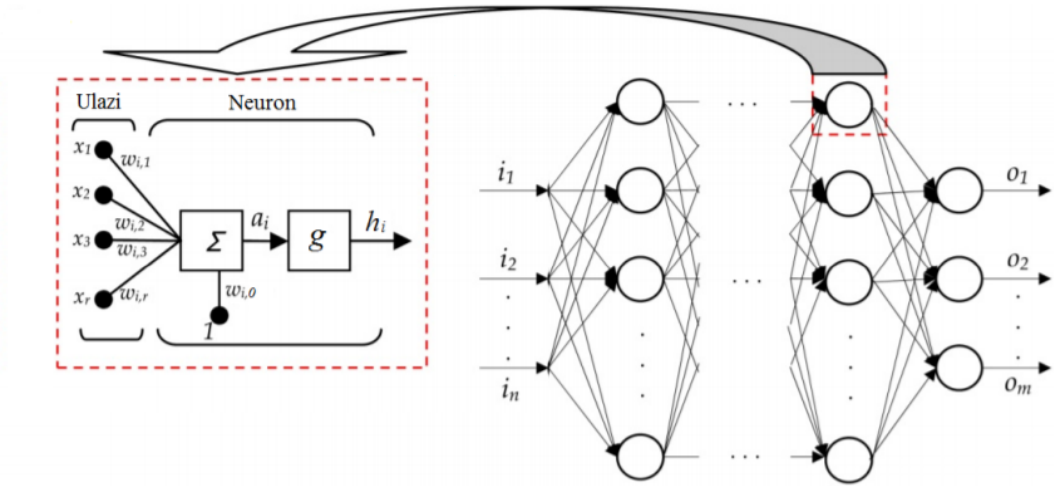


Potpuno povezane neuronske mreže

- Vezu između neurona smo predstavi šematski
- Da vidimo sada šta ta veza predstavlja matematički
- Strelice između neurona označavaju *komponovanje* funkcija
- Ova neuronska mreža je jedna složena funkcija nastala kompozicijom velikog broja jednostavnih funkcija



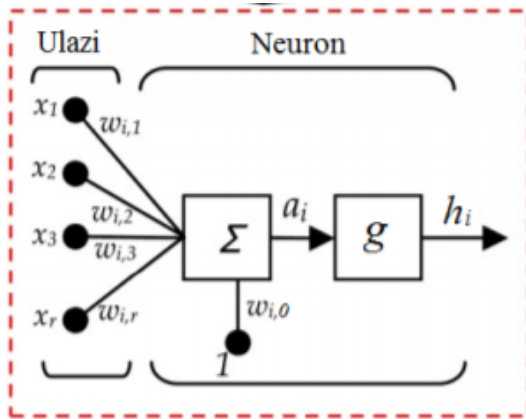
Potpuno povezane neuronske mreže



- Razmotrimo koje se funkcije nalaze unutar jednog neurona

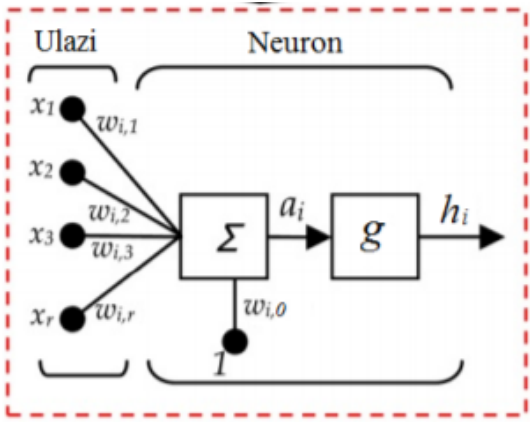
Potpuno povezane neuronske mreže

- Ulaz u neuron su promenljive $x_1, \ldots, x_r$
- Promenljive $w_{i,1}, \ldots, w_{i,r}$ predstavljaju parametre modela
- Svaki neuron je parametrizovana funkcija koja najpre vrši linearnu kombinaciju svojih ulaza (linearni model, kao kod linearne regresije ili kod logističke regresije) a onda vrednost tog linearnog modela prosleđuje u funkciju g
- Funkcija g je neka nelinearna funkcija koju nazivamo *aktivacionom funkcijom*



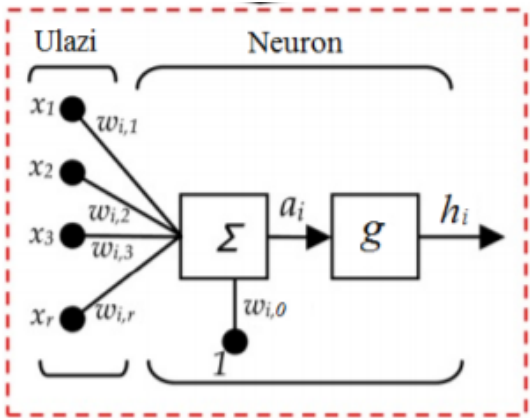
Potpuno povezane neuronske mreže

- Na primer, ukoliko za g uzmemo sigmoidnu funkciju i onda će jedan neuron predstavljati jedan model logističke regresije



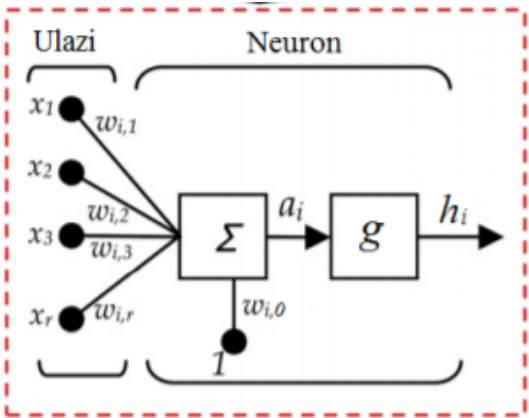
Potpuno povezane neuronske mreže

- Na primer, ukoliko za g uzmemo sigmoidnu funkciju i onda će jedan neuron predstavljati jedan model logističke regresije
- Ukoliko to uradimo za svaki neuron, onda će neuronska mreža biti kompozicija velikog broja logističkih modela



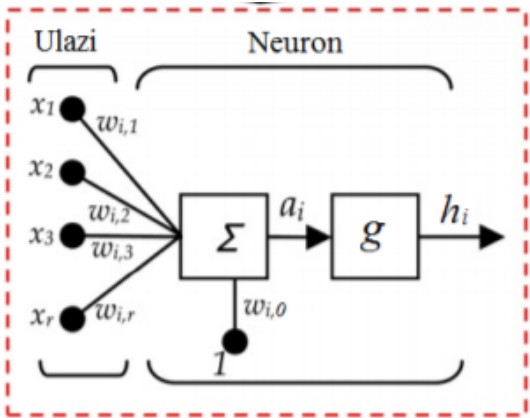
Potpuno povezane neuronske mreže

- Na primer, ukoliko za g uzmemo sigmoidnu funkciju i onda će jedan neuron predstavljati jedan model logističke regresije
- Ukoliko to uradimo za svaki neuron, onda će neuronska mreža biti kompozicija velikog broja logističkih modela
- Funkcija g treba da zadovoljava neka svojstva - neprekidnost, diferencijabilnost i monotonost



Potpuno povezane neuronske mreže

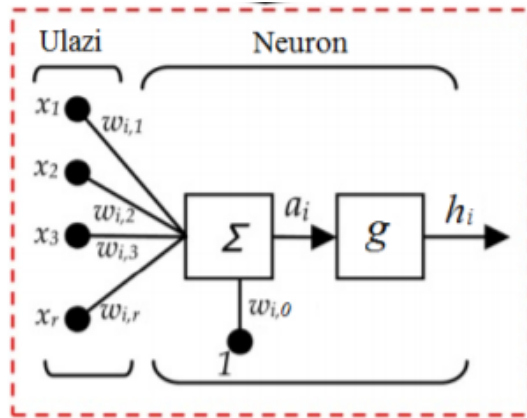
- Na primer, ukoliko za g uzmemo sigmoidnu funkciju i onda će jedan neuron predstavljati jedan model logističke regresije
- Ukoliko to uradimo za svaki neuron, onda će neuronska mreža biti kompozicija velikog broja logističkih modela
- Funkcija g treba da zadovoljava neka svojstva - neprekidnost, diferencijabilnost i monotonost
- Sigmoidna funkcija zadovoljava ove uslove i stoga predstavlja tipičnu aktivacionu funkciju, a ima i drugih



Formulacija modela

- Formalno, model se definiše na sledeći način:

$$h_0 = x$$
$$h_i = g(W_i h_{i-1} + w_{i0}) \qquad i = 1, 2, \ldots, L$$

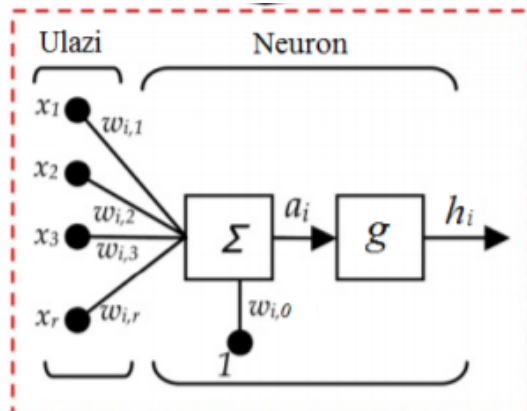


Formulacija modela

- Formalno, model se definiše na sledeći način:

$$h_0 = x$$
$$h_i = g(W_i h_{i-1} + w_{i0}) \qquad i = 1, 2, \ldots, L$$

- L je broj slojeva

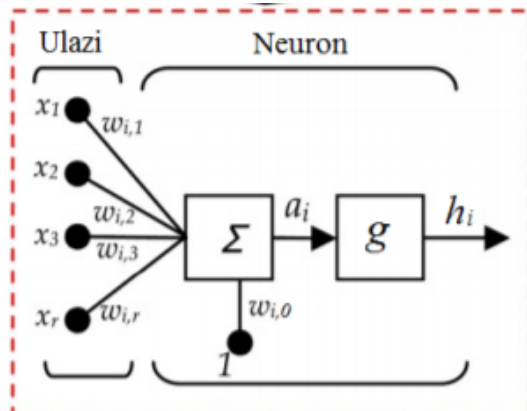


Formulacija modela

- Formalno, model se definiše na sledeći način:

$$h_0 = x$$
$$h_i = g(W_i h_{i-1} + w_{i0}) \qquad i = 1, 2, \ldots, L$$

- L je broj slojeva
- W_i je matrica čija j-ta vrsta predstavlja vektor vrednosti parametara jedinice j u sloju i

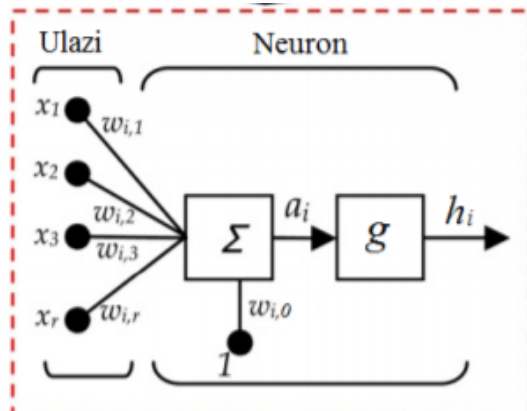


Formulacija modela

- Formalno, model se definiše na sledeći način:

$$h_0 = x$$
$$h_i = g(W_i h_{i-1} + w_{i0}) \qquad i = 1, 2, \ldots, L$$

- L je broj slojeva
- W_i je matrica čija j-ta vrsta predstavlja vektor vrednosti parametara jedinice j u sloju i
- Za vektor v, $g(v)$, predstavlja vektor $(g(v_1), g(v_2) \ldots, g(v_m))^T$, gde je m dimenzionalnost vektora



Teorema o univerzalnoj aproksimaciji

Teorema. Neka je g ograničena i strogo rastuća neprekidna funkcija. Tada za svaku funkciju $f \in C[0,1]^n$ i svako $\varepsilon > 0$, postoji broj $m \in \mathbb{N}$, matrica $W \in \mathbb{R}^{m \times n}$, vektor $w_0 \in \mathbb{R}^m$ i vektor $v \in \mathbb{R}^m$, tako da za svako $x \in [0,1]^n$ važi

$$|v^T g(Wx + w_0) - f(x)| < \varepsilon$$

- Izražava bitno svojstvo neuronskih mreža koje ne važi za ostale ML modele

Teorema o univerzalnoj aproksimaciji

Teorema. Neka je g ograničena i strogo rastuća neprekidna funkcija. Tada za svaku funkciju $f \in C[0,1]^n$ i svako $\varepsilon > 0$, postoji broj $m \in \mathbb{N}$, matrica $W \in \mathbb{R}^{m \times n}$, vektor $w_0 \in \mathbb{R}^m$ i vektor $v \in \mathbb{R}^m$, tako da za svako $x \in [0,1]^n$ važi

$$|v^T g(Wx + w_0) - f(x)| < \varepsilon$$

- Izražava bitno svojstvo neuronskih mreža koje ne važi za ostale ML modele
- Svaka funkcija koja je neprekidna na kompaktnom skupu može se proizvoljno dobro aproksimirati neuronskom mrežom sa jednim skrivenim slojem i dovoljnim brojem neurona

Teorema o univerzalnoj aproksimaciji

Teorema. Neka je g ograničena i strogo rastuća neprekidna funkcija. Tada za svaku funkciju $f \in C[0,1]^n$ i svako $\varepsilon > 0$, postoji broj $m \in \mathbb{N}$, matrica $W \in \mathbb{R}^{m \times n}$, vektor $w_0 \in \mathbb{R}^m$ i vektor $v \in \mathbb{R}^m$, tako da za svako $x \in [0,1]^n$ važi

$$|v^T g(Wx + w_0) - f(x)| < \varepsilon$$

- Izražava bitno svojstvo neuronskih mreža koje ne važi za ostale ML modele
- Svaka funkcija koja je neprekidna na kompaktnom skupu može se proizvoljno dobro aproksimirati neuronskom mrežom sa jednim skrivenim slojem i dovoljnim brojem neurona
- Liči na teoremu o separabilnosti neprekidnih funkcija (da se mogu polinomima proizvoljno dobro aproksimirati)

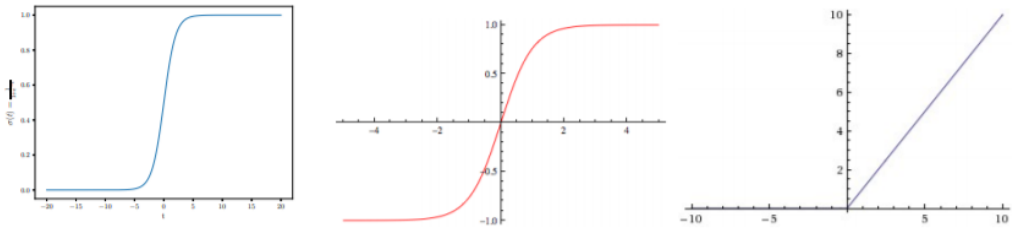
Teorema o univerzalnoj aproksimaciji

Teorema. Neka je g ograničena i strogo rastuća neprekidna funkcija. Tada za svaku funkciju $f \in C[0,1]^n$ i svako $\varepsilon > 0$, postoji broj $m \in \mathbb{N}$, matrica $W \in \mathbb{R}^{m \times n}$, vektor $w_0 \in \mathbb{R}^m$ i vektor $v \in \mathbb{R}^m$, tako da za svako $x \in [0,1]^n$ važi

$$|v^T g(Wx + w_0) - f(x)| < \varepsilon$$

- Izražava bitno svojstvo neuronskih mreža koje ne važi za ostale ML modele
- Svaka funkcija koja je neprekidna na kompaktnom skupu može se proizvoljno dobro aproksimirati neuronskom mrežom sa jednim skrivenim slojem i dovoljnim brojem neurona
- Liči na teoremu o separabilnosti neprekidnih funkcija (da se mogu polinomima proizvoljno dobro aproksimirati)
- Ova teorema nam govori da je moguće konstruisati neuronsku mrežu koja proizvoljno dobro aproksimira neku funkciju ali ne i da je to lako kao ni da je to nužno dobro

Aktivacione funkcije

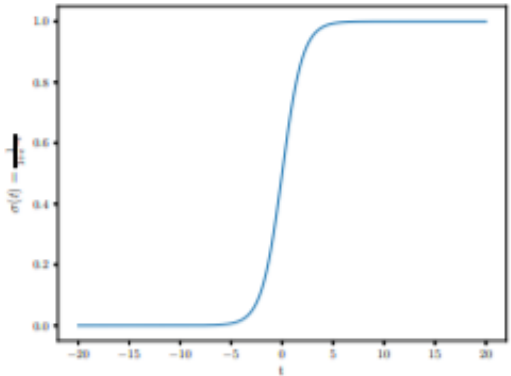


Slika 10.2: Najčešće korišćene aktivacione funkcije – sigmoidna, tangens hiperbolički i ispravljačka linearna jedinica.

Aktivacione funkcije

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

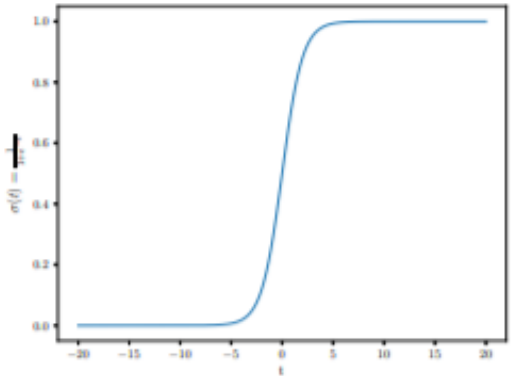
- *sigmoidna funkcija* je široko korišćena u mašinskom učenju i dugo je bila najčešće korišćena aktivaciona funkcija u neuronskim mrežama



Aktivacione funkcije

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

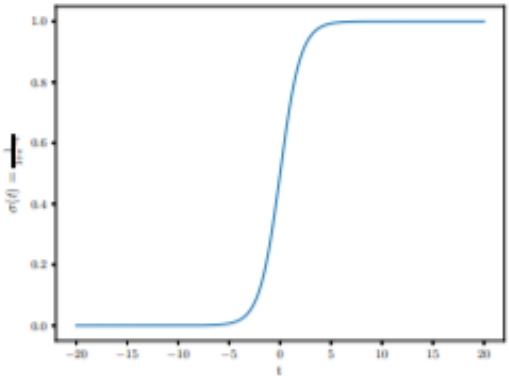
- *sigmoidna funkcija* je široko korišćena u mašinskom učenju i dugo je bila najčešće korišćena aktivaciona funkcija u neuronskim mrežama
- Ipak, ova funkcija u praksi nije najpogodnija za upotrebu, pre svega zbog problema u optimizaciji



Aktivacione funkcije

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

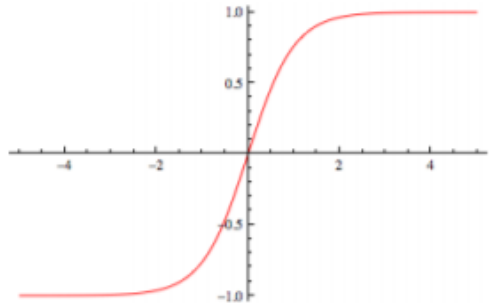
- *sigmoidna funkcija* je široko korišćena u mašinskom učenju i dugo je bila najčešće korišćena aktivaciona funkcija u neuronskim mrežama
- Ipak, ova funkcija u praksi nije najpogodnija za upotrebu, pre svega zbog problema u optimizaciji
- Naime, ona je skoro konstantna osim u okolini nule, pa stoga dovodi praktično do anuliranja gradijenta, što otežava ili onemogućava učenje



Aktivacione funkcije

$$\tanh(x) = \frac{e^{2x} - 1}{e^{2x} + 1}$$

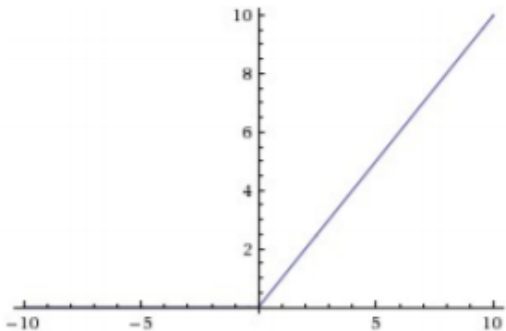
- tangens hiperbolički je takođe bila u širokoj upotrebi
- obično sa nešto većim uspehom od sigmoidne funkcije sa kojom je vrlo srodna (važi $\tanh(x) = 2\sigma(2x) - 1$), pre svega zato što je u okolini nule bliska identitetu, što čini model sličnijim linearnom i u nekoj meri olakšava optimizaciju



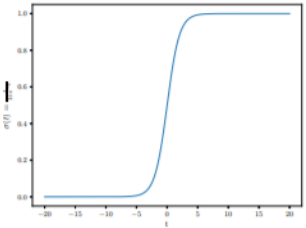
Aktivacione funkcije

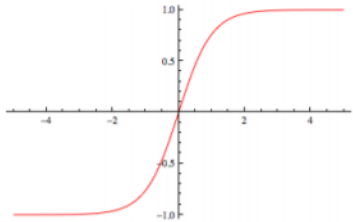
$$rlu(x) = \max(0, x)$$

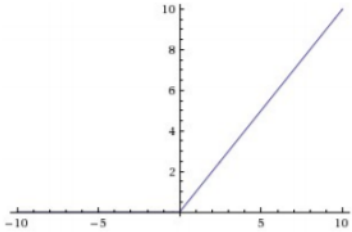
- *ispravljena linearna jedinica* (eng. *rectified linear unit*, skraćeno ReLu) predstavlja trenutno najčešće korišćenu aktivacionu funkciju.
- Razlog za njenu popularnost su pogodnija svojstva pri optimizaciji, uprkos svojoj nediferencijabilnosti.
- Naime, šanse da se naleti na tačku nediferencijabilnosti u procesu optimizacije nisu velike, a i ako se to desi, u kasnijem toku optimizacije greška će tipično biti nadomešćena



Aktivacione funkcije

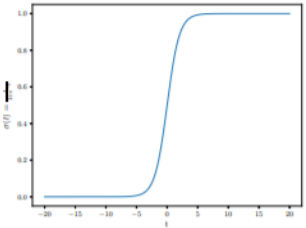


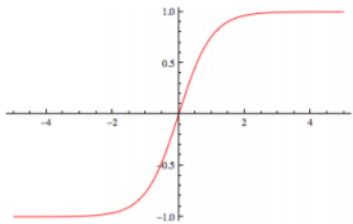


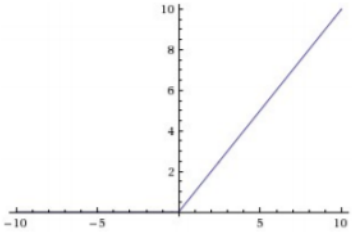


- Analizirajmo zašto je ReLU pogodnija za optimizaciju od sigmoidne funkcije i tanh

Aktivacione funkcije

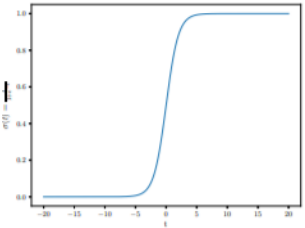


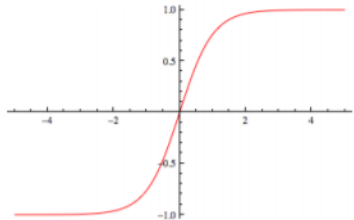


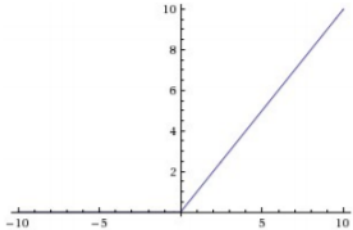


- Analizirajmo zašto je ReLU pogodnija za optimizaciju od sigmoidne funkcije i tanh
- Kod ReLu, izvod je konstantan svuda, u pozitivnom delu x ose je jednak 1 (što pogoduje gradijentnim metodama) a u negativnom 0 (što ne pogoduje gradijentnim metodama jer tada nema kretanja)

Aktivacione funkcije

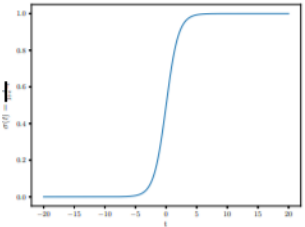


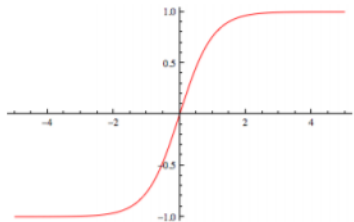


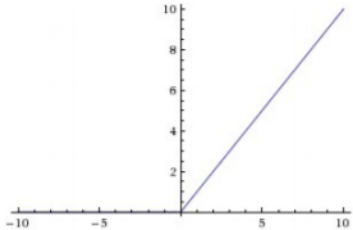


- Analizirajmo zašto je ReLU pogodnija za optimizaciju od sigmoidne funkcije i tanh
- Kod ReLu, izvod je konstantan svuda, u pozitivnom delu x ose je jednak 1 (što pogoduje gradijentnim metodama) a u negativnom 0 (što ne pogoduje gradijentnim metodama jer tada nema kretanja)
- I σ i tanh u regionu oko nule imaju nagib (izvod nije 0) ali u ostalim delovima nisu konstantne, ali kao da jesu (izvod kao da je 0)

Aktivacione funkcije



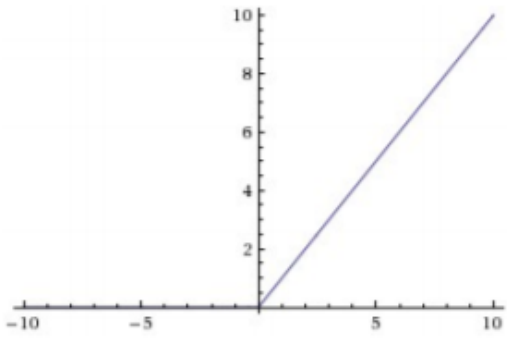




- Analizirajmo zašto je ReLU pogodnija za optimizaciju od sigmoidne funkcije i tanh
- Kod ReLu, izvod je konstantan svuda, u pozitivnom delu x ose je jednak 1 (što pogoduje gradijentnim metodama) a u negativnom 0 (što ne pogoduje gradijentnim metodama jer tada nema kretanja)
- I σ i tanh u regionu oko nule imaju nagib (izvod nije 0) ali u ostalim delovima nisu konstantne, ali kao da jesu (izvod kao da je 0)
- ReLu je konstantna samo ulevo (izvod je 0) a udesno je izvod uvek jednak 1

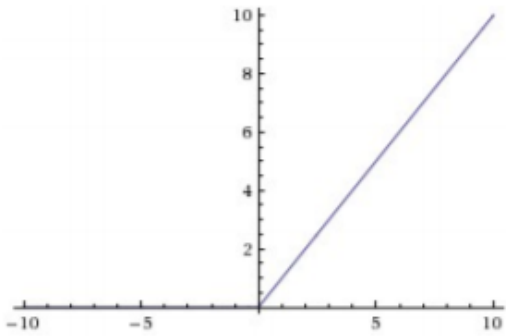
Aktivacione funkcije

- ReLu je konstantna samo ulevo (izvod je 0) a udesno je izvod uvek jednak 1



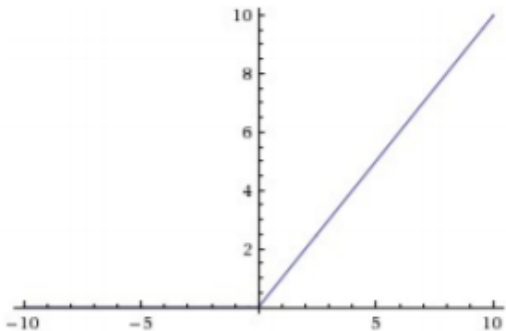
Aktivacione funkcije

- ReLu je konstantna samo ulevo (izvod je 0) a udesno je izvod uvek jednak 1
- Jedna modifikovana varijanta je *nakošena ispravljena linearna jedinica* (eng. *leaky rectified linear unit*), koja levo od nule uzima vrednost αx, za malu vrednost parametra α, poput 0.01.



Aktivacione funkcije

- ReLu je konstantna samo ulevo (izvod je 0) a udesno je izvod uvek jednak 1
- Jedna modifikovana varijanta je *nakošena ispravljena linearna jedinica* (eng. *leaky rectified linear unit*), koja levo od nule uzima vrednost αx, za malu vrednost parametra α, poput 0.01.
- Umesto da je levi deo funkcije ravan, biće nagnut pod malim uglom i izvod će biti različit od nule



Izlazne jedinice i greška

- Neuronske mreže se mogu koristiti kako za regresiju funkcija sa vrednostima iz $\mathbb{R}^n$, tako i za klasifikaciju.

Izlazne jedinice i greška

- Neuronske mreže se mogu koristiti kako za regresiju funkcija sa vrednostima iz $\mathbb{R}^n$, tako i za klasifikaciju.
- U zavisnosti od problema koji rešavamo, razlikuju se aktivacione funkcije na poslednjem, izlaznom sloju neuronske mreže

Izlazne jedinice i greška

- Neuronske mreže se mogu koristiti kako za regresiju funkcija sa vrednostima iz $\mathbb{R}^n$, tako i za klasifikaciju.
- U zavisnosti od problema koji rešavamo, razlikuju se aktivacione funkcije na poslednjem, izlaznom sloju neuronske mreže
- Ako radimo regresiju, onda bismo u izlaznom sloju imali onoliko neurona koliko promenljivih predviđamo

Izlazne jedinice i greška

- Neuronske mreže se mogu koristiti kako za regresiju funkcija sa vrednostima iz $\mathbb{R}^n$, tako i za klasifikaciju.
- U zavisnosti od problema koji rešavamo, razlikuju se aktivacione funkcije na poslednjem, izlaznom sloju neuronske mreže
- Ako radimo regresiju, onda bismo u izlaznom sloju imali onoliko neurona koliko promenljivih predviđamo
- Kod regresionih neuronskih mreža, poslednji sloj neurona neće imati aktivacionu funkciju

Izlazne jedinice i greška

- Neuronske mreže se mogu koristiti kako za regresiju funkcija sa vrednostima iz $\mathbb{R}^n$, tako i za klasifikaciju.
- U zavisnosti od problema koji rešavamo, razlikuju se aktivacione funkcije na poslednjem, izlaznom sloju neuronske mreže
- Ako radimo regresiju, onda bismo u izlaznom sloju imali onoliko neurona koliko promenljivih predviđamo
- Kod regresionih neuronskih mreža, poslednji sloj neurona neće imati aktivacionu funkciju
  - kad predviđamo neprekidne promenljive, treba da imamo mogućnost da predvidimo sve vrednosti u intervalu $(-\infty, \infty)$ a ako primenimo neku od pomenutih aktivacionih funkcija, suzićemo interval

Izlazne jedinice i greška

- Neuronske mreže se mogu koristiti kako za regresiju funkcija sa vrednostima iz $\mathbb{R}^n$, tako i za klasifikaciju.
- U zavisnosti od problema koji rešavamo, razlikuju se aktivacione funkcije na poslednjem, izlaznom sloju neuronske mreže
- Ako radimo regresiju, onda bismo u izlaznom sloju imali onoliko neurona koliko promenljivih predviđamo
- Kod regresionih neuronskih mreža, poslednji sloj neurona neće imati aktivacionu funkciju
  - kad predviđamo neprekidne promenljive, treba da imamo mogućnost da predvidimo sve vrednosti u intervalu $(-\infty, \infty)$ a ako primenimo neku od pomenutih aktivacionih funkcija, suzićemo interval
- S obzirom da izlazni neuroni u regresionoj mreži nemaju aktivacionu funkciju, možemo ih posmatrati kao neurone koji rade linearnu regresiju nad izlazima prethodnog sloja

Izlazne jedinice i greška

- Neuronske mreže se mogu koristiti kako za regresiju funkcija sa vrednostima iz $\mathbb{R}^n$, tako i za klasifikaciju.
- U zavisnosti od problema koji rešavamo, razlikuju se aktivacione funkcije na poslednjem, izlaznom sloju neuronske mreže
- Ako radimo regresiju, onda bismo u izlaznom sloju imali onoliko neurona koliko promenljivih predviđamo
- Kod regresionih neuronskih mreža, poslednji sloj neurona neće imati aktivacionu funkciju
  - kad predviđamo neprekidne promenljive, treba da imamo mogućnost da predvidimo sve vrednosti u intervalu $(-\infty, \infty)$ a ako primenimo neku od pomenutih aktivacionih funkcija, suzićemo interval
- S obzirom da izlazni neuroni u regresionoj mreži nemaju aktivacionu funkciju, možemo ih posmatrati kao neurone koji rade linearnu regresiju nad izlazima prethodnog sloja
- Greška se definiše kao srednje kvadratna greška

Izlazne jedinice i greška

- Kod klasifikacije, ako predviđamo jednu promenljivu, imaćemo onoliko neurona koliko imamo klasa

Izlazne jedinice i greška

- Kod klasifikacije, ako predviđamo jednu promenljivu, imaćemo onoliko neurona koliko imamo klasa
- Svaki od tih neurona će biti jedan linearni model

Izlazne jedinice i greška

- Kod klasifikacije, ako predviđamo jednu promenljivu, imaćemo onoliko neurona koliko imamo klasa
- Svaki od tih neurona će biti jedan linearni model
- Kao aktivacionu funkciju ćemo upotrebiti *softmax* kog se sećamo iz multinomijalne logističke regresije

Izlazne jedinice i greška

- Kod klasifikacije, ako predviđamo jednu promenljivu, imaćemo onoliko neurona koliko imamo klasa
- Svaki od tih neurona će biti jedan linearni model
- Kao aktivacionu funkciju ćemo upotrebiti *softmax* kog se sećamo iz multinomijalne logističke regresije
- Softmax neće biti unutar svakog neurona posebno već će biti primenjena na sve neurone, kao kod multinomijalne logističke regresije

$$softmax(x) = \left(\frac{e^{x_1}}{\sum_{i=1}^{C} e^{x_i}}, \ldots, \frac{e^{x_C}}{\sum_{i=1}^{C} e^{x_i}} \right)$$

Izlazne jedinice i greška

- Kod klasifikacije, ako predviđamo jednu promenljivu, imaćemo onoliko neurona koliko imamo klasa
- Svaki od tih neurona će biti jedan linearni model
- Kao aktivacionu funkciju ćemo upotrebiti *softmax* kog se sećamo iz multinomijalne logističke regresije
- Softmax neće biti unutar svakog neurona posebno već će biti primenjena na sve neurone, kao kod multinomijalne logističke regresije

$$softmax(x) = \left(\frac{e^{x_1}}{\sum_{i=1}^{C} e^{x_i}}, \ldots, \frac{e^{x_C}}{\sum_{i=1}^{C} e^{x_i}} \right)$$

- Softmax funkcija transformiše izlaze tako da su nenegativni i sumiraju se na 1 i stoga se mogu interpretirati kao raspodela verovatnoće po mogućim klasama, pri čemu za svaku klasu postoji tačno jedan izlaz u poslednjim sloju

Izlazne jedinice i greška

- Kod klasifikacije, ako predviđamo jednu promenljivu, imaćemo onoliko neurona koliko imamo klasa
- Svaki od tih neurona će biti jedan linearni model
- Kao aktivacionu funkciju ćemo upotrebiti *softmax* kog se sećamo iz multinomijalne logističke regresije
- Softmax neće biti unutar svakog neurona posebno već će biti primenjena na sve neurone, kao kod multinomijalne logističke regresije

$$softmax(x) = \left(\frac{e^{x_1}}{\sum_{i=1}^{C} e^{x_i}}, \ldots, \frac{e^{x_C}}{\sum_{i=1}^{C} e^{x_i}} \right)$$

- Softmax funkcija transformiše izlaze tako da su nenegativni i sumiraju se na 1 i stoga se mogu interpretirati kao raspodela verovatnoće po mogućim klasama, pri čemu za svaku klasu postoji tačno jedan izlaz u poslednjim sloju
- Greška se definiše kao negativna vrednost logaritma verodostojnosti parametara

Izlazne jedinice i greška

- Regresija se može videti kao linearna regresija nad atributima koje konstruiše poslednji sloj

Izlazne jedinice i greška

- Regresija se može videti kao linearna regresija nad atributima koje konstruiše poslednji sloj
- Klasifikacija može videti kao multinomijalna logistička regresija nad atributima koje konstruiše poslednji sloj

Izlazne jedinice i greška

- Regresija se može videti kao linearna regresija nad atributima koje konstruiše poslednji sloj
- Klasifikacija može videti kao multinomijalna logistička regresija nad atributima koje konstruiše poslednji sloj
- Jedino kad imamo binarnu klasifikaciju nam je dovoljan samo jedan neuron kao kod logističke regresije i onda klasifikaciona neuronska mreža binarni slučaj predstavlja logistučku regresiju nad atributima koje je izračunao poslednji skriveni sloj

Kako se vrši trening?

- Parametri bi se mogli varirati na razne načine kako bi se videlo u kom pravcu se može izvršiti pomak kako bise dobila manja vrednost greške

Kako se vrši trening?

- Parametri bi se mogli varirati na razne načine kako bi se videlo u kom pravcu se može izvršiti pomak kako bise dobila manja vrednost greške
- Takav postupak je računski skup

Kako se vrši trening?

- Parametri bi se mogli varirati na razne načine kako bi se videlo u kom pravcu se može izvršiti pomak kako bise dobila manja vrednost greške
- Takav postupak je računski skup
- Gradijent daje pravac pomeranja u kojem se greška *lokalno* najbrže povećava

Kako se vrši trening?

- Parametri bi se mogli varirati na razne načine kako bi se videlo u kom pravcu se može izvršiti pomak kako bise dobila manja vrednost greške
- Takav postupak je računski skup
- Gradijent daje pravac pomeranja u kojem se greška *lokalno* najbrže povećava
- Kako izračunati gradijent za grešku neuronske mreže?

Propagacija unazad

- Algoritam propagacije unazad služi za izračunavanje gradijenata računarske mreže

Propagacija unazad

- Algoritam propagacije unazad služi za izračunavanje gradijenata računarske mreže
- Algoritam koji je omogućio efikasan trening neuronskih mreža korišćenjem gradijenta

Propagacija unazad

- Algoritam propagacije unazad služi za izračunavanje gradijenata računarske mreže
- Algoritam koji je omogućio efikasan trening neuronskih mreža korišćenjem gradijenta
- Predstavlja specijalizovan postupak izračunavanja izvoda složene funkcije kada ona ima strukturu neuronske mreže

Propagacija unazad

- Algoritam propagacije unazad služi za izračunavanje gradijenata računarske mreže
- Algoritam koji je omogućio efikasan trening neuronskih mreža korišćenjem gradijenta
- Predstavlja specijalizovan postupak izračunavanja izvoda složene funkcije kada ona ima strukturu neuronske mreže
- Zasnovan na pravilu za izvod kompozicije funkcija

Propagacija unazad

- Algoritam propagacije unazad služi za izračunavanje gradijenata računarske mreže
- Algoritam koji je omogućio efikasan trening neuronskih mreža korišćenjem gradijenta
- Predstavlja specijalizovan postupak izračunavanja izvoda složene funkcije kada ona ima strukturu neuronske mreže
- Zasnovan na pravilu za izvod kompozicije funkcija
- Neka su funkcije $g : \mathbb{R}^m \to \mathbb{R}^n$ i $f : \mathbb{R}^n \to \mathbb{R}$

$$\partial_i(f \circ g) = \sum_{j=1}^{n} (\partial_j f \circ g)\partial_i g_j$$

Izvod složene funkcije - primer

$$f(g(h(x)))' = \underbrace{f'(g(h(x)))}_{d} g(h(x))' = \underbrace{f'(g(h(x)))g'(h(x))}_{d} h(x)' = \underbrace{f'(g(h(x))g'(h(x))h'(x)}_{d}$$

Izvod složene funkcije - primer

$$f(g(h(x)))' = \underbrace{f'(g(h(x)))}_{d} g(h(x))'$$

- Prva iteracija

Izvod složene funkcije - primer

$$f(g(h(x)))' = \underbrace{f'(g(h(x)))}_{d} g(h(x))'$$

- Prva iteracija
- Najpre izračunamo izvod poslednje funkcije u kompoziciji - to je f'

Izvod složene funkcije - primer

$$f(g(h(x)))' = \underbrace{f'(g(h(x)))}_{d} g(h(x))'$$

- Prva iteracija
- Najpre izračunamo izvod poslednje funkcije u kompoziciji - to je f'
- Ovaj izvod računamo u tački $g(h(x))$

Izvod složene funkcije - primer

$$f(g(h(x)))' = \underbrace{f'(g(h(x)))}_{d} g(h(x))'$$

- Prva iteracija
- Najpre izračunamo izvod poslednje funkcije u kompoziciji - to je f'
- Ovaj izvod računamo u tački $g(h(x))$
- Potrebna nam je vrednost $g(h(x))$, što znači da ćemo ove vrednosti morati jednom da izračunamo *unapred* u neuronskoj mreži i da ih zapamtimo kako bismo ih kasnije iskoristili za računanje gradijenta

Izvod složene funkcije - primer

$$f(g(h(x)))' = \underbrace{f'(g(h(x)))}_{d} g(h(x))'$$

- Prva iteracija
- Najpre izračunamo izvod poslednje funkcije u kompoziciji - to je f'
- Ovaj izvod računamo u tački $g(h(x))$
- Potrebna nam je vrednost $g(h(x))$, što znači da ćemo ove vrednosti morati jednom da izračunamo *unapred* u neuronskoj mreži i da ih zapamtimo kako bismo ih kasnije iskoristili za računanje gradijenta
- Dakle, $g(h(x))$ smo izračunali krećući se kroz mrežu *unapred* a sada se, prilikom računanja gradijenta, vraćamo *unazad*

Izvod složene funkcije - primer

$$f(g(h(x)))' = \underbrace{f'(g(h(x)))}_{d} g(h(x))'$$

- Prva iteracija
- Najpre izračunamo izvod poslednje funkcije u kompoziciji - to je f'
- Ovaj izvod računamo u tački $g(h(x))$
- Potrebna nam je vrednost $g(h(x))$, što znači da ćemo ove vrednosti morati jednom da izračunamo *unapred* u neuronskoj mreži i da ih zapamtimo kako bismo ih kasnije iskoristili za računanje gradijenta
- Dakle, $g(h(x))$ smo izračunali krećući se kroz mrežu *unapred* a sada se, prilikom računanja gradijenta, vraćamo *unazad*
- Izračunati izvod $f'(g(h(x)))$ čuvamo u promenljivoj d i u narednoj iteraciji ga *proširujemo*

Izvod složene funkcije - primer

$$f(g(h(x)))' = \underbrace{f'(g(h(x)))}_{d} g(h(x))' = \underbrace{f'(g(h(x)))g'(h(x))}_{d} h(x)'$$

- Druga iteracija

Izvod složene funkcije - primer

$$f(g(h(x)))' = \underbrace{f'(g(h(x)))}_{d} g(h(x))' = \underbrace{f'(g(h(x)))g'(h(x))}_{d} h(x)'$$

- Druga iteracija
- *Proširujemo* prizvod d tako što ćemo ga pomnožiti sa $g'(h(x))$

Izvod složene funkcije - primer

$$f(g(h(x)))' = \underbrace{f'(g(h(x)))}_{d} g(h(x))' = \underbrace{f'(g(h(x)))g'(h(x))}_{d} h(x)'$$

- Druga iteracija
- *Proširujemo* prizvod d tako što ćemo ga pomnožiti sa $g'(h(x))$
- Ostaje nam da izračunamo izvod ostatka $h(x)'$ (to nije izvod složene funkcije)

Izvod složene funkcije - primer

$$f(g(h(x)))' = \underbrace{f'(g(h(x)))}_{d} g(h(x))' = \underbrace{f'(g(h(x)))g'(h(x))}_{d} h(x)' = \underbrace{f'(g(h(x))g'(h(x))h'(x)}_{d}$$

- Treća iteracija

Izvod složene funkcije - primer

$$f(g(h(x)))' = \underbrace{f'(g(h(x)))}_{d} g(h(x))' = \underbrace{f'(g(h(x)))g'(h(x))}_{d} h(x)' = \underbrace{f'(g(h(x))g'(h(x))h'(x)}_{d}$$

- Treća iteracija
- *Proširujemo* prizvod d tako što ćemo ga pomnožiti sa $h'(x)$

Izvod složene funkcije - primer

$$f(g(h(x)))' = \underbrace{f'(g(h(x)))}_{d} g(h(x))' = \underbrace{f'(g(h(x)))g'(h(x))}_{d} h(x)' = \underbrace{f'(g(h(x))g'(h(x))h'(x)}_{d}$$

- U terminima neuronskih mreža, ovu računicu možemo iskazati ovako:

Izvod složene funkcije - primer

$$f(g(h(x)))' = \underbrace{f'(g(h(x)))}_{d} g(h(x))' = \underbrace{f'(g(h(x)))g'(h(x))}_{d} h(x)' = \underbrace{f'(g(h(x))g'(h(x))h'(x)}_{d}$$

- U terminima neuronskih mreža, ovu računicu možemo iskazati ovako:
- Imamo do sada izračunati parcijalni izvod, d

Izvod složene funkcije - primer

$$f(g(h(x)))' = \underbrace{f'(g(h(x)))}_{d} g(h(x))' = \underbrace{f'(g(h(x)))g'(h(x))}_{d} h(x)' = \underbrace{f'(g(h(x))g'(h(x))h'(x)}_{d}$$

- U terminima neuronskih mreža, ovu računicu možemo iskazati ovako:
- Imamo do sada izračunati parcijalni izvod, d
- Proširujemo taj proizvod izvodom aktivacione funkcije za tekući nivo

Izvod složene funkcije - primer

$$f(g(h(x)))' = \underbrace{f'(g(h(x)))}_{d} g(h(x))' = \underbrace{f'(g(h(x)))g'(h(x))}_{d} h(x)' = \underbrace{f'(g(h(x))g'(h(x))h'(x)}_{d}$$

- U terminima neuronskih mreža, ovu računicu možemo iskazati ovako:
- Imamo do sada izračunati parcijalni izvod, d
- Proširujemo taj proizvod izvodom aktivacione funkcije za tekući nivo
- Idemo po nivoima, poslednji nivo, pa prethodni, pa prethodni, itd. i na svakom nivou pomnožimo d izvodom aktivacione funkcije

Propagacija unazad

- Greška za mrežu sa k slojeva na jednoj instanci (h_k je funkcija parametara w):

$$E(w) = L(y, h_k) + \lambda\Omega(w)$$

$d = \nabla_{h_k} L(y, h_k)$
repeat
  $d = d \odot g'(a_k)$
  $\nabla_{w_{k0}} E(w) = d + \lambda \nabla_{w_{k0}} \Omega(w)$
  $\nabla_{W_k} E(w) = d {h_{k-1}}^T + \lambda \nabla_{W_k} \Omega(w)$
  $d = {W_k}^T d$
  $k = k - 1$
until $k = 0$;

Propagacija unazad

- Greška za mrežu sa k slojeva na jednoj instanci (h_k je funkcija parametara w):

$$E(w) = L(y, h_k) + \lambda\Omega(w)$$

- Gradijent na celom skupu podataka se dobija sabiranjem gradijenata na pojedinačnim instancama

$d = \nabla_{h_k} L(y, h_k)$
repeat
  $d = d \odot g'(a_k)$
  $\nabla_{w_{k0}} E(w) = d + \lambda \nabla_{w_{k0}} \Omega(w)$
  $\nabla_{W_k} E(w) = d {h_{k-1}}^T + \lambda \nabla_{W_k} \Omega(w)$
  $d = {W_k}^T d$
  $k = k - 1$
until $k = 0$;

Propagacija unazad

- Greška za mrežu sa k slojeva na jednoj instanci (h_k je funkcija parametara w):

$$E(w) = L(y, h_k) + \lambda\Omega(w)$$

- Gradijent na celom skupu podataka se dobija sabiranjem gradijenata na pojedinačnim instancama
- Pretpostavlja se da je urađeno izračunavanje unapred, tako da su h_i poznati

$d = \nabla_{h_k} L(y, h_k)$
repeat
  $d = d \odot g'(a_k)$
  $\nabla_{w_{k0}} E(w) = d + \lambda \nabla_{w_{k0}} \Omega(w)$
  $\nabla_{W_k} E(w) = d {h_{k-1}}^T + \lambda \nabla_{W_k} \Omega(w)$
  $d = {W_k}^T d$
  $k = k - 1$
until $k = 0$;

Propagacija unazad - primer

- Uzmimo za primer vrlo jednostavnu neuronsku mrežu sa dva sloja:

$$f_w(x) = h_2 = \sigma(w_{20} + w_{21}\sigma(w_{10} + w_{11}x))$$

Propagacija unazad - primer

- Uzmimo za primer vrlo jednostavnu neuronsku mrežu sa dva sloja:

$$f_w(x) = h_2 = \sigma(w_{20} + w_{21}\sigma(w_{10} + w_{11}x))$$

  - Imamo ulazni podatak x (jednodimenzioni vektor)

Propagacija unazad - primer

- Uzmimo za primer vrlo jednostavnu neuronsku mrežu sa dva sloja:

$$f_w(x) = h_2 = \sigma(w_{20} + w_{21}\sigma(w_{10} + w_{11}x))$$

  - Imamo ulazni podatak x (jednodimenzioni vektor)
  - Na njega primenimo linearnu regresiju pomoću vektora parametara iz prvog sloja w_{10} i w_{11}

Propagacija unazad - primer

- Uzmimo za primer vrlo jednostavnu neuronsku mrežu sa dva sloja:

$$f_w(x) = h_2 = \sigma(w_{20} + w_{21}\sigma(w_{10} + w_{11}x))$$

  - Imamo ulazni podatak x (jednodimenzioni vektor)
  - Na njega primenimo linearnu regresiju pomoću vektora parametara iz prvog sloja w_{10} i w_{11}
  - Na dobijenu vrednost primenimo aktivacionu funkciju *sigma* u jedinom skrivenom sloju (jedini neuron) ove mreže

Propagacija unazad - primer

- Uzmimo za primer vrlo jednostavnu neuronsku mrežu sa dva sloja:

$$f_w(x) = h_2 = \sigma(w_{20} + w_{21}\sigma(w_{10} + w_{11}x))$$

  - Imamo ulazni podatak x (jednodimenzioni vektor)
  - Na njega primenimo linearnu regresiju pomoću vektora parametara iz prvog sloja w_{10} i w_{11}
  - Na dobijenu vrednost primenimo aktivacionu funkciju *sigma* u jedinom skrivenom sloju (jedini neuron) ove mreže
  - Izlazna vrednost iz tog neurona ide u izlazni sloj gde na nju primenjujemo linearnu regresiju pomoću vektora parametara iz drugog sloja w_{20} i w_{21}

Propagacija unazad - primer

- Uzmimo za primer vrlo jednostavnu neuronsku mrežu sa dva sloja:

$$f_w(x) = h_2 = \sigma(w_{20} + w_{21}\sigma(w_{10} + w_{11}x))$$

  - Imamo ulazni podatak x (jednodimenzioni vektor)
  - Na njega primenimo linearnu regresiju pomoću vektora parametara iz prvog sloja w_{10} i w_{11}
  - Na dobijenu vrednost primenimo aktivacionu funkciju *sigma* u jedinom skrivenom sloju (jedini neuron) ove mreže
  - Izlazna vrednost iz tog neurona ide u izlazni sloj gde na nju primenjujemo linearnu regresiju pomoću vektora parametara iz drugog sloja w_{20} i w_{21}
- Za ovakav regresioni model računamo kvadratnu grešku:

$$E(w) = (h_2 - y)^2$$

Propagacija unazad - primer

- Za ovakav regresioni model računamo kvadratnu grešku:

$$E(w) = (h_2 - y)^2$$

Propagacija unazad - primer

- Za ovakav regresioni model računamo kvadratnu grešku:

$$E(w) = (h_2 - y)^2$$

- U narednim razmatranjima, pretpostavićemo da je $y = 1$ i računaćemo grešku $E(w) = (h_2 - 1)^2$ na jednoj instanci kod koje je $y = 1$

Propagacija unazad - primer

- Za ovakav regresioni model računamo kvadratnu grešku:

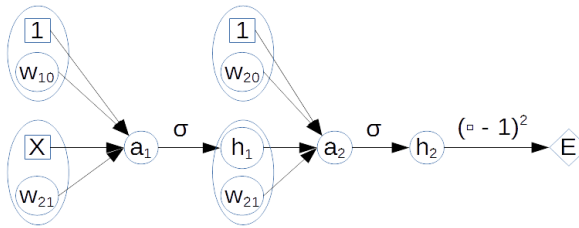
$$E(w) = (h_2 - y)^2$$

- U narednim razmatranjima, pretpostavićemo da je $y = 1$ i računaćemo grešku $E(w) = (h_2 - 1)^2$ na jednoj instanci kod koje je $y = 1$
- Gradijent po svim instancama se dobija kao zbir gradijenata po pojedinačnim instancama

Propagacija unazad - primer

$E(w, \mathcal{D}) = (h_2 - 1)^2$

$f_w(x) = h_2 = \sigma(w_{20} + w_{21}\sigma(w_{10} + w_{11}x))$



$d = \nabla_{h_k} L(y, h_k)$

repeat

$d = d \odot g'(a_k)$

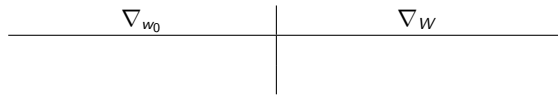
$\nabla_{w_{k0}} E(w) = d + \lambda \nabla_{w_{k0}} \Omega(w)$

$\nabla_{W_k} E(w) = d {h_{k-1}}^T + \lambda \nabla_{W_k} \Omega(w)$

$d = {W_k}^T d$

$k = k - 1$

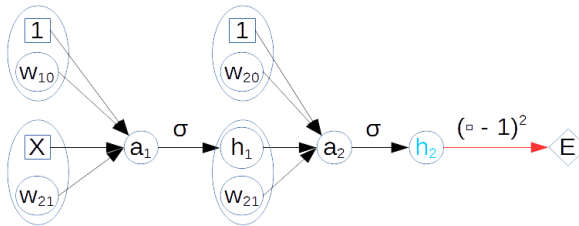
until $k = 0$;



Propagacija unazad - primer

$E(w, \mathcal{D}) = (h_2 - 1)^2$

$f_w(x) = h_2 = \sigma(w_{20} + w_{21}\sigma(w_{10} + w_{11}x))$



$d = \nabla_{h_k} L(y, h_k)$

repeat

$d = d \odot g'(a_k)$

$\nabla_{w_{k0}} E(w) = d + \lambda \nabla_{w_{k0}} \Omega(w)$

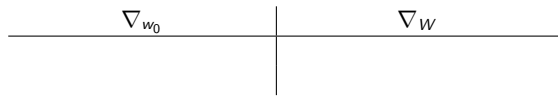
$\nabla_{W_k} E(w) = d {h_{k-1}}^T + \lambda \nabla_{W_k} \Omega(w)$

$d = {W_k}^T d$

$k = k - 1$

until $k = 0$;

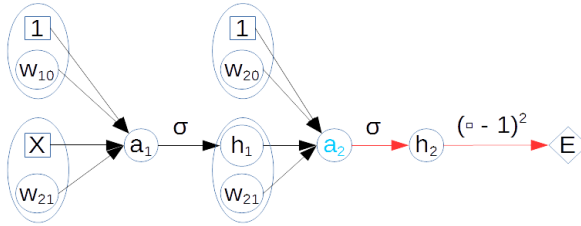
$d = 2(h_2 - 1)$



Propagacija unazad - primer

$$E(w, \mathcal{D}) = (h_2 - 1)^2$$

$$f_w(x) = h_2 = \sigma(\underbrace{w_{20} + w_{21}\sigma(w_{10} + w_{11}x)}_{a_2})$$



$d = \nabla_{h_k} L(y, h_k)$

repeat

- $d = d \odot g'(a_k)$
- $\nabla_{w_{k0}} E(w) = d + \lambda \nabla_{w_{k0}} \Omega(w)$
- $\nabla_{W_k} E(w) = d {h_{k-1}}^T + \lambda \nabla_{W_k} \Omega(w)$
- $d = {W_k}^T d$
- $k = k - 1$

until $k = 0$;

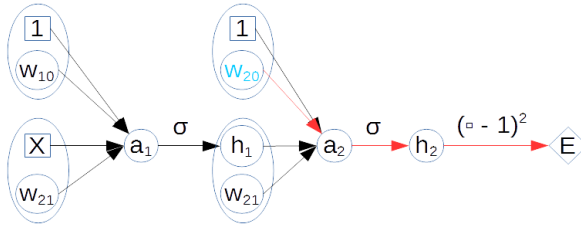
$$d = 2(h_2 - 1)\sigma'(a_2)$$

∇_{w_0}	∇_W

Propagacija unazad - primer

$$E(w, \mathcal{D}) = (h_2 - 1)^2$$

$$f_w(x) = h_2 = \sigma(\underbrace{w_{20} + w_{21}\sigma(w_{10} + w_{11}x)}_{a_2})$$



$d = \nabla_{h_k} L(y, h_k)$

repeat

$d = d \odot g'(a_k)$

$\nabla_{w_{k0}} E(w) = d + \lambda \nabla_{w_{k0}} \Omega(w)$

$\nabla_{W_k} E(w) = d {h_{k-1}}^T + \lambda \nabla_{W_k} \Omega(w)$

$d = {W_k}^T d$

$k = k - 1$

until $k = 0$;

$$d = 2(h_2 - 1)\sigma'(a_2)$$

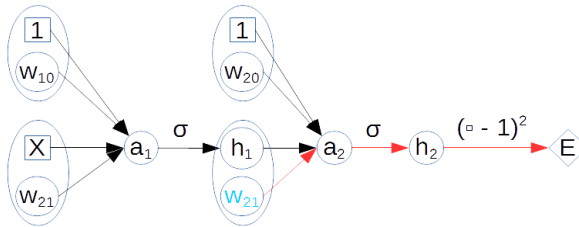
∇_{w_0}	∇_W
$2(h_2 - 1)\sigma'(a_2)$	

Propagacija unazad - primer

$$E(w, \mathcal{D}) = (h_2 - 1)^2$$

$$f_w(x) = h_2 = \sigma(w_{20} + w_{21}\sigma(\underbrace{w_{10} + w_{11}x}_{a_1}))$$

a_2



$d = \nabla_{h_k} L(y, h_k)$

repeat

- $d = d \odot g'(a_k)$
- $\nabla_{w_{k0}} E(w) = d + \lambda \nabla_{w_{k0}} \Omega(w)$
- $\nabla_{W_k} E(w) = d {h_{k-1}}^T + \lambda \nabla_{W_k} \Omega(w)$
- $d = {W_k}^T d$
- $k = k - 1$

until $k = 0$;

$$d = 2(h_2 - 1)\sigma'(a_2)$$

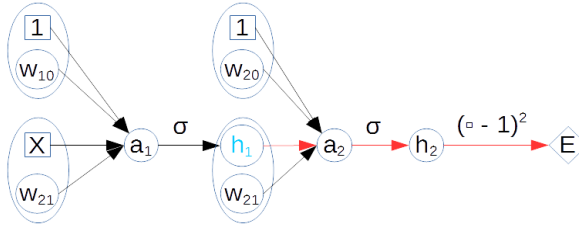
∇_{w_0}	∇_W
$2(h_2 - 1)\sigma'(a_2)$	$2(h_2 - 1)\sigma'(a_2)\sigma(a_1)$

Propagacija unazad - primer

$$E(w, \mathcal{D}) = (h_2 - 1)^2$$

$$f_w(x) = h_2 = \sigma(w_{20} + w_{21}\sigma(\underbrace{w_{10} + w_{11}x}_{a_1}))$$

a_2



$d = \nabla_{h_k} L(y, h_k)$

repeat

- $d = d \odot g'(a_k)$
- $\nabla_{w_{k0}} E(w) = d + \lambda \nabla_{w_{k0}} \Omega(w)$
- $\nabla_{W_k} E(w) = d {h_{k-1}}^T + \lambda \nabla_{W_k} \Omega(w)$
- $d = {W_k}^T d$
- $k = k - 1$

until $k = 0$;

$$d = 2w_{21}(h_2 - 1)\sigma'(a_2)$$

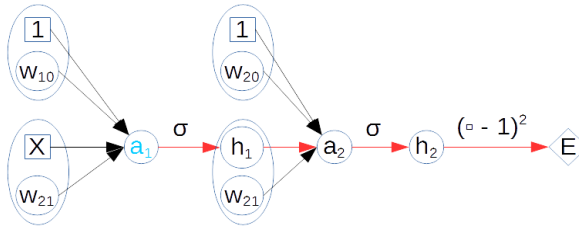
∇_{w_0}	∇_W
$2(h_2 - 1)\sigma'(a_2)$	$2(h_2 - 1)\sigma'(a_2)\sigma(a_1)$

Propagacija unazad - primer

$$E(w, \mathcal{D}) = (h_2 - 1)^2$$

$$f_w(x) = h_2 = \sigma(w_{20} + w_{21}\sigma(\underbrace{w_{10} + w_{11}x}_{a_1}))$$

a_2



$d = \nabla_{h_k} L(y, h_k)$

repeat

- $d = d \odot g'(a_k)$
- $\nabla_{w_{k0}} E(w) = d + \lambda \nabla_{w_{k0}} \Omega(w)$
- $\nabla_{W_k} E(w) = d {h_{k-1}}^T + \lambda \nabla_{W_k} \Omega(w)$
- $d = {W_k}^T d$
- $k = k - 1$

until $k = 0$;

$$d = 2w_{21}(h_2 - 1)\sigma'(a_2)\sigma'(a_1)$$

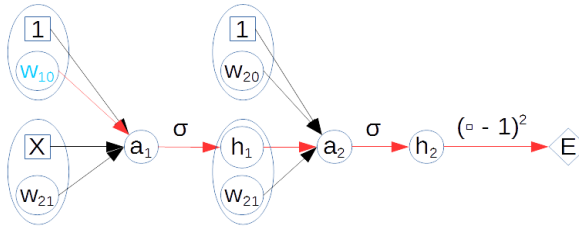
∇_{w_0}	∇_W
$2(h_2 - 1)\sigma'(a_2)$	$2(h_2 - 1)\sigma'(a_2)\sigma(a_1)$

Propagacija unazad - primer

$$E(w, \mathcal{D}) = (h_2 - 1)^2$$

$$f_w(x) = h_2 = \sigma(w_{20} + w_{21}\sigma(\underbrace{w_{10} + w_{11}x}_{a_1}))$$

a_2



$d = \nabla_{h_k} L(y, h_k)$

repeat

- $d = d \odot g'(a_k)$
- $\nabla_{w_{k0}} E(w) = d + \lambda \nabla_{w_{k0}} \Omega(w)$
- $\nabla_{W_k} E(w) = d {h_{k-1}}^T + \lambda \nabla_{W_k} \Omega(w)$
- $d = {W_k}^T d$
- $k = k - 1$

until $k = 0$;

$$d = 2w_{21}(h_2 - 1)\sigma'(a_2)\sigma'(a_1)$$

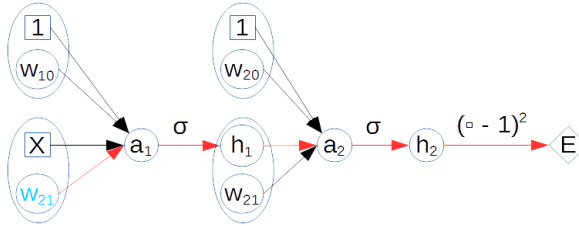
∇_{w_0}	∇_W
$2w_{21}(h_2 - 1)\sigma'(a_2)\sigma'(a_1)$	
$2(h_2 - 1)\sigma'(a_2)$	$2(h_2 - 1)\sigma'(a_2)\sigma(a_1)$

Propagacija unazad - primer

$$E(w, \mathcal{D}) = (h_2 - 1)^2$$

$$f_w(x) = h_2 = \sigma(w_{20} + w_{21}\sigma(\underbrace{w_{10} + w_{11}x}_{a_1}))$$

a_2



$d = \nabla_{h_k} L(y, h_k)$

repeat

- $d = d \odot g'(a_k)$
- $\nabla_{w_{k0}} E(w) = d + \lambda \nabla_{w_{k0}} \Omega(w)$
- $\nabla_{W_k} E(w) = d {h_{k-1}}^T + \lambda \nabla_{W_k} \Omega(w)$
- $d = {W_k}^T d$
- $k = k - 1$

until $k = 0$;

$$d = 2w_{21}(h_2 - 1)\sigma'(a_2)\sigma'(a_1)$$

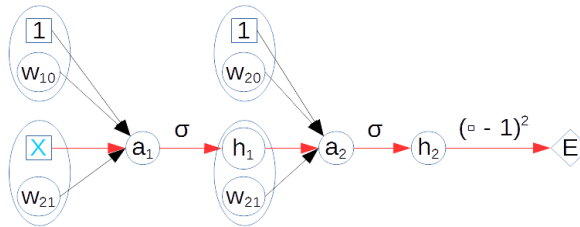
∇_{w_0}	∇_W
$2w_{21}(h_2 - 1)\sigma'(a_2)\sigma'(a_1)$	$2w_{21}(h_2 - 1)\sigma'(a_2)\sigma'(a_1)x$
$2(h_2 - 1)\sigma'(a_2)$	$2(h_2 - 1)\sigma'(a_2)\sigma(a_1)$

Propagacija unazad - primer

$$E(w, \mathcal{D}) = (h_2 - 1)^2$$

$$f_w(x) = h_2 = \sigma(w_{20} + w_{21}\sigma(\underbrace{w_{10} + w_{11}x}_{a_1}))$$

a_2



$d = \nabla_{h_k} L(y, h_k)$

repeat

- $d = d \odot g'(a_k)$
- $\nabla_{w_{k0}} E(w) = d + \lambda \nabla_{w_{k0}} \Omega(w)$
- $\nabla_{W_k} E(w) = d {h_{k-1}}^T + \lambda \nabla_{W_k} \Omega(w)$
- $d = {W_k}^T d$
- $k = k - 1$

until $k = 0$;

$$d = 2w_{11}w_{21}(h_2 - 1)\sigma'(a_2)\sigma'(a_1)$$

∇_{w_0}	∇_W
$2w_{21}(h_2 - 1)\sigma'(a_2)\sigma'(a_1)$	$2w_{21}(h_2 - 1)\sigma'(a_2)\sigma'(a_1)x$
$2(h_2 - 1)\sigma'(a_2)$	$2(h_2 - 1)\sigma'(a_2)\sigma(a_1)$

Propagacija unazad

- Algoritam propagacije unazad nije lako primenljiv na duboke neuronske mreže, zbog velikog broja uzastopnih množenja.

Propagacija unazad

- Algoritam propagacije unazad nije lako primenljiv na duboke neuronske mreže, zbog velikog broja uzastopnih množenja.
- Konkretno, zbog toga često dolazi do toga da vrednosti parcijalnih izvoda koje se računaju budu praktično nula (kada se množe brojevi manji od jedan) ili da budu ogromne ili čak da dođe do prekoračenja ili prosto nestabilnosti optimizacionog metoda (kada se množe brojevi veći od jedan)

Propagacija unazad

- Algoritam propagacije unazad nije lako primenljiv na duboke neuronske mreže, zbog velikog broja uzastopnih množenja.
- Konkretno, zbog toga često dolazi do toga da vrednosti parcijalnih izvoda koje se računaju budu praktično nula (kada se množe brojevi manji od jedan) ili da budu ogromne ili čak da dođe do prekoračenja ili prosto nestabilnosti optimizacionog metoda (kada se množe brojevi veći od jedan)
- Ovaj problem se naziva problemom *nestajućih* i *eksplodirajućih* gradijenata

Propagacija unazad

- Algoritam propagacije unazad nije lako primenljiv na duboke neuronske mreže, zbog velikog broja uzastopnih množenja.
- Konkretno, zbog toga često dolazi do toga da vrednosti parcijalnih izvoda koje se računaju budu praktično nula (kada se množe brojevi manji od jedan) ili da budu ogromne ili čak da dođe do prekoračenja ili prosto nestabilnosti optimizacionog metoda (kada se množe brojevi veći od jedan)
- Ovaj problem se naziva problemom *nestajućih* i *eksplodirajućih* gradijenata
- Ovaj problem je blaži ukoliko se kao aktivaciona funkcija koristi nakošena ispravljena linearna jedinica, pošto za pozitivne vrednosti ima vrednost izvoda 1.

Mane neuronskih mreža

- Nije ih lako trenirati

Mane neuronskih mreža

- Nije ih lako trenirati
- Visoka računska zahtevnost

Mane neuronskih mreža

- Nije ih lako trenirati
- Visoka računska zahtevnost
- Visoka konfigurabilnost, ali bez jasnih smernica kako izabrati konfiguraciju

Mane neuronskih mreža

- Nije ih lako trenirati
- Visoka računska zahtevnost
- Visoka konfigurabilnost, ali bez jasnih smernica kako izabrati konfiguraciju
- Visok potencijal za preprilagođavanje

Mane neuronskih mreža

- Nije ih lako trenirati
- Visoka računska zahtevnost
- Visoka konfigurabilnost, ali bez jasnih smernica kako izabrati konfiguraciju
- Visok potencijal za preprilagođavanje
- Traže velike količine podataka

Pregled

Potpuno povezane neuronske mreže

Konvolutivne neuronske mreže

Rekurentne neuronske mreže

Praktične tehnike i napredni koncepti

Konvolutivne neuronske mreže

- Ključni domen primene je obrada signala (npr. slike, zvuk ili slično)

Konvolutivne neuronske mreže

- Ključni domen primene je obrada signala (npr. slike, zvuk ili slično)
- Zasnovane na operaciji *konvolucije* koja se koristi u obradi signala

Konvolutivne neuronske mreže

- Ključni domen primene je obrada signala (npr. slike, zvuk ili slično)
- Zasnovane na operaciji *konvolucije* koja se koristi u obradi signala
- U tradicionalnoj obradi signala ljudi su konstruisali tzv. filtere

Konvolutivne neuronske mreže

- Ključni domen primene je obrada signala (npr. slike, zvuk ili slično)
- Zasnovane na operaciji *konvolucije* koja se koristi u obradi signala
- U tradicionalnoj obradi signala ljudi su konstruisali tzv. filtere
- Filteri predstavljaju matrice koju kada u operaciji konvolucije primenimo na neki signal možemo doći do informacija kao što su gde se nalaze ivice na slici

Konvolutivne neuronske mreže

- Ključni domen primene je obrada signala (npr. slike, zvuk ili slično)
- Zasnovane na operaciji *konvolucije* koja se koristi u obradi signala
- U tradicionalnoj obradi signala ljudi su konstruisali tzv. filtere
- Filteri predstavljaju matrice koju kada u operaciji konvolucije primenimo na neki signal možemo doći do informacija kao što su gde se nalaze ivice na slici
- Ključna promena je da konvolutivne mreže nauče filtere tako da rade ono što je približno najkorisnije u problemu koji se rešava

Konvolutivne neuronske mreže

- Ključni domen primene je obrada signala (npr. slike, zvuk ili slično)
- Zasnovane na operaciji *konvolucije* koja se koristi u obradi signala
- U tradicionalnoj obradi signala ljudi su konstruisali tzv. filtere
- Filteri predstavljaju matrice koju kada u operaciji konvolucije primenimo na neki signal možemo doći do informacija kao što su gde se nalaze ivice na slici
- Ključna promena je da konvolutivne mreže nauče filtere tako da rade ono što je približno najkorisnije u problemu koji se rešava
- Jedna od njihovih glavnih prednosti je da iz podataka automatski konstruišu atribute koji su relevantni u problemu koji se rešava

Konvolutivne neuronske mreže

- Izlazi jedinica na nižim slojevima se mogu smatrati atributima dobijenim na osnovu vrednosti prethodnih slojeva

Konvolutivne neuronske mreže

- Izlazi jedinica na nižim slojevima se mogu smatrati atributima dobijenim na osnovu vrednosti prethodnih slojeva
- Svaki sloj vrši *ekstrakciju atributa*, pri čemu je smisao atributa na višim nivoima kompleksniji od smisla onih na nižim

Konvolutivne neuronske mreže

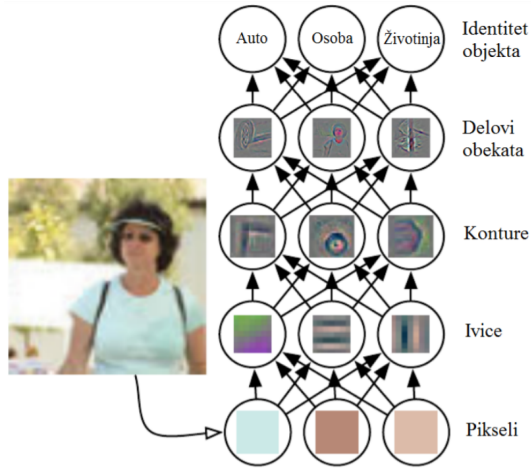
- Izlazi jedinica na nižim slojevima se mogu smatrati atributima dobijenim na osnovu vrednosti prethodnih slojeva
- Svaki sloj vrši *ekstrakciju atributa*, pri čemu je smisao atributa na višim nivoima kompleksniji od smisla onih na nižim
- Omogućava primenu neuronskih mreža na sirove podatke, bez prethodnog definisanja atributa od strane eksperata

Konvolutivne neuronske mreže

- Izlazi jedinica na nižim slojevima se mogu smatrati atributima dobijenim na osnovu vrednosti prethodnih slojeva
- Svaki sloj vrši *ekstrakciju atributa*, pri čemu je smisao atributa na višim nivoima kompleksniji od smisla onih na nižim
- Omogućava primenu neuronskih mreža na sirove podatke, bez prethodnog definisanja atributa od strane eksperata
- Vrlo korisno za obradu slike, videa i zvuka

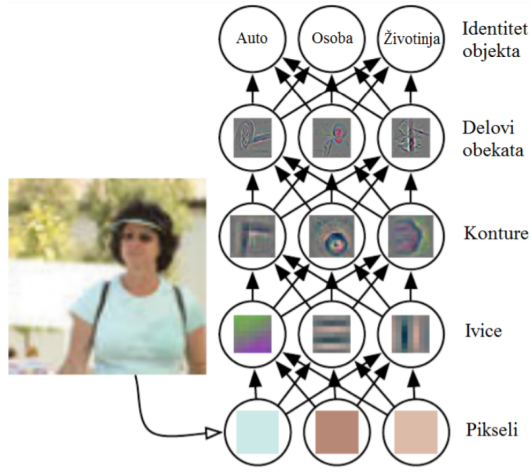
Konvolutivne neuronske mreže

- Na ulazu se mreži daju sirovi pikseli



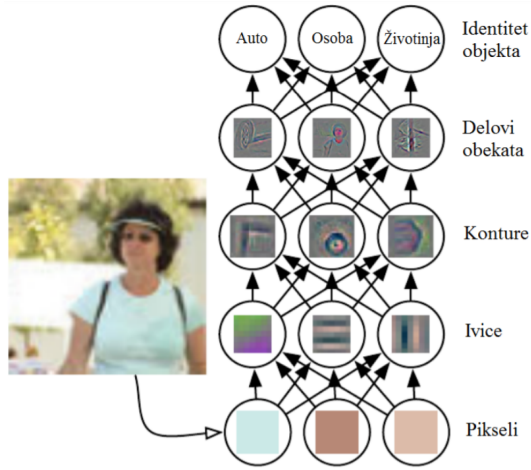
Konvolutivne neuronske mreže

- Na ulazu se mreži daju sirovi pikseli
- Kako idemo kroz slojeve mreže, prvi slojevi detektuju jednostavne atribute kao što su ivice (uspravne, horizontalne, itd.)



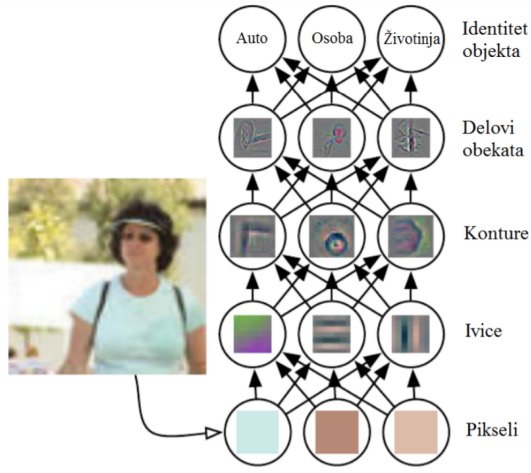
Konvolutivne neuronske mreže

- Na ulazu se mreži daju sirovi pikseli
- Kako idemo kroz slojeve mreže, prvi slojevi detektuju jednostavne atribute kao što su ivice (uspravne, horizontalne, itd.)
- Sledeći nivo može, polazeći od onoga što je prethodni nivo detektovao, u terminima ivica može da izrazi konture



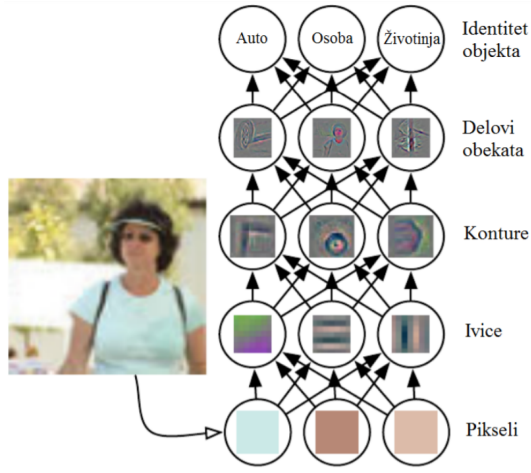
Konvolutivne neuronske mreže

- Na ulazu se mreži daju sirovi pikseli
- Kako idemo kroz slojeve mreže, prvi slojevi detektuju jednostavne atribute kao što su ivice (uspravne, horizontalne, itd.)
- Sledeći nivo može, polazeći od onoga što je prethodni nivo detektovao, u terminima ivica može da izrazi konture
- Sledeći nivo može da prepoznaje nekakve delove objekta na osnovu prethodno prepoznatih kontura

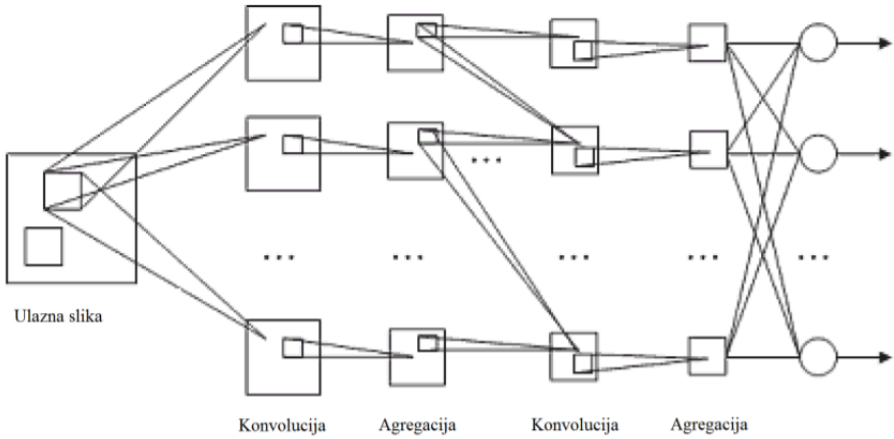


Konvolutivne neuronske mreže

- Na ulazu se mreži daju sirovi pikseli
- Kako idemo kroz slojeve mreže, prvi slojevi detektuju jednostavne atribute kao što su ivice (uspravne, horizontalne, itd.)
- Sledeći nivo može, polazeći od onoga što je prethodni nivo detektovao, u terminima ivica može da izrazi konture
- Sledeći nivo može da prepoznaje nekakve delove objekta na osnovu prethodno prepoznatih kontura
- Na poslednjem nivou, na osnovu delova objekata, na kraju može da se uradi klasifikacija pronađenih objekata

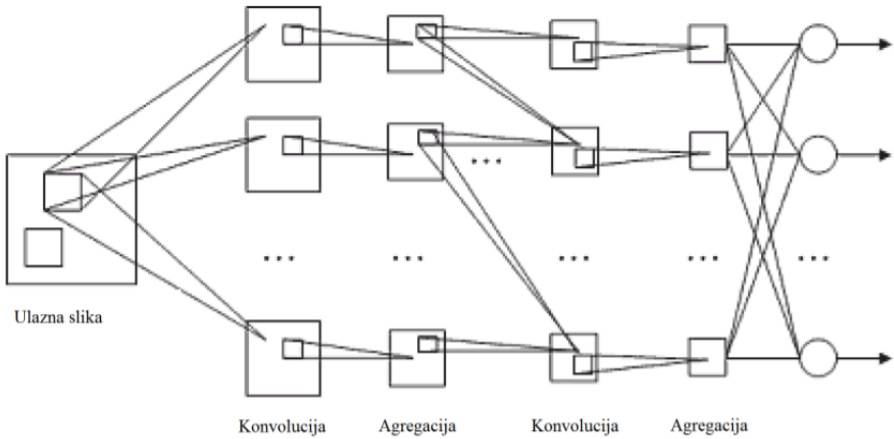


Shema konvolutivne mreže



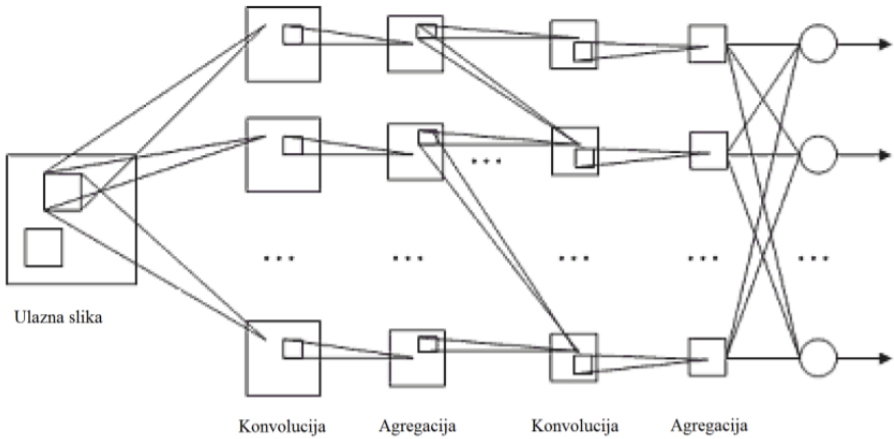
Konvolutivna mreža se sastoji od konvolutivnih slojeva (eng. convolution layer), slojeva agregacije (eng. pooling layer) i standardne potpuno povezane neuronske mreže

Shema konvolutivne mreže



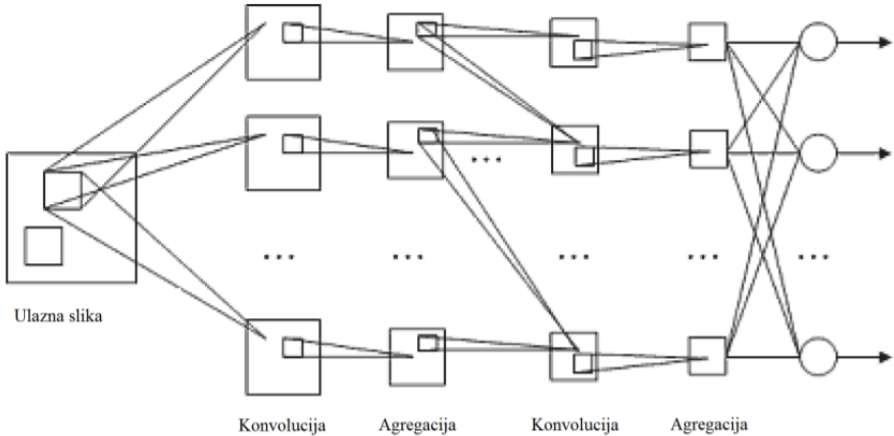
Konvolutivni slojevi i slojevi agregacije se smenjuju jedan za drugim, pri čemu su im dimenzije sve manje

Shema konvolutivne mreže



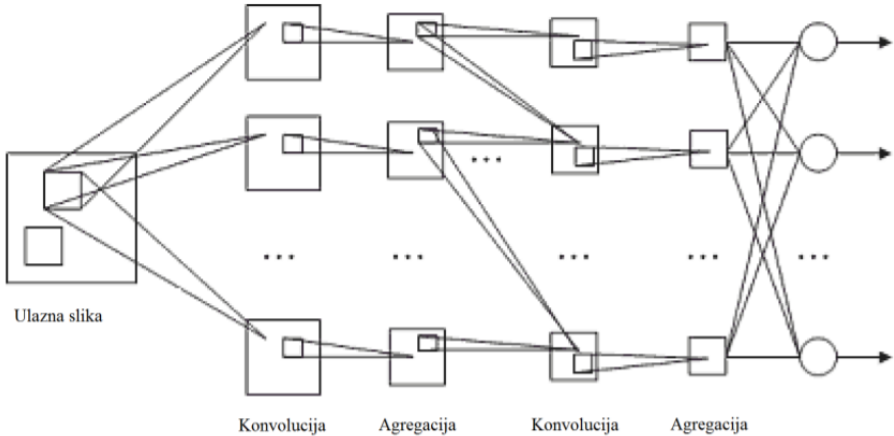
Standardna neuronska mreža je povezana na izlaze poslednjeg sloja agregacije i uči nad atributima koje prethodni slojevi konstruišu

Shema konvolutivne mreže



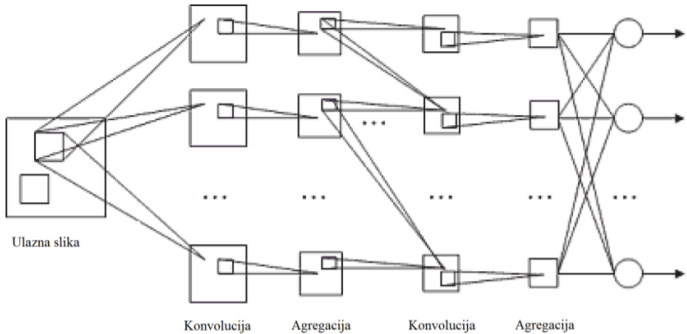
- Uloga slojeva konvolucije je da vrše operacije konvolucije i tako konstruišu atribute

Shema konvolutivne mreže



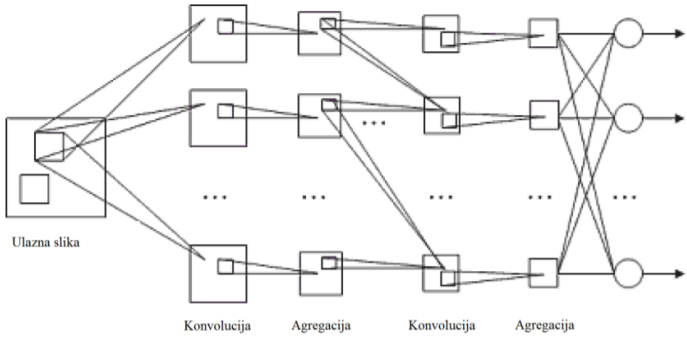
- Uloga slojeva konvolucije je da vrše operacije konvolucije i tako konstruišu atribute
- Ako je na ulazu slika, izlaz iz sloja konvolucije će biti neka nova slika

Shema konvolutivne mreže



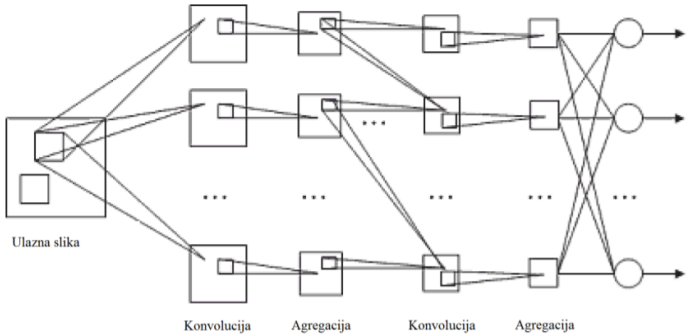
- Uloga slojeva agregacije je da agregiraju informaciju dobijenu iz konvolutivnog sloja (npr. da svedu sliku na manju rezoluciju uprosečavanjem, maksimumom ili slično)

Shema konvolutivne mreže



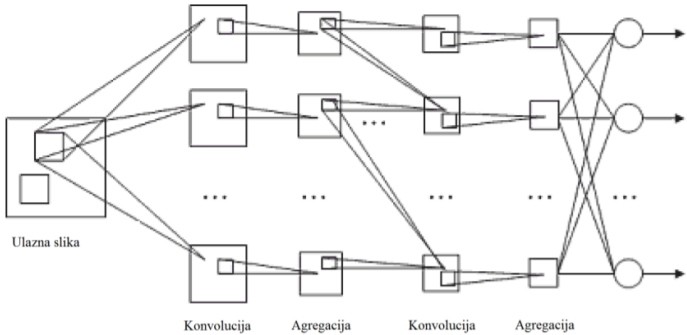
- Uloga slojeva agregacije je da agregiraju informaciju dobijenu iz konvolutivnog sloja (npr. da svedu sliku na manju rezoluciju uprosečavanjem, maksimumom ili slično)
- Sa slikama manje rezolucije računske operacije su jeftinije a broj parametara smanjen

Shema konvolutivne mreže



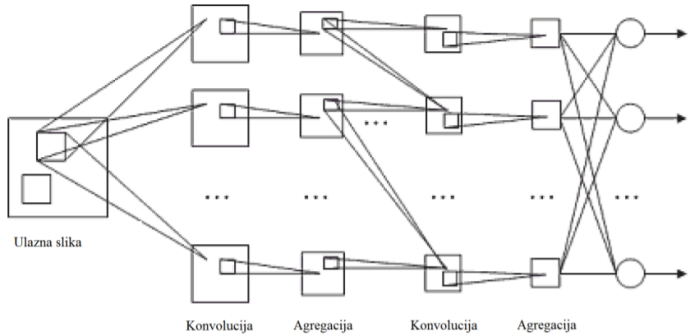
- Od ulazne slike u prvom sloju konvolucije dobijamo nekoliko novih slika

Shema konvolutivne mreže



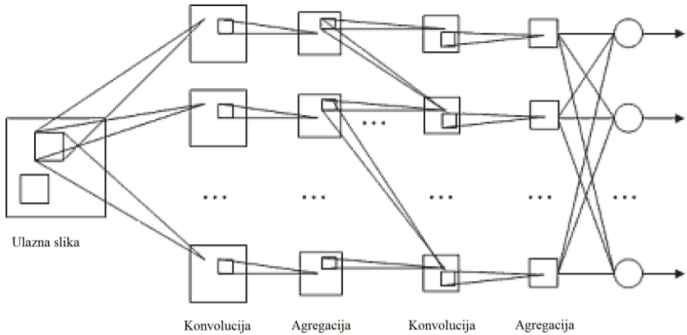
- Od ulazne slike u prvom sloju konvolucije dobijamo nekoliko novih slika
- Svaka od novih slika se dobija kao rezultat primene jednog *konvolutivnog filtera* na ulaznu sliku

Shema konvolutivne mreže



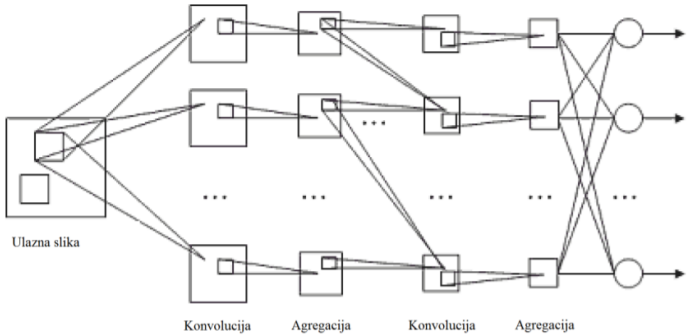
- Od ulazne slike u prvom sloju konvolucije dobijamo nekoliko novih slika
- Svaka od novih slika se dobija kao rezultat primene jednog *konvolutivnog filtera* na ulaznu sliku
- Konvolutivni filter je funkcija koja jednu sliku preslikava u drugu (npr. na rezultujućim slikama su iscrtane ivice (na jednoj horizontalne, na drugoj vertikalne, itd.) sa polazne slike)

Shema konvolutivne mreže



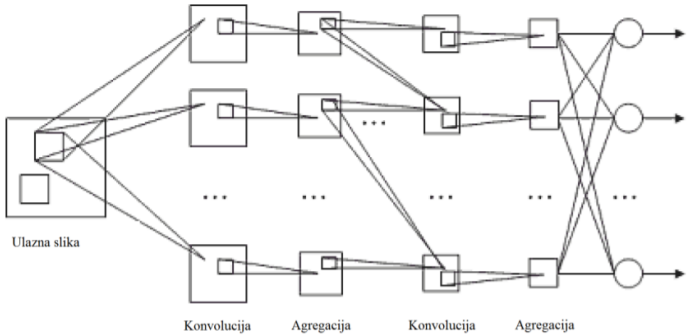
- Konvolutivni filteri dakle od polazne slike daju različite nove slike koje predstavljaju različite aspekte polazne slike odnosno njene *atribute*

Shema konvolutivne mreže



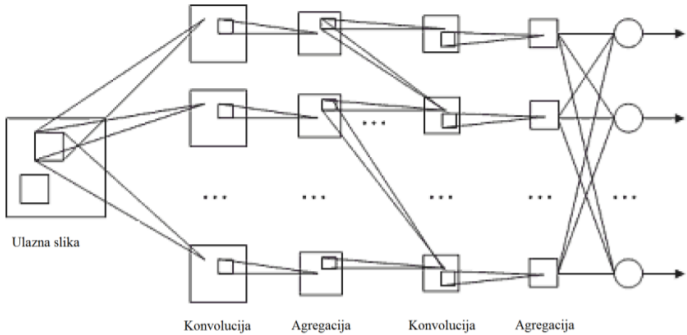
- Konvolutivni filteri dakle od polazne slike daju različite nove slike koje predstavljaju različite aspekte polazne slike odnosno njene *atribute*
- Na svaku od dobijenih slika se primeni agregacija tako da se slika smanji a informacija se ukrupni (agregira)

Shema konvolutivne mreže



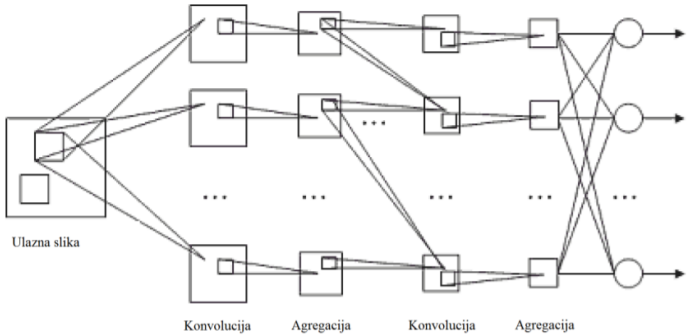
- Konvolutivni filteri dakle od polazne slike daju različite nove slike koje predstavljaju različite aspekte polazne slike odnosno njene *atribute*
- Na svaku od dobijenih slika se primeni agregacija tako da se slika smanji a informacija se ukrupni (agregira)
- Nakon toga, na nove, smanjene slike, možemo ponovo da primenimo konvolutivne filtere i ponovo da dobijemo nove slike

Shema konvolutivne mreže



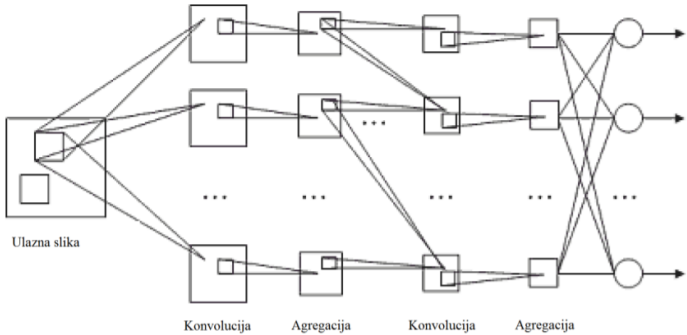
- Na kraju dolazimo do finalnih slika male dimenzije

Shema konvolutivne mreže



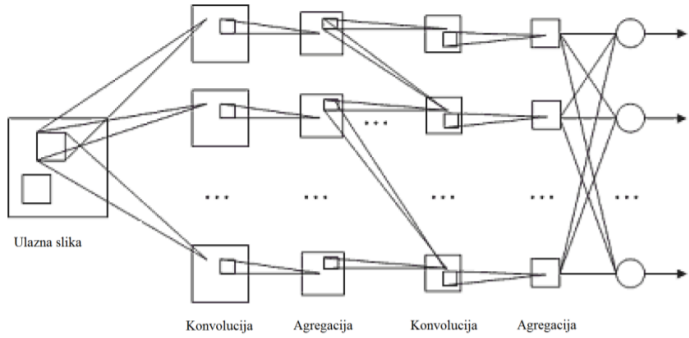
- Na kraju dolazimo do finalnih slika male dimenzije
- Svaka slika je jedna matrica

Shema konvolutivne mreže



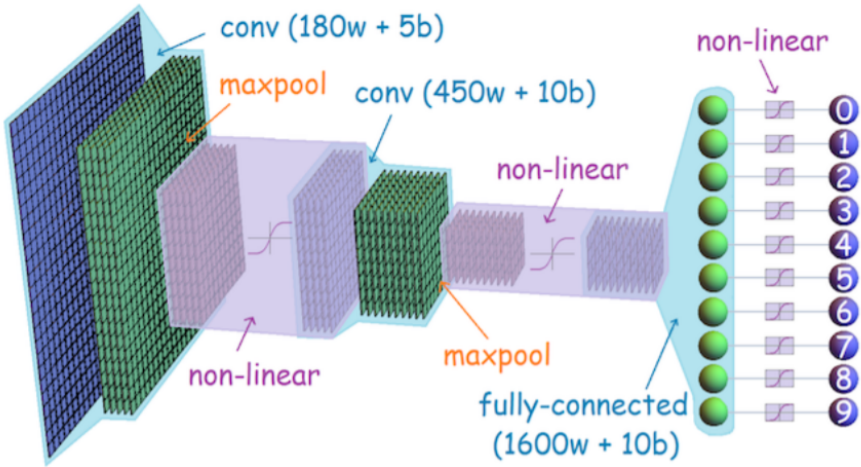
- Na kraju dolazimo do finalnih slika male dimenzije
- Svaka slika je jedna matrica
- Tada se tipično sve vrednosti koje se nalaze u ovim matricama poređaju u jedan vektor i taj vektor se koristi kao ulaz u potpuno povezanu mrežu

Shema konvolutivne mreže



- Na kraju dolazimo do finalnih slika male dimenzije
- Svaka slika je jedna matrica
- Tada se tipično sve vrednosti koje se nalaze u ovim matricama poređaju u jedan vektor i taj vektor se koristi kao ulaz u potpuno povezanu mrežu
- Deo mreže do potpuno povezane mreže možemo posmatrati kao automatski generator novih atributa za potpuno povezanu mrežu

Shema konvolutivne mreže

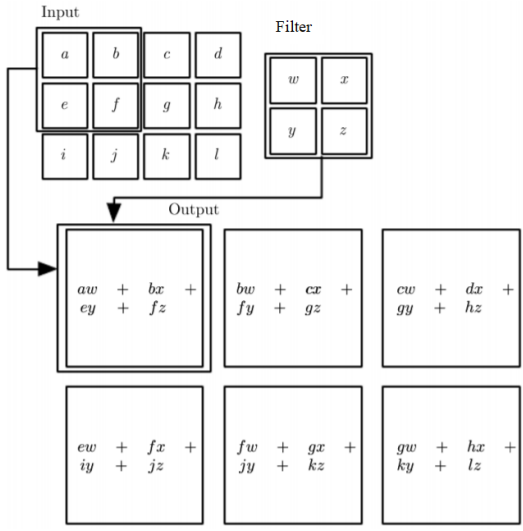


- Izlaz iz konvolutivnog sloja se transformiše nelinearnom funkcijom

Konvolucija

- Neka su f i g dve matrice dimenzija $m \times n$ i $p \times q$. Konvolucija je operacija definisana u diskretnom dvodimenzionalnom slučaju na sledeći način

$$(f * g)_{ij} = \sum_{k=0}^{p-1} \sum_{l=0}^{q-1} f_{i-k,i-l} g_{k,l}$$

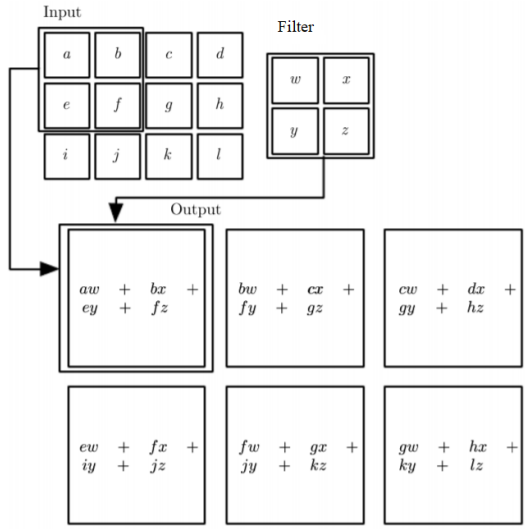


Konvolucija

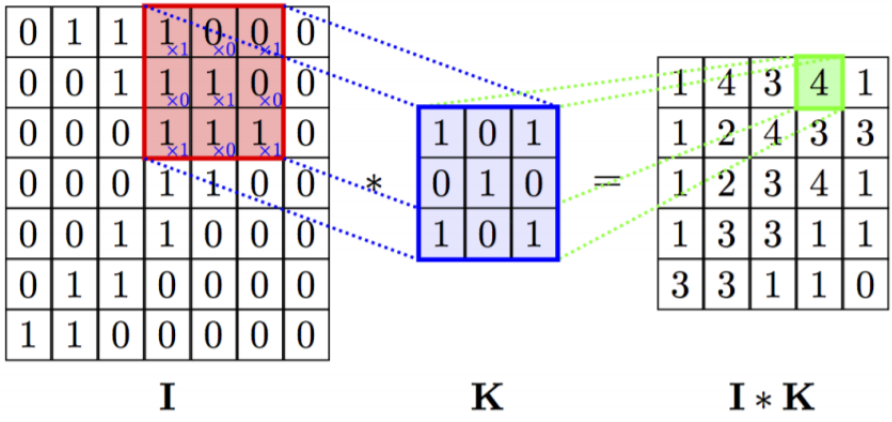
- Neka su f i g dve matrice dimenzija $m \times n$ i $p \times q$. Konvolucija je operacija definisana u diskretnom dvodimenzionalnom slučaju na sledeći način

$$(f * g)_{ij} = \sum_{k=0}^{p-1} \sum_{l=0}^{q-1} f_{i-k,i-l} g_{k,l}$$

- Matrica f je obično ulaz, poput slike, dok je matrica g *filter* – pomoćna matrica koja definiše transformaciju koja se vrši nad ulazom i izdvaja neku vrstu informacije iz ulaza

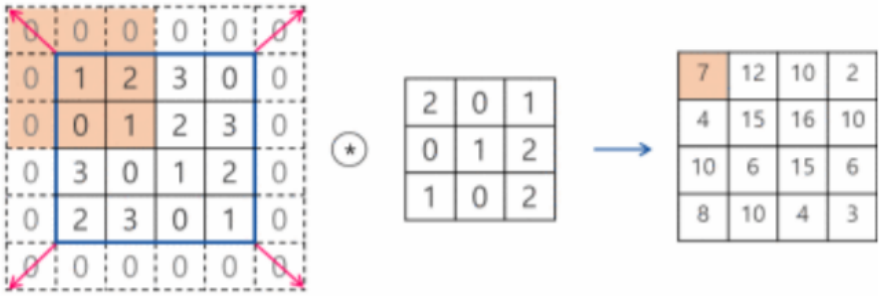


Konvolucija



- Primetimo da filter smanjuje sliku što u nekim primenama može da pogoduje a u nekim ne

Konvolucija



- Možemo proširiti sliku nulama ili vrednostima na obodu (eng. *padding*) i tako zadržati polaznu dimenziju slike nakon konvolucije

Konvolucija

- Ukoliko nam ne odgovara pomeranje filtera za korak 1 (npr. zato što je primena filtera računski skupa), možemo pomerati filter i za veći korak (eng. *stride*) i tako dodatno smanjiti matricu

Konvolucija

$$* \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} =$$

- Primer detekcije uspravnih ivica

Konvolucija

$$* \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} =$$

- Primer detekcije uspravnih ivica
- Primenu konvolucije možemo posmatrati kao traženje nečega što vrlo liči na filter

Konvolucija

$$* \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} =$$

- Primer detekcije uspravnih ivica
- Primenu konvolucije možemo posmatrati kao traženje nečega što vrlo liči na filter
- Što je sličnost dela slike i filtera veća, to će i vrednost konvolucije biti veća

Konvolucija

$$* \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} =$$

- Primer detekcije uspravnih ivica

Konvolucija

$$* \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} =$$

- Primer detekcije uspravnih ivica
- Ovaj filter nije naučen već je dizajniran

Konvolucija

$$* \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} =$$

- Primer detekcije uspravnih ivica
- Ovaj filter nije naučen već je dizajniran
- Osnovna ideja konvolutivnih mreža je da ovakvi (a potencijalno i korisniji) filteri ne moraju biti dizajnirani, već *naučeni*

Konvolucija

- Tako, postojanje nosa na slici se može utvrditi na osnovu specifičnog rasporeda uspravnih, kosih i horizontalnih linija, koje detektuje prethodni konvolutivni sloj

Konvolucija

- Tako, postojanje nosa na slici se može utvrditi na osnovu specifičnog rasporeda uspravnih, kosih i horizontalnih linija, koje detektuje prethodni konvolutivni sloj
- Svaki konvolutivni sloj raspolaže nizom parametrizovanih filtera koji vrše ove poslove

Konvolucija

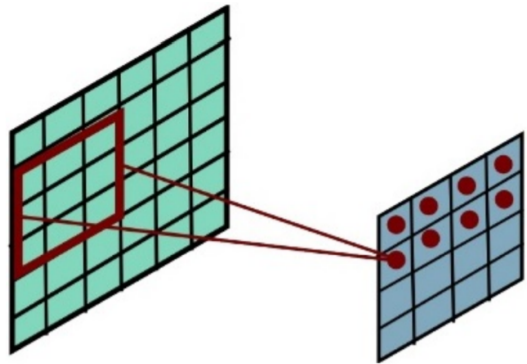
- Tako, postojanje nosa na slici se može utvrditi na osnovu specifičnog rasporeda uspravnih, kosih i horizontalnih linija, koje detektuje prethodni konvolutivni sloj
- Svaki konvolutivni sloj raspolaže nizom parametrizovanih filtera koji vrše ove poslove
- Na primer, ako jedan filter detektuje vertikalne linije, prirodno je imati paralelno, nad istim prethodnim slojem (ili ulazom) i filter koji detektuje horizontalne linije

Konvolucija

- Tako, postojanje nosa na slici se može utvrditi na osnovu specifičnog rasporeda uspravnih, kosih i horizontalnih linija, koje detektuje prethodni konvolutivni sloj
- Svaki konvolutivni sloj raspolaže nizom parametrizovanih filtera koji vrše ove poslove
- Na primer, ako jedan filter detektuje vertikalne linije, prirodno je imati paralelno, nad istim prethodnim slojem (ili ulazom) i filter koji detektuje horizontalne linije
- Upravo, parametri filtera predstavljaju parametre konvolutivne mreže i bivaju naučeni u procesu obučavanja mreže.

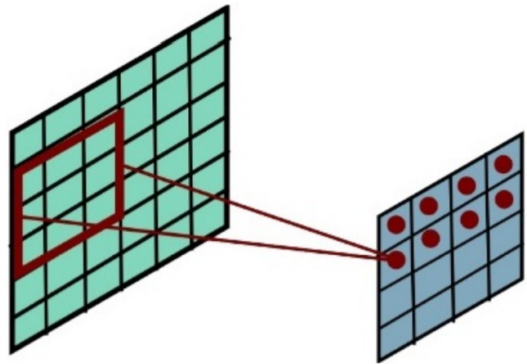
Konvolucija

- Konvolucija se realizuje jedinicama koje su organizovane u niz (u slučaju jednodimenzionih signala poput zvuka) ili matricu (u slučaju dovodimenzionih signala poput slike) i *dele vrednosti parametara*



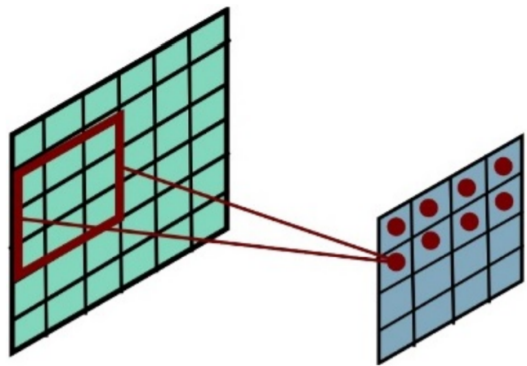
Konvolucija

- Konvolucija se realizuje jedinicama koje su organizovane u niz (u slučaju jednodimenzionih signala poput zvuka) ili matricu (u slučaju dovodimenzionih signala poput slike) i *dele vrednosti parametara*
- Obično kao ulaze uzimaju vrednosti susednih jedinica iz prethodnog sloja



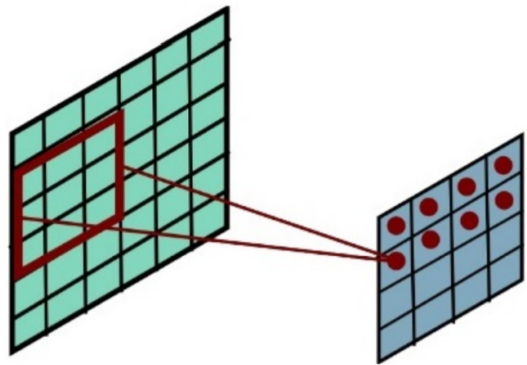
Konvolucija

- Konvolucija se realizuje jedinicama koje su organizovane u niz (u slučaju jednodimenzionih signala poput zvuka) ili matricu (u slučaju dovodimenzionih signala poput slike) i *dele vrednosti parametara*
- Obično kao ulaze uzimaju vrednosti susednih jedinica iz prethodnog sloja
- Na primer, u slučaju slike, to mogu biti vrednosti 3×3 jedinice iz prethodnog sloja

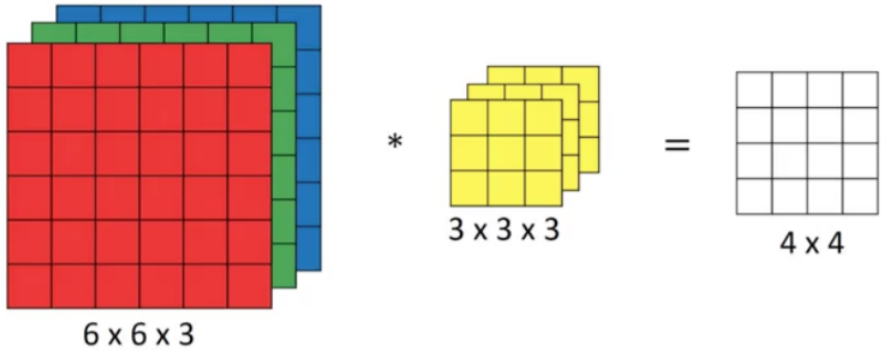


Konvolucija

- Konvolucija se realizuje jedinicama koje su organizovane u niz (u slučaju jednodimenzionih signala poput zvuka) ili matricu (u slučaju dovodimenzionih signala poput slike) i *dele vrednosti parametara*
- Obično kao ulaze uzimaju vrednosti susednih jedinica iz prethodnog sloja
- Na primer, u slučaju slike, to mogu biti vrednosti 3×3 jedinice iz prethodnog sloja
- Skup piksela koje jedna jedinica može da vidi se zove *receptivno polje*

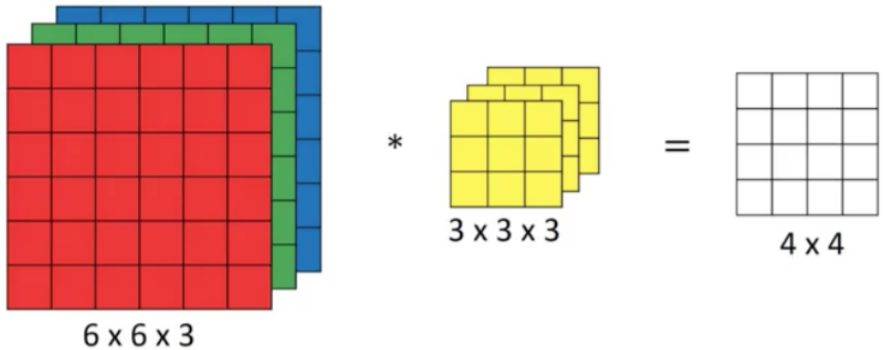


Konvolucija



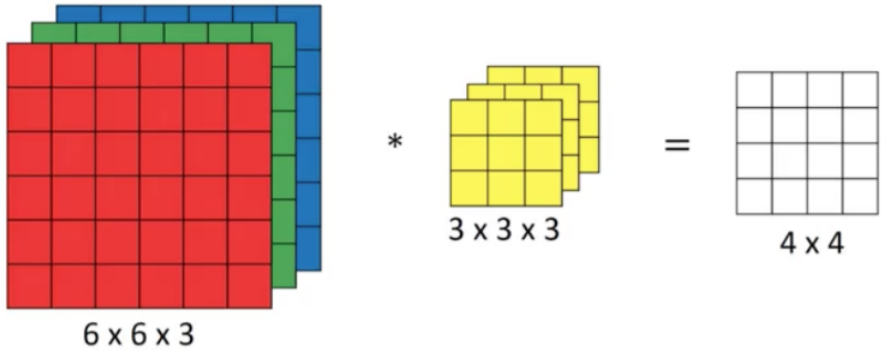
- Slike su često date u RGB formatu, pomoću tri matrice

Konvolucija



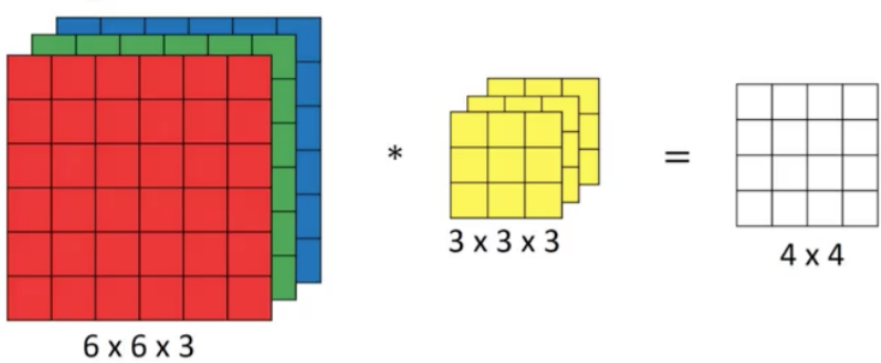
- Slike su često date u RGB formatu, pomoću tri matrice
- Kažemo da je ovakav signal *višekanalni*

Konvolucija



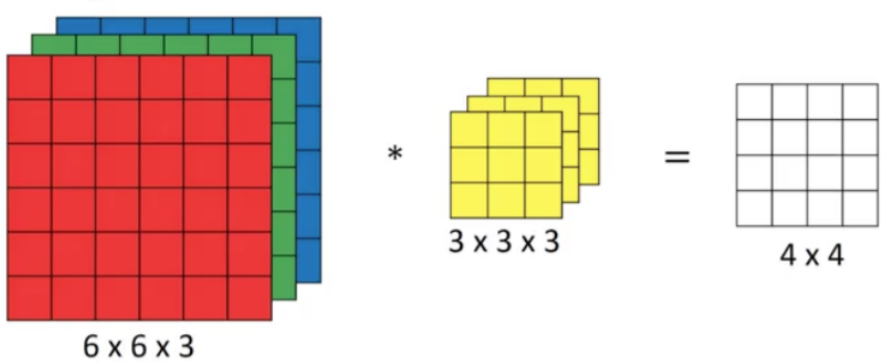
- Slike su često date u RGB formatu, pomoću tri matrice
- Kažemo da je ovakav signal *višekanalni*
- Na takav signal se primenjuju *višekanalne operacije konvolucije*

Konvolucija



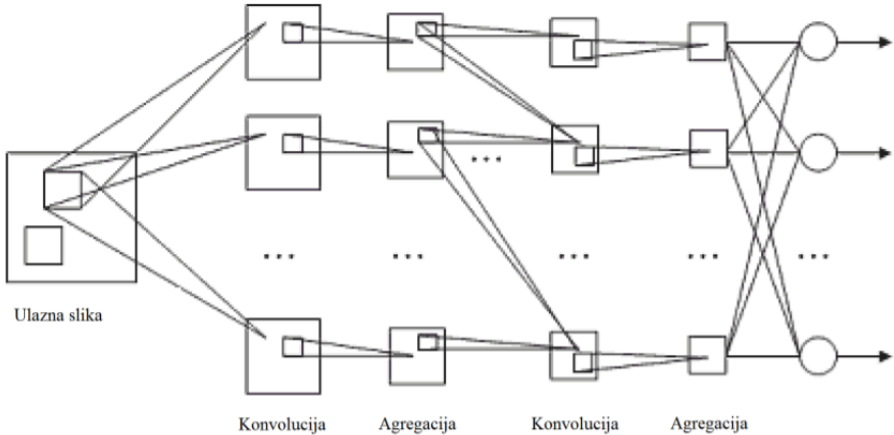
- Ovakav ulaz i matrica konvoluciji predstavljaju *tenzore* (višedimenzionalne nizove)

Konvolucija



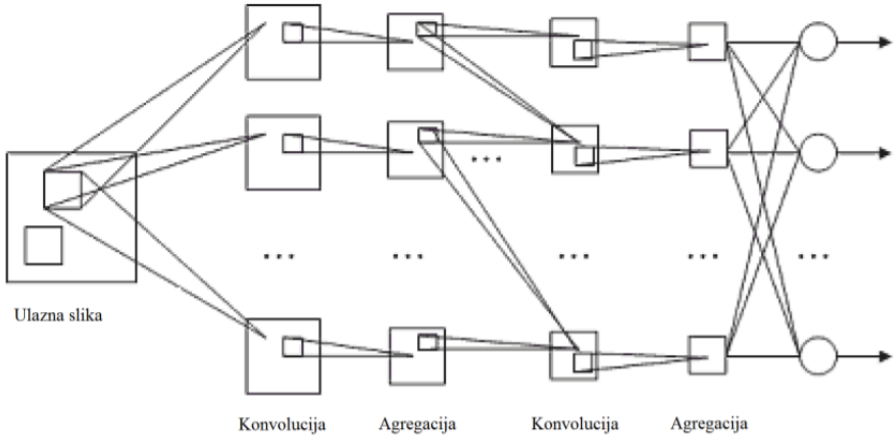
- Ovakav ulaz i matrica konvoluciji predstavljaju *tenzore* (višedimenzionalne nizove)
- Primetimo da izlaz nije tenzor već obična matrica

Konvolucija



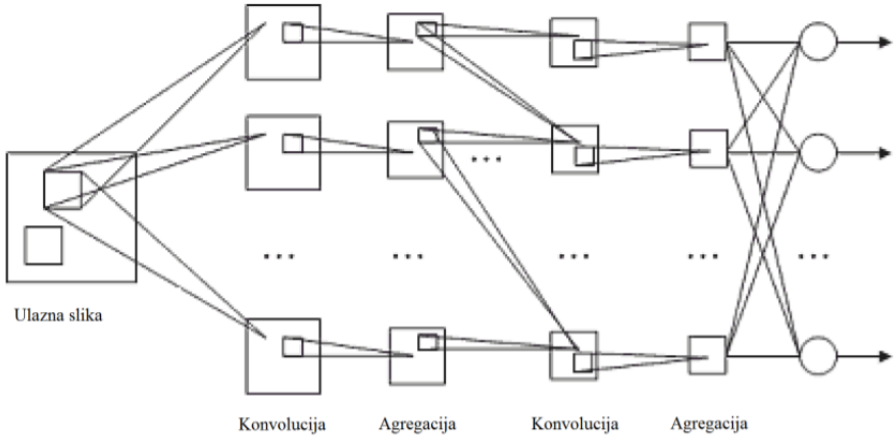
- Višekanalna konvolucija je važna i iz još jednog razloga

Konvolucija



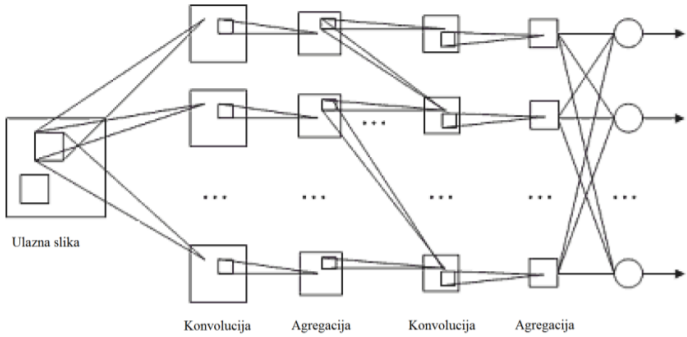
- Višekanalna konvolucija je važna i iz još jednog razloga
- Kada u prvom sloju imamo n filtera koji se primenjuju na polaznu sliku, dobijamo n slika

Konvolucija



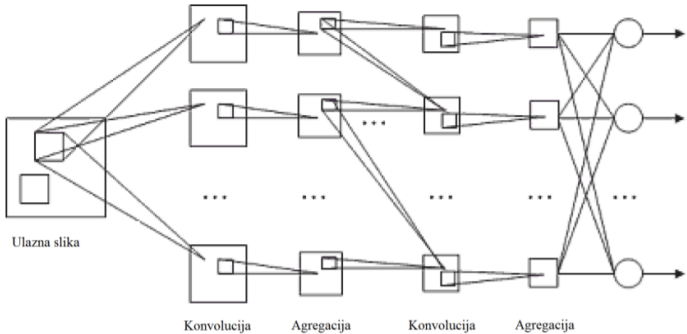
- Višekanalna konvolucija je važna i iz još jednog razloga
- Kada u prvom sloju imamo n filtera koji se primenjuju na polaznu sliku, dobijamo n slika

Konvolucija



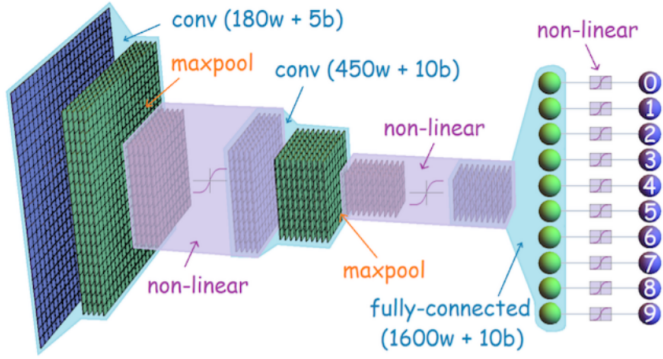
- u nastavku rada sa mrežom, treba da radimo sa tenzorima koji sadrže n dvodimenzionalnih slika (npr. ako je slika 200×200 i imamo 32 filtera, imaćemo tenzor $200 \times 200 \times 32$)

Konvolucija



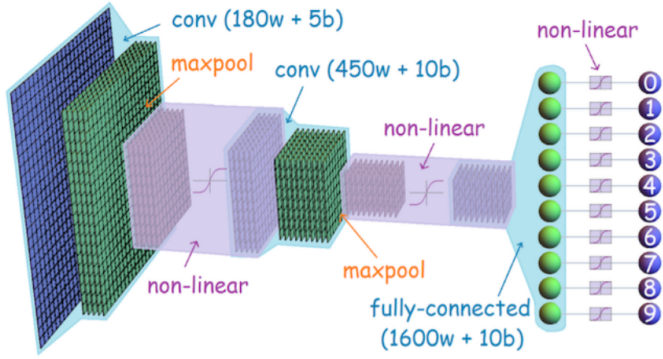
- u nastavku rada sa mrežom, treba da radimo sa tenzorima koji sadrže n dvodimenzionalnih slika (npr. ako je slika 200×200 i imamo 32 filtera, imaćemo tenzor $200 \times 200 \times 32$)
- svaka slika u sledećem konvolutivnom sloju se dobija primenom jedne operacije višekanalne konvolucije nad svim slikama iz prethodnog sloja

Konvolucija



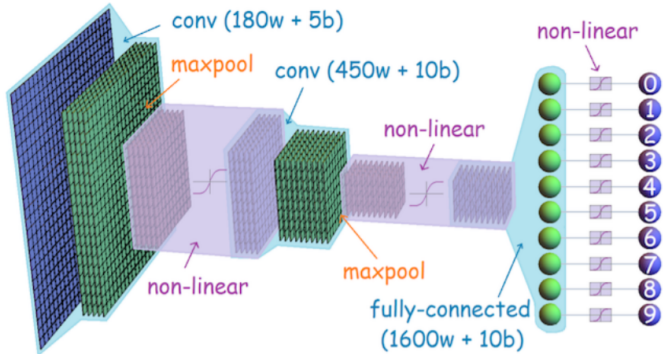
- imamo 5 kanala i na njih primenjujemo konvoluciju i dobijemo 10 kanala

Konvolucija



- imamo 5 kanala i na njih primenjujemo konvoluciju i dobijemo 10 kanala
- kako smo od 5 slika dobili 10?

Konvolucija



- imamo 5 kanala i na njih primenjujemo konvoluciju i dobijemo 10 kanala
- kako smo od 5 slika dobili 10?
- ako imamo 10 izlaznih slika, znači da imamo 10 filtera a svaki od filtera ima matricu koja je petokanalna, tj. svaki filter predstavlja petokanalni tenzor

Konvolucija

- Jedan konvolutivni sloj se sastoji od neurona koji dele vrednosti parametara i stoga ih aktivira ista vrsta ulaza

Konvolucija

- Jedan konvolutivni sloj se sastoji od neurona koji dele vrednosti parametara i stoga ih aktivira ista vrsta ulaza
- Svaka jedinica konvolutivnog sloja je dodeljena jednom delu ulaza (npr. slike)

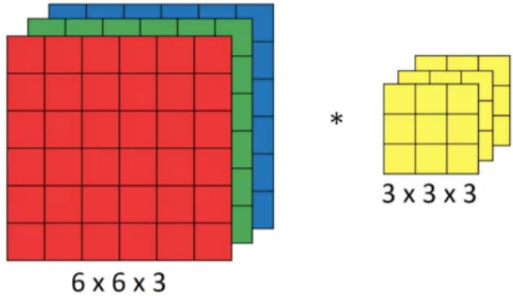
Konvolucija

- Jedan konvolutivni sloj se sastoji od neurona koji dele vrednosti parametara i stoga ih aktivira ista vrsta ulaza
- Svaka jedinica konvolutivnog sloja je dodeljena jednom delu ulaza (npr. slike)
- Tako konvolutivni sloj detektuje gde se u ulazu nalazi neka zakonitost

Konvolucija

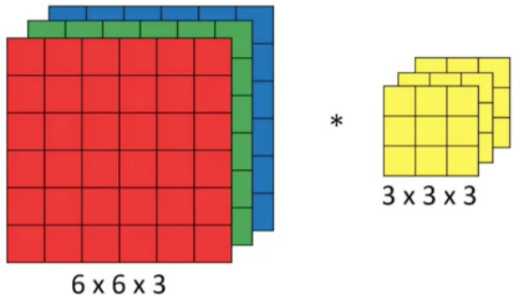
- Jedan konvolutivni sloj se sastoji od neurona koji dele vrednosti parametara i stoga ih aktivira ista vrsta ulaza
- Svaka jedinica konvolutivnog sloja je dodeljena jednom delu ulaza (npr. slike)
- Tako konvolutivni sloj detektuje gde se u ulazu nalazi neka zakonitost
- Deljenje parametara omogućava treniranje mreža sa ogromnim brojem jedinica

Agregacija



- Slojevi agregacije agregiraju informaciju iz konvolutivnih slojeva

Agregacija



- Slojevi agregacije agregiraju informaciju iz konvolutivnih slojeva
- Na primer, ako je ulaz iz prethodnog sloja dimenzije 6×6, a agregacija se primenjuje na 3×3, tada nećemo pomerati 3×3 matricu korak po korak kao kod konvolucije nego ćemo podeliti polaznu matricu na podmatrice dimenzije 3×3, na vrednosti koje se nalaze u podmatrici primeniti agregacionu funkciju i kao rezultat dobiti matricu dimenzije 2×2

Agregacija

- Kao agregaciona funkcija se najčešće koristi maksimum

Agregacija

- Kao agregaciona funkcija se najčešće koristi maksimum
- Zašto ne prosek?

Agregacija

- Kao agregaciona funkcija se najčešće koristi maksimum
- Zašto ne prosek?
- Maksimum radi bolje u praksi.

Agregacija

- Kao agregaciona funkcija se najčešće koristi maksimum
- Zašto ne prosek?
- Maksimum radi bolje u praksi.
- Setimo se da se konvolucija može interpretirati kao da filterom tražimo neki obrazac u slici i da visoke vrednosti kažu da smo uspeli da nađemo taj obrazac a niske da nismo

Agregacija

- Kao agregaciona funkcija se najčešće koristi maksimum
- Zašto ne prosek?
- Maksimum radi bolje u praksi.
- Setimo se da se konvolucija može interpretirati kao da filterom tražimo neki obrazac u slici i da visoke vrednosti kažu da smo uspeli da nađemo taj obrazac a niske da nismo
- Kada nad takvom matricom uradimo maksimum, on će nam reći da li smo na nekoj podmatrici 3×3 uspeli da nađemo traženi obrazac

Agregacija

- Kao agregaciona funkcija se najčešće koristi maksimum
- Zašto ne prosek?
- Maksimum radi bolje u praksi.
- Setimo se da se konvolucija može interpretirati kao da filterom tražimo neki obrazac u slici i da visoke vrednosti kažu da smo uspeli da nađemo taj obrazac a niske da nismo
- Kada nad takvom matricom uradimo maksimum, on će nam reći da li smo na nekoj podmatrici 3×3 uspeli da nađemo traženi obrazac
- Kada bismo nad takvom matricom uradili prosek, poništile bi se niže i više vrednosti konvolucije i izgubili bismo informaciju da li je obrazac pronađen

Agregacija

- Agregacija smanjuje dimenzionalnost podataka

Agregacija

- Agregacija smanjuje dimenzionalnost podataka
- Sve slike će biti manjih dimenzija i sve operacije sa njima će biti jeftinije

Agregacija

- Agregacija smanjuje dimenzionalnost podataka
- Sve slike će biti manjih dimenzija i sve operacije sa njima će biti jeftinije
- Takve operacije će i memorijski biti manje zahtevne što je važno u kontekstu ograničene RAM memorije na grafičkim karticama

Agregacija

- Agregacija smanjuje dimenzionalnost podataka
- Sve slike će biti manjih dimenzija i sve operacije sa njima će biti jeftinije
- Takve operacije će i memorijski biti manje zahtevne što je važno u kontekstu ograničene RAM memorije na grafičkim karticama
- Na kraju se nalazi potpuno povezana mreža

Agregacija

- Agregacija smanjuje dimenzionalnost podataka
- Sve slike će biti manjih dimenzija i sve operacije sa njima će biti jeftinije
- Takve operacije će i memorijski biti manje zahtevne što je važno u kontekstu ograničene RAM memorije na grafičkim karticama
- Na kraju se nalazi potpuno povezana mreža
- Što imamo više agregacionih slojeva, to će slike koje se generišu kasnije biti manjih dimenzija i dimenzija ulaza potpuno povezane mreže će biti manja

Agregacija

- Agregacija smanjuje dimenzionalnost podataka
- Sve slike će biti manjih dimenzija i sve operacije sa njima će biti jeftinije
- Takve operacije će i memorijski biti manje zahtevne što je važno u kontekstu ograničene RAM memorije na grafičkim karticama
- Na kraju se nalazi potpuno povezana mreža
- Što imamo više agregacionih slojeva, to će slike koje se generišu kasnije biti manjih dimenzija i dimenzija ulaza potpuno povezane mreže će biti manja
- Ako radimo sa niskodimenzionalnim vektorima na ulazu potpuno povezane mreže, onda će ova mreža imati manje parametara nego da radimo sa visokodimenzionalnim vektorima

Agregacija

- Agregacija smanjuje dimenzionalnost podataka
- Sve slike će biti manjih dimenzija i sve operacije sa njima će biti jeftinije
- Takve operacije će i memorijski biti manje zahtevne što je važno u kontekstu ograničene RAM memorije na grafičkim karticama
- Na kraju se nalazi potpuno povezana mreža
- Što imamo više agregacionih slojeva, to će slike koje se generišu kasnije biti manjih dimenzija i dimenzija ulaza potpuno povezane mreže će biti manja
- Ako radimo sa niskodimenzionalnim vektorima na ulazu potpuno povezane mreže, onda će ova mreža imati manje parametara nego da radimo sa visokodimenzionalnim vektorima
- Manji je rizik od preprilagođavanja

Sloj agregacije

- Slojevi agregacije agregiraju informaciju iz konvolutivnih slojeva

Sloj agregacije

- Slojevi agregacije agregiraju informaciju iz konvolutivnih slojeva
- Svaka jedinica sloja agregacije je dodeljena jednom delu prethodnog konvolutivnog sloja

Sloj agregacije

- Slojevi agregacije agregiraju informaciju iz konvolutivnih slojeva
- Svaka jedinica sloja agregacije je dodeljena jednom delu prethodnog konvolutivnog sloja
- Agregacija se najčešće vrši primenom maksimuma

Sloj agregacije

- Slojevi agregacije agregiraju informaciju iz konvolutivnih slojeva
- Svaka jedinica sloja agregacije je dodeljena jednom delu prethodnog konvolutivnog sloja
- Agregacija se najčešće vrši primenom maksimuma
- Time se smanjuje dimenzionalnost podataka, a čuva se informacija da li je negde u ulazu pronađena zakonitost koju konvolutivni sloj pronalazi

Sloj agregacije

- Slojevi agregacije agregiraju informaciju iz konvolutivnih slojeva

Sloj agregacije

- Slojevi agregacije agregiraju informaciju iz konvolutivnih slojeva
- Svaka jedinica sloja agregacije je dodeljena jednom delu prethodnog konvolutivnog sloja

Sloj agregacije

- Slojevi agregacije agregiraju informaciju iz konvolutivnih slojeva
- Svaka jedinica sloja agregacije je dodeljena jednom delu prethodnog konvolutivnog sloja
- Agregacija se najčešće vrši primenom maksimuma

Sloj agregacije

- Slojevi agregacije agregiraju informaciju iz konvolutivnih slojeva
- Svaka jedinica sloja agregacije je dodeljena jednom delu prethodnog konvolutivnog sloja
- Agregacija se najčešće vrši primenom maksimuma
- Time se smanjuje dimenzionalnost podataka, a čuva se informacija da li je negde u ulazu pronađena zakonitost koju konvolutivni sloj pronalazi

Interpretabilnost

- Konvolucija traži oblike koji najviše liče na matricu filtera

Interpretabilnost

- Konvolucija traži oblike koji najviše liče na matricu filtera
- Niži slojevi konvolutivne mreže detektuju jednostavne oblike.

Interpretabilnost

- Konvolucija traži oblike koji najviše liče na matricu filtera
- Niži slojevi konvolutivne mreže detektuju jednostavne oblike.
- Vrlo je lako ustanoviti koji oblik detektuju jedinice prvog konvolutivnog nivoa.

Interpretabilnost

- Konvolucija traži oblike koji najviše liče na matricu filtera
- Niži slojevi konvolutivne mreže detektuju jednostavne oblike.
- Vrlo je lako ustanoviti koji oblik detektuju jedinice prvog konvolutivnog nivoa.
- To je oblik za koji one daju najviše vrednosti na izlazima.

Interpretabilnost

- Konvolucija traži oblike koji najviše liče na matricu filtera
- Niži slojevi konvolutivne mreže detektuju jednostavne oblike.
- Vrlo je lako ustanoviti koji oblik detektuju jedinice prvog konvolutivnog nivoa.
- To je oblik za koji one daju najviše vrednosti na izlazima.
- Kako je aktivaciona funkcija rastuća, to je oblik koji daje najvišu vrednost linearne kombinacije koju jedinica računa.

Interpretabilnost

- Konvolucija traži oblike koji najviše liče na matricu filtera
- Niži slojevi konvolutivne mreže detektuju jednostavne oblike.
- Vrlo je lako ustanoviti koji oblik detektuju jedinice prvog konvolutivnog nivoa.
- To je oblik za koji one daju najviše vrednosti na izlazima.
- Kako je aktivaciona funkcija rastuća, to je oblik koji daje najvišu vrednost linearne kombinacije koju jedinica računa.
- Ako su koeficijenti jedinice w, oblik je dat kao rešenje problema

$$\max_{\|x\|=1} w \cdot x$$

Interpretabilnost

- Konvolucija traži oblike koji najviše liče na matricu filtera
- Niži slojevi konvolutivne mreže detektuju jednostavne oblike.
- Vrlo je lako ustanoviti koji oblik detektuju jedinice prvog konvolutivnog nivoa.
- To je oblik za koji one daju najviše vrednosti na izlazima.
- Kako je aktivaciona funkcija rastuća, to je oblik koji daje najvišu vrednost linearne kombinacije koju jedinica računa.
- Ako su koeficijenti jedinice w, oblik je dat kao rešenje problema

$$\max_{\|x\|=1} w \cdot x$$

- Normiranje je važno pošto bi u slučaju da je skalarni proizvod pozitivan, povećavanjem intenziteta vektora x, uvek mogla biti dostignuta proizvoljna vrednost skalarnog proizvoda.

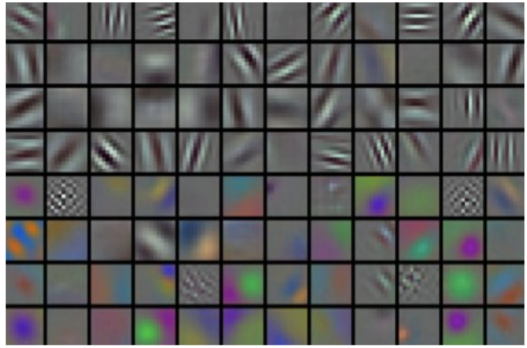
Interpretabilnost

- Konvolucija traži oblike koji najviše liče na matricu filtera
- Niži slojevi konvolutivne mreže detektuju jednostavne oblike.
- Vrlo je lako ustanoviti koji oblik detektuju jedinice prvog konvolutivnog nivoa.
- To je oblik za koji one daju najviše vrednosti na izlazima.
- Kako je aktivaciona funkcija rastuća, to je oblik koji daje najvišu vrednost linearne kombinacije koju jedinica računa.
- Ako su koeficijenti jedinice w, oblik je dat kao rešenje problema

$$\max_{\|x\|=1} w \cdot x$$

- Normiranje je važno pošto bi u slučaju da je skalarni proizvod pozitivan, povećavanjem intenziteta vektora x, uvek mogla biti dostignuta proizvoljna vrednost skalarnog proizvoda.
- Pod tim uslovom, lako se ustanovljava (recimo, postavljanjem parcijalnih izvoda po x na 0) da se maksimalna vrednost dostiže za vrednost $w/\|w\|$.

Interpretabilnost



- Stoga, da bi se prikazao oblik koji jedinica najnižeg konvolutivnog sloja prepoznaje, dovoljno je vizualizovati njene koeficijente u vidu slike čije dimenzije odgovaraju dimenzijama filtera koji ta jedinica predstavlja.

Interpretabilnost

- Prikaz oblika koje prepoznaju jedinice višeg nivoa mreže koja prepoznaje objekte na slikama.

Interpretabilnost

- Prikaz oblika koje prepoznaju jedinice višeg nivoa mreže koja prepoznaje objekte na slikama.
- Vizualizacija oblika koje prepoznaju viši nivoi konvolutivne mreže je komplikovanija i zahteva neki vid optimizacije, kako bi se našli ulazi u mrežu za koje ove jedinice daju najviše vrednosti.

Kvaliteti

- *proređene interakcije*, pod čime se podrazumeva da je svaka jedinica povezana samo sa malim brojem jedinica iz prethodnog sloja, umesto sa svim, kao što u slučaju potpuno povezanih mreža

Kvaliteti

- *proređene interakcije*, pod čime se podrazumeva da je svaka jedinica povezana samo sa malim brojem jedinica iz prethodnog sloja, umesto sa svim, kao što u slučaju potpuno povezanih mreža
- *deljenje parametara*, koje se odnosi na to da sve jedinice jednog kanala imaju iste parametre – definisane filterom tog kanala

Kvaliteti

- *proređene interakcije*, pod čime se podrazumeva da je svaka jedinica povezana samo sa malim brojem jedinica iz prethodnog sloja, umesto sa svim, kao što u slučaju potpuno povezanih mreža
- *deljenje parametara*, koje se odnosi na to da sve jedinice jednog kanala imaju iste parametre – definisane filterom tog kanala
  - Zahvaljujući proređenim interakcijama i deljenju parametara postoje neka lepa svojstva koja se uglavnom tiču dobrih mogućnosti paralelizacije

Kvaliteti

- *proređene interakcije*, pod čime se podrazumeva da je svaka jedinica povezana samo sa malim brojem jedinica iz prethodnog sloja, umesto sa svim, kao što u slučaju potpuno povezanih mreža
- *deljenje parametara*, koje se odnosi na to da sve jedinice jednog kanala imaju iste parametre – definisane filterom tog kanala
  - Zahvaljujući proređenim interakcijama i deljenju parametara postoje neka lepa svojstva koja se uglavnom tiču dobrih mogućnosti paralelizacije
  - Možemo računati da ćemo imati brže učenje u istom poslu u odnosu na onaj gde primenjujemo rekurentnu mrežu

Kvaliteti

- *neosetljivost na translacije*, svojstvo da će neki signal koji mreža traži biti nađen bez obzira kako je transliran na slici

Kvaliteti

- *neosetljivost na translacije*, svojstvo da će neki signal koji mreža traži biti nađen bez obzira kako je transliran na slici
- *specijalizovanost za topologiju signala*

Kvaliteti

- *neosetljivost na translacije*, svojstvo da će neki signal koji mreža traži biti nađen bez obzira kako je transliran na slici
- *specijalizovanost za topologiju signala*
  - Konvolutivne mreže imaju u vidu da su pikseli stvari koje su susedne i da je ta susednost na neki način relevantna - susedni pikseli ulaze u istu matricu konvolucije

Kvaliteti

- *neosetljivost na translacije*, svojstvo da će neki signal koji mreža traži biti nađen bez obzira kako je transliran na slici
- *specijalizovanost za topologiju signala*
  - Konvolutivne mreže imaju u vidu da su pikseli stvari koje su susedne i da je ta susednost na neki način relevantna - susedni pikseli ulaze u istu matricu konvolucije
  - To nije slučaj sa potpuno povezanim mrežama

Kvaliteti

- *neosetljivost na translacije*, svojstvo da će neki signal koji mreža traži biti nađen bez obzira kako je transliran na slici
- *specijalizovanost za topologiju signala*
  - Konvolutivne mreže imaju u vidu da su pikseli stvari koje su susedne i da je ta susednost na neki način relevantna - susedni pikseli ulaze u istu matricu konvolucije
  - To nije slučaj sa potpuno povezanim mrežama
- *delimična interpretabilnost*, zahvaljujući tome što se mogu vizualizovati ulazi na koje filteri daju najjači odgovor.

Mane

- mreža je osetljiva na neke druge transformacije, poput rotacije i skaliranja (homotetije)

Mane

- mreža je osetljiva na neke druge transformacije, poput rotacije i skaliranja (homotetije)
  - Za navedenu osnovnu vrstu konvolutivnih mreža ne bismo imali lak način da se borimo sa slikama različitih veličina

Mane

- mreža je osetljiva na neke druge transformacije, poput rotacije i skaliranja (homotetije)
  - Za navedenu osnovnu vrstu konvolutivnih mreža ne bismo imali lak način da se borimo sa slikama različitih veličina
  - Jedino tako što bismo ih skalirali tako da budu sve iste veličine ali pritom bismo sigurno došlo do gubitka informacije

Mane

- mreža je osetljiva na neke druge transformacije, poput rotacije i skaliranja (homotetije)
  - Za navedenu osnovnu vrstu konvolutivnih mreža ne bismo imali lak način da se borimo sa slikama različitih veličina
  - Jedino tako što bismo ih skalirali tako da budu sve iste veličine ali pritom bismo sigurno došlo do gubitka informacije
  - Postoje posebne mreže za ulaze različitih veličina

Mane

- mreža je osetljiva na neke druge transformacije, poput rotacije i skaliranja (homotetije)
  - Za navedenu osnovnu vrstu konvolutivnih mreža ne bismo imali lak način da se borimo sa slikama različitih veličina
  - Jedino tako što bismo ih skalirali tako da budu sve iste veličine ali pritom bismo sigurno došlo do gubitka informacije
  - Postoje posebne mreže za ulaze različitih veličina
  - Postoje posebni načini da se mreža učini robusnom na rotacije

Mane

- mreža je osetljiva na neke druge transformacije, poput rotacije i skaliranja (homotetije)
  - Za navedenu osnovnu vrstu konvolutivnih mreža ne bismo imali lak način da se borimo sa slikama različitih veličina
  - Jedino tako što bismo ih skalirali tako da budu sve iste veličine ali pritom bismo sigurno došlo do gubitka informacije
  - Postoje posebne mreže za ulaze različitih veličina
  - Postoje posebni načini da se mreža učini robusnom na rotacije
- Šta tačno u arhitekturi konvolutivne mreže sprečava primenu na slike različitih veličina?

Mane

- mreža je osetljiva na neke druge transformacije, poput rotacije i skaliranja (homotetije)
  - Za navedenu osnovnu vrstu konvolutivnih mreža ne bismo imali lak način da se borimo sa slikama različitih veličina
  - Jedino tako što bismo ih skalirali tako da budu sve iste veličine ali pritom bismo sigurno došlo do gubitka informacije
  - Postoje posebne mreže za ulaze različitih veličina
  - Postoje posebni načini da se mreža učini robusnom na rotacije
- Šta tačno u arhitekturi konvolutivne mreže sprečava primenu na slike različitih veličina?
  - kako potpuno povezana mreža koja se nalazi na kraju niza slojeva konvolutivne mreže uvek mora imati fiksiran broj ulaza, tako i konvolutivni deo mora proizvesti izlaz tačno određenih dimenzija, što zbog fiksiranog broja slojeva agregacije i fiksirane rezolucije agregiranja (npr. 3×3 vrednosti se zamenjuju jednom), znači i da ulazi moraju biti istih dimenzija.

Mane

- mreža je osetljiva na neke druge transformacije, poput rotacije i skaliranja (homotetije)
  - Za navedenu osnovnu vrstu konvolutivnih mreža ne bismo imali lak način da se borimo sa slikama različitih veličina
  - Jedino tako što bismo ih skalirali tako da budu sve iste veličine ali pritom bismo sigurno došlo do gubitka informacije
  - Postoje posebne mreže za ulaze različitih veličina
  - Postoje posebni načini da se mreža učini robusnom na rotacije
- Šta tačno u arhitekturi konvolutivne mreže sprečava primenu na slike različitih veličina?
  - kako potpuno povezana mreža koja se nalazi na kraju niza slojeva konvolutivne mreže uvek mora imati fiksiran broj ulaza, tako i konvolutivni deo mora proizvesti izlaz tačno određenih dimenzija, što zbog fiksiranog broja slojeva agregacije i fiksirane rezolucije agregiranja (npr. 3×3 vrednosti se zamenjuju jednom), znači i da ulazi moraju biti istih dimenzija.
  - Ovaj problem se može rešiti tehnikom *global average pooling*

Mane

- Pored diskutovanih problema koji su specifični za konvolutivne mreže, problemi diskutovani u slučaju potpuno povezanih mreža su takođe prisutni

Mane

- Pored diskutovanih problema koji su specifični za konvolutivne mreže, problemi diskutovani u slučaju potpuno povezanih mreža su takođe prisutni
- To što su ublaženi navedenim kvalitetima konvolutivnih mreža, vodi povećanju ambicija i izgradnji većih mreža, sa većim brojem parametara, pa isti problemi ponovo dolaze do izražaja

Mane

- Pored diskutovanih problema koji su specifični za konvolutivne mreže, problemi diskutovani u slučaju potpuno povezanih mreža su takođe prisutni
- To što su ublaženi navedenim kvalitetima konvolutivnih mreža, vodi povećanju ambicija i izgradnji većih mreža, sa većim brojem parametara, pa isti problemi ponovo dolaze do izražaja
- Ove mreže su duboke pa postoji opasnost od nestajućih i eksplodirajućih gradijenata

Mane

- Pored diskutovanih problema koji su specifični za konvolutivne mreže, problemi diskutovani u slučaju potpuno povezanih mreža su takođe prisutni
- To što su ublaženi navedenim kvalitetima konvolutivnih mreža, vodi povećanju ambicija i izgradnji većih mreža, sa većim brojem parametara, pa isti problemi ponovo dolaze do izražaja
- Ove mreže su duboke pa postoji opasnost od nestajućih i eksplodirajućih gradijenata
- Gradijenti se ponovo izračunavaju pomoću algoritma propagacije unazad koji radi veliki broj množenja proporcionalan dimenziji mreže, pa ako je mpreza npr. dubine 100, očekujemo da gradijent eksplodira ili nestane, ili na nekim koordinatama eksplodira a na drugim nestane

Mane

- Pored diskutovanih problema koji su specifični za konvolutivne mreže, problemi diskutovani u slučaju potpuno povezanih mreža su takođe prisutni
- To što su ublaženi navedenim kvalitetima konvolutivnih mreža, vodi povećanju ambicija i izgradnji većih mreža, sa većim brojem parametara, pa isti problemi ponovo dolaze do izražaja
- Ove mreže su duboke pa postoji opasnost od nestajućih i eksplodirajućih gradijenata
- Gradijenti se ponovo izračunavaju pomoću algoritma propagacije unazad koji radi veliki broj množenja proporcionalan dimenziji mreže, pa ako je mpreza npr. dubine 100, očekujemo da gradijent eksplodira ili nestane, ili na nekim koordinatama eksplodira a na drugim nestane
- Postoje i rešenja za to

Pregled

Potpuno povezane neuronske mreže

Konvolutivne neuronske mreže

Rekurentne neuronske mreže

Praktične tehnike i napredni koncepti

Rekurentne neuronske mreže

- Rekurentne neuronske mreže predstavljaju arhitekturu mreža specijalizovanu za obradu sekvencijalnih podataka, poput rečenica prirodnog jezika i vremenskih serija

Rekurentne neuronske mreže

- Rekurentne neuronske mreže predstavljaju arhitekturu mreža specijalizovanu za obradu sekvencijalnih podataka, poput rečenica prirodnog jezika i vremenskih serija
- Za sekvencijalne podatke se mogu koristiti i konvolutivne mreže

Rekurentne neuronske mreže

- Rekurentne neuronske mreže predstavljaju arhitekturu mreža specijalizovanu za obradu sekvencijalnih podataka, poput rečenica prirodnog jezika i vremenskih serija
- Za sekvencijalne podatke se mogu koristiti i konvolutivne mreže
- Trenutni trend je da se smanjuje upotreba rekurentnih mreža i da se sve više insistira na upotrebi konvolutivnih mreža i u situacijama za koje rekurentne mreže deluju kao prirodno rešenje jer ih nije lako trenirati

Rekurentne neuronske mreže

- Rekurentne neuronske mreže predstavljaju arhitekturu mreža specijalizovanu za obradu sekvencijalnih podataka, poput rečenica prirodnog jezika i vremenskih serija
- Za sekvencijalne podatke se mogu koristiti i konvolutivne mreže
- Trenutni trend je da se smanjuje upotreba rekurentnih mreža i da se sve više insistira na upotrebi konvolutivnih mreža i u situacijama za koje rekurentne mreže deluju kao prirodno rešenje jer ih nije lako trenirati
- Ipak, to ne znači da su rekurentne mreže izašle iz upotrebe

Rekurentne neuronske mreže

- Razmotrimo primer automatskog prevođenja sa jednog jezika na drugi

Rekurentne neuronske mreže

- Razmotrimo primer automatskog prevođenja sa jednog jezika na drugi
- Prevođenje se vrši na nivou rečenice, ne na nivou reči

Rekurentne neuronske mreže

- Razmotrimo primer automatskog prevođenja sa jednog jezika na drugi
- Prevođenje se vrši na nivou rečenice, ne na nivou reči
- Rečenice mogu biti različitih dužina

Rekurentne neuronske mreže

- Razmotrimo primer automatskog prevođenja sa jednog jezika na drugi
- Prevođenje se vrši na nivou rečenice, ne na nivou reči
- Rečenice mogu biti različitih dužina
- Treba nam arhitektura mreže koja je prirodno formulisana za slučaj ulaza različitih dužina

Rekurentne neuronske mreže

- Razmotrimo primer automatskog prevođenja sa jednog jezika na drugi
- Prevođenje se vrši na nivou rečenice, ne na nivou reči
- Rečenice mogu biti različitih dužina
- Treba nam arhitektura mreže koja je prirodno formulisana za slučaj ulaza različitih dužina
- Ako bismo probali da implementiramo prevođenje pomoću potpuno povezanih mrežama,

Rekurentne neuronske mreže

- Razmotrimo primer automatskog prevođenja sa jednog jezika na drugi
- Prevođenje se vrši na nivou rečenice, ne na nivou reči
- Rečenice mogu biti različitih dužina
- Treba nam arhitektura mreže koja je prirodno formulisana za slučaj ulaza različitih dužina
- Ako bismo probali da implementiramo prevođenje pomoću potpuno povezanih mrežama,
  - problem je fiksna dužina ulaza

Rekurentne neuronske mreže

- Razmotrimo primer automatskog prevođenja sa jednog jezika na drugi
- Prevođenje se vrši na nivou rečenice, ne na nivou reči
- Rečenice mogu biti različitih dužina
- Treba nam arhitektura mreže koja je prirodno formulisana za slučaj ulaza različitih dužina
- Ako bismo probali da implementiramo prevođenje pomoću potpuno povezanih mrežama,
  - problem je fiksna dužina ulaza
  - može se nadomestiti tako što će se za ulaz uvek uzimati dužina najduže rečenice pri čemu će ulazi koji se ne koriste biti posebno obeleženi

Rekurentne neuronske mreže

- Razmotrimo primer automatskog prevođenja sa jednog jezika na drugi
- Prevođenje se vrši na nivou rečenice, ne na nivou reči
- Rečenice mogu biti različitih dužina
- Treba nam arhitektura mreže koja je prirodno formulisana za slučaj ulaza različitih dužina
- Ako bismo probali da implementiramo prevođenje pomoću potpuno povezanih mrežama,
  - problem je fiksna dužina ulaza
  - može se nadomestiti tako što će se za ulaz uvek uzimati dužina najduže rečenice pri čemu će ulazi koji se ne koriste biti posebno obeleženi
  - neekonomično jer će većina rečenice biti kraća od najduže što će otežavati učenje, a najduže rečenice će se retko pojavljivati

Rekurentne neuronske mreže

- Razmotrimo primer klasifikacije logističkom regresijom ili linearnom regresijom

Rekurentne neuronske mreže

- Razmotrimo primer klasifikacije logističkom regresijom ili linearnom regresijom
- Za linearne modele je karakteristično (a potpuno povezana neuronska mreža je sastavljena od linearnih modela) da svaki od ulaza ima specifično značenje: npr. ako klasifikujemo pacijente po tome da li imaju neko oboljenje ili ne, to radimo na osnovu nekih atributa gde je na primer prvi hemoglobin, drugi transaminaze, itd. i taj redosled mora biti uvek isti

Rekurentne neuronske mreže

- Razmotrimo primer klasifikacije logističkom regresijom ili linearnom regresijom
- Za linearne modele je karakteristično (a potpuno povezana neuronska mreža je sastavljena od linearnih modela) da svaki od ulaza ima specifično značenje: npr. ako klasifikujemo pacijente po tome da li imaju neko oboljenje ili ne, to radimo na osnovu nekih atributa gde je na primer prvi hemoglobin, drugi transaminaze, itd. i taj redosled mora biti uvek isti
- Kada govorimo o rečenicama, ne možemo to očekivati

Rekurentne neuronske mreže

- Razmotrimo primer klasifikacije logističkom regresijom ili linearnom regresijom
- Za linearne modele je karakteristično (a potpuno povezana neuronska mreža je sastavljena od linearnih modela) da svaki od ulaza ima specifično značenje: npr. ako klasifikujemo pacijente po tome da li imaju neko oboljenje ili ne, to radimo na osnovu nekih atributa gde je na primer prvi hemoglobin, drugi transaminaze, itd. i taj redosled mora biti uvek isti
- Kada govorimo o rečenicama, ne možemo to očekivati
  - nije fiksirano značenje prve reči prve reči u rečenici ili druge reči u rečenici

Rekurentne neuronske mreže

- Razmotrimo primer klasifikacije logističkom regresijom ili linearnom regresijom
- Za linearne modele je karakteristično (a potpuno povezana neuronska mreža je sastavljena od linearnih modela) da svaki od ulaza ima specifično značenje: npr. ako klasifikujemo pacijente po tome da li imaju neko oboljenje ili ne, to radimo na osnovu nekih atributa gde je na primer prvi hemoglobin, drugi transaminaze, itd. i taj redosled mora biti uvek isti
- Kada govorimo o rečenicama, ne možemo to očekivati
  - nije fiksirano značenje prve reči prve reči u rečenici ili druge reči u rečenici
  - npr. ne možemo očekivati da će uvek ići prvo subjekat pa predikat itd.

Rekurentne neuronske mreže

- Razmotrimo primer klasifikacije logističkom regresijom ili linearnom regresijom
- Za linearne modele je karakteristično (a potpuno povezana neuronska mreža je sastavljena od linearnih modela) da svaki od ulaza ima specifično značenje: npr. ako klasifikujemo pacijente po tome da li imaju neko oboljenje ili ne, to radimo na osnovu nekih atributa gde je na primer prvi hemoglobin, drugi transaminaze, itd. i taj redosled mora biti uvek isti
- Kada govorimo o rečenicama, ne možemo to očekivati
  - nije fiksirano značenje prve reči prve reči u rečenici ili druge reči u rečenici
  - npr. ne možemo očekivati da će uvek ići prvo subjekat pa predikat itd.
  - možemo u rečenici permutovati reči tako da rečenica drugačije zvuči ali znači jednu te istu stvar

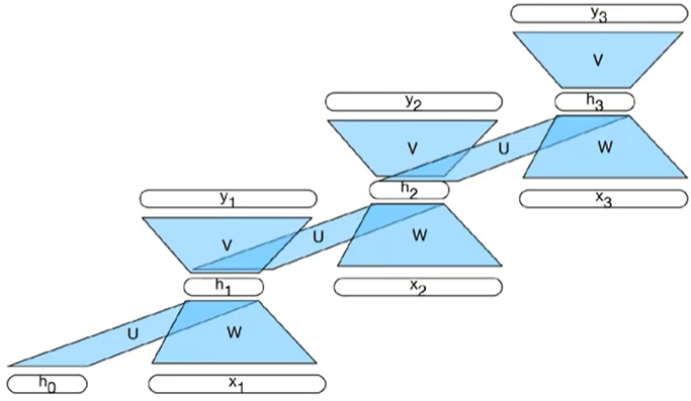
Rekurentne neuronske mreže

- Razmotrimo primer klasifikacije logističkom regresijom ili linearnom regresijom
- Za linearne modele je karakteristično (a potpuno povezana neuronska mreža je sastavljena od linearnih modela) da svaki od ulaza ima specifično značenje: npr. ako klasifikujemo pacijente po tome da li imaju neko oboljenje ili ne, to radimo na osnovu nekih atributa gde je na primer prvi hemoglobin, drugi transaminaze, itd. i taj redosled mora biti uvek isti
- Kada govorimo o rečenicama, ne možemo to očekivati
  - nije fiksirano značenje prve reči prve reči u rečenici ili druge reči u rečenici
  - npr. ne možemo očekivati da će uvek ići prvo subjekat pa predikat itd.
  - možemo u rečenici permutovati reči tako da rečenica drugačije zvuči ali znači jednu te istu stvar
- Potpuno povezane mreže nemaju nikakav mehanizam da to uzmu u obzir

Rekurentne neuronske mreže

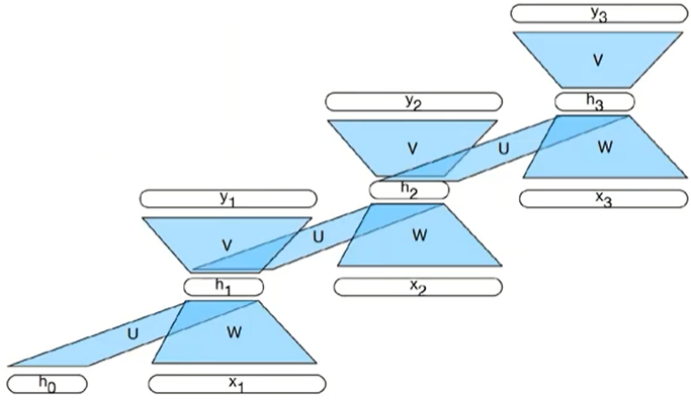
- Razmotrimo primer klasifikacije logističkom regresijom ili linearnom regresijom
- Za linearne modele je karakteristično (a potpuno povezana neuronska mreža je sastavljena od linearnih modela) da svaki od ulaza ima specifično značenje: npr. ako klasifikujemo pacijente po tome da li imaju neko oboljenje ili ne, to radimo na osnovu nekih atributa gde je na primer prvi hemoglobin, drugi transaminaze, itd. i taj redosled mora biti uvek isti
- Kada govorimo o rečenicama, ne možemo to očekivati
  - nije fiksirano značenje prve reči prve reči u rečenici ili druge reči u rečenici
  - npr. ne možemo očekivati da će uvek ići prvo subjekat pa predikat itd.
  - možemo u rečenici permutovati reči tako da rečenica drugačije zvuči ali znači jednu te istu stvar
- Potpuno povezane mreže nemaju nikakav mehanizam da to uzmu u obzir
- Teorijski one jesu univerzalni aproksimatori i mogu sve da rade, ovo bi bilo moguće, ali mreža koja bi mogla da nauči otpornost na permutacije bi bila ogromna i pitanje je da li bi mogla da se istrenira

Formulacija modela



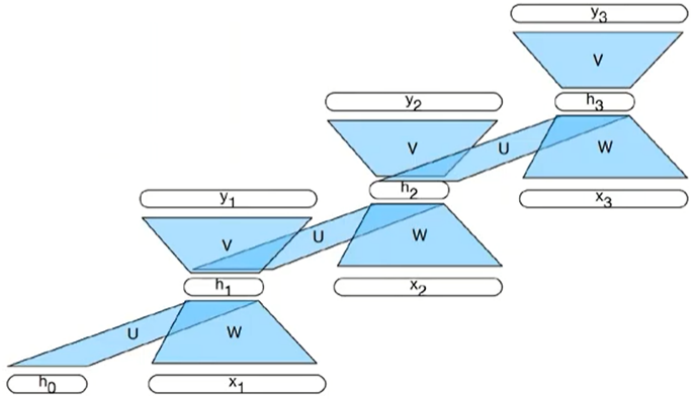
- Rekurentna mreža obrađuje jedan element sekvence u jednom trenutku (npr. jednu reč iz rečenice)

Formulacija modela



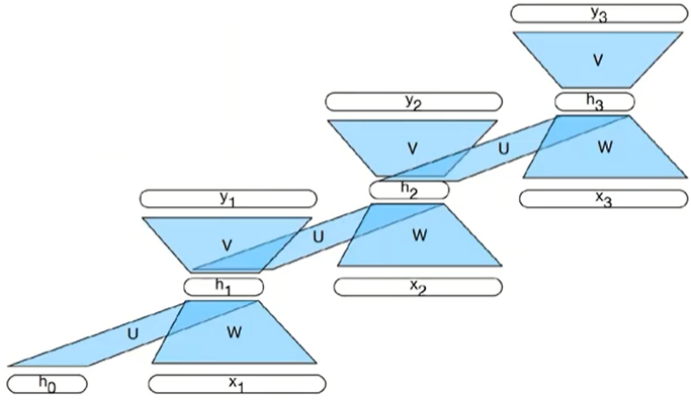
- Rekurentna mreža obrađuje jedan element sekvence u jednom trenutku (npr. jednu reč iz rečenice)
- Na ulaze dovodimo jednu po jednu reč iz rečenice

Formulacija modela



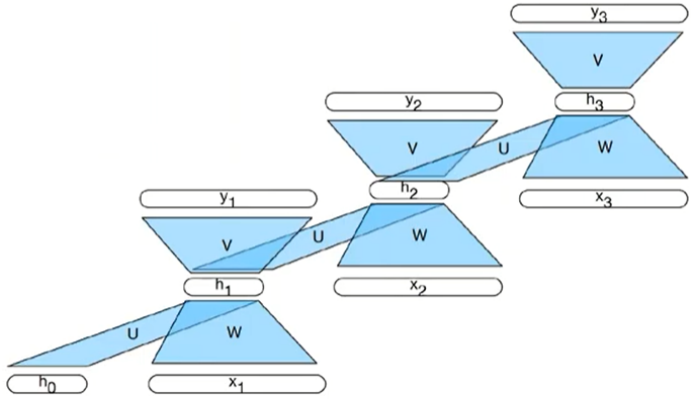
- Rekurentna mreža obrađuje jedan element sekvence u jednom trenutku (npr. jednu reč iz rečenice)
- Na ulaze dovodimo jednu po jednu reč iz rečenice
- Sledeća reč se dovodi na ulaz nakon što je izračunavanje za prethodnu reč završeno

Formulacija modela



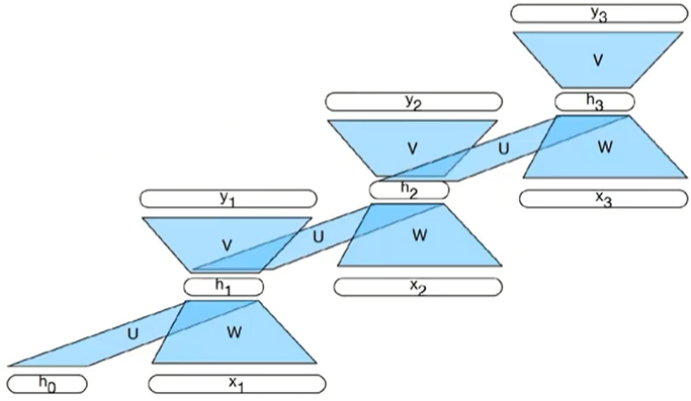
- Rekurentna mreža održava tzv. skriveno stanje (vektori h) i u svakom koraku ima tekuće skriveno stanje

Formulacija modela



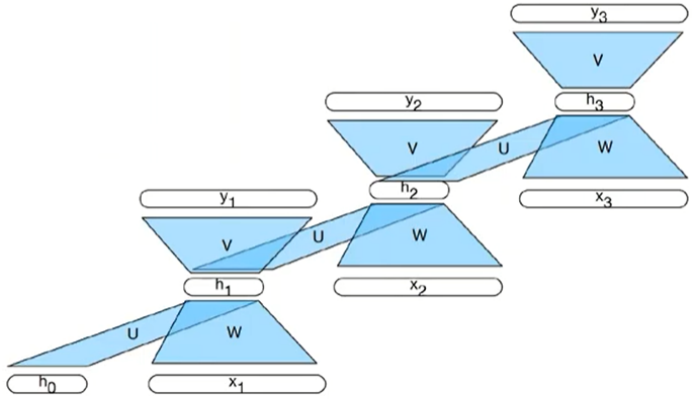
- Rekurentna mreža održava tzv. skriveno stanje (vektori h) i u svakom koraku ima tekuće skriveno stanje
- Za dobijeni ulaz (jednu reč), na osnovu tog ulaza i tekućeg skrivenog stanja (npr. h_0 i x_1) izračunava novo skriveno stanje (npr. x_1) kao linearna kombinacija matrica U i W sa vektorima tekućeg skrivenog stanja i tekućeg ulaza

Formulacija modela



- Na dobijeni rezultat se primeni aktivaciona funkcija i tako se dobije novo skriveno stanje (npr. h_1)

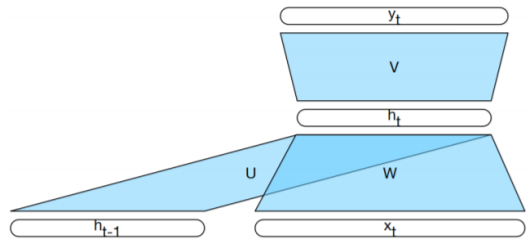
Formulacija modela



- Na dobijeni rezultat se primeni aktivaciona funkcija i tako se dobije novo skriveno stanje (npr. h_1)
- onda dolazi na red nova reč (npr. x_2), na osnovu tekućeg stanja (npr. h_1) i reprezentacije reči se uz pomoć istih marica U i W izvrše linearne transformacije, primeni aktivaciona funkcija i dobije novo stanje (npr. h_2) i tako redom za sve reči

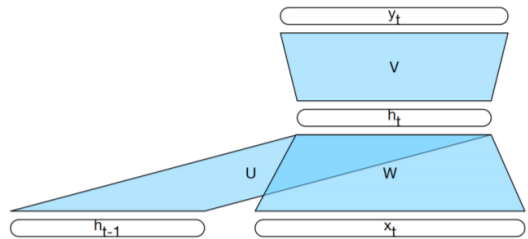
Formulacija modela

- u svakom koraku se iz skrivenog stanja može generisati neki izlaz (y_i)



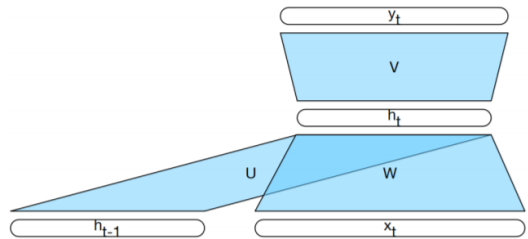
Formulacija modela

- u svakom koraku se iz skrivenog stanja može generisati neki izlaz (y_i)
- takav izlaz može da zatreba ako želimo da određujemo vrstu reči u rečenici



Formulacija modela

- u svakom koraku se iz skrivenog stanja može generisati neki izlaz (y_i)
- takav izlaz može da zatreba ako želimo da određujemo vrstu reči u rečenici
- nekad za datu rečenicu želimo da odredimo njen sentiment (npr. ljutnja, tuga) i tada nam ne treba izlaz u svakom koraku

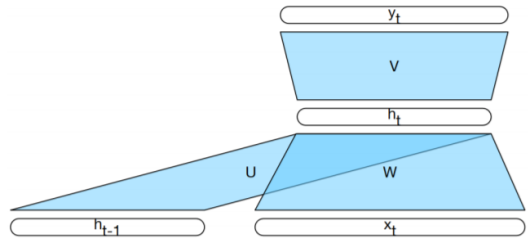


Formulacija modela

$$h^{(0)} = 0$$

$$h^{(t)} = g\left(b_h + Wh^{(t-1)} + Ux^{(t)}\right)$$

$$o^{(t)} = g\left(b_o + Vh^{(t)}\right)$$



Formulacija modela

- Kako se ovakav model trenira?

Formulacija modela

- Kako se ovakav model trenira?
- Kako se kod ovakvog modela određuje funkcija greške?

Formulacija modela

- Kako se ovakav model trenira?
- Kako se kod ovakvog modela određuje funkcija greške?
- Greška se računa na osnovu sekvence vrednosti y i sekvence predviđanja, odnosno funkcija greške ima oblik:

$$L([y^{(1)}, \ldots, y^{(T)}], [f_w^{(1)}, \ldots, f_w^{(T)}])$$

gde se uglasim zagradama označava sekvenca, a $f_w^{(t)}$ predstavlja predviđanje modela u koraku t i moglo bi se zapisati kao

$$f_w^{(t)} = f_w([x^{(1)}, \ldots, x^{(t)}])$$

Formulacija modela

- Kako se ovakav model trenira?
- Kako se kod ovakvog modela određuje funkcija greške?
- Greška se računa na osnovu sekvence vrednosti y i sekvence predviđanja, odnosno funkcija greške ima oblik:

$$L([y^{(1)}, \ldots, y^{(T)}], [f_w^{(1)}, \ldots, f_w^{(T)}])$$

gde se uglasim zagradama označava sekvenca, a $f_w^{(t)}$ predstavlja predviđanje modela u koraku t i moglo bi se zapisati kao

$$f_w^{(t)} = f_w([x^{(1)}, \ldots, x^{(t)}])$$

- Česta je pretpostavka da se funkcija greške može razložiti na sledeći način:

$$L([y^{(1)}, \ldots, y^{(T)}], [f_w^{(1)}, \ldots, f_w^{(T)}]) = \sum_{t=1}^{T} L(y^{(t)}, f_w^{(t)})$$

Formulacija modela

$$L([y^{(1)}, \ldots, y^{(T)}], [f_w^{(1)}, \ldots, f_w^{(T)}]) = \sum_{t=1}^{T} L(y^{(t)}, f_w^{(t)})$$

- Ovakav oblik funkcije greške je pogodan za probleme poput određivanja vrste reči, gde svakoj reči odgovara neka vrsta

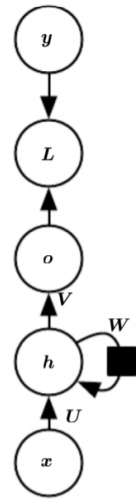
Formulacija modela

$$L([y^{(1)}, \ldots, y^{(T)}], [f_w^{(1)}, \ldots, f_w^{(T)}]) = \sum_{t=1}^{T} L(y^{(t)}, f_w^{(t)})$$

- Ovakav oblik funkcije greške je pogodan za probleme poput određivanja vrste reči, gde svakoj reči odgovara neka vrsta
- Kod problema gde nam je rezultat krajnji izlaz (npr. određivanje sentimenta), ima smisla određivati grešku samo za njega umesto za ceo vektor izlaza

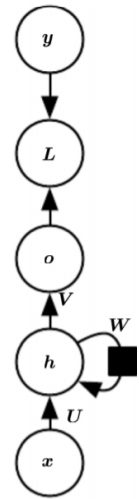
Graf izračunavanja

- Opisuje izračunavanja koje mreža vrši



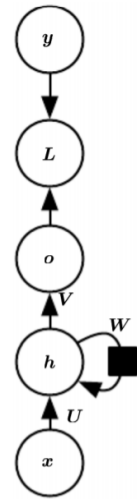
Graf izračunavanja

- Opisuje izračunavanja koje mreža vrši
- Kvadrat na grani nam ukazuje na to da se uzima prethodna vrednost h, množi sa W i onda se na to dodaje proizvod Ux i čuva kao novo stanje h



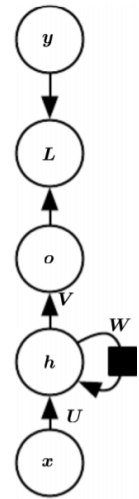
Graf izračunavanja

- Opisuje izračunavanja koje mreža vrši
- Kvadrat na grani nam ukazuje na to da se uzima prethodna vrednost h, množi sa W i onda se na to dodaje proizvod Ux i čuva kao novo stanje h
- Operacije predstavljene granama sa kvadratom su u različitim trenucima a one predstavljene granama bez kvadrata su sinhrone



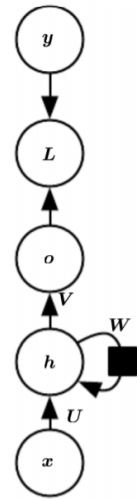
Graf izračunavanja

- Opisuje izračunavanja koje mreža vrši
- Kvadrat na grani nam ukazuje na to da se uzima prethodna vrednost h, množi sa W i onda se na to dodaje proizvod Ux i čuva kao novo stanje h
- Operacije predstavljene granama sa kvadratom su u različitim trenucima a one predstavljene granama bez kvadrata su sinhrone
- Nakon toga se h množi sa V i dobija se izlaz o



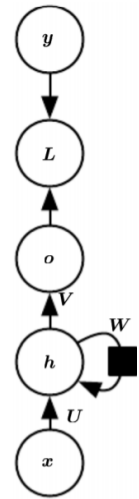
Graf izračunavanja

- Opisuje izračunavanja koje mreža vrši
- Kvadrat na grani nam ukazuje na to da se uzima prethodna vrednost h, množi sa W i onda se na to dodaje proizvod Ux i čuva kao novo stanje h
- Operacije predstavljene granama sa kvadratom su u različitim trenucima a one predstavljene granama bez kvadrata su sinhrone
- Nakon toga se h množi sa V i dobija se izlaz o
- S obzirom da je potrebno minimizovati funkciju greške (L), izlaz o prosleđujemo zajedno sa tačnim rešenjem y funkciji greške L



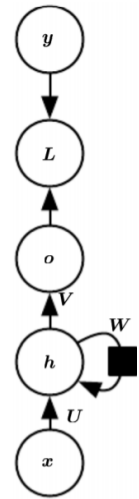
Graf izračunavanja

- Da bi se minimizovala funkcija greške, potrebno je izračunati njene gradijente



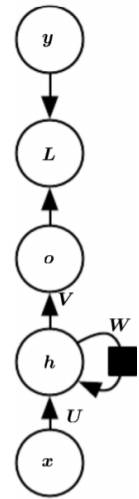
Graf izračunavanja

- Da bi se minimizovala funkcija greške, potrebno je izračunati njene gradijente
- Nad čvorom L pokrećemo algoritam propagacije unazad i izračunavamo gradijente u skladu sa ovim grafom i koristimo tako izračunate gradijente u standardnim optimizacionim metodama



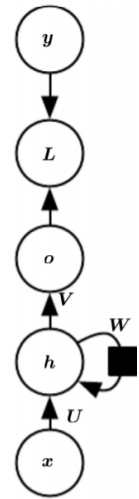
Graf izračunavanja

- Da bi se minimizovala funkcija greške, potrebno je izračunati njene gradijente
- Nad čvorom L pokrećemo algoritam propagacije unazad i izračunavamo gradijente u skladu sa ovim grafom i koristimo tako izračunate gradijente u standardnim optimizacionim metodama
- Kako se izračunava gradijent kada imamo rekurentne veze?



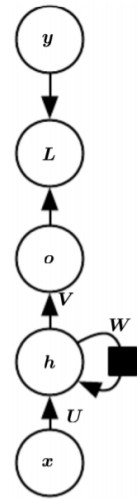
Graf izračunavanja

- Da bi se minimizovala funkcija greške, potrebno je izračunati njene gradijente
- Nad čvorom L pokrećemo algoritam propagacije unazad i izračunavamo gradijente u skladu sa ovim grafom i koristimo tako izračunate gradijente u standardnim optimizacionim metodama
- Kako se izračunava gradijent kada imamo rekurentne veze?
- Algoritam propagacije unazad koji smo videli za potpuno povezane mreže se može uopštiti za konvolutivne mreže jer se one mogu predstaviti usmerenim acikličnim grafovima



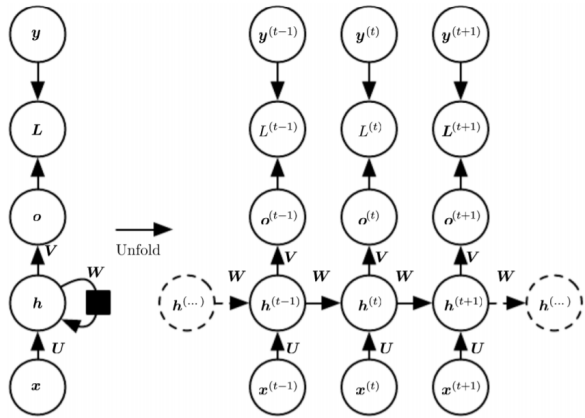
Graf izračunavanja

- Da bi se minimizovala funkcija greške, potrebno je izračunati njene gradijente
- Nad čvorom L pokrećemo algoritam propagacije unazad i izračunavamo gradijente u skladu sa ovim grafom i koristimo tako izračunate gradijente u standardnim optimizacionim metodama
- Kako se izračunava gradijent kada imamo rekurentne veze?
- Algoritam propagacije unazad koji smo videli za potpuno povezane mreže se može uopštiti za konvolutivne mreže jer se one mogu predstaviti usmerenim acikličnim grafovima
- Deluje da u ovom grafu imamo ciklus ali kvadrat na grani nam govori da to nije zapravo ciklus jer ne radimo sa istim h



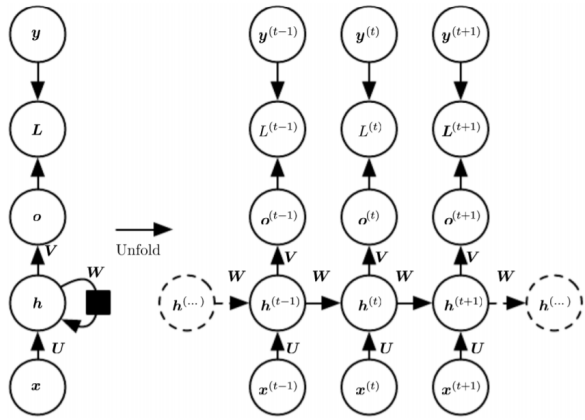
Graf izračunavanja

- Da bismo mogli da primenimo algoritam propagacije unazad, potrebno je da nad ovim grafom uradimo tzv. razmotavanje u odnosu na vreme



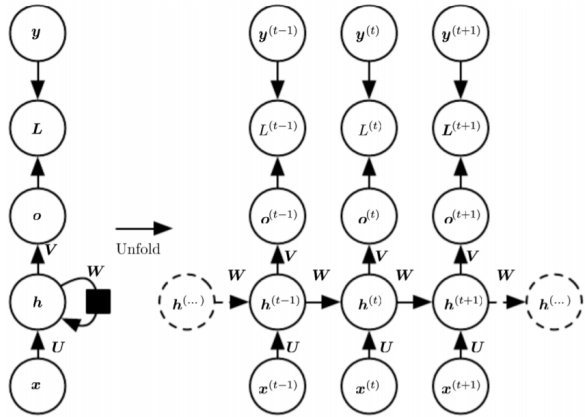
Graf izračunavanja

- Da bismo mogli da primenimo algoritam propagacije unazad, potrebno je da nad ovim grafom uradimo tzv. razmotavanje u odnosu na vreme
- Ako je rečenica dužine 3, imaćemo 3 ovakva stupca

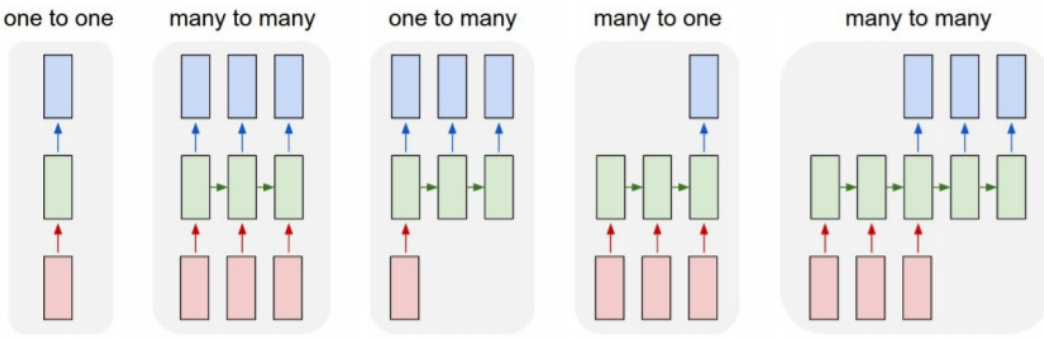


Graf izračunavanja

- Da bismo mogli da primenimo algoritam propagacije unazad, potrebno je da nad ovim grafom uradimo tzv. razmotavanje u odnosu na vreme
- Ako je rečenica dužine 3, imaćemo 3 ovakva stupca
- Za svaku rečenicu koju rekurentna mreža čita, napravimo jednu ovakvu strukturu, za svaku rečenicu odvojeno izračunamo gradijent, na kraju sve gradijente saberemo i dobijemo ukupni gradijent

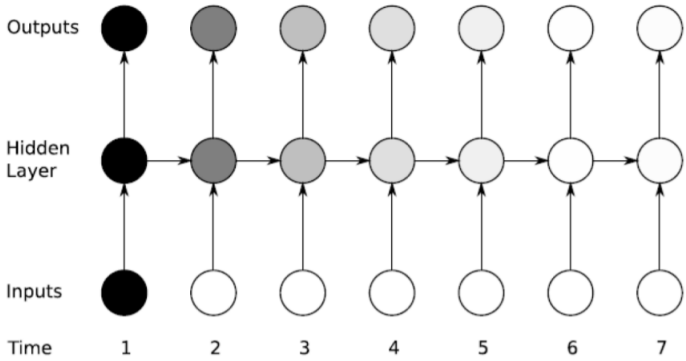


Varijacije arhitekture



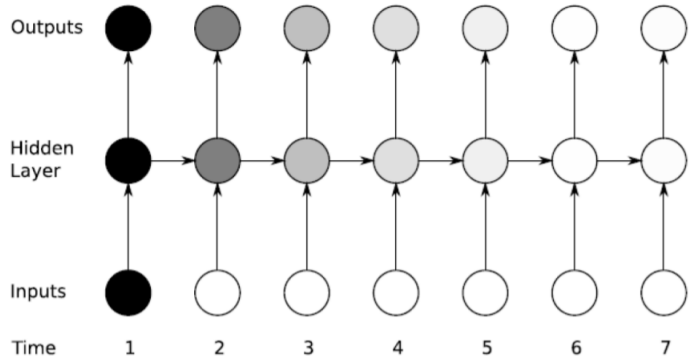
Duga kratkoročna memorija (LSTM)

- Predstavili smo osnovnu varijantu rekurentnih mreža koja se baš i ne koristi u praktičnim primenama



Duga kratkoročna memorija (LSTM)

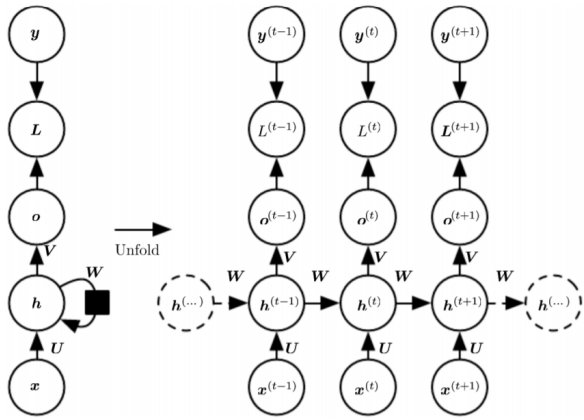
- Predstavili smo osnovnu varijantu rekurentnih mreža koja se baš i ne koristi u praktičnim primenama



- U praksi se pokazalo da ako imamo zavisnost između elemenata sekvence koji su na rastojanju dužem od 10 (npr. 100), obična rekurentna mreža to ne može da modeluje, tj. nema način da pamti dugo informaciju da bi sačekala 100 koraka do trenutka kada je ta informacija potrebna

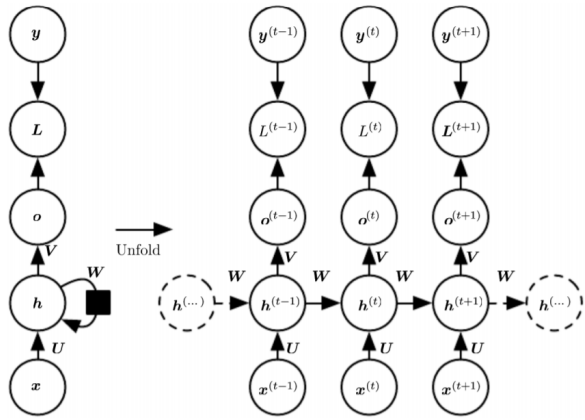
Duga kratkoročna memorija (LSTM)

- Graf izračunavanja razmotavamo u skladu sa dužinom sekvence



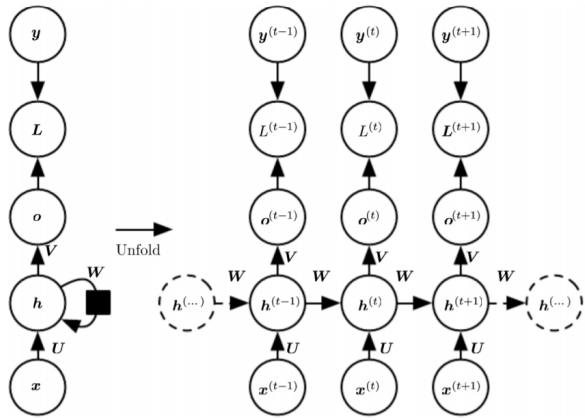
Duga kratkoročna memorija (LSTM)

- Graf izračunavanja razmotavamo u skladu sa dužinom sekvence
- Sekvence mogu biti duge nekoliko stotina ili hiljada elemenata



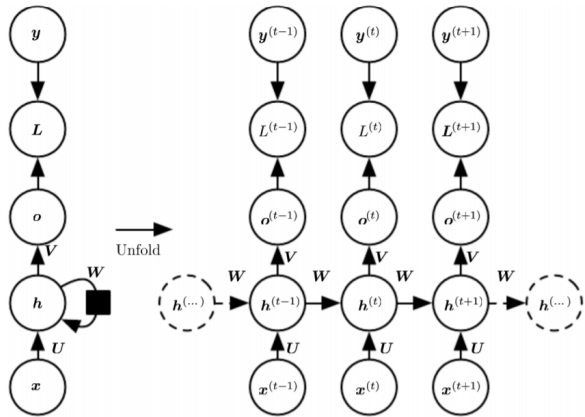
Duga kratkoročna memorija (LSTM)

- Graf izračunavanja razmotavamo u skladu sa dužinom sekvence
- Sekvence mogu biti duge nekoliko stotina ili hiljada elemenata
- Na taj način dobijamo veoma duboku mrežu



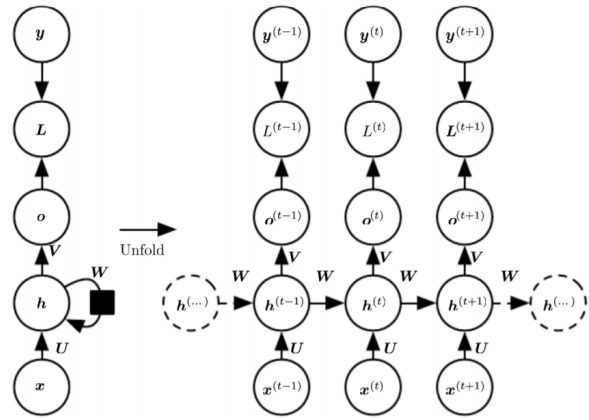
Duga kratkoročna memorija (LSTM)

- Graf izračunavanja razmotavamo u skladu sa dužinom sekvence
- Sekvence mogu biti duge nekoliko stotina ili hiljada elemenata
- Na taj način dobijamo veoma duboku mrežu
- Algoritam propagacije unazad računa izvod složene funkcije koje se zasniva na množenju izvoda, a takvih izvoda će biti onoliko koliko iznosi dužina sekvence



Duga kratkoročna memorija (LSTM)

- Graf izračunavanja razmotavamo u skladu sa dužinom sekvence
- Sekvence mogu biti duge nekoliko stotina ili hiljada elemenata
- Na taj način dobijamo veoma duboku mrežu
- Algoritam propagacije unazad računa izvod složene funkcije koje se zasniva na množenju izvoda, a takvih izvoda će biti onoliko koliko iznosi dužina sekvence
- To može dovesti ili do toga da takav proizvod ode u nulu (ako su činioci manji od 1) ili da eksplodira (ako su veći)



Duga kratkoročna memorija (LSTM)

- LSTM može omogućiti dugu memoriju

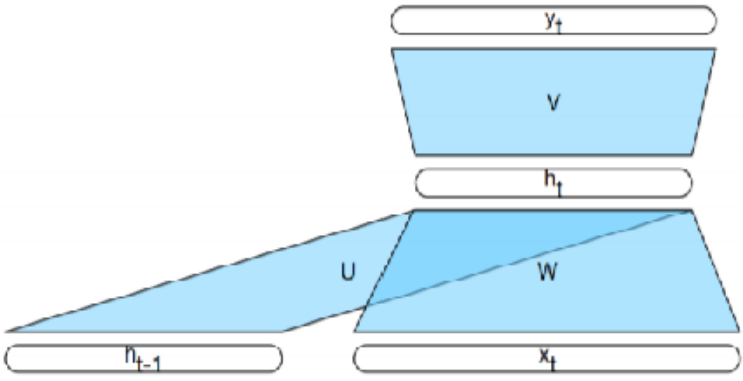
Duga kratkoročna memorija (LSTM)

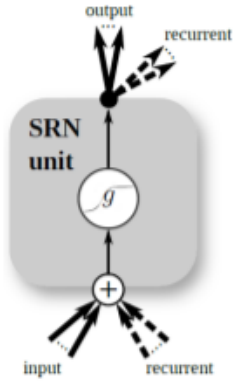
- LSTM može omogućiti dugu memoriju
- Ublažava problem nestajućih gradijenata

Duga kratkoročna memorija (LSTM)

- LSTM može omogućiti dugu memoriju
- Ublažava problem nestajućih gradijenata
- U praksi se vidi da rekurentne mreže i dalje imaju problema sa obučavanjem i zbog toga se u nekim primenama teži njihovom izbegavanju

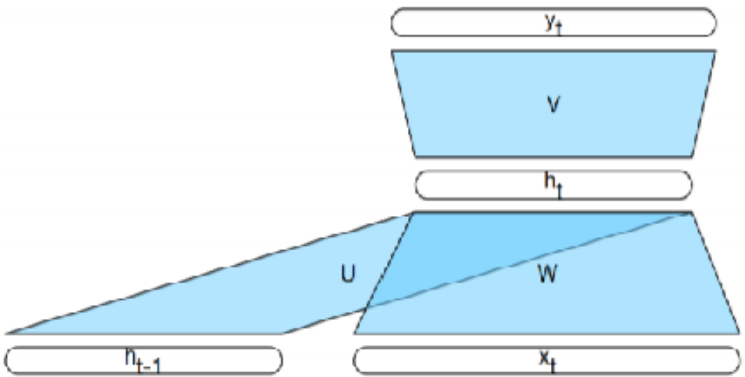
Duga kratkoročna memorija (LSTM)

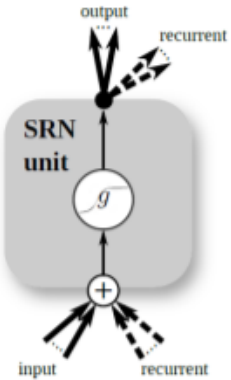




- kod standardne rekurentne jedinice, matrice U, W, V su fiksirane

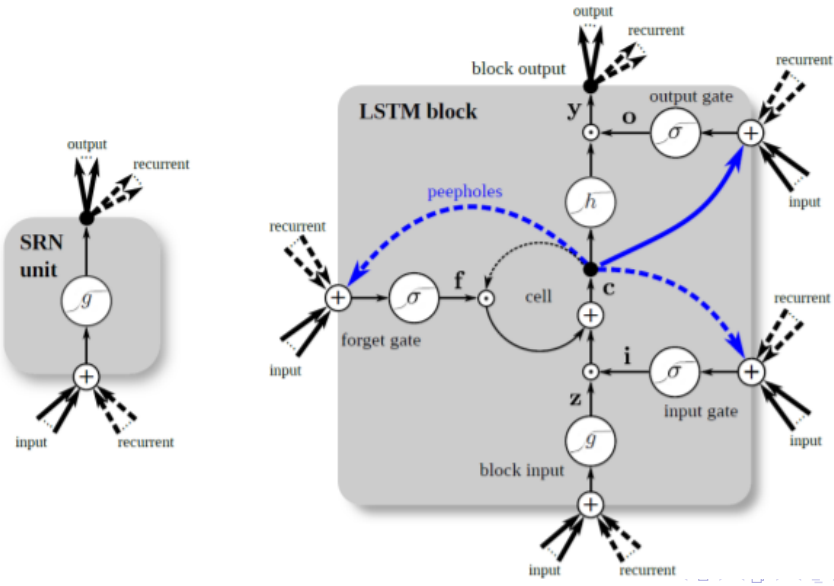
Duga kratkoročna memorija (LSTM)



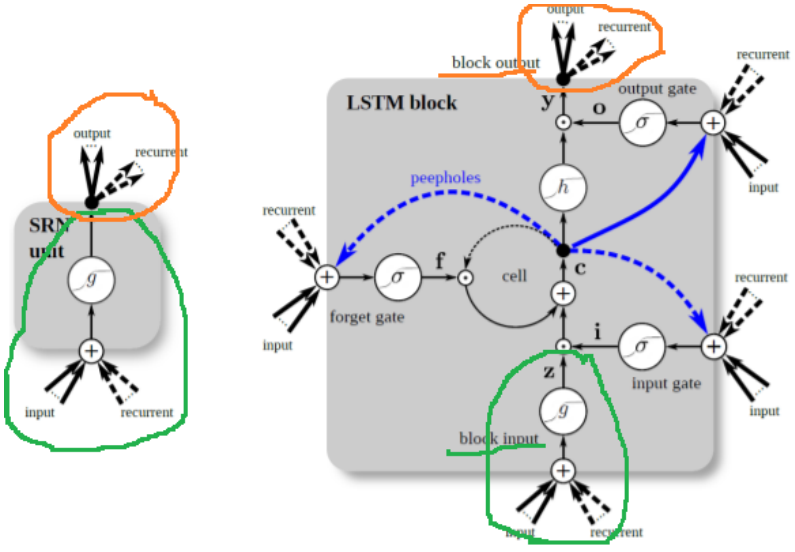


- kod standardne rekurentne jedinice, matrice U, W, V su fiksirane
- ideja za poboljšanje: umesto da su uvek iste, od njih hoćemo da napravimo *parametrizovane funkcije* koje će *učiti* svoje parametre tokom treninga

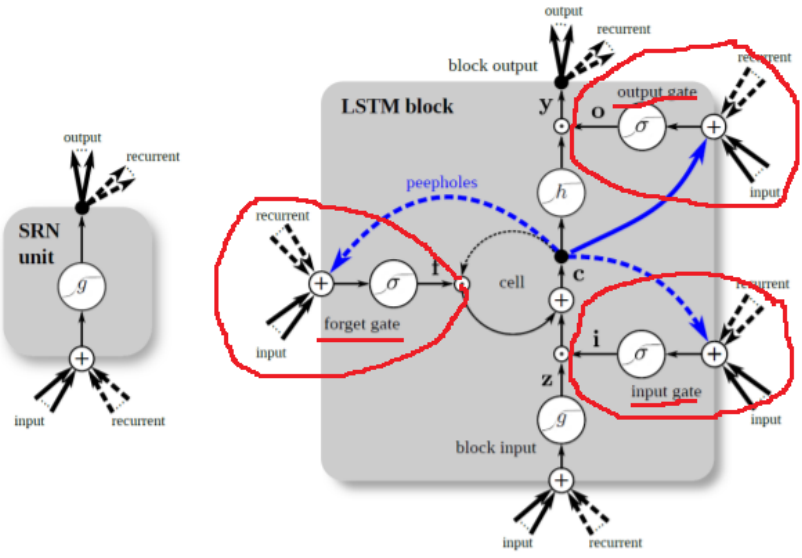
Duga kratkoročna memorija (LSTM)



Duga kratkoročna memorija (LSTM)

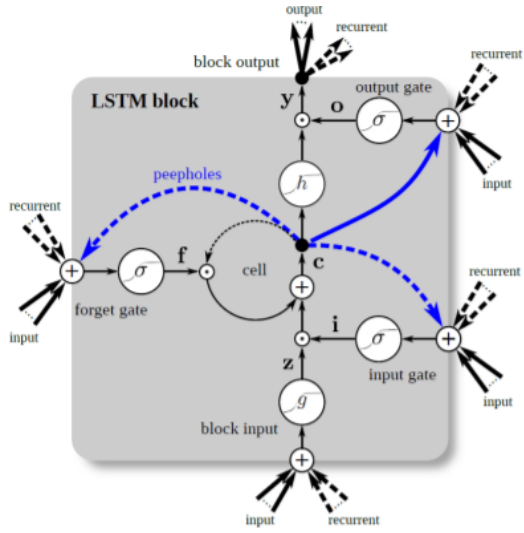


Duga kratkoročna memorija (LSTM)



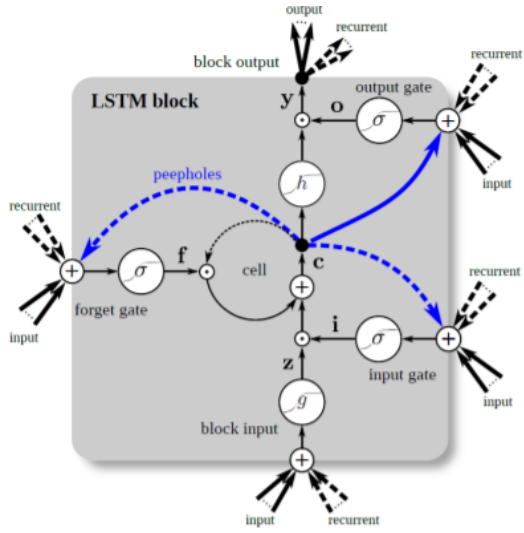
Duga kratkoročna memorija (LSTM)

- Osnovna ideja LSTM-a je postojanje takozvane *ćelije* (c) koja čuva skriveno stanje, uz kontrolu pisanja, čitanja i zaboravljanja, koja se vrši na osnovu naučenih pravila.



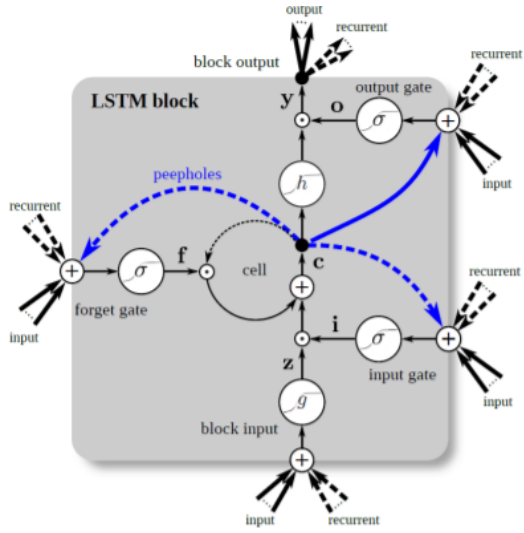
Duga kratkoročna memorija (LSTM)

- Osnovna ideja LSTM-a je postojanje takozvane *ćelije* (c) koja čuva skriveno stanje, uz kontrolu pisanja, čitanja i zaboravljanja, koja se vrši na osnovu naučenih pravila.
- Ovu kontrolu sprovode *kapije* (*input gate, output gate, forget gate*)



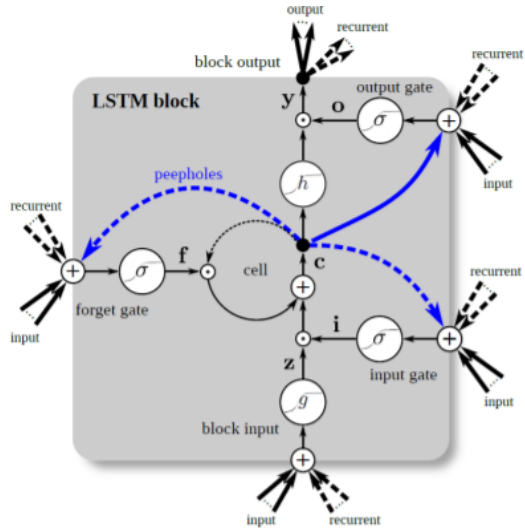
Duga kratkoročna memorija (LSTM)

- Osnovna ideja LSTM-a je postojanje takozvane *ćelije* (c) koja čuva skriveno stanje, uz kontrolu pisanja, čitanja i zaboravljanja, koja se vrši na osnovu naučenih pravila.
- Ovu kontrolu sprovode *kapije* (*input gate, output gate, forget gate*)
- Svaka od kapija ima svoj skup parametara



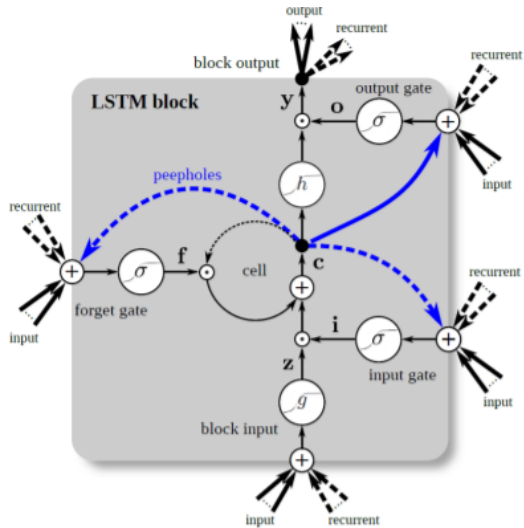
Duga kratkoročna memorija (LSTM)

- Osnovna ideja LSTM-a je postojanje takozvane *ćelije* (c) koja čuva skriveno stanje, uz kontrolu pisanja, čitanja i zaboravljanja, koja se vrši na osnovu naučenih pravila.
- Ovu kontrolu sprovode *kapije* (*input gate, output gate, forget gate*)
- Svaka od kapija ima svoj skup parametara
- Svaka od kapija ima istu strukturu kao jedinica standardne rekurentne mreže i na osnovu ulaza koje dobija i pridruženih im parametara odlučuje o tome da li i u kolikoj meri dozvoljava izvršavanje operacije koju kontroliše



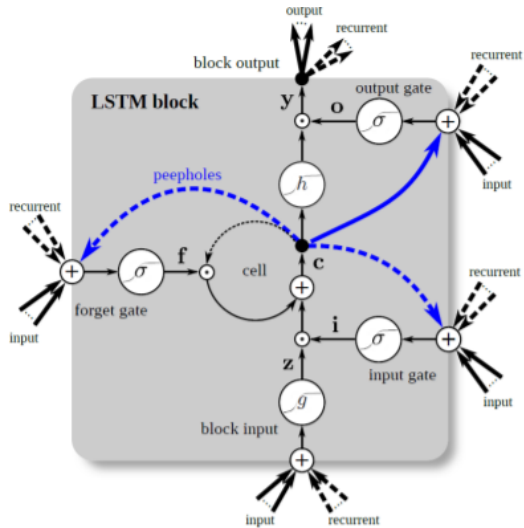
Duga kratkoročna memorija (LSTM)

- Na primer, ulazna kapija kontroliše da li će propustiti ulaz ka ćeliji i to radi tako što na osnovu ulaza (prva suma u definiciji ulazne aktivacije) i svog stanja u prethodnom koraku (druga suma) računa koeficijent (i) koji se koristi za kontrolu uticaja ulaza na stanje ćelije.



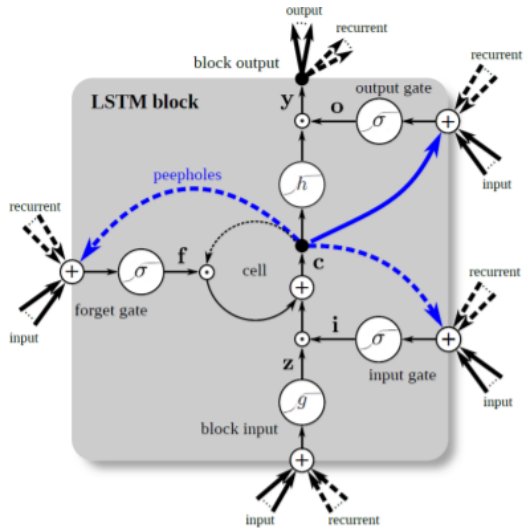
Duga kratkoročna memorija (LSTM)

- Na primer, ulazna kapija kontroliše da li će propustiti ulaz ka ćeliji i to radi tako što na osnovu ulaza (prva suma u definiciji ulazne aktivacije) i svog stanja u prethodnom koraku (druga suma) računa koeficijent (i) koji se koristi za kontrolu uticaja ulaza na stanje ćelije.
- Ona može da odluči da pusti ulaz (koeficijent blizu 1), da ne pusti ulaz (koeficijent blizu 0), ili da pusti sa nekim koeficijentom između



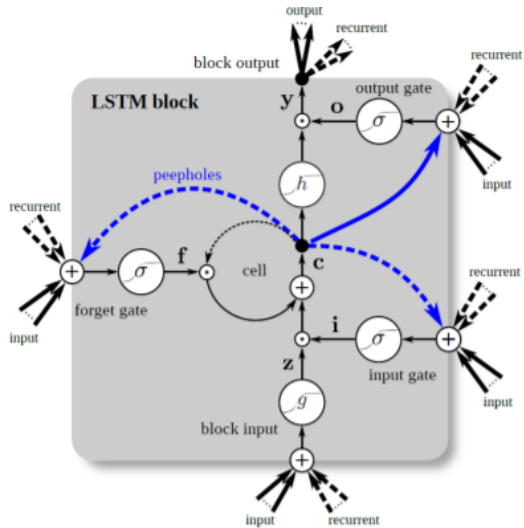
Duga kratkoročna memorija (LSTM)

- Na primer, ulazna kapija kontroliše da li će propustiti ulaz ka ćeliji i to radi tako što na osnovu ulaza (prva suma u definiciji ulazne aktivacije) i svog stanja u prethodnom koraku (druga suma) računa koeficijent (i) koji se koristi za kontrolu uticaja ulaza na stanje ćelije.
- Ona može da odluči da pusti ulaz (koeficijent blizu 1), da ne pusti ulaz (koeficijent blizu 0), ili da pusti sa nekim koeficijentom između
- Na taj način se kontroliše ulaz novih informacija u ćeliju



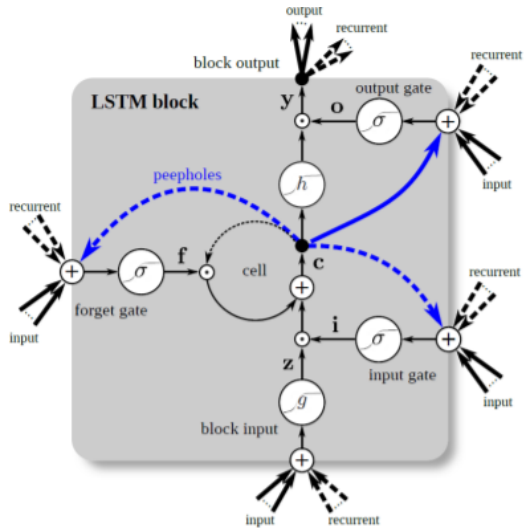
Duga kratkoročna memorija (LSTM)

- Na sličan način *forget* kapija kontroliše da li će prethodno stanje imati udela u računanju novog stanja preko koeficijenta (f)



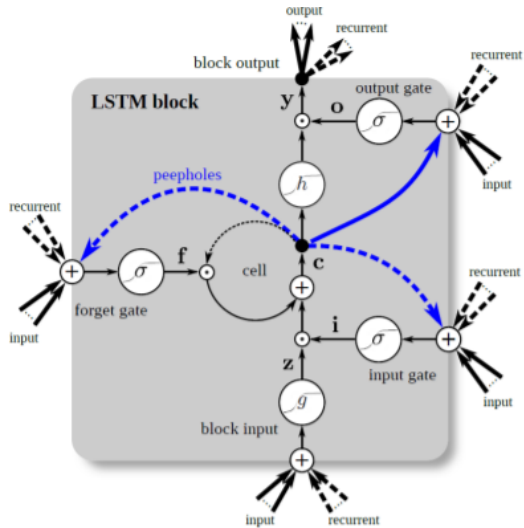
Duga kratkoročna memorija (LSTM)

- Na sličan način *forget* kapija kontroliše da li će prethodno stanje imati udela u računanju novog stanja preko koeficijenta (f)
- Ona može da odluči da pusti (zapamti) prethodno stanje (koeficijent blizu 1), da ne pusti (zaboravi) (koeficijent blizu 0), ili da pusti sa nekim koeficijentom između (delimično zapamti)



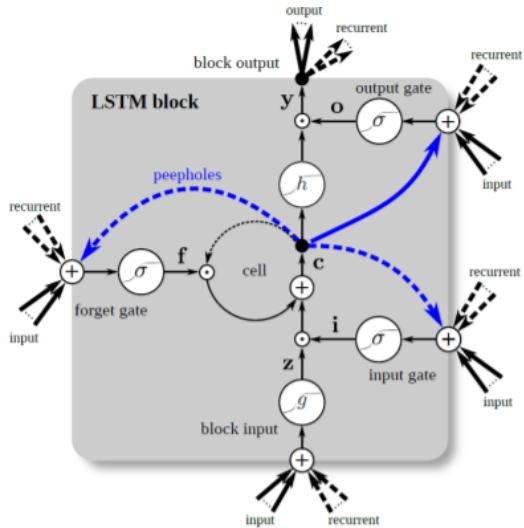
Duga kratkoročna memorija (LSTM)

- Kombinacija koeficijenata f i i utiče na računanje trenutnog stanja



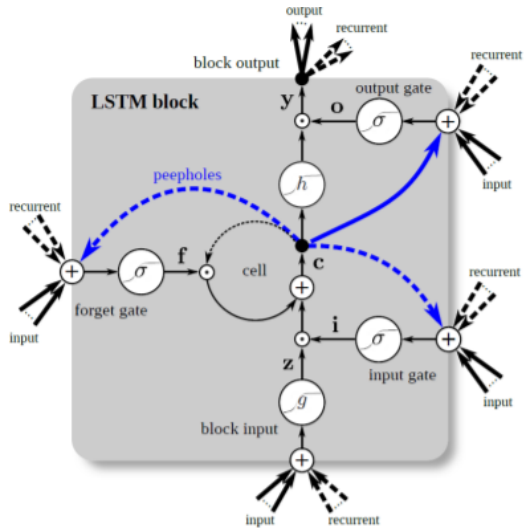
Duga kratkoročna memorija (LSTM)

- Kombinacija koeficijenata f i i utiče na računanje trenutnog stanja
  - ako želimo da zadržimo prethodno stanje, f će biti blizu 1, a i blizu 0



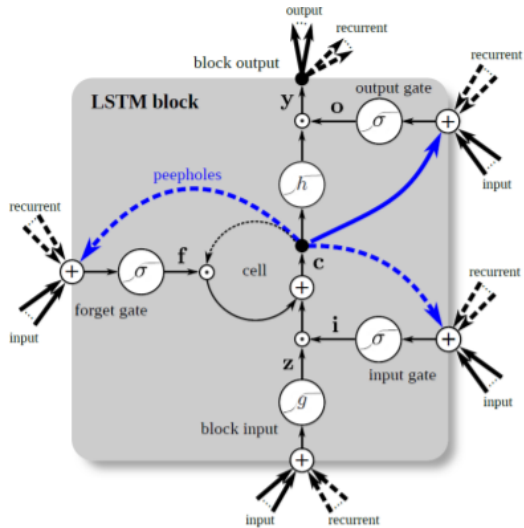
Duga kratkoročna memorija (LSTM)

- Kombinacija koeficijenata f i i utiče na računanje trenutnog stanja
  - ako želimo da zadržimo prethodno stanje, f će biti blizu 1, a i blizu 0
  - ako želimo da zaboravimo prethodno stanje i da novo stanje računamo samo na osnovu ulaza biće obrnuto



Duga kratkoročna memorija (LSTM)

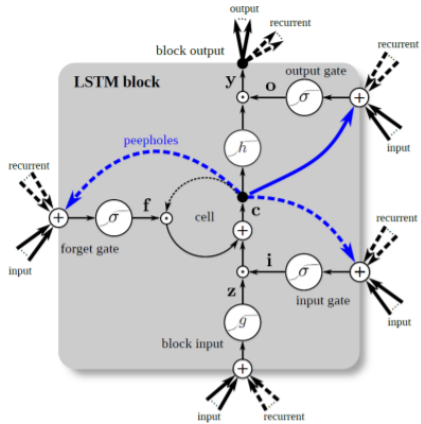
- Kombinacija koeficijenata f i i utiče na računanje trenutnog stanja
  - ako želimo da zadržimo prethodno stanje, f će biti blizu 1, a i blizu 0
  - ako želimo da zaboravimo prethodno stanje i da novo stanje računamo samo na osnovu ulaza biće obrnuto
  - koeficijenti se mogu postaviti i negde između, npr. 0.5 i 0.5



Duga kratkoročna memorija (LSTM)

$$
\begin{aligned}
c^{(0)} &= 0 \\
h^{(0)} &= 0 \\
z^{(t)} &= g(W_z x^{(t)} + U_z h^{(t-1)} + b_z) && \text{Transformisani ulaz} \\
i^{(t)} &= \sigma(W_i x^{(t)} + U_i h^{(t-1)} + b_i) && \text{Ulazna kapija} \\
f^{(t)} &= \sigma(W_f x^{(t)} + U_f h^{(t-1)} + b_f) && \text{Kapija zaboravljanja} \\
c^{(t)} &= z^{(t)} \odot i^{(t)} + c^{(t-1)} \odot f^{(t)} && \text{Ćelija} \\
q^{(t)} &= \sigma(W_q x^{(t)} + U_q h^{(t-1)} + b_q) && \text{Izlazna kapija} \\
h^{(t)} &= g(c^{(t)}) \odot q^{(t)} && \text{Izlaz}
\end{aligned}
$$

- q u formulama je o na slici
- obratimo pažnju da se vrednost novog stanja ($c^{(t)}$) dobija bez primene aktivacione funkcije



Duga kratkoročna memorija (LSTM)

- LSTM je značajan iz dva razloga.

Duga kratkoročna memorija (LSTM)

- LSTM je značajan iz dva razloga.
- Prvi razlog iz kojeg je LSTM značajan je ublažavanje problema nestajućih gradijenata.

Duga kratkoročna memorija (LSTM)

- LSTM je značajan iz dva razloga.
- Prvi razlog iz kojeg je LSTM značajan je ublažavanje problema nestajućih gradijenata.
- Naime, u slučaju obične rekurentne mreže, nova vrednost skrivenog sloja se dobija tako što se kao ulaz uzima izlaz tog sloja u prethodnom koraku, koji je transformisan aktivacionom funkcijom.

Duga kratkoročna memorija (LSTM)

- LSTM je značajan iz dva razloga.
- Prvi razlog iz kojeg je LSTM značajan je ublažavanje problema nestajućih gradijenata.
- Naime, u slučaju obične rekurentne mreže, nova vrednost skrivenog sloja se dobija tako što se kao ulaz uzima izlaz tog sloja u prethodnom koraku, koji je transformisan aktivacionom funkcijom.
- Nadovezivanje aktivacionih funkcija u računskom grafu vodi tome da se prilikom računanja izvoda, množe izvodi aktivacionih funkcija.

Duga kratkoročna memorija (LSTM)

- LSTM je značajan iz dva razloga.
- Prvi razlog iz kojeg je LSTM značajan je ublažavanje problema nestajućih gradijenata.
- Naime, u slučaju obične rekurentne mreže, nova vrednost skrivenog sloja se dobija tako što se kao ulaz uzima izlaz tog sloja u prethodnom koraku, koji je transformisan aktivacionom funkcijom.
- Nadovezivanje aktivacionih funkcija u računskom grafu vodi tome da se prilikom računanja izvoda, množe izvodi aktivacionih funkcija.
- U slučaju sigmoidne funkcije izvod može biti najviše 0.25, a u slučaju tangensa hiperboličkog je strogo manji od 1.

Duga kratkoročna memorija (LSTM)

- LSTM je značajan iz dva razloga.
- Prvi razlog iz kojeg je LSTM značajan je ublažavanje problema nestajućih gradijenata.
- Naime, u slučaju obične rekurentne mreže, nova vrednost skrivenog sloja se dobija tako što se kao ulaz uzima izlaz tog sloja u prethodnom koraku, koji je transformisan aktivacionom funkcijom.
- Nadovezivanje aktivacionih funkcija u računskom grafu vodi tome da se prilikom računanja izvoda, množe izvodi aktivacionih funkcija.
- U slučaju sigmoidne funkcije izvod može biti najviše 0.25, a u slučaju tangensa hiperboličkog je strogo manji od 1.
- Množenjem ovakvih brojeva gradijent nestaje.

Duga kratkoročna memorija (LSTM)

- S druge strane, prilikom računanja nove vrednosti ćelije, nema primene aktivacione funkcije.

Duga kratkoročna memorija (LSTM)

- S druge strane, prilikom računanja nove vrednosti ćelije, nema primene aktivacione funkcije.
- Naravno, prethodna vrednost se množi vrednošću kapije za zaboravljanje, koja uključuje sigmoidnu aktivacionu funkciju, pa se njen izvod mora pojaviti negde u računanju gradijenata, ali postojanje putanja u računskom grafu koje nisu zahvaćene ovim problemom ima osetan efekat.

Duga kratkoročna memorija (LSTM)

- S druge strane, prilikom računanja nove vrednosti ćelije, nema primene aktivacione funkcije.
- Naravno, prethodna vrednost se množi vrednošću kapije za zaboravljanje, koja uključuje sigmoidnu aktivacionu funkciju, pa se njen izvod mora pojaviti negde u računanju gradijenata, ali postojanje putanja u računskom grafu koje nisu zahvaćene ovim problemom ima osetan efekat.
- Na primer, standardna rekurentna mreža obično ne modeluje dobro zavisnosti na rastojanju većem od desetak koraka.

Duga kratkoročna memorija (LSTM)

- S druge strane, prilikom računanja nove vrednosti ćelije, nema primene aktivacione funkcije.
- Naravno, prethodna vrednost se množi vrednošću kapije za zaboravljanje, koja uključuje sigmoidnu aktivacionu funkciju, pa se njen izvod mora pojaviti negde u računanju gradijenata, ali postojanje putanja u računskom grafu koje nisu zahvaćene ovim problemom ima osetan efekat.
- Na primer, standardna rekurentna mreža obično ne modeluje dobro zavisnosti na rastojanju većem od desetak koraka.
- S druge strane, rekurentnim mrežama sa LSTM jedinicom se uspešno modeluju zavisnosti i na rastojanju od više stotina koraka.

Duga kratkoročna memorija (LSTM)

- Drugo, zahvaljujući kapijama koje mogu da kontrolišu ulaz u ćeliju, ćelija ne mora da prihvata ulazne signale i stoga može dugo da čuva informaciju o dalekim delovima sekvence.

Duga kratkoročna memorija (LSTM)

- Drugo, zahvaljujući kapijama koje mogu da kontrolišu ulaz u ćeliju, ćelija ne mora da prihvata ulazne signale i stoga može dugo da čuva informaciju o dalekim delovima sekvence.
- Na primer, kapije mogu da nauče da kontrolišu ulaz u ćeliju tako što će neke koordinate stanja čuvati informacije iz starih koraka (dugo će čuvati neke informacije) a na nekim koordinatama će se možda brže ažurirati informacije

Duga kratkoročna memorija (LSTM)

- Drugo, zahvaljujući kapijama koje mogu da kontrolišu ulaz u ćeliju, ćelija ne mora da prihvata ulazne signale i stoga može dugo da čuva informaciju o dalekim delovima sekvence.
- Na primer, kapije mogu da nauče da kontrolišu ulaz u ćeliju tako što će neke koordinate stanja čuvati informacije iz starih koraka (dugo će čuvati neke informacije) a na nekim koordinatama će se možda brže ažurirati informacije
- Da li treba da prima ulazne signale ili ne, je nešto što se uči zahvaljujući tome što je ulazna kapija parametrizovana.

Duga kratkoročna memorija (LSTM)

- Drugo, zahvaljujući kapijama koje mogu da kontrolišu ulaz u ćeliju, ćelija ne mora da prihvata ulazne signale i stoga može dugo da čuva informaciju o dalekim delovima sekvence.
- Na primer, kapije mogu da nauče da kontrolišu ulaz u ćeliju tako što će neke koordinate stanja čuvati informacije iz starih koraka (dugo će čuvati neke informacije) a na nekim koordinatama će se možda brže ažurirati informacije
- Da li treba da prima ulazne signale ili ne, je nešto što se uči zahvaljujući tome što je ulazna kapija parametrizovana.
- Treba imati u vidu da se u praksi paralelno koristi veći broj ovakvih jedinica, tako da dok neke čuvaju informaciju iz ranijih delova sekvence, druge mogu da obrađuju tekući ulaz, da ga kombinuju sa informacijom iz onih prvih i slično.

Kvaliteti i mane

- Kada se pominju rekurentne mreže, u ogromnom broju slučajeva se podrazumeva LSTM

Kvaliteti i mane

- Kada se pominju rekurentne mreže, u ogromnom broju slučajeva se podrazumeva LSTM
- Kvaliteti

Kvaliteti i mane

- Kada se pominju rekurentne mreže, u ogromnom broju slučajeva se podrazumeva LSTM
- Kvaliteti
  - možemo da obrađujemo sekvencijalne podatke

Kvaliteti i mane

- Kada se pominju rekurentne mreže, u ogromnom broju slučajeva se podrazumeva LSTM
- Kvaliteti
  - možemo da obrađujemo sekvencijalne podatke
  - možemo da generišemo sekvencijalne izlaze

Kvaliteti i mane

- Kada se pominju rekurentne mreže, u ogromnom broju slučajeva se podrazumeva LSTM
- Kvaliteti
  - možemo da obrađujemo sekvencijalne podatke
  - možemo da generišemo sekvencijalne izlaze
  - možemo da imamo dugoročno čuvanje informacija i da modelujemo zavisnosti koje su na velikoj udaljenosti u okviru sekvence

Kvaliteti i mane

- Kada se pominju rekurentne mreže, u ogromnom broju slučajeva se podrazumeva LSTM
- Kvaliteti
  - možemo da obrađujemo sekvencijalne podatke
  - možemo da generišemo sekvencijalne izlaze
  - možemo da imamo dugoročno čuvanje informacija i da modelujemo zavisnosti koje su na velikoj udaljenosti u okviru sekvence
- Mane

Kvaliteti i mane

- Kada se pominju rekurentne mreže, u ogromnom broju slučajeva se podrazumeva LSTM
- Kvaliteti
  - možemo da obrađujemo sekvencijalne podatke
  - možemo da generišemo sekvencijalne izlaze
  - možemo da imamo dugoročno čuvanje informacija i da modelujemo zavisnosti koje su na velikoj udaljenosti u okviru sekvence
- Mane
  - postoje problemi u obučavanju vezani za gradijent

Kvaliteti i mane

- Kada se pominju rekurentne mreže, u ogromnom broju slučajeva se podrazumeva LSTM
- Kvaliteti
  - možemo da obrađujemo sekvencijalne podatke
  - možemo da generišemo sekvencijalne izlaze
  - možemo da imamo dugoročno čuvanje informacija i da modelujemo zavisnosti koje su na velikoj udaljenosti u okviru sekvence
- Mane
  - postoje problemi u obučavanju vezani za gradijent
  - od svih mreža koje smo do sada videli, rekurentne mreže se najteže paralelizuju zbog svoje suštinski sekvencijalne prirode

Pregled

Potpuno povezane neuronske mreže

Konvolutivne neuronske mreže

Rekurentne neuronske mreže

Praktične tehnike i napredni koncepti

Praktične tehnike i napredni koncepti

- Predstavili smo osnovne modele neuronskih mreža

Praktične tehnike i napredni koncepti

- Predstavili smo osnovne modele neuronskih mreža
- Da bi se one uspešno trenirale u praksi, potrebno je razmotriti još neke detalje njihove upotrebe

Inicijalizacija parametara

- Kad radimo logističku regresiju, možemo da uzmemo bilo koju vrednost kao početnu za parametre w, pošto je funkcija konveksna, svejedno je šta ćemo odabrati, globalni optimum će u svakom slučaju biti pronađen

Inicijalizacija parametara

- Kad radimo logističku regresiju, možemo da uzmemo bilo koju vrednost kao početnu za parametre w, pošto je funkcija konveksna, svejedno je šta ćemo odabrati, globalni optimum će u svakom slučaju biti pronađen
- Kod neuronskih mreža, gde je funkcija greške komplikovana, nekonveksna, sa nekakvim lokalnim minimumima, to nije slučaj

Inicijalizacija parametara

- Kad radimo logističku regresiju, možemo da uzmemo bilo koju vrednost kao početnu za parametre w, pošto je funkcija konveksna, svejedno je šta ćemo odabrati, globalni optimum će u svakom slučaju biti pronađen
- Kod neuronskih mreža, gde je funkcija greške komplikovana, nekonveksna, sa nekakvim lokalnim minimumima, to nije slučaj
- Pokazalo se da pitanje inicijalizacije parametara nije naivno i da nije svejedno od koje početne vrednosti gradijentni spust kreće sa optimizacijom

Inicijalizacija parametara

- Mogli bismo da naivno pretpostavimo da je najbolje sve koeficijente postaviti na 0

Inicijalizacija parametara

- Mogli bismo da naivno pretpostavimo da je najbolje sve koeficijente postaviti na 0
- Ova pretpostavka je loša, ne zbog vrednosti 0 već zbog toga što je odabrana ista vrednost za različite jedinice

Inicijalizacija parametara

- Mogli bismo da naivno pretpostavimo da je najbolje sve koeficijente postaviti na 0
- Ova pretpostavka je loša, ne zbog vrednosti 0 već zbog toga što je odabrana ista vrednost za različite jedinice
- U tom slučaju će se na početku sve jedinice ponašati identično i mreži će biti potrebno vremena da ih specijalizuje za uočavanje različitih atributa

Inicijalizacija parametara

- Mogli bismo da naivno pretpostavimo da je najbolje sve koeficijente postaviti na 0
- Ova pretpostavka je loša, ne zbog vrednosti 0 već zbog toga što je odabrana ista vrednost za različite jedinice
- U tom slučaju će se na početku sve jedinice ponašati identično i mreži će biti potrebno vremena da ih specijalizuje za uočavanje različitih atributa
- Stoga je bolja strategija da se inicijalizuju na različite vrednosti

Inicijalizacija parametara

- Mogli bismo da naivno pretpostavimo da je najbolje sve koeficijente postaviti na 0
- Ova pretpostavka je loša, ne zbog vrednosti 0 već zbog toga što je odabrana ista vrednost za različite jedinice
- U tom slučaju će se na početku sve jedinice ponašati identično i mreži će biti potrebno vremena da ih specijalizuje za uočavanje različitih atributa
- Stoga je bolja strategija da se inicijalizuju na različite vrednosti
- Najčešće se koristi uniformna ili normalna inicijalizacija, npr. $U\left[-\frac{1}{\sqrt{n}}, \frac{1}{\sqrt{n}}\right]$

Unutrašnja standardizacija

- Batch normalization

Unutrašnja standardizacija

- Batch normalization
- Intenzivno se koristi i predstavlja standard

Unutrašnja standardizacija

- Batch normalization
- Intenzivno se koristi i predstavlja standard
- Kada primenjujemo metode mašinskog učenja poželjno je ulaz nekako standardizovan

Unutrašnja standardizacija

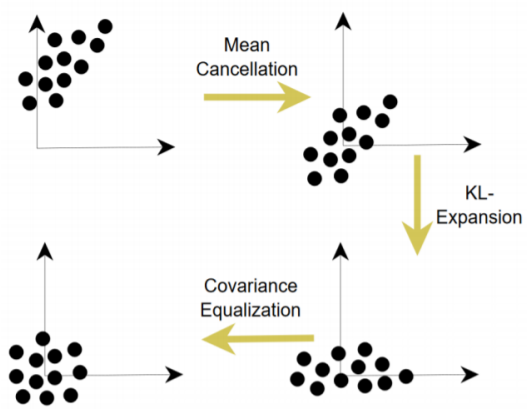
- Batch normalization
- Intenzivno se koristi i predstavlja standard
- Kada primenjujemo metode mašinskog učenja poželjno je ulaz nekako standardizovan
- Pored toga, još je pogodnije ako su dodatno svi atributi dekorelisani

Unutrašnja standardizacija

- Batch normalization
- Intenzivno se koristi i predstavlja standard
- Kada primenjujemo metode mašinskog učenja poželjno je ulaz nekako standardizovan
- Pored toga, još je pogodnije ako su dodatno svi atributi dekorelisani
- U suprotnom se očekuje problem izduženih kontura funkcije i cik-cak kretanje gradijentnih metoda

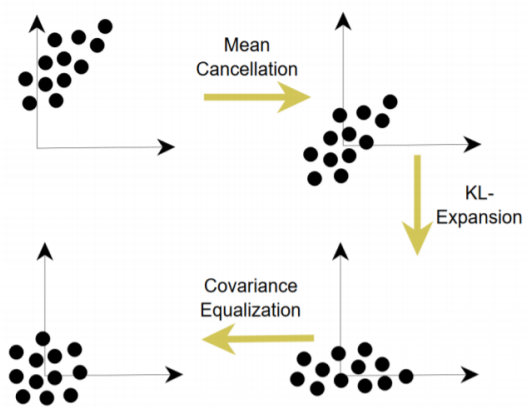
Unutrašnja standardizacija

- Primer podataka sa dva atributa koji su korelisani



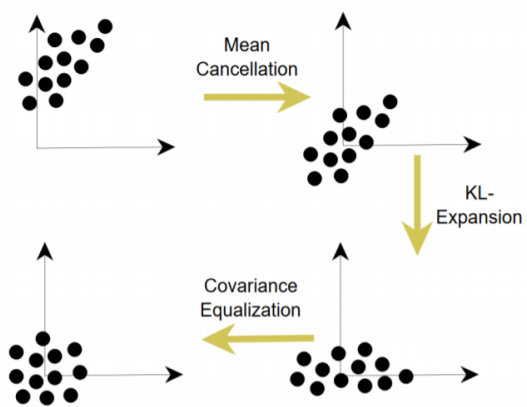
Unutrašnja standardizacija

- Primer podataka sa dva atributa koji su korelisani
- Bilo bi najbolje da podatke možemo da centriramo, zarotiramo tako da su paralelni sa koordinatnim osama i još da ih skaliramo



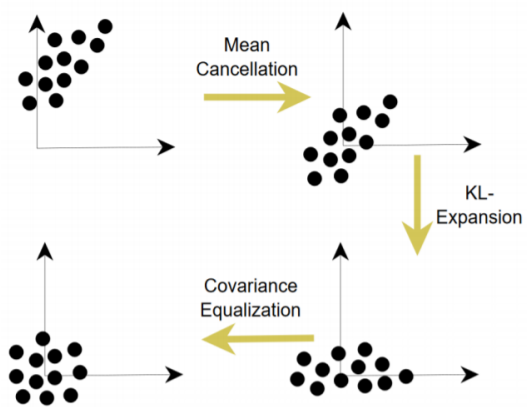
Unutrašnja standardizacija

- Primer podataka sa dva atributa koji su korelisani
- Bilo bi najbolje da podatke možemo da centriramo, zarotiramo tako da su paralelni sa koordinatnim osama i još da ih skaliramo
- Ova procedura je skupa i stoga se ne radi



Unutrašnja standardizacija

- Primer podataka sa dva atributa koji su korelisani
- Bilo bi najbolje da podatke možemo da centriramo, zarotiramo tako da su paralelni sa koordinatnim osama i još da ih skaliramo
- Ova procedura je skupa i stoga se ne radi
- Zbog toga, ulazi se obično samo standardizuju



Unutrašnja standardizacija

- Kada npr. radimo standardizaciju za logističku regresiju, standardizujemo ulaze i imamo samo jedan izlaz koji se na osnovu toga računa

Unutrašnja standardizacija

- Kada npr. radimo standardizaciju za logističku regresiju, standardizujemo ulaze i imamo samo jedan izlaz koji se na osnovu toga računa
- Kada imamo neuronsku mrežu, možemo je posmatrati kao niz logističkih regresija i neuroni na ulaznom nivou će dobiti standardizovane ulaze, ali već sledeći sloj neće jer je transformacija koju prvi sloj radi takva da ne poštuje standardizaciju

Unutrašnja standardizacija

- Kada npr. radimo standardizaciju za logističku regresiju, standardizujemo ulaze i imamo samo jedan izlaz koji se na osnovu toga računa
- Kada imamo neuronsku mrežu, možemo je posmatrati kao niz logističkih regresija i neuroni na ulaznom nivou će dobiti standardizovane ulaze, ali već sledeći sloj neće jer je transformacija koju prvi sloj radi takva da ne poštuje standardizaciju
- Standardizacija se ne održava na višim nivoima

Unutrašnja standardizacija

- Kada npr. radimo standardizaciju za logističku regresiju, standardizujemo ulaze i imamo samo jedan izlaz koji se na osnovu toga računa
- Kada imamo neuronsku mrežu, možemo je posmatrati kao niz logističkih regresija i neuroni na ulaznom nivou će dobiti standardizovane ulaze, ali već sledeći sloj neće jer je transformacija koju prvi sloj radi takva da ne poštuje standardizaciju
- Standardizacija se ne održava na višim nivoima
- Ideja: uraditi standardizaciju i unutar mreže, ne samo na ulaznom sloju

Unutrašnja standardizacija

- Kada npr. radimo standardizaciju za logističku regresiju, standardizujemo ulaze i imamo samo jedan izlaz koji se na osnovu toga računa
- Kada imamo neuronsku mrežu, možemo je posmatrati kao niz logističkih regresija i neuroni na ulaznom nivou će dobiti standardizovane ulaze, ali već sledeći sloj neće jer je transformacija koju prvi sloj radi takva da ne poštuje standardizaciju
- Standardizacija se ne održava na višim nivoima
- Ideja: uraditi standardizaciju i unutar mreže, ne samo na ulaznom sloju
- Standardizujemo podatke za ulazni sloj a onda, nakon što prođu prvi sloj neurona, ne prosleđuju se direktno u drugi sloj veše prvo izvrši standardizacija (izračuna se prosek svih vrednosti i standardna devijacija i od svake vrednosti se oduzme prosek a potom podeli sa standardnom devijacijom)

Unutrašnja standardizacija

- Ovako opisanu standardizaciju nećemo raditi nad celim skupom podataka već na nekom njegovom podskupu (*batch*)

Unutrašnja standardizacija

- Ovako opisanu standardizaciju nećemo raditi nad celim skupom podataka već na nekom njegovom podskupu (*batch*)
- Ako posmatramo jedinicu u i podskup instanci $\mathcal{B}$, biće izračunati prosek $\mu_{\mathcal{B}}^{u}$ i standardna devijacija $\sigma_{\mathcal{B}}^{u}$ na nivou tog podskupa

Unutrašnja standardizacija

- Ovako opisanu standardizaciju nećemo raditi nad celim skupom podataka već na nekom njegovom podskupu (*batch*)
- Ako posmatramo jedinicu u i podskup instanci $\mathcal{B}$, biće izračunati prosek $\mu_{\mathcal{B}}^{u}$ i standardna devijacija $\sigma_{\mathcal{B}}^{u}$ na nivou tog podskupa
- Ove vrednosti će biti korišćene za standardizaciju izlaza iz jednog sloja neurona pre nego što ove vrednosti budu prosleđene sledećem sloju

Unutrašnja standardizacija

- Ovako opisanu standardizaciju nećemo raditi nad celim skupom podataka već na nekom njegovom podskupu (*batch*)
- Ako posmatramo jedinicu u i podskup instanci $\mathcal{B}$, biće izračunati prosek $\mu_{\mathcal{B}}^{u}$ i standardna devijacija $\sigma_{\mathcal{B}}^{u}$ na nivou tog podskupa
- Ove vrednosti će biti korišćene za standardizaciju izlaza iz jednog sloja neurona pre nego što ove vrednosti budu prosleđene sledećem sloju
- To je i dalje sve diferencijabilno i propagacija unazad može da izračuna gradijente i možemo da radimo gradijentni spust

Unutrašnja standardizacija

- Problem sa ovakvom standardizacijom u praksi je u tome što ograničavanjem vrednosti na ovaj način previše ograničavamo mrežu i smanjuje njenu izražajnu moć

Unutrašnja standardizacija

- Problem sa ovakvom standardizacijom u praksi je u tome što ograničavanjem vrednosti na ovaj način previše ograničavamo mrežu i smanjuje njenu izražajnu moć
- Zato je predloženo sledeće rešenje: ako je $\hat{h}^u$ standardizovana vrednost izlaza jedinice u, umesto te vrednosti, koristi se vrednost $\alpha^u \hat{h}^u + \beta^u$, gde su α^u i β^u parametri koji se uče kao i ostali parametri mreže.

Unutrašnja standardizacija

- Problem sa ovakvom standardizacijom u praksi je u tome što ograničavanjem vrednosti na ovaj način previše ograničavamo mrežu i smanjuje njenu izražajnu moć
- Zato je predloženo sledeće rešenje: ako je $\hat{h}^u$ standardizovana vrednost izlaza jedinice u, umesto te vrednosti, koristi se vrednost $\alpha^u \hat{h}^u + \beta^u$, gde su α^u i β^u parametri koji se uče kao i ostali parametri mreže.
- Ne deluje da ovakva transformacija ima smisla - upravo smo transformisali izlaznu vrednost tako što smo od nje oduzeli prosek od izlazne vrednosti i podelili je standardnom devijacijom, a sada na tu transformisanu vrednost primenjujemo inverznu transformaciju

Unutrašnja standardizacija

- Problem sa ovakvom standardizacijom u praksi je u tome što ograničavanjem vrednosti na ovaj način previše ograničavamo mrežu i smanjuje njenu izražajnu moć
- Zato je predloženo sledeće rešenje: ako je $\hat{h}^u$ standardizovana vrednost izlaza jedinice u, umesto te vrednosti, koristi se vrednost $\alpha^u \hat{h}^u + \beta^u$, gde su α^u i β^u parametri koji se uče kao i ostali parametri mreže.
- Ne deluje da ovakva transformacija ima smisla - upravo smo transformisali izlaznu vrednost tako što smo od nje oduzeli prosek od izlazne vrednosti i podelili je standardnom devijacijom, a sada na tu transformisanu vrednost primenjujemo inverznu transformaciju
- Pokazalo se da ovo rešenje u praksi dovodi do drastično brže konvergencije (čak 10 puta brže) i da čini regularizaciju nepotrebnom jer i sama predstavlja vid regularizacije zasnovan na unošenju šuma

Unutrašnja standardizacija

- Problem sa ovakvom standardizacijom u praksi je u tome što ograničavanjem vrednosti na ovaj način previše ograničavamo mrežu i smanjuje njenu izražajnu moć
- Zato je predloženo sledeće rešenje: ako je $\hat{h}^u$ standardizovana vrednost izlaza jedinice u, umesto te vrednosti, koristi se vrednost $\alpha^u \hat{h}^u + \beta^u$, gde su α^u i β^u parametri koji se uče kao i ostali parametri mreže.
- Ne deluje da ovakva transformacija ima smisla - upravo smo transformisali izlaznu vrednost tako što smo od nje oduzeli prosek od izlazne vrednosti i podelili je standardnom devijacijom, a sada na tu transformisanu vrednost primenjujemo inverznu transformaciju
- Pokazalo se da ovo rešenje u praksi dovodi do drastično brže konvergencije (čak 10 puta brže) i da čini regularizaciju nepotrebnom jer i sama predstavlja vid regularizacije zasnovan na unošenju šuma
- Ali kako ovo radi?

Unutrašnja standardizacija

- Kad nemamo unutrašnju standardizaciju, raspodela izlaza neurona negde na visokim slojevima zavisi od svih parametara koji učestvuju u generisanju njegovih ulaza

Unutrašnja standardizacija

- Kad nemamo unutrašnju standardizaciju, raspodela izlaza neurona negde na visokim slojevima zavisi od svih parametara koji učestvuju u generisanju njegovih ulaza
- Ova raspodela je posledica složenih interakcija koje se dešavaju na putanji od ulaza do tog neurona

Unutrašnja standardizacija

- Kad nemamo unutrašnju standardizaciju, raspodela izlaza neurona negde na visokim slojevima zavisi od svih parametara koji učestvuju u generisanju njegovih ulaza
- Ova raspodela je posledica složenih interakcija koje se dešavaju na putanji od ulaza do tog neurona
- U gradijentnom spustu sve vreme menjamo težine i na svakom nivou dolazi do promena u raspodeli koje sve zavise od promena duboko do početka mreže

Unutrašnja standardizacija

- Kad nemamo unutrašnju standardizaciju, raspodela izlaza neurona negde na visokim slojevima zavisi od svih parametara koji učestvuju u generisanju njegovih ulaza
- Ova raspodela je posledica složenih interakcija koje se dešavaju na putanji od ulaza do tog neurona
- U gradijentnom spustu sve vreme menjamo težine i na svakom nivou dolazi do promena u raspodeli koje sve zavise od promena duboko do početka mreže
- Promena raspodele svakog nivoa je vrlo složena i zavisi od toga kako se menjaju parametri na prethodnim slojevima

Unutrašnja standardizacija

- Kad nemamo unutrašnju standardizaciju, raspodela izlaza neurona negde na visokim slojevima zavisi od svih parametara koji učestvuju u generisanju njegovih ulaza
- Ova raspodela je posledica složenih interakcija koje se dešavaju na putanji od ulaza do tog neurona
- U gradijentnom spustu sve vreme menjamo težine i na svakom nivou dolazi do promena u raspodeli koje sve zavise od promena duboko do početka mreže
- Promena raspodele svakog nivoa je vrlo složena i zavisi od toga kako se menjaju parametri na prethodnim slojevima
- Sa druge strane, u ovom slučaju uspevamo da to razvežemo, ne prenosimo kompleksne interakcije raspodela sa prethodnog nivoa na naredne nivoe i naš prosek i standardna devijacija zavise od parametara α i β a ne od toga sta se desavalo pre

Unutrašnja standardizacija

- Objasnili smo kako se vrši treniranje ali šta se dešava sa predviđanjem?

Unutrašnja standardizacija

- Objasnili smo kako se vrši treniranje ali šta se dešava sa predviđanjem?
- Kada hoćemo da vršimo predviđanje, imamo jednu instancu

Unutrašnja standardizacija

- Objasnili smo kako se vrši treniranje ali šta se dešava sa predviđanjem?
- Kada hoćemo da vršimo predviđanje, imamo jednu instancu
- Kada jednu instancu propustimo kroz mrežu, ne možemo da za nju izračunamo standardnu devijaciju (prosek možemo, to je taj broj)

Unutrašnja standardizacija

- Objasnili smo kako se vrši treniranje ali šta se dešava sa predviđanjem?
- Kada hoćemo da vršimo predviđanje, imamo jednu instancu
- Kada jednu instancu propustimo kroz mrežu, ne možemo da za nju izračunamo standardnu devijaciju (prosek možemo, to je taj broj)
- Onda se nađemo u situaciji da kada vršimo predviđanje za jednu instancu, ne možemo da ispravno uradimo standardnu devijaciju

Unutrašnja standardizacija

- Objasnili smo kako se vrši treniranje ali šta se dešava sa predviđanjem?
- Kada hoćemo da vršimo predviđanje, imamo jednu instancu
- Kada jednu instancu propustimo kroz mrežu, ne možemo da za nju izračunamo standardnu devijaciju (prosek možemo, to je taj broj)
- Onda se nađemo u situaciji da kada vršimo predviđanje za jednu instancu, ne možemo da ispravno uradimo standardnu devijaciju
- Zbog toga se za vreme treniranja parametri μ i σ za svaki neuron pamte i u vreme predviđanja se koristi procena ovih parametara na osnovu vrednosti izračunatih tokom treniranja najčešće pomoću tzv. *pokretnog proseka*

Unutrašnja standardizacija

- Objasnili smo kako se vrši treniranje ali šta se dešava sa predviđanjem?
- Kada hoćemo da vršimo predviđanje, imamo jednu instancu
- Kada jednu instancu propustimo kroz mrežu, ne možemo da za nju izračunamo standardnu devijaciju (prosek možemo, to je taj broj)
- Onda se nađemo u situaciji da kada vršimo predviđanje za jednu instancu, ne možemo da ispravno uradimo standardnu devijaciju
- Zbog toga se za vreme treniranja parametri μ i σ za svaki neuron pamte i u vreme predviđanja se koristi procena ovih parametara na osnovu vrednosti izračunatih tokom treniranja najčešće pomoću tzv. *pokretnog proseka*
- Pokretni prosek m_i u trenutku i nekog niza vrednosti $a_0, \ldots, a_n$ se računa kao $m_i = \alpha m_{i-1} + (1 - \alpha) a_i$ za neko $0 \leq \alpha \leq 1$

Stohastičko obučavanje

- Ne koristimo pun gradijentni spust izračunat nad celim skupom podataka već nad nekim njegovim podskupom

Stohastičko obučavanje

- Ne koristimo pun gradijentni spust izračunat nad celim skupom podataka već nad nekim njegovim podskupom
- Taj podskup se naziva *minibatch* mada je češće u upotrebi termin *batch*

Stohastičko obučavanje

- Ne koristimo pun gradijentni spust izračunat nad celim skupom podataka već nad nekim njegovim podskupom
- Taj podskup se naziva *minibatch* mada je češće u upotrebi termin *batch*
- Odredimo batch (veličine nekoliko desetina, tipično 32), na njemu izračunamo gradijent, uradimo jedan korak u optimizaciji, pa onda sledeći i tako redom

Stohastičko obučavanje

- Ne koristimo pun gradijentni spust izračunat nad celim skupom podataka već nad nekim njegovim podskupom
- Taj podskup se naziva *minibatch* mada je češće u upotrebi termin *batch*
- Odredimo batch (veličine nekoliko desetina, tipično 32), na njemu izračunamo gradijent, uradimo jedan korak u optimizaciji, pa onda sledeći i tako redom
- Prilikom formiranja batch-a važno je pažiti da instance unutar njega budu koliko-toliko raznovrsne

Stohastičko obučavanje

- Ne koristimo pun gradijentni spust izračunat nad celim skupom podataka već nad nekim njegovim podskupom
- Taj podskup se naziva *minibatch* mada je češće u upotrebi termin *batch*
- Odredimo batch (veličine nekoliko desetina, tipično 32), na njemu izračunamo gradijent, uradimo jedan korak u optimizaciji, pa onda sledeći i tako redom
- Prilikom formiranja batch-a važno je pažiti da instance unutar njega budu koliko-toliko raznovrsne
- Na primer, ako radimo klasifikaciju sa 50 klasa i svaki put batch sadrži samo instance koje se odnose samo na jednu klasu, dok dođemo do 50, mreža će zaboraviti da je prva klasa i postojala

Stohastičko obučavanje

- Ne koristimo pun gradijentni spust izračunat nad celim skupom podataka već nad nekim njegovim podskupom
- Taj podskup se naziva *minibatch* mada je češće u upotrebi termin *batch*
- Odredimo batch (veličine nekoliko desetina, tipično 32), na njemu izračunamo gradijent, uradimo jedan korak u optimizaciji, pa onda sledeći i tako redom
- Prilikom formiranja batch-a važno je pažiti da instance unutar njega budu koliko-toliko raznovrsne
- Na primer, ako radimo klasifikaciju sa 50 klasa i svaki put batch sadrži samo instance koje se odnose samo na jednu klasu, dok dođemo do 50, mreža će zaboraviti da je prva klasa i postojala
- Zato je dobro izmešati instance nasumice

Stohastičko obučavanje

- Ne koristimo pun gradijentni spust izračunat nad celim skupom podataka već nad nekim njegovim podskupom
- Taj podskup se naziva *minibatch* mada je češće u upotrebi termin *batch*
- Odredimo batch (veličine nekoliko desetina, tipično 32), na njemu izračunamo gradijent, uradimo jedan korak u optimizaciji, pa onda sledeći i tako redom
- Prilikom formiranja batch-a važno je pažiti da instance unutar njega budu koliko-toliko raznovrsne
- Na primer, ako radimo klasifikaciju sa 50 klasa i svaki put batch sadrži samo instance koje se odnose samo na jednu klasu, dok dođemo do 50, mreža će zaboraviti da je prva klasa i postojala
- Zato je dobro izmešati instance nasumice
- Kada radimo sa rekurentnim mrežama, važno je da cela sekvenca uđe u batch ako je to moguće

Odsecanje gradijenata

- Kod rekurentnih mreža, nestajući gradijenti su u nekoj meri ublaženi LSTM-om

Odsecanje gradijenata

- Kod rekurentnih mreža, nestajući gradijenti su u nekoj meri ublaženi LSTM-om
- Razmotrimo problem eksplodirajućih gradijenata

Odsecanje gradijenata

- Kod rekurentnih mreža, nestajući gradijenti su u nekoj meri ublaženi LSTM-om
- Razmotrimo problem eksplodirajućih gradijenata
- Kod nedostajućih gradijenata je problem što su gradijenti vrlo mali, kretanje je vrlo sporo i to će sporo konvergirati

Odsecanje gradijenata

- Kod rekurentnih mreža, nestajući gradijenti su u nekoj meri ublaženi LSTM-om
- Razmotrimo problem eksplodirajućih gradijenata
- Kod nedostajućih gradijenata je problem što su gradijenti vrlo mali, kretanje je vrlo sporo i to će sporo konvergirati
- Kod eksplodirajućih gradijenata je problem što su gradijenti vrlo veliki i kretanje divlje osciluje u optimizaciji i ništa ze zaista ne nauči

Odsecanje gradijenata

- Kod rekurentnih mreža, nestajući gradijenti su u nekoj meri ublaženi LSTM-om
- Razmotrimo problem eksplodirajućih gradijenata
- Kod nedostajućih gradijenata je problem što su gradijenti vrlo mali, kretanje je vrlo sporo i to će sporo konvergirati
- Kod eksplodirajućih gradijenata je problem što su gradijenti vrlo veliki i kretanje divlje osciluje u optimizaciji i ništa ze zaista ne nauči
- Ideja za ublažavanje problema eksplodirajućih gradijenata je skraćivanje gradijenta na neki način tako da gradijent zadržava pravac ali mu se intenzitet smanjuje

Odsecanje gradijenata

- Jedan pristup - sve koordinate gradijenta koje izlaze iz intervala $[-c, c]$ za neki meta parametar c se zaokruže na bližu od vrednosti $-c, c$

Odsecanje gradijenata

- Jedan pristup - sve koordinate gradijenta koje izlaze iz intervala $[-c, c]$ za neki meta parametar c se zaokruže na bližu od vrednosti $-c, c$
  - *gradient clipping*

Odsecanje gradijenata

- Jedan pristup - sve koordinate gradijenta koje izlaze iz intervala $[-c, c]$ za neki meta parametar c se zaokruže na bližu od vrednosti $-c, c$
  - *gradient clipping*
  - menja se pravac gradijenta

Odsecanje gradijenata

- Jedan pristup - sve koordinate gradijenta koje izlaze iz intervala $[-c, c]$ za neki meta parametar c se zaokruže na bližu od vrednosti $-c, c$
  - *gradient clipping*
  - menja se pravac gradijenta
- Drugi pristup - odsecanje norme gradijenta tako što se ukoliko važi $\|\nabla E(w)\| > c$, gradijent zameni vektorom

$$c\frac{\nabla E(w)}{\|\nabla E(w)\|}$$

Odsecanje gradijenata

- Jedan pristup - sve koordinate gradijenta koje izlaze iz intervala $[-c, c]$ za neki meta parametar c se zaokruže na bližu od vrednosti $-c, c$
  - *gradient clipping*
  - menja se pravac gradijenta
- Drugi pristup - odsecanje norme gradijenta tako što se ukoliko važi $\|\nabla E(w)\| > c$, gradijent zameni vektorom

$$c\frac{\nabla E(w)}{\|\nabla E(w)\|}$$

  - ne menja se pravac gradijenta

Odsecanje gradijenata

- Jedan pristup - sve koordinate gradijenta koje izlaze iz intervala $[-c, c]$ za neki meta parametar c se zaokruže na bližu od vrednosti $-c, c$
  - *gradient clipping*
  - menja se pravac gradijenta
- Drugi pristup - odsecanje norme gradijenta tako što se ukoliko važi $\|\nabla E(w)\| > c$, gradijent zameni vektorom

$$c\frac{\nabla E(w)}{\|\nabla E(w)\|}$$

  - ne menja se pravac gradijenta
  - ovo rešenje deluje bolje od prethodnog

Odsecanje gradijenata

- Jedan pristup - sve koordinate gradijenta koje izlaze iz intervala $[-c, c]$ za neki meta parametar c se zaokruže na bližu od vrednosti $-c, c$
  - *gradient clipping*
  - menja se pravac gradijenta
- Drugi pristup - odsecanje norme gradijenta tako što se ukoliko važi $\|\nabla E(w)\| > c$, gradijent zameni vektorom

$$c\frac{\nabla E(w)}{\|\nabla E(w)\|}$$

  - ne menja se pravac gradijenta
  - ovo rešenje deluje bolje od prethodnog
- U praksi se oba rešenja ponašaju približno jednako

Odsecanje gradijenata

- U praksi se oba rešenja ponašaju približno jednako

Odsecanje gradijenata

- U praksi se oba rešenja ponašaju približno jednako
- Jedno od objašnjenja zbog čega se to dešava je taj što gradijent u praksi uvek računamo na batch-u koji ima nekoliko desetina instanci iz skupa od nekoliko hiljada instanci

Odsecanje gradijenata

- U praksi se oba rešenja ponašaju približno jednako
- Jedno od objašnjenja zbog čega se to dešava je taj što gradijent u praksi uvek računamo na batch-u koji ima nekoliko desetina instanci iz skupa od nekoliko hiljada instanci
- Zavisno od toga koje su nam instance naišle u koji batch, dobićemo gradijente različitih intenziteta

Odsecanje gradijenata

- U praksi se oba rešenja ponašaju približno jednako
- Jedno od objašnjenja zbog čega se to dešava je taj što gradijent u praksi uvek računamo na batch-u koji ima nekoliko desetina instanci iz skupa od nekoliko hiljada instanci
- Zavisno od toga koje su nam instance naišle u koji batch, dobićemo gradijente različitih intenziteta
- Za jedan batch će nam norma ispasti prevelika, za drugi nece, i samo u onim batch-evima gde je prevelika cemo uraditi skraćivanje

Odsecanje gradijenata

- U praksi se oba rešenja ponašaju približno jednako
- Jedno od objašnjenja zbog čega se to dešava je taj što gradijent u praksi uvek računamo na batch-u koji ima nekoliko desetina instanci iz skupa od nekoliko hiljada instanci
- Zavisno od toga koje su nam instance naišle u koji batch, dobićemo gradijente različitih intenziteta
- Za jedan batch će nam norma ispasti prevelika, za drugi nece, i samo u onim batch-evima gde je prevelika cemo uraditi skraćivanje
- Kada bismo uradili uprosečavanje takvih vektora, ne bismo dobili isto kao kada ne bismo imali skraćivanje

Odsecanje gradijenata

- U praksi se oba rešenja ponašaju približno jednako
- Jedno od objašnjenja zbog čega se to dešava je taj što gradijent u praksi uvek računamo na batch-u koji ima nekoliko desetina instanci iz skupa od nekoliko hiljada instanci
- Zavisno od toga koje su nam instance naišle u koji batch, dobićemo gradijente različitih intenziteta
- Za jedan batch će nam norma ispasti prevelika, za drugi nece, i samo u onim batch-evima gde je prevelika cemo uraditi skraćivanje
- Kada bismo uradili uprosečavanje takvih vektora, ne bismo dobili isto kao kada ne bismo imali skraćivanje
- Uprosecavanje bez skraćivanja aproksimira stvarni gradijent - setimo se da kod stohastickog gradijentnog spusta biramo vektor cije je očekivanje pravi vektor

Odsecanje gradijenata

- U praksi se oba rešenja ponašaju približno jednako
- Jedno od objašnjenja zbog čega se to dešava je taj što gradijent u praksi uvek računamo na batch-u koji ima nekoliko desetina instanci iz skupa od nekoliko hiljada instanci
- Zavisno od toga koje su nam instance naišle u koji batch, dobićemo gradijente različitih intenziteta
- Za jedan batch će nam norma ispasti prevelika, za drugi nece, i samo u onim batch-evima gde je prevelika cemo uraditi skraćivanje
- Kada bismo uradili uprosečavanje takvih vektora, ne bismo dobili isto kao kada ne bismo imali skraćivanje
- Uprosecavanje bez skraćivanja aproksimira stvarni gradijent - setimo se da kod stohastickog gradijentnog spusta biramo vektor cije je očekivanje pravi vektor
- Sa ovim skraćivanjem više ne aproksimiramo stvarni gradijent jer smo poremetili ove gradijente na batch-evima

Regularizacija

- Govorili smo o regularizaciji kao konceptu i o nekim specifičnim regularizacijama

Regularizacija

- Govorili smo o regularizaciji kao konceptu i o nekim specifičnim regularizacijama
- Kod neuronskih mreža, metode regularizacije mogu izgledati nešto drugačije

Regularizacija

- Govorili smo o regularizaciji kao konceptu i o nekim specifičnim regularizacijama
- Kod neuronskih mreža, metode regularizacije mogu izgledati nešto drugačije
- Neuronske mreže su često modeli visoke fleksibilnosti pa je upotreba nekog vida regularizacije neophodna

Regularizacija

- Govorili smo o regularizaciji kao konceptu i o nekim specifičnim regularizacijama
- Kod neuronskih mreža, metode regularizacije mogu izgledati nešto drugačije
- Neuronske mreže su često modeli visoke fleksibilnosti pa je upotreba nekog vida regularizacije neophodna
- Unutrašnja standardizacija može smanjiti potrebu za regularizacijom ali ne znači da će to uvek biti slučaj

Regularizacija

- Imajmo u vidu da se kod neuronskih mreža model najčešće ne definiše tako što krećemo od nule

Regularizacija

- Imajmo u vidu da se kod neuronskih mreža model najčešće ne definiše tako što krećemo od nule
- Vrlo često postoje arhitekture koje su unapred definisane nad nekim podacima, npr. nad slikama najrazličitijih vrsta životinja, predmeta, itd.

Regularizacija

- Imajmo u vidu da se kod neuronskih mreža model najčešće ne definiše tako što krećemo od nule
- Vrlo često postoje arhitekture koje su unapred definisane nad nekim podacima, npr. nad slikama najrazličitijih vrsta životinja, predmeta, itd.
- Ovakvi modeli se mogu prilagoditi za konkretan problem tako što bismo krenuli od formiranog modela i onda bismo ga dotrenirali za konkretne podatke

Regularizacija

- Imajmo u vidu da se kod neuronskih mreža model najčešće ne definiše tako što krećemo od nule
- Vrlo često postoje arhitekture koje su unapred definisane nad nekim podacima, npr. nad slikama najrazličitijih vrsta životinja, predmeta, itd.
- Ovakvi modeli se mogu prilagoditi za konkretan problem tako što bismo krenuli od formiranog modela i onda bismo ga dotrenirali za konkretne podatke
- Ovakav pristup često vodi tome da nismo u prilici da podešavamo broj slojeva, broj neurona po sloju, broj filtera i slično od nule, već da uzmemo neku predefinisanu arhitekturu koja je možda malo većeg kapaciteta nego što bismo imali da smo gradili model od nule

Regularizacija

- Imajmo u vidu da se kod neuronskih mreža model najčešće ne definiše tako što krećemo od nule
- Vrlo često postoje arhitekture koje su unapred definisane nad nekim podacima, npr. nad slikama najrazličitijih vrsta životinja, predmeta, itd.
- Ovakvi modeli se mogu prilagoditi za konkretan problem tako što bismo krenuli od formiranog modela i onda bismo ga dotrenirali za konkretne podatke
- Ovakav pristup često vodi tome da nismo u prilici da podešavamo broj slojeva, broj neurona po sloju, broj filtera i slično od nule, već da uzmemo neku predefinisanu arhitekturu koja je možda malo većeg kapaciteta nego što bismo imali da smo gradili model od nule
- Ovo povećanje kapaciteta koje preti preprilagođavanjem onda rešavamo tako što pojačamo regularizaciju

Regularizacija

- Opšte vrste regularizacija u kontekstu neuronskih mreža

Regularizacija

- Opšte vrste regularizacija u kontekstu neuronskih mreža
  - ℓ_1 regularizacija koja se koristi za proređene modele i daje interpretabilnost se ne viđa cesto

Regularizacija

- Opšte vrste regularizacija u kontekstu neuronskih mreža
  - ℓ_1 regularizacija koja se koristi za proređene modele i daje interpretabilnost se ne viđa cesto
  - ℓ_2 regularizacija se često koristi

Regularizacija - rano zaustavljanje

- na prvi pogled ne deluje kao regularizacija ali jeste i postoji objašnjenje veze u određenim kontekstima sa ℓ_2 regularizacijom

Regularizacija - rano zaustavljanje

- na prvi pogled ne deluje kao regularizacija ali jeste i postoji objašnjenje veze u određenim kontekstima sa ℓ_2 regularizacijom
- problem preprilagođavanja nastaje tako što sa modelom visokog kapaciteta radimo prilagođavanje do kraja, dok grešku ne minimizujemo dokle god je moguće, dok ne dođemo do minimuma

Regularizacija - rano zaustavljanje

- na prvi pogled ne deluje kao regularizacija ali jeste i postoji objašnjenje veze u određenim kontekstima sa ℓ_2 regularizacijom
- problem preprilagođavanja nastaje tako što sa modelom visokog kapaciteta radimo prilagođavanje do kraja, dok grešku ne minimizujemo dokle god je moguće, dok ne dođemo do minimuma
- rano zaustavljanje podrazumeva da ne idemo do kraja, već da se zaustavimo pre nego što dodjemo do minimuma, i tako mrežu nećemo istrenirati do kraja i nećemo se prilagoditi do kraja

Regularizacija - rano zaustavljanje

- na prvi pogled ne deluje kao regularizacija ali jeste i postoji objašnjenje veze u određenim kontekstima sa ℓ_2 regularizacijom
- problem preprilagođavanja nastaje tako što sa modelom visokog kapaciteta radimo prilagođavanje do kraja, dok grešku ne minimizujemo dokle god je moguće, dok ne dođemo do minimuma
- rano zaustavljanje podrazumeva da ne idemo do kraja, već da se zaustavimo pre nego što dodjemo do minimuma, i tako mrežu nećemo istrenirati do kraja i nećemo se prilagoditi do kraja
- postavlja se pitanje kako ćemo znati kada da se zaustavimo

Regularizacija - rano zaustavljanje

- postoje sofisticirane metode koje tačku zaustavljanja određuju na osnovu performansi tekućeg modela na odvojenom validacionom skupu (ne na trening skupu!)

Regularizacija - rano zaustavljanje

- postoje sofisticirane metode koje tačku zaustavljanja određuju na osnovu performansi tekućeg modela na odvojenom validacionom skupu (ne na trening skupu!)
- ako nam greška na trening skupu pada a na validacionom raste, znači da nema smisla dalje trenirati jer se preprilagođavamo

Regularizacija - rano zaustavljanje

- postoje sofisticirane metode koje tačku zaustavljanja određuju na osnovu performansi tekućeg modela na odvojenom validacionom skupu (ne na trening skupu!)
- ako nam greška na trening skupu pada a na validacionom raste, znači da nema smisla dalje trenirati jer se preprilagođavamo
- kada nam greška skoči na validacionom skupu, ne mora da znači da je to trajno, možda će da posle toga padne, možda je taj skok bio relativno mali pa će ponovo da se nastavi pad

Regularizacija - rano zaustavljanje

- postoje sofisticirane metode koje tačku zaustavljanja određuju na osnovu performansi tekućeg modela na odvojenom validacionom skupu (ne na trening skupu!)
- ako nam greška na trening skupu pada a na validacionom raste, znači da nema smisla dalje trenirati jer se preprilagođavamo
- kada nam greška skoči na validacionom skupu, ne mora da znači da je to trajno, možda će da posle toga padne, možda je taj skok bio relativno mali pa će ponovo da se nastavi pad
- mogu da postoje razne strategije kada se prestaje sa treniranjem ali je ideja sledeća

Regularizacija - rano zaustavljanje

- postoje sofisticirane metode koje tačku zaustavljanja određuju na osnovu performansi tekućeg modela na odvojenom validacionom skupu (ne na trening skupu!)
- ako nam greška na trening skupu pada a na validacionom raste, znači da nema smisla dalje trenirati jer se preprilagođavamo
- kada nam greška skoči na validacionom skupu, ne mora da znači da je to trajno, možda će da posle toga padne, možda je taj skok bio relativno mali pa će ponovo da se nastavi pad
- mogu da postoje razne strategije kada se prestaje sa treniranjem ali je ideja sledeća
- treniramo veliki broj iteracija i nakon svake iteracije pratimo grešku na validacionom skupu, pamtimo kad god naiđemo na novu minimalnu vrednost greške i nastavljamo dalje da treniramo

Regularizacija - rano zaustavljanje

- postoje sofisticirane metode koje tačku zaustavljanja određuju na osnovu performansi tekućeg modela na odvojenom validacionom skupu (ne na trening skupu!)
- ako nam greška na trening skupu pada a na validacionom raste, znači da nema smisla dalje trenirati jer se preprilagođavamo
- kada nam greška skoči na validacionom skupu, ne mora da znači da je to trajno, možda će da posle toga padne, možda je taj skok bio relativno mali pa će ponovo da se nastavi pad
- mogu da postoje razne strategije kada se prestaje sa treniranjem ali je ideja sledeća
- treniramo veliki broj iteracija i nakon svake iteracije pratimo grešku na validacionom skupu, pamtimo kad god naiđemo na novu minimalnu vrednost greške i nastavljamo dalje da treniramo
- nakon toga, upotrebimo model koji je bio najbolji

Regularizacija - unošenje šuma

- Ne koristi se toliko kod današnjih neuronskih mreža

Regularizacija - unošenje šuma

- Ne koristi se toliko kod današnjih neuronskih mreža
- Podrazumeva smetanje mreži na najrazličitije načine, npr. kada neuron izračuna neku vrednost, dodamo slučajan šum na tu vrednost koji nije prevelik, šum se može primeniti i na parametre modela i na taj način gradijent nema mogućnost da se precizno usmeri prema minimumu i da se preprilagodi

Regularizacija - regularizacija izostavljanjem

- Imamo mrežu i radimo propagaciju unapred idući od ulaza prema izlazima

Regularizacija - regularizacija izostavljanjem

- Imamo mrežu i radimo propagaciju unapred idući od ulaza prema izlazima
- Posle toga računamo gradijent i radimo propagaciju unazad

Regularizacija - regularizacija izostavljanjem

- Imamo mrežu i radimo propagaciju unapred idući od ulaza prema izlazima
- Posle toga računamo gradijent i radimo propagaciju unazad
- Kada radimo propagaciju unapred, za svaki neuron sa verovatnoćom p odlučujemo da li ćemo da ga zadržimo ili ćemo da ga izostavimo (postavimo na 0)

Regularizacija - regularizacija izostavljanjem

- Imamo mrežu i radimo propagaciju unapred idući od ulaza prema izlazima
- Posle toga računamo gradijent i radimo propagaciju unazad
- Kada radimo propagaciju unapred, za svaki neuron sa verovatnoćom p odlučujemo da li ćemo da ga zadržimo ili ćemo da ga izostavimo (postavimo na 0)
- za jednu instancu ćemo tom neuronu možda dati 0, ali za sledeću ćemo možda zadržati njegovu vrednost

Regularizacija - regularizacija izostavljanjem

- Imamo mrežu i radimo propagaciju unapred idući od ulaza prema izlazima
- Posle toga računamo gradijent i radimo propagaciju unazad
- Kada radimo propagaciju unapred, za svaki neuron sa verovatnoćom p odlučujemo da li ćemo da ga zadržimo ili ćemo da ga izostavimo (postavimo na 0)
- za jednu instancu ćemo tom neuronu možda dati 0, ali za sledeću ćemo možda zadržati njegovu vrednost
- za svaku instancu odlučujemo da li cemo zadržati vrednost neurona ili ne

Regularizacija - regularizacija izostavljanjem

- Imamo mrežu i radimo propagaciju unapred idući od ulaza prema izlazima
- Posle toga računamo gradijent i radimo propagaciju unazad
- Kada radimo propagaciju unapred, za svaki neuron sa verovatnoćom p odlučujemo da li ćemo da ga zadržimo ili ćemo da ga izostavimo (postavimo na 0)
- za jednu instancu ćemo tom neuronu možda dati 0, ali za sledeću ćemo možda zadržati njegovu vrednost
- za svaku instancu odlučujemo da li cemo zadržati vrednost neurona ili ne
- odlučivanje se vrši na osnovu generisanih pseudoslučajnih brojeva

Regularizacija - regularizacija izostavljanjem

- Imamo mrežu i radimo propagaciju unapred idući od ulaza prema izlazima
- Posle toga računamo gradijent i radimo propagaciju unazad
- Kada radimo propagaciju unapred, za svaki neuron sa verovatnoćom p odlučujemo da li ćemo da ga zadržimo ili ćemo da ga izostavimo (postavimo na 0)
- za jednu instancu ćemo tom neuronu možda dati 0, ali za sledeću ćemo možda zadržati njegovu vrednost
- za svaku instancu odlučujemo da li cemo zadržati vrednost neurona ili ne
- odlučivanje se vrši na osnovu generisanih pseudoslučajnih brojeva
- ovo je ponovo neka vrsta smetanja mreži u procesu minimizacije

Regularizacija - regularizacija izostavljanjem

- ako mreža ima visok kapacitet, može da se preprilagodi podacima

Regularizacija - regularizacija izostavljanjem

- ako mreža ima visok kapacitet, može da se preprilagodi podacima
- ako krenemo da joj povremeno anuliramo neke neurone, koji možda računaju neke vrlo korisne atribute, šta mreža mora da uradi?

Regularizacija - regularizacija izostavljanjem

- ako mreža ima visok kapacitet, može da se preprilagodi podacima
- ako krenemo da joj povremeno anuliramo neke neurone, koji možda računaju neke vrlo korisne atribute, šta mreža mora da uradi?
  - mora da obezbedi da čak i kada je taj neuron anuliran, informaciju koju on nosi i dalje ima

Regularizacija - regularizacija izostavljanjem

- ako mreža ima visok kapacitet, može da se preprilagodi podacima
- ako krenemo da joj povremeno anuliramo neke neurone, koji možda računaju neke vrlo korisne atribute, šta mreža mora da uradi?
  - mora da obezbedi da čak i kada je taj neuron anuliran, informaciju koju on nosi i dalje ima
- a kako će to da obezbedi?

Regularizacija - regularizacija izostavljanjem

- ako mreža ima visok kapacitet, može da se preprilagodi podacima
- ako krenemo da joj povremeno anuliramo neke neurone, koji možda računaju neke vrlo korisne atribute, šta mreža mora da uradi?
  - mora da obezbedi da čak i kada je taj neuron anuliran, informaciju koju on nosi i dalje ima
- a kako će to da obezbedi?
  - tako što će naterati i neke druge neurone da i oni nose neku sličnu informaciju

Regularizacija - regularizacija izostavljanjem

- ako mreža ima visok kapacitet, može da se preprilagodi podacima
- ako krenemo da joj povremeno anuliramo neke neurone, koji možda računaju neke vrlo korisne atribute, šta mreža mora da uradi?
  - mora da obezbedi da čak i kada je taj neuron anuliran, informaciju koju on nosi i dalje ima
- a kako će to da obezbedi?
  - tako što će naterati i neke druge neurone da i oni nose neku sličnu informaciju
- to znači da će različite putanje u mreži biti u stanju da izračunaju iste informacije čime ćemo imati neku redundantnost informacija kroz mrežu

Regularizacija - regularizacija izostavljanjem

- ako mreža ima visok kapacitet, može da se preprilagodi podacima
- ako krenemo da joj povremeno anuliramo neke neurone, koji možda računaju neke vrlo korisne atribute, šta mreža mora da uradi?
  - mora da obezbedi da čak i kada je taj neuron anuliran, informaciju koju on nosi i dalje ima
- a kako će to da obezbedi?
  - tako što će naterati i neke druge neurone da i oni nose neku sličnu informaciju
- to znači da će različite putanje u mreži biti u stanju da izračunaju iste informacije čime ćemo imati neku redundantnost informacija kroz mrežu
- na taj način, mreži smo efektivno smanjili kapacitet

Regularizacija - regularizacija izostavljanjem

- na taj način, mreži smo efektivno smanjili kapacitet

Regularizacija - regularizacija izostavljanjem

- na taj način, mreži smo efektivno smanjili kapacitet
- ona sada ne može da koristi najrazličitije neurone da se specijalizuje za neke sitne detalje u podacima i da se tako preprilagodi tekućem trening skupu

Regularizacija - regularizacija izostavljanjem

- na taj način, mreži smo efektivno smanjili kapacitet
- ona sada ne može da koristi najrazličitije neurone da se specijalizuje za neke sitne detalje u podacima i da se tako preprilagodi tekućem trening skupu
- mora da se uhvati grubljih crta, a ne najfinijih detalja i da pritom njihovo izračunavanje umnoži kroz mrežu

Regularizacija - regularizacija izostavljanjem

- na taj način, mreži smo efektivno smanjili kapacitet
- ona sada ne može da koristi najrazličitije neurone da se specijalizuje za neke sitne detalje u podacima i da se tako preprilagodi tekućem trening skupu
- mora da se uhvati grubljih crta, a ne najfinijih detalja i da pritom njihovo izračunavanje umnoži kroz mrežu
- više nema pun kapacitet na raspolaganju za puno prilagođavanje tekućim podacima i njihovim najsitnijim detaljima već mora da se fokusira na važnije stvari i na taj način izbegava preprilagođavanje

Objašnjenje uspešnosti dubokih neuronskih mreža - dubina

- Teorema o univerzalnoj aproksimaciji za neuronske mreže tvrdi da neuronske mreže imaju svojstvo univerzalnog aproksimatora

Objašnjenje uspešnosti dubokih neuronskih mreža - dubina

- Teorema o univerzalnoj aproksimaciji za neuronske mreže tvrdi da neuronske mreže imaju svojstvo univerzalnog aproksimatora
- Svaka neprekidna funkcija se može proizvoljno dobro aproksimirati neuronskom mrežom ako ona ima dovoljno širok skriveni sloj i to čak može biti mreža dubine jedan

Objašnjenje uspešnosti dubokih neuronskih mreža - dubina

- Teorema o univerzalnoj aproksimaciji za neuronske mreže tvrdi da neuronske mreže imaju svojstvo univerzalnog aproksimatora
- Svaka neprekidna funkcija se može proizvoljno dobro aproksimirati neuronskom mrežom ako ona ima dovoljno širok skriveni sloj i to čak može biti mreža dubine jedan
- U teoremi nije precizirano koliko širok taj skriveni sloj treba da bude

Objašnjenje uspešnosti dubokih neuronskih mreža - dubina

- Teorema o univerzalnoj aproksimaciji za neuronske mreže tvrdi da neuronske mreže imaju svojstvo univerzalnog aproksimatora
- Svaka neprekidna funkcija se može proizvoljno dobro aproksimirati neuronskom mrežom ako ona ima dovoljno širok skriveni sloj i to čak može biti mreža dubine jedan
- U teoremi nije precizirano koliko širok taj skriveni sloj treba da bude
- Možda treba da bude ogroman i pitanje je da li nam ova teorema govori nešto što je praktično upotrebljivo

Objašnjenje uspešnosti dubokih neuronskih mreža - dubina

- Ispostavlja se da pod odredjenim pretpostavkama povećanje dubine mreže može da dovede do eksponencijalnog smanjenja u broju neophodnih neurona po jednom nivou mreže

Objašnjenje uspešnosti dubokih neuronskih mreža - dubina

- Ispostavlja se da pod odredjenim pretpostavkama povećanje dubine mreže može da dovede do eksponencijalnog smanjenja u broju neophodnih neurona po jednom nivou mreže
- Ako hocemo da mrežu koja je duboka predstavimo pomocu jednog skrivenog sloja, broj neurona u tom skrivenom sloju neće biti jednak dubini polazne mreže već može biti eksponencijalan u odnosu na dubinu polazne mreže

Objašnjenje uspešnosti dubokih neuronskih mreža - dubina

- Ispostavlja se da pod odredjenim pretpostavkama povećanje dubine mreže može da dovede do eksponencijalnog smanjenja u broju neophodnih neurona po jednom nivou mreže
- Ako hocemo da mrežu koja je duboka predstavimo pomocu jednog skrivenog sloja, broj neurona u tom skrivenom sloju neće biti jednak dubini polazne mreže već može biti eksponencijalan u odnosu na dubinu polazne mreže
- Postoje objašnjenja koja kažu da je dubina mreže suštinski bitna stvar

Objašnjenje uspešnosti dubokih neuronskih mreža - dubina

- Ispostavlja se da pod odredjenim pretpostavkama povećanje dubine mreže može da dovede do eksponencijalnog smanjenja u broju neophodnih neurona po jednom nivou mreže
- Ako hocemo da mrežu koja je duboka predstavimo pomocu jednog skrivenog sloja, broj neurona u tom skrivenom sloju neće biti jednak dubini polazne mreže već može biti eksponencijalan u odnosu na dubinu polazne mreže
- Postoje objašnjenja koja kažu da je dubina mreže suštinski bitna stvar
- Možemo neuporedivo kompleksnije funkcije da izrazimo ako imamo dublju mrežu nego širu mrezu

Objašnjenje uspešnosti dubokih neuronskih mreža - dubina

- Ispostavlja se da pod odredjenim pretpostavkama povećanje dubine mreže može da dovede do eksponencijalnog smanjenja u broju neophodnih neurona po jednom nivou mreže
- Ako hocemo da mrežu koja je duboka predstavimo pomocu jednog skrivenog sloja, broj neurona u tom skrivenom sloju neće biti jednak dubini polazne mreže već može biti eksponencijalan u odnosu na dubinu polazne mreže
- Postoje objašnjenja koja kažu da je dubina mreže suštinski bitna stvar
- Možemo neuporedivo kompleksnije funkcije da izrazimo ako imamo dublju mrežu nego širu mrezu
- Dubina predstavlja nov kvalitet i povećavanje dubine mreže se ne može izjednačiti sa povećavanjem širine

Objašnjenje uspešnosti dubokih neuronskih mreža - lokalni optimumi

- Vrlo bitna linija istraživanja teme uspešnosti neuronskih mreža su pitanja vezana za optimizaciju

Objašnjenje uspešnosti dubokih neuronskih mreža - lokalni optimumi

- Vrlo bitna linija istraživanja teme uspešnosti neuronskih mreža su pitanja vezana za optimizaciju
- Naime, kada imamo funkciju sa velikim brojem parametara, one su nekonveksne i očekivanje je da imaju ogroman broj lokalnih minimuma koje će sprečiti algoritam optimizacije da pronađe globalni minimum

Objašnjenje uspešnosti dubokih neuronskih mreža - lokalni optimumi

- Vrlo bitna linija istraživanja teme uspešnosti neuronskih mreža su pitanja vezana za optimizaciju
- Naime, kada imamo funkciju sa velikim brojem parametara, one su nekonveksne i očekivanje je da imaju ogroman broj lokalnih minimuma koje će sprečiti algoritam optimizacije da pronađe globalni minimum
- Dugo je postojala tradicija da kada neuronska mreža ne daje očekivane rezultate, da kažemo da je završila u lokalnom optimumu

Objašnjenje uspešnosti dubokih neuronskih mreža - lokalni optimumi

- Vrlo bitna linija istraživanja teme uspešnosti neuronskih mreža su pitanja vezana za optimizaciju
- Naime, kada imamo funkciju sa velikim brojem parametara, one su nekonveksne i očekivanje je da imaju ogroman broj lokalnih minimuma koje će sprečiti algoritam optimizacije da pronađe globalni minimum
- Dugo je postojala tradicija da kada neuronska mreža ne daje očekivane rezultate, da kažemo da je završila u lokalnom optimumu
- Više grupa je istraživalo ove fenomene i kristalisalo se mišljenje da lokalni optimumi ne predstavljaju veliki problem

Objašnjenje uspešnosti dubokih neuronskih mreža - lokalni optimumi

- Vrlo bitna linija istraživanja teme uspešnosti neuronskih mreža su pitanja vezana za optimizaciju
- Naime, kada imamo funkciju sa velikim brojem parametara, one su nekonveksne i očekivanje je da imaju ogroman broj lokalnih minimuma koje će sprečiti algoritam optimizacije da pronađe globalni minimum
- Dugo je postojala tradicija da kada neuronska mreža ne daje očekivane rezultate, da kažemo da je završila u lokalnom optimumu
- Više grupa je istraživalo ove fenomene i kristalisalo se mišljenje da lokalni optimumi ne predstavljaju veliki problem
- Kada bi predstavljali, pitanje je da li bi mreže mogle da se ponašaju toliko dobro u mnogim vrlo složenim problemima (igra GO, computer vision i slično)

Objašnjenje uspešnosti dubokih neuronskih mreža - lokalni optimumi

- Vrlo bitna linija istraživanja teme uspešnosti neuronskih mreža su pitanja vezana za optimizaciju
- Naime, kada imamo funkciju sa velikim brojem parametara, one su nekonveksne i očekivanje je da imaju ogroman broj lokalnih minimuma koje će sprečiti algoritam optimizacije da pronađe globalni minimum
- Dugo je postojala tradicija da kada neuronska mreža ne daje očekivane rezultate, da kažemo da je završila u lokalnom optimumu
- Više grupa je istraživalo ove fenomene i kristalisalo se mišljenje da lokalni optimumi ne predstavljaju veliki problem
- Kada bi predstavljali, pitanje je da li bi mreže mogle da se ponašaju toliko dobro u mnogim vrlo složenim problemima (igra GO, computer vision i slično)
- Samim tim što se mreže ovako dobro ponašaju u teškim problemima, bilo je postavljeno pitanje da li problem lokalnih minimuma uopšte postoji

Objašnjenje uspešnosti dubokih neuronskih mreža - lokalni optimumi

- Jedno istraživanje je pokazalo da je pod određenim pretpostavkama verovatnoća da se mreža završiti u lokalnom minimumu utoliko manja ukoliko je lokalni minimum lošiji (veći od globalnog)

Objašnjenje uspešnosti dubokih neuronskih mreža - lokalni optimumi

- Jedno istraživanje je pokazalo da je pod određenim pretpostavkama verovatnoća da se mreža završiti u lokalnom minimumu utoliko manja ukoliko je lokalni minimum lošiji (veći od globalnog)
- Pretpostavke pod kojima je ovo dokazano nisu bile nimalo realistične pa sam dokaz nije bio od velikog značaja

Objašnjenje uspešnosti dubokih neuronskih mreža - lokalni optimumi

- Jedno istraživanje je pokazalo da je pod određenim pretpostavkama verovatnoća da se mreža završiti u lokalnom minimumu utoliko manja ukoliko je lokalni minimum lošiji (veći od globalnog)
- Pretpostavke pod kojima je ovo dokazano nisu bile nimalo realistične pa sam dokaz nije bio od velikog značaja
- Sledeće istraživanje godinu dana kasnije je pokazalo da sa neuporedivo manje pretpostavki nego u prethodnom da su svi lokalni optimumi istovremeno i globalni optimumi

Objašnjenje uspešnosti dubokih neuronskih mreža - lokalni optimumi

- Jedno istraživanje je pokazalo da je pod određenim pretpostavkama verovatnoća da se mreža završiti u lokalnom minimumu utoliko manja ukoliko je lokalni minimum lošiji (veći od globalnog)
- Pretpostavke pod kojima je ovo dokazano nisu bile nimalo realistične pa sam dokaz nije bio od velikog značaja
- Sledeće istraživanje godinu dana kasnije je pokazalo da sa neuporedivo manje pretpostavki nego u prethodnom da su svi lokalni optimumi istovremeno i globalni optimumi
- Iako su pretpostavke dosta olabavljene, i dalje su bile nerealistične

Objašnjenje uspešnosti dubokih neuronskih mreža - lokalni optimumi

- Jedno istraživanje je pokazalo da je pod određenim pretpostavkama verovatnoća da se mreža završiti u lokalnom minimumu utoliko manja ukoliko je lokalni minimum lošiji (veći od globalnog)
- Pretpostavke pod kojima je ovo dokazano nisu bile nimalo realistične pa sam dokaz nije bio od velikog značaja
- Sledeće istraživanje godinu dana kasnije je pokazalo da sa neuporedivo manje pretpostavki nego u prethodnom da su svi lokalni optimumi istovremeno i globalni optimumi
- Iako su pretpostavke dosta olabavljene, i dalje su bile nerealistične
- Vremenom se kristališe ubeđenje da iako ove teoreme nisu dokazane pod realističnim pretpostavkama, ipak nesto slično važi i za realne situacije

Objašnjenje uspešnosti dubokih neuronskih mreža - lokalni optimumi

- Jedan intuitivan argument bi bio sledeći.

Objašnjenje uspešnosti dubokih neuronskih mreža - lokalni optimumi

- Jedan intuitivan argument bi bio sledeći.
- U lokalnom optimumu, funkcija mora biti konveksna, odnosno hesijan funkcije mora biti pozitivno semidefinitan, odnosno mora imati nenegativne sopstvene vrednosti.

Objašnjenje uspešnosti dubokih neuronskih mreža - lokalni optimumi

- Jedan intuitivan argument bi bio sledeći.
- U lokalnom optimumu, funkcija mora biti konveksna, odnosno hesijan funkcije mora biti pozitivno semidefinitan, odnosno mora imati nenegativne sopstvene vrednosti.
- Iako je realističnost ove pretpostavke upitna, zamislimo da se znak sopstvene vrednosti bira bacanjem novčića.

Objašnjenje uspešnosti dubokih neuronskih mreža - lokalni optimumi

- Jedan intuitivan argument bi bio sledeći.
- U lokalnom optimumu, funkcija mora biti konveksna, odnosno hesijan funkcije mora biti pozitivno semidefinitan, odnosno mora imati nenegativne sopstvene vrednosti.
- Iako je realističnost ove pretpostavke upitna, zamislimo da se znak sopstvene vrednosti bira bacanjem novčića.
- Verovatnoća da sve sopstvene vrednosti budu pozitivne (nenegativne) eksponencijalno opada sa dimenzionalnošću prostora nad kojim je funkcija definisana.

Objašnjenje uspešnosti dubokih neuronskih mreža - lokalni optimumi

- Jedan intuitivan argument bi bio sledeći.
- U lokalnom optimumu, funkcija mora biti konveksna, odnosno hesijan funkcije mora biti pozitivno semidefinitan, odnosno mora imati nenegativne sopstvene vrednosti.
- Iako je realističnost ove pretpostavke upitna, zamislimo da se znak sopstvene vrednosti bira bacanjem novčića.
- Verovatnoća da sve sopstvene vrednosti budu pozitivne (nenegativne) eksponencijalno opada sa dimenzionalnošću prostora nad kojim je funkcija definisana.
- Odnosno, postojanje velikog broja minimuma u visoko dimenzionalnim prostorima nije mnogo verovatno.

Objašnjenje uspešnosti dubokih neuronskih mreža - lokalni optimumi

- Jedan intuitivan argument bi bio sledeći.
- U lokalnom optimumu, funkcija mora biti konveksna, odnosno hesijan funkcije mora biti pozitivno semidefinitan, odnosno mora imati nenegativne sopstvene vrednosti.
- Iako je realističnost ove pretpostavke upitna, zamislimo da se znak sopstvene vrednosti bira bacanjem novčića.
- Verovatnoća da sve sopstvene vrednosti budu pozitivne (nenegativne) eksponencijalno opada sa dimenzionalnošću prostora nad kojim je funkcija definisana.
- Odnosno, postojanje velikog broja minimuma u visoko dimenzionalnim prostorima nije mnogo verovatno.
- Verovatnije je da će se naći neki pravac bekstva iz tog minimuma, definisan sopstvenim vektorom hesijana koji odgovara negativnoj sopstvenoj vrednosti.

Objašnjenje uspešnosti dubokih neuronskih mreža - lokalni optimumi

- Jedan intuitivan argument bi bio sledeći.
- U lokalnom optimumu, funkcija mora biti konveksna, odnosno hesijan funkcije mora biti pozitivno semidefinitan, odnosno mora imati nenegativne sopstvene vrednosti.
- Iako je realističnost ove pretpostavke upitna, zamislimo da se znak sopstvene vrednosti bira bacanjem novčića.
- Verovatnoća da sve sopstvene vrednosti budu pozitivne (nenegativne) eksponencijalno opada sa dimenzionalnošću prostora nad kojim je funkcija definisana.
- Odnosno, postojanje velikog broja minimuma u visoko dimenzionalnim prostorima nije mnogo verovatno.
- Verovatnije je da će se naći neki pravac bekstva iz tog minimuma, definisan sopstvenim vektorom hesijana koji odgovara negativnoj sopstvenoj vrednosti.
- Zapravo, u skorije vreme je sve čvršće ubeđenje da glavni problem obučavanja neuronskih mreža nije vezan za lokalne minimume, već pre za platoe malog nagiba na kojima gradijenti nestaju, što dovodi do neefikasnosti gradijentnih metoda.

Objašnjenje uspešnosti dubokih neuronskih mreža - stohastički gradijentni spust

- U poslednjih par godina se intenzivnije ispituje ponašanje stohastičkog gradijentnog spusta

Objašnjenje uspešnosti dubokih neuronskih mreža - stohastički gradijentni spust

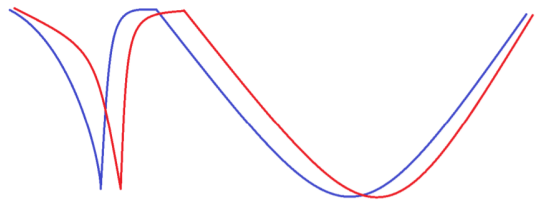
- U poslednjih par godina se intenzivnije ispituje ponašanje stohastičkog gradijentnog spusta
- Empirijski se pokazalo da stohastički gradijentni spust, iako nije metod regularizacije već čist metod optimizacije, ima i određene efekte regularizacije

Objašnjenje uspešnosti dubokih neuronskih mreža - stohastički gradijentni spust

- U poslednjih par godina se intenzivnije ispituje ponašanje stohastičkog gradijentnog spusta
- Empirijski se pokazalo da stohastički gradijentni spust, iako nije metod regularizacije već čist metod optimizacije, ima i određene efekte regularizacije
- Dešava se da se stohastičkim gradijentnim spustom, doduše sporije, neretko dolazi do rezultata koji su posle bolji u generalizaciji

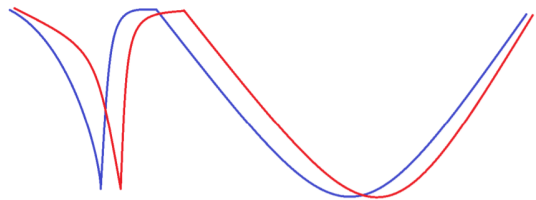
Objašnjenje uspešnosti dubokih neuronskih mreža - stohastički gradijentni spust

- Plavo - rizik - teorijska funkcija greške kada ne baratamo pojedinačnim podacima iz uzorka već celom raspodelom po tim podacima, ona funkcija koju aproksimiramo pomoću uzorka



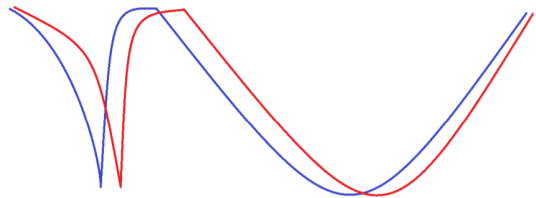
Objašnjenje uspešnosti dubokih neuronskih mreža - stohastički gradijentni spust

- Plavo - rizik - teorijska funkcija greške kada ne baratamo pojedinačnim podacima iz uzorka već celom raspodelom po tim podacima, ona funkcija koju aproksimiramo pomoću uzorka
- Želimo da nađemo jedan od dva minimuma, odnosno parametre koji odgovaraju jednoj od dve minimalne vrednosti rizika



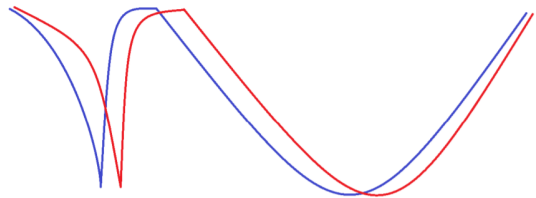
Objašnjenje uspešnosti dubokih neuronskih mreža - stohastički gradijentni spust

- Nemamo pristup plavoj funkciji



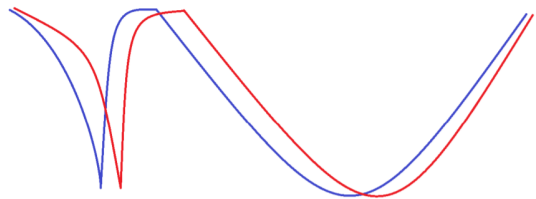
Objašnjenje uspešnosti dubokih neuronskih mreža - stohastički gradijentni spust

- Nemamo pristup plavoj funkciji
- Na svojim podacima možemo da izračunamo empirijski rizik - crvenu funkciju koja aproksimira ovu plavu



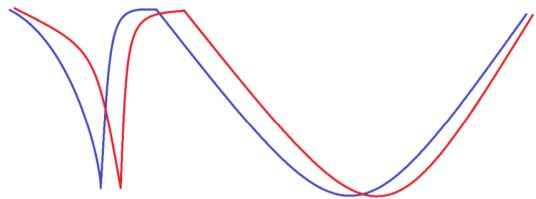
Objašnjenje uspešnosti dubokih neuronskih mreža - stohastički gradijentni spust

- Nemamo pristup plavoj funkciji
- Na svojim podacima možemo da izračunamo empirijski rizik - crvenu funkciju koja aproksimira ovu plavu
- Ako imamo preciznu metodu optimizacije, njoj je svejedno da li će otići u jedan ili drugi minimum, podjednako su dobri



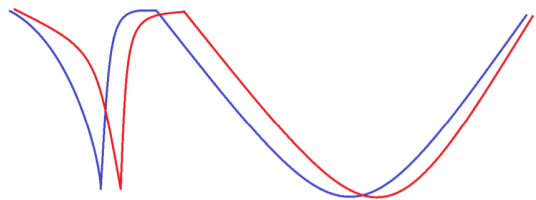
Objašnjenje uspešnosti dubokih neuronskih mreža - stohastički gradijentni spust

- Nemamo pristup plavoj funkciji
- Na svojim podacima možemo da izračunamo empirijski rizik - crvenu funkciju koja aproksimira ovu plavu
- Ako imamo preciznu metodu optimizacije, njoj je svejedno da li će otići u jedan ili drugi minimum, podjednako su dobri
- S obzirom da je metoda precizna, to znači da može da se spusti u ovaj vrlo uzak minimum



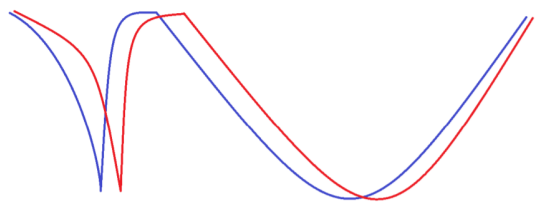
Objašnjenje uspešnosti dubokih neuronskih mreža - stohastički gradijentni spust

- Pretpostavimo da je metoda pronašla ovaj uzani minimum



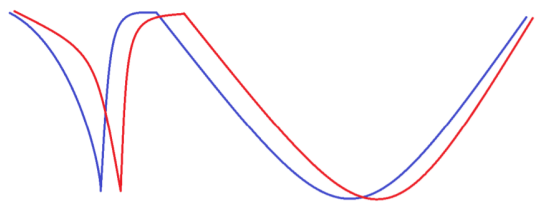
Objašnjenje uspešnosti dubokih neuronskih mreža - stohastički gradijentni spust

- Pretpostavimo da je metoda pronašla ovaj uzani minimum
- Tu je mala vrednost funkcije greške koju optimizujemo ali vrednost stvarne funkcije greške je ogromna



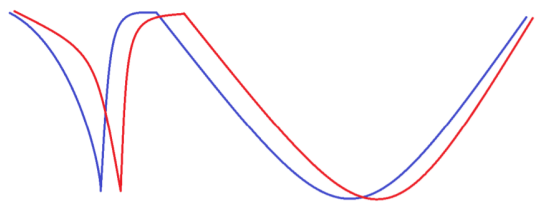
Objašnjenje uspešnosti dubokih neuronskih mreža - stohastički gradijentni spust

- Pretpostavimo da je metoda pronašla ovaj uzani minimum
- Tu je mala vrednost funkcije greške koju optimizujemo ali vrednost stvarne funkcije greške je ogromna
- To znači da na novim podacima koji dolaze iz raspodele na osnovu koje smo crtali plavu funkciju kada izračunamo takvu grešku, imaćemo vrlo loše rezultate



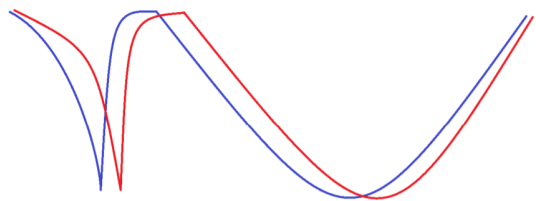
Objašnjenje uspešnosti dubokih neuronskih mreža - stohastički gradijentni spust

- Pretpostavimo da je metoda pronašla ovaj uzani minimum
- Tu je mala vrednost funkcije greške koju optimizujemo ali vrednost stvarne funkcije greške je ogromna
- To znači da na novim podacima koji dolaze iz raspodele na osnovu koje smo crtali plavu funkciju kada izračunamo takvu grešku, imaćemo vrlo loše rezultate
- Sa druge strane, ako bi našao široki optimum, to neće biti slučaj



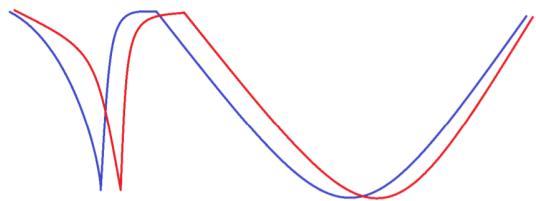
Objašnjenje uspešnosti dubokih neuronskih mreža - stohastički gradijentni spust

- Stohastički gradijentni spust nije precizna metoda optimizacije



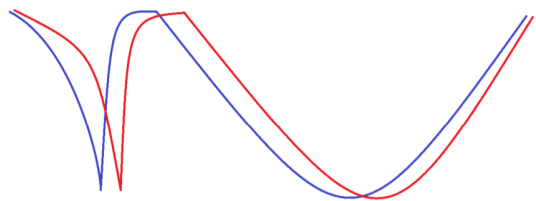
Objašnjenje uspešnosti dubokih neuronskih mreža - stohastički gradijentni spust

- Stohastički gradijentni spust nije precizna metoda optimizacije
- On bira loše pravce da bi optimizovao



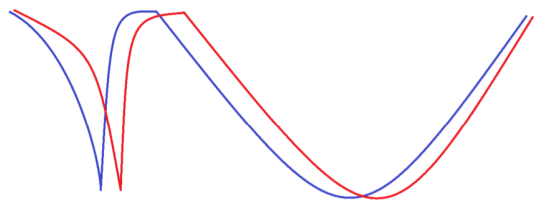
Objašnjenje uspešnosti dubokih neuronskih mreža - stohastički gradijentni spust

- Stohastički gradijentni spust nije precizna metoda optimizacije
- On bira loše pravce da bi optimizovao
- Možda će nabasati negde na region uskog lokalnog optimuma ali pošto je toliko stohastičan, ne može da se usmeri i ubrzo će jednim lošim potezom iskočiti



Objašnjenje uspešnosti dubokih neuronskih mreža - stohastički gradijentni spust

- Stohastički gradijentni spust nije precizna metoda optimizacije
- On bira loše pravce da bi optimizovao
- Možda će nabasati negde na region uskog lokalnog optimuma ali pošto je toliko stohastičan, ne može da se usmeri i ubrzo će jednim lošim potezom iskočiti
- Kod širokog optimuma, ima više širine da stohastički gradijentni spust iskonvergira do minimuma



Objašnjenje uspešnosti dubokih neuronskih mreža - stohastički gradijentni spust

- Stohastički gradijentni spust nije precizna metoda optimizacije
- On bira loše pravce da bi optimizovao
- Možda će nabasati negde na region uskog lokalnog optimuma ali pošto je toliko stohastičan, ne može da se usmeri i ubrzo će jednim lošim potezom iskočiti
- Kod širokog optimuma, ima više širine da stohastički gradijentni spust iskonvergira do minimuma
- Ispostavlja se da će lakše iskonvergirati ka širokim optimumima a oni bolje generalizuju