Optimizacija

Mašinsko učenje 2020/21.

Matematički fakultet
Univerzitet u Beogradu

Optimizacija

- Sve vreme govorimo o želji da minimizujemo neke funkcije nad nekim domenom (npr. nad R^n) po nekim promenljivim (tipično w)

Optimizacija

- Sve vreme govorimo o želji da minimizujemo neke funkcije nad nekim domenom (npr. nad R^n) po nekim promenljivim (tipično w)
- Opšti *problem optimizacije* je obično oblika:

$$\min_{x \in \mathcal{D}} f(x)$$
$$\text{pri uslovima} \quad g_i(x) \leq 0 \qquad i = 1, \ldots, L$$

Optimizacija

- Sve vreme govorimo o želji da minimizujemo neke funkcije nad nekim domenom (npr. nad R^n) po nekim promenljivim (tipično w)
- Opšti *problem optimizacije* je obično oblika:

$$\min_{x \in \mathcal{D}} f(x)$$
$$\text{pri uslovima} \quad g_i(x) \leq 0 \qquad i = 1, \ldots, L$$

- Često nećemo diskutovati ovakve uslove i razmatraćemo minimizaciju funkcije bez dodatnih ograničenja

Optimizacija

- Želimo da nađemo minimum funkcije i to po mogućstvu *globalni* minimum

Optimizacija

- Želimo da nađemo minimum funkcije i to po mogućstvu *globalni* minimum
- Međutim, pošto ne postoje egzaktne metode globalne optimizacije, govorimo o lokalnoj optimizaciji i o nalaženju lokalnih optimuma

Optimizacija

- Želimo da nađemo minimum funkcije i to po mogućstvu *globalni* minimum
- Međutim, pošto ne postoje egzaktne metode globalne optimizacije, govorimo o lokalnoj optimizaciji i o nalaženju lokalnih optimuma
- Za funkcije koje su konveksne, pronađeni lokalni optimum je sigurno globalni optimum

Optimizacija

- Želimo da nađemo minimum funkcije i to po mogućstvu *globalni* minimum
- Međutim, pošto ne postoje egzaktne metode globalne optimizacije, govorimo o lokalnoj optimizaciji i o nalaženju lokalnih optimuma
- Za funkcije koje su konveksne, pronađeni lokalni optimum je sigurno globalni optimum
- Za funkcije koje nisu konveksne, pronađeni lokalni optimum ne mora biti globalni optimum i tada možemo zapasti u lokalni optimum

Optimizacija

- Želimo da nađemo minimum funkcije i to po mogućstvu *globalni* minimum
- Međutim, pošto ne postoje egzaktne metode globalne optimizacije, govorimo o lokalnoj optimizaciji i o nalaženju lokalnih optimuma
- Za funkcije koje su konveksne, pronađeni lokalni optimum je sigurno globalni optimum
- Za funkcije koje nisu konveksne, pronađeni lokalni optimum ne mora biti globalni optimum i tada možemo zapasti u lokalni optimum
- Ipak, u praksi se pokazuje da je i to dovoljno dobro

Pregled

Gradijentni spust

- Osnova svih metoda optimizacije

Gradijentni spust

- Osnova svih metoda optimizacije
- Najjednostavnija i najpoznatija metoda optimizacije prvog reda (što znači da koristi samo parcijalne izvode prvog reda) za diferencijabilne funkcije

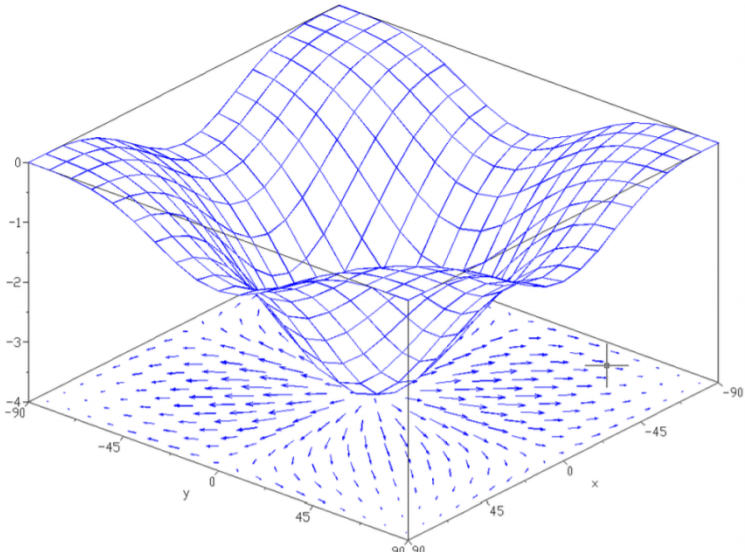
Gradijentni spust

- Osnova svih metoda optimizacije
- Najjednostavnija i najpoznatija metoda optimizacije prvog reda (što znači da koristi samo parcijalne izvode prvog reda) za diferencijabilne funkcije
- Podsetimo se, gradijentom nazivamo vektor parcijalnih izvoda

$$\nabla f(x_1, \ldots, x_n) = \left(\frac{\partial f}{\partial x_1}, \ldots, \frac{\partial f}{\partial x_n} \right)$$

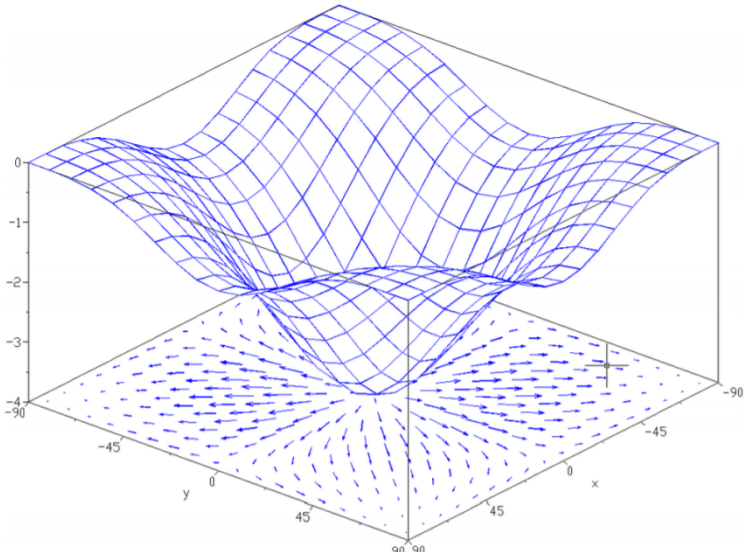
Gradijentni spust

- Posmatrajmo grafik funkcije greške za koju tražimo minimum



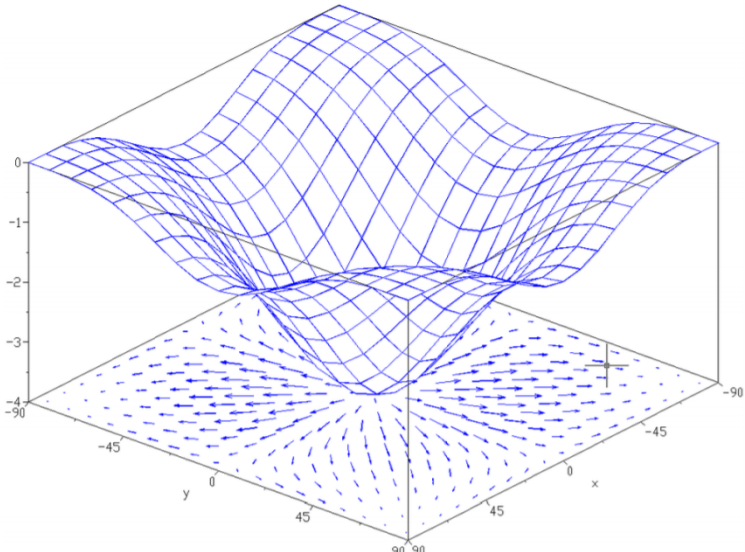
Gradijentni spust

- Posmatrajmo grafik funkcije greške za koju tražimo minimum
- Ovo je funkcija dve promenljive i stoga je gradijent dvodimenzioni vektor



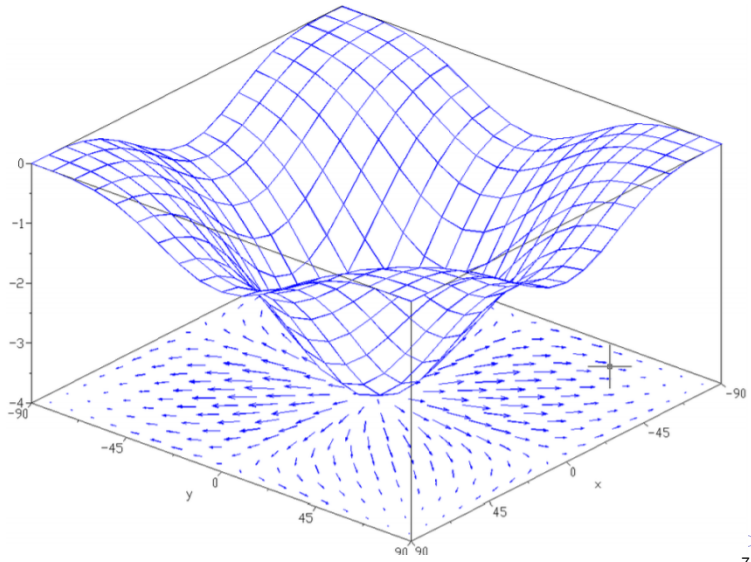
Gradijentni spust

- Posmatrajmo grafik funkcije greške za koju tražimo minimum
- Ovo je funkcija dve promenljive i stoga je gradijent dvodimenzioni vektor
- Gradijent zbog toga crtamo kao polje vektora (u nekim izvorima se pogrešno prikazuje kao tangenta na ovu površ)



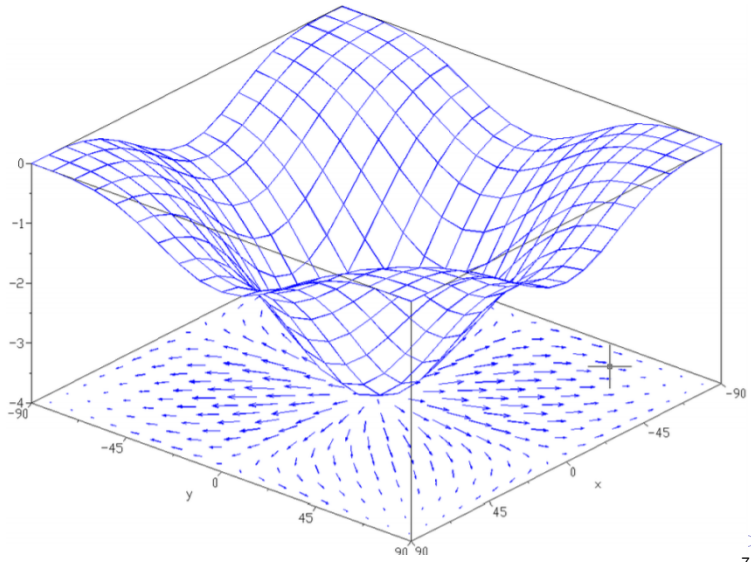
Gradijentni spust

- Za svaku tačku funkcije x i y gradijent u toj tački koji nam govori u kom pravcu funkcija najbrže raste



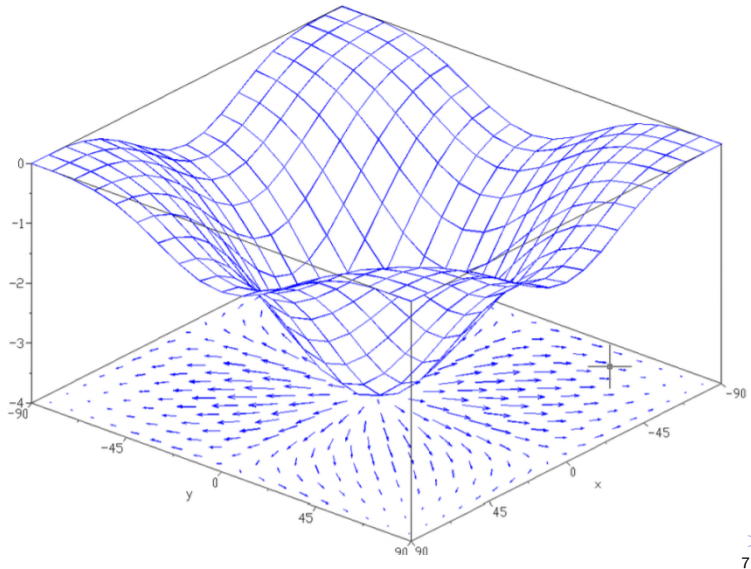
Gradijentni spust

- Za svaku tačku funkcije x i y gradijent u toj tački koji nam govori u kom pravcu funkcija najbrže raste
- Što funkcija brže raste, to je veći intenzitet gradijenta i obrnuto



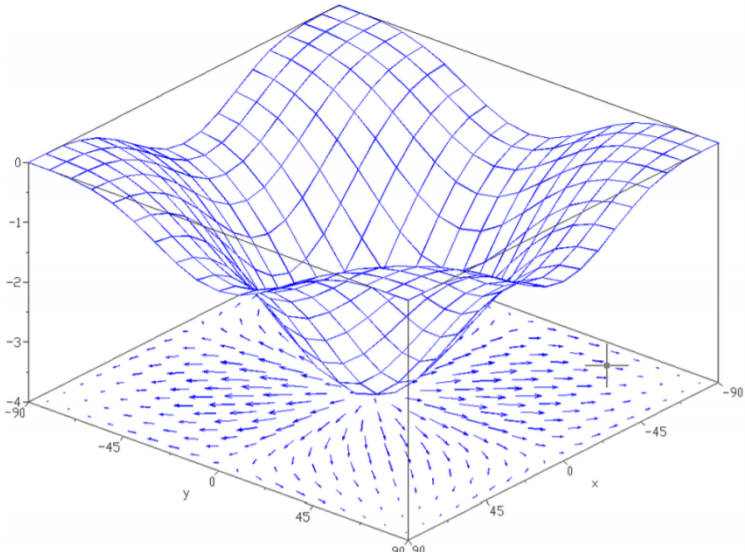
Gradijentni spust

- Za svaku tačku funkcije x i y gradijent u toj tački koji nam govori u kom pravcu funkcija najbrže raste
- Što funkcija brže raste, to je veći intenzitet gradijenta i obrnuto
- Na krajevima gde je funkcija ravna, gradijenti su ispali kao tačkice a gde je vrlo strma, tu su dugački



Gradijentni spust

- Osnovna ideja svih metoda prvog reda: ako želimo da dođemo do minimuma funkcije, treba da se krećemo u smeru suprotnom od gradijenta, odnosno da nekako oduzimamo gradijent možda pomnožen nekim skalarom i da se kroz niz koraka približimo minimumu



Gradijentni spust

- Da vidimo kako je formulisan gradijentni spust

Gradijentni spust

- Da vidimo kako je formulisan gradijentni spust
- Pretpostavimo da krećemo od neke polazne tačke x_0

Gradijentni spust

- Da vidimo kako je formulisan gradijentni spust
- Pretpostavimo da krećemo od neke polazne tačke x_0
  - Otkud nam ova tačka?

Gradijentni spust

- Da vidimo kako je formulisan gradijentni spust
- Pretpostavimo da krećemo od neke polazne tačke x_0
  - Otkud nam ova tačka?
  - Vrlo često je nasumice inicijalizovana

Gradijentni spust

- Da vidimo kako je formulisan gradijentni spust
- Pretpostavimo da krećemo od neke polazne tačke x_0
  - Otkud nam ova tačka?
  - Vrlo često je nasumice inicijalizovana
  - Ako imamo predstavu gde bi mogao biti minimum, možemo inicijalizovati x_0 tako da bude blizu

Gradijentni spust

- Da vidimo kako je formulisan gradijentni spust
- Pretpostavimo da krećemo od neke polazne tačke x_0
  - Otkud nam ova tačka?
  - Vrlo često je nasumice inicijalizovana
  - Ako imamo predstavu gde bi mogao biti minimum, možemo inicijalizovati x_0 tako da bude blizu
- Dalje, u svakom koraku (k je redni broj koraka) se pomeramo u narednu tačku (x_k) na sledeći način:

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k)$$

Gradijentni spust

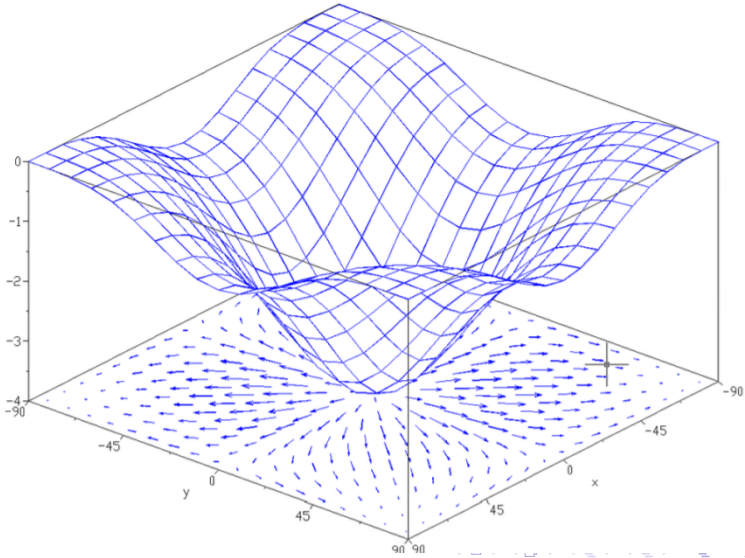
- Da vidimo kako je formulisan gradijentni spust
- Pretpostavimo da krećemo od neke polazne tačke x_0
  - Otkud nam ova tačka?
  - Vrlo često je nasumice inicijalizovana
  - Ako imamo predstavu gde bi mogao biti minimum, možemo inicijalizovati x_0 tako da bude blizu
- Dalje, u svakom koraku (k je redni broj koraka) se pomeramo u narednu tačku (x_k) na sledeći način:

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k)$$

- Skalar α nam dozvoljava da kontrolišemo dužinu koraka (dužinu vektora $\nabla f(x_k)$) i samim tim brzinu kretanja

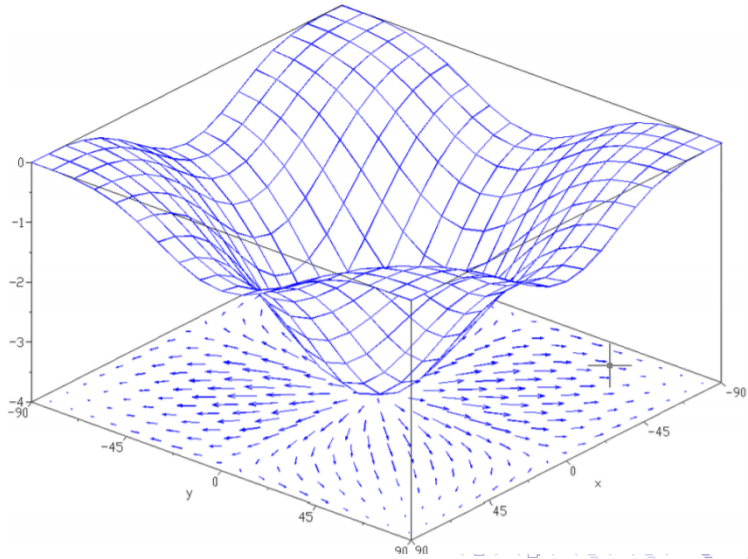
Gradijentni spust

- Šta se dešava kada je α malo?



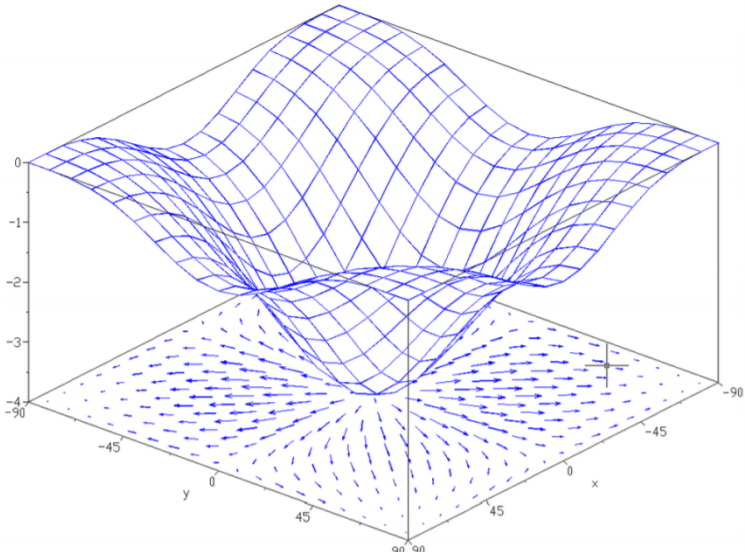
Gradijentni spust

- Šta se dešava kada je α malo?
  - Krećemo se malim koracima



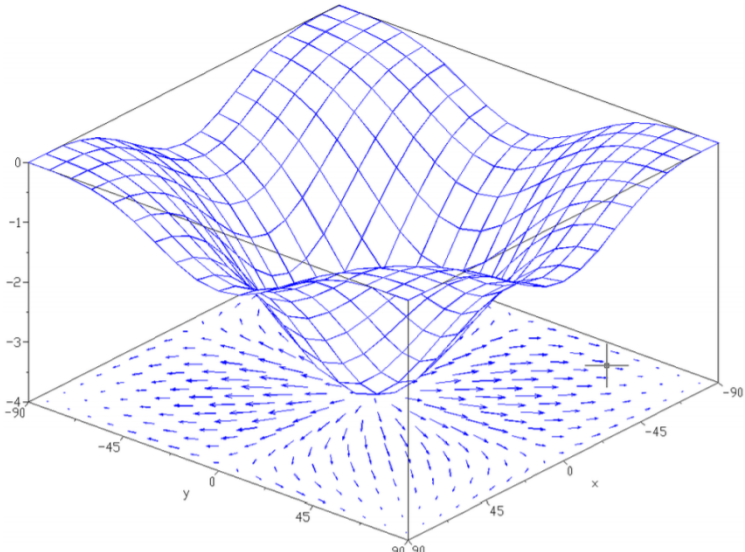
Gradijentni spust

- Šta se dešava kada je α malo?
  - Krećemo se malim koracima
  - Sporo ćemo konvergirati do minimuma



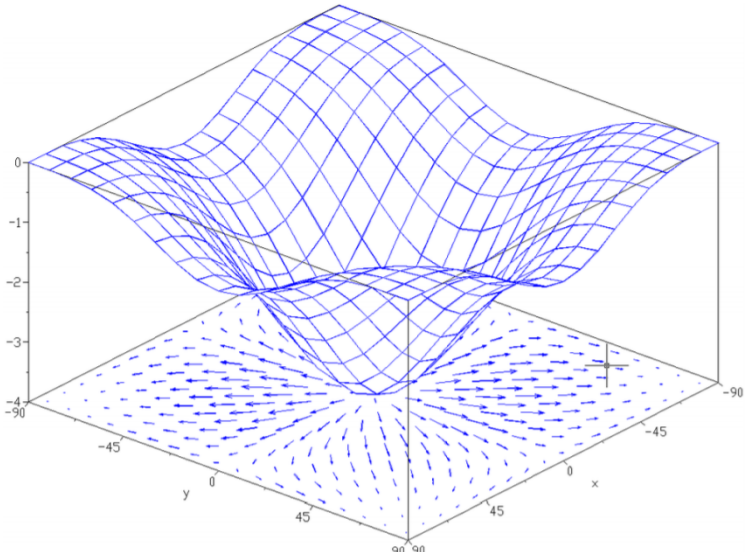
Gradijentni spust

- Šta se dešava kada je α malo?
  - Krećemo se malim koracima
  - Sporo ćemo konvergirati do minimuma
- Šta se dešava kada je α veliko?



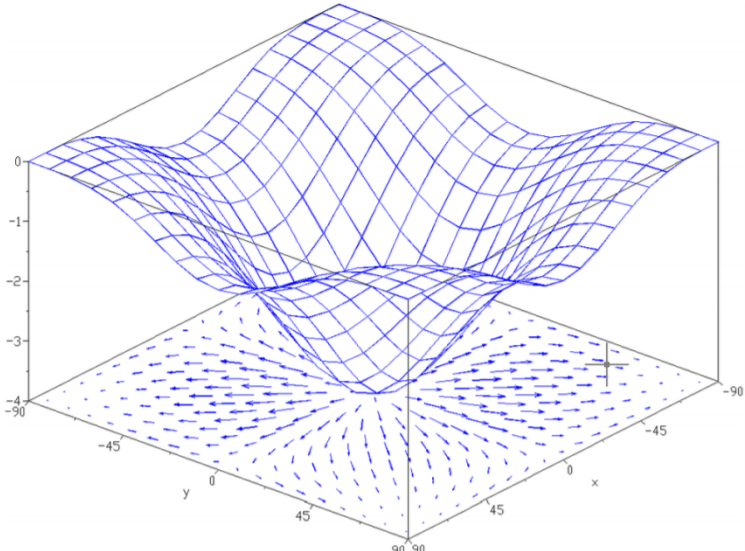
Gradijentni spust

- Šta se dešava kada je α malo?
  - Krećemo se malim koracima
  - Sporo ćemo konvergirati do minimuma
- Šta se dešava kada je α veliko?
  - Krećemo se velikim koracima



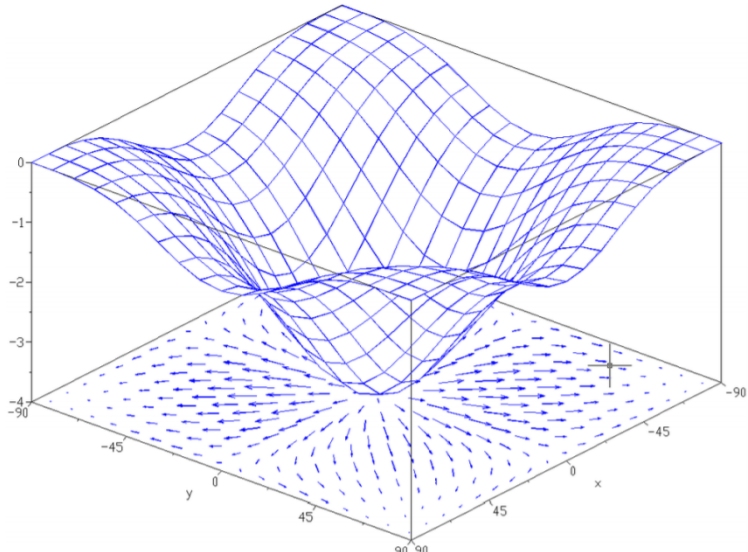
Gradijentni spust

- Šta se dešava kada je α malo?
  - Krećemo se malim koracima
  - Sporo ćemo konvergirati do minimuma
- Šta se dešava kada je α veliko?
  - Krećemo se velikim koracima
  - Može se desiti da preskočimo minimum



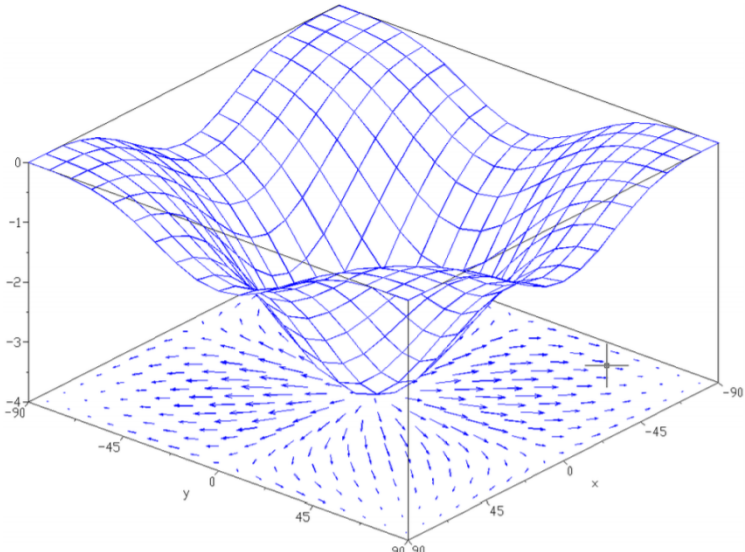
Gradijentni spust

- Kako da izaberemo α?



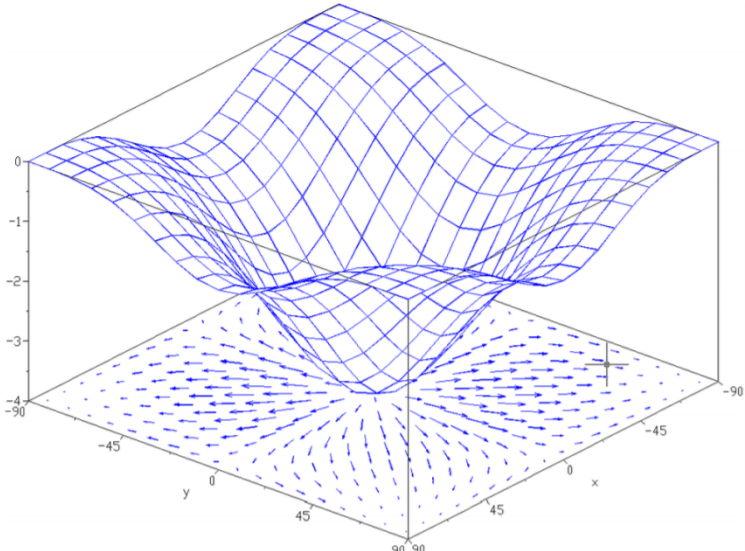
Gradijentni spust

- Kako da izaberemo α?
- Ako se izabere konstantno α, može se desiti da se približimo minimuma i da onda osciliramo oko te tačke



Gradijentni spust

- Kako da izaberemo α?
- Ako se izabere konstantno α, može se desiti da se približimo minimuma i da onda osciliramo oko te tačke
- Drugi pristup je da se α smanjuje i za to postoje različite strategije



Gradijentni spust

- Karakteristike parametara α koji garantuju konvergenciju dati su Robins-Monroovim uslovima:

$$\sum_{k=1}^{\infty} \alpha_k = \infty \qquad \sum_{k=1}^{\infty} \alpha_k^2 < \infty$$

Gradijentni spust

- Karakteristike parametara α koji garantuju konvergenciju dati su Robins-Monroovim uslovima:

$$\sum_{k=1}^{\infty} \alpha_k = \infty \qquad \sum_{k=1}^{\infty} \alpha_k^2 < \infty$$

- Suma koeficijenata treba da divergira

Gradijentni spust

- Karakteristike parametara α koji garantuju konvergenciju dati su Robins-Monroovim uslovima:

$$\sum_{k=1}^{\infty} \alpha_k = \infty \qquad \sum_{k=1}^{\infty} \alpha_k^2 < \infty$$

- Suma koeficijenata treba da divergira
  - koraci nisu previše mali već su dovoljno veliki da se negde odmaknemo

Gradijentni spust

- Karakteristike parametara α koji garantuju konvergenciju dati su Robins-Monroovim uslovima:

$$\sum_{k=1}^{\infty} \alpha_k = \infty \qquad \sum_{k=1}^{\infty} \alpha_k^2 < \infty$$

- Suma koeficijenata treba da divergira
  - koraci nisu previše mali već su dovoljno veliki da se negde odmaknemo
  - neće se desiti da brzo stanemo i zaustavimo se u optimizaciji i ni ne stignemo blizu minimuma

Gradijentni spust

- Karakteristike parametara α koji garantuju konvergenciju dati su Robins-Monroovim uslovima:

$$\sum_{k=1}^{\infty} \alpha_k = \infty \qquad \sum_{k=1}^{\infty} \alpha_k^2 < \infty$$

- Suma koeficijenata treba da divergira
  - koraci nisu previše mali već su dovoljno veliki da se negde odmaknemo
  - neće se desiti da brzo stanemo i zaustavimo se u optimizaciji i ni ne stignemo blizu minimuma
- Suma kvadrata koeficijenata treba da konvergira

Gradijentni spust

- Karakteristike parametara α koji garantuju konvergenciju dati su Robins-Monroovim uslovima:

$$\sum_{k=1}^{\infty} \alpha_k = \infty \qquad \sum_{k=1}^{\infty} \alpha_k^2 < \infty$$

- Suma koeficijenata treba da divergira
  - koraci nisu previše mali već su dovoljno veliki da se negde odmaknemo
  - neće se desiti da brzo stanemo i zaustavimo se u optimizaciji i ni ne stignemo blizu minimuma
- Suma kvadrata koeficijenata treba da konvergira
  - α će tipično da opada, teži nuli

Gradijentni spust

- Karakteristike parametara α koji garantuju konvergenciju dati su Robins-Monroovim uslovima:

$$\sum_{k=1}^{\infty} \alpha_k = \infty \qquad \sum_{k=1}^{\infty} \alpha_k^2 < \infty$$

- Suma koeficijenata treba da divergira
  - koraci nisu previše mali već su dovoljno veliki da se negde odmaknemo
  - neće se desiti da brzo stanemo i zaustavimo se u optimizaciji i ni ne stignemo blizu minimuma
- Suma kvadrata koeficijenata treba da konvergira
  - α će tipično da opada, teži nuli
  - u jednom trenutku će postati < 1 pa će kvadrat biti još manji

Gradijentni spust

- Karakteristike parametara α koji garantuju konvergenciju dati su Robins-Monroovim uslovima:

$$\sum_{k=1}^{\infty} \alpha_k = \infty \qquad \sum_{k=1}^{\infty} \alpha_k^2 < \infty$$

- Suma koeficijenata treba da divergira
  - koraci nisu previše mali već su dovoljno veliki da se negde odmaknemo
  - neće se desiti da brzo stanemo i zaustavimo se u optimizaciji i ni ne stignemo blizu minimuma
- Suma kvadrata koeficijenata treba da konvergira
  - α će tipično da opada, teži nuli
  - u jednom trenutku će postati < 1 pa će kvadrat biti još manji
  - vrednosti su dovoljno male da α^2 ipak konvergira, a dovoljno velike da donekle dobacimo

Gradijentni spust

- Karakteristike parametara α koji garantuju konvergenciju dati su Robins-Monroovim uslovima:

$$\sum_{k=1}^{\infty} \alpha_k = \infty \qquad \sum_{k=1}^{\infty} \alpha_k^2 < \infty$$

- Suma koeficijenata treba da divergira
  - koraci nisu previše mali već su dovoljno veliki da se negde odmaknemo
  - neće se desiti da brzo stanemo i zaustavimo se u optimizaciji i ni ne stignemo blizu minimuma
- Suma kvadrata koeficijenata treba da konvergira
  - α će tipično da opada, teži nuli
  - u jednom trenutku će postati < 1 pa će kvadrat biti još manji
  - vrednosti su dovoljno male da α^2 ipak konvergira, a dovoljno velike da donekle dobacimo
  - vrednosti su dovoljno male da se u nekom razumnom vremenu zaustavimo kada smo već u okolini minimuma

Gradijentni spust

- Karakteristike parametara α koji garantuju konvergenciju dati su Robins-Monroovim uslovima:

$$\sum_{k=1}^{\infty} \alpha_k = \infty \qquad \sum_{k=1}^{\infty} \alpha_k^2 < \infty$$

- Suma koeficijenata treba da divergira
  - koraci nisu previše mali već su dovoljno veliki da se negde odmaknemo
  - neće se desiti da brzo stanemo i zaustavimo se u optimizaciji i ni ne stignemo blizu minimuma
- Suma kvadrata koeficijenata treba da konvergira
  - α će tipično da opada, teži nuli
  - u jednom trenutku će postati < 1 pa će kvadrat biti još manji
  - vrednosti su dovoljno male da α^2 ipak konvergira, a dovoljno velike da donekle dobacimo
  - vrednosti su dovoljno male da se u nekom razumnom vremenu zaustavimo kada smo već u okolini minimuma
- $\alpha_k = \frac{1}{k}$

Gradijentni spust

- Kada se zaustavljamo?

Gradijentni spust

- Kada se zaustavljamo?
- Obično se definiše neka preciznost aproksimacije i kada je ona zadovoljena, mi stajemo (ε)

Gradijentni spust

- Kada se zaustavljamo?
- Obično se definiše neka preciznost aproksimacije i kada je ona zadovoljena, mi stajemo (ε)
- nakon unapred zadatog broja iteracija

Gradijentni spust

- Kada se zaustavljamo?
- Obično se definiše neka preciznost aproksimacije i kada je ona zadovoljena, mi stajemo (ε)
- nakon unapred zadatog broja iteracija
- nakon što razlika između susednih koraka $\|x_{k+1} - x_k\|$ postane manja od unapred zadate vrednosti ε

Gradijentni spust

- Kada se zaustavljamo?
- Obično se definiše neka preciznost aproksimacije i kada je ona zadovoljena, mi stajemo (ε)
- nakon unapred zadatog broja iteracija
- nakon što razlika između susednih koraka $\|x_{k+1} - x_k\|$ postane manja od unapred zadate vrednosti ε
- nakon što razlika između vrednosti funkcije u susednim koracima $|f(x_{k+1}) - f(x_k)|$ postane manja od ε

Gradijentni spust

- Kada se zaustavljamo?
- Obično se definiše neka preciznost aproksimacije i kada je ona zadovoljena, mi stajemo (ε)
- nakon unapred zadatog broja iteracija
- nakon što razlika između susednih koraka $\|x_{k+1} - x_k\|$ postane manja od unapred zadate vrednosti ε
- nakon što razlika između vrednosti funkcije u susednim koracima $|f(x_{k+1}) - f(x_k)|$ postane manja od ε
- nakon što ova razlika u odnosu na polaznu vrednost funkcije $|f(x_{k+1} - f(x_k)|/|f(x_0)|$ postane manja od ε

Gradijentni spust

- Kada se zaustavljamo?
- Obično se definiše neka preciznost aproksimacije i kada je ona zadovoljena, mi stajemo (ε)
- nakon unapred zadatog broja iteracija
- nakon što razlika između susednih koraka $\|x_{k+1} - x_k\|$ postane manja od unapred zadate vrednosti ε
- nakon što razlika između vrednosti funkcije u susednim koracima $|f(x_{k+1}) - f(x_k)|$ postane manja od ε
- nakon što ova razlika u odnosu na polaznu vrednost funkcije $|f(x_{k+1} - f(x_k)|/|f(x_0)|$ postane manja od ε
- Moguće je kombinovati i više ovakvih kriterijuma.

Gradijentni spust

- Koliko brzo ovaj algoritam konvergira?

Gradijentni spust

- Koliko brzo ovaj algoritam konvergira?
- Konstantan korak

Gradijentni spust

- Koliko brzo ovaj algoritam konvergira?
- Konstantan korak
- moguće je dokazati konvergenciju metoda ka pravom rešenju uz nesavladivu grešku

Gradijentni spust

- Koliko brzo ovaj algoritam konvergira?
- Konstantan korak
- moguće je dokazati konvergenciju metoda ka pravom rešenju uz nesavladivu grešku
- što je korak veći, do je i greška veća

Gradijentni spust

- Koliko brzo ovaj algoritam konvergira?

[1]Funkcija f je Lipšic neprekidna ako postoji konstanta L, takva da za svake dve tačke x i y važi $|f(x) - f(y)| \leq L\|x - y\|$.

Gradijentni spust

- Koliko brzo ovaj algoritam konvergira?
- Promenljiv korak koji ispunjava Robins-Monroove uslove

[1]Funkcija f je Lipšic neprekidna ako postoji konstanta L, takva da za svake dve tačke x i y važi $|f(x) - f(y)| \leq L\|x - y\|$.

Gradijentni spust

- Koliko brzo ovaj algoritam konvergira?
- Promenljiv korak koji ispunjava Robins-Monroove uslove
- pretpostavimo da je funkcija koju minimizujemo konveksna i da ima Lipšic neprekidni gradijent[1]

[1]Funkcija f je Lipšic neprekidna ako postoji konstanta L, takva da za svake dve tačke x i y važi $|f(x) - f(y)| \leq L\|x - y\|$.

Gradijentni spust

- Koliko brzo ovaj algoritam konvergira?
- Promenljiv korak koji ispunjava Robins-Monroove uslove
- pretpostavimo da je funkcija koju minimizujemo konveksna i da ima Lipšic neprekidni gradijent[1]
- tada se može pokazati da je greška metode $\|x_k - x^*\|$ u koraku k, gde je x^* tačka minimuma, reda $O\left(\frac{1}{k}\right)$

[1]Funkcija f je Lipšic neprekidna ako postoji konstanta L, takva da za svake dve tačke x i y važi $|f(x) - f(y)| \leq L\|x - y\|$.

Gradijentni spust

- Koliko brzo ovaj algoritam konvergira?

Gradijentni spust

- Koliko brzo ovaj algoritam konvergira?
- Promenljiv korak koji ispunjava Robins-Monroove uslove

Gradijentni spust

- Koliko brzo ovaj algoritam konvergira?
- Promenljiv korak koji ispunjava Robins-Monroove uslove
- Pretpostavimo dodatno da je funkcija koju minimizujemo jako konveksna i da ima Lipšic neprekidni gradijent

Gradijentni spust

- Koliko brzo ovaj algoritam konvergira?
- Promenljiv korak koji ispunjava Robins-Monroove uslove
- Pretpostavimo dodatno da je funkcija koju minimizujemo jako konveksna i da ima Lipšic neprekidni gradijent
- tada se može pokazati da je greška metode $\|x_k - x^*\|$ u koraku k, gde je x^* tačka minimuma, reda $O\left(c^k\right)$ za neko $0 < c < 1$

Gradijentni spust

- Koliko brzo ovaj algoritam konvergira?

Gradijentni spust

- Koliko brzo ovaj algoritam konvergira?
- Promenljiv korak koji ispunjava Robins-Monroove uslove

Gradijentni spust

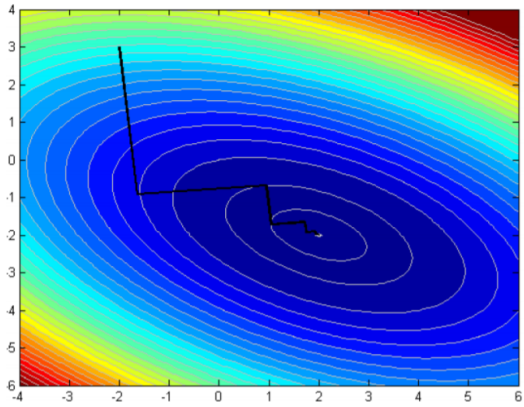
- Koliko brzo ovaj algoritam konvergira?
- Promenljiv korak koji ispunjava Robins-Monroove uslove
- pretpostavimo da funkcija koju minimizujemo nije konveksna i da ima Lipšic neprekidni gradijent

Gradijentni spust

- Koliko brzo ovaj algoritam konvergira?
- Promenljiv korak koji ispunjava Robins-Monroove uslove
- pretpostavimo da funkcija koju minimizujemo nije konveksna i da ima Lipšic neprekidni gradijent
- tada se može pokazati da gradijentni spust i njegove varijante konvergiraju, ali navedene brzine konvergencije ne važe

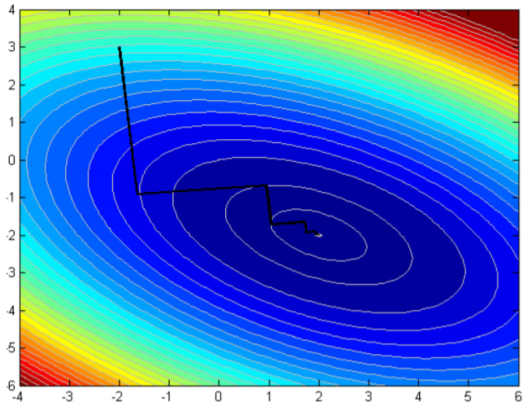
Geometrija gradijentnog spusta

- Posmatrajmo grafik funkcije dve promenljive



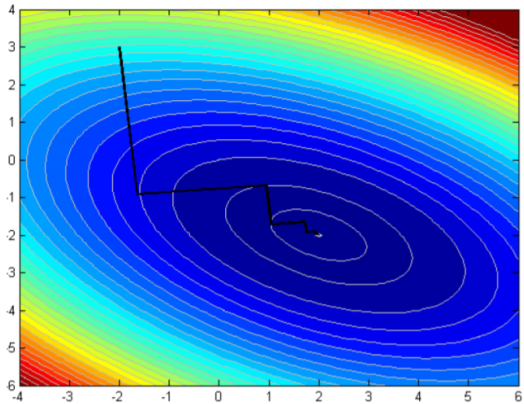
Geometrija gradijentnog spusta

- Posmatrajmo grafik funkcije dve promenljive
- Tamno plava je najmanja vrednost, tamno crvena je najveća vrednost



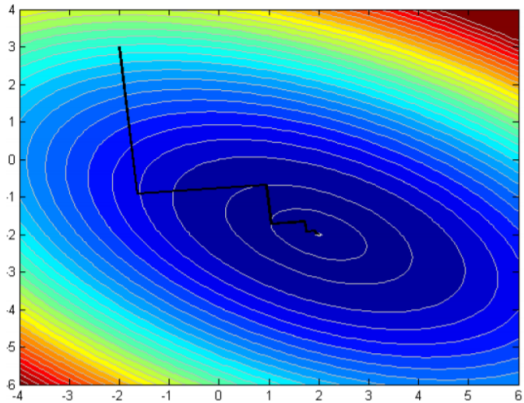
Geometrija gradijentnog spusta

- Posmatrajmo grafik funkcije dve promenljive
- Tamno plava je najmanja vrednost, tamno crvena je najveća vrednost
- Gradijentni spust se kreće nacrtanom trajektorijom dok ne dođe do minimuma



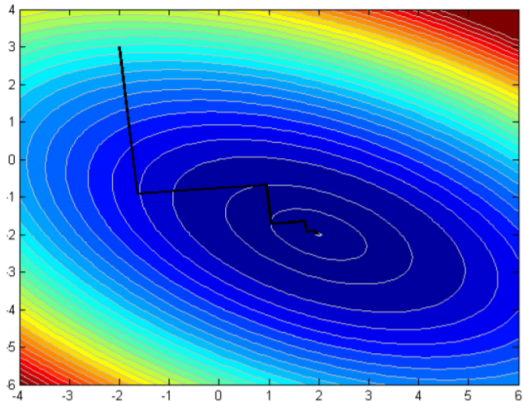
Geometrija gradijentnog spusta

- Pravac gradijenta u nekoj tački je normalan na konturu funkcije u toj tački



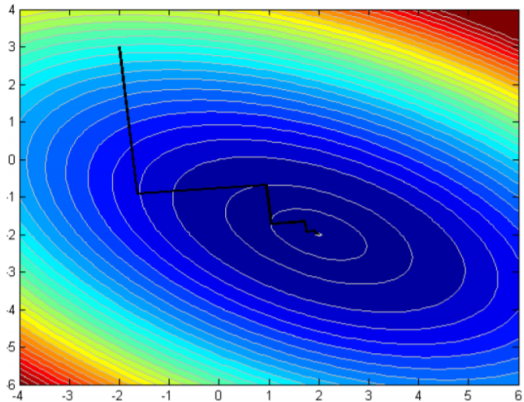
Geometrija gradijentnog spusta

- Pravac gradijenta u nekoj tački je normalan na konturu funkcije u toj tački
- Kontura je skup svih tačaka koje imaju istu vrednost



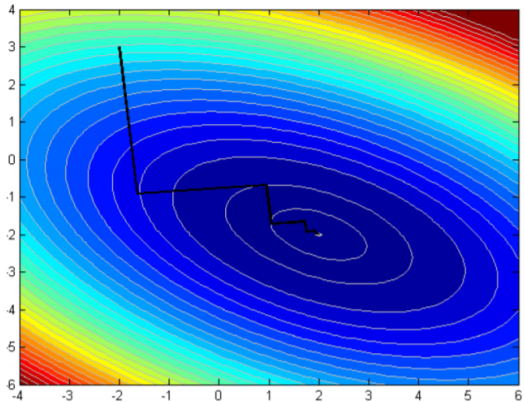
Geometrija gradijentnog spusta

- Pravac gradijenta u nekoj tački je normalan na konturu funkcije u toj tački
- Kontura je skup svih tačaka koje imaju istu vrednost
- Na posmatranom grafiku, za tačku koja se nalazi na elipsi, njenu konturu čine sve tačke na toj elipsi



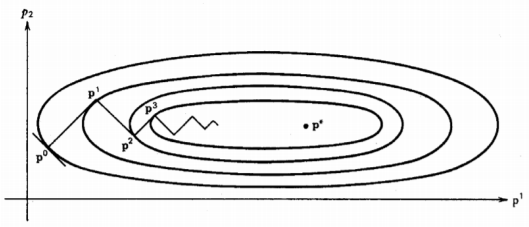
Geometrija gradijentnog spusta

- Pravac gradijenta u nekoj tački je normalan na konturu funkcije u toj tački
- Kontura je skup svih tačaka koje imaju istu vrednost
- Na posmatranom grafiku, za tačku koja se nalazi na elipsi, njenu konturu čine sve tačke na toj elipsi
- U slučaju kakvih funkcija će se na osnovu ovog zapažanja gradijentni spust ponašati lepo, tj. brzo će konvergirati, a kada suprotno?



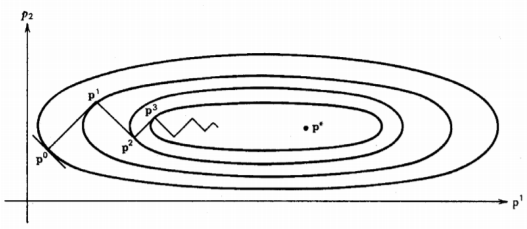
Geometrija gradijentnog spusta

- Pokazuje se da gradijentni spust sporije konvergira kada su konture u obliku jako izduženih elipsa (to se dešava kada su podaci jako korelirani)



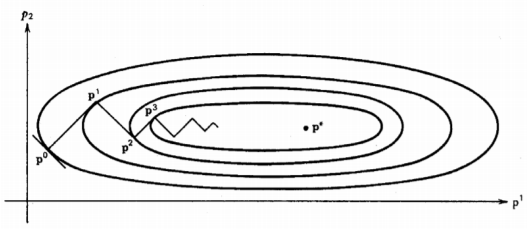
Geometrija gradijentnog spusta

- Pokazuje se da gradijentni spust sporije konvergira kada su konture u obliku jako izduženih elipsa (to se dešava kada su podaci jako korelirani)
- U takvim situacijama, gradijentni spust bira tačke koje leže duž cik-cak putanje ka minimumu i broj koraka do zadovoljavajućeg rešenja može biti veliki



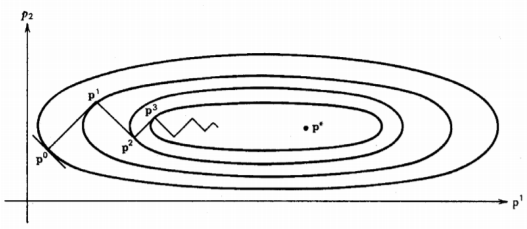
Geometrija gradijentnog spusta

- Pokazuje se da gradijentni spust sporije konvergira kada su konture u obliku jako izduženih elipsa (to se dešava kada su podaci jako korelirani)
- U takvim situacijama, gradijentni spust bira tačke koje leže duž cik-cak putanje ka minimumu i broj koraka do zadovoljavajućeg rešenja može biti veliki
- Očito, pravac najbržeg spusta uopšte ne mora biti pravac najbržeg kretanja ka minimumu



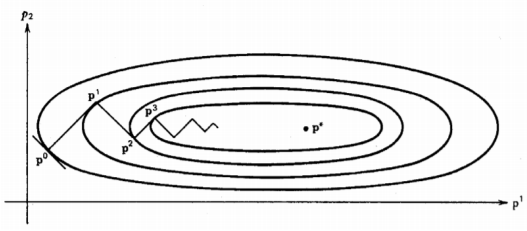
Geometrija gradijentnog spusta

- Pokazuje se da gradijentni spust sporije konvergira kada su konture u obliku jako izduženih elipsa (to se dešava kada su podaci jako korelirani)
- U takvim situacijama, gradijentni spust bira tačke koje leže duž cik-cak putanje ka minimumu i broj koraka do zadovoljavajućeg rešenja može biti veliki
- Očito, pravac najbržeg spusta uopšte ne mora biti pravac najbržeg kretanja ka minimumu
- Možemo potrošiti značajno vreme za računanje gradijenta koji je samo lokalno optimalan



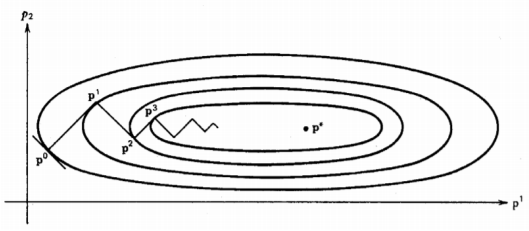
Geometrija gradijentnog spusta

- Gradijent jeste pravac najbržeg spusta ali taj pravac nije direktan pravac ka minimumu



Geometrija gradijentnog spusta

- Gradijent jeste pravac najbržeg spusta ali taj pravac nije direktan pravac ka minimumu
- Cik-cak kretanje je velika mana



Pregled

Metod inercije

- Postoji način da se cik-cak kretanje ublaži
- Do cik-cak kretanja dolazi usled toga što se pravac gradijenta menja naglo a u gradijentnom spustu se strogo držimo pravca gradijenta
- Videli smo da pravac gradijenta često i nije tako dobar pravac pa i nema potrebe da ga se tako strogo držimo
- S druge strane, pošto gradijent naglo menja pravac, možda bismo mogli da, odstupivši od njega, nekako konstruišemo novi pravac koji ne osciluje tako brzo

Metod inercije

- Kako možemo sprečiti brzo oscilovanje vrednosti?
- Ako posmatramo veliki broj uzoraka neke slučajne promenljive, tada možemo imati veliku varijansu, ali ako posmatramo njihove proseke, tada nemamo
- Mogli bismo da gradijentne pravce nekako uprosečujemo
- *Metod inercije* se zasniva na ideji akumuliranja prethodnih gradijenata, pri čemu je značaj starijih gradijenata manji, a novijih veći,
- Umesto gradijenta u datoj tački koristi ukupan akumulirani gradijent.
- Kako prosek nekih vrednosti manje varira nego same vrednosti, ovakva tehnika dovodi do manjih promena pravca u gradijentu i često do povećanja brzine konvergencije.

Metod inercije

- pravac u kom se krećemo je pravac d

$$d_0 = 0$$

$$d_{k+1} = \beta_k d_k + \alpha_k \nabla f(x_k)$$

$$x_{k+1} = x_k - d_{k+1}$$

pri čemu važi $0 \leq \beta_k < 1$

Metod inercije

- pravac u kom se krećemo je pravac d
- kako izračunavamo d?
- na početku ga inicijalizujemo na nulu

$$d_0 = 0$$

$$d_{k+1} = \beta_k d_k + \alpha_k \nabla f(x_k)$$

$$x_{k+1} = x_k - d_{k+1}$$

pri čemu važi $0 \leq \beta_k < 1$

Metod inercije

- pravac u kom se krećemo je pravac d
- kako izračunavamo d?
- na početku ga inicijalizujemo na nulu
- u svakom sledećem koraku, na pravac kojim smo se kretali u prethodnom koraku dodajemo gradijent
- taj pravac otežavamo nekim koeficijentom β a uz gradijent kao i ranije imamo koeficijent α
- tako će svako d_{k+1} biti suma gradijenata u prethodnim tačkama $\nabla f(x_1), \ldots, \nabla f(x_k)$ koji su otežani koeficijentima α i β

$$d_0 = 0$$

$$d_{k+1} = \beta_k d_k + \alpha_k \nabla f(x_k)$$

$$x_{k+1} = x_k - d_{k+1}$$

pri čemu važi $0 \leq \beta_k < 1$

Metod inercije

- tako će svako d_{k+1} biti suma gradijenata u prethodnim tačkama $\nabla f(x_1), \ldots, \nabla f(x_k)$ koji su otežani koeficijentima α i β
- pritom, $\nabla f(x_k)$ će biti otežano sa α_k, $\nabla f(x_{k-1})$ će biti otežano sa $\alpha_{k-1}\beta_k$, $\nabla f(x_{k-2})$ će biti otežano sa $\alpha_{k-2}\beta_{k-1}\beta_k, \ldots$
- za β tražimo da je strogo manje od 1 što čini da doprinosi starih gradijenata eksponencijalno opadaju

$$d_0 = 0$$

$$d_{k+1} = \beta_k d_k + \alpha_k \nabla f(x_k)$$

$$x_{k+1} = x_k - d_{k+1}$$

pri čemu važi $0 \leq \beta_k < 1$

Metod inercije

- tako imamo eksponencijalno opadajuće doprinose starih gradijenata tako da oni posle nekog vremena postaju potpuno nebitni ali poslednjih nekoliko gradijenata još ima neku težinu i ne dozvoljava da novi pravac tako lako odstupi od njih
- osnovni princip inercije je da ćemo nekako usrednjavati sve gradijente ali tako da su stari manje bitni a da su noviji bitniji i onda da se ne krećemo uvek u pravcu gradijenta nego u nekom novom pravcu koji usrednjava te gradijente

$$d_0 = 0$$

$$d_{k+1} = \beta_k d_k + \alpha_k \nabla f(x_k)$$

$$x_{k+1} = x_k - d_{k+1}$$

pri čemu važi $0 \leq \beta_k < 1$

Pregled

Nestorovljev ubrzani gradijentni spust

- zanimljiv je po tome što je asimptotski optimalan algoritam prvog reda

Nestorovljev ubrzani gradijentni spust

- zanimljiv je po tome što je asimptotski optimalan algoritam prvog reda
- može se pokazati da je kod konveksnih funkcija sa Lipšic neprekidnim gradijentom njegova greška reda $O\left(\frac{1}{k^2}\right)$, naspram $O\left(\frac{1}{k}\right)$ u slučaju običnog gradijentnog spusta

Nestorovljev ubrzani gradijentni spust

- zanimljiv je po tome što je asimptotski optimalan algoritam prvog reda
- može se pokazati da je kod konveksnih funkcija sa Lipšic neprekidnim gradijentom njegova greška reda $O\left(\frac{1}{k^2}\right)$, naspram $O\left(\frac{1}{k}\right)$ u slučaju običnog gradijentnog spusta
- ako imamo informacije samo o prvim izvodima (nemamo druge izvode, što može biti problematično kod podataka visoke dimenzionalnosti), onda za konveksne funkcije ne može brže od $O\left(\frac{1}{k^2}\right)$

Nestorovljev ubrzani gradijentni spust

- ova modifikacija je potekla iz analitičkog razmatranja i nema laku geometrijsku interpretaciju,

$$d_0 = 0$$
$$d_{k+1} = \beta_k d_k + \alpha_k \nabla f(x_k - \beta_k d_k)$$
$$x_{k+1} = x_k - d_{k+1}$$

Nestorovljev ubrzani gradijentni spust

- ova modifikacija je potekla iz analitičkog razmatranja i nema laku geometrijsku interpretaciju,
- ne računamo gradijent u tački u kojoj smo sada vec u onoj tački u kojoj bismo bili da smo produžili još malo u pravcu u kom smo se do sada kretali

$$d_0 = 0$$
$$d_{k+1} = \beta_k d_k + \alpha_k \nabla f(x_k - \beta_k d_k)$$
$$x_{k+1} = x_k - d_{k+1}$$

Nestorovljev ubrzani gradijentni spust

- ova modifikacija je potekla iz analitičkog razmatranja i nema laku geometrijsku interpretaciju,
- ne računamo gradijent u tački u kojoj smo sada vec u onoj tački u kojoj bismo bili da smo produžili još malo u pravcu u kom smo se do sada kretali
- ova modifikacija čini da algoritam asimptotski brže konvergira

$$d_0 = 0$$
$$d_{k+1} = \beta_k d_k + \alpha_k \nabla f(x_k - \beta_k d_k)$$
$$x_{k+1} = x_k - d_{k+1}$$

Pregled

Adam

- Adam (eng.*adaptive moment estimation*)

Adam

- Adam (eng.*adaptive moment estimation*)
- jedan od algoritama sa prilagodljivom dužinom koraka

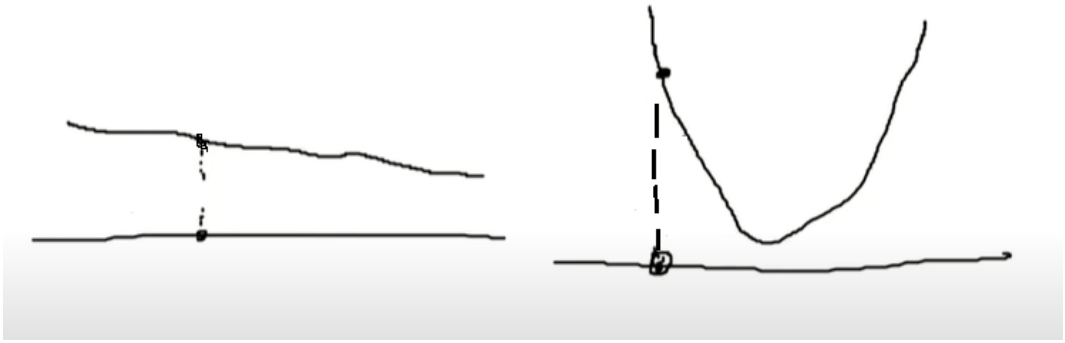
Adam

- Adam (eng.*adaptive moment estimation*)
- jedan od algoritama sa prilagodljivom dužinom koraka
- u dosadašnjim metodama, definišemo kako se parametar α menja i ako smo to loše definisali, nema kompenzacije, sporije će konvergirati

Adam

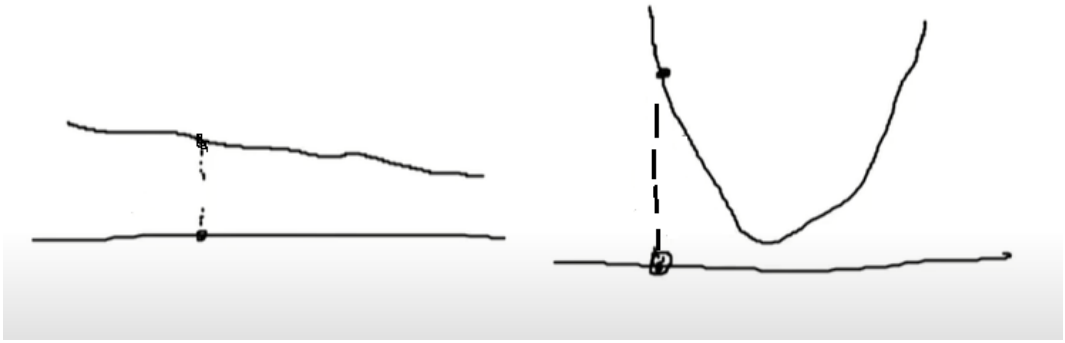
- Adam (eng.*adaptive moment estimation*)
- jedan od algoritama sa prilagodljivom dužinom koraka
- u dosadašnjim metodama, definišemo kako se parametar α menja i ako smo to loše definisali, nema kompenzacije, sporije će konvergirati
- ideja Adama je da ipak proba da automatski koriguje dužinu svog koraka prema delu funkcije u kom se trenutno nalazi

Adam



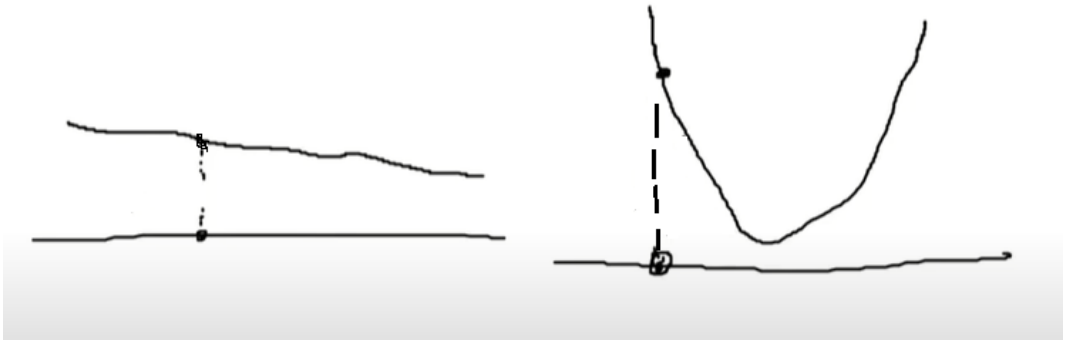
- Posmatrajmo dve funkcije različitih nagiba

Adam



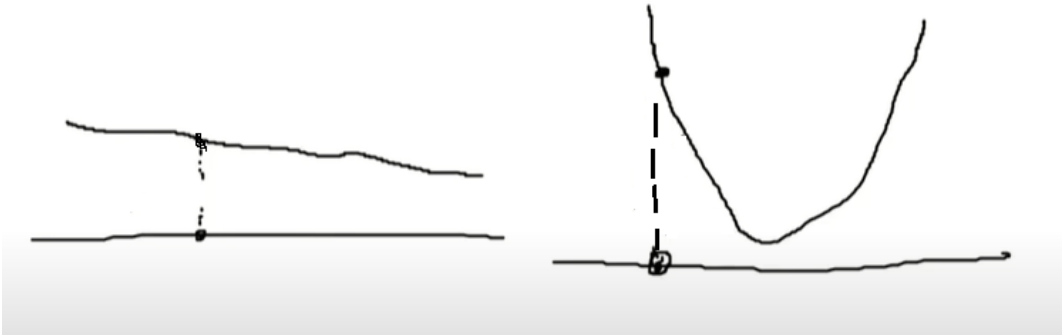
- Posmatrajmo dve funkcije različitih nagiba
- U svakoj od njih imamo označenu jednu tačku iz koje počinjemo da tražimo minimum

Adam



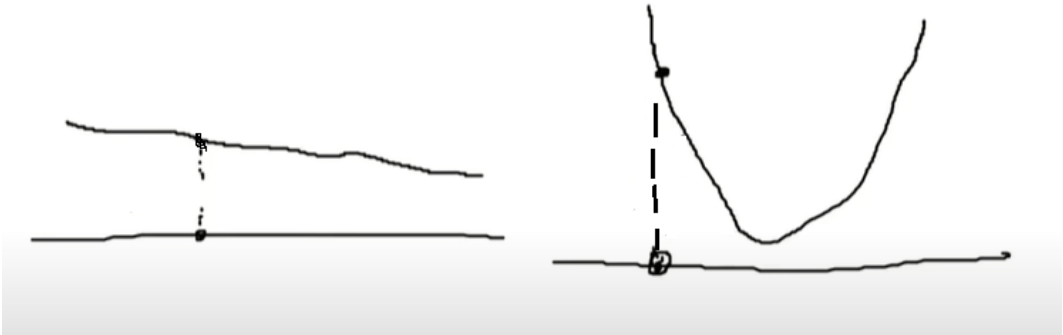
- Posmatrajmo dve funkcije različitih nagiba
- U svakoj od njih imamo označenu jednu tačku iz koje počinjemo da tražimo minimum
- Kakav korak (koliko veliki) treba da bude u prvom a kakav u drugom slučaju?

Adam



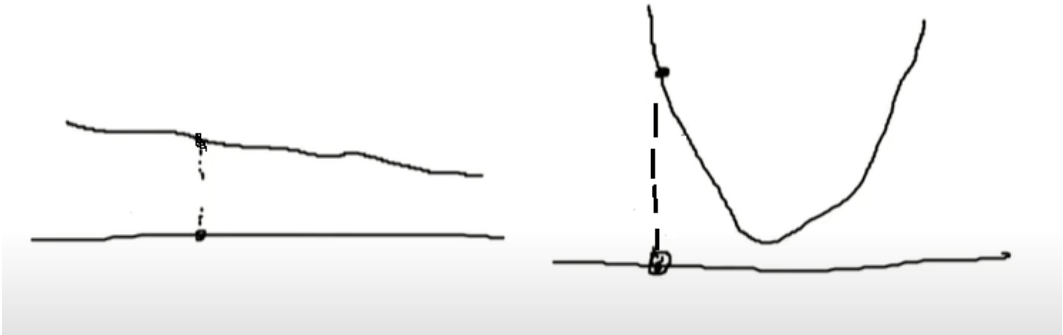
- Desno

Adam



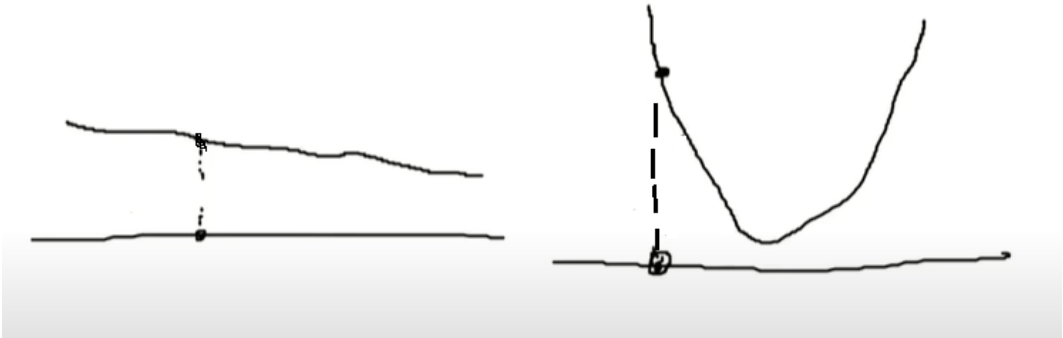
- Desno
- treba nam manji korak da ne bismo preskočili minimum

Adam



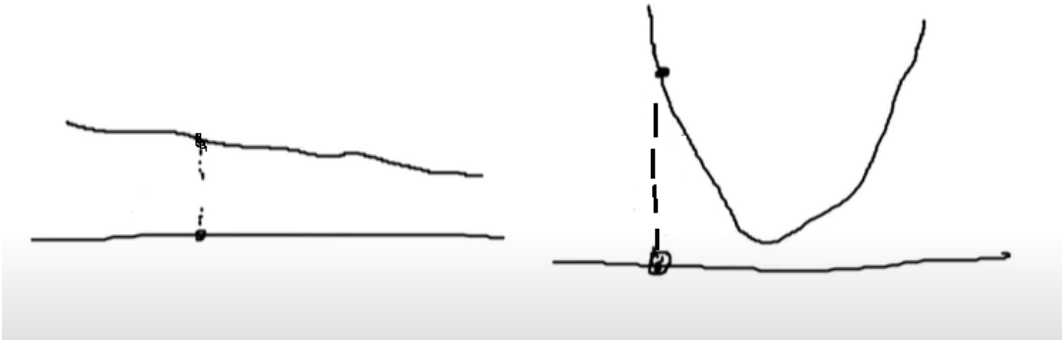
- Desno
- treba nam manji korak da ne bismo preskočili minimum
- kad je nagib veliki, gradijent je po normi gradijent veliki a minimum nam je blizu,

Adam



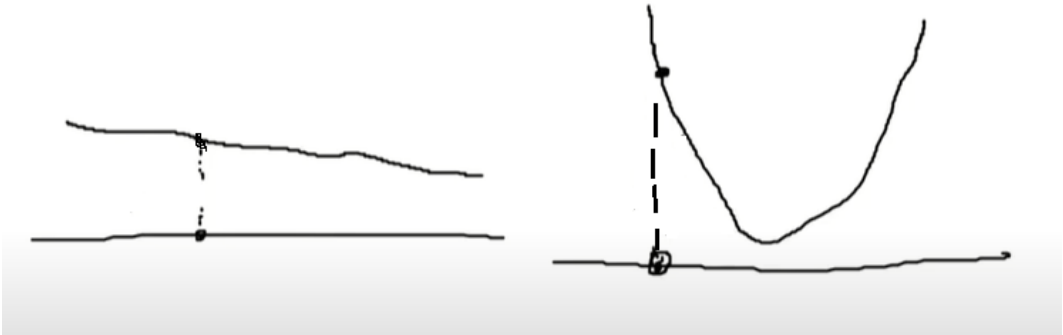
- Desno
- treba nam manji korak da ne bismo preskočili minimum
- kad je nagib veliki, gradijent je po normi gradijent veliki a minimum nam je blizu,
- ako krenemo da se krecemo proporcionalno tekucem gradijentu ici cemo jako brzo i možda ćemo preskočiti minimum

Adam



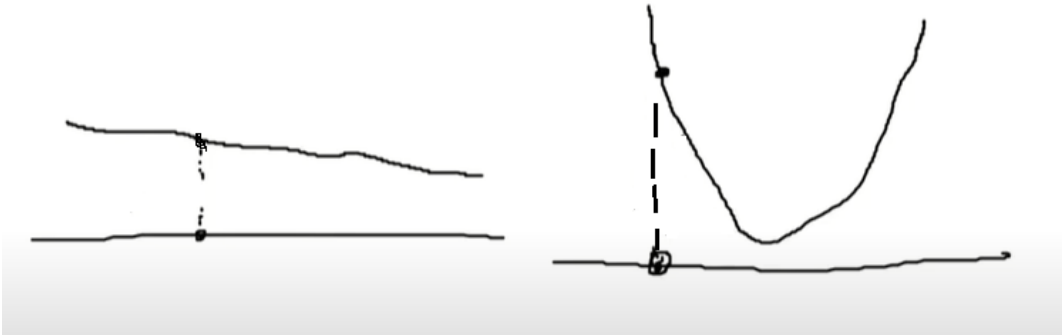
- Levo

Adam



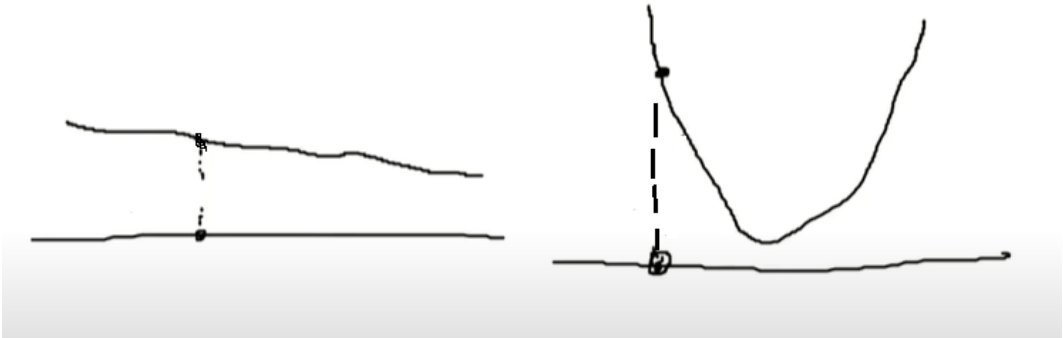
- Levo
- treba nam veći korak jer je nagib mali

Adam



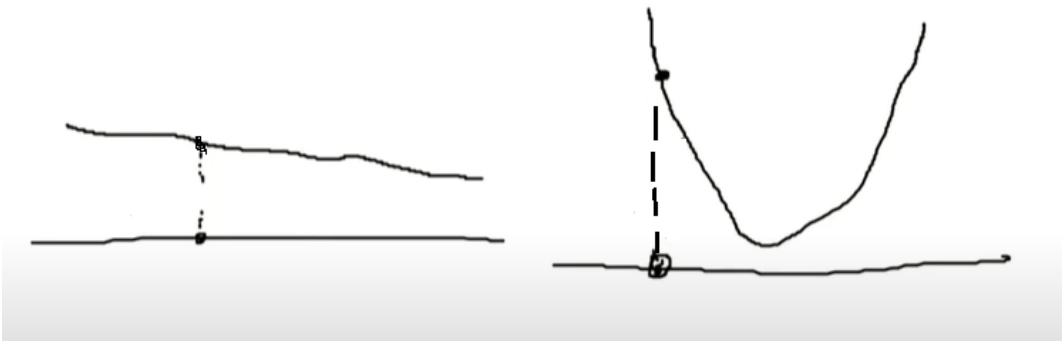
- Levo
- treba nam veći korak jer je nagib mali
- kad je nagib mali, gradijent je mali, blizu nule

Adam



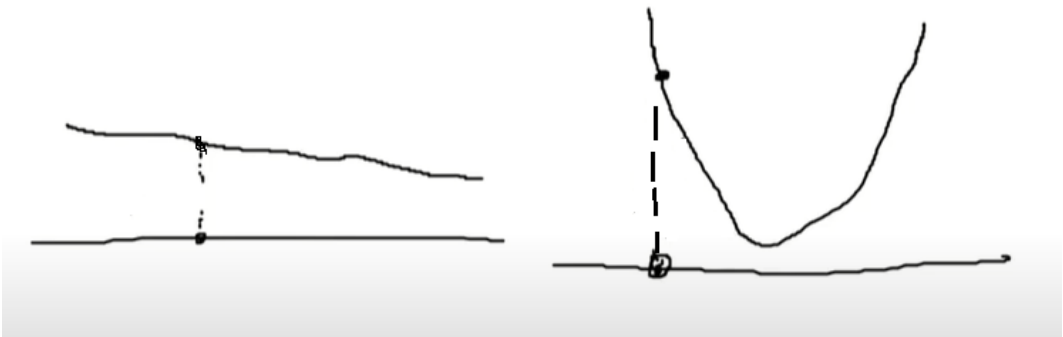
- Levo
- treba nam veći korak jer je nagib mali
- kad je nagib mali, gradijent je mali, blizu nule
- ako krenemo da se krecemo proporcionalno tekucem gradijentu ići cemo jako sporo, a deluje da minimum nije blizu, daleko je a mi idemo malim koracima ako pratimo gradijent

Adam



- Dužina gradijenta u ovim slučajevima je upravo suprotna od onog što bi trebalo da bude

Adam



- Dužina gradijenta u ovim slučajevima je upravo suprotna od onog što bi trebalo da bude
- Ako je tako dosledna, možda to svojstvo možemo upotrebiti

Adam

- imamo dve pomoćne promenljive m i v

$$m_0 = 0$$

$$v_0 = 0$$

$$m_{k+1} = \beta_1 m_k + (1 - \beta_1)\nabla f(x_k)$$

$$v_{k+1} = \beta_2 v_k + (1 - \beta_2)\nabla f(x_k) \odot \nabla f(x_k)$$

$$x_{k+1} = x_k - \alpha_{k+1} \frac{\widehat{m}_{k+1}}{\sqrt{\widehat{v}_{k+1}} + \varepsilon}$$

Adam

- imamo dve pomoćne promenljive m i v
- m je analogon prethodno pomenutog pravca kretanja d

$$m_0 = 0$$

$$v_0 = 0$$

$$m_{k+1} = \beta_1 m_k + (1 - \beta_1)\nabla f(x_k)$$

$$v_{k+1} = \beta_2 v_k + (1 - \beta_2)\nabla f(x_k) \odot \nabla f(x_k)$$

$$x_{k+1} = x_k - \alpha_{k+1} \frac{\widehat{m}_{k+1}}{\sqrt{\widehat{v}_{k+1}} + \varepsilon}$$

Adam

- imamo dve pomoćne promenljive m i v
- m je analogon prethodno pomenutog pravca kretanja d
- m računamo vrlo slično kao d - usrednjavaju se gradijenti zajedno sa novim gradijentom koji im se pridružuje

$$m_0 = 0$$

$$v_0 = 0$$

$$m_{k+1} = \beta_1 m_k + (1 - \beta_1)\nabla f(x_k)$$

$$v_{k+1} = \beta_2 v_k + (1 - \beta_2)\nabla f(x_k) \odot \nabla f(x_k)$$

$$x_{k+1} = x_k - \alpha_{k+1} \frac{\widehat{m}_{k+1}}{\sqrt{\widehat{v}_{k+1}} + \varepsilon}$$

Adam

- imamo dve pomoćne promenljive m i v
- m je analogon prethodno pomenutog pravca kretanja d
- m računamo vrlo slično kao d - usrednjavaju se gradijenti zajedno sa novim gradijentom koji im se pridružuje
- to je pravac u kojem cemo se kretati

$$m_0 = 0$$

$$v_0 = 0$$

$$m_{k+1} = \beta_1 m_k + (1 - \beta_1)\nabla f(x_k)$$

$$v_{k+1} = \beta_2 v_k + (1 - \beta_2)\nabla f(x_k) \odot \nabla f(x_k)$$

$$x_{k+1} = x_k - \alpha_{k+1} \frac{\widehat{m}_{k+1}}{\sqrt{\widehat{v}_{k+1}} + \varepsilon}$$

Adam

- prilikom računanja v, ponovo se akumuliraju gradijenti

$$m_0 = 0$$

$$v_0 = 0$$

$$m_{k+1} = \beta_1 m_k + (1 - \beta_1)\nabla f(x_k)$$

$$v_{k+1} = \beta_2 v_k + (1 - \beta_2)\nabla f(x_k) \odot \nabla f(x_k)$$

$$x_{k+1} = x_k - \alpha_{k+1} \frac{\widehat{m}_{k+1}}{\sqrt{\widehat{v}_{k+1}} + \varepsilon}$$

Adam

- prilikom računanja v, ponovo se akumuliraju gradijenti
- operacija $\odot$ predstavlja pokoordinatno množenje vektora u kojoj kao rezultat dobijamo vektor sa kvadriranim koordinatama gradijenta

$$m_0 = 0$$

$$v_0 = 0$$

$$m_{k+1} = \beta_1 m_k + (1 - \beta_1)\nabla f(x_k)$$

$$v_{k+1} = \beta_2 v_k + (1 - \beta_2)\nabla f(x_k) \odot \nabla f(x_k)$$

$$x_{k+1} = x_k - \alpha_{k+1} \frac{\widehat{m}_{k+1}}{\sqrt{\widehat{v}_{k+1}} + \varepsilon}$$

Adam

- prilikom računanja v, ponovo se akumuliraju gradijenti
- operacija $\odot$ predstavlja pokoordinatno množenje vektora u kojoj kao rezultat dobijamo vektor sa kvadriranim koordinatama gradijenta
- ako je gradijent bio veliki po normi, dobijeni vektor ce biti jos veći po normi i obrnuto

$$m_0 = 0$$

$$v_0 = 0$$

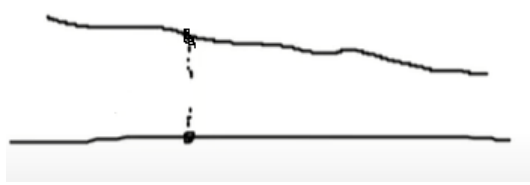
$$m_{k+1} = \beta_1 m_k + (1 - \beta_1)\nabla f(x_k)$$

$$v_{k+1} = \beta_2 v_k + (1 - \beta_2)\nabla f(x_k) \odot \nabla f(x_k)$$

$$x_{k+1} = x_k - \alpha_{k+1} \frac{\widehat{m}_{k+1}}{\sqrt{\widehat{v}_{k+1}} + \varepsilon}$$

Adam

- Analizirajmo količnik m i v



$$m_0 = 0$$

$$v_0 = 0$$

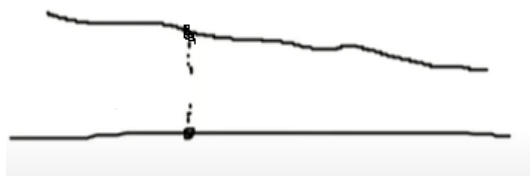
$$m_{k+1} = \beta_1 m_k + (1 - \beta_1)\nabla f(x_k)$$

$$v_{k+1} = \beta_2 v_k + (1 - \beta_2)\nabla f(x_k) \odot \nabla f(x_k)$$

$$x_{k+1} = x_k - \alpha_{k+1} \frac{\widehat{m}_{k+1}}{\sqrt{\widehat{v}_{k+1}} + \varepsilon}$$

Adam

- Analizirajmo količnik m i v
- kada je nagib mali, gradijent nam stalno pokazuje u istom pravcu i pritom je mali po normi



$$m_0 = 0$$

$$v_0 = 0$$

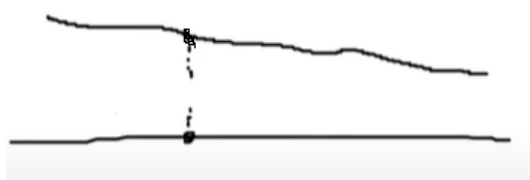
$$m_{k+1} = \beta_1 m_k + (1 - \beta_1)\nabla f(x_k)$$

$$v_{k+1} = \beta_2 v_k + (1 - \beta_2)\nabla f(x_k) \odot \nabla f(x_k)$$

$$x_{k+1} = x_k - \alpha_{k+1} \frac{\widehat{m}_{k+1}}{\sqrt{\widehat{v}_{k+1}} + \varepsilon}$$

Adam

- Analizirajmo količnik m i v
- kada je nagib mali, gradijent nam stalno pokazuje u istom pravcu i pritom je mali po normi
- pošto pokazuje stalno u istom pravcu, m se neće razlikovati mnogo od tog gradijenta



$$m_0 = 0$$

$$v_0 = 0$$

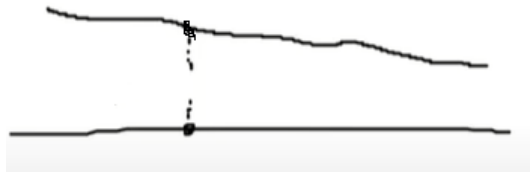
$$m_{k+1} = \beta_1 m_k + (1 - \beta_1)\nabla f(x_k)$$

$$v_{k+1} = \beta_2 v_k + (1 - \beta_2)\nabla f(x_k) \odot \nabla f(x_k)$$

$$x_{k+1} = x_k - \alpha_{k+1} \frac{\widehat{m}_{k+1}}{\sqrt{\widehat{v}_{k+1}} + \varepsilon}$$

Adam

- Analizirajmo količnik m i v
- kada je nagib mali, gradijent nam stalno pokazuje u istom pravcu i pritom je mali po normi
- pošto pokazuje stalno u istom pravcu, m se neće razlikovati mnogo od tog gradijenta
- kod v, na maloj nizbrdici je gradijent mali po normi pa će kvadrat biti još manji



$$m_0 = 0$$

$$v_0 = 0$$

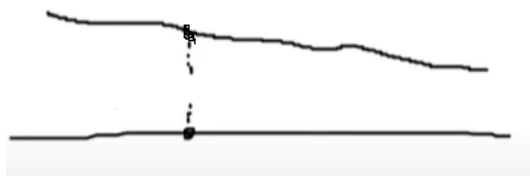
$$m_{k+1} = \beta_1 m_k + (1 - \beta_1)\nabla f(x_k)$$

$$v_{k+1} = \beta_2 v_k + (1 - \beta_2)\nabla f(x_k) \odot \nabla f(x_k)$$

$$x_{k+1} = x_k - \alpha_{k+1} \frac{\widehat{m}_{k+1}}{\sqrt{\widehat{v}_{k+1}} + \varepsilon}$$

Adam

- Analizirajmo količnik m i v
- kada je nagib mali, gradijent nam stalno pokazuje u istom pravcu i pritom je mali po normi
- pošto pokazuje stalno u istom pravcu, m se neće razlikovati mnogo od tog gradijenta
- kod v, na maloj nizbrdici je gradijent mali po normi pa će kvadrat biti još manji
- količnik m/v je vektor približno istog pravca kao m samo je duži jer smo delili malim brojem



$$m_0 = 0$$

$$v_0 = 0$$

$$m_{k+1} = \beta_1 m_k + (1 - \beta_1)\nabla f(x_k)$$

$$v_{k+1} = \beta_2 v_k + (1 - \beta_2)\nabla f(x_k) \odot \nabla f(x_k)$$

$$x_{k+1} = x_k - \alpha_{k+1} \frac{\widehat{m}_{k+1}}{\sqrt{\widehat{v}_{k+1}} + \varepsilon}$$

Adam

- Analizirajmo količnik m i v
- kada je nagib mali, gradijent nam stalno pokazuje u istom pravcu i pritom je mali po normi
- pošto pokazuje stalno u istom pravcu, m se neće razlikovati mnogo od tog gradijenta
- kod v, na maloj nizbrdici je gradijent mali po normi pa će kvadrat biti još manji
- količnik m/v je vektor približno istog pravca kao m samo je duži jer smo delili malim brojem
- na taj način ćemo nizbrdicu brže preći

$$m_0 = 0$$

$$v_0 = 0$$

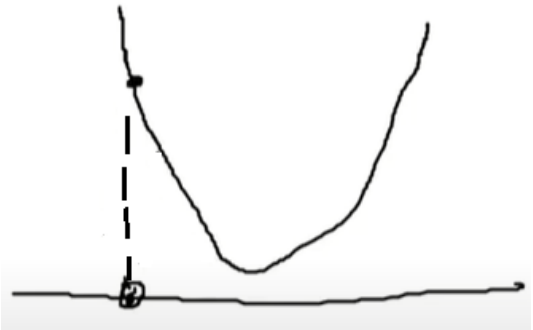
$$m_{k+1} = \beta_1 m_k + (1 - \beta_1)\nabla f(x_k)$$

$$v_{k+1} = \beta_2 v_k + (1 - \beta_2)\nabla f(x_k) \odot \nabla f(x_k)$$

$$x_{k+1} = x_k - \alpha_{k+1} \frac{\widehat{m}_{k+1}}{\sqrt{\widehat{v}_{k+1}} + \varepsilon}$$

Adam

- Analizirajmo količnik m i v



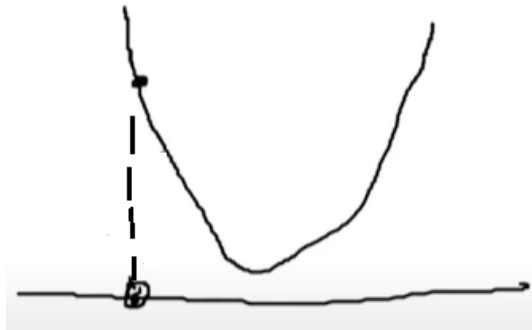
$$m_{k+1} = \beta_1 m_k + (1 - \beta_1)\nabla f(x_k)$$

$$v_{k+1} = \beta_2 v_k + (1 - \beta_2)\nabla f(x_k) \odot \nabla f(x_k)$$

$$x_{k+1} = x_k - \alpha_{k+1} \frac{\widehat{m}_{k+1}}{\sqrt{\widehat{v}_{k+1}} + \varepsilon}$$

Adam

- Analizirajmo količnik m i v
- kada se nalazimo na velikoj nizbrdici, minimum blizu pa postoji opasnost da počnemo da preskačemo s jedne strane minimuma i da vrlo sporo konvergiramo



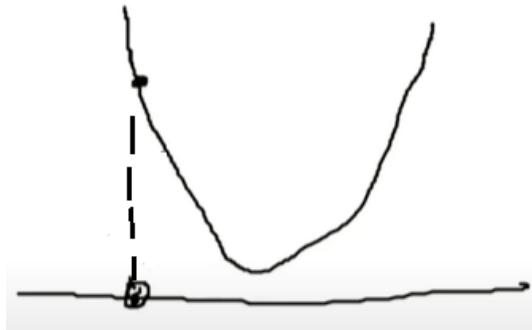
$$m_{k+1} = \beta_1 m_k + (1-\beta_1)\nabla f(x_k)$$

$$v_{k+1} = \beta_2 v_k + (1-\beta_2)\nabla f(x_k) \odot \nabla f(x_k)$$

$$x_{k+1} = x_k - \alpha_{k+1}\frac{\widehat{m}_{k+1}}{\sqrt{\widehat{v}_{k+1}} + \varepsilon}$$

Adam

- Analizirajmo količnik m i v
- kada se nalazimo na velikoj nizbrdici, minimum blizu pa postoji opasnost da počnemo da preskačemo s jedne strane minimuma i da vrlo sporo konvergiramo
- kakvo će biti m posle nekoliko takvih oscilacija?



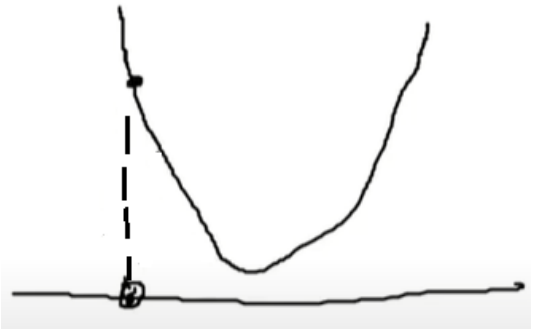
$$m_{k+1} = \beta_1 m_k + (1 - \beta_1)\nabla f(x_k)$$

$$v_{k+1} = \beta_2 v_k + (1 - \beta_2)\nabla f(x_k) \odot \nabla f(x_k)$$

$$x_{k+1} = x_k - \alpha_{k+1} \frac{\widehat{m}_{k+1}}{\sqrt{\widehat{v}_{k+1}} + \varepsilon}$$

Adam

- Analizirajmo količnik m i v
- kada se nalazimo na velikoj nizbrdici, minimum blizu pa postoji opasnost da počnemo da preskačemo s jedne strane minimuma i da vrlo sporo konvergiramo
- kakvo će biti m posle nekoliko takvih oscilacija?
- kada akumuliramo oscilujuce gradijente, prosek će biti mali zato sto su neki od njih pozitivni a neki negativni (skačemo levo-desno, smerovi su suprotni)



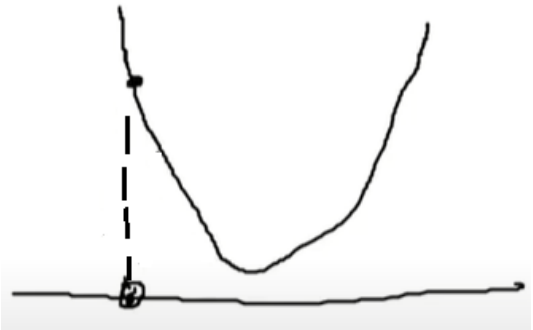
$$m_{k+1} = \beta_1 m_k + (1-\beta_1)\nabla f(x_k)$$

$$v_{k+1} = \beta_2 v_k + (1-\beta_2)\nabla f(x_k) \odot \nabla f(x_k)$$

$$x_{k+1} = x_k - \alpha_{k+1}\frac{\widehat{m}_{k+1}}{\sqrt{\widehat{v}_{k+1}} + \varepsilon}$$

Adam

- Analizirajmo količnik m i v



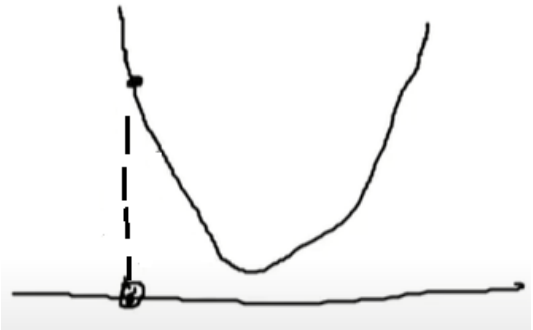
$$m_{k+1} = \beta_1 m_k + (1 - \beta_1)\nabla f(x_k)$$

$$v_{k+1} = \beta_2 v_k + (1 - \beta_2)\nabla f(x_k) \odot \nabla f(x_k)$$

$$x_{k+1} = x_k - \alpha_{k+1} \frac{\widehat{m}_{k+1}}{\sqrt{\widehat{v}_{k+1}} + \varepsilon}$$

Adam

- Analizirajmo količnik m i v
- kada se nalazimo na velikoj nizbrdici



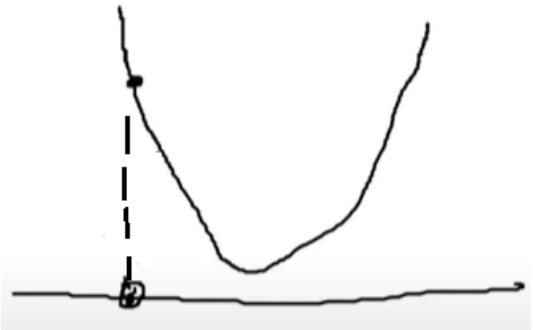
$$m_{k+1} = \beta_1 m_k + (1 - \beta_1)\nabla f(x_k)$$

$$v_{k+1} = \beta_2 v_k + (1 - \beta_2)\nabla f(x_k) \odot \nabla f(x_k)$$

$$x_{k+1} = x_k - \alpha_{k+1} \frac{\widehat{m}_{k+1}}{\sqrt{\widehat{v}_{k+1}} + \varepsilon}$$

Adam

- Analizirajmo količnik m i v
- kada se nalazimo na velikoj nizbrdici
- kakvo će biti v posle nekoliko takvih oscilacija?



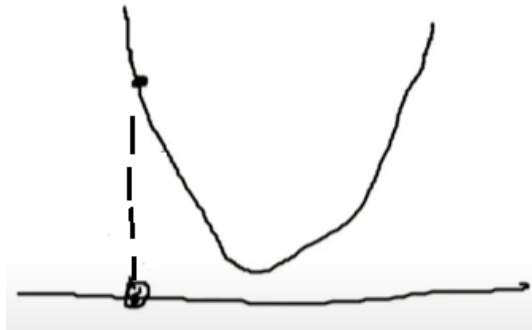
$$m_{k+1} = \beta_1 m_k + (1 - \beta_1)\nabla f(x_k)$$

$$v_{k+1} = \beta_2 v_k + (1 - \beta_2)\nabla f(x_k) \odot \nabla f(x_k)$$

$$x_{k+1} = x_k - \alpha_{k+1} \frac{\widehat{m}_{k+1}}{\sqrt{\widehat{v}_{k+1}} + \varepsilon}$$

Adam

- Analizirajmo količnik m i v
- kada se nalazimo na velikoj nizbrdici
- kakvo će biti v posle nekoliko takvih oscilacija?
- prosek gradijenata će biti mali ali su sa druge strane gradijenti po normi veliki pa ce i v biti veliko



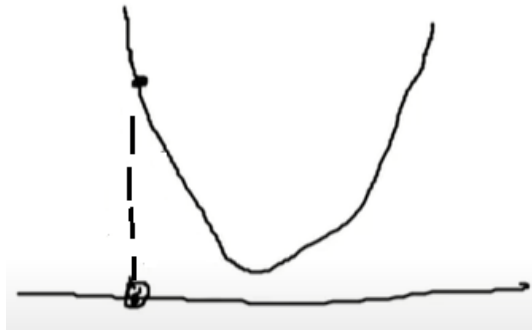
$$m_{k+1} = \beta_1 m_k + (1-\beta_1)\nabla f(x_k)$$

$$v_{k+1} = \beta_2 v_k + (1-\beta_2)\nabla f(x_k) \odot \nabla f(x_k)$$

$$x_{k+1} = x_k - \alpha_{k+1} \frac{\widehat{m}_{k+1}}{\sqrt{\widehat{v}_{k+1}} + \varepsilon}$$

Adam

- Analizirajmo količnik m i v
- kada se nalazimo na velikoj nizbrdici
- kakvo će biti v posle nekoliko takvih oscilacija?
- prosek gradijenata će biti mali ali su sa druge strane gradijenti po normi veliki pa ce i v biti veliko
- to znači da cemo u količniku m/v deliti mali broj velikim brojem i dobiti još manji broj



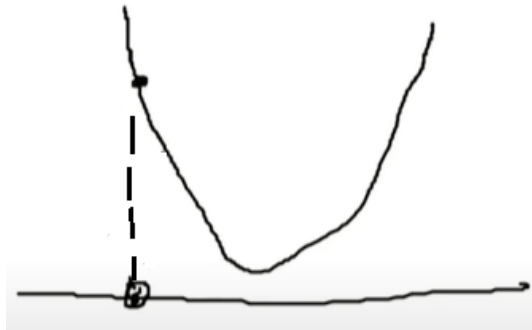
$$m_{k+1} = \beta_1 m_k + (1-\beta_1)\nabla f(x_k)$$

$$v_{k+1} = \beta_2 v_k + (1-\beta_2)\nabla f(x_k) \odot \nabla f(x_k)$$

$$x_{k+1} = x_k - \alpha_{k+1}\frac{\widehat{m}_{k+1}}{\sqrt{\widehat{v}_{k+1}} + \varepsilon}$$

Adam

- Analizirajmo količnik m i v
- kada se nalazimo na velikoj nizbrdici
- kakvo će biti v posle nekoliko takvih oscilacija?
- prosek gradijenata će biti mali ali su sa druge strane gradijenti po normi veliki pa ce i v biti veliko
- to znači da cemo u količniku m/v deliti mali broj velikim brojem i dobiti još manji broj
- na taj način ćemo nizbrdicu sporije preći



$$m_{k+1} = \beta_1 m_k + (1-\beta_1)\nabla f(x_k)$$

$$v_{k+1} = \beta_2 v_k + (1-\beta_2)\nabla f(x_k) \odot \nabla f(x_k)$$

$$x_{k+1} = x_k - \alpha_{k+1} \frac{\widehat{m}_{k+1}}{\sqrt{\widehat{v}_{k+1}} + \varepsilon}$$

Adam

- ova analiza ukazuje na jedno vrlo zanimljivo svojstvo Adama

Adam

- ova analiza ukazuje na jedno vrlo zanimljivo svojstvo Adama
- Adam će imati procene proseka i normi gradijenta (m i v), po svakoj koordinati

Adam

- ova analiza ukazuje na jedno vrlo zanimljivo svojstvo Adama
- Adam će imati procene proseka i normi gradijenta (m i v), po svakoj koordinati
- može se desiti da se po nekim koordinatama kreće brzo a po nekim da se kreće sporo

Adam

- ova analiza ukazuje na jedno vrlo zanimljivo svojstvo Adama
- Adam će imati procene proseka i normi gradijenta (m i v), po svakoj koordinati
- može se desiti da se po nekim koordinatama kreće brzo a po nekim da se kreće sporo
- to nije moguće kod gradijentnog spusta, gde postoji jedan parametar α koji kontroliše dužinu koraka

Adam

- ova analiza ukazuje na jedno vrlo zanimljivo svojstvo Adama
- Adam će imati procene proseka i normi gradijenta (m i v), po svakoj koordinati
- može se desiti da se po nekim koordinatama kreće brzo a po nekim da se kreće sporo
- to nije moguće kod gradijentnog spusta, gde postoji jedan parametar α koji kontroliše dužinu koraka
- Adam je u stanju da prilagodi dužinu koraka malo više po nekim pravcima nego po drugim i samim tim efektivno menjajući pravac u kom se kreće u odnosu na gradijent

Adam

- ova analiza ukazuje na jedno vrlo zanimljivo svojstvo Adama
- Adam će imati procene proseka i normi gradijenta (m i v), po svakoj koordinati
- može se desiti da se po nekim koordinatama kreće brzo a po nekim da se kreće sporo
- to nije moguće kod gradijentnog spusta, gde postoji jedan parametar α koji kontroliše dužinu koraka
- Adam je u stanju da prilagodi dužinu koraka malo više po nekim pravcima nego po drugim i samim tim efektivno menjajući pravac u kom se kreće u odnosu na gradijent
- kod Adama postoji efektivno preračunavanje brzine kretanja po različitim koordinatama nezavisno

Pregled

Gradijentni spust - podsećanje

- Odaberemo x_0 najčešće nasumice

Gradijentni spust - podsećanje

- Odaberemo x_0 najčešće nasumice
- U svakom koraku (k je redni broj koraka) se pomeramo u narednu tačku (x_k) na sledeći način:

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k)$$

Gradijentni spust - podsećanje

- Odaberemo x_0 najčešće nasumice
- U svakom koraku (k je redni broj koraka) se pomeramo u narednu tačku (x_k) na sledeći način:

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k)$$

- Kada primenjujemo gradijentni spust u mašinskom učenju, tada funkcija f predstavlja funkciju greške koja se računa po pojedinačnim instancama kojih može biti mnogo

Gradijentni spust - podsećanje

- Odaberemo x_0 najčešće nasumice
- U svakom koraku (k je redni broj koraka) se pomeramo u narednu tačku (x_k) na sledeći način:

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k)$$

- Kada primenjujemo gradijentni spust u mašinskom učenju, tada funkcija f predstavlja funkciju greške koja se računa po pojedinačnim instancama kojih može biti mnogo
- Na primer, ako radimo sa slikama, svaka instanca je slika potencijalno visoke rezolucije i računanje greške po svim instancama može biti vrlo skupo

Gradijentni spust - podsećanje

- Odaberemo x_0 najčešće nasumice
- U svakom koraku (k je redni broj koraka) se pomeramo u narednu tačku (x_k) na sledeći način:

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k)$$

- Kada primenjujemo gradijentni spust u mašinskom učenju, tada funkcija f predstavlja funkciju greške koja se računa po pojedinačnim instancama kojih može biti mnogo
- Na primer, ako radimo sa slikama, svaka instanca je slika potencijalno visoke rezolucije i računanje greške po svim instancama može biti vrlo skupo
- Pored toga, tu je i ograničena brzina hardvera

Gradijentni spust - podsećanje

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k)$$

- Naime, za neke modele mašinskog učenja (npr. za neuronske mreže) se kao specijalizovani hardver koriste grafičke kartice za matrične operacije

Gradijentni spust - podsećanje

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k)$$

- Naime, za neke modele mašinskog učenja (npr. za neuronske mreže) se kao specijalizovani hardver koriste grafičke kartice za matrične operacije
- gradijenti će se neuporedivo brže izračunati na grafičkoj kartici nego na glavnom procesoru računara

Gradijentni spust - podsećanje

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k)$$

- Naime, za neke modele mašinskog učenja (npr. za neuronske mreže) se kao specijalizovani hardver koriste grafičke kartice za matrične operacije
- gradijenti će se neuporedivo brže izračunati na grafičkoj kartici nego na glavnom procesoru računara
- međutim, da bismo nešto radili na grafičkoj kartici, moramo odgovarajuće podatke prebaciti na njen RAM koji može biti nedovoljno veliki, možda manji od RAM-a računara

Gradijentni spust - podsećanje

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k)$$

- Naime, za neke modele mašinskog učenja (npr. za neuronske mreže) se kao specijalizovani hardver koriste grafičke kartice za matrične operacije
- gradijenti će se neuporedivo brže izračunati na grafičkoj kartici nego na glavnom procesoru računara
- međutim, da bismo nešto radili na grafičkoj kartici, moramo odgovarajuće podatke prebaciti na njen RAM koji može biti nedovoljno veliki, možda manji od RAM-a računara
- sa druge strane, i RAM računara može biti mali u odnosu na celokupne podatke

Gradijentni spust - podsećanje

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k)$$

- Naime, za neke modele mašinskog učenja (npr. za neuronske mreže) se kao specijalizovani hardver koriste grafičke kartice za matrične operacije
- gradijenti će se neuporedivo brže izračunati na grafičkoj kartici nego na glavnom procesoru računara
- međutim, da bismo nešto radili na grafičkoj kartici, moramo odgovarajuće podatke prebaciti na njen RAM koji može biti nedovoljno veliki, možda manji od RAM-a računara
- sa druge strane, i RAM računara može biti mali u odnosu na celokupne podatke
- i sada, kad hoćemo da računamo gradijent, onda ne bismo mogli da celokupan skup podataka učitamo odjednom već bismo morali da to radimo prebacujući deo po deo podataka sa hard diska

Gradijentni spust - podsećanje

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k)$$

- Naime, za neke modele mašinskog učenja (npr. za neuronske mreže) se kao specijalizovani hardver koriste grafičke kartice za matrične operacije
- gradijenti će se neuporedivo brže izračunati na grafičkoj kartici nego na glavnom procesoru računara
- međutim, da bismo nešto radili na grafičkoj kartici, moramo odgovarajuće podatke prebaciti na njen RAM koji može biti nedovoljno veliki, možda manji od RAM-a računara
- sa druge strane, i RAM računara može biti mali u odnosu na celokupne podatke
- i sada, kad hoćemo da računamo gradijent, onda ne bismo mogli da celokupan skup podataka učitamo odjednom već bismo morali da to radimo prebacujući deo po deo podataka sa hard diska
- pritom, ne pomeramo se nigde dok ne izračunamo gradijent i tek kada izračunamo gradijent, možemo da napravimo jedan korak u optimizaciji

Gradijentni spust - podsećanje

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k)$$

- za računanje gradijenta smo potrošili mnogo vremena a, prisetimo se, taj gradijent uopšte ne mora biti optimalan pravac (ako su konture izdužene, to je vrlo loš pravac)

Gradijentni spust - podsećanje

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k)$$

- za računanje gradijenta smo potrošili mnogo vremena a, prisetimo se, taj gradijent uopšte ne mora biti optimalan pravac (ako su konture izdužene, to je vrlo loš pravac)
- trošenje velike količine vremena za računanje pravca koji će nam doneti vrlo malo napretka svakako nije ono što bismo želeli

Gradijentni spust - podsećanje

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k)$$

- za računanje gradijenta smo potrošili mnogo vremena a, prisetimo se, taj gradijent uopšte ne mora biti optimalan pravac (ako su konture izdužene, to je vrlo loš pravac)
- trošenje velike količine vremena za računanje pravca koji će nam doneti vrlo malo napretka svakako nije ono što bismo želeli
- ako već računamo loš pravac, hoćemo da ga računamo vrlo jeftino

Stohastički gradijentni spust

- Gradijentni spust:

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k)$$

Stohastički gradijentni spust

- Gradijentni spust:

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k)$$

- Pretpostavimo da se funkcija koju minimizujemo moše predstaviti kao prosek drugih jednostavnijih funkcija:

$$f(x) = \frac{1}{N} \sum_{i=1}^{N} f_i(x)$$

Stohastički gradijentni spust

- Gradijentni spust:

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k)$$

- Pretpostavimo da se funkcija koju minimizujemo moše predstaviti kao prosek drugih jednostavnijih funkcija:

$$f(x) = \frac{1}{N} \sum_{i=1}^{N} f_i(x)$$

- Funkcija greške koju ćemo minimizovati u ML modelima ispunjava ovo svojstvo

Stohastički gradijentni spust

- Gradijentni spust:

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k)$$

- Pretpostavimo da se funkcija koju minimizujemo moše predstaviti kao prosek drugih jednostavnijih funkcija:

$$f(x) = \frac{1}{N} \sum_{i=1}^{N} f_i(x)$$

- Funkcija greške koju ćemo minimizovati u ML modelima ispunjava ovo svojstvo
  - f_i tu predstavlja istu funkciju primenjenu na različite delove promenljive x koji predstavlja ceo skup podataka

Stohastički gradijentni spust

- Gradijentni spust:

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k)$$

- Pretpostavimo da se funkcija koju minimizujemo moše predstaviti kao prosek drugih jednostavnijih funkcija:

$$f(x) = \frac{1}{N} \sum_{i=1}^{N} f_i(x)$$

- Funkcija greške koju ćemo minimizovati u ML modelima ispunjava ovo svojstvo
  - f_i tu predstavlja istu funkciju primenjenu na različite delove promenljive x koji predstavlja ceo skup podataka
- Stohastički gradijentni spust podrazumeva da se novi korak izračunava prema sledećem pravilu:

$$x_{k+1} = x_k - \alpha_k \nabla f_i(x_k)$$

Stohastički gradijentni spust

- Gradijentni spust: $x_{k+1} = x_k - \alpha_k \nabla f(x_k)$

Stohastički gradijentni spust

- Gradijentni spust: $x_{k+1} = x_k - \alpha_k \nabla f(x_k)$
- Funkcija greške: $f(x) = \frac{1}{N} \sum_{i=1}^{N} f_i(x)$

Stohastički gradijentni spust

- Gradijentni spust: $x_{k+1} = x_k - \alpha_k \nabla f(x_k)$
- Funkcija greške: $f(x) = \frac{1}{N} \sum_{i=1}^{N} f_i(x)$
- Stohastički gradijentni spust: $x_{k+1} = x_k - \alpha_k \nabla f_i(x_k)$

Stohastički gradijentni spust

- Gradijentni spust: $x_{k+1} = x_k - \alpha_k \nabla f(x_k)$
- Funkcija greške: $f(x) = \frac{1}{N} \sum_{i=1}^{N} f_i(x)$
- Stohastički gradijentni spust: $x_{k+1} = x_k - \alpha_k \nabla f_i(x_k)$
- Gradijent je zamenjen *realizacijom nekog slučajnog vektora koji je kolinearan sa gradijentom*

Stohastički gradijentni spust

- Gradijentni spust: $x_{k+1} = x_k - \alpha_k \nabla f(x_k)$
- Funkcija greške: $f(x) = \frac{1}{N} \sum_{i=1}^{N} f_i(x)$
- Stohastički gradijentni spust: $x_{k+1} = x_k - \alpha_k \nabla f_i(x_k)$
- Gradijent je zamenjen *realizacijom nekog slučajnog vektora koji je kolinearan sa gradijentom*
- To nije bilo kakav slučajni vektor već slučajni vektor čije je čekivanje kolinearno sa gradijentom

Stohastički gradijentni spust

- Gradijentni spust: $x_{k+1} = x_k - \alpha_k \nabla f(x_k)$
- Funkcija greške: $f(x) = \frac{1}{N} \sum_{i=1}^{N} f_i(x)$
- Stohastički gradijentni spust: $x_{k+1} = x_k - \alpha_k \nabla f_i(x_k)$
- Gradijent je zamenjen *realizacijom nekog slučajnog vektora koji je kolinearan sa gradijentom*
- To nije bilo kakav slučajni vektor već slučajni vektor čije je čekivanje kolinearno sa gradijentom
- Ako je očekivanje slučajnog vektora kolinearno sa gradijentom, ne znači da je svaka njegova realizacija kolinearna sa gradijentom nego je očekivanje tačno na liniji gradijenta a realizacije mogu odstupati

Stohastički gradijentni spust

- U stohastičkom gradijentnom spustu, vektor gradijenta je zamenjen nekom njegovom nepreciznom ali nepristrasnom aproksimacijom

Stohastički gradijentni spust

- U stohastičkom gradijentnom spustu, vektor gradijenta je zamenjen nekom njegovom nepreciznom ali nepristrasnom aproksimacijom
- Nepristrasnom jer je očekivanje vektora koji koristimo baš gradijent

Stohastički gradijentni spust

- U stohastičkom gradijentnom spustu, vektor gradijenta je zamenjen nekom njegovom nepreciznom ali nepristrasnom aproksimacijom
- Nepristrasnom jer je očekivanje vektora koji koristimo baš gradijent
- Nepreciznom jer aproksimacija nije jednaka gradijentu (može odstupati)

Stohastički gradijentni spust

- U stohastičkom gradijentnom spustu, vektor gradijenta je zamenjen nekom njegovom nepreciznom ali nepristrasnom aproksimacijom
- Nepristrasnom jer je očekivanje vektora koji koristimo baš gradijent
- Nepreciznom jer aproksimacija nije jednaka gradijentu (može odstupati)
- Ispostavlja se da i dalje važe teorijske garancije konvergencije (metoda konvergira)

Stohastički gradijentni spust

- U stohastičkom gradijentnom spustu, vektor gradijenta je zamenjen nekom njegovom nepreciznom ali nepristrasnom aproksimacijom
- Nepristrasnom jer je očekivanje vektora koji koristimo baš gradijent
- Nepreciznom jer aproksimacija nije jednaka gradijentu (može odstupati)
- Ispostavlja se da i dalje važe teorijske garancije konvergencije (metoda konvergira)
- Gubi se asimptotski na efikasnosti (nije više $\frac{1}{k}$ nego $\frac{1}{\sqrt{k}}$)

Stohastički gradijentni spust

- Funkcija greške: $f(x) = \frac{1}{N} \sum_{i=1}^{N} f_i(x)$

Stohastički gradijentni spust

- Funkcija greške: $f(x) = \frac{1}{N} \sum_{i=1}^{N} f_i(x)$
- Stohastički gradijentni spust: $x_{k+1} = x_k - \alpha_k \nabla f_i(x_k)$

Stohastički gradijentni spust

- Funkcija greške: $f(x) = \frac{1}{N} \sum_{i=1}^{N} f_i(x)$
- Stohastički gradijentni spust: $x_{k+1} = x_k - \alpha_k \nabla f_i(x_k)$
- Analizirajmo računanje koraka u stohastičkom gradijentnom spustu

Stohastički gradijentni spust

- Funkcija greške: $f(x) = \frac{1}{N} \sum_{i=1}^{N} f_i(x)$
- Stohastički gradijentni spust: $x_{k+1} = x_k - \alpha_k \nabla f_i(x_k)$
- Analizirajmo računanje koraka u stohastičkom gradijentnom spustu
- U x_{k+1} ćemo doći tako što ćemo da se od x_k krećemo za korak α suprotno od gradijenta izračunatog *samo u odnosu na jednu od funkcija koju uprosečavamo*

Stohastički gradijentni spust

- Funkcija greške: $f(x) = \frac{1}{N} \sum_{i=1}^{N} f_i(x)$
- Stohastički gradijentni spust: $x_{k+1} = x_k - \alpha_k \nabla f_i(x_k)$
- Analizirajmo računanje koraka u stohastičkom gradijentnom spustu
- U x_{k+1} ćemo doći tako što ćemo da se od x_k krećemo za korak α suprotno od gradijenta izračunatog *samo u odnosu na jednu od funkcija koju uprosečavamo*
- U slučaju da radimo sa slikama, to znači da ćemo izračunati gradijent samo u jednoj slici i odmah preduzimamo korak

Stohastički gradijentni spust

- Funkcija greške: $f(x) = \frac{1}{N} \sum_{i=1}^{N} f_i(x)$
- Stohastički gradijentni spust: $x_{k+1} = x_k - \alpha_k \nabla f_i(x_k)$
- Analizirajmo računanje koraka u stohastičkom gradijentnom spustu
- U x_{k+1} ćemo doći tako što ćemo da se od x_k krećemo za korak α suprotno od gradijenta izračunatog *samo u odnosu na jednu od funkcija koju uprosečavamo*
- U slučaju da radimo sa slikama, to znači da ćemo izračunati gradijent samo u jednoj slici i odmah preduzimamo korak
  - nećemo kao kod gradijenta da čekamo da izračunamo funkciju greške i njen gradijent na svim slikama kao kod gradijentnog spusta nego samo na jednoj, pa se pomerimo, pa na sledećoj, pa se opet pomerimo i tako dalje

Stohastički gradijentni spust

- Kako znamo da je vektor kojim smo zamenili gradijent predstavlja realizaciju slučajne promenljive čije je očekivanje gradijent?

Stohastički gradijentni spust

- Kako znamo da je vektor kojim smo zamenili gradijent predstavlja realizaciju slučajne promenljive čije je očekivanje gradijent?
- Definišemo slučajnu promenljivu kao skup uzoraka koji su dati sabircima na osnovu kojih se dobija funkcija greške: $f(x) = \frac{1}{N} \sum_{i=1}^{N} f_i(x)$

Stohastički gradijentni spust

- Kako znamo da je vektor kojim smo zamenili gradijent predstavlja realizaciju slučajne promenljive čije je očekivanje gradijent?
- Definišemo slučajnu promenljivu kao skup uzoraka koji su dati sabircima na osnovu kojih se dobija funkcija greške: $f(x) = \frac{1}{N} \sum_{i=1}^{N} f_i(x)$
- Očekivanje tako definisane slučajne promenljive je upravo prosek njenih uzoraka

Stohastički gradijentni spust

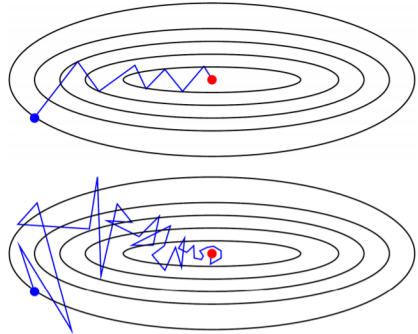
- Kako znamo da je vektor kojim smo zamenili gradijent predstavlja realizaciju slučajne promenljive čije je očekivanje gradijent?
- Definišemo slučajnu promenljivu kao skup uzoraka koji su dati sabircima na osnovu kojih se dobija funkcija greške: $f(x) = \frac{1}{N} \sum_{i=1}^{N} f_i(x)$
- Očekivanje tako definisane slučajne promenljive je upravo prosek njenih uzoraka
- To znači da uvek umesto proseka možemo da uzmemo jedan sabirak koji ulazi u prosek i da ga koristimo za izračunavanje gradijenta

Stohastički gradijentni spust

- Kako znamo da je vektor kojim smo zamenili gradijent predstavlja realizaciju slučajne promenljive čije je očekivanje gradijent?
- Definišemo slučajnu promenljivu kao skup uzoraka koji su dati sabircima na osnovu kojih se dobija funkcija greške: $f(x) = \frac{1}{N} \sum_{i=1}^{N} f_i(x)$
- Očekivanje tako definisane slučajne promenljive je upravo prosek njenih uzoraka
- To znači da uvek umesto proseka možemo da uzmemo jedan sabirak koji ulazi u prosek i da ga koristimo za izračunavanje gradijenta
- Na taj način pravimo napredak u optimizaciji - umesto da kao kod gradijentnog spusta čekamo da bude izračunat gradijent po svim podacima pre nego što napravimo korak u optimizaciji, sada ćemo moći da izračunamo gradijent bilo svake pojedinačne slike bilo nekog podskupa celokupnog skupa slika (*minibatch*) koliko može da stane u RAM grafičke kartice i da čim izračunamo jednu turu gradijenata, odmah to upotrebimo za dalje kretanje

Stohastički gradijentni spust

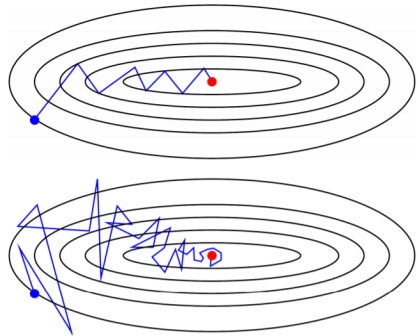
- Pomenuli smo da stohastički gradijentni spust ima lošiji asimptotski red konvergencije



Slika 9.4: Ponašanje gradijentnog spusta i stohastičkog gradijentnog spusta.

Stohastički gradijentni spust

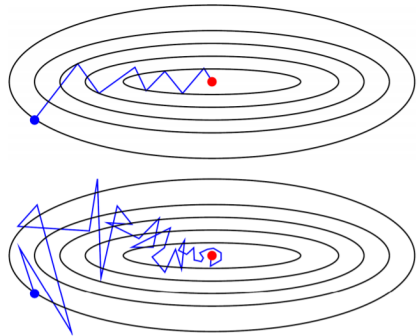
- Pomenuli smo da stohastički gradijentni spust ima lošiji asimptotski red konvergencije
- Zaista, kod stohastičkog gradijentnog spusta biće potrebno više koraka nego kod gradijentnog spusta ali je u prvom slučaju dužina trajanja jednog koraka mnogo manja nego u drugom jer se u svakom koraku koristi samo jedna instanca ili na manji podskup instanci



Slika 9.4: Ponašanje gradijentnog spusta i stohastičkog gradijentnog spusta.

Stohastički gradijentni spust

- Pomenuli smo da stohastički gradijentni spust ima lošiji asimptotski red konvergencije
- Zaista, kod stohastičkog gradijentnog spusta biće potrebno više koraka nego kod gradijentnog spusta ali je u prvom slučaju dužina trajanja jednog koraka mnogo manja nego u drugom jer se u svakom koraku koristi samo jedna instanca ili na manji podskup instanci
- Stohastički gradijentni spust ima veliku prednost u odnosu na gradijentni spust prilikom rada sa redundantnim podacima



Slika 9.4: Ponašanje gradijentnog spusta i stohastičkog gradijentnog spusta.