Deep Learning

2023-11-27

Ämne: GD varianter och optimization algoritms.

Smartr

Agenda

- Gradient descent och optimeringsalgoritmer
- Vanishing/exploding gradient problem
- Batch
- Databehandling

Ordlista träning

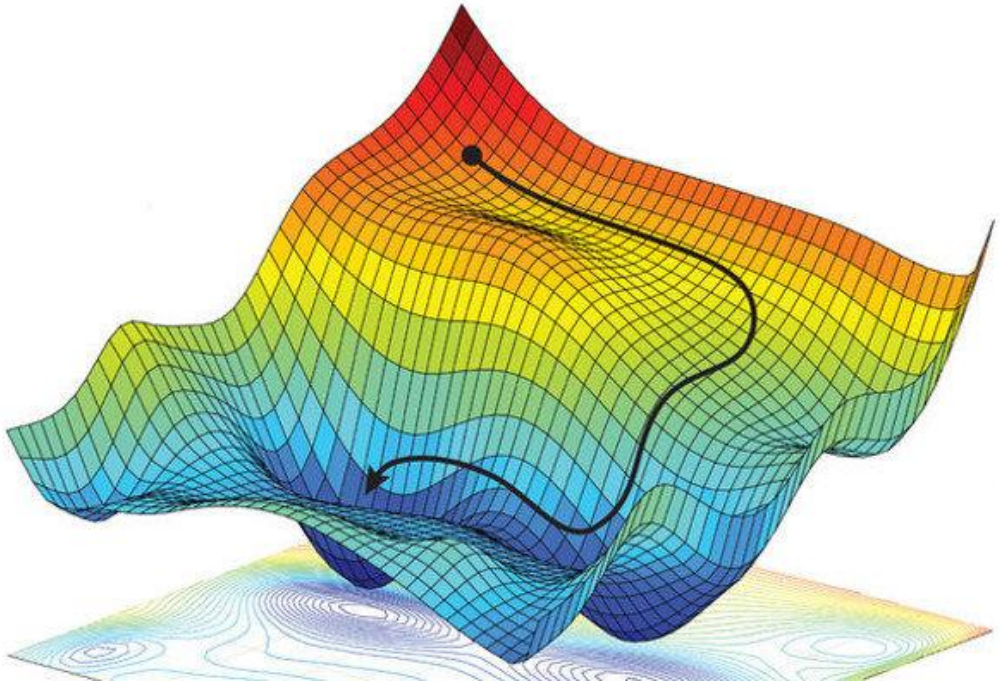
- **Iteration** – en bakåtpropagering med viktuppdatering.
- **Batch** – Under en iteration så skickar man ibland inte in all data på samma gång. Batch är en delmängd data som oftast samplas slumpmässigt från totala datamängden. Återkommer om batch senare under lektionern.
- **Epok** – När man itererat (forward pass + bakåtprop) igenom all data 1 gång. Använder man batches blir detta flera iterationer. Notera att om man använder hela datamängden vid en iteration så är 1 iteration=1 epok.

Gradient descent fortsättning

- Gradient descent handlar om att hitta minimum av en stor funktion med jättemånga dimensioner.
- Används för att uppdatera vikterna så att nätverket ”lär sig”.

- ”bollen rullar neråt i backen”

- Här är en visualisering av GD



Smartr

GD varianter och optimeringsalgoritmer

Det finns smarta varianter på GD samt optimeringsalgoritmer för att hitta minimum snabbare utan att fastna i lokala minima.

Populära varianter på GD:

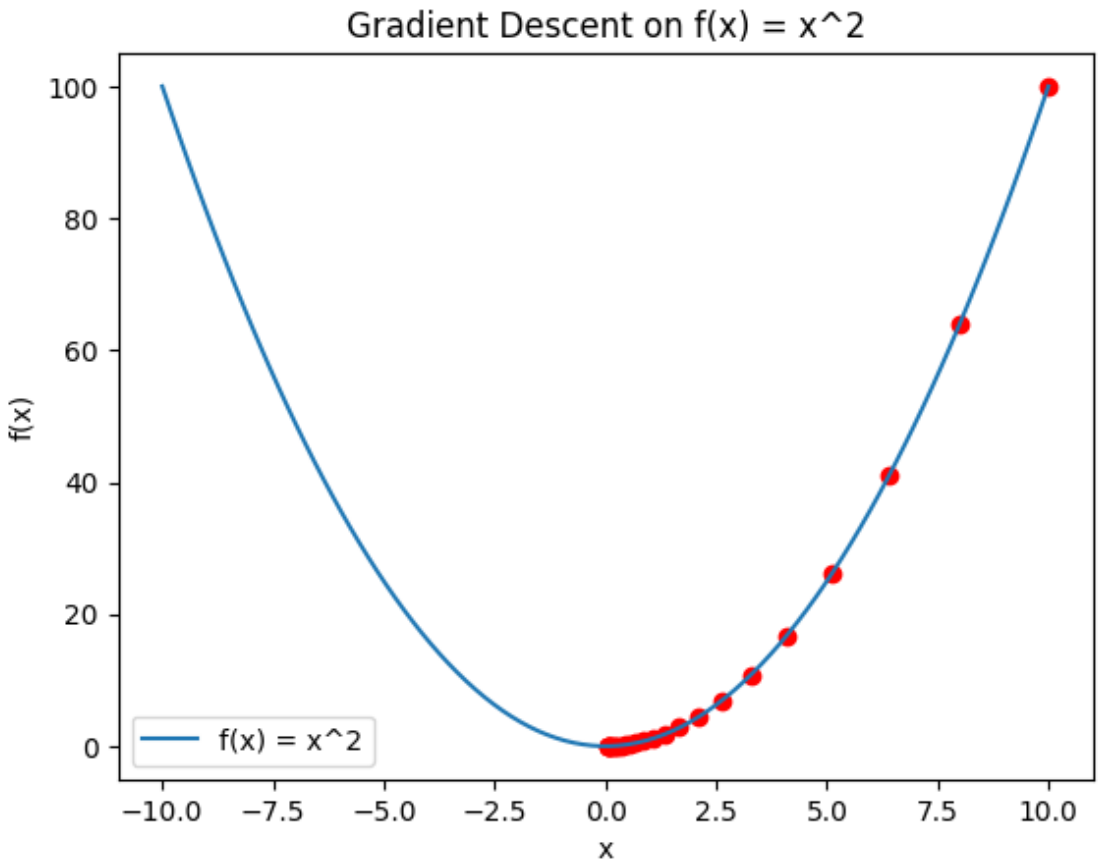
- Batch gradient descent
- Stocastic gradient descent (SGD)
- Mini-batch stocastic gradient descent

Populära optimeringsalgoritmer:

- Momentum
- Adam
- RMSProp
- Finns många fler men vi täcker inte dem i denna kursen.

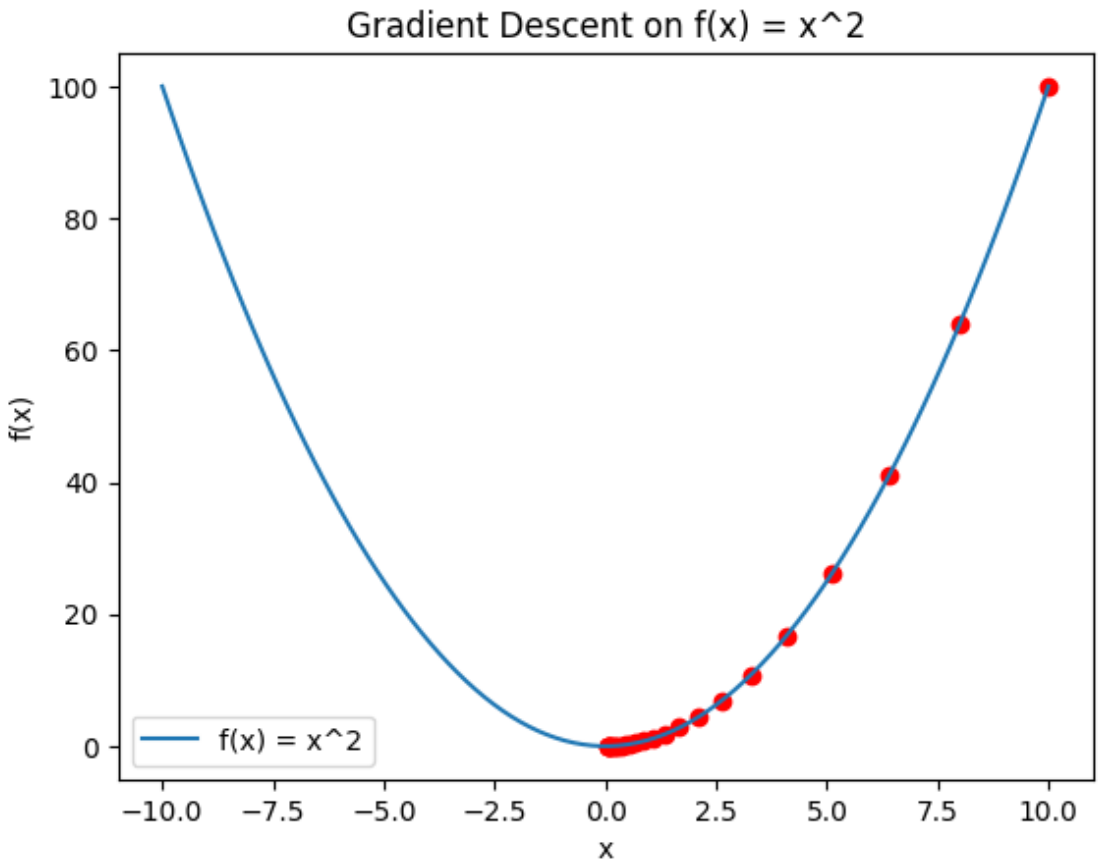
Batch GD

- Vi gör vårat hopp när modellen sett all data.
  - En iteration = En enpok.
- Medelvärdet på gradienten för alla träningsexempel.



Batch GD

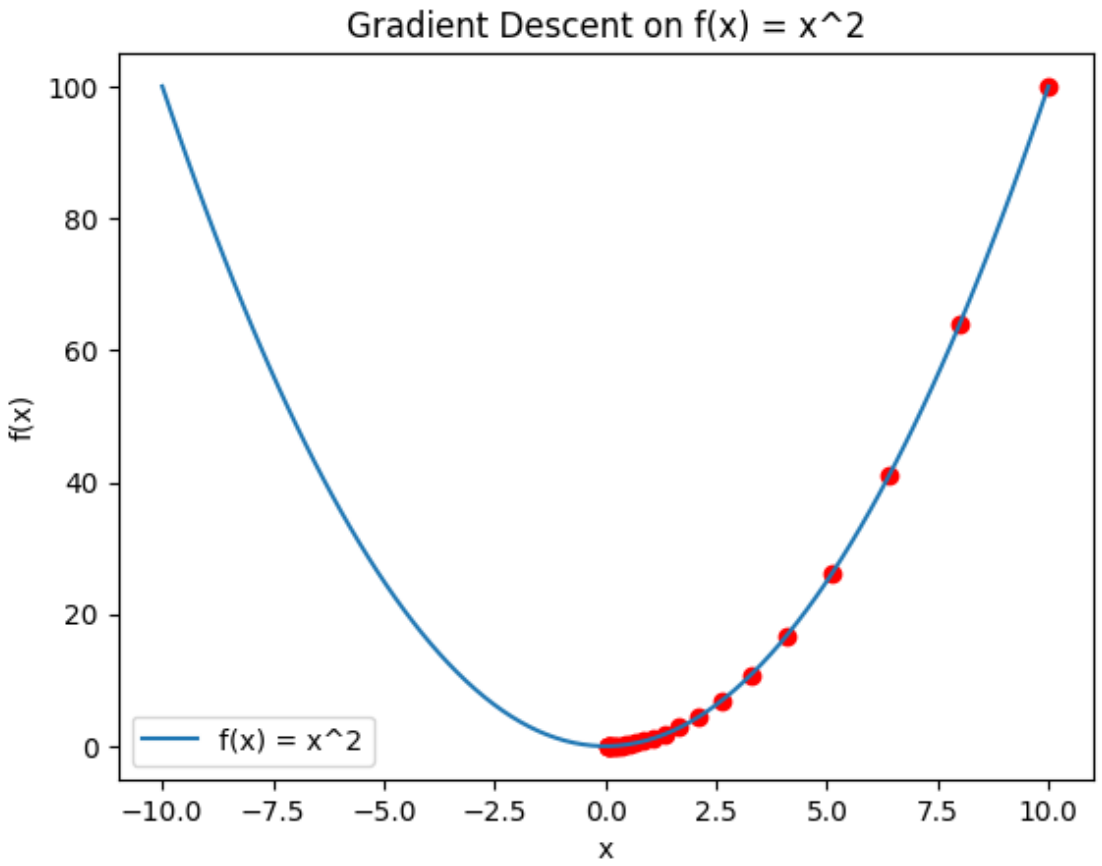
- Stabil uppdatering
- Fungerar bra på små dataset
- Enklare att parallellisera
- Kräver mycket beräkningskraft
  - Kan bli slö för stora dataset
- Kräver mycket minne
- Kan fastna i lokala minimum



Stocastic batch GD

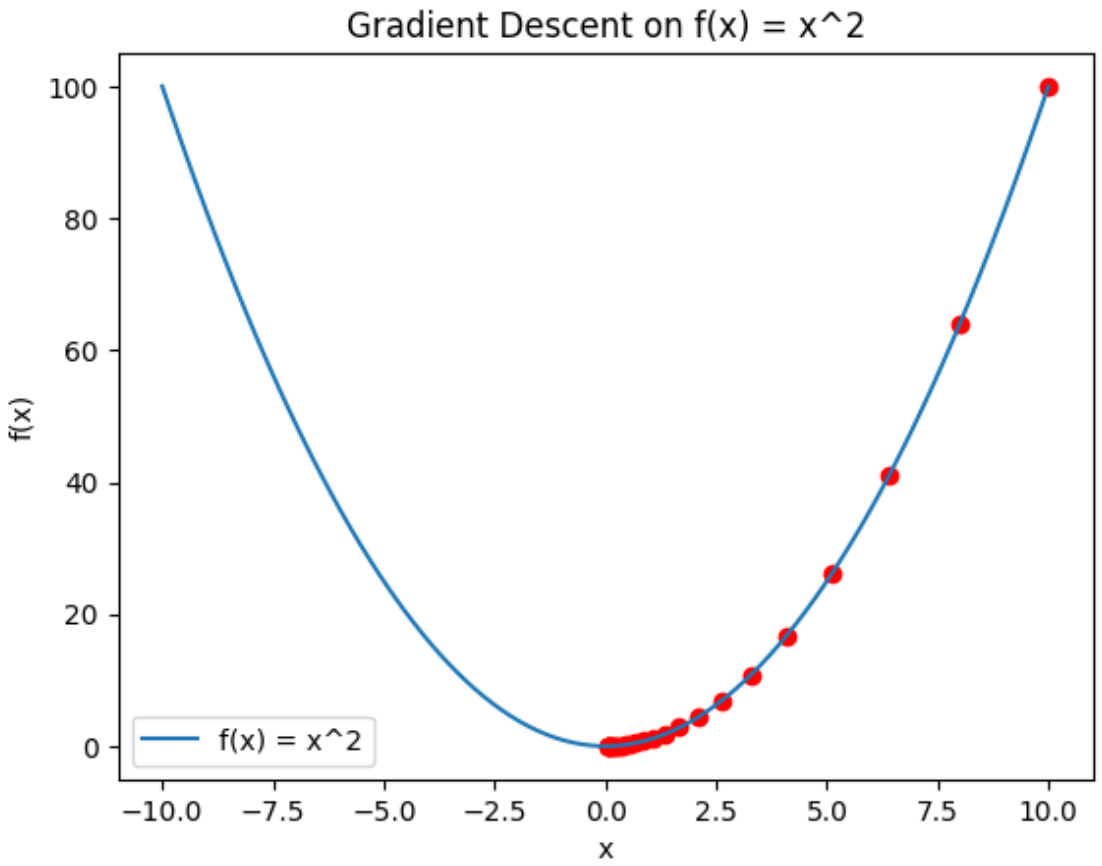
Vi gör ett hopp för varje träningspunkt i datan.

1. Ta en träningspunkt
2. Kör den igenom nätverket
3. Beräkna gradienten
4. Uppdatera vikterna
5. Repetera 1-4 för alla datapunkter i träningsdatan



Stocastic batch GD

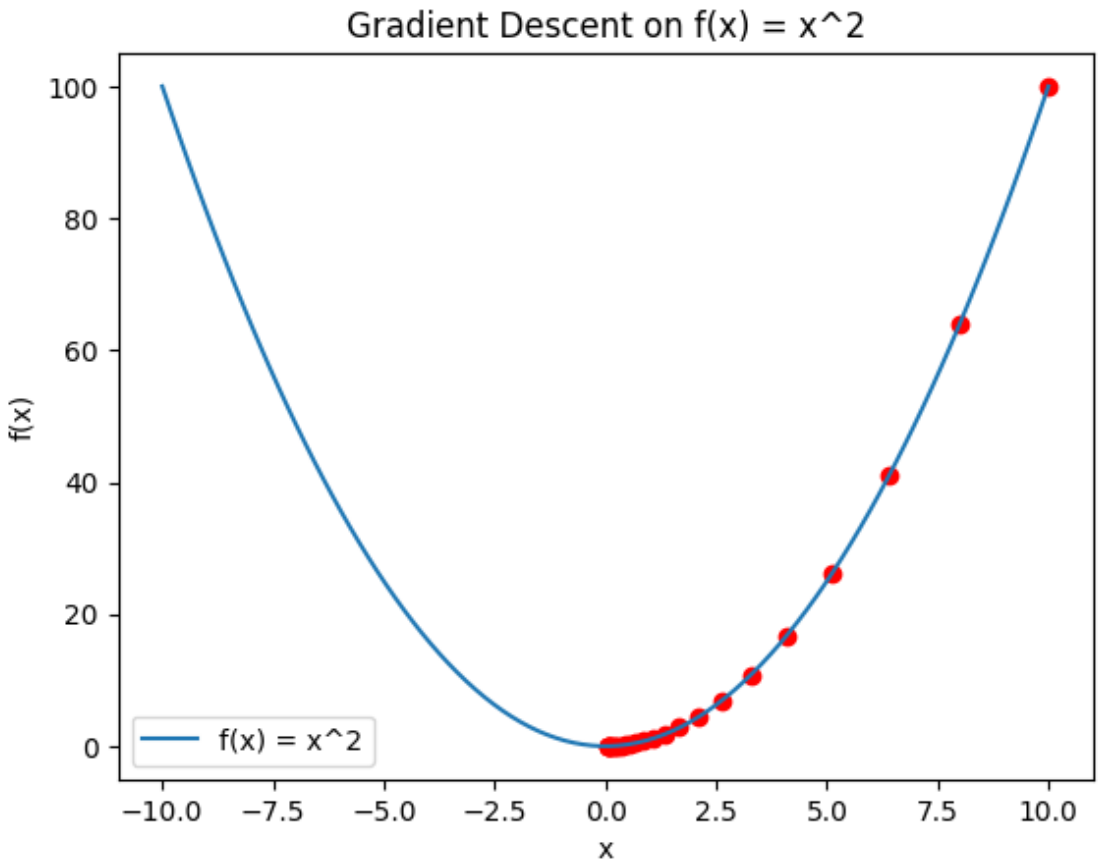
- Snabba iterationer.
- Fungerar bra på stora dataset.
- Kommer oftast ur lokala minimum.
- Hög varians mellan iterationerna.
- Kan ta längre tid att konvergera.



Stochastic mini-batch GD

Stochastic mini-batch GD (även kalat mini batch) är en blanding av Batch GD och Stochastic batch GD.

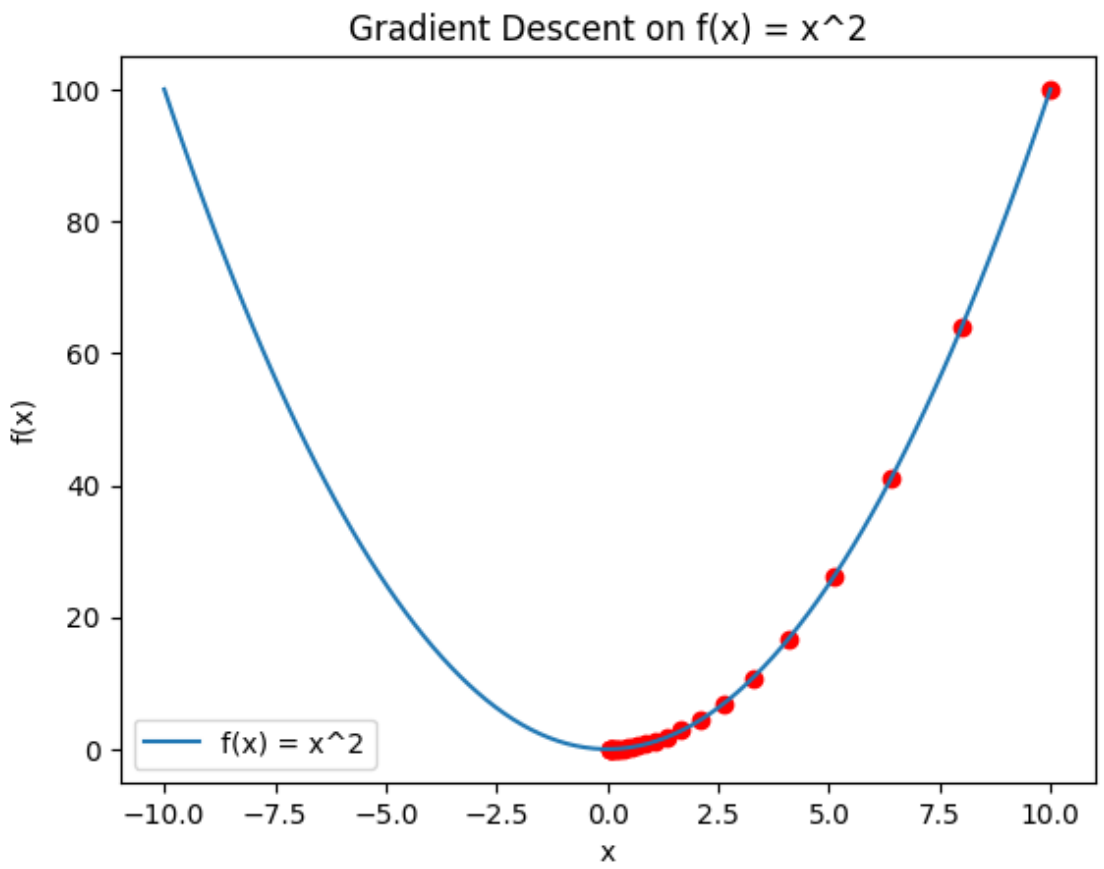
Här bestämmer vi hur många exempel modellen ska få se varje iteration.



Stochastic mini-batch GD

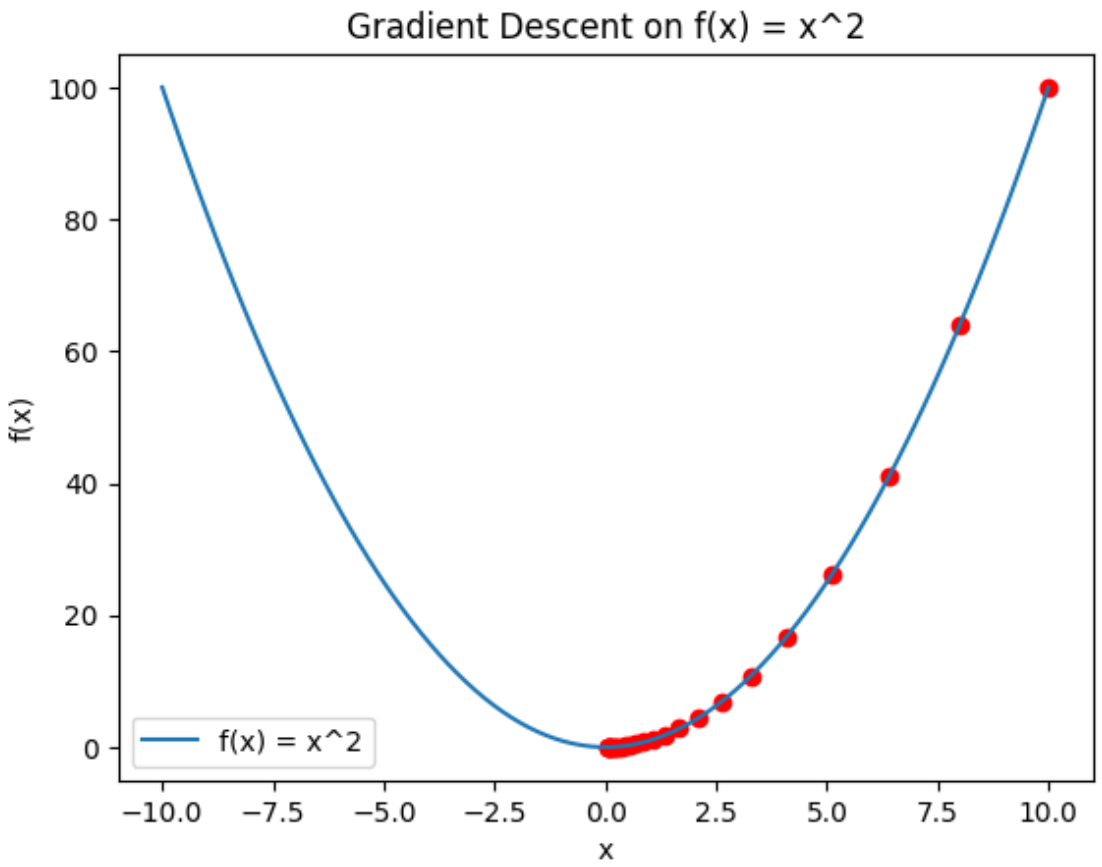
Vi gör ett hopp för varje batch av data.

1. Ta x träningspunkter
2. Kör dem igenom nätverket
3. Beräkna medelvärdet av gradienten
4. Uppdatera vikterna
5. Repetera 1-4 för alla batcher



Stochastic mini-batch GD

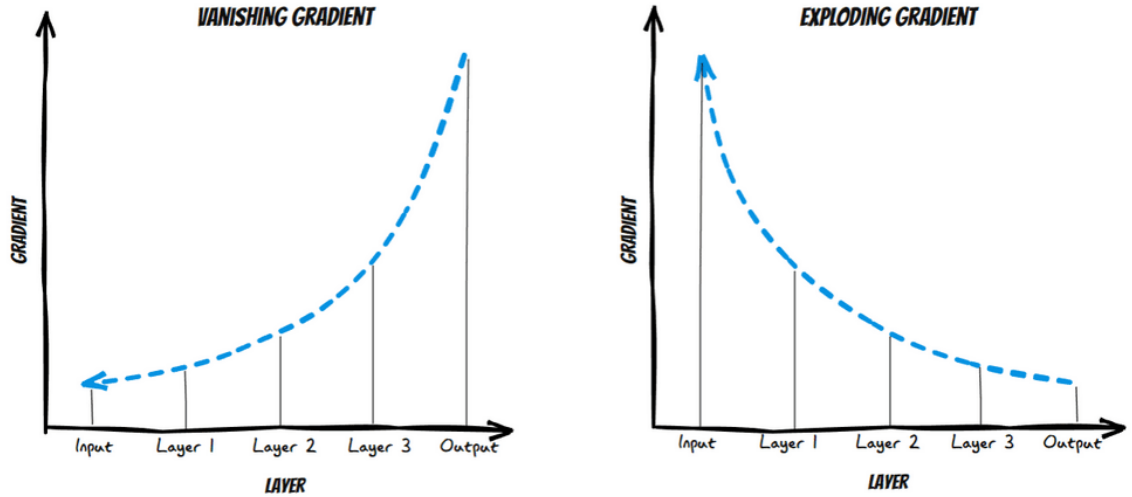
- Plockar fördelarna från både Batch GD och SGD.
- Effektivt på stora dataset.
  - Fungerar även på små (bara att ändra batch size).
- Introducerar ytterligare en parameter.
- Storleken din mini-batch kan påverka träningstiden (och slutresultatet).



Gradient problem

Får vi för små eller för stora gradienter kan det leda till att nätverket inte uppdateras eller att det blir instabilt.

Detta kallas för Vanishing gradient och Exploding gradient.

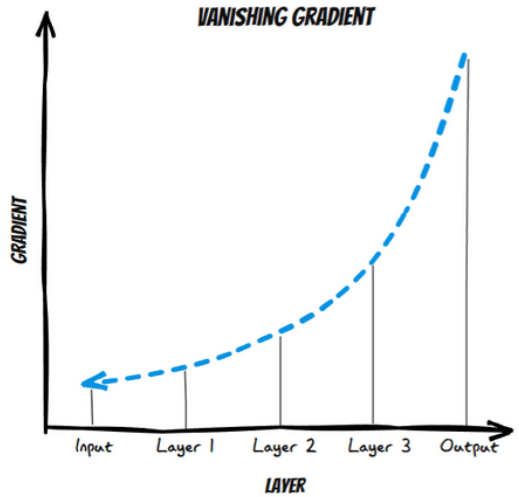


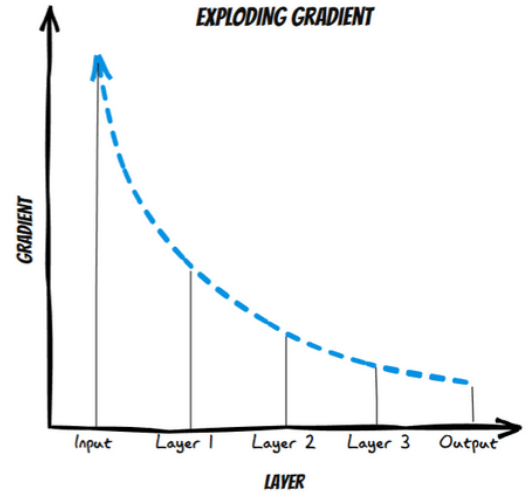
Vanishing gradient problem

Vanishing gradient sker när gradienten sjunker exponentiellt genom nätverkets lager, vilket leder till att vikterna i tidiga lager inte uppdateras.

Det kan bland annat bero på

- Ett väldigt djupt nätverk
- Aktiveringsfunktionen
  - Exempelvis sigmoid och tanh har små gradienter som kan leda till Vanishing gradient

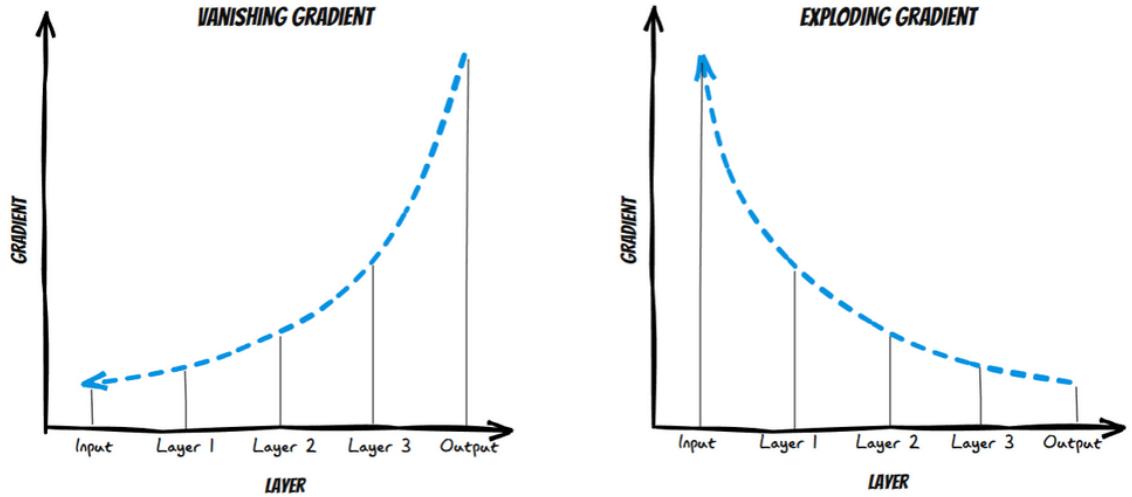




Vanishing gradient problem

För att undvika/motverka vanishing gradient kan vi göra ändringar i nätverket.

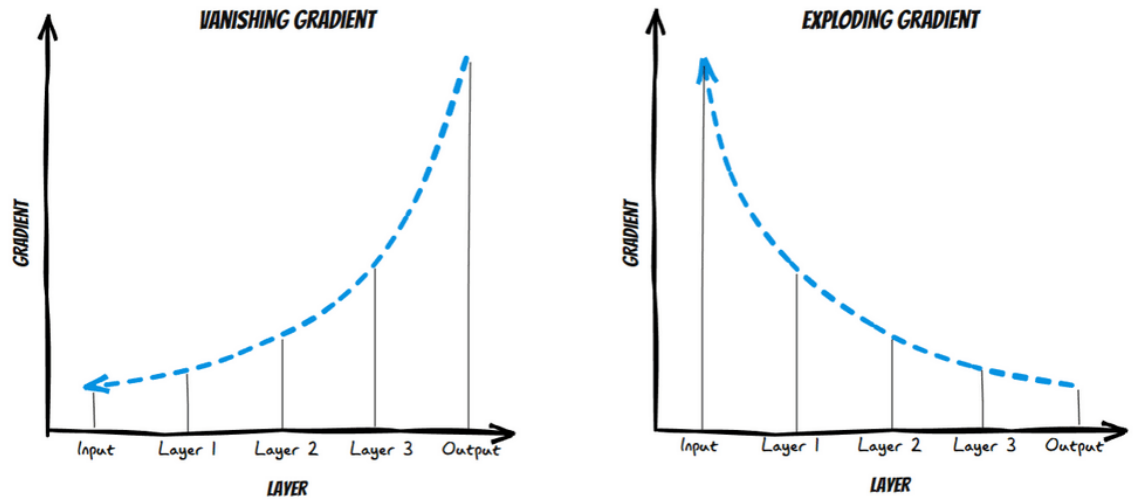
- Ändra aktiveringsfunktion (till exempelvis ReLU).
- Korta ner nätverket.
- Ändra learning rate och optimeringsalgoritm.



Exploding gradient problem

Exploding gradient innebär att gradienten blir väldigt stor när vi rör oss bakåt i nätveket.

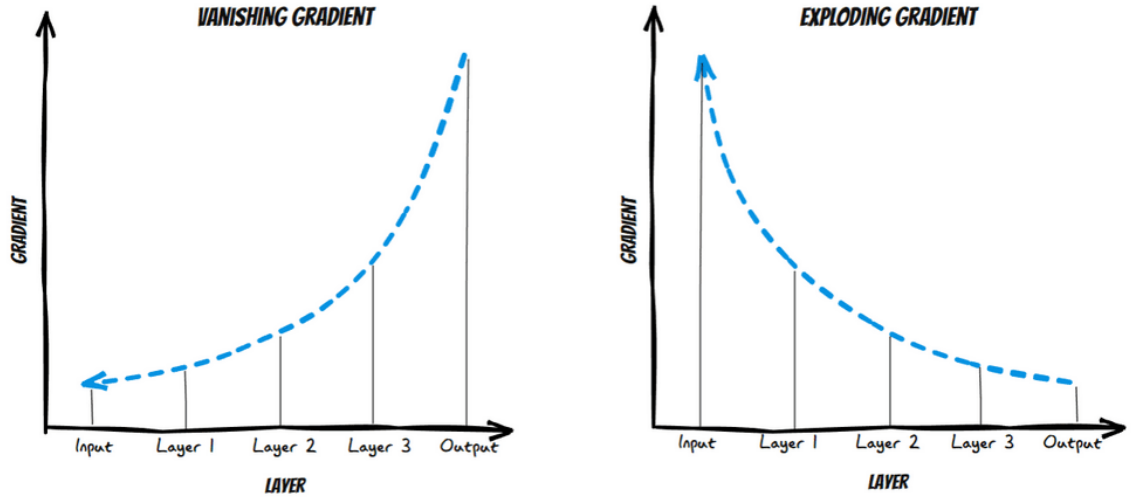
- Modellvikter blir NaN väldigt snabbt.
- Model loss blir NaN



Exploding gradient problem

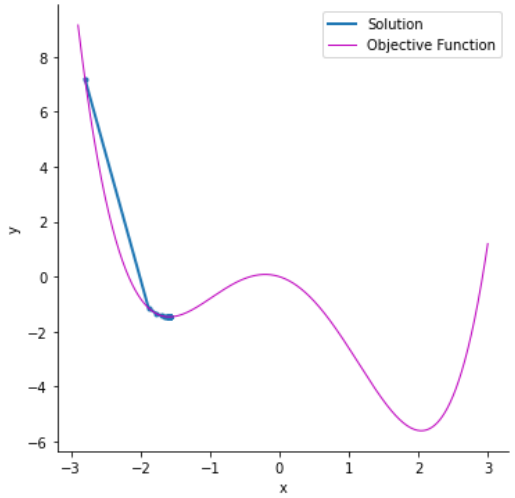
För att undvika/motverka explodinggradient kan vi göra ändringar i nätverket.

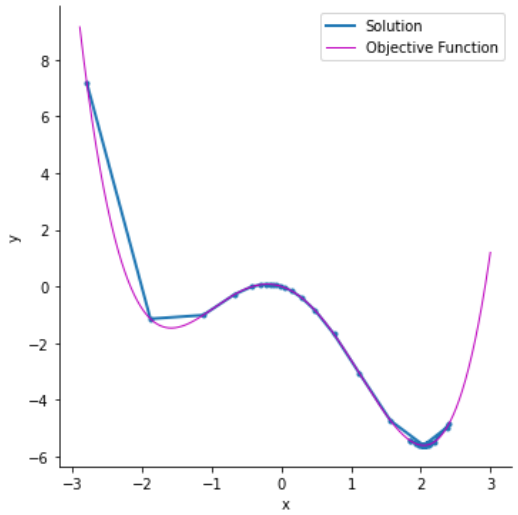
- L2 Regulaisering
  - Vi straffar stora vikter
- Ändra nätverksarkitekturen
  - Minska nätverkets djup
- Ändra learning rate
  - Minska värdet
- Gradient clipping
  - Sätt ett maxvärde på gradienten



Momentum

- Momentum fungerar precis som momentum i fysik.
  - Tänk en boll som rullar i nerförsbacke.
- Adderar en historieparameter.

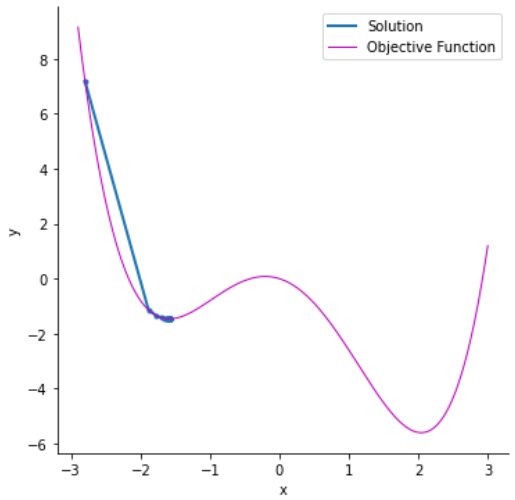


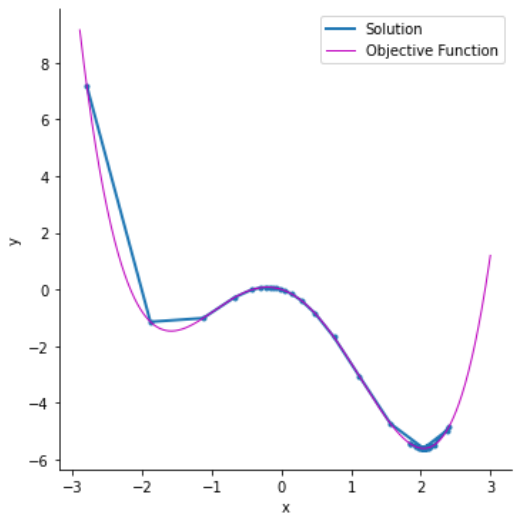


Smartr

Momentum

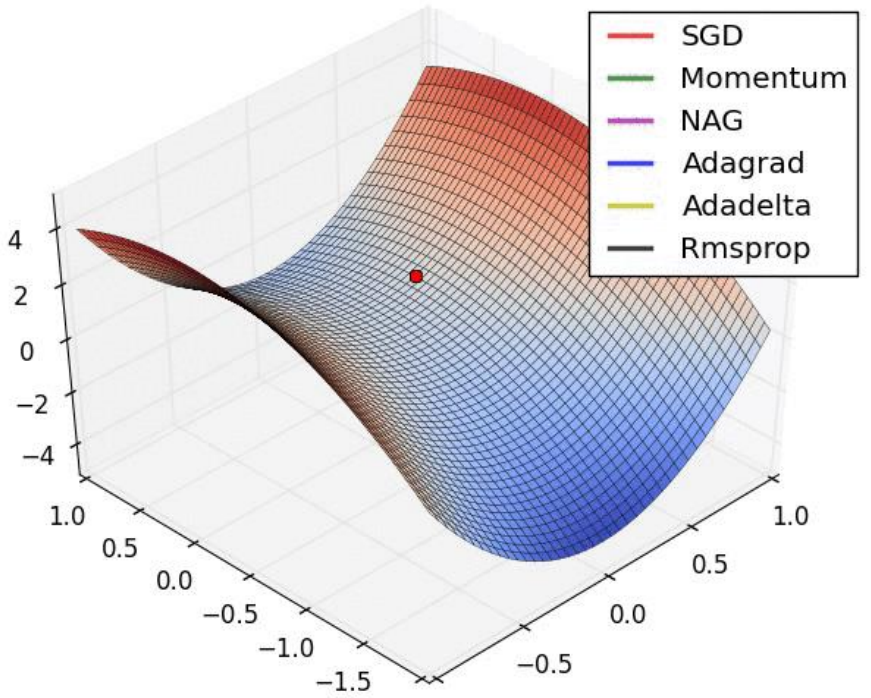
- Konvergera snabbare.
- Kan undvika att fastna i lokala minimum.
- Introducerar en ny parameter.





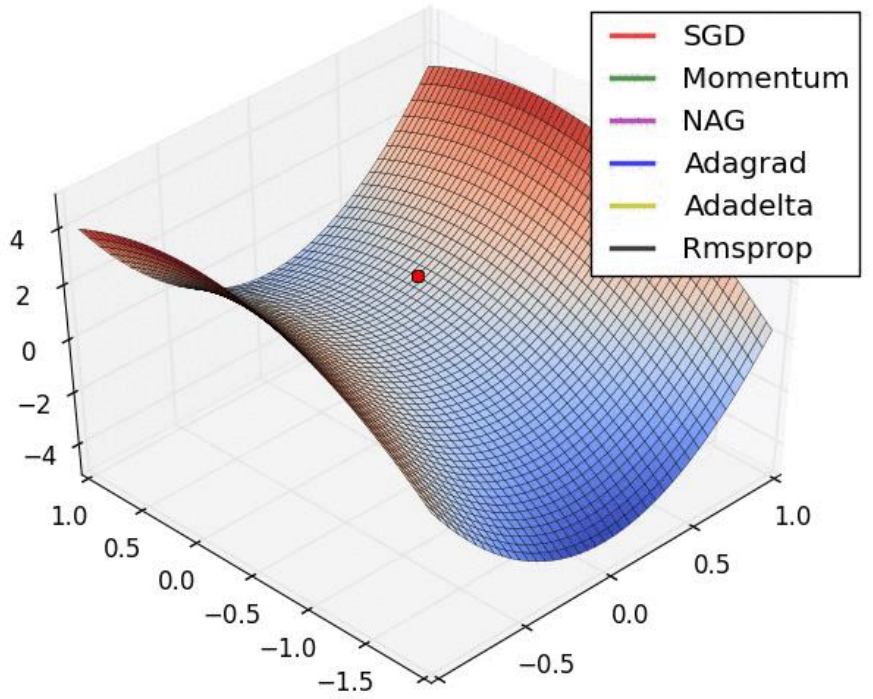
RMSProp

- Root Mean Square Propagation
- Addaptiv learning rate
- Minskar stegen för stora gradienter
- Ökar stegen för små gradidenter



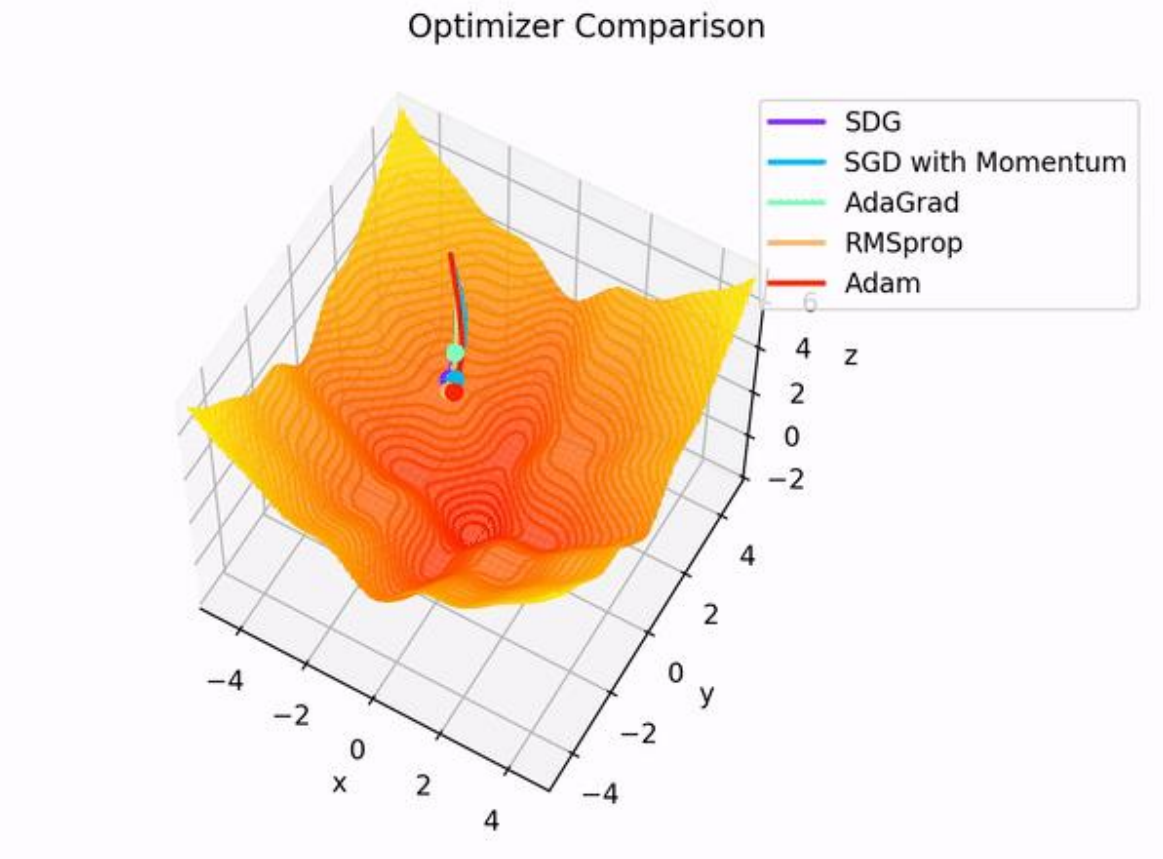
RMSProp

- Konvergera snabbare.
- Kan undvika att fastna i lokala minimum.
- Minskar risk för exploding och vanishing gradient.
- Introducerar en ny parameter.



Adam

- Adaptive Moment Estimation
- Addaptiv learning rate
- Kombinerar Momentum och RMSProp.
- En väldigt vanlig optimeringsalgoritm.
  - Typisk första val när man testar.



Adam

- Hyfsat beräkningseffektiv.
  - Men dyrare än enklare algoritmer.
- Kräver lite minne.
- Fungerar bra med stora dataset med många parametrar.
  - Typiskt vanligt för deep learning problem.
- Har få nackdelar.
- Kan overifita på små dataset.