



Artificiella Neurala Nätverk



Agenda

- Varför nu
- Neuron
- Neuralt Nätverk - delar
- Activation Functions
- Hur fungerar ett ANN
- Gradient Descent
- Stochastic Gradient Descent
- Backpropagation
- Keras
- Modellen

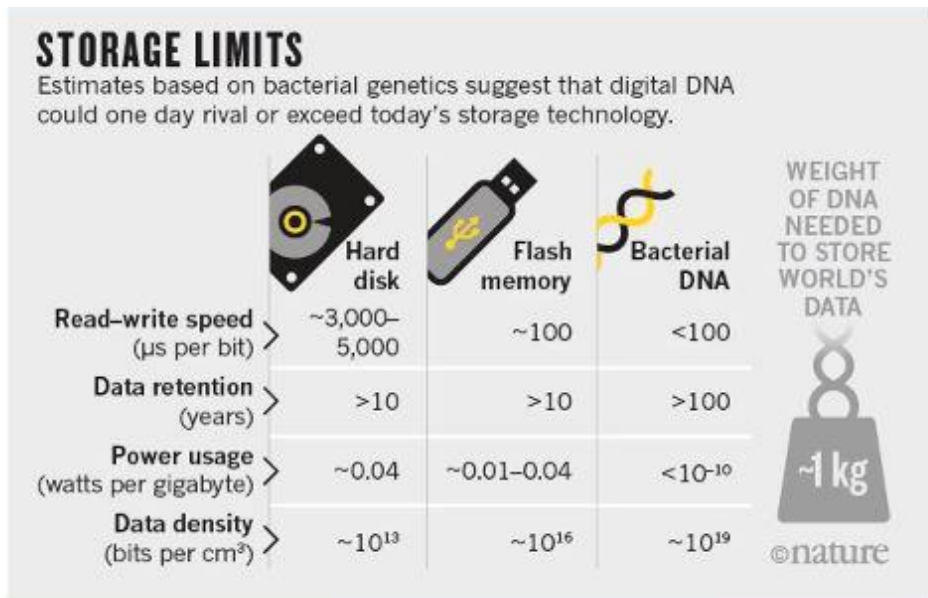
Kod

- ANN i Tensorflow
-

ANN - Varför nu?



- Minneskapaciteten
- Priset
- Processor
- DNA som “data storage”? Forskarna jobbar på det!



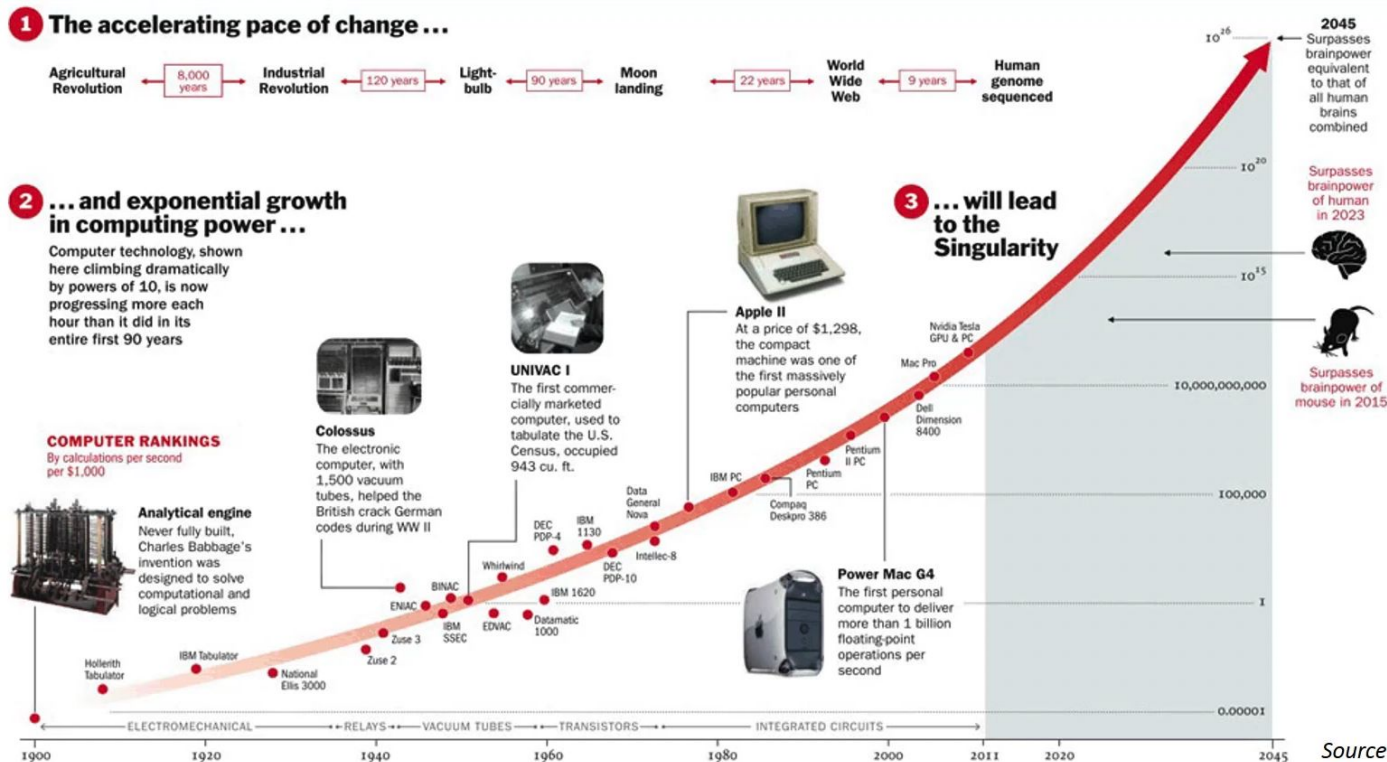
RE STORY...

ANN - Framtiden

"One of the fertile areas for quantum computing is AI (Artificial Intelligence), which relies on processing huge amounts of complex datasets."
- [Tom Taulli](#)

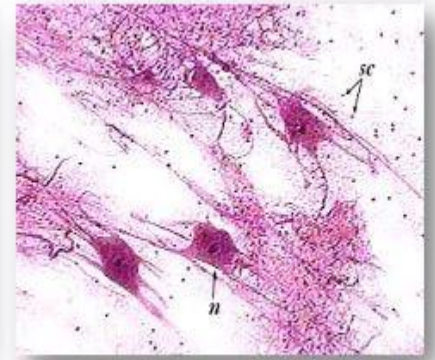
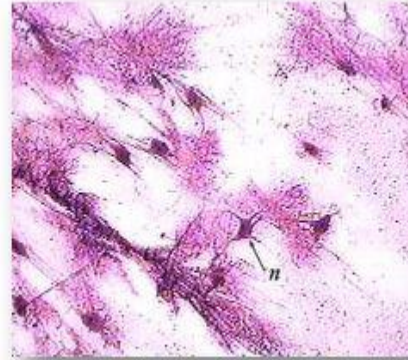
Utveckling av
processorkraft.

Hur passar
quantum
computing in?



Neurala Nätverk - Vad

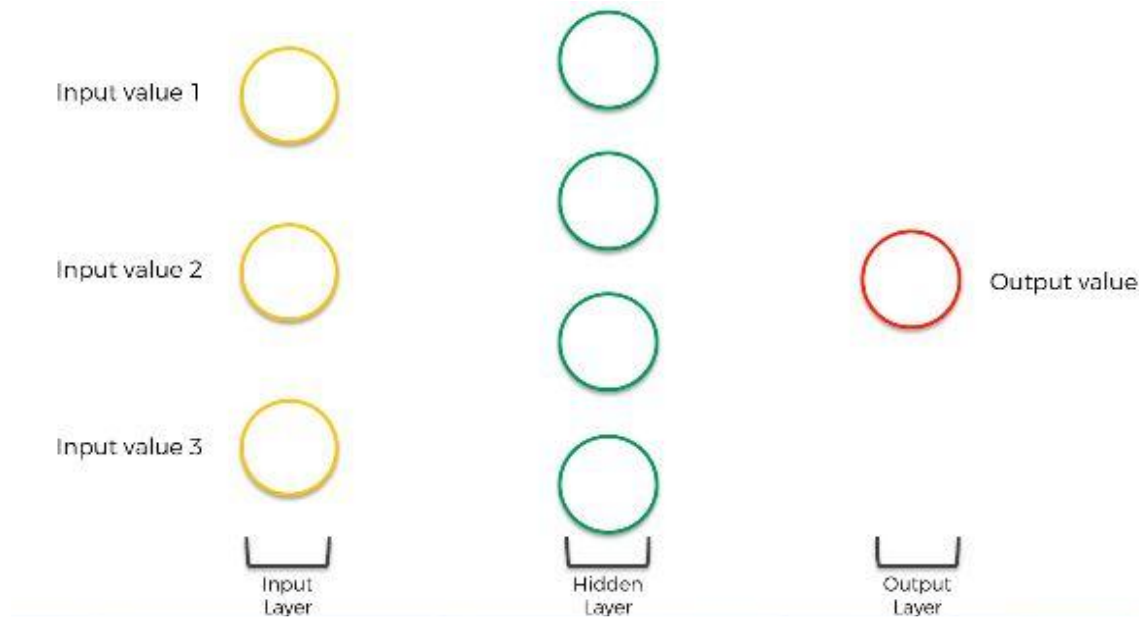
- Efterlikna hjärnans funktion.
- Neuron - neurala kopplingar.
- 100 biljoner kopplingar.



ANN - Delar

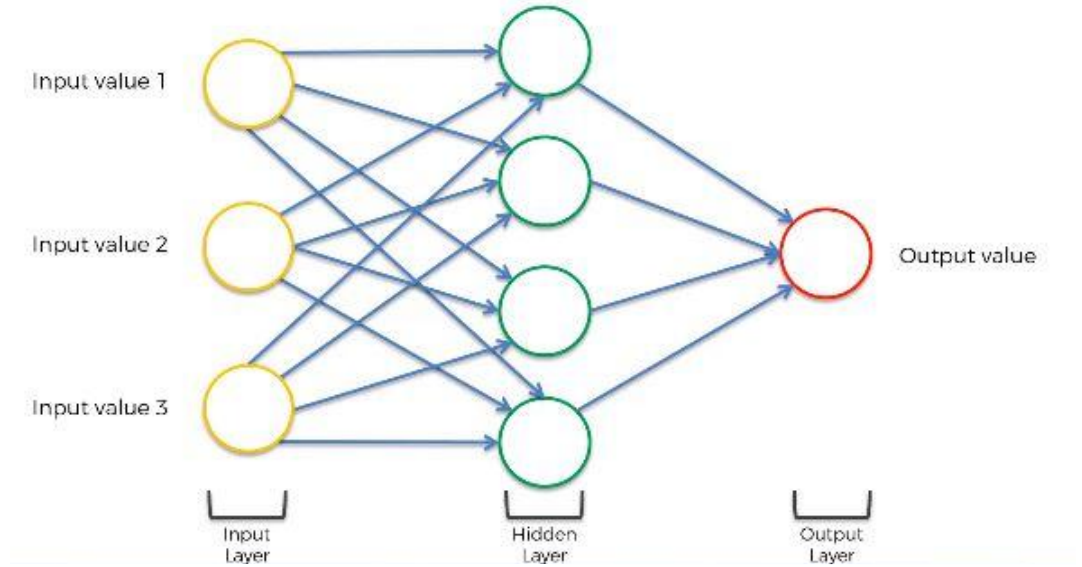
Enkelt neuralt nätverk

- Input
- "Gömt" lager
- Output



ANN - Delar

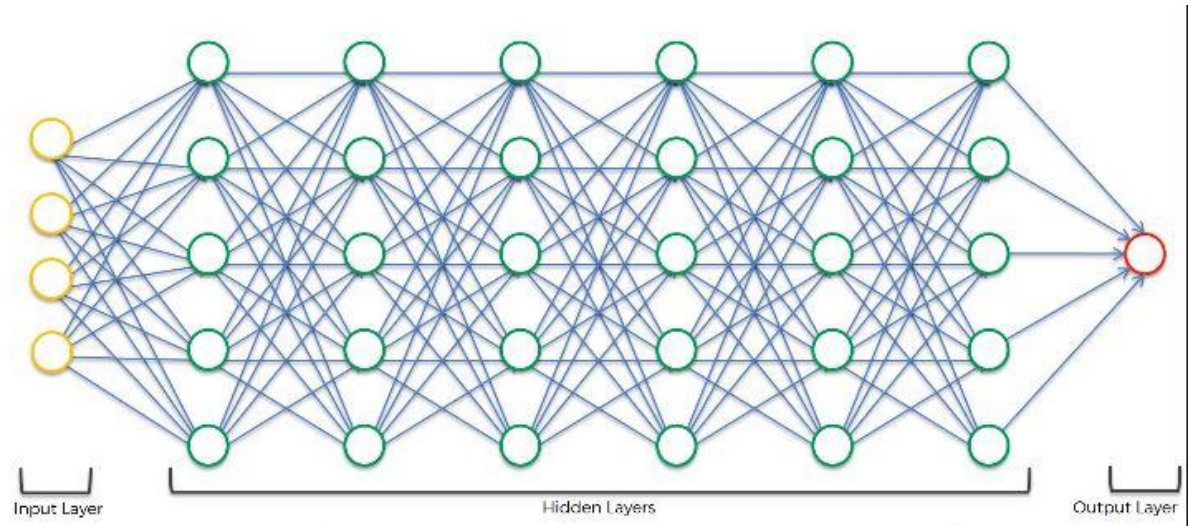
Lägg till förbindelser.



ANN - Delar

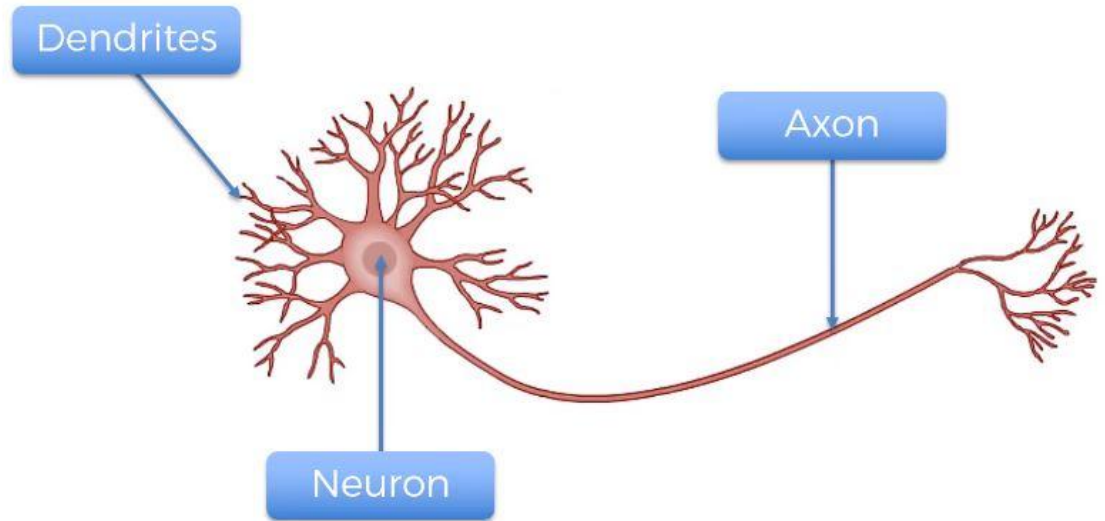
Djupt neuralt nätverk

Flera "gömda" lager.



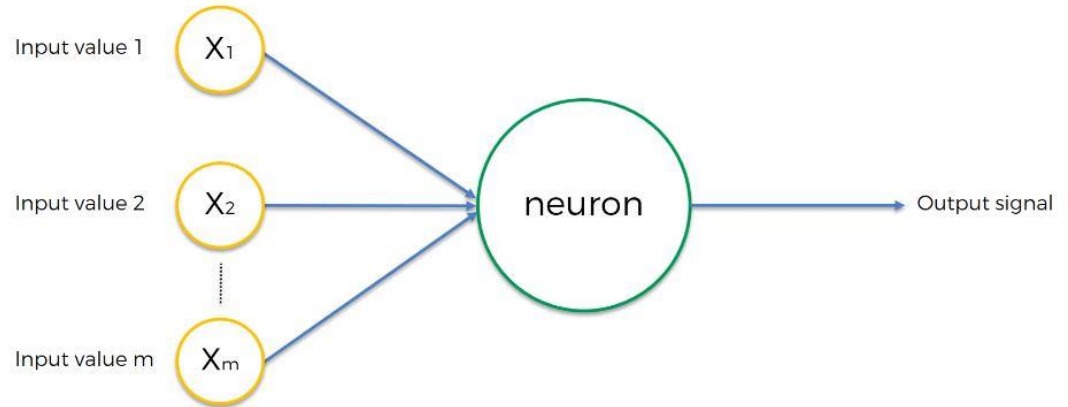
ANN - Neuron

Schematisk bild av neuron



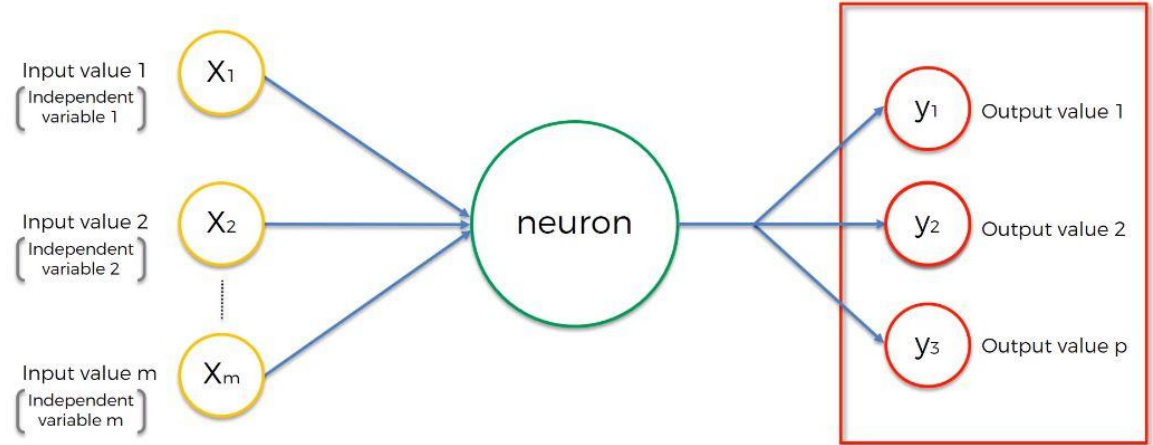
ANN - Delar

Matematisk representation av neuron.



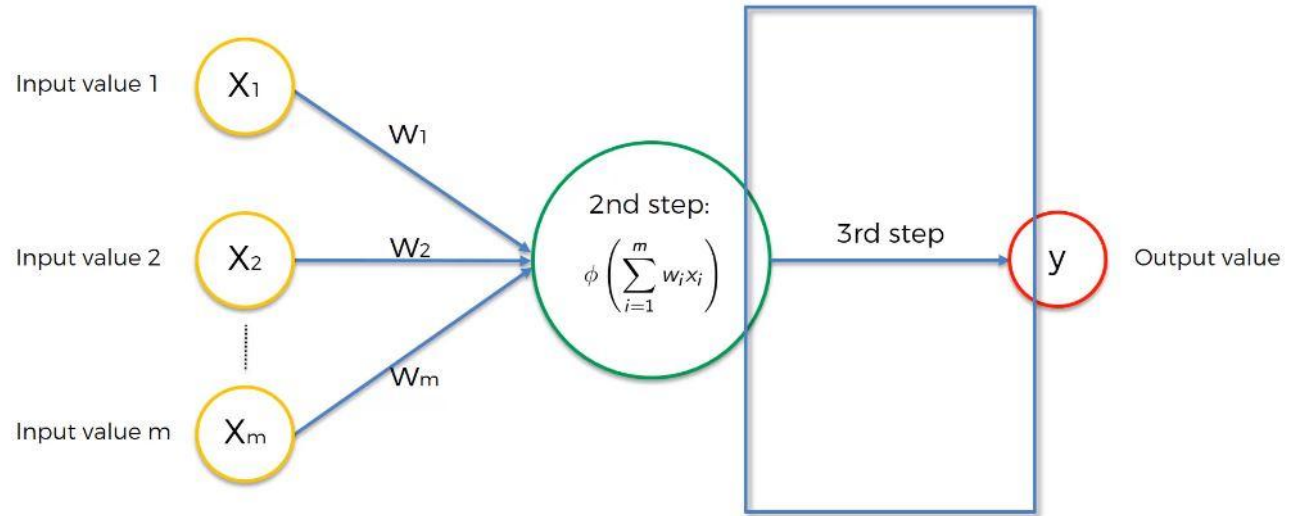
ANN - Delar

Flera outputs - kategorier



ANN - Delar

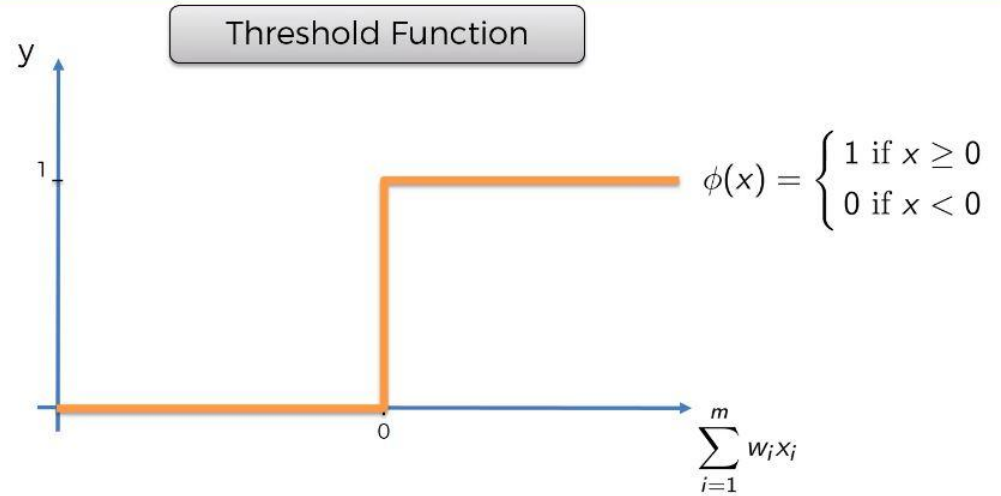
Beräkningssteg



ANN - Aktiveringsfunktion

Steg funktion.

Passar en binär klassifikation.



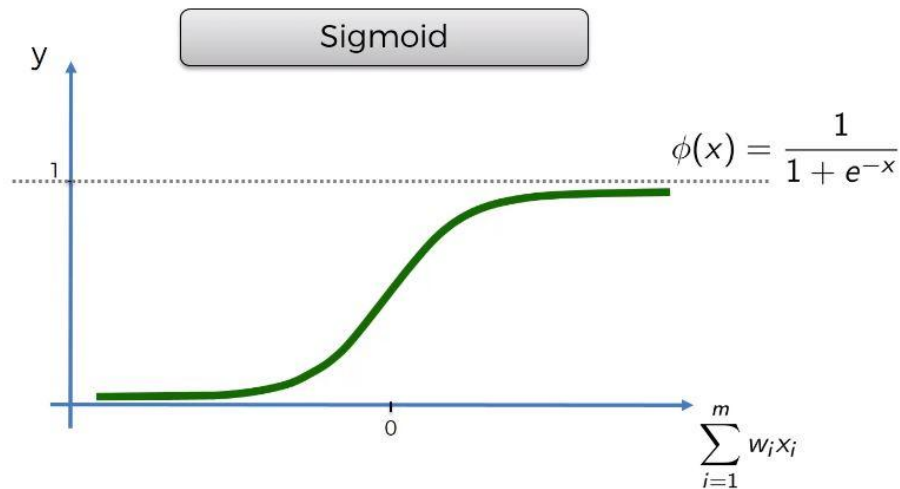
ANN - Aktiveringsfunktion

Sigmoid

Har vanligtvis en output mellan [0, 1].

Passar ett binärt problem.

Ger sannolikheten för ett visst val.

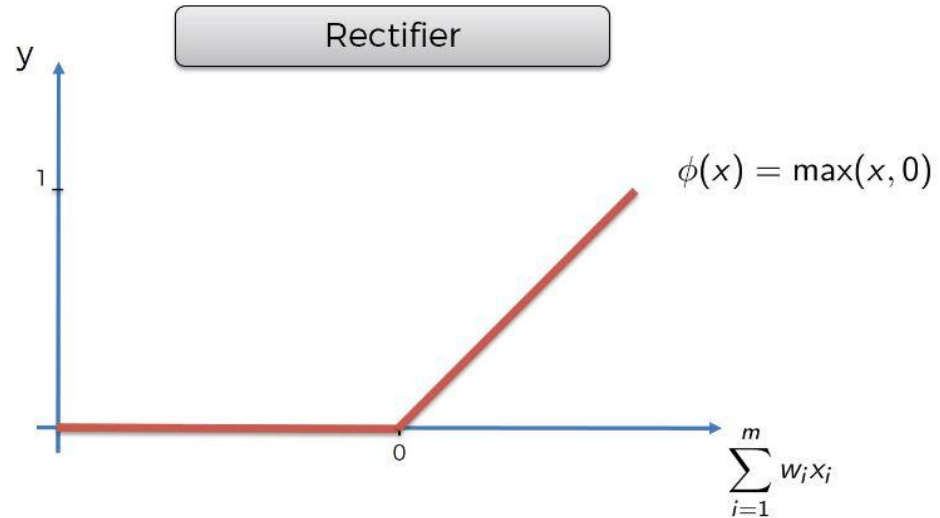


ANN - Aktiveringsfunktion

Relu

Output mellan $[0, \infty]$

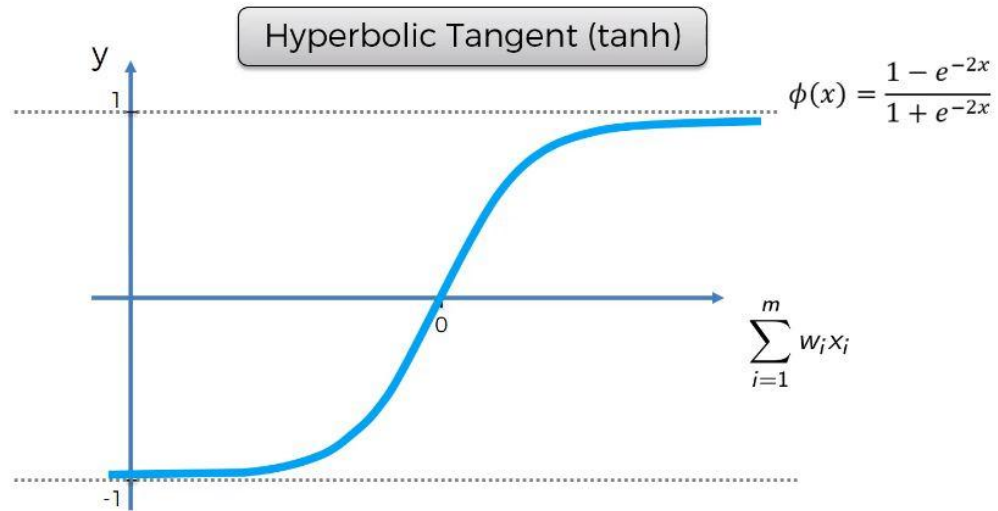
Används mycket i CNN.



ANN - Aktiveringsfunktion

Tangens Hyperbolicus

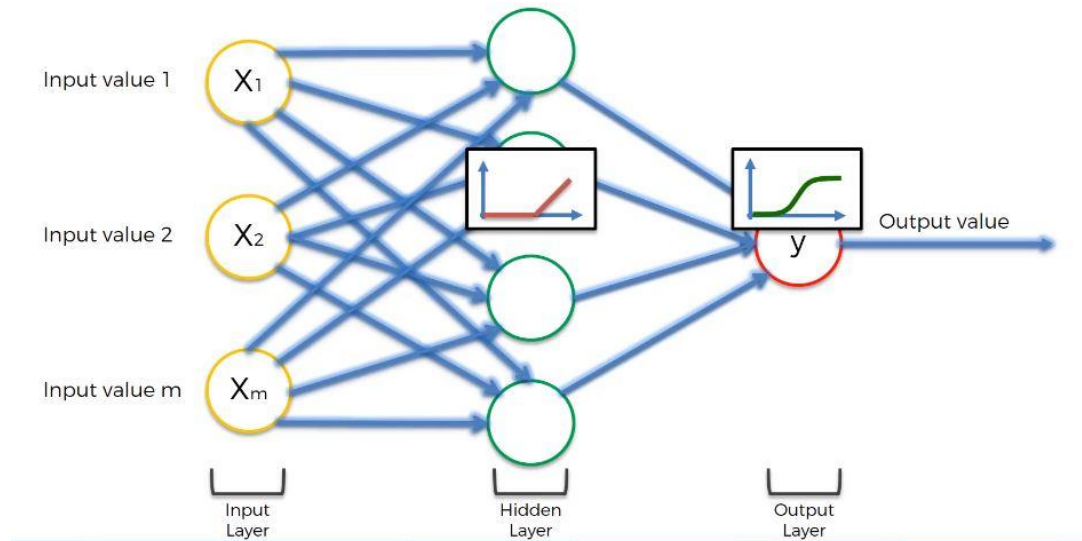
Värden mellan [-1, 1].



ANN - Aktiveringsfunktioner

Hur används aktiveringsfunktioner.

Vanligt med en Relu funktion i de gömda lagren och en Sigmoid för att få sannolikheten.





Hur fungerar ett ANN

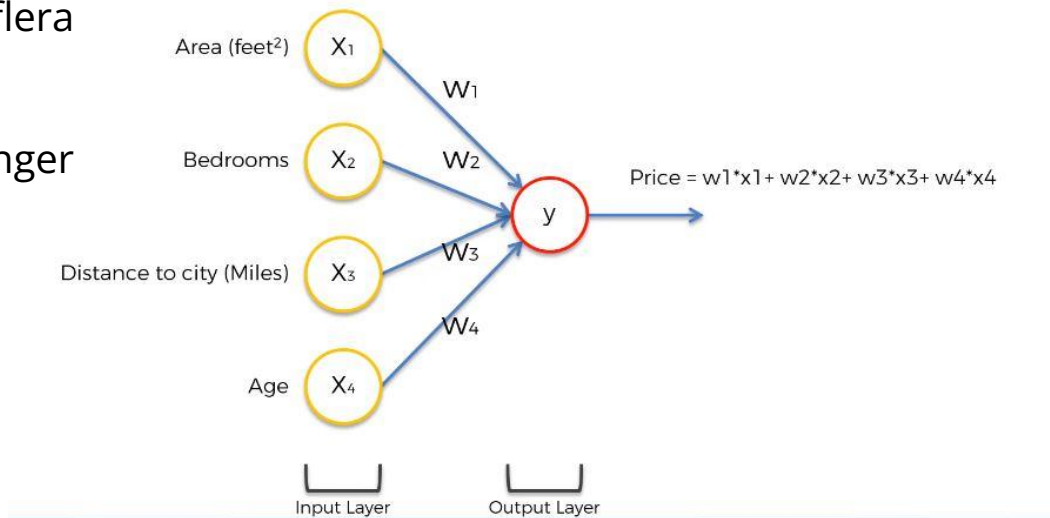


RESTRY...

ANN - Vikter

Den mest enkla neurala nätverket: flera input kopplade direkt till ett output.

Output är summan av varje input gånger sin vikt.



ANN - Vikter

Med gömda lager

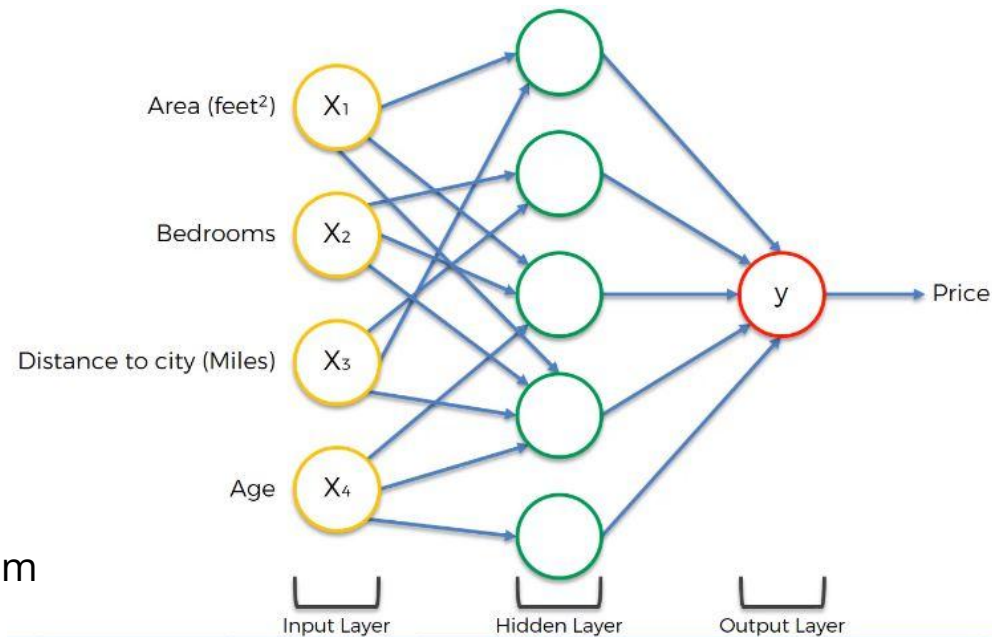
Mer kraftfullt och flexibelt.

Kopplingarna styrs av viktens storlek.

Hur många kopplingar har varje "gömd" neuron?

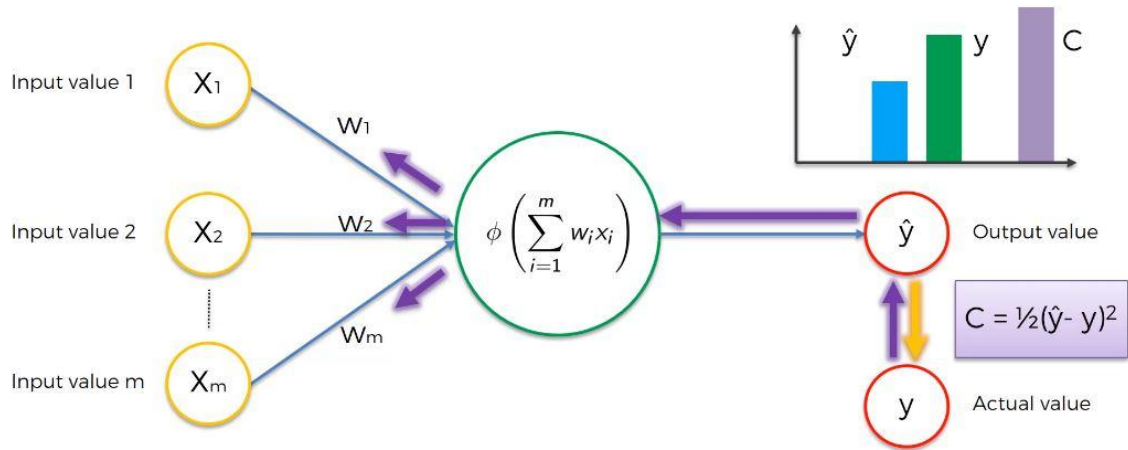
Det gömda lagret skapar nya attribut som är en kombination av sina starkaste input.

Detta gör hela nätverket mer flexibelt.



*"Helheten är större än
summan av sina delar."*

ANN - Perceptron



“Sjämlärande”

Kostnadsfunktion - räknar ut skillnaden mellan det riktiga värdet och modellens värde. Dvs. error.

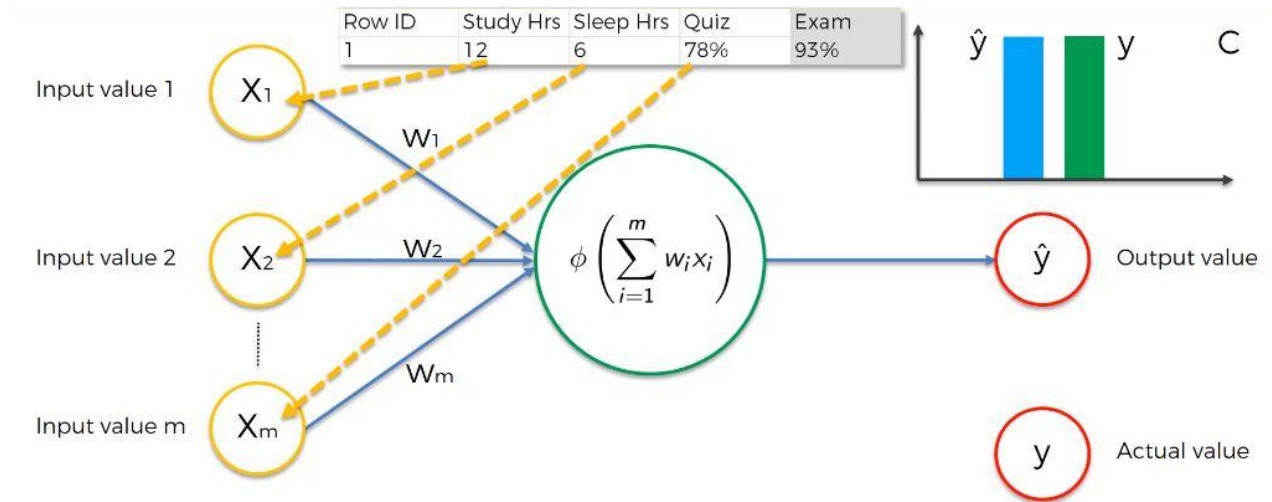
Framåt propagering.

Feedback för att uppdatera vikterna.

ANN - Perceptron

Processen upprepas tills kostnadsfunktionen är tillräckligt litet.

En runda = en epoch.



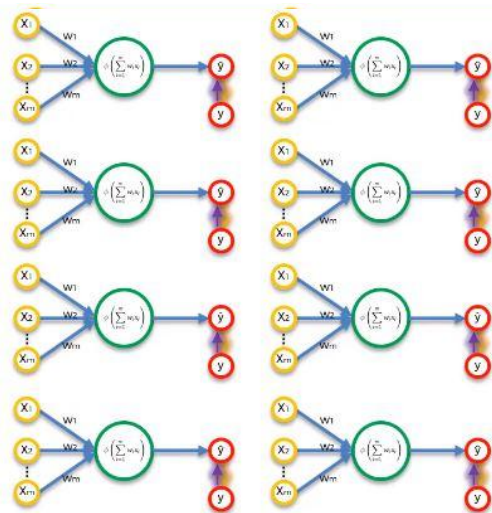
ANN - Perceptron

Evaluera många värden på en gång.

Kombinerad kostnadsfunktion.

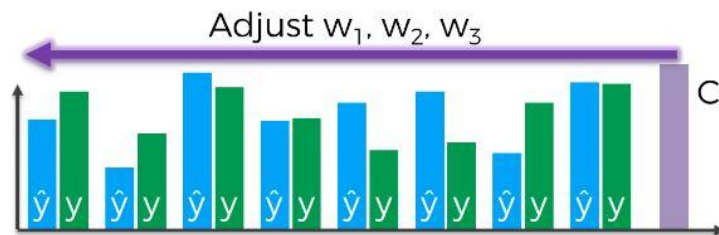
Bakåt propagering - uppdatera vikterna.

Minimera error.



Row ID	Study Hrs	Sleep Hrs	Quiz	Exam
1	12	6	78%	93%
2	22	6.5	24%	68%
3	115	4	100%	95%
4	31	9	67%	75%
5	0	10	58%	51%
6	5	8	78%	60%
7	92	6	82%	89%
8	57	8	91%	97%

$$C = \sum \frac{1}{2}(\hat{y} - y)^2$$



RESTORY...



Gradient Descent

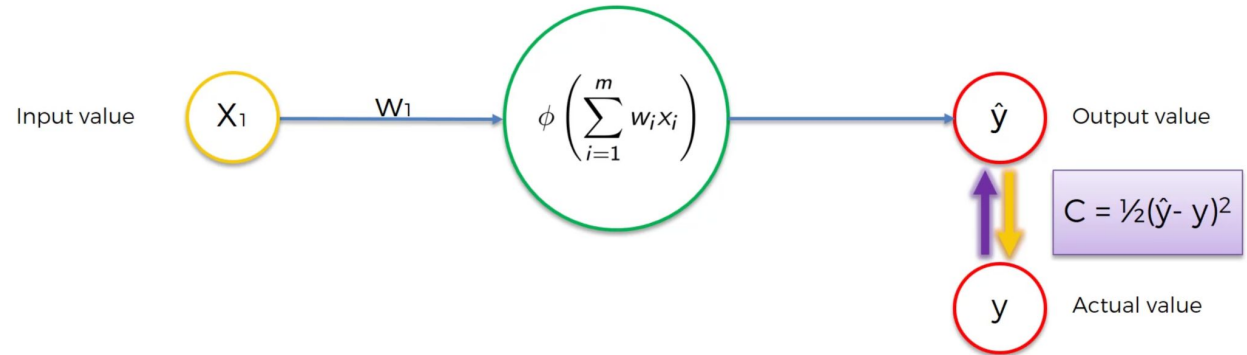


RESTORY...

ANN - Gradient Descent

En standard neuron

Hur kan vi minimera errorn?



RESTOR\...

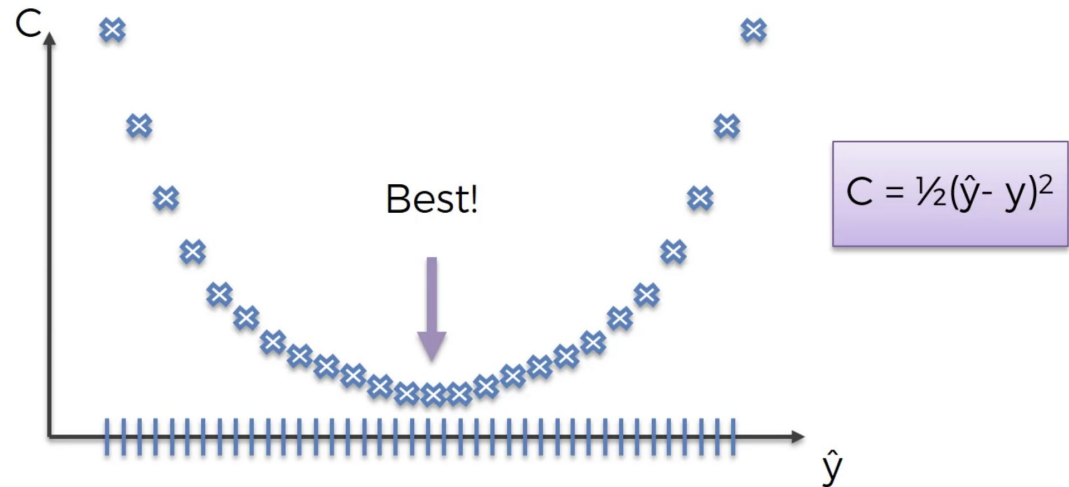
ANN - Gradient Descent

Kostnadsfunktion beroende på vikter.

Hur uppdaterar vi vikterna?

Testa en massa olika vikter?

Vilket problem kommer vi få då?



RESTOR\...

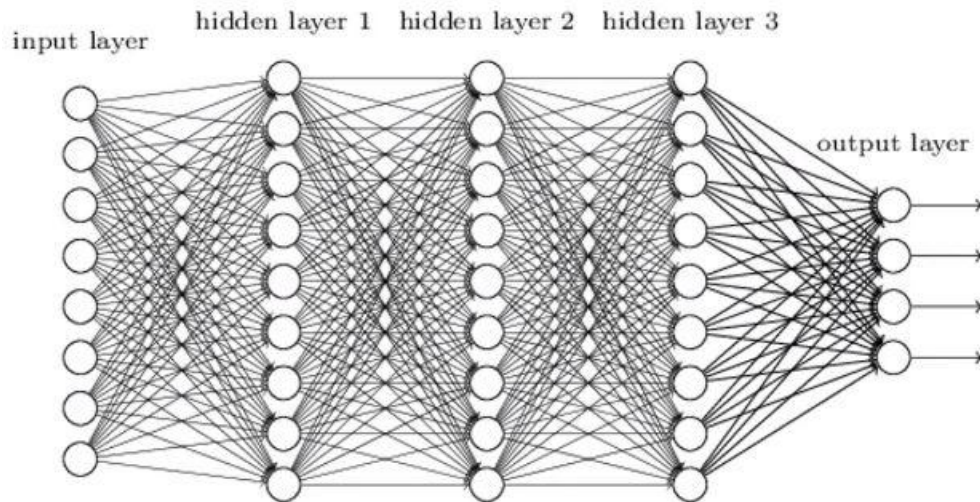
Neurala Nätverk

“The Curse of Dimensionality”

Alla neuroner i ett lager, kopplade till varje neuron i nästa lager, osv.

Beräkningarnas antal växer snabbt.

Hur många operationer endast i detta nätverk om vi testade 10 olika viktuppdateringar per vikt i alla lager?



ANN - Gradient Descent

Nedstigning utmed gradient

Vad är linjens lutning?

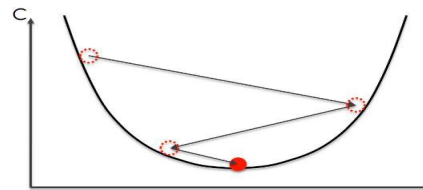
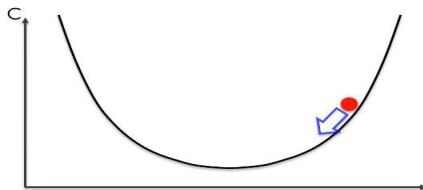
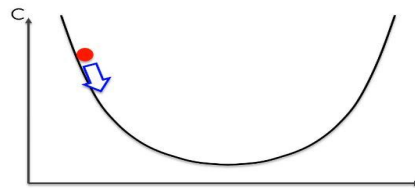
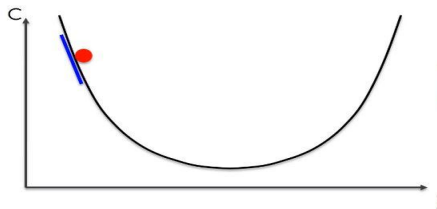
Negativ - neråt i kurvan

Positiv - uppåt

Hjälper oss uppdatera vikterna med små steg.

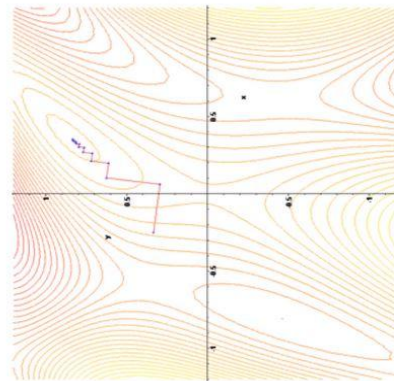
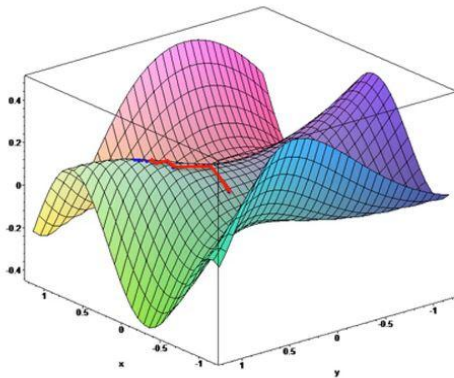
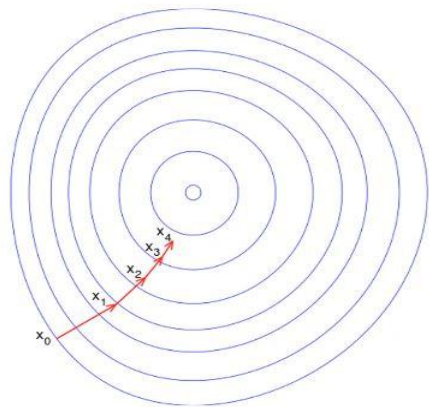
Stegens storlek minskar ju närmare "minima".

$$C = \frac{1}{2}(\hat{y} - y)^2$$



RESTRY...

ANN - Gradient Descent



RESTORY...



Stochastic Gradient Descent



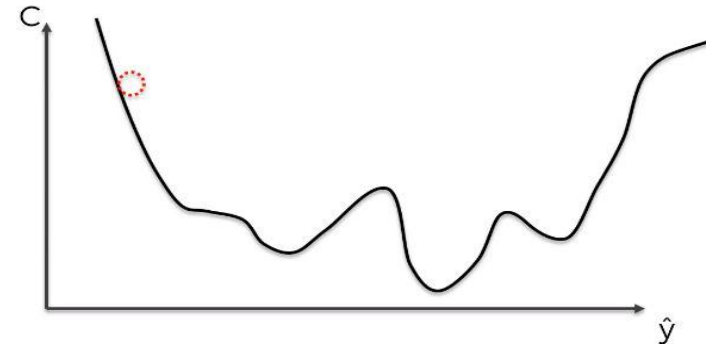
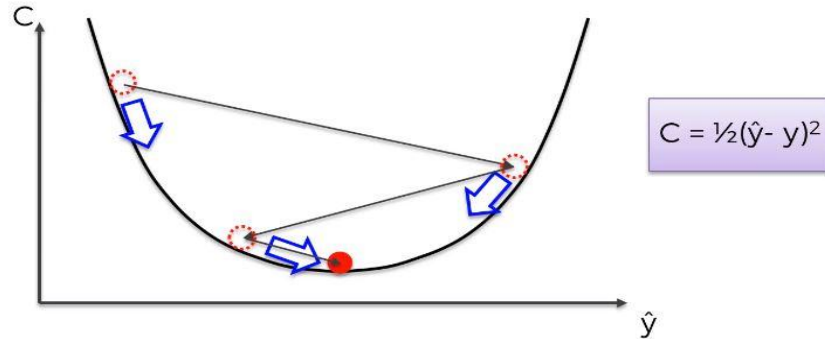
RESTORY...

ANN - Stochastic Gradient Descent

Flera "gropar" - mer komplex kostnadsfunktion eller multidimensionellt.

Inlärningshastighet – Learning rate

"Storleken på steget"



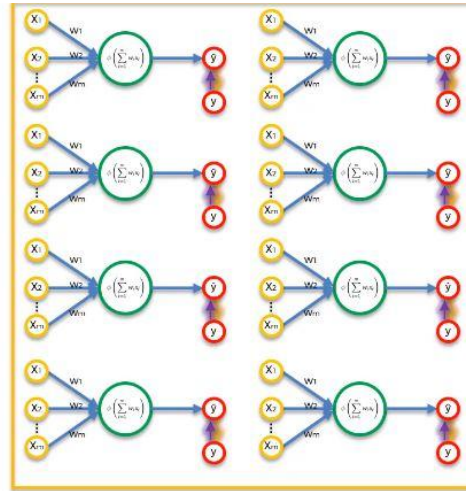
Lokal minima.

RESTOR\...

ANN - Gradient Descent

Original Gradient Descent

Batch - "alla" rader processas innan vikterna uppdateras.



Row ID	Study Hrs	Sleep Hrs	Quiz	Exam
1	12	6	78%	93%
2	22	6.5	24%	68%
3	115	4	100%	95%
4	31	9	67%	75%
5	0	10	58%	51%
6	5	8	78%	60%
7	92	6	82%	89%
8	57	8	91%	97%

$$C = \sum \frac{1}{2}(\hat{y} - y)^2$$



RESTOR...

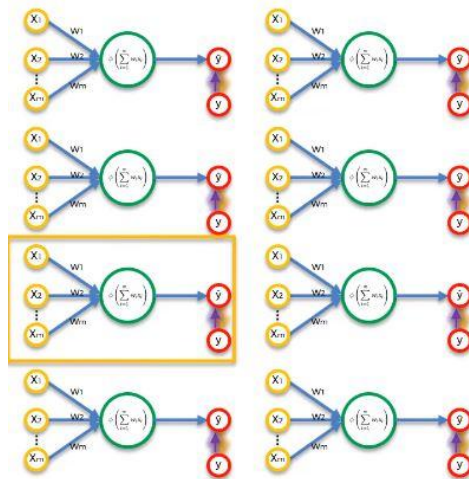
ANN - Stochastic Gradient Descent

En rad i taget.

Stokastisk (slumpmässig).

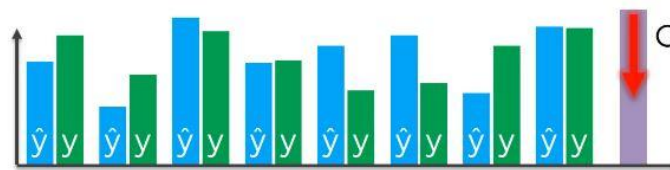
Kostnadsfunktionen behöver ej vara korrekt.

Uppdaterar vikterna per rad eller i "mini-batch".



Row ID	Study Hrs	Sleep Hrs	Quiz	Exam
1	12	6	78%	93%
2	22	6.5	24%	68%
3	115	4	100%	95%
4	31	9	67%	75%
5	0	10	58%	51%
6	5	8	78%	60%
7	92	6	82%	89%
8	57	8	91%	97%

$$C = \sum \frac{1}{2}(\hat{y} - y)^2$$

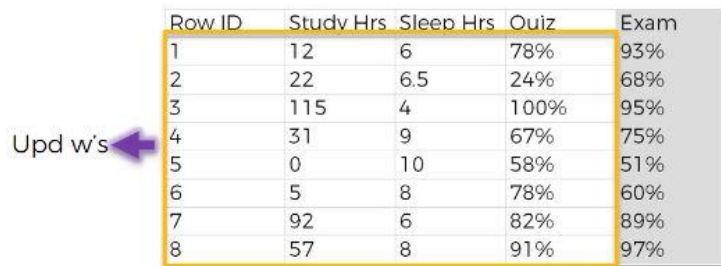


RESTORY...

ANN - SGD vs Batch

Batch

- Deterministisk - samma resultat, om samma vikter.



A diagram illustrating Batch Stochastic Gradient Descent. It features a table with 8 rows of data. A purple arrow labeled 'Upd w's' points to the entire table, indicating that all data points are used for each weight update.

Row ID	Study Hrs	Sleep Hrs	Quiz	Exam
1	12	6	78%	93%
2	22	6.5	24%	68%
3	115	4	100%	95%
4	31	9	67%	75%
5	0	10	58%	51%
6	5	8	78%	60%
7	92	6	82%	89%
8	57	8	91%	97%

Batch

SGD

- Har mycket större fluktuationer per uppdatering.
- Kan "hoppa ur" en lokal minima. Hitta global minima lättare.
- Väljer rader slumpmässigt.
- Snabbare.



A diagram illustrating Stochastic Gradient Descent. It features a table with 8 rows of data. To the left of the table, the text 'Upd w's' is repeated 8 times, with a purple arrow pointing to a single row of the table for each instance, indicating that only one data point is used for each weight update.

Row ID	Study Hrs	Sleep Hrs	Quiz	Exam
1	12	6	78%	93%
2	22	6.5	24%	68%
3	115	4	100%	95%
4	31	9	67%	75%
5	0	10	58%	51%
6	5	8	78%	60%
7	92	6	82%	89%
8	57	8	91%	97%

Stochastic

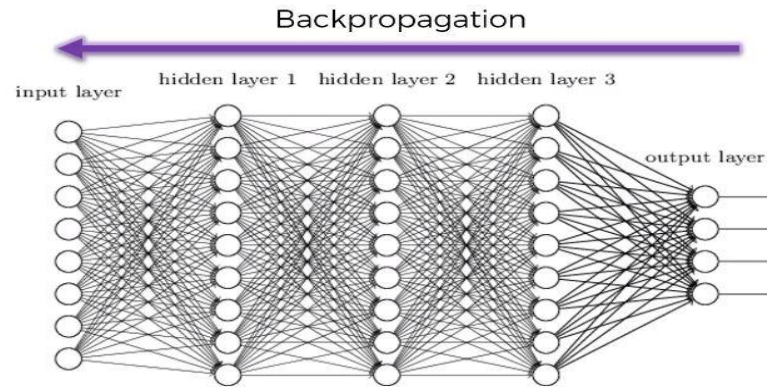
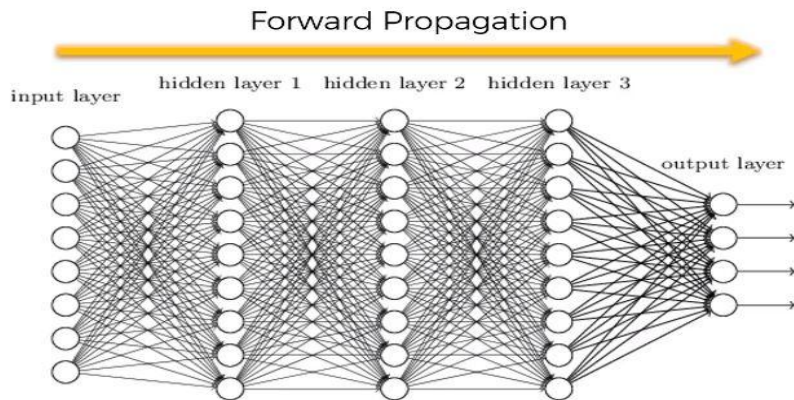
RE STORY...

Neurala Nätverk - Propagation

Input - neuroner - lager - output

Jämför output med riktigt värde - räkna ut error

Alla vikter uppdateras på en gång.



Neurala Nätverk - Träning

1. Initiera slumpmässigt vikterna till tal nära, men inte lika med 0.
2. Stoppa in den första observationen av datasetet i inmatningslagret (input layer), varje dataegenskap (feature) motsvaras av en inmatningsnod (input node).
3. Framåt propagering (forward-propagation). Från vänster till höger aktiveras neuronerna på ett sätt så att varje neurons aktiveringsfunktion (activation) begränsas av vikterna
4. Jämför det slutgiltigt beräknade resultatet med det verkliga resultatet. Mät felet som genereras. (Kostnadsfunktion)
5. Bakåt propagering (back-propagation) Felen propageras bakåt från höger till vänster. Vikterna anpassas efter hur mycket det bidrar till felet
6. Repetera steg 1 till 5 för varje observation
7. När hela träningsmängden har gått genom nätverket säger man att en epok (epoch) avverkats. Repetera epoker och observera hur felet minskar

Neurala Nätverk - Stochastic Gradient Descent

STEP 1: Randomly initialise the weights to small numbers close to 0 (but not 0).



STEP 2: Input the first observation of your dataset in the input layer, each feature in one input node.



STEP 3: Forward-Propagation: from left to right, the neurons are activated in a way that the impact of each neuron's activation is limited by the weights. Propagate the activations until getting the predicted result y .



STEP 4: Compare the predicted result to the actual result. Measure the generated error.



STEP 5: Back-Propagation: from right to left, the error is back-propagated. Update the weights according to how much they are responsible for the error. The learning rate decides by how much we update the weights.



STEP 6: Repeat Steps 1 to 5 and update the weights after each observation (Reinforcement Learning). Or:
Repeat Steps 1 to 5 but update the weights only after a batch of observations (Batch Learning).



STEP 7: When the whole training set passed through the ANN, that makes an epoch. Redo more epochs.

ANN - Keras

- Keras är ett "high level" bibliotek skrivet specifikt för deep learning.
- Sequential model - varje "layer" matas in i nästa i den ordning vi bestämt.
- Finns "non-sequential" modeller, kallas "functional". Kan dela sig.

"A "model" is a directed acyclic graph of layers. You can think of a model as a "bigger layer" that encompasses multiple sublayers and that can be trained via exposure to data."

- Keras.io

ANN - Modellen

- Varje "layer" är ett "Dense" objekt - helt kopplat till nästa lager.
- Dense funktionen tar 3 parametrar:
 - antal noder i det lagret
 - "input shape" - endast för första lagret.
 - aktiveringsfunktion
- Välja antalet neuroner i det gömda lagret - testa lite olika.
- Skapa modellen och sedan lägg till lager
 - `add(Dense(...))`

- Compile
 - optimizer - uppdaterar vikterna
 - `adam` - SGD
 - loss - hur skillnaden mellan modellens värde och det riktiga värdet räknas ut
 - för binär klassifikation - `binary_crossentropy`
 - för multiklass - `categorical_crossentropy` (måste då även ändra activation till `softmax`)
 - metrics - en lista på hur du vill evaluera ANN nätverket
 - `["accuracy"]`
- Train - fit
 - `batch_size` = 32 som standard.
 - `epochs` - ej för litet, 100 som standard, experimentera för att se om du får bättre resultat med annat värde.

Länkar

- [DNA - A storage solution](#)
 - [Quantum computing and AI](#)
 - [Keras for beginners](#)
 - [Sequential vs functional models](#)
 - [Keras sequential model explained](#)
 - [Neural Network in 13 lines](#)
 - [Neural networks & DL - Bok!](#)
 - [ANN in 5 minutes](#)
 - [What is an ANN? - 3B1B](#)
 - [Gradient descent](#)
 - [Stochastic Gradient Descent](#)
- [Keras & Sklearn](#)