

Gymnasielærerdag 2024

AI-projektet ved Ege Rubak og Malene Cramer Engebjerg



Program

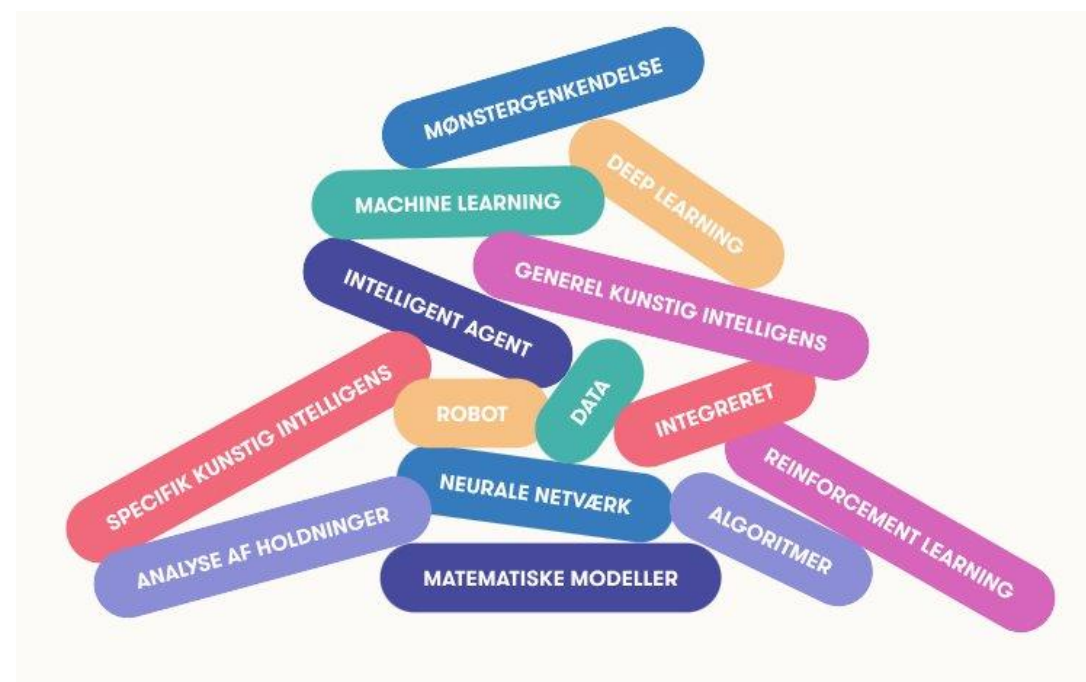
- Hvad er AI?
- AI projektet
- Præsentation af aimat.dk
- Eksempler på forløb (korte og lange)
- SRP
- Perceptroner
- Simple neurale netværk
- Kunstige neurale netværk
- Overfitting og krydsvalidering



Hvad er kunstig intelligens



- Der er ingen definition! Og den løse definition der måtte være ændrer sig over tid.
- For tiden er generativ AI og specifikt chatGPT for mange lig "rigtig" AI.
- Vi fortolker det bredt:
 - Metoder/algoritmer der kan implementeres på en computer og bruges til prædiktion/beslutningsstøtte på baggrund af data
 - Inkluderer f.eks. lineær regression



Fra elementsofai.dk



AI projektet



- Projektet Aalborg Intelligence, som er finansieret af Novo Nordisk Fonden, er forankret på Institut for Matematiske Fag på Aalborg Universitet (AAU), og inkluderer en repræsentant fra de fem STX-gymnasier i Aalborg.

Forskere fra AAU

- Ege Rubak (projektleder)
- Lisbeth Fajstrup
- Anne Marie Svane
- Søren Højsgaard

Gymnasielærere

- Malene Cramer Engebjerg (Aalborghus Gymnasium)
- Jan B. Sørensen (Aalborg City Gymnasium)
- Allan Frendrup (Nørresundby Gymnasium)
- Nikolaj Hess-Nielsen (Katedralskolen)
- Mette Kristensen (Hasseris Gymnasium)



Kernestof i AI-projektet

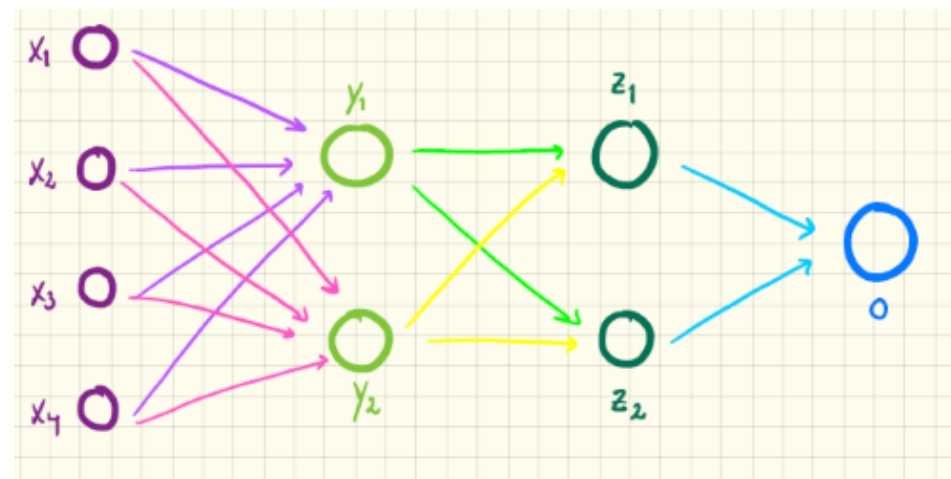
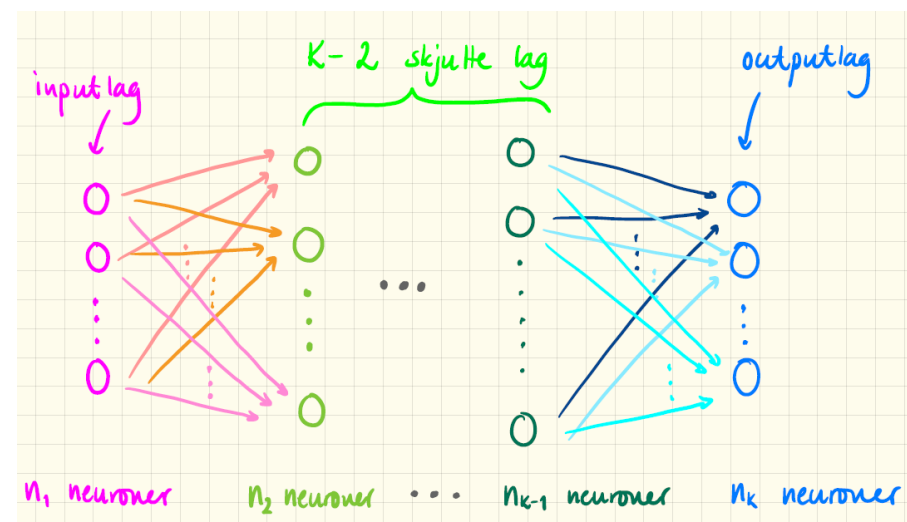


- Vi bruger AI-metoder som motiverende eksempler til at lære/træne en masse kernestof:
 - Koordinatsystemer og afstande
 - Sammensatte funktioner
 - Sandsynlighedsregning
 - Differentiation
 - Lineær regression
 - Polynomier
 - ...



Hvilken gymnasiematematik indgår?

- Et kunstigt neuralt netværk er en kæmpestor sammensat *funktion*.
- Justering af parametre – læring – anvender kædereglen.





Aktiveringsfunktioner

- Funktioner i neurale netværk:
Aktiveringsfunktioner

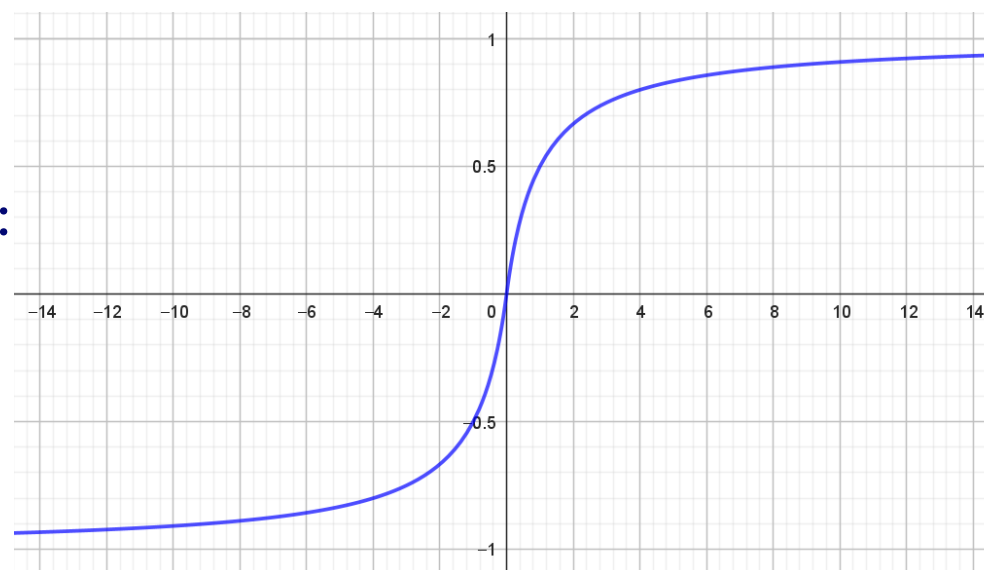
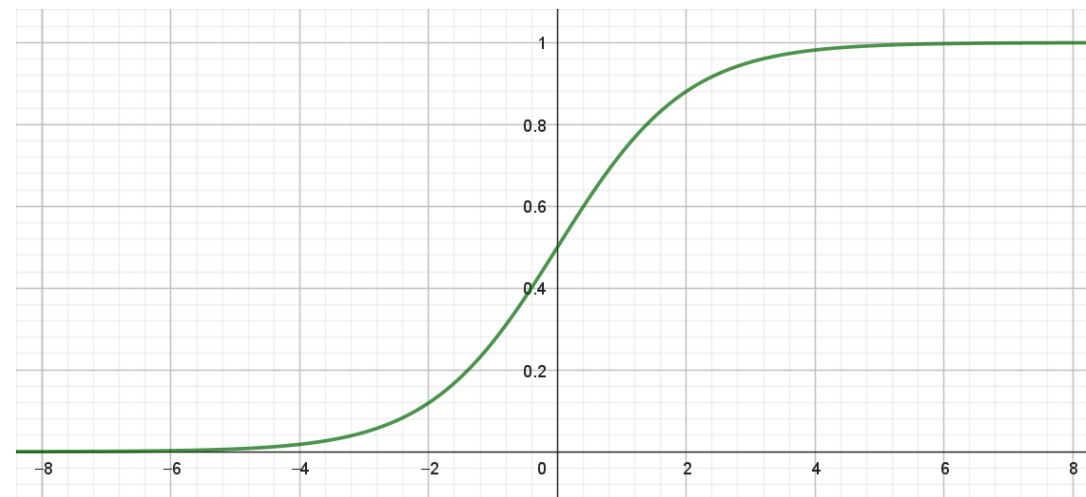
Sigmoid: $\sigma(x) = \frac{1}{1+e^{-x}}$

SoftSign: $f(x) = \frac{x}{1+|x|}$

Den afledte udregnes fra funktionsværdierne:

$$\sigma'(x) = \sigma(x) \cdot (1 - \sigma(x))$$

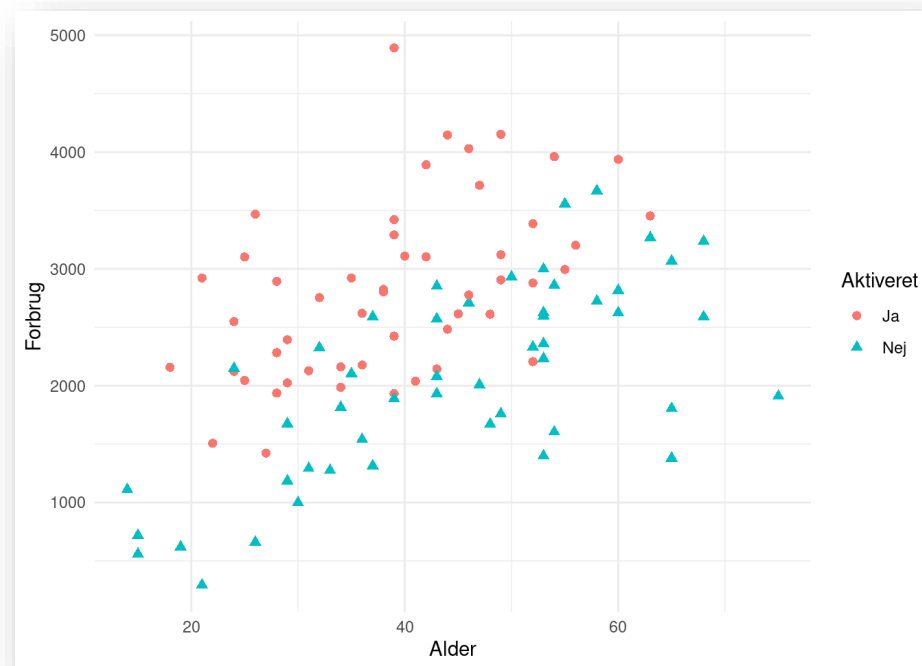
$$f'(x) = (1 - |f(x)|)^2$$





Klassifikation og geometri

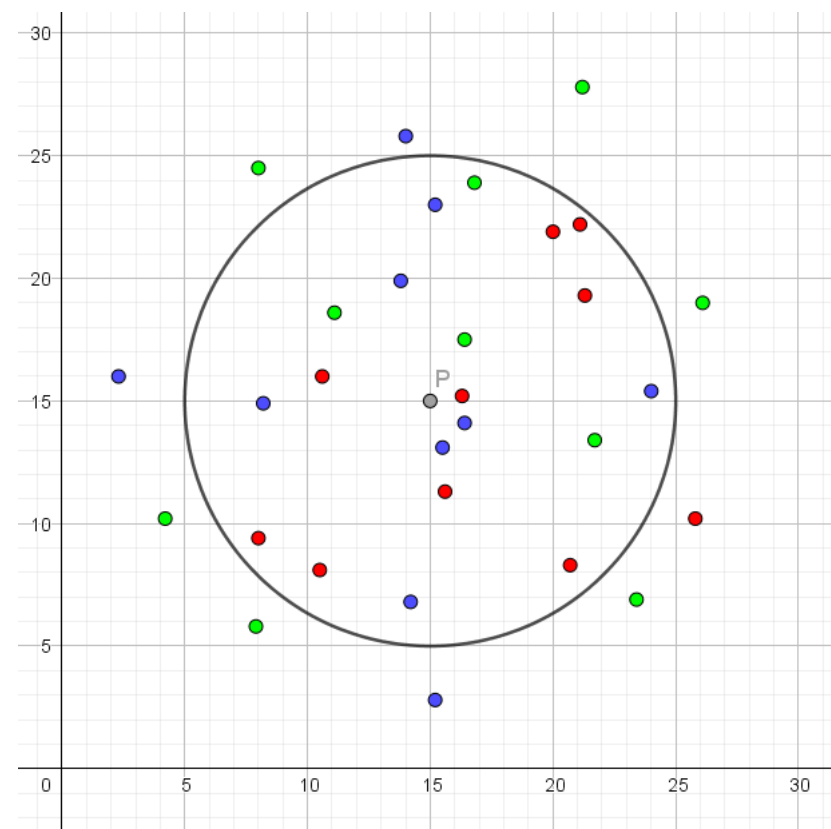
- Find den linje, der *bedst* opdeler kendt data (træningsdata).
- Nyt punkt klassificeres efter, hvilken side af linjen, det ligger på.
- OBS! Bedste rette linje er ikke det, eleverne plejer at se.





Geometri og klassifikation

- Eller: Klassificer det nye punkt efter, hvilke punkter (fra træningsdata), der ligger i nærheden.
- De k nærmeste naboer - kNN
- Flertallet blandt naboerne.
- Hvad skal k være?
Krydsvalidering.







aimat.dk



MAT

MATEMATIKKEN BAG MAGIEN

 Undervisningsforløb Materialer SRO SRP Apps Referencer Om os 

Hvad er AI-matematik? Det er al den spændende matematik, som ligger bag kunstig intelligens!

På denne side kan du lære, hvordan gymnasimatematikken bruges i en masse former for kunstig intelligens. Til gengæld kan du *ikke* lære, hvordan du får en kunstig intelligens til at løse dine matematikopgaver!

Undervisningsforløb

Forskellige undervisningsforløb til matematik i gymnasiet, som inddrager AI. Der findes forløb til både A-, B- og C-niveau.

Materialer

Noter om diverse AI relaterede emner.

SRO

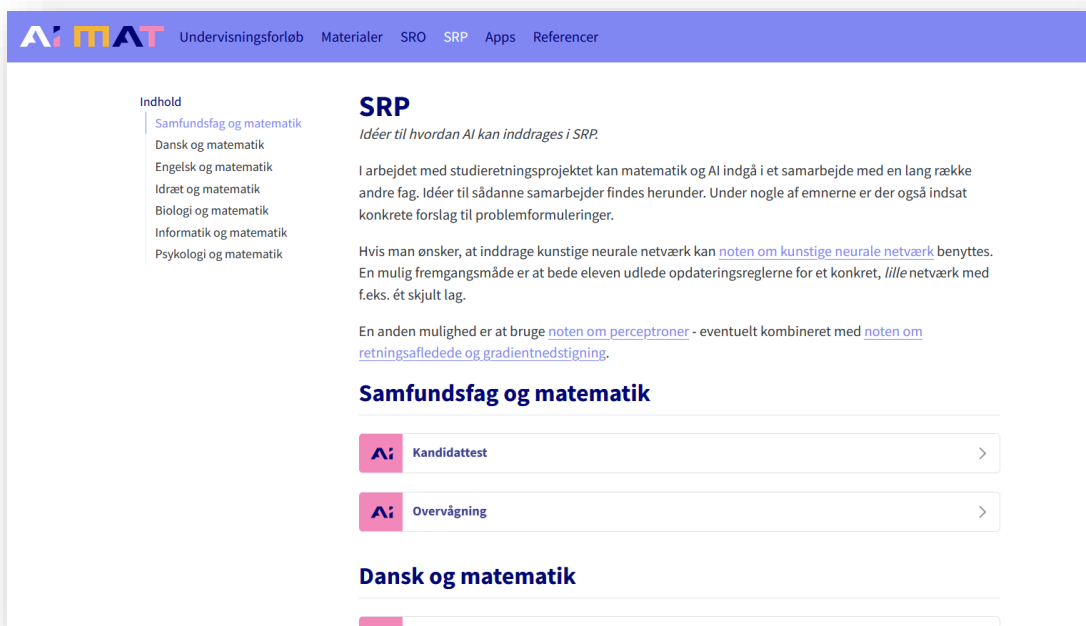
Idéer til hvordan AI kan inddrages i SRO.

SRP

Idéer til hvordan AI kan inddrages i SRP.



SRP / SOP / SSO



- Idéer til store skriftlige opgaver
- Man kan fint lade eleverne arbejde med simple neurale netværk
- Generelle kunstige neurale netværk:

Redegør for hvad der forstås ved et kunstigt neuralt netværk med ét skjult lag samt hvordan et sådant netværk trænes. Forklar hvordan krydsvalidering kan anvendes til at vurdere anvendeligheden af et kunstigt neuralt netværk.



Forløb (korte og lange)



Aktiveringsfunktioner

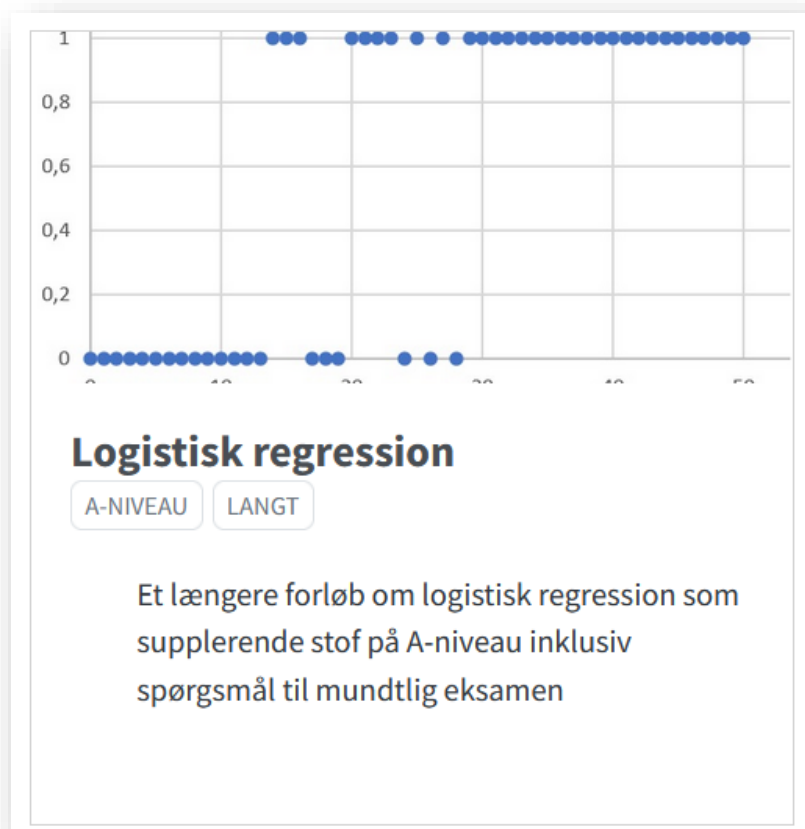
A-NIVEAU KORT

I opbygningen af kunstige neurale netværk er aktiveringsfunktioner helt centrale. Og aktiveringsfunktioner skal differentieres – det handler dette forløb om.

- Differentiation af forskellige aktiveringsfunktioner:
 - Sigmoid
 - Softsign
 - Hyperbolsk tangens
 - ReLU
- Vigtig pointe: For en aktiveringsfunktion f kan $f'(x)$ udtrykkes simpelt ved $f(x)$.



Forløb (korte og lange)



- Baseret på note om logistisk regression med tilhørende videoer.
- Forløbet er delt op i en række dele, som eleverne kan arbejde med mere eller mindre selvstændigt.



Forløb (korte og lange)



Idéer til nye forløb modtages med
kyshånd 😊



What's next?



- Vi arbejder på materialer og forløb om store sprogmodeller 😊
- Glæd jer – det er virkelig smart. Og godt til samarbejde med Almen Sprogforståelse. Matematikken bag er vektorer, vinkler, regression, Bayes (på speed), og meget, meget mere.



Perceptroner



Kan jeg lave et
computerprogram,
som kan genkende
hundebilleder?

Perceptroner

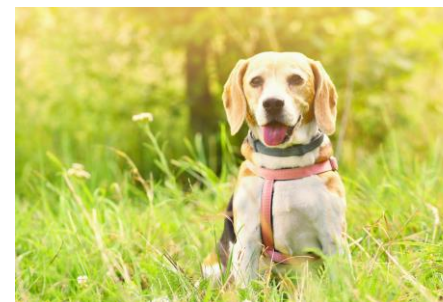
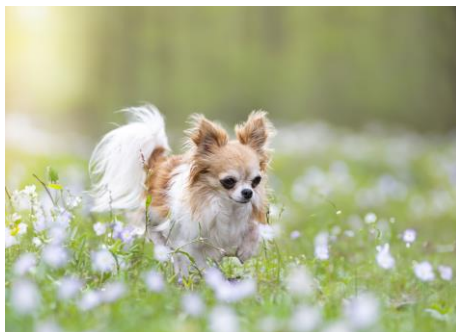
- Hvordan lærte du at genkende hunde på billeder?





Perceptroner – hunde versus mus

- Lad computeren lære at kende forskel på hunde og mus!





Perceptroner – hunde versus mus



- Vi måler nogle *features* og fortæller computeren, om der er en hund eller en mus på billedet





Perceptroner – hunde versus mus



- Vi måler nogle *features* og fortæller computeren, om der er en hund eller en mus på billedet

	Graden af trekantede ører	Halelængde (ift. kropslængde)	Targetværdi
			Hund
			Mus



Perceptroner – hunde versus mus



- Vi måler nogle *features* og fortæller computeren, om der er en hund eller en mus på billedet

	Graden af trekantede ører	Halelængde (ift. kropslængde)	Targetværdi
	0,85		Hund
	0,15		Mus



Perceptroner – hunde versus mus

- Vi måler nogle *features* og fortæller computeren, om der er en hund eller en mus på billedet

	Graden af trekantede ører	Halelængde (ift. kropslængde)	Targetværdi
	0,85	0,25	Hund
	0,15	1,05	Mus



Perceptroner – hunde versus mus

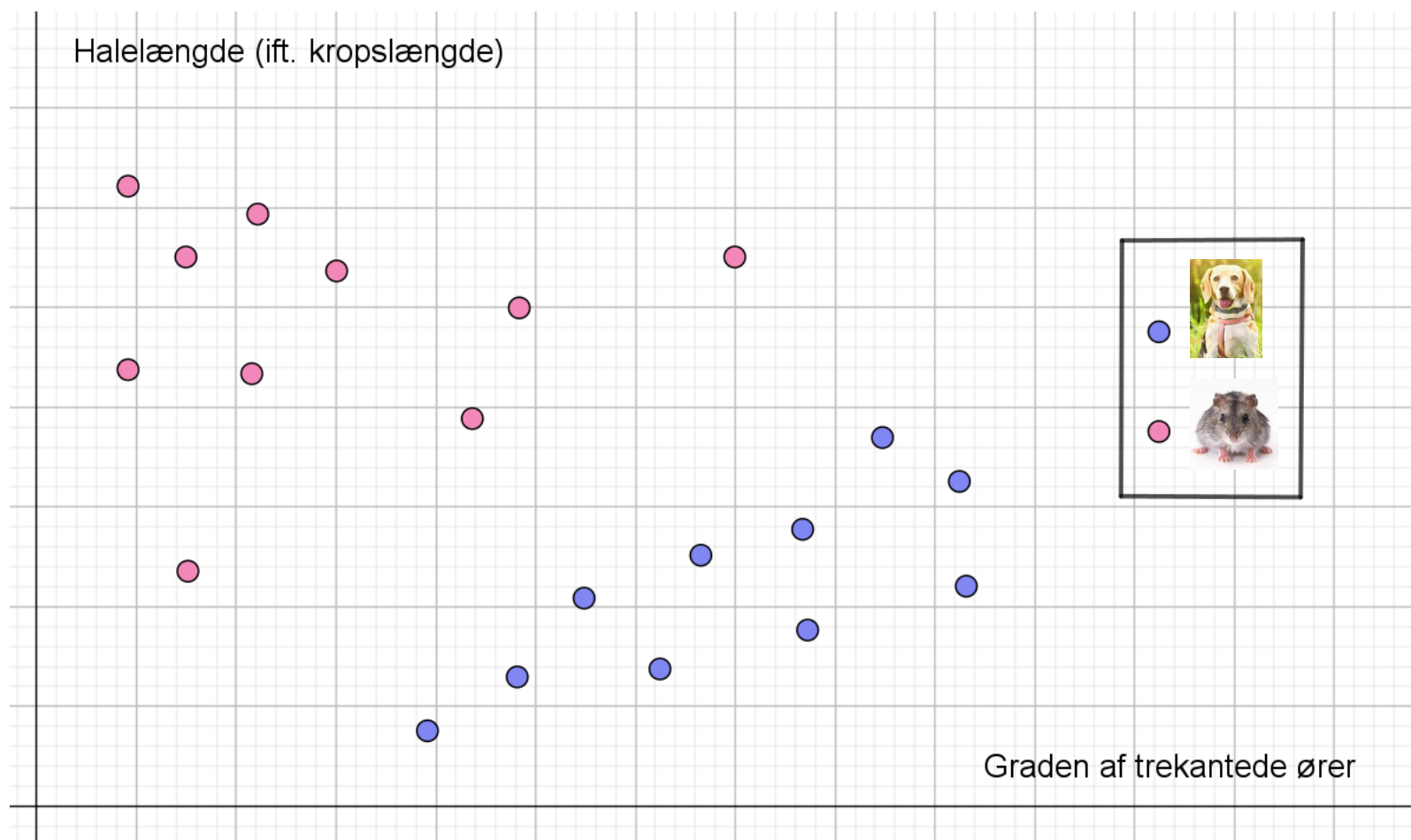
- Vi måler nogle *features* og fortæller computeren om der er en hund eller en mus på billedet

	Graden af trekantede ører	Halelængde (if...	erdi
			Hund
	0,15	1,05	Mus

Træningsdata

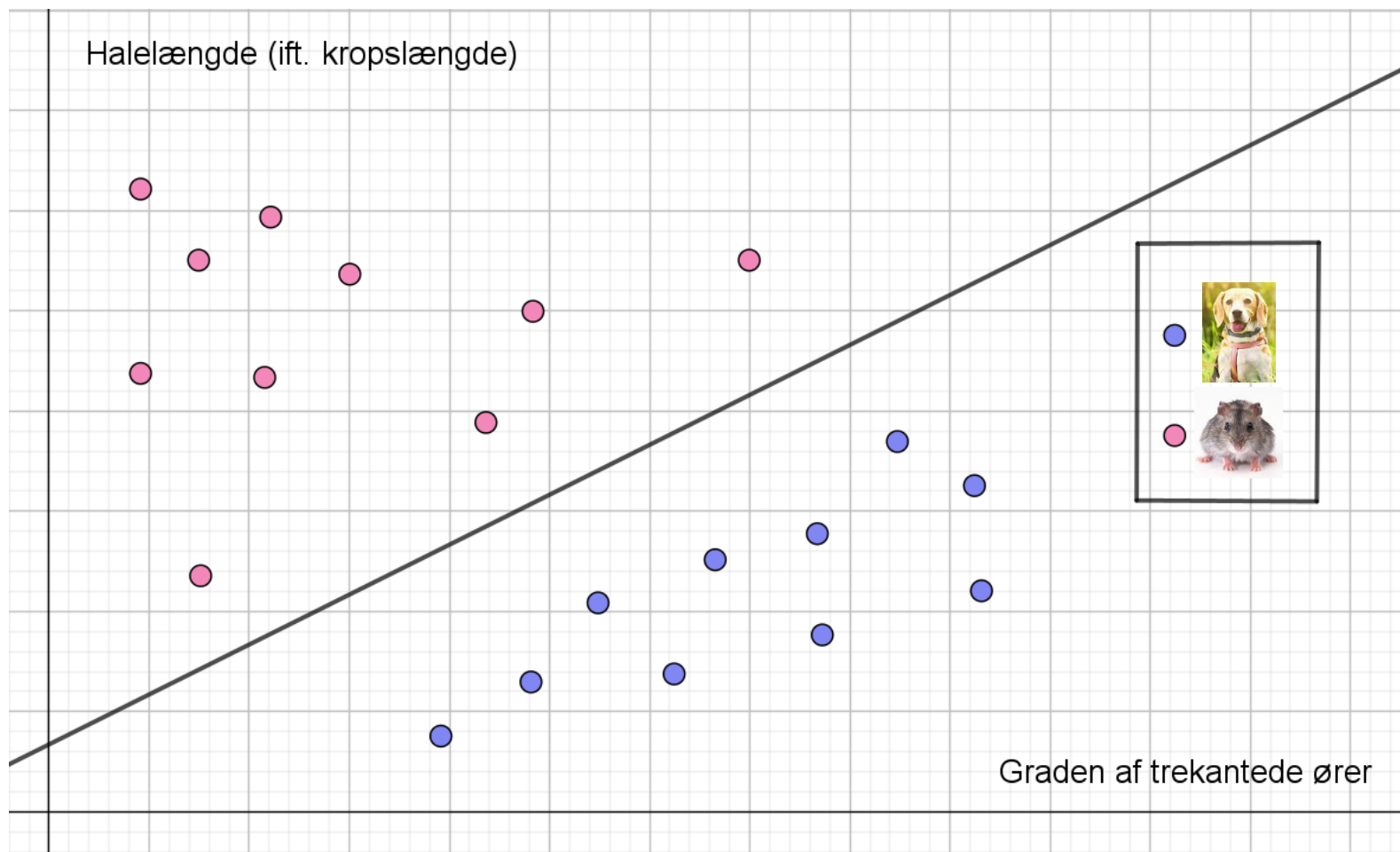


Perceptroner – hunde versus mus



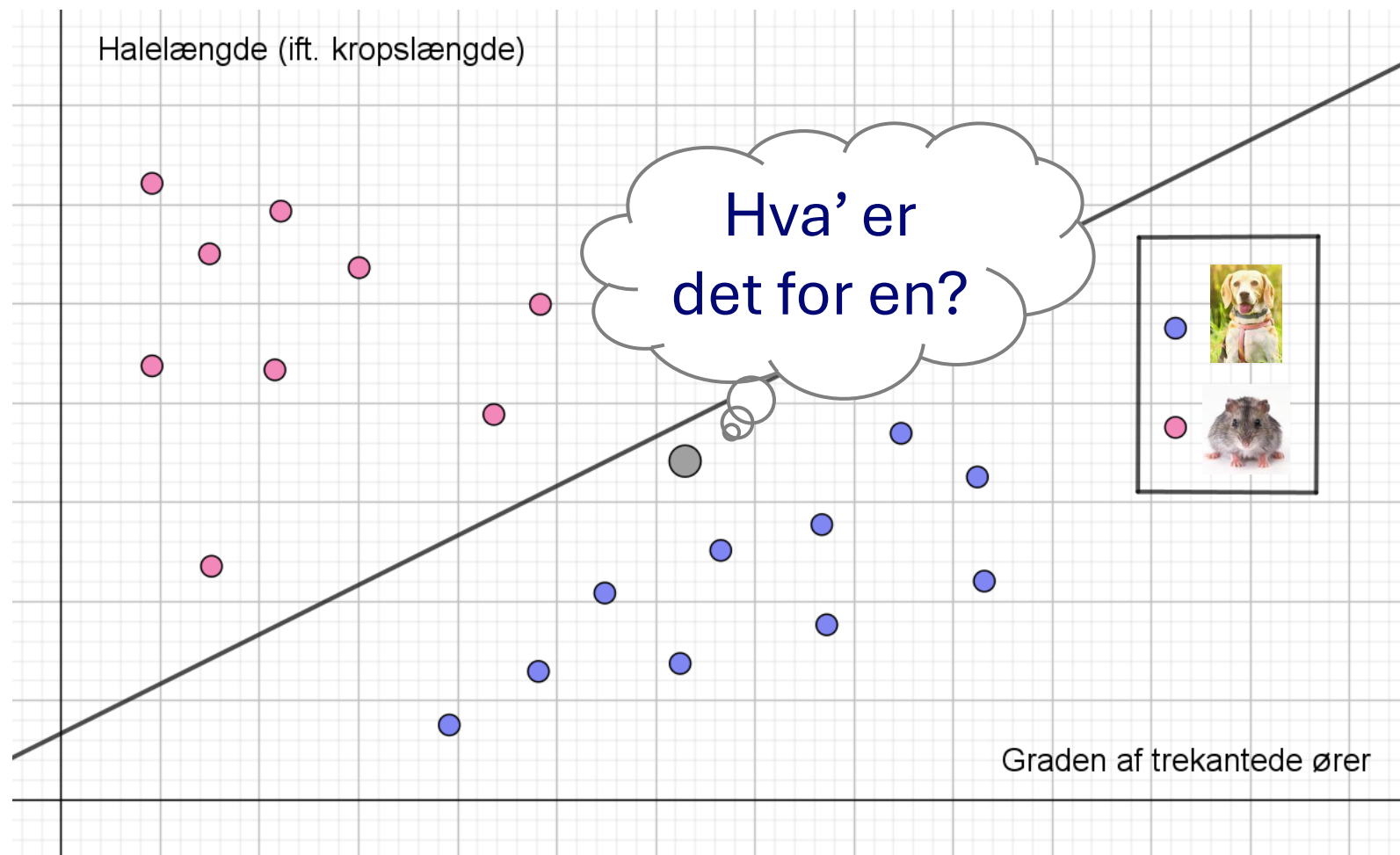


Perceptroner – hunde versus mus





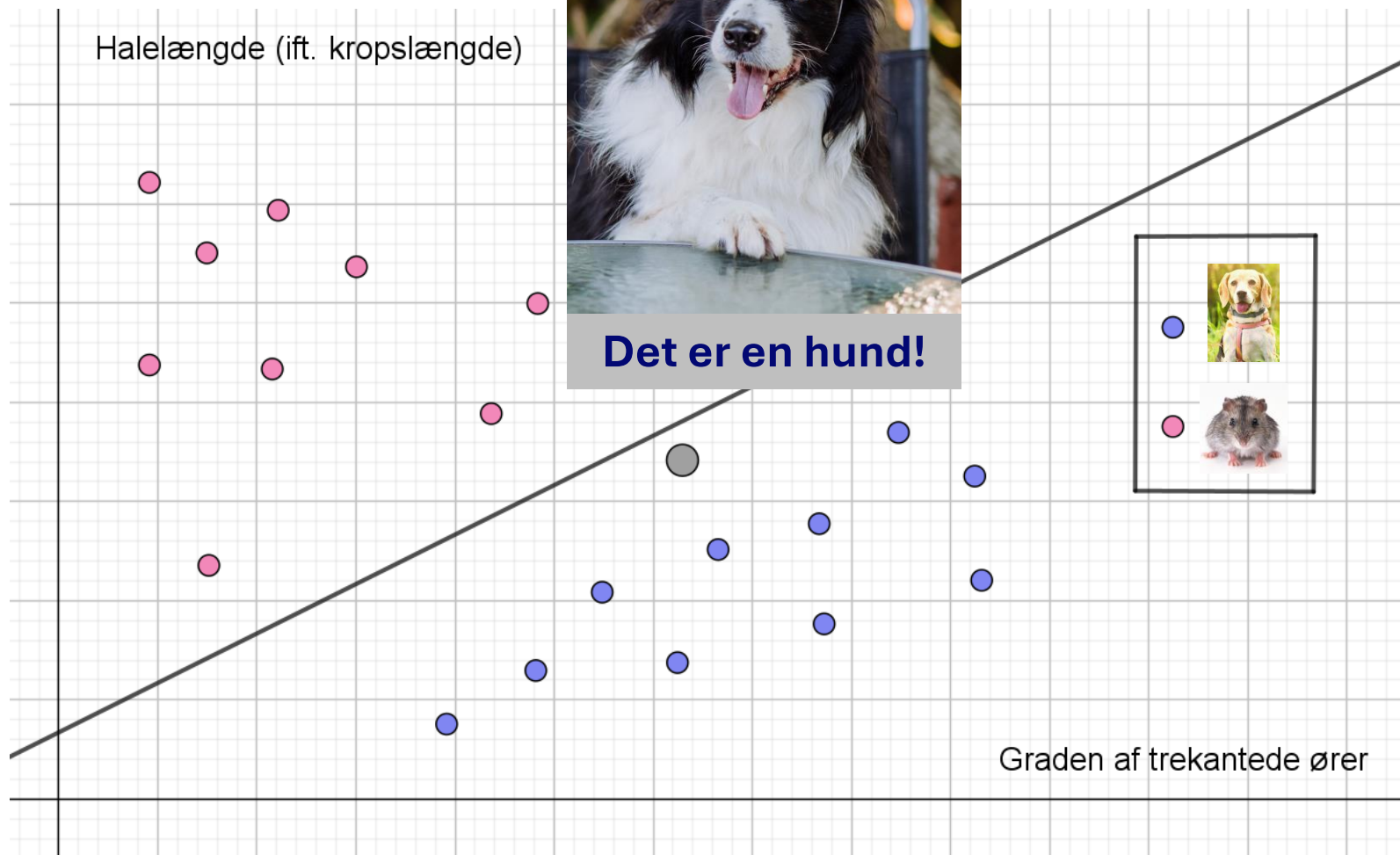
Perceptroner – hunde versus mus





Perceptroner -

versus mus





Matematikken bag (i meget grove træk)



MATEMATIKKEN BAG MAGIEN

Features / inputvariable

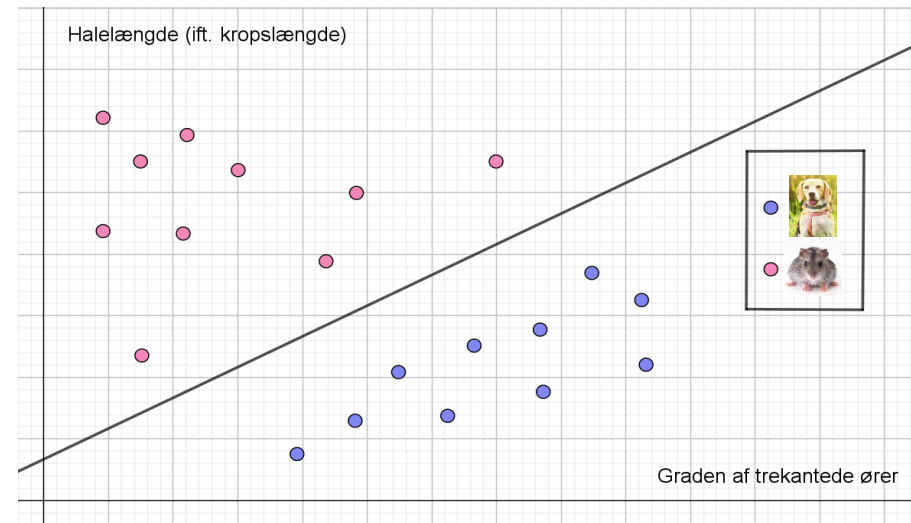
x_1 : feature 1

\vdots

x_n : feature n

Target

$$t = \begin{cases} -1 & \text{hvis det er en hund} \\ 1 & \text{hvis det er en mus} \end{cases}$$



Perceptronen bruges til at bestemme vægte $w_0, w_1, w_2, \dots, w_n$ baseret på træningsdata.

Ligningen

$$w_0 + w_1 \cdot x_1 + w_2 \cdot x_2 + \dots + w_n \cdot x_n = 0$$

beskriver en linje (plan, hyperplan), som bruges til at adskille de blå punkter fra de pinke.

Efterfølgende laves prædiktions baseret på de n features x_1, x_2, \dots, x_n :

- Hvis $w_0 + w_1 \cdot x_1 + w_2 \cdot x_2 + \dots + w_n \cdot x_n < 0$: det er en hund
- Hvis $w_0 + w_1 \cdot x_1 + w_2 \cdot x_2 + \dots + w_n \cdot x_n \geq 0$: det er en mus



Hvordan bestemmes vægtene?

Perceptron Learning Algoritmen

- Sæt alle vægte w_0, w_1, \dots, w_n til et tilfældigt tal
- Tag et træningseksempel (x_1, x_2, \dots, x_n) med tilhørende targetværdi t .
- Udregn outputværdien o :

$$o = \begin{cases} 1 & \text{hvis } w_0 + w_1 \cdot x_1 + \dots w_n \cdot x_n \geq 0 \\ -1 & \text{hvis } w_0 + w_1 \cdot x_1 + \dots w_n \cdot x_n < 0 \end{cases}$$

- Opdatér alle vægtene:

$$w_i \leftarrow w_i + \eta \cdot (t - o) \cdot x_i$$

Her er η et tal mellem 0 og 1, som kaldes for en *learning rate*

- Start forfra med det næste træningseksempel, indtil værdien af vægtene ikke ændrer sig.



Hvordan bestemmes vægtene?

Perceptron Learning Algoritmen

- Sæt alle vægte w_0, w_1, \dots, w_n til et tilfældigt tal
- Tag et træningseksempel (x_1, x_2, \dots, x_n) med tilhørende targetværdi t .
- Udregn output

Konvergerer kun hvis data er
lineært separable!

- Opdatér a

Her er η et tal mellem 0 og 1, som kaldes for en *learning rate*

- Start forfra med det næste træningseksempel, indtil værdien af vægtene ikke ændrer sig.



Hvordan bestemmes vægtene?

ADALINE

Træningseksempel 1: $(x_{1,1}, x_{1,2}, \dots, x_{1,n}, t_1)$

\vdots

Træningseksempel M: $(x_{M,1}, x_{M,2}, \dots, x_{M,n}, t_M)$

Tabsfunktion:

$$E(w_0, w_1, \dots, w_n) = \frac{1}{2} \sum_{m=1}^M \left(t_m - (w_0 + w_1 \cdot x_{m,1} + \dots + w_n \cdot x_{m,n}) \right)^2$$



Gradientnedstigning (hvis der kun var to vægte)

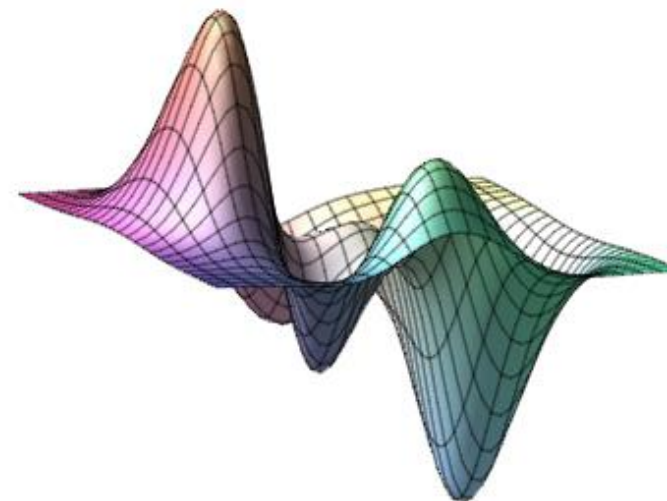
- Stil dig et tilfældigt sted på bakken.

- Gradienten:

$$\nabla E(v_0, w_0) = \begin{pmatrix} \frac{\partial E}{\partial v_0} \\ \frac{\partial E}{\partial w_0} \end{pmatrix}$$

angiver den retning, hvor det går allermest opad bakke.

- Minus gradienten $-\nabla E(v_0, w_0)$ angiver den retning, hvor det går allermest nedad bakke.





Derfor opdaterer vi alle vægtene på denne måde...

$$w \leftarrow w - \eta \cdot \frac{\partial E}{\partial w}$$

Learning rate



Opdateringsregler (ADALINE)

$$w_i \leftarrow w_i + \eta \cdot \sum_{m=1}^M \left(t_m - (w_0 + w_1 \cdot x_{m,1} + \dots + w_n \cdot x_{m,n}) \right) \cdot x_{m,i}$$

- OBS! Baseret på alle træningsdata!
- Konvergerer også i det tilfælde, hvor data ikke er lineær separabel.
- Men der er drawbacks:

t_m sammenlignes med $w_0 + w_1 \cdot x_{m,1} + \dots + w_n \cdot x_{m,n}$



Simple neurale netværk



Target

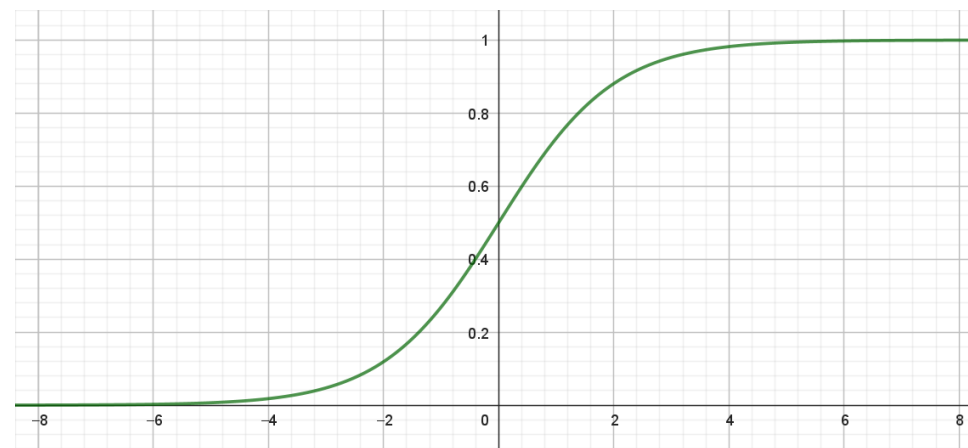
$$t = \begin{cases} 0 & \text{hvis det er en hund} \\ 1 & \text{hvis det er en mus} \end{cases}$$

Outputværdien

$$o = \sigma(w_0 + w_1 \cdot x_1 + \cdots w_n \cdot x_n)$$

hvor σ er sigmoid-funktionen:

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





Simple neurale netværk



Target

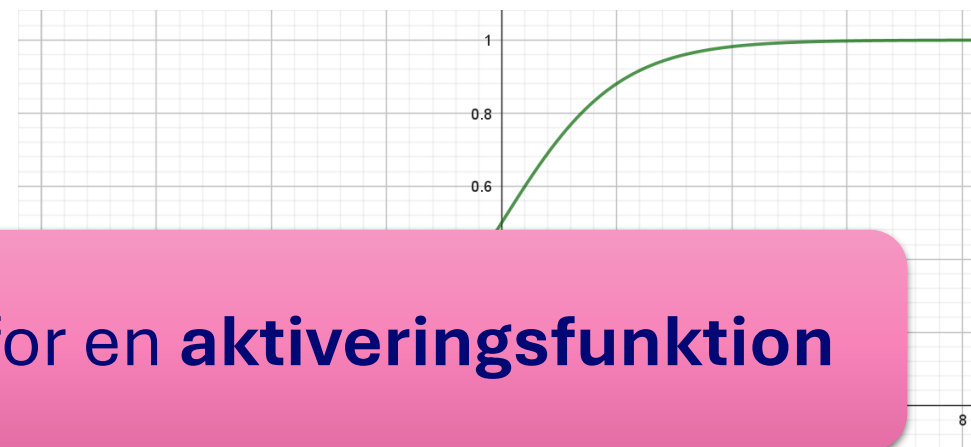
$$t = \begin{cases} 0 & \text{hvis det er en hund} \\ 1 & \text{hvis det er en mus} \end{cases}$$

Outputværdien

$$o = \sigma(w_0 + w_1 \cdot x_1 + \dots w_n \cdot x_n)$$

hvor σ er sigmoid-funktionen:

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



σ kaldes for en **aktiveringsfunktion**

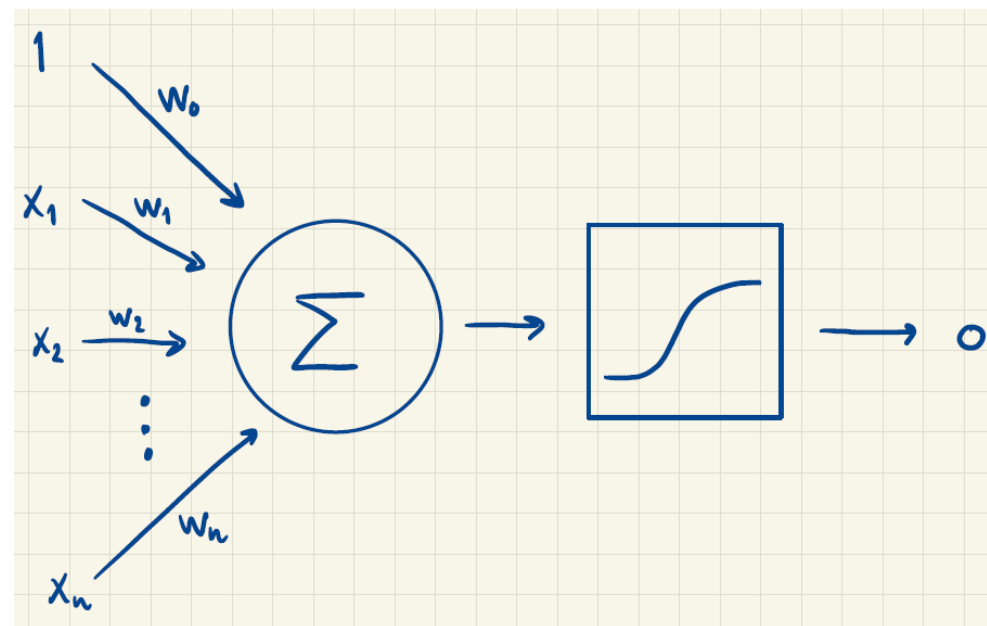


Simple neurale netværk



Tabsfunktion

$$E(w_0, w_1, \dots, w_n) = \frac{1}{2} \sum_{m=1}^M \left(t_m - \sigma(w_0 + w_1 \cdot x_{m,1} + \dots + w_n \cdot x_{m,n}) \right)^2$$





Simple neurale netværk



Tabsfunktion

$$E(w_0, w_1, \dots, w_n) = \frac{1}{2} \sum_{m=1}^M \left(t_m - \sigma(w_0 + w_1 \cdot x_{m,1} + \dots + w_n \cdot x_{m,n}) \right)^2$$

Differentiation af tabsfunktionen mht. w_i

$$\frac{\partial E}{\partial w_i} = \frac{1}{2} \sum_{m=1}^M 2 \cdot \left(t_m - \sigma(w_0 + \dots + w_n \cdot x_{m,n}) \right) \cdot \left(-\sigma'(w_0 + \dots + w_n \cdot x_{m,n}) \right) \cdot x_{m,i}$$



Simple neurale netværk

Differentiation af tabsfunktionen mht. w_i

$$\frac{\partial E}{\partial w_i} = \frac{1}{2} \sum_{m=1}^M 2 \cdot \left(t_m - \sigma(w_0 + \dots + w_n \cdot x_{m,n}) \right) \cdot \left(-\sigma'(w_0 + \dots + w_n \cdot x_{m,n}) \right) \cdot x_{m,i}$$

Man kan vise, at

$$\sigma'(x) = \sigma(x) \cdot (1 - \sigma(x))$$

- Beregningsmæssig fordel: den afledede aktiveringsfunktion kan udtrykkes ved funktionsværdien selv!



Simple neurale netværk



Differentiation af tabsfunktionen mht. w_i

$$\begin{aligned}\frac{\partial E}{\partial w_i} &= \frac{1}{2} \sum_{m=1}^M 2 \cdot \left(t_m - \sigma(w_0 + \dots + w_n \cdot x_{m,n}) \right) \cdot \left(-\sigma'(w_0 + \dots + w_n \cdot x_{m,n}) \right) \cdot x_{m,i} \\ &= - \sum_{m=1}^M \left(t_m - \sigma(w_0 + \dots + w_n \cdot x_{m,n}) \right) \cdot (\sigma(\dots) \cdot (1 - \sigma(\dots))) \cdot x_{m,i}\end{aligned}$$



Simple neurale netværk



Differentiation af tabsfunktionen mht. w_i

$$\frac{\partial E}{\partial w_i} = - \sum_{m=1}^M \left(t_m - \sigma(w_0 + \dots + w_n \cdot x_{m,n}) \right) \cdot \left(\sigma(w_0 + \dots + w_n \cdot x_{m,n}) \cdot (1 - \sigma(w_0 + \dots + w_n \cdot x_{m,n})) \right) \cdot x_{m,i}$$

Sætter vi $o_m = \sigma(w_0 + w_1 \cdot x_{m,1} + \dots + w_n \cdot x_{m,n})$, får vi

$$\frac{\partial E}{\partial w_i} = - \sum_{m=1}^M (t_m - o_m) \cdot (o_m \cdot (1 - o_m)) \cdot x_{m,i}$$



Simple neurale netværk

$$\frac{\partial E}{\partial w_i} = - \sum_{m=1}^M (t_m - o_m) \cdot (o_m \cdot (1 - o_m)) \cdot x_{m,i}$$

Gradientnedstigning giver derfor

$$w_i \leftarrow w_i + \eta \cdot \sum_{m=1}^M (t_m - o_m) \cdot o_m \cdot (1 - o_m) \cdot x_{m,i}$$

hvor

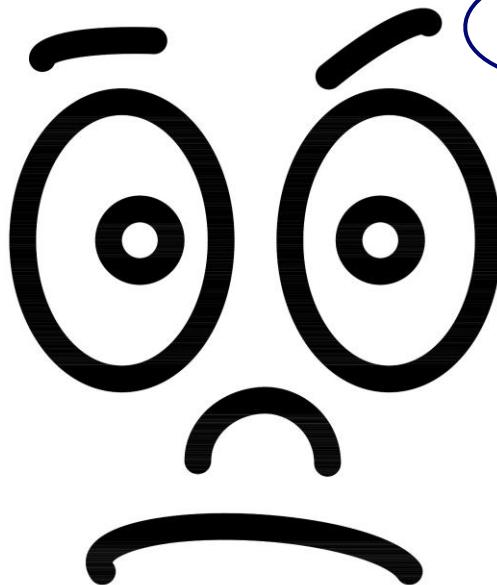
$$o_m = \sigma(w_0 + w_1 \cdot x_{m,1} + \dots + w_n \cdot x_{m,n})$$



Men vi er ikke ret gode til at udvælge features...

Halelængde (ift.
kropslængde)...?

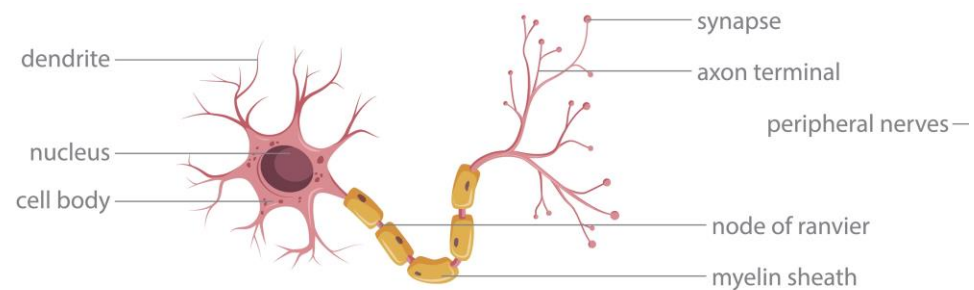
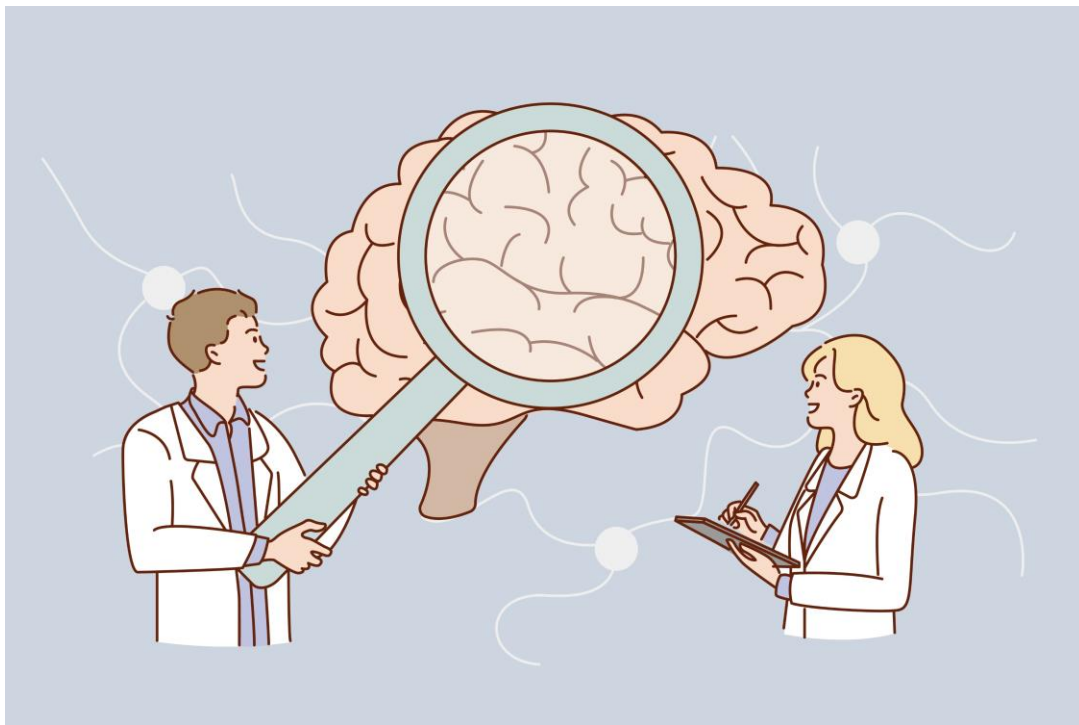
Graden af
trekantede ører...?





Kunstige neurale netværk

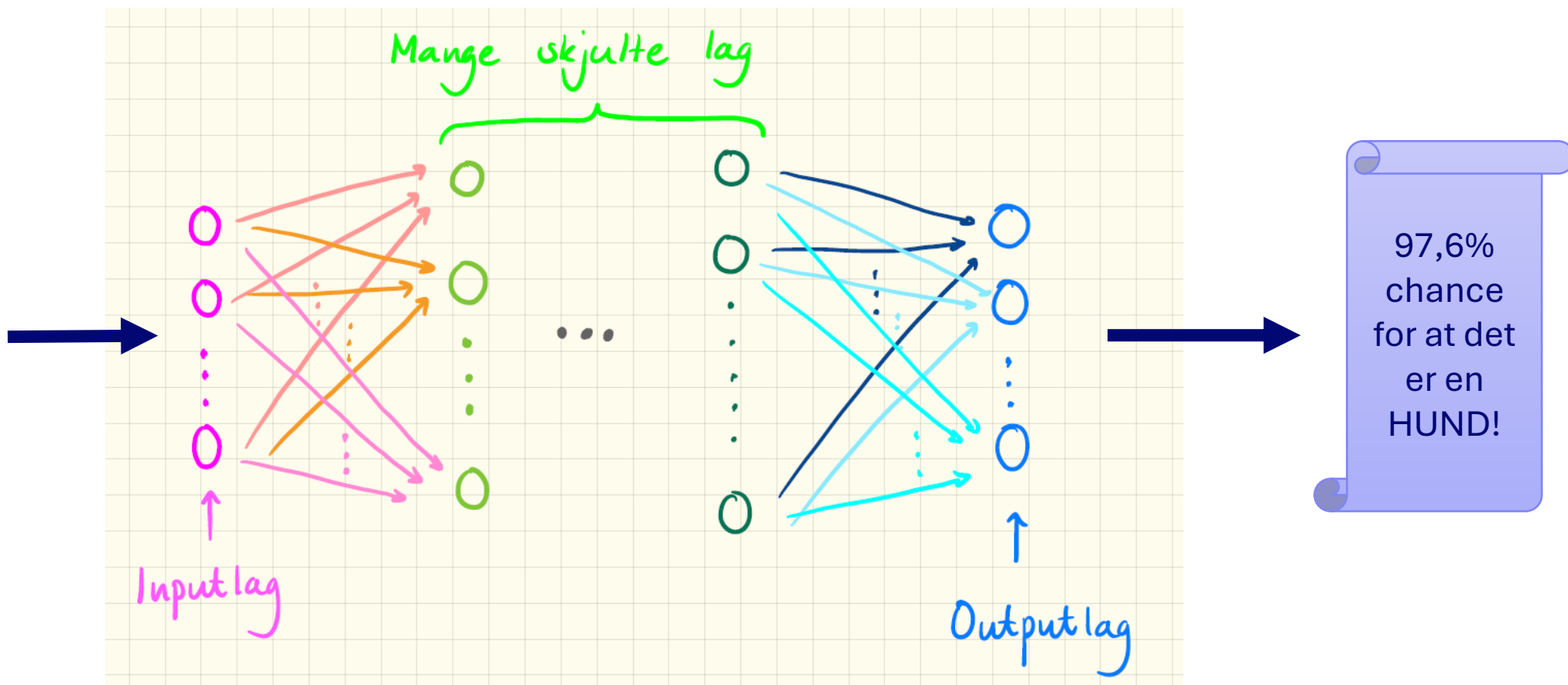
- Vi laver en model af den menneskelige hjerne





Kunstige neurale netværk

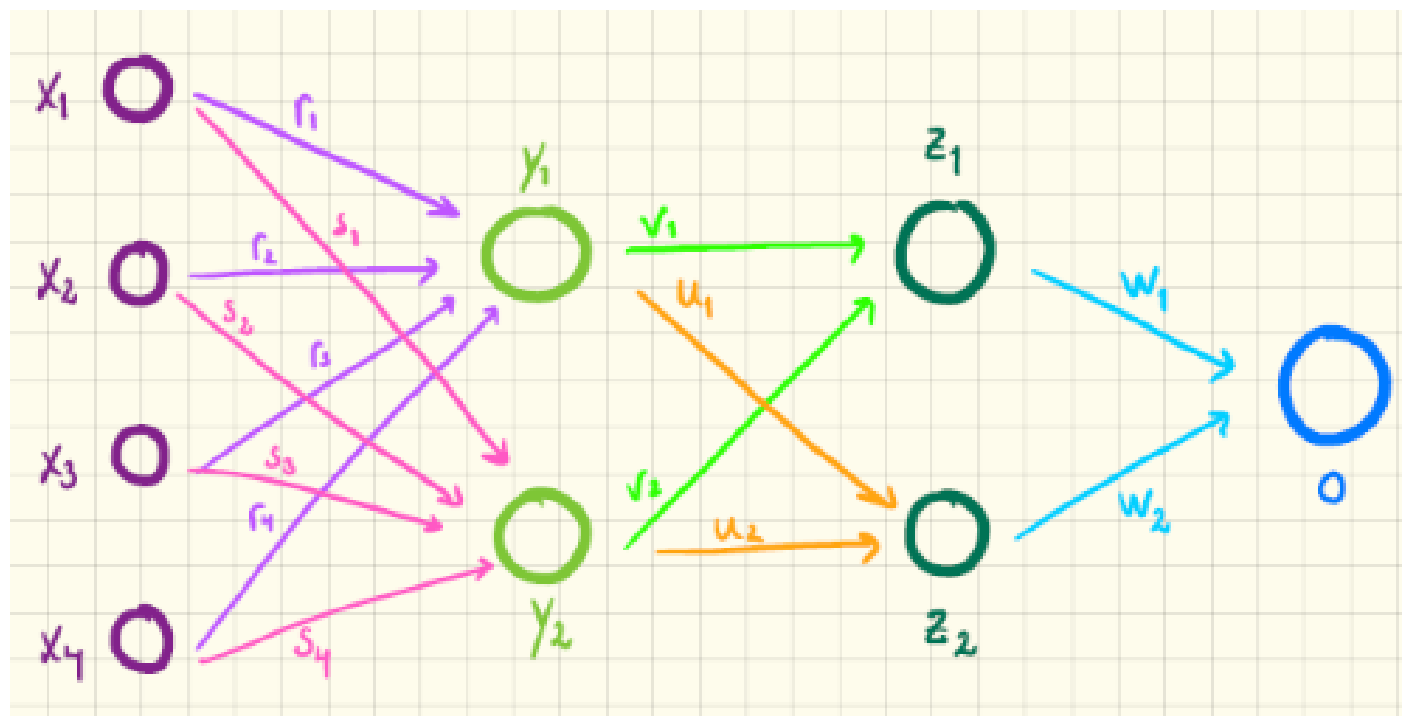
- Vi laver en model af den menneskelige hjerne





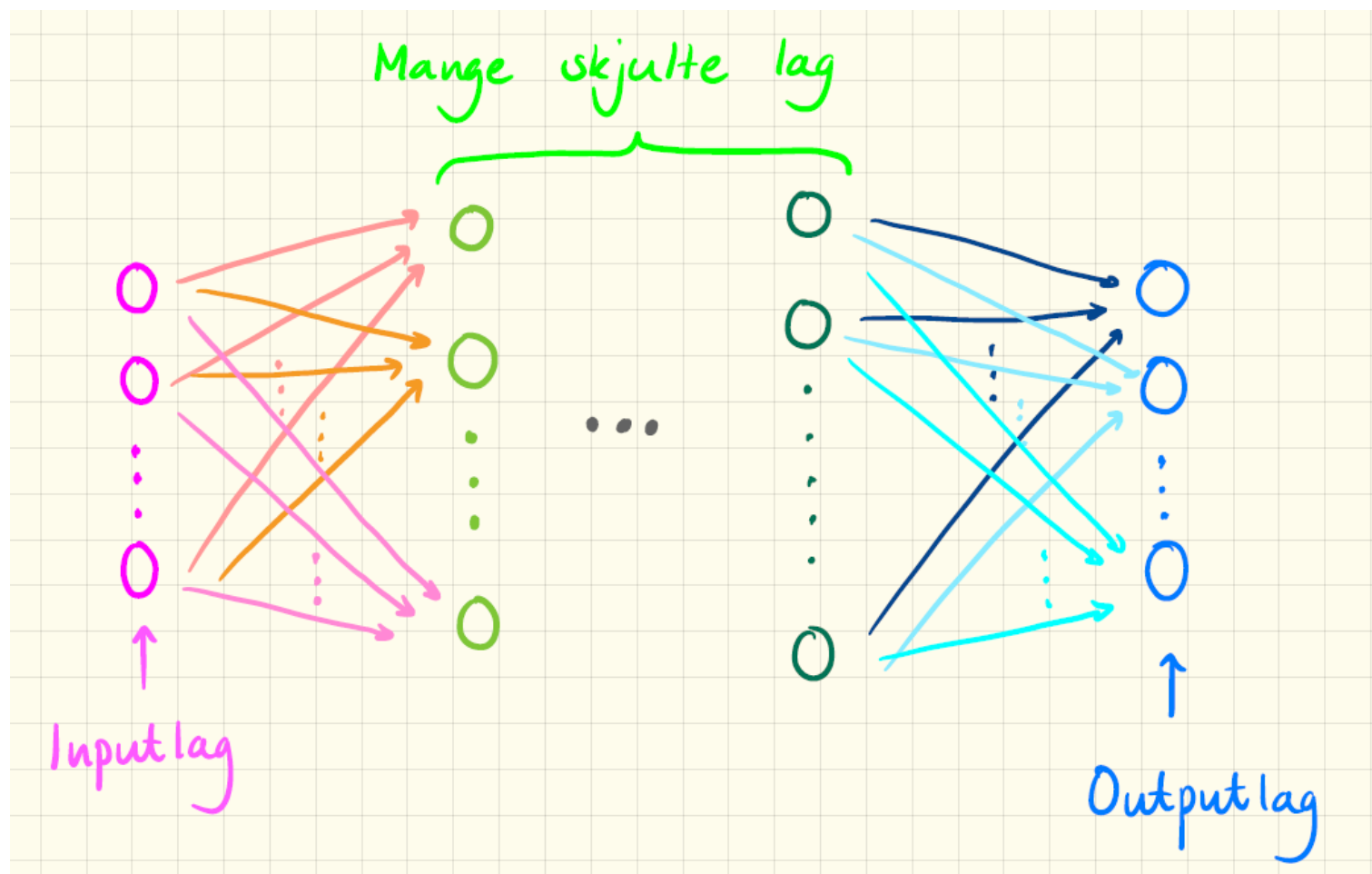
Kunstig neurale netværk

- I virkeligheden bare en sammensætning af mange simple neurale netværk 😊





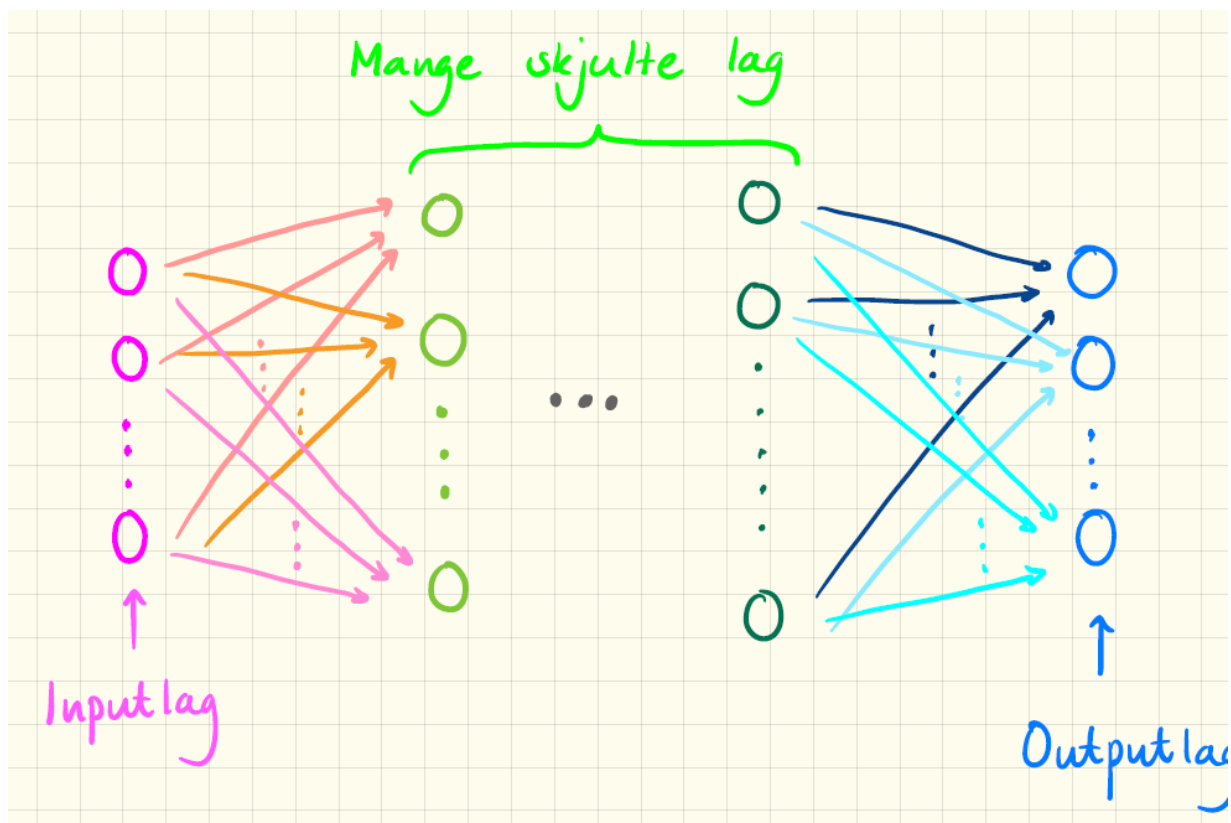
Kunstige neurale netværk



- Alle pilene svarer til en vægt som skal læres/justeres.
- I virkelighedens verden er der tale om *millioner* af vægte!

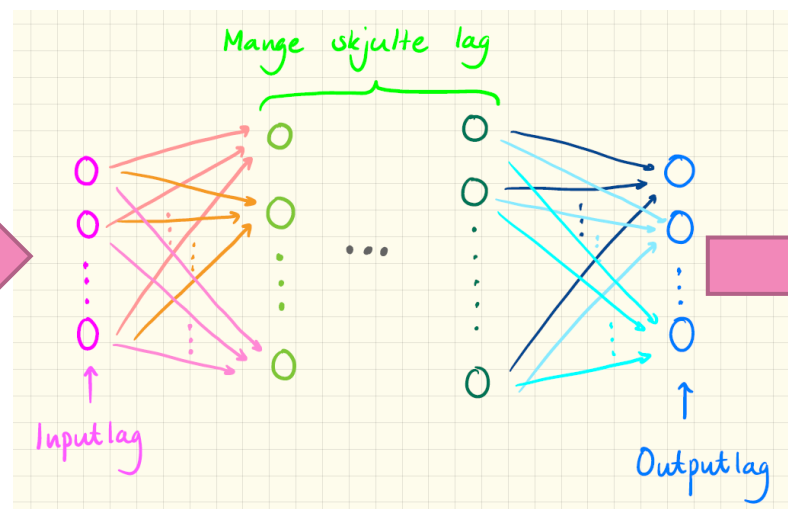
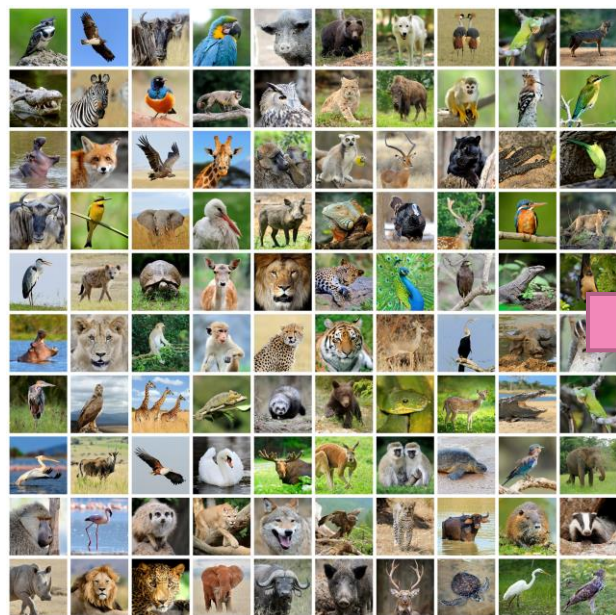


Kunstige neurale netværk – justering af vægte?!





Kunstige neurale netværk

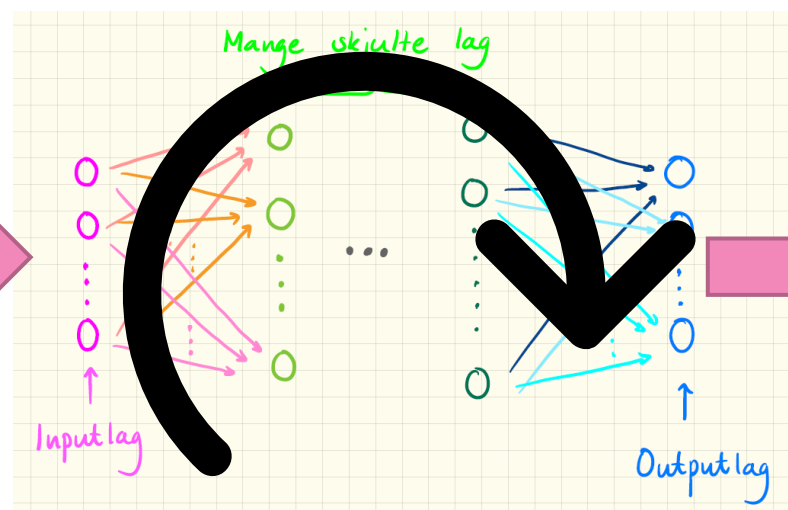
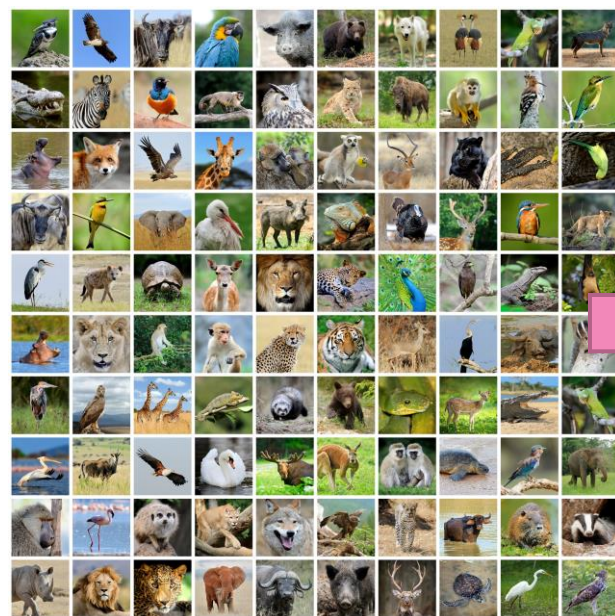


Hvor godt er netværket til at genkende billederne?

Juster vægtene så netværket er lidt bedre næste gang!



Kunstige neurale netværk

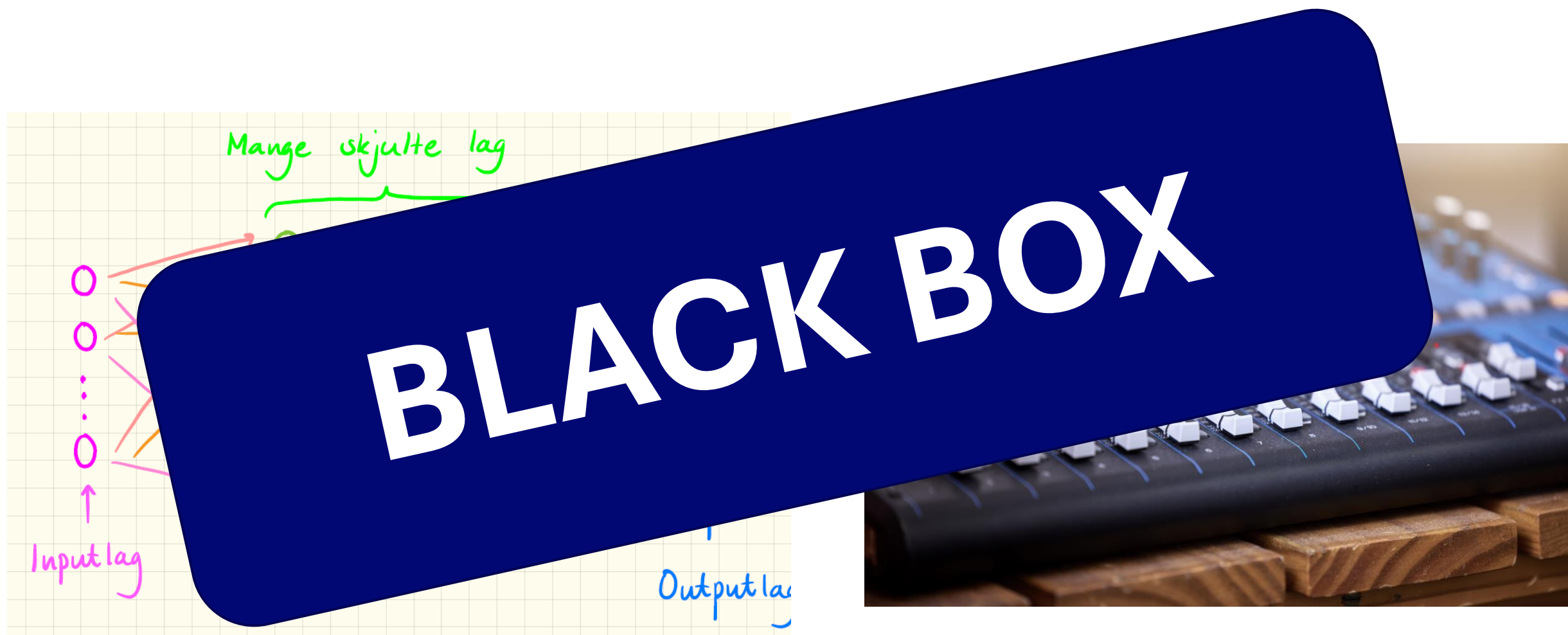


Hvor godt er netværket til at genkende billederne?

Juster vægtene så netværket er lidt bedre næste gang!



Kan du lige forklare mig den der AI model?





Hvordan vurderer man, hvor godt netværket er?

- 1) Hvis man har ekstra data (TESTDATA) kan man prøve at lade netværket prædiktere på det datasæt



Hvordan vurderer man hvor godt netværket er?

- 1) Hvis man har ekstra data (TESTDATA) kan man prøve at lade netværket prædiktere på det

Confusion matrix:

	Prædikteret hund	Prædikteret ikke hund
Faktisk hund	41	4
Faktisk ikke hund	2	53




Hvordan vurderer man, hvor godt netværket er?

- 1) Hvis man har ekstra data (TESTDATA) netværket prædiktere på det.

Classification accuracy
(CA) på 94%

Confusion matrix:



	Prædikeret hund	Prædikeret ikke hund
Faktisk hund	41	4
Faktisk ikke hund	2	53



Hvordan vurderer man, hvor godt netværket er?

2) Hvis man ikke har testdata

KRYDSVALIDERING



Fem folds krydsvalidering



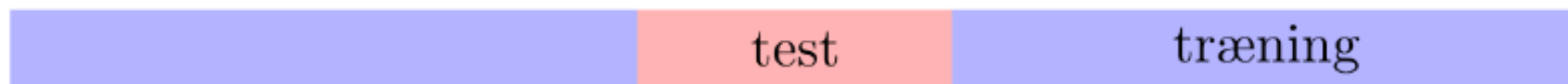
Fold 1



Fold 2



Fold 3



Fold 4

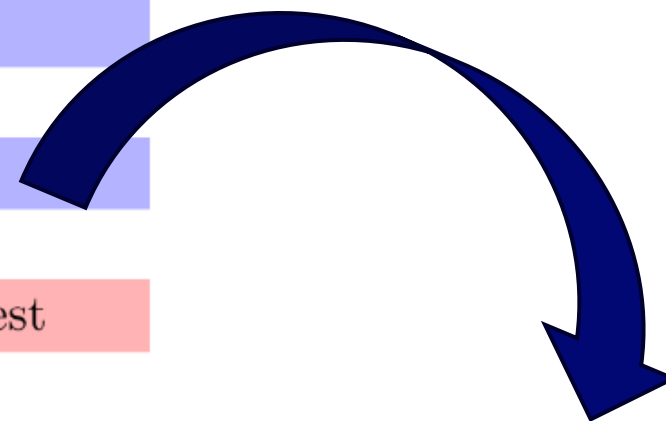
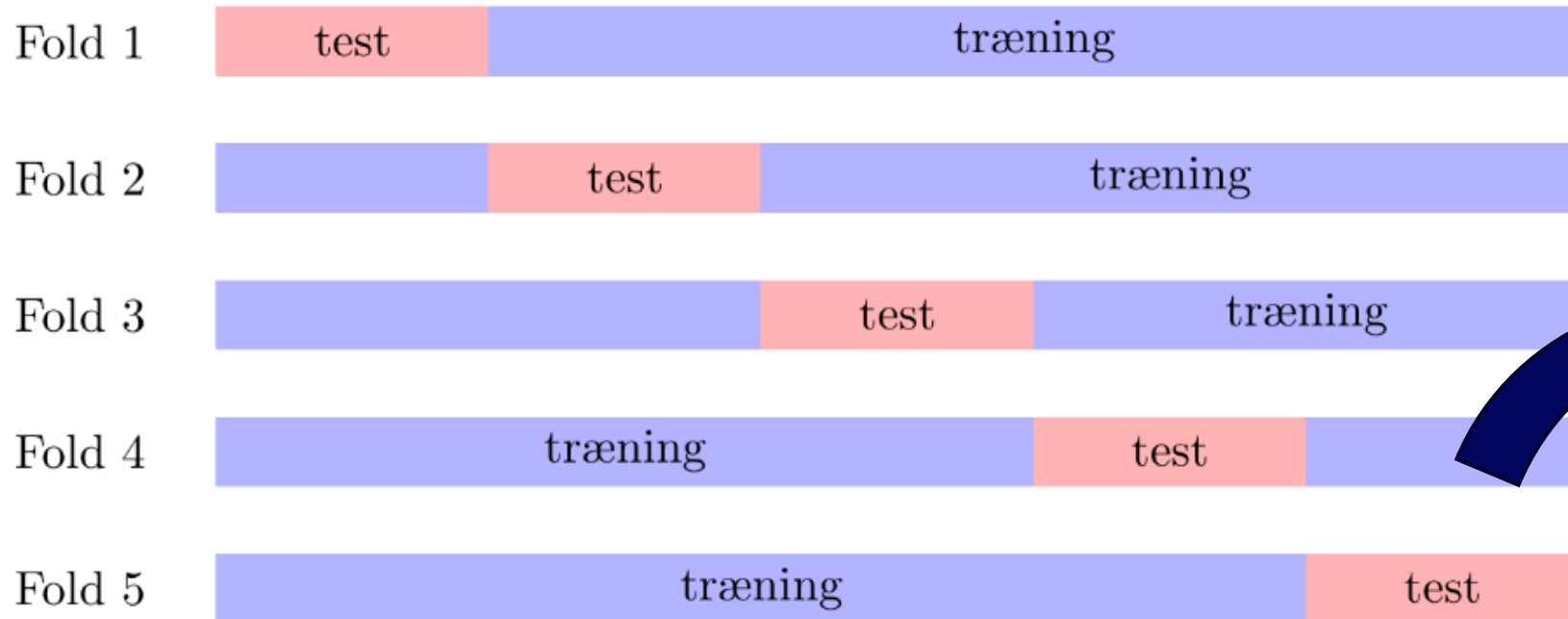


Fold 5





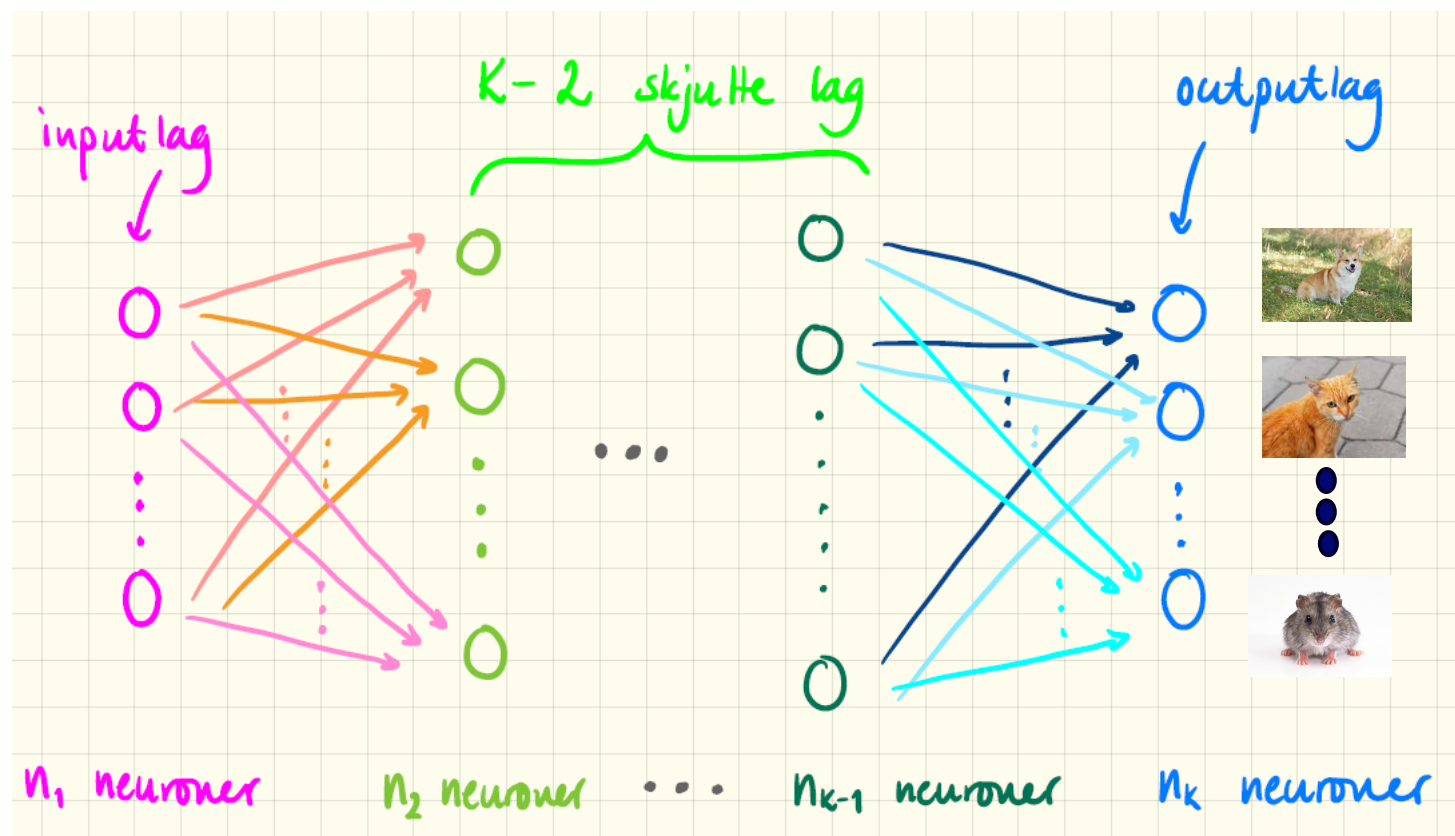
Fem folds krydsvalidering



Confusion matrix

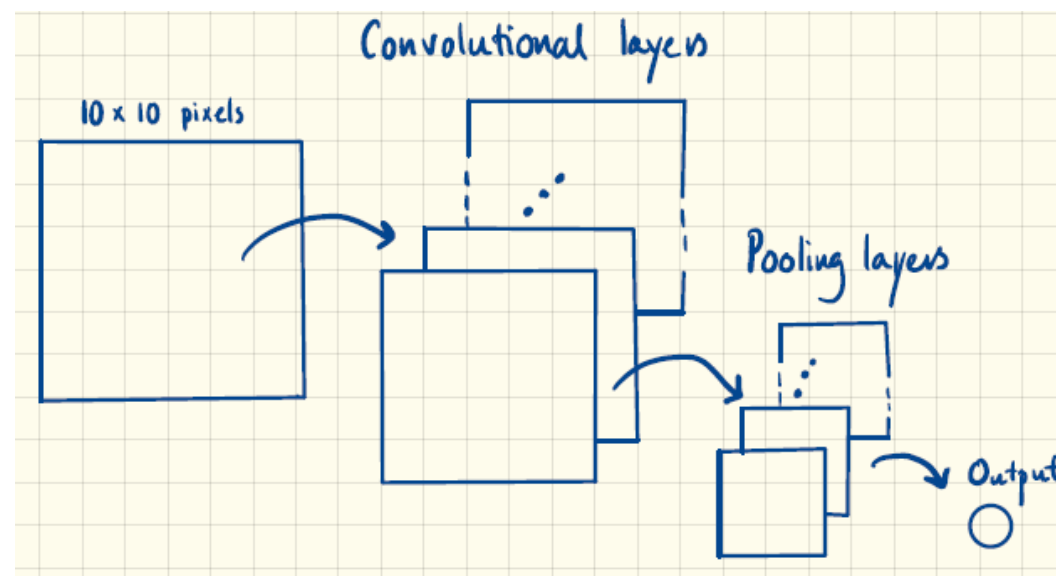
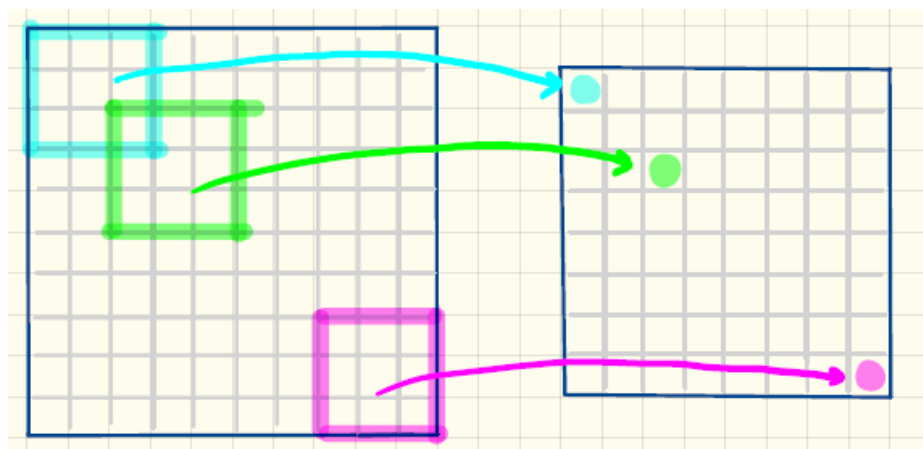


Neuralt netværk med flere outputværdier





Specielle netværk som kan genkende billeder...



Convolutional neural network





aimat.dk



MAT

MATEMATIKKEN BAG MAGIEN

 Undervisningsforløb Materialer SRO SRP Apps Referencer Om os 

Hvad er AI-matematik? Det er al den spændende matematik, som ligger bag kunstig intelligens!

På denne side kan du lære, hvordan gymnasimatematikken bruges i en masse former for kunstig intelligens. Til gengæld kan du *ikke* lære, hvordan du får en kunstig intelligens til at løse dine matematikopgaver!

Undervisningsforløb

Forskellige undervisningsforløb til matematik i gymnasiet, som inddrager AI. Der findes forløb til både A-, B- og C-niveau.

Materialer

Noter om diverse AI relaterede emner.

SRO

Idéer til hvordan AI kan inddrages i SRO.

SRP

Idéer til hvordan AI kan inddrages i SRP.