

Gymnasielærerdag 2025

AI-projektet ved Ege Rubak og Malene Cramer Engebjerg



Program

Del 1 (Malene)

- Oplæg: Kunstige neuroner – herunder aktiveringsfunktioner, tabsfunktioner, gradientnedstigning og slow learning
- Hands-on: Langt forløb om kunstige neuroner (herunder eksamensspørgsmål til mundtlig eksamen)

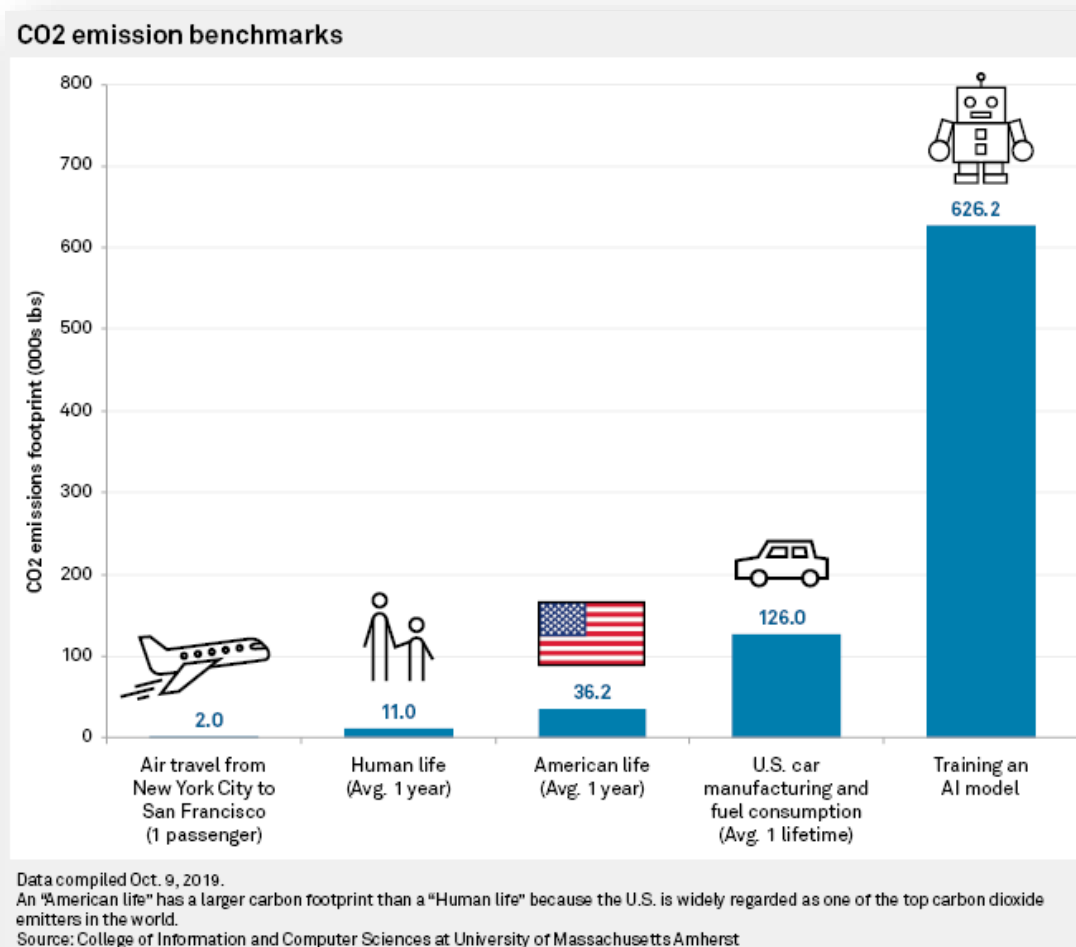
Del 2 (Ege)

- Oplæg: Store sprogmodeller – herunder Word2vec algoritmen
- Hands-on: Forløb om Word2vec

Evaluering



Grøn dannelse og AI



Kilde: <https://libguides.ecu.edu/c.php?g=1395131&p=10318505>



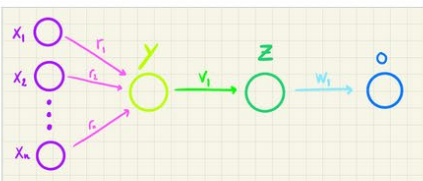
Materialer om kunstige neurale netværk

 Undervisningsforløb Materialer SRP Apps

Om os  

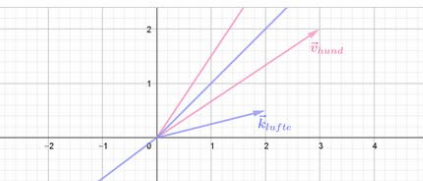
Materialer

Vi har inddelt vores AI materialer i følgende fire grupper:



Materialer om kunstige neurale netværk

Forskellige noter – med forskelligt fokus og sværhedsgrad – om kunstige neurale netværk.



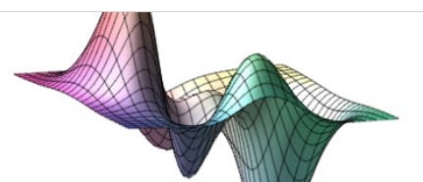
Materialer om sprogmodeller

Forskellige noter – med forskelligt fokus og sværhedsgrad – om sprogmodeller.

		Sand klasse	
		Rød	Blå
rædikeret klasse	Rød	Sand positiv	Falsk positiv
	Blå	Falsk negativ	Sand negativ

Øvrige AI materialer

Forskellige noter – med forskelligt fokus og sværhedsgrad – om forskellige andre AI materialer (end neurale net og sprogmodeller).



Baggrundsmateriale

Forskellige noter om noget af det baggrundsstof, som de forskellige AI metoder bygger på.



Materiale om kunstige neuroner

AI MAT Undervisningsforløb Materialer SRP Apps Om os

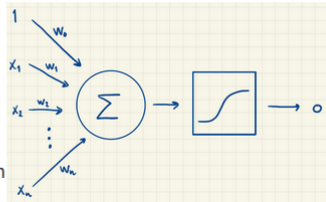
Indhold

- Materialer om kunstige neurale netværk
- Perceptroner
- Kunstige neuroner
- Simple neurale netværk
- Simple kunstige neurale netværk til multipel klassifikation
- Kunstige neurale netværk

Sværhedsgrad:

Kunstige neuroner

Noten behandler byggestenene til de mere generelle kunstige neurale netværk nemlig de *kunstige neuroner*. De kunstige neuroner bruges – som de beskrives her – til binær klassifikation. Opdateringsreglerne til justering af vægtene udledes ved at minimere *squared error* tabsfunktionen ved hjælp af gradientnedstigning. Til forskel fra ADALINE algoritmen anvendes her **sigmoid**-funktionen som aktiveringsfunktion, og der er også henvisning til en side, som beskriver hvilken fordel det har sammenlignet med ADALINE.

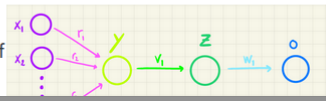


Type af klassifikation	Aktiveringsfunktion	Tabsfunktion
Binær	Sigmoid	Squared error

Sværhedsgrad: **

Simple neurale netværk

Noten behandler et kunstigt neuralt netværk, dog med den væsentlige forsimpning, at der kun er to skjulte lag, hvor hvert af disse lag kun består af én kunstig neuron. Det har den fordel,

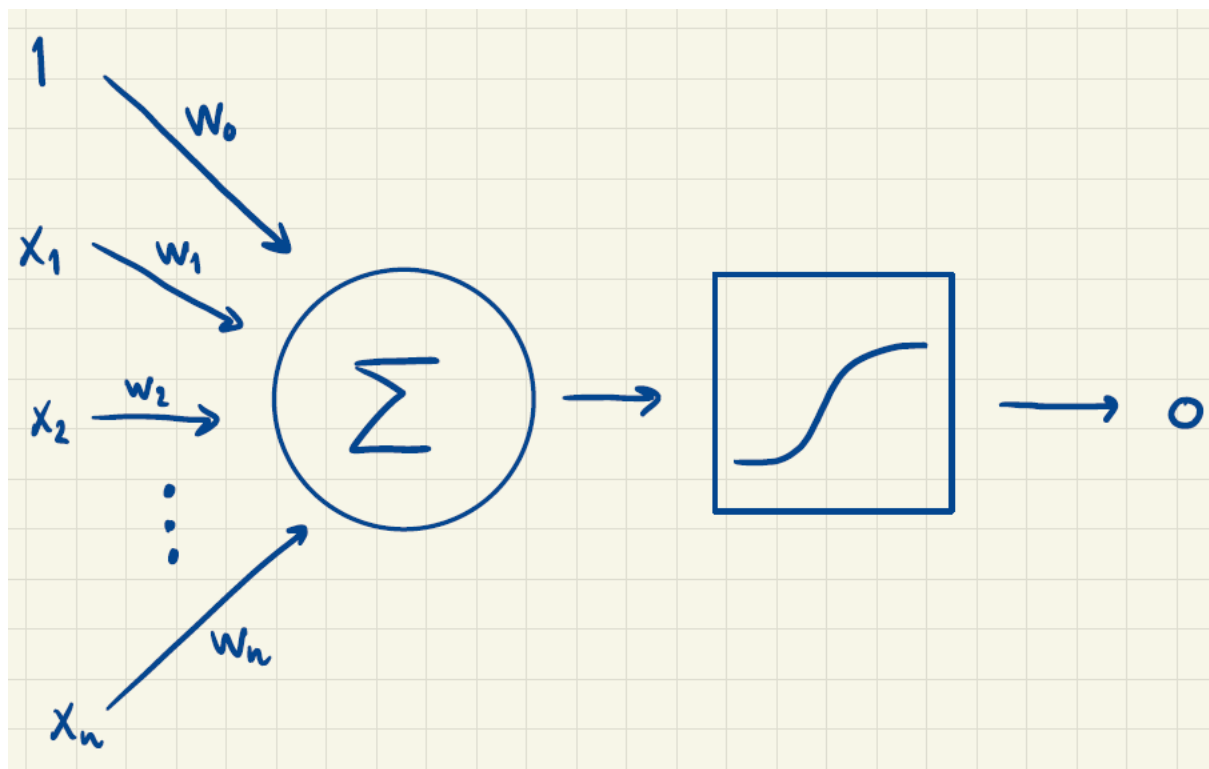




Kunstige neuroner



- De byggesten, som et kunstigt neuralt netværk består af





Eksempel: Medlemsapp til Good Food

- Medlemsapp til kunderne i Good Food, som skal vise et begrænset udvalg af tilbud, som kunden skal aktivere for at få rabat.

Idé: En midaldrende mand med et stort forbrug, skal have vist et andet tilbud end en teenagepige, som kun sjældent handler i Good Food.





Good Food indsamler data

- x_1 : kundens alder målt i år
- x_2 : kundens forbrug i Good Food den seneste måned målt i kr.

Samtidig har de i en periode for hver kunde registreret, om kunden har aktiveret et bestemt tilbud:

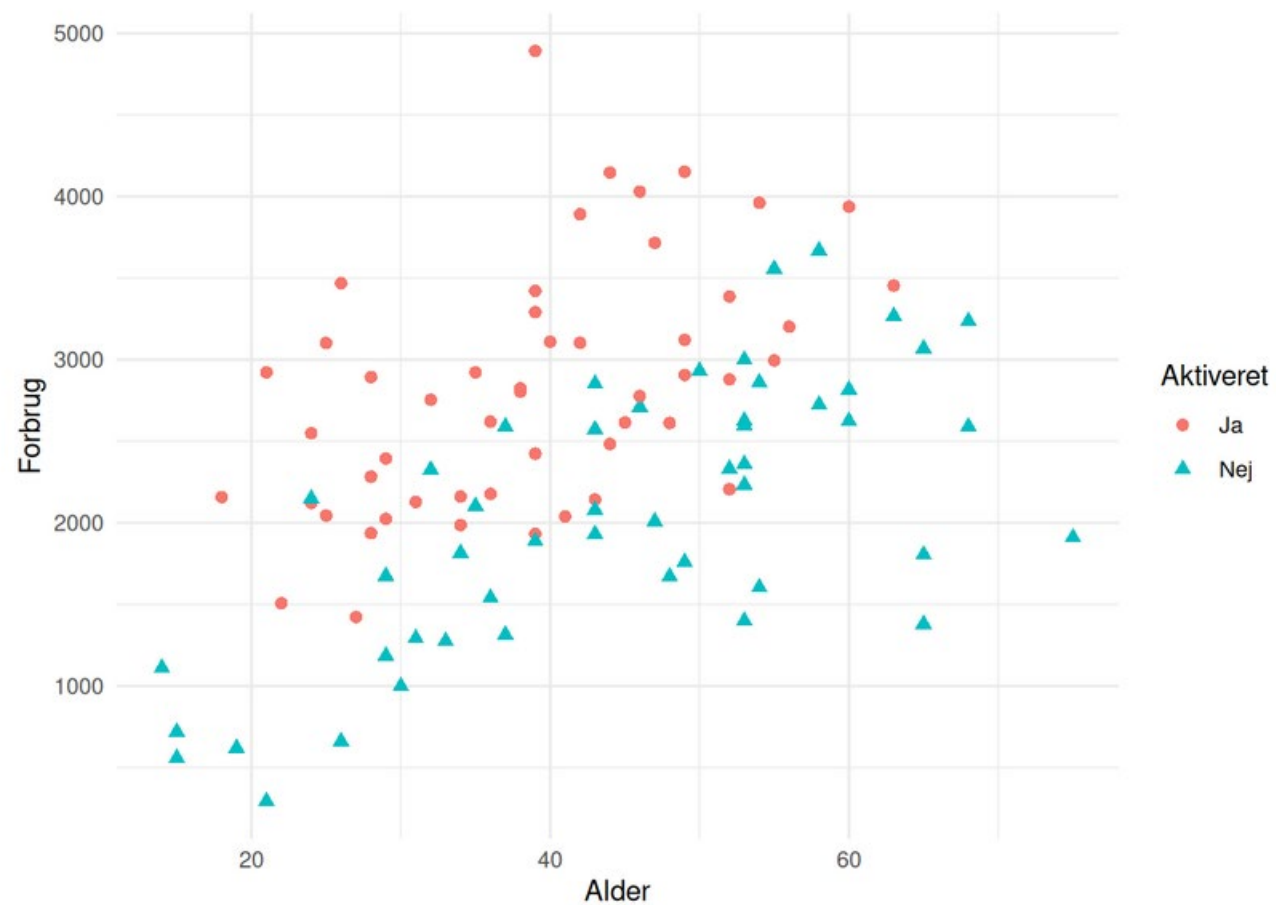
$$t = \begin{cases} 1 & \text{hvis tilbuddet aktiveres} \\ 0 & \text{hvis tilbuddet ikke aktiveres} \end{cases}$$

t kaldes for en **targetværdi**





Plot af træningsdata





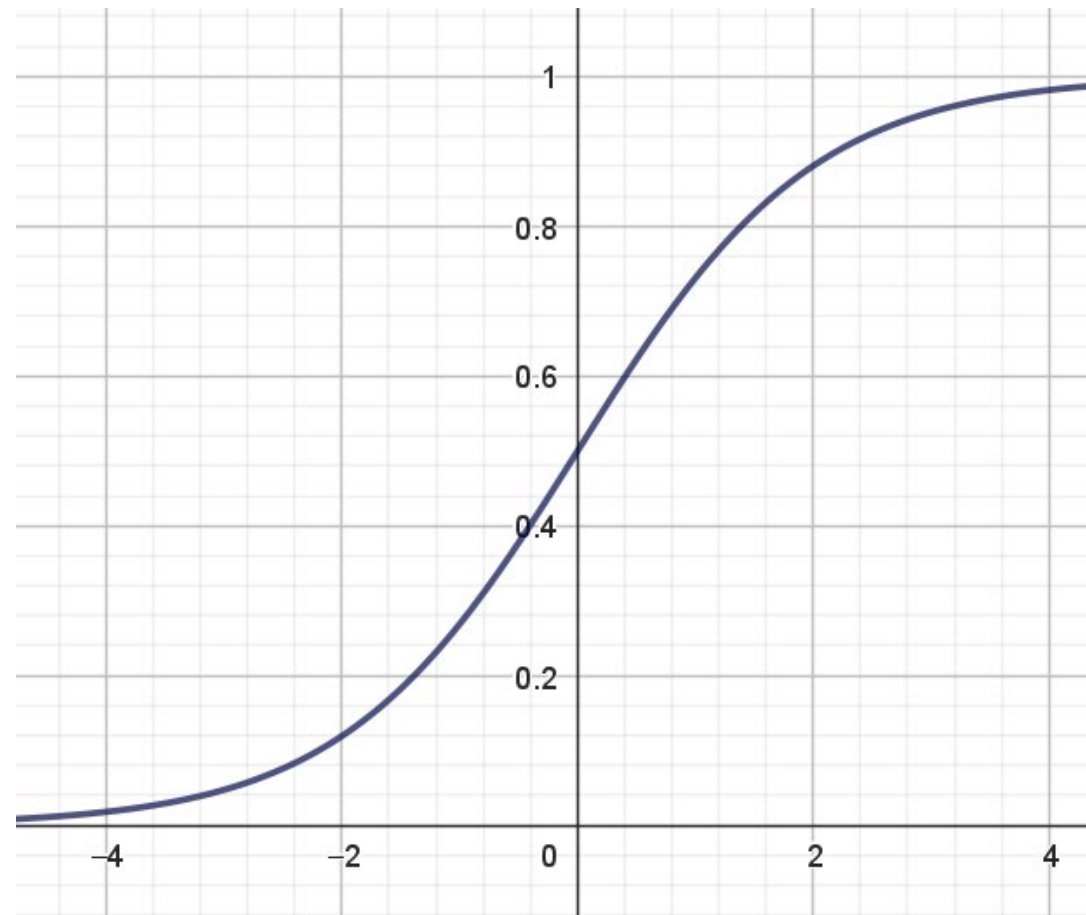
Beregning af outputværdi o

- Outputværdien o skal være et bud på sandsynligheden for at en kunde aktiverer et tilbud
- Vi beregner en outputværdi o

$$o = \sigma(w_0 + w_1 \cdot x_1 + w_2 \cdot x_2)$$

hvor σ er sigmoid-funktionen:

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





Grafisk illustration

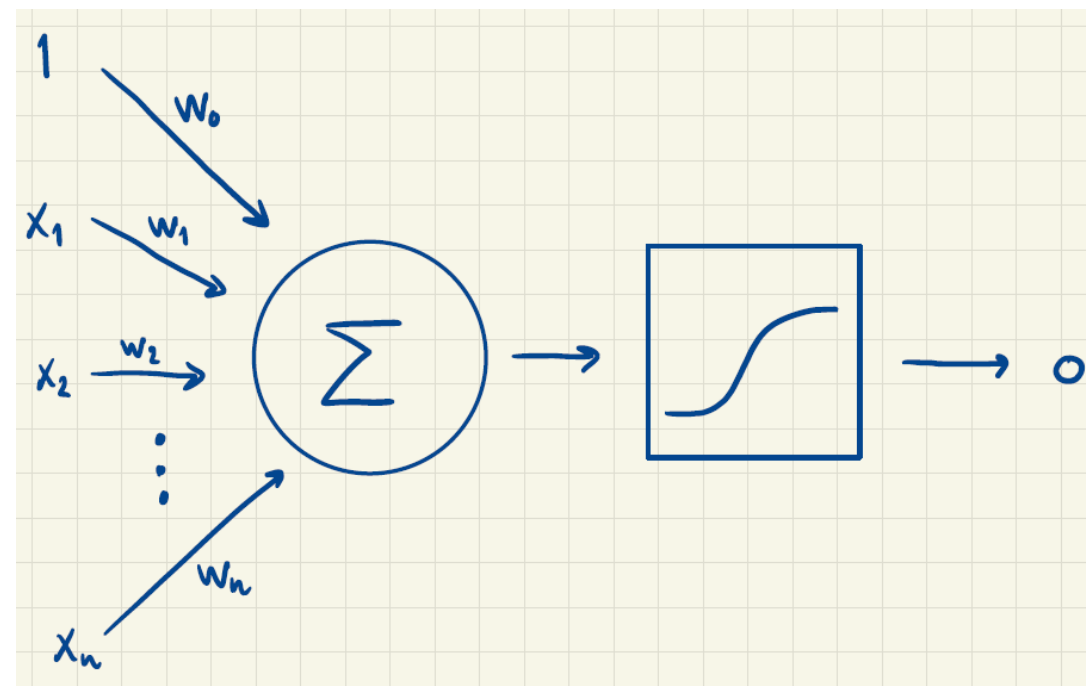


$$o = \sigma(w_0 + w_1 \cdot x_1 + w_2 \cdot x_2)$$



Aktiveringsfunktion

(og der findes andre end sigmoid)





Tabsfunktion (*squared error*)

Vi definerer *squared error* tabsfunktionen:

$$E = \frac{1}{2} \sum (t - o)^2$$

Det vil sige

$$E = \frac{1}{2} \sum (t - \sigma(w_0 + w_1 \cdot x_1 + w_2 \cdot x_2))^2$$

Vægtene w_0 , w_1 og w_2 skal bestemmes, så tabsfunktionen minimeres.



Tilbage til træningsdata



- Træningseksempel 1: $(x_1^{(1)}, x_2^{(1)}, \dots, x_n^{(1)}, t^{(1)})$
⋮
- Træningseksempel m : $(x_1^{(m)}, x_2^{(m)}, \dots, x_n^{(m)}, t^{(m)})$
⋮
- Træningseksempel M : $(x_1^{(M)}, x_2^{(M)}, \dots, x_n^{(M)}, t^{(M)})$



Tabsfunktionen generelt



$$\begin{aligned} E(w_0, w_1, \dots, w_n) \\ = \frac{1}{2} \sum_{m=1}^M \left(t^{(m)} - \sigma \left(w_0 + w_1 \cdot x_1^{(m)} + \dots + w_n \cdot x_n^{(m)} \right) \right)^2 \end{aligned}$$



Gradientnedstigning

Vægtene opdateres ved at gå et lille stykke i den negative gradients retning:

$$w_i^{(ny)} \leftarrow w_i - \eta \cdot \frac{\partial E}{\partial w_i}$$

for $i \in \{0, 1, \dots, n\}$.

η : learning rate



Partielle afledede 😊



Husk, at

$$E(w_0, w_1, \dots, w_n) = \frac{1}{2} \sum_{m=1}^M \left(t^{(m)} - \sigma \left(w_0 + w_1 \cdot x_1^{(m)} + \dots + w_n \cdot x_n^{(m)} \right) \right)^2$$

Det vil sige, at

- Vi får brug for sum- og kædereglen
- Vi får brug for at differentiere sigmoid-funktionen



At differentiere aktiveringsfunktioner er en ting!

- Man kan vise (og det er sjovt 😊), at

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

- Det er en eftertragtet egenskab ved aktiveringsfunktioner: at deres afledede kan udtrykkes simpelt ved funktionsværdien selv (det er en beregningsmæssig kæmpe fordel)



Partielle afledede 😊

$$E(w_0, w_1, \dots, w_n) = \frac{1}{2} \sum_{m=1}^M \left(t^{(m)} - \sigma \left(w_0 + w_1 \cdot x_1^{(m)} + \dots + w_n \cdot x_n^{(m)} \right) \right)^2$$

Derfor er

$$\frac{\partial E}{\partial w_i} = - \sum_{m=1}^M \left(t^{(m)} - \sigma(\dots) \right) \cdot \sigma'(\dots) \cdot x_i^{(m)}$$

$$\frac{\partial E}{\partial w_i} = - \sum_{m=1}^M \left(t^{(m)} - o^{(m)} \right) \cdot o^{(m)} \cdot (1 - o^{(m)}) \cdot x_i^{(m)}$$

hvor $o^{(m)} = \sigma \left(w_0 + w_1 \cdot x_1^{(m)} + \dots + w_n \cdot x_n^{(m)} \right)$ for $i \in \{1, \dots, n\}$



Partielle afledede 😊

$$E(w_0, w_1, \dots, w_n) = \frac{1}{2} \sum_{m=1}^M \left(t^{(m)} - \sigma \left(w_0 + w_1 \cdot x_1^{(m)} + \dots + w_n \cdot x_n^{(m)} \right) \right)^2$$

Det vil sige, at

$$\frac{\partial E}{\partial w_0} = - \sum_{m=1}^M \left(t^{(m)} - o^{(m)} \right) \cdot o^{(m)} \cdot (1 - o^{(m)})$$

og

$$\frac{\partial E}{\partial w_i} = - \sum_{m=1}^M \left(t^{(m)} - o^{(m)} \right) \cdot o^{(m)} \cdot (1 - o^{(m)}) \cdot x_i^{(m)}$$

for $i \in \{1, \dots, n\}$



Opdateringsregler for kunstige neuroner

$$w_0^{(ny)} \leftarrow w_0 + \eta \cdot \sum_{m=1}^M (t^{(m)} - o^{(m)}) \cdot o^{(m)} \cdot (1 - o^{(m)})$$

og

$$w_i^{(ny)} \leftarrow w_i + \eta \cdot \sum_{m=1}^M (t^{(m)} - o^{(m)}) \cdot o^{(m)} \cdot (1 - o^{(m)}) \cdot x_i^{(m)}$$

for $i \in \{1, \dots, n\}$.



Prædiktion



For en ny kunde kan vi lave følgende prædiktion:

$$\text{Kunden aktiverer tilbuddet: } \begin{cases} \text{Ja} & o \geq 0.5 \\ \text{Nej} & o < 0.5 \end{cases}$$

hvor

$$o = \sigma(w_0 + w_1 \cdot x_1 + w_2 \cdot x_2)$$



Skillelinje (eller plan/hyperplan)

Bemærk, at

$$\sigma(w_0 + w_1 \cdot x_1 + \dots + w_n \cdot x_n) = 0.5$$

svarer til

$$\frac{1}{1 + e^{-(w_0 + w_1 \cdot x_1 + \dots + w_n \cdot x_n)}} = 0.5$$

Det vil sige, at

$$e^{-(w_0 + w_1 \cdot x_1 + \dots + w_n \cdot x_n)} = 1$$

Altså

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



Træning af kunstig neuron

AI MAT

Undervisningsforløb Materialer SRP Apps

Om os

Indhold

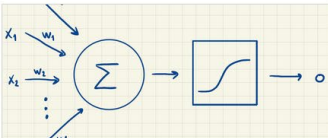
Generelle apps til træning af kunstig intelligens

Særlige apps som bruges i konkrete forløb

AI apps

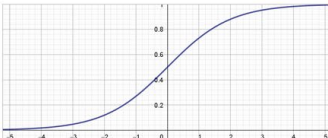
Her på siden finder du apps til beregninger og illustrationer i forbindelse med AI.

Generelle apps til træning af kunstig intelligens



Kunstig neuron app

Her kan du træne en kunstig neuron med dine egne data. Der er mulighed for at vælge forskellige aktiveringsfunktioner og forskellige tabsfunktioner.



Logistisk regression app

Her kan du lave logistisk regression med dine egne data.

Særlige apps som bruges i konkrete forløb



0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

MNIST app

App som viser MNIST datasættet (håndskrevne cifre). Bruges i forløbet "Kan vi genkende håndskrevne tal?".

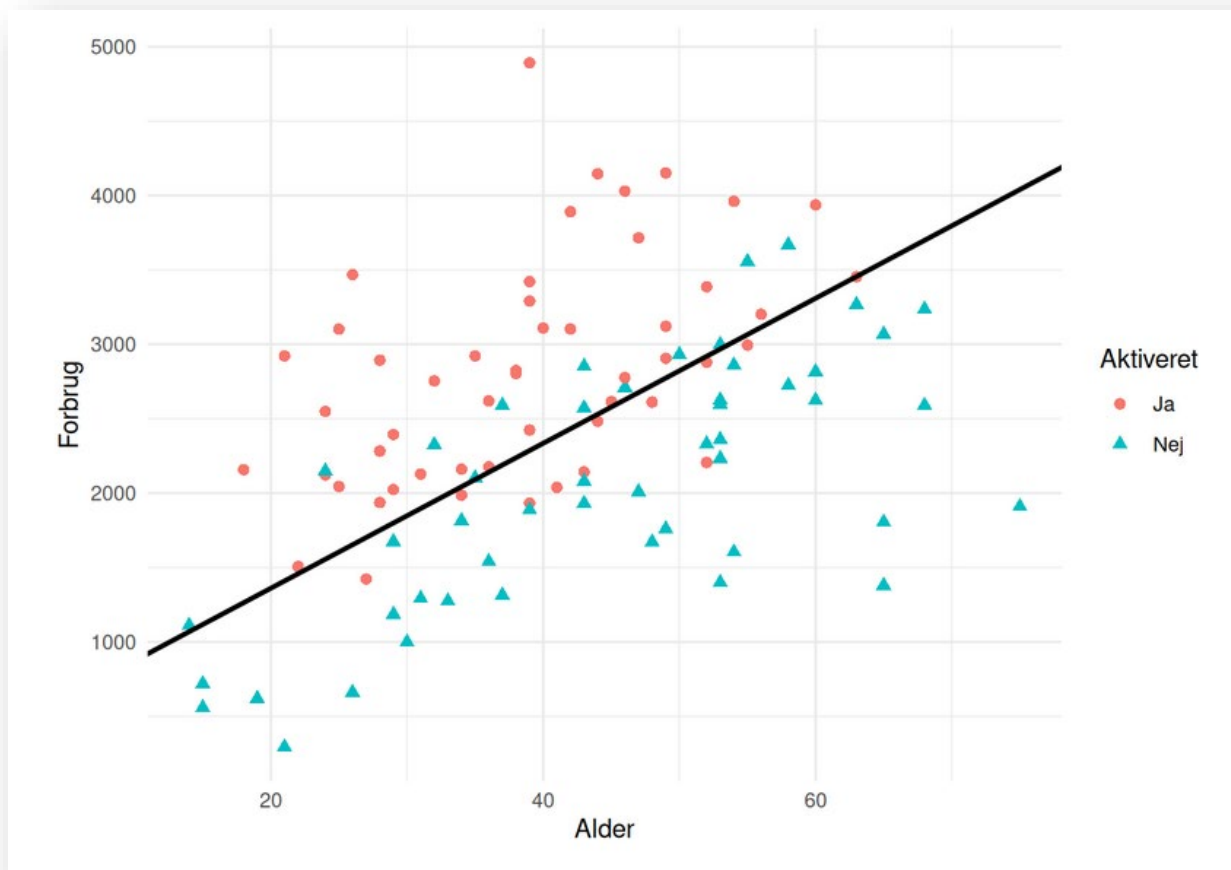


Gendata app

Denne app kan bruges til at generere normalfordelte data med en given middelværdi og spredning. Bruges i forløbet "Screeningsprogrammer".



Træning baseret på Good Food datasæt



$$w_0 = -0.298$$

$$w_1 = -0.0375$$

$$w_2 = 0.000769$$



Sig hej til Hans og Gerda



Gerda

Alder: 65 år
Forbrug: 2800 kr.



Hans

Alder: 50 år
Forbrug: 3500 kr.



Skal vi vise tilbuddet til Hans og/eller Gerda?

Hans

$$o = \sigma(-0.298 - 0.0375 \cdot 50 + 0.000769 \cdot 3500) = 0.627$$

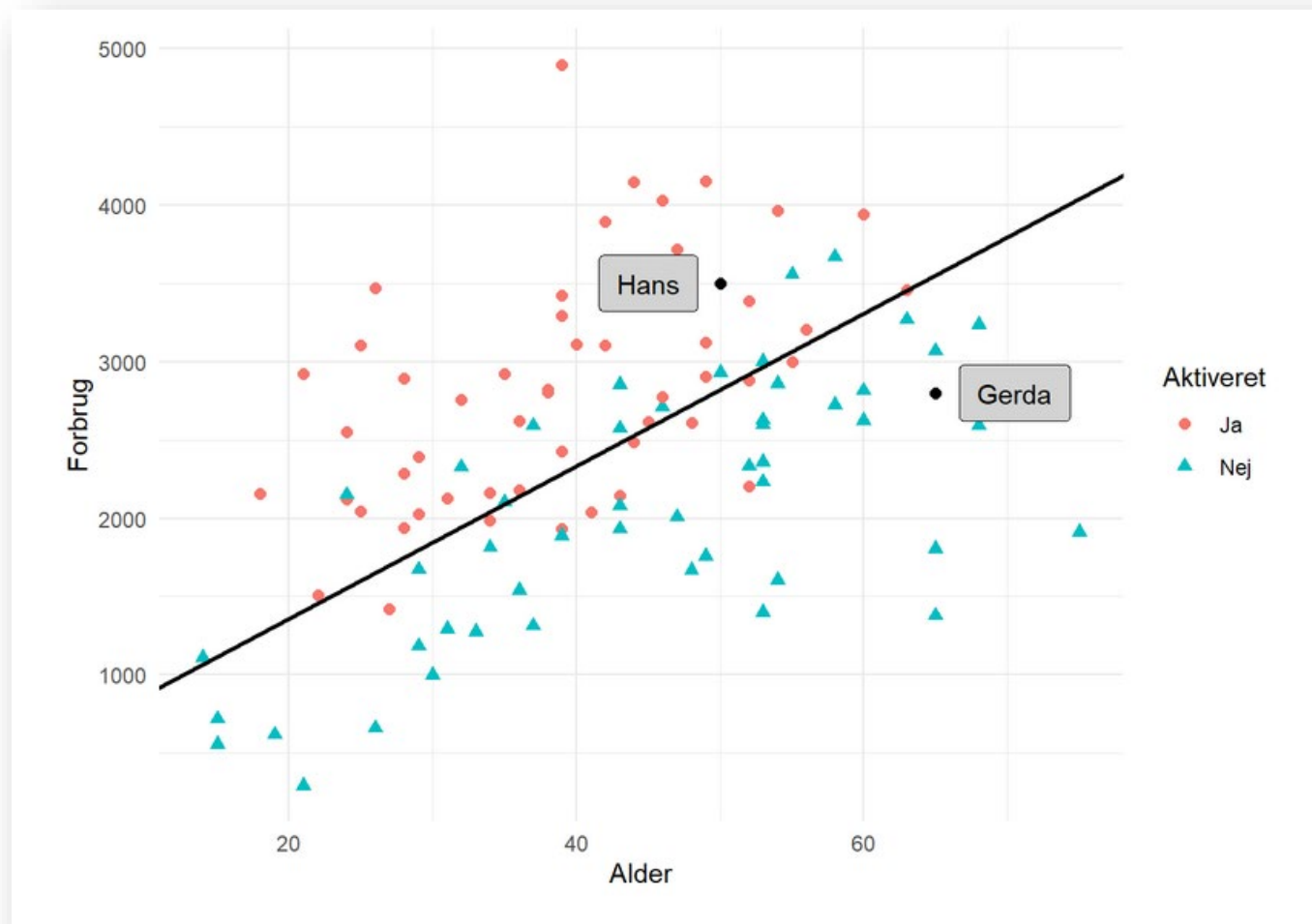
Gerda

$$o = \sigma(-0.298 - 0.0375 \cdot 65 + 0.000769 \cdot 2800) = 0.358$$

Vis tilbuddet til Hans, men ikke til Gerda!



Hans og Gerda i koordinatsystemet

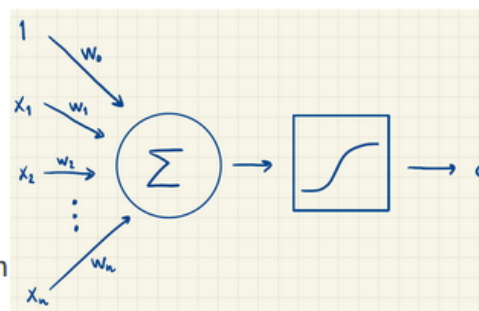




Det er sådan cirka indholdet af noten

Kunstige neuroner

Noten behandler byggestenene til de mere generelle kunstige neurale netværk nemlig de *kunstige neuroner*. De kunstige neuroner bruges – som de beskrives her – til binær klassifikation. Opdateringsreglerne til justering af vægtene udledes ved at minimere *squared error* tabsfunktionen ved hjælp af gradientnedstigning. Til forskel fra ADALINE algoritmen anvendes her **sigmoid**-funktionen som aktiveringsfunktion, og der er også henvisning til en side, som beskriver hvilken fordel det har sammenlignet med ADALINE.

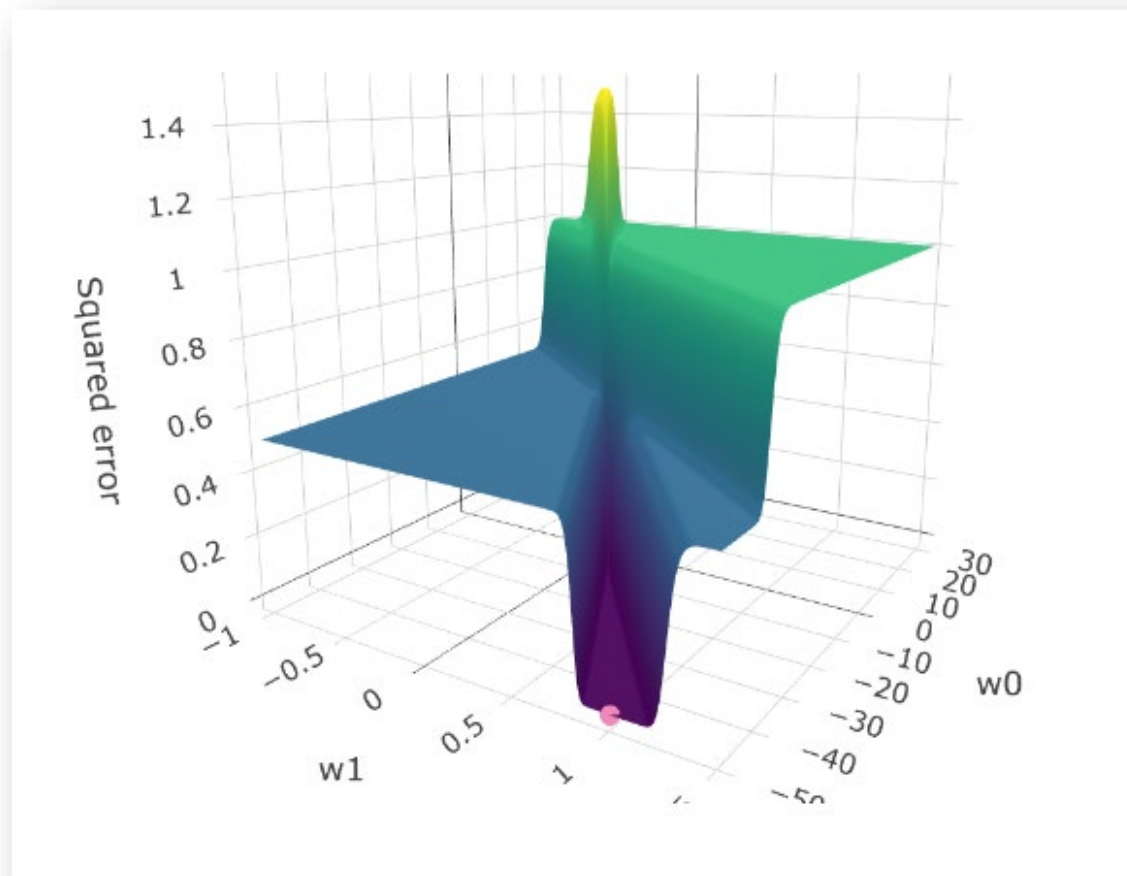


Type af klassifikation	Aktiveringsfunktion	Tabsfunktion
Binær	Sigmoid	<i>Squared error</i>

Sværhedsgrad: **



Problemer med *squared error* tabsfunktionen





Slow learning



Kunstig Neuron App

Data Variable **Træning** Krydsvalidering

Startvægte (alle sættes til samme værdi)
10

Learning rate (f.eks. 0.001)
0,001

Antal iterationer
100000

Tabsfunktion
Squared

Aktiveringsfunktion
Sigmoid

☒ Brug feature-skalering

Kør modellen

Tilbage Næste

Klassifikationsnøjagtighed (CA): 0.81

Graf for tabsfunktion

Slutværdi af tabsfunktion: 8.076

Vægte:

Show 10 entries

	feature	navn	vægt
	bias	w0	-5.212
Alder	Alder	w1	-0.5351
Forbrug	Forbrug	w2	0.01172

Previous 1 Next



Not so slow



Kunstig Neuron App

Data Variable **Træning** Krydsvalidering

Startvægte (alle sættes til samme værdi)

0

Learning rate (f.eks. 0,001)

0,001

Antal iterationer

100000

Tabsfunktion

Squared

Aktiveringsfunktion

Sigmoid

☒ Brug feature-skalering

Kør modellen 💡

Tilbage Næste

Klassifikationsnøjagtighed (CA): 0.79

Graf for tabsfunktion

Slutværdi af tabsfunktion: 6.804

Vægte:

Show 10 entries

	feature	navn	vægt
	bias	w0	-0.7462
Alder	Alder	w1	-0.145
Forbrug	Forbrug	w2	0.002865

Previous 1 Next



Slow learning...

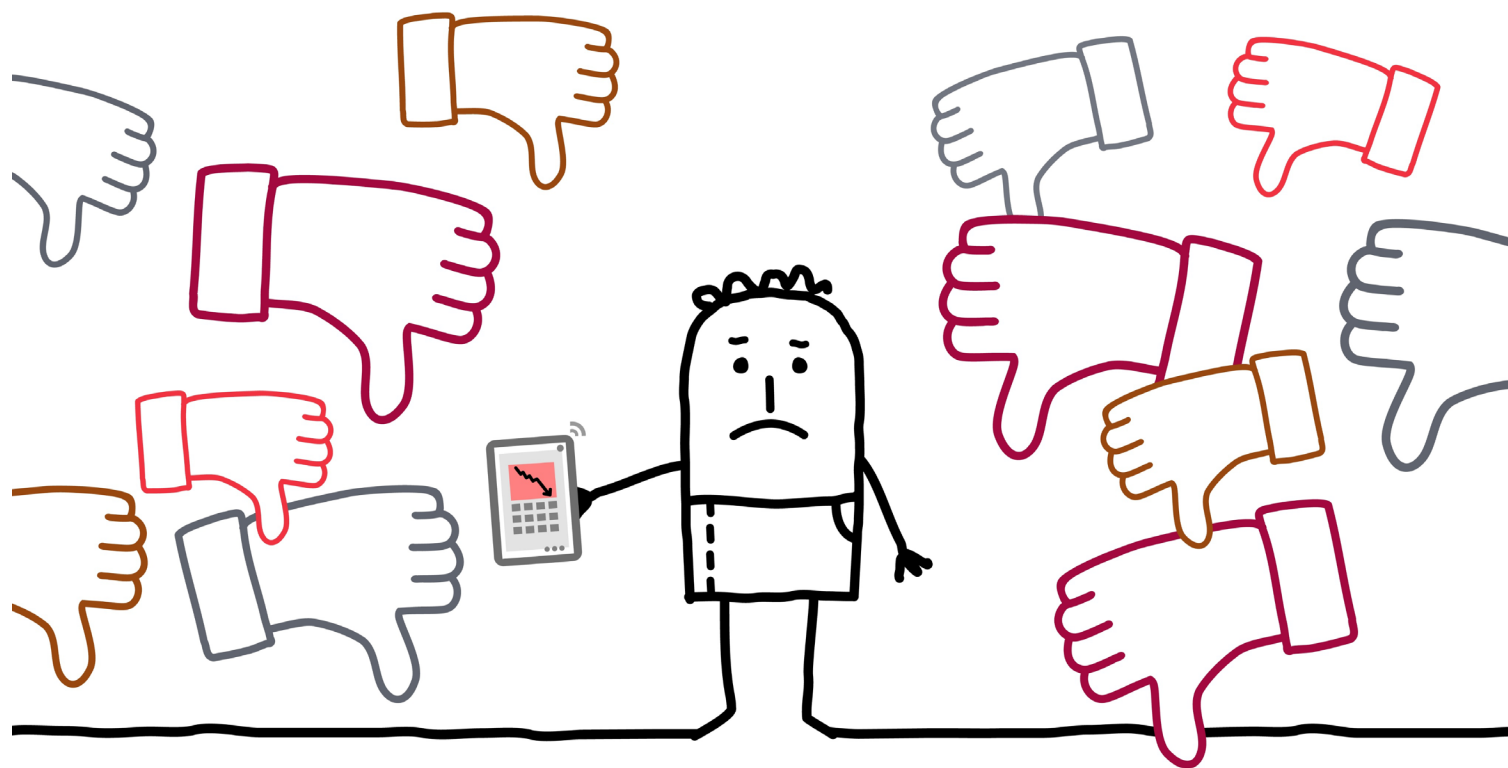


... opstår når neuronene er fejlagtigt mættet!

$$w_i^{(ny)} \leftarrow w_i + \eta \cdot \sum_{m=1}^M \left(t^{(m)} - o^{(m)} \right) \cdot o^{(m)} \cdot (1 - o^{(m)}) \cdot x_i^{(m)}$$



Squared error og sigmoid er et dårligt match!





Egenskaber ved en tabsfunktion



Egenskaber ved en tabsfunktion E



- Tabsfunktionen skal være positiv eller 0: $E \geq 0$ (vi vil ikke operere med negative fejl).
- Hvis AI modellen er god til at prædiktere, skal E være tæt på 0, mens hvis AI modellen er dårlig til at prædiktere, skal E være langt væk fra 0.



Cross-entropy tabsfunktioner

$$E(w_0, w_1, \dots, w_n) = - \sum_{m=1}^M (t^{(m)} \cdot \ln(o^{(m)}) + (1 - t^{(m)}) \cdot \ln(1 - o^{(m)}))$$

hvor

$$o^{(m)} = \sigma(w_0 + w_1 \cdot x_1^{(m)} + \dots + w_n \cdot x_n^{(m)})$$

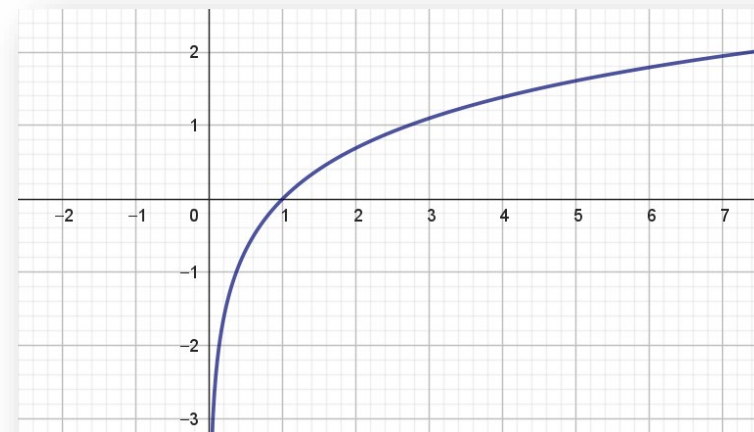


Cross-entropy tabsfunktioner

$$E(w_0, w_1, \dots, w_n) = - \sum_{m=1}^M (t^{(m)} \cdot \ln(o^{(m)}) + (1 - t^{(m)}) \cdot \ln(1 - o^{(m)}))$$

Hvis:

- $t^{(m)} = 0$: $(t^{(m)} \cdot \ln(o^{(m)}) + (1 - t^{(m)}) \cdot \ln(1 - o^{(m)})) = \ln(1 - o^{(m)})$
- $t^{(m)} = 1$: $(t^{(m)} \cdot \ln(o^{(m)}) + (1 - t^{(m)}) \cdot \ln(1 - o^{(m)})) = \ln(o^{(m)})$





Opdateringsregler med cross-entropy

$$w_i^{(ny)} \leftarrow w_i + \eta \cdot \sum_{m=1}^M (t^{(m)} - o^{(m)}) \cdot x_i^{(m)}$$

Bemærk, at faktoren $o^{(m)} \cdot (1 - o^{(m)})$ er ude!



Træning med cross-entropy

Kunstig Neuron App

Data Variable **Træning** Krydsvalidering

Startvægte (alle sættes til samme værdi)
10

Learning rate (f.eks. 0.001)
0,001

Antal iterationer
100000

Tabsfunktion
Cross-entropy

Activeringsfunktion
Sigmoid

☒ Brug feature-skaling

Kør modellen

Tilbage Næste

Klassifikationsnøjagtighed (CA): 0.77

Graf for tabsfunktion

Slutværdi af tabsfunktion: 40.99

Vægte:

Show 10 entries

	feature	navn	vægt
	bias	w0	-0.9323
Alder	Alder	w1	-0.1595
Forbrug	Forbrug	w2	0.003157

Previous 1 Next



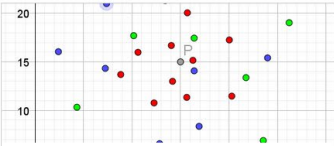
Langt forløb baseret på noten

 Undervisningsforløb Materialer SRP Apps

Om os  

Undervisningsforløb

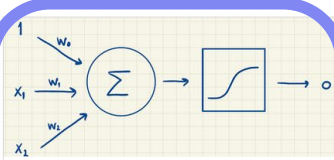
Forskellige undervisningsforløb til matematik i gymnasiet, som inddrager AI. Der findes forløb til både A-, B- og C-niveau.



Hvem ligner du mest?

C-NIVEAU | KORT

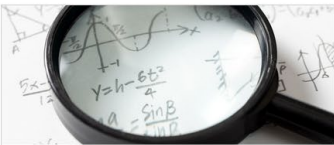
Evnen til at skelne mellem forskellige kategorier er helt central for os som mennesker. Vi kan langt oftest kende forskel på et æble og en pære, på en cykel og en knallert eller på om en person er en kvinde eller en mand. Men hvordan får man en computer til at gøre det samme?



Kunstige neuroner

A-NIVEAU | LANGT

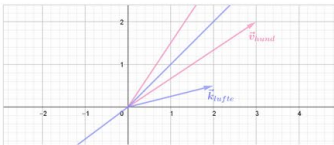
Et længere forløb om kunstige neuroner, som er den grundlæggende byggesten i kunstige neurale netværk. Forløbet kan bruges som supplerende stof på A-niveau, og der er forslag til spørgsmål til mundtlig eksamen.



Aktiveringsfunktioner

A-NIVEAU | KORT

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



Word2vec

A-NIVEAU | KORT

Algoritmen Word2Vec er et bud på, hvordan ord kan laves om til vektorer, hvilket er centralt for at forstå de store sprogmodeller. Idéen er at repræsentere hvert ord med en vektor på en sådan måde, at alle ord, hvis betydning ligner hinanden, repræsenteres

Kategorier

Alle (15)
A-niveau (7)
B-niveau (4)
Biologi (1)
Bioteknologi (1)
C-niveau (4)
Filosofi (1)
Kort (13)
Langt (2)
Psykologi (1)



Langt forløb om kunstige neuroner

Indhold

- Formål
- Forløbet
- Til læreren

Kunstige neuroner

A-NIVEAU LANGT

Forudsætninger og tidsforbrug

>

Formål

Dette længere forløb handler om kunstige neurale netværk, som er en vigtig metode indenfor kunstig intelligens.

Forløbet indeholder også et forslag til et spørgsmål til mundtlig eksamen, og kan derfor dække kravet på stx A om, at mindst ét spørgsmål skal stilles med udgangspunkt i det supplerende stof.



Langt forløb om kunstige neuroner

Forløbet

Forløbet består af følgende dele:

-  Del 1: Aktiveringsfunktioner
-  Del 2: Gradientnedstigning
-  Del 3: Tabsfunktioner og opdateringsregler
-  Del 4: Træning af kunstig neuron
-  Del 5: Krydsvalidering og modeludvælgelse
-  Del 6: Andre tabsfunktioner
-  Eksamensspørgsmål

Til læreren

Herunder findes diverse fif til læreren.

-  Til læreren



Eksamensspørgsmål 1

Differentialregning og kunstige neurale netværk

- Forklar, hvad der forstås ved en kunstig neuron, herunder hvordan sigmoid-funktionen kan bruges som aktiveringsfunktion.
- Gør rede for, hvordan sigmoid-funktionen

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

differentieres og vis, at

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



Eksamensspørgsmål 2

Differentialregning og kunstige neurale netværk

- Forklar, hvad der forstås ved en kunstig neuron, herunder hvordan *squared error* tabsfunktionen defineres.
- Gør rede for, hvordan man i tilfældet med et enkelt træningseksempel differentierer tabsfunktionen

$$E = \frac{1}{2} (t - \sigma(w_0 + w_1 \cdot x))^2$$

med hensyn til w_1 , samt hvordan dette bruges til opdatering af w_1 -vægten.



Eksamensspørgsmål 3

Differentialregning og kunstige neurale netværk

- Forklar, hvad der forstås ved en kunstig neuron, herunder hvordan *cross-entropy* tabsfunktionen defineres.
- Gør rede for, hvordan man i tilfældet med et enkelt træningseksempel differentierer tabsfunktionen

$$E = -(t \cdot \ln(o) + (1 - t) \cdot \ln(1 - o)),$$

hvor outputværdien o er

$$o = \sigma(w_0 + w_1 \cdot x)$$

med hensyn til w_1 , samt hvordan dette bruges til opdatering af w_1 -vægten.



Hands-on og kaffe

Forløbet

Forløbet består af følgende dele:

- AI** Del 1: Aktiveringsfunktioner
- AI** Del 2: Gradientnedstigning
- AI** Del 3: Tabsfunktioner og opdateringsregler
- AI** Del 4: Træning af kunstig neuron
- AI** Del 5: Krydsvalidering og modeludvælgelse
- AI** Del 6: Andre tabsfunktioner
- AI** Eksamensspørgsmål

Til læreren

Herunder findes diverse fif til læreren.

- AI** Til læreren



Start gerne her!





Evaluering



Evaluering Gymnasielærerdagen
2025

