

DANMARKS TEKNISKE UNIVERSITET

KURSUSNAVN	INTRODUKTION TIL INTELLIGENTE SYSTEMER
KURSUSNUMMER	02461
HJÆLPEMIDLER	ALLE HJÆLPEMIDLER ER TILLADT
VARIGHED	2 TIMER
VÆGTNING	ALLE DELOPGAVER VÆGTES ENS

INDHOLD

OPGAVE A: INDHOLDS-MODERERING	2
OPGAVE B: MODEL AF EN SVINGNING	4
OPGAVE C: CLUSTERING	6
OPGAVE D: SØGNING I EN LABYRINT	8
OPGAVE E: NEURALT NETVÆRK MED SiLU	9

BESVARELSE AF OPGAVERNE

Alle spørgsmål skal besvares med et entydigt resultat, som skal angives med understregning sidst i hver besvarelse. Besvarelsen skal altid underbygges med relevante betragtninger og/eller beregninger. Det skal klart fremgå hvilke teorier og formler der tages udgangspunkt i, og alle relevante mellemregninger skal medtages.

Opgave A Indholds-moderering

En social medievirksomhed har lavet en machine-learning model som bruges til at screene opslag. Modellen beslutter enten at opslaget publiceres med det samme, eller at det sendes til review hos en menneskelig redaktør. Redaktøren vurderer derefter om opslaget er acceptabelt eller problematisk, og vi antager at denne vurdering er sand og fair. Virksomheden ønsker nu at undersøge om machine learning modellen er fair over for to beskyttede grupper. Her ses virksomhedens data:

M: Model-beslutning	R: Redaktør-beslutning	G: Gruppe	
		A: Gruppe A	B: Gruppe B
Pub: Publicer		4 500	8 000
Rev: Review	Acc: Acceptabel	250	1 800
Rev: Review	Pro: Problematisk	250	200
Total		5 000	10 000

Spørgsmål A.1: Beregn sandsynligheden for at et opslag sendes til review, givet at den er skrevet af Gruppe B. Med andre ord, beregn $P(M=\text{Rev}|G=B)$.

Svar A.1:

$$P(M=\text{Rev}|G=B) = \frac{1\,800 + 200}{10\,000} = 20\%$$

Spørgsmål A.2: Beregn sandsynligheden for at et opslag vurderes som problematisk, givet at den er sendt til review. Med andre ord, beregn $P(R=\text{Pro}|M=\text{Rev})$.

Svar A.2:

$$P(R=\text{Pro}|M=\text{Rev}) = \frac{250 + 200}{250 + 250 + 1\,800 + 200} = 18\%$$

Spørgsmål A.3: Opskriv kriteriet for “Demografisk paritet” i samme notation som ovenfor, og beregn om kriteriet er opfyldt.

Svar A.3: Kriteriet “Demografisk paritet” kan skrives som

$$P(M=\text{Rev}|G=A) = P(M=\text{Rev}|G=B), \quad \text{eller ækvivalent} \quad P(M=\text{Pub}|G=A) = P(M=\text{Pub}|G=B).$$

Vi kan beregne

$$P(M=\text{Rev}|G=A) = \frac{250 + 250}{5\,000} = 10\%, \quad P(M=\text{Rev}|G=B) = \frac{1\,800 + 200}{10\,000} = 20\%.$$

Nej, demografisk paritet er ikke opfyldt da $10\% \neq 20\%$.

Spørgsmål A.4: Opskriv kriteriet for “Equalized opportunity” i samme notation som ovenfor, og beregn om kriteriet er opfyldt.

Svar A.4: Kriteriet “Equalized opportunity” kan skrives som

$$P(R=\text{Pro}|M=\text{Rev}, G=A) = P(R=\text{Pro}|M=\text{Rev}, G=B)$$

Vi kan beregne

$$P(\mathbf{R}=\mathbf{Pro}|\mathbf{M}=\mathbf{Rev}, \mathbf{G}=\mathbf{A}) = \frac{250}{250 + 250} = 50\%, \quad P(\mathbf{R}=\mathbf{Pro}|\mathbf{M}=\mathbf{Rev}, \mathbf{G}=\mathbf{A}) = \frac{200}{1\,800 + 200} = 10\%$$

Nej, equalized opportunity er ikke opfyldt da $50\% \neq 10\%$.

A ■

Opgave B Model af en svingning

Et elektrisk kredsløb, som varierer ved en sinus-svingning, kan beskrives med følgende formel

$$\hat{y}(t) = \sin(2\pi t + \theta),$$

hvor $\hat{y}$ er modellen af det målte signal, t er tiden, og θ er en ukendt parameter (fasen). Vi ønsker nu at bestemme fasen θ ud fra følgende måledata:

t	0	1
y	0.8	0.9

Vi benytter den kvadratiske fejl som loss-funktion

$$L(\theta) = \sum_{i=1}^2 (y_i - \hat{y}(t_i))^2.$$

Spørgsmål B.1: Falder denne model inden for det vi kalder "lineær regression"? Beskriv hvorfor / hvorfor ikke.

Svar B.1: Her er tale om en regressionsmodel, men det er ikke lineær regression da funktionen ikke er lineær i parameteren θ .

Spørgsmål B.2: Beregn loss-funktionens værdi for $\theta = 0$.

Svar B.2:

$$\begin{aligned} L(0) &= \sum_{i=1}^2 (y_i - \sin(2\pi t_i + \theta))^2 \\ &= (0.8 - \sin(2\pi \cdot 0 + 0))^2 + (0.9 - \sin(2\pi \cdot 1 + 0))^2 \\ &= 0.8^2 + 0.9^2 = 0.64 + 0.81 = \underline{1.45} \end{aligned}$$

Spørgsmål B.3: Beregn gradienten af loss-funktionen for $\theta = 0$.

Svar B.3:

$$\begin{aligned} \left. \frac{dL}{d\theta} \right|_{\theta=0} &= -2 \sum_{i=1}^2 (y_i - \hat{y}_i) \cdot \cos(2\pi t_i + \theta) \\ &= -2 \cdot ((0.8 - \sin(2\pi \cdot 0 + 0)) \cos(2\pi \cdot 0) + (0.9 - \sin(2\pi \cdot 1 + 0)) \cdot \cos(2\pi \cdot 1)) \\ &= -2 \cdot (0.8 + 0.9) = \underline{-3.4} \end{aligned}$$

Loss-funktionen kan minimeres ved gradient-nedstigning, som er implementeret i den følgende PyTorch kode; en del af koden mangler, markeret med et spørgsmålstegn.

```
t, y = torch.tensor([0., 1.]), torch.tensor([0.8, 0.9]) # Data
theta = torch.tensor([0.], requires_grad=True) # Parameter to optimize
lr = 0.1 # Learning rate
for step in range(100): # Gradient descent loop
    loss = torch.sum((y - torch.sin(2*torch.pi*t + theta))**2) # Forward pass
    loss.backward() # Backward pass
    with torch.no_grad():
        theta -= ? # Gradient descent update
    theta.grad.zero_() # Zero gradients
```

Spørgsmål B.4: Hvad skal der stå på spørgsmålstegnets plads?

Svar B.4: Det er selve gradient-opdateringen der mangler, og her skal gradienten ganges med learning rate. Så der skal stå `lr * theta.grad`.

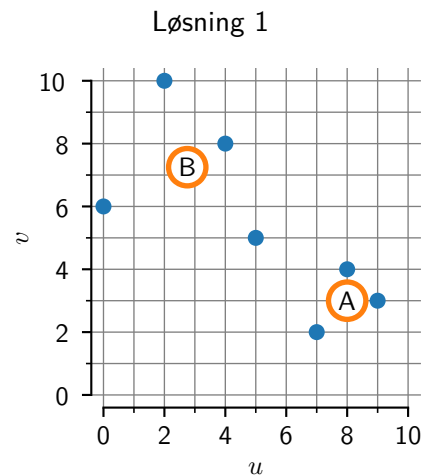
B ■

Opgave C Clustering

Figuren til højre viser $N = 7$ datapunkter (blå cirkler), $\mathbf{x}_i = (u_i, v_i)$. Koordinaterne for alle datapunkter er heltal. Ved brug af k-means clustering er punkterne delt op i to grupper. Centrene for de to grupper, $\mathbf{z}_A$ og $\mathbf{z}_B$, er vist som orange cirkler.

K-means clustering minimerer den følgende cost-funktion:

$$L = \sum_{i=1}^N \|\mathbf{x}_i - \mathbf{z}_{c_i}\|^2.$$



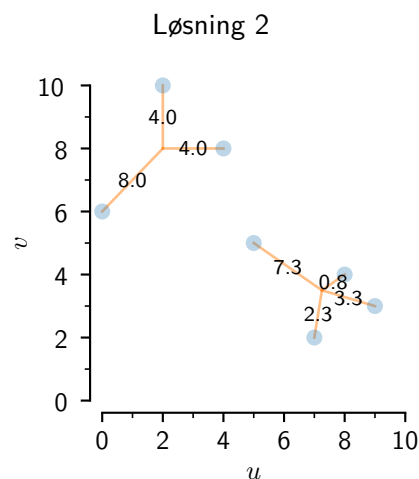
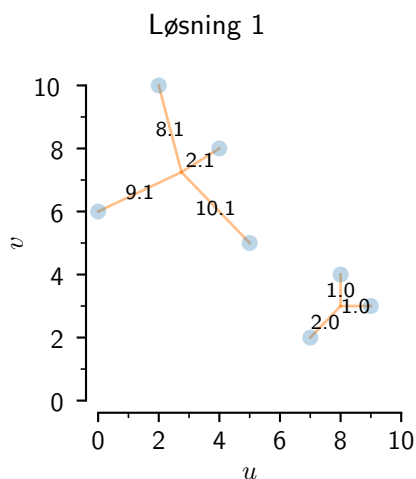
Spørgsmål C.1: Beregn koordinaterne for cluster center B.

Svar C.1: Det ses af figuren at de fire datapunkter længst til venstre tilhører cluster B, hvis koordinater kan beregnes som gennemsnittet heraf:

$$u_B = \frac{0 + 2 + 4 + 5}{4} = 2.75, \quad \text{og} \quad v_B = \frac{6 + 10 + 8 + 5}{4} = 7.25.$$

Koordinaterne er dermed $(2.75, 7.25)$.

Figurene nedenfor viser de kvadrerede afstande fra hvert punkt til det nærmeste cluster center for to forskellige mulige løsninger. Figuren nedenfor til venstre svarer til løsningen ovenfor.



Spørgsmål C.2: Beregn værdien af cost-funktionen for de to løsninger og bestem hvilken løsning der er bedst.

Svar C.2: Cost-funktionen er summen af de kvadratiske afstande. For de to løsninger får vi

$$L_1 = 8.1 + 9.1 + 2.1 + 10.1 + 1.0 + 1.0 + 2.0 = 33.4$$

$$L_2 = 4.0 + 4.0 + 8.0 + 7.3 + 0.8 + 3.3 + 2.3 = 29.7$$

Heraf ses at vi foretrækker Løsning 2.

Spørgsmål C.3: Hvis vi standardiserer u -koordinaterne ved hjælp af formlen $\tilde{u} = \frac{u - \mu_u}{\sigma_u}$, hvad bliver da den nye $\tilde{u}$ -koordinat for datapunktet der ligger ved $u = 5$?

Svar C.3: Først må vi beregne middelværdien af u -koordinaterne:

$$\mu_u = \frac{0 + 2 + 4 + 5 + 7 + 8 + 9}{7} = 5$$

Herefter ses det at vi ikke behøver beregne standardafvigelsen, idet

$$\tilde{u} = \frac{u - \mu_u}{\sigma_u} = \frac{5 - 5}{\sigma_u} = \underline{0}.$$

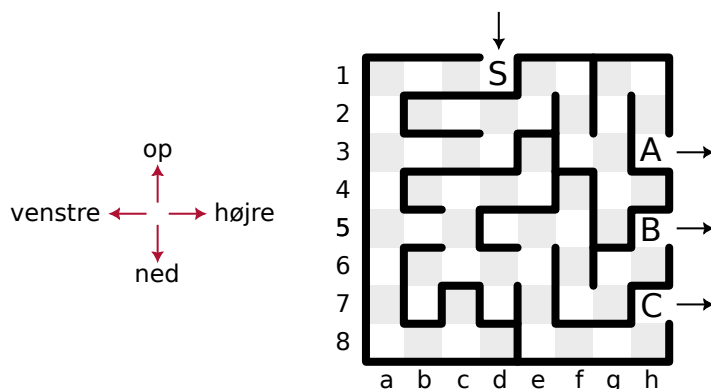
Spørgsmål C.4: Beregn variansen af u .

Svar C.4: Vi har fra tidligere at $\mu_u = 5$. Variansen kan så beregnes som

$$\begin{aligned}\sigma_u^2 &= \frac{1}{N} \sum_{i=1}^N (u_i - \mu_u)^2 = \frac{1}{7} \left((0 - 5)^2 + (2 - 5)^2 + (4 - 5)^2 + (5 - 5)^2 + (7 - 5)^2 + (8 - 5)^2 + (9 - 5)^2 \right) \\ &= \frac{1}{7} (5^2 + 3^2 + 1^2 + 0^2 + 2^2 + 3^2 + 4^2) = \frac{25 + 9 + 1 + 4 + 9 + 16}{7} = \frac{64}{7} \approx \underline{9.14}.\end{aligned}$$

Opgave D Søgning i en labyrint

Labyrinten nedenfor har i alt $n = 8 \cdot 8 = 64$ felter. Vi vil benytte søgning til at finde en vej fra indgangen til labyrinten (S) til en af udgangene (A, B, eller C). Når vi udforsker labyrinten i søgnings-algoritmen, benytter vi følgende rækkefølge: 1. højre, 2. venstre, 3. op, 4. ned, i henhold til det viste koordinatsystem.



Spørgsmål D.1: Hvis vi benytter en bredde-først-søgning, hvilken udgang finder vi da først? Giv en kort forklaring.

Svar D.1: Ved bredde-først-søgning finder man først alle felter der ligger 1 skridt fra startfeltet, herefter alle der ligger 2 skridt fra osv. Derfor vil den udgang der kan nås fra startfeltet i færrest skridt blive fundet. Det er udgang C.

Spørgsmål D.2: Hvis vi benytter dybde-først-søgning, hvilken udgang finder vi da først? Giv en kort forklaring.

Svar D.2: Ved dybde-først-søgning afgøres resultatet af den rækkefølge søgningen benytter. Det afgørende øjeblik er når vi står på felt a3. Her afsøges først mod højre, og dybde-først-søgningen finder således først udgang A.

Spørgsmål D.3: Hvor mange felter vil blive besøgt under dybde-først-søgningen? Angiv antallet inkl. start-feltet (S) og slut-feltet (A, B, eller C).

Svar D.3: Søgning vil i dette tilfælde følge den korteste sti til A: Ved a3 går vi først til højre; ved d2 går vi igen først til højre; og ved g3 går vi først op. Dermed besøger vi i alt 21 felter.

Forestil dig nu en vilkårlig labyrint med et vilkårligt antal felter, n . Labyrinten har altid én indgang og mindst én udgang, som kan nås fra indgangen. Vi vil måle beregnings-kompleksiteten ud fra, hvor mange felter der besøges under søgningen, og vil undersøge, hvordan beregnings-kompleksiteten vokser med labyrintens størrelse.

Spørgsmål D.4: Hvad er worst-case-kompleksiteten (i store-O notation) for bredde-først-søgning, udtrykt i antallet af felter n i labyrinten? Giv en kort forklaring.

Svar D.4: I det værste tilfælde kan vi risikere at skulle besøge alle felter i labyrinten. Dermed er kompleksiteten lineær, $O(n)$.

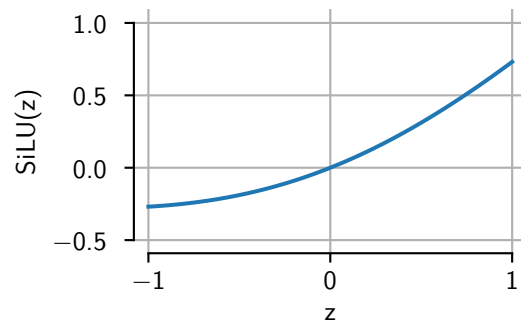
Opgave E Neurtalt netværk med SiLU

En ofte anvendt aktiveringsfunktion i neurale netværk kaldes SiLU. Den er defineret således

$$\text{SiLU}(z) = \frac{z}{1 + e^{-z}}.$$

En neuron med to input, x_1 og x_2 , og SiLU aktivering kan defineres således:

$$f(x_1, x_2) = \text{SiLU}(w_0 + w_1 \cdot x_1 + w_2 \cdot x_2).$$



Følgende kode implementerer denne neuron i PyTorch.

```
model = torch.nn.Sequential(torch.nn.Linear(2,1), torch.nn.SiLU())
```

Vi antager nu at parametrene er givet ved $w_0 = w_1 = w_2 = 1$.

Spørgsmål E.1: Beregn neuronens output for inputtet $x_1 = 1$ og $x_2 = 2$, dvs. $f(1, 2)$.

Svar E.1: Outputtet er givet ved

$$f(1, 2) = \text{SiLU}(1 + 1 \cdot 1 + 1 \cdot 2) = \text{SiLU}(4) = \frac{4}{1 + e^{-4}} \approx \underline{3.93}$$

Spørgsmål E.2: Beregn gradienten af funktion f for inputtet $x_1 = 1$ og $x_2 = -2$, dvs $\nabla f(1, -2)$.

Svar E.2: Gradienten er givet ved

$$\nabla f(1, -2) = \left[\frac{\partial f}{\partial w_0}, \frac{\partial f}{\partial w_1}, \frac{\partial f}{\partial w_2} \right]^\top$$

Vi kan benytte kædereolen,

$$\frac{\partial f}{\partial w_i} = \frac{d\text{SiLU}(z)}{dz} \frac{\partial z}{\partial w_i} = \frac{d\text{SiLU}(z)}{dz} x_i$$

hvor vi definerer $x_0 = 1$. Vi definerer nu $z^* = w_0 + w_1 \cdot x_1 + w_2 \cdot x_2 = 1 + 1 \cdot 1 + 1 \cdot (-2) = 0$.

Lad os nu beregne den afledte af SiLU funktionen vha. kvotientreglen

$$\left. \frac{d\text{SiLU}(z)}{dz} \right|_{z=z^*} = \frac{1 + e^{-z} + ze^{-z}}{(1 + e^{-z})^2} = \frac{1 + e^0 + 0e^0}{(1 + e^0)^2} = \frac{2}{4} = 0.5$$

og indsætter i udtrykket ovenfor hvilket giver det endelige resultat

$$\nabla f(1, -2) = [0.5, 0.5, -1]^\top$$

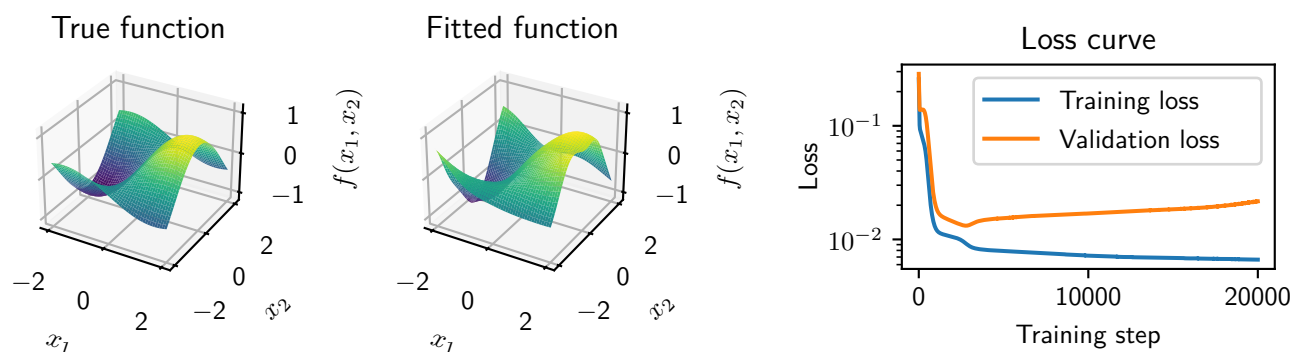
Nu definerer vi et neuralt netværk med et skjult lag bestående af 32 neuroner med SiLU-aktivering, efterfulgt af et lineært outputlag.

```
model = torch.nn.Sequential(  
    torch.nn.Linear(2,32),  
    torch.nn.SiLU(),  
    torch.nn.Linear(32,1)  
)
```

Spørgsmål E.3: Hvor mange parametre har dette neurale netværk i alt?

Svar E.3: Første lag har 32 neuroner med 3 parametre i hver (2 vægte + 1 bias), hvor andet lag har 32 vægte + 1 bias. I alt 129 parametre.

Vi har nu trænet det neurale netværk på nogle støjfyldte observationer af en underliggende funktion. Nedenfor ses både den sande funktion og netværkets tilpassede approximation. Derudover vises udviklingen i loss på træningssættet samt på et uafhængigt valideringssæt genereret fra den sande funktion.



Spørgsmål E.4: Ses der nogen tegn på overfitting? Begrund dit svar.

Svar E.4: Man kan se, at validerings-loss først falder, men derefter begynder at stige igen. Det viser, at den endelige model ikke opnår den bedste generalisering. Den stigende validerings-loss er et tydeligt tegn på overfitting.