Danmarks Tekniske Universitet

Kursusnavn	Introduktion til Intelligente Systemer
Kursusnummer	02461
Hjælpemidler	Alle hjælpemidler er tilladt
Varighed	2 timer
Vægtning	Alle delopgaver vægtes ens

Indhold

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 En ny behandlingsform

En eksisterende behandlingsform har gennem mange år vist sig at være 67% effektiv til at kurere en sygdom. Nu er der opfundet en ny behandlingsform, og vi er interesserede i at fastslå om den er bedre end den gamle. I et mindre pilotstudie har den nye behandling vist sig effektiv på 24 ud af 30 patienter.

Spørgsmål A.1: Kan du på baggrund af pilotstudiet konkludere at den ny behandling er mere effektiv end den gamle?

Svar A.1: Den estimerede effektivitet af den nye behandling er

$$p = \frac{24}{30} = 0.8 = 80\%$$

hvilket er højere end den eksisterende behandlings effektivitet på 67%. Vi må dog undersøge om der er tilstrækkelig stor statistisk evidens for at den nye behandlings effektivitet reelt set er større. Et 95% konfidensinterval for effektiviteten beregnes som

$$p \pm 1.96 \cdot \sqrt{\frac{p(1-p)}{n}} = 0.8 \pm 1.96 \cdot \sqrt{\frac{0.8(1-0.8)}{30}} \approx 0.8\% \pm 0.14$$

Grænserne for intervallet er således [66%, 94%]. Da effektiviteten af den eksisterende behandling på 67% er indeholdt i intervallet, kan vi ikke med sikkerhed konkludere at den nye behandling er mere effektiv.

Nej

Vi ønsker nu at lave et større klinisk forsøg, hvor vi afprøver den nye behandling på n patienter. Vi vil gerne fastlå behandlingens effektivitet i form af et 95% konfidensinterval med en fejlmargin på ± 1 procentpoint.

Spørgsmål A.2: Hvilken stikprøvestørrelse n vil du anbefale til det større kliniske forsøg?

Svar A.2: Et estimat af den nødvendige stikprøvestørrelse beregnes med følgende formel, hvor vi indsætter $p = 0.8$ fra pilotstudiet

$$n = 1.96^2 \frac{p(1-p)}{e^2} = 1.96^2 \frac{0.8(1-0.8)}{0.01^2} \approx \underline{6147}$$

Af forskellige årsager ender man med at lave et forsøg med $n = 1000$ patienter, hvor behandlingen viser sig effektiv i 801 ud af de 1000 patienter.

Spørgsmål A.3: Beregn et 95% konfidensinterval for behandlingens effektivitet. Angiv intervallets øvre og nedre grænse.

Svar A.3: I det større studie er den estimerede effekt

$$p = \frac{801}{1000} = 0.801$$

Konfidensintervallet beregnes som

$$p \pm 1.96 \cdot \sqrt{\frac{p(1-p)}{n}} = 0.8 \pm 1.96 \cdot \sqrt{\frac{0.801(1-0.801)}{1000}} \approx 0.80\% \pm 0.025$$

Konfidensintervallets grænser er således $\underline{[78\%, 83\%]}$.

I forbindelse med forsøget bliver man opmærksom på, at behandlingens effektivitet muligvis påvirkes af hvorvidt patienterne er vaccineret for influenza. Forsøgets resultater er opsummeret i følgende tabel:

	Vaccineret	Ikke vaccineret	Total
Kureret	482	319	801
Ikke kureret	118	81	199
Total	600	400	1000

Spørgsmål A.4: Hvad kan du konkludere om vaccinens indflydelse på behandlingen på baggrund af disse data?

Svar A.4: For de vaccinerede kan et konfidensintval for effektiviteten beregnes som

$$p = \frac{482}{600} \approx 0.803 p \pm 1.96 \cdot \sqrt{\frac{p(1-p)}{n}} = 0.803 \pm 1.96 \cdot \sqrt{\frac{0.803(1-0.803)}{600}} \approx 0.80\% \pm 0.032$$

mens det for de ikke-vaccinerede er

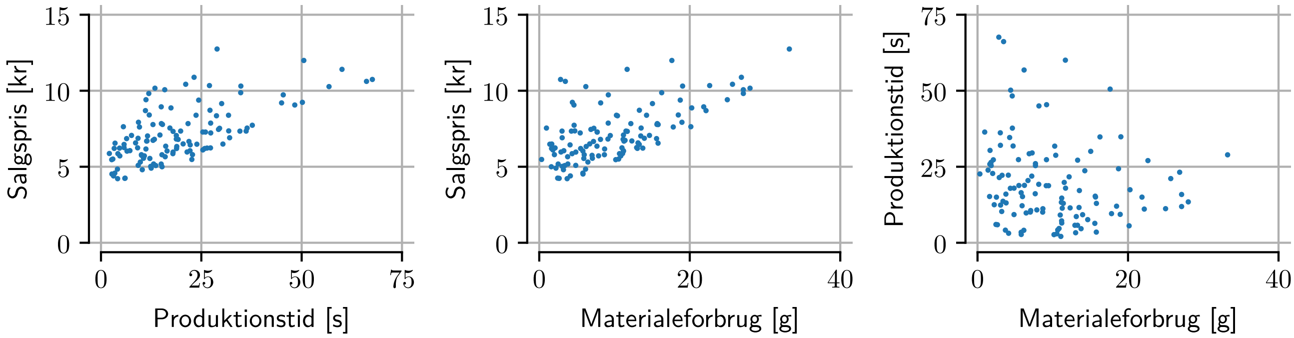
$$p = \frac{319}{400} \approx 0.798 p \pm 1.96 \cdot \sqrt{\frac{p(1-p)}{n}} = 0.798 \pm 1.96 \cdot \sqrt{\frac{0.798(1-0.798)}{400}} \approx 0.80\% \pm 0.039$$

Da effekten for de to grupper er stort set identisk og konfidensintervallerne overlapper, kan vi konkludere at der er ikke ud fra dette studie evidens for at vaccinen har indflydelse på behandlingens effekt.

A

Opgave B Metalprodukter

En virksomhed producerer 120 forskellige små metalprodukter (skruer, møtrikker mv.) og ønsker nu at undersøge sammenhængen mellem materialeforbrug (M) i gram, produktionstid (P) i sekunder, og salgspris (S) i kroner. Derfor har de indsamlet disse data for alle deres produkter, vist herunder.



Virksomheden har fundet frem til følgende lineære regression, som nogenlunde beskriver sammenhængen:

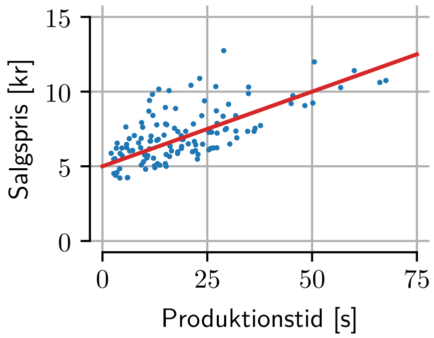
$$S = 3 + 0.2 \cdot M + 0.1 \cdot P$$

Spørgsmål B.1: Hvad vil prisen være for et produkt med et materialeforbrug på 20 gram og en produktionstid på 75 sekunder, ifølge virksomhedens model?

Svar B.1: Prisen for et produkt med materialeforbrug $M = 20$ og produktionstid $P = 75$ beregnes til

$$S = 3 + 0.2 \cdot M + 0.1 \cdot P = 3 + 0.2 \cdot 20 + 0.1 \cdot 75 = \underline{14.5 \text{ [s]}}$$

I figuren nedenfor har man ud fra modellen indtegnet en regressionsline, hvor materialeforbruget M er holdt fast på den gennemsnitlige materialeforbrug μ_M.



Spørgsmål B.2: Bestem det gennemsnitlige materialeforbrug μ_M.

Svar B.2: Figuren kan aflæses i punktet $P = 0$ til værdien $S = 5$. For dette punkt kan regressionsligningen skrives som

$$5 = 3 + 0.2 \cdot \mu_M + 0.2 \cdot 0$$

Vi kan nu løse for det gennemsnitlige materialeforbrug

$$\mu_M = \frac{5-3}{0.2} = \underline{10 \text{ [g]}}$$

Virksomhedens ejer ønsker nu at benytte en ikke-lineær model til at beskrive sammenhængen, og overvejer følgende model,

$$S = \tanh(w_0 + w_1 \cdot M + w_2 \cdot P).$$

Det påtænkes at bestemme koefficienterne w_0, w_1, og w_2 ved hjælp af gradient-nedstignings-algoritmen og at benytte en kvadratisk fejl-funktion.

Spørgsmål B.3: Er denne ikke-lineære model og tilgang til parameterestimering fornuftig efter din mening?

Svar B.3: Anvendelse af en ikke-linær model og parameterestimering med gradientnedstigning kan som sådan være en udmærket ide. Vi bemærker dog at der foreslås en model hvor S modelleres med en hyperbolsk tangens som aktiveringsfunktion, som har værdimængden $]-1,1[$. Da salgspriserne cirka ligger mellem 4 og 13 kr, kan vi se at det <u>ikke vil være en passende tilgang.</u>

Spørgsmål B.4: Giv ud fra graferne et (groft) estimat af korrelations-koefficienten mellem Materialeforbrug og Produktionstid.

Svar B.4: Ud fra plottet af materialeforbrug og produktionstid, ses ikke nogen lineær sammenhæng mellem de to størrelser. Et groft, visuelt estimat er derfor at <u>korrelationskoefficienten er cirka nul.</u>

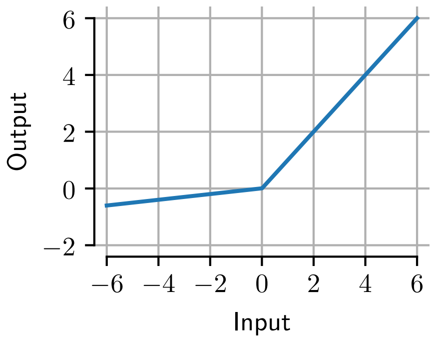
B

Opgave C En neuron med LeakyReLU aktivering

LeakyReLU er en ofte brugt aktiveringsfunktion i neurale netværk, og er defineret som

$$\text{LeakyReLU}(z) = \begin{cases} z & z \geq 0 \\ \alpha \cdot z & z < 0 \end{cases}$$

hvor α er et lille positivt tal. Herunder er funktionen vist for $\alpha = 0.1$:



En neuron med tre input, x_1, x_1, og x_3, og LeakyReLU aktivering kan defineres som

$$f(x_1, x_2, x_3) = \text{LeakyReLU}(w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3).$$

I de følgende to spørgsmål antager vi at $\alpha = 0.1$, input er givet ved $x_1 = 1$, $x_2 = 2$, $x_3 = 3$, og vægtene er givet ved $w_0 = -10$, $w_1 = w_2 = w_3 = 1$.

Spørgsmål C.1: Beregn neuronens output f.

Svar C.1: Neuronens output kan beregnes som

$$f = \text{LeakyReLU}(-10 + 1 \cdot 1 + 1 \cdot 2 + 1 \cdot 3) = \text{LeakyReLU}(-4) = 0.1 \cdot (-4) = \underline{-0.4}$$

Spørgsmål C.2: Beregn gradienten af funktionen f med hensyn til modellens parametre.

Svar C.2: Gradienten er givet ved

$$\nabla f = \left[\frac{\partial f}{\partial w_0}, \frac{\partial f}{\partial w_1}, \frac{\partial f}{\partial w_2}, \frac{\partial f}{\partial w_3}\right]^\top,$$

hvor hver partielle afledte kan beregnes som

$$\frac{\partial f}{\partial w_i} = \frac{\partial \text{LeakyReLU}(z)}{\partial z} \frac{\partial z}{\partial w_i} = 0.1 \cdot x_i,$$

hvor $z = \sum_{i=0}^{3} w_i x_i$ og vi indfører $x_0 = 1$, og $\frac{\partial \text{LeakyReLU}(z)}{\partial z} = 0.1$ idet z er negativ. Vi får dermed

$$\nabla f = [0.1 \cdot 1, 0.1 \cdot 1, 0.1 \cdot 2, 0.1 \cdot 3]^\top = \underline{[0.1, 0.1, 0.2, 0.3]^\top}$$

I Pytorch kan man i praksis finde gradienten af en sådan neuron på følgende måde:

```
linear = torch.nn.Linear(3, 1)
activation = torch.nn.LeakyReLU(0.1)
net = torch.nn.Sequential(linear, activation)
x = torch.tensor([1., 2., 3.])
f = net(x)
f.backward()
```

Her er `net` et objekt som indeholder to elementer: Et lineært lag og en LeakyReLU aktiveringsfunktion. Det lineære lag indeholder to parametre `bias` og `weight` som henholdsvis svarer til vægtene $[w_0]$ og $[w_1, w_2, w_3]$.

Spørgsmål C.3: Beskriv hvad der foregår, når funktionen `f.backward()` kaldes i koden ovenfor. Hvad beregnes og hvordan foregår det?

Svar C.3: Når funktionen `f.backward()` kaldes udføres "reverse mode automatic differentiation" på den beregningsgraf der underligger variable `f`. Herved beregnes gradienten af f med hensyn til alle modelparametre.

Spørgsmål C.4: Hvis vi kører ovenstående kode og undersøger resultatet af gradientberegningen i `linear.bias.grad` og `linear.weight.grad` så vil de ikke nødvendigvis svare overens med den gradient vi har beregnet i det forgående spørgsmål. Hvorfor ikke, og hvilke resultater er mulige?

Svar C.4: Når den ovenstående kode køres, initialiseres modelles parametre $w_0, \ldots w_3$ med tilfældige værdier, på en måde som er defineret i Pytorch, og dermed er det ukendt om z er positiv eller negativ. Hvis z er negativ fås den gradient vi har beregnet ovenfor, men hvis z er positiv fås

$$\nabla f = [0.1 \cdot 1, 0.1 \cdot 1, 0.1 \cdot 2, 0.1 \cdot 3]^\top = [1, 1, 2, 3]^\top ,$$

og disse er de to eneste muligheder.

C

Opgave D De største tal

Funktionen nedenfor tager som input en liste med tal, og returnerer en ny liste som indeholder alle de tal der er større end 1000.

```
def get_big(numbers):
    big_numbers = []
    for num in numbers:
        if num > 1000:
            big_numbers.append(num)
    return big_numbers
```

Hver sammenligning (`num>1000`) og hver tilføjelse til listen (`big_numbers.append(num)`) tæller som 1 operation, og længden af listen `numbers` er n.

Spørgsmål D.1: Hvad er henholdsvis best-case og worst-case tidskompleksiteten $T(n)$ af funktionen målt i antal operationer?

Svar D.1: Algoritmen gennemløber alle tal på listen og sammenligner med 1000. Dette kræver n sammenligninger. Alle tal der er større end 1000 tilføjes til den nye liste. Best case tidskompleksiteten (hvis alle tal er mindre end 1000) er $\underline{T(n) = n}$, og worst case tidskompleksiteten (hvis alle tale er størren end 1000) er $\underline{T(n) = 2n}$.

Spørgsmål D.2: Hvad er kompleksiteten i store-O notation?

Svar D.2: Både best og worst case kompleksiteten er lineær, da største term i udtrykket er proportionalt med n, så kompleksiteten er $\underline{O(n)}$.

En simpel måde at finde de fem største tal i en liste er at sortere listen og udtrække de fem sidste værdier:

```
top_five = sorted(numbers)[-5:]
```

Som bekendt er kompleksiteten af at sortere en liste $O(n \log n)$, hvor n er længden af listen. Alternativt kan man benytte følgende algoritme:

```
top_five = numbers[0:5]
smallest_in_top_five = min(top_five)
for num in numbers[5:]:
    if num > smallest_in_top_five:
        top_five.remove(smallest_in_top_five)
        top_five.append(num)
        smallest_in_top_five = min(top_five)
```

Algoritmen starter med at tage de første fem tal på listen som den gemmer i en ny liste kaldet `top_five`. Herefter gennemløber den de resterende tal på listen, og hver gang den finder et tal som er større end det mindste på listen `top_five`, erstatter den det mindste tal med det den har fundet. Når hele listen `numbers` er gennemløbet vil listen `top_five` således indeholde de fem største tal.

Spørgsmål D.3: Hvad skal gælde for listen `numbers` for at opnå henholdsvis et best-case og et worst-case tilfælde?

Svar D.3: I denne algoritme er best case når de fem største tal ligger først på listen, således at `top_five` aldrig skal opdateres.

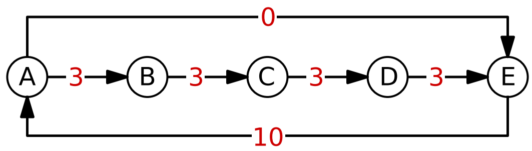
Worst case er hvor hvert nyt tal på listen er større end det mindste i `top_five`, fx. hvis listen er sorteret i stigende rækkefølge.

Spørgsmål D.4: Hvad er kompleksiteten af denne algoritme i store-O notation?

Svar D.4: Algoritmen gennemløber alle tal på listen og foretager en konstant, $O(1)$, mængde beregninger for hvert element. Heraf kan konkluderes at algoritmen samlet er $\underline{O(n)}$.

Opgave E System med fem tilstande

Figuren nedenfor illustrerer et system med fem mulige tilstande, A–E.



Pilene mellem tilstandende angiver de mulige transitioner, og tallene på pilene angiver den belønning der er forbundet med hver transition. Vi benytter en discount-factor $\gamma = 0.9$, og får oplyst at værdien af tilstand D er givet som $v(D) = 50.4$

Spørgsmål E.1: Hvad er værdien af tilstand C?

Svar E.1: Værdien kan beregnes udfra den rekursive definition af værdi:

$$v(s) = \max_a \left(r(s,a) + \gamma v(s')\right)$$

og da der kun er en mulig action får vi

$$V(C) = 3 + 0.9 \cdot v(D) = 3 + 0.9 \cdot 50.4 \approx \underline{48.4}$$

Vi får yderligere oplyst at værdien af tilstand B er givet som $v(B) = 46.5$ og værdien af tilstand E er givet som $v(E) = 52.6$.

Spørgsmål E.2: Hvad er værdien af tilstand A?

Svar E.2: Igen benyttes den rekursive definition af værdi. Denne gang er det to mulige actions, nemlig at gå til tilstand B eller tilstand E. Vi får dermed

$$V(A) = \max\left(3 + \gamma V(B), 0 + \gamma V(E)\right) = \max\left(3 + 0.9 \cdot 46.5, 0 + 0.9 \cdot 52.6\right) \approx \max(44.9, 47.3) = \underline{47.3}$$

Spørgsmål E.3: Hvad er den optimale policy?

Svar E.3: For at finde den optimale policy, behøver vi kun at fastlå den optimale action i state A, som er eneste sted der er mere end en mulig action. At gå fra A til B giver os værdien

$$3 + \gamma V(B) = 3 + 0.9 \cdot 46.5 \approx 44.9.$$

At gå fra A til E giver os værdien

$$0 + \gamma V(E) = 0 + 0.9 \cdot 52.6 \approx 47.3.$$

Den optimale policy er dermed at gå $\underline{\text{A}\rightarrow\text{E og E}\rightarrow\text{A}}$.

Spørgsmål E.4: Hvad ville ændre sig, hvis vi i stedet benyttede en discount-factor på $\gamma = 0.1$?

Svar E.4: Hvis vi i stedet benyttede en discount-factor på 0.1 vil værdien af alle states være anderledes. Discount-faktoren bestemmer vægtningen mellem umiddelbar og langsigtet belønning, og med en lavere værdi af γ vil algoritmen i højere grad søge at opnå kortsigtet belønning, hvilket vil have indflydelse på hvad der ses som den optimale policy.