

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: EN NY BEHANDLINGSFORM	2
OPGAVE B: METALPRODUKTER	3
OPGAVE C: EN NEURON MED LEAKYReLU AKTIVERING	4
OPGAVE D: DE STØRSTE TAL	5
OPGAVE E: SYSTEM MED FEM TILSTANDE	6

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.

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?

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 fejlmargen på ± 1 procentpoint.

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

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.

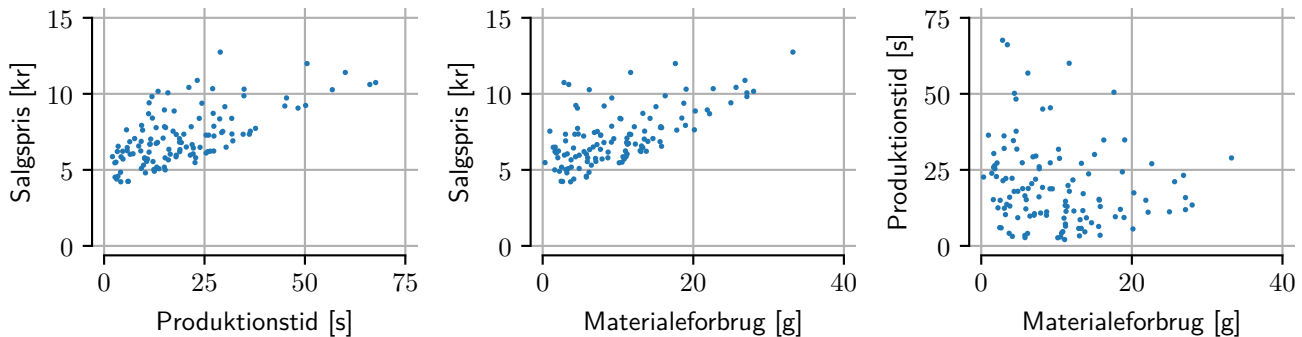
I forbindelse med forsøget bliver man opmærksom på, at behandlingens effektivitet muligvis påvirkes af hvorvidt patienterne er vaccineret for influenza. Forsøgets resultater er opsummeret i følgende tabel:

	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 vaccins indflydelse på behandlingen på baggrund af disse data?

Opgave B Metalprodukter

En virksomhed producerer 120 forskellige små metalprodukter (skrue, 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.

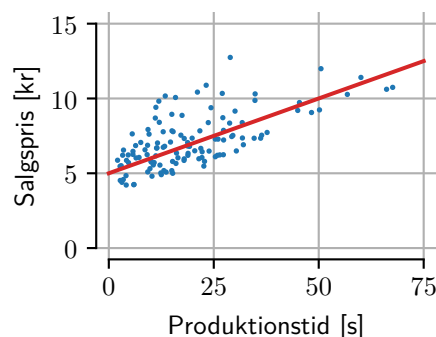


Virksomheden har fundet frem til følgende lineære regression, som nogenlunde beskriver sammenhængen:

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

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?

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 .

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?

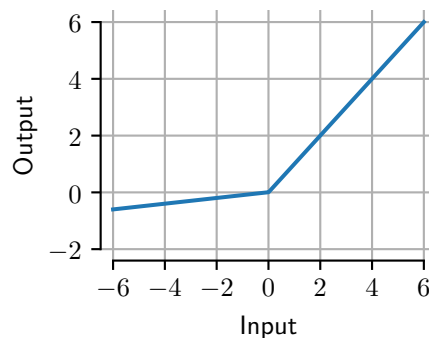
Spørgsmål B.4: Giv ud fra graferne et (groft) estimat af korrelations-koefficienten mellem Materialeforbrug og Produktionstid.

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_2 , og x_3 , og LeakyReLU aktivering kan defineres som

$$f(x_1, x_2, x_3) = \text{LeakyReLU}(w_0 + w_1x_1 + w_2x_2 + w_3x_3).$$

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 .

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

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?

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?

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 ($\text{num} > 1000$) 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?

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

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)
```

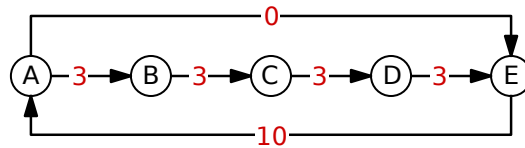
Algoritmen starter med at tage de første fem tal på listen som den gemmer i en ny liste kaldet `top_five`. Herefter gennemløber den de resterende tal på listen, og hver gang den finder et tal som er større end det mindste på listen `top_five`, erstatter den det mindste tal med det den har fundet. Når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

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

Opgave E System med fem tilstande

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



Pilene mellem tilstandene 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?

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?

Spørgsmål E.3: Hvad er den optimale policy?

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