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 Nedbør

Vi måler hver dag nedbørsmængden i en have. I løbet af de seneste 365 dage har det regnet 180 dage og været tørt de resterende 185 dage. Andelen af dage det har regnet er dermed $p = \frac{180}{365} \approx 49\%$. Den gennemsnitlige nedbørsmængde pr. dag er målt til $m_x = 1.9$ [mm], og variansen er beregnet til $s_x^2 = 9.5$ [mm^2].

Spørgsmål A.1: Beregn et 95% konfidensinterval for andelen af regnvejrsdage på et år og angiv grænserne for intervallet.

Svar A.1: Et 95% konfidensinterval for proportionen af regnvejrsdage kan beregnes som

$$p \pm 1.96 \cdot \sqrt{\frac{p(1-p)}{n}} = 0.49 \pm 1.96 \cdot \sqrt{\frac{0.49(1-0.49)}{365}} \approx 0.49\% \pm 0.05$$

Grænserne for intervallet er således [44%, 54%].

Spørgsmål A.2: Hvis vi i stedet havde talt antallet af regnvejrsdage i en 10-årig periode, hvordan ville bredden af konfidensintervallet da omtrentligt være i forhold til bredden af intervallet beregnet ovenfor?

Svar A.2: Med en større stikprøvestørrelse vil konfidensintervallet blive smallere. Da kvadratroden af stikprøvestørrelsen indgår i nævneren, vil intervallet omtrentligt være $\frac{1}{\sqrt{10}} \approx 0.32$ gange så bredt.

Ifølge Danmarks Meteorologiske Institut er den genemsnitlige nedbørsmængde pr. dag i Danmark præcis 2.1 [mm].

Spørgsmål A.3: Kan du på baggrund af målingerne fra haven konkludere at der falder mindre regn i haven end der i gennemsnit gør i Danmark?

Svar A.3: Et 95% konfidensinterval for nedbørsmængden i haven kan beregnes som

$$m_x \pm 1.96 \cdot \frac{s_x}{\sqrt{n}} = 1.9 \pm 1.96 \cdot \frac{\sqrt{9.5}}{\sqrt{365}} = 1.9 \pm 0.32$$

Intervallets grænser er således [1.6, 2.2], og den gennemsnitlige nedbørsmængde i Danmark ligger inden for dette interval. Vi kan ikke konkludere at nedbørsmængden i haven er lavere en gennemsnittet i Danmark.

Vi vil nu bestemme den gennemsnitlige nedbørsmængde pr. dag i en anden have et andet sted i landet, og vi ønsker et 95% konfidensinterval med en fejlmargin på ± 0.5 [mm]. Vi vil derfor måle nedbørsmængden på n tilfældigt udvalgte dage.

Spørgsmål A.4: Hvilken stikprøvestørrelse n vil du anbefale?

Svar A.4: Ved at løse for n i udtrykket for konfidensintervallet fås

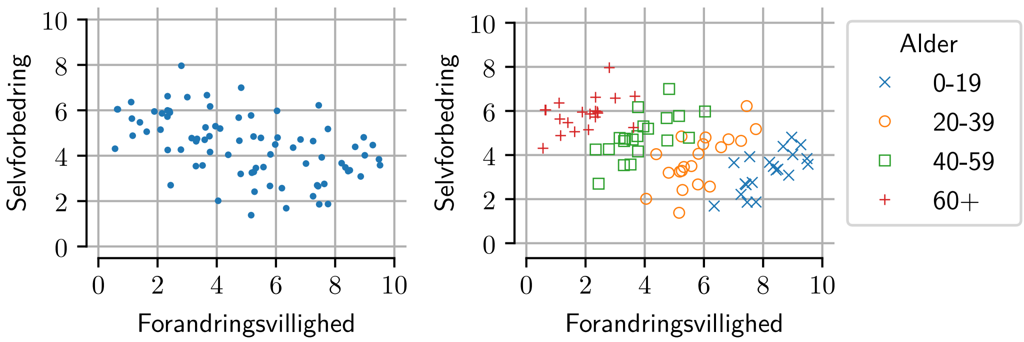
$$n = 1.96^2 \cdot \frac{s_x^2}{e^2}$$

Da vi ikke kender variansen af nedbørsmængden i den nye have, kan vi vælge at indsætte vores bedste gæt i form af variansen fra den gamle have. Vi får dermed en stikprøvestørrelse på

$$n = 1.96^2 \cdot \frac{9.5}{0.5^2} \approx= \underline{146 \text{ [dage]}}.$$

Opgave B Personlige værdier

Figurerne nedenfor viser to kvantitative målinger af de personlige værdier *forandringsvillighed* (F) og *selvforbedring* (S) som måles på en skala fra 0 til 10. Plottene viser målinger for 80 personer (taget fra en kunstig data-population). På plottet til højre er data yderligere inddelt i fire aldersgrupper.



Hvis vi kun ser på aldersgruppen 60+, så kan en lineær regressionsmodel bestemmes til

$$S = 4.6 + 0.6 \cdot F$$

Spørgsmål B.1: Hvad vil værdien af selvforbedring (S) være ifølge ovenstående model for en 60+ årig med en forandringsvillighed på $F = 6$?

Svar B.1: Selvforbedringen kan beregnes som

$$S = 4.6 + 0.6 \cdot F = 4.6 + 0.6 \cdot 6 = \underline{8.2}$$

Vi ønsker nu at forudsige selvforbedring ud fra forandringsvillighed i en samlet model for alle aldersgrupper ved hjælp af en lineær regression

$$S = w_0 + w_1 \cdot F$$

Spørgsmål B.2: Giv ud fra figuren til venstre et groft estimat af modellens parametre, w_0 og w_1.

Svar B.2: En lineær regressionslinje vil omtrentligt gå gennem punkterne $(F_0 = 0, S_0 = 6)$ og $(F_1 = 10, S_1 = 3)$. Ud fra disse punkter kan parametrene bestemmes til

$$w_0 \approx F_0 = \underline{6}, \qquad w_1 \approx \frac{S_1 - S_0}{F_1 - F_0} = \frac{3-6}{10-0} = \underline{-0.3}$$

(Svar i intervallet $4.5 \leq w_0 \leq 8$ og $-0.6 \leq w_1 \leq -0.2$ anses for korrekt).

Vi vil nu betragte en lineær regressionsmodel som benytter både forandingsvillighed (F) og aldersgrupe til at forudsige selvforbedring (S). I den forbindelse indfører vi fire binære variable A_{0-19}, A_{20-39}, A_{40-59}, og A_{60+}. Hver af disse variable er lig 1 for den aldersgruppe en person tilhører og ellers 0. For eksempel vil en person på 17 år have værdierne $A_{0-19} = 1$ og $A_{20-39} = A_{40-59} = A_{60+} = 0$. Vi opstiller nu en lineær regressionsligning

$$S = a \cdot F + b_1 \cdot A_{0-19} + b_2 \cdot A_{20-39} + b_3 \cdot A_{40-59} + b_4 \cdot A_{60+}$$

Spørgsmål B.3: Giv udfra figuren til højre et groft estimat af modellens hældningskoefficient a.

Svar B.3: Denne model har en hældningskoefficient der er ens for alle fire aldersgrupper, mens hver aldersgruppe har sit eget skæringspunkt. Derfor vil hældningskoefficienten omtrent være lig med den der er angivet i første delopgave for aldersgruppen 60+, nemlig $\underline{a = 0.6}$.

Vi kan også finde et estimat med udgangspunkt i en af aldersgrupperne, fx. 20–39. Her vurderes det visuelt at en regressionslinje omtrentligt vil gå gennem punkterne $(F_0 = 4, S_0 = 3)$ og $(F_1 = 8, S_1 = 5)$. Ud fra disse punkter kan parametrene bestemmes til

$$a = \frac{5-3}{8-4} \approx \underline{0.5}$$

(Svar i intervallet $0.2 \leq a \leq 1$ anses for korrekt).

Figuren til venstre viser at personer med relativt høj forandringsvillighed typisk har relativt lav selvforbedring.

Spørgsmål B.4: Hvad kan vi ud fra data sige om henholdvis korrelationen og den kausale sammenhæng mellem forandringsvillighed og selvforbedring?

Svar B.4: For den samlede population er der en negativ korrelation mellem F og S. Data siger i dette tilfælde kun noget om den statistiske sammenhæng, og vi kan ikke sige noget om den kausale sammenhæng. Vi bemærker også at der i hver af de viste aldersgrupper er der en positiv korrelation mellem F og S, hvilket underbygger at vi ikke kan sige noget om den kausale sammenhæng.

Opgave C Unikke værdier

Den følgende funktion tager som input en liste, `values`, og returnerer en liste med de unikke værdier i input-listen.

```
def find_unique(values):
    unique_values = []
    for x in values:
        if not x in unique_values:
            unique_values.append(x)
    return unique_values
```

Hver sammenligning og hver tilføjelse til en liste tæller som 1 operation, og længden af listen `values` er n. Det vil sige at tidskompleksiteten af operationen `x not in unique_values` er k, hvor k er længden af listen `unique_values`, og tidskompleksiteten af operationen `unique_values.append(x)` er 1.

Spørgsmål C.1: Hvad skal gælde for listen `values` for at opnå et best-case tilfælde, og hvad er best-case tidskompleksiteten $T(n)$ af funktionen målt i antal operationer?

Svar C.1: Best-case tilfældet opnås når alle værdier på listen er identiske, da listen `unique_values` således kun får længde $k = 1$.

Første værdi x sammenlignes med en tom liste (0 operationer) og tilføjes til listen af unikke værdier (1 operation). De resterende $n-1$ værdier sammenlignes med en liste med længde $k = 1$ (1 operation) og skal ikke tilføjes da de er identiske (0 operationer). Samlet set bliver best-case tidskompleksiteten således $T(n) = n$.

Spørgsmål C.2: Hvad skal gælde for listen `values` for at opnå et worst-case tilfælde, og hvad er worst-case tidskompleksiteten $T(n)$ af funktionen målt i antal operationer?

Svar C.2: Worst-case tilfældet opnås når alle værdier på listen er forskellige, da listen `unique_values` således bliver maksimal.

Første værdi x sammenlignes med en tom liste (0 operationer) og tilføjes til listen af unikke værdier (1 operation). Næste værdi sammenlignes med en liste med længde $k = 1$ (1 operation) og tilføjes til listen af unikke værdier (1 operation). Samlet set skal der således udføres $1 + 2 + 3 + \cdots + n = \frac{n(n+1)}{2}$ operationer, og worst-case tidskompleksiteten bliver således $T(n) = \frac{n(n+1)}{2}$.

Spørgsmål C.3: Hvad er worst-case kompleksiteten i store-O notation?

Svar C.3: Da største led i worst-case tidskompleksiteten er proportionalt med n^2 er kompleksiteten $O(n^2)$.

Den følgende funktion udfører samme opgave som `find_unique` men benytter en anden datastruktur til at gemme de unikke værdier.

```
def fast_find_unique(values):
    unique_set = set()
    for x in values:
        if not x in unique_set:
            unique_set.add(x)
    return unique_set
```

Denne datastruktur, kaldet `set`, har den egenskab at kompleksiteten af både operationen `x not in unique_set` og operationen `unique_set.add(x)` er $O(1)$.

Spørgsmål C.4: Hvad er worst-case kompleksiteten af funktionen `fast_find_unique` i store-O notation?

Svar C.4: Det antal operationer der skal udføres for hver værdi i listen `values` er nu af kompleksitet $O(1)$ for hvert af de n elementer, og den samlede kompleksitet bliver således $\underline{O(n)}$.

C

Opgave D Frit fald

Ifølge Newton's love er den tid t det tager et objekt at falde fra en højde h [m] givet ved formlen

$$t = \sqrt{\frac{2 \cdot h}{g}}$$

hvor g er gravitationskonstanten. I et forsøg har vi målt den tid det tager et objekt at falde fra $N = 11$ forskellige højder, $h \in \{0, 1, \dots, 10\}$ [m] (se figuren til venstre nedenfor). Vi ønsker nu at bestemme sammenhængen mellem højde og falde-tid (dvs. bestemme gravitationskonstanten g) ud fra en lineær regression.

Spørgsmål D.1: Vis hvorledes formlen for falde-tiden kan omskrives til standard-formen for en lineær regression, $t = w_0 + w_1 \cdot x$, og angiv udtrykkene for w_0, w_1 (udtrykt ud fra g) og x (udtrykt ud fra h).

Svar D.1: Vi kan for eksempel omskrive formlen som

$$t = \sqrt{\tfrac{2}{g}} \cdot \sqrt{h} = w_0 + w_1 \cdot x$$

og herfra kan vi se at

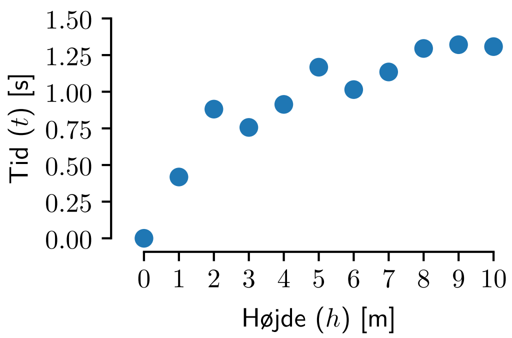
$$\underline{w_0 = 0, \quad w_1 = \sqrt{\frac{2}{g}}, \quad x = \sqrt{h}}$$

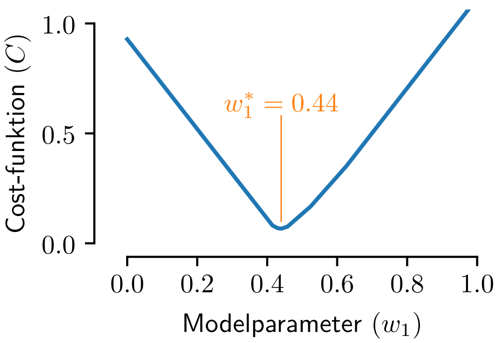
Ækvivalent kan vi gange x med en konstant c og dividere w_1 med samme konstant.

I det følgende benytter vi $w_0 = 0$ og ser på regressions-ligningen $t = w_1 \cdot x$ hvor $x = \sqrt{h}$. Vi ønsker at benytte absolut-værdien af forskellen mellem den målte og estimerede faldetid som loss-funktion—det vil sige at vi ønsker at minimere følgende cost

$$C(w_1) = \frac{1}{N} \sum_{i=1}^{N} |t_i - w_1 \cdot x_i|$$

Et plot af cost-funktionen, som har sit minimum i $w_1^* = 0.44$ ses i figuren nedenfor til højre.





Spørgsmål D.2: Hvad er ifølge den optimale model den estimerede værdi af gravitationskonstanten g?

Svar D.2: Vi har fundet udtrykket $w_1 = \sqrt{\frac{2}{g}}$, hvori vi kan isolere g og indsætte den optimale parameter-værdi $w_1^* = 0.44$:

$$g = \frac{2}{(w_1^*)^2} = \frac{2}{0.44^2} \approx \underline{10.3}$$

Spørgsmål D.3: Giv ud fra figuren et groft estimat af værdien af gradienten af cost-funktionen for parameterværdien $w_1 = 0.75$.

Svar D.3: Vi kan aflæse gradienten i figuren til højre som cost-funktionens hældning i punktet $w_1 = 0.75$. Når vi bevæger os omkring 0.5 enhed til højre på w_1-aksen svarer det til at vi går ca. 1 enhed op på C-aksen, og gradienten kan således estimeres til ca. $\underline{\nabla C \approx 2}$.

Det oplyses endvidere at summen af de observerede x-værdier er beregnet til $\sum_{i=1}^{N} x_i = 22.47$

Spørgsmål D.4: Beregn gradienten af cost-funktionen med hensyn til modellens parameter i punktet $w_1 = 0$ (som en tal-værdi). Hint: $\frac{d|z|}{dz} = \frac{z}{|z|}$.

Svar D.4: Gradienten er givet ved

$$\nabla C(w_1) = \frac{\partial C}{\partial w_1} = \frac{\partial}{\partial w_1} \frac{1}{N} \sum_{i=1}^{N} |t_i - w_1 \cdot x_i| = \frac{1}{N} \sum_{i=1}^{N} -\frac{t_i - w_1 \cdot x_i}{|t_i - w_1 \cdot x_i|} x_i$$

Vi indsætter nu $w_1 = 0$ og får

$$\nabla C(0) = \frac{1}{N} \sum_{i=1}^{N} -\frac{t_i}{|t_i|} x_i$$

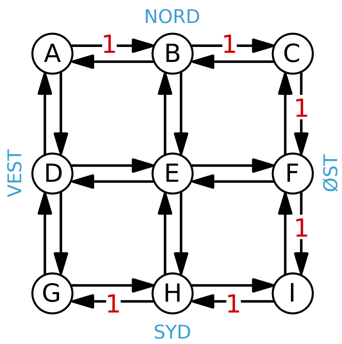
og da alle værdier t_i er positive kan vi skrive

$$\nabla C(0) = \frac{1}{N} \sum_{i=1}^{N} -x_i = \frac{1}{11} \cdot (-22.47) \approx \underline{-2.04}$$

D

Opgave E Gitterverden

Figuren nedenfor viser en gitterverden med 9 tilstande (A–I). Pilene i figuren viser de mulige transitioner mellem tilstandene, og et tal på pilen viser den tilhørende belønning (reward). Transitioner der er angivet med en pil uden et tal har en belønning på nul. Tabellen nedenfor viser kvaliteten (quality) for hver af de mulige transitioner fundet via tabulær q-læring med en discount-faktor på $\gamma = 0.9$.



	Nord	Syd	Øst	Vest
A	—	6.66	8.23	—
B	—	?	8.03	7.40
C	—	7.81	—	7.23
D	7.40	6.00	6.50	—
E	7.23	6.30	6.81	6.66
F	7.03	7.57	—	6.50
G	6.66	—	6.30	—
H	6.50	—	6.57	7.00
I	6.81	—	—	7.30

Spørgsmål E.1: Hvad er den optimale handling (action) hvis vi står i tilstand E?

Svar E.1: Den optimale handling kan aflæses som den handling med højest værdi i q-tabellen. For tilstand E er højeste værdi 7.23 for handlingen Nord.

I q-tabellen mangler værdien for $q(\mathrm{B}, \mathrm{Syd})$.

Spørgsmål E.2: Beregn $q(\mathrm{B}, \mathrm{Syd})$.

Svar E.2: At gå mod syd fra tilstand B leder til tilstand E. Vi kan derfor beregne

$$q(\mathrm{B}, \mathrm{Syd}) = R(\mathrm{B}, \mathrm{Syd}) + \gamma \cdot \max\left(q(\mathrm{E}, a)\right) = 0 + 0.9 \cdot \max(\{7.23, 6.30, 6.81, 6.66\}) = 0.9 \cdot 7.23 \approx \underline{6.51}$$

Spørgsmål E.3: Hvis vi kørte q-lærings-algoritmen med en ændret discount-faktor på $\gamma = 0.99$, hvad ville der så ske med med tallene i q-tabellen? Vil de blive større, mindre, eller forblive uændret?

Svar E.3: Tallene i q-tabellen betegner den discountede kumulative belønning. Hvis γ forøges til 0.99 vil fremtidig belønning vægte højere og den kumulative belønning vil blive større.

Spørgsmål E.4: Hvis vi i stedet for q-læring havde benyttet *value iteration* til at løse problemet, hvad ville værdien af tilstand A, $v(\mathrm{A})$ være?

Svar E.4: Tallene i q-tabellen er, ligesom tilstands-værdierne i value iteration, et udtryk for den discountede kumulative reward. Derfor kan værdien af tilstand A aflæses som den højeste q-værdi for tilstand A

$$v(\mathrm{A}) = \max(\{6.66, 8.32\}) = \underline{8.23}$$

At dette er tilfældet kan ses ud fra definitionen af quality:

$$q(s, a) = r(s, a) + \gamma \cdot \max_{a'} q(s', a').$$

Hvis vi maksimerer over a på begge sider af lighedstegnet fås

$$\max_{a} q(s, a) = \max_{a} r(s, a) + \gamma \cdot \max_{a'} q(s', a')$$

og betegner vi $\max_a q(s, a)$ som $v(a)$ har vi

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

som er den rekursive definition af value.

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 Clairvoyance

En ven hævder at være synsk. For at teste hans evner, foreslår du at lave et videnskabeligt forsøg hvor du tænker på en af jeres fem venner: Anna, Bettina, Christian, Dennis, eller Elinor. Din ven skal gætte hvem du tænker på, hvorefter du kan afsløre om det er korrekt, og forsøget gentages n gange.

Din ven påstår at han i hvert fald kan gætte 30% korrekt. Hvis man gætter tilfældigt vil man kun gætte 20% korrekt i gennemsnit. Efter at have udført forsøget vil I beregne et 95% konfidensinterval for proportionen af korrekte gæt, og hvis intervallet ikke indeholder 20% vil I konkludere at han er synsk.

Spørgsmål A.1: Beregn et estimat af den nødvendige stikprøve-størrelse n.

Svar A.1: Vi benytter formlen for beregning af stikprøve-størrelse

$$n = 1.96^2 \frac{p(1-p)}{e^2}$$

Her anvender vi den synske vens påståede proportion af korrekte gæt, $p = 0.3$, og da vi vil sikre at samplestørrelsen er stor nok til at kunne udelukke 20% skal bredden af intervallet være højst 10 procentpoint, $e = 0.1$. Det giver os følgende stikprøve-størrelse

$$n = 1.96^2 \frac{0.3(1-0.3)}{0.1^2} \approx \underline{81}$$

I ender med at benytte en stikprøvestørrelse på $n = 100$, og din ven gætter rigtigt i $n_x = 29$ tilfælde.

Spørgsmål A.2: Beregn et 95% konfidensinterval for proportionen, og angiv svaret som $0.29 \pm e$ hvor e er bredden af intervallet.

Svar A.2: Vi benytter formlen for et 95% konfidensinterval for en proportion og beregner

$$e = 1.96\sqrt{\frac{p(1-p)}{n}} = 1.96\sqrt{\frac{0.29(1-0.29)}{100}} \approx 0.089,$$

hvilket giver os intervallet $\underline{0.29 \pm 0.089}$.

Spørgsmål A.3: Hvis din ven havde gættet korrekt i færre end $n_x = 29$ tilfælde, ville det beregnede konfidensinterval da blive bredere, smallere, eller forblive uændret?

Svar A.3: Konfidensintervallet for en proportion er breddest for $p = 0.5$. Det kan vi huske fra formlen for beregning af sample størrelse, hvor $p = 0.5$ kan benyttes som et worst-case estimat. Det kan også ses fra formlen for bredden af intervallet, eller ved at indsætte et taleksempel, fx. $p = 0.28$ hvilket giver en interval-bredde på

$$1.96\sqrt{\frac{0.28(1-0.28)}{100}} \approx 0.088,$$

hvilket er lidt mindre end de 0.089 vi havde før. Med andre ord, hvis proportionen falder til mindre end 0.29, vil konfidensintervallet blive smallere.

Det beregnede interval indeholder ikke 20%, så derfor erklærer din ven, at det nu er endegyldigt bevist, at han er synsk. Men du mener dog at en så ekstra-ordinær påstand kræver ekstra-ordinær bevisførelse.

Spørgsmål A.4: Angiv mindst to måder hvorpå forsøgs-metoden kunne ændres, således at vi med større statistisk sikkerhed vil kunne fastslå om din ven er synsk.

Svar A.4: Da vi har benyttet et 95% konfidensinterval, vil den sande proportion kun i 95% af de gange vi benytter proceduren ligge inden for intervallet. Desuden er det beregnede interval ganske bredt, på grund af den forholdsvis lille sample størrelse. For at få større sikkerhed for konklusionen kan vi benytte et højere konfidens-niveau og/eller en større stikprøve-størrelse.

A

Opgave B Afstand mellem to sæt af tal

Et *talsæt* er en liste af tal, hvor rækkefølgen ikke har nogen betydning. En måde at måle hvor "ens" to talsæt X og Y er, kan være at beregne summen af den kvadratiske afstand fra hvert tal i sættet X til det nærmeste tal i sættet Y. Matematisk kan det skrives som

$$d(X, Y) = \sum_i \min_j \left((X_i - Y_j)^2 \right),$$

hvilket kan implementeres i Python med følgende funktion:

```python
def set_distance(X, Y):
    d = 0
    for x in X:
        d += min([(x-y)**2 for y in Y])
    return d
```

Spørgsmål B.1: Hvis de to sæt er givet ved

$$X = \{5, 8, 6, 9, 6\}, \quad Y = \{3, 5, 5, 4, 7\}$$

hvad er da afstanden $d(X, Y)$?

Svar B.1: Afstanden kan beregnes ved at finde det nærmeste tal i sættet Y for hvert tal i X og summe de kvadrerede forskelle:

$$d(X, Y) = (5-5)^2 + (8-7)^2 + (6-5)^2 + (9-7)^2 + (6-5)^2 = \underline{7}$$

Vi antager i det følgende at sættene X og Y begge indeholder n tal, og at n kan antage en vilkårlig værdi.

Spørgsmål B.2: Hvis vi tæller anvendelsen af plus (`+=`), minus (`-`) eller opløftet i anden potens (`**2`) som én operation hver, hvad er da tidskompleksiteten, $T(n)$, af algoritmen i Python-koden?

Svar B.2: For hvert tal i X udføres 1 plus-operation når tallene lægges sammen i variablen `d`, dvs. n operationer. Derudover udføres én minus-operation samt én potens-operation for hvert par af tal i X og Y, dvs. $2n^2$ operationer. Samlet får vi $\underline{T(n) = 2n^2 + n}$.

Spørgsmål B.3: Hvad er køretidskompleksiteten af Python-algoritmen i store-O notation?

Svar B.3: Da samtlige tal i de to sæt af størrelse n skal sammenlignes parvis, er kompleksiteten $\underline{O(n) = n^2}$.

En kollega foreslår at omskrive Python-koden ved hjælp af en list-comprehension, og når frem til følgende funktion, som er mere kort og kompakt, men måske også lidt sværere at læse og forstå:

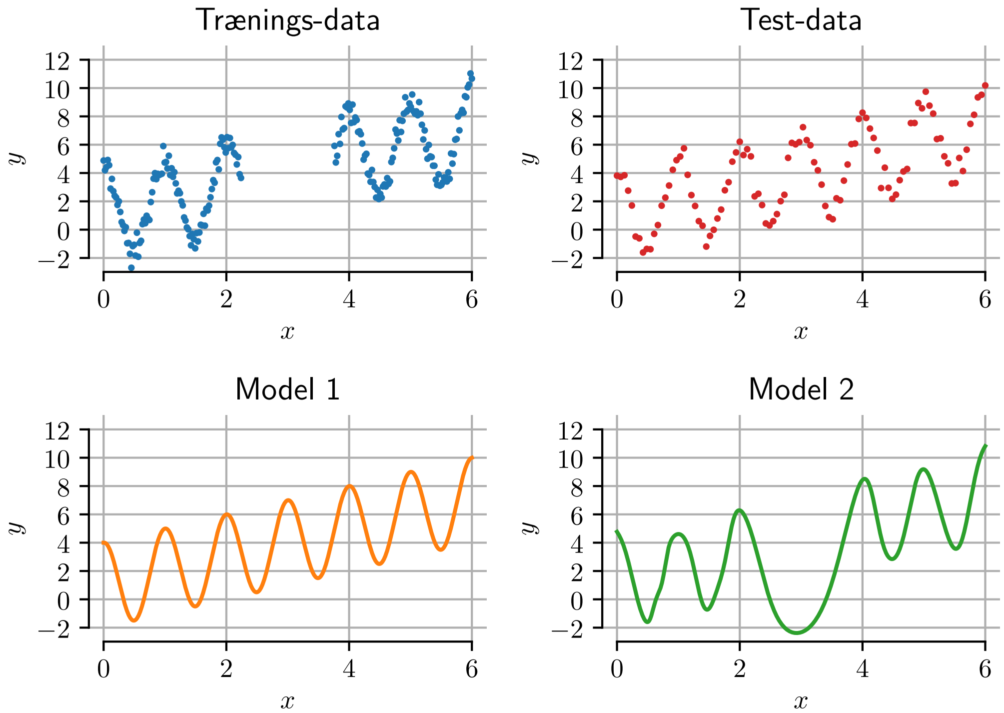
```python
def set_distance_new(X, Y):
    return sum([min([(x-y)**2 for y in Y]) for x in X])
```

Spørgsmål B.4: Sammenlignet med funktionen `set_distance`, er køretidskompleksiteten $O(n)$ af den nye funktion `set_distance_new` da større, mindre, eller uændret?

Svar B.4: Hvorvidt funktionen implementeres med en for-løkke eller en list-comprehension har ingen indflydelse på den asymptotiske køretidskompleksitet, som er <u>uændret</u>.

Opgave C Regression med en periodisk og stigende funktion

Vi ønsker at finde en model $f(x)$ der kan forudsige værdien af y for et givet punkt x. I plottet øverst til venstre ses trænings-data som vi vil benytte til at fitte modellen. Plottet øverst til højre viser test data som vi efterfølgende vil benytte til at estimere modellens generaliseringsfejl.



Da data ser ud til at være vel beskrevet ved et bias, en lineær komponent, samt en cosinus-komponent med perioden 2π, vil vi fitte følgende funktion, som vi kalder *Model 1*:

$$f(x) = w_0 + w_1 \cdot x + w_2 \cdot \cos(2\pi x)$$

Spørgsmål C.1: Hvilken konkret metode kan passende benyttes til effektivt at estimere parametrene w_0, w_1 og w_2?

Svar C.1: Da Model 1 er lineær i parametrene, kan vi passende benytte en ordinær lineær regression. Vi kan anvende den kvadratiske fejl som cost-funktion og finde parametrene ved den gængse metode med at sætte den afledede af cost-funktionen lig nul og løse de tre ligninger med tre ubekendte der kan opstilles.

Et fit af Model 1 er vist i plottet nederst til venstre, og det oplyses at amplituden af cosinus-komponenten er fundet til $w_2 = 3$.

Spørgsmål C.2: Giv ud fra figuren et estimat af parametren w_0.

Svar C.2: I punktet $x = 0$ aflæser vi $f(0) = 4$. Vi har dermed

$$f(0) = w_0 + w_1 \cdot 0 + w_2 \cdot \cos(2\pi 0) = w_0 + w_2 = 4 \Rightarrow w_0 = 4 - 3 = \underline{1}.$$

Som et alternativ har vi også fittet et neuralt netværk bestående af ét skjult lag med 50 neuroner og en hyperbolsk tangens aktiveringsfunktion, defineret med følgende Pytorch kode:

```
model2 = torch.nn.Sequential(
    torch.nn.Linear(1, 50),
    torch.nn.Tanh(),
    torch.nn.Linear(50, 1))
```

Denne model, som vi kalder *Model 2*, er fittet til data og vist i plottet nederst til højre. Model 2 er væsentligt mere fleksibel end Model 1, da den har mange flere parametre.

Spørgsmål C.3: Hvor mange parametre har Model 2?

Svar C.3: Det er oplyst at Model 2 er et neuralt netværk med ét skjult lag med 50 neuroner. Da input er en skalar, har hver neuron 2 parametre (1 vægt og 1 bias). I output-laget kombineres de 50 neuroner i en output-neuron med 50 input. Den har dermed 51 parametre (50 vægte og 1 bias) kombineres. Det samlede antal parametre er dermed

$$(1+1)\cdot 50 + (50+1) = \underline{151}$$

Spørgsmål C.4: Hvilken model har den laveste generaliseringsfejl (dvs. fejl på test-data)?

Svar C.4: Modellerne ser begge ud til at fitte trænings-data ganske fint, men de er meget forskellige i intervallet omkring $x = 3$, hvor der ikke er trænings-data til rådighed. I dette interval ses det at Model 1 stemmer godt overens med test-data hvorimod Model 2 ikke er så god. Det kan derfor konkluderes at Model 1 har den laveste generaliseringsfejl.

Opgave D Programmerings-prøve

Vi forestiller os en undersøgelse foretaget blandt studerende på et stort universitet: I alt 100 tilfældigt udvalgte studerende medbringer deres egen computer og tager en svær prøve i programmering. Det noteres hvilken computer-type de har medbragt og om de består prøven. Resultatet er følgende:

	Ikke bestået	Bestået	Total
Mac	40	10	50
Windows	24	24	48
Linux	0	2	2
Total	64	36	100

Resultatet viser at $\frac{10}{50} = 20\%$ af Mac-brugerne, $\frac{24}{48} = 50\%$ af Windows-brugerne og $\frac{2}{2} = 100\%$ af Linux-brugerne bestod prøven.

Spørgsmål D.1: Selv om undersøgelsen viser, at 100% af Linux-brugerne bestod, kan vi *ikke konkludere* at alle ville bestå, hvis universitetet pålagde alle at benytte Linux. Giv *to* argumenter for hvorfor vi ikke kan konkludere dette.

Svar D.1: Med kun to Linux-brugere i undersøgelsen har vi en for lille sample størrelse til at vi kan sige noget klart om beståelses-procenten blandt alle Linux-brugere i populationen.

Derudover er der ikke påvist en kausal sammenhæng mellem brug af Linux og beståelse af prøven.

Spørgsmål D.2: Er der statistisk belæg for at konkludere, at mindre end 50% af Mac-brugere blandt hele universitetes population vil kunne bestå prøven?

Svar D.2: Vi kan beregne 95% konfidens-intervallet for proportionen som

$$p_{\text{Mac}} = 0.2 \pm 1.96\sqrt{\tfrac{0.2\cdot(1-0.2)}{48}} \approx 0.2 \pm 0.11$$

Da konfidensintervallet ikke indeholder 50% er det klart, at der er statistisk belæg for konklusionen.

Universitetes administration laver en ny undersøgelse hvor 1000 studerende bliver bedt om at tage prøven på en computer som stilles til rådighed af universitetet. Halvdelen af de studerende udvælges tilfældigt til at tage prøven med en Mac, og den anden halvdel med en Windows computer. Resultatet af den nye undersøgelse er følgende:

	Ikke bestået	Bestået	Total
Mac	324	176	500
Windows	329	171	500
Total	650	350	1000

Spørgsmål D.3: Hvad kan du, ud fra de to undersøgelser, konkludere om den kausale sammenhæng mellem computer type og beståelse af prøven?

Svar D.3: I undersøgelsen hvor computer-typen tildeles tilfældigt er beståelsesprocenten i de to grupper næsten ens. Der kan ikke påvises nogen statistisk signifikant forskel, og sample størrelsen $n = 1000$ er forholdsvis stor. Det konkluderes, at der er ingen signifikant kausal sammenhæng mellem computer type og beståelse af prøven.

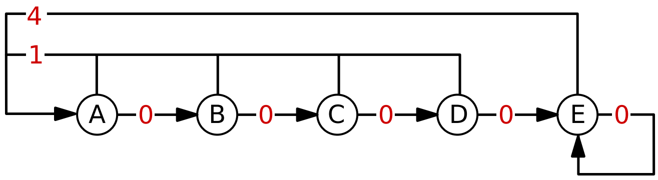
Spørgsmål D.4: Giv et eksempel på en mulig confounder, som kan forklare forskellen mellem de to undersøgelser.

Svar D.4: En confounder vil være en ikke-observeret variabel, som både er medvirkende årsag til de studerendes valg af computer-type og deres chance for at bestå. Det kunne fx. være at studerende som på forhånd har større erfaring med programmering er mere tilbøjelige til at købe en Windows computer. En mulig confounder kunne således være forudgående programmeringserfaring.

D

Opgave E Reinforcement learning

Vi betragter følgende miljø som indeholder fem tilstande (states), A–E. I hver tilstand er der to mulige actions: at gå `op` eller til `højre`. Pilene angiver de mulige transitioner mellem tilstandene, og tallene på pilene angiver de tilhørende belønninger (rewards). At gå `op` fører altid til tilsgand A.



At gå `op` fra tilstand A–D giver en belønning på 1, og at gå `op` fra tilstand E giver en belønning på 4. Alle andre belønninger er nul.

Spørgsmål E.1: Hvad er den optimale action, når man står i tilstand E?

Svar E.1: Hvis man går til `højre` i tilstand E får man en reward på nul og ender tilbage i tilstand E. Dermed vil den kumulative discountede reward være nul, uanset hvilken discount faktor der benyttes. Hvis man derimod går `op` får man en reward på 4, og ender i tilstand A, hvorfra man kan opnå yderligere positiv reward. Da der ikke er nogen negativ reward i miljøet, er det klart at der er større kumulativ discounted reward forbundet med at gå `op`. Dermed må den optimale action fra tilstand E være <u>`op`</u>.

I de følgende tre delopgaver anvender vi en discount factor på $\gamma = 0.9$. Det oplyses at værdien af tilstandene A og E er henholdsvis $v(A) = 10$ og $v(E) = 13$. Det oplyses endvidere at den optimale action, når man står i tilstand B, er at gå `op`.

Spørgsmål E.2: Hvad er værdien af tilstand B, $v(B)$?

Svar E.2: Da vi kender den optimale action, kan værdien af tilstanden kan beregnes som den kumulative discountede reward:

$$v(B) = r(B, \texttt{op}) + \gamma \cdot v(A) = 1 + 0.9 \cdot 10 = 1 + 9 = \underline{10}.$$

Spørgsmål E.3: Hvad er værdien af tilstand D, $v(D)$?

Svar E.3: Værdien kan beregnes ud fra formlen

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

Vi indsætter for de to mulige actions, `op` og `højre`, hvilket giver os

$$\begin{aligned} v(D) &= \max\Big(r(D, \texttt{op}) + \gamma \cdot v(A), \quad r(D, \texttt{højre}) + \gamma \cdot v(E)\Big) \\ &= \max\Big(1 + 0.9 \cdot 10, \quad 0 + 0.9 \cdot 13\Big) = \max(10, \quad 11.7) = \underline{11.7} \end{aligned}$$

Spørgsmål E.4: Beskriv den optimale policy, hvis man i stedet anvender en discount faktor på $\gamma = 0$.

Svar E.4: Ved $\gamma = 0$ vil den kumulative discountede reward kun vægte den reward man får i det næste skridt, og slet ikke vægte fremtidig reward. Dermed bliver den optimale policy at tage den action der giver højest umiddelbar reward. Af figuren kan det ses at denne optimale policy er at <u>altid gå `op`</u>.

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.

<u>Nej</u>

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

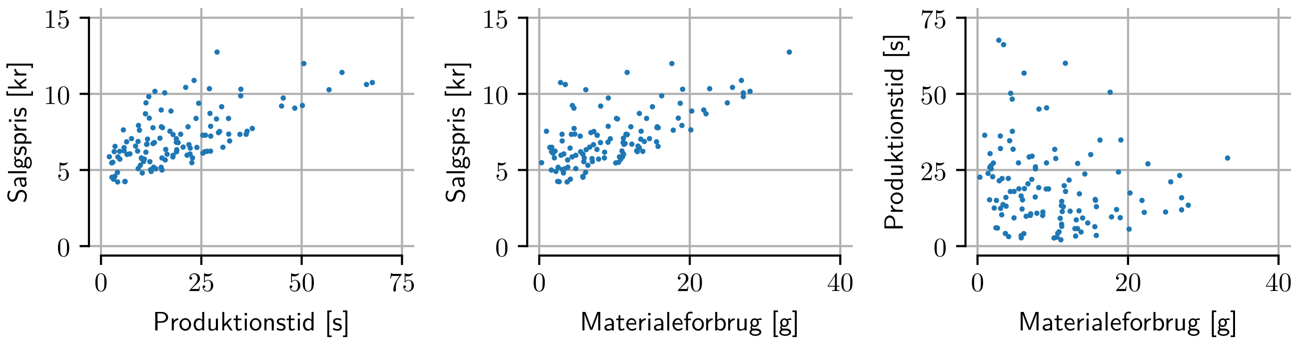
$$p = \frac{319}{400} \approx 0.798 p \pm 1.96 \cdot \sqrt{\frac{p(1-p)}{n}} = 0.798 \pm 1.96 \cdot \sqrt{\frac{0.798(1-0.798)}{400}} \approx 0.80\% \pm 0.039$$

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:

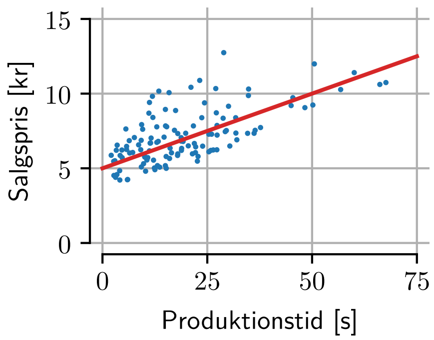
$$S = 3 + 0.2 \cdot M + 0.1 \cdot P$$

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 ikke vil være en passende tilgang.

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 korrelationskoefficienten er cirka nul.

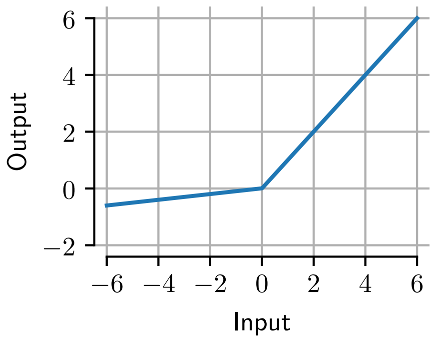
B

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 større n 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 <u>de fem største tal ligger først på listen</u>, således at `top_five` aldrig skal opdateres.

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

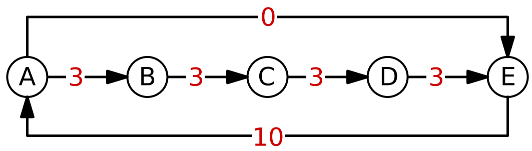
Spørgsmål D.4: Hvad er kompleksiteten af denne algoritme i store-O notation?

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 $\underline{O(n)}$.

D

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.