Logistisk regresjon (5 poeng for rett svar)

Vi har estimert en multinomisk logistisk regresjonsmodell med to trekk og tre klasser: c_1, c_2 og c_3. Parametere for denne modellen er som følges:

- Vektvektor $\mathbf{w_1}$ = [-2, 3] og konstantledd b_1 = 1
- Vektvektor $\mathbf{w_2}$ = [-1, 4] og konstantledd b_2 = -3
- Vektvektor $\mathbf{w_3}$ = [-1,-3] og konstantledd b_3 = 2

Gitt et nytt punkt med inputverdier [0, 2], hva er sannsynligheten (mellom 0 og 1) for klassen c_1 ?

Svar:

Vi starter med å beregne logits $z_i = \mathbf{w}_i\mathbf{x} + b_i$ for i = 1,2 og 3:

- $z_1 = -2 \cdot 0 + 3 \cdot 2 + 1 = 7$
- $z_2 = -1 \cdot 0 + 4 \cdot 2 - 3 = 5$
- $z_3 = -1 \cdot 0 + (-3) \cdot 2 + 2 = -4$

Softmax-sannsynligheten er deretter:

$$P(c_1) = \frac{e^{z_1}}{e^{z_1} + e^{z_2} + e^{z_3}} \approx \frac{1096.63}{1245.06} \approx 0.88$$

Logistisk regresjon 2 (2.5 poeng for rett svar)

La oss igjen ta en multinomisk logistisk regresjonsmodell med tre klasser c_1, c_2 og c_3. Etter å ha estimert modellens parametere anvender vi den på et nytt datapunkt **x**. For **x** predikerer modellen følgende sannsynlighetsfordeling:

$P(c_1|\mathbf{x}) = 0.88$

$P(c_2|\mathbf{x}) = 0.11$

$P(c_3|\mathbf{x}) = 0.01$

Det viser seg at den sanne klassen for punktet **x** er klasse c_2. Hva blir tapet, gitt at vi bruker kategorisk kryssentropi (*categorical cross-entropy*) som tapsfunksjon?

Oppgi svaret med to desimaler.

Svar:

Gitt et punkt $\mathbf{x}^{(k)}$ med fasitkategori $y^{(k)}$, og en sannsynlighetsfordeling $\hat{\mathbf{y}}^{(k)}$ prediktert av modellen er kategorisk kryssentropi definert slik:

$$l_{CCE}(\hat{\mathbf{y}}^{(k)}, y^{(k)}) = -\sum_{i=1}^{C} y_i^{(k)} log\ (\hat{y}_i^{(k)})$$

hvor $y_i^{(k)}$ er en binær koding av fasiten $y^{(k)}$, altså $y_i^{(k)} = \begin{cases} 1\ hvis\ \ y^{(k)} = i \\ 0\ ellers \end{cases}$.

Da kan tapet beregnes som $-(0 \cdot log\ (0.88)\ +\ 1\ \cdot\ log(0.11)\ +\ 0\ \cdot\ log(0.01))\ =\ 2.207$

Nevrale Nettverk – Antall Parametere (2 poeng for rett svar)

Dette nevrale nettverket jobbet du med i Oblig 2b. Hvor mange parametere har dette nettverket? Oppgi svaret som et heiltall. Anta at alle noder i de skjulte lagene og utdata-laget har én bias-inndata.

```
neural_network = NeuralNetwork(
    layer_sizes=[784, 32, 16, 10],
    activation_functions=[ReLU(), ReLU(), IdentityActivation()],
)
```

Svar:

Lagene her er fullt sammenkoblet, så mellom Lag A og Lag B er det (layer_sizes[A]*layer_sizes[B]) vekter. I tillegg kommer de layer_sizes[B] bias-inndataene. Merk at inn til det første laget har vi ikke noen bias-inndata, siden dette laget ikke selv beregner noen vektet sum, men bare sender videre dataen som kommer inn til nettverket.

Totalt antall parametere er dermed:

(784*32) + 32 + (32*16) + 16 + (16*10) + 10 = 25 818

Nevrale Nettverk - Regneoppgave (maksimalt 4.5 poeng)

Vi har et lite flerlags nevralt nettverk med:

- To input-nevroner: x_1, x_2
- Ett skjult nevron h med ReLU-aktivering
- Ett ut-nevron med **ingen** aktiveringsfunksjon (identitetsfunksjon):

Anta at:

- $x_1 = 1,\ x_2 = -2$
- $w_{h1} = 2,\ w_{h2} = 1,\ b_h = 1$ (der w_{h1} indikerer vekten fra input-node 1 til den skjulte noden, b_h indikerer bias-inndata til den skjulte noden)
- $w_o = -3,\ b_o = 2$ (der w_o indikerer vekten fra den skjulte noden til ut-noden, b_o indikerer bias-inndata til ut-noden

Hva blir z_h, a_h og $\hat{y}$?

(hint: z indikerer verdien inn til en node, a indikerer aktiveringsverdien)

Svar:

z_h er verdien inn i det skjulte nevronet. Det er den vektede summen av inndata:

$$z_h = w_{h1} * x_1 + w_{h2} * x_2 + b_h = 2 * 1 + 1 * (-2) + 1 = 1$$

a_h er verdien ut av det skjulte nevronet. Siden aktiveringsfunksjonen er ReLU, og inndata er 1, blir utverdien også 1.

$\hat{y}$ vil være det samme som inndata til ut-nevronet, siden dette har identitetsfunksjonen som aktiveringsfunksjon. Vi får da:

$$\hat{y} = w_o * a_h + b_o = (-3) * 1 + 2 = -1$$

Beslutningstrær og entropi (maksimalt 5 poeng)

Du skal trene et beslutningstre for å klassifisere om en kunde vil kjøpe et produkt (målvariabelen Kjøp, med verdiene Ja og Nei). Du har følgende treningsdatasett:

Kunde	Alder	Inntekt	Kjøp
1	Ung	Lav	Nei
2	Ung	Høy	Nei
3	Middel	Lav	Ja
4	Middel	Høy	Ja
5	Eldre	Lav	Ja
6	Eldre	Høy	Nei

Du vurderer å gjøre den første splittingen i treet på én av følgende attributter:

1. Attributtet Alder (verdier: Ung, Middel, Eldre)
2. Attributtet Inntekt (verdier: Lav, Høy)

Entropien til en node med to klasser beregnes som:

$$H(S) = -p_{\text{Ja}} \log_2(p_{\text{Ja}}) - p_{\text{Nei}} \log_2(p_{\text{Nei}})$$

Du beregner **vektet gjennomsnittlig entropi** etter en splitt ved å ta entropien i hver barne-node og vekte den med andelen eksempler som havner i noden.

Velg ett eller flere alternativer:

a) Splitt på Alder gir lavere vektet entropi enn splitt på Inntekt (RIKTIG)
b) Splitt på Inntekt gir lavere vektet entropi enn splitt på Alder
c) Splitt på Alder og splitt på Inntekt gir lik vektet entropi
d) I treet med lavest vektet entropi, vil alder ligge nærmest roten. (RIKTIG)
e) I treet med lavest vektet entropi, vil inntekt ligge nærmest roten.

Utregning:

Med intuisjon om hva entropi måler og hvordan beslutningstrær bygges kan man gjette ganske sikkert her uten å regne. Vi ser at splitt på inntekt gir kun urene noder (både «ja» og «nei» på begge sider), mens splitt på alder gir 2 rene noder (ung=nei, middel=ja) og 1 uren (eldre = ja og nei). Splitt på alder gir derfor sannsynligvis lavest vektet entropi, og blir derfor splitten nærmest roten. Vi kan også regne det ut hvis vi vil være helt sikre:

1. Entropi hvis vi splitter på Alder

Verdier: Ung, Middel, Eldre.

Node: Alder = Ung

Kunder: 1, 2 (begge «nei»)

Da blir: $p_{\text{Ja}} = 0, p_{\text{Nei}} = 1.\ H(\text{Alder = Ung}) = -0\log_2(0) - 1\log_2(1) = 0$
(Entropien er 0 når alle eksemplene har samme klasse.)

Node: Alder = Middel

Kunder: 3, 4 (begge «ja»)

Da blir: $p_{\text{Ja}} = 1,\ p_{\text{Nei}} = 0.\ H(\text{Alder = Middel}) = -1\log_2(1) - 0\log_2(0) = 0$

Node: Alder = Eldre

Kunder: 5, 6 (én «ja», én «nei»)

Da blir: $p_{\text{Ja}} = \frac{1}{2}, p_{\text{Nei}} = \frac{1}{2}.\ H(\text{Alder = Eldre}) = -\frac{1}{2}\log_2\left(\frac{1}{2}\right) - \frac{1}{2}\log_2\left(\frac{1}{2}\right) = 1$

Vektet gjennomsnittlig entropi etter splitt på Alder

Hver alder-verdi har 2 eksempler, totalt 6 eksempler. Vekter: $\frac{2}{6}$ for hver node.

$$H_{\text{etter}}(\text{Alder}) = \frac{2}{6} \cdot H(\text{Ung}) + \frac{2}{6} \cdot H(\text{Middel}) + \frac{2}{6} \cdot H(\text{Eldre})$$
$$H_{\text{etter}}(\text{Alder}) = \frac{2}{6} \cdot 0 + \frac{2}{6} \cdot 0 + \frac{2}{6} \cdot 1 = \frac{2}{6} = \frac{1}{3} \approx \mathbf{0.33}$$

2. Entropi hvis vi splitter på inntekt

Verdier: Lav, Høy.

Node: Inntekt = Lav

Kunder: 1, 3, 5 (to «ja», én «nei»).

Da blir: $p_{\text{Ja}} = \frac{2}{3}, p_{\text{Nei}} = \frac{1}{3}.\ H(\text{Inntekt = Lav}) = -\frac{2}{3}\log_2\left(\frac{2}{3}\right) - \frac{1}{3}\log_2\left(\frac{1}{3}\right) \approx 0.918$

Node: Inntekt = Høy

Kunder: 2, 4, 6 (én «ja», to «nei»).

Da blir: $p_{\text{Ja}} = \frac{1}{3}, p_{\text{Nei}} = \frac{2}{3}.\ H(\text{Inntekt = Høy}) = -\frac{1}{3}\log_2\left(\frac{1}{3}\right) - \frac{2}{3}\log_2\left(\frac{2}{3}\right) \approx 0.918$

Vektet gjennomsnittlig entropi etter splitt på Inntekt

Hver node har 3 eksempler, totalt 6. Vektene er $\frac{3}{6} = \frac{1}{2}$.

$$H_{\text{etter}}(\text{Inntekt}) = \frac{1}{2} \cdot H(\text{Lav}) + \frac{1}{2} \cdot H(\text{Høy})$$
$$H_{\text{etter}}(\text{Inntekt}) = \frac{1}{2} \cdot 0.918 + \frac{1}{2} \cdot 0.918 = 0.918$$

3. Sammenligning

- Vektet entropi etter splitt på Alder: $H_{\text{etter}}(\text{Alder}) \approx 0.33$
- Vektet entropi etter splitt på Inntekt: $H_{\text{etter}}(\text{Inntekt}) \approx 0.918$

Lavest entropi (mest «rene» noder) får vi ved å splitte på Alder. Alder havner dermed også nærmest roten i treet.

Poenggiving: 2 poeng per rett svar, -1 poeng per feil svar, 1 bonuspoeng for alt rett

Ensembler – Uavhengighet (maksimalt 4 poeng)

Hvordan kan vi gjøre modellene i et ensemble mer uavhengige av hverandre? Nevn to ting vi kan gjøre for å øke uavhengigheten.

Svar:

Vi kan for eksempel:

- Trene hver modell på ulike deler av datasettet
- Bruke ulike typer algoritmer i ensemblet
- Gi tilfeldige underutvalg av trekk til hver splitt i trærne

Forsterkende Læring – Handlingsvalg (maksimalt 4 poeng)

En agent er i en tilstand s og har tre mulige handlinger a_1, a_2, a_3 med følgende Q-verdier:

- $Q(s, a_1) = 0$
- $Q(s, a_2) = 1$
- $Q(s, a_3) = 2$

Agenten bruker **softmax-handlingsvalg** (merk at softmax ofte har en temperatur-hyperparameter, den behøver dere ikke ta med her - vi kan anta den har verdi 1 og dermed ikke påvirker formelen).

Velg **alle** utsagn som er korrekte (avrund sannsynligheter til to desimaler).
A. Softmax vil gi høyere sannsynlighet for å velge a_1 enn det grådig handlingsvalg ville gjort (RIKTIG)
B. $P(a_1 \mid s) \approx 0.24$

C. $P(\,a_2 \mid s\,) \approx 0.50$
D. $P(\,a_3 \mid s\,) \approx 0.67$ (RIKTIG)
E. Softmax vil gi høyere sannsynlighet for å velge a_3 enn a_2 (RIKTIG)

Utregning:

Formelen for softmax-handlingsvalg (for valg av handling $\boldsymbol{a}$ i tilstand $\boldsymbol{s}$) er:

$$\boldsymbol{P(a|s) = \frac{e^{Q(s,a)}}{\sum_{k=1}^{n} e^{Q(s,a)}}}$$

Vi har:

- $e^0 = 1, e^1 \approx 2.72, e^2 \approx 7.39$
- Nevner: $1 + e + e^2 \approx 1 + 2.72 + 7.39 \approx 11.11$
- $P(a_1 \mid s) = \frac{1}{11.11} \approx 0.09$
- $P(a_2 \mid s) = \frac{e}{11.11} \approx \frac{2.72}{11.11} \approx 0.24$
- $P(a_3 \mid s) = \frac{e^2}{11.11} \approx \frac{7.39}{11.11} \approx 0.67$

Poenggiving: 1 per rette svar, -0.5 per feil svar. 1 ekstra poeng ved alt rett.

Generativ KI – Evaluering (maksimalt 4 poeng)

Hvorfor er evaluering av tekstgenerering vanskelig, sammenlignet med tradisjonell evaluering av ML-systemer? Gi en kort begrunnelse (1-2 setninger).

Gi et eksempel på hvordan evaluering allikevel kan gjennomføres i lys av denne utfordringen.

Løsningsforslag:

Evaluering av tekstgenerering er vanskelig fordi det vil finnes mange ulike måter å svare på et spørsmål som alle kan sies å besvare spørsmålet på en god måte. Dette står i motsetning til tradisjonell evaluering av ML-systemer det som regel kun finnes ett korrekt svar.

Evaluering av språkmodeller kan allikevel gjennomføres ved én av følgende metoder*:*

- omforming av generering til klassifikasjon der modellen vurderes ved sin evne til å generere svaralternativer i en flervalgsoppgave
- evaluering skjer ved automatiske mål som ser på overlapp i ord mellom et generert svar og et menneskelig produsert svar (for eksempel ROUGE)
- LLM-som-dommer oppsett: en annen språkmodell benyttes til å vurdere genererte svar mtp ulike kriterier som eksempelvis relevans, flyt, sannferdighet osv

Filosofi/framtid (maksimalt 5 poeng)

Hvilken sammenheng ser vi ofte mellom ytelse og forklarbarhet i KI-modeller? Nevn en algoritme som typisk resulterer i høy forklarbarhet og en som typisk resulterer i lav forklarbarhet. Begrunn svaret kort.

Løsningsforslag:

Vi ser ofte en invers sammenheng mellom ytelse og forklarbarhet, der de største og kraftigste KI-modellene også er de det er vanskeligst å forklare beslutningene til.

Algoritmer som typisk resulterer i lav forklarbarhet: Dype nevrale nettverk, der enormt mange parametere sammen bestemmer oppførselen men det er vanskelig å tolke akkurat hva nettverket har lært ved å kikke på disse parameterne.

Algoritmer som typisk resulterer i høy forklarbarhet: Beslutningstrær, lineær regresjon, logistisk regresjon. Enklere modeller med færre parametere, der det i større grad kan tolkes ut ifra parameterne (for lineær/logistisk regresjon) eller strukturen til den trente modellen (for beslutningstrær) hva som er lært.

Etikk (maksimalt 5poeng)

Hva menes med at teknologi kan ha "dobbel bruk" ("dual use")? Gi minst ett eksempel på dobbel bruk forbundet med KI.

Løsningsforslag:

Teknologi har dobbel bruk dersom den kan brukes til både fredelige og miltære formål, eller mer generelt til både uskadelige og skadelige formål. Når det gjelder KI vil et eksempel kunne være bruk av KI i forbindelse med autonome våpen, men også bruk av KI til desinformasjon eller overvåking. Her vil teknologi som i utgangspunktet er utviklet for ikke-skadelige formål (KI for autonome systemer eller tekstgenerering) kunne benyttes til til å forårsake skade (krigføring, propaganda).