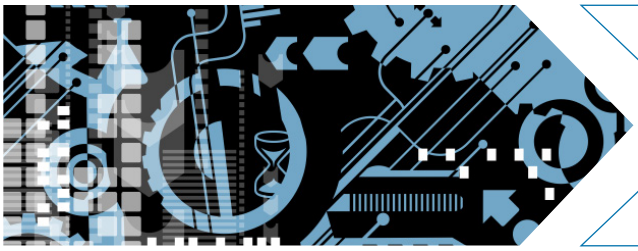


SPISS
Naturfaglige artikler av elever i videregående opplæring

# Aktiveringsfunksjonens påvirkning på nevrale nettverks læringsevne

*Forfatter: Mats Lunde, Vestby videregående skole*

*Aktiveringsfunksjonene i nevrale nettverk utgjør en viktig del av kunstig intelligens. En aktiveringsfunksjon er den funksjonen i et nevralt nettverk som leverer resultatet basert på datainputen. Denne studien sammenliknet læringsevnen til nevrale nettverk med ulike aktiveringsfunksjoner. De seks aktiveringsfunksjonene PReLU, ReLU, Sigmoid, Softplus, Tanh og den lineære funksjonen, f(x) = x ble testet på 600 nevrale nettverk. Nettverkene ble trent og testet på MNIST-datasettet som består av 80 000 bilder av håndskrevne tall. Resultatene viste at det var signifikante forskjeller mellom læringsevnen til nettverkene som brukte ulike aktiveringsfunksjoner. Nettverkene som brukte PReLU oppnådde høyest identifiseringsrate, og hadde lavest standardavvik.*

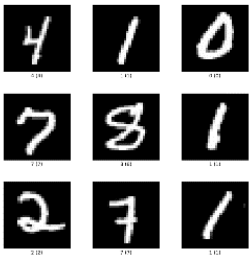
## Introduksjon

Bruk av nevrale nettverk blir stadig viktigere innenfor flere områder i samfunnet, og mer vanlige i vårt hverdagslige liv. Vi finner nevrale nettverk i applikasjoner som for eksempel smarttelefoner, biler, kunstig intelligens som chatboten Chat GPT og den virtuelle assistenten Siri. For at disse systemene skal være pålitelige, er man avhengig av høy presisjon i resultatene, noe som krever mye trening av nettverkene. En av de store utfordringene med treningen av nevrale nettverk, er at det krever enorme mengder data, tid og prosesseringskraft. Treningen kan effektiviseres ved bruk av optimale aktiveringsfunksjoner for de spesifikke nevrale nettverkene. Dette kan bidra til å forbedre læringshastigheten og øke identifiseringsraten til nettverket (Sharam, S., Sharam, S., Athaiya., 2020). I denne studien defineres identifiseringsrate som andelen bilder det nevrale nettverket klarte å identifisere. Målet med studien var å kartlegge identifiseringsratene og vurdere effekten aktiveringsfunksjonene har på nettverkenes læringsevne. Nullhypotesen var at aktiveringsfunksjoner ikke påvirker et nevralt nettverks læringsevne og -hastighet. Den alternative hypotesen var at en av de testede aktiveringsfunksjonene resulterte i en økt læringsevne og -hastighet, i forhold til de andre aktiveringsfunksjonene som ble testet.

## Metode

For å analysere hvordan aktiveringsfunksjoner påvirker nevrale nettverks læringsevne, ble det laget ett program i Python 3.10.7 som produserte alle de nevrale nettverkene i studien. De seks aktiveringsfunksjonene PReLU, ReLU, Sigmoid, Softplus, Tanh og den lineære funksjonen, f(x) = x ble testet på 600 nevrale nettverk (100 per aktiveringsfunksjon). MNIST-datasettet, bestående av 80 000 bilder av håndskrevne tall (se figur 1), ble brukt til å trene og teste læringsevnen til nettverkene. For å unngå «overfitting» ble kun 60 000 bilder brukt til trening av de nevrale nettverkene, og de resterende 20 000 bildene ble brukt til å teste identifiseringsraten.

For å kunne vurdere nettverkenes læringsevnen underveis i prosessen, ble identifiseringsraten testet syv ganger etter trening på hhv. 0, 96, 992, 9 984, 20 000, 40 000 og 60 000 bilder. Identifiseringsraten i prosent ble regnet ut ved å dele antall bilder som ble riktig identifisert på 20 000 og multiplisere med 100 %.



*Figur 1 viser 9 eksempler fra MNIST-datasettet*

En funksjon som genererte startvariabler (vekt- og biasmatriser) ble definert i programmet som produserte de nevrale nettverkene. Figur 2 viser et utdrag av koden til programmet i Python. Videre i beskrivelsen av koden er det henvist til linjenumrene i kolonnen til venstre i figuren. Vekt- og biasmatrisene ble generert (linje 11) og nettverkets identifiseringsrate ble testet før trening (linje 112). Deretter delte en for-løkke opp treningsmaterialet i «partier» med en størrelse på [784 * 32] og [10 * 32] (linje 113-117). Her representerer tallet 784 antall piksler i bildet, tallet 10 representerer antall mulige resultater og tallet 32 representerer antall bilder i hvert parti. For-løkken ble kjørt til nettverket hadde trent på alle de 60 000 bildene, altså 1875 ganger. Partistørrelsen ble valgt basert på resultatene fra Thakur (2022). I linje 119 blir datainput prosessert og resultatet av prosesseringen brukes videre. Resultatet brukes til å justere startvariablene i linje 120 og 121. Slik minimaliseres feil, og det nevrale nettverket lærer. Fra linje 122 og videre testes og lagres identifiseringsraten etter trening på 96, 992, 9 984, 20 000, 40 000 og 60 000 bilder. Intervallene er valgt med hensyn på antall bilder i hvert parti (32), og for å generere punkter til regresjonsanalysene av nettverkenes læringsevne (se figur 4).

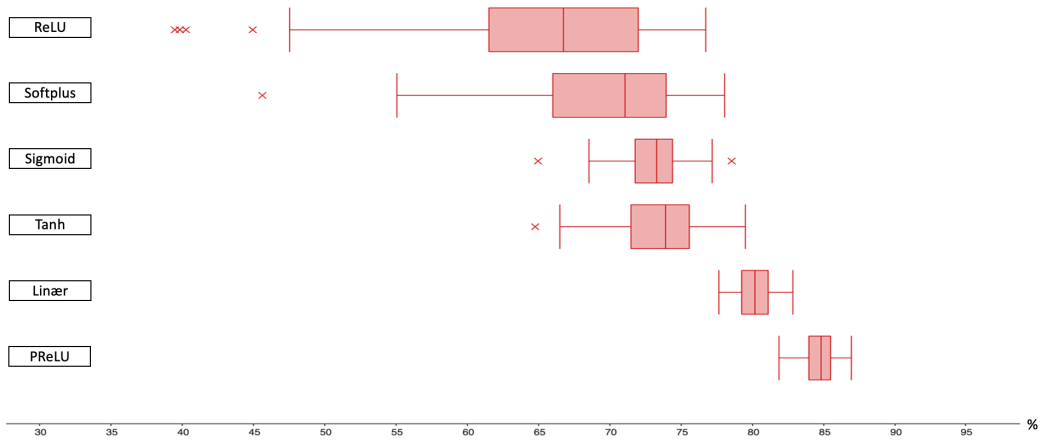
Ved bruk av GeoGebra ble den statistiske signifikansen til resultatene beregnet i en t-test. I denne studien settes signifikansnivået (p) til 0,01. Regresjonsanalysene og deriveringen av grafene ble også utført i GeoGebra.

```
110  def gradient_descent(X, Y, alpha, iterations):
111      W1, b1, W2, b2 = start_parametere() # 1
112      logged_data_0.append(round(test_accuracy(W1, b1, W2, b2),4)) # 2
113      for i in range(iterations): # 3
114          X = X.T # 4.1
115          X_use = X[(i*32):(i*32)+32] # 4.2
116          X = X.T # 4.3
117          X_use = X_use.T # 4.4
118          Y_use = Y[(i*32):((i*32)+32)] # 4.5
119          Z1, A1, Z2, A2 = forward_prop(W1, b1, W2, b2, X_use) # 5
120          dW1, db1, dW2, db2 = backward_prop(Z1, A1, Z2, A2, W1, W2, X_use, Y_use
121          W1, b1, W2, b2 = update_params(W1, b1, W2, b2, dW1, db1, dW2, db2, alph
122          if i * 32 == 96: # 8.1
123              logged_data_100.append(round(test_accuracy(W1, b1, W2, b2),4))
124          if i * 32 == 992:
125              logged_data_1000.append(round(test_accuracy(W1, b1, W2, b2),4))
126          if i * 32 == 4992:
127              logged_data_5000.append(round(test_accuracy(W1, b1, W2, b2),4))
128          if i * 32 == 9984:
129              logged_data_10000.append(round(test_accuracy(W1, b1, W2, b2),4))
130          if i * 32 == 20000:
131              logged_data_20000.append(round(test_accuracy(W1, b1, W2, b2),4))
```

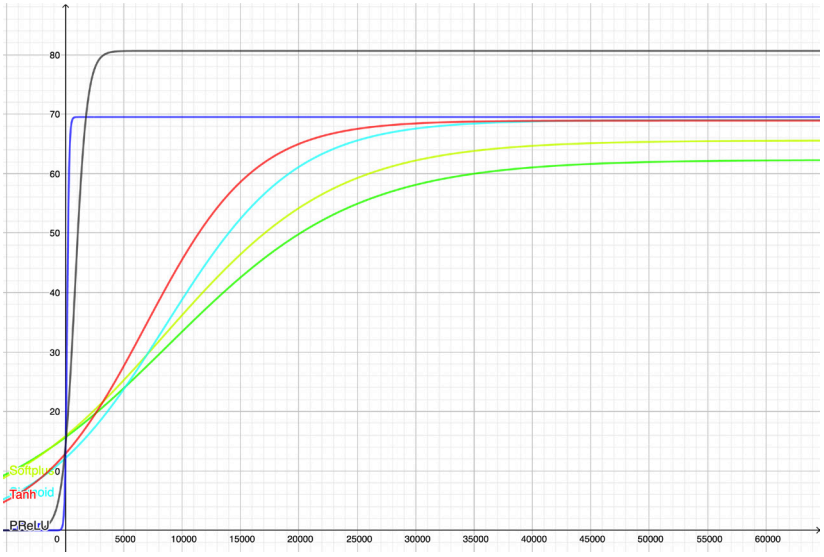
*Figur 2 viser et utsnitt av koden bak det nevrale nettverket. Kolonnen til venstre er linjenummer.*

## Resultater og Diskusjon

En høy identifiseringsrate er avgjørende for utvikling av pålitelige nevrale nettverk. Figur 3 viser identifiseringsraten til de nevrale nettverkene med ulike aktiveringsfunksjoner etter trening på 60 000 bilder. Gjennomsnittlig identifiseringsrate varier fra 65,7 % (ReLU) til 84,7 % (PReLU), mens standardavvikene varier fra 8,1 % (ReLU) til 1,1 % (PReLU). I figur 4 vises utviklingen av identifiseringsraten basert på antall bilder de nevrale nettverkene har trent på. Figuren viser at både læringsevnen og læringshastigheten varierer for de ulike aktiveringsfunksjonene.



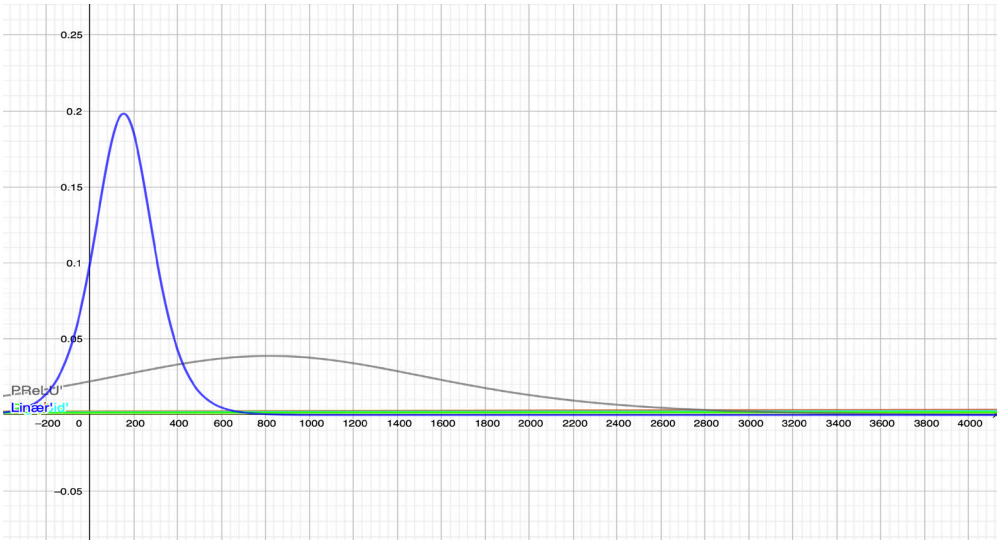

*Figur 3 viser identifiseringsratene til de forskjellige aktiveringsfunksjonene*



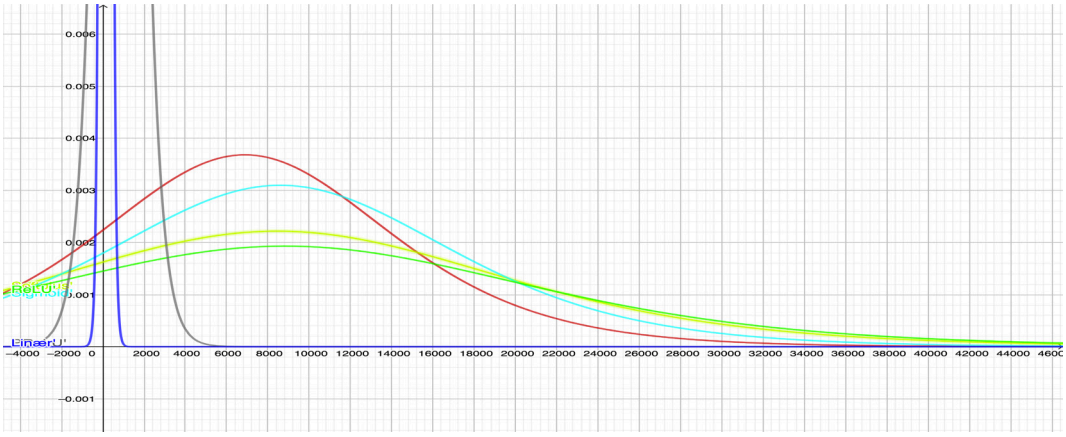

*Figur 4 viser regresjonsanalyser for den gjennomsnittlige utviklingen av identifiseringsraten med hensyn på antall bilder det nevrale nettverket har trent på. X-aksen og Y-aksen representerer henholdsvis antall bilder og identifiseringsrate. Hver funksjon representerer en aktiveringsfunksjon: PReLU (mørk grå), Linær (mørk blå), Tanh (rød), Sigmoid (lys blå), Softplus (gul) og ReLU (grønn).*

PReLU var den aktiveringsfunksjonen som oppnådde høyest identifiseringsrate (87,0 %), og som fikk den høyeste gjennomsnittlige identifiseringsraten (84,7 %). Dette forsterker den alternative hypotesen. Den lineære funksjonen hadde nest høyest identifiseringsrate (82,9 %) med et gjennomsnitt på 81,8 %. En T-test med nullhypotese (PReLU - Linær = 0) og alternativ hypotese (PReLU - Linær > 0) resulterte i p-verdien 0.00, en signifikant forskjell mellom de to beste aktiveringsfunksjonene. Siden PReLU er signifikant forskjellig fra den lineære funksjonen, kan man anta at det også er signifikante forskjeller mellom PReLU og de fire andre aktiveringsfunksjonene.

Et fellestrekk mellom PReLU og den lineære funksjonen, som skiller disse to aktiveringsfunksjonene fra de fire andre, er at læringshastigheten i starten er mye høyere. Figur 5 viser at den lineære funksjonen har høyest læringshastighet i starten, før den raskt avtar etter 200 bilder og går mot null. PReLU har en noe mer stabil læringshastighet sammenlignet med den lineære. Hastigheten avtar etter 800 bilder, og ligger rundt 0 ved 6 000 bilder. De fire andre aktiveringsfunksjonene (Softplus, Sigmoid, Tanh og ReLU) har mer stabile læringshastigheter (figur 6), men læringshastighetene ligger rundt 0,003 på det høyeste i motsetning til PReLU og den lineære som ligger rundt hhv 0,04 og 0,2. Den mer stabile, men tregere læringshastigheten, kan tyde på at disse funksjonene krever mer treningsmateriale. Det betyr ikke nødvendigvis at de ikke kan oppnå like høye identifiseringsrater.




*Figur 5 Figuren viser den deriverte av funksjonene fra figur 4. X-aksen og Y-aksen representerer henholdsvis antall bilder og læringshastighet. Merk at X-aksen går fra 0 - 4 000 og ikke inkluderer alle 60 000 bildene. Hver funksjon representerer en aktiveringsfunksjon: PReLU (mørk grå), Linær (mørk blå), Tanh (rød), Sigmoid (lys blå), Softplus (gul) og ReLU (grønn).*




*Figur 6 Figuren viser den deriverte av funksjonene fra figur 4. X-aksen og Y-aksen representerer henholdsvis antall bilder og læringshastighet. Merk at X-aksen går fra 0 - 46 000 og ikke inkluderer alle 60 000 bilder. Hver funksjon representerer en aktiveringsfunksjon: PReLU (mørk grå), Linær (mørk blå), Tanh (rød), Sigmoid (lys blå), Softplus (gul) og ReLU (grønn).*

Softplus, Sigmoid, Tanh og ReLU har størst spredning i identifiseringsrate etter trening på 60 000 bilder (figur 3). Sammenlignet med PReLU og den lineære som hadde et standardavvik på hhv. 1,1 % og 1,3 % hadde Softplus (6,1 %), Sigmoid (2,1 %), Tanh (2,9 %) og ReLU (8,1 %) vesentlig høyere standardavvik. En mulig årsak til at Softplus, Sigmoid og Tanh hadde større standardavvik enn PReLU og den lineære, er at de er eksponentielle funksjoner. De eksponentielle aktiveringsfunksjonene er sensitive rundt x = 0 fordi det er en stor forskjell mellom x og f(x). Dette kan medføre at små forandringer i datainput fører til endringer i resultatet. ReLU, som også hadde større standardavvik enn PReLU og den lineære funksjonen, er en delt funksjon som består av to lineære funksjoner. De to lineære funksjonene ($f(x) = x$ for $x > 0$ og $f(x) = 0$ for $x < 0$) gjør at også denne aktiveringsfunksjonen er sensitiv rundt x = 0. PReLU og den lineære er ikke like sensitive rundt x = 0 eller for andre x-verdier, som kan være årsaken til at de er mer presise.

Hvordan aktiveringsfunksjonene oppfører seg i området rundt 0 [-1, 1] er spesielt viktig i denne studien fordi dataene ble normalisert slik at de falt innenfor området [0, 1]. Dataene ble normalisert fordi det er gunstig for nettverkets læringsevne og prosesseringstid (Ahsan, M.M., Mahmud, M. A. P., Saha, P. K., Gupta, K. D., Siddique, Z., 2021) (Maheshkar. S, 2022). Vektene og biasene hadde verdier i området [-0.5, 0.5], noe som gjør at verdiene det første laget med nevroner mottar er i området [-1, 1]. At verdiene havner i området [-1, 1] gjør at aktiveringsfunksjoner der $f(x) < 1$ og $f(x) > -1$ for alle x-verdier i området [-1, 1] aldri vil kunne få verdier utenfor området [-1, 1]. Eksponentielle funksjoner vil i mange tilfeller ikke følge $f(x) < 1$ og $f(x) > -1$ for alle x-verdier i området [-1, 1]. Dette gjør at nettverket kan ende med relativt høye eller lave verdier, som igjen gjør nettverket sensitivt til små forandringer i datainput (Chieng, H. H., 2021). At nettverket er sensitivt til små forandringer i datainput kan føre til stor spredning i identifiseringsratene. Dette kan være årsaken til at resultatene viser relativt stor spredning i identifiseringsratene til de nevrale nettverkene som benytter eksponentielle funksjoner.

Denne studien inkluderer bare et begrenset antall aktiveringsfunksjoner, og er kun testet mot et datasett. MNIST-datasettet er et relativt enkelt datasett å prosessere, siden det kun består av bilder av tall fra 0 til 9. Dette gjør at ren gjetting over lengre tid vil gi en identifiseringsrate som går mot 10 %. Aktiviseringsfunksjonene brukt i studien kan derfor gi andre resultater når de brukes på mer komplekse datasett.

Det er to hoved usikkerhetsmomenter i studien, regresjonsanalysen og de nevrale nettverkene. Regresjonsanalysen baserer seg kun på syv punkter. For få punkter kan føre til at regresjonsanalysen ikke gir et representativt innblikk i dataene. Grunnen til at det kun er benyttet syv punkter er at Geogebra ikke klarer å behandle større mengder data. De nevrale nettverkene ble utviklet av en med lite erfaring og fagkunnskap. Dette kan medføre feil i programmeringen slik at nettverkene ikke oppfører seg som planlagt. Noen aktiveringsfunksjoner kan få økt eller redusert identifiseringsrate som følge av dette.

## Konklusjon

Studien forsterker den alternative hypotesen: En av de testede aktiveringsfunksjonene resulterte i en økt læringsevne og -hastighet, i forhold til de andre aktiveringsfunksjonene som ble testet. Der av de seks aktiveringsfunksjonene som ble testet var PReLU den beste til identifisering av bilder fra MNIST-datasettet etter trening på 60 000 bilder. Resultatene viste at valg av riktig aktiveringsfunksjon er viktig for å oppnå presise og pålitelige nevrale nettverk, og at det også har betydning for nettverkes læringsevne og læringshastighet. Videre forskning på andre aktiviseringsfunksjoner, og på mer komplekse datasett er nødvendig for videre optimalisering av nevrale nettverk.

## Kilder

Thakur. A. (2022). What's the Optimal Batch Size to Train a Neural Network?. *Weights & Biases Fully Connected*. https://wandb.ai/ayush-thakur/dl-question-bank/reports/What-s-the-Optimal-Batch-Size-to-Train-a-Neural-Network---VmlldzoyMDkyNDU

Chieng, H. H., Wahid, N., & Ong, P. (2021). Parametric Flatten-T Swish: An adaptive non-linear activation function for deep learning. Journal of Information and Communication Technology, 20(1), 21-39. https://doi.org/10.32890/jict.20.1.2021.9267

Sharam, S., Sharam, S., Athaiya., Dept. of Computer Science and Engineering, Global Institute of Technology, Jaipur. (April, 2020). *Activation functions in neural networks* (ISSN No. 2455-2143). Ijeast.com. https://www.ijeast.com/papers/310-316,Tesma412,IJEAST.pdf

Maheshkar, S. (2022). Normalization Series: What is Batch Normalization?. *Weights & Biases Fully Connected*. https://wandb.ai/wandb_fc/Normalization/reports/Normalization-Series-What-is-Batch-Normalization---VmlldzoxMjk2ODcz

Ahsan, M.M., Mahmud, M. A. P., Saha, P. K., Gupta, K. D., Siddique, Z. Technologies. (24, Juli, 2021). Effect of Data Scaling Methods on Machine Learning Algorithms and Model Performance. Utgivelsessted: MDPI. https://shareok.org/bitstream/handle/11244/334438/Ahsan_2021_EffectOfDataScaling.pdf;jsessionid=63DF4EECB8F081150B859892057152EE?sequence=1