Dataset: Star

NN Classification

Markus Kinn

I dette dokumentet kommer jeg til å diskutere valg jeg har tatt og sammenligne resultater fra ulike algoritmer. Jeg har ett dokument for hvert datasett.

For å forbedre resultatene på modellene har jeg prøvd å implementere så mye so mulig fra *Machine Learning Performance Improvement Cheat Sheet.*

**Generelt om datasettet:**

Datasettet er lagd for å kunne klassifisere stjerner ut ifra disse attributtene: Temperature, Luminosity, Radius, Absolute magnitude, Star color og Spectral class. Forklaring av hver kolonne finnes i EDA notebook

Datasettet har ingen feil.

Her måtte jeg også one-hot encode alle kategoriske variabler og deretter gjøre alle features om til tensors.

**Neural Network 1:**

**Før tuning:**

**Text

Description automatically generated**

**Etter tuning:**

Jeg møtte på mange problemer med Automatisk hyperparameter tuningen når det gjaldt NN og endte derfor opp med å gjøre dette manuelt. Jeg dokumenterer derfor prosessen her:

Først bestemte jeg meg for epcohs som ble 200, av den grunn at dette er en fin balanse mellom tid og gevinst. Optimizer funksjonen ble Adam da jeg som regel har hatt best resultat med denne fra tidligere erfaring. Loss funksjonen ble categorical crossentropy da dette er et klassifiserings problem med mange klasser. Jeg gjorde dette for å kutte ned på mengden variabler jeg måtte teste og testet derfor variabler som kun hørte til selve NN-arkitekturen. Jeg vet at dette er langt fra optimalt, men fra mye testing på workshop oppgavene var det ofte disse verdiene som gikk igjen.

Etter det fant jeg ut hvilke activation functions som egnet seg for datasettet. Jeg visste fra før at Sigmoid ofte egner seg som aktiverings funksjon i output-laget for klassifiserings problemer. Jeg måtte da teste ut activation functions på alle de skulte lagene. For å spare tid brukte jeg den samme aktiverings funksjonen på alle skulte lag om gangen. Softmax, sigmoid, tanh og relu var de aktiverings funksjonene jeg valgte å teste. Av disse var tanh best.

Etter activation function gikk jeg over til batch size. Her fant jeg ut at den beste verdien er 6.

Neste gikk jeg over til å teste antall neuroner i hvert lag. Her hadde det vært ekstremt nyttig med en automatisk versjon av dette, da det finnes enormt mange ulike kombinasjoner man kan velge mellom. Her valgte jeg å teste relativt tilfeldig. Her presterte modellen desidert gjennomsnittlig best med en flat struktur. Med dette mener jeg at mengden neuroner var lik i alle skjulte lag

Den beste modellen ble seende sånn ut:

Text

Description automatically generated

**Diskusjon av resultater:**

Baseline modellen presterte ok og hadde tilnærmet ingen over-fitting.

Tuning modellen fikk en ekstremt høy økning i treffsikkerhet, sammenlignet med baseline modellen, men har litt over-fitting. Det høye resultatet kommer som et resultat av ny aktiverings funksjon, samt redusert batch size.

**Neural Network 2:**

I denne arkitekturen valgte jeg å legge til noen drop-out lag. Dette gjorde jeg for å prøve å fjerne over-fittingen som oppstod i den første arkitekturen.

**Før tuning:**

**Text

Description automatically generated**

**Etter tuning:**

Jeg fulgte den samme tuning prosessen som i den første arkitekturen.

**Text

Description automatically generated**

**Diskusjon av resultater:**

Baseline modellen sliter med en del over-fitting, men scorer generelt dårligere enn baseline modellen til den første arkitekturen.

Her kan jeg ikke helt rettferdig sammenligne baseline og tuning modellen, da jeg økte antall epochs til 300 for tuning modellen. Jeg oppdaget nemlig at baseline modellen ble stoppet mens loss per epoch fortsatt sank drastisk.

Jeg vil fremdeles si at tuning modellen er den soleklare vinneren.

**Sammenligning av alle resultater:**

På dette datasettet presterte den andre arkitekturen best. Jeg klarte å øke test scoren, samtidig som training scoren holdt seg helt lik. Dette gjorde jeg ved å introdusere drop-out-lag, samtidig som jeg byttet over tid en pyramide struktur med tanke på neuroner.