ITMAL Øvelser – Artificial Neural Networks

Øvelse 1 :

Afprøv Scikit-learns version af supervised neural network – mere specifikt arbejder vi her med 2-lags Multi Layer Perceptron (klassisk feedforward ANN, der findes mange andre typer ANNs). Scikit Learns funktion til regression hedder MLPRegression.

OBS: Når I laver dybe neurale netværk (DNN) – herunder CNN etc., så er det nok bedre at bruge andre biblioteker såsom Keras. Men Scikit learns funktion er ret simpel og god til denne opgave.

Vi antager at I har fået data, som beskrevet nederst i "ANN_example.py" (i denne uges materiale) – dvs. feature x = np.linspace(-10,10,1000) og output værdi y = np.sinc(x), altså en såkaldt sinc-funktion (Bemærk – det her er blot til illustration af, at MLP regressionsmodeller kan fitte vilkårlige funktioner, og også i flere dimensioner). Opgaven er nu at "lære"/"træne" netværket til dette output.

a. Fit modellen til data. Benyt i starten kun 2 skjulte neuroner (hidden_layer_sizes)
b. Tegn den grafiske model for netværket – skriv vægtenes værdi på grafen (gerne i hånden). Husk bias.
c. Opskriv udtrykket for y – dvs. i stil med y = 0.3* tanh(2 * x + 0.1) + 0.3 * tanh(5 * x + 3) + 1. OBS: I kan godt nøjes med fx. 1-2 betydende decimaler.
d. Plot funktionen vha. "np.tanh" – dvs. i stil med "y = 0.3*np.tanh(2 * x +..." hvor x er input data.
e. Plot også første del af funktionen samt anden del, hver for sig (fx. "0.3* tanh(2 * x + 0.1)" og " 0.3 * tanh(5 * x + 3)") – summen af disse to skal jo gerne give den samlede funktion (pånær bias-leddet)
f. Prøv også at fitte funktionen med flere led (fx. 5). Plot resultatet.
g. Optional : Prøv at ændre alpha til fx. 1e5 og 1e-1 – forklar hvad der sker (tip: regularisering).

Tips : I kan tage udgangspunkt i koden til denne uge, også til fitting af modellen. I kan blot ændre fx. hidden_layer_sizes.