Øving 4 - Artificial Neural Networks

Marius Aarsnes

October 2017

1 Oppgave A: Betrakt læringsregelen til et perceptron, se foiler fra forelesning eller side 172 i utdelt kapittel på ANN (Artificial Neural Networks). Kod algoritmen i et valgfritt programmeringsspråk

Koden ligger vedlagt i filen "Perceptron.py". Det er gjort et forsøk på å forklare det meste, for å gjøre ting så forståelig som mulig. Koden består av en klasse Perceptron som tar in en rekke verdier ved initiering og en main-metode som oppretter to instanser av Perceptron og trener de opp til å forstå AND og OR, hver for seg.

2 Oppgave B: Kjør læringsalgoritmen i A) med input fra AND og OR og legg ved utskrift på hvordan vektene justerer seg. Du kan legge inn dataene direkte i koden. Blir vektene de samme når du forandrer startverdier på vektene w_i og terskelen Θ?

Det ser ut som om verdien til Θ krever å være innenfor et visst intervall, eller over en viss verdi for at de ulike læringsprosessene skal fungere. OR-perceptronet ser ut til å feile hver gang Θ er negativ, mens AND-perceptronet ser ut til å "kreve" at Θ er minst 0.3 for at den skal klare å garantere suksess. Bilde av resultat fra kjøringen ligger vedlagt.

Bildene som viser perceptronene feile er uten endringene i vekt fordi det ville blitt en veldig lang remse. Måten koden fungerer på er at den håndterer runtime-error (uendelig loop). Det tar en del iterasjoner før Python oppfatter dette. Generelt var det ingen endring i vektene.

3 Oppgave 3: I denne oppgaven skal vi regne på én iterasjon med backpropagation på et nevralt nett som vist i figuren nedenfor.

$$x_1 = 0, x_2 = 1, y_{d5} = 0, y_{d6} = 1, \alpha = 0.1$$

$$w_{13} = 0.5, w_{14} = 0.0, w_{23} = 0.0, w_{24} = 0.9$$

$$w_{35} = 0.4, w_{36} = 1.0, w_{45} = -1.2, w_{46} = 1.1$$

$$\Theta_3 = 0.8, \Theta_4 = -0.1, \Theta_5 = 0.3, \Theta_6 = 0.5$$

Hva vil verdiene til vektene og thresholdene være etter en ny iterasjon?

$$y_3 = sigmoid(x_1 \times w_{13} + x_2 \times w_{23} - \Theta_3) = \frac{1}{1 + e^{-(0.0 \times 0.5 + 1 \times 0.0 - 0.8}} = \underline{0.4493}$$

$$y_4 = sigmoid(x_1 \times w_{14} + x_2 \times w_{24} - \Theta_4) = \frac{1}{1 + e^{-(0 \times 0.0 + 1 \times 0.9 + 0.1}} = \underline{0.7311}$$

$$y_5 = sigmoid(y_3 \times w_{35} + y_4 \times w_{45} - \Theta_5) = \frac{1}{1 + e^{-(0.4493 \times 0.4 + 0.7311 \times (-1.2) - 0.3}} = \underline{0.2694}$$

$$y_6 = sigmoid(y_3 \times w_{36} + y_4 \times w_{46} - \Theta_6) = \frac{1}{1 + e^{-(0.4493 \times 1.0 + 0.7311 \times 1.1 - 0.5}} = \underline{0.6799}$$

$$e_5 = y_{d5} - y_5 = 0 - 0.2694 = -0.2694$$

$$e_6 = y_{d6} - y_6 = 1 - 0.6799 = -3201$$

$$\delta_5 = y_5 \times (1 - y_5) \times e_5 = 0.2694 \times (1 - 0.2694) \times (-0.2694) = -0.0530$$

$$\delta_6 = y_6 \times (1 - y_6) \times e_6 = 0.6799 \times (1 - 0.6799) \times (-3201) = -0.06967$$

$$\delta_3 = y_3 \times (1 - y_3) \times \delta_5 \times w_{35} \times \delta_6 \times w_{36} = 0.000365$$

$$\delta_4 = y_4 \times (1 - y_4) \times \delta_5 \times w_{45} \delta_6 \times w_{46} = -0.000958$$

$$\Delta\Theta_3 = \alpha \times (-1)\delta_3 = 0.1 \times (-1) \times 0.000365 = -0.0000365$$

$$\Delta\Theta_4 = \alpha \times (-1)\delta_4 = 0.1 \times (-1) \times (-0.000958) = 0.0000958$$

$$\Delta\Theta_5 = \alpha \times (-1)\delta_5 = 0.1 \times (-1) \times (-0.0530) = 0.0053$$

$$\Delta\Theta_6 = \alpha \times (-1)\delta_6 = 0.1 \times (-1) \times (-0.06967) = 0.006967$$

$$\Delta w_{35} = \alpha \times y_3 \times \delta_5 = 0.1 \times 0.4493 \times (-0.0530) = -0.00238$$

$$\Delta w_{36} = \alpha \times y_3 \times \delta_6 = 0.1 \times 0.4493 \times (-0.06967) = -0.00313$$

$$\Delta w_{45} = \alpha \times y_4 \times \delta_5 = 0.1 \times 0.7311 \times (-0.0530) = -0.003874$$

$$\Delta w_{46} = \alpha \times y_4 \times \delta_6 = 0.1 \times 0.7311 \times (-0.06967) = -0.00509$$

$$\Delta w_{13} = \alpha \times x_1 \times \delta_3 = 0.1 \times 0... = 0$$

$$\Delta w_{14} = \alpha \times x_1 \times \delta_4 = 0.1 \times 0... = 0$$

$$\Delta w_{23} = \alpha \times x_2 \times \delta_3 = 0.1 \times 1 \times (-0.0000365) = -0.00000365$$

$$\Delta w_{24} = \alpha \times x_2 \times \delta_4 = 0.1 \times 1 \times 0.0000958 = 0.00000958$$

$$w_{13} = w_{13} + \Delta w_{13} = 0.5 + 0 = \underline{\underline{0.5}}$$

$$w_{14} = w_{14} + \Delta w_{14} = 0.0 + 0 = \underline{\underline{0.0}}$$

$$w_{23} = w_{23} + \Delta w_{23} = 0.0 + (-0.00000365) = \underline{\underline{-0.00000365}}$$

$$w_{24} = w_{24} + \Delta w_{24} = 0.9 + (0.00000958) = \underline{\underline{0.90000958}}$$

$$w_{35} = w_{35} + \Delta w_{35} = 0.4 + (-0.00238) = \underline{\underline{0.39762}}$$

$$w_{36} = w_{36} + \Delta w_{36} = 1.0 + (-0.00313) = \underline{\underline{0.99687}}$$

$$w_{45} = w_{45} + \Delta w_{45} = -1.2 + (-0.003874) = \underline{\underline{-1203874}}$$

$$w_{46} = w_{46} + \Delta w_{46} = 1.1 + (-0.006967) = \underline{\underline{1.093033}}$$

$$\Theta_3 = \Theta_3 + \Delta\Theta_3 = 0.8 + (-0.0000365) = \underline{\underline{0.7999635}}$$

$$\Theta_4 = \Theta_4 + \Delta\Theta_4 = -0.1 + (0.0000958) = \underline{\underline{0.0999042}}$$

$$\Theta_5 = \Theta_5 + \Delta\Theta_5 = 0.3 + (0.0053) = \underline{\underline{0.30530}}$$

$$\Theta_6 = \Theta_6 + \Delta\Theta_6 = 0.5 + 0.006967 = \underline{\underline{0.506967}}$$

Det er gjort forbehold om at det kan være ulike former for slurving og/eller andre feil i utregning.

4 I den neste delen av øvingen skal vi utforske et feed-forward-nettverk. Det finnes mange biblioteker som kan hjelpe med maskinlæring gjennom nevrale nettverk. Vi skal lage et nevralt feed-forward nettverk hvor input er lik output (kalles en autoenkoder). Vi vil undersøke hvor stort nettverket må være for å gjenskape oppførselen vi ønsker

Koden finnes vedlagt i filen 'feed_forward.py'. Files inneholder en klasse som oppretter test-data-set og et netverkt med et git antall hidden-layer noder. hovedprogrammet oppretter instanser av FeedForward-klassen og kjører testing med ulike verdier.

1. Det er litt vanskelig å si hva som er den absolutt laveste grensen for å få gode resultater, men det ser ut som om 3, og oppover, antall hidden-layer gir stabilt resultat innenfor det gitte intervallet.

2. Ser ut som regresjon. Netverket endrer vektene slik slik at de passer til input-output forholdet. Netverket genererer altså output ved å bruke data fra trainingen.

3. Det ser ut som om, generelt, vil input utenfor det gyldige intervallet gir merkelige resultat. Der jo lenger unna, jo merkeligere. Verediene som er nærme det gyldige intervallet, men ikke inni gir passe riktige svar. Desimaltall inne i intervallet gir passe riktige svar.