

Tutorial Artificial Neural Network – Perceptron

Di tutorial ini, kita akan membahas mengenai bagaimana kita melakukan training pada ANN – perceptron. Seperti yang sudah dijelaskan di pertemuan di kelas, ANN dapat berupa perceptron dan sigmoid perceptron. Hal yang membuat mereka berbeda adalah activation function atau $g(in_j)$ nya. Perceptron itu menggunakan hard threshold, yang artinya kalau lebih besar dari 0, hasilnya jadi 1. Lalu kalau lebih kecil dari 0, hasilnya jadi -1 atau 0 bergantung pada masalah yang ingin dipecahkan. Biasanya perceptron digunakan untuk mengkategorikan data secara linear (atau dalam bentuk garis lurus di 2 dimensi).

$$a_j = g(in_j) = g\left(\sum_{i=0}^n w_{i,j}a_i\right)$$

Langkah-langkah untuk melakukan perceptron learning adalah sebagai berikut:

- Pilih sample data secara random dari training set
- Apabila pengkategorian (classification)nya benar, maka tidak perlu melakukan apa apa
- Apabila salah, kita perlu mengupdate weight (bobot) yang ada sehingga dapat melakukan pengkategorian dengan benar
 - Update dengan: $w_i(t+1) = w_i(t) + \eta(d - y)a_i$
 - w adalah weight / bobot
 - t adalah iterasi ke berapa
 - η adalah learning rate
 - d adalah output yang diinginkan
 - y (a_j) adalah output yang dihasilkan
 - a_i adalah input dari activation function
- Lakukan prosedur di atas berulang kali sampai semua data di training set dapat dikategorikan secara benar

Sekarang kita akan coba mengaplikasikannya pada contoh kasus di slide binusmaya lama – Single perceptron to represent OR

- 2 inputs
- 1 output
- Step function, kalau $g(in_j)$ lebih besar dari 0.5 maka outputnya adalah 1

Seperti yang kita ketahui, untuk kasus OR pada binary ada 4 data yaitu:

a_1	a_2	d
1	1	1
1	0	1
0	1	1
0	0	0

Lalu kita mempunyai inisial weight dan learning rate yang diberikan yaitu:

- $w_1 = 0.1$
- $w_2 = 0.6$
- $\eta = 0.1$

Masalah yang harus diselesaikan adalah, update weight yang diberikan sehingga dapat melakukan operasi OR dengan baik

- Iterasi 1 (Input: 1,1 = 1)
 - Hitung hasil dari activation function
 - $g(in_j) = 0.1*1 + 0.6*1 = 0.7$
 - Lakukan thresholding pada hasil dari activation function
 - $y = g(in_j) > 0.5 = 0.7 > 0.5 = 1$
 - Cek apakah hasil dari thresholding sama dengan output yang diinginkan (d)
 - Jika sama, ($1 == 1$), tidak dilakukan apa apa
- Iterasi 2 (Input: 1,0 = 1)
 - Hitung hasil dari activation function
 - $g(in_j) = 0.1*1 + 0.6*0 = 0.1$
 - Lakukan thresholding pada hasil dari activation function
 - $y = g(in_j) > 0.5 = 0.1 > 0.5 = 0$
 - Cek apakah hasil dari activation function lebih besar dari threshold
 - Jika tidak sama, ($0 == 1$), update weight
 - Update weight → cek equation di atas

- $w_1 = 0.1 + 0.1 * (1-0) * 1 = 0.2$
 - $w_2 = 0.6 + 0.1 * (1-0) * 0 = 0.6$
- Iterasi 3 (Input: 0,1 = 1)
 - Hitung hasil dari activation function
 - $g(in_j) = 0.2*0 + 0.6*1 = 0.6$
 - Lakukan thresholding pada hasil dari activation function
 - $y = g(in_j) > 0.5 = 0.6 > 0.5 = 1$
 - Cek apakah hasil dari thresholding sama dengan output yang diinginkan (d)
 - Jika sama, ($1 == 1$), tidak dilakukan apa apa
- Iterasi 4 (Input: 0,0 = 0)
 - Hitung hasil dari activation function
 - $g(in_j) = 0.2*0 + 0.6*0 = 0$
 - Lakukan thresholding pada hasil dari activation function
 - $y = g(in_j) > 0.5 = 0 > 0.5 = 0$
 - Cek apakah hasil dari thresholding sama dengan output yang diinginkan (d)
 - Jika sama, ($0 == 0$), tidak dilakukan apa apa
- Iterasi 5 (Input: 1,0 = 1)
 - Hitung hasil dari activation function
 - $g(in_j) = 0.2*1 + 0.6*0 = 0.2$
 - Lakukan thresholding pada hasil dari activation function
 - $y = g(in_j) > 0.5 = 0.2 > 0.5 = 0$
 - Cek apakah hasil dari activation function lebih besar dari threshold
 - Jika tidak sama, ($0 == 1$), update weight
 - Update weight → cek equation di atas
 - $w_1 = 0.2 + 0.1 * (1-0) * 1 = 0.3$
 - $w_2 = 0.6 + 0.1 * (1-0) * 0 = 0.6$
- Iterasi 6 (Input: 1,0 = 1)
 - Hitung hasil dari activation function
 - $g(in_j) = 0.3*1 + 0.6*0 = 0.3$
 - Lakukan thresholding pada hasil dari activation function
 - $y = g(in_j) > 0.5 = 0.3 > 0.5 = 0$
 - Cek apakah hasil dari activation function lebih besar dari threshold
 - Jika tidak sama, ($0 == 1$), update weight
 - Update weight → cek equation di atas

- $w_1 = 0.3 + 0.1 * (1-0) * 1 = 0.4$
 - $w_2 = 0.6 + 0.1 * (1-0) * 0 = 0.6$
- Lakukan iterasi terus menerus sampai mendapatkan $w_1 = 0.6$ dan $w_2 = 0.6$