

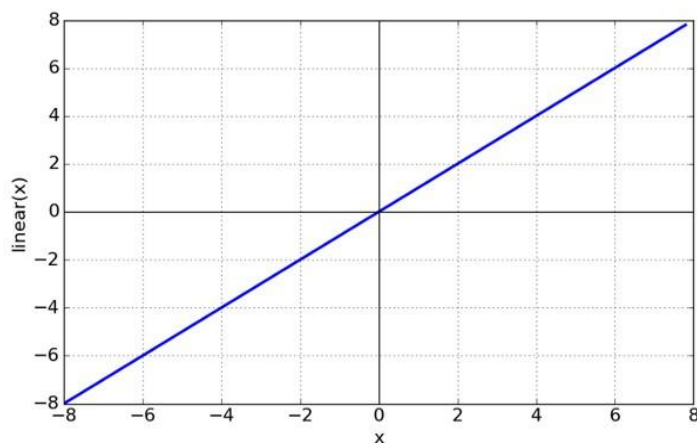
DEEP LEARNING

Oleh : Oscar Hadikaryana, ST, MT

Pada bagian ini merupakan lanjutan dari pembelajaran AI, sehingga tidak terlalu membahas tentang Artificial Neural Network.

Activation Function

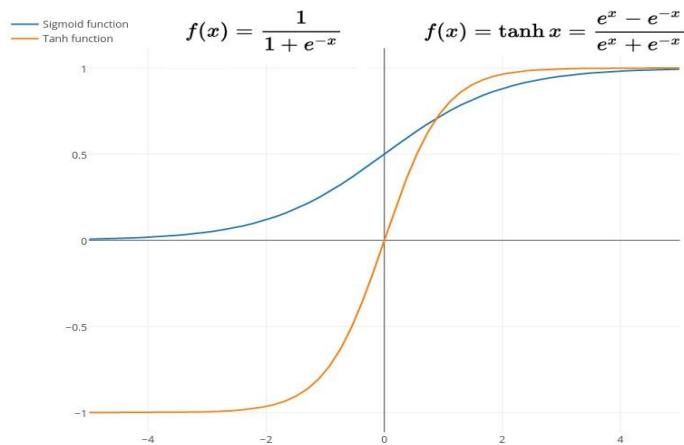
Nah, mungkin banyak yang bingung apa dan untuk apa activation function? Sesuai dengan namanya, activation function berfungsi untuk menentukan apakah neuron tersebut harus “aktif” atau tidak berdasarkan dari weighted sum dari input. Secara umum terdapat 2 jenis activation function, Linear dan Non-Linear Activation function.



Gmb. 1, Linier Activation Funtion

Bisa dikatakan secara “default” activation function dari sebuah neuron adalah Linear. Jika sebuah neuron menggunakan linear function, maka keluaran dari neuron tersebut adalah weighted sum dari input + bias.

Sigmoid and Tanh Function (Non-Linear)

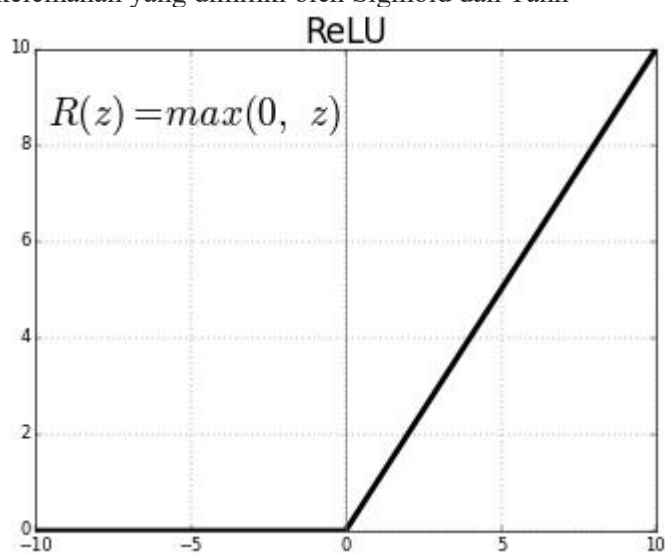


Gmb. 2, Sigmoid Function

Sigmoid function mempunyai rentang antara 0 hingga 1 sedangkan rentang dari Tanh adalah -1 hingga 1. Kedua fungsi ini biasanya digunakan untuk klasifikasi 2 class atau kelompok data. Namun terdapat kelemahan dari kedua fungsi ini, nanti akan coba saya jelaskan di part berikutnya.

ReLU (Non-Linear)

Pada dasarnya ReLU melakukan “treshold” dari 0 hingga infinity. ReLU juga dapat menutupi kelemahan yang dimiliki oleh Sigmoid dan Tanh



Gmb.3, Relu (Non Linier)

Neural Network Architectures

Arsitektur diatas biasa disebut sebagai Multi Layer Perceptron (MLP) atau Fully-Connected Layer. Arsitektur pertama mempunyai 3 buah neuron pada Input Layer dan 2 buah node Output Layer. Diantara Input dan Output, terdapat 1 Hidden Layer dengan 4 buah neuron. Sedangkan spesifikasi Weight dan Activation function adalah sebagai berikut:

Weight and Bias

Setiap neuron pada MLP saling berhubungan yang ditandai dengan tanda panah pada gambar diatas. Tiap koneksi memiliki weight yang nantinya nilai dari tiap weight akan berbeda-beda.

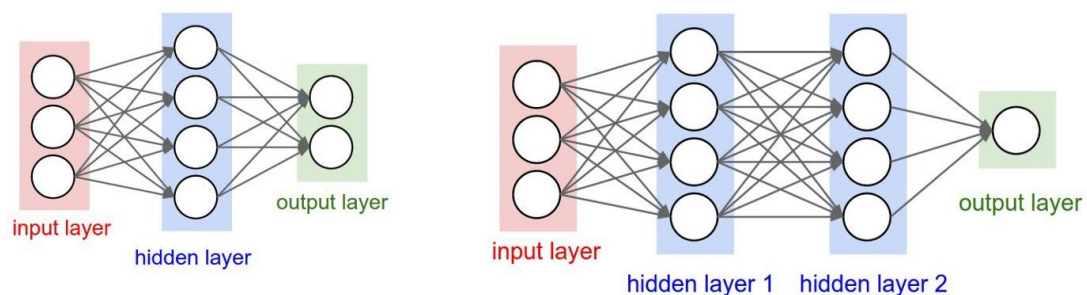
Hidden layer dan output layer memiliki tambahan “input” yang biasa disebut dengan bias (Tidak disebutkan pada gambar diatas).

Sehingga pada arsitektur pertama terdapat 3×4 weight + 4 bias dan 4×2 weight + 2 bias. Total adalah 26 parameter yang pada proses training akan mengalami perubahan untuk mendapatkan hasil yang terbaik. Sedangkan pada arsitektur kedua terdapat 41 parameter.

Activation Function

Neuron pada input layer tidak memiliki activation function, sedangkan neuron pada hidden layer dan output layer memiliki activation function yang kadang berbeda tergantung daripada data atau problem yang kita miliki.

Training a Neural Network



Gmb. 4, Arsitektur NN

Pada Supervised Learning menggunakan Neural Network, pada umumnya Learning terdiri dari 2 tahap, yaitu training dan evaluation. Namun kadang terdapat tahap tambahan yaitu testing, namun sifatnya tidak wajib.

Pada tahap training setiap weight dan bias pada tiap neuron akan diupdate terus menerus hingga output yang dihasilkan sesuai dengan harapan. Pada tiap iterasi akan dilakukan proses evaluation yang biasanya digunakan untuk menentukan kapan harus menghentikan proses training (stopping point)

Pada part selanjutnya, akan saya bahas bagaimana proses training pada neural network lebih mendalam. Namun pada part ini akan dijelaskan garis besarnya saja. Proses training terdiri dari 2 tahap :

- Forward Pass
- Backward Pass

Forward Pass

Forward pass atau biasa juga disebut forward propagation adalah proses dimana kita membawa data pada input melewati tiap neuron pada hidden layer sampai kepada output layer yang nanti akan dihitung errornya

$$dot_j = \sum_i^3 w_{ji}x_i + b_j$$

$$h_j = \sigma(dot_j) = \max(0, dot_j)$$

Persamaan diatas adalah contoh forward pass pada arsitektur pertama (lihat gambar arsitektur diatas) yang menggunakan ReLU sebagai activation function. Dimana i adalah node pada

input layer (3 node input), j adalah node pada hidden layer sedangkan h adalah output dari node pada hidden layer.

Backward Pass

Error yang kita dapat pada forward pass akan digunakan untuk mengupdate setiap weight dan bias dengan learning rate tertentu.

Kedua proses diatas akan dilakukan berulang-ulang sampai didapatkan nilai weight dan bias yang dapat memberikan nilai error sekecil mungkin pada output layer (pada saat forward pass)

Tugas :

Buat code programming untuk proses training forward pass dan backward pass serta tunjukan hasilnya. Untuk kasusnya adalah melakukan prediksi nilai dari 7, 8, 9 dan 10

Luaran

Video program dan hasilnya di youtube chanel sendiri.

Referensi :

Rikie Primartha, Belajar Machine Learning-Teori dan Praktek, Pnb. Informatika, Bandung, 2018

<https://medium.com/@samuelsena/pengenalan-deep-learning-8fbb7d8028ac>