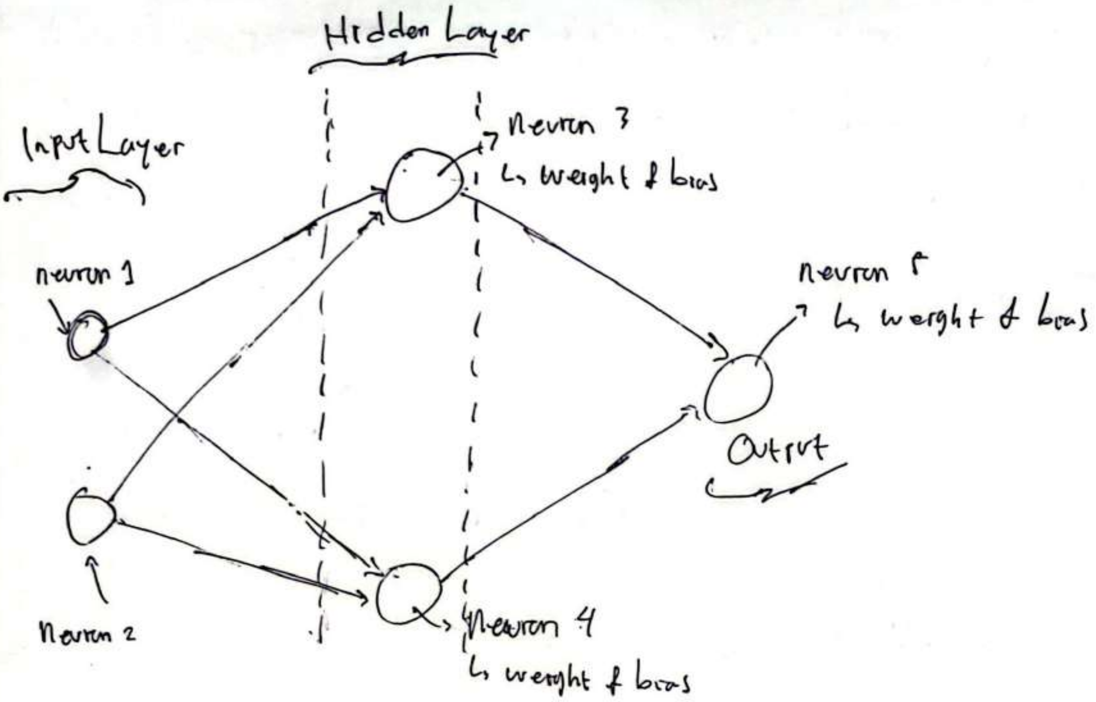
Basic Neural Networks



~~Secara Umum~~ Konsep Umum dari Neural Network itu ialah Perancangan Struktur end to end yang iteratif dan murni bergantung pada optimasi bobot/weight (kurang lebih sama seperti model Linear Regresi).

Model Neural Network ini mewajibkan kita membangun struktur dari end to end, yaitu dimulai dari Input Layer, dimana pada input layer semua variabel/kolom data kita terdefinisi sebagai neuron yang mana neuron-neuron pada input layer ini akan diproses di hidden layer. Hidden Layer berfungsi mendoping neuron² dari input layer dengan parameter yg nanti teriterasi (Weight dan bias).

~~Weight dan bias~~ Setiap neuron pada input layer dihubungkan ke semua neuron yang ada pada neuron untuk semuanya dikalkulasikan bobotnya. Oiya kenapa neuron pada Hidden Layer di set banyak; puluhan, bahkan ratusan?

ini dilakukan karena untuk meningkatkan kemampuan representasi model, mempelajari pola yang kompleks dan nonlinear, yaitunya agar model yg kita bangun ini Reliable.

ex. $n_3 : n_1 \cdot Weight(3,1) + n_2 \cdot Weight(3,2) + b_3 \rightarrow \sigma(n_3)$

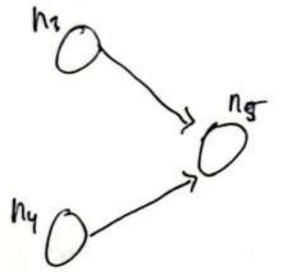
$n_4 : n_1 \cdot Weight(4,1) + n_2 \cdot Weight(4,2) + b_4 \rightarrow \sigma(n_4)$

$n_5 : \sigma(n_3) \cdot Weight(5,3) + \sigma(n_4) \cdot Weight(5,2) + b_5 \rightarrow output$

Setelah semua neuron di hidden layer terkalkulasikan bobot dan biasnya,

kemudian setiap output perhitungan neuron di hidden layer dikenakan fungsi aktivasi.
Inilah yg juga menjadi pembeda dari pada model regresi linear pada ML tradisional.
fungsi non linear ini memungkinkan jaringan mempelajari pola rumit dan kompleks dalam data. Misal fungsi aktivasi seperti Relu ($\max(0,h)$), Sigmoid ($\frac{1}{1+e^{-z}}$), dll.

Semua perhitungan fungsi aktivasi dari semua neuron di hidden layer kemudian collapse ke dalam satu pengukuran / satu neuron untuk mendapat hasil prediksi.



$\sigma(n_3) \cdot W + \sigma(n_4) \cdot W + b_5$, σ → fungsi aktivasi

↳ $\sigma(n_5)$ = hasil prediksi akhir.

Untuk mengukur nilai keakuratan model neural network ini dengan label sebenarnya, dilakukan perhitungan dengan menghitung fungsi kerugian (loss function) dari output.
misal fungsi kerugian yg sederhana seperti Mean Square error → $(Y - \hat{Y})^2$
kemudian loss function tersebut akan diiterasi sampai konvergen (minimum) dengan menghitung gradient loss function terhadap setiap bobot dan bias dalam proses sebelumnya.
Istilah ini dikenal dalam ~~pr~~ algoritma Backpropagation atau dikenal Training data.

Backpropagation

Backpropagation sesuai dengan namanya, yg berarti algoritma ini berjalan mundur dari layer terakhirnya ke awal. Backpropagation dibuat untuk men train data yakni meng adjust tiap bobot dan bias yg telah dilalui forward pass sebelumnya dengan menghitung gradient nya untuk meminimalkan kesalahan..

Algoritma ini didahului oleh Loss function, misal pada perhitungan ini menggunakan
MSE → $L = (Y_{true} - Y_{pred})^2$

Backpropagation (Implementasi dari turunan)

• Gradient terhadap output (a_3)

↳ intuisinya seberapa salah prediksi model thd output

$$L = (y_{true} - a_3)^2$$

$$\frac{\partial L}{\partial a_3} = -2(y_{true} - a_3) \rightarrow \text{turunan parsial}$$

• hitung Gradient terhadap z_3 (input aktivasi output)

$$\frac{\partial L}{\partial z_3} = \frac{\partial L}{\partial a_3} \cdot \frac{\partial a_3}{\partial z_3} \rightarrow a_3 = \sigma(z_3)$$

$$\frac{\partial L}{\partial z_3} = \underbrace{-2(y - a_3)}_{\text{turunan Loss thd output}} \cdot \underbrace{\sigma'(z_3)}_{\text{turunan Sigmoid}}$$

$$\sigma = \frac{1}{1+e^{-x}} = \frac{d}{dx}\sigma = \frac{d}{dx}\left(\frac{1}{1+e^{-x}}\right)$$

$$\sigma' = -1 \cdot (1+e^{-x})^{-2} \cdot (-e^{-x})$$

$$= \frac{e^{-x}}{(1+e^{-x})^2}$$

$$\sigma'(x) = \frac{1}{1+e^{-x}} \cdot \frac{e^{-x}}{1+e^{-x}}$$

$$= \sigma(x) \cdot \frac{e^{-x}}{1+e^{-x}}$$

$\frac{e^{-x}}{1+e^{-x}}$ ⇒ kita manipulasi bentuknya ke $\sigma(x)$

$$\frac{e^{-x}}{1+e^{-x}} + \frac{1}{1+e^{-x}} = 1 \Rightarrow 1 - \sigma(x)$$

$$\hookrightarrow \sigma'(x) = \sigma(x)\,(1 - \sigma(x))$$

• hitung Gradient thd bobot output (W_o, B_o)

↳ $z_3 = a_1 \cdot W\_o[0] + a_2 \cdot W\_o[1] + B\_o$

$$\frac{\partial L}{\partial W_o[0]} = \frac{\partial L}{\partial z_3} \cdot \frac{\partial z_3}{\partial W_o[0]} = \frac{\partial L}{\partial z_3} \cdot a_1$$

$$\frac{\partial L}{\partial W_o[1]} = \frac{\partial L}{\partial z_3} \cdot \frac{\partial z_3}{\partial W_o[1]} = \frac{\partial L}{\partial z_3} \cdot a_2$$

$$\frac{\partial L}{\partial B_o} = \frac{\partial L}{\partial z_3} \cdot 1$$

> hitung gradien terhadap Hidden Layer (a_1, a_2)

$$\frac{\partial L}{\partial a_1} = \frac{\partial L}{\partial z_3} \cdot \frac{\partial z_3}{\partial a_1} = \frac{\partial L}{\partial z_3} \cdot W_o[0]$$

$$\frac{\partial L}{\partial a_2} = \frac{\partial L}{\partial z_3} \cdot \frac{\partial z_3}{\partial a_2} = \frac{\partial L}{\partial z_3} \cdot W_o[1]$$

-> hitung gradien terhadap z_1, z_2 (input aktivasi hidden)

$$\frac{\partial L}{\partial z_1} = \frac{\partial L}{\partial a_1} \cdot \frac{\partial a_1}{\partial z_1} = \frac{\partial L}{\partial a_1} \cdot \sigma'(z_1)$$

$$\frac{\partial L}{\partial z_2} = \frac{\partial L}{\partial a_2} \cdot \frac{\partial a_2}{\partial z_2} = \frac{\partial L}{\partial a_2} \cdot \sigma'(z_2)$$

-> hitung gradien terhadap bobot input (W_h, B_h)

* neuron hidden pertama -> $z_1 = x_0 \cdot W_h[0][0] + x_1 \cdot W_h[1][0] + B_h[0]$

$$\frac{\partial L}{\partial W_h[0][0]} = \frac{\partial L}{\partial z_1} \cdot \frac{\partial z_1}{\partial W_h[0][0]} = \frac{\partial L}{\partial z_1} \cdot x_0$$

$$\frac{\partial L}{\partial W_h[1][0]} = \frac{\partial L}{\partial z_1} \cdot \frac{\partial z_1}{\partial W_h[1][0]} = \frac{\partial L}{\partial z_1} \cdot x_1$$

$$\frac{\partial L}{\partial B_h[0]} = \frac{\partial L}{\partial z_1} \cdot \frac{\partial z_1}{\partial B_h[0]} = \frac{\partial L}{\partial z_1}$$

* neuron hidden kedua -> $z_2 = x_0 \cdot W_h[0][1] + x_1 \cdot W_h[1][1] + B_h[1]$

$$\frac{\partial L}{\partial W_h[0][1]} = \frac{\partial L}{\partial z_2} \cdot \frac{\partial z_2}{\partial W_h[0][1]} = \frac{\partial L}{\partial z_2} \cdot x_0$$

$$\frac{\partial L}{\partial W_h[1][1]} = \frac{\partial L}{\partial z_2} \cdot \frac{\partial z_2}{\partial W_h[1][1]} = \frac{\partial L}{\partial z_2} \cdot x_1$$

$$\frac{\partial L}{\partial B_h[1]} = \frac{\partial L}{\partial z_2}$$