

Artificial Neural Network (ANN)

Evelyn Yosiana / 13522083

1. Konsep ANN yaitu pembentukan jaringan neuron (syaraf) “tiruan” yang dikelompokkan jadi 3 jenis layer utama: input, hidden, dan output layer. Input layer menerima data yang ingin digunakan sebagai data latih ataupun prediksi. Hidden layer yang berfungsi untuk melakukan mayoritas kalkulasi terhadap data input. Output layer berfungsi untuk “mengonversi” hasil kalkulasi dari hidden layer menjadi nilai yang diinginkan.

Proses train:

- Diawali dengan inisialisasi weights pada tiap neuron secara random dan bias untuk setiap neuron bernilai 0.
- Melalui tahap feed forward maka data latih (X) akan dibagi dengan nilai batch size menghasilkan batch.
- Setiap batch dimasukkan ke input layer dan dikalikan dengan weight setiap neuron melalui operasi dot dan ditambahkan bias, secara berurutan.
- Masih dalam tahap feed forward maka data latih yang telah melalui input layer akan melalui fungsi aktivasi dan dioper ke hidden layer
- Di hidden layer terjadi hal serupa di mana X dikalikan dengan weights setiap neuron melalui operasi dot dan ditambahkan bias.
- Setelah itu, X dari hidden layer akan melalui fungsi aktivasi terakhir di output layer. Fungsi aktivasi terakhir di output layer ini menyesuaikan dengan tugas yang perlu dilakukan oleh ANN. Dalam tugas kali ini, fungsi yang digunakan adalah sigmoid, karena tugas ANN adalah melakukan binary classification.
- Hasil dari output layer dihitung performanya dengan suatu loss function, dalam tugas kali ini adalah cross entropy.
- Tidak berakhir di sana, setelah tahap feedforward adalah tahap back propagate, di mana weights dan bias setiap neuron dimodifikasi berdasarkan nilainya di cross entropy, dimulai dari weights di hidden layer dan akan bergerak “mundur” ke input layer. Proses modifikasi weights dan bias ini (optimize), bila menggunakan Stochastic Gradient Descent (SGD), masing-masing melalui rumus:

$$w' = w - \text{learning rate} * dw \text{ dan } b' = b - \text{learning rate} * db$$

dengan w' dan b' adalah weights dan bias baru yang akan digunakan setelah modifikasi, w dan b adalah weights dan bias terakhir yang sebelum dimodifikasi, learning rate adalah faktor pengali untuk menentukan seberapa berpengaruh gradien memodifikasi weights dan bias, serta dw dan db adalah gradien untuk memodifikasi weights dan bias. Alternatif metode optimize yang diimplementasikan adalah momentum-based optimizer yang rumusnya:

$$mw' = (\text{beta} * mw + (1 - \text{beta}) * dw)$$

$$w' = w - \text{learning rate} * mw'$$

$$mb' = (\text{beta} * mb + (1 - \text{beta}) * db)$$

$$b' = b - \text{learning rate} * mb'$$

- Proses feed forward dan back propagate ini dilakukan sampai setiap batch dari X sudah melalui feed forward dan back propagate.
- Proses penerapan feed forward dan back propagate terhadap setiap batch dari X dilakukan sejumlah epoch.

Proses predict: untuk proses ini, data yang akan diprediksi (X_{test}) akan melalui input layer, hidden layer, dan output layer dalam 1 kali feed forward.

4. Pada model ini, hasil perbandingan model yang saya buat dan model dari library kurang sama. Hal ini dapat disebabkan oleh beberapa faktor, antara lain:
 - **Library Optimasi:** MLP dari scikit-learn menggunakan teknik optimasi performa (LAPACK) ataupun optimasi inisialisasi weights (Glorot)
 - **Learning Rate Dinamis:** MLP dari scikit-learn memiliki adaptive learning rate schedules yang menyesuaikan learning rate secara dinamis.
5. Improvement yang dapat saya lakukan antara lain:
 - **Hyperparameter tuning** untuk mendapatkan kombinasi parameter yang optimal (salah satunya dengan menggunakan metode grid search atau library optuna).
 - Menggunakan **confusion metrics** untuk model evaluation.
 - **Gradient Clipping:** Membatasi range gradient agar tidak terlalu besar ataupun terlalu kecil sehingga bisa mengganggu stabilitas training

Lampiran

Contoh perhitungan manual:

Example :

Evelyn Yosiana / 13522083

$$\text{size} = [2, 3, 1]$$

$$w_1 = \begin{bmatrix} 0,1 & -0,2 \\ 0,3 & 0,4 \\ -0,5 & 0,6 \end{bmatrix}$$

$$b_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$w_2 = [-0,7 \quad 0,8 \quad -0,9]$$

$$b_2 = [0]$$

$$\text{input} = \begin{bmatrix} 0,5 \\ -0,5 \end{bmatrix}$$

$$z_1 = w_1 \cdot x + b_1$$

$$z_1 = \begin{bmatrix} 0,1 & -0,2 \\ 0,3 & 0,4 \\ -0,5 & 0,6 \end{bmatrix} \cdot \begin{bmatrix} 0,5 \\ -0,5 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0,1 \cdot 0,5 + (-0,2) \cdot (-0,5) \\ 0,3 \cdot 0,5 + 0,4 \cdot (-0,5) \\ (-0,5) \cdot 0,5 + 0,6 \cdot (-0,5) \end{bmatrix} = \begin{bmatrix} 0,15 \\ -0,05 \\ -0,55 \end{bmatrix}$$

$$A_1 = \sigma(z_1) = \begin{bmatrix} \sigma(0,15) \\ \sigma(-0,05) \\ \sigma(-0,55) \end{bmatrix} = \begin{bmatrix} 1 \\ 1 + e^{\dots} \end{bmatrix} = \begin{bmatrix} 0,5374 \\ 0,4875 \\ 0,3657 \end{bmatrix}$$

$$z_2 = w_2 \cdot A_1 + b_2$$

$$z_2 = [-0,7 \quad 0,8 \quad -0,9] \begin{bmatrix} 0,5374 \\ 0,4875 \\ 0,3657 \end{bmatrix} + [0] = -0,31531$$

$$A_2 = \sigma(z_2) = \frac{1}{1 + e^{0,31531}} \approx 0,42182$$