

Artificial Neural Network

1. Jelaskan cara kerja dari algoritma tersebut! (boleh dalam bentuk *pseudocode* ataupun narasi)

⇒ Data masukan diproses pada layer pertama (*forward propagation*), tergantung fungsi aktivasinya bagaimana, kemudian hasil dari layer pertama diproses kembali pada layer berikutnya sesuai fungsi aktivasi pada layer berikutnya juga, begitu seterusnya sampai layer terakhir dan akhirnya mengeluarkan output (kalau dalam kasus ini 2 output karena 2 klasifikasi). Tidak berhenti sampai disitu, model ini kemudian melakukan *backward propagation* dengan tujuan untuk mendapatkan nilai dW dan db yang nanti akan digunakan untuk memperbarui *weights* dan *biases*. Hal tersebut terus menerus dilakukan sebanyak jumlah iterasi yang ditentukan di awal.

```
layer1      =      Layer(self.X.shape[1],      self.n_neurons,      activation="relu",
learning_rate=learning_rate, init_method=self.init_method)
layer2      =      Layer(self.n_neurons,      len(np.unique(self.y)),      activation="softmax",
learning_rate=learning_rate, init_method=self.init_method)
layer1.setNextLayer(layer2)
```

```
for i in range(1,n_epoch+1):
    layer1.forward(self.X)
    layer2.forward(layer1.output)

    self.predict(layer2.output)
    accuracy = self.getAccuracy()
    loss = self.getLoss(layer2.output, self.loss)

    layer2.backward(self.y) # one hot encoding included in backward
    layer1.backward(self.y)

    layer1.updateParam()
    layer2.updateParam()
    if(i%batch_size == 0):
        print("Epoch:",i,"\t\t","Accuracy:",accuracy,"\t\t","Loss:",loss)
```

2. Bandingkan hasil evaluasi model from scratch dan *library*, bagaimana hasil perbandingannya? Jika ada perbedaan, jelaskan alasannya!

Akurasi Akhir	
<i>Scratch</i>	<i>Library</i>
0.735	0.678
Waktu Eksekusi	
<i>Scratch</i>	<i>Library</i>
0.295 detik	0.649

Berdasarkan data hasil percobaan tersebut, akurasi model buatan sendiri lebih tinggi daripada hasil akurasi dari model *library*. Hal tersebut dapat dikarenakan perbedaan implementasi model karena untuk yang dari *library*, saya menggunakan `MLPClassifier` dari modul *sklearn neural network*. Untuk waktu eksekusinya juga model buatan sendiri lebih cepat dapat dikarenakan saya tidak menspesifikasikan jumlah *neuron* pada model *library*.

3. Jelaskan *improvement* apa saja yang bisa Anda lakukan untuk mencapai hasil yang lebih baik dibandingkan dengan hasil yang Anda punya saat ini! *Improvement* yang dimaksud tidak terbatas pada bagaimana algoritma diimplementasikan, namun juga mencakup tahap sebelum *modeling and validation*.

⇒ Preprocessing : melakukan *dimensionality reduction* karena dengan memilih fitur yang relevan, model dapat menghasilkan hasil yang lebih baik lagi.

⇒ Hyperparameter tuning : komponen *hyperparameter* pada model sejenis *neural network* sangatlah krusial untuk kinerja model, maka dari itu pemilihan hyperparameter yang pas akan meningkatkan kinerja model sehingga bisa mendapatkan hasil yang lebih baik lagi.