

Opdracht 6 - K-Means

Aymane Machrouki en Berke Ozmuk

1. Call tree

```
main()
----import_dataset()
----import_labels()
----normalize_data()
----onehot_encoding_labels()
----split_dataset()
----split_dataset()
----train()
-----estimate_outputs()
-----feed_forward()
-----activation_sigmoid()
-----feed_forward()
-----activation_sigmoid()
-----backpropagation()
-----derivative_sigmoid()
-----derivative_sigmoid()
-----call_update_weights()
-----update_weights()
-----update_weights()
----testing_NN()
-----estimate_outputs()
-----feed_forward()
-----activation_sigmoid()
-----feed_forward()
-----activation_sigmoid()
-----one_hot_comparing()
----testing_NN()
-----estimate_outputs()
-----feed_forward()
-----activation_sigmoid()
-----feed_forward()
-----activation_sigmoid()
-----one_hot_comparing()
```

2. Totaal-uitleg

Aan het begin wordt eerst de dataset geïmporteerd (**import_dataset()**) en de labels geïmporteerd (**import_labels()**). Daarna wordt de data genormaliseerd (**normalize_data()**) en worden de labels gencode naar binary encoded labels (**onehot_encoding_labels()**). Dit wordt gedaan op de onehot encoding manier, omdat de labels strings zijn. En om ze bruikbaar te maken is het nodig om ze te encoden naar binary encoded labels. Daarna wordt de dataset en labels gesplitst in een training set en test set (**split_dataset()**). Dit wordt gedaan op basis van een percentage. Nadat alle data succesvol is geïmporteerd wordt de neurale netwerk aangemaakt (**neural_network()**). Vervolgens worden de training dataset en labels getraind (**train()**). De **train()** functie voert in de range van het aantal epochs eerst een **feed_forward()** propagatie uit per neuron. Dit wordt gedaan met behulp van de **estimated_outputs()** functie. Deze functie schat de outputs van het neurale netwerk aan de hand van de input data van de dataset. Daarna wordt de **backpropagation()** function uitgevoerd per neuron. Als laatste worden alle weights geupdate (**call_update_weights()**, **update_weights()**). Nadat het neurale netwerk is getraind kan er nu getest worden. De test, test de nauwkeurigheid van het neurale netwerk (**testing_NN**). De test zijn gedaan met de Iris dataset. Er wordt getest op de gesplitste dataset en op de gehele dataset.

3. Resultaten

Nadat het neurale netwerk is getraind kan er nu getest worden.

De test, test de nauwkeurigheid van het neurale netwerk (**testing_NN**).

De test zijn gedaan met de Iris dataset.

Er wordt getest op de gesplitste dataset en op de gehele dataset.

De tests zijn gedaan met de volgende specificaties:

- Het neurale netwerk is getraind met 82% van de genormaliseerde data.
- De inputlayer is 4, de hiddenlayer is 8 en de outputlayer is 3.
- De learningrate is 0.1.

De nauwkeurigheid van het getraind neurale netwerk op de test set bij 100 epochs en een learningrate van 0,1:

```
Test results with test data:  
Test results: correct_estimations is: 28 and the precentage is: 100.0%
```

De nauwkeurigheid van het getraind neurale netwerk op de gehele dataset bij 100 epochs en een learningrate van 0,1:

```
Test results with whole dataset  
Test results: correct_estimations is: 144 and the precentage is: 96.0%
```

De nauwkeurigheid van het getraind neurale netwerk op de test set bij 1000 epochs en een learningrate van 0,1:

```
Test results with test data:  
Test results: correct_estimations is: 23 and the precentage is: 82.14285714285714%
```

De nauwkeurigheid van het getraind neurale netwerk op de gehele dataset bij 1000 epochs en een learningrate van 0,1:

```
Test results with whole dataset  
Test results: correct_estimations is: 144 and the precentage is: 96.0%
```