**Classification of Iris Dataset using K-fold Cross Validation**

Vidiskiu Fortino Kurniawan

16/395999/PA/17340

**Introduction**

In this assignment, we will explore 2 new concepts and implement them. First, K-fold validation, and secondly the different splitting of dataset i.e. training data and validation data. K-fold cross validation is a method to estimate the performance a model. In this method, dataset is “split” into K different segments. The process of data training and validating will be done k times where in each iteration the segments will be rotated to be the training and validation data. Whereas the training and testing of the model is similar with the previous assignment, it is simply said that the previous process was done k amount of times with different composition of training and validation records, hence the update on the previous assignment’s code. Validation data is distinct from testing data, in a sense that we will still have the real category of the given data and is mainly used for evaluation purpose. Moreover, validation data is different from training data in a way that it will not update the weights of the model when it is used to query an answer. Hence, this assignment will implement the classification of Iris dataset using SLP, augmented with k-fold cross validation to further evaluate the model.

**Method**

* Python programming language will be used to implement this process.
* Dataset used is Iris dataset with 150 records evenly split by loop of 5 into groups of 30 in each segment.
* Input nodes = 4, Hidden nodes = 0, Output nodes = 2, Epochs = 100, Learning rate = 0.1, K = 1
* Initial weights, same as task 1 for evaluation purposes



Figure 1. Initial Weights

* 5 will be used as k, meaning that:
  + 100 epochs of 5 iterations for the entirety of the program
  + Dataset will be segmented into 5 segments, and rotated accordingly
  + 120 training data and 30 validation data in each iteration

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Segment 1 | Segment 2 | Segment 3 | Segment 4 | Segment 5 |
| 30 | 30 | 30 | 30 | 30 |
| 120 | | | | 30 |
| Training data | | | | Testing data |

Table 1. Visualization of data splitting

* The data splitting will be the combination of the 5 segments, hence be as follow:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| K-Fold | Validation data | Training data | | | |
| K = 1 | Segment 1 | Segment 2 | Segment 3 | Segment 4 | Segment 5 |
| K = 2 | Segment 1 | Segment 2 | Segment 3 | Segment 4 | Segment 5 |
| K = 3 | Segment 1 | Segment 2 | Segment 3 | Segment 4 | Segment 5 |
| K = 4 | Segment 1 | Segment 2 | Segment 3 | Segment 4 | Segment 5 |
| K = 5 | Segment 1 | Segment 2 | Segment 3 | Segment 4 | Segment 5 |

Table 2. Visualization of data splitting in each iteration

* Other methodologies followed previous assignments.

**Experiment**

The following is the graphical result of the experiment followed by the tables:

|  |  |  |
| --- | --- | --- |
| K-Fold | Accuracy Graphs | Error Graphs |
| K = 1 | C:\Users\vidis\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Figure_1.png | C:\Users\vidis\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Figure_2.png |
| K = 2 |  | C:\Users\vidis\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Figure_4.png |
| K = 3 |  | C:\Users\vidis\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Figure_6.png |
| K = 4 |  |  |
| K = 5 |  |  |
| Average | C:\Users\vidis\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Figure_13.png | C:\Users\vidis\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Figure_14.png |

The following shows further tabular information in each graphs:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| K-fold | Min Error | Avg Error | Max Acc | Avg Acc | Avg Val Error | Avg Val Acc |
| K = 1 | 0.213 | 0.246 | 0.858 | 0.821 | 0.37 | 0.657 |
| K = 2 | 0.203 | 0.235 | 0.875 | 0.837 | 0.371 | 0.634 |
| K = 3 | 0.215 | 0.252 | 0.85 | 0.82 | 0.339 | 0.667 |
| K = 4 | 0.204 | 0.24 | 0.875 | 0.832 | 0.391 | 0.653 |
| K = 5 | 0.205 | 0.243 | 0.875 | 0.824 | 0.367 | 0.653 |
| Avg K | 0.208 | 0.243 | 0.86 | 0.827 | 0.073 | 0.653 |

We observe the linearity of the graphs between the training data evaluation and validation data evaluation. This shows that there aren’t any case of overfitting in all k cases. It is observed that the highest average accuracy occurs when k = 2, although it wasn’t much different across all the k-value. Overfitting occurs when the model is trained so well to the training data that it over fits the training data and causes false evaluation of the validation data. In this case, there is no increased error as the error of the training data increases, and so does for the accuracy. Hence, no overfitting occurred during the course of the experiment.

**Documentation**

The code listing is available in the following github link :

https://github.com/Vidiskiu/SLP\_iris\_dataset/tree/k-fold-validation-implementation

Additional processes have been explained in the inline codes.

**Conclusion**

The use of k-fold validation allows further evaluation of the SLP model. The results of the SLP model is heavily dependent on the distribution of data as well as the hyper parameters that was assigned to the model.