Aireen assignment classifier report

Introduction

We have used a DNN classifier from TensorFlow library for the classification of the data provided by the Aireen company assignment (hereinafter mentioned as the “data” and the “assignment” respectively).
For the same data we have tried not only a DNN classifier but also a sequential model for classification of the data but the results were unexpectedly low (by the level of magnitude) compared to those produced by the tensorflow DNN classifier. For this reason we show only DNN classifier results.

Training:

Brief data description:
Each file in the data contains several ‘polygon_object’ features:

- ‘class_name’
- ‘class_score’
- ‘num_points’
- ‘points’
- ‘coco-mask‘
- ‘uuid’

For training purposes we have considered only ‘class_name’ and ‘class_score’ features

We have extracted each feature influencing ‘DR’ or ‘No DR’ class possibility of the result from the data. We have considered the following values of the “class_name” feature: `{'ma', 'hma', 'he', 'od', 'fov', 'cw'}` - these variables were used as numeric classes for the input vector (each numeric input represents the count of the class occurrence per input).

The classifier was trained several times with the 3 following variables of the training setup:

- the batch size - the batch size of the training data repeated.
  - `(32, 64, 128, 256)`
- the network layers - organization of particular hidden layers of the classifier
  - `([20, 10], [40, 20], [10, 5, 2], [20, 10, 5], [40, 20, 10])`
- the classes score flag - the indicator whether or not we have considered “class_score” value in the data. If so, each of the classes was considered if and only if its “class_score” rank was over 70%.

Training results

As our experiments have shown, the batch size does not influence the result much (higher values are slightly better). The network layers exhibit the best results for the [40, 20] and [40, 20, 10] setup, thus we can consider the additional third layer as not useful. However, when class scores were considered the average accuracy of the network was higher byt approximately 2-3% compared to the setup where the class scores were ignored. The average accuracy for the best setup is 82%.

In the appendix of this report we provide graphs and particular (average) values of the classifier.

Appendix

Below we show evaluation results of several different inputs.
Each input consists of the average accuracy of the network (average of 15 training trials of the same classifier setup).

Graphs

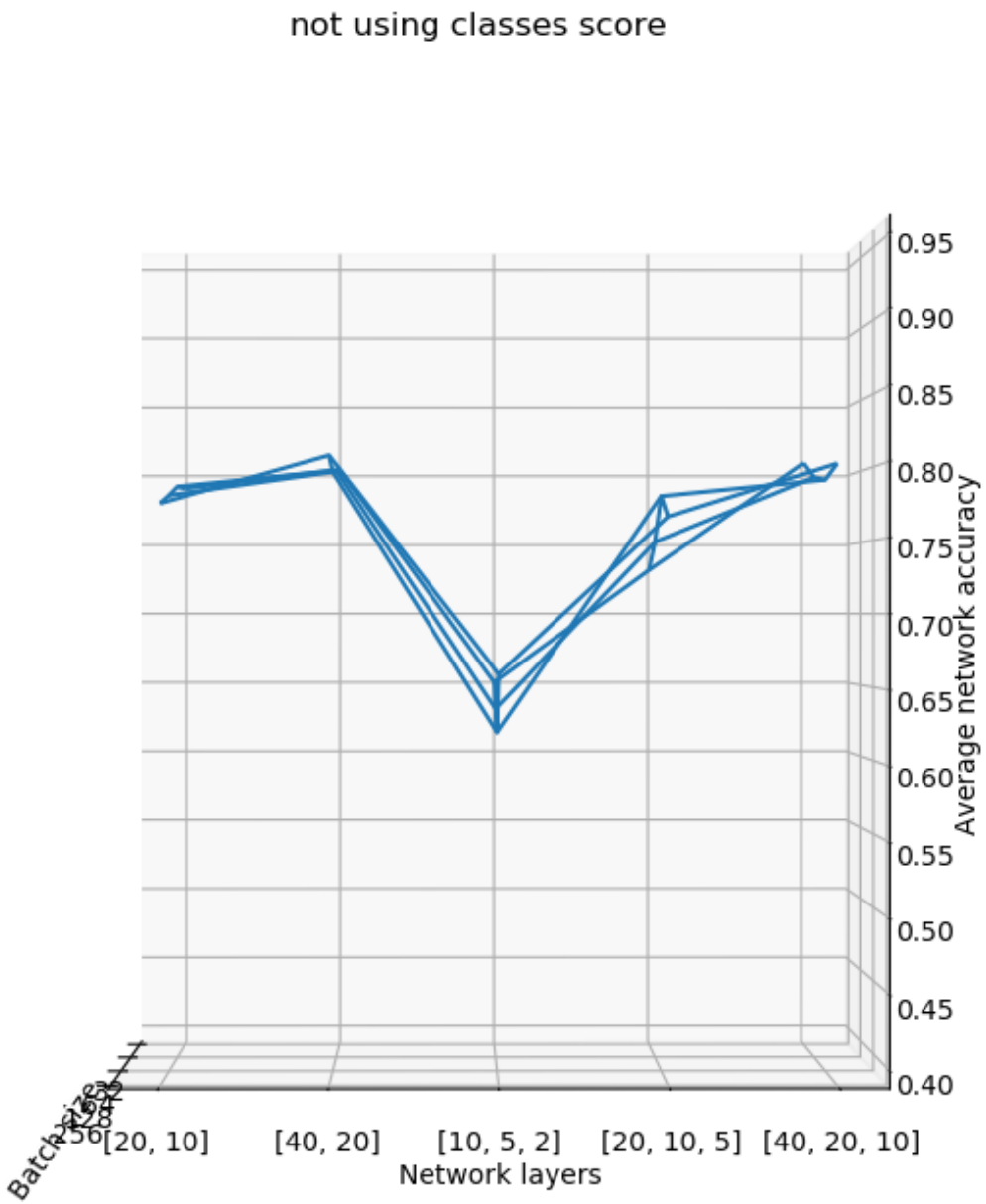

Image 1: Training results of the classifier when not using class score. (Focus on network

layers)

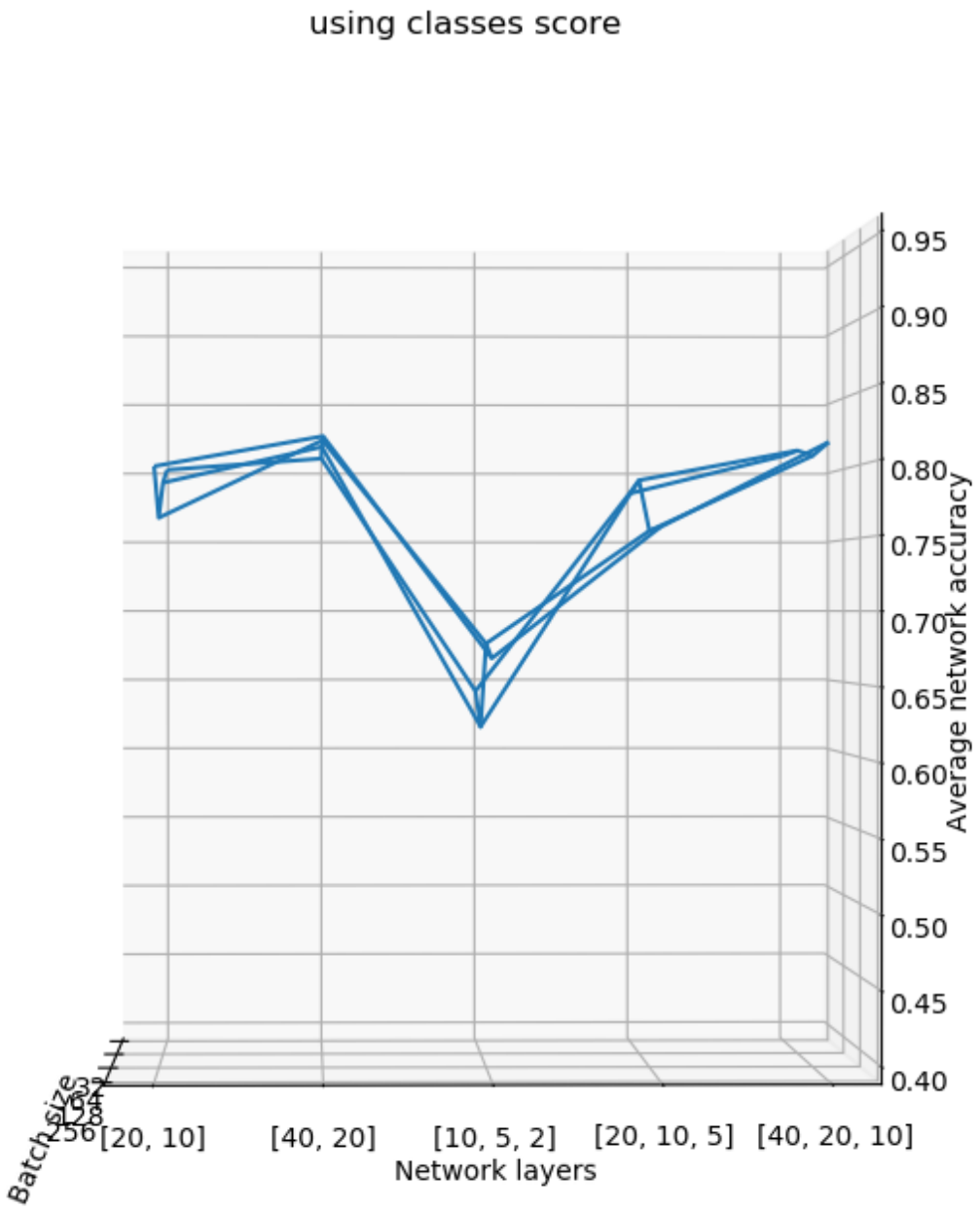

Image 2: Training results of the classifier when using class score. (Focus on network layers)

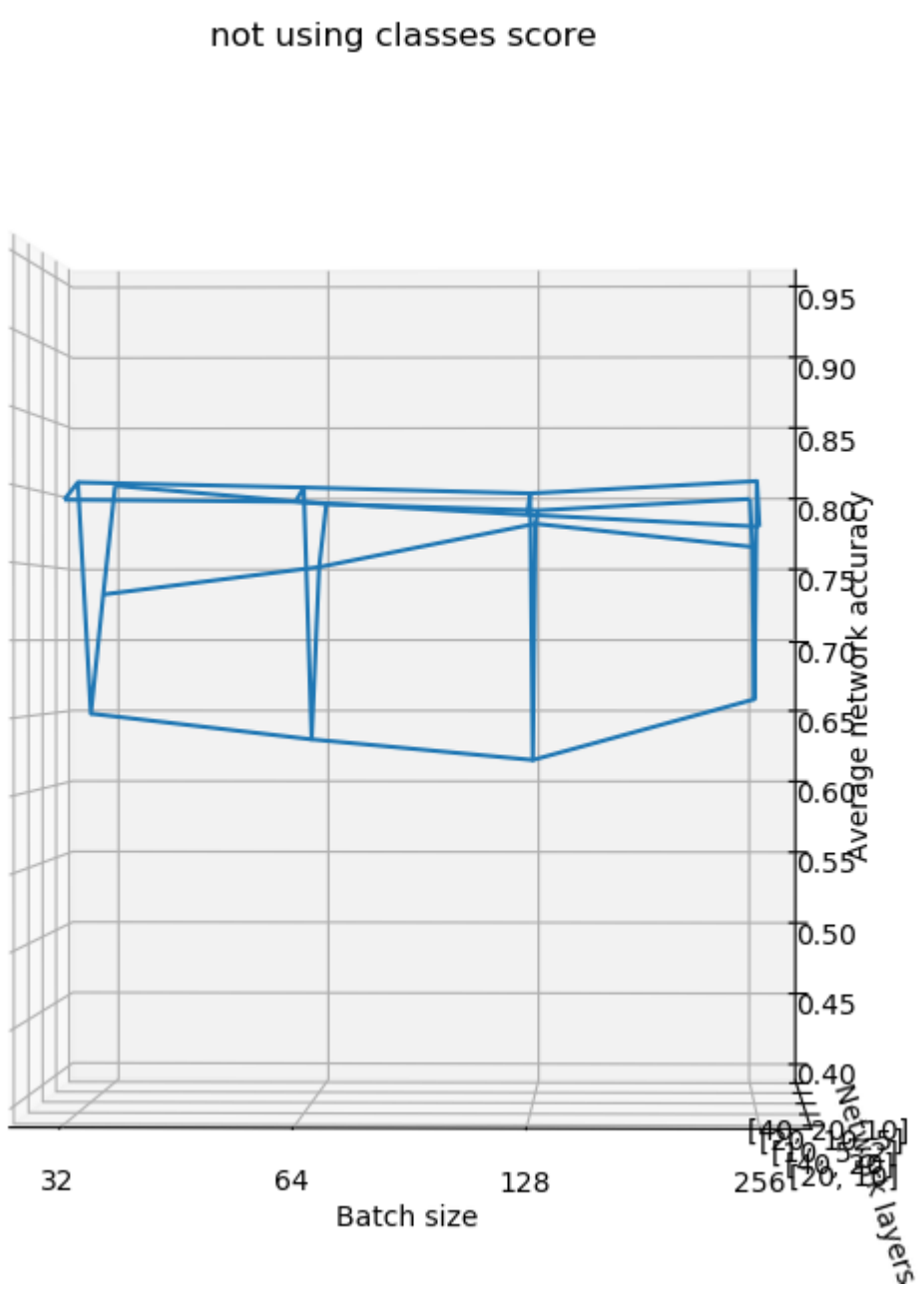

Image 3: Training results of the classifier when not using class score. (Focus on Batch size)

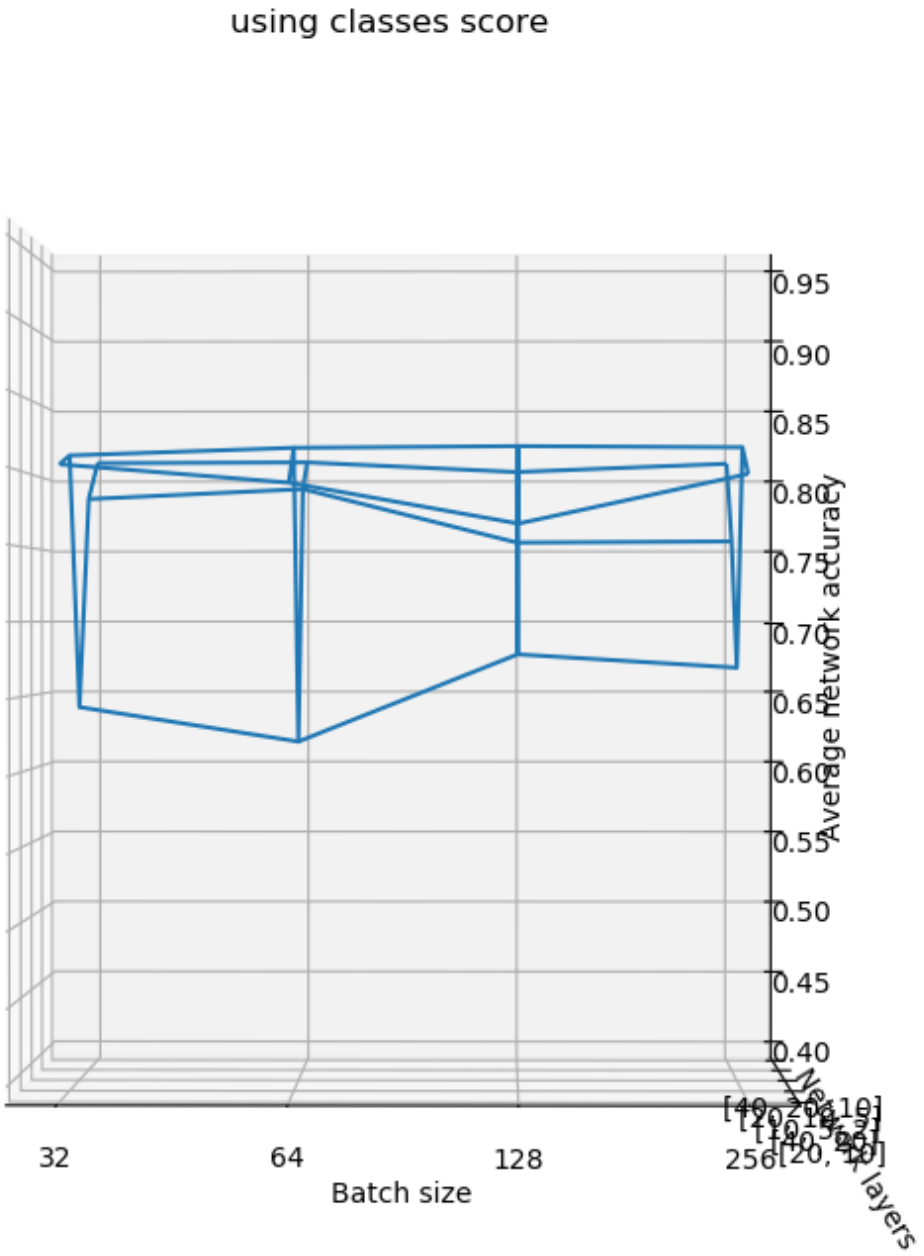

Image 4: Training results of the classifier when using class score. (Focus on Batch size)

Values

In both matrices rows represent the network layers values:

- `([20, 10], [40, 20], [10, 5, 2], [20, 10, 5], [40, 20, 10])`

Columns represent the batch sizes.

- `(32, 64, 128, 256)`

Graph values for the setup where class scores were not considered:

([[0.79043825, 0.78911023, 0.78061089, 0.77290837],
[0.80398406, 0.80053121, 0.79654714, 0.80451527],
[0.65046481, 0.63320053, 0.61938911, 0.66029216],
[0.7314741 , 0.75033201, 0.78007968, 0.76414343],
[0.80876494, 0.79575033, 0.79123506, 0.79920319]])

Graph values for the setup where class scores were considered:

([[0.80212483, 0.79017264, 0.76361222, 0.79575033],
[0.81035857, 0.81540505, 0.81646746, 0.81567065],
[0.64196547, 0.6185923 , 0.67782205, 0.6687915],
[0.78512616, 0.79203187, 0.75484728, 0.75564409],
[0.8124834 , 0.81301461, 0.8061089 , 0.81195219]])