Assignment 2: Machine Learning Project

Classify Images of Road Traffic Signs

COSC2673 Machine Learning

Nguyen Ngoc Khanh Linh *S3927588*
Nguyen Duc Quang *s3927198*

Table of Contents

I. Machine learning algorithms considered 2

1. Multi Layers Perceptron 2
2. Convolutional Neural Network model 2

II. Why did you select these approaches? 2

III. Evaluations of the performance of trained model(s) 4

1. Evaluation of Traditional Neural Network model with SGD optimizer 4
   a. On shape classification 4
   b. On type classification 5
2. Evaluation of Convolutional Neural Network model with RMSprop Optimizer 6
   a. Shape classification 6
   b. Type classification 7

IV. Your ultimate judgement with supporting analysis and evidence 7

I. Machine learning algorithms considered

1. Multi Layers Perceptron

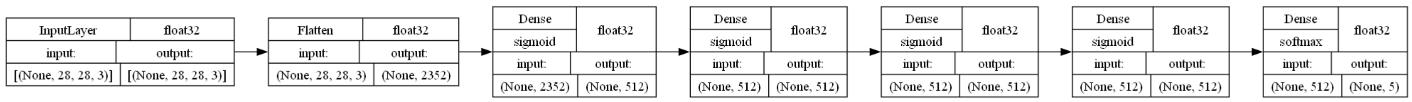


Figure 1.1. Architecture for the base model of MLP

Figure 1.1 describes the architecture for the base model of MLP. In this architecture, every node from the previous layer is connected to every node from the next layer. The parameters of the model are trained using feed forward and back propagation. Nodes in the hidden layers are activated with Re LU activation function and nodes in the output layer are activated with the SoftMax activation function. The input layer represents the flattened pixels on an image, each node represents the grayscale value of a corresponding pixel. The output layer nodes correspond to each class of the classification problem. From this point, this structure will be referred to as the Fully Connected (FC) layer.

2. Convolutional Neural Network model

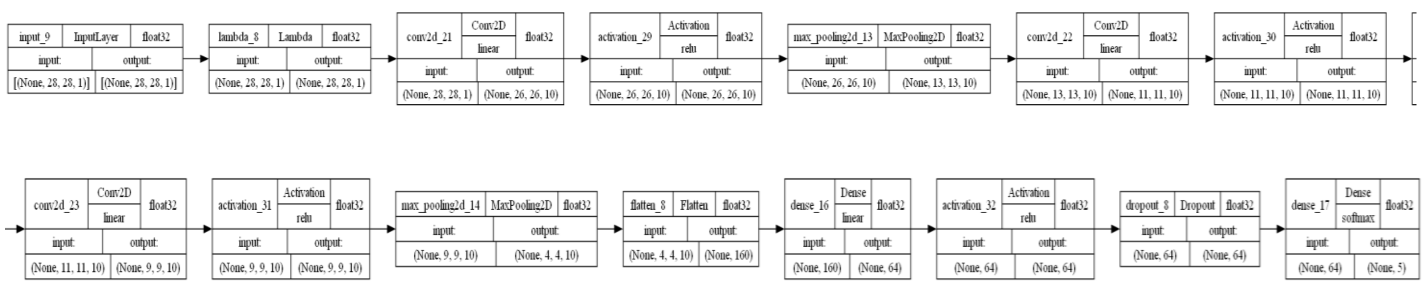


Figure 1.2. Architecture for the Convolutional Neural Network model

Figure 1.2 describes the architecture for the more advanced version of MLP, which incorporates convolutions and pooling in the process of extracting and learning advanced features from the input. The convolution and pooling layers come before the FC layer. The convolution layers act as feature detection and the pooling layers reduces the dimension of the input to eventually pass the parameters to the FC layer.

II. Why did you select these approaches?

Advantages of MLP:

- Can be used in various types of problems.
- Can create complex models with the help of non-linear transformations.

- Can generate new features via the connections amongst nodes in the network.

Advantages of CNN for visual recognition:

- Features detection:
    - CNN layers in the network provide filters for the network to learn different features of the image.
    - Different types of filters can recognize different types of features based on their effect on the image during the convolution operation.
- Dimension reduction for higher level feature recognition:
    - Pooling layers provide different options to reduce the dimension of the images, such as max-pooling and average-pooling.
    - Pooling allows the model to focus on important features of the image, by omitting under-represented pixels after the convolution layers have made the important features pop up.

Advantages of Re LU:

- Computationally efficient
- Decrease the chances of encountering vanishing gradients.

Advantages of F1 score and class weight balancing:

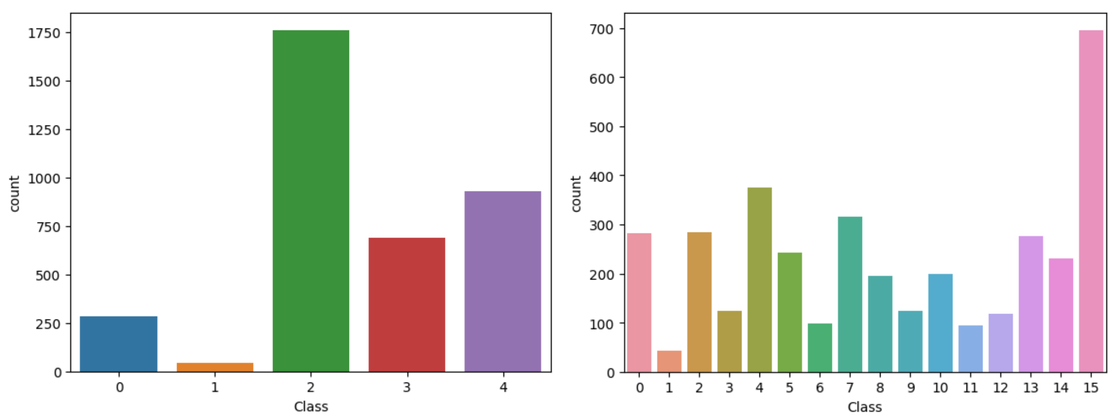


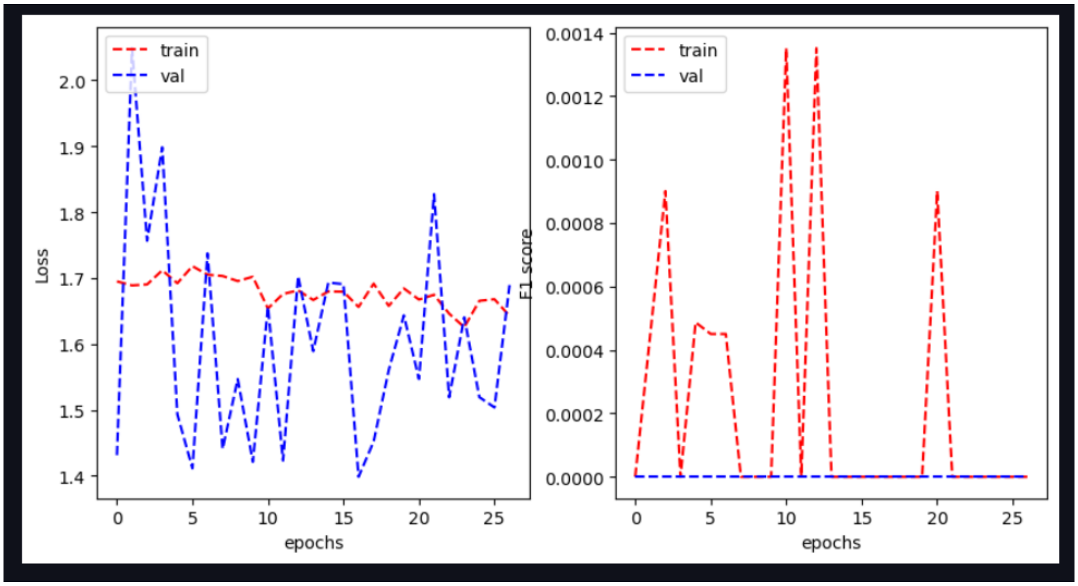
Figure 2.1

- Suitable for datasets with class imbalance:
    - As can be seen from the two graphs, two tasks have high class imbalances, calling for a solution to balance their weights. To solve this, the final model would use F1 score as the evaluation metric and the class weights are recomputed.

III. Evaluations of the performance of trained model(s)

1. Evaluation of Traditional Neural Network model with SGD optimizer

a. On shape classification



Plot of Loss and F1-score of Neural Network model on train and validation datasets (shape classification)

```
model.evaluate(test_generator, batch_size=len(test_df['Class']))
✓ 0.5s
23/23 [==============================] - 0s 15ms/step - loss: 1.6768 - f1_m: 0.0000e+00 - accuracy: 0.0565

[1.6768113374710083, 0.0, 0.056521739810705185]
```

Score of Traditional Neural Network model on independent dataset

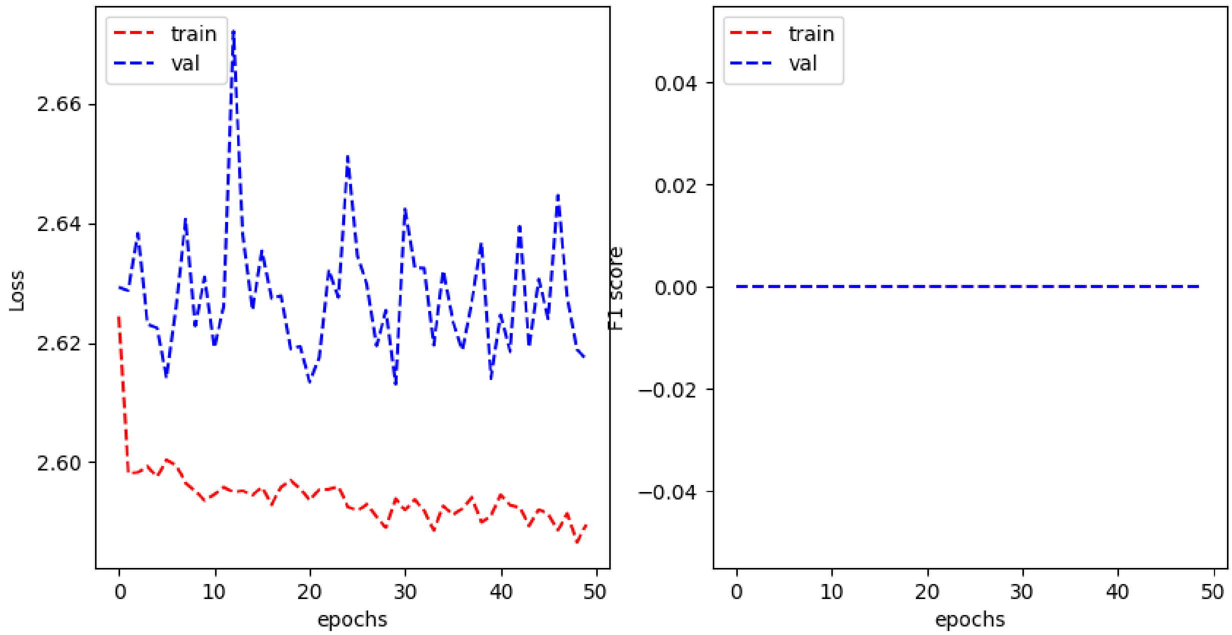
It can be seen from the plot that the scores of NN model on train and validation datasets are oscillating with:

- On train dataset, loss oscillating between 1.6 and 1.7, and f1 score oscillating between 0 and 0.0014.
- On validation dataset, loss oscillating between 1,4 and 2 and f1 score is nearly 0.

The score suggests that the baseline model has considerably low scores on both train and validation data.

When tested on independent test data, the model achieved relatively low scores as well (as indicated from the screenshot). The accuracy is 0.056.

b. On type classification



Plot of Loss and F1-score of Neural Network model on train and validation datasets (types classification)

- Oscillation is witnessed in both training loss and validation loss with a higher extent on validation (blue dashed line).
- There is no significant downward trend in the loss scores, suggesting that the NN model is not learning well through epochs.
- F1-score is a straight line at 0, indicating that the baseline model does not make any correct predictions for the positive class, which is a critical issue worth investigated.

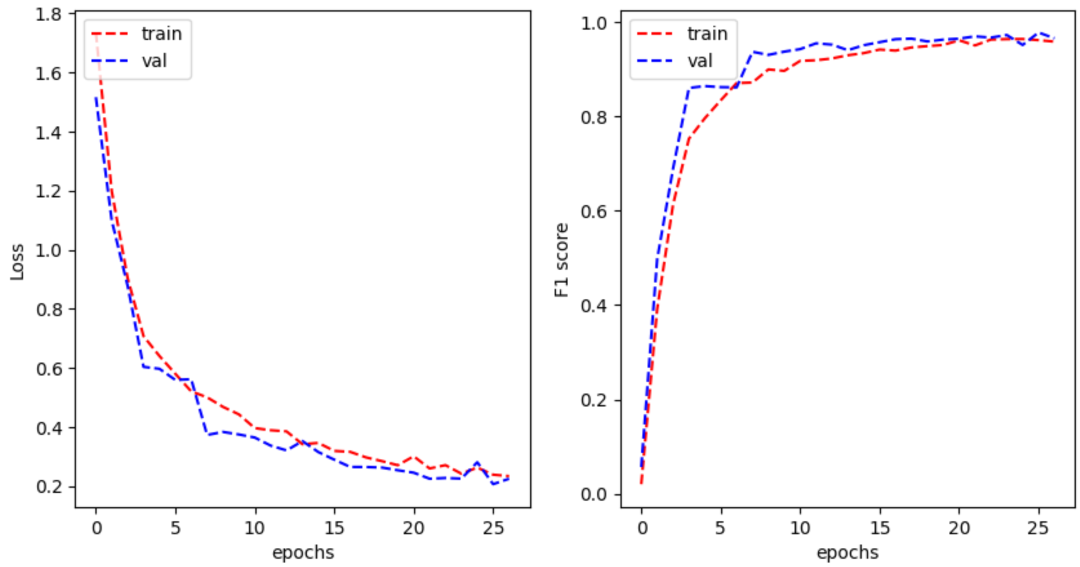
```
model.evaluate(test_generator, )
✓ 0.8s
23/23 [==============================] - 1s 30ms/step - loss: 2.7868 - f1_m: 0.0000e+00 - accuracy: 0.0587

[2.7867813110351562, 0.0, 0.05869565159082413]
```

When tested on independent test data, the model achieves a critically low accuracy (=0.0587). F1- score is 2.7868, indicating that the predictions are deviating far from the actual targets. Overall, the traditional neural network model does not seem to classify the types for traffic sign well.

2. Evaluation of Convolutional Neural Network model with RMSprop Optimizer

a. Shape classification

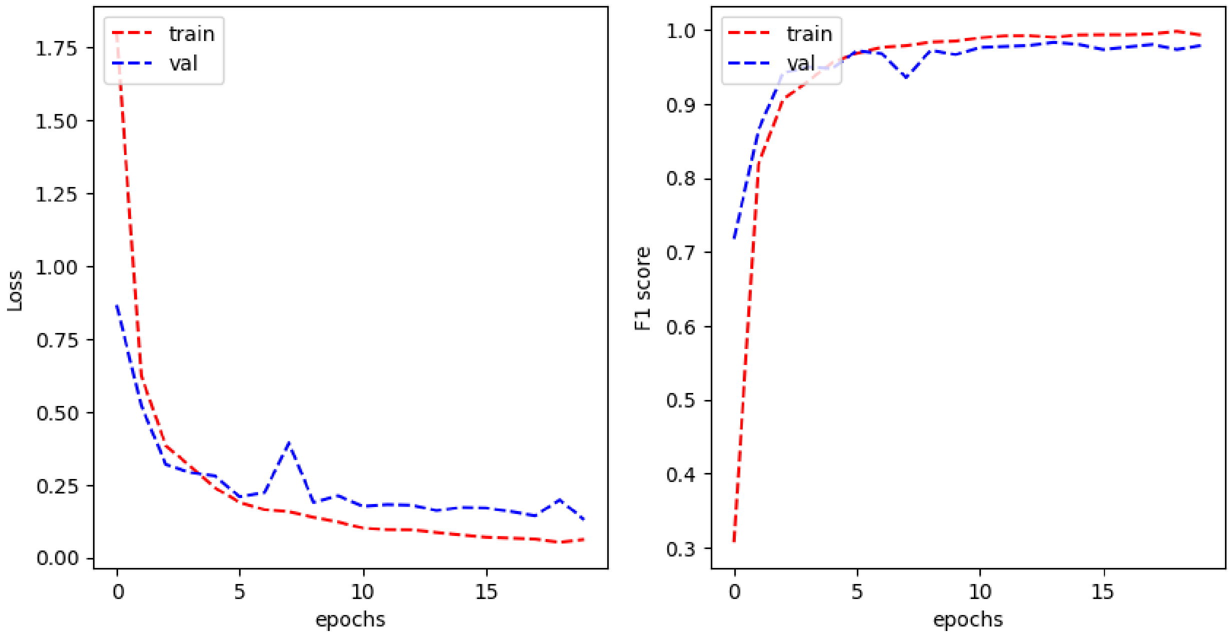



```
model.evaluate(test_generator, batch_size=len(test_df['Class']))
✓ 0.4s
23/23 [==============================] - 0s 12ms/step - loss: 1.1610 - f1_m: 0.7723 - categorical_accuracy: 0.7717

[1.1610491275787354, 0.772275984287262, 0.77173912525177]
```

- Training loss and validation loss both improved and reached 0 after 20 epochs. There is a small fluctuation in validation loss after epoch 20 but overall, the score still remains relatively low.
- F1-score for both training and validation reach 1 after 10 epochs.
- When tested on independent dataset, the model achieve a good accuracy of 0.7717, indicating its ability to predict up to 77% of the targets correctly. Loss score is 1.16 which is smaller compared to the previous model. F1-score on test data is 0.77, emphasizing an improvement in model performance.

b. Type classification



Loss and F1-score plot on type classification

```
model.evaluate(test_generator, )
✓ 0.9s
23/23 [==============================] - 1s 29ms/step - loss: 5.6731 - f1_m: 0.2075 - categorical_accuracy: 0.4543

[5.673060894012451, 0.20746538043022156, 0.4543478190898895]
```

Result on independent test set (type classification)

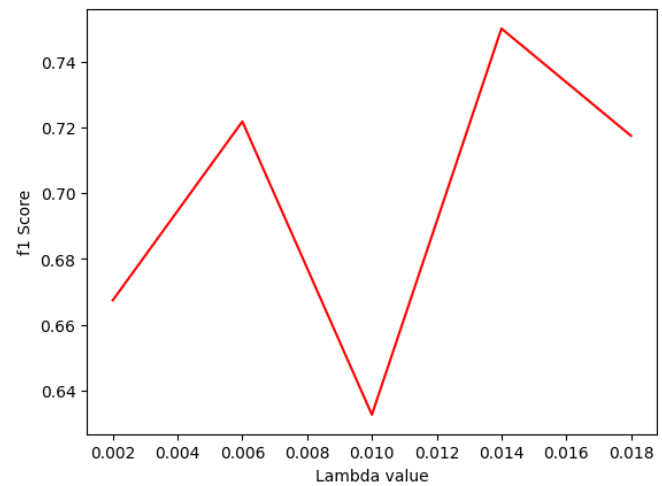
- Training loss and validation loss have been improved compared to those of the baseline model. The training loss decreases considerably and stays stable at nearly 0 after 10 epochs, with validation loss also shares the same pattern and stays around 0.2 with much less oscillations compared with the previous model. The model is learning well through epochs.
- F1-scores also increase sharply and reach around 1 after 10 epochs.
- Compared to the previous model, the accuracy of CNN model on independent test set has improved critically. It achieves an accuracy score of 0.4543, indicating it predicts 45% of the data correctly. However, the loss score is higher compared to the baseline model.

IV. Your ultimate judgement with supporting analysis and evidence

- The ultimate model chosen is Convolutional Neural Network with:

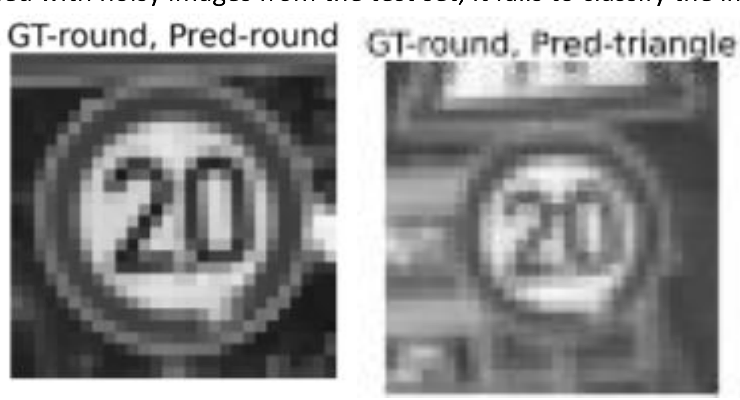
- RMSProp optimizer:
  - As oscillations in scores are observed during training with SGD optimizer, RMSProp is used to substitute as it can help stabilize the training process by adjusting to the gradient.
  - During the model developing process, a tuning of learning rate does not show much impact on model's improvement. RMSProp can be effective as the gradient scaling can help with the optimization process regardless of the learning rate.
- Regularization Lambda value of 0.014

  Tuning lambda value would help to prevent overfitting during the model learning.



  The plot shows f1-score of the ultimate models on test data with different regularization lambda values. The plot suggests the lambda with value of 0.014 will achieve the highest f1 score on the test.

- The Convolutional Neural Network model is observed to be overfit, due to the fact that the training and validation data do not deviate too much from each other. Therefore, when the model is introduced with noisy images from the test set, it fails to classify the images correctly.



The two images above are labeled as round. However, since the left image has some triangle edge on top, the model mistakenly classifies it as triangle. This goes to show the lack of variance of data being used to train and validate, leading to heavy overfitting.

- In real-world settings, the application can be applied into self-driving cars to enhance safety and improve navigation as well as routing. Detection of speed limit would also allows self-driving cars to comply with traffic rules. In order to apply the application into real-world settings, image processing, machine learning/ deep learning and sensor are required to develop the high-performance application. Challenges are laid on the accuracy of the models with a variability of traffic signs, require further investigation and tuning process for the model. Real-time processing performance is also another challenge to be considered.

Video presentation link: https://rmiteduau-my.sharepoint.com/:v:/g/personal/s3927198_rmit_edu_vn/EfdO0weXKQhNqUZoMYrb4qEBAtqmwJY0IyXBkdeXTy_pqg?e=bWfsfL&nav=eyJyZWZlcnJhbEluZm8iOnsicmVmZXJyYWxBcHAiOiJTdHJlYW1XZWJBcHAiLCJyZWZlcnJhbFZpZXciOiJTaGFyZURpYWxvZy1MaW5rIiwicmVmZXJyYWxBcHBQbGF0Zm9ybSI6IldlYiIsInJlZmVycmFsTW9kZSI6InZpZXcifX0%3D