COSC2667 – Computational Machine Learning

Assignment II – Machine Learning Project

Traffic sign classifier

Jigar Mangukiya | S3715807

Semester 1, 2020 Master of Data Science

Royal Melbourne Institute of Technology

Contents

1. Introduction

- Purpose of the study was to develop classifier models that can identify different shapes and types of traffic signs in Belgian traffic signs dataset[1]. Focus of the study was on making analytical design choices to create an end-to-end ML project going through all major phases like Data loading and exploration, pre-processing, modelling, and evaluation.
- In the initial phases of the project the efforts were to select appropriate ML algorithm for developing the classifiers and after deciding to go ahead with Neural Networks & Deep learning, Subsequent steps of Data Loading and initial exploration were undertaken. Multiple models were tested, and appropriate hyper parameter tuning methods were employed to identify right parameter set to get best performance. With the best fit model, evaluations were made on model with multiple performance metrics to understand characteristics of the model.
- After the evaluation based on testing datasets, independent evaluations were done by forming mini testing datasets from multiple open and available datasets to understand how model performs against unforeseen data.
- This brief report highlights key decisions and rationale behind them that were made through out the course of this project.

2. Algorithm Selection

- Classification is a supervised machine learning technique, that classifies new data into one of many pre-defined classes. Prior to making the classification judgements, the classifier needs to be trained with labelled data.
- Classification is highly researched ML technique and used in many deployed solutions nowadays like in robotics, banks and vehicles and it is becoming increasingly common at other places.
- For this particular image classification problem, there were multiple algorithm alternatives which can work, like Support Vector Machines (SVM), Decision trees and Random forests. However, all of these are traditional ML algorithms, which, to work with image data, will require tremendous amount of feature engineering to craft new features from image, due to their lack of ability to work directly with images. In essence, you will have to describe most prominent features of the images to any of these algorithms in order for them to learn.
- However, Neural Networks and CNN are highly capable of working with large feature vectors like images and can learn the features using convolution layers. These family of algorithms also generalise well with unseen image data and are capable to fit the training data with out over-fitting by providing finer control on training process with hyper parameters.
- CNN was found to be most widely recommended algorithm for image classification problem during the research and has been found to consistently out-perform other traditional ML algorithms.
- Hence, this project utilised simple MLP neural network algorithm to set a baseline and utilised CNN to run against the baseline.

3. Dataset Description

- Provided dataset was formed as a portion of larger Belgian Traffic Sign (BTS) Dataset. Dataset is arranged in 2 way labelling, with “shape” labels being the primary directories in dataset folder and “sing” labels being subdirectories of primary “shape” directories. All the images were collected along with both “shape” and “sing” labels in dataframe of 3699*3 dimension.
- Original dateset can be sourced from [1].

4. Data Exploration

- Dataset contains grayscale images of 28 x 28 pixels. figure 4.1 is an example image. Image is of “hex” shape and “stop” sign. Smaller image dimensions will help ease the training process of models.
- To gain better understanding of data, class frequency distribution was examined for the “shape” and “sign” labels.
- Dataset was found to be imbalanced as can be seen in figure 4.2.
- “hex” shape in particular, had very small number of examples which would not be sufficient for training, validation and testing set to reflect true performance characteristics of model for “hex” category.
- According to sign target variable, “stop” had small number of images as well, which corresponds to “hex” data points as “hex” shape has only one type of “sign” images which are the “stop” signs.

Figure 4.1 Sample image from dataset

- There were two alternatives to rectify this, oversampling and performing random augmentations on the fly with Keras ImageGenerators or creating new augmented images from original images and storing them. Later option was utilised in interest of consistency across different invocations of Generators. 200 augmented images with different shear, rotation and brightness were generated and added to “hex/stop” directory.
- Other classes which had as little as 100 images were not augmented during this stage but could’ve been augmented in same manner if model performance reflected insufficient examples.

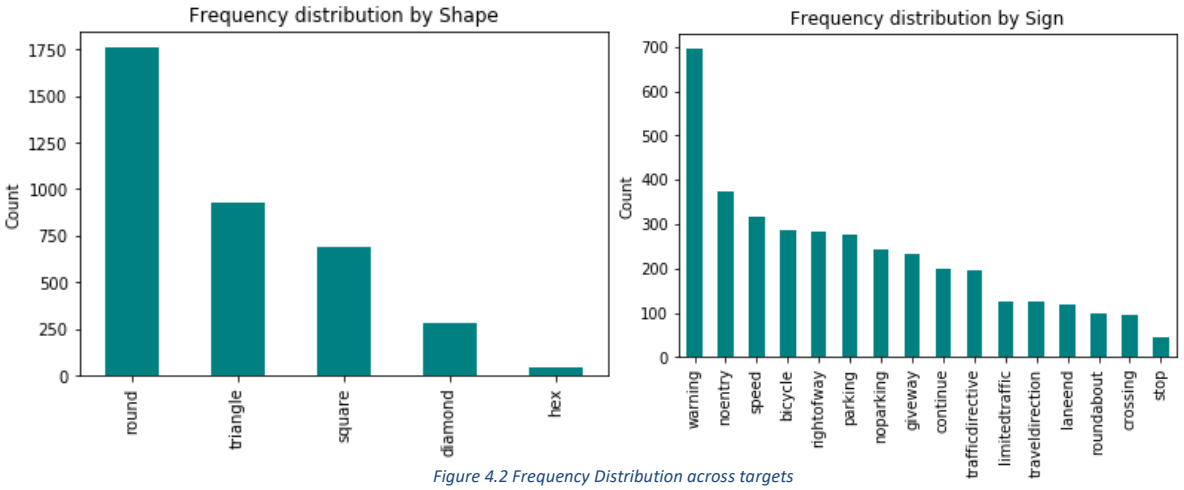


Figure 4.2 Frequency Distribution across targets

- This was done, not to forcibly balance the dataset and increase size of training and validation sets to keep the run time practical to perform tuning.
- For rest of the cases, dataset was already pre-process to an extent and all images were of same size (28 * 28) and grayscale so no other cleaning or pre-processing steps were necessary.

5. Approach & Model Architectures

Baseline NN model

- For both, shape and sign, targets similar flow was observed, where in, initially a simple neural network with one input layer (input size (28,28,3)), one fully connected hidden layer (size 64) and one output layer (size 5 or 16, parameterized) was trained, validated and tested to set up a baseline score and understand how the model behaves.

- ImageGenerators were used each of the training, validation, and testing phase, for efficient memory utilization. As this was just baseline model, image augmentations were not performed at this stage in training ImageGenerator instance.
- Hidden layer uses 'ReLU' activation function, as the images are scaled in (0,1), there are no negative values and non-saturation gradient, which can ultimately result in faster convergence of optimisers.
- Output layer of NN uses 'softmax' activation function as this is a multi-class classification problem and softmax outputs probabilities that sums to 1. Which makes it most ideal output layer activation.
- SGD optimizer was used for efficiency and faster convergence with an empirical learning rate.
- Due to the dataset being imbalanced and it being a multiclass classification problem, the algorithm monitors accuracy, precision and recall to avoid penalties to smaller classes in the sample.

CNN model

- Next, a CNN model was created in which number of (Convolution + Convolution + pooling) layers were parameterized to allow it to be tuned, along with other parameters. For a dry run, 3 x (Convolution + Convolution + pooling) layer model was trained with empirical parameters and augmented training images to classify shape. The architecture is loosely based on VGG-13 model as described by [*Simonyan et. Al][2015][3].* However, Conv3-512 layers (marked with cross) are not implemented in this project's model to keep running time practical and number of Conv2, Conv2 & MaxPool as well as FC layers are parameterised to avoid over-fitting the significantly smaller sample image size current model will be dealing with.

Figure 5.1 Original VGG 13 Architecture, img source – [4]

- The model uses 'ReLU' activation function throughout the network except in the output layer where, 'softmax' is used to get the probabilities that can be converted to labels.
- Optimizer is changed from the base SGD to RMSProp, as it was found to be more suitable for deep networks with its adaptive learning rate. Hyper parameters are determined by tuning.
- This model notes accuracy, precision and recall as well, as dataset is multiclass imbalanced one. (It was found that accuracy was in fact categorical accuracy, which Keras picked by itself form multiple classes)
- Dropout was introduced in CNN as baseline was slightly over-fitting.
- Both models use categorical_crossentropy as loss function as the generators are using 'categorical' as class which encodes the labels in one-hot mode. (otherwise it would be sparse_categorical_crossentropy).

6. Model training, tuning and evaluation

- Train:validation:test split – (64,16,20) for baseline – decided from what general research prompted to start with, due to good performance, did not have to revisit.

- Sampling – Through out the project, stratified sampling was used to make sure that all the classes are represented as equal proportion of their frequency in samples. (this prompts looking at macro average in evaluation section)
- Both baseline models were trained with non-augmented training data which was 64% stratified sample with shape and sign class respectively for shape and sign classification.
- Shape baseline model was lightly over-fitting the data with training accuracy 100 % and validation and testing accuracy around 94%. This prompted use of dropout and augmentation in the CNN model.
- CNN model in dry run performed much better with 3 x (Conv2 * Conv2 * Maxpool) layers and empirically provided parameters. With training accuracy 0.99, validation accuracy 0.98 and testing accuracy 0.98. Over-fitting was dealt off by the dropout of (0.125) in conv layer.
- After dry run, both models were put for hyper parameter tuning.
- RandomizedSearchCV() method was used with 20 models with 3 fold cross validation for shape targets and 10 models with 3 fold cross validation for sign targets. 80% data was fed to tuners as there was no need for separate validation set due to CV.
- [As GridSearchCV was not practical (took too long) & unnecessary (models were doing quite good)]
- [*When evaluating this assignment, this step may take too long so the results are commented for your reference*]
- Due to very small image size, smaller kernels and filters were able to achieve great results.

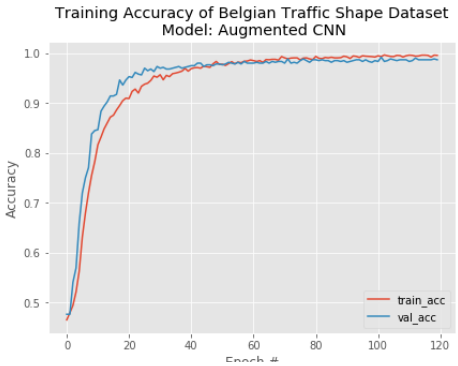
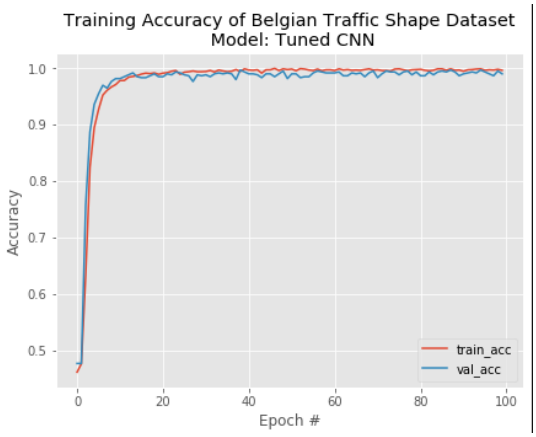
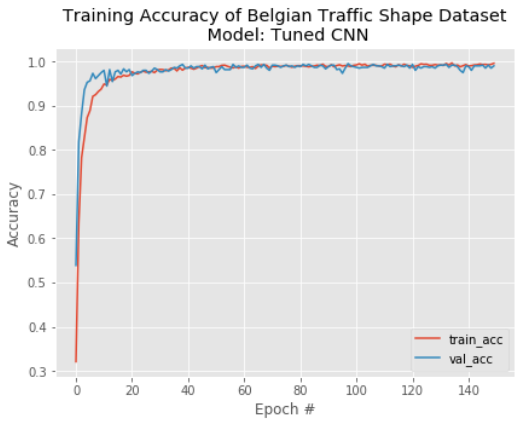


Figure 6.1 Accuracy plots for shape target



	precision	recall	f1-score	support
diamond	1.00	0.96	0.98	56
hex	1.00	0.78	0.88	9
round	0.99	1.00	0.99	352
square	0.99	1.00	0.99	138
triangle	0.99	0.99	0.99	185
accuracy			0.99	740
macro avg	0.99	0.95	0.97	740
weighted avg	0.99	0.99	0.99	740

Figure 6.2 Accuracy plot and classification report of test setfor Shape target



	precision	recall	f1-score	support
rightofway	1.00	1.00	1.00	57
stop	1.00	1.00	1.00	40
bicycle	0.90	1.00	0.95	19
limitedtraffic	0.98	1.00	0.99	46
noentry	1.00	1.00	1.00	24
noparking	0.92	0.92	0.92	25
roundabout	0.96	0.96	0.96	75
speed	0.96	1.00	0.98	48
trafficdirective	1.00	1.00	1.00	55
traveldirection	1.00	0.96	0.98	56
continue	1.00	1.00	1.00	20
crossing	1.00	1.00	1.00	63
laneend	1.00	0.89	0.94	9
parking	0.93	1.00	0.96	39
giveway	0.96	0.88	0.92	25
warning	1.00	0.98	0.99	139
accuracy			0.98	740
macro avg	0.98	0.97	0.97	740
weighted avg	0.98	0.98	0.98	740

Figure 6.3 Accuracy plot and classification report of test set for Sign target (Typo in the title of right plot)

7. Independent Evaluation

- To test how the trained model performs with other datasets, the models were used to make classifications on other available traffic sign datasets. 2 Datasets were collected
  1. German traffic sign dataset [5]
  2. Chinese traffic sign dataset [6]
- For both dataset, a small sample was selected as per the target categories manually, keeping in mind to have enough variation in the datasets to thoroughly test the model. It was expected that trained models will do well with German dataset as Germany and Belgium follow Vienna convention rules and hence have great similarity in the traffic sign notions. However, china had mostly different cases and hence the results were not that good, which again was an expected behaviour. (samples are not highly imbalanced)

	precision	recall	f1-score	support
diamond	0.97	0.64	0.77	50
hex	1.00	0.54	0.70	50
round	0.52	1.00	0.68	50
square	0.58	0.47	0.52	15
triangle	0.89	0.82	0.85	50
accuracy			0.73	215
macro avg	0.79	0.69	0.71	215
weighted avg	0.83	0.73	0.74	215

	precision	recall	f1-score	support
rightofway	0.00	0.00	0.00	11
stop	0.77	0.77	0.77	13
bicycle	0.00	0.00	0.00	14
limitedtraffic	1.00	0.90	0.95	20
noentry	0.00	0.00	0.00	16
noparking	0.76	0.94	0.84	17
roundabout	0.93	0.72	0.81	18
speed	0.11	0.06	0.08	16
trafficdirective	0.83	0.83	0.83	6
traveldirection	0.83	0.62	0.71	8
continue	0.90	0.75	0.82	12
crossing	0.83	0.95	0.88	20
laneend	1.00	0.81	0.90	16
parking	0.93	0.72	0.81	18
giveway	0.70	0.50	0.58	14
warning	0.42	1.00	0.59	30
accuracy			0.64	249
macro avg	0.63	0.60	0.60	249
weighted avg	0.62	0.64	0.61	249

Figure 7.1 German traffic Dataset shape (top) Sign(bottom)

	precision	recall	f1-score	support
diamond	0.00	0.00	0.00	0
hex	0.00	0.00	0.00	20
round	0.70	0.91	0.79	33
square	0.39	0.60	0.47	15
triangle	0.86	0.90	0.88	20
accuracy			0.65	88
macro avg	0.39	0.48	0.43	88
weighted avg	0.52	0.65	0.58	88

	precision	recall	f1-score	support
rightofway	0.50	1.00	0.67	12
stop	0.90	1.00	0.95	9
bicycle	1.00	0.17	0.29	12
limitedtraffic	0.80	0.33	0.47	12
noentry	0.00	0.00	0.00	4
noparking	0.11	1.00	0.20	1
roundabout	1.00	0.22	0.36	18
speed	0.11	0.05	0.07	20
trafficdirective	0.50	1.00	0.67	2
traveldirection	0.00	0.00	0.00	13
continue	0.00	0.00	0.00	14
crossing	0.21	0.17	0.19	18
laneend	0.00	0.00	0.00	11
parking	0.00	0.00	0.00	10
giveway	0.00	0.00	0.00	0
warning	0.00	0.00	0.00	0
accuracy			0.24	156
macro avg	0.32	0.31	0.24	156
weighted avg	0.39	0.24	0.25	156

Figure 7.2 Chinese traffic Dataset shape (top) Sign(bottom)

- German dataset had macro f1-score average of 0.71 & 0.60 for shape and sign respectively while Chinese data set did okay with shape target with macro f1-score avg of 43 but suffered with shape target. This can be looked at as ML algorithm's characteristics of knowing what it has seen. Chinese signs are much different than Belgium signs, however, German traffic signs are relatively similar. Hence, this model cannot be deployed directly in a global market. One way of generalising can be to use a mixed multi country dataset to train the model, as can be seen, CNN holds really good potential when trained with proper data.

Ultimate Judgement of models

- Shape target - CNN classifier with 3 (Conv2 * Conv2 * Maxpool layer) and 1 dense 1024 sized layer, with kernel size of 5 , pool size 3 and 16 filters & learning rate of 0.001 with RMSProp
- Sign target – CNN classifier with 3 (Conv2 * Conv2 * Maxpool layer) and 2 dense 1024, 512 sized layers respectively, with kernel size of 3, pool size 3 and 16 filters & learning rate of 0.001 with RMSProp

References

1. *Btsd.ethz.ch. 2020. Belgiumts - Belgian Traffic Sign Dataset. [online] Available at: <https://btsd.ethz.ch/shareddata/> [Accessed 29 May 2020].*

2. *Btsd.ethz.ch. 2020. Belgiumts - Belgian Traffic Sign Dataset. [online] Available at: <https://btsd.ethz.ch/shareddata/> [Accessed 29 May 2020].*

3. *Arxiv.org. 2020. VGG13 Model. [online] Available at: <https://arxiv.org/pdf/1409.1556.pdf> [Accessed 29 May 2020].*

4. *Medium. 2020. Review: Vggnet — 1St Runner-Up (Image Classification), Winner (Localization) In ILSVRC 2014. [online] Available at: <https://medium.com/coinmonks/paper-review-of-vggnet-1st-runner-up-of-ilsvlc-2014-image-classification-d02355543a11> [Accessed 29 May 2020].*

5. *Benchmark.ini.rub.de. 2020. German Traffic Sign Benchmarks. [online] Available at: <http://benchmark.ini.rub.de/?section=gtsrb&subsection=dataset> [Accessed 29 May 2020].*

6. *Kaggle.com. 2020. Chinese Traffic Signs. [online] Available at: <https://www.kaggle.com/dmitryyemelyanov/chinese-traffic-signs?select=annotations.csv> [Accessed 29 May 2020].*