Machine Learning Assignment 2: Classifying Images of Road Signs

By Luke Macdonald (s3888490)

Introduction:

Aim: The aim of this project was to construct a machine learning model that could be used in classifying different road signs according to their shape (e.g. round, square) and their type (e.g. stop, give way).

Dataset: The dataset that was used for the training of these models was a modified version of a dataset of the Belgium Traffic Sign Classification Benchmark

Algorithm: The algorithm that was used to construct the models was a class of a neural network called "Convolution Neural Network" or CNN, (Analysis of how CNN works and why it was selected will be discussed later in a later section.

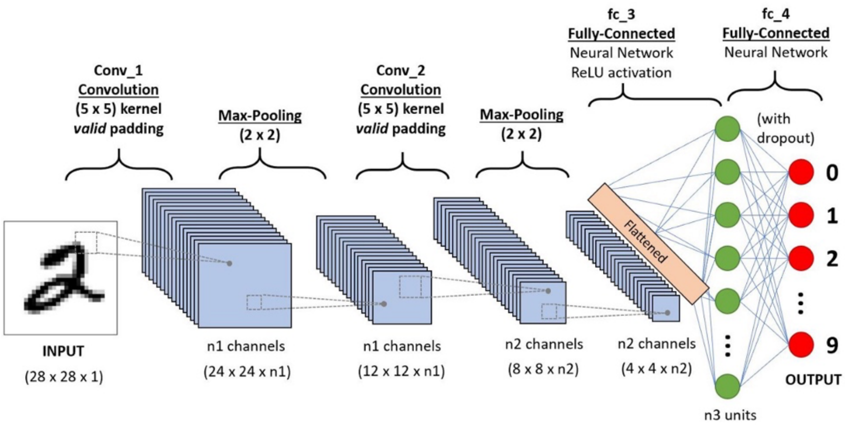
Machine Learning Algorithms Considered

Decision Tree: A Decision tree is a tree like structure where each internal node in the tree represents a test on an attribute and each leaf node contains a class label (E.g. round for testing shape)

Random Forest: Is a classification/regression model that works by fitting multiple decision trees on various sub-samples of data and uses averaging to improve the predictive accuracy and control over-fitting.

Neural Network: Neural networks are algorithms inspired by the human brain, more specifically the neurons. Neural networks work by having multiple layers of node's and depending on the input certain nodes will activate and fire and output of the neural network will be the class of which the input belongs to. For this project a class of neural network called CNN is going to be used. CNN is going to be used because it is the best fit for classifying images based on its ability to have a single node representing a single pixel for all the pixels in an image. This will allow the model to predict the class of an input more accurately than the other algorithms discussed.

Architecture of Selected Algorithm



The CNN architecture is made up of several layers

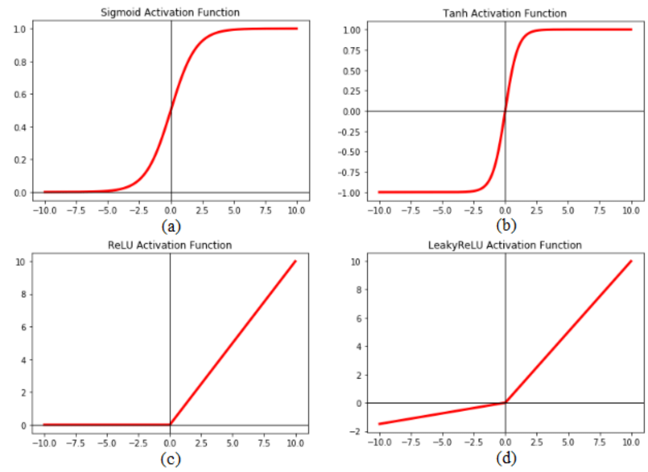
1. **Input Layer:** The input layer for this model is the raw pixel values of an image represented as a 3D matrix
    - Dimensions of the Matrix are Width × Depth × Height
        - Height is the number of pixels along the vertical axis
        - Width is the number of pixels along the horizontal axis
        - Depth corresponds to the colour changes (RBG) in the image. However, for the data provided the images have been grey scaled therefore the size of their depth is one.

2. **Convolutional Layer:** The convolution layer is the layer in which convolution filtering occurs.
    - Convolution filtering is the operation of “sliding” a filter over an image and calculating the convolution product and each portion of the scanned image

    Hyperparameters of Convolution Layer:
    - **Stride**: Refers to the number of pixels by which the filter moves across after each operation.
        - Stride is used in both the convolutional and pooling operations and for these models the stride is going be set to be set to their default value 1, i.e. move 1 pixel at a time
    - **Padding**: Refers to the operation of adding pixels to an image. Padding ensures that no border pixels of an image is missed during the convolution operation.
        - For these models the padding is going to be set to 'same', this will keep the same image size by adding zeros around the image

3. **Activation Layer:**
    An activation function in a neural network defines how the weighted sum of the input is transformed into an output from a node or nodes in a layer of the network.
    - The activation determines the relevancy of a given node(s) and if a node is found to be relevant by the model, than that node will "fire" after passing through the activation function. There are multiple types of activation function:

    Examples of Activation Functions



    - **ReLU:** The ReLU function is a non-linear activation function with its main advantage over the other activation functions (e.g. sigmoid) is that that it does not activate all the neurons at the same time. This means that the neurons will only be deactivated if the output of the linear transformation is less than 0

$$\max(0, x)$$

    - **Sigmoid Function:** The sigmoid function is a function that when values are entered into it, it will guarantee an output between 0 and 1 which is why it is primarily used in a model using binary classification. The sigmoid function is also another non-linear

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

    - **Tanh:** The Tanh function is very similar to the sigmoid function and can in fact be derived from the sigmoid function. The Tanh function differs from the sigmoid function by having its output value be between -1 and 1 while sigmoid output is between 0 and 1.

$$Tanh(x) = \frac{2}{1 + e^{-2x}} - 1$$

Choosing Activation Function:
The activation function selected to be used for the models in this project was ReLU.
- ReLU was selected as it less computationally expensive than tanh and sigmoid due to it having simpler mathematical operations
- I.e. ReLU is much faster than Sigmoid and Tanh

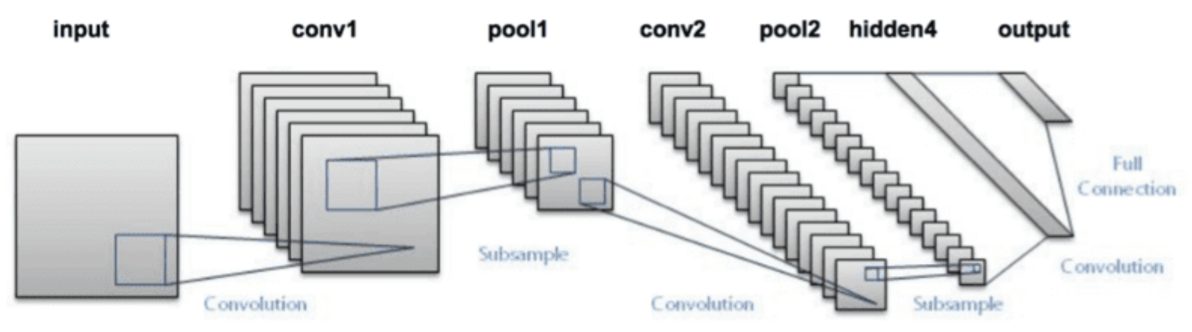
4. **Pooling Layer:** Pooling in CNN's is used for generalizing features that have been extracted by convolutional filters and for helping the network recognize features independent of their location in the image. The method of pooling is very similar to convolution procedure itself. A filter of a certain size (for this report the size of filter is 2x2) is selected and then slide it over the output feature map of the convolution layer. There are two main types of pooling methods.
    - Max Pooling: The filter selects the maximum pixel value in the field
    - Average Pooling: The filter calculates the average of all the pixel values in the field.

   Choosing Pooling Method:
    - The pooling method used for the CNN models in this project is going the be the max pooling because it will select the sharpest (maximum) features of an image which are the best representation of an image at this lower level.
5. **Fully Connected Layer**: The fully connected layer is the layer in which the output from the previous layer is taken in as input and then calculates the class scores and outputs the results in the form of a 1-D array where the size is equal to the number of classes.
    - The number of classes for this project is either going to be 5 (if predicting for shape of sign) or 16 (predicting type of sign)

   Choosing Activation Function:
    - The activation function chosen for the project in this layer of the models was the "SoftMax" activation function.
    - The SoftMax is an activation function that is only ever used in the last layer of the neural networks
    - The SoftMax function converts the vector of numbers into the vector of the probabilities which is why it has been selected as the activation function in the fully connected layer as it will produce probabilities of each class that will let the model know which class the input most likely belongs to

Evaluation of Trained Models

Architecture of Model



The actual architecture of the models was modelled after the LeNet Architecture. Developed in the 1990's and is considered "the classic CNN architecture that demonstrated a successful Deep network design for image recognition".

Data Augmentation: Due to the small proportion of data available to train the model, a method is available to make random changes to the data provided (E.g. rotating the image) to increase the diversity of the dataset. This method is called data augmentation.

Data Augmentations increase the variance of the training data in a variety of ways, including random rotation, increase/decreasing brightness, shifting object positions, and horizontally/vertically flipping images.

Evaluation of Classifying Shape Models

Figure 1. Loss and Accuracy Score of Base Model for Classifying Shape

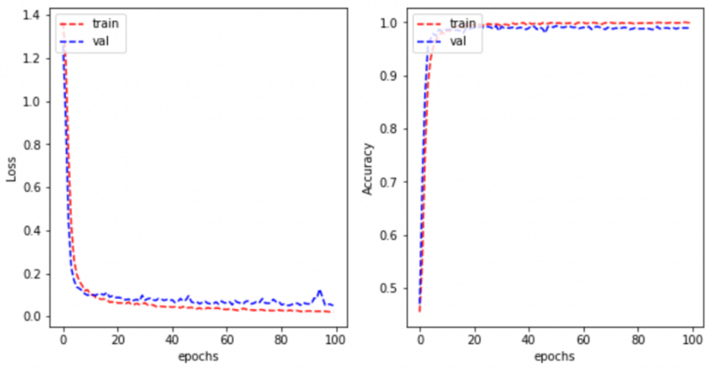
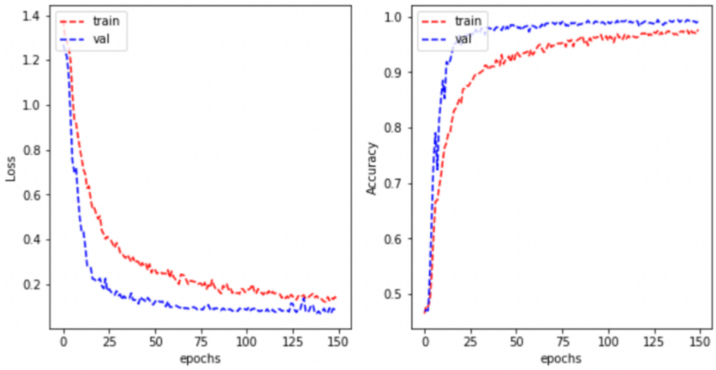


Figure 2. Loss and Accuracy Score of Augmented Model for Classifying Shape



For the base model it can be shown that the accuracy scores are very high for both validation and the training data and the loss is very low as well (≈ 0.1). It can also be seen that at around epoch 10-20 is when the accuracy score and loss begin remaining constant. This can allow for the model to be changed and improved by instead of running for 100 epochs it can be run for roughly 20-40 epochs which would be much more efficient as running around 100 epochs is very time consuming. For the model which has had its data augmented to increase the amount of data available for the model to train with it can be shown by looking at the accuracy scores between the training values and the validation values it can be seen that the validation accuracy score was higher than that of training data. This can indicate that model has been overtrained. Therefore by comparing the two models together the base model would be the best model to classify the images accurately.

Evaluation of Classifying Type Models

Figure 3. Loss and Accuracy Score of Base Model for Classifying Sign Type

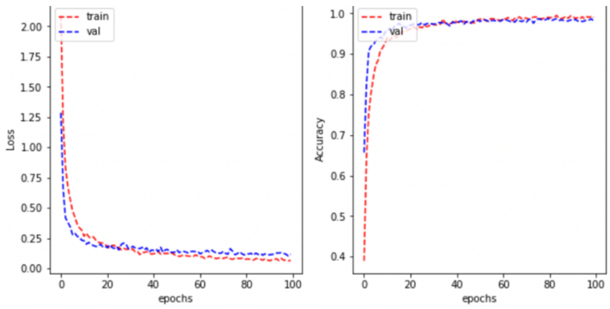
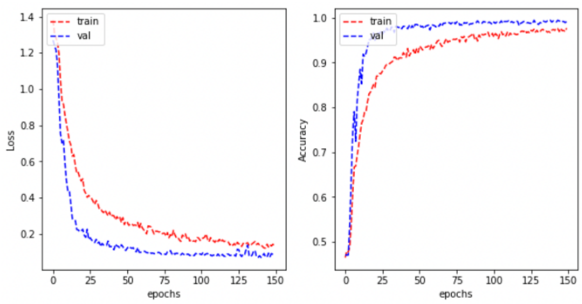


Figure 4. Loss and Accuracy Score of Augmented Model for Classifying Sign Type



The above graphs show similar results to classifying the shape of images. I.e. the model with augmented data has a higher accuracy score than the training data. Therefore indicating once again that the model was overtrained.

Ultimate Judgement

The above results have illustrated that using a convolutional neural network to an accurate way to train a model to classify road signs based on their shape and type. The results have also shown through the differing results of the model with augmented data and one without, that while data augmenting can be useful in increasing the amount of data available when the original dataset was minimal it also increases the risk of overfitting the model. Therefore for this reason when performing classification on the data gathered from the real world the base model is going to be used in classifying it as it was shown it have the best accuracy score in both classifying shape and type and also limited the risk of overfitting the model.

Independent Evaluation

<u>Gathering the Data:</u>

- The images collected for the independent evaluation for testing the models against real world data, were collected from the Vic Roads website and driverknowledgetests.com

<u>Pre-processing Data:</u>

Before classifying the images with the models the images had to be processed so that they would be able to accurately match against the model. This process involved having to greyscale the images as this is what the model was trained against. Also had to alter the size of the images as for the images the model was trained on they were 28x28 pixels in size.

<u>Results:</u>

The model was found to perform well one some of the images while on others it didn't accurately predict it well. This was mainly observed in classifying the type of sign. The reason for this is discussed in the limitations. Overall while the model classifying type may not be as effective in modelling unseen data as it was against the data it was trained with due to the different sign types having different shapes in different parts of the world. The model classifying the shape of the model was accurate and can be used to accurately classify signs according to shapes on unseen data.

<u>Limitations:</u>

The greatest limitation faced when performing the analyse on the independent data was getting images that were the same size as the images in the provided dataset (28x28 pixels) that the model was trained against. This was difficult when collecting the data because the original images sourced for the independent evaluation were originally images sourced by taking photos of street signs locally. These images however were of size roughly 3000x4000 pixels and when trying to get time to size 28x28 the pixels became disoriented, and the image was completely unrecognisable. For this reason the images instead had to be sourced online. Another limitation was that not all street sign types have the same shape, for example the roundabout sign from the dataset provided was round in shape while from the image gathered from the VicRoads website the roundabout image is diamond shape, this resulted in an inaccurate classification.

References

Machine Learning Mastery 2022. A Gentle Introduction to the Rectified Linear Unit (ReLU) [online] Available at: https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/#:~:text=The%20rectified%20linear%20activation%20function,otherwise%2C%20it%20will%20output%20zero [Accessed 28 May]

Research Gate 2022[online] Available at: https://www.researchgate.net/figure/Plot-of-different-activation-functions-a-Sigmoid-activation-function-b-Tanh_fig4_339991922 [Accessed 20 May]

GeeksforGeeks.com 2022 Activation functions in Neural Networks [online] Available at: https://www.geeksforgeeks.org/activation-functions-neural-networks/[Accessed 19 May]

vidyasheela.com 2022 Softmax Activation Function in Neural Network [online] Available at: https://vidyasheela.com/post/softmax-activation-function-in-neural-network-formula-included [Accessed 25 May]

Pyimagesearch.com 2022.LeNet – Convolutional Neural Network in Python [online] Available at: https://pyimagesearch.com/2016/08/01/lenet-convolutional-neural-network-in-python/ [Accessed 29 May]