Multi-Class Classification Model for VehicleX: Genetic Algorithm and Distinctiveness Pruning in the Shallow Neural Network

Linwei Zhang

Research School of Computer Science, Australian National University
U6758419@anu.edu.au

Abstract.
Neural network is a highly advanced approach of machine learning that can be exploited to work on numerous fields. As part of deep learning, a good model built using the neural network might require a large and real dataset. However, synthetic dataset is favoured as the collection of large and real data often brings up other issues such as privacy and data security. Hence, the paper aims to build a shallow neural network model that classifies the synthetic data and its true label with a detailed analysis on the choice of the parameters. Feature selection based on genetic algorithm with distinctiveness pruning and fine-tuning techniques are proposed that further improved the model performance in terms of the test accuracy. The model is tuned and evaluated on the validation and test dataset respectively to provide a reliable result that boost the confidence of the model. The integration of genetic algorithm and distinctiveness pruning achieved approximately 37.0% and 36.5% test accuracy respectively while the standard neural network achieve approximately 36.4% test accuracy. The hybrid approach that combines the genetic algorithm and the distinctiveness pruning achieves 37.2% test accuracy.

Keywords: • Multi-Class Classification • Synthetic Dataset • Genetic Algorithm • Distinctiveness Pruning

1 Introduction and Related Work

1.1 Introduction

With the advancement of machine learning (ML) and the improvement of computing power available in the modern days, many real-world applications are built based on the machine learning tools and approaches [11]. The neural network (NN), being a highly advanced approach of machine learning, can be exploited to work on numerous fields. As part of deep learning branch in ML, NN approach requires datasets to learn and build its model before applying into usage. More often than not, NN needs a large dataset to achieve a high accuracy of the model. However, there are many difficulties in obtaining a large dataset, such as the restriction of data collection due to privacy issues, data security concerns, the cost of labor and the potential error in the processing of labelling the data [1]. Hence, synthetic dataset is favored for the highly practicality and flexibility in generating the synthetic data [1]. Yet, the synthetic dataset might not accurately mimic the actual object and the model built using the synthetic dataset might have low performance. Hence, the paper aims to build a shallow neural network model that predicts the relationship between the feature of the synthetic VehicleX dataset and its corresponding true vehicle type. The paper would explore on adjusting the hyperparameters to build a baseline prediction model. Furthermore, the paper aims on equipping the baseline prediction model with the input feature selection based on genetic algorithm and the distinctiveness pruning and fine-tuning on the hidden layer to improve the overall performance of the model on classifying the vehicle instances. With the increased in the test accuracy, the model might be able to learn from the synthetic dataset while put into practical usage in future. Given the convenience of generating the synthetic data[1], such models reduce its reliance on the large-scale real dataset for the success of the neural network-based application. The model is tuned and evaluated on the validation and test dataset respectively to provide a reliable result that boost the confidence of the model.

1.2 Related Work

Yao et al. [1] worked on narrowing the content domain gap between the synthetic and the real data using a large and synthetic vehicle-related dataset, named VehicleX. Yao et al. [1] proposed the attribute descent approach to minimize the discrepancy between the vehicle and the real data calculated using Fréchet Inception Distance. The modified dataset is extended by classifying the dataset according to its vehicle ID. The performance result is evaluated using the mean average precision where the performance has a significant improvement contributed jointly by the synthetic and real data, with the use of the attribute descent approach [1].

2 Methods

The main objective of the methods is to build the NN and improve the model's performance on multi-class classification using the synthetic features of vehicleX that are extracted from Resnet which pre-trained on ImageNet. The methods involve input coding analysis and decision making, the use of cross-entropy loss and the Adam optimiser with a choice of activation functions for the model's hidden and output layers. Moreover, input feature selection based on genetic algorithm, and distinctiveness pruning and fine-tuning on the hidden layer techniques are applied to further improve the model's performance. The model is tuned and evaluated using the given validation and test set respectively.

2.1 Input Coding Techniques

2.1.1 Data Analysis and Decision Making

The dataset contains a total of 75516 vehicle instances which is divided into training, testing and validation datasets, with 45438, 15142 and 14936 images for training, testing and validation respectively. Each vehicle instance has 2048 feature data with a detailed labelled on other attributes such as the vehicle type, colour and orientation. With the observation that the dataset has 1362 unique vehicle IDs and 11 unique vehicle types and colour. Thereby, the dataset can be trained and classified according to the vehicle ID, type or colour with either 1362 or 11 classes.

With the choice between 1362 and 11 classes, classification on 11 vehicle type classes is preferred in the situation of limited computational resource support[1]. Given the large amount of data instances, a 11-class classification requires fewer hidden and output neurons, and a smaller number of hidden layers which allows a realistic and efficient training in the resource-limited situation. As shown in Figure 1, the 11 vehicle type are Sedan, SUV, Van, Hatchback, MPV, Pickup, Bus, Truck, Estate, Sports car and RV. The ratio between the number of vehicle instance to the vehicle type between the three datasets are similar. Given the large dataset and the fair split of the data points, the model trains on the train set. The model and its parameters are selected and tuned using the validation set. The test set would be used in the final selected model and parameters for performance evaluation in section 3 of the paper. The value derived from the validation or the test set are the average value taken from running the model, under the same setting, three times to counter the influence of the model's random weight initialisation on its performance. The usage of validation set for modelling and test set for performance evaluation prevents overfitting.

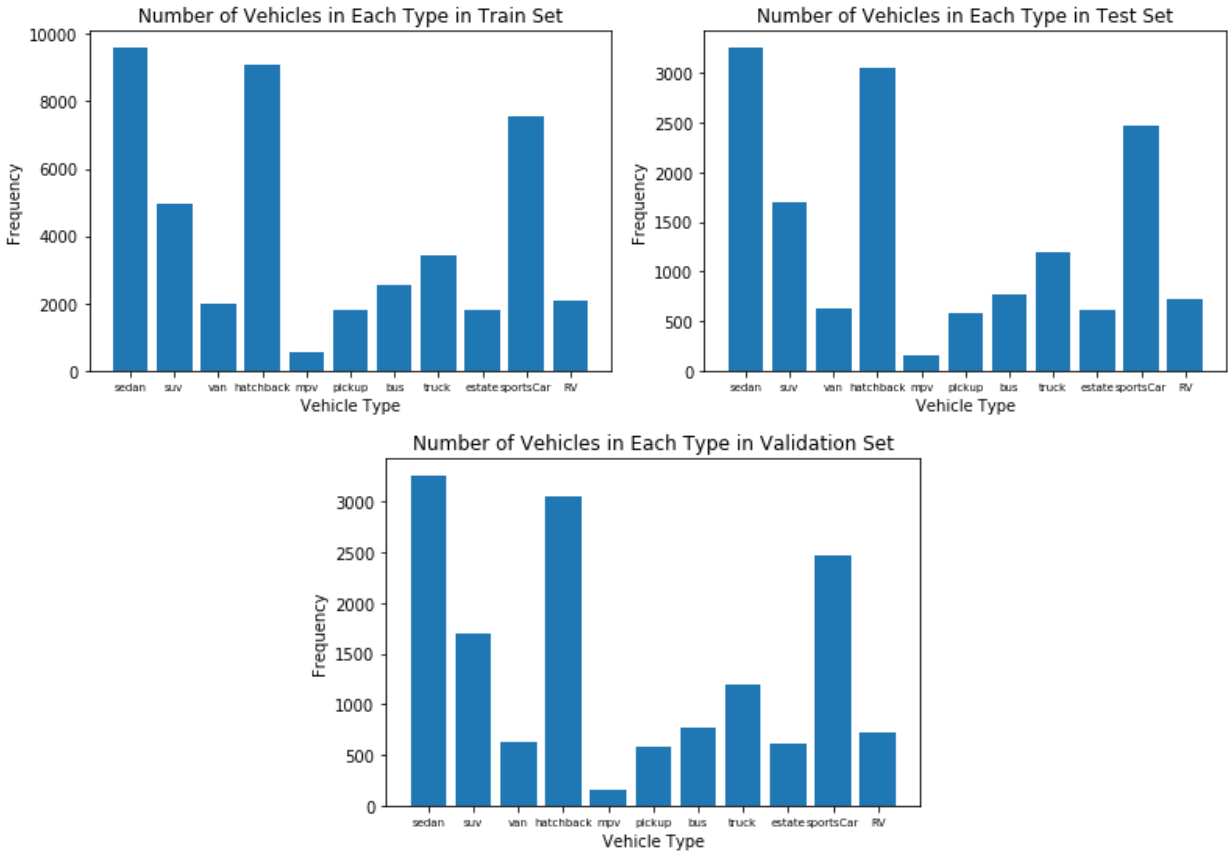

Fig. 1. Distribution of Vehicle Instances according to the Eleven Vehicle Types on Train, Test and Validation Set

[1] For instance, the experiment environment is lack of GPU support

As each vehicle instance has 2048 low-dimensional feature data extracted from Resnet that is pre-trained on ImageNet, the 2048 feature data can represent the actual image for the multi-class classification task sophistically. However, 2048 features means each vehicle instance has features in 2048 dimensions, which is hard to understand and learn [13]. Principal Component Analysis (PCA), being one of the popular dimensionality reduction technique, can be applied to the feature data [13]. As PCA calculates the covariance matrix for dimensionality reduction, the input data has to be normalized for PCA to be properly performed. Hence, Z-score, follows equation (1), is applied to the data points to ensure that the data are gaussian distributed for PCA to function well.

$$Z = \frac{x - \mu}{\sigma} \quad (1)$$

2.1.2 Neuron Network Layer Size

The number of neurons in the input layer depends on the number of feature of a vehicle instance. Since we apply PCA on the input feature data, the number of input feature is determined on a trials and error basis. When apply PCA, about 250 selected components in the train set could explain approximately 90% variance of the train set. As such, 250 draws the baseline for the PCA component selection. As shown in Table.1, with the increase in component size, the accuracy increases simultaneously. This is a result of having more features retained by a larger PCA component size. A higher number of feature remains suggests a higher dimensionality of the feature data in a vehicle instance. The shallow neural network would require a longer time to learn the complex feature [13]. Table.1 shows a clear trade-off between the accuracy and the training time where the training time increases by approximately 2 second for every 50 additional components in the validation set. The rate of increase in the accuracy decreases when the component size reaches 400. The mean accuracy difference from 300 to 350, 350 to 400 and 400 to 450 is 0.37%, 0.2% and 0.17% respectively. With a trade-off between the training time and the accuracy, the model uses the PCA on 400 components as it still increases 0.2% for the additional 50 components from PCA on 350 as compared to less than 0.2% increase for PCA on 450 components. Additionally, the decrease in the rate of increase in the mean validation accuracy suggests that the model has tried its maximum abilities to learn the input data. The model performance would not be heavily dependent on the number of input features after adapting 400 PCA components. Therefore, the NN is built with 400 input neurons.

Table 1. Mean Validation Accuracy for Different PCA Component Size

Component Size	Mean Validation Accuracy (in %)	Mean Training Time (in sec)
250	34.14	17.09
300	34.65	21.33
350	35.02	24.52
400	**35.22**	**26.51**
450	35.39	29.32

As mentioned in section 2.1.1, the paper aim to predict the true vehicle type of each vehicle instance. Since there are 11 vehicle types, the NN is built with 11 output neurons.

Due to the black box nature of the NN [4], it is difficult to determine the hidden layer size and the number of training epochs that allow fast converge while avoiding overfitting of the model. As the model would be implemented with distinctiveness pruning that will be discussed in section 2.3, we can simply fix the hidden neuron with a large number. Since the model is a simple neural network, the hidden neurons should be set between 400 and 11. With the limited computational resource, the hidden neurons is heuristically set as 100. By analysing Figure 2, the weight regularization is implemented to reduce the overfitting of the model. The weight regularization works in such a way that it adds a penalty weight to the loss function and slows down the learning. The model implemented L2 norm weight regularization at a rate empirically set as 0.1.

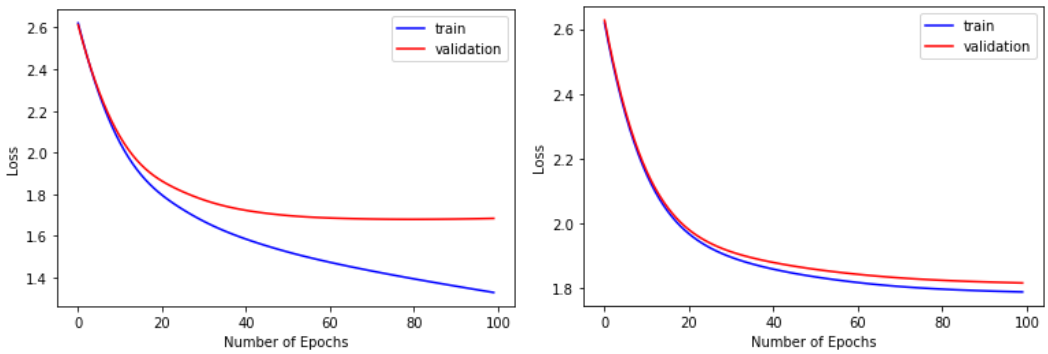

Fig. 2. Loss against Iterations Using 100 Hidden Neurons. (Left) shows the loss over 100 epochs without implementing weight regularization. (Right) shows the loss over 100 epochs with the implementation of weight regularization

By analysing Figure 2 and Figure 3, the model converges at around 50 epochs using the 100 hidden neurons with the validation accuracy reaches at around 34%.

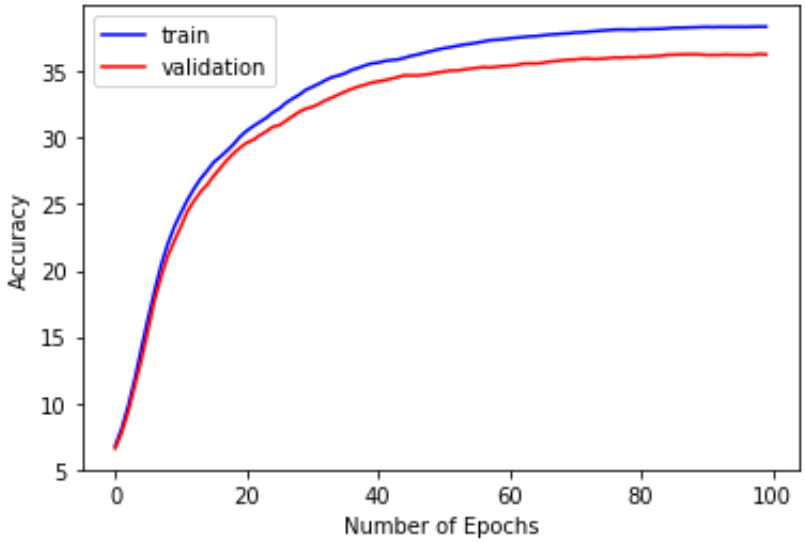

Fig. 3. Accuracy against Iterations Using 100 Hidden Neurons with weight regularization

2.1.3 Activation Functions

Since the objective of the work is to perform a multi-class classification, softmax function is typically use [5] on the output layer as it calculates the categorical probabilities distribution. The softmax function restricts the probabilities in the range between 0 and 1 with the sum of the probabilities as 1 using equation (2). Sigmoid is not considered as it works well on binary classification only.

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \tag{2}$$

There are three activation functions that takes into consideration when deciding the activation functions for the hidden layer of the basic model: Sigmoid, Hyperbolic Tangent (Tanh) and Rectified Linear Unit (ReLU). Sigmoid is not considered as it works well on binary classification only and its vanishing gradient problem [5]. Tanh is similar to Sigmoid but it has a steeper gradient and is symmetric about the origin [5]. Tanh is preferred over Sigmoid for its better performance. ReLU is also considered during the implementation of the basic NN model as it does not have the vanishing gradient problem as compared to Sigmoid and Tanh. However, the difference in the test accuracy and the time taken for ReLU and Tanh to converge is approximately 1% under the same experiment environment with empirical testing.

There are no significant variance in terms of the performance contributed by the Tanh and ReLU in the NN model built. As the range of the ReLU and Tanh is between 0 and its positive input and between -1 and 1 respectively, Tanh is chosen as the hidden layer activation function for its zero-centerd [5] property which would be discussed in chapter 2.2.

2.1.5 Loss Function

The cross-entropy/negative log likelihood loss is chosen as the loss function to calculate the error between the predicted and the true value. The formula of the cross-entropy loss function is shown as equation (3). It provides high penalty and a steep gradient for big difference between the predicted and the true value. Furthermore, it reduces the impact of the vanishing gradient problem caused by the Tanh activation functions and results in faster learning.

$$L(y,t) = -\sum_{n=1}^{N}\sum_{k=1}^{K} t_{nk} \log y_{nk} \tag{3}$$

2.1.6 Optimiser

As the input data is the dimension reduced on low-dimensional feature data, the model uses Adaptive Moment Estimation (Adam) optimiser as Adam computes the adaptive learning rate for each parameter based on the lower-order moments and is less sensitive to the parameters [6]. A less sensitivity to the parameters reduces the amount of time require to tune all parameters into their optimal combination for the model to be well-learnt. Hence, a small learning rate of 0.001 is initialised to avoid the model from having high sensitivity to the input data and learn the parameters quickly before reaching its optimal set of weights. In terms of the overall performance, Adam speeds up the converge of the model which is computationally efficient.

2.2 Feature Selection Based on Genetic Algorithm

Although the number of feature of each vehicle instance is reduced to 400 by PCA, the feature selected might not be the best component set as it is affected by the overall distributed of the data. As an evolutionary algorithm that inspired by the biological evolution, genetic algorithms (GA) are widely used in feature selection problems [12]. GA is especially useful in the feature selection as it discards the poorly performed subsets and use the better-performed datasets to generate the new datasets in the effort of obtaining N features that further improves the model performance. It works much better than brute force to find the best combination of N features as the search space is very large. For instance, trying to choose 1000 features from the 2048 feature data leads to a total of $3.247 * 10^{614}$ combinations. Hence, feature selection based on GA is proposed to exploit the GA's capabilities in finding the optimal set of feature before applying the PCA [13]. With a smaller set of feature data, the computation complexity for PCA to form the covariance matrix and extract the highest 400 components would also be lower [13]. As such, GA is used to select a subset of N features from the 2048 feature in each vehicle instance in the dataset. GA approach implemented is performed as shown below [12]:

1. The algorithm initialises K subset of N vehicle features which obtains K datasets with row as the number of vehicle instances and N feature columns. The N features are randomly selected and PCA is applied to reduce the feature size to 400 components.
2. The model is iteratively trained on each dataset of the K datasets and the accuracy is used to evaluate the quality of the datasets of N features.
3. The K datasets is then randomly split into two groups without replacement. The new subsets of N feature data are formed using the single point crossover of pairs of datasets from the two groups. Each pair formed a another two new datasets with features inherits equally from each dataset in the pair [12].
4. Mimicking the biological evolution, the new datasets have the tendency of mutation. Mutation occurs when a randomly generated number exceeds the probability pre-defined. The probability of mutation is set as 0.5 [12].
5. The model trained on the new datasets to obtain the accuracy.
6. Comparing all the accuracy of the available datasets and retaining only the top-K best accuracy datasets.
7. Repeat step 3 to 6 until the optimal value found or the pre-defined number of iterations over the crossover reached.
8. Output the best accuracy found.

As shown in Table 2, the higher the K and the crossover iterations, the more the time required. The number of K and the iterations over the crossover are pre-defined with the trade-off between the GA's effectiveness and the cost of computational resources and time. As compared to the baseline model, with K and number of crossover iterations set as 2 and 5 respectively lead to an increase in the validation accuracy within a reasonable time. Therefore, the model would train on the test set using K and number of crossover iterations as 2 and 5 respectively.

Table 2. Relationship between K, Crossover Iteration, Time Taken and the Best Validation Accuracy

K	Number of Crossover Iterations	Time Taken (in sec)	Best Validation Accuracy (in %)
2	2	337.56	35.46
5	2	870.41	35.43
2	**5**	**810.82**	**35.53**
5	5	1574.49	35.54
2	10	1467.87	35.41

Table 3. Relationship between the dataset of N feature, Best Validation Accuracy and Time Taken

N	Best Validation Accuracy (in %)	Time Taken (in sec)
1300	34.71	301.22
1400	34.87	311.67
1500	35.01	321.71
1600	35.30	330.86
1700	**35.46**	**337.56**
1800	35.37	349.12

By experiment on a trial and error basis, N is selected as 1700. The experiments in Table 3 are conducted using K and the number of crossover iterations as 2 for efficient search over the suitable parameters.

The random split in Step 3 avoids the redundant generation of the same dataset while maintaining the randomness nature of GA. In the case of mutation, a feature column of the dataset would be randomly selected and modified by the element-wise multiplication of random values. The mutation is done this way as the feature data are of low-dimensional. A random scaling between 0 and 1 would not largely destroy the variance of the overall dataset.

As shown in Table 4, the best validation accuracy found increases as the training epoch increases. By analysing Figure 4, the model converges at the 70 training epoch. Hence, set the training epochs as 70 for the model equipped with GA.

Table 4. Relationship between the Training Epochs and the Best Validation Accuracy

Training Epoch	Best Validation Accuracy (in %)
50	35.53
60	36.09
70	**36.35**

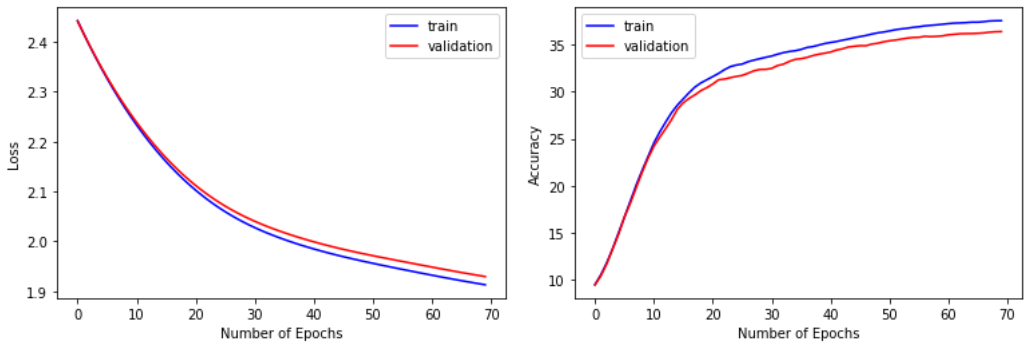

Fig. 4. Metrics against Iterations. (Left) Loss over 70 epochs. (Right) Validation Accuracy over 70 epoch

2.3 Distinctiveness Pruning

Distinctiveness pruning is a technique that removes the hidden neuron according to the angle computed between output activation vectors of hidden neurons on the data. As discussed in section 2.1.2, the black box nature of the NN [4] causes the difficulty in determine the hidden neuron size. While setting the hidden neuron size empirically might not lead to the optimal solution, finding the best-fit hidden neuron size experimentally might be time consuming. Hence, the distinctiveness pruning is proposed to analysis its effectiveness in reducing the hidden neuron size while maintaining the model performance. It takes a few steps to perform distinctiveness pruning with some modifications to suit the model [2]:
1. Train the model for 50 epochs and obtain the hidden layer's activation output.
2. Compute angles between each output activation vector with the rest of the output activation vectors use equation (4).
3. If the angle between any 2 neurons is less than 35 degrees, they are perceived as having similar functionality and one of them should be removed. The remained neuron would be adjust with the addition of the removed neuron's weight.
4. If the angle between any 2 neurons is more than 150 degrees, they are perceived as complementary and both of them should be removed.
5. Repeated step 1 to 4 with 1 training epochs until no further removal of the neurons.

$$angle = \cos^{-1}\left(\frac{i \cdot j}{||i|| \cdot ||j||}\right) \quad (4)$$

While the time for pruning might affect the overall performance, the model, in generally, does not have any pair of hidden neurons identified as similar or complementary before the training finishes. Therefore, pruning would occurs upon the completion of the training.

No normalisation is required as the Tanh activation function is symmetry and centred at the origin which allows the angle of any two neurons to be in between 0 and 180 degrees. The pruning algorithm is modified such that if two neurons are similar to each other while one of them is in complementary with another neurons, step 3 would be performed and none of the neurons is to be removed in step 4. This is because the comparison between neurons are done in step 2 and the changes made in step 3 might change the neurons' functionality, where the previously observed complementary pair of neurons might no longer in complement of each other in step 4. Therefore, none of the hidden neurons would be removed in step 4 to avoid the removal of useful neurons in this case.

By experiment, the minimum angle that the baseline model reaches upon the completion of the training varies between 29 and 33 inclusively as shown in Table 5. Hence, the minimum angle is chosen as 35 degrees to counteract the uncommon cases and to analysis the effect of pruning on the model performance. The choice of the angle at 35 degrees is further supported by Table 6 where the model achieved the highest average validation accuracy with the minimum angle at 35.

As shown in Table 5, the maximum angle that the baseline model could reach is approximately 150 degrees. It is much smaller than the suggested angle at 165 degrees for a complementary pair of neurons [3]. The model is unlikely to have complementary pairs of neurons with the use of 50 training epochs and 100 hidden neurons on the baseline model. The threshold for a complementary pair of hidden neurons is set as 150 to counteract uncommon cases.

Table 5. Initial Minimum and Maximum Angles of the Model

Current iteration	Minimum Angle (in degrees)	Maximum Angle (in degrees)
1	31.5	144.3
2	33.4	141.1
3	30.4	146.8
4	29.0	141.9
5	29.9	146.9

Table 6. Different Minimum Angle Threshold on the Model's Performance

Minimum Angle (in degrees)	Mean Validation Accuracy (in %)	Pruning Ratio (in %)
30	35.32	0.66
35	**35.47**	**5**
40	34.55	11

Although Gedeon's research [2] states that no further training is required for the network, the accuracy decreases slightly without retrain in the experiment. Fine-tuning is introduced in step 5 to recover the lost accuracy by retraining the model with the remaining neurons and an extra training epoch for the graph to converge [9]. Distinctiveness pruning and fine-tuning work in tandem to reduce the adverse effect of pruning.

2.4 Hybrid Approach

The hybrid approach is the integration of the feature selection based on GA with the distinctiveness pruning. The parameters pre-defined for GA in section 2.2 is retained. In the approach, distinctiveness pruning serves as an extension to the model integrated with GA to analysis the possible improvement in the model performance. The distinctiveness pruning works in the model training phase in GA. Upon the completion of training the model with the dataset, the algorithm applies the distinctiveness pruning steps stated in section 2.3.

By experiment, the initial minimum and the maximum angle generated by output activation vectors of the model's hidden neurons with GA integrated is usually at approximately 20 and 160 degrees. Not experiment to further examine the minimum and the maximum angle threshold as the set of feature varies by GA whereby the poor feature sets tend to perform poorly on the pruning. Furthermore, it is hard to analyse the performance of the multiple feature sets to reach a meaningful finding of the threshold. Thus, the minimum and maximum angle is set as 20 and 165 simply to counteract the possible unusual cases.

By training on the validation set, the average validation accuracy is approximately 36.40% at around 1890 seconds.

3 Results and Discussion

3.1 Model Performance Comparison

The feature selection on the input data based on GA is used to find the optimal subset of features from the dataset. The optimal dataset is expected to improve the model's performance. As shown in Table 7, the model integrated with GA has its mean test accuracy increases by 0.59% from the baseline model. The increase in the test accuracy suggests the usefulness of GA in selecting a good subset of features before applying PCA which boosts the model's performance.

Table 7. Mean Test Accuracy of the Model

Model	Mean Test Accuracy (in %)
Baseline	36.36
Baseline + GA	36.95
Baseline + Prune	36.49
Baseline + GA + Prune (Hybrid)	37.23

The distinctiveness pruning should improve the prediction of the model [10]. As shown in Table 7, the model integrated with distinctiveness pruning technique has its mean test accuracy increases by 0.13% from the baseline model. The distinctiveness pruning and fine-tuning work in tandem, by eliminating the unnecessary hidden neurons and retraining for a finer model, contribute to the improvement of the model performance.

The hybrid approach of the feature selection based on GA and distinctiveness prune has the highest test among all models. It out-performed the baseline model by the mean test accuracy of 0.87%.

The paper successfully built a shallow NN that classify the vehicle instances according the vehicle type. All techniques proposed have improved the test accuracy from the baseline model as shown in Table 7.

By analysing the order of the experimental time consumed for different combination of models shown in equation (5), the time taken by the hybrid approach is the longest. The result is reasonable as the hybrid approach takes computation time to perform both GA and distinctiveness pruning. A portion of the time is required for pruning on low-performed datasets generated by GA.

$$\text{Baseline} \leq \text{Baseline} + \text{Prune} < \text{Baseline} + \text{GA} < \text{Baseline} + \text{GA} + \text{Prune} \quad (5)$$

3.2 Feature Selection Based on Genetic Algorithm

The feature selection based on GA are widely used as a wrapper feature selection technique [12]. In the paper, GA is used to select multiple subsets of 1700 features from the 2048 feature data and find the optimal subset evaluated on the model performance in classifying the vehicle instances according to its vehicle type. The boost in the test accuracy showed in Table 7 is reasonable as GA tends to find the most effective set of features by generating the new datasets using the previously better-performed datasets in the effort of obtaining 1700 features that further improves the model performance. As the iteration goes, GA select the more relevant features and reduce the less significant feature for modelling. By comparing Figure 2, 3 and 5, model has a narrower difference between its loss and accuracy as a result of the better selected set of 1700 features. The test accuracy varies approximately from 36.65% to 37.22% inclusively. The 0.6% difference might be a result of the randomness in both the weight initialization of the model and the randomness in the GA. For instance, the randomness in the feature selection and mutation of the GA.

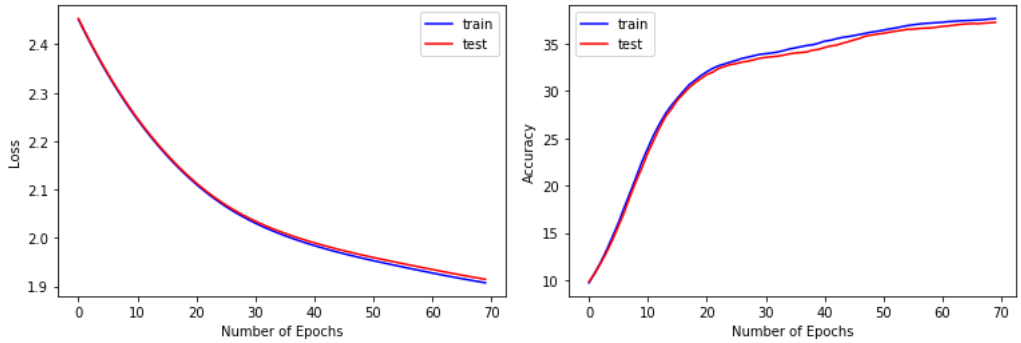

Fig. 5. Metrics against Iterations. (Left) Loss over 70 epochs. (Right) Test Accuracy over 70 epoch

3.3 Distinctiveness Pruning

With the reference to Table 6, the model performs the best using the minimum angle at 35 degrees. The phenomena is reasonable as the pruning ratio is approximately 0.66%, 5% and 11% with the minimum angle set at 30, 35 and 40 respectively. The pair of neurons with angle smaller than 35 degrees might be considered as similar for the model which shares the same functionality, but the pruning technique cannot identify the pair when the minimum angle is set at 30 degrees. When the minimum angle sets at 40 degrees, the model removes some useful hidden neurons that falls between 35 and 40 degrees.

Although the mean pruning ratio for angle at 35 degrees is approximately 5%, model performance could be slightly negatively influenced by prune. As shown in the experiment 2 of Table 8, the test accuracy drops by 0.12% from the initial performance. This could be a result of having a number of hidden neurons pruned at an instance while retraining on 1 epoch is not sufficient for the model to be finely tuned. In Table 5, the model could achieve an initial test accuracy that ranges approximately between 35.9% and 36.5%, which is 0.6% accuracy difference. Such small accuracy difference suggests the impact of the random weight initialization of the NN on its model and the technique's performance. As the sets of weight used in the model are randomly initialized, the number of training epochs required varies slightly. Sets of weight that better fits the feature data requires less training to reach a higher accuracy. Since a lower test accuracy indicates that the model is less well-trained and might not fully converged, the hidden neurons might be prune mistakenly. This is evident in experiment 2 of Table 5 where more hidden neurons are pruned on the model with a lower test accuracy and the final test accuracy drops slightly. Nonetheless, a drop of 0.12% in the test accuracy is small and acceptable.

Moreover, retraining on 1 epoch is sufficient as it is evident to improve the model performance displayed in the experiment 1 of Table 8. The 0.6% accuracy difference could be recovered by having few more training epochs but might not be necessary as the mean test accuracy of the model with distinctiveness pruning performs better than the baseline model as shown in Table 7.

Table 8. Impact of Hidden Neuron Size on Test Accuracy

Experiment 1		Experiment 2	
No. of Hidden Neurons Pruned	Test Accuracy (in %)	No. of Hidden Neurons Pruned	Test Accuracy (in %)
0	36.49	0	35.87
1	36.93	3	35.20
2	36.98	5	35.75

Table 9. Time Taken for Retraining on 1 epoch

No. of Hidden Neurons Remains	Time Taken to Retrain (in sec)
99	1.67
98	1.26
97	1.05
96	0.81

With the elimination of more hidden neurons, the model would have a smaller size and a lower cost of training [10]. As shown in Table 9, the distinctiveness pruning speeds up the time for the model to retrain with a smaller number of hidden neurons trained on an extra epoch. The increase in the training efficiency is reasonable as less time required for a smaller model to retrain in each and every epoch. Retrain on 1 epoch does not take too much computational resource while the hidden neuron size is optimized by the technique.

3.4 Hybrid Approach

The test accuracy of the hybrid approach ranges between 37% to 37.3%. As subsets of the 2048 features are randomly generated before applying to PCA, there are occasions of generating feature subsets that do not well-represent the original vehicle instance. The model would not learn well on a less relevant feature set and results in low classification accuracy. As discussed in section 3.3, more hidden neurons would be pruned with a lower performance model with the overall accuracy drops slightly without sufficient training and that the model performance might be influenced by the random weight initialisation. The distinctiveness pruning acts as a filter that tried to eliminate the seemingly good performed feature subsets by verifying the functionality of the hidden neurons as a good feature subset should be robust against the potential negative impact of the random weight initialisation. When pruning occurs on the seemingly good performed feature set, the accuracy decreases by pruning, the poorly generated feature set would have a higher chance of being discarded by GA. The new datasets are thus generated with the robust feature subsets that better explained the vehicle instances. As such, the mean test accuracy is stable at above 37% while applying GA alone still have the tendency of obtaining accuracy lower than 37%, under the same parameter setting.

According to section 2.3, the distinctiveness pruning stops only when no neurons are identified for removal. It takes a significant time to prune and fine-tune the model trained on the poorly generated feature subsets, leading to the long training time. GA and the distinctiveness pruning technique work in tandem to provide the optimal feature subset for the model to learn and perform better.

4 Conclusion and Future Work

The objective of the paper is to build a NN model that classifies the elements in the synthetic VehicleX dataset according to its vehicle type. A baseline prediction model is built using one hidden layer NN using Tanh and softmax activation functions with Adam optimizer for faster converge. Due to the large dataset size, the model is tuned using the validation set before applied to the test set. The use of the average accuracy when deriving the model performance in terms of the validation and test sets' accuracy provides a reliable result for tuning and evaluating the model performance respectively. In addition, the feature selection based on GA and the distinctiveness pruning followed by fine-tuning technique are proposed in an effort of improving the overall performance of the shallow NN model. The distinctiveness pruning compresses the model and improves the training efficiency. Among all approaches, the model integrated with the hybrid approach works the best with an average test accuracy reaches 37.23%. When we want to improve the model's performance with sufficient computational resource, the hybrid approach is recommended for its effectiveness in boosting the model performance. If the experiment environment has limited resource, the distinctiveness pruning is favored.

Due to the limited computational resource, the experiment conducted on a small initialization of the dataset and iterations in GA. The number of initialized dataset and iterations can be larger to explore more possible combination of 1700 features. Moreover, GA can be extend to find the optimal set of thresholds for the distinctiveness pruning to boost the model performance with the optimal solution on hybrid approach. The work can also extend on multi-task learning by including additional attributes, such as the vehicle orientation [8]. The stratified and repeated k-fold cross-validation might applied to split the datasets with similar proportion of the vehicle instance according to their vehicle type with repeated k-fold decreases the variance of the prediction error [7].

With reference to Table 7, a simple neural network model's performance can be improved with proper algorithms applied in its classification of the synthetic dataset. As the accuracy of predicting the actual object using the model trained on the synthetic dataset increases, the synthetic dataset would play an important role in the advancement of the machine learning and the utilization of NN would be in a greater scale in the near future.

References

1. Yao, Y., Zheng, L., Yang, X., Naphade, M., & Gedeon, T. (2020). Simulating Content Consistent Vehicle Datasets with Attribute Descent. ECCV.

2. Gedeon, T. D. (1995, November). Indicators of hidden neuron functionality: the weight matrix versus neuron behaviour. In Proceedings 1995 Second New Zealand International Two-Stream Conference on Artificial Neural Networks and Expert Systems (pp. 26-29). IEEE.

3. Gedeon, T. D., & Harris, D. (1991). Network reduction techniques. In Proceedings International Conference on Neural Networks Methodologies and Applications (Vol. 1, pp. 119-126).

4. Buhrmester, V., Münch, D., & Arens, M. (2019). Analysis of explainers of black box deep neural networks for computer vision: A survey. arXiv preprint arXiv:1911.12116.

5. Sharma, S. (2017). Activation functions in neural networks. towards data science, 6.

6. Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

7. Rodriguez, J. D., Perez, A., & Lozano, J. A. (2009). Sensitivity analysis of k-fold cross validation in prediction error estimation. IEEE transactions on pattern analysis and machine intelligence, 32(3), 569-575.

8. Tang, Z., Naphade, M., Birchfield, S., Tremblay, J., Hodge, W., Kumar, R., ... & Yang, X. (2019). Pamtri: Pose-aware multi-task learning for vehicle re-identification using highly randomized synthetic data. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 211-220).

9. Chandakkar, P. S., Li, Y., Ding, P. L. K., & Li, B. (2017, June). Strategies for re-training a pruned neural network in an edge computing paradigm. In 2017 IEEE International Conference on Edge Computing (EDGE) (pp. 244-247). IEEE.

10. Zhu, M., & Gupta, S. (2017). To prune, or not to prune: exploring the efficacy of pruning for model compression. arXiv preprint arXiv:1710.01878.

11. Ou, G., & Murphey, Y. L. (2007). Multi-class pattern classification using neural networks. Pattern Recognition, 40(1), 4-18.

12. A. M. Abo El-Maaty and A. G. Wassal, "Hybrid GA-PCA Feature Selection Approach for Inertial Human Activity Recognition," 2018 IEEE Symposium Series on Computational Intelligence (SSCI), 2018, pp. 1027-1032, doi: 10.1109/SSCI.2018.8628702.

13. S. Sehgal, H. Singh, M. Agarwal, V. Bhasker and Shantanu, "Data analysis using principal component analysis," 2014 International Conference on Medical Imaging, m-Health and Emerging Communication Systems (MedCom), 2014, pp. 45-48, doi: 10.1109/MedCom.2014.7005973.