Neural Network Development for Predicting Instant Fuel Economy

Bryon Spells
bspells

Erick Siavichay-Velasco
esiavich

Peter Karnchanapimonkul
phatk

June 2019

Abstract

Modern engine designers have many constraints to meet when creating a designs. These constraints include emissions regulations, high fuel economy and consumer performance expectations. These constraints are highly dependent on each other and can be competing factors due to this. When the vehicle is deployed, models are stored on the vehicle for the control architecture to use for engine optimization. Accurate, physics-based models are computationally expensive so instead simple, static look-up tables are used to reduce required computational resources but adaptive models are more appropriate. Adaptive models are better suited for engine control because the engine performance depends on many highly variable factors such as environmental conditions, region-specific emissions regulations, and driver behavior. The current work investigates neural networks (NN) as a way to bypass the computationally expensive physics calculations and still capture complexities. Instantaneous fuel economy (IFE) is a parameter related to the factors mentioned above, and is a measure of a vehicle's current performance. This performance will be predicted by a NN model. A baseline model was created using linear regression (LR) on 19 features which resulted in 7.5102 % accuracy. The final NN which contains 4 hidden layers with 320 neurons in each layer resulted in 89.33% accuracy. This indicates that NN's are a viable method for predicting engine performance and can be used as a model for engine control.

Introduction

Engines are mechanical systems which rely on thermodynamic relationships to produce propulsion force. A fuel-air mixture enters the engine's pistons and is ignited after compression in order to convert the potential energy of the fuel to heat. This causes the fuel to expand which results in the piston moving upward and this upward motion turns a shaft which creates rotational power for the vehicle to move. Since heat, pressure and mechanical power are so intermingled, the data labels that were chosen are related to the engine operating conditions regarding temperatures, pressures and fuel flow rates. The input parameters are { Absolute load value (%), Absolute throttle position (%), Ambient air temperature (F), Barometric pressure (inHg), Calculated load value (%), Commanded fuel rail pressure A (inHg), Engine coolant temperature (F), Engine RPM (RPM), Fuel level input (%), Fuel rail pressure (psi), Fuel/Air commanded equivalence ratio, Ignition timing advance for 1 cylinder (deg), Intake air temperature (F), Intake manifold absolute pressure (inHg), Mass air flow rate (lb/min), Vehicle speed (MPH), Fuel rate (gal/hr) }. These parameters are used to predict the target parameter: IFE in miles per gallon (MPG).

The current work determines if using a NN to predict engine performance parameters is a viable replacement for current engine modeling techniques. If the NN is able to predict a single output regarding engine performance without considering physical intuition in the model, then that is a strong indication that it can be used to predict others and will be a good model for engine control. IFE is chosen as the value to predict. To do this, the parameters described above were collected from a 2017 Mazda 3's on-board diagnostic port via an OBDLink LX by ScanTool. By the Universal Approximation Theorem, a NN can approximate any continuous function (e.g. the physics of the engine), meaning the NN can model the engine in at less computational expense than traditional physical models. This will enable adaptive models that enable advanced engine control techniques.

Related Work

NN's have gained increasing interest in the automotive industry with the aim of optimizing design, manufacturing and control of engines. Relevant applications of NN's range from simple applications of predicting average fuel economy based on high level vehicle features [3] to real-time prediction and optimization of engine operation parameters [1]. The former is done by considering weight,

model year, horsepower, number of cylinders in the motor, and other factors of the vehicle to predict an average fuel economy. This work can only be used to predict the average fuel economy which is not useful for real-time optimization.

Y. He et al and D. Egan et al proved that NN's can be used to improve engine modeling for real-time optimization [1], [2]. In [4], an artificial NN is used to model emissions, cylinder temperature, pressure and heat transfer of a turbocharged diesel engine. It was discovered that this technique resulted in a mean squared error of 1.2E-4 and 5.89E-4 for training and testing sets respectively. The implications of [2] and [4] are that while the current modeling techniques such as non-linear physics-based models and computational fluid dynamics are highly accurate, they are computationally expensive and the current engine control units cannot support them. However, by introducing the NN in place of the model [2] and combining physics-based techniques with a NN to linearize the models, it was found that the computational power was decreased and high accuracy was maintained [4]. In Figure 1, this becomes clear because the predictions for NOx for the artificial neural network and the proven, physic-based computational fluid dynamics model (KIVA) are similar.

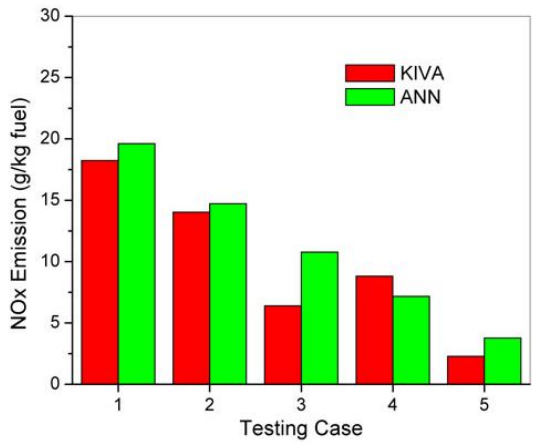

Figure 1: Neural Network predictions for NOx versus traditional model

While the results of these studies give high confidence that NN's are a viable method for decreasing computational expense while maintaining the high accuracy of physics-based non-linear models, both studies are conducted using data obtained in the lab. A common problem that machine learning practitioners face is good performance on lab data but unacceptable performance on real-world data. This is because real-world data is noisy and often the outputs are affected by factors that were not accounted for in the lab. The current work addresses this problem by using real-world data to train and validate the NN which will ensure the model generalizes well when deployed in a real setting.

Data Set

Engines are mechanical systems which rely on thermodynamic relationships to produce propulsion force. A fuel-air gaseous mixture enters the engine's pistons and is compressed and ignited in order to convert the potential energy of the fuel to heat. This causes the pistons in the gas to expand which results in the piston moving upward and this cyclic motion turns a shaft which creates rotational power for the vehicle to move. Since heat, pressure and mechanical power are so intermingled, all available data with labels related to the engine operating conditions such as temperatures temperatures, pressures, flow rates and load (force on the vehicle) were captured.

Figure 2: Example set up using OBDLink LX

Original data was obtained by driving in the highway and city for around 2.05 hours total. No particular region was chosen for the driving as it was determined irrelevant to repeat the route. Driving with no specific region in mind would introduce randomness in the data similar to the real world which helps ensure the generality of the NN. It was necessary to drive in both highway and city for similar amounts of time so that all behaviors were captured to comprehensively model the engine. City driving is characterized by quick, frequent acceleration and deceleration due to close traffic and traffic management which results in lower fuel economy and lower speeds. Highway driving is characterized by high, constant speed due to less traffic and reduced traffic management which results in higher fuel economy and higher speeds.

The 17 features mentioned above were streamed to a laptop while driving and stored as a csv file. Initially, 20 parameters were chosen but it was decided that Long term fuel % trim - Bank 3 (%) is a redundant feature because it is simply a calculation of the adjustment in the fuel flow rate which has been logged and is already fully

accounted for. The time label was also removed from the data as the engine's performance is not dependent on time, but rather it's instantaneous operation. Finally, the total fuel economy has been removed since it is simply and integrated average of instant fuel economy values and thus redundant. No data points were removed. The final data set contains 4861 data points. This data was split into training, validation, and testing sets - 60/20/20 (%) respectively.

	Absolute load value (%)	Absolute throttle position (%)	Ambient air temperature (F)	Barometric pressure (inHg)	Calculated load value (%)	Commanded fuel rail pressure A (inHg)	Engine coolant temperature (F)	Engine RPM (RPM)	Fuel level input (%)
0	21.56863	9.411765	64.4	29.8254	15.29412	885.9030	183.2	1431.00	81.17647
1	19.60784	11.764710	59.0	29.8254	33.72549	933.1512	185.0	575.75	89.41177
2	41.56863	20.392160	59.0	29.8254	44.70588	1866.3020	183.2	1424.75	62.35294
3	25.09804	14.509800	64.4	29.8254	34.11765	1163.4860	186.8	1053.75	70.98039
4	79.21568	32.941180	59.0	29.8254	96.07843	2769.9230	185.0	1400.75	60.39216
5	11.37255	11.764710	71.6	29.5301	13.72549	885.9030	188.6	1221.50	92.94118
6	10.98039	12.156860	62.6	29.5301	12.94118	885.9030	181.4	1477.50	50.19608

Figure 3: Data Example

Methods

In order to validate perceived benefits of the NN in predicting engine performance, a baseline model is needed. LR is a method commonly used for modeling linear relationships between independent variables and a target dependent variable and as such is well suited for a baseline modeling technique because of its ease of interpretability and implementation. The goal of linear regression is to create a linear function in dimension n-1 to predict data points in n dimensional space. The error between the true value y and the predicted value $\hat{y}$ computed by the function is computed using least squares regression, where the total error is given by

$$error = \sum_i (y_i - \hat{y}_i)^2$$

The error metric that was used to evaluate the LR model is given by

$$\text{percent error} = \frac{|\text{experimental value - accepted value}|}{\text{accepted value}} * 100\%$$

This equation is then used to evaluate the accuracy of the model, which is $acc = 1 - \text{percent error}$. Four neural networks were created to test out different architectures. The final implementation (model D) contained an input layer of size 17, reflecting the number of features, 4 hidden layers with 330 neurons each, and an output layer with 1 neuron. Regularization and bias units were also implemented. Each layer uses the activation function ReLU and the final output layer has no activation. This specific architecture was chosen because performance was saturated beyond these values - either the model performed worse or nearly the same. As a comparison, model A had an input layer, a hidden layer with 5 neurons, and an output layer.

The ReLU activation function is given by

$$f(x) = x^+ = max(0, x)$$

This activation function was chosen because it is nonlinear and layers that use this activation can be stacked to predict any function. Furthermore, this activation function can save computational power since some of the randomly initialized parameters can be negative to which ReLU returns 0, and thus many connected networks in the NN can have zero activation [6].

17 features are used as inputs to train the model. The parameters are then chosen by the gradient descent algorithm which works by taking the gradient of the cost function. The gradient descent algorithm works to minimize the cost of the model. It does so by initializing random parameters Θ and taking the gradient of the cost function with respect to Θ. The algorithm advances in the direction of decreasing derivative and settles in a minimum of the cost function. The parameters are rewarded by making their value larger if the chosen parameters decrease the cost and penalized by making the value smaller if the cost increases. The architecture of the NN was optimized systematically by created four models to see what hyperparameters to tune. To implement the LR model, sci-kit learn's Linear Regression python library was used. The results of this implementation are discussed in the next section in order to give a good comparison of the baseline and final models.

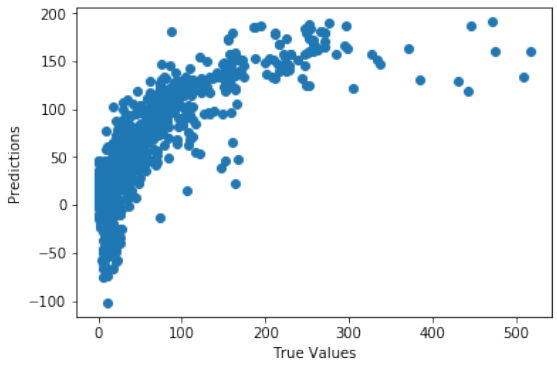

Figure 4: True labels vs predictions

If the model had good performance at predicting correct values, it is expected that this graph should have a near one-to-one correspondence between the true values and the prediction (i.e, the graph would be a diagonal line with positive slope). It does not, so a more complex model is necessary to capture the relationships between input features.

Experiments/Results/Discussion

The following tuning approach is taken to achieve a desired prediction accuracy (90%) for the NN: An initial NN is created, Model B, which has a single hidden layer. Model B's nodes in the hidden layer are increased until the loss is saturated. If the loss achieved is not satisfactory, an additional hidden layer is added and the process for determining the nodes is repeated. Each sequentially named model has an additional hidden layer and is developed based on the hyper-parameters of its predecessor. As seen in Figure 5, the accuracy of Model B saturated at about 130 epochs. Figures 6, 7 and 8 show the loss as a function of epochs as well which can be used to determine when to stop training the new model. A summary for the losses of each NN model is also seen in Figure 9:

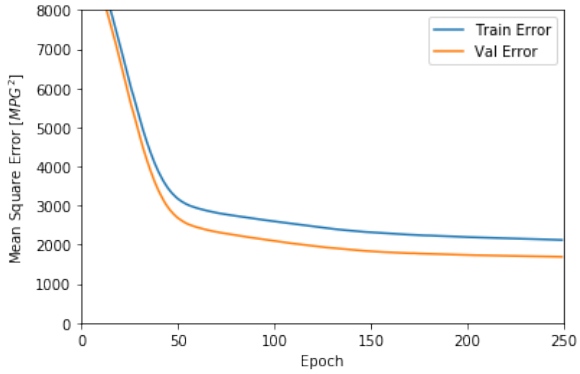

Figure 5: Model A Loss

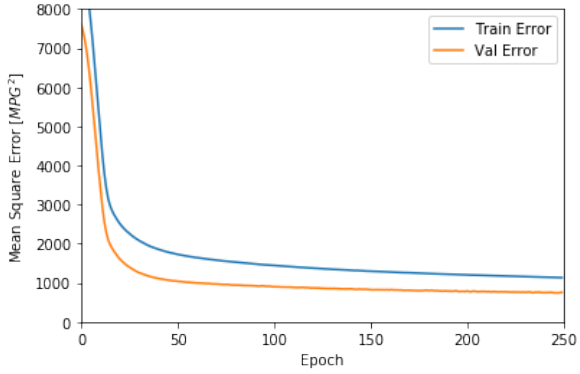

Figure 6: Model B Loss

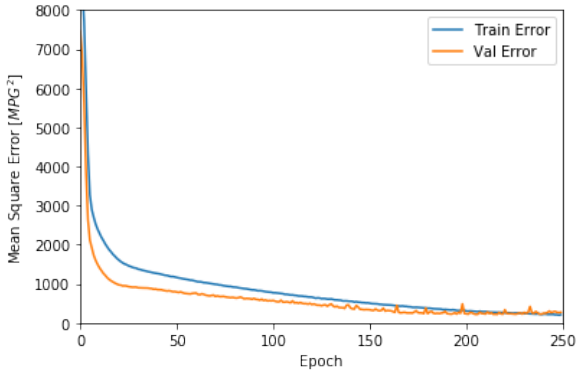

Figure 7: Model C Loss

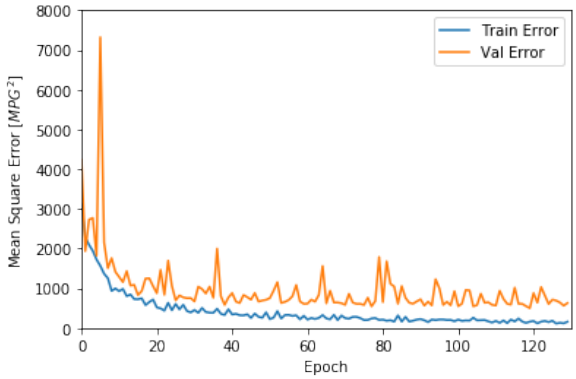

Figure 8: Model D Cost

Furthermore, the accuracies of all models are summarized:

Experiments	Learning Rate	Epochs	Num of Layers	Num of Neurons	Accuracy (%)
Linear Regression	n/a	n/a	n/a	n/a	7.51
NN Model A	0.001	250	1	5	42.19
NN Model B	0.001	250	1	20	57.13
NN Model C	0.001	130	3	15,10,5	74.29
NN Model D	0.001	130	4	330,330,330,330	89.33

Figure 9: Summary of Performance

The model is evaluated on the basis that the performance on the training and validation sets agree. This means that the gap between the two errors cannot be diverging as epoch's increases (variance) or that they are they are too close (high bias). The model is developed using the training set, made sure that the model generalizes well using the cross-validation set and confirmed using the test set.

This method lead to a NN with 4 hidden layers with 330 neurons in each layer respectively. The accuracy of this model versus the initial model is 89.33% versus 42.19%. The final model is significantly more accurate and the accuracy of the final model ranges from 86 to 90 percent depending on the random initialization of parameters.

Model A, the linear regression model, achieved an accuracy of 7.51% on the test set. As seen in Figure 5,

linear regression performed very poorly because its predictions are very biased. This can be attributed to the non-linearity of the data, which case it is expected that a NN would perform better because it can learn non-linear functions.

The comparison of Model A and the final model, Model D, shows that the NN algorithm is better at approximating engine performance than linear regression.

Conclusion/Future Work

Increasing emissions, performance and fuel economy regulations require advanced engine designs and control algorithms. The current work discusses a key aspect of the latter requirement. Current physics-based models are highly accurate but computationally expensive and not suited for the computing hardware in vehicles. Due to this, simpler lookup tables are created in lab testing which ignores the influence of the driver and environment on engine performance in real-world scenarios.

Neural networks are investigated as a solution to this problem as they can ignore the complex physics and still identify relationships between engine input parameters. First, a linear regression model is created with these parameters and determined to have an accuracy of 7.51 %. This serves as the baseline to compare the NN performance to. The neural network, which is developed by adding nodes in a hidden layer until the performance saturates then introducing a new layer, is determined to have an accuracy of 89.6 %. This is clearly an improvement on the baseline performance which indicates that NN are well-suited for supplying the accurate predictions needed for advanced control algorithms.

While the accuracy achieved is decent, the current work has many areas for improvement. The first being that more data is needed for definitive conclusion. The time-step of the current data acquisition tool made it hard to acquire tens of thousands of data points which would result in a robust model. Improving the acquisition tool, identifying which parameters are insignificant and driving for more time would address this shortcoming. Furthermore, the neural network was only trained for one vehicle, if more time allowed then other vehicles could be tested and the model could be assessed for how well it generalizes.

The results of this study are consistent with previous ones like [1],[2], [3] in suggesting that NN's can have significant benefit to engine control to help meed modern regulations and standards. In addition to confirming this hypothesis, the work introduces a new element which is training the model with real-world data. This is a novel approach as most NN development is done using stationary, ideal conditions in a lab. Further work is underway to address the shortcomings of this study.

References

1. Brahma, I. and Rutland, C., "Optimization of Diesel Engine Operating Parameters Using Neural Networks," SAE Technical Paper 2003-01-3228, 2003, https://doi.org/10.4271/2003-01-3228.

2. Egan, D., Koli, R., Zhu, Q., and Prucka, R., "Use of Machine Learning for Real-Time Non-Linear Model Predictive Engine Control," SAE Technical Paper 2019-01-1289, 2019, https://doi.org/10.4271/2019-01-1289.

3. Jamala, M. and Abu-Naser, S., "Predicting MPG for Automobile Using Artificial NeuralNetwork Analysis," IJAISR, 2018.

4. He, Y. and Rutland, C., "Modeling of a Turbocharged DI Diesel Engine Using Artificial Neural Networks," SAE Technical Paper 2002-01-2772, 2002, doi:10.4271/2002-01-2772.

5. Van Blagarian, A., Kozarac, D., Reinhard, S., Chen, J., Cattolica, R., Dibble, R., "Spark-ignited engine NOx emissions in a low-nitrogen oxycombustion environment", Applied Energy, https://doi.org/10.1016/j.apenergy.2013.12.007

6. Sharma, A., "Understanding Activation Functions in Neural Networks", Medium.com.