KSME

Springer

Journal of Mechanical Science and Technology 35 (4) 2021

Original Article
DOI 10.1007/s12206-021-0337-2

Keywords:
· Artificial intelligence (AI)
· Deep learning (DL)
· Deep neural network (DNN)
· Diesel engines
· NOx prediction model
· Prediction in steady-state

Correspondence to:
Kyoungdoug Min
kdmin@snu.ac.kr

Citation:
Lee, S., Lee, Y., Lee, Y., Shin, S., Kim, M., Park, J., Min, K. (2021). Proposal of a methodology for designing engine operating variables using predicted NOx emissions based on deep neural networks. Journal of Mechanical Science and Technology 35 (4) (2021) 1747~1756. http://doi.org/10.1007/s12206-021-0337-2

Received July 7th, 2020
Revised November 25th, 2020
Accepted December 29th, 2020

† Recommended by Editor
Yong Tae Kang

Proposal of a methodology for designing engine operating variables using predicted NOx emissions based on deep neural networks

Sangyul Lee[1], Yongjoo Lee[2], Youngbok Lee[2], Seunghyup Shin[2], Minjae Kim[3], Jihwan Park[2] and Kyoungdoug Min[2]

[1]Division of Mechanical and Electronics Engineering, Hansung University, Seoul 02867, Korea, [2]Department of Mechanical Engineering, Seoul National University, Seoul 08826, Korea, [3]Department of Mechanical Engineering, Myongji University, Yongin 17058, Korea

Abstract The process used by engine manufacturers for the development of a new engine includes the planning and conceptual design phases, followed by the detailed design phase, in which the design target specifications are met. In the conceptual design phase, a rough specification of the target engine is presented to facilitate a detailed design and the additional cost of modification is reduced exponentially. In the conceptual design phase, however, not only is there no real engine. but there are also no 1D and 3D models present, so it is impossible to test and simulate them. Therefore, at this stage, a model that can predict emission and performance only according to the specifications and operating conditions of the engine would be very useful. Previous studies developed an EGR prediction model that can be used in the 0-D NOx prediction using a deep learning method. In this study, a NOx prediction model with high accuracy using only the operating conditions as input variables, without ECU data, was developed using deep neural networks. The developed model has high accuracy with an R-square of 0.988. The feature of this model is that all the input parameters for the deep neural network come from the operating conditions of the engine. Therefore, this model can be used in the early stages of the development of new engines when testing and simulation cannot be performed because they do not exist. The designer can set the range of the operating conditions such that they do not exceed the NOx limits at the specific operating point (specific rpm and BMEP). This variable operating design methodology is expected to be useful in the development of new engines for automobile manufacturers with various engine data.

1. Introduction

Nitrogen oxides (NOx) and particulate matter (PM) are the main sources of fine dust and have recently emerged as a global issue [1]. In particular, NOx acts as a precursor to the substance that causes smog [1]. Emissions regulations have been tightened every year, and engine manufacturers must meet the regulations on emissions that will be applied in the region during the time period when the engine will be sold. North America and the European Union have proposed their own emissions regulations, and most countries use them as is or with some modifications. The CARB (California Air Resources Board) enforces more stringent regulations than the emissions regulations of the US EPA (Environmental Protection Agency), but recently, the EPA's Tier 3 and CARB's LEV III apply the same NOx standards.

In diesel engines, NOx is a function of temperature and oxygen content. Therefore, inside the engine, exhaust gas recirculation (EGR) is used to reduce NOx by lowering the combustion temperatures or reducing the oxygen content. Once produced, NOx must be removed through an aftertreatment system, such as a SCR (selective catalytic reduction) or LNT (lean NOx trap).

Meanwhile, according to Ullman, the process of engineering design is divided into the stages of product discovery, product planning, product definition, conceptual design, product development, and product support [2]. Among these, in the conceptual design phase, detailed targets should be set by analyzing customer requirements, competitor analysis, and government regulations [2]. According to this process, the conceptual design phase should be conducted in the process of developing a new engine, and at this stage, the requirements for the engine to be developed should be established in consideration of customer requirements, competitor analysis, and government regulations. For the new engine to compete against existing engines and competitors' engines, performance improvements such as fuel efficiency and durability must be clearly requested, and demands for exhaust emissions due to exhaust regulations must also be clearly established. In addition, the concept of the engine to be developed to satisfy the set requirements must be determined. However, in this process, 1D or 3D models for simulation as well as actual hardware have not yet been developed. Therefore, the design engineer must make a preliminary judgment as to whether a nonexistent product can meet the performance requirements and exhaust emission requirements. At this time, if the prototype created from the detailed design and production based on the established concept is wrong, the concept must be corrected, and the project must be resumed. However, the cost of the product increases as the product development process goes back for more design changes [2]; as many specifications as possible are determined in the design step, and they should not be changed. Therefore, it can be said that it is most important to quickly create an accurate conceptual design in the conceptual design stage. In general, at this stage, the results of similar engines are collected and analyzed by an experimental design method to create polynomials that can be applied with changing variables.

The deep learning methodology, which has recently been spotlighted, has excellent performance in predicting the nonlinear characteristics of the system. The minimum component of deep learning is one perceptron. For the perceptron developed in 1958, the input value was multiplied by the weight, and these values were added to obtain one value [3]. The added value was derived by adding a nonlinear characteristic through the activation function [3]. At the time of its development, the perceptron was evaluated as a groundbreaking algorithm that could simulate human judgment but showed limitations, such as the failure to classify simple XORs [4].

The multilayer perceptron solved the XOR classification problem by adding an intermediate layer called the hidden layer [5]. The algorithm in which this concept was introduced is called an artificial neural network (ANN). From the 1990s to the present, ANNs have been actively applied in various fields and to research on internal combustion engines. Various studies have been conducted since Ayeb et al. developed a dynamic model of the SI engine with a time delay neural network and a diagonal recurrent neural network and compared it with a real-time simulator [6].

Najafi et al. predicted the performance of a gasoline-ethanol blend engine using an ANN [7]. The brake power, torque, brake specific fuel consumption (BSFC), volumetric efficiency, engine speed and emission components using different gasoline–ethanol blends were set as inputs. In addition, engine performance and emissions were successfully predicted [7]. Mehra et al. used an ANN to learn the results of a CHNG engine experiment at a fixed engine speed and successfully predicted the performance of the HCNG engine from this model [8]. BSFC, torque, NOx, CO, THC, and CH4 were predicted from four inputs (excess air ratio, engine load, ignition timing, and HCNG blends) [8]. In addition, various internal combustion engine studies have been conducted using ANNs.

Deep learning is able to simulate very complex nonlinear characteristics by constructing deeper existing neural networks. The existing ANN has 0 or 1 hidden layers, but a DNN has 2 or more hidden layers. It was easy to conclude that a DNN would perform better than an ANN. However, the number of layers could not be increased due to the calculation time and the problem of not defining the error rate of the hidden layers. Yet, with the remarkable performance improvement of CPUs and GPUs and with the development of back-propagation methods, DNNs can now be used.

Several previous studies have applied deep learning methodology to internal combustion engines. Shin et al. predicted engine performance and emissions (NOx, CO) with high accuracy under various conditions by training deep neural networks as a result of SI engine simulations [9]. Park et al. diagnosed the timing of knock onset in SI engines using a deep neural network with high accuracy [10]. Using this model, Cho et al. analyzed the knock phenomenon of SI engines [11]. Lee et al. developed a high-accuracy EGR prediction model by conducting a study to predict the EGR of a DI engine under steady conditions [12].

In particular, a previous study [12] successfully predicted the EGR rates at the steady state engine operating points, which were used as the input values of the 0-D NOx prediction model. In this study, it was determined that the NOx value itself could be predicted as well as the EGR rates under normal conditions, and this was confirmed. Additionally, if NOx could be predicted from the steady-state engine operating points, then it would be useful for new engine development, and attempts have been made to predict NOx using only the operating point without the ECU data. First, experiments were performed on various steady-state operating conditions of the target diesel engine. Next, a DNN model was developed using various operating conditions and ECU data as input values, and the performance was verified. This was done to compare the results when a NOx prediction model is developed using only the operating conditions as input values. Then, the ECU data were removed from the input values, and a model with performance similar to that of the previous model was developed and verified by means of only the operating conditions as inputs.

In this study, there is only one research engine, and the

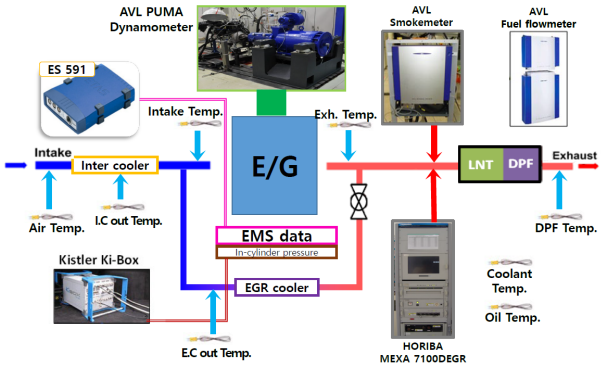

Fig. 1. Schematic experimental setup.

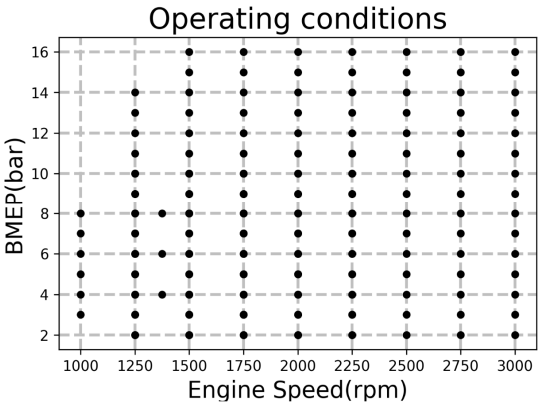

Fig. 2. Operating points in this study.

specification parameters of the engine are fixed. If experimental results and engine specifications exist for various engines, a deep learning model with engine specifications and operating conditions as input variables can be developed, which is considered to be very useful when developing a new engine.

In deep learning models, the configurations and performances of the model were confirmed, and the results were discussed. The conclusion section concludes this study.

2. Experimental setup and datasets

2.1 Experimental setup

The engine used in this study is a high-speed four-cylinder inline diesel engine. Fig. 1 is a schematic diagram of the experimental setup. The engine used has an engine displacement of 1.6 L and a compression ratio of approximately 16. The fuel used was diesel, and NOx was measured using the MEXA 7100 DEGR, a Horiba exhaust gas measuring device.

The experiment was performed under operating conditions that could cover the NEDC (New European Driving Cycle) operating area. The experiment was carried out in the same way as in a previous study [12], but the experimental case was increased to obtain more data. The experiments were performed while changing the injection timings and quantities of the main injection, pilot 1 and 2; the injection pressure in the engine from 1000 to 3000 rpm; and the load BMEP from 2 bar to 16 bars. The experimental case was selected as the DOE D optimal plan. The detailed experimental case is listed in Fig. 2. The total number of experimental cases performed was 696.

For more various data, the experiments were performed while changing the injection timings and quantities of main injection, pilot 1 and 2 and injection pressure in the range of engine speed 1000 to 3000 rpm and load BMEP 2 bar to 16 bar. The experimental case was selected as the DOE D optimal plan. The detailed experimental case is listed in Fig. 2. The total experimental case was 696 cases.

2.2 Datasets

Of the total 696 cases, 60 % of the experimental cases were used to train the deep learning model, and 20 % were used for validation during the training. The remaining 20 % of the experimental cases were used to test the finished model. Eighty percent of the data (training + validation) were recycled using the K-fold cross validation technique. The K-fold cross validation is a statistical technique used when the number of data points is small. After dividing 80 % of the data into k folds, the first fold was used as validation data, and the rest of the data were used as training data to perform learning and verification. Then, the second fold was used as validation data, and the rest of the data were used as training data to perform learning and verification. In this way, if the last fold was used as validation data, underfitting and overfitting could be reduced even while learning with less data. In this study, k was set to 5.

The input data were normalized to ensure that training progressed well and that the method did not diverge. Normalization was performed by subtracting each average value for each parameter and dividing by each standard deviation. The average value and standard deviation were extracted only from the training data so that the validation set and test set did not affect learning.

The output data of the DNN was the NOx value measured in ppm by the Horiba exhaust gas measuring device, and the total number of input data was 22 and was obtained from the dynamometer and ECU. The full parameters were the engine speed, engine load (BMEP), total air mass, boost pressure, engine torque, power, injection pressure, fuel masses of pilots 1 and 2, main and post injections, injection timings of pilots 1 and 2, ambient temperature, humidity, EGR rate (from the ECU model), and concentrations of CO, CO_2, O_2 and THC.

3. Deep learning backgrounds

3.1 Libraries

All the codes used in this study were written based on Python 3.6. The deep learning library uses Keras, which uses Tensor-Flow developed by Google as a backend [13].

TensorFlow provides a library for the operating CPUs and GPUs. Keras wraps APIs to use TensorFlow more easily and

conveniently. Additionally, the model training used in this study was performed using a GPU (Titan V).

3.2 Deep neural networks

The neural network adds all the input parameters multiplied by the initial weights and then passes the activation function to apply nonlinearity to the output values. In a deep neural network, this step is repeated several times, and the input parameter values are transformed into nonlinear output values. Supervised learning, which contains the correct answer in the dataset, compares the final output values (predicted values) with the correct answers and reflects the differences in these values by means of weights. If this process is repeated for all the training data, the weights are continuously updated to finally complete the training. When the training is completed, the performance of the developed model is evaluated using the test data. The number of hidden layers and nodes entering the deep neural network was tested with a parametric study. In general, since the deep neural network means that the number of hidden layers is two or more, the number of hidden layers starts from two.

In training, the difference between the correct answer and the predicted value is confirmed by the root mean squared error (RMSE) or the mean squared error (MSE). The difference can also be checked by the mean absolute error, but the average of the difference in values is not appropriate because it can be offset by adding positive and negative errors. Since (R)MSE is the sum of squares of differences, this offset does not occur and is a suitable method for determining the difference in values.

The Adam method was used as the optimization algorithm to find the optimal value. A combined method of Adagrad + RMSProp was used, and the main advantage is that the step size is not affected by gradient rescaling [14]. The step size is bound even when the gradient is large, so it is possible to stably descend for optimization no matter what objective function is used. In addition, the step size can be adapted by referring to the past gradient size. Many studies have been conducted using the Adam method, and previous algorithms have performed well with the Adam optimizer [12]. The Adam optimizer was also used as an optimization algorithm in this study.

The exponential linear unit (ELU) function was used as the activation function. The activation function that has been frequently used is the rectified linear unit (ReLU) function, which is a characterization function that solves the vanishing gradient problem [15]. The ReLU function has many variants that complement the disadvantages of the ReLU function. Among them, the ELU function was developed by Clevert et al. [16]. The ELU function includes all the advantages of the ReLU while eliminating the disadvantage of the deactivation of the node value when the input value is less than 0 by setting the slope to 0 [16]. Fig. 3 shows the forms of the ReLU function, ELU function, and other activation functions, such as sigmoid and hyperbolic tangent (tanh).

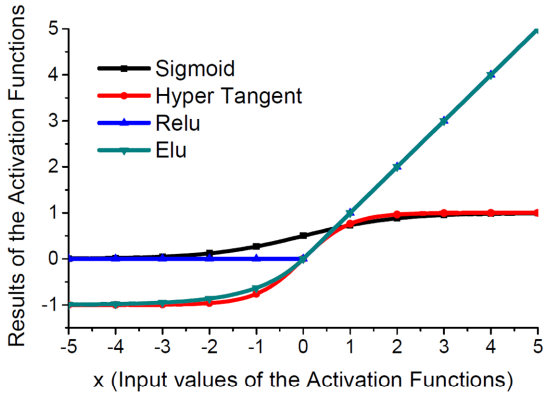

Fig. 3. Various activation functions.

3.3 Techniques to avoid overfitting

Overfitting means overtraining learning data in machine learning. In general, training data are a subset of actual data. Therefore, it is natural that the error decreases with respect to the training data as learning progresses. However, if overfitting occurs, the error increases with respect to the actual data, even if there is little error in the training data, which usually occurs when the training data are too small or the model is too large for a given number of training data. Therefore, the model should be developed so that the size of the model is not too large, and there are several well-known techniques as follows.

The drop-out technique randomly deletes the node's connection from the previous layer to the next layer [17]. When learning progresses, random nodes are selected and deleted so that no signal is transmitted. In this case, the effect of randomly selecting training data occurs, and the effect of learning with more training data than the given training data occurs.

Batch normalization refers to normalizing the output values of each layer. In other words, batch normalization is the task of normalizing the distribution of the data in each layer of a neural network [18]. Batch normalization is relatively free from the initialization problem (the initial weights have random values, so the results can be different each training time) because the outputs are normalized every training time. If batch normalization is performed, the learning rate can be set to a larger value so that the learning becomes faster. Additionally, since the distributions of input values are similar, gradient exploding can be prevented.

L1 and L2 regularization is a method of reducing the risk of overfitting by updating the general cost function by adding another term known as the regularization term. Due to the addition of this regularization term, the values of the weights decrease because it assumes that a DNN with smaller weights leads to simpler models. Therefore, regularization will reduce overfitting to quite an extent [19].

These techniques reduce the overfitting of the model, and all of these techniques were applied in this study. The detailed parameter values are documented in Sec. 4.

4. DNN models

4.1 Deep neural networks with ECU data

First, a DNN model was developed by including ECU data as input parameters to create guidelines for the best performance of the DNN model, which was developed without excluding the ECU data. The total number of input parameters is 22, and they are specified in Sec. 2. The input parameters included the EGR rate and O_2 concentration. NOx is generally considered to have a very high prediction accuracy because it has a strong relationship with the EGR rate and O_2 concentration. It was performed that the task of optimizing the various hyperparameters required for DNN, such as the number of layers, the number of nodes, and the learning rate. The goal was to develop a model that predicts NOx with high accuracy without overfitting.

Even if overfitting occurs, only the number of hidden layers and the number of nodes were adjusted to develop a model that can provide the best performance first. At this stage, the number of nodes and the number of layers are determined. Next, over-fitting was reduced by using a dropout. If the number of layers and nodes is too large for the amount of data, the possibility of overfitting increases. Therefore, it is important to determine the appropriate number of layers as the model should have as few layers as possible. In this study, the limitations of deep neural networks with 2 hidden layers to 5 hidden layers were compared. The performance index of the model was set to the MSE value of the training data at the time when the MSE value of the validation data did not decrease further as the learning progresses. When overfitting occurs, the MSE value of the validation data no longer decreases or increases as learning progresses. In addition, if learning continues and overfitting occurs, the MSE of the given learning data converges to zero. As described above, at first, the MSE of the training data was compared at this stage because a decision was made to focus on the performance of the model without worrying about overfitting. Table 1 lists the MSE values of the training data and validation data for various hidden layer numbers and node numbers. The model with the best performance was a model with 5 hidden layers, and the nodes formed a 22-44-66-44-22-4-1 architecture. The MSE for the training data of this model was 30, and it can be confirmed that the training data were fitted very accurately. (It can be seen that the prediction occurred with an error of approximately 5 to 6 ppm.) In addition, the MSE for the validation data was 290, and it can be confirmed that overfitting occurred, but the overfitting was not as substantial compared to that of other models. Fig. 4 shows the structure of the completed DNN model with 5 hidden layers, and the nodes formed a 22-44-66-44-22-4-1 architecture. The model was set in such a way that the size of the dimension initially doubled and tripled and then reduced again. Initially, several pieces of information were compressed in the individual input parameters, and then, as the dimensions increased (the number of nodes increased), each piece was gradually decompressed and the separated information was placed in the increased node. Then, as the dimensions decreased, the

Table 1. Results of raw DNN models with 23 parameters (including ECU data).

No. of hidden layers	Architecture of DNN model	MSE (training data)	MSE (validation data)
1	22-11-1	256	809
	22-22-1	208	408
	22-44-1	163	639
2	22-11-5-1	132	661
	22-22-11-1	209	1048
	22-11-11-1	206	677
	22-44-22-1	74	1139
	22-44-44-1	86	488
	22-33-22-1	45	480
	22-33-11-1	37	443
3	22-44-22-11-1	47	402
	22-44-22-5-1	90	649
	22-44-11-5-1	59	890
	22-44-44-22-1	33	378
	22-11-5-2-1	663	792
	22-11-11-11-1	61	367
	22-33-44-11-1	34	358
	22-33-22-11-1	56	352
	22-33-22-5-1	63	410
4	22-44-44-22-11-1	58	785
	22-44-22-22-11-1	48	266
	22-44-22-11-5-1	74	499
	22-44-44-11-5-1	269	484
	22-44-66-44-11-1	59	203
	22-44-88-44-22-1	48	831
	22-33-33-22-11-1	49	448
	22-33-44-22-11-1	41	465
	22-33-44-11-4-1	47	259
	22-33-44-33-11-1	42	278
	22-33-22-11-4-1	67	483
	22-22-22-22-4-1	65	663
	22-22-22-11-4-1	71	285
	22-11-6-3-2-1	153	420
5	22-44-66-44-22-11-1	55	394
	22-44-66-44-22-4-1	30	290
	22-44-66-22-11-4-1	705	648
	22-44-66-11-4-2-1	211	2232
	22-44-88-44-22-11-1	98	474
	22-44-88-22-11-4-1	62	209
	22-44-44-44-22-11-1	63	1002
	22-44-44-44-22-4-1	87	327
	22-44-44-22-11-4-1	68	943
	22-44-22-11-4-2-1	47	336
	22-44-22-22-11-4-1	42	239
	22-33-44-22-11-4-1	53	336
	22-33-33-33-22-11-1	42	313
	22-33-33-22-11-4-1	48	251
	22-22-22-22-11-4-1	46	319
	22-22-22-11-4-2-1	86	878

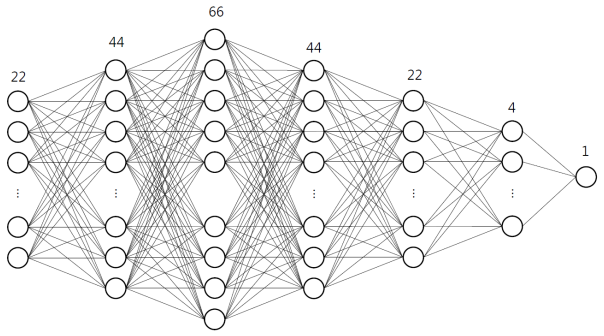

Fig. 4. Fully connected deep neural network with ECU data.

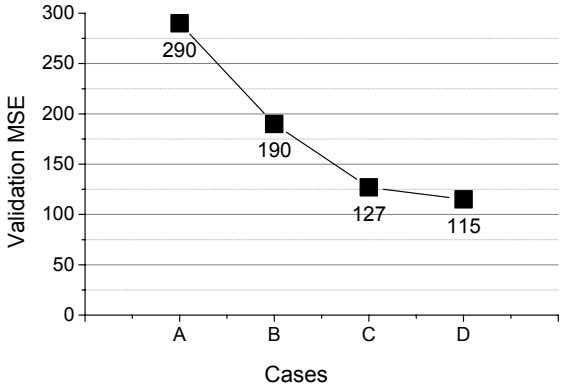

Fig. 5. Validation MSE according to each condition (A: Base condition, B: A+batch normalization, C: B+dropout (only 2 layers, 5 %) D: C+weight regularization 0.0001).

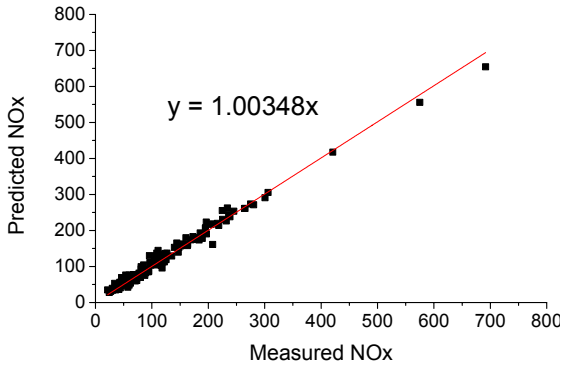

Fig. 6. Comparison of measured NOx and predicted NOx data using full parameter model with test data set and fitted equation with these data.

parameters that were not related to NOx were removed, and only the parameters related to NOx survived. This process is described in information engineering as 'the tangled thread is solved as the dimensions increase' [20]. When comparing the models of 22-11-1, 22-11-5-1, 22-11-11-1, 22-11-5-2-1, 22-11-6-3-2-1, 22-22-22-22-11-4-1, and 22-22-22-11-4-2-1, which are the cases in which the number of nodes is smaller than the number of input parameters, with the results of the optimal results model, a large difference in performance is observed. It is judged that performance is deteriorated when the dimensions are greatly reduced while various information related to NOx is compressed in individual parameters.

Next, various overfitting prevention techniques were applied to this model. Attempts were made to reduce overfitting using the batch normalization, dropout, and L1 and L2 regularization techniques mentioned in Sec. 3.3. Batch normalization plays the most important role. When batch normalization was added to all the hidden layers, the MSE of the validation data was reduced from 290 to 190. The result of the dropout technique was different depending on how many ratios were applied to each layer, and finally, the lowest MSE was found when 5 % was applied only to layers 1 and 2. At this time, the MSE of the validation was 115. Last, the L1 and L2 regularization also showed different results depending on the value applied to each layer, and finally, when both L1 and L2 were applied to only layer 1, the value of 0.0001 gave the best results. Fig. 5 shows that the validation MSE value decreases as various techniques are applied.

Finally, NOx prediction was performed by inputting the test set into the completed model. The test set was conducted with data that did not participate in the learning in the previous step. First, the test data were normalized using the average values and standard deviations of the training data that were stored in advance. Then, the stored best model was loaded to predict the NOx value under individual test conditions, and the NOx value was compared with the measured value (correct answer). Fig. 6 shows these results. The R2 value was calculated to be 0.9924. For these data, the linear fit equation was calculated using the measured NOx on the x-axis and the predicted NOx on the y-axis. As a result, it was confirmed that the slope was 1.00348, which is close to 1. The R2 and the slope of the linear fit mean that the measured values and the predicted values agree very well.

4.2 NOx prediction model with operating conditions

Among the previously selected 22 parameters, a predictive model was developed using 8 operating conditions as input data, excluding the data that could be obtained from the ECU. The 8 driving conditions are as follows: engine speed, BMEP, EGR rate, air mass, fuel mass, injection timing, boost pressure and injection pressure. These parameters are values that can be determined in the engine conceptual design stage, and the more accurate the values are, the lower the cost of the design stage.

Because the reduced prediction model also differs depending on the hyperparameters, a parameter study was conducted. The types and results of the various parameter study cases are listed in Table 2. For the same reason as in Sec. 4.1, the MSE of the training data at the time of overfitting was compared. As expected, it was confirmed that the performance deteriorated compared to the 22-parameter model. The model that shows

Table 2. Results of raw DNN models with 8 parameters.

No. of hidden layers	Architecture of DNN model	MSE (training data)	MSE (validation data)
1	8-4-1	1168	1599
	8-8-1	603	790
	8-16-1	991	1194
2	8-4-2-1	1438	3008
	8-8-4-1	869	1119
	8-4-4-1	551	650
	8-16-8-1	492	785
	8-16-16-1	353	586
	8-12-8-1	414	696
	8-12-4-1	629	800
3	8-16-16-8-4-1	230	326
	8-16-8-8-4-1	194	329
	8-16-8-4-2-1	869	1195
	8-16-16-4-2-1	334	616
	8-16-24-16-4-1	294	1878
	8-16-32-16-8-1	1414	618
	8-12-12-8-4-1	736	727
	8-12-16-8-4-1	311	1549
	8-12-16-4-2-1	1288	555
4	8-12-16-12-4-1	234	438
	8-12-8-4-2-1	208	528
	8-8-8-8-2-1	223	356
	8-8-8-4-2-1	1584	1147
	8-6-4-3-2-1	790	909
	8-16-8-8-4-1	392	505
	8-16-8-4-2-1	283	623
	8-16-16-4-2-1	969	694
	8-16-24-16-4-1	2325	2861
	8-16-32-16-8-1	277	444
	8-12-12-8-4-1	2894	4030
	8-12-16-8-4-1	944	1148
	8-12-16-4-2-1	1372	806
	8-12-16-12-4-1	362	1240
5	8-16-24-16-8-4-1	176	478
	8-16-24-16-8-2-1	263	515
	8-16-24-8-4-2-1	213	403
	8-16-24-4-3-2-1	1090	1174
	8-16-32-16-8-4-1	324	554
	8-16-32-8-4-2-1	186	531
	8-16-16-16-8-4-1	286	986
	8-16-16-16-8-2-1	216	679
	8-16-16-8-4-2-1	821	910
	8-16-8-4-3-2-1	313	407
	8-16-8-8-4-2-1	472	421
	8-12-16-8-4-2-1	268	718
	8-12-12-12-8-4-1	210	775
	8-12-12-8-4-2-1	254	805
	8-8-8-8-4-2-1	442	439
	8-8-8-4-3-2-1	573	386

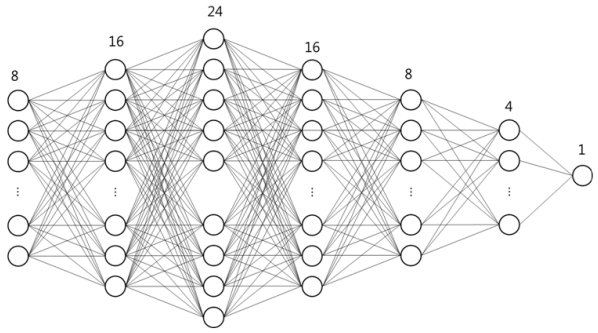

Fig. 7. Fully connected deep neural network with only operating conditions.

that the model that performs best has 5 hidden layers with nodes that formed a 8-16-24-16-8-4-1 architecture. The MSE for the training data of this model is 176, and it can be confirmed that the performance is not as good compared to the performance of the model with 22 parameters. However, the square root of the MSE is approximately 13, which is considered satisfactory considering that it indicates that the difference between the predicted value and the measured value is approximately 13 ppm. In addition, the MSE for the validation data is 478, which is confirmed to be overfitting, but the overfitting was not as substantial compared to that of other models. Fig. 7 shows the structure of the completed DNN model. As in the case of using 22 parameters, the model was set such that the size of the dimensions was initially doubled and tripled and then decreased again. However, unlike in the case of using 22 parameters, in the model using 8 parameters, the performance deteriorated in some cases where the dimensions increased and decreased. This is because the model is more sensitive to the disappearance of parameters that play a minor role.

Next, the MSE of the validation set was minimized by applying various overfitting prevention techniques to this model. Attempts were made to reduce overfitting using the batch normalization, dropout, and L1 and L2 regularization techniques mentioned in Sec. 3.3. First, when batch normalization was added to all hidden layers, the MSE of the validation data decreased from 478 to 276. When various dropout and L1 and L2 regularization hyperparameters were applied, the final MSE of validation was 198. Fig. 8 shows that the validation MSE value decreases as various techniques are applied.

Finally, NOx prediction was performed by inputting the test set into the completed model. The test set was conducted with data that did not participate in the learning of the previous step. First, the test data were normalized using the average values and standard deviations of the training data that were stored in advance. Then, the stored best model was loaded to predict the NOx value under individual test conditions, and the predicted NOx value was compared with the measured value (correct answer). Fig. 9 shows this result. The R2 value was calculated as 0.9880. For these data, a linear fitting equation was calculated using the measured NOx on the x-axis and the predicted NOx on the y-axis. As a result, it was confirmed that the slope was 0.97968, which is close to 1. The R2 and the slope

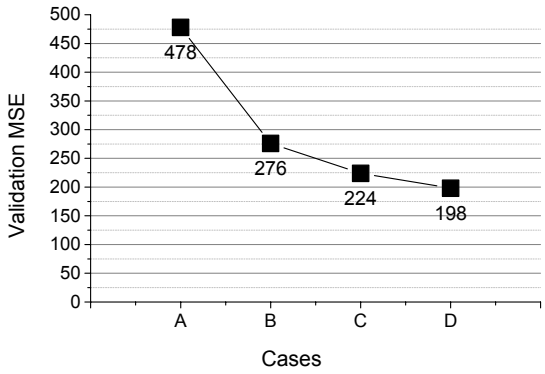

Fig. 8. Validation MSE according to each condition (A: Base condition, B: A+batch normalization, C: B+weight regularization 0.0001 D: C+dropout 0.1 only 2 layer).

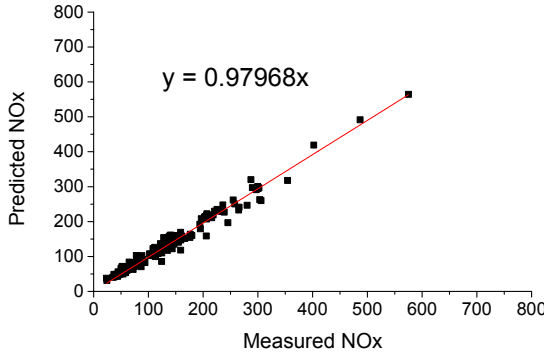

Fig. 9. Comparison of measured NOx and predicted NOx data using the 8-parameter model with the test data set and the equation fitted to these data.

of the linear fit means that the measured values and the predicted values agree very well. In addition, it can be seen that there is no substantial difference from the slope of 1.00348 of the 22-parameter model. This result shows that it is possible to predict NOx emissions sufficiently accurately using only 8 operating conditions.

In previous work [21], the 0-D equation was developed using various coefficients and constants. These constants and coefficients contain specific characteristics of the research engine, such as physical characteristics. In the developed DNN model, weights and activation functions play this role. If the constants and coefficients simulate only the characteristics of a specific engine, an overfit model can be developed only for the target engine. Likewise, if a DNN model is developed with only data of the same engine, it can overfit one engine data. This is a limitation of this study, and data from various engines are required to solve this.

4.3 Conceptual design with DNN model (example)

Engine manufacturers generally develop new engines when they need to meet new emission regulations. Design engineers do not know the NOx emissions of the target engine through experiments because there is no real target engine. Since 1D and 3D models do not exist, it is also impossible to predict using simulation. Therefore, the engineer should be able to predict the approximate NOx emission value using only the conditions that he or she has, and the conditions that are possessed are the overall specifications and operating conditions of the engine. The conditions used in this model are engine speed, BMEP, EGR rate, air volume, fuel volume, injection timing, boost pressure, and injection pressure as described above. The engine speed, BMEP, is the driving point. From the target fuel efficiency and the target AFR generated from the PM target, the fuel amount and air amount are roughly determined. Therefore, the designer can find the parameters that can satisfy the target NOx by controlling the boost pressure, main injection timing, and EGR rate.

Table 3. Example of 8 operating conditions and the predicted and measured NOx emissions.

Parameter	Unit	Source	Value
Engine speed	rpm	Operating point	1500
Load (BMEP)	bar	Operating point	4
Total fuel mass	mg/str	Target fuel consumption	10.23
Total air mass	mg/str	Target AFR by PM	444
Fuel injection timing (main)	degCA	Similar engine (variable)	-0.2
EGR rate	%	Similar engine (variable)	36
Injection pressure	bar	Similar engine (variable)	640
Boost pressure	bar	Similar engine (variable)	1.3
Predicted NOx	ppm	DNN model	108
Measured NOx	ppm	Horiba exhaust gas analyzer	113

For example, the designer wants to know in advance how much NOx emission is generated at a specified engine operating point (specific rpm and load condition). This is because the rough specifications of the hardware to be designed are determined according to the main injection timing, the EGR rate, etc. From past experimental results, the designer can check the injection pressure and boost pressure of an engine similar to the engine under development. The total fuel quantity can be calculated from the development target fuel consumption. The target AFR is calculated from the development target PM emission. Since the total fuel amount has been calculated, the required air mass is easily calculated. The main injection timing and EGR rate can be checked from similar engines. Table 3 lists the assumptions of the input parameters, predicted NOx emissions of those conditions by the DNN model, and experimental measurements of similar conditions. As a result of applying the conditions in the table to the developed reduced deep learning model, the engine-out NOx emissions are predicted to be 108 ppm. The NOx measured experimentally under the above conditions is 113 ppm. It can be confirmed that the prediction was accurate for the predicted value using only

the driving conditions.

Of course, parameters such as engine geometric specifications are reflected in the weights of the DNN model. Therefore, it cannot be applied to engine development with different geometric specifications. This is because this study is the result of experiments with one engine at the laboratory level. To apply this methodology to engines of other specifications, major engine geometric specifications must be added to the input variables. Engine manufacturers have various specifications for engine test data, so it is considered that they can be applied well.

5. Conclusions

In the conceptual design phase of the new engine, it is impossible to simulate an experiment, as there is no real engine. Therefore, designers interpolate as much data as possible from similar engine experiment data to select parameters at the various operating points. However, since this method has difficulty in combining various variables, many modifications occur in the detailed design stage and prototype development stage, which leads to an increase in development cost.

In this study, it was developed that a deep neural network model that can predict NOx emissions, which are the main design targets of engines, using the deep learning methodology at the engine conceptual design stage. First, 696 cases of experimental data were obtained under the conditions of an engine speed of 1000-3000 rpm and BMEP 2-16 bar. To check the upper limit of the performance of the model, the model was first developed and verified using 22 parameters, including ECU data. The NOx prediction model developed with 22 parameters was accurate enough to record an R2 of 0.9924 when the value was predicted using a test set not participating in the learning.

The performance of this model was considered to be the upper limit, and the model was reconstructed by selecting eight operating conditions: engine speed, BMEP, EGR rate, air mass, total fuel mass, injection timing, boost pressure, and injection pressure. The model that had 5 hidden layers and nodes with an 8-16-24-16-8-4-1 architecture showed the best performance. The developed NOx prediction model had an R2 of 0.9882 when the NOx value was predicted with a test set that did not participate in learning. It was confirmed that the prediction was accurate for the predicted value using only the driving conditions.

This model allows designers to predict NOx emissions with sufficient accuracy, even when the parameters are simultaneously adjusted, which is expected to be very helpful for new engine development. However, since the main geometric specifications of the engine are reflected in the weights of the deep learning model, it is difficult to directly apply the model to other engines. This limitation can be solved by obtaining various engine experiment data from the engine manufacturers and applying the key geometric specifications of the target engine as additional input parameters.

Acknowledgments

This research was supported by the Hyundai Motor Group and the SNU IAMD for Kyoungdoug Min, and also financially supported by Hansung University for Sangyul Lee.

Nomenclature

ANN : Artificial neural networks
BMEP : Brake mean effective pressure
DNN : Deep neural networks
ECU : Engine control unit
EGR : Exhaust gas recirculation
ELU : Exponential linear unit
NOx : Nitrogen oxides
PM : Particulate matters
RELU : Rectified linear unit
(R)MSE : (Root) mean squared error

References

[1] EPA (Environmental Protection Agency), *Nitrogen Oxides, Why and How They are Controlled*, Technical Bulletin, EPA 456/F-99-006R (2012).

[2] D. G. Ullman, *The Mechanical Design Process,* New York, McGraw-Hill, 2 (1992).

[3] F. Rosenblatt, The perceptron: a probabilistic model for information storage and organization in the brain, *Psychological Review*, 65 (6) (1956) 386-408.

[4] M. Minsky and S. Papert, *Perceptrons: An Introduction to Computational Geometry (Expanded Edition),* Editorial the MIT Pres. Libro publicado, 28 (1987).

[5] J. L. McClelland, D. E. Rumelhart and PDP Research Group, *Parallel Distributed Processing,* Cambridge, MA: MIT press, 2 (1987).

[6] M. Ayeb, D. Lichtenthäler, T. Winsel and H. J. Theuerkauf, SI engine modeling using neural networks, *SAE Technical Paper*, 980790 (1998).

[7] G. Najafi, B. Ghobadian, T. Tavakoli, D. R. Buttsworth, T. F. Yusaf and M. Faizollahnejad, Performance and exhaust emissions of a gasoline engine with ethanol blended gasoline fuels using artificial neural network, *Applied Energy*, 86 (5) (2009) 630-639.

[8] R. K. Mehra, H. Duan, S. Luo, A. Rao and F. Ma, Experimental and artificial neural network (ANN) study of hydrogen enriched compressed natural gas (HCNG) engine under various ignition timings and excess air ratios, *Applied Energy*, 228 (2018) 736-754.

[9] S. H. Shin, S. Y. Lee, M. J. Kim, J. H. Park and K. D. Min, Performance and emission prediction of gasoline engine using deep neural networks, *Korean Society of Automotive Engineering Autumn Conference* (2018).

[10] J. H. Park, S. W. Cho, C. H. Song, S. Y. Lee, M. J. Kim and K. D. Min, Development of a knock onset determination model in SI engine by deep learning, *Korean Society of Automotive En-*

gineering Autumn Conference (2018).
[11] S. Cho, J. Park, C. Song, S. Oh, S. Lee, M. Kim and K. Min, Prediction modeling and analysis of knocking combustion using an improved 0D RGF model and supervised deep learning, *Energies*, 12 (5) (2019) 844.
[12] S. Lee, Y. Lee, Y. Lee, M. Kim, S. Shin, J. Park and K. Min, EGR prediction of diesel engines in steady-state conditions using deep learning method, *Int. J. of Automotive Tech*., 21 (3) (2020) 571-578.
[13] Keras Documents, *https://keras.io*.
[14] P. Kingma and J. Ba, Adam: a method for stochastic optimization, *arXiv preprint arXiv: 1412.6980* (2014).
[15] V. Nair and G. Hinton, Rectified linear units improve restricted Boltzmann machines, *International Conference on Machine Learning* (2010).
[16] D. A. Clevert and T. Unterthiner, Fast and accurate deep network learning by exponential linear units (ELUs), *arXiv:1511.07289* (2015).
[17] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, *The Journal of Machine Learning Research* (2014) 1929-1958.
[18] S. Ioffe and C. Szegedy, Batch normalization: accelerating deep network training by reducing internal covariate shift, *arXiv:1502.03167 [cs.LG]* (2015).
[19] A. Krogh and J. A. Hertz, A simple weight decay can improve generalizaztion, *NIPS'91: Proceedings of the 4th International Conference on Neural Information Processing Systems* (1991) 950-957.
[20] F. Chollet, *Deep Learning mit Python unt Keras: Das Praxis-Handbuch vom Entwicker der Keras-Bibliothek*, MITP-Verlags GmbH & Co. KG. (2018).
[21] W. Park, J. Lee, K. Min, J. Yu, S. Park and S. Cho, Prediction of real-time NO based on the incylinder pressure in diesel engines, *Proc. Combustion Institute,* 34 (2) (2013) 3075-3082.

Sangyul Lee obtained his B.S. (2006), M.S. (2008) and Ph.D. (2013) in Mechanical Engineering from Seoul National University, respectively. Presently he is an Assistant Professor in Division of Mechanical and Electronic Engineering at Hansung University, Seoul, S. Korea.

Kyoungdoug Min received his B.S. and M.S. degrees from the Department of Mechanical Engineering at Seoul National University in 1986 and 1988, respectively. He obtained his Ph.D. degree from M.I.T in 1994. He is now a Professor in the School of Mechanical and Aerospace Engineering at Seoul National University.