Volumetric efficiency modelling of internal combustion engines based on a novel adaptive learning algorithm of artificial neural networks.

José Manuel Luján, Héctor Climent *, Luis Miguel García-Cuevas, Ausias Moratal

CMT Motores Térmicos, Universitat Politècnica de València, Spain

*Corresponding author: hcliment@mot.upv.es. Telephone: (+34) 96 387 76 50. Postal address: CMT Motores Térmicos. Universitat Politècnica de València. Camino de Vera s/n. 46022. Valencia. Spain.

Abstract

Air mass flow determination is one of the main variables on the control of internal combustion engines. Effectiveness of intake air systems is evaluated through the volumetric efficiency coefficient. Intake air systems characterization by means of physical models needs either significant amount of input data or notable calculation times. Because of these drawbacks, empirical approaches are often used by means of black-box models based on Artificial Neural Networks. As alternative to the standard gradient descendent method an adaptive learning algorithm is developed based on the increase of hidden layer weight update speed. The results presented in this paper show that the proposed adaptive learning method performs with higher learning speed, reduced computational resources and lower network complexities. A parametric study of several Multiple Layer Perceptron (MLP) networks is carried out with the variation of the number of epochs, number of hidden neurons, momentum coefficient and learning algorithm. The training and validation data are obtained from steady state tests carried out in an automotive turbocharged diesel engine.

Keywords

Artificial Neural Networks; Adaptive learning; Diesel engines modeling; Volumetric efficiency.

1. Introduction

Pollutant emissions in automotive diesel engines have become as a major subject of research. Combustion process is affected by the in-cylinder trapped mass and gas composition. Stratified combustion of diesel engines makes necessary a well performance of the air mass flow system. Excess of air plays an important role in the combustion efficiency since it speeds up the mixing of fuel with air and ensures complete combustion of fuel. If sufficient oxygen is not provided to the engine during combustion process, complete conversion of carbon and hydrogen is impossible to attain and that leads to particulates, hydrocarbon and carbon monoxide resulting in increased exhaust emissions [1]. If the excess air ratio is too high, the peak in-cylinder pressure is also relatively high, which has a negative influence on the reliability of the engine [2]. Higher excess of air also carries high NOx emissions and excessive exhaust gas temperature [3].

The performance of the air mass flow system is determined by the volumetric efficiency. Pressure drops, gas temperature increase trough heat transfer from the intake pipes and cylinder walls, gas inertia, overlapping valve and pressure waves at the intake manifold can difficult the cylinder filling [4, 5, 6].

Many authors have studied and modelled the intake air systems with the application of thermo-fluid dynamic governing equations. 1D wave action models are the most popular physical models in the intake air system analysis because of the tradeoff between accuracy and computational cost [7, 8, 9, 10, 11].

However, the thermo and fluid-dynamic processes occurring in an internal combustion engine are so nonlinear and complex that it is usually impossible to model all of them. Modeling based on solving real physical equations governing the engine, although accurate, is too time consuming and not suitable for a control purpose [12]. In addition, the presence of controllers and actuators has increased in the intake engine systems. A variety of new diesel engine air-path actuators, such as variable geometry turbocharger (VGT), two-stage turbo-charging system, single- and dual-loop exhaust gas recirculation (EGR), throttling valves at the intake and exhaust for EGR enhancement and variable valve actuation (VVA), have been recently developed for providing the authorities of controlling the intake manifold gas conditions in both steady-state and transient operations [13]. The added systems involve an important increase of physical model complexity, being necessary a lot of data about such systems performance and therefore making difficult their accurate implementation.

Because of the important non linarites of the thermo and fluid dynamic engine process, several researchers have developed empirical models based on Artificial Neural Networks (ANNs) [14, 15]. Volumetric efficiency is commonly modelled empirically as a black-box function of a combination of engine speed, intake manifold pressure, intake manifold temperature, and exhaust manifold pressure [16]. ANNs have become an important tool in empirical engine process modelling. The Multiple Layer Perceptron (MLP) performs as one of the most popular ANNs architectures in order to implement such models [17, 18, 19].

In this paper a novel gradient descendent ANN model is proposed with the aim to improve the learning process. An adaptive learning rate between network layers is applied in order to increase the learning speed of the network. The outcome of the proposed model is compared to the standard ANNs models showing the improvement obtained with the use of the adaptive model.

The paper is described as follows. Section 2 details the experimental set up carried out at the laboratory showing the features of the measurement equipment. In section 3 the learning speed limitation of backpropagated ANNs is described. An adaptive learning scheme is proposed as solution of learning speed reduction problem. The methodology of ANNs implementation is included in this section. In section 4 results are shown pointing out the optimal ANN architecture and its prediction capacity. Finally, in section 5 conclusions are presented.

2. Experimental methodology

2.1. Test cell description

Experiments with an in line 4 cylinder, 1.6 l, turbocharged HSDI diesel engine were conducted. In Table 1 the features of the engine are shown. The engine was run under steady state conditions at different operating points that covered the whole engine torque and speed range. The engine layout is shown in Fig. 1, where the engine operative variables used for the ANN model implementation are marked with blue arrows in the point of measurement.

Relevant variables needed for the volumetric model implementation were recorded, such as: engine speed, torque, intake manifold pressure, turbine inlet pressure, intake manifold temperature, EGR rate and air mass flow rate. In order to assess the volumetric efficiency according to real engine conditions, EGR was performed depending on the engine load of the different running points.

Engine speed was measured through a KYSTLER encoder with an error of 0.02 Crank Angle Degree (CAD). Engine torque was measured by the dynamometer SCHENK DYNAS3, with an error of 0.1%. Temperatures were measured with thermocouples type K of TCA brand, with a measurement error of 2%. Gas pressure was measured with KISTLER pressure sensors with an error of 0.3%. Air mass flow rate was measured by means of a hot wire anemometer of Sensycon brand, with a measurement error of 1%.

Horiba Mexa 7100 DEGR was used to measure O_2, CO_2, CO, using a non-dispersive infrared analyzer. The error of the gas analyzer is in the range of 2%. Both intake and exhaust CO_2 measurements were recorded in order to obtain the LP EGR rates. The EGR rate is defined as:

$$X_{EGR} = \frac{\dot{m}_{egr}}{\dot{m}_{air} + \dot{m}_{egr}} \qquad [1]$$

where $\dot{m}_{egr}$ and $\dot{m}_{air}$ are the mass flow of EGR gas and fresh air, respectively. Eq. [1] can be expressed as a function of a specific pollutant concentration, like CO_2, measured in the intake and exhaust manifold:

$$X_{EGR}=\frac{[CO_{2\ INT}]-[CO_{2\ ATM}]}{[CO_{2\ EXH}]-[CO_{2\ ATM}]} \qquad [2]$$

where $[CO_{2\ INT}]$, $[CO_{2\ ATM}]$ and $[CO_{2\ EXH}]$ are the carbon dioxide concentration in the intake, ambient and exhaust place respectively.

2.2. Experimental data

71 running points were conducted at different steady engine load conditions of the operative range of an EURO V engine. Fig. 2 shows the EGR rate as a function of the engine speed and torque, where each circle corresponds to an engine operating point. The range of the operation points was as follows: 1250 to 3750 rpm for engine speed, 5 to 317 Nm for engine torque, and 0% to 38% for LP EGR rates. Engine speed Variables at each steady point were obtained from the average of 300 points sampled at 10 Hz. Volumetric efficiency is calculated from:

$$\eta_{vol}=\frac{\dot{m}_{total}}{\frac{P_{intake}}{R\cdot T_{intake}}\cdot V_{cyl}\cdot z\cdot\frac{n}{2}} \qquad [3]$$

where P_{intake} and T_{intake} are inlet manifold pressure and temperature respectively, R is the ideal gas constant, V_{cyl} is the cylinder displacement, z is the number of cylinders and n is the engine speed. $\dot{m}_{total}$is the total mass flow rate coming into the cylinders, calculated as:

$$\dot{m}_{total}=\dot{m}_{air}+\dot{m}_{egr} \qquad [4]$$

$$\dot{m}_{total}=\dot{m}_{air}+\dot{m}_{air}\cdot\frac{X_{EGR}}{1-X_{EGR}} \qquad [5]$$

$$\dot{m}_{total}=\frac{\dot{m}_{air}}{1-X_{EGR}} \qquad [6]$$

3. ANN model

3.1 ANNs basics

An artificial neural network, usually called neural network, is a mathematical model which is inspired by the structure and functional aspects of a biological nervous system. They have been shown to exhibit many abilities, such as learning, generalization and abstraction [20]. The most common network structure used in ANNs is the Multiple Layer Perceptron. MLP network has an input layer, followed by one or more hidden layers and an output layer. Each layer has some artificial neurons (nodes) with their biases (b), a weight matrix (w), and an output vector. A layer of neurons that receives inputs directly from outside the network is called input layer. A layer that produces the output of network is called output layer and layers that are between the input and output layers are called hidden layers [21]. Fig. 3 shows the general layout of an ANN. In the case of a network with one hidden layer the network output is computed according to the following equation, [22, 23].

$$f(x_{in\_1}, x_{in\_2}, \ldots, x_{in\_n}) = \theta\left(b_{21} + \sum_{m=1}^{M} w_{2m} \cdot \theta\left(b_{1j} + \sum_{i=1}^{n} w_{1i} \cdot x_i\right)\right) \qquad [7]$$

Where x_{in} are the input variables of the model. θ is the neural function, b are the biases of neurons, w are the synaptic weights, x are the outputs of the neural functions, n is the number of input variables of the model and M the number of hidden neurons.

The use of Back Propagation (BP) learning method to train feedforward neural networks has been proven to provide powerful tools for analyzing real world and complex problems [24]. However, the BP learning method has some shortcomings such as slow learning speed associated with computational complexity [24, 25]. BP learning is based on the backward propagation of the updating synaptic weights from the output to the input layer. In this paper a descendent gradient BP with momentum is implemented as learning algorithm. Logistic function is used as neuron. Weights are updated according to delta rule with momentum [26]:

$$\Delta w_{ij}{}^{t+1} = -\alpha \cdot \frac{\partial E}{\partial w_{ij}} + \mu \cdot \Delta w_{ij}{}^{t} \qquad [8]$$

where $\Delta w_{ij}{}^{t+1}$ is the increase of the ij weight, α the learning rate, μ is the momentum coefficient, $\Delta w_{ij}{}^{t}$ is the last weight update and E is the network error function defined as:

$$E = \frac{1}{2} \cdot \sum_{p=1}^{P} \left(y_p - o_p\right)^2 \qquad [9]$$

where P is the number of input/output patterns, y_p is the target value of the p-pattern and o_p is the predicted output value of the p-pattern. Delta rule can be obtained as a combination of Eq. [8] and [9]. The final expression, without including the momentum term, is shown in Eq. [10]. It shows the general equation of descendent gradient with backpropagation in a neural network. Its representation in vector notation for a network of L layers is expressed as:

$$[\Delta w]^{l} = -\alpha \cdot \sum_{p=1}^{p=P} \left(y_p - o_p\right) \cdot \left(\prod_{\substack{k=l+1 \\ l<L}}^{k=L} \left[FW^{k}\right]_p \right) \odot \left(\left[\frac{\partial \theta}{\partial z}\right]^{l}_{p} \cdot [x]^{l}{}_{p} \right) \qquad [10]$$

$\odot$ represents the element-wise product of matrices, also known as Hadamard or Schurd product.

l is the layer under study in the network.

$\left[FW^{k}\right]$ is the matrix obtained at the layer k as:

$$\left[FW^{k}\right] = \left[\frac{\partial \theta}{\partial z}\right]^{k} \cdot [w]^{k} \qquad [11]$$

$\left[\frac{\partial \theta}{\partial z}\right]^{k}$ is the derivate of each neuron function with respect to the net neuron input z, calculated at the layer k.

z is the weighted neuron input. For a neuron i of the layer k it is defined as:

$$z_i = b_i + \sum_{j=1}^{n} w_{ij} \cdot x_j \qquad [12]$$

Where b_i is the bias of the neuron i, w_{ij} are the weights that connects the neuron i with the n neurons of the previous layer and x_j is the output of the previous neurons. In the case of the first hidden layer x_j is the input variable of the network.

The product operator multiplies all $[\mathrm{FW}^{\mathrm{k}}]$ matrices of layers located between the layer $l+1$ until the output layer. In case of $l = L$ is defined as the unit.

The last term of the Eq. [10], $\left[\frac{\partial\theta}{\partial z}\right]^l \cdot [\mathrm{x}]^l$, belongs to the layer that is being updated.

Neuron biases are calculated by the application of the gradient descendent rule too. The expression of biases, Eq. [13], is quite similar than the weights, Eq. [10].

$$[\Delta \mathrm{b}]^l = -\ \alpha\cdot \sum_{\mathrm{p=1}}^{\mathrm{p=P}}(\mathrm{y_p}\text{-}\mathrm{o_p})\cdot\left(\prod_{\substack{\mathrm{k}=l+1\\ \mathrm{l<L}}}^{\mathrm{k=L}}[\mathrm{FW^k}]_{\mathrm{p}}\right)\odot\left(\left[\frac{\partial\theta}{\partial z}\right]^l_{\mathrm{p}}\right) \qquad [13]$$

3.2 Adaptive learning

The learning schema shown in the previous section entails different learning speeds between layers. As the number of hidden layers increases, the term $[\mathrm{FW}^{\mathrm{k}}]$ reduces because of the logistic neural function derivate is bounded between 0-0.2. This fact drives to the phenomenon named as “vanishing gradient problem” [27, 28]. Vanishing produces lower update velocities in the weights that belong to shallow layers (layers near to the network input). As solution to the different learning speed is proposed an adaptive learning rate fit by means of the layer depth in the network. In this paper, allusions to this adaptive learning rate are referred to as ADDELE (ADaptive DEpth LEarning). The aim of this learning is to make equal the learning speed between layers through the application of different learning rates. Eq. [10] can be written as:

$$[\Delta \mathrm{w}]^l = -\ \alpha\cdot \sum_{\mathrm{p=1}}^{\mathrm{p=P}}(\mathrm{y_p}\text{-}\mathrm{o_p})\cdot \mathrm{O}\left(\mathrm{h}^{\mathrm{L}\text{-}l}\right)_{\mathrm{p}}\odot\left(\mathrm{O(h)_p}\cdot[\mathrm{x}]^l{}_{\mathrm{p}}\right) \qquad [14]$$

Where $O\left(h^{L-l}\right)$ is the function that represents the product operator and $O(h)$ the $\frac{\partial\theta}{\partial z}$ term of the Eq. [10]. The delta weight function at the hidden layers is expressed as:.

$$\Delta \mathrm{w}^{\mathrm{l}} = \ \mathrm{O}\left(\mathrm{h}^{\mathrm{L}\text{-}l+1}\right) \qquad [15]$$

In case of the output layer, $\cdot \mathrm{O}\left(\mathrm{h}^{\mathrm{L}-l}\right)_{\mathrm{p}} = 1$, the delta weight expression is reduced to the following equation:

$$\Delta \mathrm{w}^{\mathrm{L}} = \mathrm{O}(\mathrm{h}) \qquad [16]$$

Despite the terms $\left[\mathrm{FW}^{\mathrm{k}}\right]_{\mathrm{p}}$ are not bounded, they usually take values lower than unit, producing the aforementioned vanishing problem at the hidden layers. In order to reduce vanishing, the weight update function of the hidden layer l, Eq. [15], is forced to have the same degree as the weight function of the output layer, Eq. [16].

As $\mathrm{O}\left(\mathrm{h}^{\mathrm{L}-l+1}\right)_{\mathrm{p}}$ is defined by each pattern, the minimum function of the whole patterns is calculated and used as reference function for the adaptive learning calculation:

$$\mathrm{O}\left(\mathrm{h}^{\mathrm{L}-l+1}\right) = \min\left(\prod_{\substack{\mathrm{k}=\mathrm{l}+1 \\ \mathrm{l}<\mathrm{L}}}^{\mathrm{k}=\mathrm{L}} \left[\mathrm{FW}^{\mathrm{k}}\right]_{\mathrm{p}} \cdot \left[\frac{\partial \theta}{\partial \mathrm{z}}\right]^{l}_{\mathrm{p}}\right), \mathrm{p} = \{0,1,2 \ldots \mathrm{P}\} \qquad [17]$$

Once $\mathrm{O}\left(\mathrm{h}^{\mathrm{L}-l+1}\right)$ is defined by the network layer, the new learning rate coefficient is calculated at each hidden layer:

$$\alpha_L \cdot \mathrm{O}\left(\mathrm{h}^{\mathrm{L}-l+1}\right)^{\frac{1}{\mathrm{L}-l+1}} = \alpha_l \cdot \mathrm{O}\left(\mathrm{h}^{\mathrm{L}-l+1}\right) \qquad [18]$$

$$\alpha_l = \alpha_{\mathrm{L}} \cdot \frac{\mathrm{O}\left(\mathrm{h}^{\mathrm{L}-l+1}\right)^{\frac{1}{\mathrm{L}-l+1}}}{\mathrm{O}\left(\mathrm{h}^{\mathrm{L}-l+1}\right)} \qquad [19]$$

$$\alpha_l = \alpha_L \cdot \mathrm{O}\left(\mathrm{h}^{\mathrm{L}-l+1}\right)^{\frac{l-\mathrm{L}}{\mathrm{L}-l+1}} \qquad [20]$$

where α_{L} is the learning rate of the output layer and α_l is the learning rate of the hidden layer that is under study.

As $\mathrm{O}\left(\mathrm{h}^{\mathrm{L}-l+1}\right)$ is a function obtained from the minimum of patterns, it can drive, in some conditions, to excessive learning rates. It was observed that too high learning rates, α_l, can be obtained at the first epochs of learning due to the sensibility to the random weight initialization. In order to avoid instabilities the α_l is bounded to 0.5. The learning rate limitation was defined by trial and error procedure. The same procedure can be applied in bias update correction obtaining the same mathematical expression than Eq. [20].

3.3 ANN methodology

In this study a one hidden layer, feed-forward, neural network is implemented. Logistic function is used as neuron activation function at the hidden and output layers.

80% of the whole experimental data is used for training the ANN and the other 20% is used for model validation. The division of data in subsets is randomly obtained according to cross random validation technique [29].

The input model variables are: engine speed, torque, intake manifold pressure, turbine inlet pressure and intake manifold temperature. Model input variables were selected according to the bibliography available in the field of volumetric efficiency modelling applied to internal combustion engines [15, 17, 23, 30]. Volumetric efficiency is defined as output variable. Variables are normalized according to min-max scaling:

$$x_n = \frac{x_i - \min(x)}{\max(x) - \min(x)} \qquad [21]$$

where x_n is the normalized value of x_i in the range of [0,1].

Training process is analyzed by means of the R^2 coefficient [31], defined as:

$$R^2 = 1 - \frac{\sum_{p=1}^{p=P} \left(y_p - o_p\right)^2}{\sum_{p=1}^{p=P} \left(y_p - \bar{y}_p\right)^2} \qquad [22]$$

where y_p is the target value, o_p is the network output of each pattern and $\bar{y}_p$ is the mean of target values.

Validation process is evaluated by the maximum relative error of the 95% of validation samples, denoted as “error 95” in this paper. In addition the Mean Absolute Percentage Error (MAPE) is included as validation coefficient [32] and it is defined as:

$$\mathrm{MAPE} = \frac{1}{P} \cdot \sum_{P=1}^{p=P} \left| \frac{y_p - o_p}{y_p} \right| \cdot 100 \qquad [23]$$

However, the analysis of validation only by means of MAPE can drive to ANNs with outliers that would be neglected by this coefficient.

ANNs architecture selection is obtained from a parametric study of one hidden layer neural network. The number of hidden neurons covers the range between 6 to 16. The momentum coefficient, Eq. [8], is varied between 0.7 to 0.9. Fixed and adaptive (ADDELE) learning is included as a variable of the parametric study. The risk of overfitting, [33, 34], mainly at high number of epochs as well as high number of neurons, makes necessary to include the early stop technique [35, 36]. The number of training epochs is included in the study in the range of 200 to 15000 epochs. Because of the important computational cost, all available combinations are not tested and the number of different ANNs architectures proposed was 504.

The cross validation procedure for each ANN architecture, defined by the parametric study, is repeated 40 times to reduce random initialization issues. Learning and validation performance is obtained from the average of the 40 ANNs set. Therefore, the total number of ANNs implemented in this parametric study is 20160.

4. Results and discussion

ANNs learning performance is represented in Fig. 4 (a-c), for the momentum coefficients of 0.7, 0.8 and 0.9 respectively. Each chart shows the learning capacity (R^2) of both fixed and ADDELE algorithm. R^2 is calculated from the average of the 40 repetitions of each ANN. The scatter of points is interpolated according to the Delaunay triangulation by means of matplotlib programming libraries [37]. The colorbar legend represents the R^2 value at the interpolated surface.

The result of the interpolation shows two different surfaces per chart, where the top surface always belongs to the ADDELE schema. This fact points out the higher learning prediction of ADDELE compared to the fixed learning. The learning R^2 evolves along the epochs number as increasing monotonic function. The increase in the learning outcome with the number of epochs is notable mainly when the variation of epochs is done in the low number of epochs zone, as observed by regarding the slope of the surface between number of epochs and R^2. The increase of R^2 with the number of hidden neurons is noticed too, but with lower impact than the number of epochs.

Regarding the difference between figures due to the momentum term, the higher the momentum is the better the learning performs. Higher values of R^2, as well as higher learning speeds , are obtained with the increase of momentum.

However, the ANNs performance cannot be analyzed only by means of the learning outcome. Both learning and validation performance have to be analyzed together. Fig. 5 shows the relation between learning and validation of the whole ANN set formed by the 504 different networks. Each mark in the chart represents the average learning R^2 and error 95 (Error 95) of the 40 repeated networks. According to the figure, as it was shown in Fig. 4, it is obtained higher learning R^2 with ADDELE, compared to the fixed learning. ADDELE increases the velocity of learning and therefore, for the same network architecture, ANNs with higher R^2 are obtained. In addition to the learning speed it is remarkable the tendency observed between learning and validation performance. As the learning capacity increases the validation error (error 95) increases too. This tendency is the proof of the overfitting phenomenon, where accuracy and generalization are faced. This outcome matches the conclusions obtained by several researchers [38, 39].

The ANNs selection is a tradeoff between the learning and validation performance. In order to carry out this selection an objective function that relates both coefficients, R^2 and error 95, is defined. The objective function, named as Normalized Error Function (NEF), represents the global error of both learning and validation. NEF is calculated by each ANN (denoted as $\mathrm{NEF\_i}$) that has a R^2 higher than 0.75 with the expression:

$$\mathrm{NEF\_i}=\mathrm{C_{R2}}\cdot\left(1-\mathrm{R}^2_{\mathrm{norm\_i}}\right)+(1-\mathrm{C_{R2}})\cdot\mathrm{E}^{95}_{\mathrm{norm\_i}}, R_i{}^2 \geq 0.75 \qquad [24]$$

where $\mathrm{C_{R2}}$ is the weight of the learning term, bounded between 0 – 1. $\mathrm{R}^2_{\mathrm{norm\_i}}$ and $\mathrm{E}^{95}_{\mathrm{norm\_i}}$ are the normalized R^2 and error 95 respectively by each ANN:

$$R^2_{norm\_i} = \frac{R_i^2 - \min([R^2])}{\max([R^2]) - \min([R^2])} \quad [25]$$

$$E^{95}_{norm\_i} = \frac{E_i^{95} - \min([E^{95}])}{\max([E^{95}]) - \min([E^{95}])} \quad [26]$$

where R_i^2 is the determination coefficient by ANN and $[R^2]$ is the vector of R_i^2 that are higher than 0.75. E_i^{95} is the error 95 of each ANN validation and $[E^{95}]$ is a vector of E_i^{95} of ANNs whose R_i^2 are higher than 0.75.

The C_{R2} coefficient of the objective function NEF is fitted to 0.4 in order to give more importance to the generalization capacity of neural networks.

Table 2 shows the best 5 ANNs architectures of the whole simulated set. The first two ANNs at the top of that table show similar NEF values. The first one with the ADDELE schema while the second with the fixed algorithm. Despite similar global performance is achieved, the number of hidden neurons as well as the number of learning epochs needed in case of ADDELE is significantly lower than the fixed learning. Learning calculation time depends on the number of training epochs as well as on the number of mathematical operations performed per epoch. The calculation time is proportional to the number of learning epochs because of the calculation time by epoch remains constant during the training process. Regarding the mathematical operations by epoch, they can be divided in sums, products and neuron function evaluations. According to the backpropagation learning, Eq.[10] and Eq.[13], the weights and biases update requires firstly an evaluation of the network output and then, the application of the chain rule from the output layer to the network layer where the weight under update is placed.

The number of operations to evaluate the output of an ANN model is defined by the number of hidden layers, neurons and model input variables. In the case of one hidden layer neural networks with one neuron at the output layer the number of both sums and multiplications is as follows:

$$\chi_+ = \chi_x = (n+1) \cdot M \quad [27]$$

Where χ_+ and χ_x represent the number of sums and multiplications, n is the number of inputs to the network and M is the number of hidden neurons. The number of neurons function evaluations is:

$$\chi_N = M + 1 \qquad [28]$$

Where χ_N is the number of neuron function evaluations. From the sum of the three kind of operations is obtained:

$$\chi_T = 2 \cdot (n + 1) \cdot M + (M + 1) \qquad [29]$$

Where χ_T represents the total number of operations for network output calculation. As the number of inputs is not modified between the different ANN performed in this paper, the total number of operations can be expressed as a function only dependent on the number of hidden neurons. In the particular case of six input variables the total number of operations for network output evaluation can be expressed as:

$$\chi_T = 15 \cdot M + 1 \qquad [30]$$

The number of operations per layer carried out by the application of the chain rule to update the weights and biases in a one hidden layer network can be expressed as:

$$\chi_x = \sum_{l=1}^{l=L-1} (L - l + 2) \cdot M_{l-1} \cdot M_l \qquad \text{Weights.} \qquad [31]$$

$$\chi_x = \sum_{l=1}^{l=L-1} (L - l + 1) \cdot M_l \qquad \text{Biases.} \qquad [32]$$

Where L is the total number of layers in the network (input, hidden and output), l is the layer number where the weight is placed, M is the number of neurons in a layer. The learning rate used for weight velocity update control is included in the above expressions. In the case presented in this paper a six input neurons networks is propose, so the total number of operations according to the chain rule required by the backpropagation technique is:

$$\chi_T = 32 \cdot M \qquad [33]$$

The above expression is slightly higher in case of the ADDELE learning application, because of the learning rate (α) has to be calculated by layer at each training epoch. The number of multiplications in the case of ADDELE learning is:

$$\chi_T = 32 \cdot M + 2 \quad [34]$$

Considering the Eq [30], Eq [33] and Eq [34] the total amount of operations is as follows.

$$\chi_{\text{learning}} = 47 \cdot M + 1 \text{ Fixed learning} \quad [35]$$

$$\chi_{\text{learning}} = 47 \cdot M + 3 \text{ ADDELE learning} \quad [36]$$

In case of neural networks with 6 hidden neurons or more the above expressions can be approximated, with an error lower than 1%, to:

$$\chi_{\text{learning}} = 47 \cdot M \quad [37]$$

Therefore, the calculation time can be considered as proportional to the number of hidden neurons and epochs.

Comparing the two best ANN, placed at the top of the Table 2, the total reduction of time by using the ADDELE algorithm is 57%. With the ADDELE algorithm the number of learning epochs has been reduced to the half and the number of hidden neurons have been reduced in two. So, taking into account the lower complexity and calculation time of the ADDELE ANN, it is selected as the optimal network.

The performance of the selected network, 6-12-1 topology, momentum=0.9, learning epochs=6000 with ADDELE algorithm, is shown in Fig. 6, 7, 8 and 9. The best fixed ANN learning (6-14-1 topology, momentum=0.9, learning epochs=12000) is included in the charts too.

Fig. 6 is a goodness of fit chart where network outcomes of the training data are plotted against the respective target values. Dotted lines represent the 5% relative error above and below the measured value. Higher dispersion is observed in case of fixed learning schema, which shows higher amount of points that are closer or outside the boundary of the 5% relative error. In Fig. 7 the validation goodness of fit is depicted. The network makes predictions of data that have not been shown during the learning process and therefore the accuracy of prediction reduces. The coefficient of determination in the case of validation data is 0.3 for both fixed and adaptive learning algorithms. The reason of this low fitting value is twofold: on one hand, is caused by the low range of the target variable since the volumetric efficiency is constrained to values between 0.73-1 with high density of points (50% of total data) between 0.8-0.9. The lower the natural variability is the higher the accuracy of a model has to be in order to obtain high determination coefficients. On the other hand, the complex non linear relation between the input variables and the volumetric efficiency makes difficult the validation process. For getting higher determination coefficients the data sample size for the training process would need to be significantly higher, which for internal combustion engines experimental testing involves high cost on laboratory measurements.

Fig. 8 shows the distribution of the relative error in ANN validation. The histogram error roughly meets the characteristic normal distribution in both learning schemas: fixed and ADDELE. Anomalous pattern is observed at the positive tail edge of the normal distribution, where the error is the highest and reaches the 25%. The worst performance of both models occurs in the same experimental point. The poor prediction at this point is observed in the different repetitions of ANNs. These two facts drive to consider this experimental point as an outlier because the high prediction error has no dependency on the model features. The prediction error without considering the aforementioned outlier is shown in Fig. 9 where the error pattern fits better to the normal distribution and the maximum error is reduced to around 20%.

5. Conclusions

The application of a neural network method to predict the volumetric efficiency in a diesel engine has been examined. Experimental data of an engine map were used as input variables. Measurement points were carried out at steady state conditions.

A cross validation methodology based on the minimization of an objective function was proposed. The learning and validation performance was analyzed together.

An adaptive learning algorithm based on the application of different learning rates between network layers as a way to enhance the learning network speed was proposed. The results of the optimization procedure showed that ADDELE architectures performs with higher learning speed without sacrificing prediction capacity than fixed networks.

Despite vanishing phenomenon has higher impact on deep neural networks, the effect on learning speed is remarkable even in the architecture of one hidden layer analyzed in this paper. The learning acceleration through ADDELE drives to lower computational costs and lower network complexities.

The maximum generalization error of the neural network, according to the validation analysis, was bounded to around 13% with an average relative error of 5.5%. The learning coefficient of determination was 85%.

Aknowledgments

Authors want to acknowledge the “Apoyo para la investigación y Desarrollo (PAID)” ,grant for doctoral studies (FPI S1 2015 2512), of Universitat Politècnica de València.

References

[1]. P. Baskar, A. Senthilkumar, Effects of oxygen enriched combustion on pollution and performance characteristics of a diesel engine, Engineering Science and Technology, an International Journal, Volume 19, Issue 1, March 2016, Pages 438-443.

[2]. XiangRong Li, Haiqin Zhou, Liwang Su, Yanlin Chen, Zhenyang Qiao, FuShui Liu, Combustion and emission characteristics of a lateral swirl combustion system for DI diesel engines under low excess air ratio conditions, Fuel, Volume 184, 15 November 2016, Pages 672-680.

[3]. Shah, A., Tunestal, P. and Johansson, B., Investigation of Performance and Emission Characteristics of a Heavy Duty Natural Gas Engine Operated with Pre-Chamber Spark Plug and Dilution with Excess Air and EGR. SAE Int. J. Engines 5(4):2012, doi:10.4271/2012-01-1980.

[4]. Haitham Mezher, David Chalet, Jérôme Migaud, Pascal Chesse, Frequency based approach for simulating pressure waves at the inlet of internal combustion engines using a parameterized model, Applied Energy, Volume 106, June 2013, Pages 275-286.

[5]. Haitham Mezher, David Chalet, Jéròme Migaud, Pascal CHesse. The Application of a Wave Action Design Technique with Minimal Cost on a Turbocharged Engine Equipped with Water Cooled Charge Aimed for Energy Management. TerraGrenn 13 International Conference 2013 – Advancements in Renewable Energy and Clean Environment. Energy Procedia 36 (2013) 948 – 957.

[6]. Jensen Samuel, Prasad NS, and Kumarasamy Annamalai. Effect of Variable Length Intake Manifold on aTurbocharged Multi-Cylinder Diesel Engine. 2013 SAE International.

[7]. T. Cerri, A. Onorati, E. Mattarelli. 1D Engine Simulation of a Small HSDI Diesel Engine Applying a Predictive Combustion Model. Journal of engineering for gas turbines and power. 2007.

[8]. Xiangrui Zeng, Junmin Wang, A physics-based time-varying transport delay oxygen concentration model for dual-loop exhaust gas recirculation (EGR) engine air-paths, Applied Energy, Volume 125, 15 July 2014, Pages 300-307.

[9]. José Manuel Luján, Héctor Climent, Pablo Olmeda, Víctor Daniel Jiménez, Heat transfer modeling in exhaust systems of high-performance two-stroke engines, Applied Thermal Engineering, Volume 69, Issues 1–2, August 2014, Pages 96-104.

[10]. Dariusz Cieslar, Paul Dickinson, Alex Darlington, Keith Glover, Nick Collings, Model based approach to closed loop control of 1-D engine simulation models, Control Engineering Practice, Volume 29, August 2014, Pages 212-224.

[11]. L. Cornolti, A. Onorati, T. Cerri, G. Montenegro, F. Piscaglia, 1D simulation of a turbocharged Diesel engine with comparison of short and long EGR route solutions, Applied Energy, Volume 111, November 2013, Pages 1-15.

[12]. Amir-Mohammad Shamekhi, Amir H. Shamekhi, A new approach in improvement of mean value models for spark ignition engines using neural networks, Expert Systems with Applications, Volume 42, Issue 12, 15 July 2015, Pages 5192-5218.

[13]. Fengjun Yan, Junmin Wang, Control of diesel engine dual-loop EGR air-path systems by a singular perturbation method, Control Engineering Practice, Volume 21, Issue 7, July 2013, Pages 981-988.

[14]. Richard Fiifi Turkson, Fuwu Yan, Mohamed Kamal Ahmed Ali, Jie Hu, Artificial neural network applications in the calibration of spark-ignition engines: An overview, Engineering Science and Technology, an International Journal, Volume 19, Issue 3, September 2016, Pages 1346-1359.

[15]. Abdullah Uzun, Air mass flow estimation of diesel engines using neural network, Fuel, Volume 117, Part A, 30 January 2014, Pages 833-838.

[16]. L Kocher, E Koeberlein, D G Van Alstine, K Stricker, and G Shaver. Physically based volumetric efficiency model for diesel engines utilizing variable intake valve actuation. International Journal of Engine Research (2011).

[17]. Jamil El Hadef, Guillaume Colin, Vincent Talon, Yann Chamaillard. Neural Model for Real-Time Engine Volumetric Efficiency Estimation. SAE paper (2013).

[18]. Hoffmann, K., Seebach, D., Pischinger, S., & Abel, D. (2007). Neural Networks for Modelling and Controlling Future Low Temperature Combustion Technologies. IFAC Proceedings Volumes, 40(21), 19-24.

[19]. Arsie, I., Di Iorio, S., Pianese, C., Rizzo, G., & Sorrentino, M. (2008). Recurrent neural networks for air-fuel ratio estimation and control in spark-ignited engines. IFAC Proceedings Volumes, 41(2), 8508-8513.

[20]. Bolan Liu, Changlu Zhao, Fujun Zhang, Tao Cui, Jianyun Su, Misfire detection of a turbocharged diesel engine by using artificial neural networks, Applied Thermal Engineering, Volume 55, Issues 1–2, June 2013, Pages 26-32.

[21]. Sadra Azizi, Ebrahim Ahmadloo, Prediction of heat transfer coefficient during condensation of R134a in inclined tubes using artificial neural network, Applied Thermal Engineering, Volume 106, 5 August 2016, Pages 203-210.

[22]. Yaşar Önder Özgören, Selim Çetinkaya, Suat Sarıdemir, Adem Çiçek, Fuat Kara, Predictive modeling of performance of a helium charged Stirling engine using an artificial neural network, Energy Conversion and Management, Volume 67, March 2013, Pages 357-368.

[23]. De Nicolao, G., R. Scattolini, and C. Siviero. Modelling the volumetric efficiency of IC engines: parametric, non-parametric and neural techniques. Control Engineering Practice 4.10 (1996): 1405-1415.

[24]. J. T. Lalis, B. D. Gerardo and Y. Byun. An Adaptive Stopping Criterion for Backpropagation Learning in Feedforward Neural Network. International Journal of Multimedia and Ubiquitous Engineering Vol.9, No.8 (2014), pp.149-156.

[25]. Wilamowski, B. M., & Yu, H. (2010). Neural network learning without backpropagation. IEEE Transactions on Neural Networks, 21(11), 1793-1803.

[26]. M. Mohanraj, S. Jayaraj, C. Muraleedharan, Applications of artificial neural networks for refrigeration, air-conditioning and heat pump systems—A review, Renewable and Sustainable Energy Reviews, Volume 16, Issue 2, February 2012, Pages 1340-1358.

[27]. Hochreiter, S. (1998). The vanishing gradient problem during learning recurrent neural nets and problem solutions. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 6(02), 107-116.

[28]. Stefan Glüge, Ronald Böck, Günther Palm, Andreas Wendemuth, Learning long-term dependencies in segmented-memory recurrent neural networks with backpropagation of error, Neurocomputing, Volume 141, 2 October 2014, Pages 54-64.

[29]. Devon K. Barrow, Sven F. Crone, Cross-validation aggregation for combining autoregressive neural network forecasts, International Journal of Forecasting, Volume 32, Issue 4, October–December 2016, Pages 1120-1137.

[30]. Desantes, J. M., et al. Air mass flow estimation in turbocharged diesel engines from in-cylinder pressure measurement. Experimental Thermal and Fluid Science 34.1 (2010): 37-47.

[31]. Jahirul, M. I., Senadeera, W., Brooks, P., Brown, R. J., Situ, R., Pham, P. X., & Masri, A. R. (2013, December). An artificial neutral network (ANN) model for predicting biodiesel kinetic viscosity as a function of temperature and chemical compositions. In MODSIM2013, 20th International Congress on Modelling and

Simulation (pp. 1561-1567). Modelling and Simulation Society of Australia and New Zealand.

[32]. Fatih Şahin, Effects of engine parameters on ionization current and modeling of excess air coefficient by artificial neural network, Applied Thermal Engineering, Volume 90, 5 November 2015, Pages 94-101.

[33]. Raymond Liu, Duncan F. Gillies, Overfitting in linear feature extraction for classification of high-dimensional image data, Pattern Recognition, Volume 53, May 2016, Pages 73-86.

[34]. Katsuyuki Hagiwara, Kenji Fukumizu, Relation between weight size and degree of over-fitting in neural network regression, Neural Networks, Volume 21, Issue 1, January 2008, Pages 48-58.

[35]. Sarle, W. S. (1996). Stopped training and other remedies for overfitting. Computing science and statistics, 352-360.

[36]. Adam P. Piotrowski, Jarosław J. Napiorkowski, A comparison of methods to avoid overfitting in neural networks training in the case of catchment runoff modelling, Journal of Hydrology, Volume 476, 7 January 2013, Pages 97-111.

[37]. Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. Computing in science and engineering, 9(3), 90-95.

[38]. Hunter, D., Yu, H., Pukish III, M. S., Kolbusz, J., & Wilamowski, B. M. (2012). Selection of proper neural network sizes and architectures—a comparative study. IEEE Transactions on Industrial Informatics, 8 (2), 228-240.

[39]. Tetko, I. V., Livingstone, D. J., & Luik, A. I. (1995). Neural network studies. 1. Comparison of overfitting and overtraining. Journal of chemical information and computer sciences, 35 (5), 826-833.

Appendix A. tables.

Table 1

Engine specifications.

Cylinder number	In-line 4
Bore x stroke (mm)	80 x 79.5
Displacement (cm3)	1598
Compression ratio	15.4:1
Valve number	16
Valvetrain	Double cam shaft over head
Fuel delivery system	Common rail. Direct injection.
EGR system	HP and LP cooled EGR
Intake boosting	Turbocharger with VGT
Maximum power (kW/rpm)	96/4000
Maximum torque (Nm/rpm)	320/1750
Torque at maximum power (Nm)	315
Specific power (kW/liter)	60.86

Table 2.

Parametric study outcome. Ranking of ANNs by NEF

Hidden neurons	Momentum	Epochs	R^2	Error 95 (%)	MAPE (%)	Learning algorithm	NEF
12	0.90	6000	0.85	12.92	5.52	ADDELE	0.333
14	0.90	12000	0.79	12.01	4.95	FIXED	0.340
14	0.70	9000	0.77	11.68	5.06	ADDELE	0.351
8	0.80	15000	0.80	12.26	5.29	ADDELE	0.356
16	0.90	7000	0.89	13.86	5.8	ADDELE	0.360

Appendix B. Figures.

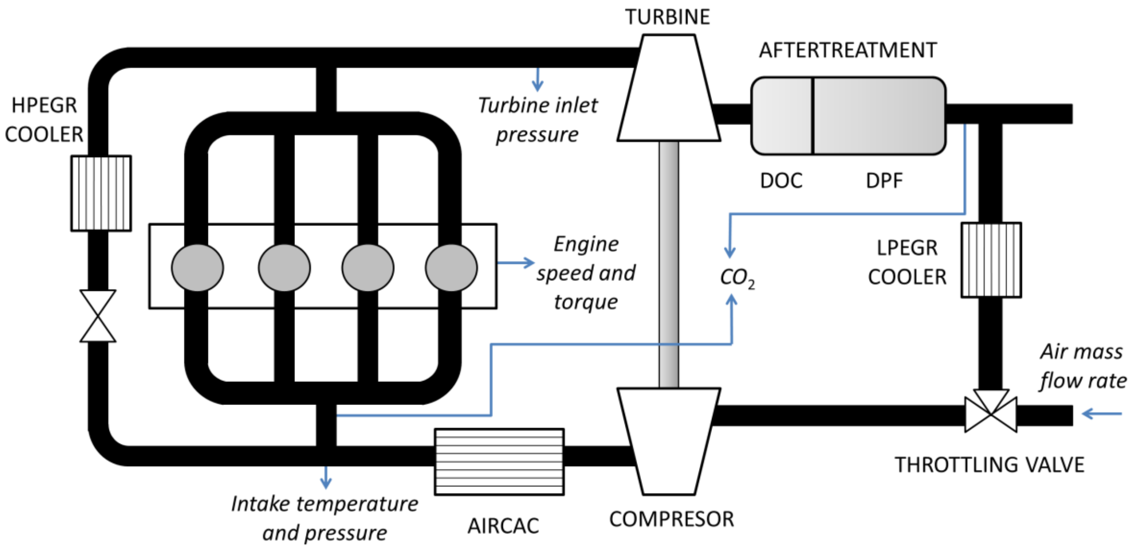

Figure 1. Engine layout. In capital letters the main components of the engine are indicated. In italics and pointed with blue arrows the measured operative engine variables used as model inputs are denoted.

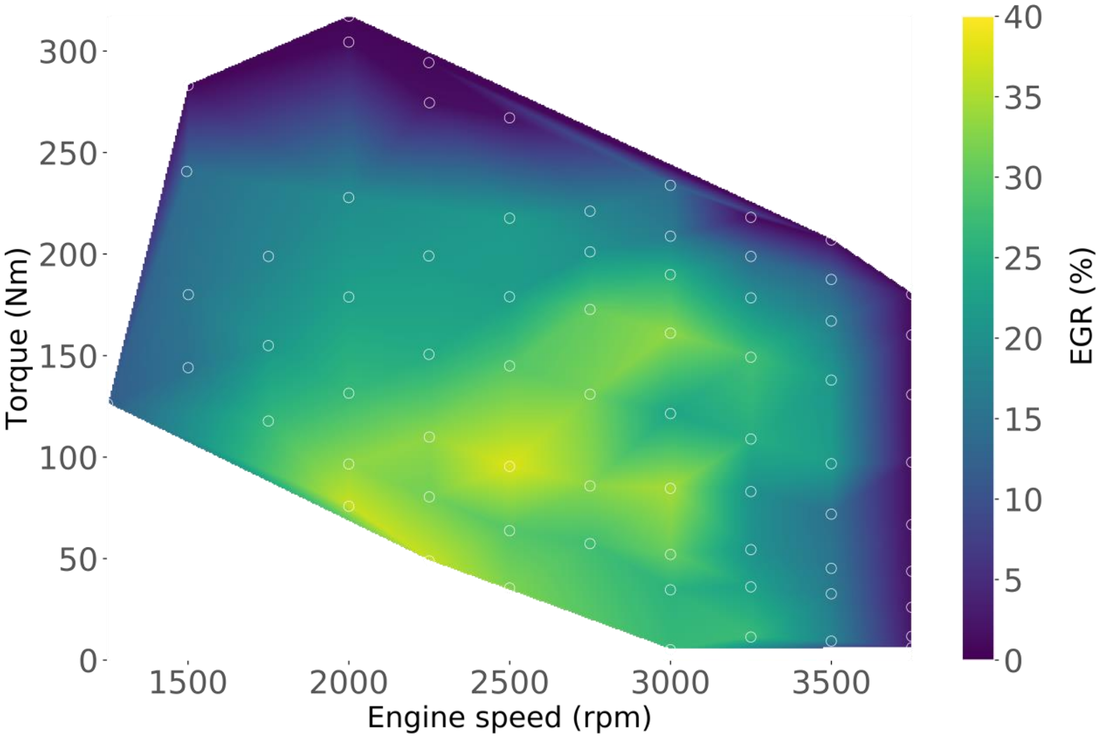

Figure 2. Engine map for engine speed torque and EGR rate.

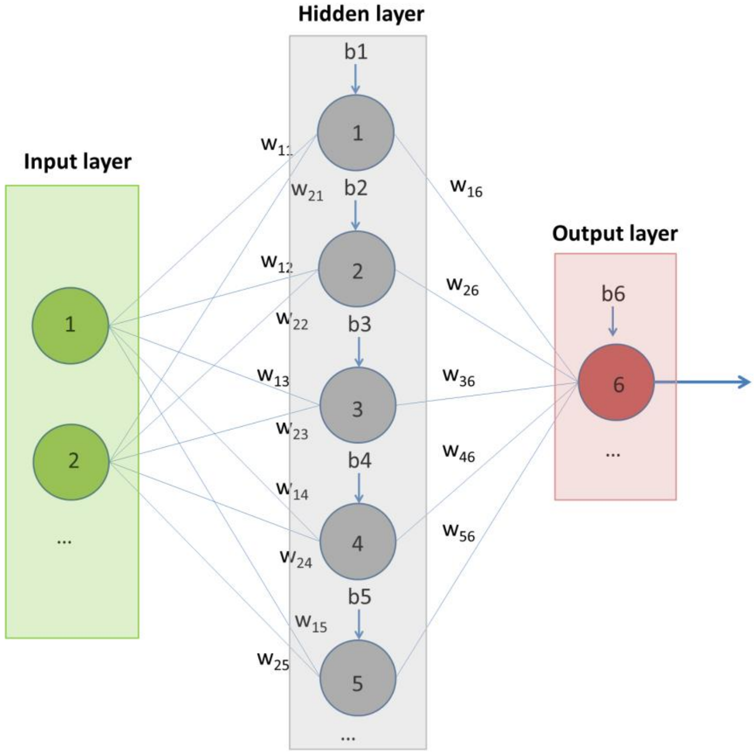

Figure 3. ANN general layout.

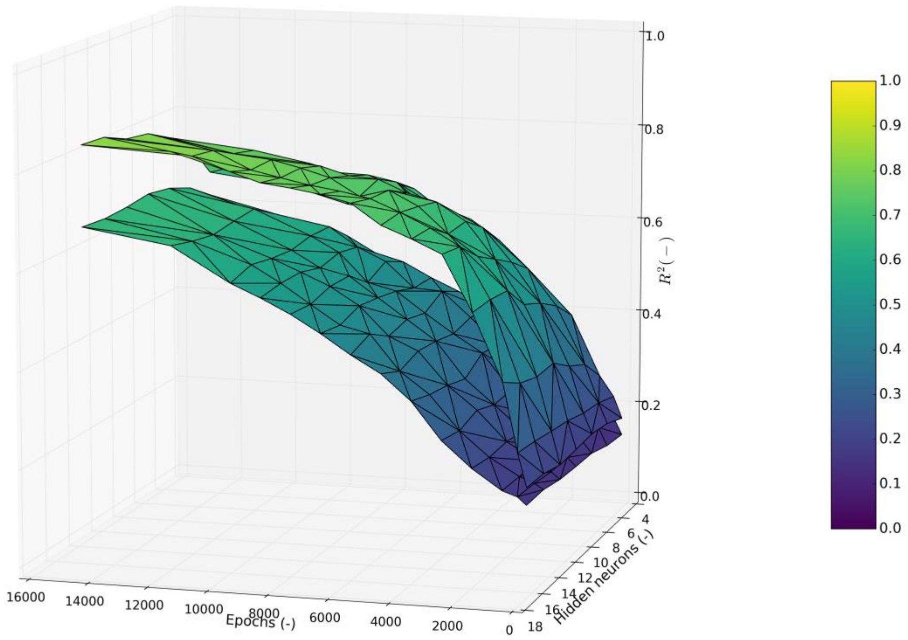

Figure 4(a). ANNs learning outcome for moment term of 0.7.

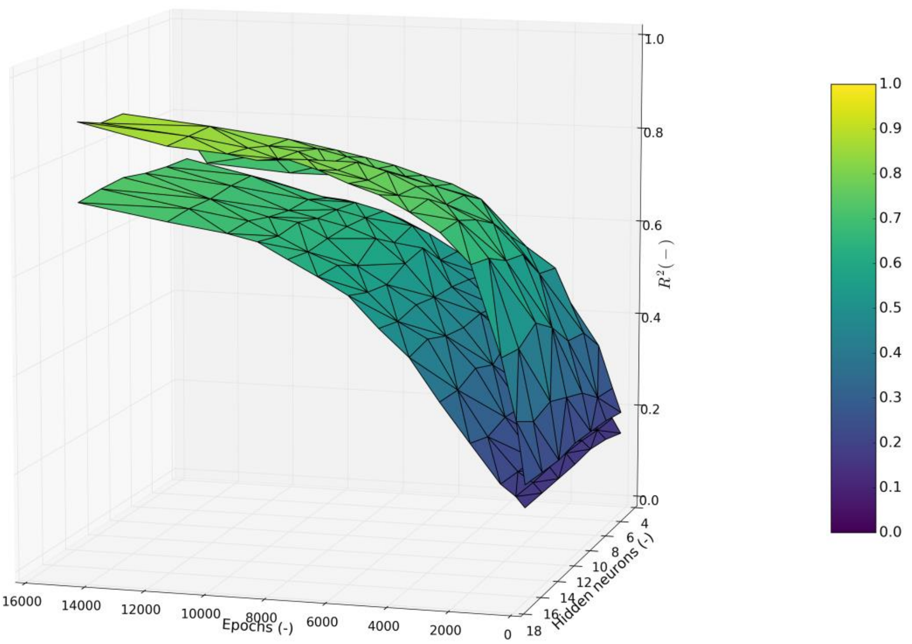

Figure 4(b). ANNs learning outcome for moment term of 0.8.

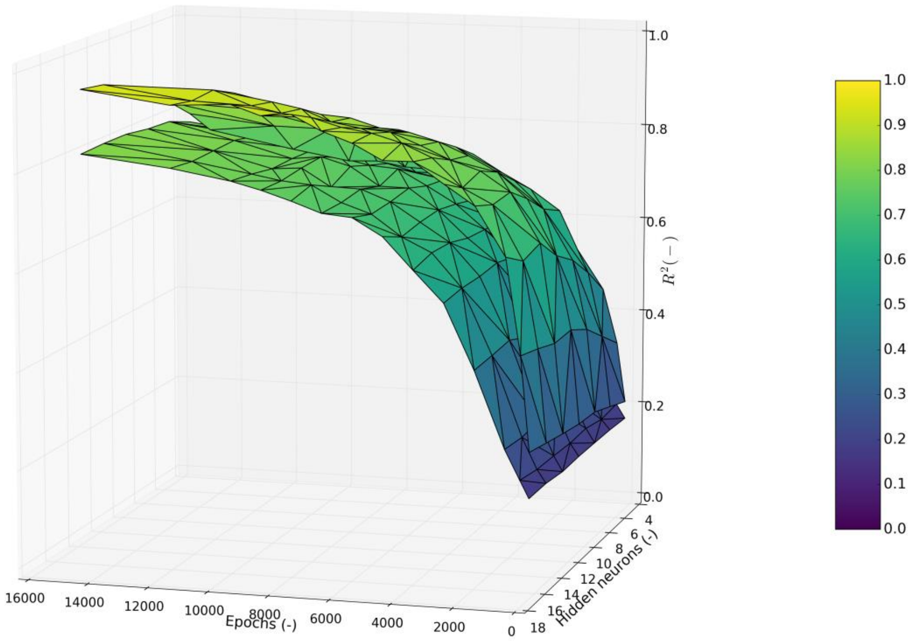

Figure 4(c). ANNs learning outcome for moment term of 0.9.

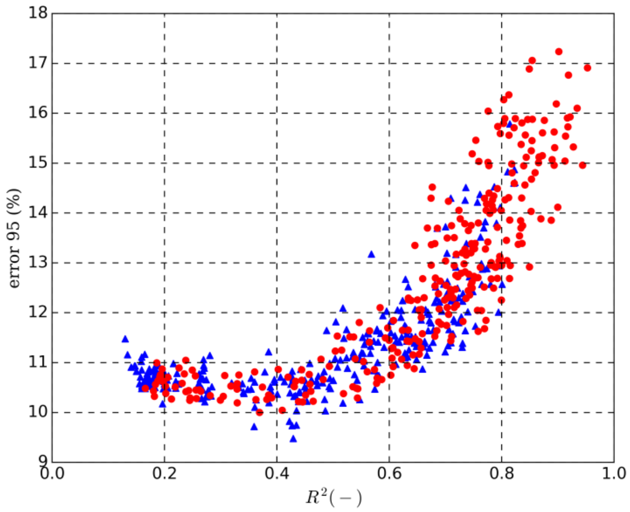

Figure 5. Parametric study. Validation (error 95) and learning (R^2) of the different ANNs. In red circles adaptive learning (ADDELE), in blue triangles fixed learning.

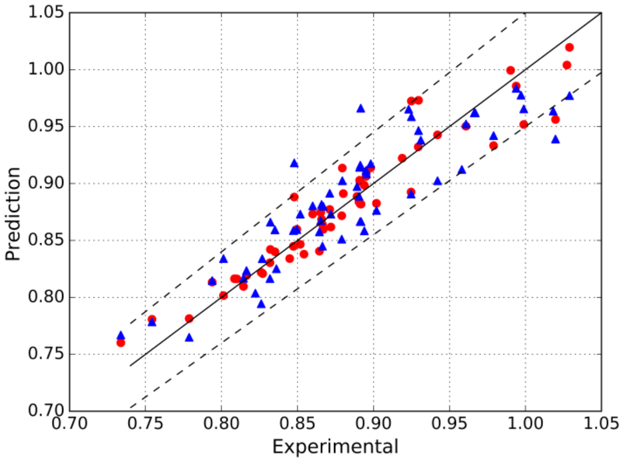

Figure 6. Predicted and measured volumetric efficiency. In red circles ADDELE algorithm. In blue triangles fixed learning schema.

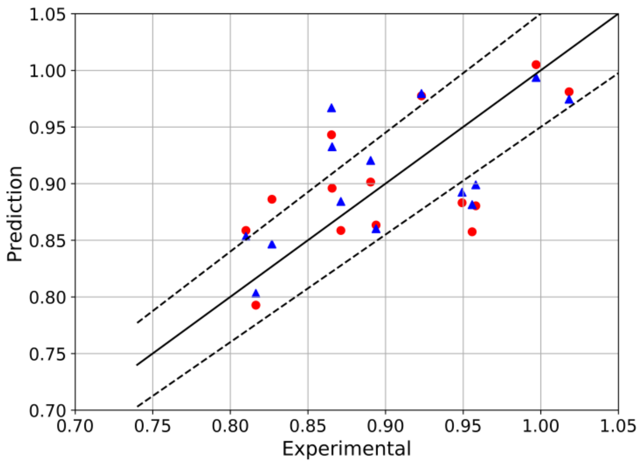

Figure 7. Predicted and measured volumetric efficiency validation data. In red circles ADDELE algorithm. In blue triangles fixed learning schema.

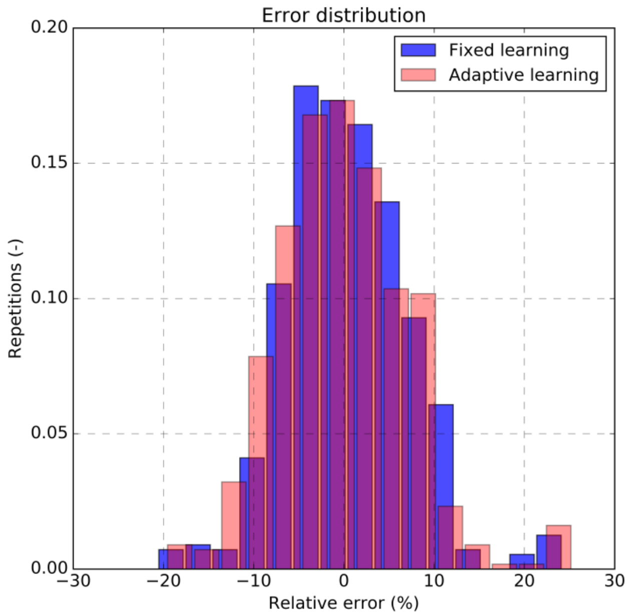

Figure 8. Validation relative error distribution of the whole validation sample. In red ADDELE algorithm. In blue fixed learning.

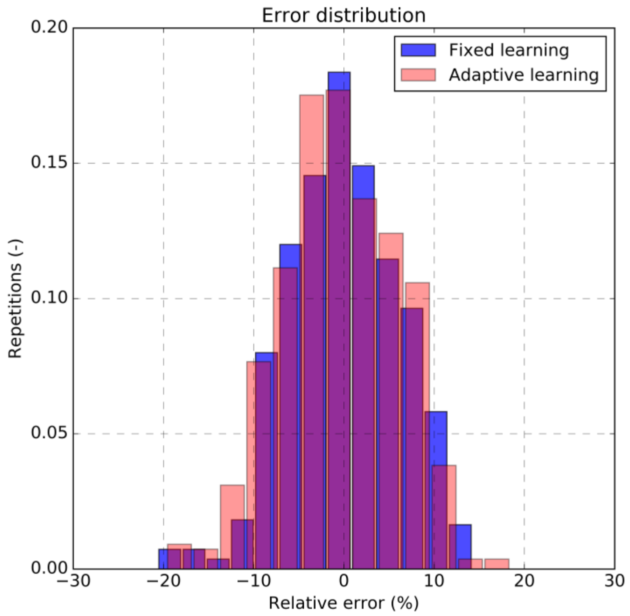

Figure 9. Validation relative error distribution without including the outlier value. In red ADDELE algorithm. In blue fixed learning.