Analytic Learning for Feed-Forward Deep Neural Network with Correlation Projection

Manu Pratap Singh[1] and Pratibha Rashmi [2]
[1] Department of Computer Science, Dr. Bhimarao Ambedkar University, Agra, UP, India
[2] Department of Computer Science, Dr. Bhimarao Ambedkar University, Agra, UP, India
E-mail: pratibha.rashmi@gmail.com

Abstract: A multilayer deep neural network uses mini-batch stochastic gradient descent learning to determine the unknown parameters of the hidden and output layers. The performance of the network for classification is affected by overfitting. The expansion of a feedforward neural network with many hidden layers with optimization of interconnection considers as deep neural network. This modification explored remarkable contributions in the various pattern classification tasks. The mini-batch stochastic gradient descent is also employed as the optimizer for the least mean square error, but the problems with the stochastic gradient descent algorithm degrade the performance of these networks. Therefore, the analytic learning with label encoding for the lower layers is used for various types of networks and gained popularity for the solution of different pattern recognition tasks.

In this present paper, we consider a different type of neural network structure to implement analytic learning. In proposed neural network architecture, we consider the modified Moore-Penrose inverse learning rule to provide dimension-dependent label encoding to the hidden layers and explore correlation projection methods to extract features from the lower layers. The extracted features are further used with a pseudo-inverse rule and a gradient-based least square optimizer to determine the network weights. The proposed technique is applied to the English spoken digit sounds dataset, and simulated results are obtained. The performance of the proposed correlation projection neural network is compared with other optimizers on same dataset, and experimental results indicate better performance compared with several state-of-the-arts.

Keywords: Analytic Learning, Correlation Projection Method, Pattern Recognition, Deep Neural Network, stochastic gradient descent optimizer.

1. Introduction:

Multilayer Feedforward Neural Network may consist with several hidden layers in fully connected manner. The expansion of feed forward neural network with many hidden layers in fully connected manner considered also as Deep Neural Networks. This architecture

of neural network founded to be for remarkable contributions in the various academic and industrial area (Goodfellow et al., 2016).

Multilayer fully connected feedforward neural networks exhibited its ability to approximate complex non-linear mapping directly from the input patterns (Blank & Brown, 1993; Tamura & Tateishi, 1997). It has been also applied for handling the various pattern classification tasks (Nguyen & Widrow, 1990). The performance of neural network for various pattern recognition tasks depends on the learning techniques and it has been investigated by many researchers (Karayiannis & Venetsanopoulos, 2013). The gradient descent based backpropagation algorithm gained the popularity to train the multilayer feedforward neural networks (Aizenberg & Moraga, 2007). It has been widely observed that the backpropagation learning has the problem of slow convergence and the searching for the global minimum point of an error function may be trapped at local minima during gradient descent and if a network has large bounded input disturbances, the global minimum point may not be found (Lu et al., 2006). Several modifications and improvements have been proposed to overcome the problems of backpropagation learning (Becker & Le Cun, 1988; Sarkar, 1995). However, tuning the hyperparameters such as learning rate, initial weights, optimizer selection & number of epochs could be tiresome without proper guidelines. Hence, with such limitation of gradient descent learning, an effective training algorithm for single hidden layer feedforward neural network is proposed (Huynh & Won, 2011). In which, the input weights including hidden layer biases are chosen randomly and the output weights are determined by the pseudo inverse operation of hidden layer output matrix. This proposed approach avoids the problems of gradient descent methods, but it considered more hidden units and takes longer time for classification of input patterns. Further, this problem has been addressed by evolutionary extreme learning machine (Zhu et al., 2005). In this approach, the input weights and hidden layer biases are determined by using the differential evaluation algorithm and the output weights are obtained with analytic learning. However, many variants of analytic learning with different approaches and architectures were presented to improve the classification accuracy(He et al., 2020; Zhuang et al., 2020) but all these approaches were considered with single layer architecture and beside this the large number of hidden units used. Although, Extreme Learning Machine (ELM) attracted the attention of many researcher due to its ability to handle pattern classification problem but ELM can only be used for the single layer feedforward neural network architecture and it could not be found suitable for the Deep Neural Network (Barreto & Barros, 2016; Huang et al., 2004). The

analytic learning approach is further considered for the tuning of input weight & bias besides the output weight matrix. In proposed approach of regularized least square ELM, analytic learning is used to determine both input weights including hidden bias and output weights and this proposed approach applied to datasets with very high input dimensionality(Huynh et al., 2008). Hence, in all these attempts, the analytic learning converted the nonlinear network learning problems into linear segments which can be further tackled by solving matrix equations. Thus, the analytic learning considers higher interpretability owing to the straightforward matrix manipulation during training (Bartlett, 1996). In spite of these abilities of analytic learning, the conventional analytic learning methods worked only for the single layer of shallow networks (Chen et al., 1992; Martínez-Rego et al., 2012). Lots of attempts have been made to apply analytic learning for the deeper or multilayer neural networks. The stochastic gradient descent (SGD) learning and its variants applied for deeper neural network to provide learning for the unknown parameters of the hidden layer along with the parameters of output layer. Stochastic gradient descent approach is of iterative nature and it incurs various problems during the learning for deeper network. The earlier attempts to apply analytic learning constrained with single layer neural network architecture and could not materialized for the deeper network. Lots of attempts have been applied to incorporates analytic leaning for the deeper networks. In this process, an approach of local supervised learning with label encoding method is considered to provide the analytic learning to hidden layer units (Toh, 2018b). The Moore-Penrose method is used to provide analytical learning for the multi-layer neural networks. Further, the multilayer extreme leaning machine is proposed with invertible activation function (Ben-Israel & Greville, 2006). In all these approaches, the problem to deal with deeper network and large dataset created the huddle to improve the performance of the network. Different approaches have also been used to find the possibility for incorporating the analytic learning for multi-layer feed forward neural network models. In this process, the hierarchical structures are included with feed forward neural network(Guo et al., 2001), but it increases the complexity in the analysis of the system. In another approach, the Moore- Penrose leaning method is used to train the neural network by adding extra layers to eliminate the training error (Guo et al., 2001) but it consumes large memory for a large dataset. In next development, the ANnet (Toh, 2018a) is used to train the networks by considering the normalized transpose of the feature matrix of its hidden weights. This method performed well on several dataset but suffer from the memory problem. Similarly, the feed forward design (FF) technique is used to projects the label information by using K-nearest neighbour clustering techniques and principal component analysis (PCA)

(Kuo et al., 2019). This PCA improves the feature extraction process of the feed forward method but increases the complexity and memory consumption. There were various other attempts have been made to incorporate the least-square based analytic learning in shallow networks and deep networks (Pan & Yang, 2009).

In all these approaches, the least square based analytic learning is applied only in the output layer due to lacking of label information to performed learning for the hidden layers. Further, the Restricted Boltzmann Machines are used for the feature extraction followed by the least square estimator in the output layer to obtain analytic learning (Wang et al., 2017). The pre-trained VGG-19 and ResNet-156 models of deep neural networks are used with stacking analytic method (Low et al., 2019). These models performed well on large datasets but the problem of label information for the nodes of hidden layer to perform the learning for hidden layers still unaddressed. A method called PILAE introduced a stacking multilayer autoencoder using the input as label for hidden layers (Guo et al., 2021). This stacking-based technique for designing the network and to incorporate the analytic learning is further used by correlation projection networks (Zhuang et al., 2021). It contains multiple two-layer modules with multiple existing systems and it can be treated as shallow networks. Earlier, the backpropagation based gradient method is used for learning in the stacking based multilayer networks but due to iterative nature, it incurs various problems such as gradient vanishing, non-convergence, and long training time (Wu et al., 2017). Different approaches like PIL (Guo et al., 2001), PILAE (Guo et al., 2021), and ANet (Toh, 2018a) discarded the use of gradient-based backpropagation through learning the network analytically. The interesting observation can be made about the analytic learning. In a broader perceptive, the analytic learning can be considered as the greedy local learning technique (Belilovsky et al., 2020) for the neural networks. The greedy local learning generally considers the local backpropagation which receive error gradient from an auxiliary network (Jaderberg et al., 2017). Most of the stacking-based multilayer networks adopt such type of greedy-like learning strategy but involve a fine-tune process to adjust the weights of hidden layers. It can be seen in deep belief network (Hinton et al., 2006) that it stacks multiple RBMs as hidden layers and tunes the lower layers whenever a new RBM introduces in network. Further, a tensor deep stacking network is considered with each module containing two parallel hidden layers to extract features from different sources (Hutchinson et al., 2012). Thus, the stacking-based multilayer networks usually give competitive prediction performance but on the other hand, it occupies the longer training time due to the need of adjusting the lower layers for every additional

layer. Hence, the problem with BP algorithms degrades the performance of these networks. Therefore, the analytic learning with label encoding for the lower layers used for various types of networks and gained popularity for the solution of different pattern recognition tasks. The analytic learning technique usually requires only one visit of the dataset, leading to significantly faster learning pace than that of the BP learning approach. Hence, different methods have been proposed to facilitate the label information into hidden layers to achieve multilayer analytic learning (Toh, 2018b). The sequential moore - penrose (MP) inverse operation is used in most of the cases to provide the label information for the hidden layers but the appropriate information coding of target output for the hidden layers involves some inadequacy with the successive projection of MP inverse operation alone (Guo et al., 2021).

In this paper, a different type of neural network structures is proposed to implement analytic learning. Thus, proposed multilayer feedforward neural network uses modified MP inverse learning rule to provide dimension dependent label encoding to the hidden layers and also uses the correlation projection technique to extract features from the lower layers. The extracted features are further used with pseudo-inverse rule and gradient based least square optimiser to determine the network weights. The proposed architecture of multilayer network contains several single layer feedforward neural network models with own locally supervised learning rule. The locally supervised learning rule contains correlation projection pseudo inverse term and the gradient based least square optimiser term with label encoding for the hidden layer of each module of the network. The proposed approach has been applied on the datasets of English spoken digit sound samples. The simulation results have been obtained to analyses the performance of proposed architecture with other existing pre-trained modules of deep neural network.

The main contributions of the work include:

-Facilitate the locally supervised learning to encode the input weights for the hidden layer in proposed architecture of multilayer feedforward neural network using label encoding.

-Design of the dimension dependent label encoding technique to provide label information into hidden layers of the several single layer feedforward neural network to encode the input pattern information.

-Design of the correlation projection technique to extract features from the lower layers and to use the moore- penrose (MP) inverse rule with gradient based least square optimiser to determine the network weights.

-Observations from the simulation results for the classification performance and learning efficiency of the proposed network with analytic learning.

Rest of the paper is organised as following: section 2 presents the proposed architecture of multilayer feedforward neural network with label encoding technique. Mathematical formulation and design of correlation projection technique to extract features and MP inverse learning rule to determine the weights are discussed in Section 3. Experimental results & discussion is found in section 4. Conclusion of the work is presented in Section 5 followed by the references.

2. Feed-Forward Neural Networks with Label Encoding

Single Hidden Layer Neural Networks (SHNNs) are used for the construction of proposed multilayer feedforward neural network architecture. The proposed architecture considered the modules of several single hidden layer feedforward neural networks. The output of SHNN works as the input for the next single hidden layer neural network and so on. It continues till the classification layer of the whole multilayer neural networks. Each module of this architecture consists with three layers with two weight vectors. Hence, each module accepts the input and propagates it for the hidden layer. The output of the hidden layer further propagates as input to the second layer i.e., the output layer of the module. The output layer of the first module works as input layer for the second module and similarly the output layer of the $(m-1)^{th}$ module serves as the input layer for the m^{th} module. Each module considers the locally supervised learning rules to update the weight vectors. There are two weight vectors for each module. The inner weight vector exists between input layer & hidden layer and the outer weight vector exists between hidden layer and output layer. The inner weight vector uses the correlation projection encoding to extract the features from input and the moore-penrose (MP) inverse with gradient based least square optimization is used to determine the weights analytically for outer weight vectors. The MP inverse rule with gradient based optimization performs well for each module, if the information about target pattern is available to outer layer of each module. Thus, it becomes necessary to encode the target information for each module. Hence, the Label encoding process is considered for this purpose. It encodes the label information to each module in local view or to hidden layers in global view with appropriate dimension conversion. The proposed correlation projection feedforward neural network architecture to explore the learning analytically can be visualised in Fig. 1.

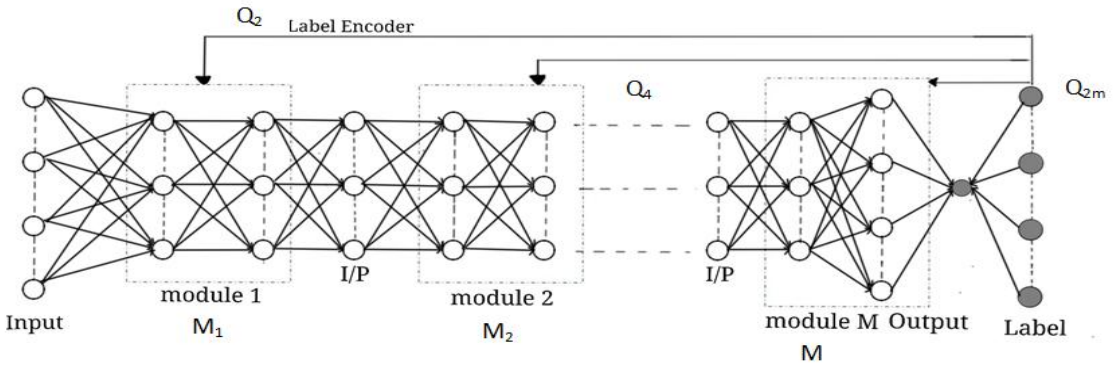

Figure 1: Global view of proposed correlation project feed forward neural network

The architecture of an individual module can be visualized as of Fig 2.

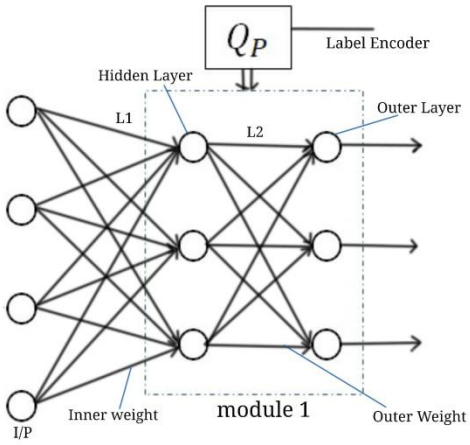

Figure 2: Local view for each module of the CPNetwork

The proposed architecture of correlation projected feedforward neural network includes the label encoding process that provides the target pattern information to each module, which therefore facilitates the locally supervised learning. In this architecture, within each module the first layer (L_1) uses the correlation projection method to extract the features from the input and encoded them in the inner weight vector. The second layer (L_2) maps the extracted features analytically onto the encoded label and uses the MP inverse learning with gradient descent least square optimiser to obtain the outer weight vector. The output of each module serves as the input pattern for its successive module and each module has its local label encoded information. The input to the first module i.e., M_1 considers the pattern information from the presented input stimuli and the last module i.e., M_m provides the final output pattern information corresponding to the presented input at module M_1 . The target pattern information is initially presented for the module M_m. The label encoding process encodes the

label information to all the previous modules i.e., M_1 to M_{m-1} with appropriate dimension conversion. Hence, to exhibit the functioning of the proposed architecture with Label encoding, inner & outer weights and outputs of the modules we consider the layer index $l = 1, 2, 3 \ldots L$, with $2m-1$ and $2m$ for the module index $m = 1, 2, \ldots\ldots M$ end $M = {}^{L}/_{2}$, with the depth L being an even integer. Thus, the first module M_1 considers the layer 1 and 2, module M_2 considers the layer 3 and 4, similarly the module M_m consider the layer $2L-1$ and $2L$. In the same way, the label encoded for the module M_m is Q_{2m} and it considered as the outputs of both the layer for each module as Y_{2m-1} and Y_{2m} whereas, the input to each module is Y_{2m-2}. Similarly, the inner & outer weights for the module are considered as W_{2m-1} and W_{2m} respectively.

Now, we consider a training set of N samples, which consists with input and corresponding target pattern vectors. Let X and T represent the input and target vectors:

$$X = [P_0, P_1, \ldots\ldots\ldots\ldots P_N] \quad \text{where, } P_i = \begin{bmatrix} x_1^i \\ x_2^i \\ \vdots \\ \vdots \\ x_{d_o}^i \end{bmatrix} \quad \forall\, i = 1\ to\ N$$

and,

$$T = [T_0, T_1, \ldots\ldots\ldots T_N] \quad \text{where } T_i = \begin{bmatrix} y_1^i \\ y_2^i \\ \vdots \\ \vdots \\ y_{d_y}^i \end{bmatrix} \quad \forall\, i = 1\ to\ N$$

Hence, $X \in \mathbb{R}^{d_o \times N}$ and $T \in \mathbb{R}^{d_y \times N}$ are respectively the data and label pattern vectors. The label pattern information is available for the output layer of the M^{th} module i.e., the last module in the stack of multilayer neural network architecture. This label information is encoded into the hidden layer to assist supervised learning in each module.

In proposed label encoding process, the size of target class is set according to the dimension of hidden layer. Hence, by adopting the linear transformation the size difference of hidden layer from the label layer can be accommodated. The label or target matrix $T \in \mathbb{R}^{d_y \times N}$ is on the output layer. The target label for the hidden layers indexed by $2m$ can be represented as:

$$\widehat{T}_{2m} = Q_{2m} T;\ \forall\, m = 1\ to\ M-1 \qquad (1)$$

Hence, Q_{2m} and $\hat{T}_{2m}$ are respectively referred to as the label encoder and the encoded label matrix in the $2m^{th}$ layer with $\hat{T}_{2m} \in \mathbb{R}^{d_{2m} \times N}$ and $Q_{2m} \in \mathbb{R}^{t \times d_{2m}}$. Thus, the d_{2m} is representing the number of units in the respective hidden layer and t is showing the dimension of each target pattern vector. Normal distribution has been used to consider the label encoder value (Q) randomly for each hidden layer indexed by $2m$. The last module i.e., M is directly linked to the actual label pattern i.e., $\hat{T}_{2m} = T$.

Therefore, the expected target or label information for the hidden layer indexed by $2m$ module can be constructed with the encrypted form of the true label information through linear transformation.

3. Correlation Projection and Analytic Learning

The label encoding process i.e., $\hat{T}_{2m} = Q_{2m}T$ is invertible such that

$$T = \hat{T}_{2m}{Q_{2m}}^{+} = Q_{2m}{Q_{2m}}^{+}T = T \quad (2)$$

if the Q_{2m} is a full row rank. Thus, the label encoding converts the multilayer network into multiple 2-layer modules. As per our proposed multilayer feed forward neural network architecture, the output of the last layer can be expressed as:

$$Y_L = f_L(f_{L-1}(\ldots . f_2(f_1(W_1X)W_2)\ldots .)W_L \quad (3)$$

For convenience in computation, we consider the linear activation function for all the layers. In that condition, we have

$$Y_L = (W_1X)W_2)\ldots\ldots\ldots\ldots)W_L \quad (4)$$

Here, $Y_L \in \mathbb{R}^{d_y \times N}$ and weight matrices for each layer, $W_1 \in \mathbb{R}^{d_1 \times d_o}; \ldots\ldots\ldots W_l \in \mathbb{R}^{d_l \times d_{l-1}}, \ldots\ldots\ldots . W_L \in \mathbb{R}^{d_y \times d_{L-1}}$. Since the objective of the analytic learning is to train the network to obtain the optimal weight vector $\{W_1 \ldots\ldots\ldots\ldots W_L\}$ for the given input pattern vector X, and the label output pattern vector T, such that the actual output Y_L approaches to the target label T with the minimization of objective function i.e. least square error as:

$$E_{\min\{W_1 \ldots\ldots\ldots\ldots W_l\}} = \|T - Y_l\|^2 \quad (5)$$

Hence, after label encoding process for the hidden layers and partition the network into the modules as shown in Fig. 1, the class label information is available for each module. Now, the process to determine the inner weight and outer weight is initiated. The training process

for the network explores the weight matrices in each module, it includes the Label encoding process for target propagation as per equation 2, the correlation projection method for feature extraction and encodes it in the inner weight matrices and MP inverse with gradient descent optimizer for least square estimates to determine the outer weight matrices. Let us consider the input pattern vector $X_o \in \mathbb{R}^{d_o \times N}$ presents to the first module (M_1) of the network. Hence, the activation of the first module as shown in Fig. 2 can be expressed as:

$$y_1 = f_o(X_o X_o^T P_1) \tag{6}$$

Where, P is non-zero matrix of dimension $d_o \times N$ i.e., $P_1 \in \mathbb{R}^{d_1 \times N}$ and considered as the correlation projector that projects the correlation matrix $X_o X_o^T \in \mathbb{R}^{N \times N}$ onto $X_o X_o^T P_1 \in \mathbb{R}^{d_1 \times N}$ for the hidden layer size d_1. Since, hidden layer size d_1 is considered as the correlation projection layer which is used to determine the inner weight matrices i.e., W_1. The elements of the projector P_1 are also randomly determined through normal distribution. Now, the equation 3 can be rewrite as:

$$Y_1 = f_o(X_o \widehat{W}_1) \text{ or } Y_1 = X_o \widehat{W}_1 \tag{7}$$

where, $\widehat{W}_1 = X_o^T P_1$

Hence, the inner weight matrix for the module M_1 is $\widehat{W}_1 = X_o^T P_1$. The output Y_1 maps on the second layer of the module M_2 which contains the encoded label matrix $\widehat{T}_2$ as:

$$\widehat{T}_2 = TQ_2 \tag{8}$$

This label encoding information and the output value of the first layer is used to determine the outer weight for the second layer. The objective function for the second layer is considered as the least square estimation i.e.,

$$E_1 = {}^1\!/_2 (\widehat{T}_2 - W_2 . Y_1)^2 \tag{9}$$

Therefore, the weight will be determined to minimize the error function as:

$$\text{We have, } \left\| \widehat{T}_2 - W_2 . Y_1 \right\|^2 = 0 \tag{10}$$

$$\text{Hence, } \widehat{T}_2 - W_2 . Y_1 \text{ and } W_2 = \widehat{T}_2 Y_1^+ \tag{11}$$

The equation 11 indicates that, the error should be vanished, and it will reflect the perfect situation for the convergence of learning rule but the input pattern samples do not present in deterministic order and there is specific no rule that how many times a pattern will present.

Therefore, the obtained error is not deterministic and global one. Instead of this, it is of random nature and it is a local unknown error for each presented pattern. Therefore, the gradient descent optimiser is incorporated with the MP inverse rule to obtain the global minima of the error as:

$$\widehat{W}_2 = \hat{T}_2 Y_1^+ + \eta\left(\hat{T}_2 - Y_2\right) \cdot Y_1$$

or, $\widehat{W}_2 = \hat{T}_2 Y_1^+ + \eta\left(\hat{T}_2 - \widehat{W}_2 Y_2\right) \cdot Y_1$

or, $\widehat{W}_2 = \hat{T}_2 Y_1^+ + \eta\delta_2 \cdot Y_1$

and we have, $\widehat{W}_2 = \hat{T}_2 Y_1^+ + \eta\delta_2 \widehat{W}_1 X_0 = \hat{T}_2 Y_1^+ + \eta\delta_2 X_0^T X_0 P_1$ (12)

Similarly, for the second M_2 we have,

$Y_3 = Y_2 \widehat{W}_3$ where $\widehat{W}_3 = Y_2^T P_3$

and $\hat{T}_4 = TQ_4$

The LMS is $E_2 = {}^1/_2 (\hat{T}_4 - \widehat{W}_4 . Y_3)^2$

The outer weight matrices for second module can be expressed as:

$\widehat{W}_4 = \hat{T}_4 Y_3^+ + \eta\left(\hat{T}_4 - \widehat{W}_4 Y_3\right) . Y_3$

or, $\widehat{W}_4 = \hat{T}_4 Y_3^+ + \eta\delta_4 . Y_3$

or, $\widehat{W}_4 = \hat{T}_4 Y_3^+ + \eta\delta_4 . Y_2 \widehat{W}_3$

$= \hat{T}_4 Y_3^+ + \eta\delta_4 \widehat{W}_2 Y_1 \widehat{W}_3$

$= \hat{T}_4 Y_3^+ + \eta\delta_4 \widehat{W}_2 \widehat{W}_3 \widehat{W}_1 X_O$

Finally, we have $\widehat{W}_4 = \hat{T}_4 Y_3^+ + \eta\delta_4 \widehat{W}_2 \widehat{W}_3 X_O^T X_O P_1$ (13)

Similarly, we can determine the inner and outer weights for the outer module. Let us consider the generalize case for representing the expression for inner weight & outer weight matrices. Hence, for the any arbitrary m^{th} module, we have the expressions of activation, correlation projection and weight matrices as:

$$Y_{2m-1} = Y_{2m-2} \widehat{W}_{2m-1} \quad (14)$$

$$\widehat{W}_{2m-1} = Y_{2m-2}^T P_{2m-1} \quad (15)$$

$W_{2m} = Y_{2m-1}^+ T_{2m} + \eta(\hat{T}_{2m} - \widehat{W}_{2m} Y_{2m-1}) Y_{2m-1}$

or, $W_{2m} = Y_{2m-1}^+ T_{2m} + \eta\delta_{2m} (\widehat{W}_{2m-1} \widehat{W}_{2m-2} \ldots . \widehat{W}_1) X_O$

or, $W_{2m} = Y_{2m-1}^{+} T_{2m} + \eta\delta_{2m}(\widehat{W}_{2m-1}\widehat{W}_{2m-2}\ldots.\widehat{W}_{2})X_O^T X_O P_1$ (16)

here, $T_{2m} = TQ_{2m}$

or $T = T_{2m}Q_{2m}^{+}$ (if Q_{2m} has full row rank) (17)

Hence, after learning the weights in the m^{th} module, the output for the module can be expressed as:

$$Y_{2m} = Y_{2m-1}\widehat{W}_{2m} \quad (18)$$

Thus, module m learns its parameters i.e. $\widehat{W}_{2m-2}$ and $\widehat{W}_{2m}$ using the input Y_{2m-1} and the label $\widehat{T}_{2m}$. The input for each module can be treated as the feature extracted from a pre-trained network that consists of all the learned lower layers. The learning of each module as specified in equation 16 considers the information of previous module layer weight matrices and correlation term with presented input information. It suggests that the weight update for each module considers the actual information. Therefore, the extracted features are available for weight update of each module. This approach improves the efficiency of the learning for the entire network. The correlation projected term of the first layer is also available for the weight update of final layer of last module with a combined matrix which contains the information of all the updated weights of previous module. Thus, this technique improves the learning in the network with better convergence and accuracy in classification.

4. Performance Analysis of Proposed Learning

The proposed architecture of multilayer feed forward neural network considered the correlation projection and analytic learning for updating the inner and outer weight matrices of each module. The weight update equations have been described in equation 15 and 16 with label encoding equation 17.

Now, we consider the performance analysis of the proposed neural network model for the least square estimation E_m of the m^{th} module. This performance analysis considers the prediction of output for the modules and for the whole network. Let us consider with M^{th} module i.e., the last module. Thus, the predicted output Y_{2m} can be expressed as:

$Y_{2m} = Y_{2m-1}W_{2m}$ for the M^{th}module

$$= Y_{2m-1}[Y_{2m}^{+}T + \eta(T - Y_{2m})Y_{2m-1}]$$

$$= Y_{2m-1}Y_{2m}^{+}T + \eta(T - Y_{2m})Y_{2m-1}^{2}$$

Let us consider $\eta = 1$,

So we have, $Y_{2m} = Y_{2m-1}Y_{2m}^{+}T + TY_{2m-1}^{2} - Y_{2m}Y_{2m-1}^{2}$

$= T + TY_{2m-1}^{2} - Y_{2m}Y_{2m-1}^{2}$

or, $Y_{2m} + Y_{2m}Y_{2m-1}^{2} = T + TY_{2m-1}^{2} = T(1 + Y_{2m-1}^{2})$

or, $Y_{2m}(1 + Y_{2m-1}^{2}) = T(1 + Y_{2m-1}^{2})$

Hence, $Y_{2m} = T$

Thus, the last module i.e.M^{th} produces the actual output pattern vector Y which is same as the desired class label or target output pattern vector T. The same performance analysis we can perform also for the other modules. Now, we explore the performance analysis for the m^{th}module. Let the output for m^{th} module is δ_{2m}.

Hence, $\delta_{2m} = Y_{2m-1}W_{2m}$

or, $\delta_{2m} = Y_{2m-1}[Y_{2m-1}^{+}T_{2m} + \eta[T_{2m} - \delta_{2m}]]Y_{2m-1}$

$= Y_{2m-1}Y_{2m-1}^{+}T_{2m} + \eta[T_{2m} - \delta_{2m}]Y_{2m-1}^{2}$

$= T_{2m} + [T_{2m} - \delta_{2m}]Y_{2m-1}^{2}$ since, $Y_{2m}Y_{2m-1}^{+} = I$ as Y is a two rank matrix and $\eta = 1$

Now, we have

$\delta_{2m} = T_{2m} + T_{2m}Y_{2m-1}^{2} - \delta_{2m}Y_{2m-1}^{2}$

$\delta_{2m} + \delta_{2m}Y_{2m-1}^{2} = T_{2m} + T_{2m}Y_{2m-1}^{2} = T_{2m}(1 + Y_{2m-1}^{2})$

or, $\delta_{2m}(1 + Y_{2m-1}^{2}) = T_{2m}(1 + Y_{2m-1}^{2})$

Hence, $\delta_{2m} = T_{2m} = TQ_{2m}$

Thus, the output for the m^{th} module is same as the expected target output for the m^{th} module.

Here, we also consider a theorem regarding the prediction of expected output for any arbitrary m^{th} module of the feed forward neural network.

Theorem: if T & T_{2m} are the full row rank then the expected output of the m^{th} module is same the label decoded intermediation of the m^{th} module.

Proof: Let the expected output of the m^{th} module is $\widehat{Y}_{2m}$. Hence, for an M module correlation projection network, the m^{th} $(m < M)$ module gives the following prediction.

$\widehat{Y}_{2m} = X(\widehat{W}_1|W_2 \ldots\ldots)\widehat{W}_{2m}Q_{2m}^{+}$

Or simply; $\widehat{Y}_{2m} = Y_{2m-1}\widehat{W}_{2m}Q_{2m}^{+}$ (19)

Where, Q_{2m}^{+} is the label decoder in the module m which is the MP inverse of the corresponding label encoder.

From equation 19 we have,

$\widehat{Y}_{2m} = Y_{2m-1}\left[Y_{2m}^{+}T_{2m} + \eta\left[T_{2m} - \widehat{Y}_{2m}\right]Y_{2m-1}\right]Q_{2m}^{+}$

$= \left[Y_{2m-1}Y_{2m}^{+}T_{2m} + \eta\left[T_{2m} - \widehat{Y}_{2m}\right]Y_{2m-1}^{2}\right]Q_{2m}^{+}$

$= T_{2m}Q_{2m}^{+} + \eta\left[T_{2m} - \widehat{Y}_{2m}\right]Y_{2m-1}^{2}Q_{2m}^{+}$

Let $\eta = 1$, so that,

$\widehat{Y}_{2m} = T_{2m}Q_{2m}^{+} + T_{2m}Q_{2m}^{+}Y_{2m-1}^{2} - \widehat{Y}_{2m}Y_{2m-1}^{2}Q_{2m}^{+}$

or, $\widehat{Y}_{2m} + \widehat{Y}_{2m}Y_{2m-1}^{2}Q_{2m}^{+} = T_{2m}Q_{2m}^{+} + T_{2m}Q_{2m}^{+}Y_{2m-1}^{2}$

$= T_{2m}Q_{2m}^{+}\left[1 + Y_{2m-1}^{2}\right]$

Here, $\widehat{Y}_{2m}\left[1 + Y_{2m-1}^{2}Q_{2m}^{+}\right] = T_{2m}Q_{2m}^{+}\left[1 + Y_{2m-1}^{2}\right]$

or, $\widehat{Y}_{2m} = \frac{T_{2m}Q_{2m}^{+}\left[1+Y_{2m-1}^{2}\right]}{1+Y_{2m-1}^{2}Q_{2m}^{+}}$

$= \frac{T\left[1+Y_{2m-1}^{2}\right]}{1+Y_{2m-1}^{2}Q_{2m}^{+}}$ since, $T = T_{2m}Q_{2m}^{+}$ from equation 2

$= \frac{T\left[1+Y_{2m-1}^{2}\right]}{1+Y_{2m-1}^{2}TT_{2m}^{+}}$

$= \frac{T\left[1+Y_{2m-1}^{2}\right]}{1+Y_{2m-1}^{2}}$ since, T & T_{2m}^{+} are the full row rank

$= T_{2m}Q_{2m}^{+}$

Hence, $\widehat{Y}_{2m} = T_{2m}Q_{2m}^{+}$

Thus, the theorem is proved and it can be seen that the expected output of the m^{th} module is equal to the label decoded information of the module, i.e., the m^{th} module gives identical prediction. Hence, the performance analysis and theorem for the proposed correlation projection network demonstrates that the prediction gives by the last layer, i.e. M module obtain the exact prediction as of final class label and the prediction of the subnetworks i.e. module 1, module 2….module M-1 obtain the predication of corresponding target output or the label encoded information of the module as long as the label encoders are of full row rank. Apart from that, the expected output of each module obtains the exact label decoded information of the module.

The training & prediction phase of the proposed correlation projection network can be designed as:

Algorithm: CPNet

[This algorithm presents the training for the presented input matrix $X \in \mathbb{R}^{d_O \times N}$, label $T \in \mathbb{R}^{d_y \times N}$ for the network of layer index $l = 1, 2, \ldots\ldots L$ with $2m-1$ and $2m$ for the module index $m = 1, 2, \ldots M$ and $M = L/2$ with the depth L being an even integer]. The label encoders are $Q_2, \ldots\ldots Q_{2m}$ and correlation projector are $P_1, \ldots\ldots P_{2m-1}$. initialize $Y_o = X_o$for $m = 1\ to\ M$:

compute: $\widehat{T}_{2m} = TQ_{2m}$

$\widehat{W}_{2m-1} = Y_{2m-2}^T P_{2m-1}$

$Y_{2m-1} = Y_{2m-2}\widehat{W}_{2m-1}$

$\widehat{W}_{2m} = Y_{2m+1}^+ T_{2m} + \eta\delta_{2m}\left(\widehat{W}_{2m-1}\widehat{W}_{2m-2}\ldots W_2\right)X_o^T Y_o P_1$

Compute the output: $Y_{2m} = Y_{2m-1}\widehat{W}_{2m}$

and compute expected outcome $\widehat{Y}_{2m} = Y_{2m-1}\widehat{W}_{2m}Q_{2m}^+$

return $\widehat{W}_1, \widehat{W}_2, \ldots \widehat{W}_o$

The learning of each module as specified in equation 15 and 16 considers the one gradient term with MP inverse term to compute the multilayer analytic learning with least square optimizer. Therefore, the proposed correlation projection network enables the locally

supervised learning beside flexibility adjust the network structure through the correlation projection. Thus, module m learns its parameters i.e. $\widehat{W}_{2m-1}$ and $\widehat{W}_{2m}$ using the input $\widehat{T}_{2m-2}$ and the label information $\widehat{T}_{2m}$. It can be viewed as the module-wise learning as $Y_{2m} = X(\widehat{W}_1, \widehat{W}_2 .....)\widehat{W}_{2m}$. Thus, the input for each module is treated as the feature expected from the pre-trained network that consists of all the learned lower layers.

5. Result and Discussion

We considered the `multilayer neural network architecture with correlation projection labeling and analytic learning for the spoken English digit dataset Speech Commands v1 (SCV1). The Speech Commands V1 (SCV1) dataset is composed of single spoken English words. It consists of 64,727 one-second .wav audio file of 30 common speech commands. The audio files are arranged into the folders based on the word they contain. So we have used only the folders that contain the digits. We used the total 1,000 samples in which 800 for training and 200 samples for testing the models and total ten (10) classes are used for the classification of input samples. In our proposed architecture six modules correlation projection networks are used. The ReLU function is used as the activation function. In the proposed experiment, we simplify the unknown parameters search by putting the number of units in each hidden layer identical. It means the number of hidden layers and the number of units in each hidden layers are considered same. Further, we use 10-fold cross validation for the training set. The training set consists with mel frequency spectrum image pattern with respective target output patterns. Here, total 10 classes are considered and 10 binary digits are used to represent each class. Thus, our target output pattern is of size 10 x 1. Each input pattern vector of mel frequency spectrum of English spoken digits are considered of size 2024 x 1. Total 1000 sound samples are used. Among the 1000 samples, 800 samples are used for the training and remaining 200 samples are used for the testing. Thus, the training set consists of size 2024 x 800 for input patterns and 10 x 800 for the output patterns. The simulated results have been obtained for training set and for test set. We re-train the model to determine the hidden neuron size that gives the best validation performance. In each module the 200 units are used for the hidden layer and 100 units for the output layer. The optimizer determined the unknown hyperparameters for each module. The training performance for the sound samples data is presented in Table 1. and Fig. 3.

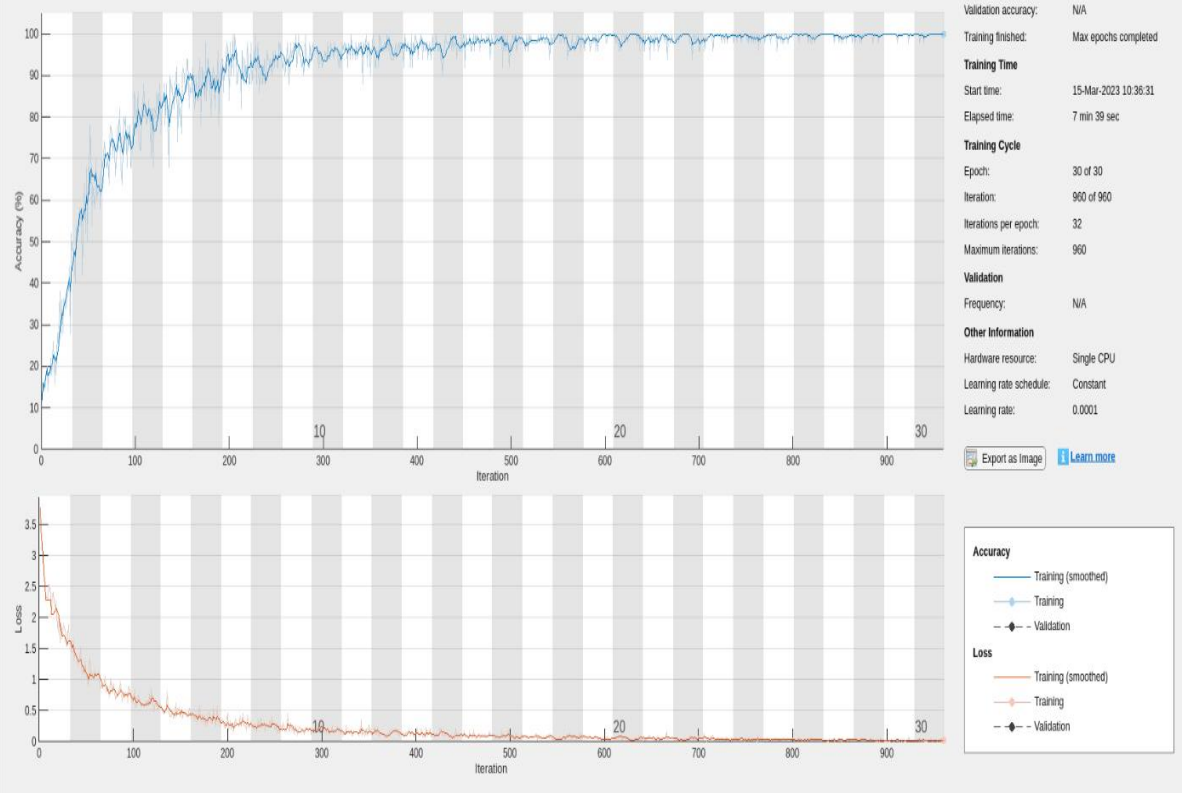

Figure 3: Training Performance of the Sound Samples

Table 1: Training accuracy results after each run on CPNet Architecture

Solution	Accuracy score in %
I Run	94.2
II Run	95.8
III Run	97.4
IV Run	98.56

The result of first run was 94.2% accuracy and after the re-training it reached upto 98.56% accuracy. There is no overfitting found after the re-training. The proposed method took more time in learning but the validation accuracy improved with respect to other existing approach of classification for the spoken digit sound samples. The confusion matrix for test pattern is obtained and represented in the Fig. 4.

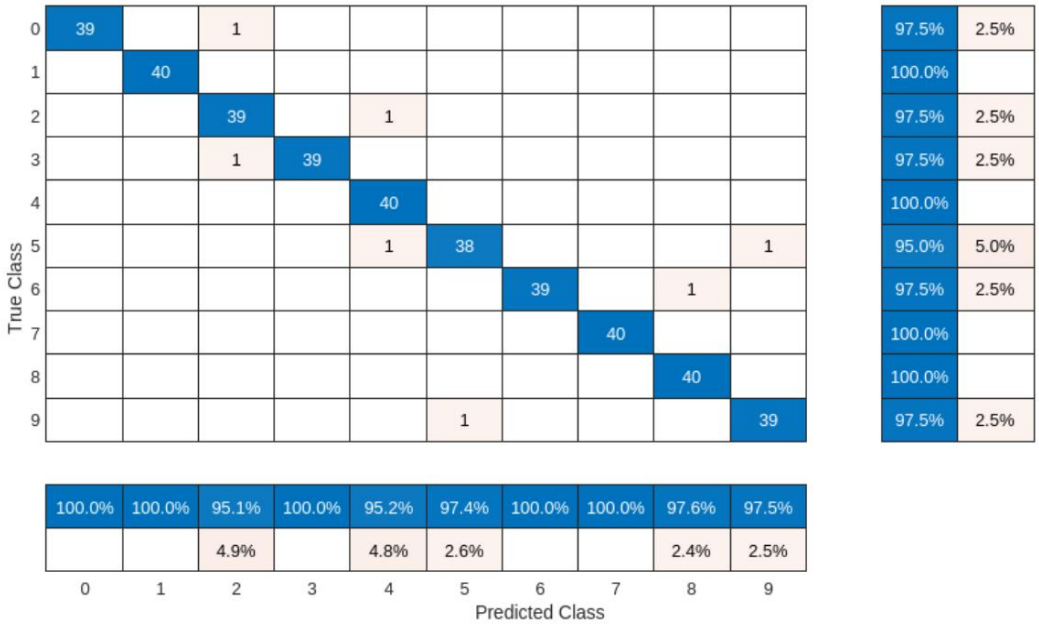

Figure 4: Confusion matrix of CPNet architecture for testing

The error rate against the number of epochs is presented in Fig. 5.

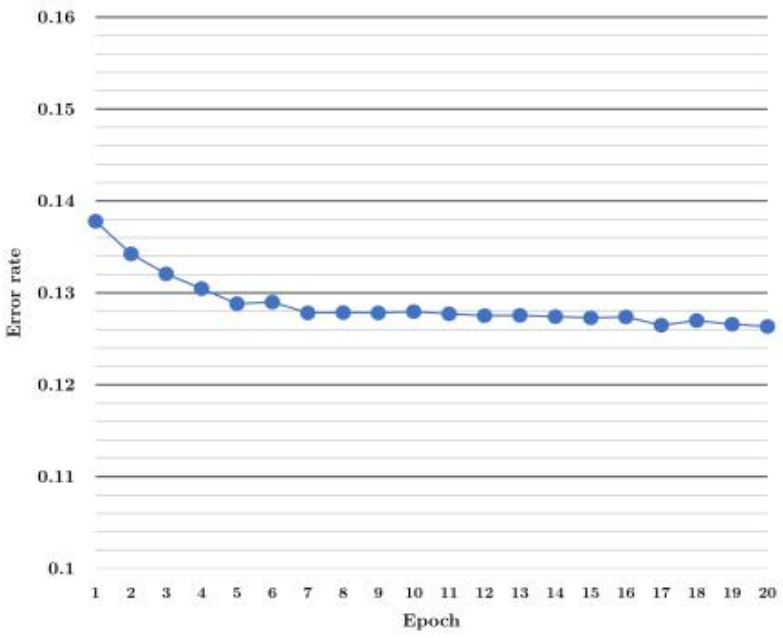

Figure 5: Error rate against number of epochs

The proposed network performance is compared with other neural network approaches. The performance analysis of the proposed CPNet with other methods is presented in Table 2.

Table 2: Performance Comparison with different models

Methods	Number of Units	Test Accuracy (%)
KARnet	1000	91.24
ANnet	5000	94.37
RBM-GI	700-500-500-10000	97.58
PILEA	1000	96.92
VGNet	2024-1056-578-325-165	97.45
Proposed CPNet	1000X7	98.56

The proposed network architecture shows the better performance in comparison to other methods. Therefore, the following issues can be considered from the simulated results.

- The proposed neural network architecture demonstrates competitive results in classification tasks. It outperforms the compared methods on the given English spoken digit dataset.
- The re-training and analytic learning with correlation labeling improves the performance of classification accuracy.
- The extra modules can help the proposed model to achieve better performance.
- It trains the network and performs as a better optimizer with respect to mini-batch stochastic gradient descent learning rule.

6. Conclusion

In this present paper, we introduced a multilayer feed-forward neural network with correlation projection and analytic learning. The proposed model consists with independent single layer feed-forward neural network models with own locally supervised learning rule. The locally supervised learning rule contains correlation projection pseudo inverse term and gradient based least square optimizer term with label encoding for the hidden layer of each module of the network. The target pattern information is only available at the last module of neural network architectures. The other neural networks consider different number of hidden layer's unit from the last module output layer. The target output pattern is labelled and correlated with each individual neural network and modified MP inverse learning rule provide dimension dependent label coding to the hidden layers and used the correlation projection technique to extract features from the previous model layer. Therefore, the entire network architecture exhibits the feature of deep neural network and instead of using mini-batch SGD learning it employs the pseudo inverse method. The simulated results have been obtained with six modules, each of 2-layer modules. To achieve analytic learning, the label is

first encoded into the hidden layer through the label encoding process. Subsequently, in each module, the first layer extracts features using the correlation projection process while the second layer maps the features onto the encoded label by solving least square problem. The simulated results show that the proposed network performs better than several competing state-of-the-art methods. The simulated results indicated that the proposed network improved the performance during the re-train process. The experiment has been concluded on the English spoken digit dataset. The following observations were noticed.

- The proposed network outperform for classification accuracy in 4^{th} re-run process in comparison of other models.
- It considered less time in training with respect to mini batch SGD learning method.
- It reduces the possibility of overfitting and provides labelled target information to each module of 2-layer feed forward neural network.

Thus, the analytic based learning is found to be more effective than the other learning approaches of feed forward deep neural networks but it contain the constraint of indicate number of units in hidden layers with respect to previous module output layer units.

Acknowledgment

The authors thankfully acknowledge the financial support of the Uttar Pradesh government, Lucknow, India in the form of a major research project: 89/2022/1585/seventy-4-2022/001-4-32-2022.

Funding Information

This research is funded by the Uttar Pradesh government, Lucknow, India in the form of a major research project: 89/2022/1585/seventy-4-2022/001-4-32-2022.

Author's Contributions

Manu Pratap Singh: Conceptualization, methodology, written original drafted, supervision, mathematically modeled.

Pratibha Rashmi: Data curation, implementation, visualization, results and validation.

Ethics

This article is original and contains unpublished material. The corresponding author confirms that all of the other authors have read and approved the manuscript and that no ethical issues are involved.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this study.

Availability of Data and Material

Data collection of voice samples of environmental sounds has been considered from the existing Speech Commands V1 (SCV1) dataset.

Conflicts of Interest Statement

The authors whose names are listed immediately below certify that they have no affiliations with or involvement in any organization or entity with any financial interest (such as honoraria; educational grants; participation in speakers' bureaus; membership, employment, consultancies, stock ownership, or other equity interest; and expert testimony or patent-licensing arrangements), or non-financial interest (such as personal or professional relationships, affiliations, knowledge or beliefs) in the subject matter or materials discussed in this manuscript.

References:

Aizenberg, I., & Moraga, C. (2007). Multilayer feedforward neural network based on multi-valued neurons (MLMVN) and a backpropagation learning algorithm. *Soft Computing*, *11*, 169–183.

Barreto, G. A., & Barros, A. L. B. (2016). A robust extreme learning machine for pattern classification with outliers. *Neurocomputing*, *176*, 3–13.

Bartlett, P. (1996). For valid generalization the size of the weights is more important than the size of the network. *Advances in Neural Information Processing Systems*, *9*. https://proceedings.neurips.cc/paper/1996/hash/fb2fcd534b0ff3bbed73cc51df620323-Abstract.html

Becker, S., & Le Cun, Y. (1988). Improving the convergence of back-propagation learning with second order methods. *Proceedings of the 1988 Connectionist Models Summer School*, 29–37. http://yann.lecun.com/exdb/publis/psgz/becker-lecun-89.ps.gz

Belilovsky, E., Eickenberg, M., & Oyallon, E. (2020). Decoupled greedy learning of cnns. *International Conference on Machine Learning*, 736–745. http://proceedings.mlr.press/v119/belilovsky20a.html

Ben-Israel, A., & Greville, T. N. (2006). *Generalized inverses: Theory and applications*. Springer Science & Business Media. https://books.google.com/books?hl=en&lr=&id=abEPBwAAQBAJ&oi=fnd&pg=PR5&dq=Ben-Israel+A,+Greville+TNE+(2003)+Generalized+inverses:+theory+and+applications,+2nd+edn.+Springer,+New+York&ots=gGkDbfzAE5&sig=9WfeeN_aoujMknmOTSs0RHttTes

Blank, T. B., & Brown, S. D. (1993). Nonlinear multivariate mapping of chemical data using feed-forward neural networks. *Analytical Chemistry*, *65*(21), 3081–3089. https://doi.org/10.1021/ac00069a023

Chen, S., Grant, P. M., & Cowan, C. F. N. (1992). Orthogonal least-squares algorithm for training multioutput radial basis function networks. *IEE Proceedings F Radar and Signal Processing*, *139*(6), 378. https://doi.org/10.1049/ip-f-2.1992.0054

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT press. https://books.google.com/books?hl=en&lr=&id=omivDQAAQBAJ&oi=fnd&pg=PR5&dq=goodfellow+deep+learning+mit&ots=MOO0bolIQT&sig=0K1Ifa7k4z6ZqBnDSr1Keyv_THg

Guo, P., Lyu, M. R., & Mastorakis, N. E. (2001). Pseudoinverse learning algorithm for feedforward neural networks. *Advances in Neural Networks and Applications*, *1*(321–326).

Guo, P., Wang, K., & Zhou, X. L. (2021). *PILAE: A Non-gradient Descent Learning Scheme for Deep Feedforward Neural Networks* (arXiv:1811.01545). arXiv. http://arxiv.org/abs/1811.01545

He, K., Peng, Y., Liu, S., & Li, J. (2020). Regularized Negative Label Relaxation Least Squares Regression for Face Recognition. *Neural Processing Letters*, *51*(3), 2629–2647. https://doi.org/10.1007/s11063-020-10219-6

Hinton, G. E., Osindero, S., & Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. *Neural Computation*, *18*(7), 1527–1554.

Huang, G.-B., Zhu, Q.-Y., & Siew, C.-K. (2004). Extreme learning machine: A new learning scheme of feedforward neural networks. *2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No. 04CH37541)*, *2*, 985–990. https://ieeexplore.ieee.org/abstract/document/1380068/

Hutchinson, B., Deng, L., & Yu, D. (2012). Tensor deep stacking networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(8), 1944–1957.

Huynh, H. T., & Won, Y. (2011). Regularized online sequential learning algorithm for single-hidden layer feedforward neural networks. *Pattern Recognition Letters*, *32*(14), 1930–1935.

Huynh, H. T., Won, Y., & Kim, J.-J. (2008). AN IMPROVEMENT OF EXTREME LEARNING MACHINE FOR COMPACT SINGLE-HIDDEN-LAYER FEEDFORWARD NEURAL NETWORKS. *International Journal of Neural Systems*, *18*(05), 433–441. https://doi.org/10.1142/S0129065708001695

Jaderberg, M., Czarnecki, W. M., Osindero, S., Vinyals, O., Graves, A., Silver, D., & Kavukcuoglu, K. (2017). Decoupled neural interfaces using synthetic gradients. *International Conference on Machine Learning*, 1627–1635. http://proceedings.mlr.press/v70/jaderberg17a.html

Karayiannis, N., & Venetsanopoulos, A. N. (2013). *Artificial neural networks: Learning algorithms, performance evaluation, and applications* (Vol. 209). Springer Science & Business Media. https://books.google.com/books?hl=en&lr=&id=K4XdBwAAQBAJ&oi=fnd&pg=PP11&dq=Karayiannis+and+A.+N.+Venetsanopoulos,+%E2%80%9CArtificial+neural+networks:+Learning+algorithms,+performance+evaluation,+and+applications,+Kluver+Academic,+Boston,+MA,+(1993).&ots=6Qqyfk9lVs&sig=3nJsmPNRwzACMwsz4VW2EBp5gVc

Kuo, C.-C. J., Zhang, M., Li, S., Duan, J., & Chen, Y. (2019). Interpretable convolutional neural networks via feedforward design. *Journal of Visual Communication and Image Representation*, *60*, 346–359.

Low, C.-Y., Park, J., & Teoh, A. B.-J. (2019). Stacking-based deep neural network: Deep analytic network for pattern classification. *IEEE Transactions on Cybernetics*, *50*(12), 5021–5034.

Lu, J., Yuan, X., & Yahagi, T. (2006). A method of face recognition based on fuzzy clustering and parallel neural networks. *Signal Processing*, *86*(8), 2026–2039.

Martínez-Rego, D., Fontenla-Romero, O., & Alonso-Betanzos, A. (2012). Nonlinear single layer neural network training algorithm for incremental, nonstationary and distributed learning scenarios. *Pattern Recognition*, *45*(12), 4536–4546.

Nguyen, D., & Widrow, B. (1990). Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights. *1990 IJCNN International Joint Conference on Neural Networks*, 21–26. https://ieeexplore.ieee.org/abstract/document/5726777/

Pan, S. J., & Yang, Q. (2009). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, *22*(10), 1345–1359.

Sarkar, D. (1995). Methods to speed up error back-propagation learning algorithm. *ACM Computing Surveys*, *27*(4), 519–544. https://doi.org/10.1145/234782.234785

Tamura, S., & Tateishi, M. (1997). Capabilities of a four-layered feedforward neural network: Four layers versus three. *IEEE Transactions on Neural Networks*, *8*(2), 251–255.

Toh, K.-A. (2018a). Kernel and Range Approach to Analytic Network Learning: *International Journal of Networked and Distributed Computing*, *7*(1), 20. https://doi.org/10.2991/ijndc.2018.7.1.3

Toh, K.-A. (2018b). Learning from the kernel and the range space. *2018 IEEE/ACIS 17th International Conference on Computer and Information Science (ICIS)*, 1–6. https://ieeexplore.ieee.org/abstract/document/8527565/

Wang, X.-Z., Zhang, T., & Wang, R. (2017). Noniterative deep learning: Incorporating restricted Boltzmann machine into multilayer random weight neural networks. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *49*(7), 1299–1308.

Wu, J., Qiu, S., Zeng, R., Kong, Y., Senhadji, L., & Shu, H. (2017). Multilinear principal component analysis network for tensor object classification. *IEEE Access*, *5*, 3322–3331.

Zhu, Q.-Y., Qin, A. K., Suganthan, P. N., & Huang, G.-B. (2005). Evolutionary extreme learning machine. *Pattern Recognition*, *38*(10), 1759–1763.

Zhuang, H., Lin, Z., & Toh, K.-A. (2020). Training a multilayer network with low-memory kernel-and-range projection. *Journal of the Franklin Institute*, *357*(1), 522–550.

Zhuang, H., Lin, Z., & Toh, K.-A. (2021). Correlation Projection for Analytic Learning of a Classification Network. *Neural Processing Letters*, *53*(6), 3893–3914. https://doi.org/10.1007/s11063-021-10570-2