#6341:supplementary materials

This supplementary materials give the detailed proof [Serre, 2010] of Equations (8)∼(12) in the paper, and we also present the detailed description and increment algorithms of four proposed novel network variants. In addtion, external increment experiments have been performed and results are presented in this materials.

1 Proof

Given a matrix $\mathbf{B} \in M_{p\times q}(\mathbb{C})$ and a vector $a \in \mathbb{C}^P$. Then we can get $A = (\mathbf{B}, a) \in M_{p\times(q+1)}(\mathbb{C})$[1]

We difine $d = \mathbf{B}^+ a, \quad c = a - \mathbf{B}d$ and

$$b = \begin{cases} c^+ & if \quad c \neq 0 \\ \\ (1+|d|^2)^{-1} d^* \mathbf{B}^+ & if \quad c = 0 \end{cases}$$

then

$$A^+ = \begin{pmatrix} \mathbf{B}^+ - db \\ b \end{pmatrix} \tag{1}$$

1.1 Lemma

Four lemmas [Serre, 2010] used for proof are summarized as follows:

Lemma 1: If $\mathbf{A}^+$ is the pseudo-inverse of matrix $\mathbf{A}$, they must satisfy:

$$\mathbf{A}\mathbf{A}^+\mathbf{A} = \mathbf{A} \tag{2}$$

$$\mathbf{A}^+\mathbf{A}\mathbf{A}^+ = \mathbf{A}^+ \tag{3}$$

$$\mathbf{A}\mathbf{A}^+ \in \mathbf{H} \tag{4}$$

$$\mathbf{A}^+\mathbf{A} \in \mathbf{H} \tag{5}$$

where $\mathbf{H}$ means Hermitian Matrix set.

Lemma 2: Given $\mathbf{A} \in M_{p\times q}(\mathbb{C})$, if $\mathbf{A}$ has pseudo-inverse matrix $\mathbf{A}^+$, we have $\mathbf{A}^+ = \|\mathbf{A}\|_2^{-2}\mathbf{A}^*$.[2]

Lemma 3 [Ben-Israel and Greville, 2003]: Given $\mathbf{B} \in M_{p\times q}(\mathbb{C})$ and $\mathbf{B} \neq 0$, $\mathbf{B}$ has pseudo-inverse matrix $\mathbf{B}^+$. Then let $\mathbf{B} = \mathbf{X}\mathbf{Y}$, where $\mathbf{X}$ and $\mathbf{Y}$ are respectively column and row full rank matrices, then we have

$$\mathbf{B}^+ = \mathbf{Y}^*(\mathbf{Y}\mathbf{Y}^*)^{-1}(\mathbf{X}^*\mathbf{X})^{-1}\mathbf{X}^* \tag{6}$$

The proof of ***Lemma 3*** is presented as follows: we verify that the equation(6) is a pseudo-inverse of $\mathbf{B}$ based on ***Lemma 1***. According to the properties of Hermitian Matrix, if $\mathbf{B} \in \mathbf{H}$, we have $\mathbf{B} = \mathbf{B}^*$.

$$\begin{aligned} \mathbf{B}\mathbf{B}^+\mathbf{B} &= (\mathbf{Y}\mathbf{X})\mathbf{X}^*(\mathbf{X}\mathbf{X}^*)^{-1}(\mathbf{Y}^*\mathbf{Y})^{-1}\mathbf{Y}^*(\mathbf{Y}\mathbf{X}) \\ &= \mathbf{Y}\mathbf{X} \\ &= \mathbf{B} \end{aligned}$$

$$\begin{aligned} &\mathbf{B}^+\mathbf{B}\mathbf{B}^+ \\ &= \mathbf{X}^*(\mathbf{X}\mathbf{X}^*)^{-1}(\mathbf{Y}^*\mathbf{Y})^{-1}\mathbf{Y}^*(\mathbf{Y}\mathbf{X})\mathbf{X}^*(\mathbf{X}\mathbf{X}^*)^{-1}(\mathbf{Y}^*\mathbf{Y})^{-1}\mathbf{Y}^* \\ &= \mathbf{X}^*(\mathbf{X}\mathbf{X}^*)^{-1}(\mathbf{Y}^*\mathbf{Y})^{-1}\mathbf{Y}^* \\ &= \mathbf{B}^+ \end{aligned}$$

$$\begin{aligned} (\mathbf{B}\mathbf{B}^+)^* &= \mathbf{Y}(\mathbf{Y}^*\mathbf{Y})^{-1}(\mathbf{X}\mathbf{X}^*)^{-1}\mathbf{X}\mathbf{X}^*\mathbf{Y}^* \\ &= \mathbf{Y}(\mathbf{Y}^*\mathbf{Y})^{-1}\mathbf{Y}^* \\ &= \mathbf{Y}(\mathbf{X}\mathbf{X}^*)(\mathbf{X}\mathbf{X}^*)^{-1}(\mathbf{Y}^*\mathbf{Y})^{-1}\mathbf{Y}^* \\ &= \mathbf{B}\mathbf{B}^+ \end{aligned}$$

$$\begin{aligned} (\mathbf{B}^+\mathbf{B})^* &= \mathbf{X}^*\mathbf{Y}^*\mathbf{Y}(\mathbf{Y}^*\mathbf{Y})^{-1}(\mathbf{X}\mathbf{X}^*)^{-1}\mathbf{X} \\ &= \mathbf{X}^*(\mathbf{X}\mathbf{X}^*)^{-1}\mathbf{X} \\ &= \mathbf{X}^*(\mathbf{X}\mathbf{X}^*)^{-1}(\mathbf{Y}^*\mathbf{Y})^{-1}(\mathbf{Y}^*\mathbf{Y})\mathbf{X} \\ &= \mathbf{B}^+\mathbf{B} \end{aligned}$$

Lemma 4: Given $\mathbf{B} \in M_{p*q}(\mathbb{C})$ and $\mathbf{B} \neq 0$, $\mathbf{B}$ has pseudo-inverse matrix $\mathbf{B}^+$, then we have

$$(\mathbf{B}^+)^*\mathbf{B}^+\mathbf{B} = (\mathbf{B}^+)^* \tag{7}$$

The brief proof of ***Lemma 4*** is presented as following: According to Lemma 3, we get:

$$\begin{aligned} (\mathbf{B}^+)^*\mathbf{B}^+\mathbf{B} = \mathbf{X}[(\mathbf{X}^*\mathbf{X})^{-1}]^*[(\mathbf{Y}\mathbf{Y}^*)^{-1}]^*\mathbf{Y} \\ \mathbf{Y}^*(\mathbf{Y}\mathbf{Y}^*)^{-1}(\mathbf{X}^*\mathbf{X})^{-1}\mathbf{X}^*\mathbf{X}\mathbf{Y} \end{aligned}$$

where:

$$\begin{aligned} &(\mathbf{B}^+)^* = \mathbf{X}[(\mathbf{X}^*\mathbf{X})^{-1}]^*[(\mathbf{Y}\mathbf{Y}^*)^{-1}]^*\mathbf{Y} \\ &\mathbf{B}^+ = \mathbf{Y}^*(\mathbf{Y}\mathbf{Y}^*)^{-1}(\mathbf{X}^*\mathbf{X})^{-1}\mathbf{X}^* \\ &\mathbf{B} = \mathbf{X}\mathbf{Y} \end{aligned}$$

[1] $\mathbb{C}$ means the complex set, so B is a complex matrix and the size is $p \times q$. a is a complex vector.

[2] $\mathbf{A}^* = \bar{\mathbf{A}}^T$

Surprisingly, we find: $\mathbf{YY}^*(\mathbf{YY}^*)^{-1}(\mathbf{X}^*\mathbf{X})^{-1}\mathbf{X}^*\mathbf{X} = \mathbf{I}_n$

then we get: $\mathbf{X}[(\mathbf{X}^*\mathbf{X})^{-1}]^*[(\mathbf{YY}^*)^{-1}]^*\mathbf{Y} = (\mathbf{B}^+)^*$

$$(\mathbf{B}^+)^*\mathbf{B}^+\mathbf{B} = (\mathbf{B}^+)^*$$

1.2 Demonstration

In order to prove Equation (1) is the pseudo-inverse matrix of $\mathbf{A}$, it should be ensured that they must satisfy the Equations (2), (3), (4) and (5).

1.2.1 When c=0

First, we have

$$\begin{aligned}\mathbf{AA}^+ &= (\mathbf{B}, a)\begin{bmatrix}\mathbf{B}^+ - db \\ b\end{bmatrix} \\ &= \mathbf{B}(\mathbf{B}^+ - db) + ab \\ &= \mathbf{BB}^+\end{aligned}$$

so we get: $\mathbf{AA}^+ = \mathbf{BB}^+$

Proof of the Equation(2):

$$\begin{aligned}\mathbf{AA}^+\mathbf{A} &= (\mathbf{AA}^+)\mathbf{A} \\ &= \mathbf{BB}^+(\mathbf{B}, a) \\ &= (\mathbf{B}, \mathbf{BB}^+ a) \\ &= (\mathbf{B}, \mathbf{B}d) \\ &= (\mathbf{B}, a) \\ &= \mathbf{A}\end{aligned}$$

Proof of the Equation(3):

$$\mathbf{A}^+\mathbf{AA}^+ = \begin{bmatrix}\mathbf{B}^+\mathbf{BB}^+ - db\mathbf{BB}^+ \\ b\mathbf{BB}^+\end{bmatrix}$$

$b\mathbf{BB}^+ = (1+|d|^2)^{-1}d^*\mathbf{B}^+\mathbf{BB}^+ = (1+|d|^2)^{-1}d^*\mathbf{B}^+ = b$, so we have $\mathbf{A}^+\mathbf{AA}^+ = \mathbf{A}^+$

Proof of the Equation(4):

because $\mathbf{AA}^+ = \mathbf{BB}^+$, we know $\mathbf{AA}^+ \in H$

Proof of the Equation(5):

$\mathbf{A}^+\mathbf{A} = \begin{bmatrix}\mathbf{B}^+ - d(1+|d|^2)^{-1}d^*\mathbf{B}^+ \\ (1+|d|^2)^{-1}d^*\mathbf{B}^+\end{bmatrix}(\mathbf{B}, a) =$

$$\begin{bmatrix}\mathbf{B}^+\mathbf{B} - (1+|d|^2)^{-1}dd^*\mathbf{B}^+\mathbf{B} & \mathbf{B}^+a - d(1+|d|^2)^{-1}d^*d \\ (1+|d|^2)^{-1}d^*\mathbf{B}^+\mathbf{B} & (1+|d|^2)^{-1}d^*d\end{bmatrix}$$

According to ***Lemma 4***, we can derive:

$$\begin{aligned}d^*\mathbf{B}^+\mathbf{B} &= (\mathbf{B}^+a)^*\mathbf{B}^+\mathbf{B} \\ &= a^*(\mathbf{B}^+)^*\mathbf{B}^+\mathbf{B} \\ &= a^*(\mathbf{B}^+)^* \\ &= d^*\end{aligned}$$

so we have

$\mathbf{A}^+\mathbf{A}$

$$\begin{aligned}&= \begin{bmatrix}\mathbf{B}^+\mathbf{B} - (1+|d|^2)^{-1}dd^* & \mathbf{B}^+a - d(1+|d|^2)^{-1}d^*d \\ (1+|d|^2)^{-1}d^* & (1+|d|^2)^{-1}d^*d\end{bmatrix} \\ &= \begin{bmatrix}\mathbf{B}^+\mathbf{B} - (1+|d|^2)^{-1}dd^* & d - d(1+|d|^2)^{-1}|d|^2 \\ (1+|d|^2)^{-1}d^* & (1+|d|^2)^{-1}d^*d\end{bmatrix} \\ &= \begin{bmatrix}\mathbf{B}^+\mathbf{B} - (1+|d|^2)^{-1}|d|^2 & (1+|d|^2)^{-1}d \\ (1+|d|^2)^{-1}d^* & (1+|d|^2)^{-1}|d|^2\end{bmatrix}\end{aligned}$$

$(1+|d|^2)^{-1}|d|^2$ is not a matrix but a value. Then we know $\mathbf{A}^+\mathbf{A} \in H$

1.2.2 When c≠0

In order to prove this part smoothly, we should first deduce $\mathbf{A}^+\mathbf{A}$. Therefore, We first prove Equation (5).

Proof of the Equation(5):

We introduce ***lemma 2*** here, so we have $c^+ = ||c||_2^{-2}c^*$ and $c^+\mathbf{B} = ||c||_2^{-2}c^*\mathbf{B} = ||c||_2^{-2}a^*(I - \mathbf{BB}^+)\mathbf{B} = 0$

Notice:

$$\begin{aligned}(\mathbf{I} - \mathbf{BB}^+)^2 &= (\mathbf{I} - \mathbf{BB}^+)(\mathbf{I} - \mathbf{BB}^+) \\ &= \mathbf{I} - 2\mathbf{BB}^+ + \mathbf{BB}^+\mathbf{BB}^+ \\ &= \mathbf{I} - \mathbf{BB}^+\end{aligned}$$

Then $c^+a = ||c||_2^{-2}a^*(\mathbf{I} - \mathbf{BB}^+)a = ||c||_2^{-2}a^*(\mathbf{I} - \mathbf{BB}^+)(\mathbf{I} - \mathbf{BB}^+)a = ||c||_2^{-2}c^*c = 1$

so we have

$$\begin{aligned}\mathbf{A}^+\mathbf{A} &= \begin{bmatrix}\mathbf{B}^+\mathbf{B} & (\mathbf{B}^+ - dc^+)a \\ 0 & c^+a\end{bmatrix} \\ &= \begin{bmatrix}\mathbf{B}^+\mathbf{B} & 0 \\ 0 & 1\end{bmatrix}\end{aligned}$$

then we know $\mathbf{A}^+\mathbf{A} \in H$

Proof of the Equation(2):

$$\begin{aligned}\mathbf{AA}^+\mathbf{A} &= \mathbf{A}\begin{bmatrix}\mathbf{B}^+\mathbf{B} & 0 \\ 0 & 1\end{bmatrix} \\ &= (\mathbf{BB}^+\mathbf{B}, a) \\ &= (\mathbf{B}, a) \\ &= \mathbf{A}\end{aligned}$$

Proof of the Equation(3):

where:

$$\begin{aligned}\mathbf{A}^+\mathbf{AA}^+ &= \begin{bmatrix}\mathbf{B}^+\mathbf{B} & 0 \\ 0 & 1\end{bmatrix}\mathbf{A}^+ \\ &= \begin{pmatrix}\mathbf{B}^+\mathbf{BB}^+ - \mathbf{B}^+\mathbf{B}dc^+ \\ c^+\end{pmatrix} \\ &= \mathbf{A}^+\end{aligned}$$

Proof of the Equation(4):

Notice:

$$\mathbf{AA}^+ = (\mathbf{B}, a)\begin{pmatrix} \mathbf{B}^+ - dc^+ \\ c^+ \end{pmatrix}$$
$$= \mathbf{BB}^+ - \mathbf{B}dc^+ + ac^+$$
$$= \mathbf{BB}^+ + cc^+$$

then we have $\mathbf{AA}^+ \in H$

2 Incremental Learning

Incremental learning is an important feature of BLS. In this section, the incremental algorithm of BLS network deformation will be introduced. There are four network variants of BLS have been proposed, including CFBLS-pyramid, CEBLS-dense, CFBLS-dense and CEBLS-dense. It should be noted that the incremental part can only be placed after the original matrix. $A = (\mathbf{B}, a)$ in Equation(1) show that incremental part a must be placed after original matrix $\mathbf{B}$.

There are three types of incremental algorithms, the increment of additional enhancement nodes, the increment of additional feature nodes and the increment of additional input data. [Chen and Liu, 2018] has introduced the basic implementation of the incremental algorithm. In this paper, we will introduce some typical incremental algorithms of the four network variants.

2.1 Incremental Learning for CFBLS-pyramid: Increment of Additional Feature Nodes

CFBLS-pyramid, as the network deformation of BLS, inherits the incremental learning perfectly. Because the incremental learning on the enhancement nodes and input data is similar to the BLS[3], we will focus on incremental learning of feature nodes.

Assume that the initial structure consists of N' feature node which is divided into N layers and M enhancement nodes. Considering that the $(N'+1)$th feature nodes are added. There are two possible scenarios, if the new feature node is the first feature node of the layer, as shown in Figure 1(a), it can be denoted as:

$$\mathbf{Z}_{N'+1} = \phi(\mathbf{X}\mathbf{W}_{\mathbf{e}_{N'+1}} + \beta_{e_{N'+1}}) \tag{8}$$

In other cases, as shown in Figure 1(b), it can be denoted as:

$$\mathbf{Z}_{N'+1} = \phi(\mathbf{Z}_{N'}\mathbf{W}_{\mathbf{e}_{N'+1}} + \beta_{e_{N'+1}}) \tag{9}$$

where $\mathbf{Z}_{N'}$ is the last feature node in initial structure.

When the number of feature nodes increases, the number of enhancement nodes also increases by M'[4]. The additional enhancement nodes are randomly generated as follows:

$$\mathbf{H}^{M'} = [\zeta(\mathbf{Z}_{N'+1}\mathbf{W}_{h_{x1}} + \beta_{h_{x1}}), \ldots, \zeta(\mathbf{Z}_{N'+1}\mathbf{W}_{h_{xM'}} + \beta_{h_{xM'}})] \tag{10}$$

[3]Compared with BLS, C only makes changes to the acquisition mode of feature nodes, so its input data and increment of enhancement nodes do not change much. Same thing with CFBLS-dense and CEBLS-dense

[4]M' is a predetermined constant

where $\mathbf{W_h}, \beta_e$ are randomly generated. Then, we can get $\mathbf{D}^M_{N'+1} = [\mathbf{D}^M_{N'}|\mathbf{Z}_{N'+1}|\mathbf{H}^{M'}]$, the upgraded pseudoinverse matrix should be achieved as follows:

$$(\mathbf{D}^M_{N'+1})^+ = \begin{bmatrix} (\mathbf{D}^M_{N'})^+ - \mathbf{QB}^T \\ \mathbf{B}^T \end{bmatrix} \tag{11}$$

where $\mathbf{Q} = (\mathbf{D}^M_{N'})^+[\mathbf{Z}_{N'+1}|\mathbf{H}^{M'}]$

$$\mathbf{B}^T = \begin{cases} \mathbf{C}^+ & if \quad \mathbf{C} \neq 0 \\ \\ (1 + \mathbf{Q}^T\mathbf{Q})^{-1}\mathbf{Q}^T(\mathbf{D}^M_{N'})^+ & if \quad \mathbf{C} = 0 \end{cases} \tag{12}$$

and $\mathbf{C} = [\mathbf{Z}_{N'+1}|\mathbf{H}^{M'}] - \mathbf{D}^M_{N'}\mathbf{Q}$

So, the new weights are:

$$\mathbf{W}^M_{N'+1} = \begin{bmatrix} \mathbf{W}^M_{N'} - \mathbf{QB}^T\mathbf{Y} \\ \mathbf{B}^T\mathbf{Y} \end{bmatrix} \tag{13}$$

The incremental algorithm of the increment feature node is shown in Algorithm 1.

2.2 Incremental Learning for CFBLS-dense: Increment of Additional Feature Nodes

CFBLS-dense, is another kind of variant network about feature nodes. The incremental algorithm of feature nodes of CFBLS-dense is similar to the incremental algorithm of feature nodes of CFBLS-pyramid. Next, we will describe this algorithm of CFBLS-dense.

Assume that the initial structure consists of N feature node and M broad enhancement nodes. Considering that the $(N+1)$th feature nodes are added, unlike CFBLS-pyramid, there is just one situation for CFBLS-dense (see Figure 2), it can be denoted as:

$$\mathbf{Z}_{N+1} = \phi([\mathbf{X}, \mathbf{Z}^N]\mathbf{W}_{\mathbf{e}_{N+1}} + \beta_{e_{N+1}}) \tag{14}$$

where $\mathbf{Z}^N$ is a collection of feature nodes in initial structure.

Just like CFBLS-pyramid, The new feature nodes in CFBLS-dense also bring new enhancement nodes. The number of new enhancement nodes is M' which is also determined, The corresponding enhancement nodes are randomly generated as follows:

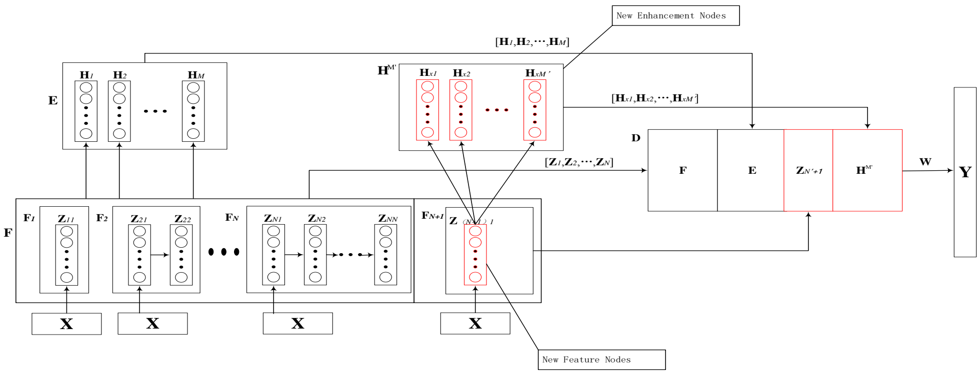
$$\mathbf{H}^{M'} = [\zeta(\mathbf{Z}_{N'+1}\mathbf{W}_{h_{x1}} + \beta_{h_{x1}}), \ldots, \zeta(\mathbf{Z}_{N'+1}\mathbf{W}_{h_{xM'}} + \beta_{h_{xM'}})] \tag{15}$$

where $\mathbf{W_h}, \beta_e$ are randomly generated. Then, we can get $\mathbf{D}^M_{N+1} = [\mathbf{D}^M_N|\mathbf{Z}_{N+1}|\mathbf{H}_P]$, the upgraded pseudoinverse matrix $(\mathbf{D}^M_{N+1})^+$ can be computed by Equation (11)$\sim$(12), and the new weights can get by Equation (13).

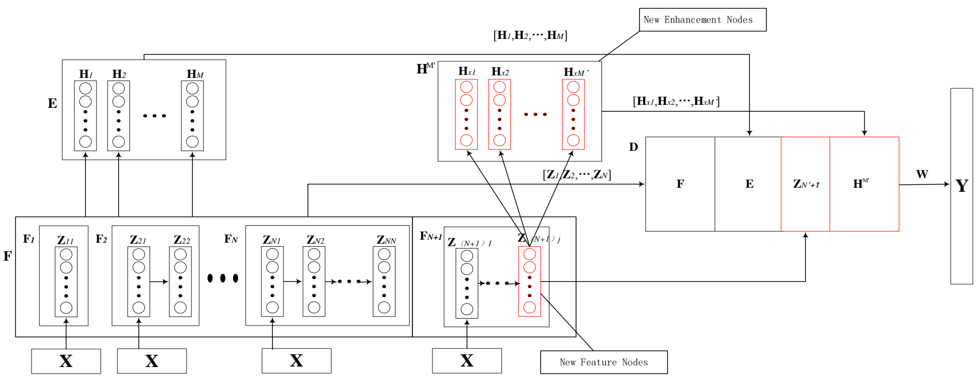
The incremental algorithm of the increment feature node is shown in Algorithm 2.

2.3 Incremental Learning for CEBLS-dense: Increment of Additional Enhancement Nodes

Enhancement nodes is very important in BLS, CEBLS-dense is a new structure which enhancement nodes are connected in a dense way. So, in this part, we will introduce the incremental algorithm of additional enhancement nodes. The process is shown in Figure 3.



(a) the new feature node is the first node of the last layer



(b) the new feature node is not the first node of the last layer

Figure 1: CFBLS-pyramid:Increment of N+1 Features

we will detail the broad expansion method for adding p enhancement nodes. Denote: $\mathbf{D}_N^{M+1} = [\mathbf{D}_N^M | \mathbf{H}_{(M+1)m}]$, where $\mathbf{H}_{(M+1)m}$ is the additional enhancement nodes. It can be culculated by follows:

$$\mathbf{H}_{(M+1)k} = \begin{cases} \zeta(\mathbf{D}_N^M \mathbf{W}_{\mathbf{h}_{(M+1)k}} + \beta_{h_{(M+1)k}}) & k = 1 \\ \\ \zeta(\mathbf{H}_{(M+1)(k-1)} \mathbf{W}_{\mathbf{h}_{(M+1)k}} + \beta_{h_{(M+1)k}}) & k = 2, \ldots, m \end{cases}$$

Then, we could deduce the pseudoinverse of the new matrix as:

$$(\mathbf{D}_N^{M+1})^+ = \begin{bmatrix} (\mathbf{D}_N^M)^+ - \mathbf{Q}\mathbf{B}^T \\ \mathbf{B}^T \end{bmatrix} \tag{16}$$

where $\mathbf{Q} = (\mathbf{D}_N^M)^+ \mathbf{H}_{(M+1)m}$

$$\mathbf{B}^T = \begin{cases} \mathbf{C}^+ & if \quad \mathbf{C} \neq 0 \\ \\ (1 + \mathbf{Q}^T\mathbf{Q})^{-1}\mathbf{Q}^T(\mathbf{D}_N^M)^+ & if \quad \mathbf{C} = 0 \end{cases} \tag{17}$$

and $\mathbf{C} = \mathbf{H}_{(M+1)m} - \mathbf{D}_N^M \mathbf{Q}$

So, the new weights are:

$$\mathbf{W}_N^{M+1} = \begin{bmatrix} \mathbf{W}_N^M - \mathbf{Q}\mathbf{B}^T\mathbf{Y} \\ \mathbf{B}^T\mathbf{Y} \end{bmatrix} \tag{18}$$

The incremental algorithm of the increment enhancement node is shown in Algorithm 3.

3 Experiments

We did three additional experiment as Table 1, Table 2 and Table 3 shown. The data set we used is MNIST data set [LeCun *et al.*, 1998]. [Chen and Liu, 2018] has proved that the incremental learning algorithms are very effective. The purpose of this experiment is to prove that the proposed network deformation can also achieve incremental and good results. The experiments are conducted on a MacOS 10.14 operating system, and the processor is Intel Core i7. And the experimental data in the table are the average of multiple experiments.

We did two kind of incremental experiments, increment of feature nodes and increment of enhancement nodes. Incremental experiments of feature nodes are implemented on CFBLS-pyramid and CFBLS-dense, Table 1 and Table 2 show the results. Incremental experiment of enhancement nodes is implemented on CEBLS-dense, Table 3 lists the results..

For Table 1, we tested the feature node incremental algorithm on CFBLS-pyramid. We selected 45 feature nodes and 600 enhancement nodes as initial data. Each increment adds 15 feature nodes and 900 enhancement nodes. From Table 1, we can see, with the increasing of nodes, the accuracy is improved gradually. The increase in training time is also in the acceptable range.

For Table 2, we tested the feature node incremental algo-

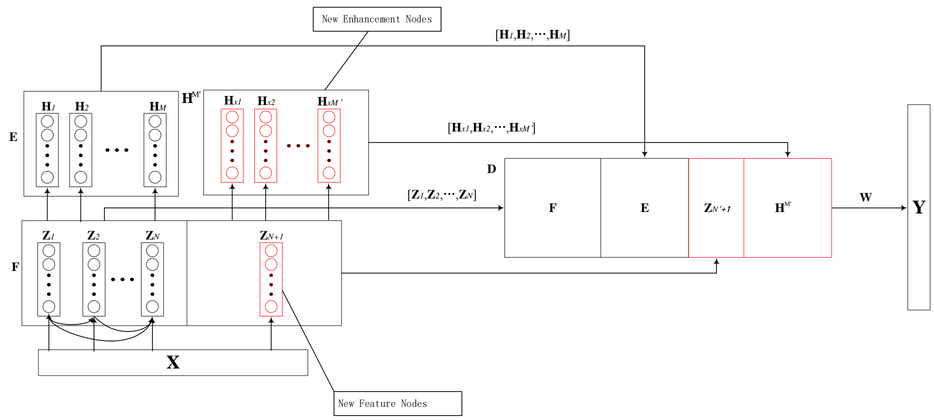


Figure 2: Incremental Learning for CFBLS-dense: Increment of Additional Feature Nodes.

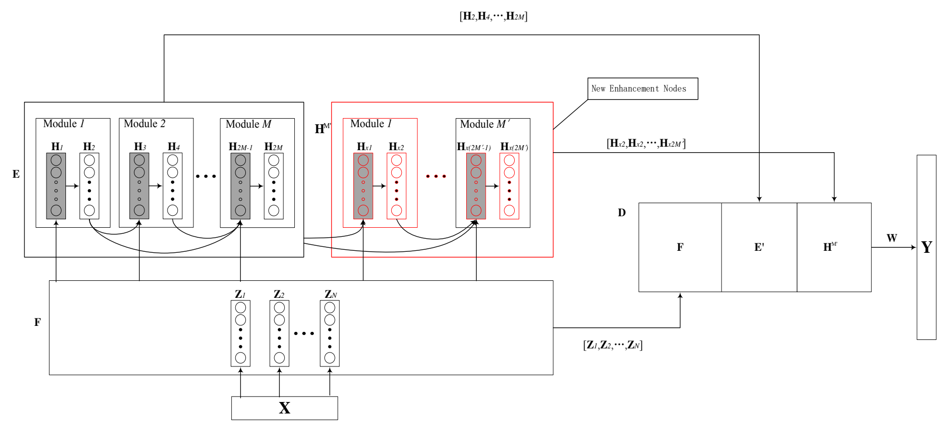


Figure 3: Incremental Learning for CEBLS-dense: Increment of Additional Enhancement Nodes.

rithm on CFBLS-dense. The initial number of feature nodes and enhancement nodes is 30 and 600. Each increment adds 15 feature nodes and 900 enhancement nodes. Table 2 shows the experimental results. It proves the effectiveness of incremental algorithm on CFBLS-dense.

For Table 3, we tested the enhancement node incremental algorithm on CEBLS-dense. We did an incremental experiment with enhancement nodes. At first, it has 45 feature nodes and 900 enhancement nodes, and each increment adds 400 enhancement nodes. Because of the disadvantage of enhancing the node increment, the data in the Table 3 is poorer than that in Table 1 and Table 2, but it also proves that CEBLS-dense can realize the increment algorithm of enhancement node, and can get the effect.

References

[Ben-Israel and Greville, 2003] Adi Ben-Israel and Thomas NE Greville. *Generalized inverses: theory and applications*, volume 15. Springer Science & Business Media, 2003.

[Chen and Liu, 2018] Clp Chen and Z. Liu. Broad learning system: An effective and efficient incremental learning system without the need for deep architecture. *IEEE Transactions on Neural Networks & Learning Systems*, 29(1):10–24, 2018.

[LeCun *et al.*, 1998] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[Serre, 2010] Denis Serre. Matrices. *Graduate Texts in Mathematics*, 216, 2010.

Number of Feature Nodes	Number of Enhancement Nodes	Testing Accuracy	Additional Training Times	Accumulative Training Time
45	600	95.15%	2.95s	2.95s
45 $\rightarrow$ 60	600 $\rightarrow$ 1500	96.85%	4.71s	7.66s
50 $\rightarrow$ 75	1500 $\rightarrow$ 2400	97.56%	6.38s	14.04s
75 $\rightarrow$ 90	2400 $\rightarrow$ 3300	97.89%	10.72s	24.76s
90 $\rightarrow$ 105	3300 $\rightarrow$ 4200	98.12%	13.14s	37.90s

Table 1: The Esperiments of CFBLS-pyramid on MNIST: Increment of N+1 Features Nodes

Number of Feature Nodes	Number of Enhancement Nodes	Testing Accuracy	Additional Training Times	Accumulative Training Time
30	600	95.2%	2.87s	2.87s
30 $\rightarrow$ 45	600 $\rightarrow$ 1500	97.24%	4.34s	7.21s
45 $\rightarrow$ 60	1500 $\rightarrow$ 2400	97.75%	6.27s	13.48s
50 $\rightarrow$ 75	2400 $\rightarrow$ 3300	97.94%	8.38s	21.86s
75 $\rightarrow$ 90	3300 $\rightarrow$ 4200	98.17%	10.53s	32.39s

Table 2: The Esperiments of CFBLS-dense on MNIST: Increment of N+1 Features Nodes

Number of Feature Nodes	Number of Enhancement Nodes	Testing Accuracy	Additional Training Times	Accumulative Training Time
45	900	96.47%	4.93s	4.93s
45	900 $\rightarrow$ 1300	96.84%	2.49s	7.42s
45	1300 $\rightarrow$ 1700	97.23%	2.98s	10.40s
45	1700 $\rightarrow$ 2100	97.33%	3.50s	13.90s
45	2100 $\rightarrow$ 2500	97.55%	3.99s	17.89s

Table 3: The Esperiments of CEBLS-dense on MNIST: Increment of N+1 Enhancement Nodes

Algorithm 1 CFBLS-pyramid:Increment of N+1 Features

Input: training samples $\mathbf{X}$
Output: $\mathbf{W}$

```
for $i = 1; i \leq N$ do
    Random $\mathbf{W}_{\mathbf{e}_{i1}}, \beta_{e_{i1}}$;
    Calculate $\mathbf{Z}_{i1} = \phi(\mathbf{X}\mathbf{W}_{\mathbf{e},ij} + \beta_{e,ij})$;
    for $j = 2; j \leq i$ do
        Random $\mathbf{W}_{\mathbf{e}_{ij}}, \beta_{e_{ij}}$;
        Calculate $\mathbf{Z}_{ij} = \phi(\mathbf{Z}_{ij-1}\mathbf{W}_{\mathbf{e}_{ij}} + \beta_{e_{ij}})$;
    end for
end for
All feature nodes:$\mathbf{Z}^{N'} = [\mathbf{Z}_{11}, \mathbf{Z}_{21}, \mathbf{Z}_{22}, \ldots, \mathbf{Z}_{NN}]$;
for $k = 1; k \leq M$ do
    Random $\mathbf{W}_{\mathbf{h}_k}, \beta_{h_k}$;
    Calculate $\mathbf{H}_k = \zeta(\mathbf{Z}^{N'} W_{\mathbf{h}_k} + \beta_{h_k})$
end for
All enhancement nodes: $\mathbf{H}^M = [\mathbf{H}_1, \ldots, \mathbf{H}_M]$
Set $\mathbf{D}^M_{N'} = [\mathbf{Z}^{N'} | \mathbf{H}^M]$
Calculate $(\mathbf{D}^M_{N'})^+$ and $\mathbf{W}^M_{N'}$
while *The number of feature nodes increased is less than the set number* do
    Random $\mathbf{W}_{\mathbf{e}_{N'+1}}, \beta_{e_{N'+1}}$;
    if $\mathbf{Z}_{N'+1}$ is the first feature node of the layer then
        Calculate $\mathbf{Z}_{N'+1} = \phi(\mathbf{X}\mathbf{W}_{\mathbf{e}_{N'+1}} + \beta_{e_{N'+1}})$;
    else
        Calculate $\mathbf{Z}_{N'+1} = \phi(\mathbf{Z}_{N'}\mathbf{W}_{\mathbf{e}_{N'+1}} + \beta_{e_{N'+1}})$;
    end if
    Random $\mathbf{W}_{\mathbf{h}_{xp}}, \beta_{e_{xp}}, p = 1, \ldots, M'$;
    Calculate $\mathbf{H}^{M'} = [\zeta(\mathbf{Z}_{N'+1}\mathbf{W}_{h_{x1}} + \beta_{h_{x1}}), \ldots, \zeta(\mathbf{Z}_{N'+1}\mathbf{W}_{h_{xM'}} + \beta_{h_{xM'}})]$
    Update $\mathbf{D}^M_{N'+1} = [\mathbf{D}^M_{N'} | \mathbf{Z}_{N'+1} | \mathbf{H}^{M'}]$
    Calculate $(\mathbf{D}^M_{N'+1})^+$ by Equation(11)∼(12)
    Calculate $\mathbf{W}^M_{N'+1}$ by Equation(13)
    N'=N'+1;
end while
return($\mathbf{W}^M_{N'+1}$)
```

Algorithm 2 CFBLS-dense:Increment of n+1 Features

Input: training samples $\mathbf{X}$
Output: $\mathbf{W}$

```
Random $\mathbf{W}_{\mathbf{e}_1}, \beta_{e_1}$;
Calculate $\mathbf{Z}_1 = \phi(\mathbf{X}\mathbf{W}_{\mathbf{e}_1} + \beta_{e_1})$;
for $i = 2; i \leq N$ do
    Random $\mathbf{W}_{\mathbf{e}_i}, \beta_{e_i}$;
    Calculate $\mathbf{Z}_i = \phi([\mathbf{X}, \mathbf{Z}_1, \ldots, \mathbf{Z}_{i-1}]\mathbf{W}_{\mathbf{e}_i} + \beta_{e_i})$;
end for
All feature nodes:$\mathbf{Z}^N = [\mathbf{Z}_1, \ldots, \mathbf{Z}_N]$;
for $j = 1; j \leq M$ do
    Random $\mathbf{W}_{\mathbf{h}_j}, \beta_{h_j}$;
    Calculate $\mathbf{H}_j = \zeta(\mathbf{Z}^N \mathbf{W}_{\mathbf{h}_j} + \beta_{h_j})$
end for
All enhancement nodes: $\mathbf{H}^M = [\mathbf{H}_1, \ldots, \mathbf{H}_M]$
Set $\mathbf{D}^M_N = [\mathbf{Z}^N | \mathbf{H}^M]$
Calculate $(\mathbf{D}^M_N)^+$ and $\mathbf{W}^M_N$
while *The number of feature nodes increased is less than the set number* do
    Random $\mathbf{W}_{\mathbf{e}_{N+1}}, \beta_{e_{N+1}}$;
    Calculate$\mathbf{Z}_{N+1} = \phi([\mathbf{X}, \mathbf{Z}^N]\mathbf{W}_{\mathbf{e}_{N+1}} + \beta_{e_{N+1}})$
    Random $\mathbf{W}_{\mathbf{h}_{xp}}, \beta_{e_{xp}}, p = 1, \ldots, M'$;
    Calculate $\mathbf{H}^{M'} = [\zeta(\mathbf{Z}_{N'+1}\mathbf{W}_{h_{x1}} + \beta_{h_{x1}}), \ldots, \zeta(\mathbf{Z}_{N'+1}\mathbf{W}_{h_{xM'}} + \beta_{h_{xM'}})]$
    Update $\mathbf{D}^M_{N+1} = [\mathbf{D}^M_N | \mathbf{Z}_{N+1} | \mathbf{H}^M]$
    Calculate $(\mathbf{D}^M_{N+1})^+$ by Equation(11)∼(12)
    Calculate $\mathbf{W}^M_{N+1}$ by Equation(13)
    N=N+1;
end while
return($\mathbf{W}^M_{N+1}$)
```

Algorithm 3 CEBLS-dense:Increment of n+1 enhancenmet

Input: training samples $\mathbf{X}$

Output: $\mathbf{W}$

for $i = 1; i \leq N$ **do**
  Random $\mathbf{W}_{\mathbf{e},i}, \beta_{e,i}$;
  Calculate $\mathbf{Z}_{\boldsymbol{i}} = \phi(\mathbf{X}\mathbf{W}_{\boldsymbol{e,i}} + \beta_{e,i})$;
end for
All feature nodes:$\mathbf{Z}^{\boldsymbol{N}} = [\mathbf{Z}_{\boldsymbol{1}}, \ldots, \mathbf{Z}_{\boldsymbol{N}}]$;
for $j = 1; j \leq M$ **do**
  for $k = 1; k \leq m$ **do**
    if $j = 1$ AND $k = 1$ **then**
      Random $\mathbf{W}_{\mathbf{h},1}, \beta_{h,1}$;
      Calculate $\mathbf{H}_{\boldsymbol{11}} = \zeta(\mathbf{Z}^{\boldsymbol{N}}\mathbf{W}_{\boldsymbol{h,11}} + \beta_{h,11})$;
    else if $j \neq 1$ AND $k = 1$ **then**
      Random $\mathbf{W}_{\mathbf{h},jk}, \beta_{h,jk}$;
      Calculate $\mathbf{H}_{\boldsymbol{jk}} = \zeta([\mathbf{Z}^{\boldsymbol{N}}, \mathbf{H}_{\boldsymbol{1m}}, \ldots, \mathbf{H}_{\boldsymbol{(j-1)m}}]\mathbf{W}_{\boldsymbol{h,jk}} + \beta_{h,jk})$;
    else
      Random $\mathbf{W}_{\mathbf{h},jk}, \beta_{h,jk}$;
      Calculate $\mathbf{H}_{\boldsymbol{jk}} = \zeta(\mathbf{H}_{\boldsymbol{j(k-1)}}\mathbf{W}_{\boldsymbol{h,jk}} + \beta_{h,jk})$;
    end if
  end for
end for
All enhancement nodes: $\mathbf{H}^M = [\mathbf{H}_{1m}, \ldots, \mathbf{H}_{Mm}]$
Set $\mathbf{D}_N^M = [\mathbf{Z}^N | \mathbf{H}^M]$
Calculate $(\mathbf{D}_N^M)^+$ and $\mathbf{W}_N^M$
while *The number of enhancement nodes increased is less than the set number* **do**
  for $k = 1; k \leq m$ **do**
    if $k = 1$ **then**
      Random $\mathbf{W}_{\mathbf{h},(M+1)1}, \beta_{h,(M+1)1}$;
      Calculate $\mathbf{H}_{\boldsymbol{(M+1)1}} = \zeta([\mathbf{Z}^{\boldsymbol{N}}, \mathbf{H}^M]\mathbf{W}_{\boldsymbol{h,(M+1)1}} + \beta_{h,(M+1)1})$;
    else
      Random $\mathbf{W}_{\mathbf{h},(M+1)k}, \beta_{h,(M+1)k}$;
      Calculate $\mathbf{H}_{\boldsymbol{(M+1)k}} = \zeta(\mathbf{H}_{\boldsymbol{(M+1)(k-1)}}\mathbf{W}_{\boldsymbol{h,(M+1)k}} + \beta_{h,(M+1)k})$;
    end if
  end for
  Update $\mathbf{D}_N^{M+1} = [\mathbf{D}_N^M | \mathbf{H}_{(M+1)m}]$,
  Calculate $(\mathbf{D}_N^{M+1})^+$ by Equation(16)$\sim$(17)
  Calculate $\mathbf{W}_N^{M+1}$ by Equation(18)
  M=M+1;
end while
return($\mathbf{W}_{N+1}^M$)