Semi-adaptive Synergetic Two-way Pseudoinverse Learning System

CB.SC.U4AIE24205-SAI REDDY

CB.SC.U4AIE24213-DEVANA

CB.SC.U4AIE24251-MANAS

CB.SC.U4AIE24261-ZAHWA

Problem Statement

- **Limitations of Gradient Descent (GD):**
  - Requires iterative training → high computational cost
  - Sensitive to learning rate and hyperparameters
  - Suffers from vanishing/exploding gradients in deep networks
- **Architectural Challenges:**

- Choosing optimal network depth is manual and non-adaptive

- Over-or under-fitting due to fixed architecture

Why Pseudoinverse Learning?

General linear system:

$$Y = WX$$

To estimate W, pseudoinverse is used:

$$W = X'\mathrm{Y}$$

Where:

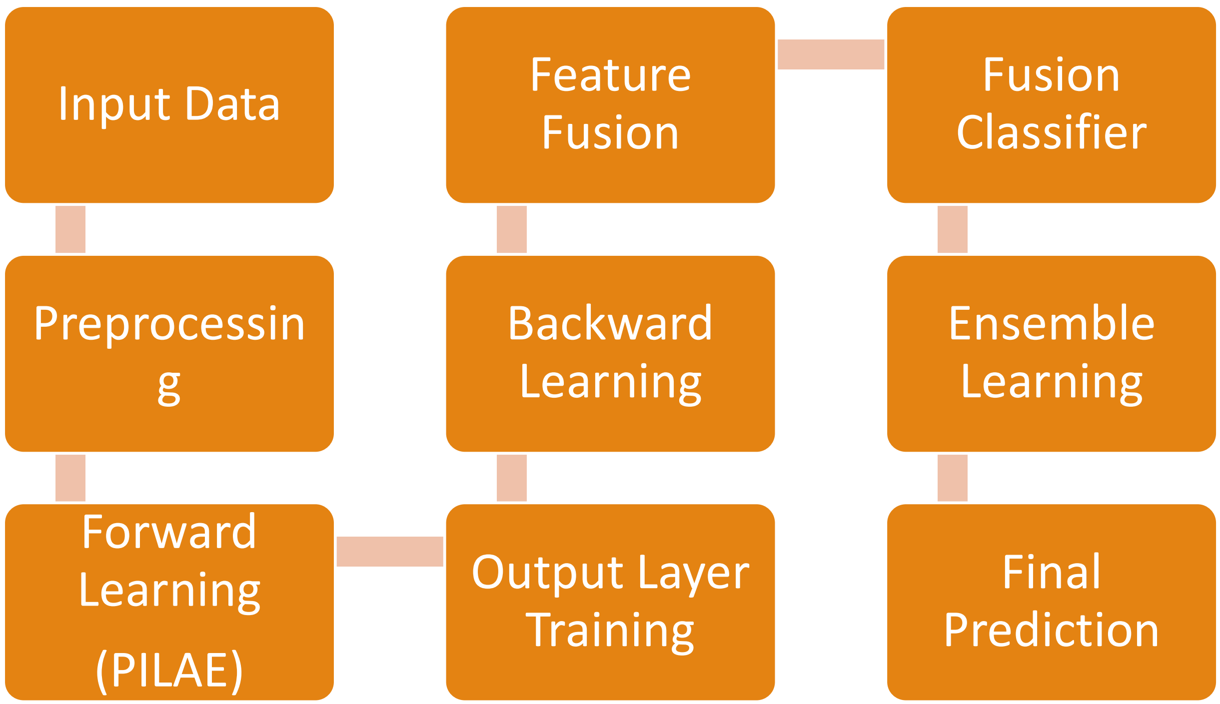
X’ -> Moore-Penrose pseudoinverse

X: Input data matrix

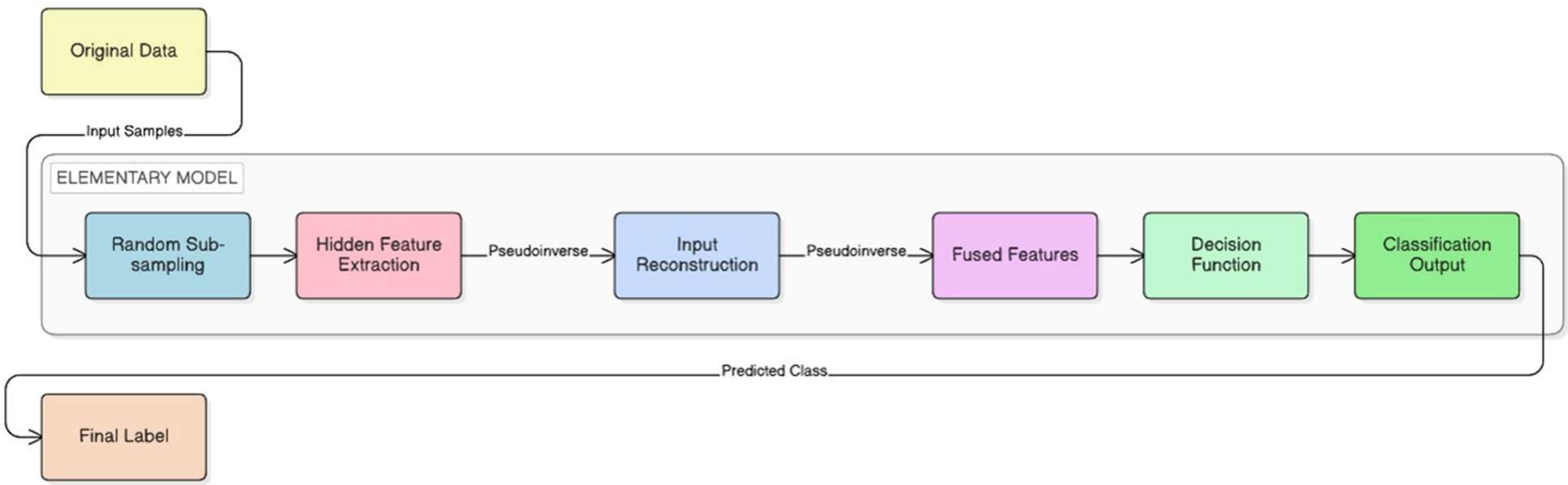
W: Weight matrix

Y: Output matrix

METHODOLOGY



Two-way Pseudoinverse Learning: Single Elementary Model



Toy example

Step 1: Input Data

Assume a very small dataset with **2 samples and 2 features**:

Sample	Feature 1	Feature 2
x_1	1	0
x_2	0	1

Input matrix : $X = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

Step 2: Forward Learning (Input → Hidden)

Assume a **randomly chosen forward weight matrix**:

$$W_1 = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

W_1 = Forward weight matrix

$H = X \times W_1$

$$= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \text{x} \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

$H=\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$ H = hidden matrix

Step 3: Pseudoinverse Learning (Hidden → Output)

- To reconstruct the input **without gradient descent**, compute output weights using pseudoinverse:

$W_2 = \text{pinv}(H) \times X$

W_2 = backward weight matrix

- Since H is square and invertible here:

  $\text{pinv}(H) = H^{-1}$

$W_2 = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}$

Step 4: Backward Learning (Reconstruction)

Reconstructed input:

$\check{X} = H \times W_2$

$= \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \text{x} \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

So $\check{X} = X$

Step 5: Feature Fusion (Two-way Learning)

Combine forward and backward representations:

Fused Features = [H | $\hat{X}$]

$$= \begin{bmatrix} 1 & 1 & | & 1 & 0 \\ 0 & 1 & | & 0 & 1 \end{bmatrix}$$

$$F = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

F = Feature fused matrix

Pseudoinverse Learning Based Autoencoder (PILAE)

Reconstruction Objective:

$$\min \parallel X - W_d H \parallel_F^2$$

Encoding:

$$H = f(W_e^T X)$$

Analytical Weight Solution:

$$W_d = XH^{\dagger}$$

Where:

X: input data

H: hidden representation

$H^{\dagger}$:Moore–Penrose pseudoinverse

$\mathrm{W_e}$: $enocoder\ Weights$

$\mathrm{W_d}$: decoder Weights

Forward Learning

Layer-wise forward feature extraction:

$$H_1 = \sigma(W_1^T X)$$

$$H_2 = \sigma(W_2^T H_1)$$

$$\vdots$$

$$F(X) = H_L = \sigma(W_L^T H_{L-1})$$

Where:

W_l –analytically learned weight matrix of layer l

$\sigma(\cdot)$ –activation function (tanh)

L– number of layers (determined adaptively using validation)

$F(X)$ –final forward-learned feature representation

Output Layer Training

Prediction:

$$Y = W_0 F(X)$$

Closed-form solution for output weights:

$$W_0 = TF^{\dagger}$$

(Regularized form):

$$W_0 = TF^T \left(FF^T + \lambda I\right)^{-1}$$

Where:

$F(X)$ –forward learned feature matrix

T– true label matrix

$F^{\dagger}$ –Moore–Penrose pseudoinverse

λ– regularization parameter

Backward Learning

Label-driven backward feature reconstruction:

$$H_L^b = W_0^T\, T$$

$$H_{l-1}^b = W_l^b\, \sigma^{-1}\left(H_l^b\right), l = L, \dots, 2$$

Where:

H_l^b –backward features at layer l

W_0 –analytically learned output weight matrix

W_l^b –backward (decoder) weight of layer l

$\sigma^{-1}(\cdot)$ –inverse activation function

T– target label matrix

Feature Fusion

Fusion of forward and backward features:

$$Z = \begin{bmatrix} H_f \\ H_b \end{bmatrix}$$

Feature normalization:

$$Z \leftarrow Norm(Z)$$

Fusion classifier training:

$$W_f = Z^{\dagger} T$$

Where:

H_f –forward learned features

H_b –backward learned features

Z– fused feature representation

$Z^{\dagger}$ –Moore–Penrose pseudoinverse

T– target label matrix

DATASET

MNIST Dataset

Standard benchmark dataset for handwritten digit recognition

10 classes (digits 0–9)

60,000 training samples

10,000 testing samples

Image size: **28 × 28 pixels**

Each image flattened into a **784-dimensional feature vector**

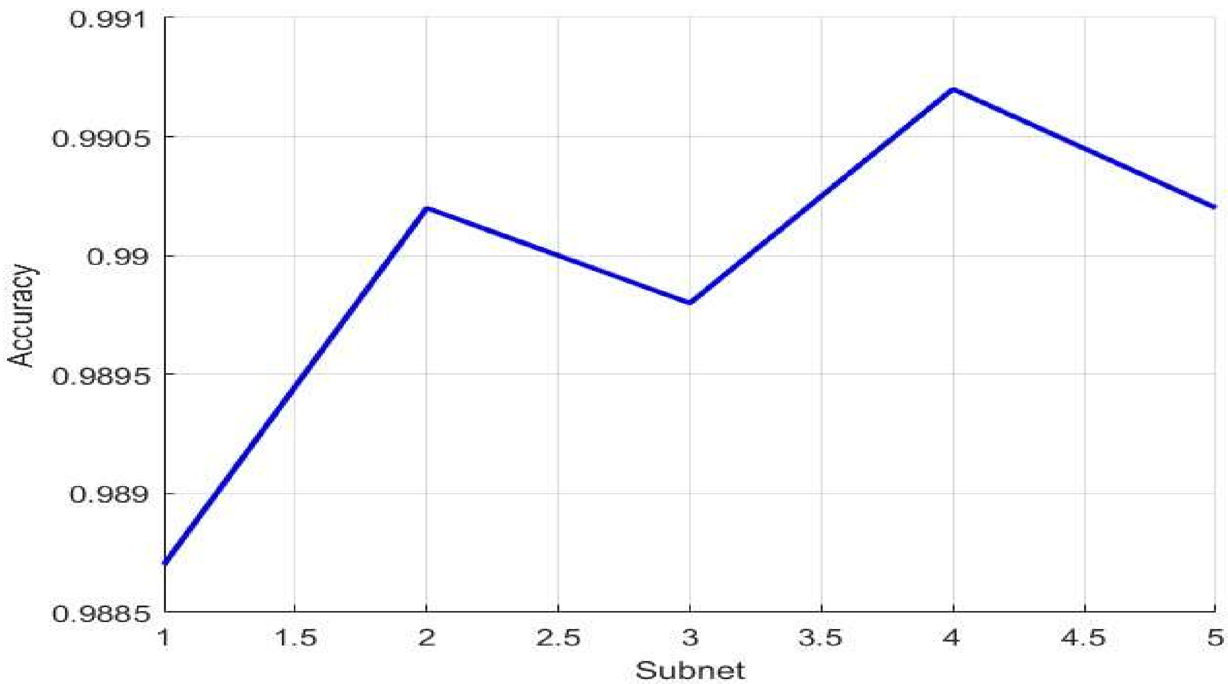
Preprocessing:

Z-score normalization of input features

One-hot encoding of target labels

RESULTS

Model component	Accuracy
Forward subnet -1	96.02 %
Forward subnet -2	99.03%
Forward subnet -3	98.8%
Forward subnet -4	99.07%
Forward subnet -5	99.03 %
Final ensemble	99.02 %


0.991
0.9905
0.99
0.9895
0.989
0.9885
Accuracy
1
1.5
2
2.5
3
3.5
4
4.5
5
Subnet

CONCLUSION

- Two-way learning = richer features
- Synergetic subsystems improve robustness
- Non-gradient training = faster learning
- Adaptive depth using early stopping