Machine Learning

Prof.G.K.Mahanti
Dept. of ECE
National Institute of Technology
Durgapur

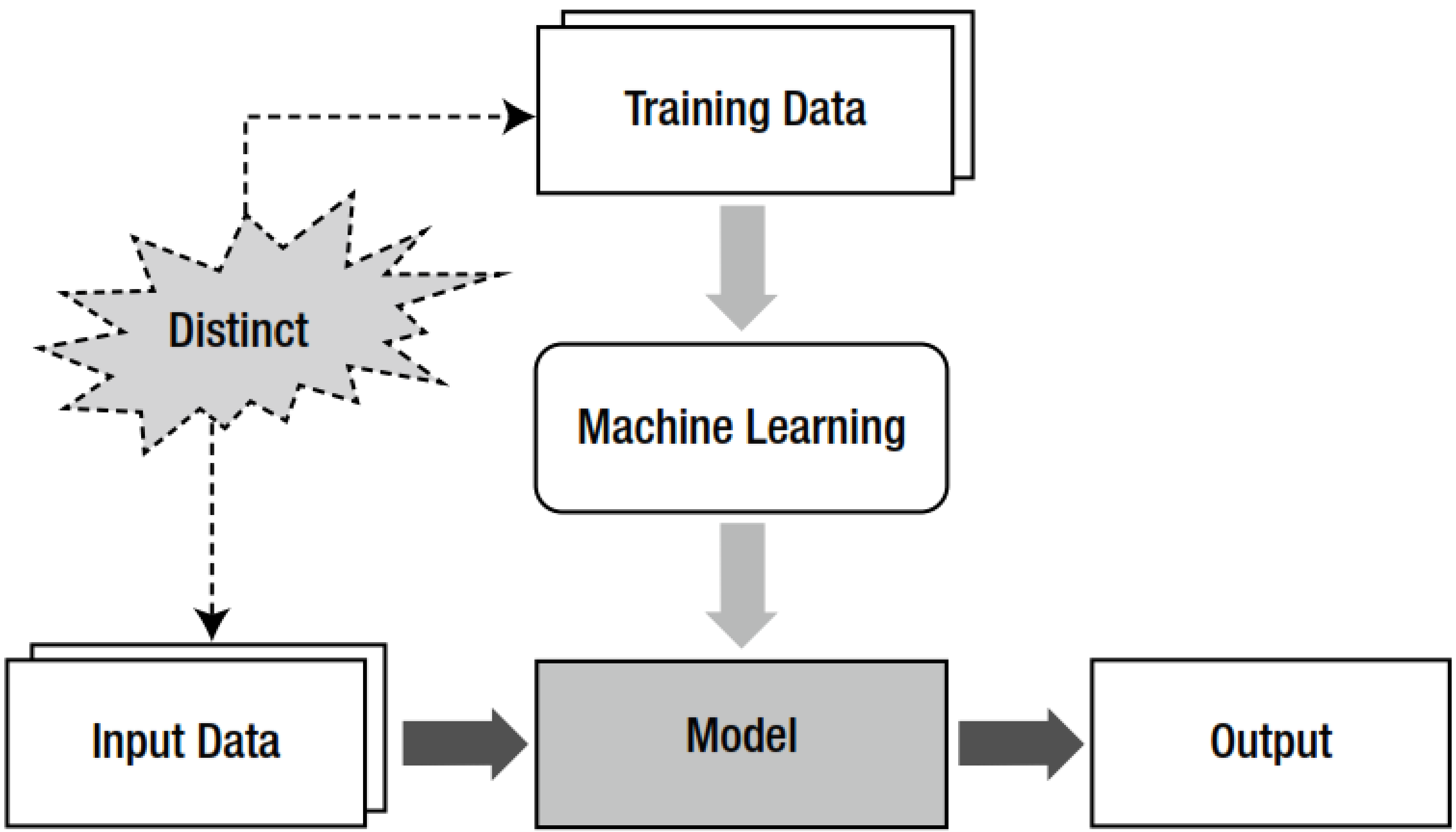
Deep learning versus Machine learning:

Deep Learning is a kind of Machine Learning, and Machine Learning is a kind of Artificial Intelligence

Artificial Intelligence is a very common word that may imply many different things. It may indicate any form of technology that includes some intelligent aspects rather than pinpoint a specific technology field.

Machine Learning is a technique that figures out the “model” out of “data.” Here, the data literally means information such as documents, audio, images, etc. The “model” is the final product of Machine Learning.

Machine Learning is the technique used to find (or learn) a model from the data. It is suitable for problems that involve intelligence, such as image recognition and speech recognition, where physical laws or mathematical equations fail to produce a model.



The distinctness of the training data and input data is the structural challenge that Machine Learning faces. It is no exaggeration to say that every problem of Machine Learning originates from this.

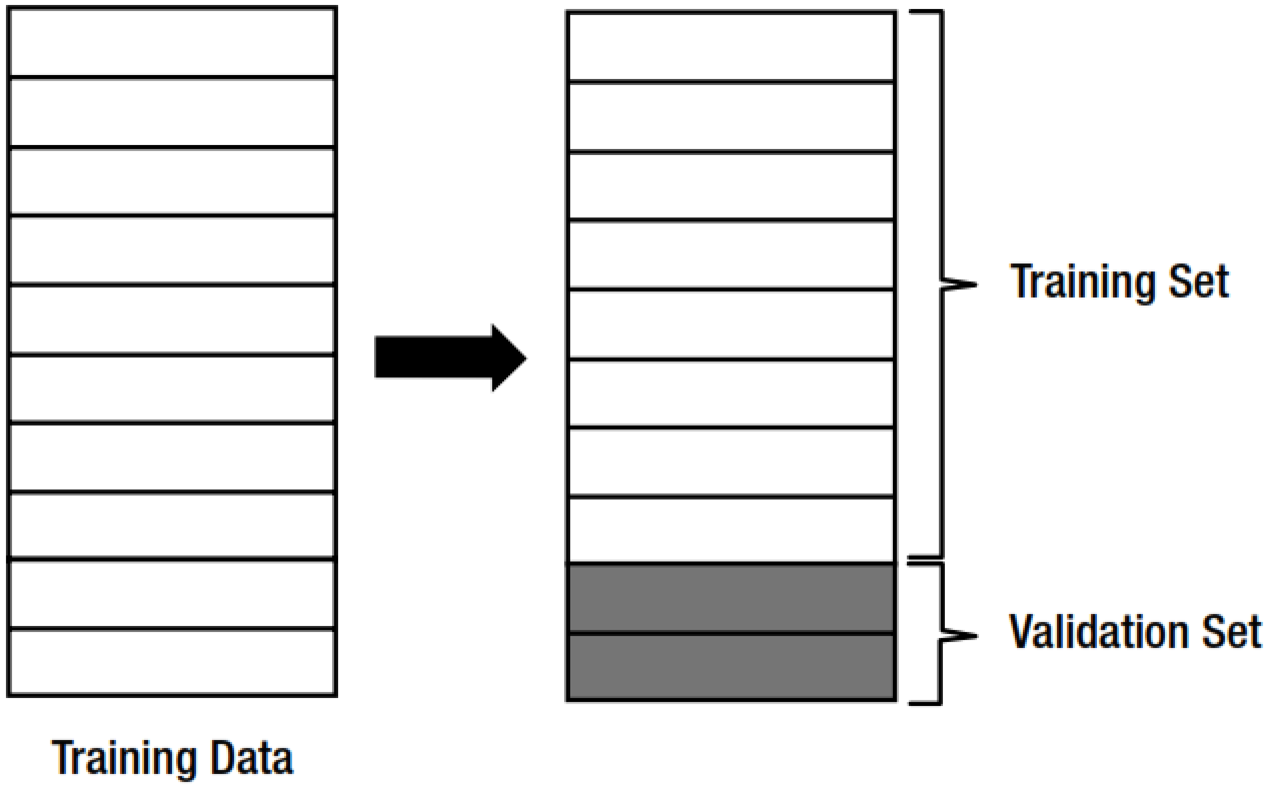
The process used to make the model performance consistent regardless of the training data or the input data is called generalization. The success of Machine Learning relies heavily on how well the generalization is accomplished.

Overfitting:

One of the primary causes of corruption of the generalization process is *overfitting.*

The validation is a process that reserves a part of the training data and uses it to monitor the performance. The validation set is not used for the training process. Because the modeling error of the training data fails to indicate overfitting, we use some of the training data to check if the model is overfitted.

We can say that the model is overfitted when the trained model yields a low level of performance to the reserved data input. In this case, we will modify the model to prevent the overfitting.



When validation is involved, the training process of Machine Learning proceeds by the following steps:

1. Divide the training data into two groups: one for training and the other for validation. As a rule of thumb, the ratio of the training set to the validation set is 8:2.
2. Train the model with the training set.
3. Evaluate the performance of the model using the validation set.

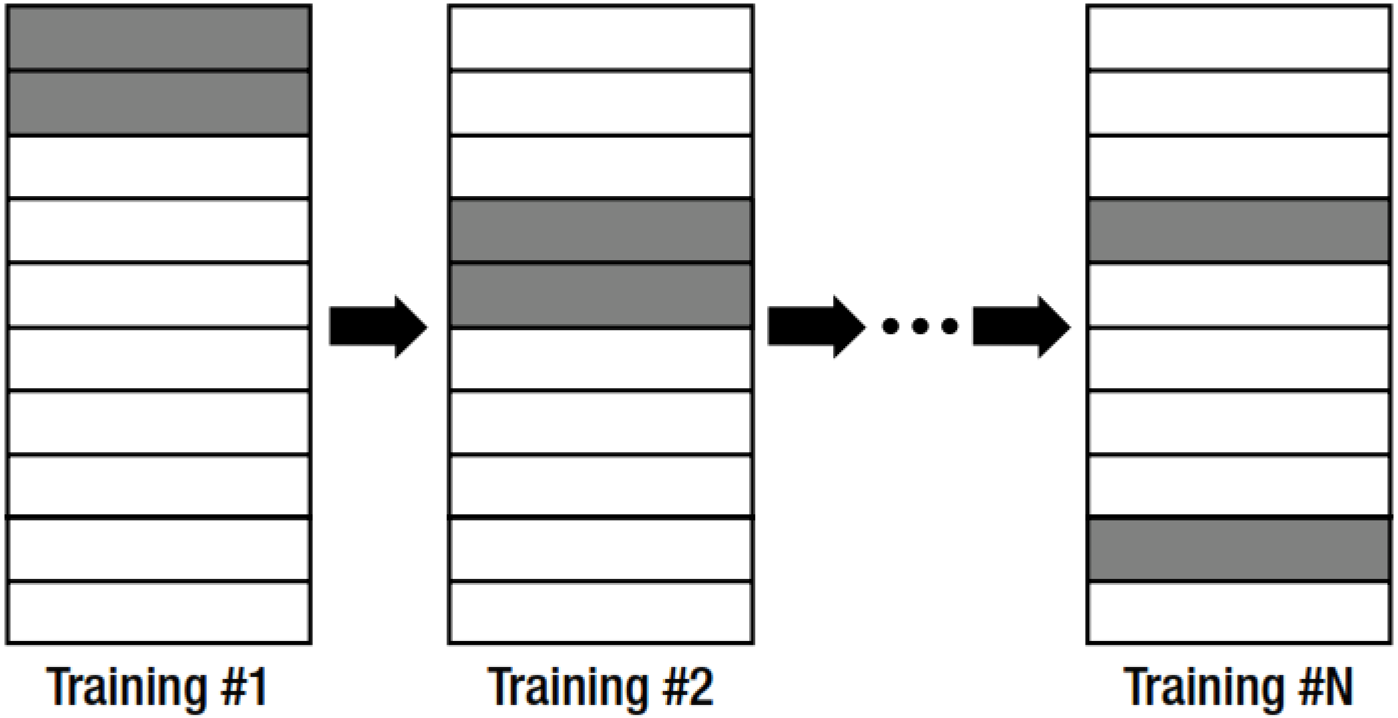
a. If the model yields satisfactory performance, finish the training.

b. If the performance does not produce sufficient results, modify the model and repeat the process from Step 2.

Cross-validation is a slight variation of the validation process. It still divides the training data into groups for the training and validation, but keeps changing the datasets.

Instead of retaining the initially divided sets, cross-validation repeats the division of the data. The reason for doing this is that the model can be overfitted even to the validation set when it is fixed.

As the cross-validation maintains the randomness of the validation dataset, it can better detect the overfitting of the model

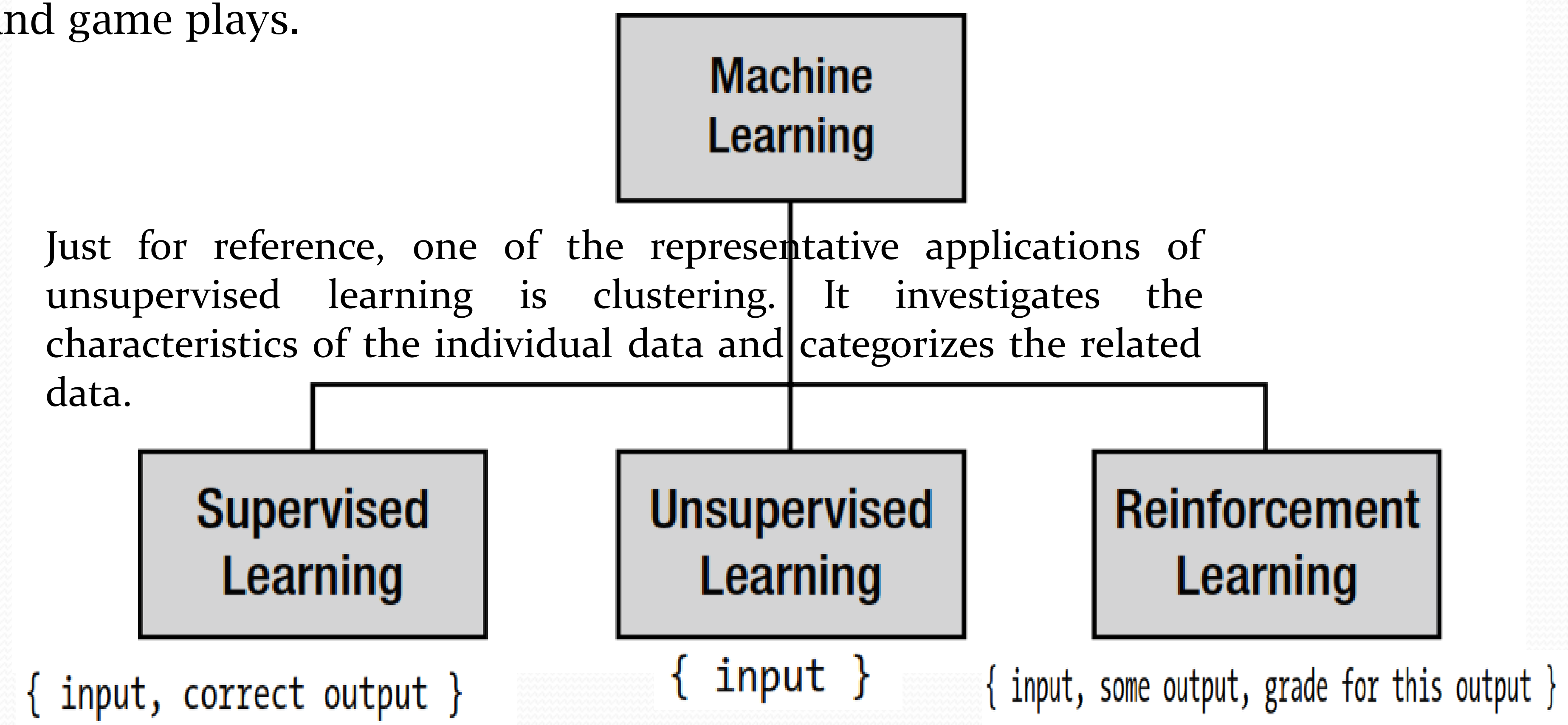


Types of Machine Learning:

Unsupervised learning is generally used for investigating the characteristics of the data and pre-processing the data.

Reinforcement learning is generally used when optimal interaction is required, such as control and game plays.

Just for reference, one of the representative applications of unsupervised learning is clustering. It investigates the characteristics of the individual data and categorizes the related data.



Classification and Regression:

The two most common types of application of supervised learning are classification and regression. These words may sound unfamiliar, but are actually not so challenging.

In the classification problem, we want to know which class the input belongs to. So the data pair has the class in place of the correct output corresponding to the input.

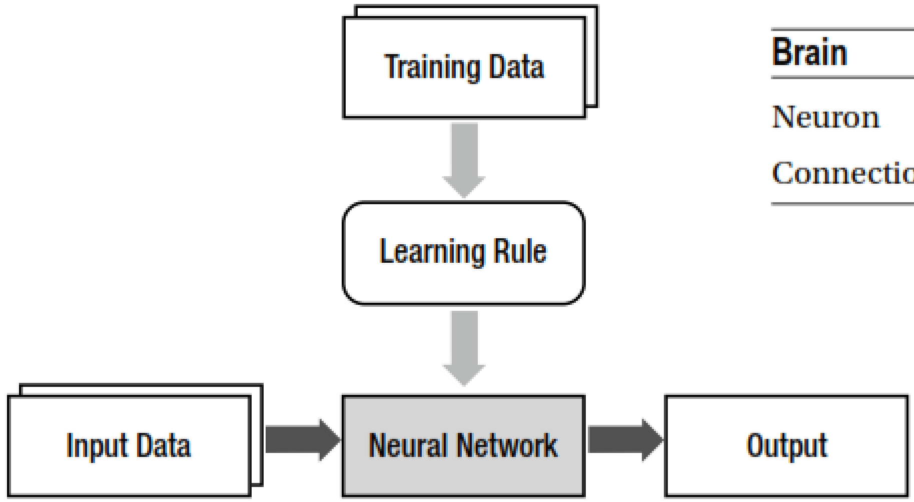
Digit recognition service ➔ Classifies the digit image into one of 0-9
Face recognition service ➔ Classifies the face image into one of the registered users

```
{ input, class }
```

In contrast, the regression does not determine the class. Instead, it estimates a value. As an example, if you have datasets of age and income (indicated with a •) and want to find the model that estimates income by age, it becomes a regression problem .

Neural Network:

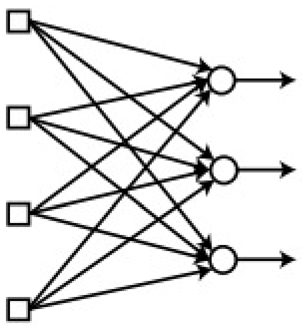
Neural network is widely used as the model for Machine Learning.



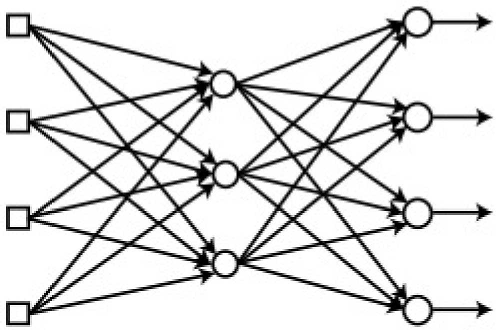
Brain	Neural Network
Neuron	Node
Connection of neurons	Connection weight

The neural network imitates the mechanism of the brain. As the brain is composed of connections of numerous neurons, the neural network is constructed with connections of nodes, which are elements that correspond to the neurons of the brain. The neural network mimics the neurons' association, which is the most important mechanism of the brain, using the weight value. The above table summarizes the analogy between the brain and neural network.

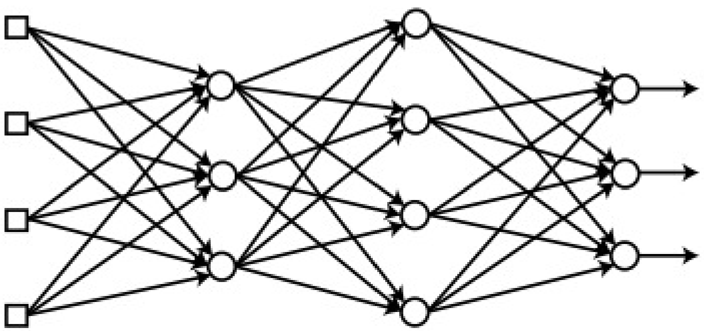
Single-Layer Neural Network		Input Layer – Output Layer
Multi-Layer Neural Network	Shallow Neural Network	Input Layer – Hidden Layer – Output Layer
	Deep Neural Network	Input Layer – Hidden Layers – Output Layers



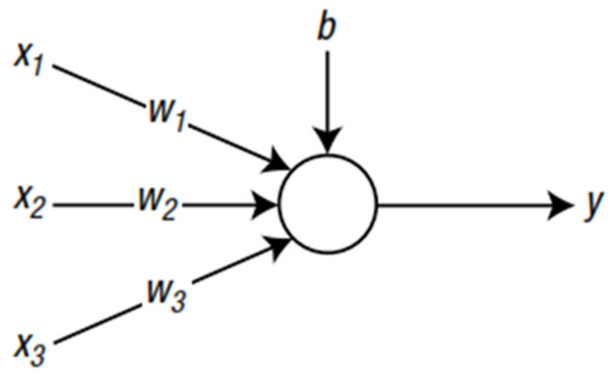
Single-layer Neural Network



(Shallow) Multi-layer Neural Network



Deep Neural Network



$$v = wx + b$$

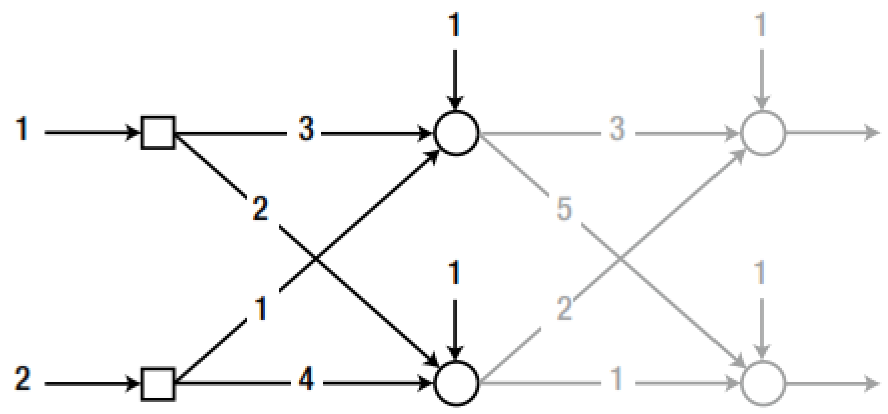
where w and x are defined as:

$$w = \begin{bmatrix} w_1 & w_2 & w_3 \end{bmatrix} \qquad x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

Finally, the node enters the weighted sum into the activation function and yields its output. The activation function determines the behavior of the node.

$$y = \varphi(v)$$

$\varphi(\cdot)$ of this equation is the activation function.



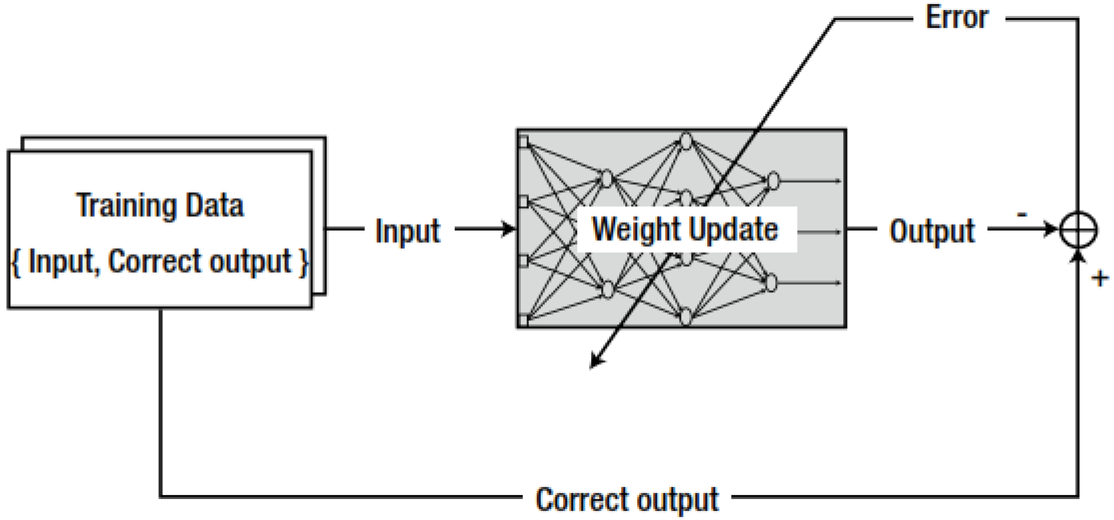
$$v = \begin{bmatrix} 3\times1+1\times2+1 \\ 2\times1+4\times2+1 \end{bmatrix} = \begin{bmatrix} 3 & 1 \\ 2 & 4 \end{bmatrix}\begin{bmatrix} 1 \\ 2 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 6 \\ 11 \end{bmatrix}$$

$$v = Wx + b$$

Weighted sum: $v = \begin{bmatrix} 3 & 2 \\ 5 & 1 \end{bmatrix}\begin{bmatrix} 6 \\ 11 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 41 \\ 42 \end{bmatrix}$

Output: $y = \varphi(v) = v = \begin{bmatrix} 41 \\ 42 \end{bmatrix}$

Supervised learning:



Extreme learning machine:

- Extreme learning machine(ELM)is a new learning algorithm proposed for both Single-Layer Feed-forward Networks (SLFNs) and multi layered feed-forward networks(MLFNs).
- Compared with the conventional neural network learning algorithm it overcomes the slow training speed and over-fitting problem.
- In the ELM algorithm, the learning parameters of hidden nodes, including input weights and biases, can be randomly assigned independently, and the output weights of the network can be analytically calculated.

Now , considering a ELM of x inputs and y outputs:

- The hidden layer output H is calculated with the help of the activation function of the hidden layer. Mathematically it can be represented as:

$$H = \phi(wX + b)$$

- Where ϕ is the activation function;
- w is the input weights;
- b is the input bias

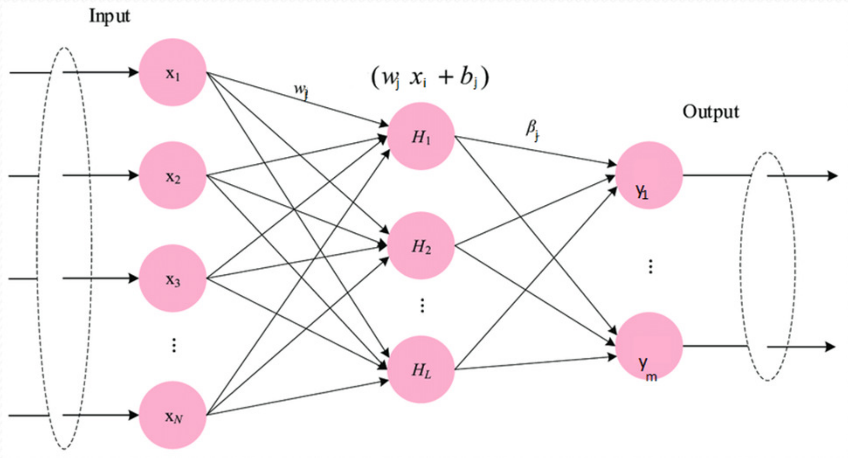


fig2:representation of a ELM with w i/p weight ,b i/p bias to the hidden layer

- From the fig.2 which have a set of N distinct training samples (x_i) , with $i \in [[1, N]]$, $x_i \in R$. Then a SLFN with L hidden neurons has the following output equation:

$$y_i = \sum_{j=1}^{L} \beta_j \phi(w_j x_i + b_j), i \in [[1, N]] \quad \text{...equation 2}$$

; where β is the output weight.

- The output weights β must be determined analytically using the pseudo-inverse of the hidden layer.
- Activation function ϕ can any of the function of: 'sig', 'tansig', 'logsig', 'elliotsig', 'radbas', 'tribas', 'sin', 'cos', 'hardlim'.

In the matrix form of ELM

- In the matrix form, it can be written as: Output from hidden layer $(N \times L)$,

$$Y_{N\times L} = AX' + B$$

where A=input weights matrix of size $(N \times M)$;

X=input matrix of size$(L \times M)$.

where L=no. of input sequences

M=no. of attributes.

- Final output=$Z = Y'C$

where B=bias matrix of size$(N \times L)$,

C=output weight matrix of size(N × P)(unknown)

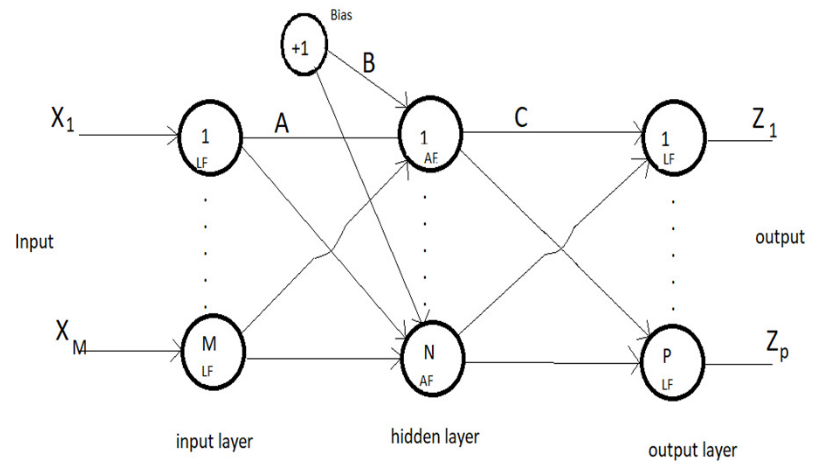


fig3:matrix form of a ELM

where,A ∈ (0,1)...connecting weights

B ∈ (-1,1)...biasing weight

C...unknown weight

- so, final output : $Y^T C = Z$,Z=final output of size($L \times P$)

$YY^T C = YZ$; C=inv(Y^T)Z

C $= (YY')^{-1} YZ$

C= pinv $(Y')Z$ ← moore-penrose pseudo-inverse of matrix

If N<L (under-determined)

$C = (YY^T + \lambda I)^{-1}$YZ ← Ridge Regression

If N>L (over-determined)

$C = ((Y'Y + \lambda I)^{-1} Y')' Z$ ← Ridge Regression

References:

- C.A. Balanis, Antenna Theory (Wiley, New York, 1997).
- A Multiple Hidden Layers Extreme Learning Machine Method and Its Application: Dong Xiao, Beijing Li, and Yachun Mao.
- High-Performance Extreme Learning Machines:A Complete Toolbox for Big Data Applications: ANTON AKUSOK, KAJ-MIKAEL BJÖRK, YOAN MICHE, AND AMAURY LENDASSE
- G.B. Huang, Q.-Y. Zhu, and C.-K. Siew, ‘‘Extreme learning machine: A new learning scheme of feedforward neural networks,’’ in Proc. IEEE Int. Joint Conf. Neural Network., vol. 2. Jul. 2004, pp. 985–990.
- A. Patnaik, D.E. Anagnostou et al. Applications of neural networks in wireless communication. IEEE Antenna Propagation. Mag., 46(3), 130–137 (2004).
- C. E. Rasmussen and C. K. I. Williams, Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning). The MIT Press, 2005.
- Neural computation of resonant frequency of electrically thin and thick rectangular micro-strip antennas :D. Karaboga, K.Guney ,S.Sagiroglu ,M.Erler .

Thank you