Neural Networks

Lecture 10

Learning and Adaptation (1)

Md. Mijanur Rahman, Prof. Dr.

Dept. of Computer Science and Engineering, Jatiya Kabi Kazi Nazrul Islam University, Bangladesh.

Web: www.mijanrahman.com | Email: mijanjkkniu@gmail.com; mijan@jkkniu.edu.bd

Chapter Contents

- **This chapter covers the following topics:**
  - **Learning in ANN**
  - **Different Learning Paradigms**
  - **Offline or Online Learning**
  - **Training patterns and teaching input**
  - Learning curve and error measurement
  - Gradient Descent Algorithm and Optimization Procedures
  - Cost Function and The Learning Rate
  - Different Learning Rules

Learning and Adaptation

- As stated earlier, ANN is completely inspired by the way biological nervous system, i.e. the human brain works. The most impressive characteristic of the human brain is to learn, hence the same feature is acquired by ANN.
- It is accepted that during the learning procedure, the brain's neural structure is altered, increasing or decreasing the capacity of its synaptic connections relying on their activity.
- This is the reason why more relevant information is simpler to review than information that has not been reviewed for a long time.
- Thus, more significant information will have powerful synaptic connections, and less applicable information will gradually have its synaptic connections weaken, making it harder to review.

Learning in ANN

- **What is Learning in ANN?**
  - Basically, learning means to do and adapt the change in itself as and when there is a change in environment.
  - As ANN is a complex system, we can say that it is a complex adaptive system, which can change its internal structure based on the information passing through it.
- **Why is It important?**
  - Being a complex adaptive system, learning in ANN implies that a processing unit is capable of changing its input/output behavior due to the change in environment.
  - The importance of learning in ANN increases because of the fixed activation function as well as the input/output vector, when a particular network is constructed.
  - Due to change the input/output behavior, we need to adjust the weights.

Different Paradigms of Learning…

- Learning in ANN is a comprehensive term. A learning system changes itself in order to adapt to, e.g., environmental changes. A neural network could learn from many things but, of course, there will always be the question of how to implement it.
- In principle, a neural network changes when its components are changing. Theoretically, a neural network could learn by-
  1. developing new connections,
  2. deleting existing connections,
  3. changing connecting weights,
  4. changing the threshold values of neurons,
  5. varying one or more of the three neuron functions (remember: activation function, propagation function and output function),
  6. developing new neurons, or
  7. deleting existing neurons (and so, of course, existing connections).

Different Paradigms of Learning…

- A learning procedure is always an algorithm that can easily be implemented by means of a programming language.
- As discussed previous chapter, there are three essential paradigms of learning by presenting the differences between their regarding training sets in ANN. A training set is a set of training patterns, which we use to train our neural net.

1. Unsupervised learning.
2. Reinforcement learning
3. Supervised learning

Different Paradigms of Learning

- **Unsupervised learning provides input patterns to the network, but no learning aids.**
  - **Definition (Unsupervised learning):** The training set only consists of input patterns; the network tries by itself to detect similarities and to generate pattern classes.
- **Reinforcement learning methods provide feedback to the network, whether it behaves well or badly.**
  - **Definition (Reinforcement learning):** The training set consists of input patterns. After completing a sequence, a value is returned to the network, indicating whether the result was correct or incorrect and, possibly, how close it was.
- **Supervised learning methods provide training patterns together with appropriate desired outputs**
  - **Definition (Supervised learning):** The training set consists of input patterns with correct results so that the network can receive a precise error vector.

Learning Example: Classification

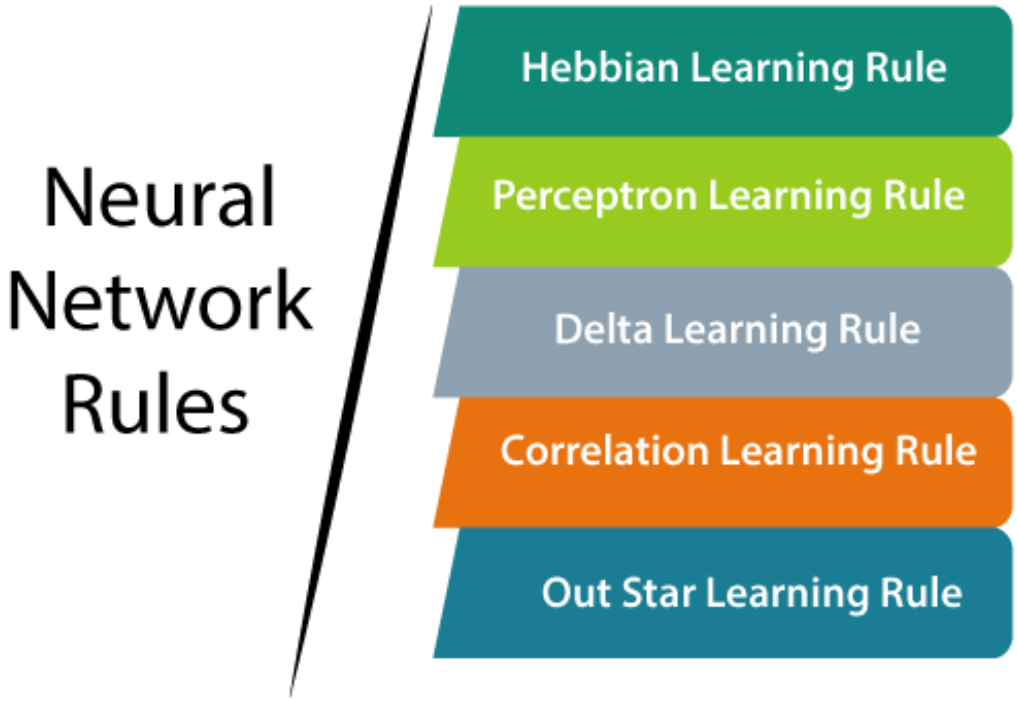
- **Classification** may be defined as the process of learning to distinguish the data of samples into different classes by finding common features between the samples of the same classes.
- For example, to perform training of ANN, we have some training samples with unique features, and to perform its testing, we have some testing samples with other unique features.
- Classification is an example of supervised learning.

Neural Network Learning Rules…

- The learning rule is a technique or a mathematical logic which encourages a neural network to gain from the existing condition and uplift its performance.
- It is an iterative procedure. During ANN learning, to change the input/output behavior, we need to adjust the weights.
- Hence, a method is required with the help of which the weights can be modified. **This method is known as a learning rule or learning process** that boosts the Artificial Neural Network's performance and implements this rule over the network.
- The learning rules refreshes the weights and bias levels of a network when a network mimics in a particular data environment.

Neural Network Learning Rules

- These methods or learning rules are **simply algorithms or equations**.
- Following are some learning rules for the neural network –



Offline or Online Learning…

- Learning can be **offline**
  - a set of training samples is presented, then the weights are changed, the total error is calculated by means of a error function operation or simply accumulated
- or **online**
  - after every sample presented the weights are changed.
- Both procedures have advantages and disadvantages.

Offline or Online Learning

- Offline training procedures are also called **batch training procedures** since a batch of results is corrected all at once.
- Such a training section of a whole batch of training samples including the related change in weight values is called epoch.

- **Definition**
  - **Offline learning:** Several training patterns are entered into the network at once, the errors are accumulated and it learns for all patterns at the same time.
  - **Online learning:** The network learns directly from the errors of each training sample.

Questions you should answer before learning

- The application of such schemes certainly requires preliminary thoughts about some questions, if possible, answer them before learning:
  - Where does the learning input come from and in what form?
  - How must the weights be modified to allow fast and reliable learning?
  - How can the success of a learning process be measured in an objective way?
  - Is it possible to determine the "best" learning procedure?
  - Is it possible to predict if a learning procedure terminates, i.e. whether it will reach an optimal state after a finite time or if it, for example, will oscillate between different states?
  - How can the learned patterns be stored in the network?
  - Is it possible to avoid that newly learned patterns destroy previously learned associations (the so-called stability/plasticity dilemma)?
- **All these questions cannot be generally answered but that they have to be discussed for each learning procedure and each network topology individually.**

Training Patterns…

- Before knowing the learning rules, we need to introduce the teaching input.
- In (this) case of supervised learning we assume a training set consisting of **training patterns** and the corresponding **correct output values** we want to see at the output neurons after the training.
- While the network has not finished training, i.e. as long as it is generating wrong outputs, these output values are referred to as **teaching input**, and that for each neuron individually.
- Thus, for a neuron *j* with the **incorrect output o_j , t_j is the teaching input**, which means it is the correct or desired output for a training pattern p.

Training Patterns…

- **Training Patterns:**
  - A training pattern is an input vector p with the components $p_1, p_2, \ldots, p_n$ whose desired output is known.
  - By entering the training pattern into the network we receive an output that can be compared with the teaching input, which is the desired output.
  - The set of training patterns is called P. It contains a finite number of ordered pairs (p, t) of **training patterns with corresponding desired output**.

Training Patterns…

- **Training patterns** are often simply called patterns, that is why they are referred to as p; they are called synonymously patterns, training samples etc.
- **Teaching input:**
  - Let j be an output neuron. The teaching input t_j is the desired and correct value j should output after the input of a certain training pattern.
  - Analogously to the vector p the teaching inputs $t_1, t_2, \ldots, t_n$ of the neurons can also be combined into a vector t.
  - t always refers to a specific training pattern p and is contained in the set P of the training patterns.

Training Patterns…

- It is useful to divide the set of training samples or the training set into-
  - one training set really used to train,
  - and one verification set to test.
- It is provided that there are enough training samples. The usual division relations are, for instance, **70% for training data and 30% for verification data (randomly chosen)**. We can finish the training when the network provides good results on the training data as well as on the verification data.

- **Note:** If the verification data provide poor results, do not modify the network structure until these data provide good results – otherwise you run the risk of tailoring the network to the verification data. This means that these data are included in the training, even if they are not used explicitly for the training. **The solution is a third set of validation data used only for validation after a successful training**

Training Patterns…

- **Error vector:** For several output neurons $\Omega_1, \Omega_2, \ldots, \Omega_n$ the difference between output vector and teaching input under a training input p is given by:

$$E_p = \begin{pmatrix} t_1 - y_1 \\ \vdots \\ t_n - y_n \end{pmatrix}$$

  is referred to as error vector, sometimes it is also called difference vector.

- Depending on whether you are learning offline or online, the difference vector refers to a specific training pattern, or to the error of a set of training patterns which is normalized in a certain way.

Training Patterns

- The vectors defined in ANN are summarized below:
  - **Input vector x**, which can be entered into the neural network. Depending on the type of network being used the neural network will output an output vector y.
  - Basically the **training sample p** is nothing more than an input vector. We only use it for training purposes because we know the corresponding **teaching input t** which is nothing more than **the desired output vector** to the training sample.
  - **The error vector Ep** is the difference between the teaching input t and the actual output y.

LEARNING AND ADAPTATION

To be continued…