Introduction to Neural Networks and Deep Learning

Learning Process

Andres Mendez-Vazquez

September 14, 2019

Outline

Cinvestav

Outline

Cinvestav

Learning

How do we define learning in a Neural Network?

The property that is of primary significance for a neural network is

- The ability of the network to learn from its environment.
- To improve its performance through learning.

Learning

How do we define learning in a Neural Network?

The property that is of primary significance for a neural network is

- The ability of the network to learn from its environment.
- To improve its performance through learning.

Thus, we use the following definition by Mendel and McClaren [1]

"Learning is a process by which the free parameters of neural network are adapted through a process of stimulation by the environment in which the network is embedded. The type of learning is determined by the manner in which the parameter changes take place."

Learning

How do we define learning in a Neural Network?

The property that is of primary significance for a neural network is

- The ability of the network to learn from its environment.
- To improve its performance through learning.

Thus, we use the following definition by Mendel and McClaren [1]

"Learning is a process by which the free parameters of neural network are adapted through a process of stimulation by the environment in which the network is embedded. The type of learning is determined by the manner in which the parameter changes take place."

Learning

How do we define learning in a Neural Network?

The property that is of primary significance for a neural network is

- The ability of the network to learn from its environment.
- To improve its performance through learning.

Thus, we use the following definition by Mendel and McClaren [1]

"Learning is a process by which the free parameters of neural network are adapted through a process of stimulation by the environment in which the network is embedded. The type of learning is determined by the manner in which the parameter changes take place."

Outline

Cinvestav

Implications of the Definition of Learning

First

The neural network is stimulated by an environment.

Second

The neural network undergoes changes in its free parameters as a result of this stimulation.

Third

The neural network responds in a new way to the environment because of the changes that have occurred in its internal structure.

Implications of the Definition of Learning

First

The neural network is stimulated by an environment.

Second

The neural network undergoes changes in its free parameters as a result of this stimulation.

Third

The neural network responds in a new way to the environment because of the changes that have occurred in its internal structure.

Implications of the Definition of Learning

First

The neural network is stimulated by an environment.

Second

The neural network undergoes changes in its free parameters as a result of this stimulation.

Third

The neural network responds in a new way to the environment because of the changes that have occurred in its internal structure.

Solution of Learning Problem $\approx$ Learning Algorithm

Quite Interesting

There is no unique learning algorithm for the design of neural networks.

What we have is more

A "kit of tools" represented by a diverse variety of learning algorithms:

- They depend on the type of architecture used in the Neural Network!!!

Cinvestav

Solution of Learning Problem $\approx$ Learning Algorithm

Quite Interesting

There is no unique learning algorithm for the design of neural networks.

What we have is more

A "kit of tools" represented by a diverse variety of learning algorithms:

- They depend on the type of architecture used in the Neural Network!!!

Outline

Cinvestav

Error Correcting Learning

Error Signal

We have an input $\boldsymbol{x}(t)$ (Here assume a time t) to a neuron k:

1. Desired response : $d_k(t)$.
2. Output signal : $y_k(t)$.

Error Correcting Learning

Error Signal

We have an input $\boldsymbol{x}(t)$ (Here assume a time t) to a neuron k:

1. Desired response : $d_k(t)$.
2. Output signal : $y_k(t)$.

Error Signal

$$e_k(t) = d_k(t) - y_k(t) \quad (1)$$

Error Correcting Learning

Error Signal

We have an input $\boldsymbol{x}(t)$ (Here assume a time t) to a neuron k:

1. Desired response : $d_k(t)$.
2. Output signal : $y_k(t)$.

Error Signal

$$e_k(t) = d_k(t) - y_k(t) \quad (1)$$

Error Correcting Learning

Error Signal

We have an input $\boldsymbol{x}(t)$ (Here assume a time t) to a neuron k:

1. Desired response : $d_k(t)$.
2. Output signal : $y_k(t)$.

Error Signal

$$e_k(t) = d_k(t) - y_k(t) \quad (1)$$

Cinvestav

Error?

This is used

- As a control mechanism.

The strategy to be followed

- It is s to apply a sequence of corrective adjustments to the synaptic weights of neuron k.

What do we want?

We want

$$\lim_{t \to \infty} y_k(t) = d_k$$

if we assume $d_k(t) = d_k$ a constant.

Cinvestav

Error?

This is used

- As a control mechanism.

The strategy to be followed

- It is s to apply a sequence of corrective adjustments to the synaptic weights of neuron k.

What do we want?

We want

$$\lim_{t \to \infty} y_k(t) = d_k$$

if we assume $d_k(t) = d_k$ a constant.

Error?

This is used

- As a control mechanism.

The strategy to be followed

- It is s to apply a sequence of corrective adjustments to the synaptic weights of neuron k.

What do we want?

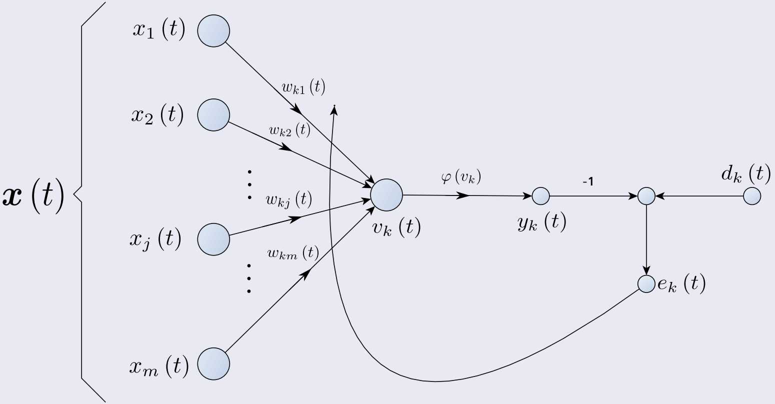
We want

$$\lim_{t \to \infty} y_k(t) = d_k$$

if we assume $d_k(t) = d_k$ a constant.

The Architecture

Graphical Representation



Outline

Cinvestav

How do we do this?

We need to use this error in some way

For this we use a well know convex function

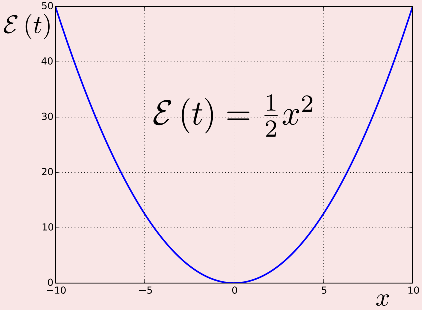
Quadratic function $\mathcal{E}(t) = \frac{1}{2}e^2$

How do we do this?

We need to use this error in some way

For this we use a well know convex function

Quadratic function $\mathcal{E}(t) = \frac{1}{2}x^2$



Error Correcting Learning

Cost Function with the error in it

$$\mathcal{E}(t) = \frac{1}{2} e_k^2(t) \tag{2}$$

We need to minimize this error

For this, we use the a learning rule called Delta Rule or Widrow-Hoff Rule [2]!!

Error Correcting Learning

Cost Function with the error in it

$$\mathcal{E}(t) = \frac{1}{2} e_k^2(t) \tag{2}$$

We need to minimize this error

For this, we use the a learning rule called Delta Rule or Widrow-Hoff Rule [2]!!

How this is done

We take derivative (Gradient) with respect to $w_{kj'}$ by imagining that we fix all the other values

$$\mathcal{E}(t) = \frac{1}{2}\left(d_k(t) - \sum_{j=0}^{m} w_{kj}(t)x_j(t)\right)^2$$

Assuming: $y_k(t) = \sum_{j=0}^{m} w_{kj}(t)x_j(t)$

Thus, we have

$$\mathcal{E}(t) = \frac{1}{2}\left(C_k(t) - w_{kj'}(t)x_{j'}(t)\right)^2$$

Where: $C_k(t) = d_k(t) - \sum_{\substack{j=1 \\ j \neq j'}}^{m} w_{kj}(t)x_j(t)$

How this is done

We take derivative (Gradient) with respect to $w_{kj'}$ by imagining that we fix all the other values

$$\mathcal{E}(t) = \frac{1}{2}\left(d_k(t) - \sum_{j=0}^{m} w_{kj}(t) x_j(t)\right)^2$$

Assuming: $y_k(t) = \sum_{j=0}^{m} w_{kj}(t) x_j(t)$

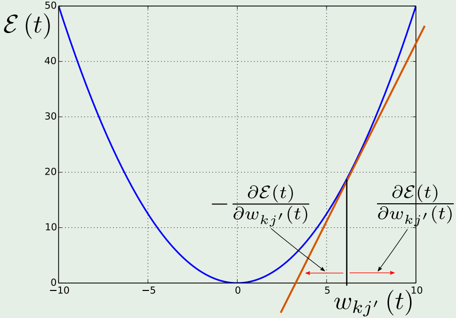
Thus, we have

$$\mathcal{E}(t) = \frac{1}{2}\left(C_k(t) - w_{kj'}(t) x_{j'}(t)\right)^2$$

Where: $C_k(t) = d_k(t) - \sum^{m}_{\substack{j=1 \\ j \neq j'}} w_{kj}(t) x_j(t)$

We have something like this

The Intuitive Idea



What information do we get from the Gradient?

We get the following information

The directions of greatest change in the axis of $w_{kj}(t)$

We use this to adjust the learning

For each weight element storing information for the neuron k.

Thus, we can state the delta rule [2]

"***The adjustment made to a synaptic weight of a neuron is proportional to the product of the error signal and the input signal of the synapse in question.***"

What information do we get from the Gradient?

We get the following information

The directions of greatest change in the axis of $w_{kj}(t)$

We use this to adjust the learning

For each weight element storing information for the neuron k.

Thus, we can state the delta rule [2]

"The adjustment made to a synaptic weight of a neuron is proportional to the product of the error signal and the input signal of the synapse in question."

What information do we get from the Gradient?

We get the following information

The directions of greatest change in the axis of $w_{kj}(t)$

We use this to adjust the learning

For each weight element storing information for the neuron k.

Thus, we can state the delta rule [2]

"***The adjustment made to a synaptic weight of a neuron is proportional to the product of the error signal and the input signal of the synapse in question.***"

This is simple

It comes from taking the gradient

$$\frac{\partial \mathcal{E}(t)}{\partial w_{kj'}} = \frac{\partial \frac{1}{2}\left(C_k(t) - w_{kj'}(t)x_{j'}(t)\right)^2}{\partial w_{kj'}}$$
$$= -\left(C_k(t) - w_{kj'}(t)x_{j'}(t)\right)x_{j'}(t)$$
$$= -e_k(t)\,x_{j'}(t)$$

Outline

Cinvestav

Thus, we get

Delta Rule or Widrow-Hoff Rule

$$\Delta w_{kj'}(t) = \eta e_k(t)\, x_{j'}(t) \tag{3}$$

With η absorbing the negative sign and representing the learning rate!!!

Actually this is know as Gradient Descent

$$w_{kj'}(t+1) = w_{kj'}(t) + \Delta w_{kj'}(t) \tag{4}$$

Thus, we get

Delta Rule or Widrow-Hoff Rule

$$\Delta w_{kj'}(t) = \eta e_k(t) x_{j'}(t) \quad (3)$$

With η absorbing the negative sign and representing the learning rate!!!

Actually this is know as Gradient Descent

$$w_{kj'}(t+1) = w_{kj'}(t) + \Delta w_{kj'}(t) \quad (4)$$

It is more

We have that

In effect, $w_{kj'}(t)$ and $w_{kj'}(t+1)$ may be viewed as the ***old*** and ***new*** values of synaptic weight $w_{kj'}$, respectively.

In computational method, we can also write that

$$w_{kj'}(t) = z^{-1}\left[w_{kj'}(t+1)\right],$$

where z^{-1} is the unit-delay operator. In other words, z^{-1} represents a storage element.

It is more

We have that

In effect, $w_{kj'}(t)$ and $w_{kj'}(t+1)$ may be viewed as the ***old*** and ***new*** values of synaptic weight $w_{kj'}$, respectively.

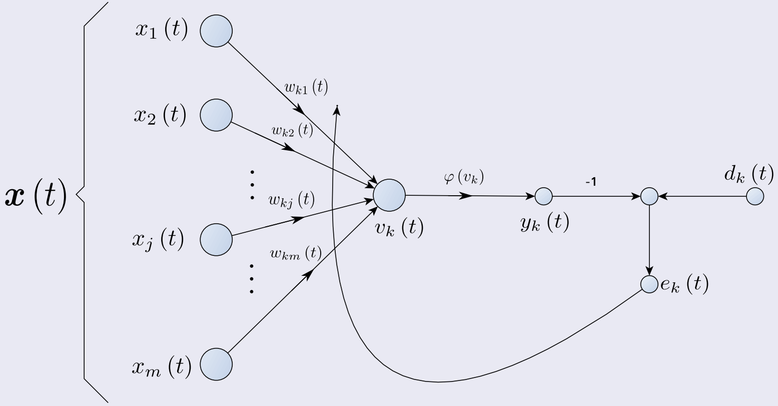
In computational method, we can also write that

$$w_{kj'}(t) = z^{-1}\left[w_{kj'}(t+1)\right],$$

where z^{-1} is the unit-delay operator. In other words, z^{-1} represents a storage element.

The Error-Correction Architecture

We have the graphical representation of the error-correction learning



This is an example of a closed-loop feedback system

Outline

Cinvestav

Outline

Cinvestav

What is this?

We store information

All (or most) of the past experiences are explicitly stored in a large memory of correctly classified input-output examples.

Formally

Input-Output examples : $\{(x_i, d_i)\}_{i=1}^{N}$

Where

- x_i denotes an input vector.
- d_i denotes the corresponding desired response.

What is this?

We store information

All (or most) of the past experiences are explicitly stored in a large memory of correctly classified input-output examples.

Formally

Input-Output examples : $\{(\boldsymbol{x}_i, d_i)\}_{i=1}^{N}$

Where

- x_i denotes an input vector.
- d_i denotes the corresponding desired response.

Cinvestav

What is this?

We store information

All (or most) of the past experiences are explicitly stored in a large memory of correctly classified input-output examples.

Formally

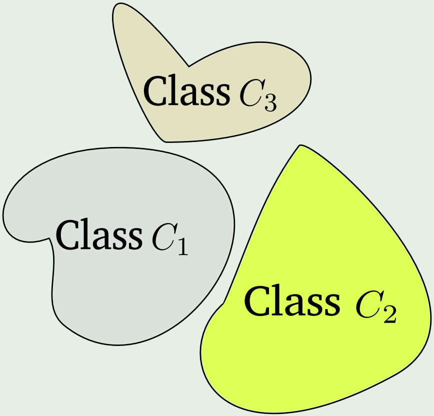
Input-Output examples : $\{(\boldsymbol{x}_i, d_i)\}_{i=1}^{N}$

Where

- $\boldsymbol{x}_i$ denotes an input vector.
- d_i denotes the corresponding desired response.

In addition, we split these samples into classes

Each of these inputs are in a Class C_j



Then

Given all this information

We can build an algorithm to classify not seen before samples $\boldsymbol{x}_{test}$.

This algorithm works as follow

The algorithm responds by retrieving and analyzing the training data in a "local neighborhood" of $\boldsymbol{x}_{test}$.

Then

Given all this information

We can build an algorithm to classify not seen before samples $\boldsymbol{x}_{test}$.

This algorithm works as follow

The algorithm responds by retrieving and analyzing the training data in a "local neighborhood" of $\boldsymbol{x}_{test}$.

Cinvestav

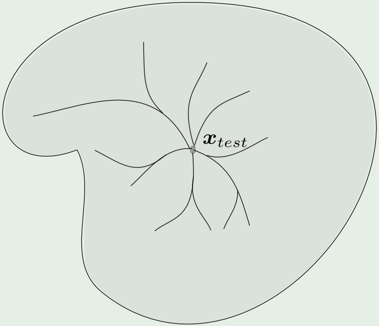
Outline

Cinvestav

Ingredients

First

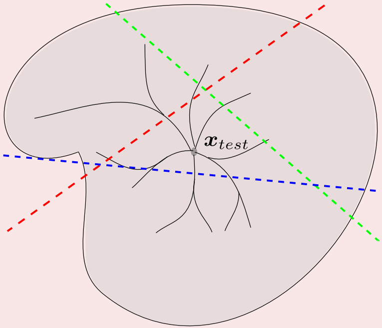
Criterion used for defining the local neighborhood of the test vector $\boldsymbol{x}_{test}$.



Ingredients

Second

Learning rule applied to the training examples in the local neighborhood of $\boldsymbol{x}_{test}$.



Remark

Something Notable

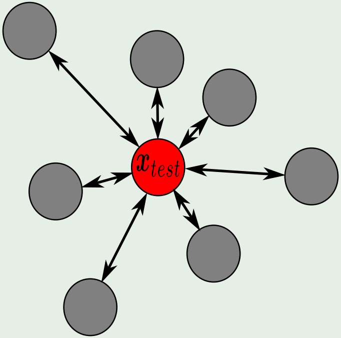
Algorithms are different between each other depending on how these two ingredients are defined

Outline

Example: K-Nearest Neighbor Classifier

First Step

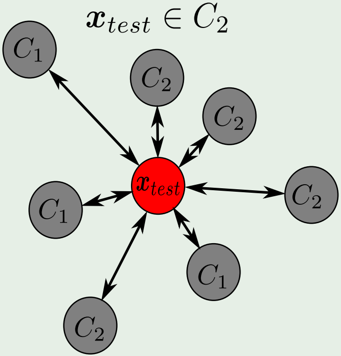
Identify the k classified patterns that lie nearest to the test vector $\boldsymbol{x}_{test}$ for some integer k.



Example: K-Nearest Neighbor Classifier

Second Step

Assign $\boldsymbol{x}_{test}$ to the class that is most frequently represented in the k nearest neighbors to $\boldsymbol{x}_{test}$.



Cover and Hart (1967) have studied the nearest neighborhood rule (Simplest Version of K-NN)

Their analysis (Chapter 3) is based in the following assumptions

- The classified examples $(\boldsymbol{x}_i, d)$ are independently and identically distributed (iid), according to the joint probability distribution of the example $(\boldsymbol{x}, d)$.
- The sample size N is infinitely large.

Cover and Hart (1967) have studied the nearest neighborhood rule (Simplest Version of K-NN)

Their analysis (Chapter 3) is based in the following assumptions

- The classified examples $(\boldsymbol{x}_i, d)$ are independently and identically distributed (iid), according to the joint probability distribution of the example $(\boldsymbol{x}, d)$.
- The sample size N is infinitely large.

Then it is possible to prove

- It is shown that the probability of classification error incurred by the nearest neighbor rule is bounded above by twice the Bayes probability of error.
- This is known as the minimum probability of error over all decision rules.

Cover and Hart (1967) have studied the nearest neighborhood rule (Simplest Version of K-NN)

Their analysis (Chapter 3) is based in the following assumptions

- The classified examples $(\boldsymbol{x}_i, d)$ are independently and identically distributed (iid), according to the joint probability distribution of the example $(\boldsymbol{x}, d)$.
- The sample size N is infinitely large.

Then it is possible to prove

- It is shown that the probability of classification error incurred by the nearest neighbor rule is bounded above by twice the Bayes probability of error.
- This is known as the minimum probability of error over all decision rules.

Cover and Hart (1967) have studied the nearest neighborhood rule (Simplest Version of K-NN)

Their analysis (Chapter 3) is based in the following assumptions

- The classified examples $(\boldsymbol{x}_i, d)$ are independently and identically distributed (iid), according to the joint probability distribution of the example $(\boldsymbol{x}, d)$.
- The sample size N is infinitely large.

Then it is possible to prove

- It is shown that the probability of classification error incurred by the nearest neighbor rule is bounded above by twice the Bayes probability of error.
- This is known as the minimum probability of error over all decision rules.

Outline

Cinvestav

Outline

Cinvestav

Hebb's postulate of learning

It is the oldest of all learning rules

In Hebb's "The Organization of Behavior (1949, p.62)"[3]:

- When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic changes take place in one or both cells such that A's efficiency as one of the cells firing B, is increased.

This can be rephrased as...

Something Notable

It is possible to expand this as a two-part rule...(Stent. 1973; Changeux and Danchin. 1976)

First

If two neurons on either side of a synapse (connection) are activated simultaneously (i.e., synchronously), then the strength of that synapse is selectively increased.

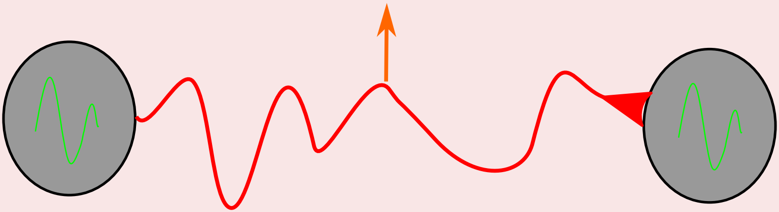
This can be rephrased as...

Something Notable

It is possible to expand this as a two-part rule...(Stent. 1973; Changeux and Danchin. 1976)

First

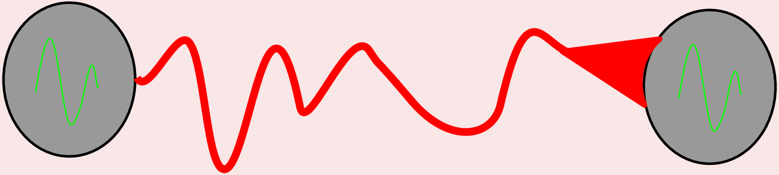
If two neurons on either side of a synapse (connection) are activated simultaneously (i.e., synchronously). then the strength of that synapse is selectively increased.



Thus

First

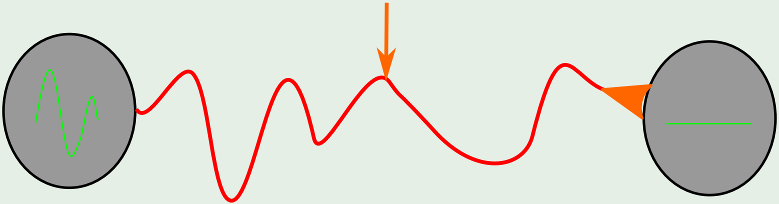
If two neurons on either side of a synapse (connection) are activated simultaneously (i.e., synchronously). then the strength of that synapse is selectively increased.



Then

Second

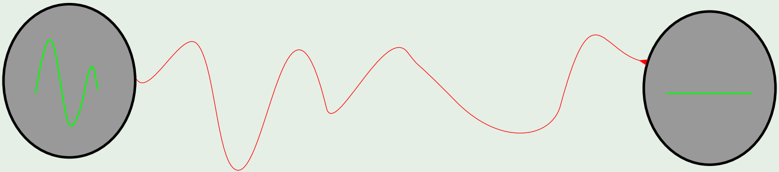
If two neurons on either side of a synapse are activated asynchronously, then that synapse is selectively weakened or eliminated.



Finally

Second

If two neurons on either side of a synapse are activated asynchronously, then that synapse is selectively weakened or eliminated.



Outline

Cinvestav

The Hebbian Synapse

Definition

A Hebbian synapse is a synapse that uses a:

1. Time-Dependant
2. Highly Local
3. Strongly Interactive

Mechanism to increase synaptic efficiency as a function of the correlation between the presynaptic and postsynaptic activities.

The Hebbian Synapse

Definition

A Hebbian synapse is a synapse that uses a:

1. Time-Dependant
2. Highly Local
3. Strongly Interactive

Mechanism to increase synaptic efficiency as a function of the correlation between the presynaptic and postsynaptic activities.

The Hebbian Synapse

Definition

A Hebbian synapse is a synapse that uses a:

1. Time-Dependant
2. Highly Local
3. Strongly Interactive

Mechanism to increase synaptic efficiency as a function of the correlation between the presynaptic and postsynaptic activities.

The Hebbian Synapse

Definition

A Hebbian synapse is a synapse that uses a:

1. Time-Dependant
2. Highly Local
3. Strongly Interactive

Mechanism to increase synaptic efficiency as a function of the correlation between the presynaptic and postsynaptic activities.

The Hebbian Synapse

Definition

A Hebbian synapse is a synapse that uses a:

1. Time-Dependant
2. Highly Local
3. Strongly Interactive

Mechanism to increase synaptic efficiency as a function of the correlation between the presynaptic and postsynaptic activities.

Key Mechanism of Hebbian Synapse

Time-dependent mechanism

It refers to the dependence of the synapse to the exact time of occurrence of the presynaptic and postsynaptic signals.

Local Mechanism

It refers to how the local information is used by the Hebbian synapse to make modifications to the synapse itself.

Interactive Mechanism

A Hebbian form of learning depends on a "true interaction" between presynaptic and postsynaptic signals which can be deterministic or statistical!!!

Cinvestav

Key Mechanism of Hebbian Synapse

Time-dependent mechanism

It refers to the dependence of the synapse to the exact time of occurrence of the presynaptic and postsynaptic signals.

Local Mechanism

It refers to how the local information is used by the Hebbian synapse to make modifications to the synapse itself.

Interactive Mechanism

A Hebbian form of learning depends on a "true interaction" between presynaptic and postsynaptic signals which can be deterministic or statistical!!!

Key Mechanism of Hebbian Synapse

Time-dependent mechanism

It refers to the dependence of the synapse to the exact time of occurrence of the presynaptic and postsynaptic signals.

Local Mechanism

It refers to how the local information is used by the Hebbian synapse to make modifications to the synapse itself.

Interactive Mechanism

A Hebbian form of learning depends on a "true interaction" between presynaptic and postsynaptic signals which can be deterministic or statistical!!!

Cinvestav

Key Mechanism of Hebbian Synapse

Conjunctional or correlational mechanism

- The co-occurrence of presynaptic and postsynaptic signals (within a short interval of time) is sufficient to produce the synaptic modification.
- It is for this reason that a Hebbian synapse is sometimes referred to as a conjunctional synapse.
- In addition, the correlation over time between presynaptic and postsynaptic signals is viewed as being responsible for a synaptic change.

Key Mechanism of Hebbian Synapse

Conjunctional or correlational mechanism

- The co-occurrence of presynaptic and postsynaptic signals (within a short interval of time) is sufficient to produce the synaptic modification.
- It is for this reason that a Hebbian synapse is sometimes referred to as a conjunctional synapse.
- In addition, the correlation over time between presynaptic and postsynaptic signals is viewed as being responsible for a synaptic change.

Key Mechanism of Hebbian Synapse

Conjunctional or correlational mechanism

- The co-occurrence of presynaptic and postsynaptic signals (within a short interval of time) is sufficient to produce the synaptic modification.
- It is for this reason that a Hebbian synapse is sometimes referred to as a conjunctional synapse.
- In addition, the correlation over time between presynaptic and postsynaptic signals is viewed as being responsible for a synaptic change.

Outline

Cinvestav

Initial Setup

Consider

- A synaptic weight w_{kj} of neuron k
- Presynaptic signal x_j
- Postsynaptical signal y_k

Initial Setup

Consider

- A synaptic weight w_{kj} of neuron k
- Presynaptic signal x_j
- Postsynaptical signal y_k

The general adjustment

$$\Delta w_{kj}(t) = F(y_k(t), x_j(t))$$

- The signals $x_j(t)$ and $y_k(t)$ are often treated dimensionless signals
- The previous formula admits many forms!!! We will look at two of them: Hebb's and Covariance hypothesis

Initial Setup

Consider

- A synaptic weight w_{kj} of neuron k
- Presynaptic signal x_j
- Postsynaptical signal y_k

The general adjustment

$$\Delta w_{kj}(t) = F(y_k(t), x_j(t))$$

Where

- F is a function of both postsynaptic and presynaptic signals.
- The signals $x_j(t)$ and $y_k(t)$ are often treated dimensionless signals.
- The previous formula admits many forms!!! We will look at two of them: **Hebb's and Covariance hypothesis**.

Initial Setup

Consider

- A synaptic weight w_{kj} of neuron k
- Presynaptic signal x_j
- Postsynaptical signal y_k

The general adjustment

$$\Delta w_{kj}(t) = F(y_k(t), x_j(t))$$

Where

- F is a function of both postsynaptic and presynaptic signals.
- The signals $x_j(t)$ and $y_k(t)$ are often treated dimensionless signals.
- The previous formula admits many forms!!! We will look at two of them: **Hebb's and Covariance hypothesis.**

Initial Setup

Consider

- A synaptic weight w_{kj} of neuron k
- Presynaptic signal x_j
- Postsynaptical signal y_k

The general adjustment

$$\Delta w_{kj}(t) = F\left(y_k(t), x_j(t)\right)$$

Where

- F is a function of both postsynaptic and presynaptic signals.
- The signals $x_j(t)$ and $y_k(t)$ are often treated dimensionless signals.
- The previous formula admits many forms!!! We will look at two of them: Hebb's and Covariance hypothesis.

Initial Setup

Consider

- A synaptic weight w_{kj} of neuron k
- Presynaptic signal x_j
- Postsynaptical signal y_k

The general adjustment

$$\Delta w_{kj}(t) = F(y_k(t), x_j(t))$$

Where

- F is a function of both postsynaptic and presynaptic signals.
- The signals $x_j(t)$ and $y_k(t)$ are often treated dimensionless signals.
- The previous formula admits many forms!!! We will look at two of them: Hebb's and Covariance hypothesis.

Initial Setup

Consider

- A synaptic weight w_{kj} of neuron k
- Presynaptic signal x_j
- Postsynaptical signal y_k

The general adjustment

$$\Delta w_{kj}(t) = F(y_k(t), x_j(t))$$

Where

- F is a function of both postsynaptic and presynaptic signals.
- The signals $x_j(t)$ and $y_k(t)$ are often treated dimensionless signals.
- The previous formula admits many forms!!! We will look at two of them: **Hebb's and Covariance hypothesis**.

Hebb's Hypothesis

The simplest form of Hebbian learning is described by

$$\Delta w_{kj}(t) = \eta y_k(t) x_j(t) \tag{5}$$

Where

- η is a positive constant that determines the learning rate.

Cinvestav

Hebb's Hypothesis

The simplest form of Hebbian learning is described by

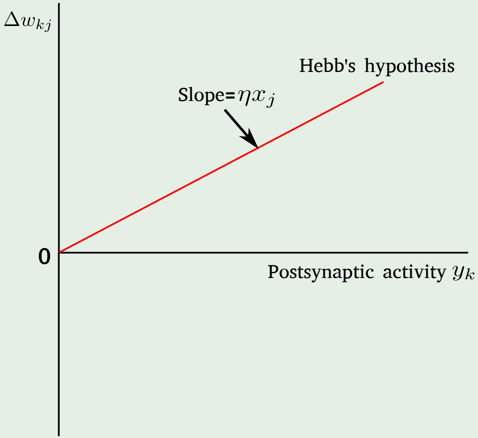
$$\Delta w_{kj}(t) = \eta y_k(t) x_j(t) \quad (5)$$

Where

- η is a positive constant that determines the learning rate.
- Clearly (Eq. 5) emphasizes the correlational nature of a Hebbian rate of synapse.
- It is referred as the **activity product rule**.

Graphical Representation

Δw_{kj} plotted versus the output signal (Quite similar to RELU)



Problem

Something Notable

The repeated application of the input signal (presynaptic activity) x_j leads to an increase in y_k

It is more

It leads to exponential growth that finally drives the synaptic connection into saturation!!!

Thus

At that point no information will be stored in the synapse and selectivity is lost.

Problem

Something Notable

The repeated application of the input signal (presynaptic activity) x_j leads to an increase in y_k

It is more

It leads to exponential growth that finally drives the synaptic connection into saturation!!!

Thus

At that point no information will be stored in the synapse and selectivity is lost.

Problem

Something Notable

The repeated application of the input signal (presynaptic activity) x_j leads to an increase in y_k

It is more

It leads to exponential growth that finally drives the synaptic connection into saturation!!!

Thus

At that point no information will be stored in the synapse and selectivity is lost.

Cinvestav

To solve this, we have...

Covariance Hypothesis (Sejnowski, 1974) [4]

In this hypothesis, the presynaptic and postsynaptic signals are replaced by the departure of presynaptic and postsynaptic signals from their respective average values over a certain time interval.Xi

Given the values $\overline{x}$ and $\overline{y}$ with respect to time

$$\Delta w_{kj} = \eta \left(x_j - \overline{x} \right) \left(y_k - \overline{y} \right) \quad (6)$$

Cinvestav

To solve this, we have...

Covariance Hypothesis (Sejnowski, 1974) [4]

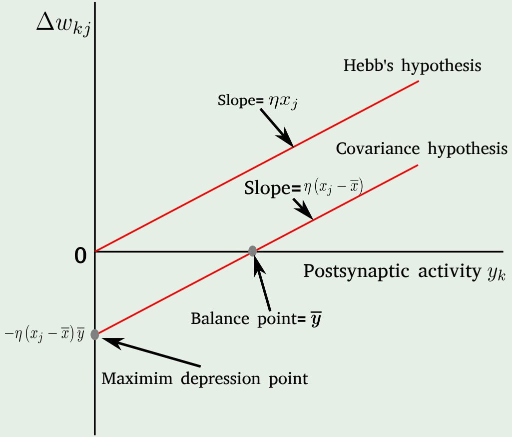
In this hypothesis, the presynaptic and postsynaptic signals are replaced by the departure of presynaptic and postsynaptic signals from their respective average values over a certain time interval.Xi

Given the values $\overline{x}$ and $\overline{y}$ with respect to time

$$\Delta w_{kj} = \eta \left(x_j - \overline{x}\right)\left(y_k - \overline{y}\right) \tag{6}$$

Graphical Representation

Δw_{kj} plotted versus the output signal



Properties of the Covariance Hypothesis

The covariance hypothesis allows for the following

- Convergence to a nontrivial state, which is reached when $x_k = \overline{x}$ and $y_j = \overline{y}$.
- Prediction of both synaptic potentiation (i.e., increase in synaptic strength) and synaptic depression (i.e., decrease in synaptic strength).

Properties of the Covariance Hypothesis

The covariance hypothesis allows for the following

- Convergence to a nontrivial state, which is reached when $x_k = \overline{x}$ and $y_j = \overline{y}$.
- Prediction of both synaptic potentiation (i.e., increase in synaptic strength) and synaptic depression (i.e., decrease in synaptic strength).

In addition, we have the following observations from (Eq. 6)

- Synaptic weight w_{kj} is enhanced if
  - $x_j > \overline{x}$ i.e. enough presynaptic input
  - $y_j > \overline{y}$ i.e. enough postynaptic output

Properties of the Covariance Hypothesis

The covariance hypothesis allows for the following

- Convergence to a nontrivial state, which is reached when $x_k = \overline{x}$ and $y_j = \overline{y}$.
- Prediction of both synaptic potentiation (i.e., increase in synaptic strength) and synaptic depression (i.e., decrease in synaptic strength).

In addition, we have the following observations from (Eq. 6)

- Synaptic weight w_{kj} is enhanced if
  - $x_j > \overline{x}$ i.e. enough presynaptic input
  - $y_j > \overline{y}$ i.e. enough postsynaptic output

Properties of the Covariance Hypothesis

The covariance hypothesis allows for the following

- Convergence to a nontrivial state, which is reached when $x_k = \overline{x}$ and $y_j = \overline{y}$.
- Prediction of both synaptic potentiation (i.e., increase in synaptic strength) and synaptic depression (i.e., decrease in synaptic strength).

In addition, we have the following observations from (Eq. 6)

- Synaptic weight w_{kj} is enhanced if
  - $x_j > \overline{x}$ i.e. enough presynaptic input
  - $y_j > \overline{y}$ i.e. enough postynaptic output

Cinvestav

Properties of the Covariance Hypothesis

In addition, we have the following observations from (Eq. 6)

- Synaptic weight w_{kj} is depressed if there is either
  - A presynaptic activation ($x_j > \overline{x}$) in the absence of sufficient postsynaptic activation ($y_j > \overline{y}$).
  - A postsynaptic activation ($y_k > \overline{y}$) in the absence of sufficient presynaptic activation ($x_j < \overline{x}$).

Properties of the Covariance Hypothesis

In addition, we have the following observations from (Eq. 6)

- Synaptic weight w_{kj} is depressed if there is either
  - A presynaptic activation $(x_j > \overline{x})$ in the absence of sufficient postsynaptic activation $(y_j > \overline{y})$.
  - A postsynaptic activation $(y_k > \overline{y})$ in the absence of sufficient presynaptic activation $(x_j < \overline{x})$.

Properties of the Covariance Hypothesis

In addition, we have the following observations from (Eq. 6)

- Synaptic weight w_{kj} is depressed if there is either
  - A presynaptic activation $(x_j > \overline{x})$ in the absence of sufficient postsynaptic activation $(y_j > \overline{y})$.
  - A postsynaptic activation $(y_k > \overline{y})$ in the absence of sufficient presynaptic activation $(x_j < \overline{x})$.

Remarks

First

This behavior may be regarded as a form of temporal competition between the incoming patterns.

Second

There is strong physiological evidence for Hebbian learning in the area of the brain called the hippocampus.

Remarks

First

This behavior may be regarded as a form of temporal competition between the incoming patterns.

Second

There is strong physiological evidence for Hebbian learning in the area of the brain called the hippocampus.

Outline

Cinvestav

The meaning

Intuition

In competitive learning the output neurons of a neural network compete among themselves to become active (fired).

Not only that

In Hebbian learning several output neurons may be active simultaneously, but in competitive learning only a single output neuron is active at any one time!!!

Important

This makes competitive learning highly suited to discover statistically salient features that may be used to classify a set of input patterns.

The meaning

Intuition

In competitive learning the output neurons of a neural network compete among themselves to become active (fired).

Not only that

In Hebbian learning several output neurons may be active simultaneously, but in competitive learning only a single output neuron is active at any one time!!!

Important

This makes competitive learning highly suited to discover statistically salient features that may be used to classify a set of input patterns.

The meaning

Intuition

In competitive learning the output neurons of a neural network compete among themselves to become active (fired).

Not only that

In Hebbian learning several output neurons may be active simultaneously, but in competitive learning only a single output neuron is active at any one time!!!

Important

This makes competitive learning highly suited to discover statistically salient features that may be used to classify a set of input patterns.

Outline

Cinvestav

Basic Elements (Rumelhart and Zipser, 1985) [5]

First

A set of neurons that are all the same except for some randomly distributed synaptic weights, and which therefore respond differently to a given set of input patterns.

Second

A limit imposed on the "strength" of each neuron.

- The neuron that wins the competition is called a winner-take-all neuron.

Basic Elements (Rumelhart and Zipser, 1985) [5]

First

A set of neurons that are all the same except for some randomly distributed synaptic weights, and which therefore respond differently to a given set of input patterns.

Second

A limit imposed on the "strength" of each neuron.

Third

- A mechanism that permits the neurons to compete for the right to respond to a given subset of inputs, such that only one output neuron is active (i.e. "ON").
- The neuron that wins the competition is called a winner-takes-all neuron.

Basic Elements (Rumelhart and Zipser, 1985) [5]

First

A set of neurons that are all the same except for some randomly distributed synaptic weights, and which therefore respond differently to a given set of input patterns.

Second

A limit imposed on the "strength" of each neuron.

Third

- A mechanism that permits the neurons to compete for the right to respond to a given subset of inputs, such that only one output neuron is active (i.e. “ON”).
- The neuron that wins the competition is called a winner-takes-all neuron.

Basic Elements (Rumelhart and Zipser, 1985) [5]

First

A set of neurons that are all the same except for some randomly distributed synaptic weights, and which therefore respond differently to a given set of input patterns.

Second

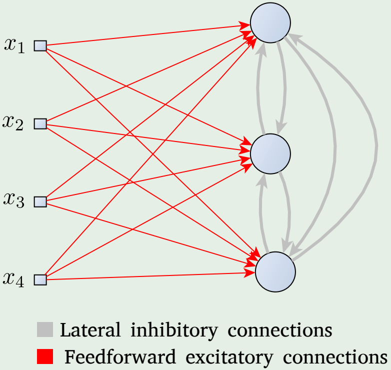
A limit imposed on the "strength" of each neuron.

Third

- A mechanism that permits the neurons to compete for the right to respond to a given subset of inputs, such that only one output neuron is active (i.e. “ON”).
- The neuron that wins the competition is called a winner-takes-all neuron.

Simple Example

Architecture of a competitive learning network



Mathematical Model

We want the following

For a neuron k to be the winning neuron, its induced local field v_k, for a specified input pattern x must be the largest among all the neurons in the network.

Thus, we have

$$y_k = \begin{cases} 1 & \text{if } v_k > v_j \ \forall j, j \neq k \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

With

v_k represents the combined action of all the forward and feedback inputs to neuron k.

Mathematical Model

We want the following

For a neuron k to be the winning neuron, its induced local field v_k, for a specified input pattern x must be the largest among all the neurons in the network.

Thus, we have

$$y_k = \begin{cases} 1 & \text{if } v_k > v_j \ \forall j, j \neq k \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

With

v_k represents the combined action of all the forward and feedback inputs to neuron k.

Mathematical Model

We want the following

For a neuron k to be the winning neuron, its induced local field v_k, for a specified input pattern x must be the largest among all the neurons in the network.

Thus, we have

$$y_k = \begin{cases} 1 & \text{if } v_k > v_j \ \forall j, j \neq k \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

With

v_k represents the combined action of all the forward and feedback inputs to neuron k.

Cinvestav

In addition

We want each neuron to have a fixed amount of synaptic weight

$$\sum_j w_{kj} = 1 \quad (8)$$

Thus

A neuron learns by shifting synaptic weights from inactive to active inputs.

Then, we have the following competitive learning rule

$$\Delta w_{kj} = \begin{cases} \eta \left(x_j - w_{kj} \right) & \text{if neuron } k \text{ wins the competition} \\ 0 & \text{if neuron } k \text{ loses the competition} \end{cases} \quad (9)$$

In addition

We want each neuron to have a fixed amount of synaptic weight

$$\sum_{j} w_{kj} = 1 \tag{8}$$

Thus

A neuron learns by shifting synaptic weights from inactive to active inputs.

Then, we have the following competitive learning rule

$$\Delta w_{kj} = \begin{cases} \eta \left(x_j - w_{kj} \right) & \text{if neuron } k \text{ wins the competition} \\ 0 & \text{if neuron } k \text{ loses the competition} \end{cases} \tag{9}$$

In addition

We want each neuron to have a fixed amount of synaptic weight

$$\sum_j w_{kj} = 1 \tag{8}$$

Thus

A neuron learns by shifting synaptic weights from inactive to active inputs.

Then, we have the following competitive learning rule

$$\Delta w_{kj} = \begin{cases} \eta \left(x_j - w_{kj} \right) & \text{if neuron } k \text{ wins the competition} \\ 0 & \text{if neuron } k \text{ loses the competition} \end{cases} \tag{9}$$

Then

What is going to happened?

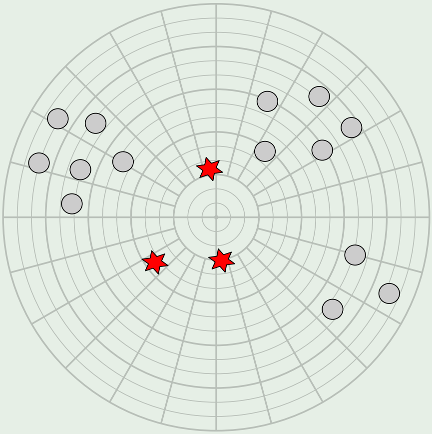
$$\boldsymbol{w}_k \longrightarrow \boldsymbol{x} \tag{10}$$

i.e. moving the weight in neuron k toward the pattern $\boldsymbol{x}$.

Example

We have then (Given $\sum_j w_{kj} = 1$)

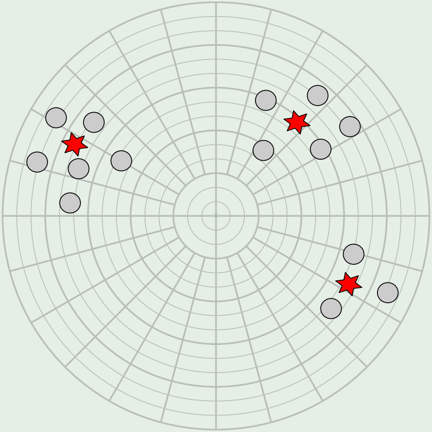
Here, the gray dots are the patterns and the red stars are the state of the networks!!!



Example

We have then

Here, the gray dots are the patterns and the red stars are the state of the networks!!!



Meaning

It represents

The ability of a neural network to perform **clustering** through competitive learning.

However

The competitive learning is "stable" only if the patterns have well defined groups.

Otherwise

The network may be unstable!!!

Meaning

It represents

The ability of a neural network to perform **clustering** through competitive learning.

However

The competitive learning is "stable" only if the patterns have well defined groups.

Otherwise

The network may be unstable!!!

Cinvestav

Meaning

It represents

The ability of a neural network to perform **clustering** through competitive learning.

However

The competitive learning is “stable” only if the patterns have well defined groups.

Otherwise

The network may be unstable!!!

Outline

Cinvestav

Boltzmann Learning

Ludwig Boltzmann

- Ludwig Eduard Boltzmann was an Austrian physicist and philosopher (1844-1906).
- Greatest achievement was in the development of statistical mechanics, which explains and predicts how the properties of atoms determine the physical properties of matter.

Cinvestav

Boltzmann Learning

Ludwig Boltzmann

- Ludwig Eduard Boltzmann was an Austrian physicist and philosopher (1844-1906).
- Greatest achievement was in the development of statistical mechanics, which explains and predicts how the properties of atoms determine the physical properties of matter.

This learning

It is based in using a stochastic learning algorithm.

Boltzmann Learning

Ludwig Boltzmann

- Ludwig Eduard Boltzmann was an Austrian physicist and philosopher (1844-1906).
- Greatest achievement was in the development of statistical mechanics, which explains and predicts how the properties of atoms determine the physical properties of matter.

This learning

It is based in using a stochastic learning algorithm.

Neural Networks

Neural Networks based on this learning are called Boltzmann machines (Ackley et a\., 1985; Hinton and Sejnowski, 1986).

Boltzmann Learning

Ludwig Boltzmann

- Ludwig Eduard Boltzmann was an Austrian physicist and philosopher (1844-1906).
- Greatest achievement was in the development of statistical mechanics, which explains and predicts how the properties of atoms determine the physical properties of matter.

This learning

It is based in using a stochastic learning algorithm.

Neural Networks

Neural Networks based on this learning are called Boltzmann machines (Ackley et a\., 1985; Hinton and Sejnowski, 1986).

Boltzmann Machines

Here

The neurons constitute a recurrent structure operating in a binary manner!!!

Boltzmann Machines

Here

The neurons constitute a recurrent structure operating in a binary manner!!!

Binary Manner?

- They are in an ON state, denoted $+1$.
- Or they are in an OFF state, denoted -1.

Boltzmann Machines

Here

The neurons constitute a recurrent structure operating in a binary manner!!!

Binary Manner?

- They are in an ON state, denoted $+1$.
- Or they are in an OFF state, denoted -1.

Cinvestav

Boltzmann Machines

The machine is characterized by an energy function

$$E = -\frac{1}{2}\sum_{\substack{j \\ j\neq k}}\sum_{k} w_{kj}x_k x_j \tag{11}$$

Boltzmann Machines

The machine is characterized by an energy function

$$E = -\frac{1}{2}\sum_{\substack{j \\ j \neq k}}\sum_{k} w_{kj} x_k x_j \tag{11}$$

Here

- The x_j is the state of neuron j.
- w_{kj} is the synaptic weight connecting neuron j to neuron k.
- The fact that $j \neq k$ means simply that none of the neurons in the machine has a self-feedback.

Boltzmann Machines

The machine is characterized by an energy function

$$E = -\frac{1}{2}\sum_{\substack{j \\ j \neq k}}\sum_{k} w_{kj} x_k x_j \tag{11}$$

Here

- The x_j is the state of neuron j.
- w_{kj} is the synaptic weight connecting neuron j to neuron k.
- The fact that $j \neq k$ means simply that none of the neurons in the machine has a self-feedback.

Boltzmann Machines

The machine is characterized by an energy function

$$E = -\frac{1}{2}\sum_{\substack{j \\ j\neq k}}\sum_{k} w_{kj}x_k x_j \tag{11}$$

Here

- The x_j is the state of neuron j.
- w_{kj} is the synaptic weight connecting neuron j to neuron k.
- The fact that $j \neq k$ means simply that none of the neurons in the machine has a self-feedback.

Cinvestav

Operation of a Boltzmann Machine

Take a neuron randomly

Then change the state of the neuron k from state x_k to state $-x_k$ at some temperature T with probability

$$P\left(x_k \longrightarrow -x_k\right) = \frac{1}{1 + \exp\left\{-\frac{\Delta E_k}{T}\right\}} \tag{12}$$

Operation of a Boltzmann Machine

Take a neuron randomly

Then change the state of the neuron k from state x_k to state $-x_k$ at some temperature T with probability

$$P\left(x_k \longrightarrow -x_k\right) = \frac{1}{1 + \exp\left\{-\frac{\Delta E_k}{T}\right\}} \tag{12}$$

Where

- ΔE_k is the energy change inside of the machine given the flip.
- T is a pseudotemperature - See chapter 1.

Operation of a Boltzmann Machine

Take a neuron randomly

Then change the state of the neuron k from state x_k to state $-x_k$ at some temperature T with probability

$$P\left(x_k \longrightarrow -x_k\right) = \frac{1}{1 + \exp\left\{-\frac{\Delta E_k}{T}\right\}} \tag{12}$$

Where

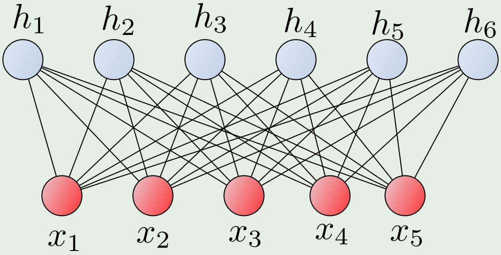
- ΔE_k is the energy change inside of the machine given the flip.
- T is a pseudotemperature - See chapter 1.

Cinvestav

Architecture for the Boltzmann Machines

We have

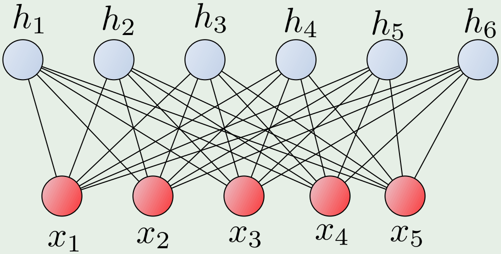
The neurons of a Boltzmann machine partition into two functional groups: visible and hidden.



Architecture for the Boltzmann Machines

We have

The neurons of a Boltzmann machine partition into two functional groups: visible and hidden.



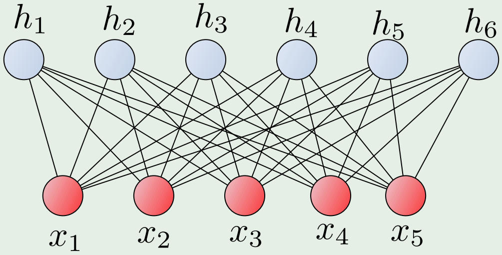
Thus

- The visible neurons provide an interface between the network and the environment in which it operates.
- The hidden neurons always operate freely

Architecture for the Boltzmann Machines

We have

The neurons of a Boltzmann machine partition into two functional groups: visible and hidden.



Thus

- The visible neurons provide an interface between the network and the environment in which it operates.
- The hidden neurons always operate freely

Modes of Operation

Clamped condition

The visible neurons are all clamped onto specific states determined by the environment.

Free-running condition

All the neurons (visible and hidden) are allowed to operate freely.

- ρ_{kj} denotes the correlation between the states of neurons j and k under free running.

Modes of Operation

Clamped condition

The visible neurons are all clamped onto specific states determined by the environment.

Free-running condition

All the neurons (visible and hidden) are allowed to operate freely.

Now

- ρ_{kj}^{+} denotes the correlation between the states of neurons j and k under clamping.
- ρ_{kj}^{-} denotes the correlation between the states of neurons j and k under free running.

Modes of Operation

Clamped condition

The visible neurons are all clamped onto specific states determined by the environment.

Free-running condition

All the neurons (visible and hidden) are allowed to operate freely.

Now

- ρ_{kj}^{+} denotes the correlation between the states of neurons j and k under clamping.
- ρ_{kj}^{-} denotes the correlation between the states of neurons j and k under free running.

A little taste

Here is one definition

$$
\begin{aligned}
\rho_{kj}^{+} &= \langle x_k x_j \rangle^{+} \\
&= \sum_{\boldsymbol{x}_\alpha \in T} \sum_{\boldsymbol{x}_\beta} P\left(\boldsymbol{X}_\beta = \boldsymbol{x}_\beta | \boldsymbol{X}_\alpha = \boldsymbol{x}_\alpha\right) x_k x_j
\end{aligned}
$$

- $\boldsymbol{x}_\alpha$ training inputs from environment.
- $\boldsymbol{x}_\beta$ hidden responses.

Modes of Operation

Important

Both correlations are averaged over all possible states of the machine when it is in thermal equilibrium.

Boltzmann learning rule

The change Δw_{kj} applied to synaptic weight w_{kj} from neuron j to neuron k (Hinton and Sejnowski, 1986):

$$\Delta w_{kj} = \eta \left(\rho_{kj}^{+} - \rho_{kj}^{-} \right) \quad (13)$$

More about this

Chapter 11 Haykin's Book.

Cinvestav

Modes of Operation

Important

Both correlations are averaged over all possible states of the machine when it is in thermal equilibrium.

Boltzmann learning rule

The change Δw_{kj} applied to synaptic weight w_{kj} from neuron j to neuron k (Hinton and Sejnowski, 1986):

$$\Delta w_{kj} = \eta \left(\rho_{kj}^{+} - \rho_{kj}^{-} \right) \quad (13)$$

More about this

Chapter 11 Haykin's Book.

Modes of Operation

Important

Both correlations are averaged over all possible states of the machine when it is in thermal equilibrium.

Boltzmann learning rule

The change Δw_{kj} applied to synaptic weight w_{kj} from neuron j to neuron k (Hinton and Sejnowski, 1986):

$$\Delta w_{kj} = \eta \left(\rho_{kj}^{+} - \rho_{kj}^{-} \right) \quad (13)$$

More about this

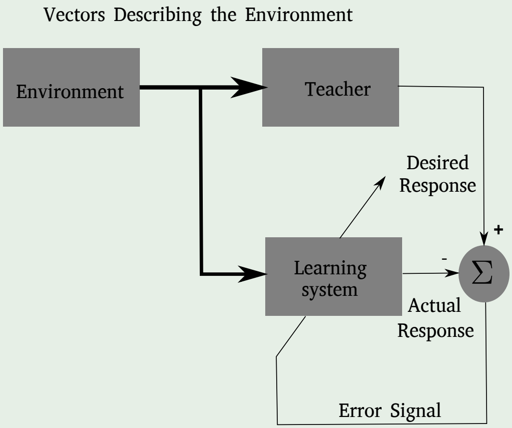
Chapter 11 Haykin's Book.

Outline

Cinvestav

A Graphical View

Error Correcting Version



Outline

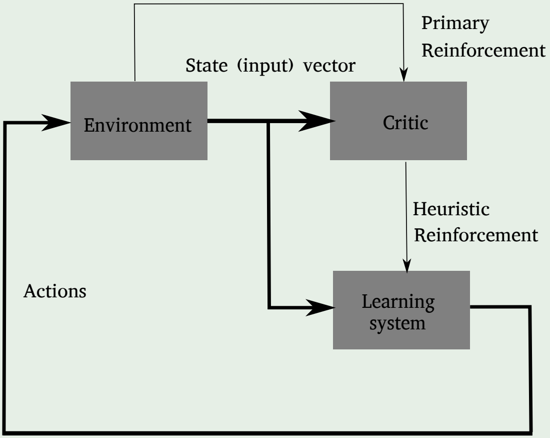
Cinvestav

Outline

Cinvestav

Reinforcement learning/Neurodynamic programming

Barto et al., 1983



Remark: Reinforcement learning is closely related to dynamic programming, which was developed by Bellman (1957) in the context of optimal control theory.

Observations on Reinforcement Learning

First

We want to minimize a cost-to-go function defined as the expectation of the cumulative cost of actions taken over a sequence of steps.

Second

The system is designed to learn under delayed reinforcement.

Delayed Reinforcement

The system observes a temporal sequence of stimuli (i.e., state vectors) also received from the environment, which eventually result in the generation of the heuristic reinforcement signal.

Observations on Reinforcement Learning

First

We want to minimize a cost-to-go function defined as the expectation of the cumulative cost of actions taken over a sequence of steps.

Second

The system is designed to learn under delayed reinforcement.

Delayed Reinforcement

The system observes a temporal sequence of stimuli (i.e., state vectors) also received from the environment, which eventually result in the generation of the heuristic reinforcement signal.

Observations on Reinforcement Learning

First

We want to minimize a cost-to-go function defined as the expectation of the cumulative cost of actions taken over a sequence of steps.

Second

The system is designed to learn under delayed reinforcement.

Delayed Reinforcement

The system observes a temporal sequence of stimuli (i.e., state vectors) also received from the environment, which eventually result in the generation of the heuristic reinforcement signal.

Problems with Delayed-Reinforcement Learning

It is difficult to perform

- There is no teacher to provide a desired response at each step of the learning process.
- The delay incurred in the generation of the primary reinforcement signal implies that the learning machine must use an expected next signal.

Cinvestav

Problems with Delayed-Reinforcement Learning

It is difficult to perform

- There is no teacher to provide a desired response at each step of the learning process.
- The delay incurred in the generation of the primary reinforcement signal implies that the learning machine must use an expected next signal.

Outline

Cinvestav

Unsupervised Learning

Here

In unsupervised or self-organized learning there is no external teacher or critic to oversee the learning process

Something Notable

Provision is made for a taskindependent measure of the quality of representation that the network is required to learn, and the free parameters of the network are optimized with respect to that measure

How is this done?

This develops the ability to form internal representations for encoding features of the input and thereby to create new classes automatically (Becker, 1991).

Unsupervised Learning

Here

In unsupervised or self-organized learning there is no external teacher or critic to oversee the learning process

Something Notable

Provision is made for a taskindependent measure of the quality of representation that the network is required to learn, and the free parameters of the network are optimized with respect to that measure

How is this done?

This develops the ability to form internal representations for encoding features of the input and thereby to create new classes automatically (Becker, 1991).

Unsupervised Learning

Here

In unsupervised or self-organized learning there is no external teacher or critic to oversee the learning process

Something Notable

Provision is made for a taskindependent measure of the quality of representation that the network is required to learn, and the free parameters of the network are optimized with respect to that measure

How is this done?

This develops the ability to form internal representations for encoding features of the input and thereby to create new classes automatically (Becker, 1991).

Outline

Cinvestav

Learning Tasks

Pattern Association

- An associative memory is a brainlike distributed memory that learns by association.

$$\boldsymbol{x_k} \underset{\boldsymbol{Associate}}{\longrightarrow} \boldsymbol{y}_k, k = 1, 2, ..., q$$

Learning Tasks

Pattern Recognition

- It is is formally defined as the process whereby a received pattern/signal is assigned to one ofa prescribed number of classes (categories).

Learning Tasks

Function Approximation

- Consider an input-output mapping $d = f(\boldsymbol{x})$
- The requirement is to design a neural network that approximates the unknown function $f(\boldsymbol{x})$ using a function $F(\boldsymbol{x})$

$$\|F(\boldsymbol{x}) - f(\boldsymbol{x})\| < \epsilon, \forall \boldsymbol{x}$$

Learning Tasks

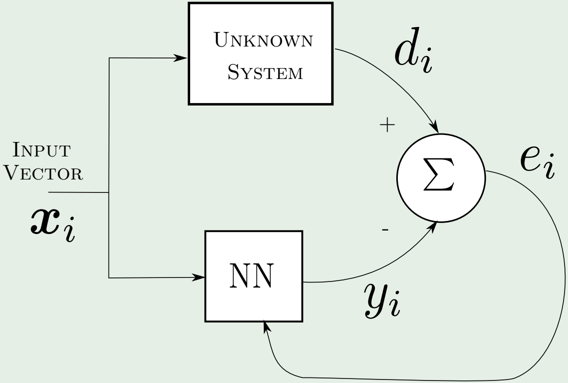
Function Approximation

- Consider an input-output mapping $d = f(\boldsymbol{x})$
- The requirement is to design a neural network that approximates the unknown function $f(\boldsymbol{x})$ using a function $F(\boldsymbol{x})$

$$\|F(\boldsymbol{x}) - f(\boldsymbol{x})\| < \epsilon, \forall \boldsymbol{x}$$

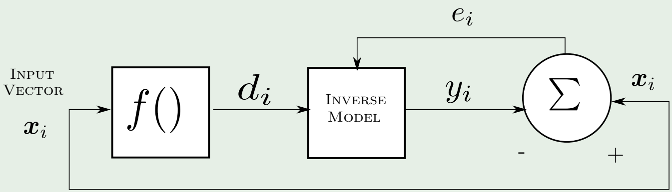
Function approximation can be used

System Identification



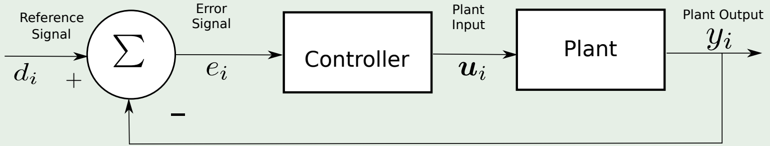
Function approximation can be used

Inverse system



In addition

Neural Networks can be used for Control



Cinvestav

Finally, Filtering

We may use a filter to perform three basic information processing tasks

1. **Filtering.** This task refers to the extraction of information about a quantity of interest at discrete time n by using data measured up to and including time n.
2. **Smoothing.** This second task differs from filtering in that information about the quantity of interest need not be available at time n, and data measured later than time n can be used in obtaining this information.
3. **Prediction.** This task is the forecasting side of information processing.

Cinvestav

Finally, Filtering

We may use a filter to perform three basic information processing tasks

1. **Filtering.** This task refers to the extraction of information about a quantity of interest at discrete time n by using data measured up to and including time n.
2. **Smoothing.** This second task differs from filtering in that information about the quantity of interest need not be available at time n, and data measured later than time n can be used in obtaining this information.
3. **Prediction.** This task is the forecasting side of information processing.

Cinvestav

Finally, Filtering

We may use a filter to perform three basic information processing tasks

1. **Filtering.** This task refers to the extraction of information about a quantity of interest at discrete time n by using data measured up to and including time n.
2. **Smoothing.** This second task differs from filtering in that information about the quantity of interest need not be available at time n, and data measured later than time n can be used in obtaining this information.
3. **Prediction.** This task is the forecasting side of information processing.

- J. M. Mendel and R. McLaren, "8 reinforcement-learning control and pattern recognition systems," in *Mathematics in Science and Engineering*, vol. 66, pp. 287–318, Elsevier, 1970.
- S. Haykin, *Neural Networks and Learning Machines*.
No. v. 10 in Neural networks and learning machines, Prentice Hall, 2009.
- D. O. Hebb, *The organization of behavior: a neuropsychological theory*.
Science Editions, 1962.
- T. Sejnowski, "On the stochastic dynamics of neuronal interaction," *Biological cybernetics*, vol. 22, no. 4, pp. 203–211, 1976.
- D. E. Rumelhart and D. Zipser, "Feature discovery by competitive learning," *Cognitive science*, vol. 9, no. 1, pp. 75–112, 1985.

Cinvestav