



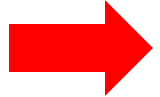
Connectionist Learning Procedures

Siamak Saliminejad

Overview

1. Introduction
2. Connectionist Models
3. Connectionist Research Issues
4. Associative Memories without Hidden Units
5. Simple Supervised Learning Procedures
6. Back propagation
7. Boltzman Machines
8. Maximizing Mutual Information
9. Unsupervised Hebbian Learning
10. Competitive Learning
11. Reinforcement Learning Procedures
12. Generalization

Overview



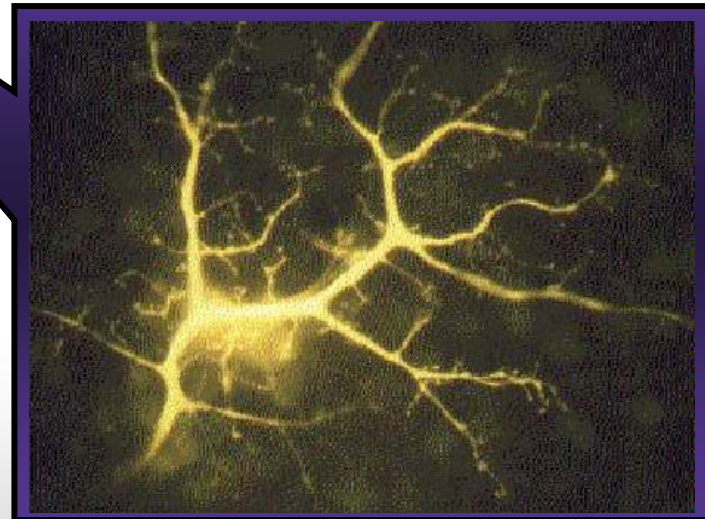
1. Introduction
2. Connectionist Models
3. Connectionist Research Issues
4. Associative Memories without Hidden Units
5. Simple Supervised Learning Procedures
6. Back propagation
7. Boltzman Machines
8. Maximizing Mutual Information
9. Unsupervised Hebbian Learning
10. Competitive Learning
11. Reinforcement Learning Procedures
12. Generalization

Introduction

How internal representations can be learned in "connectionist" networks??

First: Resemble the brain more closely than conventional methods

Second: Are massively parallel



Overview

1. Introduction

 2. Connectionist Models

3. Connectionist Research Issues

4. Associative Memories without Hidden Units

5. Simple Supervised Learning Procedures

6. Back propagation

7. Boltzman Machines

8. Maximizing Mutual Information

9. Unsupervised Hebbian Learning

10. Competitive Learning

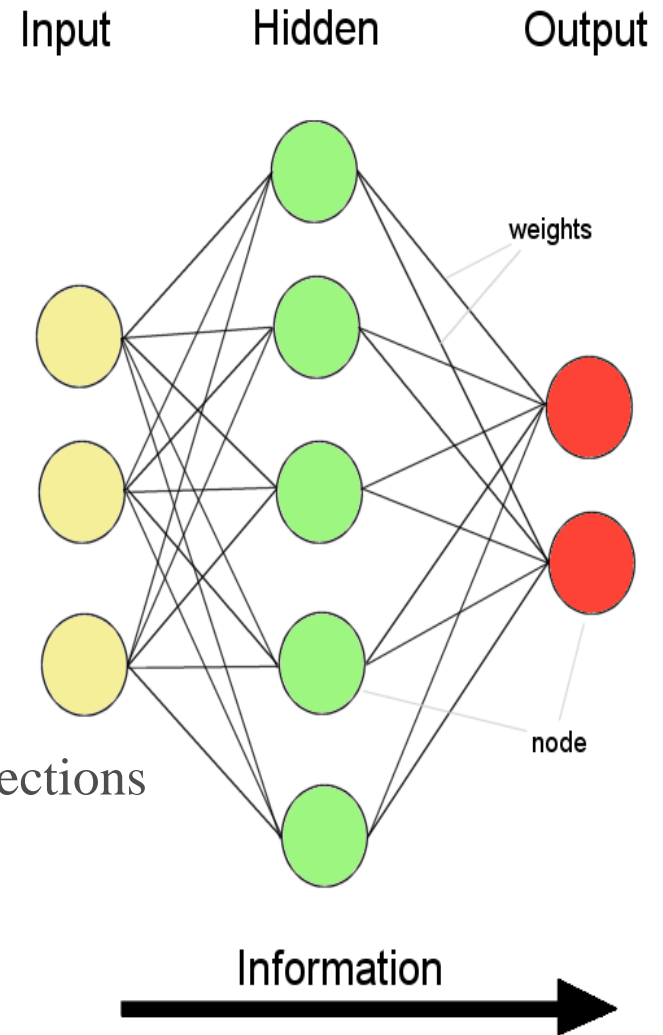
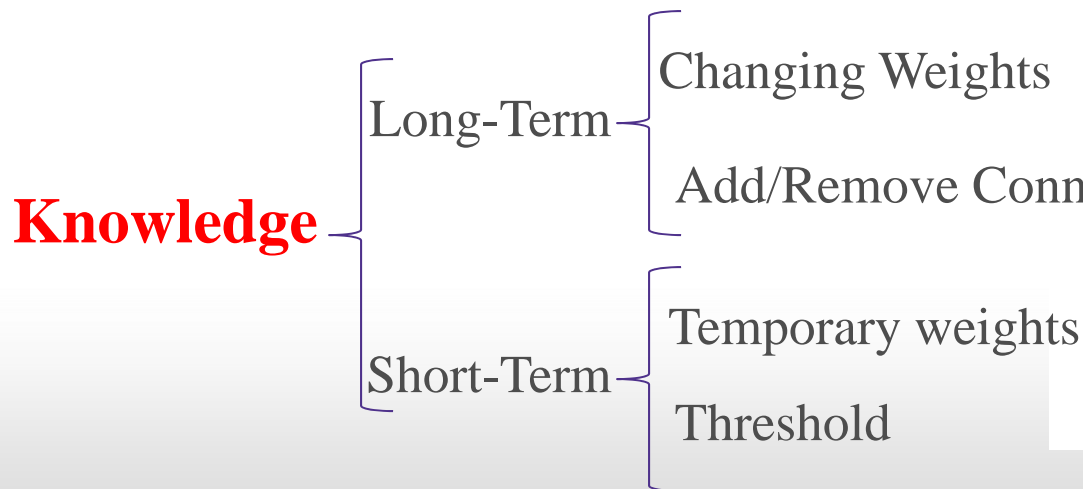
11. Reinforcement Learning Procedures

12. Generalization

Connectionist Models

Units: simple, neuron-like processing elements called "units" that interact using weighted connections

State (Activity Level): is determined by the input received from other units in the network



Overview

1. Introduction

2. Connectionist Models

 3. Connectionist Research Issues

4. Associative Memories without Hidden Units

5. Simple Supervised Learning Procedures

6. Back propagation

7. Boltzman Machines

8. Maximizing Mutual Information

9. Unsupervised Hebbian Learning

10. Competitive Learning

11. Reinforcement Learning Procedures

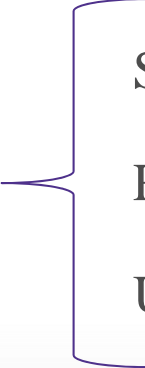
12. Generalization

Connectionist Research Issues

➤ **Search**

➤ **Representation** 

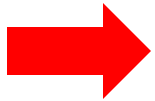
- Local
- Distributed

➤ **Learning** 

- Supervised
- Reinforcement
- Unsupervised

Overview

1. Introduction
2. Connectionist Models
3. Connectionist Research Issues
4. Associative Memories without Hidden Units
5. Simple Supervised Learning Procedures
6. Back propagation
7. Boltzman Machines
8. Maximizing Mutual Information
9. Unsupervised Hebbian Learning
10. Competitive Learning
11. Reinforcement Learning Procedures
12. Generalization



Associative Memories without Hidden Units

- No hidden unit
- The aim is simply to store a set of associations between input & output vectors by modifying the weights.

Linear ANs

The state of an output unit is a linear function of the total input that it receives from the input units

Perfect recall If the input vectors are orthogonal and have length 1

Nonlinear ANs

Associations which have nonorthogonal input vectors

Deficiencies

Most tasks are nonlinear and complex

Overview

1. Introduction
2. Connectionist Models
3. Connectionist Research Issues
4. Associative Memories without Hidden Units
5. Simple Supervised Learning Procedures
6. Back propagation
7. Boltzman Machines
8. Maximizing Mutual Information
9. Unsupervised Hebbian Learning
10. Competitive Learning
11. Reinforcement Learning Procedures
12. Generalization



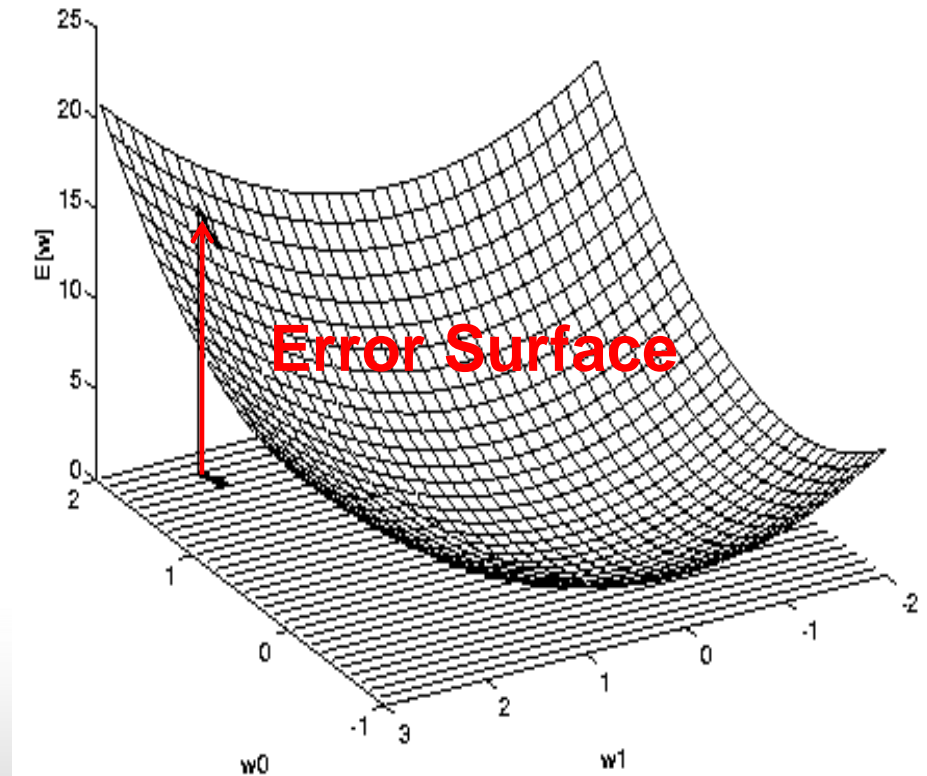
Simple Supervised Learning Procedures

Input units are directly connected to output units

States are a continuous smooth function of their total input

$$E = \frac{1}{2} \sum_{j,c} (y_{j,c} - d_{j,c})^2$$

$$\frac{\partial E}{\partial w_{ji}} = \sum_{\text{cases}} (y_j - d_j) \frac{dy_j}{dx_j} y_i$$



Simple Supervised Learning Procedures

$$\frac{\partial E}{\partial w_{ji}} = \sum_{\text{cases}} (y_j - d_j) \frac{dy_j}{dx_j} y_i$$

Nets with **linear** output units and **no hidden** unit:
always find the minimum

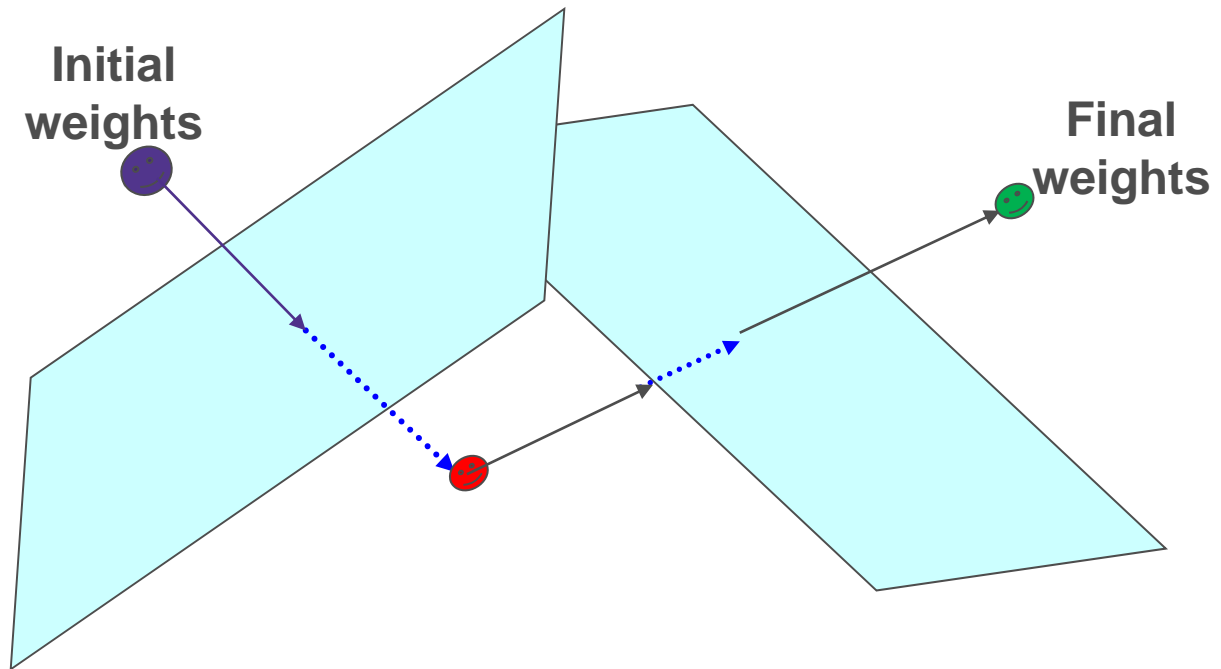
Nets with **nonlinear** output units and **monotonic** input-output function:
always find the minimum

Batch Version

Online Version

Simple Supervised Learning Procedures

Perceptron Convergence Procedure



Ignores magnitude of error.

Does not settle down when there is no perfect set of weights

Does not work when the idea of ideal region breaks down (Multilayer nets)

Simple Supervised Learning Procedures

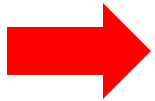
Deficiencies

Most "interesting" cannot be captured by any combination of weights in simple nets

Gradient descent may be very slow if the elliptical cross-section of the error surface is very elongated

Overview

1. Introduction
2. Connectionist Models
3. Connectionist Research Issues
4. Associative Memories without Hidden Units
5. Simple Supervised Learning Procedures
6. Back propagation
7. Boltzman Machines
8. Maximizing Mutual Information
9. Unsupervised Hebbian Learning
10. Competitive Learning
11. Reinforcement Learning Procedures
12. Generalization

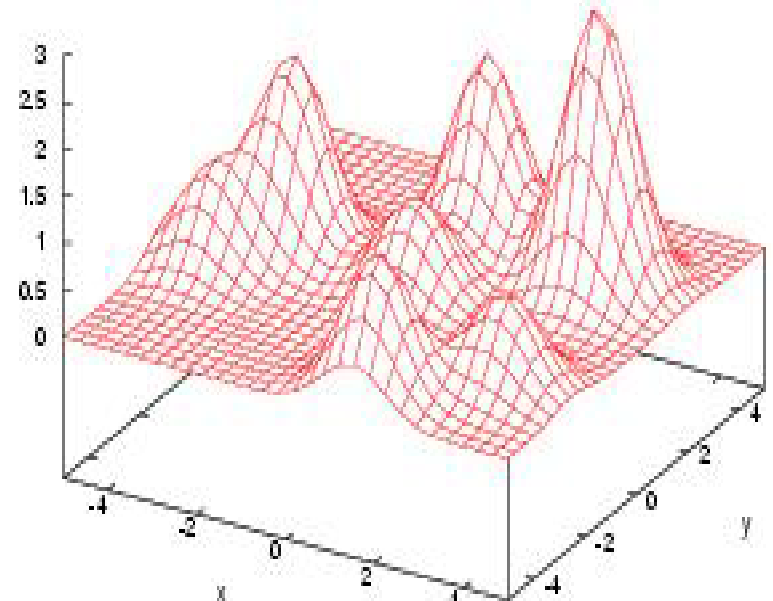


Back Propagation

Is a generalization of the least squares procedure that works for networks with hidden layers

The central idea: derivatives can be computed efficiently by starting with the output layer and working backwards through the layers.

Error surface in networks with hidden layers:



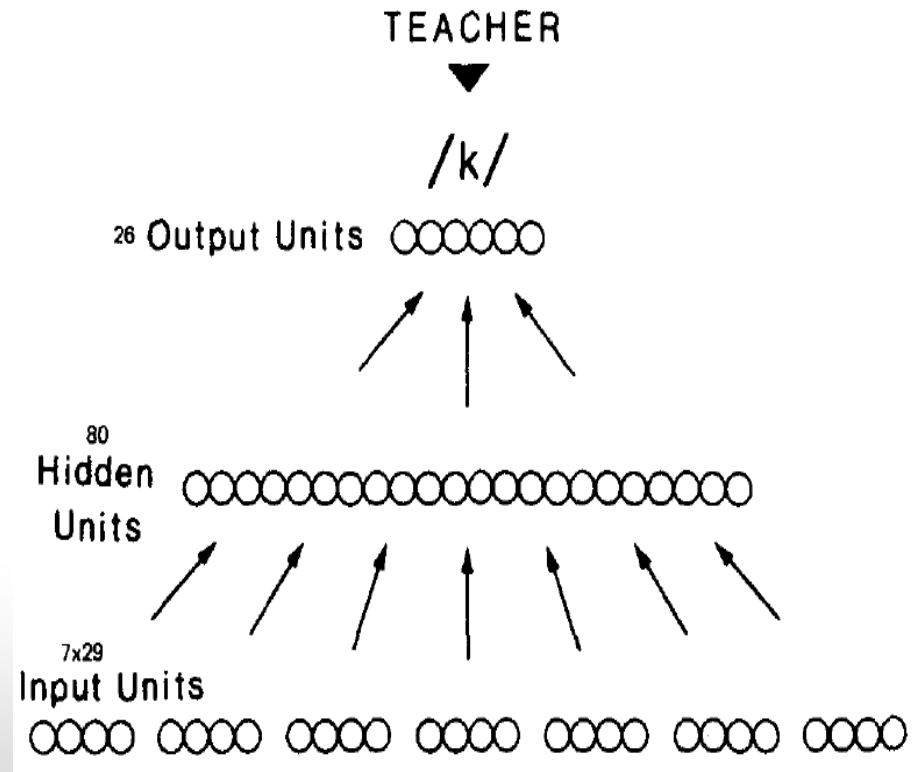
Back Propagation

Mapping text to speech

I am **Presenting**

Discovering Semantic Features

Phoneme Recognition



Back Propagation

- Reinforcement Version

First, the mental model learns to predict expected reinforcement

Second, derivative of the expected reinforcement can be backpropagated

- As a Maximum Likelihood Procedure


Interpret each output vector as a specification of a conditional probability distribution

Minimizing the squared error is equivalent to maximum likelihood estimation if output vectors are treated as the centers of Gaussian pdf.

- Deficiency

Is not adequate for large tasks because learning time scales poorly

Overview

1. Introduction
2. Connectionist Models
3. Connectionist Research Issues
4. Associative Memories without Hidden Units
5. Simple Supervised Learning Procedures
6. Back propagation
-  7. Boltzman Machines
8. Maximizing Mutual Information
9. Unsupervised Hebbian Learning
10. Competitive Learning
11. Reinforcement Learning Procedures
12. Generalization

Boltzman Machines

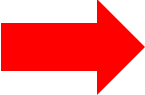
- Is a generalization of Hopfield network
- The units update their state according to a stochastic decision rule
- The units have state 0 or 1 according to the following probability

$$p_j = \frac{1}{1 + e^{-\Delta E_j / T}}$$

→ Temp.
→ Total Input

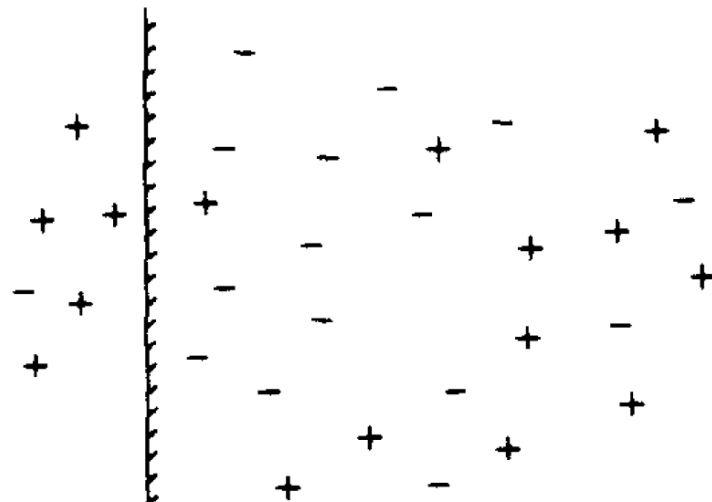
- If this rule is applied repeatedly, the network will reach thermal equilibrium
- The simplicity of the Boltzmann distribution leads to a very simple learning procedure which adjusts the weights so as to use the hidden units in an optimal way.

Overview


1. Introduction
2. Connectionist Models
3. Connectionist Research Issues
4. Associative Memories without Hidden Units
5. Simple Supervised Learning Procedures
6. Back propagation
7. Boltzman Machines
-  8. Maximizing Mutual Information
9. Unsupervised Hebbian Learning
10. Competitive Learning
11. Reinforcement Learning Procedures
12. Generalization

Maximizing Mutual Information

- One semisupervised method is to provide it with information about what category the input vector came from
- Its incoming weights are modified so as to maximize the information that the state of the unit provides about the category of the input vector
- The derivative of the mutual information is relatively easy to compute and so it can be maximized by gradient ascent



Overview


1. Introduction
2. Connectionist Models
3. Connectionist Research Issues
4. Associative Memories without Hidden Units
5. Simple Supervised Learning Procedures
6. Back propagation
7. Boltzman Machines
8. Maximizing Mutual Information
-  9. Unsupervised Hebbian Learning
10. Competitive Learning
11. Reinforcement Learning Procedures
12. Generalization

Unsupervised Hebbian Learning

- The weight modification depends on both presynaptic and postsynaptic activity
- It is shown that an unsupervised Hebbian learning procedure in which the weight change depends on the correlation of presynaptic and postsynaptic activity can produce a surprising number of the known properties of the receptive fields of neurons in visual cortex



Overview

1. Introduction
2. Connectionist Models
3. Connectionist Research Issues
4. Associative Memories without Hidden Units
5. Simple Supervised Learning Procedures
6. Back propagation
7. Boltzman Machines
8. Maximizing Mutual Information
9. Unsupervised Hebbian Learning
-  10. Competitive Learning
11. Reinforcement Learning Procedures
12. Generalization

Competitive Learning

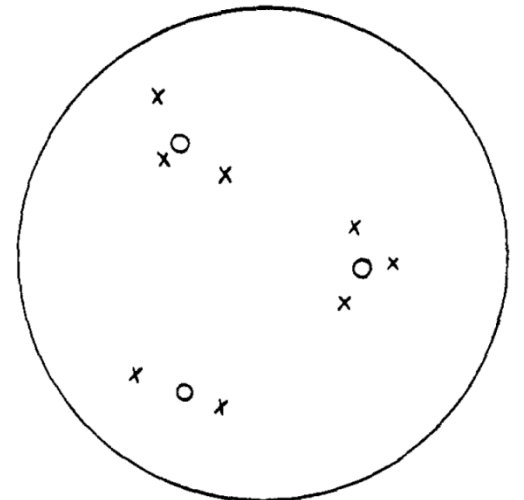
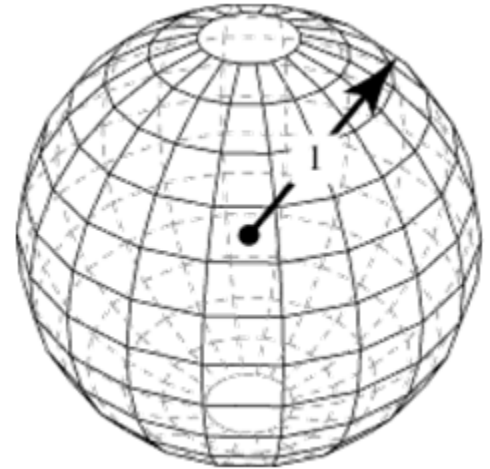
- Unsupervised learning
- Clusters inputs
- There is a set of hidden units which compete with one another to become active.
- When an input vector is presented to the network, the hidden unit which receives the greatest total input wins the competition and turns on with an activity level of 1.
- A constraint on each weight vector should be imposed to keep the sum of the weights (or the sum of their squares) constant

Competitive Learning

A constraint on each weight vector should be imposed to keep the sum of the weights (or the sum of their squares) constant

$$\sqrt{x^2 + y^2 + z^2} = 1$$

Simple Geometric Model



Overview

1. Introduction
2. Connectionist Models
3. Connectionist Research Issues
4. Associative Memories without Hidden Units
5. Simple Supervised Learning Procedures
6. Back propagation
7. Boltzman Machines
8. Maximizing Mutual Information
9. Unsupervised Hebbian Learning
10. Competitive Learning
11. Reinforcement Learning Procedures
12. Generalization



Reinforcement Learning Procedures

We can assign credit to a local decision by *measuring how it correlates with the global* reinforcement signal.

Advantage

It is easy to implement because does not require any special apparatus for *computing derivatives*.

Disadvantage

It is very inefficient when there are more than a few local variables.

A second disadvantage is that gradient ascent may get stuck in local optima.

Reinforcement Learning Procedures

- **Delayed reinforcement**

In many real systems, there is a delay between an action and the resultant reinforcement

Temporal credit assignment is performed by explicitly computing the effect of each activity level on the eventual outcome.

- **Genetic Algorithm**

Genetic algorithms operate on a population of individuals to produce a better adapted population.

There is a fitness function which assigns a real-valued fitness to each individual and the aim of the "learning" is to raise the average fitness of the population.

Overview

1. Introduction
2. Connectionist Models
3. Connectionist Research Issues
4. Associative Memories without Hidden Units
5. Simple Supervised Learning Procedures
6. Back propagation
7. Boltzman Machines
8. Maximizing Mutual Information
9. Unsupervised Hebbian Learning
10. Competitive Learning
11. Reinforcement Learning Procedures
12. Generalization



Generalization

A major goal of connectionist learning is to produce networks that generalize correctly to new cases after training on a sufficiently large set of typical cases from some domain.

Improve Generalization

To introduce an extra term into the error function. This term penalizes large weights and it can be viewed as a way of building in an a priori bias in favor of simple models

To impose equality constraints between weights that encode symmetries in the task.

Conclusion

There are now many different connectionist learning procedures and many more variations will be discovered in the next few years.

Major new advances can be expected on

- Making the learning time scale better
- To apply connectionist procedures to difficult tasks like speech recognition
- Simulating much larger networks
- Interpreting the behavior of real neural networks.