Connectionist Models

- *artificial neural networks*
- *parallel distributed processing*
- *subsymbolic AI*

Connectionist Models

- "The idea of parallel distributed processing . . . intelligence emerges from the interactions of large numbers of simple processing units" (Rumelhart et al. 1986, p. ix).
- This idea actually encompasses more than connectionist models. Since the processing units are unconstrained, except for being relatively simple, networks of automata, with relatively simple finite state machines at each mode, would qualify.
- The statement also doesn't restrict the way nodes are connected.

Connectionism : Lloyd

- Connectionism can be thought of as the brain model of cognition. One models notions of cognition on how brains apparently do it

- Lloyd, a philosopher (1989, p. 90): "The central idea of connectionism is that cognition can be modeled as the simultaneous interaction of many highly interconnected neuronlike units." Lloyd's version constrains the processors to be neuronlike and highly interconnected. "Highly interconnected" should be taken to require "many" connections from one unit to others

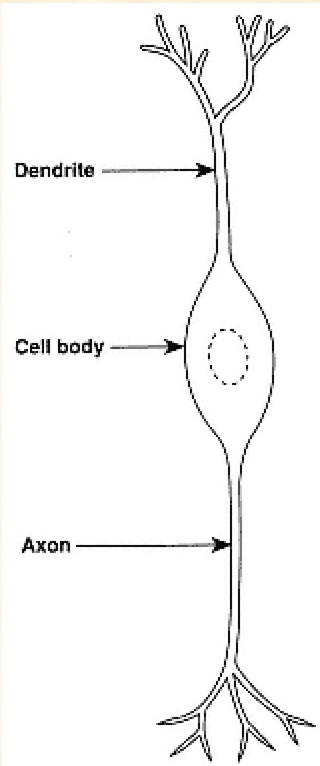
Chapman : essential connectionism

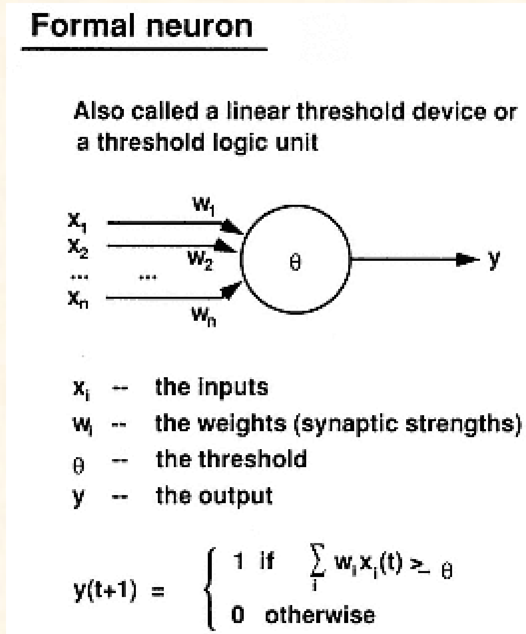
- A connectionist model must be brainlike, means?
- The essential connectionist facts are that the brain :
  - is made up of a great many components (about 10^{11} neurons)
  - each of which is connected to many other components (about 10^4)
  - and each of which performs some relatively simple computation (whose nature is unclear)
  - slowly (less than a kHz[3])
  - and based mainly on the information it receives from its local connections.

Simplifying assumptions

- Connectionist models are mathematical or computer models based roughly on the local structure of nervous systems of animals.
- many of the features of nervous systems are suppressed and that several simplifying assumptions are made
- The simplifying assumptions are much too great to allow these models typically to interest neuroscientists

Modeling a Neuron





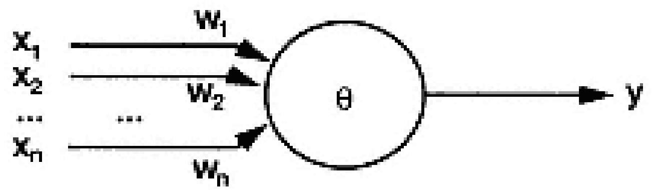
Its dendritic tree collects excitatory and inhibitory inputs from other neurons, and passes these messages, as voltages, on to the cell body or soma. These voltages are added to its current voltage, if excitatory, or subtracted, if inhibitory. When a threshold is exceeded, an output voltage signal is transmitted down an axon to synapses that connect the leaves of the axonic tree to the dendrites of other neurons.

Modeling a Neuron mathematically

Each synapse is a chemical connection between the axon in one neuron and a dendrite of the next. When the signal arrives at a synapse, vesicles of a chemical neurotransmitter are popped. The neurotransmitter then disperses across the synaptic gap, where it is picked up as an excitatory or inhibitory input on a dendrite of the postsynaptic neuron. And the process continues.

Formal neuron

Also called a linear threshold device or a threshold logic unit



x_i -- the inputs
w_i -- the weights (synaptic strengths)
θ -- the threshold
y -- the output

$$y(t+1) = \begin{cases} 1 & \text{if } \sum_i w_i x_i(t) \geq \theta \\ 0 & \text{otherwise} \end{cases}$$

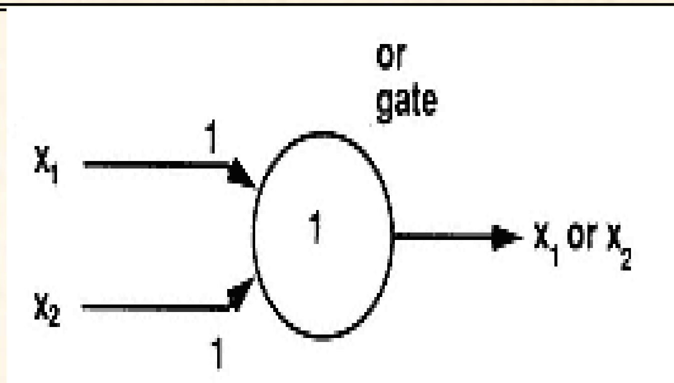
The mathematical Neuron

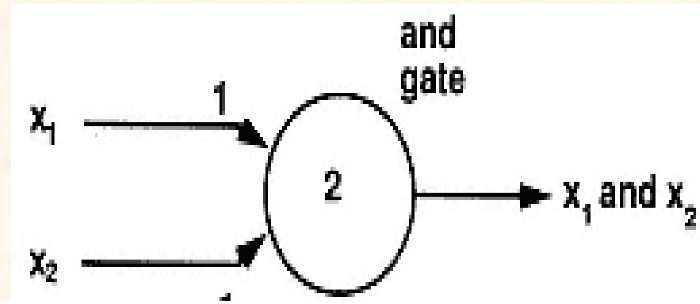
- Note that our formal neuron, or linear threshold device, is discrete both in time and in input/output values. It is discrete in time in that time proceeds in discrete moments enumerated by t = 0, 1, 2, 3, . . . , t, t + 1, . . . The input and output values are restricted to the discrete set {0, 1}. Linear threshold devices are due to McCulloch and Pitts (1943).
- The signals arriving at synapses are modeled by the inputs, the synaptic efficiency by the weights, the accumulation of voltages in the cell body by the weighted sum, the actual threshold by the formal threshold, and the voltage traveling down the axon by the output

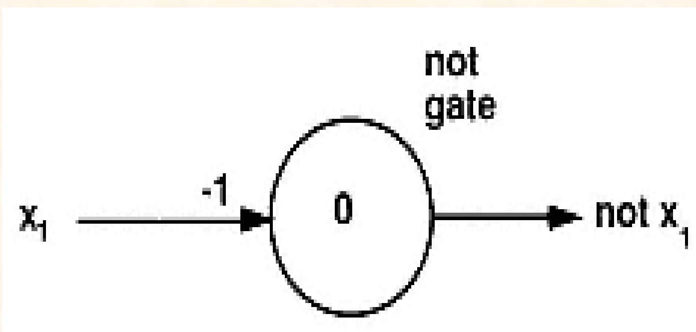
Major simplifications

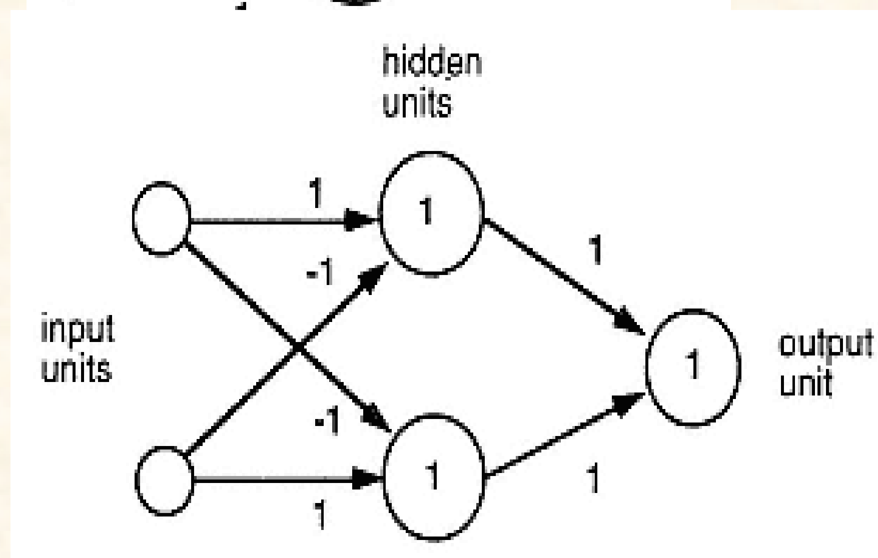
- In our model the output signal is represented by the magnitude of the output y, whereas in real neurons the significance of the signal most frequently is carried by the rate of firing, not the magnitude of a single firing.
- Another simplification allows a single neuron to excite one subsequent neuron via a positively weighted synapse, and to inhibit another via a negatively weighted synapse. In real nervous systems, individual neurons are either inhibitory or excitatory to all subsequent neurons to which they connect

Computation with Formal Neurons







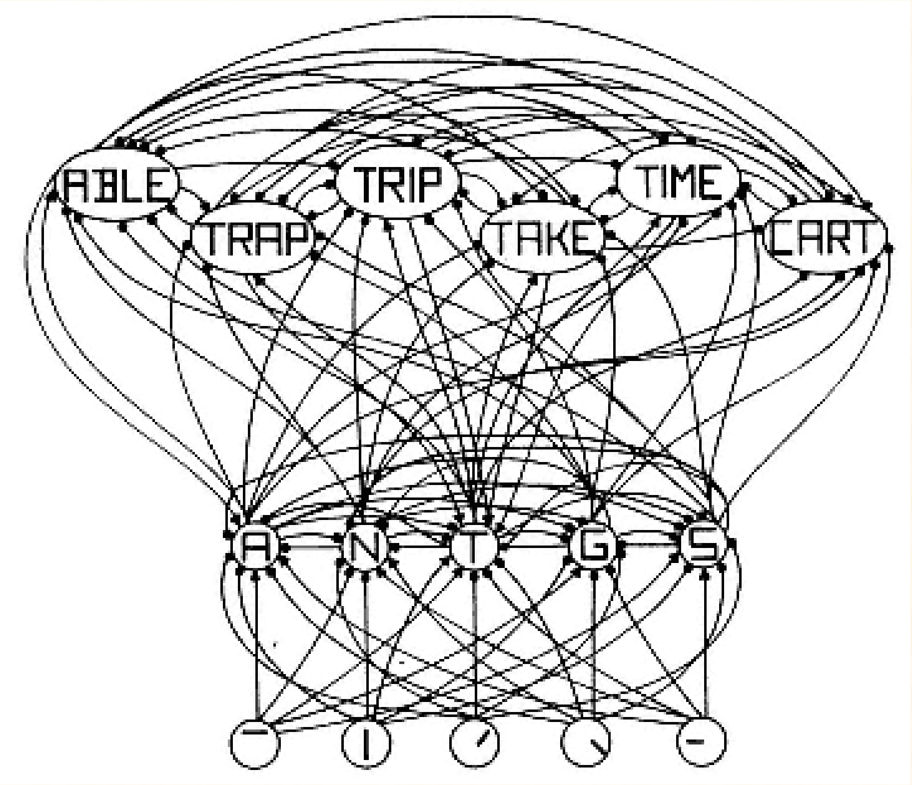


An XOR network

Feedback and Word Recognition

- McClelland and Rumelhart (1981; Rumelhart and McClelland 1982) tackled the problem of word recognition
- Their idea for recognizing words is to begin by recognizing letters. Their idea for recognizing letters is to recognize the strokes that produced them. To keep the project computationally feasible, they restricted themselves to four-letter words. For each of the four possible letter positions in a word, twenty-six letter units locally represent the letters of the English alphabet

Word Recognition Network

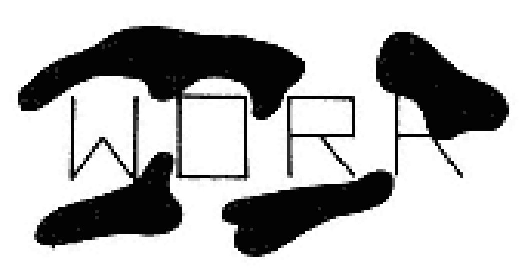


Word Recognition Network

- For the purpose of recognizing words, strokes are unrelated to one another. Hence, there are no connections between units in any of the stroke layers. The stroke layer and the letter layer, for each letter position, are*fully interconnected*, meaning that each stroke unit is connected to each letter unit. A stroke unit excites a letter unit if the corresponding stroke is used in forming the corresponding letter, and inhibits it otherwise. Excitatory connections are shown with little arrows and inhibitory connections end with small disks. Within the letter layer, for each letter position, a single unit must emerge victorious. Thus each letter unit within a given position inhibits every other letter unit, producing a *winner-take-all* situation. Similarly, each word unit inhibits every other word unit, resulting in another winner-take-all strategy that serves to recognize exactly one word.

Word Recognition Network : Generalization

- Connections between the word layer and the four letter layers occur in both directions. A unit representing a word, say ABLE, with the letter A in the first position, is excited by the unit representing the letter A in the first position, and is inhibited by each unit representing some other letter in that position. And similarly for the B in the second position. Feedback occurs when the A in the first position unit is excited by the ABLE unit, as is the B in the second position unit, and so on..



- What will be the output for this noisy input?
- we reason that the last letter must be either an R or a K, and that WORR isn't a word

Word Recognition Network

- The units representing W, O, and R should rapidly gain ascendancy in their respective positions because each of their strokes is present and no others are. Units representing K and R should compete for ascendancy in the fourth position. Having the same strokes present, they should initially rise about the same amount. The feedback from the word level eventually distinguishes between them.
- At the word level, units representing WORD, WORK, and FORK, among others (say WEAK, or WEEK), may be initially highly activated by virtue of several of their corresponding letter units running high. These letter units are, in turn, strengthened by the three word units. But D cannot win out in the fourth position because too little activation came from the strokes. Lack of input from D eventually weakens WORD. Similarly, FORK can't win out for lack of input from F in the first position. The network settles on WORK
- The network is acting as a *content addressable memory*, able to access the appropriate data record from part of its contents.

ANN is different from Symbolic Mind

- ANN’s ability to look at incomplete information and come to a correct conclusion. The popular computer maxim, "Garbage in, garbage out," seems to be violated here.
- From a cognitive science point of view, the network accomplishes word recognition, a form of categorization. From a computer science point of view, the network is acting as a *content addressable memory*, able to access the appropriate data record from part of its contents.
- Note that this process arrives at its correct conclusion via a form of statistical calculating, giving rise to a global dynamics that settles into some stable situation.
- This process apparently is not rule based. There's nothing that looks quite like the syntactic operators applied to symbols.

Representation

- Are representations important ?
- Symbolic Mind :

  - "The representation principle: Once a problem is described using an appropriate representation, the problem is almost solved." (Winston , 1992)

  - Horgan and Tienson (1989) concludes that humans typically operate using representations but not rules.
- Cognitive cultures people maintain that representation can be dispensed with altogether (Maturana and Varela 1980; Winograd and Flores 1986; Brooks 1991; Varela et al. 1991)
- Connectionism : Representation is necessary but not at the symbolic level

Local and distributed connectionist representations

- *Local representation* employs one unit to represent one object, one concept, or one hypothesis. A one-to-one correspondence is set up between the units in the network and the items to be represented. Thus, with n units I could represent n objects; the ability to represent increases linearly with the number of units. The XOR and word recog networks use local representations.
- With *distributed representation*, on the other hand, each unit may participate in the representation of several items, and conversely, each item is represented by a pattern of activity over several different units. Thus you can represent as many items as you have subsets of units; n units can represent 2^n items. Distributed representation is much more computationally compact than local representation, whereas local representation is much easier to comprehend
- A third type of artificial neural network representation is *featural representation*(Lloyd 1989)

featural representation and types of connectionist models

- It occupies a middle ground between local and distributed representation.
- Here individual items are represented distributively by patterns of activity over sets of units, and the individual units locally represent features of the given item.
- Featural representations can be programmed into a system or can occur with no human forethought leading to spontaneous generalization.
- Lloyd suggests classifying connectionist models according to their type of representation.
- Locally represented systems model at the fully cognitive level. Since units represent concepts, the dynamics of the network would model the dynamics of thought itself.
- The distinction between the computational view of mind held by symbolic AI and traditional cognitive science, and a strong form of the connectionist view, is particularly clear here. In a locally represented connectionist model, there is nothing that looks at all like symbol manipulation via rules. There is no recognizable stored program. All this conflicts with the computer model of mind.

types of connectionist models

- Featurally represented systems model at what Lloyd calls the microcognitive level
- At the conceptual level, patterns of activation interact in complex ways. Thoughts are then activation vectors. At the featural level, units can still be interpreted, inferences can sometimes be discerned, and the cognitive level is emergent.
- Fully distributed systems model at the subcognitive level, since the individual units are uninterpreted.
- Cognition emerges only at the whole system level. Thus the connectionist model may be implementing some production system when viewed from a higher level of abstraction.

Characteristics of Artificial Neural Networks

Inherent advantages in their architecture :

- **One of the most important properties of an artificial neural network is its lack of a central executive**. Nobody's in charge. No one has the overall picture and makes decisions. Control is distributed.
- All decisions are made on local information. Each unit decides on its output solely on the basis of the input it gets from its neighbors, possibly including itself, and its internal state.
- the network incurs no expense gathering global information and arriving at a global decision.
- **Another benefit of the architecture is the automatic presence of default assignments**.
- The network is going to do something, no matter what. That something is a default value for the situation at hand. With symbolic systems, a programmer laboriously builds in whatever defaults are to be present.

Characteristics of Artificial Neural Networks

- Artificial neural networks complete patterns such as the corrupted word WORK. Pattern completion allows content addressability. Some of these networks have been designed as little databases.

  If you want information out, just put part of it in.
- Symbolic AI systems tend to be brittle, whilethe connectionist networks, using distributed representation, tend to degrade more gracefully.

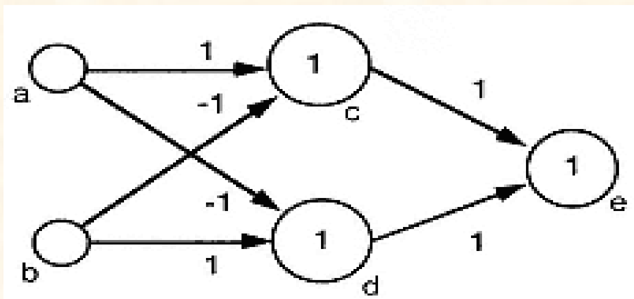
  In symbolic systems, a single misplaced comma in its program can result in catastrophic failure, while in connectionist systems The failure of a small number of units is more likely to cause some degradation of performance rather than a catastrophic failure.
- Spontaneous generalizations in ANN often exhibit global behaviors beyond the scope of any of their individual units. These are called emergent behaviors, and are considered by some to be a distinct advantage.

Learning by Artificial Neural Networks

- And finally, there's the feature of artificial neural networks that brought them fame and fortune — learning.
- The output of a unit is often referred to as its *activation*. The pattern of this activation at a given time is called the network's *configuration* at that time. The configuration can be called as short-term memory, or what the network is representing at the moment.
- The pattern of weights, on the other hand, is more like long-term memory, representing what the network knows. This pattern of weights determines how the network will react to a given stimulus.
- In order for an artificial neural network to learn, it must change some of its behaviors, that is, it must respond differently to a given stimulus. For this to happen, weights must change.
- **Thus *learning in artificial neural networks takes place via changing of weights*.**

Learning can be viewed as a search problem in weight space.

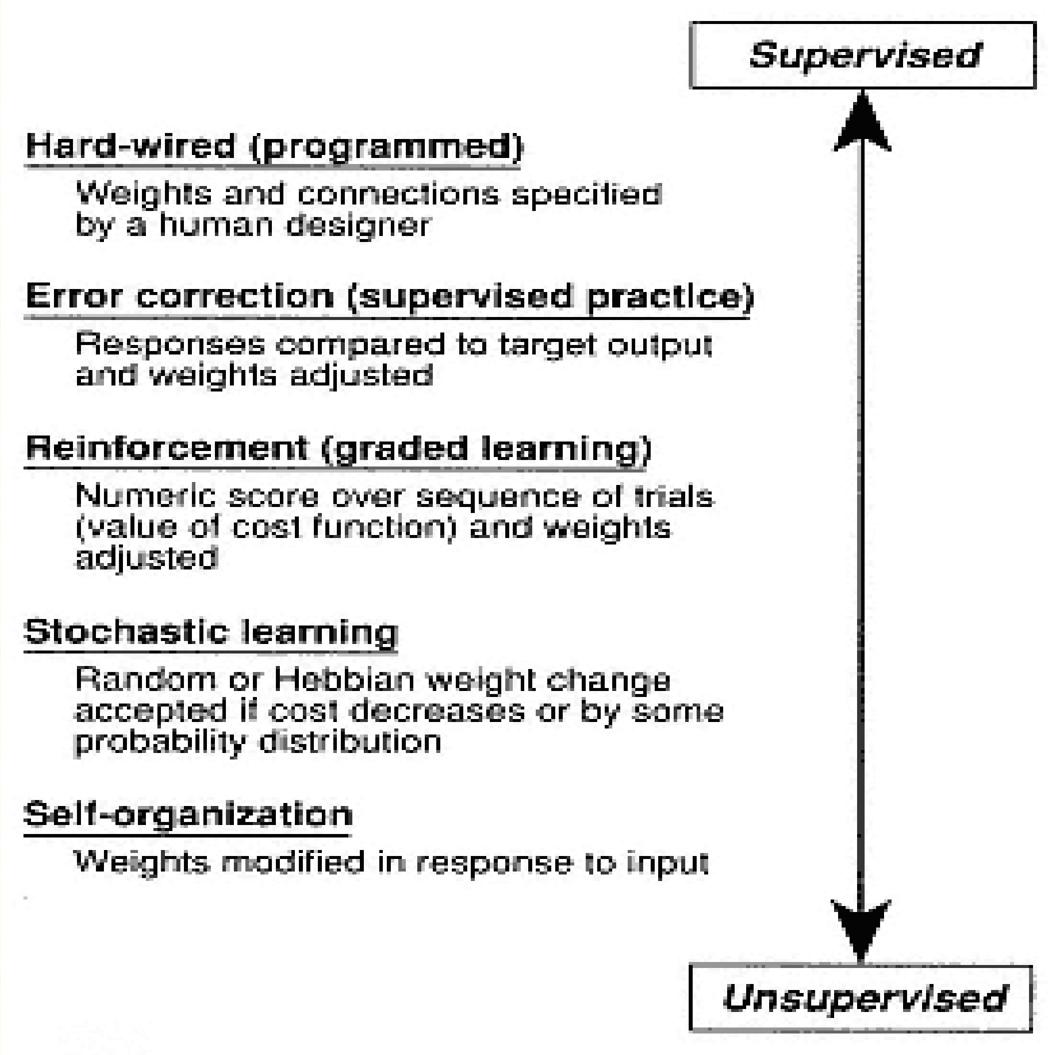
- An artificial neural network is build on a directed graph, or digraph,[11] called the *architecture* of the network. The units of the network are the nodes of the digraph, and the weighted links are its arcs. Given a particular architecture, the set of all its possible weight matrices comprises its *weight space*.



$$
\begin{array}{c}
\begin{array}{ccccc} a & b & c & d & e \end{array} \\
\begin{array}{c} a \\ b \\ c \\ d \\ e \end{array}
\begin{bmatrix}
0 & 0 & 1 & -1 & 0 \\
0 & 0 & -1 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0
\end{bmatrix}
\end{array}
$$

The architecture of the XOR network above is a digraph consisting of five nodes and six arcs. Its weight space will be composed of all 5 × 5 matrices containing 0's in all but the six positions specified by the existing arcs in the digraph. A 5 × 5 weight matrix can be thought of as a point in a 25-dimensional space. The weight space can then be thought of as a subset of this 25-dimensional space. Learning, then, can be viewed as a search problem in weight space, looking from one weight matrix to another, trying to find the one that will do the job best.

Learning (Training) Approaches : Searching Heuristics



Weight Updation approaches