Deriving Rules From Data

Deriving Rules from Data
Machine Learning Algorithms

Neural Nets

Khasha Dehnad

Neural Networks

Simulating the Brain to Solve Problems
Artificial Neural Networks (ANN)

Overview

- **Computer emulation of biological neural systems for building models:**
  - **initially theorized in 1943 by McColloch and Pitts of University of Chicago,**
  - **simulates the brain's cognitive learning process,**
  - **"learns" patterns directly from the data,**
  - **searches for complex relationships,**
  - **automatically builds models,**
  - **predicts - compares - adjusts,**
  - **corrects the model's mistakes over and over again,**
  - **input: Data,**
  - **output: Prediction**
  - **tool: the Model "learned" from the Data.**

Neural Networks
Biological Principles Underlying the Neural Network Technology

The idea of neurons as the structural constituent of the brain was first introduced by Ramon y Cajal (French) 1911

- **Human Brain:**
  - **a network of individual but interconnected nerve cells called *neurons* (10^11 *neurons*)**
  - **neurons are connected to each other via huge number of so-called *synapses* (10^15 synapses or connections),**
  - **a given neuron is connected to 10 thousand other neurons by these synapses,**
  - **neurons can receive information from the outside world at various points in the network,**
  - **these pieces of information are called *stimuli,***
  - **a neuron transfers information on to other neurons by firing chemicals called *neurotransmitters,***
  - **these transfers occur over synapses like bursts of electricity,**
  - **the more important a particular stimulus is, the stronger the burst will be at the synapses,**
  - **the information received by a nerve cell at one of the synapses either *excite* or *inhibit* the cell,**
  - **if the receiving cell is excited, it will pass the information to other neurons,**
  - **if the receiving cell is inhibited, it will damp the impact of the information,**
  - **each nerve cell processes the raw input but passes it on only if it is important,**
  - **the information travels through the network by generating new internal signals,**
  - **the stimuli are processed by brain and nervous system and ultimately a response is produced.**

Neural Networks
Artificial Neural Networks

–**a system of neurodes (*nodes*) and *weighted connections* (synapses) inside the memory of a computer,**

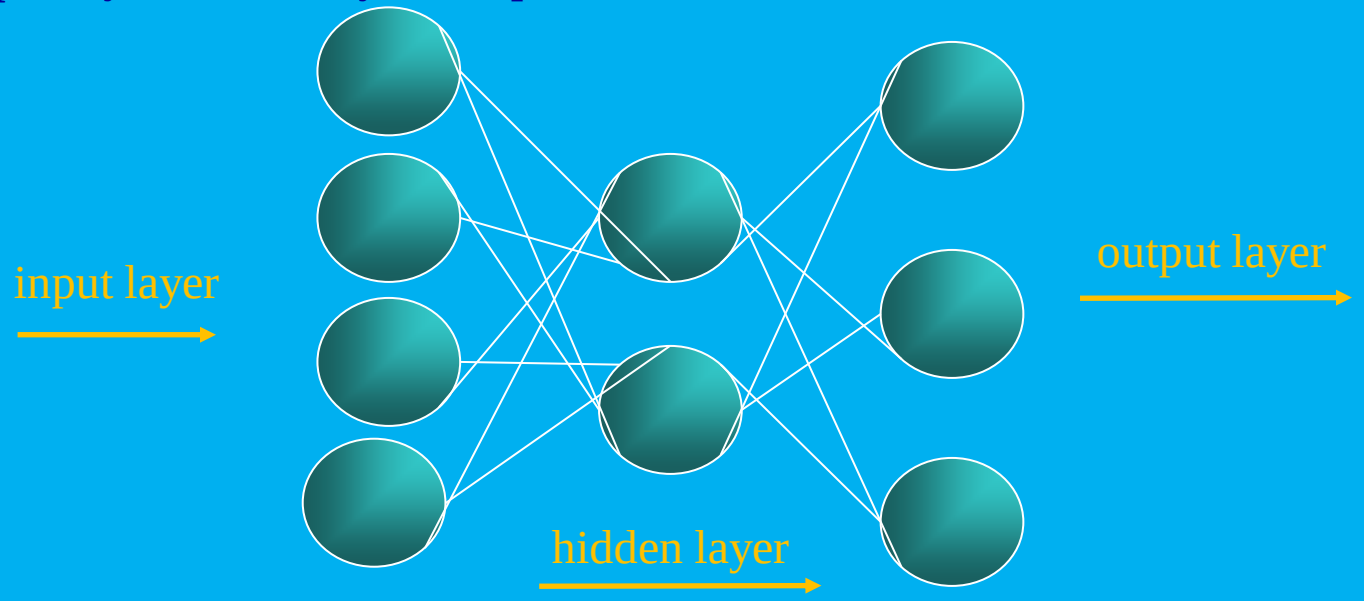
–**nodes are data storage locations (like variables in a program, cells in a spreadsheet),**

– **nodes are arranged in *layers* with weighted connections running between layers,**

– ***balls* represent nodes and *lines* represent connection weights,**

– ***input* layer nodes receive the data,**

– ***output* layer nodes relay the response of the neural network out of the net,**



A Simple Neural Network

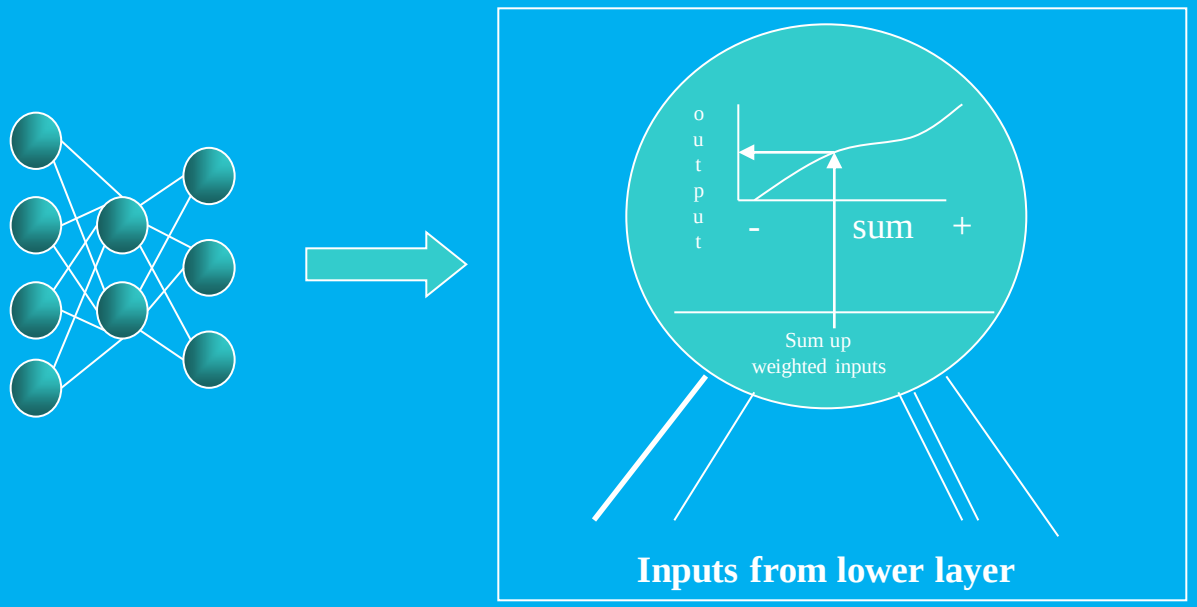
Neural Networks
ANN (Continued)

- ***hidden*** **layer nodes (hidden from the outside world) conduct the internal processing,**
- **data are fed into the net through the input nodes,**
- **data are processed internally by hidden nodes, based on the inter-node connection weights,**
- **result are passed on to the outside world by output nodes,**
- **"learning" takes place through adjusting connection weights,**
- **a "learned" neural network has adjusted its weights properly**

ANN operates in the same way as the biological model on which it is based.

Neural Networks
Application of a Learned Neural Network

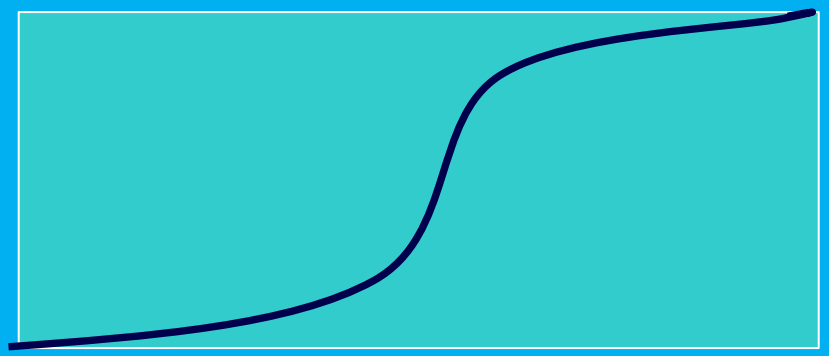
- **Integration Function:**
  - **each neuron receives a set of raw data (input),**
  - **the neuron multiplies each input by the connecting weight leading into it,**
  - **connection weight determines the importance of a given input in contribution to the output of the neuron,**
  - **more important inputs will have bigger weights and less important ones will have smaller weights,**
  - **the *integration function* of the neurode calculates a weighted sum of all inputs.**



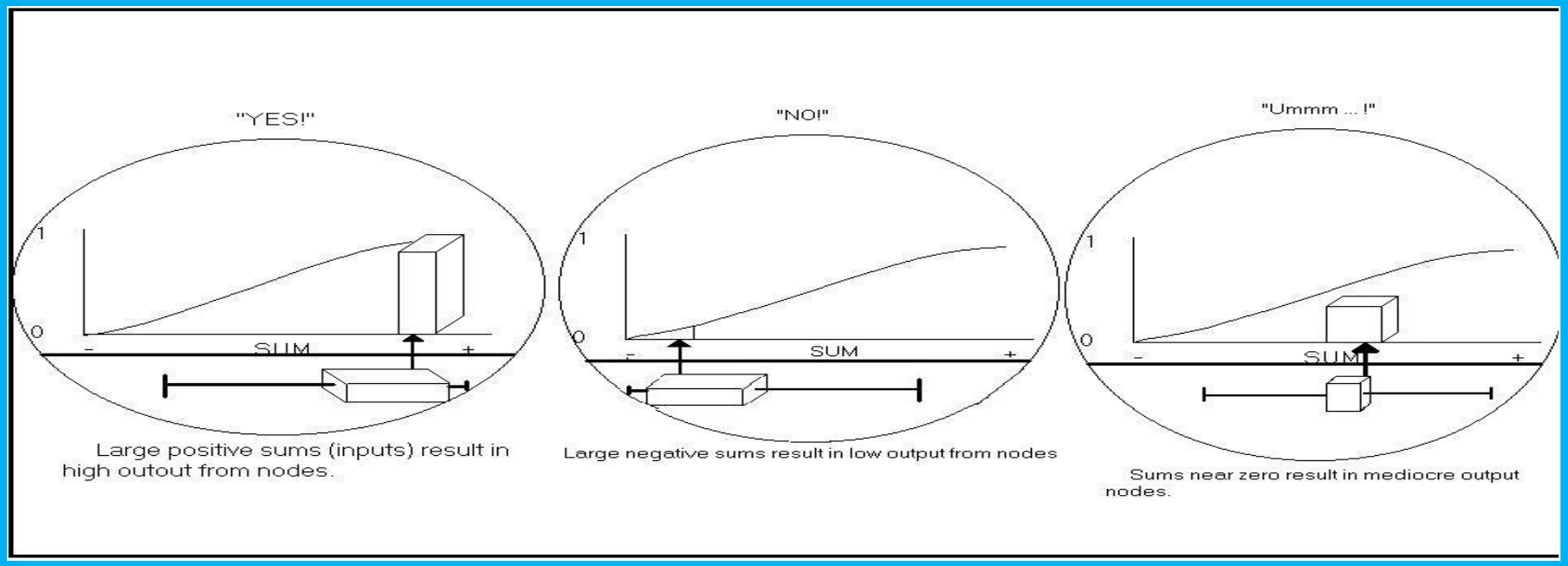
Neural Networks
Application of a Learned Neural Network (Continued)

- **Transfer Function:**
  - **the weighted sum is converted into an output value using a mathematical function called *transfer function*,**
  - **transfer function normalizes the output into the range of [0,1],**
  - **it serves as a kind of "dimmer" switch for turning the neuron "on" and "off ",**
  - **the transfer function's value will be *high* (excited) when the sum of the inputs is large & positive; and *low* (inhibited) when the sum is large negative,**
  - **the transfer function determines the degree at which a given sum will cause a neurode to fire.**

$$f(\text{net} - z) = 1/(1 + e^{-x})$$



Neural Networks
Application of a Learned Neural Network (Continued)



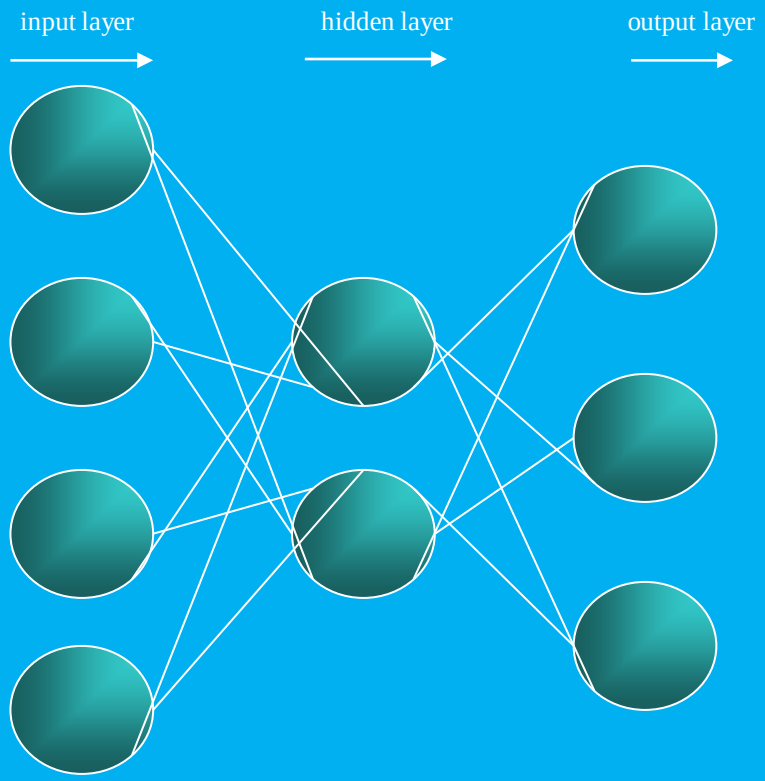
$$W_{ij}New = W_{ij}Current + \Delta w_{ij}$$

$$\Delta W_{ij} = \eta \delta_j X_{ij}$$

$$\delta_j =$$

$$output_j(1 - output_j)(actual_j - output_j)$$

$$output_j(1 - output_j)\sum W_{jk}\ \delta_j$$



A Simple Neural Network

Neural Networks
Application of a Learned Neural Network (Continued)

- See the Excel file

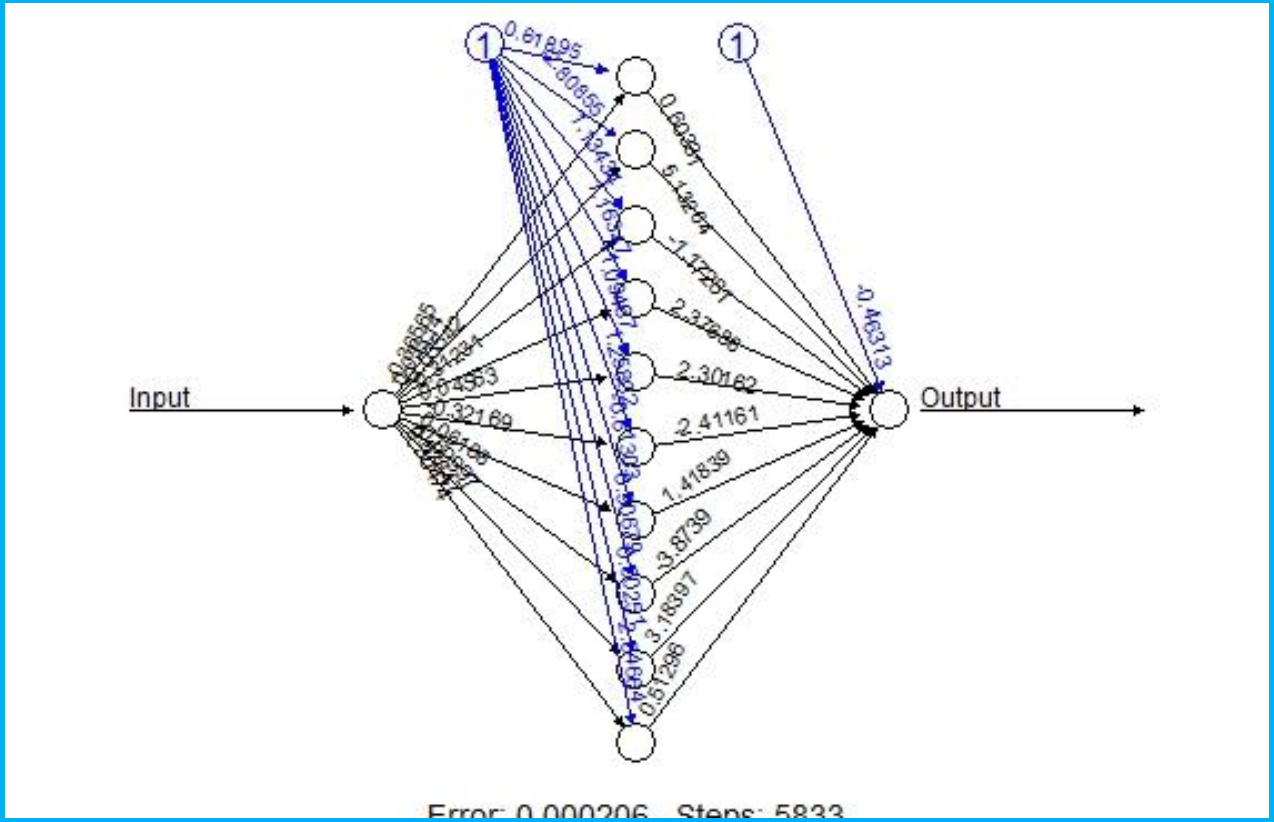
Neural Networks

Application of a Learned Neural Network (Continued)

	X	Y
1	64.577965694	8.036041668
2	81.738664140	9.040943764
3	12.522565899	3.538723767
4	29.109585518	5.395329973
5	16.068743286	4.008583701
6	9.047381557	3.007886560
7	55.963591277	7.480881718
8	91.689337464	9.575454948
9	46.862516948	6.845620275
10	46.512397029	6.819999782
11	20.373451896	4.513696035
12	70.124539430	8.374039612
13	36.501144245	6.041617684
14	44.268320873	6.653444286
15	24.348102487	4.934379646
16	98.336538277	9.916478119
17	6.548202015	2.558945489
18	84.601293202	9.197896129
19	25.546731777	5.054377487
20	23.428429803	4.840292326

Neural Networks
Application of a Learned Neural Network (Continued)

X Y



net.sqrt <- neuralnet(Output~Input,trainingdata, hidden=10, threshold=0.01)

Neural Networks

- **Steps: 0: decide on NN architecture: input nodes, hidden layers/nodes, output nodes**
  - **step 1: start with a set of "training data" where both the inputs and the output(s) are known, "correct" output, the experience to be used to train the net,**
  - **step 2: set all the initial connection weights to arbitrary small random numbers, sum about 0, output about 0.5, initial neutral position,**
  - **step 3: present the net with one case of input data, let the net predict the output (Yk), compare the predicted output with the "correct" output (Dk),**
  - **step 4: prediction matches the correct output: Yk –Dk=0, do nothing,**
  - **step 5: prediction deviates from the correct output: calculate the error, adjust weights to reduce the error, : predict - compare - adjust,**
  - **step 6: repeat the predict - compare - adjust process for each of the cases (often many) of the "training data" until the network has been "trained", that is, the prediction error has been *"minimized"*.**
  - **Step 7: test the trained net with a set of "testing data" that it had never seen before. If the performance is satisfactory, then we have a "learned" net model.**

Neural Networks
A Case Study: Real State Appraisal

- **Business Need:** An accurate appraisal of the market value of a real state property is crucial for a large mortgage company that needs to assess the risk of an individual loan.

- **Goal:** A model to appraise market value of a home based on characteristics of the property.

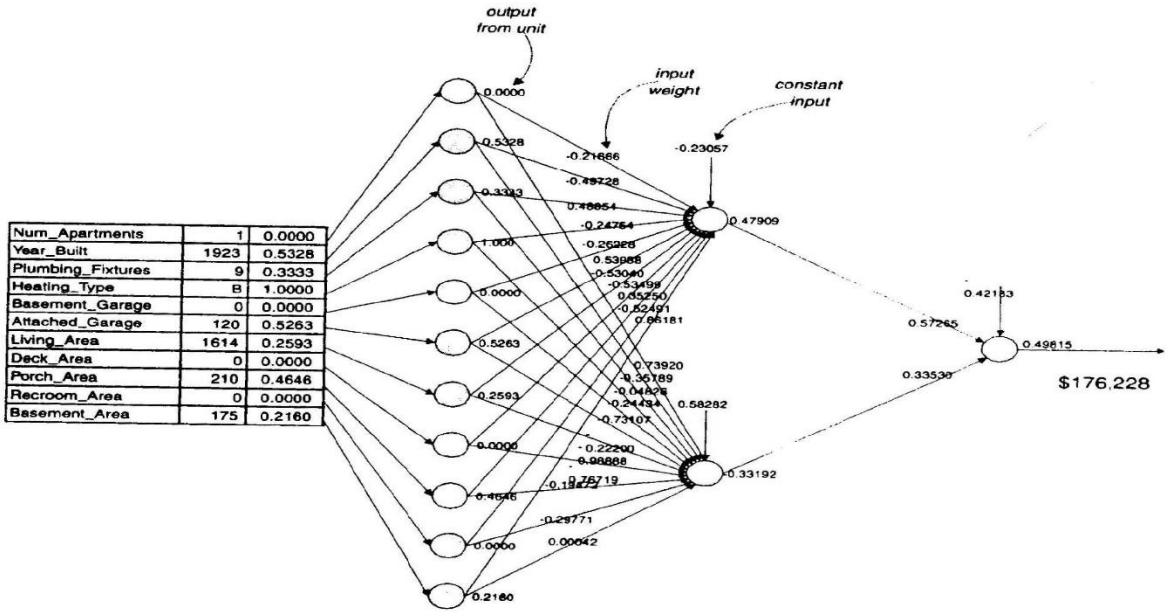
- **Input (Independent) Variables:** Common variables (features) describing a house

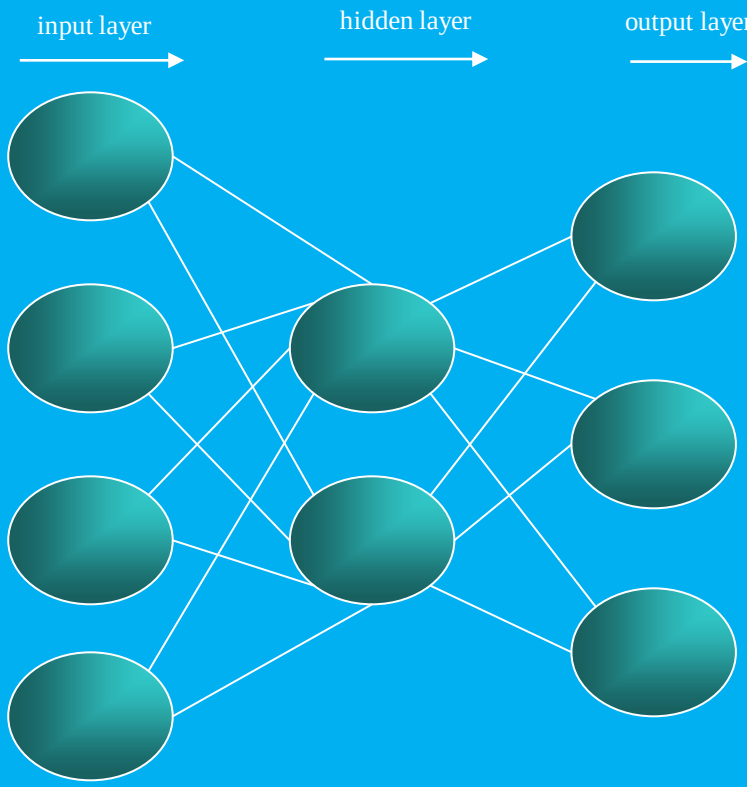
Feature	Description	Values
Num_Apartments	Number of dwelling units	1-3
Year_Built	Year Built	1850-1986
Plumbing_Fixtures	Number of plumbing fixtures	5-17
Heating_Type	Heating system type	Coded as A or B
Basement_Garage	Basement garage (Number of cars)	0-2
Attached_Garage	Attached frame Garage Area (Square feet)	0-228
Living_Area	Total living area (Square Feet)	714-4185
Deck_Area	Deck/open porch area(Square feet)	0-738
Porch_area	Enclosed porch area(Square feet)	0-452
Recroom_area	Recreation room area(Square feet)	0-672
Basement_area	Finished basement area(Square feet)	0-810

Neural Networks
A Case Study: Real State Appraisal (contd.)

- **Output (Dependent) Variable (s):**
  - **sales price (correct output for training, market value appraisal for prediction),**
- **Data Source: Federal Home Loan Mortgage Corporation**
- **Neural Net Topology (Architecture):**
  - **input layer consisting of 11 nodes (one node for each input variable),**
  - **one hidden layer consisting of two nodes fully-connected to all input nodes,**
  - **output layer consisting of one node (one output is required: market value of home) fully-connected to all hidden nodes.**

Neural Networks
A Case Study: Real State Appraisal
Real Estate Appraisal Neural Net





A Simple Neural Network

Neural Networks
Artificial Neural Networks

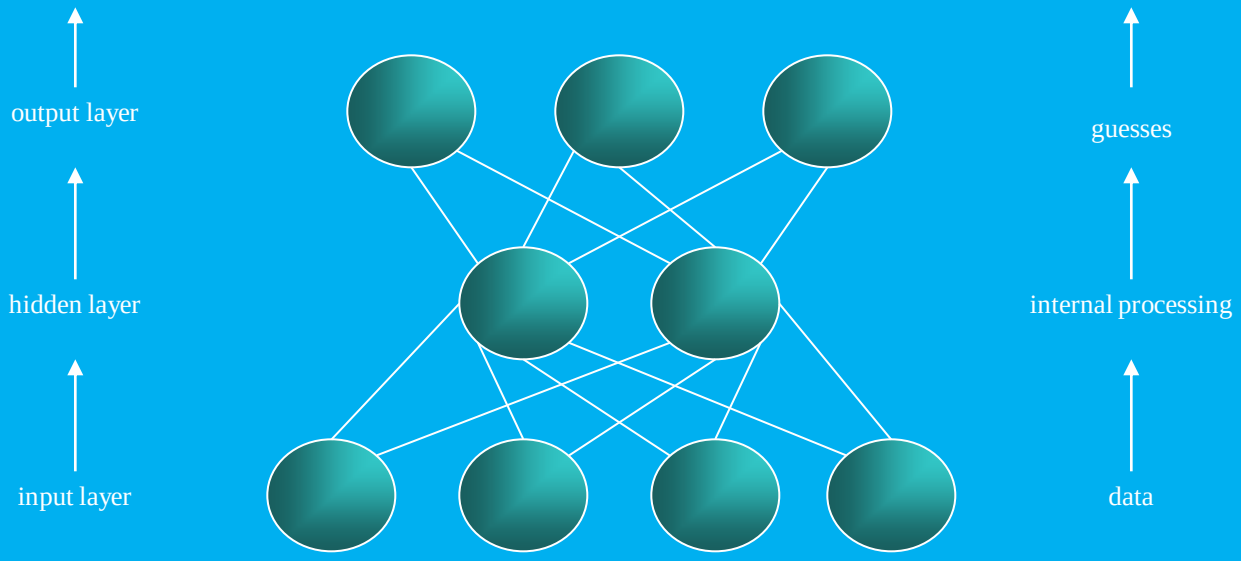
$$W_{ij}\,New = W_{ij}\,Current + \Delta w_{ij}$$

$$\Delta W_{ij} = \eta \delta_j X_{ij}$$

$$\delta_j =$$

$$output_j (1 - output_j)(actual_j - output_j)$$ Output layer

$$output_j (1 - output_j) \sum W_{jk}\ \delta_j$$ Hidden layer



A Simple Neural Network

Neural Networks
Artificial Neural Networks

–a system of neurodes (*nodes*) and *weighted connections* (synapses) inside the memory of a computer,

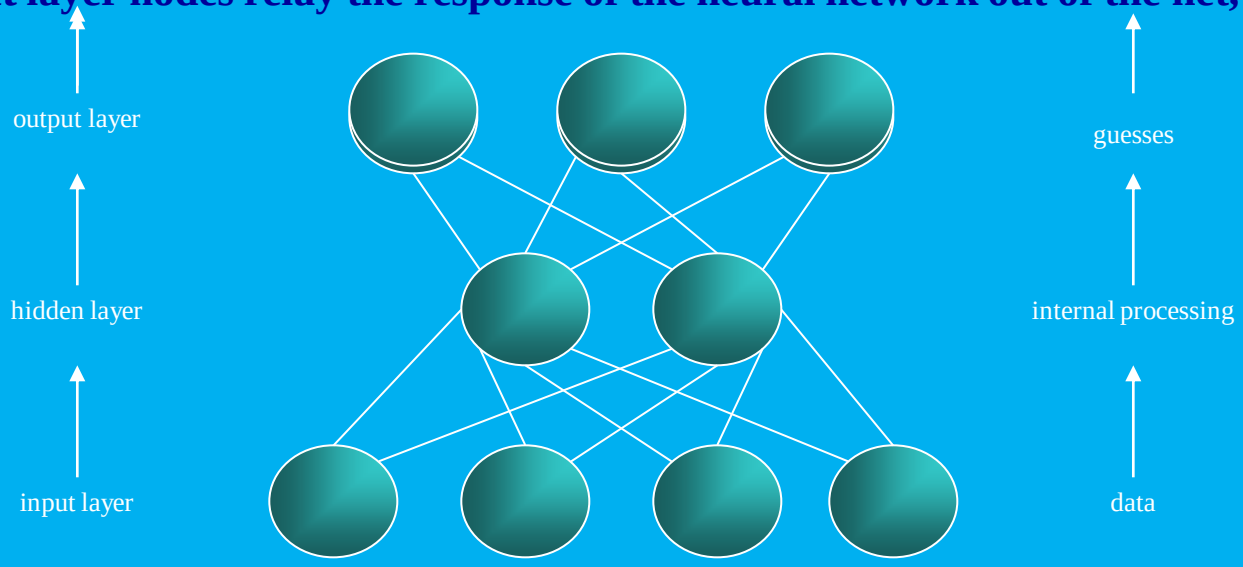
–nodes are data storage locations (like variables in a program, cells in a spreadsheet),

– nodes are arranged in *layers* with weighted connections running between layers,

– *balls* represent nodes and *lines* represent connection weights,

– *input* layer nodes receive the data,

– *output* layer nodes relay the response of the neural network out of the net,



A Simple Neural Network

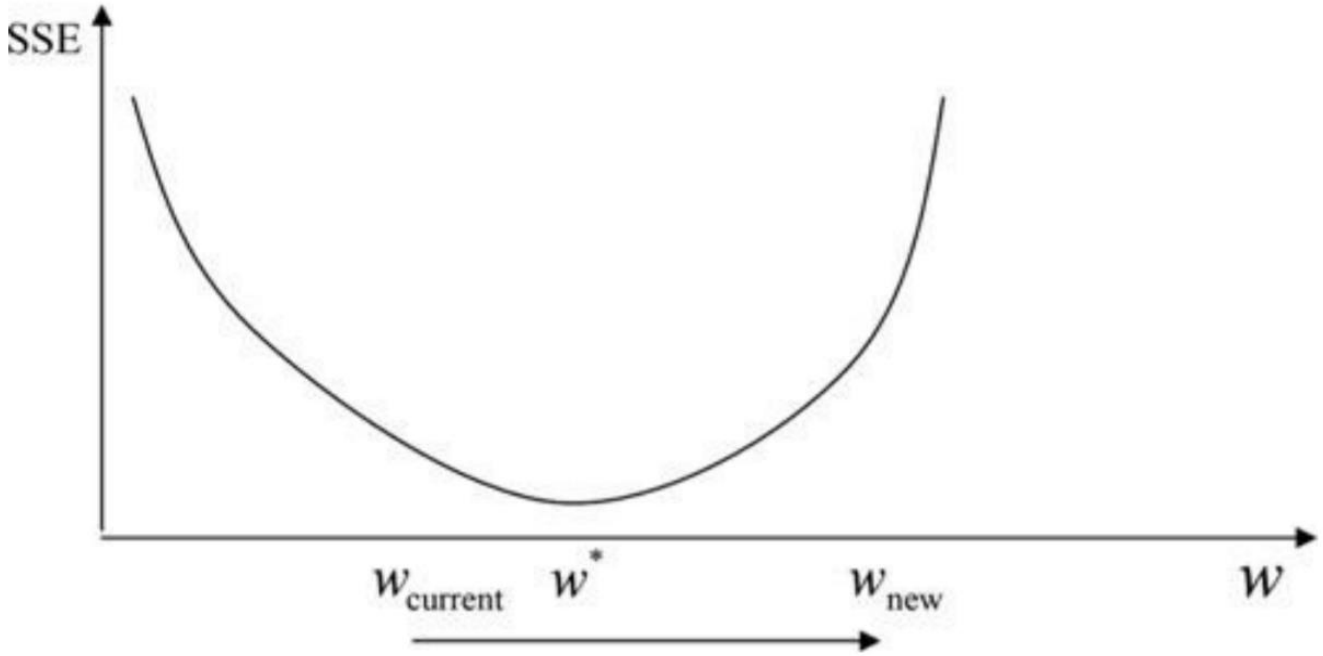


Figure 9.5 Large η may cause algorithm to overshoot global minimum.

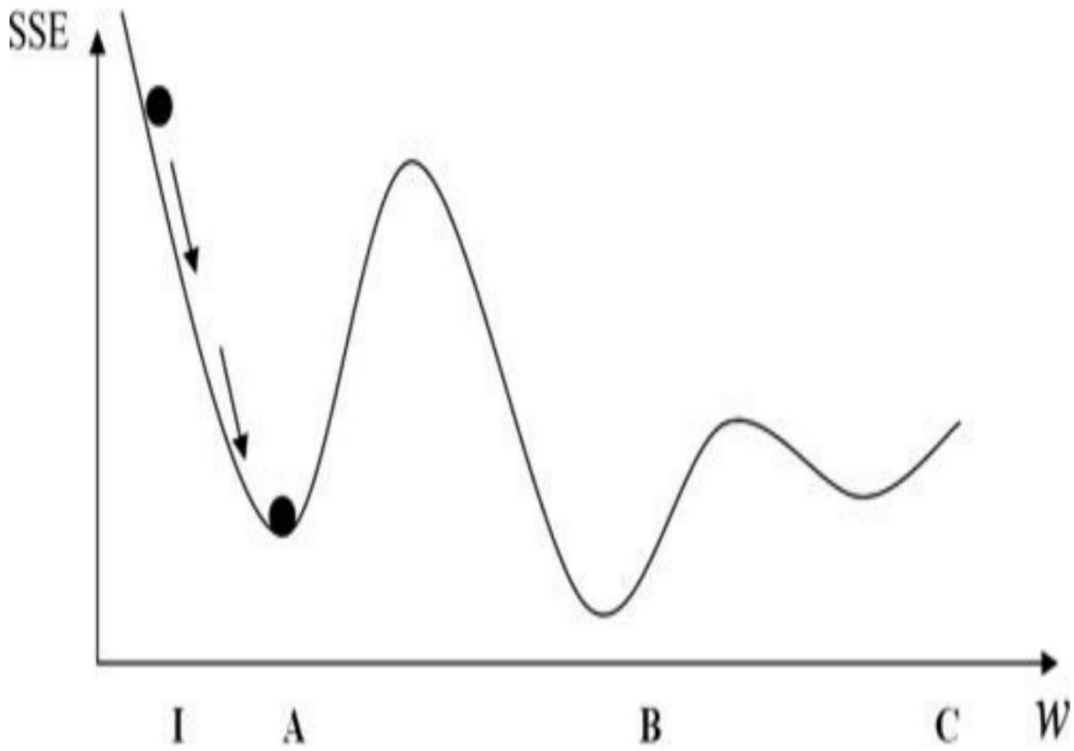


Figure 9.6 Small momentum α may cause algorithm to undershoot global minimum.