Artificial Intelligence

Unit 07

Artificial Neural Networks

By:

Syeda Saleha Raza

AL NAFI,
A company with a focus on education,
wellbeing and renewable energy.

اَللَّهُمَّ إِنِّي أَسْأَلُكَ عِلْمًا نَافِعًا،

وَرِزْقًا طَيِّبًا، وَعَمَلًا مُتَقَبَّلًا،

(O Allah, I ask You for beneficial knowledge,
goodly provision and acceptable deeds)

اے اللہ ، میں آپ سے سوال کرتی ہوں نفع بخش علم کا، طیّب رزق کا، اور اس عمل کا

(Sunan Ibn Majah: 925)

Outline

- Artificial Neural Networks
    - Inspiration
    - Architecture of ANN
    - Learning ANN
    - Strength and Weaknesses
    - Code Demo

Artificial Neural Networks (ANN)

Artificial Neural Networks (ANN)

- Neural networks are biologically motivated computing structures that are conceptually modeled after the brain.

How Our Brain Works?

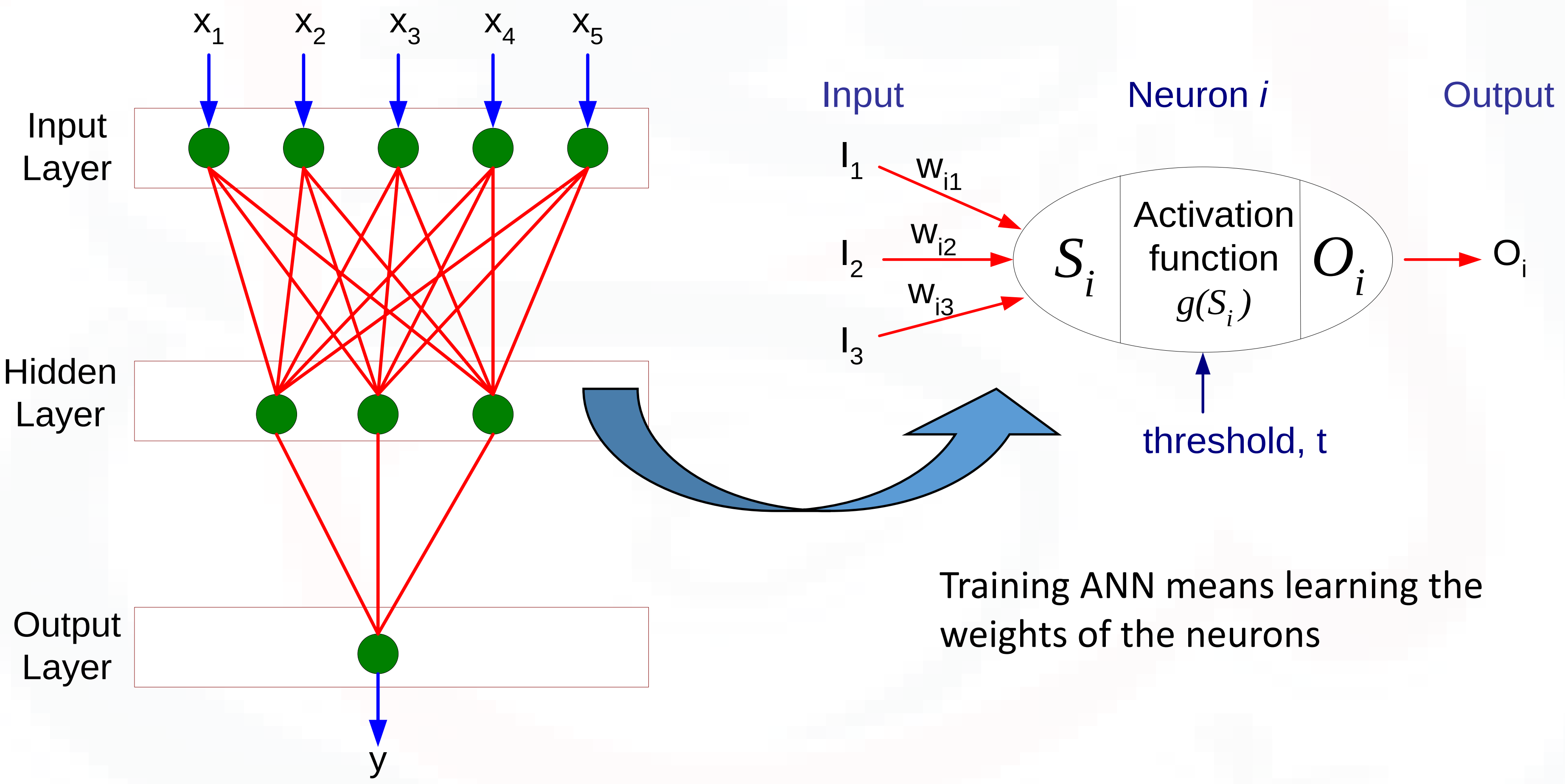
- The neural network is made up of a highly connected network of individual computing elements (mimicking neurons) that collectively can be used to solve interesting and difficult problems.

Artificial Neural Network - Introduced

Multilayer Feed-Forward Networks

- Multilayer feed-forward networks are one of the most important and most popular classes of ANNs in real-world applications.
- They are commonly referred to as multilayer perceptrons, which represent a generalization of the simple perceptron.

General Structure of MLP

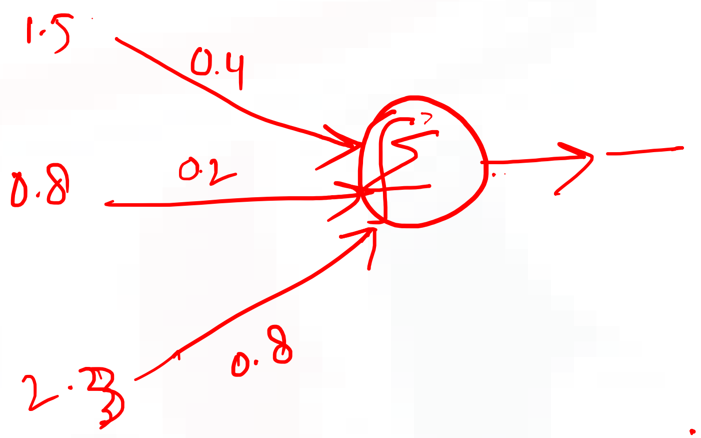


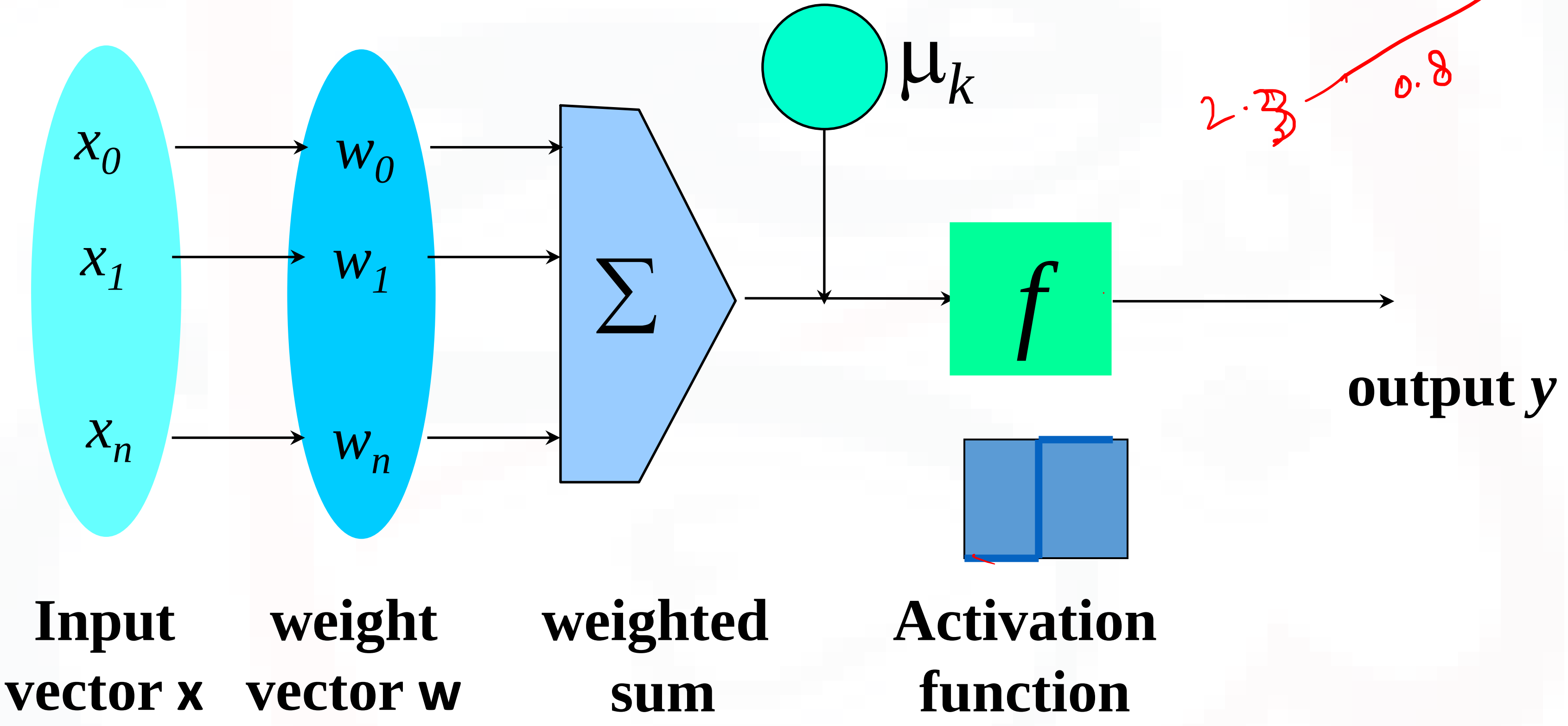
Training ANN means learning the weights of the neurons

Artificial Neuron

- An artificial neuron is an information-processing unit that is fundamental to the operation of an ANN. It consists of three basic elements:
  - A set of connecting links from different inputs, each of which is characterized by a weight or strength.
  - An adder for summing the input signals weighted by the respective synaptic strengths.
  - An activation function for limiting the amplitude of the output of a neuron.

A Neuron (= a perceptron)





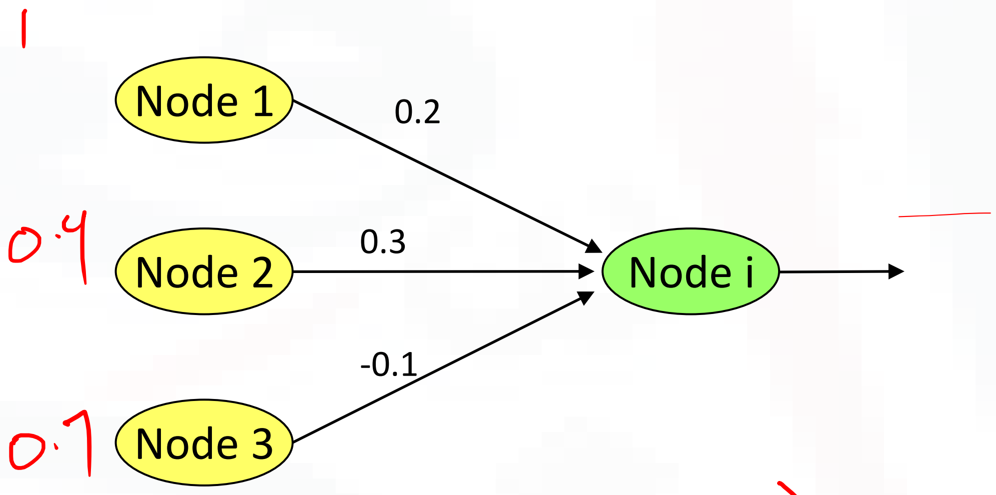
Sigmoid

$$\frac{1}{1+e^{-x}}$$

- The n-dimensional input vector **x** is mapped into variable y by means of the scalar product and a nonlinear function mapping

Feed-forwarding Data

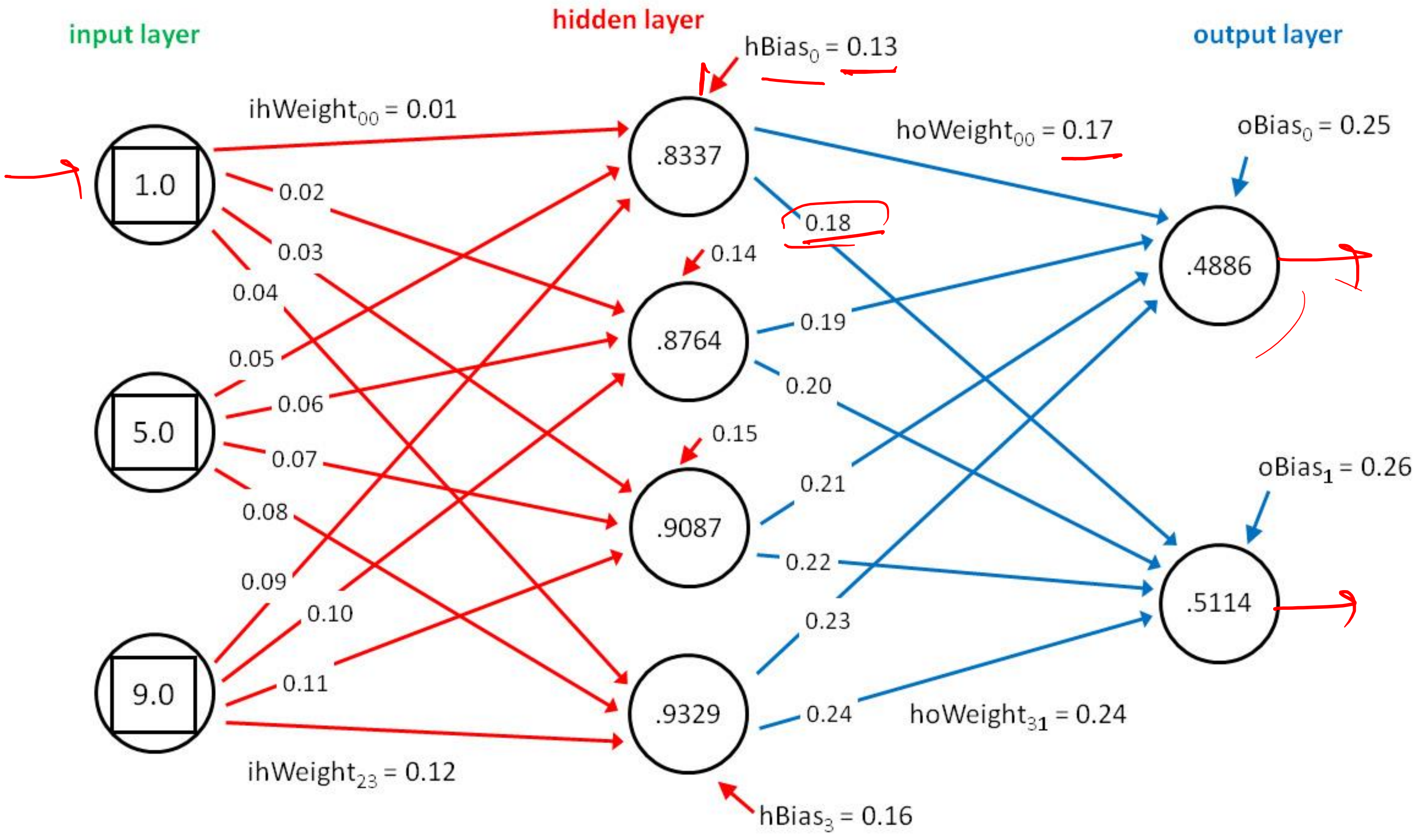
- Input = {1.0, 0.4, 0.7}
- Output of Node I = ??



$$f\left((1\times 0.2)+(0.4\times 0.3)+(0.7\times -0.1)\right)$$

$$\frac{1}{1+e^{-x}} =$$

Learning a Neural Network



https://visualstudiomagazine.com/articles/2014/11/01/use-python-with-your-neural-networks.aspx

Topology of ANN

Specification of ANN

- The number of input attributes found within individual instances determines the number of input layer nodes.
- The user specifies the number of hidden layers as well as the number of nodes within a specific hidden layer.

Venue	Type of Wicket	Type of match	Batted first	Winning Team
Pakistan	Normal	T20	Pakistan	Pakistan
India	Fast	Test	Pakistan	Pakistan
India	Slow	ODI	India	India
Pakistan	Slow	ODI	Pakistan	India
Third country	Fast	ODI	India	Pakistan
India	Normal	ODI	India	India
Pakistan	Normal	T20	India	Pakistan
Third country	Fast	Test	Pakistan	India
Third country	Slow	Test	India	Pakistan
Third country	Slow	ODI	Pakistan	Pakistan
Pakistan	Normal	T20	Pakistan	India
Third country	Slow	Test	Pakistan	Pakistan
Pakistan	Fast	ODI	India	Pakistan
Third country	Fast	Test	Pakistan	India
India	Slow	ODI	Pakistan	???

3

Input Format

- The input to individual neural network nodes should be numeric and fall in the closed interval range [0,1].
- We need a way to numerically represent categorical data.
  - Attribute Color: {Red, Green, Blue, Yellow}
- We also need a conversion method for numerical data falling outside the [0,1] range.
  - Values: 100, 200, 300, 400

One-hot Encoding

Color
Red
Red
Yellow
Green
Yellow

Red	Yellow	Green
1	0	0
1	0	0
0	1	0
0	0	1

Topology of NN?

- How many neurons are required in the input layer?

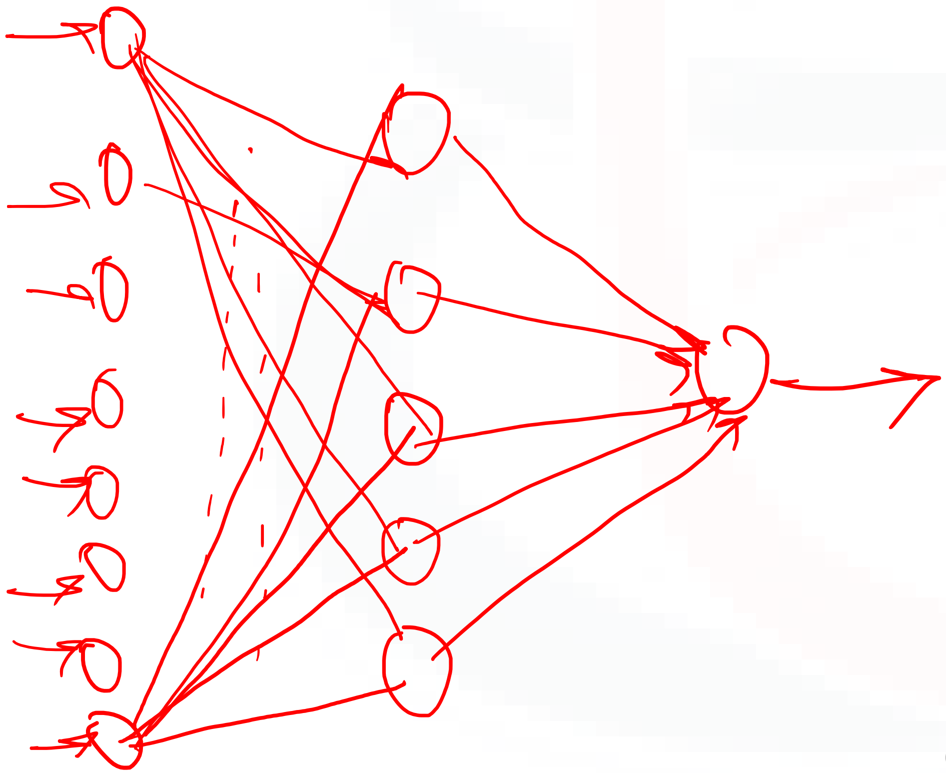
Name	Give Birth	Can Fly	Live in Water	Have Legs	Class
human	yes	no	no	yes	mammals
python	no	no	no	no	non-mammals
salmon	no	no	yes	no	non-mammals
whale	yes	no	yes	no	mammals
frog	no	no	sometimes	yes	non-mammals
komodo	no	no	no	yes	non-mammals
bat	yes	yes	no	yes	mammals
pigeon	no	yes	no	yes	non-mammals
cat	yes	no	no	yes	mammals
leopard shark	yes	no	yes	no	non-mammals
turtle	no	no	sometimes	yes	non-mammals
penguin	no	no	sometimes	yes	non-mammals
porcupine	yes	no	no	yes	mammals
eel	no	no	yes	no	non-mammals
salamander	no	no	sometimes	yes	non-mammals
gila monster	no	no	no	yes	non-mammals
platypus	no	no	no	yes	mammals
owl	no	yes	no	yes	non-mammals
dolphin	yes	no	yes	no	mammals
eagle	no	yes	no	yes	non-mammals

Topology of NN?

- How many neurons are required in the input layer?

Outlook	Temperature	Humidity	Windy	Class
sunny	hot	high	false	N
sunny	hot	high	true	N
overcast	hot	high	false	P
rain	mild	high	false	P
rain	cool	normal	false	P
rain	cool	normal	true	N
overcast	cool	normal	true	P
sunny	mild	high	false	N
sunny	cool	normal	false	P
rain	mild	normal	false	P
sunny	mild	normal	true	P
overcast	mild	high	true	P
overcast	hot	normal	false	P
rain	mild	high	true	N

3 3 1 1 1



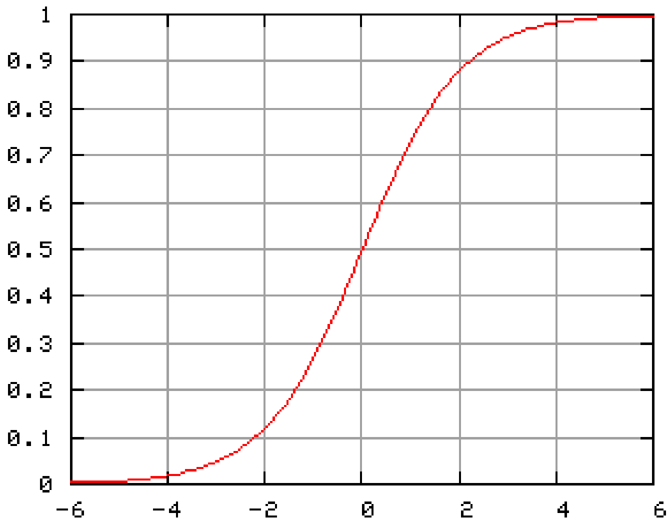
Output Format

- The nodes of the input layer pass input attribute values to the hidden layer unchanged.
- A hidden or output layer node takes input from the connected nodes of the previous layer, combines the previous layer node values into a single value, and uses the new value as input to an evaluation function.
- The output of the evaluation function is a number in the closed interval [0, 1].

Activation Function

Sigmoid Function

- The first criterion of an evaluation function is that the function must output values in the [0, 1] interval range.
- A second criterion is that the function should output a value close to 1 when sufficiently excited.
- The sigmoid function meets both criterion and is often used for node evaluation.
  - $f(x) = 1 / (1 + e^{-x})$

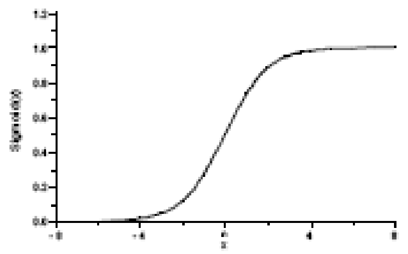


Transfer Functions

Sigmoid Functions These are smooth (differentiable) and monotonically increasing.

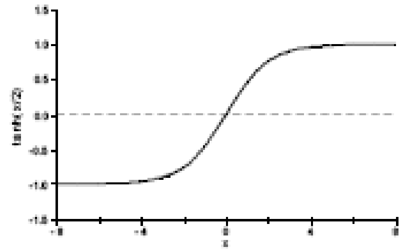
The logistic function

$$\text{Sigmoid}(x) = \frac{1}{1+e^{-x}}$$



Hyperbolic tangent

$$\tanh\left(\tfrac{x}{2}\right) = \frac{1-e^{-x}}{1+e^{-x}}$$

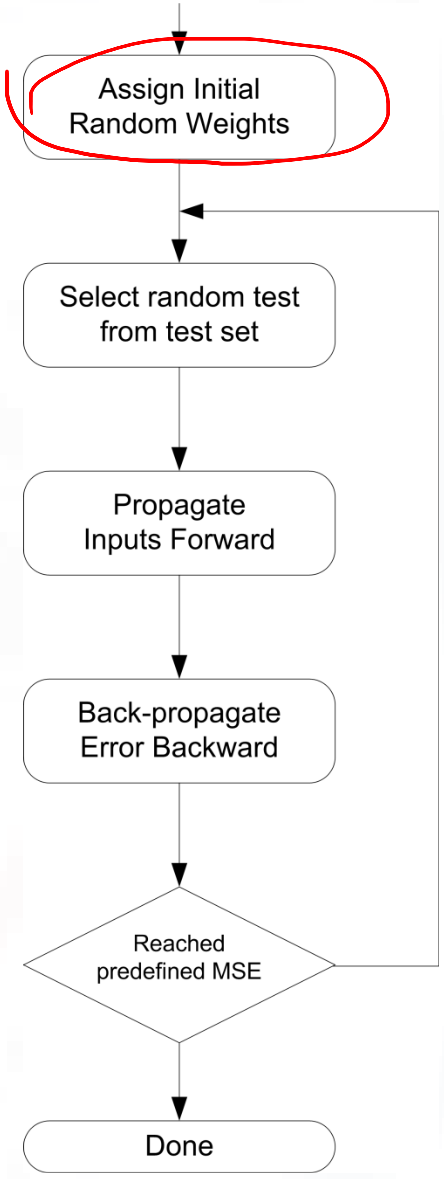


Learning of ANN

Backpropagation

Learning of ANN

- Learning is accomplished by modifying network connection weights while a set of input instances is repeatedly passed through the network.
- Once trained, an unknown instance passing through the network is classified according to the value(s) seen at the output layer.



Feed-forwarding Data

w_{1i}=0.20, w_{1j}=0.10, w_{2i}=0.30, w_{2j}=-0.10, w_{3i}=-0.10, w_{3j}=0.20, w_{ik}=0.10, w_{jk}=0.50, T=0.65

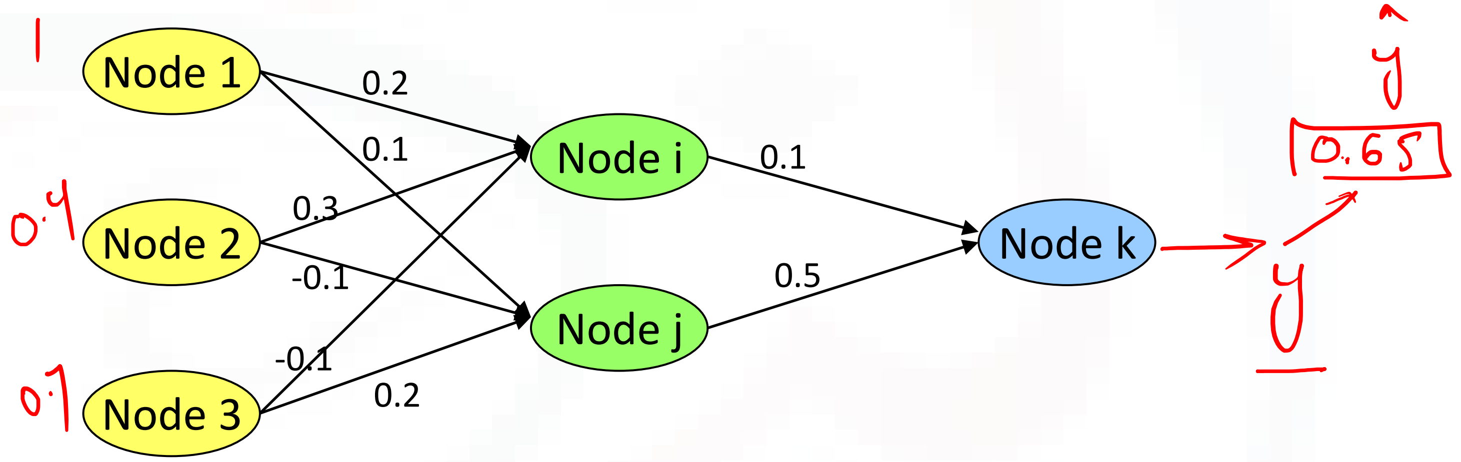
- Input = {1.0, 0.4, 0.7}
- Input to node i = 0.2x1.0 + 0.3x0.4 - 0.1x0.7 = 0.25
- Now apply the sigmoid function: = 0.562

ε

- Input to node j = ?
- Output of node j = ?

- Input to node k = ?
- Output of node k = ?

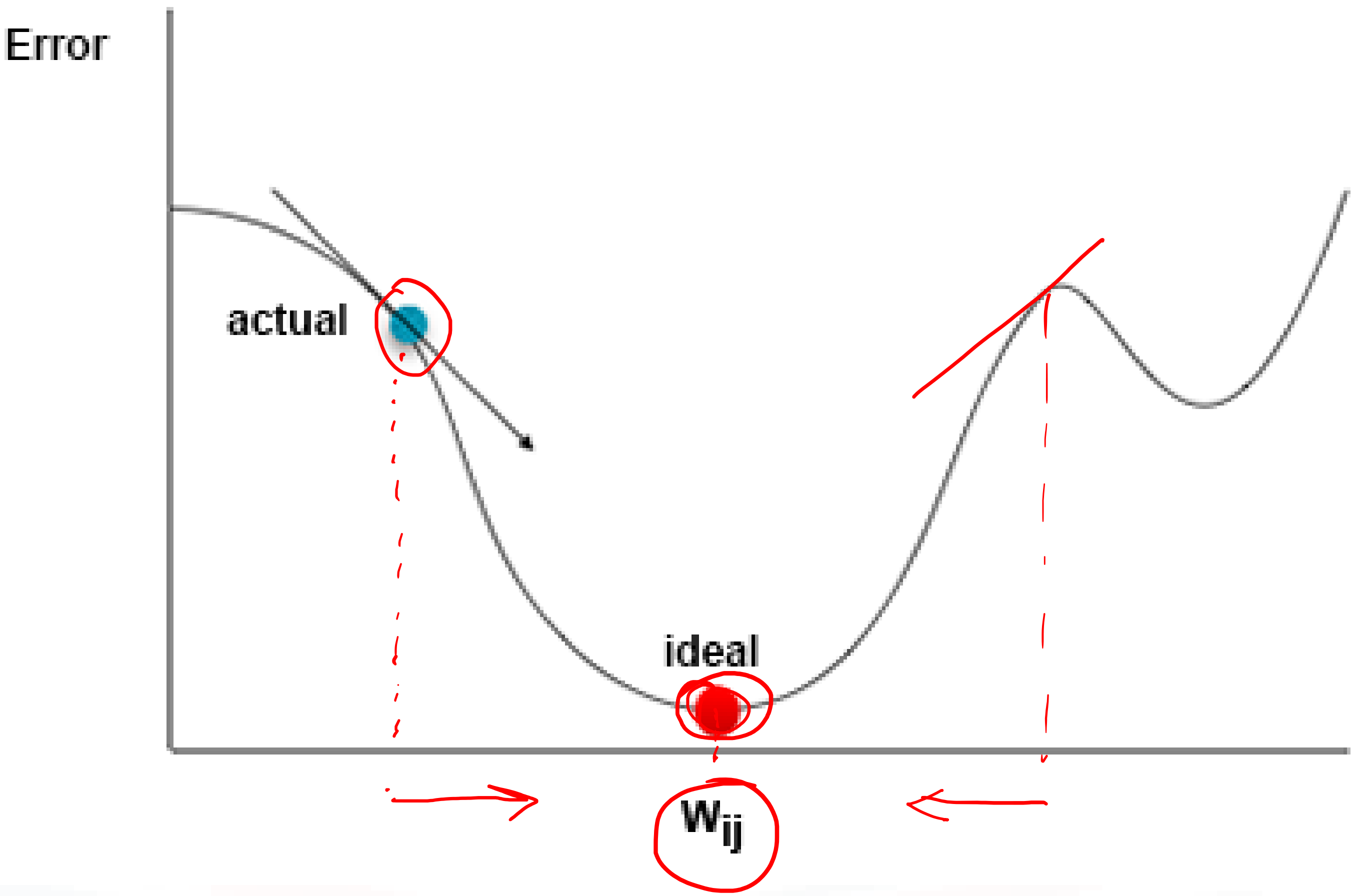
- Input node k =?
- Output of node k =?



Gradient Descent

- At a theoretical level, gradient descent is an algorithm that minimizes functions. Given a function defined by a set of parameters, gradient descent starts with an initial set of parameter values and iteratively moves toward a set of parameter values that minimize the function. This iterative minimization is achieved using calculus, taking steps in the negative direction of the function gradient.

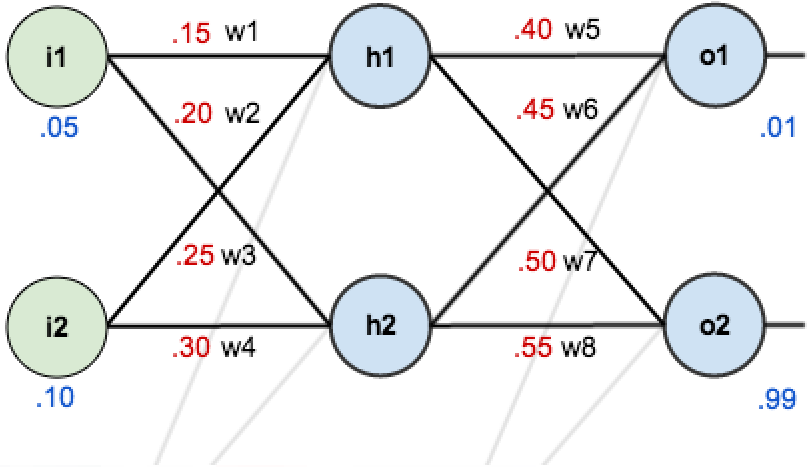
Gradient

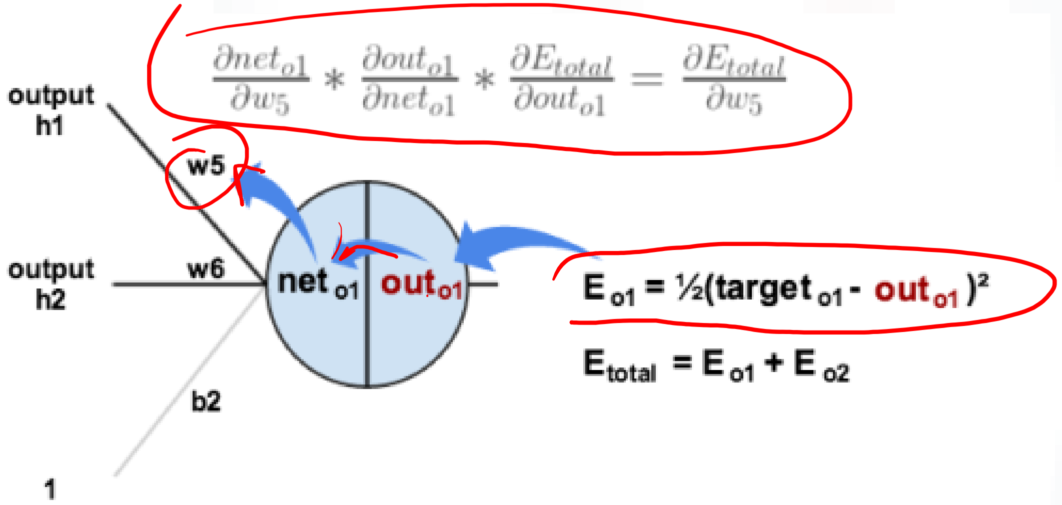


Gradient

- The gradient of each weight gives an indication about how to modify each weight to achieve the expected output (or reduce the error).
- Each weight has a gradient that is slope of the error function.
    - Zero gradient implies theta the weight is not contributing to the error
    - Negative gradient implies that the weight should be increased to achieve a lower error
    - Positive gradient implies that the weight should be decreased to achieve a lower error

Backward Pass





- https://mattmazur.com/2015/03/17/a-step-by-step-backpropagation-example/

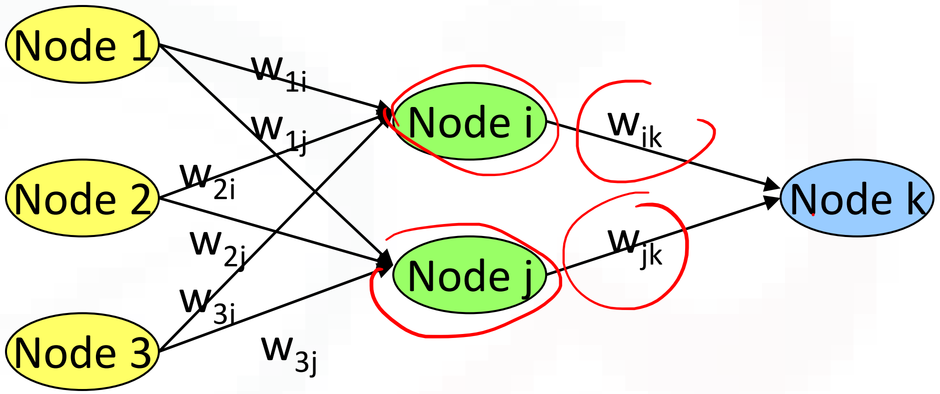
Explanation of the Backpropagation Algorithm

w_{1i}=0.20, w_{1j}=0.10, w_{2i}=0.30, w_{2j}=-0.10, w_{3i}=-0.10, w_{3j}=0.20, w_{ik}=0.10, w_{jk}=0.50, T=0.65

- Input = {1.0, 0.4, 0.7}
- Input to node i = 0.2x1.0 + 0.3x0.4 - 0.1x0.7 = 0.25
- Now apply the sigmoid function: f(0.25) = 0.562

epoch

- Output of node i = 0.562
- Output of node j = 0.549
- Output of node k = 0.58



- Error(k) = $(T – O_k)\ O_k\ (1 – O_k)$
  - T = the target output
  - O_k = the computed output at node k
- Error(k) = (0.65 – 0.58)0.58(1-0.58) = 0.017

Explanation of the Backpropagation Algorithm

w_{1i}=0.20, w_{1j}=0.10, w_{2i}=0.30, w_{2j}=-0.10, w_{3i}=-0.10, w_{3j}=0.20, w_{ik}=0.10, w_{jk}=0.50

- Error(i) = Error(k) w_{ik} O_i (1 – O_i)

  = 0.017 * 0.1 * 0.562* (1-0.56)

  = 0.0004
- Error(j) = 0.017 * 0.5 * 0.549* (1-0.549)

  =0.02
- The next step is to update the weights associated with the individual node connections.
- Weight adjustments are made using the delta rule
  - To minimize the sum of the square errors, where error is defined as the distance between computed and actual output

Explanation of the Backpropagation Algorithm

w_{1i}=0.20, w_{1j}=0.10, w_{2i}=0.30, w_{2j}=-0.10, w_{3i}=-0.10, w_{3j}=0.20, w_{ik}=0.10, w_{jk}=0.50

- $w_{ik} = w_{ik}$ (current) + Δw_{ik}

- Δw_{ik} = r x Error(k) x O_i
  - where r is learning rate parameter, $0 < r < 1$

$w_{ik} = 0.1 + 0.8 * 0.017 * 0.562$
$= 0.107$

$w_{jk} = 0.5 + 0.8 * 0.017 * 0.549$
$= 0.507$

- Compute: Δw_{ik} Δw_{1i} Δw_{2i} Δw_{3i}

Algorithm

- Initialize the network:
  - Create the network topology by choosing the number of nodes for the input, hidden, and output layers.
  - Initialize weights for all node connections to arbitrary values between -1.0 and 1.0.
  - Choose a value between 0 and 1 for the learning parameter.
  - Choose a terminating condition.
- For all the training instances:
  - Feed the training instance through the network.
  - Determine the output error.
  - Updated the network weights.
- If the terminating condition has not been met, repeat step 2.
- Test the accuracy of the network on a test dataset. If the accuracy is less than optimal, change one or more parameters of the network topology and start over.

Training/Testing of ANN

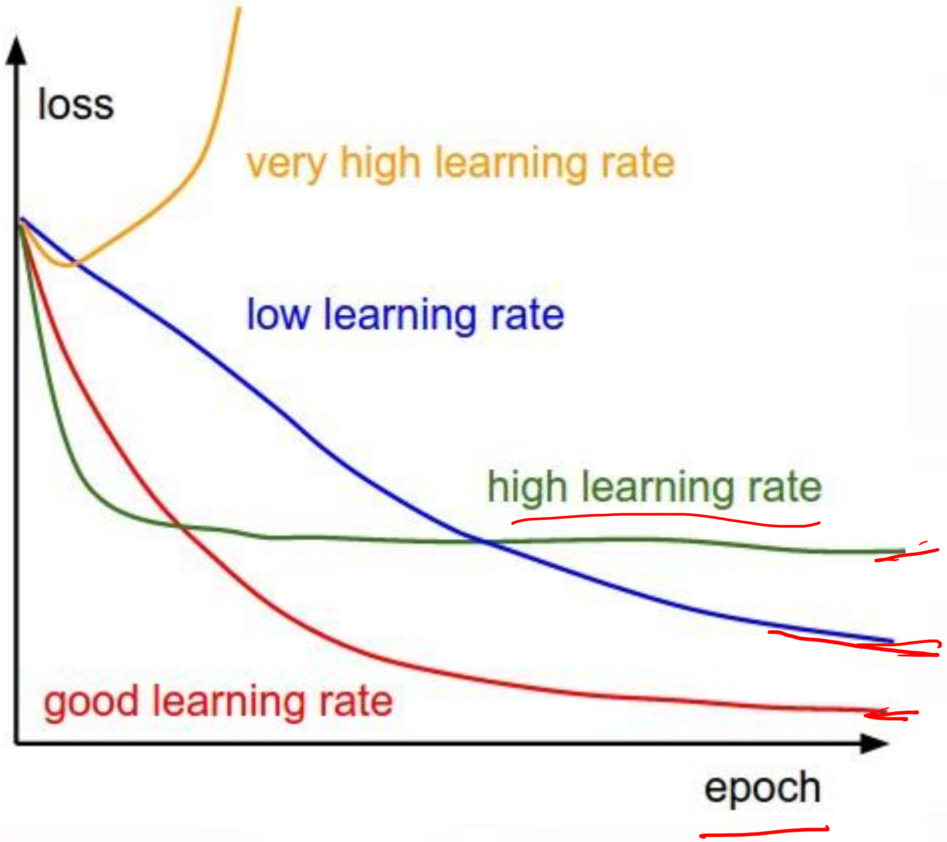
- During the training phase, training instances are repeatedly passed through the network while individual weight values are modified.
- The purpose of changing the connection weights is to minimize training set error rate.
- Network training continues until a specific terminating condition is satisfied.
- The terminating condition can be convergence of the network to a minimum total error value, a specific time criterion, or a maximum number of iterations.

More Parameters

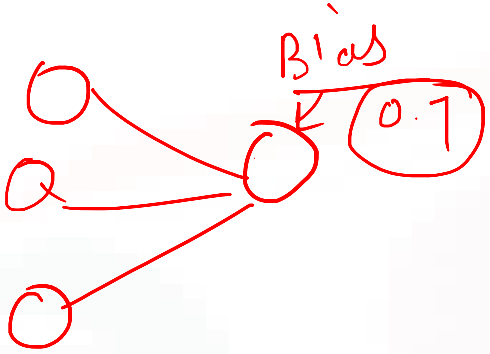
Learning Rate

0.8

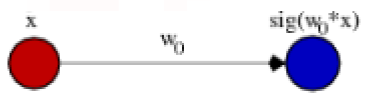
0 – 1

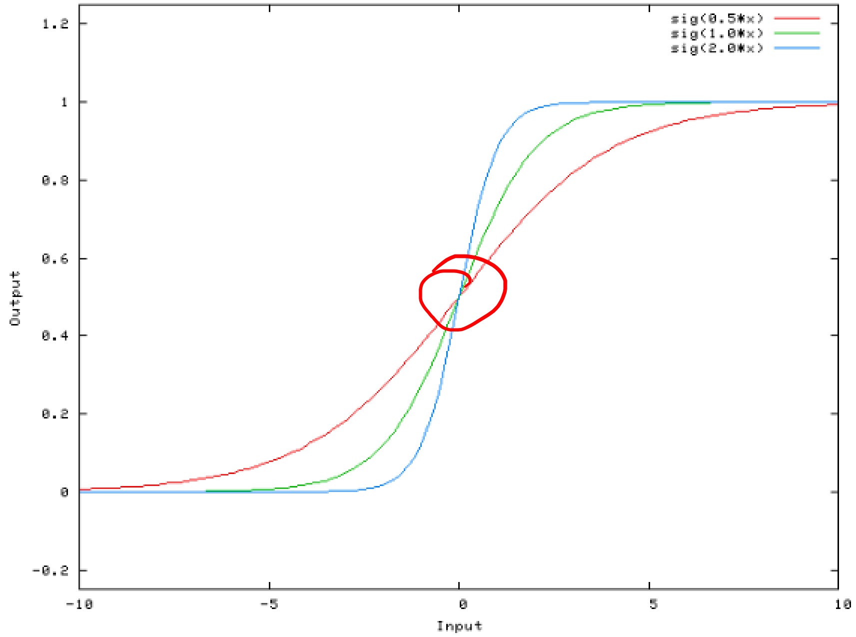


Bias in ANN

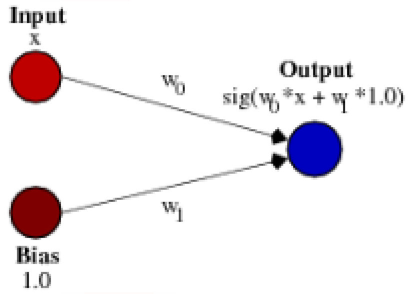


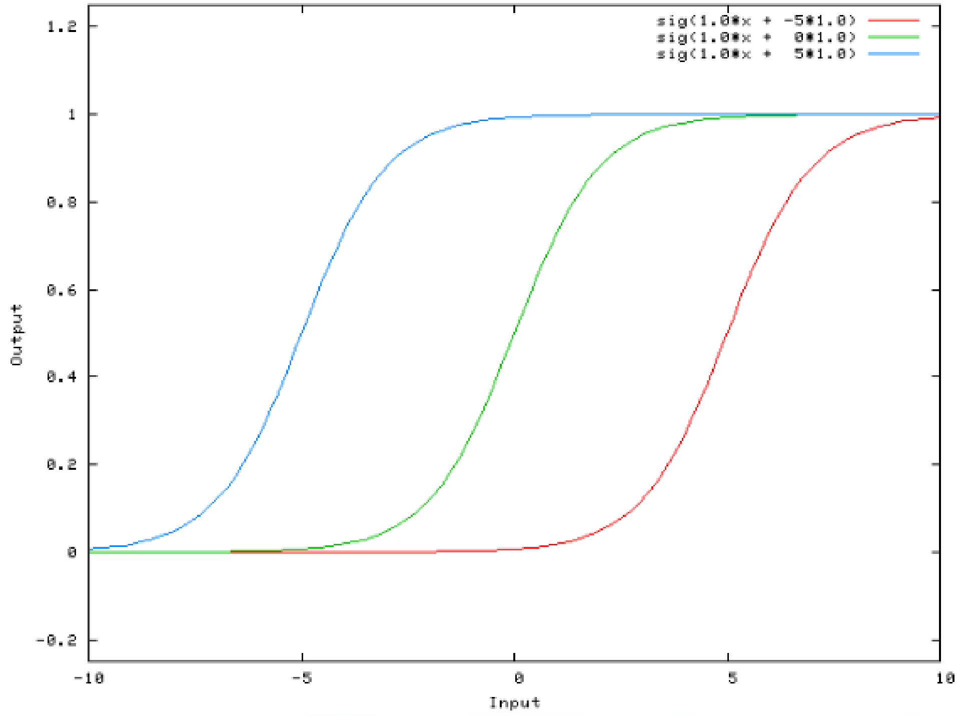
- **A bias value allows you to shift the activation function to the left or right**, which may be critical for successful learning.





Bias in ANN


Input
x
w0
Output
sig(w0*x + w1*1.0)
w1
Bias
1.0


sig(1.0*x + -5*1.0)
sig(1.0*x + 0*1.0)
sig(1.0*x + 5*1.0)
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
Output
-10
-5
0
5
10
Input

General Considerations

- What input attributes will be used to build the network?
- How will the network output be represented?
- How many hidden layers should the network contain?
- How many nodes should there be in each hidden layer?
- What conditions will terminate network training?

Limitations

- The biggest criticism of neural networks is that they lack the ability to explain their behavior.
- Neural networks can easily be over trained to the point of working well on the training data but poorly on test data.
- They are very resource intensive and requires a lot of training data and computational power for training.

Code Walkthrough
Using ANN for text classification

Resources

- https://www.coursera.org/learn/machine-learning/lecture/du981/backpropagation-intuition
- https://mattmazur.com/2015/03/17/a-step-by-step-backpropagation-example/
- https://scikit-neuralnetwork.readthedocs.io/en/latest/index.html
- 7_NB.pdf (stanford.edu)

جزاك الله

To ask questions, Please use communities link (for respective course) within portal

https://portal.alnafi.com/enrollments