Introduction to Artificial Intelligence

Unit # 8

Acknowledgement

- The slides of this lecture have been taken from the lecture slides of CS307 – "Introduction to Artificial Intelligence" and CSE652 – "Knowledge Discovery and Data mining" by Dr. Sajjad Haider.

Popular Machine Learning Techniques

- Classification
  - Classification Trees ✔
  - Naïve Bayes ✔
  - **Neural Networks**
- Clustering
  - K-Means

- In this course, the focus is on the classification techniques

How to Estimated Classification Accuracy or Error Rates

- Partition: Training-and-testing
  - use two independent data sets, e.g., training set (2/3), test set(1/3)
  - used for data set with large number of exmples
- Cross-validation
  - divide the data set into *k* subsamples
  - use *k-1* subsamples as training data and one sub-sample as test data—*k*-fold cross-validation
  - for data set with moderate size

Decision Tree Based Classification

- Advantages:
  - Inexpensive to construct
  - Extremely fast at classifying unknown records
  - Easy to interpret for small-sized trees
  - Accuracy is comparable to other classification techniques for many simple data sets

Metrics for Performance Evaluation

- Focus on the predictive capability of a model
  - Rather than how fast it takes to classify or build models, scalability, etc.
- Confusion Matrix:

		PREDICTED CLASS	
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	a	b
	Class=No	c	d

a: TP (true positive)
b: FN (false negative)
c: FP (false positive)
d: TN (true negative)

Metrics for Performance Evaluation…

		PREDICTED CLASS	
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	a (TP)	b (FN)
	Class=No	c (FP)	d (TN)

- Most widely-used metric:

$$\text{Accuracy} = \frac{a+d}{a+b+c+d} = \frac{TP+TN}{TP+TN+FP+FN}$$

Limitation of Accuracy

- Consider a 2-class problem
  - Number of Class 0 examples = 9990
  - Number of Class 1 examples = 10
- If model predicts everything to be class 0, accuracy is 9990/10000 = 99.9 %
  - Accuracy is misleading because model does not detect any class 1 example

Cost Matrix

	PREDICTED CLASS		
ACTUAL CLASS	C(i|j)	**Class=Yes**	**Class=No**
	Class=Yes	C(Yes|Yes)	C(No|Yes)
	Class=No	C(Yes|No)	C(No|No)

C(i|j): Cost of misclassifying class j example as class i

Cost Matrix (Cont'd)

	PREDICTED CLASS		
ACTUAL CLASS		True	False
	True	10	5
	False	1	14

	PREDICTED CLASS		
ACTUAL CLASS		True	False
	True	10	3
	False	3	14

	PREDICTED CLASS		
ACTUAL CLASS		True	False
	True	10	6
	False	0	14

All three confusion matrices have the same accuracy value, i.e., **24 / 30**

What if the cost of misclassification is not the same for both type of errors?

Cost Matrix (Cont'd)

	PREDICTED CLASS		
ACTUAL CLASS		True	False
	True	10	5x5
	False	1	14

	PREDICTED CLASS		
ACTUAL CLASS		True	False
	True	10	3x5
	False	3	14

	PREDICTED CLASS		
ACTUAL CLASS		True	False
	True	10	6x5
	False	0	14

Suppose the cost of misclassifying True as False is 5 while the cost of misclassifying False as True is 1.

Accuracy values are:
24/50, 24/42, 24/54

Cost Matrix (Cont'd)

	PREDICTED CLASS		
ACTUAL CLASS		True	False
	True	10	5x4
	False	1	14

	PREDICTED CLASS		
ACTUAL CLASS		True	False
	True	10	3x4
	False	3	14

	PREDICTED CLASS		
ACTUAL CLASS		True	False
	True	10	6x4
	False	0	14

Suppose the cost of misclassifying True as False is **4** while the cost of misclassifying False as True is 1.

Accuracy values are:
24/45, 24/39, 24/48

Cost-Sensitive Measures

$$\text{Precision (p)} = \frac{a}{a+c}$$

$$\text{Recall (r)} = \frac{a}{a+b}$$

$$\text{F-measure (F)} = \frac{2rp}{r+p} = \frac{2a}{2a+b+c}$$

- Precision is biased towards C(Yes|Yes) & C(Yes|No)
- Recall is biased towards C(Yes|Yes) & C(No|Yes)
- F-measure is biased towards all except C(No|No)

$$\text{Weighted Accuracy} = \frac{w_1 a + w_4 d}{w_1 a + w_2 b + w_3 c + w_4 d}$$

Recall and Precision

Actual	Prediction
T	T
T	F
F	T
F	F
F	T
T	T
T	T
T	F
F	T
T	T

Recall and Precision

Actual	Prediction
T	T
T	F
F	T
F	F
F	T
T	T
T	T
T	F
F	T
T	T

- Recall = 4 / 6

Recall and Precision

Actual	Prediction
T	T
T	F
F	T
F	F
F	T
T	T
T	T
T	F
F	T
T	T

- Recall = 4 / 6
- Precision = 4 / 7
- F-Measure = 8 / 13

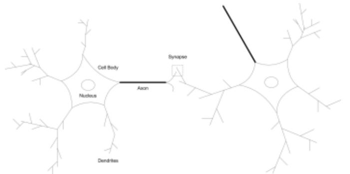
Artificial Neural Networks

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

How Our Brain Works?

- While neural networks are modeled after our understanding of the way in which our brain works, surprisingly little is known about how our brains actually function.
- Through various types of inspection, we can see our brain in operation, but because of the massive number of neurons and interconnections between these neurons, how it works remains a mystery (though many theories exist).

Neurons Inside Our Body



- We are born with about 100 billion neurons
- A neuron may connect to as many as 100,000 other neurons
- Signals “move” via electrochemical signals
- The synapses release a chemical transmitter – the sum of which can cause a threshold to be reached – causing the neuron to “fire”

Definition of Neurons (from Wikipedia)

- **Neurons** are responsive cells in the nervous system that process and transmit information by chemical signals within the neuron.
- A number of different types of neurons exist: sensory neurons respond to touch, sound, light and numerous other stimuli affecting cells of the sensory organs that then send signals to the spinal cord and brain.
- Motor neurons receive signals from the brain and spinal cord and cause muscle contractions and affect glands.
- Inter-neurons connect neurons to other neurons within the brain and spinal cord.
- **Neurons respond to stimulus and communicate the presence of that stimuli to the central nervous system, which processes that information and sends responses to other parts of the body for action.**

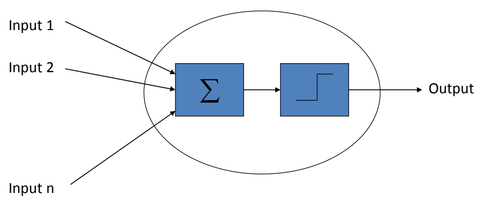
Birth of Artificial Neural Networks

- The story of neural networks is interesting because, like AI itself, it's one of grand visions, eventual disappointment, and finally, silent adoption.
- In 1943, McCulloch and Pitts developed a neural network model based on their understanding of neurology, but the models were typically limited to formal logic simulations (simulating binary operations).

Artificial Neuron

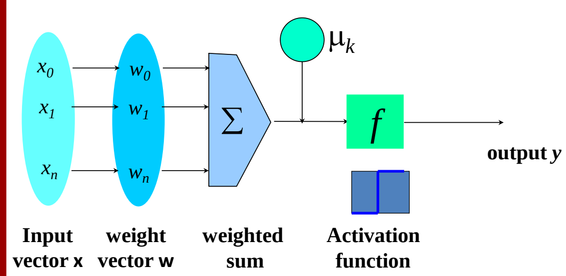
- An artificial neuron is an information-processing unit that is fundamental to the operation of an ANN. It consists of three basic elements:
  - A set of connecting links from different inputs, each of which is characterized by a weight or strength. In general, the weights of an artificial neuron may lie in a range that includes negative as well as positive values.
  - 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.

McCulloch-Pitts Neuron



- The basic idea was proposed in 1943.
- A set of synapses (connections) brings in activations from other neurons.
- A processing unit sums the inputs, and then applies a non-linear activation function.
- An output line transmits the results to other neurons.

A Neuron (= a perceptron)



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

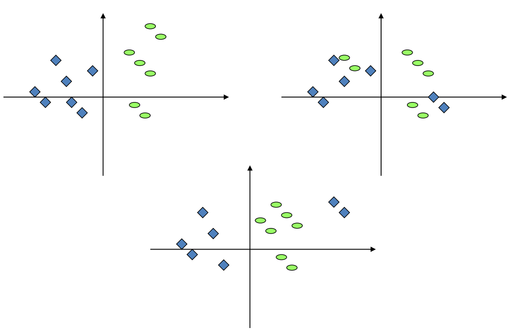
Dark Period

- In 1969, the growing popularity of neural networks was brought to a halt.
- Marvin Minsky and Seymour Papert wrote a book entitled "Perceptrons" in which limitations of single-layer perceptrons were discussed.
- The result was severe reductions in neural network research funding, and a corresponding reduction in the effort applied to the field.

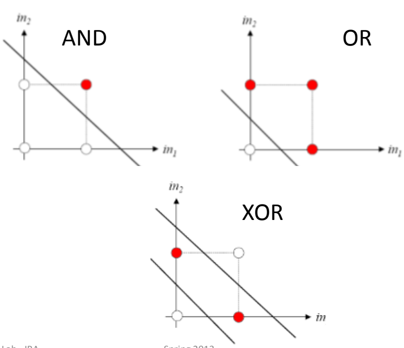
Revival

- In 1974, Paul Werbos developed the **backpropagation** algorithm, which permitted successful learning in multilayer neural networks.
- Since the 1970s, research and successful results in neural network design have attracted scientists back to the field.

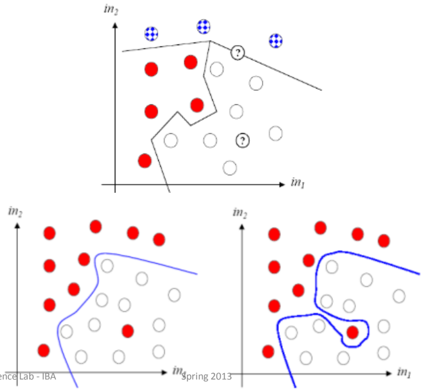
Linearly Separable Data



AND, OR, XOR



Decision Boundaries



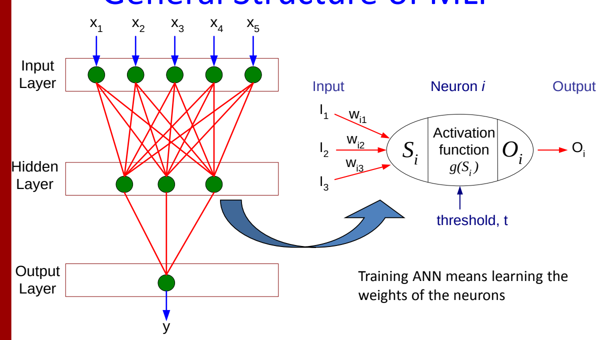
Multilayer Feed-Forward Networks

- 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.

Characteristics of MLP

- A multilayer perceptron has three distinctive characteristics
  – The model of each neuron in the network includes usually a nonlinear activation function, sigmoid or hyperbolic.
  – The network contains one or more layer of hidden neurons that are not a part of the input or output of the network. These hidden nodes enable the network to learn complex and highly nonlinear tasks by extracting progressively more meaningful features from the input patterns.
  – The network exhibits a high degree of connectivity from one layer to the next one.

General Structure of MLP



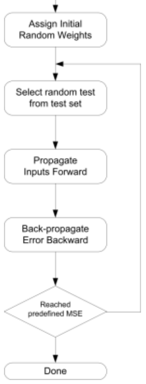
Training ANN means learning the weights of the neurons

How A Multi-Layer Neural Network Works?

- The **inputs** to the network correspond to the attributes measured for each training tuple.
- Inputs are fed simultaneously into the units making up the **input layer.**
- They are then weighted and fed simultaneously to a **hidden layer.** The number of hidden layers is arbitrary, although usually only one
- The weighted outputs of the last hidden layer are input to units making up the **output layer**, which emits the network's prediction
- From a statistical point of view, networks perform **nonlinear regression**: Given enough hidden units and enough training samples, they can closely approximate any function.

Working 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.



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.

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

Architecture 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

Architecture of NN?

- How many neurons are required in the input layer?

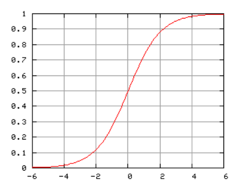
Outlook	Temperature	Humidity	Windy	Class
sunny	hot	high	false	N
sunny	hot	high	true	N
overcast	hot	high	false	P
rain	mild	high	false	P
rain	cool	normal	false	P
rain	cool	normal	true	N
overcast	cool	normal	true	P
sunny	mild	high	false	N
sunny	cool	normal	false	P
rain	mild	normal	false	P
sunny	mild	normal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

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].

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.
  - $f(x) = 1 / (1 + e^{-x})$

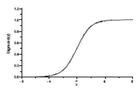


Transfer Functions

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

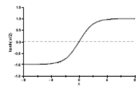
The logistic function

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

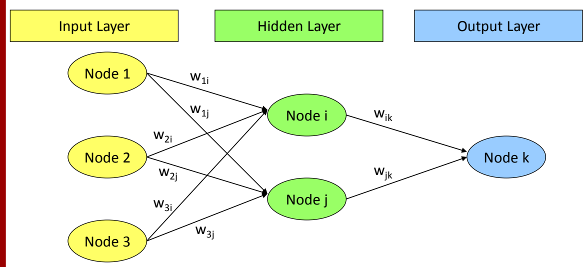


Hyperbolic tangent

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



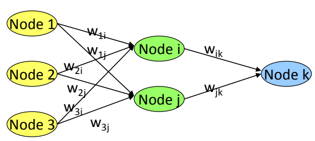
A Fully Connected Feed-Forward Network



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
- Input to node j = ?
- Input to node k = ?
- Error(k) = $(T - O_k)\, O_k\, (1 - O_k)$
  - T = the target output
  - O_k = the computed output at node k
- Error(k) = ?



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) $w_{ik}\, O_i\, (1 - O_i)$
  = ?
- Error(j) = ?
- 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

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

- $w_{ik} = w_{ik}\,(\text{current}) + \Delta w_{ik}$
- $\Delta w_{ik} = r \times \text{Error}(k) \times O_i$
  - where r is learning rate parameter, $0 < r < 1$
- 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.

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?

Weakness

- The biggest criticism of neural networks is that they lack the ability to explain their behavior.
- The algorithm is not guaranteed to converge to an optimal solution.
  - Manipulation of various learning parameter
- Neural networks can easily be over trained to the point of working well on the training data but poorly on test data.
  - Division of data into training and testing sets.