Artificial Intelligence

Lecture 25

Uncertainty

Prepared by:

Md. Mijanur Rahman, Prof. Dr.

Dept. of CSE, Jatiya Kabi Kazi Nazrul Islam University

Email: mijanjkkniu@gmail.com

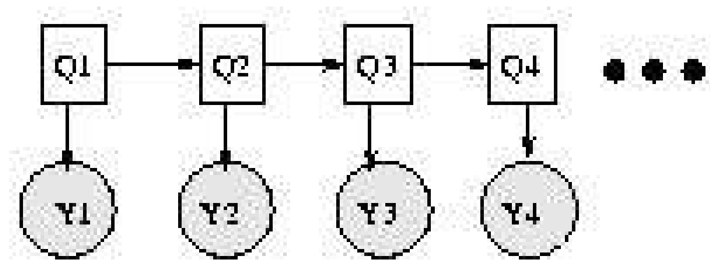
Outlines

- ...
- Methods to handle uncertainty
  - o Hidden Markov Model (HMM)
  - o Artificial Neural Network (ANN)

Hidden Markov Model (HMM)…

- Hidden Markov Model (HMM) is a statistical Markov model in which the system being modeled is assumed to be a Markov process, call it X, with unobservable ("*hidden*") states. HMM assumes that there is another process Y whose behavior "depends" on X. The goal is to learn about X by observing Y. HMM stipulates that, for each time instance n_0, the conditional probability distribution of Y_{n0} given the history $\{X_n = x_n\}_{n<=n0}$ must not depend on $\{x_n\}_{n<n0}$.

- A Markov model is a state transition diagram with probabilities on the edges
  - We use a Markov model to compute the probability of a certain sequence of states (see the figure)

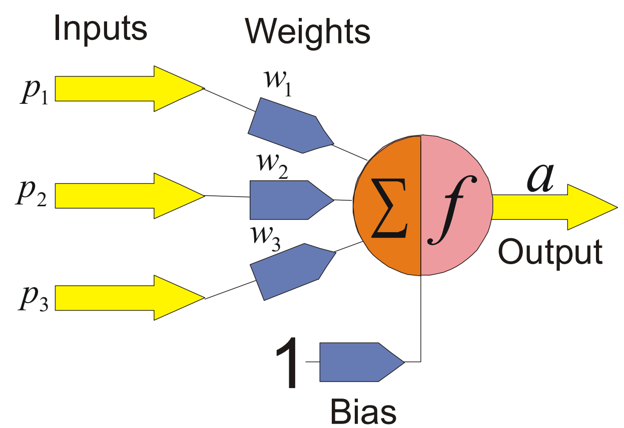


Hidden Markov Model (HMM)

- In many problems, we have observations to tell us what states have been reached, but observations may not show us all of the states
  - Intermediate states (those that are not identifiable from observations) are hidden
  - In the figure, the observations are Y1, Y2, Y3, Y4 and the hidden states are Q1, Q2, Q3, Q4
- The HMM allows us to compute the most probable path that led to a particular observable state
  - This allows us to find which hidden states were most likely to have occurred
- This is extremely useful for recognition problems where we know the end result but not how the end result was produced
  - we know the patient's symptoms but not the disease that caused the symptoms to appear
  - we know the speech signal that the speaker uttered, but not the phonemes that made up the speech signal

Artificial Neural Network (ANN)

- **Artificial neural networks** (**ANNs**), usually simply called **neural networks** (**NNs**), are computing systems inspired by the biological neural networks that constitute human brains.
- An ANN is based on a collection of connected units or nodes called artificial neurons. Each connection, like the synapses in a biological brain, can transmit a signal to other neurons.
- An artificial neuron that receives a signal then processes it and can transmit signal to the connected neurons.
- The "signal" at a connection is a real number, and the output of each neuron is computed by some non-linear function of the sum of its inputs.



$$a = f(p_1 w_1 + p_2 w_2 + p_3 w_3 + b) = f\left(\sum p_i w_i + b\right)$$

Reasons for Studying ANN…

- **Artificial neural networks are interesting to study for a number of reasons:**

1) As part of neuroscience, to understand real neural systems, researchers are simulating the neural systems of simple animals such as worms, which promises to lead to an understanding about which aspects of neural systems are necessary to explain the behavior of these animals.
2) Some researchers seek to automate not only the functionality of intelligence (which is what the field of artificial intelligence is about) but also the mechanism of the brain, suitably abstracted. One hypothesis is that the only way to build the functionality of the brain is by using the mechanism of the brain.

Reasons for Studying ANN

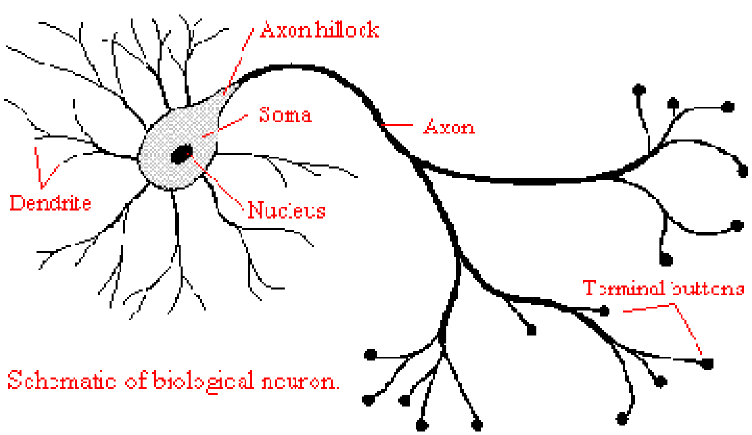
:

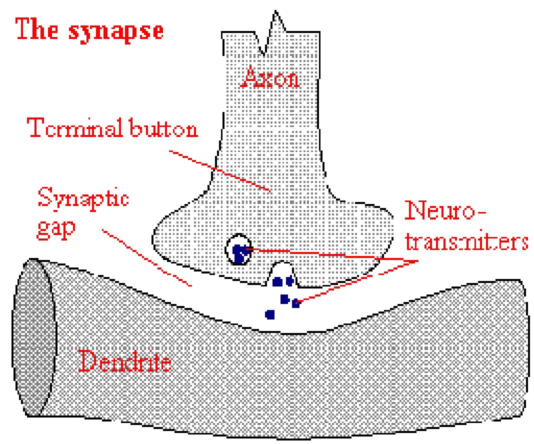
3) The brain inspires a new way to think about computation that contrasts with currently available computers. Unlike current computers, which have a few processors and a large but essentially inert memory, the brain consists of a huge number of asynchronous distributed processes, all running concurrently with no master controller.

4) As far as learning is concerned, neural networks provide a different measure of simplicity as a learning bias than, for example, decision trees. Multilayer neural networks, like decision trees, can represent any function of a set of discrete features.

Biological Neurons

- The brain is a collection of about 10 billion interconnected neurons. Each neuron is a cell that uses biochemical reactions to receive, process and transmit information.
- Each terminal button is connected to other neurons across a small gap called a synapse.
- A neuron's dendritic tree is connected to a thousand neighbouring neurons. When one of those neurons fire, a positive or negative charge is received by one of the dendrites. The strengths of all the received charges are added together through the processes of spatial and temporal summation.



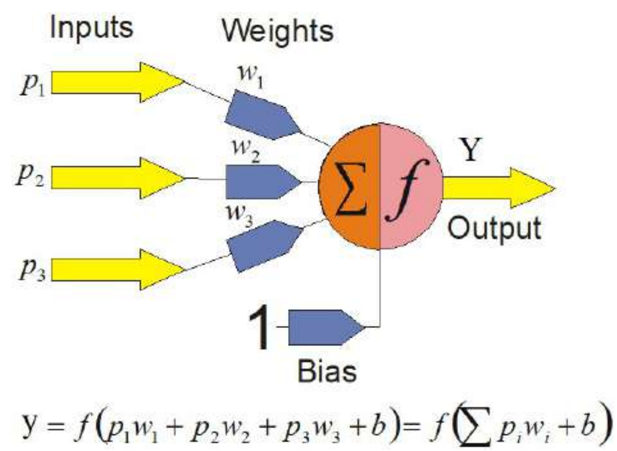


Artificial Neurons

- Artificial neurons are based on biological neurons. Each neuron in the network receives one or more inputs.
- An activation function is applied to the inputs, which determines the output of the neuron – the activation level
- A typical activation function works as follows:

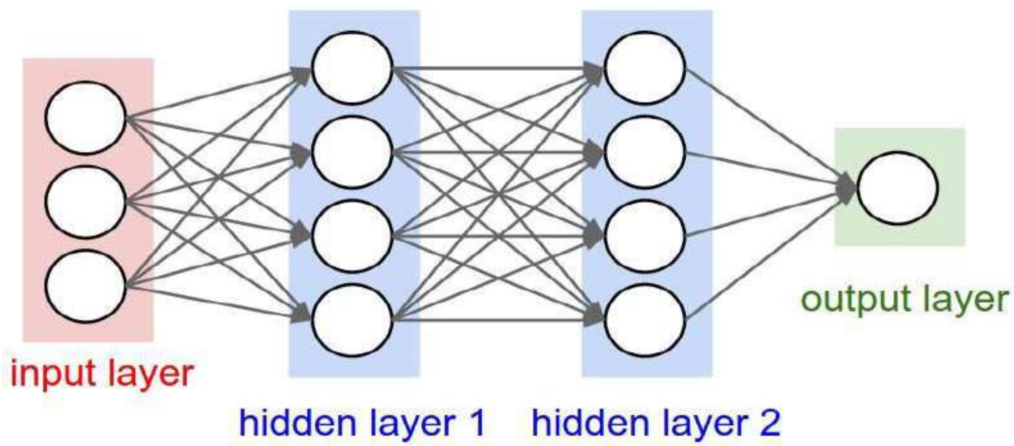
$$X = \sum_{i=1}^{n} w_i p_i \qquad Y = f(X+b) \qquad Y = f(X) = \begin{cases} 1 & for\ X > t \\ 0 & for\ X \le t \end{cases}$$

- Each node i has a weight, w_i associated with it. The input to node i is x_i and t is the threshold.
- So if the weighted sum of the inputs to the neuron is above the threshold, then the neuron fires.



Multilayer Neural Networks

- Multilayer neural networks can classify a range of functions, including non linearly separable ones. These (multilayer perceptron) networks are organized in different layers of neurons, such as input layer, hidden layer and output layer.
- The networks contain one input layer, one or more hidden layer(s) and one output layer. Each input layer neuron connects to all neurons in the hidden layer, and the neurons in the hidden layer connect to all neurons in the output layer.
- These networks also known as multilayer feed-forward neural networks.



Components of ANNs

- **Neurons and layers**
- **Connections and weights:** The network consists of connections, each connection providing the output of one neuron as an input to another neuron. Each connection is assigned a weight that represents its relative importance. A given neuron can have multiple input and output connections.
- **Activation function:** The *activation function* computes the input to a neuron from the outputs of its predecessor neurons and their connections as a weighted sum. A *bias* term can be added to the result of the propagation (activation). Example- sigmoid function, tanh function, etc.
- **Learning:** Learning is the adaptation of the network to better handle a task by considering sample observations. Learning involves adjusting the weights (and optional thresholds) of the network to improve the accuracy of the result. This is done by minimizing the observed errors between the actual output and the target.

Learning…

- The three major learning paradigms are-
  - Supervised learning
  - Unsupervised learning, and
  - Reinforcement learning.
- **Supervised Learning**
- Supervised learning uses a set of paired inputs and desired outputs. The learning task is to produce the desired output for each input. In this case the cost function is related to eliminating incorrect deductions. A commonly used cost is the mean-squared error (MSE), which tries to minimize the average squared error between the network's output and the desired output. Errors are then propagated back through the system, causing the system to adjust the weights which control the network. Tasks suited for supervised learning are pattern recognition or classification, regression or approximation, etc.

Learning…

- **Unsupervised Learning**
- In unsupervised training, the network is provided with inputs but not with desired outputs. The system itself must then decide what features it will use to group the input data. This is often referred to as self-organization or adaption.
- Tasks that fall within the paradigm of unsupervised learning are in general estimation problems; the applications include clustering, the estimation of statistical distributions, compression and filtering.

Learning

- **Reinforcement Learning**
- In applications such as playing video games, an actor takes a string of actions, receiving a generally unpredictable response from the environment after each one. The goal is to win the game, i.e., generate the most positive (lowest cost) responses.
- In reinforcement learning, the aim is to weight the network to perform actions that minimize long-term cost. At each point in time the agent performs an action and the environment generates an observation and an instantaneous cost, according to some (usually unknown) rules. The rules and the long-term cost usually only can be estimated. At any juncture, the agent decides whether to explore new actions to uncover their costs or to exploit prior learning to proceed more quickly.
- These networks applied to problems, such as vehicle routing, video game, natural resource management, medicine, etc.

Backpropagation Algorithm…

- Backpropagation (backprop, BP) is a supervised learning algorithm, for training feedforward neural networks.
- While designing a Neural Network, in the beginning, we initialize weights with some random values or any variable for that fact.
- The Backpropagation algorithm looks for the minimum value of the error function in weight space using a technique called the delta rule or gradient descent.
- The weights that minimize the error function is then considered to be a solution to the learning problem.

Backpropagation Algorithm

- **Backpropagation Algorithm:**

1) Create a feed-forward network with n_i inputs, n_h hidden units, and n_o output units.
2) Initialize all network weights to small random numbers
3) Until the termination condition is met, Do

For each <*x*,*t*> in training examples, Do

(a) *Propagate the input forward through the network:*

(i) Input the instance x to the network and compute the output o_u of every unit u in the network

(b) *Propagate the errors backward through the network:*

(i) For each network output unit k, calculate its error term δ_k :

$$\delta_k \leftarrow o_k(1-o_k)(t_k-o_k)$$

(ii) For each hidden unit h, calculate its error term δ_h :

$$\delta_h \leftarrow o_h(1-o_h) \sum_{k \in outputs} w_{kh}\, \delta_k$$

(iii) Update each network weight w_{ji} :

$$w_{ji} \leftarrow w_{ji} + \Delta w_{ji}$$

where

$$\Delta w_{ji} = \alpha \delta_j x_{ji}$$

Uncertainty

THE END