Artificial Neural Networks

Priti Srinivas Sajja
Professor
Department of Computer Science
Sardar Patel University

Visit **pritisajja.info** *for details*

Artificial Neural Networks

- Introduction
- Architectures
- Hopfield
- Perceptron
- MLP
- MLP Examples
- Hybrid Systems
- Applications
- Acknowledgement

- Name: **Dr. Priti Srinivas Sajja**
- **Communication:**
  - **Email :** priti@pritisajja.info
  - **Mobile :** +91 9824926020
  - URL :**http://pritisajja.info**
- ***Academic qualifications*** **: Ph. D in Computer Science**
- ***Thesis title***: **Knowledge-Based Systems for Socio-**
- **Economic Rural Development (2000)**
- ***Subject area of specialization*** **: Artificial Intelligence**
- ***Publications*** **: 216 in Books, Book Chapters, Journals and in Proceedings of International and National Conferences**

Artificial Neural Networks

Bio-inspired
Architectures
Hopfield
Perceptron
MLP
MLP Examples
Hybrid Systems
Applications
Acknowledgement

Bio-Inspired Computing

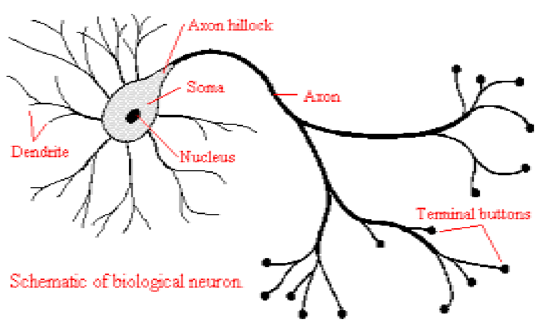
- New approaches to AI
- Taking inspiration form nature and biological systems
- Includes models such as
  - Artificial Neural Network (ANN)
  - Genetic Algorithm(GA)
  - Swarm Intelligence(SI), etc.
- Nature has virtues of self learning, evolution, emergence and immunity
- The objective of bio-inspired models and techniques to take inspiration from Mother Nature and solve problems in more effective and intelligent way

Artificial Neural Networks

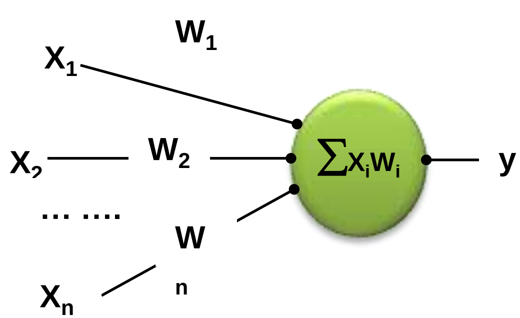
- **Neural Network**
- Architectures
- Hopfield
- Perceptron
- MLP
- MLP Examples
- Hybrid Systems
- Applications
- Acknowledgement

Artificial Neural Network (ANN)

- An artificial neural network (ANN) is **connectionist model** of programming using computers.
- An ANN attempts to give computers humanlike abilities by **mimicking the human brain's** functionality.
- The human brain consists of a network of more than a hundred billions **interconnected neurons** working in a **parallel fashion**.
- Such network are **fault tolerant**.



A biological neuron

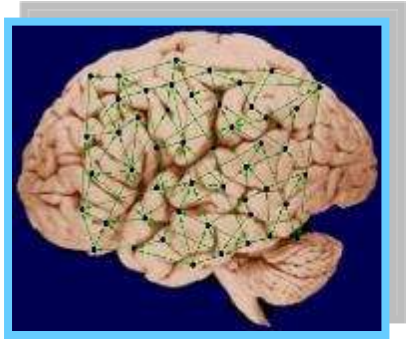


An artificial neuron

Artificial Neural Networks

- **Neural Network**
- Architectures
- Hopfield
- Perceptron
- MLP
- MLP Examples
- Hybrid Systems
- Applications
- Acknowledgement

Objective: Not to mimic brain functionality but to receive inspiration from the fact about **how brain is working.**



Characterized by:

- A **large number** of very simple neuron like processing elements.
- A large number of **weighted connection** between the elements. This weights encode the knowledge of a network.
- Highly **parallel and distributed** control.
- Emphasis on **learning** internal representation automatically.

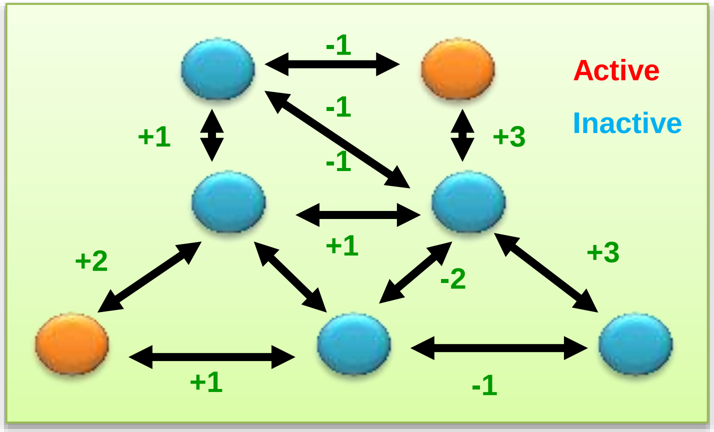
Architectures of ANN

- Hopfield network
- Perceptron
- Multi-layer Perceptron
- Self Organizing Network
- etc.

Artificial Neural Networks

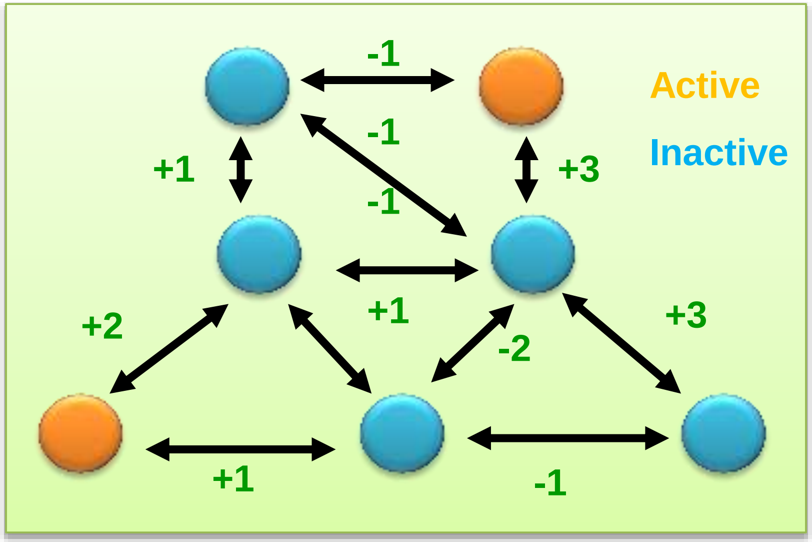
- Introduction
- Architectures
- **Hopfield Model**
- Perceptron
- MLP
- MLP Examples
- Hybrid Systems
- Applications
- Acknowledgement

- **In a Hopfield network, all processing units/elements are in two states either active or inactive.**
- **Units are connected to each other with weighted Connections.**



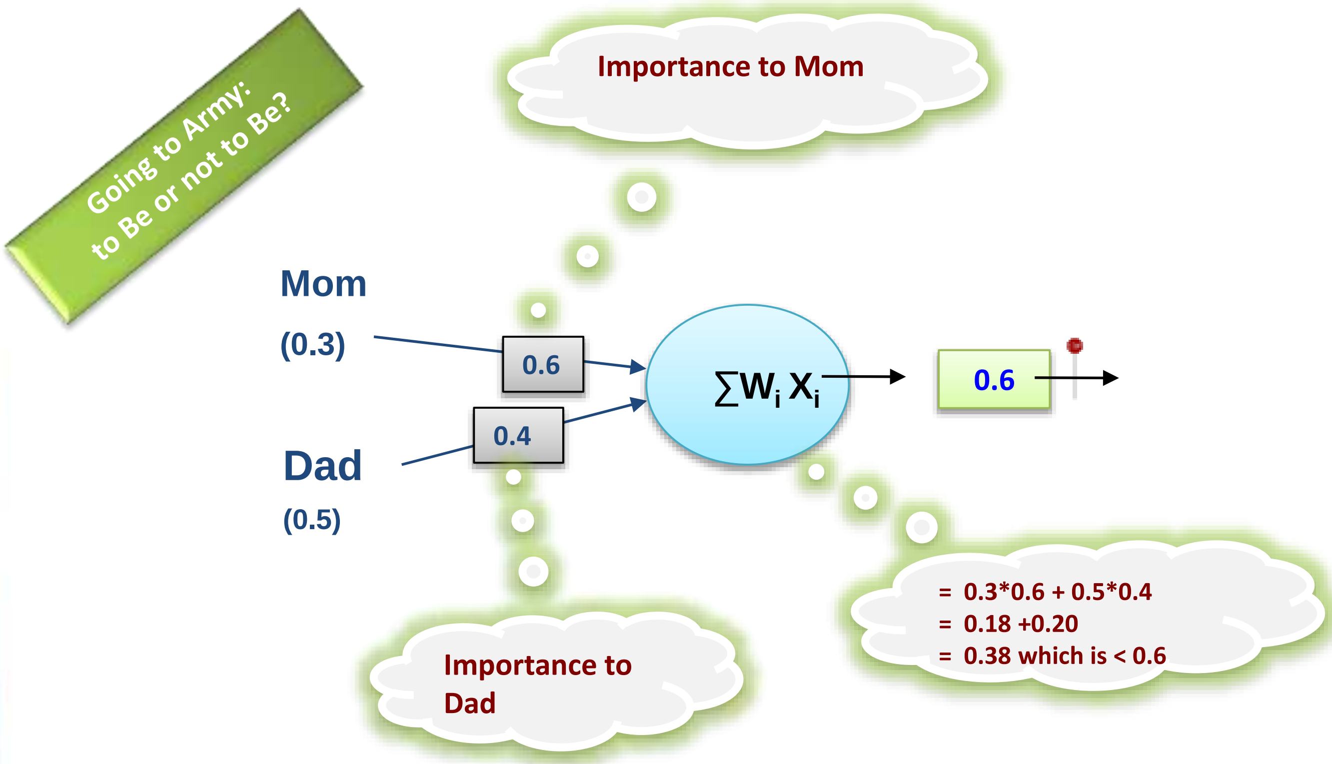
- **A positively weighted connection indicates that the units tend to active each other.**
- **A negative connection allows an active unit to deactivate a neighboring unit.**

Parallel Relaxation

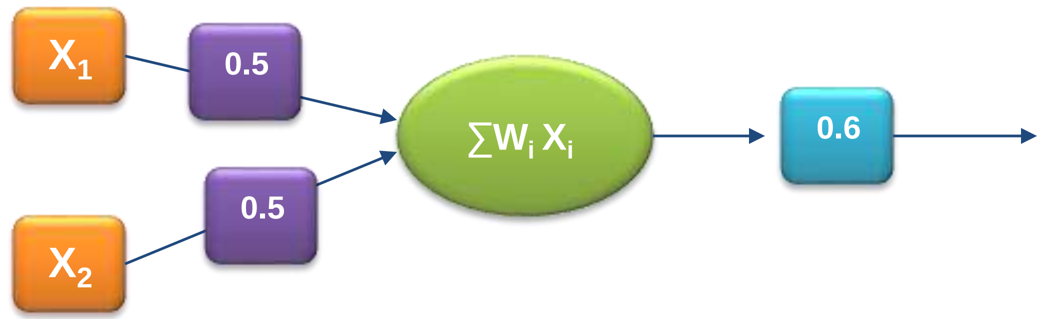


- A random unit is chosen.
- If any of its neighbors are active, the unit computes the sum of weights on the connections to those active neighbors.
- If the sum is positive, the unit becomes active else new random unit is chosen.
- This process will continue till the network become stable. That is no unit can change its status. This process is known as parallel relaxation.

Artificial Neural Networks

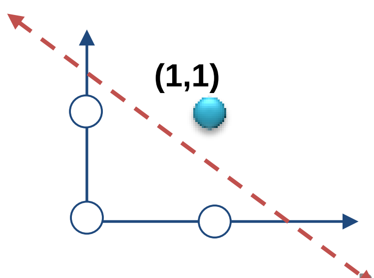


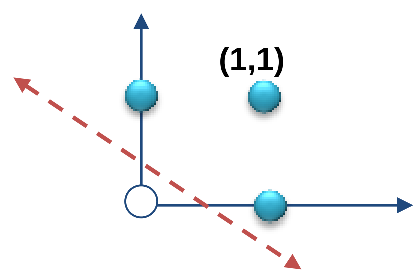
Logical Gate AND and OR



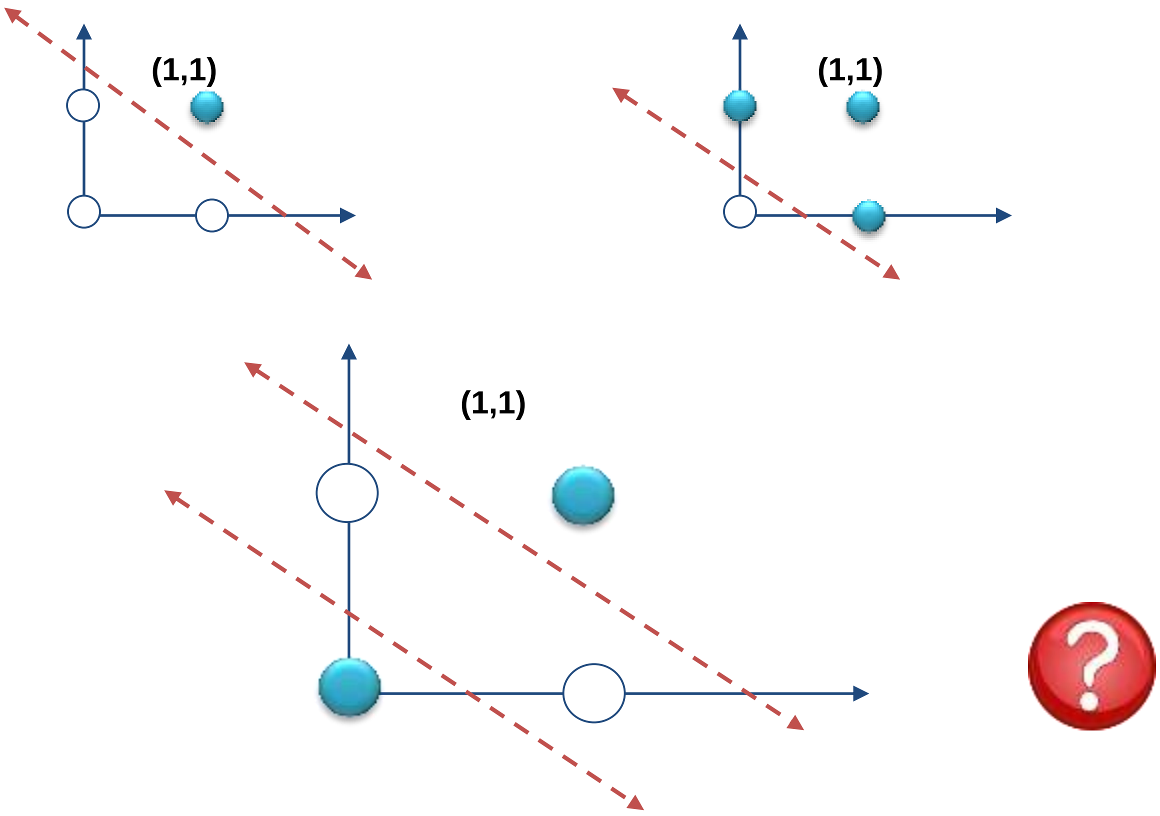
Logical AND Truth Table

X_1	X_2	X_1 AND X_2
0	0	0*0.5 + 0*0.5 = 0 <0.6 → 0
0	1	0*0.5 + 1*0.5 = 0 <0.6 → 0
1	0	1*0.5 + 0*0.5 = 0.5 <0.6 → 0
1	1	1*0.5 + 1*0.5 = 1 >0.6 → 1

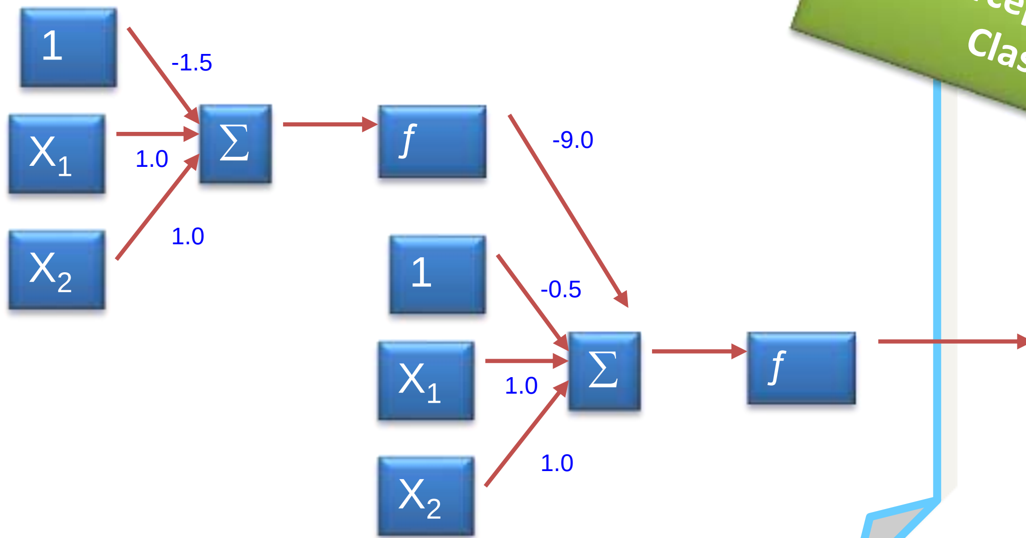


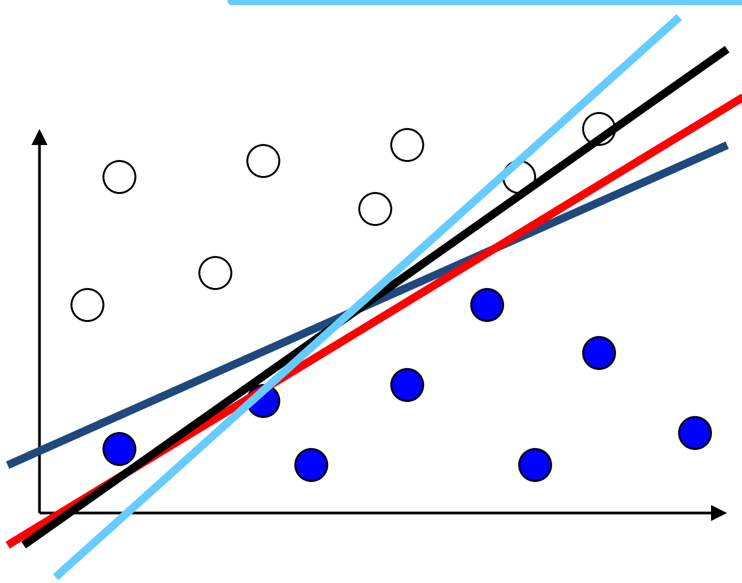


Logical Gate AND and OR



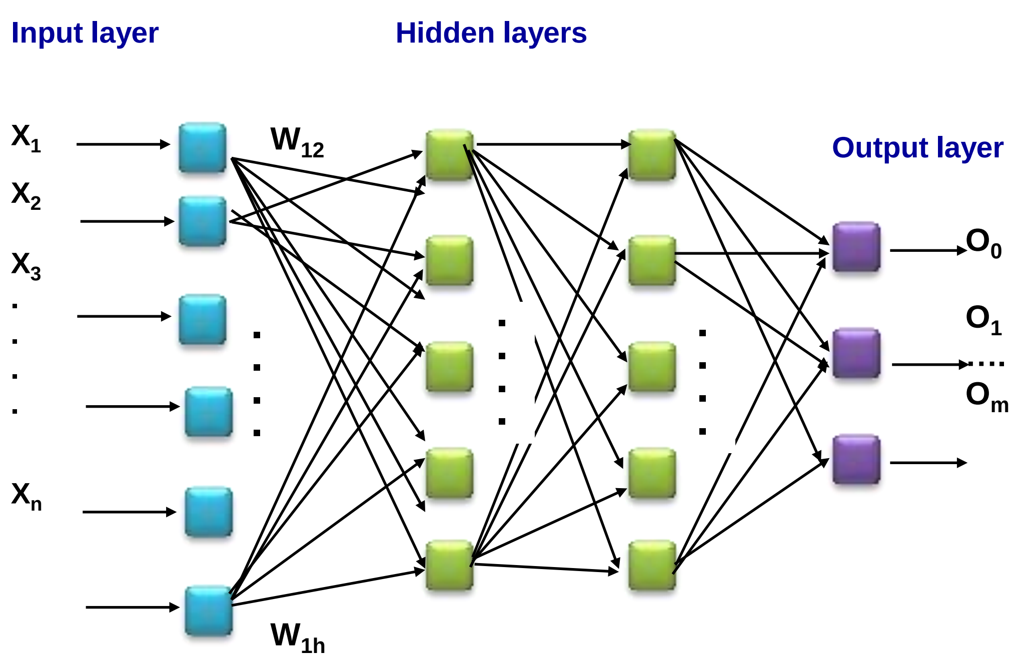
A Perception Learning to Solve a Classification Problem





K	w_0	w_1	w_2
10	.41	-.17	.14
100	.22	-.14	.11
300	-.1	-.008	.07
635	-.49	-.1	.14

Artificial Neural Networks



A multilayer perceptron

Artificial Neural Networks

ANN Design Heuristic

- Verify the nature of the problem. Typically where many data are available but there is a lack of generalized logic, one may go for multilayer perceptron ANN.
- Select critical parameters that play an important role in decision making. For this, one needs to study the data available on hand. Alternatively, a few successful cases where such decisions are made can be considered. Total number of such important and critical parameters is, say ‘n’.
- Create an input layer (I) containing ‘n’ number of neurons. Also, assign its activation function as the value of the input.
- Identify possible choices/output options for the problem. Say this number is ‘m’ .

Artificial Neural Networks

- Introduction
- Architectures
- Hopfield
- Perceptron
- **MLP**
- MLP Examples
- Hybrid Systems
- Applications
- Acknowledgement

ANN Design Heuristic

- Create one or two hidden layers (H1 and H2) containing an average of input and output number of nodes; that is (‘n’ + ‘m’)/2.
- Assign activation function in each neuron of every layer. Typical activation functions are weighted sum, sigmoid, hyperbolic tangent, rectified linear activation unit, etc.
- Activation function is also known as **transfer function** or **processing function**.
- The activation function at the first hidden layer involves input values from the input layer nodes with their weights.
- The activation function at the second hidden layer involves the previous layer (hidden) nodes’ values with their weights.

Artificial Neural Networks

- Introduction
- Architectures
- Hopfield
- Perceptron
- **MLP**
- MLP Examples
- Hybrid Systems
- Applications
- Acknowledgement

ANN Design Heuristic

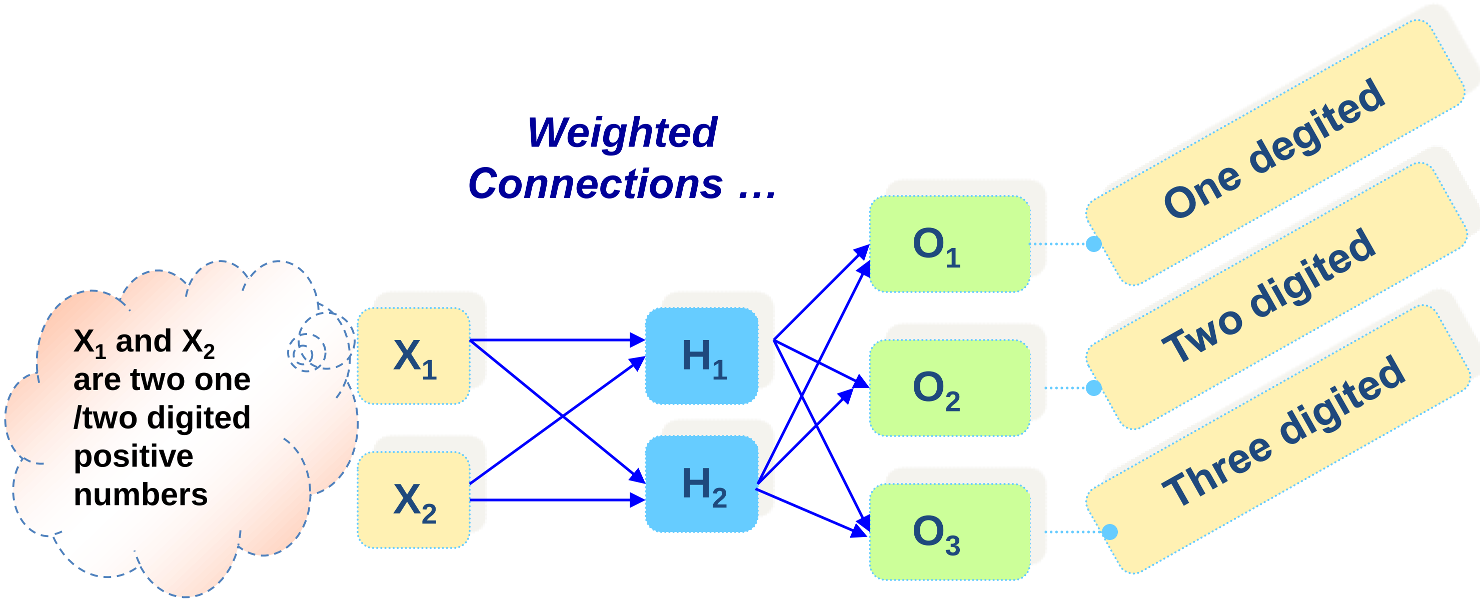
- Create an output layer (O) containing ‘m’ number of nodes. Assign an output activation function to each neuron in the output layer. The activation function at the output layer involves values from the last hidden layer nodes with their weights.
- Connect all neurons in such a way that ‘each neuron is connected in a forward direction to every neuron of the adjacent layer’. This makes the network **fully connected**, feed-forward (as all the connections are in a forward direction only) multilayer neural network .
- Assign random weights to each connection .
- Train the network with collected valid data sets.
- Quality of learning depends on the training data sets.
- Its called **supervised learning**.

Back Propagation Learning in ANN

Table 2.6 Major steps in backpropagation

Phase 0: Initialization of the network by assigning random weights
For each training data set do following
Phase 1: Forward Pass
1. Take the input data from the current data set and feed it to the input layer 2. Calculate the hidden layer values and output layer values as per the activation functions assigned 3. Let neural network output 'what it thinks'
Phase 2: Backward Pass
1. Compute error by finding the difference between the calculated output and correct output from the data set. The error can be calculated for each neuron and stored as 'Delta' typically 2. Back propagate the error and with delta values, update the network weights. Use the formula as: weight = weight + learning_rate * error * input. Repeat the forward and backward pass until the error is acceptable

Examples of Multilayer Perceptron



Training Set Data

2, 3, 1, 0, 0

10, 10, 0, 1, 0

90, 90, 0, 0, 1

…… ….. …

Selection of Mobile

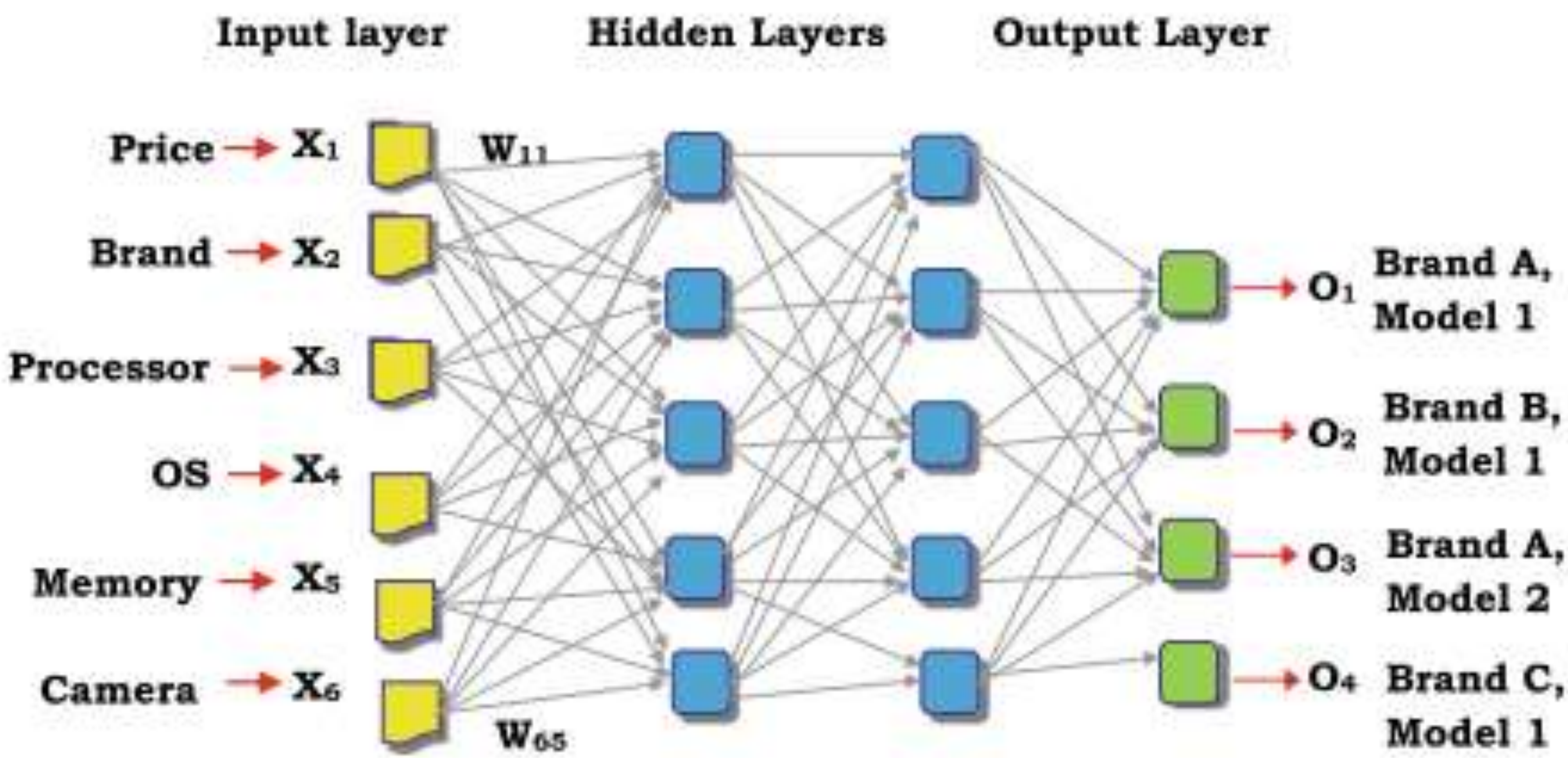


Fig. 4.5 Neural network to select mobile

Source: Sajja, P.S. "Illustrated computational intelligence: Examples and applications", Springer International Publishing, Singapore (Nov'20)

Selection of Mobile

Table 4.2 Training data sets for mobile selection

Inputs						Outputs			
X_1	X_2	X_3	X_4	X_5	X_6	A1	B1	A2	C1
1.0	1.0	0.8	0.7	0.6	0.7	1	0	0	0
1.0	0.8	0.9	0.7	0.8	0.8	0	1	0	0
0.8	0.6	0.8	0.7	0.8	0.8	0	1	0	0
0.6	0.3	0.8	0.8	0.7	0.6	0	1	0	0
0.3	0.4	0.5	0.4	0.6	0.5	0	0	0	1
0.5	0.3	0.4	0.5	0.4	0.4	0	0	0	1
0.4	0.6	0.5	0.4	0.4	0.4	0	0	1	0
...	...	...	...	...	...	...	...	...	...

Input variables: X_1: *Price*, X_2: *Brand*, X_3: *Processor*, X_4: *Operating System*, X_5: *Memory*, X_6: *Camera*.
Output variables: ***A1***: *Brand A Model 1*, ***B1***: *Brand B Model 1*, ***A2***: *Brand A Model 2*, ***C1***: *Brand C Model 1*

Source: Sajja, P.S. "Illustrated computational intelligence: Examples and applications", Springer International Publishing, Singapore (Nov'20)

Selection of Course

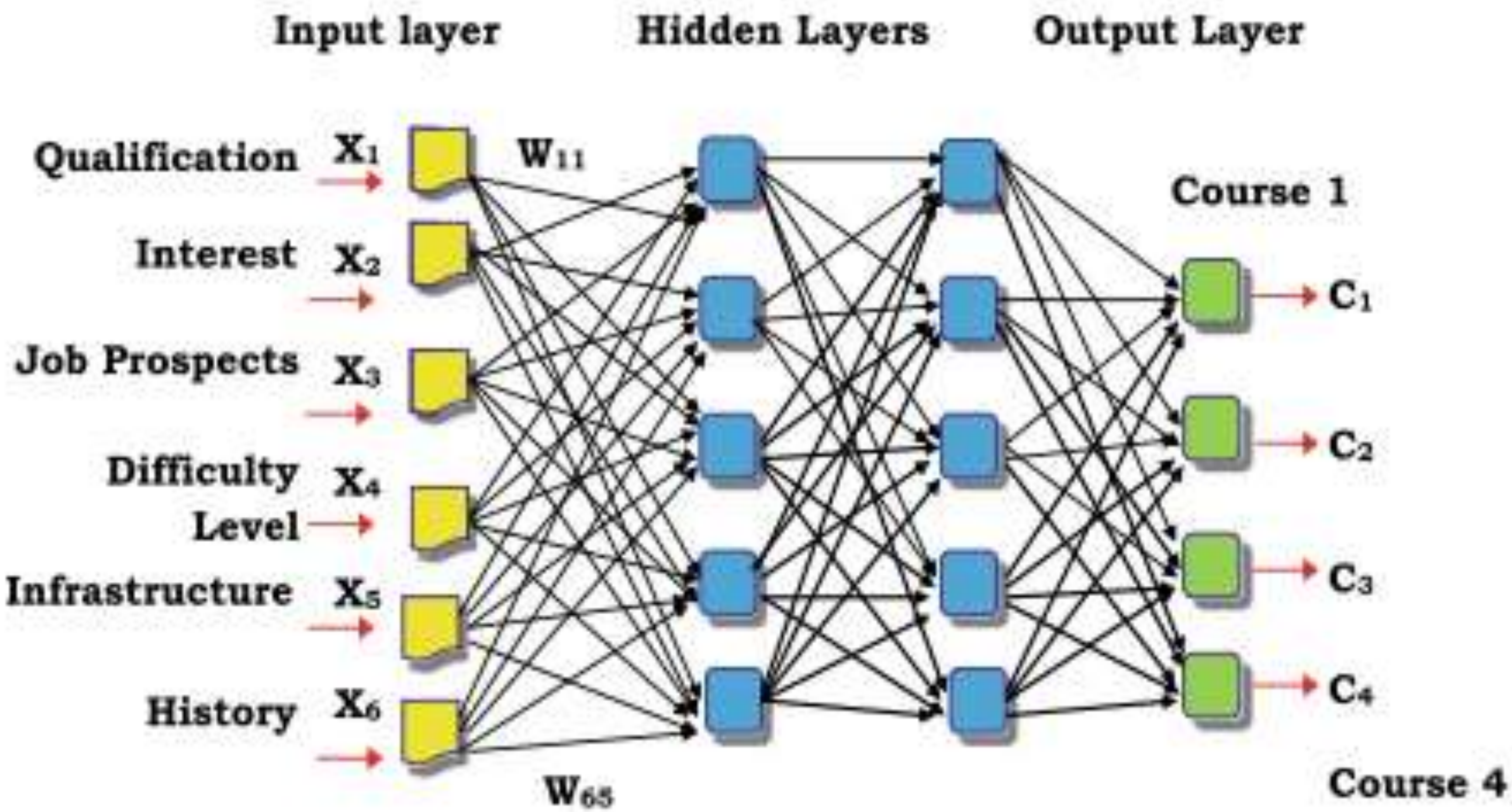


Fig. 4.7 Neural network for course selection

Source: Sajja, P.S. "Illustrated computational intelligence: Examples and applications", Springer International Publishing, Singapore (Nov'20)

Selection of Course

Table 4.4 Training data sets for course selection

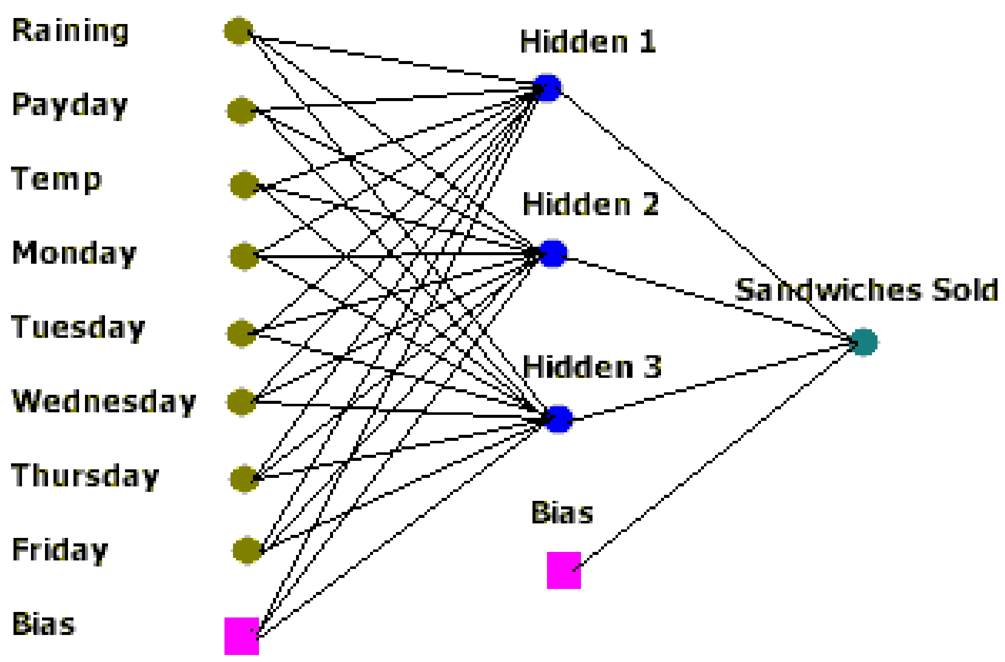
Inputs						Outputs			
X_1	X_2	X_3	X_4	X_5	X_6	C_1	C_2	C_3	C_4
1.0	1.0	1.0	0.8	0.6	0.7	1	0	0	0
0.8	1.0	1.0	0.8	0.6	0.7	1	0	0	0
0.5	2.0	0.8	0.6	0.7	0.8	0	1	0	0
0.8	3.0	0.6	0.7	0.6	0.6	0	0	1	0
0.7	4.0	0.9	0.6	0.8	0.7	0	0	0	1
0.7	1.0	1.0	0.8	0.6	0.7	1	0	0	0
0.4	4.0	0.7	0.6	0.8	0.7	0	0	0	1
1.0	2.0	0.8	0.6	0.7	0.8	0	1	0	0
...	...	...	...	...	...	...	..		..

Input variables: X_1: *Qualification*, X_2: *Interest*, X_3: *Job Prospects*, X_4: *Difficulty*, X_5: *Infrastructure*, X_6: *History*
Output variables: C_1: *Course 1*, C_2: *Course 2*, C_3: *Course 3*, C_4: *Course 4*

Source: Sajja, P.S. "Illustrated computational intelligence: Examples and applications", Springer International Publishing, Singapore (Nov'20)

Artificial Neural Networks

Predicting Sandwich Sales



http://www.wardsystems.com

Artificial Neural Networks

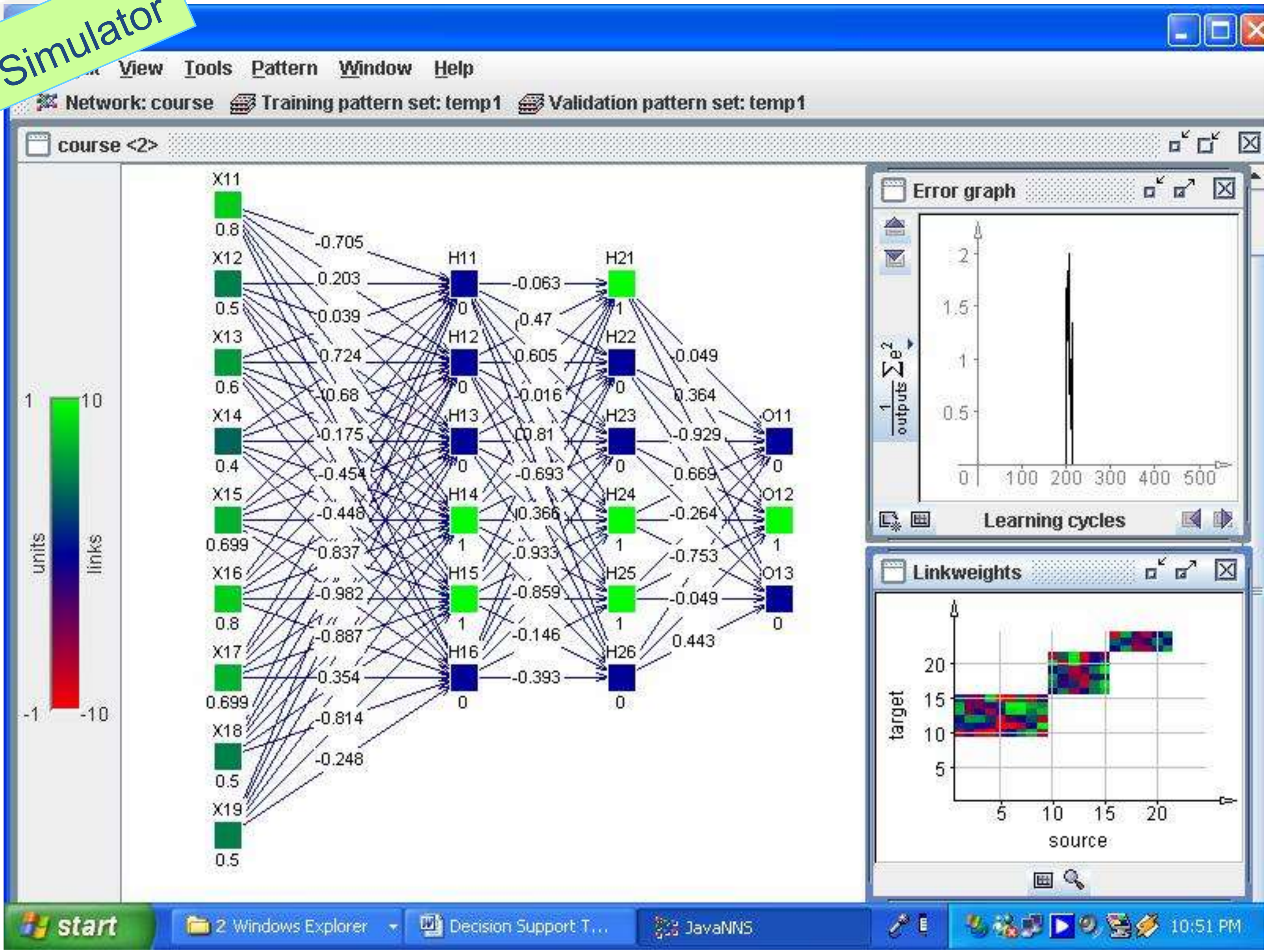
Training Data for Sandwich Case

Bob's New York Style Deli Sandwich Sales Neural Net Predictor

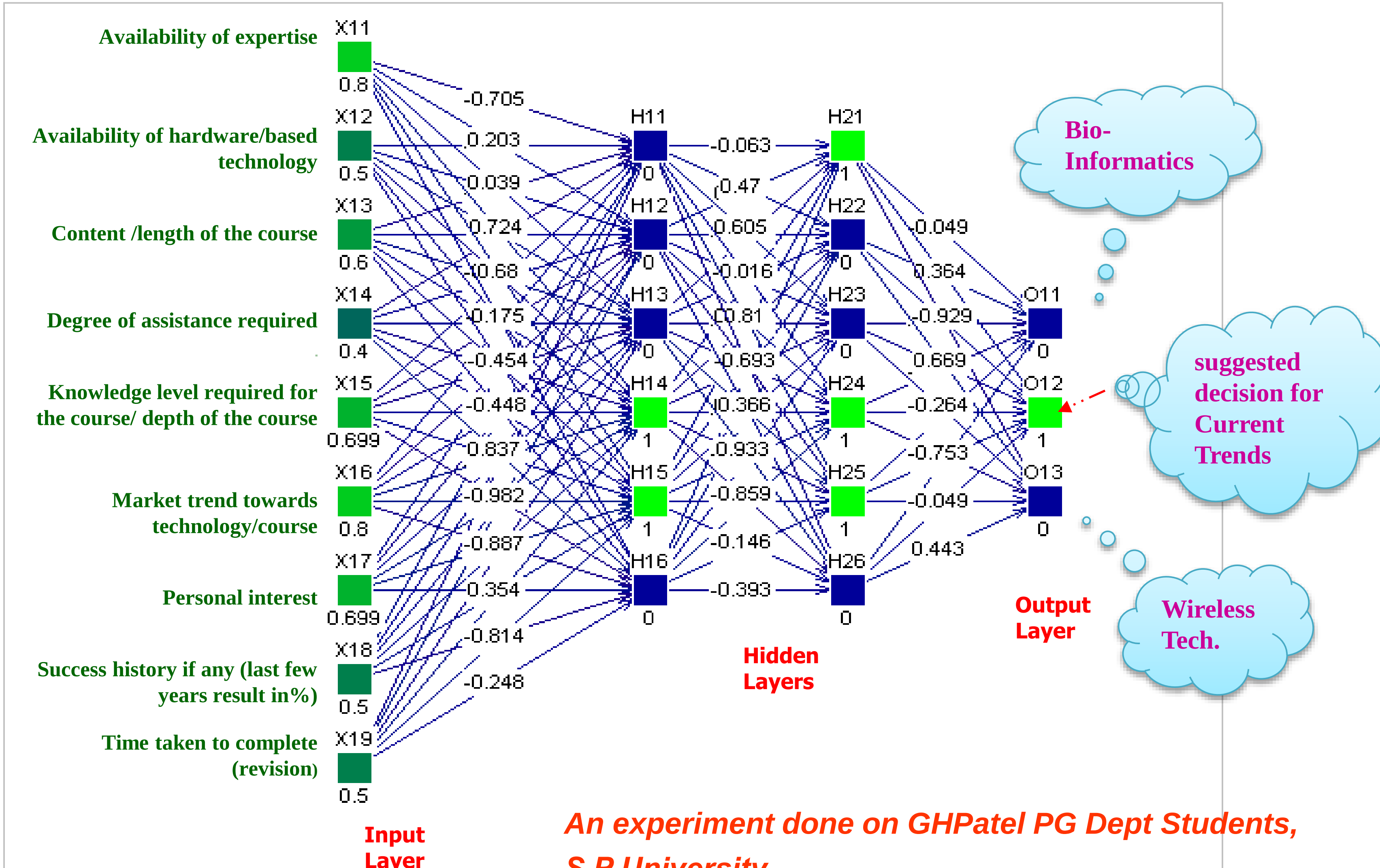
Input 1	Input 2		Input 3	In 4	In 5	In 6	In 7	In 8	Output
Raining?	Payday in area?	Temperature Fahrenheit	Scaled temperature	Mon	Tue	Wed	Thu	Fri	Sandwiches Sold
0	0	55	-0.80	1					120
0	0	60	-0.60		1				120
0	0	59	-0.64			1			132
1	0	65	-0.40				1		103
0	0	65	-0.40					1	152
0	1	66	-0.36	1					157
1	0	50	-1.00		1				79
0	0	70	-0.20			1			157
0	0	65	-0.40				1		130
1	0	60	-0.60					1	115
0	0	77	0.08	1					168

http://www.wardsystems.com

ANN Simulator

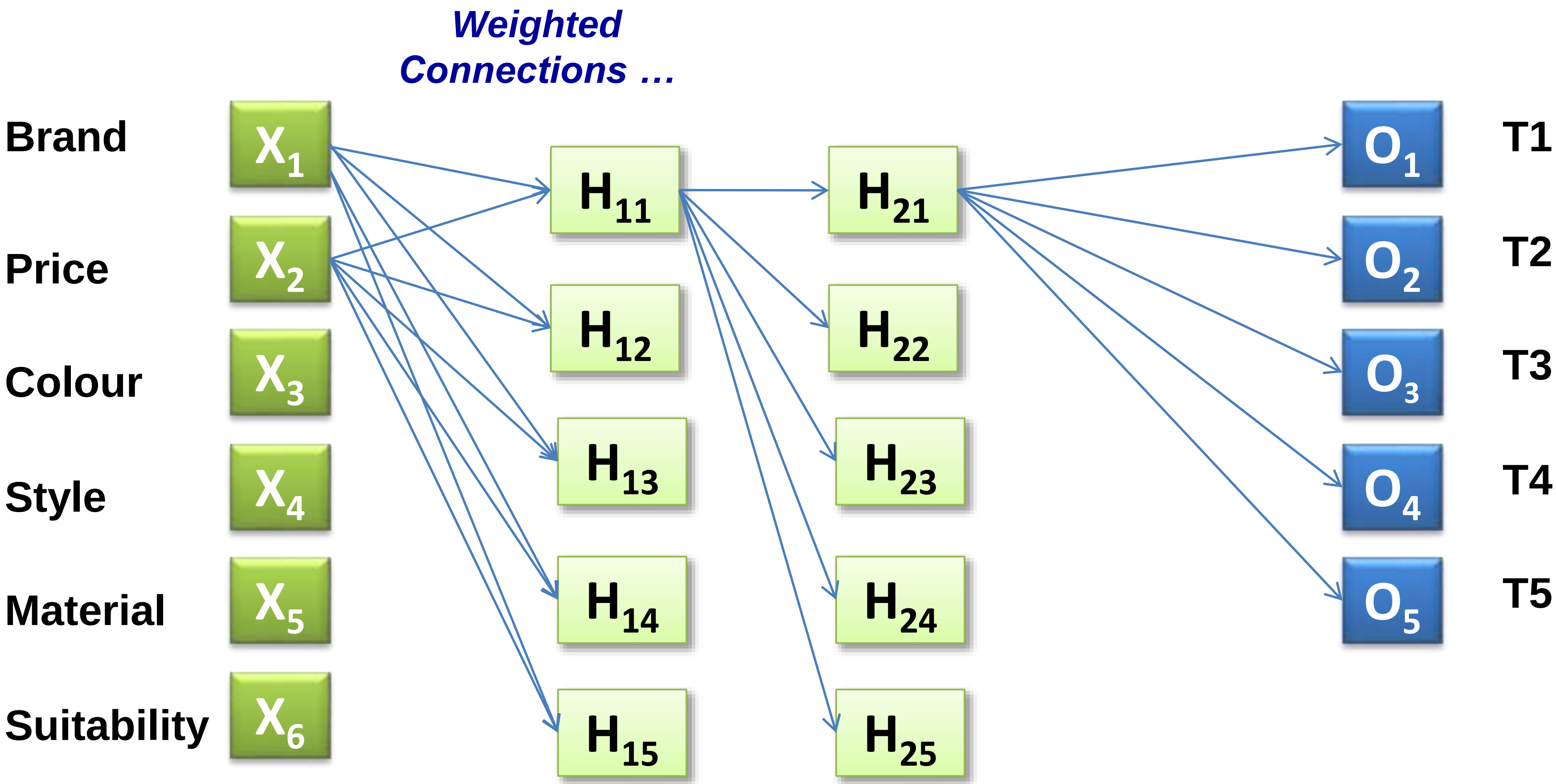


Elective Course Selection using ANN



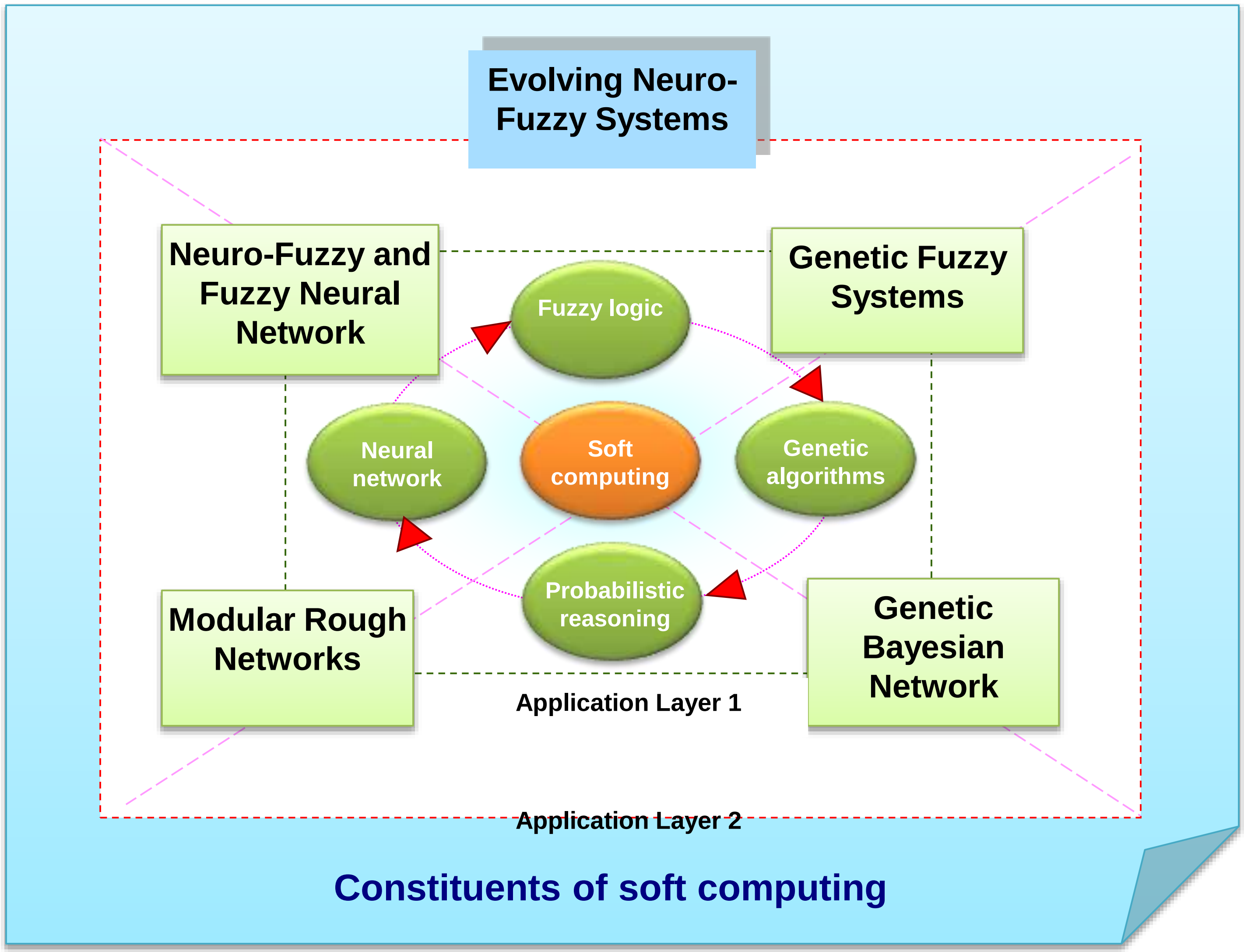
An experiment done on GHPatel PG Dept Students, S P University…

T Shirt Selection



Complete the connections ……
(each node is connected with every in adjacent layer)

Hybrid Systems/Soft Computing



Strength of a Hybrid Soft Computing System

Field	Strength Offered
Fuzzy logic	Approximate reasoning, impressions
Artificial neural network	Learning and implicit knowledge representation
Genetic algorithm	Natural evolution and optimization
Probabilistic reasoning	Uncertainty

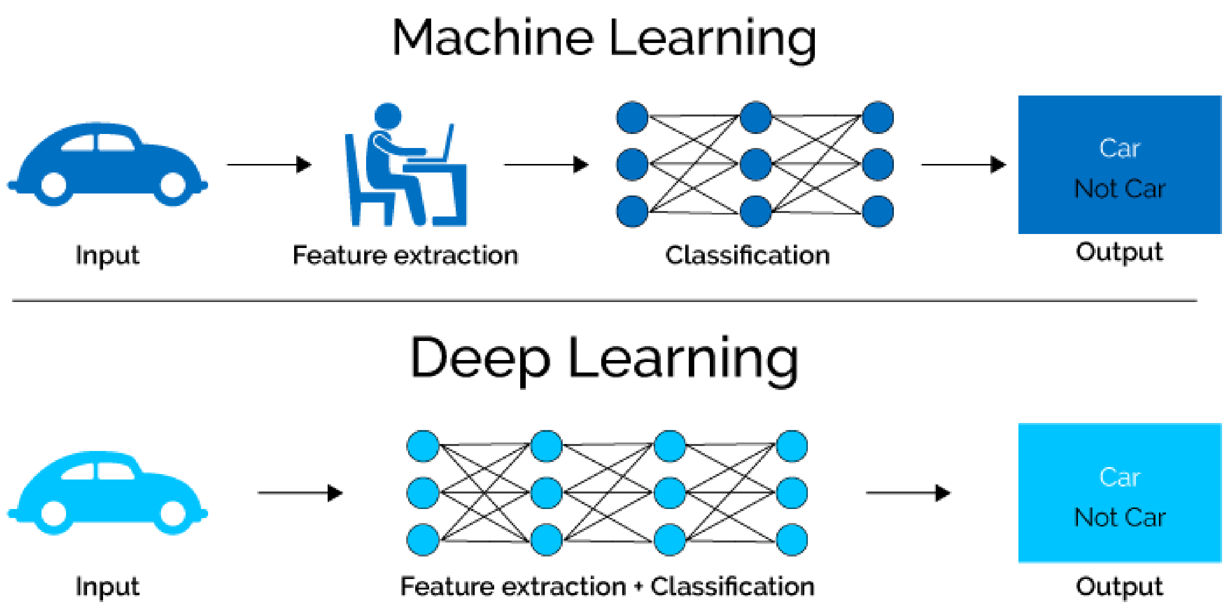
Artificial Neural Networks

- **Machine Learning (ML)** is defined as an ability to learn without being explicitly programmed.
- **Supervised ML**: “trained” on a pre-defined set of “training examples”
- **Unsupervised ML**: using bunch of data, the machine must find patterns
- **Deep Learning:** several stages of non-linear information processing in hierarchical architectures are utilized for pattern classification and for feature learning.

Artificial Neural Networks

- Introduction
- Architectures
- Hopfield
- Perceptron
- MLP
- MLP Examples
- **Hybrid Systems**
- Applications
- Acknowledgement

Deep Learning As extension of Machine Leaning



https://medium.com/swlh/ill-tell-you-why-deep-learning-is-so-popular-and-in-demand-5aca72628780

Artificial Neural Networks

Applications of Machine Learning

- Natural language processing and natural query.
- Film industry where dubbing of film, re-colouring film prints and adding sounds to the silent films.
- News aggregators
- Machine Translation (partly/ fully) with sentence corrections.
- Automatic prescription of handwriting production and understanding.
- Managing knowledge wallet.
- Chatting agents.
- Intelligent games.
- Google's automatic statistician project.
- Intelligent web applications including searching and intelligent crawling.
- Image, speech and multimedia mining.
- Utilization of social network platform for various activities.
- Development for resources and sustainable development, population information, governance (weather forecasting, infrastructural development, natural resources).
- Sensor web, agricultural information, decision support systems in domains like forestry and fisheries.

References

- llustrationsOf.com, biocomp.unibo.it
- www.gadgetcage.com,
- Prersentermedia.com
- Presentationmagazine.com
- Clikr.com, Engadget.com
- scenicreflections.com
- lih.univ-lehavre.fr, business2press.com
- globalswarminghoneybees.blogspot.com
- https://machinelearningmastery.com
- https://www.analyticsvidhya.com
- ***Knowledge-based systems***, Akerkar RA and Priti Srinivas Sajja, Jones & Bartlett Publishers, Sudbury, MA, USA (2009)
- Akerkar R.A. and Sajja, P.S. “Intelligent techniques for data science”, Springer International Publishing, Switzerland (Oct’16)
- Sajja, P.S. “Illustrated computational intelligence: Examples and applications”, Springer International Publishing, Singapore (Dec’20) https://www.springer.com/gp/book/9789811595882