

Supervised Learning: Classification with Artificial Neural Network (ANN)

Dr. Yuzana Win (Nagasaki University, Japan)

Lecturer

Department of Computer Engineering and
Information Technology

Lecture Objectives

- To introduce
 - Classification Techniques or Methods
 - What is Artificial Neural Network (ANN)?
 - How does the ANN algorithm works?
 - Pros and Cons

Classification Techniques or Methods

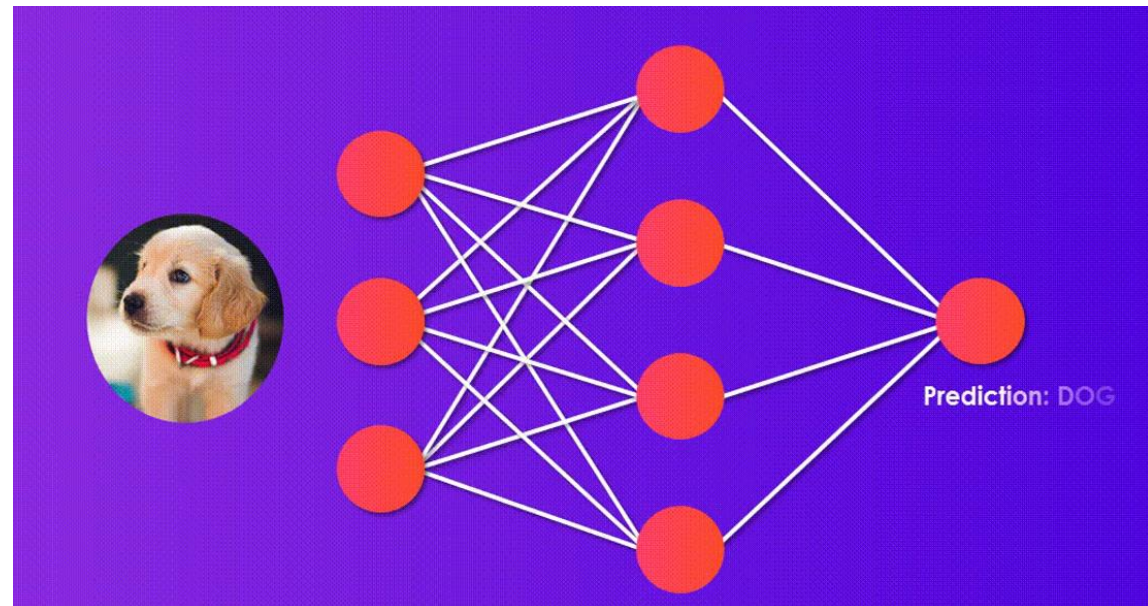
- Naïve Bayes Classifier *
- Support Vector Machines *
- Neural Networks (Deep Learning)
- Decision Tree based Methods
- Rule-based Methods
- K-Nearest Neighbors (kNN) *

Classification Techniques or Methods

- Naïve Bayes Classifier *
- Support Vector Machines *
- **Neural Networks (Deep Learning)**
- Decision Tree based Methods
- Rule-based Methods
- K-Nearest Neighbors (KNN) *

What is Artificial Neural Network?

- Basic idea:
 - Artificial Neural Network is an information processing model **like a human brain** does every second
 - Human brain learning models **based on current 'understanding'**

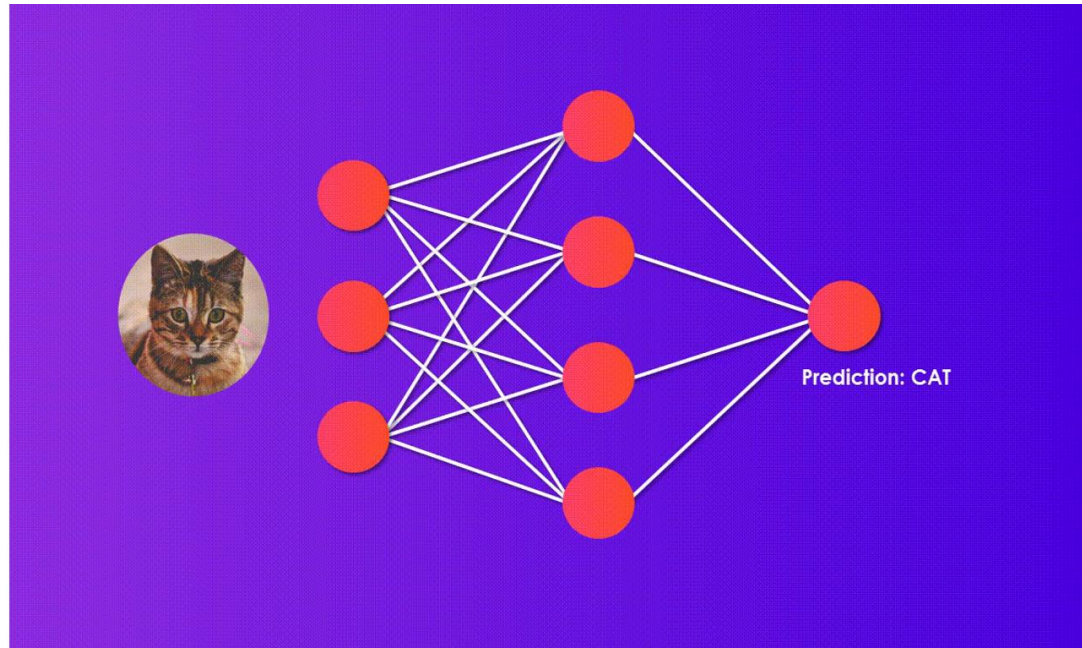


What are Artificial Neural Network used for?

- Neural Networks are used for a variety of tasks, a popular use is **classification**.
- They are also used in
 - Self driving cars
 - Character recognition
 - Image compression
 - Stock market prediction and
 - Lots of other interesting applications

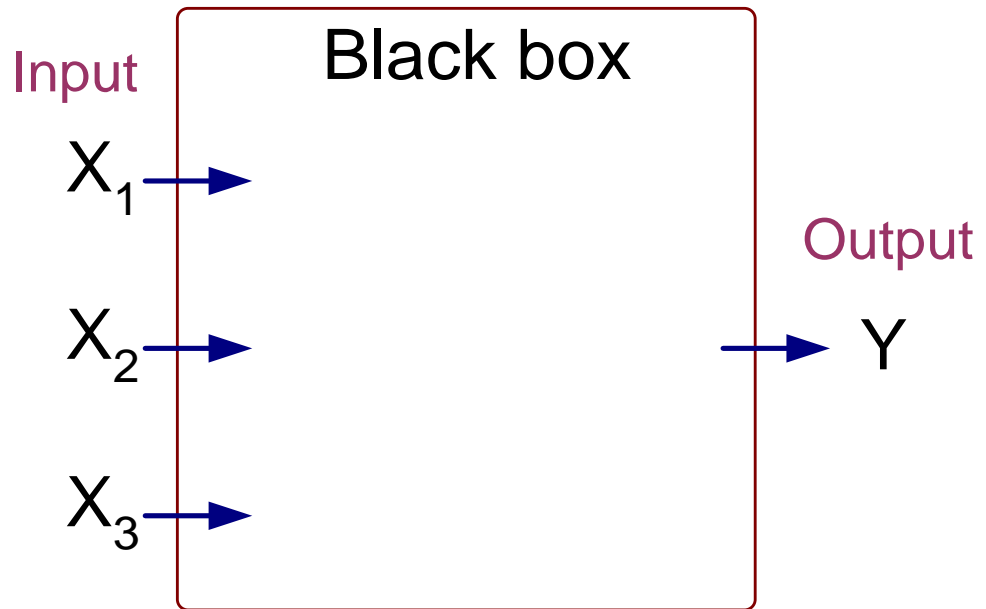
What are Artificial Neural Network used for?

- For example, collect datasets of images of different breeds of **cats** and then **train a neural network** on the images. If a **new image of a cat** it will give a statistical score on how closely the new image matches the model and then **will output what breed of cat the image is**.



How does a Artificial Neural Network (ANN) works?

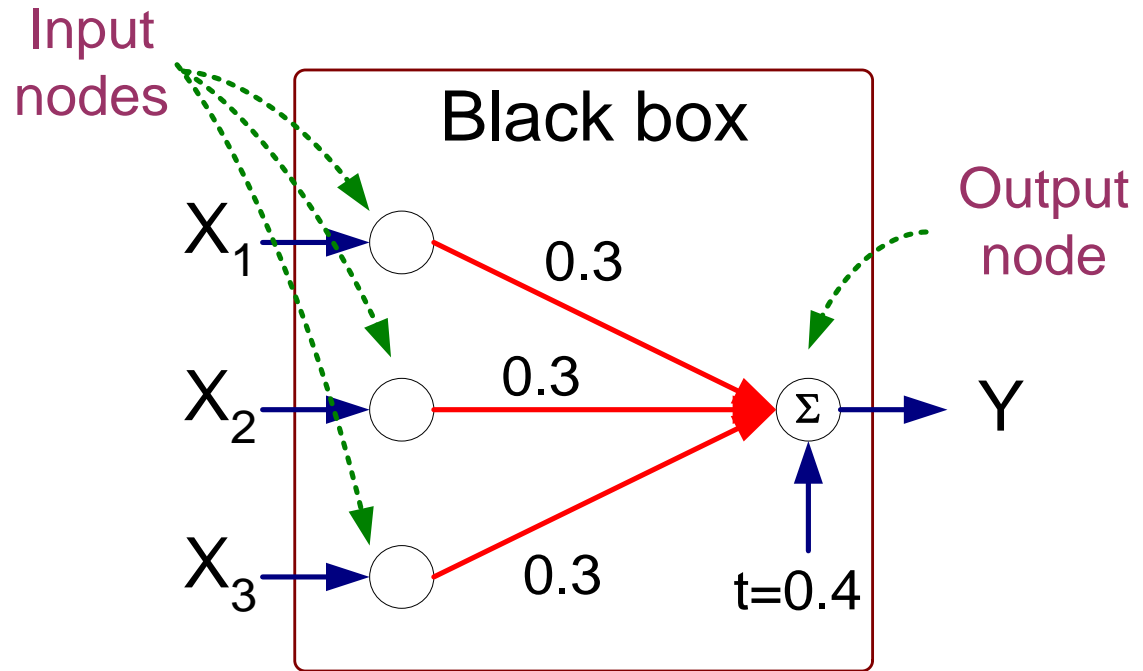
X_1	X_2	X_3	Y
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1
0	0	1	0
0	1	0	0
0	1	1	1
0	0	0	0



Output Y is 1 if at least two of the three inputs are equal to 1.

Artificial Neural Networks (ANN)

X_1	X_2	X_3	Y
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1
0	0	1	0
0	1	0	0
0	1	1	1
0	0	0	0

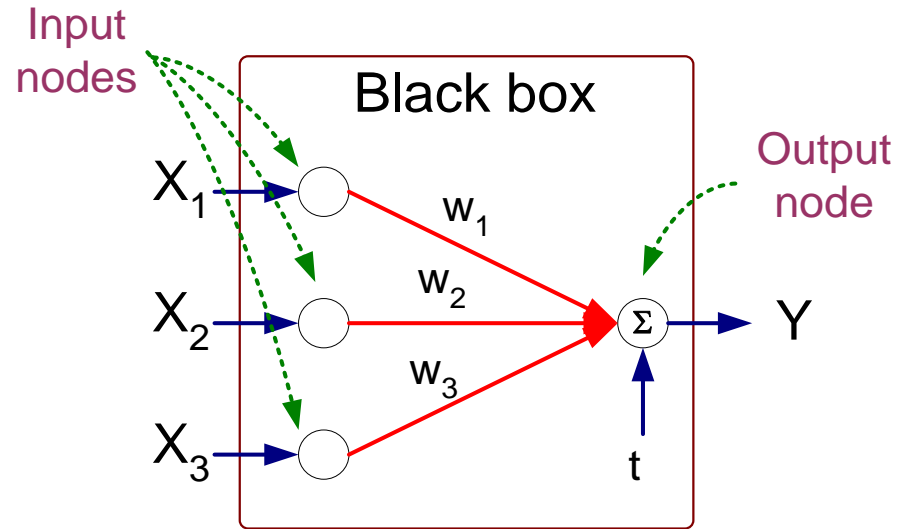


$$Y = I(0.3X_1 + 0.3X_2 + 0.3X_3 - 0.4 > 0)$$

$$\text{where } I(z) = \begin{cases} 1 & \text{if } z \text{ is true} \\ 0 & \text{otherwise} \end{cases}$$

Artificial Neural Networks (ANN)

- Model is an assembly of inter-connected nodes and weighted links
- Output node sums up each of its input value according to the weights of its links
- Compare output node against some threshold t

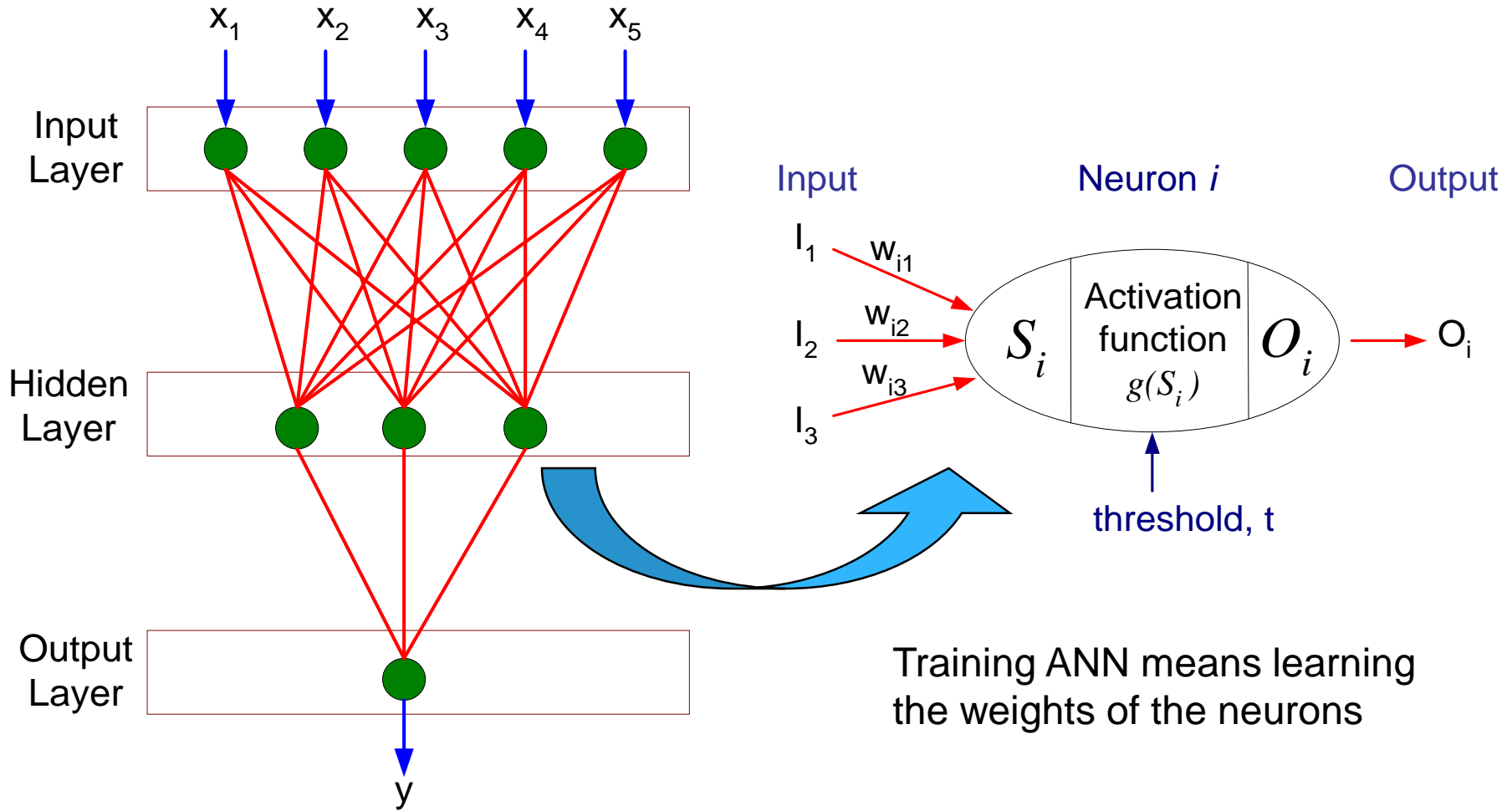


Perceptron Model

$$Y = I\left(\sum_i w_i X_i - t\right) \quad \text{or}$$

$$Y = \text{sign}\left(\sum_i w_i X_i - t\right)$$

General Structure of ANN



Algorithm for learning ANN

- Initialize the weights (w_0, w_1, \dots, w_k)
- Adjust the weights in such a way that the output of ANN is consistent with class labels of training examples

- Objective function:

$$E = \sum_i [Y_i - f(w_i, X_i)]^2$$

- Find the weights w_i 's that minimize the above objective function
 - e.g., forward propagation algorithm

ANN Forward Propagation

- Bias Nodes
 - Add one node to each layer that has constant output
- Forward propagation
 - Calculate from input layer to output layer
 - For each neuron:
 - Calculate weighted average of input
 - Calculate activation function

Neuron Model

- Firing Rules:
 - Threshold rules:
 - Calculate weighted average of input
 - Fire if larger than threshold
 - Perceptron rule
 - Calculate weighted average of input input
 - Output activation level is

$$\phi(v) = \begin{cases} 1 & v \geq \frac{1}{2} \\ v & 0 \leq v \leq \frac{1}{2} \\ 0 & v \leq 0 \end{cases}$$

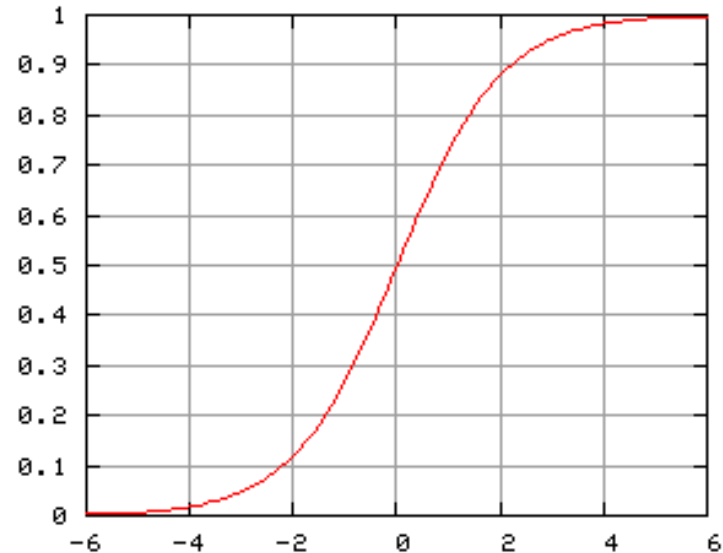
Neuron Model

- Firing Rules: Sigmoid functions:
 - Hyperbolic tangent function

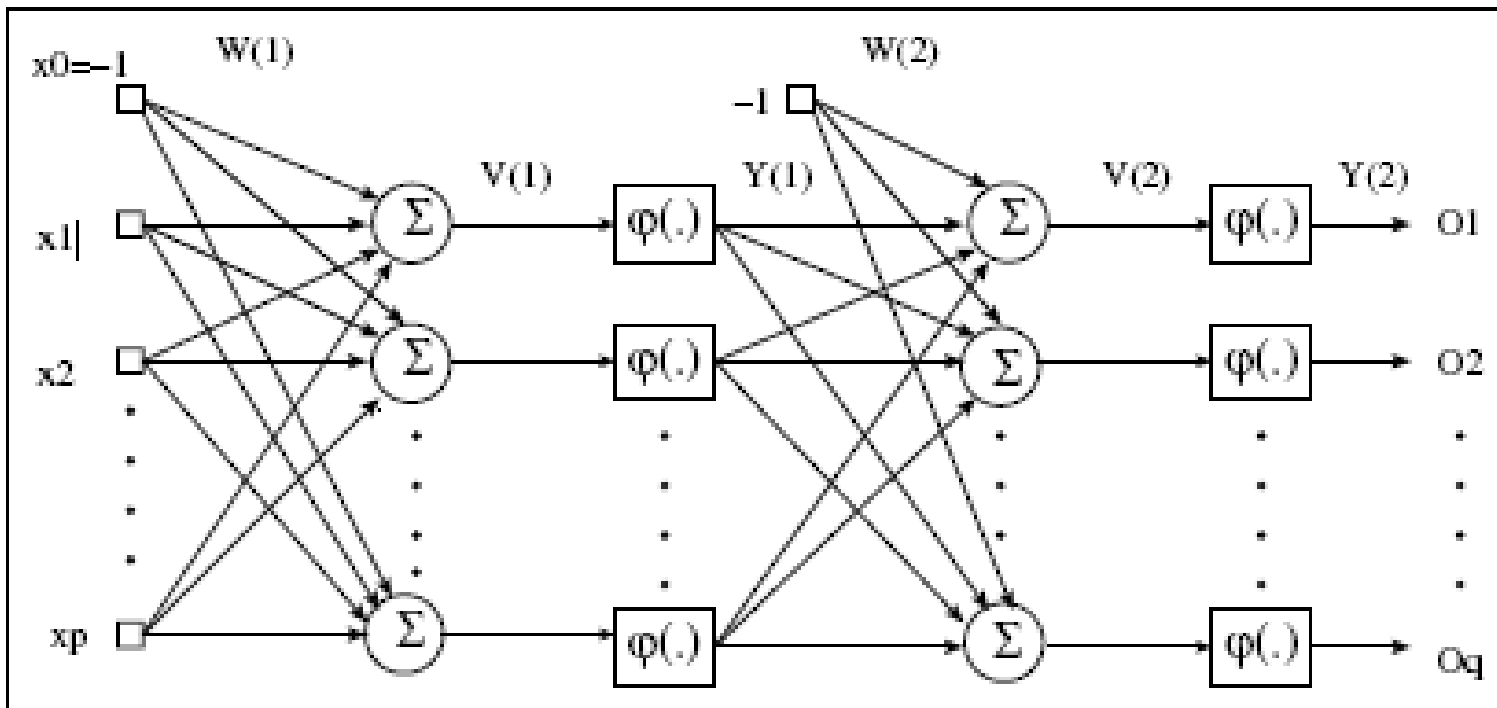
$$\varphi(v) = \tanh(v/2) = \frac{1 - \exp(-v)}{1 + \exp(-v)}$$

- Logistic activation function

$$\varphi(v) = \frac{1}{1 + \exp(-v)}$$



ANN Forward Propagation



ANN Forward Propagation

- Apply input vector \mathbf{X} to layer of neurons.
- Calculate

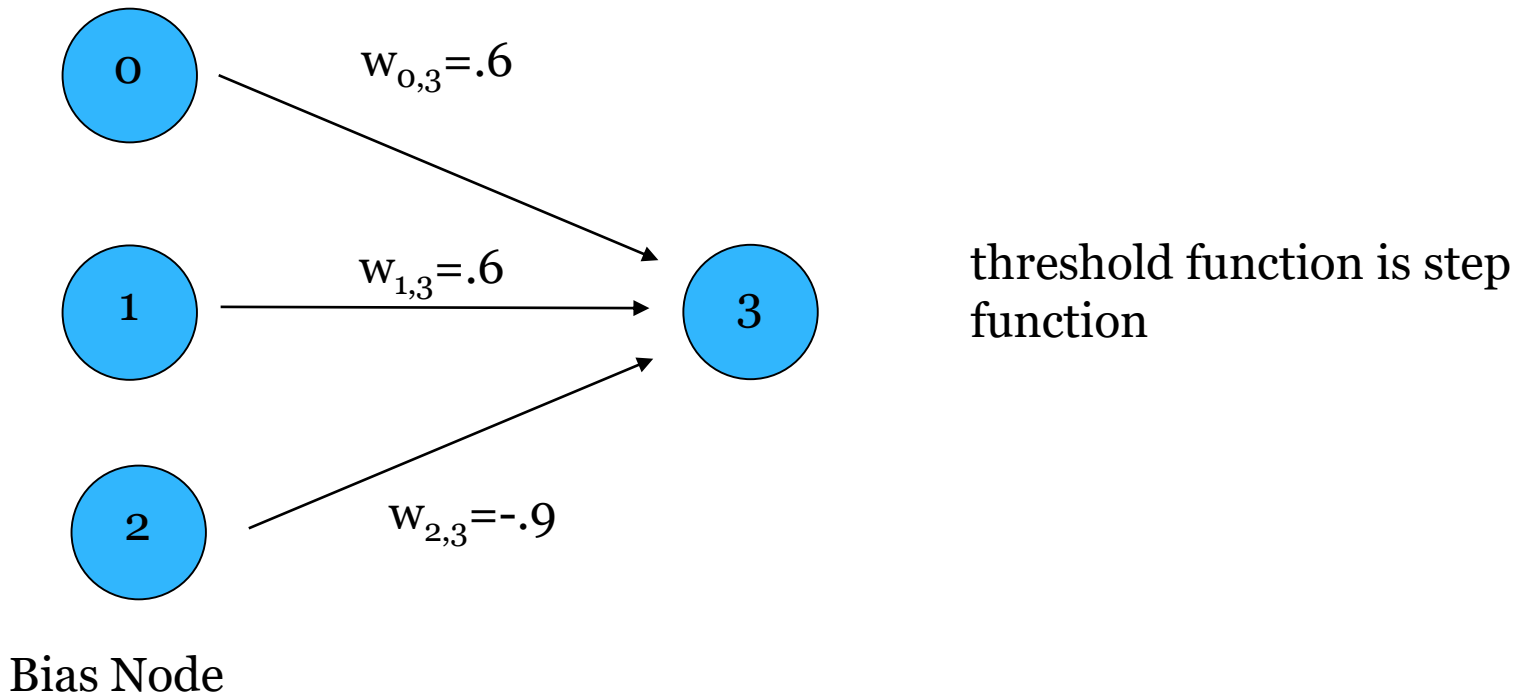
$$V_j(n) = \sum_{i=1}^p (W_{ji} X_i + \text{Threshold})$$

- where X_i is the activation of previous layer neuron i
- W_{ji} is the weight of going from node i to node j
- p is the number of neurons in the previous layer
- Calculate output activation

$$Y_j(n) = \frac{1}{1 + \exp(-V_j(n))}$$

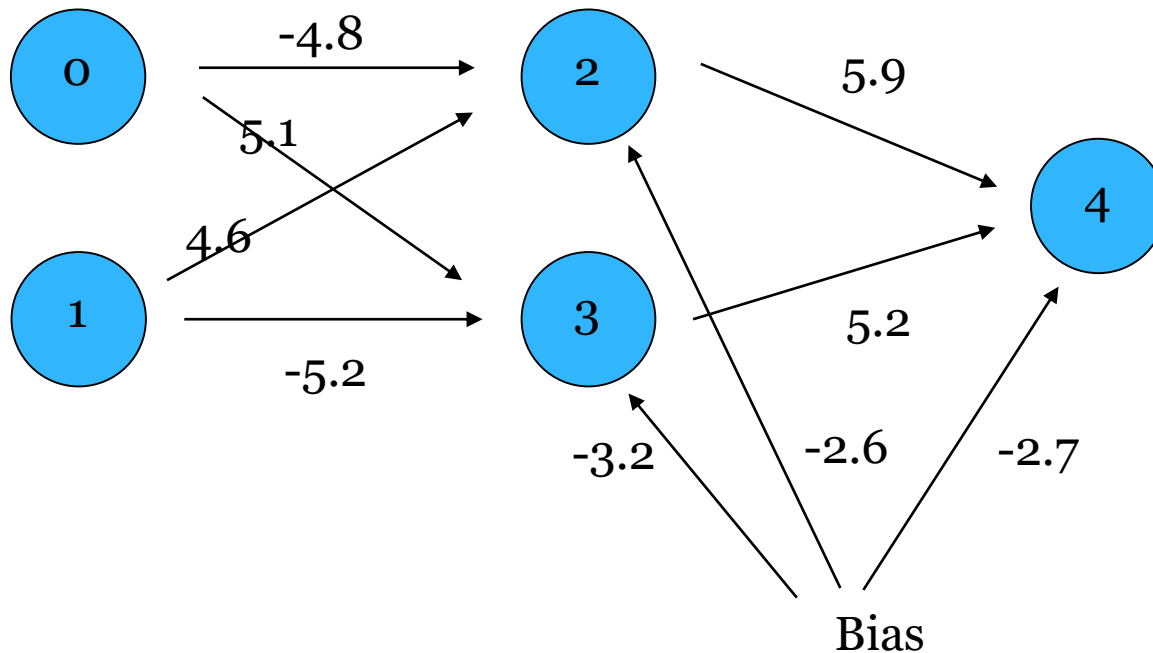
ANN Forward Propagation

- Example: ADALINE Neural Network
 - Calculates and of inputs



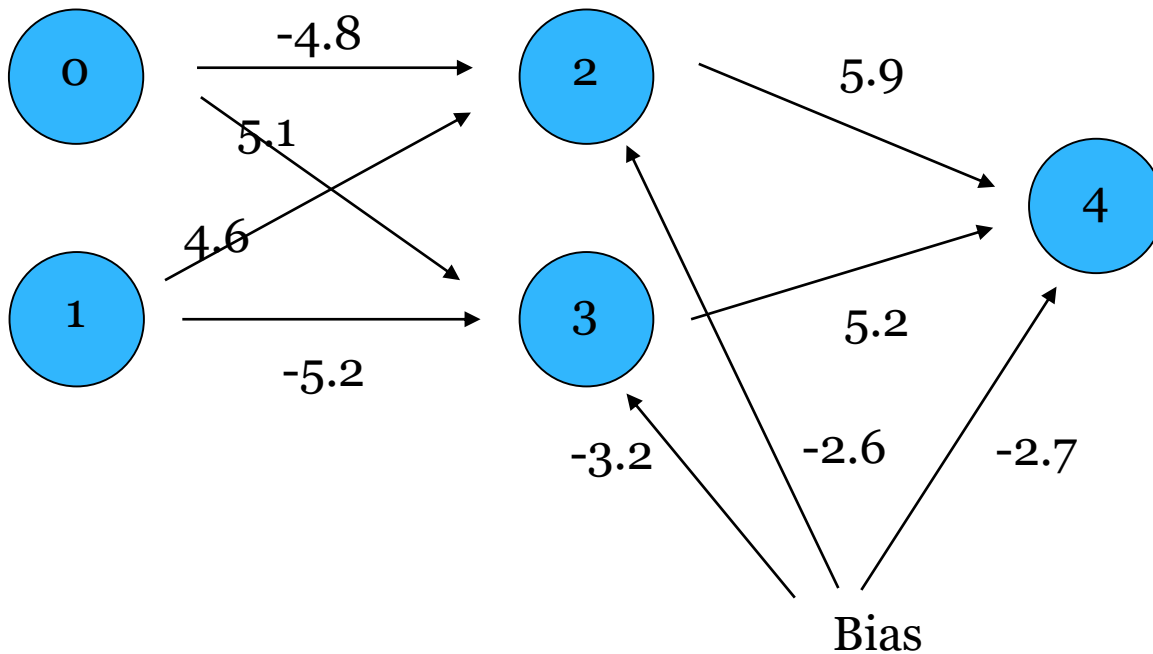
ANN Forward Propagation

- Example: Three layer network
 - Calculates xor of inputs



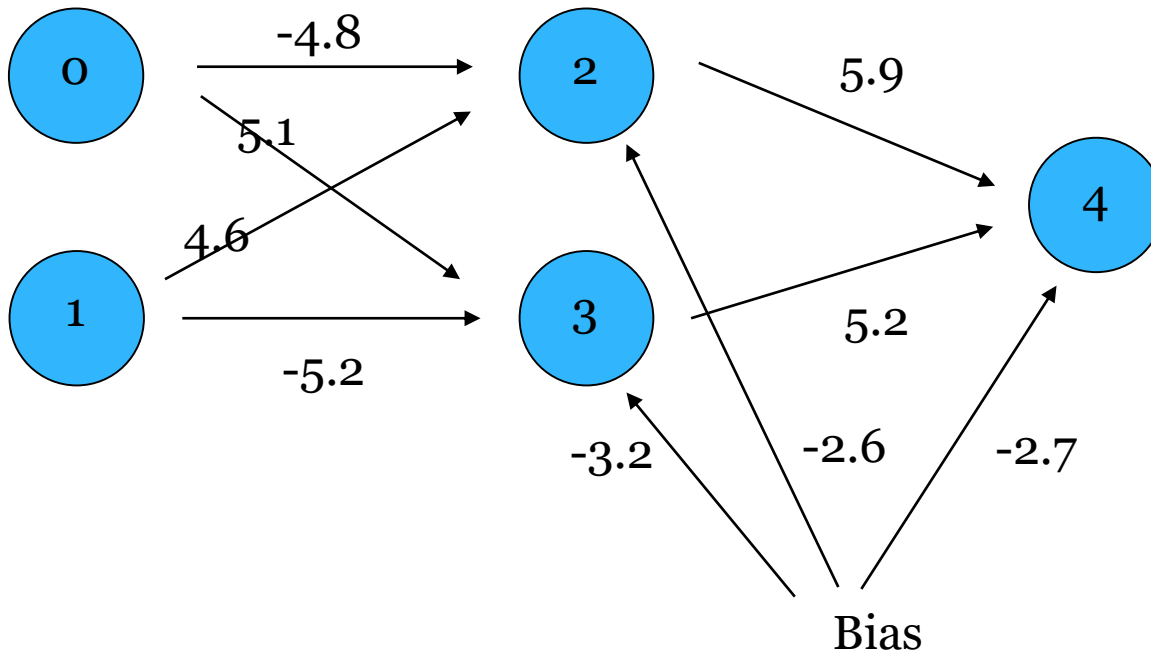
ANN Forward Propagation

- Input (0,0)



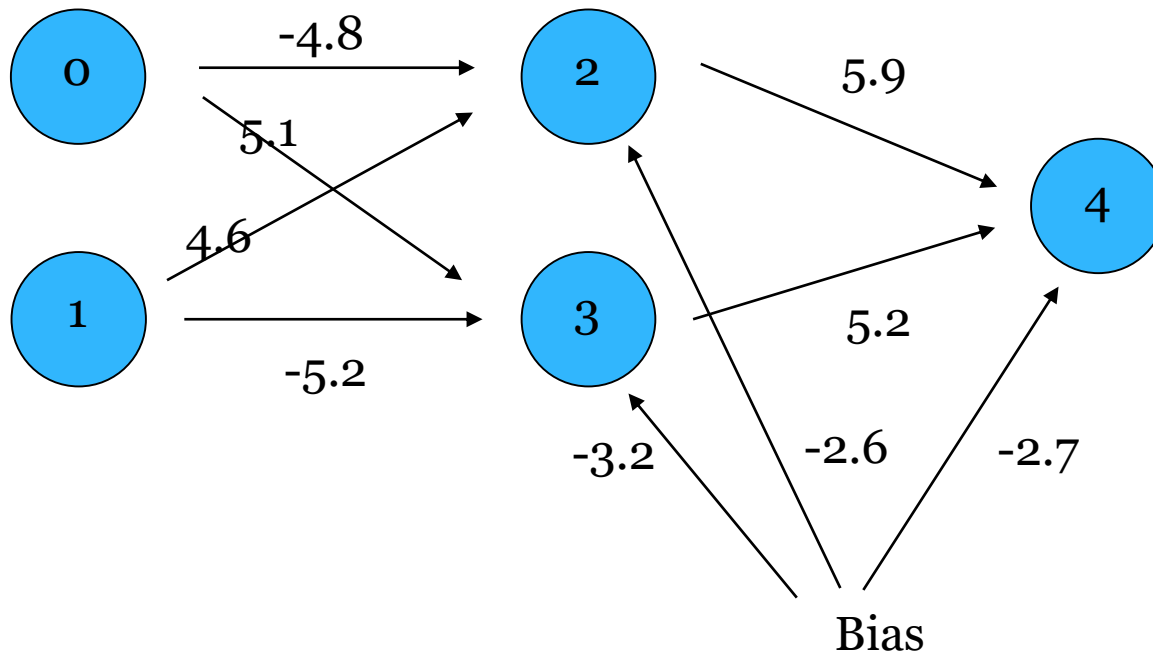
ANN Forward Propagation

- Input (0,0)
 - Node 2 activation is $\varphi(-4.8 \cdot 0 + 4.6 \cdot 0 - 2.6) = 0.0691$



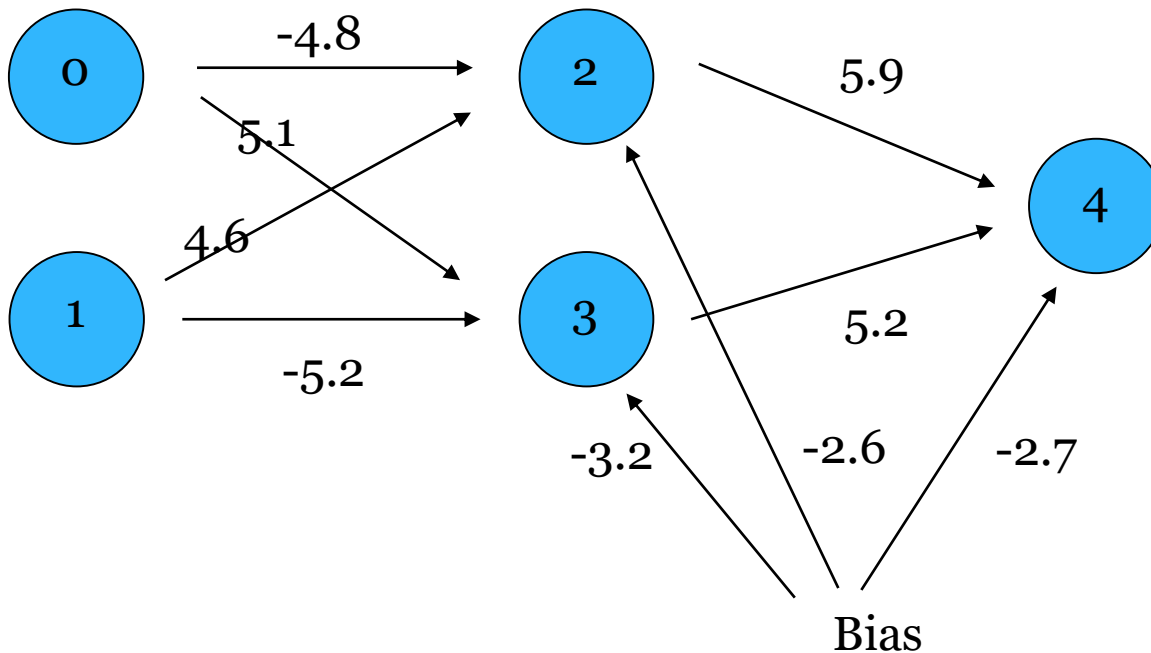
ANN Forward Propagation

- Input (0,0)
 - Node 3 activation is $\phi(5.1 \cdot 0 - 5.2 \cdot 0 - 3.2) = 0.0392$



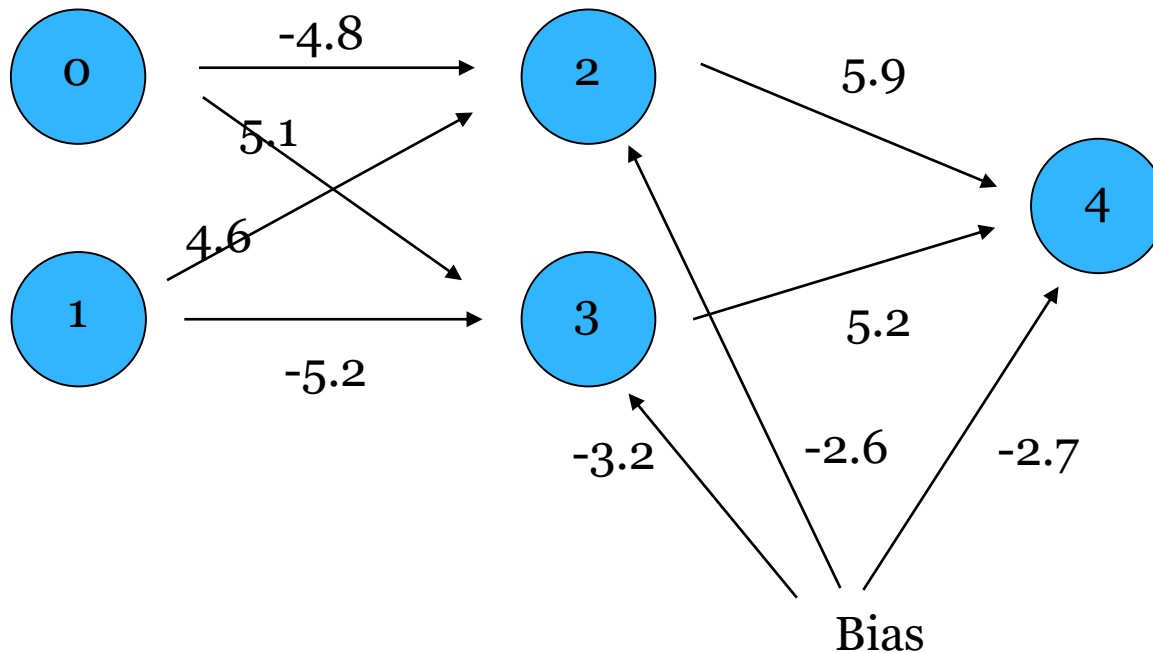
ANN Forward Propagation

- Input (0,0)
 - Node 4 activation is $\phi(5.9 \cdot 0.069 + 5.2 \cdot 0.069 - 2.7) = 0.110227$



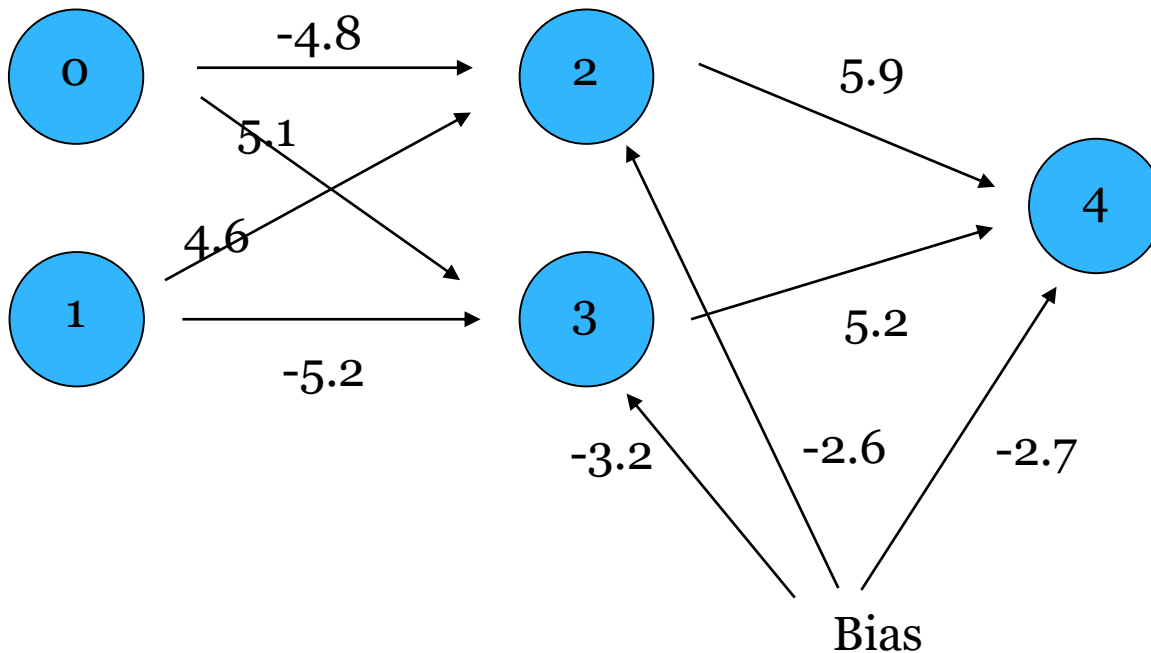
ANN Forward Propagation

- Input (0,1)
 - Node 2 activation is $\phi(4.6 - 2.6) = 0.153269$



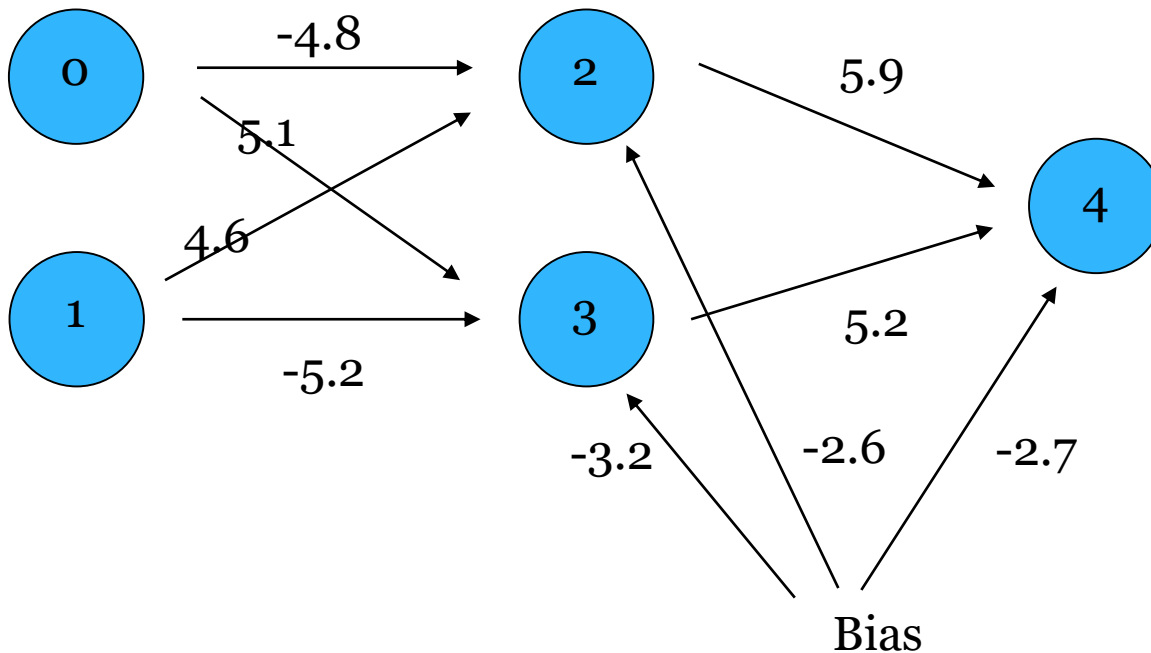
ANN Forward Propagation

- Input (0,1)
 - Node 3 activation is $\phi(-5.2 - 3.2) = 0.000224817$



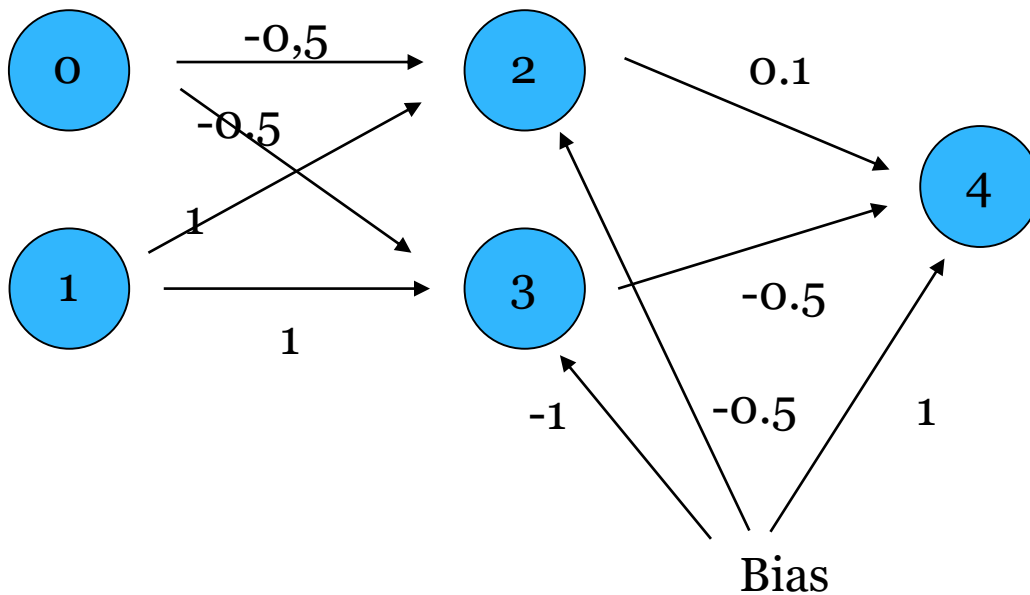
ANN Forward Propagation

- Input (0,1)
 - Node 4 activation is $\phi(5.9 \cdot 0.153269 + 5.2 \cdot 0.000224817 - 2.7) = 0.923992$



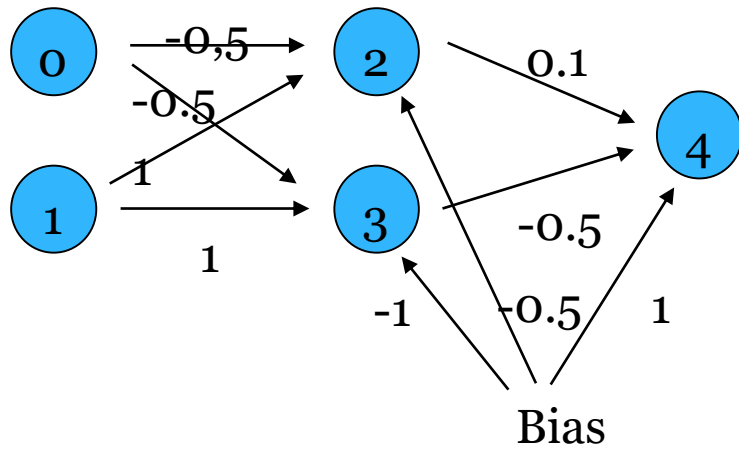
ANN Training Example

- Start out with random, small weights



x1	x2	y
0	0	0.687349
0	1	0.667459
1	0	0.698070
1	1	0.676727

ANN Training Example



x1	x2	y	Error
0	0	0.69	0.472448
0	1	0.67	0.110583
1	0	0.70	0.0911618
1	1	0.68	0.457959

Average Error is 0.283038

Advantages

- Human like thinking
- Handle noisy or missing data
- Can work with large number of variables or parameters
- Provide general solutions with good predictive accuracy

Disadvantages

- black box” nature
- Greater computational processing time
- Hardware dependence
- Determination of proper network structure

Next Week Lecture

- Supervised Learning: Classification with Support Vector Machine (SVM)