What is a Neural Network?

Neural Network (Part 1)

Lecture Video Slides

Pre-requisites

- From Linear Regression:
  - Linear Combination
  - Sum of Squared Residuals (SSR)
- From Logistic Regression:
  - Logistic function
- From CART:
  - Entropy
- From Calculus:
  - First Derivative (Differentiation)
  - Tangent line interpretation

Story:
Open the Green Door.

Imagine Opening a Door and Reacting to Information Signals…

Source: Story in Chew C.H. (2020) AI, Analytics and Data Science Vol 1, Chap 9, Section 9.2.

Story:
Open the Green Door.

Imagine Opening a Door and Reacting to Information Signals…

Small fire on paper.

Source: Story in Chew C.H. (2020) AI, Analytics and Data Science Vol 1, Chap 9, Section 9.2.

Story:
Open the Red Door.

Imagine Opening a Door and Reacting to Information Signals…

Source: Story in Chew C.H. (2020) AI, Analytics and Data Science Vol 1, Chap 9, Section 9.2.

Story:
Open the Red Door.

Imagine Opening a Door and Reacting to Information Signals…

Huge fire everywhere…

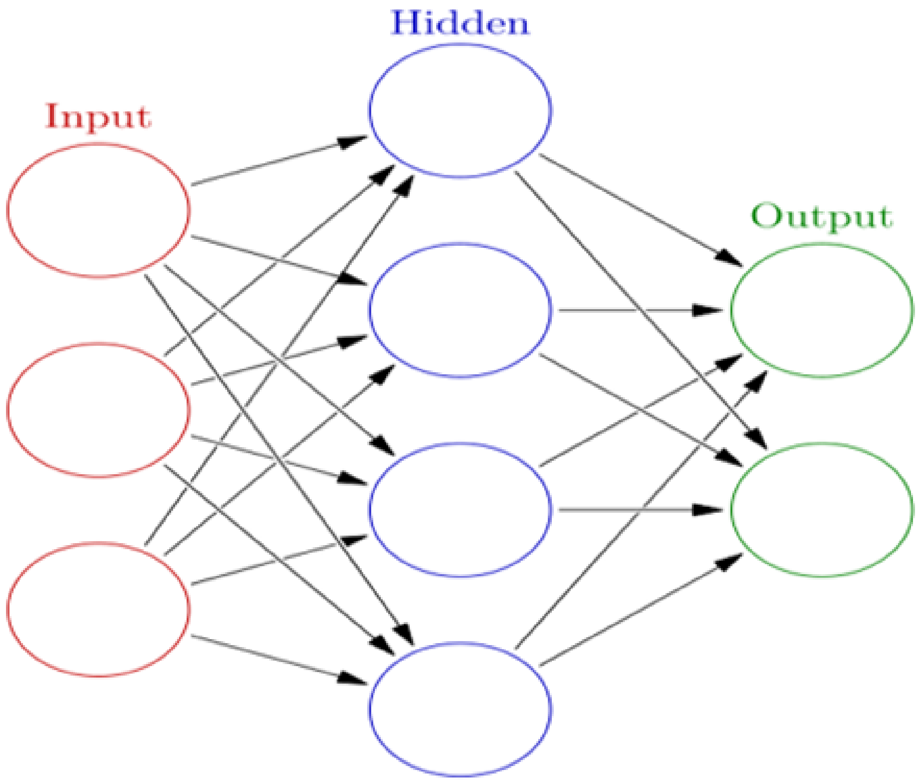
Source: Story in Chew C.H. (2020) AI, Analytics and Data Science Vol 1, Chap 9, Section 9.2.

Processing of Information in the Brain

- The human brain receives information from various sense receptors (eyes, noise, skin, …etc).
- Neurons in the brain process and try to make sense of all the signals received.
- If information is too complex to make sense of immediately, it can be passed forward to another neuron for further processing.
- After sufficient processing, a decision is made.

3-4-2 Neural Network Representation in a Computer

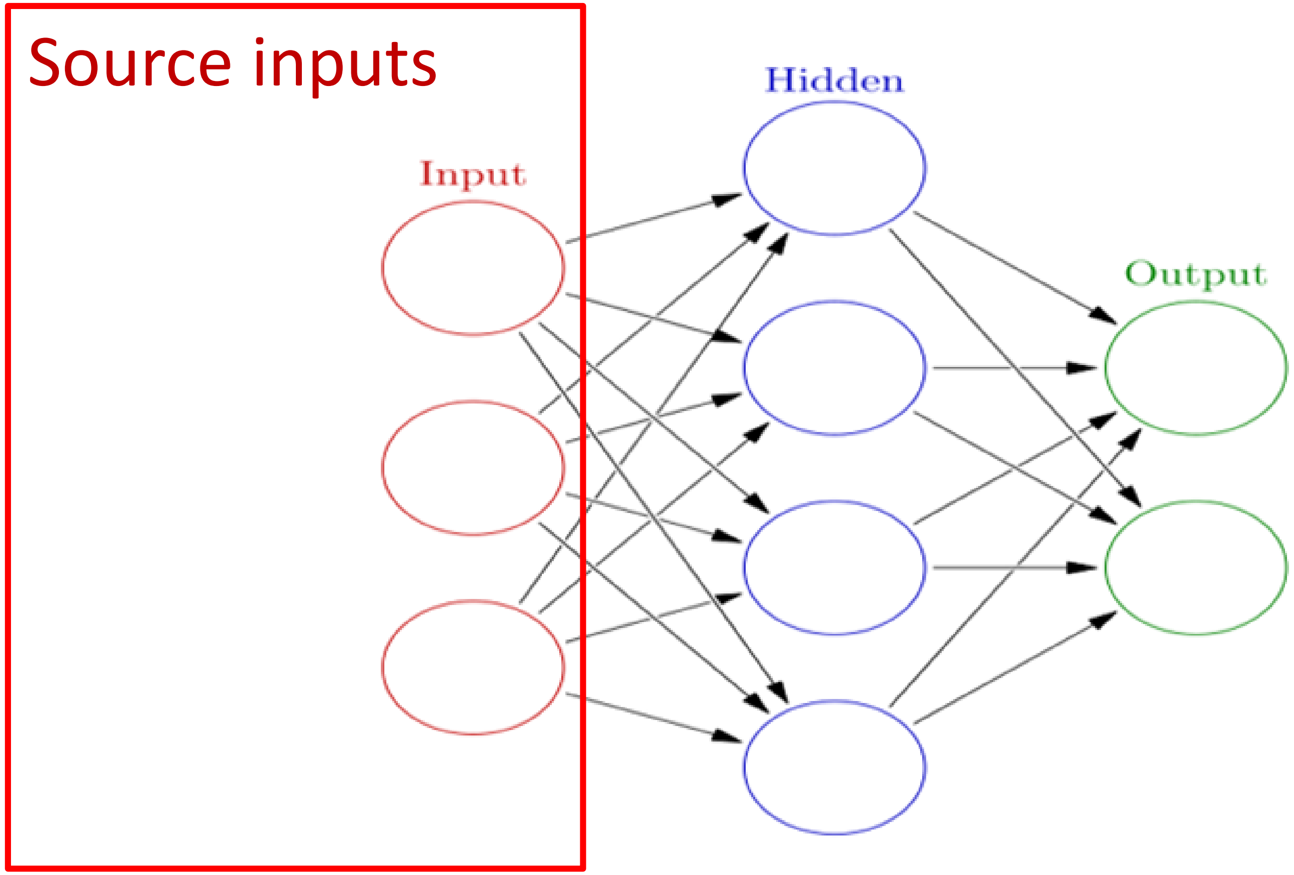
Figure 9.1: A Neural Network with 1 hidden layer (4 hidden nodes)



Source: Chew C.H. (2020) AI, Analytics and Data Science Vol 1, Chap 9.

Input Layer in Neural Network Rep Source

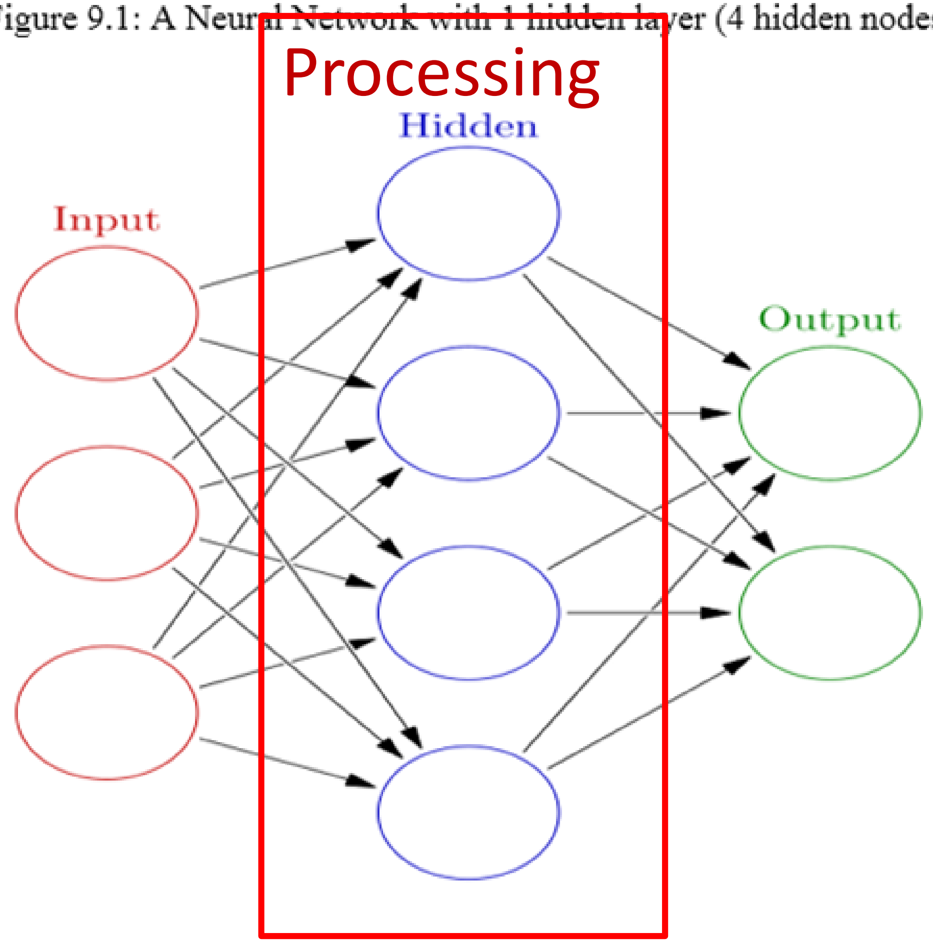
Figure 9.1: A Neural Network with 1 hidden layer (4 hidden nodes)



Source: Chew C.H. (2020) AI, Analytics and Data Science Vol 1, Chap 9.

Hidden Layer(s) in Neural Network Rep Processing

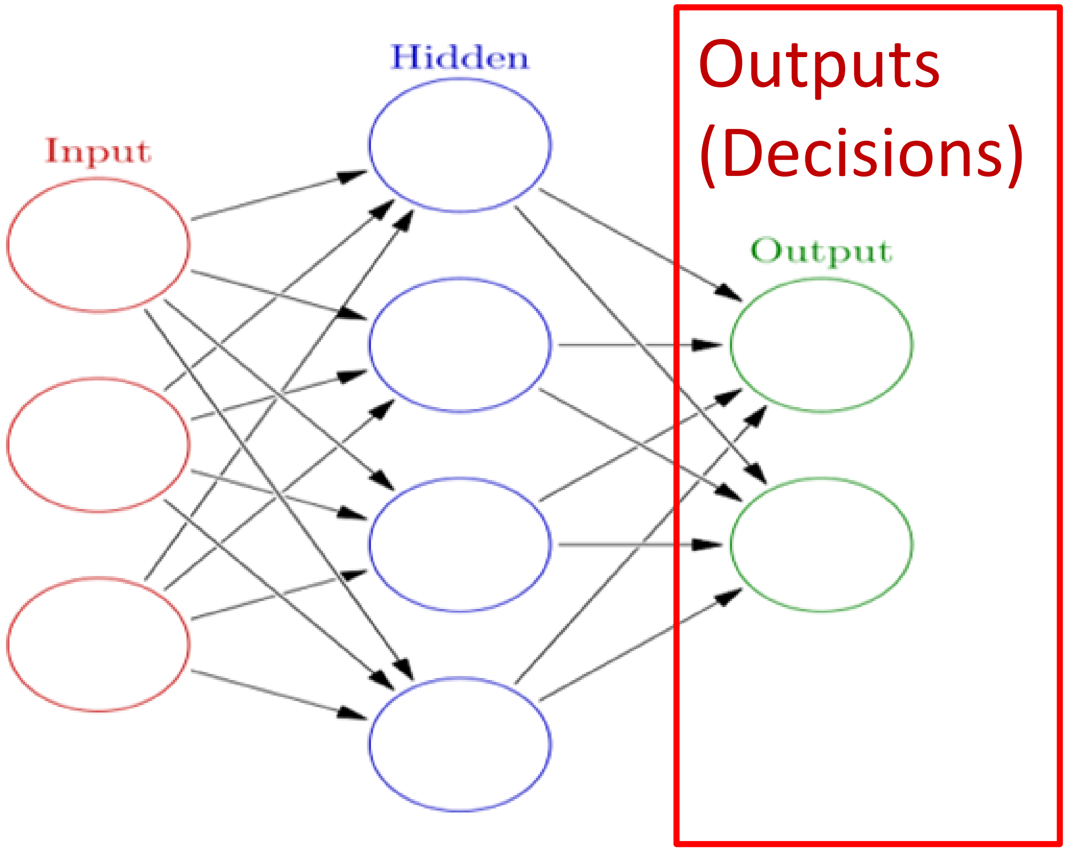
Figure 9.1: A Neural Network with 1 hidden layer (4 hidden nodes)



Source: Chew C.H. (2020) AI, Analytics and Data Science Vol 1, Chap 9.

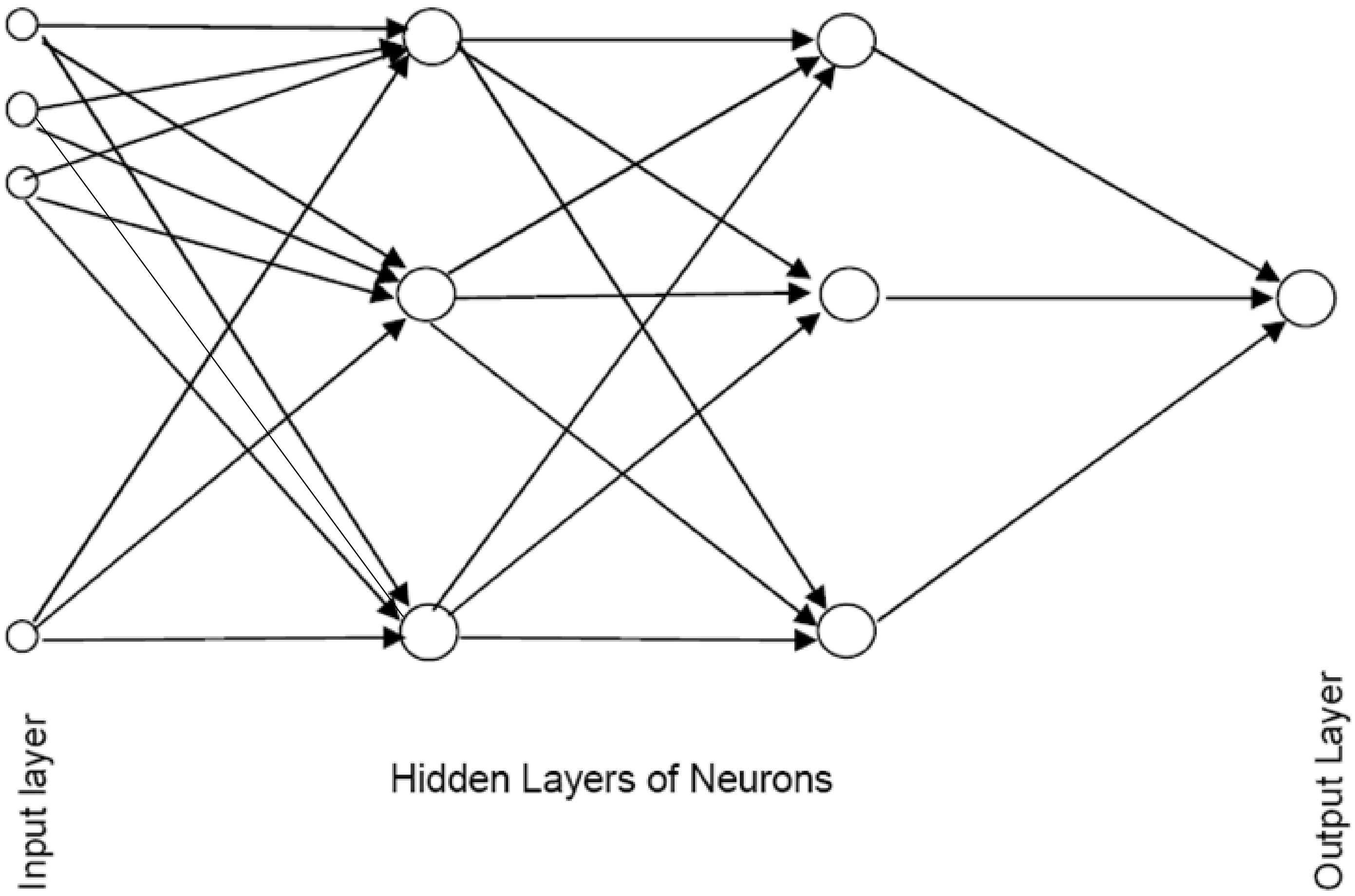
Output Layer in Neural Network Rep Decision(s)

Figure 9.1: A Neural Network with 1 hidden layer (4 hidden nodes)



Source: Chew C.H. (2020) AI, Analytics and Data Science Vol 1, Chap 9.

4-3-3-2 Neural Network



Lines connect neurons from one layer to neurons in the next layer.

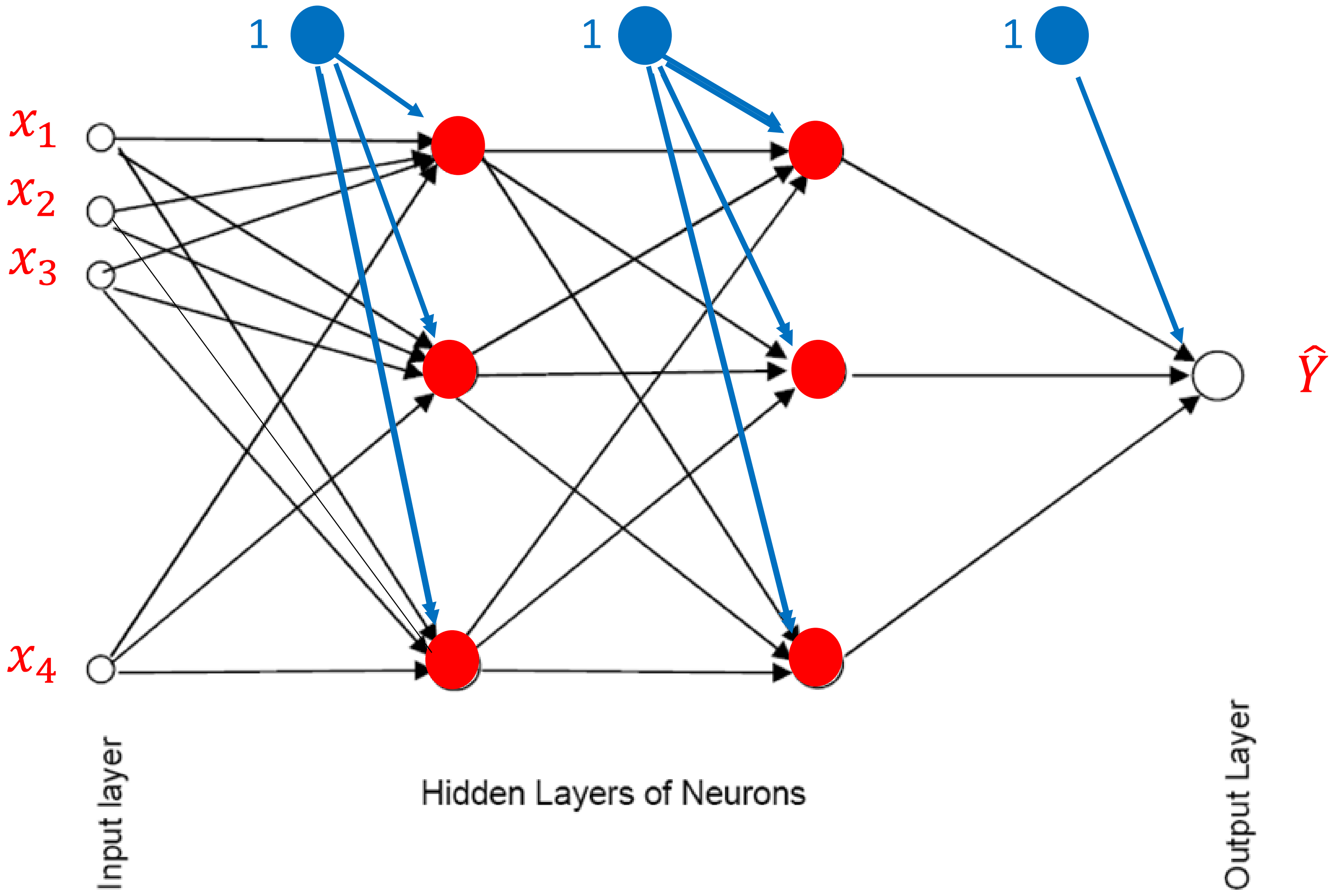
- Nodes within layers act as neurons.
- Basic Neural Network have only a few hidden layer and hidden nodes within each hidden layer.
- Deep learning uses many hidden layers and hidden nodes, and a special activation function.
- Lines with Weights (aka coefficients) connect neurons in one layer to neurons in next layer.
- Weights are the only key parameters.
- Neural Network Parameters:
  - Weight on each line.
  - Bias (aka Intercept) term for each layer after the first layer.

1. Initialized the Neural Network: Set number of hidden layers and hidden nodes.

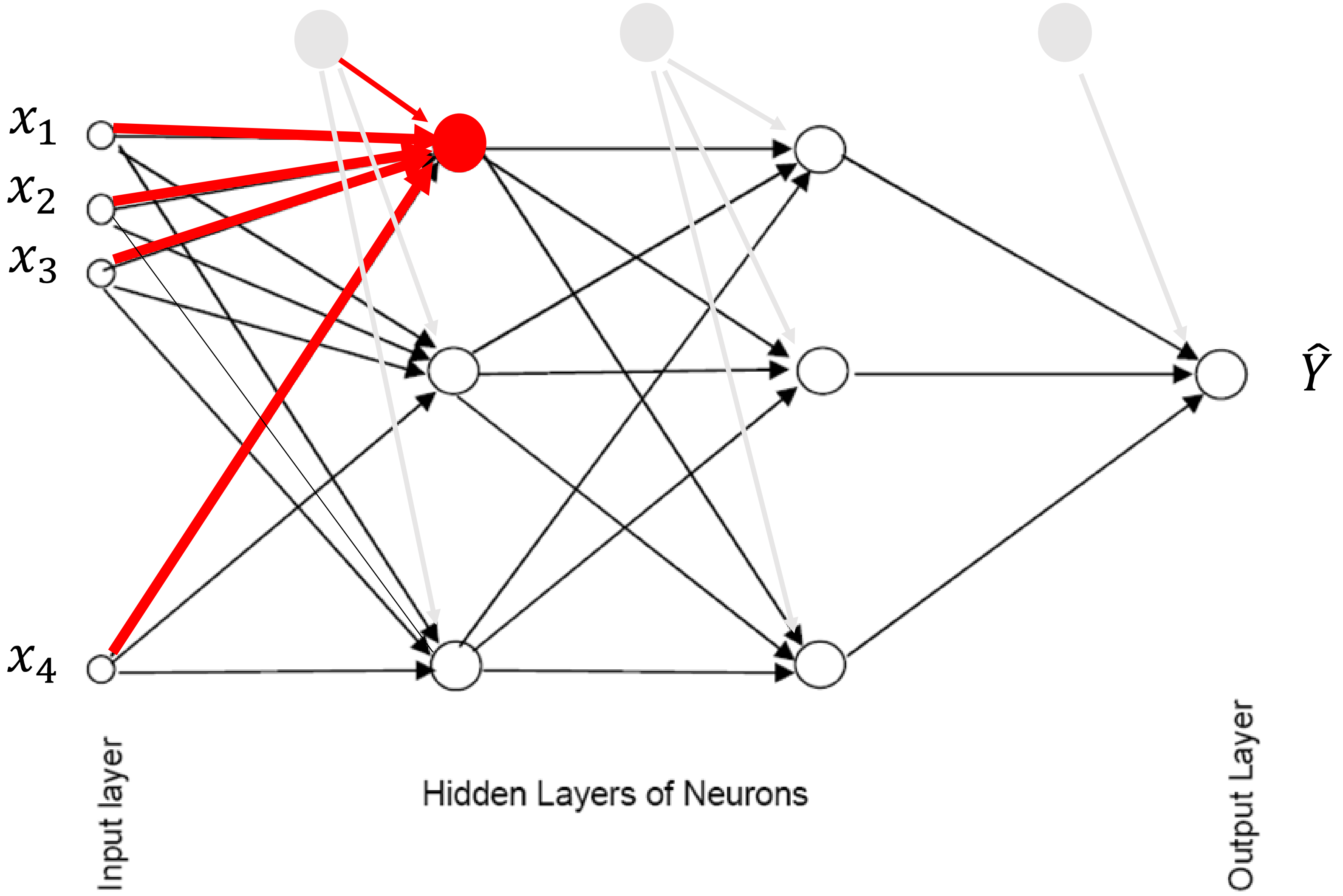
All paths are randomly assigned weights;

Each input variable X is represented as a node in the input layer.

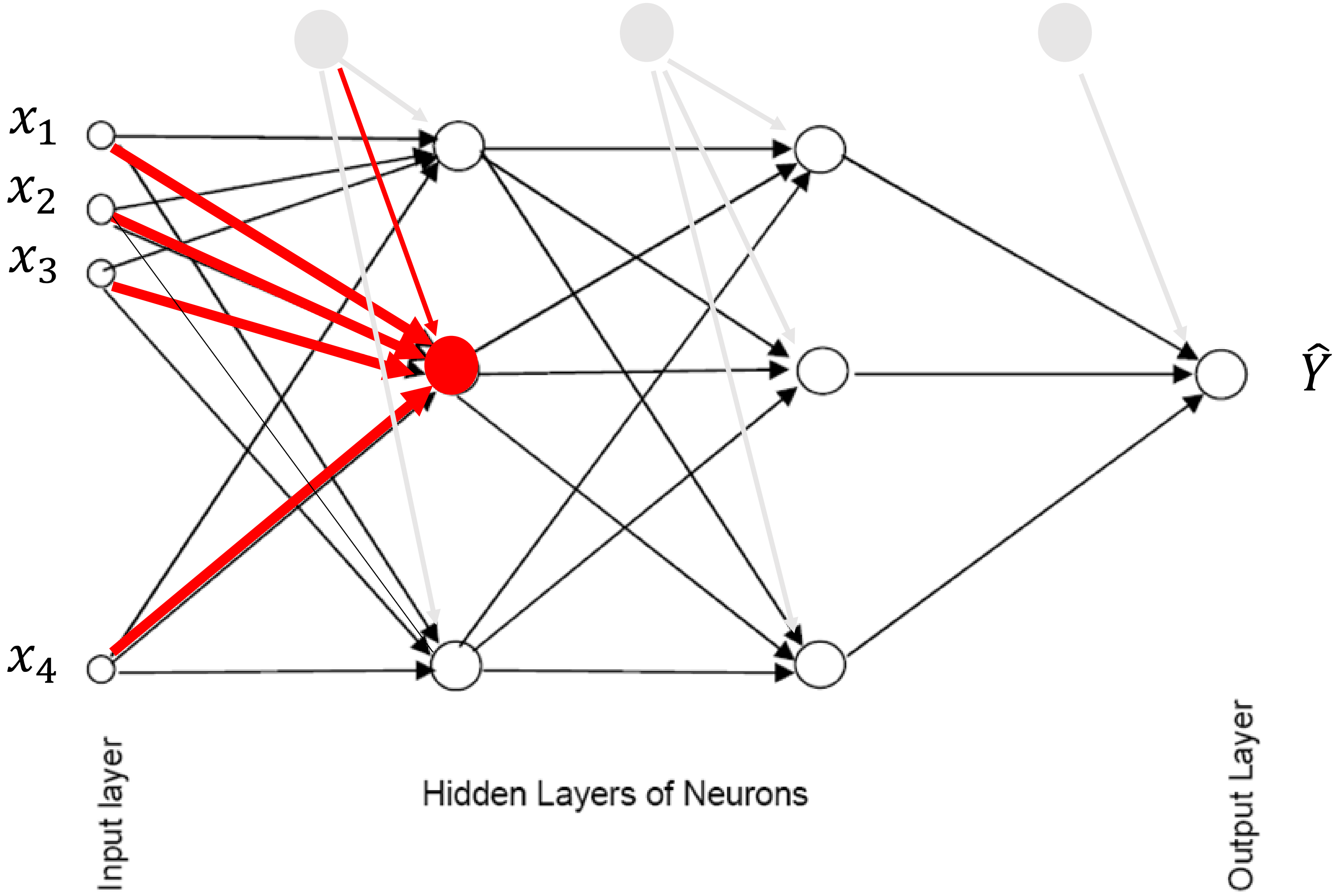
A bias node [blue] is created for each layer after the input layer.



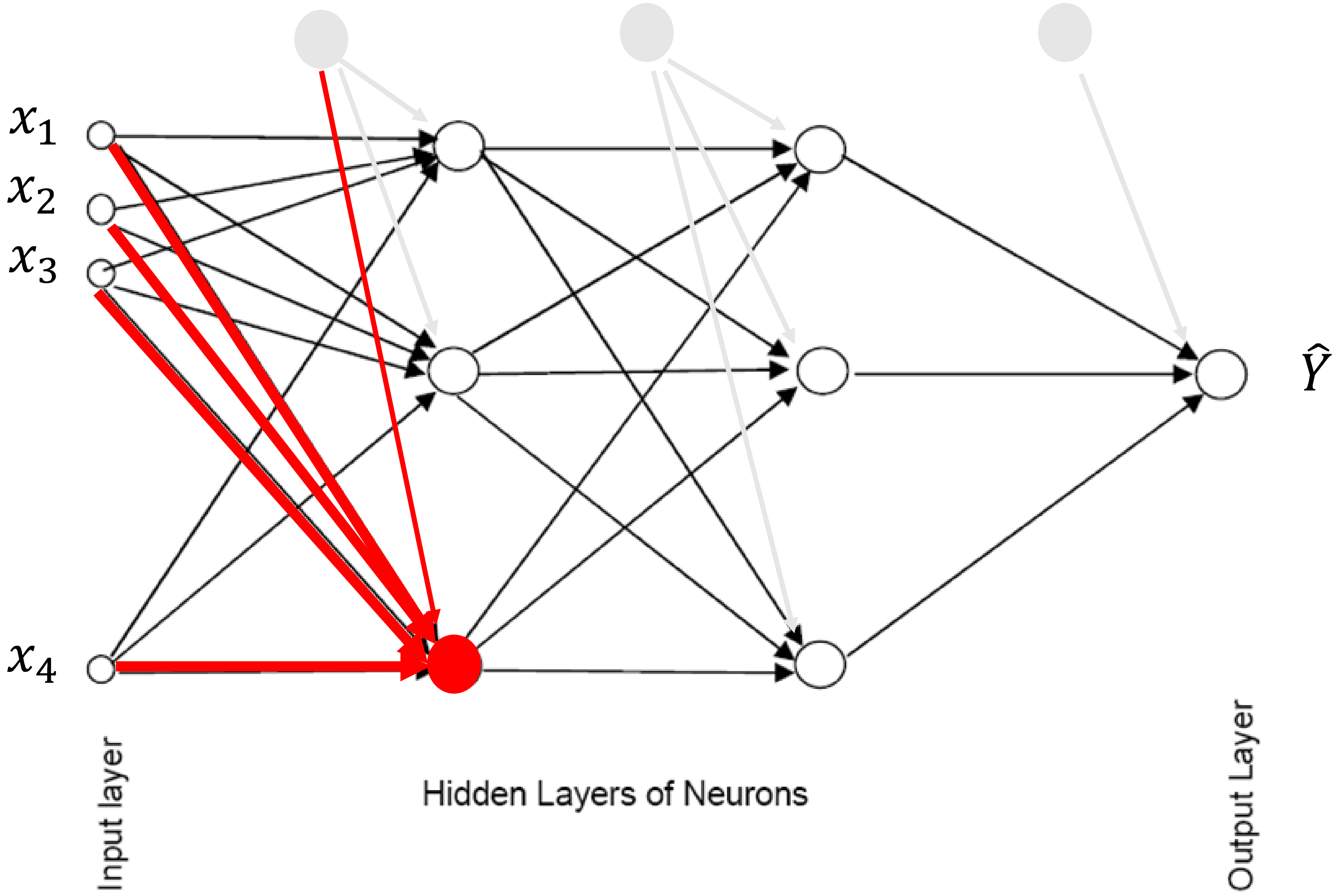
2.1 Transmit <u>weighted</u> information from all nodes in input layer to 1st node in hidden layer 1.



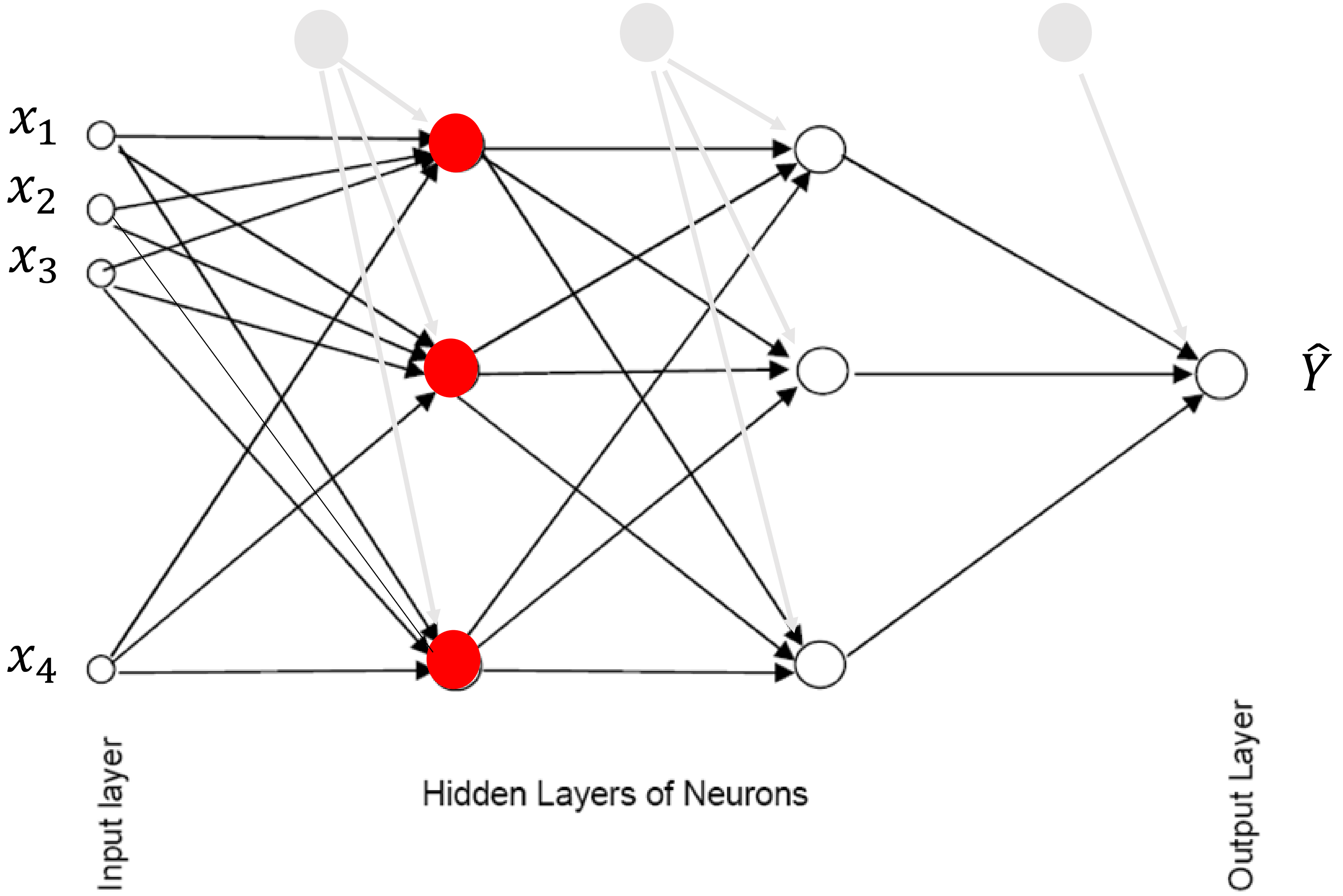
2.2 Transmit <u>weighted</u> information from all nodes in input layer to 2nd node in hidden layer 1.



2.3 Transmit <u>weighted</u> information from all nodes in input layer to 3rd node in hidden layer 1.



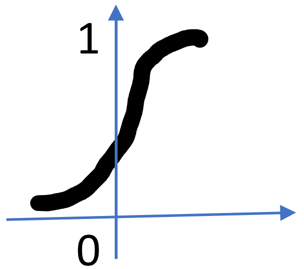
3. All receiving nodes in the hidden layer processed the weighted sum of incoming information via an activation function.



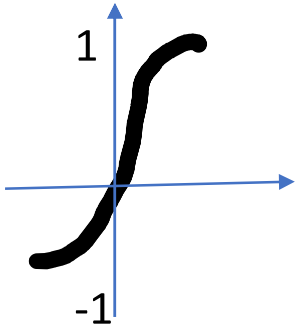
Choice of Activation Functions at Hidden Nodes

- Activation Functions (How all sources of incoming information are combined and processed to form an opinion)
  - Logistic (aka Sigmoidal or Softmax)
  - Hyperbolic Tangent (aka Tanh)
  - ReLU (Rectified Linear Unit)
  - Others

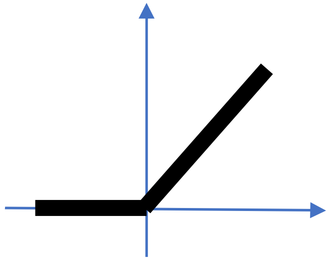
- What does a logistic function look like? Why is it useful?
  - Recall from Logistic Regression class or Textbook (Vol. 1) Chapter 7.



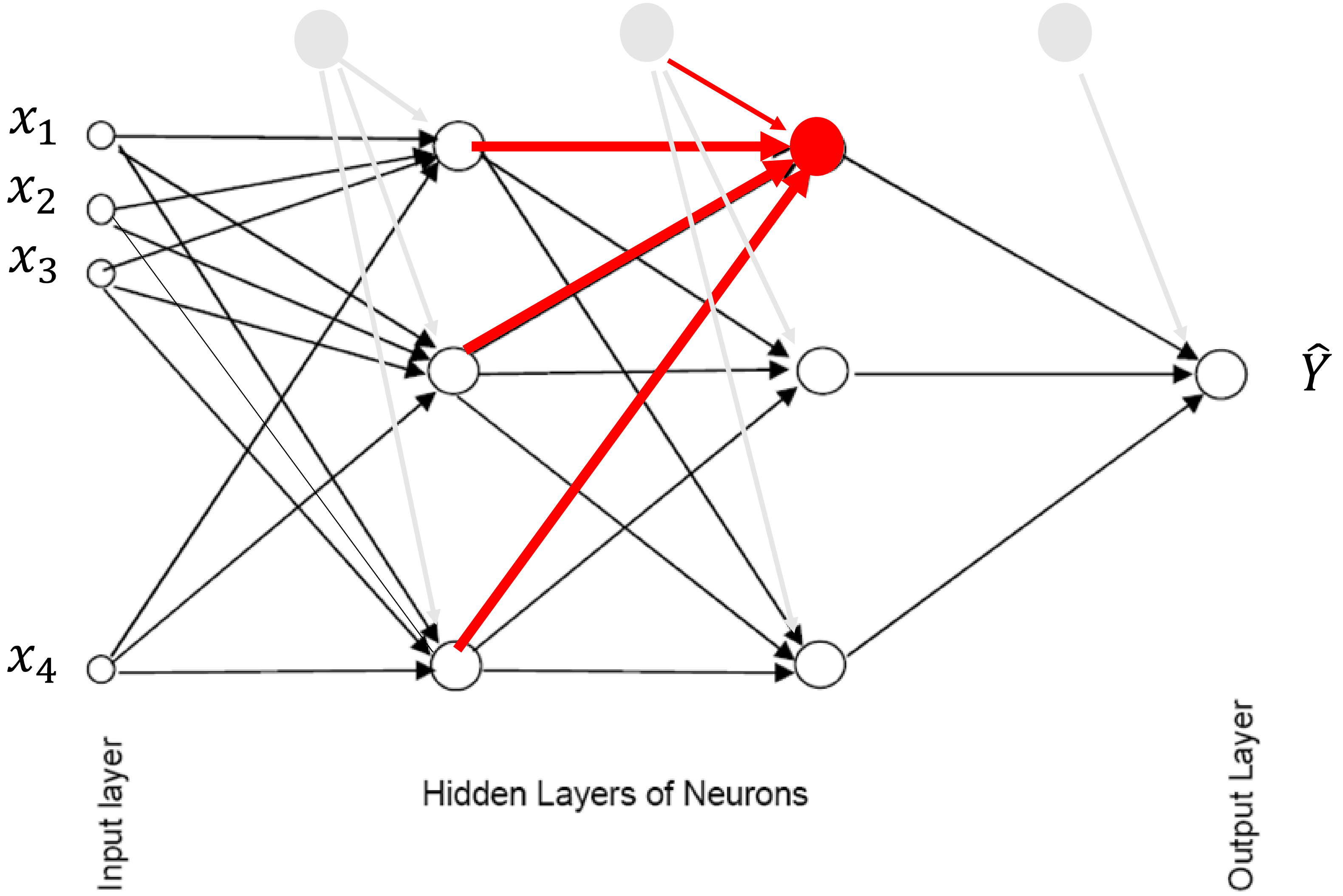
- What does a Tanh function look like?



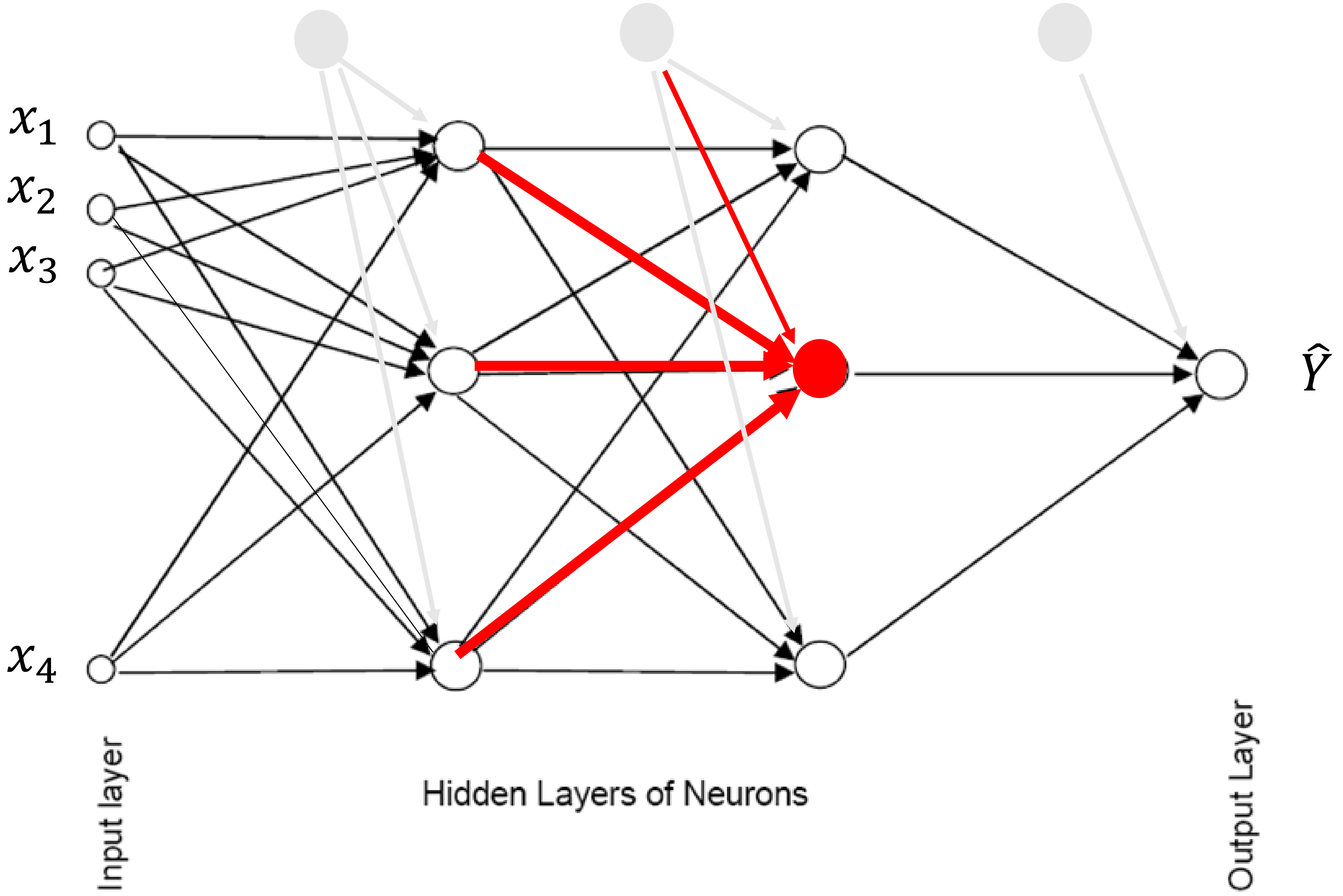
- What does a ReLU function look like?
  - max(0, X)



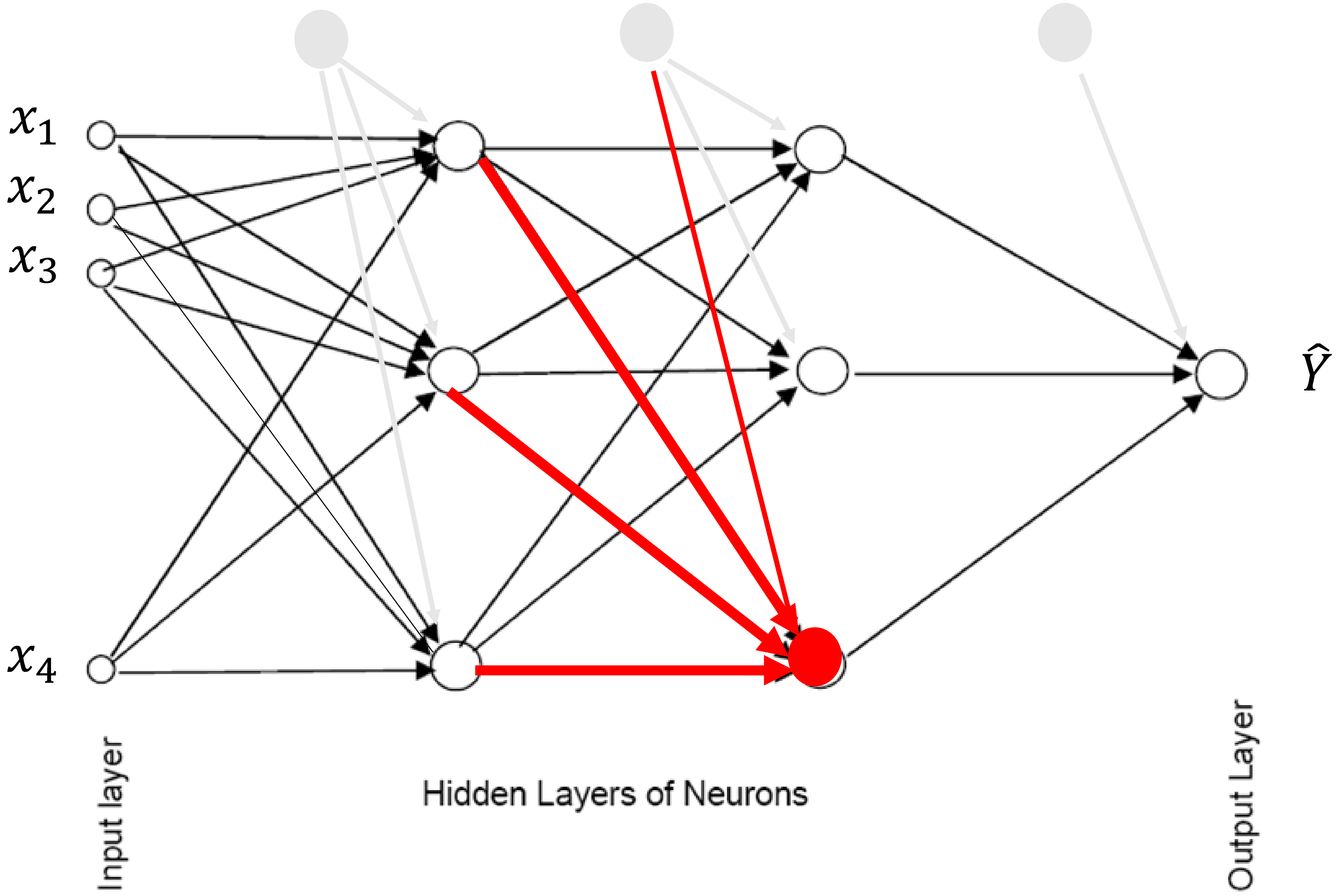
4.1 Transmit processed and <u>weighted</u> information from all nodes in previous hidden layer to 1st node in current hidden layer 2.



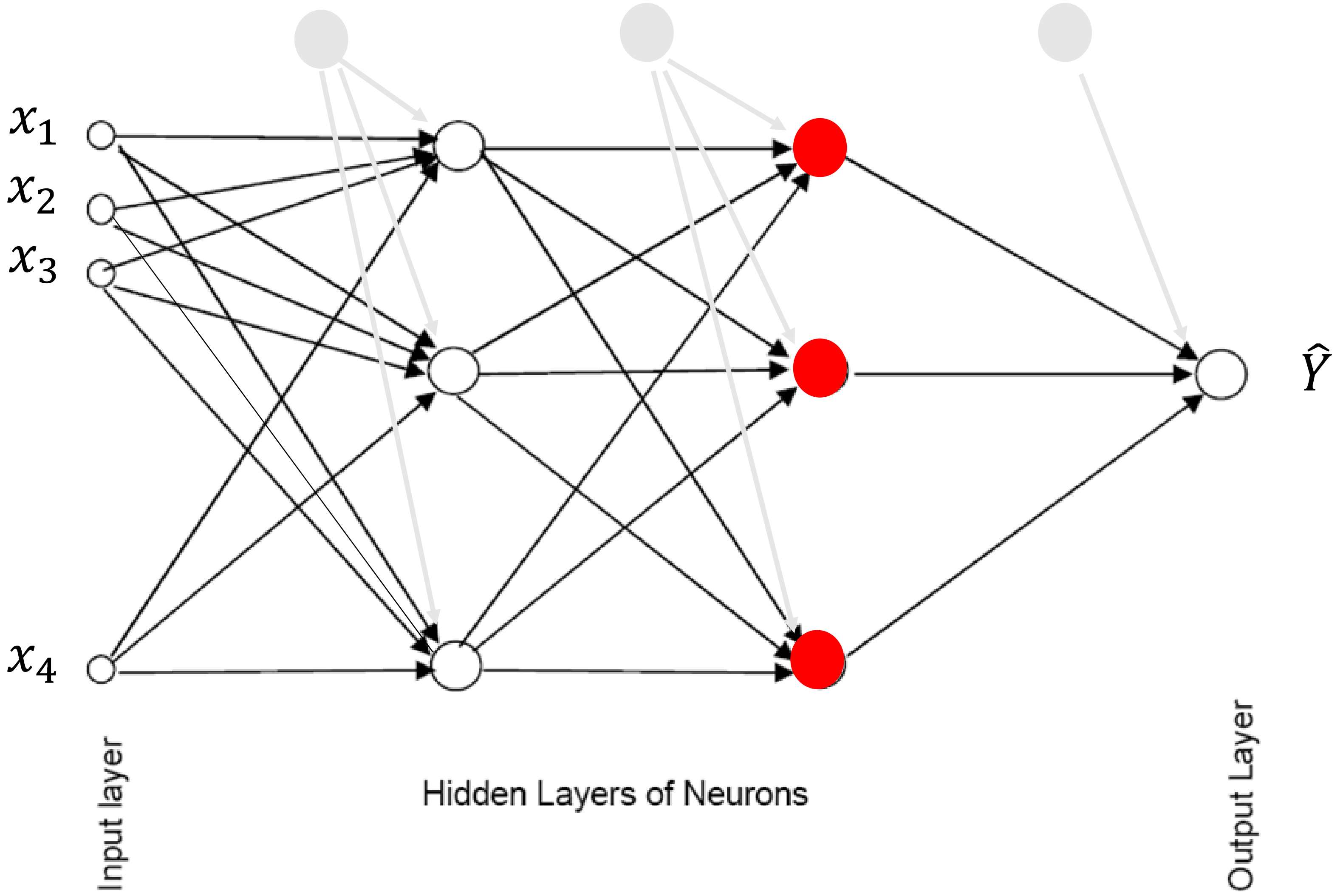
4.2 Transmit processed and <u>weighted</u> information from all nodes in previous hidden layer to 2nd node in current hidden layer 2.



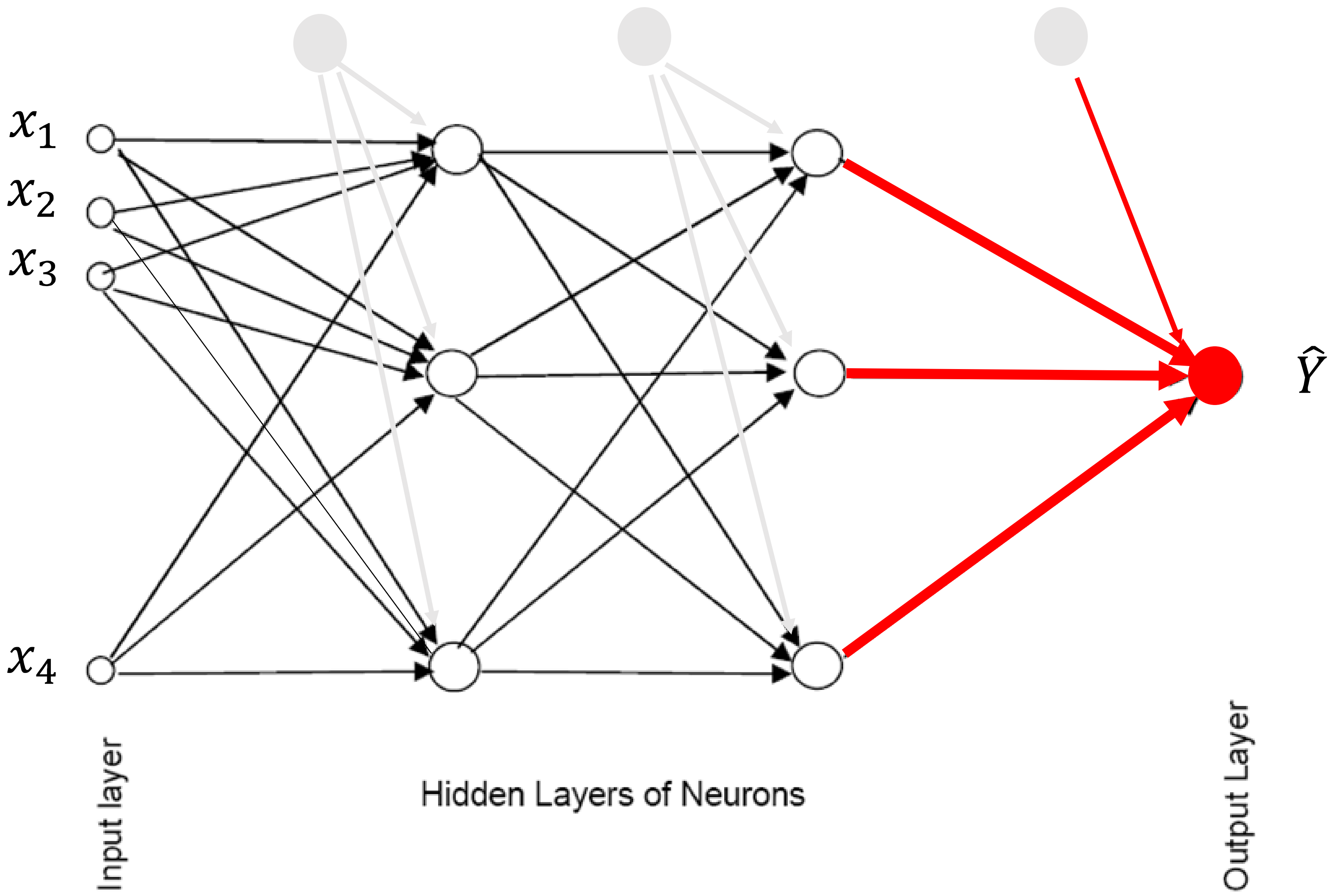
4.3 Transmit processed and <u>weighted</u> information from all nodes in previous hidden layer to 3rd node in current hidden layer 2.



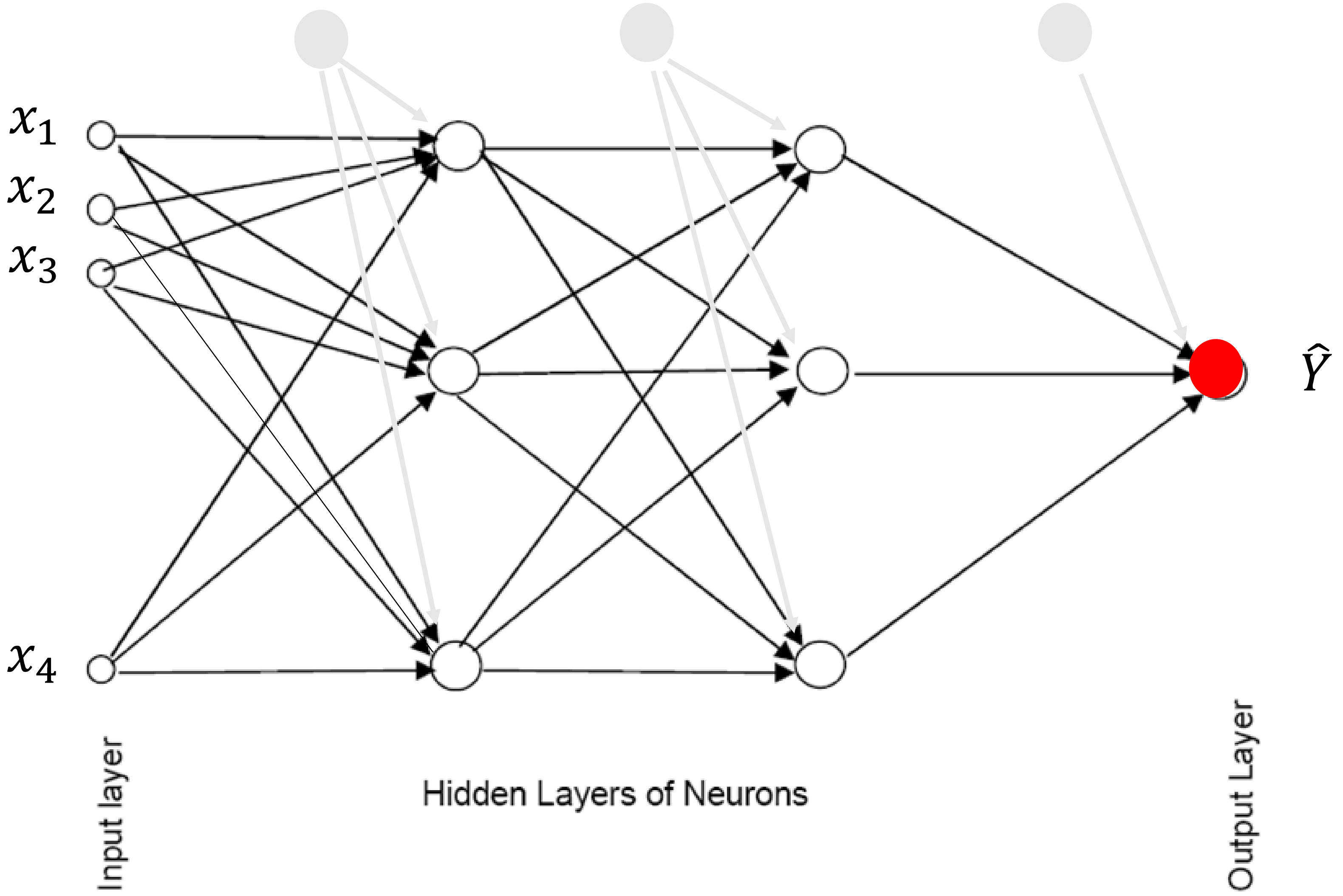
5. All receiving nodes in the current hidden layer processed the weighted sum of incoming information via an activation function.



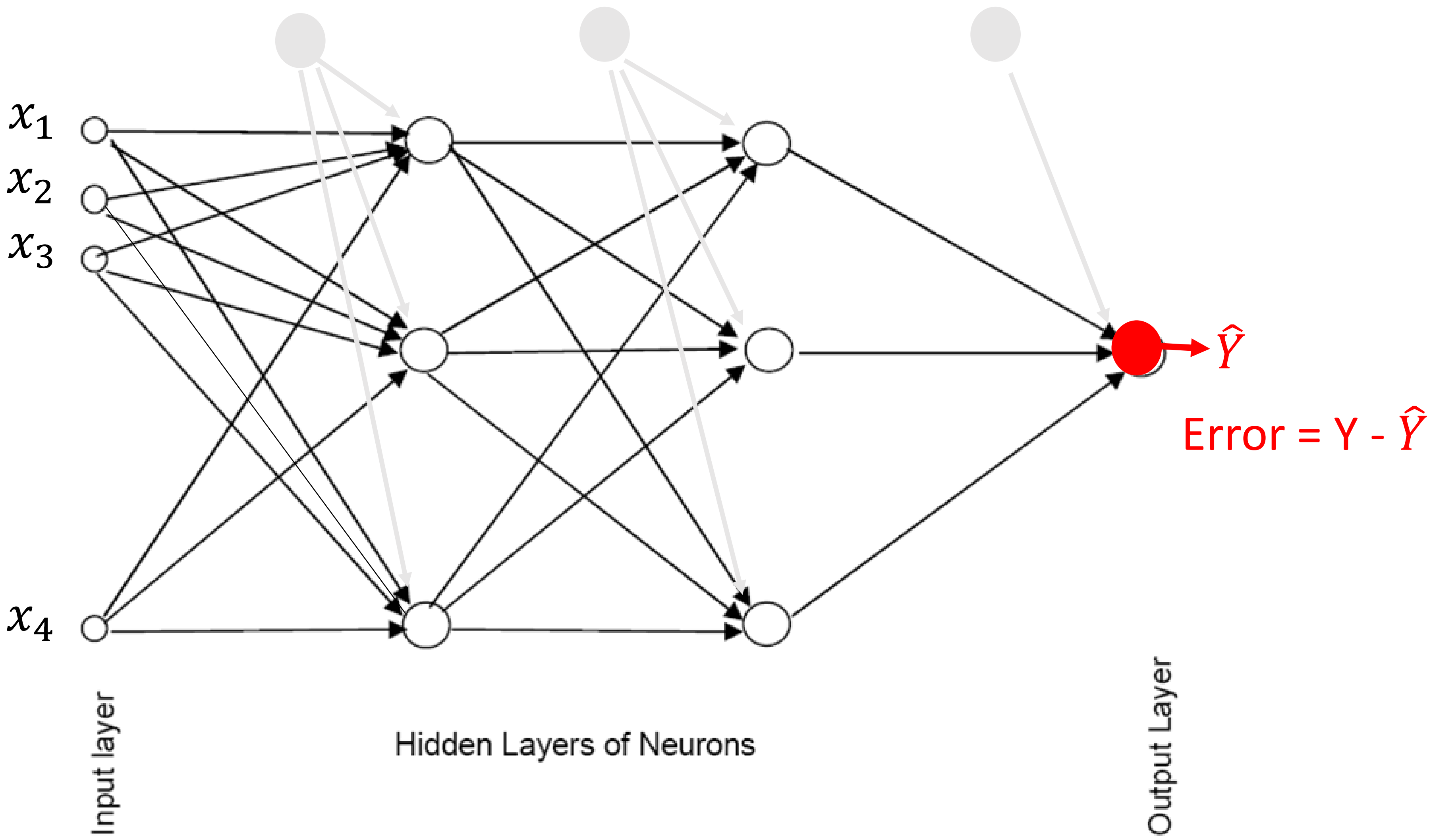
6. Transmit processed and weighted information from all nodes in previous hidden layer to node in output layer.



7. The receiving node in the output layer processed the weighted sum of incoming information via a final activation function.



8. The result of the final activation function determines the predicted value of Y. The error is computed by comparing against actual Y.



Final Activation Function at Output Node(s)

- Depends on the nature of Y
  - Continuous Y
    - Linear
  - Categorical Y
    - Logistic

Error of a Neural Network on a Dataset (n cases)

- Depends on the nature of Y
  - Continuous Y
    - SSE.
  - Categorical Y
    - Confusion Matrix cannot be used. [why?]
    - Cross Entropy (CE).

Cross Entropy as Error Metric for Categorical Y Prediction

Binary Y (0 or 1):

$$CE = -\frac{1}{n}\sum_{i=1}^{n}[y_i \log(\hat{p}_i) + (1 - y_i)\log(1 - \hat{p}_i)]$$

- $\hat{p}_i$: Predicted probability of $y_i = 1$ (from the model)

Multicategory Y with m categories:

$$CE = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{m} y_{ij}\log\left(\hat{p}_{ij}\right)$$

- $\hat{p}_{ij}$: Predicted probability of $y_i = category\ j$ (from the model)

Note:
The negative sign is due to log(prob) < 0.
Negative 1 * negative number will be positive.

Understanding Cross Entropy (Binary Y)

- If Y = 1 and $\hat{p}_i$ = 0.9 (i.e. the model predicts correctly),

$-\{y_i \log(\hat{p}_i) + (1 - y_i) \log(1 - \hat{p}_i)\} = -\{1 \times \log(0.9)\} = -\log(0.9) \approx 0.105$

- If Y = 1 and $\hat{p}_i$ = 0.7 (i.e. the model predicts correctly but less confidently),

$-\{y_i \log(\hat{p}_i) + (1 - y_i) \log(1 - \hat{p}_i)\} = -\{1 \times \log(0.7)\} = -\log(0.7) \approx 0.357$.
The error is higher.

- If Y = 1 and $\hat{p}_i$ = 0.2 (i.e. the model predicts wrongly),

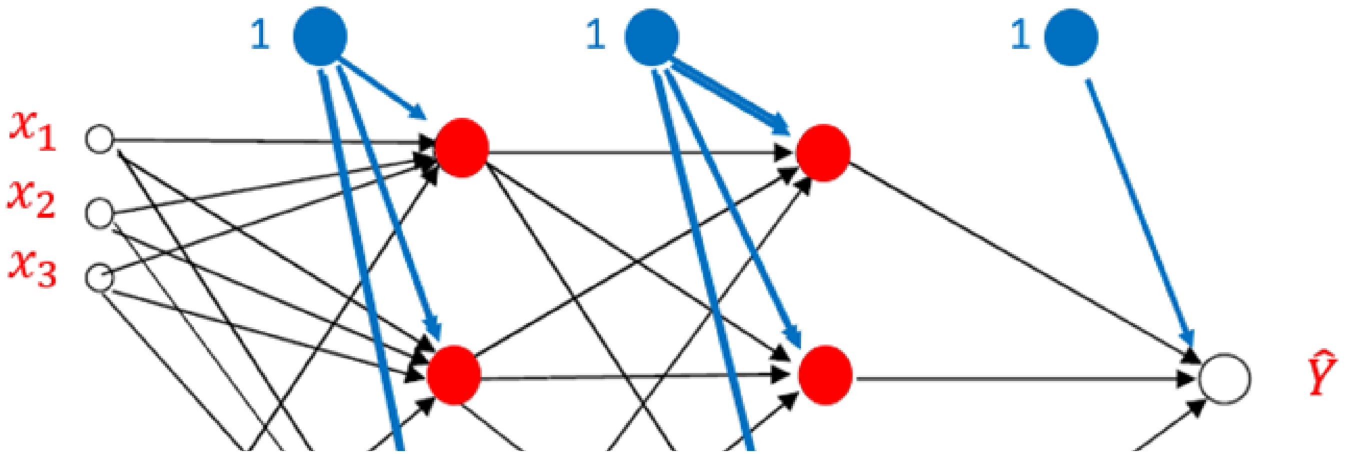
$-\{y_i \log(\hat{p}_i) + (1 - y_i) \log(1 - \hat{p}_i)\} = -\{1 \times \log(0.2)\} = -\log(0.2) \approx 1.609$.
The error is much higher!

Thus if Y = 1, errors depend on $\hat{p}_i$. The closer $\hat{p}_i$ is to 1, the smaller the error and vice versa.

Similarly if Y = 0, errors depend on $\hat{p}_i$. The closer $\hat{p}_i$ is to 0, the smaller the error and vice versa.

All paths use randomized weights initially. How to optimise?

1. Initialized the Neural Network: Set number of hidden layers and hidden nodes.
   All paths are randomly assigned weights;
   Each input variable X is represented as a node in the input layer.
   A bias node [blue] is created for each layer after the input layer.



Next Video

- How optimal weights can be found.
- Simple Example.
- Rcode.