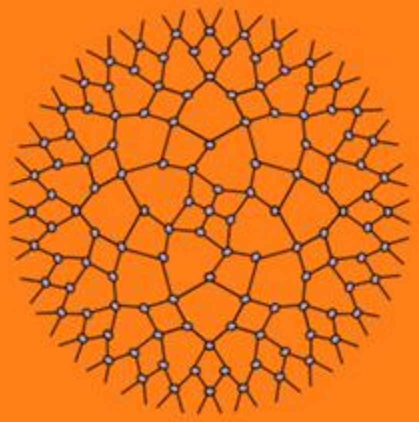
ML Algorithms

NEURAL NETWORKS



Class

A Detailed Look At Neural Networks

Topic

Estimating A Neural Network

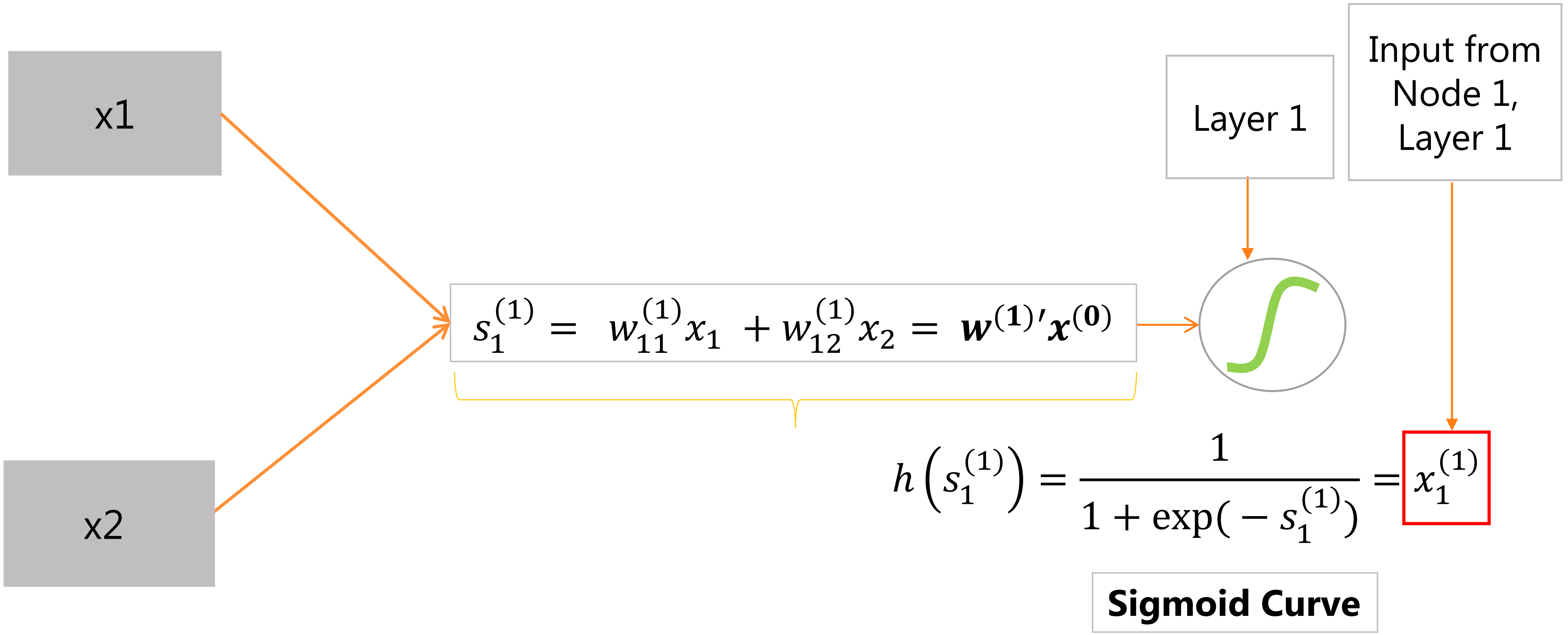
Estimating a Neural Network

How to estimate parameters in a network of neurons

- The neural network has a cost function
- It is a generalization of the cost function for a single neuron

- L layers (L – 2 hidden layers)
- $d^{(l)}$ nodes in Layer l, $0 \leq l \leq L$
- Weights: $w_{ij}^{(l)},\ i \leq d^{(l-1)}$ & $j \leq d^{(l)}$

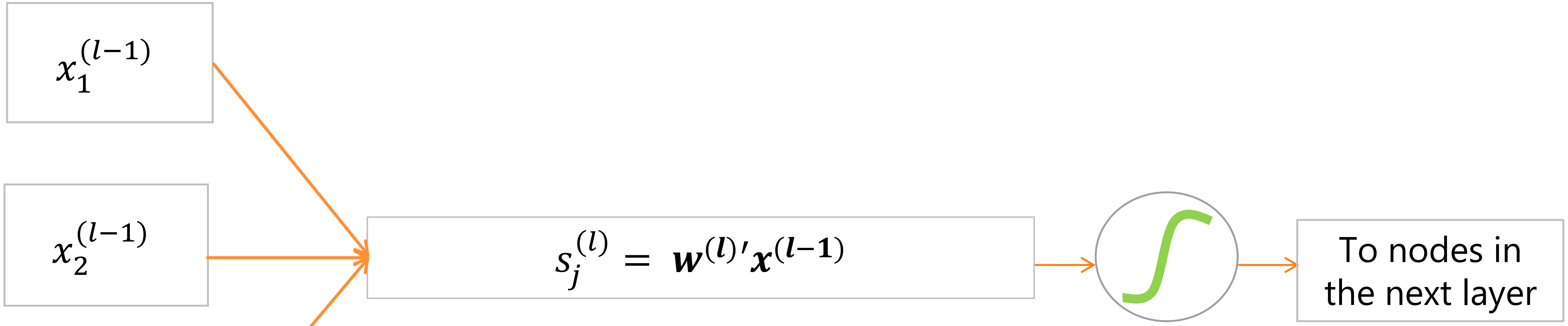
Estimating a Neural Network



$$s_1^{(1)} = w_{11}^{(1)} x_1 + w_{12}^{(1)} x_2 = \boldsymbol{w}^{(1)\prime} \boldsymbol{x}^{(0)}$$

$$h\left(s_1^{(1)}\right) = \frac{1}{1 + \exp(-s_1^{(1)})} = \boxed{x_1^{(1)}}$$

Estimating a Neural Network



$$h\left(s_{\mathrm{j}}^{(1)}\right) = \frac{1}{1 + \exp(-s_{\mathrm{j}}^{(1)})} = x_{\mathrm{j}}^{(1)}$$

$$x_i^{(l)} = F(\boldsymbol{w}^{(1)}, \boldsymbol{w}^{(2)}, \dots . \boldsymbol{w}^{(l)})$$

Estimating a Neural Network

This is the feed forward mechanism

$x_1^{(l-1)}$

$x_2^{(l-1)}$

$x_i^{(l-1)}$

$$s_j^{(l)} = \boldsymbol{w}^{(l)\prime}\boldsymbol{x}^{(l-1)}$$

l + 1

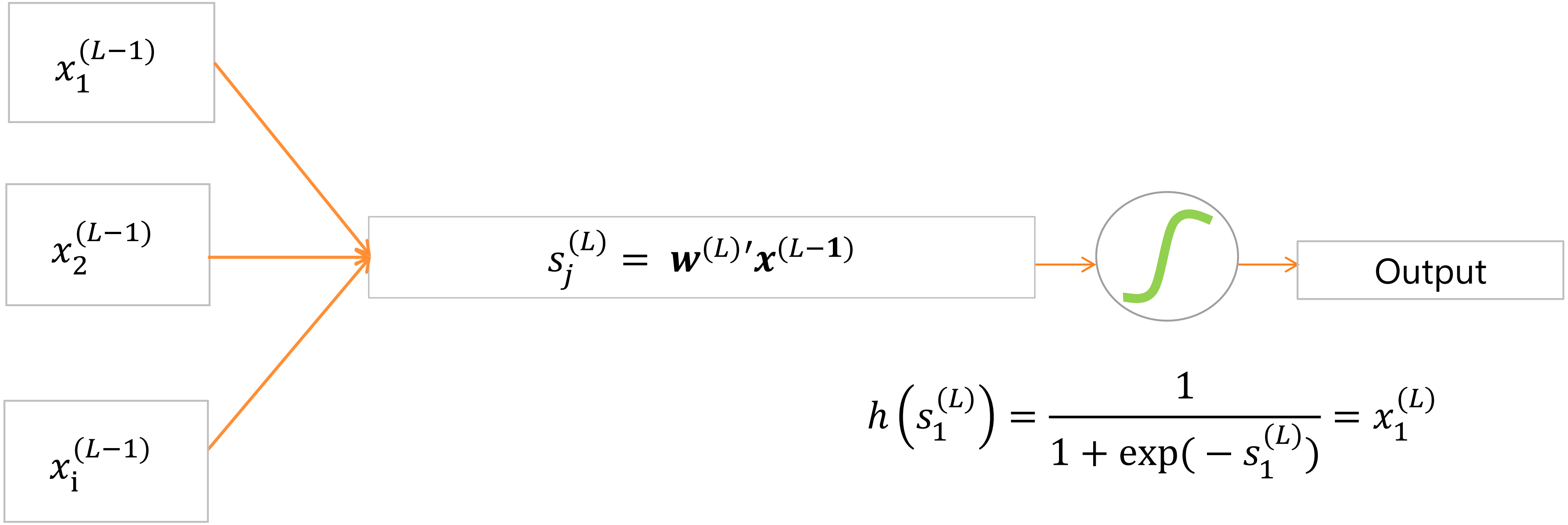
To nodes in the next layer

$$h\left(s_{\mathrm{j}}^{(1)}\right) = \frac{1}{1 + \exp(-s_{\mathrm{j}}^{(1)})} = \boxed{x_{\mathrm{j}}^{(1)}}$$

$$x_i^{(l)} = F(\boldsymbol{w}^{(1)}, \boldsymbol{w}^{(2)}, \ldots.\boldsymbol{w}^{(l)})$$

Estimating a Neural Network

What the network looks like in the final 2 layers



$$h\left(s_1^{(L)}\right) = \frac{1}{1 + \exp(-s_1^{(L)})} = x_1^{(L)}$$

Estimating a Neural Network

A neural network that only performs binary classifications has 1 output node: L

$x_1^{(L-1)}$

$x_2^{(L-1)}$

$x_i^{(L-1)}$

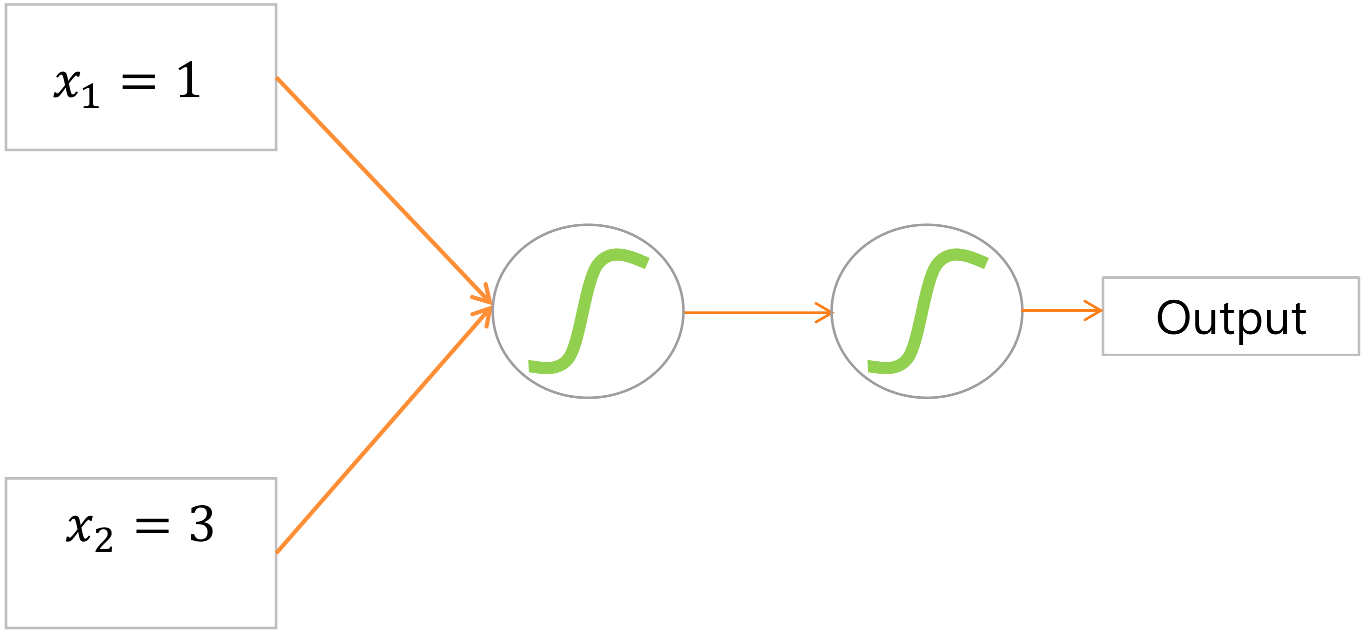
$$s_j^{(L)} = \boldsymbol{w}^{(L)\prime}\boldsymbol{x}^{(L-1)}$$

Output

$$h\left(s_1^{(L)}\right) = \frac{1}{1 + \exp(-s_1^{(L)})} = x_1^{(L)}$$

Estimating a Neural Network

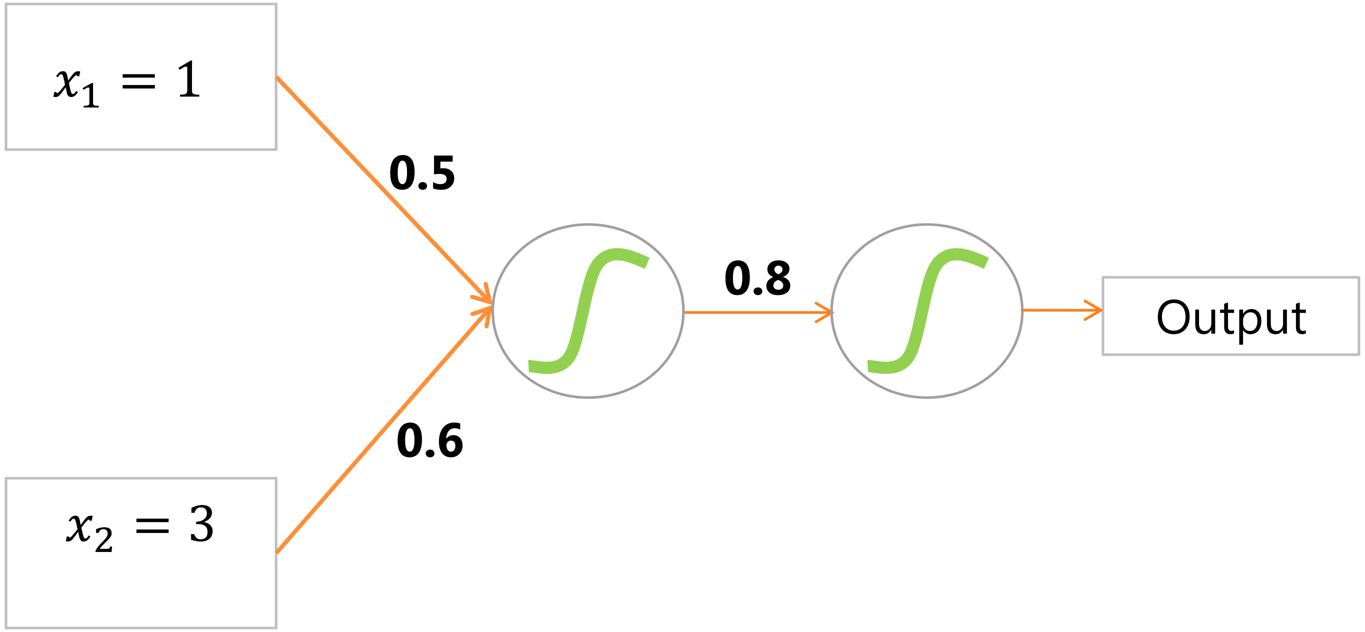
How a neural network uses data to produce an output



Architecture of a neural network

Estimating a Neural Network

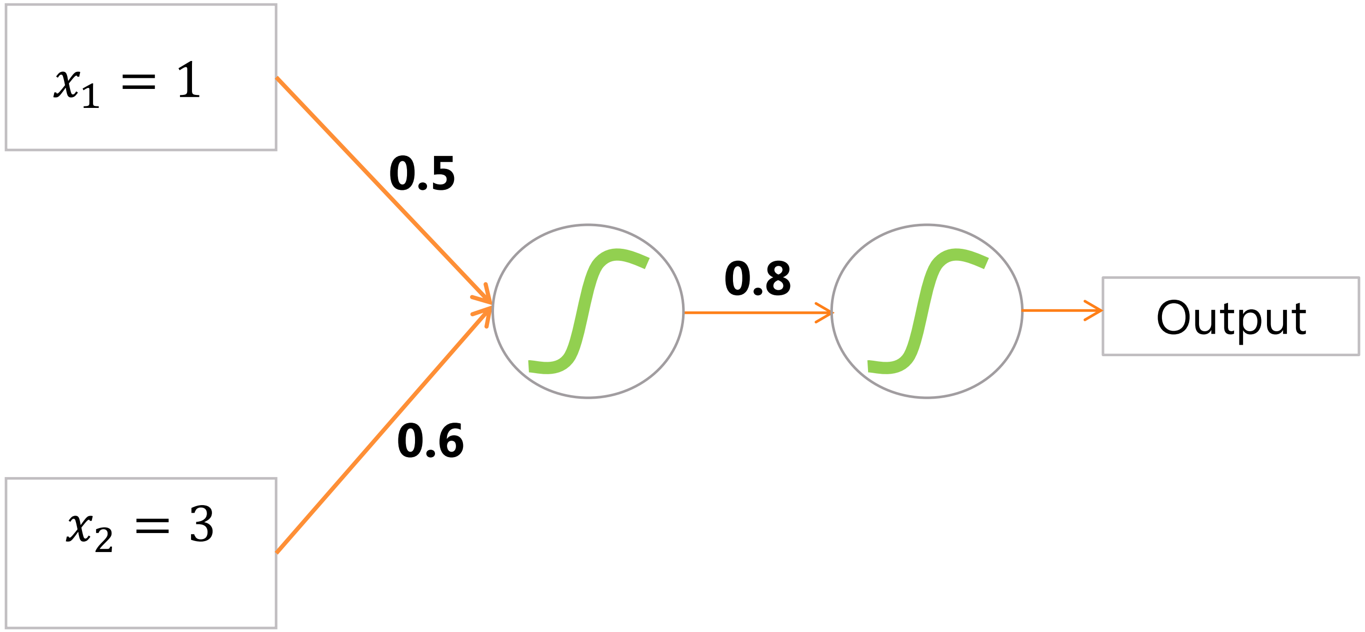
How a neural network uses data to produce an output



Architecture of a neural network

Estimating a Neural Network

Compute the output produced by this neural network for data point 1, 3



Architecture of a neural network

Estimating a Neural Network

- Step 1: Compute the signal from going into the node in the hidden layer
  - ❑ $s_1^{(1)} = 0.5 * 1 + 0.6 * 3 = 2.3$
- Step 2: Compute the output coming out of the node in the hidden layer
  - ❑ $x_1^{(1)} = h\left(s_1^{(1)}\right) = h(2.3) = 1\frac{1}{1+\exp(-2.3)} = 0.91$
- Step 3: Compute the signal going into the node in the output layer
  - ❑ $s_1^{(2)} = 0.8 * x_1^{(1)} = 0.8 * 0.91 = 0.728$
- Step 4: Compute the output coming out of the node in the output layer
  - ❑ $x_1^{(2)} = h\left(s_1^{(2)}\right) = h(0.728) = \mathbf{0.67}$

Estimating a Neural Network: Cost Function

- Vector of weights : $\boldsymbol{w} = [w_{ij}^{(l)}]$
- Output for observation i: $\mathrm{h}(\boldsymbol{w})_{\boldsymbol{i}}$
- Cost for observation $\boldsymbol{i}$: $C_i(\boldsymbol{w}) = y_i \log(\mathrm{h}(\boldsymbol{w})_{\boldsymbol{i}}) + (1 - y_i) \log(1 - \mathrm{h}(\boldsymbol{w})_{\boldsymbol{i}})$
- Cost over the entire data: $C(\boldsymbol{w}) = \sum_{i=1}^{n} C_i(\boldsymbol{w})$

- To minimize a multivariate function use a gradient/stochastic gradient descent
- Both methods require partial derivatives of the cost function, with respect to the weights
- Gradient of the Cost Function: $\nabla C = [\frac{\partial C}{\partial w_{ij}^{(l)}}]$

- The gradient of the cost function in a neural network is a long vector
- Modern day neural networks have millions of weights
- Computing each partial derivative is not feasible

Recap

- Estimating a neural network
- Estimating a neural network: Cost function

JIGSAW ACADEMY

THE ONLINE SCHOOL OF ANALYTICS