

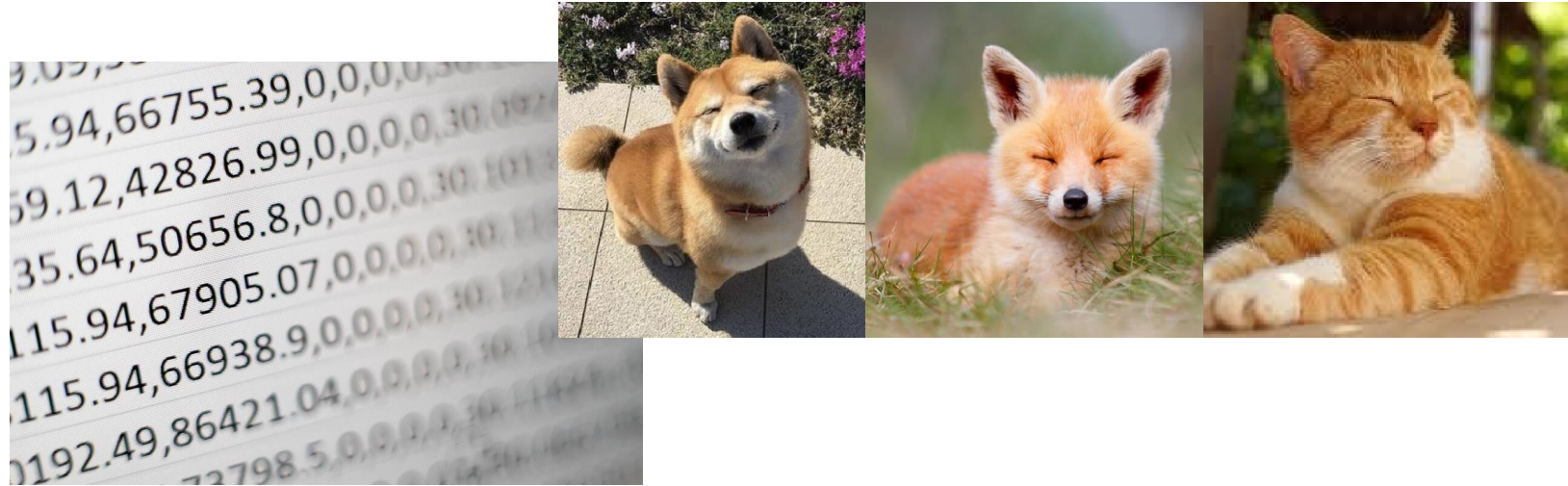
Introduction to Deep Learning

Recitation 1a
Spring 2021

Intelligence

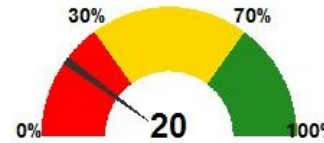
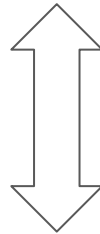
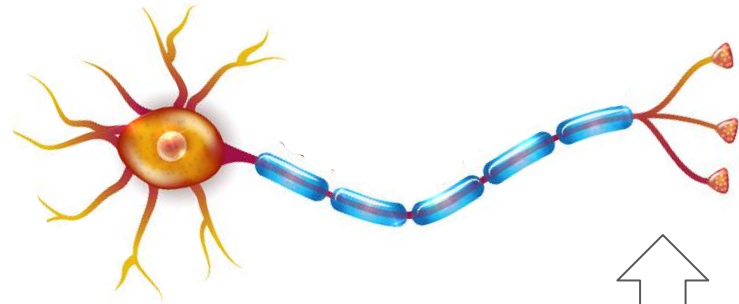
Tasks that are easy for humans can be difficult for machines.

Likewise, tasks that are easy for machines can be difficult for humans.



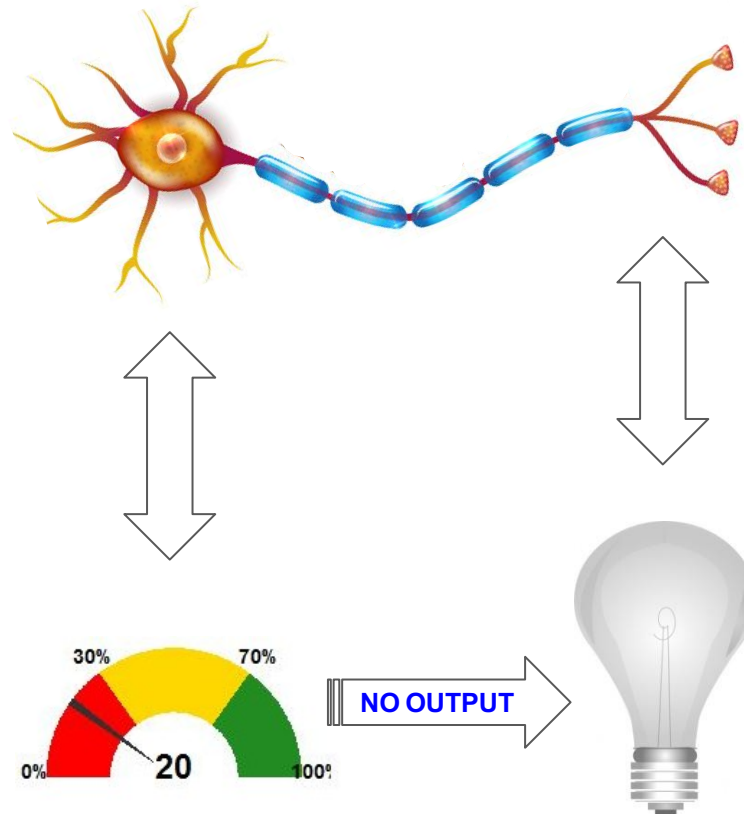
Neurons and Thresholds

Biological brains, made of connected neurons, are the inspiration for artificial neural networks.



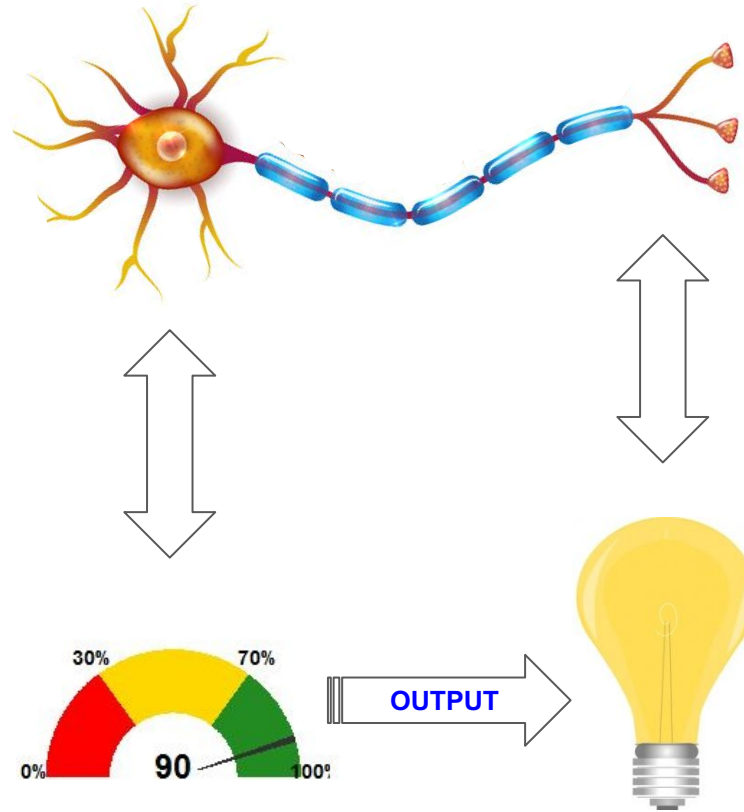
Neurons and Thresholds

Biological brains, made of connected neurons, are the inspiration for artificial neural networks.



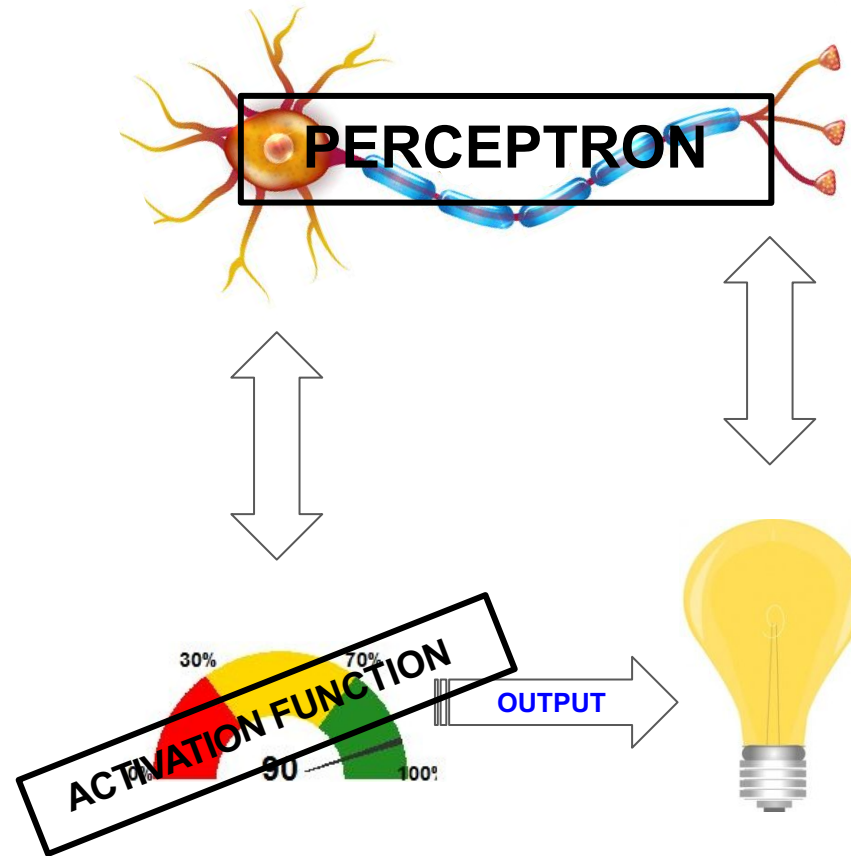
Neurons and Thresholds

Biological brains, made of connected neurons, are the inspiration for artificial neural networks.



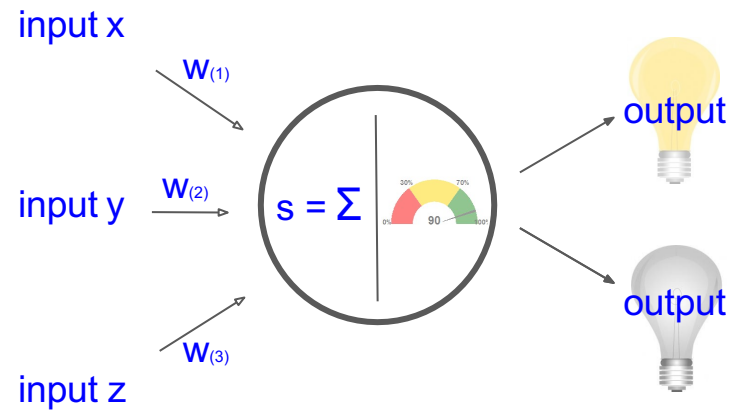
Neurons and Thresholds

Biological brains, made of connected neurons, are the inspiration for artificial neural networks.



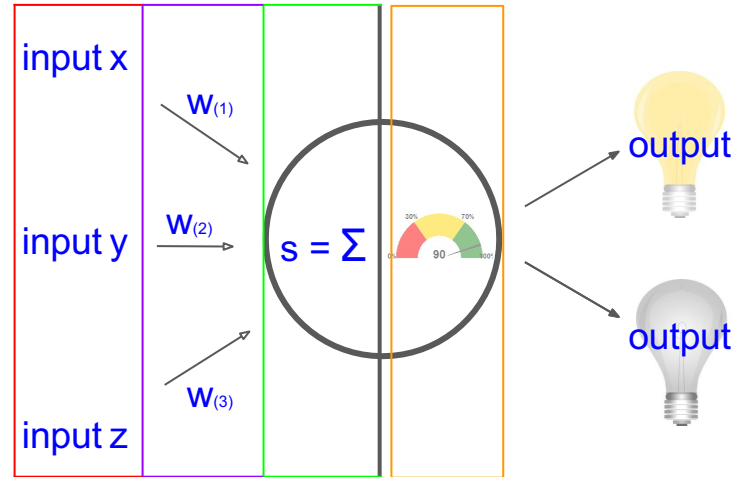
Perceptrons

- ... are single layer neural networks



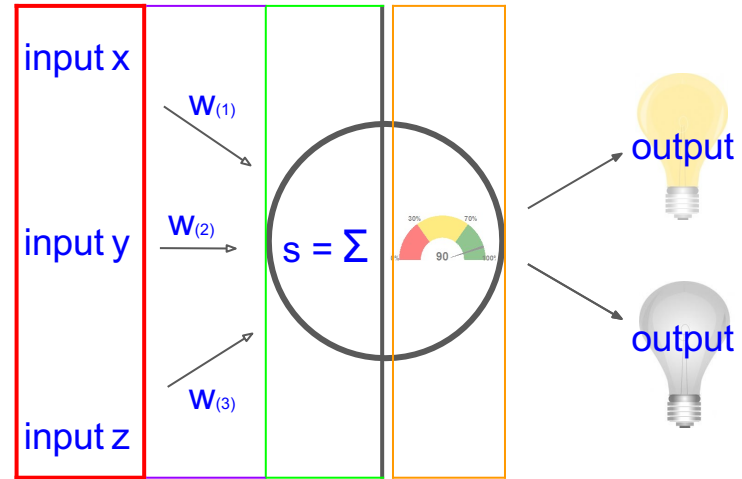
Perceptrons

- ... are single layer neural networks
- ... have four parts



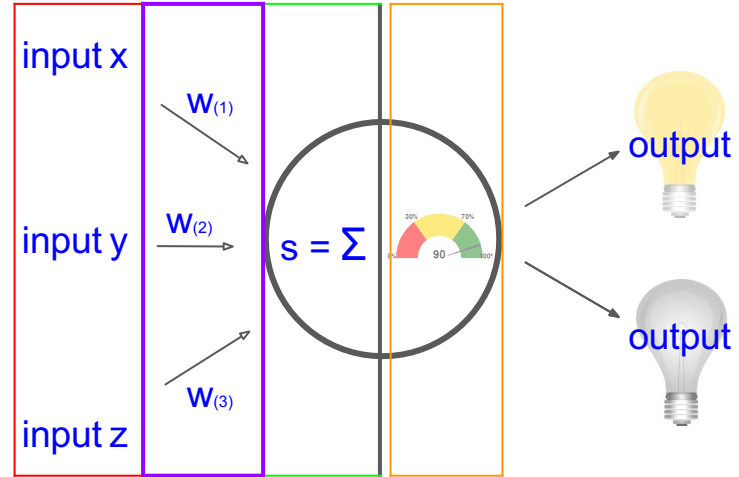
Perceptrons

- ... are single layer neural networks
- ... have four parts
 - Inputs



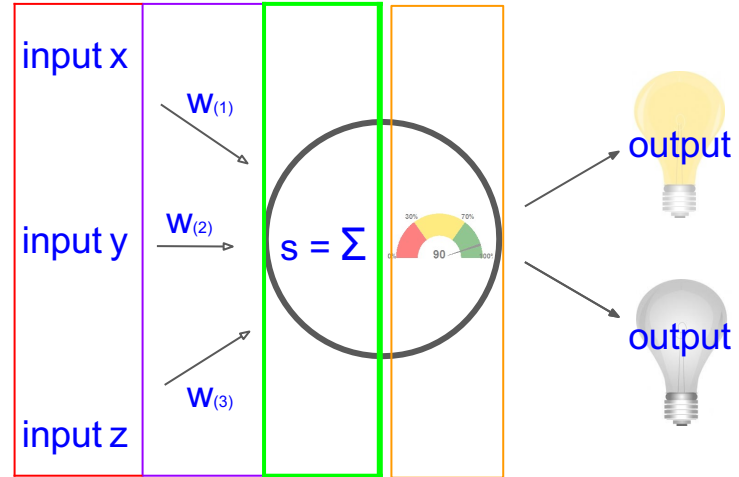
Perceptrons

- ... are single layer neural networks
- ... have four parts
 - Inputs
 - Weights



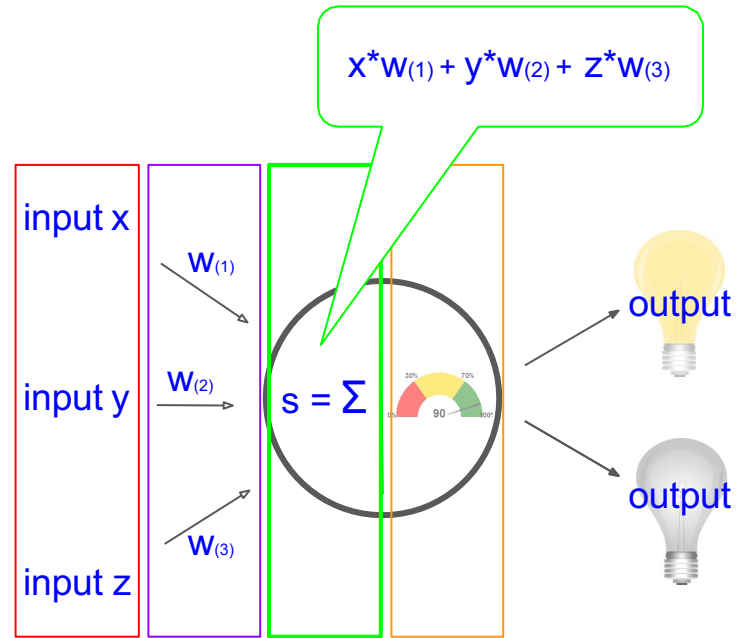
Perceptrons

- ... are single layer neural networks
- ... have four parts
 - Inputs
 - Weights
 - Weighted Sum



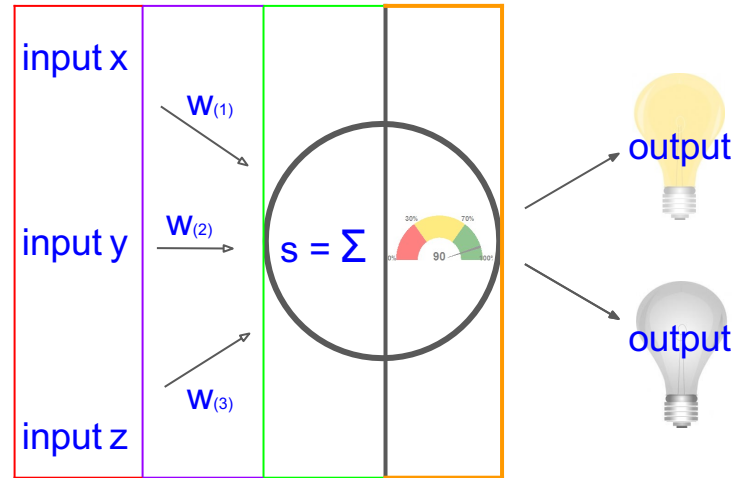
Perceptrons

- ... are single layer neural networks
- ... have four parts
 - Inputs
 - Weights
 - Weighted Sum



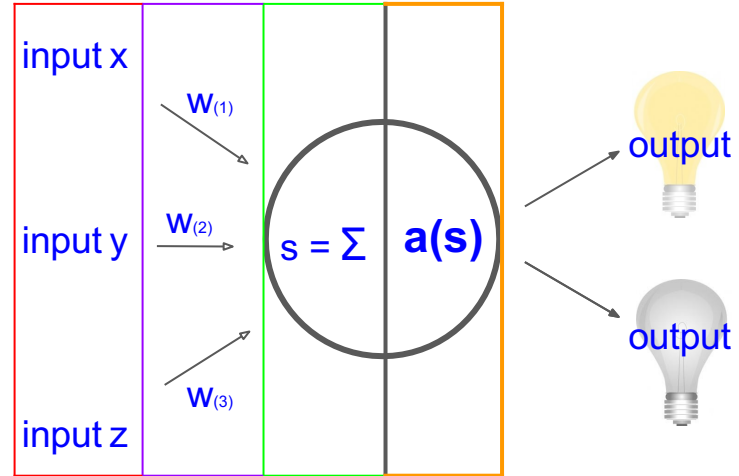
Perceptrons

- ... are single layer neural networks
- ... have four parts
 - Inputs
 - Weights
 - Weighted Sum
 - Threshold



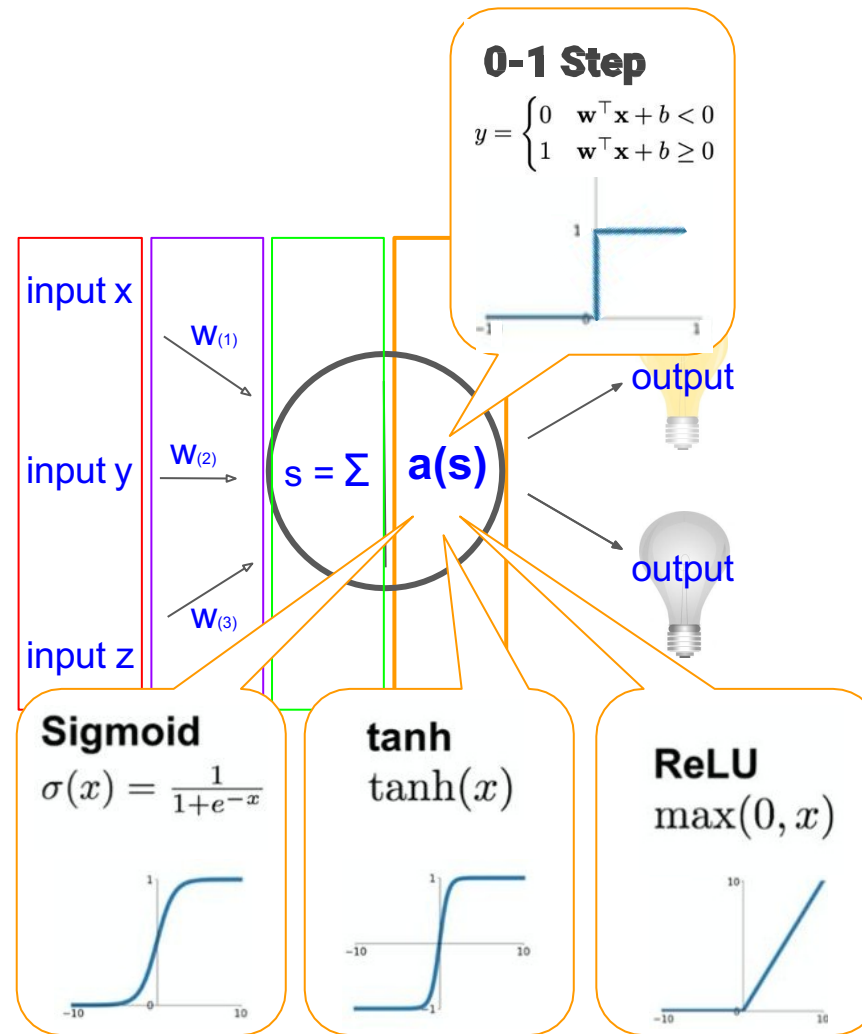
Perceptrons

- ... are single layer neural networks
- ... have four parts
 - Inputs
 - Weights
 - Weighted Sum
 - Threshold Activation Functions



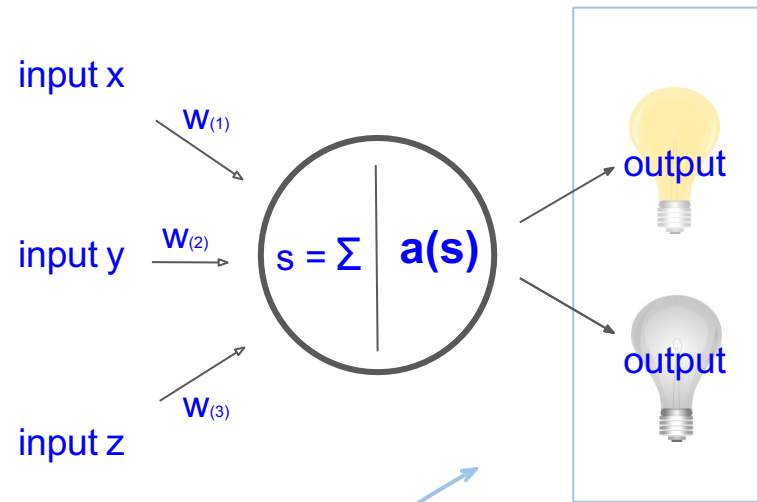
Perceptrons

- ... are single layer neural networks
- ... have four parts
 - Inputs
 - Weights
 - Weighted Sum
 - Threshold Activation Functions

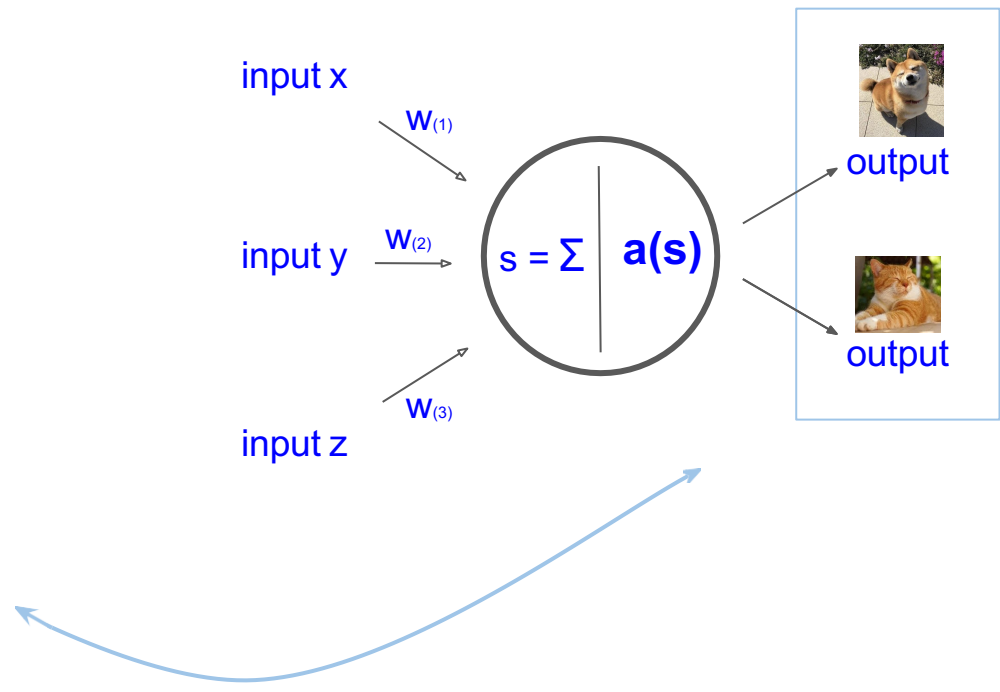
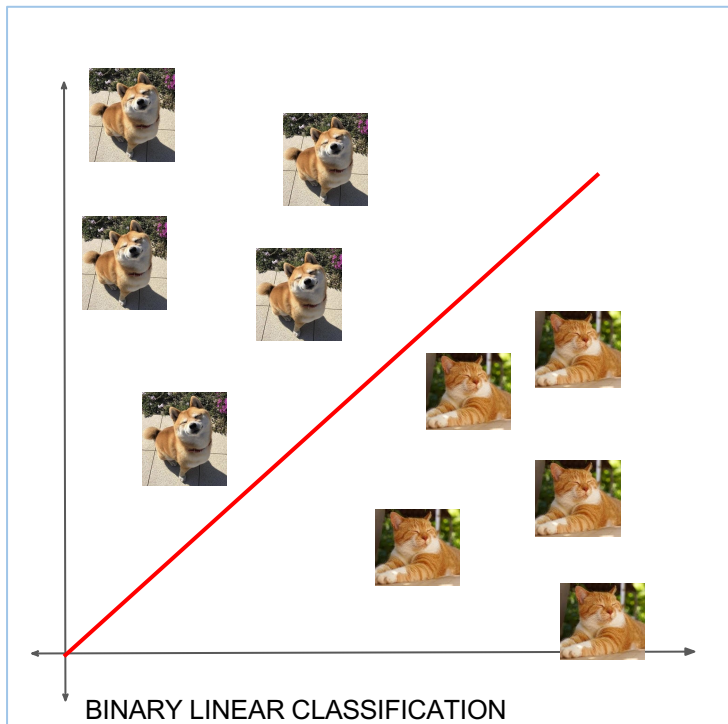


Perceptrons

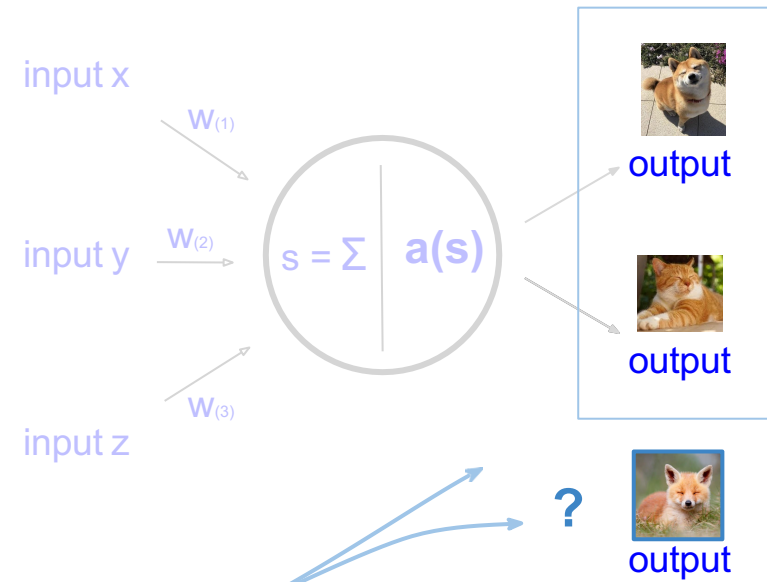
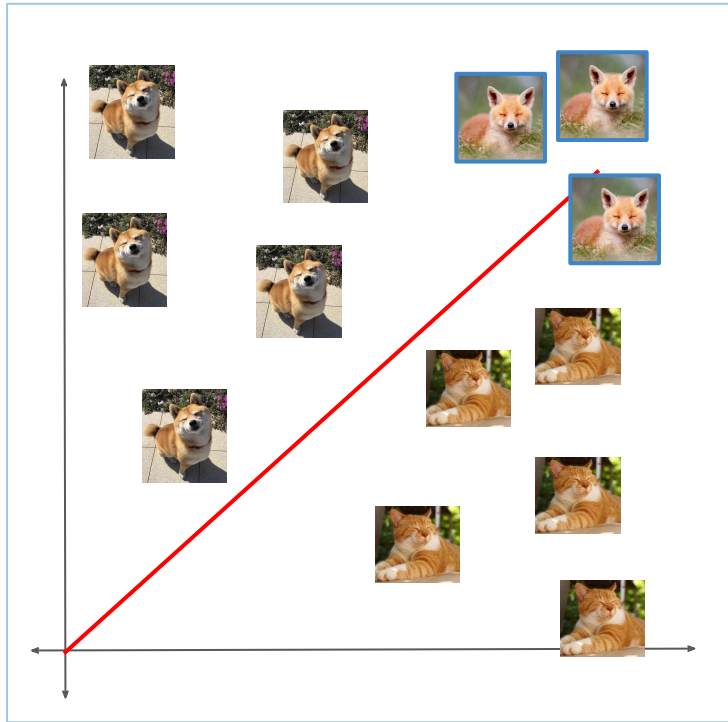
- ... are single layer neural networks
- ... have four parts
 - Inputs
 - Weights
 - Weighted Sum
 - ~~Threshold~~ Activation Functions
- ... are binary linear classifiers



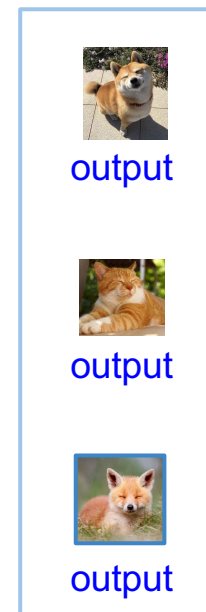
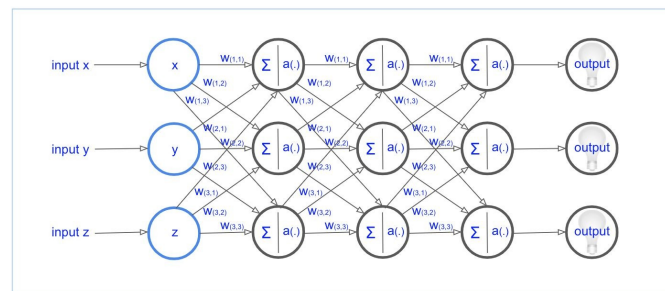
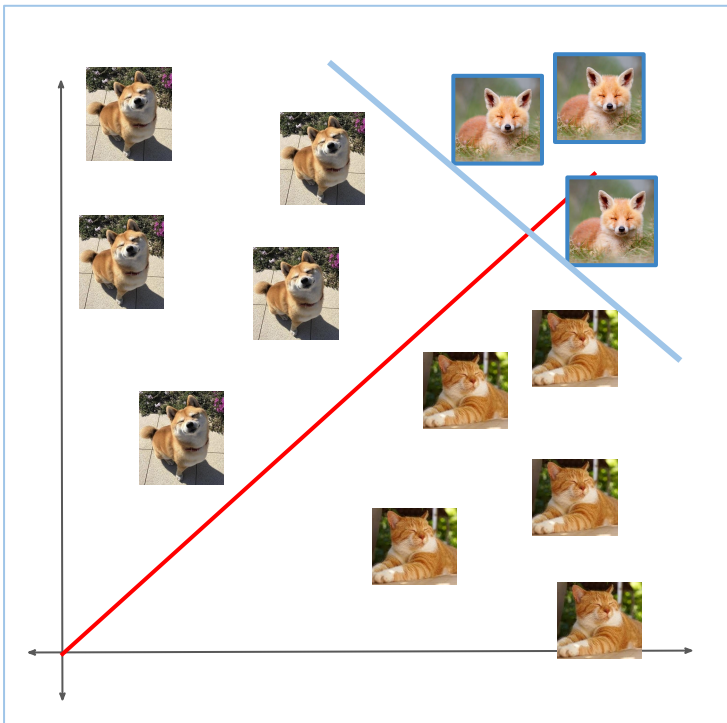
Perceptrons



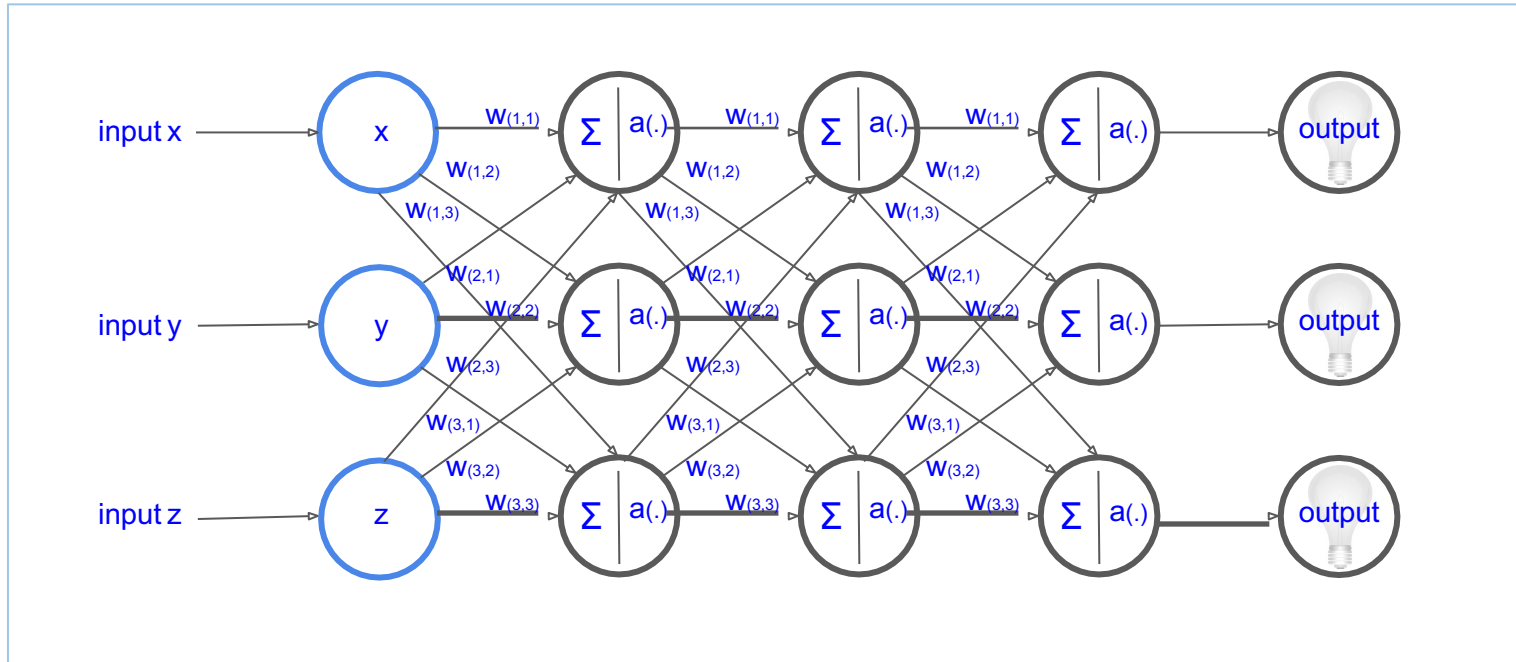
Multi-Layer Perceptrons



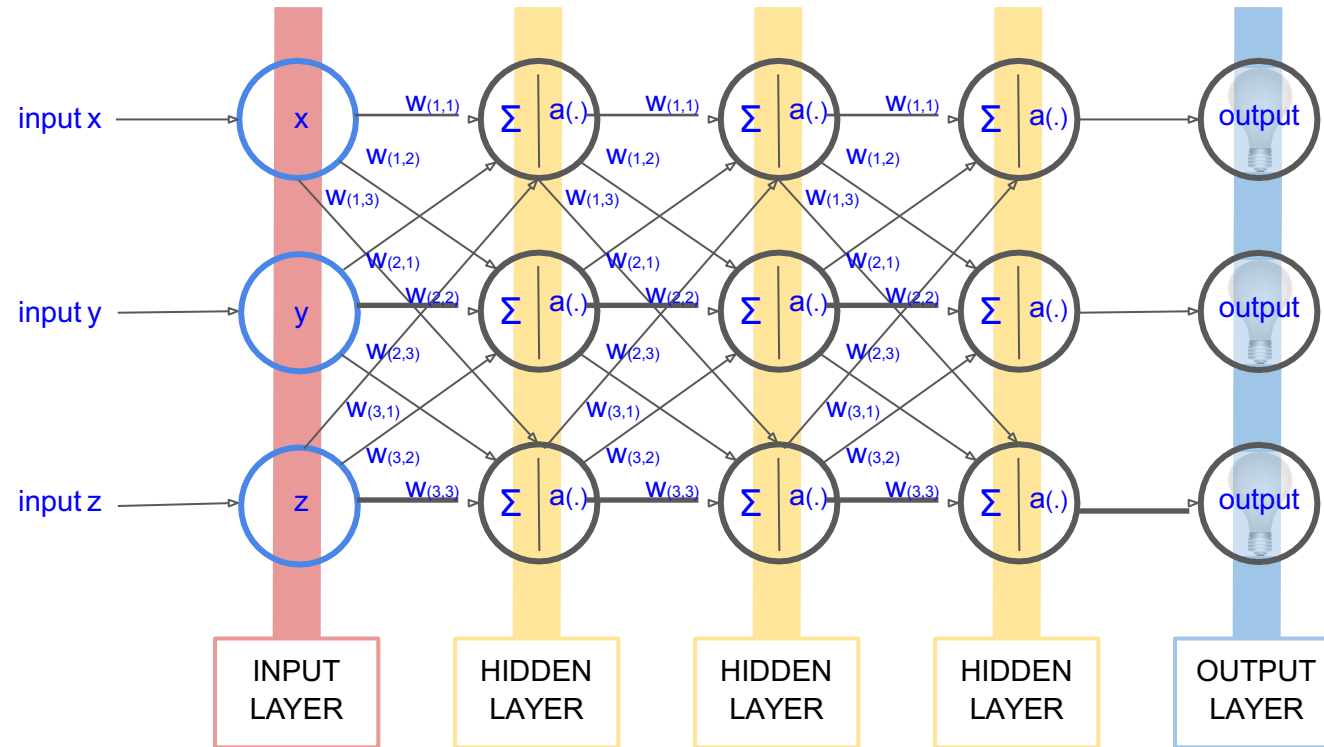
Multi-Layer Perceptrons



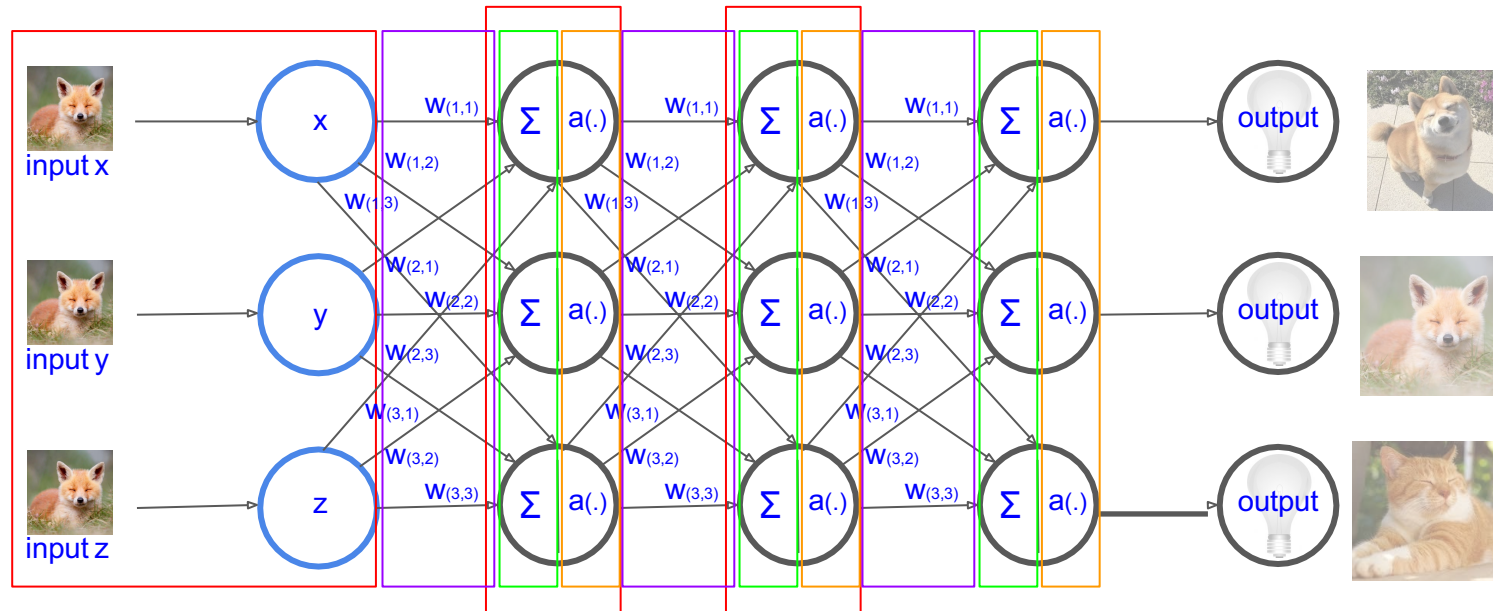
Multi-Layer Perceptrons



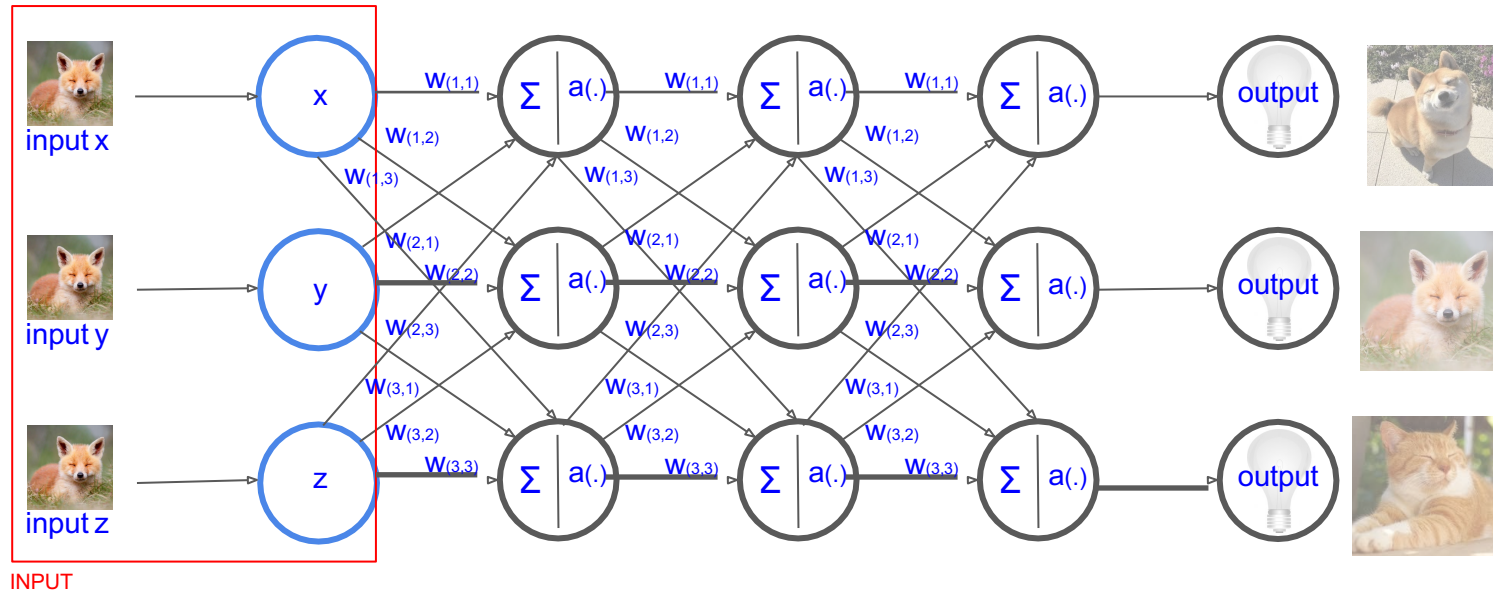
Multi-Layer Perceptrons



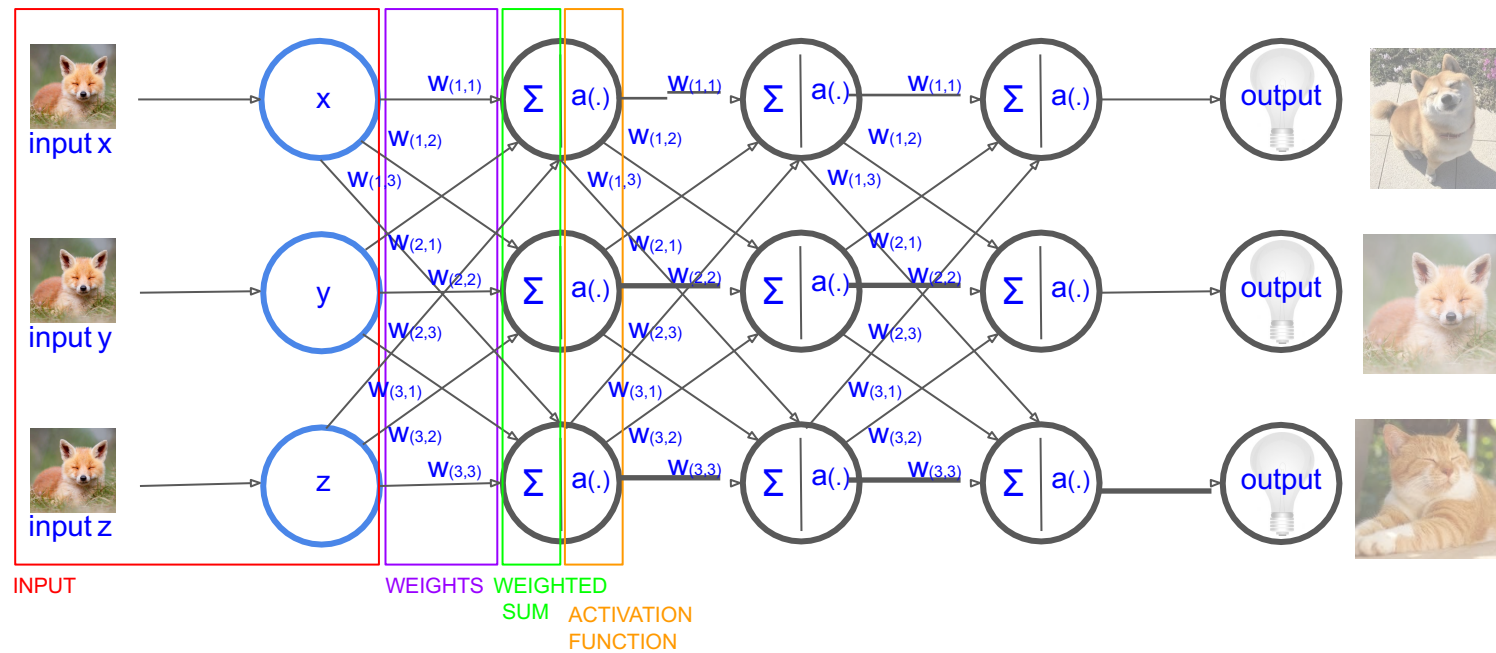
Forward Signal



Forward Signal

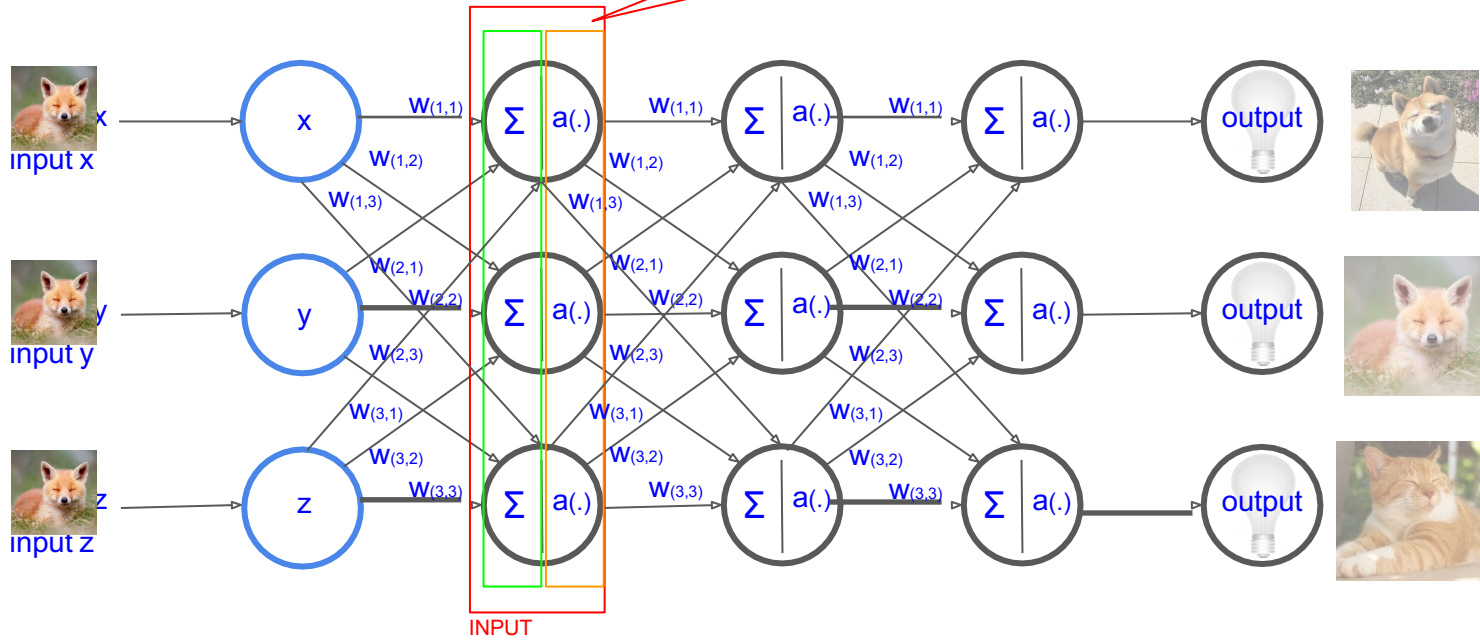


Forward Signal

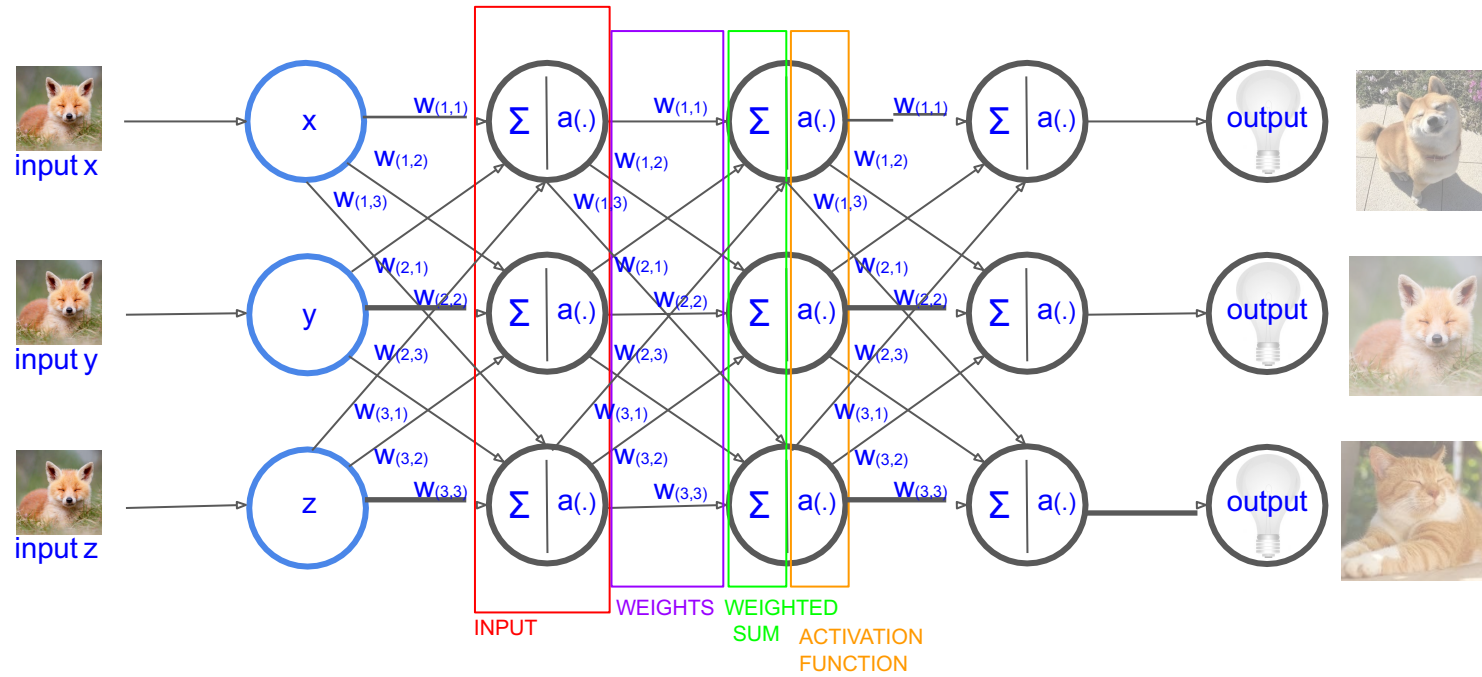


Forward Signal

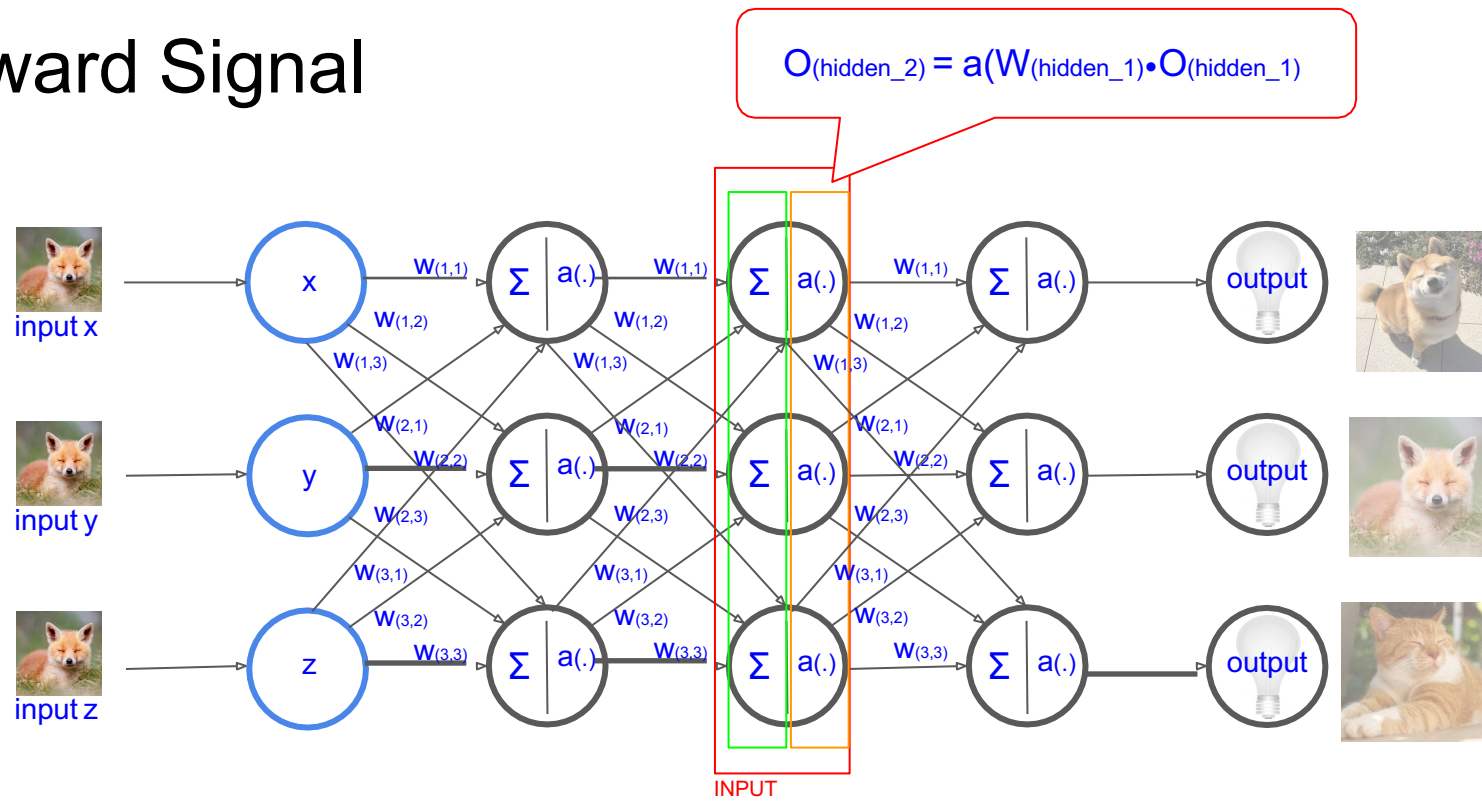
$$O_{(\text{hidden}_1)} = a(W_{(\text{input})} \cdot I)$$



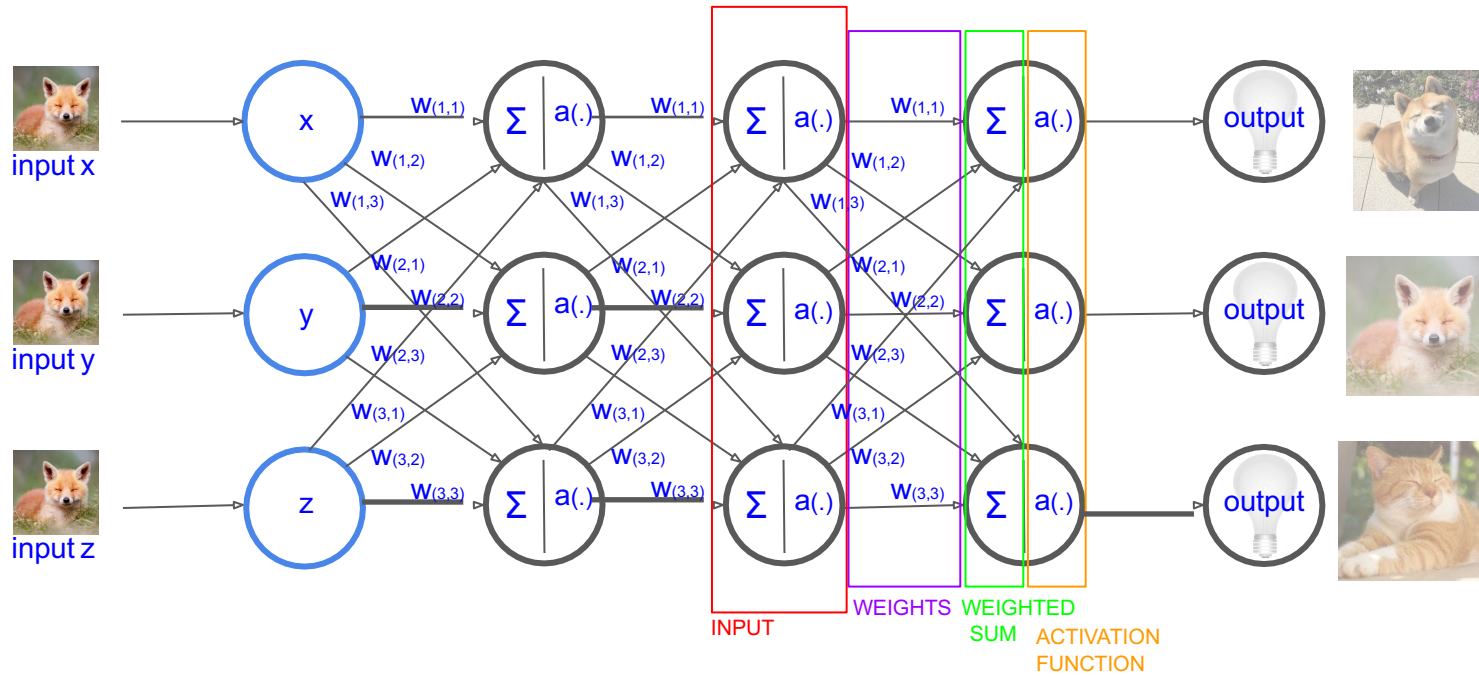
Forward Signal



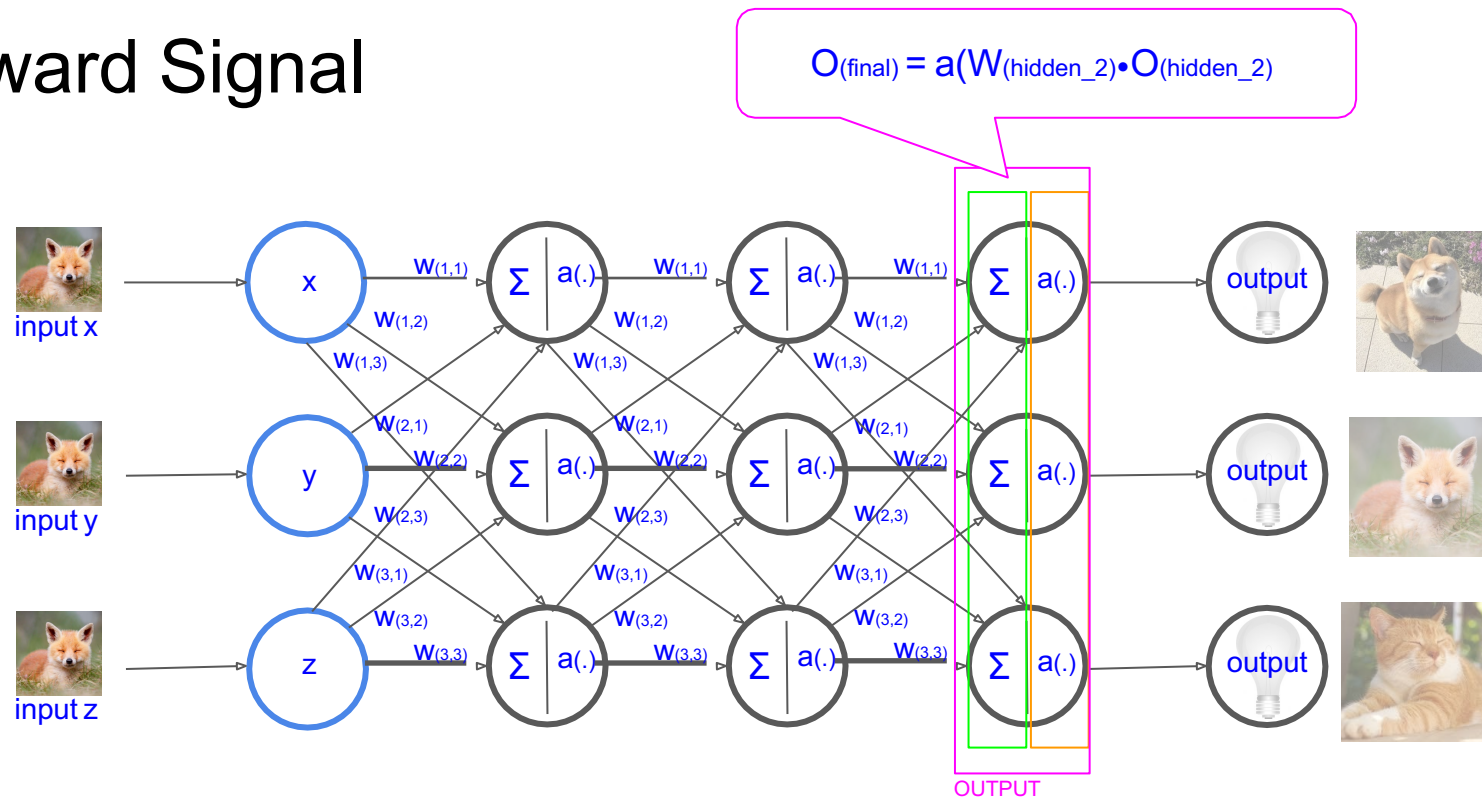
Forward Signal



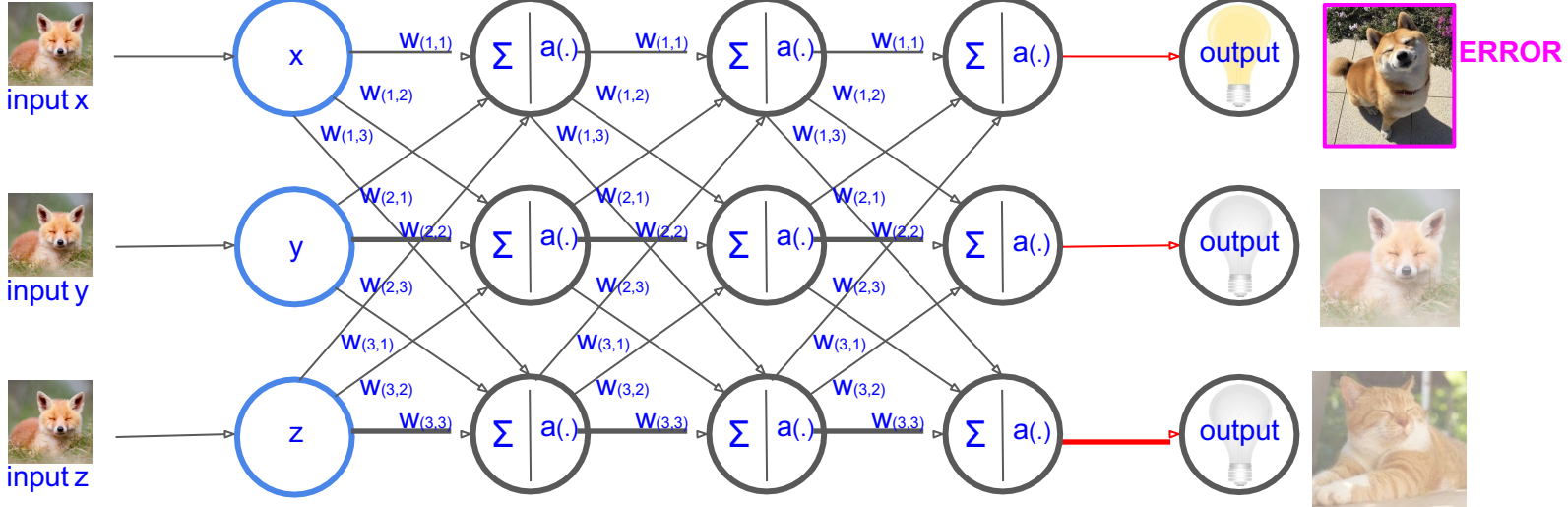
Forward Signal



Forward Signal



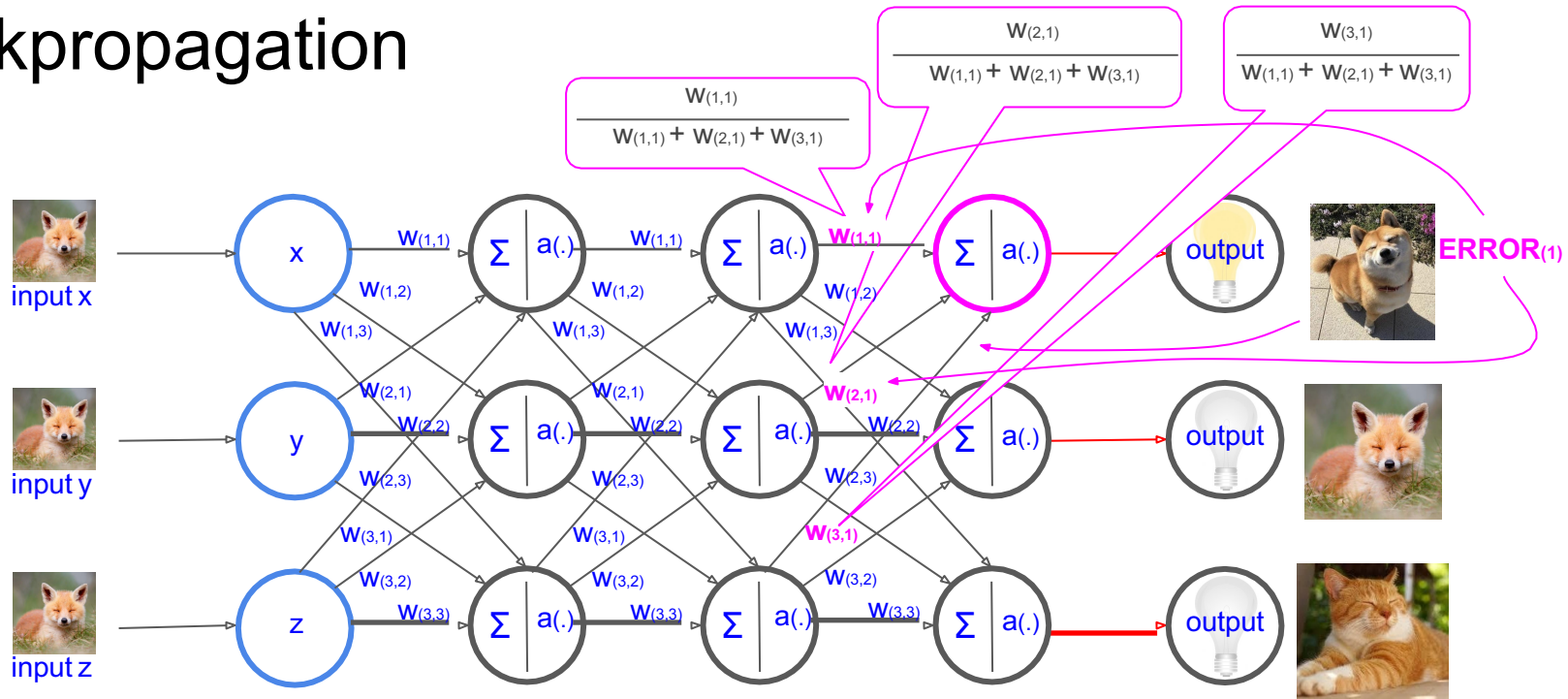
Error



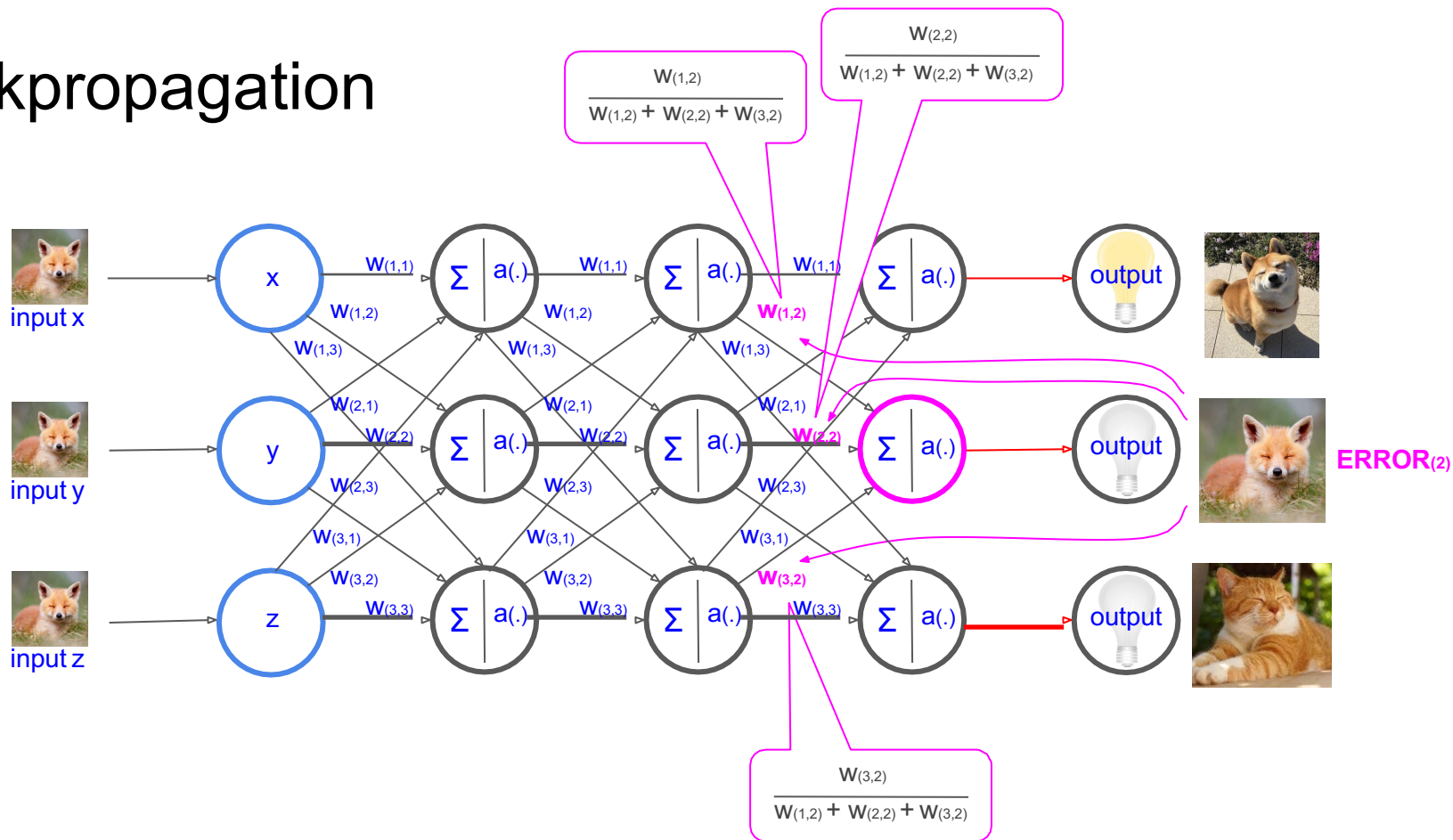
Error



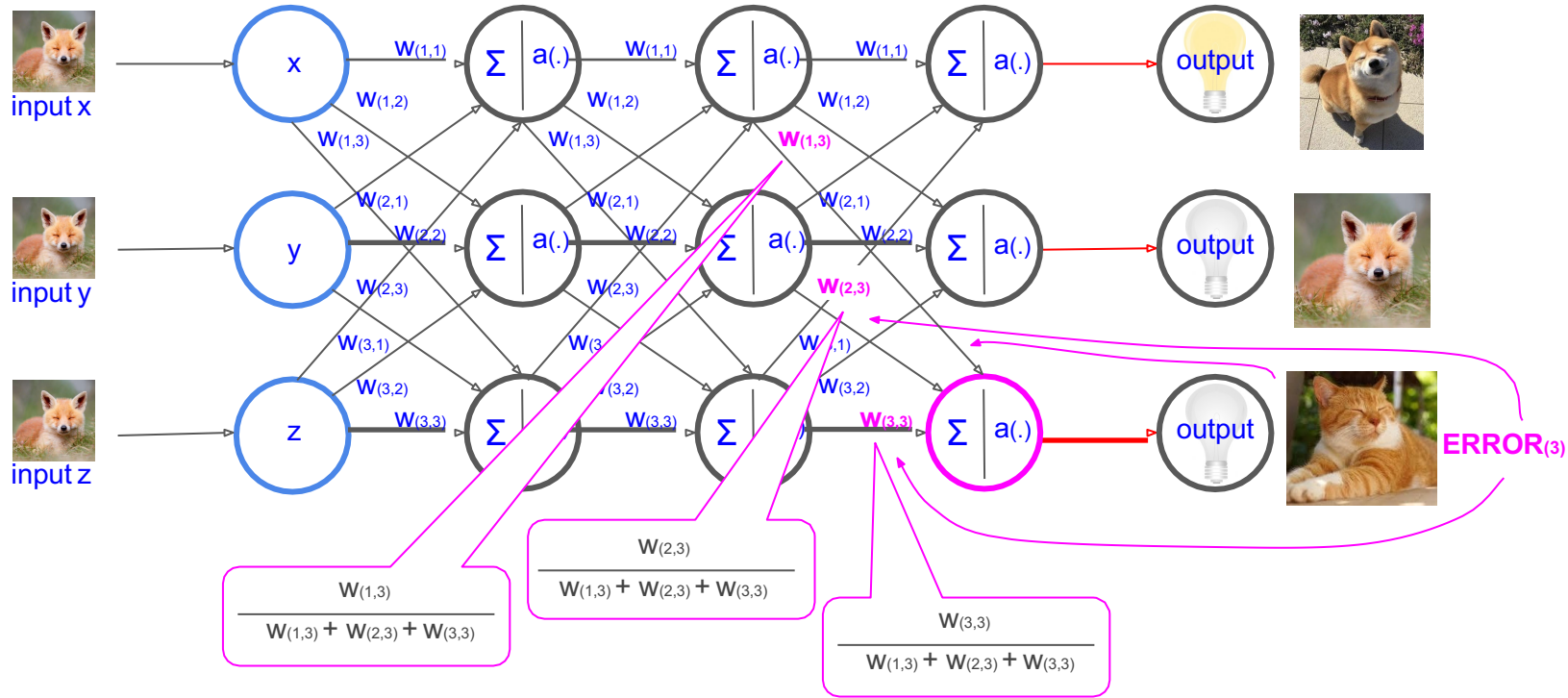
Backpropagation



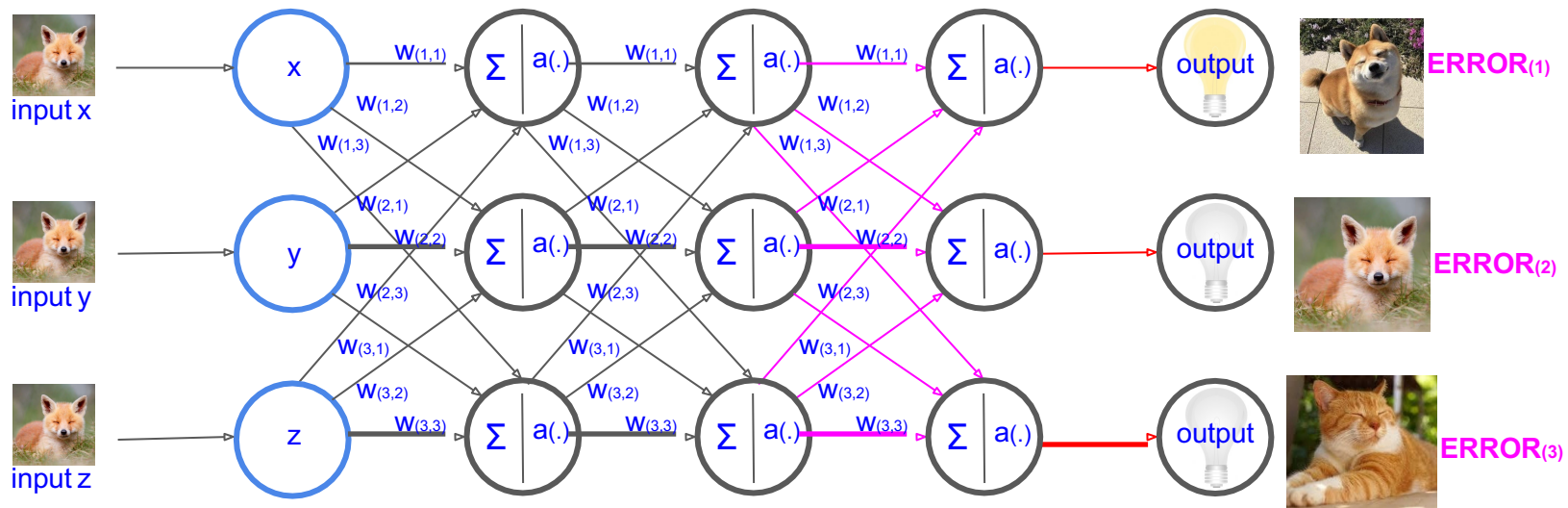
Backpropagation



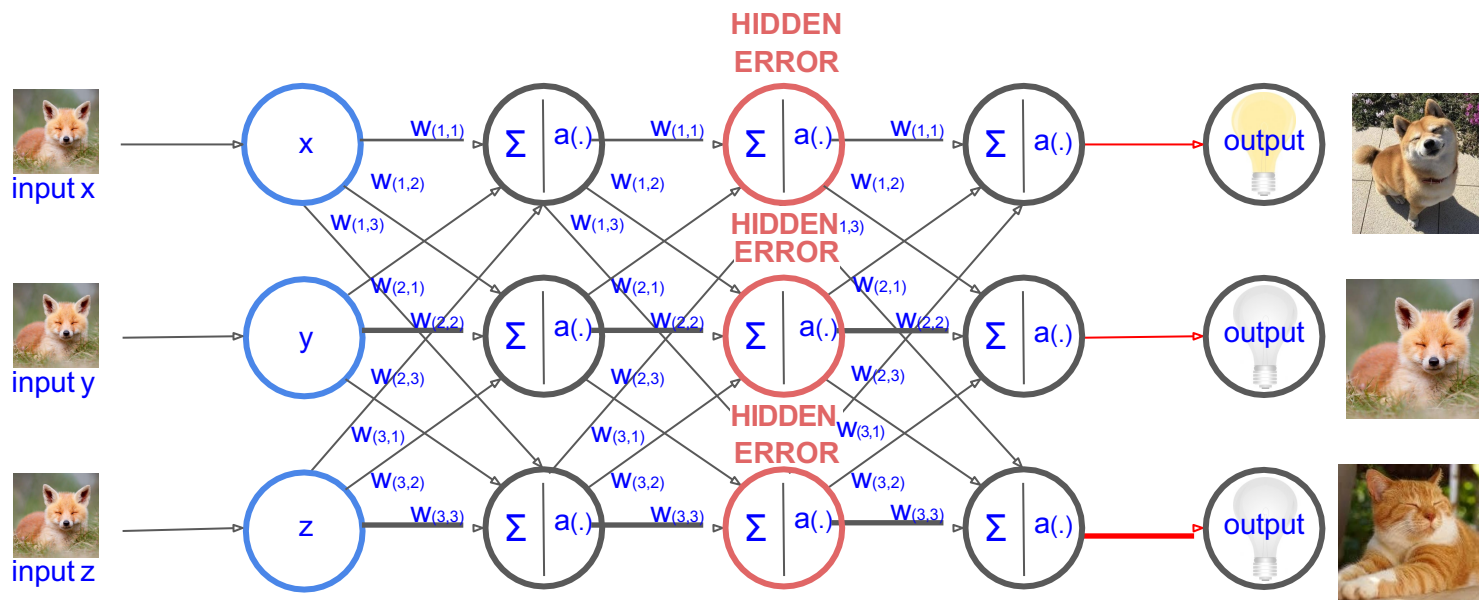
Backpropagation



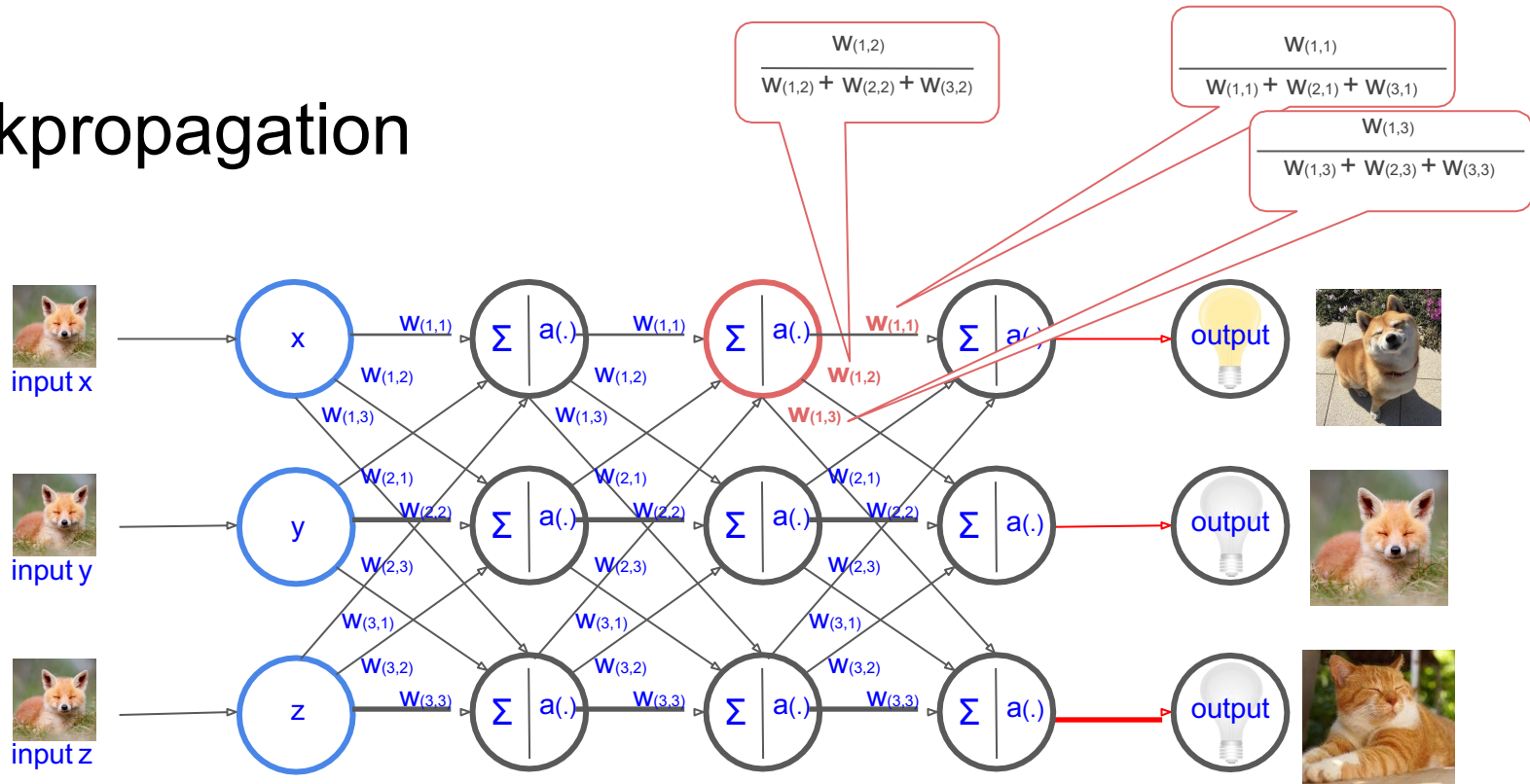
Backpropagation



Backpropagation

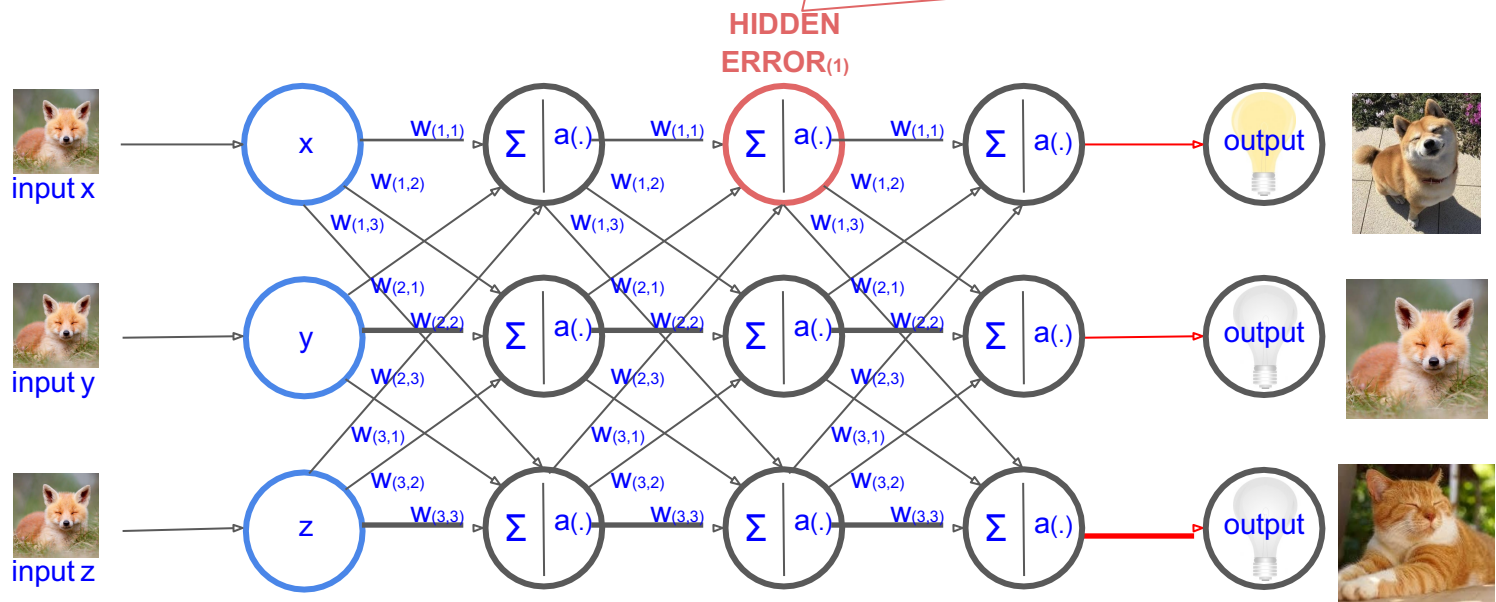


Backpropagation

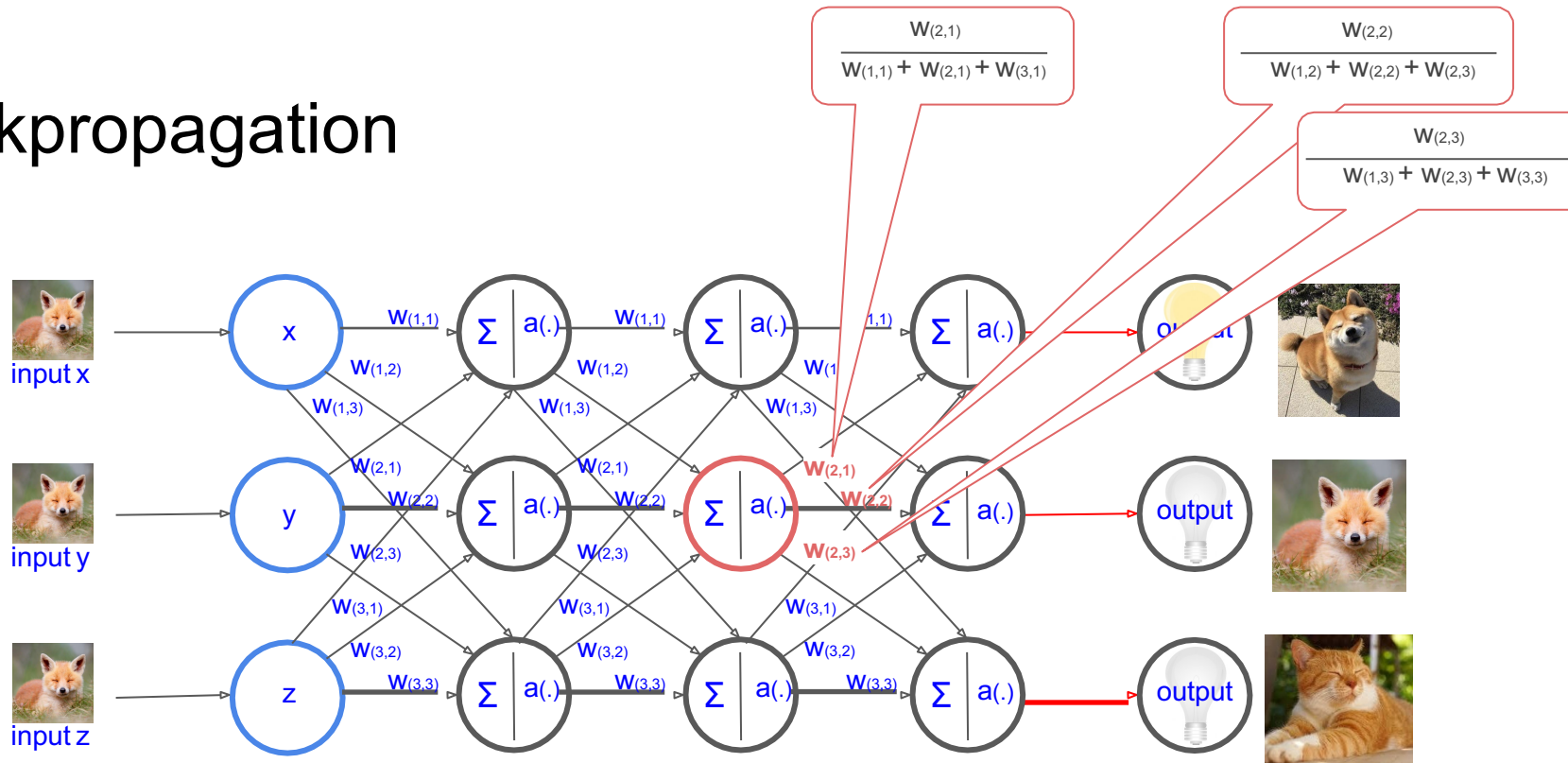


Backpropagation

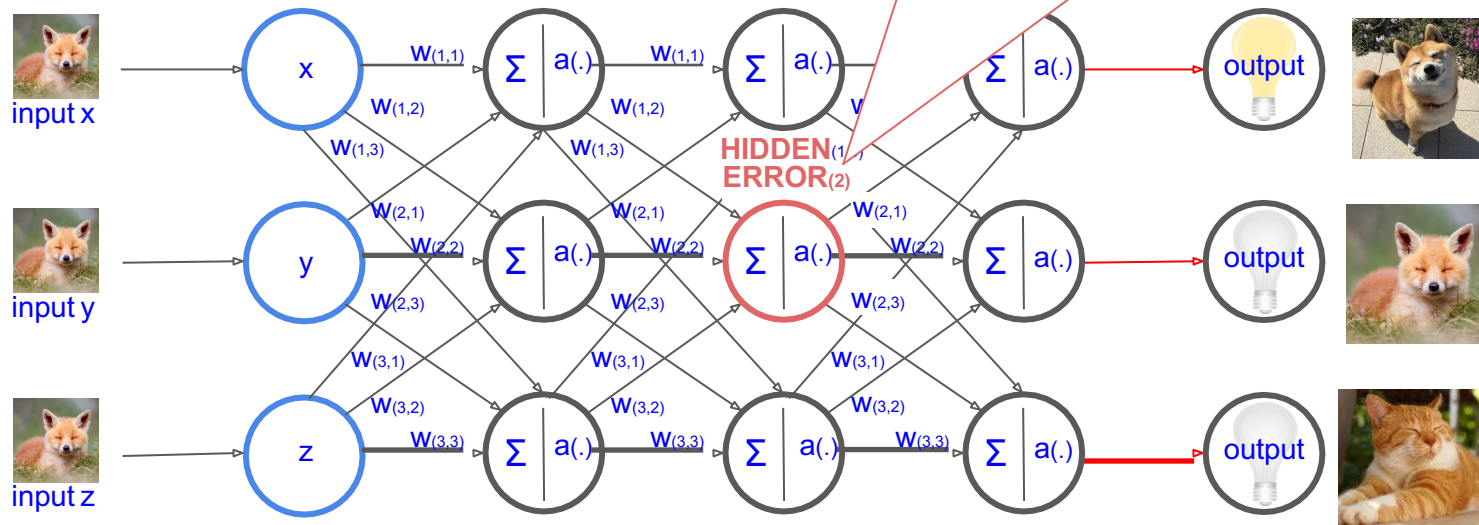
$$\text{error}(2) * \frac{W(1,2)}{W(1,2) + W(2,2) + W(3,2)} + \text{error}(1) * \frac{W(1,1)}{W(1,1) + W(2,1) + W(3,1)} + \text{error}(3) * \frac{W(1,3)}{W(1,3) + W(2,3) + W(3,3)}$$



Backpropagation

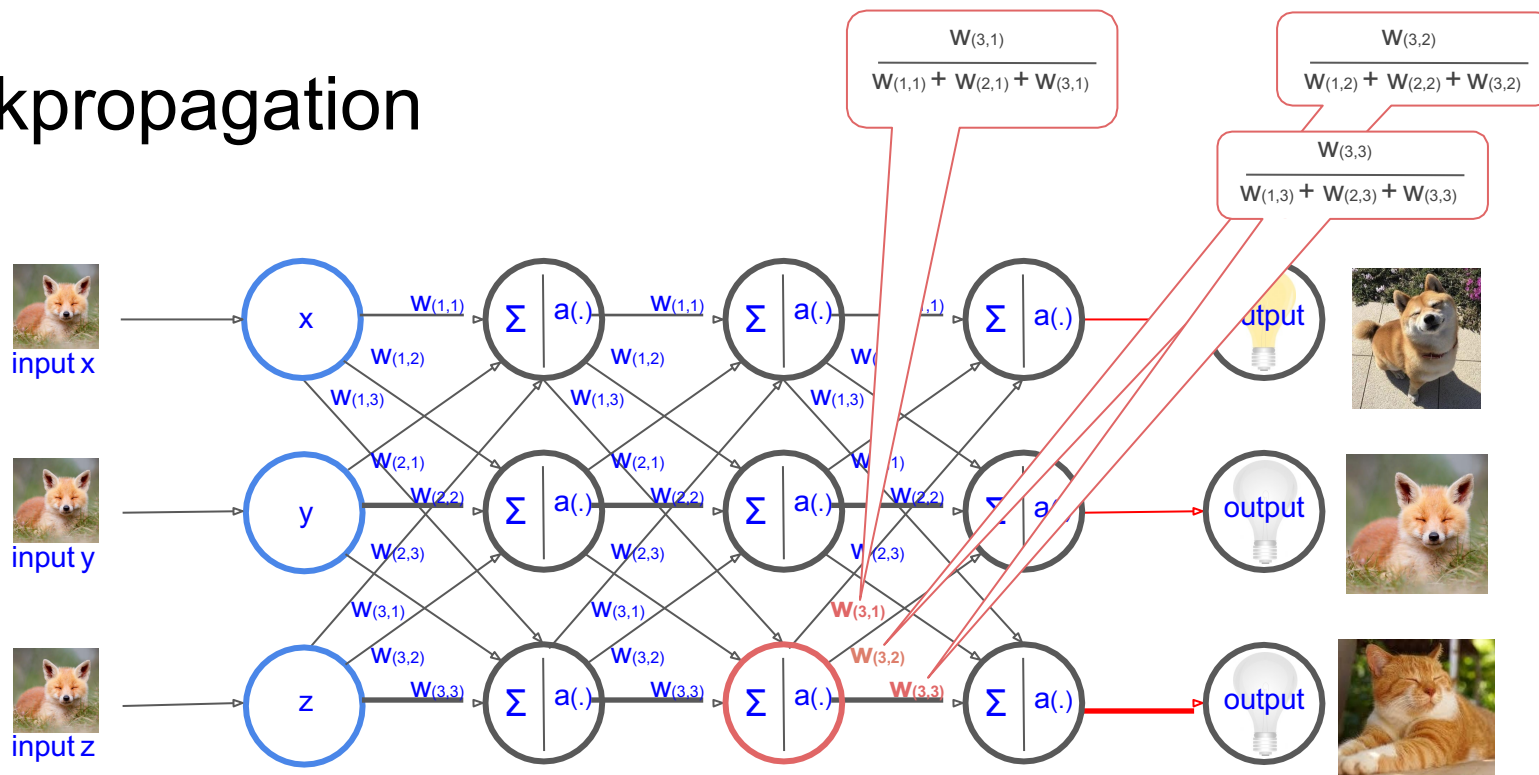


Backpropagation

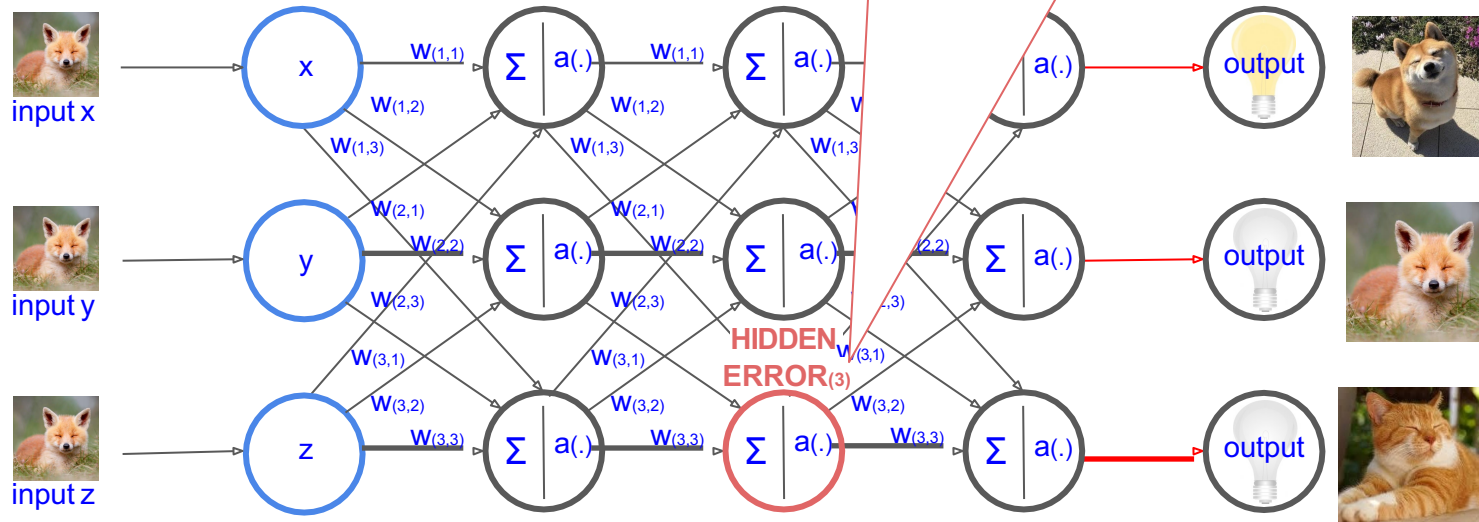


$$\text{error}(1) * \frac{W(2,1)}{W(1,1) + W(2,1) + W(3,1)} + \text{error}(2) * \frac{W(2,2)}{W(1,2) + W(2,2) + W(2,3)} + \text{error}(3) * \frac{W(2,3)}{W(1,3) + W(2,3) + W(3,3)}$$

Backpropagation



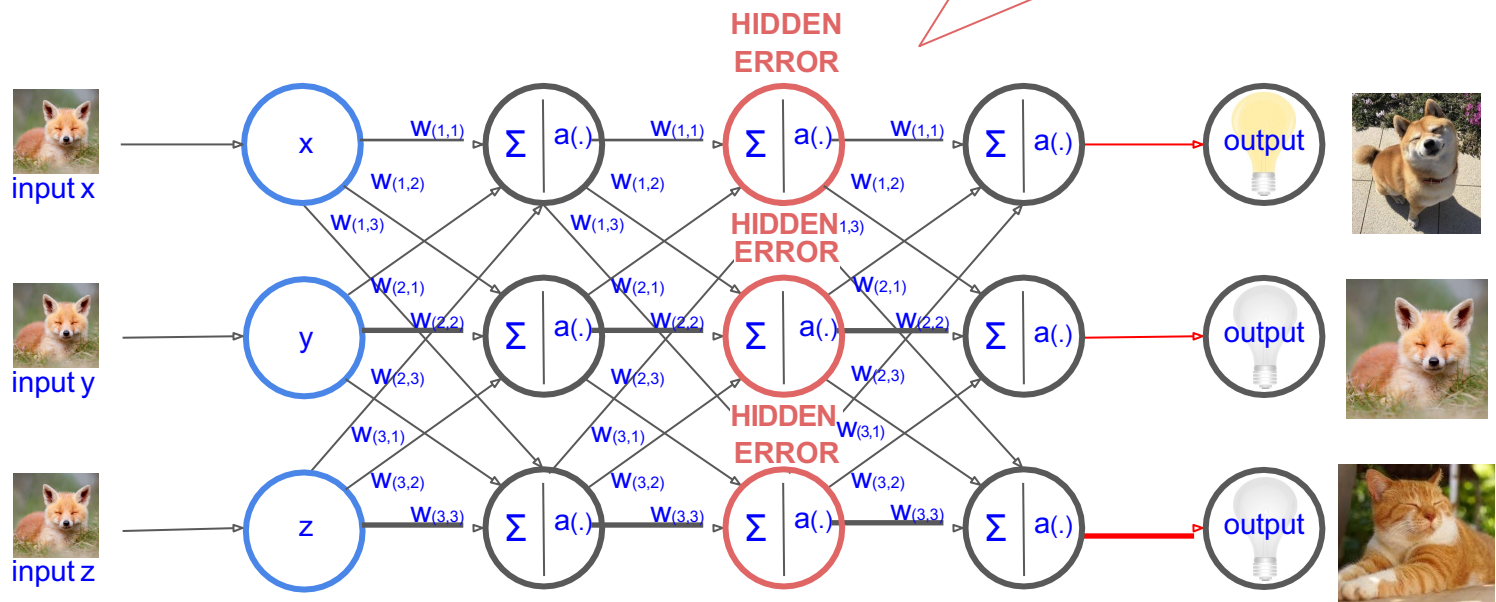
Backpropagation



$$\text{error}(1) * \frac{W(3,1)}{W(1,1) + W(2,1) + W(3,1)} + \text{error}(2) * \frac{W(3,2)}{W(1,2) + W(2,2) + W(3,2)} + \text{error}(3) * \frac{W(3,3)}{W(1,3) + W(2,3) + W(3,3)}$$

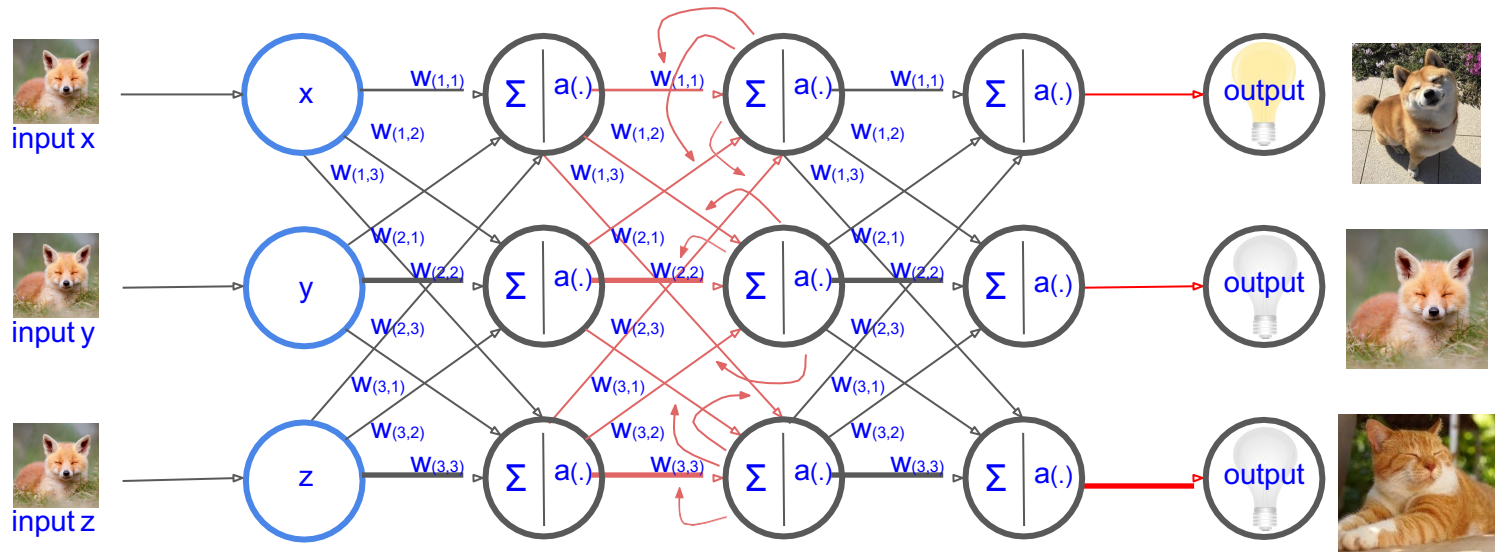
Backpropagation

$$E_{(\text{hidden})} = W^T_{(\text{hidden_output})} \cdot E_{(\text{output})}$$



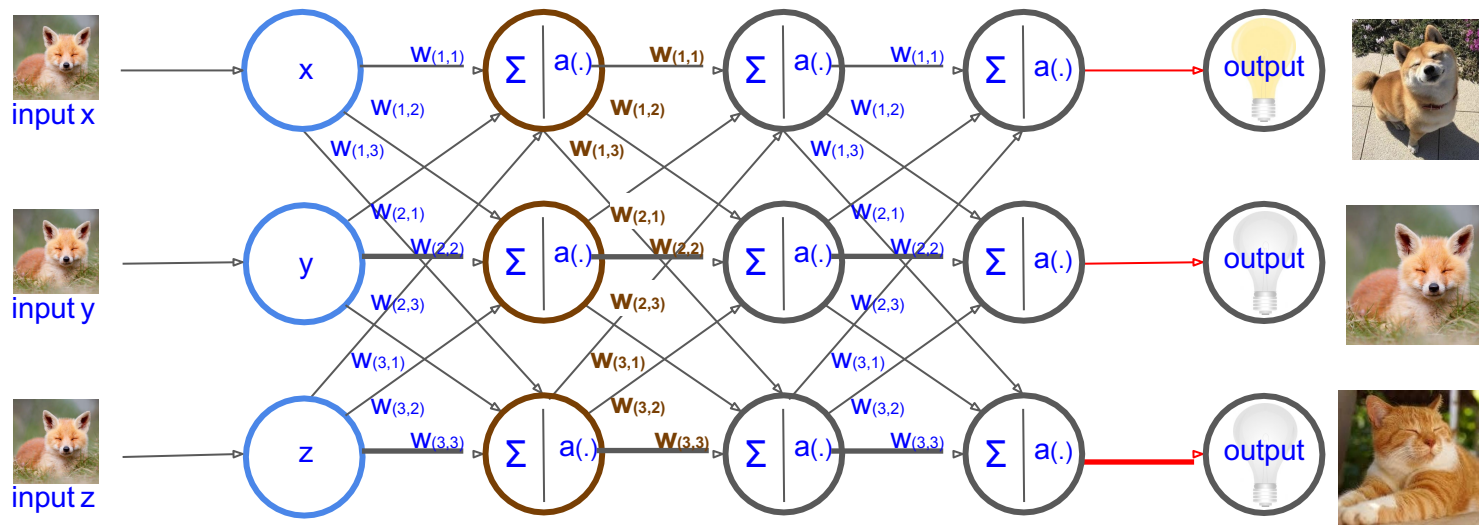
Backpropagation

$$E_{(\text{hidden})} = W^T_{(\text{hidden_output})} \cdot E_{(\text{output})}$$



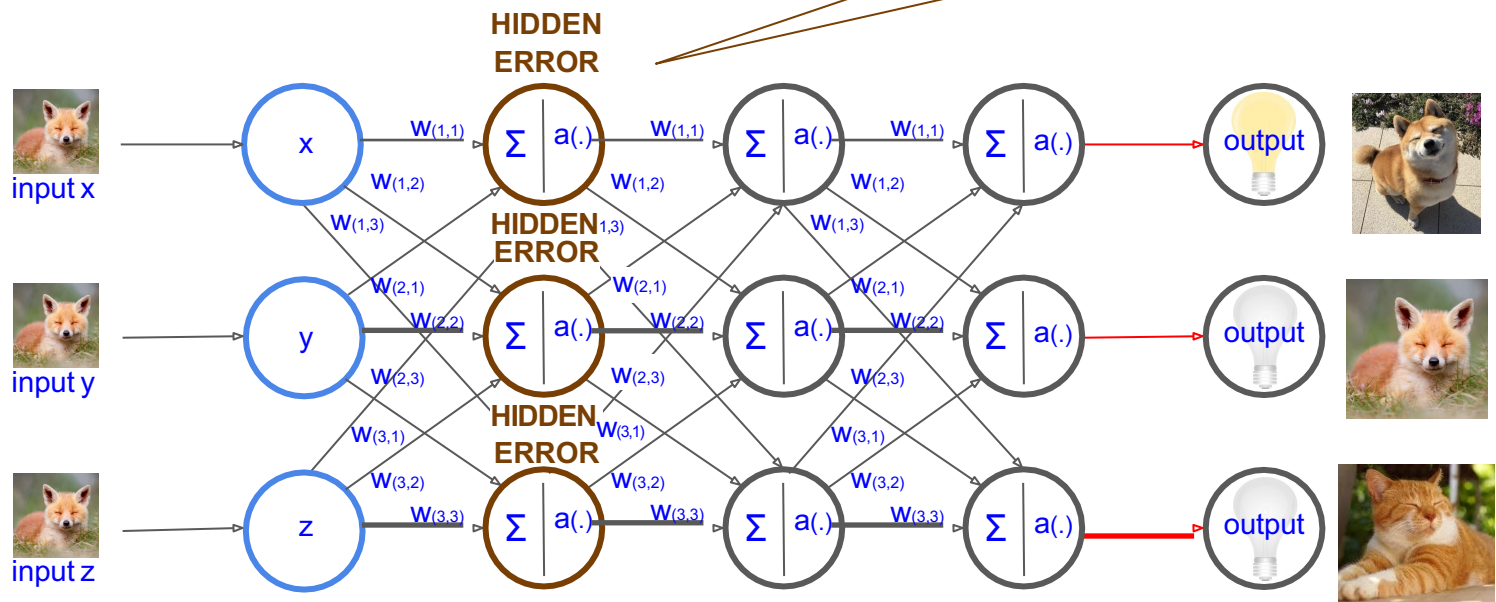
Backpropagation

$$E_{(\text{hidden})} = W^T_{(\text{hidden_output})} \cdot E_{(\text{output})}$$



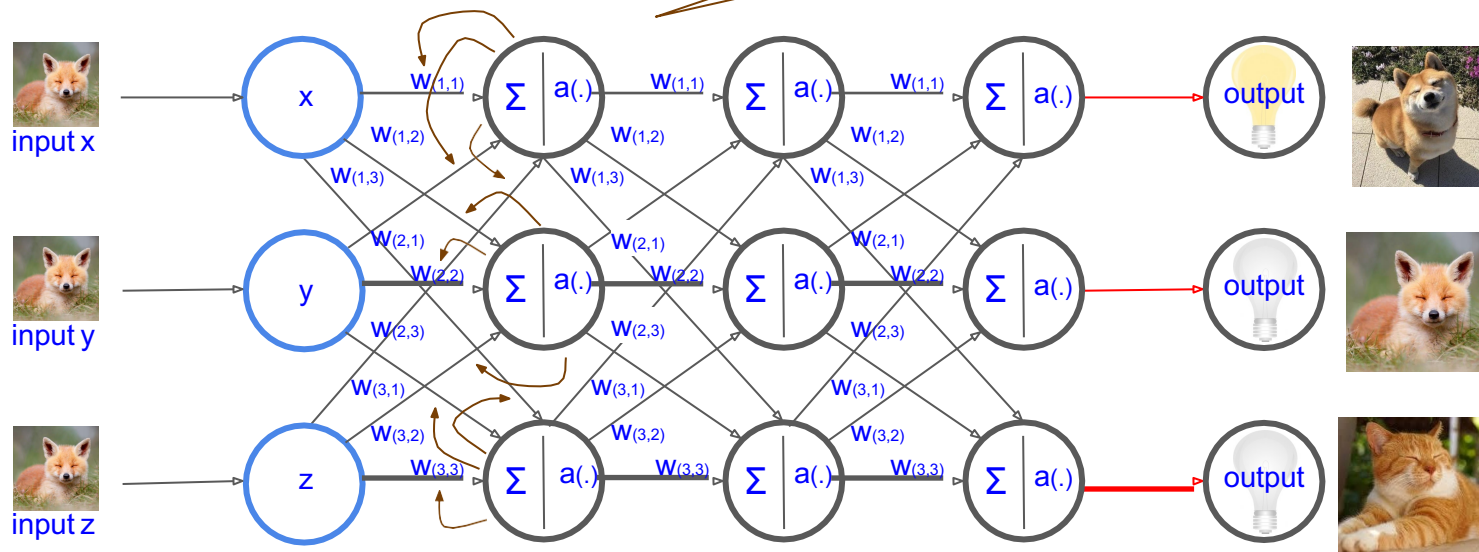
Backpropagation

$$E_{(\text{hidden})} = W^T_{(\text{hidden_output})} \cdot E_{(\text{output})}$$

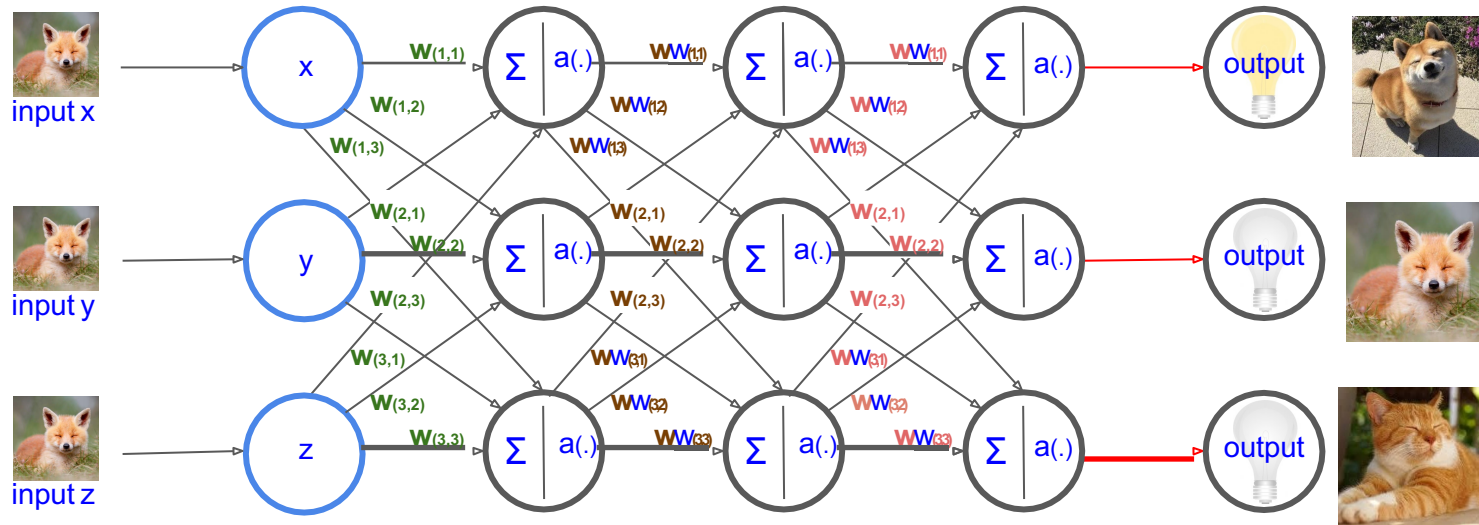


Backpropagation

$$E_{(\text{hidden})} = W^T_{(\text{hidden_output})} \cdot E_{(\text{output})}$$

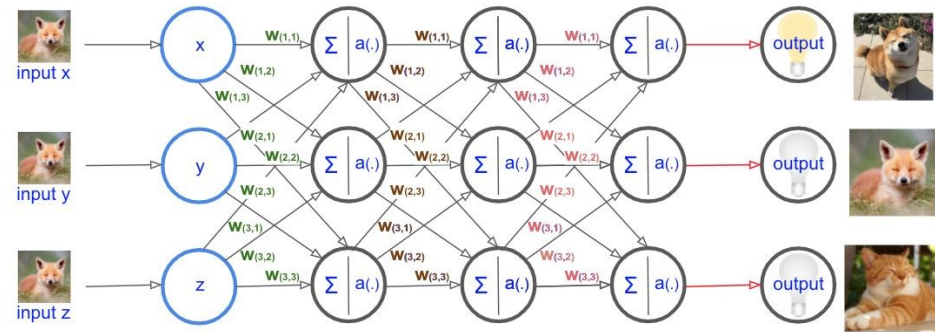
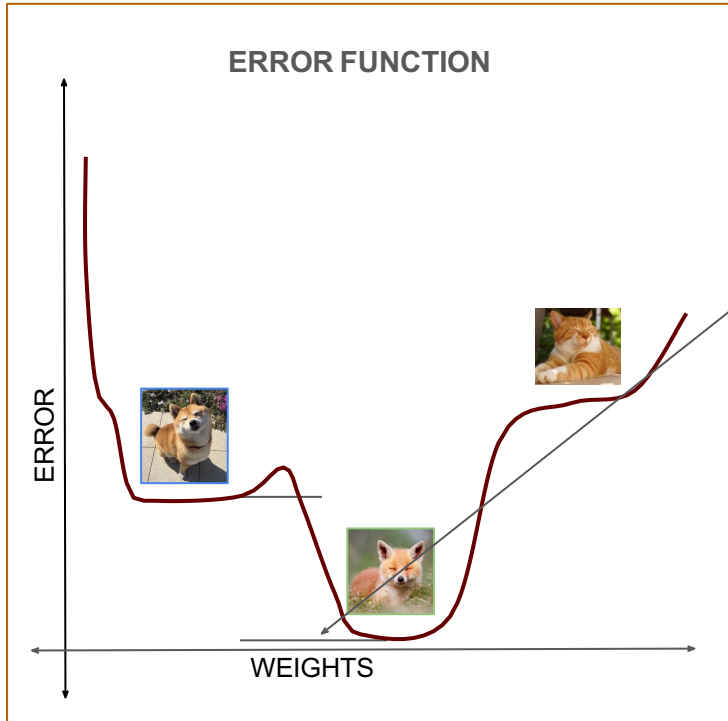


Updating Weights



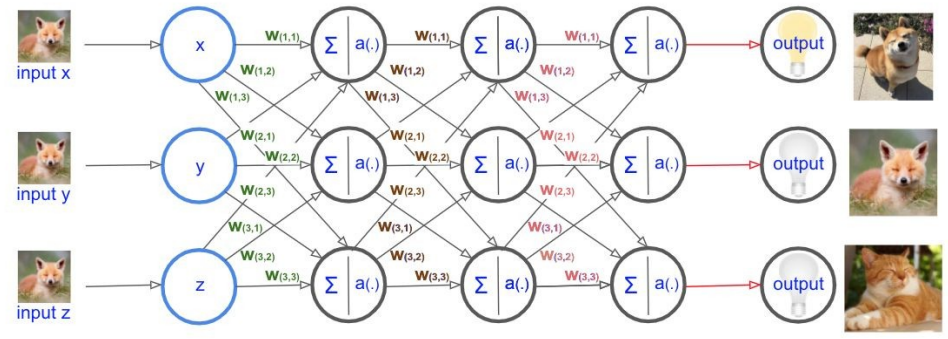
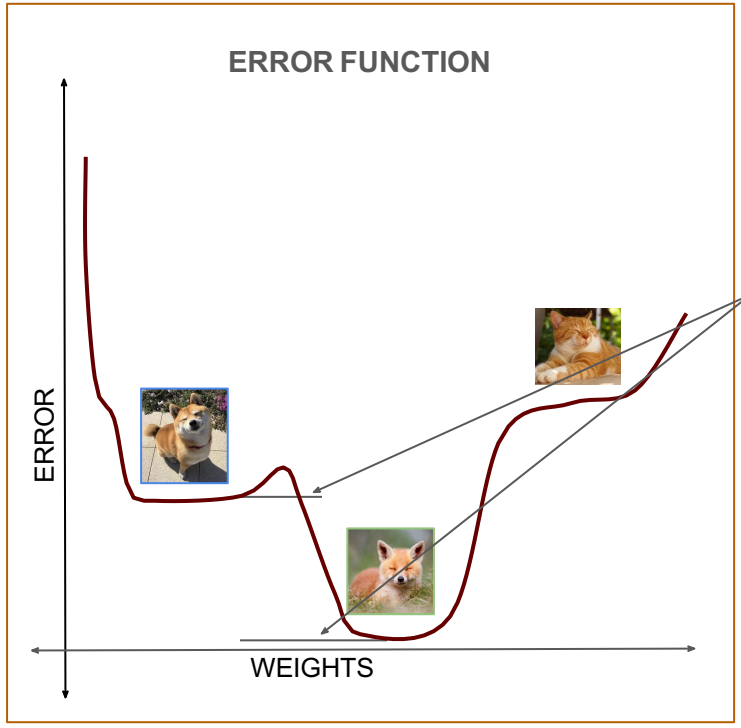
Updating Weights

$$\text{error} = \text{desired target} - \text{actual output}$$



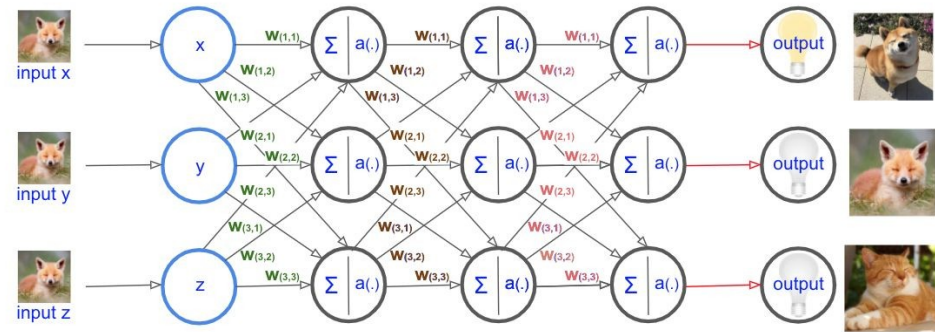
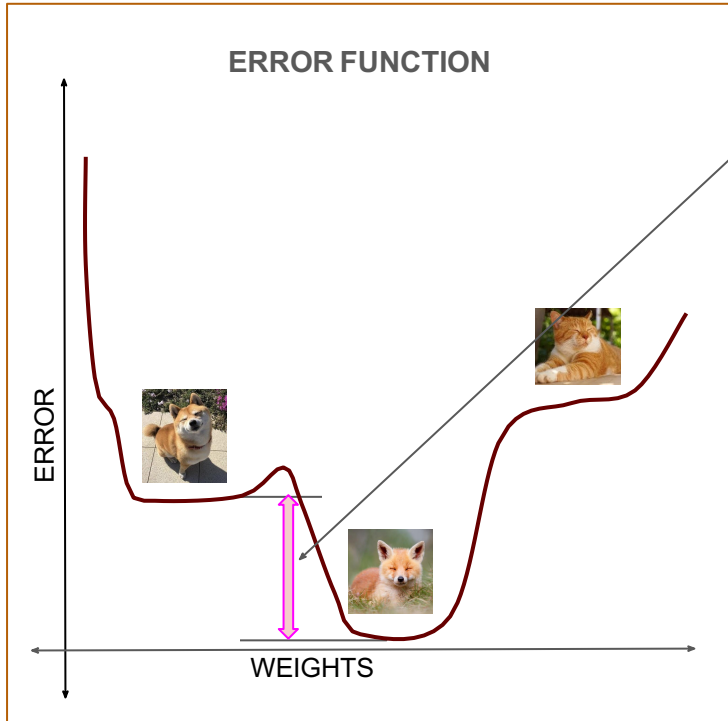
Updating Weights

$$\text{error} = \text{desired target} - \text{actual output}$$



Updating Weights

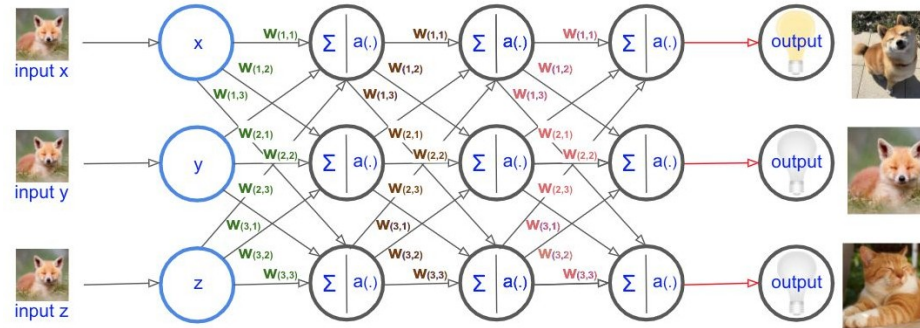
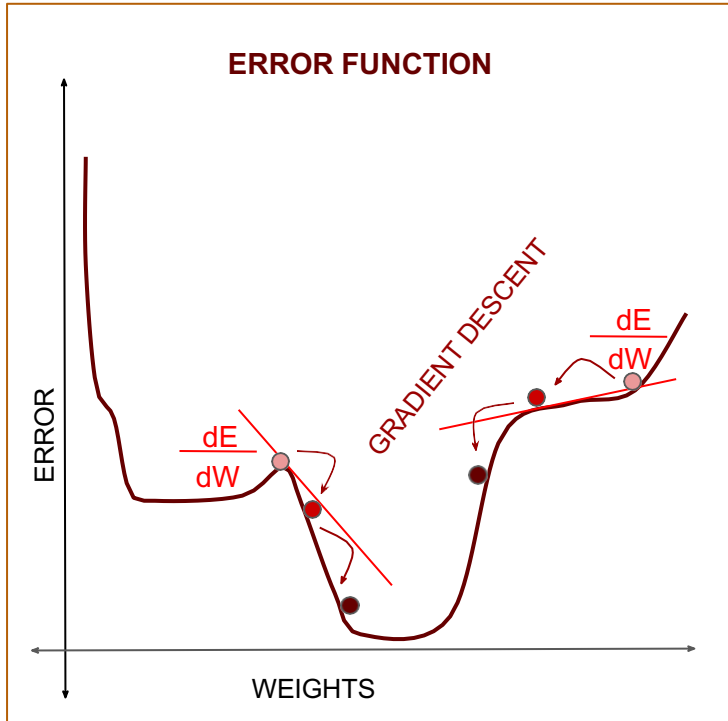
$$\text{error} = (\text{desired target} - \text{actual output})$$



Updating Weights

$$\text{error} = (\text{desired target} - \text{actual output})$$

$$\text{new_W} = (\text{old_W} - \text{lr} * \frac{dE}{d(\text{old_W})})$$

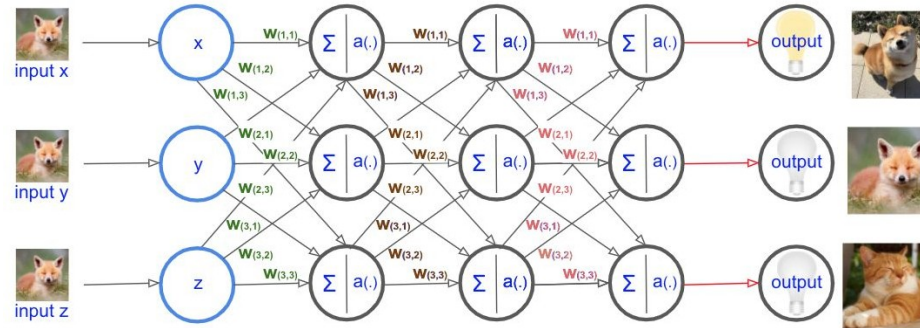
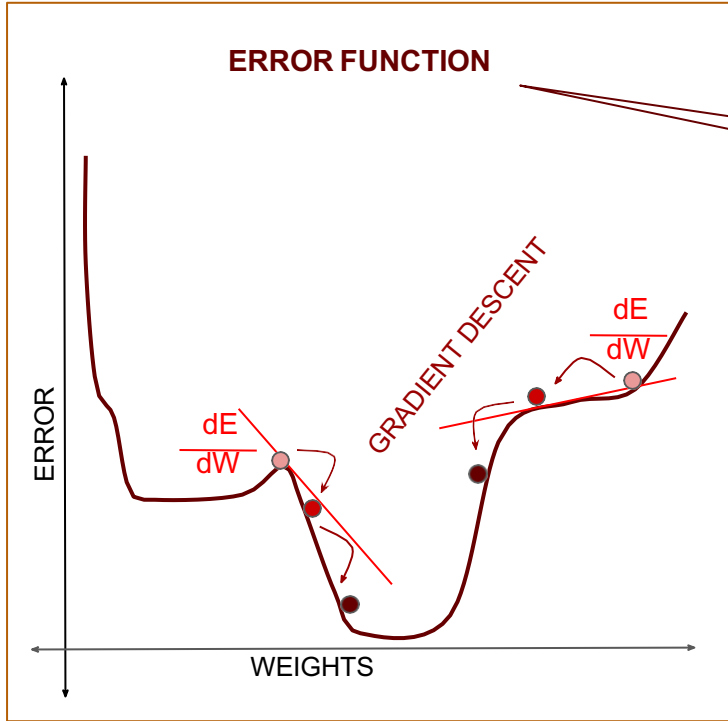


Updating Weights

$$\text{error} = (\text{desired target} - \text{actual output})$$

$$\text{new_W} = (\text{old_W} - \text{lr} * \frac{dE}{d(\text{old_W})})$$

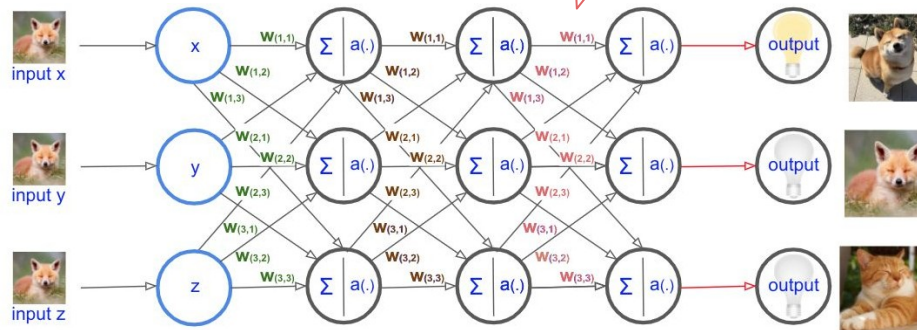
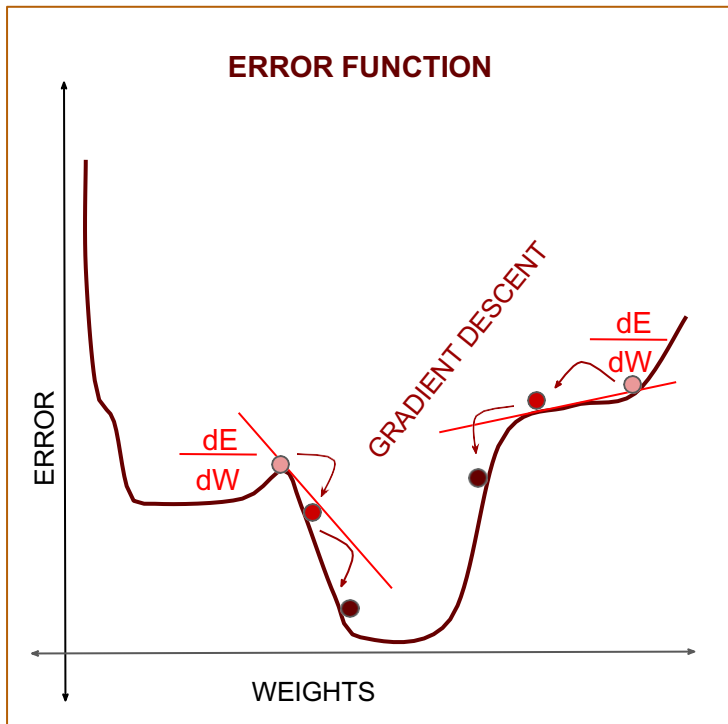
$$(\text{desired target} - \text{actual output})^2$$



Updating Weights

$$\text{error} = (\text{desired target} - \text{actual output})$$

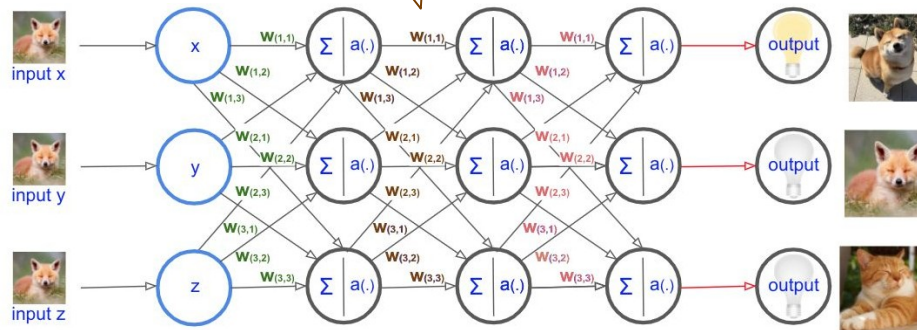
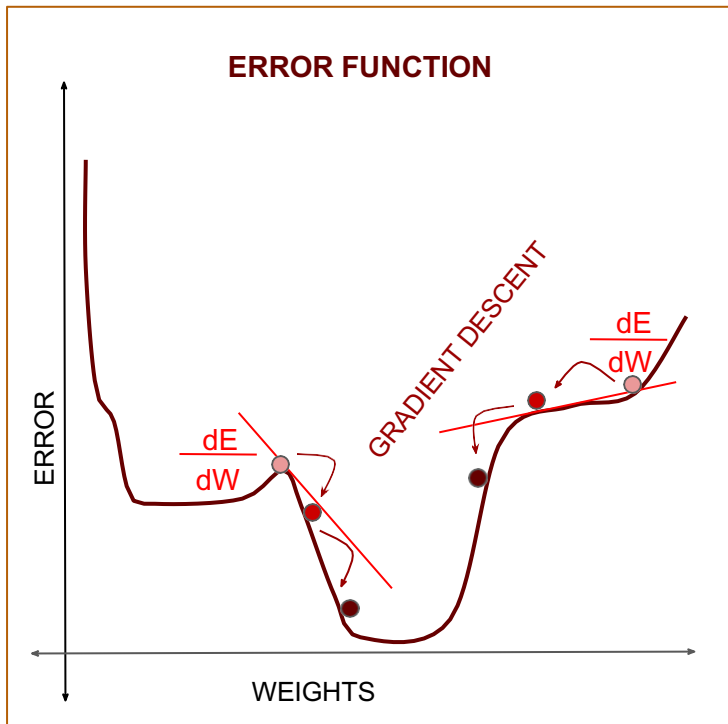
$$\text{new_W} = (\text{old_W} - \text{lr} * \frac{dE}{d(\text{old_W})})$$



Updating Weights

$$\text{error} = (\text{desired target} - \text{actual output})$$

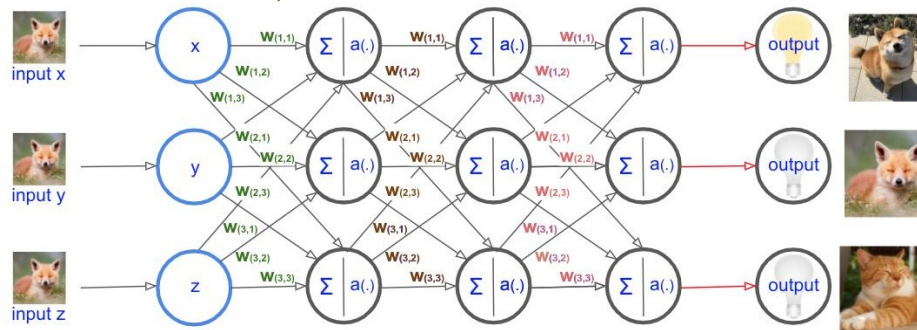
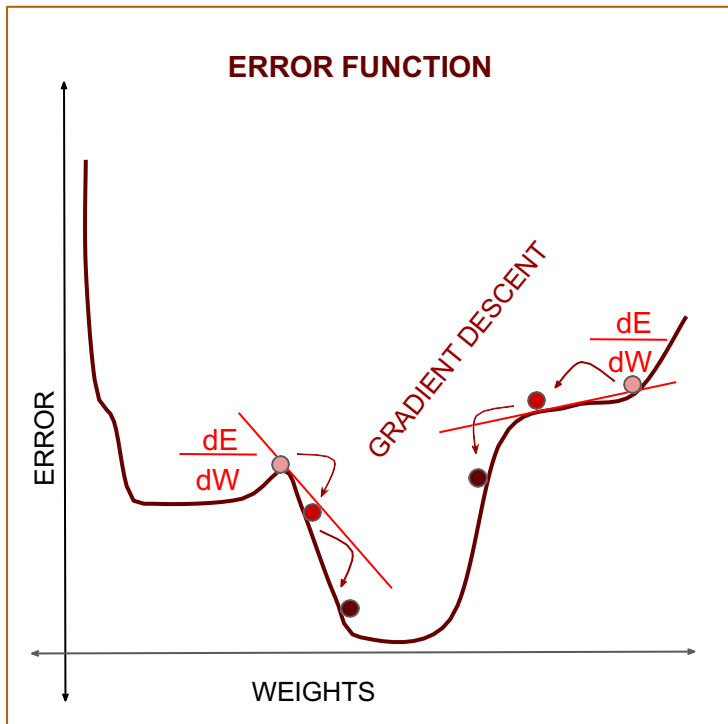
$$\text{new_W} = (\text{old_W} - \text{lr} * \frac{dE}{d(\text{old_W})})$$



Updating Weights

$$\text{error} = (\text{desired target} - \text{actual output})$$

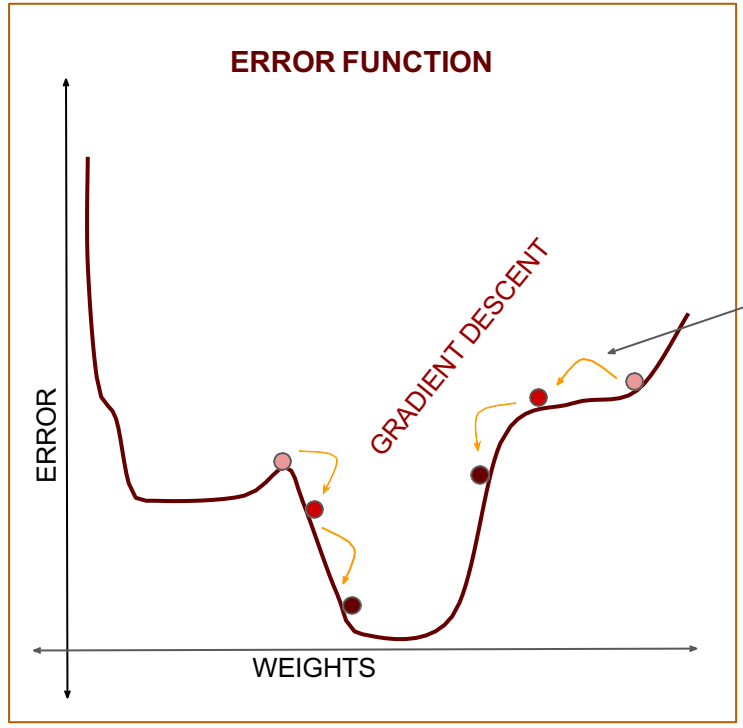
$$\text{new_W} = (\text{old_W} - \text{lr} * \frac{dE}{d(\text{old_W})})$$



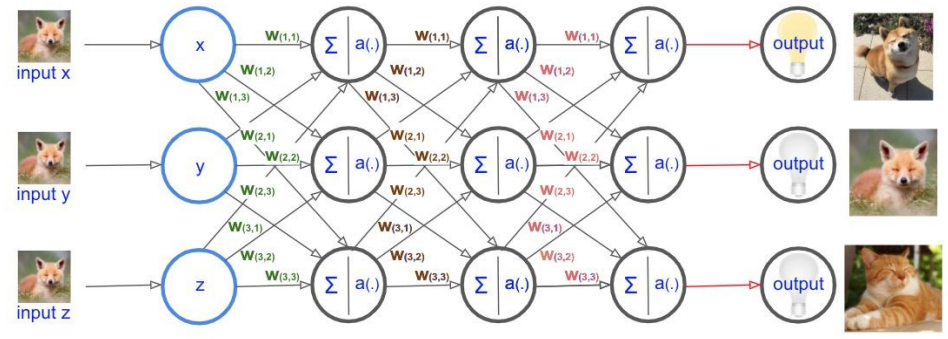
Updating Weights

$$\text{error} = (\text{desired target} - \text{actual output})$$

$$\text{new_W} = (\text{old_W} - \text{lr} * \frac{dE}{d(\text{old_W})})$$



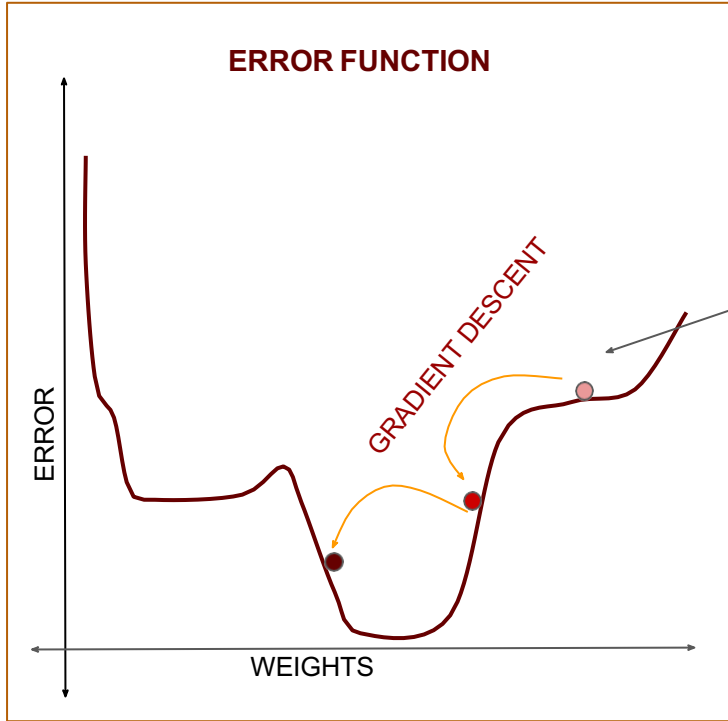
LEARNING RATE



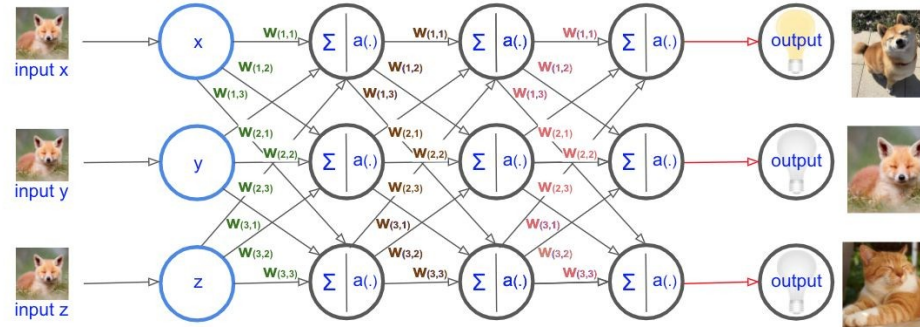
Updating Weights

$$\text{error} = (\text{desired target} - \text{actual output})$$

$$\text{new_W} = (\text{old_W} - \text{lr} * \frac{dE}{d(\text{old_W})})$$



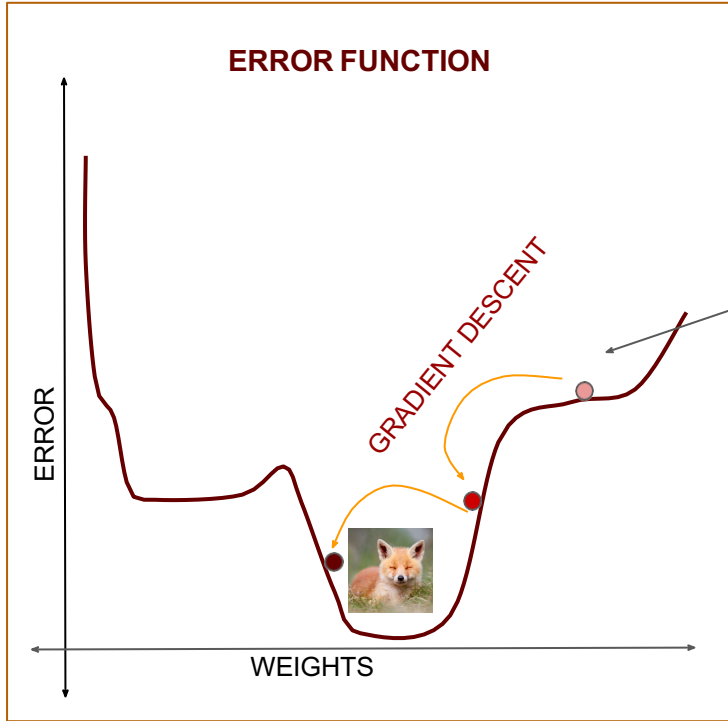
LEARNING RATE



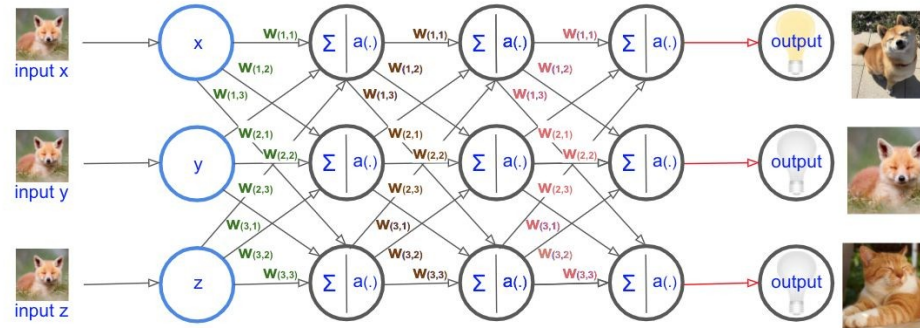
Updating Weights

$$\text{error} = (\text{desired target} - \text{actual output})$$

$$\text{new_W} = (\text{old_W} - \text{lr} * \frac{dE}{d(\text{old_W})})$$



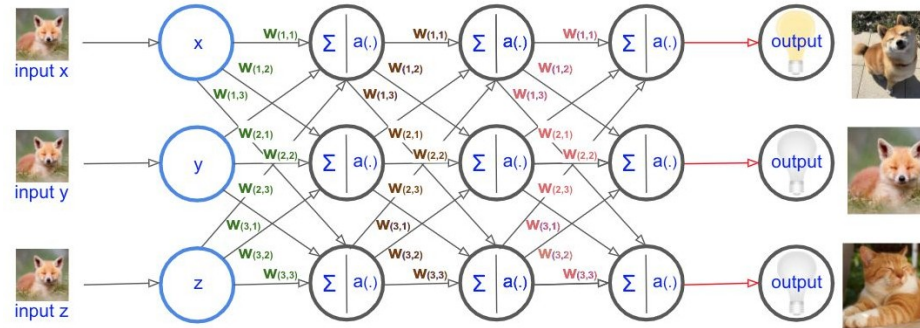
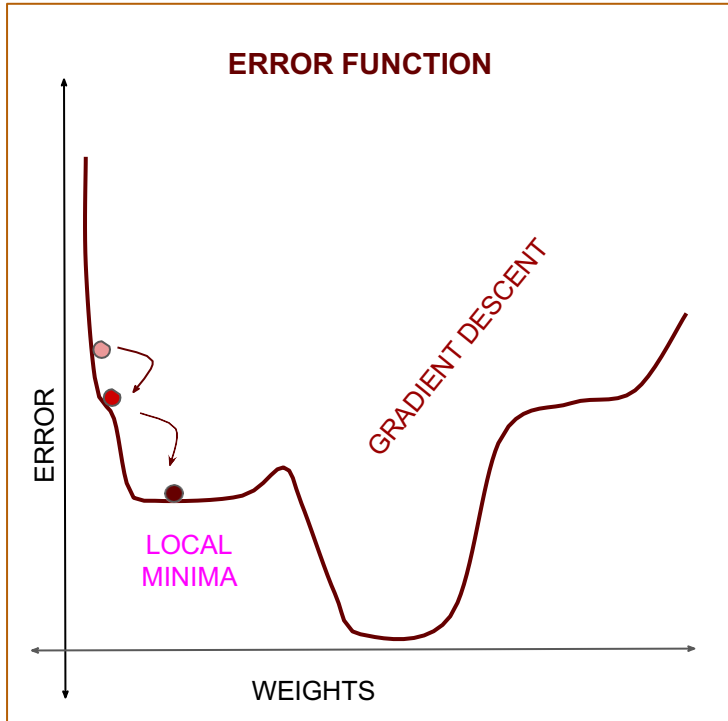
LEARNING RATE



Updating Weights

$$\text{error} = (\text{desired target} - \text{actual output})$$

$$\text{new_W} = (\text{old_W} - \text{lr} * \frac{dE}{d(\text{old_W})})$$



Updating Weights

$$\text{error} = (\text{desired target} - \text{actual output})$$

$$\text{new_W} = (\text{old_W} - \text{lr} * \frac{dE}{d(\text{old_W})})$$

