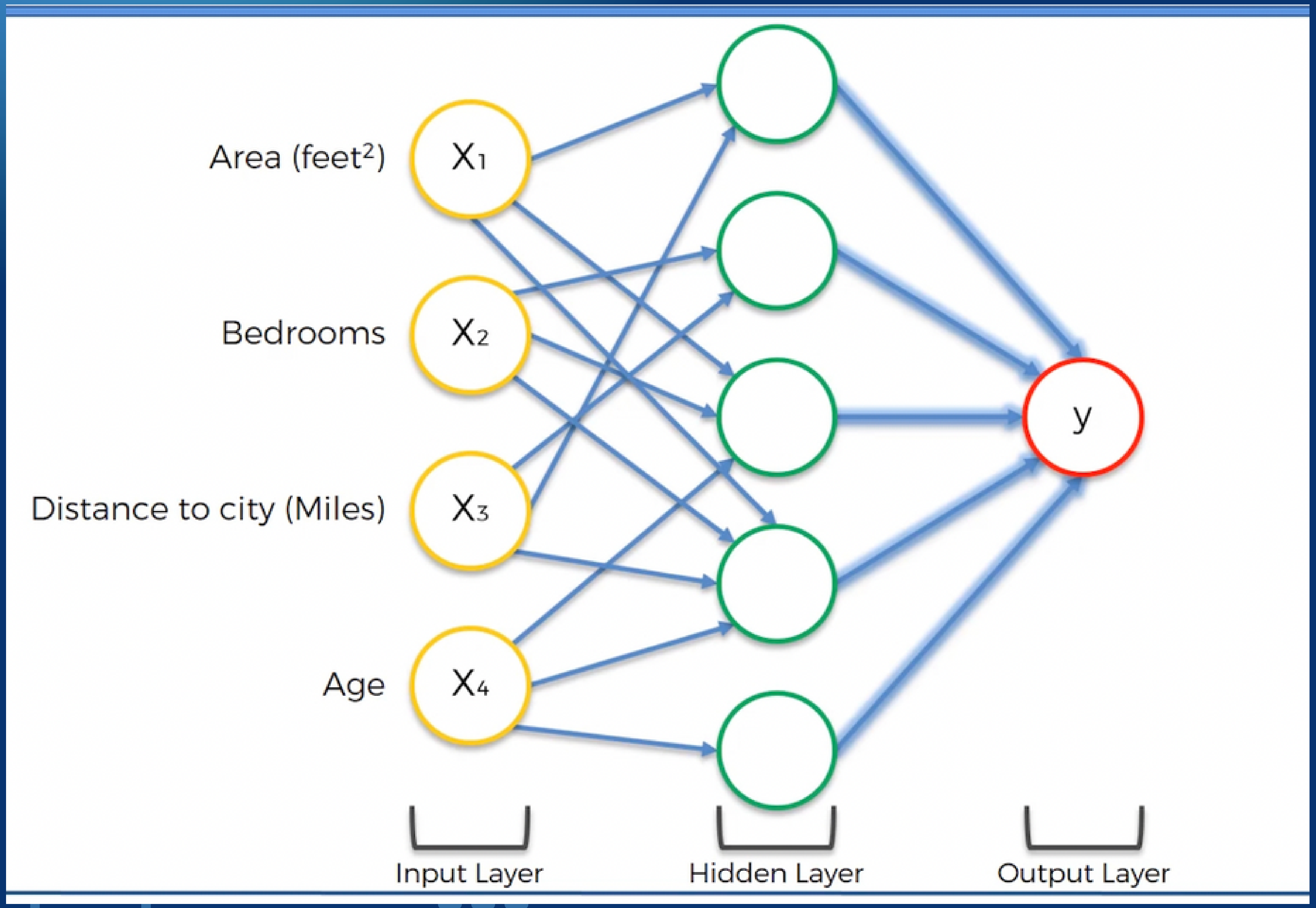
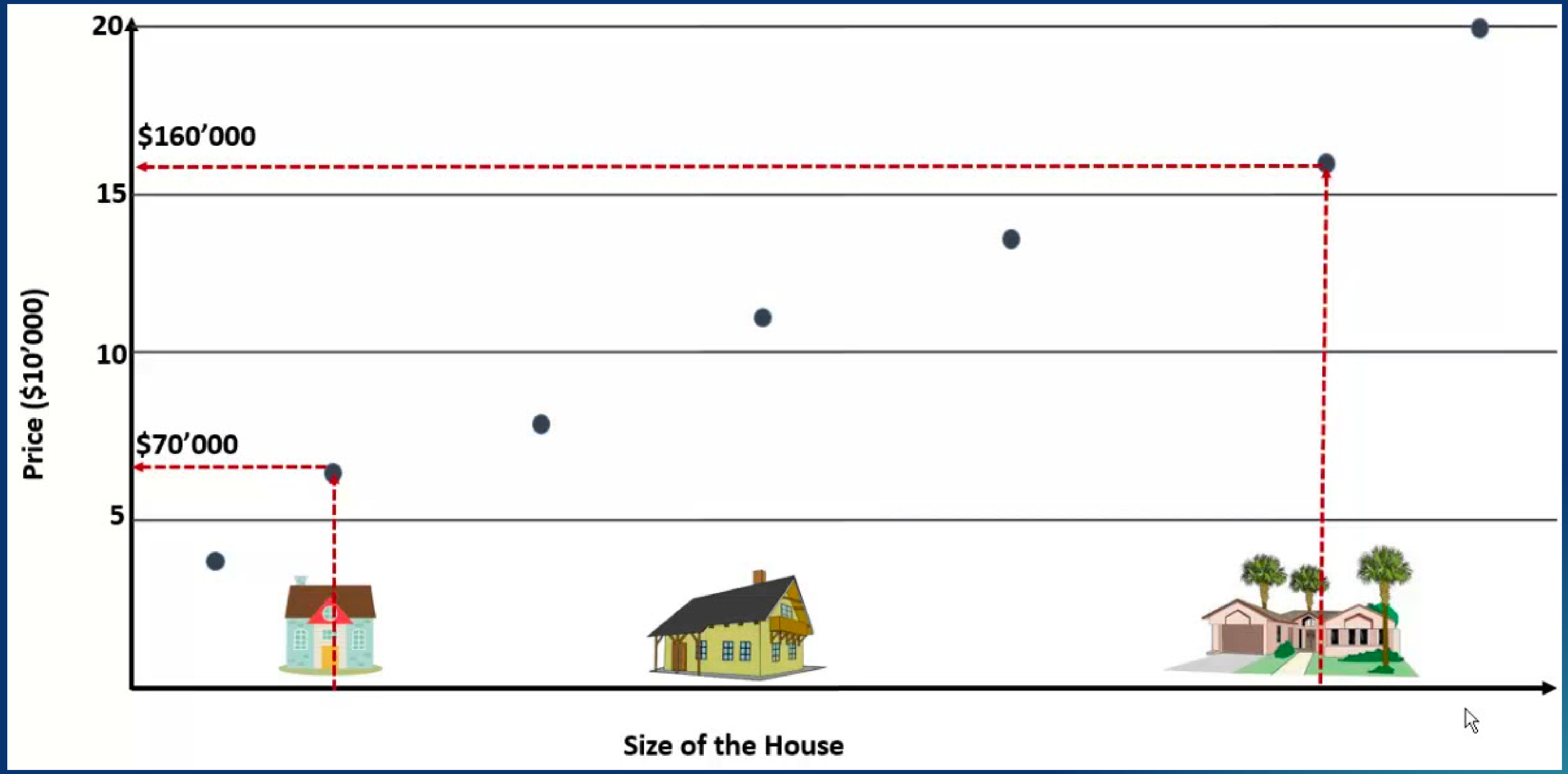
NEURAL NETWORKS

INTRODUCTION TO RECURRENT NEURAL NETWORKS

Abhiram Siddanthi & Sreelekshmi Kishore

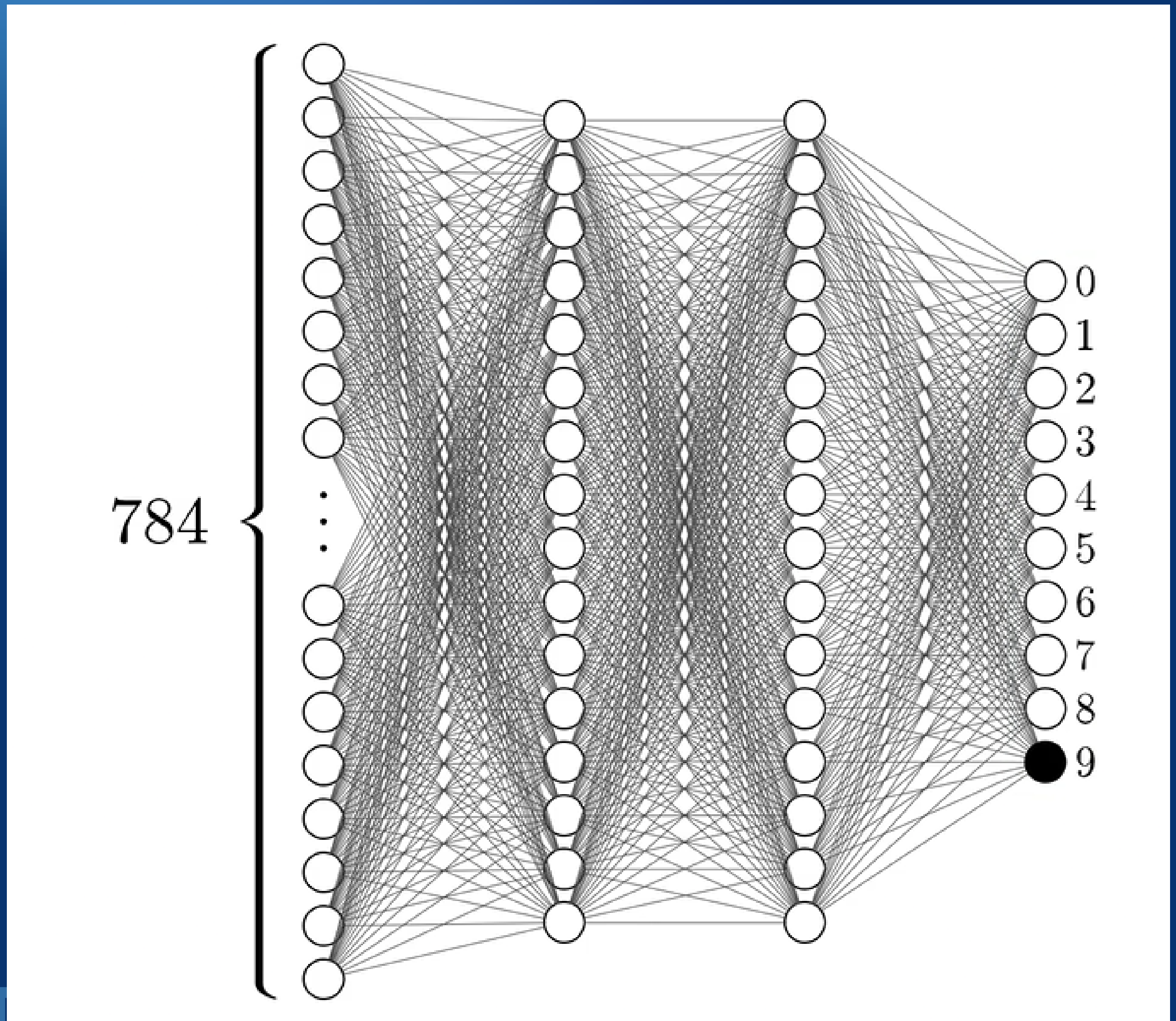
DEFINING THE IDEA





NEURAL NETWORK - STRUCTURE

Plain Vanilla form of a network

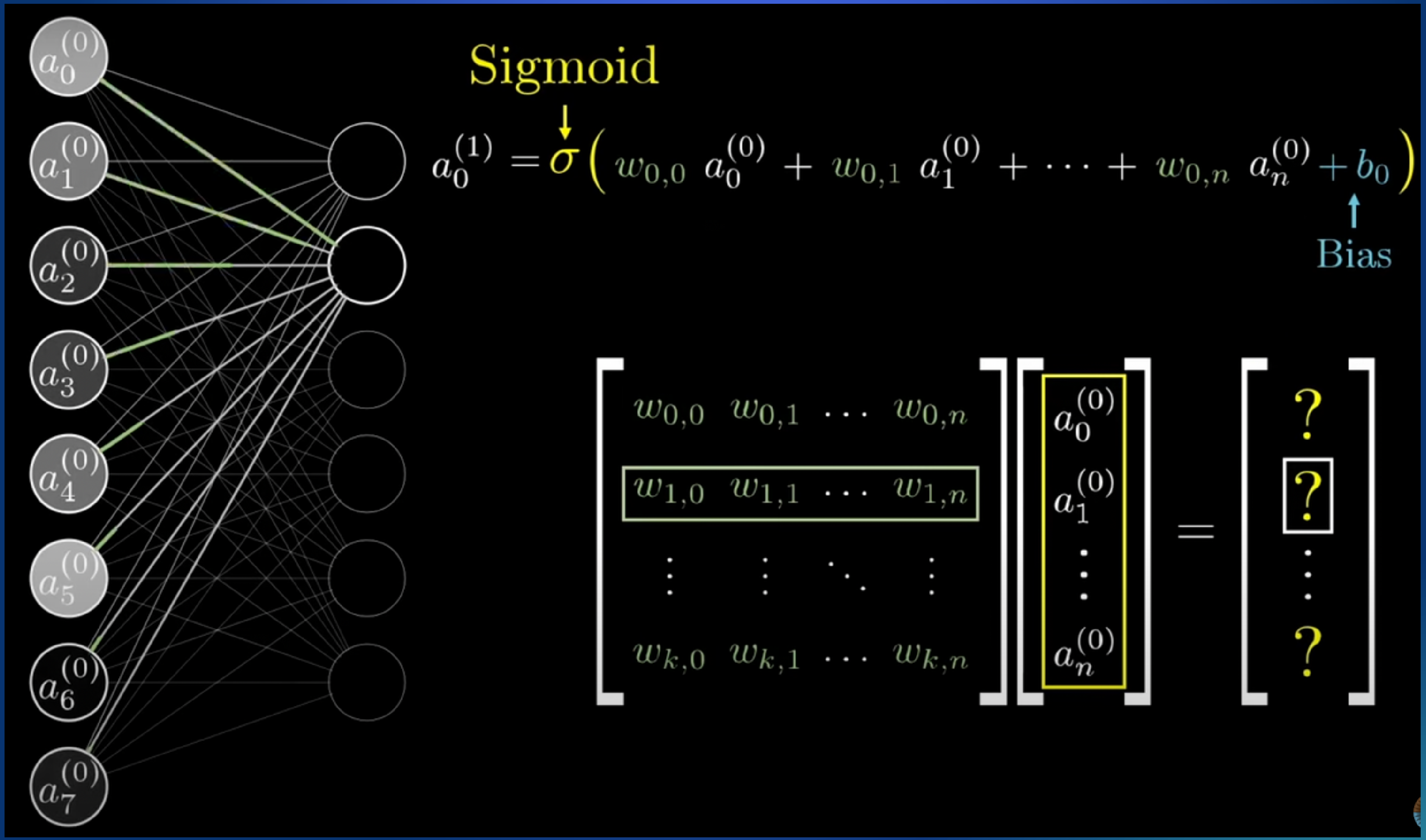


NEURONS

- Assuming each neuron stores a value between 0 and 1. The first layer of neurons takes the initial input and the last layer gives the output
- Activation in one layer determines the activation of the next layer
- The input given by the user decides the initial activations. The middle layers/ hidden layers compute the required result and the final layer gives the required output

HOW THE NETWORK WORKS

- Activation is calculated as: (a1w1 + a2w2 + a3w3 + ... + anwn - x]
- But since activation has to be between 0 & 1 so we use the sigmoid function or Relu to achieve this
- It is done for each connection in the network and the activation value for next neuron is calculated
- Weights and biases are decided through gradient descent



$$\sigma\left(\mathbf{W}\mathbf{a}^{(0)} + \mathbf{b}\right)$$

GRADIENT DESCENT

The basic concept of gradient descent is finding the minima like a regular calculus problem

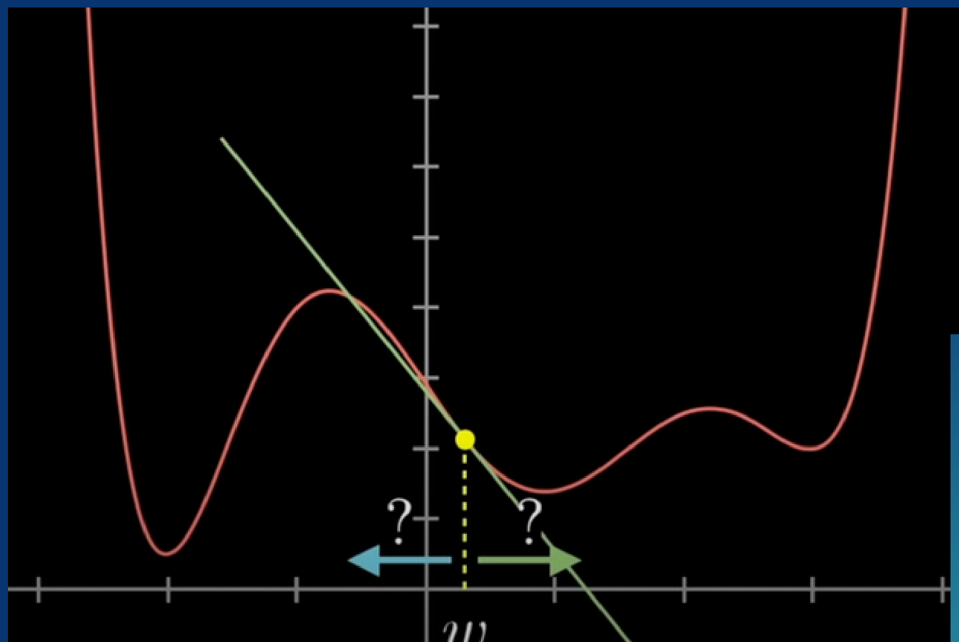
- Then we find a cost function which is our way of calculating the amount of error that we have in our training example.
- Cost of a single training ~~example: sum of squares of differences between each of~~ the output activations and the value we require them to have.

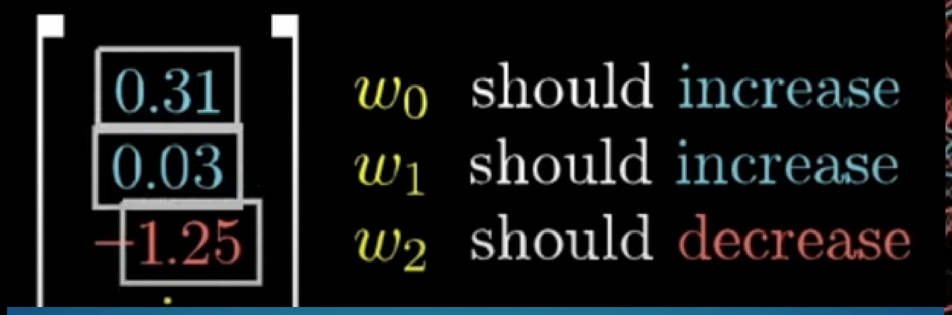
Goal: Reduce Cost

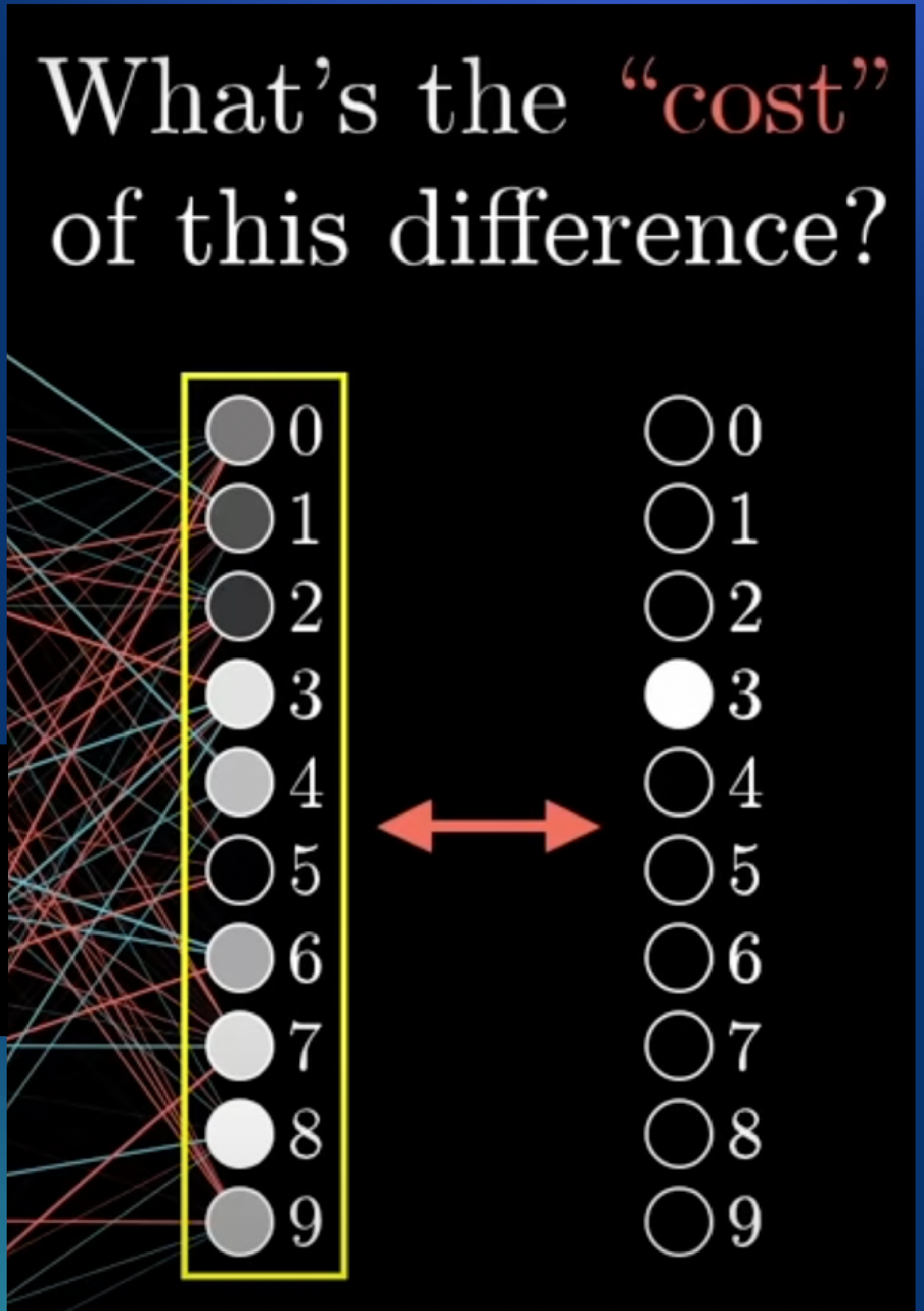
Method: Change the values of the initially initialized weights and biases

Cost Low

Cost High



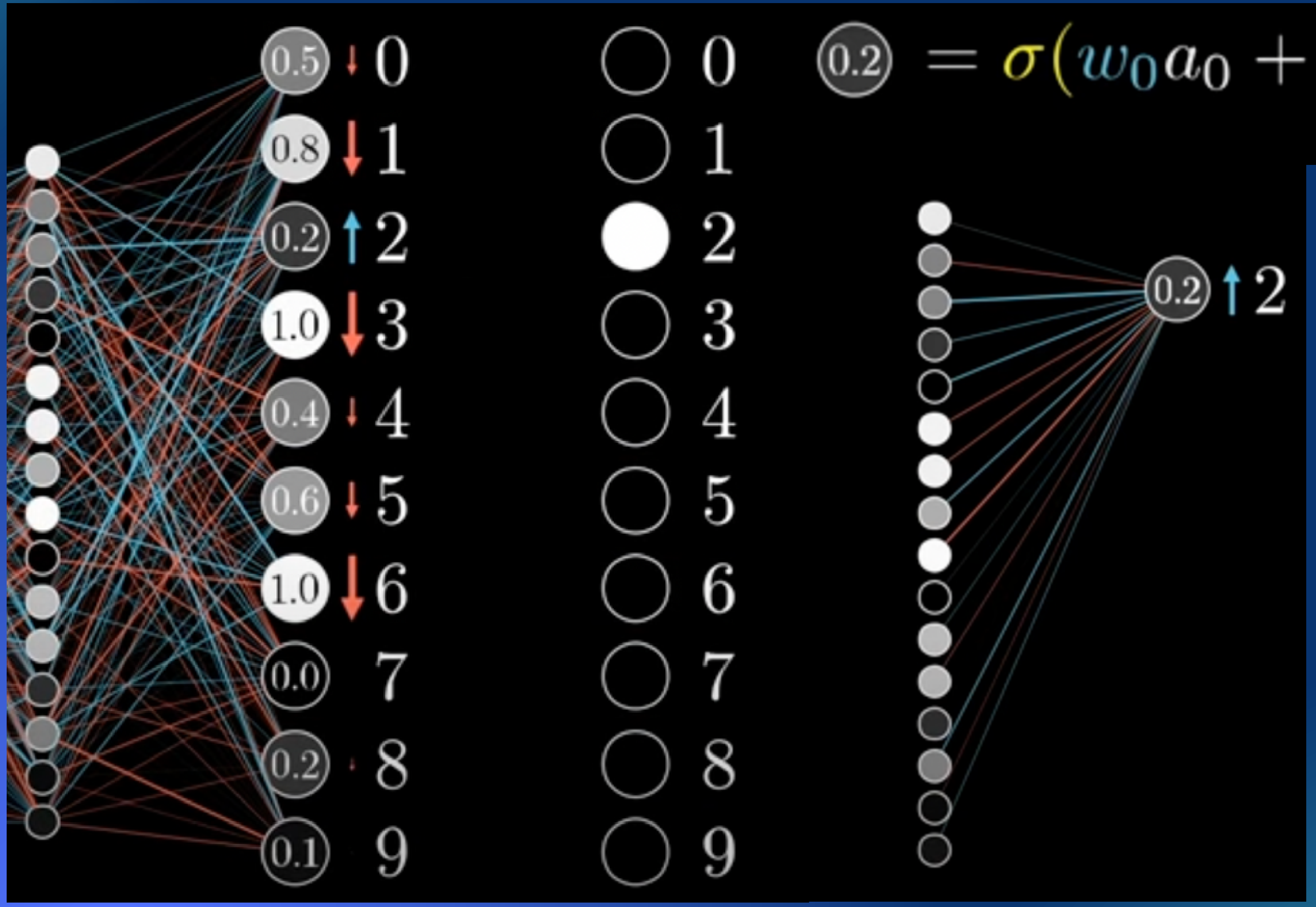




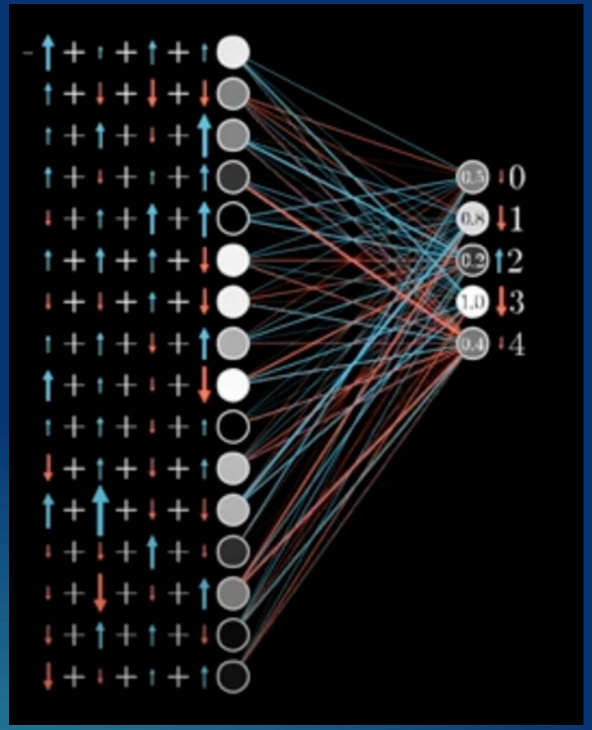
BACKPROPOGATION

Start out with the final layer- we know how it is supposed to look like.

Using the information for the required output we can then decide what the activations of the previous layers' neurons need to be.



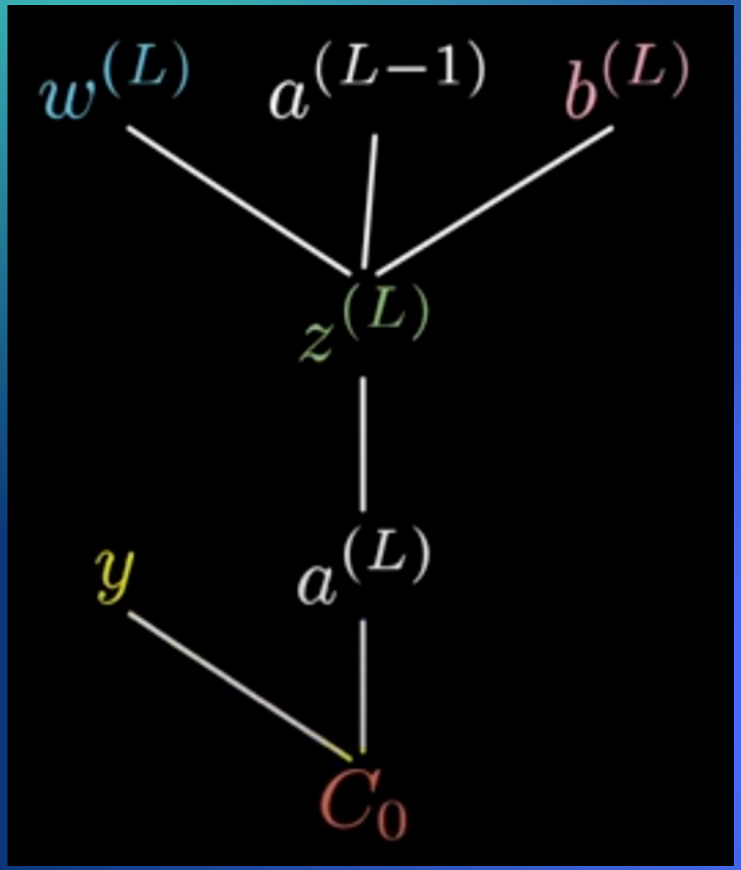
increase b
increase weights
increase activation



Higher the activation of neuron → more contribute to the sum → weight is more sensitive → change in thiss will have higher impact

"Neurons that fire together wire together."

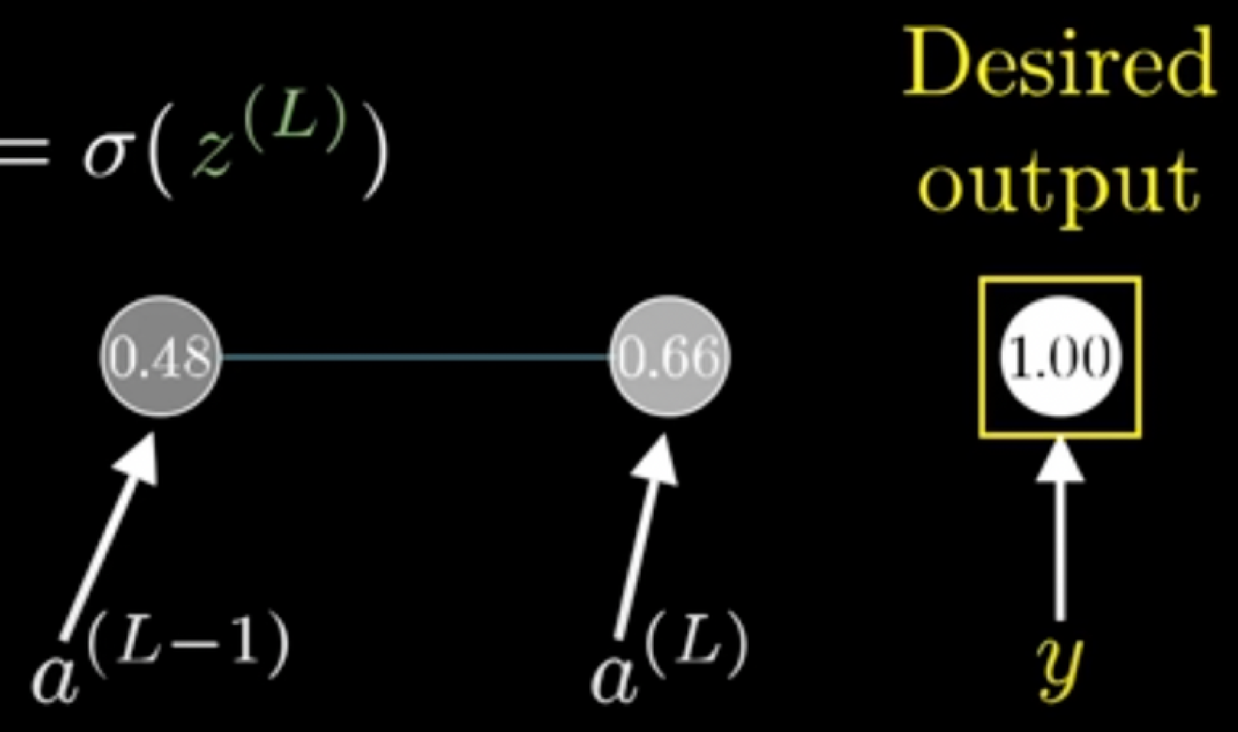
BACKPROPOGATION- CALCULUS



$$\text{Cost} \longrightarrow C_0(\dots) = (a^{(L)} - y)^2$$

$$z^{(L)} = w^{(L)} a^{(L-1)} + b^{(L)}$$

$$a^{(L)} = \sigma(z^{(L)})$$



$$\frac{\partial C_0}{\partial w^{(L)}} = \frac{\partial z^{(L)}}{\partial w^{(L)}} \frac{\partial a^{(L)}}{\partial z^{(L)}} \frac{\partial C0}{\partial a^{(L)}}$$

$$\frac{\partial C0}{\partial a^{(L)}} = 2(a^{(L)} - y)$$

$$\frac{\partial a^{(L)}}{\partial z^{(L)}} = \sigma'(z^{(L)})$$

$$\frac{\partial z^{(L)}}{\partial w^{(L)}} = a^{(L-1)}$$

- The activation value of the previous neuron plays a large role in the sensitivity of cost function to the change in weight.
- To find the effect of all the weights of training examples we need to sum up all the derivatives and take their average
- Column matrix containing the changes of each weight and bias for every training example

ARTIFICIAL NEURAL NETWORK

- ANNs have densely interconnected small units known as neurons and each unit takes a real-valued input and gives out a real-valued output.
- Different frameworks for ANNs including Feedforward neural networks, Recurrent neural networks, Convolutional neural networks, and Autoencoders
- These are defined as the way the neurons in ANN are connected to each other
- The neural network structure initially discussed is the most simple form of a neural network whose architecture is referred to as a Feedforward neural network where the information flows through the neurons during computation only in the forward direction.

RECURRENT NEURAL NETWORK

- It is a network that takes the output from the previous step and uses it as input for the next step.
- It also retains information from the past and uses that information to process new input.
- Used for ordinal or temporal problems like language translation, natural language processing(NLP), speech recognition, and image captioning.

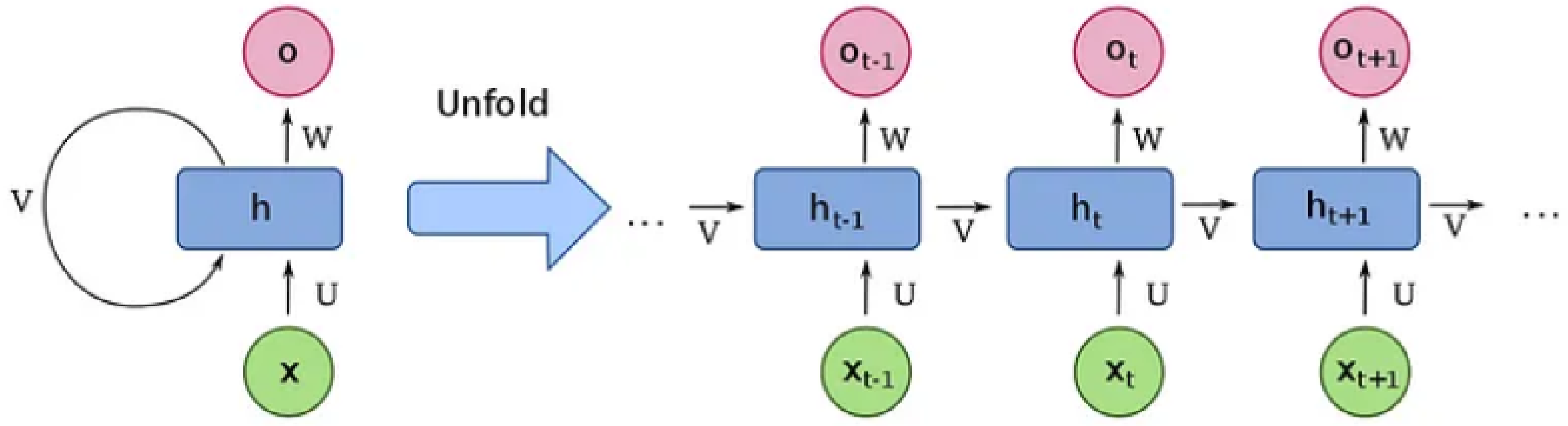
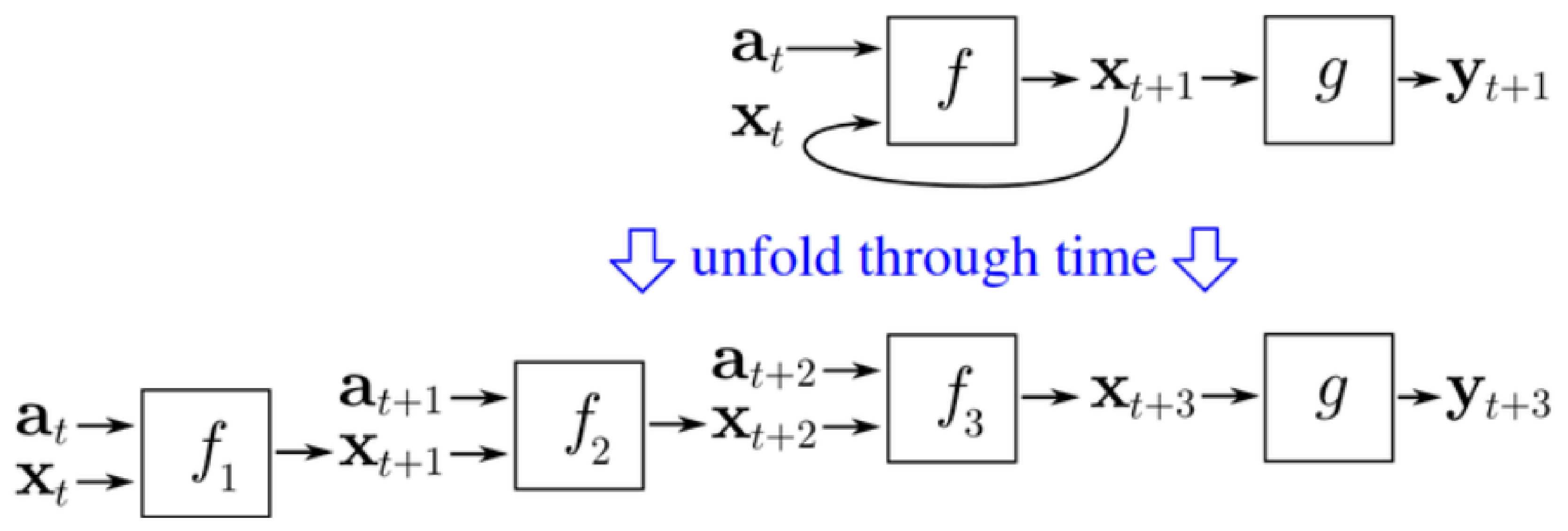


Fig 2: Unfolded RNN layer

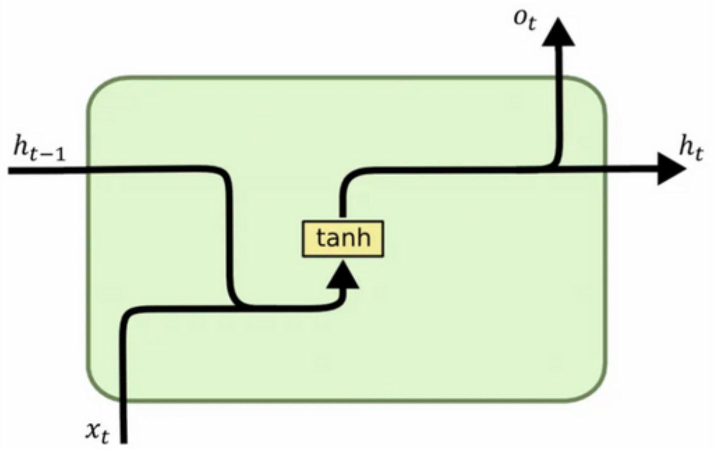
BACK PROPAGATION THROUGH TIME

- We use backpropagation to set the weights and biases using the value of the gradient.
- We know that the value of the gradient at time t is dependent on the value of the gradient at all the times after time t.
- BPTT begins by unfolding a recurrent neural network in time.
- Consider an example of a neural network that contains a recurrent layer f and a feedforward layer g.
- The aggregated cost is always the average of the costs of each of the time steps.



MATH BEHIND RNN

- It is a network that takes the output from the previous step and uses it as input for the next step. It also retains information from the past and uses that information to process new input.
- Used for ordinal or temporal problems like language translation, natural language processing(NLP), speech recognition, and image captioning.



$$softmax(z_i) = \frac{exp(z_i)}{\sum_j exp(z_j)}$$

$$a_t = X_t U + h_{t-1} V + b$$
$$h(t) = f(a_t)$$
$$O_t = c + W h_t$$
$$\hat{y}_t = softmax(O_t)$$

THANK YOU