Deep Neural Networks

FPT UNIVERSITY

Learning Objectives

- Describe the successive block structure of a deep neural network
- Build a deep L-layer neural network
- Analyze matrix and vector dimensions to check neural network implementations
- Use a cache to pass information from forward to back propagation
- Explain the role of hyperparameters in deep learning

Deep Neural Networks

FPT UNIVERSITY

1 Deep L-layer Neural network
2 Forward Propagation in a Deep Network
3 Getting your matrix dimensions right
4 Why deep representations?
5 Building blocks of deep neural networks
6 Forward and backward propagation
7 Parameters vs Hyperparameters
8 What does this have to do with the brain?

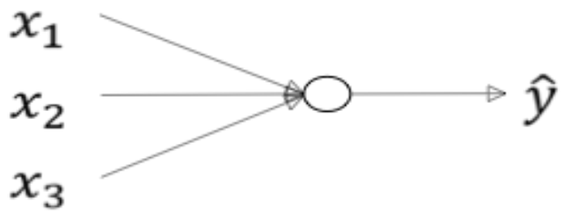
FPT Education

FPT UNIVERSITY

Deep L-layer Neural network

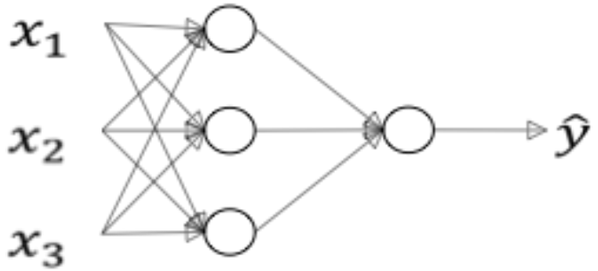
What is a deep neural network?

"Shallow"

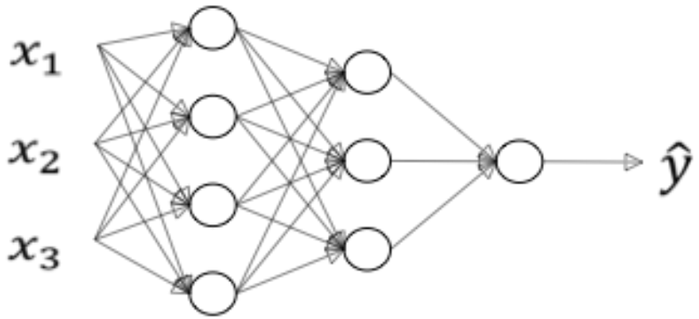


logistic regression

"2 layer" NN

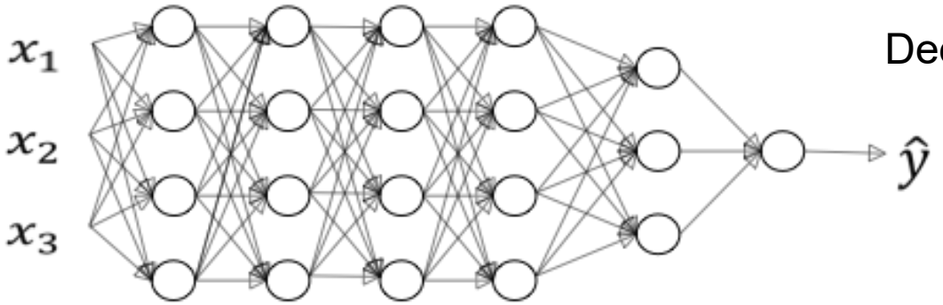


1 hidden layer



2 hidden layers

Deep

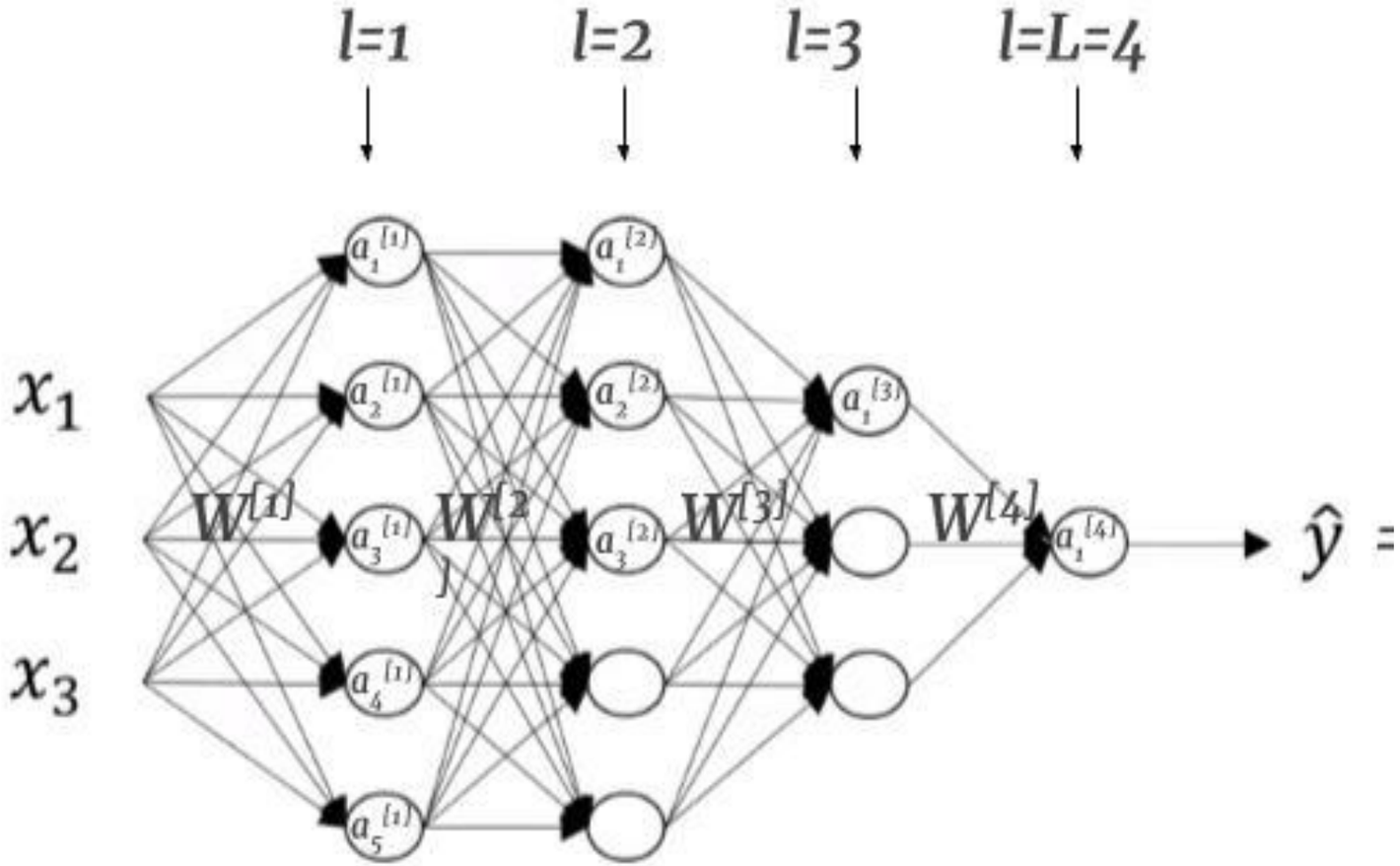


5 hidden layers

Deep neural network notation

- Deep neural networks have multiple hidden layers and can learn functions that shallower models cannot.
- The notation used for deep neural networks:
  - l, the number of layers
  - $n^{[l]}$, the number of nodes in layer l
  - $a^{[l]}$ is the activation in layer l
  - $W^{[l]}$ are the weights for computing the value $z^{[l]}$ in layer l
  - $b^{[l]}$ is used to compute $z^{[l]}$.
  - The input features are called x
  - $\hat{y}$ is the predicted output of the neural network.

Deep neural network notation



FPT Education

FPT UNIVERSITY

Deep Neural Networks

Forward Propagation in a Deep Network

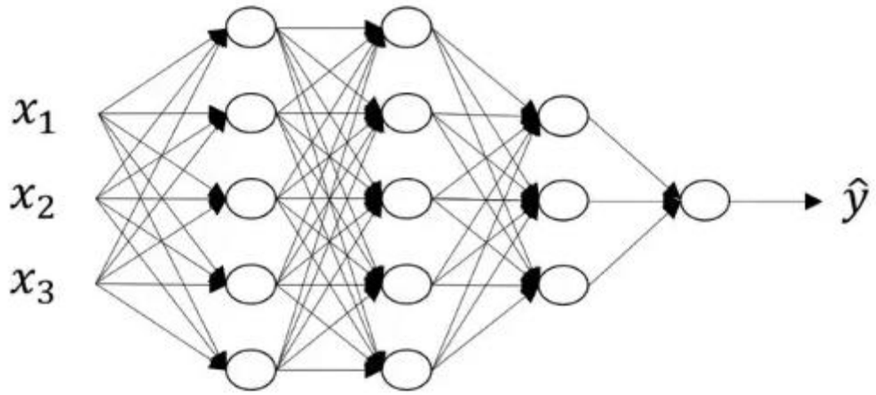
Forward Propagation in a Deep Network

- Forward propagation is the process of computing the output of a deep neural network given an input. Here are the general rules of forward propagation in a deep network:
  - Each neuron in a layer is connected to every neuron in the next layer.
  - The output of each neuron in a layer is computed by applying a non-linear activation function to the weighted sum of the inputs to that neuron.
  - The weights connecting the neurons in each layer are learned through a process called backpropagation, which involves calculating the gradient of the loss function with respect to the weights and updating them using an optimization algorithm like stochastic gradient descent.

Forward Propagation in a Deep Network

- The input to the network is fed into the first layer of neurons, which compute their outputs using the input and the weights connecting them to the input layer.
- The outputs of the neurons in the first layer are then fed into the next layer of neurons, which compute their outputs using the outputs of the previous layer and the weights connecting them.
- This process continues until the output of the final layer is computed, which represents the output of the network given the input.

Forward Propagation in a Deep Network



Do the calculation with each hidden layer → output layer is similar to neural network with 2 layers.

Hidden layer 1: $z^{[1]} = W^{[1]}a^{[0]} + b^{[1]} \quad (a^{[0]} = x) \quad a^{[1]} = g(z^{[1]})$

Hidden layer 2: $z^{[2]} = W^{[2]}a^{[1]} + b^{[2]} \quad a^{[2]} = g(z^{[2]})$

Generalization for layer n: $z^{[l]} = W^{[l]}a^{[l-1]} + b^{[l]} \quad a^{[l]} = g(z^{[l]})$

Perform vectorization for all training examples for l = 1..L

$$Z^{[l]} = W^{[l]}A^{[l-1]} + b^{[l]}$$

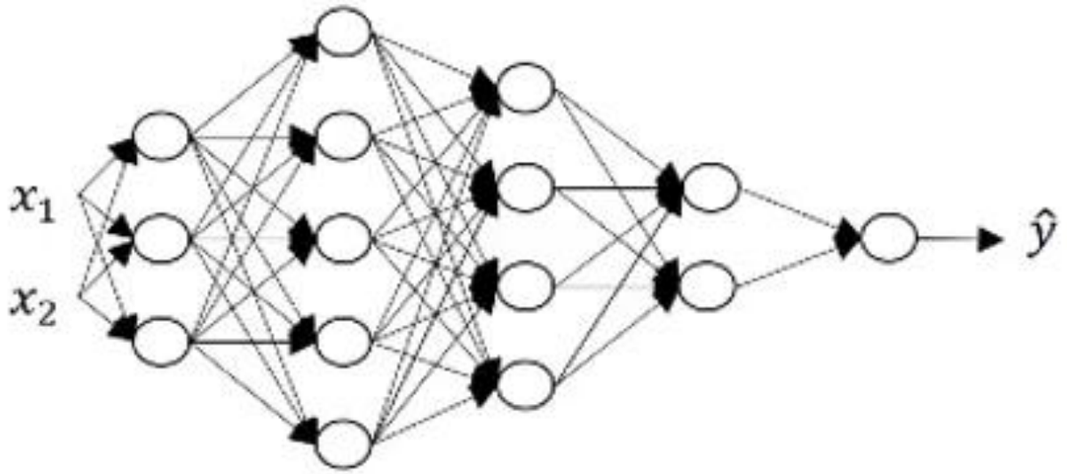
$$A^{[l]} = g(Z^{[l]}) \quad \text{Where } A^{[0]} = X$$

FPT UNIVERSITY

Getting your matrix dimensions right

Getting your matrix dimensions right

- Analyzing the dimensions of a matrix is one of the best debugging tools to check how correct our code is. We will discuss what should be the correct dimension for each matrix in this section. Consider the following example:



- There are 4 hidden layers and 1 output layer. The units in each layer are:

$$n^{[0]} = 2, n^{[1]} = 3, n^{[2]} = 5, n^{[3]} = 4, n^{[4]} = 2, \text{ and } n^{[5]} = 1$$

Getting your matrix dimensions right

- The generalized form of dimensions of W, b and their derivatives is:
  - $W^{[l]} = (n^{[l]}, n^{[l-1]})$
  - $b^{[l]} = (n^{[l]}, 1)$
  - $dW^{[l]} = (n^{[l]}, n^{[l-1]})$
  - $db^{[l]} = (n^{[l]},1)$
  - Dimension of $Z^{[l]}, A^{[l]}, dZ^{[l]}, dA^{[l]} = (n^{[l]},m)$
- where ‘m’ is the number of training examples. These are some of the generalized matrix dimensions which will help you to run your code smoothly.

FPT UNIVERSITY

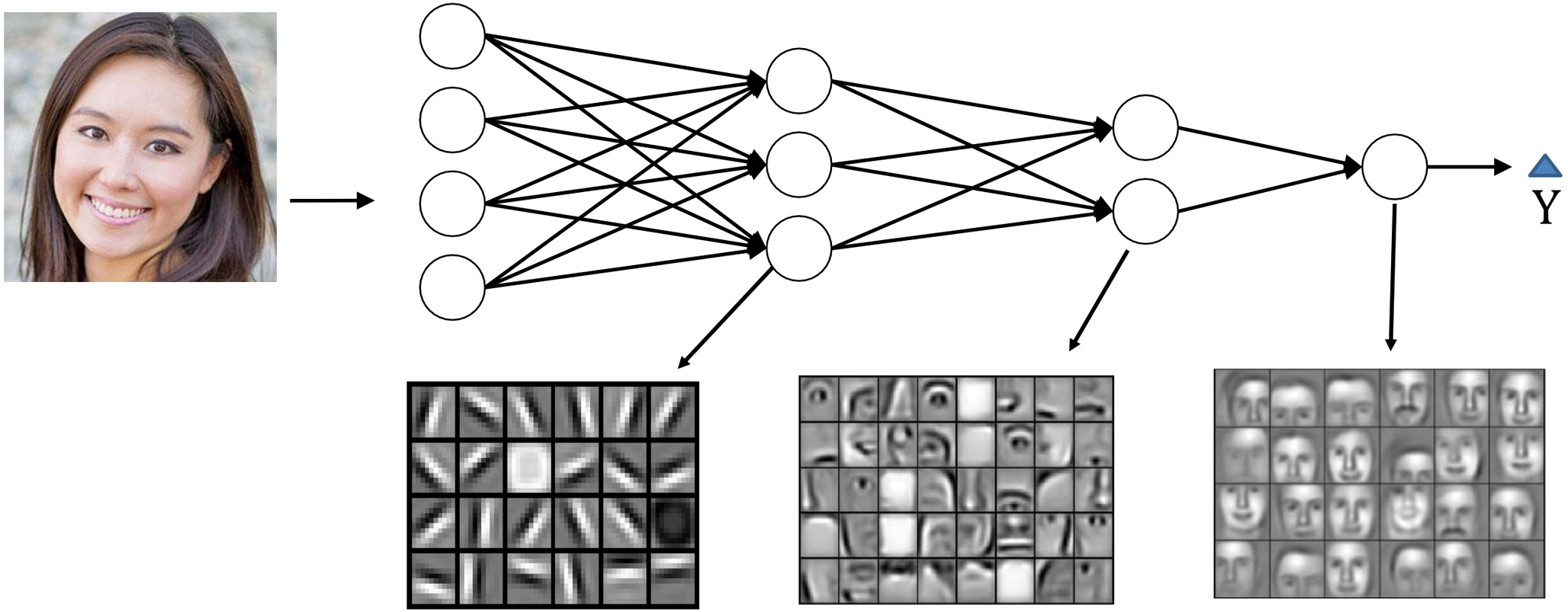
Why deep representations?

Intuition about deep representation

- In deep neural networks, we have a large number of hidden layers. What are these hidden layers actually doing? To understand this, consider the below image:

- Deep neural networks find relations with the data (simpler to complex relations). What the first hidden layer might be doing, is trying to find simple functions like identifying the edges in the above image. And as we go deeper into the network, these simple functions combine together to form more complex functions like identifying the face. Some of the common examples of leveraging a deep neural network are:
- Face Recognition
  - Image ==> Edges ==> Face parts ==> Faces ==> desired face
- Audio recognition
- Audio ==> Low level sound features like (sss, bb) ==> Phonemes ==> Words ==> Sentences

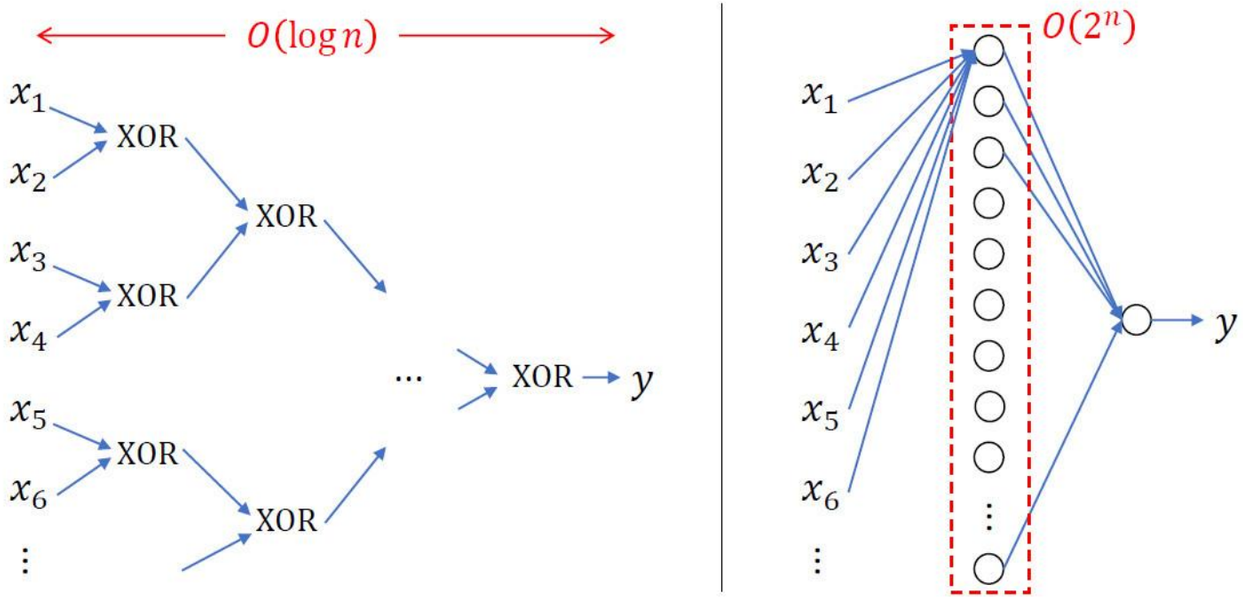
Intuition about deep representation



Circuit theory and deep learning

- Informally: There are functions you can compute with a "small" L-layer deep neural network that shallower networks require exponentially more hidden units to compute.
  - For example, computing the exclusive OR function requires exponentially more hidden units in a shallow network than in a deep network. This result highlights the advantages of having a deep architecture in neural networks.

$$y = x_1 \text{ XOR } x_2 \text{ XOR } x_3 \text{ XOR } \ldots \text{ XOR } x_n$$



Circuit theory and deep learning

- In practice, it is important to find the right depth for a neural network. Starting with a simple model like logistic regression and gradually adding hidden layers can help find the optimal depth for a given problem.
- However, for some applications, very deep neural networks with many dozens of layers can be the best model.
- To implement deep neural networks, both forward and backward propagation need to be implemented.

FPT UNIVERSITY

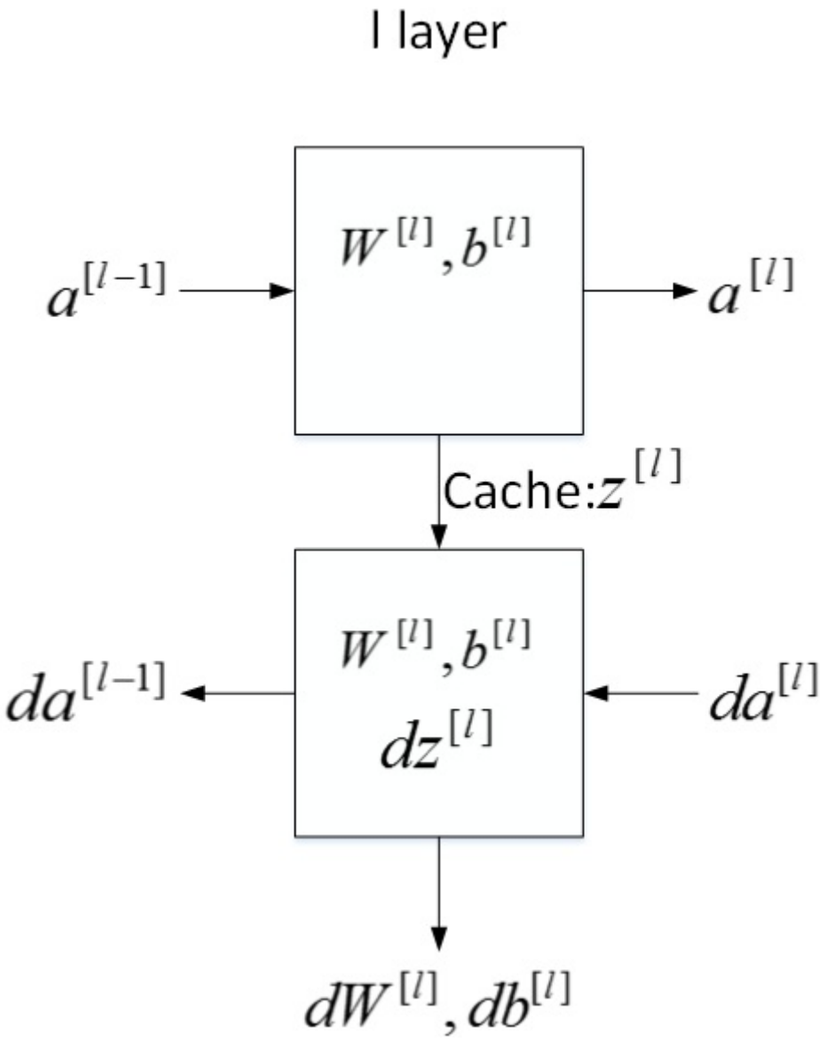
Deep Neural Networks

Building blocks of deep neural networks

Forward and backward functions

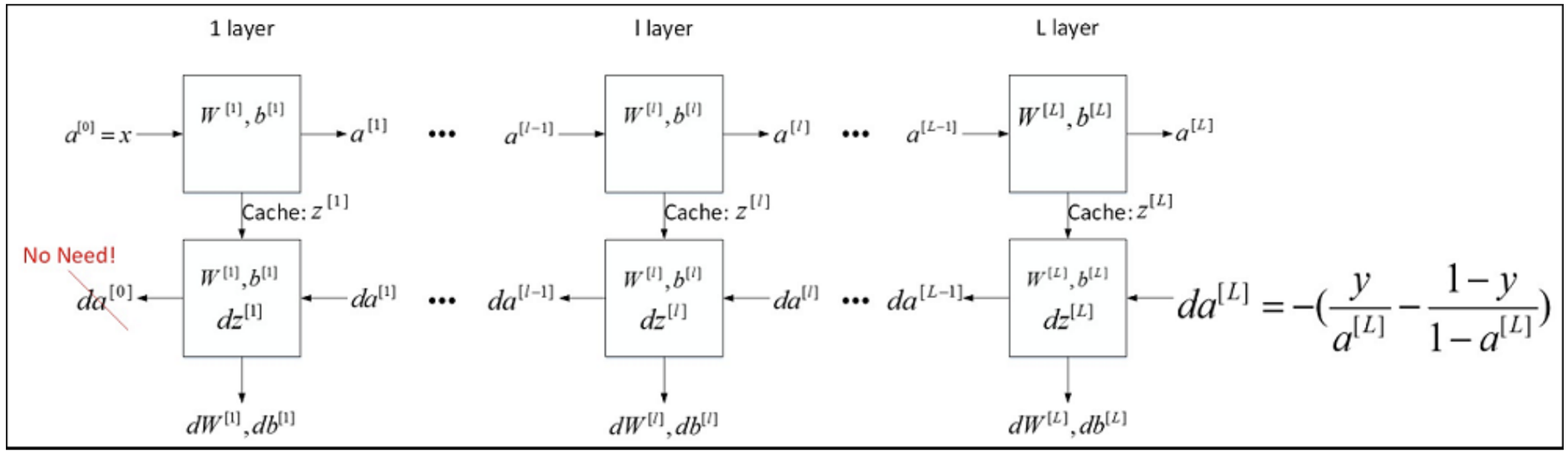
- Consider any layer in a deep neural network. The input to this layer will be the activations from the previous layer (l-1), and the output of this layer will be its own activations.
- **Input**: $a^{[l-1]}$
- **Output**: $a^{[l]}$
- This layer first calculates the $z^{[l]}$ on which the activations are applied. This $z^{[l]}$ is saved as cache.
- For the backward propagation step, it will first calculate $da^{[l]}$, i.e., derivative of the activation at layer l, derivative of weights $dw^{[l]}$, $db^{[l]}$, $dz^{[l]}$, and finally $da^{[l-1]}$. Let's visualize these steps to reduce the complexity:

Forward and backward functions



Forward and backward functions

- The overall process of training the neural network involves feeding in the input features (a[0]), using forward propagation to compute the output (ŷ), and then using back propagation to compute the derivatives with respect to the parameters and activations of each layer.
- The cache stores not only the value of z for the backward function but also the parameters (w and b) for convenience in implementation.



FPT Education

FPT UNIVERSITY

Deep Neural Networks

Forward and backward propagation

Forward propagation for layer l

- Input: $a^{[l-1]}$
- Outputs: $a^{[l]}$ and cache $z^{[l]}$, which is a function of $w^{[l]}$ and $b^{[l]}$.
- So, the vectorized form to calculate $Z^{[l]}$ and $A^{[l]}$ is:

$$Z^{[l]} = W^{[l]} * A^{[l-1]} + b^{[l]}$$

$$A^{[l]} = g^{[l]}(Z^{[l]})$$

- We will calculate Z and A for each layer of the network. After calculating the activations, the next step is **backward propagation**, where we update the weights using the derivatives.

Backward propagation for layer l

- Input: $da^{[l]}$
- Output: $da^{[l-1]}$, $dW^{[l]}$ and $db^{[l]}$.
- Let's look at the vectorized equations for backward propagation:

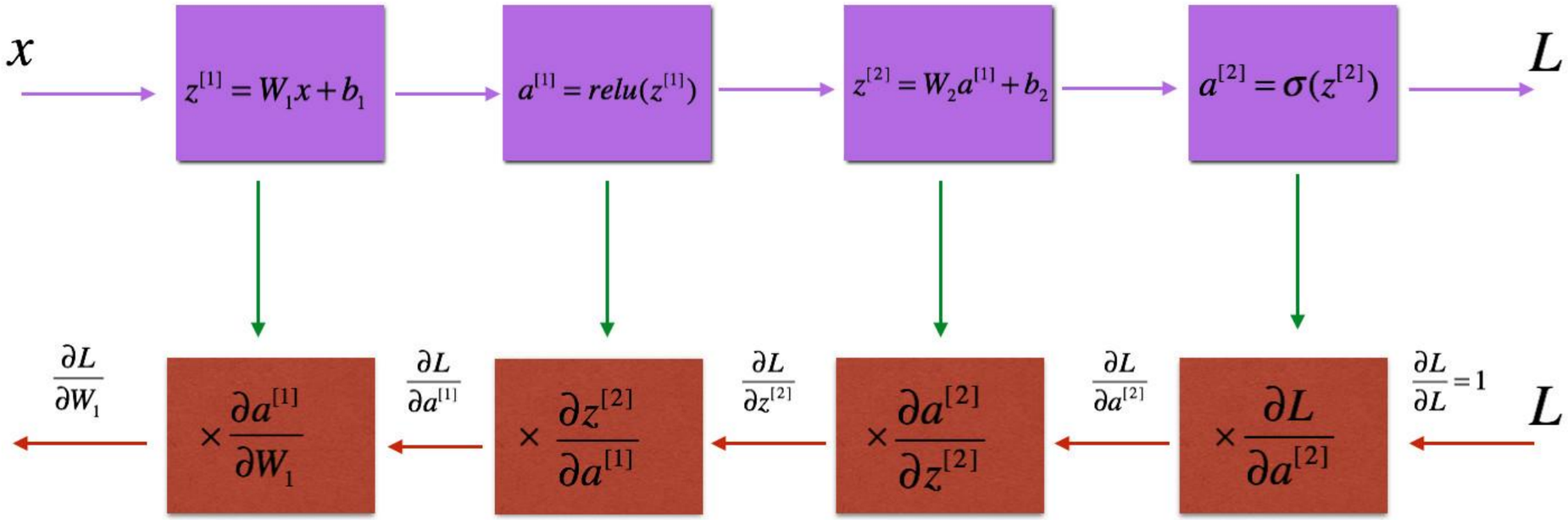
$$dZ^{[l]} = dA^{[l]} * g'^{[l]}(Z^{[l]})$$
$$dW^{[l]} = 1/m * (dZ^{[l]} * A^{[l-1]}.T)$$
$$db^{[l]} = 1/m * np.sum(dZ^{[l]}, axis = 1, keepdims = True)$$
$$dA^{[l-1]} = w^{[l]}.T * dZ^{[l]}$$

Forward and backward propagation

FPT Education
FPT UNIVERSITY

- Deep Neural Networks perform surprisingly well (maybe not so surprising if you've used them before!). Running only a few lines of code gives us satisfactory results. This is because we are feeding a large amount of data to the network and it is learning from that data using the hidden layers.
- Choosing the right hyperparameters helps us to make our model more efficient. We will cover the details of hyperparameter tuning in the next article of this series.

Summary



$$\frac{\partial L}{\partial W_1} = \frac{\partial L}{\partial a^{[2]}} \frac{\partial a^{[2]}}{\partial z^{[2]}} \frac{\partial z^{[2]}}{\partial a^{[1]}} \frac{\partial a^{[1]}}{\partial W_1}$$

FPT UNIVERSITY

Parameters vs Hyperparameters

What are hyperparameters?

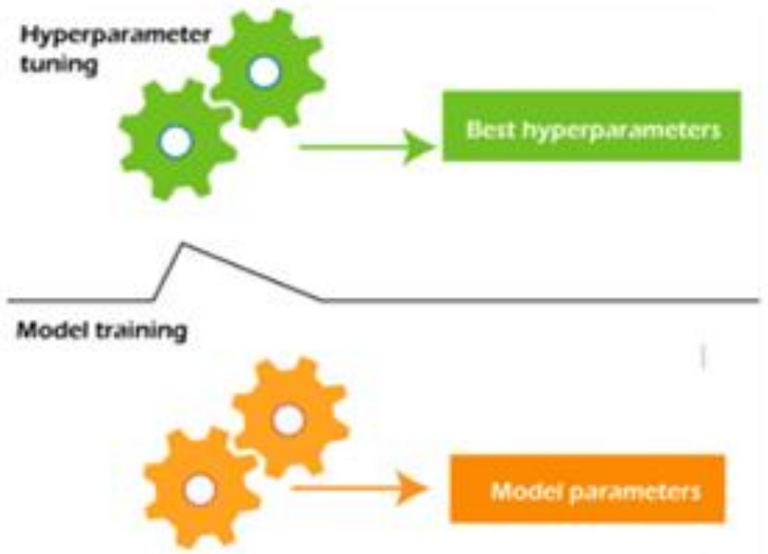
- Hyperparameters are parameters that are not learned by the neural network itself, but rather are set by the programmer or data scientist during the training process.
- These parameters control the behavior of the training algorithm, such as
  - Learning rate – α
  - Number of iterations
  - Number of hidden layers
  - Units in each hidden layer
  - Choice of activation function

What are hyperparameters?

- They ultimately determine the final value of the learned parameters W and B of the neural network.
- Parameters of a deep neural network are W and b, which the model updates during the backpropagation step.
- The major difference between parameters and hyperparameters is that parameters are learned by the model during the training time, while hyperparameters can be changed before training the model.

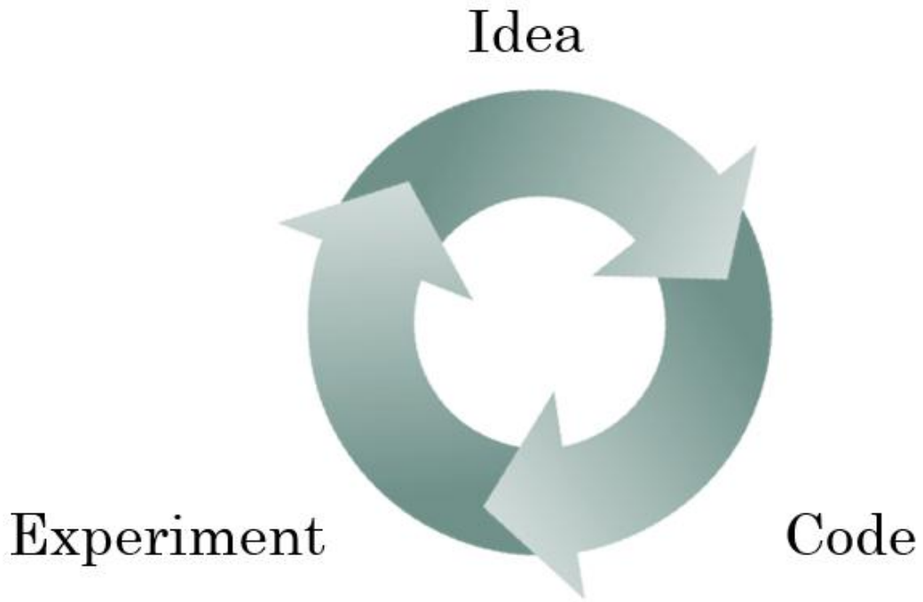
What are hyperparameters

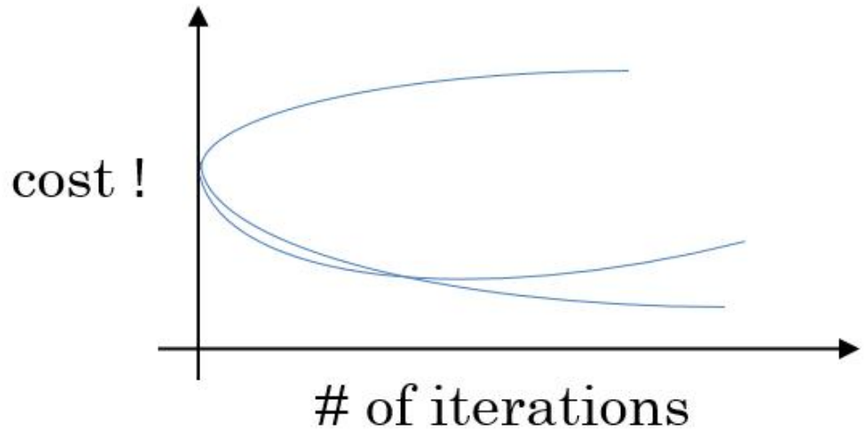
- Some examples of Hyperparameters in Machine Learning
  - The k in KNN or K-Nearest Neighbour algorithm.
  - Learning rate for training a neural network.
  - Train-test split ratio.
  - Batch Size.
  - Number of Epochs.
  - Branches in Decision Tree.
  - Number of clusters in Clustering Algorithm.



Applied deep learning is a very empirical process

- In deep learning, there are many hyper parameters that need to be set, and the best values for these hyper parameters are often not known in advance.
- Therefore, it is an empirical process to try out different values and see what works best for a particular application. This can involve trying out many different values and iterating until the best set of hyper parameters is found.





Applied deep learning is a very empirical process

- Additionally, the best set of hyper parameters for a particular application can change over time, so it is important to periodically re-evaluate and adjust these parameters as necessary.
- While there are some systematic ways to explore the space of hyper parameters, deep learning research is still advancing in this area and better guidance for the best hyper parameters to use may emerge over time.

FPT UNIVERSITY

What does this have to do with the brain?

Forward and backward propagation

- While neural networks do have some loose similarities to biological neurons, this analogy is oversimplified and not entirely accurate.
- Even neuroscientists do not fully understand the workings of a single neuron, and it is unknown whether the human brain uses an algorithm like backpropagation or gradient descent to learn.
- Therefore, deep learning is focused on learning complex input-output mappings, and the analogy to the brain is breaking down as the field progresses.

Forward and backward propagation

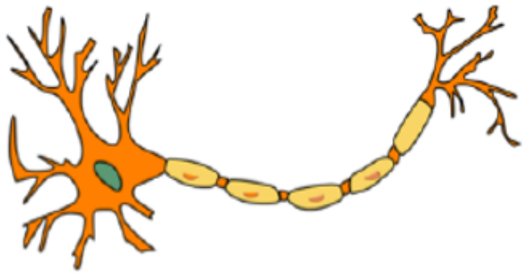
FPT Education
FPT UNIVERSITY

Forward propagation

$$Z^{[1]} = W^{[1]}X + b^{[1]}$$
$$A^{[1]} = g^{[1]}(Z^{[1]})$$
$$Z^{[2]} = W^{[2]}A^{[1]} + b^{[2]}$$
$$A^{[2]} = g^{[2]}(Z^{[2]})$$
$$\vdots$$
$$A^{[L]} = g^{[L]}(Z^{[L]}) = \hat{Y}$$

Backward propagation

$$dZ^{[L]} = A^{[L]} - Y$$
$$dW^{[L]} = \frac{1}{m} dZ^{[L]} A^{[L]^T}$$
$$db^{[L]} = \frac{1}{m} np.\text{sum}(\text{d}Z^{[L]}, axis = 1, keepdims = True)$$
$$dZ^{[L-1]} = dW^{[L]^T} dZ^{[L]} g'^{[L]}(Z^{[L-1]})$$
$$\vdots$$
$$dZ^{[1]} = dW^{[L]^T} dZ^{[2]} g'^{[1]}(Z^{[1]})$$
$$dW^{[1]} = \frac{1}{m} dZ^{[1]} A^{[1]^T}$$
$$db^{[1]} = \frac{1}{m} np.\text{sum}(\text{d}Z^{[1]}, axis = 1, keepdims = True)$$



Summarization

FPT Education
FPT UNIVERSITY

- Deep neural networks have multiple layers of interconnected neurons (L). Information flows from input through hidden layers to the output layer.
- Forward propagation computes predictions by activating neurons using learned parameters.
- Ensuring correct matrix dimensions is crucial for efficient computations.
- Deep networks learn complex representations from raw data, capturing intricate patterns.
- Fundamental components include neurons, layers, activation functions, and parameters.
- Forward propagation predicts, while backward propagation updates parameters during training.
- Parameters (weights and biases) are learned, while hyperparameters (learning rate, layers) control learning and are set before training.
- Deep neural networks draw inspiration from the structure and functioning of the human brain, mimicking synapses and adaptation.

Questions

1. What is the "cache" used for in forward and backward propagation?
2. Which of these are parameters, hyperparameters?
3. True or False: Deeper layers typically compute more complex features than earlier layers.
4. True or False: Vectorization eliminates the layer loop in forward propagation.
5. For a neural network with one input, two hidden, and one output layer, how many total layers (L) and hidden layers are there?
6. True or False: Backpropagation needs the activation function used in the corresponding forward propagation step.
7. True or False: Deep networks can compute some functions with exponentially smaller circuits than shallow networks.
8. For a 2 hidden layer neural network, which of these are the correct shapes? (Select all that apply).