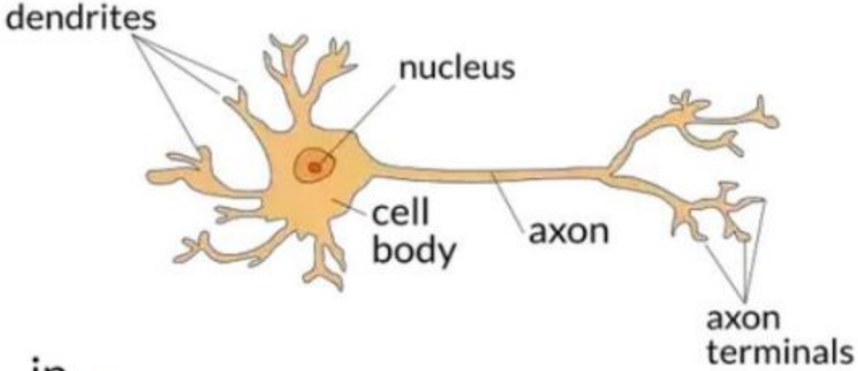
This is CS4045!

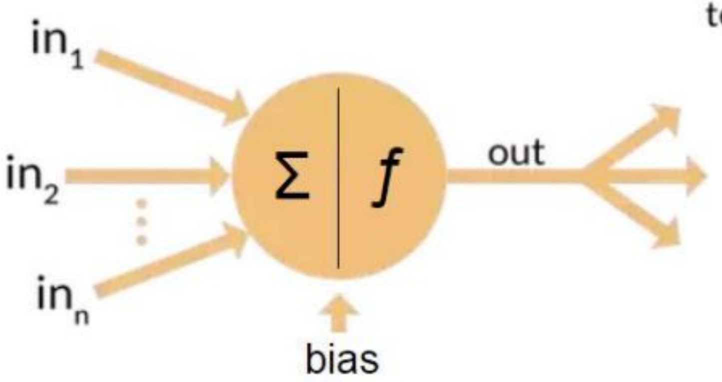
GCR : dxuxugo

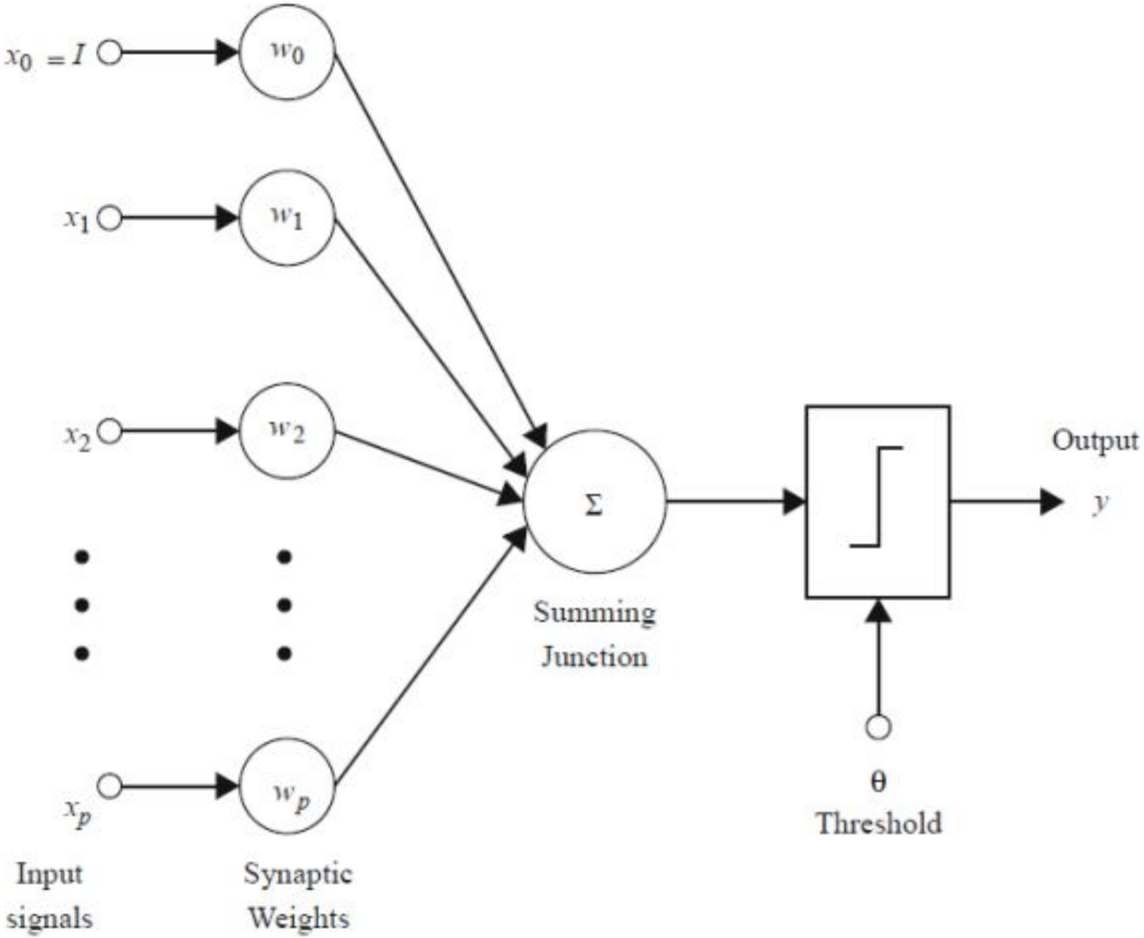
P.S. These slides are useless if you do not attend classes

Neural Networks

Structure of Neuron / Perceptron







x0 = I
x1
x2
xp
w0
w1
w2
wp
Σ
Output
y
Summing
Junction
θ
Threshold
Input
signals
Synaptic
Weights

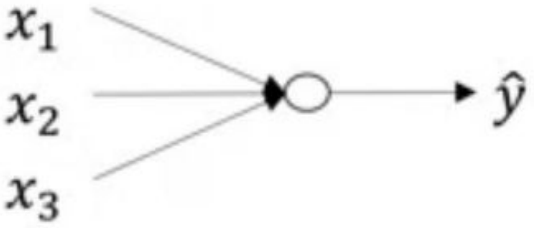
Artificial Neural Network

A mathematical model of the neuron, is called the perceptron

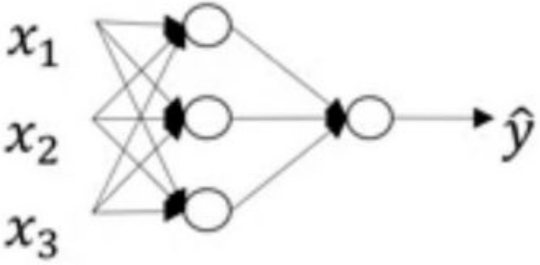
Try and mimic our understanding of the functioning of the brain, in particular its parallel processing characteristics, in order to emulate some of its pattern recognition capabilities

An artificial neural network is a parallel system, which is capable of resolving paradigms that linear computing cannot resolve

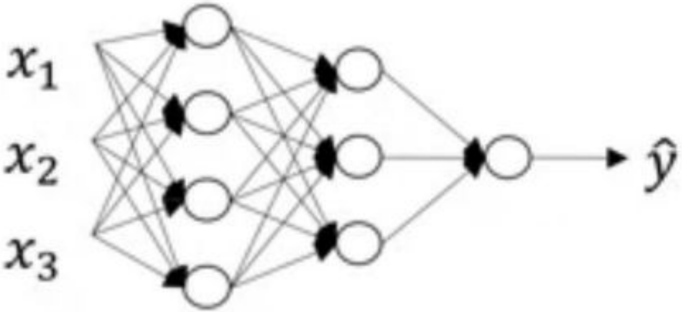
Like its biological predecessor, an ANN is an adaptive system, i.e.,parameters can be changed during operation (training) to suit the problem



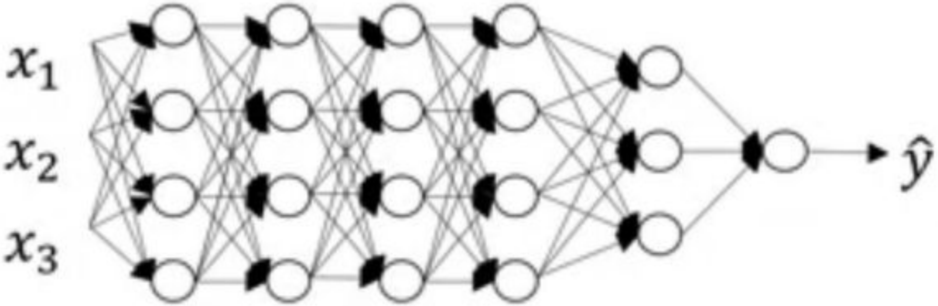
logistic regression



1 hidden layer



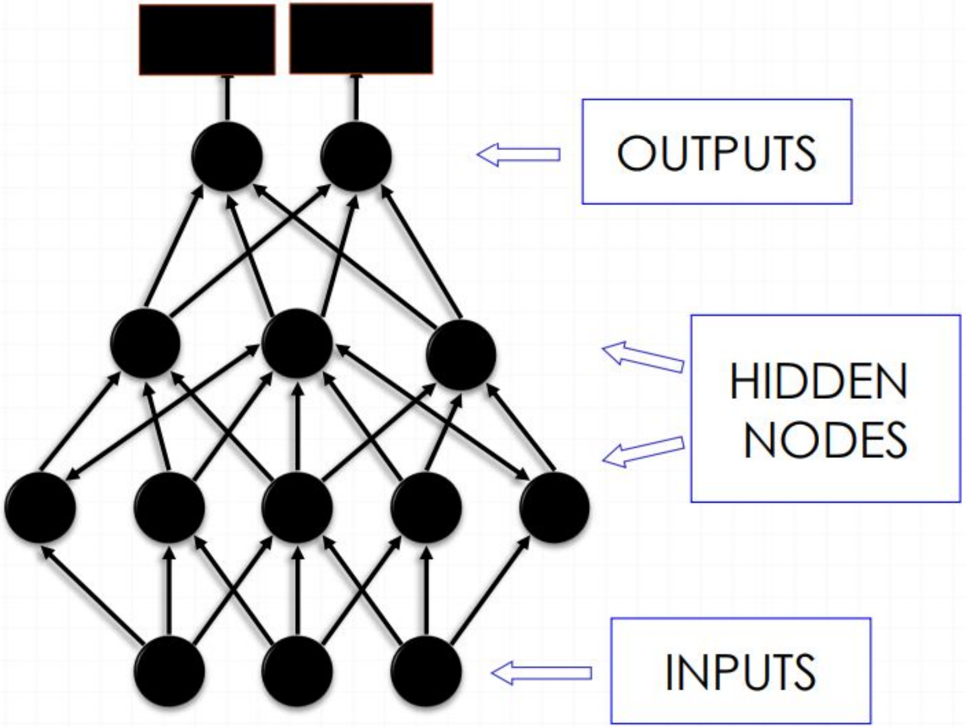
2 hidden layers



5 hidden layers

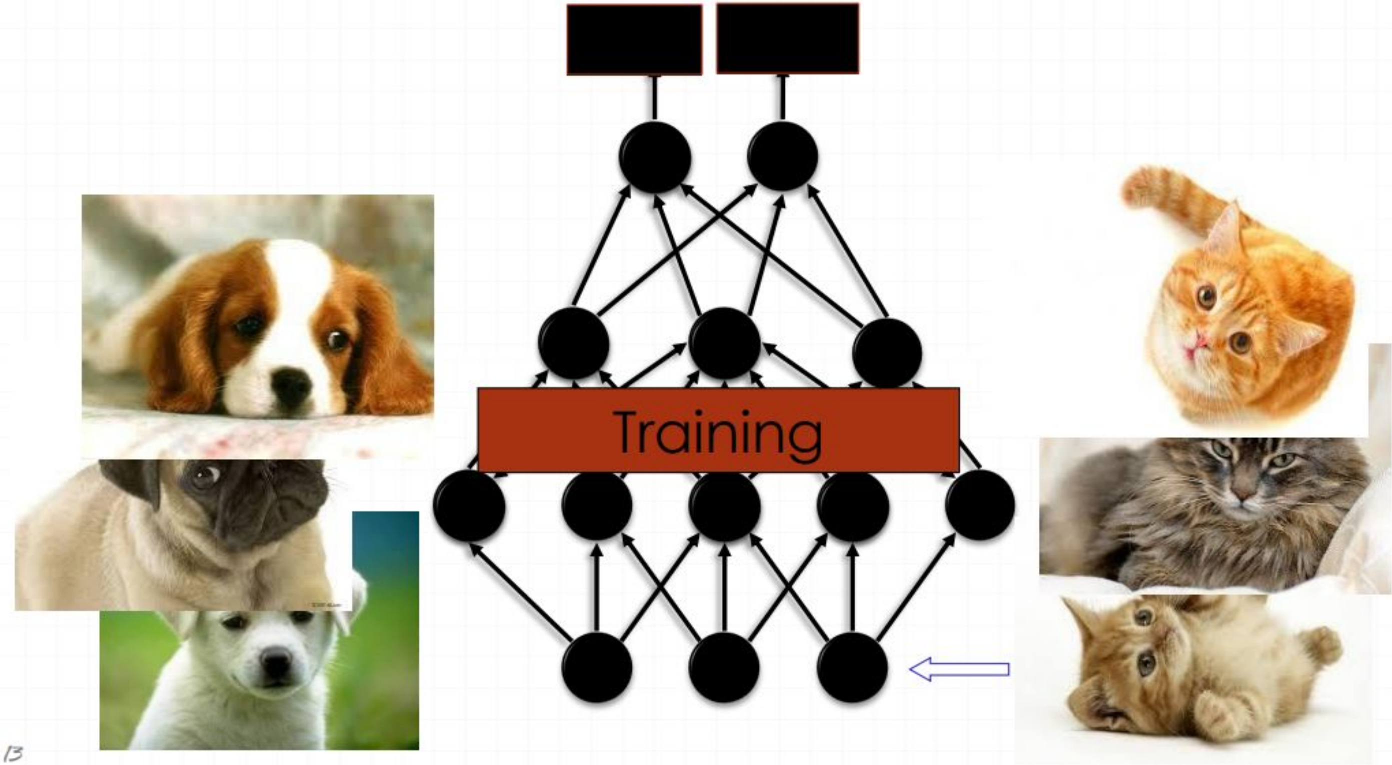
Deep Learning – is a set of machine learning algorithms based on multi-layer networks

Deep Learning Basics



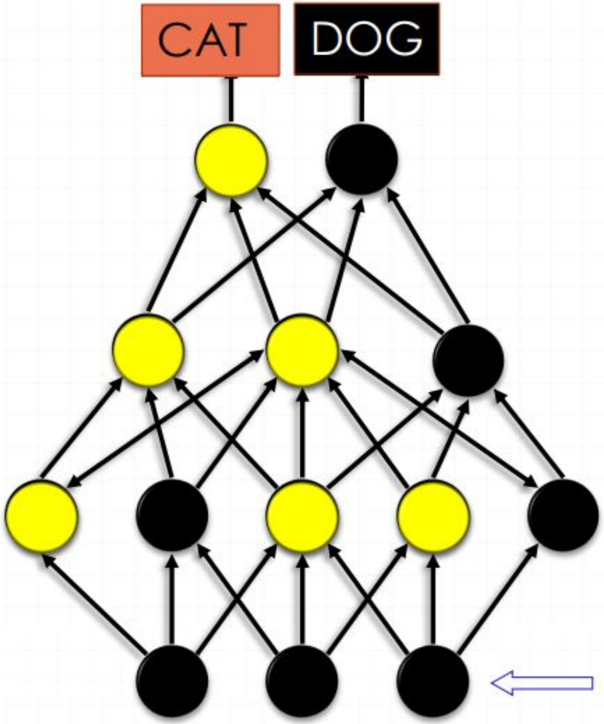
Deep Learning Basics

Deep Learning – is a set of machine learning algorithms based on multi-layer networks



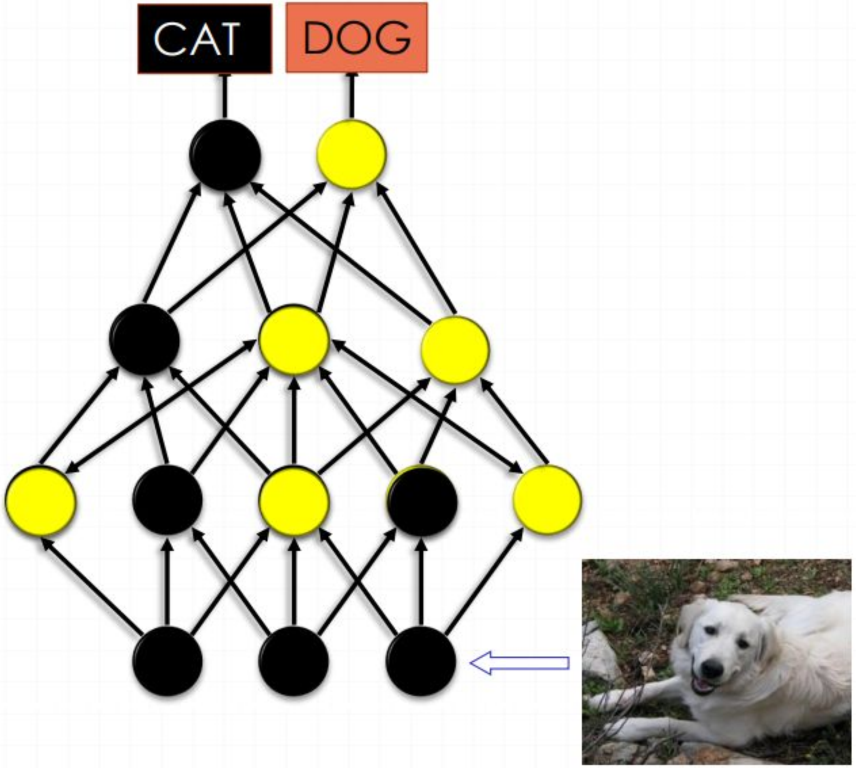
Deep Learning Basics

Deep Learning – is a set of machine learning algorithms based on multi-layer networks



Deep Learning Basics

Deep Learning – is a set of machine learning algorithms based on multi-layer networks

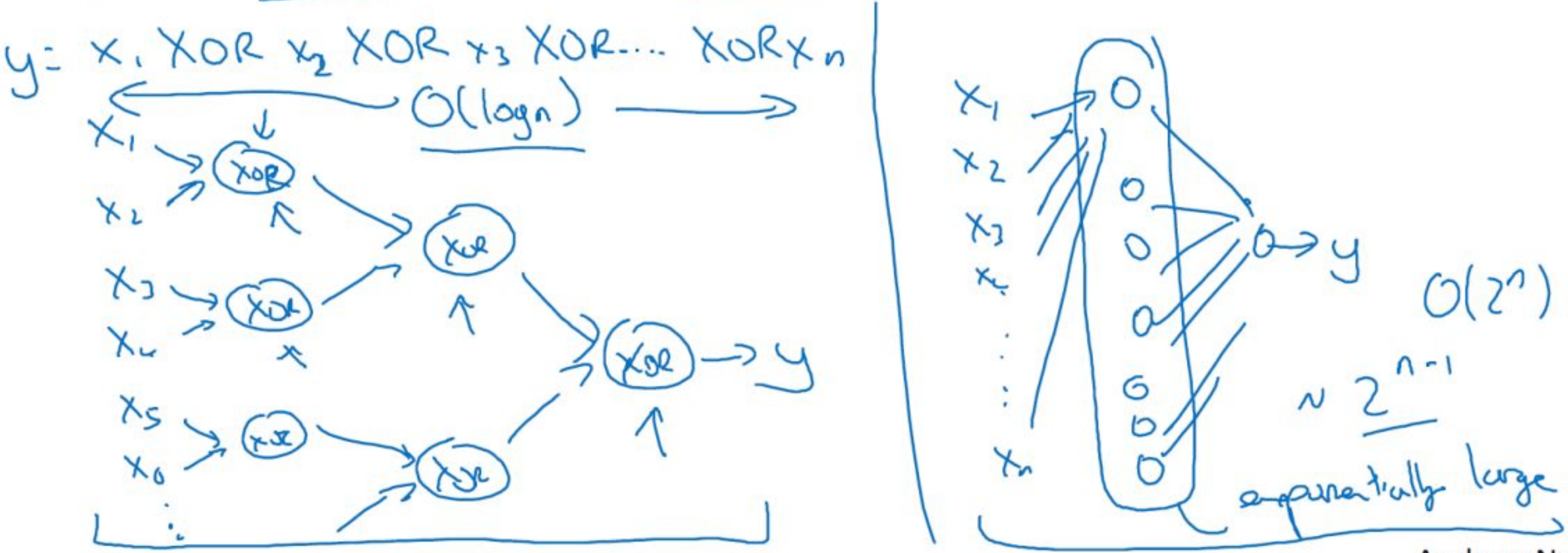


Shallow NN

- Logistic Regression can also be considered as Shallow NN
- Shallow NN only consists of 1 or 2 layer NN
- Shallow NN underfits the data

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.



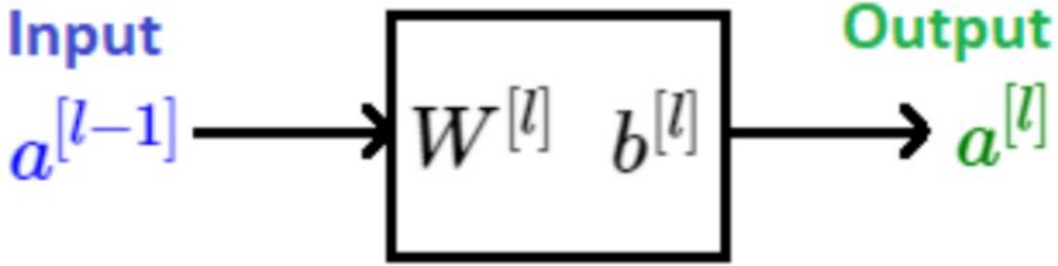


Diagram of a Forward pass through layer l

$$z^{[l]} = \mathbf{W}^{[l]} a^{[l-1]} + b^{[l]}$$

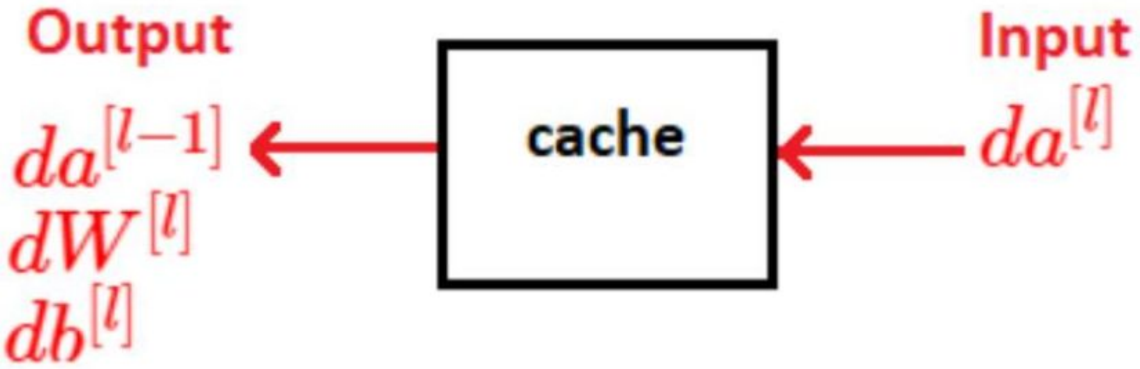
$$a^{[l]} = g(z^{[l]})$$

where $g(z^{[l]})$ is an activation function in the layer l.

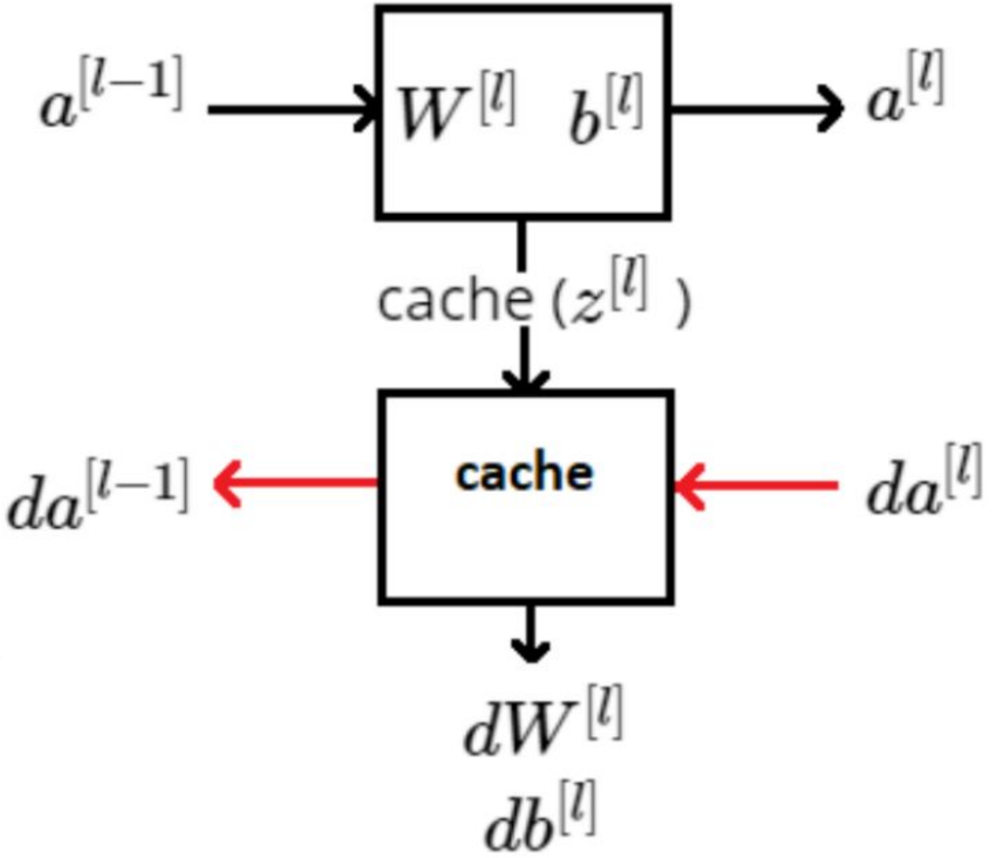
Backward Pass

It is good to cache the value of $z^{[l]}$ for calculations in backwardpass.

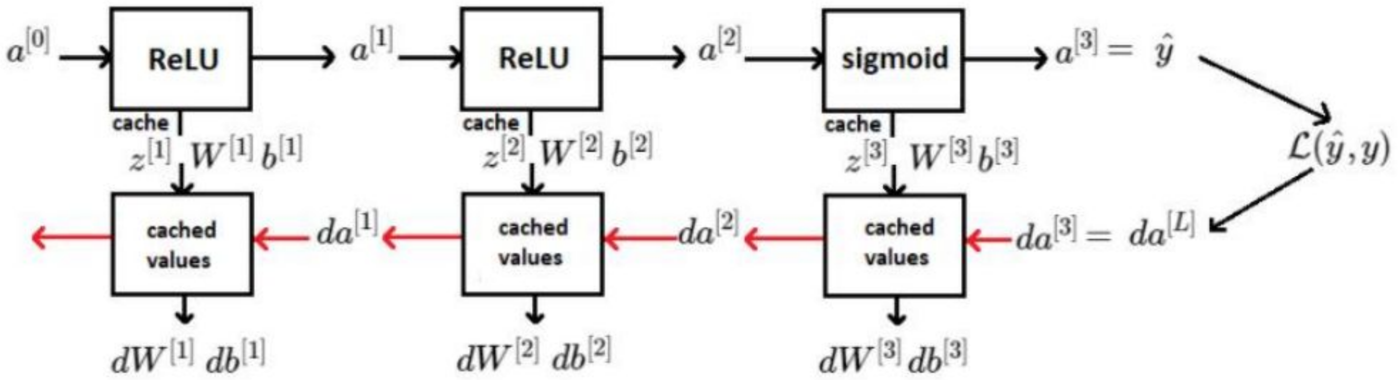
Backward pass is done as we input $da^{[l]}$ and we get the output $da^{[l-1]}$, as presented in the following graph. We will always draw backward passes in red.



Forward and Backward Pass of Layer L



Forward and Backward Pass of Layer L

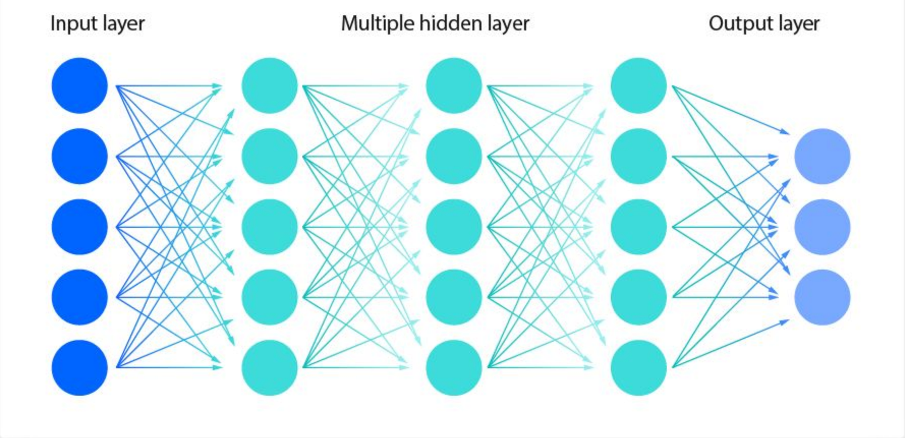


$$\mathbf{W}^{[l]} := \mathbf{W}^{[l]} - \alpha \mathbf{dW}^{[l]}$$

$$b^{[l]} := b^{[l]} - \alpha db^{[l]}$$

$$da^{[L]} = -\frac{y}{a} + \frac{1-y}{1-a}$$

Deep Learning Terms



- Input Layer
- Output Layer
- Hidden Layer
- Dense Layer
- Neurons/ Nodes
- Shallow Neural Network
- Deep Neural Network
- Epoch
- Hyperparameter vs Parameters
- Activation Function

Multilayer backpropogation

Algorithm: Backpropagation. Neural network learning for classification or numeric prediction, using the backpropagation algorithm.

Input:

- D, a data set consisting of the training tuples and their associated target values;
- l, the learning rate;
- *network*, a multilayer feed-forward network.

Output: A trained neural network.

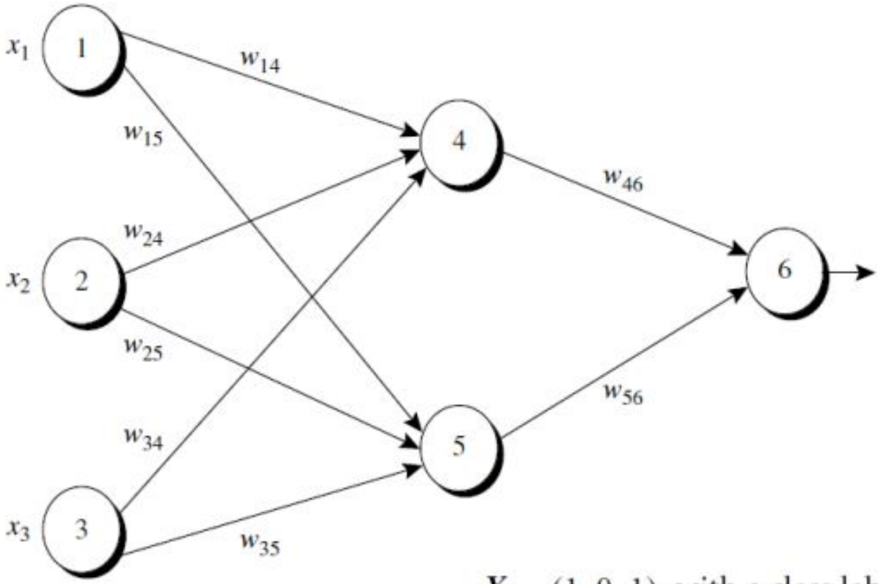
Weights initialization

- **Initialize the weights:** The weights in the network are initialized to small random numbers (e.g., ranging from -1.0 to 1.0, or -0.5 to 0.5). Each unit has a *bias associated with*. The biases are similarly initialized to small random numbers

Convergence condition

- **Terminating condition:** Training stops when
- All Δwij *in the previous epoch are so small as to be below some specified* threshold, or
- The percentage of tuples misclassified in the previous epoch is below some threshold, or
- A prespecified number of epochs has expired.

Worked example



$\boldsymbol{X} = (1, 0, 1)$, with a class label of 1.

Initial Input, Weight, and Bias Values

x_1	x_2	x_3	w_{14}	w_{15}	w_{24}	w_{25}	w_{34}	w_{35}	w_{46}	w_{56}	θ_4	θ_5	θ_6
1	0	1	0.2	−0.3	0.4	0.1	−0.5	0.2	−0.3	−0.2	−0.4	0.2	0.1

Net Input and Output Calculations

Unit, j	*Net Input, I_j*	*Output, O_j*
4	$0.2+0-0.5-0.4=-0.7$	$1/(1+e^{0.7})=0.332$
5	$-0.3+0+0.2+0.2=0.1$	$1/(1+e^{-0.1})=0.525$
6	$(-0.3)(0.332)-(0.2)(0.525)+0.1=-0.105$	$1/(1+e^{0.105})=0.474$

Calculation of the Error at Each Node

Unit, j	*Err_j*
6	$(0.474)(1-0.474)(1-0.474)=0.1311$
5	$(0.525)(1-0.525)(0.1311)(-0.2)=-0.0065$
4	$(0.332)(1-0.332)(0.1311)(-0.3)=-0.0087$

BackPropagation

Calculations for Weight and Bias Updating

Weight or Bias	*New Value*
w_{46}	$-0.3 + (0.9)(0.1311)(0.332) = -0.261$
w_{56}	$-0.2 + (0.9)(0.1311)(0.525) = -0.138$
w_{14}	$0.2 + (0.9)(-0.0087)(1) = 0.192$
w_{15}	$-0.3 + (0.9)(-0.0065)(1) = -0.306$
w_{24}	$0.4 + (0.9)(-0.0087)(0) = 0.4$
w_{25}	$0.1 + (0.9)(-0.0065)(0) = 0.1$
w_{34}	$-0.5 + (0.9)(-0.0087)(1) = -0.508$
w_{35}	$0.2 + (0.9)(-0.0065)(1) = 0.194$
θ_6	$0.1 + (0.9)(0.1311) = 0.218$
θ_5	$0.2 + (0.9)(-0.0065) = 0.194$
θ_4	$-0.4 + (0.9)(-0.0087) = -0.408$

Classification of unknown datapoint

- To classify an unknown tuple, X, the tuple is input to the trained network, and the net input and output of each unit are computed. (There is no need for computation and/or backpropagation of the error)
- If there is one output node per class, then the output node with the highest value determines the predicted class label for X
- If there is only one output node, then output values greater than or equal to 0.5 may be considered as belonging to the positive class, while values less than 0.5 may be considered negative

Critique of ANN

- Neural networks involve long training times and are therefore more suitable for applications where this is feasible.
- They require a number of parameters that are typically best determined empirically such as the network topology or "structure." Neural networks have been criticized for their poor interpretability
- For example, it is difficult for humans to interpret the symbolic meaning behind the learned weights and of "hidden units" in the network. These features initially made neural networks less desirable for data mining

Advantage

- Advantages of neural networks, however, include their high tolerance of noisy data as well as their ability to classify patterns on which they have not been trained. They can be used when you may have little knowledge of the relationships between attributes and classes.
- They are well suited for continuous-valued inputs *and outputs, unlike most* decision tree algorithms. They have been successful on a wide array of real-world data, including handwritten character recognition, pathology and laboratory medicine, and training a computer to pronounce English text

- Neural network algorithms are inherently parallel; parallelization techniques can be used to speed up the computation process

- It can perform tasks which a linear classifier cannot
- Multilayer feed-forward networks, given enough hidden units and enough training samples, can closely approximate any function