Introduction to Deep Neural Networks

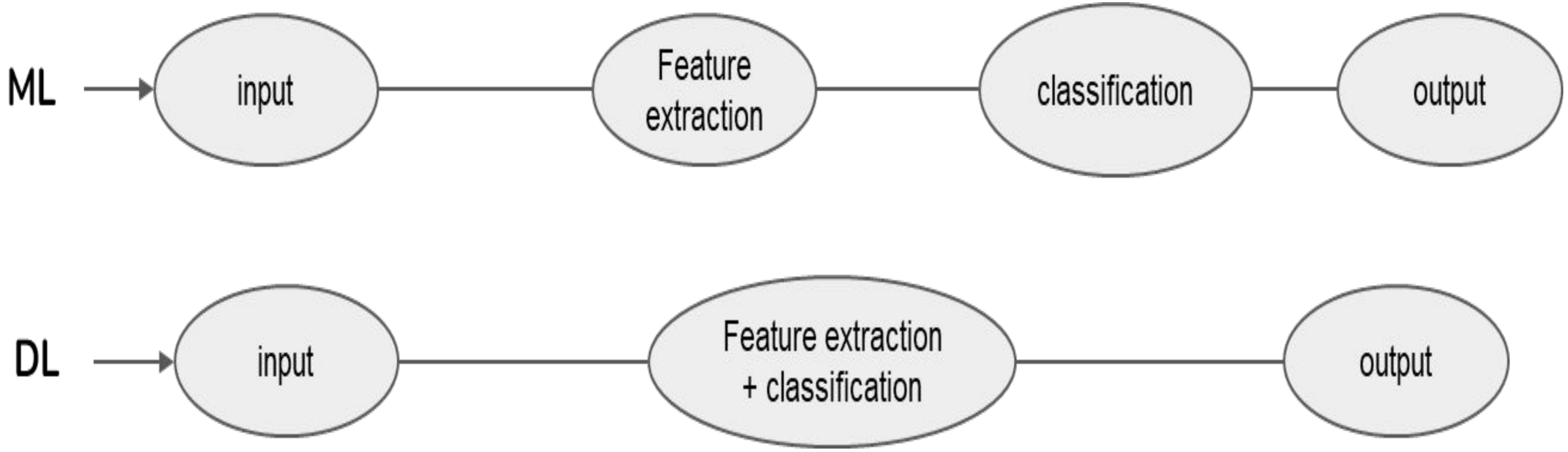
Discussion Question

1. What distinguishes deep learning from machine learning?
2. What are the differences between McCulloch Pitts Neurons and Perceptron?
3. What is an activation function and what are its different types?
4. Which activation functions are used in hidden and output layer?
5. How is forward and Backward propagation performed?

What distinguishes deep learning from machine learning?

Machine Learning(ML)	Deep Learning (DL)
It can easily train on lesser data and requires lesser time to train	It requires large data(Not necessarily) and longer time to train
It trains on CPU	It trains on GPU
Some ML algorithms can be easily interpreted	Difficult to interpret.

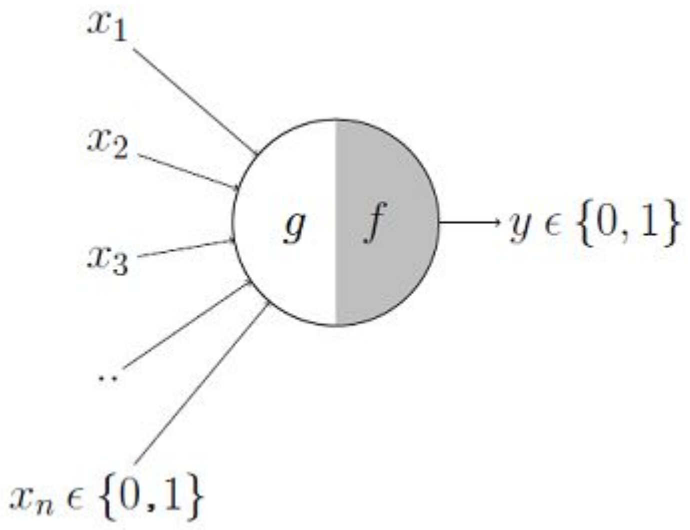
How is deep learning is different from machine learning? (Contd)



What are the differences between McCulloch Pitts Neurons and Perceptron?

Warren McCulloch & Walter Pitts Neuron

1. It has an input layer that acts like dendrites.
2. It has two parts the first part, g the weighted addition of inputs. The weights are manually initialized, and all have the same weight.
3. The weighted sum is passed through the second part an activation function f which yields a 1 if the threshold is crossed else 0.



$$g(x_1, x_2, x_3, ..., x_n) = g(\mathbf{x}) = \sum_{i=1}^{n} x_i$$

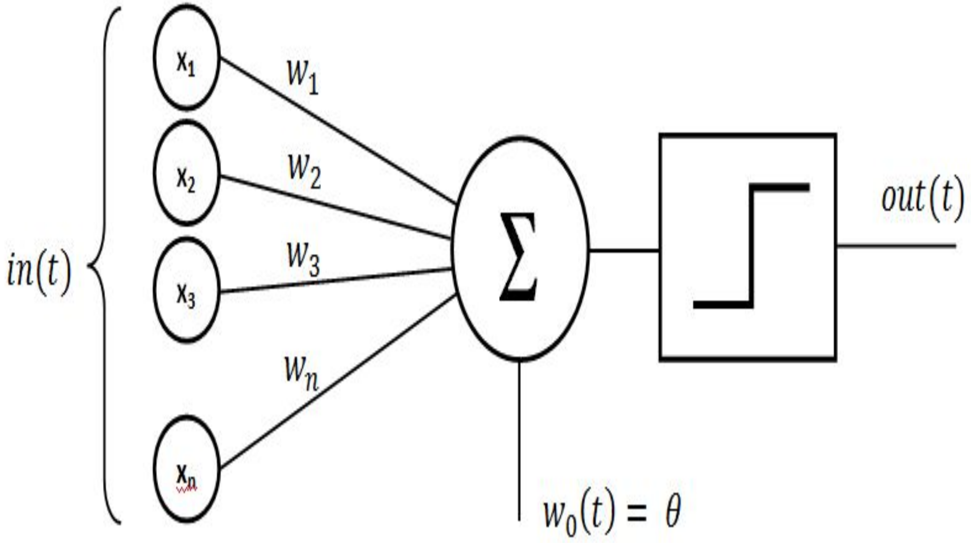
$$\begin{aligned} y = f(g(\mathbf{x})) &= 1 \quad if \quad g(\mathbf{x}) \geq \theta \\ &= 0 \quad if \quad g(\mathbf{x}) < \theta \end{aligned}$$

Ref: https://towardsdatascience.com/mcculloch-pitts-model-5fdf65ac5dd1

What are the differences between McCulloch Pitts Neurons and Perceptron?(Contd)

Rosenblatt's Perceptron

1. It has an Input layer that acts as dendrites.
2. It has two parts, the first part is weighted addition Each input is multiplied with a weight (which is typically initialized with some random value)
3. The sum is then passed through an activation function which yields a 1 if the threshold is crossed.
4. The step function can be defined in such a way that output can range from -1 to +1.
5. It captured the error and had a logic built in to adjust the weights automatically to reduce the error.



Ref:
https://commons.wikimedia.org/wiki/File:Perceptron_moj.png#/media/File:Perceptron_moj.png

What is an activation function and what are its different types?

Activation functions are one of the main components of Neural Networks, which defines the output of input or set of inputs. They basically decide to activate or deactivate neurons to get the desired output. It also performs a nonlinear transformation on the input to get better results on a complex **neural network**.

Types of activation functions are as follows:

1. Step function
2. ReLU - Rectified Linear Unit
3. Leaky ReLU
4. Sigmoid function
5. Tanh function
6. Softmax

Which activation functions are used in hidden and output layer?

Activation functions used in hidden layers:

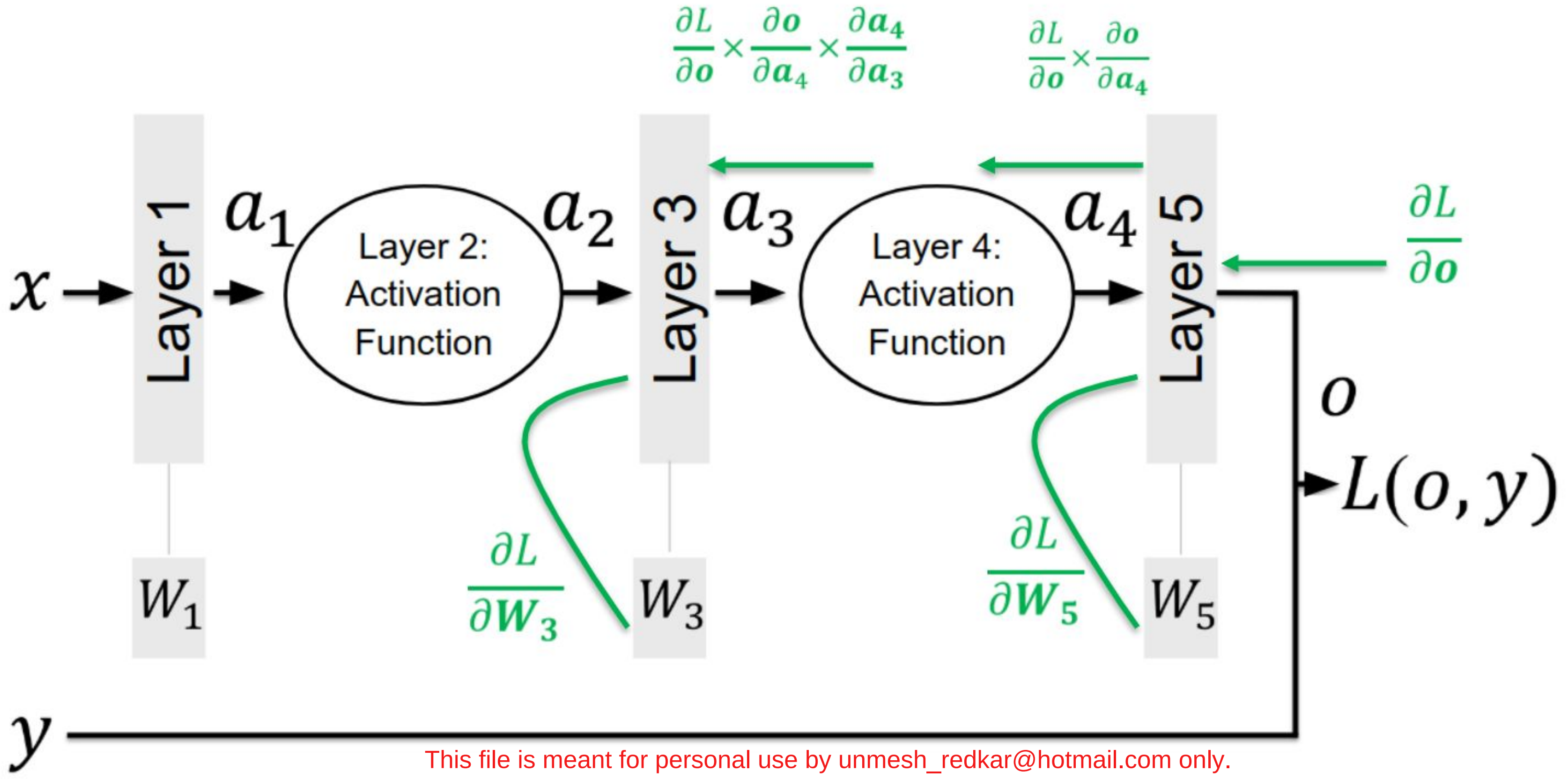
1. Step function
2. Sigmoid
3. Tanh
4. ReLU

Most common activation function used in Hidden layers is ReLU, because it is both simple to implement and effective at overcoming the limitations of other previously popular activation functions, such as Sigmoid and Tanh.

Activation functions used in the output layer:

1. Sigmoid [For binary classification problems with 1 neuron in output layer]
2. Softmax [For multi-class classification problems with number of neuron equal to the class present in the problem
3. Linear [Regression type of problems with 1 neuron in output layer

How is forward and Backward propagation performed?



$\frac{\partial L}{\partial o} \times \frac{\partial o}{\partial a_4} \times \frac{\partial a_4}{\partial a_3}$
$\frac{\partial L}{\partial o} \times \frac{\partial o}{\partial a_4}$
x
Layer 1
a_1
Layer 2: Activation Function
a_2
Layer 3
a_3
Layer 4: Activation Function
a_4
Layer 5
$\frac{\partial L}{\partial o}$
o
$L(o, y)$
W_1
$\frac{\partial L}{\partial W_3}$
W_3
$\frac{\partial L}{\partial W_5}$
W_5
y

How is forward and Backward propagation performed?

Forward Propagation

- Input data is propagated forward from the input layer to the hidden layer till it reaches the final layer where predictions are emitted.
- At every layer, data gets transformed non-linearly in every neuron.
- There may be multiple hidden layers with multiple neurons in each layer.
- The last layer is the output layer which may have a softmax or sigmoid function (if the network is a multi-class or binary classifier respectively)
- **Forward prop steps –**
  a. Calculate the weighted input to the hidden layer by multiplying X by the hidden weight Wi
  b. Apply the activation function and pass the result to the final layer
  c. At output layer, repeat step b replacing X by the hidden layer output

How is forward and Backward propagation performed?

Backward Propagation

- Backpropagation is the process of learning that the neural network employs to re-calibrate the weights and bias at every layer and every node to minimize the error in the output layer.
- During the first pass of forward propagation, the weights and bias are small random numbers.
- The output of the first iteration is almost always incorrect. The difference between actual value/class and predicted value/class is the error.
- All the nodes in all the preceding layers have contributed to the error and hence need to get their share of the error and correct their weights.
- This process of allocating a proportion of the error to all the nodes in the previous layer is backpropagation.
- The goal of backpropagation is to adjust weights and bias in proportion to the error contribution and in the iterative process identify the optimal combination of weights.

How is forward and Backward propagation performed?

- Here the loss is computed at each layer while propagating from backward
- The loss is represented by L(o,y)
- The loss in the output layer is computed using the error loss as dl/do
- The loss in the layer 5 is computed as

$$\frac{\partial L}{\partial W_5} = \frac{\partial L}{\partial o} \times \frac{\partial o}{\partial a_4}$$

- The loss in layer 3 can be computed as

$$\frac{\partial L}{\partial W_3} = \frac{\partial L}{\partial o} \times \frac{\partial o}{\partial a_4} \times \frac{\partial a_4}{\partial a_3}$$

- Thus the loss is calculated at each layer and the weights are updated

greatlearning
Power Ahead

Happy Learning !