Assignment\_2

**Q – 1 Describe the structure of an artificial neuron. How its simpler to biological neuron? What are its main component?**

**Ans –** Artificial neural network (ANN) is a computational model that consists of several processing element that receive inputs and deliver outputs based on their predefine activation functions.

Artificial neuron also known as perceptron is the basic unit of the neural network. In simple terms, It is a mathematical function based on a model of biological neurons. It can also be seen as a simple logic gate with binary outputs. They are sometimes also called perceptron.

The main component is

1. Input = The inputs are simply the measure of our features.
2. Weights = Weights represent scalar multiplication.
3. Transfer function = The transfer function is different from the other components in that it takes multiple inputs.
4. Activation function
5. Bias.

**Q – 2 What are the different types of activation function properly used? Explain each of them.**

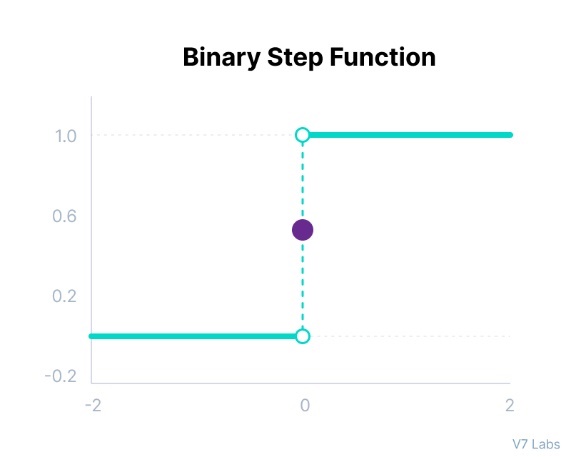
**Ans –** Activation function is means that in a simple words like **it is decides weather a neuron should be active or not.**

There are three types of activation function.

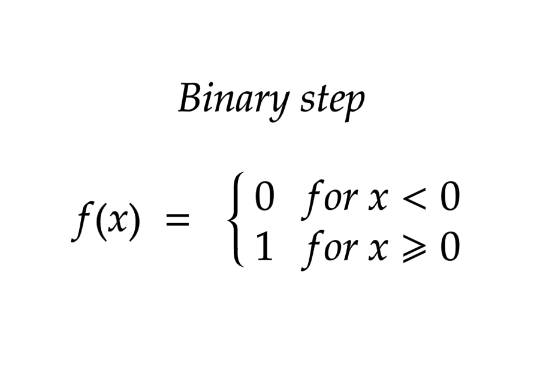
1. Binary step function.
2. Linear activation function.
3. Non linear activation function

**Binary step function =** Binary step function depends on a threshold value that decides weather a neuron should be activation or not.

The input fed to the activation function is compared to a certain threshold if the input is greater tha ot then the neuron is activation, else it is deactivated, meaning that its output is not passed on to the next hidden layer.



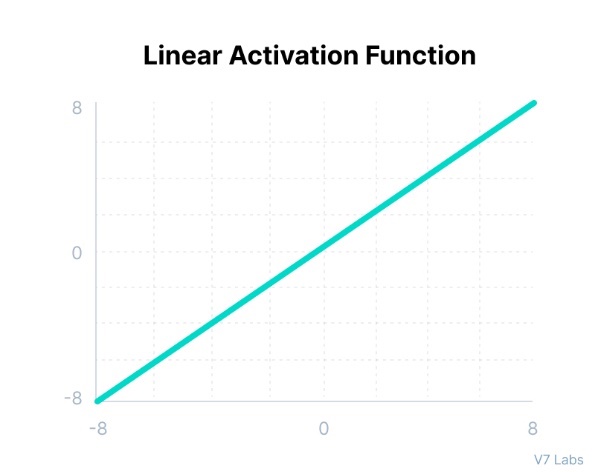
**Mathematical representation is**



**some limitation of binary functions**

1. It cannot provides the multi output function. For example it cannot be used the multi classification problem.
2. The gradient of the step function is zero, which causes the hindrance in the backpropagation**.**

**Linear Activation function =** The linear activation function also known as “no activation,” or “identify function”(multiplied x1.0), is where the activation is proportional to the input. The function doesn’t do anything to the weights sum of the inputs, it simply spits out the value it was given



**Mathematical representation is**

**F(x) = X**

**The two major problem:**

1. Its not possible for backpropagation has the derivatives of the function is a constant and has no relation of the input X.
2. All layers of the neural network will collapse into one if a linear activation function is used. No matter the number of layers in the neural network, the last layer will still be a linear function of the first layer. So essentially a linear activation function turns the neural network into just one layer.

**Non-linear Activation function =** The linear function shown above is simply a linear regression model.

Because of its limited power, this does not allow the model to create complex mappings between the network inputs and output.

**The limitation of non-linear activation function is**

1. They allow backpropagation because now the derivative function would be relate to the input and its possible to go back and understand which weights in the input neurons can provide a better prediction.
2. They allow the stacking of multiple layers of neurons as the output would now be a non-linear combination of input passed through multiple layers. Any output can be represented as a function computation in a neural network.

**Non-linear activation function is –**

1. Sigmoid activation function / logistic activation function.

F(x) = 1/1+e-x

1. Tanh function (hyperbolic activation function)

F(x) = (ex – e-x) / (ex + e-x)

1. Relu activation function

F(x) = max (0, x)

1. Leaky ReLu (derivative)
2. Parametric ReLu

F(x) = max (ax, x)

1. Elu activation function

**X for x>=0**

**a(ex – 1) for x<0**

1. **Softmax (zi) = exp(zi) / summation exp(zi)**

Finally, a few rules for choosing the activation function for your output layer based on the type of prediction problem that you solving.

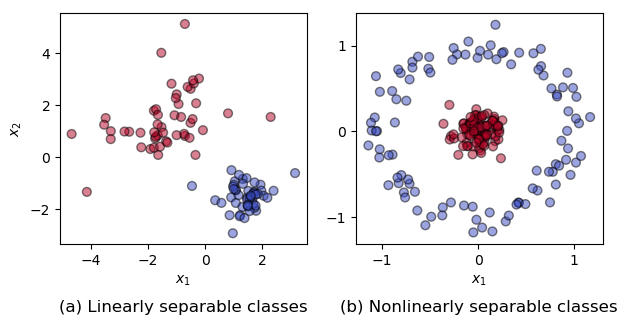
1. **Regression –** linear activation function
2. **Binary classification –** sigmoid / logistic activation function
3. **Multiclass classification –** SoftMax
4. **Multilabel classification –** sigmoid

**Q – 3 Explain in details Rosenblatt’s perceptron model. How can a set of data classified using a perceptron?**

**Ans –** Almost fifteen years after McCulloch & Pitts, the American psychologist Frank Rosenblatt (1928 - 1971), inspired by Hebbian theory of synaptic plasticity (i.e., the adaptation of brain neurons during the learning process), came up with the perceptron, a major important over the MCP neuron model. This invention granted him international recognition and, to this data the institute of Electrical and Electronic Engineers(IEEE), “the world largest innovation and excellence for the benefit of humanity”, named its annual award in his honour.

Rosenblatt’s major achievement has been to show that by relaxing some of MCP’s rule (namely the absolute inhibition, the equal contribution of all inputs as well as their integer nature), artificial neural could actually learn from data. More importantly, he came up with a supervised learning algorithm for his modified MCP neuron model that enable the artificial neuron to figure out the correct weights directly form training data by itself. Before diving into the machine learning fun stuff, let us quickly discuss the type of problem that can be addressed by the perceptron.

Binary classification is the task of classifying the element if a given set into two groups based on the a prescribed rule. The fig below depicts two instances of such a problem. On the left, the task is to identify the separatrix between two linearly separable class while, on the right, the two classes are non linearly separable.



**Q – 4 Use a simple perceptron with weights w 0 , w 1 , and w 2  as −1, 2, and 1, respectively, to classify data points (3, 4); (5, 2); (1, −3); (−8, −3); (−3, 0).**

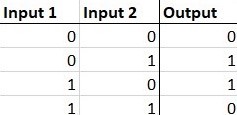
**Ans –** so -1, 2, and 1 are possible values for the weights w0, w1 and w2 respectively. Using this value the output of the perceptron for A = -1 is negative, B=2,

**Q – 5 Explain the basic structure of a multi-layer perceptron. Explain how it can solve the XOR problem.**

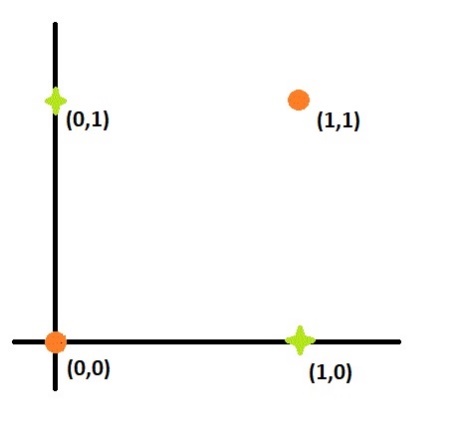
Ans – It consists of three types of layers – the input layer, output layer and hidden layer. The input layer receives the input signal to be processed. The required task such as prediction and classification is performed by the output layer.

**Solving XOR problem using the multi-layer perceptron.**

1. **XOR logic operator =** XOR, or Exclusive or is a binary logical operator that takes in Boolean inputs and given out True if and only if the two inputs are different. This logical operator is especially useful when we want to check two conditions that can’t be simultaneously true. The following is the truth for XOR function.

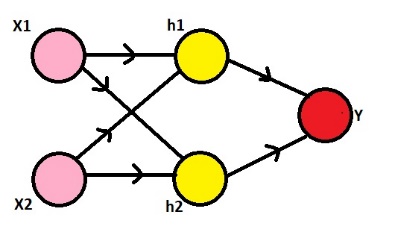


1. **The XOR problem =** The XOR problem is that we need to build a neural network to produce the truth table related to the XOR logical operator. This is a binary classifications problem. Hence supervised learning is better way to solve it. in this case we will be using perceptron. Unlayered perceptron can only work with linearly separable data. But in the following diagram drawn in accordance with truth table of the XOR logical operator, we can see that the data is not linearly separable.

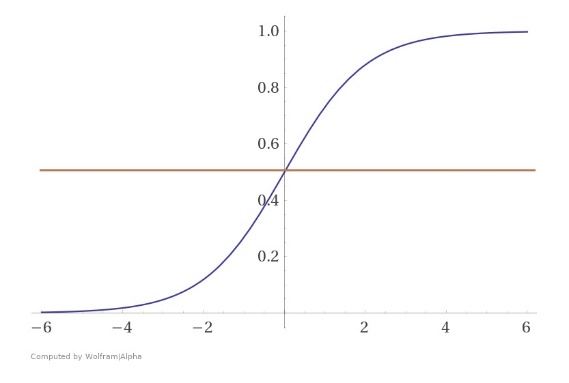


**The solution is =** To solve this problem, we add an extra layer to our vanilla perceptron, i.e,. we create a multilayer perceptron. We call this extra layer as the hidden layer. To build a perceptron, we first need to understand that the XOR gate can be written ad a combination of AND gates, NOT gates and OR gates in the following way:

a **XOR** b = (a **AND NOT** b) **OR** (b **AND NOT** a)



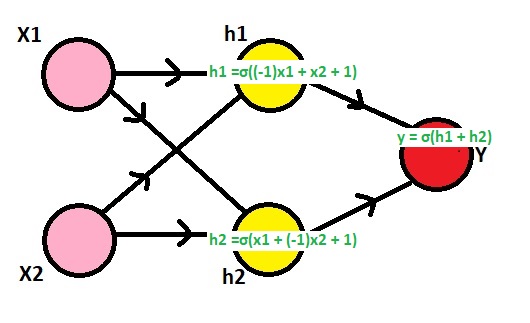
Here, we need to observe that our inputs are 0s and 1s. To make it XOR gate, we will make the h1 node to perform the (x2 AND NOT x1)operation, the h2 node to perform (x1 NAD NOT x2) operation and they node to perform (h1 OR h2) operation. The NOT gate can be produced for an input a by writing (1-a), the AND gate can be produced for inputs a and b by writing (a.b) and the OR gate can be produced fro inputs a and b by writing (a+b). Also, we will use the sigmoid function as pur activation function σ, i.e., σ(x) = 1/(1+e^(-x)) and the threshold for classification would be 0.5, i.e., any x with σ(x)>0.5 will be classified as 1 and others will be classified as 0



Now, since we have all the information, we can go on to define h1, h2 and y. Using the formulae for AND, NOT and OR gates, we get:

1. h1 = σ((1-x1) + x2) = σ((-1)x1 + x2 + 1)
2. h2 = σ(x1 + (1-x2)) = σ(x1 + (-1)x2 + 1)
3. y = σ(h1 + h2) = σ(h1 + h2 + 0)

Hence, we have built a multi layered perceptron with the following weights and it predicts the output of a XOr logical operator.



**Q – 6 What is artificial neural network (ANN)? Explain some of the salient highlights in the different architectural options for ANN.**

**Ans –** artificial neural network is a computation model that consists of several processing elements that receive input and deliver output based on there predefined activation function.

It has the ability to learn, recall and generalize from the given data by suitable assignment and adjustment of weights. The collection behaviour of the neurons describes its computational power, and so signal neuron carries specific information.

**Q – 7 Explain the learning process of ANN with example?** **the challenge in assigning**

**synaptic weights for the interconnection between neurons? How can this challenge be addressed?**

**Ans –** An artificial neural network rule or learning process is a method, mathematical logic or algorithm which improve the network performance and or training time usually this rule is applies repeatedly over the network.

In neuroscience and computer science, synaptic weight refers to strength or amplitude of a connection between two nodes, corresponding in biology to the amount of influence the firing of one neuron has on another. The term is typically used in artificial and biological neural network research.

With the soaring demand for computing power and storage, it is challenging to deploy deep neural network applications. Consequently, while implementing the neural network model fro computer vision, a lot of effort and work is put in to increase its precision and decrease the complexity of the model.

**Q – 8 Explain in details backpropagation algorithm, what are the limitation of this algorithm.**

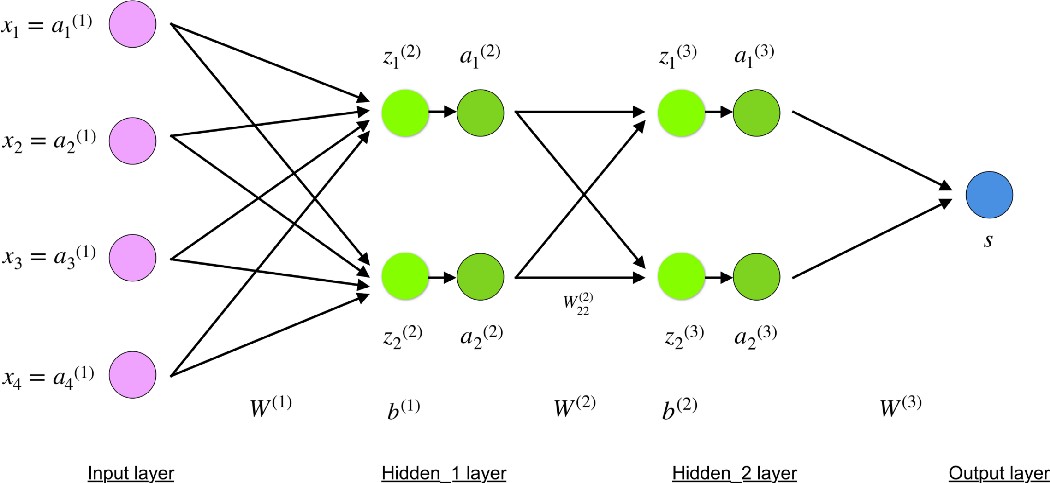
Ans – Backpropagation algorithm is probably the most fundamental building block in a neural network, it was first introduced in 1960s and almost 30 years later (1989) popularized by Rumelhart, Hinton and Williams in a paper call “Learning representations by back propagation error”.

The algorithm is used to effectively train neural network through a method call chain rule. In simple terms, after each forward pass through a network backpropagation performance pass while adjusting the model parameter (weights and biases).

In this article, I would like to go over the mathematical process of training and optimizing a simple 4-layer neural network. I believe this would help the reader understand how backpropagation works as well as realize its importance.

**Define the neural network model.**

The 4-layer neural network consists of 4 neurons for the input layer, 4 neurons for the hidden layer and 1 neuron for the output layer.



**Input layer**

The neurons, colored in purple presented in the input data. These can be as simple as scalars or more complex like vector or multidimensional materices.

**xi = ai(1), i € 1,2,3,4**

the first set of activation (a) are equal to the input values. NB: “activation” is the neurons values after applying an activation function. See below.

**Hidden layer**

The final value at the hidden neuron, colored in green, are computed using z^l – weighted inputs in layer l, a^l – activations in layer l. For layer 2 and 3 the equation are:

l = 2

Z(2) = W(1)x + b(1)

A(2) = f(Z(2))

l = 3

Z(3) = W(2)a(2) + b(2)

A(3) = f(Z(3))

W2 and w3 are the weights in layer 2 and 3 while 2 and 3 while b2 and b3  are the biases in those layers.

Activations *a²* and *a³* are computed using an activation function *f*. Typically,

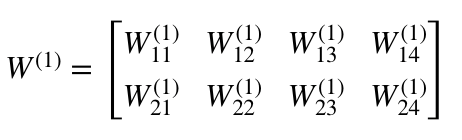
This function f is non -linear (e.g. sigmoid,ReLU,tanh) and allows the network to learn complex patterns in data. We won’t to over the details of how activation functions works, but,

If interested, I strongly recommended reading.

Looking carefully, you can see that all of *x, z², a², z³, a³, W¹, W², b¹*and*b²* are missing their subscripts presented in the 4-layer network illustration above. **The reason is that we have combined all parameter values in matrices, grouped by layers.**This is the standard way of working with neural networks and one should be comfortable with the calculations. However, I will go over the equations to clear out any confusion.

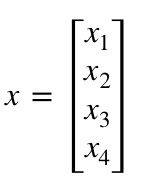
Let’s pick layer 2 and its parameters as an example. The same operations can be applied to any layer in the network.

* W1 is a weight matrix of shape (n, m) where n is the number of output neurons(neurons in the next layer) and m is the number of input neurons(neurons in the previous layer). For us, n=2 and m=4.

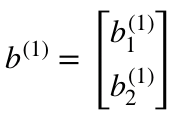


**NB:** The first number in any weights subscript matches the index of the neurons in the next layer(in our case this is the hidden\_2 layer) and the second number matches the index of the neuron in previous layer ( in out case this is the input layer).

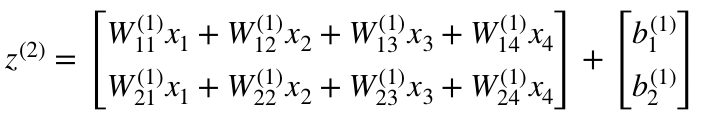
*x* is the input vector of shape *(m, 1)* where *m* is the number of input neurons. For us, *m = 4*.



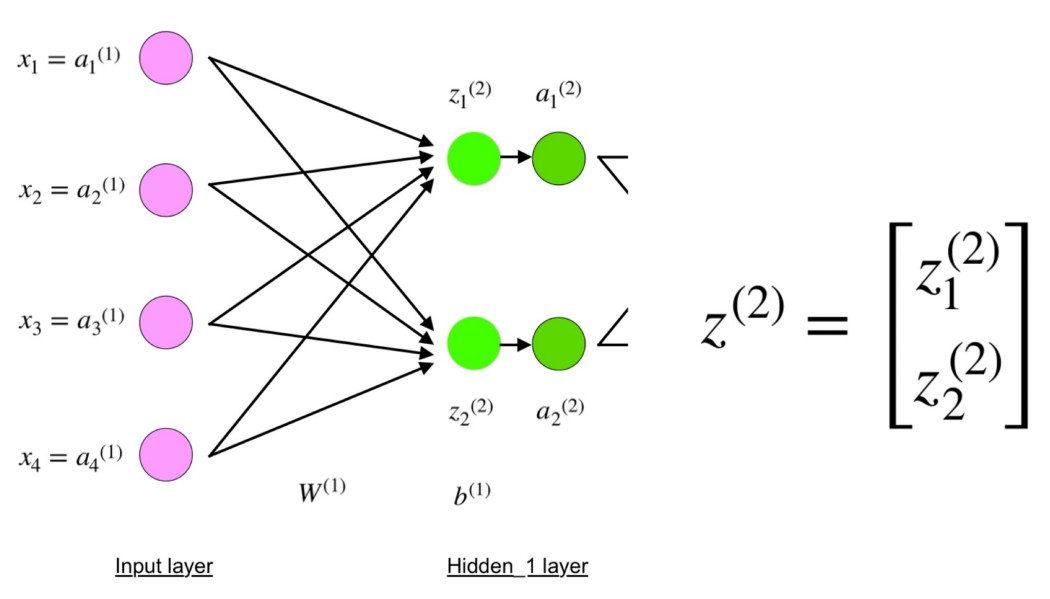
*b¹* is a bias vector of shape *(n , 1)* where *n* is the number of neurons in the current layer. For us, *n = 2*.



Following the equation for *z²,*we can use the above definitions of *W¹, x*and*b¹*to derive “*Equation for z²”*:



Now carefully observe the neural network illustration from above.



You will see that z² can be expressed using (z\_1)² and (z\_2)² where (z\_1)² and (z\_2)² are the sums of the multiplication between every input x\_i with the corresponding weight (W\_ij)¹.

This leads to the same “Equation for z²” and proofs that the matrix representations for z², a², z³ and a³ are correct.

**Output layer**

The final part of a neural network is the output layer which produces the predicated value. In our simple example, it is presented as a single neuron, colored in blue and evaluated as follows:



Again, we are using the matrix representation to simplify the equation. One can use the above techniques to understand the underlying logic.

**Q – 8 Describe, In details, the process of adjusting the interconnection weights in a multi-layer neuron network.**

**Ans –** Learning, in artificial neural network, is the method of modifying the weights of connection between the neurons of a specified network. Learning in Ann can be classified into three categories namely supervised learning, unsupervised learning and reinforcement learnig.

**Q – 9 what are the step of backpropagation algorithm? Why a multi layer neural network is required?**

**Ans –** Steps of backpropagation

1. Input X arrive through the preconnected path.
2. The input is modelled using true weights are usually chosen randomly.
3. Calculate the output of each neuron from the input layer to thehidden layer to the output layer.

Multi-layer network solve the classification problem for non linear sets by employing hidden layer, whose neuron are not directly connected to the output. The additional hidden layer can be interpreted geometrically as additional hyper-planes, which enhance the separation capacity of the network.

**Q – 10 Write a short notes.**

**Ans – 1. Artificial neuron –** An artificial neuron is a connection points in an artificial neural network. Artificial neural network, like the human body biological neural network, have a layered architecture and each network node has the capability to process input and forward output to other nodes in the network.

**2.Multi-layer perceptron –** Multilayer perceptron is a supplement of feed forward neural network. It consists of three types of layers – the input layer, the output layer and hidden layer.

**3.Deep learning –** It is a subset of machine learning, which is essentially a neural network with three or more layers. These neural network attempt to simulate the behaviour of the human brain – albeit far from matching its ability – allowing it to learn from large amount of data.

**4.Learnig rate –** in machine learning and statistics, the learnig rate is a tuning parameter in as optimization algorithm that determine the step size at each iteration while moving toward a minimum of a loss function.

**Q – 11 Difference between Activation function and threshold function.**

**Ans –** A threshold value determines whether a neuron should be activated or not activated binary step activation function. The activation function compares the input value to a threshold value. If the input value us greater then the threshold value, the neuron is activated.

**Step function and sigmoid function –** Step function is one of the simplest kind of activation functions. In this we consider a threshold value and if the value of net value say u is greater then threshold then the neuron is activatied.

Sigmoid function is a widely used activation function.

**Sigmoid function = 1/(1+e-x)**

**Single layer vs multilayer perceptron –** perceptron are two layer network with one input and one output. Multi-layered network have at least one hidden layer (all the layer between the input and the output layer hidden). A single-layer perceptron can learn linear functions, but multi-layered perceptron can also learn non-linear function.