Activation Function

Tuesday, October 9, 2018 11:29 AM

What is an activation function and why to use them?

Definition of activation function:- Activation function decides, whether a neuron should be activated or not by calculating weighted sum and further adding bias with it. The purpose of the activation function is to **introduce non-linearity** into the output of a neuron.

Explanation :-

We know, neural network has neurons that work in correspondence of *weight, bias* and their respective activation function. In a neural network, we would update the weights and biases of the neurons on the basis of the error at the output. This process is known as *back-propagation*. Activation functions make the back-propagation possible since the gradients are supplied along with the error to update the weights and biases.

From <https://www.geeksforgeeks.org/activation-functions-neural-networks/>

Activation** functions* are really important for a Artificial Neural Network to learn and make sense of something really complicated and Non-linear complex functional mappings between the inputs and response variable. *Theyintroduce non-linear properties to our Network.* ***Their main purpose is to convert a input signal of a node in a A-NN to an output signal. That output signal now is used as a input in the next layer in the stack.

Specifically in A-NN we do the sum of products of inputs(**X**) and their corresponding Weights(**W**) and apply a Activation function **f(x)** to it to get the output of that layer and feed it as an input to the next layer.

From <https://towardsdatascience.com/activation-functions-and-its-types-which-is-better-a9a5310cc8f>

Activation** functions* are really important for a Artificial Neural Network to learn and make sense of something really complicated and Non-linear complex functional mappings between the inputs and response variable. *Theyintroduce non-linear properties to our Network.* ***Their main purpose is to convert a input signal of a node in a A-NN to an output signal. That output signal now is used as a input in the next layer in the

stack.

Specifically in A-NN we do the sum of products of inputs(**X**) and their corresponding Weights(**W**) and apply a Activation function **f(x)** to it to get the output of that layer and feed it as an input to the next layer. *why we use Artificial Neural network techniques such as* ***Deep learning to make sense of something complicated ,high dimensional,non-linear - big datasets, where the model has lots and lots of hidden layers in between and has a very complicated architecture which helps us to make sense and extract knowledge form such complicated big datasets.***

Non-linear functions are those which have degree more than one and they have a curvature when we plot a Non-Linear function. Now we need a Neural Network Model to learn and represent almost anything and any arbitrary complex function which maps inputs to outputs. Neural-Networks are considered ***Universal Function Approximators***. *It means that they can compute and learn any function at all*. Almost any process we can think of can be represented as a functional computation in Neural Networks.

Hence it all comes down to this, we need to apply a Activation function f(x) so as to make the network more powerfull and add ability to it to learn something complex and complicated form data and represent non-linear complex arbitrary functional mappings between inputs and outputs. Hence using a non linear Activation we are able to generate non-linear mappings from inputs to outputs
Also another important feature of a Activation function is that it should be differentiable. We need it to be this way so as to perform backpropogation optimization strategy while propogating backwards in the network to compute gradients of Error(loss) with respect to Weights and then accordingly optimize weights using Gradient descend or any other Optimization technique to reduce Error.

From <https://towardsdatascience.com/activation-functions-and-its-types-which-is-better-a9a5310cc8f>

What is an activation function and why to use them?

Definition of activation function:- Activation function decides, whether a neuron should be activated or not by calculating weighted sum and further adding bias with it. The purpose of the activation function is to **introduce non-linearity** into the output of a neuron.

Explanation :-

We know, neural network has neurons that work in correspondence of *weight, bias* and their respective activation function. In a neural network, we would u**pdate the weights and biases of the neurons on the basis of the error at the output**. This process is known as *back-propagation*. **Activation functions make the back-propagation possible since the gradients are supplied along with the error to update the weights and biases**.

From <https://www.geeksforgeeks.org/activation-functions-neural-networks/>

1). Linear Function :-

- **Equation :** Linear function has the equation similar to as of a straight line i.e. **y = ax**
- No matter how many layers we have, if all are linear in nature, the final activation function of last layer is nothing but just a linear function of the input of first layer.
- **Range :** -inf to +inf
- **Uses : Linear activation function** is used at just one place i.e. output layer.
- **Issues :** If we will differentiate linear function to bring non-linearity, result will no more depend on *input "x"* and function will become constant, it won't introduce any ground-breaking behavior to our algorithm.

For example : Calculation of price of a house is a regression problem. House price may have any big/small value, so we can apply linear activation at output layer. Even in this case neural net must have any non-linear function at hidden layers.

2). Sigmoid Function :-

- It is a function which is plotted as **'S'** shaped graph.
- **Equation :**
  $A = 1/(1 + e^{-x})$

From <https://www.geeksforgeeks.org/activation-functions-neural-networks/>

- **Nature :** Non-linear. Notice that X values lies between -2 to 2, Y values are very steep. This means, small changes in x would also bring about large changes in the value of Y.
- **Value Range :** 0 to 1
- **Uses :** Usually used in output layer of a binary classification, where result is either 0 or 1, as value for sigmoid function lies between 0 and 1 only so, result can be predicted easily to be ***1*** if value is greater than **0.5** and ***0*** otherwise.

3). Tanh Function :- The activation that works almost always better than sigmoid function is Tanh function also knows as **Tangent Hyperbolic function**. It's actually mathematically shifted version of the sigmoid function. Both are similar and can be derived from each other.

- **Equation :-**

From <https://www.geeksforgeeks.org/activation-functions-neural-networks/>

- **Value Range :-** -1 to +1
- **Nature :-** non-linear
- **Uses :-** Usually used in hidden layers of a neural network as it's values lies between **-1 to 1** hence the mean for the hidden layer comes out be 0 or very close to it, hence helps in *centering the data* by bringing mean close to 0. This makes learning for the next layer much easier.

4). RELU :- Stands for *Rectified linear unit*. It is the most widely used activation function. Chiefly implemented in *hidden layers* of Neural network.

- **Equation :-** ***A(x) = max(0,x)***. It gives an output x if x is positive and 0 otherwise.
- **Value Range :-** [0, inf)
- **Nature :-** non-linear, which means we can easily backpropagate the errors and have multiple layers of neurons being activated by the ReLU function

From <https://www.geeksforgeeks.org/activation-functions-neural-networks/>

- **Uses :-** ReLu is less computationally expensive than tanh and sigmoid because it involves simpler mathematical operations. At a time only a few neurons are activated making the network sparse making it efficient and easy for computation.

In simple words, RELU learns *much faster* than sigmoid and Tanh function.

5). Softmax Function :- The softmax function is also a type of sigmoid function but is handy when we are trying to handle classification problems.

- **Nature :-** non-linear
- **Uses :-** Usually used when trying to handle multiple classes. The softmax function would squeeze the outputs for each class between 0 and 1 and would also divide by the sum of the outputs.
- **Ouput:-** The softmax function is ideally used in the output layer of the classifier where we are actually
- trying to attain the probabilities to define the class of each input.

CHOOSING THE RIGHT ACTIVATION FUNCTION

- The basic rule of thumb is if you really don't know what activation function to use, then simply use *RELU* as it is a general activation function and is used in most cases these days.
- If your output is for binary classification then, *sigmoid function* is very natural choice for output layer.

Foot Note :-

The **activation function** does the non-linear transformation to the input making it capable to learn and perform more complex tasks.

From <https://www.geeksforgeeks.org/activation-functions-neural-networks/>