https://github.com/prashanth0507- **REFER THIS LINK**

The choice of Activation Functions (AF)

INTRODUCTION

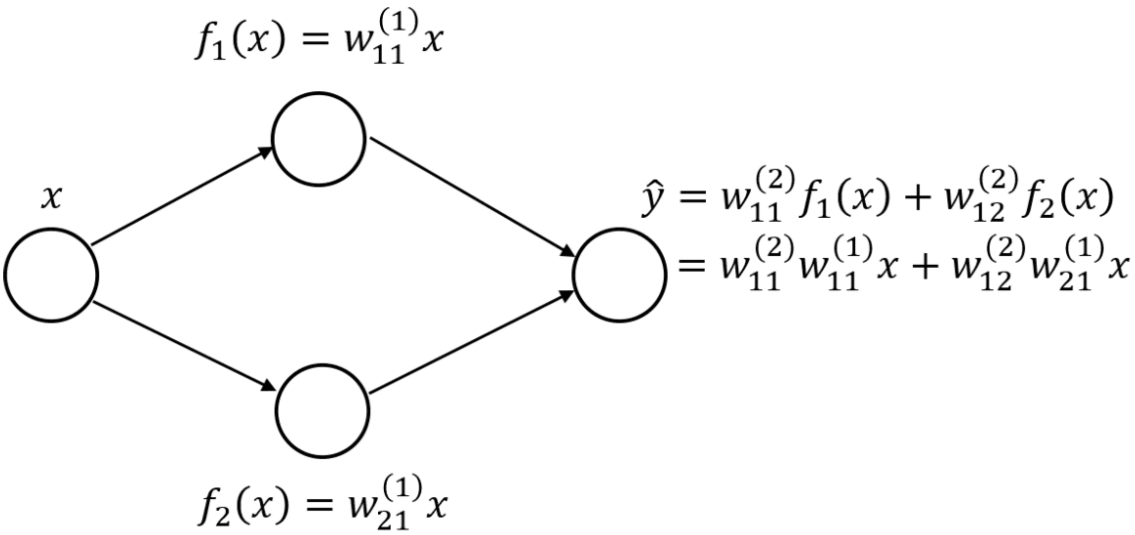
An artificial neural network (ANN) has neurons that are interconnected. Insidethe neuron, there is an activation function that determines the value issued by the neuron. There are several types of activation functions. The choice of
activation function will determine the result of an ANN. This paper describes the performance comparison of several activation functions used in an ANN inprocessing IRIS data. There are three types of activation functions being
compared, those are Sigmoid, TanH, and ReLU. The results indicated that the TanH activation function is the best choice for IRIS data.

Activation functions play an integral role in neural networks by introducingnonlinearity. This nonlinearity allows neural networks to develop complex representations and functions based on the inputs that would not be possiblewith a simple linear regression model.

Many different nonlinear activation functions have been proposed throughout the history of neural networks. The three popular ones are: sigmoid, tanh, andReLU.

Using multiple linear layers is basically the same as using a single linear layer. This can be seen through a simple example.

Let’s say you have a one hidden layer neural network, each with two hiddenneurons.


$f_1(x) = w_{11}^{(1)}x$
x
$\hat{y} = w_{11}^{(2)} f_1(x) + w_{12}^{(2)} f_2(x)$
$= w_{11}^{(2)} w_{11}^{(1)} x + w_{12}^{(2)} w_{21}^{(1)} x$
$f_2(x) = w_{21}^{(1)}x$

You can then rewrite the output layer as a linear combination of the original input variable if you used a linear hidden layer. If you had more neurons and weights, the equation would be a lot longer with more nesting and more multiplications between successive layer weights. However, the idea remains the same: You can represent the entire network as a single linear layer. To make the network represent more complex functions, you would need nonlinear activation functions

Sigmoid Function

You can then rewrite the output layer as a linear combination of the original input variable if you used a linear hidden layer. If you had more neurons and weights, the equation would be a lot longer with more nesting and more multiplications between successive layer weights. However, the idea remains the same: You can represent the entire network as a single linear layer.

Hyperbolic Tangent Function

Another activation function to consider is the tanh activation function, also known as the hyperbolic tangent function. It has a larger range of output values compared to the sigmoid function and a larger maximum gradient. The tanh function is a hyperbolic analog to the normal tangent function for circles that most people are familiar with.

Rectified Linear Unit (ReLU)

The ReLU function is a simple max(0,1) function, which can also be thought of as a piecewise function with all inputs less than 0 mapping to 0 and all inputs greater than or equal to 0 mapping back to themselves (i.e., identity function).

STEPS

To create a 1-hidden layer neural network model that adapts to the most suitable activation function for iris data, you can use Python with the help of popular libraries such as NumPy, Pandas, and scikit-learn.

we first load the Iris dataset and perform necessary preprocessing steps such as encoding the categorical target variable, splitting the data into training and testing sets, and scaling the features. We then define a function `create_model()` to create the neural network model with a specified activation function.

Next, we iterate through a list of activation functions (`sigmoid`, `tanh`, `relu`) and evaluate the model's performance using each activation function. The model is trained and tested for each activation function, and the accuracy is calculated and printed. The best activation function with the highest accuracy is stored in the `best_activation` variable.

Finally, the code displays the best activation function for the Iris dataset along with its corresponding accuracy.

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.