1)what is it generally preferable to use a logistic,,Regression classifier rather than a classical perceptron? How can you tweak a perceptron to make it equivalent to logistic regression classifier ?

Ans : A classical Perceptron will converge only if the dataset is linearly separable, and it won't be able to estimate class probabilities. In contrast, a Logistic Regression classifier will converge to a good solution even if the dataset is not linearly separable, and it will output class probabilities.

If you change the Perceptron's activation function to the logistic activation function (or the softmax activation function if there are multiple neurons), and if you train it using Gradient Descent, then it becomes equivalent to a Logistic Regression classifier.

2)why was the logistic activation function to key ingredient in training the first MLPs ?

Ans : Because the derivative of the logistic function is always nonzero, so Gradient Descent can always roll down the slope. When the activation function is a step function, Gradient Descent cannot move, as there is no slope at all.

The backpropagation algorithm may be used with other activation functions, instead of the logistic

function.

3)name the three popular activation function.can you draw them.

Ans :Logistic/logit/sigmoid

step/threshold

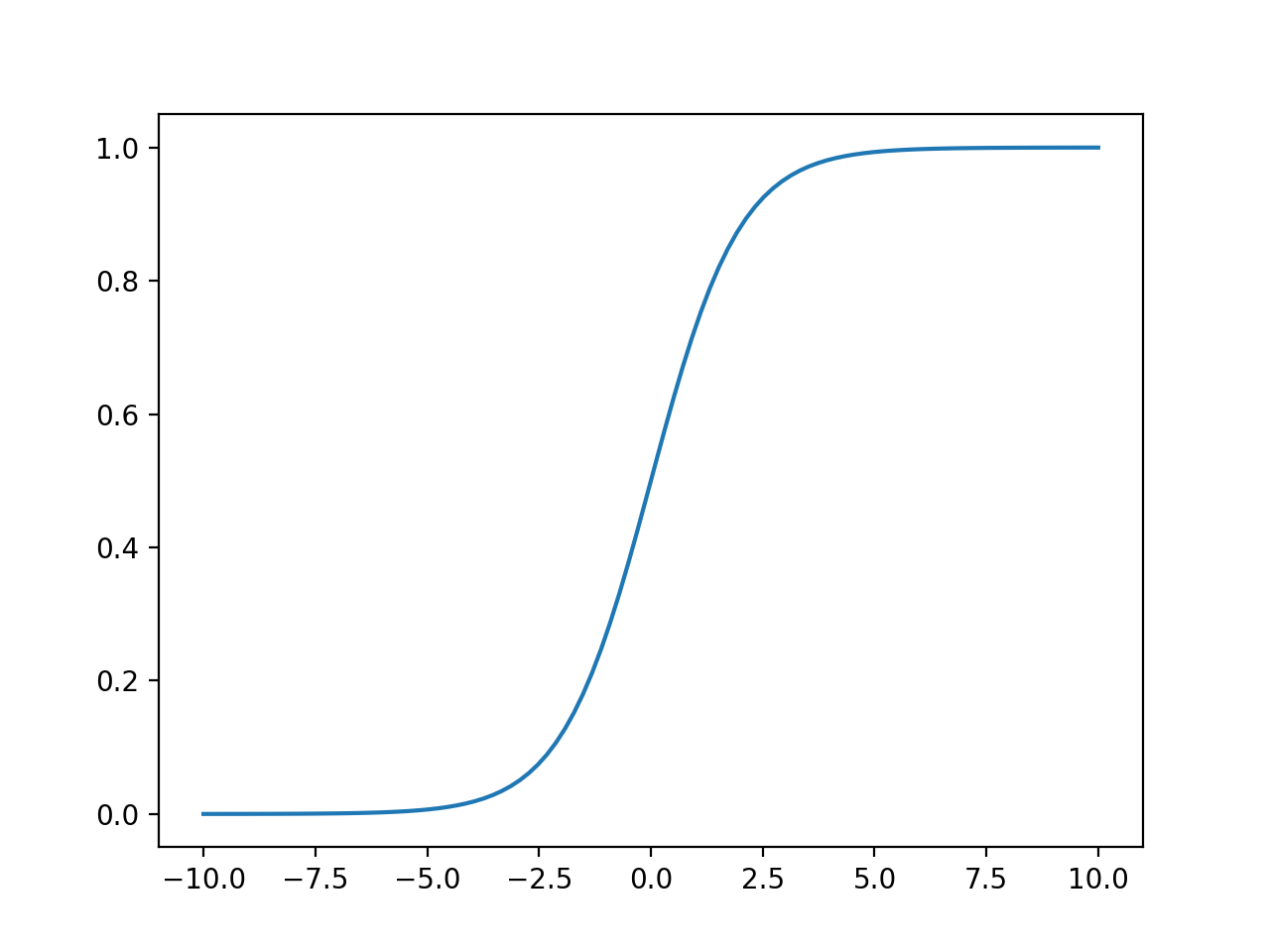
ReLU

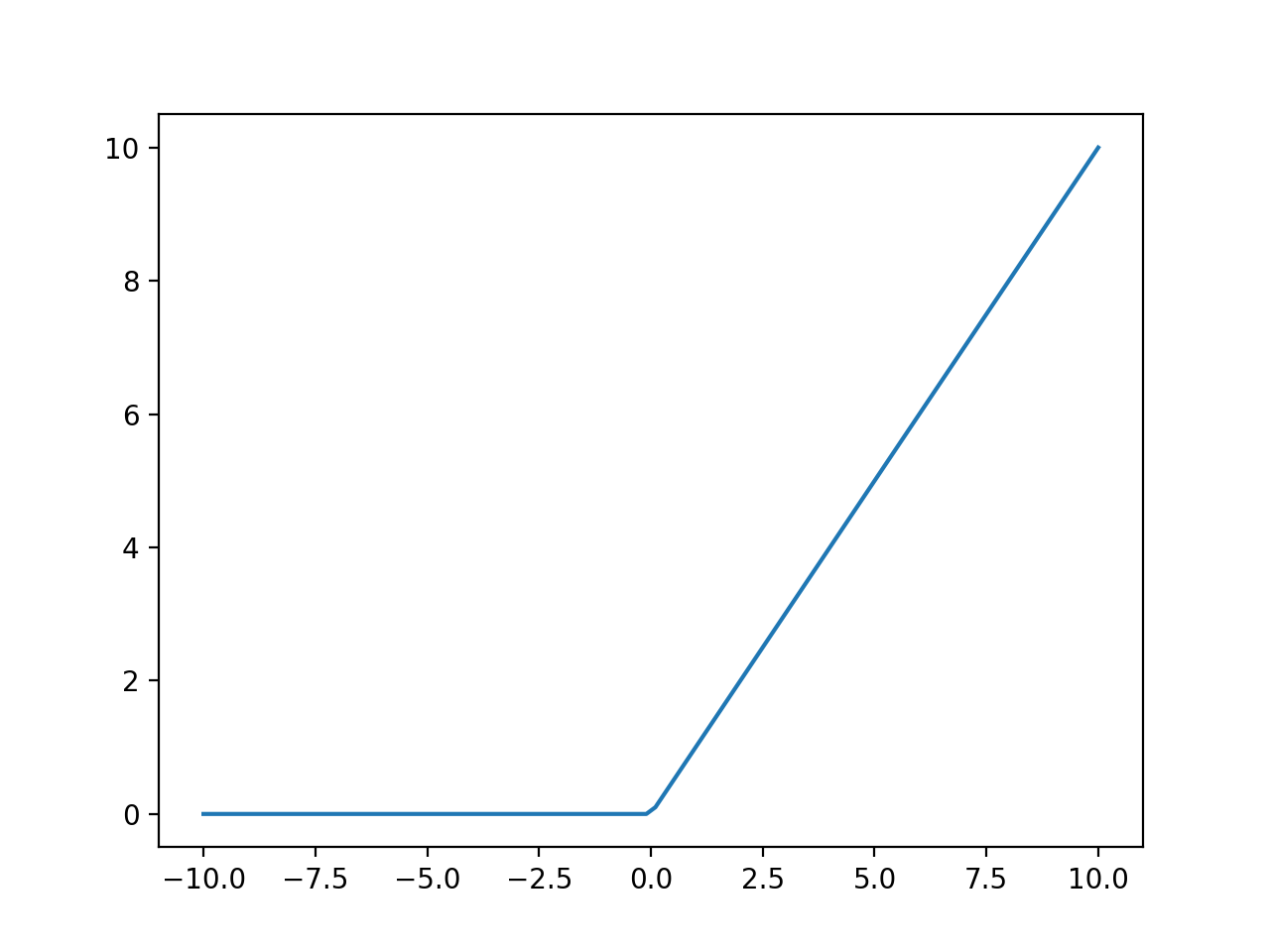
hyperbolic tangent tanh

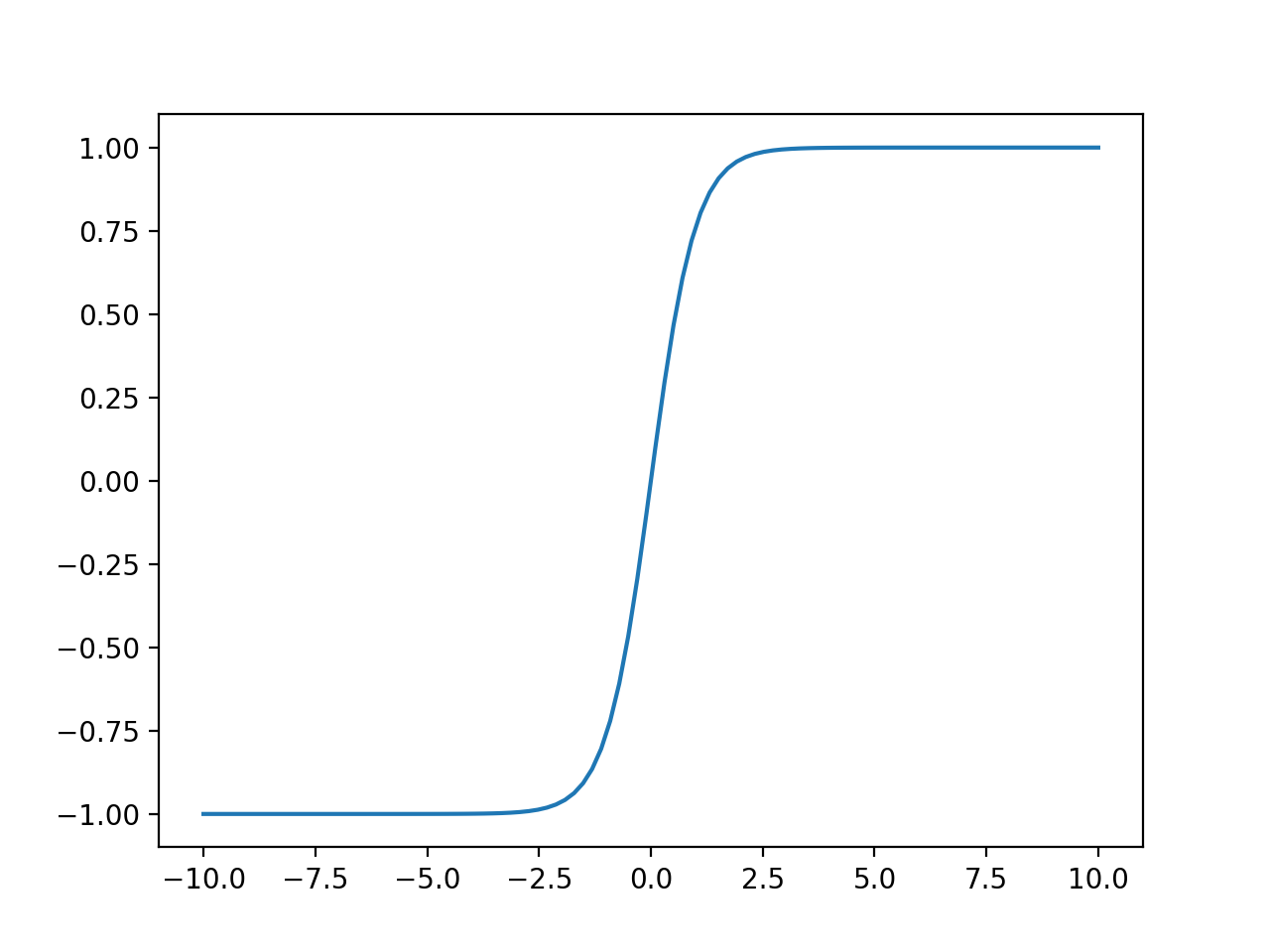
piecewise linear

ELU.

•Logistic



•ReLU

•Tanh.

4 )suppose you have an MLP composed of one input layer with 10 passthrough Neurons.followed by one hidden layer with 50 artificial neuron and finally one output layer with 3 artificial neuron.all artificial neuron use the ReLU activation function ?

•what is the shape of the input matrix X?

Ans : m x 10

•what about shape of the hidden layers weight vector Wh and the shape of its bias vector bh ?

Ans : Wh: 10 × 50

bh: 50 × 1

•what is the shape of the networks output matrix Y ?

Ans : Y: m × 3

• write the equation that computers the networks output matrix Y as a function of X,Wh,bh,Wo and bo

Ans :

Y = relu(X × Wh + bh) \* Wa + b0.

5)How many neurons do you need the output layer if you want to classify email into spam or ham ?what activation function should you use in the output layer ?if instead you want to tackle MNIST ,how many neurons do you need in the output layers,using what activation function?

Ans : 1.You just need one neuron in the output layer of a neural network—for example, indicating the probability that the email is spam. You would typically use the logistic activation function in the output layer when estimating a probability.

2.In most cases, you can use the ReLU activation function in the hidden layers. It is a bit faster to compute than other activation functions, and Gradient Descent does not get stuck as much on plateaus, thanks to the fact that it does not saturate for large input values (as opposed to the logistic function or the hyperbolic tangent function, which saturate at 1).

For the output layer, logistic for binary classification, softmax for multiclass classification, or no activation function for regression.

3.You need 10 neurons in the output layer, and you must use the softmax activation function, which can handle multiple classes, outputting one probability per class.

6)what is the backpropagation and how does it work ?what is the difference between backpropagation and reverse mode autodiff ?

Ans : Backpropagation is a technique used to train artificial neural networks.

It first makes a prediction and computes the error, then goes through each layer in reverse to measure the error contribution of every model parameter (all the weights and biases). Finally, it performs a Gradient Descent step using these gradients (it slightly tweaks connection weights to reduce error).

This backpropagation step is typically performed thousands or millions of times, using many training batches, until the model parameters converge to values that minimize the cost function.

To compute the gradients, backpropagation uses reverse-mode autodiff. Reverse-mode autodiff performs a forward pass through a computation graph, computing every node's value for the current training batch, and then it performs a reverse pass, computing all the gradients at once. the difference Well, backpropagation refers to the whole process of training an artificial neural network using multiple backpropagation steps, each of which computes gradients and uses them to perform a Gradient Descent step. In contrast, reverse-mode autodiff is a simply a technique to compute gradients efficiently, and it happens to be used by backpropagation.

7)can you list all the hyperparameters you can tweak in an MLP?if the MLP overfits the training data,how could you tweak these hyperparameters to try to solve this problem ?

Ans :

number of hidden layers

number of neurons in each hidden layer

activation function used in each hidden layer and in the output layer

the weight initialization logic

If the MLP overfits the training data, you can try reducing the number of hidden layers and reducing the number of neurons per hidden layer.

8)Train a deep MLP on the MNIST dataset and see if you can get over 98% precision.try adding all the bells and whistles.