Read Me

In these project, I first define a sigmoid function which will be called for many times, then defined the accuracy function based on prediction and original data.
Then defined a Neural class, in this class you should input 4 variables learning_rate, max_iteration, hidden_layers, convergence_threshold. Then I defined a function in this class to calculate multicost between different layers. Then I defined a function to calculate the next layers' predict based on the input. Then I made a function call fit to build the model, it will keep working on updating the model and will not stop until the difference between last two cost is smaller than convergence_threshold and the iterate times is larger than 1000.

For the data, I put them into 3 part, the first part contain 90 samples, each kind contain 30 samples. And both of the validation and test set contains 30 samples, each kind contain 10 samples.

accuracy on training data is 0.9777777777777777
accuracy on validation data is 0.9666666666666667
accuracy on test data is 1.0

And you can also input the data by yourself.

Reference:
Get assistance from Yiqiao Xin, Jinyong Hu
https://blog.csdn.net/u010177286/article/details/79606304
http://neuralnetworksanddeeplearning.com/
https://towardsdatascience.com/how-to-build-your-own-neural-network-from-scratch-in-python-68998a08e4f6
https://iamtrask.github.io/2015/07/12/basic-python-network/
https://www.kdnuggets.com/2018/10/simple-neural-network-python.html