Neural Network

1.The data set you have chosen and its main characteristics.

The datasets with ids 53 and 602 were selected on the uci website.The dataset with id 53 is Iris, with a small number of features, only four, and all of them are continuous and size-differentiated feature data.The dataset with id 602 is a dry bean dataset with a high number of features, 16 dimensions, and tens of thousands of samples.

Due to the neural network's ability to handle relatively complex tasks, I selected the image dataset of mnist, which contains ten digital categories from 0 to 9, and the number of training samples reached 6w, which can better test the training effect of the model.

ps:There is no missing data in the above three datasets.

2.The data preprocessing operations you have applied to improve data quality.

- For numerical features, there are characteristics of uneven distribution and large order of magnitude gap. Normalizing them can effectively avoid the risk of weight bias and gradient explosion during training.
- For non-numeric features, a one-hot approach is used to encode them so that they can be applied to the loss function calculation for logistic regression. However, this is prone to excessive number of features and slow computation, so pca is used for dimensionality reduction.
- Mnist provides a channel of 28 * 28 image data, which is flattened to transform the data into 728 dimensional data
- There are multiple categories in the dataset target, and when dividing the dataset, the distribution of the data will be balanced to ensure that the number of samples in each category is as equal as possible

3.The design of the different modules of your program. In describing each module, you should relate your design to the theories you have learnt in the class.

- **common.py** holds the hyperparameters for model training and model testing, including training epoch, iteration count, gradient descent parameters, etc.
- The **config** folder contains settings for model hyperparameters for different datasets
- Under the **process** folder **ucidata.py** and **mnistdata.py** are the modules that process the data and improve the quality of the data. After reading the data into memory through the data import method provided by uci and mnist, the data is then processed in the way mentioned in question 2, and the data is categorized into raw data, training set data, test set data, metadata, etc., and put into a folder with the name **id**.
- The **layer** folder defines classes for various network layers, which are fullyconnet layer,Relu layer,softmax layer.
- The **mlp.py** defines neural networks belonging to different datasets.
- Due to the large size of some sample data, batch processing was adopted. The data was randomly sorted in the **dataset. py** , and the data was extracted according to certain rules and placed in the neural network
- **test.py** is a model testing module that outputs the correctness or incorrectness of the predictions and the probability of belonging to a certain category.
- **train.py** is the model training module, which uses batch stochastic gradient descent to make the model converge more quickly.

4.The prediction results and your interpretation of the results.

The code is set up with a training set to test set ratio of **4:1**.When training the model, the number of iterations per epoch is the same for the same dataset

Firstly, the following assertion is given: without adding a hidden layer, neural networks are no different from using multi class logistic regression methods.

4.1 Iris dataset

The number of features in this dataset is **4,** and the category is **3**. Here are the results of the neural network on the test set with different hidden layers and node numbers.

hidden layer nums	node nums	convergence epoch	accuracy
0	None	1000	100%
1	6	600	97%
1	32	250	97%
2	16,8	None	93%

It can be seen that for simple datasets, excellent fitting results can be achieved without the need for hidden layers, which is consistent with Resnet's view that the fitting upper limit of networks with increased depth is definitely not as good as shallow networks. There are too many layers in the network, which may even result in an inability to converge.

4.2 Dry bean dataset

The number of features in this dataset is **16**, and the category is **7**. Here are the results of the neural network on the test set with different hidden layers and node numbers.

hidden layer nums	node nums	convergence epoch	accuracy
0	None	200	98%
1	10	150	92%
2	64,32	100	92%

It can be seen that for simple datasets, excellent fitting results can be achieved without the need for hidden layers, which is consistent with Resnet's view that the fitting upper limit of networks with increased depth is definitely not as good as shallow networks. But adding a hidden layer can make the network converge faster.

4.3 Mnist dataset

Due to the low configuration of the author's computer, adjusting the batch size will result in an out of memory issue, so only 20 epochs were run and the results were output. The following are the results on the test set, indicating that the three-layer neural network has achieved good fitting results.

hidden layer nums	node nums	convergence epoch	accuracy
0	None	10	89%
1	256	20	92%
2	256,64	20	89%
2	128,64	15	83%
3	256,128,64	None	98%

This indicates that for slightly more complex classification tasks, better results can be achieved by increasing the number of hidden layers.

5.Limitations and possible improvements of the program.

- Code lacks handling of missing or ambiguous data.
- More network layers can be designed to achieve learning goals by imitating ideas such as resnet and batch normalization.
- Since the model training is simple and there are not a large number of hyperparameters, there is no division of the validation set.