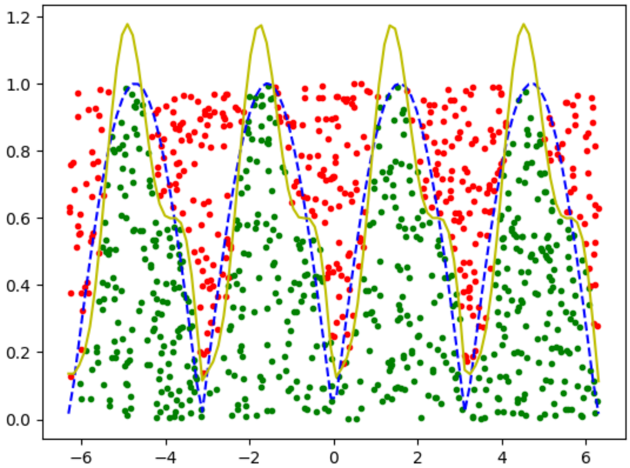
AI Systems Work4

Name: CAO Xinyang
HDU ID: 20321308

1. Job description

Approximate a function: Absolute(Sin(x)) X: -6,3..6.3

2. Graphs of Network training dynamics



3. Each hyperparameter description and its impact on accuracy.

Learning rate:

It controls how well the model responds to estimation errors each time the model weights are updated. In gradient descent, the value of the learning rate is very important, it will determine the speed at which the parameter moves to the optimal value. If the learning rate is too large, the function may exceed the optimal value without convergence or even divergence. If the learning rate is too small, the optimization efficiency may be too low, the convergence speed may be too slow, and the algorithm may fall into local optimum. Under the premise of ensuring convergence, an appropriate learning rate should be able to converge as soon as possible. In Adam optimization algorithms, the learning rate decreases with increasing time.

Batch size:

It refers to the number of images that are put into the network model at a time. A small sample of data in the training set is used to update the backpropagation parameters of the model weights.

Epoch:

It refers to a complete training process, that is, the process of converting the data of the entire training set into a neural network model. Usually more is better. But the longer it takes, the more likely you are to over assemble.

Layers count:
Hidden layer counting is mainly achieved through "feature extraction". By adjusting the weights, cryptographic neural units can respond to a pattern. In general, the more hidden layers, the better the neural network's ability to express the data and the easier it is to fit the training set. The smaller the number of hidden layers, the shorter the training time, but it is prone to misfit. But too many hidden layers can also lead to too long training time, overfitting, etc.

Neurons count per layer:
Using too few neurons in the hidden layer can lead to underfitting. Conversely, using too many neurons can also cause problems. First, too many neurons can lead to overfitting. When there are too many neural network nodes, that is, too much information processing power, the limited information contained in the training set cannot train all neurons in the hidden layer, resulting in overfitting. And even if there is enough information in the training data, too many neurons in the hidden layer will increase the training time and make our goals more difficult to achieve.

Regularization L1:
Add regularization terms to the loss function to describe the complexity of the model, multiply the parameter matrices, and add the elements in the matrix. Increasing the regularization coefficient within a certain range can help reduce the occurrence of overfitting, but too large a regularization coefficient can lead to underfitting.

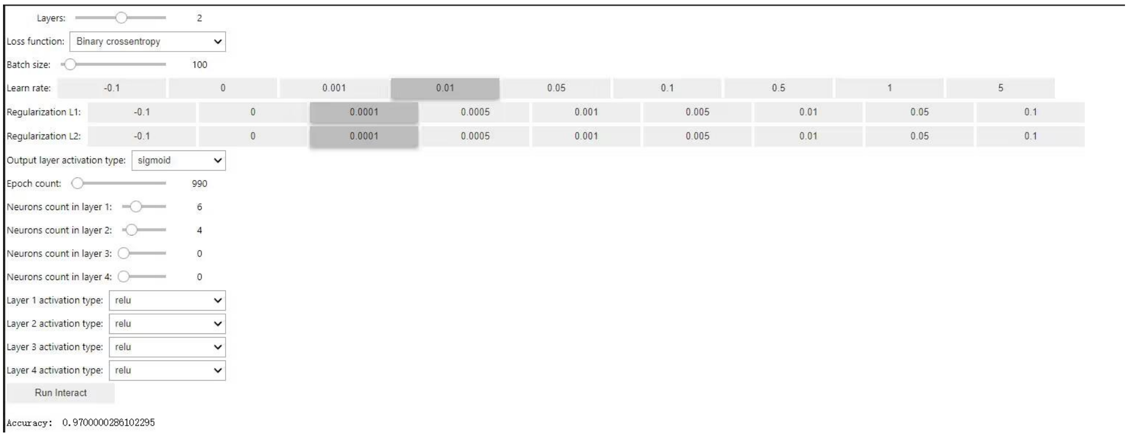
Regularization L2:
Include regularization terms in the loss function to describe the complexity of the model. Multiplies the parameter matrix with the square matrix and adds the square matrix elements of the matrix. Increasing the regularization coefficient by a certain range can help reduce the occurrence of overfitting, but excessive regularization coefficients can lead to underfitting.

Activation function (Layer activation type):
It is the activation function of each layer. The activation function is a nonlinear function. Since the expression ability of linear models cannot be insufficient, we should add nonlinear models to enhance the expression ability of neural networks. Without an activation function, any data entering a neural network is just a linear combination of fixed functions, with no so-called "perception".

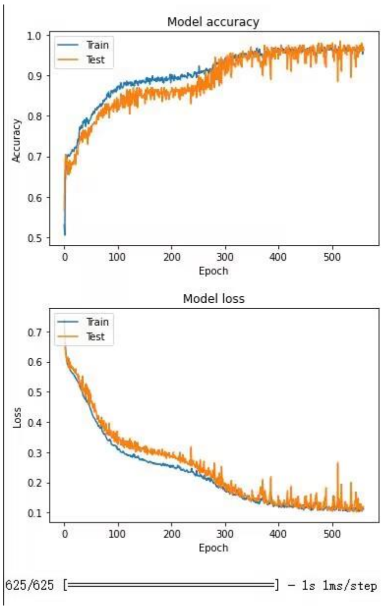
Loss function:
The loss function is a function of the distance between the current output of the algorithm and the desired output. It is a way to evaluate the effectiveness of algorithms in dataset modeling. If your prediction is completely wrong, your loss function will output a higher number. If the estimate is correct, it will output a lower number. When the tuning algorithm attempts to improve the model, the loss function reflects whether the model is improving.

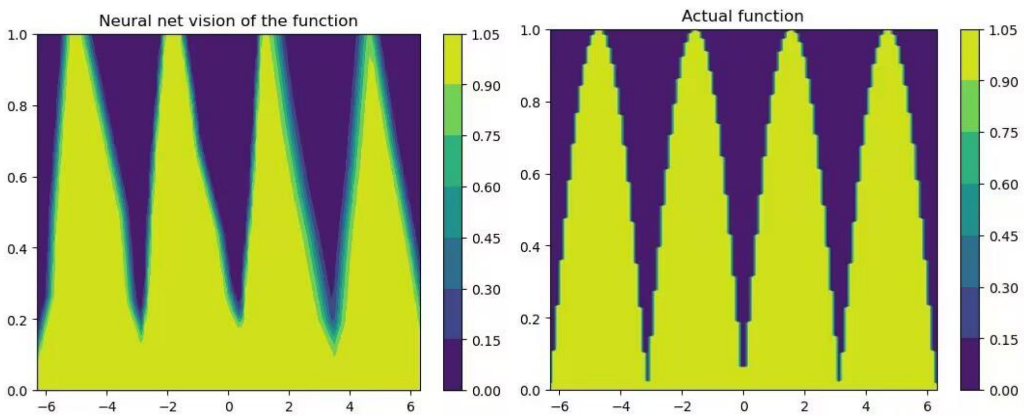
But most importantly, the model is trained to continuously reduce the loss function. When the loss function reaches a minimum, the model is trained to an optimal degree.

4. Hyperparameters' values which were used to reach accuracy value 0.95:



5. Results for these hyperparameters' values:





Conclusion:

Through this lab work, we understand the basic function of neural network and its parameters, and have a certain grasp of how to set the parameters. This will be of great use in my future study.