ML HW3 Report

110062333

林家逸

**1. Difficulty I encountered**

Since the description and the formula is very clear in this homework, I don’t think I encountered any serious problem. If must give one, I would say that I’ve been stuck in implementing the backward function of Activation class for a while. When implementing the softmax + focal loss part, I forgot to add = 1e-5 in log since it doesn’t write in the formula but in the description.

**2. Structure of my model**

For my binary classifier, I only have one input layer and one output layer. Both activation functions are ‘sigmoid’ with learning rate 0.06, iteration times 77500 and loss function ‘cross\_entropy’. My model is very simple since I’ve tried a few more complex models with more layers or more type of activation functions, and I found that all of them will be overfitting when I saw the accuracy of train set is closed to 97% or even to 100%, but the accuracy of validation set is around 75%-85%. For this very simple model, the performance is quite good I think, because I finally got accuracy 90.78% with f1 score average 0.87 in train set and accuracy 87.32% with f1 score average 0.85 in validation set.

For my multi-class classifier, I have one input layer, one hidden layer, and one output layer. The activation functions of them are [‘relu’, ‘relu’, ‘softmax’] with learning rate 0.01, iteration times 20, batch size 128, gamma 4 and loss function ‘focal\_loss’. Since alpha is corresponded to the frequency of each class, I first try to show the bar chart of whole data set and the train set to ensure that both have similar distribution. After that, I try to calculate the frequency of each class and store the result into ‘distribution’ list. Finally, I set the alpha based on the ‘distribution’ list, the formula of my alpha is

With above model, I got accuracy 95.23% with f1 score average 0.93 in train set and accuracy 95.28% with f1 score 0.93 in validation set.

**3. Effort I put to improve my model**

As mentioned above, I’ve tried adding more layers into my model, and it led to overfitting. I also tried to use different activation functions in different layers, but the result is quite bad in the end.

Apart from trying different hyper parameters and structure, I also try to split my data with different random seed to let the train set have more similarity distribution to the whole data. And I personally think that this step is also important since if the distribution of train set is not like the whole data, the model will have more probability of bad performance with new data.