National Tsing Hua University
Fall 2023 11210IPT 553000
Deep Learning in Biomedical Optical Imaging
Homework 2

SHIU-FENG CHENG[1]

[1] *Brain Research Center, National Tsing Hua University, Hsinchu 30013, Taiwan.*

Student ID:111080527

1. Introduction

Deep leaning models are powerful tools for biological research. Currently they have been applied to many tasks such as image segmentation, identification, or even image quality enhancement. The aim of this report is to have a deeper understanding to neural network (NN) structure and its performance. In this report I'm going to compare: 1. NN with binary cross entropy (BCE) and cross entropy (CE) as loss function. 2. Study the least neurons needed for the specific task by tuning neuron numbers and dropout between layers.

2. Result

2.1 The comparison of BCE loss and CE loss NN model

First of all an obvious difference between these two models are the loss function:

$$BCE(y, \hat{y}) = -[y * ln(\hat{y}) + (1 - y) * \ln(1 - \hat{y})]$$
$$CE(y, \hat{y}) = -\sum_{i} y^i * ln\hat{y}^i$$

However, for binary task:

$$\begin{cases} i = 0, 1 \\ y^0, y^1 = 0, 1 \\ \hat{y}^1 = 1 - \hat{y}^0 \end{cases}$$

By applying these additional conditions to the loss function it turns out that BCE loss and CE loss are equivalent (if $\hat{y}^1{}_{BCE} = \hat{y}_{CE}$). But does this mean that these two models are the same? No, the normalization function applied to CE models is SoftMax while BCE model adopt Sigmoid function. As you can see the outcome of these two functions will be the same only if $\hat{y}^0{}_{BCE} = 0$. Furthermore, if $\hat{y}^0$ and $\hat{y}^1$ are both positive or negative the output of SoftMax is less polarized.

$$SoftMax(\hat{y}^i) = \frac{e^{\hat{y}^i}}{\sum_i e^{\hat{y}^i}}$$
$$Sigmoid(\hat{y}) = \frac{e^{\hat{y}}}{e^{\hat{y}} + 1}$$

Though $\hat{y}^0$ and $\hat{y}^1$ are not constraint to positive value my experiment result do show difference on model loss between BCE and CE model. In my experiment BCE have its training loss converges to 0.016 while CE model get 0.329 loss after 90 epochs of training (Fig.1a). By further tracking the outputs of both models I found that the outputs of CE model are more polarized, thus causing a greater impact on total loss while having predictions. Due to this reason CE model has a total loss higher than BCE model while having similar accuracy (Fig.1b).

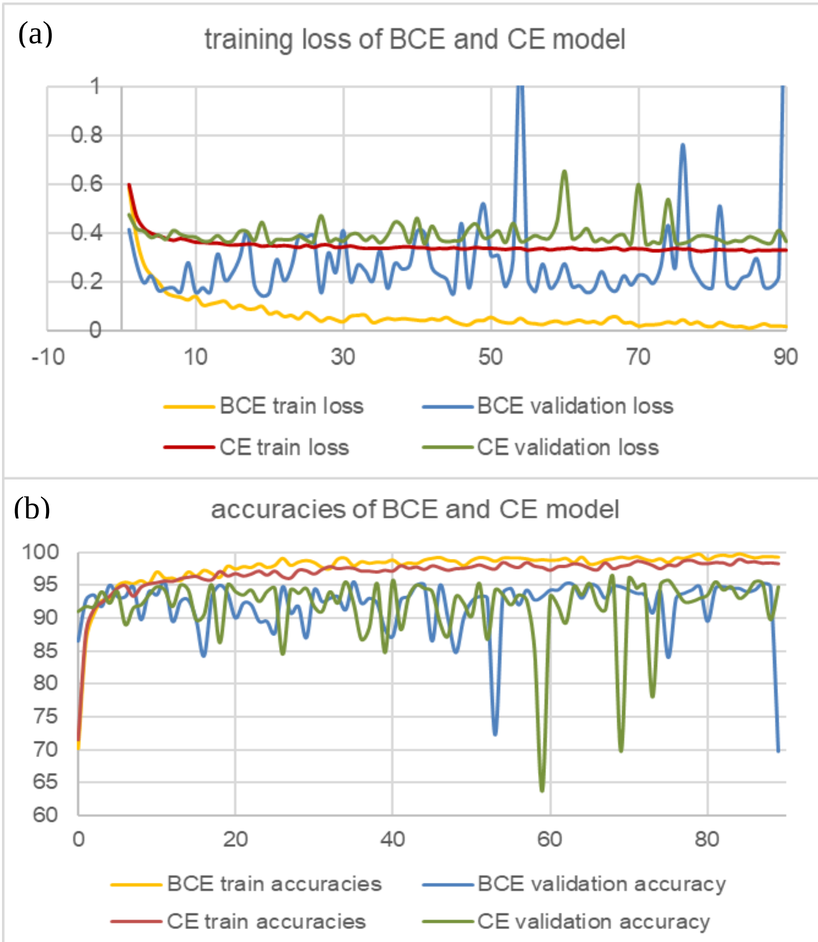

Fig. 1. Performance of BCE and CE during 90 epochs training. (a) The total loss (b) prediction accuracies of training and validation dataset.

2.2 The impact of neuron number on neural network performance

To simplify the discussion about impact of neuron number in this task, I applied a relatively simple BCE loss model which consist of only one hidden layer. The experiment method is done by adjusting the number of neurons in the only hidden layer. According to my experimental results, (Fig.2) I found that there is a performance jump when neuron number enhanced from 16 to 32. While having similar performance for training dataset, the prediction accuracy to validation dataset enhanced to ~94.7% (from 93.5%). And the average loss lowered from 0.4 to 0.3. This indicates that there might be some features helps verification that require >16 neurons.

The other interesting thing is that when having low neuron number there is a probability of a validation accuracy fall (to 50%) and average loss stuck at 0.693 while training. The 50% accuracy indicates that the neural network do not have the ability to effectively predict the status. By tracing the parameters, I found that when the fall occurs the outputs of every input becomes 0.5 which make sense because it is the solution to get the lowest expected value of loss (with 50% accuracy).

$$expected\ BCE(0.5, \hat{y}) = -[0.5 * ln(\hat{y}) + 0.5 * \ln(1-\hat{y})] = -0.5 * \ln[(\hat{y})(1-\hat{y})]$$

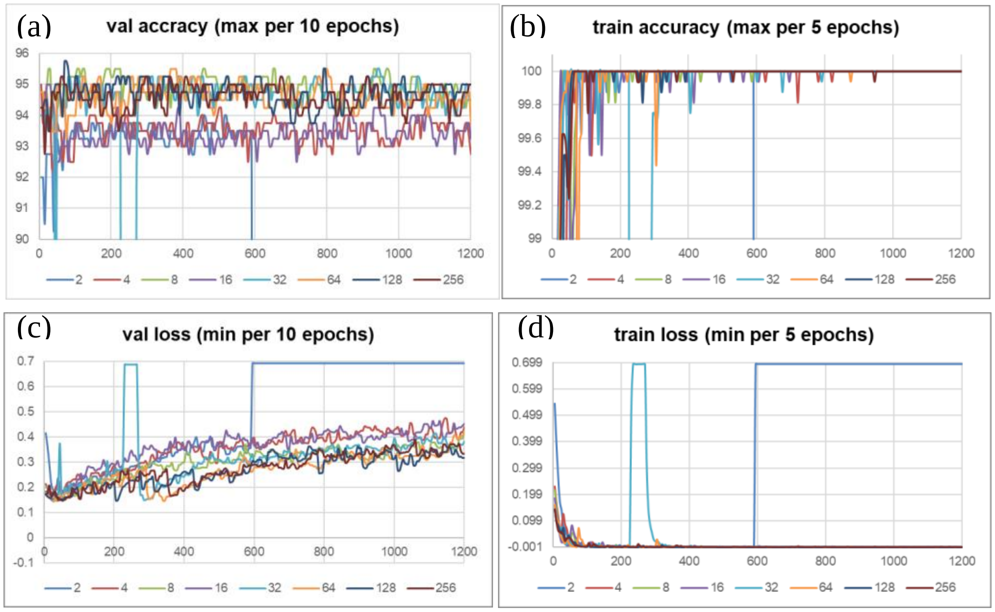

Fig. 2. Performance of single layer BCE model with different number of neurons (labeled in different colors). The prediction accuracies (a, b) and total loss of training (c, d) and validation dataset. To flatten the fluctuation between epochs, the max value of every 10 epochs is adopted for plotting validation accuracy and the minimum of every 10 epochs are presented in validation loss plot. Since training dataset have less variance data points are selected for every 5 epochs instead.

2.3 The impact of dropout on neural network performance

In this section I'm aiming to find out the influence of dropout between layers. The neurons model used in 2.1 section with 256 neurons is slightly adjusted by adding a dropout layer between the only hidden layer and output layer. Expecting the result to be similar to reducing neuron numbers, I choose to experiment with dropout value: 0, 0.5, 0.75, 0.875 which corresponds to 256, 128, 64, 32 working neurons while training. The result however is out of my expectation, the performance is greatly degraded when elevating the dropout ratio. The most significant impact is the score on training dataset, the accuracy falls to ~60% and the BCE loss achieves ~0.65 with 0.875 dropout ratio (Fig.3b,d). What is the reason of the varying performance comparing the outcome of section 2.2? The difference between dropout and having less neuron is that dropout forces the network to predict with more features while reducing neuron numbers limited the number of features the network can learn. Setting high dropout rather like the story of the blind men inspecting the elephant thus gives a terrible score for training. On the other hand, having less neuron numbers limits the network only learn few feature that is representative to the training data thus might have bias and have lower accuracy for validation dataset.

Another thing I learnt from this experiment is why people usually choose a dropout ratio of 0.5 instead of a higher value. In my result dropout ratio higher than 0.5 apparently do not enhance the score on validation dataset which is the motivation of adding this layer. The principle of improving the NN with dropout layer is by randomly remove neurons output so that the network does not get too rely on few neuron sets. When adding dropout ration from 0 to 0.5 it normalized the contribution of each node, however this normalizing effect saturated when the ratio is higher than 0.5. What's worse is that the high dropout ratio indicates fewer working neurons while training as discussed in previous paragraph this lowers the learning quality.

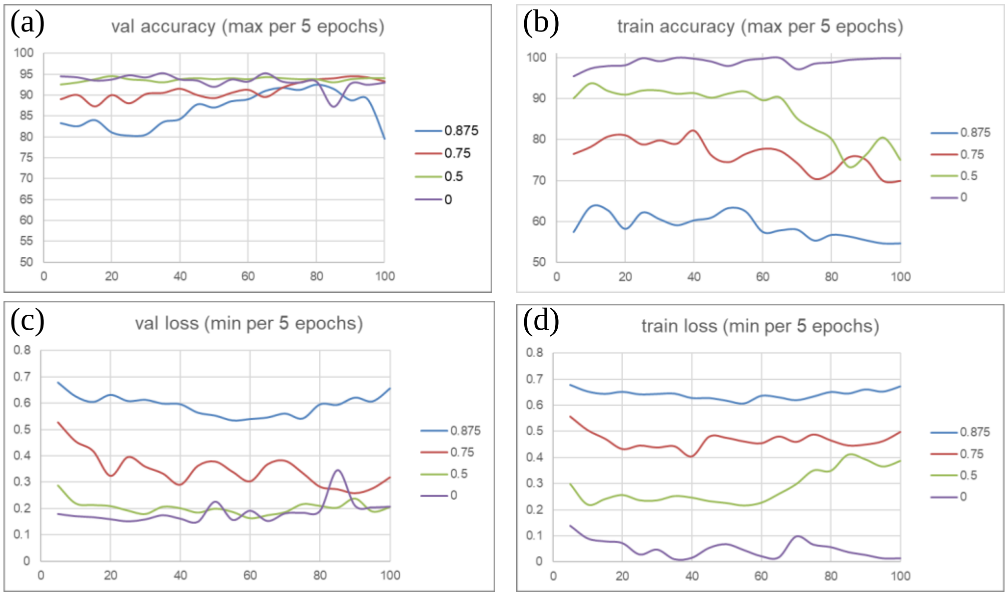

Fig. 3. Performance of single layer BCE model with different dropout value (labeled in different colors). The prediction accuracies (a, b) and total loss of training (c, d) and validation dataset. To flatten the fluctuation between epochs, the max value of every 5 epochs is adopted for plotting accuracy and the minimum of every 5 epochs are presented in loss plots.

3. Material and Method

3.1 BCE/CE model comparison

Modified from lab3 code. The network is composed of 3 hidden layers of 64 linear neurons with batch normalization, ReLU activate function, and 0.5 dropout ratio. The third hidden layer is then merged to one or two neurons in output layer, depends on if it is a BCE loss or CE loss model.
The code can be found at:
https://github.com/Frank497/NTHU_2023_DLBOI_HW/blob/main/hw2/report%20code/hw2_BCE_.ipynb
https://github.com/Frank497/NTHU_2023_DLBOI_HW/blob/main/hw2/report%20code/hw2_CE.ipynb

3.2 neuron numbers & dropout layer

The model is a BCE loss NN modified from lab3 code. The network is composed of 1 hidden layers of linear neurons with batch normalization, ReLU activate function, and dropout ratio. Then the hidden layer is connected to the output layer. Depending on the experimental design, different number of neurons and dropout ratio is set in the network.
The code can be found at:
https://github.com/Frank497/NTHU_2023_DLBOI_HW/blob/main/hw2/report%20code/hw2_BCE_neuron_numbers_%26_dropout.ipynb