Q1 Answer Report

1. Introduction

This experiment is an attempt based on a simple version of the model mentioned in paper 'Obtaining Spatially Resolved Tumor Purity Maps Using Deep Multiple Instance Learning In A Pan-cancer Study'. By imitating the architecture mentioned in this paper, this experiment successfully achieves the prediction of the proportion of handwritten digits 0 and 7 in unknown bags. After training, the coefficient of determination of the regression model reached more than 0.99, showing its good performance. All the code of the experiment is saved in a file named 'MNIST_0_7_NN_Regression.ipynb' and the generated model is saved in 'model_20000bags.pt', both in the same folder as this report.

2. Implementation details

As the simple version does not need to process complex pathological section images like the original paper, but only needs to process MNIST data sets with fewer pixels, the author simplified and replaced some modules used in the original architecture, such as feature extractor and pooling layer, in the actual construction of the model. It should be emphasized that although the model has made some simplifications, these simplifications are intended to make the model better or faster to handle the current data set and got an acceptable result in the final test.

2.1 Data obtaining and pre-processing

Using 'torchvision.datasets.MNIST' can download MNIST dataset directly from torchvision and translate it into tensor, but the original dataset contains 10 numbers from 0 to 9, so the dataset need to be pre-processed.

First, traverse the data set and store the tensors of the digits 0 and 7, and divide these tensors into the training tensor and the test tensor by about 5:1. Then, random function is used to randomly generate an integer N from 0 to 100. N tensors of digit 0 and (100-N) tensors of digit 7 are put together and mixed, so as to generate a bag with a ratio of N/100 of digits 0, where N is completely random and N/100 is the label of this bag. In this way, 20000 and 2000 different bags are generated from the previously divided training tensors and test tensors respectively, which constitute the training set and test set of the model.

The reason for such tedious pre-processing steps is to simulate the actual scene in the original paper as much as possible. A bag is equivalent to a patient's tissue slides, and the 100 images contained in the bag are equivalent to 100 patches captured on the slides. Assuming that the images of digits 0 represent patches of tumor and the images of digits 7 represent normal patches, thus predicting the proportion of the digits 0 in a bag is equivalent to predicting the purity of the tumor in a tissue slide. When bags were generated, about one-third of them were set to contain only the digits 0 without the digit 7 to simulate tissue slides of normal people with 0 tumor purity.

2.2 Feature extraction

In the original paper, a multi-layer residual neural network (ResNet) is used to extract information from complex tissue slide images. In this simple version, however, the image is much simpler and has significantly fewer pixels, requiring less cumbersome methods to extract information. After the comparison and analysis, the author found that directly using the reshape tensor of the image as feature vectors can have very good effect, and also do not need additional time to train more complex neural network. So here, tensors from the previous data set are loaded directly into the 'DataLoader' as feature vectors and applied to subsequent models.

2.3 Pooling layer

The pooling layer in the model is also simplified to suit the actual scenario. The maximum pooling layer used in this model can well reduce the information redundancy of feature vectors, reduce computational overhead and retain the main features of vectors, so it is placed in front of the Representation Transformation module. As for the parameters, kernel_size is set to 3, padding is set to 0, and all other parameters are default.

2.4 Representation transformation

After getting the pooled vectors, a transformer is needed to convert the vectors to a fraction result. Here, the author chooses to use the same multi-layer perceptron neural network as the original paper, but each layer contains a slightly different number of neural units. As can be seen from Figure 1, the input layer of the perceptron receives 8100 vectors, while the two hidden layers have 900, 100 neural units respectively. Finally, an output is obtained in the output layer, in which the activation function of all neural units uses ReLu function. The optimizer's learning rate is set to 0.0002, in which case the model converges faster.

```
MLP(
  (pooling): MaxPool2d(kernel_size=3, stride=3, padding=0, dilation=1, ceil_mode=Fals
e)
  (_representation_transformation): RepresentationTransformation(
    (fc1): Linear(in_features=8100, out_features=900, bias=True)
    (fc2): Linear(in_features=900, out_features=100, bias=True)
    (fc3): Linear(in_features=100, out_features=1, bias=True)
  )
)
```

Figure 1. A screenshot of the model of pooling and bag-level transformation

3. Result and evaluation

For a regression model, the coefficient of determination (R^2) can be used to measure how well the predicted values fit the truth value. In the optimal case, when all the predicted values are equal to the truth value, R^2 reaches the maximum value 1. Therefore, this experiment also uses R^2 as the standard to judge the quality of the model. At the beginning of the training, the R^2 score of the model was only -1.0965, but after training 50 batches of data, the value rose to 0.97761, and after training 100 batches of data, R^2 remained above 0.99. Finally, the R^2 of predicted values and labels obtained by the test set is 0.99544, indicating that the model has a really good fitting ability. Matplotlib was used to draw the visualization results of the first 50 predicted values and true values in the test set, as shown in Figure 2. The closer the blue point in the figure is to the red line (y=x), indicates the better the fitting ability of the model is.

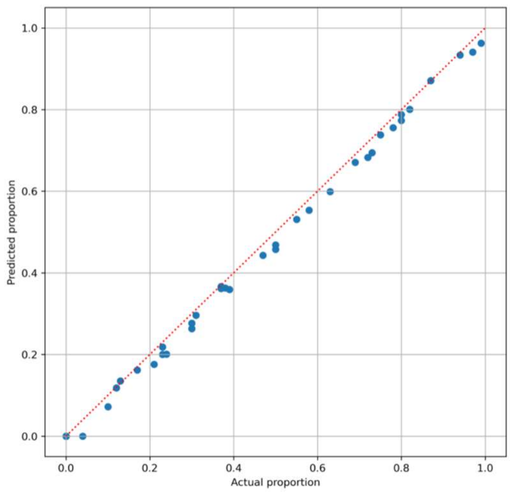

Figure 2. A visualization results of the first 50 predicted values and labels in the test set