Human Kidney Cortex Cells Classification

COM2028 Artificial Intelligence Coursework, Department of Computer Science, University of Surrey

Wish Suharitdamrong (URN: 6605825)

GitHub repository link

1.Introduction

This is an Image Classification project, aiming to classify a different classes of images provided from a dataset. Image classification is a Supervised Learning (SL) problem in Machine Learning (ML) where there are a data and its label. The objective of this project is to classify 8 different classes (types) of cells from an image by using Image Classification method called Convolutional Neural Network (CNN or ConvNet). This project will be using Machine Learning library provided from TensorFlow.

About Dataset

This dataset is about human kidney cortex cell images. The image set was originally extracted from the Ashkar Lab at Indiana University School of Medicine, Indianapolis, Indiana, which is available from the Broad Bioimage Benchmark Collection [1] and other published result using this dataset [2]. In the dataset, there are 200,000 images in 28 x 28 pixels in grayscale as shown in Figure 1.

Figure 1. example of an image in dataset

2.Analysis of the problem

Technical Issue

An issue occurred was importing a train and validation dataset into a batch dataset because its labels are given in CSV file instead of containing in a subdirectory that belongs to its labels. There are several ways to solve this problem, how ever I choose to use a preprocessing method provided from TensorFlow called ImageDataGenerator. This allows me to match images to its labels in CSV file by using its method called flow_from_dataframe.

Data Analysis and Preprocessing

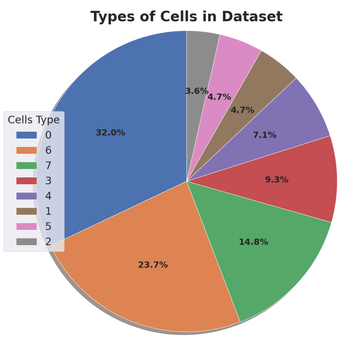

Figure 2. Pie Chart show the proportion of an images belonging to each type of cell.

The dataset is spilt into the train and test set by ratio of 80:20 (150,000 and 50,000 respectively). The dataset contains 8 different classes of cells as shown in Figure 2. It illustrates most of the images belong to class '0' , '6' , '7' for 32%, 23.7% and 14.8% respectively. The other remaining classes are under 10% which can be insufficient for training set and can cause overfitting in a model during training as shown in Figure 3.

In order to avoid overfitting, I have decided to apply Regularization technique as followed:

1. Data Augmentation (random flip, rotation and contrast)
2. L2 Regularization
3. Dropout
4. Batch Normalization

Data Augmentation technique is applied in a preprocessing layer before entering to the CNN layers to increase limited amount of training data and its diversity in training set. After the images pass through the augmentation layer, the images are randomly modified as shown in Figure 4. This technique and other Regularization techniques would reduce the number of high variance, increase a model's robustness and make model generalize better.

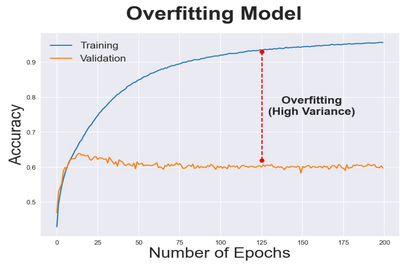

Figure 3. Learning curve of model without data augmentation and regularization

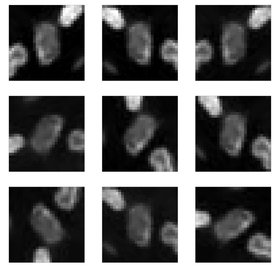

Figure 4. An image of from dataset after applying data augmentation

Training Model

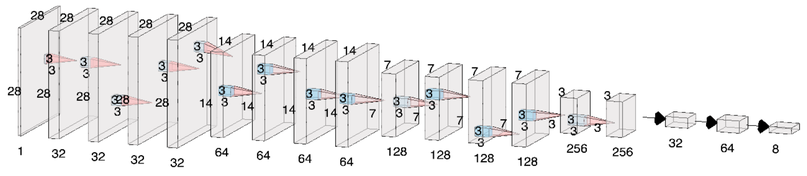

Figure 5. Convolutional Neural Network Model

The model consists of:

- 14 Convolutional Layers of kernel size 3x3 with ReLu activation function and padding same
- 4 MaxPooling Layers of size 2 x 2
- 10 Droupout with rate of 0.15
- 3 Fully Connected layers with ReLu activation function and Softmax in the last layer

The Figure 5 demonstrates a diagram of this model. This diagram was generated by using NN-SVG [3]. For the learning rate, I was experimenting between 1e-3 to 4e-3 and end up with 1e-3 as the most optimal value for learning rate. Therefore, I have tested model with different optimizers with a learning rate at 1e-3 on 200 epochs. The results are shown in Figure 6.

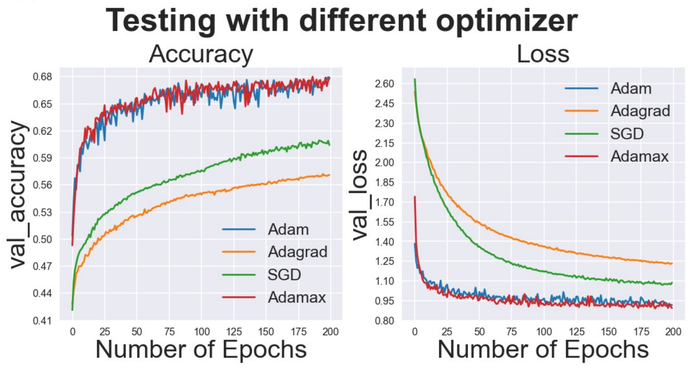

Figure 6. Testing model with different optimizer

It clearly shown that both Adagrad and Stochastic Gradient Descent (SGD) optimizer are not suitable for this model. For Adam and Adamax, both validation accuracy is very close to each other. For the validation loss, Adamax has the least minimum loss overall from all of them.

3.Evaluation

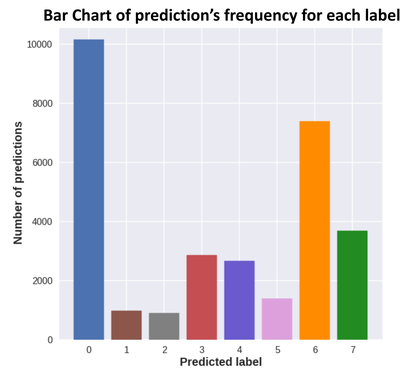

Figure 7. Bar Chart showing frequency of prediction for each labels on validation set

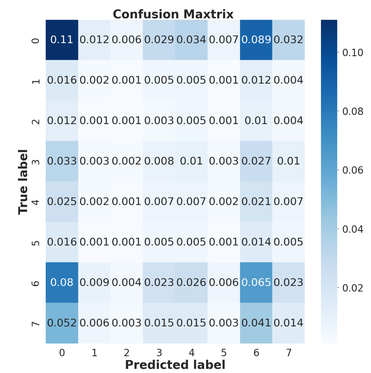

Figure 8. Normalize Confusion matrix of prediction on validation set

During the training, training set is again split up 80:20 for training and validation set (120000 and 30000 images respectively), using a model and its hyperparameter stated in Training Model section with a batchsize of 128 on 200 epochs and categorical crossentropy as the loss function.

Figure 7 visualizes in the frequency of labels predicted in the validation set, its mostly shown labels predicted by the model being class '0' and '6'. The normalize confusion matrix of prediction in validation set (Figure 8) shows that there are many false positive on columns of labels 0 and 6 during the prediction. This is because most of the images in a training dataset belong to class 0 and 6 which is insufficient amount of training data for other labels (1,2,3,4,5,7) as mentioned earlier.

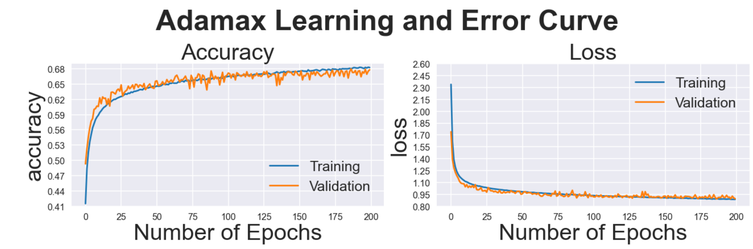

Figure 9. Learning Curve and Error Curve of a model using Adamax optimizer

In order to increase the accuracy of the model, greater amount of training data for other labels is required in order to increase diversity of dataset and have amount of each labels equally balanced. This would allow a model to improve its learning on other labels and improve its accuracy. Also, further optimization in a model can be done to decrease the training time. To conclude this projects, I have decided to use Adamax as the final optimizer since it provides the highest score of 67.23% accuracy on the test set, whereas Adam accuracy was only 65.6-66.9%. The learning and error curve of model using Adamax optimizer is shown in Figure 9.

References

[1] Ljosa, V., Sokolnicki, K. & Carpenter, A. Annotated high-throughput microscopy image sets for validation. Nat Methods 9, 637 (2012). https://doi.org/10.1038/nmeth.2083

[2] A. Woloshuk et al. "In situ classification of cell types in human kidney tissue using 3D nuclear staining." Cytometry Part A, vol. 99, no. 7, pp. 707-721, 2021.

[3] Lenail, A. (2022) NN-SVG [online] available at http://alexlenail.me/NN-SVG/AlexNet.html [Accessed 15th April 2022].