Project Proposal

White Blood Cells Classification with Deep Learning

Shilong Yin sy2792 Diwei Xiong dx2183

Problem Statement

In biological system, white blood cells (also called leukocytes) play a very important role in protecting the body from infections of disease and foreign invaders. They are widely spread throughout the body and the number of them can usually be treated as an indicator of illness. In modern blood test, the number of White Blood Cells (WBC) will be included with specification to the number of each cell type (Neutrophil, Eosinophil, Basophil, Lymphocyte and Monocyte). This offers us a glimpse into our body immune system and any potential risk we are facing. For example, the number of leukocytes and morphological changes or blasts of them provide valuable information for the positive results of disease such as Acute Lymphocytic Leukemia (ALL). [1][2][3]

To fully understand the health condition of body immune system, we have to look deep into the morphology and distribution of white cells through microscope images. Therefore, we want to use deep learning to design an automated system with the ability of categorizing leukocytes so that physicians and scientists may have more time on medical care and research.

Data Description

In this project, we are going to use 10000+ augmented JPEG images containing different type of blood cells to train and test our model (https://www.kaggle.com/paultimothymooney/blood-cells). And the original dataset is 410 images from (https://github.com/Shenggan/BCCD_Dataset). Cell types are Eosinophil, Lymphocyte, Monocyte, and Neutrophil and the data structure will be:

	Training	Test
Eosinophil	2497	623
Lymphocyte	2483	620
Monocyte	2478	620
Neutrophil	2499	624

Table. 1: Structure of Dataset

As for each image, its size will be 320*240 pixels and it will contain only one single type of white blood cells.[4] Therefore, our task will be multi-class single-label categorization with the label set size of 4.

Model

Since we are dealing with image recognition, we would like to use convolutional neural network (CNN) model. Images are structured data and have a large number of features, so it is computationally expensive to use normal feature selection algorithms or dense connected neural network. For CNN, however, we do not need to do feature selections just like other algorithms; CNN can serve as an auto extractor, effectively down sampling the images by using adjacent pixels when doing convolutional

computation. Also, with sparse connection between neurons, CNN enabled weight sharing, which reduces the number of parameters so that computational cost and the efficiency could be further improved. For our case, we would first try Keras and pretrained model based on ImageNet dataset, due to the fact that pretrained model can facilitate the training process by decreasing the amount of time spent on training. Specifically, Conv2D, BatchNormalization and Dropout layers are method we would use when we are about to implement our CNN due to the image feature recognition nature of our project.[5]

Assessment

Since our dataset is not very large, to make sure our model will not overfit, we will split the training data into train set and validation set and monitor their loss and accuracy. During the training process, we can use this to adjust learning rate or even stop the process earlier. Besides, we will plot the loss and accuracy curve as well as the confusion matrix so that we are able to assess the model performance intuitively.

Based on our task, we will use categorical cross entropy as our loss function and backpropagate according to that. The final output will be probabilities assigned each of the four labels and the predicted output is the one with largest probability.

The baseline for the accuracy of this model with be 25% (random guess). We can evaluate the performance of our model compared with that as well as with pre-existing Kernels on Kaggle. They have achieved a test accuracy of 85%.[6]

Responsibility Allocation

Shilong Yin: Data generator, model design and test, report and presentation.
Diwei Xiong: Model analysis, background research, report and presentation.

Reference

[1] Wikipedia. White Blood Cells. [Online] Available: https://en.wikipedia.org/wiki/White_blood_cell.

[2] Dhruv Parthasarathy. Classifying White Blood Cells with Deep Learning. [Online] Available: https://blog.athelas.com/classifying-white-blood-cells-with-convolutional-neural-networks-2ca6da239331

[3] Sedat Nazlibilek, Deniz Karacor, Korhan Levent Ertürk, etc. White Blood Cells Classification by SURF Image Matching, PCA and Dendrogram. [Online] Available: http://www.alliedacademies.org/articles/white-blood-cells-classifications-by-surf-image-matching-pca-and-dendrogram.html.

[4] Paul Mooney. Blood Cell Images. [Online] Available: https://www.kaggle.com/paultimothymooney/blood-cells.

[5] Ian Goodfellow and Yoshua Bengio and Aaron Courville. Deep Learning. MIT Press. 2016.

[6] nh4cl. Deep Learning from Scratch + Insights. [Online] Available: https://www.kaggle.com/placidpanda/deep-learning-from-scratch-insights