BREAST CANCER IMAGE CLASSIFICATION WITH TRANSFER LEARNING

CONCEPT

Diagnosing diseases using computers is a quite common application of deep learning methods. In this project, the task is classifying breast histopathology images to two classes, Invasive Ductal Carcinoma (IDC) negative and Invasive Ductal Carcinoma (IDC) positive with using transfer learning.

OBJECTIVES

There are two main objectives of this demonstrated project. One objective is getting high accuracies on classifying both IDC negative classes and IDC positive classes, which is fundamentally main objective of classification models. Other objective that I am interested in is reducing the training dataset and try to understand can we get still good results with not using whole training data.

DATASET

For this project, I used Breast Histopathology Images dataset [1] from Kaggle, which has 2 classes as IDC negative and IDC positive, in training set, it has 178451 IDC negative images, and 70849 IDC positive images. In validation set, it has 9894 IDC negative images and 3956 IDC positive images. In test set, it has 9939 IDC negative images and 3911 IDC positive images. Datasets consist of batches, in total train set got 277 batch with each batch has 900 images, validation and test set got 277 batches with each batch 50 images. All images are 50x50 resolution. In this project, I did not used whole train dataset, but I used whole validation and test set to evaluate the results. One of my objectives is to understand can we still get good results with a subset of a large training dataset.

GENERAL WORKFLOW

For starting to train my model, I need to prepare the training set. The general idea is the following: I will use 1 batch from training set to train the model and evaluate on the whole validation and test set. This will give me a initial idea on the training set, and I will expand the training set according to the results to start getting better accuracies in the end.

DATA PREPROCESSING

The dataset I am using is quite large, so I did not used data augmentation for the dataset. But the images are 50x50 resolution, so I need to resize them to 224x224 to be able to use them in AlexNet backbone. Also, I normalized them to get better results.

MODEL DEFINITION

I used AlexNet as backbone for transfer learning. AlexNet consists of 2 main sequential blocks called features and classifier. In features part, it starts with a convolution with 11x11 kernel with 4x4 strides with 2x2 padding. Then a ReLU applied, it followed with a max pool with kernel size 3, stride 2, 0 padding, with dilation 1. Then another convolution applied with kernel size 5x5, stride 1x1, padding 2x2. Then applied another ReLU followed with the same max pool. Then it decreases the convolution kernel size to 3x3 and continue. Between features and classifier, there is an adaptive average pooling operation. Finally, the classifier consists of dropout layers and linear layers with ReLU, finished with a linear layer to classify into N classes.

HYPERPARAMETER SELECTION

To select the hyperparameters, I used a try-and-see approach. First, I tried learning rate of 0.01, which seems too high for the model, so I adjusted it to 0.001. To loss function selection, I used Cross Entropy Loss, but I saw class imbalance can be a potential issue, so I decided to use weighted version. After 10 epochs, I saw training reached it's potential so I didn't continue training for more epochs, because validation accuracy also stopped improving.

MODEL MODIFICATIONS

For training, I freezed the parameters from the AlexNet, I just changed the last linear layers' output channel number to 2 instead of 1000 because I am doing a binary classification.

MODEL COMPILATION

The model uses Stochastic Gradient Descent (SGD) with Momentum as optimizer with 0.001 learning rate and 0.9 momentum. For loss function, I used Cross Entropy Loss with weights, to manage to handle possible class imbalance issues. Also, I used a learning rate scheduler, in each 7 epoch, with gamma 0.1 for 10 epochs. To evaluate the results of the model I checked total accuracy, and class 0 total accuracy and class 1 total accuracy. Also to get an idea, I calculated precision, recall and F1-score. I calculated confidence interval for the results to understand model results even further.

MODEL TRAINING

First the model trained on 1 batch from original training dataset. Then according to the results, more batches and more class 1 images added to the train set for a new training.

TRAINING WITH 1 BATCH

In this step, I used first batch from the original dataset for training dataset. This batch consists of 837 class 0 images and 63 class 1 images. As it seems from this, in this batch there is a significant class imbalance problem. As a result of that, class 0 accuracy on test set is 100%, but on class 1 is 2%, which shows model did not learn anything. To solve this problem, I tried to use weighted Cross Entropy Loss instead of normal one. This time, class 0 accuracy on test set is 100%, and class 1 accuracy is 12% percent. It increased, but it is obvious that we can not overcome this kind of imbalanced data distribution with just weighted loss function. Therefore, I decided to add some more class 1 data from other batches create a balanced dataset.

TRAINING WITH BALANCED SMALL DATASET

For make the dataset balanced, I add class 1 labeled images from different batches, to make it more distributed, I did not use batches sequentially but also choose batches from the last batches too. Also add 1 full batch which is balanced between class 1 and class 0 to make the model also see some more class 0 images. In this new training set, there are 1279 class 0 images and 1464 class 1 images, which can be said balanced. Even though we solved the class imbalance problem, and we still using weighted Cross Entropy Loss, in the whole test set, class 0 accuracy is 84%, and class 1 accuracy is 54%. This result shows us even though we got more class 1 image in our training dataset, we still did better predict class 0 images. Therefore, I claimed that learning class 1 (IDC positive) images is harder than learning class 0 (IDC negative) images, so the model should see more class 1 images. So, I decided to add more class 1 images to the train dataset, even though it will make the dataset imbalanced again.

TRAININING WITH CLASS 1 DOMINANT DATASET

With the addition from other batches, I ended up with a training set that has 1868 class 0 images and 4446 class 1 images. It is an imbalanced, class 1 dominant training dataset, to make this class imbalance not a big problem, we still use weighted Cross Entropy Loss. But against the imbalanced distribution of classes, on the test set which has got 9939 class 0 images and 3911 class 1 images, model got 88% accuracy on class 0 and 68% accuracy on class 1, on average total 83% accuracy, which is the best result between other training datasets.

EVALUATION & RESULTS

To evaluate the model performance further, I checked the confidence intervals of the 3 classifiers, first classifier is 1 batch classifier, second classifier is class balanced classifier, and third classifier is the class 1 dominant classifier.

CLASSIFIER	CLASS 0 CORRECT	CLASS 0 FALSE	CLASS 0 TOTAL	CLASS 1 CORRECT	CLASS 1 FALSE	CLASS 1 TOTAL
C1	9726	213	9939	477	3434	3911
C2	8372	1567	9939	2155	1756	3911
C3	8790	1149	9939	2644	1267	3911

CLASSIFIER	90% CONFIDENCE LEVEL (CLASS 0)	95% CONFIDENCE LEVEL (CLASS 0)	98% CONFIDENCE LEVEL (CLASS 0)	99% CONFIDENCE LEVEL (CLASS 0)
C1	0.019<error<0.024	0.019<error<0.024	0.018<error<0.025	0.018<error<0.025
C2	0.152<error<0.164	0.150<error<0.165	0.149<error<0.166	0.148<error<0.167
C3	0.110<error<0.121	0.109<error<0.122	0.108<error<0.123	0.107<error<0.124

CLASSIFIER	90% CONFIDENCE LEVEL (CLASS 1)	95% CONFIDENCE LEVEL (CLASS 1)	98% CONFIDENCE LEVEL (CLASS 1)	99% CONFIDENCE LEVEL (CLASS 1)
C1	0.869<error<0.887	0.868<error<0.888	0.866<error<0.890	0.865<error<0.891
C2	0.436<error<0.462	0.433<error<0.465	0.430<error<0.468	0.428<error<0.470
C3	0.312<error<0.336	0.309<error<0.339	0.307<error<0.341	0.305<error<0.343

From confidence intervals of 3 classifiers, both classifiers tend to make mistakes on class 1, and I got quite improvement on class 1 results with not losing much accuracy on class 0 with classifier 3 comparing to other classifiers. When we consider the main objectives of this demonstrated work, we can see that with a subset of the main training dataset, we can still have good accuracies, which touches an important problem of model training, using a lot of data is not an intelligent approach to handle the problems.

FUTURE WORK

Then main objectives of this demonstrated work is to investigate whether can we use a subset of training dataset to get good results. In this work, we focused on selecting random subsets from the main dataset and evaluating those results and continue improving the training dataset randomly additions. An interesting approach to investigate in future can be understanding the importance of the data, so we can select the data additions in a more advance manner instead of random additions. There are some works on active learning field that investigate the importance of different data and add them training set accordingly. This can be important especially in medical image field, because getting annotations is hard and time consuming, so before getting annotations for images, we can check which images can be more informative for the task and ask for annotations of those images. If the results are not good enough, we can ask for least informative image annotations to reach the better results, which can help us to work on smaller datasets without losing information, that will help us to reduce the complexity of computations and save resources.

CONCLUSION

In conclusion, the demonstrated work shows that to get good results, we do not need to use whole training dataset, which we will appreciate if the dataset is too large. This work shows even with random selection of additions to the dataset, we can get good results with just ~6k training image, we can get 82% test accuracy on ~14k test dataset. To further improve the performance, there can be defined a informativeness metric to select images for training set more intelligently, which can even further improve the results and may result in using even less training images. Especially in medical image field, getting precise annotations is challenging and time consuming, so this can be quite helpful.

REFERENCES

[1] https://www.kaggle.com/datasets/paultimothymooney/breast-histopathology-images

Berk Coşar 69557