DATS_6203

Machine Learning II

Group Report

Estimation of Breast Cancer Density in Hematoxilin-Eosin slides using a Convolution deep neural network

The main goal of the project is the estimation of cellularity in extracts of hematoxylin- Eosin (H&E) slides from 64 breast cancer patients. The number of cancer cells in a tumor sample plays an important role in determining how aggressive a tumor is alongside other indicators both clinically and pathologically. In turn this would determine the clinical management of patients with cancer and their ability to receive neoadjuvant chemotherapy and subsequent need for radiation therapy. Thus automating the process of cellularity estimation would facilitate the reading of pathology slides for cancer patients and may help reduce the inter pathologist variation.

The project uses a dataset which is available through the SPIE (international society for optics and photonics) 2018 challenge. The challenge consists of estimating the cellularity of a portion of a slide and submitting the model for review and consideration for publication. The dataset consists of 2570 images (512*512 px) in .tif format. They are divided into a train and test data. They each have a label attached to them which is a number between 0 and 1.

Loading the data was the hardest part given the format in which the data was provided as well as some technical problems with my PyCharm. After these were fixed a combination of code from multiple sources finally yielded a valid dataset which was confirmed to be correct. Then a data loader was used to put in the model. This part of the project was inspired by a Kaggle challenge which we were considering of using as our dataset. It provided multiple kernels for loading a .tif images and transforming the folders appropriately into usable paths. (https://www.kaggle.com/c/histopathologic-cancer-detection). The dataset had to be combined into one as we needed as many slides as possible for training and then split into a testing and a training datasets. This was done by using a skitlearn train/test split function.

An exploratory data analysis was also performed with a simple visualization of the images (a random sample of 10) and a histogram of the different labels.

The general algorithm developed was a CNN which would be a regression algorithm rather than a classification algorithm. This is a difficult task given that the dataset is small and will need to be augmented for valid and meaningful results to be obtained. Thus the first step was to create a set of image transformations and augment the dataset.

Second, the output layer activation function was left as a linear function, but the loss function had to be modified and both the MSE loss function and the KLDiv functions were used. Both are designed for regression problems.

The optimizer choice was Adam. Other optimizers were later tried and did not yield an appropriate RMSE.

The model design was a group effort and we decided to do a few different models. The base model was a 4 layer CNN, with two convolutional layers as well as an ReLU and a final Linear layer. The code was adapted from the example provided in class.

The testing of the model was also modified and included the calculation of the RMSE, again using a skit learn function. These modifications were carried out by me. This served as a model evaluation for the project.

Also for the completion of the initial code, I added exploratory analyses. Namely a display of the images initially, to make sure the images are being read correctly, as well as a histogram of the labels to see the distribution of the labels for the dataset.

After the model trained and the testing showed an RMSE of 0.23. I then proposed we use this on a deeper network. Some reading pointed me to using a custom made resnet9 model which was adapted from (https://www.kaggle.com/sermakarevich/complete-handcrafted-pipeline-in-pytorch-resnet9/data?scriptVersionId=10694803). The model was modified and then a regression layer added and run. The RMSE was better than the prior model.

Further I ran a Densenet121 model for this data. This was not a brilliant move, as densenet seem to be a much better adapted model for classification problems. However the model was modified to include a linear layer suited for regression and the model was tried. The loss function had to be modeified to a cross entropy loss to get the model running. The RMSE was acceptable at 0.12, however, the model is not suitable for use without significant modification and a better understanding of its weights and biases distribution for an RMSE loss function to be used.

For the first two models, only epoch=5 was used given the time constraints for the project. The loss function stabilized after 2 epochs and no major variations were noted. KLDiv provided an RMSE of 0 or a very very low RMSE consistently.

Model	Loss function	Epochs	RMSE
CNN - 8	MSE	5	0.23
CNN - 8	KLDiv	5	0.21
CNN - 8	MSE	10	0.01
Resnet - 9	MSE	5	0.01
Resnet - 9	KLDiv	5	0.47
Densenet - 121	Cross entropy	5	0.26

The loss functions were all plotted and examples are included here. All of the loss functions are summarized in the main group report.

Fig1: Loss function for CNNKLDiv

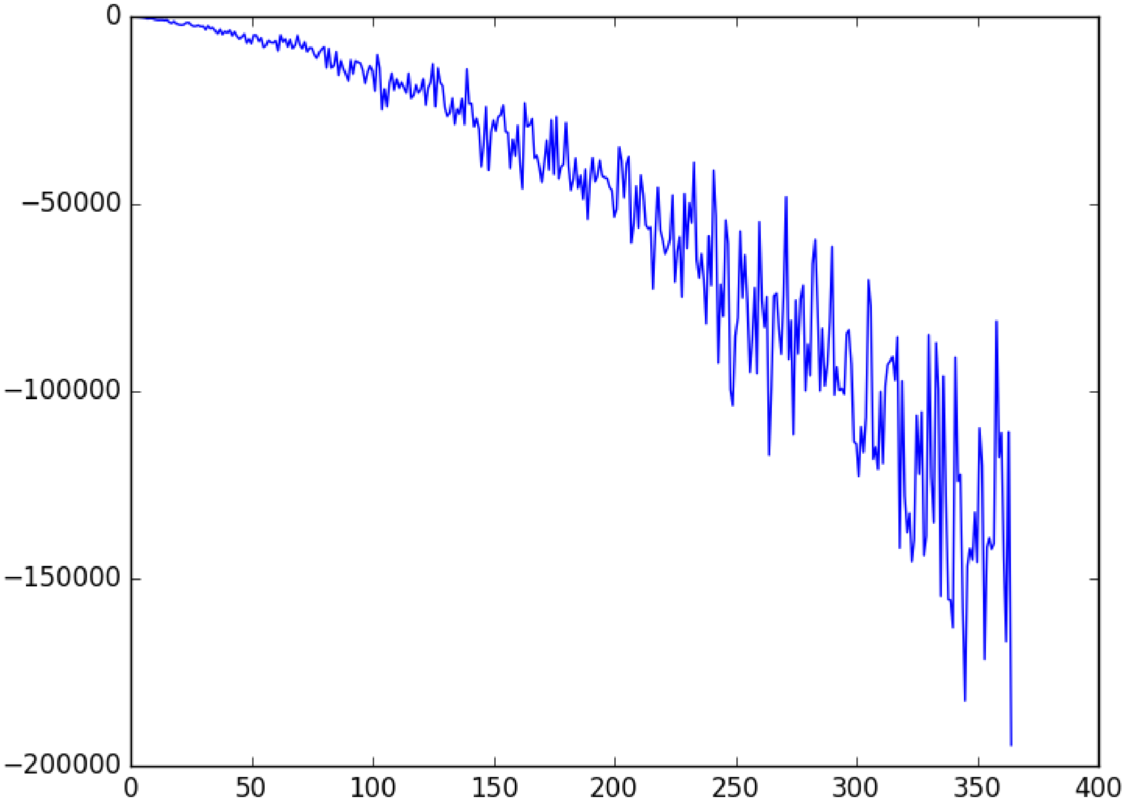


Fig2: Loss function for CNN MSE

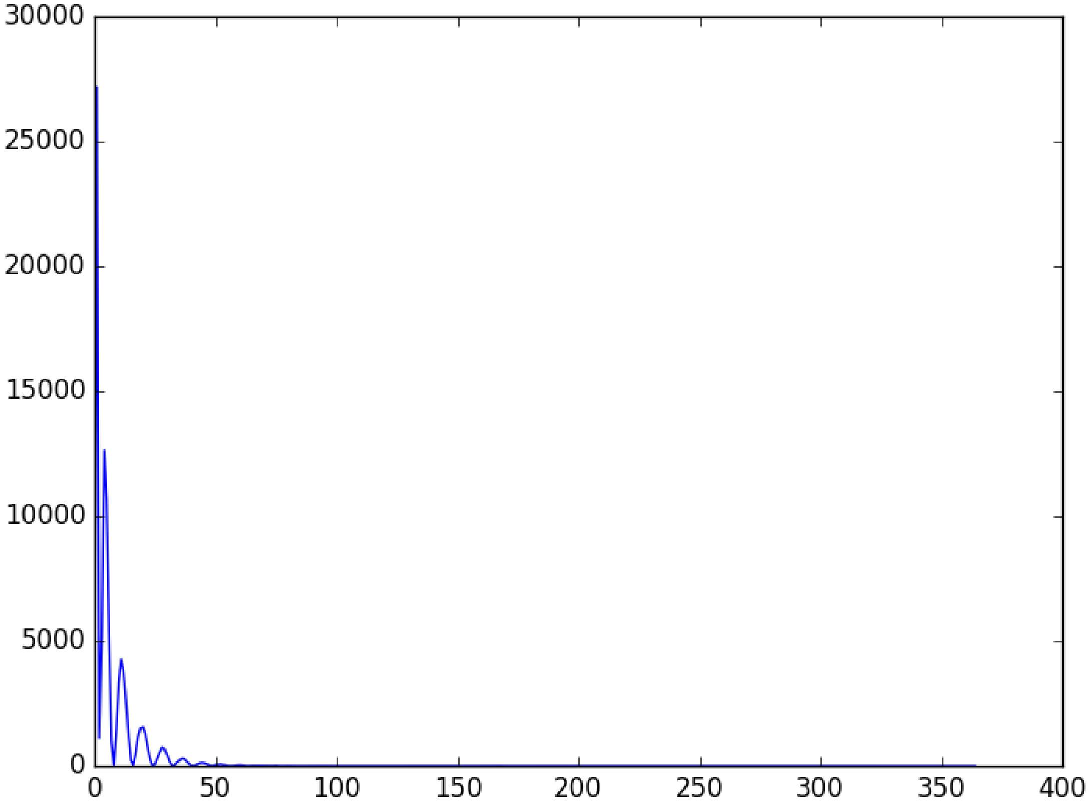
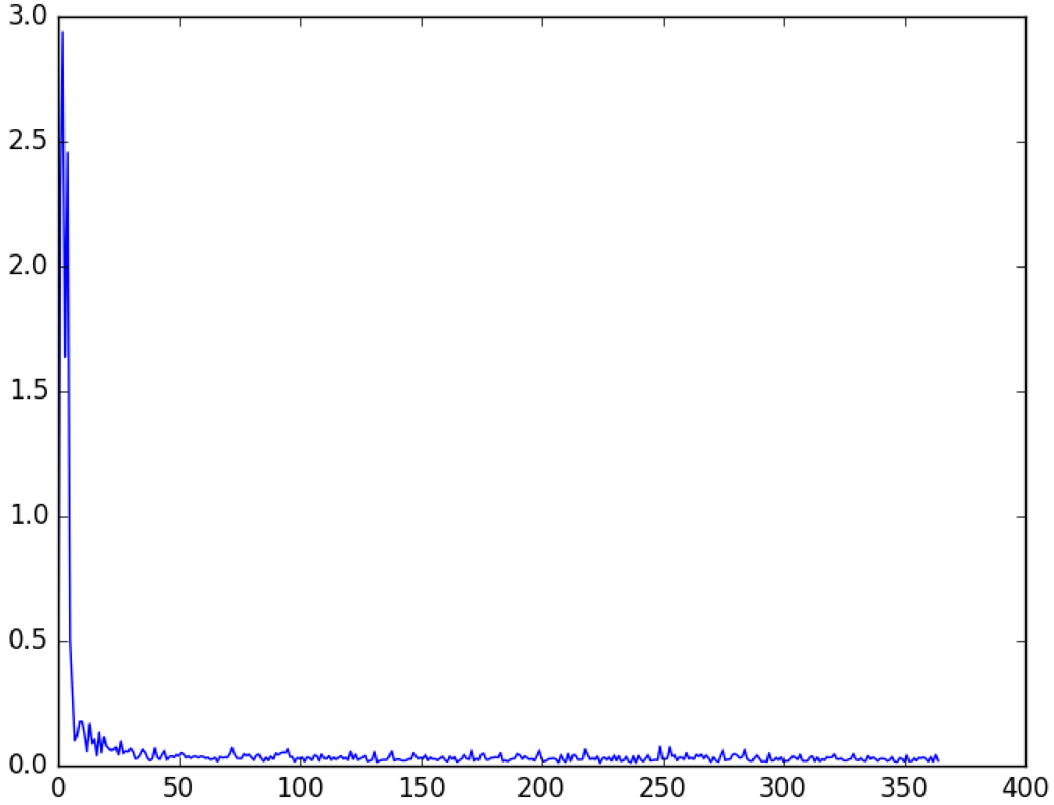


Fig3: Loss function for resnet9-MSE



In summary, a CNN model is likely to be a valuable automated tool for computer vision in the field of cancer histopathology. The use of a handcrafterd residual convolution network seem to yield consistent and excellent results and this would be the model to be further developed. This part was designed by me and executed by my colleague and will be attached to the group project.

However, several limitations are seen here. First the dataset is small for a consistent training of a regression algorithm. Further data transformations are needed to augment the data. In addition, the literature suggests the use of larger dilated CNNs to circumvent some of the limitation posed by the use of a smaller dataset in preventing overfitting.

The approximate amount of copied code is about 60-65%. Multiple sources of code were used to aid in the making of the models and the metrics used.

References

http://spiechallenges.cloudapp.net/competitions/14#learn_the_details-overview

https://www.kaggle.com/sermakarevich/complete-handcrafted-pipeline-in-pytorch-resnet9/data?scriptVersionId=10694803

https://www.kaggle.com/artgor/simple-eda-and-model-in-pytorch

https://www.kaggle.com/soumya044/histopathologic-cancer-detection

https://www.kaggle.com/c/histopathologic-cancer-detection/kernels

https://www.kaggle.com/eiffelwong1/basic-cnn-for-cancer-detection-pytorch

http://cs231n.stanford.edu/reports/2017/pdfs/203.pdf

Google.com, stackskills