Indian Institute of Technology, Delhi

ELL-888

Advanced Machine Learning

Submitted To:
Dr. A P Prathosh
Department of Electrical Engineering.

Submitted By :
Pushpendra Singh Dahiya(2017EET2680)
Ajinkya Purushottam Wasnik(2018EET2569)

Contents

1 Introduction

1.1 Problem Statement

In this assignment we were required to design a binary classifier to distinguish between the normal and abnormal brain images.

1.2 About Problem

Binary or binomial classification is the task of classifying the elements of a given set into two groups(predicting which group each one belongs to) on the basis of a classification rule.

Medical testing is to be done if a patient has certain tumor or not, the classification property is the presence of the tumor.

BraTS utilizes multi−institutional pre−operative MRI scans and focuses on the segmentation of intrinsically heterogeneous(in appearance, shape and histology) brain tumors, namely gliomas.

BraTS challenge is originally setup for deducing type of tumor, so it mostly have tumorous data which we'll be training on so we would have to get some different data from some source for training the model.

Since its an image classification problem we'll be using CNNs for training the model.

2 Dataset

2.1 About dataset

The image dataset given was BraTS(BRAin Tumor image Segmentation) dataset which had the following four MRI contrasts:

```
i) T1: T1-weighted, native img, sagittal or axial 2D acquisitions
    with 1-6 mm slice thickness.
ii) T1c: T1-weighted, contrast-enhanced image, with 3D
    acquisition and 1 mm isotropic voxel size for most patients.
iii)T2: T2-weighted,axial 2D acq, with 2-6mm slice thickness.
iv) FLAIR:T2-weighted FLAIR image, axial, coronal, or sagittal 2D
    acquisitions, 2to6 mm slice thickness.
```

All training is done on T1. Alongwith these we also had the corresponding ground truth for all images.Total 274 patients dataset was available in which each person brain's MRI had 155 more slices image.For our training we have taken the Axial view which is 240x240x155.

2.2 Dataset Extraction

We had all 4 MRI images flavours alongwith corresponding Ground Truth file in three named file.
The images were stored in ".mha" format as signed 16-bit integers, out of which positive values were required to be used. SimpleITK package is used to read these image data.The training data also comprises of ground truth in a separate .mha file.

2.3 Preprocessing Training Dataset

The BraTS data given is having skewness so training wouldn't be any better if we train on original data, so we augmented data in different ways(Described in next
Part).
Basically while we firstly create NumPy arrays that contain only zeros we provide it with the number of slices(SN) we would we using like

```
        X_train = np.zeros((274*SN, 240,240), dtype = np.uint16)
  &     Y_train = np.zeros((274*SN, 2), dtype = np.uint16)
```

and joined the image T1 and corresponding ground truth(GT) in a dictionary in python.
Also the GT value is taken as either 1/0 if sum of all pixels in GT>0 in that particular range then 1 otherwise 0.

2.4 Data Augmentation

Augmentation of data was also one of the essentials in the process as the data was highly skewed, hence would have trained model with a bias towards tumored class of MRIs. So we fed data in three ways to our NN and shifted to next augmentation way because of issues with previous method.

>Firstly, we run on basic dataset i.e. BraTS dataset(the data which was highly skewed). So it was bias towards one class so we had to move towards different way.

>Secondly, we took the healthy brain images and the duplicated all for certain count so that skewness is removed and then trained the resulting dataset by shuffling up images.(See fig4)

>Thirdly, we took the dataset of healthy brain from web and then augmented that with tumorous images of MRI scans we got for training.(This involved downloading images, then using tool(runs typically on centOS) namely FSL and then took out image from .nii.tar.gz & applied that FSL tool for removing skull and eyes parts) and recorded corresponding accuracy.

2.5 Test Dataset

We used two versions of test data:

> Firstly, the original one which sir provided which was a different data altogether from different MRI machine and with noise like skull, ears, eyes, etc which therefore returned not much good test accuracy.

> Secondly, so we had to remove those artefacts for which we tried some matlab code for image segmentation in which we did opening operation and then found connected component of resulting image obtained after opening whose accuracy of segmentation was not too good(nearly 60). But still was better than original data set as it was more free from noise and similar to what we have given as training dataset for healthy Brain images in ".nii" files using FSL tool.

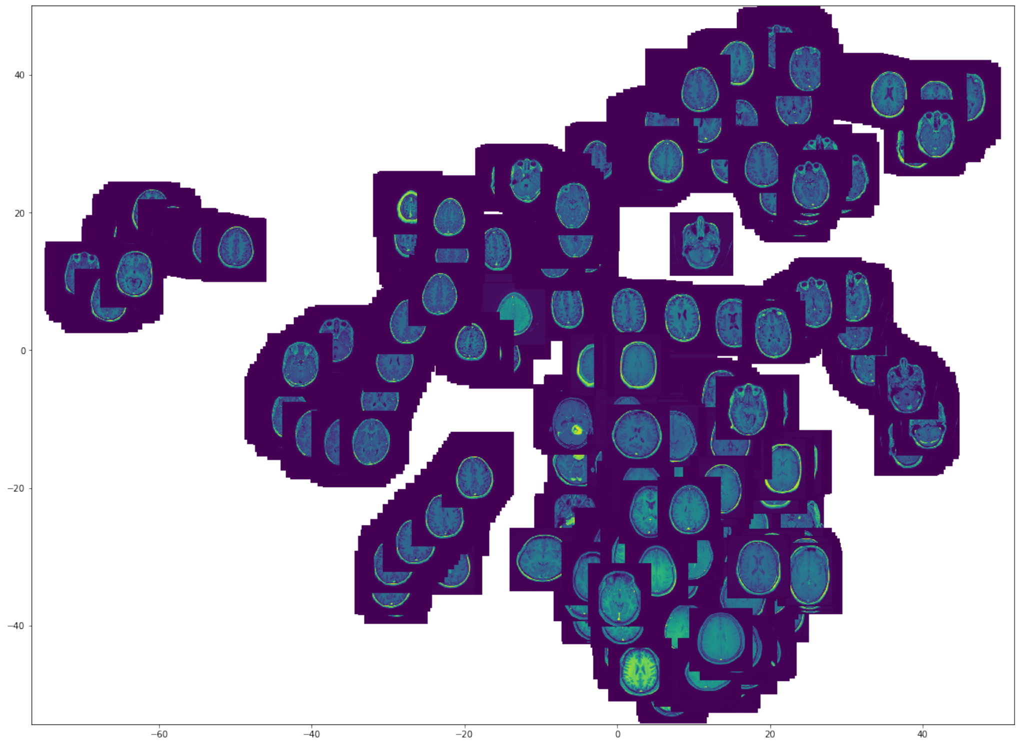

fig 1: T-SNE done on sir's testdata & on examination, the classes were found to be clustering into multiple clusters which we need to separate by a multidimensional hyperplane.

3 Model

3.1 Introduction

We had a task in hand where we didn't had standard dataset for test while training data was also skewed so we felt it was better to try a model for classification which is standard enough so we've used AlexNet. But as the original model was for 1000 class classification we've changed it for binary classification.

>We've Relu instead of Tanh to add non-linearity. It accelerates the speed by 6 times at the same accuracy.

>We've used dropout as regularisation to deal with over-fitting. However the training time is doubled with the dropout rate of 0.5.

3.2 Different models trained

i) AlexNet
ii) UNet

3.3 Experimentation on Model and Datasets

Experimenting with different Architectures:

We tried different methodologies for the task at hand.
Two class classification using CNN:
i) Simple 3 stage CNN with ReLU activation & binary cross entropy loss
ii) Alexnet with ReLU activations and Softmax at the final layer and binary cross entropy loss

*Semantic segmentation using CNN:
-> Unet architecture for segmentation of the tumorous portion from the rest.

Experimenting with Data:
Since the training data was biased towards one class, Any classifier learnt could have learnt this bias and always output one class.

To solve this problem following approaches were take:
i) For the Classifiers different class weights were given to the two classes so that the loss calculations is not biased.
ii) Additional data was gathered from the internet and added to the BraTS data.
iii) Test were also performed on the Sir given dataset to see how the architecture performs on the combined data.

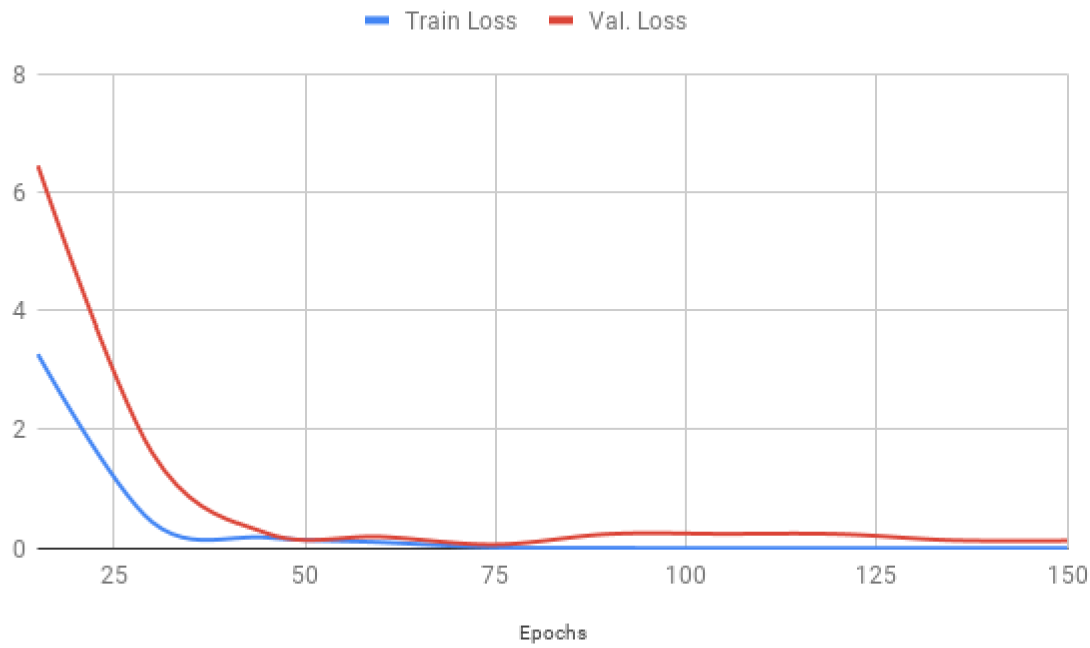

fig2: Training loss and Validation Loss w.r.t epochs. Model with least validation loss is selected

4 Results

Training on BraTS Data		
Data Distribution	Loss	Accuracy
Training (80%)	0.0612	0.9688
Validation (20%)	1.5625	0.6684
Test (Sir's Data)	1.9171	0.56263

Metrics	No Tumor (Class 0)	Tumor (Class 1)
Precision	0.562632696	0.54363266
Recall	0.563	0.553
F1-score	0.56	0.55

fig 3: Different parameters and metrics values when trained on BraTS dataset.

Training on Sir's Data		
Data Distribution	Loss	Accuracy
Training (80%)	0.0302	0.9855
validation (10%)	0.051	0.9882
Test (10%)	4.203026543	0.64

Metrics	No Tumor (Class 0)	Tumor (Class 1)
Precision	0.6451612903	0.6261682243
Recall	0.6	0.67
F1-score	0.621761658	0.6473429952

fig 4: Different parameters and metrics values when trained on Sir's test data.

Training on BraTS Data + Normal Brain Images from other datasets + Sir's Data

	Loss	Accuracy
Training (80%)	0.2035	0.9219
Validation (20%)	0.4709	0.7989
Test (Sir's Data)	1.29229	0.584755

Metrics	Tumor (Class 0)	Tumor (Class 1)
Precision	0.58663	0.56963
Recall	0.57	0.56
F1-score	0.574	0.562

fig 5: Different parameters and metrics values when trained on mixed dataset.

5 Conclusion

In this assignment we were required to design a binary classifier to distinguish between normal and abnormal brain images.

The learning from the task was immense and two fold, first we learnt to make 2 class classifiers using CNN, second we learnt how to deal with biased datasets, which is generally found in any real world problem.

We tried different techniques including making changes in architecture and changes in dataset to deal with the skewness. We tried Semantic segmentation, but the method was not found useful in the current classification scenario, we can probably use it in the coming assignments.