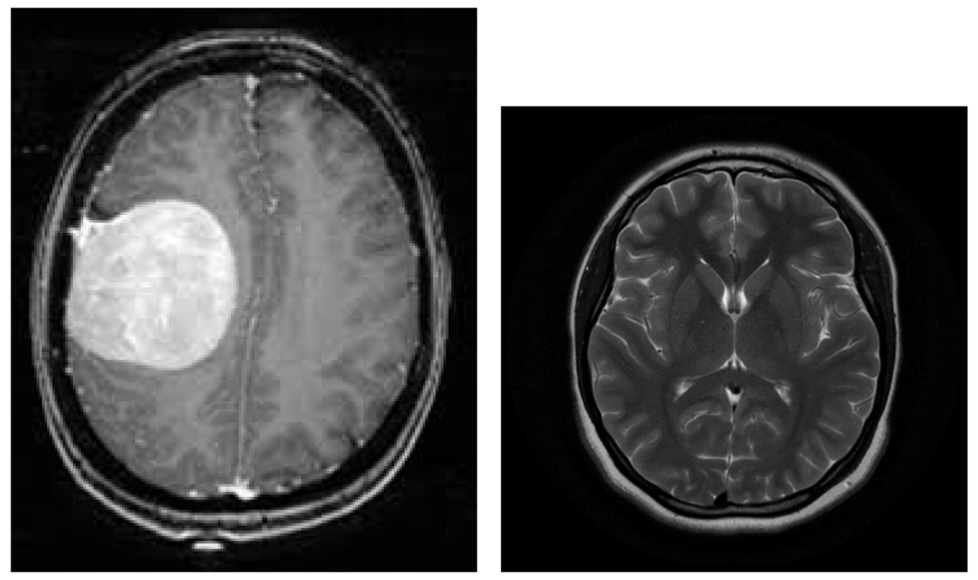
Autoencoders for Unsupervised Learning

Rudraharsh Tewary

University of Toronto

Background



(a) Tumor (b) No Tumor

Figure 1. Two Brain MRI Images

We wish to tackle the problem of Unsupervised learning in the case of brain MRI classification. Specifically concerning the case of MRI images containing tumours or the absence of one.

Challenges Encountered

The task presented here seems simple, however pulling it off requires us to execute a rather complicated approach. Here are a few challenges we would face with this task

- MRI images are high dimensional and due to the sensitivity of the task, dimensionality reduction may not be appropriate.
- Due to significant variability between brains of different people simple models may not be able to generalize well.
- Also, due to a lower population of positive images. A more sophisticated model is required for deeper feature extraction.

Limitations of classical approaches

Here, we present the main motivation to adopt a deep learning based approach for the task at hand. We provide the limitations of classical algorithms like clustering here

- High dimensionality:- the images at hand may contain thousands of features. This limits the usability of methods like k-means due to the curse of dimensionality
- Non-Linearity :- The relationships between pixel and pixel densities in the images might be non-linear. Which cannot be captured well by classic clustering algorithms
- Heterogenous Data :- as discussed above, there will be a lot of variation in already complex image data across observations. This adds an additional insurmountable challenge for classic methods

Now, following this we will define the approach we adpoted

Method :- Autoencoder

Let us introduce the neural network powered form of unsupervised learning. The Autoencoder architecture is comprised of an encoder and decoder which are explained below :-

1. Encoder :- The encoder is composed of fully-connected or convolutional layers. Whose purpose is to create a lower dimensional representation of the input data.
2. Latent Space :- The latent space is a compressed representation of the data, created by the encoder it represents similar points grouped together.
3. Decoder :- Its purpose is to take the encoded representations from the latent space, and construct a replicate of the input which is as close as possible.

Convolutional Autoencoder

As we are dealing with image data, we have to employ the use of convolutional autoencoders. This architecture replaces the fully connected layers in traditional autoencoders with convolutional layers. The benefit of this method is its ability to detect spatial hierarchies and extract local patterns[1].

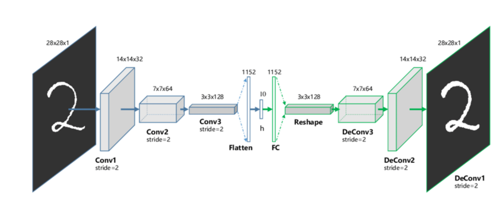
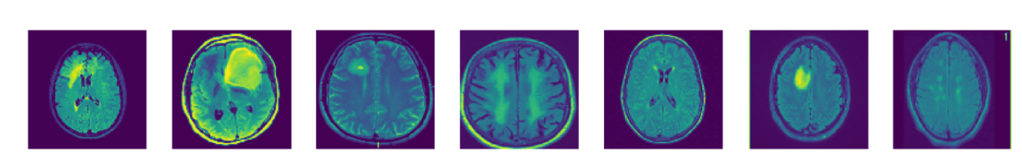


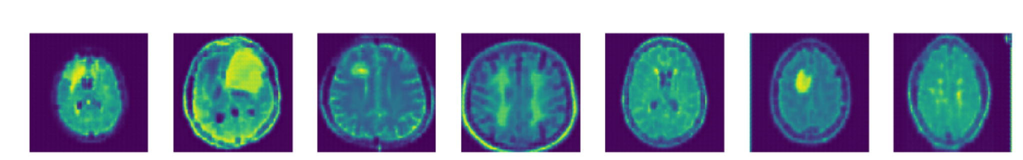
Figure 2. Convolutional Autoencoder architecture [2]

Parameters and Results

We employ a 2-layer Convolutional Autoencoder, trained on a T4-GPU, with starting latent dimensions of 2048 and training time of 50 epochs.



(a) Original Images



(b) Autoencoder Reconstruction

One can visually analyze the Performance of our Autoencoder by comparing the quality of the original and reconstructed images as presented here. As we can see, even for a simple implementation like ours the reconstruction quality is decently high and close enough to the original. Which tells us that the network was able to decently learn deep features present in the dataset

Training regime

A key factor in the network is the loss function. In our case we use the Mean Square Error (MSE) to optimize the model

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N} \|x_i - \hat{x}_i\|^2$$

Here, x_i represents the input image, while $\hat{x}_i$ represents the reconstructed output from the decoder layer. The main constraint for training is the network minimizing the MSE for our defined training hyperparameters.

Results with Classification

So, to evaluate how well our method was able to extract deep features from the dataset. We compare the performance of random forest for classification in two cases, one where it receives normal training data. The other where it receives an encoded representation of the training data from the Autoencoder.

Test Data Size	Model without Autoencoder	Model with Autoencoder
20%	82%	85%
30%	82%	82%
40%	79%	81%
50%	74%	81%

Table 1. Comparison of accuracy for Random Forest and Random Forest with Autoencoder on different test data sizes.

We can see a very interesting result. Our classifier shows equal performance in the case of normal train/test splits. However, when we employ higher test data sizes and make the training dataset smaller. The model using encoded representations retains most of its accuracy. Signalling to us that its highly usable when validation data is sparse.

References

[1] Min Chen, Xiaobo Shi, Yin Zhang, Di Wu, and Mohsen Guizani. Deep feature learning for medical image analysis with convolutional autoencoder neural network. *IEEE Transactions on Big Data*, 7:750–758, 2017.

[2] Xifeng Guo, Xinwang Liu, En Zhu, and Jianping Yin. Deep clustering with convolutional autoencoders. pages 373–382, 10 2017.