Brain Tumor MRI Classification with Vision Transformers (ViT)

Olivia Lee

Boston University MET Computer Science Department

CS 767 Advanced Machine Learning Neural Networks

Term Project

Spring 2023

INTRODUCTION

The integration of machine learning in the healthcare industry will revolutionize patient care and treatment plans. By using machine learning algorithms to read and classify scans, we can significantly improve our ability to assist physicians in making diagnoses, and aim to mitigate the potential human error and bias.

This project demonstrates the powerful capabilities of Vision Transformers in the field of medical image analysis, specifically in the classification of brain MRI scans as either tumor or no tumor. We will cover concepts such as data augmentation, patches, attention and transformer encoders. Hyperparameter tuning is also an essential part of building machine learning models, so we will evaluate models with various learning rate schedulers and optimizers to determine the best model.

THE DATASET

The dataset used for this model consists of JPEG image files of brain MRI scans in grayscale. There are a total of 3000 images, comprising 1500 scans with brain tumors and 1500 scans without brain tumors.

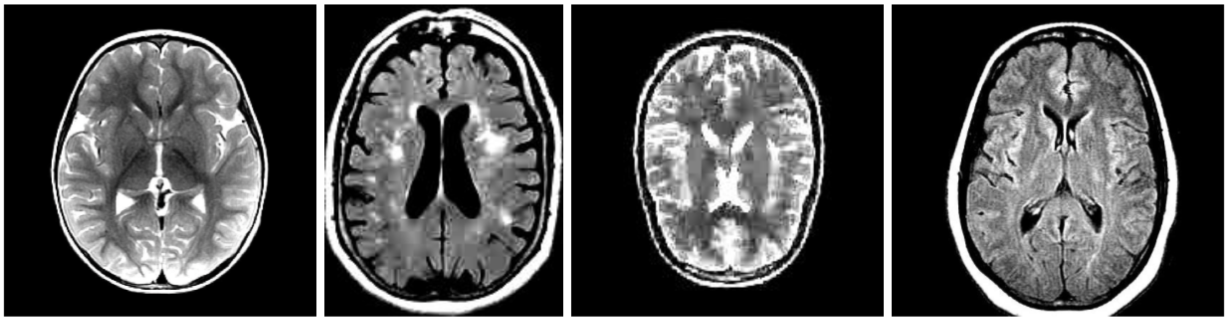

Figure 1: Sample brain MRI scans with no tumor

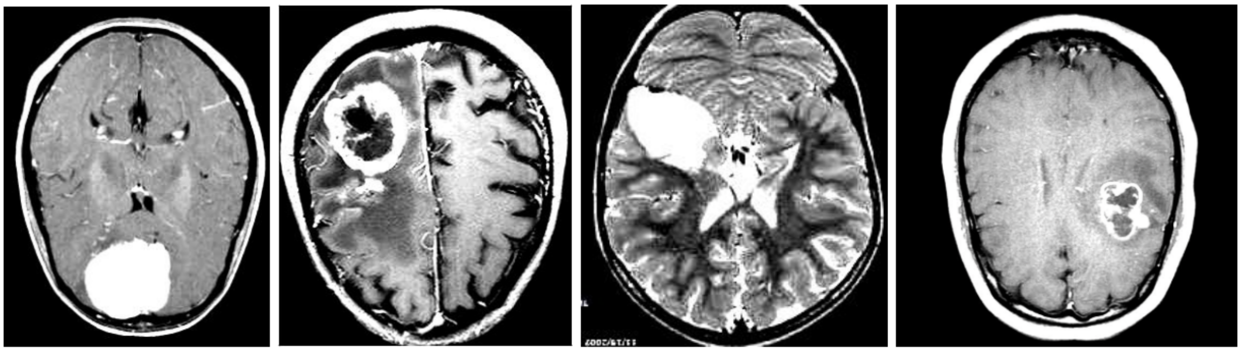

Figure 2: Sample brain MRI scans with tumor

DATA PREPARATION

1. Reading images: This will convert the images into arrays with size (x, y, 1) with x and y representing the dimensions of the photo and 1 representing the grayscale channel.
2. Resizing images: All images are resized into a square image with the dimensions 128x128.
3. Creating feature matrix, X: The feature matrix is read MRI scans and will have the shape (3000, 128, 128, 1) with 3000 representing the number of images.
4. Creating label vector, y: Simultaneously, a label vector is created that indicates 1 for images with tumor and 0 for images without tumor.
5. Splitting data: Data is split into 60%, 20% and 20%, for the training set, test set and validation set respectively.

THE MODEL: VISION TRANSFORMER

Hyperparameter Selection

Initial hyperparameters are defined in this section. The hyperparameters are:

1. Batch size
2. Number of epochs
3. Resize shape for data augmentation

4. Patch size
5. Number of patches
6. Projection dimensions for transformer units
7. Number of heads for attention
8. Transformer units
9. Number of transformer layers
10. Size of dense layers of final classifier

Later in this project, hyperparameter tuning will be done to optimize the model's accuracy.

Data Augmentation

Data augmentation is a technique in computer vision to artificially increase the size and diversity of a dataset. By applying various transformations to the original images, data augmentation can generate new variations of the images that will improve the accuracy of the model by preventing overfitting. In this model, the data augmentation layer will normalize the pixels, resize the images, and randomly flip horizontally, rotate, and zoom in on the images.

Feed Forward

Feed forward neural networks flow information in one direction. To prevent overfitting, ReLU (Rectified Linear Unit) is used as an activation function for each dense layer. Another technique implemented to prevent overfitting is dropout, which randomly drops out a portion of the neurons during training. In this project, a dropout rate of 0.1 is chosen. Each hidden layer is followed by a dropout layer.

Image Patching

In image processing and computer vision, a patch refers to a small region or subset of an image of a fixed size. Each image is divided into patches, and each patch is processed separately. They are put into a patch encoder to extract lower-level features from the image. When the patches are passed through the transformer encoder layers, it allows the model to discover complex relationships between the patches and learn to classify the images. By breaking the images down into smaller, more manageable sizes, we can build a more efficient model that is able to learn from large and complex datasets.

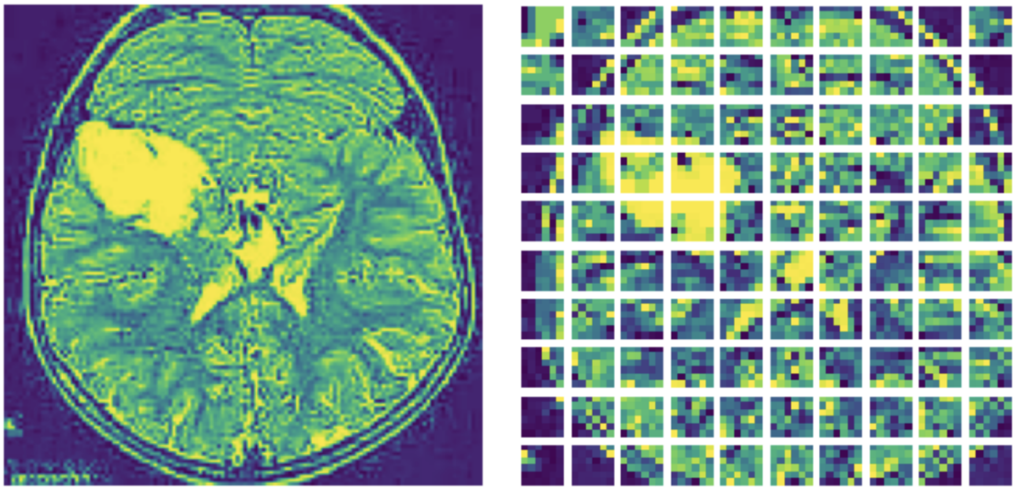

Figure 3: Visualization of image patching

Vision Transformer Architecture

1. Input layer: This is the layer of input data with an input shape of (128, 128, 1).
2. Data augmentation layer: Data augmentation is performed in preparation of image patching.
3. Image patching layer: Images are split into patches.
4. Patch encoding layer: Each patch is encoded to be passed through the transformer encoder.
5. Transformer block: Each transformation layer will consist of the following layers:
    a. Normalization layer 1
    b. Multi-head attention layer
    c. Skip connection layer 1
    d. Normalization layer 2
    e. Feed forward layers
    f. Skip connection layer 2

6. Creating a [batch size, projection dimensions] tensor: This creates output representation.
    a. Normalization layer
    b. Flattening layer
    c. Dropout layer
7. Feed forward layers: This creates a feature layer to be used for the final layer.
8. Logits layer: Probabilities for each class are defined in this layer.

MODEL TRAINING

Components

1. Loss function: Sparse categorical cross entropy is used for the loss function.
2. Metrics: Sparse categorical accuracy is used to observe metrics.
3. Checkpoint: This saves the weights of the neurons.
4. Early stopping: An overfitting technique that monitors validation loss and terminates model training after 5 epochs without improvement in its performance on the validation set.
5. Fitting the model: Uses training data to fit the model with an initial number of epochs of 50.
6. Loading weights: Returns final weights of the model to be used for testing.
7. Model evaluation: Evaluates model on test set with loss and accuracy.

HYPERPARAMETER TUNING

Learning Rate Scheduler

A step decay learning rate scheduler is used to tune the learning rate of the model during training. In this case, the first ten epochs will use the initial learning rate, and following epochs will decrease the preceding learning rate by a factor of 0.1.

Exponential decay and cosine decay learning rate schedulers are used in the optimizers. We will use both types of learning rate schedulers to determine the best to use for our model.

Optimizers

Three optimizers will be used in this model: Adam, SGD (Stochastic Gradient Descent) and RMSProp.

From these three optimizers and two learning rate schedulers, six combinations of optimizers are created. All six optimizers are used for model compilation and each model will be assessed to determine the best learning rate scheduler and optimizer for our dataset.

1. Adam optimizer with exponential decay
2. Adam optimizer with cosine decay
3. SGD optimizer with exponential decay
4. SGD optimizer with cosine decay
5. RMSProp optimizer with exponential decay
6. RMSProp optimizer with cosine decay

PERFORMANCE

Optimizer	**Learning Rate Scheduler**	**Test Loss**	**Test Accuracy**
Adam	Exponential Decay	0.39	81.17%
Adam	Cosine Decay	0.39	83.17%
SGD	Exponential Decay	0.58	70.5%
SGD	Cosine Decay	0.64	62.5%
RMSProp	Exponential Decay	0.33	88.83%
RMSProp	Cosine Decay	0.39	83.0%

Table 1: Performance summary of each model

BEST MODEL

The best model uses RMSProp optimizer with exponential decay learning rate scheduler, with a test loss of 0.33 and a test accuracy of 88.83%.

CONCLUSION

This project created a Vision Transformer model that successfully classified brain MRI scans into tumor or no tumor. The model utilized overfitting prevention techniques such as data augmentation and early stopping, and implemented image patching and transformer encoders to build the ViT. Different learning rate schedulers and optimizers were used in hyperparameter tuning with the best model with a test accuracy of 88.83%. Although transformers are more widely used for natural language processing purposes, we have seen the power of Vision Transformers in image processing, and in this case image classification.