Technical Report

By: Victor Afonso Teixeira Santos 108212, Gabriel Moreira Marques 108207

Introduction

The paper *AN IMAGE IS WORTH 16X16 WORDS:TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE* introduces a new usage of the transformer architecture, the ViT(Vision Transformer) architecture. Traditionally, convolutional neural networks (CNNs) have been the main architecture used for image recognition. The authors seek to demonstrate that Transformer-based models can outperform CNNs. Furthermore, they investigate whether Transformers can be adapted and scaled for image recognition tasks by treating images in a similar way to how words are handled in NLP, specifically by dividing them into patches. The paper utilized the ImageNet-21k dataset for pretraining and fine-tuned on ImageNet-1k for evaluation.

The second paper, *Facial Expression Recognition with Visual Transformers and Attentional Selective Fusion* uses the ViT on the task of expression recognition. Capturing nuanced and spatially distributed features can be a hard task, and this paper succeeds in showing how well the ViT with attentional selective fusion mechanisms works well for such usage. It used datasets such as RAF-DB, AffectNet, and FER2013 and was able to show improvements in comparison in their usage of the datasets.

Main differences between our project and the authors implementation

The original authors implemented ViT in JAX , while this project uses PyTorch. The code simplifies certain aspects, like potentially reducing the number of transformer encoder layers and modifying the hyperparameters (e.g., patch size, embedding dimension) to suit the Tiny ImageNet dataset and limited computational resources. Additional preprocessing steps, such as organizing Tiny ImageNet validation data, are custom and not present in the original implementation.

Implementation

The VisionTransformer code implements a Vision Transformer (ViT) using the Tiny ImageNet dataset. It mixes both PyTorch components and custom-designed classes to adapt the ViT architecture.

The data preparation stage includes organizing the dataset and applying preprocessing using transforms, such as resizing images to 64x64, normalizing them, and converting them into tensors. The dataset is then loaded into PyTorch DataLoaders, allowing efficient batch processing during training and validation.

The ViT model starts by dividing images into small patches with the PatcHEmbbeding class, which uses a convolutional layer to create feature embeddings for each patch. The PositionalEmbedding adds learnable positional information so the model understands the order of the patches. These steps are combined in the VisionTransformerInput module, which prepares the input for the Transformer.

The main part of the model is the TransformerEnconderBlock. It uses multi-head self-attention to find relationships between patches and a feedforward network to process features, with residual connections to make the learning better. Multiple encoder blocks are attatched to create deeper representations. The model also adds a special classification token to focus on the final output and uses an MLP head to make predictions.

For training, the code defines functions to handle one epoch of training or validation. It uses a cross-entropy loss function and an AdamW optimizer to update the model's weights. A cosine scheduler adjusts the learning rate during training. The train_and_save function runs multiple training epochs, tracks performance, saves checkpoints, and plots training progress.

This version of ViT simplifies the original by reducing the number of layers and other parameters to make it suitable for the dataset and available computational resources. It combines custom-written parts, like the patch handling, with PyTorch components to efficiently implement the model.

The fine tuning code uses the technique in a Vision Transformer (ViT) model to classify facial expressions using the FER2013 dataset.

This code fine-tunes a Vision Transformer model for recognizing facial expressions using the FER2013 dataset. It starts by preparing the dataset, resizing images to 224x224 pixels, normalizing them, and splitting the data into training and validation sets. These sets are loaded in batches using PyTorch dataloaders to ensure efficient processing. The pre-trained ViT model is loaded and modified to classify seven facial expressions by setting the number of output labels to match the dataset. The training process uses cross-entropy loss to measure prediction errors and the AdamW optimizer to adjust the model's weights. The training runs over multiple epochs, where the model alternates between learning from the training set and validating its performance on the validation set. During each epoch, the code calculates and stores the loss and accuracy for both sets. Checkpoints are saved every five epochs to preserve the model's progress. Once training is complete, the fine-tuned model is saved for use in recognizing facial expressions.

Experimental Setup

These are our experimental setup:

- Hardware: GPU T4 x2 on kaggle environment.
- Number of encoder layers: 8
- Number of heads: 4
- Hiddens dimensions: 768
- Batch size: 32
- Learning rate: 1e-4

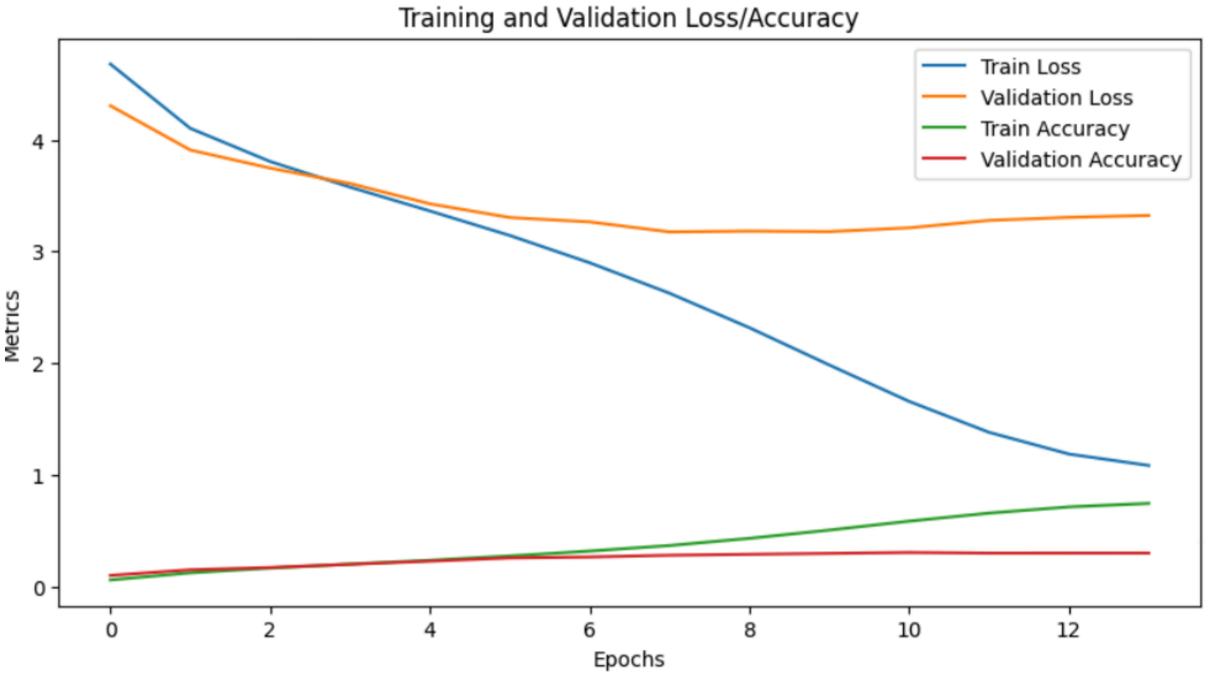
- adam weight decay = 0.1
- betas = (0.9,0.99)
- patch size = 8
- Number of epochs for training the ViT = 14
- Number of epochs for fine tuning = 8
- Size of the model:

The dataset used on this project differs from the original paper datasets, instead of using the Image21k and for fine tuning the AffectNet, the used ones was the Tiny ImageNet with 200 classes and 100000 64x64 RGB images, and for fine tuning the FER2013 48x48 RGB facial expressions pictures.

Our experimental setup also differs from the original author, the on the base ViT model uses 12 encoder layers, 8 head, and a bunch of other differences that we had to change so it could be possible to train our model in a reasonable time, considering that we had limited GPU hours in kaggle.

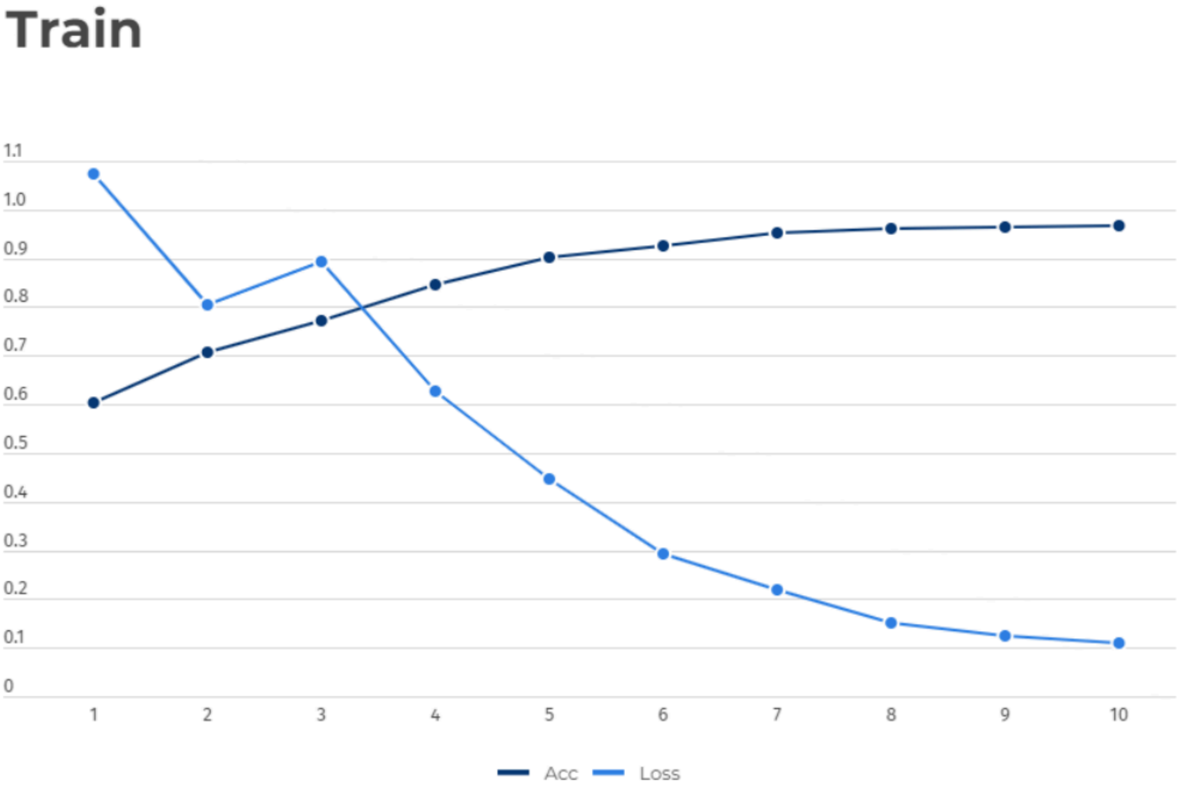
Results

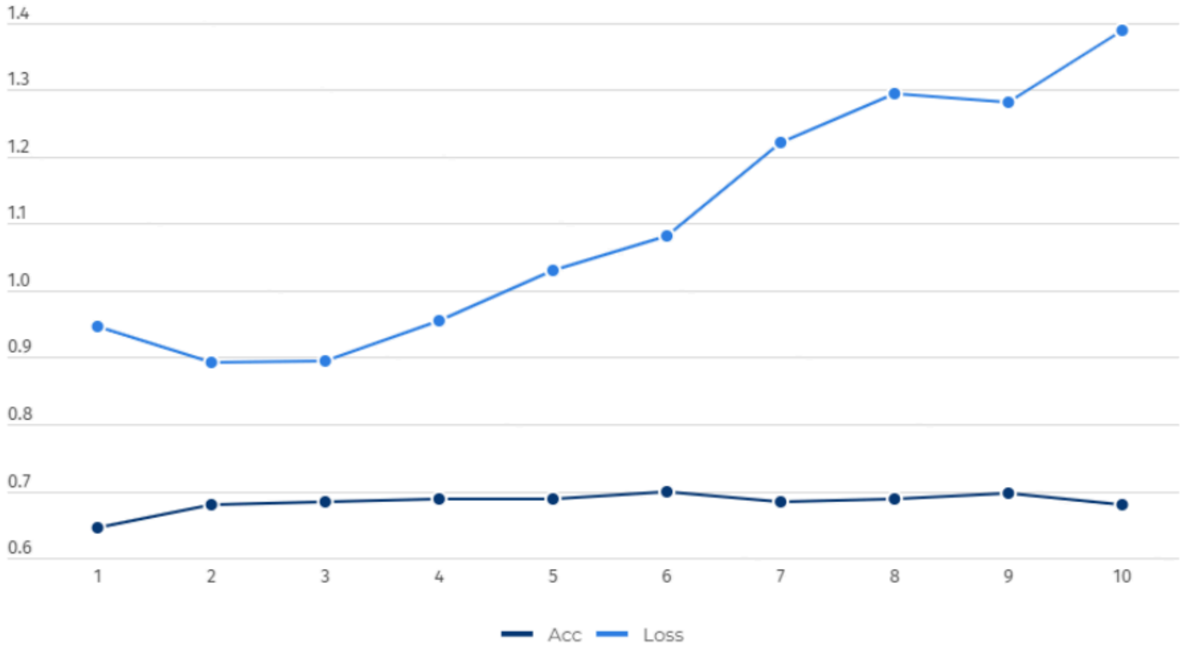
ViT Training and Validation:



With the graphs we can see that even though the training accuracy got better, and the loss got lower till the end, the validation accuracy around the 6th epoch reached a threshold and the same happened to the loss. With this information we can deduce that our model is overfitted, it didn’t generalize as we could believe if we only looked at the training statistics. In the paper VISION TRANSFORMERS IN 2022: AN UPDATE ON TINY IMAGENET they could even achieve 86.43% of precision!

Fine tuning Training and Validation:





As the graphics show, a similar thing as the first one happened, on the training the curves suggests that our model is learning, but looking at the validation, the model could get much better through the epochs, and the loss got higher instead of lower. The repository (https://huggingface.co/motheecreator/vit-Facial-Expression-Recognition) that we based our firsts ideas on, could reach 84.43% of accuracy.

We also used this youtube video to guide our implementation: https://www.youtube.com/watch?v=Vonyoz6Yt9c