ViT Project Questions

Sarthak Madan, Mridang Sheth, Shubham Banavalikar, Qianfei Hu

December 2022

1 Implement the Notebook

This paper intended to implement a ViT for image classification by utilizing a standard transformer with some data transforms to allow for a standard NLP transformer to work on images. They then decided to compare this network to state-of-the art Convolutional Networks which dominate image classification. Interestingly they found that for large datasets this ViT was actually able to perform better than CNN's. However, this performance was not maintained when trained on small/medium datasets.

In this homework we hope to explore how the paper implemented its model on a smaller dataset. In order to do so, first we will take an image and do a data transformation to convert it to a positional-encoded image with 16x16 patches. We will then feed that into a ViT (Vision Transformer) Model and train this model on the CIFAR-10 Dataset(a small dataset). To get an understanding of its performance we will compare it to a pre-trained CNN also trained on CIFAR-10. This should give us an understanding of what the paper actually implemented code wise on a small easy to train dataset.

We will then explore some questions about how transformers are able to achieve similar performance as CNN's without having the same inductive biases. We will then also explore some results found in the paper relating to positional encoding, computation estimations, and scaling. This will allow us to understand how the paper achieved the results it did on medium and larger datasets without needing insane compute. Reference Paper: https://arxiv.org/pdf/2010.11929.pdf

Implement the functions and answer the analytical questions in the given Google Collab notebook. Make sure you upload into the collab session storage Vit.py transformer.py, Utils.py, transformer-attention.py so you can run all the cells.

(a) First use our custom dataloader to create the train/validation split and choose the probabilities for each data augmentation option.

(b) Next follow the instruction in the notebook to implement patching in ViT.py and make sure you understand the visualizations.

(c) Now implement the Transform Encoder in transformer-student.py and complete the call method in ViT.py to feed the patches into the ViT model. Once you have done this and are passing the auto-graded tests you should be able to train your model. You should achieve an accuracy of around 50.

(d) Now train the provided ResNet as a baseline comparison in terms of accuracy. You should tune hyper parameters to achieve an accuracy of around 60.

(e) Answer the analytical questions at the end of the notebook. These questions will mostly stress concepts explained in the paper.

Solution: Please check the attached Python Files.

2 ViT-Related Questions

Please answer following questions related to the ViT and CNN models.

(a) We saw that the RESNET Model performed very well on the CIFAR10 DataSet. What are some of the inductive biases that allow CNN's in general to perform well for image classification?

Solution: Any answer discussing some combination of Locality, Weight Sharing, Hierarchical Structures.

(b) Do Transformers have any inherent inductive bias? In the paper we see transformers hold up and surpass accuracy bench mark tests against the CNN's for image classification? How is this possible?

Solution: Transformers don't have any inherent inductive bias making them very flexible models. However, transformers are data hungry models that require large datasets to get decent results. In the paper, the use of a pre-trained transformer model combined with a massive dataset for training allowed good accuracy results.

(c) Does transforming the images into 16x16 patches and feeding the positional locations of such patches give the transformer an inductive bias towards locality or spatial relation? Would embedding the 2D location of each patch yield significant improvements?

Solution: No. As Quoted in the Paper "the position embeddings at initialization time carry no information about the 2D positions of the patches and all spatial relations between the patches have to be learned from scratch". The paper found that the model is very clearly able to learn and attribute spatial location from the current embedding. While adding 2D location embedding wouldn't hurt the model, the model is expressive enough and shows that it already learns this, thus it wouldn't yield significant performance results.

(d) Why is scaling necessary in prepossessing the images in CIFAR-10?

Solution: In our raw dataset, the color of each pixel is in Int in the range of 0 to 255, but the models are expected to get Float between 0 and 1. Datasets downloaded from PyTorch or TensorFlow directly should have implemented this change without any further action, but a raw dataset lacks this scaling.

(e) Can we simply add the position encoding to each patch? Please explain.

Solution: Yes, we can just add the position encoding to each patch. Consider the number of pixels in each patch, which would make each patch basically a vector in high-dimensions. Therefore, it would be really difficult for two patches to collide in high-dimensions after we add the position encoding, and this is why we can safely do it.

(f) How does the accuracy of the transformer model change when trained on different size datasets? With this information when is it appropriate to use a transformer ViT over a ResNet?

Solution: ViT models perform better as the dataset increases. For massive datasets bigger than ImageNet using a ViT may yield better accuracy results than a ResNet while being easier to train.