Final Project Report for DS4440: Practical Neural Networks

Michael Wheeler*

December 11, 2020

1 Abstract

For my final project I replicated the model architecture presented in *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale* by Dosovitskiy et al. [1] in order to explore emerging research surrounding image classification with the Transformer architecture. I used the auto-differentiation library PyTorch to implement a scaled-down version of the Vision Transformer model in a self-contained Jupyter notebook, and ran the notebook on Google Colab to execute a small training experiment against a baseline model.

2 Background

For the past ten years or more convolutional neural networks (abbr. CNNs) have been considered the state-of-the-art approach for the task of supervised image classification. [2] Within the past year however, a handful of prominent researchers have begun to explore the application of the Transformer — a sequence-to-sequence architecture originally popularized for machine translation [3] — to image classification and other computer vision tasks. [1, 4, 5] As a continuation of the work I did exploring these cutting-edge approaches for my midterm topic survey, I attempted to build my own Transformer model to classify images.

In particular, I chose to focus on the Vision Transformer by Dosovitskiy et al. due to both the simplicity of the approach and the availability of reference code: the researchers tout the ability to use an out-of-the-box Transformer implementation as an advantage of their methodology, and made their implementation open-source on GitHub. [6] The authors of [1] performed experiments on models as large as 632 million parameters, trained on datasets with as many as 300 million images, making their original methodology out of reach for a half-semester's worth of work. Thus, the main contribution of my project is a scaled-down approach: implementing a model identical in structure with fewer parameters and evaluating on a smaller dataset. Specifically I chose the CIFAR-10 benchmark dataset for the project, which contains 50,000 training examples and 10,000 test examples of 32-pixel square images each belonging to exactly one of ten classes. [7, 8]

3 Implementation

The model implementation I produced for this project generally follows the structure of the reference implementation [6] with a few modifications. Image patching — the novel means by which [1] converts an image into a sequence — happens on a smaller scale (4px in length vs 14px or greater) due to the difference in size between my training images and theirs. I used PyTorch's built-in Transformer encoder components rather than rewriting them from scratch, so the structure of the encoder layers differs slightly from that of the original. Specifically the application of LayerNorm happens after the multi-headed self-attention and feed-forward components instead of before, in line with [3]. Finally I made use of smaller embedding sizes at several points throughout the model: a hidden size of 288 versus 768 or greater in the original, and a feed-forward layer size of 512 in the encoder layers versus 3072 or greater in the original. All told, my Vision Transformer consisted of approximately 8 million total parameters, versus 86 million or greater in the models described in [1].

* wheeler.m@northeastern.edu

I evaluated my Transformer implementation by also training a simple CNN as a point of comparison. The baseline model consisted of approximately 44,000 total parameters, making use of two alternating convolutional and pooling layers followed by three linear layers. In this case I use ReLU activation for all hidden states except those produced by pooling, and apply the softmax function at the end to produce the classification probabilities. Both the Vision Transformer and the baseline model make use of the same optimizer: Adam, with a learning rate of 0.00015 and betas equal to 0.9 and 0.999 respectively. I measured the performance of both models by calculating top-1 accuracy on the CIFAR-10 test set, taken after every five epochs over a training lifetime of 300 or more epochs. (see Figure 1 for details)

4 Results

Figure 1 shows the test set performance of the Vision Transformer versus the baseline model as a function of training time. The progression over epochs is clear for both models: due to the low learning rate used for each model's optimizer, the curves show a slow and steady progression towards each model's top performance on the classification task. Despite having relatively fewer data points from the Transformer's training process, it is clear to see that the Transformer model plateaus earlier and performs better by the end of training. Still, the Transformer's performance is nowhere near the >0.95 accuracies reported in [1], and there remains much room for improvement.

5 Future Work

There are a number of things I would try to further improve this model given the time. Perhaps the most important one is acquiring more training data: to come closer to the scale at which the original implementation achieved such great results I would be interested in using a much larger dataset and training on larger batches. OpenImages V6, for instance, consists of about 9 million total images with over one thousand classes, far surpassing CIFAR-10 in richness and diversity. I also believe that implementing a learning rate schedule would improve the training performance during the later epochs: the original implementation emphasized the importance of learning rate decay in their pre-training experiments. Finally, it is likely that further experimentation with hyperparameters such as the embedding dimension, number of encoder layers, and patch size would yield higher accuracy without a substantial increase in development or training time.

References

[1] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *arXiv:2010.11929 [cs]*, October 2020. arXiv: 2010.11929.

[2] Waseem Rawat and Zenghui Wang. Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review. *Neural computation*, 29(9):2352–2449, September 2017. Place: United States.

[3] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pages 6000–6010, Red Hook, NY, USA, December 2017. Curran Associates Inc.

[4] Bichen Wu, Chenfeng Xu, Xiaoliang Dai, Alvin Wan, Peizhao Zhang, Masayoshi Tomizuka, Kurt Keutzer, and Peter Vajda. Visual Transformers: Token-based Image Representation and Processing for Computer Vision. *arXiv:2006.03677 [cs, eess]*, July 2020. arXiv: 2006.03677.

[5] Mark Chen, Alec Radford, Rewon Child, Jeff Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative Pretraining from Pixels. In *Proceedings of the 37th International Conference on Machine Learning*, Vienna, Austria, July 2020. Proceedings of Machine Learning Research.

[6] google-research/vision_transformer, November 2020. original-date: 2020-10-21T12:35:02Z.

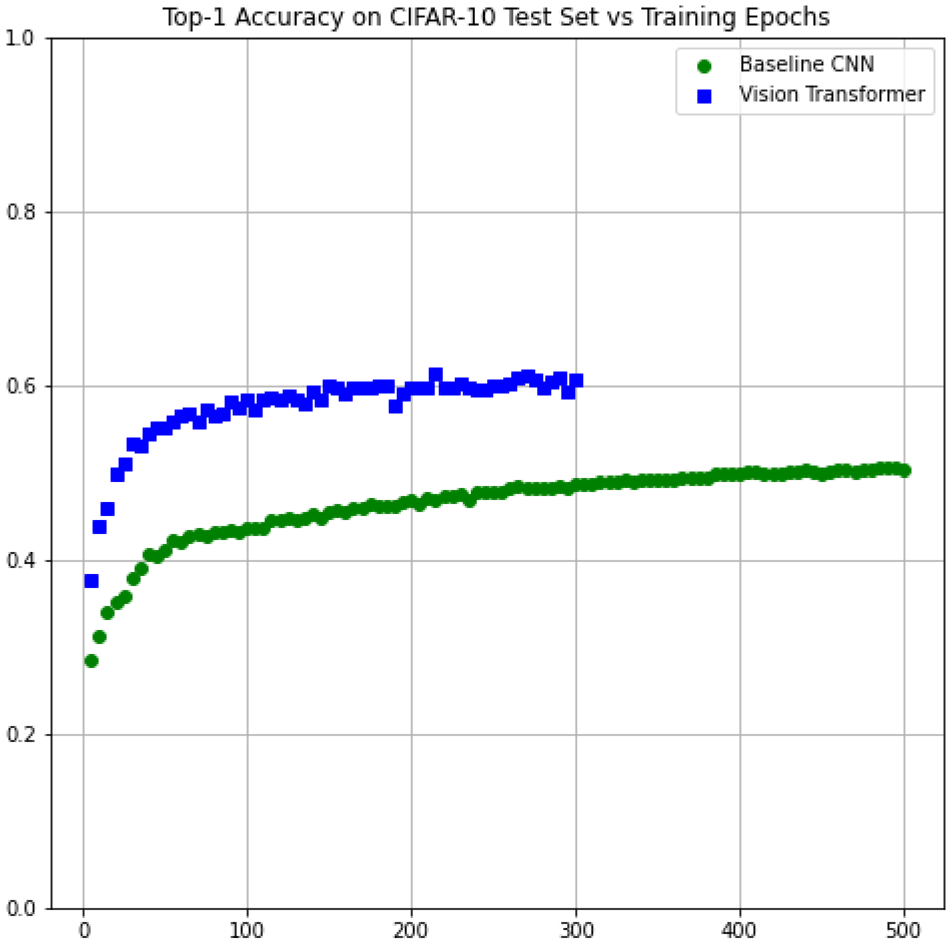

Figure 1: A plot of the Vision Transformer's performance on the CIFAR-10 test set across training time compared to a baseline CNN. Top-1 accuracy was evaluated every five epochs for both models; while the baseline model ran for 500 epochs, the Vision Transformer could only run for 300 epochs due to constraints from Google Colab.

[7] CIFAR-10 and CIFAR-100 datasets.

[8] A. Krizhevsky. Learning Multiple Layers of Features from Tiny Images, 2009.