Mini Project 4: Reproducibility in ML

Maxime Buteau (260868661)

Charles Couture (260923463)

Eric Pelletier (260895863)

Abstract

In this project, we worked on reproducibility in machine learning. We used the paper *Visual Transformers: Token-based Image Representation and Processing for Computer Vision* [Wu et al., 2020], which discusses the use of visual tokens for image classification problems. We tried to reproduce some of the experiments performed in the paper to see if we can achieve similar results. We were able to test the effect of various numbers of tokens. We did not quite achieve the same results as the paper, most likely due to the lack of computing power that the *Google Colab* platform has. Finally, we performed some more tests to analyze the robustness of the ViTResNet model presented in the paper.

Reproducibility Summary

1 Introduction

During this project, we reproduced methods and experiments illustrated in the research paper *Visual Transformers: Token-based Image Representation and Processing for Computer Vision* [Wu et al., 2020] to compare our results with the ones illustrated in the paper. Computer vision aims at understanding features inside of an image by representing images as uniformly-arranged pixel arrays and convoluting high-level features. However, this process does not take into account the importance of pixels relative to one another. For example, background pixels should not impact the convolution of features as much as the foreground pixels. In addition, all concepts do not appear in all images. More specifically, cat features do not appear in flowers images. Therefore, applying high-level filters is computationally inefficient. Also, since each convolutional filter is applied to a small region of the image, most computer vision models fail to relate spatially-distant concepts which is vital to the accuracy of the model [Wu et al., 2020].

More specifically, in this project we focus on the effect that varying number of tokens have on our model. Tokenizers are based on the idea that "an image can be summarized by a few handfuls of words, or visual tokens" [Wu et al., 2020]. To implement this intuition into the model, the paper states that they "introduce a tokenizer model to convert feature maps into a compact set of visual tokens" [Wu et al., 2020]. Therefore, based on the number of tokens used, we can get varying accuracies in our model.

2 Scope of reproducibility

Since convolutions tend to treat image pixels equally without considering their importance, the experiments conducted in the research paper try to represent images as semantic visual tokens and densely model token relationships using transformers [Wu et al., 2020]. In our experiments, we try to recreate a subset of the research paper's experiments while following their methods to the best of our capabilities. More specifically, due to the computational resources available during our experiments, we focus on the effects of varying numbers of visual tokens has on the model's accuracy. As stated in the original paper, Visual Transformers already capture a wide variety of concepts with just a few tokens.

Since the authors of the paper had access to much more performant computers than we did, we did not try to get similar accuracy results. However, we did try to show that increasing the number of visual tokens past 16 will have little to no effect on the model's accuracy.

33rd Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

3 Methodology

For the experiments, we used the code provided by on this Github repository which is a PyTorch implementation of Visual Transformer model described in the research paper. In addition, we used the resources provided by google colab which is the Python 3 Google Compute Engine backend's resources.

3.1 Model descriptions

The model used in the paper is a Visual Transformer residual neural network, or ViTResNet. It uses pytorch to help with its implementation and is not pretrained. We had to train it from scratch for all of our tests, which explains why we could not very closely reproduce the test results. Some of its parameters include the type of building block that it uses and the block configurations, the number of visual tokens, the batch size, the number of epochs, the momentum, and the weight decay.

3.2 Datasets

The experiments in the research paper were conducted on the ImageNet dataset with around 1.3 million images in the training set and 50 thousand images in the validation set. However, due to computational resources, we conducted our experiments on the CIFAR-10 dataset which consist of 60000 32x32 colour images in 10 classes, with 6000 images per class. The CIFAR-10 dataset is separated into 50000 images for the training set and 10000 images for the test set. The CIFAR-10 dataset can be downloaded directly from here.

3.3 Hyperparameters

We have performed some hyperparameter tuning, but we were somewhat limited with our computing resources. First, we tested the number of epochs. The paper uses 90 epochs, which is far too much for our Google colab resources. Running a single epoch took us around 9 minutes. Table 1 shows the results of the two epoch related experiments we ran.

# of epochs	Accuracy	Average Test Loss	Execution time (seconds)
3	34.83%	1.7887	1756.59
5	54.17%	1.6165	2700.55

Table 1: ViTResNet accuracy, average test loss and execution time on varying numbers of epochs.

We determined that this 54.17% accuracy was good enough since this training already takes 45 minutes.

Regarding batch sizes, we kept them at 100 even if the paper sometimes uses batch sizes of 256. Once again, this is for the sake of reducing training time. However, we also noticed that only a subset of the training data would be used with batches larger than 200, so this was another reason for keeping the batch sizes at 100.

For momentum and weight decay, we simply used the values provided in the paper. They are as follows:

- **momentum**: 0.9
- **weight decay**: 1×10^{-4}

3.4 Experimental setup and code

The experiments were set up by first setting the number of epochs to 5. Then, our ViTResNet model was initialized with a batch size of 100 and basic blocks of 3 layers with 3 blocks each and the optimizer was initialized with torch optim's Adam algorithm and a learning rate of 0.003. For each experiment, the hyperparameter being tested was changed accordingly. Following this setup, the model was trained using the model's train function and the training dataset for each epoch. After every iteration of training, the model's accuracy, average loss and execution time were evaluated on the test dataset using the model's evaluate function. The link to the Google colab project with our code can be found here.

3.5 Computational requirements

The experiments were conducted with the resources provided by Google colab. The CPU utilized was a single core hyper threaded Intel Xeon Processor operating at 2.20GHz with a cache size of 56 thousand kB. As well, the total

memory available during the experiments was 12 million kB. The runtime of our ViTRestNet model for each experiment will be displayed in the results section.

4 Results

The results measured during our experiments differ slightly from the main claim of the original paper.

4.1 Results reproducing original paper

In this experiment, we tested out the claim from the original paper regarding how varying number of visual tokens will have negligible to no increase in the model's accuracy by evaluating our ViTResNet model with 64, 32 and 16 tokens. Our results observed during this experiment differ slightly from the results stated in the original paper.

4.1.1 Experiment Result

The results obtained during this experiment are illustrated in table 2.

# of Tokens	Accuracy	Average Test Loss	Execution time (seconds)
16	51.45%	1.4095	3165.83
32	51.61%	1.3826	4248.07
64	54.49%	1.3099	5832.39

Table 2: Our results for ViTResNet accuracy, average test loss and execution time on varying numbers of tokens.

Although all accuracies remain relatively close, there is a jump of about 3% when going from 32 to 64 tokens.

4.1.2 Original Paper Result

The results illustrated in the original paper corresponding to this experiment can be found in the table 3.

# of Tokens	Accuracy	FLOPs	Params
16	71.8%	1579	11.6
32	71.9%	1711	11.6
64	72.1%	1979	11.6

Table 3: Paper's results for ViTResNet accuracy, average test loss and execution time on varying numbers of tokens.

As we can see, the accuracies are much closer together, with a maximum difference of 0.3%.

4.2 Additional Experiments

4.2.1 Effect of the number of classes

The ViTResNet seems to still perform relatively well with larger numbers of classes. Table 4 shows the results we obtained for 10, 20, and 50 classes. All these tests were performed with the same hyperparameters and 5 epochs.

# of classes	Accuracy	Average Test Loss	Execution time (seconds)
10	54.17%	1.6165	2700.55
20	56.09%	1.6242	2748.20
50	49.55%	1.6707	2722.07

Table 4: Effect of different number of classes

There does not really seem to be a trend, our model seems to perform fairly evenly for different numbers of classes.

Weight decay	Momentum	Average Test Loss	Execution time (seconds)
1×10^{-4}	22.71%	1.9603	2911.37

Table 5: Effect of adding weight and momentum

4.2.2 Adding momentum and weight decay

We can see that the accuracy is much lower than without using weight decay and momentum. However, this is likely because we don't use enough epochs and too small batch sizes. In the paper, an accuracy of 72.1% was achieved by adding weight decay and momentum.

5 Discussion

In our experiment, we noticed that there was very little difference in accuracy between our model with 16 tokens and our model with 32 tokens similar to the results obtained in the original paper. However, our model's accuracy improved significantly when implemented with 64 tokens which differs from the original claim of the paper and the results illustrated in the original paper.

Our results may differ form the original paper's results due to the fact that we evaluated our model's accuracy on a different dataset which is a lot smaller than the dataset stated in the original paper. The CIFAR-10 dataset that we utilized during this experiment has a 1:5 ratio of validation set to training set size, while the ImageNet dataset that is stated in the original paper has a 1:26 ratio of validation set to training set size. Therefore, our model's accuracy may differ from the original paper's model due to this reason.

In addition, our model was implemented with a batch size of 100 and was trained for only 5 epochs due to the limitation of computational resources we had available, while the model described in the paper was implemented with a batch size of 265 and trained for 90 epochs. Therefore, these differences in the model's implementation may affect the results that we obtained compared the results illustrated in the original paper.

5.1 Additional tests and ablation studies

With our additional experiments, we were able to see that the ViTResNet is rather robust regardless of the number of classes for the images. This shows that this tokenizing approach indeed seems to be a good approach for image classification.

In terms of ablation studies, we tested the ViTResNet both with and without weight decay and momentum. Our tests showed that the model performed much better without these extra parameters (54.49% without vs. 22.71% with). However, this is very likely due to the fact we could only run 5 epochs and batches of 100 in reasonable time. In the paper, 90 epochs were run and the batches had a size of 256.

5.2 What was easy

The code used in our experiment was easy to implement since a GitHub repository created a PyTorch implementation of the model used in the original paper.

5.3 What was difficult

The model's parameters were hard to adjust since the provided code did not have all the different functions used to evaluate various parameters and the math used to describe the algorithms in the original paper requires advanced knowledge of calculus and linear algebra to follow. In addition, the experiments described in the original paper took very long to run, therefore, we needed to perform parameter optimisation to find model parameters that at the same time provide good accuracy and decreased the run time by a large amount.

5.4 Communication with original authors

We did not have any communication with the original authors.

References

Bichen Wu, Chenfeng Xu, Xiaoliang Dai, Alvin Wan, Peizhao Zhang, Zhicheng Yan, Masayoshi Tomizuka, Joseph Gonzalez, Kurt Keutzer, and Peter Vajda. Visual transformers: Token-based image representation and processing for computer vision. *arXiv preprint arXiv:2006.03677*, 2020.