An Image is Worth 16 by 16 Words: Transformers for Image Recognition at Scale (This presentation is all you need)

By Curtis Fox
Outline

- Introduction
- Related Work and Motivation
- Architecture
- Results
- Conclusion
Introduction
Introduction

- As we’ve discussed in previous MLRG’s, the main use of transformers has been for NLP tasks
  - This involves training on some large corpus, then doing fine-tuning on some smaller dataset
- Do not need CNN’s to effectively perform image tasks, and their dominance in vision tasks is being challenged by the Vision Transformer (ViT)
- This talk will instead focus on the discussion of transformers for computer vision tasks
  - Will discuss more details about this in a bit
Related Work and Motivation
Related Work and Motivation

- The simplest use of self-attention for images would be to have every pixel in the image attend to every other pixel.
  - This has quadratic cost in the number of pixels, will not scale for larger inputs
- Another approach is to apply self-attention only in local neighbourhood for each query pixel
- Finally another approach is to do a type of approximation to global self-attention, referred to as Sparse Transformers
- The paper I will discuss takes a simple approach to scale transformers to images, which allows an almost direct application of transformers
Related Work and Motivation

- When pre-training on smaller datasets, transformers do not outperform CNN’s for image tasks
  - This is because transformers lack many of the inductive biases that CNN’s have for images. As in, CNN’s are designed in such a way to be used for image data, which is not the case for transformers.
- Require large pre-training datasets to actually see the usefulness of the ViT models
  - This motivates the work discussed in this paper
Architecture
Architecture

- Break image into patches of a chosen size (say 16 by 16 pixels)
  - Can think of these patches as corresponding to words in the NLP setting
- Flatten these patches into vectors
- Multiply these flattened patches by a matrix $E$
  - This is the linear projection step that transforms the input into a smaller $D$-dimensional space
Architecture

- Add learnable class embeddings
  - Can think of this as being used to learn labels for image
- Add learnable positional embeddings to each of the patch embeddings
  - These embeddings only hold 1D information, as the authors tried using 2D information and found this did not help much
- This is then passed to the transformer encoder that we’ve previously seen (same as in BERT paper)
Architecture

- The paper also discusses a hybrid model which combines transformers and CNN's.
- Instead of directly multiplying the image patches by the matrix E (which performs the projection), we replace the image patches with patches extracted from a CNN feature map.
  - The feature map is the result of applying a filter to the image of interest.
Results
## Results

<table>
<thead>
<tr>
<th>Model</th>
<th>Layers</th>
<th>Hidden size $D$</th>
<th>MLP size</th>
<th>Heads</th>
<th>Params</th>
</tr>
</thead>
<tbody>
<tr>
<td>ViT-Base</td>
<td>12</td>
<td>768</td>
<td>3072</td>
<td>12</td>
<td>86M</td>
</tr>
<tr>
<td>ViT-Large</td>
<td>24</td>
<td>1024</td>
<td>4096</td>
<td>16</td>
<td>307M</td>
</tr>
<tr>
<td>ViT-Huge</td>
<td>32</td>
<td>1280</td>
<td>5120</td>
<td>16</td>
<td>632M</td>
</tr>
</tbody>
</table>

- Train 3 versions of their model, of varying sizes
- For context, the Base and Large models are taken from BERT
This table summarizes the number of CPU days taken to pre-train each of the various models, as well as the achieved test accuracies on various datasets.

Overall the huge VIT model seems to perform the best.

The ViT models also took a lot less time to pre-train.
This table analyzes the dataset VTAB by task group. In particular:
- Natural: Pets, CIFAR, etc,
- Specialized: Medical and Satellite Imagery
- Structured: Tasks that require geometric understanding like localization
ViT performs well across different types of task groups
Results

Table 9: Breakdown of VTAB-1k performance across tasks.

<table>
<thead>
<tr>
<th></th>
<th>Caltech101</th>
<th>CIFAR-100</th>
<th>DTD</th>
<th>Flowers102</th>
<th>Pets</th>
<th>Sun397</th>
<th>SYNH</th>
<th>Camelyon</th>
<th>EuroSAT</th>
<th>Reise45</th>
<th>Retinopathy</th>
<th>Cleve-Count</th>
<th>Cleve-Dist</th>
<th>DMLab</th>
<th>dSpr-Loc</th>
<th>dSpr-Ori</th>
<th>KITI-Dist</th>
<th>sNORB-Azim</th>
<th>sNORB-Elev</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>ViT-H/14 (JFT)</td>
<td>95.3</td>
<td>85.5</td>
<td>75.2</td>
<td>99.7</td>
<td>97.2</td>
<td>65.0</td>
<td>88.9</td>
<td>83.3</td>
<td>96.7</td>
<td>91.4</td>
<td>76.6</td>
<td>91.7</td>
<td>63.8</td>
<td>53.1</td>
<td>79.4</td>
<td>63.3</td>
<td>84.5</td>
<td>33.2</td>
<td>51.2</td>
<td>77.6</td>
</tr>
<tr>
<td>ViT-L/16 (JFT)</td>
<td>95.4</td>
<td>81.9</td>
<td>74.3</td>
<td>99.7</td>
<td>96.7</td>
<td>63.5</td>
<td>87.4</td>
<td>83.6</td>
<td>96.5</td>
<td>89.7</td>
<td>77.1</td>
<td>86.4</td>
<td>63.1</td>
<td>49.7</td>
<td>74.5</td>
<td>60.5</td>
<td>82.2</td>
<td>36.2</td>
<td>51.1</td>
<td>76.3</td>
</tr>
<tr>
<td>ViT-L/16 (I21k)</td>
<td>90.8</td>
<td>84.1</td>
<td>74.1</td>
<td>99.3</td>
<td>92.7</td>
<td>61.0</td>
<td>80.9</td>
<td>82.5</td>
<td>95.6</td>
<td>85.2</td>
<td>75.3</td>
<td>70.3</td>
<td>56.1</td>
<td>41.9</td>
<td>74.7</td>
<td>64.9</td>
<td>79.9</td>
<td>30.5</td>
<td>41.7</td>
<td>72.7</td>
</tr>
</tbody>
</table>

- Breaks down the task groups into their separate tasks
- Further experiments to support/justify the use of their larger model
- In some places the dataset pre-trained on makes a big difference in task accuracies
When pre-training on smaller datasets, the ResNets outperform the ViT’s.

As you increase the pre-training dataset size, the ViT begins to perform better.
- It seems like more data really helps with ViT’s.
Here a subset of the JFT is used for pre-training
   - As the subset used is increased again we eventually see the ViT models overtake ResNets
For the same amount of computing power, the ViT models outperform ResNet.

The Hybrid models outperform the ViT models for smaller amounts of compute, but as we use more computational resources, the ViT model also outperforms the hybrid methods.
Results

- This data shows how the error changes as we vary different parameters.
- Varying all parameters proportionally seems to work well.
- Varying depth seems better than width, which seems to level off.
- Scaling the patch size seems to also help as well.
Results

- This image shows the learned position embeddings of the model.
- The ViT model actually learns to encode distance within the images provided based on the similarity of position embeddings.
- Patches closer together have similar position embeddings.
Results

- Shows that the transformer is attending to the important regions of the provided images
Additional Results (beyond the original paper)
Additional Results

- An even larger ViT model was trained, and beat the previous “huge” ViT model
- Also beats the ResNet results (grey dots)
Additional Results

- Some more results with the larger model, showing that scaling up even further can make a difference
Additional Results

- Scaling down the ViT seems to really hurt performance
  - For the same sampling rate, the larger models perform significantly better
- As well, keeping the amount of data used for training constant and scaling down the model hurts performance significantly
We see here that as we continually scale up the models and provide larger pre-training datasets we can get better results.

Also see a leveling off for each of the models, and providing larger datasets provides marginal gains in error rate improvements.

Can read more about this in the “Scaling Laws for Neural Language Models” paper.
Conclusion
Conclusion

- Require significantly fewer resources to pre-train than previous methods, but perform even better
  - Increasing pre-training dataset sizes and the transformer model sizes help with performance
- Even though transformers were originally designed for NLP tasks, they have extended well to image classification tasks
  - What other tasks could transformers be used for?
- Some challenges that still remain are how transformers can be extended to other image tasks such as image segmentation
Conclusion

- Let us conclude with some issues with this work (at least in my opinion)
- Those of us with fewer computational resources cannot replicate these results, limiting this line of work to very large companies
- All of the usual bias and prejudice issues that come from labeling image data
- Main takeaway from the paper as far as I can tell is that bigger is better, which I don’t consider overly insightful
Thank you for listening!

Questions?
References

- https://www.youtube.com/watch?v=BP5CM0YxbP8