Deep Learning Project Report

Investigating Possible Inductive Biases in Local Sparse Attention ViT Architectures Against Traditional CNNs

A. Feer, B. L. Hilmarsson, M. Noruišis, S. Ray

ABSTRACT

Deep learning models for vision tasks have been shown to work well due to factors like inductive biases, generalization of models and so on. In this project, we explore the possible inductive biases or lack of them across different architectures used in vision tasks such as Convolutional Neural Networks and Transformer architectures. We present a series of experiments which help understand these inductive bias, and their role in helping these models generalize.

1. INTRODUCTION

The inductive bias of machine learning algorithms refers to the set of underlying assumptions made by the model in order to generalize better on inputs which have not been seen before. Typically, in learning problems, the algorithm is presented with training examples, from which it tries to derive a relationship between the input and output data points. However, the model should also be able to extend its prediction capabilities to input values never seen during the training phase and predict a reasonably correct output for these. As outputs for these samples may be arbitrary, it is often needed to have a set of assumptions based on which the relationship between inputs and outputs may be generalized better, which is known as the inductive bias.

In deep learning-based computer vision tasks, Convolutional Neural Networks (CNNs) have long been used for a variety of tasks, including classification and generation. Pre-trained CNNs, such as VGG19 [1] and ResNet50 [2], have been especially powerful. CNNs usually make use of a number of inductive biases, one being the *spatial bias* [3], which assumes a certain type of spatial structure is present in the data. CNNs also use the principle of *local structure* as a bias, which assumes that pixels close to each other in the image have a higher likelihood of being more important in relation to each other, compared to pixels which are far away from each other [4]. Thus, local structure in images tends to be more prominent than global structure for the algorithm [5]. Inductive biases help CNNs to perform well for general image and vision tasks.

With transformer-based models such as the Vision Transformer things work a little differently. Transformers use attention matrices in their architecture in order to focus on different areas of the input vectors and to not treat them uniformly. However, computing the full attention matrix can be computationally infeasible at times and thus we have sparse attention transformers [6]. In these models, we make use of the observation that attention matrices tend to be sparse and low rank, and, using this fact, computations using the attention matrix can be made much faster, with negligible performance loss (sparse matrices approximate well). Also, local attention models have been proposed where attention is applied over smaller sliding windows, instead of the entire feature vector [7]. We want to explore the principles, causes and effects of local sparse attention and how they compare to the inductive biases of CNNs, especially for the locality principle [8, 9].[1]

2. BACKGROUND

2.1. ResNet50 Architecture

A residual neural network (ResNet) [2] is an artificial neural network (ANN) that stacks residual blocks on top of each other. ResNet50 is used to denote the variant of residual CNN with 50 neural network layers. The first ResNet architectures involved the insertion of shortcut connections in turning a plain network into its residual network counterpart. The plain network was inspired by VGG neural networks (VGG16), with the convolutional networks having 3×3 filters. The layers had the same number of filters for the same output feature map size, and the number of filters doubled with halving feature map size to preserve the time complexity per layer. The building block for ResNet50 is a bottleneck design due to concerns over the time taken to train the layers. This used a stack of 3 layers, forming the ResNet50 architecture.

[1] Our implementation can be found at https://github.com/Siddhant-Ray/Deep-Learning-Project-2021

2.2. ViT Architecture

The Vision Transformer (ViT) architecture was introduced by Dosovitskiy et al. [10] in 2021. The architecture successfully employs attention based techniques, which previously were primarily used in language processing, to image recognition. Contrary to previous attempts to apply transformers to vision, this architecture makes no use of convolutional layers. Therefore, it lacks inductive biases such as translation invariance or locality. The authors of ViT show that it can compete with or outperform convolutional architectures such as ResNet in various benchmarks.

2.3. Local ViT Architecture

The Local Vision Transformer (Local ViT) was introduced by Han et al. [7] in 2021. This architecture builds – among others – on the ViT architecture described in Section 2.2. The authors of Local ViT reintroduce some mechanisms similar to those employed in traditional convolutional architectures. These mechanisms include the following: i) *sparse connectivity* – in a layer, some input neurons and output neurons are not interconnected; ii) *weight sharing* – some connection weights are shared by multiple connections. These changes possibly reintroduce some biases common in CNNs.

3. OUR PROPOSED METHOD

We devise several experiments to investigate possible inductive bias. In particular, we set the following goals:

- To benchmark the performance of a CNN-based architecture and local sparse attention transformers on a standard image dataset.
- To study the heuristics of the performances and attempt to correlate them with the inductive biases which we started upon. Particularly, to try to determine if the inductive biases work in local attention transformers as well as in CNNs, the extent to which the performances match, where they differ, etc.

3.1. Training from scratch on CIFAR-10

We have already talked about the principle of local structure which is an inherent inductive bias present in CNN architectures. This helps CNNs learn texture and shape for images, based on the condition that pixels which are close to each other are more important with respect to each other. We also know that the architecture of ViT overcomes this inductive bias, as the image is processed by the model as a subsequence of smaller areas of the image. By doing so, the notion of global structure and local structure is significantly reduced and the learning problem is treated as a sequential data identification problem. Finally, we know that for Local ViT, the attention on the sequence is applied independently over small local windows, over the entire sequence. For our task, we train the models from scratch on an image dataset with relatively small images and we choose the CIFAR-10 image dataset for this.

The CIFAR-10 dataset [11] consists of 60000 32x32 color images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images. The 10 classes for the images are: airplane, automobile, bird, cat, deer, dog, frog, horse, ship and truck. The images are mutually exclusive without any overlap. Figure 1 contains a random sample from the dataset, with an instance of every image class. We choose the CIFAR-10 dataset due to its size as it does not have enough images for the transformer to generalize. As it is known that transformers generalize very well on large datasets, training on huge datasets here would not allow us to capture the behavior of the inductive biases as the full ViT would outperform every other model.

We believe that with our choice of dataset, a local vision transformer should also inherently capture the same inductive bias of local structure due to its local attention principle. To evaluate this, we run and expect the following hypotheses to hold:

- We use ResNet50 as our CNN model to train on the CIFAR-10 dataset from scratch. We already know that it uses the inductive bias of local structure and we expect the model to learn well from the dataset, despite the small size of the dataset.
- We use the ViT as our full attention transformer model to train on the dataset from scratch. We expect the model to perform significantly worse on the small dataset than the ResNet50 model, due to transformers being extremely data hungry to train, and the lack of the bias of learning local structures should also reflect the lack of similar performance levels.
- We use the Local ViT as our local attention transformer model on the dataset from scratch. We expect this model to perform similar to the ResNet50 on the small dataset as we expect it to incorporate the same inductive bias of learning local structures, compared to only learning global structures.

For our experiments and hypothesis to hold, we ensure that we train all our models with exactly the same hyperparameter configurations, and use the same optimizing method across all. Our hypothesis boils down to, while keeping all parameters the same, on a dataset like CIFAR-10, CNNs should perform extremely well, full attention transformers should perform worse worse and local attention transformers should be similar in performance to the CNN, when trained from scratch.

Fig. 1. Example images for all CIFAR-10 classes

3.2. Classification on combined CIFAR-10 images

To evaluate the models, we introduce two different custom datasets that are designed to test for inductive bias. The first such dataset is called *Combined CIFAR-10* and is defined as follows:

- An image is a composition of 4 different sub-images arranged in a square of size 64x64. The sub-images are all randomly chosen from CIFAR-10 and represent distinct CIFAR-10 classes. This means that there are exactly 4 classes present in the image.
- A label l is a 10-dimensional vector l. One dimension for each class in CIFAR-10. The entry l_i is set to 1 if and only if the corresponding image contains a sub-image of class i. All other entries in l are set to 0.

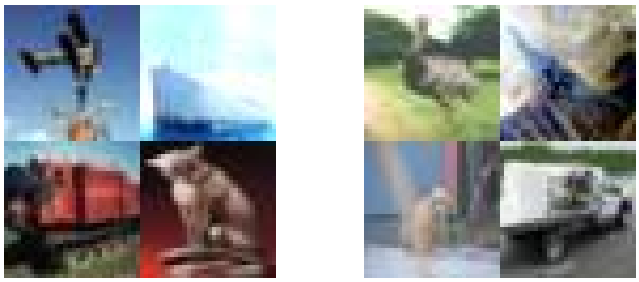

Fig. 2. Example images combined from single CIFAR-10 images.

This dataset can be used to test how well the models generalize with images that contain multiple classes. We are interested in seeing how these architectures behave, even though they were trained on a single-class dataset.

Given that each image contains 4 different classes and cannot be assigned to a single class, the Combined CIFAR-10 dataset does not pose a classification task in the traditional sense. We introduce a new metric called *Top 4 Matching* to evaluate performance of a model on the Combined Images. Top 4 Matching is defined as follows:

- Given the ground truth label l and the likelihood prediction p, we compute the 4 classes that are most likely according to the prediction p. We then count how many of the 4 predicted classes are actually present in the ground truth label. This yields an integer in $\{0, ..., 4\}$ for each image prediction in Combined CIFAR-10.
- Top 4 Matching values can be summarized by averaging or visualized in a histogram.

3.2.1. Random Classifier

To get an intuitive baseline for our new Top 4 Matching metric, we analyze how well a random classifier would perform on the Combined CIFAR-10 dataset. We are interested in the expected Top 4 Matching value of a classifier that uniformly at random chooses 4 out of 10 classes. W.l.o.g. we can evaluate a single trial, since the classes all have the same size, and images are generated uniformly at random. We define $\mathbb{E} = \sum_{i=1}^{4} E_i p_i$ as the expected number of correctly classified images. The reason we go from 1 to 4, is because there are 4 cases: either we classify one correctly, two correctly and so on. Note that zero correctly classified images adds nothing to the expectation. Clearly $E_i = i$. The number of ways we can choose some classification of 4 images is $\binom{10}{4}$. The number of i correct classifications is simply counting how many times we can select i correct images out of 4, multiplied by how many times we can select $4 - i$ wrong ones out of 6. Therefore $p_i = \frac{\binom{4}{i} \cdot \binom{6}{4-i}}{\binom{10}{4}}$ and thus conclude our equation[2].

$$\mathbb{E} = \sum_{i=1}^{4} E_i p_i = \frac{\sum_{i=1}^{4} \binom{4}{i} \cdot \binom{6}{4-i} \cdot i}{\binom{10}{4}} = 1.6 \qquad (1)$$

3.3. Classification on images with large background

Most of the objects in the CIFAR-10 dataset fully cover the image. So we designed an experiment to see how these models would react to the objects being scaled down and put on a random background. Thus, we created the second custom dataset which we call *Background* and define as follows:

- Out of 20 large background pictures one is selected. These images consist of less hectic information, such as grass, desert, water, patterns and more. None of the background images have any objects which coincide with our classes. Then a random 128x128 part is cut out.
- Next an image from CIFAR-10 is selected and pasted on the image at a random position. The label of the class is also the same as of the CIFAR-10 image.

[2]Also verified via code.

This results in a dataset of 200 images. We proceed to run the models, which were trained CIFAR-10 dataset. It is good to keep in mind that a uniformly random classifier should achieve an accuracy of 10% on expectation.

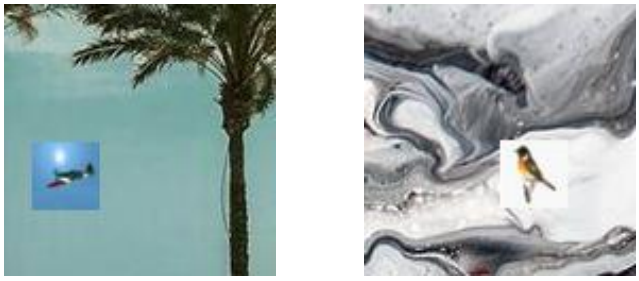

Fig. 3. Example images with a large background.

4. EXPERIMENTAL RESULTS

4.1. Training models from scratch on CIFAR-10

4.1.1. Setup

We setup three architectures for ResNet50, ViT and Local ViT models. The first two being standard implementations in the PyTorch and HuggingFace repositories respectively, and the third can be found here. All our implementations use PyTorch. For training, we use 8 GPUs in parallel using the DataParallel package and the summary of our training hyperparameters is given in Table 1, which are the same across all models.

Hyperparameter	Value
number of epochs	80
learning rate	1e-5
batch size	64
weight decay	5e-4

Table 1. Summary of our model hyperparameters

We use the Adam optimizer [12] and the OneCycle learning rate scheduler [13], across all our models. In order to speed up the training process, we also use mixed precision loss supported by PyTorch [14].

4.1.2. Evaluation and discussion

To evaluate the performance of our models, we measure the training loss until it reduces to an acceptable level and then measure the test accuracy across all our models.

Figure 4 shows the training loss and test accuracy across models. We choose the top 1 accuracy as the metric for evaluation, based on the fact that the dataset is balanced in number of items per class. According to our initial hypothesis, due to the principle of local structure being an inductive bias for CNNs, we expect it to perform better than the full transformer. We see that the test accuracy for our ResNet50 model is much better than that of the full ViT. Also, it can clearly be observed that the test accuracy for Local ViT is very similar to that for ResNet50, which gives a strong indication that the idea of attention over local windows is a similar inductive bias to the local structure principle used in CNNs.

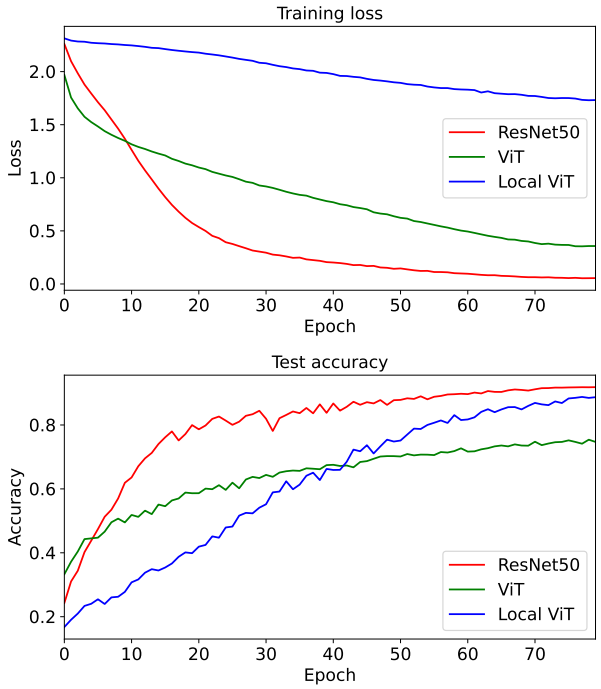

Fig. 4. Performance metrics across models

4.2. Classification on combined CIFAR-10 images

We now evaluate the performance of the individual models for our task of identifying the classes from the combined images. As mentioned in Section 3.2, we use the *Top 4 Matching* metric and measure how many of the classes have been correctly classified by each model. We plot a histogram for the same in Figure 5. Also, it is interesting that the Local ViT performs significantly well for detecting 3 classes correctly but does not ever detect 4 classes correctly.

Model	Average Top 4 Matching
Random Classifier	1.6
ResNet50	2.09
ViT	1.93
Local ViT	2.13

Table 2. Top 4 Matching for each model on the combined dataset.

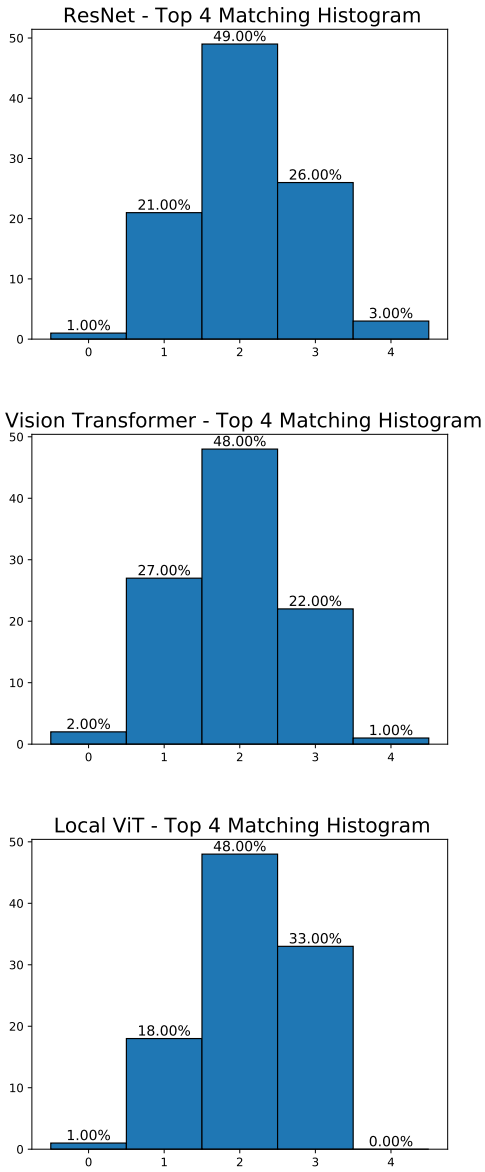

Fig. 5. Accuracy on combined CIFAR-10 images for each model.

In Table 2, we see the average Top 4 Matching for all models. From Equation 1 we know that a random classifier in this case will only have a Top 4 Matching of 1.6. For us, we can clearly see that all models perform better at identifying multiple classes in an image, even though the training data only had mutually exclusive classes. It is interesting to see that the histograms for Local ViT and ResNet50 show that they can detect a higher number of correct classes per image (more right skewed) whereas the ViT can detect a lower number of correct classes (more left skewed). We kept all parameters for training and evaluation the same for all models, so the fact that the Local ViT Top 4 Matching metrics are similar to those of ResNet50, is a strong indication that this is due to the inductive bias of local structure, with the attention applied over local windows.

4.3. Classification on images with large background

We already know that CNNs depend heavily on the local structure of images, in order to learn the feature and shape of the different localized similar pixels. By training our model on images where the object covers the entire image we expect our ResNet50 model to be heavily biased towards recognizing images of only this type and not generalizing towards smaller objects. Thus, by introducing images with a very large background, and the actual target only being present in a small part of the image, we expected the ResNet50 model to perform poorly on these images. As we have also hypothesized that the Local ViT uses a similar inductive bias, we expected similar results for it too.

Model	Accuracy
Random Classifier	10.0 %
ResNet50	15.0 %
ViT	10.5 %
Local ViT	14.5 %

Table 3. Accuracy for each model on background dataset.

However, as the ViT processes the image as batches of smaller sequences and does not use the bias of local structure, we expected the ViT to actually perform better than the other models here. Interestingly, the results are quite surprising here. Table 3 shows that the ResNet50 model and the Local ViT model, do not perform well and are similar. However, the performance of ViT here is even worse, the same as a random classifier. This does not conform with our original hypothesis, and the reason for this is probably that the transformer cannot generalize due to insufficient training data, and also does not have the bias towards local structure, which makes the overall performance much worse.

5. CONCLUSIONS

In our project, we explored the inductive bias of local structure across ResNet50, ViT and Local ViT. Based on the architectures of the models, we had an initial hypothesis that the Local ViT, due to its attention over small local windows, incorporates the same bias of local structure present in images as CNNs. To explore this, we trained the models from scratch and compared their performance, and observed the similarity in performance between ResNet50 and Local ViT, keeping all parameters the same. Further, we carried out more evaluation experiments to assert this similarity. Based on the results, we believe that there is a strong likelihood that the Local ViT does indeed incorporate the inductive bias of local structure, whereas the full ViT does not.

6. REFERENCES

[1] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[3] B. R. Mitchell, "The spatial inductive bias of deep learning," 2017. [Online]. Available: http://jhir.library.jhu.edu/handle/1774.2/40864

[4] M. Jagadeesan, I. Razenshteyn, and S. Gunasekar, "Inductive bias of multi-channel linear convolutional networks with bounded weight norm," 2021.

[5] B. Neyshabur, "Towards learning convolutions from scratch," 2020.

[6] S. d'Ascoli, H. Touvron, M. Leavitt, A. Morcos, G. Biroli, and L. Sagun, "Convit: Improving vision transformers with soft convolutional inductive biases," 2021.

[7] Q. Han, Z. Fan, Q. Dai, L. Sun, M.-M. Cheng, J. Liu, and J. Wang, "Demystifying local vision transformer: Sparse connectivity, weight sharing, and dynamic weight," 2021.

[8] A. Roy, M. Saffar, A. Vaswani, and D. Grangier, "Efficient content-based sparse attention with routing transformers," 2020.

[9] C. Meister, S. Lazov, I. Augenstein, and R. Cotterell, "Is sparse attention more interpretable?" 2021.

[10] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," 2021.

[11] "Cifar-10 dataset." [Online]. Available: https://www.cs.toronto.edu/~kriz/cifar.html

[12] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017.

[13] L. N. Smith and N. Topin, "Super-convergence: Very fast training of neural networks using large learning rates," 2018.

[14] "Mixed precision loss pytorch." [Online]. Available: https://pytorch.org/docs/stable/amp.html#torch.cuda.amp.autocast