Model Performance Comparison between CNN, MLP and Vision Transformer for Supervised Image Recognition

Savannah Wang
Columbia University, IEOR 4540 Final Project
April 23rd, 2021

Abstract

Supervised image recognition has been a wide area of research in machine learning, and convolutional architectures play a dominant role in this area, especially CNN. On the other hand, transformers have been used widely in the area of natural language processing. Inspired by the Transformer scaling successes in NLP, in this project, I am going to explore the performance of transformers on image recognition. I only trained the model locally, so I elimited the project focus to models' performance on insufficient amounts of data. In this case, we will first focus on the FashionMNIST dataset.

Introduction

Language: For the programming language, I use python, specifically, pytorch for model training. I attached the link of the tutorial I used for the learning purpose in the reference list.

Dataset: FashionMNIST dataset contains 28*28 pixels image and each image belongs to a category of fashion product, (shown in Fig 1) The training set contains 60000 data points and the testing set contains 10000 data points.

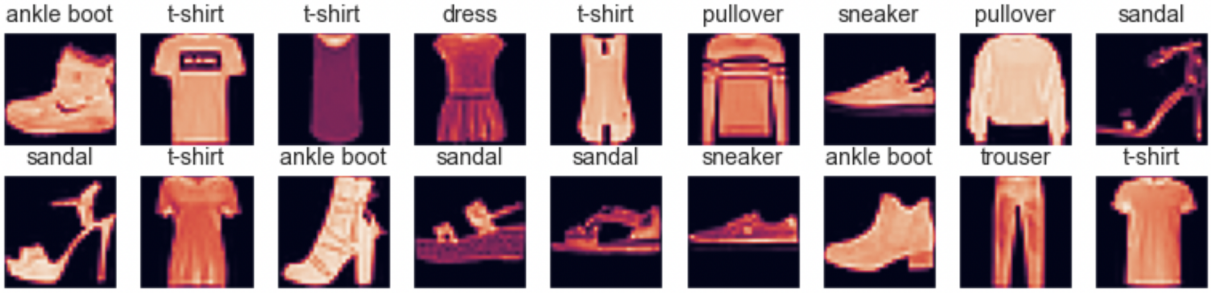

Fig 1

Experiment

In order to explore the advantages and drawback of standard transformer performance on Supervised image recognition. I apply three models (CNN,MLP,ViT) on the Fashion MNIST dataset and compare their loss, accuracy and time taken per epoch.

Candidates model

1) CNN:
   Here, I initially chose CNN as a baseline model. I trained 5 epochs in total with learning rate 0.001, and ReLU as the activation function.

```
ConvNet(
  (layer1): Sequential(
    (0): Conv2d(1, 16, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2))
    (1): BatchNorm2d(16, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU()
    (3): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  )
  (layer2): Sequential(
    (0): Conv2d(16, 32, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2))
    (1): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU()
    (3): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  )
  (fc): Linear(in_features=1568, out_features=10, bias=True)
)
```

2) MLP:
   For MLP, I used a three layers fully connected network, with gule as the activation function and with a learning rate of 0.003.

```
Mlp(
  (fc1): Linear(in_features=784, out_features=156, bias=True)
  (act): GELU()
  (fc2): Linear(in_features=156, out_features=156, bias=True)
  (fc3): Linear(in_features=156, out_features=10, bias=True)
  (drop): Dropout(p=0.0, inplace=False)
)
```

3) Vision transformer: I applied a standard transformer directly to images by splitting an image into patches and providing the sequence of linear embeddings of these patches as an input to the Transformer by treating these Image patches in the same way as tokens (words) in an NLP application. From the hugging face website, a vision transformer optimizer is built, and the ViTForImageClassification is used in my experiment.

Result

Cross Entropy Loss:

Use different numbers of epochs for different models, so here I only capture 3 epochs to see detect the convergence rate. It is obvious to see from figure 2 that CNN has the fastest convergence rate and Vision Transformer converges the slowestly in the first three epochs.

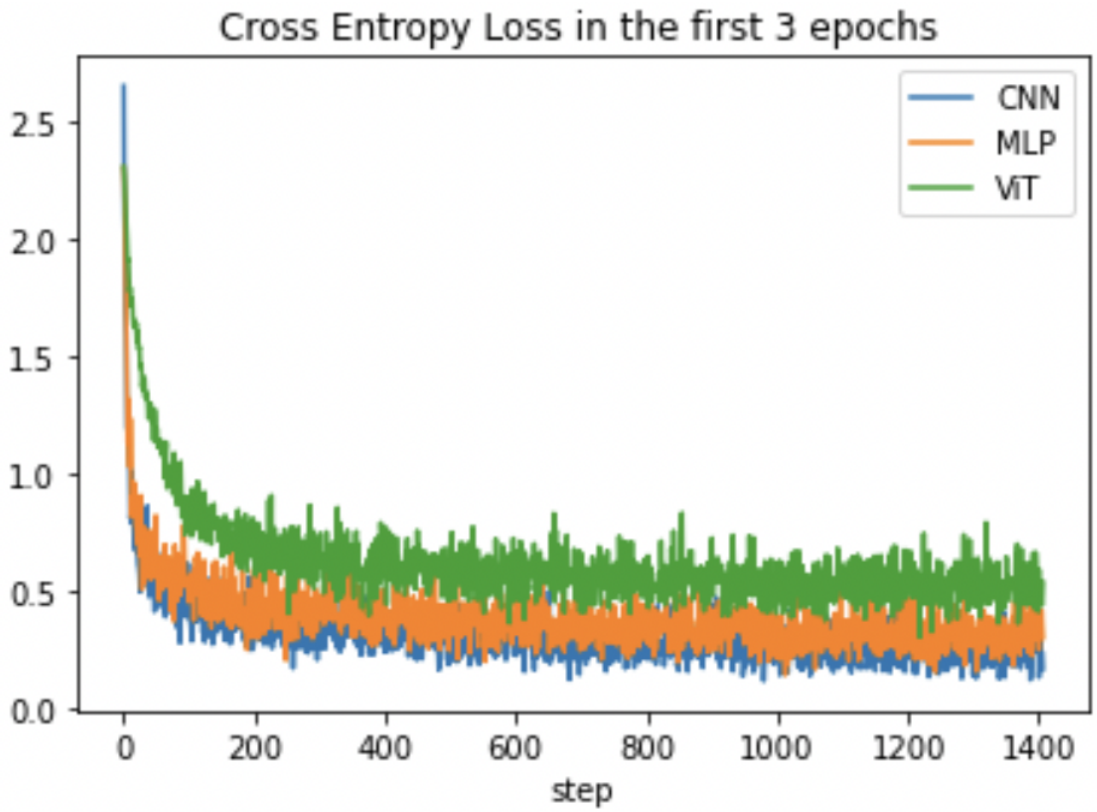

Fig 2

Time per epoch (in seconds):

In our experiment, each epoch takes 469 steps.

Although CNN gives a slightly better accuracy than MLP, both ViT and MLP are much faster than CNN per epoch. As shown in figure 3, CNN needs approximately 10 times more than MLP needs.

	accuracy	time_per_epoch(s)
CNN	0.8993	44.799968
MLP	0.8868	4.427626
ViT	0.8333	7.331787

Fig 3

Accuracy score:
Accuracy = (number of correct prediction)/(number of all testing data points)
As described in the abstract, CNN is the most popular used model for image classification and not surprisingly, it provides a generally good accuracy model. MLP gives similar accuracy. For the standard vision transformer, unfortunately, after some parameter tuning, the final accuracy turns out to be discouraging.

Result Analysis:
One result that might lead the disappointing performance of Vision transformer will be Transformers are not able to capture the inductive biases like CNN does, such as translation equivariance and locality, so when dealing with larger amounts of data (14M-300M images) , Vision Transformer (ViT) attains excellent results when pre-trained at sufficient scale and transferred to tasks with fewer data points. [Dosovitskiy, 2020]

Future Work

1) Apply the models from the experiment to Medical Dataset (X-ray images) from CovidNet open source data and compare the performance of it.
2) Try to modify the architecture of ViT classification to work better for insufficient data sets.

References

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby.An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *ICLR*, 2021.