Introduction

The ResNet-152 model was developed to solve the problem of Deep Learning where deeper networks tend to suffer from vanishing gradient problems and thus reduce accuracy and increase loss when number of layers are increased. (Fig. 1) To solve the problem, concepts of Residual Connection and skip connection were introduced which made deep neural networks possible. Many newer convnets have replaced ResNet and it is now used as a reference only.

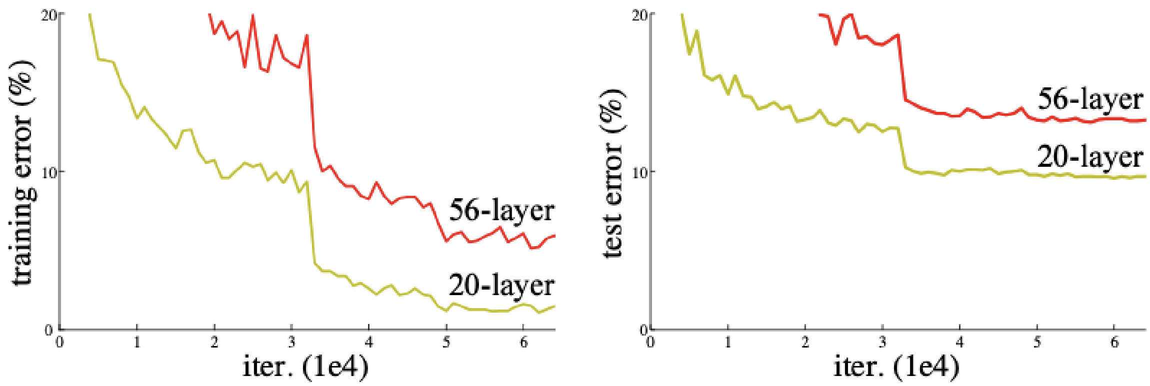

Fig (1.1)

Implications of training ResNet-152 from scratch

Resnet-152 is a very deep CNN with heavy compute. Training it from scratch on small datasets will cause models to memorize data. Therefore if one needs to use ResNet they should freeze all or most layers so that model can use its pretrained parameters which have already learnt about shapes and filters and different aspects of how to distinguish among images. Freezing layers let models use pretrained weights. It saves computations and overfitting.

Training and Validation Accuracy and Loss Comparison

Training loss decreases gradually from 0.7327 to 0.5695 and validation loss also decreases from 0.5853 to 0.4972 in first 7 epochs and then rises a little and stabilizes at 0.5654. Further training may have reduced it further. I was ablke to get 83.14% validation accuracy in just 10 epochs with all backbone freezed and only classification head active.

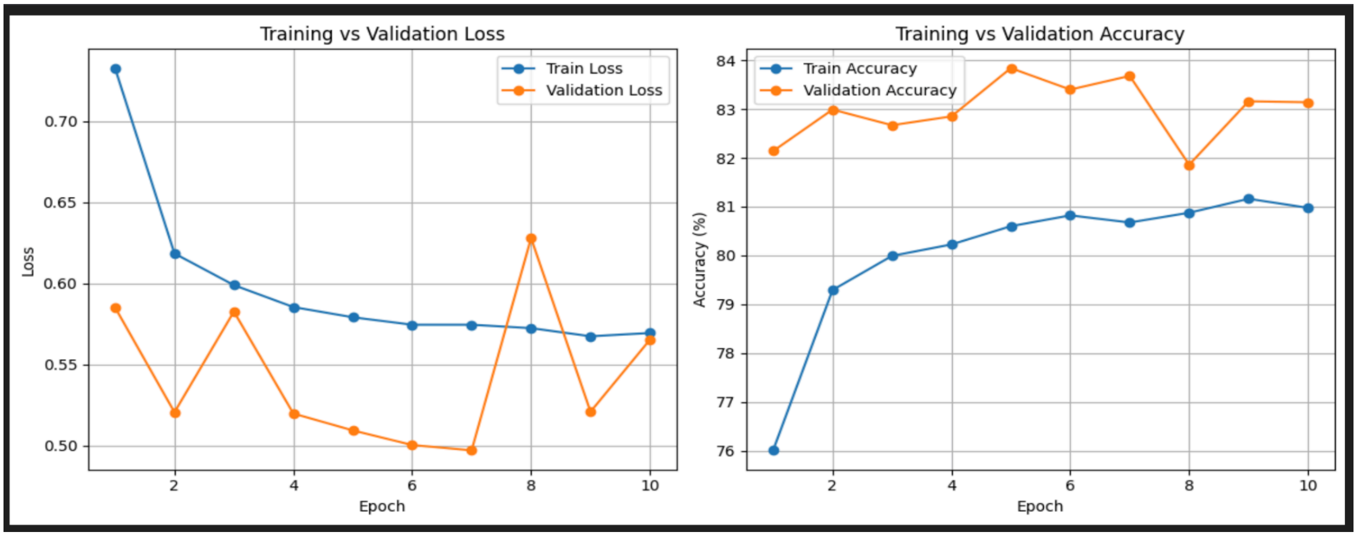

Fig (1.2)

Feature Hierarchies and Representations

I have evaluated the model architecture to understand the layers and how training different layers will impact the accuracy. t-SNE and UMAP representations of the dataset are shown in Fig (1.3 and 1.4). The architecture of layers with batch size 128 is:

Early —-—-> (128, 256, 56, 56),
Middle ——----> (128, 1024, 14, 14)
Late ——----> (128, 2048, 1, 1)

Clear clusters are visible with full fine tuning which in turn has increased accuracy to 96.32%.

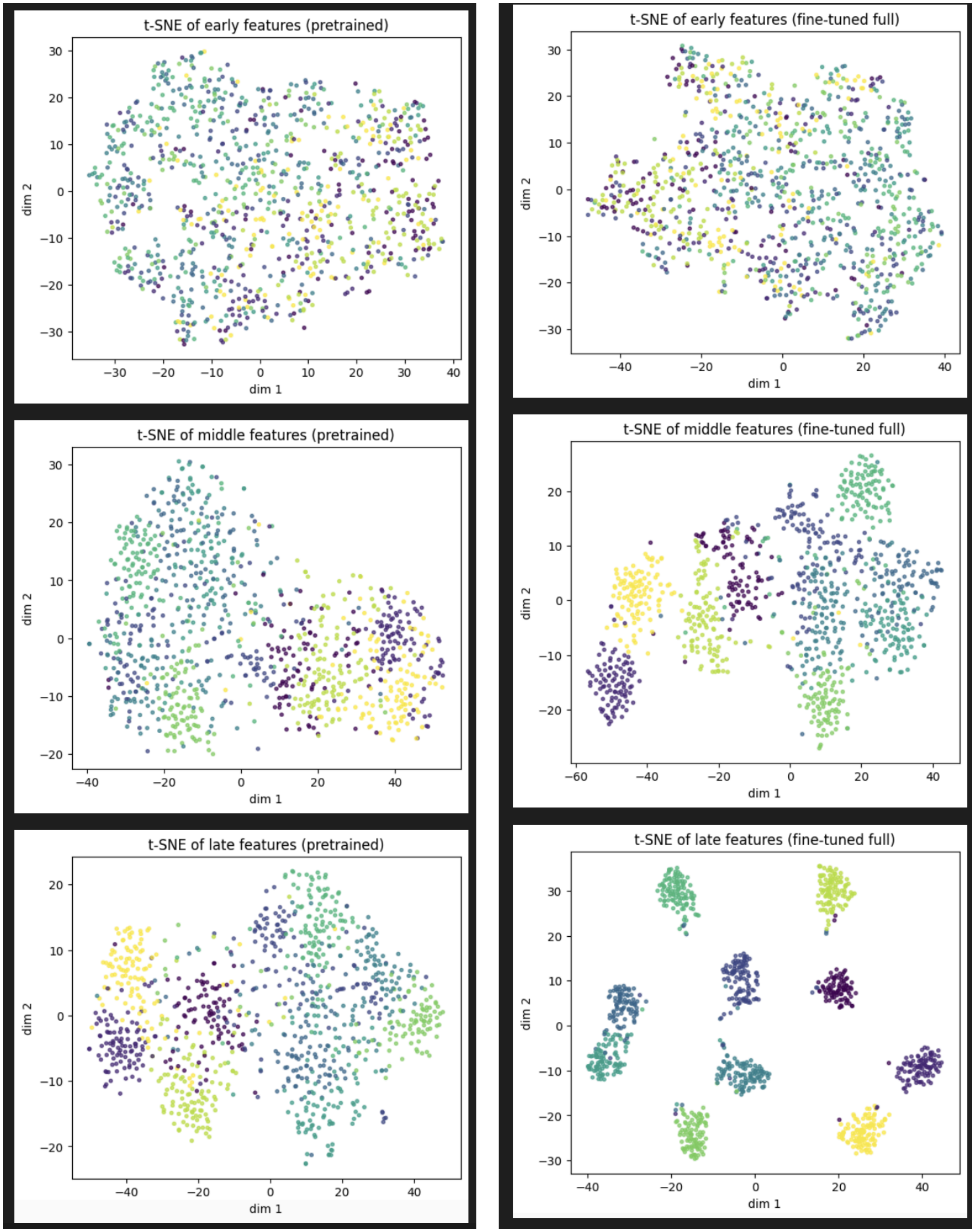
t-SNE of early features (pretrained)
t-SNE of early features (fine-tuned full)
t-SNE of middle features (pretrained)
t-SNE of middle features (fine-tuned full)
t-SNE of late features (pretrained)
t-SNE of late features (fine-tuned full)
dim 1
dim 2

Fig (1.3)

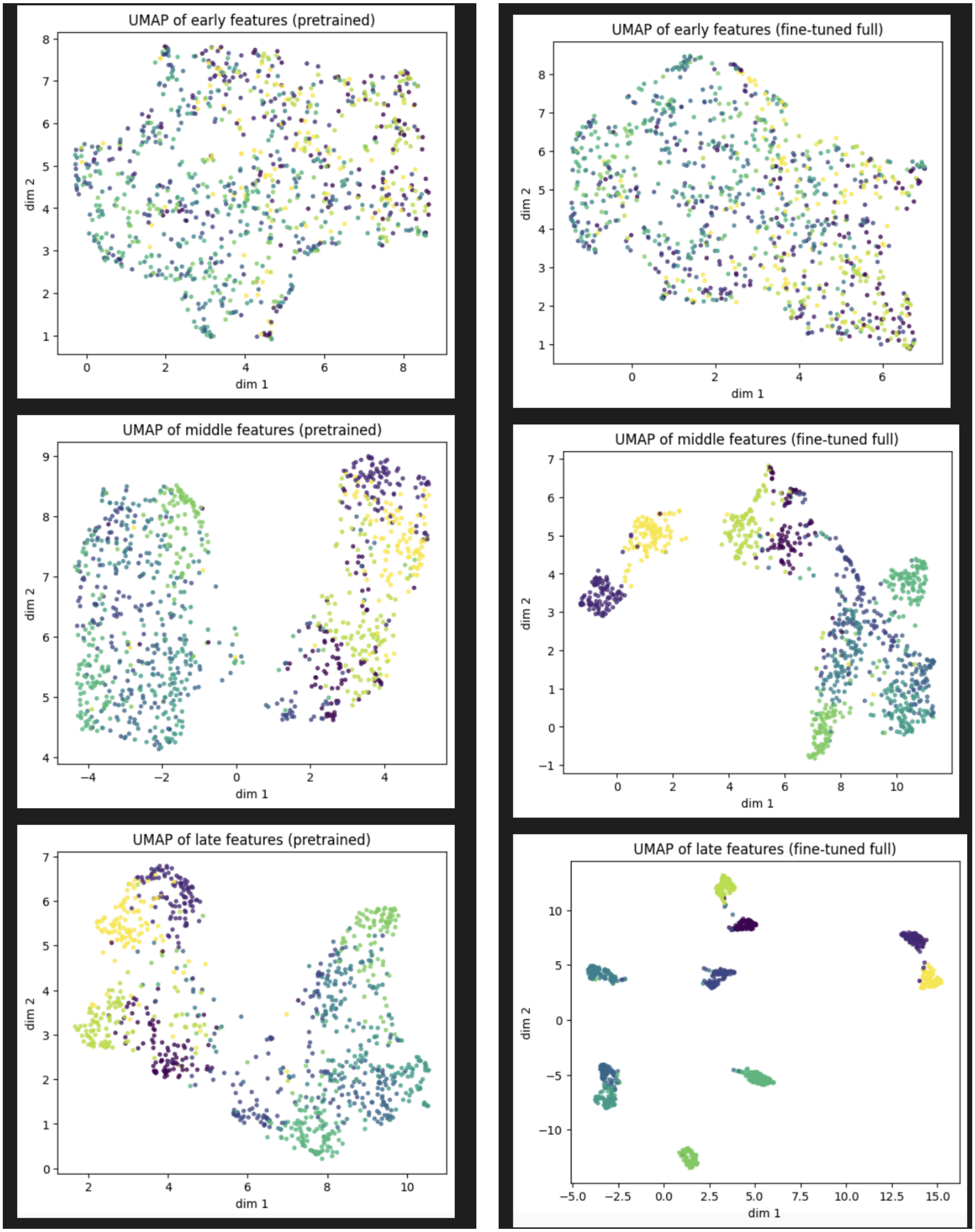
UMAP of early features (pretrained)
dim 1
dim 2
UMAP of early features (fine-tuned full)
dim 1
dim 2
UMAP of middle features (pretrained)
dim 1
dim 2
UMAP of middle features (fine-tuned full)
dim 1
dim 2
UMAP of late features (pretrained)
dim 1
dim 2
UMAP of late features (fine-tuned full)
dim 1
dim 2

Fig (1.4)

On the CIFAR-10 dataset, the results show a clear gap between using a pretrained model and starting from scratch. With linear probing, the pretrained model reached about 79.3% validation accuracy and 80.4% test accuracy, while random initialization performed poorly at only around 15% on both validation and test. Fine-tuning gave a big boost to the results, tuning just the last block improved accuracy to 91.0% on validation and 91.49% on the test set. Full fine-tuning worked best, achieving 96.34% validation accuracy and 96.32% on the test set. Overall, the results highlight how much pretrained models help and show that full fine-tuning leads to the strongest performance.