Mixture Models

Sai Kiran Maddela, Ian Trowbridge
University of Texas at Austin

Code Repo: https://github.com/trowk/CS395T_Final

Abstract

Using existing model architectures, we aggregate them together into "Mixture Models" to observe the effect of combining such models for image classification. We outline our results specifically for a Mixture Model made up of Convolutional Neural Network and Vision Transformer architectures. We observe and compare the resulting accuracy and runtime of the new Mixture Model versus the individual components and a baseline.

1. Introduction

In most deep learning tasks, different model architectures are used completely disjoint from each other. There are many image-classification models that have offered state-of-the-art results; however, there has not been much study on interactions between these models. We believe that combining different, well-established models into "Mixture Models" can lead to more robust models that encapsulate the best parts of the individual architectures. There are many pre-existing architectures that can be merged to observe a performance boost.

For instance, a vision transformer architecture and convolutional network share some differences. We can combine them into a Mixture Model. In this project, we do this in two ways - alternating vs joint. The alternating approach focuses on passed input from one architecture into another. One way of doing is this is by passing in the input of the convolutional network to the vision transformer which then passes it to another optional convolution network. Secondly, there is a joint variant of a Mixture Model. We simply pass inputs to both the convolutional network and the vision transformer and add their results together.

2. Related works

Vision Transformers (ViT). [1] are a more recent architecture introduced to utilize attention for image classification. The image is broken up into patches that are then transformed into embeddings. They still keep positional information by adding positional embeddings. We also used this architecture for the image classification Mixture Model.

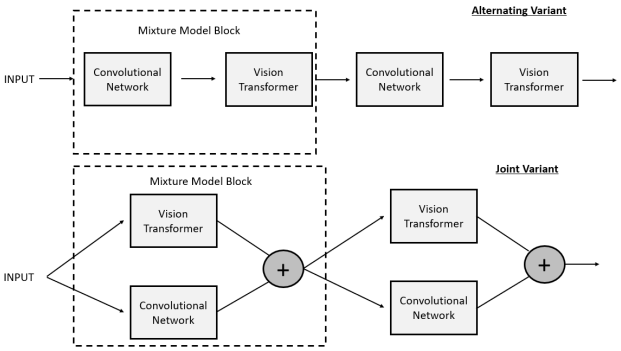

Figure 1. **Mixture Model Block Architectures –** The Alternating Variant is shown above followed by the Joint Variant.

Perceiver. [2]variant of transformer that modifies the attention module. The perceiver encodes the input to fixed-size latent arrays so as to allow it to scale to very large inputs. A transformer tower then maps this latent array to another latent array.

Convolutional Neural Network (CNNs). [3] are a popular approach used in deep learning to get state-of-the-art results particularly for image classification tasks. They are able to take advantage of the spatial location of pixels and are shift invariant. CNNs are also well suited for GPUs because they essentially involve matrix operations. We used a CNN in our Mixture Model blocks.

3. Method

3.1. Model Architecture

We used PyTorch for all model implementations.The ViT model was obtained from the "vit-pytorch" python package and the Resnet18 model was loaded from the "torchvision.models." We used a PyTorch implementation of Perceiver for a different Mixture Model by "lucidrains". Larger versions of these models formed baselines for our

Model	Top-1 Test Accuracy	Total Training Time (Minutes)	Number of Parameters
1 Block Alternating MM (Conv + ViT)	0.5987	4.6	4719167
2 Block Alternating MM (Conv + ViT)	0.5729	11.3	9407604
1 Block Joint MM (Conv + ViT)	0.5832	7.9	9708095
2 Block Joint MM (Conv + ViT)	0.62	19.88	19385460
Resnet 18 (Baseline)	0.5894	4.1	11689512
VIT (2 depth, 4 attention head)	0.293	7.1	10791946

Figure 2. **STL-10 (CNN + ViT) –** The Accuracy, Runtime, and other details are shown for each Model Tested. As shown, the Joint MM with 2 Blocks performed the best.

Model	Top-1 Test Accuracy	Total Training Time (Minutes)	Number of Parameters
Perceiver Baseline	0.404	125	1625586
Perceiver	0.431	50	301170
Joint (1 Block)	0.49	30	1466518
Joint (2 Block)	0.4243	70	2656540
Alternating (1 Block)	0.6438	100	1693278
Alternating (2 Block)	0.6146	300	3110066
CNN	0.4574	33	480054
ResNet-18	0.5699	--	11689512

Figure 3. **STL - 10 (CNN + Perceiver) –** The Accuracy, Runtime, and other details are shown for each Model Tested. As shown, the Alternating Mixture Model had the best results.

comparisons with our Mixture Model .

When implementing our Mixture Models for the alternating variant, we formed a Mixture Model block as a CNN followed by a ViT. The output of the ViT was then either fed to a linear layer to classify the inputs, or it was fed into the next Mixture Model block. The CNN in our Mixture Model Block was composed of 5 layers (Conv2d + BatchNorm + ReLU) and residual connections between layers, where we increased the number of channels from 3 to 128 before outputting a 3-channel tensor. This output was fed through a ViT model with a depth of 2 and 2 attention heads. The output dimension for ViT was set to 1024 and the Multilayer Perceptron (MLP) dimension was set to 2048. Before passing the output of the ViT, we passed it along another convolutional layer that reduced channel size from 256 to 3 to have the output resemble an image.

We also created a Joint-variant of our Mixture Models. The Joint-variant had the outputs of the CNN and ViT models added together before being sent through a Feed-Forward Network (FFN). The ViT model was kept the same as the alternating variant, but the CNN lacked the residual connection. We also used a similar convolutional layer to convert the ViT output to have 3 channels.

For the CNN + Perceiver Mixture Model, the model architecture is roughly the same as detailed above, except that the ViT is replaced by Perceiver.

3.2. Originality

After extensive literature review, we found limited published work that pertain to the idea of Mixture Models discussed in this paper. The act of combining completely different architectures like in this report is not prevalent in Deep Learning.

3.3. Fit

Overall, Mixture Models were an appropriate solution to the problem of image classification. By combining these models in either the Joint or Alternating variants, we were able to achieve results similar to the baseline models in our experiments. This even occurred in instances where our Mixture Models had as little as half as many parameters as our baseline. We were even able to see that Mixture Models improved certain aspects of the individual components - such as getting results better than our ViT baseline while converging much faster. Refer to Section 4 for more in depth results.

3.4. Breath - Alternatives

We created and tested different kinds of Mixture Models. For instance, the number of blocks in a Mixture Model is a hyperparameter that we considered in our experiments. We also tried out different transformer architectures namely the Perceiver and the Vision Transformer architectures. There

is scope to try out all kinds of architectures and construct a variety of Mixture Models. Computational limits played a big role in constraining the number of different architectures we could have tried.

3.5. Training Details

For our training, we used the ADAM optimizer with a weight decay of 10^{-4}. Since this is an image classification task, we used Cross Entropy Loss. The batch size depended on the model we tested, as well as with the number of epochs per experiment. The learning rate was kept constant at 10^{-3} for most experiments, while set to $3 * 10^{-4}$ for the Perceiver experiment.

4. Results

4.1. Experiments

Our experiments were conducted on the CIFAR-10, CIFAR-100, and STL-10 datasets. We experimented with the performance of alternating vs joint single block mixture model against the baseline ResNet18 and ViT models which have similar number of model parameters. The 1-block Alternating used half the parameters of ResNet18 to keep the training time comparable. We also compared the two-block mixture models against the baseline models. The training times and number of parameters of the individual models were also recorded. All validation accuracies reported are the highest achieved validation accuracies found during runtime.

4.2. CIFAR-10

On the CIFAR-10 dataset, the Alternating Models, the 2-block joint model, and the ResNet18 model all got similar results with the ResNet18 model having the highest validation accuracy at 0.7915. The ResNet18 also overfitted the most, achieving a training accuracy of 0.9327 while the next highest training accuracy was 0.8631 with the 1-block Joint model. The ViT model severely underperformed when compared to the other models tested.

4.3. CIFAR-100

The CIFAR-100 dataset achieved similar results to the CIFAR-10 dataset. Only single block models were tested on this dataset due to time constraints, and the ResNet18 model achieved the highest validation accuracy of 0.4703. The 1-block Alternating model achieved the next highest validation accuracy of 0.4511 despite having half the parameters of ResNet18. The ViT model once again underperformed in comparison to the other models.

4.4. STL-10

Due to limited computational resources, we resized the STL-10 images from 96×96 to 32×32 size images before training and inference. In this dataset, the 2-block Joint model and the 1-block Alternating model were able to outperform the 0.5894 validation accuracy achieved by ResNet18 with validation accuracies of 0.6200 and 0.5988 validation accuracies respectively. The ResNet18 model also overfitted the most achieving a training accuracy of 0.9454 while the next highest training accuracy occurred with the 2-block Alternating model at 0.8792 training accuracy.

4.5. Ablation

We created a variation of our Mixture Models where we replaced the ViT model with a Perceiver and ran tests on the STL-10 dataset. In this experiment, in addition to training our 1-block and 2-block mixture models with the Perceiver, we also trained a large Perceiver model as an additional baseline and trained the individual components of the Mixture Models separately (the CNN and Perceiver components that are combined to form a mixture model). We found that our Alternating Models outperformed both the ResNet18 and Perceiver baseline and that the individual components of the Mixture Models underperformed when compared to the Alternating models and the 1-block Joint model.

Another ablation we ran was specifically with the Joint Mixture Model. We tried to study the effects of normalizing individual component network's output. To our surprise, this led to a considerable performance increase. We believe this is due to the individual networks producing outputs at different scales. Since we were directly adding them without normalization, the results we were obtaining were suboptimal. The results from our experiments pertaining to the Joint Mixture Model were with this normalization modification.

5. Conclusion

5.1. Future Work

We can translate this work to more challenging datasets and different tasks (i.e. NLP). There are more combinations of architectures that we did not try out that could be explored. Certain implementation decisions pertaining to the Mixture Models could be fine-tuned for potentially better results. Lastly, with more computational resources, we can test out Mixture Models with more blocks and study the impact of larger individual model architectures.

5.2. Closing

Our Mixture Models were able to outperform baseline models even with fewer parameters. Overall, there is more that can be done for this particular topic with regards to expanding it for different tasks and architectures. We believe this project provides significant evidence to support the idea

that combining different architectures into one is a viable approach.

References

[1] Alexey Dosovitskiy and Lucas et. al. Beyer. An Image is Worth 16 x 16 Words: Transformers for Image Recognition at Scale. 2021. 1

[2] Andrew Jaegle and Felix et. al. Gimeno. Perceiver: General Perception with Iterative Attention. 2021. 1

[3] Yann LeCun, Patrick Haffner, and Yoshua Bengio. Object Recognition with Gradient-Based Learning. 1999. 1

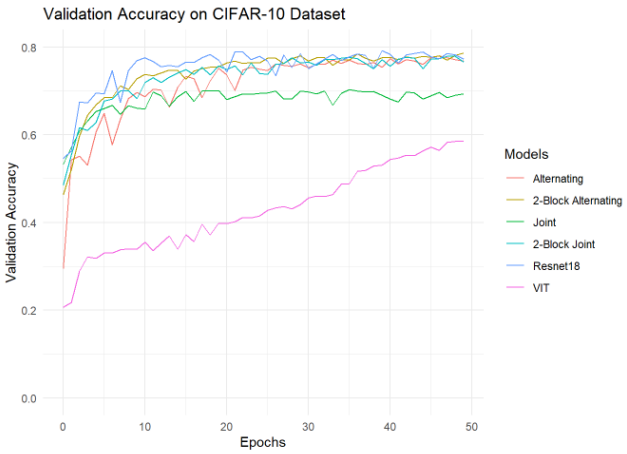

Figure 4. **CIFAR-10**

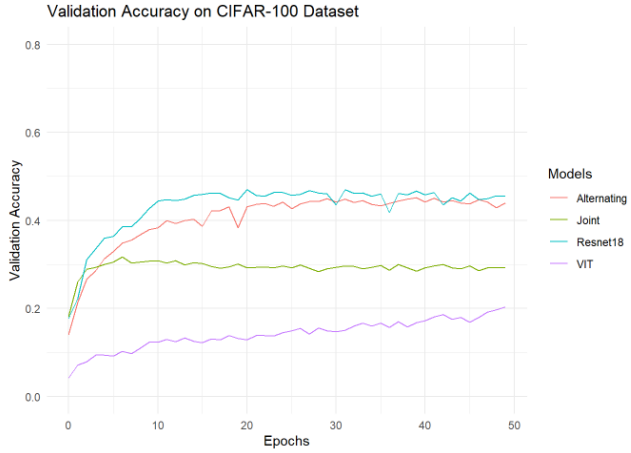

Figure 5. **CIFAR-100**

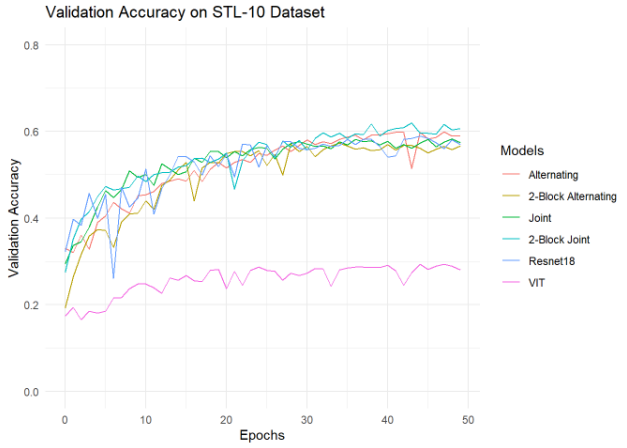

Figure 6. **STL-10**

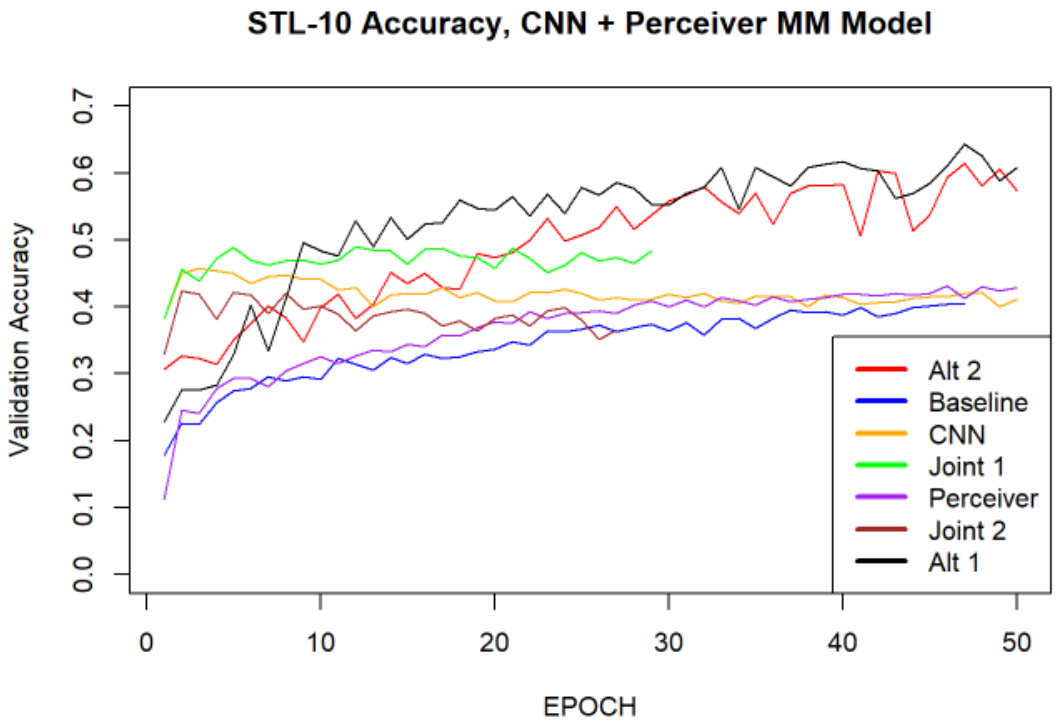

Figure 7. **Perceiver Experiments Graph**

Model	Top-1 Test Accuracy	Total Training Time (Minutes)	Number of Parameters
1 Block Alternating MM (Conv + ViT)	0.7797	49.6	4719167
2 Block Alternating MM (Conv + ViT)	0.786	117.5	9407604
1 Block Joint MM (Conv + ViT)	0.7026	77.5	9708095
2 Block Joint MM (Conv + ViT)	0.7795	535.9	19385460
Resnet 18 (Baseline)	0.7915	35.8	11689512
VIT (2 depth, 4 attention head)	0.5834	72.1	10791946

Figure 8. **CIFAR-10 Results Table**

Model	Top-1 Test Accuracy	Total Training Time (Minutes)	Number of Parameters
1 Block Alternating MM (Conv + ViT)	0.4511	46.1	4995737
1 Block Joint MM (Conv + ViT)	0.3169	77.2	9984665
Resnet 18 (Baseline)	0.4703	42.9	11689512
VIT (2 depth, 4 attention head)	0.203	70.1	10884196

Figure 9. **CIFAR-100 Results Table**