Pruning Vision Transformers

1. Problem Statement

Vision Transformers (ViTs) have demonstrated exceptional performance in visual tasks due to their ability to model global context through self-attention. However, this comes with significant computational and memory costs, as all image patches are processed equally, even when some may provide redundant information. MHSA computations grow quadratically with the number of patches, leading to inefficiency. This inefficiency motivates the need for a mechanism to identify and skip redundant patches without compromising the model's classification performance. Many of the existing solutions focus on accuracy and computation. The goal is to develop a novel pruning approach for ViTs, for a trade-off between computation and accuracy while providing valuable insights into the model's explainability.

2. Proposed Solution

Our approach introduces a lightweight MLP for dynamic patch pruning in Vision Transformers. During training, scaled cosine similarity between patches across consecutive layers is computed to generate binary labels (keep or skip) based on a predefined cosine threshold. The MLP is trained using cross-entropy loss to predict pruning scores, to mimic the cosine similarity property.

At the inference, the MLP predicts pruning scores for each patch, which are compared against a score threshold to decide whether to retain or skip patches. Skipped patches can be reintroduced in subsequent layers, ensuring adaptability. This method leverages cosine similarity to identify redundant patches, balances computation-accuracy trade-offs, and enhances model explainability by revealing patch-level relevance dynamically.

3. About Dataset

CIFAR-100 is a widely used benchmark dataset in machine learning, consisting of 60,000 32x32 color images categorized into 100 classes. It is divided into 50,000 training images and 10,000 test images. The dataset provides a diverse set of images with 600 images per class, designed to evaluate classification performance.

4. Review of Existing Literature

Some of the previously done work, which greatly helped in building our approach.

- **ViT with Patch Diversification:** Deeper layers in Vision Transformers (ViTs) exhibit high cosine similarity among patch embeddings, leading to a loss of discriminative features. To mitigate this, the approach introduces novel loss functions aimed at reducing cosine similarity and promoting patch diversity.
- **Patch Slimming:** This method identifies effective patches in the final layer and uses them to guide pruning in preceding layers. It evaluates patch contributions to output features, discarding less impactful patches.
- **IARED^2 (Interpretability-Aware Redundancy Reduction):** A model-agnostic and task-agnostic approach dynamically removes uncorrelated patches using a policy network (multi-head interpreter). By identifying and discarding uninformative patches, it achieves up to 1.4x speed-up with less than 0.7% accuracy loss for DeiT.
- **Anti-Oversmoothing in Deep Vision Transformers:** Deeper Vision Transformers suffer from attention collapse, where patch embeddings become highly similar due to low-pass filtering of high-frequency details in self-attention. This suppresses localized details and increases cosine similarity, resulting in coarse representations and loss of finer features.
- **Skip Attention: Improving Vision Transformers by Paying Less Attention:** This paper explores the inefficiencies of Vision Transformers by proposing a method to skip attention in less critical areas, improving computational efficiency without sacrificing accuracy.
- **No Token Left Behind: Efficient Vision Transformer via Dynamic Token Idling:** This approach introduces dynamic token idling, where unselected tokens are kept idle and can re-enter subsequent layers, preventing permanent loss of crucial information during early-stage pruning.

5. Our Approach and Implementation

5.1. Background for the approach

The high cosine similarity observed in deeper Vision Transformer layers suggests redundancy in patch embeddings, where patches become less discriminative as layers deepen. We can observe this trend in the figure 1. This phenomenon, often referred to as oversmoothing, indicates that certain patches contribute minimally to the model's predictions, and they are not updated much.

The average cosine similarity increases in the later layers of ViT, indicating convergence to a unified representation. Middle layers show a dip, reflecting more diverse and decorrelated feature representations, which can be depicted in the figure 2.

Figure 1. Fig: Cosine similarity between same-positioned patches across adjacent layers in CIFAR-100

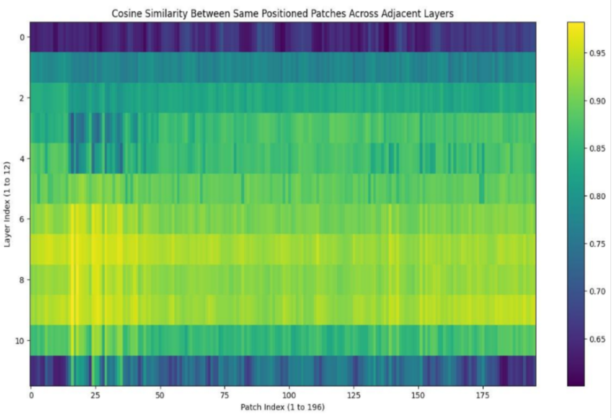
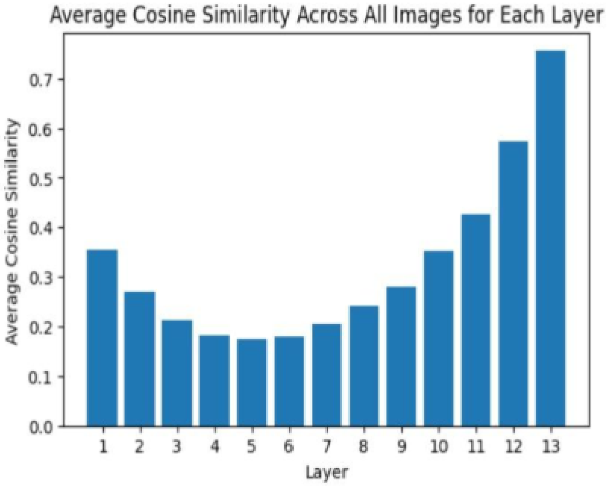


Figure 2. Fig: Layer-wise average cosine similarity (CIFAR-100)



5.2. Our Approach: Pruning with MLP

Proposed Workflow (Training)

1. Compute scaled cosine similarity between corresponding patches across layers i and $(i + 1)$.
2. Generate Labels Based on cosine_threshold:
   - Label = 0 (Skip Patch), if the scaled cosine similarity ¿ cosine_threshold.
   - Label = 1 (Keep Patch), otherwise.
3. Use a lightweight MLP to predict a pruning score between [0, 1] for each patch, which is inversely proportional to the scaled cosine similarity score.
4. Compute the Cross-Entropy loss between the generated labels (from step 2) and the MLP-predicted scores. Backpropagate the loss to optimize the MLP.Note: Skipped patches, bypassed for computation in the current layer, can be reintroduced in subsequent layers.

Proposed Workflow (Inference)

1. For each patch, feed it into the lightweight MLP to compute the pruning score.
2. Apply Threshold for Binary Decision:
   - Keep Patch (Label = 1): If the MLP score ¿ score_threshold, patch retained for further computation.
   - Skip Patch (Label = 0): If the MLP score $\leq$ score_threshold, patch bypassed for the current layer.
3. Accuracy Calculation:
   - Compute the scaled cosine similarity for each patch.
   - Generate ground truth labels based on the scaled cosine similarity and cosine_threshold.
   - Compare the ground truth labels with the predicted labels from the MLP to compute accuracy.

Characteristics

- **Dynamic Pruning of Unevolving Patches:** High cosine similarity identifies redundant patches that are not evolving in terms of features.
- **Dynamic Evaluation:** Our MLP-based mechanism dynamically evaluates patch relevance at each layer.
- **Reintroduction of Idle Patches:** Unlike methods that permanently drop tokens/patches, idle patches are passed to subsequent layers for reevaluation.

5.3. Challenges Faced

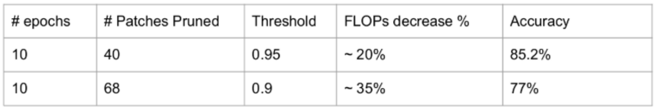
1. **Variable Patch Count:** Pruning reduces the number of patches per image, leading to varying patch counts across a batch. This inconsistency complicates batch processing and model training.
2. **Modified Attention Mechanism:** The bypassing of patches was based on the fact that they are not updating themselves, but unpruned patches may still need to attend to all patches, including the bypassed ones while updating themselves. To address this, we had to modify PyTorch's Attention function (Q, K, V) accordingly to properly ensure that the unpruned patches can fully leverage the context from all patches.

3. **Suboptimal Pruning Results:** The method reduced computational cost but did not achieve state-of-the-art performance in image classification tasks.
4. **MLP Limitations:** The MLP used to predict pruning scores struggles as only a single patch is passed, as it lacks contextual information. However, passing all patches to the MLP for context will increase computation significantly, decreasing the benefits of pruning.

5.4. Results:

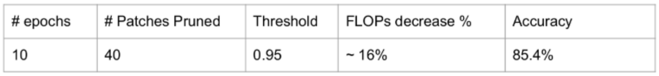
MLP architecture : 768 x 64 x 1
Accuracy on CIFAR - 100 without pruning : 89.8 %

Figure 3. Fig:: Accuracy of VIT and Average patches pruned at different thresholds



# epochs	# Patches Pruned	Threshold	FLOPs decrease %	Accuracy
10	40	0.95	~ 20%	85.2%
10	68	0.9	~ 35%	77%

Figure 4. Fig:: Accuracy of VIT and Average patches pruned at threshold = 0.95 using modified attention



# epochs	# Patches Pruned	Threshold	FLOPs decrease %	Accuracy
10	40	0.95	~ 16%	85.4%

5.5. Observations

We can see that, figure 5 and figure 6 shows that, our model is able to get the exact bounding of the class object in the image.

1. **Progressive Skipping Across Layers:** Skipped patches (blacked-out areas) gradually increase in deeper layers, starting significantly from the 7th layer onward. This suggests that deeper layers handle more redundant patches, as is supported by the increasing cosine similarity plot we saw.
2. **Early Layers Retain More Information:** In the initial layers (1st to 6th), most patches are retained, which may indicate that initial layers focus on extracting general features. Pruning aggressively at these stages might harm performance.
3. **Effective Skipping at Layers 8 and 10:** Observing the outputs at layers 8 and 10, the pruning strategy seems to correctly black out irrelevant patches while retaining critical ones for prediction.

6. Conclusion

This project highlights the importance of patch stability analysis as a key factor for understanding the dynamics of Vision Transformers and guiding pruning methodologies. The results underscore the potential of similarity based metrics in identifying redundant or less-informative or idle patches across layers. The study and work opens doors for improving efficiency and performance in dense prediction tasks.

Figure 5. Fig:: Visualisation of Skipped Patches at each layer(CIFAR-100 Snake)

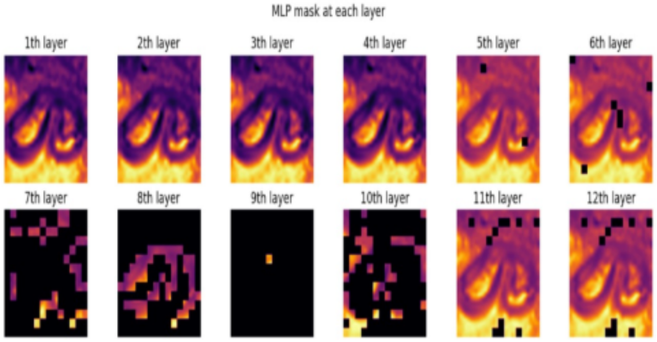
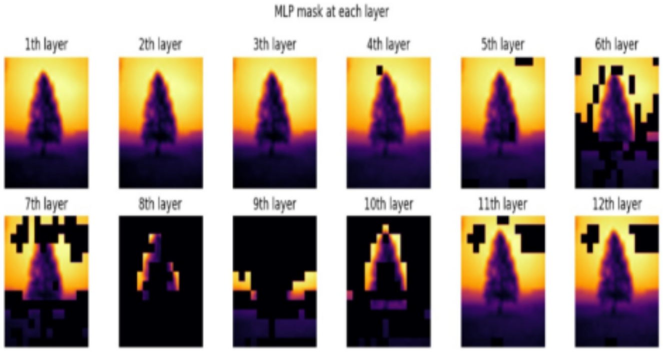


Figure 6. Fig::Visualisation of Skipped Patches at each layer(CIFAR-100 Tree)



7. Resources

- Patch slimming: [arxiv:2106.02852]
- XPruner: [arxiv:2303.04935]
- IARED² (Interpretability-Aware Redundancy Reduction): [arxiv:2106.12620]
- ViT with Patch Diversification: [arxiv:2104.12753]
- Anti-Oversmoothing in Deep Vision Transformers: [arxiv:2203.05962]
- Skip Attention: Improving Vision Transformers by Paying Less Attention: [arxiv:2301.02240]
- No Token Left Behind: Efficient Vision Transformer via Dynamic Token Idling: [arxiv:2310.05654]