Advanced Topics in Machine Learning (PA0): ResNets, Vision Transformers, and GANs

Sameer Rahil [1]

Abstract

This report documents experiments conducted for the Programming Assignment 0 (PA0) of Advanced Topics in Machine Learning course offered by Sir Tahir, instructor at the Computer Science department at the Lahore University of Management Sciences. The assignment focuses on three main areas: (1) the inner workings and transfer learning properties of ResNet-152, (2) interpretability and robustness of Vision Transformers (ViTs), and (3) training dynamics and common pathologies of Generative Adversarial Networks (GANs). For each task, I will outline methodology, report results, and provide discussion. All code and reproducible experiments are available at: `https://github.com/SameerRahil/Advanced-Topics-in-ML-PA0.git`.

1. Introduction

Deep learning models such as Convolutional Neural Networks (CNNs), Vision Transformers (ViTs), and Generative Adversarial Networks (GANs) form the foundation of modern computer vision and generative modeling. This assignment explores three aspects:

- **Task 1:** Understanding the transferability, residual connections, and feature hierarchies of ResNet-152. I froze the backbone and trained the model on only the classification head, expanding it towards disabling of skip connections.

- **Task 2:** Exploring interpretability, visualizing and analyzing the attention maps, and robustness in Vision Transformers. I also went on to mask a fraction of input patches and observed the effects on accuracy, and performed linear probing and mean of patch tokens to see which pooling method performs better.

- **Task 3:** Building a GAN on MNIST and analyzing common training issues: vanishing gradients, mode collapse, and discriminator overfitting. I tweaked the Generator and Discriminator in a series of steps to see which one fools the other one, and if the Discriminator is successful in identifying fake versus real images.

The objective is to gain a deeper understanding of model behavior, interpretability, and stability of training under different perturbations.

2. Task 1: Inner Workings of ResNet-152

2.1. Methodology

I used a pre-trained ResNet-152 from PyTorch, replacing the final classification head with the one for CIFAR-10 instead of the original one. Experiments included:

1. Training only the head with frozen backbone.
2. Disabling skip connections in selected blocks.
3. Extracting features from early, mid, and late layers for visualization.
4. Comparing head-only fine-tuning vs. last block vs. full fine-tuning vs. training from scratch.

2.2. Results

- **Baseline vs. No-skip:** Baseline reached $\sim$79.14% validation accuracy after 1 epoch, no-skip dropped to just$\sim$27.29%.

- **Feature Hierarchies:** t-SNE showed poor separability in early features, partial clustering mid-network, and clear clusters in late layers (layers that I observed were layer 1, layer 3, and layer 4).

- **Transfer Learning:** Pretrained models outperformed random initialization. Best trade-off that I observed was fine-tuning only the last block.

[1] Department of Computer Science, Lahore University of Management Sciences, Lahore, Pakistan. Correspondence to: Sameer Rahil <sameerrahil07@gmail.com>.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025. Copyright 2025 by the author(s).

2.3. Discussion

- **Skip Connections:** A residual block outputs $y = F(x) + x$, where $F(x)$ is some layers of convolutions + ReLUs and the skip adds the input back unchanged. When you back-prop a loss L, the gradient to the block's input is : (chain rule):

  $$\frac{dL}{dx} = \frac{dL}{dy} \cdot \left(\frac{dF}{dx} + I \right).$$

  Without the skip connection:

  $$\frac{dL}{dx} = \frac{dL}{dy} \cdot \frac{dF}{dx}.$$

  So, the $+I$ identity term is an identity gradient path - "a highway", so even if $\frac{dF}{dx}$ is small the identity keeps gradient from vanishing. With skips, it is faster, more stable convergence in very deep nets, giving typically a higher validation accuracy. Removing skips causes gradients to must pass only through $F(\cdot)$ so optimization becomes harder and hence a slower convergence and usually lower accuracy.

- **Feature Hierarchy:** I extracted features from early layer (1), middle layer (3), and a late layer (4) of ResNet-152 with frozen backbone on CIFAR-10. Using forward hooks I collected per-image activation and flattened them, and visualized each layer's feature space with t-SNE.

  The early layer features (edges/textures) showed limited class separability, points from different classes were mingled. Mid-level features showed partially formed clusters. Late layer features showed clearest clusters, indicating early layers capture low-level cues, while deeper layers integrate parts into object-level abstractions like dog vs. truck vs. cat.

- **Transferability:** I compared four settings on CIFAR-10 using ResNet-152: (i) pretrained + head-only, (ii) pretrained + last block (layer4), (iii) pretrained + full fine-tune, (iv) random init + full. Validation accuracies after one epoch were: 70.34%, 71.01%, 74.82%, and 14.93% respectively.

  Pretrained models clearly outperformed training from scratch. Fine-tuning only the last block achieved accuracy close to full fine-tuning but with less compute, making it the best trade-off. Early and middle layers (edges, textures, motifs) are highly transferable across datasets, while late layers are task-specific and benefit from fine-tuning. Thus, unfreezing the final block provides a good balance between efficiency and accuracy. Training from random initialization underperformed because CIFAR-10 is too small relative to model capacity.

3. Task 2: Understanding Vision Transformers

3.1. Methodology

I used HuggingFace's ViT-Base/16 model with ImageNet pretraining. Steps:

1. Classify test images (resized to 224×224).
2. Extract last-layer CLS→ visualizing patch attentions, reshape to 14×14, upsample, and overlay as heatmaps.
3. Mask patches randomly vs. structurally (center).
4. Train linear probes on CLS token vs. mean of patch embeddings.

3.2. Results

- **Classification:** Predictions of my model were reasonable (e.g., "tabby cat," "speedboat" were identified correctly).
- **Attention Maps:** Heatmaps appeared blurry/diffused/blocky, often highlighting background context (e.g., water for speedboats).
- **Masking:** Random masking preserved predictions; structured (center) masking caused misclassifications in most cases.
- **Probes:** Mean-pooled patches achieved slightly higher accuracy (95.40% vs. 95.10%) though CLS had lower loss.

3.3. Discussion

- **Attention Maps:** They turned out to be coarse due to 14×14 patches (blurry) and the averaging of heads washed out structure. Unlike Grad-CAM, ViT attention is built-in but less spatially sharp, as observed in the outputs. The intensity of red on the images was directly proportional to the attention points, and to my surprise, the model was giving attention to patches that I otherwise would not have guessed. For example, for the speedboat, it was highlighting mainly the water body under the boat (and some features of the boat such as edges and shape).
- **Masking Robustness:** Self-attention distributes information throughout, so random missing patches degrade gracefully and prediction is still relevant and accurate. However, masking in the center center removes critical evidence, as many objects in an image are usually centred.
- **CLS vs. Mean Pooling:** CLS is optimized for classification during supervised pretraining; mean pooling benefits more under self-supervised pretraining.

4. Task 3: Understanding GAN Dynamics

4.1. Methodology

We trained an MLP-GAN on MNIST using Binary-Cross-Entropy loss, Adam ($\eta = 2e{-}4, \beta_1 = 0.5, \beta_2 = 0.999$), with label smoothing (0.9 for real). Generator: 100→256→512→784. Discriminator: 784→256→256→1 with LeakyReLU.

I conducted baseline training and three perturbations:

1. Overpowered D (vanishing gradients).
2. Overpowered G (mode collapse).
3. Reduced dataset (D overfitting).

4.2. Results

Baseline: I observed that losses oscillated as expected and digits became recognizable after training, with moderate diversity. Compared to PyTorch-GAN reference, our digits were blurrier, due to smaller network capacity and fewer epochs (I only trained on 20 epochs, not the ideal number).

Vanishing Gradients:

- D's loss →0, D(x) ≈1, D(G(z)) ≈0.
- G's loss flat/high (stalled learning).
- Fix (label smoothing, non-saturating loss) restored gradient flow; G's loss decreased and our samples improved.

Mode Collapse:

- With high G learning rate and weak D, generator outputs collapsed to near-identical digits.
- Diversity metrics dropped, the printed sample grids confirmed repetition.
- Fix (stronger D, balanced updates, with D having higher learning rate) restored the variety in outputs.

Discriminator Overfitting:

- Training on 1000 MNIST samples for 5 epochs showed no strong overfit gap (train/val close).
- Adding Dropout(0.4) kept D uncertain, preventing drift toward memorization.

4.3. Discussion

- Across training, the discriminator loss dropped quickly at the start (D learns to separate real vs. initial noise), then oscillated around a band rather than converging to zero; the generator loss increased initially (because D is strong early), then also settled into oscillations. This is the expected adversarial dynamics: as G improves, D's job gets harder (D loss rises), and when D improves, G's loss rises—so both losses move in counterphase rather than monotonically decreasing. The scalar confidences told the same story: $D(x)$ on real hovered high (typically ~ 0.8–0.95), while $D(G(z))$ on fakes stayed lower (e.g., ~ 0.1–0.3) but fluctuated as G learned. Brief spikes and small oscillations are normal; persistent saturation (D loss ≈ 0, $D(G(z)) \approx 0$) would indicate vanishing gradients, which we avoided with Adam ($\beta_1 = 0.5$) and the non-saturating G loss (and optional real-label smoothing).

- In the vanishing gradients experiment, when the discriminator (D) was overpowered, its accuracy rapidly saturated close to 100% ($D(x) \approx 0.99$, $D(G(z)) \approx 0.01$) and its loss dropped near zero. As a result, the generator's (G) loss curve remained high and flat, showing that G had stalled and was unable to learn. After introducing one-sided label smoothing (real labels = 0.9) and switching to the non-saturating loss $L_G = -\log D(G(z))$, G's loss began to decline and sample quality improved, while D's loss stabilized at higher values ($D(x) \approx 0.8$–0.9, $D(G(z)) \approx 0.2$–0.4), confirming that gradients were more informative. In the mode collapse setup, increasing G's learning rate led to low diversity: generated grids repeatedly showed near-identical digits (e.g., many "3"s with similar strokes), and diversity metrics flattened. By rebalancing with stronger D updates (2:1 ratio) and reducing G's learning rate, sample variety was restored and collapse mitigated. Finally, when examining discriminator overfitting with a reduced dataset (1,000 MNIST images), both train and validation scores decreased similarly (e.g., D(real) fell from ~ 0.92 at epoch 1 to ~ 0.58 at epoch 5 without dropout, and to ~ 0.55 with dropout), indicating no severe memorization in this short run. Nonetheless, adding dropout regularization kept D more uncertain, providing steadier gradients for G and preventing overconfidence. Overall, these results demonstrate three key GAN pathologies: vanishing gradients when D saturates, mode collapse when G exploits a weak D, and overfitting when D memorizes a small dataset. In each case, targeted remedies such as label smoothing, non-saturating losses, diversity-promoting tricks, or dropout kept the adversarial dynamics balanced and improved the generator's learning signal.

5. Conclusion

This assignment deepened understanding of three key ML paradigms. For ResNets, we confirmed the necessity of skip connections for gradient flow and showed feature hierarchies' evolution. For ViTs, we explored interpretability via attention, robustness to missing patches, and pooling strategies. For GANs, we reproduced expected adversarial dynamics, demonstrated training pathologies (vanishing gradients, mode collapse, overfitting), and applied mitigations.

These experiments highlight the importance of architecture choices, pretraining objectives, and training strategies in shaping model performance and stability. Future work could extend to convolutional GANs, larger datasets, and self-supervised ViT pretraining for stronger conclusions.

Acknowledgements

Thanks to the course instructor - Dr. Tahir and teaching assistants Haseeb Tahir And Abdullah Bin Faisal for guidance.

Impact Statement

This work aims to strengthen conceptual understanding of deep learning models. Broader societal impact is limited to educational purposes.

References

[1] PyTorch ResNet Documentation. https://docs.pytorch.org/vision/stable/models/resnet.html

[2] PyTorch nn.Module.requires_grad Documentation. https://pytorch.org/docs/stable/generated/torch.nn.Module.html#torch.nn.Module.requires_grad