Practical Work: Out-of-Distribution Detection, OOD Scoring Methods, and Neural Collapse

Marie DIDIER, Damien LEGRAND

February 20, 2026

Abstract

This report details the training of a ResNet-18 classifier on CIFAR-100 and evaluates several post-hoc Out-of-Distribution (OOD) detection methods (MSP, Max Logit, Mahalanobis, Energy, and ViM). We also analyze the Neural Collapse (NC) phenomenon during the Terminal Phase of Training (TPT) and implement NECO, an OOD detection score inspired by this geometric behavior.

Contents

1 Introduction

1.1 Objective

The goal of this project is to study how a ResNet-18 classifier behaves at the end of training on CIFAR-100, and how its internal representations can be used for Out-of-Distribution (OOD) detection. We compare standard post-hoc OOD scoring methods and analyze the Neural Collapse phenomenon.

1.2 The Problem: Overconfidence

Deep neural networks often assign high probabilities to inputs it has never seen before and on the edge of the original distribution. This overconfidence is a major issue for deploying models in real-world scenarios, for example a system might not flag the need for manual override, even though it has never seen such data before. OOD detection addresses this by quantifying uncertainty, in order to flag or reject samples that don't belong to the training distribution.

1.3 Why Neural Collapse is Interesting

Neural Collapse shows that at the terminal phase of training, a network's latent representations reorganize into a highly structured, symmetrical geometry where samples of the same class perfectly cluster together. This predictable geometry can be highly useful for OOD detection:

- **A Strict Mathematical Baseline:** It provides a rigorous, low-dimensional framework (a Simplex) that describes exactly how known In-Distribution (ID) data should be represented.
- **Geometric Anomaly Detection:** Because the ID feature space is so organized, OOD samples naturally deviate from this structure, often falling orthogonal to the known class centers. We can exploit this with methods like NECO, which detect OOD data by measuring how poorly a test sample projects on this expected ID subspace.

2 Training ResNet-18 on CIFAR-100

2.1 Architecture modifications

The standard ResNet-18 was designed for ImageNet images (224×224). Applying it directly to CIFAR-100 (32×32) would reduce the feature map too much, so the model wouldn't learn the fine-grained local patterns necessary to distinguish between 100 different classes. Therefore, we modified the architecture as follows:

- **Input Resolution:** The first layer was adapted for a 32×32 input size instead of the original 224×224 .
- **First layers Modifications:** We replaced the 7×7 convolution (stride 2) with a 3×3 convolution (stride 1, padding 1) and removed the initial max pooling layer. These layers were originally needed to reduce computational load by downsampling

large images. For 32×32 inputs, this reduction is not needed, and removing it keeps the spatial resolution for the residual blocks.

- **Output Layer:** The final fully connected layer was modified to have an output size of 100 instead of 1000 to match the number of classes in CIFAR-100.

2.2 Training Dynamics

We trained the model for 200 epochs using the Adam optimizer (learning rate = 0.001) and a batch size of 64. We also used data augmentation(`RandomHorizontalFlip` and `RandomCrop`) because of the few examples per class (500).

As shown in the figures, the training loss converges near zero while test accuracy stabilizes around 70%. This ensures the model has reached the terminal phase of training, allowing for a good analysis of Neural Collapse.

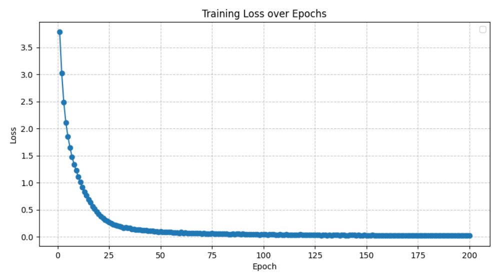

Figure 1: Training Loss

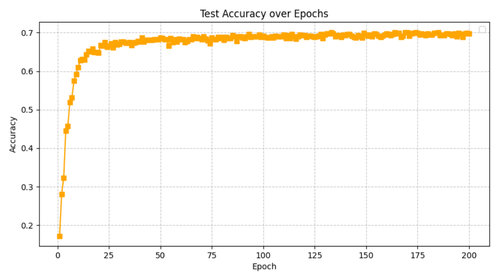

Figure 2: Test Accuracy

3 Out-of-Distribution (OOD) Detection

3.1 Context

OOD detection determines whether a test input belongs to the training data distribution (In-Distribution, ID) or a different one. Here, CIFAR-100 is the ID dataset, and SVHN is the OOD dataset.

3.2 Mathematical Definitions

We evaluate five post-hoc scores using the pre-trained classifier. Let x be the input, $f(x)$ the extracted features before the final layer, and $z(x)$ the output logits.

- **Max Softmax Probability (MSP):**

$$S_{MSP}(x) = \max_c \frac{\exp(z_c(x))}{\sum_{i=1}^{C} \exp(z_i(x))} \tag{1}$$

- **Maximum Logit Score:** Avoids softmax normalization to prevent overconfidence.

$$S_{MaxLogit}(x) = \max_c z_c(x) \tag{2}$$

- **Mahalanobis Distance:** Measures distance to the closest class-conditional mean μ_c using the empirical covariance matrix Σ.

$$S_{Mahalanobis}(x) = -\min_c \left((f(x)-\mu_c)^T \Sigma^{-1}(f(x)-\mu_c)\right) \tag{3}$$

- **Energy Score:** Maps logits to an energy scalar.

$$S_{Energy}(x) = \log \sum_{i=1}^{C} \exp(z_i(x)) \tag{4}$$

- **ViM (Virtual-logit Matching):** Combines logit scores with a feature-space projection penalty.

$$S_{ViM}(x) = \max_c z_c(x) - \alpha \|P^{\perp}(f(x)-\mu)\|_2 \tag{5}$$

3.3 Implementation Details

We evaluate the OOD scores using our pre-trained ResNet-18 model. The testing process is done in two steps:

1. **Fitting Parameters:** Some methods (Mahalanobis, ViM, and NECO) need reference statistics from the In-Distribution data. We extract the features from the CIFAR-100 training set to compute the class means, the covariance matrix, and the PCA components (using 64 dimensions).
2. **Scoring:** We then compute the OOD scores for the CIFAR-100 test set (our ID data) and the SVHN test set (our OOD data).

Finally, we compare the ID and OOD scores, generate the ROC curves, and calculate the Area Under the Curve (AUC) for each method.

3.4 Results and Analysis

Figure 8 shows the ROC curves. The AUC scores are: Mahalanobis (0.745), MSP (0.700), Max Logit (0.695), ViM (0.622), and Energy Score (0.564).

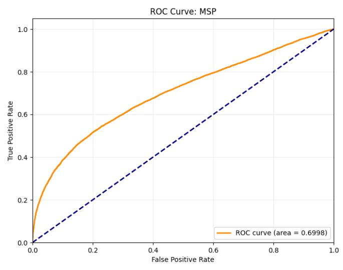

Figure 3: ROC: MSP

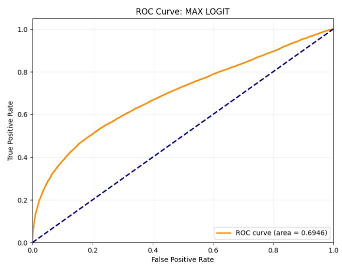

Figure 4: ROC: Max Logit

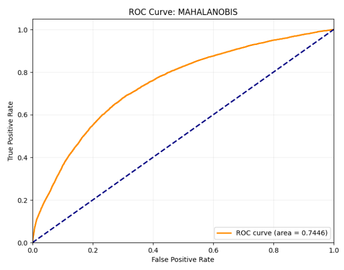

Figure 5: ROC: Mahalanobis

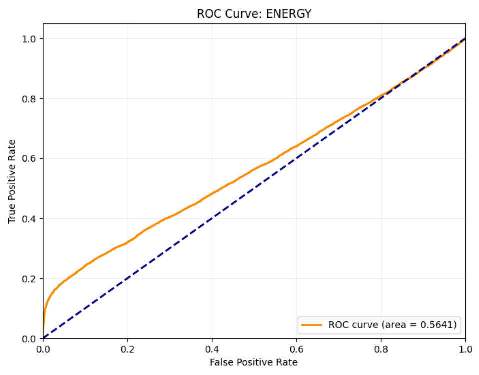

Figure 6: ROC: Energy Score

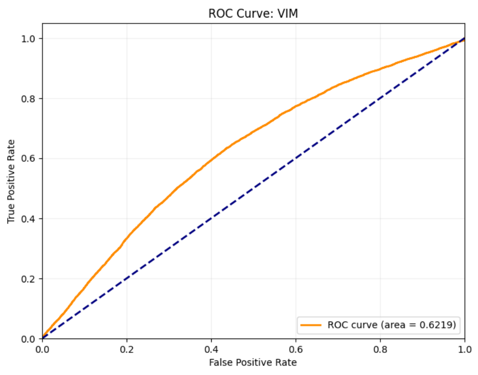

Figure 7: ROC: ViM

Figure 8: ROC Curves for OOD scoring methods (CIFAR-100 vs SVHN).

Mahalanobis performs best among these methods because it uses the distribution of intermediate features rather than just final the logits. MSP and Max Logit show acceptable results (0.70), indicating our model is not severely overconfident. ViM and Energy underperform here. This is probably because of the lack of temperature scaling for Energy, and because we didn't optimize VIM (we hardcoded PCA dimension (64 out of 512)).

4 Neural Collapse Analysis

Neural Collapse is described by four main properties during the Terminal Phase of Training :

- **NC1:** Variability collapse (features cluster perfectly around class means).
- **NC2:** Convergence to a Equiangular Tight Frame (ETF).
- **NC3:** Convergence to self-duality (classifier weights align with class means).
- **NC4:** Simplification to nearest-class center decision.

4.1 NC1: Variability collapse

NC1 says that within-class variance should drop to zero. However, Figure 9 shows it increasing after an initial drop.

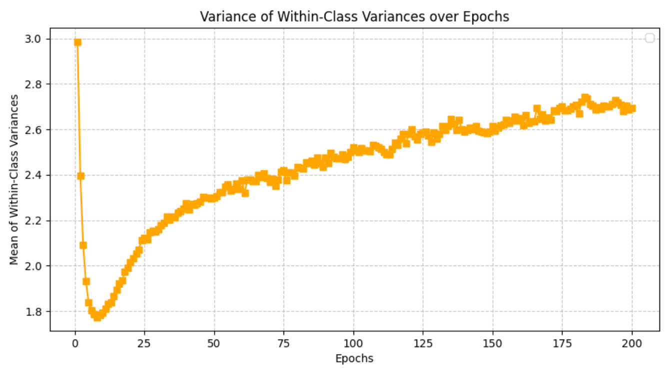

Figure 9: Within-class variance evolution.

This is a problem linked with out experimental setup:

1. **The Moving Target:** We originally extracted features inside the active training loop. Because the optimizer updates the weights at every batch, the model is constantly changing. This means we were computing the variance using features produced by slightly different versions of the network. This weight drift artificially increases the measured variance and hides the actual collapse.

2. **Constant learning rate:** Neural collapse usually requires strong Weight Decay and a decaying learning rate.

4.2 NC2 and NC3: Geometry and Alignment

Figures 10 and 11 evaluate NC2 (convergence to a Simplex ETF) and NC3 (self-duality).

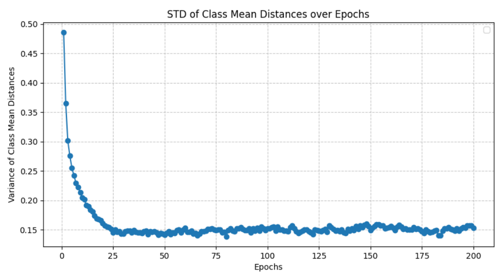

Figure 10: STD of Class Mean Distances.

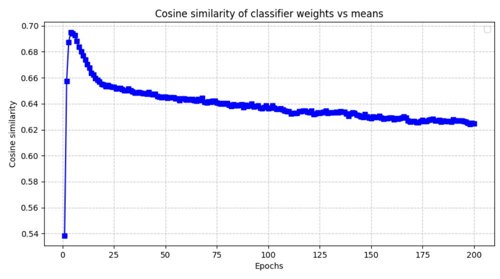

Figure 11: Cosine similarity: weights vs means.

For NC2, Figure 10 shows that the standard deviation of distances between class means drops rapidly and stabilizes at a low value (around 0.15).This confirms that the class centers are arranging themselves at roughly equal distances from each other, forming the highly symmetrical ETF structure predicted by the theory.

For NC3, Figure 11 tracks the cosine similarity between the classifier weights and the class means. While theory predicts a perfect convergence to 1.0 (self-duality), our model stabilizes around 0.62. Just like the NC1 variance, this imperfect alignment is an artifact of the moving target setup and the constant learning rate. The continuous weight updates prevent the network from completely freezing into perfect alignment. Nevertheless, the rapid initial spike and subsequent stabilization demonstrate the network's strong tendency to align its final layer weights with the learned class prototypes.

4.3 NC4: Simplification to nearest-class center decision

Neural Collapse theory states that at convergence, the linear classifier simplifies to a Nearest Center Classifier (NCC). Under perfect collapse, the standard linear decision boundary and the NCC would give identical predictions.

However, our model did not reach this perfect state. Because the within-class variance did not fully collapse (NC1) and the classifier weights did not perfectly align with the class means (NC3), the mathematical prerequisites for NC4 are unmet. The continuous weight updates from our moving target setup prevented the network from fully freezing, meaning we excpect a measurable discrepancy between the standard logits and an NCC.

5 NECO: Neural Collapse Inspired OOD Detection

5.1 Methodology

NECO is built to tackle NC5, which states that OOD features tend to be orthogonal to the ID features' ETF subspace.

We fit a PCA projection matrix P on the ID training features. For a test image x, we compute the ratio of its feature's norm within the principal subspace to its total norm:

$$S_{NECO}(x) = \frac{\|Ph_\omega(x)\|_2}{\|h_\omega(x)\|_2} \tag{6}$$

A score near 1.0 means the sample lies inside the learned ID subspace. A lower score indicates orthogonal energy, typical of OOD samples.

5.2 Results

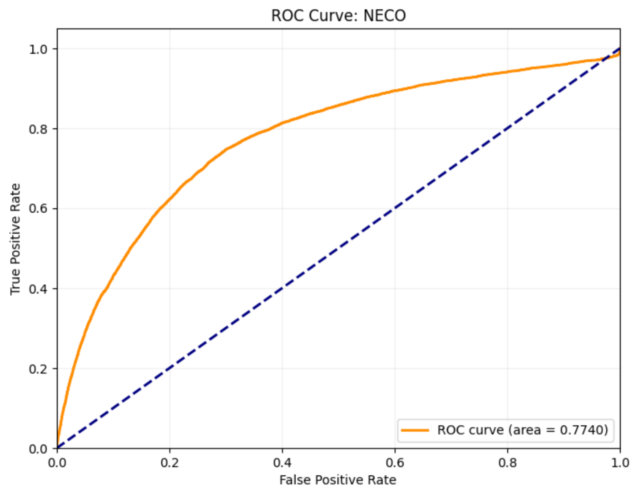

Figure 12: ROC Curve for NECO Score.

NECO clearly outperforms all other evaluated methods. The final ranking is:

1. **NECO:** 0.7740
2. **Mahalanobis:** 0.7446
3. **MSP:** 0.6998
4. **Max Logit:** 0.6946
5. **ViM:** 0.6219
6. **Energy Score:** 0.5641

Conclusion on NECO: Mahalanobis uses an inverted covariance matrix computed over all 512 dimensions, which captures unwanted noise. NECO solves this by relying strictly on the low-dimensional ETF subspace. It efficiently filters out orthogonal OOD noise, resulting in the best AUC score.

6 Conclusion

We trained a ResNet-18 on CIFAR-100 to compare OOD detection methods and analyze Neural Collapse.

Evaluating standard metrics showed that distance-based methods like Mahalanobis perform better than logit-based methods. Besides, observing the Terminal Phase of Training allowed us to observe the emergence of Neural Collapse properties. Although our specific training dynamics prevented a perfect theoretical collapse, the strong alignment between classifier weights and class means (NC3) and the symmetrical geometry of the classes (NC2) were clearly visible.

By exploiting the geometry of the ID subspace (NC5), the NECO method achieved the highest detection performance (AUC 0.7740). This project demonstrates that understanding a network's theoretical geometry at convergence is a highly effective way to build safer, more robust classifiers.