Understanding CNN Behaviour through Class Activation Manifolds

A Dissertation
Presented to the Faculty of Computer Science
of
Ashoka University
in partial fulfillment of the requirements for the Degree of
Postgraduate Diploma in Advanced Studies and Research (DipASR)

by
Niranjan Rajesh

Advisor: Debayan Gupta

May, 2024

Contents

Acknowledgements

I would like to acknowledge my deepest gratitude to my advisor, Professor Debayan Gupta for mentoring me since my first year in Ashoka. His guidance has been invaluable and my thesis would not be of this level without his input. I would also like to thank Professor Venkatakrishnan Ramaswamy for his support and his expertise over the last year.

I am also extremely grateful for my support system - my friends in and out of Ashoka as well as my loving family. The thesis would not have been possible without their presence in my life.

Finally, I would like to thank the Computer Science Department and Ashoka University in its entirety for equipping me with the skills and academic luxuries that has enabled me to achieve this feat. I hope this is the first of my many journeys of scientific inquiry!

Chapter 1

Introduction

Over the last couple of decades, the advent of Deep Learning has taken the fields of Computer Vision [1], Natural Language Processing [2] and more by storm. Convolutional Neural Networks (CNNs) [3] have been at the forefront of Computer Vision, leading the charge in recent attempts to solve the problem of Image Recognition. With the aide of improving levels of compute power due to hardware breakthroughs and the increased availability of large, annotated datasets [4], these networks have gained acclaim due to their near human-like performances in image classification tasks. The success in generic domains have allowed for their adoption in domains like social media [5] retail [6] while also being employed in critical systems of healthcare [7] and automotive industries [8] as well.

Despite the widespread prevalence and acclaim of the state-of-the-art Convolutional Neural Networks, they do have their shortcomings. One such problem is a lack of robustness to variations in images. This can be at the object-level where the model performs poorly when trying to classify data that is out of it's training data distribution [9]. Additionally, CNNs are also vulnerable to pixel-level perturbations [10, 11]. *Adversarial Attacks* to images involve applying slight perturbations to an image such that the image is perceived to be the same by humans but vastly different by the model. Vulnerability to such attacks make the decision to integrate such networks into critical systems, a more difficult one as potential lives could be at stake. There has been considerable amount of work aimed at addressing these problems by designing new attack algorithms and then algorithms to subsequently defend against such attacks [12, 13]. However, such defense algorithms involve further training on perturbed inputs on top of normal training which are less-than-ideal in two ways.

Firstly, such training measures substantially increase the data and compute requirements of an already-inefficient training regime. Secondly, measures like adversarial training will harm the model's accuracy on clean images, reducing its overall classification power. Such insights lead us to look for a more mechanistic understanding of neural networks and why they are learning visual representations different to the ones humans learn, making the models susceptible to perturbations while they remain unnoticeable to us.

On the quest to understand the inner workings of a neural network, we turn to theoretical neuroscience for assistance. As the fields of theoretical and computational neuroscience has evolved, assisted by techniques to record neuronal signals, an approach to understand higher-level population dynamics of a biological neural networks began to take form. Task-dependent neural manifolds represent the neural states involved in performing a certain task. A single point on the manifold represents the firing of all (recorded) neurons' activities which are represented by their corresponding coordinates [14]. Due to co-modulation in between neurons and activity that is dependent on each other, neural activity for a task cannot span the entire N-dimensional neural space (N = number of neurons recorded), but they span a lower-dimensional manifold within the larger neural space. Neuroscientists have gained insights about motor activity [15], skill learning [16] and visual recognition [17] by estimating intrinsic neural manifolds that underlie population activity. Furthermore, in visual neuroscience, 'object manifolds' estimated from neural data in the visual pathway correspond to the sight of a specific object across variations like rotation, occlusion, brightness, etc. The brain's role in object recognition is hypothesised as the visual cortex applying transformations to make the manifolds more linearly separable until it is classified by a linear decision function [18].

In this work, we generalise object manifolds to **Class Activation Manifolds** which contain CNN representations of objects across different instances of the same class. Each point on

the manifold represents the activations of neural units on the CNN for a particular image of an object that belongs to that class. We estimate the dimensionality of this intrinsic manifold, using activations from the penultimate layer of a CNN to investigate the behaviour of CNNs. In particular, we investigate how the dimension of Class Activation Manifolds vary with robustness of models. If a pattern emerges, insights on the representational causes for vulnerability to attacks may be obtained, allowing us to potentially develop more complete and effective solutions against these problematic behaviours of CNNs.

Chapter 2

Background & Previous Work

Convolutional Neural Networks (CNNs)

The first neural networks from the 1950s were multi-layer perceptrons (MLPs) [19] built with collections of computational units: 'neurons' in a hierarchical network that was directly inspired by the architecture of the brain. Later on in the 1980s, Convolutional Neural Networks (CNNs) [3, 20] were designed to process visual information, again with inspiration from the simple and complex cells discovered by Hubel and Wiesel [21].

A typical CNN architecture can be seen in fig 2.1. The characteristic components of a CNN are its **convolutional and pooling layers**. Convolutional layers extract features from an image. This is done when a filter (with trained weights, responsible of identifying certain features) is slid across the image, and a convolution operation is performed at each step to produce a feature map. These filters that make up a convolutional layer constitute the receptive field of a network. The filters are matrices of trainable weights that are 'learned' during the training stage as the model learns to identify important visual features for the task. These convolutional filters are spatially invariant feature detectors and are modelled after the simplex and complex cells discovered in the Hubel and Wiesel experiments. Once the feature maps are computed, they go through pooling layers to compress the features, typically by means of averaging. A typical CNN involves a sequence of convolution and pooling operations before the computed features are handed over to fully connected, classification layers that interpret the visual information processed and make the final decision of classification. After the inception of CNNs in the early 1990s, it has only been in the

last decade that CNNs have been adopted widely for object recognition tasks. This can be attributed to the hardware developments in Graphical Processing Units (GPUs), the availability of large-scale image datasets [22] and the architectural designs of modern Deep CNNs [23, 24, 25, 26, 27, 28].

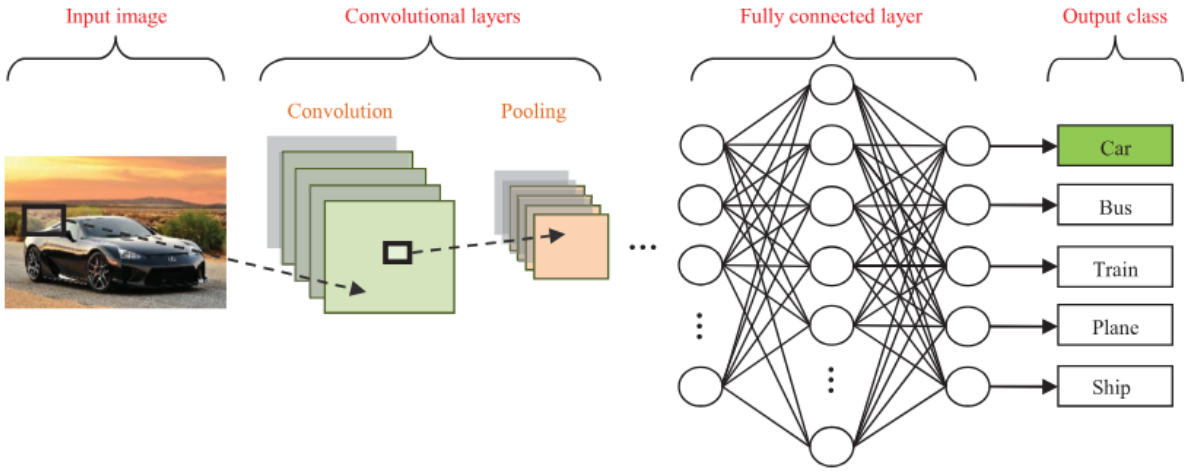

Figure 2.1: A typical CNN architecture used for image classification from [29]

2.1 Adversarial Robustness and CNNs

The study of deep learning robustness has been an active one over the last decade. The first report that state-of-the-art CNNs (of the time) were susceptible to adversarial attacks was made by Szegedy et al [10]. By adding imperceptible perturbations to an image, they were able to show that model predictions were negatively affected and it became more likely to misclassify such images. An example of a perturbed image is shown in fig 2.2. Generating an adversarial attack involves solving the following equation:

$$x^{adv} = \underset{\hat{x}:\|\hat{x}-x\|_p \leq \epsilon}{\operatorname{argmax}} L(\hat{x}, t)$$

To ensure that the model is imperceptible, we often impose a perturbation budget of ϵ such that the p-norm of the difference between the adversarial image and the original image must

be less than or equal to ϵ. L is a typical classification loss and the equation tries to find the $\hat{x}$ that maximises L while remaining under the perturbation budget. The solution to this equation will give us the adversarial image, x^{adv}. Many approaches to generate adversarial attacks have been proposed in recent years, including gradient-based attacks [11, 12, 30]. These attacks leverage the gradient of the models' loss function with respect to its input which means that generating the attacks need access to the entire model, leading to their name of 'white box attacks'.

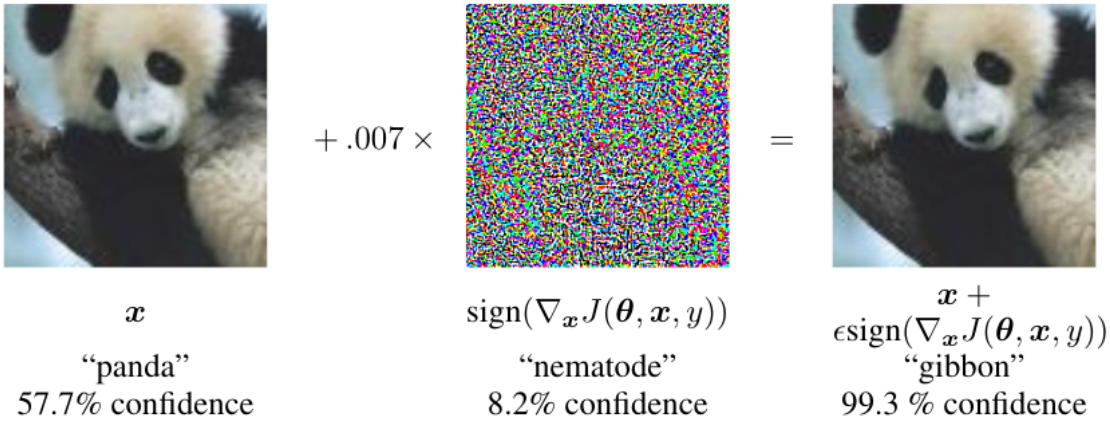

Figure 2.2: An example of an adversarial attack from [11]. Despite remaining imperceptible to humans, the perturbation causes CNNs to misclassify with confidence.

2.2 Neural Manifolds

The advancement of neural recording technology and with it, the ability to record multiple neurons in the brain simultaneously during a task has allowed for progress to be made to understand neural population dynamics and how they attribute to different aspects of biological intelligence. If N neurons in the brain are recorded, neural activity does not necessarily span the entire N-dimensional. This is because of underlying constraints of the neural network like the co-modulation of neurons limit the possible patterns of neural activity for a specific task to span a lower (than N) dimensional manifold within the larger neural space [15,16,31].

In visual neuroscience, the task of object recognition has also been viewed through the lens of neural manifolds. In this context, neural manifolds can also be referred to as object manifolds. Where the 'task of identifying a particular object' leads to neural activity that make up points on these object manifolds as seen in fig 2.3a. Visual variations of the same object lead to varying neural activity and thus, various points that make up the manifold. In the pixel space, however, these object manifolds are intertwined and a 'good neural space' (fig 2.3b) needs to be found such that a decision function hyperplane can distinguish between different object manifolds. DiCarlo et al [17, 18] proposes that this 'untangling' of object manifolds is the task carried out by the animal visual pathway before a decision function distinguishes between objects and a visual percept is formed in our mind.

We believe that a similar process is being carried out in CNNs and the dimensionality or 'smoothness' of the object manifolds may be responsible for the robustness (or lack thereof) in these models.

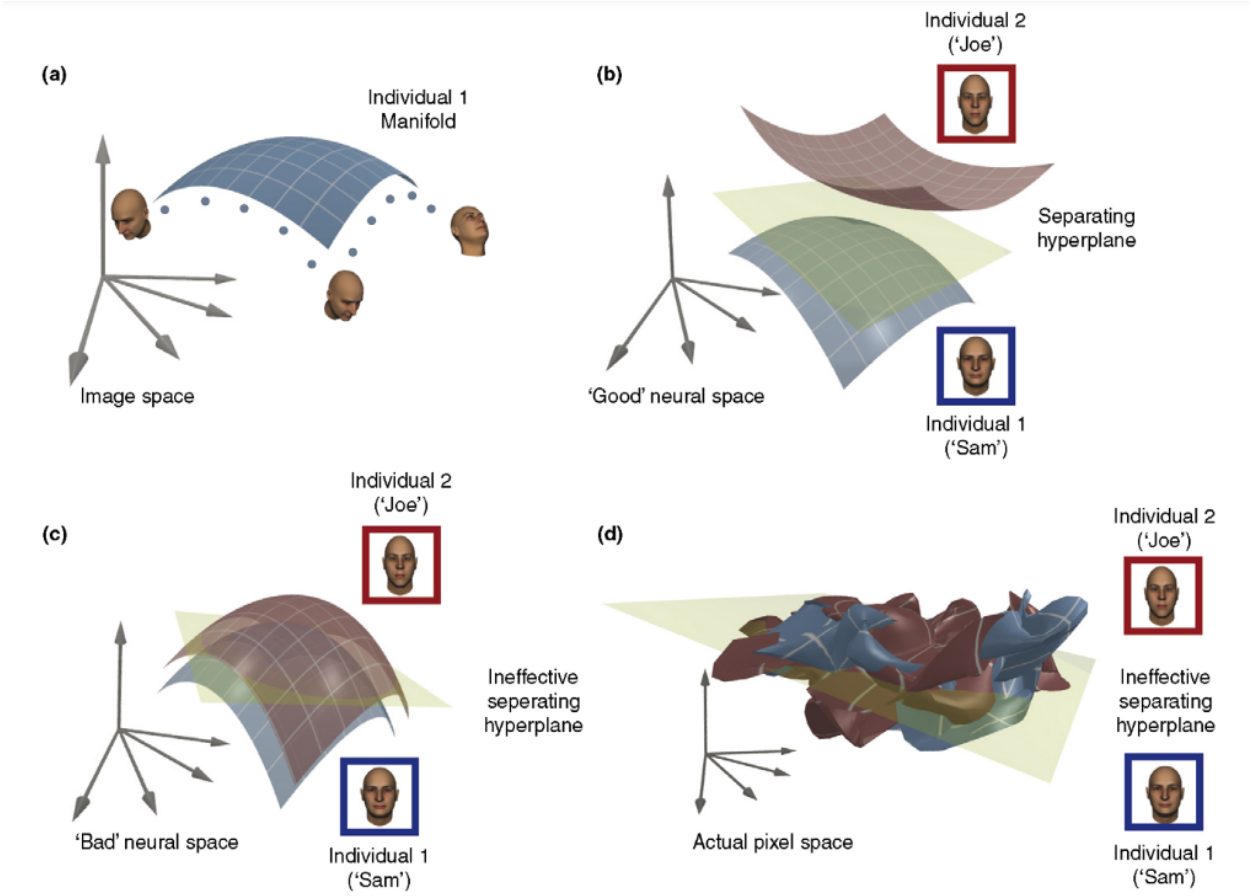

Figure 2.3: Figure from [18] depicting the untangling hypothesis. (a) shows the variation of an object being the points on the object manifold. (b) and (c) show good and bad neural spaces where a simple decision function is able to and unable to distinguish between two objects respectively. (d) shows the object manifold being heavily entangled. The untangling hypothesis suggests that when an image is captured at the eye, the signals from the retina is insufficient to distinguish between two objects. Transformations must be carried out to effectively untangle these manifolds such that a separating hyperplane decision function can classify them. A similar analogy can be made for images taken from a camera and their RGB values not being separable enough and a CNN untangling them over the course of its layers before a final decision function (classification layer) makes the decision of object recognition.

Chapter 3

Problem Statement

In this work, we aim to seek mechanistic insights about the problematic behaviour of CNNs, especially their susceptibility to adversarial attacks. We plan on achieving this by borrowing the tool of neural manifolds to obtain an understanding of the population dynamics at play during inference of perturbed images. Generalising from object manifold like in section 2.2, for CNNs, we utilise **Class Activation Manifolds (CAMs)**.

CAMs are obtained by extracting the activations of the final non-classification layer when images in the same class are input into the model. We treat each set of flattened activations for a single class as points on a point-cloud manifold. We then use Principal Component Analysis [32] to estimate the dimensionality of the point-cloud class activation manifold. We then draw insights on the behaviour of the model as we contrast the dimensions of CAMs and the class-wise adversarial robustness of that model. We follow this by collecting the same measurements for different CNNs and benchmarking the relationship between CAM dimensionality and robustness across CNNs.

Chapter 4

Methodology

4.1 Architecture(s) and Dataset

In this work, we primarily work with **ResNet50**. The ResNet family of networks introduced in [24] solved a problem that kept networks from growing deeper - the vanishing gradient problem [33]. This involves the degradation of gradients as backpropagations occurs continuously over multiple layers. ResNets solved this by introducing residual connections between layers allowing for improved gradient flow. The architecture can be viewed in fig 4.1. The initial input undergoes a primary convolution with some important postprocessing like batch-normalisation and a non-linearity before it is pooled and passed to the multiple convolution blocks. The convolution blocks have residual skip connections that send a layers output to a layer deeper in the network, preserving previous processed information.

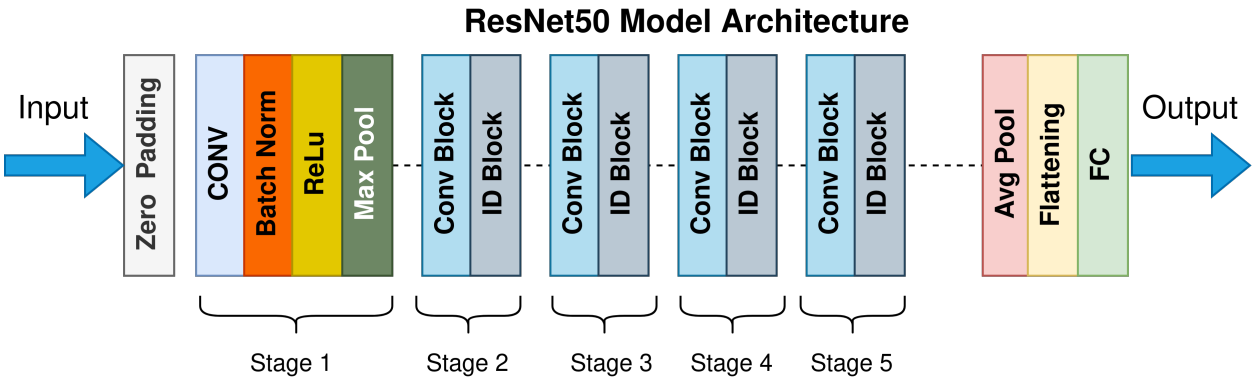

Figure 4.1: Architecture diagram of a ResNet50.

We choose to use ResNet50 as our base model primarily due to its popularity and its

prevalance in computer vision tasks. The ResNet family has remained in (or around) the state-of-the-art for image recognition since its inception in 2015. We also use other architectures like **ConvNeXt-B** (2022) [27], **(**2014) [28] , **VGG-16** (2014) [26] and **AlexNet** (2012) [23] in order to make comparisons with varying levels of robustness. All models were pre-trained on **ImageNet** [22] dataset and were used for inference in this work.

4.2 Adversarial Attacks

In this work, we will use **Projected Gradient Descent (PGD)** [12] with a perturbation budget of ϵ. PGD is a simple, yet powerful attack as it leverages multiple gradient steps steps to compute an attack. The intuition behind the computation of a PGD attack is to add perturbations to the original input where the perturbation at each step is the direction of the gradient of the loss function. The formal algorithm for PGD under a l_{infty} perturbation budget from [12] is shown in algorithm 1

Algorithm 1: Projected Gradient Descent (PGD) Adversarial Attack (l_∞)

Input: Original image x, Target class y, Loss function $J(\theta, x, y)$, Perturbation size ϵ, Step size α, Number of iterations K

Output: Adversarial example x_{adv}

Sample random noise n from Uniform distribution in range $(-\epsilon, \epsilon)$;

Initialize $x_{\text{adv}} = x + n$;

for $k = 1$ **to** K **do**

Compute the gradient of the loss function w.r.t. the input:

grad $\coloneqq \nabla_x J(\theta, x_{\text{adv}}, y)$;

Compute the step necessary for the adversarial attack:

step $\coloneqq$ sign(grad);

Compute the adversarial input:

step $\coloneqq x_{\text{adv}} + \alpha \cdot$ step

Clip the step to ensure it lies within $[x - \epsilon, x + \epsilon]$:

$x_{\text{adv}} \coloneqq$ clip(step, $x - \epsilon, x + \epsilon$);

end

4.3 CAM dimension estimation

In order to estimate the intrinsic dimensionality of a Class Activation Manifold, algorithm 2 was followed. The ImageNet [4] pretrained CNN were fed input images from a single ImageNet class and the activations of the penultimate layer was extracted. These activations were then used to estimate the dimensionality of the CAM using Principal Component Analysis [32]. As PCA is a powerful and simple dimensionality reduction algorithm that attempts to identify the principal components - orthogonal eigenvectors of the covariance matrix - that maximise the variance in the data. To estimate the dimensionality of the CAM, we treat the activations like co-ordinates on a point cloud manifold. These co-ordinates are then input into PCA which identifies principal components and their respective explained

variances. We can pick a number of principal components that explain a certain amount of variance (we chose 95% in this study). This number of components it takes to explain the threshold variance is our estimated dimensionality of the Class Activation Manifold. This process is repeated for as many classes of ImageNet as required.

Algorithm 2: Estimation of Class Activation Manifold Dimensionality

Input: Number of classes n, Number of images per class m, Threshold for variance explained γ=0.95

Output: List of dimensions for each class d_{classes}

Randomly sample n ImageNet classes;

for *each class* **do**

  Sample m images from the class;

  for *each image* **do**

    Extract activations of the final non-classification layer with D neurons;

    Record activations as D-dimensional list, a;

  end

  Concatenate recorded activations into a single $m \times D$ matrix A;

  Perform PCA on matrix A;

  Compute cumulative explained variance ratio;

  $d \coloneqq$ Number of principal components required to explain 95% of variance;

end

Concatenate all d's into a list d_{classes}

Code Availability

Code required to reproduce the results can be found on our **GitHub repository** .

Chapter 5

Experimentation and Results

5.1 Dimensionality of a CAM in ResNet50

Performing algorithm 2 allows us to find the the dimension of the CAM for multiple classes. Using the algorithm on a few classes, we get the results shown in 5.1. This experiment was run using ResNet50 and four classes sampled from ImageNet.

If we take the threshold for explained variance to be 0.95, we get dimensions in the range of

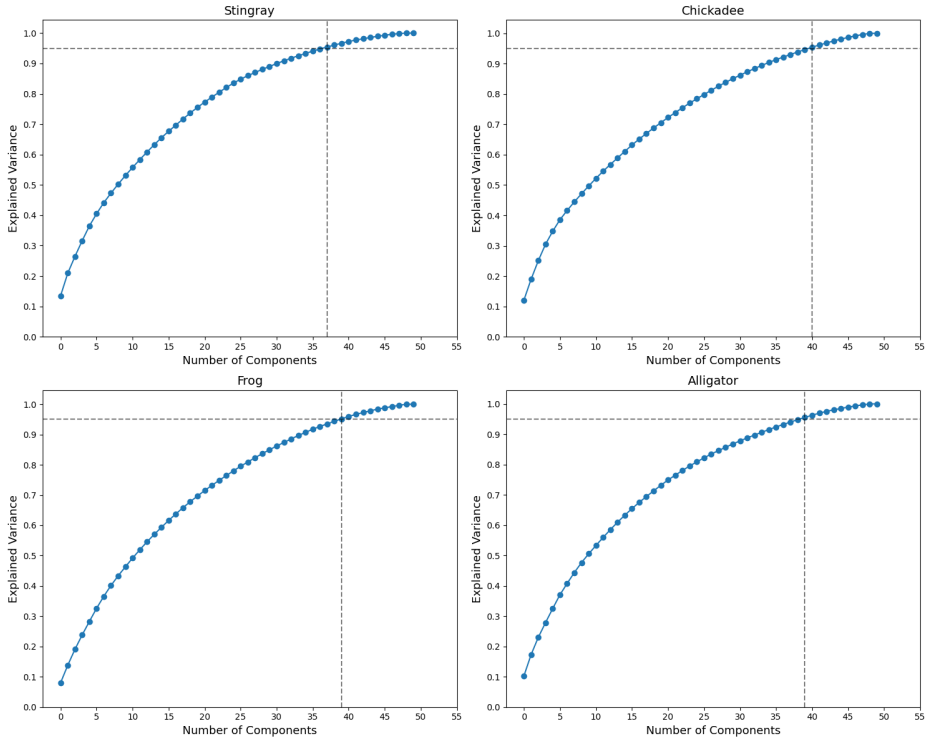

Figure 5.1: The explained variance of principal components identified by PCA for 4 different classes' activations in the ResNet layer

35 and 40. The average pool layer (layer right before classification) from which the activations

are being extracted output a vector of 2048 dimensions. The estimated dimensionality of this point cloud manifold being around 35-40 suggests that the intrinsic class manifold occupies a substantially smaller subspace within the larger neural space of 2048 dimensions.

5.2 CAM dimensionality and Adversarial Robustness in ResNet50

Next, we generated adversarially-perturbed images for 100 randomly sampled classes. The estimated dimensions of the CAMs for these classes in ResNet50 was also computed in order to investigate whether there was a direct relationship between dimensionality and class-wise adversarial accuracy. The perturbations were generated for the ImageNet classes using PGD described in algorithm 1 with $\epsilon = 0.005$. A frozen ResNet50 made predictions on these perturbed images and its performance was evaluated by computing an adversarial accuracy on each class.

Figures 5.2 and 5.3 show the relationship between class-wise adversarial accuracy and the Class Activation Manifold dimension. 5.2 shows a scatter plot of 100 classes' and their respective adversarial accuracy as well as CAM dimension. This shows a generally decreasing trend but due to slight variations in adversarial accuracy in between points and large differences in CAM dimension, the plot is quite noisy. For this reason, the classes' accuracies were binned to the nearest 5% and if there are multiple classes in the same bin, their CAM dimensions were averaged. **This plot shows a strong negative correlation (Pearson's: -0.939,** $p << 0.05$**) between the class-wise adversarial accuracy and CAM dimensions of those classes.**

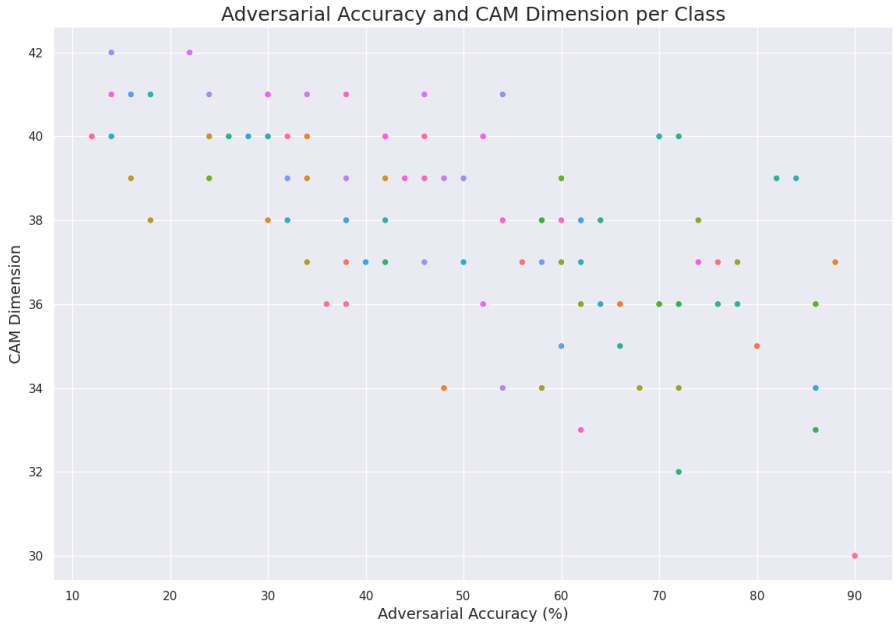

Figure 5.2: The relationship between class-wise accuracy on adversarial images and their respective CAM dimensions. Each coloured dot represents one of the 100 classes sampled from the 1000 ImageNet classes.

5.3 Benchmarking Relationships with more CNNs

To investigate whether the negative relationship between CAM dimensions and Adversarial Accuracy remains consistent across CNN architectures, we ran the same experiments with AlexNet [23], VGG16 [26] and ConvNeXt Base [27] models. As 100 class-wise adversarial accuracies and CAM dimension values would be hard to visualise across 4 models, we average both variables. The results can be seen in fig 5.4. A clear inverse relationship can be identified between the average adversarial accuracy and the average CAM dimensions. ResNet50 and ConvNeXt perform well on the adversarial data while also exhibiting CAMs of lower dimensions. On the contrary, VGG16 and AlexNet have lower adversarial accuracy scores and larger CAM dimensions. This strengthens the hypothesised relationship between CAM dimensions and robustness of models.

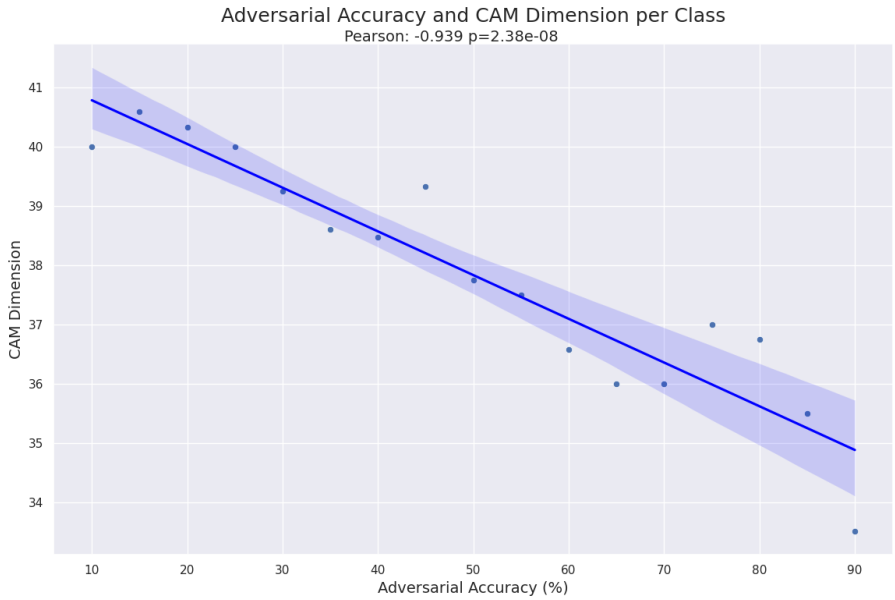

Figure 5.3: A binned version of fig 5.2 with a linear regression fit and the Pearson Correlation Score reported alongside a p value to show statistical significance of the relationship.

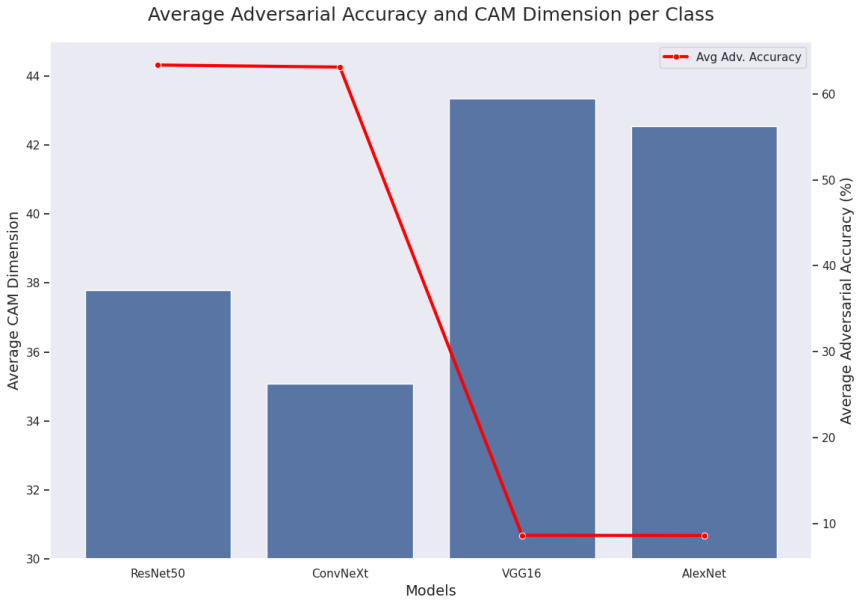

Figure 5.4: The relationship between average adversarial accuracy and average CAM dimensionality across CNNs. The bars represent CAM dimensions and the red line represents the adversarial performance of each model.

Chapter 6

Conclusion

In this study, we utilise an analogue of neural object manifolds in CNNs - Class Activation Manifolds (CAMs) to draw insights about the behaviour of CNNs. We map the activations of the final pre-classification layer of CNNs onto a point-cloud manifold and estimate the manifolds' dimensionality by using PCA and thresholding the explained variance. Using the methodology summarised above, we make three observations:

- The average dimensions of a Class Activation Manifold is significantly lower than the output shape of the activations layer. This suggests that the 'neural activity' measured while identifying objects in a single class can be mapped to a lower dimensional manifold.

- There is a strong negative correlation between a CNN's (ResNet50) class-wise adversarial robustness and the respecitive CAM dimensions. This could be because 'smoother' (lower dimensional) manifolds may be more robust as they are less likely to cause unexpected behaviour near the high-dimensional decision boundaries.

- This negative relationship maintains across a few other State-of-the-Art CNNs. This suggests that this could be a property of neural networks despite slight variations in architecture.

Future Work

Firstly, a lot more work needs to be done to verify this relationship further and strengthen this claim. This can be done through averaging over multiple runs of sampling different

subsets of images. Further, more CNNs and datasets can also be used to verify.

Additionally, we would also like to verify whether such patterns exist with robustness to out-of-distribution data. As the lack of robustness has been due to the inability to generalise near the decision boundaries.

We believe that this line of work holds much promise as attempts to mechanistically interpret the workings of a CNN has not been carried out rigorously in previous literature. If such properties can explain certain behaviour of CNNs, procedures to align the manifolds of a neural network in a certain way to make it more robust might be possible and a more effective solution than current measures to tackle this problem.

Bibliography

[1] Athanasios Voulodimos, Nikolaos Doulamis, Anastasios Doulamis, and Eftychios Protopapadakis. Deep Learning for Computer Vision: A Brief Review. *Computational Intelligence and Neuroscience*, 2018:1–13, 2018.

[2] Daniel W. Otter, Julian R. Medina, and Jugal K. Kalita. A Survey of the Usages of Deep Learning for Natural Language Processing. *IEEE Transactions on Neural Networks and Learning Systems*, 32(2):604–624, February 2021. Conference Name: IEEE Transactions on Neural Networks and Learning Systems.

[3] Backpropagation Applied to Handwritten Zip Code Recognition | MIT Press Journals & Magazine | IEEE Xplore.

[4] ImageNet: A large-scale hierarchical image database | IEEE Conference Publication | IEEE Xplore.

[5] Nanlir Sallau Mullah and Wan Mohd Nazmee Wan Zainon. Advances in Machine Learning Algorithms for Hate Speech Detection in Social Media: A Review. *IEEE Access*, 9:88364–88376, 2021. Conference Name: IEEE Access.

[6] Yuchen Wei, Son Tran, Shuxiang Xu, Byeong Kang, and Matthew Springer. Deep Learning for Retail Product Recognition: Challenges and Techniques. *Computational Intelligence and Neuroscience*, 2020:1–23, November 2020.

[7] D. R. Sarvamangala and Raghavendra V. Kulkarni. Convolutional neural networks in medical image understanding: a survey. *Evolutionary Intelligence*, 15(1):1–22, March 2022.

[8] [1906.08834] Deep Learning in the Automotive Industry: Recent Advances and Application Examples.

[9] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The Many Faces of Robustness: A Critical Analysis of Out-of-Distribution Generalization, July 2021. arXiv:2006.16241 [cs, stat].

[10] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks, February 2014. arXiv:1312.6199 [cs].

[11] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and Harnessing Adversarial Examples, March 2015. arXiv:1412.6572 [cs, stat].

[12] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards Deep Learning Models Resistant to Adversarial Attacks, September 2019. arXiv:1706.06083 [cs, stat].

[13] Eric Wong and J. Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope, June 2018. arXiv:1711.00851 [cs, math].

[14] Chad Giusti, Eva Pastalkova, Carina Curto, and Vladimir Itskov. Clique topology reveals intrinsic geometric structure in neural correlations. *Proceedings of the National Academy of Sciences*, 112(44):13455–13460, November 2015. Publisher: Proceedings of the National Academy of Sciences.

[15] Juan A. Gallego, Matthew G. Perich, Lee E. Miller, and Sara A. Solla. Neural Manifolds for the Control of Movement. *Neuron*, 94(5):978–984, June 2017.

[16] Patrick T. Sadtler, Kristin M. Quick, Matthew D. Golub, Steven M. Chase, Stephen I. Ryu, Elizabeth C. Tyler-Kabara, Byron M. Yu, and Aaron P. Batista. Neural constraints on learning. *Nature*, 512(7515):423–426, August 2014.

[17] James J. DiCarlo, Davide Zoccolan, and Nicole C. Rust. How does the brain solve visual object recognition? *Neuron*, 73(3):415–434, February 2012.

[18] James J. DiCarlo and David D. Cox. Untangling invariant object recognition. *Trends in Cognitive Sciences*, 11(8):333–341, August 2007.

[19] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

[20] Kunihiko Fukushima. Neocognitron: A hierarchical neural network capable of visual pattern recognition. *Neural networks*, 1(2):119–130, 1988.

[21] David H Hubel and Torsten N Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, 160(1):106, 1962.

[22] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[23] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[25] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.

[26] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[27] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A ConvNet for the 2020s, March 2022. arXiv:2201.03545 [cs].

[28] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely Connected Convolutional Networks, January 2018. arXiv:1608.06993 [cs].

[29] Waseem Rawat and Zenghui Wang. Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review. *Neural Computation*, 29(9):2352–2449, September 2017. Conference Name: Neural Computation.

[30] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *Proceedings of the 37th International Conference on Machine Learning*, pages 2206–2216. PMLR, November 2020. ISSN: 2640-3498.

[31] Michael Okun, Nicholas A. Steinmetz, Lee Cossell, M. Florencia Iacaruso, Ho Ko, Péter Barthó, Tirin Moore, Sonja B. Hofer, Thomas D. Mrsic-Flogel, Matteo Carandini, and Kenneth D. Harris. Diverse coupling of neurons to populations in sensory cortex. *Nature*, 521(7553):511–515, May 2015. Publisher: Nature Publishing Group.

[32] Andrzej Maćkiewicz and Waldemar Ratajczak. Principal components analysis (PCA). *Computers & Geosciences*, 19(3):303–342, March 1993.

[33] The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions | International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems.