Regularization and Sparsity for Adversarial Robustness and Stable Attribution

Daniel Schwartz, Yigit Alparslan, and Edward Kim

Drexel University, Philadelphia PA 19104, USA
{des338,ya332,ek826}@drexel.edu

Abstract. In recent years, deep neural networks (DNNs) have had great success in machine learning and pattern recognition. It has been shown that these networks can match or exceed human-level performance in difficult image recognition tasks. However, recent research has raised a number of critical questions about the robustness and stability of these deep learning architectures. Specifically, it has been shown that they are prone to adversarial attacks, i.e. perturbations added to input images to fool the classifier, and furthermore, trained models can be highly unstable to hyperparameter changes. In this work, we craft a series of experiments with multiple deep learning architectures, varying adversarial attacks, and different class attribution methods on the CIFAR-10 dataset in order to study the effect of sparse regularization to the robustness (accuracy and stability), in deep neural networks. Our results both qualitatively show and empirically quantify the amount of protection and stability sparse representations lend to machine learning robustness in the context of adversarial examples and class attribution.

Keywords: Robust Machine Learning, Regularization, Sparsity, Attribution, Artificial Intelligence Safety, Adversarial Attacks, Image Perturbation, Black-box Approach

1 Introduction

In recent decades, advances in deep neural networks (DNN) have allowed computers to achieve or even exceed human-level performance on difficult image recognition tasks. DNNs are widely used today in several critical fields, such as bio-authentication systems, facial recognition, autonomous vehicles, malware detection, and spam filtering. These DNNs, and other machine learning models, typically maximize or minimize some objective function while enforcing some regularization in the training process.

Regularization in machine learning (ML) offers many benefits when optimizing an algorithm. Regularization induces sparsity on the activations and parameters of the system, improving generalizability and interpretability [7]. Mathematically, the form of regularization we investigate constrains the coefficients of the system, driving the estimates towards zero. This technique is known to discourage the learning of complex models, reduce the flexibility of the model, increase sparsity, and avoid overfitting to the training data.

However, in the context of robust machine learning, the impact of regularization has not been thoroughly explored. We hypothesize that regularization increases robustness in non-traditional ways. If we define robust ML as the ability of an algorithm to be consistent across training and testing, the overfitting properties of regularization are important. If we further define robust ML as the ability of the algorithm to maintain a stable performance after the addition of noise to the dataset, the generalizability of regularization will help. In this work, we explore robustness with respect towards two major unsolved research questions in deep learning, i.e. robustness of deep learning to adversarial examples, and to interpretation via classification attribution. Specifically, we ask the question, does sparse regularization improve the robustness against different adversarial examples? Does the introduction of sparse regularization maintain deep neural network models' attribution consistent across parameterizations?

2 Background

We will mathematically define a regularization term (or regularizer) $R(f)$ as the following term added to a loss function,

$$\min_f \sum_{i=1}^{N} V(f(x_i), y_i) + \lambda R(f) \tag{1}$$

Where V is a loss function that quantifies the cost of predicting $f(x)$ when the label or ground truth is y, and where N is the size of the training set and i refers to a single sample. The λ term is a hyperparameter that controls the weight of the regularizer. A more flexible model would be allowed to increase the magnitude of its coefficients, while a more constrained model would have a larger value of λ and thus have smaller valued coefficients.

If we define $f(x) = x \cdot w$, i.e. the approximation of y as characterized by an unknown vector of parameters (weights), w, we can then define $R(f)$ as $||w||_2^2$ for the case of L2 regularization e.g. Ridge regression, $||w||_1$ in the case of L1 regularization e.g. Lasso, and $(\alpha||w||_1 + (1-\alpha)||w||_2^2), \alpha \in [0,1]$ for Elastic Net. The L2 penalizes large values of w, while the L1 norm drives some of the coefficients exactly to zero, enforcing sparsity.

2.1 Related Work in Adversarial Attacks

Although state-of-the-art deep neural networks have achieved high recognition for various image classification tasks, the architectures used for these tasks have been shown to be unstable to small, well-sought, perturbations of images. Szegedy et al. [16] showed that adversarial examples on ImageNet were so minute and fine-grained that they were indistinguishable to the human eye and could generalize across many different architectures on different folds of the dataset. Thus, the architectures can be seen more as "memorizing" a mapping from the

images to a text classification as opposed to understanding the underlying meaning and generalizing concepts across different images. Furthermore, deep learning classification has a tendency to learn surface regularities in the data, and not truly learn the abstract concepts of classes and objects [8].

Current attacks that have been studied in the field, such as those of [2] and [12] have been studied as proof-of-concepts, where adversarial attackers are assumed to have full knowledge of the classifier (e.g. model, architecture, model weights, parameters, training and testing datasets). The strongest attack in the literature at the time of writing this article is Carlini's attack based on the L2 norm, and it is a white-box attack requiring full knowledge of the model. Much of this research has been interested in developing the most effective attacks possible, to be used as standards against which to test the robustness of image classifier DNNs. With less knowledge of the classifier model, the effectiveness of the attack decreases.

2.2 Related Work in Image Attribution

Image attribution is the concept of determining what parts of the image contribute to the classification, and how important are these parts of the image to the end result. Most attribution methods work by either perturbing the input signal in some way and observing the change in the output, or by backtracking the influence of the input via a modification of backpropagation. The use of backpropogation only requires a single forward and backwards pass through the model, and are thus efficient to compute [3]. The perturbation-based methods do not require access to the model, and thus can be leveraged on black-box models.

We can visualize the attribution which provide insight into the classifier decision. These visualizations have been used to characterize which parts of an input are most responsible for the output. This lends some interpretability to the model, and can be used to explain the prediction result. We propose the robustness of the attribution methods can be measured by quantifying the change in attribution when choosing different hyperparameters for the model, or altering the type and scale of perturbation to the input image [1].

3 Methodology

In this section, we describe the different deep learning architectures, the adversarial attacks, and attribution techniques used to evaluate our hypothesis on regularization and sparsity on robust machine learning.

3.1 Image Classification Architectures

ResNet [5] -It is becoming more popular and common in the machine learning community to increase the depth of deep learning architectures to improve accuracy and generalizability. However, as the network increases in depth, the

performance may begin to drop due to the vanishing gradient problem. Moreover, accuracy gets saturated and degrades rapidly resulting in the problem of degradation when the depth of a network increases. ResNet introduces a solution by construction to the deeper model and adding layers of identity shortcuts. The concept of an identity shortcut connection is that it can skip one or more layers performing identity mappings. They show it is easier to optimize the residual mapping than to optimize the original, unreferenced mapping. The intuition for this architecture is that the deep residual learning framework of a few stacked layers fitting an underlying mapping can be explicitly fit by a residual mapping.

MobileNet [6] - There have been various computer vision models proposed with a convolutional neural network (CNN) architecture for the task of image classification. In the field of utilizing computer vision models in mobile and embedded vision applications results in either the compression of pre-trained networks or to train small networks. On the other hand, MobileNet is Google's "mobile-first" computer vision model for TensorFlow that maximizes accuracy while utilizing limited resources. One of the novel contributions in MobileNet is leveraging a Depthwise Separable Convolution within its architecture, a depthwise convolution followed by a pointwise convolution where a depthwise convolution is the channel-wise $D_k \times D_k$ spatial convolution and a pointwise convolution is the 1×1 convolution to change the dimension. This unique convolution reduces the amount of operations to be computed significantly while only losing 1% in accuracy.

VGG16 [15] - VGG-16 is a deep convolutional neural network that was adapted from AlexNet [9] in which it replaces the large kernels with multiple, smaller 3×3 kernel filters. The introduction of this model showed the advantages of adding depth complexity to convolutional neural networks in order to improve its accuracy and resulted in significant improvements to prior deep learning models. VGG is broken up into 5 groups, each with a convolutional layer followed by a max-pooling layer, with the last part of the architecture consisting of fully-connected layers. AlexNet has been found to capture more unrelated background information in its final convolutional layer that confuses the prediction, whereas VGG-16 helps cover the full receptive field with larger feature maps and thus, outperforms AlexNet.

3.2 Adversarial Attacks

Deep Fool [11] - Deep Fool is an untargeted white box attack that misclassifies an image with the minimal amount of perturbation possible. For intuition, in a binary classification problem, there exists a hyperplane separating two classes; DeepFool takes an input x and projects it onto the hyperplane while pushing it a little beyond, misclassifying it. As a result, in the multi-class extension of the problem of image classification, DeepFool projects the input, x, to the closest hyperplane and misclassifies it. Equation 2 represents the function to compute

the closest hyperplane given an input x_0 where f are class labels and w are the gradients.

$$\hat{l}\left(\mathbf{x}_0\right) = \underset{k \neq \hat{k}(\mathbf{x}_0)}{\arg\min} \frac{\left|f_k\left(\mathbf{x}_0\right) - f_{\hat{k}(\mathbf{x}_0)}\left(\mathbf{x}_0\right)\right|}{\left\|\mathbf{w}_k - \mathbf{w}_{\hat{k}(\mathbf{x}_0)}\right\|_2} \tag{2}$$

Fast Gradient Sign Method (FGSM) [4] - Most image classification architectures are neural networks, which learn by updating weights via a backpropagation algorithm that computes gradients. Fast Gradient Sign Method uses the gradients of the neural network to generate an adversarial example by using the gradients of the loss with respect to the input image x to create a new image which maximizes this loss. Furthermore, the input image x is manipulated by adding or subtracting a small error ϵ to each pixel depending upon the sign of the gradient for a pixel. Equation 3 represents the simple, cheap cost function to obtain the optimal max-norm constrained perturbation of an input image x, with parameters of the model θ, the cost to train the model $J(\theta, \mathbf{x}, y)$ and a small multiplier ϵ to guarantee small perturbations. The addition of errors in the direction of the gradient results in misclassification.

$$\eta = \epsilon \text{sign}\left(\nabla_{\mathbf{x}} J\left(\theta, \mathbf{x}, y\right)\right) \tag{3}$$

Projected Gradient Descent (PGD) [10] - This targeted white box attack is an extension of FGSM and is often referred to as Iterative-Fast Gradient Sign Method (I-FGSM), where FGSM is applied to an image iteratively. Since this is a targeted class, the objective is not to simply just misclassify the image but to classify the image to a specific desired class. In FGSM, the loss is calculated with respect to the true class and added the gradients computed with respect to the true class increasing loss for the true class and misclassifying it. However, in this case, the loss is calculated with respect to the target class to minimize the loss for the target class moving in the direction of the target class. This process can be described as projecting onto a l_p ball with a defined radius and clipping the values of the adversarial sample so that it lies within the set data range. This multi-step variant of FGSM is shown in Equation 4 where $\mathcal{S}$ is :

$$x^{t+1} = \prod_{x+\mathcal{S}} \left(x^t + \alpha \text{sign}\left(\nabla_x L\left(\theta, x, y\right)\right)\right) \epsilon \tag{4}$$

3.3 Attribution Methods

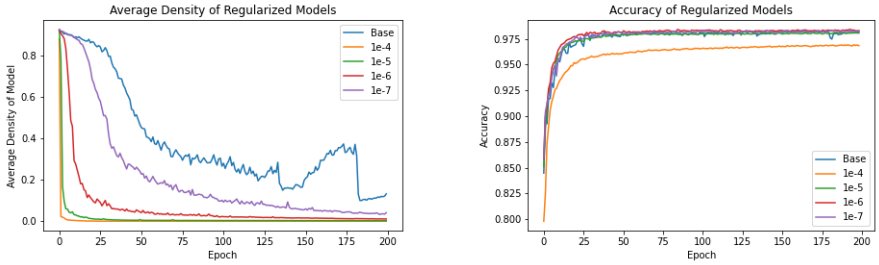
Sliding Patch Method [17] - In this input perturbation attribution method, one can systematically occlude different portions of the input and monitor the output of the classifier. We slide an occlusion patch of different sizes across the input image. By investigating the changes that occur in prediction, one can create a heat attribution map. However, given different size occlusion patches, a robust machine learning method should generate a consistent attribution map. Our experiments look at the consistency of maps as a function of patch size.

Grad-CAM [14] - Gradient-weighted Class Activation Mapping (Grad-CAM) is a backtracking method that uses the gradients of a target in the final convolutional layer to produce a coarse localization map highlighting important regions in the image for the final prediction. Grad-CAM is able to localize class-discriminative regions while being orders of magnitude cheaper to compute than occlusion methods. These types of attribution visualizations are critical to provide interpretability to a model and build trust with the end user. For machine learning robustness, we postulate that slight transformations to the input image should not significantly alter the attribution maps.

4 Experiments and Results

For our experiments, we first validate that sparse regularization is able to maintain a high level of accuracy over a range of sparse penalty parameterizations. We then experiment with adversarial examples and empirically validate the effects of adding regularization to the attacked model. Lastly, we validate that sparse regularization has a consistency effect when exploring parameterizations and input perturbations for class attribution.

4.1 Sparse Regularization Effect on Average Density



(a) Average Density of Neurons in CNN

(b) Training Accuracy of CNN

Fig. 1: A comparison of the average density of a single layer CNN and its regularized variants. This experiment shows the intuition that sparse CNNs (models with low average density) perform just as well if not better as dense CNNs.

While we are researching different regularization methods, we note that there is a clear connection between regularization and sparsity in the weights of the model and output of representation. Sparsity is induced by forms of regularization (L1, Elastic Net, Dropout, etc.), which provide us with the many benefits of regularized models. A common metric to measure sparsity is the Average Activity Ratio (AVR) or the average density of neurons activated per stimulus. Representationally the activation of a neuron denotes the use of an additional

dimension used to encode the data. Thus, the least amount of neurons activated per projection into the output space, the sparser the representation of the data.

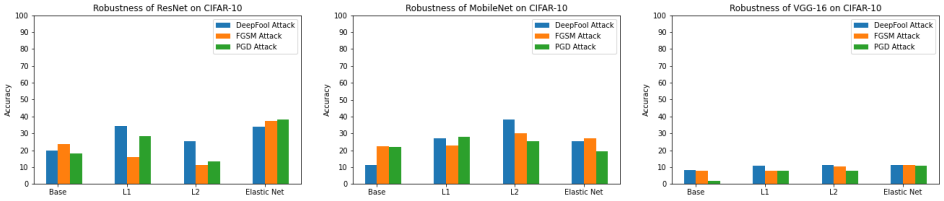
Our first experiment is to empirically show that sparse regularization performs on par with a non-regularized counterpart. The distinguishing feature of sparse coding compared to local or dense code is that its activity ratio lies in the range of $[0, 0.5]$. For intuition, we train a single layer CNN and impose a sparsity constraint on a cross entropy loss in Equation 5,

$$L = -\frac{1}{m}\sum_{i=1}^{m}(y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i)) + \lambda \sum_{j} |a_j| \tag{5}$$

Where m indicates the number of examples and i indicates the specific example being trained. Additionally, y_i refers to the ground truth target and $\hat{y}_i$ is the predicted output from the single layer CNN. Lastly, the last term of the loss function represents the regularization on the network, where $|a_{i,j}|$ is the absolute value of the activation for the hidden neuron j in the CNN layer and for each example the sum of activations for all hidden neurons is regularized by λ, a hyperparameter that affects the sparsity constraint. The closer λ is to 1, the more sparsity encouraged and the closer λ is to 0, the less sparsity is encouraged. We set λ to 0 for the base model and scale λ from $1e-4$ (0.0001) down to $1e-7$ (0.0000001) to examine the different effects sparsity has on training accuracy.

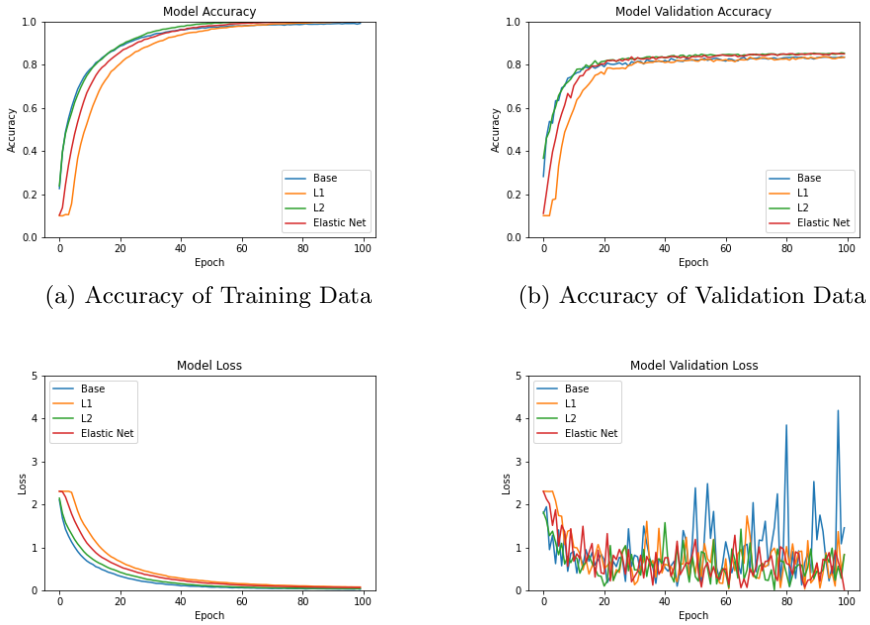
In Figure 1, we achieve sparse code without losing much in terms of accuracy. There are $39,200$ trainable neurons in the network and the graph of the Average Density denotes the decimal equivalent of the quotient of number of neurons activated divided by the total number of trainable neurons. It is clear that the regularization encourages the Average Density to decrease, but the accuracy of the less dense models are just as accurate, if not better.

4.2 Robustness Against Adversarial Attacks



(a) Attacks on ResNet (b) Attacks on MobileNet (c) Attacks on VGG-16

Fig. 2: A comparison of the accuracy achieved on different architectures when attacked by an adversarial algorithm (DeepFool, FGSM, and PGD). In nearly all cases, regularization helps improve the robustness of the model. Elastic net regularization is most consistent with robustness against adversarial attack.



(a) Accuracy of Training Data

(b) Accuracy of Validation Data

(c) Loss Curve of Training Data

(d) Loss Curve of Validation Data

Fig. 3: VGG-16 training and validation accuracy and loss on CIFAR-10. In (d) the base model validation loss is beginning to show signs of overfitting as the loss starts to demonstrate high variance.

Next, we run experimentation on how regularization can improve robustness against adversarial attacks on the architectures: ResNet, MobileNet, and VGG-16. We train each architecture: ResNet-50, MobileNet, and VGG-16 on the CIFAR-10 dataset, then generate adversarial images using the various attacks: DeepFool, Fast Gradient Sign Method (FGSM), and Projected Gradient Descent from [13]. Next, we retrain the networks on the training set augmented with the adversarial images i.e. adversarial training, and evaluate the classifier on the test set and record the accuracy of the network on the adversarial images. To regularize each of the tested network, we impose layer weight regularizers on all 2D convolutional layers as well as all fully-connected layers. We do this by applying a regularizer penalty on the layer's kernel. We implement three different versions of each architecture with a distinct regularizer. We compute the penalty for the layer's kernel by the l_1 norm in which the loss is equivalent to $\mathcal{L}(x) = l_1 \times \|x\|$ where l_1 is the regularization factor set to 0.01. We also run a series of experiments where the models are regularized by the l_2 norm such that the loss is computed as $\mathcal{L}(x) = l_2 \times x^2$ where l_2 is the regularization factor set to 0.01. Lastly, we run a series of experiments where the models are regularized by both the l_1 and l_2 norms, e.g. elastic net. As we can see in Figure

Model	Regularization	DeepFool[11]	FGSM[4]	PGD[10]
ResNet	**Base**	19.60	23.80	18.00
	L1	**34.20**	16.00	28.40
	L2	25.20	11.40	13.20
	Elastic Net	33.80	**37.40**	**38.20**
MobileNet	**Base**	11.40	22.20	22.00
	L1	27.20	22.60	**28.00**
	L2	**38.40**	**30.20**	25.40
	Elastic Net	25.40	27.00	19.40
VGG16	**Base**	8.40	7.60	1.60
	L1	10.80	8.00	8.00
	L2	11.30	10.20	8.00
	Elastic Net	**11.40**	**11.40**	**10.80**

Table 1: Accuracy on testing samples that have been attacked by different adversarial algorithms (DeepFool, FGSM, and PGD) on CV Models Trained on CIFAR-10 (ResNet, MobileNet, VGG-16). Quantitatively, we see gains in robustness for regularization techniques across all architectures, and all attacks.

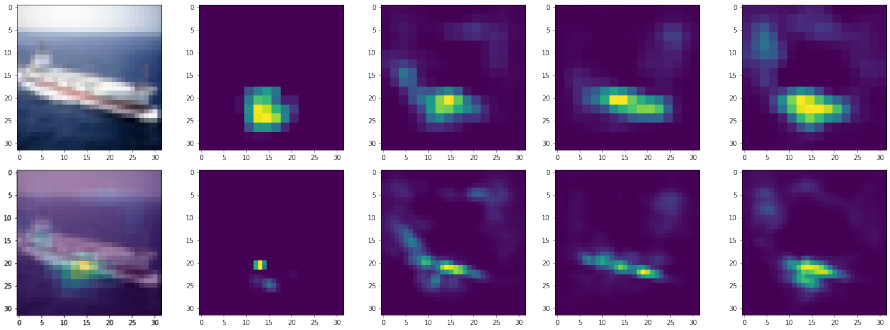
2 and in Table 1, imposing sparse regularization on these deep learning architectures ensures a more robust generalized model. We believe the restriction on the number of nodes activated encourages only the most important features to be represented when encoding into an embedding. Indeed, it appears that in every account, the regularized models demonstrate some effectiveness towards mitigating adversarial attacks.

The training process, e.g. model accuracy and loss, can be seen in Figure 3. We can see that the regularized models take slightly longer to converge, but all models are able to achieve the same accuracy and loss on the training and validation sets. However, in the case of no regularization, the validation loss begins to vary wildly, Figure 3(d). This is likely due to the fact that the model is beginning to overfit to the training data.

4.3 Stability in Class Attribution

For our final experiment, we investigate how regularization can improve robustness in the attribution task. Namely, we propose that for a model to be robust, it should maintain a level of consistency in the explanation e.g. class attribution maps, as the hyperparameters of the system are perturbed. If the attribution maps drastically vary from small changes in the input size, or patch size, then the model would not be considered robust. We can quantitatively measure the level of consistency between attribution maps by a pixel-wise sum of squared distances (SSD) between examples. For the attribution task, we use two distinct methods - Grad-CAM a gradient based attribution method, and an occlusion method using Sliding Patches.

We are able to visualize the attribution maps given example CIFAR images for the occlusion Sliding Patch method, Figure 4. In this method, we first slide a



(a) Original (b) Base (Δ26.7) (c) L1 (Δ9.0) (d)L2 (Δ12.1) (e) L1 L2 (Δ12.8)

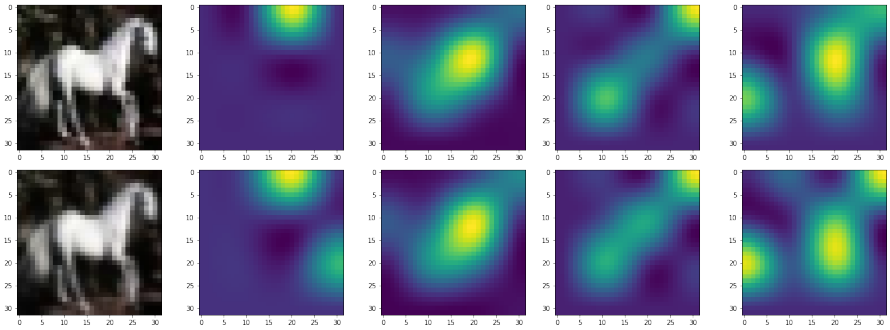
Fig. 4: Attribution heat maps generated by the Sliding Patch method. The first row shows the original image (a) and the heat map generated for the resulting class using a patch size of 4 and a jump size of 2. The second row shows an overlay of the heat map on the original image (a) and the heat map generated using a patch size of 2 and a jump size of 2. The sum of squared differences between row 1 and row 2 are shown as Δ for each of the methods.

gray patch of size 4×4 across the image with a stride (jump parameter) of 2. We observe the output of the model and can quantify how much the occluded patch effects the classification output. The attribution heat maps for the 4×4 patch can be seen in first row of Figure 4. We then change the parameters of the system by sliding a 2×2 patch across the image with a stride of 2. We can see that this parameter change does have an effect on the end result as seen in row 2 of Figure 4. We compute the SSD from row 1 and row 2 to compute the difference between maps. A more consistent map would have lower SSD. As shown numerically over 1,000 random CIFAR-10 images, see Table 2, L1 regularization has the best and lowest overall attribution change for the Sliding Patch method.

Next, we evaluate the Grad-CAM method, which can be seen in Figure 5. Similar to the occlusion method, we can compare the SSD between row 1 and row 2 attribution maps. However, in this case there is no internal parameterization of the Grad-CAM method. Thus, in order to evaluate stability of the method, we slightly transform the input image. For our experiment, we chose to blur the original image via a Gaussian kernel of size $\sigma = 0.5$. The intuition is that a slightly blurred version of an image should not change the attribution map drastically. And in fact, we show that the regularized models do improve the stability of attribution as quantitatively shown in Table 2.

5 Conclusion

In this paper, we have analyzed the performance of image classification architectures and the effect that robustness and sparsity has on their robustness against



(a) Original (b) Base (Δ15.6) (c) L1 (Δ2.6) (d)L2 (Δ4.4) (e) L1 L2 (Δ18.2)

Fig. 5: Attribution heat maps generated by the Grad-CAM method. The first row shows the original image (a) and the heat map generated for the resulting class. The second row shows the original image blurred by a Gaussian kernel with $\sigma = 0.5$ (a) and the heat map generated on the blurred image. The sum of squared differences between row 1 and row 2 are shown as Δ for each of the methods.

Table 2: Sum of Squared Differences (SSD) of Attribution heat maps when altering a hyperparameter of the system. For Sliding Patches, we change the patch size from 4 pixels to 2 pixels. For Grad-CAM, we transform the input image by a Gaussian blur kernel with $\sigma = 0.5$.

Attribution	**Base**	**L1**	**L2**	**Elastic Net**
Sliding Patch[17]	26.31 ± 27.19	**18.51 ± 15.28**	18.62 ± 13.48	20.92 ± 17.83
Grad-CAM[14]	24.53 ± 34.19	15.13 ± 24.95	15.12 ± 20.92	**13.96 ± 22.51**

adversarial attacks and the stability when performing image attribution. We have shown that enforcing sparsity, especially in the form of regularization upon the convolutional and fully-connected layers within these deep neural architectures has helped the robustness of the model and outperform the base model without regularization in correctly classifying images despite various adversarial attacks against the dataset. We looked at different attacks exploiting vulnerabilities in the architectures themselves as well as some attacks whose only objective was to misclassify the image. Our results as indicated in Table 1 and visualized in Figure 2, show the benefits of imposing regularization to combat adversarial attacks to computer vision models. Furthermore, we show that sparse regularization creates stability within image attribution frameworks. Specifically, Elastic Net seemed to produce the most robust results across all attacks and all attribution methods.

References

1. Bansal, N., Agarwal, C., Nguyen, A.: Sam: The sensitivity of attribution methods to hyperparameters. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8673–8683 (2020)
2. Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. IEEE Symposium on Security and Privacy (2017), https://arxiv.org/pdf/1608.04644.pdf
3. Fong, R., Patrick, M., Vedaldi, A.: Understanding deep networks via extremal perturbations and smooth masks. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2950–2958 (2019)
4. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples (2014)
5. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. CoRR **abs/1512.03385** (2015), http://arxiv.org/abs/1512.03385
6. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. CoRR **abs/1704.04861** (2017), http://arxiv.org/abs/1704.04861
7. Kim, E., Hannan, D., Kenyon, G.: Deep sparse coding for invariant multimodal halle berry neurons. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1111–1120 (2018)
8. Kim, E., Rego, J., Watkins, Y., Kenyon, G.T.: Modeling biological immunity to adversarial examples. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4666–4675 (2020)
9. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Commun. ACM **60**(6), 84–90 (May 2017). https://doi.org/10.1145/3065386, https://doi.org/10.1145/3065386
10. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083 (2017)
11. Moosavi-Dezfooli, S., Fawzi, A., Frossard, P.: Deepfool: a simple and accurate method to fool deep neural networks. CoRR **abs/1511.04599** (2015), http://arxiv.org/abs/1511.04599
12. N. Papernot, P. McDaniel, X.W.S.J., Swami, A.: Distillation as a defense to adversarial perturbations against deep neural networks. IEEE Symposium on Security and Privacy (2016), https://arxiv.org/pdf/1511.04508.pdf
13. Nicolae, M., Sinn, M., Minh, T.N., Rawat, A., Wistuba, M., Zantedeschi, V., Molloy, I.M., Edwards, B.: Adversarial robustness toolbox v0.2.2. CoRR **abs/1807.01069** (2018), http://arxiv.org/abs/1807.01069
14. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE international conference on computer vision. pp. 618–626 (2017)
15. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
16. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. In: International Conference on Learning Representations (2014), http://arxiv.org/abs/1312.6199
17. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: European conference on computer vision. pp. 818–833. Springer (2014)