A Survey of Various Techniques for Predictions of Unrecognizable Images

Uttra Singh
Department of Information Technology
NIIST
Bhopal, India
singh.uttra93@gmail.com

Asst. Prof. Angad Singh
Department of Information Technology
NIIST
Bhopal, India
Angada2007@gmail.com

Abstract— Due to great advancement in "deep learning" designs schemes that approximately emulate the visual or auditory cortex, with a objective of holding out image or video (i.e. a live web-cam stream) or sound processing tasks have been required numerous concentration both in the technical community and the admired social media. In large training, Deep Neural Networks (DNNs), including important achievements in training convolutional neural networks (convnets) to recognize natural images.

Index Terms— Deep learning, Deep Neural Networks, Convolution, Convolutional Neural Network

I. Introduction

Recent discoveries uncovered laws in machine learning algorithms for example deep neural networks. Deep neural networks seem susceptible to little amounts of non-casual noise generated by exploiting the input to output mapping of the network concerning this noise to a participation image significantly reduces classification presentation. It has showed its usefulness in many areas i.e. bioinformatics [1], speech recognition [2], and computer vision [3]. Moreover, it has produced state-of-the-art results in various applications. Hence, deep learning is appropriate more and more popular nowadays over other learning algorithms. The research ideally along both paths to have a better understanding about the robustness of deep networks, what makes them unstable and how we can stabilize them? So far, we have found a simple method to extract the adversarial perturbations in deep networks quite fast. Then, we can concentrate more on formalizing the notion of robustness in deep networks and make efforts to quantify that. As a long-term goal, we would be interested to deal with the difficulty of designing robust learning algorithms.

Deep networks have produced significant gains for various visual recognition problems show the way to high contact educational and business applications. Modern effort in deep networks emphasized i.e. it is simple to produce images that humans would not at each and every one categorize as a particular object class, yet networks categorize such images high assurance as that given class – deep network are easily fooled with images humans do not consider meaningful. The congested set environment of deep networks forces them to pick from one of the known classes leading to such objects. Modern research in deep networks has considerably got better many characteristics of visual recognition [4, 5]. Co-evolution of prosperous illustrations, scalable classification techniques and large datasets has caused in many business applications [6]. Alternatively, an extensive choice of operational challenges occurs while organizing recognition schemes in the dynamic and ever-changing authentic world. A huge collection of recognition schemes are planned for a static closed world, where the most essential theory is that all categories are known a priori. Deep networks, like many standard machine learning tools are aimed to carry out closed set recognition.

Deep learning or the make use of of deep (i.e., many-layered) convolutional neural networks for machine recognition and classification, is advancing the limits of performance in domains as varied as computer vision, speech, and text [7, 8]. Improvements in both hardware and software performance have enabled the development of larger networks that have achieved record results [9]. The promise of deep learning is to automate feature engineering, a task that otherwise requires application of both domain expertise and machine learning expertise. Neural networks present a logical structure to train comprehensive models that compose

featurization and classification components in a unified pipeline.

However, when exposed to noise, reliability of these networks stays questionable. Szegedy et al. [10] have discovered significant interesting characteristics of these networks. In their research, they require of deep neural networks' robustness against small unperceivable perturbations on the input images has been discovered. These perturbations caused the network to misclassify images with large confidence. Such perturbed images are in literature referred to as adversarial examples. By causing classification of an image that belongs to one class as an image from another class based only on a subtle change in a few pixels, they pose a big problem for neural networks' trustworthiness that is addressed in this work.

II. Theoretical Background

Pattern recognition is a branch of machine learning methods for discovering and recognizing regularities in data. It is broadly used and a successful area with several of algorithms at disposal. Pattern recognition models take images or patterns as input, and produce output values called predictions. Several widely used social networks have implemented face recognition algorithms [11] to recognize faces on user-provided photos. Apart from the Internet-based applications, computer vision tasks play an explanation responsibility in commercial and industrial applications, starting from camera autofocus up to the autonomous robots (or vehicles) perception system. Among the large variety of computer vision applications, military use is possible, e.g. for detection of opposing force's military objects.

Currently, the most successful models for visual recognition are the deep neural networks (DNNs) [9]. DNNs are neural networks consisting of several layers. Their depth enables them to learn deeper representations of data leading to overwhelming performance over other machine learning methods. Over the earlier period, DNNs have gained a group of interest by researchers in addition to by the industry. However DNNs were designed in early 80s, there were insufficient computational resources and knowledge how to train such networks. Training of deep neural networks proposed in early 80s became feasible the recent years, with the vast improvement in computational performance, resulting in shorter training time. Huge databases of images essential for training DNNs became available due to enhancement of communication availability and bandwidth. The circumstances of the skill deep neural networks show remarkable results in complex tasks for instance image classification and speech recognition [9, 12].

III. Convolutional Neural Networks

In general, an artificial neural network consists of a succession of layers of so-called neurons. A neuron calculates a function on inputs from the previous layer and exceeds the effect from time to time called the neuron's activation to outputs in the succeeding layer. Within each layer all neurons calculate the similar function but individual neurons may have separate sets of inputs and outputs and may allocate different weights to their inputs. Different types of layers are characterized by the number and pattern of connections between neurons.

In a fully connected layer, the neurons receive input from every output in the preceding layer. In a locally connected layer, the neurons are indexed spatially and each only gets input from close to outputs. A convolutional layer is a kind of locally connected layer where the weights that each neuron is appropriating to its inputs are split in a particular method. Neural network design for image classification joins a variety of functions and connectivity arrangements using several layers. The first layers are convolutional and produce a factorized representation of an image. Afterwards, a non-linear transformation is often applied, followed by a linear classifier such as logistic regression or SVM. The amount produced of the network (usually a vector of predicted probabilities) can be assessed relative to a true image label, and the result can be used by an optimization algorithm like gradient decline to train the network. The enormous demand of neural networks is that training can be useful to the featurization layers in addition to the classifier. This end-to-end training algorithm called back-propagation is the existing high-tech in image classification and other areas such as speech recognition [8]. Typically, training is an iterative procedure that involves multiple passes of the input data until the model converges.

Convolution: The convolution of an image is created by affecting a filter to image and creates a new image. A filter is a k×k weight-matrix where k is an odd number i.e. create unique center matrix. Pixels in the convolved image are produced by placing the filter on top of the image, with its center aligned at the corresponding input pixel, and computing the dot product of the filter with the pixels below it. The convolution can be visualized as the result of moving a filter across the image that replaces each pixel with some function of its neighborhood. This process is demonstrated in Figure-1 [13].

In the context of neural networks, a convolutional layer applies many filters to its input to generate a feature map, which is essentially a stack of convolved images, or equivalently, one convolved image with an arbitrary number of channels per pixel. In addition, convolutional layers are often bundled with several auxiliary layers that apply a fixed transformation to the convolved feature map. These auxiliary layers include normalization (of pixel values within a neighborhood), pooling (aggregation of small patches of pixels, for example, by averaging or taking the maximum pixel value), down-sampling, and the application of various non-linear functions to pixel values.

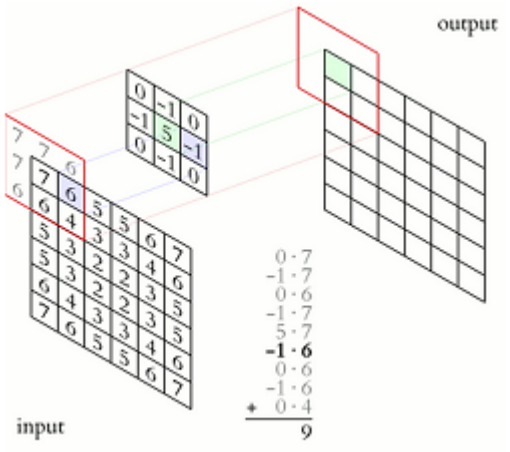

Figure-1: Image Convolution [13].

So far, our discussion of convolution has been inclined to the abstract, and the reader would be justified to ask "what's the point?" In fact, convolutions are capable of transforming images in many useful and concrete ways, like emphasizing edges and computing gradients of hue and value. Moreover, deep successions of convolutions have been shown to produce image encodings that are favorable for classification, namely due to invariance to translation and deformation [14]. But exactly what is computed—and its usefulness for classification—depends on the filters used, and therefore success of a convolutional network crucially depends crucially on choosing good filters.

Design Space: Recent success in image classification has come from going deeper: using more filters in more layers. Back-propagation automates the training of the filter weights in these deep networks, and with larger data sets, like ImageNet deeper and richer models can be trained. But the ease of training deep networks belies the difficulty of their design.

While we have presented a basic overview of the workings of a convolutional neural network, we have glossed over several details of the network structure that represent design points that model's builder must consider. Particular parameters that must be tuned include:

- **Size of filters** - Determining an appropriate size for convolutional filters is not a precise science. Too small and the features are in some sense "too common", too huge and model complication blow up with minute benefit.
- **Number of layers** - Additional layers seems to improve model performance, but they increase model complexity, and too many layers may cause the signal-to-noise ratio during back-propagation to be too low for the first few layers to be trained into anything useful.
- **Filters per layer** - Again, models generally perform better with more filters, but at what point are diminishing returns outweighed by the increased model complexity and training time?
- **Layer connectivity** - Besides convolutional layers, what other types of layers should be used? Some top results have mixed fully-connected and locally-connected layers with convolutional ones to huge result [9].
- **Initialization** - Should we initialize our weights uniformly, randomly, or to some structure? Does it make a difference?
- **Auxiliary layers** - The choice of pooling and normalization function can enclose a significant impact on model accuracy, and each comes bundled with several numeric parameters. How do you tune them?
- **Non-linear functions** - Surprisingly, the choice of what non-linearity to apply after a convolution can have dramatic impact on training run-time performance. Indeed, [9] note that the exploit of the "relu" non-linearity instead of the sigmoid function makes a large difference in their models' performance.
- **Optimization parameters** - As with any ML model, learning parameters like step size and regularization must be tuned to maximize accuracy and convergence speed. Algorithms like AdaGrad [15] are frequently making use of manage some of these parameters; functional dependencies between parameters can make tuning difficult.

All of these parameters can have a spectacular contact on model performance and complexity.

IV. Conv Nets: A Modular Perspective

In some years, deep neural networks have showed the way to penetrate effects on a selection of pattern recognition problems, i.e., computer vision and voice recognition. One of the fundamental parts important to these effects has been a unique type of neural network called a convolutional neural network.

At its most essential, convolutional neural networks can be consideration of as a type of neural network that uses many matching copies of the similar neuron. This permits the network to have groups of neurons and communicate computationally big models while maintenance the number of actual constraints – the values telling how neurons behave – that require to become skilled at moderately small.

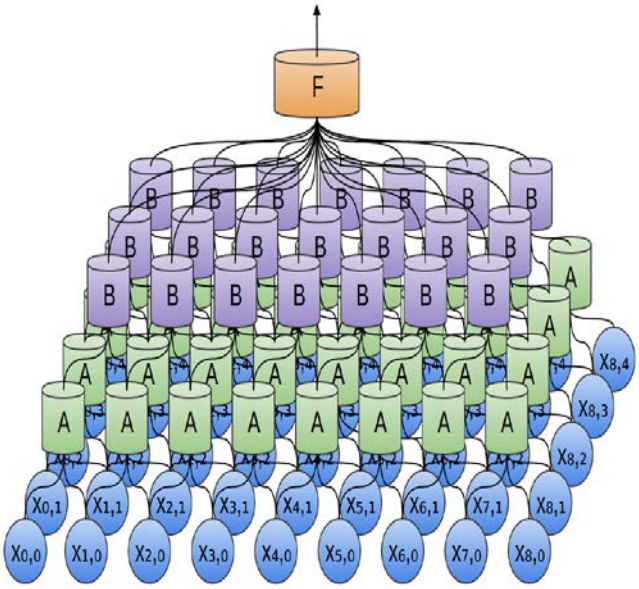

Figure-2: 2D Convolutional Neural Network

This deception of having numerous copies of the similar neuron is approximately equivalent to the idea of functions in mathematics and computer science. When programming, we inscribe a function once and utilize it in many positions – not writing the similar code a hundred times in different positions creates it earlier to program and effects in smaller amount of errors and bugs. Similarly, a convolutional neural network can gain knowledge of a neuron formerly and use it in many places, making it easier to become skilled at the model and reducing error. For image classification, it is common to use convolutional neural networks (CNNs) [17] as they were designed to extract information from 2D and higher order input spaces.

Convolutional neural networks, as their multiple levels of characteristic extracting layers make use of a minimum of preprocessing; hence it is not necessary to consider feature extraction issues. CNN's weights are designed to shape a convolutional filter that is replicated over the whole visual field. All units of the convolutional layer share the same weights within the layer, what decreases number of free parameters to learn, thus simplifies training process. The filter is utilized to convolve an image; each filter convolves pixels it covers. Outputs of all these filters form a feature map. Convolutional layers usually contain several feature maps for richer representation of the image content. Each feature map is created by a different filter. Convolutional layer is typically defined by number of characteristic maps, kernel size i.e. size of the filter and by stride parameter i.e. a size of the step over image pixels when applying filter.

V. Deep neural networks

The basic concept of learning algorithms is to learn features of the data with the purpose of produce some output, for example the prime concern of predictive modeling is to forecast the probable class. True classes and correct ranking can be either provided by a teacher in the training phase in so called supervised learning, or has to be found by an algorithm in case of learning without teacher, the unsupervised learning. Input object is represented by different stages of features. For visual problems, levels vary from low level features found in local neighborhood to higher level features in form of curves or shapes. Probably, the most broadly used deep neural networks are the convolutional neural networks (CNNs) [9]. In CNNs, features are captured at different levels by convolutional layers (see Figure 3). To take out higher level features, more convolutional layers are necessary. By these means, deep learning methods are able to learn dissimilar stages of abstraction, example of such levels (from highest to lowest) for a single object can be: animal, mammal, dog, German shepherd.

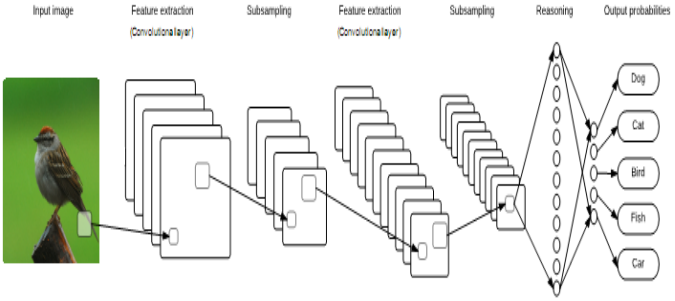

Figure 3: An example of deep convolutional neural network.

Deep neural networks are applied mostly for the pattern recognition problems. Robustness of DNNs is thus examined within visual recognition problems, especially within one of the most frequent problems, the optical character recognition. Processing power is not the only challenge when training DNNs. Typically used gradient descent optimization was insufficient to train a DNN. The problem lies in its random initialization, after which a model is unable to converge to the global optima during the training. This issue has been solved recently by greedy layer-wise unsupervised pre-training [16], which initializes weights close to a local optima.

VI. Literature Survey

Szegedy et al. [10] opened a whole new branch for research of DNNs. Instead of describing improvement in DNNs' generalization performance, they have focused on discovering neural networks' weaknesses. Firstly, Szegedy et al. showed that it is the entire space of activations rather than individual units that contains semantic information. The rest of their work is oriented on finding DNN's blind spots.

The most important findings made by Szegedy et al. in [10] are:

1. For all the networks studied, for every tested image, the authors were able to generate an adversarial example, which was for humans visually almost indistinguishable from original image that was misclassified by the original network.

2. Cross model generalization: a huge amount of adversarial examples are misclassified by networks trained with different hyper-parameters (number of layers, initial weights, etc.).
3. Cross training-set generalization: a huge amount of examples are misclassified by networks trained on a disjoint training set.

The discoveries pose questions, how the universal approximators can be so vulnerable to such subtle changes. These discoveries undermine smoothness property of neural networks, claiming that inputs close to each other are supposed to be assigned to the same class. Their experimental effects propose that using adversarial examples in training process may improve generalization performance.

Nguyen et al. [18] have investigated a reverse problem. From original image data set, they have created visually meaningless images not recognizable by humans, which are classified by a neural network as one of the classes with confidence reaching 99.99%. The authors named these examples "fooling" images. This problem can be explained by creating a special class for fooling images. Training a network this way make it difficult to find new fooling images, since the network has learned features generic to these fooling images. Nguyen et al. made a hypothesis that these fooling examples are based by the discriminative character of classifier, permitting algorithm to find an example that is far away from discriminative boundary with from all the data that has been seen before.

The research of Goodfellow et al. [19] provides a discussion about reasons, why the adversarial examples exist. Opinions connecting adversarial examples with high-nonlinearity of DNNs are opposed by later assumptions made by Goodfellow et al. that claim, being of adversarial examples stem from models being too linear. Authors believe, adversarial perturbations are dependent on model's weights, which are similar for different models learned to perform the similar task. They observed a generalization of adversarial noise across different natural examples is caused by the fact that adversarial perturbations are not dependent on specific point in space but on direction of the perturbation. Further in the work by Goodfellow et al., experiments comparing resistance of models with different capacity against adversarial and fooling examples have been performed. In the paper [19], it was shown that models, which are simple to optimize yield easily to adversarial and fooling examples, thus they have no capacity to resist these perturbations. One of Goodfellow et al. studies

S. No.	Paper	Author	Advantages	Issues
1	Convolutional-recursive deep learning for 3D object classification	Socher, R., Huval, B., Bhat [21]	Here they reported work using a deep learning neural network to recognize patterns in YouTube videos	Overall accuracy at recognizing patterns in videos was not particularly high
2	Dropout: A simple way to prevent neural networks from over fitting	N. Srivastava, G. Hinton, A. Krizhevsky [22]	The principle of dropout technique is to randomly deactivate neurons and their connections during the training phase.	Learning models suffer from overfitting, i.e. co-adapting to specific input data, which leads to poor generalization of unseen observations.
3.	Provable bounds for learning some deep representations	S. Arora, A. Bhaskara [23]	A one should analyze the correlation statistics of the last layer and cluster them into groups of units with high correlation	To find the optimal local construction and to repeat it spatially.
4.	Deep learning with cots hpc systems	Coates, A., Huval, B. [24]	Availability of larger training data sets along with increased computation power through heterogeneous computing. This system enables larger models to be trained on more data, while also reducing turnaround time	Computational power is so essential to development in deep learning, they built a supercomputer planned for deep learning
5.	DeCAF: A deep convolutional activation feature for generic visual recognition.	Donahue, J., Jia, Y., Vinyals [25]	It provides non-parametric analysis of invariance, showing which patterns from the training set activate the feature map to show visualizations that identify patches within a dataset that are responsible for strong activations at higher layers in the model.	The problem is that for higher layers, the invariances are extremely complex so are poorly captured by a simple quadratic approximation.
6.	Building high-level features using large scale unsupervised learning	Le, Q. V., Ranzato [26]	To perform inverse optimization on a network trained by unsupervised learning to construct the optimal inputs for particular neurons. In particular, they find single deep neurons trained to respond to faces	Feature inversion has been applied to convolutional neural networks to obtain several interesting patterns

examined shallow RBF (radial basis function) networks, which proved to be more resistant against adversarial and fooling examples in a way, where inputs on which the network is not sure are classified with low confidence. Adversarial examples are often classified by RBF network incorrectly, but with low confidence. Adversarial training is presented by Goodfellow et al. as a possible tool for even further regularization than dropout.

Gu & Rigazio [20] used various preprocessing methods to diminish adversarial perturbations. They have tested several denoising procedures including injection of additional Gaussian noise and subsequent Gaussian blurring. More sophisticated methods using autoencoder trained on adversarial examples or standard denoising autoencoder proved to be more effective. Autoencoders could easily learn simple structure of adversarial perturbations in order to eliminate them. Despite the ability of DNN stacked to the top of the autoencoder to handle adversarial perturbations of the original network, the stacked network became more sensitive to new adversarial examples. New adversarial examples required smaller perturbations than adversarial examples of the original network to perturb it. Gu & Rigazio believe DNN's sensitivity is affected by training procedure and objective function rather than by network topology. As a feasible explanation to achieve local generalization in the input space, they propose a deep contractive neural network.

VII. Conclusion

As tremendous development in training dominant, deep neural network models that are forthcoming and even beating human abilities on a range of difficult machine learning tasks with better learned probabilistic models over the input and activations of higher layers, a huge quantity additional arrangement may be visible as a expansion atmosphere for training deep neural networks.

References

[1] Jian Zhou and Olga G. Troyanskaya. Deep supervised and convolutional generative stochastic network for protein secondary structure prediction. In Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014, pages 745–753, 2014.

[2] Geoffrey Hinton, Li Deng, Dong Yu, Abdel rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara Sainath George Dahl, and Brian Kingsbury. Deep neural networks for acoustic modeling in speech recognition. IEEE Signal Processing Magazine, 29(6):82–97, November 2012.

[3] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, editors, Advances in Neural Information Processing Systems 25, pages 1097–1105. Curran Associates, Inc., 2012

[4] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on. IEEE, 2015.

[5] Ken Chatfield, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In Proceedings of the British Machine Vision Conference, (BMVC), 2014.

[6] Thomas Dean, Mark A Ruzon, Mark Segal, Jonathon Shlens, Sudheendra Vijayanarasimhan, and Jay Yagnik. Fast, accurate detection of 100,000 object classes on a single machine. In Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, pages 1814–1821. IEEE, 2013

[7] Zeiler, M. D., and Fergus, R. Visualizing and Understanding Convolutional Networks. arXiv.org 2013.

[8] Dean, J., Corrado, G. S., Monga, R., Chen, K., Devin, M., Le, Q. V., Mao, M. Z., Ranzato, M., Senior, A., Tucker, P., Yang, K., And Ng, A. Y. Large scale distributed deep networks. In NIPS, 2012

[9] Krizhevsky, A., Sutskever, I., And Hinton, G. Imagenet classification with deep convolutional neural networks. 1106–1114, 2012.

[10] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus, “Intriguing properties of neural networks," CoRR, vol. abs/1312.6199, 2013.

[11] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, “Deepface: Closing the gap to human-level performance in face verification," in Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, pp. 1701-1708, June 2014.

[12] G. Hinton, L. Deng, D. Yu, G. Dahl, A. rahman Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury, “Deep neural networks for acoustic modeling in speech recognition," Signal Processing Magazine, 2012.

[13] WIKIPEDIA, Kernel (image processing) — Wikipedia, the free encyclopedia. 2013.

[14] Bruna, J., And Mallat, S. Invariant scattering convolution networks. arXiv preprint arXiv:1203.1513. 2012.

[15] Duchi, J., Hazan, E., And Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. J. Mach. Learn. Res., 2121–2159, 2011.

[16] Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle, et al., “Greedy layer-wise training of deep networks," Advances in neural information processing systems, vol. 19, p. 153, 2007.

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks," in Advances in Neural Information Processing Systems 25 (F. Pereira, C. Burges, L. Bottou, and K. Weinberger, eds.), pp. 1097-1105, Curran Associates, Inc., 2012.

[18] A. Nguyen, J. Yosinski, and J. Clune, “Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," 2015.

[19] I. J. Goodfellow, J. Shlens, and C. Szegedy, \Explaining and harnessing adversarial examples," CoRR, vol. abs/1412.6572, 2014.

[20] S. Gu and L. Rigazio, “Towards deep neural network architectures robust to adversarial examples," CoRR, vol. abs/1412.5068, 2014.

[21] Socher, R., Huval, B., Bhat, B., Manning, C.D., Ng, A.Y.: Convolutional-recursive deep learning for 3d object classification. In: Advances in Neural Information Processing Systems 25, p. 665-673 2012.

[22] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from over fitting," Journal of Machine Learning Research, vol. 15, pp. 1929-1958, 2014.

[23] S. Arora, A. Bhaskara, R. Ge, and T. Ma. Provable bounds for learning some deep representations. CoRR, abs/1310.6343, 2013.

[24] Coates, A., Huval, B.,Wang, T.,Wu, D.J., Catanzaro, B.C., and Ng, A.Y. Deep learning with cots hpc systems. In ICML (3)’13, pp. 1337–1345, 2013.

[25] Le, Q. V., Ranzato, M., Monga, R., Devin, M., Chen, K., Corrado, G. S., Dean, J., And Ng, A. Y. Building high-level features using large scale unsupervised learning. arXiv preprint arXiv:1113.6209, 2011.