CONFIDENCE ESTIMATION IN DEEP NEURAL NETWORKS VIA DENSITY MODELLING

Akshayvarun Subramanya *Suraj Srinivas* *R.Venkatesh Babu*

Video Analytics Lab, Department of Computational and Data Sciences
Indian Institute of Science, Bangalore
akshayvarun07@gmail.com, surajsrinivas@grads.cds.iisc.ac.in, venky@cds.iisc.ac.in

ABSTRACT

State-of-the-art Deep Neural Networks can be easily fooled into providing incorrect high-confidence predictions for images with small amounts of adversarial noise. Does this expose a flaw with deep neural networks, or do we simply need a better way to estimate confidence? In this paper we consider the problem of accurately estimating predictive confidence. We formulate this problem as that of density modelling, and show how traditional methods such as softmax produce poor estimates. To address this issue, we propose a novel confidence measure based on density modelling approaches. We test these measures on images distorted by blur, JPEG compression, random noise and adversarial noise. Experiments show that our confidence measure consistently shows reduced confidence scores in the presence of such distortions - a property which softmax often lacks.

***Index Terms*—** Deep Neural Networks, Deep Learning, Density Modelling, Confidence Estimation

1. INTRODUCTION

Deep neural networks have contributed to tremendous advances in Computer Vision during recent times [1, 2, 3]. For classification tasks, the general practice has been to apply a *softmax* function to the network's output. The main objective of this function is to produce a probability distribution over labels such that most of the mass is situated at the maximum entry of the output vector. While this is essential for training, softmax is often retained at test time, and the output of this function is often interpreted as an estimate of the true underlying distribution over labels given the image.

Images can often be corrupted by artifacts such as random noise and filtering. We require classifiers to be robust to such distortions. Recently, Goodfellow *et al*. [4] showed that it is possible for an adversary to imperceptibly change an image leading to high-confidence false predictions. This places Deep Neural Networks at a severe disadvantage when it comes to applications in forensics or biometrics.

Recent works [5, 6] have empirically demonstrated that the softmax function is often ineffective at producing accurate uncertainty estimates. By producing better estimates, is it possible to detect such adversarial examples? This leads us to ask - what constitutes a good uncertainty / confidence estimate? Is there a fundamental flaw in estimating confidences using softmax? This paper discusses these issues and proposes a novel density modelling-based solution to this end. The overall contributions of this paper are:

- We discuss the general problem of uncertainty / confidence estimation and show how softmax can exhibit pathological behaviour.
- We propose a novel method for estimating predictive confidence based on density modelling.
- We provide experimental evidence showing that the proposed method is indeed superior to softmax at producing confidence estimates.

This paper is organized as follows. Section 2 describes different approaches that have been taken to tackle the confidence estimation problem. Section 3 introduces terminology and describes our approach. Section 4 describes experimental setup and results. Finally, in Section 5 we present our conclusions.

2. RELATED WORKS

Uncertainty or Confidence estimation has gained a lot of attention in recent times. Gal *et al*. [5] presented a method of estimating the uncertainty in neural network model by performing dropout averaging of predictions during test time. Bendale *et al*. [6] presented Open set deep networks, which attempt to determine whether a given image belongs to any of the classes it was trained for. However, both these methods use the softmax function to compute uncertainty. We will show shortly that uncertainty estimates from softmax contain certain pathologies, making them unsuitable for this task.

Modern neural network architectures are sensitive to adversarial examples [7]. These are images produced by the addition of small perturbations to correctly classified samples. Generating adversarial examples to *fool* the classifier is one of the active areas of research in the Deep Learning Community. Goodfellow *et al*. [4] presented a method to generate adversarial examples and also showed that retraining the

network with these examples can be used for regularization. Nyugen *et al*. [8] also strengthened the claim that networks can be fooled easily by generating *fooling images*. Moosavi *et al*. [9] presented an effective and fast way of generating the perturbations required to misclassify any image. Moosavi *et al*. [10] also show that there exists universal adversarial perturbations given a classifier, that can be applied to any image for misclassification . Researchers have also shown that results of face recognition algorithms can be tampered by wearing a specific design of eyeglass frames [11]. Such examples present a challenge to the idea of using neural networks for commercial applications, since security could be easily compromised. These can be overcome (in principle) by using a good uncertainty estimate of predictions.

3. CONFIDENCE ESTIMATION

We first introduce the concept of predictive confidence in a neural network in a classification setting. Let (X, y) be random variables representing the training data where $X \in \Re^D, y \in \Re^N$ and $f(X) : \Re^D \rightarrow \Re^N$ be a function that represents the mapping of D-dimensional input data to the pre-softmax layer (z) of N dimensions, where N is the number of output classes. In this case, we define confidence estimation for a given input X as that of estimating $P(y|X)$ from z.

Intuitively, confidence estimates are also closely related to accuracy. While accuracy is a measure of how well the classifier's outputs align with the ground truth, confidence is the model's estimate of accuracy in absence of ground truth. Ideally we would like our estimates to be correlated with accuracy, i.e high accuracy $\Rightarrow$ high confidence on average, for a given set of samples. A model with low accuracy and high confidence indicates that the model is very confident about making incorrect predictions, which is undesirable.

3.1. Pathologies with Softmax and Neural Networks

Given the definitions above, and pre-softmax activation vector with elements $z = [z_1, z_2, ...z_N]$, the softmax function is defined as follows.

(a) 0.7 × Image **Softmax**: 0.8791 (b) Original Image **Softmax**: 0.9249 (c) 1.3 × Image **Softmax**: 0.9687

Fig. 1: An illustration of the softmax pathology on an image from the ImageNet dataset using the VGG-16 classifier.

$$P_s(y_i|X) = s_i(z) = \frac{e^{z_i}}{\sum_i e^{z_i}} \quad (1)$$

Here, $P_s(y_i|X)$ denotes the softmax estimate of the desired probability $P(y_i|X)$ for label y_i. We can easily see that for neural networks with monotonically increasing activation functions, $f(kX) \geq f(X)$, for any $k > 1$. This is because this property of linear scaling applies to all layers of a neural network - convolutions, fully connected layers, max-pooling, batch normalization and commonly used activation functions. As a result, it applies to the entire neural network as a whole. Hence, the pre-softmax vector transforms to $\|z'\| = \|f(kX)\| \geq \|z\|$. This also trivially holds for multi-class linear classifiers of the form $g(X) = W^T X$. In such cases, the following lemma applies.

Lemma 3.1. *Let $z = [z_1, z_2, ...z_i, ...z_N]$ be a pre-softmax activation vector, such that $z_i = max(z_1, ...z_N)$. Given a positive scalar $k > 1$ and the softmax activation function $s_i(z)$ given in Equation (1), the following statement is always true*

$$s_i(kz) > s_i(z)$$

The proof for this lemma appears in the Appendix. This implies that for softmax, $P_s(y_i|kX) > P_s(y_i|X)$. This also indicates that irrespective of the structure of data, an input X with large ℓ_2 norm always produces higher confidence than that with a lower ℓ_2 norm. This exposes a simple way to boost confidence for any image - by simply increasing the magnitude. This is illustrated with an example in Figure 1. Clearly, this method of computing confidence has pathologies and therefore, must be avoided.

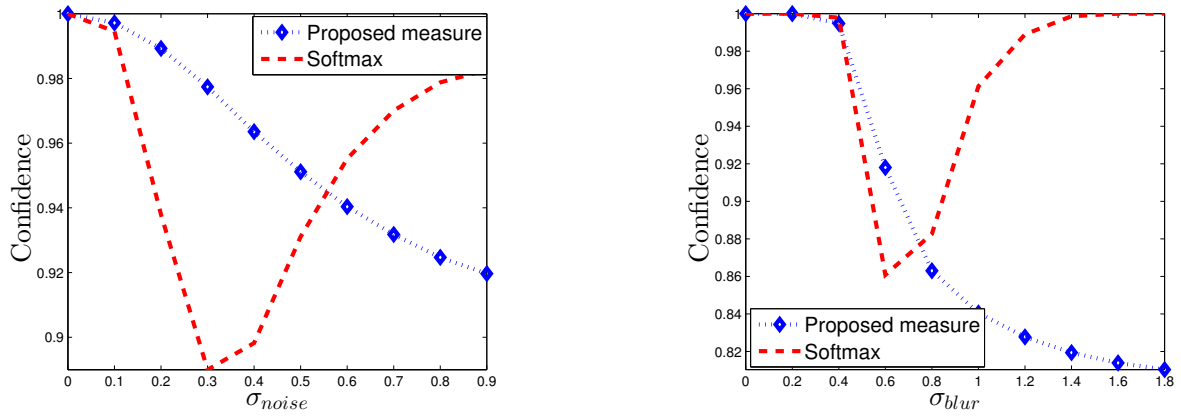
3.2. How to estimate better confidence?

The traditional notion of confidence for linear classifiers relies on the concept of distance from the separating hyper-plane, i.e.; the farther the point is from the hyper-plane, the more confident we are of the point belonging to one class [12]. However, this reasoning is fundamentally flawed - points with large ℓ_2 norms are more likely to lie far away from a given hyper-plane. As a result, such points are always in one class or another with very high confidence. This clearly ignores the structure and properties of the specific dataset in question.

A much better notion of confidence would be to measure distances with points of either class. If a given point is closer to points of one class than another, then it is more likely to fall in that class. Points at infinity are at a similar distance from points of all classes. This clearly provides a much better way to estimate confidence. We shall now look at ways to formalize this intuition.

3.3. Proposed method: Density Modelling

According to the intuition presented above, we require to characterize the set of all points belonging to a class. The



(a) Gaussian Noise

(b) Gaussian Blurring

Fig. 2: Comparison of confidence measures for different distortions applied on MNIST images. A good confidence measure must exhibit a monotonically decreasing profile. We see that the proposed method does indeed have such a profile, whereas softmax shows high confidence even for very high distortions. Note that here both confidence measures are scaled such that clean images correspond to confidence of one, for better visualization.

most natural way to do that is to create a density model of points in each class. As a result, for each class y_i, we compute $P(z|y_i)$, where z is the activation of the final deep layer, previously referred to as the pre-softmax activation. Given these density models, the most natural way to obtain $P(y_i|z)$ is to use Bayes Rule.

$$P(y_i|z) = \frac{P(z|y_i)P(y_i)}{\sum_{j=1}^{N} P(z|y_i)P(y_i)} \tag{2}$$

This lets us compute $P(y_i|z)$ efficiently. However we mentioned that we wish to compute $P(y_i|X)$ rather than $P(y_i|z)$. Since the mapping from X to z is deterministic (given by a neural network), we assume that $P(z) \sim P(X)$. Although the validity of this assumption may be debatable, we empirically know that there exists a one-to-one mapping from a large family of input images X to corresponding features z. In other words, it is extremely rare to have two different natural images with exactly the same feature vector z. This assumption empirically seems to hold for a large class of natural images. Here, the prior $P(y_i)$ is based on the distribution of classes in training data.

In this work, we perform density modelling using multivariate Gaussian densities with a diagonal covariance matrix. Note that this is not a limitation of our method - the assumptions only make computations simple. As a result, if there are N classes in a classification problem, we compute parameters of N such Gaussian densities (μ, σ).

$$P(X|y_i) = \mathcal{N}(z|\mu_i, \sigma_i) \tag{3}$$

After we evaluate the likelihood for each class, we apply Bayes rule i.e Equation (2) by multiplying the prior and then normalising it, giving rise to confidence measure.

3.4. Gaussian in High Dimensions

High dimensional Gaussian densities are qualitatively different from low-dimensional ones. The following theorem explains this phenomenon.

Theorem 3.1 (Gaussian Annulus Theorem)**.** *For a d-dimensional spherical Gaussian with unit variance in each direction, for any $\beta \leq \sqrt{d}$, all but at most $3e^{-c\beta^2}$ of the probability mass lies within the annulus $\sqrt{d} - \beta \leq |x| \leq \sqrt{d} + \beta$ where c is a fixed positive coefficient.*

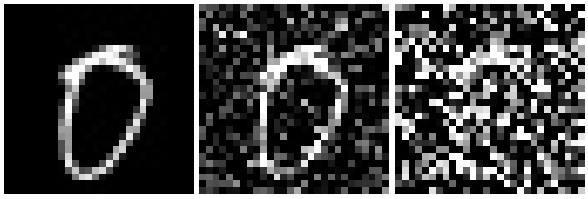
The proof for this theorem can be found in [13]. This theorem implies that for a high-dimensional Gaussian with unit variance, nearly all of the probability is concentrated in a thin annulus of width $O(1)$ with mean distance of $\sqrt{d}$ from the centre. This implies that almost all points within that density have more or less the same vanishingly small probability. This presents a problem when performing computations using Bayes rule. We require density functions such that different points have vastly differing densities.

One way to overcome this problem is to compute densities using a covariance of $d \times \sigma^2$ instead of σ^2. This ensures that majority of the points fall around the covariance rather than farther away. The resulting density values show variation among points, and do not have vanishingly small values, unlike in the previous case.

3.5. Overall process

Here we describe our confidence estimation process, and how to obtain confidence for a given new image. Training is per-

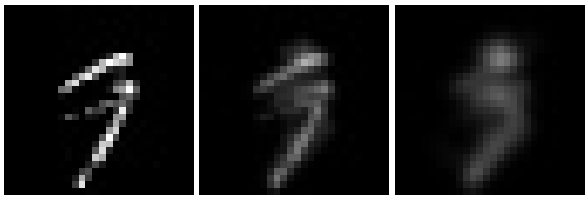
formed as usual, using the softmax activation function. After training, the training data is re-used to calculate the parameters of the density distribution in Equation 3. At test time, the label is obtained as before - by looking at the maximum entry of z (which is the same as the maximum entry of softmax output). However, confidence is obtained by first calculating all N density values and then applying Bayes' rule (Equation 2).



(a) Clean Image
Softmax: 0.999
Ours : 0.957

(b) Noise (σ=0.3)
Softmax: 0.921
Ours: 0.751

(c) Noise (σ=0.8)
Softmax: 0.994
Ours: 0.57



(d) Clean Image
Softmax: 0.388
Ours: 0.961

(e) Blur (σ=0.6)
Softmax: 0.667
Ours: 0.566

(f) Blur (σ=1.2)
Softmax: 0.807
Ours: 0.287

Fig. 3: An illustration of the effectiveness of our method on MNIST. The proposed confidence measure decreases when distortions are added to the image, while softmax remains high.

4. EXPERIMENTS

We evaluate our method on two datasets - MNIST handwritten digit dataset [14] and validation set of ILSVRC12 dataset [15]. For the MNIST dataset, we consider the LeNet-5 architecture with 2 convolution layers and 2 fully connected layers. For ImageNet, we consider VGG-16 architecture [2].

When presented with an unfamiliar image such as those with different types of distortions, a good measure of confidence must present predictions with reduced confidences. Examples of distortions include Gaussian Blurring, Gaussian Noise, JPEG Compression, Thumbnail resizing similar to those considered in [16]. In our experiments, we test this property of confidence measures on the following distortions.

- **Gaussian Noise:** Additive Noise drawn from a Gaussian distribution is one of the most common types of distortions that can occur in natural images. Here, we add incremental amounts of such noise and successively evaluate the confidence of the network. For MNIST, we vary the standard deviation of noise between 0 and 1, while for ImageNet it varies from 0 to 100. Note that the range of pixel values is [0,1] for MNIST and [0,255] for ImageNet. We see that both for Figure 2(a) and Figure 4(a), our method exhibits the desired monotonically decreasing profile whereas softmax does not.

- **Gaussian Blurring:** Gaussian Blurring represents a filtering operation that removes high-frequency image content. When there is less evidence for the presence of important features, confidence of classification must decrease. While this indeed holds for Figure 2(b) for the case of MNIST, we do not see this behaviour for ImageNet (Figure 4(b)). For MNIST, we vary the standard deviation of the Gaussian kernel from 0 to 2, while for ImageNet it is varied from 0 to 36.

- **JPEG Compression:** Another important family of distortions that we often encounter is the loss of image content that occurs due to JPEG Compression. Such a compression is lossy in nature, and this loss is decided by the quality factor used in JPEG compression, which is varied typically from 20 to 100. In this case we expect to see a monotonically increasing profile w.r.t quality index, which both softmax and the proposed method achieve.

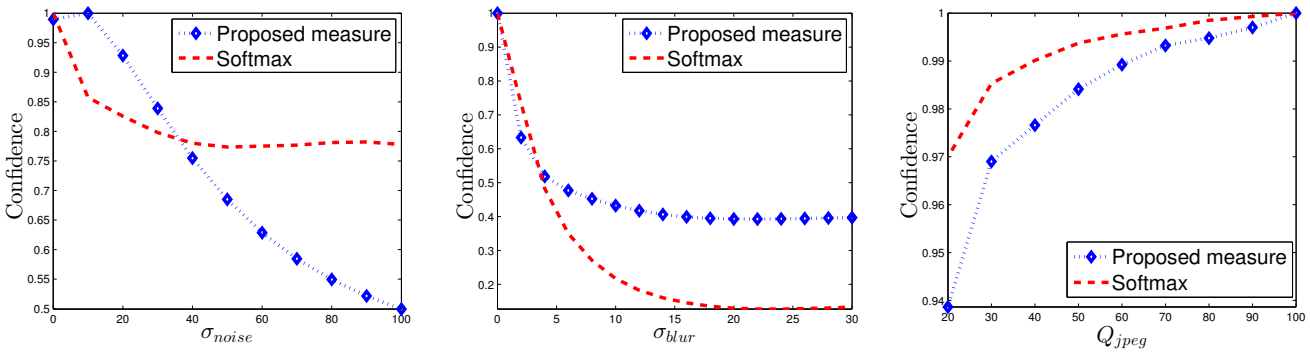
- **Adversarial examples:** Adversarial images were generated according to the algorithm provided in [9]. We generated adversarial images for the entire validation set of ILSVRC12 dataset. After presenting both the original and adversarial images and computing confidences for both, we consider a rise in confidence for the adversarial case (when compared to the clean image) as a failure. We count the number of times both methods - softmax and the proposed approach - fail, and present the results in Table 1.

# **Softmax fails**	# **Proposed measure fails**
5795	2214

Table 1: Performance of confidence measures for adversarial examples. Adversarial Images were generated for the entire validation set of ImageNet using [9].

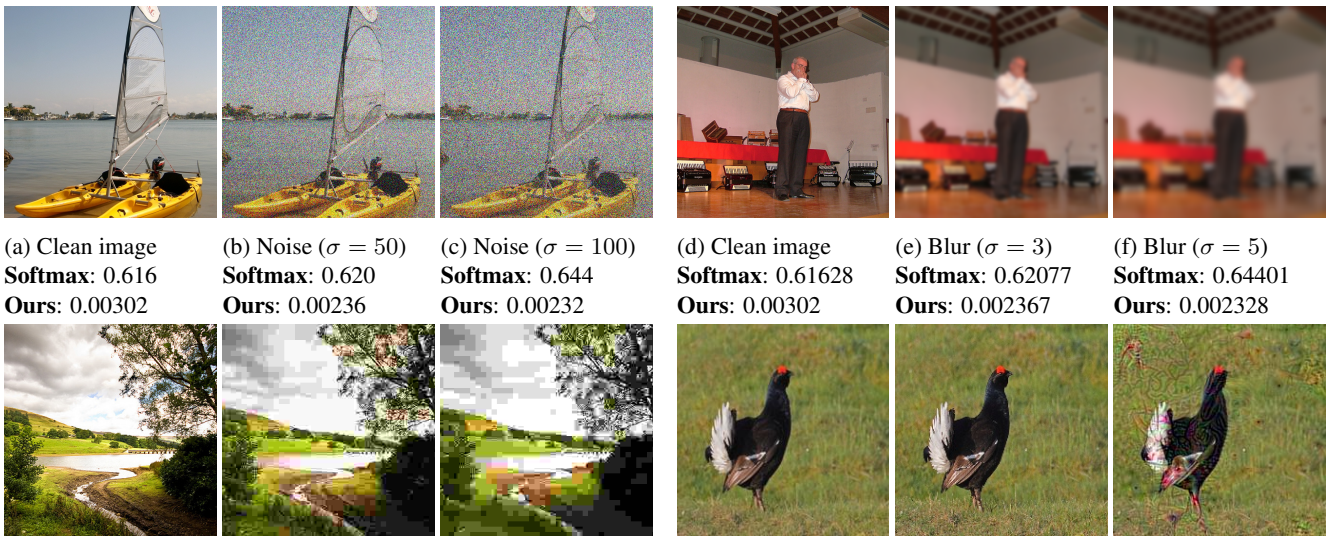
5. DISCUSSION AND CONCLUSION

We have introduced a novel method of measuring the confidence of a neural network. We showed the sensitivity of softmax function to the scale of the input, and how that is an undesirable quality. The density modelling approach to confidence estimation is quite general - while we have used a Gaussian with diagonal covariance, it is possible to use much more



(a) Gaussian Noise (b) Gaussian Blurring (c) JPEG Compression

Fig. 4: Comparison of confidence measures for various types of distortions applied on images in the ImageNet dataset. Both confidence measures are scaled such that the clean image always obtains confidence value = 1. This is done for better visualization. Figure (a) shows that the proposed approach is qualitatively better then softmax, which does not have a monotonically decreasing profile. For Figures (b-c), the proposed approach and softmax behave similarly.



(a) Clean image
Softmax: 0.616
Ours: 0.00302

(b) Noise ($\sigma = 50$)
Softmax: 0.620
Ours: 0.00236

(c) Noise ($\sigma = 100$)
Softmax: 0.644
Ours: 0.00232

(d) Clean image
Softmax: 0.61628
Ours: 0.00302

(e) Blur ($\sigma = 3$)
Softmax: 0.62077
Ours: 0.002367

(f) Blur ($\sigma = 5$)
Softmax: 0.64401
Ours: 0.002328

(g) Clean image
Softmax: 0.967
Ours: 0.00519

(h) JPEG (Q = 5)
Softmax: 0.968
Ours: 0.00508

(i) JPEG (Q = 1)
Softmax: 0.969
Ours: 0.00507

(j) Clean Image
Label: Black grouse
Softmax:0.291
Ours: 0.00283

(k) Adversarial Image
Label: Hen
Softmax: 0.378
Ours: 0.00235

(l) Adversarial Image
Label: Crane
Softmax: 0.98
Ours: 0.0018

Fig. 5: Figures (a-c) show the effect of additive Gaussian noise, Figures (d-f) show the effect of blur, Figures (g-i) show JPEG Compression, while Figures (j-k) illustrate adversarial examples. In all cases, the quality of image decreases from left to right. We see that the proposed approach shows confidence(unnormalized) drop when distortions are increased, whereas softmax confidence does not exhibit this property.

sophisticated models for the same task. Our results show that in most cases the diagonal Gaussian works well, and mostly outperforms a softmax-based approach. We hypothesize that performance suffers in case of Gaussian blurring and partly in the case of Adversarial examples due to difficulties associated with high-dimensional density estimation. Future work looking at more sophisticated density models suited to high-dimensional inference are likely to work better.

6. REFERENCES

[1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105.

[2] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in

International Conference on Learning Representations, 2015.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," *arXiv preprint arXiv:1512.03385*, 2015.

[4] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[5] Yarin Gal and Zoubin Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning," *arXiv:1506.02142*, 2015.

[6] Abhijit Bendale and Terrance E Boult, "Towards open set deep networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[7] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[8] Anh Nguyen, Jason Yosinski, and Jeff Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[9] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard, "Deepfool: a simple and accurate method to fool deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[10] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard, "Universal adversarial perturbations," *arXiv preprint arXiv:1610.08401*, 2016.

[11] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K. Reiter, "Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition," in *Proceedings of ACM SIGSAC Conference on Computer and Communications Security*, 2016.

[12] John C Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," in *Advances in Large Margin Classifiers*. Citeseer, 1999.

[13] John Hopcroft and Ravi Kannan, "Foundations of data science," 2014.

[14] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov 1998.

[15] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[16] Stephan Zheng, Yang Song, Thomas Leung, and Ian Goodfellow, "Improving the robustness of deep neural networks via stability training," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

Appendix

Here we shall elucidate the proof of Lemma 3.1.

Proof. Consider $s_i(kz) = \frac{\exp(kz_i)}{\sum_j \exp(kz_j)}$. This can be rewritten as follows.

$$\begin{aligned} s_i(kz) &= \frac{\exp(z_i)\exp((k-1)z_i)}{\sum_j \exp(kz_j)} \\ &= \frac{\exp(z_i)}{\sum_j \frac{\exp(kz_j)}{\exp((k-1)z_i)}} \\ &= \frac{\exp(z_i)}{\sum_j \exp(kz_j - (k-1)z_i)} \end{aligned}$$

Comparing the denominator terms of the above expression and of $s_i(z)$, we arrive at the following condition for $s_i(kz) > s_i(z)$, assuming that each element z_j is independent of the others. We shall complete the proof by contradiction. For this, let us assume $s_i(kz) < s_i(z)$. This implies the following.

$$\begin{aligned} \exp(kz_j - (k-1)z_i) > \exp(z_j) \ \ \forall j \in [1, ...N] \\ \rightarrow (k-1)z_j > (k-1)z_i \end{aligned}$$

The statement above is true iff exactly one of the two conditions hold:

- $k - 1 < 0, \rightarrow k < 1$. This is false, since it is assumed that $k > 1$.
- $z_j > z_i$. This is false since z_i is assumed to be the maximum of z.

We arrive at a contradiction, which shows that the premise, i.e.; $s_i(kz) < s_i(z)$ is false.

□