SANE: Towards Improved Prediction Robustness via Stochastically Activated Network Ensembles

Ibrahim Ben Daya*, Mohammad Javad Shafiee*, Michelle Karg†,
Christian Scharfenberger†, Alexander Wong*
*University of Waterloo
Waterloo, Canada
†Continental Automotive
Germany

Abstract

A major challenge to the widespread adoption and deployment of deep neural networks in real-world operational scenarios relates to issues related to robustness and ability to deal with uncertainty when making predictions. One of the most effective strategies for improving robustness and handling uncertainty used in machine learning is the use of probabilistic modelling; however, there has been limited exploration into their use in improving the robustness of deep neural networks. In this study, we propose a new framework for improving the prediction robustness of deep neural network models via the notion of stochastically activated network ensembles (SANE), where an ensemble of deep neural networks with heterogeneous architectures are stochastically activated such that a subset of networks in the ensemble that are found to be more reliable for a given input will be responsible for a prediction. The proposed SANE framework takes advantage of a probabilistic graphical model to estimate the reliability of each network in the ensemble in predicting the correct class label for an input image given the beliefs of other networks. In other words, the graphical model enables the detection of networks in the ensemble that are likely to produce reliable predictions and include them in the final prediction process. The proposed SANE framework is evaluated on both non-targeted perturbations (e.g., random perturbations) as well as targeted perturbations (e.g., adversarial perturbations). Experimental results show that the proposed SANE framework can noticeably improve prediction robustness compared to a general ensemble approach, as well as providing further improvements in robustness against targeted perturbations when combined with additional stochastic mechanisms.

1. Introduction

Deep learning has been responsible for a number of significant breakthroughs in the field of machine learning. In particular, deep neural networks have demonstrated remarkable results in the field of computer vision for a wide variety of visual perception tasks ranging from image classification and object detection to semantic segmentation and visual concept discovery [2]. Because of this, deep neural networks are increasingly being deployed in real-world operational scenarios. With this increase in real-world deployment, particularly in safety-critical and security applications, comes the question of their overall prediction robustness as well as their ability to handle uncertainty.

Deep neural networks face difficulties in capturing uncertainty directly without additional modifications [9]. While the softmax outputs have been utilized as a confidence value of network in prediction, Gal and Ghahremani [9] illustrated that a deep neural network can be uncertain in their prediction of specific samples while still providing a high softmax output. The main approaches leveraged for addressing this limitation are Bayesian techniques, particularly the use of Bayesian neural networks [22, 5]. However, training a Bayesian neural network is not a trivial process, even with different tricks such as taking advantage of drop out [9] or modeling weights within a Gaussian process [4].

Recent literature has demonstrated that deep neural networks are very vulnerable to adversarial perturbations[27], which are malicious perturbations designed to cause networks to make erroneous predictions. Such adversarial perturbations can often be so subtle that it is imperceptible to the human eye, with an extreme case requiring only one pixel to be perturbed [25]. Furthermore, such adversarial perturbations do not necessarily require direct access to the internal mechanisms of a deep neural network, as the property of *transferability* can be leveraged where an adversarial perturbation generated using one deep neural network is used to fool another deep neural network that the attacker has no access to. All of these issues with the vulnerabilities of deep neural networks raise bigger concerns over their overall prediction robustness.

A number of different approaches have been proposed

to improve the robustness of deep neural networks, particularly to adversarial perturbations. These methods can be generally grouped into the following categories: i) using modified training/input, ii) modifying the deep neural networks themselves, and iii) using external deep neural networks as network augmentations [2]. Interestingly, it has often been shown in studies that counter-countermeasures can be devised to successfully circumvent such methods for improving the robustness of networks [6, 3]. As a result, Akhtar and Mian [2] encouraged the development of new mechanisms that can provide an estimate of the robustness of deep neural networks to obvious counter-countermeasures. A recently proposed stochastic mechanism by Xie *et al*. [29] was demonstrated to provide promising results, where the effects of adversarial perturbations may be mitigated by randomly re-sizing input images and introducing random padding. Unfortunately, this stochastic mechanism, like many proposed in the literature, lowers the accuracy on unperturbed data.

Another key strategy that have been explored for improving the robustness of deep neural networks are ensemble techniques [1, 24], where the predictions of multiple deep neural networks are leveraged together to make the final prediction. In particular, this approach was utilized recently as a mechanism for improving the robustness of deep neural networks to adversarial perturbations. The intuition behind ensemble strategies is that it is less likely for multiple deep neural networks to make the same wrong prediction for an adversarially perturbed input. In addition to improving robustness to adversarially perturbations, such ensemble techniques can also improve prediction accuracy on unperturbed data. Furthermore, ensemble techniques can reduce the bias and variance in machine learning models.

Although ensemble techniques that have been previously proposed for deep neural networks can be very helpful as a mechanism for improving robustness as well as the handling of uncertainty, the networks within the ensemble are typically aggregated with equal weighting in the final prediction step of these existing approaches. As such, while existing approaches reduce the variance of the ensemble in the final prediction process, they are also susceptible to the issue where networks within the ensemble that are less reliable (i.e., with high bias) for a given scenario will result in reduced robustness of the overall ensemble.

Inspired by the promise of ensemble techniques but with the goal of mitigating some of the current drawbacks associated with existing approaches, we propose a novel probabilistic graphical model approach to network ensembling which aggregates the predictions of networks within the ensemble in a probabilistic fashion to improve robustness, reduce system bias, while at the same time reduce variance. Probabilistic modelling is one of the most effective strategies for improving robustness and handling uncertainty used in machine learning, but there has been limited exploration into their use in improving the robustness of deep neural networks. Given that the ensemble prediction made in the proposed framework is based on a subset of robust networks that are stochastically activated within the ensemble via estimating networks reliability made by the probabilistic graphical model, we will refer to the proposed method as **stochastically activated network ensembles** (SANE). It is worth noting that, while the computational complexity of utilizing an ensemble of networks is indeed a practical challenge, the main focus of this research is the feasibility and effectiveness of utilizing a probabilistic graphical model to improve the performance of ensemble techniques at improving robustness of deep neural networks.

The rest of the paper is organized as follows. First, the proposed SANE approach is described and explained in detail in Section 2. The results of comprehensive experiments where we investigated the performance of the proposed method against state-of-the-art methods (including one of the most successful methods for improving robustness to adversarial perturbations proposed in the NIPS 2017 adversarial attacks and defenses competition [18]) in improving robustness of deep neural networks to both non-targeted perturbations as well as targeted perturbations are presented and discussed in Section 3. Finally, conclusions are drawn in Section 4.

2. Methodology

One way of defining the robustness of a deep neural network is in terms of its reliability in making correct predictions under perturbation. A robust deep neural network should be able to make a consistent prediction when presented with inputs characterizing the same entity, regardless of different perturbations being applied. Here, we explore the notions of ensemble learning and committee-based decision making for constructing network ensembles as a mechanism for improving robustness to various perturbations undergone by a scene. Intuitively, making predictions using an ensemble of networks should provide better modeling accuracy and greater robustness. However, with current approaches proposed for improving robustness leveraging equal weighting of predictions across all networks in an ensemble for the final prediction process, the networks within the ensemble that are less reliable (i.e., with high bias) for a given scenario can result in reduced robustness of the overall network ensemble. To address this issue, the proposed SANE framework introduces a probabilistic graphical model for first estimating the reliability of each network in the ensemble in predicting the correct class label for the input image given the beliefs of other networks. This measure of reliability is then leveraged to stochastically activate a subset of the networks in the ensemble for the final

prediction process. Details of each of these aspects of the proposed SANE framework are described in detail in the following sections.

2.1. Probabilistic Graphical Model

The most common way to formulating the final prediction process based on a set of deep neural networks in a network ensemble is the majority voting strategy, where the class label that is predicted by the majority of networks is leveraged as the final prediction. However, one major challenge with such a strategy for determining the final prediction of a network ensemble that must be considered when leveraged for "noisy" data under various perturbations is the fact that a majority of the networks within a network ensemble may end up being wrong in their predictions at the same time, thus leading to poor robustness in the network ensemble as a whole based on majority voting.

To mitigate this important challenge, the SANE framework introduces a probabilistic graphical model for modeling the ensemble of deep neural networks, with each node in the graph representing a network in the ensemble. The underlying graph of the probabilistic model is a fully connected graph where each node in the graph is connected to all other nodes in the graph to represent the relationship of each deep neural network to all other networks in the network ensemble.

The hidden state in the probabilistic graphical model is formulated as a binary random variable encoding the reliability of a particular network in the network ensemble for participating in the final prediction process.

More specifically, the final prediction process using the probabilistic graphical model can be expressed as a two-step procedure where,

- The reliable subset of networks are identified by marginalizing the $P(H|X)$ on each random variable (i.e., each network in the ensemble)
- The decision from the subset of reliable networks are aggregated in a weighted voting scheme to calculate the final prediction.

$P(H|X)$ is the conditional probability of the set of networks providing reliable prediction or not given the output of the Softmax layer in the networks. $H = \{h_i\}_{i=1}^{N}$ encodes the state of the networks and X represents the set of Softmax outputs (confidence values). Figure 1 demonstrates the underlying graph representation of the proposed graphical model.

The hidden state $h_i = \{0, 1\}$ is a binary random variable encoding whether the associated network is reliable or not in the current situation, with 0 encoding unreliability (i.e., the network is not reliable here for contributing to the final decision) and 1 encoding reliability (i.e., the network is reliable and should be used in the final prediction process). Variational inference is performed through the graphical model, and the networks in the network ensemble with the state $h_i = 1$ at the inference time are activated to be considered in the final prediction process of predicting the class label. Finally, the predictions of the activated networks are aggregated to provide the final prediction result.

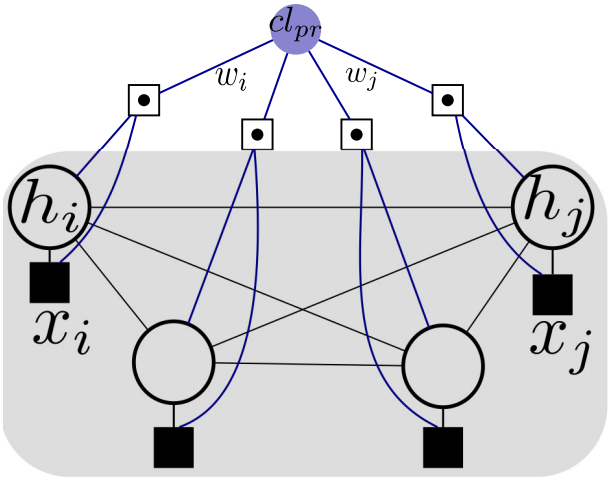

Figure 1. The proposed probabilistic graphical model in the SANE framework.

2.2. Stochastically Activated Network Ensembles

The status of each network n_i (being reliable or not) in the ensemble of networks $\mathcal{C} = \{n_1, n_2, ..., n_{|\mathcal{C}|}\}$ is encoded by h_i in the graph $\mathcal{G}(\cdot)$. Each node h_i in the graph $\mathcal{G}(\cdot)$ is associated to an observation set $\bar{x}_i \in X$ representing the set of output from the Softmax layer in the network. By formulating the ensemble of deep neural networks as a fully connected probabilistic graphical model, each network n_i in the ensemble $\mathcal{C}$ is judged by other networks $n_j, j \neq i$ in the ensemble such that the marginalized conditional probability $\sum_{h_j, j \neq i} P(H|X)$ illustrates how reliable the network n_i is in contributing to the final prediction process based on the beliefs of other networks in the graph. The conditional probability of $P(H|X)$ is formulated as a pairwise undirected graphical model:

$$P(H|X) = \frac{1}{Z} \prod_{i=1}^{|\mathcal{C}|} \phi_i(h_i, \bar{x}_i) \prod_{e=1}^{|\mathcal{E}|} \phi_e(h_{ej}, h_{ek}, \bar{x}_{ej}, \bar{x}_{ek}) \tag{1}$$

where $\phi_i(h_i, \bar{x}_i)$ is the unary potential encoding how reliable is network n_i based on prior knowledge. $\phi_e(\cdot)$ is a pairwise potential function demonstrating the belief of two end-node networks n_j and n_k of the edge $e = \{j, k\}$ on each other. $\mathcal{E}$ is the set of all edges in the graph where $|\mathcal{E}| = \frac{|\mathcal{C}| \times (|\mathcal{C}|-1)}{2}$ since the graph is fully connected.

Unary Potential. The unary potential $\phi_i(\cdot)$ encodes the prior knowledge about network n_i and how reliable it is when dealing with "noisy" data under perturbation. To formulate $\phi_i(\cdot)$ for each network n_i, here we take advantage of perturbed examples generated by all other networks (based on 1000 randomly selected images from each dataset tested), and the unreliability rate of network n_i on those perturbed examples is calculated. The goal is to find the likelihood of a wrong prediction of network n_i by a perturbed example which is misclassified by network $n_j, j \neq i$. We therefore, formulate $\phi_i(\cdot)$ as:

$$\phi_i(h_i, x_i) = \begin{cases} r_i & n_i \text{ can be fooled} \\ 1 - r_i & \text{otherwise} \end{cases} \tag{2}$$

where r_i is the transfer unreliability rate[1] with the range of $(0, 1)$ on the network n_i via the rest of networks in the ensemble.

Pairwise Potential. The role of a pairwise potential in a graphical model is to formulate the relation of two end-node random variables of an edge in the underlying graph of the conditional probability model.

The pairwise potential should encode the similarity of two-end nodes being assigned the same label (i.e., here meaning two nodes are predicting the class label correctly or the networks are not reliable in predicting the perturbed data correctly). To measure how similarly the networks behave, we take advantage of the Levenshtein distance [20] on the prediction behavior of the two networks when they are dealing with perturbed data. The Levenshtein distance is a metric for measuring the distance between two sequences of binomial data. The classification outputs of a network for a set of data are considered as a binomial sequence which encodes how the network is behaving. More specifically, a set of images, $\mathcal{I}$, is selected and perturbed. The set of perturbed images $\mathcal{I}$ is passed to each network. A sequence $\bar{s}_i$ is generated based on the predicted class label for each network n_i. For example, if we assume the networks can predict 10 different classes, the sequence $\bar{s}_i$ is comprised of values from $\{0, 1, 2, \ldots, 9\}$, such that $\bar{s}_i(k)$ shows the predicted class label for the perturbed image $I_k \in \mathcal{I}$ via the network n_i. Since the images are perturbed, each network predicts the labels differently; however, it is a fair assumption that similar networks (in terms of behaviour) would predict the same label for the same input image I_k.

To measure the similarity of two networks dealing with the perturbed images via Levenshtein distance, we use the $\bar{s}_k$ and $\bar{s}_j$ of two networks n_k and n_j with the assumption that the output sequence of labels describing how a network is behaving given a set of input. Mathematically, the Levenshtein distance of two sequences $\bar{s}_k$ and $\bar{s}_j$, is formulated as follows:

$$l_{j,k}(m, n) = \begin{cases} \max(m, n) & \text{if } \min(m, n) = 0 \\ \min \begin{cases} l_{j,k}(m-1, n) + 1 \\ l_{j,k}(n, m-1) + 1 \\ l_{j,k}(n-1, m-1) + \mathbb{1}_{(j_m \neq k_n)} \end{cases} & \text{otherwise.} \end{cases} \tag{3}$$

where $\mathbb{1}_{j_m \neq k_n}$ is an indicator function determines whether the corresponding elements m and n in the two sequences are equal or not.

The pairwise potential $\phi_e(\cdot)$ is formulated as follows:

$$\phi_e(h_{ej}, h_{ek}, \bar{x}_{ej}, \bar{x}_{ek}) = \begin{cases} \frac{l_{j,k}}{|\mathcal{I}|} \cdot \|\bar{x}_{ej} - \bar{x}_{ek}\| & h_j \neq h_k \\ (1 - \frac{l_{j,k}}{|\mathcal{I}|}) & h_j = h_k \end{cases} \tag{4}$$

where $\bar{x}_{ej}$ is the observation set for the network n_j which is the vector of confidence values corresponding to all class labels. $\mathcal{I}$ is the set of perturbed images which is used for the training purposes and computing the similarity of the networks in the training stage. The $\frac{l_{j,k}}{|\mathcal{I}|}$ is computed at the training stage and it is fixed for the network in test time.

Training. The unary and pairwise potentials have a set of parameters which need to be trained. The parameters are trained via a validation set. The parameter of the unary potential is the transferred unreliability rate (r_i) which shows how easily network n_i can make incorrect predictions on the perturbed image generated via another network. To calculate this parameter, four different perturbation levels $\epsilon = \{2, 5, 10, 20\}$ are utilized to perturb images and then the values are averaged to get r_i for network n_i.

The trainable parameters of the pairwise potential is $l_{j,k}$ encoding how similar the two networks' (n_j, n_k) behaviour is. The same validation set are utilized to calculate this set of parameters as well. The images are perturbed by the method described in section 3.3.1 and then the perturbed images are averaged together to find the final perturbed images to input to each network. The prediction output of each network for the images in the validation set are put together as a sequence and then the similarity of two networks, $l_{j,k}$ (n_j and n_k) are computed based on the Levenshtein distance between these two sequences which encodes the similarity of two networks facing perturbed images.

Inference. The underlying graph of the proposed probabilistic model is a small graph and as a result, it is possible to perform variational inference [28] algorithms efficiently. Here we leverage the message passing algorithm [28] through the designed probabilistic graphical model of the ensemble of the networks to determine the set of unreliable networks. The marginal probability of each network n_i (node in the graph) being unreliable can be computed easily based on the belief of other nodes (networks)

[1]The transfer unreliability rate r_i is defined as the ratio of the number of perturbed examples (generated by other networks) that can lead network n_i to making an incorrect prediction over the total number of perturbed examples.

in the graph passed to the node (network) n_i. After the message passing is complete, it is possible to compute the marginal probability for the network n_i as

$$P(h_i) = \sum_{h_j=\{0,1\}, j\neq i} P(h_i, H_{j\neq i}|X) \quad (5)$$

which $P(h_i = 0) > 0.5$ illustrates the network could not classify the input correctly and it is not reliable based on the belief of the ensemble of all other networks regarding to the network n_i.

3. Results and Discussion

The robustness of the proposed SANE framework and state-of-the-art algorithms for improving robustness of deep neural networks are evaluated through a series of experiments. We will detail our experimental setup for evaluating the efficacy of the proposed SANE framework, present quantitative results for our experiments related to robustness of different deep neural network architectures and the proposed SANE framework under both targeted and non-targeted perturbation scenarios. First, we examine the behaviour of deep neural networks in non-targeted perturbation scenarios to analyze the robustness of different models in general cases and compare this behavior with targeted perturbation scenarios. Next, we evaluate the efficacy of the proposed SANE framework and other tested methods for improving robustness under targeted perturbation scenarios.

3.1. Experimental Setup

The effectiveness of the proposed SANE framework is examined quantitatively via two different datasets: i) CIFAR-10 dataset, and ii) NIPS adversarial attack challenge dataset. The CIFAR-10 dataset [16] is comprised of 32×32 images of 10 different natural image classes. The NIPS adversarial attack challenge [18] dataset is a set of larger image sizes with 1000 different class of natural images derived from the ImageNet dataset [8]. For comparative analysis purposes, we also compare the performance of the proposed SANE framework against RandDef [29], one of the most successful methods for improving robustness to adversarial perturbations proposed in the NIPS 2017 adversarial attacks and defenses competition [18].

3.1.1 Networks Description

The network ensembles used in this study are comprised of 10 different deep convolutional neural network architectures for each dataset. A variety of networks with both simple to complex architectures were selected to create the ensemble of networks. Below is the list of networks selected for the CIFAR-10 and NIPS challenge experiments. Each network is assigned a reference number, which will be used when reporting the results in the following sections.

- **CIFAR-10 Dataset:** AlexNet (**N1**) [17], GoogleNet (**N2**) [26], Lenet5 (**N3**) [19], NIN (**N4**) [21], ResNet18 (**N5**) [12], SqueezeNet (**N6**) [15], VGG16 (**N7**) [23], ResNet50 (**N8**) [12], SimpNet (**N9**) [11], and DenseNet40 (**N10**) [14].
- **NIPS Challenge Dataset:** DenseNet161 (**N1**) [14], AlexNet (**N2**) [17], VGG16 (**N3**) [23], ResNet18 (**N4**) [12], ResNet50 (**N5**) [12], SqueezeNet (**N6**) [15], DualPathNet92 (**N7**) [7], ResNext101 (**N8**) [29], NASNet-Mobile (**N9**) [30], and Squeeze and Excitiation ResNet50 (**N10**) [13].

3.2. Robustness to Non-targeted Perturbations

As the first experiment, we analyze how different network architectures perform in non-targeted perturbation scenarios. We evaluate 10 different networks introduced in the previous section on the NIPS challenge dataset under different levels of Gaussian-distributed random perturbations. All networks can make correct predictions about the tested samples before they were perturbed. Figure 2 demonstrate the accuracy of different models under different levels of random perturbation.

Results show that while DualPathNet92, DenseNet191, RexNext101 and Squeeze and Excitation ResNet50 are the most robust networks in dealing with lower levels of perturbation, RexNext101 is the most robust network when the level of perturbation increases. On the other extreme, SqueezeNet architecture demonstrates the poorest robustness in dealing with random perturbations and after that, VGG, AlexNet and even ResNet50 provide the lowest robustness to random perturbations. To justify this behaviour, one reason can be the architecture complexity of the more robust networks compared to others; as the generally more complex networks are showing more robustness to the Gaussian-distributed random perturbations.

The proposed SANE framework is also examined under the non-targeted perturbations. Results demonstrate that SANE can provide noticeably greater robustness when compared to each network independently. The committee of the networks ensemble is the combination of 10 different networks introduced in the previous section. As seen in Figure 2, some of the networks are performing very poorly under the random perturbations, which suggests that having more robust networks in the committee would result in greater robustness for SANE.

3.3. Robustness to Targeted Perturbations

Here, we examine the efficacy of the proposed SANE framework in improving robustness to targeted perturba-

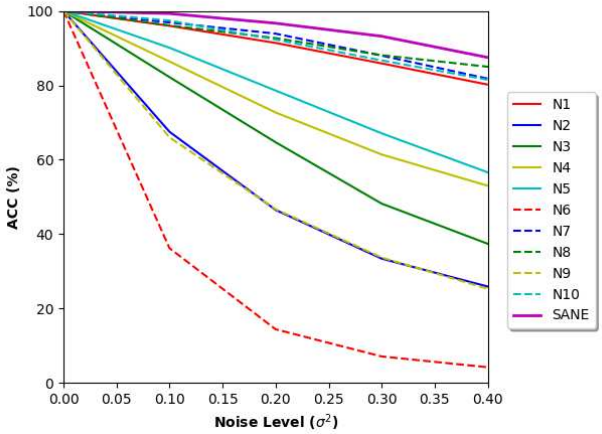

Figure 2. The robustness of different network architectures in dealing with non-targeted perturbation (in this study, Gaussian-distributed random perturbations). The input images (NIPS challenge dataset) are perturbed at different levels and are passed to the networks for making the classification prediction. Results demonstrate that the smaller networks are more prone to the random perturbations when compared to more complex architectures.

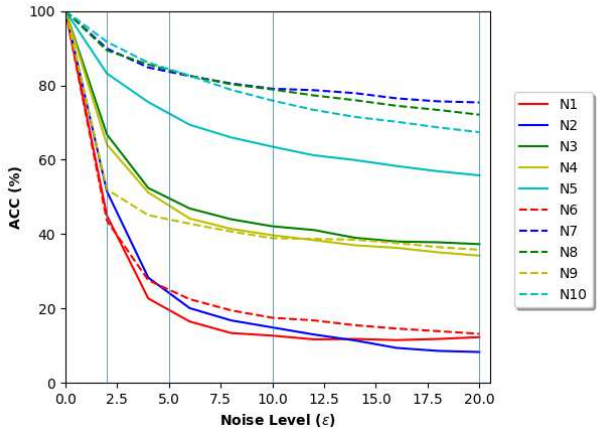

Figure 3. The robustness of different network architectures in dealing with targeted perturbations generated by FGSM method. Results show that the more robust networks seen in non-targeted perturbation experiments are among the most robust networks when dealing with targeted perturbations as well. The vertical blue lines show the specific perturbation levels used for the experiments in the next sections.

tions. Here, we will examine targeted perturbations in the form of adversarial perturbations. These types of perturbations are specifically designed to be small with respect to the image while greatly affecting the robustness of the deep neural network in making correct predictions. This particular property of adversarial perturbation makes it very well-suited for illustrating the robustness of a deep neural network, since such subtle nature of the perturbation makes them particularly devastating in real-world operational scenarios since they are designed to go visually undetected.

3.3.1 Targeted Perturbation Generation

We use fast gradient sign method (FGSM) attack to generate targeted perturbations in order to examine the robustness of deep neural networks and our proposed framework. FGSM [10] is a single-step white-box adversarial perturbation that uses the loss of the network as a perturbation to input x:

$$x^{adv} = x + \epsilon \cdot sign\Big(\nabla_x L\big(f(x;\theta), y\big)\Big) \tag{6}$$

where x^{adv} is the adversarial image, x the original image, ϵ is a small scalar that restricts the norm of the perturbation, sign$(\cdot)$ is the sign function, $\nabla f(\cdot)$ computes the gradient of the loss function $L(\cdot)$ between the network prediction y under model parameters θ.

In order to provide a targeted perturbed input to the full network ensemble, FGSM perturbation for each network in the ensemble is first found, and a universal perturbation for the network ensemble is found following a similar averaging approach used in Strauss *et al*. [24].

3.3.2 Evaluation Methods

To have a more comprehensive evaluation, the proposed SANE framework is examined with two test set of 1000 images – one from CIFAR-10 dataset, the other from NIPS adversarial attack challenge dataset. The images are randomly selected from the set of all images correctly classified by all networks in the ensemble, which is consistent with evaluation methodologies in existing literature. This is a separate set from the one used in the formulation of $\phi_i(.)$ in equation 2. The proposed SANE framework is examined under four different perturbation levels of $\epsilon = \{2, 5, 10, 20\}$ generated by FGSM and compared with two other methods, as well as combinations of these two:

- **EnsembleDef [24]:** This technique involves using a network ensemble for improving robustness of deep neural networks, where a general voting mechanism is leveraged with all networks having equal contribution in the decision making process.

- **RandDef [29]**: This technique involves randomly resizing and padding the input before being fed into the network to improve the robustness of the network by reducing the perturbation level of the input image. It was demonstrated to be one of the best mechanisms to improve robustness of networks in the NIPS 2017 adversarial attacks and defenses competition [18], even

Table 1. CIFAR-10 transferability rate experiment. This Table shows the robustness of each network when given a targeted perturbation generated from other networks. Columns represent networks used to generate the perturbed data, rows show the success rate (misclassification error) of these perturbed data on each network. The diagonal values demonstrate the success rate for targeted perturbation for on leading a network to make an incorrect prediction.

	N1	**N2**	**N3**	**N4**	**N5**	**N6**	**N7**	**N8**	**N9**	**N10**
N1	0.678	0.129	0.111	0.141	0.037	0.050	0.082	0.043	0.040	0.052
N2	0.341	0.829	0.140	0.133	0.076	0.076	0.098	0.078	0.073	0.077
N3	0.305	0.181	0.858	0.240	0.219	0.162	0.133	0.239	0.227	0.245
N4	0.289	0.109	0.197	0.478	0.123	0.093	0.064	0.136	0.121	0.148
N5	0.307	0.293	0.448	0.325	0.886	0.316	0.264	0.589	0.499	0.547
N6	0.303	0.159	0.287	0.232	0.252	0.341	0.120	0.286	0.236	0.265
N7	0.295	0.160	0.162	0.141	0.127	0.108	0.799	0.124	0.120	0.130
N8	0.251	0.237	0.384	0.258	0.445	0.237	0.230	0.856	0.463	0.467
N9	0.166	0.161	0.287	0.179	0.325	0.163	0.143	0.414	0.658	0.386
N10	0.224	0.110	0.344	0.228	0.442	0.202	0.183	0.502	0.467	0.732

Table 2. NIPS adversarial attack challenge transferability rate experiment. The same experiment as Table 1 is conducted for the ImageNet dataset. It is interesting that the networks that showed robustness against targeted perturbation (generated by FGSM) for CIFAR-10 dataset are not the robust network when the training data is NIPS challenge dataset. Therefore, it is hard to claim a single network can be robust against different forms of perturbations and different datasets.

	N1	**N2**	**N3**	**N4**	**N5**	**N6**	**N7**	**N8**	**N9**	**N10**
N1	0.805	0.009	0.032	0.050	0.065	0.010	0.041	0.064	0.013	0.030
N2	0.049	0.844	0.051	0.053	0.043	0.057	0.039	0.041	0.047	0.047
N3	0.067	0.036	0.841	0.084	0.065	0.036	0.039	0.040	0.032	0.048
N4	0.078	0.035	0.070	0.902	0.082	0.041	0.045	0.052	0.028	0.052
N5	0.092	0.013	0.051	0.081	0.771	0.018	0.051	0.071	0.024	0.041
N6	0.082	0.099	0.095	0.109	0.078	0.936	0.068	0.060	0.062	0.076
N7	0.060	0.011	0.020	0.032	0.043	0.009	0.652	0.067	0.010	0.027
N8	0.075	0.009	0.026	0.042	0.063	0.011	0.059	0.694	0.013	0.032
N9	0.103	0.089	0.108	0.119	0.010	0.094	0.088	0.089	0.736	0.087
N10	0.045	0.011	0.027	0.037	0.037	0.012	0.030	0.032	0.014	0.525

when the targeted perturbation is generated by prior knowledge about the prediction mechanism.

3.4. Results

We first analyze the effect of targeted perturbation on different networks and compare the results with when the perturbation is a non-targeted as described in Section 3.2. Figure 3 demonstrates the robustness of each network in dealing with different targeted perturbations with different levels of perturbations, as measured by the modeling accuracy. As seen the more robust networks identified during the non-targeted perturbation experiments are also the most robust networks when dealing with targeted perturbations. However, the least robust network is not Squeezenet here, which was the case in non-targeted perturbations, and in fact can provide better robustness compared to some other tested deep neural networks.

The performance results of the proposed SANE framework are compared with two different methods to address targeted perturbations as well (and combination of the two): i) EnsembleDef [24], and ii) RandDef [29], and iii) RandDef + EnsembleDef. We also provide the accuracy of the best single network on the original input image (Single-Best Network) under targeted perturbation as a reference point. Results show that it is possible to incorporate the random resizing and padding approach as pre-processing and combine it with SANE to improve the robustness of prediction in dealing with targeted perturbations.

3.4.1 Perturbation Transferability

To examine the effect of each network in the ensemble of the networks on other networks, as the first experiment the perturbation transferability of each network on another network via the two mentioned datasets are experimented. A set of 1000 selected images for the validation purposes are utilized to generate targeted perturbed images via each network in the ensemble and then the generated perturbed images (via each network) are utilized to examine other networks in the ensemble. It is worth to mention that the FGSM approach is utilized to generated the perturbed images for this experiment.

Table 1 demonstrates the perturbation transferability of each network on the rest of the networks in the ensemble based on the CIFAR-10 dataset. Each column in the Table 1 shows the success rate of the targeted perturbed images by the network model specified in the header of the column to make other networks misclassify the sample. Moreover, the diagonal values in the Table specify the error rate of the networks. The interesting observation is that the network models NIN (N4) and SqueezeNet (N6) are more robust to targeted perturbations compared to other networks. On the other hand, GoogleNet (N2), SqueezeNet (N6) and VGG16 (N7) have the lowest misclassification when dealing with targeted perturbations.

Table 2 shows the same experiment for the ImageNet dataset. DualPathNet (N7) and ResNet101 (N10) are the most robust networks in the set while SqueezeNet (N6) and NasNet-MobileNet (N9) are the weakest network in generating targeted perturbed images to cause other networks to make incorrect predictions. It is also interesting that although SqueezeNet showed the best robustness against targeted perturbation for the CIFAR-10 dataset, it has one of the lowest performances for the ImageNet dataset which shows consistent results with non-targeted perturbations.

Based on these observations, it can be seen that it is difficult to design a general network architecture that can be robust against all forms of perturbations for all different applications and it has been shown that in each application there is a specific network architecture that provides the best robustness against the perturbation. Due to this fact, having ensemble of networks for making predictions would help

Table 3. Accuracy of the proposed SANE framework compared to other mechanisms based on the CIFAR-10 dataset. The proposed SANE framework outperforms both EnsembleDef and RandDef, and achieves comparable results to when both EnsembleDef and RandDef are combined together. Furthermore, the combination of RandDef and SANE outperforms all other tested methods.

Noise Level	Single-Best Network	RandDef [29]	EnsembleDef [24]	RandDef + EnsembleDef	SANE	RandDef + SANE
FGSM ($\epsilon = 2.0$)	74.2%	52.4%	99.3%	74.0%	99.3%	72.0%
FGSM ($\epsilon = 5.0$)	66.0%	49.6%	93.6%	75.0%	96.2%	79.0%
FGSM ($\epsilon = 10$)	62.0%	46.0%	70.7%	64.0%	78.2%	57.0%
FGSM ($\epsilon = 20$)	53.7%	41.5%	43.7%	68.0%	50.3%	61.0%

Table 4. Accuracy of the proposed SANE framework compared to other methods based on the NIPS adversarial attack dataset. Results consistently show that not only does SANE outperform both RandDef and EnsembleDef, but the combination of RandDef and SANE can provide the best performance against the targeted perturbations.

Noise Level	Single-Best Network	RandDef [29]	EnsembleDef [24]	RandDef + EnsembleDef	SANE	RandDef + SANE
FGSM ($\epsilon = 2.0$)	70.4%	90.0%	99.5%	100.0%	99.6%	99.8%
FGSM ($\epsilon = 5.0$)	53.4%	70.9%	96.8%	98.2%	97.1%	98.4%
FGSM ($\epsilon = 10$)	43.3%	62.3%	89.4%	92.6%	91.3%	92.5%
FGSM ($\epsilon = 20$)	39.7%	55.3%	79.2%	83.3%	82.9%	85.3%

greatly to improve robustness.

3.4.2 Robustness Against Targeted Perturbation

In this section we analyze the robustness of the proposed SANE framework against targeted perturbation (i.e., generated by FGSM approach). Table 3 shows the experimental results for the CIFAR-10 dataset. The results show that the proposed SANE framework outperforms both RandDef and the best performing network in the ensemble across all perturbation levels. Furthermore, SANE provides similar performance as EnsembleDef for $\epsilon = 2$, but outperforms EnsembleDef for all noise levels above that. This is most illustrative by the reported result for $\epsilon = 10$ and $\epsilon = 20$ where SANE can achieve 8% and 7% higher accuracy, respectively, when compared to EnsembleDef.

The proposed SANE framework is also compared to the combination of RandDef and EnsembleDef (i.e., RandDef+EnsembleDef) as well. Furthermore, we also experimented with the combination of RandDef and SANE (i.e., RandDef+SANE). Results demonstrate that RandDef could not improve robustness when used in conjunction with EnsembleDef or SANE in this case. The poor performance of RandDef can be justified by the fact that since CIFAR-10 images are small (32×32), randomly resizing and padding them reduces the amount of information in the image and thus causes a drop in modeling accuracy.

Table 4 demonstrates the experimental results for all tested methods for the ImageNet trained models. The results show a very similar trend as those observed in the CIFAR-10 experiments, with SANE outperforming RandDef across all perturbation levels. Similarly, SANE achieved similar accuracy as EnsembleDef at $\epsilon = 2$, but outperforms EnsembleDef significantly at higher perturbation levels. What differs from the CIFAR-10 experimental observations is the fact that here, in the ImageNet experiments, RandDef performs noticeably better than the performance of the single-best network, which illustrates its effectiveness for improving robustness in the situation where the image size is sufficiently large. Finally, it is observed that the combination of RandDef with SANE (i.e., RandDef+SANE) provides additional robustness over SANE, especially at the highest perturbation levels, leading RandDef+SANE to provide the highest robustness to targeted perturbations out of all tested methods.

4. Conclusion

In this study, we proposed a new probabilistic approach to improve the robustness of ensembles of deep neural networks. In the proposed stochastically activated network ensembles (SANE) framework, a subset of reliable deep neural networks in the ensemble are determined and activated based on a fully-connected probabilistic graphical model for the final prediction process. Experimental results using CIFAR-10 and ImageNet demonstrated the effectiveness of the proposed SANE framework at improving robustness in prediction when compared to other state-of-the-art frameworks for improving robustness. In addition, we showed that it is possible to combine SANE with other mechanisms to further improve robustness to in targeted perturbation scenarios. Future work will focus on a more efficient way to leverage the proposed SANE framework for practical applications where computational constraints is limited.

References

[1] M. Abbasi and C. Gagné. Robustness to adversarial examples through an ensemble of specialists. *CoRR*, abs/1702.06856, 2017. 2

[2] N. Akhtar and A. Mian. Threat of adversarial attacks on deep learning in computer vision: A survey. *CoRR*, abs/1801.00553, 2018. 1, 2

[3] A. Athalye and N. Carlini. On the robustness of the CVPR 2018 white-box adversarial example defenses. *CoRR*, abs/1804.03286, 2018. 2

[4] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra. Weight uncertainty in neural networks. *arXiv preprint arXiv:1505.05424*, 2015. 1

[5] W. L. Buntine and A. S. Weigend. Bayesian backpropagation. *Complex systems*, 5(6):603–643, 1991. 1

[6] N. Carlini and D. A. Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. *CoRR*, abs/1705.07263, 2017. 2

[7] Y. Chen, J. Li, H. Xiao, X. Jin, S. Yan, and J. Feng. Dual path networks. In *Advances in Neural Information Processing Systems*, pages 4467–4475, 2017. 5

[8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009. 5

[9] Y. Gal and Z. Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059, 2016. 1

[10] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. corr (2015). 6

[11] S. H. Hasanpour, M. Rouhani, M. Fayyaz, M. Sabokrou, and E. Adeli. Towards principled design of deep convolutional networks: Introducing simpnet. *arXiv preprint arXiv:1802.06205*, 2018. 5

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5

[13] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. *arXiv preprint arXiv:1709.01507*, 7, 2017. 5

[14] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *CVPR*, volume 1, page 3, 2017. 5

[15] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016. 5

[16] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009. 5

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 5

[18] A. Kurakin, I. J. Goodfellow, S. Bengio, Y. Dong, F. Liao, M. Liang, T. Pang, J. Zhu, X. Hu, C. Xie, J. Wang, Z. Zhang, Z. Ren, A. L. Yuille, S. Huang, Y. Zhao, Y. Zhao, Z. Han, J. Long, Y. Berdibekov, T. Akiba, S. Tokui, and M. Abe. Adversarial attacks and defences competition. *CoRR*, abs/1804.00097, 2018. 2, 5, 6

[19] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 5

[20] V. I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710, 1966. 4

[21] M. Lin, Q. Chen, and S. Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013. 5

[22] D. J. MacKay. A practical bayesian framework for backpropagation networks. *Neural computation*, 4(3):448–472, 1992. 1

[23] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 5

[24] T. Strauss, M. Hanselmann, A. Junginger, and H. Ulmer. Ensemble methods as a defense to adversarial perturbations against deep neural networks. *arXiv preprint arXiv:1709.03423*, 2017. 2, 6, 7, 8

[25] J. Su, D. V. Vargas, and K. Sakurai. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, 2019. 1

[26] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 5

[27] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *CoRR*, abs/1312.6199, 2013. 1

[28] M. J. Wainwright, M. I. Jordan, et al. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1–2):1–305, 2008. 4

[29] C. Xie, J. Wang, Z. Zhang, Z. Ren, and A. L. Yuille. Mitigating adversarial effects through randomization. *CoRR*, abs/1711.01991, 2017. 2, 5, 6, 7, 8

[30] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le. Learning transferable architectures for scalable image recognition. *arXiv preprint arXiv:1707.07012*, 2(6), 2017. 5