Towards Robust Image Classification Using Sequential Attention Models

Daniel Zoran  Mike Chrzanowski  Po-Sen Huang  Sven Gowal  Alex Mott  
Pushmeet Kohli  
DeepMind  
London, UK  
danielzoran@google.com

Abstract

In this paper we propose to augment a modern neural-network architecture with an attention model inspired by human perception. Specifically, we adversarially train and analyze a neural model incorporating a human inspired, visual attention component that is guided by a recurrent top-down sequential process. Our experimental evaluation uncovers several notable findings about the robustness and behavior of this new model. First, introducing attention to the model significantly improves adversarial robustness resulting in state-of-the-art ImageNet accuracies under a wide range of random targeted attack strengths. Second, we show that by varying the number of attention steps (glances/fixations) for which the model is unrolled, we are able to make its defense capabilities stronger, even in light of stronger attacks — resulting in a "computational race" between the attacker and the defender. Finally, we show that some of the adversarial examples generated by attacking our model are quite different from conventional adversarial examples — they contain global, salient and spatially coherent structures coming from the target class that would be recognizable even to a human, and work by distracting the attention of the model away from the main object in the original image.

1. Introduction

Recent years have seen great advances in the use and application of deep neural network models. From large scale image classification [20] to speech recognition [24], the performance of these models has steadily improved, making use of new hardware advances, more memory and better optimization strategies. The leading model paradigm for such tasks, however, has not changed significantly since the original AlexNet paper [30]. Models are still, predominately, built in a purely feed-forward manner, interleaving convolutional layers (often with small kernels with limited support) and simple non-linearities [39]. Recently introduced ResNets [21], which are some of the most powerful models we use currently, have not changed this scenery significantly.

While there is no doubt that these models are very successful in solving some tasks, concerns have been raised about their robustness and reliability [35, 48]. Small, carefully chosen perturbations to the input, often imperceptible to a human observer, may cause these models to output incorrect predictions with high confidence [48]. These kind of perturbations are called adversarial examples [18, 48] and are a subject of ongoing research [4, 9, 55].

The current paradigm of neural network models has certainly been inspired by the human and primate visual system [43]. Early predecessor models have directly made this connection, and there is a line of work which connects between the activations of such neural network models and the neural activity in brains [8]. These parallels between models...
and biological vision systems apply mostly to early vision processing \[14\] – and specifically, the feed-forward processing which happens in time-limited scenarios \[14\]. This has been discussed in several intriguing works, including in the adversarial examples context.

There are, however, some major differences between feed-forward neural network and the primate visual system. The eye in primates has a fovea which samples different regions of the visual input field at different spatial resolutions \[16\]. Furthermore (and possibly tightly connected to the fovea) the system has a strong attentional bottleneck which has been researched in many different works \[44, 7\]. The visual cortex has many feedback and top-down recurrent connections \[40\] and it is not purely feed-forward. Additionally, humans don’t view images as a static scene, but explore the images in a series of saccades/fixations, collecting and integrating information in the process \[33\]. This has been postulated to cause humans to report different classification mistakes which are qualitatively different than those of deep neural networks \[13\].

In this work we propose to use a soft, sequential, spatial, top-down attention mechanism (which we abbreviate as S3TA) \[38\], drawing inspiration from the primate visual system. While we do not presume this to be a biologically-plausible model in any way, we do propose that this model captures some of the functionality of the visual cortex, namely the attentional bottleneck and sequential, top-down control. We adversarially train the model on ImageNet images, showing that it has state-of-the-art robustness against adversarial attacks (focusing on Projected Gradient Descent or PGD \[31, 35\] attacks). We show that by increasing the number of steps we unroll the model, we are able to better defend against stronger attacks – resulting in a “computational race” between the attacker and the defender. Finally, but importantly, we show that the resulting adversarial examples often (though not always) include global, salient structures which would be perceptible and interpretable by humans (Figure 1). Furthermore, we show that the attack often tries to attract the attention of the model to different parts of the image instead of perturbing the main object in the source image directly.

2. Related Work

Adversarial training: Adversarial training aims to create models that are robust to adversarial attacks. At their core, techniques such as the ones of \[18\] and \[35\] find the worst case adversarial examples at each training step (using Fast Gradient Sign Method or PGD attacks) and add them to the training data. Models created by \[35\] have been shown to be empirically robust on MNIST and CIFAR-10. \[27\] proposed using Adversarial Logit Pairing (ALP) to encourage the logit predictions of a network for a clean image and its adversarial counterpart to be similar. However, ALP performs poorly under stronger attacks \[15\]. \[55\] proposed feature denoising networks together with adversarial training to achieve strong performance on ImageNet \[12\]. Other methods like \[19\] achieve gradient obfuscation more explicitly by adding non-differentiable preprocessing steps. Although these gradient masking techniques make gradient-based attacks fail, more sophisticated adversaries, such as gradient-free methods \[50, 2\], can circumvent these defenses.

Recurrent attention models: Attention mechanisms have been widely used in many sequence modeling problems such as question-answering \[23\], machine translation \[6, 51\], video classification and captioning \[45, 32\], image classification and captioning \[36, 11, 17, 1, 54, 59, 52, 5, 56\], text classification \[57, 46\], generative models \[41, 58, 29\], object tracking \[28\], and reinforcement learning \[10\]. We build out model based on the one introduced in \[38\] and adapt and modify it for ImageNet scale image classification. The model uses a soft key, query, and value type of attention which have been discussed in several intriguing works, including in the focusing which happens in time-limited scenarios \[33\]. This has been postulated to cause humans to report different classification mistakes which are qualitatively different than those of deep neural networks \[13\].

3. Model

We base our model on the one proposed by \[38\] for reinforcement learning and we adapt it to ImageNet scale image classification. The model sequentially queries the input, actively attending relevant pieces of spatial information at each time step to refine its estimate of the correct label. The two key components are the sequential nature of the model and the top-down, attentional bottleneck, both of which we empirically show contribute to its resilience to adversarial attacks.
We briefly highlight the important components of the model, illustrated in Figure 2. For full details, we refer the reader to [38] and the supplementary material. The model starts by passing the input image through a "vision" net — a convolutional neural net (here we use a modified ResNet-152, see below). We use the same input image for all time steps, so the output of the ResNet needs to be calculated only once. The resulting output tensor is then split along the channel dimension to produce a keys tensor and a values tensor. To both of these tensors, we concatenate a fixed spatial basis tensor which encodes spatial locations using a Fourier representation. This spatial basis is important because our attentional bottleneck sums over space causing the spatial structure of these tensors to disappear and this basis allows passing on spatial location information.

We unroll the top-down controller for several compute steps, attending the input at each step and processing the answer through the controller to produce the output (or next state). The Top-Down controller is an LSTM core [25] whose previous state is decoded through an “query network”, an MLP, into one or more query vectors. Each query vector has the same number of channels as the keys tensor plus the number of channels in the spatial basis. We take the inner product between the query vector and key and spatial basis tensor at each spatial location, resulting in a single channel map of attention logits. We pass these through a spatial softmax to produce the attention map for this query. The resulting attention map is then point-wise multiplied with the values tensor (and the spatial basis). Note that a single map is used for all channels, we note the importance of this below. The multiplied value tensor is summed across the spatial dimensions to produce an answer vector, one for each query. These answers are fed into the LSTM as the input for the current time step (concatenating them if more than one is used). Finally, the output from the last LSTM output is decoded into class logits through an MLP. The cross-entropy loss w.r.t to the ground truth class is calculated with this output. The initial state of the LSTM is also learned. Since the model is fully differentiable, we train it end-to-end, including the ResNet, with adversarial training (as described in Section 4.1) and 5.

Several important points regarding our version of the model in this context:

- The attention bottleneck makes the decision of the model depend on potentially large extents of the image. This can be due to the shape of attention map at every time step, as well as the fact that these maps can change considerably between time steps. This should cause a local adversarial perturbation [37] to be less effective. We discuss this in Section 6 and show that indeed, we often observe that global perturbations are required by the attacker for the attack to succeed.

- Following the last point, the fact that the attention map has a single channel which modulates all value channels together constrains the content of these channels to be spatially coherent. In a regular ResNet architecture the last block output is read with a average pooling done independently on each channel - this allows the network to lose spatial structure by the time information reaches this last layer.

- In order to make the spatial element, and hence the effect of the attention bottleneck more pronounced we modify the ResNet architecture to make the final output have larger spatial dimensions. This is done by changing the strides to 1 in all but the second residual block. For ImageNet input (224 × 224 pixels) the resulting map is 28 × 28 pixels large (as opposed to 7 × 7 in a regular ResNet).

- The top-down nature of the attention mechanism is such that the queries come from the state of the LSTM and not from the input. Hence, the model can actively select relevant information depending on its internal state, rather than just the input. This allows the model to take its own uncertainty, for example, into account, when querying the image and producing the output.

- The sequential nature of the model allows for increasing computational capacity without changing the number...
of parameters. We demonstrate that this helps with robustness in Section 5.

4. Adversarial Risk

We define adversarial risk in the context of supervised learning formally in this section. Given a model \( m_\theta \) with parameters \( \theta \), we want to minimize the loss \( \ell \) on inputs \( x \) and labels \( y \) sampled from the data distribution \( D \). Formally, the objective is to minimize the expected risk: 
\[
\mathbb{E}_{(x,y) \sim D} \ell(m_\theta(x), y).
\]
Empirically, we optimize the empirical risk on a finite training set and estimate the expected risk over a held-out test set using the average loss.

As pointed out in [50], models with low expected risk may still perform poorly on any data points. In situations where a single catastrophic failure is not allowable, the empirical risk estimate may be problematic. Hence, we also need to consider the worst-case risk for the desired robust models: 
\[
\sup_{(x,y) \in \text{supp } D} \ell(m_\theta(x), y),
\]
where \( \text{supp } D \) denotes the support of \( D \). In practice, computing the supremum over the input space is intractable as the search space is exponentially large in the dimension of \( x \). We can instead use the local adversarial risk, as a proxy for the worst-case risk:
\[
\mathbb{E}_{(x,y) \sim D} \left[ \sup_{x' \in N_r(x)} \ell(m_\theta(x'), y) \right],  \tag{1}
\]
where the neighborhood \( N_r(x) \) denotes a set of points in \( \text{supp } D \) within a fixed distance \( \epsilon > 0 \) of \( x \), measured by a given metric. The adversarial risk enables us to approximate the worst-case risk in a tractable way. For example, we can use off-the-shelf optimization algorithms (such as PGD [31, 35]) to find the supremum over the neighborhood \( N_r(x) \).

In this paper, we consider the specification that the image predictions should remain the same within an \( \ell_\infty \)-ball of an image \( x \), where an allowable maximum perturbation is \( \epsilon = 16/255 \), relative to the pixel intensity scaled between 0 and 1.

Specifically, we focus on the ImageNet dataset [12] and we primarily consider the targeted PGD attack as the threat model, where the targeted class is selected uniformly at random, following [2, 27, 55], given that the untargeted attacks can result in less meaningful comparisons (e.g., misclassification of very similar dog breeds) on ImageNet.

4.1. Adversarial Training

To train models that are robust to adversarial attacks, we follow the adversarial training approach by [35] and more recently [55].

Following the adversarial risk in Eq. (1), we want to minimize the following saddle point problem:
\[
\min_\theta \rho(\theta), \quad \text{where} \quad \rho(\theta) = \mathbb{E}_{(x,y) \sim D} \left[ \max_{x' \in N_r(x)} \ell(m_\theta(x'), y) \right].  \tag{2}
\]
where the inner maximization problem is to find an adversarial perturbation of \( x \) that can maximize the loss; the outer minimization problem aims to update model parameters such that the adversarial risk \( \rho(\theta) \) is minimized.

In our experiments, we approximate the solution to the inner maximization problem with PGD. Specifically, we perform PGD on the cross entropy loss described using iterative signed gradients as in [31, 55]. During training, we use targeted PGD attacks, where the targeted class is selected at random uniformly, following [27, 55].

4.2. Adversarial Evaluation

In this paper, we use the PGD attack to evaluate the model, which is regarded as a strong attack\(^1\) in the community and several published papers use this as their benchmark.

In cases where we cannot take analytic gradients, or where they are not useful, we can approximate the gradients using gradient-free optimization. The use of gradient-free methods lets us verify whether robustness stems from gradient obfuscation by the model architecture. In this work, we use the SPSA algorithm [47], which is well-suited for high-dimensional optimization problems, even in the case of noisy objectives. We use the SPSA formulation in [50] to generate adversarial attacks. In the SPSA algorithm, it first samples a batch of \( n \) samples from a Rademacher distribution (i.e., Bernoulli \( \pm 1 \)), namely, \( v_1, \ldots, v_n \in \{1, -1\}^D \). Then, the SPSA algorithm approximates the gradient with finite difference estimates in random directions. Specifically, for the \( i \)-th sample, the estimated gradient \( g_i \) is calculated as follows:
\[
g_i = \frac{f(x_t + \delta v_i) - f(x_t - \delta v_i)}{2\delta v_i}, \tag{3}
\]
where \( \delta \) is the perturbation size, \( x_t \) is the perturbed image at the \( t \)-th iteration, and \( f \) is the model to be evaluated. Finally, SPSA aggregates the estimated gradient and performs projected gradient descent on the input \( x_t \). The whole process iterates for a predefined number of iterations.

5. Experiments

In this section, we empirically study the robustness of S3TA on the ImageNet dataset [12]. For convenience, we denote S3TA-\( k \) to be a S3TA model that is unrolled for \( k \) time steps and evaluate S3TA-2, S3TA-4, S3TA-8, and S3TA-16.

We follow the training procedure used in [55], including learning rate schedule, label smoothing, attack type during training and evaluation procedure. We find that training with a somewhat lower learning rate (0.2 initial learning rate) and a smaller batch size (1024) is more stable for our model.

---

\(^{1}\)Note that PGD is not necessarily the most suited attack for sequential models, but for lack of better alternative we use it with large number of steps.
Training S3TA-16 is more challenging than the other models due to the length of the unroll. In order to train it we start by reading off the output from the 4th step for the first 35 epochs, 8th step for the next 35 epochs and 16th step for the rest of training. All models are trained for 120 epochs. We train the model on 128 Google Cloud TPU v3 cores. Training takes between 42 and 70 hours, depending on the number of unroll steps. We use a ResNet-152 as the vision-net of the model (see Section 3) setting all strides to 1 other than the second residual block. This results in larger spatial support for the ResNet output (28 × 28 pixels) The recurrent core is an LSTM with 1024 hidden units, the query network and the output MLP are both with a single hidden layer of 1024 units. All the activation units used are ReLUs. The attention model uses 4 attention query heads in all experiments here.

5.1. Random targeted attacks

The first set of models was adversarially trained with 10 PGD steps. These are generally weaker models than models trained with 30 PGD steps (see below) but they take less time and resources to train. Figure 3 shows the top-1 accuracy of these models for the ImageNet test dataset under a wide range of random targeted PGD attack strengths compared to a ResNet-152 baseline (also trained with 10 PGD steps during adversarial training). With only 2 steps of attention the weakest model here, S3TA-2, only has a chance to send two queries, one before it even sees the image, and one after processing the answers from the first step. This puts emphasis on the attention bottleneck itself rather than the sequential nature of the model. As can be seen, the bottleneck already allows to model to improve significantly upon the ResNet-152 baseline.

By increasing the number of attention steps we can improve adversarial accuracy even further: unrolling for 16 steps (S3TA-16) significantly improves robustness - a S3TA-16 model is more robust against a 1000 PGD attack steps than a ResNet-152 model is against a 100 attack steps. In fact, a S3TA-16 model trained with 10 PGD steps during adversarial training is more robust than a ResNet-152 trained with 30 PGD steps (see Figure 4). This shows that there is a kind of “computational race” here between the strength of the attack and the number of compute steps we allow the model to have. More computation steps for the model mean better defense against stronger attacks. Going beyond a 1000 attack steps does not change the picture as most models saturate close to their 1000 step performance. Full results, including attack success rates, and nominal accuracies, can be found in Table 1 and the supplementary material.

We now turn to compare models adversarially trained with 30 PGD steps. These models are much stronger generally and achieve good robustness results across a wide range of attack strengths, but require a great deal of resources and time to train. Figure 4 shows the top-1 accuracy of a S3TA-16-30 model (“-30” denotes 30 PGD steps during training) vs. a ResNet-152 model and DENOISE [55], the latter being the current state-of-the-art in adversarial robustness. As can be seen, S3TA-16 comfortably outperform both models, setting a new state-of-the-art.

Figure 3: S3TA-2, 4, 8 and 16 vs. ResNet-152 top 1 accuracy on the ImageNet test set. All models were adversarially trained with 10 PGD steps. Note how the introduction of the attention model significantly improve performance even with 2 attention steps, and that adding more steps (S3TA-16) improves performance further: a S3TA-16 model is more robust at a 1000 attack steps than a ResNet-152 model at a 100 attack steps.

Figure 4: A S3TA-16 model adversarially trained with 30 PGD steps vs. ResNet-152 (30 steps) and DENOISE [55] top 1 accuracy on the ImageNet test set. DENOISE is the current state-of-the-art on ImageNet and as can be seen S3TA-16 performs significantly better than both models, setting a new state-of-the-art.
shows several examples of generated adversarial attacks. The main effects observed for top 1 accuracy hold here: more attention steps lower the attack success rate and more PGD steps during training help reduce it even further. S3TA-16-30 clearly has the lowest attack success rates, about 25% lower than DENoise while nominal accuracy is similar (see 1).

The success rate of attacks against S3TA-16-30 is about 25% lower than that of DENoise (lower is better).

5.2. Untargeted and gradient-free attacks

Most robustness measures in the literature are for targeted, gradient based attacks. However, a model that is only robust against targeted attacks is weaker than one robust against untargeted attacks [15]. In Table 2, we report results for untargeted attacks using 200 PGD steps for S3TA-16-30 vs. ResNet-152, DENoise and LLR [42]. Our model is very competitive in this setting, both for $\epsilon = 4/255$ and $\epsilon = 16/255$.

We also explore gradient-free methods to make sure the model does not obfuscate gradients [50, 3]. Specifically, we use random targeted SPSA [50] with a batch size of 4096 and 100 iterations under $\epsilon = 16/255$ for the gradient-free attack. We use iterative signed gradients [31, 55] with gradients estimated by SPSA. Results on a subset of 1000 randomly-chosen images can be seen in Table 3. We can observe that SPSA does not lower accuracy compared to gradient-based attacks. This provides an additional evidence that the model’s strong performance is not due to gradient masking. Given SPSA’s adversarial accuracy is weaker (that is, all models defend better than with gradient-based methods) the performance difference between the models is not very informative.

5.3. Loss landscapes

Another way of making sure gradients are not obfuscated is by visualizing the loss landscapes [42, 49]. Figure 6 shows the top-view of the loss landscape for S3TA-4 and S3TA-16. To visualize the loss landscapes, we change the input along a linear space defined by the worse perturbations found by PGD and a random direction. The $u$ and $v$ axes represent the magnitude of the perturbation added in each of these directions respectively and the $z$ axis represents the loss. For both panels, the diamond-shape stands for the projected $L_\infty$ ball of size $\epsilon = 16/255$ around the nominal image. We can observe that both loss landscapes are rather smooth, which provides an additional evidence that the strong performance is not because of gradient obfuscation.

5.4. Natural adversarial examples

A recent interesting dataset is “Natural adversarial examples” [22]. This curated dataset is composed of natural images of a subset of 200 classes from ImageNet. These images are chosen such that they cause modern image classifier to misclassify an image with high confidence, even though no actual modification to the image is done. The images often contain objects in unusual locations, photographed from unusual angles or occluded or corrupted in a variety of ways. We compare a S3TA-16 model to DENoise, the ResNet baseline and the “Squeeze and excite” [26] (ResNet+SE) variant reported in the original paper. Figure 7 shows the results, using the measures used in the paper: Top-1 accuracy, Calibration error which measures the difference between the confidence of each model and its actual error rate, and AURRA which allows calculation of accuracy while giving classifiers an opportunity to abstain if they are not confident in their prediction.

6. Analysis

We have shown that the sequential attention model improves robustness against a variety of attacks and attack strengths. Furthermore, we have seen that we can increase accuracy and defend better against stronger attacks by unrolling the model for more time steps. We now turn to analyze some of the properties of the resulting attack images and strategies.

Figure 8 shows several examples of generated adversarial examples for different attack strengths for an adversarially trained S3TA model (with 4 unrolling steps) and an adversarially trained ResNet-152. We observe that often (but...
<table>
<thead>
<tr>
<th>Model</th>
<th>Nominal Accuracy</th>
<th>10 steps</th>
<th></th>
<th>100 steps</th>
<th></th>
<th>250 steps</th>
<th></th>
<th>1000 steps</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Top-1 Attack</td>
<td>Top-1 Attack</td>
<td></td>
<td>Top-1 Attack</td>
<td></td>
<td>Top-1 Attack</td>
<td></td>
<td>Top-1 Attack</td>
<td></td>
</tr>
<tr>
<td>ResNet-152</td>
<td>70.66%</td>
<td>53.48%</td>
<td>7.63%</td>
<td>36.91%</td>
<td>43.48%</td>
<td>32.50%</td>
<td>50.93%</td>
<td>29.5%</td>
<td>55.84%</td>
</tr>
<tr>
<td>S3TA-2</td>
<td>72.30%</td>
<td>55.08%</td>
<td>9.06%</td>
<td>40.13%</td>
<td>40.31%</td>
<td>36.56%</td>
<td>46.56%</td>
<td>33.97%</td>
<td>50.40%</td>
</tr>
<tr>
<td>S3TA-4</td>
<td>72.48%</td>
<td>56.78%</td>
<td>9.10%</td>
<td>42.54%</td>
<td>39.37%</td>
<td>40.00%</td>
<td>44.33%</td>
<td>37.99%</td>
<td>47.50%</td>
</tr>
<tr>
<td>S3TA-8</td>
<td>72.14%</td>
<td>56.02%</td>
<td>9.50%</td>
<td>42.63%</td>
<td>39.06%</td>
<td>40.17%</td>
<td>44.01%</td>
<td>38.48%</td>
<td>47.09%</td>
</tr>
<tr>
<td>S3TA-16</td>
<td>72.54%</td>
<td>57.45%</td>
<td>9.33%</td>
<td>44.19%</td>
<td>38.83%</td>
<td>40.71%</td>
<td>44.82%</td>
<td>38.70%</td>
<td>48.16%</td>
</tr>
<tr>
<td>ResNet-152-30</td>
<td>63.62%</td>
<td>53.30%</td>
<td>6.56%</td>
<td>39.56%</td>
<td>34.62%</td>
<td>37.56%</td>
<td>38.68%</td>
<td>36.18%</td>
<td>41.37%</td>
</tr>
<tr>
<td>DENOISE</td>
<td>66.02%</td>
<td>56.33%</td>
<td>5.00%</td>
<td>45.84%</td>
<td>27.36%</td>
<td>44.19%</td>
<td>30.66%</td>
<td>43.39%</td>
<td>32.54%</td>
</tr>
<tr>
<td>S3TA-16-30</td>
<td>64.55%</td>
<td>55.08%</td>
<td>3.79%</td>
<td>47.18%</td>
<td>21.87%</td>
<td>46.65%</td>
<td>23.52%</td>
<td>46.11%</td>
<td>24.91%</td>
</tr>
</tbody>
</table>

Table 1: Full results for all models on random targeted PGD attacks with the ImageNet test set, with different number of attack steps. Bottom three rows are models trained with 30 PGD steps, the rest were trained with 10 PGD steps.

<table>
<thead>
<tr>
<th>Model</th>
<th>Top-1 $\epsilon = 4/255$</th>
<th>Top-1 $\epsilon = 16/255$</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResNet-152</td>
<td>39.7%</td>
<td>6.3%</td>
</tr>
<tr>
<td>DENOISE [55]</td>
<td>38.9%</td>
<td>7.5%</td>
</tr>
<tr>
<td>LLR [42]</td>
<td>47.0%</td>
<td>6.1%</td>
</tr>
<tr>
<td>S3TA-16</td>
<td>46.75%</td>
<td>9.8%</td>
</tr>
</tbody>
</table>

Table 2: Top-1 accuracy under untargeted attacks at 200 PGD steps. As can be seen, our model is very competitive with existing methods though not optimized for this particular attack method.

<table>
<thead>
<tr>
<th>Model</th>
<th>Top-1</th>
<th>Attack Success</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResNet-152</td>
<td>61.90%</td>
<td>2.20%</td>
</tr>
<tr>
<td>DENOISE [55]</td>
<td>63.70%</td>
<td>1.90%</td>
</tr>
<tr>
<td>S3TA-16</td>
<td>59.60%</td>
<td>1.90%</td>
</tr>
</tbody>
</table>

Table 3: Top-1 accuracy under random targeted SPSA attacks (batch size of 4096 and 100 iterations). SPSA is a gradient free method which provides evidence whether gradients are obfuscated. As can be seen all models perform similarly, considering they all defend better here than the corresponding gradient based attack (making the actual reported number less informative).

6.1. Distracting the Attention

Since attention is an integral part in our model, we can see whether it plays a role when the network is attacked and mislabels an image. We can visualize the attention maps generated at each time step and see how the attention is used under different attack scenarios. Figure 9 shows such attention maps for an image used to attack a S3TA-16 model. Attention is superimposed over original image - highlighted areas are more attended than dark areas. As can be seen, the attack can create stimuli which attract the some the attention heads away from the main object in the image, in this case towards something that slightly resembles the target class in the background.

7. Conclusion

In this paper, we have shown that a recurrent attention model inspired by the primate visual system is able to achieve state-of-the-art robustness against random target adversarial attacks. Allowing for more attention steps improves...
accuracy under stronger attacks. The resulting adversarial examples often (but not always) contain global structures which are visible and interpretable to a human observer.

Why is it that global structures arise when attacking a model like this? We postulate that there are two contributing factors. The attention mechanism pools data from large parts of the image, which means that the gradients propagate quickly across the whole of the image, and not just locally. Furthermore, because the the model is unrolled for several steps, more parts of the image may be potentially attended to and thus gradients may propagate there. We see evidence for this in the fact the often the attacker attracts the attention away from the main object in the image, hinting that the attention plays a crucial role in the attack strategy.

There is still work to be done to achieve adversarial robustness in complex datasets. Even models like the proposed one often fail when the attacker is strong enough, and performance is still quite low compared to nominal accuracies, but at some point we may ask — if an image has been perturbed enough such that it does not resemble the original image and looks like an image coming from the target class, is it still a valid adversarial perturbation? Models like the one presented here may allow us to reach that frontier in the future.
References


[16] David C. Van Essen and Charles H. Anderson. Information processing strategies and pathways in the primate visual system. 1995. 2


[34] Yan Luo, Xavier Boix, Gemma Roig, Tomaso A. Poggio, and Qi Zhao. Foveation-based mechanisms alleviate adversarial examples. *CoRR*, abs/1511.06292, 2015. 2


[56] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C. Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2015. 2

