Adversarial samples in machine learning: revealing flaws of neural network training

András Horváth

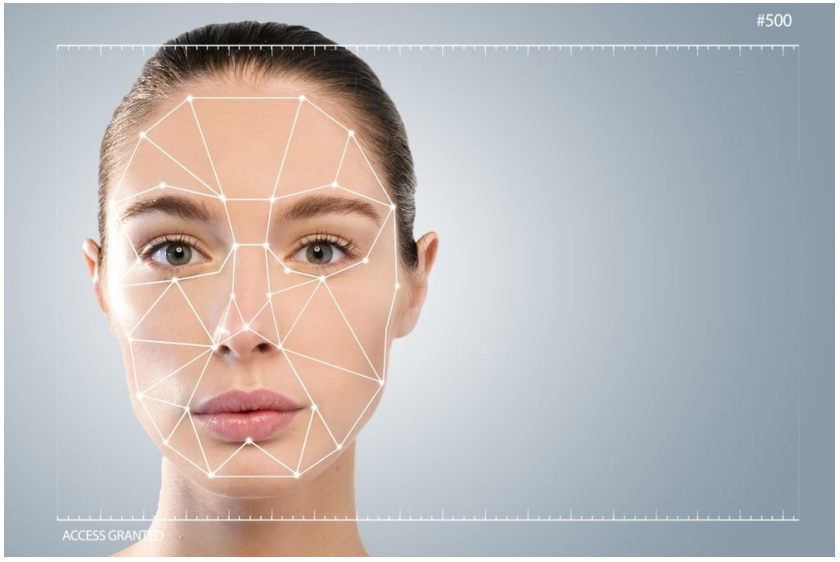
Eger, 2020. 01. 29
International Conference on Applied Informatics

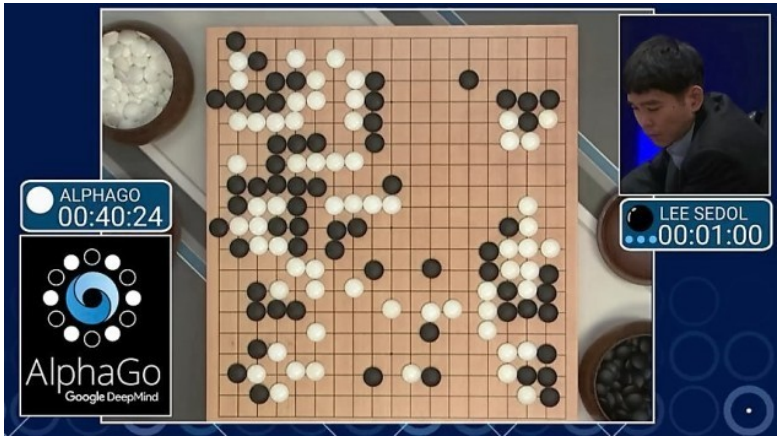
Machine Learning – Neural networks

Face recognition

Self-driving cars

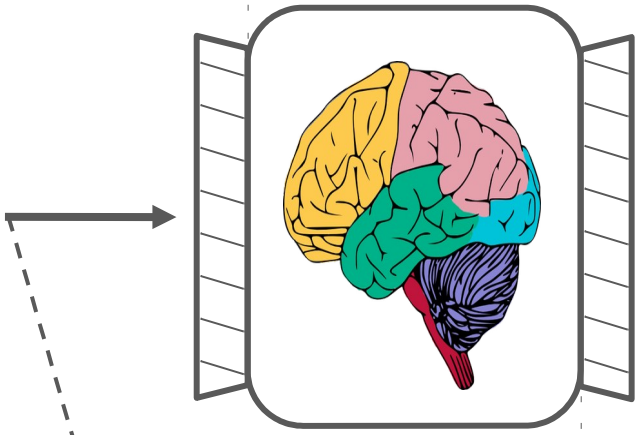
Alpha Go



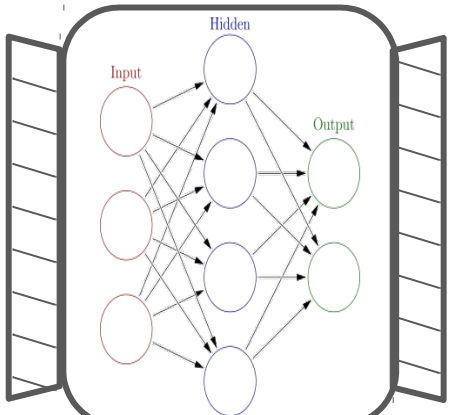


Deep Neural Network

Supervised Learning



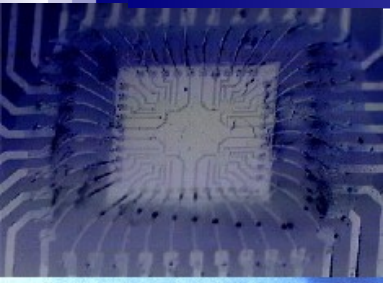
“Cat”



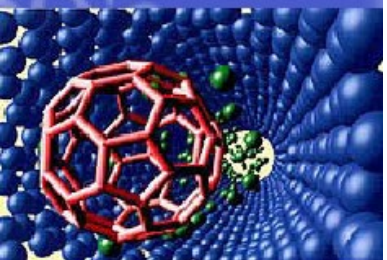
80%
“Cat”

Machine Learning – Agricultural risk assessment

Machine Learning – 3D object detection



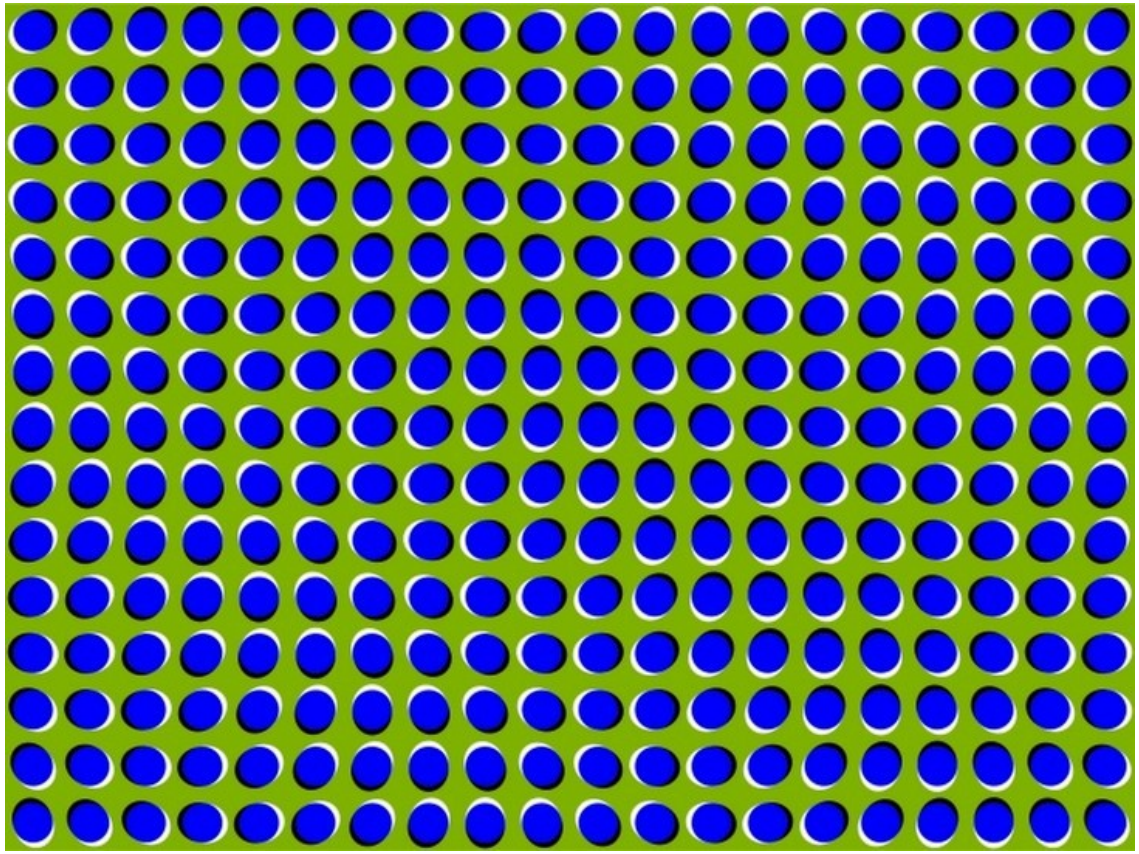




Adversarial samples

Adversarial Samples for Neural Networks

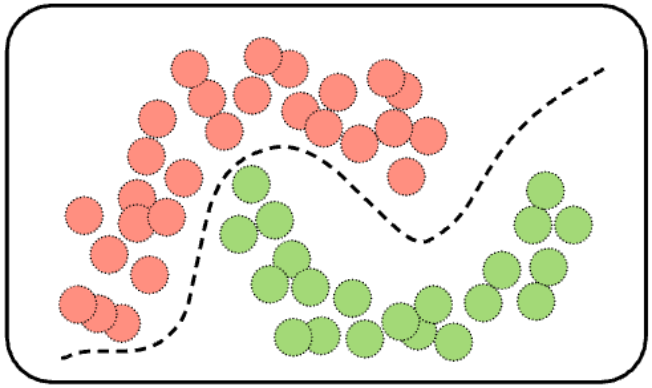
Optical Illusions for neural networks
Special, constructed elements, which can not be found in the normal input set



Adversarial attacks

We have a high number of parameters to be optimized

An even higher-dimensional input

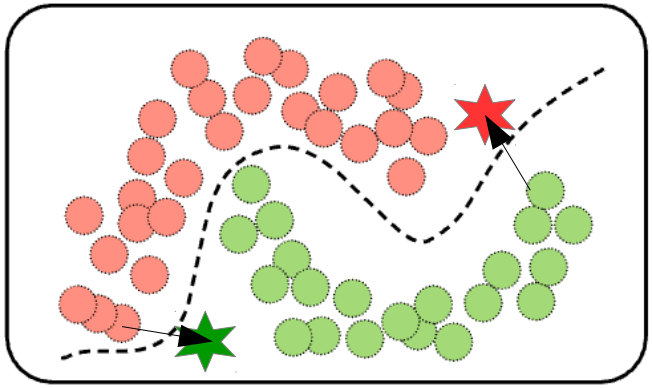


The network works well in practice, but can not cover all the possible inputs

Adversarial attacks

We have a high number of parameters to be optimized

An even higher-dimensional input

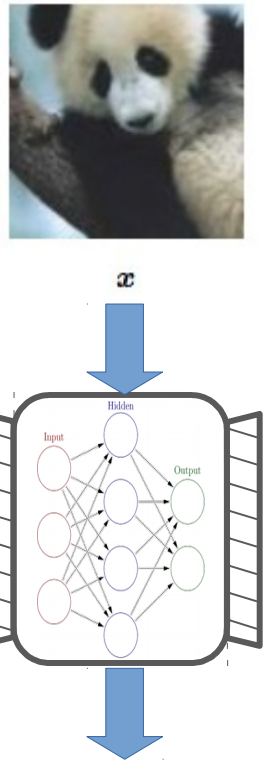


The network works well in practice, but can not cover all the possible inputs

One can exploit that there will be regions in the input domain, which were not seen during training

Adversarial noise

We have a working well-trained classifier:



Panda

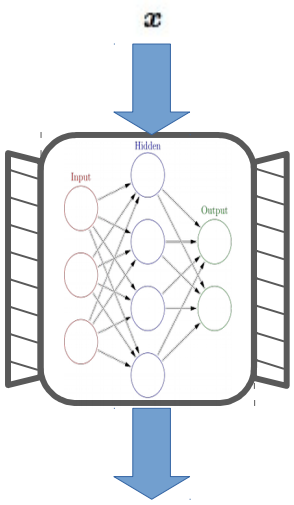
[Goodfellow, I. J., Shlens, J., & Szegedy, C. (2014). Explaining and harnessi
examples. *arXiv preprint arXiv:1412.6572*

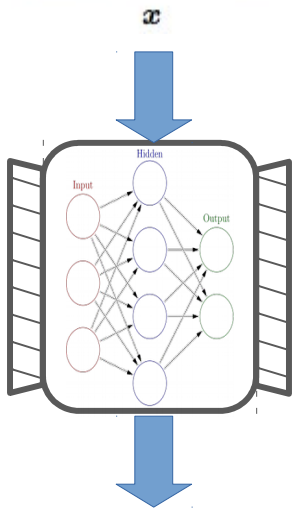
Adversarial noise

What should I add to the input to causes misclassification:

+

???





Panda

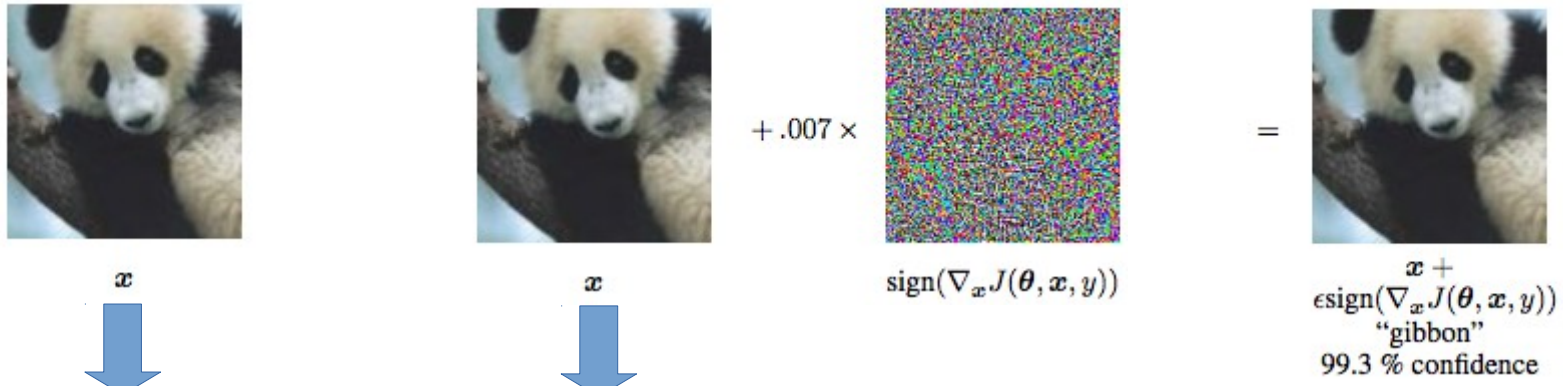
Gibbon

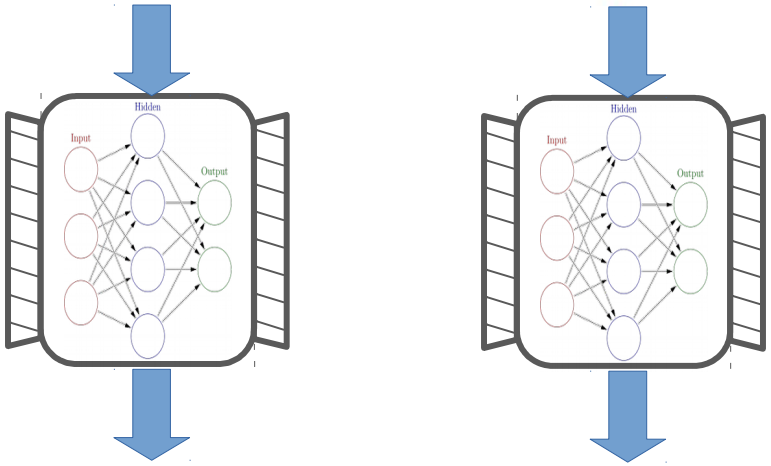
The noise is generated by gradient descent optimization

Adversarial noise

A special, low amplitude additive noise:

The two images are the same for human perception



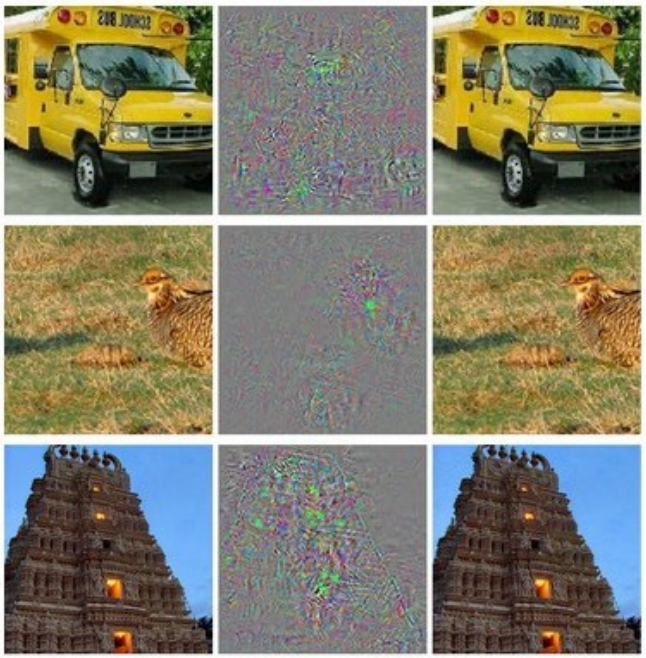


Panda **Gibbon**

The noise is generated by gradient descent optimization

Adversarial noise

Knowing a trained network one can identify modifications (which does not happen in real life), which change the network output completely

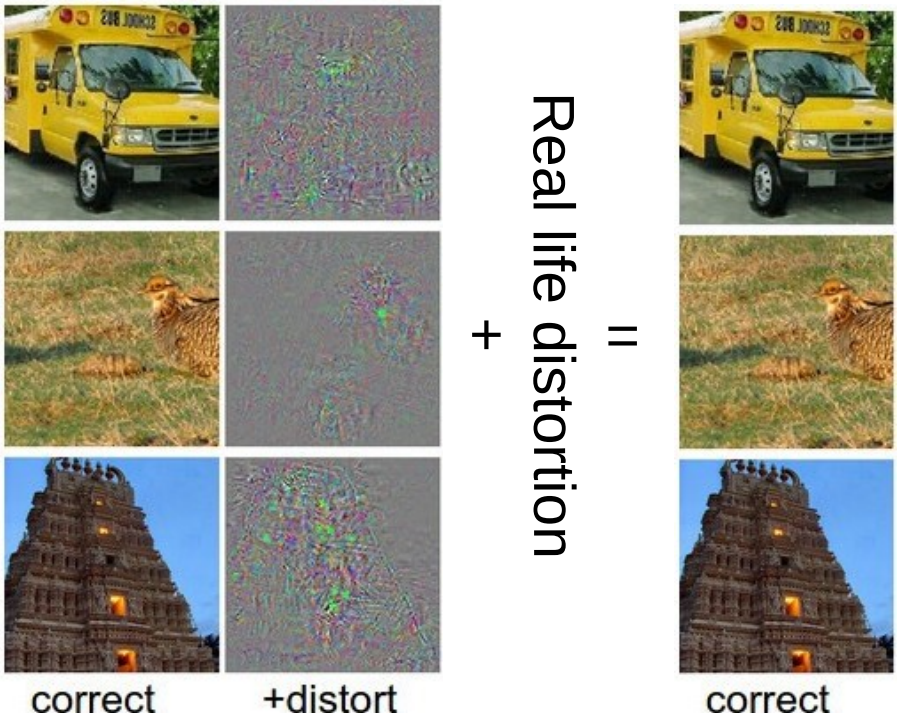


Adversarial noise – does not work in practice

Knowing a trained network one can identify modifications (which does not happen in real life), which change the network output completely

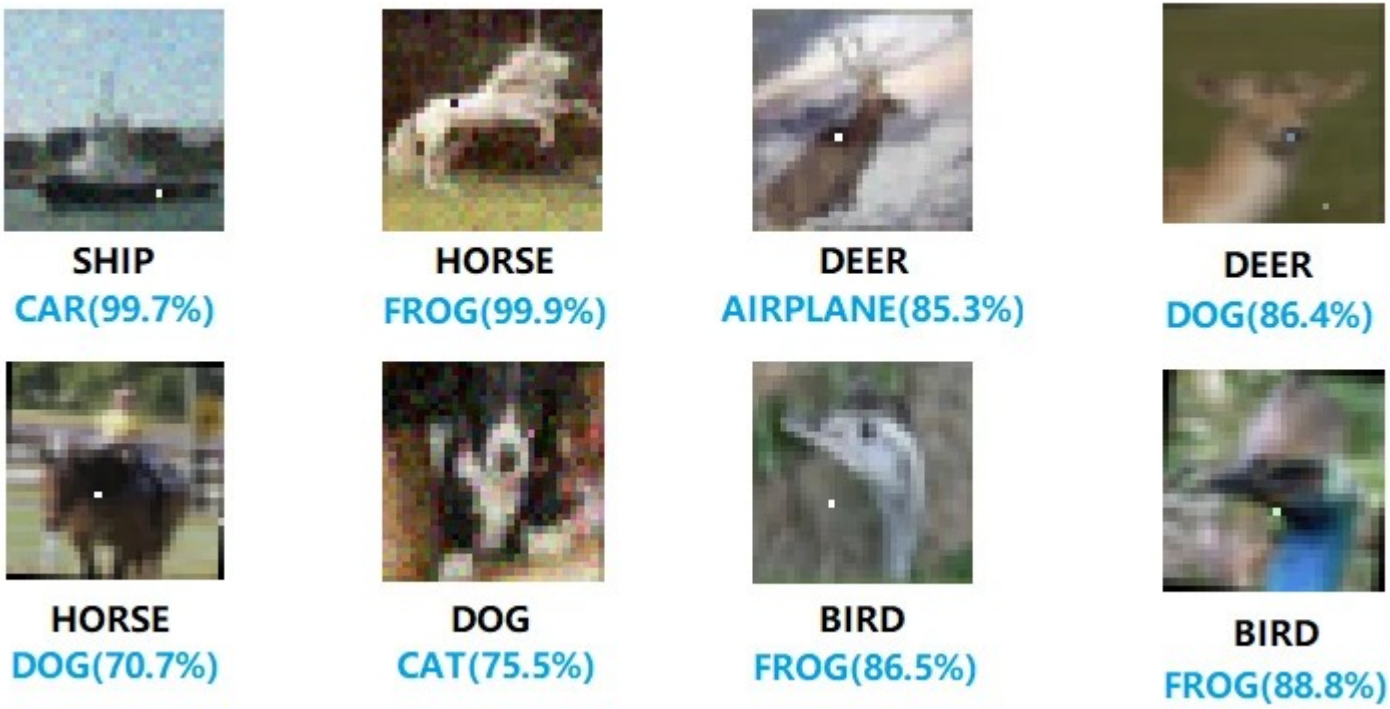
Luckily this low amplitude noise is not robust enough in real life (lens distortion and other additive noises)

Jiajun Lu, Hussein Sibai, Evan Fabry, and David Forsyth. No need to worry about adversarial examples in object detection in autonomous vehicles. 2017. URL https://arxiv.org/abs/1707.03501.g



One Pixel attacks

In many cases it is enough to modify a single pixel to change the output class of the network



Su, J., Vargas, D. V., & Sakurai, K. (2019). One pixel attack for fooling deep neural networks. IEEE Transactions on Evolutionary Computation.

Adversarial patches

Large changes concentrated over a small region

Athalye, A., Engstrom, L., Ilyas, A., & Kwok, K. (2017). Synthesizing robust adversarial examples. arXiv preprint arXiv:1707.07397.

Sticker based adversarial attacks

High intensity noise concentrated on a small region of the input image:

$$C_d = N\left(I + \sum_{i=1}^{k} St_i\left(x_i, y_i, w_i, h_i\right) + \sum_{j=1}^{l} St_j\left(x_j, y_j, w_j, h_j\right) \right)$$

Parameters are the positions (x,y) and size (w,h) of the stickers

Eykholt, K., Evtimov, I., Fernandes, E., Li, B., Song, D., Kohno, T., ... & Tramer, F. (2017). Note on Attacking Object Detectors with Adversarial Stickers. arXiv preprint arXiv:1712.08062.

Sticker based adversarial attacks

High intensity noise concentrated on a small region of the input image:

$$C_d = N\left(I + \sum_{i=1}^{k} St_i\left(x_i, y_i, w_i, h_i\right) + \sum_{j=1}^{l} St_j\left(x_j, y_j, w_j, h_j\right) \right)$$

Parameters are the positions (x,y) and size (w,h) of the stickers

It was shown that these attacks are **robust** enough to be applied in practical applications

Evtimov, I., Eykholt, K., Fernandes, E., Kohno, T., Li, B., Prakash, A., ... & Song, D. (2017). Robust physical-world attacks on machine learning models. arXiv preprint arXiv:1707.08945.

Sticker based adversarial attacks

High intensity noise concentrated on a small region of the input image:

$$C_d = N\left(I + \sum_{i=1}^{k} St_i\left(x_i, y_i, w_i, h_i\right) + \sum_{j=1}^{l} St_j\left(x_j, y_j, w_j, h_j\right) \right)$$

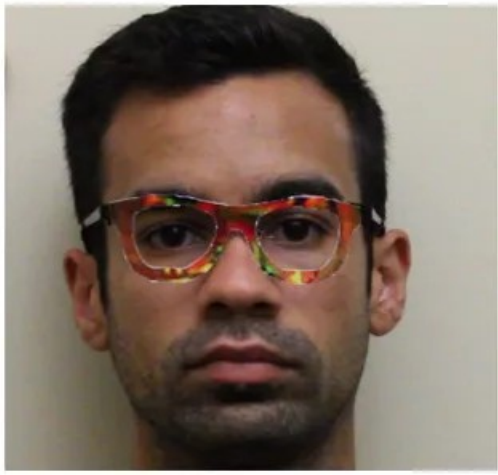
Parameters are the positions (x,y) and size (w,h) of the stickers

It was shown that these attacks are **robust** enough to be applied in practical applications

Does this mean that convolutional neural networks can not be used in critical problem in practice anymore?

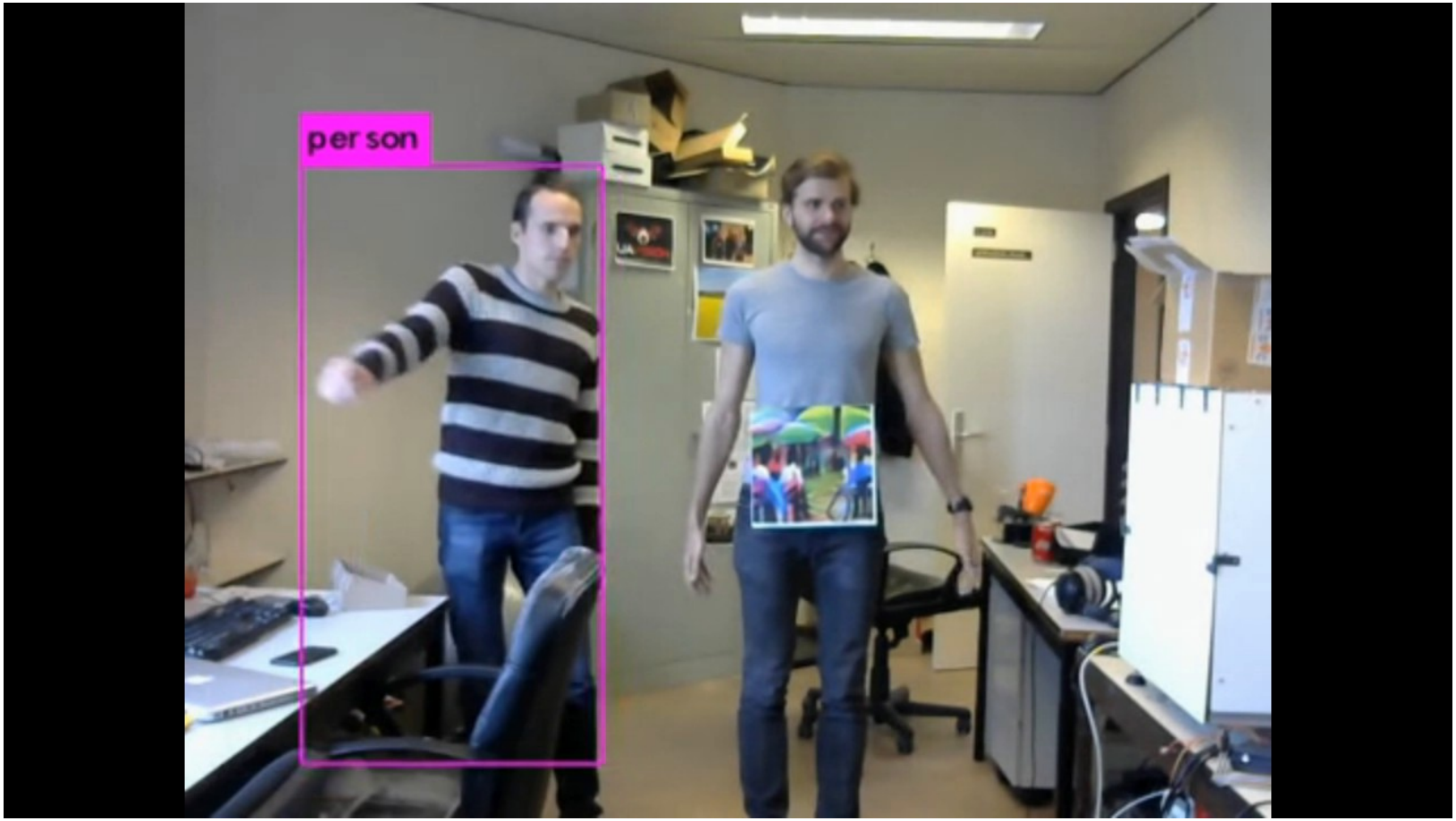
Evtimov, I., Eykholt, K., Fernandes, E., Kohno, T., Li, B., Prakash, A., ... & Song, D. (2017). Robust physical-world attacks on machine learning models. arXiv preprint arXiv:1707.08945.

Sticker based adversarial attacks



(b) (c) (d)

Sticker based attacks on detection



Liu, X., Yang, H., Liu, Z., Song, L., Li, H., & Chen, Y. (2018). DPatch: An Adversarial Patch Attack on Object Detectors. arXiv preprint arXiv:1806.02299.

Benefiting from adversarial samples

Generating hard samples

One can identify difficult samples using adversarial samples

Generating hard samples

Not the number of samples matters, but the relevance of the samples

Generating hard samples

Not the number of samples matters, but the relevance of the samples

Harnessing hard samples

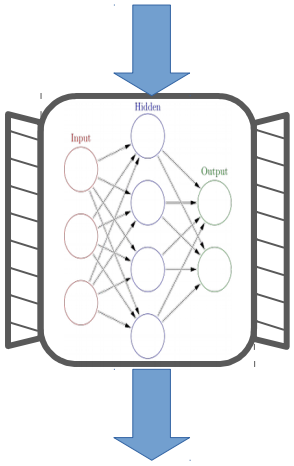
How much modification (how many steps) does it take to change the output class of the network?

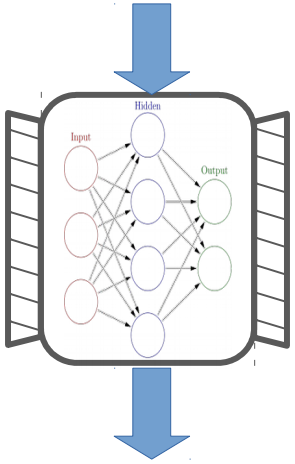
+ ???

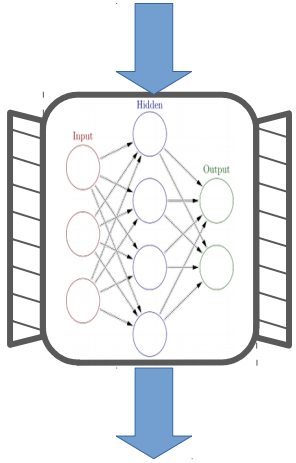
+ ???

x

x







Poodle

Poodle

Poodle

Generating hard samples

Not the number of samples matters, but the relevance of the samples

Generating hard samples

Not the number of samples matters, but the relevance of the samples

Generating hard samples

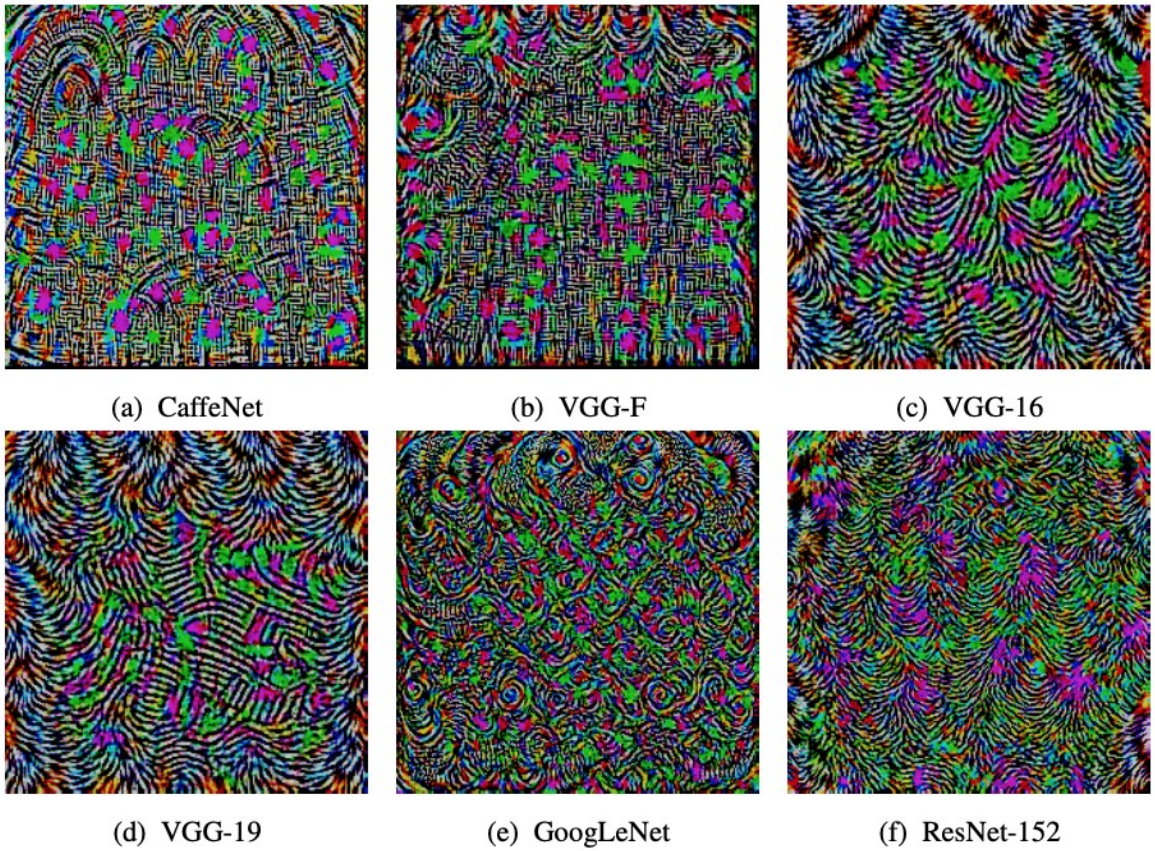
Not the number of samples matters, but the relevance of the samples

Generating hard samples

Not the number of samples matters, but the relevance of the samples

Generating hard samples

If our adversarial noise is arbitrary, we will end up with arbitrary features

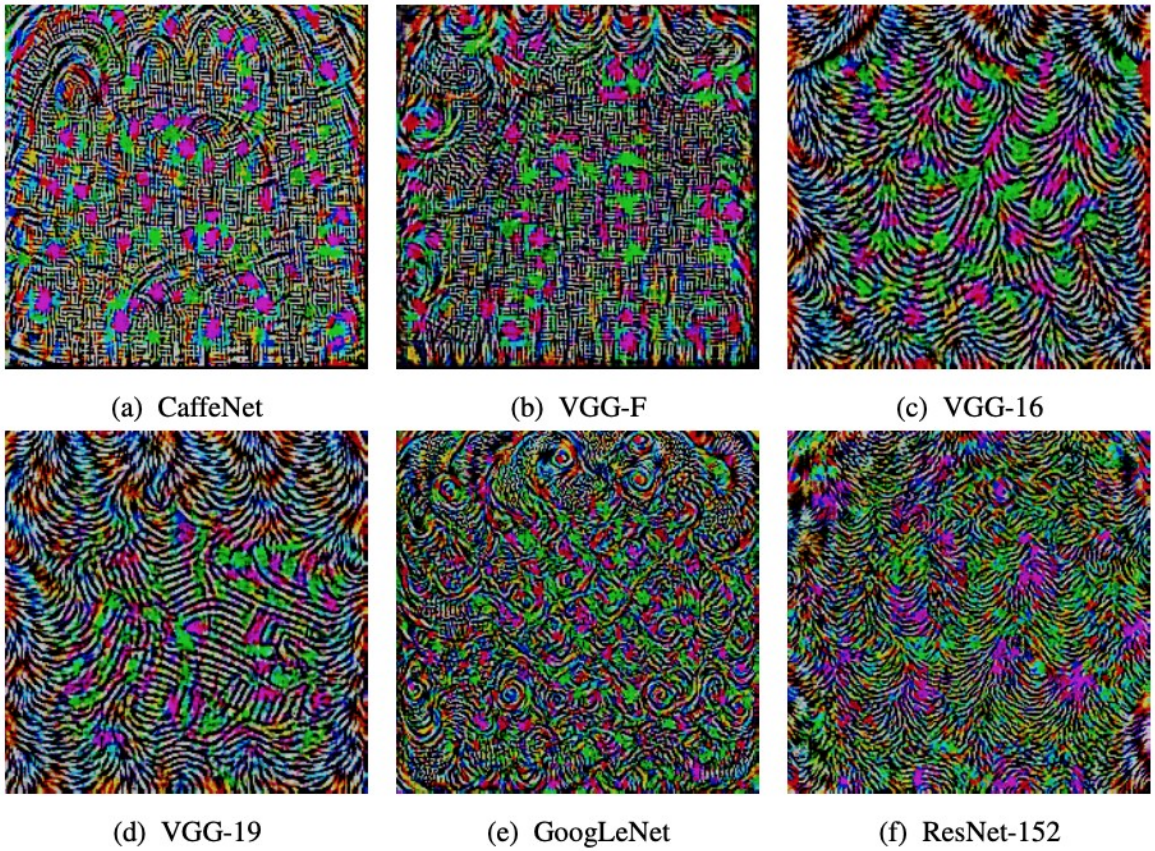


Moosavi-Dezfooli, S. M., Fawzi, A., Fawzi, O., & Frossard, P. (2017). Universal adversarial perturbations. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1765-1773).

Generating hard samples

If our adversarial noise is arbitrary, we will end up with arbitrary features

It will not reveal important (real life) features

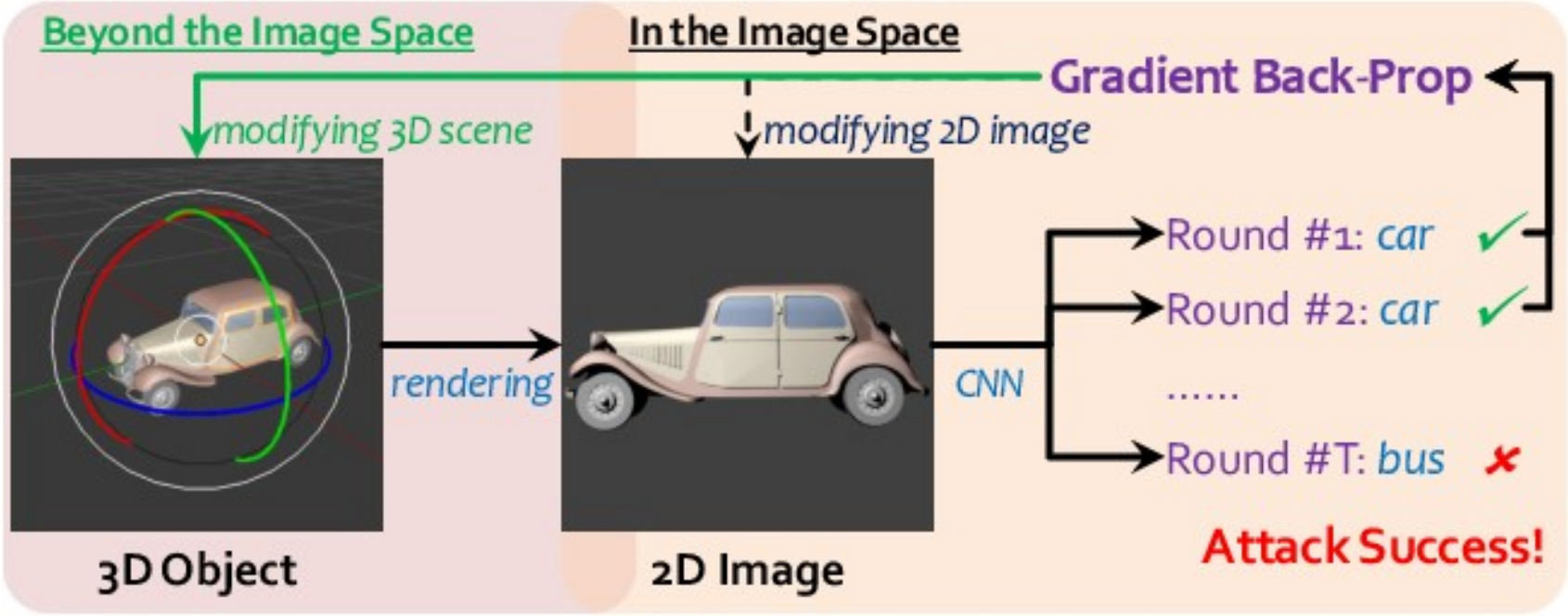


Moosavi-Dezfooli, S. M., Fawzi, A., Fawzi, O., & Frossard, P. (2017). Universal adversarial perturbations. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1765-1773).

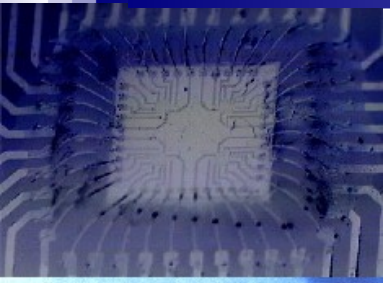
Generating hard samples – adversarial data augmentation

One can identify difficult samples using adversarial samples

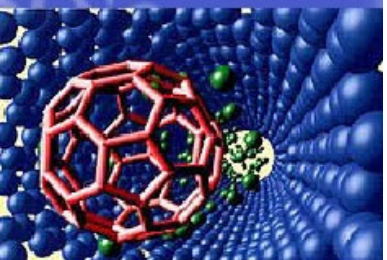
Using abstract models of the objects and applying a restricted set of transformations



Zeng, X., Liu, C., Wang, Y. S., Qiu, W., Xie, L., Tai, Y. W., ... & Yuille, A. L. (2019). Adversarial attacks beyond the image space. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 4302-4311).







Detcting adversarial samples

Understanding decisions

We might be interested in case of a single sample, what triggered the decision of the network

The network only outputs probabilities. Could we display why the network made this decision?

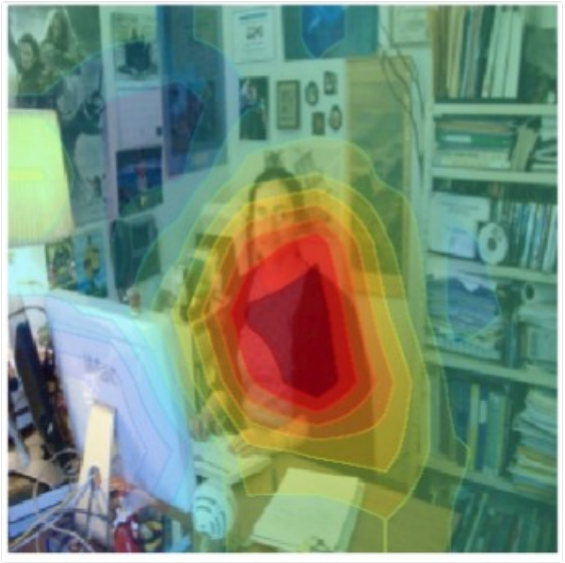
Understanding decisions

If we can understand (or even trace back) network decisions we will be able to see whether the network managed to grasp the important features in the dataset or not

Wrong

Baseline:
*A **man** sitting at a desk with a laptop computer.*

Right for the Right Reasons



Our Model:
*A **woman** sitting in front of a laptop computer.*

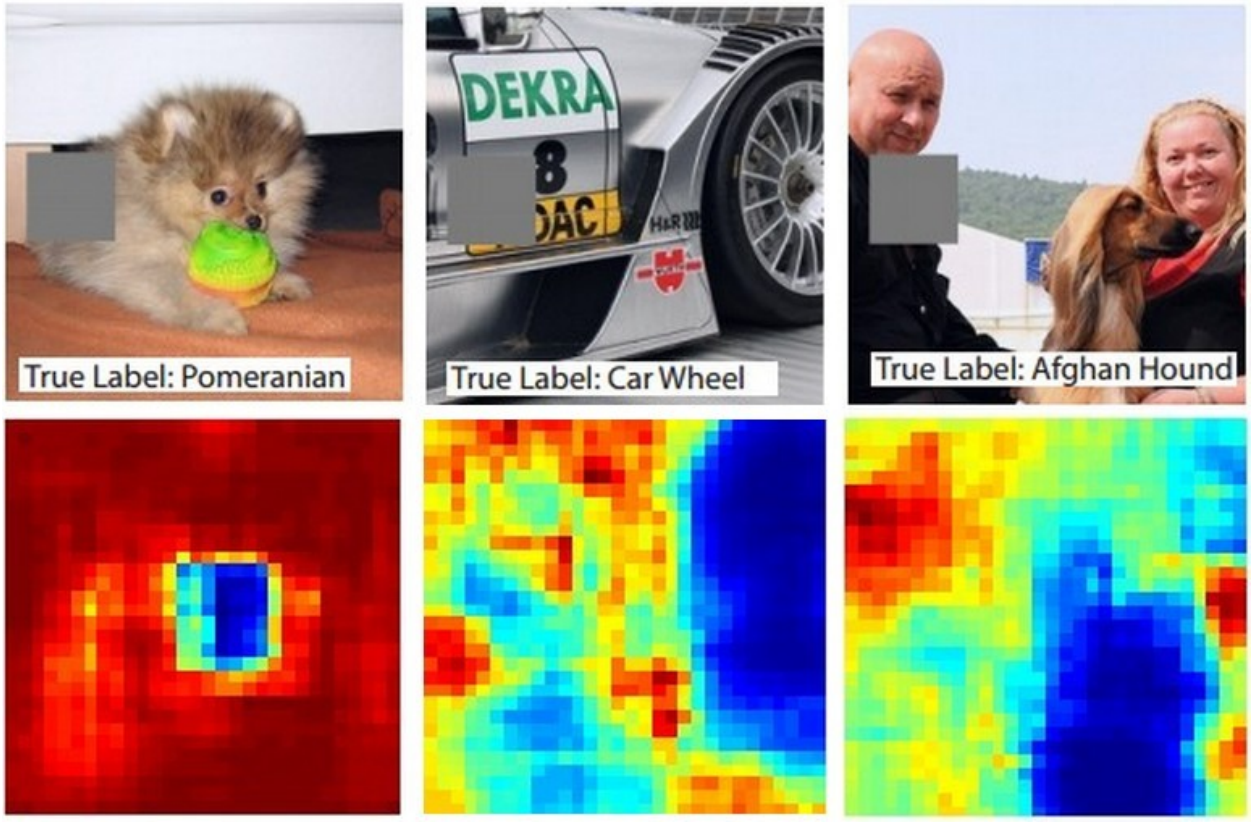
Lisa Anne Hendricks*, Kaylee Burns*, Kate Saenko, Trevor Darrell, Anna Rohrbach:Women also Snowboard: Overcoming Bias in Captioning Models

Reasoning by occlusion

We might occlude part of the input image.

If the decision does not change → the occluded part was unimportant

If the decision changes → the part was important, The importance of the part is proportional with the change



Zeiler, M. D., & Fergus, R. (2014, September). Visualizing and understanding convolutional networks. In European conference on computer vision (pp. 818-833). Springer, Cham.

Reasoning by importance

Occlusion maps are good

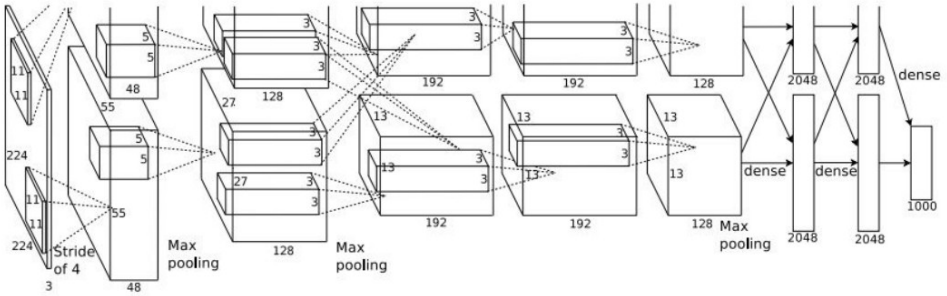
Calculating an occlusion map takes a lot of time

Could we calculate the importance of each pixel in the decision?

Reasoning by importance

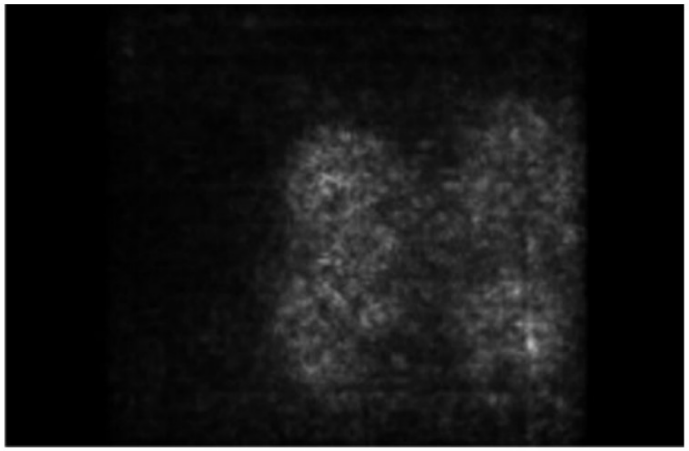
Could we calculate the importance of each pixel in the decision?

Forward pass: regular computation



Dog

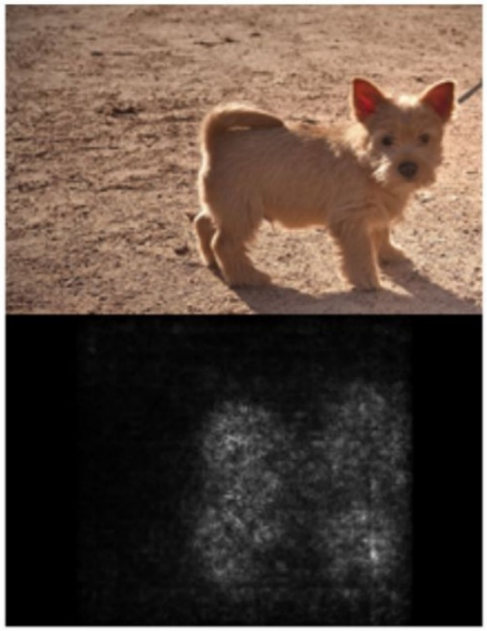
Backward pass: Computing the gradient of (unnormalized) class score
Taking their absolute value and max over RGB channels

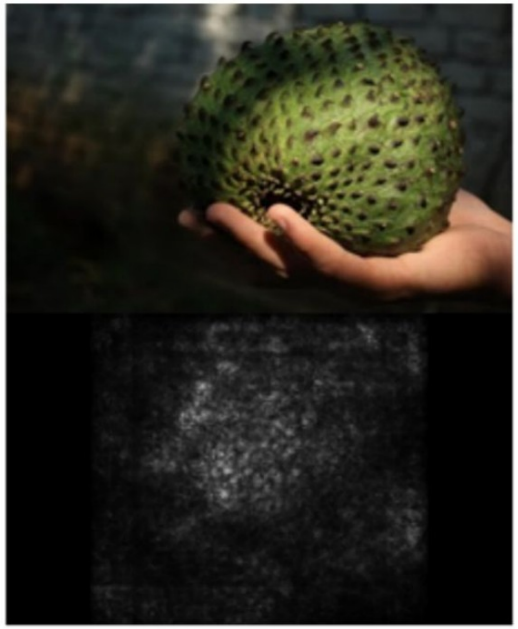


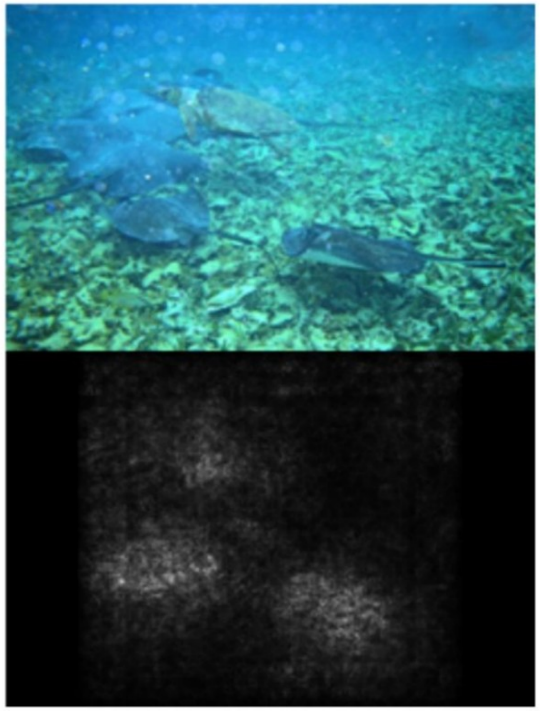
Reasoning by importance

Calculating e the importance of each pixel in the decision?

Right for the right reasons



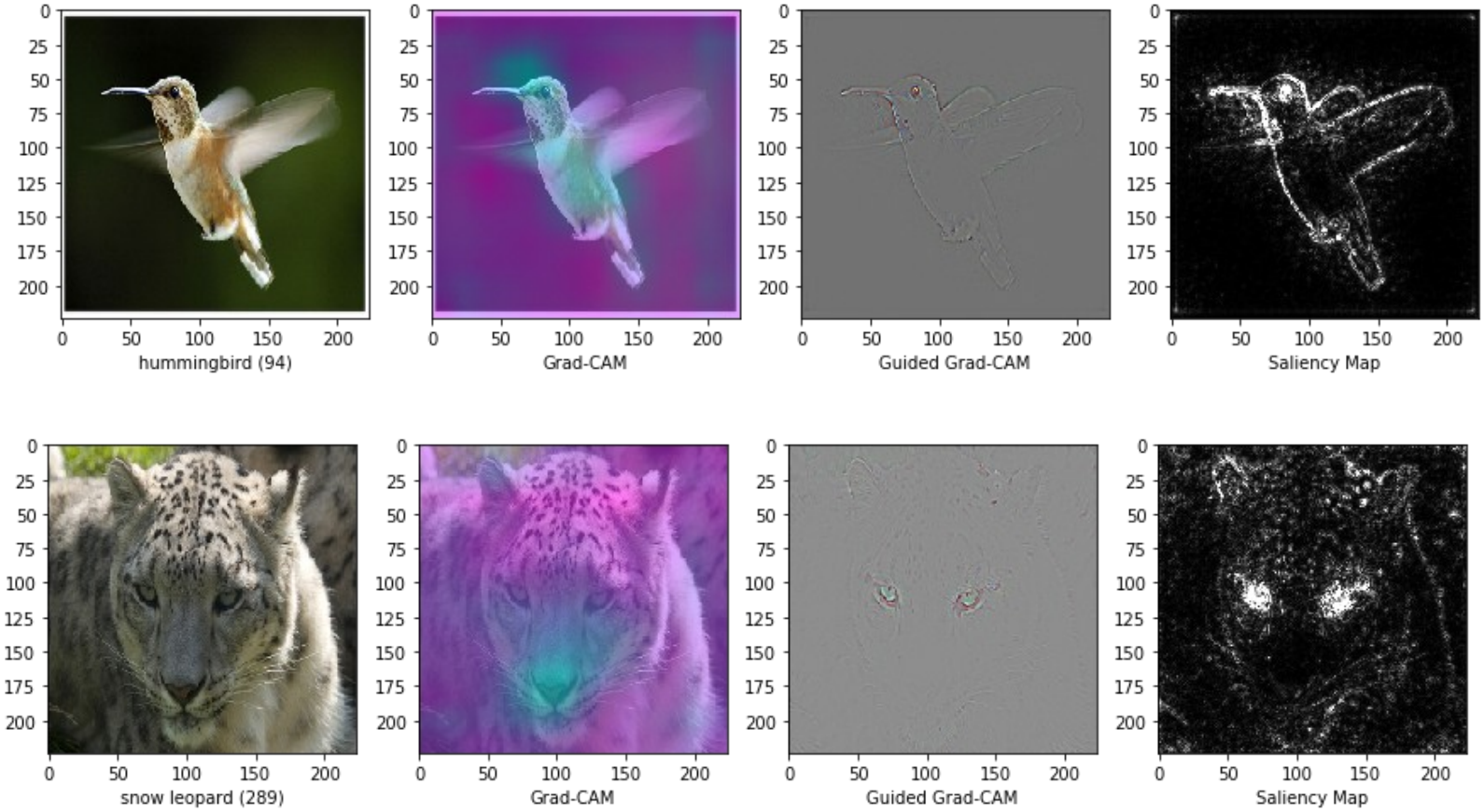




Reasoning by importance

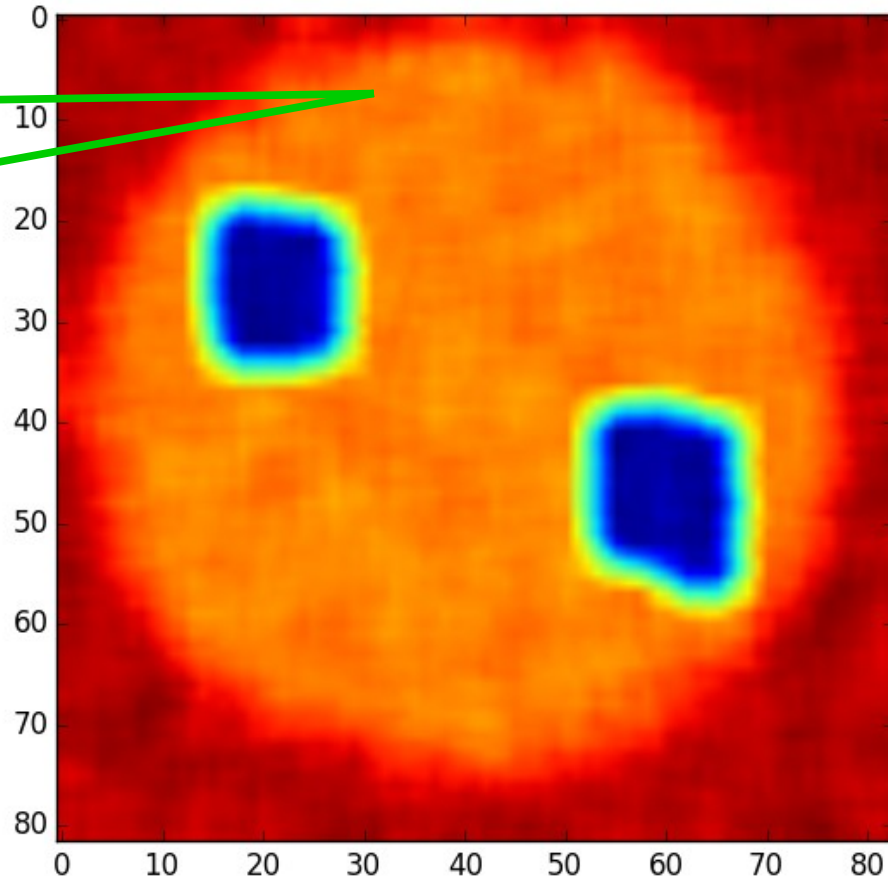
Calculating the importance of each pixel in the decision?

Right for the right reasons



Consistency based detection

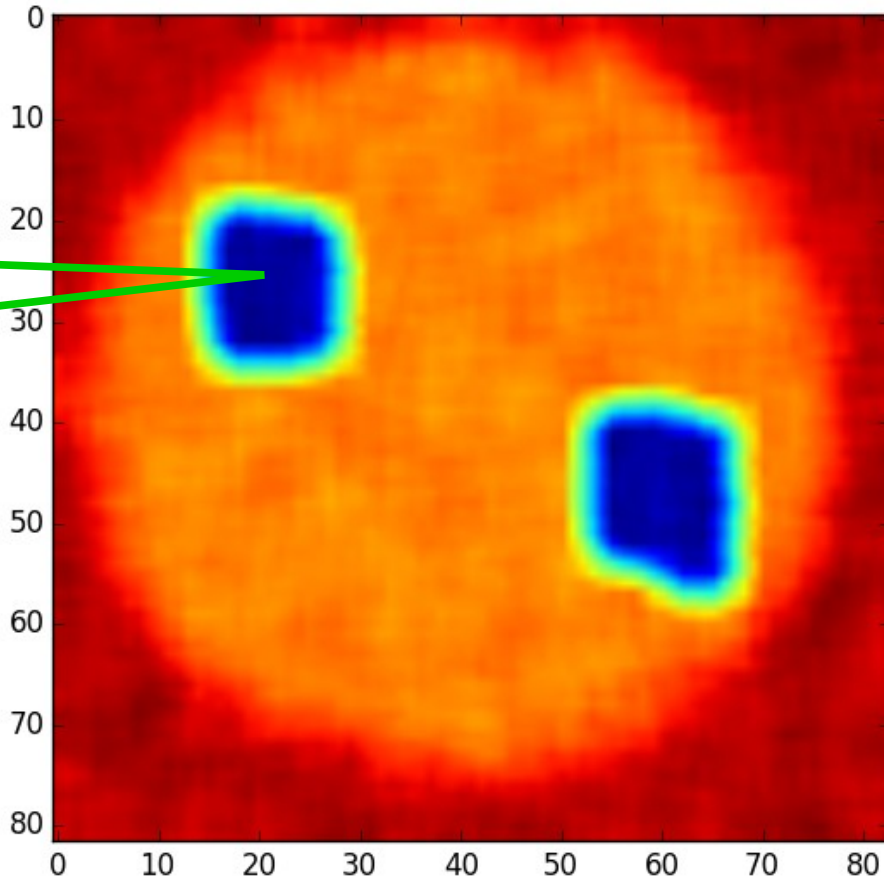
Original output: Stop Sign
Covered output: Stop Sign
Small change in output confidence



Horváth András, Csanád Egervári: Detection of sticker based adversarial attacks, ICDIP 2018

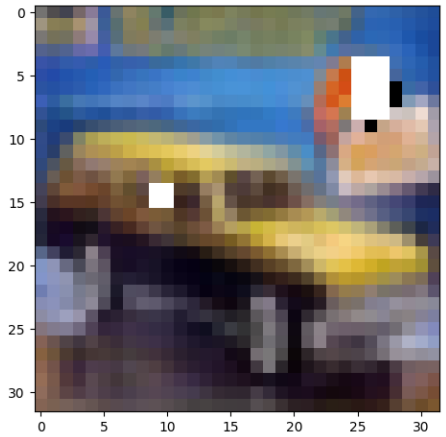
Consistency based detection

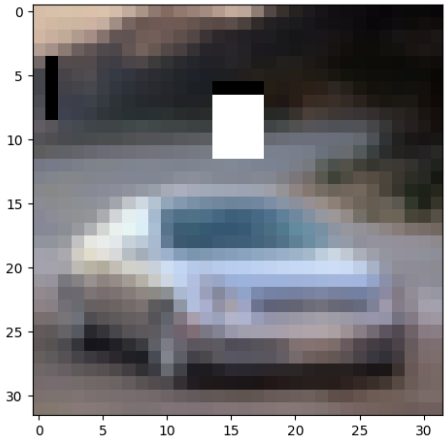
Original output: Stop Sign
Covered output: Speed Limit Sign
Large change in output confidence

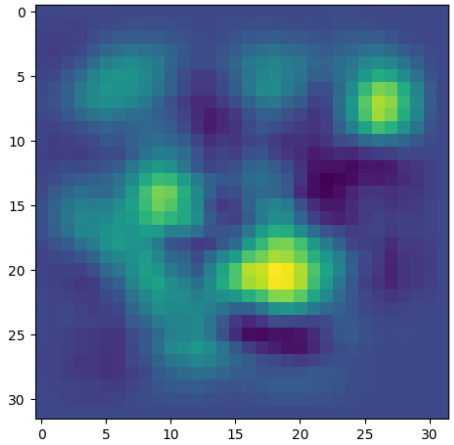


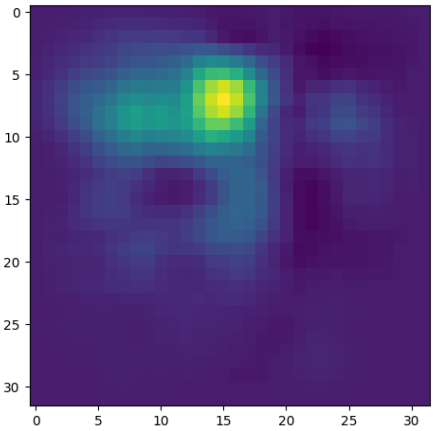
Horváth András, Csanád Egervári: Detection of sticker based adversarial attacks, ICDIP 2018

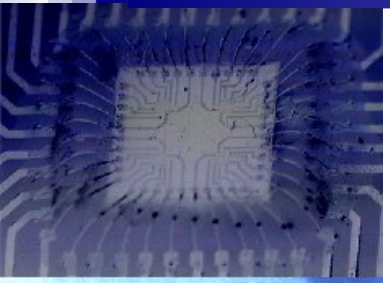
Consistency based detection on an other dataset (CIFAR 10)

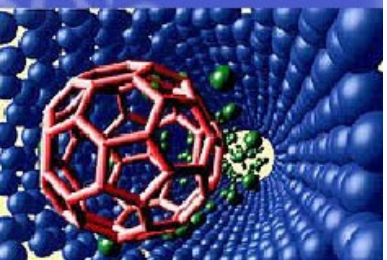












Preventing adversarial attacks

Overconfident networks

Training on CIFAR10 – Test on SVHN

Dog (100%)

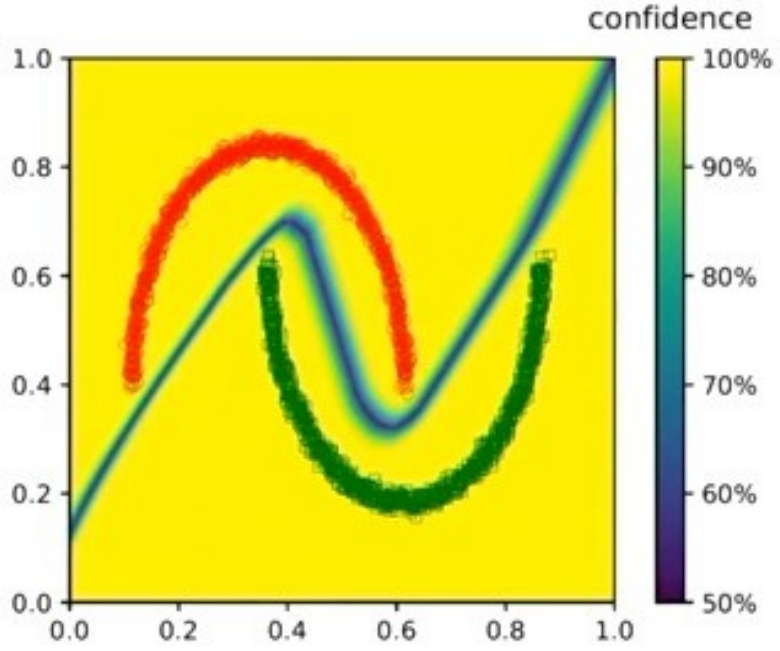
Bird (100%)

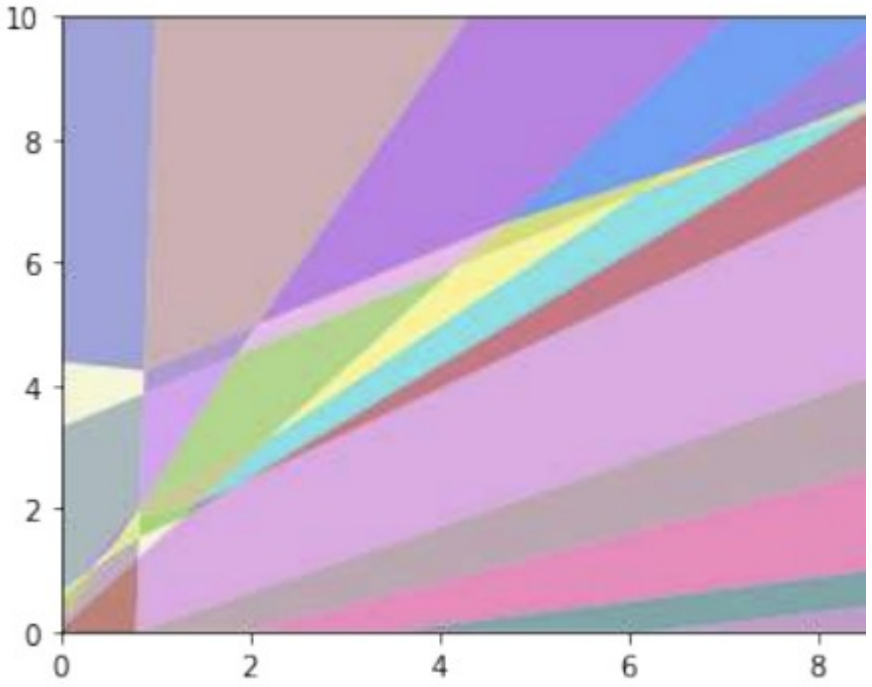
Airplane (100%)

Hein, M., Andriushchenko, M., & Bitterwolf, J. (2019). Why ReLU networks yield high-confidence predictions far away from the training data and how to mitigate th problem. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 41-50).

Overconfident networks

ReLUs allow arbitrarily large activations





Hein, M., Andriushchenko, M., & Bitterwolf, J. (2019). Why ReLU networks yield high-confidence predictions far away from the training data and how to mitigate th problem. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 41-50).

Overconfident networks

78,800 views | Apr 18, 2019, 11:46pm

This In-Ear Translator Can Interpret A Bilingual Conversation On The Fly

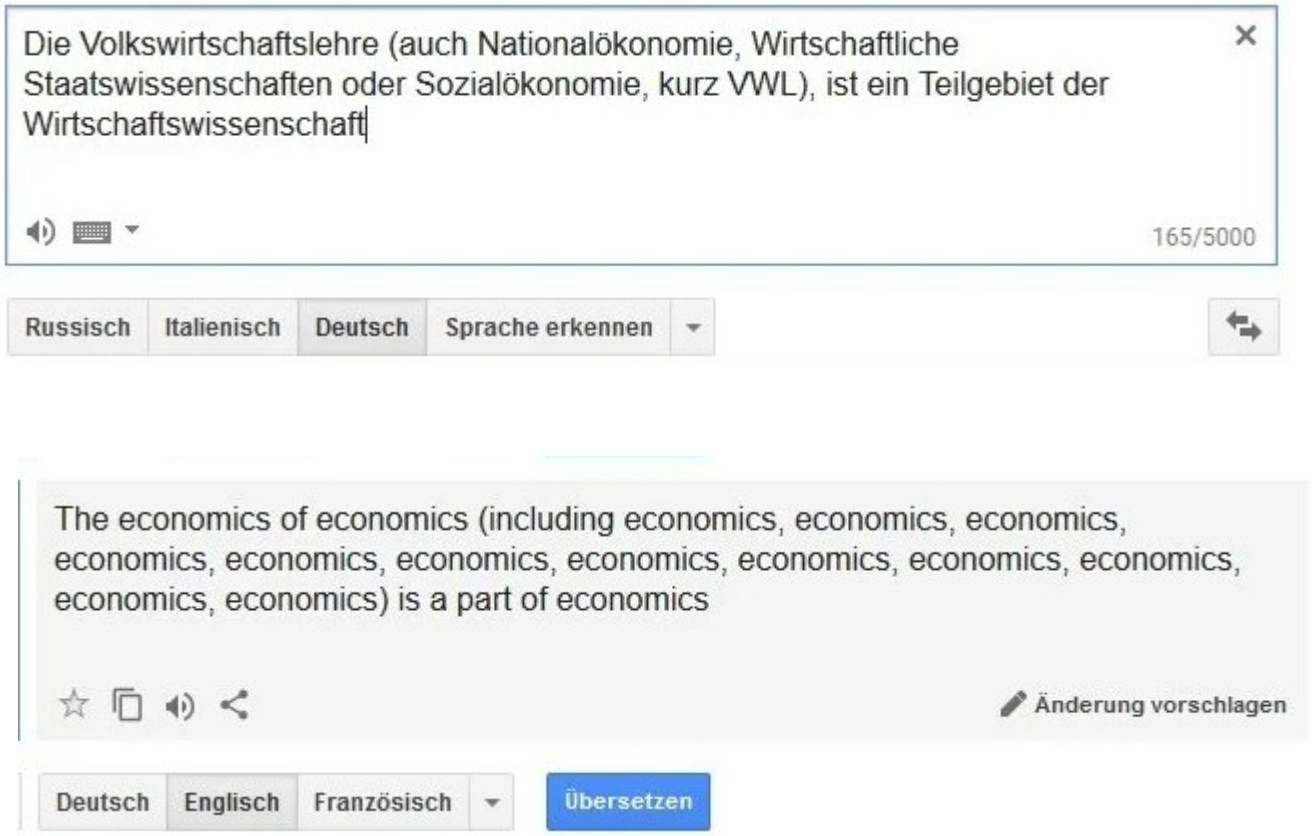
Ben Sin Contributor
I cover consumer tech in Asia

Hein, M., Andriushchenko, M., & Bitterwolf, J. (2019). Why ReLU networks yield high-confidence predictions far away from the training data and how to mitigate th problem. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 41-50).

Overconfident networks

Hein, M., Andriushchenko, M., & Bitterwolf, J. (2019). Why ReLU networks yield high-confidence predictions far away from the training data and how to mitigate th problem. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 41-50).

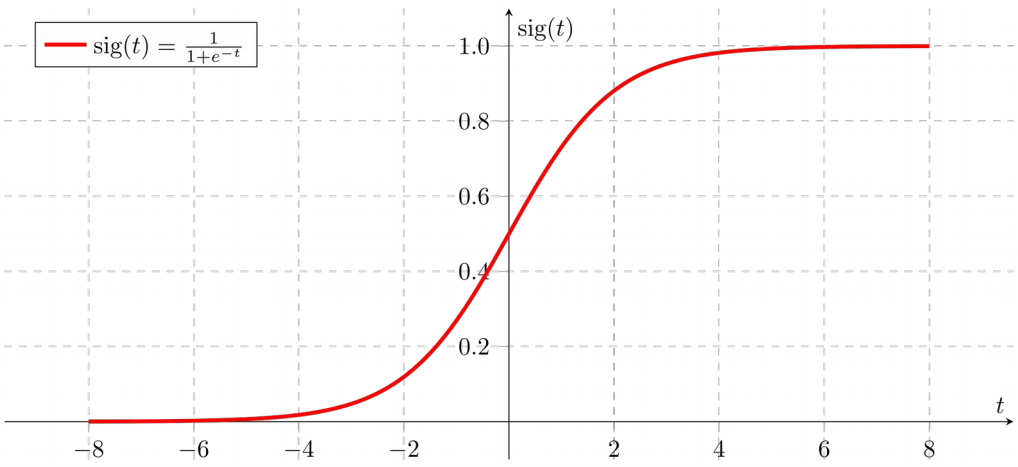
Overconfident networks



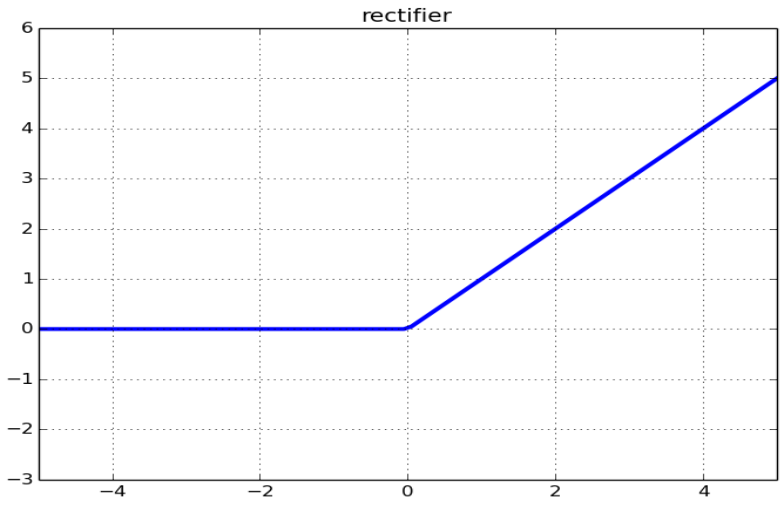
Hein, M., Andriushchenko, M., & Bitterwolf, J. (2019). Why ReLU networks yield high-confidence predictions far away from the training data and how to mitigate th problem. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 41-50).

Non-linearities

Sigmoid



ReLU

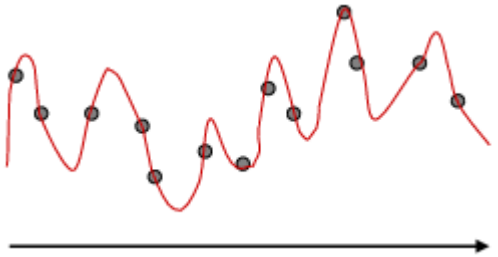


ReLU is the most commonly used non-linearity and also usually it has the best performance

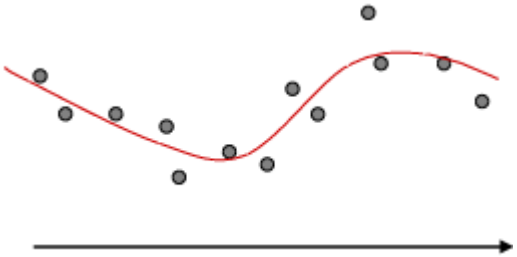
Regularization

Regularization conrtolls the weights of the network

$$\min_f |Y_i - f(X_i)|^2$$



$$\min_{f \in H} \sum_{i=1}^{n} |Y_i - f(X_i)|^2 + \lambda \|f\|_H^2$$

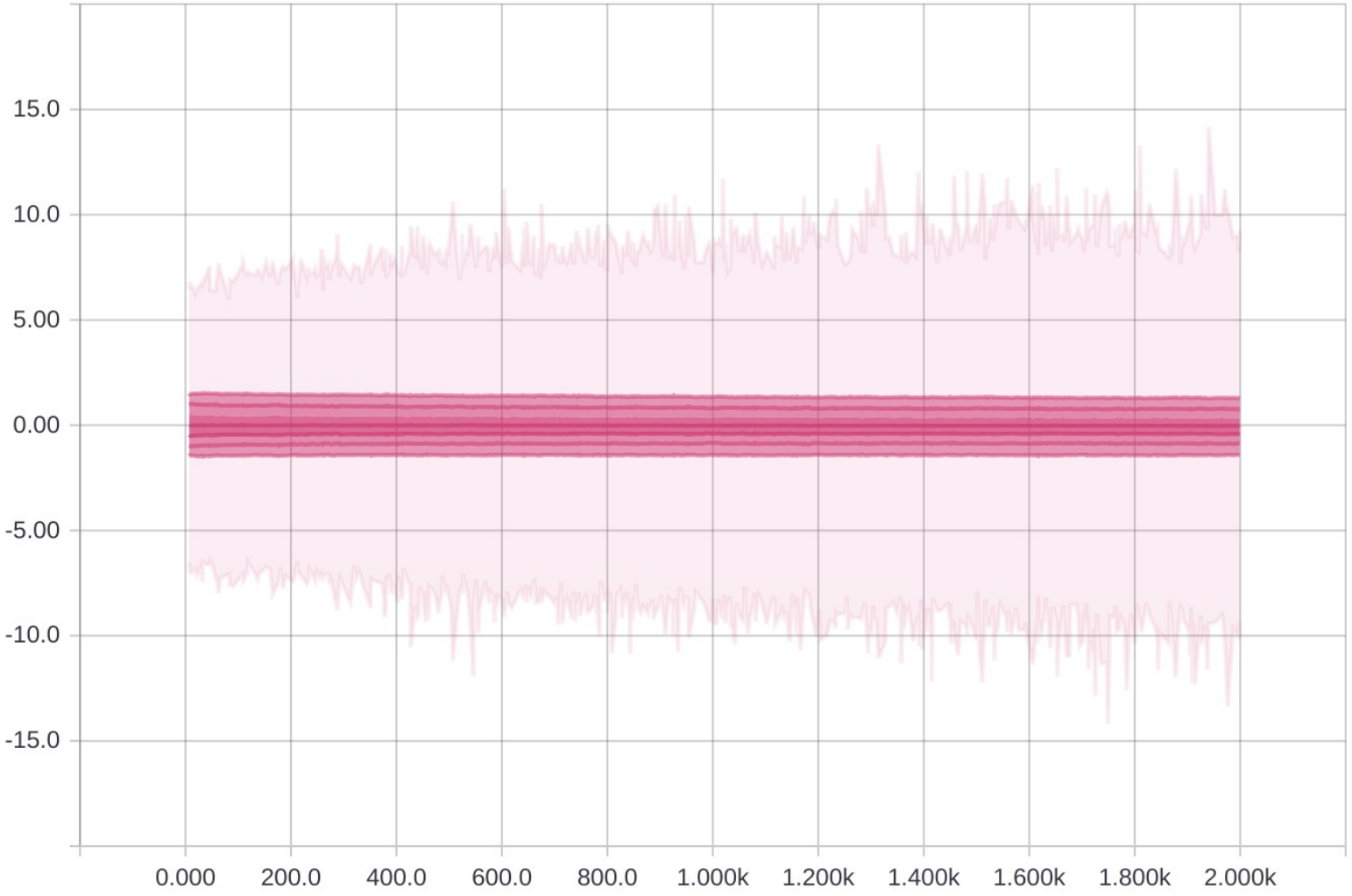


The activation of a neuron is the weight multiplied by the input

Even though the distribution of the weights is good, the distribution of the activations can have arbitrarily large elements

Activations during training

On CIFAR10 with AlexNet using Batch Normalization



Overrepresented activaitons

A network could learn the circular objects (wheels) are important elemnts of cars

Overrepresented activaitons

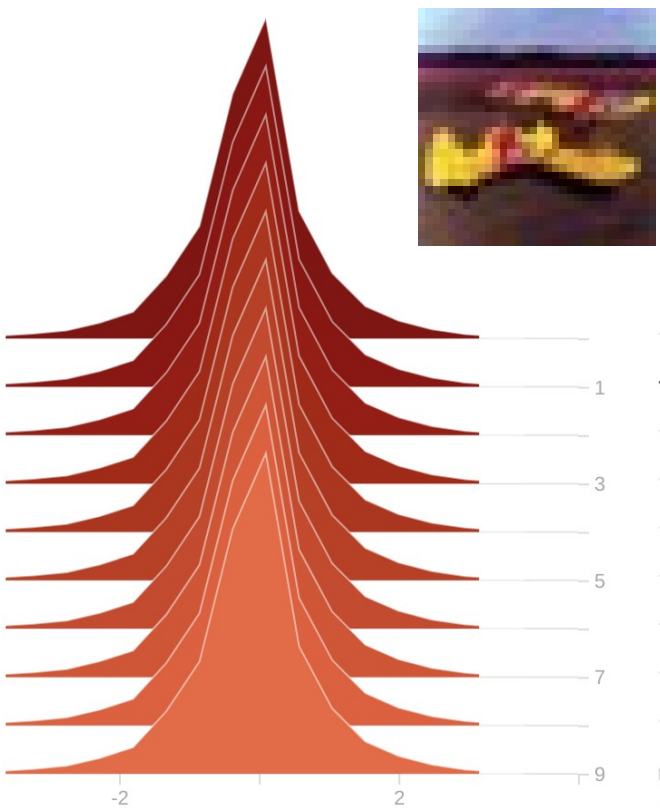
A network could learn the circular objects (wheels) are important elemnts of cars

Our dataset can have samples where certain features are overrepresented

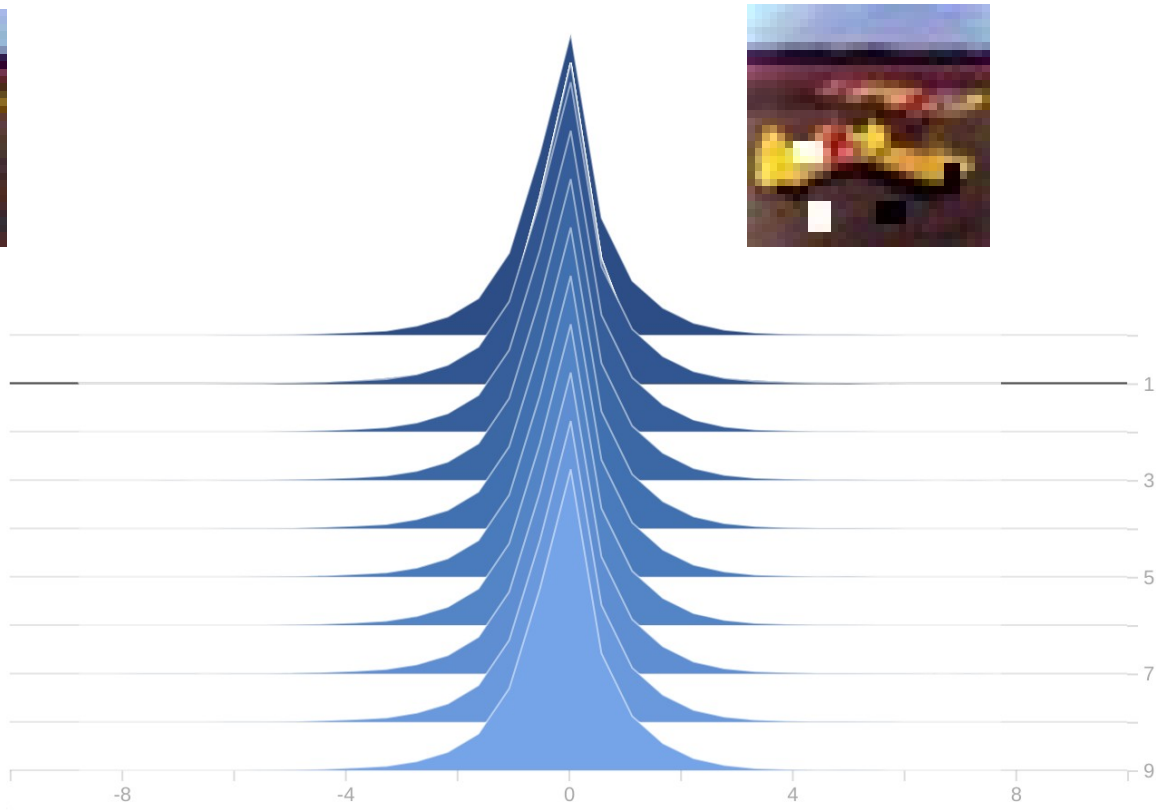
These samples can envoke arbitrarily large activations (even with regularization)

Patch based attacks are exploiting arbitrarily large activations

The evoke and arbitrarily large response in a few neurons



Activations fn the original input

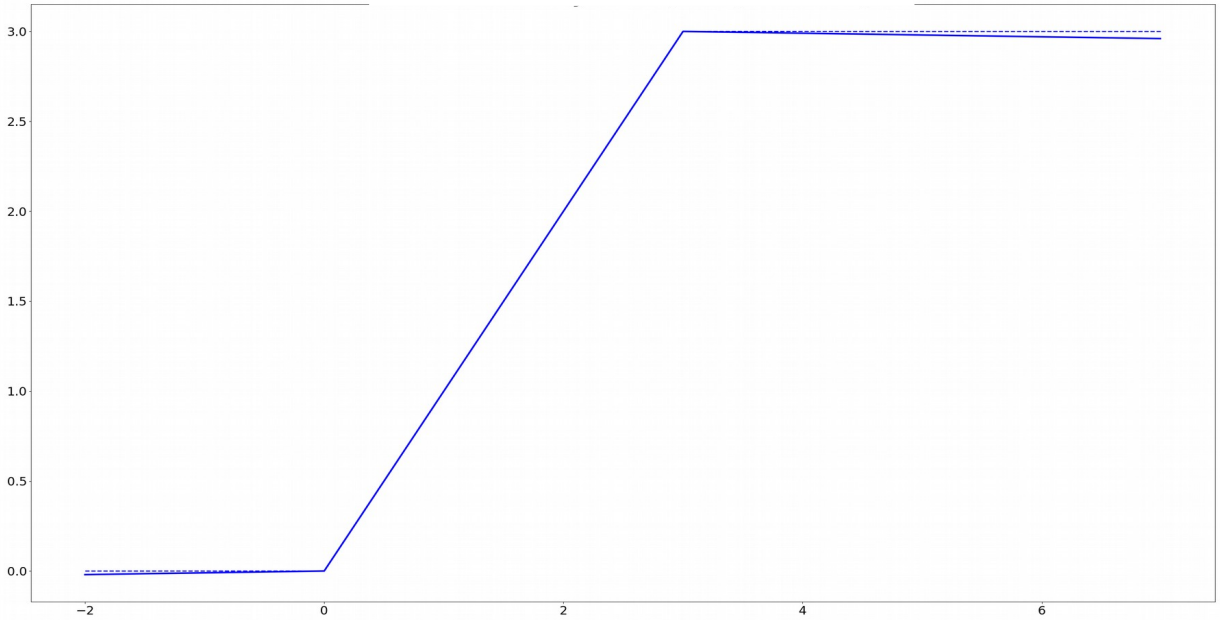


Activations of the attacked input

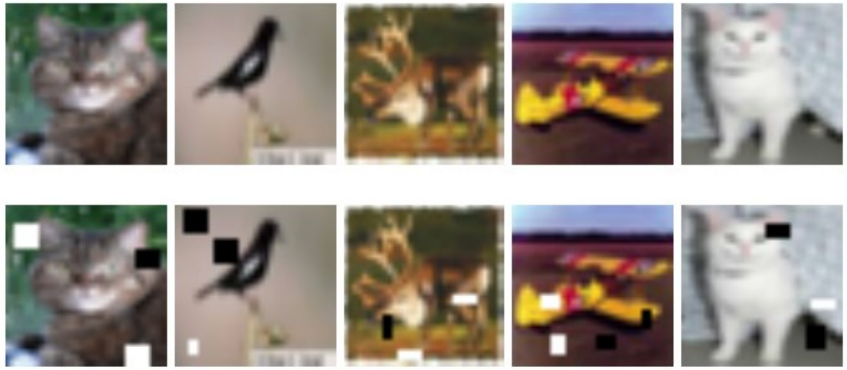
Bounded non-linearity, with a trainable upper bound

Bounded leaky Relu

$$BL\text{-}ReLU(x,b) = \begin{cases} sx & if \quad x < 0 \\ x & if \quad 0 \leq x \leq b \\ b - s(x-b) & if \quad b < x \end{cases}$$



Bounded non-linearity, with a trainable upper bound



Percent of Successful Attacks (Two Way)	
None: 3.5%	Bounded ReLU: 0.5%
Original ReLU: 88%	Both: 8%
Percent of Successful Attacks (Ten Way)	
None: 20.5%	Bounded ReLU: 0%
Original ReLU: 76%	Both: 3.5%

Percent of Successful Attacks (Two Way)	
None: 3%	Bounded ReLU: 0%
Original ReLU: 91%	Both: 6%
Percent of Successful Attacks (Forty Way)	
None: 17%	Bounded ReLU: 0%
Original ReLU: 80.5%	Both: 2.5%