Deep Learning: Why does it work? How to get it robust?

Matthias Hein

Department of Computer Science
University of Tübingen

Joint work with:

Quynh Nguyen

Maksym Andriushchenko

Talk overview

- Why does it work? Why/When can we train neural networks efficiently?
- How to get it robust? How can we guarantee that a classifier is robust against adversarial manipulation?

Success stories of Deep Learning I - Computer Vision

ImageNet Challenge has 1000 classes and 1.2 million training images

ILSVRC Winner	2010	2011	2012	2013	2014	2015	2016	2017
top-5 error in %	28,2	25,8	15,3	11,7	6,6	3,6	3,0	2,3

before deep learning and using deep learning (CNN's)

Why/when can we train neural networks efficiently?

Bad news for training neural networks...

- Training neural networks is computationally hard e.g. Blum, Rivest (1998), Sima (2002), Livni et al (2014).
- Neural networks can have exponentially many (suboptimal) local minima Auer et al (1996), Safran, Shamir (2016)

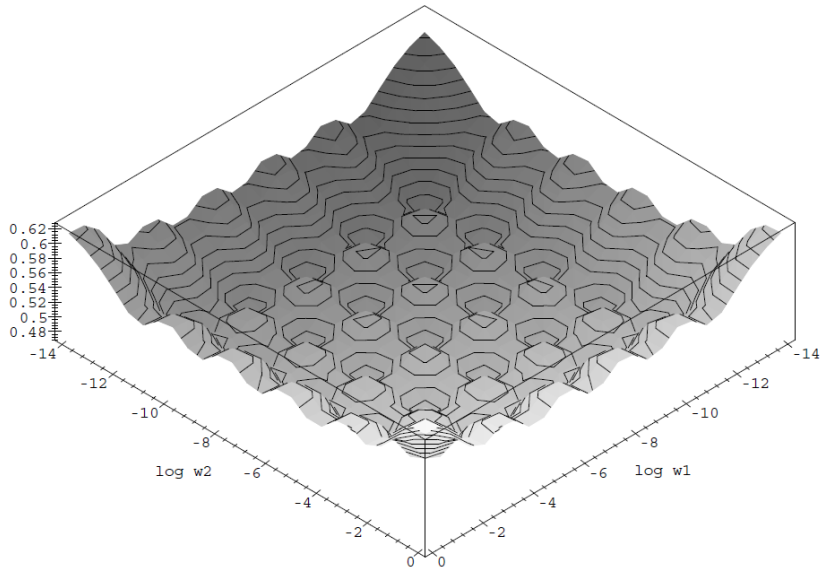


Figure 1: Error Function with 25 Local Minima (16 Visible), Generated by 10 Two-Dimensional Examples.

taken from Auer et al (1996) - single neuron for two-dimensional input

...but in practice no problems!

Empirical Observation: suboptimal local minima seem not to be a big problem in training large deep networks (Goodfellow et al, 2015)

Can one justify that (overparameterized) neural networks are "easy" to train?

- Choromanska et al (2015): randomization of ReLU-activation function plus further assumptions allow to reduce it to spin-glass model, where the local minima are concentrated
- Baldi,Hornik (1988), Kawaguchi(2016): in deep linear networks all local minima are global minima
- lots of further recent work...

Prior work: analysis limited to one-hidden layer networks, deep linear networks or use distributional or other simplifying assumptions
Our goal: analysis for deep networks which are used in practice

General assumptions

1. There are no identical training samples, $x_i \neq x_j$ for all $i \neq j$ resp. for CNNs no input patches are allowed to be identical
2. σ is analytic on $\mathbb{R}$, strictly monotonically increasing and
   1. σ is bounded or
   2. there are positive $\rho_1, \rho_2, \rho_3, \rho_4$, s.t. $|\sigma(t)| \leq \rho_1 e^{\rho_2 t}$ for $t < 0$ and $|\sigma(t)| \leq \rho_3 t + \rho_4$ for $t \geq 0$
3. $l \in C^2(\mathbb{R})$ and if $l'(a) = 0$ then a is a global minimum of l

Typical examples which satisfy the assumptions e.g.

- $\sigma_1(t) = \frac{1}{1+e^{-t}}$, $\sigma_2(t) = \tanh(t)$, $\sigma_3(t) = \frac{1}{\alpha}\log(1 + e^{\alpha t})$ for $\alpha > 0$. Smooth approximation of ReLU: $\lim_{\alpha\to\infty} \sigma_3(t) = \max\{0, x\}$.
- Squared loss $l(a) = a^2$ or twice differentiable Huber loss, but no cross-entropy loss.

Assumptions cover a fairly general class of neural networks

Setting

Multi-layer neural network:

$$f_L(x_i) = W_L\sigma\big(W_{L-1}\sigma\big(\ldots\sigma\big(W_1x_i + b_1\big)\ldots\big) + b_{L-1}\big) + b_L.$$

Objective of the optimization problem:

$$\Phi\Big((W_s, b_s)_{s=1}^L\Big) = \sum_{i=1}^{N}\sum_{j=1}^{K} l(f_{L_j}(x_i) - y_{ij}),$$

where $(x_i, y_i)_{i=1}^N$ is the training data and K the number of classes.

What can we say about the critical points of Φ, in particular when one layer k has more units n_k than the number of training points N?

Feature map at a wide layer

Notation: $F_k \in \mathbb{R}^{N \times n_k}$ is the output of the k-th layer

$$S_k = \left\{ (W_l, b_l)_{l=1}^{L} \middle| F_k, W_{k+2}, \ldots, W_L \text{ have full rank} \right\}.$$

Key technical result:

Lemma

If $n_k \geq N$ and the network is pyramidal from layer k on ($n_k \geq n_{k+1} \geq \ldots \geq n_L$), then the complement of S_k has Lebesgue measure zero.

Summary: If there exists a wide layer ($n_k \geq N$) then for almost all parameters, $(W_l, b_l)_{l=1}^{L}$, the feature map of the training data F_k at layer k is linearly independent.

Loss surface of CNNs and fully connected networks

Theorem (ICML 17/18)

If the condition $n_k \geq N$ holds for layer k, the network is pyramidal from layer k on and layer $k+1$ is fully connected, then

- *there exist infinitely many global minima in S_k with zero training error.*
- *every critical point in S_k is a global minimum with zero training error.*

Discussion:

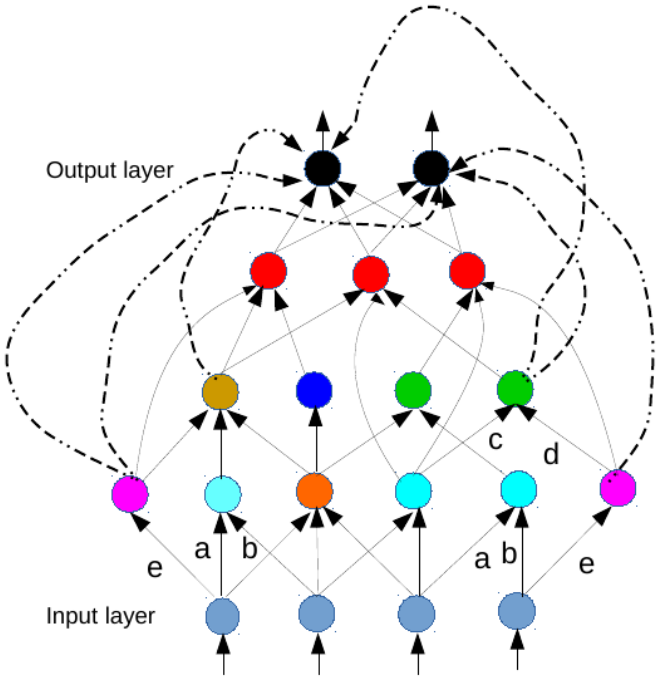
- suboptimal local minima can only exist for low rank weight matrices or if the feature map F_k has not full rank $\Longrightarrow$ one can argue (no rigorous proof) that such points do not exist
- the loss surface of over-parameterized neural networks is "easy"

Are practical networks that wide?

CNN Architecture	$M = \max_k n_k$	$M > N$
VGG(A-E)	3000K($k = 1$)	**yes**
InceptionV3	1300K($k = 3$)	**yes**
InceptionV4	1300K($k = 3$)	**yes**
SqueezeNet	1180K($k = 1$)	no
Enet	1000K($k = 1$)	no
GoogLeNet	800K($k = 1$)	no
ResNet	800K($k = 1$)	no
Xception	700K($k = 1$)	no

The maximum width of all layers in several state-of-the-art CNN architectures compared with the size of ImageNet dataset ($N \approx 1200K$). Some satisfy our condition and are extremely wide!

Neural networks with skip connections



Theorem: if there are more than N skip connections to the output layer, then for cross-entropy and squared loss

- there exist uncountably many global minima with zero training error
- there exist no suboptimal strict local minima

But wait... doesn't statistical learning theory tell us that we will overfit if we can fit everything?

The bias of stochastic gradient descent (SGD)

Zhang et al. (ICLR, 2017) showed that state-of-the-art neural networks achieve on CIFAR10 and ImageNet

- zero or close to zero training error on the initial data
- zero or close to zero training error on randomly flipped labels
- zero or close to zero training error on random inputs

Nevertheless they generalize well on the original data.

Why is there no overfitting?

- early stopping
- implicit regularization of SGD

Back to skip connections...

rand: Initialize weights up to output layer randomly, then use least squares to fit output layer $\Longrightarrow$ zero training error with probability 1.

CIFAR 10	**Sigmoid**	**Softplus**
VGG11	10	78.92
VGG11-skip (rand)	62.81 ± 0.39	64.49 ± 0.38
VGG11-skip (SGD)	$\mathbf{72.51} \pm 0.35$	$\mathbf{80.57} \pm 0.40$
VGG16	10	81.33
VGG16-skip (rand)	61.57 ± 0.41	61.46 ± 0.34
VGG16-skip (SGD)	$\mathbf{70.61} \pm 0.36$	$\mathbf{81.91} \pm 0.24$
Densenet121	**86.41**	**89.31**
Densenet121-skip (rand)	52.07 ± 0.48	55.39 ± 0.48
Densenet121-skip (SGD)	81.47 ± 1.03	86.76 ± 0.49

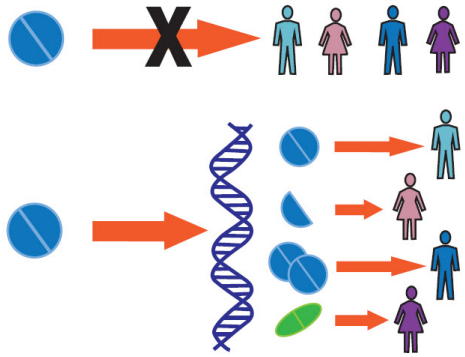
Test accuracy (%) of several CNN architectures with/without skip-connections on CIFAR10. **rand:** randomized feature map, **SGD:** full network trained with SGD.

Conclusion: Among the pool of solutions with zero training error, SGD selects one which generalizes well $\Longrightarrow$ implicit regularization of SGD.

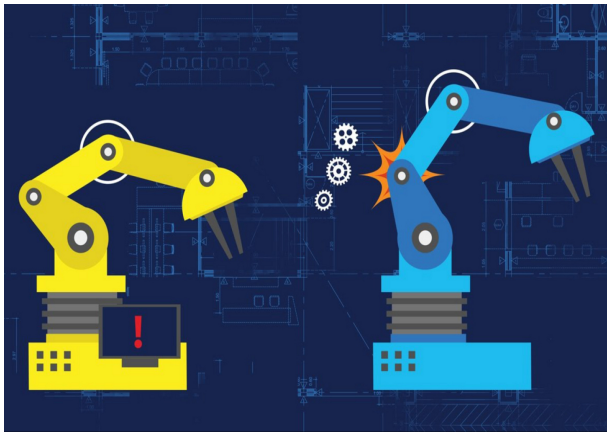
How can deep learning become robust?

Machine learning permeates industry and our society

Autonomous Driving



Personalized Medicine



Predictive Maintenance

All these applications of machine learning are safety-critical!

High requirements regarding safety and security

Lack of robustness against adversarial manipulation

Original x

Prediction: speed limit 70
99% confidence

$+$

δ

$\|\delta\|_2 = 0.67$

$=$

Adversarial Input $x + \delta$

Prediction: **speed limit 30**
99% confidence

- adversarial modification which is non-perceivable changes the decision
- high confidence in the wrong decision!

This behavior questions usage in safety critical systems!
Current classifiers less robust than visual system of humans?

Definition of adversarial input

Setting: K classes, input dimension d, classifier $f : \mathbb{R}^d \to \mathbb{R}^K$.
Input x is classified as $c = \arg\max_{j=1,\dots,K} f_j(x)$ (we assume this is correct).

Adversarial input: "Smallest" change δ such that the decision changes for $x + \delta$ (adversarial input).

$$\begin{aligned} &\min_{\delta \in \mathbb{R}^d} \ \|\delta\|_p \\ &\text{s.th.} \ \max_{l \neq c} \ f_l(x+\delta) \geq f_c(x+\delta) \\ &\qquad x + \delta \in C, \end{aligned}$$

where C is a constraint e.g., an image has to be in $[0,1]^d$.
Attention: change is only adversarial if "true" class has not changed.

Choice of p-norm has significant influence on structure of δ

Influence of distance measure on adversarial inputs

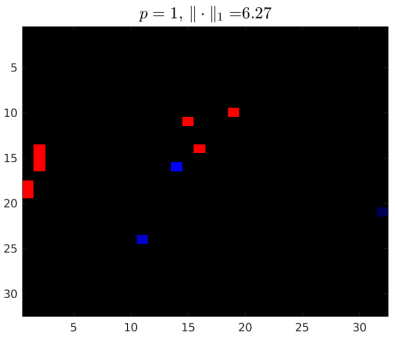
Original	δ_p	Adversarial Input
	$p=1$, $\|\cdot\|_1$=1.47	$p=1$, prediction: 8
	p=1, $\|\cdot\|_1 = 1.47$	Prediction: 8
Original Image, true class: 9	$p=2$, $\|\cdot\|_2$=0.30 Upscale: 5x	$p=2$, prediction: 8
	p=2, $\|\cdot\|_2 = 0.25$	Prediction: 8
	$p=\infty$, $\|\cdot\|_\infty$=0.02, Upscale: 10x	$p=\infty$, prediction: 8
	p=∞, $\|\cdot\|_\infty = 0.02$	Prediction: 8

Influence of distance measure on adversarial inputs II

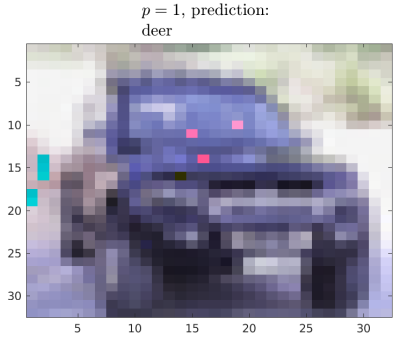
Original

δ_p

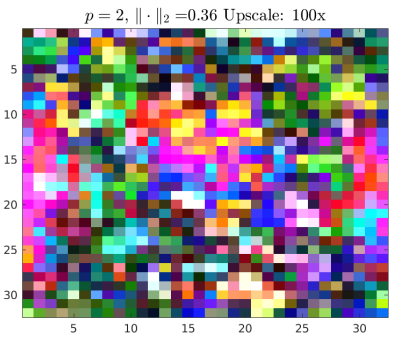
Adversarial Input



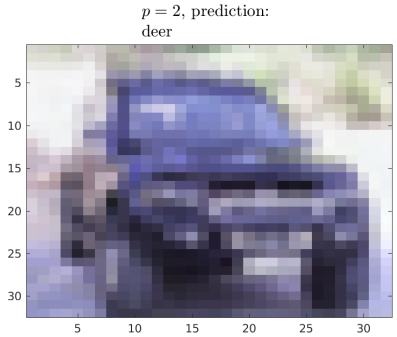
p=1, $\|\cdot\|_1 = 6.27$



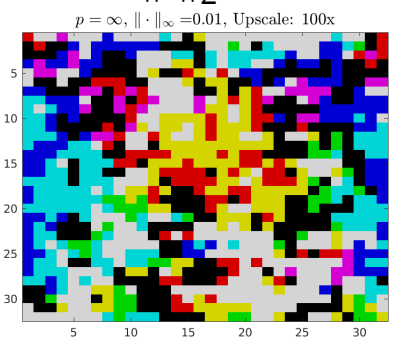
Prediction: deer



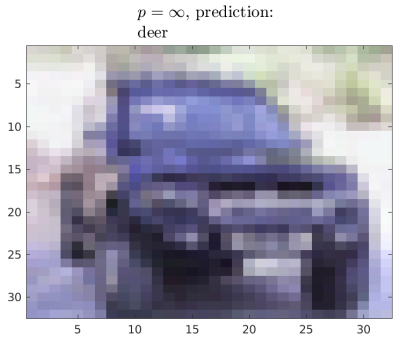
p=2, $\|\cdot\|_2 = 0.36$



Prediction: deer

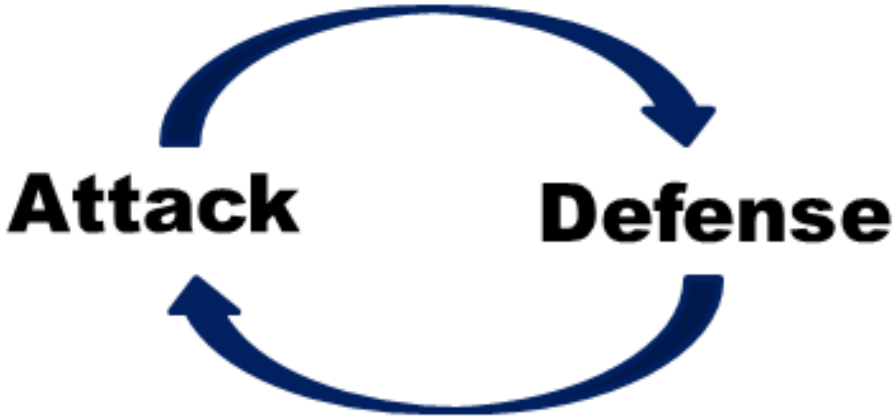


p=∞, $\|\cdot\|_\infty = 0.01$



Prediction: deer

Current state of the art



- **Attack:** come up with new ways how to modify the input
- **Defense:** add modified input during training (adversarial training)

Current approaches lead to "more robust" classifiers - but all approaches so far were broken again

Guarantees instead of "more robust" methods

For the use of machine learning in safety critical systems we need not just more robust methods, we need **formal guarantees** that the learned system **is** robust.

NIPS 2017 - formal guarantees on robustness

First formal guarantee: Let $f : \mathbb{R}^d \to \mathbb{R}^K$ be the classifier and let $x \in \mathbb{R}^d$ and $c = \underset{j=1,\ldots,K}{\arg\max}\, f_j(x)$

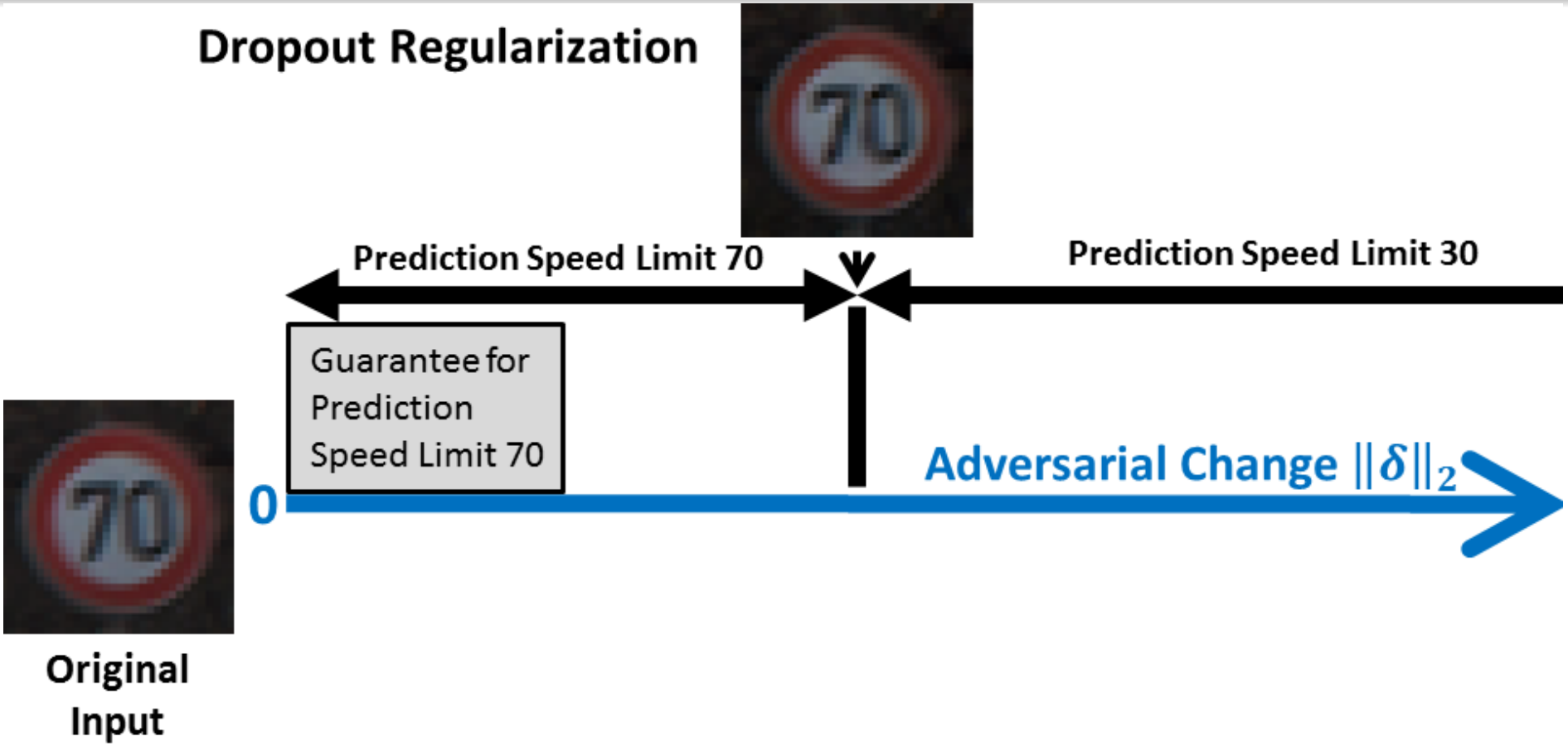
The classifier does not change its decision for $x + \delta$ if

$$\|\delta\|_2 \;\leq\; \max_{\alpha \geq 0} \; \min \left\{ \min_{j \neq c} \frac{f_c(x) - f_j(x)}{\max\limits_{y \in B(x,\alpha)} \|\nabla f_c(y) - \nabla f_j(y)\|_2}, \; \alpha \right\}.$$

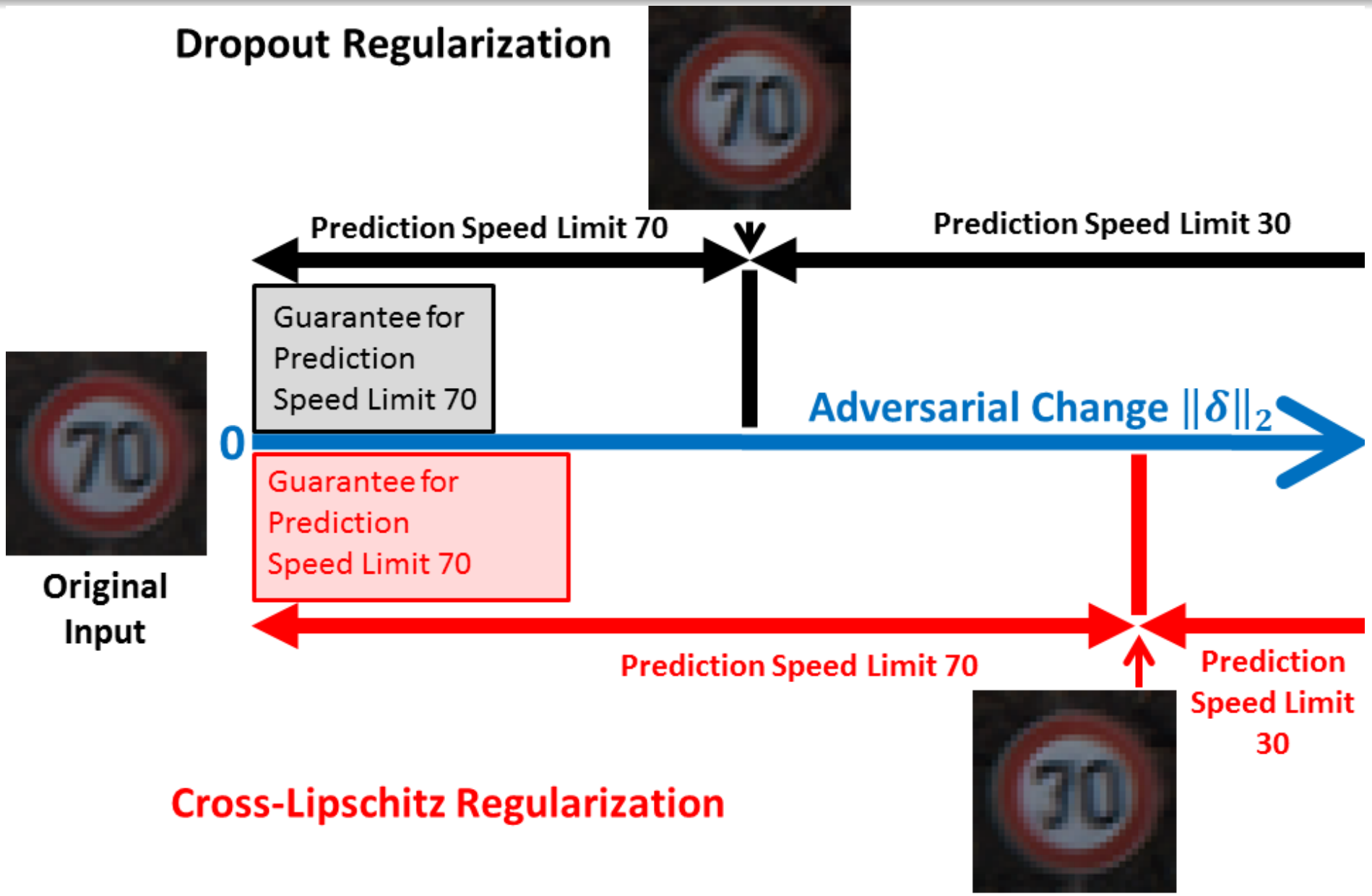
- evaluation of the bound for one hidden layer neural networks and kernel methods with Gaussian kernel
- the denominator motivates our new cross-Lipschitz regularization with the goal to maximize the guarantee

$$\Omega(f) = \frac{1}{nK^2} \sum_{i=1}^{n} \sum_{l,m=1}^{K} \|\nabla f_l(x_i) - \nabla f_m(x_i)\|_2^2 .$$

NIPS 2017 - quantitative results

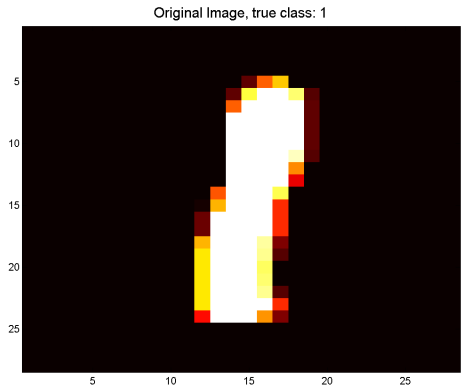


NIPS 2017 - quantitative results

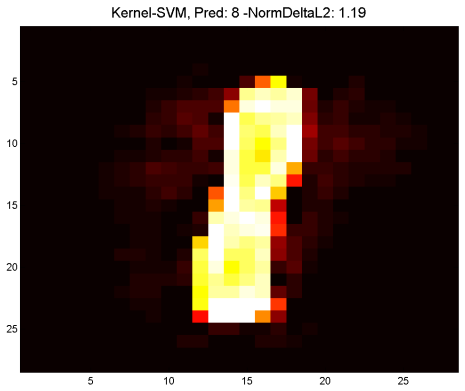


Cross-Lipschitz regularization improves robustness (guarantee)

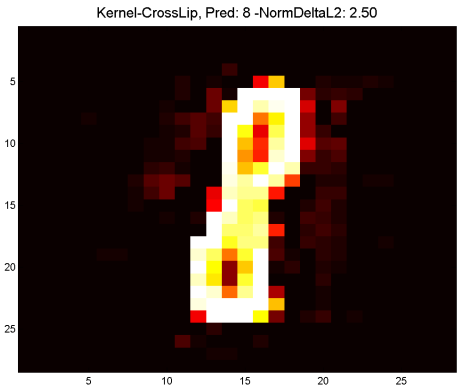
Adversarial inputs - illustration



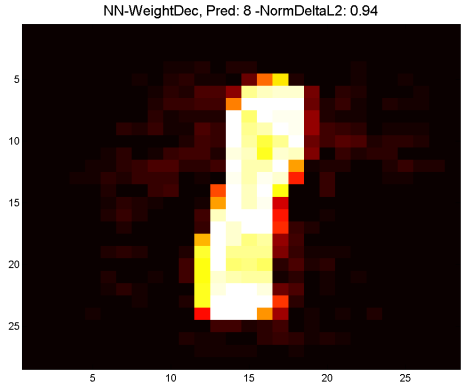
Original, Class 1



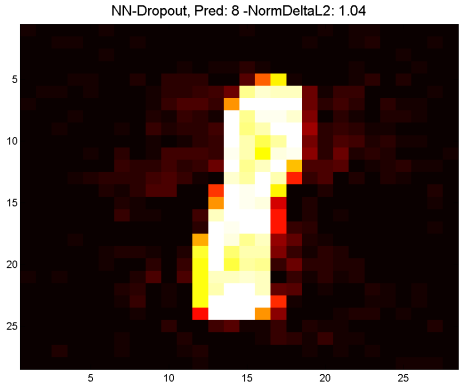
Kernel-SVM,
Pred:8, $\|\delta\|_2 = 1.2$



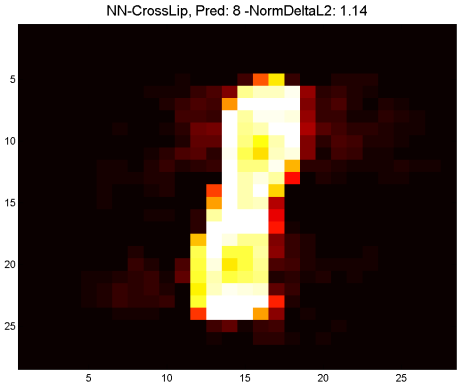
Kernel-CrossLipschitz,
Pred:8, $\|\delta\|_2 = 2.5$



NN-WeightDecay,
Pred:9, $\|\delta\|_2 = 0.9$



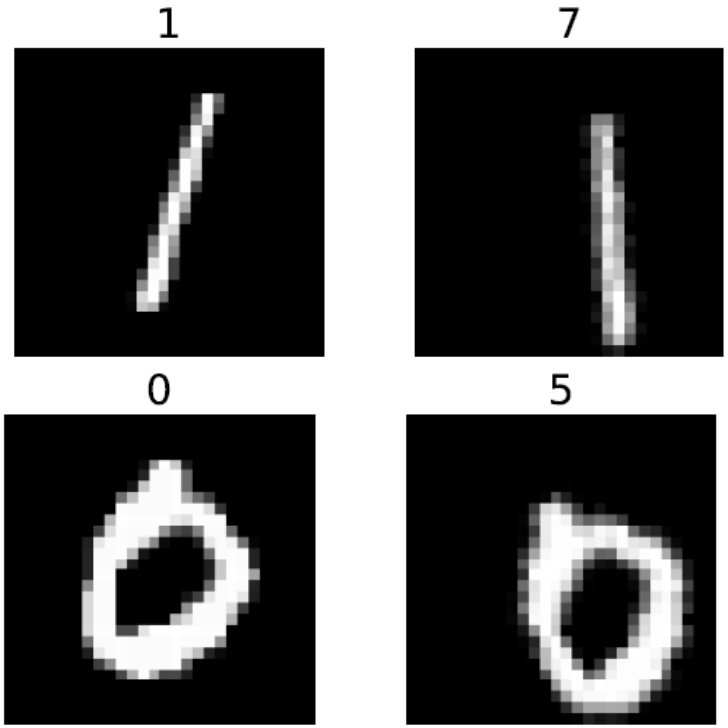
NN-Dropout,
Pred:8, $\|\delta\|_2 = 1.0$



NN-CrossLipschitz,
Pred:8, $\|\delta\|_2 = 1.1$

Attacks with generic transformations

Use of rotations and translations to produce adversarial examples:



Left: original image (correctly classified), **Right:** adversarially transformed image (taken from Engstrom et al, arXiv:1712.02779)

Open Problem:

Robustness guarantees against adversarial generic transformations

New criterion for choosing a classifier

Among different classifiers giving the same prediction performance choose the one with the better robustness guarantee!

A randomized gradient-free attack on ReLU networks

Key fact: ReLU networks produces piecewise affine functions
Adversarial input: "Smallest" change δ such that the decision changes for $x + \delta$ (adversarial input).

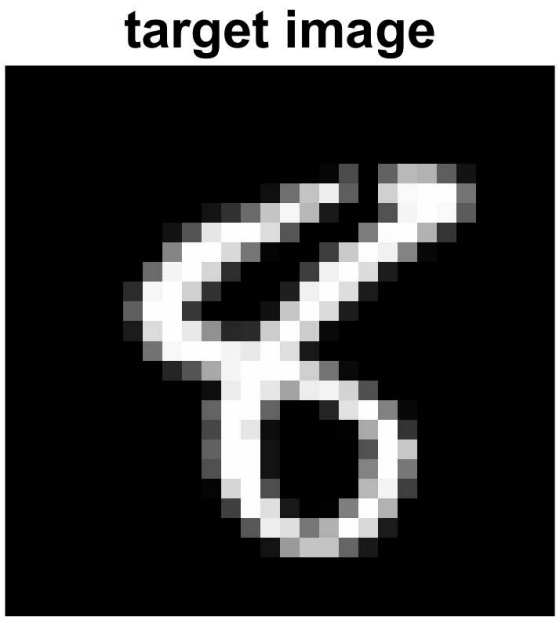
$$\begin{aligned} \min_{\delta \in \mathbb{R}^d} \ & \|\delta\|_p \\ \text{s.th.} \ & \langle w_l - w_c, x + \delta \rangle + b_l - b_c \geq 0, \\ & x + \delta \in C, \end{aligned}$$

where C is the intersection of $[0,1]^d$ and the region on which f is affine.

Observation: For each region where the ReLU network is affine, the problem for the computation of the adversarial input is convex.
Our attack: Randomly select region close to x and solve convex optimization problem $\Longrightarrow$ improves upon DeepFool or Carlini/Wagner attack by up to 10% for $p = 2$.

The attack in action on MNIST and CIFAR10

MNIST



CIFAR10

Visualization of attack scheme as it improves the attacks (and gets closer to the target image)

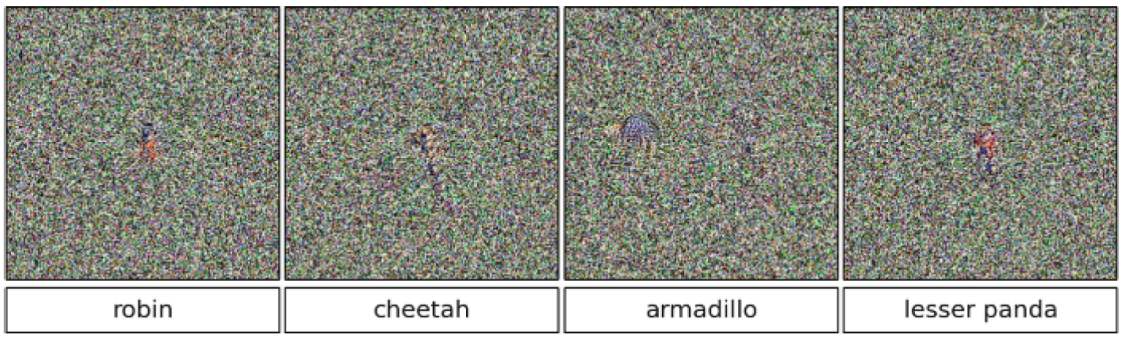
Conclusion and outlook

Conclusion:

- learning of overparameterized networks is "easy" but still lack of understanding of implicit regularization effect of SGD
- we need formal guarantees if machine learning is used in safety-critical applications
  $\Longrightarrow$ construct robust deep networks from scratch
- learning becomes more and more a multi-objective problem

Outlook:

- ensure low confidence predictions far away from the training data



taken from Nguyen et al, CVPR 2015, predictions with $\geq 99.6\%$ confidence