Intelligence Artificielle Avancée (RCP211)
Robustesse décisionnelle
Stabilité et généralisation

Nicolas Thome
Conservatoire National des Arts et Métiers (Cnam)
Laboratoire CEDRIC - équipe Vertigo

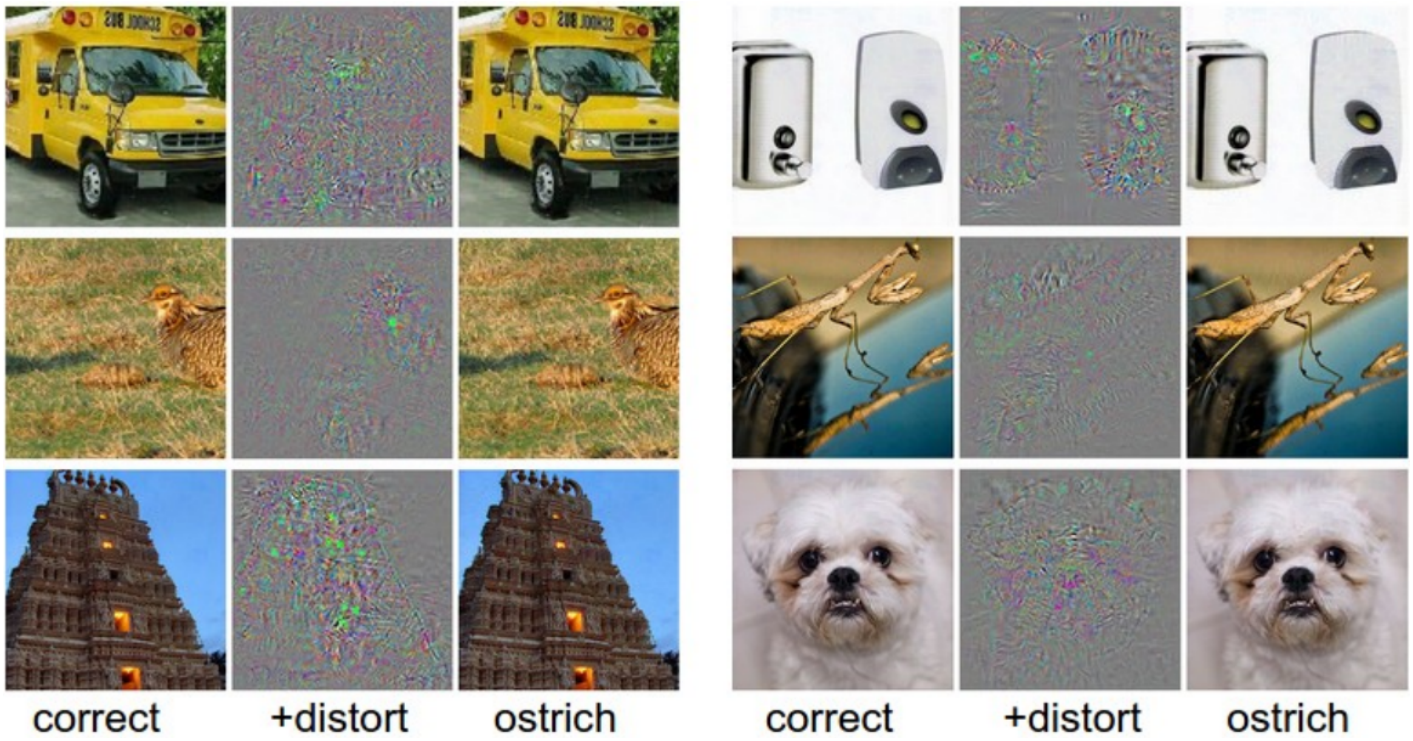
le cnam

Cédric

Outline

Deep Learning (DL) & Stability

- **Stability:** decision function with "controlled" variations
  - Small input variations $\Leftrightarrow$ reasonably small output variations on decision, *e.g.* Lipschitz property
  - **Decision function of deep Models not always stable**
    - **Ex: Adversarial Examples**



Deep Learning (DL) & Stability

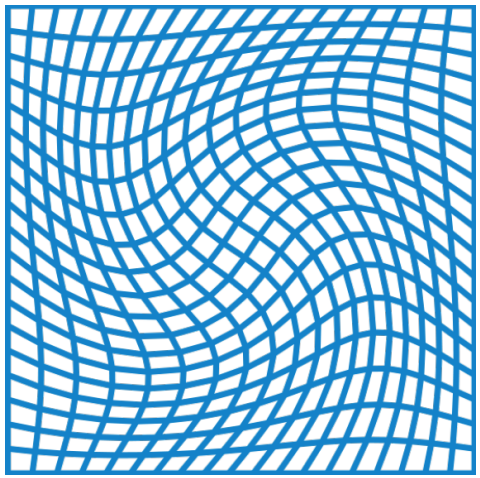
- Adversarial attacks in real-world

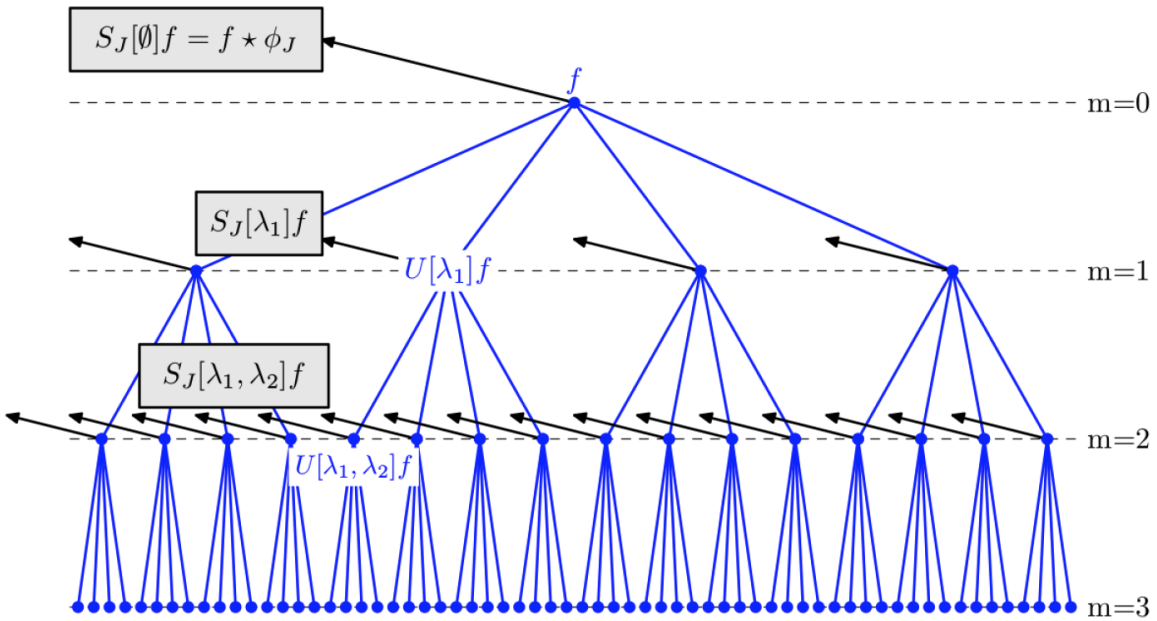
[Evtimov et al., 2017]

Deep Learning (DL) & Stability

Formal stability analysis of deep models

- Harmonic analysis in scattering operators [Mallat, 2012, Bruna and Mallat, 2013], *i.e.* "deep wavelets"
  - Show stability / invariance to diffeomorphisms
  - Stability bounds
- Generalized to deep kernel machines, closer to SoTA deep ConvNet architectures [Bietti and Mairal, 2017]

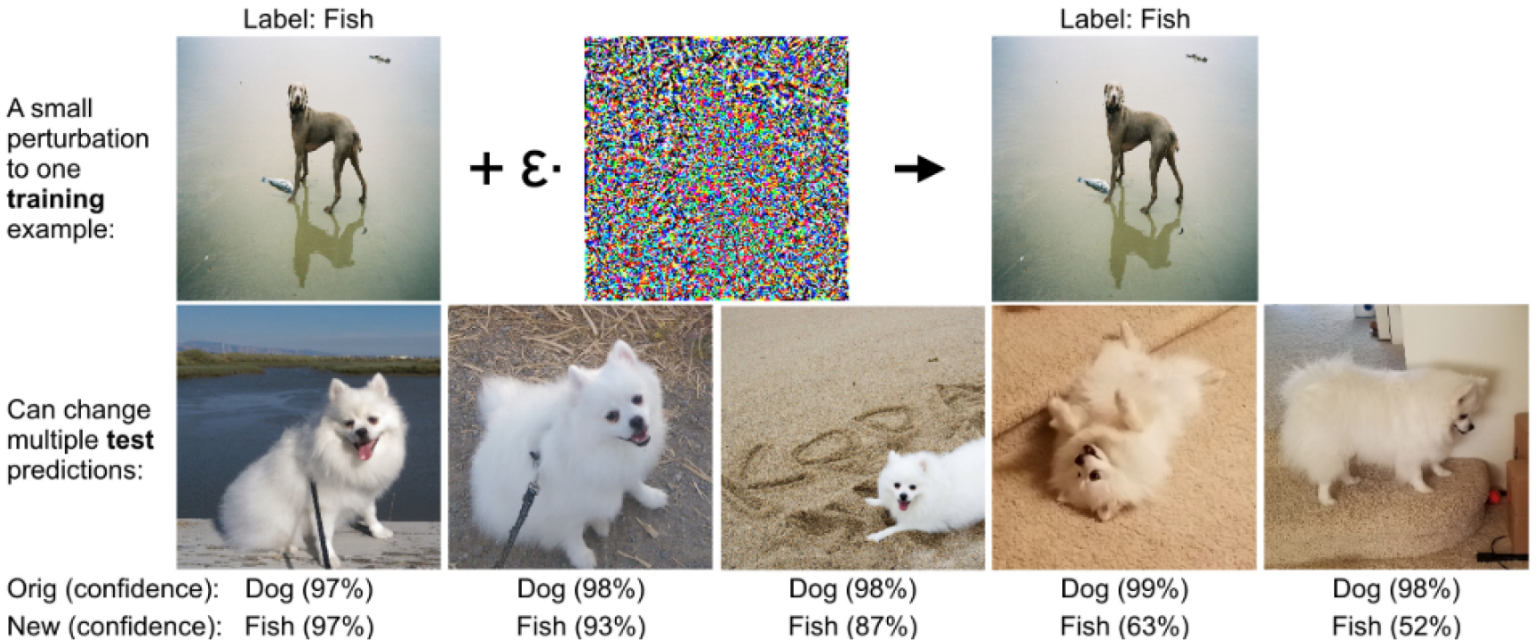




Deep Learning (DL) & Stability

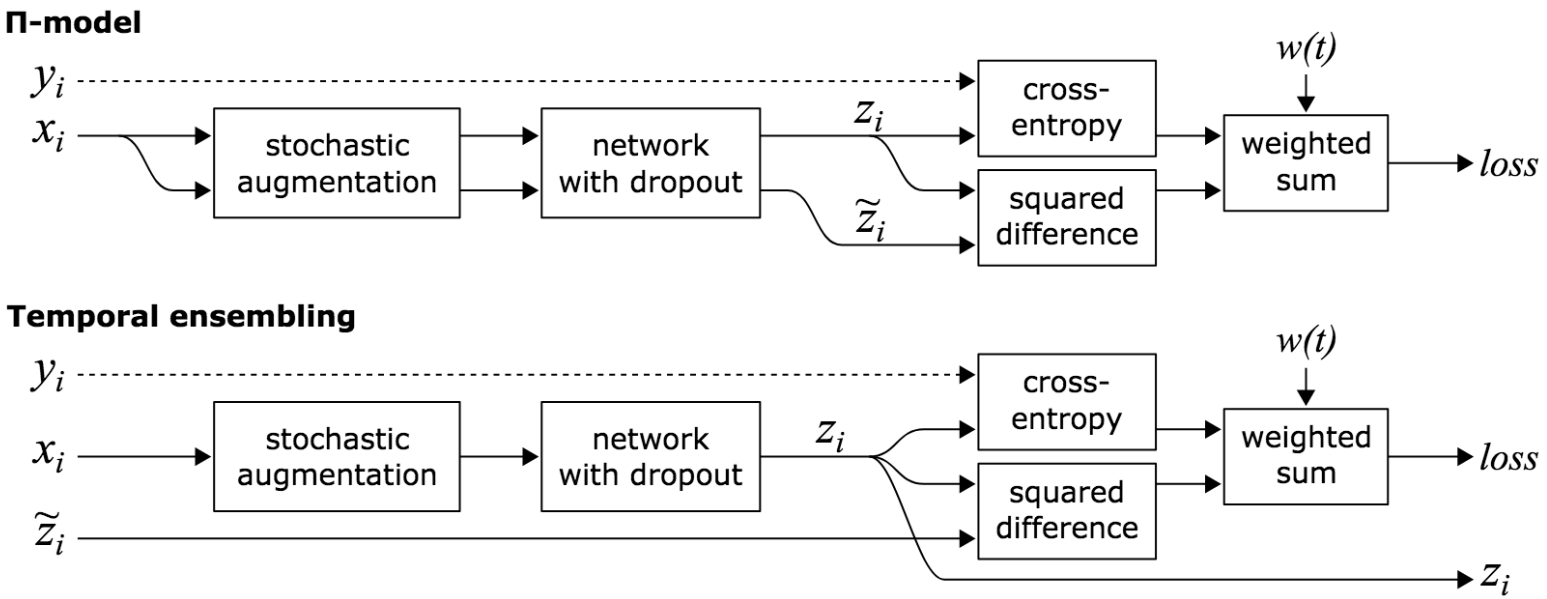
Formal stability analysis of deep models

- Influence Functions [Cook and Weisberg, 1980]
  - Characterize decision function influence on training examples
    - Removing a training point: $\mathcal{I}_{up,loss}(z, z_{test}) = -\nabla_\theta L(z_{test}, \hat{\theta})^T H_\theta^{-1} \nabla_\theta L(z, \hat{\theta})$
    - Perturbing it: $\mathcal{I}_{pert,loss}(z, z_{test})^T = -\nabla_\theta L(z_{test}, \hat{\theta})^T H_\theta^{-1} \nabla_x \nabla_\theta L(z, \hat{\theta})$
  - Adapted / applied to deep networks [Koh and Liang, 2017]
- Data poisoning



Ad hoc stability training

- Regularization criterion supporting learning stable decision function
  - Underlying model might not be stable, but helps to focus on a subet of stable functions of the family
- Robustness of the decision to transformations [Sajjadi et al., 2016], stability accross iterations [Laine and Aila, 2017, Tarvainen and Valpola, 2017]



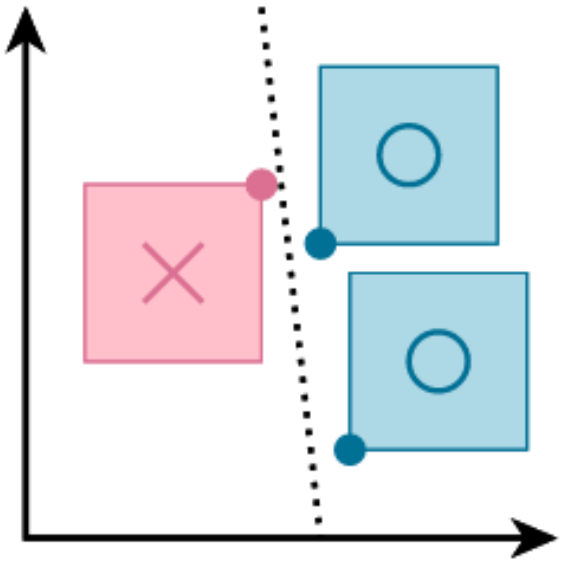
Formalization

- Training loss function: $\min_{\theta} \left[\frac{1}{N} \sum_{i=1}^{N} \ell(f_\theta(x_i), y_i^*) \right]$
  - θ model parameters, y_i^* GT supervision
- **Adversarial example** for sample x_i: $\max_{\delta \in \Delta} [\ell(f_\theta(x_i + \delta), y_i^*)]$
  - Δ: samples close do x_i, *e.g.* $\Delta := \{\delta \text{ s.t. } ||\delta|| < \epsilon\}$
- **Adversarial robustness**: $\min_{\theta} \left[\frac{1}{N} \sum_{i=1}^{N} \max_{\delta \in \Delta} \ell(f_\theta(x_i + \delta), y_i^*) \right]$
  - Training a model s.t. no adversarial example exists for each x_i
  - How to solve $\delta^*(x_i) = \arg\max_{\delta \in \Delta} [\ell(f_\theta(x_i + \delta), y_i^*)]$?
  - Danskin's theorem $\Rightarrow \min_{\theta} \left[\frac{1}{N} \sum_{i=1}^{N} \ell(f_\theta(x_i + \delta^*(x_i)), y_i^*) \right]$

Adversarial Robustness: linear models

Binary classification pb with linear models: $\ell(y_i^* \cdot w^T x_i)$

- Adversarial example: $\max_{||\delta|| < \epsilon} \left[\ell(y_i^* \cdot w^T (x_i + \delta)) \right]$
- ℓ convex wrt w (*e.g.* logistic loss) $\rightarrow \min_{||\delta|| < \epsilon} (y_i^* \cdot w^T (x_i + \delta))$
  - Ex with $||.||_\infty$: $\arg\min_{||\delta|| < \epsilon} (y_i^* \cdot w^T \delta) = -\epsilon y_i^* \text{sign}(w)$, $\min_{||\delta|| < \epsilon} (y_i^* \cdot w^T \delta) = -\epsilon ||w||_1$
- In general: $\min_{||\delta|| < \epsilon} (y_i^* \cdot w^T \delta) = -\epsilon ||w||_*$
  - $||w||_*$ dual norm of $|\delta||$ ($||.||_p$ and $||.||_q$ s.t. $\frac{1}{p} + \frac{1}{q} = 1$)



Credit: Kolter & Madry

Adversarial robust training loss

$$\min_\theta \left[\frac{1}{N} \sum_{i=1}^{N} \ell(y_i^* \cdot w^T x_i - \epsilon ||w||_*) \right]$$

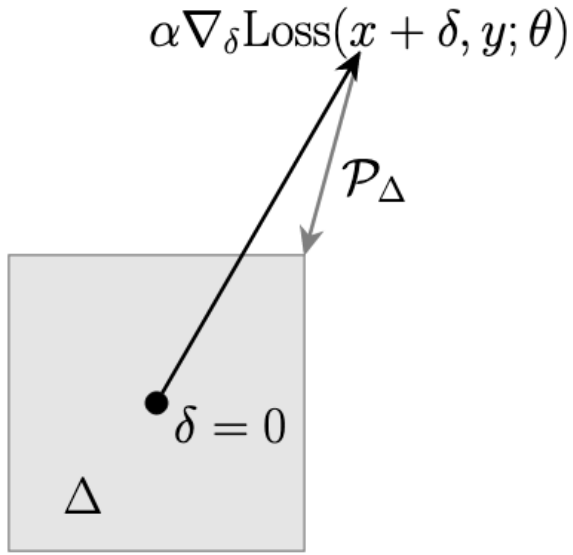
Adversarial Robustness: non-linear models

$$\delta^*(x_i) = \underset{\delta \in \Delta}{arg\ max} \left[\ell(f_\theta(x_i + \delta), y_i^*) \right]$$

- **No-closed form solution for f_θ non-linear!**
- Needs approximate solutions
- How these approximate solutions works in finding adversarial examples?
- How does it impact adversarial robust training?

Projected gradient methods

- Compute the gradient wrt input x: $\nabla_\delta \left[\ell(f_\theta(\mathrm{x}_i + \delta), \mathrm{y}_i^*)\right]$
- Do a gradient update: $\delta^{t+1} = \delta^t + \alpha \nabla_\delta \left[\ell(f_\theta(\mathrm{x}_i + \delta), \mathrm{y}_i^*)\right]$
- Project st modified δ^{t+1} remains in the admissible set $\|\delta^{t+1}\| < \epsilon$: $\mathcal{P}_\Delta\left(\delta^t + \alpha \nabla_\delta \left[\ell(f_\theta(\mathrm{x}_i + \delta), \mathrm{y}_i^*)\right]\right)$, *e.g.* with $\|.\|_\infty$: $\mathrm{Clip}(\delta^{t+1}, [-\epsilon, \epsilon])$



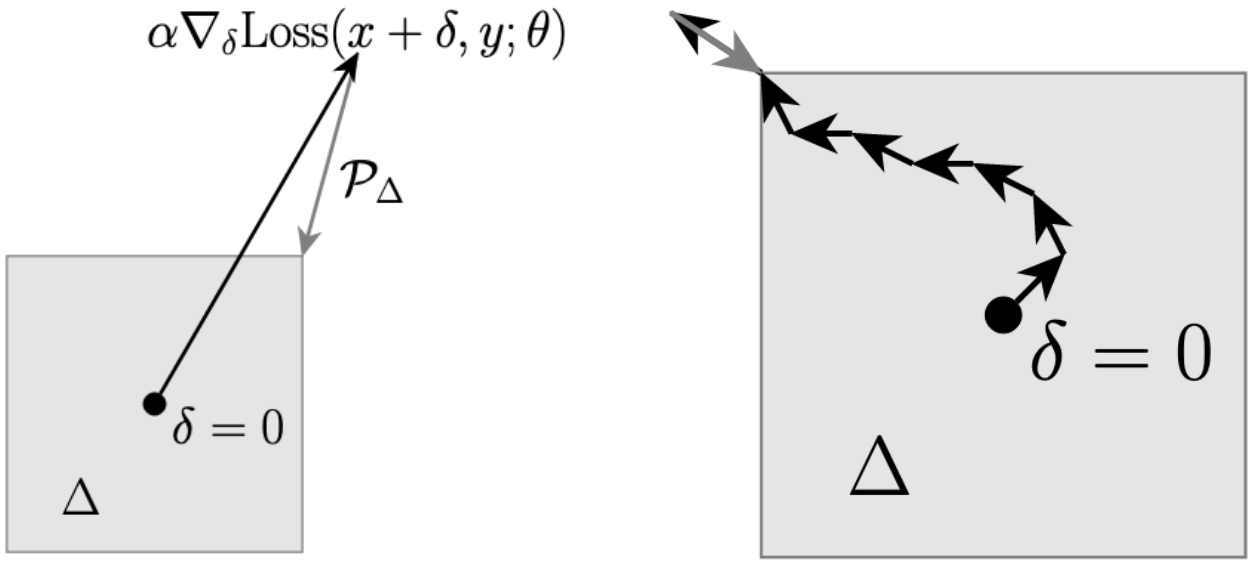
Credit: Kolter & Madry

- Local optimisation $\Rightarrow$ run from several init
- Gradient small at init $\Rightarrow$ normalized steepest descent

Projected gradient methods

$$\text{With } \|.\|_\infty\text{: Clip}\left(\left(\delta^t - \alpha\nabla_\delta\left[\ell(f_\theta(x_i+\delta), y_i^*)\right]\right), [-\epsilon, \epsilon]\right)$$

- "Large" a gradient update ($\alpha \to \infty$): reach corner of the $\|.\|_\infty$ constraint:
- $\delta^{t+1} = \epsilon\ \text{sign}(\nabla_\delta\left[\ell(f_\theta(x_i+\delta), y_i^*)\right]) \Rightarrow$ Fast Gradient Sign Method (FGSM) [Goodfellow et al., 2015]! (seminal work)
- Same update than for linear models: linear assumption
  - Projected gradient methods more accurate



Credit: Kolter & Madry

Formal verification via combinatorial optimization

- For and example x a target class $y_t \neq y^*$ (singe example here) :

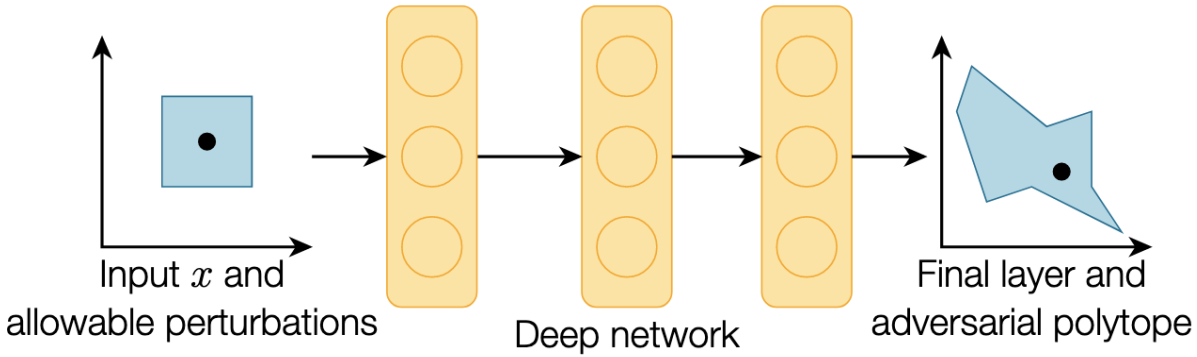
$$\min_{z_1,\dots,z_{d-1}} [z_d(y^*) - z_d(y_t)] \quad (1)$$

s.t. $\|z_1 - x\|_\infty < \epsilon$

$z_{i+1} = max\{0; W_i z_i + b_i\}, \ i \in \{1; d-2\}$

$z_d = W_{d-1} z_{d-1} + b_{d-1}$

- x input, $z_d(y^*)/z_d(y_t)$ logits for GT/target class
- $z_1 = x + \delta$ perturbed input
- z_i, $i \in \{2; d-1\}$ hidden layers
- z_d logit layer



Credit: Kolter & Madry

- Solving (1): if value >0, no adversarial example (formal proof)

Formal verification via convex relaxations

$$\min_{z_1,\ldots,z_{d-1}} \left[z_d(y^*) - z_d(y_t)\right]$$
$$\text{s.t. } \|z_1 - x\|_\infty < \epsilon$$
$$z_{i+1} = max\left\{0; W_i z_i + b_i\right\}, \; i \in \{1; d-2\}$$
$$z_d = W_{d-1} z_{d-1} + b_{d-1}$$

- $max\left\{0; W_i z_i + b_i\right\}$ non linear constraint
- convert to a (binary) mixed integer linear program (MILP):

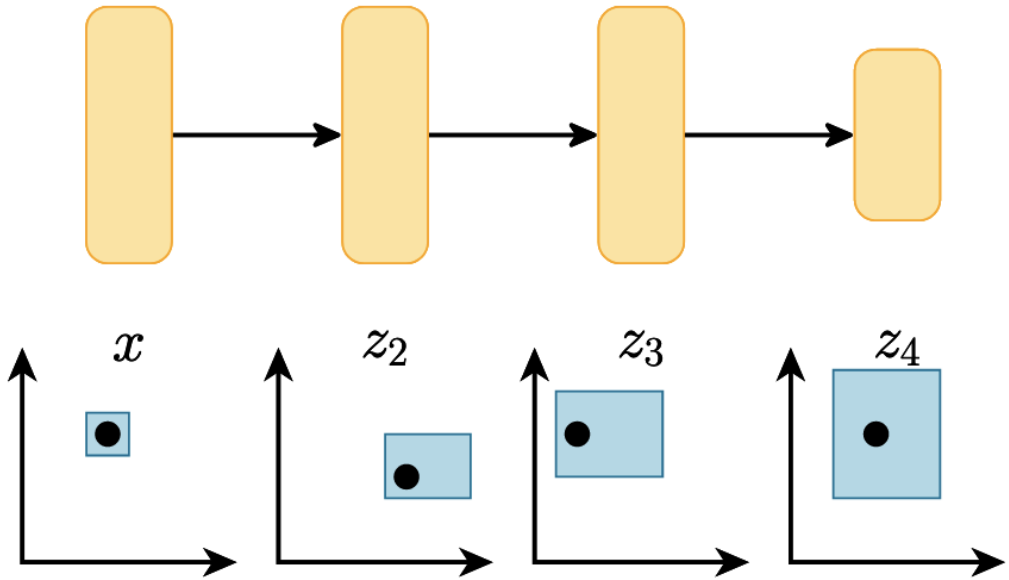
$$z_{i+1} \geq W_i z_i + b_i$$
$$z_{i+1} \geq 0$$
$$u_i \cdot v_i \geq z_{i+1}$$
$$W_i z_i + b_i \geq z_{i+1} + (1 - v_i) l_i$$
$$v_i \in \{0; 1\}^{|v_i|}$$

- u_i upper bound on $W_i z_i + b_i$
- l_i lower bound on $W_i z_i + b_i$

Formal verification via convex relaxations

- MILP solved with off-the shelf solvers (CPLEX, Gurobi)
- **BUT:** efficiency strongly depends on ii problem structure (*e.g.* ϵ value) and having tight bounds (u_i, l_i) on $W_i z_i + b_i$
- $(Wz + b)_k = \sum_j w_{kj} z_j + B_k$, assume $z_j \in [\hat{l}, \hat{u}]$
  - if $w_{kj} > 0$, $\hat{l}W + b < (Wz + b)_k < \hat{u}W + b$
  - if $w_{kj} < 0$, $\hat{u}W + b < (Wz + b)_k < \hat{l}W + b$

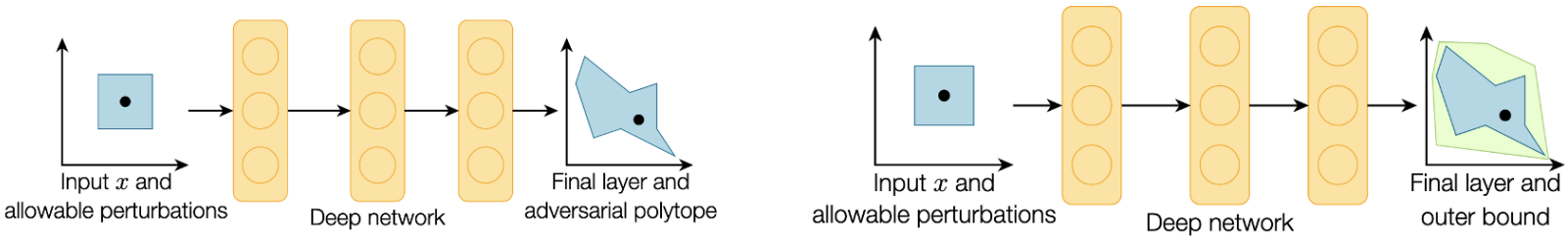
$$max(0, W)\hat{l} + min(0, W)\hat{u} + b \leq (Wz + b) \leq max(0, W)\hat{u} + min(0, W)\hat{l} + b$$



Credit: Kolter & Madry

Formal verification via convex relaxations

- MILP solvers: verified proof that adversarial examples exist or not
- Efficient only for few layers with limited size (10-50)
- **Option:** convex relaxation
  - Basically: let the binary variable $v_i \in \mathbb{R}$
  - Can still be used for verified adversarial robustness: no adversarial example with relaxed formulation $\Rightarrow$ no adversarial example in initial problem

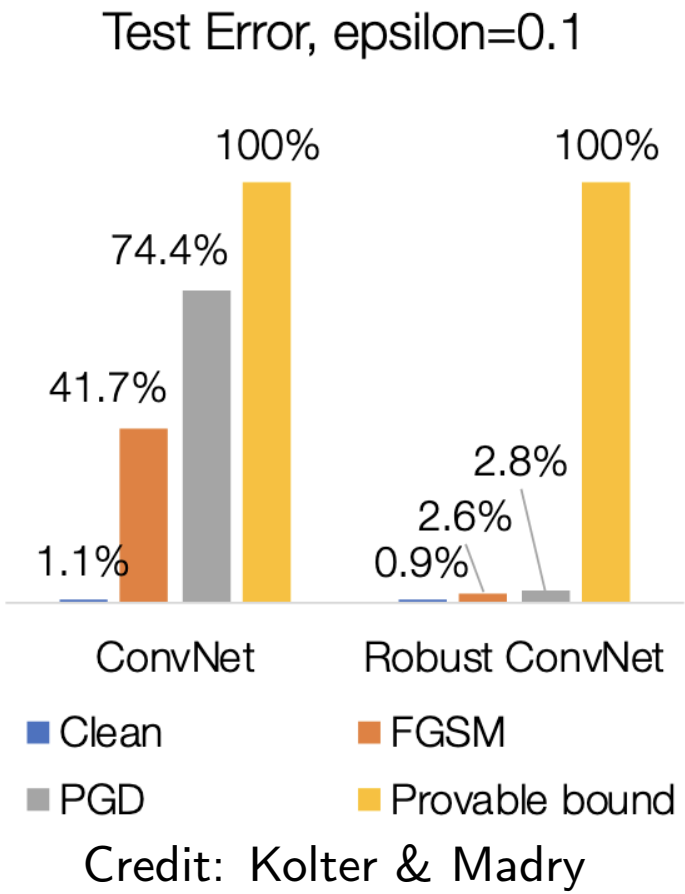


Credit: Kolter & Madry

Adversarial training in practice

Example with a ConvNet on MNIST

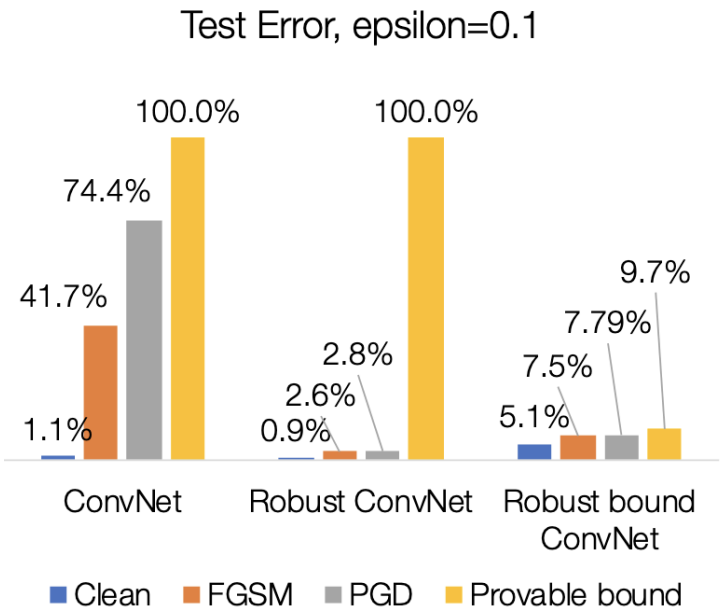
- Projected Gradient Method (PGM): works reasonably well



Adversarial training in practice

Example with a ConvNet on MNIST

- Convex relation for combinatorial optimization
  - Does not work well as it
  - Can be improved by minimizing the bounds during training (differentiable bounds)



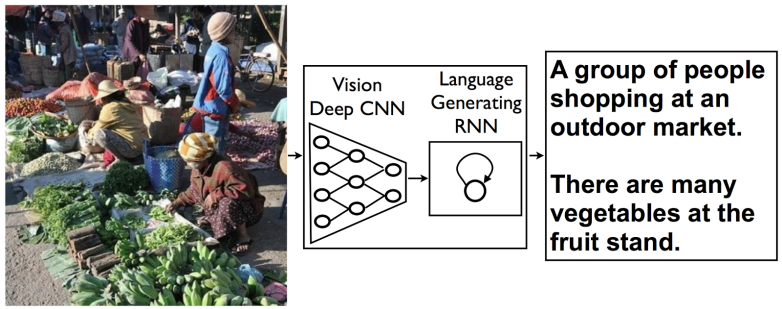
Credit: Kolter & Madry

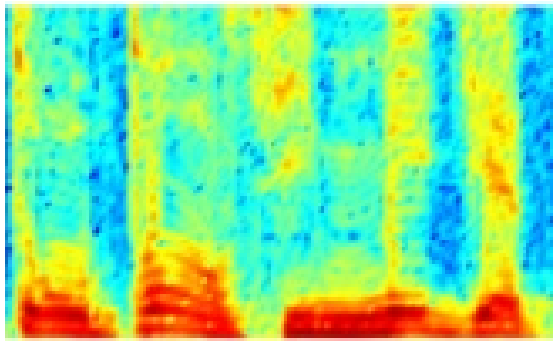
Adversarial training in practice

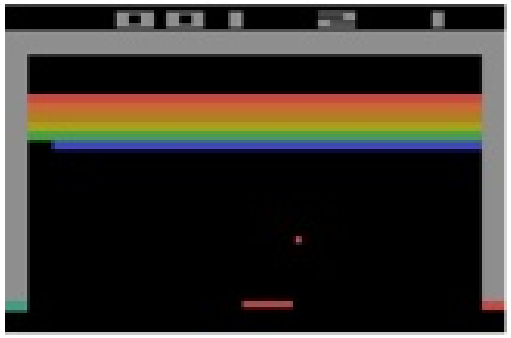
- Beyond MNIST: performances of adversarially trained deep models still an open question
- Adapting architecture for robust training?
- Adversarial training for generalization?

Outline

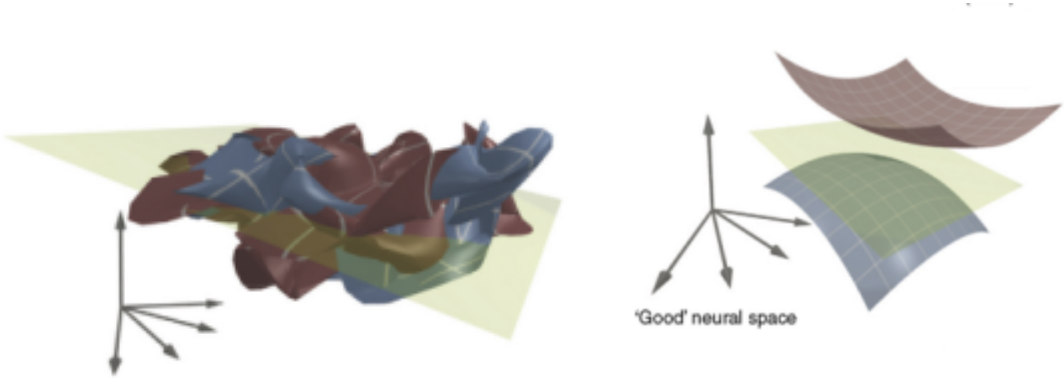
Deep Learning Theory





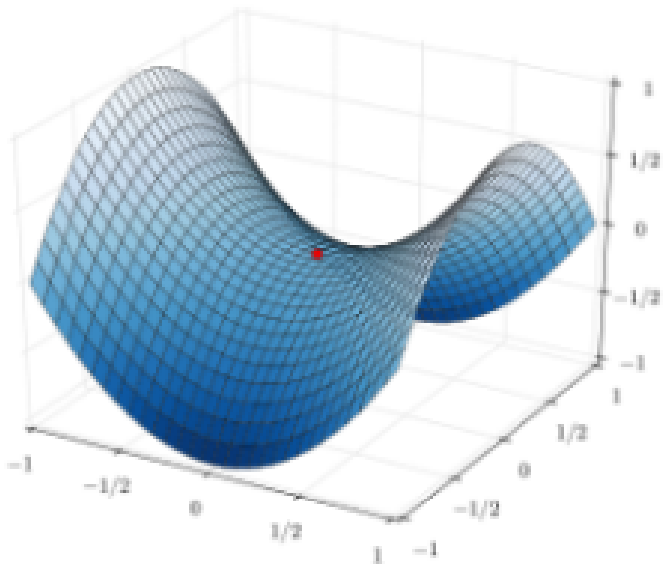


- Deep Learning: huge impact in terms of experimental results
- BUT: formal understanding still limited, other robustness issues
  - Optimization: non-convex problem
  - Generalization & over-fitting



Non-Convex Optimization

- One of the main historical shortcoming of deep neural networks
- In pratice, not really an issue with modern neural networks, WHY?
- Some preliminary answer elements:
  - In high dimension, few local minima
    but many saddle points [Dauphin et al., 2014]
    - Empirically, gradient descent methods manage to
      escape [Goodfellow and Vinyals, 2015] saddle points



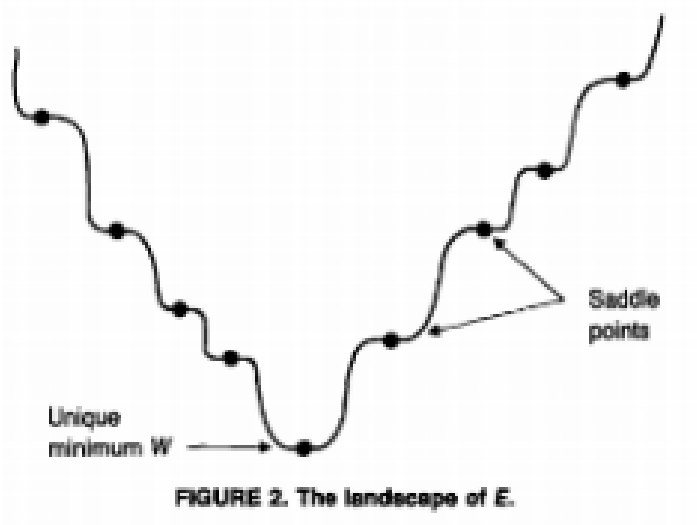
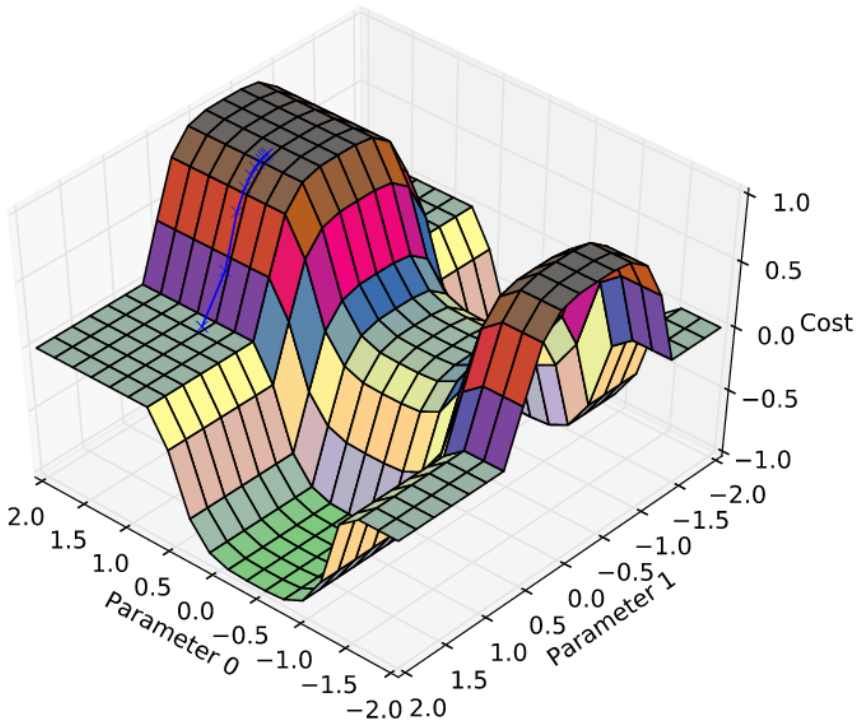


FIGURE 2. The landscape of E.

Non-Convex Optimization

- WHY non-convex optimization ist not a major practical issue for deep learning?
- Some preliminary answer elements:
  - Most of local minima have about the same objective value [Haeffele and Vidal, 2015, Choromanska et al., 2014]

(Cartoon of Dauphin et al 2014's worldview)

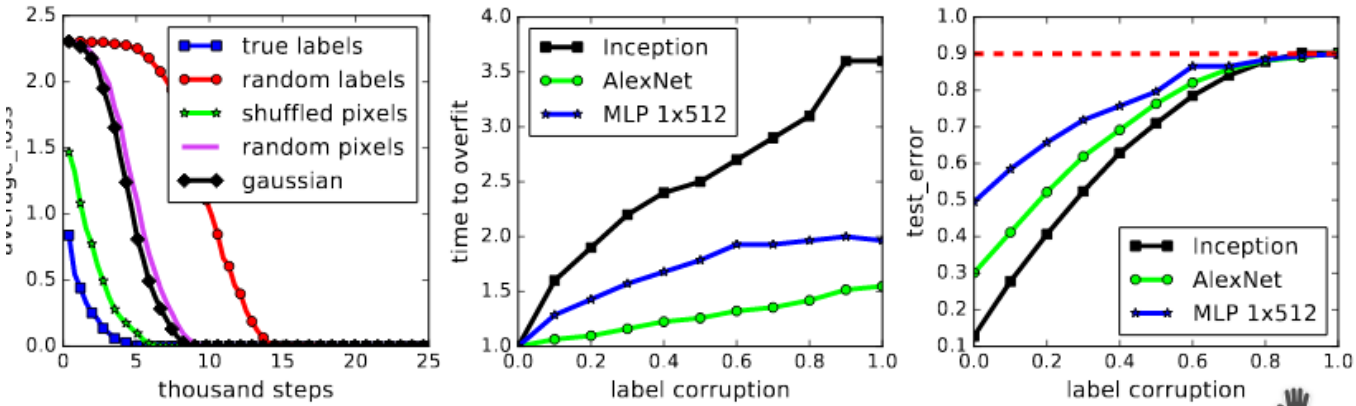


Deep Learning and generalization

- Rademacher complexity: capacity of a model to fit random label :

$$\mathcal{R}_n(\mathcal{H}) = E_\sigma \left[sup_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \sigma_i h(x_i) \right]$$

- Rethinking generalization: Zhang *et. al.* ICLR17 [Zhang et al., 2017]



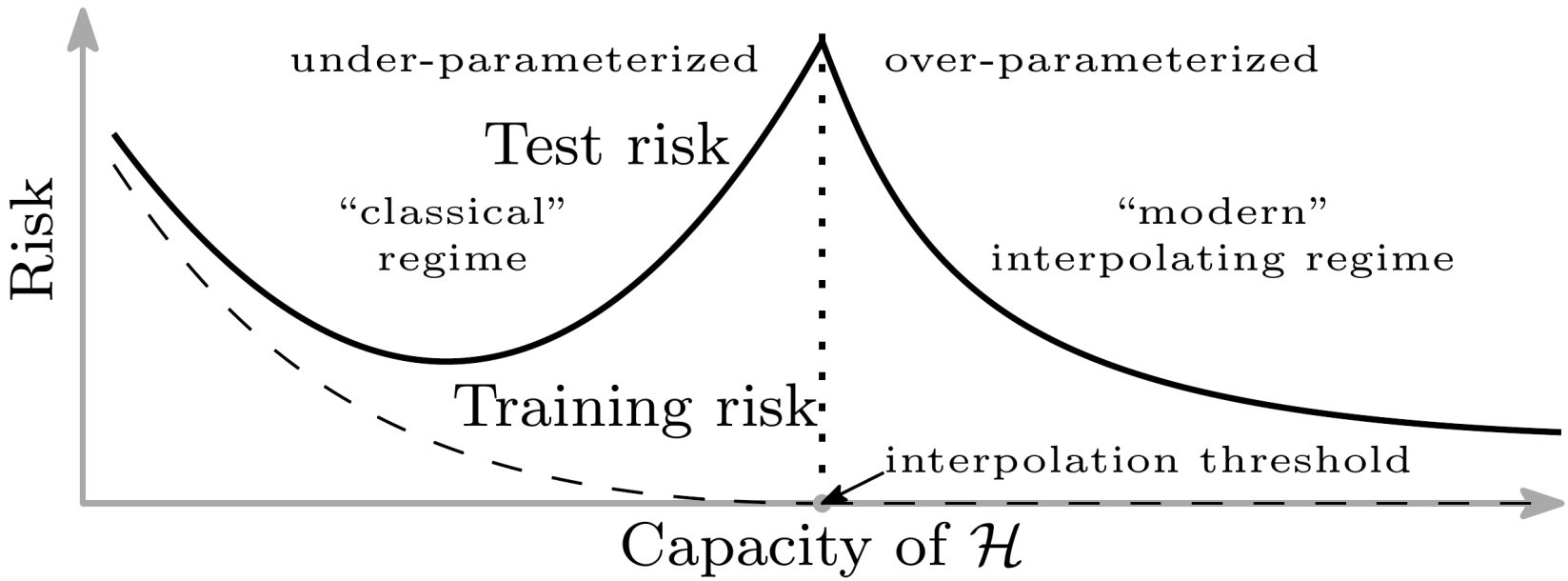
(a) learning curves (b) convergence slowdown (c) generalization error growth

  - Deep models easily fits random labels !!
  - $\mathcal{R}_n(\mathcal{H}) \approx 1 \Rightarrow$ no theoretical guarantee on generalization performances

- **Classical learning theory insufficient to explain the good generalization behavior of deep models**

Generalization and over-parametrized models

- Double U-curve phenomena observed with deep models! [Belkin et al., 2019]



Double descent illutration

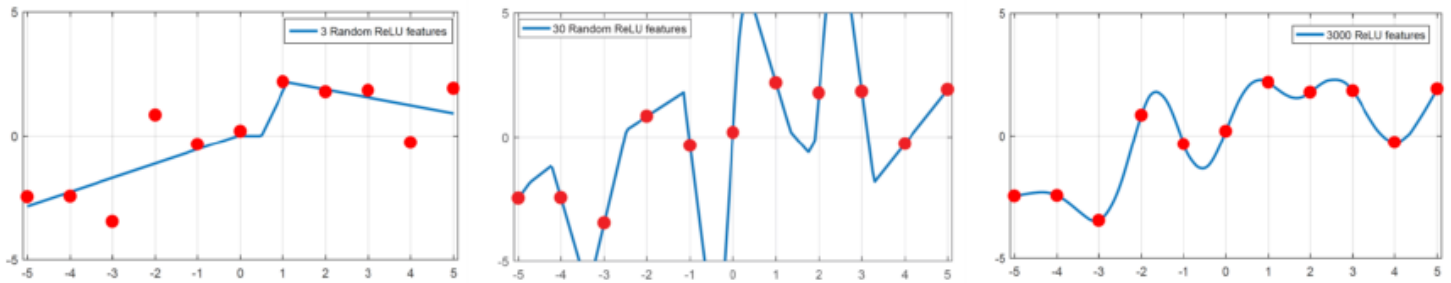


Figure 6: Illustration of double descent for Random ReLU networks in one dimension. Left: Classical under-parameterized regime (3 parameters). Middle: Standard over-fitting, slightly above the interpolation threshold (30 parameters). Right: "Modern" heavily over-parameterized regime (3000 parameters).

References I

[Belkin et al., 2019] Belkin, M., Hsu, D., Ma, S., and Mandal, S. (2019).
Reconciling modern machine-learning practice and the classical bias–variance trade-off.
Proceedings of the National Academy of Sciences, 116(32):15849–15854.

[Bietti and Mairal, 2017] Bietti, A. and Mairal, J. (2017).
Invariance and stability of deep convolutional representations.
In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30*, pages 6210–6220. Curran Associates, Inc.

[Bruna and Mallat, 2013] Bruna, J. and Mallat, S. (2013).
Invariant scattering convolution networks.
IEEE Trans. Pattern Anal. Mach. Intell., 35(8):1872–1886.

[Choromanska et al., 2014] Choromanska, A., Henaff, M., Mathieu, M., Arous, G. B., and LeCun, Y. (2014).
The loss surface of multilayer networks.
CoRR, abs/1412.0233.

[Cook and Weisberg, 1980] Cook, R. and Weisberg, S. (1980).
Characterizations of an empirical influence function for detecting influential cases in regression.
Technometrics, 22(4):495–508.

[Dauphin et al., 2014] Dauphin, Y., Pascanu, R., Gülçehre, Ç., Cho, K., Ganguli, S., and Bengio, Y. (2014).
Identifying and attacking the saddle point problem in high-dimensional non-convex optimization.
CoRR, abs/1406.2572.

[Evtimov et al., 2017] Evtimov, I., Eykholt, K., Fernandes, E., Kohno, T., Li, B., Prakash, A., Rahmati, A., and Song, D. (2017).
Robust physical-world attacks on machine learning models.
CoRR, abs/1707.08945.

[Goodfellow et al., 2015] Goodfellow, I. J., Shlens, J., and Szegedy, C. (2015).
Explaining and harnessing adversarial examples.
In *ICLR (Poster)*.

References II

[Goodfellow and Vinyals, 2015] Goodfellow, I. J. and Vinyals, O. (2015).
Qualitatively characterizing neural network optimization problems.
In *ICLR*.

[Haeffele and Vidal, 2015] Haeffele, B. D. and Vidal, R. (2015).
Global optimality in tensor factorization, deep learning, and beyond.
CoRR, abs/1506.07540.

[Koh and Liang, 2017] Koh, P. W. and Liang, P. (2017).
Understanding black-box predictions via influence functions.
In Precup, D. and Teh, Y. W., editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1885–1894, International Convention Centre, Sydney, Australia. PMLR.

[Laine and Aila, 2017] Laine, S. and Aila, T. (2017).
Temporal ensembling for semi-supervised learning.
In *International Conference on Learning Representations (ICLR)*.

[Mallat, 2012] Mallat, S. (2012).
Group invariant scattering.
Communications in Pure and Applied Mathematics, 10:1331–1398.

[Sajjadi et al., 2016] Sajjadi, M., Javanmardi, M., and Tasdizen, T. (2016).
Regularization with stochastic transformations and perturbations for deep semi-supervised learning.
In *Advances in Neural Information Processing Systems (NIPS)*.

[Tarvainen and Valpola, 2017] Tarvainen, A. and Valpola, H. (2017).
Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results.
In *Advances in Neural Information Processing Systems (NIPS)*.

[Zhang et al., 2017] Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. (2017).
Understanding deep learning requires rethinking generalization.