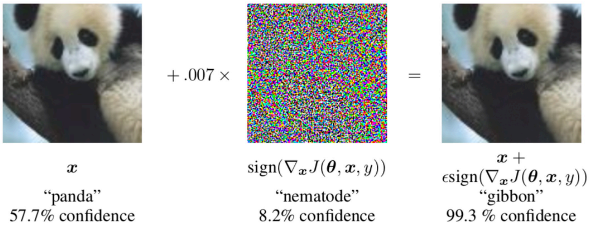
8

Adversarial Learning

In this lecture, we will discuss adversarial attack & defense for the neural network. We will give mathematical descriptions about attack and defense, and discuss the current trend in this field.

8.1 Introduction to Adversarial Learning

Motivation The earliest research on the adversarial attack is on (Goodfellow *et al.*, 2015a) and (Szegedy *et al.*, 2014), where (Szegedy *et al.*, 2014) shows that a small but specified designed perturbation of image changes the prediction of a neural net, while (Goodfellow *et al.*, 2015a) presents an example that a panda image with a small perturbation still looks like the same for humans, but it is classified as a gibbon by the neural network.



$\boldsymbol{x}$
“panda”
57.7% confidence
$+ .007 \times$
$\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
“nematode”
8.2% confidence
$=$
$\boldsymbol{x} +$
$\epsilon\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
“gibbon”
99.3 % confidence

Remark 8.1. Some comments on this adversary example:

1. An adversary attack is usually performed on several well-known deep learning models, such as *GooLeNet*.

2. The performance of an adversary attack is the *Robust Error*, i.e., the proportion of data samples that can be effectively attacked by our attacking method. In (Goodfellow *et al.*, 2015a), the robust error was 87.5% on the CIFAR-10 dataset, which was later increased to 100% by the *C&W* attack in (Carlini and Wagner, 2017). Therefore, each image can be easily attacked.

3. Transferability issue: adversarial examples designed for one kind of neural-nets can attack other kinds of neural-nets.

4. The targetted attack is also easy, i.e., we can manipulate the prediction to whatever target we want.

5. It is hard to defend.

6. A good attack can achieve

   - high robust error;
   - small number of queries of the model;
   - low magnitude of perturbation.

Type of attacks Based on whether the hacker knows the model or not, adversarial attacks can be classified as:

- White box attack: the hacker can access everything of the victim model;

- Black box attack: the hacker can only access top-k confidence scores or labels.

Based on whether the attack is aimed to manipulate the prediction, adversarial attacks can be classified as:

- Targetted attack: an example in class-A is classified into class-B;

- Untargeted attack: an example in class-A is classified into a class other than A.

Based on what kinds of data the hacker can access, adversarial attacks can be classified as:

- Evasion attack: the hacker can change testing data. We focus on this kind of attack in this lecture

- Poison attack: the hacker can change the training data.

Bibliography In 2014, the papers (Goodfellow *et al.*, 2015a) and (Szegedy *et al.*, 2014) were the first ones that introduced white-box attacks; In 2016, the papers (Carlini and Wagner, 2017) introduced black-box attacks; In 2017, (Chen *et al.*, 2017) first performed the black box attacks by zeroth-order optimization methods; In 2018, (Ilyas *et al.*, 2018) introduced the zeroth-order attacks by querying the objective as less as possible.

8.2 Mathematical Formulation of Adversary Attack

Consider the supervised setting, i.e., given the data points $\{(x_i, y_i)\}_{i=1}^n$, the modeler wants to generate a neural network f_θ such that $f(\theta; x_i) \approx y_i$ for each i. Now we give various formulations for different types of attacks.

8.2.1 Un-targetted Attack

The goal of an untargeted attacker is that given a data instance x, the perturbed input $x + \delta$ will make the neural-nets learn a wrong model. This gives an optimization formulation:

$$\begin{aligned} \min_{\delta \in \mathbb{R}^d} \quad & \|\delta\|_p && (8.1a) \\ \text{s.t.} \quad & f(x+\delta) \neq y && (8.1b) \\ & x + \delta \in [0,1]^d && (8.1c) \end{aligned}$$

where (8.1a) is to make the energy of the pertubation as small as possible; (8.1b) is to mislead the model; (8.1c) is to make the perturbed input well-defined. The inequality constraint (8.1b) makes the problem hard to solve. Thus we often reformulates this problem as

$$\begin{aligned} \max \quad & J(f(x+\delta), y) && \text{(8.2a)} \\ \text{s.t.} \quad & \|\delta\|_p \leq \epsilon && \text{(8.2b)} \\ & x + \delta \in [0,1]^d && \text{(8.2c)} \end{aligned}$$

where $J(\cdot, \cdot)$ is some loss function; the objective (8.2a) is to maximize the loss between output for perturbed data and original one; the constraint (8.2b) is to keep the perturbation in a small magnitude ε. The decision variable in this problem is the input perturbation δ instead of parameters in the neural-nets f.

The *fast gradient sign method* (FGSM) (Goodfellow *et al.*, 2015a) can be used to solve this optimization problem. First linearize the objective (8.2a) and project it into the norm constrained set, we imply that the update should be

$$\eta = \epsilon \cdot \operatorname{sign}(\nabla_x J(f(x), y)),$$

and the adversarial attack is performed by setting $\tilde{x} = x + \eta$.

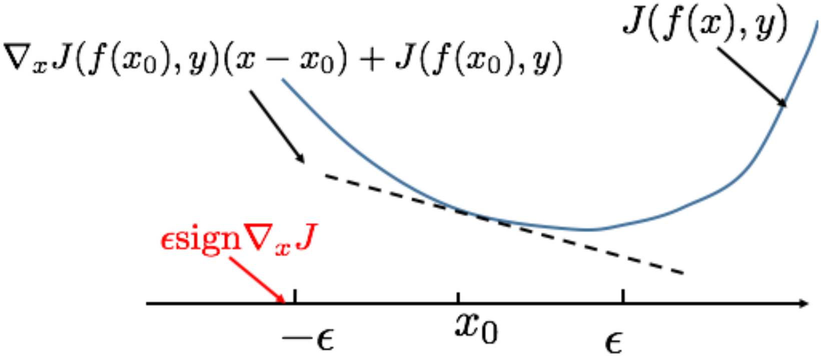

$J(f(x), y)$
$\nabla_x J(f(x_0), y)(x - x_0) + J(f(x_0), y)$
$\epsilon \mathrm{sign} \nabla_x J$
$-\epsilon$
x_0
ϵ

Figure 8.1: Illustration for Fast gradient sign method

The update rule is very easy, and only one-time update is performed.

8.2.2 Targetted Attack

The goal of targetted attack is to add a perturbation such that the output label is a specific class. Suppose that the target class is $t \in \{1, \dots, M\}$. Similar to the idea in un-targetted attack, the typical formulation is

$$\begin{aligned} \min_{\delta \in \mathbb{R}^d} \quad & \|\delta\|_p && \text{(8.3a)} \\ \text{s.t.} \quad & f(x + \delta) = t && \text{(8.3b)} \\ & x + \delta \in [0, 1]^d && \text{(8.3c)} \end{aligned}$$

This kind of attack is called *Carlini-Wagner* attack, proposed in the paper (Carlini and Wagner, 2017). Solving (8.3) is in general harder than solving (8.1). We view this problem as an equality-constrained optimization, which can be solved using *penalty method*:

- Step 1: Find a function $g(\cdot)$ such that

  $$g(\tilde{x}) \leq 0 \implies f(\tilde{x}) = t$$

  - For example, construct $g(\tilde{(x)}) = \left(\frac{1}{2} - [Z(\tilde{x})]_t\right)_+$, whre $([Z(\tilde{x})]_t)_{t=1}^M$ is the output before the softmax layer. If $g(\tilde{(x)}) \leq 0$, then the t-th entry before the softmax layer is larger than $\frac{1}{2}$, i.e., $f(\tilde{x}) = t$.
  - Similar to the idea above, construct

    $$g(\tilde{x}) = \left(\max_{i \neq t}\{[Z(\tilde{x})]_i\} - [Z(\tilde{x})]_t\right)_+$$

- Step 2: Penalize the equality constraint into the objective function:

$$\begin{aligned} \min_{\delta} \quad & \|\delta\|_p + c \cdot g(x + \delta) && \text{(8.4a)} \\ \text{s.t.} \quad & x + \delta \in [0, 1]^d && \text{(8.4b)} \end{aligned}$$

Solving Optimization Problems in Adversary Attack Either Zeroth-order or First-order optimization algorithm can be used to solve the problems like (8.1) or (8.4). The advantage of first-order methods, such as projected gradient descent and Adam (Kingma and Ba, 2015), are fast convergence rates, but they need the gradient information, i.e., using backpropagation of the neural-nets, i.e., white box attack is needed, which is in-practical in life. Therefore, people consider the zeroth-order methods, also called the *derivative-free methods.* This method estimates gradient information using objective evaluation. However, this method is *slower* than gradient methods, by order of low polynomial of problem dimension.

8.2.3 Zeroth-order Optmization

The basic ideas of zeroth-order methods are to approximate the gradient information by calling objective value several times. At point x, the Gausaain vector u is generated with correlation B^{-1}. Then the gradient at x can be approximated as:

$$g_\mu(x) = \frac{f(x+\mu u) - f(x)}{\mu} \cdot Bu,$$

or

$$\hat{g}_\mu(x) = \frac{f(x+\mu u) - f(x-\mu u)}{2\mu} \cdot Bu.$$

Given T iterations for running the algorithm, the optimality measures are $h_T := \mathbb{E}[f(x_T)] - f^*$ for convex problems, or $\tilde{h}_T := \mathbb{E}[\|\nabla f(x_T)\|^2]$ for non-convex problems. The typical zeroth order method is the random gradient free algorithm (Nesterov, 2011), i.e., in each iteration perform the gradient-descent-like update:

$$x_{k+1} = x_k - hB^{-1} g_\mu(x_k).$$

For convex objective f, $h_T = \mathcal{O}(d/T) + \mathcal{O}(\epsilon)$, where d is the problem dimension. To achieve $h_T \leq \epsilon$, $T = \mathcal{O}(d/\epsilon)$ iterations are needed. For non-conex f, $\tilde{h}_T = \mathcal{O}(d/T) + \mathcal{O}(\epsilon)$. To achieve $\tilde{h}_T \leq \epsilon$, $T = \mathcal{O}(d/\epsilon)$ iterations are needed.

8.3 Adversarial Defense

Currently, there are two kinds of defense strategies:

- Adversarial training: Defender generates training data using a known attack, and then tries to improve his model. However, the model may still be vulnerable under new kinds of attacks. Moreover, it does not perform very well even for given attacks.

- Defensive distilling: Based on distilling the network knowledge, which is effective, i.e., reducing robust error from 95% to 5%. However, it does not perform well for the $C\&W$ attacks

8.3.1 Certified Defense

This defense method borrows the idea of robust optimization, which aims to solve the min-max problem:

$$\begin{aligned} \min_{\theta} \mathbb{E}_{(x,y)\sim\mathbb{P}} \quad & \max_{\tilde{x}} \left[1\{f_\theta(\tilde{x}) \neq y\}\right] && (8.5a) \\ \text{s.t.} \quad & \|\tilde{x} - x\|_p \leq \varepsilon && (8.5b) \end{aligned}$$

The objective (8.5a) is to minimize the percentage of getting an error among all samples, and the constraint (8.5b) assumes the perturbation magnitude is bounded by ε.

The state of art for certified defense method is shown in the following table (Wong *et al.*, 2018):

Accuracy	MINST	CIFAR-10
$\varepsilon = 0.1$	3%	34%
$\varepsilon = 0.3$	34%	
$\varepsilon = 2/255$		36%
$\varepsilon = 2/255$		70%

Table 8.1: The entries in the table are the robust error, the ε denotes the norm of perturbation.

Generally speaking, it is very challenging to make models robust to different kinds of attacks. It is still an active research area.

8.4 Optimization Algorithms

8.4.1 Part I: Gradient Descent (GD) method

Consider the unconstrained problem

$$\min_{x} f(x)$$

Apply the Gradient Descent (GD) method:

$$x^{k+1} = x^k - \gamma \nabla f(x^k)$$

Following questions need to be considered:

1. When does the algorithm work?
   - Assumption for the problem:
     (a) The objective function is L-smooth;
     (b) The level set $X_\delta = \{x \mid f(x) \leq \delta\}$ is bounded for all δ.
2. What kind of solution can it compute?
   - It finds the ϵ-first-order-stationary-points, i.e., x such that $\|\nabla f(x)\| \leq \epsilon$.
3. How fast can we compute it?
   - Sublinear convergence rate. We encourage the reader to first go through the convergence proof for convex objective functions in the appendix. The related contents are typed on the undergraduate course MAT3220, *Optimization II.*

By assuming that f is L-lipschitz which is not necessarily convex, we give a convergence proof for the gradient descent method.

Theorem 8.1. Suppose that the objective function f is L-lipschitz, and gradient descent method is applied in each iteration:

$$x^{k+1} = x^k - \frac{1}{L}\nabla f(x^k).$$

Define $T^* := \min\{i : \|\nabla f(x^i)\|^2 \leq \epsilon\}$, then $T^* = \mathcal{O}(1/\epsilon)$.

Proof. 1. Step 1: GD method always makes significant progress in each iteration.

$$f(x^{r+1}) - f(x^r) \le \langle \nabla f(x^r), x^{r+1} - x^r \rangle + \frac{L}{2}\|x^{r+1} - x^r\|^2 \tag{8.6a}$$

$$\le -\frac{1}{L}\|\nabla f(x^r)\|^2 + \frac{L}{2}\|x^{r+1} - x^r\|^2 \tag{8.6b}$$

$$\le -\frac{1}{2L}\|\nabla f(x^r)\|^2 \tag{8.6c}$$

where (8.6a) is by the Lipschitzness of f; (8.6b) is by the identity $x^{r+1} - x^r = -1/L\nabla f(x^r)$; (8.6c) is by the inequality $\|x^{r+1} - x^r\| \le \frac{1}{L}\|\nabla f(x^r)\|$.

2. Step 2: Considering the best performance in first T^* iteration.

By (8.6c), we imply

$$f(x^\infty) - f(x^0) \le -\frac{1}{2L}\sum_{r=0}^{\infty}\|\nabla f(x^r)\|^2 \tag{8.7}$$

Moreover, since T^* is the first iteration where the iterate reaches the ϵ-suboptimality point,

$$\begin{aligned}\epsilon &\le \frac{1}{T^*}\sum_{r=1}^{T^*}\|\nabla f(x^r)\|^2 \\ &\le \frac{1}{T^*}\sum_{r=0}^{\infty}\|\nabla f(x^r)\|^2 \le \frac{2L}{T^*}[f(x^0) - f(x^\infty)]\end{aligned}$$

It follows that

$$T^* \le \frac{2L[f(x^0) - f(x^\infty)]}{\epsilon} = \mathcal{O}(1/\epsilon).$$

□

Remark 8.2. 1. The convergence order does matter, but before caring about that, we need to pay attention to the *optimality measure*, i.e., whether $\|\nabla f(x)\| \le \epsilon$ or $\|\nabla f(x)\|^2 \le \epsilon^2$.

2. To reach ϵ-FOSP, we need $\mathcal{O}(1/\epsilon^2)$ iteration; in deep learning training, the gradient descent operation in each iteration is expansive.

3. The gradient descent, if converge, then it converges to SOSP with probability one, and it will escape strict saddle points.

   However, the gradient descent does not necessarily converge to SOSP. Moreover, the convergence of gradient descent to SOSP is too slow, and perturbation can rescue this situation. Take one special optimization as an example.

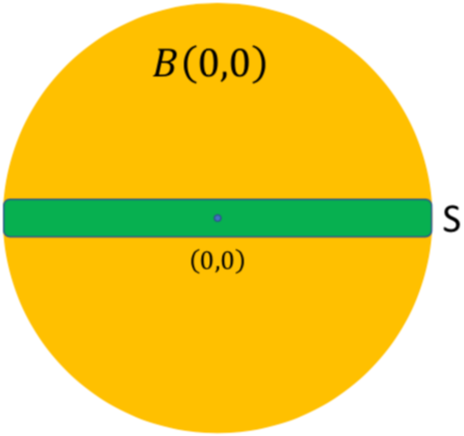


Figure 8.2: Gradient Descent cannot escape saddle point efficiently

The objective is a quadratic function

$$f(x) = \frac{1}{2} x^{\mathrm{T}} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} x$$

The gradient descent formula is given by:

$$x^{k+1} = \begin{pmatrix} 1-\gamma & 0 \\ 0 & 1+\gamma \end{pmatrix}$$

In this case, the gradient descent will get stuck around the line of the saddle point $(0,0)$, but perturbed gradient descent motivates the iterates to explore more areas around the saddle point.

In the next lecture, modern optimization methods will be introduced.

References

Auer, P., M. Herbster, and M. K. Warmuth (1996). “Exponentially many local minima for single neurons”. In: *Advances in Neural Information Processing Systems 8*. Ed. by D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo. MIT Press. 316–322. URL: http://papers.nips.cc/paper/1028-exponentially-many-local-minima-for-single-neurons.pdf.

Baldi, P. and K. Hornik (1989). “Neural networks and principal component analysis: Learning from examples without local minima”. *Neural Networks*. 2(1): 53–58. ISSN: 0893-6080. DOI: https://doi.org/10.1016/0893-6080(89)90014-2. URL: http://www.sciencedirect.com/science/article/pii/0893608089900142.

Balduzzi, D., M. Frean, L. Leary, J. P. Lewis, K. W.-D. Ma, and B. McWilliams (2017). “The Shattered Gradients Problem: If Resnets Are the Answer, then What is the Question?” In: *Proceedings of the 34th International Conference on Machine Learning - Volume 70. ICML’17*. Sydney, NSW, Australia: JMLR.org. 342–350. URL: http://dl.acm.org/citation.cfm?id=3305381.3305417.

Barron, A. R. (1994). “Approximation and estimation bounds for artificial neural networks”. *Machine Learning*. 14(1): 115–133.

Billingsley, P. (1986). *Probability and Measure*. Second. John Wiley and Sons.

Carlini, N. and D. Wagner (2017). "Towards Evaluating the Robustness of Neural Networks". In: *2017 IEEE Symposium on Security and Privacy (SP)*. 39–57. DOI: 10.1109/SP.2017.49.

Chen, P.-Y., H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh (2017). "ZOO: Zeroth Order Optimization Based Black-box Attacks to Deep Neural Networks Without Training Substitute Models". In: *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security. AISec'17*. Dallas, Texas, USA: ACM. 15–26. DOI: 10.1145/3128572.3140448.

Chen, X., S. Liu, R. Sun, and M. Hong (2019). "On the Convergence of A Class of Adam-Type Algorithms for Non-Convex Optimization". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=H1x-x309tm.

Cybenko, G. (1989). "Approximation by superpositions of a sigmoidal function". *Mathematics of Control, Signals, and Systems (MCSS)*. 2(4): 303–314. ISSN: 0932-4194. DOI: 10.1007/BF02551274. URL: http://dx.doi.org/10.1007/BF02551274.

Duchi, J., E. Hazan, and Y. Singer (2011). "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization". *J. Mach. Learn. Res.* 12(July): 2121–2159. ISSN: 1532-4435. URL: http://dl.acm.org/citation.cfm?id=1953048.2021068.

Frankle, J. and M. Carbin (2019). "The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=rJl-b3RcF7.

Garipov, T., P. Izmailov, D. Podoprikhin, D. P. Vetrov, and A. G. Wilson (2018). "Loss Surfaces, Mode Connectivity, and Fast Ensembling of DNNs". In: *Advances in Neural Information Processing Systems 31*. Ed. by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett. Curran Associates, Inc. 8789–8798. URL: http://papers.nips.cc/paper/8095-loss-surfaces-mode-connectivity-and-fast-ensembling-of-dnns.pdf.

Gilboa, D., B. Chang, M. Chen, G. Yang, S. S. Schoenholz, E. H. Chi, and J. Pennington (2019). "Dynamical Isometry and a Mean Field Theory of LSTMs and GRUs". *CoRR*. abs/1901.08987. arXiv: 1901.08987. URL: http://arxiv.org/abs/1901.08987.

Glorot, X. and Y. Bengio (2010). “Understanding the difficulty of training deep feedforward neural networks”. In: *In Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS?10). Society for Artificial Intelligence and Statistics.*

Glorot, X., A. Bordes, and Y. Bengio (2010). “Deep Sparse Rectifier Neural Networks”. In: vol. 15.

Goodfellow, I., J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio (2014). “Generative Adversarial Nets”. In: *Advances in Neural Information Processing Systems 27*. Ed. by Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger. Curran Associates, Inc. 2672–2680. URL: http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf.

Goodfellow, I., J. Shlens, and C. Szegedy (2015a). “Explaining and Harnessing Adversarial Examples”. In: *International Conference on Learning Representations*. URL: http://arxiv.org/abs/1412.6572.

Goodfellow, I., O. Vinyals, and A. Saxe (2015b). “Qualitatively Characterizing Neural Network Optimization Problems”. In: *International Conference on Learning Representations*. URL: http://arxiv.org/abs/1412.6544.

Gotmare, A., N. Shirish Keskar, C. Xiong, and R. Socher (2018). *Using Mode Connectivity for Loss Landscape Analysis.*

Han, S., J. Pool, J. Tran, and W. Dally (2015). “Learning both Weights and Connections for Efficient Neural Network”. In: *Advances in Neural Information Processing Systems 28*. Ed. by C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett. Curran Associates, Inc. 1135–1143. URL: http://papers.nips.cc/paper/5784-learning-both-weights-and-connections-for-efficient-neural-network.pdf.

Hanin, B. and D. Rolnick (2018). “How to Start Training: The Effect of Initialization and Architecture”. In: *Advances in Neural Information Processing Systems 31*. Ed. by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett. Curran Associates, Inc. 571–581. URL: http://papers.nips.cc/paper/7338-how-to-start-training-the-effect-of-initialization-and-architecture.pdf.

He, K., X. Zhang, S. Ren, and J. Sun (2015). "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification". In: *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV). ICCV '15.* Washington, DC, USA: IEEE Computer Society. 1026–1034. ISBN: 978-1-4673-8391-2. DOI: 10.1109/ICCV.2015.123. URL: http://dx.doi.org/10.1109/ICCV.2015.123.

He, K., X. Zhang, S. Ren, and J. Sun (2016). "Deep Residual Learning for Image Recognition". In: 770–778. DOI: 10.1109/CVPR.2016.90.

Hornik, K. (1991). "Approximation Capabilities of Multilayer Feedforward Networks". *Neural Netw.* 4(2): 251–257. ISSN: 0893-6080. DOI: 10.1016/0893-6080(91)90009-T. URL: http://dx.doi.org/10.1016/0893-6080(91)90009-T.

"How to comment the paper "The Lottery Ticket Hypothesis"" (n.d.). https://www.zhihu.com/question/323214798. Accessed: 2019-08-14.

Ilyas, A., L. Engstrom, A. Athalye, and J. Lin (2018). "Black-box Adversarial Attacks with Limited Queries and Information". In: *Proceedings of the 35th International Conference on Machine Learning.* Vol. 80. *Proceedings of Machine Learning Research.* PMLR. 2137–2146.

Kawaguchi, K. (2016). "Deep Learning without Poor Local Minima". In: *Advances in Neural Information Processing Systems 29.* Ed. by D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett. Curran Associates, Inc. 586–594. URL: http://papers.nips.cc/paper/6112-deep-learning-without-poor-local-minima.pdf.

Kingma, D. P. and J. Ba (2015). "Adam: A method for stochastic optimization". In: *International Conference on Learning Representations (ICLR).*

Kurach, K., M. Lucic, X. Zhai, M. Michalski, and S. Gelly (2018). "The GAN Landscape: Losses, Architectures, Regularization, and Normalization". *CoRR.* abs/1807.04720. arXiv: 1807.04720. URL: http://arxiv.org/abs/1807.04720.

Lee, J. D., M. Simchowitz, M. I. Jordan, and B. Recht (2016). “Gradient Descent Only Converges to Minimizers”. In: *29th Annual Conference on Learning Theory*. Ed. by V. Feldman, A. Rakhlin, and O. Shamir. Vol. 49. *Proceedings of Machine Learning Research*. Columbia University, New York, New York, USA: PMLR. 1246–1257. URL: http://proceedings.mlr.press/v49/lee16.html.

Li, D., T. Ding, and R. Sun (2018). *Over-Parameterized Deep Neural Networks Have No Strict Local Minima For Any Continuous Activations*.

Li, P. and P.-M. Nguyen (2019). “On Random Deep Weight-Tied Autoencoders: Exact Asymptotic Analysis, Phase Transitions, and Implications to Training”. In: *International Conference on Learning Representations*.

Lin, H. and S. Jegelka (2018). “ResNet with one-neuron hidden layers is a Universal Approximator”. In: *Advances in Neural Information Processing Systems 31*. Ed. by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett. Curran Associates, Inc. 6169–6178. URL: http://papers.nips.cc/paper/7855-resnet-with-one-neuron-hidden-layers-is-a-universal-approximator.pdf.

Nesterov, Y. (2011). “Random gradient-free minimization of convex functions”. Jan.

Pennington, J., S. S. Schoenholz, and S. Ganguli (2017). “Resurrecting the sigmoid in deep learning through dynamical isometry: theory and practice”. In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*. 4785–4795.

Pennington, J., S. S. Schoenholz, and S. Ganguli (2018). “The Emergence of Spectral Universality in Deep Networks”. In: *AISTATS*.

Poole, B., S. Lahiri, M. Raghu, J. Sohl-Dickstein, and S. Ganguli (2016). “Exponential expressivity in deep neural networks through transient chaos”. In: *Advances in Neural Information Processing Systems 29*. Ed. by D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett. Curran Associates, Inc. 3360–3368. URL: http://papers.nips.cc/paper/6322-exponential-expressivity-in-deep-neural-networks-through-transient-chaos.pdf.

Razaviyayn, M. (2014). “Successive Convex Approximation: Analysis and Applications”. In:

Reddi, S. J., S. Kale, and S. Kumar (2018). “On the Convergence of Adam and Beyond”. In: *International Conference on Learning Representations.* URL: https://openreview.net/forum?id=ryQu7f-RZ.

Saxe, A. M., J. L. Mcclelland, and S. Ganguli (2014). “Exact solutions to the nonlinear dynamics of learning in deep linear neural network”. In: *In International Conference on Learning Representations.*

Srivastava, R. K., K. Greff, and J. Schmidhuber (2015). “Highway Networks”. cite arxiv:1505.00387Comment: 6 pages, 2 figures. Presented at ICML 2015 Deep Learning workshop. Full paper is at arXiv:1507.06228. URL: http://arxiv.org/abs/1505.00387.

Szegedy, C., S. Ioffe, and V. Vanhoucke (2016). “Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning”. In: *AAAI.*

Szegedy, C., W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus (2014). “Intriguing properties of neural networks”. In: *International Conference on Learning Representations.* URL: http://arxiv.org/abs/1312.6199.

Tieleman (2012). *Lecture 6.5-rmsprop.* Available at the link https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf.

“Understanding nonconvex optimization” (n.d.). http://praneethnetrapalli.org/UnderstandingNonconvexOptimization-V5.pdf. Accessed: 2019-08-18.

Wang, J. (2019a). *MAT2006: Elementary Real Analysis.* Available at the link https://walterbabyrudin.github.io/information/Notes/MAT2006.pdf.

Wang, J. (2019b). *MAT3006: Real Analysis; Lecture 8.* Available at the link https://walterbabyrudin.github.io/information/Updates/MAT3006/Week4_Wednesday.pdf.

Wong, E., F. R. Schmidt, J. H. Metzen, and J. Z. Kolter (2018). “Scaling Provable Adversarial Defenses”. In: *Proceedings of the 32Nd International Conference on Neural Information Processing Systems. NIPS’18.* Montr&#233;al, Canada: Curran Associates Inc. 8410–8419. URL: http://dl.acm.org/citation.cfm?id=3327757.3327932.

Wu, Y. and K. He (2018). "Group Normalization". In: *The European Conference on Computer Vision (ECCV).*

Xiao, L., Y. Bahri, J. Sohl-Dickstein, S. Schoenholz, and J. Pennington (2018). "Dynamical Isometry and a Mean Field Theory of CNNs: How to Train 10,000-Layer Vanilla Convolutional Neural Networks". In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by J. Dy and A. Krause. Vol. 80. *Proceedings of Machine Learning Research*. Stockholmsmassan, Stockholm Sweden: PMLR. 5393–5402.

Xiao-Hu Yu and Guo-An Chen (1995). "On the local minima free condition of backpropagation learning". *IEEE Transactions on Neural Networks*. 6(5): 1300–1303. ISSN: 1045-9227. DOI: 10.1109/72.410380.

Zhang, H., Y. N. Dauphin, and T. Ma (2019). "Residual Learning Without Normalization via Better Initialization". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=H1gsz30cKX.

Zhang, Y., R. Tapia, and L. Velazquez (2000). "On Convergence of Minimization Methods: Attraction, Repulsion, and Selection". *Journal of Optimization Theory and Applications*. 107(3): 529–546. ISSN: 1573-2878. DOI: 10.1023/A:1026443131121. URL: https://doi.org/10.1023/A:1026443131121.