9

Optimization Algorithms

9.1 Reviewing

- In order to reach $\sqrt{\epsilon}$-FOSP such that $\|\nabla f(\theta^r)\|^2 \leq \epsilon$, the $\mathcal{O}(1/\epsilon)$ iterations are needed.
- Observe that each update of gradient descent is equivalent to minimizing a taylor-expansion-based quadratic upper bound on the objective function:

  $$x^{k+1} = x^k - t_k \nabla f(x^k) = \arg\min_{x \in \mathcal{X}} \frac{1}{2t_k} \|x - x^k\|^2 + \nabla^{\mathrm{T}} f(x^k)(x - x^k).$$

  Therefore, gradient descent is one special case of successive convex approximation (SCA) technique. See the survey paper (Razaviyayn, 2014) on the SCA technique for detail.

9.2 Variants of Gradient Descent (GD) Method

9.2.1 Scaled GD

Consider the minimization problem

$$\min_{\theta} F(\theta).$$

The intuition of the scaled GD is to left-multiply the gradient $\nabla F(\theta_t)$ with a positive definite matrix to avoid osillation:

$$\theta_{t+1} = \theta_t - \alpha_t D_t \nabla F(\theta_t), \quad D_t \succ 0.$$

Example 9.1. Consider the linear regression problem

$$\min_{\theta} \ \frac{1}{2}\|A\theta - b\|^2, \tag{9.1}$$

with

$$\begin{aligned} A &= \begin{pmatrix} a_1 & \cdots & a_d \end{pmatrix}, \\ \theta &= (\theta_i)_{i=1}^d. \end{aligned}$$

In practice, the size for the feature a_1 can be very differnt from that of a_2. In order to resolve this issue, the data transformation technique is applied, i.e., introduce a new variable

$$w \triangleq S\theta, \qquad \text{where } S = \operatorname{diag}(s_1, \dots, s_d).$$

It suffices to solve the new problem where the data matrix AS^{-1} is well-conditioned:

$$\min_{w} \|AS^{-1}W - b\|^2 \tag{9.2}$$

- The GD method for solving (9.1) is given by

$$\theta_{t+1} = \theta_t - \gamma A^{\mathrm{T}}(A\theta_t - b)$$

- The GD method for solving the transformed problem (9.2) is given by

$$w_{t+1} = w_t - \gamma S^{-1}A^{\mathrm{T}}(AS^{-1}w_t - b)$$

  Left-multiplying S^{-1} for this iteration gives

$$\theta_{t+1} = \theta_t - \gamma S^{-2}A^{\mathrm{T}}(A\theta_t - b)$$

Therefore, solving the transformed problem (9.2) is equivalent to the scaled GD method by setting $D \equiv S^{-2}$.

Example 9.2 (Newton's Method)**.** Newton's method is a special case of scaled GD by setting $D_t \equiv (\nabla^2 F(\theta_t))^{-1}$:

$$\theta_{t+1} = \theta_t - \alpha_t(\nabla^2 F(\theta_t))^{-1}\nabla F(\theta_t)$$

Newton's method performs good in linear regression problem. If we choose $\alpha_t = 1$, then Newton's method gives the optimal solution in one iteration:

$$\begin{aligned}\theta_{t+1} &= \theta_t - \alpha_t(\nabla^2 F(\theta_t))^{-1}\nabla F(\theta_t)\\ &= \theta_t - (A^{\mathrm{T}}A)^{-1}\cdot[A^{\mathrm{T}}(A\theta - b)]\\ &= (A^{\mathrm{T}}A)^{-1}A^{\mathrm{T}}b.\end{aligned}$$

Example 9.3 (Gauss-Newton Method)**.** Consider the non-linear regression problem

$$\min_{\theta}\ F(\theta) \triangleq \frac{1}{2}\sum_{i=1}^{n}[f_i(\theta)]^2$$

The gradient and Hessian are computed as follows:

$$\begin{aligned}\nabla F(\theta) &= \sum_{i=1}^{n} f_i(\theta)\nabla_\theta f_i(\theta)\\ \nabla^2 F(\theta) &= \sum_{i=1}^{n}\nabla_\theta f_i(\theta)\nabla_\theta^{\mathrm{T}} f_i(\theta) + \underbrace{\sum_{i=1}^{n} f_i(\theta)\nabla^2 f_i(\theta)}_{S(\theta)}\end{aligned}$$

The Gauss-Newton method sets

$$D_t \equiv \left(\sum_{i=1}^{n}\nabla_\theta f_i(\theta)\nabla_\theta^{\mathrm{T}} f_i(\theta)\right)^{-1},$$

which is an approximation of the inverse of the Hessian matrix. There are two advantages:

1. It reduces the Hessian computation time into gradient computation time.

2. The D_t is positive definite, which avoids the numerical instability of Newton's method. Actually, in the paper (Zhang *et al.*, 2000), Prof. Yin Zhang compares the difference between these

two methods. He found that the Gauss-Newton method possesses a desirable behavior of only seeking global (or good local) minima, while the Newton method is blindly attracted to all kinds of FOSPs.

Remark 9.1. There are generally two choices of step size for GD and variants of GD method:

- Constant step size: $\alpha_t \in (0, 1/L)$;
- Diminishing stepsize: $\alpha_t \to 0$ but satisfies the infinite travel condition
$$\sum_{t=1}^{\infty} \alpha_t = \infty.$$

For constrained minimization problem, the projected gradient descent method is applied.

9.2.2 Stochastic Gradient Descent (SGD)

Consider minimizing an objective with the finite-sum form:

$$F(\theta) = \frac{1}{N}\sum_{i=1}^{N} \ell(f_\theta(x_i), y_i) \triangleq \frac{1}{N}\sum_{i=1}^{N} F_i(\theta)$$

The SGD iteration is given by

$$\theta_{t+1} = \theta_t - \alpha_t \nabla F_{j[t]}(\theta_t),$$

where $j[t]$ is the sample randomly picked from $\{1, \ldots, N\}$ at the t-th iteration.

Remark 9.2. 1. The SGD works under the following assumptions:

- Each component function F_i is Lipschitz continuous, i.e., $\|\nabla F_i(\theta) - \nabla F_i(\xi)\| \leq L \cdot \|\theta - \xi\|$;
- The variance for estimation of gradient is uniformly bounded:
$$\mathbb{E}\left[\|\nabla F_{j[t]}(x) - \nabla f(x)\|^2\right] \leq \sigma^2$$

2. Stochastic gradient method is not a descent method, though it is frequently referred to as stochastic gradient descent in the literature.

3. We will show that α_t has to be a dimishing step-size.

4. If $j[t]$ are selected like the form $\{1, \dots, N, 1 \dots, N, \dots\}$, then it reduces to the incremental gradient algorithm.

5. People prefer to use SGD in machine learning field instead of GD. The disadvantage for SGD is that it needs $\mathcal{O}(1/\epsilon^2)$ iterations to get $\sqrt{\epsilon}$-FOSP, while GD needs only $\mathcal{O}(1/\epsilon)$ iterations. However, the complexity of the number of gradient calls in each inner iteration is different. The computation of the gradient of component functions per iteration for SGD is $\mathcal{O}(1)$, but GD requires $\mathcal{O}(N)$ calls. Due to this fact, SGD is faster than GD in practice.

	# gradient calls per iteration	# iterations
GD	n	$\boldsymbol{\epsilon^{-1}}$
SGD	$\mathbf{1}$	ϵ^{-2}

Table 9.1: Comparison of SGD and GD

Theorem 9.1. Suppose that the objective function satisfies the assumptions mentioned before, and the SGD is applied in each iteration:

$$\theta_{t+1} = \theta_t - \alpha_t g_t, \qquad \text{where } g_t := \nabla F_{j[t]}(\theta_t).$$

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

Proof. 1. Step 1: SGD always makes significant progess in each iteration.

$$\begin{aligned} F(\theta_{t+1}) &\leq F(\theta_t) + \langle \nabla F(\theta_t), \theta_{t+1} - \theta_t \rangle + \frac{L}{2}\|\theta_{t+1} - \theta_t\|^2 \\ &= F(\theta_t) - \alpha_t \langle \nabla F(\theta_t), g_t \rangle + \frac{L\alpha_t^2}{2}\|g_t\|^2 \end{aligned}$$

Then taking conditional expectation both sides given θ_t leads to

$$\begin{aligned}\mathbb{E}[F(\theta_{t+1}) \mid \theta_t] &\leq F(\theta_t) - \alpha_t \langle \nabla F(\theta_t), \mathbb{E}[g_t \mid \theta_t]\rangle + \frac{L\alpha_t^2}{2}\mathbb{E}[\|g_t\|^2 \mid \theta_t] \\ &= F(\theta_t) - \alpha_t \|\nabla F(\theta_t)\|^2 + \frac{L\alpha_t^2}{2}\mathbb{E}[\|g_t\|^2 \mid \theta_t]\end{aligned} \tag{9.3}$$

Furthermore, by the equality $\|a\|^2 \leq 2\|a-b\|^2 + 2\|b\|^2$,

$$\begin{aligned}\mathbb{E}[\|g_t\|^2 \mid \theta_t] &\leq 2\mathbb{E}[\|g_t - \nabla F(\theta_t)\|^2 \mid \theta_t] + 2\|\nabla F(\theta_t)\|^2 \\ &= 2\sigma^2 + 2\|\nabla F(\theta_t)\|^2\end{aligned}$$

Substituting this identity into the RHS of (9.3) gives

$$\begin{aligned}\mathbb{E}[F(\theta_{t+1}) \mid \theta_t] &\leq F(\theta_t) + \left(-\alpha_t + L\alpha_t^2\right)\|\nabla F(\theta_t)\|^2 + L\alpha_t^2\sigma^2 \\ &\leq F(\theta_t) - \frac{\alpha_t}{2}\|\nabla F(\theta_t)\|^2 + L\alpha_t^2\sigma^2\end{aligned}$$

where the last inequality is by choosing sufficiently small α_t. Therefore, the expected decrease in each iteration is bounded by the gradient term plus some variance:

$$\mathbb{E}[F(\theta_{t+1})] - \mathbb{E}[F(\theta_t)] \leq -\frac{\alpha_t}{2}\|\nabla F(\theta_t)\|^2 + L\alpha_t^2\sigma^2$$

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

   Suppose that the step size $\alpha_t \equiv \alpha$ for all t. Then summing over the inequality above for $t = 1, \dots, T^*$ gives

$$\begin{aligned}\frac{1}{T^*}\sum_{t=0}^{T^*}\mathbb{E}[\|\nabla F(\theta_t)\|^2] &\leq \frac{2\mathbb{E}[F(\theta_0) - F(\theta_{T^*+1})]}{\alpha T^*} + 2L\alpha\sigma^2 \\ &\leq \frac{2[F(\theta_0) - F(\theta_\infty)]}{\alpha T^*} + 2L\sigma^2\alpha\end{aligned}$$

   Choosing $\alpha = 1/\sqrt{T}$ gives

$$\frac{1}{T^*}\sum_{t=0}^{T^*}\mathbb{E}[\|\nabla F(\theta_t)\|^2] \leq \frac{2[F(\theta_0) - F(\theta_\infty)] + 2L\sigma^2}{\sqrt{T^*}}$$

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

$$\epsilon \leq \frac{1}{T^*}\sum_{t=0}^{T^*}\mathbb{E}[\|\nabla F(\theta_t)\|^2] \leq \frac{2[F(\theta_0) - F(\theta_\infty)] + 2L\sigma^2}{\sqrt{T^*}}$$

Or equivalently,

$$T^* \leq \frac{2[F(\theta_0) - F(\theta_\infty)] + 2L\sigma^2}{\epsilon^2} = \mathcal{O}(\epsilon^{-2}).$$

□

9.3 Momentum-based Method

Now we introduce several famous momentum methods.

9.3.1 Heavy-Ball Method

Each update of Heavy-Ball Method uses the information from the last and the last second iterate points:

$$\begin{cases} m_t = \beta m_{t-1} + (1-\beta)\nabla F(\theta_t) \\ \theta_{t+1} = \theta_t - \alpha_t m_t \end{cases}$$

Remark 9.3. For convex objective function, the heavy-ball method can accelerate the convergence rate from $\mathcal{O}(1/\epsilon)$ iterations into $\mathcal{O}(1/\sqrt{\epsilon})$ iterations. However, it is not clear how the heavy-ball method behaves in general non-convex problems.

9.3.2 Adaptive Gradient methods (AdaGrad) (Duchi *et al.*, 2011)

The AdaGrad can be viewed as a scaled version of the SGD:

$$\begin{aligned} &\theta_{t+1} = \theta_t - \alpha_t (D^t)^{-1/2} g_t, \\ \text{where} \quad &D^t = \tfrac{1}{t}\textstyle\sum_{s=1}^t g_s \cdot g_s^{\mathrm{T}} \end{aligned}$$

The inituition is that the step size of SGD is hard to control. The AdaGrad resolves this issue by rescaling the step size of i-th coordinate from α_t to $\frac{\alpha_t}{\sqrt{\frac{1}{t}\sum_{s=1}^t g_i g_j}}$ In practice, the inverse of D^t is difficult to compute, and therefore we approximate its inverse by simply inversing the diagonal entries of D^t:

$$\theta_{t+1} = \theta_t - \alpha_t \operatorname{diag}\left(\frac{1}{t}\sum_{s=1}^t g_s \cdot g_s^{\mathrm{T}}\right)^{-1/2} g_t$$

Or equivalently, this iteration can be written as

$$\begin{cases} v_t = \dfrac{1}{t}\displaystyle\sum_{s=0}^{t} g_s \circ g_s \\ \theta_{t+1} = \theta_t - \alpha_t (v^t)^{-1/2} \circ g_t \end{cases}$$

9.3.3 RMS-Prop (Tieleman, 2012)

The RMSprop algorithm is a variant version of AdaGrad, by doing exponential decay of previous information v_{t-1}:

$$\begin{cases} v_t = \beta v_{t-1} + (1-\beta) g_t \circ g_t \\ \theta_{t+1} = \theta_t - \alpha_t (v^t)^{-1/2} \circ g_t \end{cases}$$

Remark 9.4. Unfortunately, there is no theory concerning how to choose β optimally in literature.

9.3.4 Adam (Kingma and Ba, 2015)

Finally, let's introduce the Adam algorithm, which not simply enjoys features from heavy-ball method such that the gradient estimate is momentum, but also borrows ideas from RMSprop such that the step size is adaptively chosen with exponential decay.

$$\begin{cases} m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t \\ v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t \circ g_t \\ \theta_{t+1} = \theta_t - \alpha_t v_t^{-1/2} \circ m_t \end{cases}$$

Bibliography In 2018 ICLR, the paper (Reddi *et al.*, 2018) gives a counter-example to show that Adam does not converge for convex problems. Specifically, consider the problem

$$\min_{\theta} F(\theta) = \sum_{j=1}^{n} F_j(\theta),$$

with

$$F_j(\theta) = \begin{cases} 5.5\theta^2, & j = 1 \\ -0.5\theta^2, & j \neq 1 \end{cases}$$

The special parameter of Adam $\beta_1 = 0, \beta_2 = 1$ gives the signSGD:

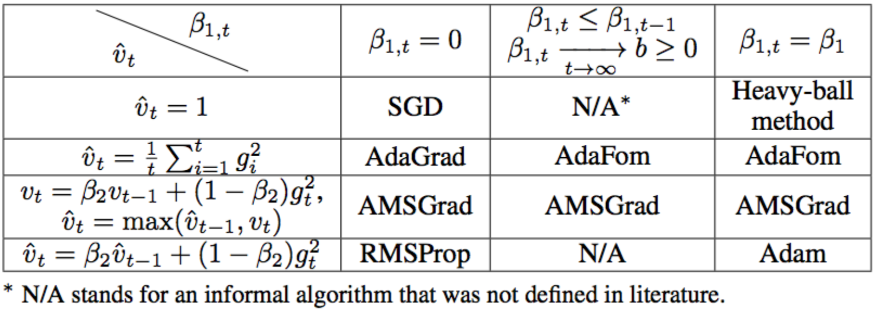
$$\theta_{t+1} = \theta_t - \alpha_t \text{sign}(g_t)$$

which is unlikely to converge. Then they propose AMSGrad to correct this algorithm. Later in 2019 ICLR, the paper (Chen *et al.*, 2019) shows that Adam-type algorithms do not converge for non-convex problems, and they propose the AdaFom to correct previous algorithms' behaviors.

Summary The Adam-Type Method can be expressed as follows:

$$\begin{cases} m_t = \beta_{1,t} m_{t-1} + (1 - \beta_{1,t}) g_t \\ \hat{v}_t = h_t(g_1, \ldots, g_t) \\ \theta_{t+1} = \theta_t - \alpha_t m_t / \sqrt{\hat{v}_t} \end{cases}$$

Some popular variants of the Adam algorithm is shown in the Table below:



$\beta_{1,t}$ $\hat{v}_t$	$\beta_{1,t} = 0$	$\beta_{1,t} \le \beta_{1,t-1}$ $\beta_{1,t} \xrightarrow[t\to\infty]{} b \ge 0$	$\beta_{1,t} = \beta_1$
$\hat{v}_t = 1$	SGD	N/A*	Heavy-ball method
$\hat{v}_t = \frac{1}{t}\sum_{i=1}^{t} g_i^2$	AdaGrad	AdaFom	AdaFom
$v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2$, $\hat{v}_t = \max(\hat{v}_{t-1}, v_t)$	AMSGrad	AMSGrad	AMSGrad
$\hat{v}_t = \beta_2 \hat{v}_{t-1} + (1-\beta_2) g_t^2$	RMSProp	N/A	Adam

* N/A stands for an informal algorithm that was not defined in literature.

Figure 9.1: Variants of Adam algorithm

9.4 Nonconvex nonconcave minimax optimization

Consider the minimax problem

$$\min_x \max_y f(x, y)$$

If we can obtain $y^* = h(x)$, then it suffices to solve the minimization problem

$$\min_x g(x) \triangleq f(x, h(x)).$$

In practice, the function $g(x)$ is always defined implicitly. This problem is the basis of generative adversarial networks and robust training. The gradient descent ascent (GDA) Algorithm is widely used in this problem, but this algorithm cannot even solve a linear problem

$$\min_x \max_y x^{\mathrm{T}} Ay.$$

Remark 9.5. 1. Now there are two state-of-the-art algorithms for solving the minimax problem. The first one is the optimistic GDA; and the second one is to applying BCD heuristic, i.e., approximately optimizing the inner problem in terms of y for few iterations, and then perform gradient descent for the outer problem for one iteration. The assumptions for the second algorithm is that the problem in terms of y is concave, and the constraint set is bounded.

2. It is even not clear what is the local optimality for the minimax problem in the general setting.

3. The GDA is widely used, but its convergence properties remain to be understood.

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.