3

Taming Explosion/Vanishing: Initialization

3.1 Reviewing

1. Does Xevier intialization allow $W_{ij}^{\ell} \sim C \cdot \text{rand}$?
   No. From the assignment we know that $\mathbb{E}W_{ij}^{\ell}$ should be zero.
2. How to pick variance of $W_{i,j}$ for non-linear activation functions?
   - Relu activiation: twice the variance.
   - other types of activiation: to be discussed today
3. Does the Chain rule work for derivative of matrix over vector?
   Not directly. We need to derive a different form

3.2 Motivation

Three topics to be discussed today:

- The difference for training *wide* versus *narrow* neural network
- Mean-field approximation
- Dynamical Isometry (spectrum analysis)

The motivation is that engineers believe that the initialization for training a neural network is important, otherwise the gradient explosion/vanishing will happen. These topics discuss the initialization with connection to gradient explosion/vanishing from different perspectives.

Example 3.1. Review the code in section (2.3). This experiment has seveal limitations:

- It only shows the signal strength does not change too much over linear networks, but what will happen if the signal undergoes a non-linear activation.
- It only shows that the signal strength does not change too much after one-layer. Does it assert that after more layers, the signal strength still remains nearly the same?
  - Last time we have shown that $\mathbb{E}(\|z^{\ell}\|^2) \approx \mathbb{E}(\|z^{\ell-1}\|^2)$. Therefore, it seems to be true that
    $$\mathbb{E}(\|z^{L}\|^2) \approx \mathbb{E}(\|z^{L-1}\|^2) \approx \cdots \approx \mathbb{E}(\|x\|^2)$$
  - However, we can run a simulation to verify the cases. If we set $L \leftarrow 100, d \leftarrow 10$, and run the following code in MATLAB:

    ```
    clear;
    L = 100;
    d = 10; % dimension for weight matrix W
    maxit = 1; % maximum iteration number

    x = ones(d,1);  norm0 = norm(x);
    for i = 1:maxit
        for l = 1:L
            W = randn(d,d)/sqrt(d);
            x = W*x;
        end
        rato = norm(x)/norm0
    end
    ```

    Then we find that the ratio of output signal strength over the input signal strength is below 10^{-3}.

We should give some more precise theoretical analysis.

1. Firstly we give a rigorous proof for that the signal strength after one-layer linear network keeps nearly the same for Xaiver Initialization. Consider the input vector $x = \mathbf{1} \in \mathbb{R}^d$ and after one-layer linear network, $z = Wx \in \mathbb{R}^d$ such that $W_{i,j} \sim \mathcal{N}(0, 1/d)$. Therfore,

$$z = \begin{pmatrix} \sum_{j=1}^d W_{1,j} \\ \sum_{j=1}^d W_{2,j} \\ \vdots \\ \sum_{j=1}^d W_{d,j} \end{pmatrix} \implies \|z\|^2 = \sum_{i=1}^d \underbrace{\left(\sum_{j=1}^d W_{i,j} \right)^2}_{\xi_i}$$

where $d \cdot \xi_i$ is the sum of the square of d i.i.d standard normal random variables, i.e., $d \cdot \xi_i \sim \chi^2(d)$. Therefore,

$$\mathbb{E}[\xi_i] = 1.$$

Moreover,

$$\begin{aligned} \mathbb{P}(|\xi_i - 1| \le \epsilon) &= \mathbb{P}(|\chi^2(d) - d| \le \epsilon \cdot d) \\ &\triangleq 1 - F((1-\epsilon)d, d) - (1 - F((1+\epsilon)d, d)) \\ &\ge 1 - ((1-\epsilon)e^{\epsilon})^{d/2} - ((1+\epsilon)e^{-\epsilon})^{d/2} \end{aligned}$$

where $F(x, d)$ denotes teh cdf of the random variable $\chi^2(d)$, and the last inequality follows from the *Chernoff bounds* on the lower and upper tails of $F(\cdot, d)$. Therefore, we conclude that the signal strength after one-layer is close to the original with high probability.

2. Then consider the signal strength after multi-layer linear network. Unrigorously, suppose that in each layer $\xi_i \approx 1 + \varepsilon$, and $\|z^\ell\|^2 = \|z^{\ell-1}\|^2(1+\varepsilon)$. Therefore, the final signal strength

$$\|z^L\|^2 \approx \|z^{L-1}\|^2(1+\epsilon) \approx \cdots \approx \|x\|^2(1+\varepsilon)^L$$

Here the parameter ε depends on d, i.e., the number of neuros in each layer. Roughly speaking, $\epsilon = \mathcal{O}(1/d)$. As a result,

$$\|z^L\|^2 = \|x\|^2(1 \pm \frac{1}{d})^L \approx \|x\|^2 e^{L/d}.$$

Remark 3.1. 1. The signal strength ratio $\|z^L\|^2/\|x\|^2 = \mathcal{O}(\exp(L/d))$, where L is the depth of the layers, d is the number of neuros in each layer. Therefore, L/d controls the width of the neural network. When L/d is very large (very small), i.e., the neural network is very narrow (very wide), it's likely that gradient explosion/vanishing will happen.

2. In practice, when $L/d > 10$, the gradient explosion/vanishing will happen; it works well if we set $L/d < 1/10$.

3. This theoretical analysis also depends on other factors such as input data and architecture. For instance, if we train CIFAR10/MINST using super-small L/d (e.g., 20 layers and 3 million neurals in widest layer), differnt from our prediction, the initilization works very well. The reason is that the data being image does make a difference, and that the neural network architecture is CNN instead of a fully connected neural network.

Bibliography For different neural network architectures such as CNN, we need to perform the similar experiments but with "fair" criteria. The paper (Glorot and Bengio, 2010) checked the CNN architecture. No one have ever performed the similar verifications for SOTA architecture.

The paper (Hanin and Rolnick, 2018) gives total regiorous proof for previous theoretical analysis based on the martingale theory:

- Failure mode 1: the signal strength normalized by the size in each layer scale in the final layer increases/decreases exponentially with the depth, i.e., $\mathbb{E}[M_L] \triangleq \mathbb{E}[1/d\|z^L\|^2] \to 0$ or ∞ as $L \to \infty$

- Failure mode 2: The empirical variance of the signal strength normalized by the size in each layer, say $\mathrm{Var}\{M_{1:L}\}$, grows exponentially with the depth. More precisely,

$$\mathbb{E}\left[\mathrm{Var}\{M_{1:L}\}\right] = \mathcal{O}\left(\exp\left(\sum_{i=1}^{L}\frac{1}{d_i}\right)\right)$$

  In particular, if all d_i's are the same, then $\mathbb{E}\left[\mathrm{Var}\{M_{1:L}\}\right] = \mathcal{O}(\exp(L/d))$.

There is some gap between Ruoyu Sun's claim and the work in this paper. What Ruoyu Sun claimed is that the variance of M_L depends on $\exp(L/d)$; but this paper claims that the empirical variance of $M_{1:L}$ depends on $\exp(L/d)$. In summary, this paper and others give a formal theory of why super-deep & super-narrow neural networks are hard to train.

3.3 General Activation

Now we discuss the answer to Question 2 raised at the beginning of this lecture.

Problem Setting Given input vector $X \in \mathbb{R}^{d\times 1}$ and random weight matrix $W \in \mathbb{R}^{d\times d}$. Define the output vector $z = \phi(Wx)$, where $\phi : \mathbb{R} \to \mathbb{R}$ is a given activation function. We are interested in the sufficient condition ensuring $\mathbb{E}[\|z\|^2] = \mathbb{E}[\|x\|^2]$.

3.3.1 The scalar-input one-layer case

Consider the case where $d = 1$ and $L = 1$. Then $x \in \mathbb{R}$ and $z = \phi(wx)$ with $w \sim \mathcal{N}(0, c)$. We want to choose c such that $\mathbb{E}[z^2] = x^2$. This problem reduces to solving a non-linear equation in terms of c:

$$x^2 = \mathbb{E}[z^2] = \mathbb{E}_w[(\phi(wx))^2] = \int (\phi(tx))^2 \,\mathrm{d}t \frac{1}{\sqrt{2\pi}} e^{-1/2t^2}$$

3.3.2 The vector-input one-layer case

Then consider the case where $d > 1$ and $L = 1$. First write down z explicitly in terms of x and w:

$$z = \phi(Wx) \implies \begin{pmatrix} z_1 \\ \vdots \\ z_d \end{pmatrix} = \begin{pmatrix} \phi(\sum_{j=1}^d w_{1j}x_j) \\ \vdots \\ \phi(\sum_{j=1}^d w_{dj}x_j) \end{pmatrix}$$

As a result,

$$\|z\|^2 = \sum_{i=1}^d \phi\left(\sum_{j=1}^d w_{ij}x_j\right)^2,$$

which implies

$$\mathbb{E}[\|z\|^2] = \mathbb{E}\left[\sum_{i=1}^{d} \phi\left(\sum_{j=1}^{d} w_{ij}x_j\right)^2\right]$$
$$= d \cdot \mathbb{E}\left[\phi\left(\sum_{j=1}^{d} w_{ij}x_j\right)^2\right] = \|x\|^2$$

where the last inequality is because that $\left\{\phi\left(\sum_{j=1}^{d} w_{ij}x_j\right)^2\right\}_{i=1:d}$ are i.i.d. This is a single equation on scalar c, which is solvable.

3.3.3 The vector-input multi-layer case

We cannot apply the techniques similar as in the previous two cases. For instance, consider the case where $d = 1$ and $L = 2$. If we have

$$z^2 = \sigma(w^2\sigma(w^1x))$$

Then we know the pdf of w^1x for fixed x, but it's hard to know the pdf of $z^1 \triangleq \sigma(w^1x)$. It's even harder to know the pdf of $z^2 = \sigma(w^2z^1)$. Instead, for the linear activation case, we have

$$\mathbb{E}\xi_1\xi_2\xi_3 = \mathbb{E}\xi_1\mathbb{E}\xi_2\mathbb{E}\xi_3, \quad \text{provided that } \xi_{1:3} \text{ are independent}$$

However, the expectation $\mathbb{E}(\xi_1(\phi(\xi_2(\phi(\xi_3)))))$ is hard to compute.

Mean-field approximation The solution is to apply the mean-field approximation, the idea in which is to approximate intermediate variabeles by Gaussian random variables.

Proposition 3.1 (Informal Claim)**.** Suppose that the $(\ell-1)$-th layer has $d_{\ell-1}$ neurons, which is denoted as $h^{\ell-1}_{1:d_{\ell-1}}$. The variables for the pre-activation in ℓ-th layer, denoted as $h^\ell = W^\ell\phi(h^{\ell-1})$ (or $h^\ell_i = \sum_{j=1}^{d_{\ell-1}} W^\ell_{ij}\phi(h^{\ell-1}_j)$ for $i = 1 : d_\ell$) can be approximated by Gaussian random variables, provided that $d_{\ell-1}$ is very large. As $d_{\ell-1} \to \infty$, the variable $h^\ell_{1:d_\ell}$ converge to Gaussain.

This method is first applied in the paper (Poole *et al.*, 2016) to analysis the propagation of the variance of pre-activations. This technique is a novel application of the central limit theorem (Billingsley, 1986):

Theorem 3.1 (Lyapunov Central Limit Theorem)**.** The sum of n independent random variables $X_1, \ldots, X_n$ converges to a Gaussian random variable as $n \to \infty$, provided that Lyapunov's condition is satisfied.

Now we apply the mean-field approximation technique in some examples, although the ∞-width neural network assumption is not satisfied:

Example 3.2. Consider the case where $d = 1$ and $L = 3$. We have

$$h^1 = w^1\phi(x), h^2 = w^2\phi(h^1), h^3 = w^3\phi(h^2)$$

with $w^{1:3} \sim \mathcal{N}(0, \sigma_w^2)$ and $x \sim \mathcal{N}(0, q_{\text{in}})$. In order to control the signal strength of h^3, it suffices to control its variance (why?).

By the hint in the note[1], we have

$$\mathbb{E}[h^1] = 0, \quad \text{Var}(h^1) = \mathbb{E}[\|wx\|^2] = \sigma_w^2 \int_{-\infty}^{\infty} \phi(t\sqrt{q_{\text{in}}})^2 \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{t^2}{2}\right) \mathrm{d}t$$

Similarly, we have

$$q^2 = \sigma_w^2 \int_{-\infty}^{\infty} \phi(t\sqrt{q^1})^2 \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{t^2}{2}\right) \mathrm{d}t,$$

where $q^i \triangleq \text{Var}(h^i)$ for $i = 1, 2$.

The general form for q^ℓ can be computed recursively:

$$q^\ell = T(q^{\ell-1}) \triangleq \sigma_w^2 \int_{-\infty}^{\infty} \phi(t\sqrt{q^{\ell-1}})^2 \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{t^2}{2}\right) \mathrm{d}t$$

Remark 3.2. It's not true for the case $d = 1$, since h^ℓ are actually not Gaussian. In multi-dimension case, this statement becomes more rigorous. For neural network with infinite-width, if $\mathbb{E}w_{ij}^\ell = 0$ and $\text{Var}(w_{ij}^\ell) = \sigma_w^2$, then

$$q^\ell = T(q^{\ell-1}; \sigma_w^2) \triangleq \sigma_w^2 \int \phi(t\sqrt{q^{\ell-1}})^2 \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{t^2}{2}\right) \mathrm{d}t \tag{3.1}$$

[1] Ruoyu Sun, Mean-field Approximation: Step-by-Step Approach

Example 3.3. We can use the Eq. (3.1) to find proper σ_w^2 such that $\{q^\ell\}$ does not explode or vanish.

- For linear neural network, Eq. (3.1) reduces to
$$q^\ell = \sigma_w^2 q^{\ell-1}.$$
Therefore, we choose $\sigma_w^2 = 1$, which is the Xavier initialization.
- For relu activation, Eq. (3.1) reduces to
$$q^\ell = \frac{1}{2}\sigma_w^2 q^{\ell-1}$$
Therefore, we choose $\sigma_w^2 = 2$, which is the Kaiming initialization.
- For other types of activation, e.g., $q^\ell = (q^{\ell-1})^2\sigma_w^2$, it's difficult to pick σ_w^2 only to get the desired result. In this case, we pick $\sigma_w^2 = 1$ and $q^0 = 1$. Note that $q^0 = \sigma_w^2\|x\|^2\frac{1}{d_0}$. Therefore, the key message is that we should always scale the input vector to make $q^0 = 1$.

3.4 Dynamical Isometry

The motivation is that for neural network with input $x \in \mathcal{X}$ and output $y = \mathcal{W}(x) \in \mathcal{Y}$, we want to have $\|x\| \approx \|y\|$.

Bibliography Dynamical Isometry has gain popularity in the research of deep learning. It is first raised in the paper (Pennington *et al.*, 2017) to analysis the behavior of deep non-linear networks. Following the previous work, the same authors also extends the dynamical isometry theory to a large number of activation functions (Pennington *et al.*, 2018). This theory also has some applications in practical training. The paper (Xiao *et al.*, 2018) applies dynamical isometry theory to train 10000-layer vanilla CNN without tricks such as Batch Normalization or ResNet. The paper (Li and Nguyen, 2019) applies this theory to tranin a deep autoencoder without any other tricks as well. The paper (Gilboa *et al.*, 2019) applies this theory to train 10000-long LSTM. The paper (Zhang *et al.*, 2019) proposes a fixed-update initialization scheme to train ResNet without Batch Normalization, and the insights are based on this theory as well.

3.4.1 Dynamical Isometry for Linear Networks

For linear network case, given $y = W^L \cdots W^1 x$, the goal is to ensure $\|y\| \approx \|x\|$. The previous technique is to choose W^ℓ to be a Gaussain matrix.

There is another perspective from singular values. It suffices to let singular values of W^ℓ to be close to 1. The simplest solution (Saxe *et al.*, 2014) is to set W^ℓ is an orthogonal matrix, provided that $d_0 = \cdots = d_L$, and the intuition is that the norm is orthogonal invariant.

Proposition 3.2 (Key Observation for Orthogonality)**.** If W^ℓ are orthogonal matrices for $\ell = 1, \ldots, L$, and $d \triangleq d_0 = \cdots = d_L$, then

$$\begin{aligned} \|z^\ell\| &= \|x\|, \quad \forall \ell \\ \|e^\ell\| &= \|e\|, \quad \forall \ell \\ \left\| \frac{\partial F}{\partial W^\ell} \right\| &= 2\|e\|\|x\|, \quad \forall \ell \end{aligned}$$

The paper (Saxe *et al.*, 2014) also runs simulation and finds that the orthogonal initialization in deep neural linear network, unlike the case for Gaussian initialization, enjoys *depth-independent* training time.

The goal that we want to achieve, i.e., all singular values of $W^L \cdots W^0$ are close to 1, is called the *dynamical isometry*.

Remark 3.3. There are two reasons that people prefer not to use orthogonal initialization. The first is that the initialization for CNN is totally different since it is not a fully connected neural network; the second is that the case for non-linear network will change a lot.

In the next lecture we will talk about the non-linear network. In particular, we will talk about the *DeltaOrthogonal* frequently used in tensorflow.

References

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

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*.

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.

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*.

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.

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*.

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.

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.