# 5

# ResNet Initialization and Landscape Analysis

## 5.1 Reviewing

- Three major tricks for State-of-The-art (SoTa) Deep Learning Training:
  - *Initialization*;
  - *Batch Normalization* (BN);
  - *ResNet* (or other architectures).
- Key ideas of Batch Normalization:
  - Motivation: Reduce condition number by normalization.
  - Treat Normalization as $\phi$, a non-linear transformation.
- Error Decomposition (*An* perspective from Prof. Ruoyu Sun):

$$\text{Testing Error} = \begin{cases} \text{Representation Error} \\ \text{Optimization Error} \end{cases}$$

$$\text{Optimization Error} = \begin{cases} \text{Finite-time Error} \\ \infty\text{-time Error} \end{cases}$$

- How to train a 10000-layer neural network with only one trick "*initilization*"?
  - Special Orthogonal Initial point (DeltaOrthogonal);
  - Based on the idea of Dynamical Isometry.

## 5.2 Initialization for ResNet

The architecture for classic ResNet is presented below:

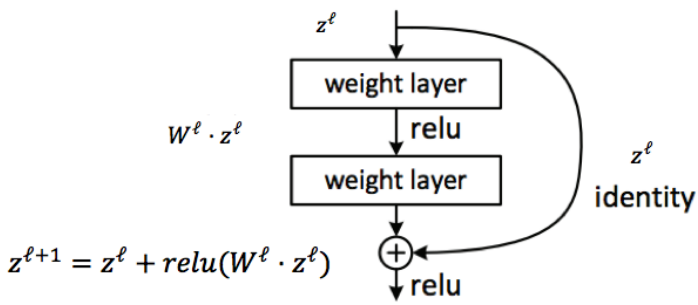


**Figure 5.1:** ResNet framework with Relu activation

How to initialize the ResNet architecture? Can we follow the previous knowleage for handling FNN? Let's give some analysis by first considering the simple linear network (with the same width).

**Example 5.1.** Let's consider the $L$-layer linear network with the same width $d$, i.e., the nonlinear activation is an identity operator. As a result, the output-input function is expressed as:

$$y = (I + W^L) \cdots (I + W^1)x, \quad \text{where } W^\ell \in \mathbb{R}^{d \times d}$$

By the Lecture 2 knowledge, we know that Xavier initialization works for fully connected linear neural networks. Now we perform the simulation of Xavier initialization for ResNet. The matlab code for the toy example where $L = 10$, $d = 100$, and input $x$ is all-one vector, is presented below.

```
clear;
L = 10;
d = 100; % dimension for weight matrix W
maxit = 10; % 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 + x;
    end
    rato = norm(x)/norm0
end
```

Unfortunately, we find that $\|y\|/\|x\| \approx 10^{14}$ in this case. Now the question is the following:

1. Why does (He *et al.*, 2016) succeed?

   It seems that combining the trick *Batch Normalization* saves him

2. Why does our simulation fail?

   In previous toy example, Xavier initialization works when the output in each layer is the multiplication of the input with a Gaussian matrix, but in our example, $(I + W^\ell)$ is not Gaussian.

Therefore, we need to re-derive the whole *initialization theory* for ResNet, i.e., for what kind of $W^\ell$, we have $\|y\|/\|x\| \approx \mathcal{O}(1)$?

1. Choose $W^\ell = \mathcal{N}(0, 1/d) - I$, then the original case for linear FNN is recovered, but we no longer enjoy the advantage of the new architecture.

2. Consider the case where $d = 1$ first, i.e., the output-input function is expressed as

   $$y = (1 + w^L) \cdots (1 + w^1)x$$

   It's feasible to choose $w^\ell = 0$ for all $\ell$, but it does not have much representation power, since any point near this initial point will

be strongly attracted to it.[1] It's reasonable to assume that $w^\ell$ follows Gaussian distribution, i.e., $w^\ell = \mathcal{N}(0, \cdot c)$. The question turns to the choice of $c$. In this case,

$$\mathbb{E}[\|y\|^2] = (1+c)^L \|x\|^2$$

Therefore, we should choose $c = \frac{1}{L}$, which implies $\mathbb{E}[\|y\|^2] = \mathcal{O}(\|x\|^2)$.

For more general $d$, we should make $W_{i,j}^\ell = \mathcal{N}(0, \frac{1}{d} \cdot \frac{1}{L})$. The proof outline is as follows:

- Consider how the term $\mathbb{E}[\|y\|^2]$ scales with $\|x\|^2$. Observe that

$$\mathbb{E}[\|z^\ell\|^2 \mid z^{\ell-1}] = (z^{\ell-1})^{\mathrm{T}} \left[\mathbb{E}(I + W^\ell)^2\right] (z^{\ell-1})$$

- It's easy to show that when $W_{i,j}^\ell = \mathcal{N}(0, \frac{1}{d} \cdot \frac{1}{L})$,

$$\mathbb{E}(I+W)^2 = \mathbb{E}(I + 2W + W^2) = I + \mathbb{E}W^2 = (1 + 1/L)I.$$

- Therefore,

$$\mathbb{E}[\|y\|^2] = (1+1/L)^L I \cdot \|x\|^2 = \mathcal{O}(\|x\|^2).$$

**Bibilogrphy** In practice, people notice that ResNet performs much better than standard architectures when networks are very deep. The paper (Balduzzi *et al.*, 2017) gives (partial) explanations for this phenomenon, and claims that one reason is that the gradients in ResNet (with BN) are far more resistant to shattering, which decays sublinearly. Then this paper proposes a new initialization scheme accordingly, which outperforms the classic He-initialization for standard architectures.

**Scaling of the Residuals** The formal analysis in the paper (Balduzzi *et al.*, 2017) is as follows. Consider a (variant of) ResNet[2] framework for Batch Normalization Disabled case:

$$z^{\ell+1} = \alpha(z^\ell + \beta \cdot W^\ell \cdot \mathrm{relu}(z^{\ell-1}))$$

where $\alpha$ and $\beta$ are rescaling factors.

[1] The formal definition of the strong attraction is presented in the paper (Zhang *et al.*, 2000).

[2] There are several variants of classic ResNet.

1. With classic initilization without batch normalization trick, set $\alpha = \beta = 1$, then the variance of the gradient at $z^\ell[i]$ is $2^L$.

2. $\alpha$-scaling: A solution to the exploding variance of resnets is to rescale layers $\alpha = 1/\sqrt{2}$, then $\mathrm{Var}(z^\ell[i]) = 1$.

3. $\beta$-scaling: In practice, $\alpha$-rescaling is not used. Instead, combining the bacth normalization trick and $\beta$-scaling[3] gives $\mathrm{Var}(z^\ell[i]) = \beta^2(L-1)+1$. Furthermore, when $\beta = 1/\sqrt{L-1}$, we see that the variance keeps constant:

$$\mathrm{Var}(z^\ell[i]) \equiv 2.$$

**Is normalization fundamental?** It's believed that normalization trick is fundamental in state-of-the-art training. The paper (Zhang *et al.*, 2019) challenges this belief by proposing *fixed-update initialization* scheme on *ResNet* to achieve state-of-the-art performance in image classification and machine translation. This initialization is motivated by solving the gradient explosion/vanishing problem at the beginning of training via properly rescaling a standard initialization. This is the only work that achieves such good results without the normalization method

This work is amazing for two reasons:

1. Batch Normalization can be annoying, since its practical implementation can have many bugs, especially for ResNet.

2. It shows the importance of basic logic, i.e., the combination of previous work could have gained more advantages. The paper (Glorot and Bengio, 2010) first proposes Xavier-initialization; and the same authors propose Relu function in (Glorot *et al.*, 2010). Combining these two tricks, the Kaiming initialization is proposed in (He *et al.*, 2015); and the same authors propose ResNet in (He *et al.*, 2016). Moreover, this paper proposes the Fixup initialization by combining these two tricks that enjoy more advantages.

The theme of this course so far: discussion on how the theory shaped the current practice.

[3]It's suggested in (Szegedy *et al.*, 2016) that $\beta \in [0.1, 0.3]$

## 5.3 Landscape of Neural-Nets

**Motivation** We are interested in when and why the non-convexity is not a big issue for the training of neural-nets. To answer this question, recall that the Optimization Error is decomposed into two kinds of errors:

$$\text{Optimization Error} \begin{cases} \text{finite-time error} \\ \infty\text{-time error} \end{cases}$$

It's believed that the major issue is due to the $\infty$-time error, since in practice most algorithms do converge in a reasonable time. The $\infty$-time error is about the optimality gap between the global minima and the local minima that the algorithm has converged, which is related to the landscape of the loss function for the training of neural-nets. This lecture will talk about both positive and negative results about the landscape of the loss function.

### 5.3.1 Positive Result: Linear Network has nice landscape

Consider the loss function for linear neural network with depth $L$:

$$F(\theta) = \|y - W^L \phi(W^{L-1}(\phi(\cdots W^1 x)))\|^2$$

The non-convexity of this loss function comes from the multi-layer and the non-linear activation. The case for $L = 2$ reduces to the Matrix Factorization problem:

$$F(\theta) = \|Y - W^2 W^1 x\|_F^2.$$

The landscape for matrix factorization is well-studied:

**Scalar Case Analysis** The loss function for the scalar case is given by:

$$F(u, v) = (1 - uv)^2, \qquad u, v \in \mathbb{R}.$$

The 3-D plot for this loss function is in Fig. 5.2.

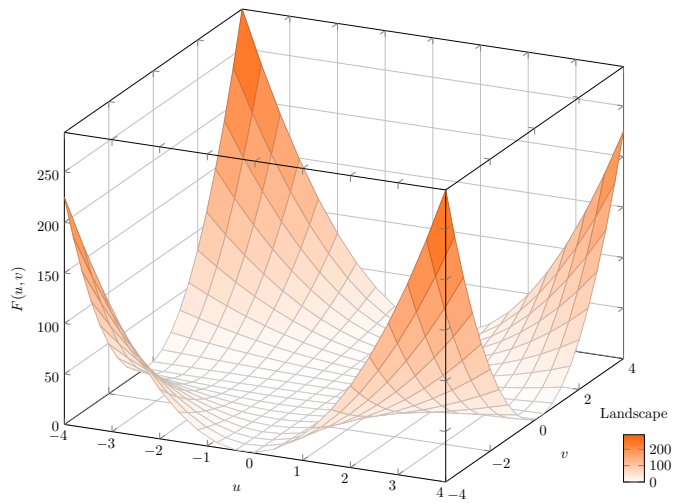


**Figure 5.2:** 3-D plot for the loss function $F(u, v) = (1 - uv)^2$

From the landscape we can see that every local minima is a global-minima, i.e., there is no bad local minima. We give a proof for this claim:

1. Step 1: By taking gradient equal to zero, we find stationary points must satisfy
$$\text{either } uv = 1 \text{ or } u = v = 0$$
2. Step 2: It suffices to show that $(u, v) = (0, 0)$ is not a local-minima. Note that
$$F(\epsilon_1, \epsilon_2) = (\epsilon_1\epsilon_2 - 1)^2 < 1$$
When $(\epsilon_1, \epsilon_2) > 0$ are sufficiently small, we find the function value decreases, i.e., $(0, 0)$ is not a local-minima.

**Remark 5.1.** However, starting from $(u, v) = (0, 0)$, the gradient descent gets stuck. Is this finding bad enough? Actually not. The paper (Lee *et al.*, 2016) shows that gradient descent, if converges, it only converges to a local minimizer, almost surely with random initialization. The recent trend for non-convex optimization is not satisfied with the convergence to first order stationary point[4], but the convergence to second order

[4] $\epsilon$-First order stationary point (FOSP) means $\|\nabla f(x)\| \le \epsilon$

stationary point.[5] The reason is that empirically second order stationary point is as good as global minima. The tutorial (*Understanding nonconvex optimization* n.d.) in ISIT2019 gives summarization on the recent progress in non-convex optimization, which is highly recommended to read.

However, for people working on optimization, focusing on second order stationary point is not good enough, since high-order saddle points do exist for deep neural-nets. Now let's focus on the second order stationary point only.

**Proposition 5.1.** Consider the function $F(u) = \|M - uu^{\mathrm{T}}\|_F^2$, where $M$ is PSD, any SOSP of this function is global-minima.

*Proof.* 1. Step 1: Check the gradient:

$$\nabla_u F = 4(M - uu^{\mathrm{T}})u = 0 \implies Mu = (u^{\mathrm{T}}u)u$$

Therefore, if $u$ is SOSP, then $(\|u\|^2, u)$ is an eigen-pair of $M$.

2. Check the Hessian of the SOSP $u$:

$$\nabla^2 F(u) = 4(u^{\mathrm{T}} + \|u\|^2 I - M) \succeq 0.$$

Combining these steps, we can show that $\|u\|^2 = \lambda_{\max}(M)$, i.e., $u$ is a global-minima. □

**Re-thinking Convexity** From this proof we can partially answer why many people work on convex optimization, and why non-convex optimization is not scary:

1. By sub-gradient inequality, we can show that FOSP together with convexity implies global-minima. Therefore, it suffices to design algorithms searching for FOSP.

2. Define $G \triangleq \{\text{SOSP implies global-minima}\}$. We find many instances belong to the set $G$:

$$\{\text{convex problems}\} \subseteq G, \qquad \{\min_u \|M - uu^{\mathrm{T}}\|_F^2\} \subseteq G.$$

Therefore, for non-convex problems belonging to the set $G$, it suffices to design algorithms searching for SOSP.

[5] Second order stationary point (SOSP) means $\nabla f(x) = 0$ and $\nabla^2 f(x) \succeq 0$.

**Bibliography** The paper (Baldi and Hornik, 1989) shows that any SOSP for the 2-layer linear network quadratic loss function is global-minima under mild conditions; the paper (Kawaguchi, 2016) shows that any SOSP for the deep linear network quadratic loss function is global-minima under mild conditions. There are many later works extending to other loss functions. Up to now, we find that multi-layer may not be an issue.

### 5.3.2 Negative Result: Nonlinearity doesn't necessarily imply Global-optimality for SOSP

Now consider the non-linear activation. For the 1-dimension case, suppose the loss function $F(w) = (y - \phi(wx))^2$. W.l.o.g., $x = y = 1$, which follows that

$$F(w) = (1 - \phi(w))^2.$$

We are interested in whether all SOSP are global-minimas.

**Relu Activation** Now we draw the landscape of $F(w)$ if $\phi(w) = \max\{w, 0\}$:

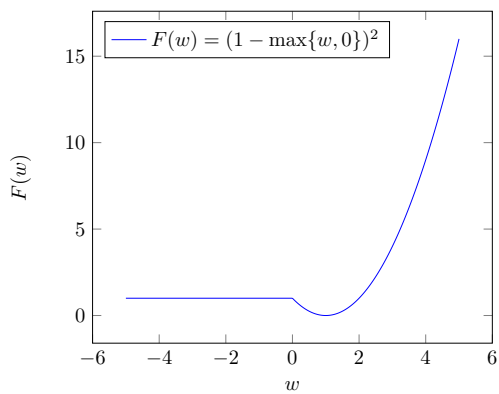


**Figure 5.3:** 2-D plot for the loss function $F(w) = (1 - w_+)^2$

We find that for $w < 0$, $w$ is still a SOSP, but no longer a global-minima. Let's try other kinds of activation functions.

- When $\phi(w) = w^2$, we find SOSPs are still the global-minima:

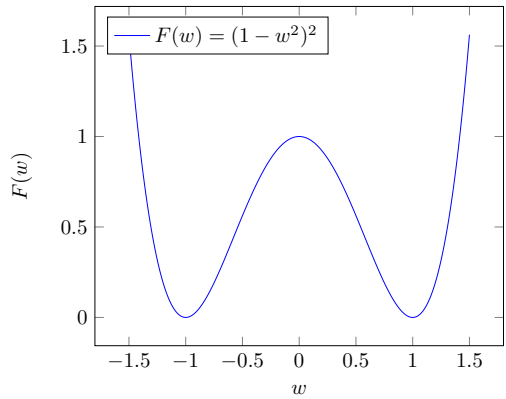


**Figure 5.4:** 2-D plot for the loss function $F(w) = (1 - w^2)^2$

- When $\phi(w) = \text{sigmoid}(w) = \frac{1}{1+e^{-w}}$, we find SOSP is still the global-minima:

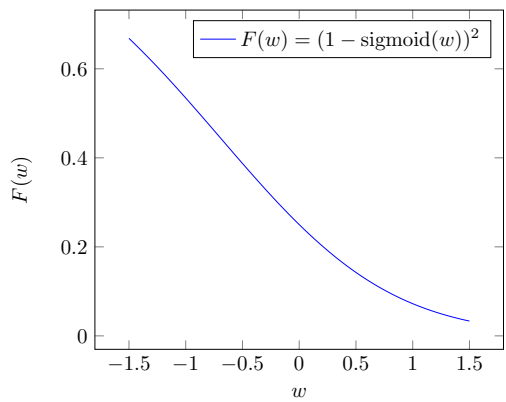


**Figure 5.5:** 2-D plot for the loss function $F(w) = (1 - \text{sigmoid}(w))^2$

Therefore, we conclude that the landscape of loss function is fine for most non-linear activations. but ReLu is not good in terms of landscape. Prof. Ruoyu Sun suggests that we can try Leaky Relu or Softplw if

Relu fails (e.g., for GAN training).

In the next lecture we will try the sum of loss functions, i.e.,

$$F(w) = (y_1 - \phi(w)x_1)^2 + (y_2 - \phi(w)x_2)^2.$$

The question is that if two functions both have good landscapes, does the summation still have a good landscape?

# References


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.

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

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.

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.

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.

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

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.

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, 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*.

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*.

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

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.

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.