



Lecture 07b

Initialization

DL4DS – Spring 2025

Agenda

- Project: [Project ideas](#), Teams, ...
- Quick tips on how to read a research paper
- Model Initialization
- Example code

Where we are



=== Foundational Concepts ===

- ✓ 02 -- Supervised learning refresher
- ✓ 03 -- Shallow networks and their representation capacity
- ✓ 04 -- Deep networks and depth efficiency
- ✓ 05 -- Loss function in terms of maximizing likelihoods
- ✓ 06 – Fitting models with different optimizers
- ✓ 07a – Gradients on deep models and backpropagation
- 07b – Initialization to avoid vanishing and exploding weights & gradients
- 08 – Measuring performance, test sets, overfitting and double descent
- 09 – Regularization to improve fitting on test sets and unseen data

=== Network Architectures and Applications ===

- 10 – Convolutional Networks
- 11 – Residual Networks
- 12 – Transformers
- Large Language and other Foundational Models
- Generative Models
- Graph Neural Networks
- ...

How to Read Research Papers

Reading Research Papers, Accelerating your Efficiency

1. Compile list of papers (research papers, Medium posts, Blogs, etc.). You can use tools like Zotero or Mendeley to organize your bibliography.
2. Skip around the list.
 - a. Glance through the list, maybe read only 10% of each paper.
 - b. Weed out duds.
 - c. Pick one of the papers to dive into.
3. 15-20 papers gives you basic understanding maybe enough to implement. 50-100 papers give you very solid understanding, can start research.
4. Shoot for reading 1-2 papers deeply per week. ⇒ Reading Group!

How to Read Research Papers - II

How to Read a Paper

Do it in multiple passes.

1. Title, abstract and figures. Especially the key figure(s).
2. Intro, Conclusions, Figures then skim the rest
3. Maybe skim or skip related work. Sometimes authors try to cite people that may be reviewers.
4. Read the paper but skim/skip the math.
5. Read whole thing but skip parts that don't make sense. Often even authors don't know what will make the biggest impact until later.
6. Go Deeper: Re-derive the math from memory.
7. Go Deeper: Re-implement the network from scratch.

How to Read Research Papers - III

Some questions to ask yourself.

- What did authors try to accomplish?
- What were the key elements of the approach?
- What can you use yourself?
- What other references do you want to follow?

Optional Exercise

Look up “[Deep Residual Learning for Image Recognition](#)” on scholar.google.com.

Take 7 minutes to scan and read the paper in multiple passes as previously discussed. Highlight key points and graphs.

Take a few minutes and jot down the key takeaways.

Model Initialization

- The need for weights initialization
- Expectations Refresher
- The He (Kaiming) Initialization

Initialization

- Consider standard building block of NN in terms of pre-activations:

$$\begin{aligned}\mathbf{f}_k &= \boldsymbol{\beta}_k + \boldsymbol{\Omega}_k \mathbf{h}_k \\ &= \boldsymbol{\beta}_k + \boldsymbol{\Omega}_k \mathbf{a}[\mathbf{f}_{k-1}]\end{aligned}$$

- How do we initialize the biases and weights?
- Equivalent to choosing starting point in our gradient descent searches

Forward Pass

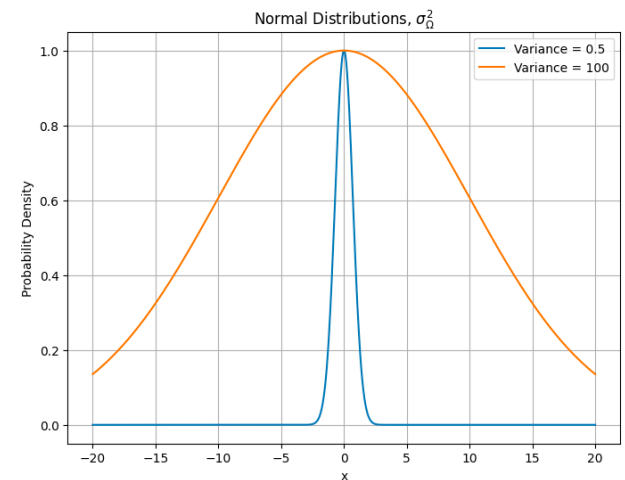
- Consider standard building block of NN in terms of *pre-activations*:

$$\begin{aligned}\mathbf{f}_k &= \boldsymbol{\beta}_k + \boldsymbol{\Omega}_k \mathbf{h}_k \\ &= \boldsymbol{\beta}_k + \boldsymbol{\Omega}_k a[\mathbf{f}_{k-1}]\end{aligned}$$

- Set all the biases to 0

$$\boldsymbol{\beta}_k = \mathbf{0}$$

- Set weights to be normally distributed
 - mean 0
 - variance σ_{Ω}^2

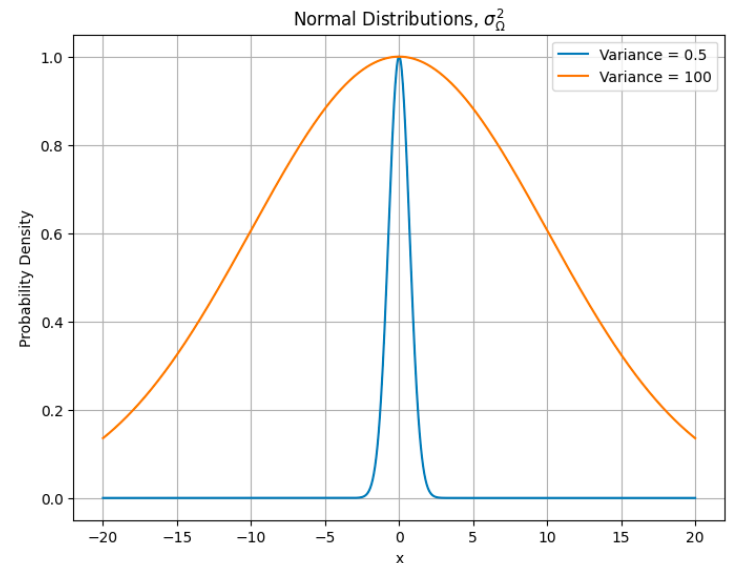


- What will happen as we move through the network if σ_{Ω}^2 is very small?
- What will happen as we move through the network if σ_{Ω}^2 is very large?

Backward Pass

$$\frac{\partial \ell_i}{\partial \mathbf{f}_{k-1}} = \mathbb{I}[\mathbf{f}_{k-1} > 0] \odot \left(\mathbf{\Omega}_k^T \frac{\partial \ell_i}{\partial \mathbf{f}_k} \right), \quad k \in \{K, K-1, \dots, 1\} \quad (7.13)$$

- What will happen as we propagate backwards through the network if σ_{Ω}^2 is very small?
- What will happen as we propagate backwards through the network if σ_{Ω}^2 is very large?



Initialize weights to different variances

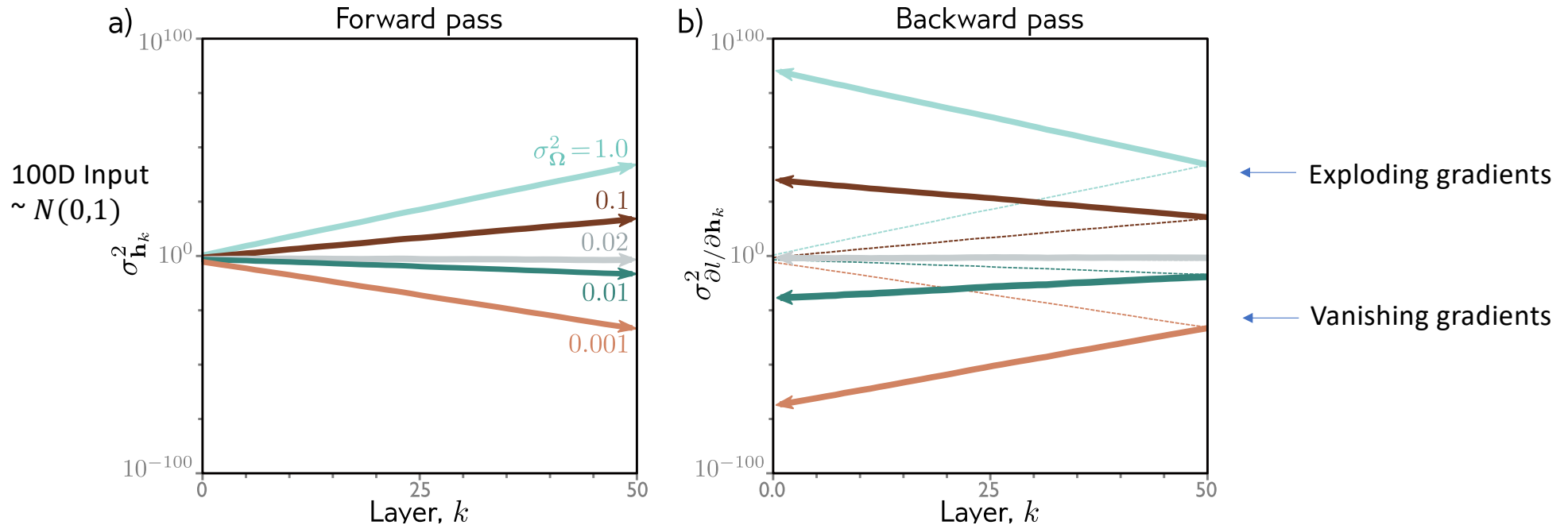


Figure 7.4 Weight initialization. Consider a deep network with 50 hidden layers and $D_h = 100$ hidden units per layer. The network has a 100 dimensional input \mathbf{x} initialized with values from a standard normal distribution, a single output fixed at $y = 0$, and a least squares loss function. The bias vectors β_k are initialized to zero and the weight matrices Ω_k are initialized with a normal distribution with mean zero and five different variances $\sigma_{\Omega}^2 \in \{0.001, 0.01, 0.02, 0.1, 1.0\}$. a)

How do we initialize weights to keep variance stable across layers?

Aim: keep variance same between two layers

$$\mathbf{f}' = \boldsymbol{\beta} + \boldsymbol{\Omega}\mathbf{h}$$

$$\mathbf{h} = \mathbf{a}[\mathbf{f}],$$

Definition of variance:

$$\sigma_{f'}^2 = \mathbb{E}[(f'_i - \mathbb{E}[f'_i])^2]$$

Agenda

- The need for weights initialization
- **Expectations Refresher**
- The He (Kaiming) Initialization

Expectations

$$\mathbb{E}[g[x]] = \int g[x]Pr(x)dx,$$

Interpretation: what is the average value of $g[x]$ when taking into account the probability of x ?

Consider discrete case and assume uniform probability so calculating $g[x]$ reduces to taking average:

$$\mathbb{E}[g[x]] \approx \frac{1}{N} \sum_{n=1}^N g[x_n^*] \quad \text{where} \quad x_n^* \sim Pr(x)$$

Common Expectation Functions

Function $g[\bullet]$	Expectation
x	mean, μ
x^k	k th moment about zero
$(x - \mu)^k$	k th moment about the mean
$(x - \mu)^2$	variance
$(x - \mu)^3$	skew
$(x - \mu)^4$	kurtosis

Table B.1 Special cases of expectation. For some functions $g[x]$, the expectation $\mathbb{E}[g[x]]$ is given a special name. Here we use the notation μ_x to represent the mean with respect to random variable x .

Rules for manipulating expectation

$$\mathbb{E}[k] = k$$

$$\mathbb{E}[k \cdot g[x]] = k \cdot \mathbb{E}[g[x]]$$

$$\mathbb{E}[f[x] + g[x]] = \mathbb{E}[f[x]] + \mathbb{E}[g[x]]$$

$$\mathbb{E}[f[x]g[y]] = \mathbb{E}[f[x]]\mathbb{E}[g[y]] \quad \text{if } x, y \text{ independent}$$

Agenda

- The need for weights initialization
- Expectations Refresher
- The He (Kaiming) Initialization

Aim: keep variance same between two layers

$$\mathbf{h} = \mathbf{a}[\mathbf{f}],$$
$$\mathbf{f}' = \boldsymbol{\beta} + \boldsymbol{\Omega}\mathbf{h}$$

Definition of variance:

$$\sigma_{f'_i}^2 = \mathbb{E}[(f'_i - \mathbb{E}[f'_i])^2]$$

Now let's prove:

$$\mathbb{E} [(x - \mu)^2] = \mathbb{E}[x^2] - \mathbb{E}[x]^2$$

Keeping in mind:

$$\mathbb{E}[x] = \mu$$

Rule 1: $\mathbb{E}[k] = k$

Rule 2: $\mathbb{E}[k \cdot g[x]] = k \cdot \mathbb{E}[g[x]]$

Rule 3: $\mathbb{E}[f[x] + g[x]] = \mathbb{E}[f[x]] + \mathbb{E}[g[x]]$

Def'n $\mathbb{E}[x] = \mu$

$$\mathbb{E}[(x - \mu)^2] = \mathbb{E}[x^2 - 2x\mu + \mu^2]$$

Rule 1: $\mathbb{E}[k] = k$

Rule 2: $\mathbb{E}[k \cdot g[x]] = k \cdot \mathbb{E}[g[x]]$

Rule 3: $\mathbb{E}[f[x] + g[x]] = \mathbb{E}[f[x]] + \mathbb{E}[g[x]]$

Def'n $\mathbb{E}[x] = \mu$

$$\begin{aligned}\mathbb{E}[(x - \mu)^2] &= \mathbb{E}[x^2 - 2x\mu + \mu^2] \\ &= \mathbb{E}[x^2] - \mathbb{E}[2x\mu] + \mathbb{E}[\mu^2]\end{aligned}$$

Rule 1: $\mathbb{E}[k] = k$


Rule 2: $\mathbb{E}[k \cdot g[x]] = k \cdot \mathbb{E}[g[x]]$

Rule 3: $\mathbb{E}[f[x] + g[x]] = \mathbb{E}[f[x]] + \mathbb{E}[g[x]]$

Def'n $\mathbb{E}[x] = \mu$

$$\begin{aligned}\mathbb{E}[(x - \mu)^2] &= \mathbb{E}[x^2 - 2x\mu + \mu^2] \\ &= \mathbb{E}[x^2] - \mathbb{E}[2x\mu] + \mathbb{E}[\mu^2] \\ &= \mathbb{E}[x^2] - 2\mu\mathbb{E}[x] + \mu^2\end{aligned}$$

Rule 1:	$\mathbb{E}[k] = k$
Rule 2:	$\mathbb{E}[k \cdot g[x]] = k \cdot \mathbb{E}[g[x]]$
Rule 3:	$\mathbb{E}[f[x] + g[x]] = \mathbb{E}[f[x]] + \mathbb{E}[g[x]]$
Def'n	$\mathbb{E}[x] = \mu$




$$\begin{aligned}\mathbb{E}[(x - \mu)^2] &= \mathbb{E}[x^2 - 2x\mu + \mu^2] \\ &= \mathbb{E}[x^2] - \mathbb{E}[2x\mu] + \mathbb{E}[\mu^2] \\ &= \mathbb{E}[x^2] - 2\mu\mathbb{E}[x] + \mu^2 \\ &= \mathbb{E}[x^2] - 2\mu^2 + \mu^2\end{aligned}$$

Rule 1:	$\mathbb{E}[k] = k$
Rule 2:	$\mathbb{E}[k \cdot g[x]] = k \cdot \mathbb{E}[g[x]]$
Rule 3:	$\mathbb{E}[f[x] + g[x]] = \mathbb{E}[f[x]] + \mathbb{E}[g[x]]$
Def'n	$\mathbb{E}[x] = \mu$

$$\begin{aligned}\mathbb{E}[(x - \mu)^2] &= \mathbb{E}[x^2 - 2x\mu + \mu^2] \\ &= \mathbb{E}[x^2] - \mathbb{E}[2x\mu] + \mathbb{E}[\mu^2] \\ &= \mathbb{E}[x^2] - 2\mu\mathbb{E}[x] + \mu^2 \\ &= \mathbb{E}[x^2] - 2\mu^2 + \mu^2 \\ &= \mathbb{E}[x^2] - \mu^2\end{aligned}$$

Rule 1:	$\mathbb{E}[k] = k$
Rule 2:	$\mathbb{E}[k \cdot g[x]] = k \cdot \mathbb{E}[g[x]]$
Rule 3:	$\mathbb{E}[f[x] + g[x]] = \mathbb{E}[f[x]] + \mathbb{E}[g[x]]$
Def'n	$\mathbb{E}[x] = \mu$



$$\begin{aligned}\mathbb{E}[(x - \mu)^2] &= \mathbb{E}[x^2 - 2x\mu + \mu^2] \\ &= \mathbb{E}[x^2] - \mathbb{E}[2x\mu] + \mathbb{E}[\mu^2] \\ &= \mathbb{E}[x^2] - 2\mu\mathbb{E}[x] + \mu^2 \\ &= \mathbb{E}[x^2] - 2\mu^2 + \mu^2 \\ &= \mathbb{E}[x^2] - \mu^2 \\ &= \mathbb{E}[x^2] - E[x]^2\end{aligned}$$

Aim: keep variance same between two layers

$$\mathbf{f}' = \boldsymbol{\beta} + \boldsymbol{\Omega}\mathbf{h}$$

$$\mathbf{h} = \mathbf{a}[\mathbf{f}],$$

$$\sigma_{f'}^2 = \mathbb{E}[(f'_i - \mathbb{E}[f'_i])^2]$$

$$\sigma_{f'}^2 = \mathbb{E}[f_i'^2] - \mathbb{E}[f_i']^2$$

$$\longrightarrow \mathbb{E}[(x - \mu)^2] = \mathbb{E}[x^2] - \mathbb{E}[x]^2$$

Aim: keep variance same between two layers

$$\mathbf{f}' = \boldsymbol{\beta} + \boldsymbol{\Omega}\mathbf{h}$$

$$\mathbf{h} = \mathbf{a}[\mathbf{f}],$$

$$\sigma_{f'}^2 = \mathbb{E}[(f'_i - \mathbb{E}[f'_i])^2]$$

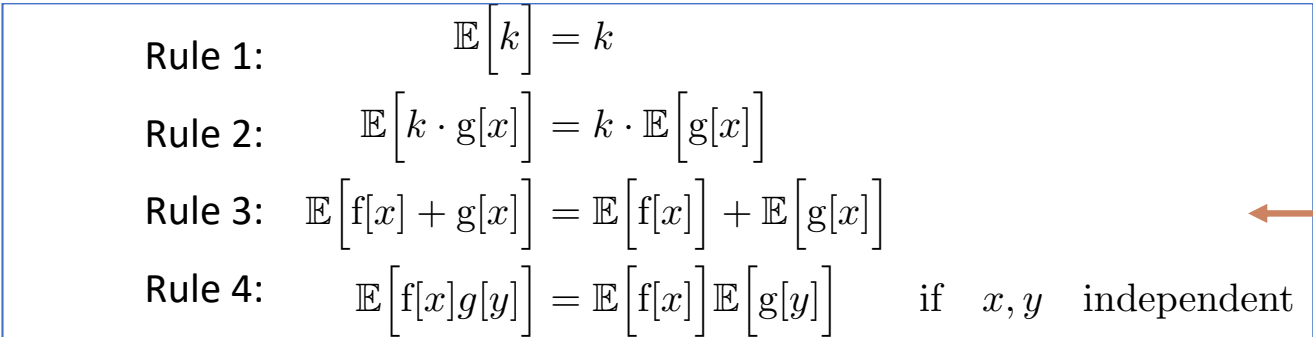
$$\sigma_{f'}^2 = \mathbb{E}[f_i'^2] - \mathbb{E}[f_i']^2$$

Aim: keep variance same between two layers


$$\mathbf{f}' = \boldsymbol{\beta} + \boldsymbol{\Omega}\mathbf{h}$$

Consider the mean of the pre-activations:

$$\mathbb{E}[f'_i] = \mathbb{E} \left[\beta_i + \sum_{j=1}^{D_h} \Omega_{ij} h_j \right]$$

- Rule 1: $\mathbb{E}[k] = k$
- Rule 2: $\mathbb{E}[k \cdot g[x]] = k \cdot \mathbb{E}[g[x]]$
- Rule 3: $\mathbb{E}[f[x] + g[x]] = \mathbb{E}[f[x]] + \mathbb{E}[g[x]]$
- Rule 4: $\mathbb{E}[f[x]g[y]] = \mathbb{E}[f[x]]\mathbb{E}[g[y]]$ if x, y independent
- 

$$\begin{aligned}\mathbb{E}[f'_i] &= \mathbb{E}\left[\beta_i + \sum_{j=1}^{D_h} \Omega_{ij} h_j\right] \\ &= \mathbb{E}[\beta_i] + \sum_{j=1}^{D_h} \mathbb{E}[\Omega_{ij} h_j]\end{aligned}$$

- Rule 1: $\mathbb{E}[k] = k$
- Rule 2: $\mathbb{E}[k \cdot g[x]] = k \cdot \mathbb{E}[g[x]]$
- Rule 3: $\mathbb{E}[f[x] + g[x]] = \mathbb{E}[f[x]] + \mathbb{E}[g[x]]$
- Rule 4: $\mathbb{E}[f[x]g[y]] = \mathbb{E}[f[x]]\mathbb{E}[g[y]]$ if x, y independent
- 

$$\begin{aligned}\mathbb{E}[f'_i] &= \mathbb{E}\left[\beta_i + \sum_{j=1}^{D_h} \Omega_{ij} h_j\right] \\ &= \mathbb{E}[\beta_i] + \sum_{j=1}^{D_h} \mathbb{E}[\Omega_{ij} h_j] \\ &= \mathbb{E}[\beta_i] + \sum_{j=1}^{D_h} \mathbb{E}[\Omega_{ij}] \mathbb{E}[h_j]\end{aligned}$$

Rule 1:	$\mathbb{E}[k] = k$
Rule 2:	$\mathbb{E}[k \cdot g[x]] = k \cdot \mathbb{E}[g[x]]$
Rule 3:	$\mathbb{E}[f[x] + g[x]] = \mathbb{E}[f[x]] + \mathbb{E}[g[x]]$
Rule 4:	$\mathbb{E}[f[x]g[y]] = \mathbb{E}[f[x]]\mathbb{E}[g[y]]$ if x, y independent

$$\begin{aligned}
\mathbb{E}[f'_i] &= \mathbb{E} \left[\beta_i + \sum_{j=1}^{D_h} \Omega_{ij} h_j \right] \\
&= \mathbb{E}[\beta_i] + \sum_{j=1}^{D_h} \mathbb{E}[\Omega_{ij} h_j] \\
&= \mathbb{E}[\beta_i] + \sum_{j=1}^{D_h} \mathbb{E}[\Omega_{ij}] \mathbb{E}[h_j] \\
&= 0 + \sum_{j=1}^{D_h} 0 \cdot \mathbb{E}[h_j] = 0
\end{aligned}$$

Set all the biases to 0

Weights normally distributed
mean 0
variance σ_{Ω}^2

Aim: keep variance same between two layers

$$\mathbf{f}' = \boldsymbol{\beta} + \boldsymbol{\Omega}\mathbf{h}$$

$$\mathbf{h} = \mathbf{a}[\mathbf{f}],$$

$$\sigma_{f'}^2 = \mathbb{E}[(f'_i - \mathbb{E}[f'_i])^2]$$

$$\sigma_{f'}^2 = \mathbb{E}[f_i'^2] - \mathbb{E}[f_i']^2 = \mathbb{E}[f_i'^2]$$

0

Rule 1:	$\mathbb{E}[k] = k$
Rule 2:	$\mathbb{E}[k \cdot g[x]] = k \cdot \mathbb{E}[g[x]]$
Rule 3:	$\mathbb{E}[f[x] + g[x]] = \mathbb{E}[f[x]] + \mathbb{E}[g[x]]$
Rule 4:	$\mathbb{E}[f[x]g[y]] = \mathbb{E}[f[x]]\mathbb{E}[g[y]]$ if x, y independent

$$\begin{aligned} \sigma_{f'}^2 &= \mathbb{E}[f_i'^2] - \mathbb{E}[f_i']^2 \\ &= \mathbb{E} \left[\left(\beta_i + \sum_{j=1}^{D_h} \Omega_{ij} h_j \right)^2 \right] - 0 \end{aligned}$$

Set all the biases to 0

Weights normally distributed
 mean 0
 variance σ_{Ω}^2

Rule 1:	$\mathbb{E}[k] = k$
Rule 2:	$\mathbb{E}[k \cdot g[x]] = k \cdot \mathbb{E}[g[x]]$
Rule 3:	$\mathbb{E}[f[x] + g[x]] = \mathbb{E}[f[x]] + \mathbb{E}[g[x]]$
Rule 4:	$\mathbb{E}[f[x]g[y]] = \mathbb{E}[f[x]]\mathbb{E}[g[y]]$ if x, y independent

$$\begin{aligned} \sigma_{f'}^2 &= \mathbb{E}[f_i'^2] - \mathbb{E}[f_i']^2 \\ &= \mathbb{E}\left[\left(\beta_i + \sum_{j=1}^{D_h} \Omega_{ij} h_j\right)^2\right] - 0 \\ &= \mathbb{E}\left[\left(\sum_{j=1}^{D_h} \Omega_{ij} h_j\right)^2\right] \end{aligned}$$

Set all the biases to 0 

Weights normally distributed

mean 0

variance σ_{Ω}^2

Rule 1:	$\mathbb{E}[k] = k$
Rule 2:	$\mathbb{E}[k \cdot g[x]] = k \cdot \mathbb{E}[g[x]]$
Rule 3:	$\mathbb{E}[f[x] + g[x]] = \mathbb{E}[f[x]] + \mathbb{E}[g[x]]$
Rule 4:	$\mathbb{E}[f[x]g[y]] = \mathbb{E}[f[x]]\mathbb{E}[g[y]]$ if x, y independent



$$\begin{aligned} \sigma_{f'}^2 &= \mathbb{E}[f_i'^2] - \mathbb{E}[f_i']^2 \\ &= \mathbb{E} \left[\left(\beta_i + \sum_{j=1}^{D_h} \Omega_{ij} h_j \right)^2 \right] - 0 \\ &= \mathbb{E} \left[\left(\sum_{j=1}^{D_h} \Omega_{ij} h_j \right)^2 \right] \\ &= \sum_{j=1}^{D_h} \mathbb{E} [\Omega_{ij}^2] \mathbb{E} [h_j^2] \end{aligned}$$

For all the cross terms, $E[\Omega_{ij}] = 0$ so only the squared terms are left, then use independence.

Set all the biases to 0

Weights normally distributed
mean 0
variance σ_{Ω}^2

Rule 1:	$\mathbb{E}[k] = k$
Rule 2:	$\mathbb{E}[k \cdot g[x]] = k \cdot \mathbb{E}[g[x]]$
Rule 3:	$\mathbb{E}[f[x] + g[x]] = \mathbb{E}[f[x]] + \mathbb{E}[g[x]]$
Rule 4:	$\mathbb{E}[f[x]g[y]] = \mathbb{E}[f[x]]\mathbb{E}[g[y]]$ if x, y independent

$$\begin{aligned} \sigma_{f'}^2 &= \mathbb{E}[f_i'^2] - \mathbb{E}[f_i']^2 \\ &= \mathbb{E} \left[\left(\beta_i + \sum_{j=1}^{D_h} \Omega_{ij} h_j \right)^2 \right] - 0 \\ &= \mathbb{E} \left[\left(\sum_{j=1}^{D_h} \Omega_{ij} h_j \right)^2 \right] \end{aligned}$$

Set all the biases to 0

Weights normally distributed
mean 0
variance σ_{Ω}^2

$$\begin{aligned} &= \sum_{j=1}^{D_h} \mathbb{E}[\Omega_{ij}^2] \mathbb{E}[h_j^2] \\ &= \sum_{j=1}^{D_h} \sigma_{\Omega}^2 \mathbb{E}[h_j^2] = \sigma_{\Omega}^2 \sum_{j=1}^{D_h} \mathbb{E}[h_j^2] \end{aligned}$$

Because the Ω 's are zero mean, this is the variance.

$$\begin{aligned}
\sigma_{f'}^2 &= \sigma_{\Omega}^2 \sum_{j=1}^{D_h} \mathbb{E} [h_j^2] \\
&= \sigma_{\Omega}^2 \sum_{j=1}^{D_h} \mathbb{E} [\text{ReLU}[f_j]^2] \\
&= \sigma_{\Omega}^2 \sum_{j=1}^{D_h} \int_{-\infty}^{\infty} \text{ReLU}[f_j]^2 Pr(f_j) df_j \quad \leftarrow \text{From the definition of expectation.} \\
&= \sigma_{\Omega}^2 \sum_{j=1}^{D_h} \int_{-\infty}^{\infty} (\mathbb{I}[f_j > 0] f_j)^2 Pr(f_j) df_j \\
&= \sigma_{\Omega}^2 \sum_{j=1}^{D_h} \int_0^{\infty} f_j^2 Pr(f_j) df_j \quad \leftarrow \text{Only positive integral limits because of ReLU} \\
&= \sigma_{\Omega}^2 \sum_{j=1}^{D_h} \frac{\sigma_f^2}{2} = \frac{D_h \sigma_{\Omega}^2 \sigma_f^2}{2} \quad \leftarrow \frac{1}{2} \text{ of the variance for zero mean distribution}
\end{aligned}$$

Aim: keep variance same between two layers

Since:

$$\sigma_{f'}^2 = \frac{D_h \sigma_{\Omega}^2 \sigma_f^2}{2}$$

Should choose:

$$\sigma_{\Omega}^2 = \frac{2}{D_h}$$

To get:

$$\sigma_{f'}^2 = \sigma_f^2$$

This is called **He initialization** or **Kaiming initialization**.

He initialization (assumes ReLU)

- Forward pass: want the variance of hidden unit activations in layer k+1 to be the same as variance of activations in layer k:

$$\sigma_{\Omega}^2 = \frac{2}{D_h} \quad \leftarrow \text{Number of units at layer k}$$

- Backward pass: want the variance of gradients at layer k to be the same as variance of gradient in layer k+1:

$$\sigma_{\Omega}^2 = \frac{2}{D_{h'}} \quad \leftarrow \text{Number of units at layer k+1}$$

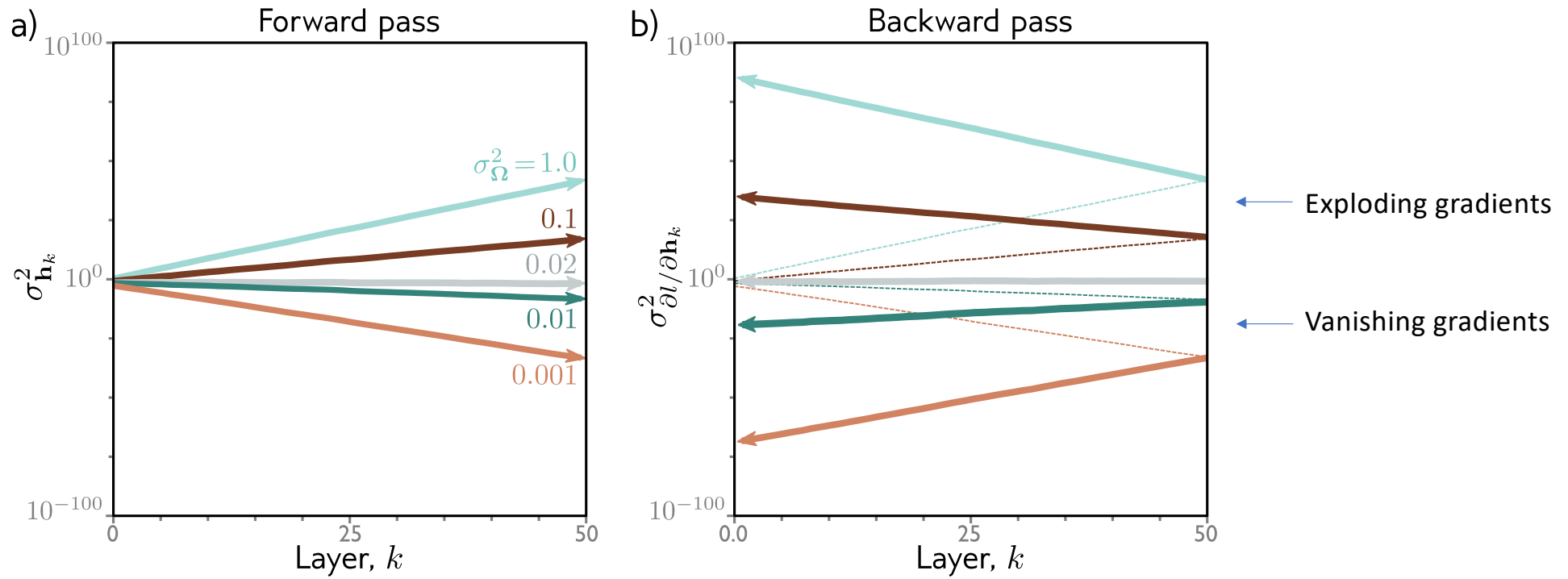


Figure 7.4 Weight initialization. Consider a deep network with 50 hidden layers and $D_h = 100$ hidden units per layer. The network has a 100 dimensional input \mathbf{x} initialized with values from a standard normal distribution, a single output fixed at $y = 0$, and a least squares loss function. The bias vectors β_k are initialized to zero and the weight matrices Ω_k are initialized with a normal distribution with mean zero and five different variances $\sigma_{\Omega}^2 \in \{0.001, 0.01, 0.02, 0.1, 1.0\}$. a)

$$\sigma_{\Omega}^2 = \frac{2}{D_h} = \frac{2}{100} = 0.02$$

Default Initialization in PyTorch

https://pytorch.org/docs/stable/nn.init.html#torch.nn.init.kaiming_uniform

```
torch.nn.init.kaiming_uniform_(tensor, a=0, mode='fan_in', nonlinearity='leaky_relu',  
generator=None) [SOURCE]
```

Fill the input *Tensor* with values using a Kaiming uniform distribution.

The method is described in *Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification* - He, K. et al. (2015). The resulting tensor will have values sampled from $\mathcal{U}(-\text{bound}, \text{bound})$ where

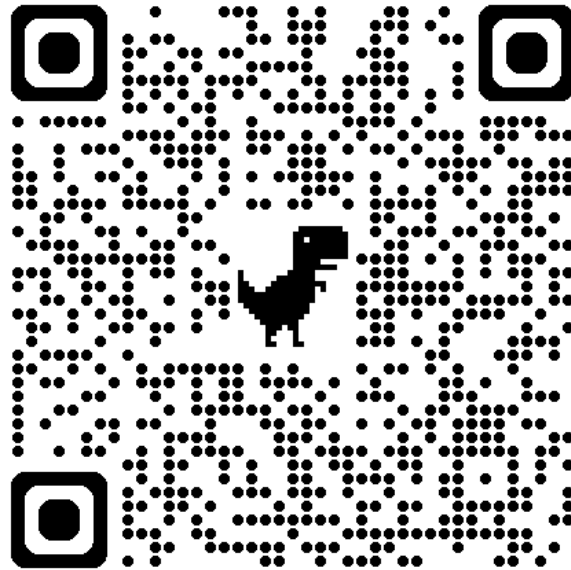
$$\text{bound} = \text{gain} \times \sqrt{\frac{3}{\text{fan_mode}}}$$

Also known as He initialization.

Example Code

- Custom Code with Initialization Examples
- UDL Simple PyTorch Example

Feedback?



[Link](#)