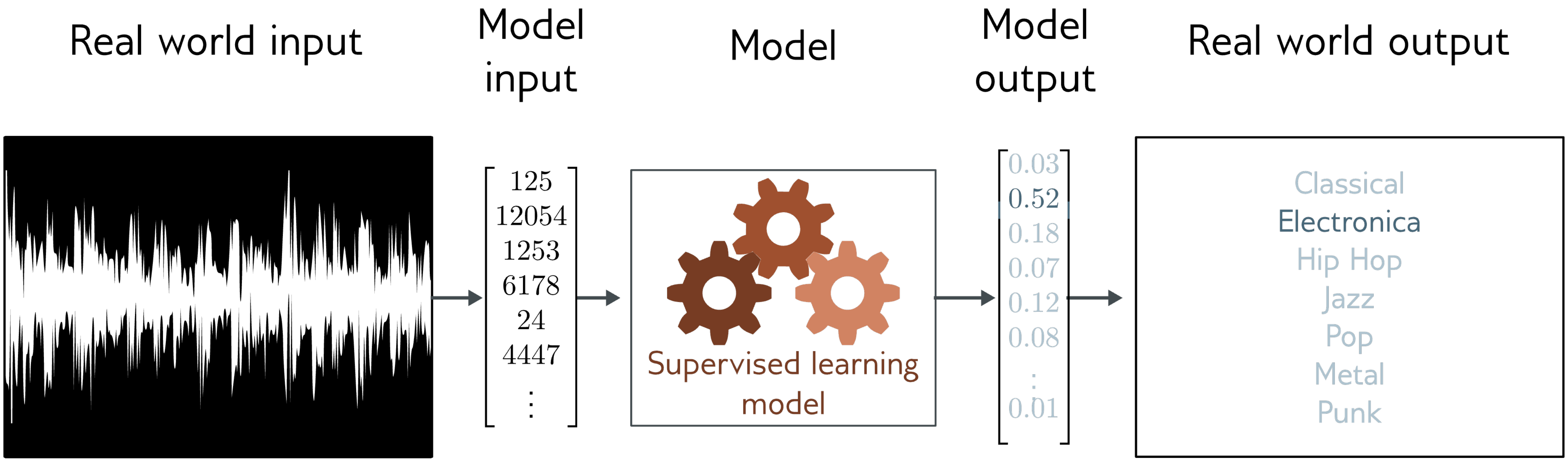
UNIVERSITY OF BATH

CM20315 - Machine Learning

Prof. Simon Prince

7a. Gradients

Music genre classification



- Multiclass classification problem (discrete classes, >2 possible values)
- Convolutional network

Loss function

- Training dataset of I pairs of input/output examples:

$$\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{I}$$

- Loss function or cost function measures how bad model is:

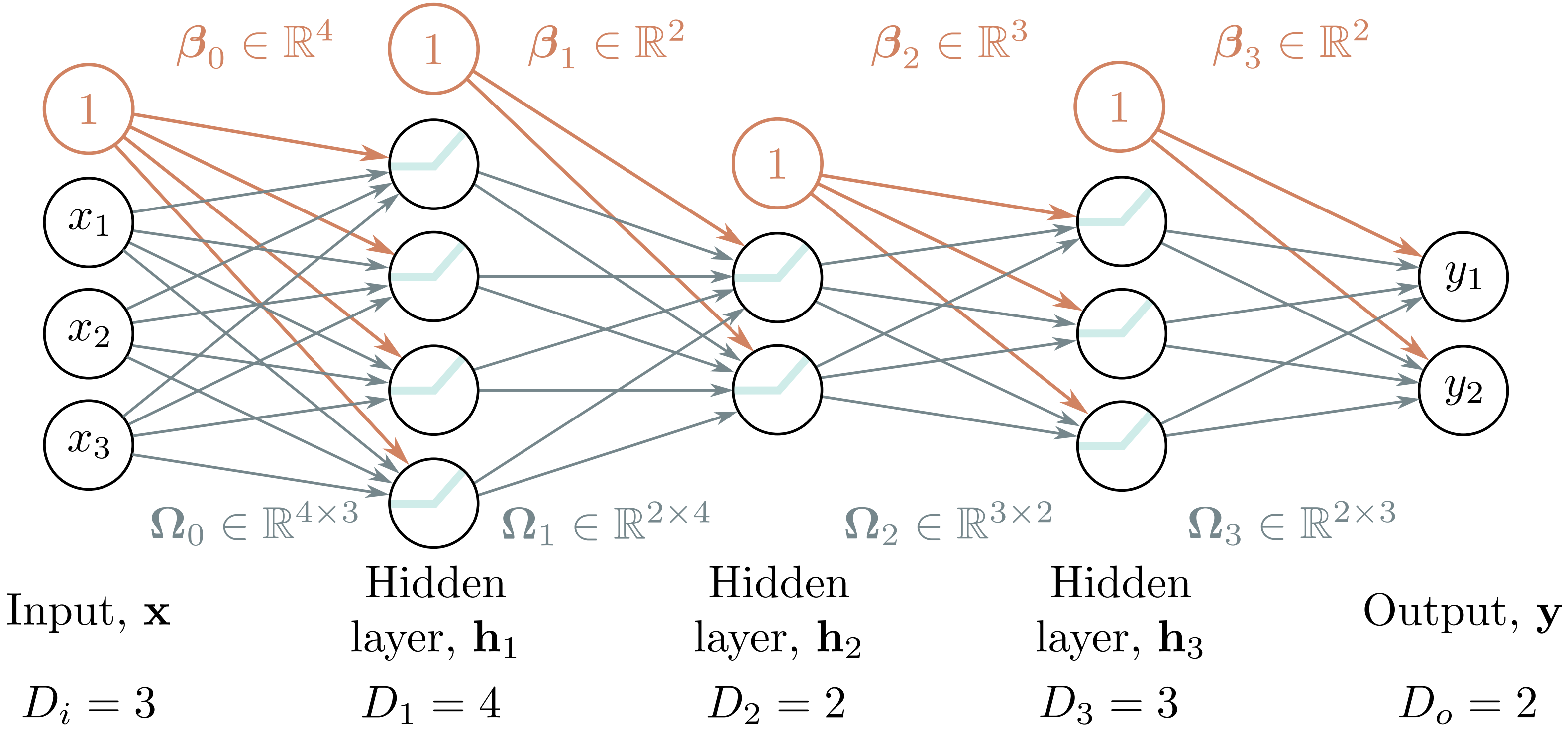
$$L[\phi, \mathrm{f}[\mathbf{x}_i, \phi], \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{I}]$$

or for short:

$$L[\phi]$$

Returns a scalar that is smaller when model maps inputs to outputs better

Example



$$
\begin{aligned}
\mathbf{h}_1 &= \mathbf{a}[\boldsymbol{\beta}_0 + \boldsymbol{\Omega}_0\mathbf{x}] \\
\mathbf{h}_2 &= \mathbf{a}[\boldsymbol{\beta}_1 + \boldsymbol{\Omega}_1\mathbf{h}_1] \\
\mathbf{h}_3 &= \mathbf{a}[\boldsymbol{\beta}_2 + \boldsymbol{\Omega}_2\mathbf{h}_2] \\
\mathbf{f}[\mathbf{x}, \boldsymbol{\phi}] &= \boldsymbol{\beta}_3 + \boldsymbol{\Omega}_3\mathbf{h}_3
\end{aligned}
$$

Problem 1: Computing gradients

Loss: sum of individual terms:

$$L[\boldsymbol{\phi}] = \sum_{i=1}^{I} \ell_i = \sum_{i=1}^{I} \mathrm{l}[\mathrm{f}[\mathbf{x}_i, \boldsymbol{\phi}], y_i]$$

SGD Algorithm:

$$\boldsymbol{\phi}_{t+1} \longleftarrow \boldsymbol{\phi}_t - \alpha \sum_{i \in \mathcal{B}_t} \frac{\partial \ell_i[\boldsymbol{\phi}_t]}{\partial \boldsymbol{\phi}}$$

Parameters:

$$\boldsymbol{\phi} = \{\boldsymbol{\beta}_0, \boldsymbol{\Omega}_0, \boldsymbol{\beta}_1, \boldsymbol{\Omega}_1, \boldsymbol{\beta}_2, \boldsymbol{\Omega}_2, \boldsymbol{\beta}_3, \boldsymbol{\Omega}_3\}$$

Need to compute gradients

$$\frac{\partial \ell_i}{\partial \boldsymbol{\beta}_k} \quad \text{and} \quad \frac{\partial \ell_i}{\partial \boldsymbol{\Omega}_k}$$

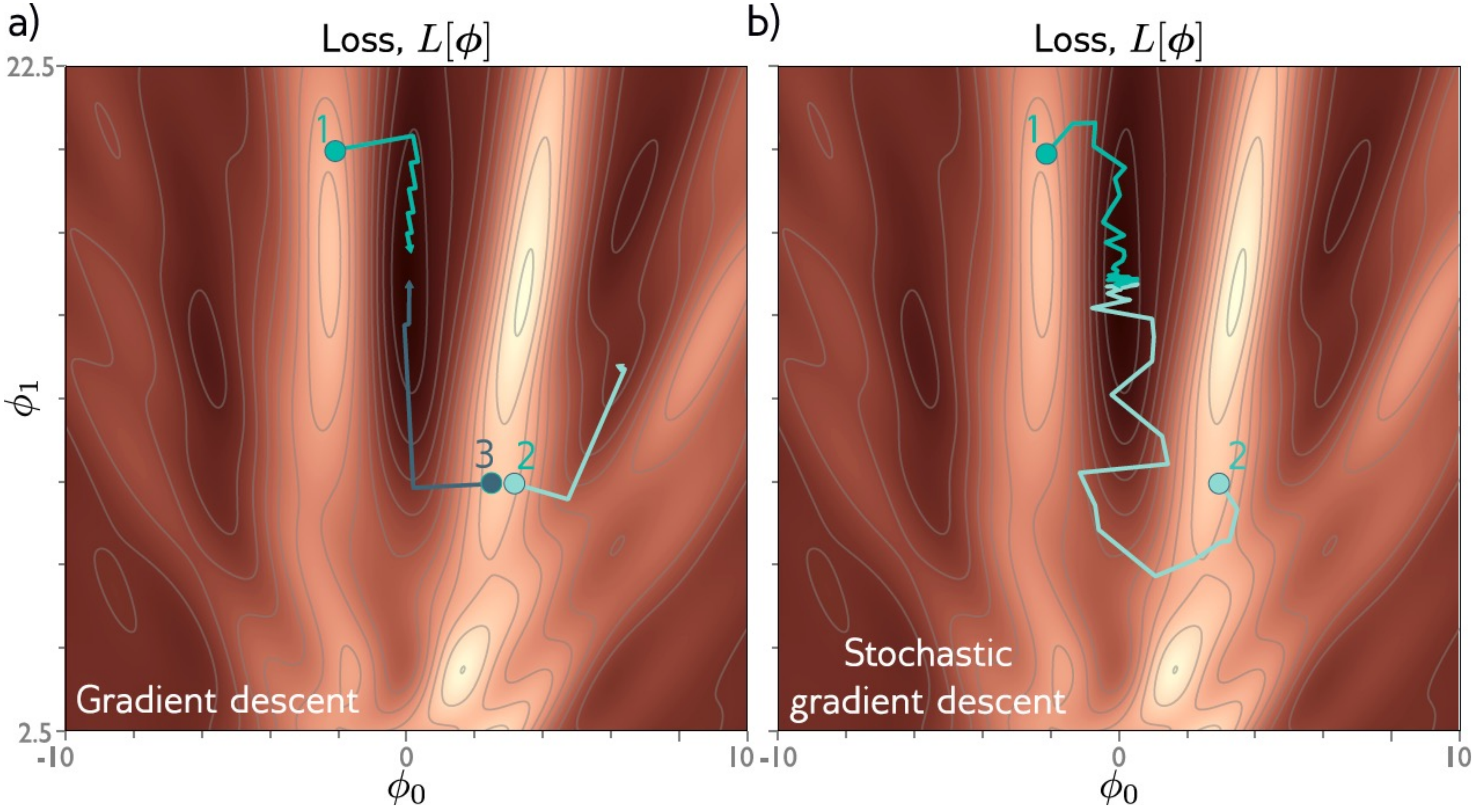
Why is this such a big deal?

- A neural network is just an equation:

$$
\begin{aligned}
y' = \phi'_0 + \phi'_1 \mathrm{a}\left[\psi_{10} + \psi_{11}\mathrm{a}[\theta_{10} + \theta_{11}x] + \psi_{12}\mathrm{a}[\theta_{20} + \theta_{21}x] + \psi_{13}\mathrm{a}[\theta_{30} + \theta_{31}x]\right] \\
+ \phi'_2 \mathrm{a}[\psi_{20} + \psi_{21}\mathrm{a}[\theta_{10} + \theta_{11}x] + \psi_{22}\mathrm{a}[\theta_{20} + \theta_{21}x] + \psi_{23}\mathrm{a}[\theta_{30} + \theta_{31}x]] \\
+ \phi'_3 \mathrm{a}[\psi_{30} + \psi_{31}\mathrm{a}[\theta_{10} + \theta_{11}x] + \psi_{32}\mathrm{a}[\theta_{20} + \theta_{21}x] + \psi_{33}\mathrm{a}[\theta_{30} + \theta_{31}x]]
\end{aligned}
$$

- But it’s a huge equation, and we need to compute derivative
  - for every parameter
  - for every point in the batch
  - for every iteration of SGD

Problem 2: initialization



Where should we start the parameters before we commence SGD?

Gradients

- Backpropagation intuition
- Toy model
- Background mathematics
- Backpropagation forward pass
- Backpropagation backward pass
- Algorithmic differentiation
- Code

Problem 1: Computing gradients

Loss: sum of individual terms:

$$L[\boldsymbol{\phi}] = \sum_{i=1}^{I} \ell_i = \sum_{i=1}^{I} \mathrm{l}[\mathrm{f}[\mathbf{x}_i, \boldsymbol{\phi}], y_i]$$

SGD Algorithm:

$$\boldsymbol{\phi}_{t+1} \longleftarrow \boldsymbol{\phi}_t - \alpha \sum_{i \in \mathcal{B}_t} \frac{\partial \ell_i[\boldsymbol{\phi}_t]}{\partial \boldsymbol{\phi}}$$

Parameters:

$$\boldsymbol{\phi} = \{\boldsymbol{\beta}_0, \boldsymbol{\Omega}_0, \boldsymbol{\beta}_1, \boldsymbol{\Omega}_1, \boldsymbol{\beta}_2, \boldsymbol{\Omega}_2, \boldsymbol{\beta}_3, \boldsymbol{\Omega}_3\}$$

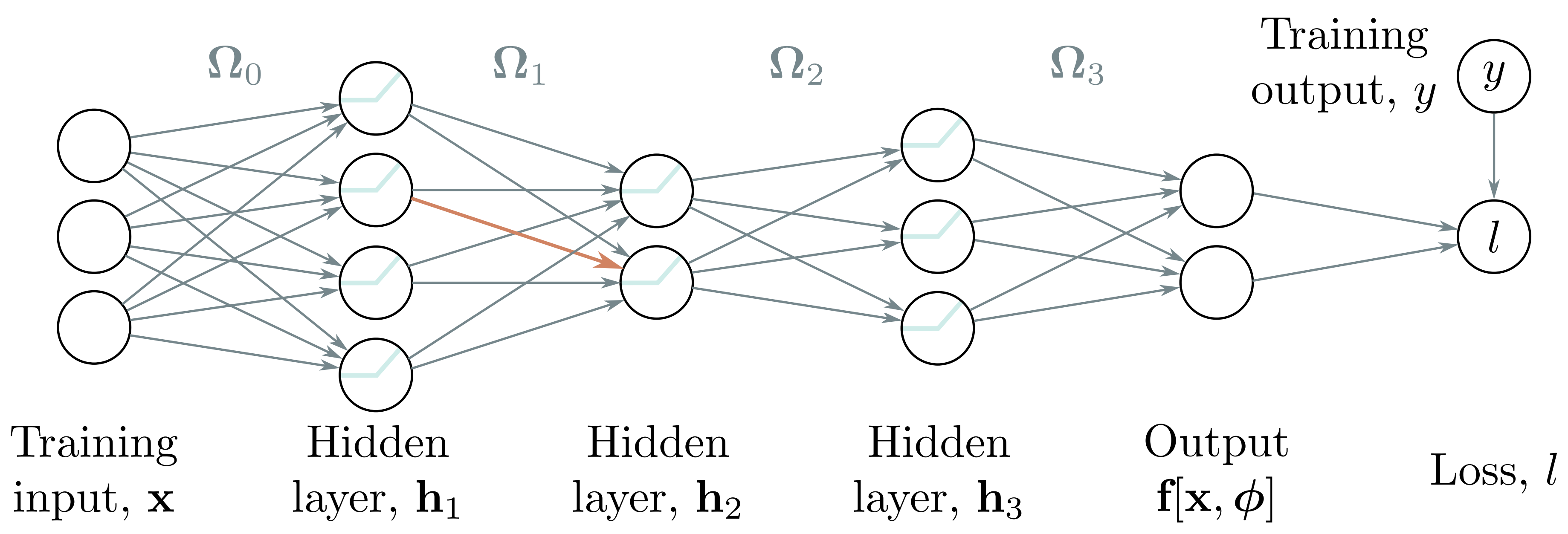
Need to compute gradients

$$\frac{\partial \ell_i}{\partial \boldsymbol{\beta}_k} \quad \text{and} \quad \frac{\partial \ell_i}{\partial \boldsymbol{\Omega}_k}$$

Algorithm to compute gradient efficiently

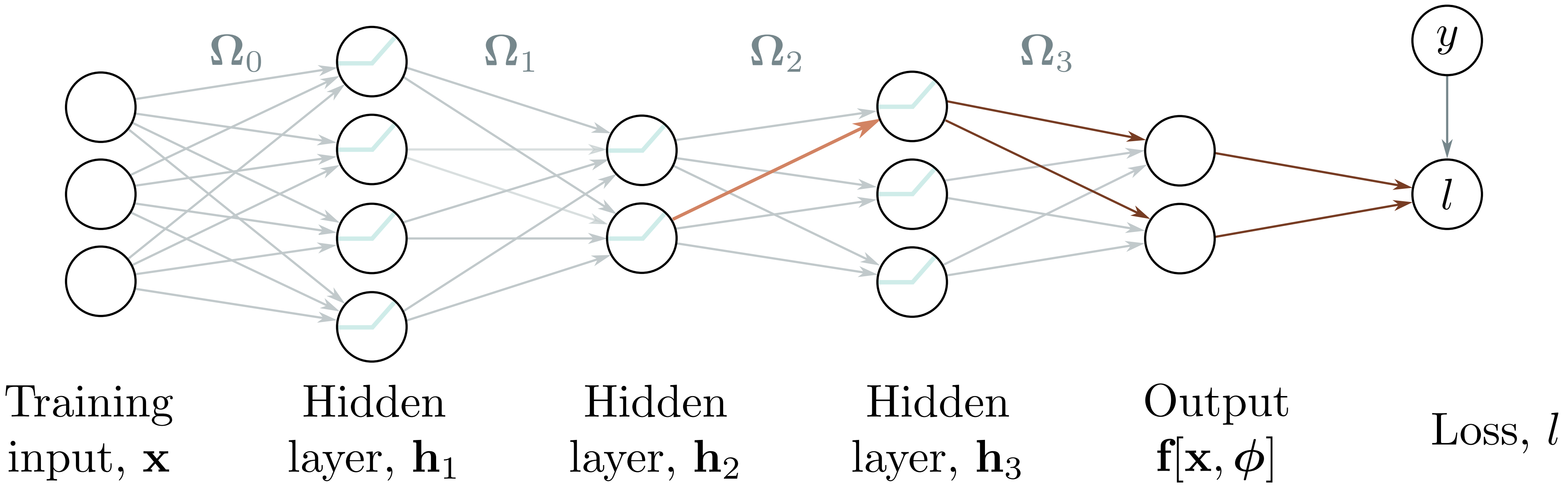
- "Backpropagation algorithm"
- Rumelhart, Hinton, and Williams (1986)

BackProp intuition #1: the forward pass



- Orange weight multiplies activation (ReLU output) in previous layer
- We want to know how change in orange weight affects loss
- If we double activation in previous layer, weight will have twice the effect
- Conclusion: we need to know the activations at each layer.

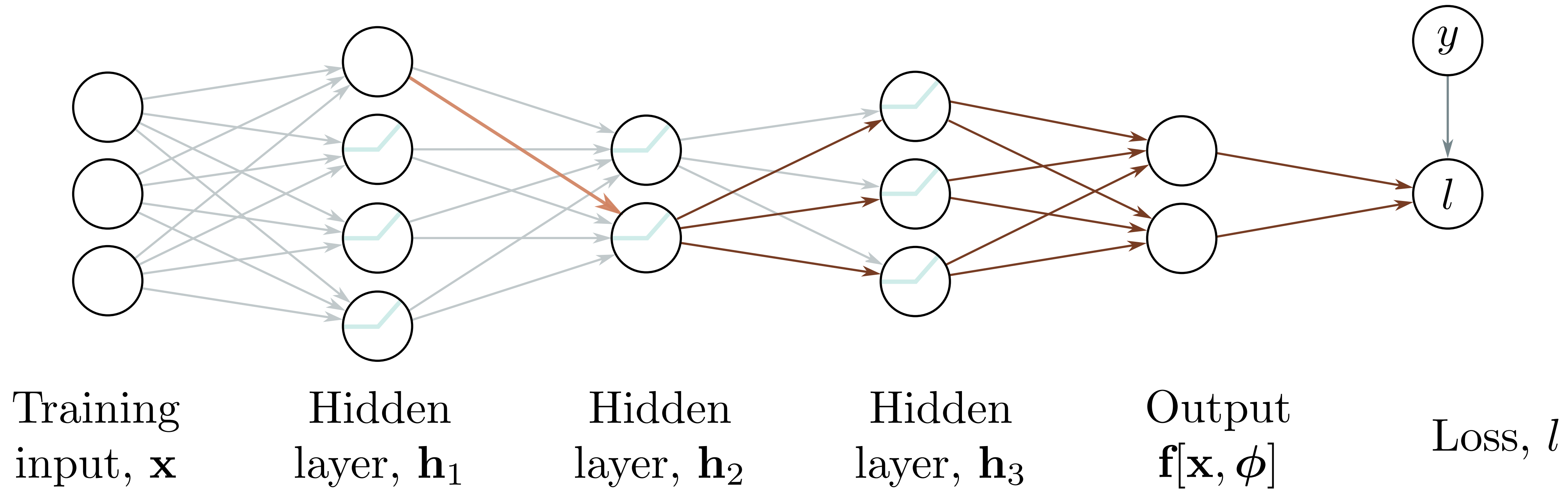
BackProp intuition #2: the backward pass



To calculate how a small change in a weight or bias feeding into hidden layer $\mathbf{h}_3$ modifies the loss, we need to know:

- how a change in layer $\mathbf{h}_3$ changes the model output $\mathbf{f}$
- how a change in model output changes the loss l

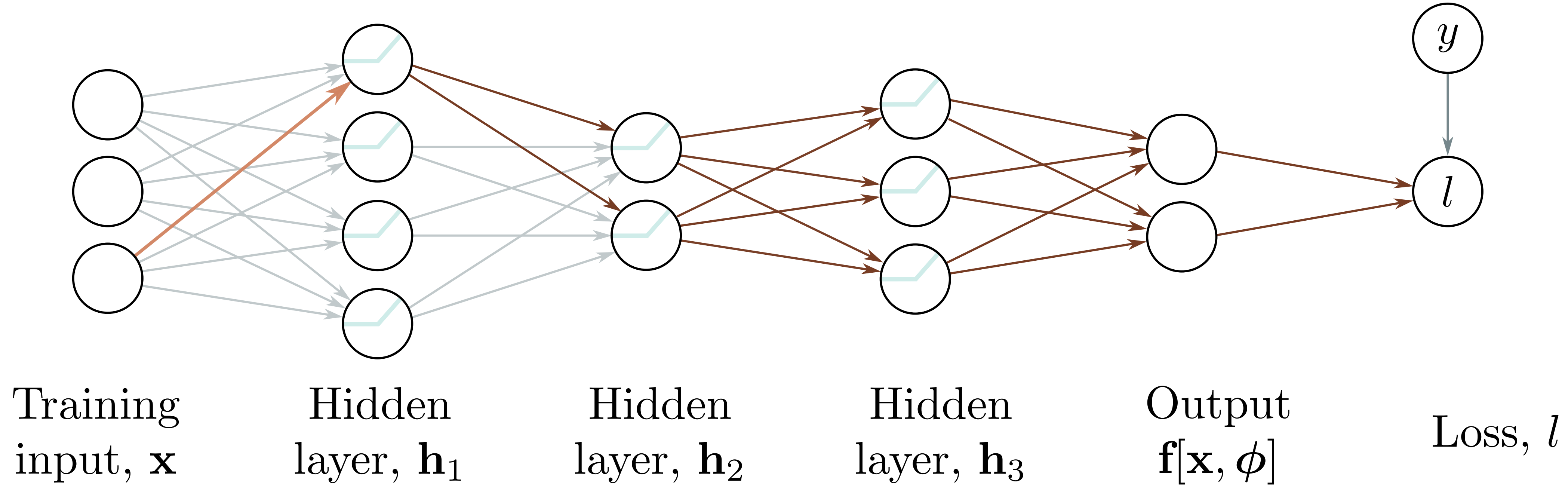
BackProp intuition #2: the backward pass



To calculate how a small change in a weight or bias feeding into hidden layer $\mathbf{h}_2$ modifies the loss, we need to know:

- how a change in layer $\mathbf{h}_2$ affects $\mathbf{h}_3$
- how $\mathbf{h}_3$ changes the model output
- how this output changes the loss

BackProp intuition #2: the backward pass



To calculate how a small change in a weight or bias feeding into hidden layer $\mathbf{h}_1$ modifies the loss, we need to know:

- how a change in layer $\mathbf{h}_1$ affects layer $\mathbf{h}_2$
- how a change in layer $\mathbf{h}_2$ affects layer $\mathbf{h}_3$
- how layer $\mathbf{h}_3$ changes the model output
- how the model output changes the loss

Gradients

- Backpropagation intuition
- Toy model
- Background mathematics
- Backpropagation forward pass
- Backpropagation backward pass
- Algorithmic differentiation
- Code

Toy function

$$\mathrm{f}[x, \boldsymbol{\phi}] = \beta_3 + \omega_3 \cdot \cos\Big[\beta_2 + \omega_2 \cdot \exp\big[\beta_1 + \omega_1 \cdot \sin[\beta_0 + \omega_0 \cdot x]\big]\Big]$$

$$\ell_i = (\mathrm{f}[x_i, \boldsymbol{\phi}] - y_i)^2$$

- Consists of a series of functions that are composed with each other.
- Unlike in neural networks just uses scalars (not vectors)
- “Activation functions” sin, exp, cos

Toy function

$$\mathrm{f}[x,\boldsymbol{\phi}] = \beta_3 + \omega_3 \cdot \cos\Big[\beta_2 + \omega_2 \cdot \exp\big[\beta_1 + \omega_1 \cdot \sin[\beta_0 + \omega_0 \cdot x]\big]\Big]$$

$$\ell_i = (\mathrm{f}[x_i,\boldsymbol{\phi}] - y_i)^2$$

Derivatives

$$\frac{\partial \cos[z]}{\partial z} = -\sin[z] \qquad \frac{\partial \exp[z]}{\partial z} = \exp[z] \qquad \frac{\partial \sin[z]}{\partial z} = \cos[z]$$

Gradients of toy function

$$\mathrm{f}[x, \boldsymbol{\phi}] = \beta_3 + \omega_3 \cdot \cos\Big[\beta_2 + \omega_2 \cdot \exp\big[\beta_1 + \omega_1 \cdot \sin[\beta_0 + \omega_0 \cdot x]\big]\Big]$$

$$\ell_i = (\mathrm{f}[x_i, \boldsymbol{\phi}] - y_i)^2$$

We want to calculate:

How does a small change in β_3 change the loss l_i for the i'th example?

$$\frac{\partial \ell_i}{\partial \beta_0}, \quad \frac{\partial \ell_i}{\partial \omega_0}, \quad \frac{\partial \ell_i}{\partial \beta_1}, \quad \frac{\partial \ell_i}{\partial \omega_1}, \quad \frac{\partial \ell_i}{\partial \beta_2}, \quad \frac{\partial \ell_i}{\partial \omega_2}, \quad \frac{\partial \ell_i}{\partial \beta_3}, \quad \text{and} \quad \frac{\partial \ell_i}{\partial \omega_3}$$

Gradients of composed functions

$$\text{f}[x, \boldsymbol{\phi}] = \beta_3 + \omega_3 \cdot \cos\Big[\beta_2 + \omega_2 \cdot \exp\big[\beta_1 + \omega_1 \cdot \sin[\beta_0 + \omega_0 \cdot x]\big]\Big]$$

$$\ell_i = (\text{f}[x_i, \boldsymbol{\phi}] - y_i)^2$$

Calculating expressions by hand:

- some expressions very complicated.
- obvious redundancy (look at sin terms in bottom equation)

$$\begin{aligned}\frac{\partial \ell_i}{\partial \omega_0} = \quad & -2\left(\beta_3 + \omega_3 \cdot \cos\Big[\beta_2 + \omega_2 \cdot \exp\big[\beta_1 + \omega_1 \cdot \sin[\beta_0 + \omega_0 \cdot x_i]\big]\Big] - y_i\right) \\ & \cdot \omega_1\omega_2\omega_3 \cdot x_i \cdot \cos[\beta_0 + \omega_0 \cdot x_i] \cdot \exp\Big[\beta_1 + \omega_1 \cdot \sin[\beta_0 + \omega_0 \cdot x_i]\Big] \\ & \cdot \sin\Big[\beta_2 + \omega_2 \cdot \exp\Big[\beta_1 + \omega_1 \cdot \sin[\beta_0 + \omega_0 \cdot x_i]\Big]\Big]\end{aligned}$$

Forward pass

$$\text{f}[x, \boldsymbol{\phi}] = \beta_3 + \omega_3 \cdot \cos\Big[\beta_2 + \omega_2 \cdot \exp\big[\beta_1 + \omega_1 \cdot \sin[\beta_0 + \omega_0 \cdot x]\big]\Big]$$

$$\ell_i = (\text{f}[x_i, \boldsymbol{\phi}] - y_i)^2$$

1. Write this as a series of intermediate calculations

2. Compute these intermediate quantities

Forward pass

$$\mathrm{f}[x, \boldsymbol{\phi}] = \beta_3 + \omega_3 \cdot \cos\Big[\beta_2 + \omega_2 \cdot \exp\big[\beta_1 + \omega_1 \cdot \sin[\beta_0 + \omega_0 \cdot x]\big]\Big]$$

$$\ell_i = (\mathrm{f}[x_i, \boldsymbol{\phi}] - y_i)^2$$

1. Write this as a series of intermediate calculations

2. Compute these intermediate quantities

$$f_0 = \beta_0 + \omega_0 \cdot x_i$$
$$h_1 = \sin[f_0]$$
$$f_1 = \beta_1 + \omega_1 \cdot h_1$$
$$h_2 = \exp[f_1]$$

$$f_2 = \beta_2 + \omega_2 \cdot h_2$$
$$h_3 = \cos[f_2]$$
$$f_3 = \beta_3 + \omega_3 \cdot h_3$$
$$\ell_i = (f_3 - y_i)^2.$$

Forward pass

$$\mathrm{f}[x, \boldsymbol{\phi}] = \beta_3 + \omega_3 \cdot \cos\Big[\beta_2 + \omega_2 \cdot \exp\big[\beta_1 + \omega_1 \cdot \sin[\beta_0 + \omega_0 \cdot x]\big]\Big]$$

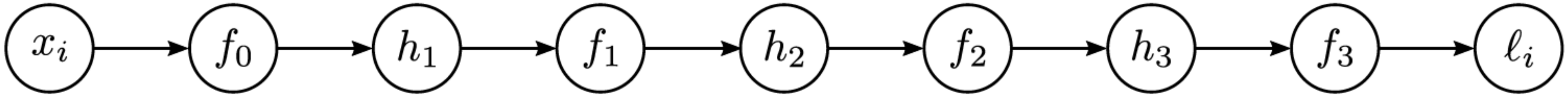
$$\ell_i = (\mathrm{f}[x_i, \boldsymbol{\phi}] - y_i)^2$$

1. Write this as a series of intermediate calculations

2. Compute these intermediate quantities

$$\begin{aligned} f_0 &= \beta_0 + \omega_0 \cdot x_i \\ h_1 &= \sin[f_0] \\ f_1 &= \beta_1 + \omega_1 \cdot h_1 \\ h_2 &= \exp[f_1] \end{aligned}$$

$$\begin{aligned} f_2 &= \beta_2 + \omega_2 \cdot h_2 \\ h_3 &= \cos[f_2] \\ f_3 &= \beta_3 + \omega_3 \cdot h_3 \\ \ell_i &= (f_3 - y_i)^2. \end{aligned}$$



Backward pass

$$\mathrm{f}[x, \boldsymbol{\phi}] = \beta_3 + \omega_3 \cdot \cos\Big[\beta_2 + \omega_2 \cdot \exp\big[\beta_1 + \omega_1 \cdot \sin[\beta_0 + \omega_0 \cdot x]\big]\Big]$$

$$\ell_i = (\mathrm{f}[x_i, \boldsymbol{\phi}] - y_i)^2$$

1. Compute the derivatives of the loss with respect to these intermediate quantities, but in reverse order.

$$\frac{\partial \ell_i}{\partial f_3}, \quad \frac{\partial \ell_i}{\partial h_3}, \quad \frac{\partial \ell_i}{\partial f_2}, \quad \frac{\partial \ell_i}{\partial h_2}, \quad \frac{\partial \ell_i}{\partial f_1}, \quad \frac{\partial \ell_i}{\partial h_1}, \quad \text{and} \quad \frac{\partial \ell_i}{\partial f_0}$$

Backward pass

$$\mathrm{f}[x, \boldsymbol{\phi}] = \beta_3 + \omega_3 \cdot \cos\Big[\beta_2 + \omega_2 \cdot \exp\big[\beta_1 + \omega_1 \cdot \sin[\beta_0 + \omega_0 \cdot x]\big]\Big]$$

$$\ell_i = (\mathrm{f}[x_i, \boldsymbol{\phi}] - y_i)^2$$

1. Compute the derivatives of the loss with respect to these intermediate quantities, but in reverse order.

$$\frac{\partial \ell_i}{\partial f_3}, \quad \frac{\partial \ell_i}{\partial h_3}, \quad \frac{\partial \ell_i}{\partial f_2}, \quad \frac{\partial \ell_i}{\partial h_2}, \quad \frac{\partial \ell_i}{\partial f_1}, \quad \frac{\partial \ell_i}{\partial h_1}, \quad \text{and} \quad \frac{\partial \ell_i}{\partial f_0}$$

Backward pass

1. Compute the derivatives of the loss with respect to these intermediate quantities, but in reverse order.

$$
\begin{aligned}
f_0 &= \beta_0 + \omega_0 \cdot x_i \\
h_1 &= \sin[f_0] \\
f_1 &= \beta_1 + \omega_1 \cdot h_1 \\
h_2 &= \exp[f_1]
\end{aligned}
\qquad
\begin{aligned}
f_2 &= \beta_2 + \omega_2 \cdot h_2 \\
h_3 &= \cos[f_2] \\
f_3 &= \beta_3 + \omega_3 \cdot h_3 \\
\ell_i &= (f_3 - y_i)^2.
\end{aligned}
$$

- The first of these derivatives is trivial

$$\frac{\partial \ell_i}{\partial f_3} = 2(f_3 - y_i)$$

Backward pass

1. Compute the derivatives of the loss with respect to these intermediate quantities, but in reverse order.

$$\begin{aligned} f_0 &= \beta_0 + \omega_0 \cdot x_i \\ h_1 &= \sin[f_0] \\ f_1 &= \beta_1 + \omega_1 \cdot h_1 \\ h_2 &= \exp[f_1] \end{aligned} \qquad \begin{aligned} f_2 &= \beta_2 + \omega_2 \cdot h_2 \\ h_3 &= \cos[f_2] \\ f_3 &= \beta_3 + \omega_3 \cdot h_3 \\ \ell_i &= (f_3 - y_i)^2. \end{aligned}$$

- The second of these derivatives is computed via the chain rule

$$\frac{\partial \ell_i}{\partial h_3} = \frac{\partial f_3}{\partial h_3}\frac{\partial \ell_i}{\partial f_3}$$

How does a small change in h_3 change l_i?

Backward pass

1. Compute the derivatives of the loss with respect to these intermediate quantities, but in reverse order.

$$f_0 = \beta_0 + \omega_0 \cdot x_i$$
$$h_1 = \sin[f_0]$$
$$f_1 = \beta_1 + \omega_1 \cdot h_1$$
$$h_2 = \exp[f_1]$$

$$f_2 = \beta_2 + \omega_2 \cdot h_2$$
$$h_3 = \cos[f_2]$$
$$f_3 = \beta_3 + \omega_3 \cdot h_3$$
$$\ell_i = (f_3 - y_i)^2.$$

- The second derivative is computed via the chain rule

$$\frac{\partial \ell_i}{\partial h_3} = \frac{\partial f_3}{\partial h_3}\frac{\partial \ell_i}{\partial f_3}$$

How does a small change in h_3 change l_i?

How does a small change in h_3 change f_3?

How does a small change in f_3 change l_i?

Backward pass

1. Compute the derivatives of the loss with respect to these intermediate quantities, but in reverse order.

$$f_0 = \beta_0 + \omega_0 \cdot x_i$$
$$h_1 = \sin[f_0]$$
$$f_1 = \beta_1 + \omega_1 \cdot h_1$$
$$h_2 = \exp[f_1]$$

$$f_2 = \beta_2 + \omega_2 \cdot h_2$$
$$h_3 = \cos[f_2]$$
$$f_3 = \beta_3 + \omega_3 \cdot h_3$$
$$\ell_i = (f_3 - y_i)^2.$$

- The second of these derivatives is computed via the chain rule

$$\frac{\partial \ell_i}{\partial h_3} = \frac{\partial f_3}{\partial h_3}\frac{\partial \ell_i}{\partial f_3}$$

How does a small change in h_3 change l_i?

ω_3

Already computed!

Backward pass

1. Compute the derivatives of the loss with respect to these intermediate quantities, but in reverse order.

$$f_0 = \beta_0 + \omega_0 \cdot x_i$$
$$h_1 = \sin[f_0]$$
$$f_1 = \beta_1 + \omega_1 \cdot h_1$$
$$h_2 = \exp[f_1]$$

$$f_2 = \beta_2 + \omega_2 \cdot h_2$$
$$h_3 = \cos[f_2]$$
$$f_3 = \beta_3 + \omega_3 \cdot h_3$$
$$\ell_i = (f_3 - y_i)^2.$$

- The remaining derivatives also calculated by further use of chain rule

$$\frac{\partial \ell_i}{\partial f_2} = \frac{\partial h_3}{\partial f_2}\left(\frac{\partial f_3}{\partial h_3}\frac{\partial \ell_i}{\partial f_3}\right)$$

Backward pass

1. Compute the derivatives of the loss with respect to these intermediate quantities, but in reverse order.

$$
\begin{aligned}
f_0 &= \beta_0 + \omega_0 \cdot x_i \\
h_1 &= \sin[f_0] \\
f_1 &= \beta_1 + \omega_1 \cdot h_1 \\
h_2 &= \exp[f_1]
\end{aligned}
\qquad
\begin{aligned}
f_2 &= \beta_2 + \omega_2 \cdot h_2 \\
h_3 &= \cos[f_2] \\
f_3 &= \beta_3 + \omega_3 \cdot h_3 \\
\ell_i &= (f_3 - y_i)^2.
\end{aligned}
$$

- The remaining derivatives also calculated by further use of chain rule

$$
\frac{\partial \ell_i}{\partial f_2} = \frac{\partial h_3}{\partial f_2}\left(\frac{\partial f_3}{\partial h_3}\frac{\partial \ell_i}{\partial f_3}\right)
$$

$-\sin[f_2]$

Already computed!

Backward pass

1. Compute the derivatives of the loss with respect to these intermediate quantities, but in reverse order.

$$f_0 = \beta_0 + \omega_0 \cdot x_i$$
$$h_1 = \sin[f_0]$$
$$f_1 = \beta_1 + \omega_1 \cdot h_1$$
$$h_2 = \exp[f_1]$$

$$f_2 = \beta_2 + \omega_2 \cdot h_2$$
$$h_3 = \cos[f_2]$$
$$f_3 = \beta_3 + \omega_3 \cdot h_3$$
$$\ell_i = (f_3 - y_i)^2.$$

- The remaining derivatives also calculated by further use of chain rule

$$\frac{\partial \ell_i}{\partial f_2} = \frac{\partial h_3}{\partial f_2}\left(\frac{\partial f_3}{\partial h_3}\frac{\partial \ell_i}{\partial f_3}\right)$$
$$\frac{\partial \ell_i}{\partial h_2} = \frac{\partial f_2}{\partial h_2}\left(\frac{\partial h_3}{\partial f_2}\frac{\partial f_3}{\partial h_3}\frac{\partial \ell_i}{\partial f_3}\right)$$

Backward pass

1. Compute the derivatives of the loss with respect to these intermediate quantities, but in reverse order.

$$f_0 = \beta_0 + \omega_0 \cdot x_i$$
$$h_1 = \sin[f_0]$$
$$f_1 = \beta_1 + \omega_1 \cdot h_1$$
$$h_2 = \exp[f_1]$$
$$f_2 = \beta_2 + \omega_2 \cdot h_2$$
$$h_3 = \cos[f_2]$$
$$f_3 = \beta_3 + \omega_3 \cdot h_3$$
$$\ell_i = (f_3 - y_i)^2.$$

- The remaining derivatives also calculated by further use of chain rule

$$\frac{\partial \ell_i}{\partial f_2} = \frac{\partial h_3}{\partial f_2}\left(\frac{\partial f_3}{\partial h_3}\frac{\partial \ell_i}{\partial f_3}\right)$$
$$\frac{\partial \ell_i}{\partial h_2} = \frac{\partial f_2}{\partial h_2}\left(\frac{\partial h_3}{\partial f_2}\frac{\partial f_3}{\partial h_3}\frac{\partial \ell_i}{\partial f_3}\right)$$
$$\frac{\partial \ell_i}{\partial f_1} = \frac{\partial h_2}{\partial f_1}\left(\frac{\partial f_2}{\partial h_2}\frac{\partial h_3}{\partial f_2}\frac{\partial f_3}{\partial h_3}\frac{\partial \ell_i}{\partial f_3}\right)$$
$$\frac{\partial \ell_i}{\partial h_1} = \frac{\partial f_1}{\partial h_1}\left(\frac{\partial h_2}{\partial f_1}\frac{\partial f_2}{\partial h_2}\frac{\partial h_3}{\partial f_2}\frac{\partial f_3}{\partial h_3}\frac{\partial \ell_i}{\partial f_3}\right)$$
$$\frac{\partial \ell_i}{\partial f_0} = \frac{\partial h_1}{\partial f_0}\left(\frac{\partial f_1}{\partial h_1}\frac{\partial h_2}{\partial f_1}\frac{\partial f_2}{\partial h_2}\frac{\partial h_3}{\partial f_2}\frac{\partial f_3}{\partial h_3}\frac{\partial \ell_i}{\partial f_3}\right)$$

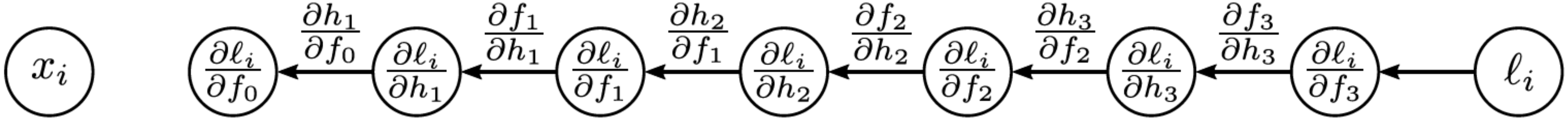
Backward pass

1. Compute the derivatives of the loss with respect to these intermediate quantities, but in reverse order.

$$f_0 = \beta_0 + \omega_0 \cdot x_i$$
$$h_1 = \sin[f_0]$$
$$f_1 = \beta_1 + \omega_1 \cdot h_1$$
$$h_2 = \exp[f_1]$$

$$f_2 = \beta_2 + \omega_2 \cdot h_2$$
$$h_3 = \cos[f_2]$$
$$f_3 = \beta_3 + \omega_3 \cdot h_3$$
$$\ell_i = (f_3 - y_i)^2.$$

- The remaining derivatives also calculated by further use of chain rule



Backward pass

2. Find how the loss changes as a function of the parameters β and ω.

$$
\begin{aligned}
f_0 &= \beta_0 + \omega_0 \cdot x_i \\
h_1 &= \sin[f_0] \\
f_1 &= \beta_1 + \omega_1 \cdot h_1 \\
h_2 &= \exp[f_1]
\end{aligned}
\qquad
\begin{aligned}
f_2 &= \beta_2 + \omega_2 \cdot h_2 \\
h_3 &= \cos[f_2] \\
f_3 &= \beta_3 + \omega_3 \cdot h_3 \\
\ell_i &= (f_3 - y_i)^2.
\end{aligned}
$$

- Another application of the chain rule

$$
\frac{\partial \ell_i}{\partial \omega_k} = \frac{\partial f_k}{\partial \omega_k}\frac{\partial \ell_i}{\partial f_k}
$$

How does a small change in ω_k change l_i?

How does a small change in ω_k change f_k?

How does a small change in f_k change l_i?

Backward pass

2. Find how the loss changes as a function of the parameters β and ω.

$$f_0 = \beta_0 + \omega_0 \cdot x_i$$
$$h_1 = \sin[f_0]$$
$$f_1 = \beta_1 + \omega_1 \cdot h_1$$
$$h_2 = \exp[f_1]$$
$$f_2 = \beta_2 + \omega_2 \cdot h_2$$
$$h_3 = \cos[f_2]$$
$$f_3 = \beta_3 + \omega_3 \cdot h_3$$
$$\ell_i = (f_3 - y_i)^2.$$

- Another application of the chain rule

$$\frac{\partial \ell_i}{\partial \omega_k} = \frac{\partial f_k}{\partial \omega_k}\frac{\partial \ell_i}{\partial f_k}$$

How does a small change in ω_k change l_i?

h_k

Already calculated in part 1.

Backward pass

2. Find how the loss changes as a function of the parameters β and ω.

$$f_0 = \beta_0 + \omega_0 \cdot x_i$$
$$h_1 = \sin[f_0]$$
$$f_1 = \beta_1 + \omega_1 \cdot h_1$$
$$h_2 = \exp[f_1]$$
$$f_2 = \beta_2 + \omega_2 \cdot h_2$$
$$h_3 = \cos[f_2]$$
$$f_3 = \beta_3 + \omega_3 \cdot h_3$$
$$\ell_i = (f_3 - y_i)^2.$$

- Another application of the chain rule
- Similarly for β parameters

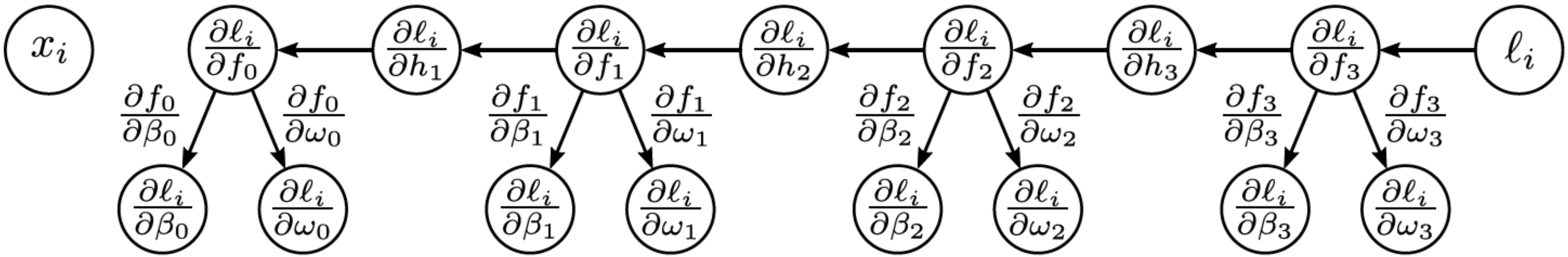
$$\frac{\partial \ell_i}{\partial \omega_k} = \frac{\partial f_k}{\partial \omega_k}\frac{\partial \ell_i}{\partial f_k}$$

$$\frac{\partial \ell_i}{\partial \beta_k} = \frac{\partial f_k}{\partial \beta_k}\frac{\partial \ell_i}{\partial f_k}$$

Backward pass

2. Find how the loss changes as a function of the parameters β and ω.

$$f_0 = \beta_0 + \omega_0 \cdot x_i$$
$$h_1 = \sin[f_0]$$
$$f_1 = \beta_1 + \omega_1 \cdot h_1$$
$$h_2 = \exp[f_1]$$
$$f_2 = \beta_2 + \omega_2 \cdot h_2$$
$$h_3 = \cos[f_2]$$
$$f_3 = \beta_3 + \omega_3 \cdot h_3$$
$$\ell_i = (f_3 - y_i)^2.$$



Gradients

- Backpropagation intuition
- Toy model
- Background mathematics
- Backpropagation forward pass
- Backpropagation backward pass
- Algorithmic differentiation
- Code

Matrix calculus

Scalar function f[] of a vector **a**

$$\mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix}$$

$$\frac{\partial f}{\partial \mathbf{a}} = \begin{bmatrix} \frac{\partial f}{\partial a_1} \\ \frac{\partial f}{\partial a_2} \\ \frac{\partial f}{\partial a_3} \\ \frac{\partial f}{\partial a_4} \end{bmatrix}$$

Matrix calculus

Scalar function f[] of a matrix **A**

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \\ a_{41} & a_{42} & a_{43} \end{bmatrix}$$

$$\frac{\partial f}{\partial \mathbf{A}} = \begin{bmatrix} \frac{\partial f}{\partial a_{11}} & \frac{\partial f}{\partial a_{12}} & \frac{\partial f}{\partial a_{13}} \\ \frac{\partial f}{\partial a_{21}} & \frac{\partial f}{\partial a_{22}} & \frac{\partial f}{\partial a_{23}} \\ \frac{\partial f}{\partial a_{31}} & \frac{\partial f}{\partial a_{32}} & \frac{\partial f}{\partial a_{33}} \\ \frac{\partial f}{\partial a_{41}} & \frac{\partial f}{\partial a_{42}} & \frac{\partial f}{\partial a_{43}} \end{bmatrix}$$

Matrix calculus

Vector function **f[]** of vector **a**

$$\mathbf{f} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} \quad \mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix}$$

$$\frac{\partial \mathbf{f}}{\partial \mathbf{a}} = \begin{bmatrix} \frac{\partial f_1}{\partial a_1} & \frac{\partial f_2}{\partial a_1} & \frac{\partial f_3}{\partial a_1} \\ \frac{\partial f_1}{\partial a_2} & \frac{\partial f_2}{\partial a_2} & \frac{\partial f_3}{\partial a_2} \\ \frac{\partial f_1}{\partial a_3} & \frac{\partial f_2}{\partial a_3} & \frac{\partial f_3}{\partial a_3} \\ \frac{\partial f_1}{\partial a_4} & \frac{\partial f_2}{\partial a_4} & \frac{\partial f_4}{\partial a_4} \end{bmatrix}$$

Comparing vector and matrix

Scalar derivatives:

$$f_3 = \beta_3 + \omega_3 h_3$$

$$\frac{\partial f_3}{\partial h_3} = \frac{\partial}{\partial h_3}(\beta_3 + \omega_3 h_3) = \omega_3$$

Comparing vector and matrix

Scalar derivatives:

$$f_3 = \beta_3 + \omega_3 h_3 \qquad \frac{\partial f_3}{\partial h_3} = \frac{\partial}{\partial h_3}(\beta_3 + \omega_3 h_3) = \omega_3$$

Matrix derivatives:

$$\mathbf{f}_3 = \boldsymbol{\beta}_3 + \boldsymbol{\Omega}_3 \mathbf{h}_3 \qquad \frac{\partial \mathbf{f}_3}{\partial \mathbf{h}_3} = \frac{\partial}{\partial \mathbf{h}_3}(\boldsymbol{\beta}_3 + \boldsymbol{\Omega}_3 \mathbf{h}_3) = \boldsymbol{\Omega}_3^T$$

Comparing vector and matrix

Scalar derivatives:

$$f_3 = \beta_3 + \omega_3 h_3$$

$$\frac{\partial f_3}{\partial \beta_3} = \frac{\partial}{\partial \omega_3} \beta_3 + \omega_3 h_3 = 1$$

Matrix derivatives:

$$\mathbf{f}_3 = \boldsymbol{\beta}_3 + \boldsymbol{\Omega}_3 \mathbf{h}_3$$

$$\frac{\partial \mathbf{f}_3}{\partial \boldsymbol{\beta}_3} = \frac{\partial}{\partial \beta_3} (\boldsymbol{\beta}_3 + \boldsymbol{\Omega}_3 \mathbf{h}_3) = \mathbf{I}$$

Homework: (keeners only)

- Consider function: $$\mathbf{f} = \mathbf{B}\mathbf{a}$$

$$\mathbf{f} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} \quad \mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix}$$

- Can write as: $$f_i = \sum_j B_{ij} a_j$$

- Now calculate:

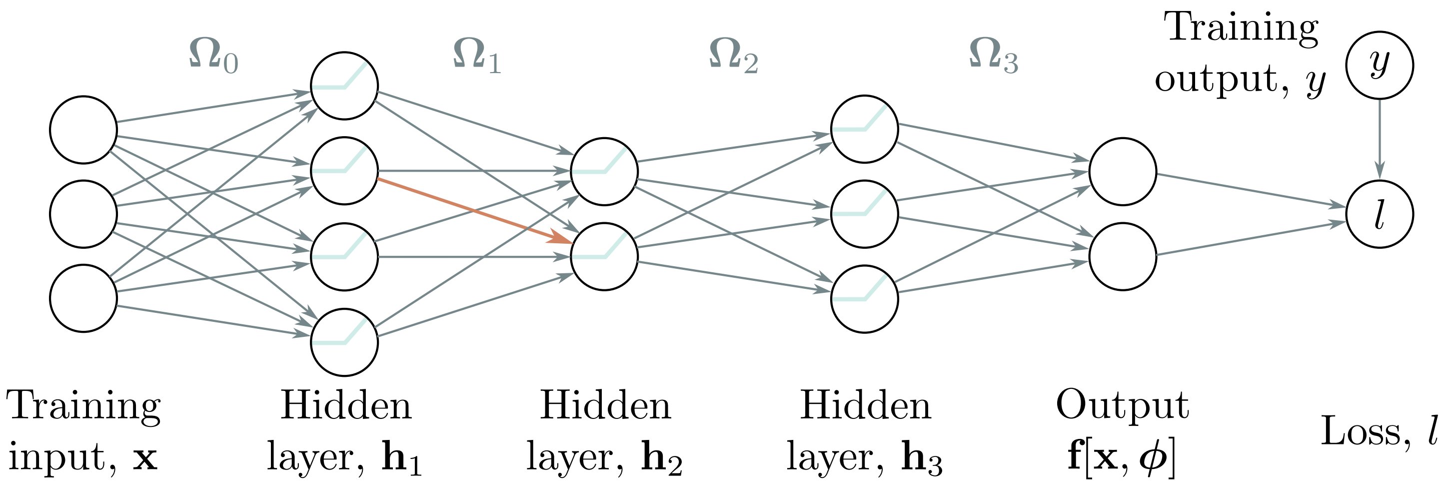
$$\frac{\partial \mathbf{f}}{\partial \mathbf{a}} = \begin{bmatrix} \frac{\partial f_1}{\partial a_1} & \frac{\partial f_2}{\partial a_1} & \frac{\partial f_3}{\partial a_1} \\ \frac{\partial f_1}{\partial a_2} & \frac{\partial f_2}{\partial a_2} & \frac{\partial f_3}{\partial a_2} \\ \frac{\partial f_1}{\partial a_3} & \frac{\partial f_2}{\partial a_3} & \frac{\partial f_3}{\partial a_3} \\ \frac{\partial f_1}{\partial a_4} & \frac{\partial f_2}{\partial a_4} & \frac{\partial f_3}{\partial a_4} \end{bmatrix}$$

- Write final expression as a matrix

Gradients

- Backpropagation intuition
- Toy model
- Background mathematics
- Backpropagation forward pass
- Backpropagation backward pass
- Algorithmic differentiation
- Code

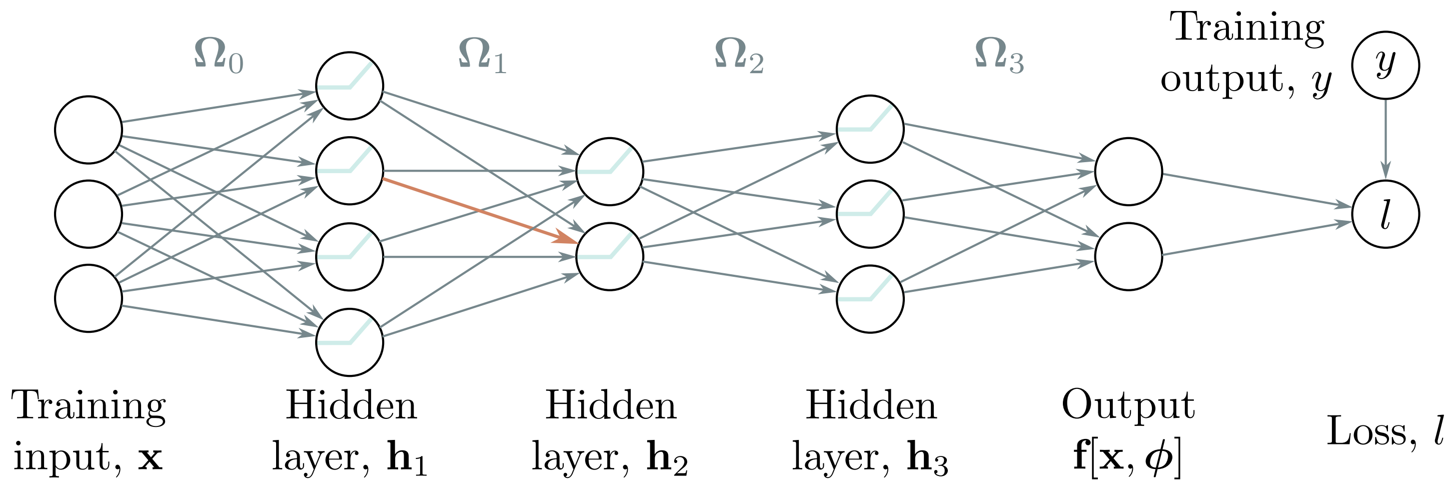
The forward pass



1. Write this as a series of intermediate calculations

$$
\begin{aligned}
\mathbf{f}_0 &= \boldsymbol{\beta}_0 + \boldsymbol{\Omega}_0 \mathbf{x}_i \\
\mathbf{h}_1 &= \mathbf{a}[\mathbf{f}_0] \\
\mathbf{f}_1 &= \boldsymbol{\beta}_1 + \boldsymbol{\Omega}_1 \mathbf{h}_1 \\
\mathbf{h}_2 &= \mathbf{a}[\mathbf{f}_1] \\
\mathbf{f}_2 &= \boldsymbol{\beta}_2 + \boldsymbol{\Omega}_2 \mathbf{h}_2 \\
\mathbf{h}_3 &= \mathbf{a}[\mathbf{f}_2] \\
\mathbf{f}_3 &= \boldsymbol{\beta}_3 + \boldsymbol{\Omega}_3 \mathbf{h}_3 \\
\ell_i &= \mathrm{l}[\mathbf{f}_3, y_i]
\end{aligned}
$$

The forward pass

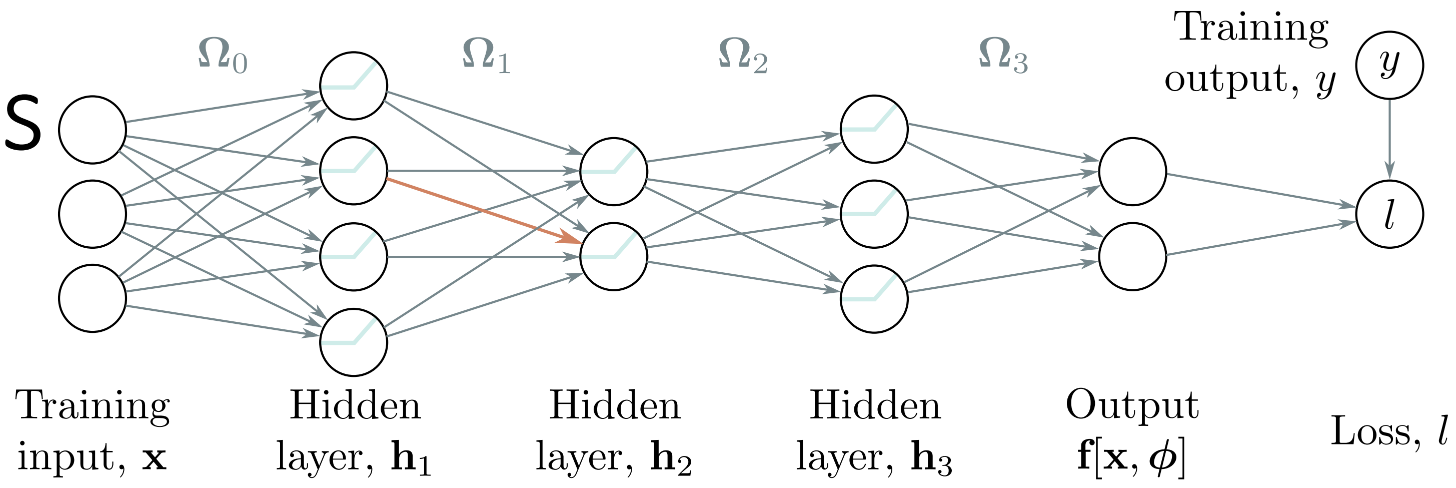


1. Write this as a series of intermediate calculations

2. Compute these intermediate quantities

$$
\begin{aligned}
\mathbf{f}_0 &= \boldsymbol{\beta}_0 + \boldsymbol{\Omega}_0 \mathbf{x}_i \\
\mathbf{h}_1 &= \mathbf{a}[\mathbf{f}_0] \\
\mathbf{f}_1 &= \boldsymbol{\beta}_1 + \boldsymbol{\Omega}_1 \mathbf{h}_1 \\
\mathbf{h}_2 &= \mathbf{a}[\mathbf{f}_1] \\
\mathbf{f}_2 &= \boldsymbol{\beta}_2 + \boldsymbol{\Omega}_2 \mathbf{h}_2 \\
\mathbf{h}_3 &= \mathbf{a}[\mathbf{f}_2] \\
\mathbf{f}_3 &= \boldsymbol{\beta}_3 + \boldsymbol{\Omega}_3 \mathbf{h}_3 \\
\ell_i &= \mathrm{l}[\mathbf{f}_3, y_i]
\end{aligned}
$$

The backward pass



1. Write this as a series of intermediate calculations

2. Compute these intermediate quantities

3. Take derivatives of output with respect to intermediate quantities

$$
\begin{aligned}
\mathbf{f}_0 &= \boldsymbol{\beta}_0 + \boldsymbol{\Omega}_0 \mathbf{x}_i \\
\mathbf{h}_1 &= \mathbf{a}[\mathbf{f}_0] \\
\mathbf{f}_1 &= \boldsymbol{\beta}_1 + \boldsymbol{\Omega}_1 \mathbf{h}_1 \\
\mathbf{h}_2 &= \mathbf{a}[\mathbf{f}_1] \\
\mathbf{f}_2 &= \boldsymbol{\beta}_2 + \boldsymbol{\Omega}_2 \mathbf{h}_2 \\
\mathbf{h}_3 &= \mathbf{a}[\mathbf{f}_2] \\
\mathbf{f}_3 &= \boldsymbol{\beta}_3 + \boldsymbol{\Omega}_3 \mathbf{h}_3 \\
\ell_i &= \mathrm{l}[\mathbf{f}_3, y_i]
\end{aligned}
$$

$$\frac{\partial \ell_i}{\partial \mathbf{f}_3}$$

$$\frac{\partial \ell_i}{\partial \mathbf{f}_2} = \frac{\partial \mathbf{h}_3}{\partial \mathbf{f}_2} \frac{\partial \mathbf{f}_3}{\partial \mathbf{h}_3} \frac{\partial \ell_i}{\partial \mathbf{f}_3}$$

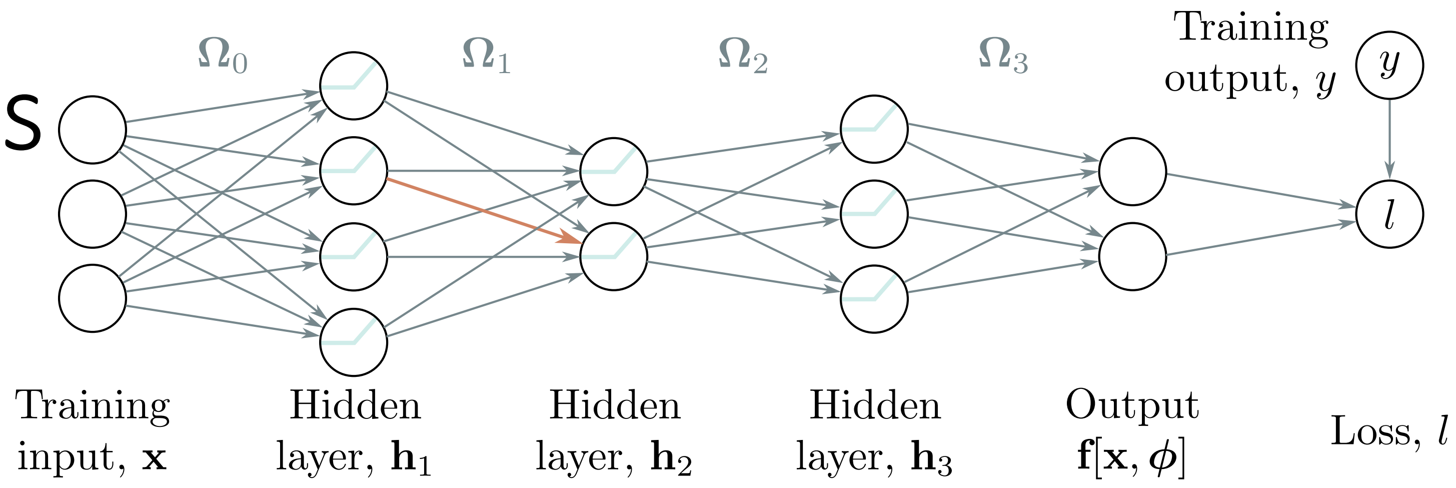
$$\frac{\partial \ell_i}{\partial \mathbf{f}_1} = \frac{\partial \mathbf{h}_2}{\partial \mathbf{f}_1} \frac{\partial \mathbf{f}_2}{\partial \mathbf{h}_2} \left(\frac{\partial \mathbf{h}_3}{\partial \mathbf{f}_2} \frac{\partial \mathbf{f}_3}{\partial \mathbf{h}_3} \frac{\partial \ell_i}{\partial \mathbf{f}_3} \right)$$

$$\frac{\partial \ell_i}{\partial \mathbf{f}_0} = \frac{\partial \mathbf{h}_1}{\partial \mathbf{f}_0} \frac{\partial \mathbf{f}_1}{\partial \mathbf{h}_1} \left(\frac{\partial \mathbf{h}_2}{\partial \mathbf{f}_1} \frac{\partial \mathbf{f}_2}{\partial \mathbf{h}_2} \frac{\partial \mathbf{h}_3}{\partial \mathbf{f}_2} \frac{\partial \mathbf{f}_3}{\partial \mathbf{h}_3} \frac{\partial \ell_i}{\partial \mathbf{f}_3} \right)$$

Gradients

- Backpropagation intuition
- Toy model
- Background mathematics
- Backpropagation forward pass
- Backpropagation backward pass
- Algorithmic differentiation
- Code

The backward pass



1. Write this as a series of intermediate calculations

2. Compute these intermediate quantities

3. Take derivatives of output with respect to intermediate quantities

$$\begin{aligned}
\mathbf{f}_0 &= \boldsymbol{\beta}_0 + \boldsymbol{\Omega}_0 \mathbf{x}_i \\
\mathbf{h}_1 &= \mathbf{a}[\mathbf{f}_0] \\
\mathbf{f}_1 &= \boldsymbol{\beta}_1 + \boldsymbol{\Omega}_1 \mathbf{h}_1 \\
\mathbf{h}_2 &= \mathbf{a}[\mathbf{f}_1] \\
\mathbf{f}_2 &= \boldsymbol{\beta}_2 + \boldsymbol{\Omega}_2 \mathbf{h}_2 \\
\mathbf{h}_3 &= \mathbf{a}[\mathbf{f}_2] \\
\mathbf{f}_3 &= \boldsymbol{\beta}_3 + \boldsymbol{\Omega}_3 \mathbf{h}_3 \\
\ell_i &= \mathrm{l}[\mathbf{f}_3, y_i]
\end{aligned}$$

$$\frac{\partial \ell_i}{\partial \mathbf{f}_3}$$

$$\frac{\partial \ell_i}{\partial \mathbf{f}_2} = \frac{\partial \mathbf{h}_3}{\partial \mathbf{f}_2} \boxed{\frac{\partial \mathbf{f}_3}{\partial \mathbf{h}_3}} \frac{\partial \ell_i}{\partial \mathbf{f}_3}$$

$$\frac{\partial \ell_i}{\partial \mathbf{f}_1} = \frac{\partial \mathbf{h}_2}{\partial \mathbf{f}_1} \frac{\partial \mathbf{f}_2}{\partial \mathbf{h}_2} \left(\frac{\partial \mathbf{h}_3}{\partial \mathbf{f}_2} \frac{\partial \mathbf{f}_3}{\partial \mathbf{h}_3} \frac{\partial \ell_i}{\partial \mathbf{f}_3} \right)$$

$$\frac{\partial \ell_i}{\partial \mathbf{f}_0} = \frac{\partial \mathbf{h}_1}{\partial \mathbf{f}_0} \frac{\partial \mathbf{f}_1}{\partial \mathbf{h}_1} \left(\frac{\partial \mathbf{h}_2}{\partial \mathbf{f}_1} \frac{\partial \mathbf{f}_2}{\partial \mathbf{h}_2} \frac{\partial \mathbf{h}_3}{\partial \mathbf{f}_2} \frac{\partial \mathbf{f}_3}{\partial \mathbf{h}_3} \frac{\partial \ell_i}{\partial \mathbf{f}_3} \right)$$

Yikes!

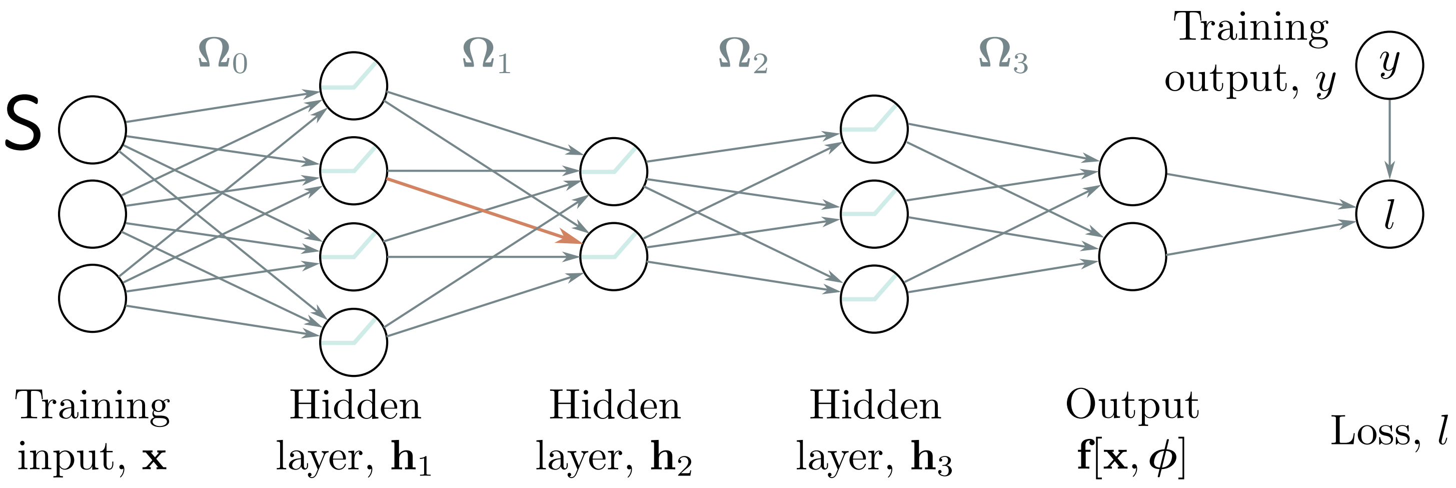
- But:

$$\frac{\partial \mathbf{f}_3}{\partial \mathbf{h}_3} = \frac{\partial}{\partial \mathbf{h}_3}\left(\boldsymbol{\beta}_3 + \boldsymbol{\Omega}_3 \mathbf{h}_3\right) = \boldsymbol{\Omega}_3^T$$

- Quite similar to:

$$\frac{\partial f_3}{\partial h_3} = \frac{\partial}{\partial h_3}\left(\beta_3 + \omega_3 h_3\right) = \omega_3$$

The backward pass



1. Write this as a series of intermediate calculations

2. Compute these intermediate quantities

3. Take derivatives of output with respect to intermediate quantities

$$
\begin{aligned}
\mathbf{f}_0 &= \boldsymbol{\beta}_0 + \boldsymbol{\Omega}_0\mathbf{x}_i\\
\mathbf{h}_1 &= \mathbf{a}[\mathbf{f}_0]\\
\mathbf{f}_1 &= \boldsymbol{\beta}_1 + \boldsymbol{\Omega}_1\mathbf{h}_1\\
\mathbf{h}_2 &= \mathbf{a}[\mathbf{f}_1]\\
\mathbf{f}_2 &= \boldsymbol{\beta}_2 + \boldsymbol{\Omega}_2\mathbf{h}_2\\
\mathbf{h}_3 &= \mathbf{a}[\mathbf{f}_2]\\
\mathbf{f}_3 &= \boldsymbol{\beta}_3 + \boldsymbol{\Omega}_3\mathbf{h}_3\\
\ell_i &= \mathrm{l}[\mathbf{f}_3, y_i]
\end{aligned}
$$

$$
\frac{\partial \mathbf{f}_3}{\partial \mathbf{h}_3} = \frac{\partial}{\partial \mathbf{h}_3}\left(\boldsymbol{\beta}_3 + \boldsymbol{\Omega}_3\mathbf{h}_3\right) = \boldsymbol{\Omega}_3^T
$$

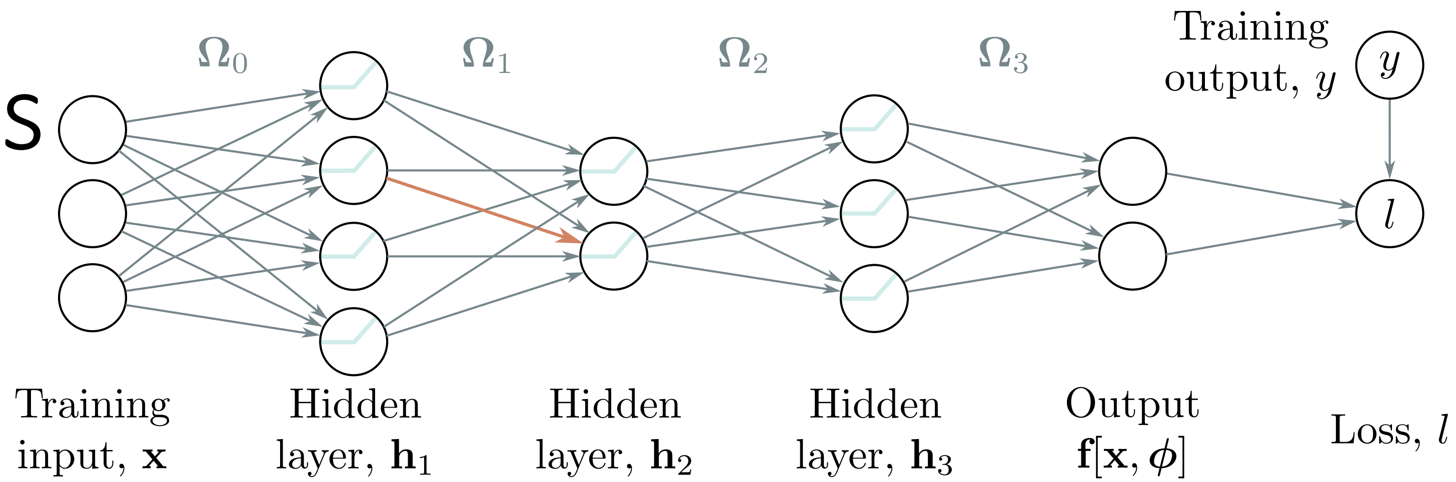
$$
\frac{\partial \ell_i}{\partial \mathbf{f}_3}
$$

$$
\frac{\partial \ell_i}{\partial \mathbf{f}_2} = \frac{\partial \mathbf{h}_3}{\partial \mathbf{f}_2}\frac{\partial \mathbf{f}_3}{\partial \mathbf{h}_3}\frac{\partial \ell_i}{\partial \mathbf{f}_3}
$$

$$
\frac{\partial \ell_i}{\partial \mathbf{f}_1} = \frac{\partial \mathbf{h}_2}{\partial \mathbf{f}_1}\frac{\partial \mathbf{f}_2}{\partial \mathbf{h}_2}\left(\frac{\partial \mathbf{h}_3}{\partial \mathbf{f}_2}\frac{\partial \mathbf{f}_3}{\partial \mathbf{h}_3}\frac{\partial \ell_i}{\partial \mathbf{f}_3}\right)
$$

$$
\frac{\partial \ell_i}{\partial \mathbf{f}_0} = \frac{\partial \mathbf{h}_1}{\partial \mathbf{f}_0}\frac{\partial \mathbf{f}_1}{\partial \mathbf{h}_1}\left(\frac{\partial \mathbf{h}_2}{\partial \mathbf{f}_1}\frac{\partial \mathbf{f}_2}{\partial \mathbf{h}_2}\frac{\partial \mathbf{h}_3}{\partial \mathbf{f}_2}\frac{\partial \mathbf{f}_3}{\partial \mathbf{h}_3}\frac{\partial \ell_i}{\partial \mathbf{f}_3}\right)
$$

The backward pass



1. Write this as a series of intermediate calculations

2. Compute these intermediate quantities

3. Take derivatives of output with respect to intermediate quantities

$$
\begin{aligned}
\mathbf{f}_0 &= \boldsymbol{\beta}_0 + \boldsymbol{\Omega}_0 \mathbf{x}_i \\
\mathbf{h}_1 &= \mathbf{a}[\mathbf{f}_0] \\
\mathbf{f}_1 &= \boldsymbol{\beta}_1 + \boldsymbol{\Omega}_1 \mathbf{h}_1 \\
\mathbf{h}_2 &= \mathbf{a}[\mathbf{f}_1] \\
\mathbf{f}_2 &= \boldsymbol{\beta}_2 + \boldsymbol{\Omega}_2 \mathbf{h}_2 \\
\mathbf{h}_3 &= \mathbf{a}[\mathbf{f}_2] \\
\mathbf{f}_3 &= \boldsymbol{\beta}_3 + \boldsymbol{\Omega}_3 \mathbf{h}_3 \\
\ell_i &= \mathrm{l}[\mathbf{f}_3, y_i]
\end{aligned}
$$

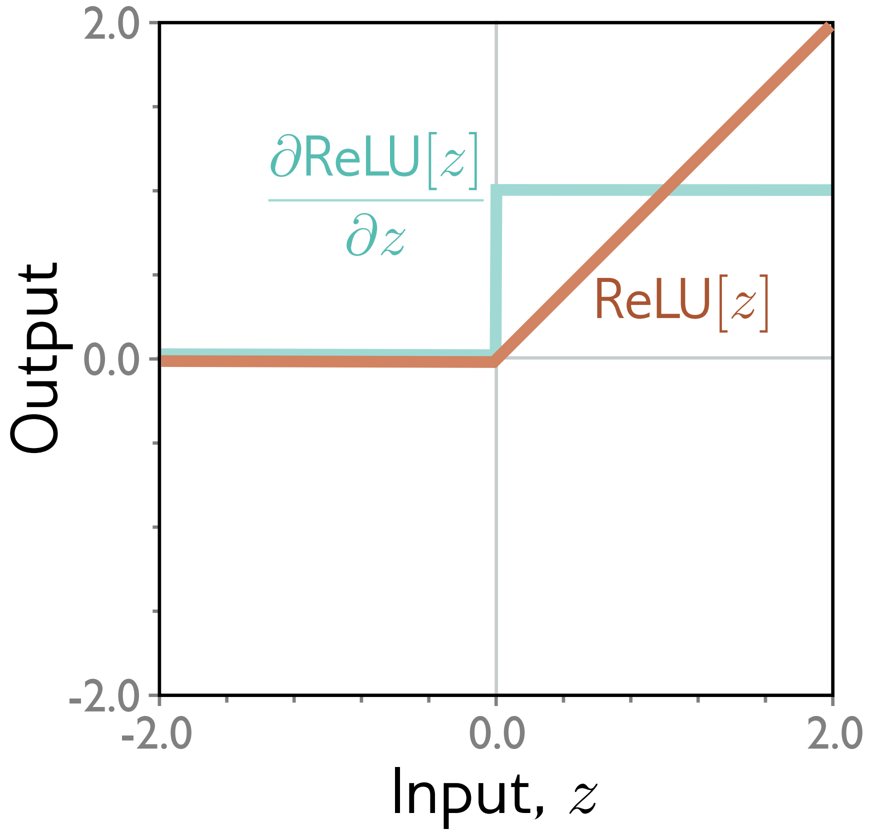
$$\frac{\partial \ell_i}{\partial \mathbf{f}_3}$$

$$\frac{\partial \ell_i}{\partial \mathbf{f}_2} = \boxed{\frac{\partial \mathbf{h}_3}{\partial \mathbf{f}_2}} \frac{\partial \mathbf{f}_3}{\partial \mathbf{h}_3} \frac{\partial \ell_i}{\partial \mathbf{f}_3}$$

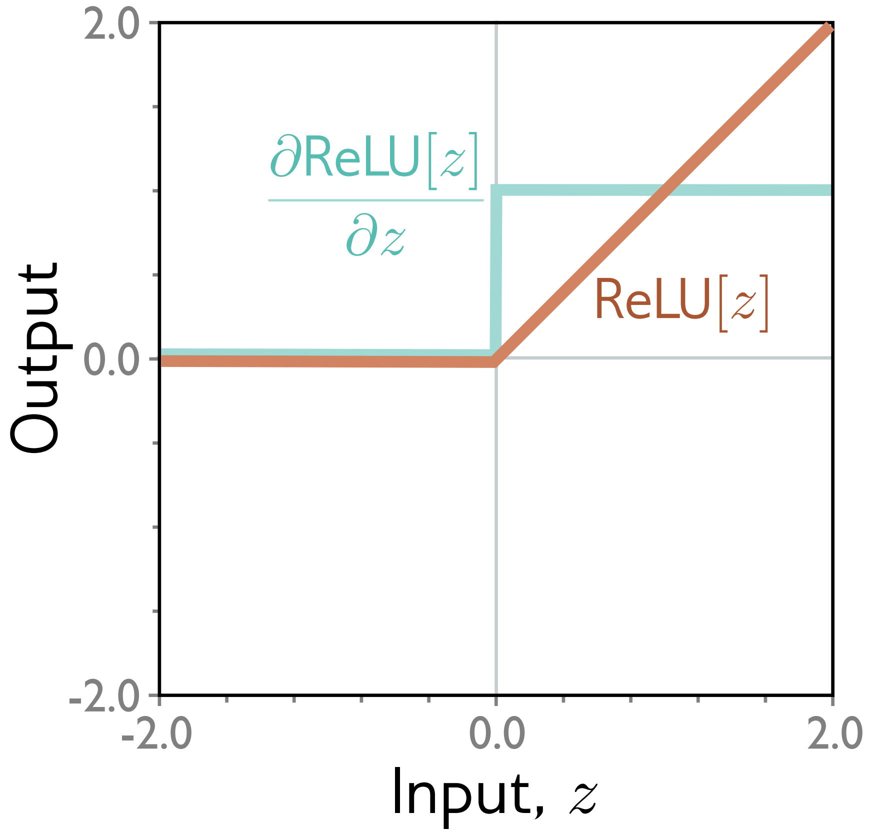
$$\frac{\partial \ell_i}{\partial \mathbf{f}_1} = \frac{\partial \mathbf{h}_2}{\partial \mathbf{f}_1} \frac{\partial \mathbf{f}_2}{\partial \mathbf{h}_2} \left(\frac{\partial \mathbf{h}_3}{\partial \mathbf{f}_2} \frac{\partial \mathbf{f}_3}{\partial \mathbf{h}_3} \frac{\partial \ell_i}{\partial \mathbf{f}_3} \right)$$

$$\frac{\partial \ell_i}{\partial \mathbf{f}_0} = \frac{\partial \mathbf{h}_1}{\partial \mathbf{f}_0} \frac{\partial \mathbf{f}_1}{\partial \mathbf{h}_1} \left(\frac{\partial \mathbf{h}_2}{\partial \mathbf{f}_1} \frac{\partial \mathbf{f}_2}{\partial \mathbf{h}_2} \frac{\partial \mathbf{h}_3}{\partial \mathbf{f}_2} \frac{\partial \mathbf{f}_3}{\partial \mathbf{h}_3} \frac{\partial \ell_i}{\partial \mathbf{f}_3} \right)$$

Derivative of ReLU



Derivative of ReLU



$$\mathbb{I}[z > 0]$$

"Indicator function"

Derivative of RELU

1. Consider:

$$\mathbf{a} = \mathbf{ReLU}[\mathbf{b}]$$

where:

$$\mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

2. We could equivalently write:

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} \text{ReLU}[b_1] \\ \text{ReLU}[b_2] \\ \text{ReLU}[b_3] \end{bmatrix}$$

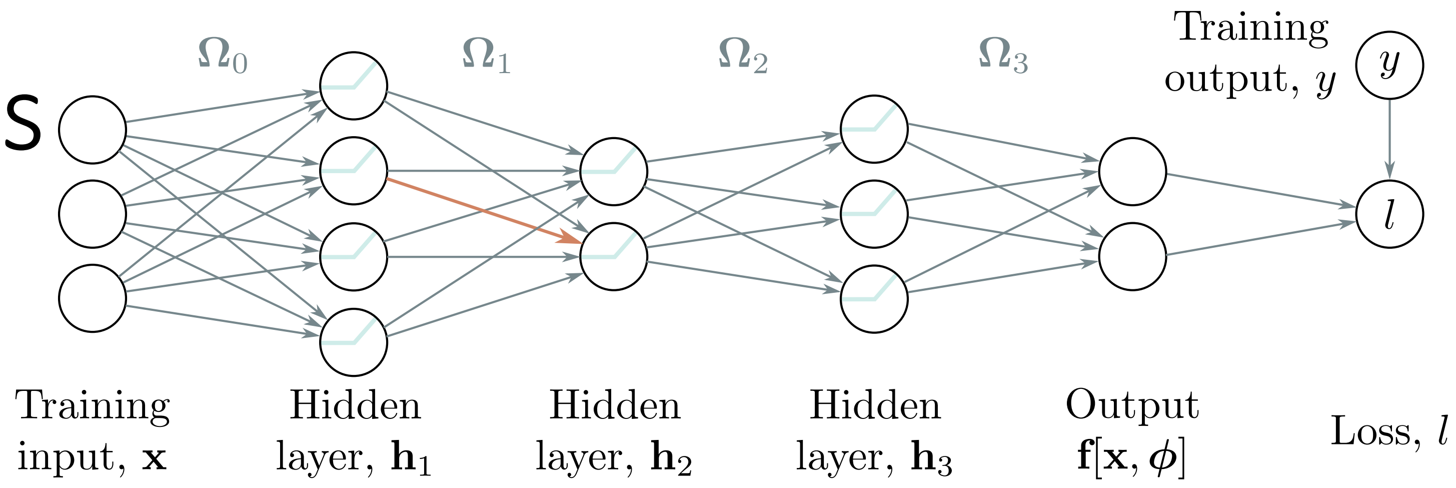
3. Taking the derivative

$$\frac{\partial \mathbf{a}}{\partial \mathbf{b}} = \begin{bmatrix} \frac{\partial a_1}{\partial b_1} & \frac{\partial a_2}{\partial b_1} & \frac{\partial a_3}{\partial b_1} \\ \frac{\partial a_1}{\partial b_2} & \frac{\partial a_2}{\partial b_2} & \frac{\partial a_3}{\partial b_2} \\ \frac{\partial a_1}{\partial b_3} & \frac{\partial a_2}{\partial b_3} & \frac{\partial a_3}{\partial b_3} \end{bmatrix} = \begin{bmatrix} \mathbb{I}[b_1 > 0] & 0 & 0 \\ 0 & \mathbb{I}[[b_2 > 0] & 0 \\ 0 & 0 & \mathbb{I}[b_3 > 0] \end{bmatrix}$$

4. We can equivalently pointwise multiply by diagonal

$$\mathbb{I}[\mathbf{b} > 0] \odot$$

The backward pass



1. Write this as a series of intermediate calculations

2. Compute these intermediate quantities

3. Take derivatives of output with respect to intermediate quantities

$$
\begin{aligned}
\mathbf{f}_0 &= \boldsymbol{\beta}_0 + \boldsymbol{\Omega}_0 \mathbf{x}_i \\
\mathbf{h}_1 &= \mathbf{a}[\mathbf{f}_0] \\
\mathbf{f}_1 &= \boldsymbol{\beta}_1 + \boldsymbol{\Omega}_1 \mathbf{h}_1 \\
\mathbf{h}_2 &= \mathbf{a}[\mathbf{f}_1] \\
\mathbf{f}_2 &= \boldsymbol{\beta}_2 + \boldsymbol{\Omega}_2 \mathbf{h}_2 \\
\mathbf{h}_3 &= \mathbf{a}[\mathbf{f}_2] \\
\mathbf{f}_3 &= \boldsymbol{\beta}_3 + \boldsymbol{\Omega}_3 \mathbf{h}_3 \\
\ell_i &= \mathrm{l}[\mathbf{f}_3, y_i]
\end{aligned}
$$

$$\mathbb{I}[\mathbf{f}_2 > 0]$$

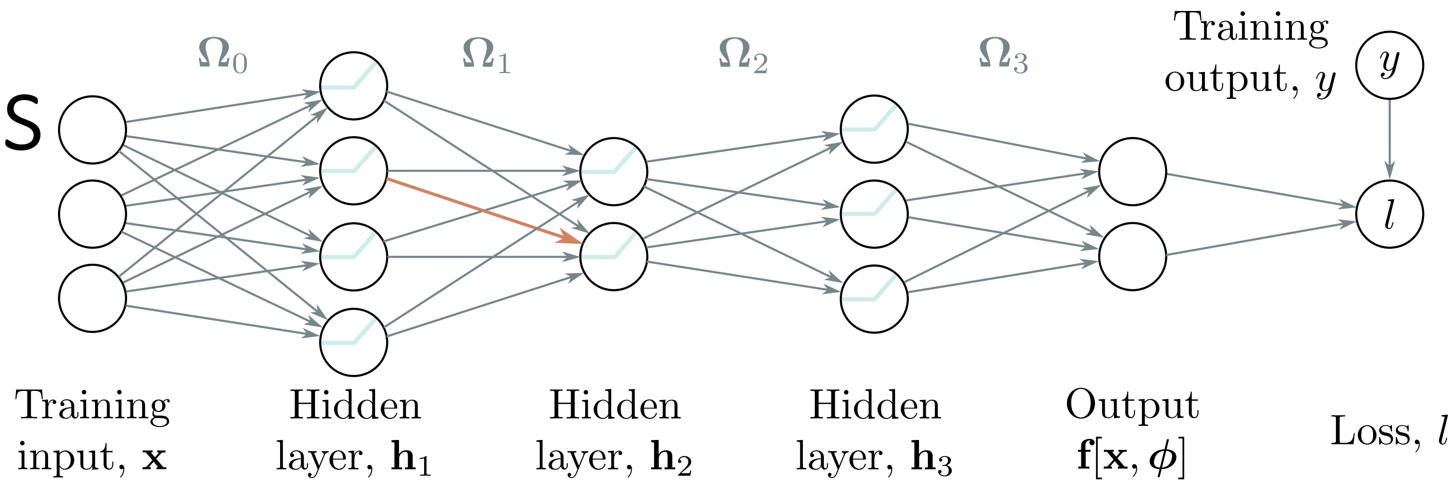
$$\frac{\partial \ell_i}{\partial \mathbf{f}_3}$$

$$\frac{\partial \ell_i}{\partial \mathbf{f}_2} = \boxed{\frac{\partial \mathbf{h}_3}{\partial \mathbf{f}_2}} \frac{\partial \mathbf{f}_3}{\partial \mathbf{h}_3} \frac{\partial \ell_i}{\partial \mathbf{f}_3}$$

$$\frac{\partial \ell_i}{\partial \mathbf{f}_1} = \frac{\partial \mathbf{h}_2}{\partial \mathbf{f}_1} \frac{\partial \mathbf{f}_2}{\partial \mathbf{h}_2} \left(\frac{\partial \mathbf{h}_3}{\partial \mathbf{f}_2} \frac{\partial \mathbf{f}_3}{\partial \mathbf{h}_3} \frac{\partial \ell_i}{\partial \mathbf{f}_3} \right)$$

$$\frac{\partial \ell_i}{\partial \mathbf{f}_0} = \frac{\partial \mathbf{h}_1}{\partial \mathbf{f}_0} \frac{\partial \mathbf{f}_1}{\partial \mathbf{h}_1} \left(\frac{\partial \mathbf{h}_2}{\partial \mathbf{f}_1} \frac{\partial \mathbf{f}_2}{\partial \mathbf{h}_2} \frac{\partial \mathbf{h}_3}{\partial \mathbf{f}_2} \frac{\partial \mathbf{f}_3}{\partial \mathbf{h}_3} \frac{\partial \ell_i}{\partial \mathbf{f}_3} \right)$$

The backward pass



1. Write this as a series of intermediate calculations

2. Compute these intermediate quantities

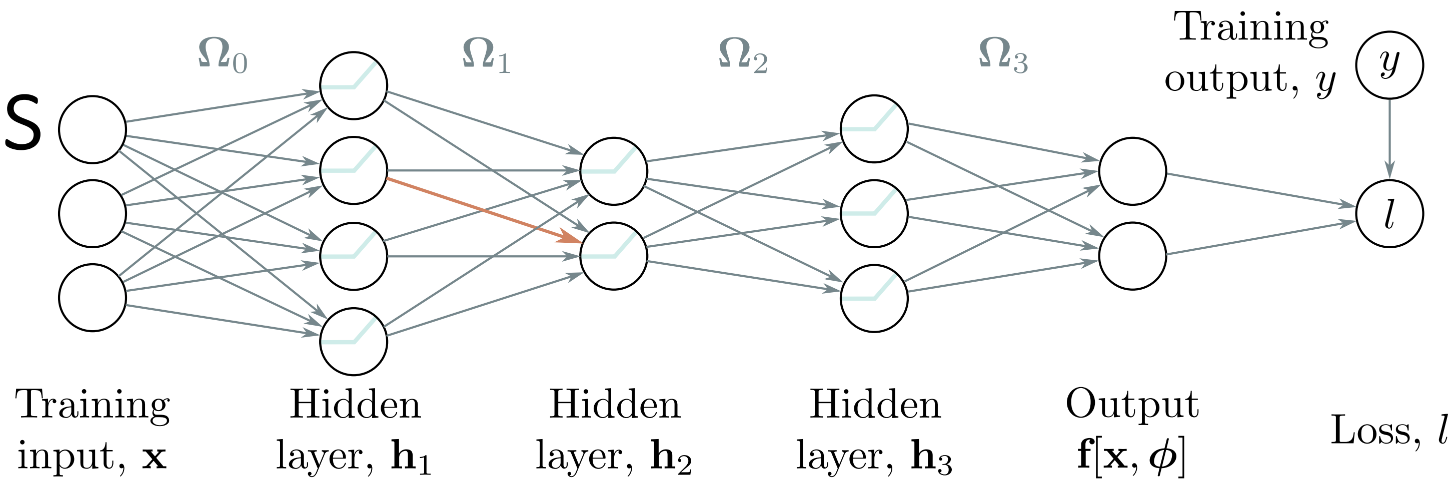
3. Take derivatives of output with respect to intermediate quantities

4. Take derivatives w.r.t. parameters

$$
\begin{aligned}
\mathbf{f}_0 &= \boldsymbol{\beta}_0 + \boldsymbol{\Omega}_0 \mathbf{x}_i \\
\mathbf{h}_1 &= \mathbf{a}[\mathbf{f}_0] \\
\mathbf{f}_1 &= \boldsymbol{\beta}_1 + \boldsymbol{\Omega}_1 \mathbf{h}_1 \\
\mathbf{h}_2 &= \mathbf{a}[\mathbf{f}_1] \\
\mathbf{f}_2 &= \boldsymbol{\beta}_2 + \boldsymbol{\Omega}_2 \mathbf{h}_2 \\
\mathbf{h}_3 &= \mathbf{a}[\mathbf{f}_2] \\
\mathbf{f}_3 &= \boldsymbol{\beta}_3 + \boldsymbol{\Omega}_3 \mathbf{h}_3 \\
\ell_i &= \mathrm{l}[\mathbf{f}_3, y_i]
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial \ell_i}{\partial \boldsymbol{\beta}_k} &= \frac{\partial \mathbf{f}_k}{\partial \boldsymbol{\beta}_k} \frac{\partial \ell_i}{\partial \mathbf{f}_k} \\
&= \frac{\partial}{\partial \boldsymbol{\beta}_k} \left(\boldsymbol{\beta}_k + \boldsymbol{\Omega}_k \mathbf{h}_k \right) \frac{\partial \ell_i}{\partial \mathbf{f}_k} \\
&= \frac{\partial \ell_i}{\partial \mathbf{f}_k},
\end{aligned}
$$

The backward pass



1. Write this as a series of intermediate calculations

2. Compute these intermediate quantities

3. Take derivatives of output with respect to intermediate quantities

4. Take derivatives w.r.t. parameters

$$
\begin{aligned}
\mathbf{f}_0 &= \boldsymbol{\beta}_0 + \boldsymbol{\Omega}_0 \mathbf{x}_i \\
\mathbf{h}_1 &= \mathbf{a}[\mathbf{f}_0] \\
\mathbf{f}_1 &= \boldsymbol{\beta}_1 + \boldsymbol{\Omega}_1 \mathbf{h}_1 \\
\mathbf{h}_2 &= \mathbf{a}[\mathbf{f}_1] \\
\mathbf{f}_2 &= \boldsymbol{\beta}_2 + \boldsymbol{\Omega}_2 \mathbf{h}_2 \\
\mathbf{h}_3 &= \mathbf{a}[\mathbf{f}_2] \\
\mathbf{f}_3 &= \boldsymbol{\beta}_3 + \boldsymbol{\Omega}_3 \mathbf{h}_3 \\
\ell_i &= \mathrm{l}[\mathbf{f}_3, y_i]
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial \ell_i}{\partial \boldsymbol{\Omega}_k} &= \frac{\partial \mathbf{f}_k}{\partial \boldsymbol{\Omega}_k} \frac{\partial \ell_i}{\partial \mathbf{f}_k} \\
&= \frac{\partial}{\partial \boldsymbol{\Omega}_k} \left(\boldsymbol{\beta}_k + \boldsymbol{\Omega}_k \mathbf{h}_k \right) \frac{\partial \ell_i}{\partial \mathbf{f}_k} \\
&= \frac{\partial \ell_i}{\partial \mathbf{f}_k} \mathbf{h}_k^T
\end{aligned}
$$

Backprop summary

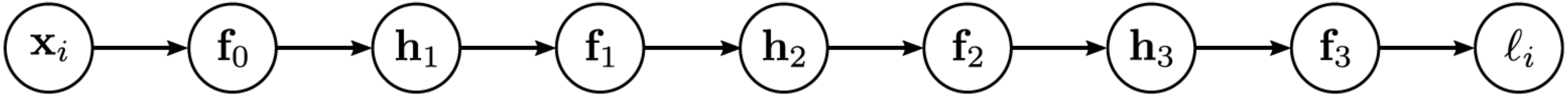
Forward pass: We compute and store the following quantities:

$$
\begin{aligned}
\mathbf{f}_0 &= \boldsymbol{\beta}_0 + \boldsymbol{\Omega}_0 \mathbf{x}_i \\
\mathbf{h}_k &= \mathbf{a}[\mathbf{f}_{k-1}] & k \in \{1, 2, \ldots K\} \\
\mathbf{f}_k &= \boldsymbol{\beta}_k + \boldsymbol{\Omega}_k \mathbf{h}_k. & k \in \{1, 2, \ldots K\}
\end{aligned}
$$

Backprop summary

Forward pass: We compute and store the following quantities:

$$\begin{aligned}
\mathbf{f}_0 &= \boldsymbol{\beta}_0 + \boldsymbol{\Omega}_0 \mathbf{x}_i \\
\mathbf{h}_k &= \mathbf{a}[\mathbf{f}_{k-1}] && k \in \{1, 2, \ldots K\} \\
\mathbf{f}_k &= \boldsymbol{\beta}_k + \boldsymbol{\Omega}_k \mathbf{h}_k. && k \in \{1, 2, \ldots K\}
\end{aligned}$$



Backprop summary

Backward pass: We start with the derivative $\partial \ell_i / \partial \mathbf{f}_K$ of the loss function ℓ_i with respect to the network output $\mathbf{f}_K$ and work backward through the network:

$$
\begin{aligned}
\frac{\partial \ell_i}{\partial \boldsymbol{\beta}_k} &= \frac{\partial \ell_i}{\partial \mathbf{f}_k} && k \in \{K, K-1, \ldots 1\} \\
\frac{\partial \ell_i}{\partial \boldsymbol{\Omega}_k} &= \frac{\partial \ell_i}{\partial \mathbf{f}_k} \mathbf{h}_k^T && k \in \{K, K-1, \ldots 1\} \\
\frac{\partial \ell_i}{\partial \mathbf{f}_{k-1}} &= \mathbb{I}[\mathbf{f}_{k-1} > 0] \odot \left(\boldsymbol{\Omega}_k^T \frac{\partial \ell_i}{\partial \mathbf{f}_k} \right), && k \in \{K, K-1, \ldots 1\} \qquad (7.13)
\end{aligned}
$$

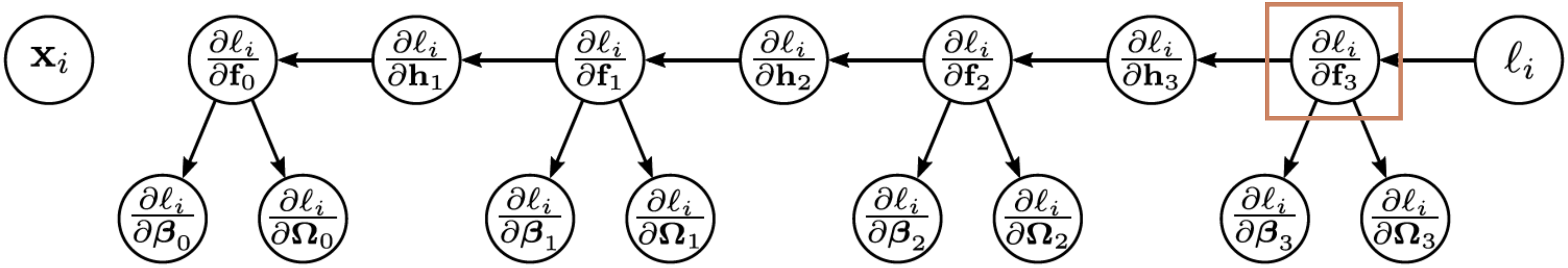
where $\odot$ denotes pointwise multiplication and $\mathbb{I}[\mathbf{f}_{k-1} > 0]$ is a vector containing ones where $\mathbf{f}_{k-1}$ is greater than zero and zeros elsewhere.

Backprop summary

Backward pass: We start with the derivative $\partial \ell_i / \partial \mathbf{f}_K$ of the loss function ℓ_i with respect to the network output $\mathbf{f}_K$ and work backward through the network:

$$\begin{aligned}
\frac{\partial \ell_i}{\partial \boldsymbol{\beta}_k} &= \frac{\partial \ell_i}{\partial \mathbf{f}_k} & k \in \{K, K-1, \ldots 1\} \\
\frac{\partial \ell_i}{\partial \boldsymbol{\Omega}_k} &= \frac{\partial \ell_i}{\partial \mathbf{f}_k}\mathbf{h}_k^T & k \in \{K, K-1, \ldots 1\} \\
\frac{\partial \ell_i}{\partial \mathbf{f}_{k-1}} &= \mathbb{I}[\mathbf{f}_{k-1} > 0] \odot \left(\boldsymbol{\Omega}_k^T \frac{\partial \ell_i}{\partial \mathbf{f}_k} \right), & k \in \{K, K-1, \ldots 1\} \qquad (7.13)
\end{aligned}$$

where $\odot$ denotes pointwise multiplication and $\mathbb{I}[\mathbf{f}_{k-1} > 0]$ is a vector containing ones where $\mathbf{f}_{k-1}$ is greater than zero and zeros elsewhere.

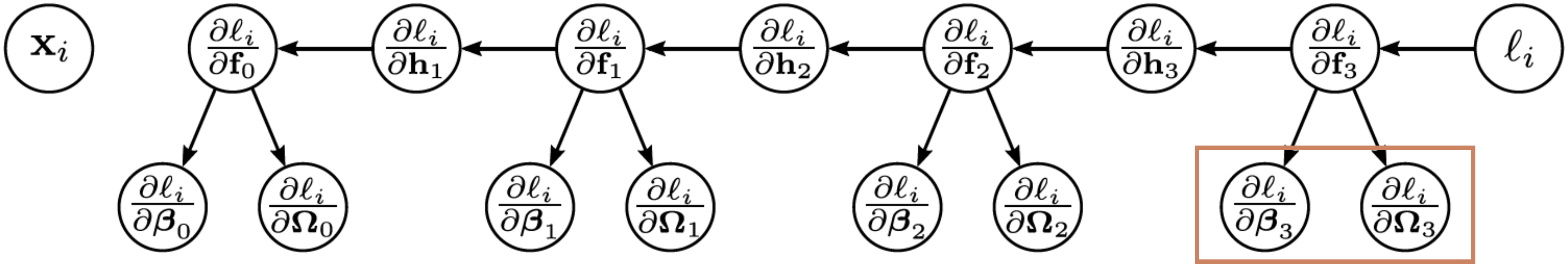


Backprop summary

Backward pass: We start with the derivative $\partial \ell_i / \partial \mathbf{f}_K$ of the loss function ℓ_i with respect to the network output $\mathbf{f}_K$ and work backward through the network:

$$
\begin{aligned}
\frac{\partial \ell_i}{\partial \boldsymbol{\beta}_k} &= \frac{\partial \ell_i}{\partial \mathbf{f}_k} && k \in \{K, K-1, \dots 1\} \\
\frac{\partial \ell_i}{\partial \boldsymbol{\Omega}_k} &= \frac{\partial \ell_i}{\partial \mathbf{f}_k} \mathbf{h}_k^T && k \in \{K, K-1, \dots 1\} \\
\frac{\partial \ell_i}{\partial \mathbf{f}_{k-1}} &= \mathbb{I}[\mathbf{f}_{k-1} > 0] \odot \left(\boldsymbol{\Omega}_k^T \frac{\partial \ell_i}{\partial \mathbf{f}_k} \right), && k \in \{K, K-1, \dots 1\} \qquad (7.13)
\end{aligned}
$$

where $\odot$ denotes pointwise multiplication and $\mathbb{I}[\mathbf{f}_{k-1} > 0]$ is a vector containing ones where $\mathbf{f}_{k-1}$ is greater than zero and zeros elsewhere.

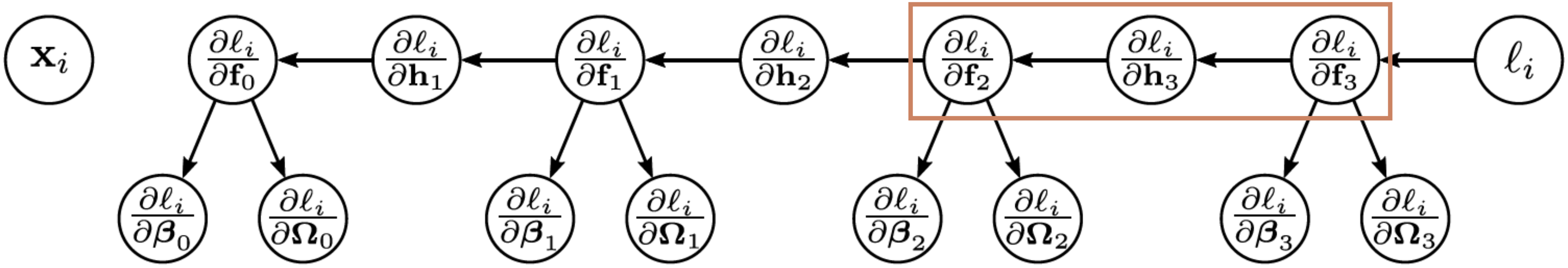


Backprop summary

Backward pass: We start with the derivative $\partial \ell_i / \partial \mathbf{f}_K$ of the loss function ℓ_i with respect to the network output $\mathbf{f}_K$ and work backward through the network:

$$
\begin{aligned}
\frac{\partial \ell_i}{\partial \boldsymbol{\beta}_k} &= \frac{\partial \ell_i}{\partial \mathbf{f}_k} && k \in \{K, K-1, \ldots 1\} \\
\frac{\partial \ell_i}{\partial \boldsymbol{\Omega}_k} &= \frac{\partial \ell_i}{\partial \mathbf{f}_k} \mathbf{h}_k^T && k \in \{K, K-1, \ldots 1\} \\
\frac{\partial \ell_i}{\partial \mathbf{f}_{k-1}} &= \mathbb{I}[\mathbf{f}_{k-1} > 0] \odot \left(\boldsymbol{\Omega}_k^T \frac{\partial \ell_i}{\partial \mathbf{f}_k} \right), && k \in \{K, K-1, \ldots 1\} \qquad (7.13)
\end{aligned}
$$

where $\odot$ denotes pointwise multiplication and $\mathbb{I}[\mathbf{f}_{k-1} > 0]$ is a vector containing ones where $\mathbf{f}_{k-1}$ is greater than zero and zeros elsewhere.



Backprop summary

Backward pass: We start with the derivative $\partial \ell_i / \partial \mathbf{f}_K$ of the loss function ℓ_i with respect to the network output $\mathbf{f}_K$ and work backward through the network:

$$\begin{aligned}
\frac{\partial \ell_i}{\partial \boldsymbol{\beta}_k} &= \frac{\partial \ell_i}{\partial \mathbf{f}_k} && k \in \{K, K-1, \ldots 1\} \\
\frac{\partial \ell_i}{\partial \boldsymbol{\Omega}_k} &= \frac{\partial \ell_i}{\partial \mathbf{f}_k} \mathbf{h}_k^T && k \in \{K, K-1, \ldots 1\} \\
\frac{\partial \ell_i}{\partial \mathbf{f}_{k-1}} &= \mathbb{I}[\mathbf{f}_{k-1} > 0] \odot \left(\boldsymbol{\Omega}_k^T \frac{\partial \ell_i}{\partial \mathbf{f}_k} \right), && k \in \{K, K-1, \ldots 1\}
\end{aligned} \tag{7.13}$$

where $\odot$ denotes pointwise multiplication and $\mathbb{I}[\mathbf{f}_{k-1} > 0]$ is a vector containing ones where $\mathbf{f}_{k-1}$ is greater than zero and zeros elsewhere. Finally, we compute the derivatives with respect to the first set of biases and weights:

$$\begin{aligned}
\frac{\partial \ell_i}{\partial \boldsymbol{\beta}_0} &= \frac{\partial \ell_i}{\partial \mathbf{f}_0} \\
\frac{\partial \ell_i}{\partial \boldsymbol{\Omega}_0} &= \frac{\partial \ell_i}{\partial \mathbf{f}_0} \mathbf{x}_i^T
\end{aligned}$$

Pros and cons

- Extremely efficient
  - Only need matrix multiplication and thresholding for ReLU functions
- Memory hungry – must store all the intermediate quantities
- Sequential
  - can process multiple batches in parallel
  - but things get harder if the whole model doesn’t fit on one machine.


GEFORC
RTX

Gradients

- Backpropagation intuition
- Toy model
- Background mathematics
- Backpropagation forward pass
- Backpropagation backward pass
- Algorithmic differentiation
- Code

Algorithmic differentiation

- Modern deep learning frameworks compute derivatives automatically
- You just have to specify the model and the loss
- How? Algorithmic differentiation
  - Each component knows how to compute its own derivative
    - ReLU knows how to compute deriv of output w.r.t. input
    - Linear function knows how to compute deriv of output w.r.t. input
    - Linear function knows how to compute deriv of output w.r.t. parameter
  - You specify how the order of the components
  - It can compute the chain of derivatives
- Works with branches as long as it's still an acyclic graph

Gradients

- Backpropagation intuition
- Toy model
- Background mathematics
- Backpropagation forward pass
- Backpropagation backward pass
- Algorithmic differentiation
- Code

CM20315 - Machine Learning

Prof. Simon Prince

7b Initialization

Initialization

- Need for initialization
- He initialization
- Interlude: Expectations
- Show that $\mathbb{E}[f_i'] = 0$
- Write variance of pre-activations *f'* in terms of activations *h* in previous layer

$$\sigma^2_{f'} = \sigma^2_\Omega \sum_{j=1}^{D_h} \mathbb{E}\left[h_j^2\right]$$

- Write variance of pre-activations f' in terms of pre-activations f in previous layer

$$\sigma^2_{f'} = \frac{D_h \sigma^2_\Omega \sigma^2_f}{2}$$

Initialization

- Consider standard building block of NN in terms of preactivations:

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

- How do we initialize the biases and weights?
- Equivalent to choosing starting point in Gabor/Linear regression models

Initialization

- Consider standard building block of NN in terms of *preactivations*:

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

- Set all the biases to 0

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

- Weights 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?

Backprop summary

Backward pass: We start with the derivative $\partial \ell_i / \partial \mathbf{f}_K$ of the loss function ℓ_i with respect to the network output $\mathbf{f}_K$ and work backward through the network:

$$
\begin{aligned}
\frac{\partial \ell_i}{\partial \boldsymbol{\beta}_k} &= \frac{\partial \ell_i}{\partial \mathbf{f}_k} & k \in \{K, K-1, \ldots 1\} \\
\frac{\partial \ell_i}{\partial \boldsymbol{\Omega}_k} &= \frac{\partial \ell_i}{\partial \mathbf{f}_k} \mathbf{h}_k^T & k \in \{K, K-1, \ldots 1\} \\
\frac{\partial \ell_i}{\partial \mathbf{f}_{k-1}} &= \mathbb{I}[\mathbf{f}_{k-1} > 0] \odot \left(\boldsymbol{\Omega}_k^T \frac{\partial \ell_i}{\partial \mathbf{f}_k} \right), & k \in \{K, K-1, \ldots 1\}
\end{aligned}
\tag{7.13}
$$

where $\odot$ denotes pointwise multiplication and $\mathbb{I}[\mathbf{f}_{k-1} > 0]$ is a vector containing ones where $\mathbf{f}_{k-1}$ is greater than zero and zeros elsewhere. Finally, we compute the derivatives with respect to the first set of biases and weights:

$$
\begin{aligned}
\frac{\partial \ell_i}{\partial \boldsymbol{\beta}_0} &= \frac{\partial \ell_i}{\partial \mathbf{f}_0} \\
\frac{\partial \ell_i}{\partial \boldsymbol{\Omega}_0} &= \frac{\partial \ell_i}{\partial \mathbf{f}_0} \mathbf{x}_i^T
\end{aligned}
$$

Initialization

- Need for initialization
- He initialization
- Interlude: Expectations
- Show that $\mathbb{E}[f_i'] = 0$
- Write ariance of pre-activations *f'* in terms of activations *h* in previous layer

$$\sigma^2_{f'} = \sigma^2_{\Omega} \sum_{j=1}^{D_h} \mathbb{E}\left[h_j^2\right]$$

- Write variance of pre-activations f' in terms of pre-activations f in previous layer

$$\sigma^2_{f'} = \frac{D_h \sigma^2_{\Omega} \sigma^2_{f}}{2}$$

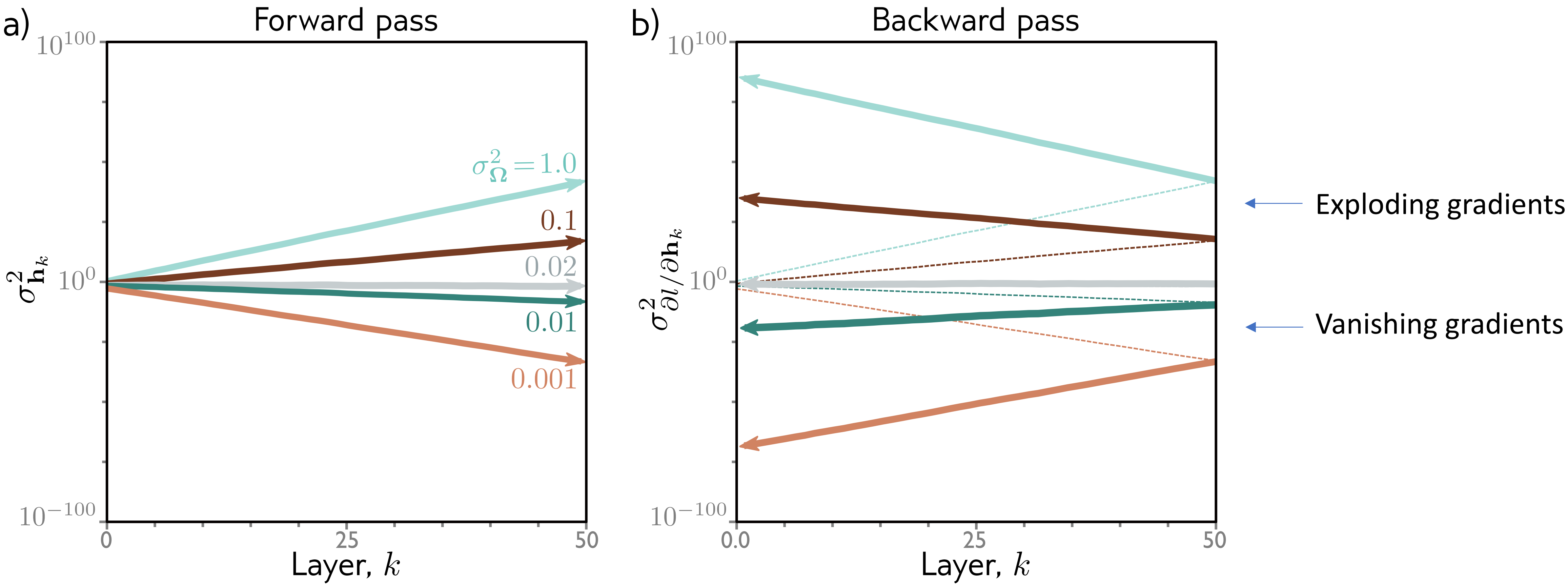


Figure 7.4 Weight initialization. Consider a deep network with 50 hidden layers and $D_h = 100$ hidden units per layer. The network has a 100 dimensional input $\mathbf{x}$ initialized with values from a standard normal distribution, a single output fixed at $y = 0$, and a least squares loss function.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

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:

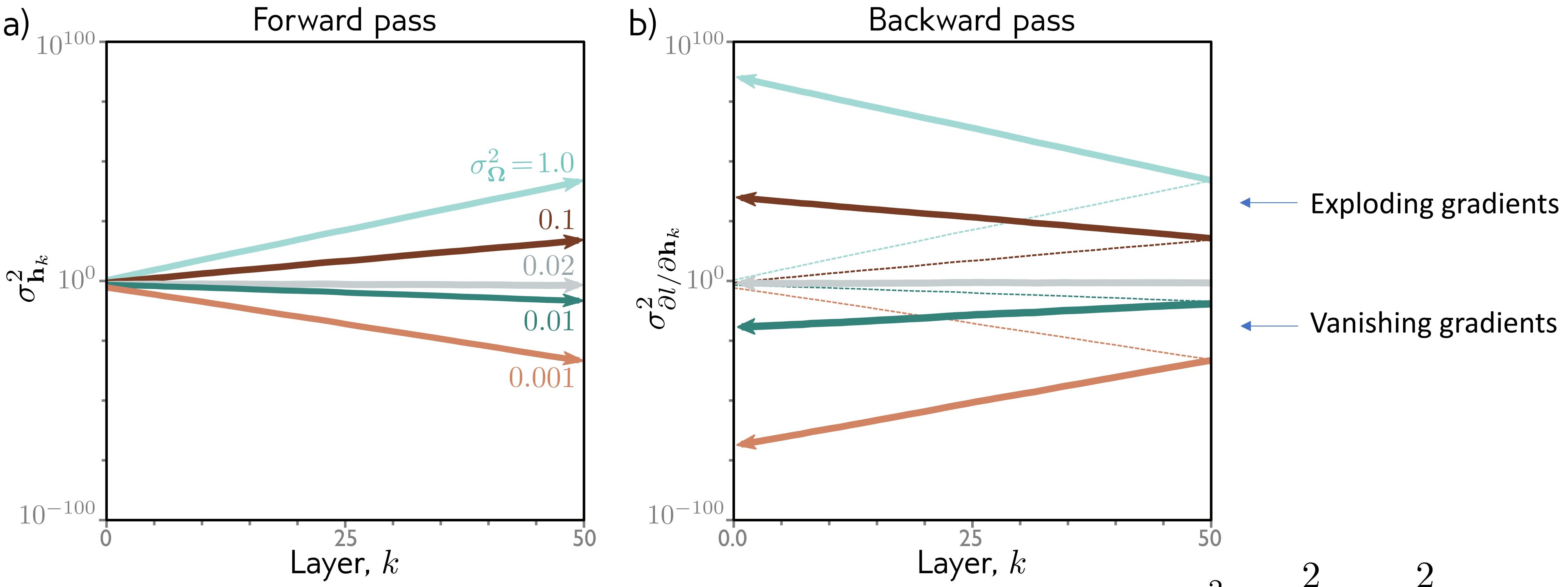
$$\sigma_\Omega^2 = \frac{2}{D_h}$$

← 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:

$$\sigma_\Omega^2 = \frac{2}{D_{h'}}$$

← Number of units at layer k+1



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

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 $\boldsymbol{\beta}_k$ are initialized to zero and the weight matrices $\boldsymbol{\Omega}_k$ are initialized with a normal distribution with mean zero and five different variances $\sigma^2_{\boldsymbol{\Omega}} \in \{0.001, 0.01, 0.02, 0.1, 1.0\}$. a)

Initialization

- Need for initialization
- He initialization
- Interlude: Expectations
- Show that $\mathbb{E}[f_i'] = 0$
- Write variance of pre-activations *f'* in terms of activations *h* in previous layer

$$\sigma_{f'}^2 = \sigma_\Omega^2 \sum_{j=1}^{D_h} \mathbb{E}\left[h_j^2\right]$$

- Write variance of pre-activations *f'* in terms of pre-activations *f* in previous layer

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

Expectations

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

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

Could approximate, by sampling many values of x from the distribution, calculating g[x], and taking average:

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

Expectations

Function $g[\bullet]$	Expectation
x	mean, μ
x^k	kth moment about zero
$(x-\mu)^k$	kth 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

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

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

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

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

Rule 1

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

$$\begin{aligned}\mathbb{E}\Big[\kappa\Big] &= \int \kappa Pr(x) dx \\ &= \kappa \int Pr(x) dx \\ &= \kappa.\end{aligned}$$

Rules for manipulating expectation

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

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

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

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

Rule 2

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

$$\begin{aligned}
\mathbb{E}\Big[\kappa \cdot \mathrm{g}[x]\Big] &= \int \kappa \cdot \mathrm{g}[x] Pr(x) dx \\
&= \kappa \cdot \int \mathrm{g}[x] Pr(x) dx \\
&= \kappa \cdot \mathbb{E}\Big[\mathrm{g}[x]\Big]
\end{aligned}$$

Rules for manipulating expectation

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

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

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

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

Rule 3

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

$$\begin{aligned}
\mathbb{E}\Big[\mathrm{f}[x] + \mathrm{g}[x]\Big] &= \int (\mathrm{f}[x] + \mathrm{g}[x]) Pr(x) dx \\
&= \int \left(\mathrm{f}[x] Pr(x) + \mathrm{g}[x] Pr(x)\right) dx \\
&= \int \mathrm{f}[x] Pr(x) dx + \int \mathrm{g}[x] Pr(x) dx \\
&= \mathbb{E}\Big[\mathrm{f}[x]\Big] + \mathbb{E}\Big[\mathrm{g}[x]\Big]
\end{aligned}$$

Rules for manipulating expectation

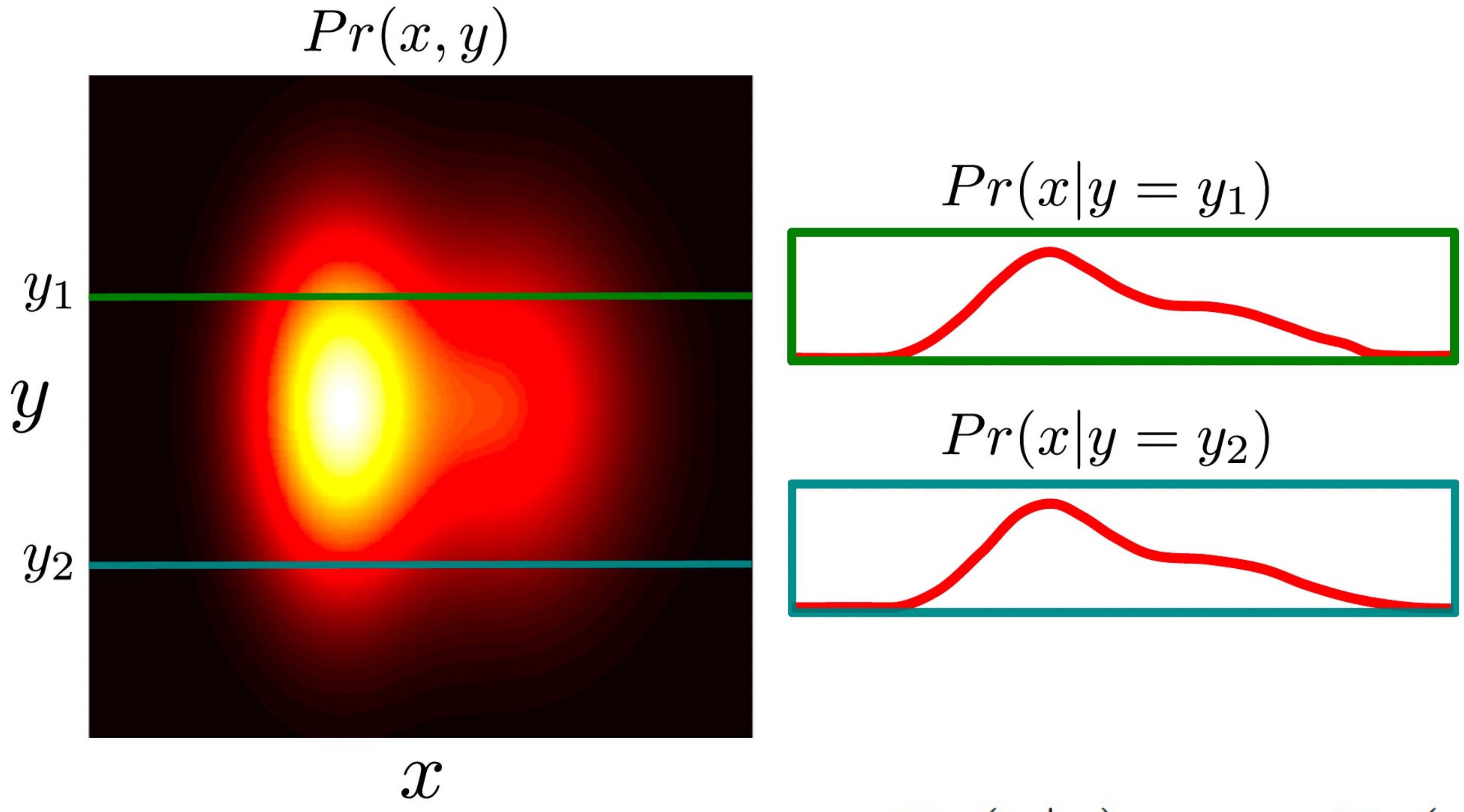
$$\mathbb{E}\Big[k\Big] = k$$

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

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

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

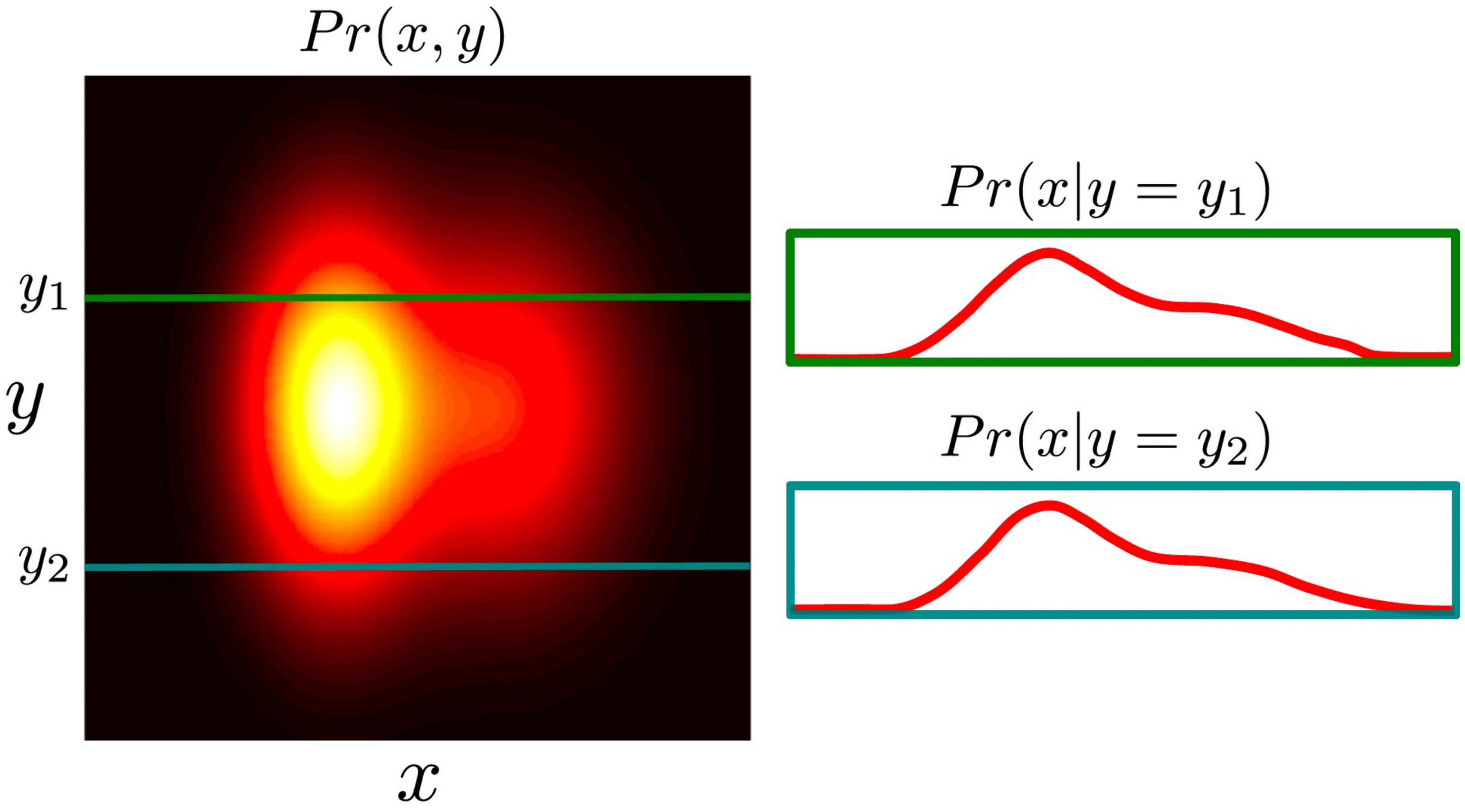
Independence



Probability of x and y

$$
\begin{aligned}
Pr(x|y) &= Pr(x) \\
Pr(y|x) &= Pr(y)
\end{aligned}
$$

Independence



$$Pr(x, y) = Pr(x)Pr(y)$$

Probability of x and y

Rule 4

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

$$\begin{aligned}
\mathbb{E}\Big[\mathrm{f}[x]\cdot\mathrm{g}[y]\Big] &= \int\int \mathrm{f}[x]\cdot\mathrm{g}[y] Pr(x,y) dx dy \\
&= \int\int \mathrm{f}[x]\cdot\mathrm{g}[y] Pr(x) Pr(y) dx dy \\
&= \int \mathrm{f}[x] Pr(x) dx \int \mathrm{g}[y] Pr(y) dy \\
&= \mathbb{E}\Big[\mathrm{f}[x]\Big]\mathbb{E}\Big[\mathrm{g}[y]\Big] \qquad \text{if} \quad x, y \quad \text{independent}
\end{aligned}$$

Because independent

Now let's prove:

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

Keeping in mind:

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

Now let's prove:

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

Keeping in mind:

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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: $\mathbb{E}\left[k\right] = k$

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

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

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

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

Initialization

- Need for initialization
- He initialization
- Interlude: Expectations
- Show that $\mathbb{E}[f_i'] = 0$
- Write variance of pre-activations *f'* in terms of activations *h* in previous layer

$$\sigma^2_{f'} = \sigma^2_{\Omega} \sum_{j=1}^{D_h} \mathbb{E}\left[h_j^2\right]$$

- Write variance of pre-activations f' in terms of pre-activations f in previous layer

$$\sigma^2_{f'} = \frac{D_h \sigma^2_{\Omega} \sigma^2_f}{2}$$

Initialization

- Consider standard building block of NN in terms of *preactivations*:

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

- Set all the biases to 0

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

- Weights 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?

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

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

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

Rule 4: $\mathbb{E}\Big[\mathrm{f}[x]g[y]\Big] = \mathbb{E}\Big[\mathrm{f}[x]\Big]\mathbb{E}\Big[\mathrm{g}[y]\Big] \quad \text{if} \quad x, y \quad \text{independent}$

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

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

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

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

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

Rule 4: $\mathbb{E}\Big[\mathrm{f}[x]g[y]\Big] = \mathbb{E}\Big[\mathrm{f}[x]\Big]\mathbb{E}\Big[\mathrm{g}[y]\Big] \quad \text{if} \quad x, y \quad \text{independent}$

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

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

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

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

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

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

Rule 4: $\mathbb{E}\Big[\mathrm{f}[x]g[y]\Big] = \mathbb{E}\Big[\mathrm{f}[x]\Big]\mathbb{E}\Big[\mathrm{g}[y]\Big] \quad \text{if} \quad x, y \quad \text{independent}$

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

Set all the biases to 0

Weights normally distributed

mean 0

variance σ_Ω^2

Initialization

- Need for initialization
- He initialization
- Interlude: Expectations
- Show that $\mathbb{E}[f_i'] = 0$
- Write variance of pre-activations *f'* in terms of activations *h* in previous layer

$$\sigma^2_{f'} = \sigma^2_{\Omega} \sum_{j=1}^{D_h} \mathbb{E}\left[h_j^2\right]$$

- Write variance of pre-activations f' in terms of pre-activations f in previous layer

$$\sigma^2_{f'} = \frac{D_h \sigma^2_{\Omega} \sigma^2_f}{2}$$

Aim: keep variance same between two layers

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

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

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

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

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

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

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

Rule 4: $\mathbb{E}\Big[\mathrm{f}[x]g[y]\Big] = \mathbb{E}\Big[\mathrm{f}[x]\Big]\mathbb{E}\Big[\mathrm{g}[y]\Big] \quad \text{if} \quad x, y \quad \text{independent}$

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

$$= \mathbb{E}\left[\left(\beta_i + \sum_{j=1}^{D_h} \Omega_{ij} h_j\right)^2\right] - 0$$

Set all the biases to 0

Weights normally distributed

mean 0

variance σ^2_Ω

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

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

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

Rule 4: $\mathbb{E}\Big[\mathrm{f}[x]g[y]\Big] = \mathbb{E}\Big[\mathrm{f}[x]\Big]\mathbb{E}\Big[\mathrm{g}[y]\Big] \quad \text{if} \quad x, y \quad \text{independent}$

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

Set all the biases to 0 ←

Weights normally distributed

mean 0

variance σ^2_Ω

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

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

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

Rule 4: $\mathbb{E}\left[\mathrm{f}[x]g[y]\right] = \mathbb{E}\left[\mathrm{f}[x]\right]\mathbb{E}\left[\mathrm{g}[y]\right] \quad \text{if} \quad x, y \quad \text{independent}$

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

Set all the biases to 0

Weights normally distributed

mean 0

variance σ_Ω^2

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

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

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

Rule 4: $\mathbb{E}\Big[\mathrm{f}[x] g[y]\Big] = \mathbb{E}\Big[\mathrm{f}[x]\Big]\mathbb{E}\Big[\mathrm{g}[y]\Big] \quad \text{if} \quad x, y \quad \text{independent}$

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

Set all the biases to 0

Weights normally distributed

mean 0

variance σ^2_Ω

Initialization

- Need for initialization
- He initialization
- Interlude: Expectations
- Show that $\mathbb{E}[f_i'] = 0$
- Write variance of pre-activations *f'* in terms of activations *h* in previous layer

$$\sigma^2_{f'} = \sigma^2_{\Omega} \sum_{j=1}^{D_h} \mathbb{E}\left[h_j^2\right]$$

- Write variance of pre-activations f' in terms of pre-activations f in previous layer

$$\sigma^2_{f'} = \frac{D_h \sigma^2_{\Omega} \sigma^2_f}{2}$$

$$
\begin{aligned}
\sigma^2_{f'} &= \sigma^2_\Omega \sum_{j=1}^{D_h} \mathbb{E}\left[h_j^2\right] \\
&= \sigma^2_\Omega \sum_{j=1}^{D_h} \mathbb{E}\left[\mathrm{ReLU}[f_j]^2\right] \\
&= \sigma^2_\Omega \sum_{j=1}^{D_h} \int_{-\infty}^{\infty} \mathrm{ReLU}[f_j]^2 Pr(f_j) df_j \\
&= \sigma^2_\Omega \sum_{j=1}^{D_h} \int_{-\infty}^{\infty} (\mathbb{I}[f_j > 0] f_j)^2 Pr(f_j) df_j \\
&= \sigma^2_\Omega \sum_{j=1}^{D_h} \int_{0}^{\infty} f_j^2 Pr(f_j) df_j \\
&= \sigma^2_\Omega \sum_{j=1}^{D_h} \frac{\sigma_f^2}{2} = \frac{D_h \sigma^2_\Omega \sigma^2_f}{2}
\end{aligned}
$$

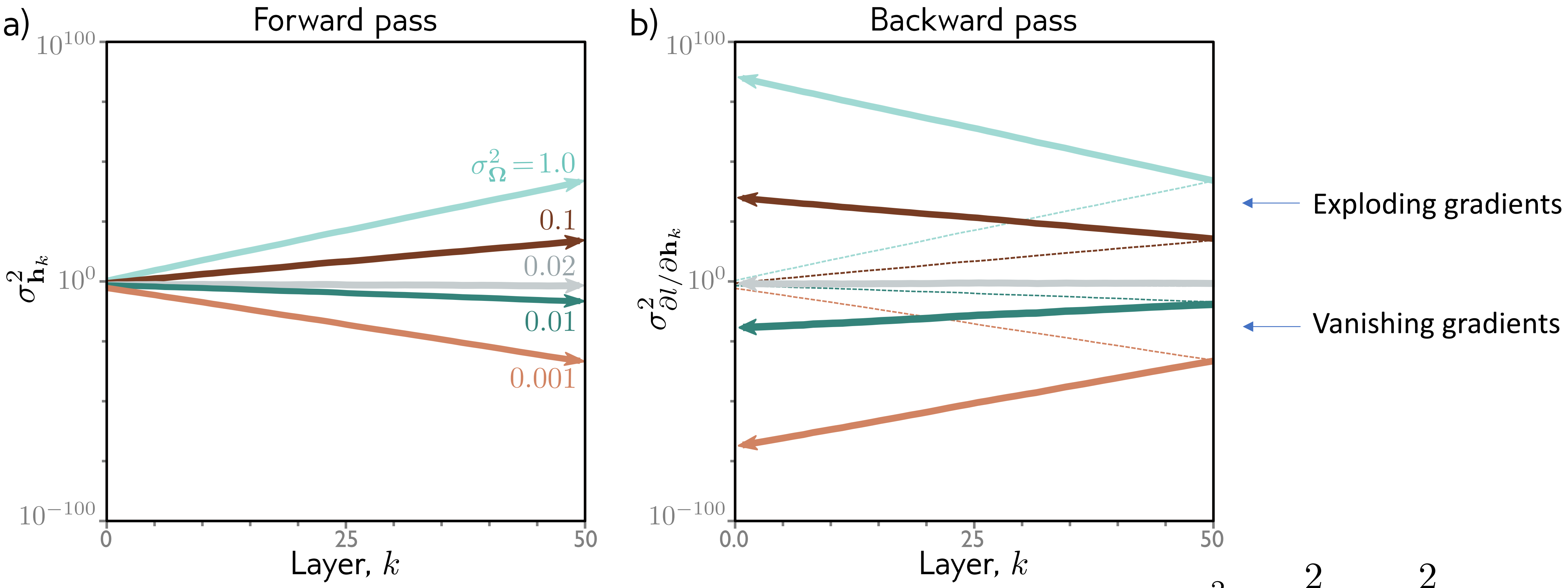
Aim: keep variance same between two layers

$$\sigma^2_{f'} = \frac{D_h \sigma^2_\Omega \sigma^2_f}{2}$$

Should choose:

$$\sigma^2_\Omega = \frac{2}{D_h}$$

This is called He initialization.



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

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 $\boldsymbol{\beta}_k$ are initialized to zero and the weight matrices $\boldsymbol{\Omega}_k$ are initialized with a normal distribution with mean zero and five different variances $\sigma^2_{\boldsymbol{\Omega}} \in \{0.001, 0.01, 0.02, 0.1, 1.0\}$. a)

PyTorch code

- Define a neural network
- Initialize params with He initialization
- Define loss function
- Choose optimization algorithm
- Choose initial learning rate
- Choose learning rates schedule
- Make some random data
- Train for 100 batches

```python
import torch, torch.nn as nn
from torch.utils.data import TensorDataset, DataLoader
from torch.optim.lr_scheduler import StepLR

# define input size, hidden layer size, output size
D_i, D_k, D_o = 10, 40, 5
# create model with two hidden layers
model = nn.Sequential(
   nn.Linear(D_i, D_k),
   nn.ReLU(),
   nn.Linear(D_k, D_k),
   nn.ReLU(),
   nn.Linear(D_k, D_o))

# He initialization of weights
def weights_init(layer_in):
   if isinstance(layer_in, nn.Linear):
      nn.init.kaiming_uniform(layer_in.weight)
      layer_in.bias.data.fill_(0.0)
model.apply(weights_init)

# choose least squares loss function
criterion = nn.MSELoss()
# construct SGD optimizer and initialize learning rate and momentum
optimizer = torch.optim.SGD(model.parameters(), lr = 0.01, momentum=0.9)
# object that decreases learning rate by half every 10 epochs
scheduler = StepLR(optimizer, step_size=10, gamma=0.5)

# create 100 dummy data points and store in data loader class
x = torch.randn(100, D_i)
y = torch.randn(100, D_o)
data_loader = DataLoader(TensorDataset(x,y), batch_size=10, shuffle=True)

# loop over the dataset 100 times
for epoch in range(100):
   epoch_loss = 0.0
   # loop over batches
   for i, data in enumerate(data_loader):
      # retrieve inputs and labels for this batch
      x_batch, y_batch = data
      # zero the parameter gradients
      optimizer.zero_grad()
      # forward pass
      pred = model(x_batch)
      loss = criterion(pred, y_batch)
      # backward pass
      loss.backward()
      # SGD update
      optimizer.step()
      # update statistics
      epoch_loss += loss.item()
   # print error
   print(f'Epoch {epoch:5d}, loss {epoch_loss:.3f}')
   # tell scheduler to consider updating learning rate
   scheduler.step()
```

PyTorch code

- Define a neural network
- Initialize params with He initialization
- Define loss function
- Choose optimization algorithm
- Choose initial learning rate
- Choose learning rates schedule
- Make some random data
- Train for 100 batches

```
import torch, torch.nn as nn
from torch.utils.data import TensorDataset, DataLoader
from torch.optim.lr_scheduler import StepLR

# define input size, hidden layer size, output size
D_i, D_k, D_o = 10, 40, 5
# create model with two hidden layers
model = nn.Sequential(
   nn.Linear(D_i, D_k),
   nn.ReLU(),
   nn.Linear(D_k, D_k),
   nn.ReLU(),
   nn.Linear(D_k, D_o))

# He initialization of weights
def weights_init(layer_in):
   if isinstance(layer_in, nn.Linear):
      nn.init.kaiming_uniform(layer_in.weight)
      layer_in.bias.data.fill_(0.0)
model.apply(weights_init)

# choose least squares loss function
criterion = nn.MSELoss()
# construct SGD optimizer and initialize learning rate and momentum
optimizer = torch.optim.SGD(model.parameters(), lr = 0.01, momentum=0.9)
# object that decreases learning rate by half every 10 epochs
scheduler = StepLR(optimizer, step_size=10, gamma=0.5)

# create 100 dummy data points and store in data loader class
x = torch.randn(100, D_i)
y = torch.randn(100, D_o)
data_loader = DataLoader(TensorDataset(x,y), batch_size=10, shuffle=True)

# loop over the dataset 100 times
for epoch in range(100):
   epoch_loss = 0.0
   # loop over batches
```

PyTorch code

- Define a neural network
- Initialize params with He initialization
- Define loss function
- Choose optimization algorithm
- Choose initial learning rate
- Choose learning rates schedule
- Make some random data
- Train for 100 batches

```
model.apply(weights_init)

# choose least squares loss function
criterion = nn.MSELoss()
# construct SGD optimizer and initialize learning rate and momentum
optimizer = torch.optim.SGD(model.parameters(), lr = 0.01, momentum=0.9)
# object that decreases learning rate by half every 10 epochs
scheduler = StepLR(optimizer, step_size=10, gamma=0.5)

# create 100 dummy data points and store in data loader class
x = torch.randn(100, D_i)
y = torch.randn(100, D_o)
data_loader = DataLoader(TensorDataset(x,y), batch_size=10, shuffle=True)

# loop over the dataset 100 times
for epoch in range(100):
   epoch_loss = 0.0
   # loop over batches
   for i, data in enumerate(data_loader):
      # retrieve inputs and labels for this batch
      x_batch, y_batch = data
      # zero the parameter gradients
      optimizer.zero_grad()
      # forward pass
      pred = model(x_batch)
      loss = criterion(pred, y_batch)
      # backward pass
      loss.backward()
      # SGD update
      optimizer.step()
      # update statistics
      epoch_loss += loss.item()
   # print error
   print(f'Epoch {epoch:5d}, loss {epoch_loss:.3f}')
   # tell scheduler to consider updating learning rate
   scheduler.step()
```