CPEN 455: Deep Learning

Lecture 4.1: Backpropagation and Initialization

Renjie Liao

University of British Columbia

Winter, Term 1, 2026

Outline

- Training Feedforward Neural Networks
  - Backpropagation
  - Weight Initialization

Outline

- Training Feedforward Neural Networks
  - **Backpropagation**
  - Weight Initialization

Learning Algorithm

Optimization

Fit parameters by reducing the training objective.

Learning Algorithm

Optimization

Fit parameters by reducing the training objective.

Gradient-based updates

SGD [1], Adam, and AdamW use gradients to choose parameter updates.

Learning Algorithm

Optimization

Fit parameters by reducing the training objective.

Gradient-based updates

SGD [1], Adam, and AdamW use gradients to choose parameter updates.

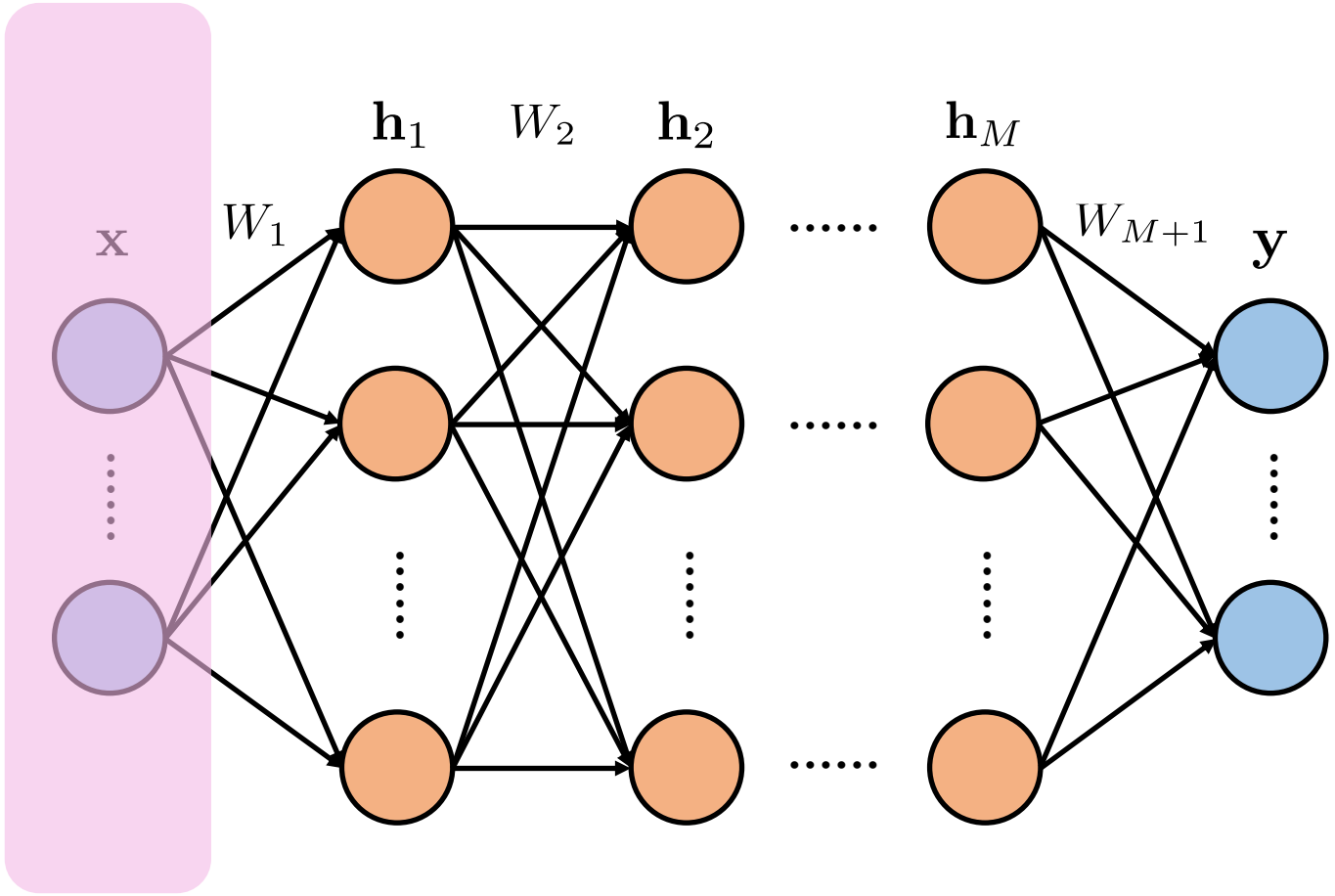
Backpropagation [2]

Compute those gradients efficiently by applying the chain rule backward.

Feedforward Neural Networks

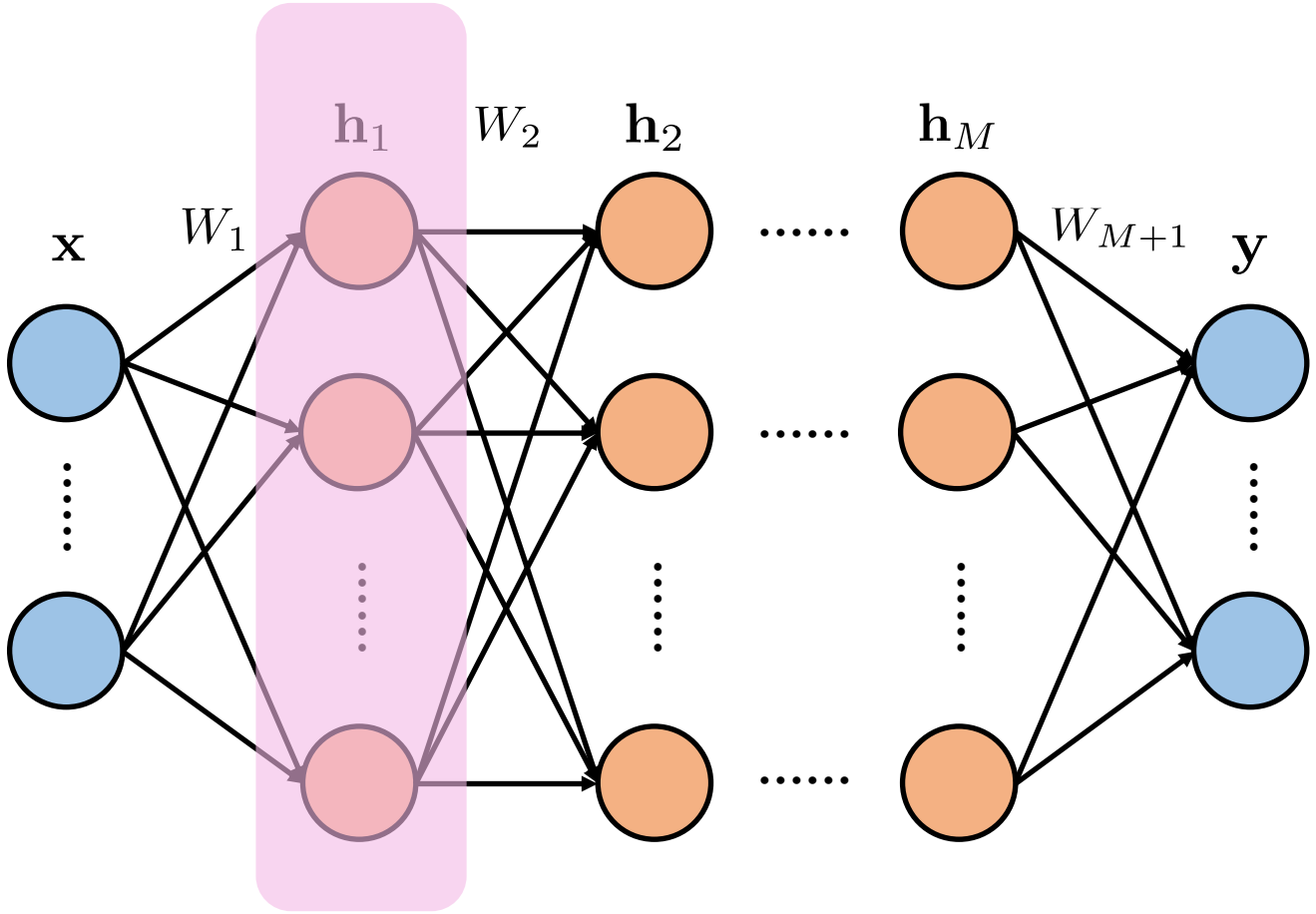
Consider an MLP. Biases are omitted here to simplify the derivation.

Input Sample



Feedforward Neural Networks

Consider an MLP. Biases are omitted here to simplify the derivation.

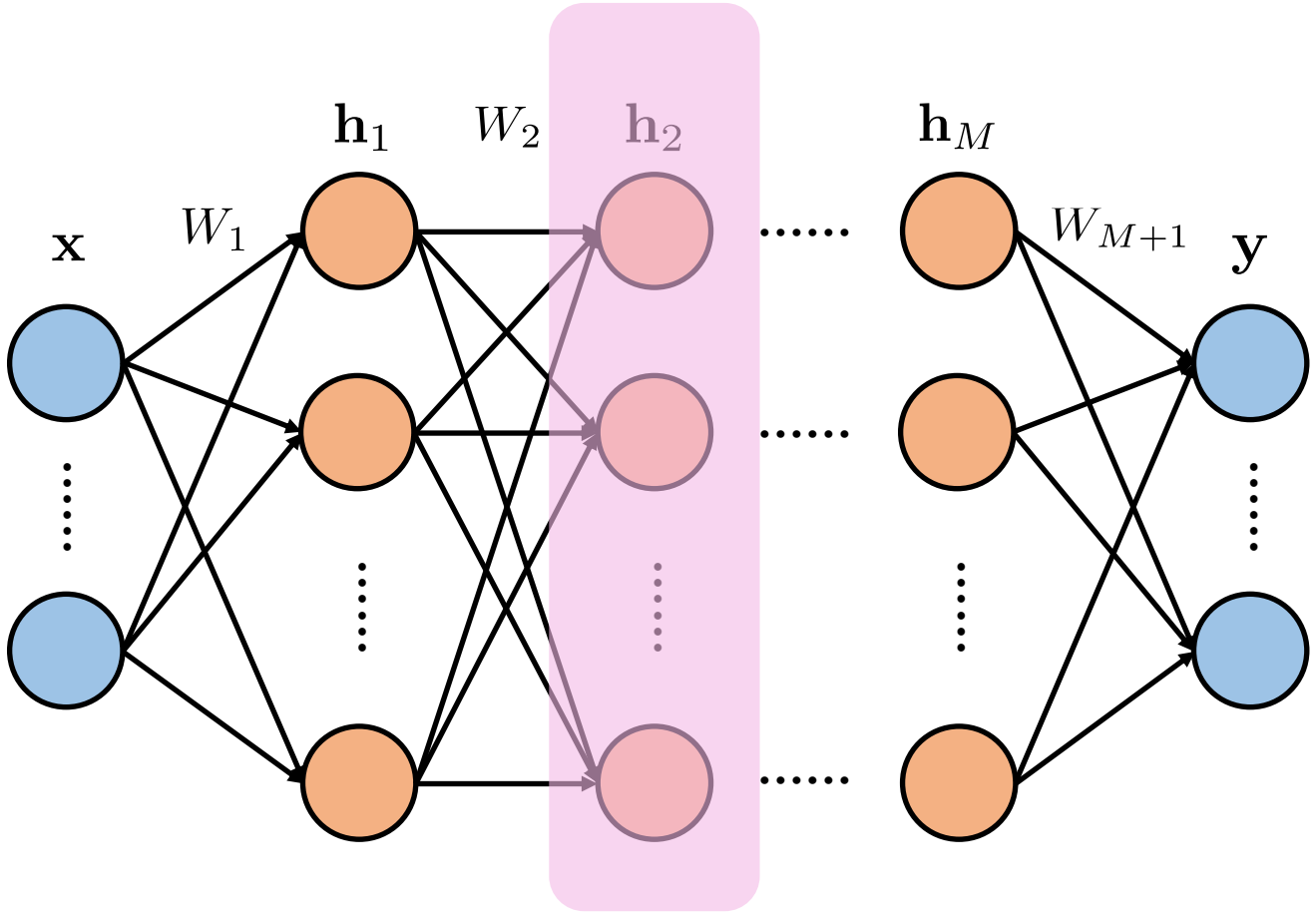


Input Sample

$$\mathbf{h}_1 = \sigma\left(W_1\mathbf{x}\right)$$

Feedforward Neural Networks

Consider an MLP. Biases are omitted here to simplify the derivation.



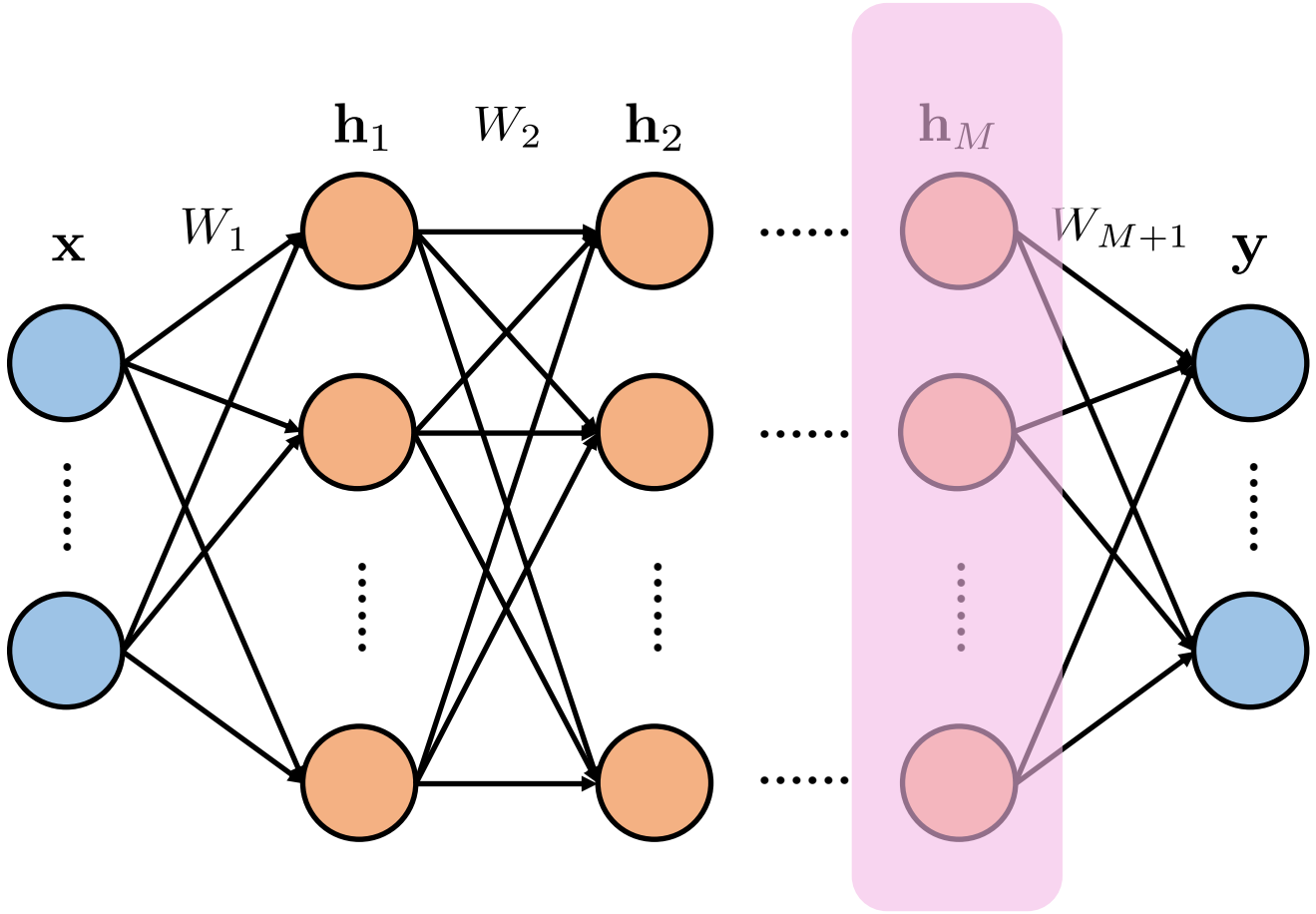
Input Sample

$$\mathbf{h}_1 = \sigma\left(W_1\mathbf{x}\right)$$

$$\mathbf{h}_2 = \sigma\left(W_2\mathbf{h}_1\right)$$

Feedforward Neural Networks

Consider an MLP. Biases are omitted here to simplify the derivation.



Input Sample

$$\mathbf{h}_1 = \sigma\left(W_1\mathbf{x}\right)$$

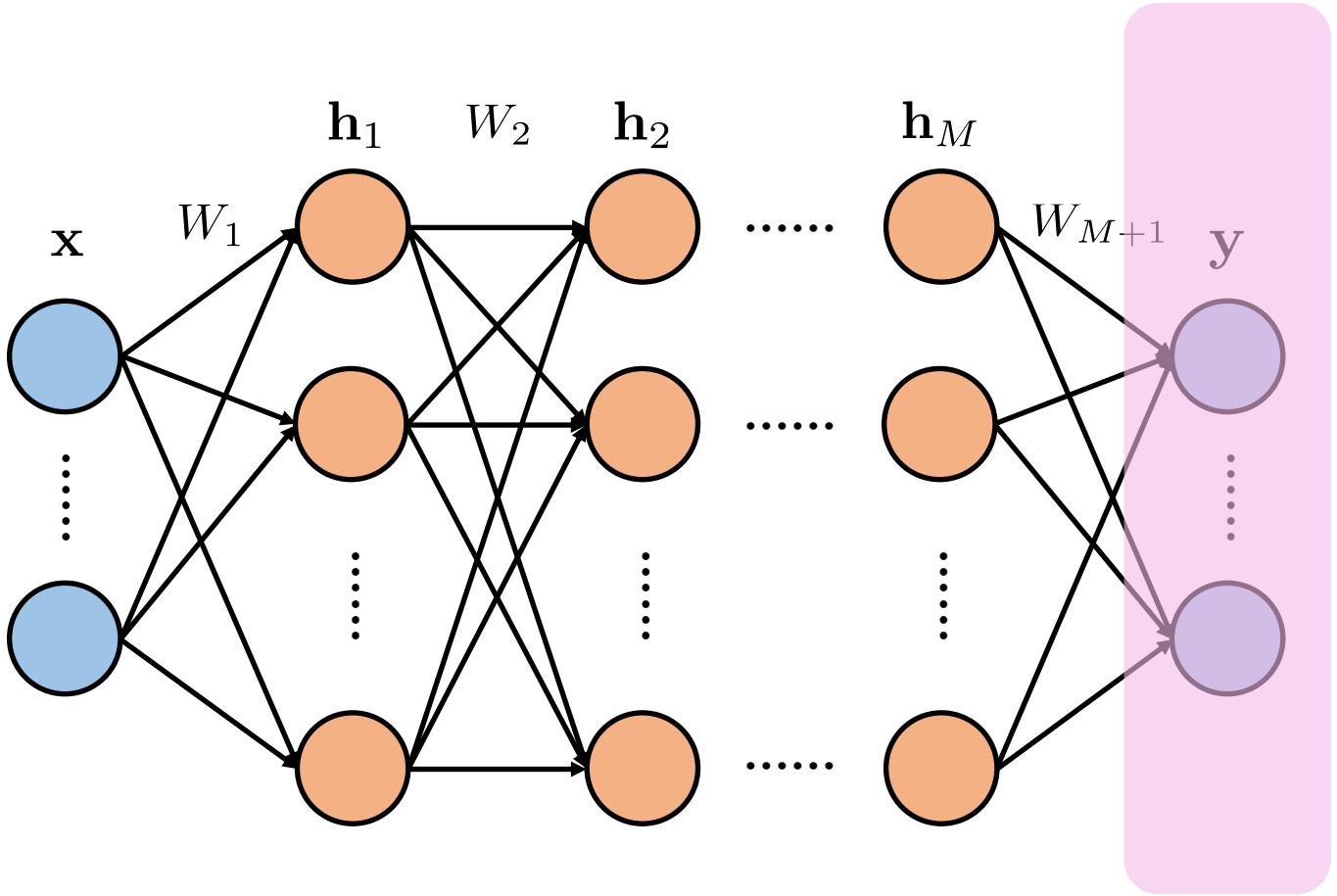
$$\mathbf{h}_2 = \sigma\left(W_2\mathbf{h}_1\right)$$

$$\vdots$$

$$\mathbf{h}_M = \sigma\left(W_M\mathbf{h}_{M-1}\right)$$

Feedforward Neural Networks

Consider an MLP. Biases are omitted here to simplify the derivation.



Input Sample

$$\mathbf{h}_1 = \sigma(W_1\mathbf{x})$$

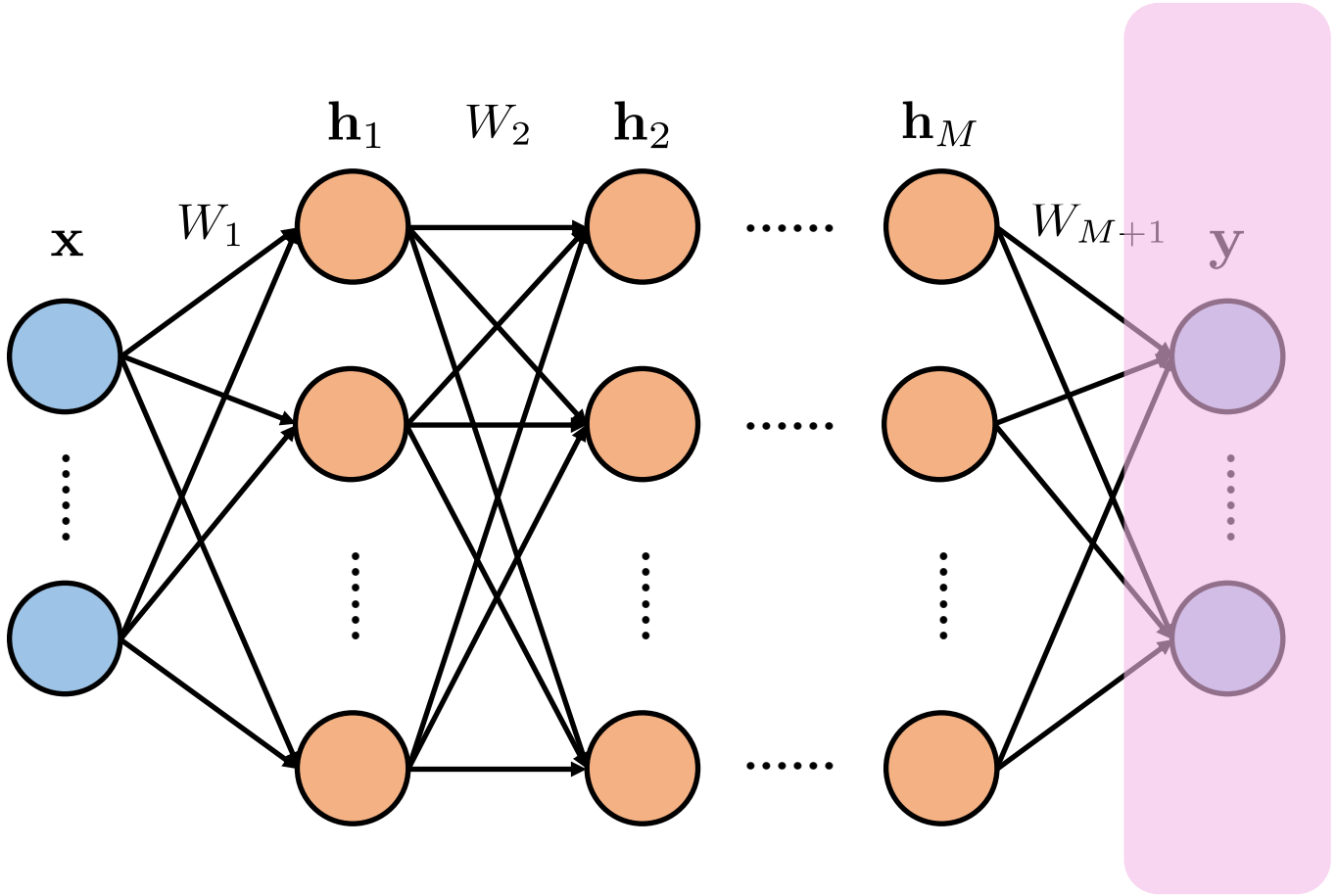
$$\mathbf{h}_2 = \sigma(W_2\mathbf{h}_1)$$

$$\mathbf{h}_M = \sigma(W_M\mathbf{h}_{M-1})$$

$$\mathbf{y} = W_{M+1}\mathbf{h}_M$$

Feedforward Neural Networks

Consider an MLP. Biases are omitted here to simplify the derivation.



Input Sample

$$\mathbf{h}_1 = \sigma\left(W_1\mathbf{x}\right)$$

$$\mathbf{h}_2 = \sigma\left(W_2\mathbf{h}_1\right)$$

$$\vdots$$

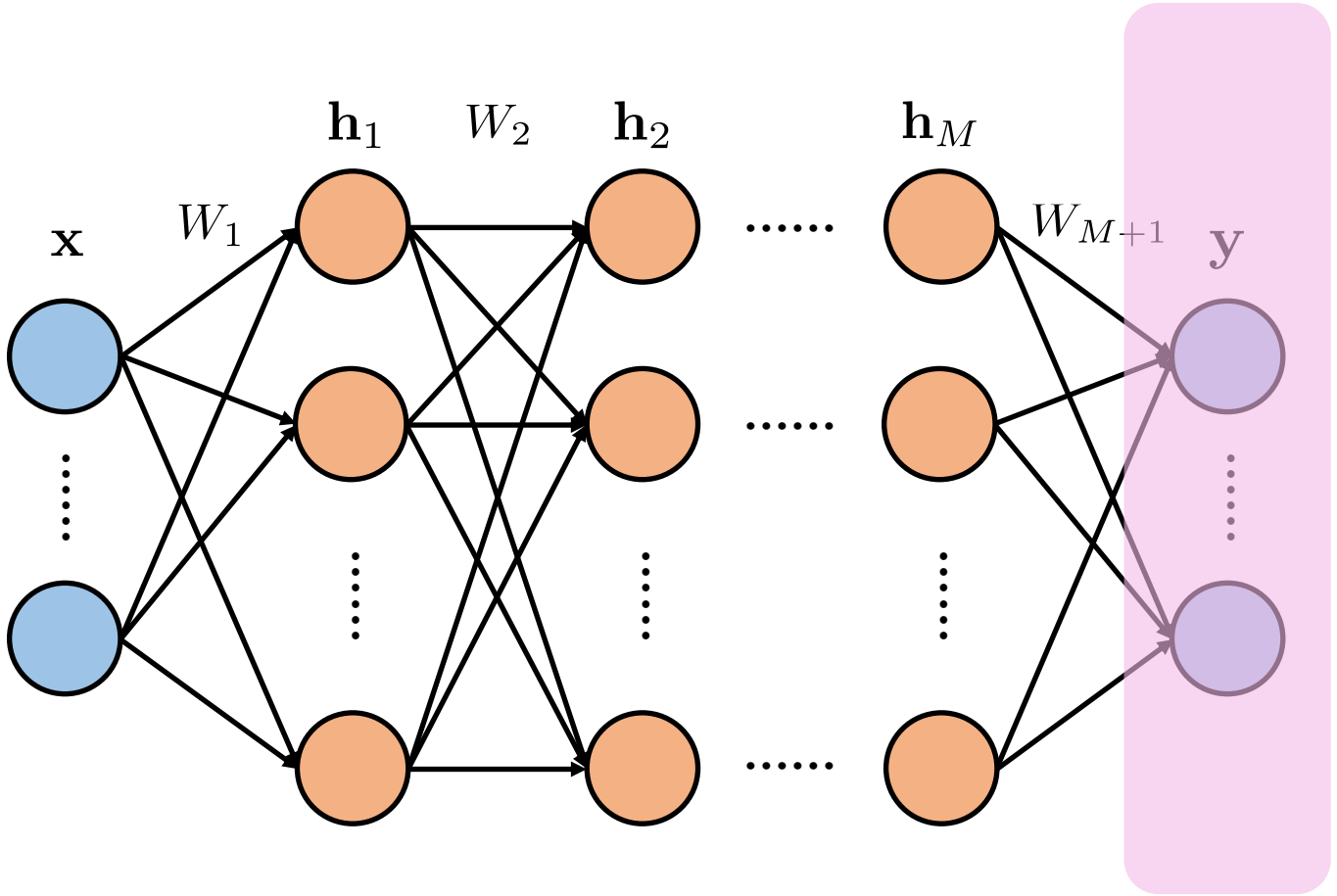
$$\mathbf{h}_M = \sigma\left(W_M\mathbf{h}_{M-1}\right)$$

$$\mathbf{y} = W_{M+1}\mathbf{h}_M$$

Loss: $L = \ell(\mathbf{y}, \bar{\mathbf{y}})$

Feedforward Neural Networks

Consider an MLP. Biases are omitted here to simplify the derivation.



Input Sample

$$\mathbf{h}_1 = \sigma(W_1\mathbf{x})$$

$$\mathbf{h}_2 = \sigma(W_2\mathbf{h}_1)$$

$$\mathbf{h}_M = \sigma(W_M\mathbf{h}_{M-1})$$

$$\mathbf{y} = W_{M+1}\mathbf{h}_M$$

Loss: $L = \ell(\mathbf{y}, \bar{\mathbf{y}})$

Mini-batch version:

$$L = \frac{1}{B}\sum_{i=1}^{B} \ell(\mathbf{y}_i, \bar{\mathbf{y}}_i)$$

Backpropagation

Before we introduce backpropagation, let us review several concepts in vector calculus.

Gradient: for a scalar-valued differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ of multiple variables, the gradient $\nabla f : \mathbb{R}^n \to \mathbb{R}^n$ evaluated at $\mathbf{p} = [\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n]^\top$ is

$$\nabla f(\mathbf{p}) = \begin{bmatrix} \frac{\partial f}{\partial x_1}(\mathbf{p}) \\ \vdots \\ \frac{\partial f}{\partial x_n}(\mathbf{p}) \end{bmatrix}$$

Backpropagation

Before we introduce backpropagation, let us review several concepts in vector calculus.

Gradient: for a scalar-valued differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ of multiple variables, the gradient $\nabla f : \mathbb{R}^n \to \mathbb{R}^n$ evaluated at $\mathbf{p} = [\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n]^\top$ is

$$\nabla f(\mathbf{p}) = \begin{bmatrix} \frac{\partial f}{\partial x_1}(\mathbf{p}) \\ \vdots \\ \frac{\partial f}{\partial x_n}(\mathbf{p}) \end{bmatrix}$$

Jacobian: for a vector-valued differentiable function $f : \mathbb{R}^n \to \mathbb{R}^m$ of multiple variables, the Jacobian matrix evaluated at $\mathbf{p} = [\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n]^\top$ is

$$\mathbf{J}_f(\mathbf{p}) = \begin{bmatrix} \nabla^{\mathrm{T}} f_1(\mathbf{p}) \\ \vdots \\ \nabla^{\mathrm{T}} f_m(\mathbf{p}) \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1}(\mathbf{p}) & \cdots & \frac{\partial f_1}{\partial x_n}(\mathbf{p}) \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1}(\mathbf{p}) & \cdots & \frac{\partial f_m}{\partial x_n}(\mathbf{p}) \end{bmatrix}$$

Backpropagation

Before we introduce backpropagation, let us review several concepts in vector calculus.

Chain Rule: the derivative of the composition of two differentiable functions in terms of the derivatives of individual functions

- Scalar-valued & single variable $f : \mathbb{R} \to \mathbb{R}$ $g : \mathbb{R} \to \mathbb{R}$

$$\frac{df(g(x))}{dx} = \frac{df(g(x))}{dg(x)} \frac{dg(x)}{dx}$$

Backpropagation

Before we introduce backpropagation, let us review several concepts in vector calculus.

Chain Rule: the derivative of the composition of two differentiable functions in terms of the derivatives of individual functions

- Scalar-valued & single variable $f : \mathbb{R} \to \mathbb{R}$ $g : \mathbb{R} \to \mathbb{R}$

$$\begin{aligned}\frac{df(g(x))}{dx} &= \frac{df(g(x))}{dg(x)}\frac{dg(x)}{dx} \\ &= \nabla f(g(x))\nabla g(x)\end{aligned}$$

- Scalar-valued & multiple variables $f : \mathbb{R}^m \to \mathbb{R}$ $g : \mathbb{R}^n \to \mathbb{R}^m$

$$\frac{df(g(x))}{dx} = \mathbf{J}_g(x)^\top \nabla f(g(x))$$

Vector derivatives in this lecture denote column gradients.

Backpropagation

Before we introduce backpropagation, let us review several concepts in vector calculus.

Chain Rule: the derivative of the composition of two differentiable functions in terms of the derivatives of individual functions

- Scalar-valued & single variable $f : \mathbb{R} \to \mathbb{R}$ $g : \mathbb{R} \to \mathbb{R}$

$$\begin{aligned}\frac{df(g(x))}{dx} &= \frac{df(g(x))}{dg(x)}\frac{dg(x)}{dx} \\ &= \nabla f(g(x))\nabla g(x)\end{aligned}$$

- Scalar-valued & multiple variables $f : \mathbb{R}^m \to \mathbb{R}$ $g : \mathbb{R}^n \to \mathbb{R}^m$

$$\frac{df(g(x))}{dx} = \mathbf{J}_g(x)^\top \nabla f(g(x))$$

We can derive it from the single variable case!

Backpropagation

Before we introduce backpropagation, let us review several concepts in vector calculus.

Chain Rule: the derivative of the composition of two differentiable functions in terms of the derivatives of individual functions

- Scalar-valued & multiple variables $f : \mathbb{R}^m \to \mathbb{R} \qquad g : \mathbb{R}^n \to \mathbb{R}^m$

$$\frac{df(g(x))}{dx} = \mathbf{J}_g(x)^\top \nabla f(g(x))$$

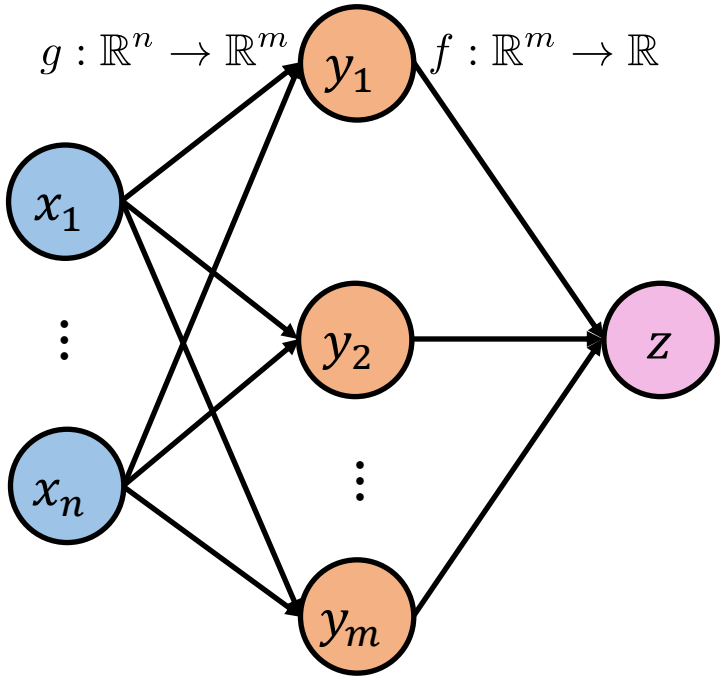
Backpropagation

Before we introduce backpropagation, let us review several concepts in vector calculus.

Chain Rule: the derivative of the composition of two differentiable functions in terms of the derivatives of individual functions

- Scalar-valued & multiple variables $f : \mathbb{R}^m \to \mathbb{R}$ $g : \mathbb{R}^n \to \mathbb{R}^m$

$$\frac{df(g(x))}{dx} = \mathbf{J}_g(x)^\top \nabla f(g(x)) = \frac{dz}{dx}$$



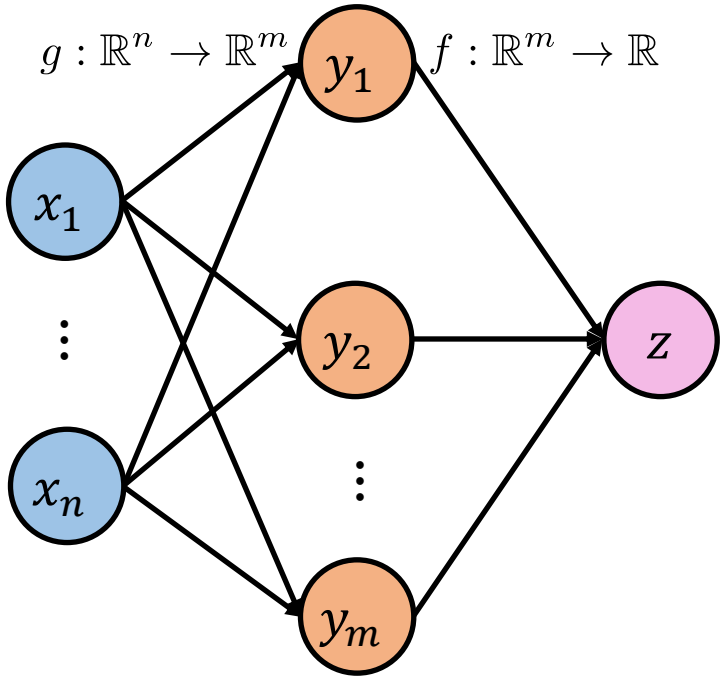
Backpropagation

Before we introduce backpropagation, let us review several concepts in vector calculus.

Chain Rule: the derivative of the composition of two differentiable functions in terms of the derivatives of individual functions

- Scalar-valued & multiple variables $f : \mathbb{R}^m \to \mathbb{R}$ $g : \mathbb{R}^n \to \mathbb{R}^m$

$$\frac{df(g(x))}{dx} = \mathbf{J}_g(x)^\top \nabla f(g(x)) = \frac{dz}{dx}$$



$$\frac{\partial z}{\partial x_1} = \sum_{i=1}^{m} \frac{\partial z}{\partial y_i} \frac{\partial y_i}{\partial x_1} = \begin{bmatrix} \frac{\partial z}{\partial y_1} \cdots \frac{\partial z}{\partial y_m} \end{bmatrix} \begin{bmatrix} \frac{\partial y_1}{\partial x_1} \\ \vdots \\ \frac{\partial y_m}{\partial x_1} \end{bmatrix}$$

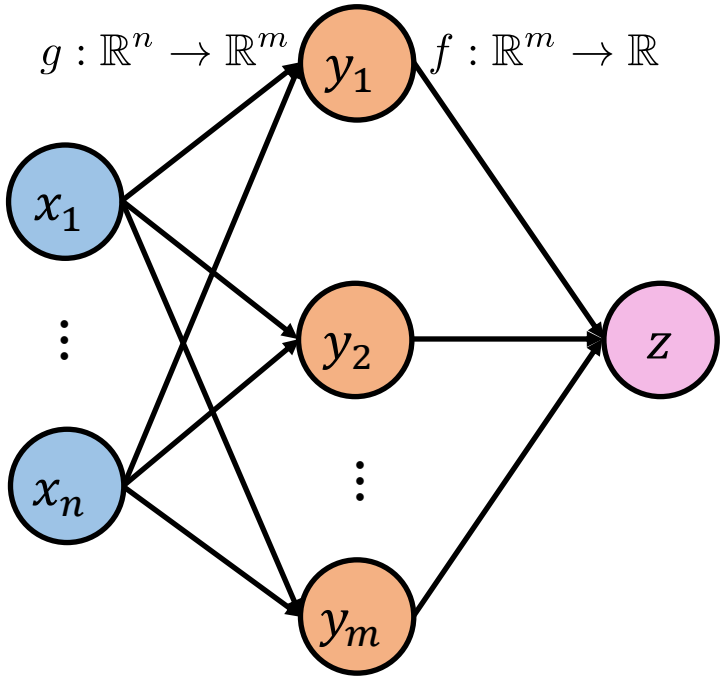
Backpropagation

Before we introduce backpropagation, let us review several concepts in vector calculus.

Chain Rule: the derivative of the composition of two differentiable functions in terms of the derivatives of individual functions

- Scalar-valued & multiple variables $f : \mathbb{R}^m \to \mathbb{R}$ $g : \mathbb{R}^n \to \mathbb{R}^m$

$$\frac{df(g(x))}{dx} = \mathbf{J}_g(x)^\top \nabla f(g(x)) = \frac{dz}{dx}$$



$$\frac{\partial z}{\partial x_1} = \sum_{i=1}^{m} \frac{\partial z}{\partial y_i} \frac{\partial y_i}{\partial x_1} = \begin{bmatrix} \frac{\partial z}{\partial y_1} \cdots \frac{\partial z}{\partial y_m} \end{bmatrix} \begin{bmatrix} \frac{\partial y_1}{\partial x_1} \\ \vdots \\ \frac{\partial y_m}{\partial x_1} \end{bmatrix}$$

Consider all possible paths from x_1 to z !

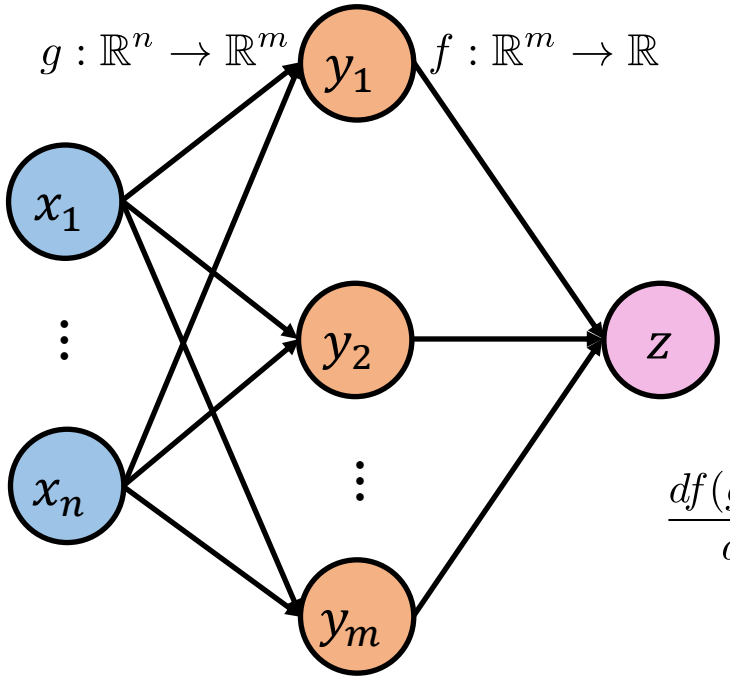
Backpropagation

Before we introduce backpropagation, let us review several concepts in vector calculus.

Chain Rule: the derivative of the composition of two differentiable functions in terms of the derivatives of individual functions

- Scalar-valued & multiple variables $f : \mathbb{R}^m \to \mathbb{R}$ $g : \mathbb{R}^n \to \mathbb{R}^m$

$$\frac{df(g(x))}{dx} = \mathbf{J}_g(x)^\top \nabla f(g(x)) = \frac{dz}{dx}$$



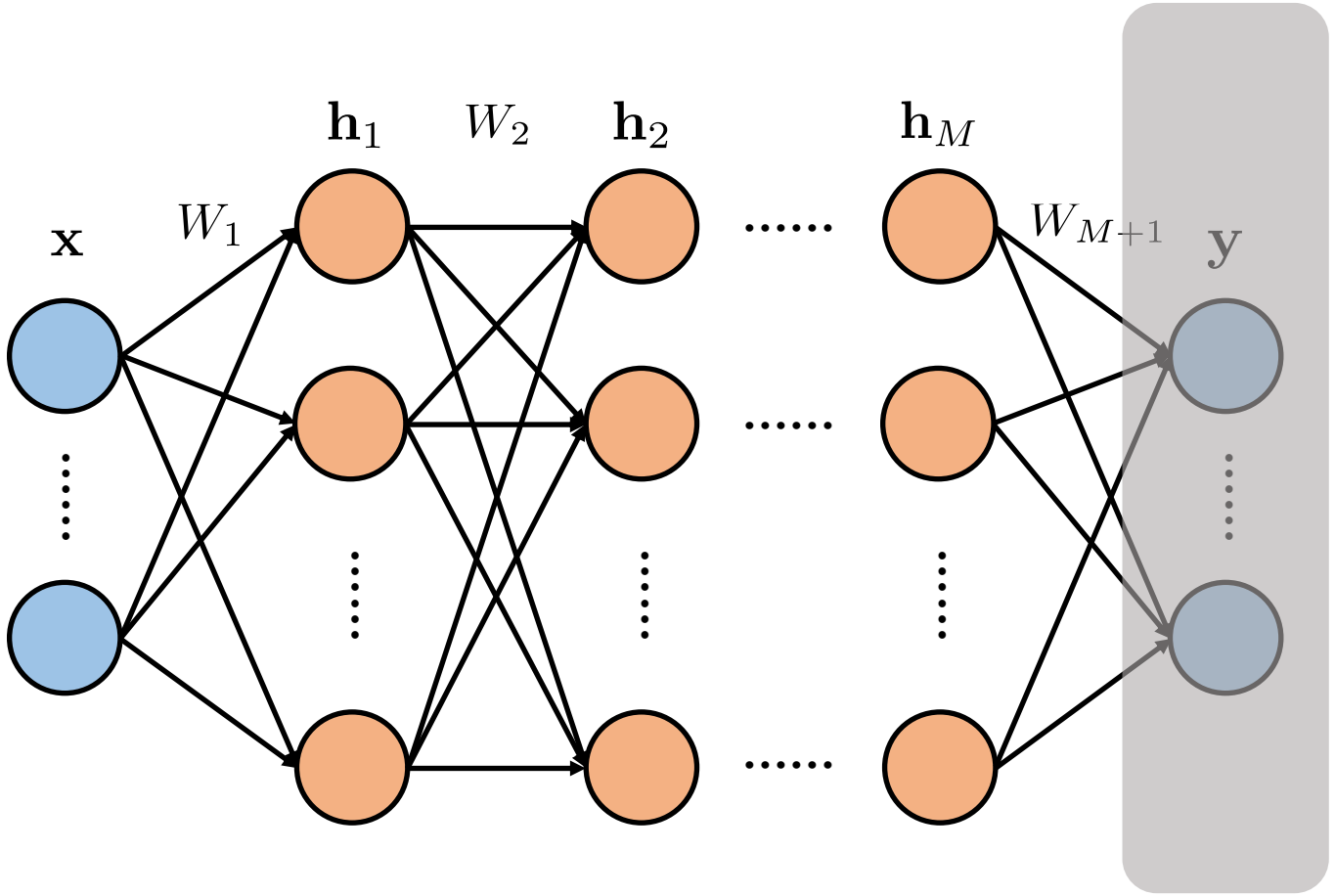
$$\frac{\partial z}{\partial x_1} = \sum_{i=1}^{m} \frac{\partial z}{\partial y_i}\frac{\partial y_i}{\partial x_1} = \begin{bmatrix} \frac{\partial z}{\partial y_1} \cdots \frac{\partial z}{\partial y_m} \end{bmatrix} \begin{bmatrix} \frac{\partial y_1}{\partial x_1} \\ \vdots \\ \frac{\partial y_m}{\partial x_1} \end{bmatrix}$$

$$\frac{df(g(x))}{dx} = \frac{dz}{dx} = \begin{bmatrix} \frac{\partial z}{\partial x_1} \\ \vdots \\ \frac{\partial z}{\partial x_n} \end{bmatrix} = \left(\begin{bmatrix} \frac{\partial z}{\partial y_1} \cdots \frac{\partial z}{\partial y_m} \end{bmatrix} \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_1}{\partial x_2} & \cdots & \frac{\partial y_1}{\partial x_n} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial y_m}{\partial x_1} & \frac{\partial y_m}{\partial x_2} & \cdots & \frac{\partial y_m}{\partial x_n} \end{bmatrix} \right)^\top = \mathbf{J}_g(x)^\top \nabla f(g(x))$$

Backpropagation

During the backward pass:

Loss: $L = \ell(\mathbf{y}, \bar{\mathbf{y}})$

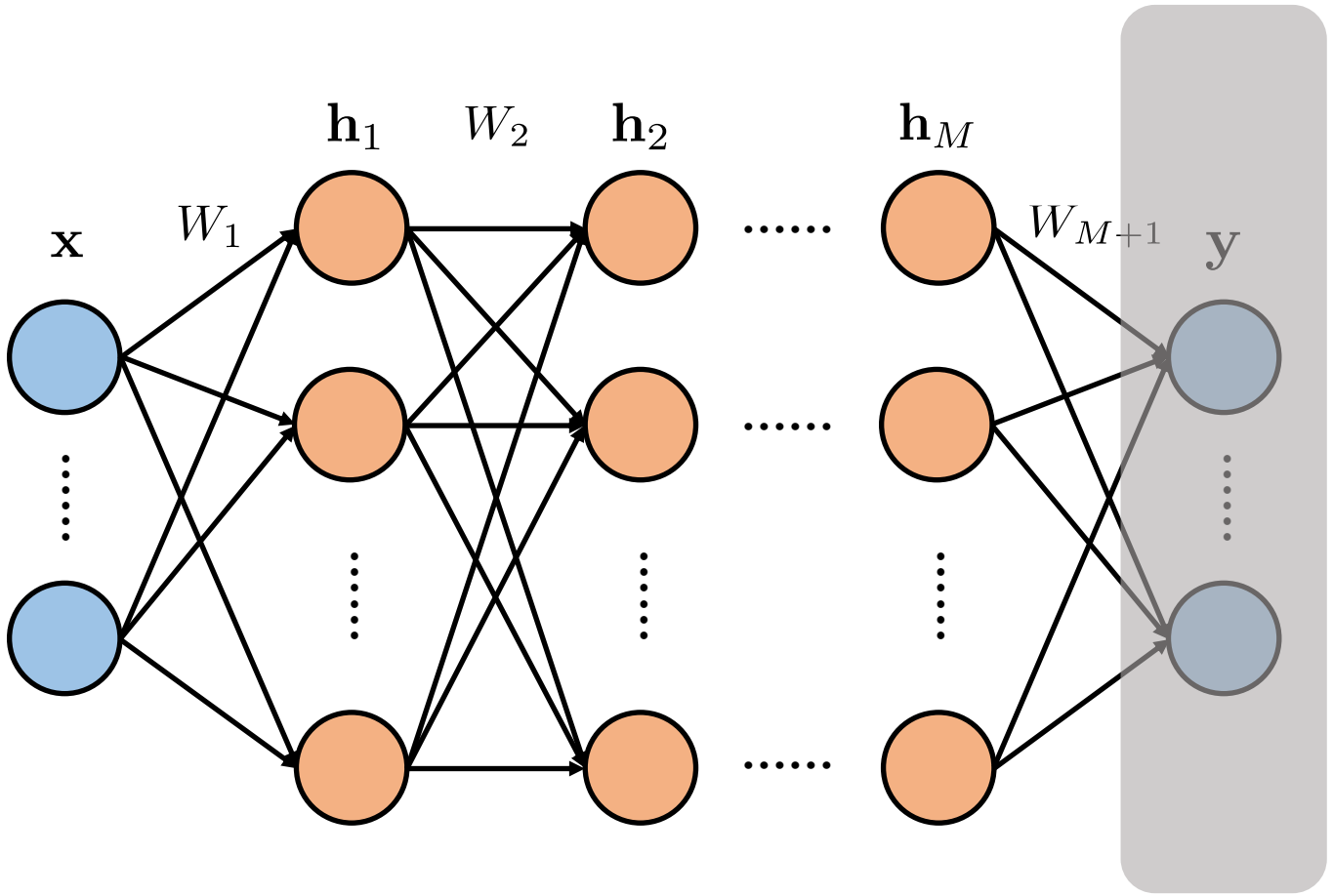


Backpropagation

During the backward pass:

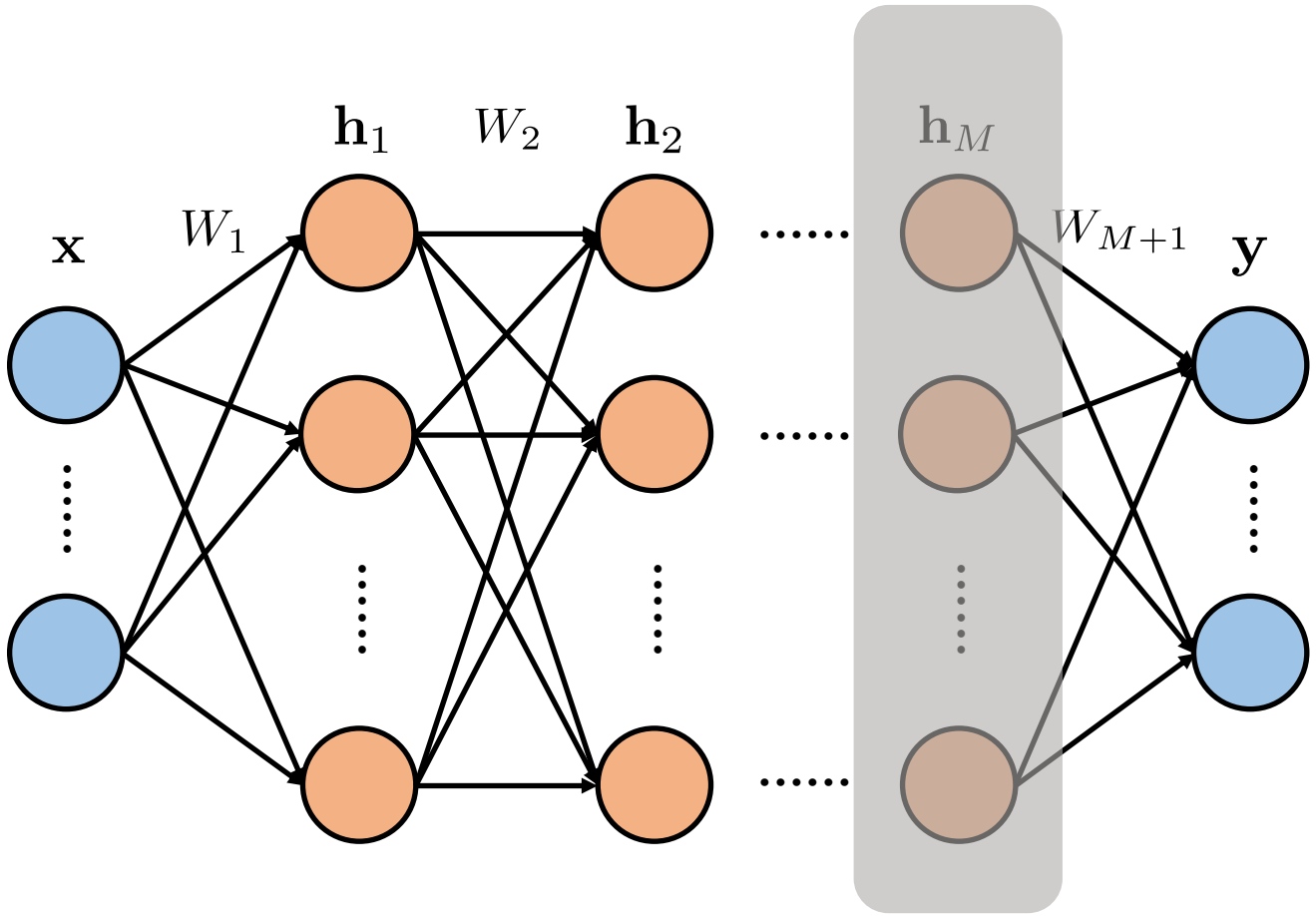
Loss: $L = \ell(\mathbf{y}, \bar{\mathbf{y}})$

Gradient of loss w.r.t. $\mathbf{y}$: $\frac{\partial L}{\partial \mathbf{y}}$



Backpropagation

During the backward pass:



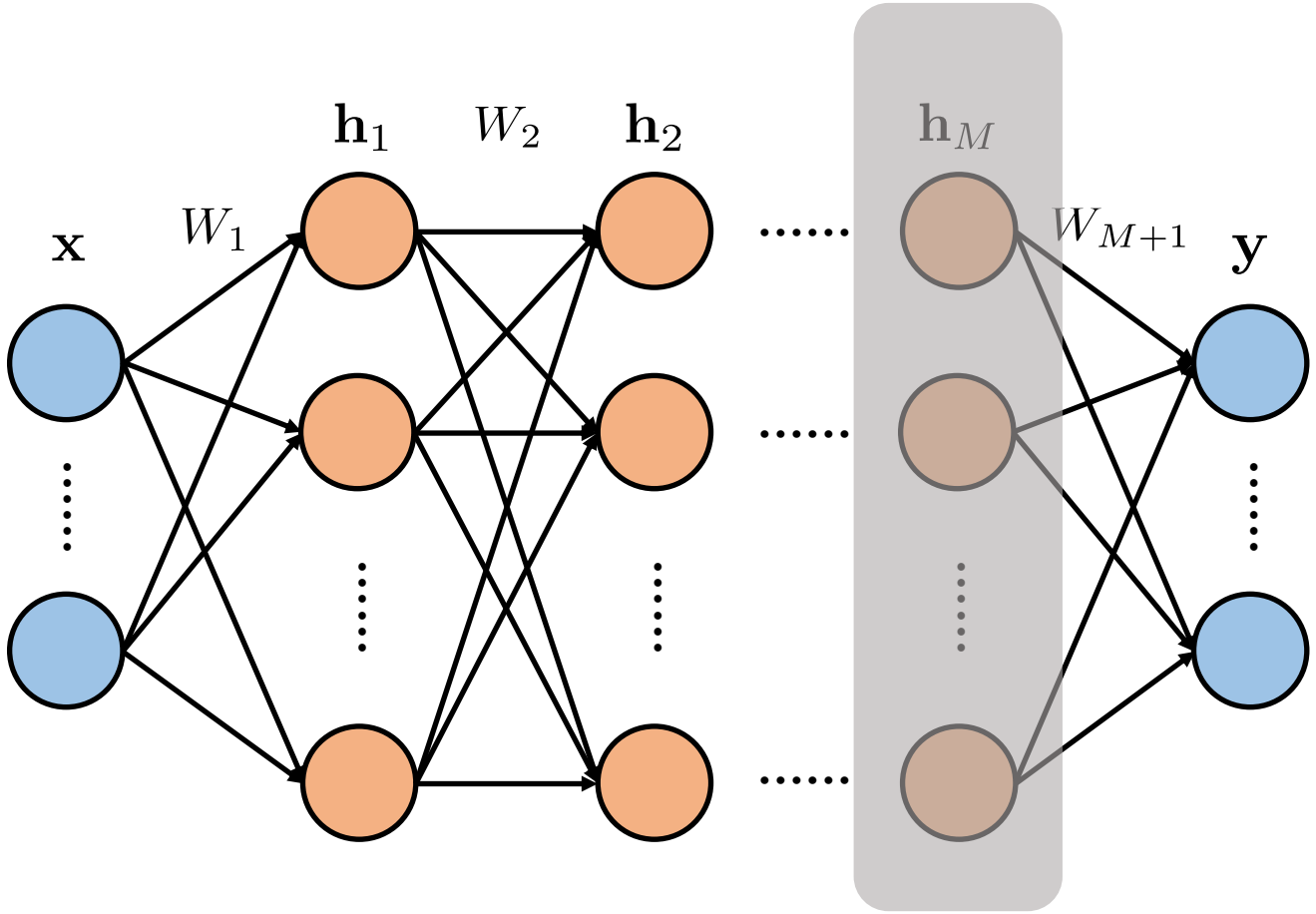
Loss: $L = \ell(\mathbf{y}, \bar{\mathbf{y}})$

Gradient of loss w.r.t. $\mathbf{y}$: $\frac{\partial L}{\partial \mathbf{y}}$

Gradient of loss w.r.t. $\mathbf{h}_M$:

Backpropagation

During the backward pass:



Loss: $L = \ell(\mathbf{y}, \bar{\mathbf{y}})$

Gradient of loss w.r.t. $\mathbf{y}$: $\frac{\partial L}{\partial \mathbf{y}}$

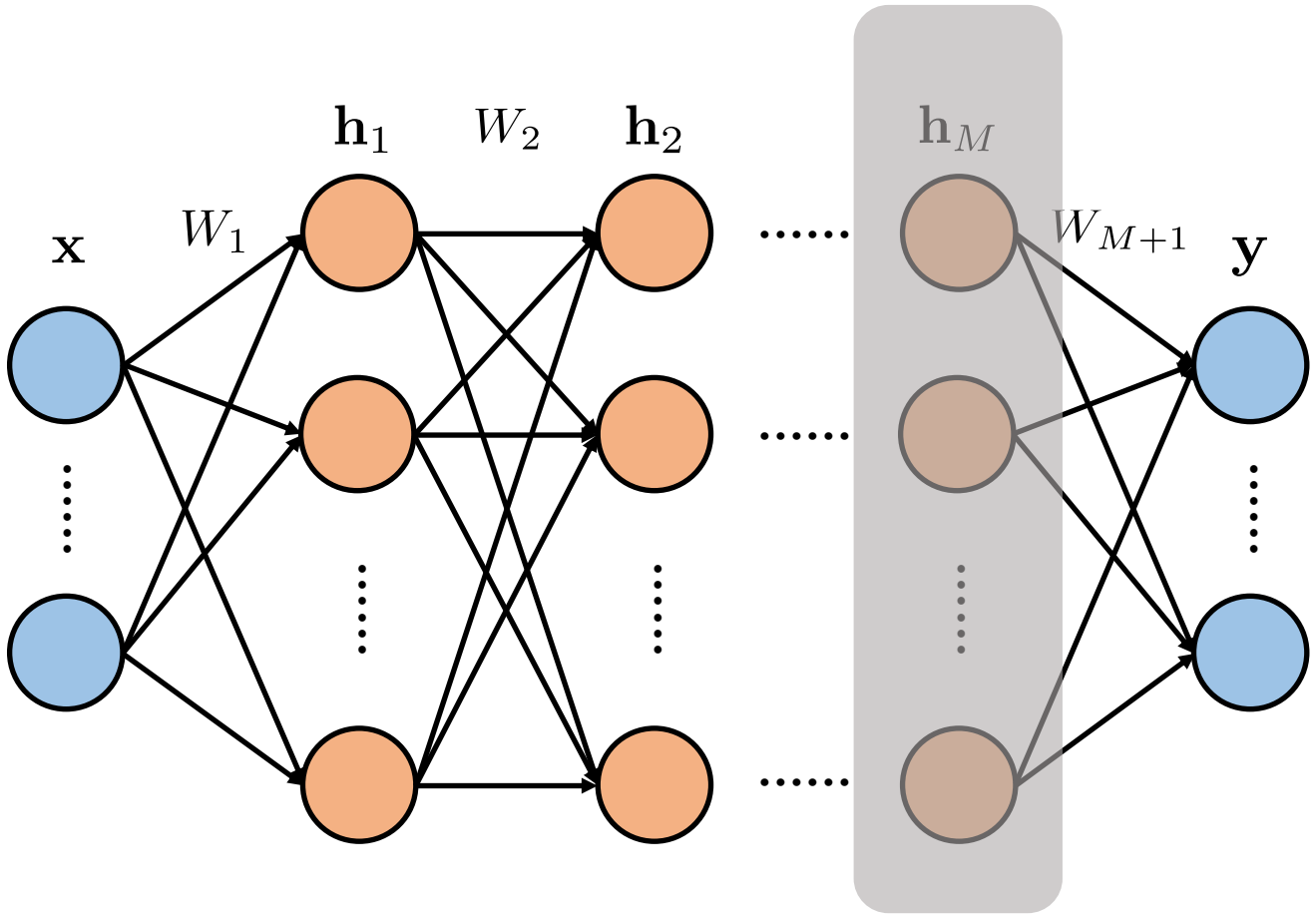
Gradient of loss w.r.t. $\mathbf{h}_M$:

Apply the chain rule we learned before:

$$\frac{\partial L}{\partial \mathbf{h}_M} = \left(\frac{\partial \mathbf{y}}{\partial \mathbf{h}_M}\right)^\top \frac{\partial \mathbf{L}}{\partial \mathbf{y}}$$

Backpropagation

During the backward pass:



Loss: $L = \ell(\mathbf{y}, \bar{\mathbf{y}})$

Gradient of loss w.r.t. $\mathbf{y}$: $\frac{\partial L}{\partial \mathbf{y}}$

Gradient of loss w.r.t. $\mathbf{h}_M$:

Apply the chain rule we learned before:

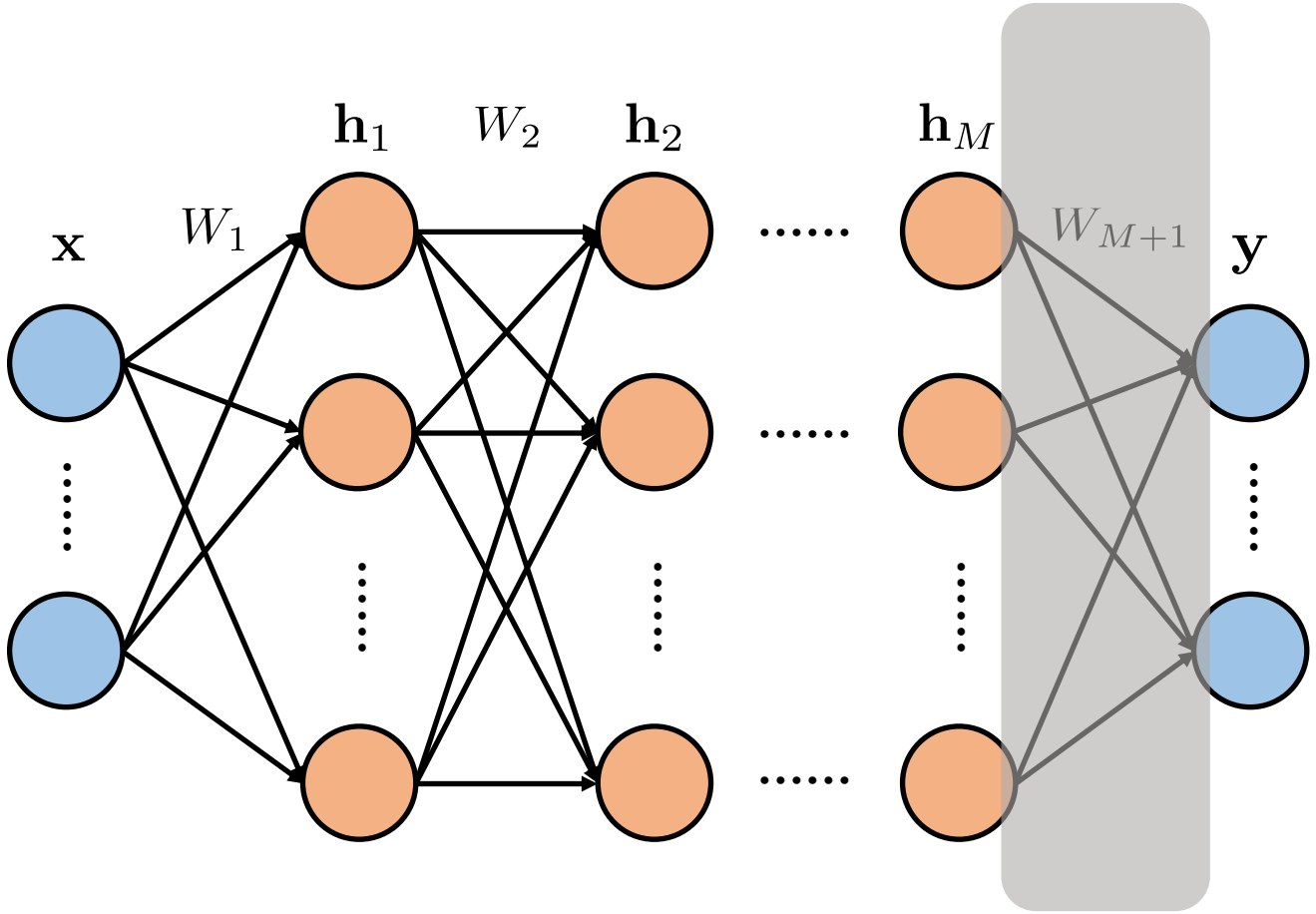
$$\frac{\partial L}{\partial \mathbf{h}_M} = \left(\frac{\partial \mathbf{y}}{\partial \mathbf{h}_M}\right)^\top \frac{\partial \mathbf{L}}{\partial \mathbf{y}}$$

Recall $\mathbf{y} = W_{M+1}\mathbf{h}_M$

We have $\frac{\partial \mathbf{y}}{\partial \mathbf{h}_M} = W_{M+1}$

Backpropagation

During the backward pass:



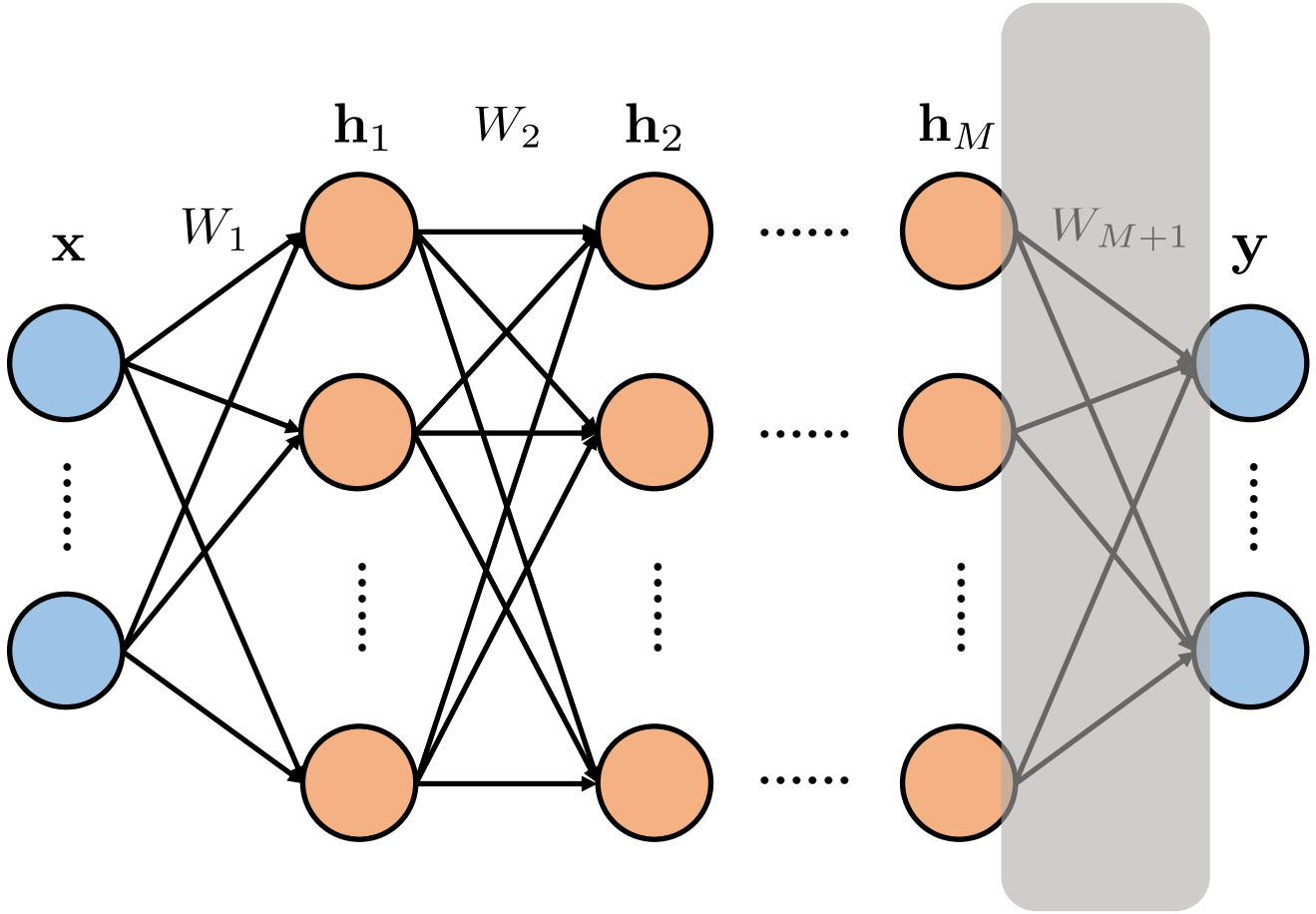
Loss: $L = \ell(\mathbf{y}, \bar{\mathbf{y}})$

Gradient of loss w.r.t. $\mathbf{y}$: $\frac{\partial L}{\partial \mathbf{y}}$

Gradient of loss w.r.t. W_{M+1} :

Backpropagation

During the backward pass:



Loss: $L = \ell(\mathbf{y}, \bar{\mathbf{y}})$

Gradient of loss w.r.t. $\mathbf{y}$: $\frac{\partial L}{\partial \mathbf{y}}$

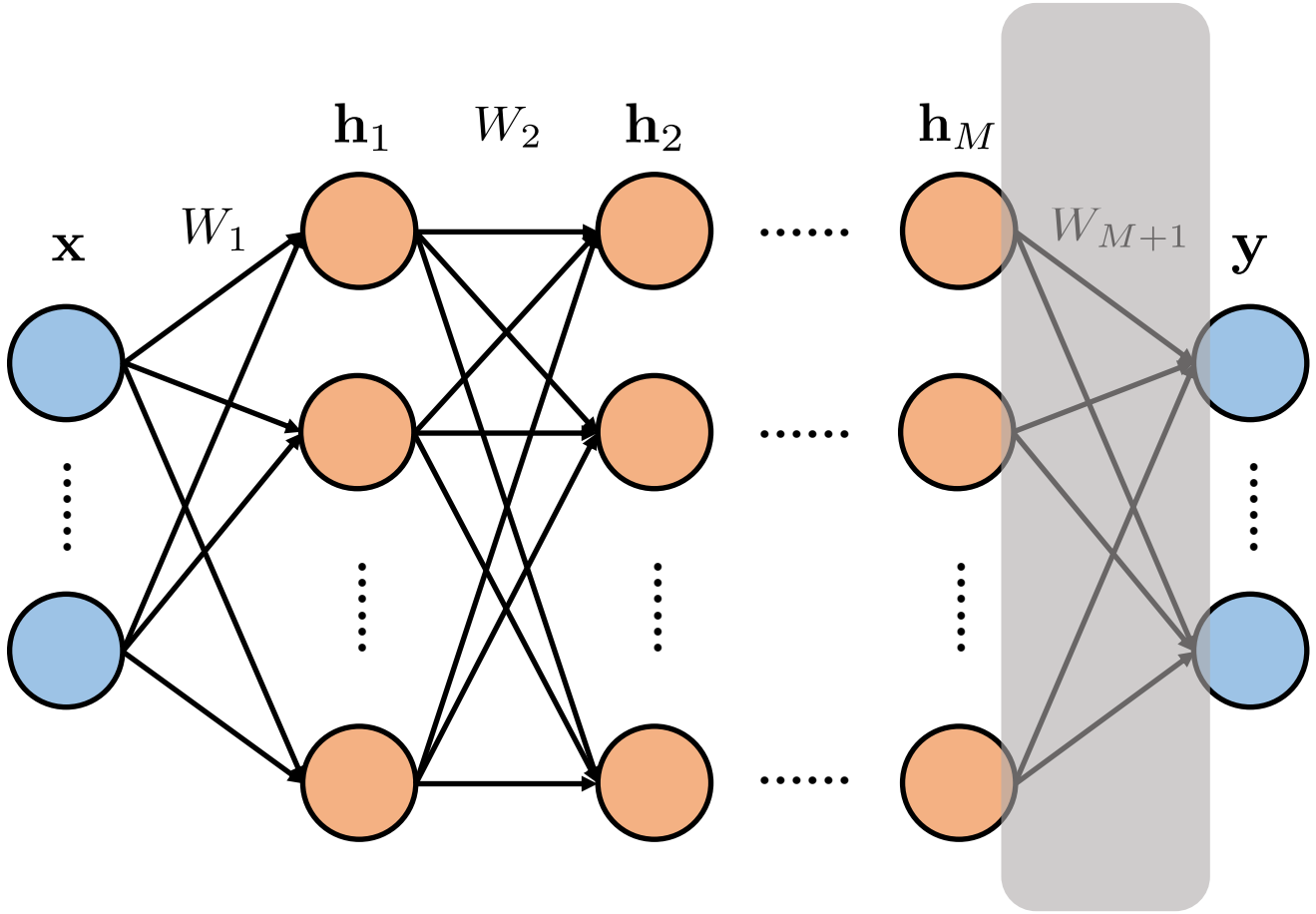
Gradient of loss w.r.t. W_{M+1} :

Ideally, chain rule should be something like

$$\frac{\partial L}{\partial W_{M+1}} = \frac{\partial L}{\partial \mathbf{y}} \frac{\partial \mathbf{y}}{\partial W_{M+1}}$$

Backpropagation

During the backward pass:



Loss: $L = \ell(\mathbf{y}, \bar{\mathbf{y}})$

Gradient of loss w.r.t. $\mathbf{y}$: $\frac{\partial L}{\partial \mathbf{y}}$

Gradient of loss w.r.t. W_{M+1} :

The chain rule suggests the shorthand

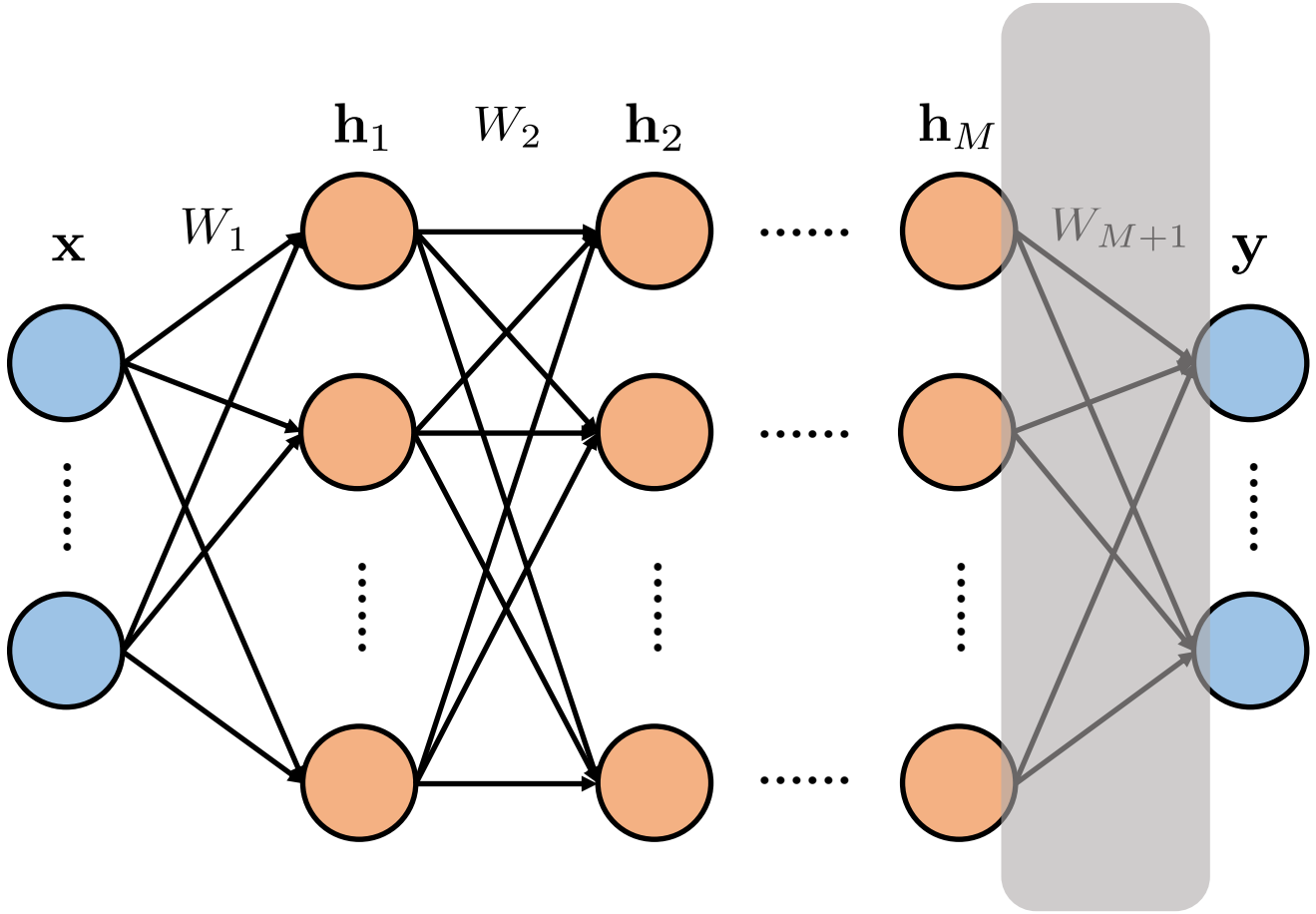
$$\frac{\partial L}{\partial W_{M+1}} = \frac{\partial L}{\partial \mathbf{y}} \frac{\partial \mathbf{y}}{\partial W_{M+1}}$$

$$\frac{\partial \mathbf{y}}{\partial W_{M+1}}$$

Use component derivatives to avoid a tensor derivative.

Backpropagation

During the backward pass:



Loss: $L = \ell(\mathbf{y}, \bar{\mathbf{y}})$

Gradient of loss w.r.t. $\mathbf{y}$: $\frac{\partial L}{\partial \mathbf{y}}$

Gradient of loss w.r.t. W_{M+1} :

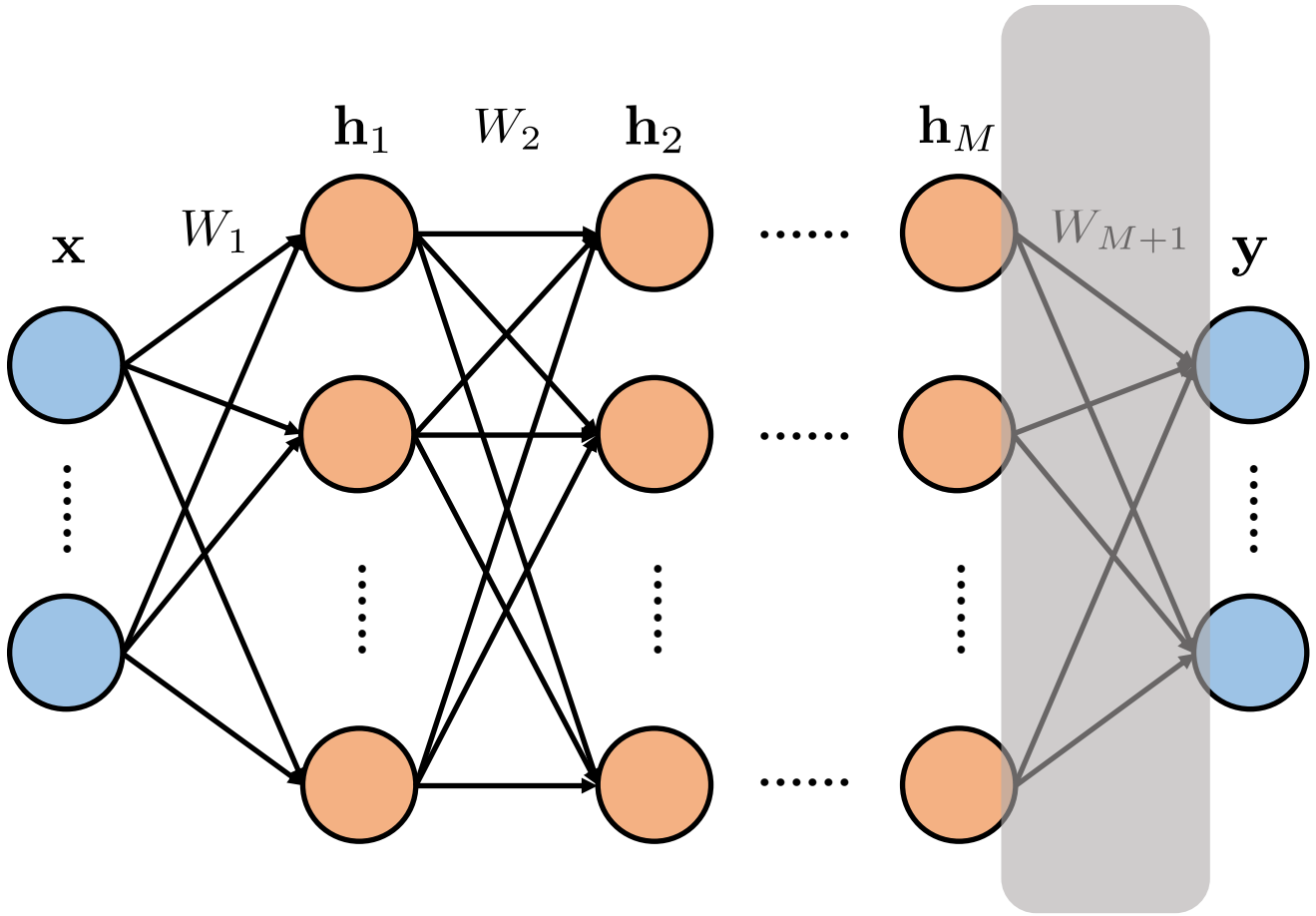
Note $\mathbf{y}[i]$ only depends on $W_{M+1}[i,:]$

does not depend on $W_{M+1}[j,:] \quad \forall j \neq i$

$$\mathbf{y}[i] = \sum_j W_{M+1}[i,j]\mathbf{h}[j]$$

Backpropagation

During the backward pass:



Loss: $L = \ell(\mathbf{y}, \bar{\mathbf{y}})$

Gradient of loss w.r.t. $\mathbf{y}$: $\frac{\partial L}{\partial \mathbf{y}}$

Gradient of loss w.r.t. W_{M+1} :

Note $\mathbf{y}[i]$ only depends on $W_{M+1}[i,:]$

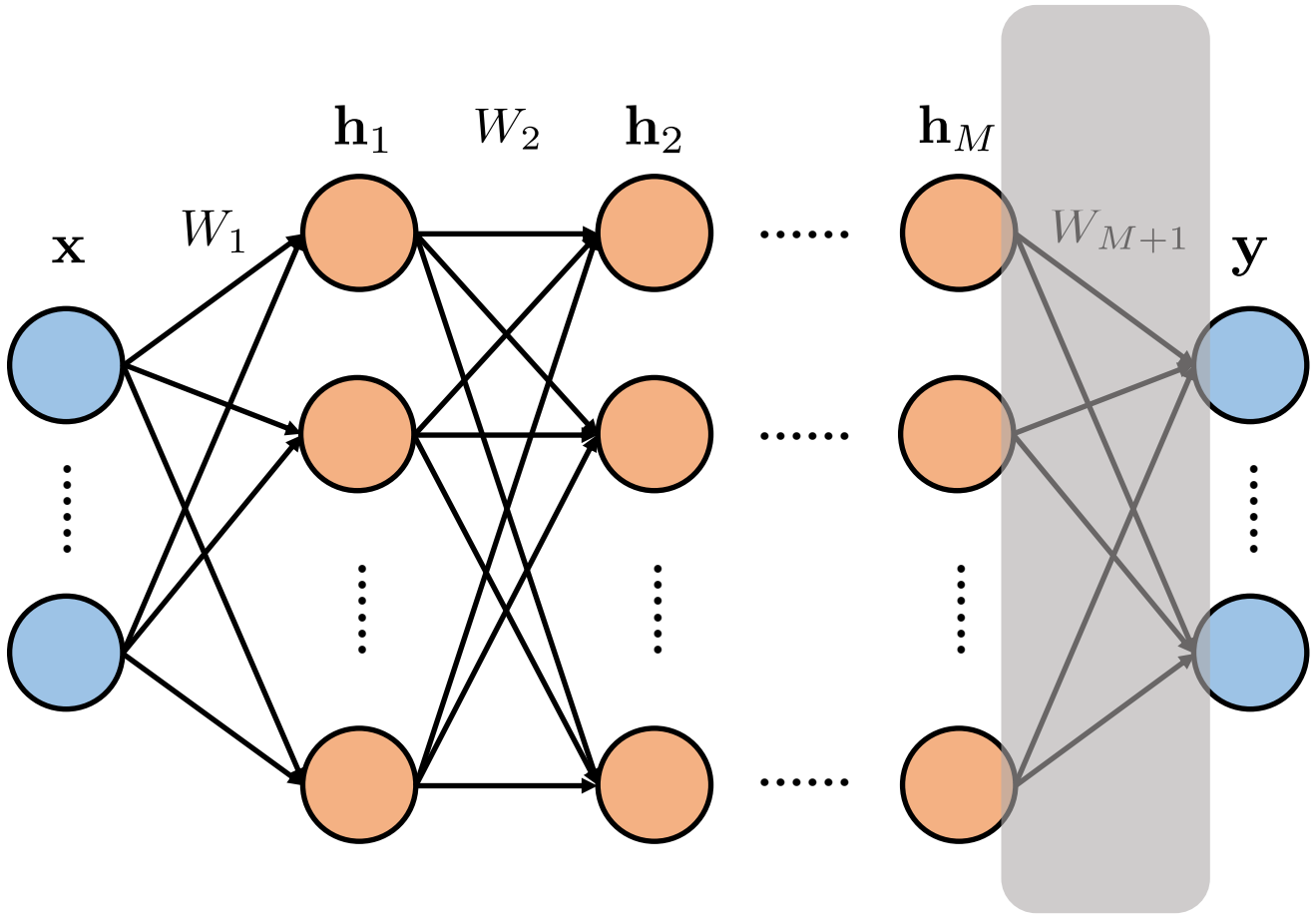
does not depend on $W_{M+1}[j,:] \quad \forall j \neq i$

$$\mathbf{y}[i] = \sum_j W_{M+1}[i,j]\mathbf{h}[j]$$

$$\frac{\partial \mathbf{y}[i]}{\partial W_{M+1}[i,j]} = \mathbf{h}[j]$$

Backpropagation

During the backward pass:



Loss: $L = \ell(\mathbf{y}, \bar{\mathbf{y}})$

Gradient of loss w.r.t. $\mathbf{y}$: $\frac{\partial L}{\partial \mathbf{y}}$

Gradient of loss w.r.t. W_{M+1} :

Note $\mathbf{y}[i]$ only depends on $W_{M+1}[i,:]$

does not depend on $W_{M+1}[j,:] \quad \forall j \neq i$

$$\mathbf{y}[i] = \sum_j W_{M+1}[i,j]\mathbf{h}[j]$$

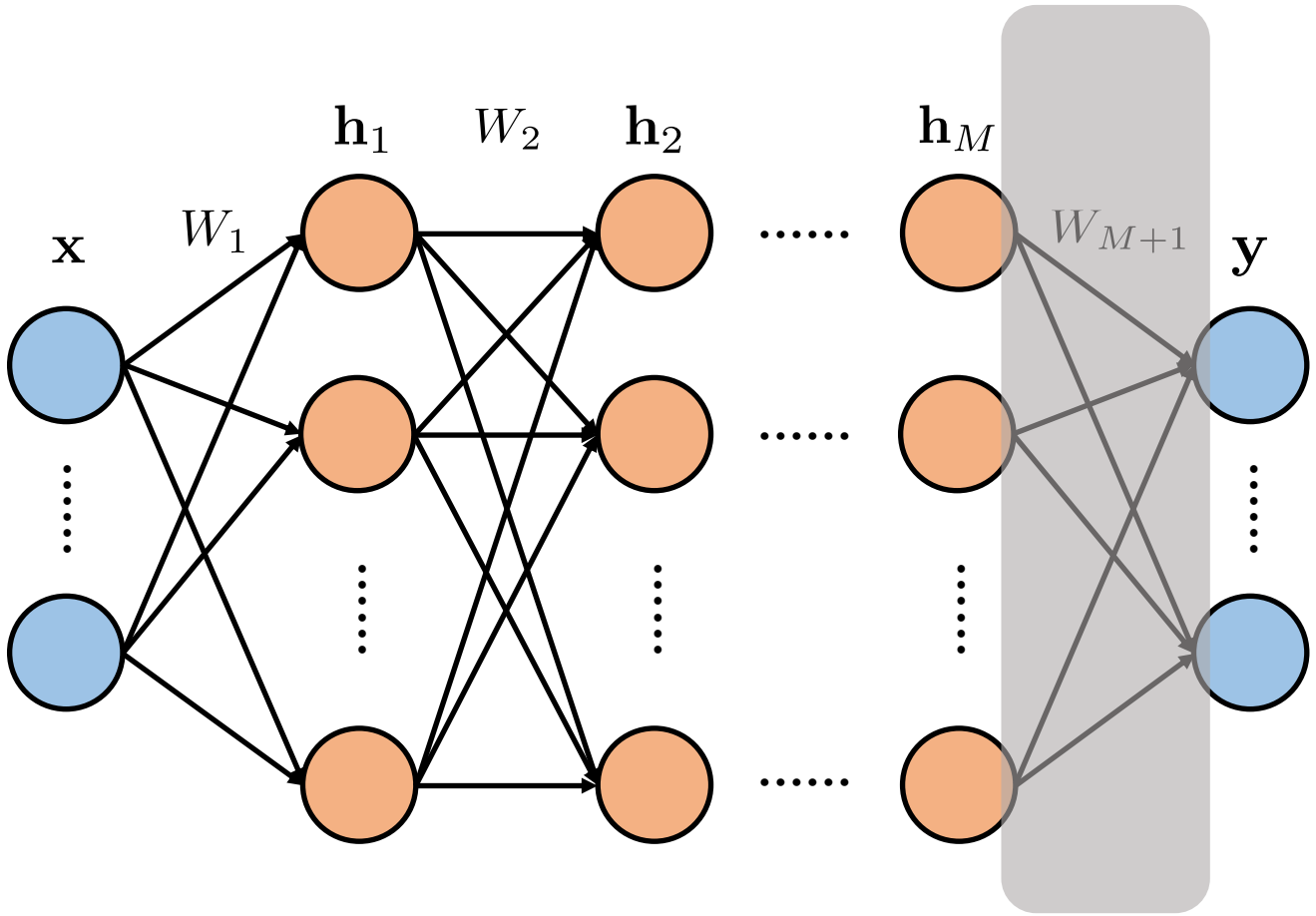
$$\frac{\partial \mathbf{y}[i]}{\partial W_{M+1}[i,j]} = \mathbf{h}[j]$$

We have

$$\frac{\partial L}{\partial W_{M+1}[i,j]} = \frac{\partial L}{\partial \mathbf{y}[i]} \frac{\partial \mathbf{y}[i]}{\partial W_{M+1}[i,j]} = \frac{\partial L}{\partial \mathbf{y}[i]} \mathbf{h}[j]$$

Backpropagation

During the backward pass:



Loss: $L = \ell(\mathbf{y}, \bar{\mathbf{y}})$

Gradient of loss w.r.t. $\mathbf{y}$: $\frac{\partial L}{\partial \mathbf{y}}$

Gradient of loss w.r.t. W_{M+1} :

Note $\mathbf{y}[i]$ only depends on $W_{M+1}[i,:]$

does not depend on $W_{M+1}[j,:] \quad \forall j \neq i$

$$\mathbf{y}[i] = \sum_j W_{M+1}[i,j]\mathbf{h}[j]$$

$$\frac{\partial \mathbf{y}[i]}{\partial W_{M+1}[i,j]} = \mathbf{h}[j]$$

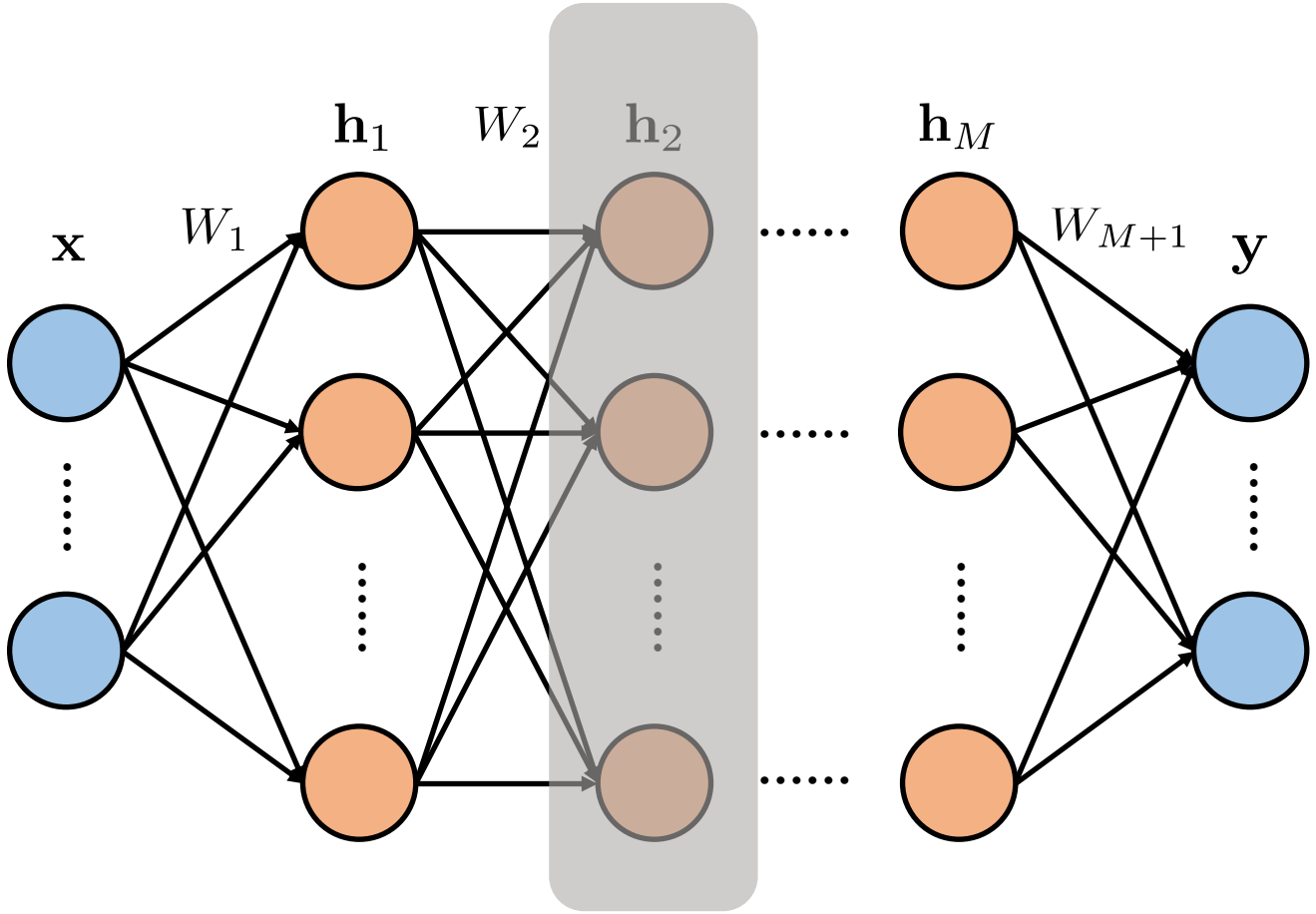
We have

$$\frac{\partial L}{\partial W_{M+1}[i,j]} = \frac{\partial L}{\partial \mathbf{y}[i]} \frac{\partial \mathbf{y}[i]}{\partial W_{M+1}[i,j]} = \frac{\partial L}{\partial \mathbf{y}[i]} \mathbf{h}[j]$$

$$\frac{\partial L}{\partial W_{M+1}} = \frac{\partial L}{\partial \mathbf{y}} \mathbf{h}^\top$$

Backpropagation

During the backward pass:



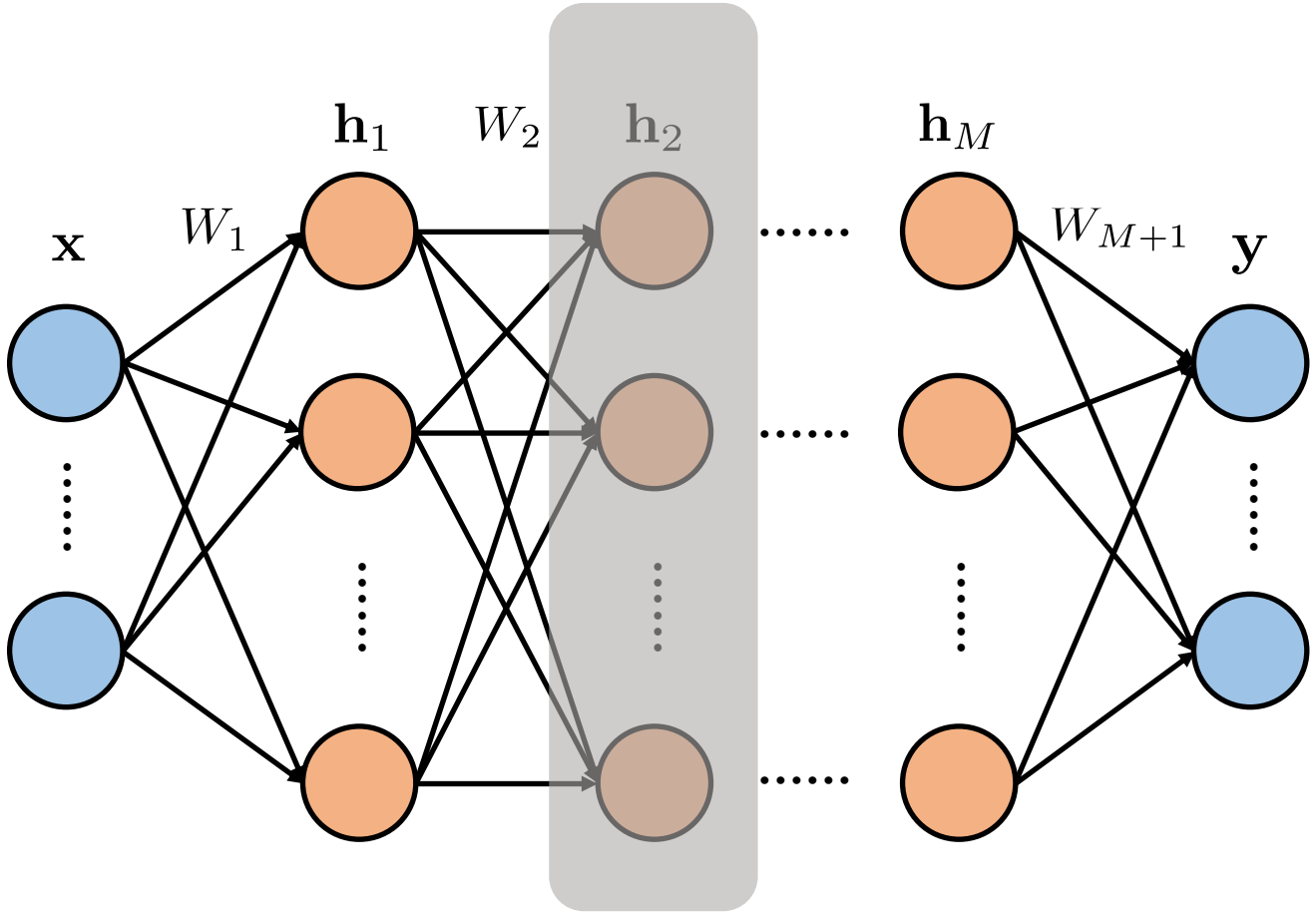
Loss: $L = \ell(\mathbf{y}, \bar{\mathbf{y}})$

Gradient of loss w.r.t. $\mathbf{y}$: $\frac{\partial L}{\partial \mathbf{y}}$

Gradient of loss w.r.t. $\mathbf{h}_2$:

Backpropagation

During the backward pass:



Loss: $L = \ell(\mathbf{y}, \bar{\mathbf{y}})$

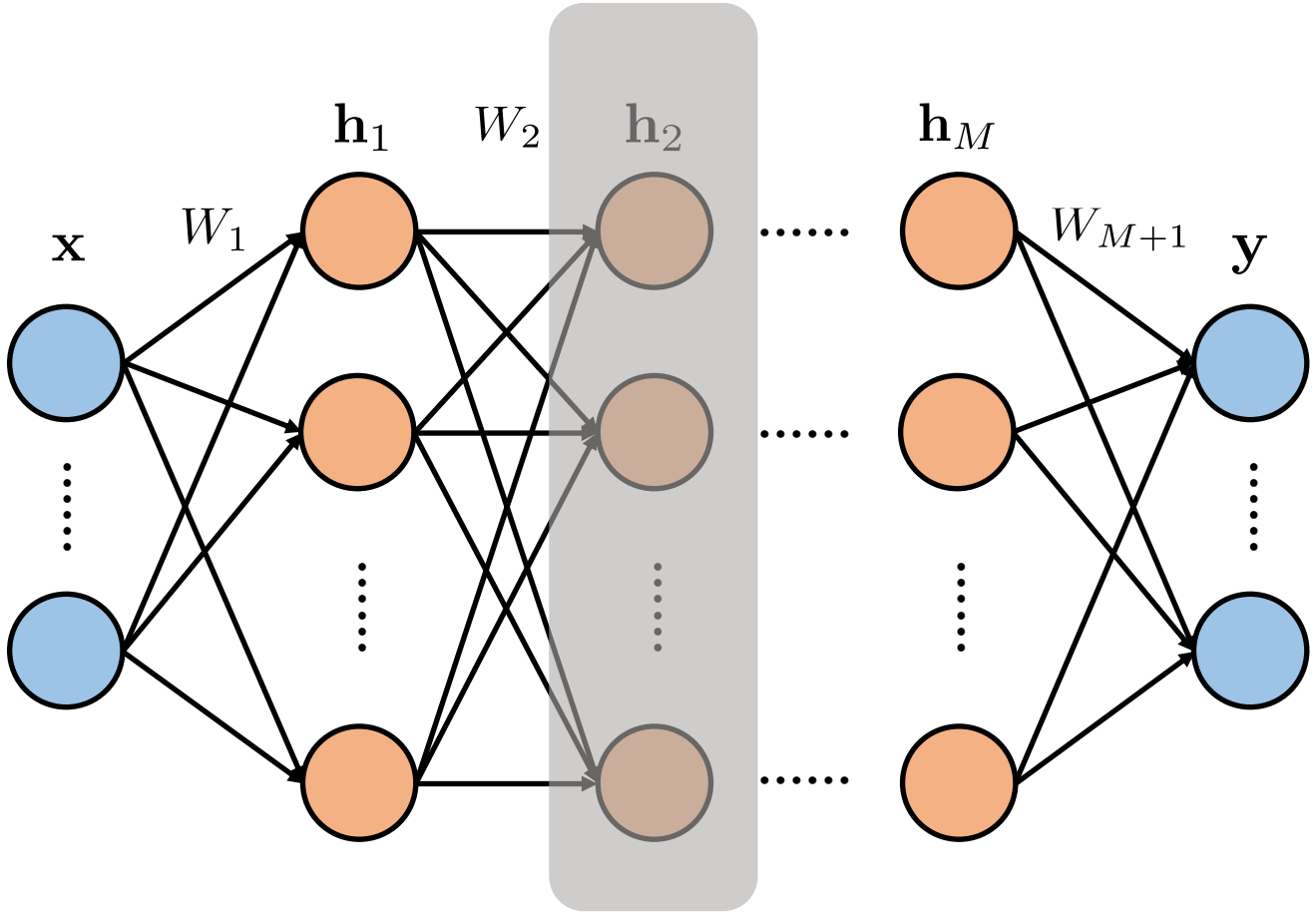
Gradient of loss w.r.t. $\mathbf{y}$: $\frac{\partial L}{\partial \mathbf{y}}$

Gradient of loss w.r.t. $\mathbf{h}_2$:
We know

$$\frac{\partial L}{\partial \mathbf{h}_M} = \left(\frac{\partial \mathbf{y}}{\partial \mathbf{h}_M}\right)^\top \frac{\partial \mathbf{L}}{\partial \mathbf{y}}$$

Backpropagation

During the backward pass:



Loss: $L = \ell(\mathbf{y}, \bar{\mathbf{y}})$

Gradient of loss w.r.t. $\mathbf{y}$: $\frac{\partial L}{\partial \mathbf{y}}$

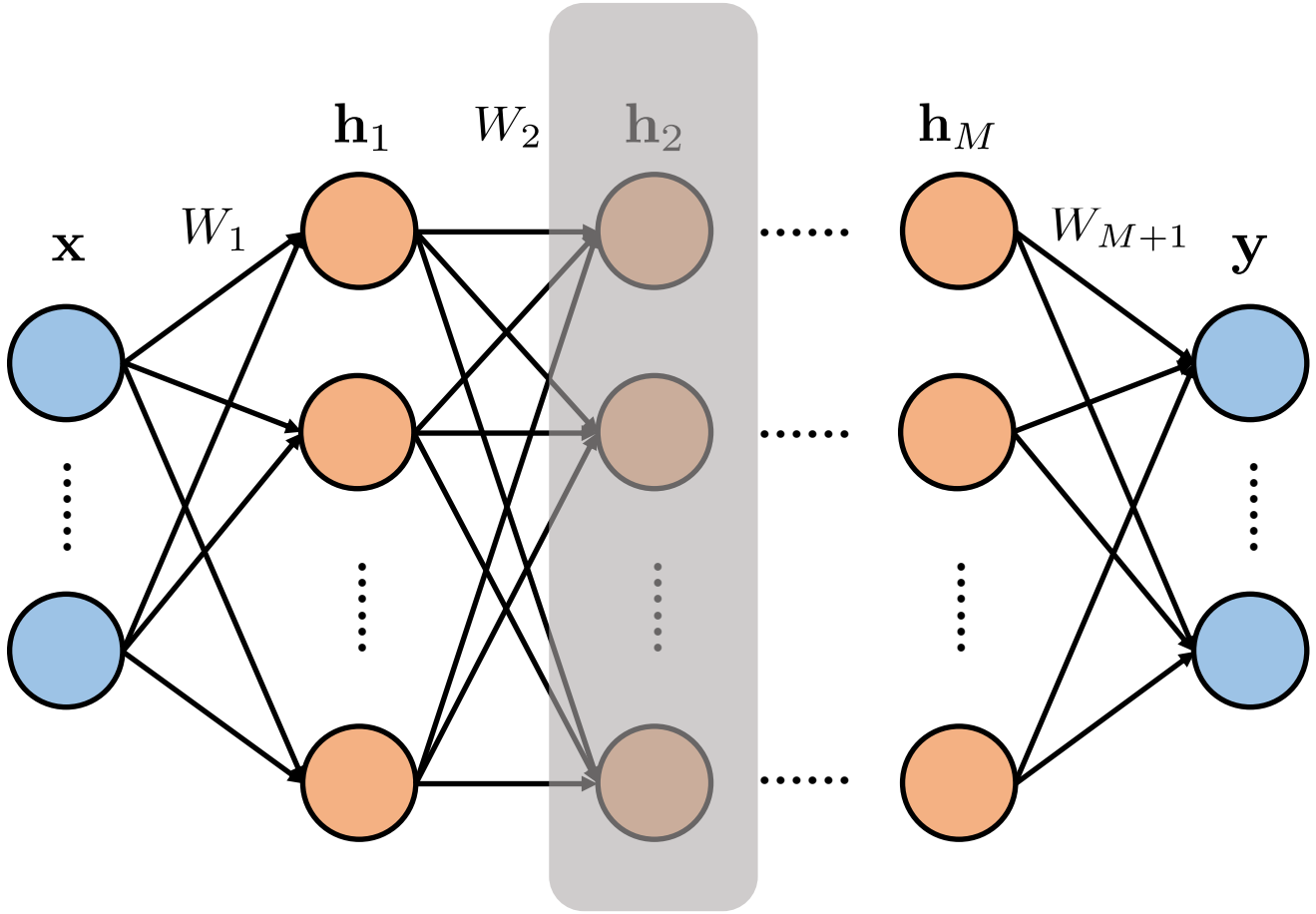
Gradient of loss w.r.t. $\mathbf{h}_2$:
We know

$$\frac{\partial L}{\partial \mathbf{h}_M} = \left(\frac{\partial \mathbf{y}}{\partial \mathbf{h}_M}\right)^\top \frac{\partial \mathbf{L}}{\partial \mathbf{y}}$$

$$\frac{\partial L}{\partial \mathbf{h}_{M-1}} = \left(\frac{\partial \mathbf{h}_M}{\partial \mathbf{h}_{M-1}}\right)^\top \frac{\partial L}{\partial \mathbf{h}_M}$$

Backpropagation

During the backward pass:



Loss: $L = \ell(\mathbf{y}, \bar{\mathbf{y}})$

Gradient of loss w.r.t. $\mathbf{y}$: $\frac{\partial L}{\partial \mathbf{y}}$

Gradient of loss w.r.t. $\mathbf{h}_2$:
We know

$$\frac{\partial L}{\partial \mathbf{h}_M} = \left(\frac{\partial \mathbf{y}}{\partial \mathbf{h}_M}\right)^\top \frac{\partial \mathbf{L}}{\partial \mathbf{y}}$$

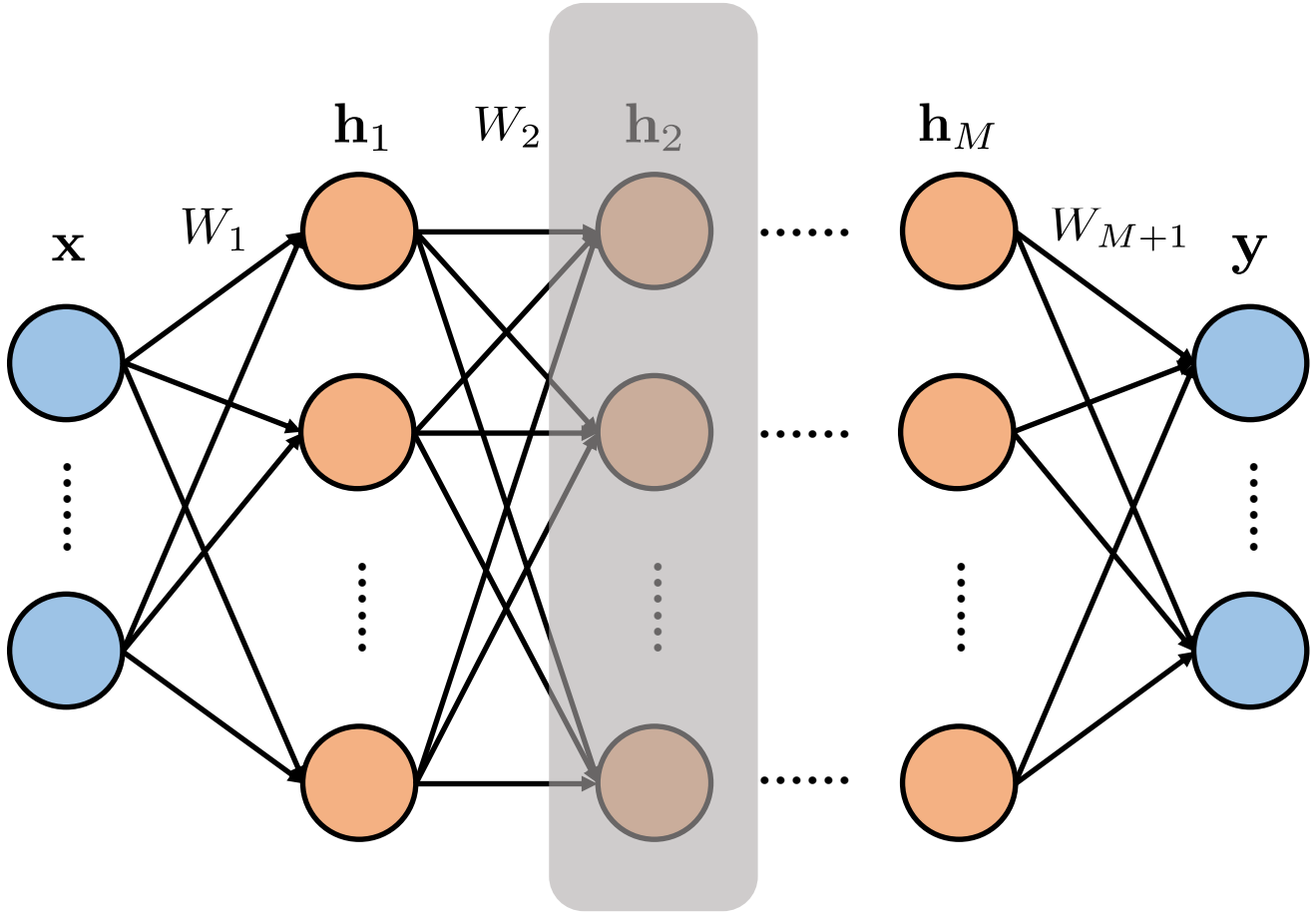
$$\frac{\partial L}{\partial \mathbf{h}_{M-1}} = \left(\frac{\partial \mathbf{h}_M}{\partial \mathbf{h}_{M-1}}\right)^\top \frac{\partial L}{\partial \mathbf{h}_M}$$

$$\vdots$$

$$\frac{\partial L}{\partial \mathbf{h}_2} = \left(\frac{\partial \mathbf{h}_3}{\partial \mathbf{h}_2}\right)^\top \cdots \left(\frac{\partial \mathbf{h}_M}{\partial \mathbf{h}_{M-1}}\right)^\top \left(\frac{\partial \mathbf{y}}{\partial \mathbf{h}_M}\right)^\top \frac{\partial L}{\partial \mathbf{y}}$$

Backpropagation

During the backward pass:



Loss: $L = \ell(\mathbf{y}, \bar{\mathbf{y}})$

Gradient of loss w.r.t. $\mathbf{y}$: $\frac{\partial L}{\partial \mathbf{y}}$

Gradient of loss w.r.t. $\mathbf{h}_2$:
We know

$$\frac{\partial L}{\partial \mathbf{h}_M} = \left(\frac{\partial \mathbf{y}}{\partial \mathbf{h}_M}\right)^\top \frac{\partial \mathbf{L}}{\partial \mathbf{y}}$$

$$\frac{\partial L}{\partial \mathbf{h}_{M-1}} = \left(\frac{\partial \mathbf{h}_M}{\partial \mathbf{h}_{M-1}}\right)^\top \frac{\partial L}{\partial \mathbf{h}_M}$$

$$\vdots$$

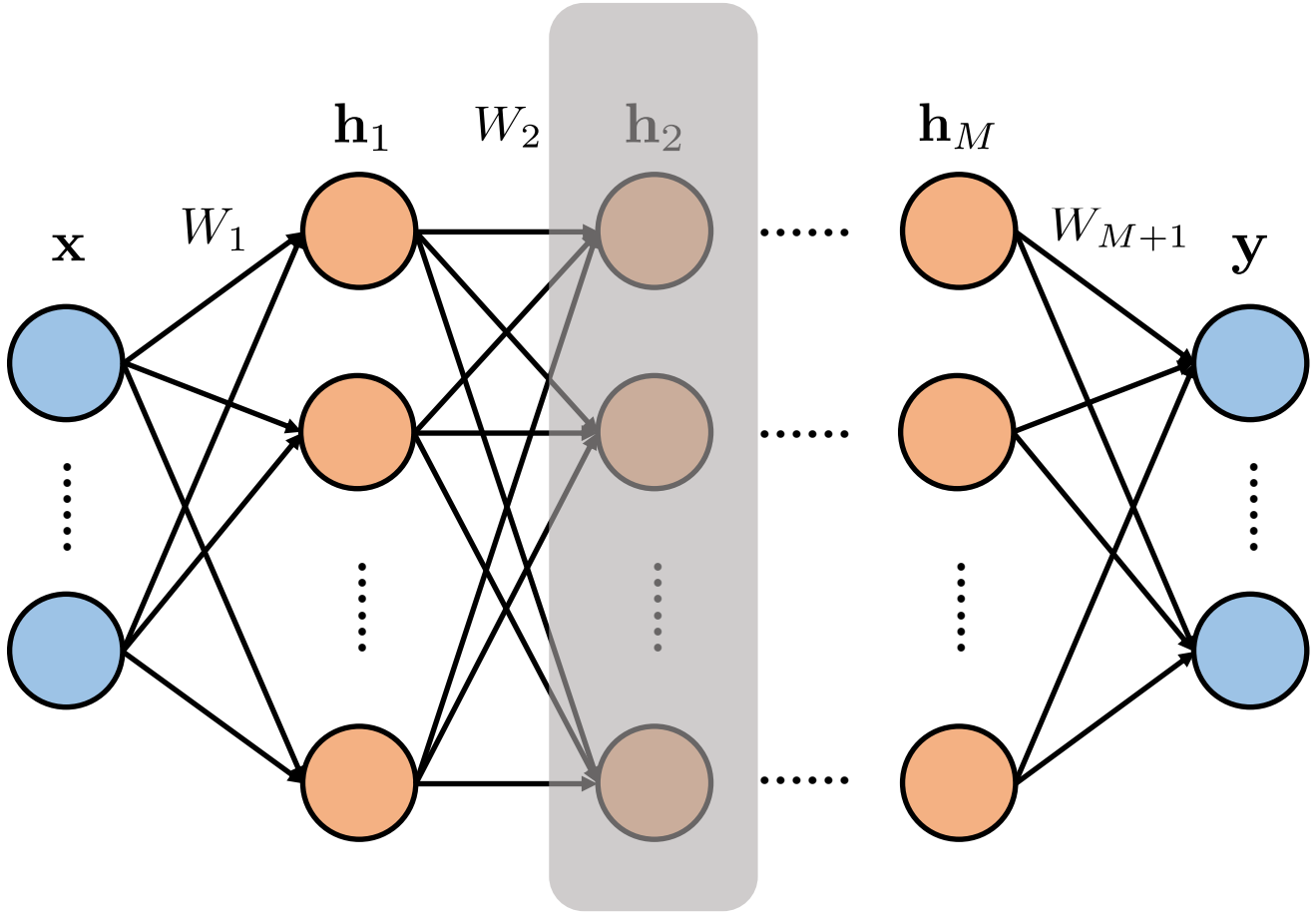
$$\frac{\partial L}{\partial \mathbf{h}_2} = \left(\frac{\partial \mathbf{h}_3}{\partial \mathbf{h}_2}\right)^\top \cdots \left(\frac{\partial \mathbf{h}_M}{\partial \mathbf{h}_{M-1}}\right)^\top \left(\frac{\partial \mathbf{y}}{\partial \mathbf{h}_M}\right)^\top \frac{\partial L}{\partial \mathbf{y}}$$

General form:

$$\frac{\partial L}{\partial \mathbf{h}_i} = \mathbf{J}_{i+1}^\top \cdots \mathbf{J}_M^\top \frac{\partial L}{\partial \mathbf{y}} \quad \text{where} \quad \mathbf{J}_{i+1} \equiv \frac{\partial \mathbf{h}_{i+1}}{\partial \mathbf{h}_i}$$
$$\mathbf{h}_{M+1} \equiv \mathbf{y}$$

Backpropagation

During the backward pass:



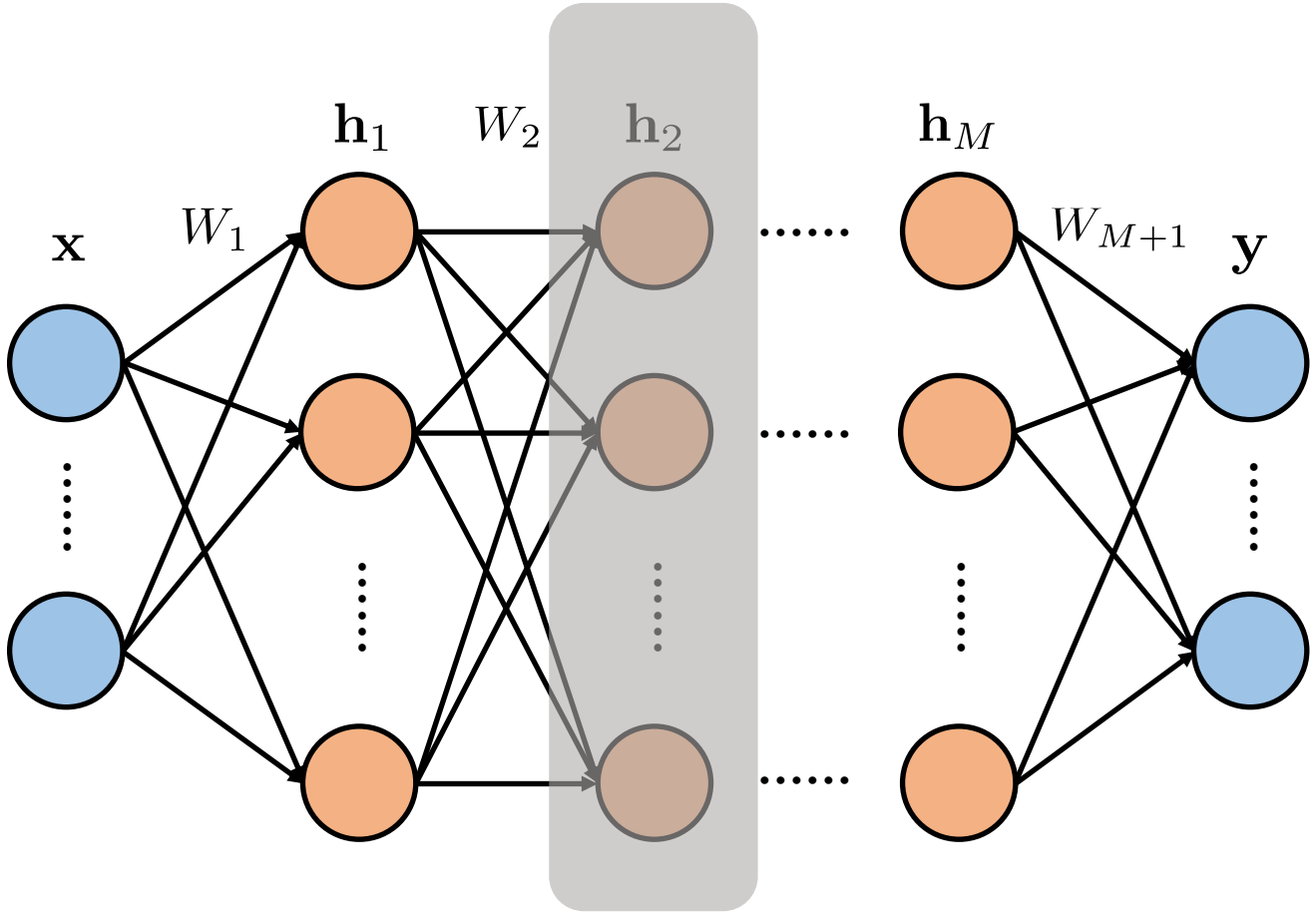
Loss: $L = \ell(\mathbf{y}, \bar{\mathbf{y}})$

Gradient of loss w.r.t. $\mathbf{y}$: $\frac{\partial L}{\partial \mathbf{y}}$

What is $\mathbf{J}_2 \equiv \frac{\partial \mathbf{h}_2}{\partial \mathbf{h}_1}$?

Backpropagation

During the backward pass:



Loss: $L = \ell(\mathbf{y}, \bar{\mathbf{y}})$

Gradient of loss w.r.t. $\mathbf{y}$: $\frac{\partial L}{\partial \mathbf{y}}$

What is $\mathbf{J}_2 \equiv \frac{\partial \mathbf{h}_2}{\partial \mathbf{h}_1}$?

Recall $\mathbf{h}_2 = \sigma(W_2 \mathbf{h}_1)$

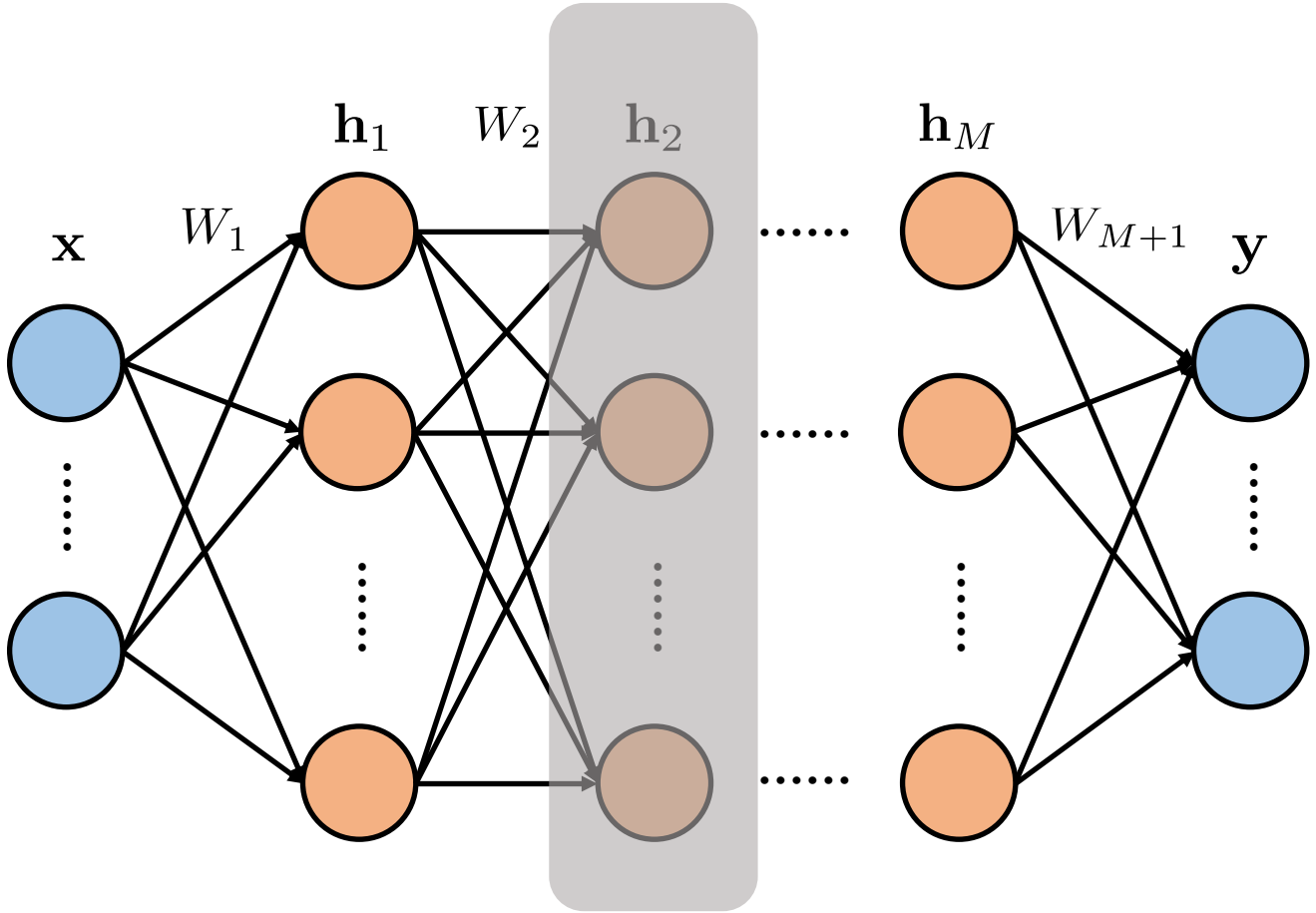
Denoting $\mathbf{z}_2 = W_2 \mathbf{h}_1$

We have

$$\frac{\partial \mathbf{h}_2}{\partial \mathbf{h}_1} = \frac{\partial \mathbf{h}_2}{\partial \mathbf{z}_2} \frac{\partial \mathbf{z}_2}{\partial \mathbf{h}_1}$$

Backpropagation

During the backward pass:



Loss: $L = \ell(\mathbf{y}, \bar{\mathbf{y}})$

Gradient of loss w.r.t. $\mathbf{y}$: $\frac{\partial L}{\partial \mathbf{y}}$

What is $\mathbf{J}_2 \equiv \frac{\partial \mathbf{h}_2}{\partial \mathbf{h}_1}$?

Recall $\mathbf{h}_2 = \sigma(W_2 \mathbf{h}_1)$

Denoting $\mathbf{z}_2 = W_2 \mathbf{h}_1$

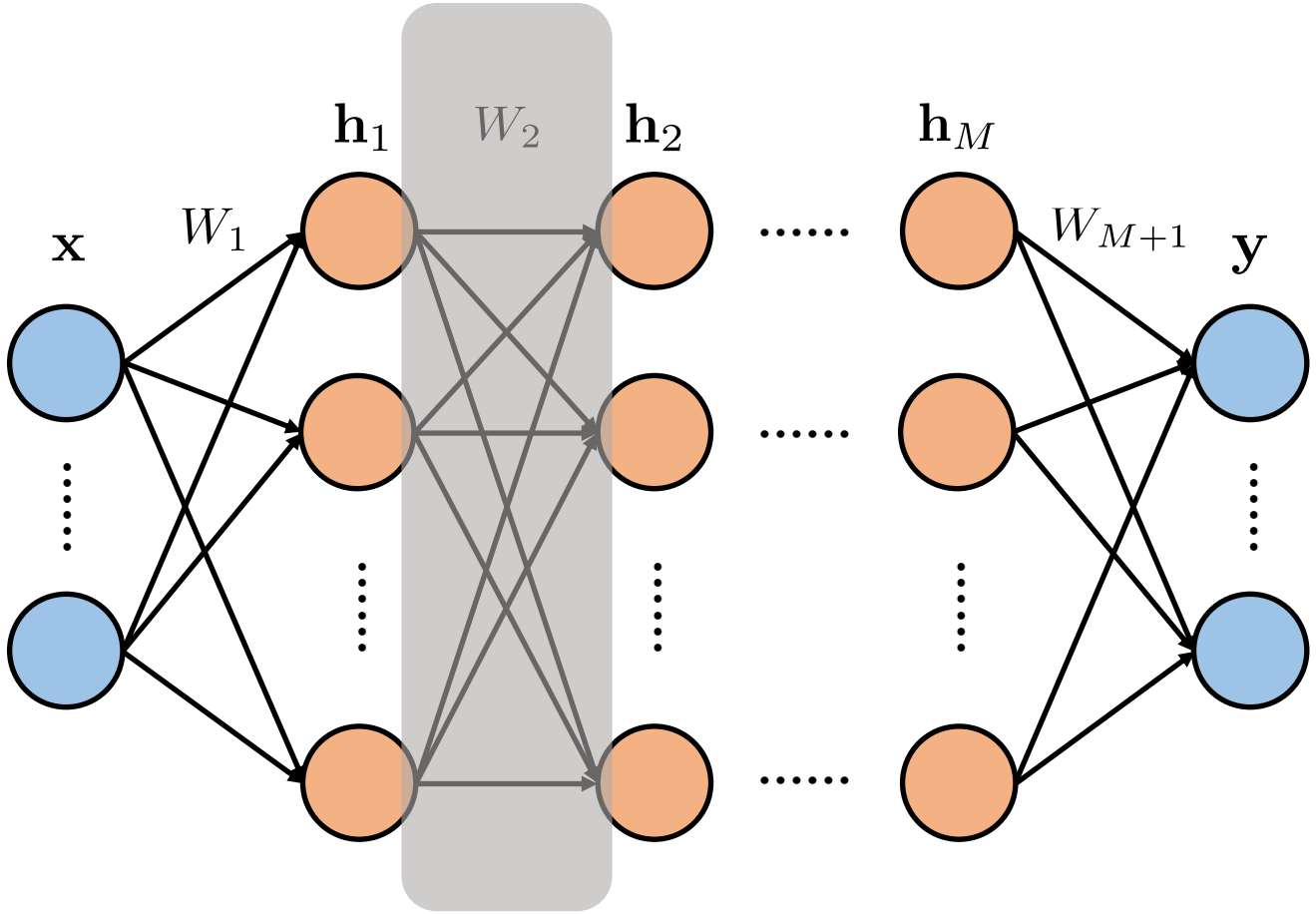
We have

$$\frac{\partial \mathbf{h}_2}{\partial \mathbf{h}_1} = \frac{\partial \mathbf{h}_2}{\partial \mathbf{z}_2} \frac{\partial \mathbf{z}_2}{\partial \mathbf{h}_1} = \mathrm{diag}(\sigma'(\mathbf{z}_2)) W_2$$

The Jacobian of element-wise nonlinearity is a diagonal matrix!

Backpropagation

During the backward pass:



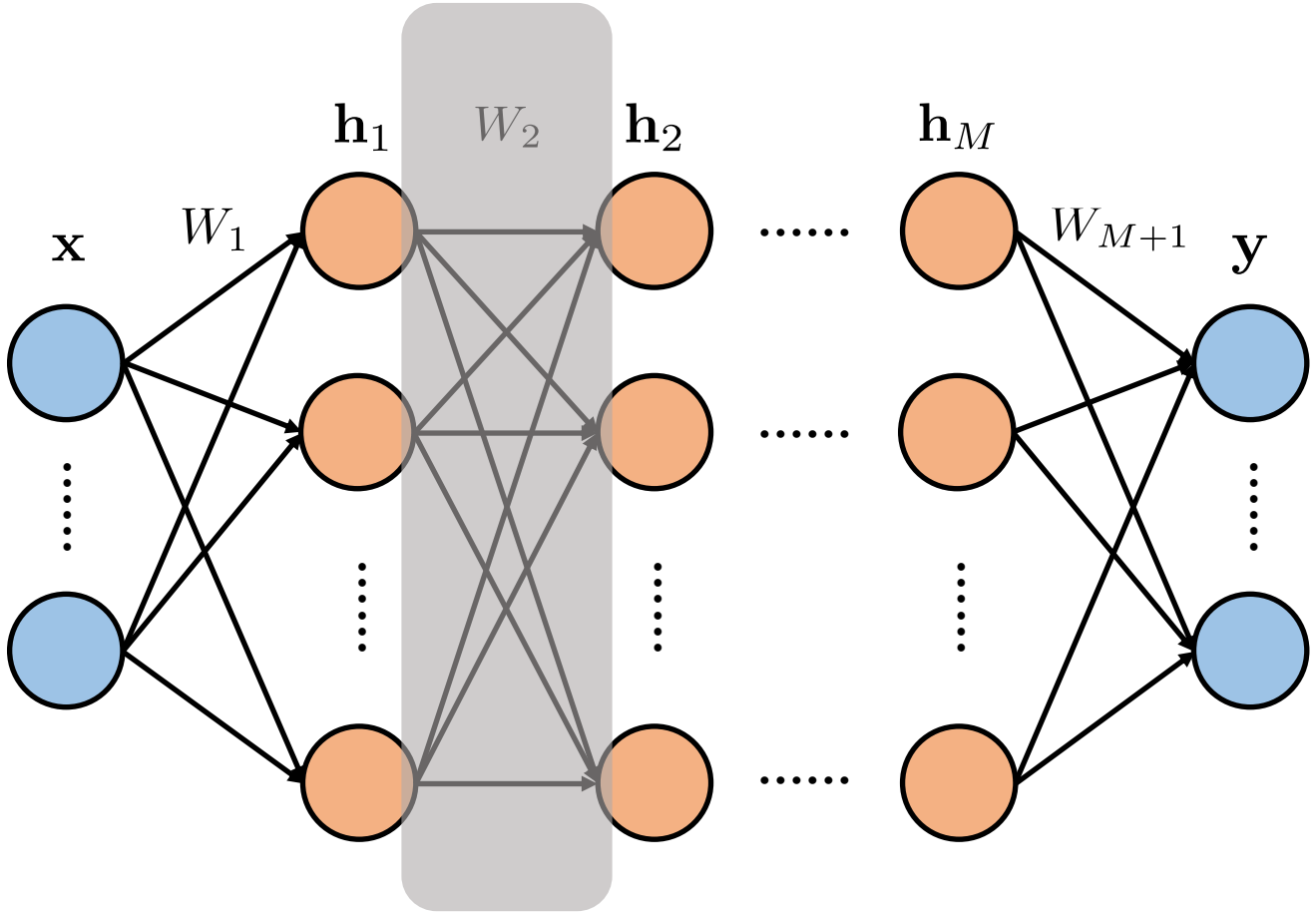
Loss: $L = \ell(\mathbf{y}, \bar{\mathbf{y}})$

Gradient of loss w.r.t. $\mathbf{y}$: $\frac{\partial L}{\partial \mathbf{y}}$

Gradient of loss w.r.t. W_2 :

Backpropagation

During the backward pass:



Loss: $L = \ell(\mathbf{y}, \bar{\mathbf{y}})$

Gradient of loss w.r.t. $\mathbf{y}$: $\frac{\partial L}{\partial \mathbf{y}}$

Gradient of loss w.r.t. W_2 :

Recall $\mathbf{h}_2 = \sigma(W_2 \mathbf{h}_1)$

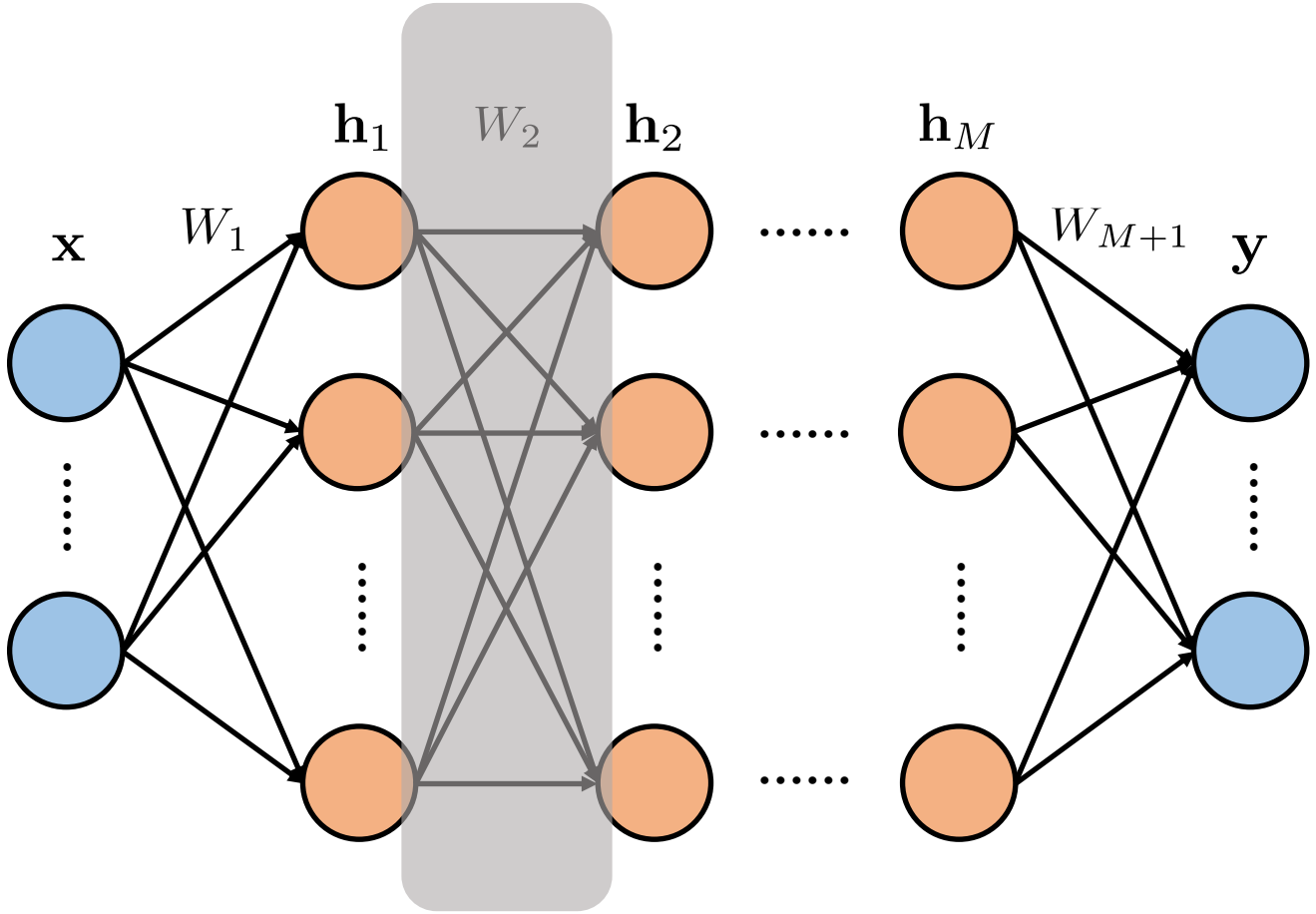
$\mathbf{z}_2 = W_2 \mathbf{h}_1$

Since the nonlinearity is element-wise, we have

$$\frac{\partial L}{\partial \mathbf{z}_2} = \left(\frac{\partial \mathbf{h}_2}{\partial \mathbf{z}_2}\right)^\top \frac{\partial L}{\partial \mathbf{h}_2}$$

Backpropagation

During the backward pass:



Loss: $L = \ell(\mathbf{y}, \bar{\mathbf{y}})$

Gradient of loss w.r.t. $\mathbf{y}$: $\frac{\partial L}{\partial \mathbf{y}}$

Gradient of loss w.r.t. W_2 :

Recall $\mathbf{h}_2 = \sigma(W_2 \mathbf{h}_1)$

$$\mathbf{z}_2 = W_2 \mathbf{h}_1$$

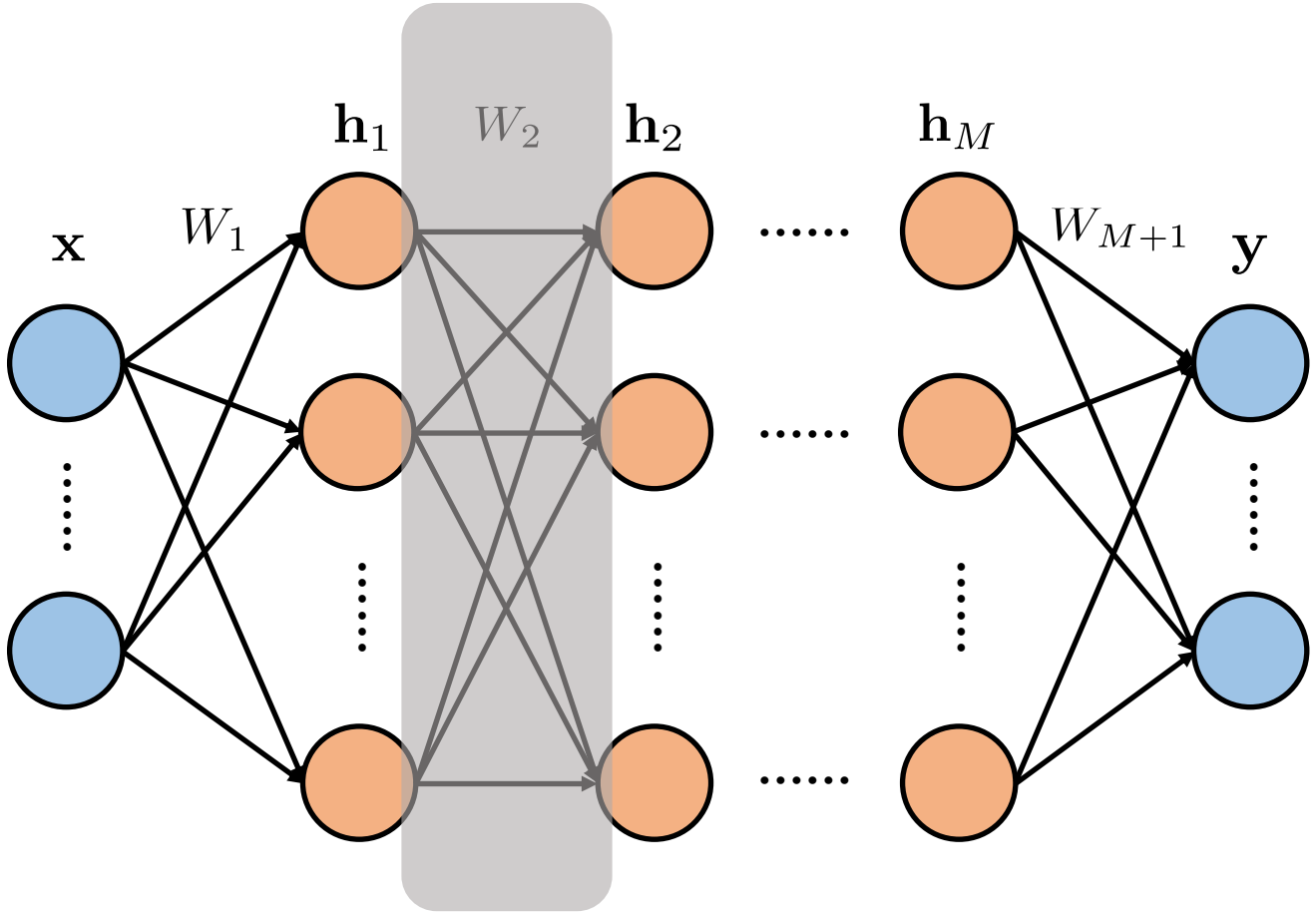
Since the nonlinearity is element-wise, we have

$$\frac{\partial L}{\partial \mathbf{z}_2} = \left(\frac{\partial \mathbf{h}_2}{\partial \mathbf{z}_2}\right)^\top \frac{\partial L}{\partial \mathbf{h}_2} = \sigma'(\mathbf{z}_2) \odot \frac{\partial L}{\partial \mathbf{h}_2}$$

Why?

Backpropagation

During the backward pass:



Loss: $L = \ell(\mathbf{y}, \bar{\mathbf{y}})$

Gradient of loss w.r.t. $\mathbf{y}$: $\frac{\partial L}{\partial \mathbf{y}}$

Gradient of loss w.r.t. W_2 :

Recall $\mathbf{h}_2 = \sigma(W_2 \mathbf{h}_1)$

$$\mathbf{z}_2 = W_2 \mathbf{h}_1$$

Since the nonlinearity is element-wise, we have

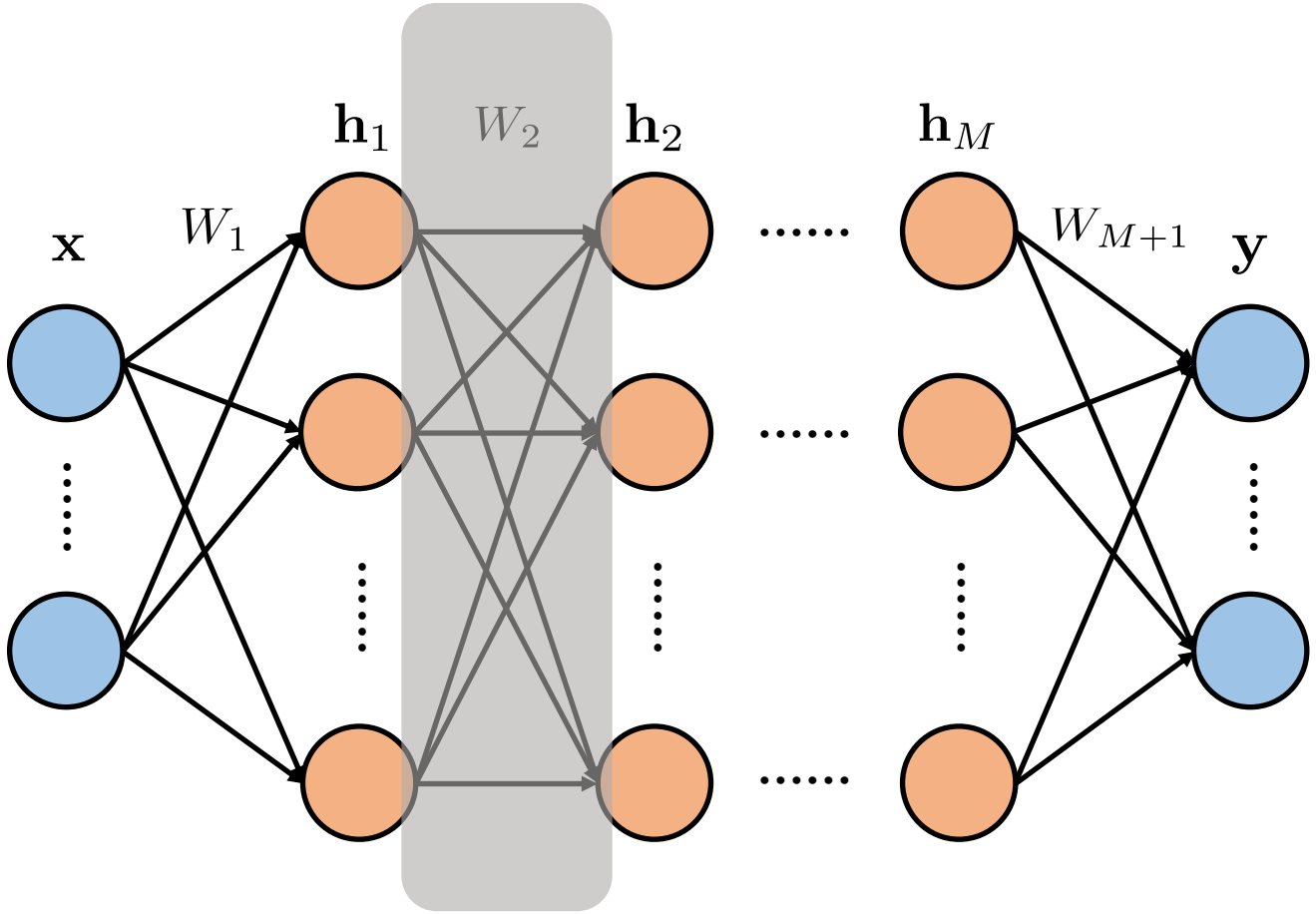
$$\frac{\partial L}{\partial \mathbf{z}_2} = \left(\frac{\partial \mathbf{h}_2}{\partial \mathbf{z}_2}\right)^\top \frac{\partial L}{\partial \mathbf{h}_2} = \sigma'(\mathbf{z}_2) \odot \frac{\partial L}{\partial \mathbf{h}_2}$$

Why?

Again, the Jacobian of element-wise nonlinearity is a diagonal matrix!

Backpropagation

During the backward pass:



Loss: $L = \ell(\mathbf{y}, \bar{\mathbf{y}})$

Gradient of loss w.r.t. $\mathbf{y}$: $\frac{\partial L}{\partial \mathbf{y}}$

Gradient of loss w.r.t. W_2 :

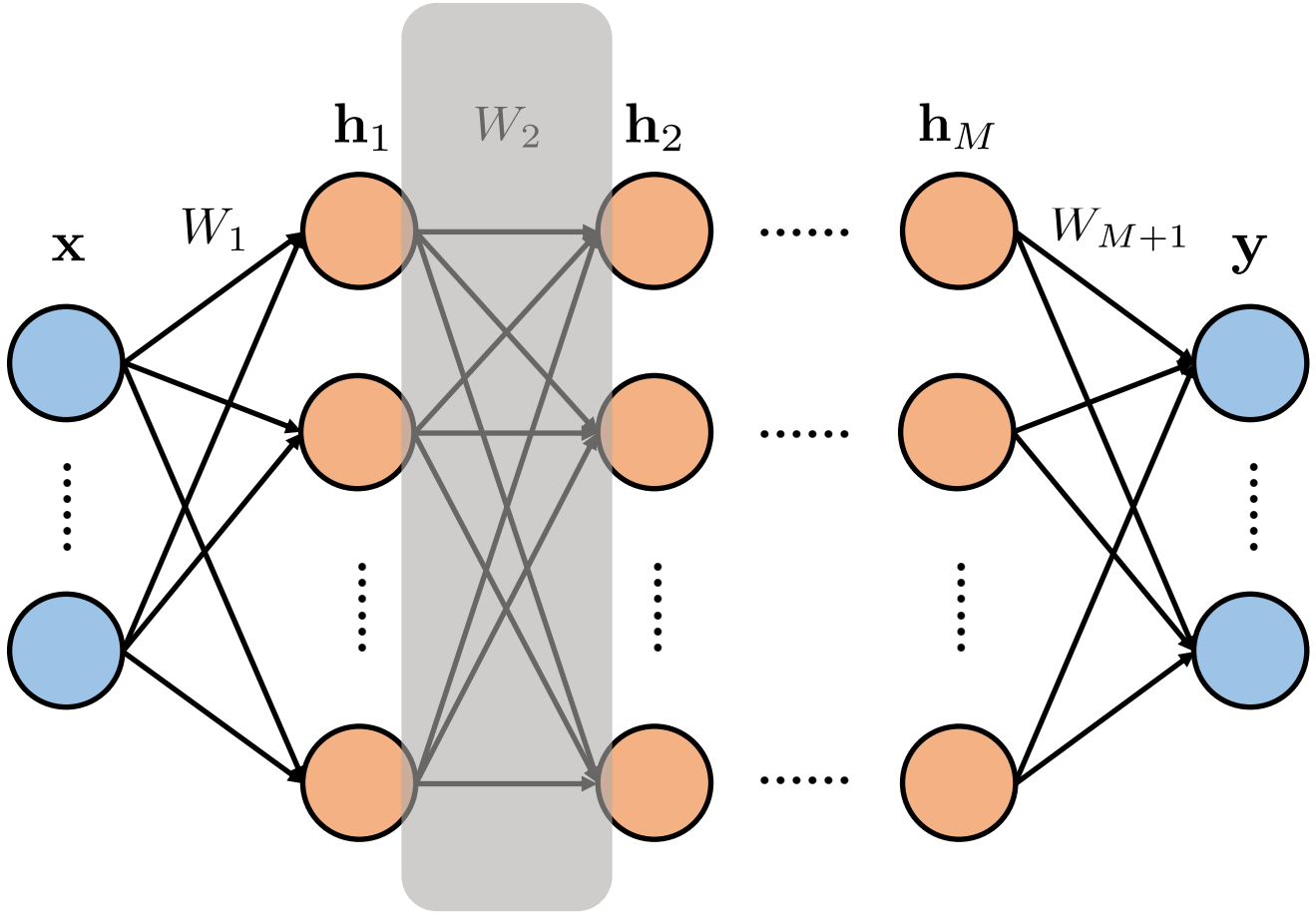
Recall $\mathbf{h}_2 = \sigma(W_2 \mathbf{h}_1)$

$$\mathbf{z}_2 = W_2 \mathbf{h}_1$$

$$\frac{\partial L}{\partial \mathbf{z}_2} = \left(\frac{\partial \mathbf{h}_2}{\partial \mathbf{z}_2}\right)^\top \frac{\partial L}{\partial \mathbf{h}_2} = \sigma'(\mathbf{z}_2) \odot \frac{\partial L}{\partial \mathbf{h}_2}$$

Backpropagation

During the backward pass:



Loss: $L = \ell(\mathbf{y}, \bar{\mathbf{y}})$

Gradient of loss w.r.t. $\mathbf{y}$: $\frac{\partial L}{\partial \mathbf{y}}$

Gradient of loss w.r.t. W_2 :

Recall $\mathbf{h}_2 = \sigma(W_2 \mathbf{h}_1)$

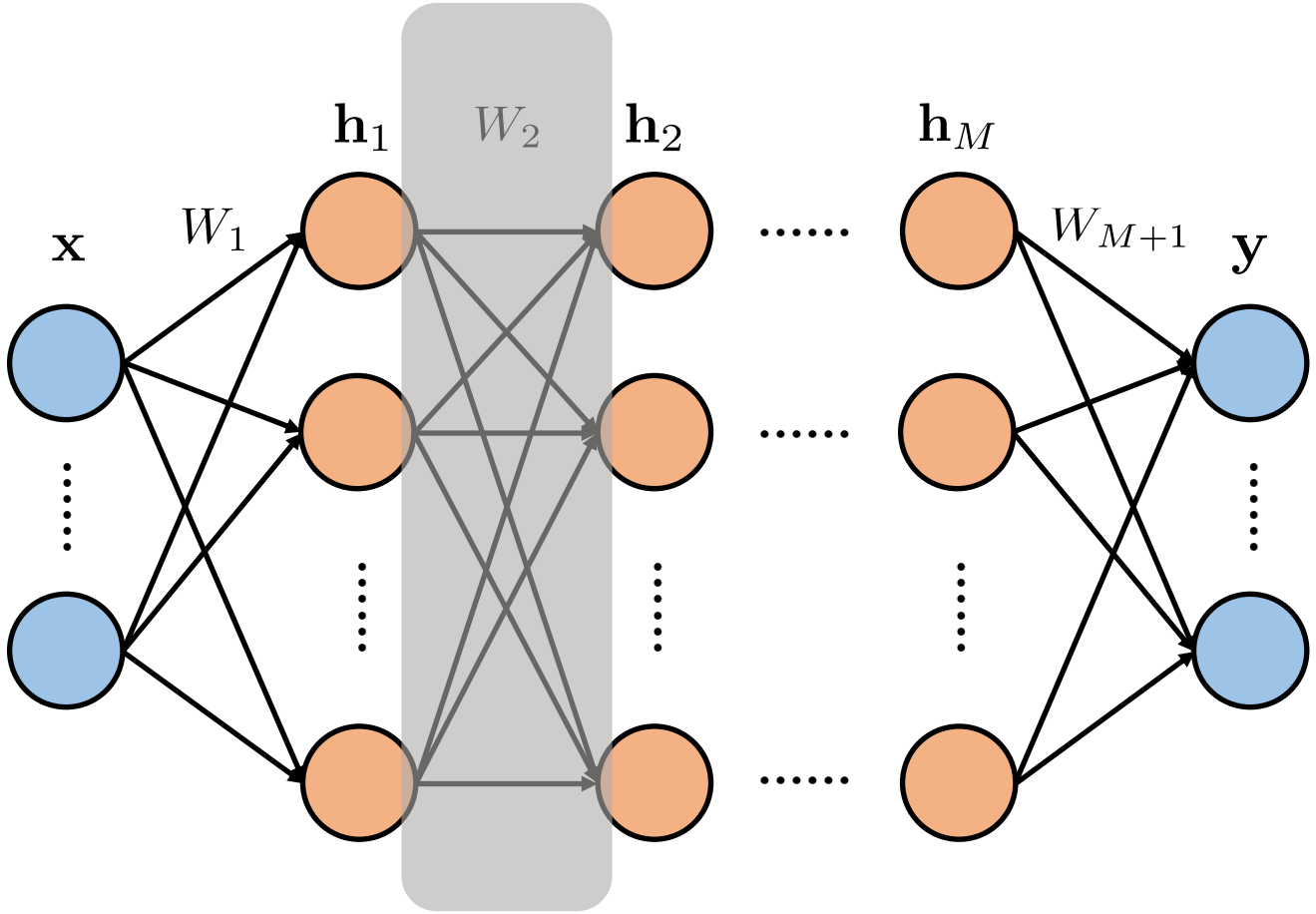
$$\mathbf{z}_2 = W_2 \mathbf{h}_1$$

$$\frac{\partial L}{\partial \mathbf{z}_2} = \left(\frac{\partial \mathbf{h}_2}{\partial \mathbf{z}_2}\right)^\top \frac{\partial L}{\partial \mathbf{h}_2} = \sigma'(\mathbf{z}_2) \odot \frac{\partial L}{\partial \mathbf{h}_2}$$

Note $\mathbf{z}_2[i] = \sum_j W_2[i,j]\mathbf{h}_1[j]$

Backpropagation

During the backward pass:



Loss: $L = \ell(\mathbf{y}, \bar{\mathbf{y}})$

Gradient of loss w.r.t. $\mathbf{y}$: $\frac{\partial L}{\partial \mathbf{y}}$

Gradient of loss w.r.t. W_2 :

Recall $\mathbf{h}_2 = \sigma(W_2 \mathbf{h}_1)$

$$\mathbf{z}_2 = W_2 \mathbf{h}_1$$

$$\frac{\partial L}{\partial \mathbf{z}_2} = \left(\frac{\partial \mathbf{h}_2}{\partial \mathbf{z}_2}\right)^\top \frac{\partial L}{\partial \mathbf{h}_2} = \sigma'(\mathbf{z}_2) \odot \frac{\partial L}{\partial \mathbf{h}_2}$$

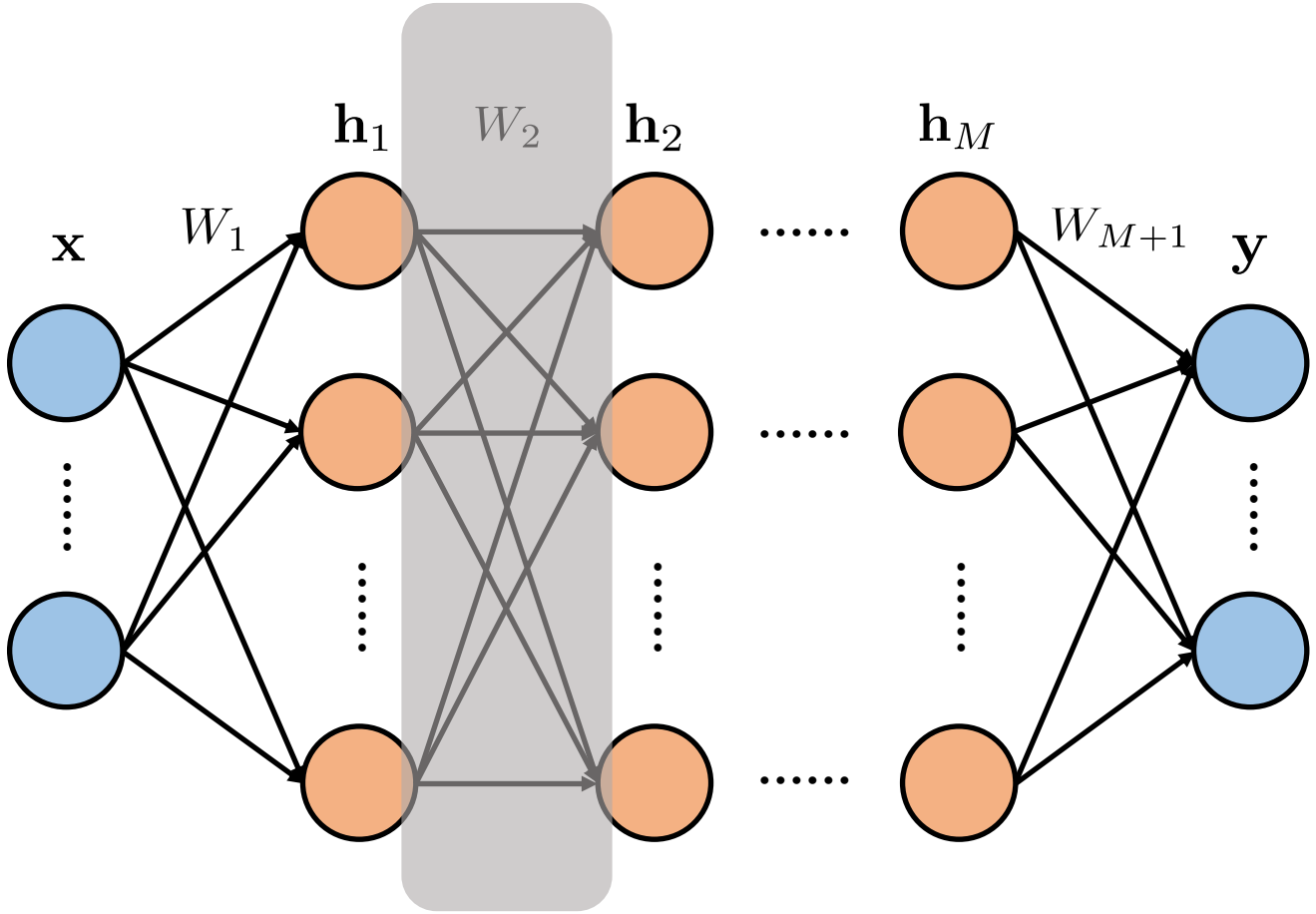
Note $\mathbf{z}_2[i] = \sum_j W_2[i,j]\mathbf{h}_1[j]$

Similarly as before, we have

$$\frac{\partial L}{\partial W_2[i,j]} = \frac{\partial L}{\partial \mathbf{z}_2[i]} \frac{\partial \mathbf{z}_2[i]}{\partial W_2[i,j]} = \frac{\partial L}{\partial \mathbf{z}_2[i]} \mathbf{h}_1[j]$$

Backpropagation

During the backward pass:



Loss: $L = \ell(\mathbf{y}, \bar{\mathbf{y}})$

Gradient of loss w.r.t. $\mathbf{y}$: $\frac{\partial L}{\partial \mathbf{y}}$

Gradient of loss w.r.t. W_2 :

Recall $\mathbf{h}_2 = \sigma(W_2 \mathbf{h}_1)$

$$\mathbf{z}_2 = W_2 \mathbf{h}_1$$

$$\frac{\partial L}{\partial \mathbf{z}_2} = \left(\frac{\partial \mathbf{h}_2}{\partial \mathbf{z}_2}\right)^\top \frac{\partial L}{\partial \mathbf{h}_2} = \sigma'(\mathbf{z}_2) \odot \frac{\partial L}{\partial \mathbf{h}_2}$$

Note $\mathbf{z}_2[i] = \sum_j W_2[i,j]\mathbf{h}_1[j]$

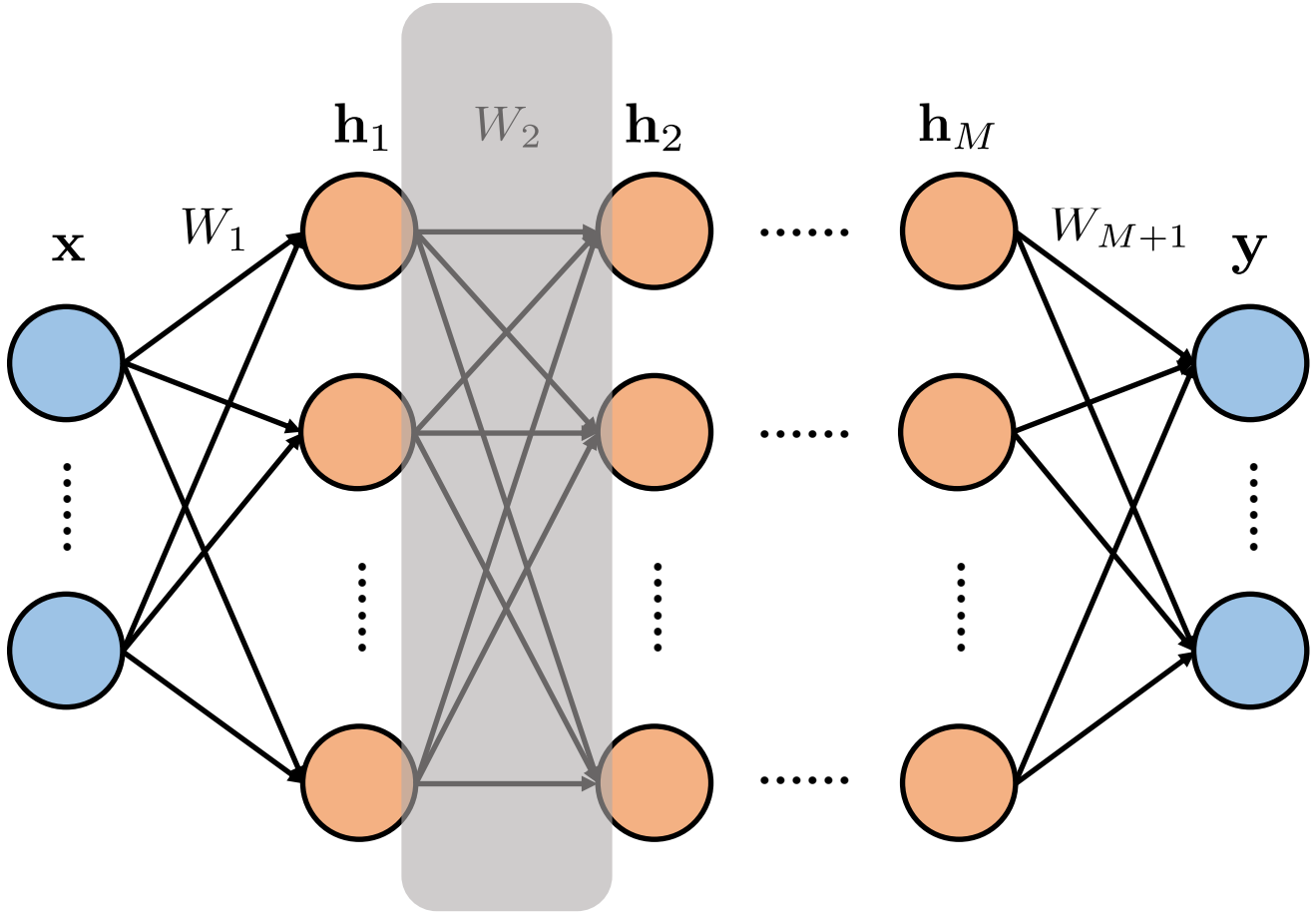
Similarly as before, we have

$$\frac{\partial L}{\partial W_2[i,j]} = \frac{\partial L}{\partial \mathbf{z}_2[i]} \frac{\partial \mathbf{z}_2[i]}{\partial W_2[i,j]} = \frac{\partial L}{\partial \mathbf{z}_2[i]} \mathbf{h}_1[j]$$

$$\frac{\partial L}{\partial W_2} = \left(\sigma'(\mathbf{z}_2) \odot \frac{\partial L}{\partial \mathbf{h}_2}\right) \mathbf{h}_1^\top$$

Backpropagation

During the backward pass:



In summary:

$$\mathbf{z}_i = W_i \mathbf{h}_{i-1}$$

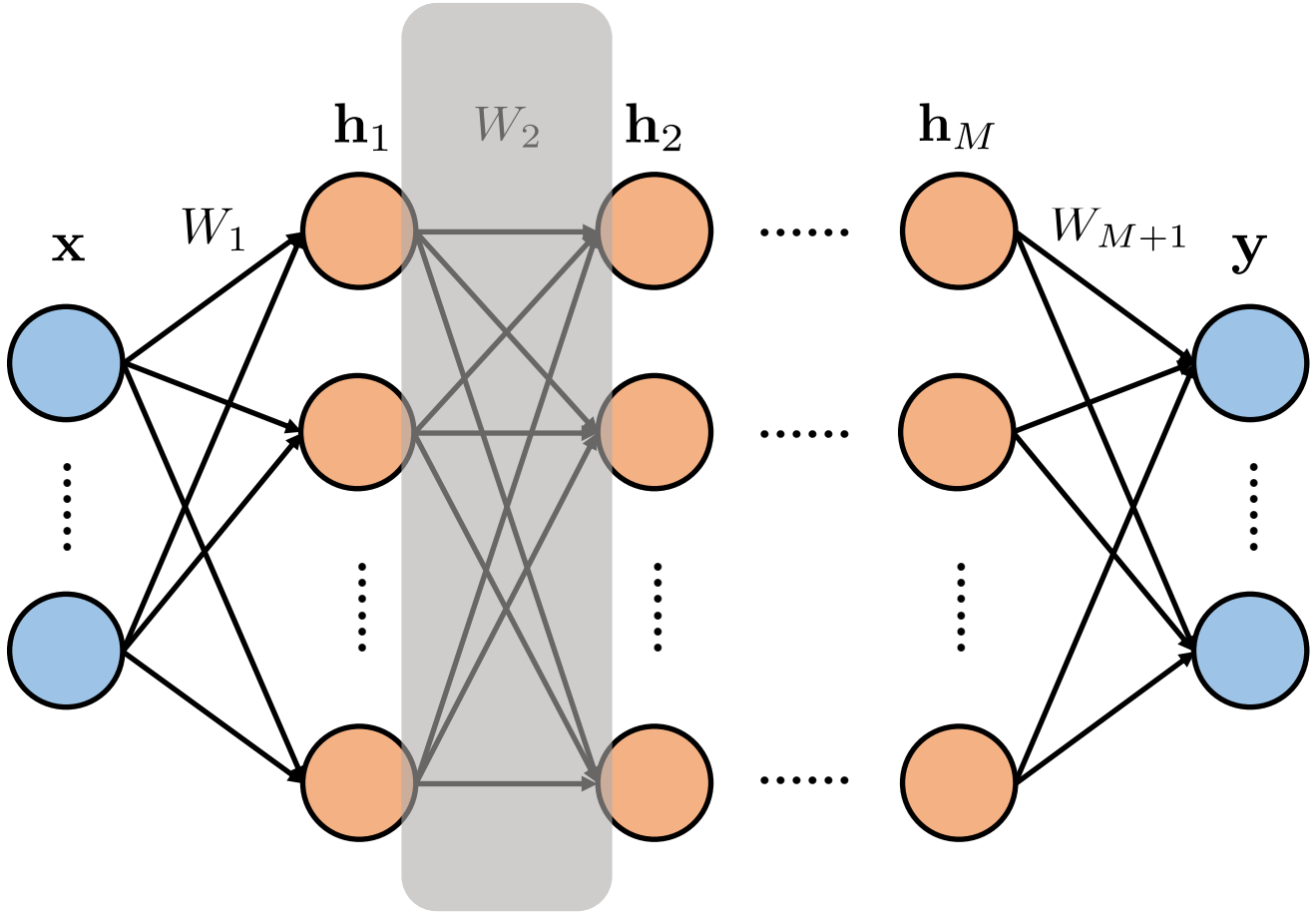
$$\mathbf{J}_i = \mathrm{diag}(\sigma'(\mathbf{z}_i)) W_i$$

$$\frac{\partial L}{\partial \mathbf{h}_i} = \mathbf{J}_{i+1}^\top \cdots \mathbf{J}_M^\top \frac{\partial L}{\partial \mathbf{y}}$$

$$\frac{\partial L}{\partial W_i} = \left(\sigma'(\mathbf{z}_i) \odot \frac{\partial L}{\partial \mathbf{h}_i} \right) \mathbf{h}_{i-1}^\top$$

Backpropagation

During the backward pass:



In summary:

$$\mathbf{z}_i = W_i \mathbf{h}_{i-1}$$

$$\mathbf{J}_i = \mathrm{diag}(\sigma'(\mathbf{z}_i)) W_i$$

$$\frac{\partial L}{\partial \mathbf{h}_i} = \mathbf{J}_{i+1}^\top \cdots \mathbf{J}_M^\top \frac{\partial L}{\partial \mathbf{y}}$$

$$\frac{\partial L}{\partial W_i} = \left(\sigma'(\mathbf{z}_i) \odot \frac{\partial L}{\partial \mathbf{h}_i} \right) \mathbf{h}_{i-1}^\top$$

We can cache these tensors in the forward pass to avoid duplicated computation in the backward pass!

Backpropagation with Biases

Forward and backward passes with biases

$$\mathbf{h}_0 = \mathbf{x}, \qquad m = 1, \ldots, M$$

$$\mathbf{z}_m = W_m \mathbf{h}_{m-1} + \mathbf{b}_m, \qquad \mathbf{h}_m = \sigma_m(\mathbf{z}_m)$$

$$\mathbf{y} = W_{M+1} \mathbf{h}_M + \mathbf{b}_{M+1}, \qquad L = \ell(\mathbf{y}, \mathbf{g})$$

The output layer is linear, and g denotes the target.

Start the backward pass with the derivative of the loss with respect to the output.

$$\boldsymbol{\delta}_{M+1} := \nabla_{\mathbf{y}} L$$

Backpropagation with Biases

Forward and backward passes with biases

$$\mathbf{z}_m = W_m \mathbf{h}_{m-1} + \mathbf{b}_m, \qquad \mathbf{h}_m = \sigma_m(\mathbf{z}_m)$$

$$\mathbf{y} = W_{M+1}\mathbf{h}_M + \mathbf{b}_{M+1}, \qquad \boldsymbol{\delta}_{M+1} = \nabla_{\mathbf{y}} L$$

$\boldsymbol{\delta}_m := \nabla_{\mathbf{z}_m} L$ Propagate the loss gradient backward through each hidden layer.

$$\boldsymbol{\delta}_m = \left(W_{m+1}^\top \boldsymbol{\delta}_{m+1}\right) \odot \sigma_m'(\mathbf{z}_m), \qquad m = M, \ldots, 1$$

The same adjoint gives both parameter gradients for the layer.

$$\nabla_{W_m} L = \boldsymbol{\delta}_m \mathbf{h}_{m-1}^\top \qquad \nabla_{\mathbf{b}_m} L = \boldsymbol{\delta}_m$$

The parameter-gradient formulas also apply to the output layer (m = M + 1).

Mini-batch Gradients

Gradients of the average mini-batch loss

$$L_{\mathcal{B}} = \frac{1}{B}\sum_{b=1}^{B} \ell^{(b)}, \qquad \boldsymbol{\delta}_m^{(b)} := \frac{\partial \ell^{(b)}}{\partial \mathbf{z}_m^{(b)}}$$

Each δ comes from the per-sample loss. Average these gradients once.

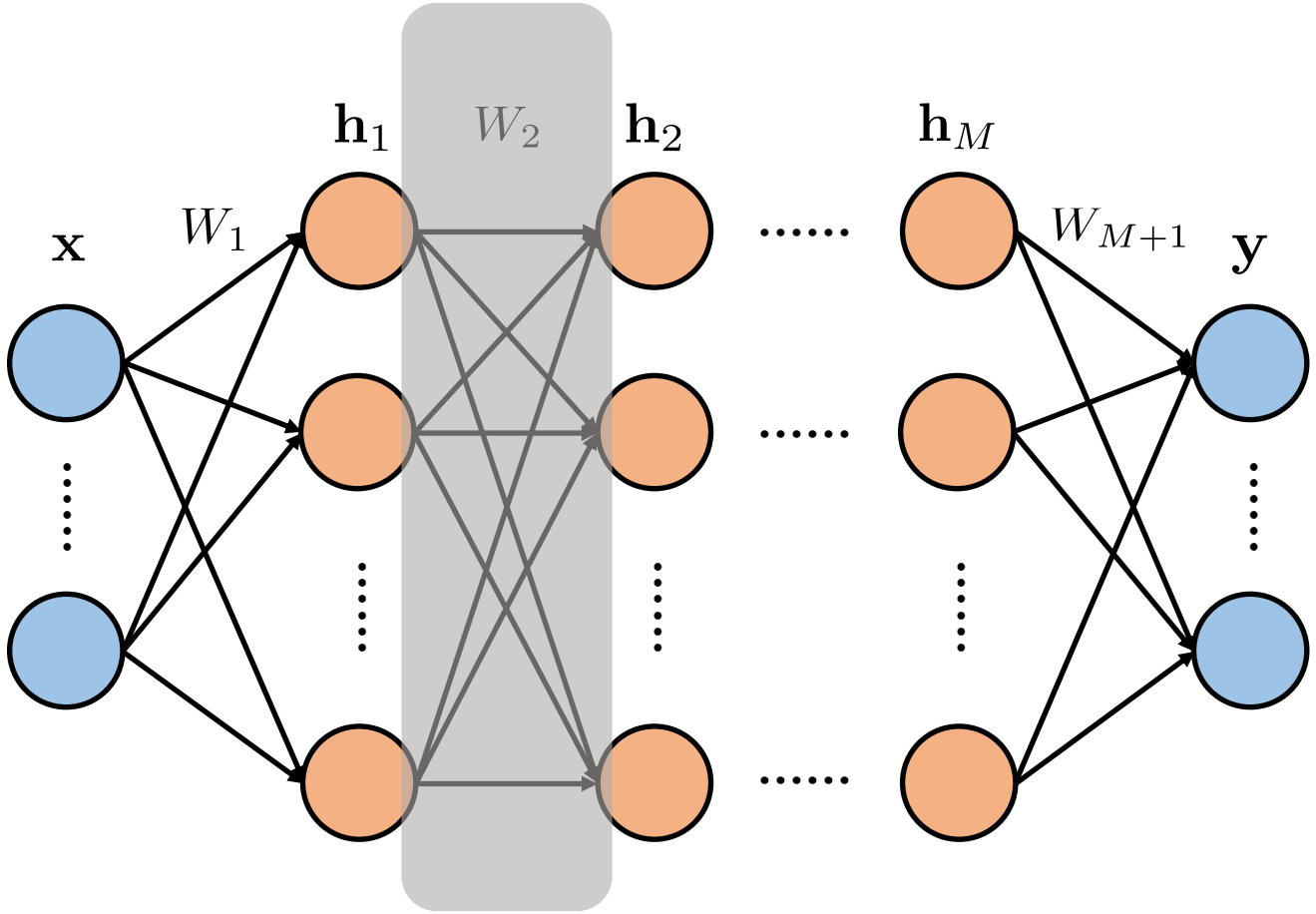
$$\nabla_{W_m} L_{\mathcal{B}} = \frac{1}{B}\sum_{b=1}^{B} \boldsymbol{\delta}_m^{(b)} \left(\mathbf{h}_{m-1}^{(b)}\right)^{\top} \qquad \nabla_{\mathbf{b}_m} L_{\mathcal{B}} = \frac{1}{B}\sum_{b=1}^{B} \boldsymbol{\delta}_m^{(b)}$$

Matrix form with samples stored as columns

$$\nabla_{W_m} L_{\mathcal{B}} = \frac{1}{B}\Delta_m H_{m-1}^{\top}, \qquad \nabla_{\mathbf{b}_m} L_{\mathcal{B}} = \frac{1}{B}\Delta_m \mathbf{1}$$

This per-example derivation excludes training-time BatchNorm, which couples examples.

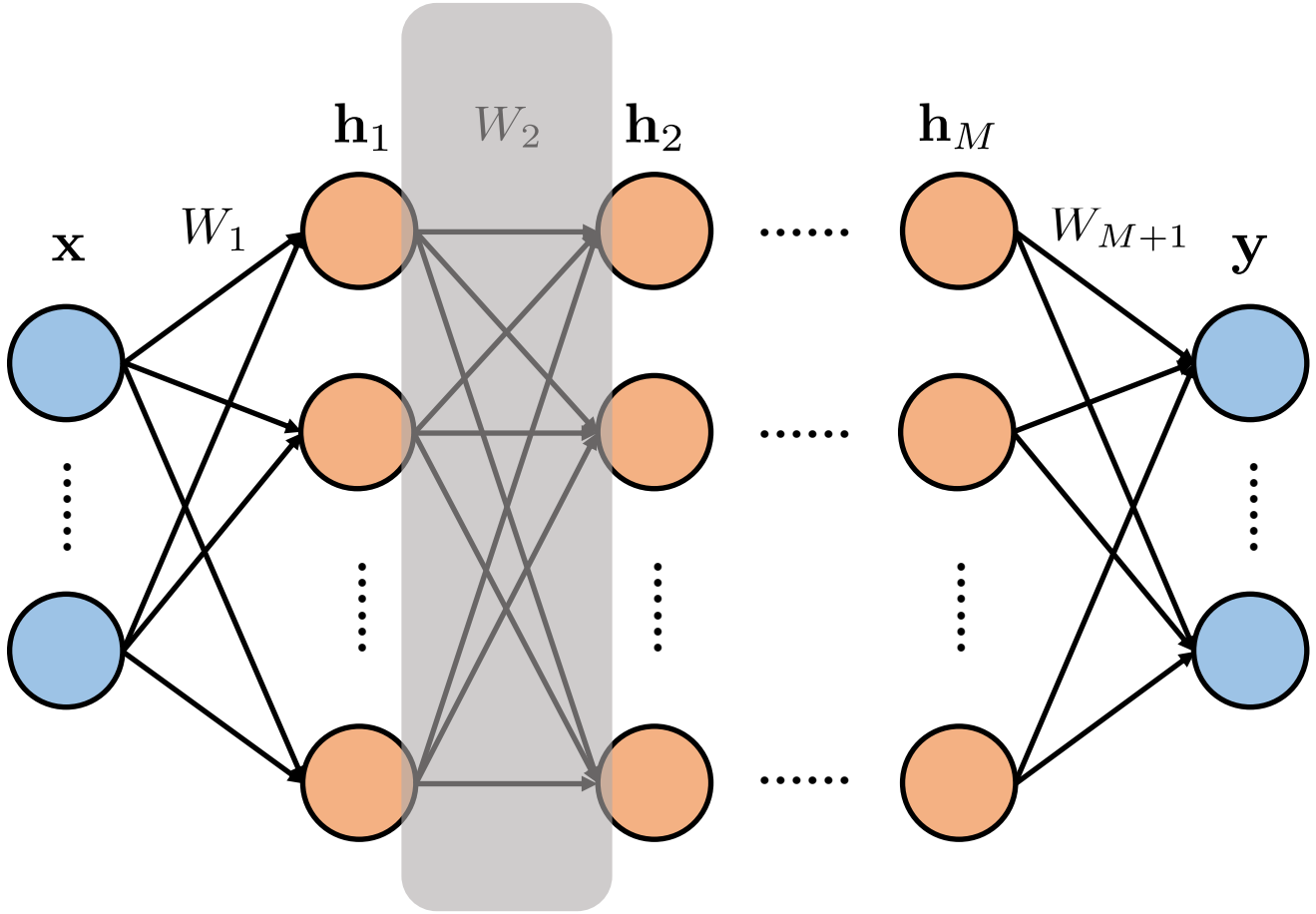
Forward vs. Backward



Computation in Forward Pass:

$$\mathbf{h}_2 = \sigma\left(W_2\mathbf{h}_1\right)$$

Forward vs. Backward



Computation in Forward Pass:

$$\mathbf{h}_2 = \sigma\left(W_2\mathbf{h}_1\right)$$

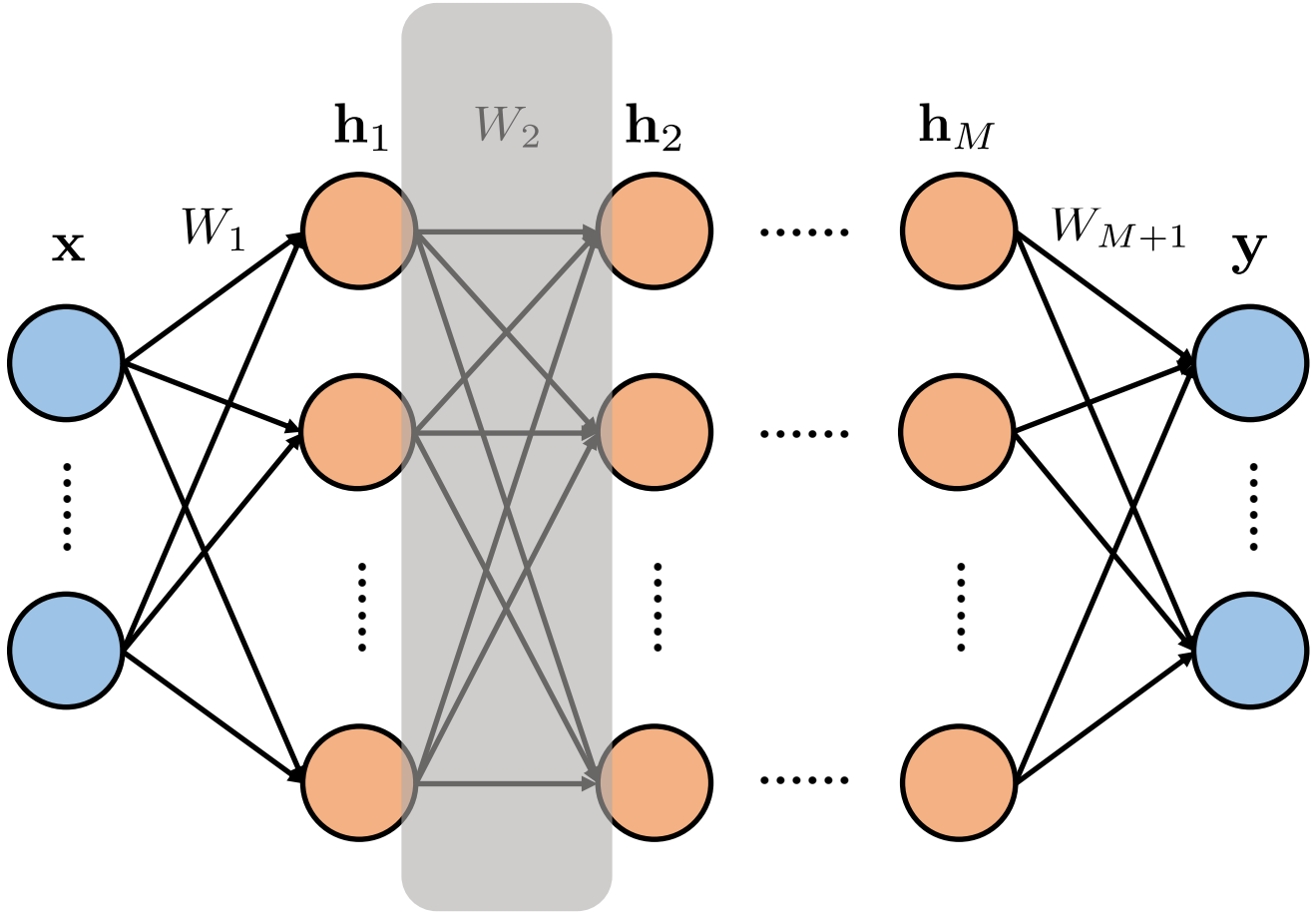
Computation in Backward Pass:

$$\frac{\partial L}{\partial \mathbf{h}_1} = \mathbf{J}_2^\top \frac{\partial L}{\partial \mathbf{h}_2}$$

$$\frac{\partial L}{\partial W_2} = \left(\sigma'(\mathbf{z}_2) \odot \frac{\partial L}{\partial \mathbf{h}_2}\right) \mathbf{h}_1^\top$$

$\mathbf{z}_2 = W_2\mathbf{h}_1$ (cached)

Forward vs. Backward



Computation in Forward Pass:

$$\mathbf{h}_2 = \sigma\left(W_2\mathbf{h}_1\right)$$

Computation in Backward Pass:

$$\frac{\partial L}{\partial \mathbf{h}_1} = \mathbf{J}_2^\top \frac{\partial L}{\partial \mathbf{h}_2}$$

$$\frac{\partial L}{\partial W_2} = \left(\sigma'(\mathbf{z}_2) \odot \frac{\partial L}{\partial \mathbf{h}_2}\right) \mathbf{h}_1^\top$$

$$\mathbf{z}_2 = W_2\mathbf{h}_1 \qquad \text{(cached)}$$

For dense layers, the two main backward matrix products cost roughly twice the forward product.

Automatic Differentiation

Backpropagation is reverse-mode automatic differentiation [4].

Apply local vector–Jacobian products, usually without forming full Jacobians.

$$\bar{\mathbf{v}} := \nabla_{\mathbf{v}} L$$

When a value feeds several later operations, the backward pass adds every contribution.

$$\bar{\mathbf{v}} = \sum_{\mathbf{u}:\, \mathbf{v} \to \mathbf{u}} J_{\mathbf{u},\mathbf{v}}^{\top} \bar{\mathbf{u}}$$

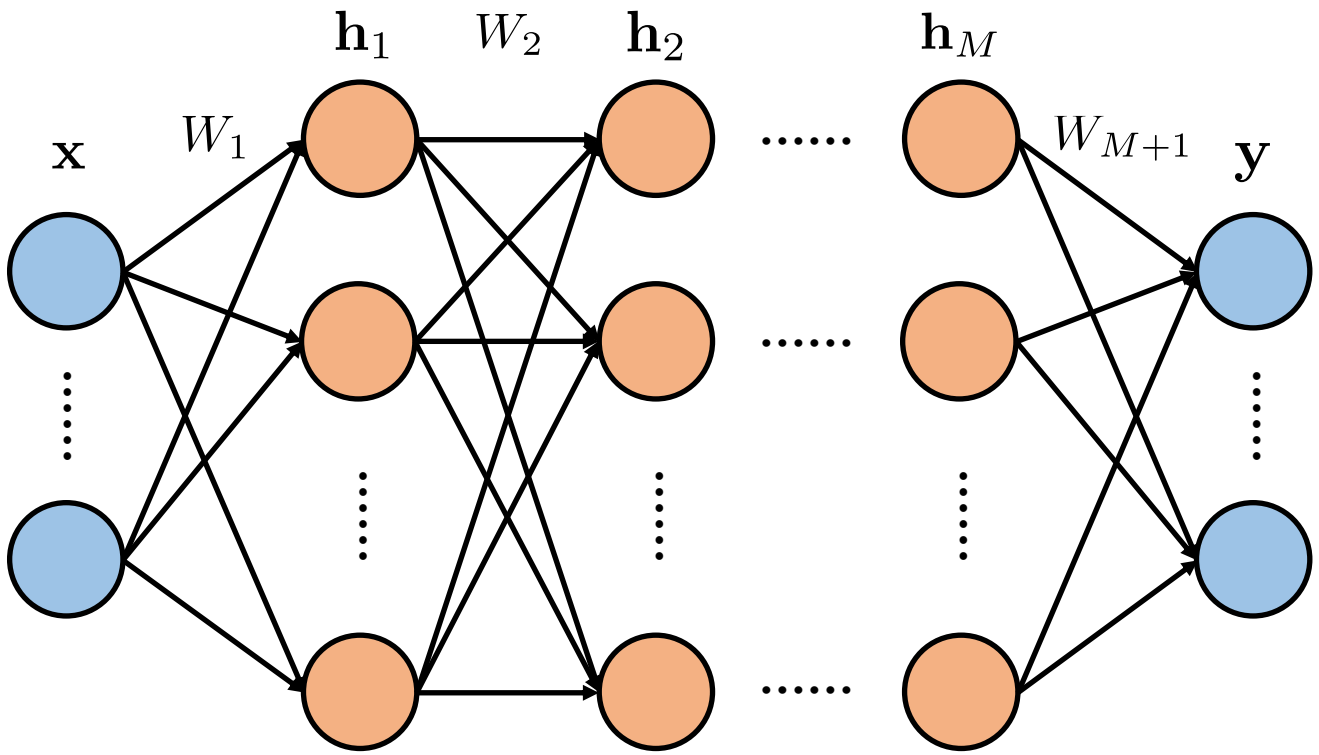
loss.backward() computes gradients. optimizer.step() updates parameters.

Outline

- Training Feedforward Neural Networks
  - Backpropagation
  - **Weight Initialization**

Initialization

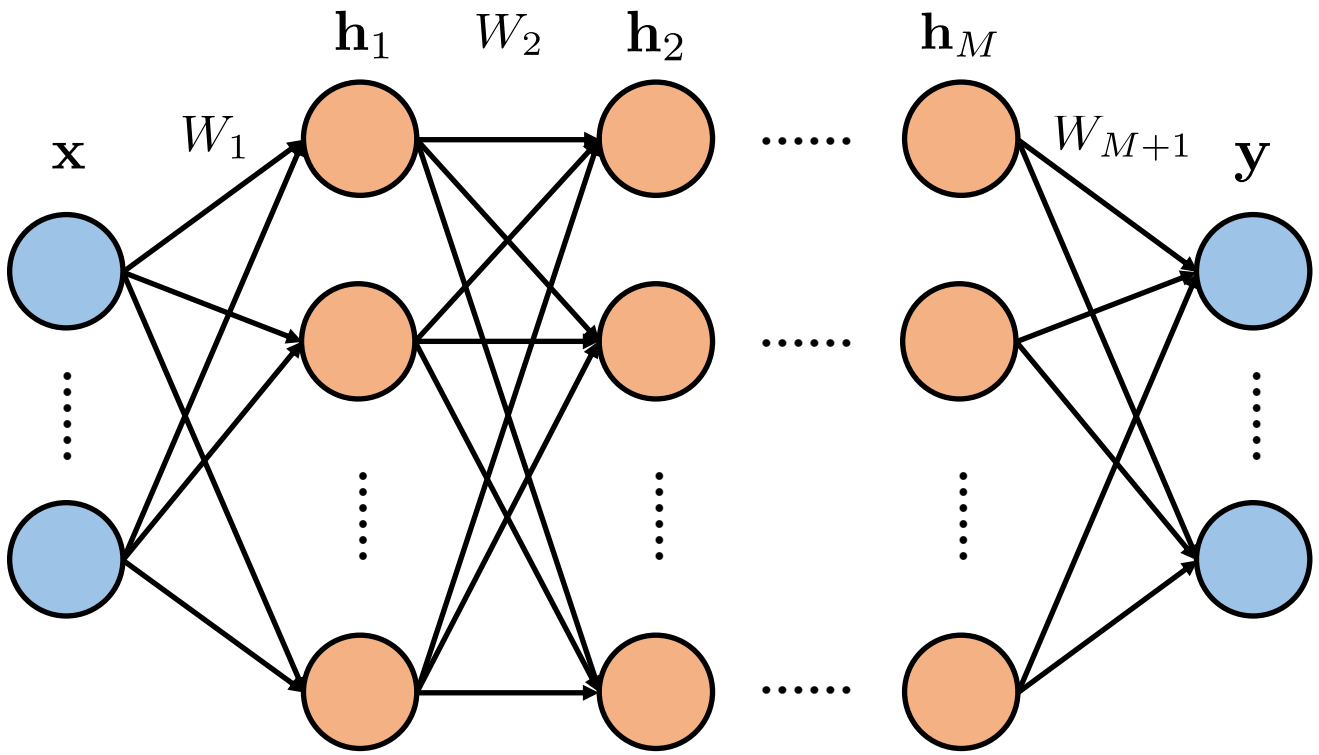
How should we initialize the neural network?



Initialization

How should we initialize the neural network?

Random weights break symmetry between hidden units.

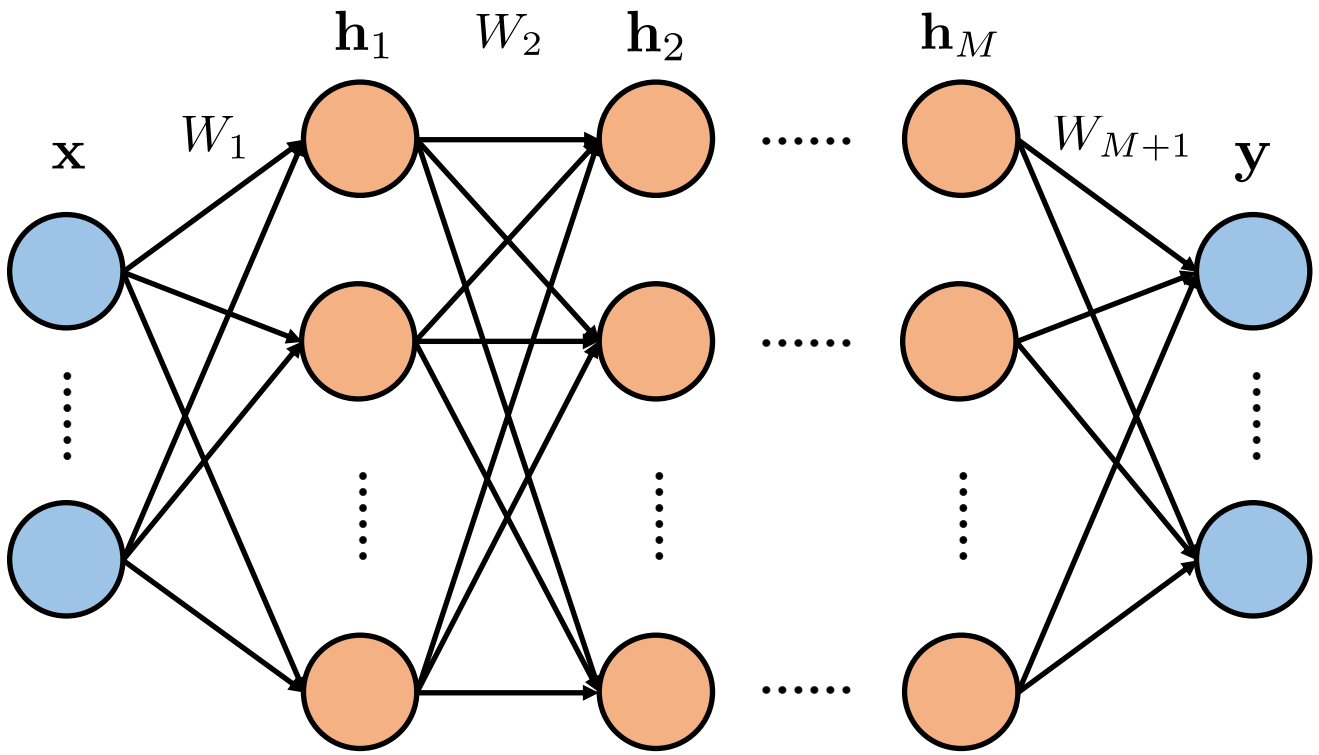


Initialization

How should we initialize the neural network?

Random weights break symmetry between hidden units.

The weight scale should keep activations and gradients stable.

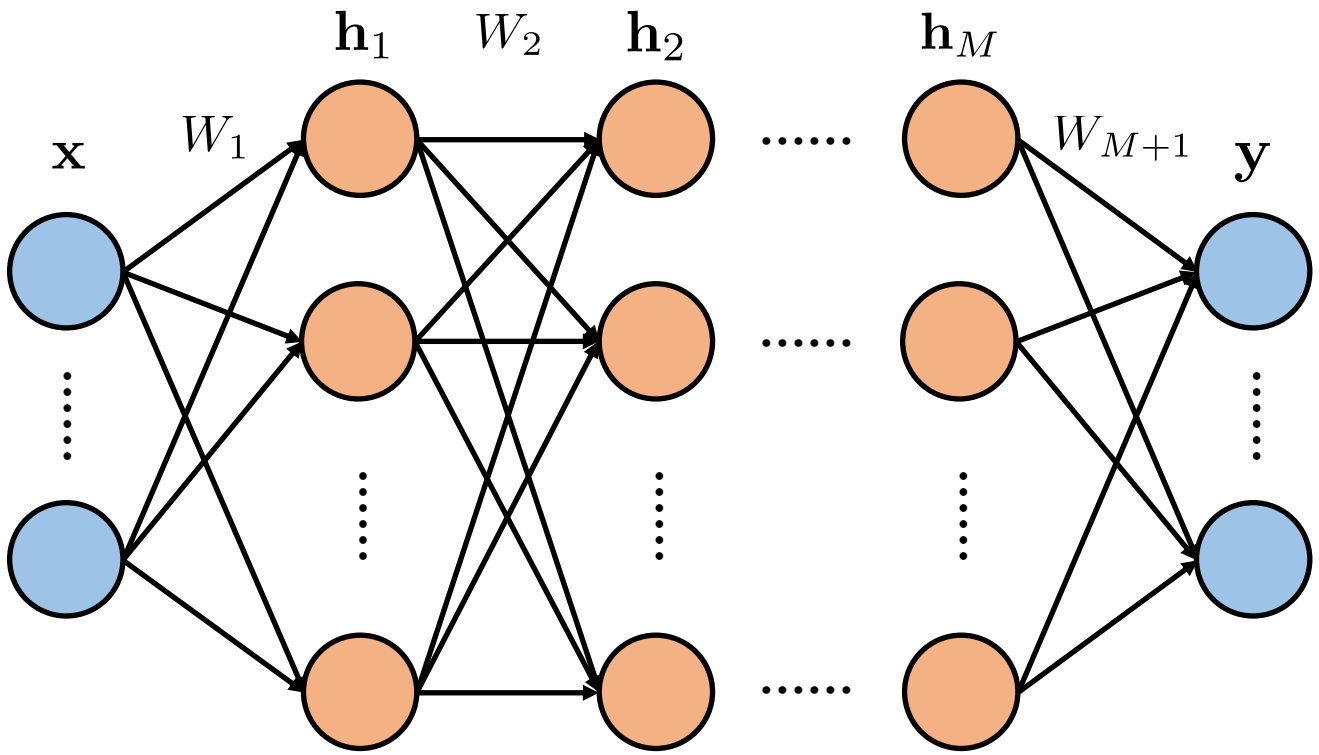


Initialization

How should we initialize the neural network?

Random weights break symmetry between hidden units.

The weight scale should keep activations and gradients stable.



A useful criterion:

Avoid exploding or vanishing signals at initialization.

Initialization

Break symmetry between hidden units

Use different random weights. Identical units can learn identical features.

Control the scale of activations and gradients

Very small or very large weights can make deep networks difficult to train.

Zero biases are usually a reasonable starting point.

Initialization

Let us recap some basic facts about expectation

Linearity $\mathbb{E}[x + y] = \mathbb{E}[x] + \mathbb{E}[y]$

Initialization

Let us recap some basic facts about expectation

Linearity $$\mathbb{E}[x+y] = \mathbb{E}[x] + \mathbb{E}[y]$$

For two independent random variables

$$\mathbb{E}[xy] = \mathbb{E}[x]\mathbb{E}[y]$$
$$\mathbb{E}[x^2y^2] = \mathbb{E}[x^2]\mathbb{E}[y^2]$$

Initialization

Let us recap some basic facts about variance

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

Initialization

Let us recap some basic facts about variance

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

For two independent random variables, we have

$$\begin{aligned}
\mathbb{V}[x+y] &= \mathbb{E}[(x+y)^2] - \mathbb{E}[x+y]^2 \\
&= \mathbb{E}[x^2 + y^2 + 2xy] - (\mathbb{E}[x] + \mathbb{E}[y])^2 \\
&= \mathbb{E}[x^2] + \mathbb{E}[y^2] + 2\mathbb{E}[xy] - \mathbb{E}[x]^2 - \mathbb{E}[y]^2 - 2\mathbb{E}[x]\mathbb{E}[y] \\
&= \mathbb{V}[x] + \mathbb{V}[y] + 2(\mathbb{E}[xy] - \mathbb{E}[x]\mathbb{E}[y]) \\
&= \mathbb{V}[x] + \mathbb{V}[y] + 2\mathrm{Cov}(x, y)
\end{aligned}$$

Initialization

Let us recap some basic facts about variance

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

For two independent random variables, we have

$$\begin{aligned}
\mathbb{V}[x+y] &= \mathbb{E}[(x+y)^2] - \mathbb{E}[x+y]^2 \\
&= \mathbb{E}[x^2 + y^2 + 2xy] - (\mathbb{E}[x] + \mathbb{E}[y])^2 \\
&= \mathbb{E}[x^2] + \mathbb{E}[y^2] + 2\mathbb{E}[xy] - \mathbb{E}[x]^2 - \mathbb{E}[y]^2 - 2\mathbb{E}[x]\mathbb{E}[y] \\
&= \mathbb{V}[x] + \mathbb{V}[y] + 2(\mathbb{E}[xy] - \mathbb{E}[x]\mathbb{E}[y]) \\
&= \mathbb{V}[x] + \mathbb{V}[y] + 2\mathrm{Cov}(x, y) \qquad\qquad \mathbb{E}[xy] = \mathbb{E}[x]\mathbb{E}[y]
\end{aligned}$$

Initialization

Let us recap some basic facts about variance

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

For two independent random variables, we have

$$\begin{aligned}
\mathbb{V}[x+y] &= \mathbb{E}[(x+y)^2] - \mathbb{E}[x+y]^2 \\
&= \mathbb{E}[x^2 + y^2 + 2xy] - (\mathbb{E}[x] + \mathbb{E}[y])^2 \\
&= \mathbb{E}[x^2] + \mathbb{E}[y^2] + 2\mathbb{E}[xy] - \mathbb{E}[x]^2 - \mathbb{E}[y]^2 - 2\mathbb{E}[x]\mathbb{E}[y] \\
&= \mathbb{V}[x] + \mathbb{V}[y] + 2(\mathbb{E}[xy] - \mathbb{E}[x]\mathbb{E}[y]) \\
&= \mathbb{V}[x] + \mathbb{V}[y] + 2\mathrm{Cov}(x, y) \qquad\qquad \mathbb{E}[xy] = \mathbb{E}[x]\mathbb{E}[y] \\
&= \mathbb{V}[x] + \mathbb{V}[y]
\end{aligned}$$

Initialization

Let us recap some basic facts about variance

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

For two independent random variables, we have

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

$$\mathbb{E}[xy] = \mathbb{E}[x]\mathbb{E}[y]$$

$$\mathbb{E}[x^2y^2] = \mathbb{E}[x^2]\mathbb{E}[y^2]$$

Initialization

Let us recap some basic facts about variance

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

For two independent random variables, we have

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

$$\mathbb{E}[xy] = \mathbb{E}[x]\mathbb{E}[y]$$

$$\mathbb{E}[x^2y^2] = \mathbb{E}[x^2]\mathbb{E}[y^2]$$

If $\mathbb{E}[x] = \mathbb{E}[y] = 0$, then $\mathbb{V}[xy] = \mathbb{V}[x]\mathbb{V}[y]$

Initialization

A simplified model for signal propagation

Use independent, zero-mean random weights within each layer.

Approximate independence between weights, activations, and backward signals.

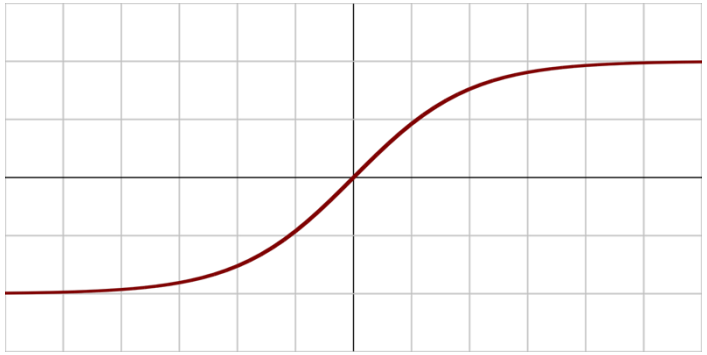
For tanh near zero, use its linear approximation.

$$\mathbb{E}[w] = 0, \quad w \perp h \quad \Longrightarrow \quad \mathrm{Var}(wh) = \mathrm{Var}(w)\,\mathbb{E}[h^2]$$

$$\mathrm{Var}\left(\sum_j u_j\right) = \sum_j \mathrm{Var}(u_j) \quad \text{for independent } u_j$$

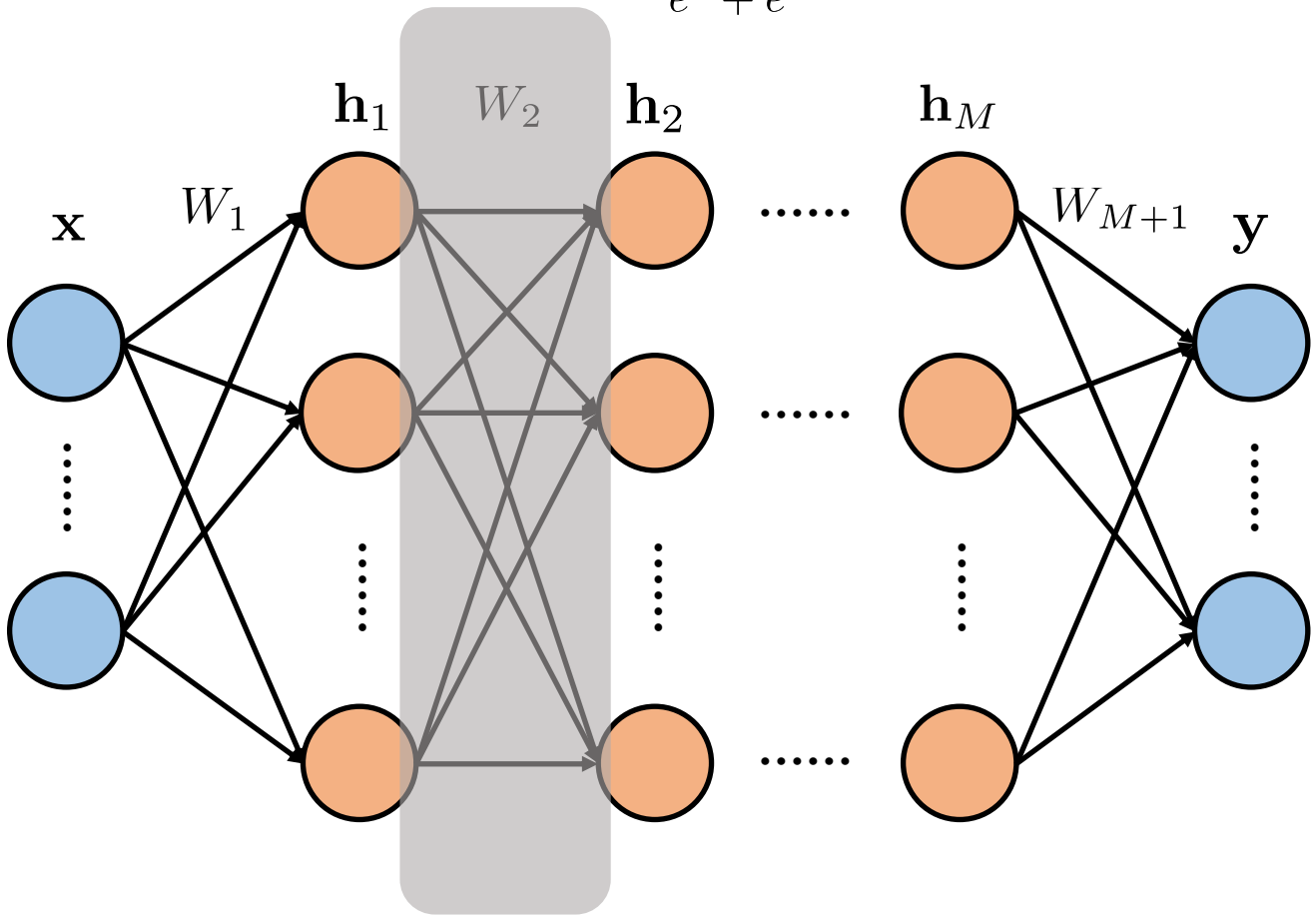
These assumptions describe initialization, not the full training trajectory.

Initialization: Forward Analysis

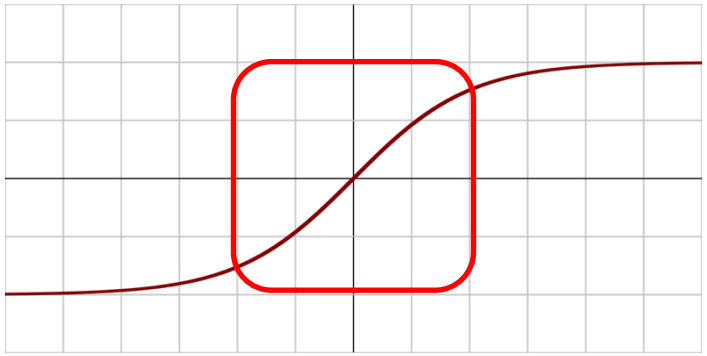


Recall $\mathbf{h}_2 = \sigma\left(W_2\mathbf{h}_1\right)$

If we assume Tanh activation $\sigma(x) = \dfrac{e^x - e^{-x}}{e^x + e^{-x}}$ and we are in the linear regime

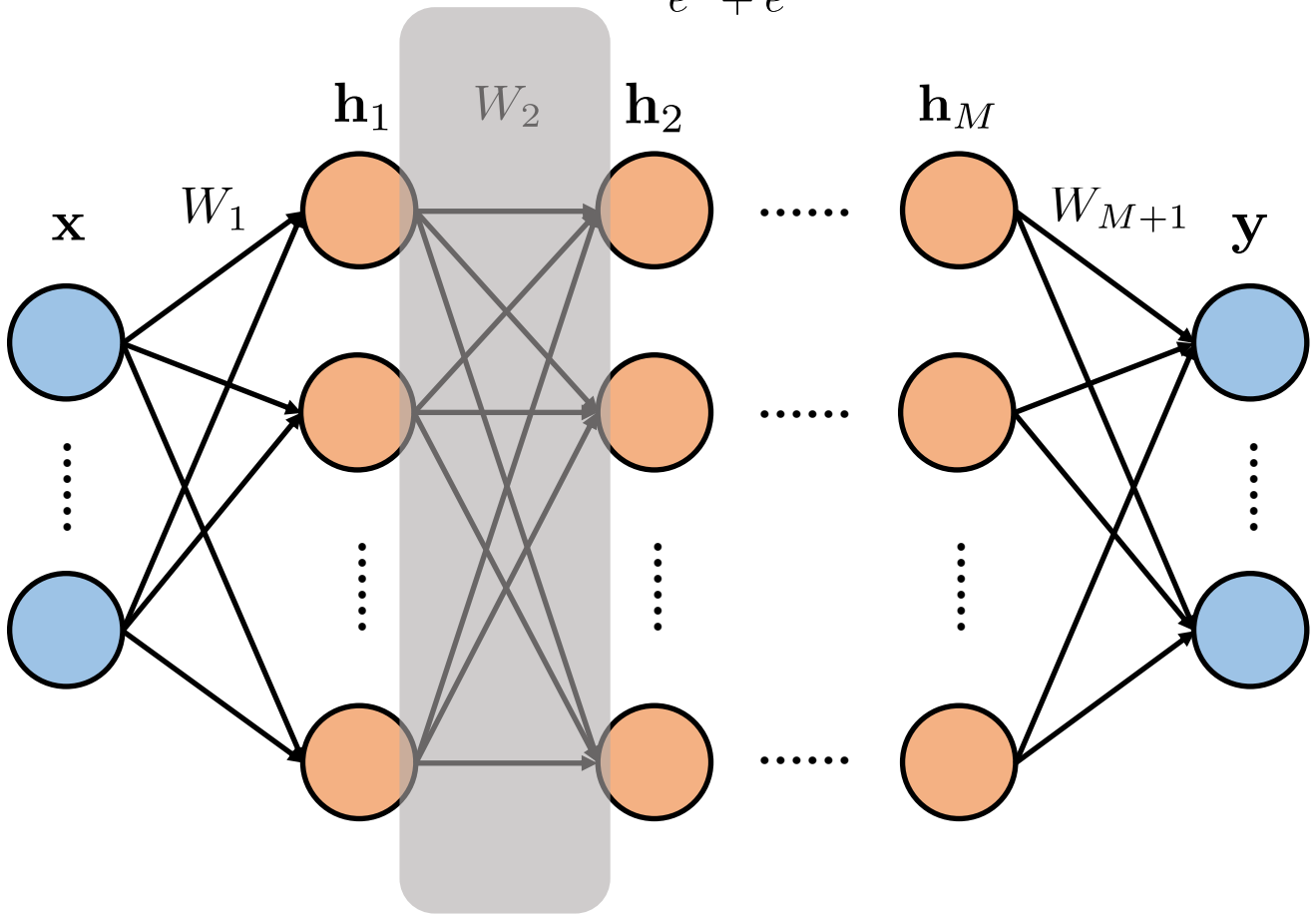


Initialization: Forward Analysis



Recall $\mathbf{h}_2 = \sigma\left(W_2\mathbf{h}_1\right)$

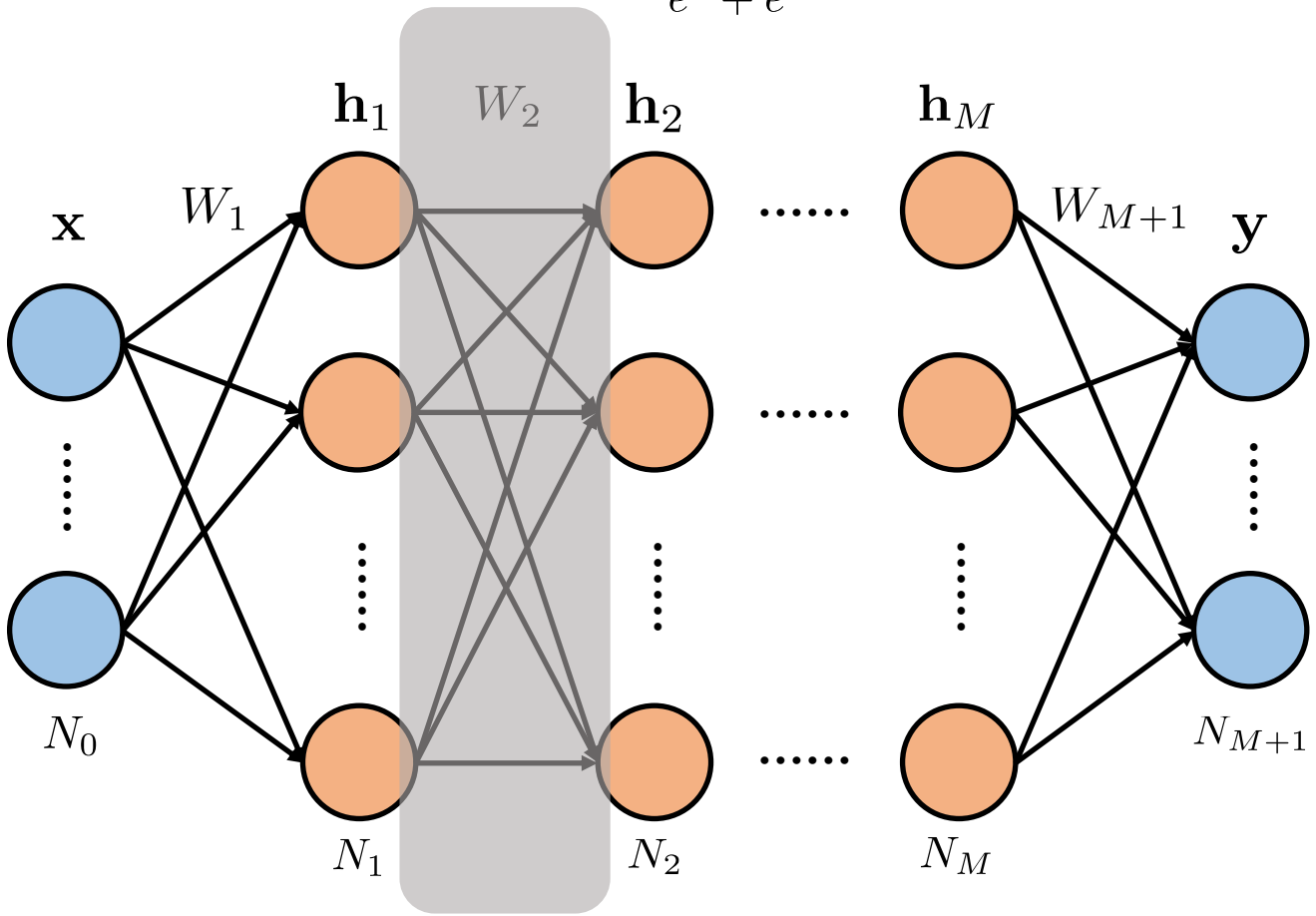
If we assume Tanh activation $\sigma(x) = \dfrac{e^x - e^{-x}}{e^x + e^{-x}}$ and we are in the linear regime

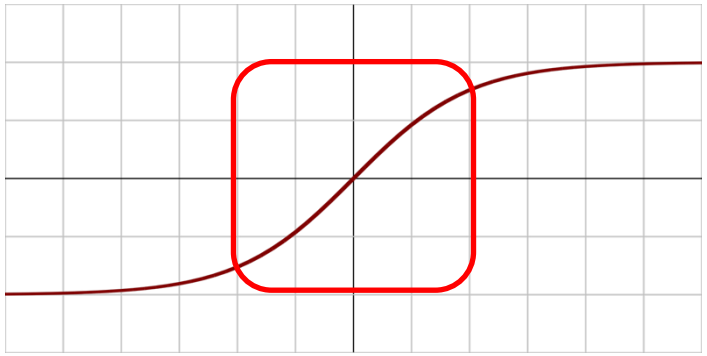


Initialization: Forward Analysis

Recall $\mathbf{h}_2 = \sigma(W_2\mathbf{h}_1)$

If we assume Tanh activation $\sigma(x) = \dfrac{e^x - e^{-x}}{e^x + e^{-x}}$ and we are in the linear regime





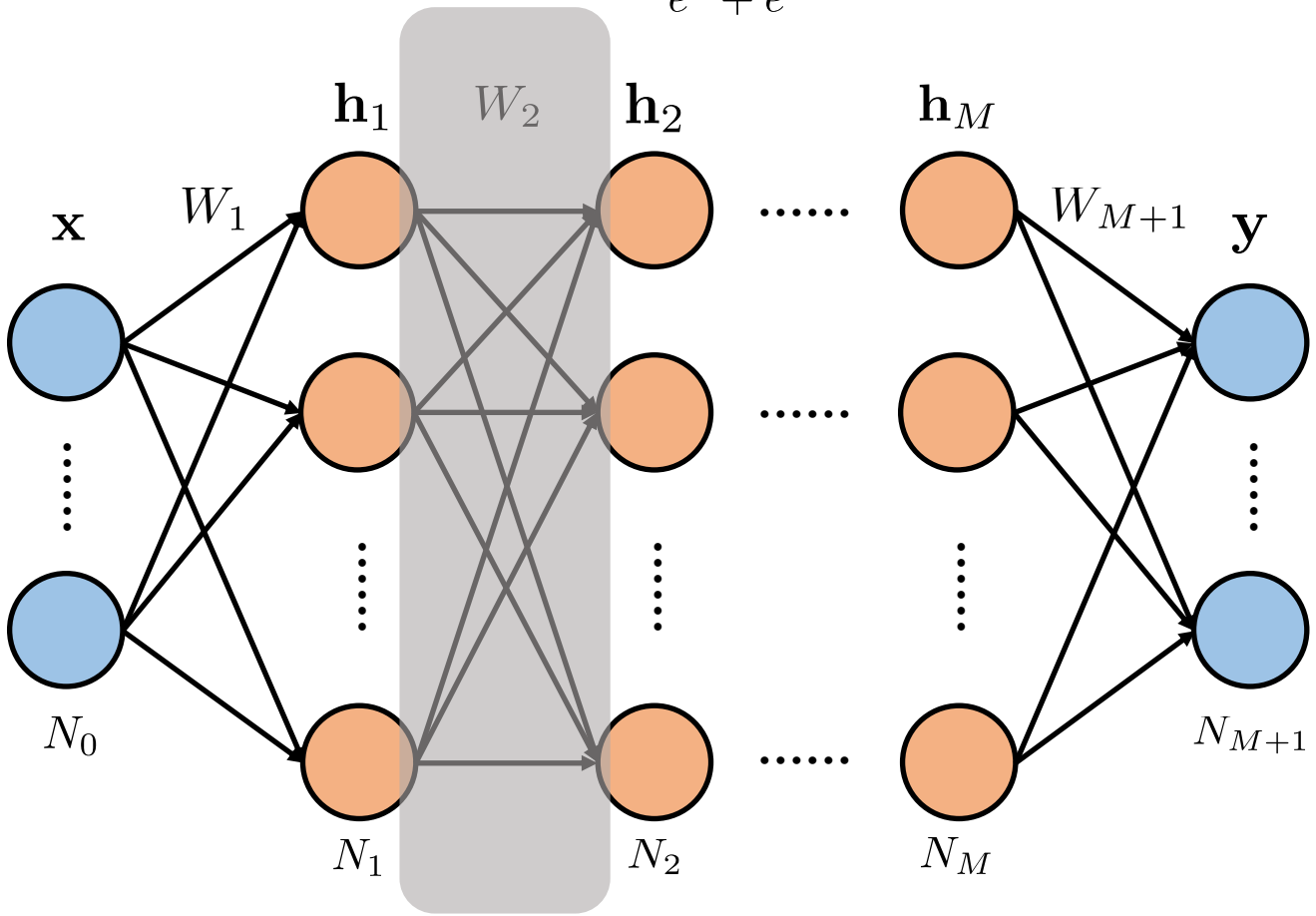
Then we have

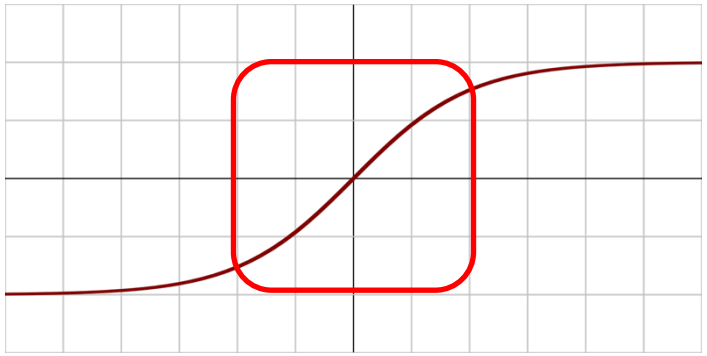
$$\mathbf{h}_2[i] = \sigma(\sum_{j}^{N_1} W_2[i,j]\mathbf{h}_1[j]) \approx \sum_{j=1}^{N_1} W_2[i,j]\mathbf{h}_1[j]$$

Initialization: Forward Analysis

Recall $\mathbf{h}_2 = \sigma(W_2\mathbf{h}_1)$

If we assume Tanh activation $\sigma(x) = \dfrac{e^x - e^{-x}}{e^x + e^{-x}}$ and we are in the linear regime





Then we have

$$\mathbf{h}_2[i] = \sigma(\sum_j^{N_1} W_2[i,j]\mathbf{h}_1[j]) \approx \sum_{j=1}^{N_1} W_2[i,j]\mathbf{h}_1[j]$$

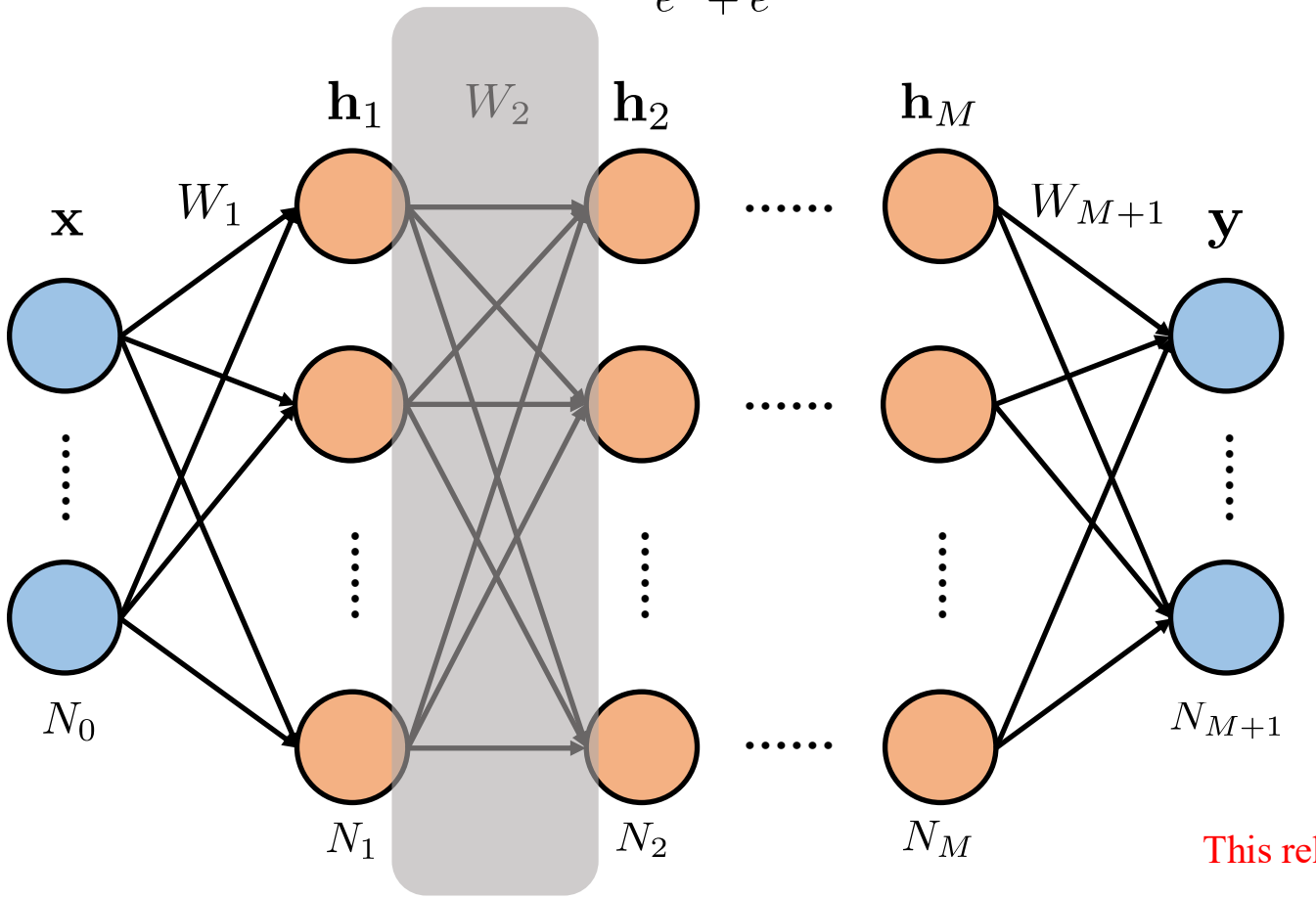
Use the previous independence assumptions:
Equal variances within each layer.

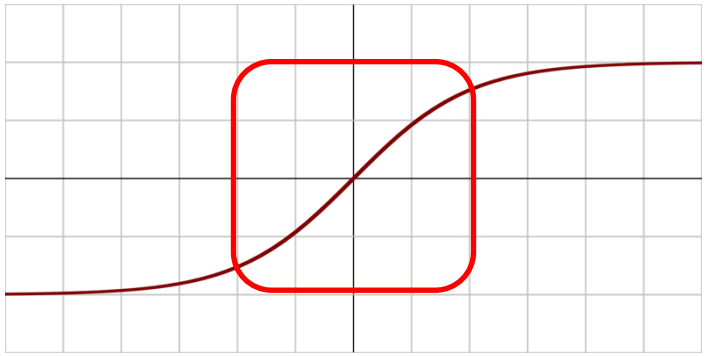
$$\begin{aligned}
\mathbb{V}[\mathbf{h}_2[i]] &\approx \mathbb{V}\left[\sum_{j=1}^{N_1} W_2[i,j]\mathbf{h}_1[j]\right] \\
&= \sum_{j=1}^{N_1} \mathbb{V}[W_2[i,j]\mathbf{h}_1[j]] \\
&= \sum_{j=1}^{N_1} \mathbb{V}[W_2[i,j]]\,\mathbb{V}[\mathbf{h}_1[j]] \\
&= N_1\mathbb{V}[W_2[i,j]]\,\mathbb{V}[\mathbf{h}_1[j]]
\end{aligned}$$

Initialization: Forward Analysis

Recall $\mathbf{h}_2 = \sigma(W_2\mathbf{h}_1)$

If we assume Tanh activation $\sigma(x) = \dfrac{e^x - e^{-x}}{e^x + e^{-x}}$ and we are in the linear regime





Then we have

$$\mathbf{h}_2[i] = \sigma(\sum_j^{N_1} W_2[i,j]\mathbf{h}_1[j]) \approx \sum_{j=1}^{N_1} W_2[i,j]\mathbf{h}_1[j]$$

Use the previous independence assumptions:
Equal variances within each layer.

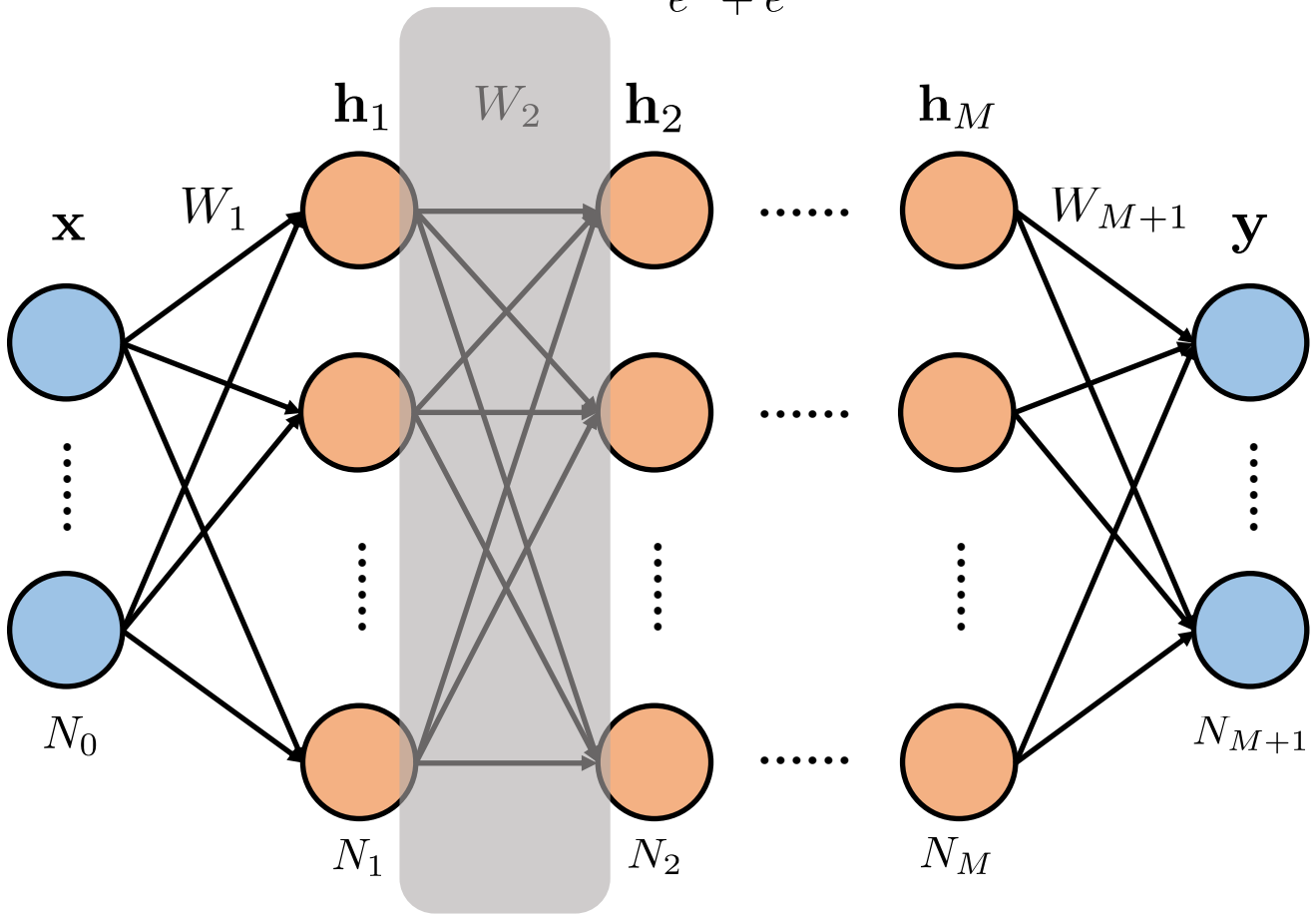
$$\begin{aligned}
\mathbb{V}[\mathbf{h}_2[i]] &\approx \mathbb{V}\left[\sum_{j=1}^{N_1} W_2[i,j]\mathbf{h}_1[j]\right] \\
&= \sum_{j=1}^{N_1} \mathbb{V}[W_2[i,j]\mathbf{h}_1[j]] \\
&= \sum_{j=1}^{N_1} \mathbb{V}[W_2[i,j]]\,\mathbb{V}[\mathbf{h}_1[j]] \\
&= N_1 \mathbb{V}[W_2[i,j]]\,\mathbb{V}[\mathbf{h}_1[j]]
\end{aligned}$$

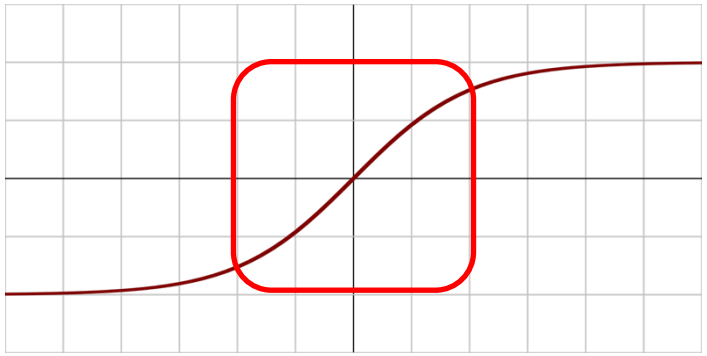
This relation holds for all layers given the assumptions!

Initialization: Forward Analysis

Recall $\mathbf{h}_2 = \sigma(W_2\mathbf{h}_1)$

If we assume Tanh activation $\sigma(x) = \dfrac{e^x - e^{-x}}{e^x + e^{-x}}$ and we are in the linear regime





Then unroll the recursion, we have

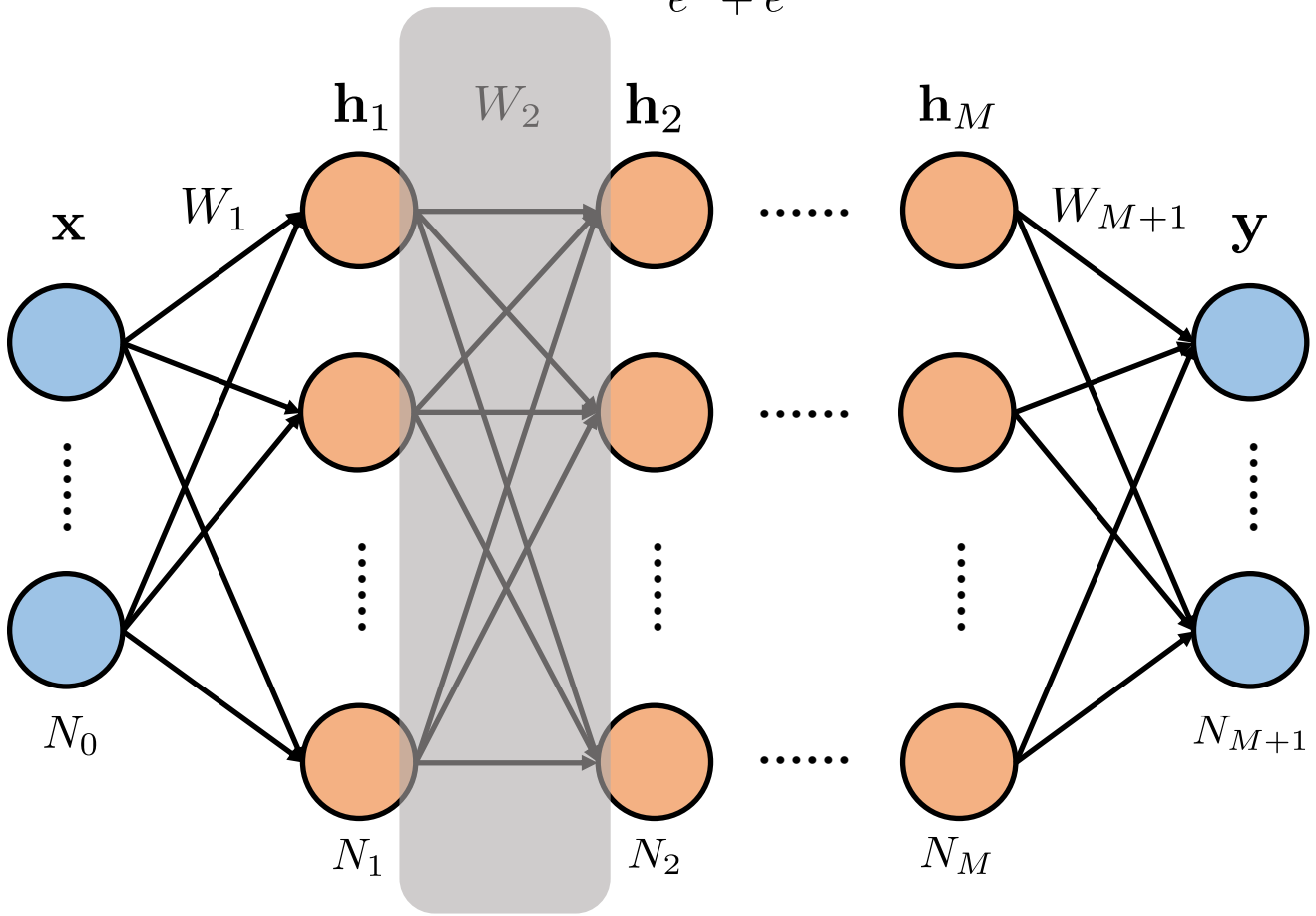
$$\mathbb{V}[\mathbf{h}_l[i]] \approx \mathbb{V}[\mathbf{x}[j]] \prod_{k=1}^{l} N_{k-1}\mathbb{V}[W_k[i,j]]$$

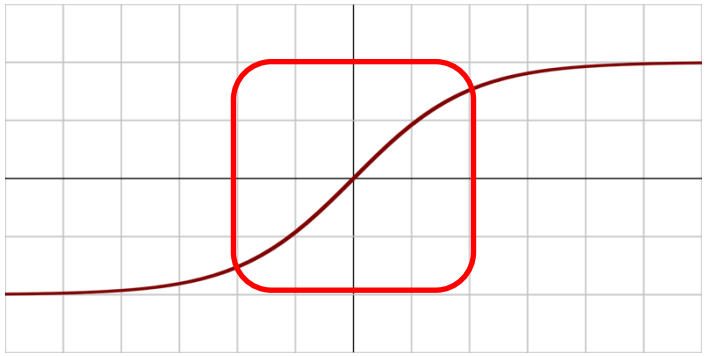
Note i and j here are arbitrary since we assume all activations have the same variance and all weights have the same variance as well!

Initialization: Forward Analysis

Recall $\mathbf{h}_2 = \sigma(W_2\mathbf{h}_1)$

If we assume Tanh activation $\sigma(x) = \dfrac{e^x - e^{-x}}{e^x + e^{-x}}$ and we are in the linear regime





Then unroll the recursion, we have

$$\mathbb{V}[\mathbf{h}_l[i]] \approx \mathbb{V}[\mathbf{x}[j]] \prod_{k=1}^{l} N_{k-1}\mathbb{V}[W_k[i,j]]$$

Within each layer, units share a variance in this approximation. Different layers may use different weight variances.

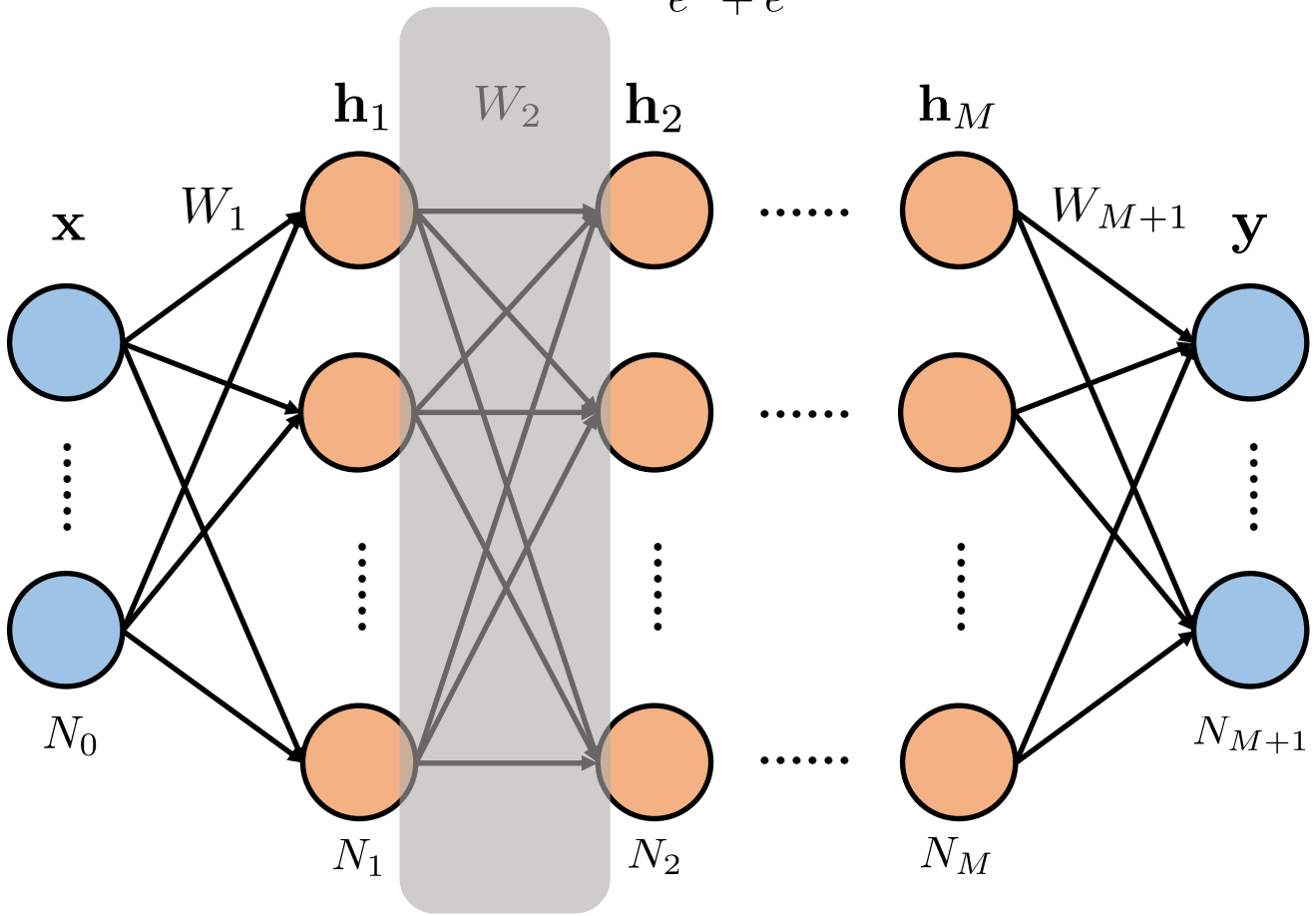
To preserve the variance of activations through forward pass, i.e.,

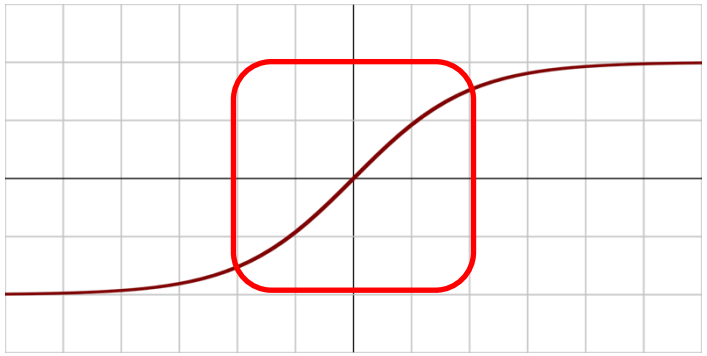
$$\mathbb{V}[\mathbf{h}_l[i]] \approx \mathbb{V}[\mathbf{x}[j]]$$

Initialization: Forward Analysis

Recall $\mathbf{h}_2 = \sigma(W_2\mathbf{h}_1)$

If we assume Tanh activation $\sigma(x) = \dfrac{e^x - e^{-x}}{e^x + e^{-x}}$ and we are in the linear regime





Then unroll the recursion, we have

$$\mathbb{V}[\mathbf{h}_l[i]] \approx \mathbb{V}[\mathbf{x}[j]] \prod_{k=1}^{l} N_{k-1}\mathbb{V}[W_k[i,j]]$$

Within each layer, units share a variance in this approximation. Different layers may use different weight variances.

To preserve the variance of activations through forward pass, i.e.,

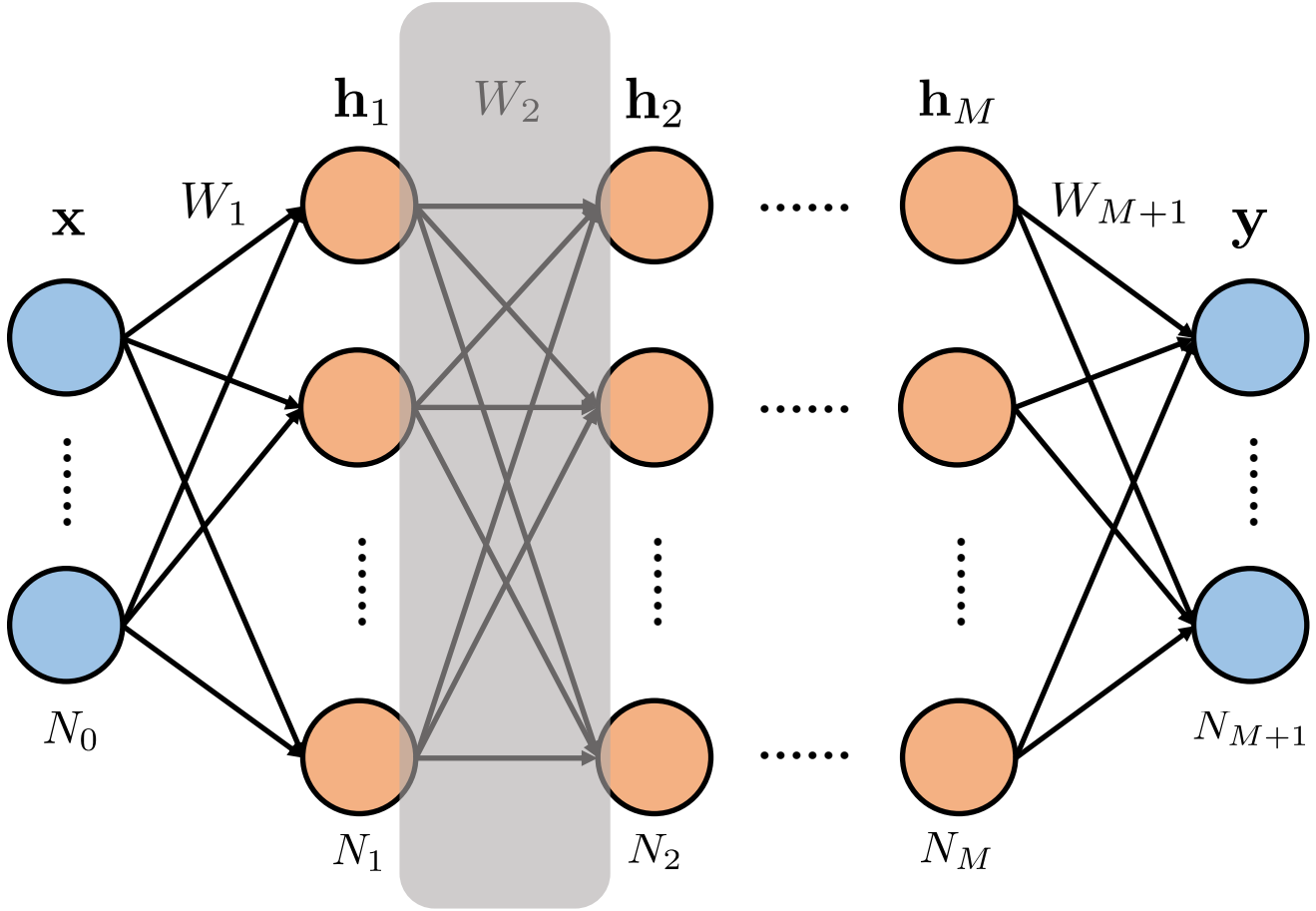
$$\mathbb{V}[\mathbf{h}_l[i]] \approx \mathbb{V}[\mathbf{x}[j]]$$

$$\mathbb{V}[W_k[i,j]] = \frac{1}{N_{k-1}}$$

Initialization: Backward Analysis

Assuming Tanh activation $\sigma(x) = \dfrac{e^x - e^{-x}}{e^x + e^{-x}}$ and we are in the linear regime $\mathbf{h}_2[i] = \sigma(\sum_j^{N_1} W_2[i,j]\mathbf{h}_1[j]) \approx \sum_{j=1}^{N_1} W_2[i,j]\mathbf{h}_1[j]$

Let us look at the gradient w.r.t. activations



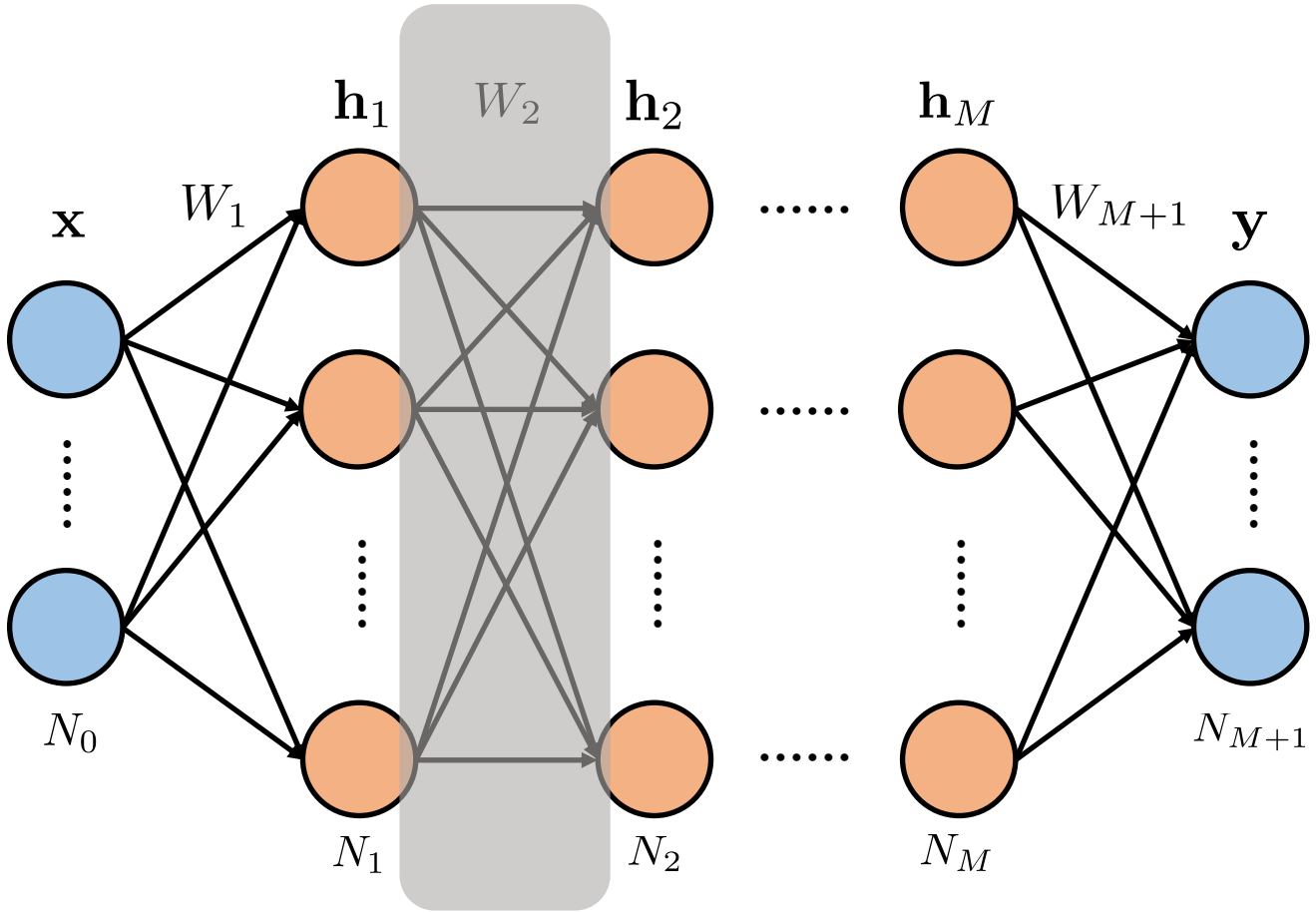
Initialization: Backward Analysis

Assuming Tanh activation $\sigma(x) = \dfrac{e^x - e^{-x}}{e^x + e^{-x}}$ and we are in the linear regime

$$\mathbf{h}_2[i] = \sigma(\sum_j^{N_1} W_2[i,j]\mathbf{h}_1[j]) \approx \sum_{j=1}^{N_1} W_2[i,j]\mathbf{h}_1[j]$$

Let us look at the gradient w.r.t. activations

$$\frac{\partial L}{\partial \mathbf{h}_1[j]} = \sum_{i=1}^{N_2} \frac{\partial L}{\partial \mathbf{h}_2[i]} \frac{\partial \mathbf{h}_2[i]}{\partial \mathbf{h}_1[j]} \approx \sum_{i=1}^{N_2} \frac{\partial L}{\partial \mathbf{h}_2[i]} W_2[i,j]$$



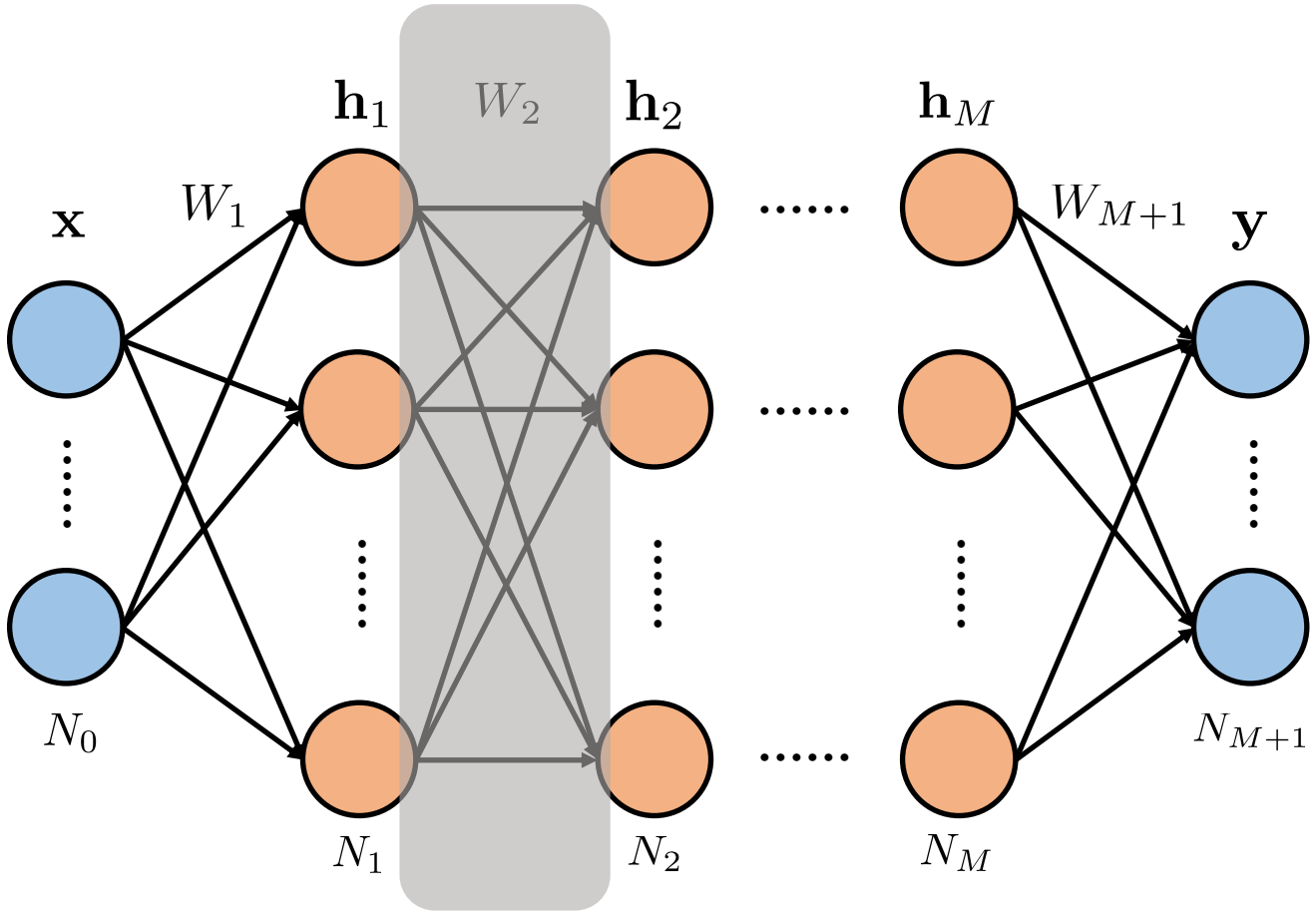
Initialization: Backward Analysis

Assuming Tanh activation $\sigma(x) = \dfrac{e^x - e^{-x}}{e^x + e^{-x}}$ and we are in the linear regime

$$\mathbf{h}_2[i] = \sigma(\sum_j^{N_1} W_2[i,j]\mathbf{h}_1[j]) \approx \sum_{j=1}^{N_1} W_2[i,j]\mathbf{h}_1[j]$$

Let us look at the gradient w.r.t. activations

$$\frac{\partial L}{\partial \mathbf{h}_1[j]} = \sum_{i=1}^{N_2} \frac{\partial L}{\partial \mathbf{h}_2[i]} \frac{\partial \mathbf{h}_2[i]}{\partial \mathbf{h}_1[j]} \approx \sum_{i=1}^{N_2} \frac{\partial L}{\partial \mathbf{h}_2[i]} W_2[i,j]$$



Use the previous independence assumptions:

Equal variances within each layer.

$$\begin{aligned}
\mathbb{V}\left[\frac{\partial L}{\partial \mathbf{h}_1[j]}\right] &\approx \sum_{i=1}^{N_2} \mathbb{V}\left[\frac{\partial L}{\partial \mathbf{h}_2[i]}\right] \mathbb{V}\left[W_2[i,j]\right] \\
&= N_2 \mathbb{V}\left[\frac{\partial L}{\partial \mathbf{h}_2[i]}\right] \mathbb{V}\left[W_2[i,j]\right] \\
&= \mathbb{V}\left[\frac{\partial L}{\partial \mathbf{y}[i]}\right] \prod_{k=2}^{M+1} N_k \mathbb{V}\left[W_k[i,j]\right]
\end{aligned}$$

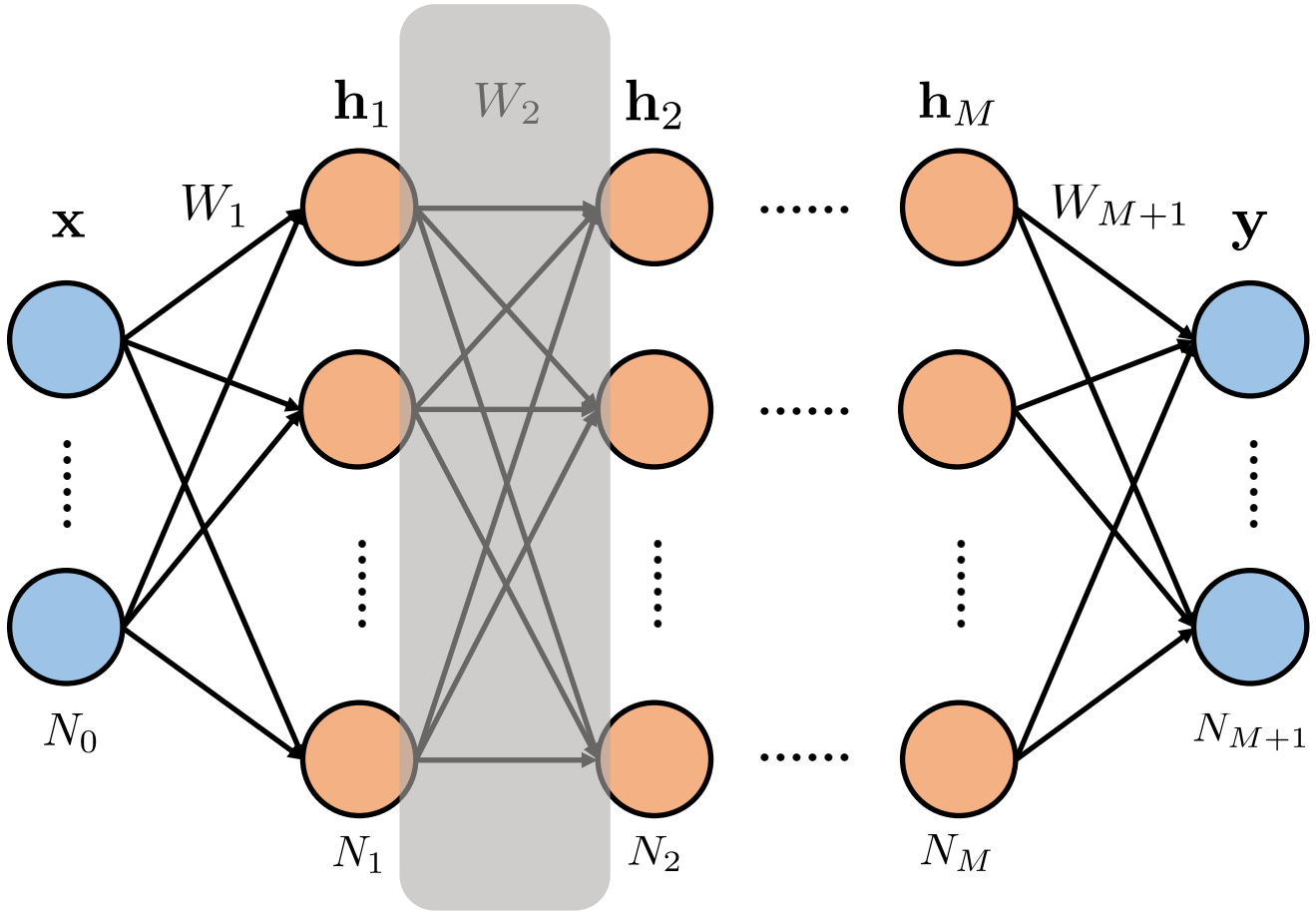
Initialization: Backward Analysis

Assuming Tanh activation $\sigma(x) = \dfrac{e^x - e^{-x}}{e^x + e^{-x}}$ and we are in the linear regime

$$\mathbf{h}_2[i] = \sigma(\sum_j^{N_1} W_2[i,j]\mathbf{h}_1[j]) \approx \sum_{j=1}^{N_1} W_2[i,j]\mathbf{h}_1[j]$$

Let us look at the gradient w.r.t. activations

$$\frac{\partial L}{\partial \mathbf{h}_1[j]} = \sum_{i=1}^{N_2} \frac{\partial L}{\partial \mathbf{h}_2[i]} \frac{\partial \mathbf{h}_2[i]}{\partial \mathbf{h}_1[j]} \approx \sum_{i=1}^{N_2} \frac{\partial L}{\partial \mathbf{h}_2[i]} W_2[i,j]$$



Use the previous independence assumptions:

Equal variances within each layer.

$$\begin{aligned}
\mathbb{V}\left[\frac{\partial L}{\partial \mathbf{h}_1[j]}\right] &\approx \sum_{i=1}^{N_2} \mathbb{V}\left[\frac{\partial L}{\partial \mathbf{h}_2[i]}\right] \mathbb{V}\left[W_2[i,j]\right] \\
&= N_2 \mathbb{V}\left[\frac{\partial L}{\partial \mathbf{h}_2[i]}\right] \mathbb{V}\left[W_2[i,j]\right] \\
&= \mathbb{V}\left[\frac{\partial L}{\partial \mathbf{y}[i]}\right] \prod_{k=2}^{M+1} N_k \mathbb{V}\left[W_k[i,j]\right]
\end{aligned}$$

Setting $\mathbb{V}\left[W_k[i,j]\right] = \dfrac{1}{N_k}$ preserves the variance of gradients w.r.t. activations!

Initialization: Backward Analysis

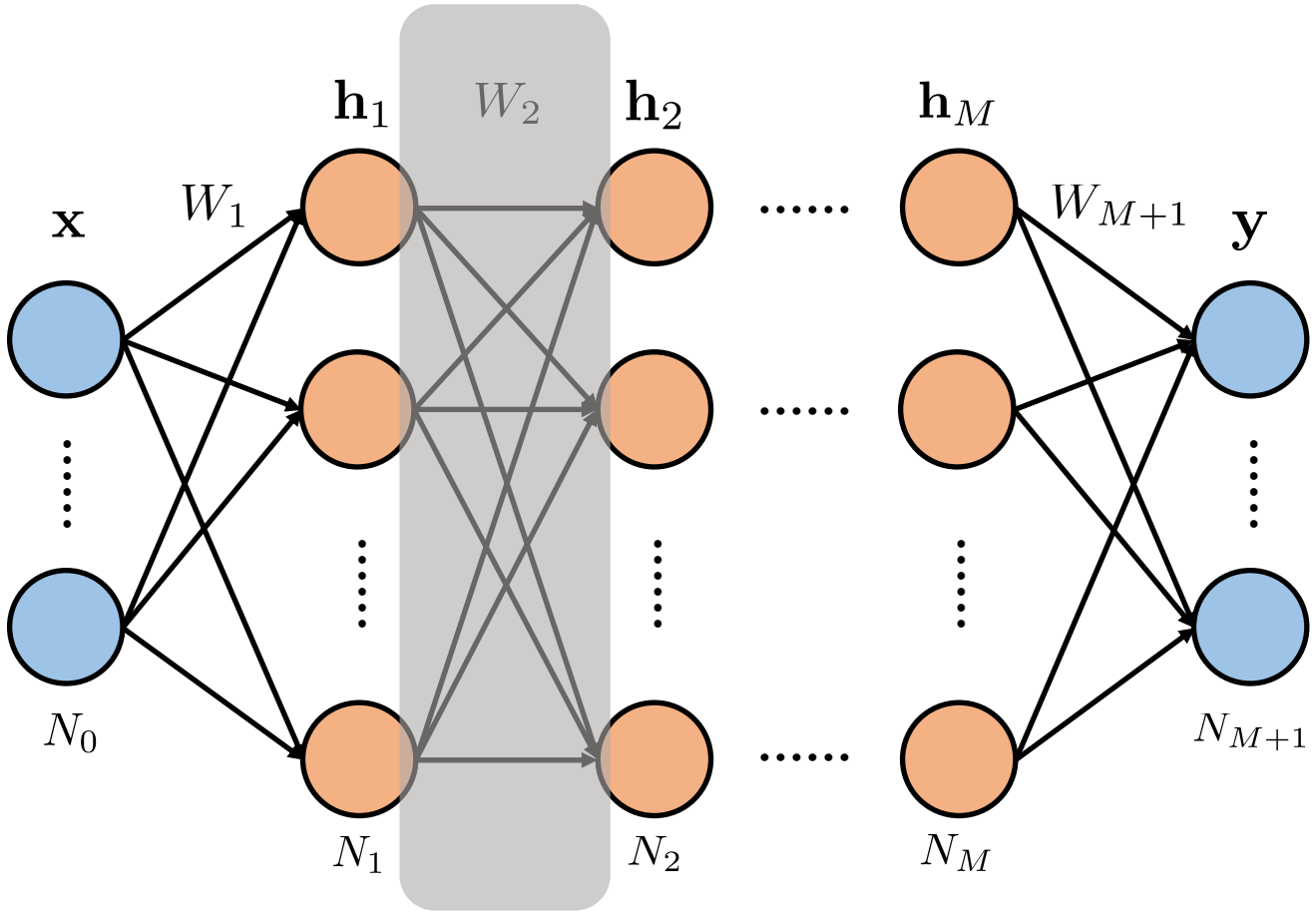
Assuming Tanh activation $\sigma(x) = \dfrac{e^x - e^{-x}}{e^x + e^{-x}}$ and we are in the linear regime

$$\mathbf{h}_2[i] = \sigma(\sum_j^{N_1} W_2[i,j]\mathbf{h}_1[j]) \approx \sum_{j=1}^{N_1} W_2[i,j]\mathbf{h}_1[j]$$

Let us look at the gradient w.r.t. activations

$$\frac{\partial L}{\partial \mathbf{h}_1[j]} = \sum_{i=1}^{N_2} \frac{\partial L}{\partial \mathbf{h}_2[i]} \frac{\partial \mathbf{h}_2[i]}{\partial \mathbf{h}_1[j]} \approx \sum_{i=1}^{N_2} \frac{\partial L}{\partial \mathbf{h}_2[i]} W_2[i,j]$$

What about the gradients w.r.t. weights?



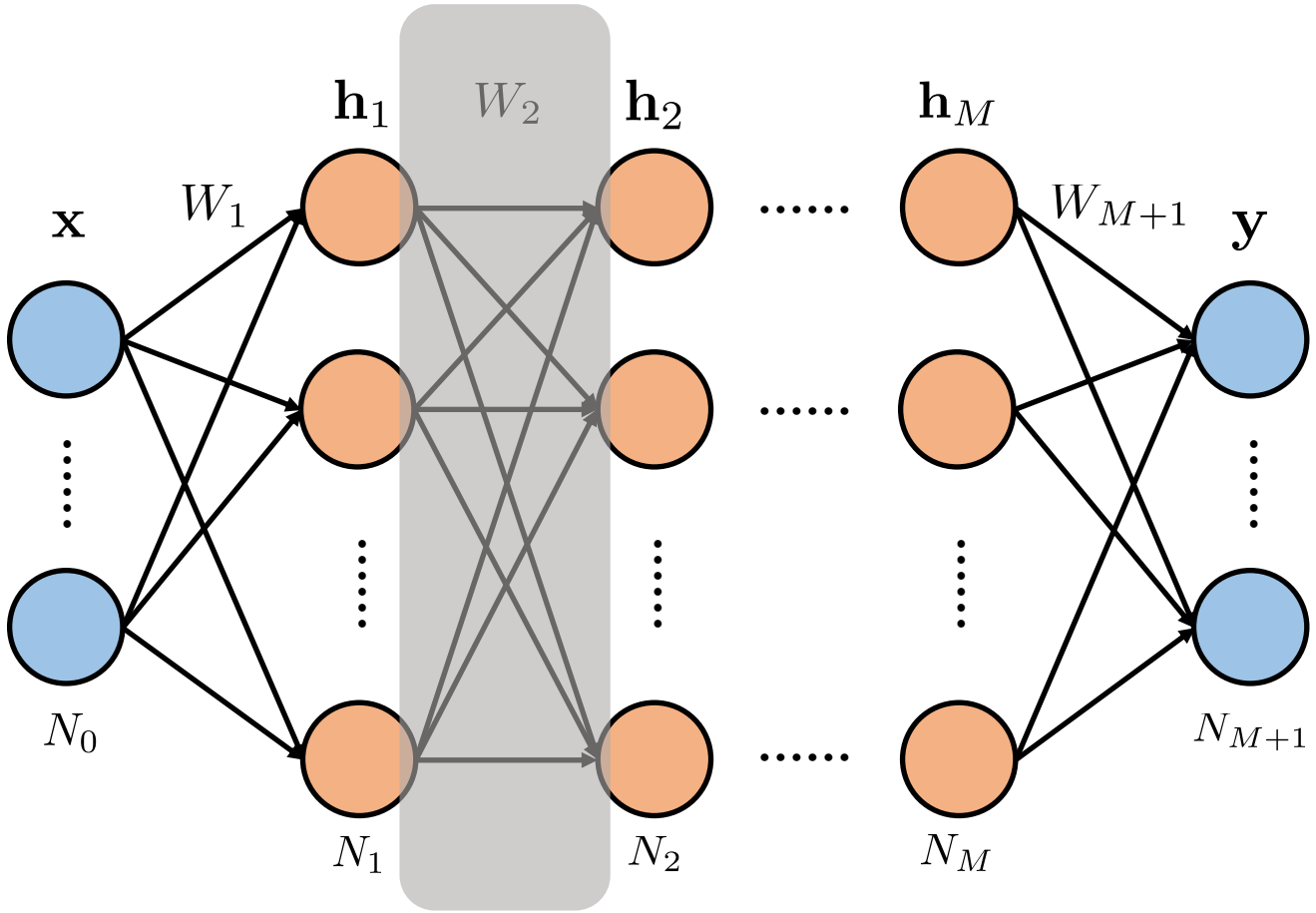
Initialization: Backward Analysis

Assuming Tanh activation $\sigma(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ and we are in the linear regime

$$\mathbf{h}_2[i] = \sigma(\sum_j^{N_1} W_2[i,j]\mathbf{h}_1[j]) \approx \sum_{j=1}^{N_1} W_2[i,j]\mathbf{h}_1[j]$$

Let us look at the gradient w.r.t. activations

$$\frac{\partial L}{\partial \mathbf{h}_1[j]} = \sum_{i=1}^{N_2} \frac{\partial L}{\partial \mathbf{h}_2[i]} \frac{\partial \mathbf{h}_2[i]}{\partial \mathbf{h}_1[j]} \approx \sum_{i=1}^{N_2} \frac{\partial L}{\partial \mathbf{h}_2[i]} W_2[i,j]$$



What about the gradients w.r.t. weights?

$$\frac{\partial L}{\partial W_2[i,j]} = \frac{\partial L}{\partial \mathbf{h}_2[i]} \frac{\partial \mathbf{h}_2[i]}{\partial W_2[i,j]} \approx \frac{\partial L}{\partial \mathbf{h}_2[i]} \mathbf{h}_1[j]$$

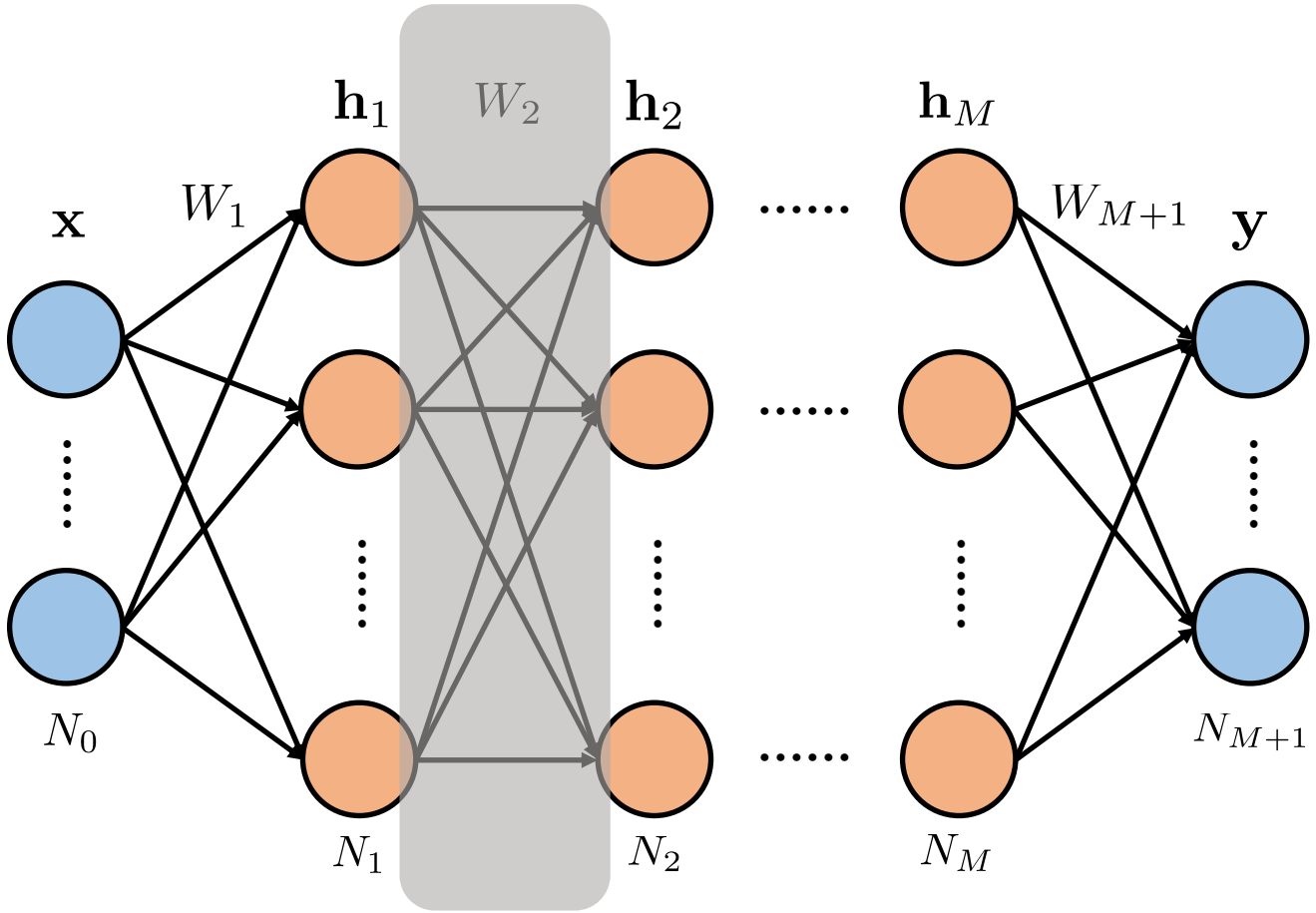
Initialization: Backward Analysis

Assuming Tanh activation $\sigma(x) = \dfrac{e^x - e^{-x}}{e^x + e^{-x}}$ and we are in the linear regime

$$\mathbf{h}_2[i] = \sigma(\sum_j^{N_1} W_2[i,j]\mathbf{h}_1[j]) \approx \sum_{j=1}^{N_1} W_2[i,j]\mathbf{h}_1[j]$$

Let us look at the gradient w.r.t. activations

$$\frac{\partial L}{\partial \mathbf{h}_1[j]} = \sum_{i=1}^{N_2} \frac{\partial L}{\partial \mathbf{h}_2[i]} \frac{\partial \mathbf{h}_2[i]}{\partial \mathbf{h}_1[j]} \approx \sum_{i=1}^{N_2} \frac{\partial L}{\partial \mathbf{h}_2[i]} W_2[i,j]$$



What about the gradients w.r.t. weights?

$$\frac{\partial L}{\partial W_2[i,j]} = \frac{\partial L}{\partial \mathbf{h}_2[i]} \frac{\partial \mathbf{h}_2[i]}{\partial W_2[i,j]} \approx \frac{\partial L}{\partial \mathbf{h}_2[i]} \mathbf{h}_1[j]$$

$$\begin{aligned}
\mathbb{V}\left[\frac{\partial L}{\partial W_2[i,j]}\right] &\approx \mathbb{V}\left[\frac{\partial L}{\partial \mathbf{h}_2[i]}\right] \mathbb{V}\left[\mathbf{h}_1[j]\right] \\
&= \left(\mathbb{V}\left[\frac{\partial L}{\partial \mathbf{y}[i]}\right] \prod_{k=3}^{M+1} N_k \mathbb{V}\left[W_k[i,j]\right]\right) \\
&\quad \left(\mathbb{V}\left[\mathbf{x}[j]\right] \prod_{k=1}^{1} N_{k-1} \mathbb{V}\left[W_k[i,j]\right]\right) \\
&= \frac{N_0}{N_1} \left(\prod_{k=1}^{1} N_k \mathbb{V}\left[W_k[i,j]\right]\right) \left(\prod_{k=3}^{M+1} N_k \mathbb{V}\left[W_k[i,j]\right]\right) \mathbb{V}\left[\frac{\partial L}{\partial \mathbf{y}[i]}\right] \mathbb{V}\left[\mathbf{x}[j]\right] \\
&= \frac{N_0}{N_1} \mathbb{V}\left[\frac{\partial L}{\partial \mathbf{y}[i]}\right] \mathbb{V}\left[\mathbf{x}[j]\right]
\end{aligned}$$

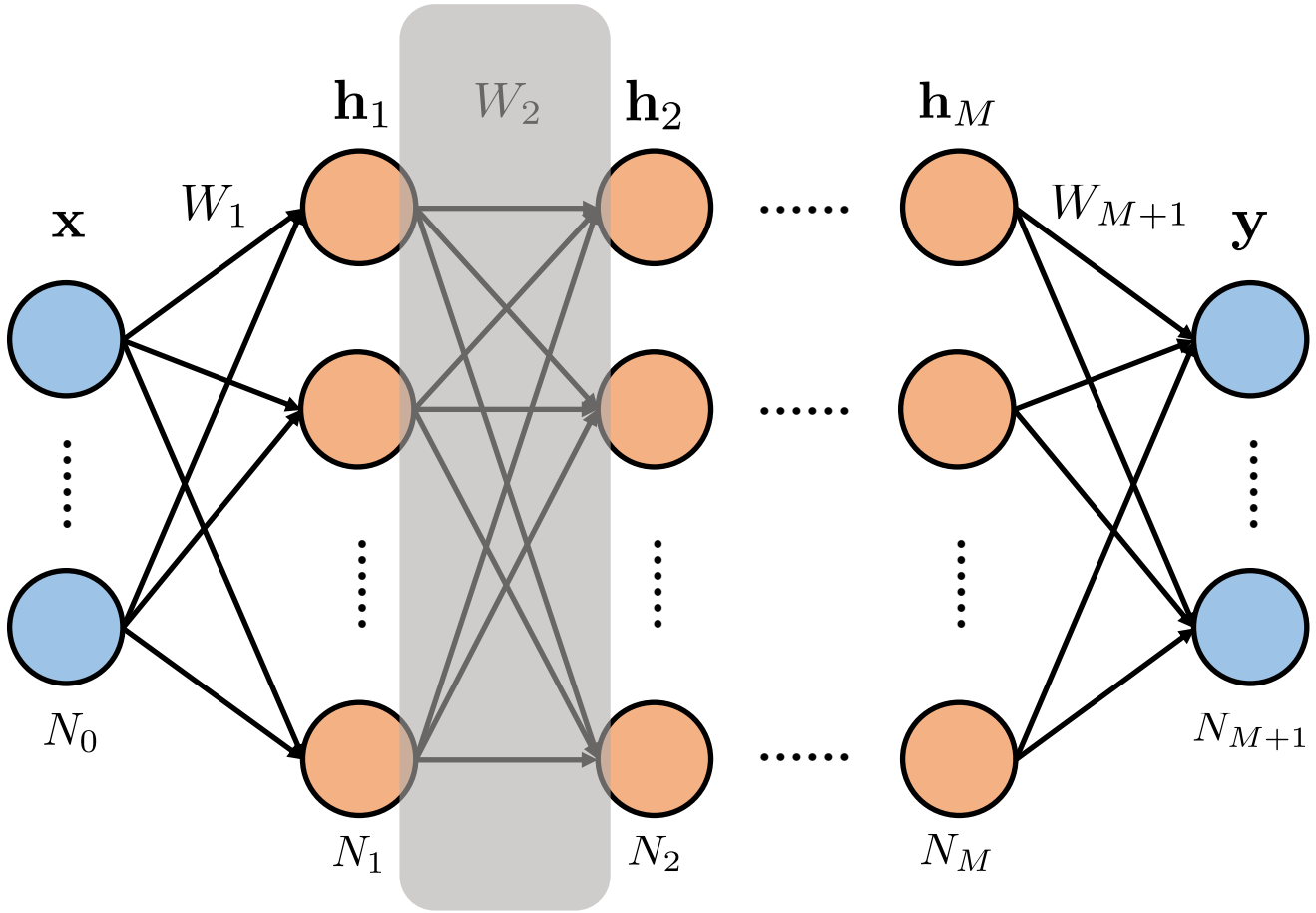
Initialization: Backward Analysis

Assuming Tanh activation $\sigma(x) = \dfrac{e^x - e^{-x}}{e^x + e^{-x}}$ and we are in the linear regime

$$\mathbf{h}_2[i] = \sigma(\sum_j^{N_1} W_2[i,j]\mathbf{h}_1[j]) \approx \sum_{j=1}^{N_1} W_2[i,j]\mathbf{h}_1[j]$$

Let us look at the gradient w.r.t. activations

$$\frac{\partial L}{\partial \mathbf{h}_1[j]} = \sum_{i=1}^{N_2} \frac{\partial L}{\partial \mathbf{h}_2[i]} \frac{\partial \mathbf{h}_2[i]}{\partial \mathbf{h}_1[j]} \approx \sum_{i=1}^{N_2} \frac{\partial L}{\partial \mathbf{h}_2[i]} W_2[i,j]$$



What about the gradients w.r.t. weights?

$$\frac{\partial L}{\partial W_2[i,j]} = \frac{\partial L}{\partial \mathbf{h}_2[i]} \frac{\partial \mathbf{h}_2[i]}{\partial W_2[i,j]} \approx \frac{\partial L}{\partial \mathbf{h}_2[i]} \mathbf{h}_1[j]$$

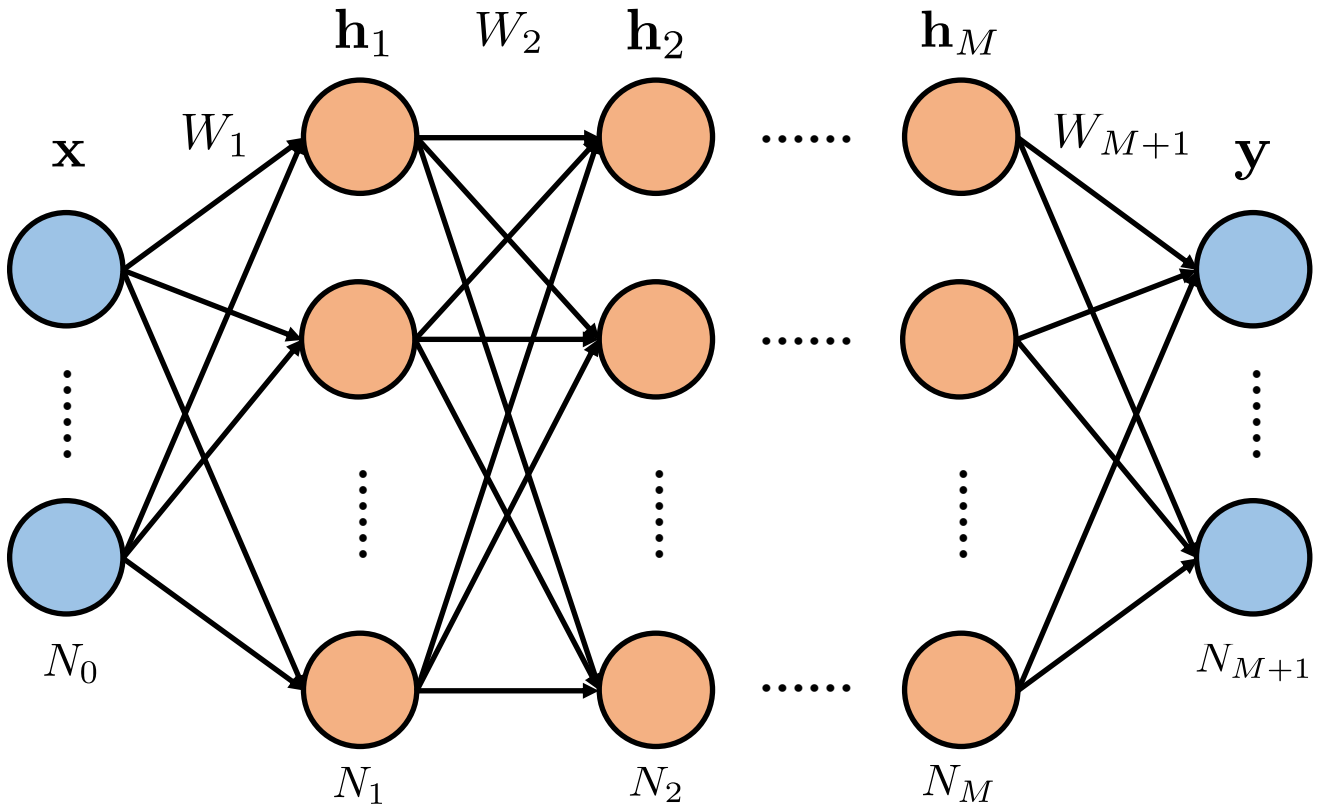
$$\begin{aligned}
\mathbb{V}\left[\frac{\partial L}{\partial W_2[i,j]}\right] &\approx \mathbb{V}\left[\frac{\partial L}{\partial \mathbf{h}_2[i]}\right] \mathbb{V}\left[\mathbf{h}_1[j]\right] \\
&= \left(\mathbb{V}\left[\frac{\partial L}{\partial \mathbf{y}[i]}\right] \prod_{k=3}^{M+1} N_k \mathbb{V}\left[W_k[i,j]\right]\right) \\
&\quad \left(\mathbb{V}\left[\mathbf{x}[j]\right] \prod_{k=1}^{1} N_{k-1} \mathbb{V}\left[W_k[i,j]\right]\right) \\
&= \frac{N_0}{N_1} \left(\prod_{k=1}^{1} N_k \mathbb{V}\left[W_k[i,j]\right]\right) \left(\prod_{k=3}^{M+1} N_k \mathbb{V}\left[W_k[i,j]\right]\right) \mathbb{V}\left[\frac{\partial L}{\partial \mathbf{y}[i]}\right] \mathbb{V}\left[\mathbf{x}[j]\right] \\
&= \frac{N_0}{N_1} \mathbb{V}\left[\frac{\partial L}{\partial \mathbf{y}[i]}\right] \mathbb{V}\left[\mathbf{x}[j]\right]
\end{aligned}$$

Setting $\mathbb{V}\left[W_k[i,j]\right] = \dfrac{1}{N_k}$ controls the variance of weight gradients in this approximation.

Initialization

In summary, to preserve the variance of activations, we set $\mathbb{V}\left[W_k[i,j]\right] = \frac{1}{N_{k-1}}$

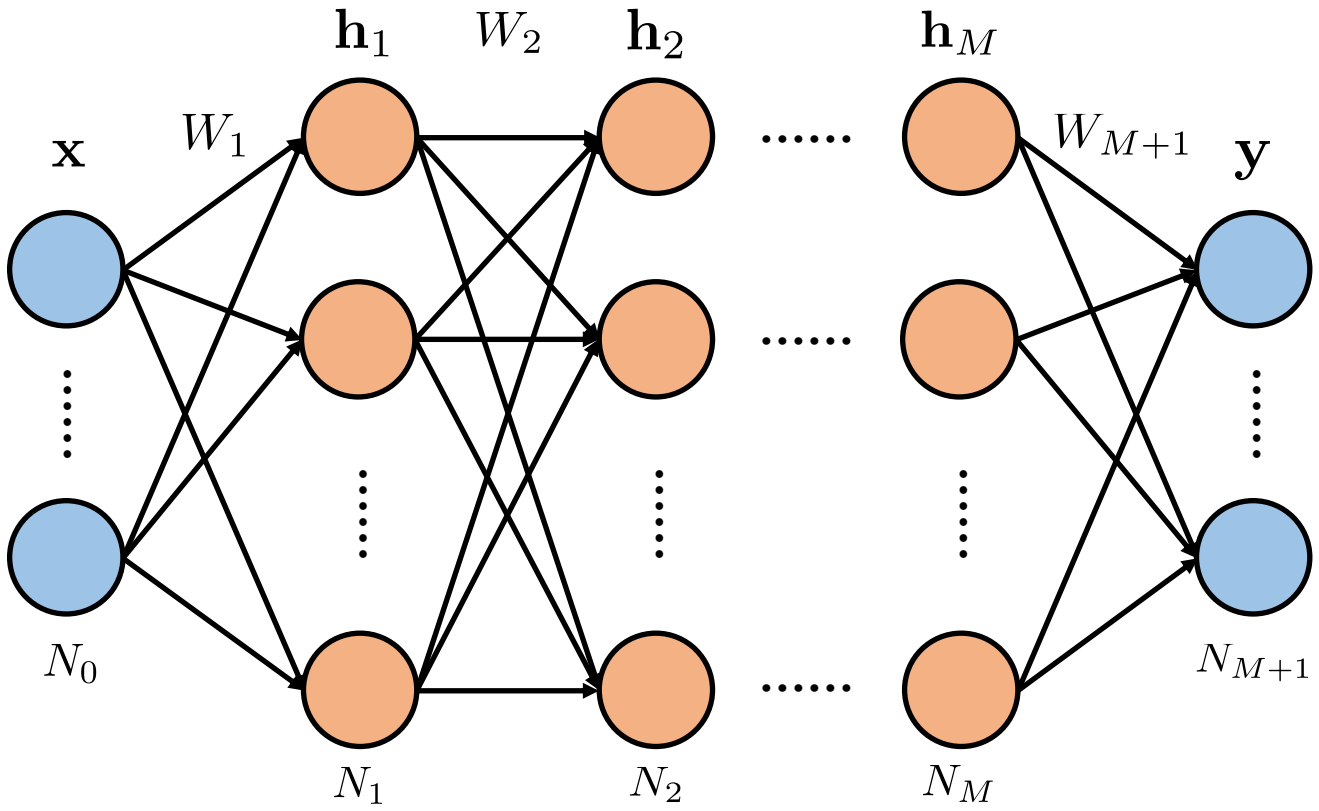
to preserve the variance of gradients w.r.t. activations, we set $\mathbb{V}\left[W_k[i,j]\right] = \frac{1}{N_k}$



Initialization

In summary, to preserve the variance of activations, we set $\mathbb{V}\left[W_k[i,j]\right] = \frac{1}{N_{k-1}}$

to preserve the variance of gradients w.r.t. activations, we set $\mathbb{V}\left[W_k[i,j]\right] = \frac{1}{N_k}$



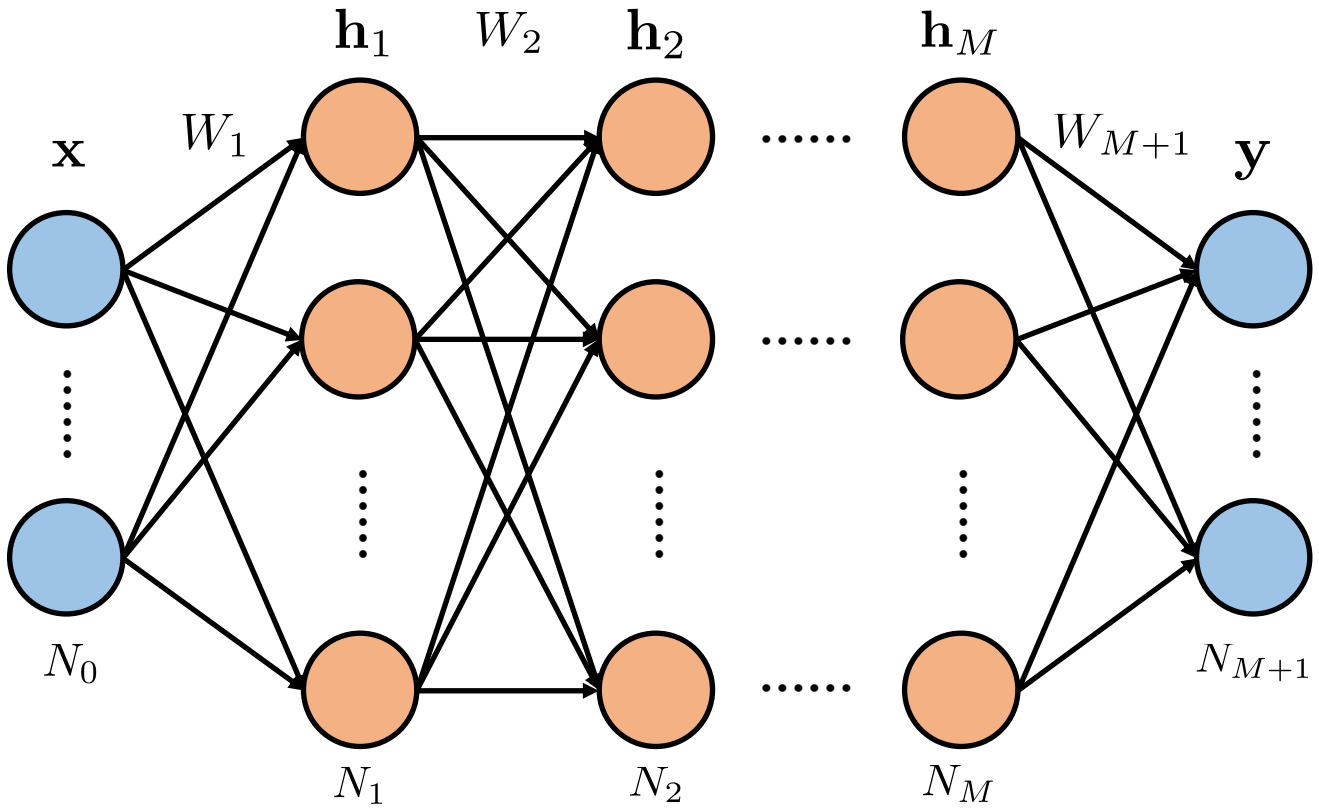
To compromise between two goals, we can take the mean of the denominators:

$$\mathbb{V}\left[W_k[i,j]\right] = \frac{1}{\frac{N_k+N_{k-1}}{2}} = \frac{2}{N_k+N_{k-1}}$$

Initialization

In summary, to preserve the variance of activations, we set $\mathbb{V}\left[W_k[i,j]\right] = \frac{1}{N_{k-1}}$

to preserve the variance of gradients w.r.t. activations, we set $\mathbb{V}\left[W_k[i,j]\right] = \frac{1}{N_k}$



To compromise between two goals, we can take the mean of the denominators:

$$\mathbb{V}\left[W_k[i,j]\right] = \frac{1}{\frac{N_k+N_{k-1}}{2}} = \frac{2}{N_k+N_{k-1}}$$

This is the so-called “Xavier Initialization” [3]!

Initialization: Uniform Distribution

Choose a symmetric uniform distribution

Sample each weight independently. For one weight:

$$w := W_k[i,j] \sim \mathcal{U}(-a,a), \quad a > 0, \qquad p(w) = \begin{cases} \dfrac{1}{2a}, & -a \le w \le a, \\ 0, & \text{otherwise.} \end{cases}$$

Initialization: Uniform Distribution

Choose a symmetric uniform distribution

Sample each weight independently. For one weight:

$$w := W_k[i,j] \sim \mathcal{U}(-a,a), \quad a > 0, \qquad p(w) = \begin{cases} \dfrac{1}{2a}, & -a \le w \le a, \\ 0, & \text{otherwise.} \end{cases}$$

Zero mean

$$\mathbb{E}[w] = \frac{1}{2a}\int_{-a}^{a} w\,dw = \frac{1}{2a}\left[\frac{w^2}{2}\right]_{-a}^{a} = 0$$

Initialization: Uniform Distribution

Choose a symmetric uniform distribution

Sample each weight independently. For one weight:

$$w := W_k[i,j] \sim \mathcal{U}(-a,a), \quad a > 0, \qquad p(w) = \begin{cases} \dfrac{1}{2a}, & -a \le w \le a, \\ 0, & \text{otherwise.} \end{cases}$$

Zero mean

$$\mathbb{E}[w] = \frac{1}{2a}\int_{-a}^{a} w\,dw = \frac{1}{2a}\left[\frac{w^2}{2}\right]_{-a}^{a} = 0$$

Second moment and variance

$$\mathbb{E}[w^2] = \frac{1}{2a}\int_{-a}^{a} w^2\,dw = \frac{1}{2a}\left[\frac{w^3}{3}\right]_{-a}^{a} = \frac{a^2}{3}$$

$$\mathbb{V}[w] = \mathbb{E}[w^2] - \mathbb{E}[w]^2 = \frac{a^2}{3}$$

Initialization: Xavier Uniform

Match the Xavier variance [3]

$$\mathbb{V}[W_k[i,j]] = \frac{a^2}{3} = \frac{2}{N_{k-1} + N_k}$$

Solving for the positive bound:

$$a^2 = \frac{6}{N_{k-1} + N_k} \qquad \Longrightarrow \qquad a = \sqrt{\frac{6}{N_{k-1} + N_k}} \quad (a > 0)$$

Uniform Xavier initialization

$$W_k[i,j] \sim \mathcal{U}\left(-\sqrt{\frac{6}{N_{k-1} + N_k}}, \sqrt{\frac{6}{N_{k-1} + N_k}}\right)$$

Gaussian and uniform versions satisfy the same target variance.

Initialization in Practice

Xavier / Glorot initialization [3]

$$W_{ij} \sim \mathcal{N}\left(0, \frac{2}{n_{\text{in}} + n_{\text{out}}}\right) \quad \text{or} \quad W_{ij} \sim \mathcal{U}\left(-\sqrt{\frac{6}{n_{\text{in}} + n_{\text{out}}}}, \sqrt{\frac{6}{n_{\text{in}} + n_{\text{out}}}}\right)$$

$$n_{\text{in}} = N_{m-1}, \qquad n_{\text{out}} = N_m$$

Zero bias is a common default. Do not initialize every weight to the same value.

For GELU / SiLU and residual networks, follow the architecture's initialization recipe.

The normal distribution shown uses variance as the second parameter.

References

[1] Robbins, H., & Monro, S. (1951). A stochastic approximation method. The Annals of Mathematical Statistics, 22(3), 400–407.

[2] Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. Nature, 323, 533–536.

[3] Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. AISTATS.

[4] Baydin, A. G., Pearlmutter, B. A., Radul, A. A., & Siskind, J. M. (2018). Automatic differentiation in machine learning: A survey. JMLR, 18(153), 1–43.

Questions?