CPEN 400D: Deep Learning

Lecture 2 (II): Backpropagation

Renjie Liao

University of British Columbia

Winter, Term 2, 2022

Outline

- Learning Algorithm for Feedforward Neural Networks:
  - Backpropagation
  - Weight Initialization
  - Learning Rate & Momentum & Adam
  - Weight Decay & Early Stopping

Learning Algorithm

- Learning algorithms are just optimization algorithms and are about **credit assignment**!

Adjust parameters based on loss ⇔ Assign credits based on contribution

Learning Algorithm

- Learning algorithms are just optimization algorithms and are about **credit assignment**!

  Adjust parameters based on loss ⇔ Assign credits based on contribution

- The most successful learning algorithm in machine learning so far is **gradient based learning**!

  Stochastic gradient descent (SGD) [1], introduced in 1951 by Herbert Robbins and Sutton Monro

Learning Algorithm

- Learning algorithms are just optimization algorithms and are about **credit assignment**!

  Adjust parameters based on loss ⇔ Assign credits based on contribution

- The most successful learning algorithm in machine learning so far is **gradient based learning**!

  Stochastic gradient descent (SGD) [1], introduced in 1951 by Herbert Robbins and Sutton Monro

- Back-propagation (BP) = an efficient SGD in the context of deep learning

  - BP has been independently discovered many times (see the history of deep learning in the 1st lecture)
  - BP was first shown to successfully train neural networks and learn useful representations in 1986 [2] by David Rumelhart, Geoffrey Hinton, and Ronald Williams
  - BP is the most successful learning algorithm so far for training feedforward neural networks

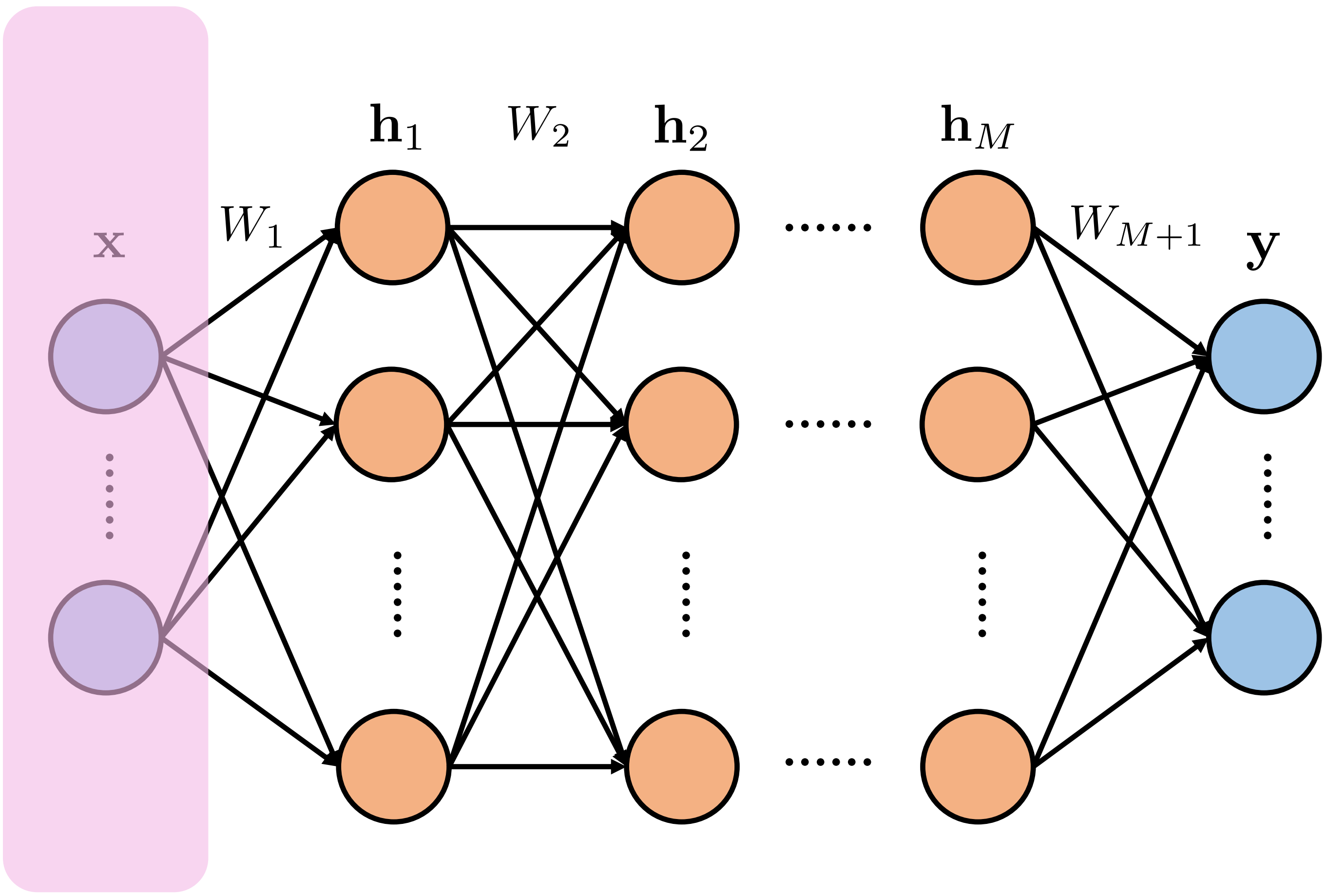
Outline

- Learning Algorithm for Feedforward Neural Networks:
    - **Backpropagation**
    - Weight Initialization
    - Learning Rate & Momentum & Adam
    - Weight Decay & Early Stopping

Feedforward Neural Networks

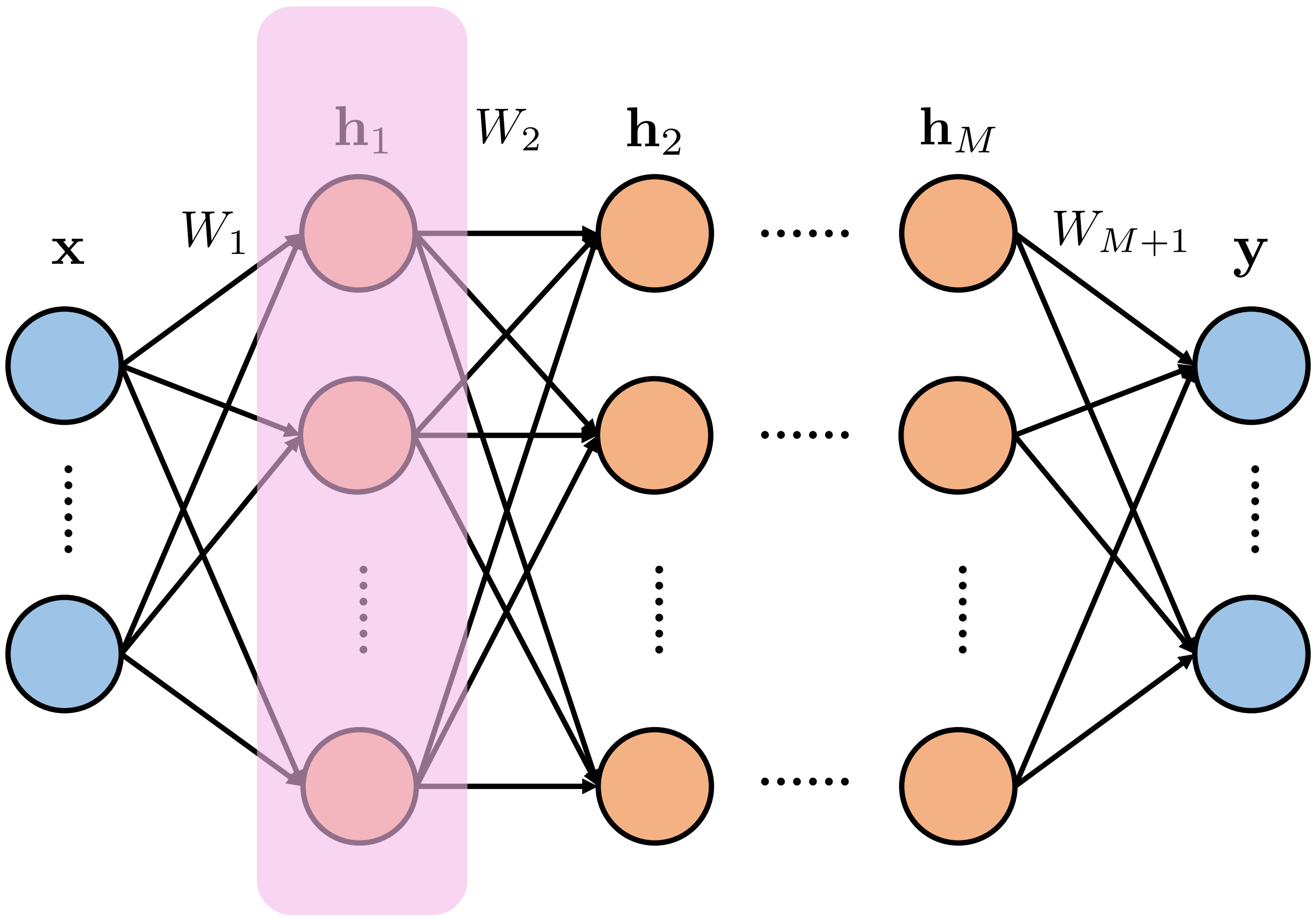
Consider a MLP as follows. Recall what we do in the forward pass:

Input Sample



Feedforward Neural Networks

Consider a MLP as follows. Recall what we do in the forward pass:

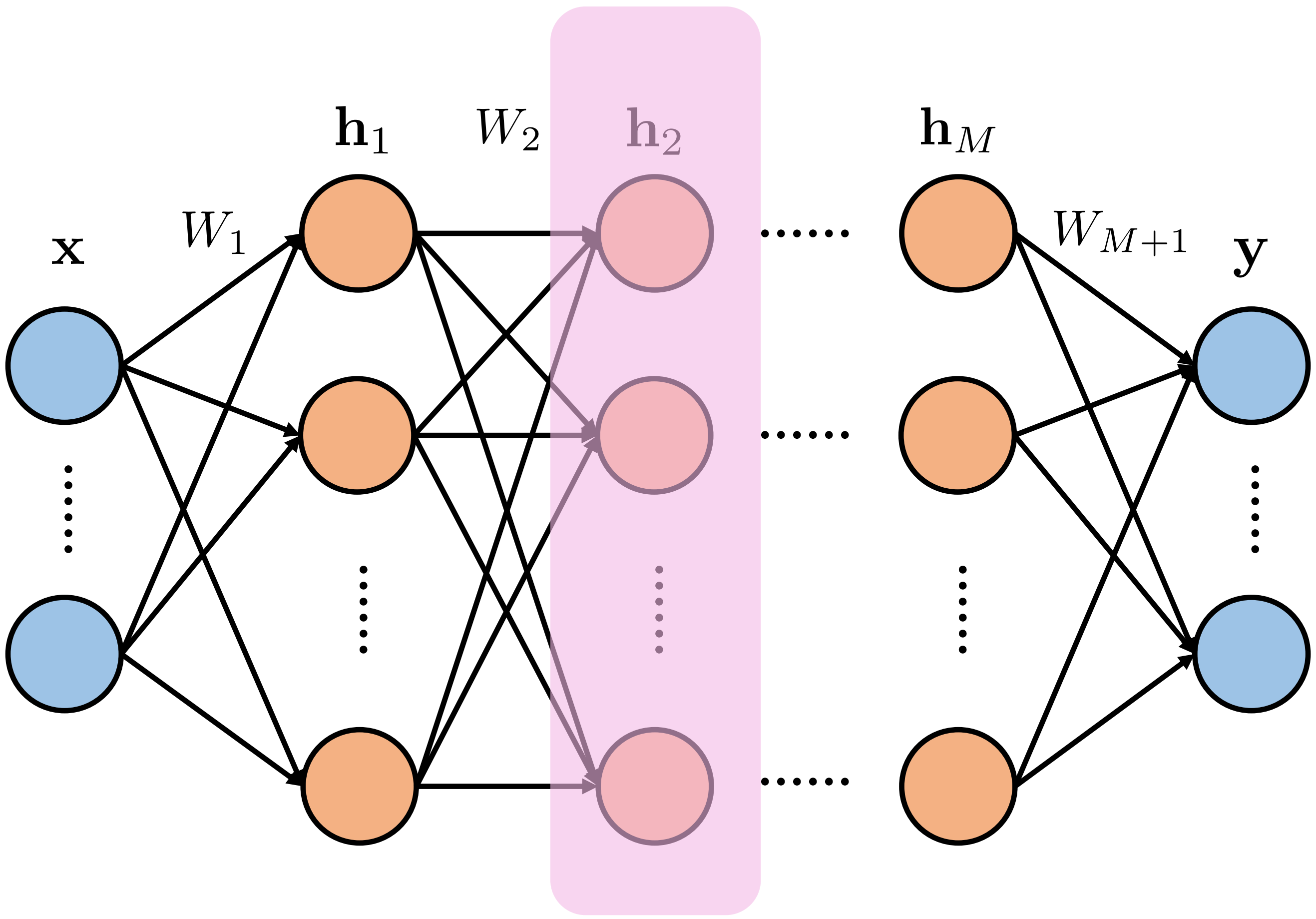


Input Sample

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

Feedforward Neural Networks

Consider a MLP as follows. Recall what we do in the forward pass:



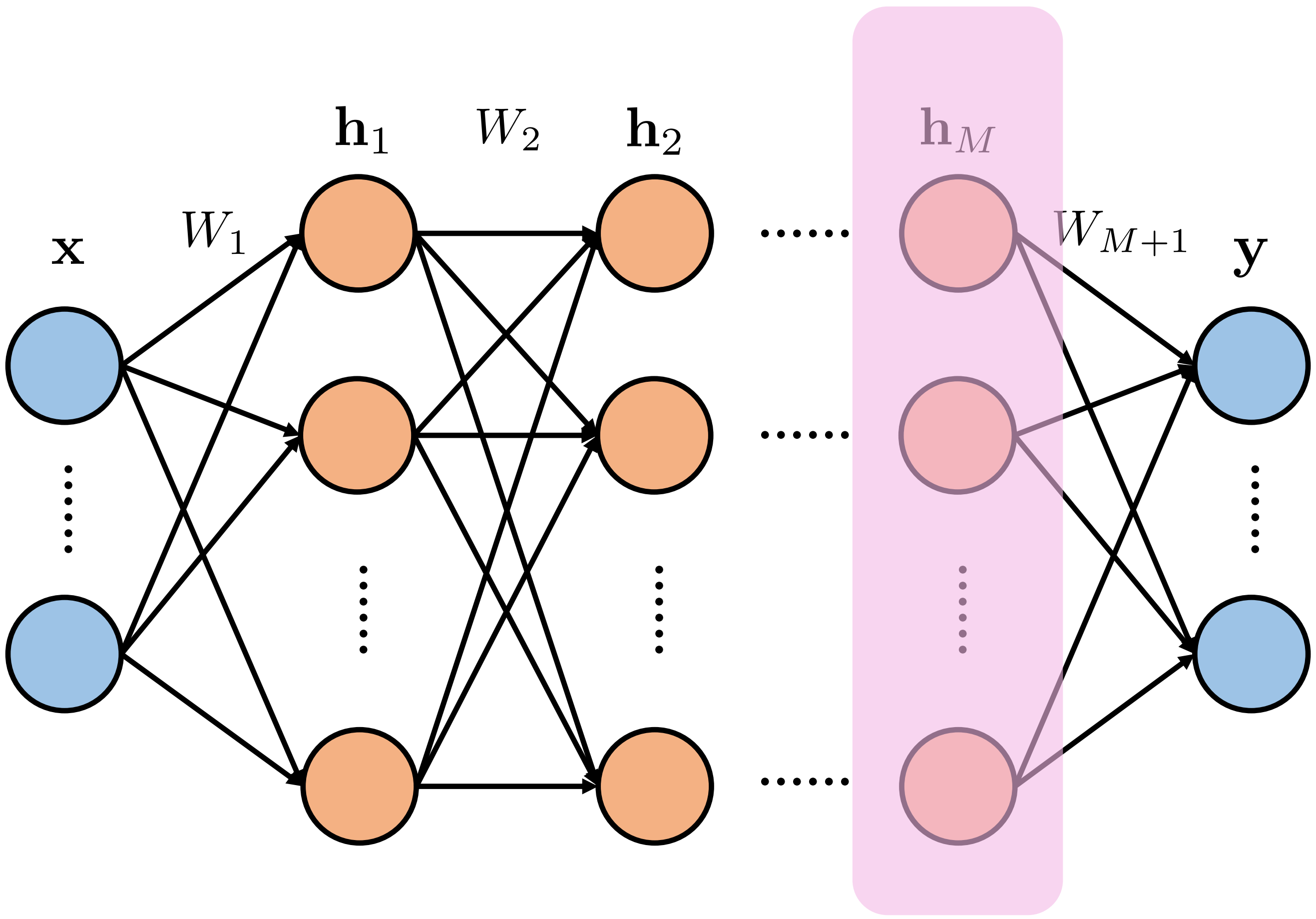
Input Sample

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

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

Feedforward Neural Networks

Consider a MLP as follows. Recall what we do in the forward pass:



Input Sample

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

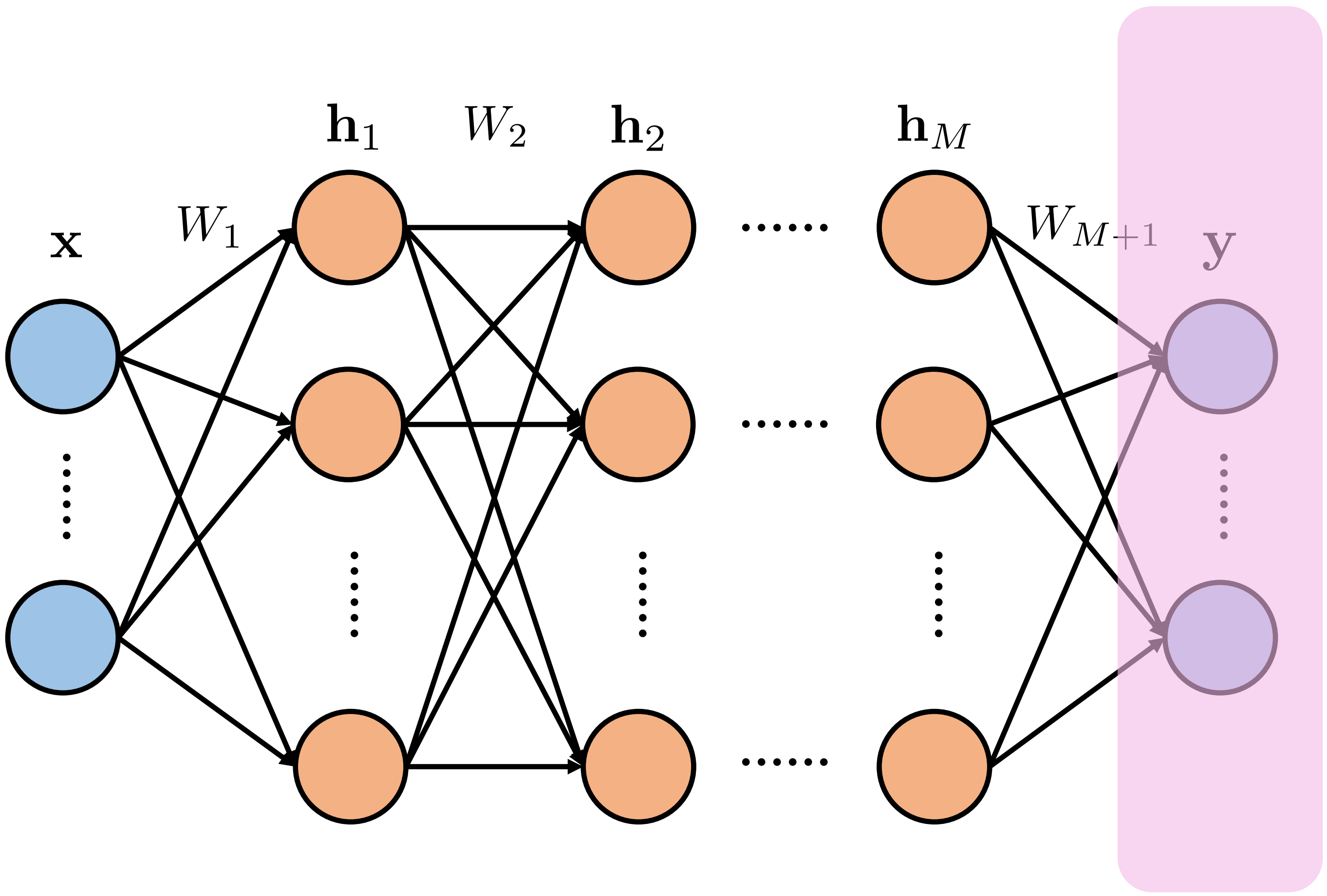
$$\mathbf{h}_2 = \sigma\left(W_2 \mathbf{h}_1\right)$$

$$\vdots$$

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

Feedforward Neural Networks

Consider a MLP as follows. Recall what we do in the forward pass:



Input Sample

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

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

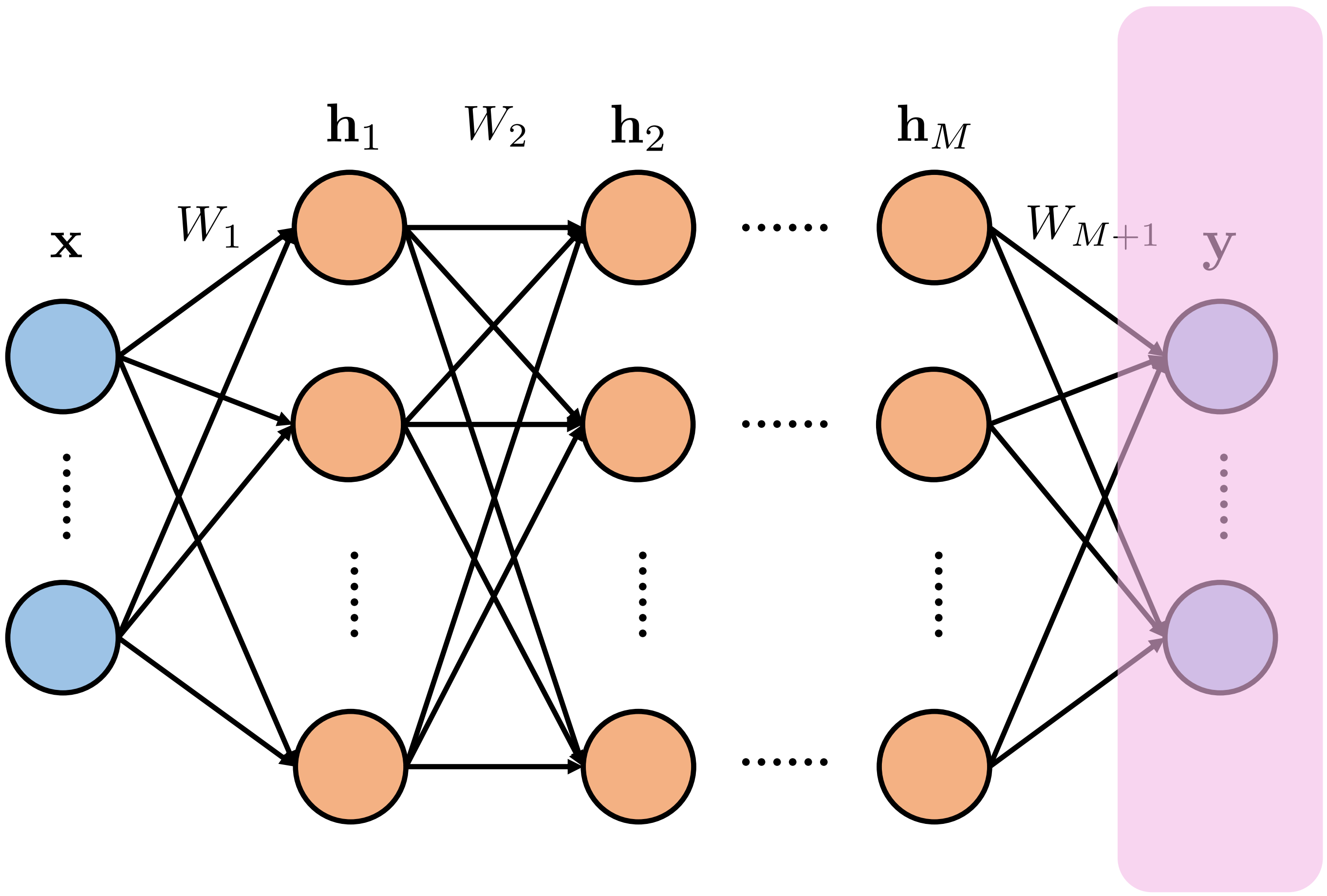
$$\vdots$$

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

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

Feedforward Neural Networks

Consider a MLP as follows. Recall what we do in the forward pass:



Input Sample

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

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

$$\vdots$$

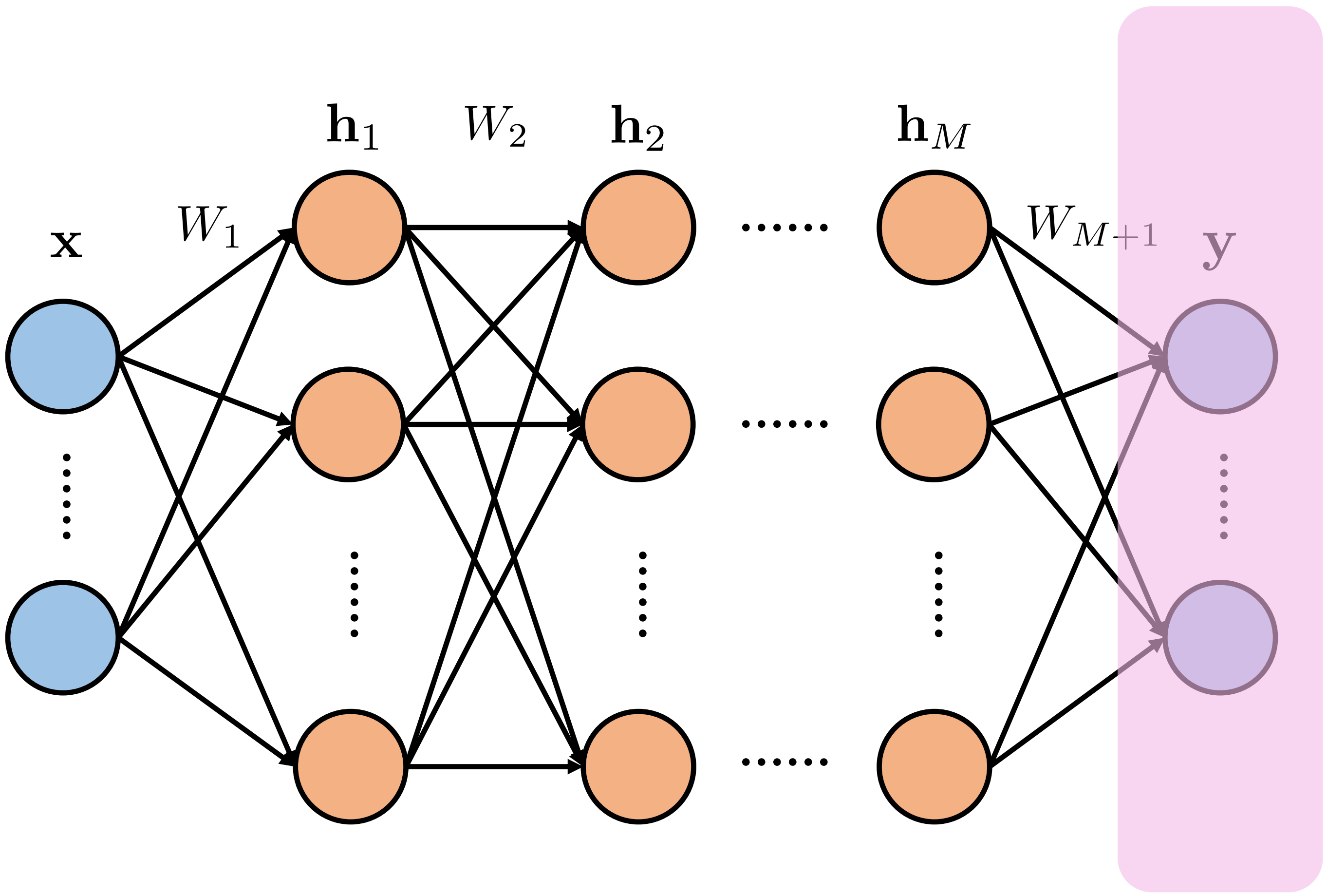
$$\mathbf{h}_M = \sigma\left(W_M\mathbf{h}_{M-1}\right)$$

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

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

Feedforward Neural Networks

Consider a MLP as follows. Recall what we do in the forward pass:



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

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

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

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 $f : \mathbb{R} \to \mathbb{R}$ $g : \mathbb{R} \to \mathbb{R}$

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 $f : \mathbb{R} \to \mathbb{R}$ $g : \mathbb{R} \to \mathbb{R}$

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

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 $f : \mathbb{R}^m \to \mathbb{R}$ $g : \mathbb{R}^n \to \mathbb{R}^m$

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

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 $f : \mathbb{R} \to \mathbb{R}$ $g : \mathbb{R} \to \mathbb{R}$

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

- 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 $f : \mathbb{R}^m \to \mathbb{R} \qquad g : \mathbb{R}^n \to \mathbb{R}^m$

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

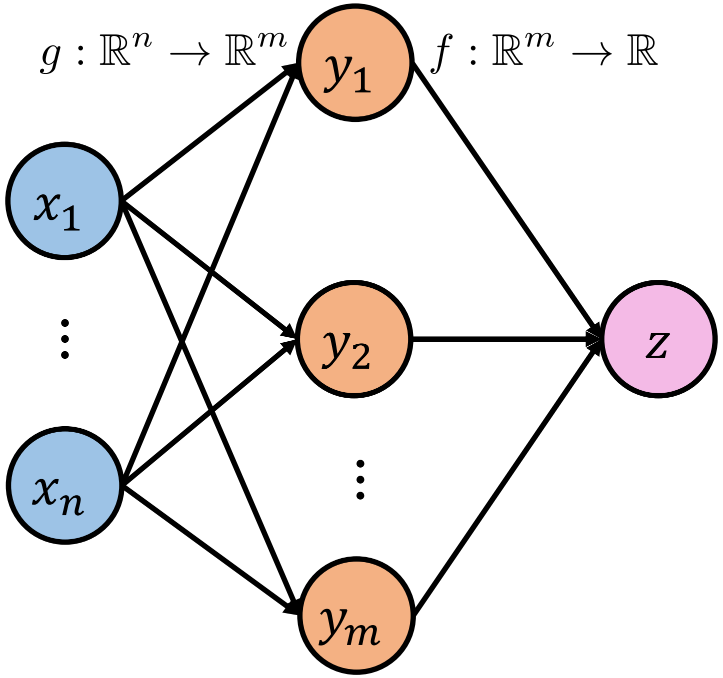
Backpropagation

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 $f : \mathbb{R}^m \to \mathbb{R} \qquad g : \mathbb{R}^n \to \mathbb{R}^m$

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



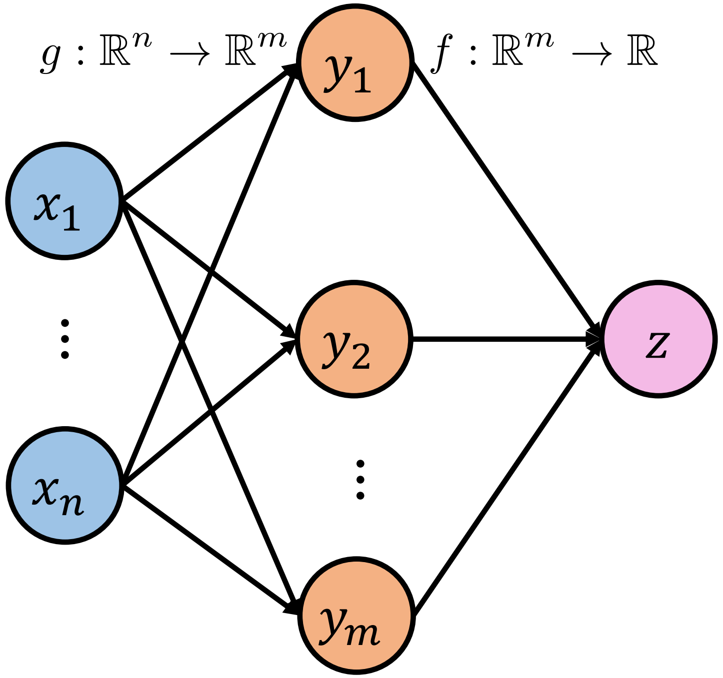
Backpropagation

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 $f : \mathbb{R}^m \to \mathbb{R}$ $\quad g : \mathbb{R}^n \to \mathbb{R}^m$

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



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

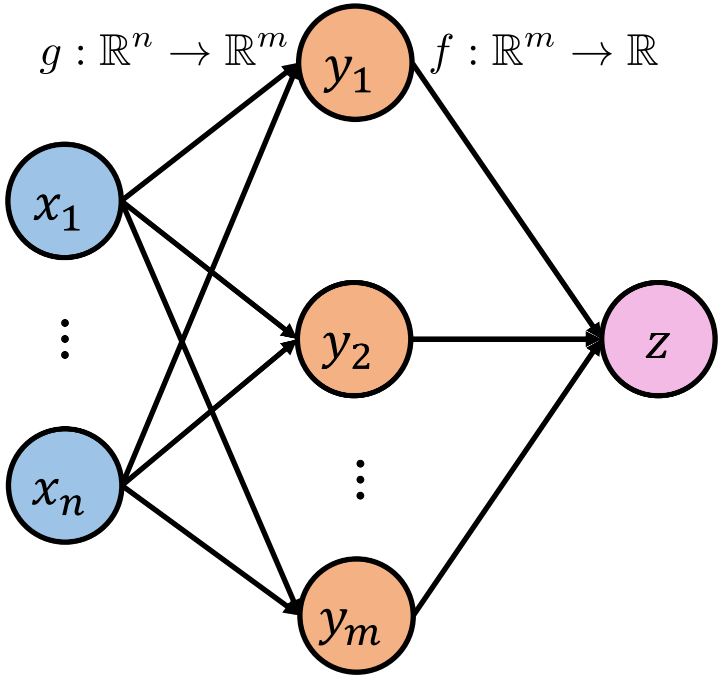
Backpropagation

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 $f : \mathbb{R}^m \to \mathbb{R}$ $\quad g : \mathbb{R}^n \to \mathbb{R}^m$

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



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

Consider all possible paths from x_1 to z!

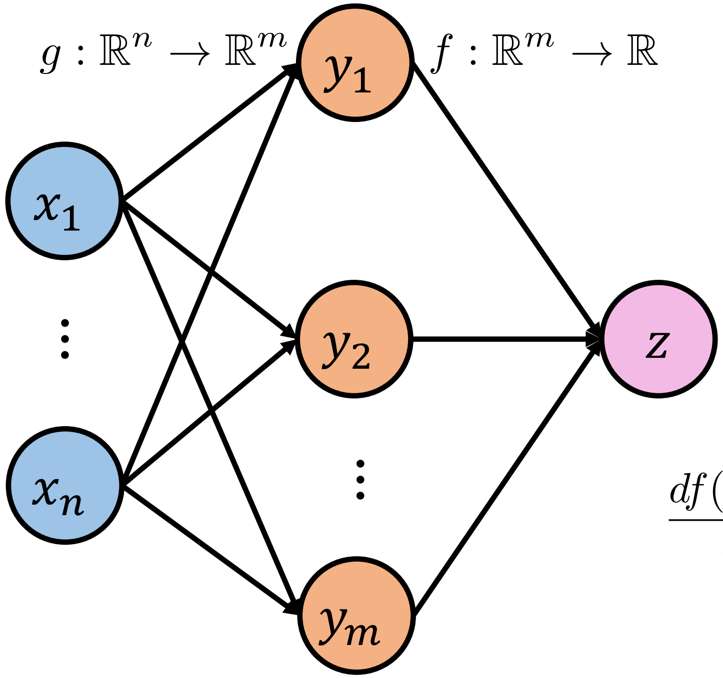
Backpropagation

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 $f : \mathbb{R}^m \to \mathbb{R} \quad g : \mathbb{R}^n \to \mathbb{R}^m$

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



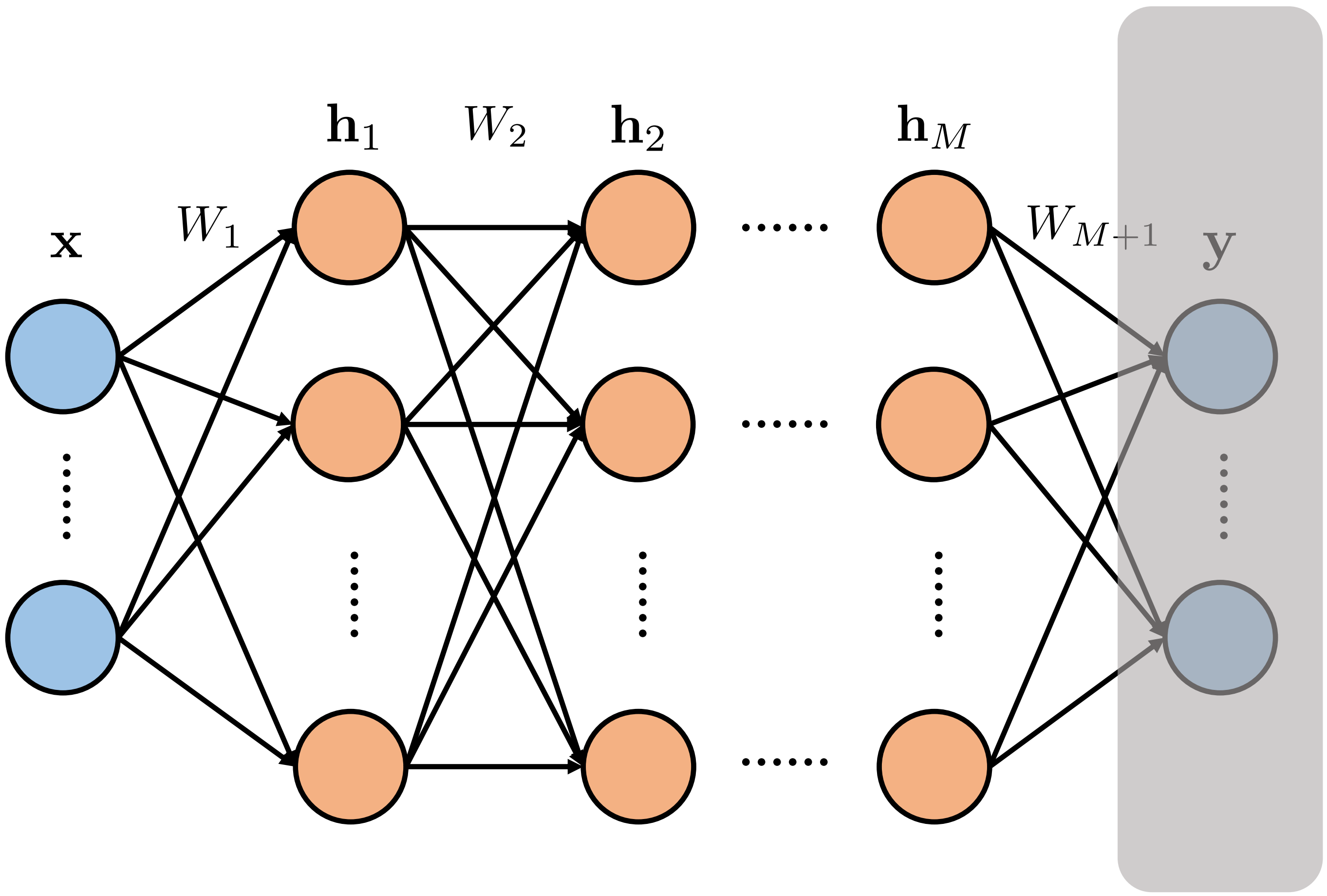
$$\frac{\partial z}{\partial x_1} = \sum_{i=1}^{m} \frac{\partial z}{\partial y_i} \frac{\partial y_i}{\partial x_1} = \begin{bmatrix} \frac{\partial z}{\partial y_1} & \cdots & \frac{\partial z}{\partial y_m} \end{bmatrix} \begin{bmatrix} \frac{\partial y_1}{\partial x_1} \\ \vdots \\ \frac{\partial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

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

Backpropagation

During the backward pass:

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

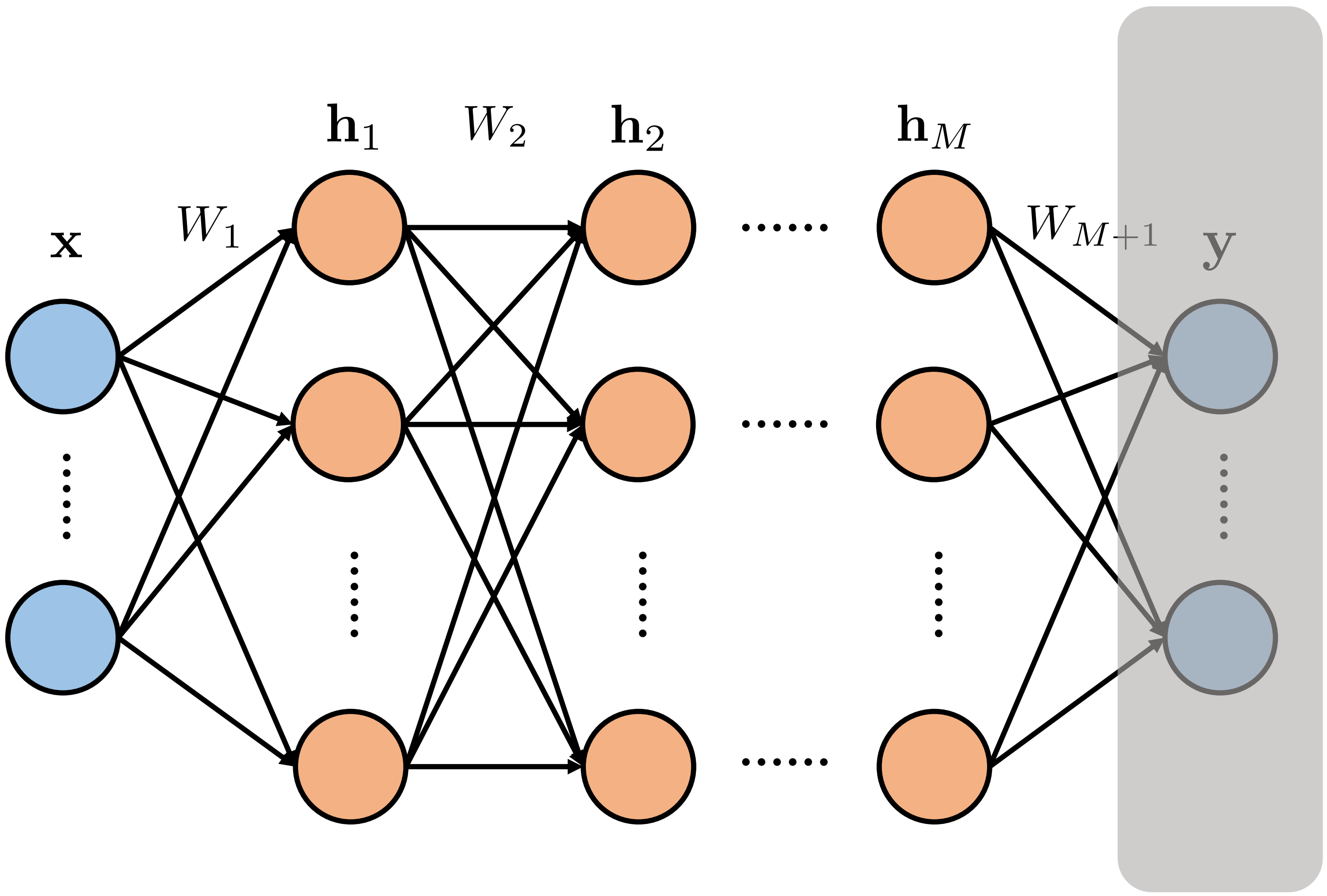


Backpropagation

During the backward pass:

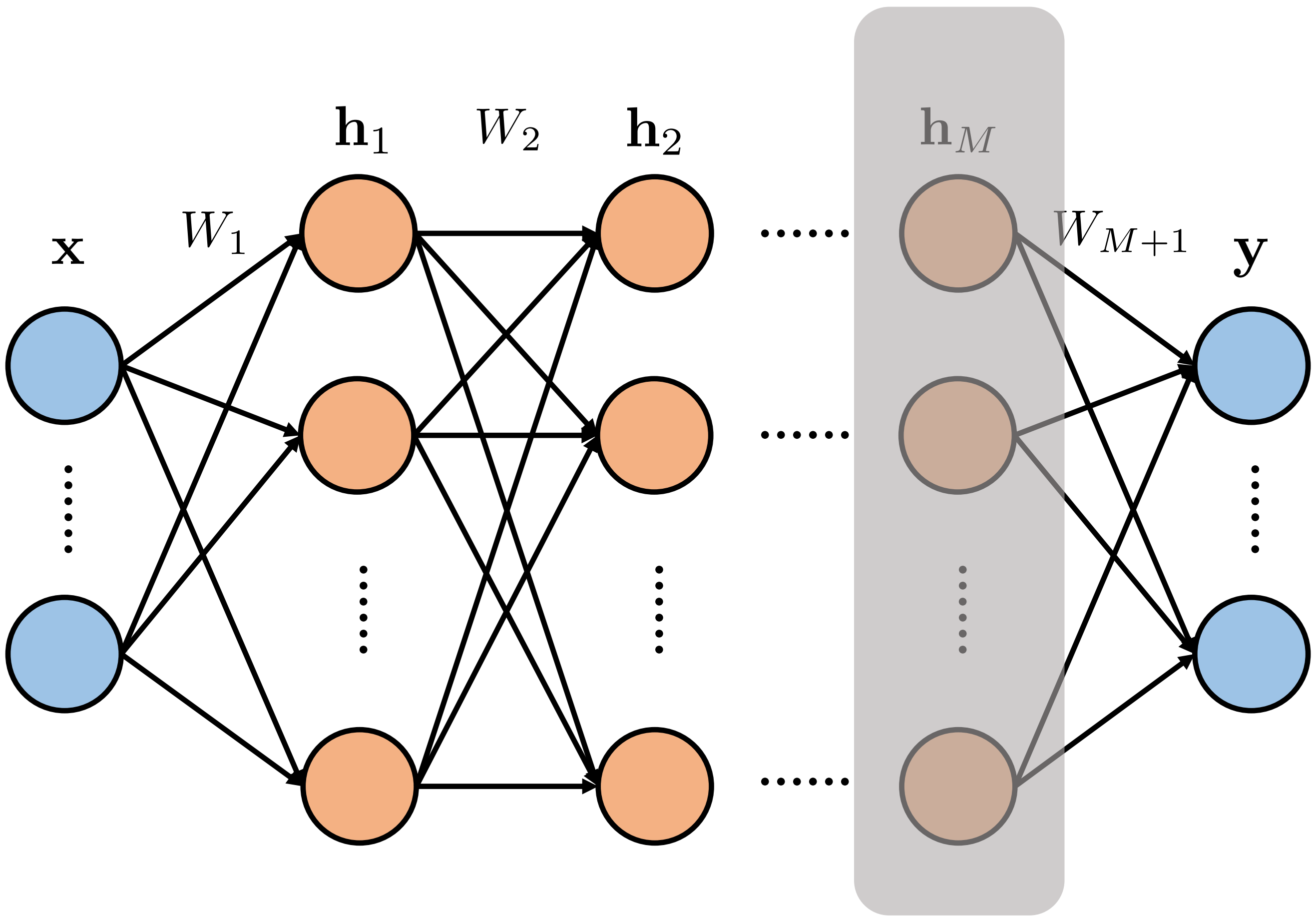
Loss: $L = \ell(\mathbf{y}, \bar{\mathbf{y}})$

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



Backpropagation

During the backward pass:



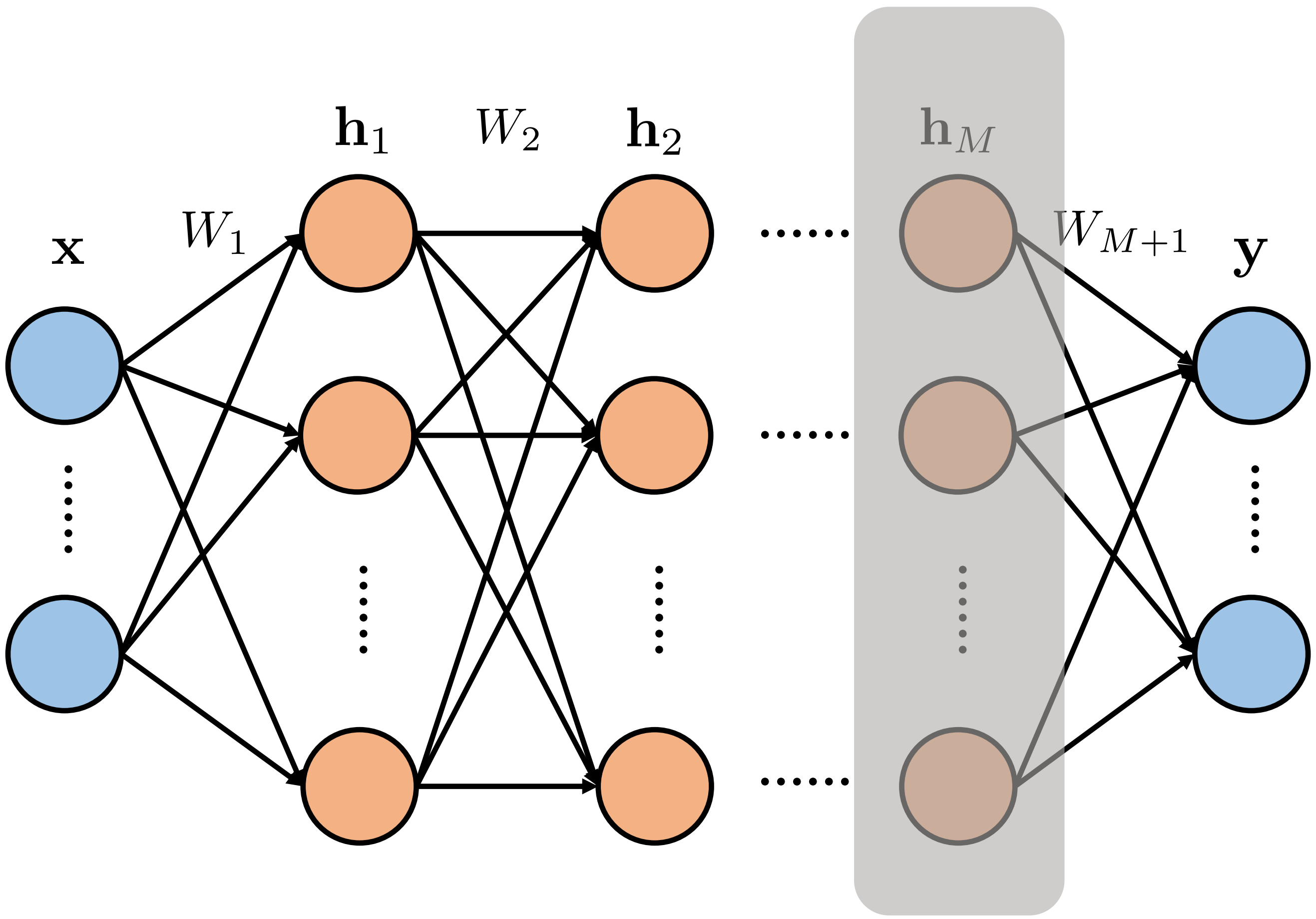
Loss: $L = \ell(\mathbf{y}, \bar{\mathbf{y}})$

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

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. $\mathbf{y}$: $\dfrac{\partial L}{\partial \mathbf{y}}$

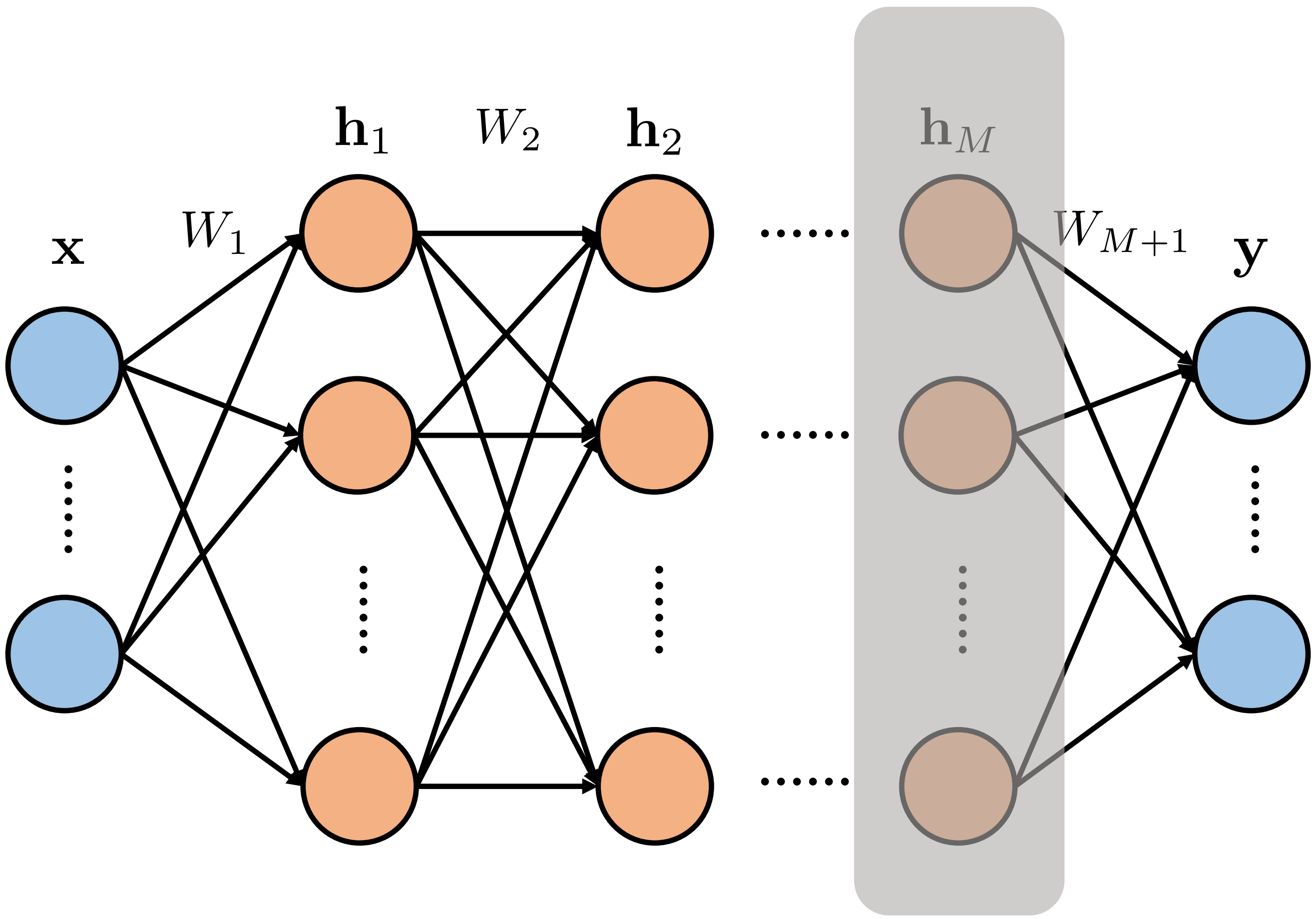
Gradient of loss w.r.t. $\mathbf{h}_M$:

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. $\mathbf{y}$: $\dfrac{\partial L}{\partial \mathbf{y}}$

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

Apply the chain rule we learned before:

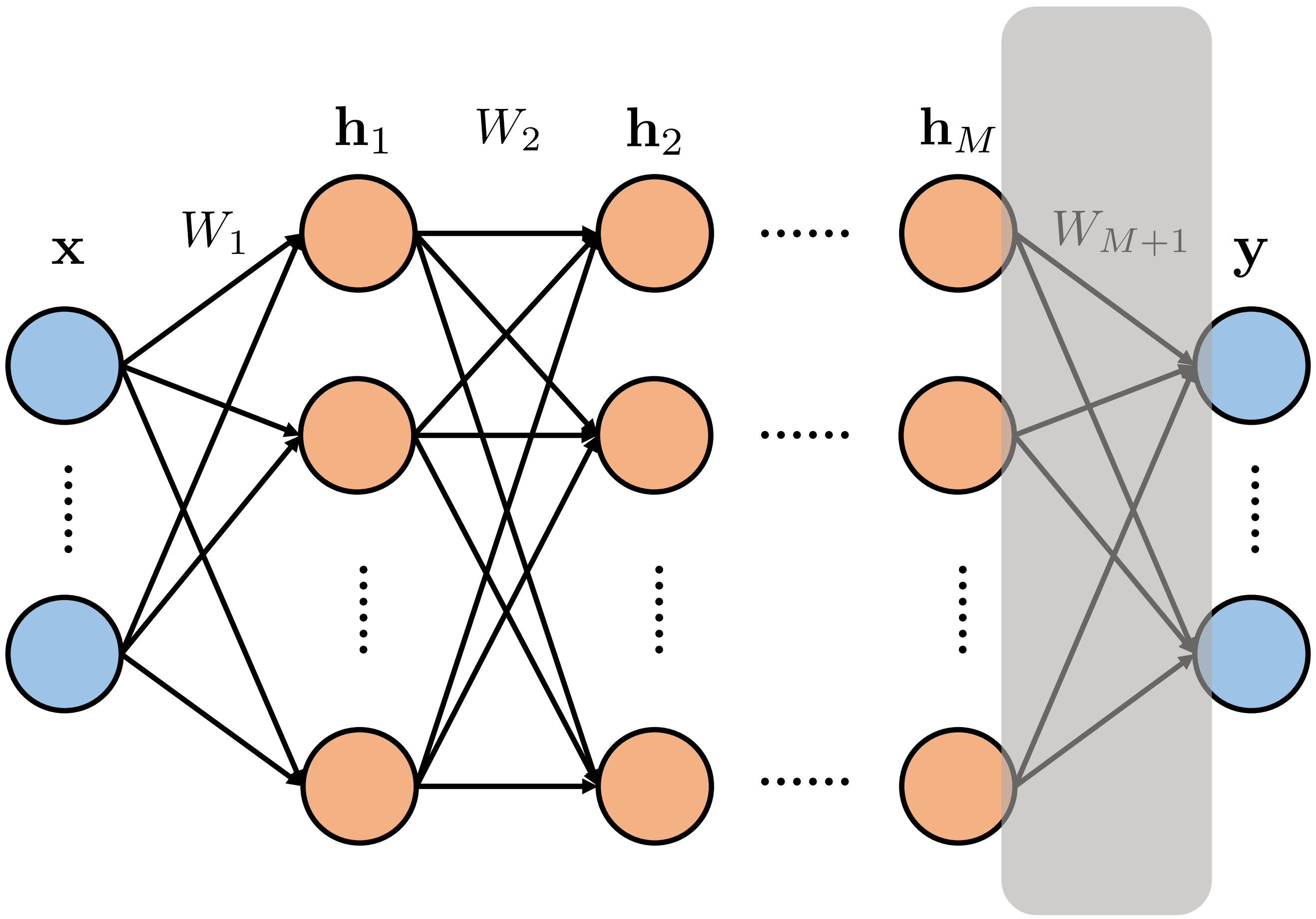
$$\frac{\partial L}{\partial \mathbf{h}_M} = \left(\frac{\partial \mathbf{y}}{\partial \mathbf{h}_M}\right)^\top \frac{\partial \mathbf{L}}{\partial \mathbf{y}}$$

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

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

Backpropagation

During the backward pass:



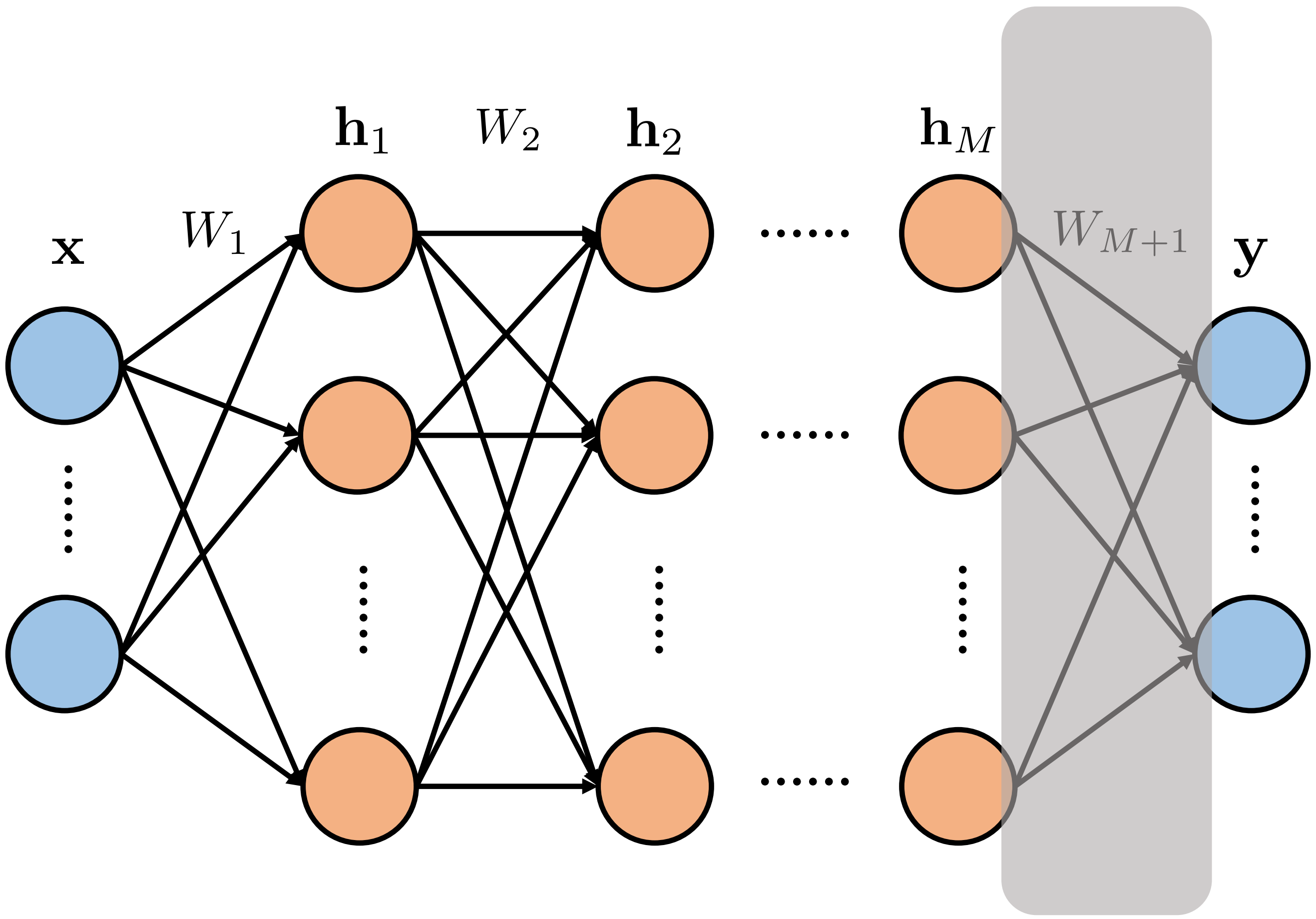
Loss: $L = \ell(\mathbf{y}, \bar{\mathbf{y}})$

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

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. $\mathbf{y}$: $\dfrac{\partial L}{\partial \mathbf{y}}$

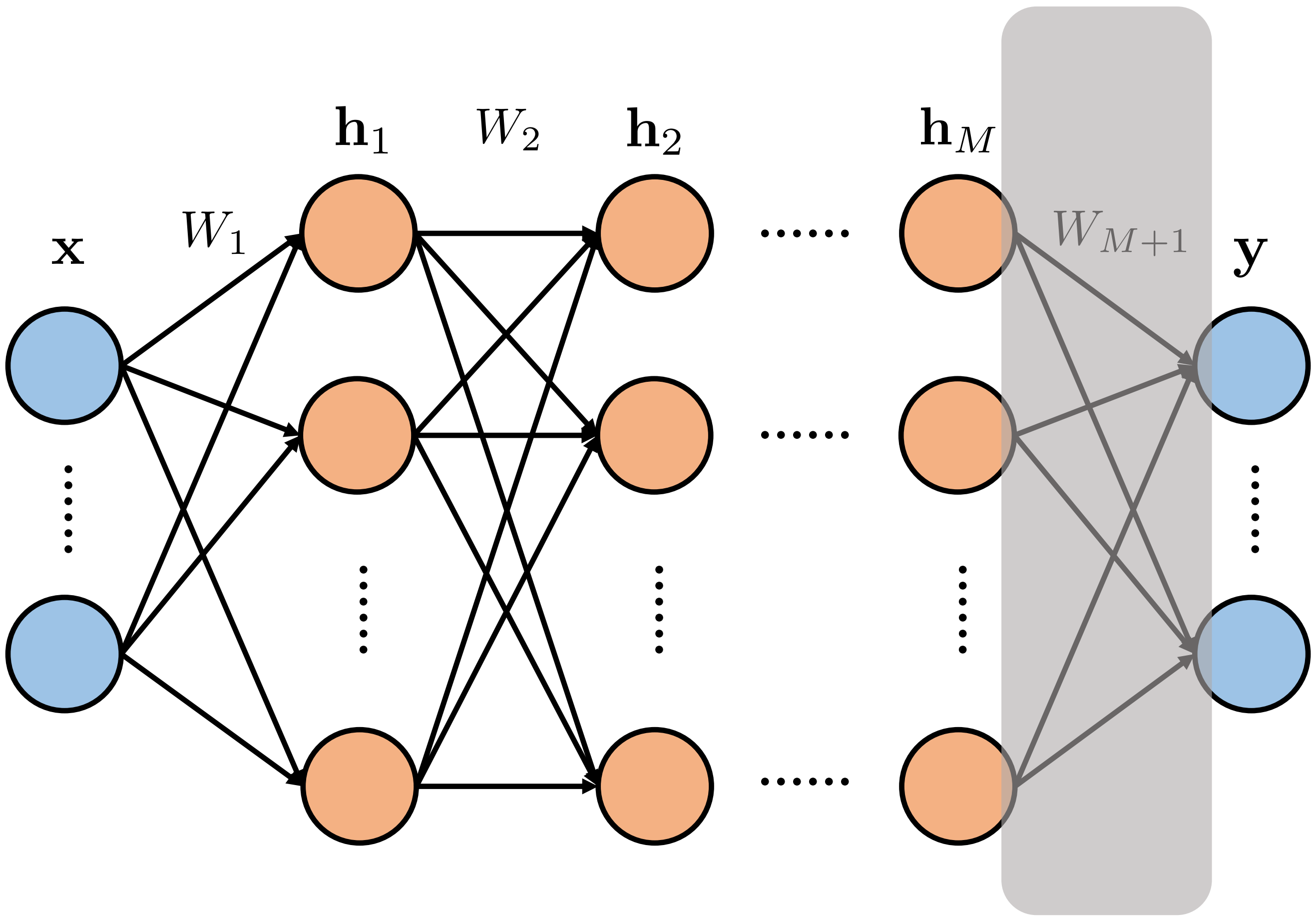
Gradient of loss w.r.t. W_{M+1} :

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. $\mathbf{y}$: $\dfrac{\partial L}{\partial \mathbf{y}}$

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

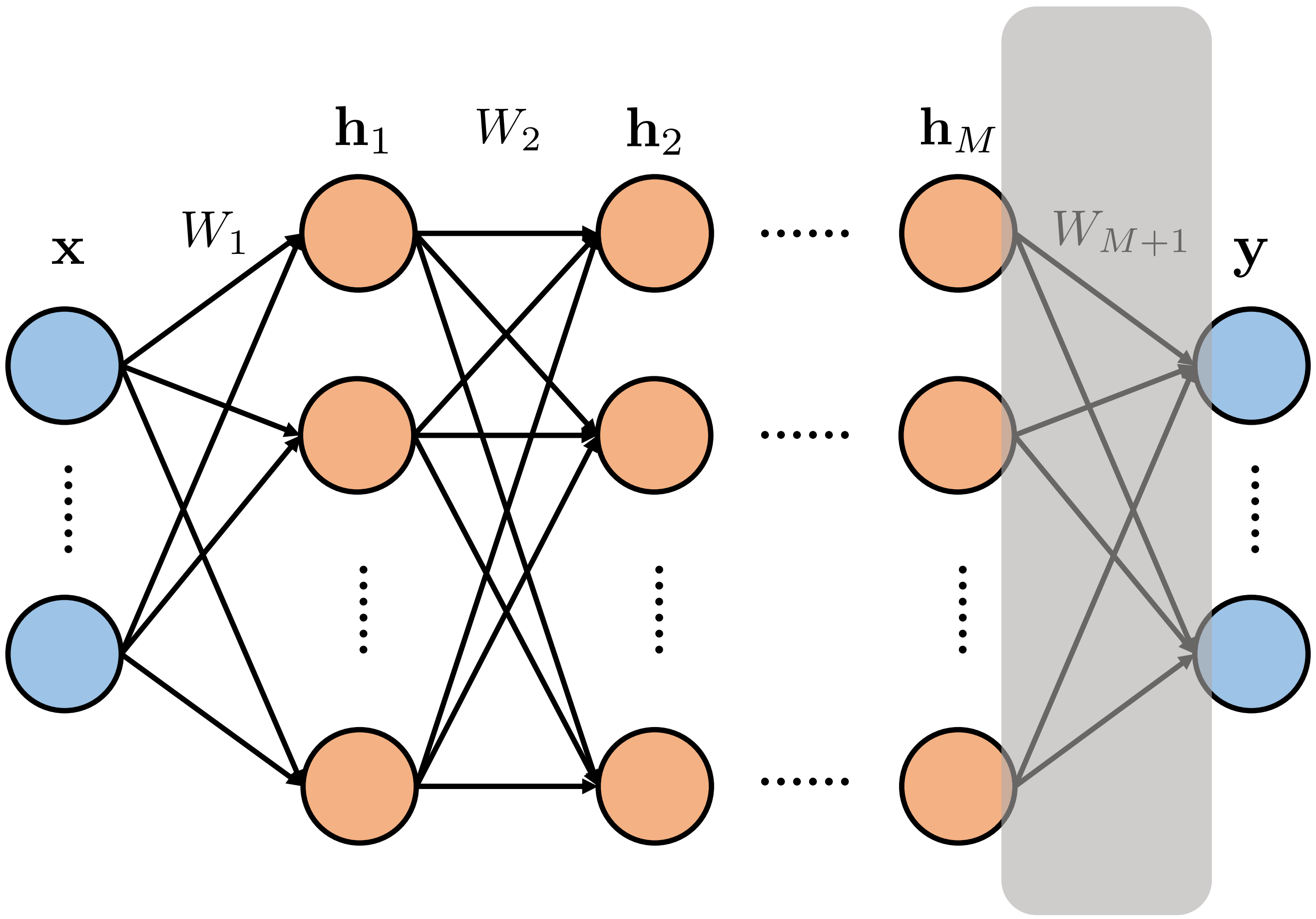
Ideally, chain rule should be something like

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

But this is wrong, since $\dfrac{\partial \mathbf{y}}{\partial W_{M+1}}$ is the derivative of a vector w.r.t. a matrix, shapes do not work out!

Backpropagation

During the backward pass:



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

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

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

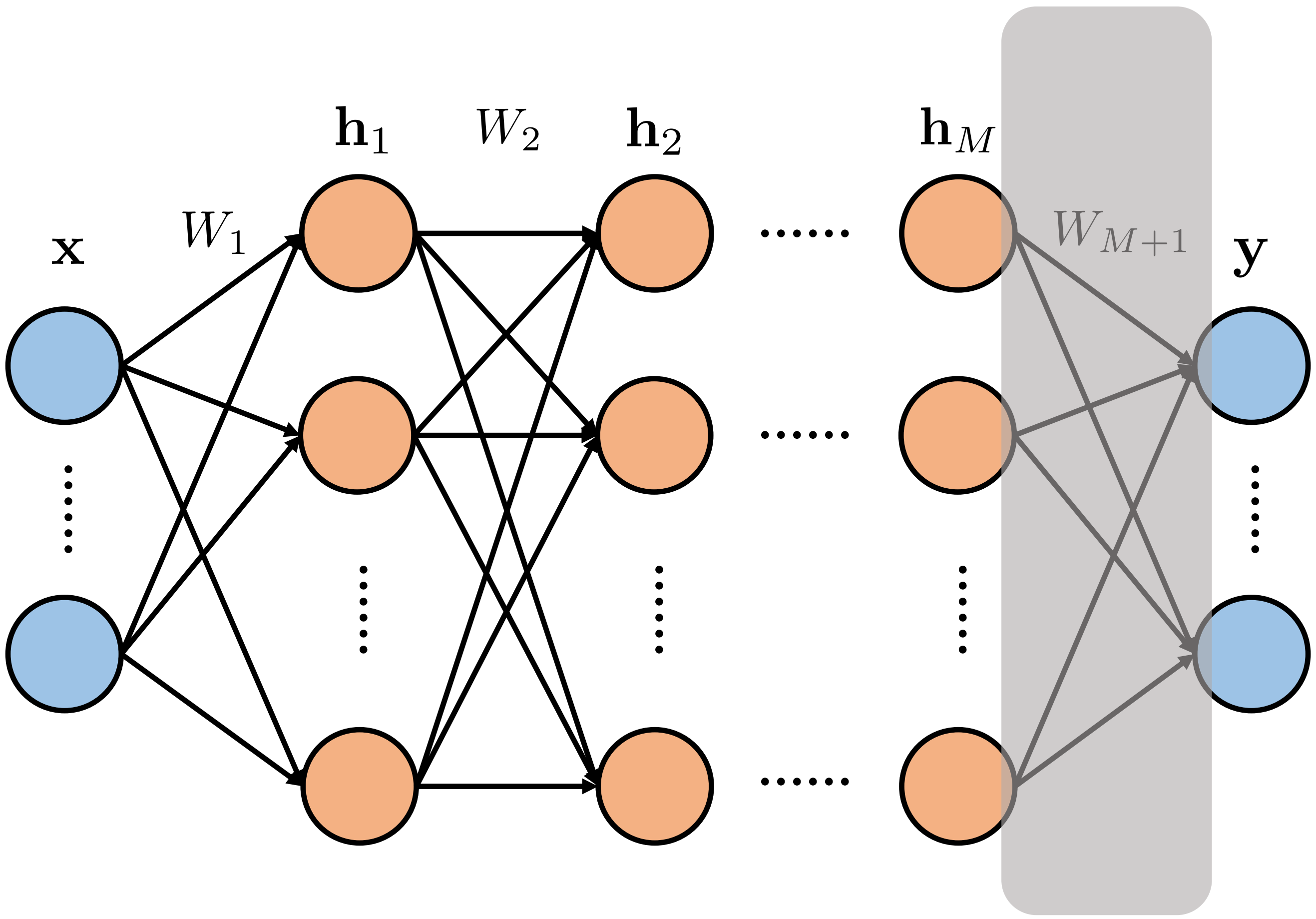
Note $\mathbf{y}[i]$ only depends on $W_{M+1}[i,:]$

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. $\mathbf{y}$: $\dfrac{\partial L}{\partial \mathbf{y}}$

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

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

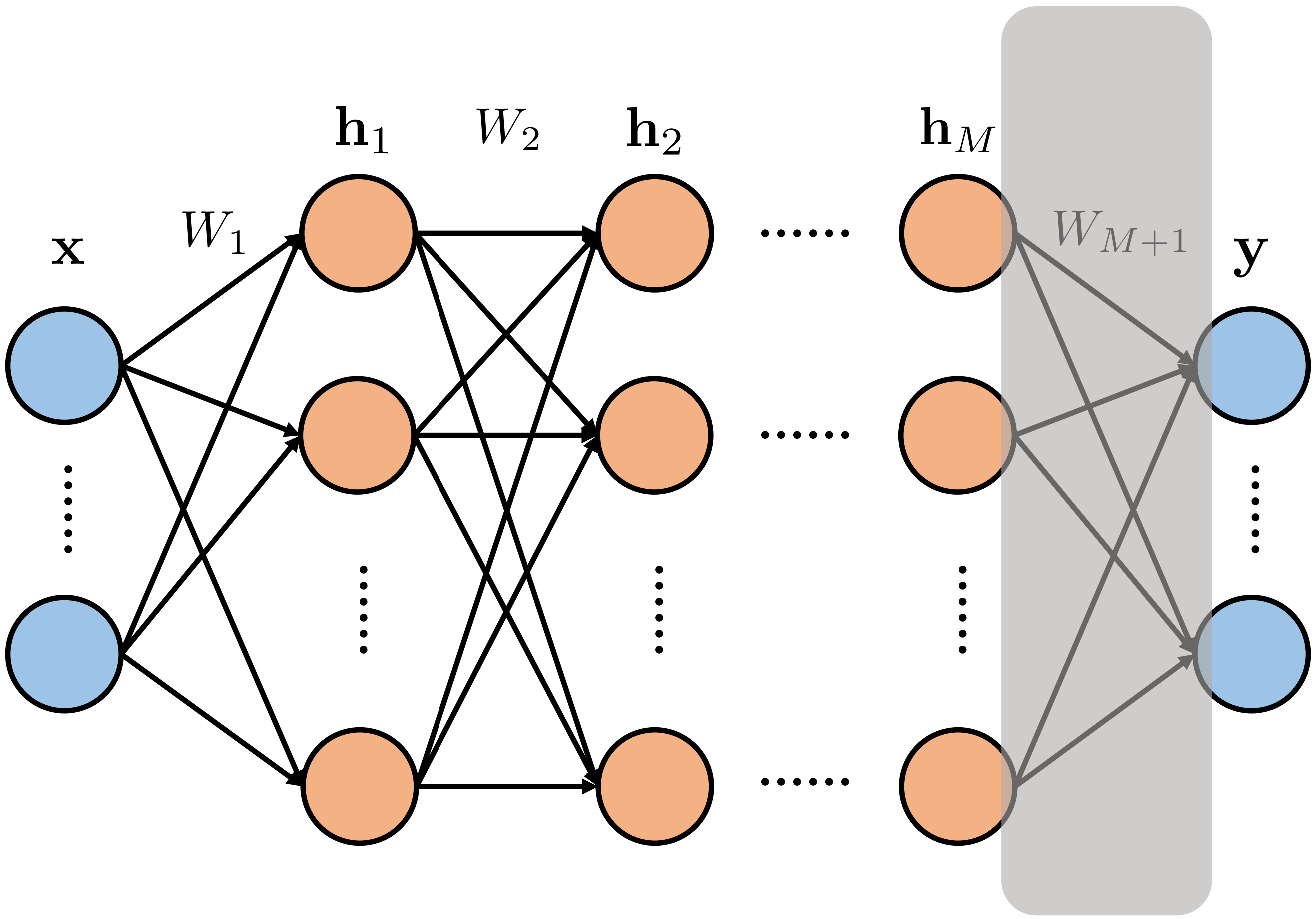
does not depend on $W_{M+1}[j,:] \quad \forall j \neq i$

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

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

Backpropagation

During the backward pass:



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

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

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

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

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

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

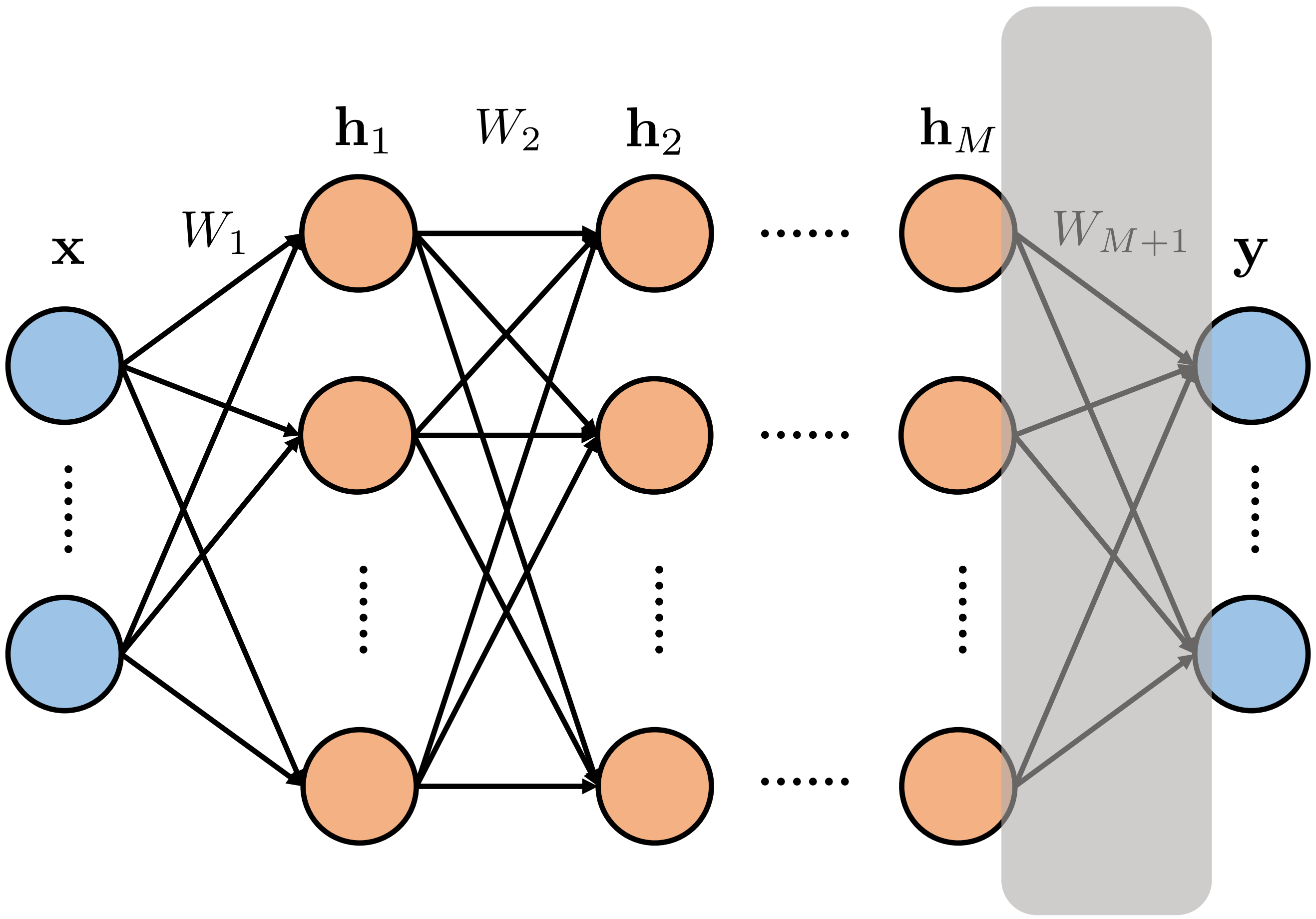
$$\frac{\partial \mathbf{y}[i]}{\partial W_{M+1}[i,j]} = \mathbf{h}[j]$$

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. $\mathbf{y}$: $\dfrac{\partial L}{\partial \mathbf{y}}$

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

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

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

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

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

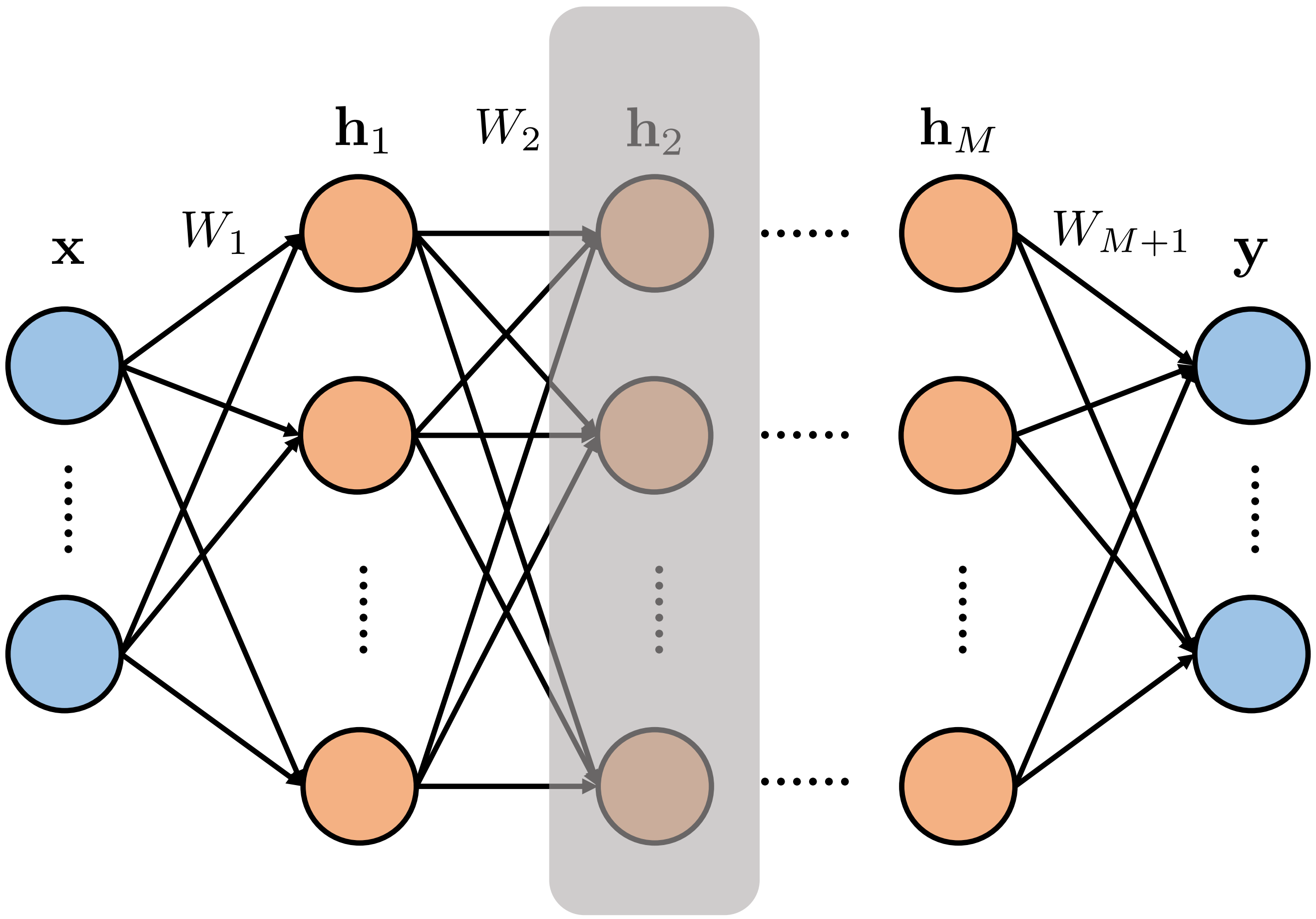
We have

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

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

Backpropagation

During the backward pass:



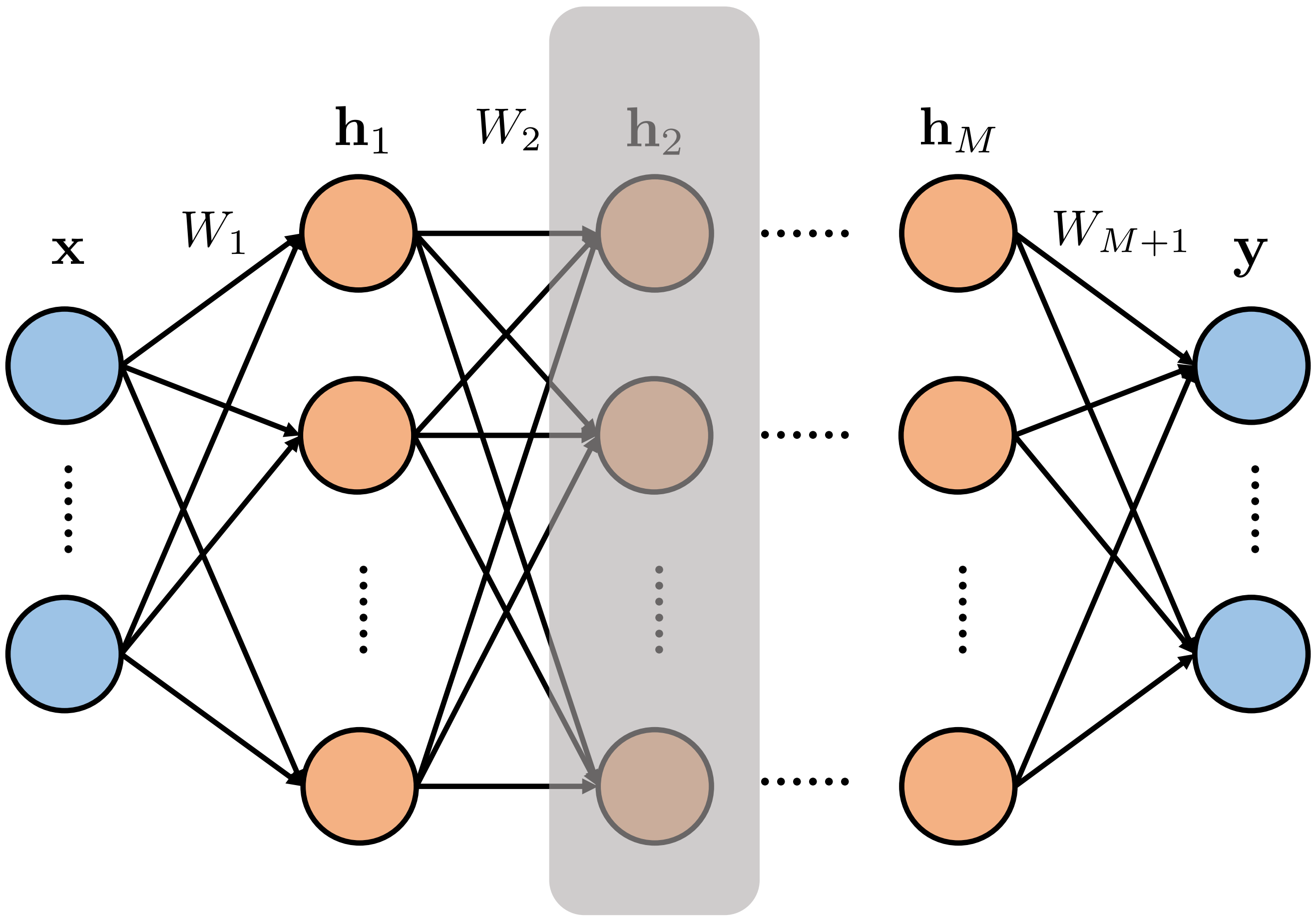
Loss: $L = \ell(\mathbf{y}, \bar{\mathbf{y}})$

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

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. $\mathbf{y}$: $\frac{\partial L}{\partial \mathbf{y}}$

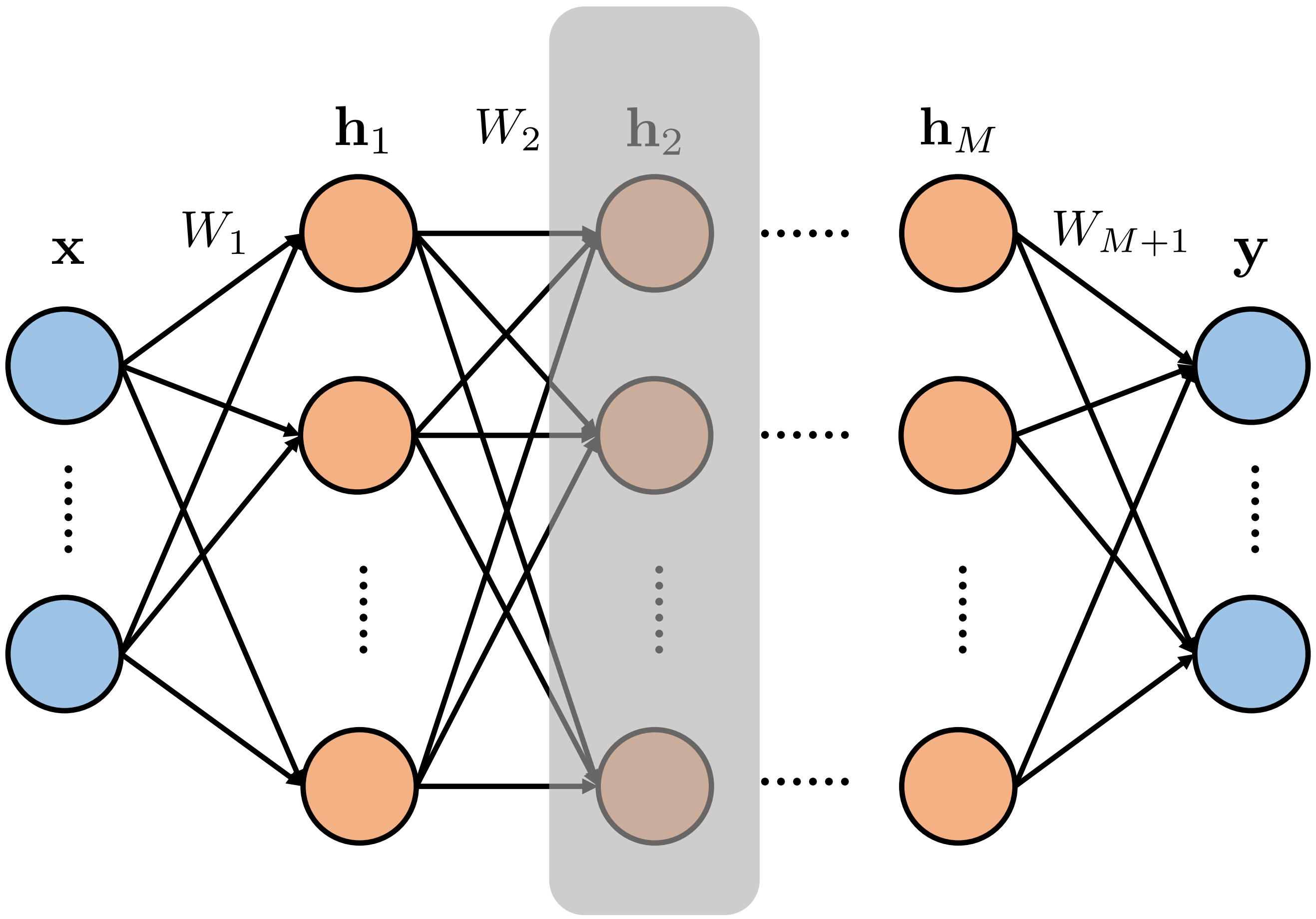
Gradient of loss w.r.t. $\mathbf{h}_2$:

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. $\mathbf{y}$: $\dfrac{\partial L}{\partial \mathbf{y}}$

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

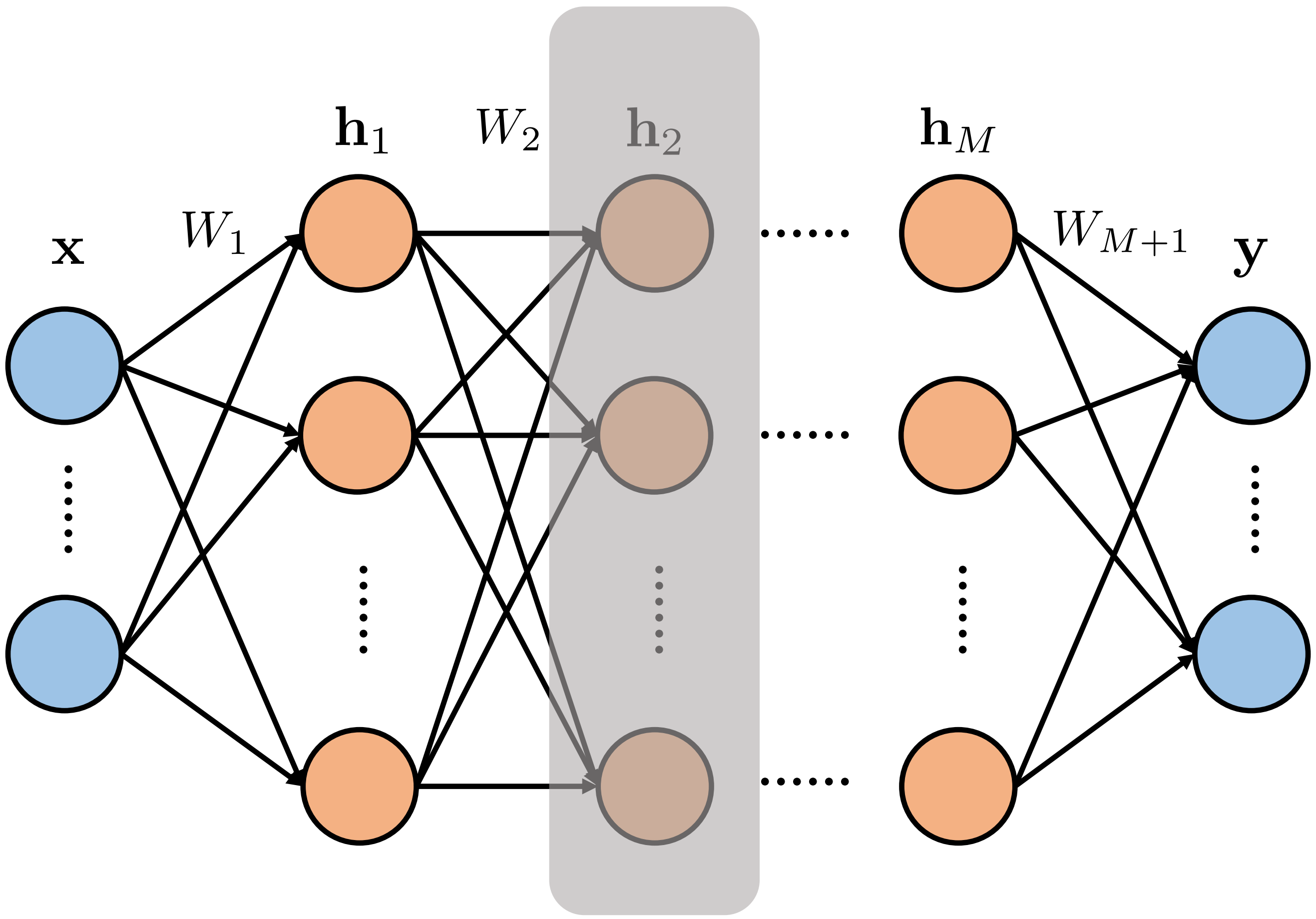
We know

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

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

Backpropagation

During the backward pass:



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

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

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

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

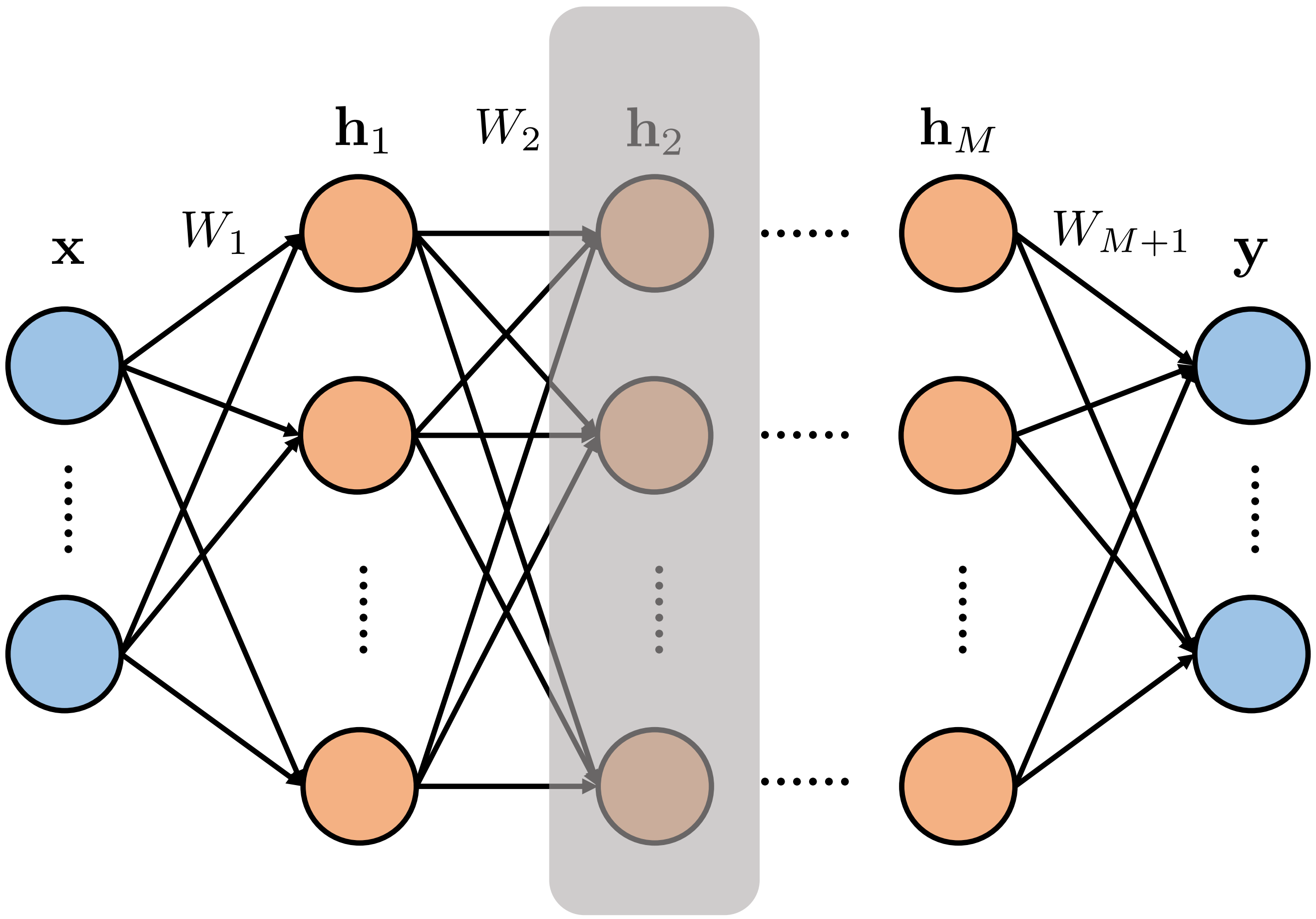
$$\frac{\partial L}{\partial \mathbf{h}_{M-1}} = \left(\frac{\partial \mathbf{h}_M}{\partial \mathbf{h}_{M-1}}\right)^\top \frac{\partial L}{\partial \mathbf{h}_M}$$

$$\vdots$$

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

Backpropagation

During the backward pass:



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

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

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

We know

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

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

$$\vdots$$

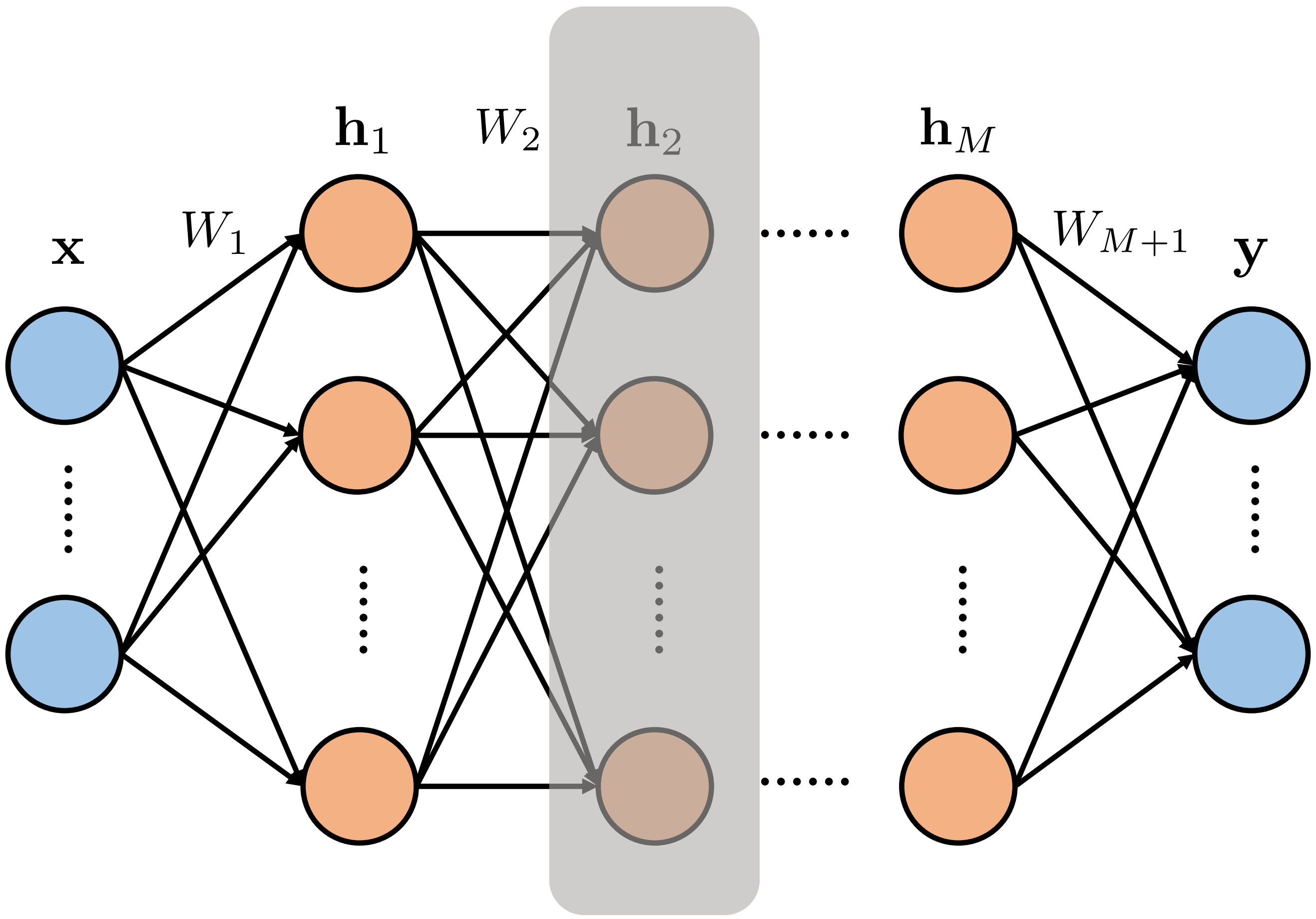
$$\frac{\partial L}{\partial \mathbf{h}_2} = \left(\frac{\partial \mathbf{h}_3}{\partial \mathbf{h}_2}\right)^\top \cdots \left(\frac{\partial \mathbf{h}_M}{\partial \mathbf{h}_{M-1}}\right)^\top \left(\frac{\partial \mathbf{y}}{\partial \mathbf{h}_M}\right)^\top \frac{\partial L}{\partial \mathbf{y}}$$

General form:

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

Backpropagation

During the backward pass:



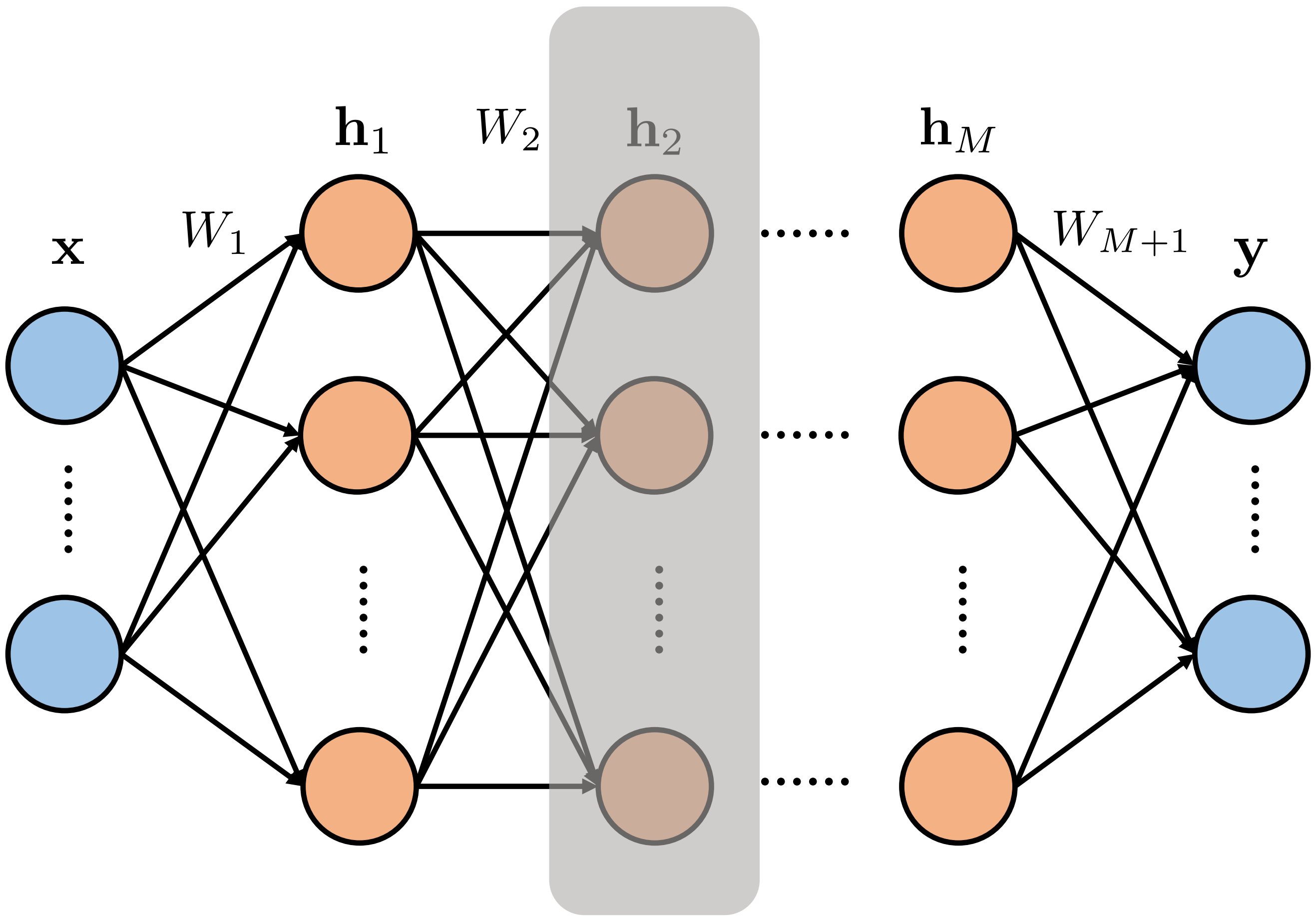
Loss: $L = \ell(\mathbf{y}, \bar{\mathbf{y}})$

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

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

Backpropagation

During the backward pass:



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

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

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

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

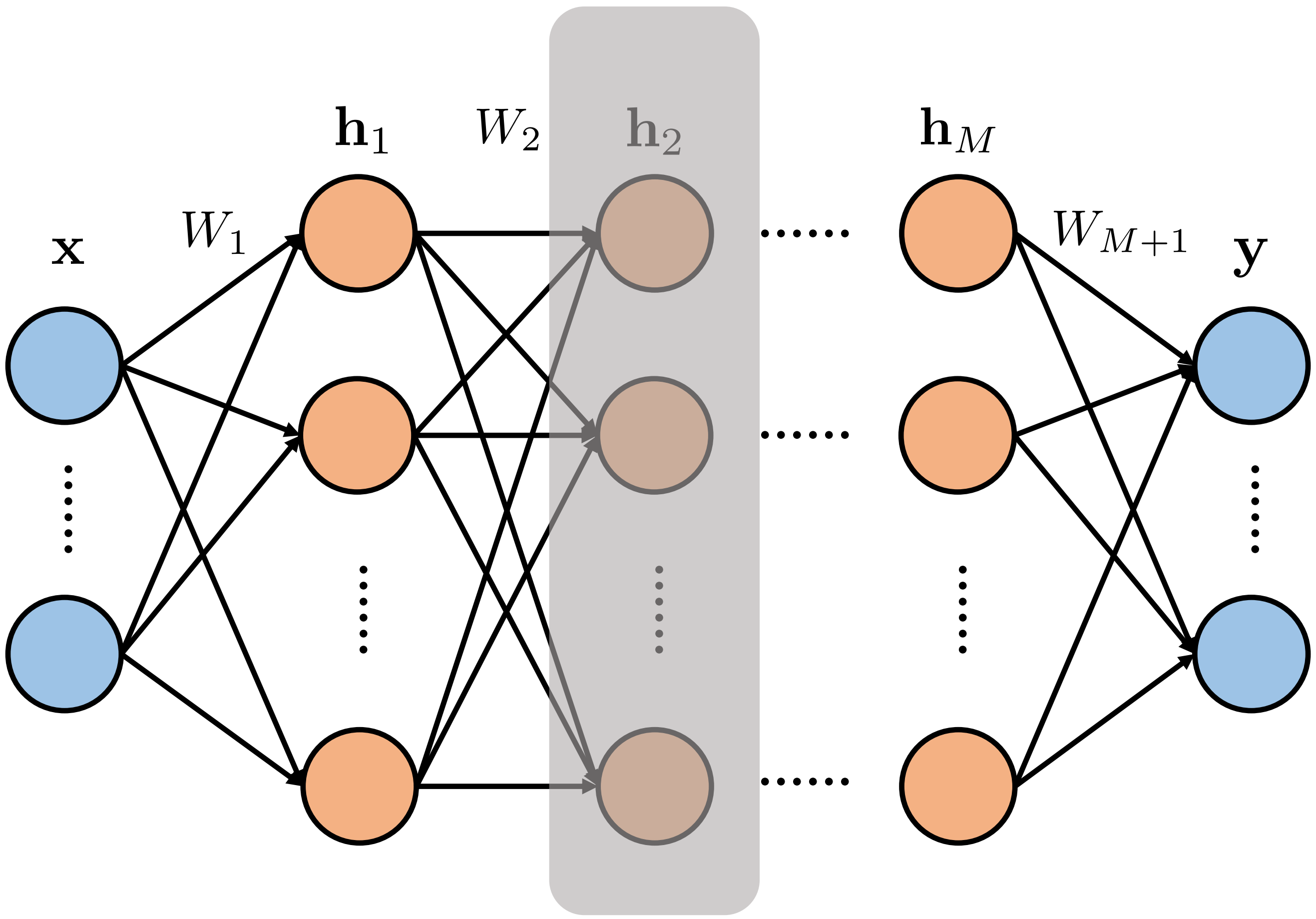
Denoting $\mathbf{z}_2 = W_2 \mathbf{h}_1$

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. $\mathbf{y}$: $\dfrac{\partial L}{\partial \mathbf{y}}$

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

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

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

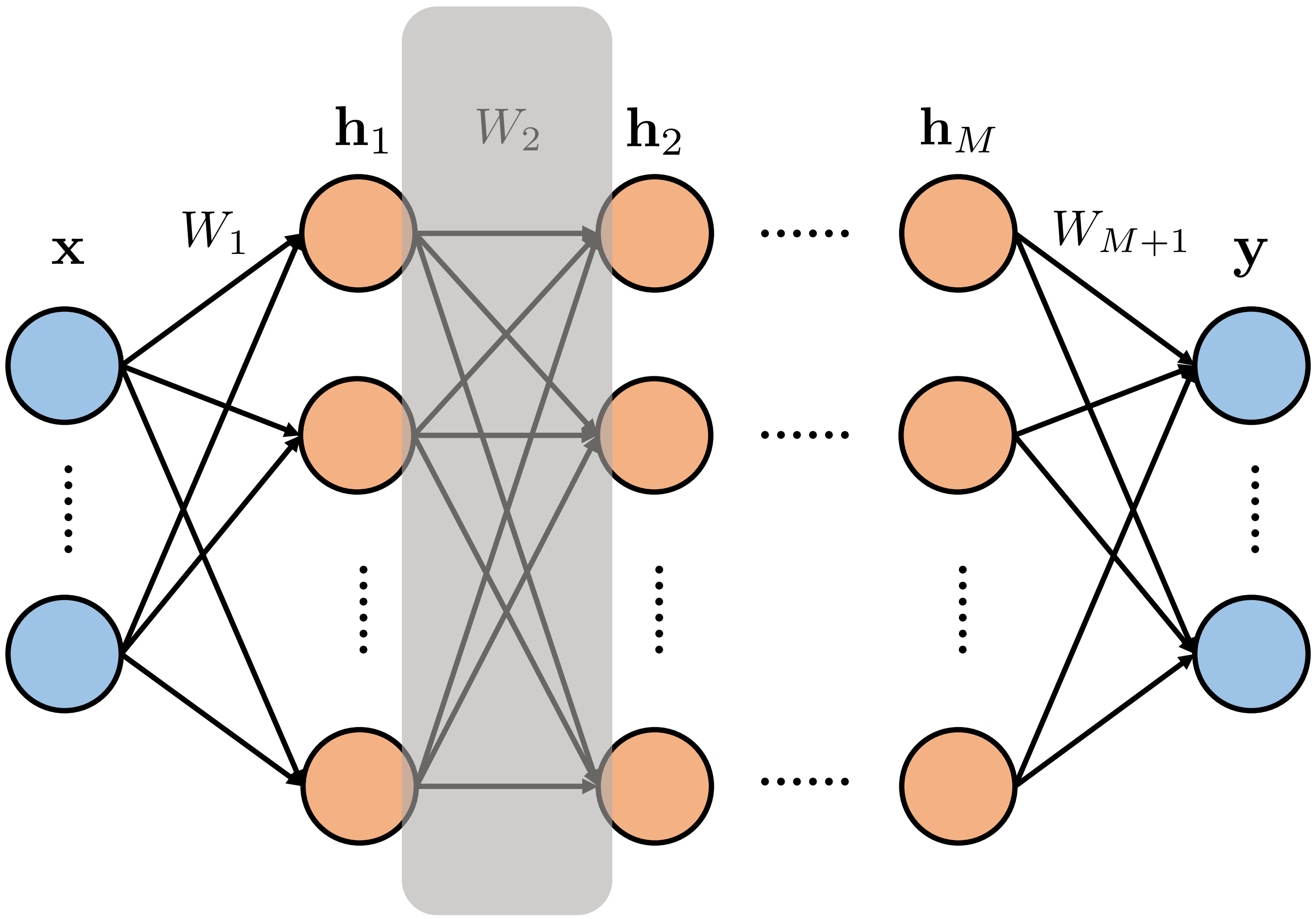
We have

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

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

Backpropagation

During the backward pass:



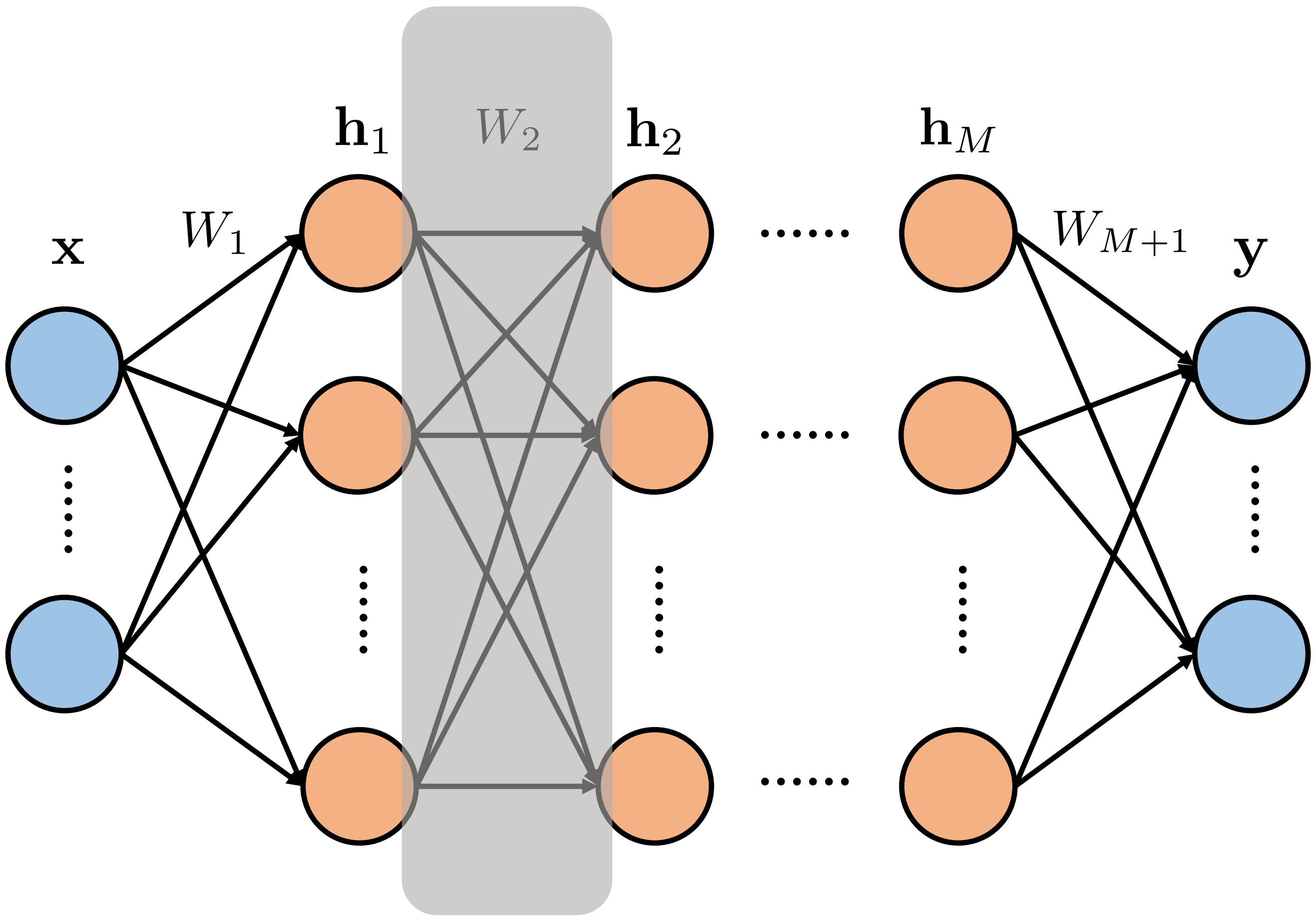
Loss: $L = \ell(\mathbf{y}, \bar{\mathbf{y}})$

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. $\mathbf{y}$: $\dfrac{\partial L}{\partial \mathbf{y}}$

Gradient of loss w.r.t. W_2 :

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

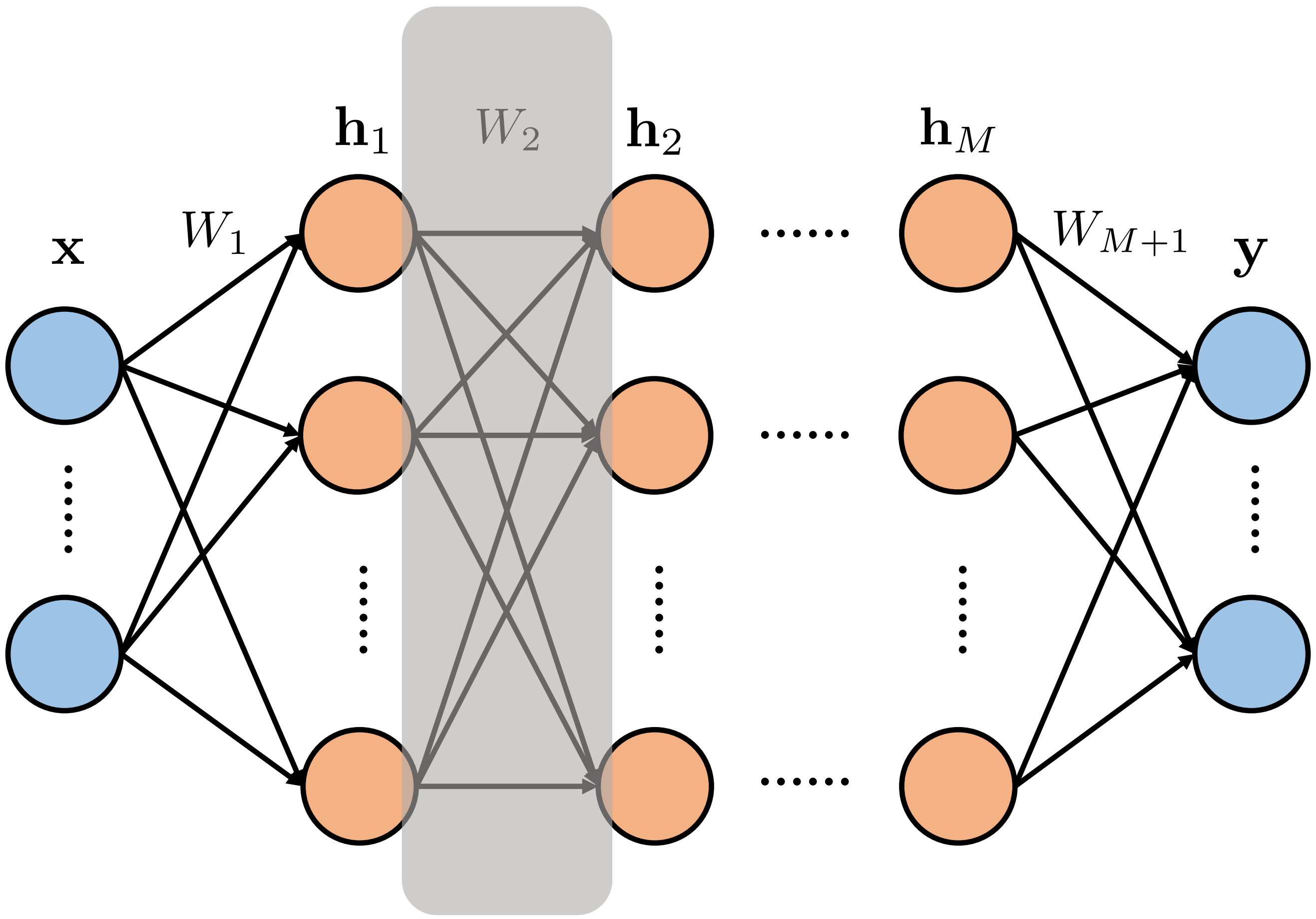
$\mathbf{z}_2 = W_2 \mathbf{h}_1$

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. $\mathbf{y}$: $\dfrac{\partial L}{\partial \mathbf{y}}$

Gradient of loss w.r.t. W_2 :

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

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

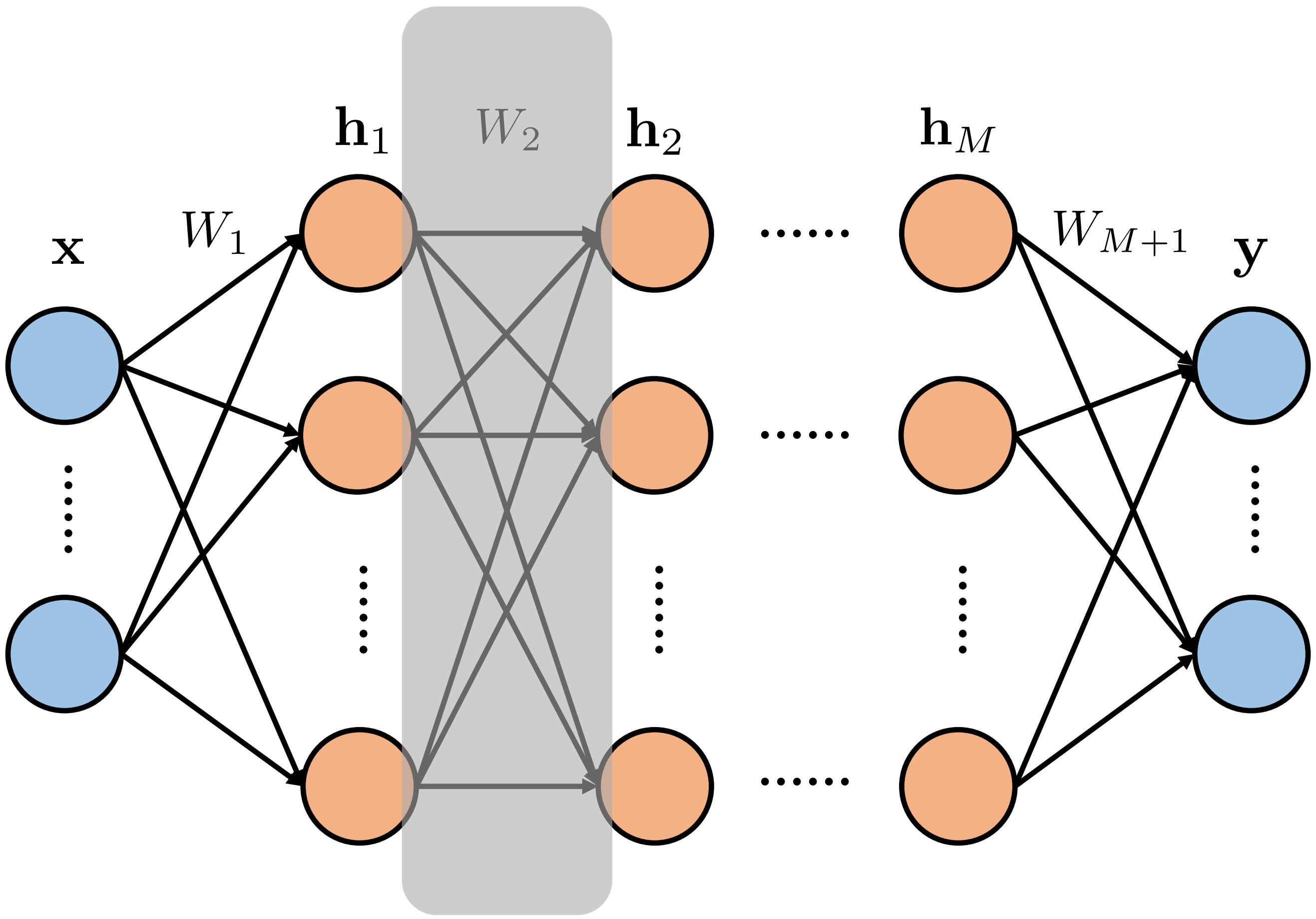
Since the nonlinearity is element-wise, we have

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

Why?

Backpropagation

During the backward pass:



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

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

Gradient of loss w.r.t. W_2 :

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

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

Since the nonlinearity is element-wise, we have

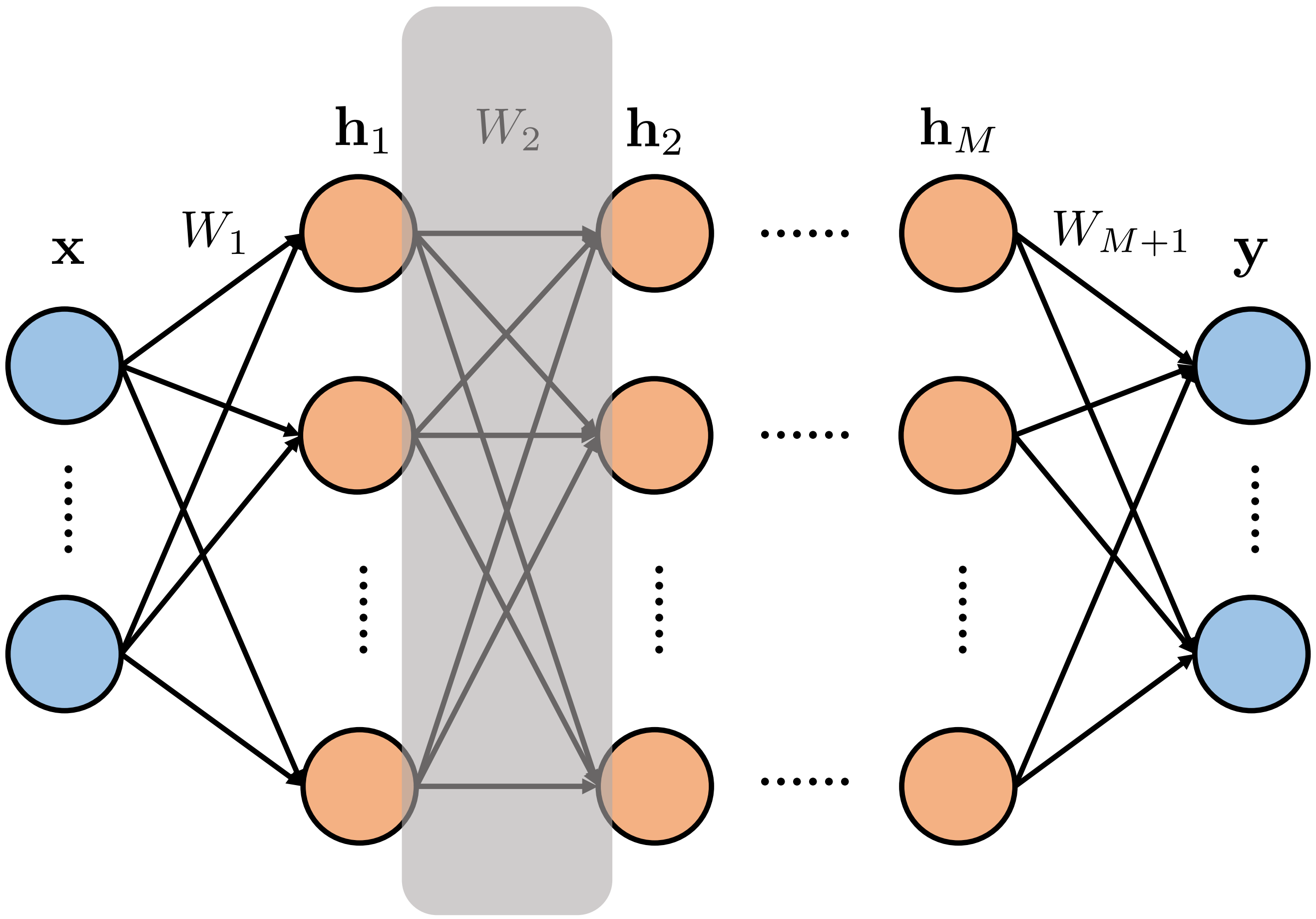
$$\frac{\partial L}{\partial \mathbf{z}_2} = \left(\frac{\partial \mathbf{h}_2}{\partial \mathbf{z}_2}\right)^\top \frac{\partial L}{\partial \mathbf{h}_2} = \sigma'(\mathbf{z}_2) \odot \frac{\partial L}{\partial \mathbf{h}_2}$$

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 :

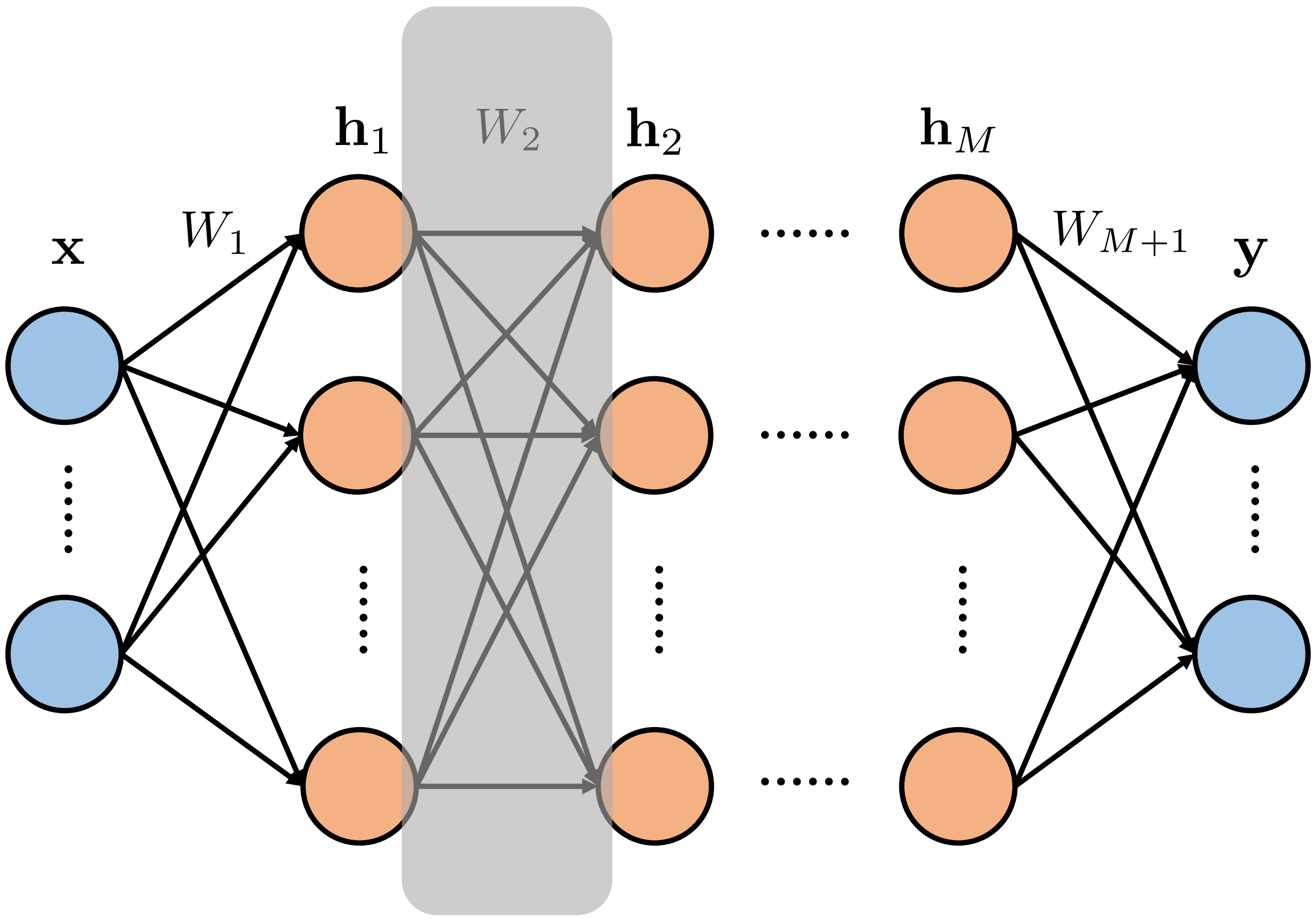
Recall $\mathbf{h}_2 = \sigma\left(W_2 \mathbf{h}_1\right)$

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

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

Backpropagation

During the backward pass:



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

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

Gradient of loss w.r.t. W_2 :

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

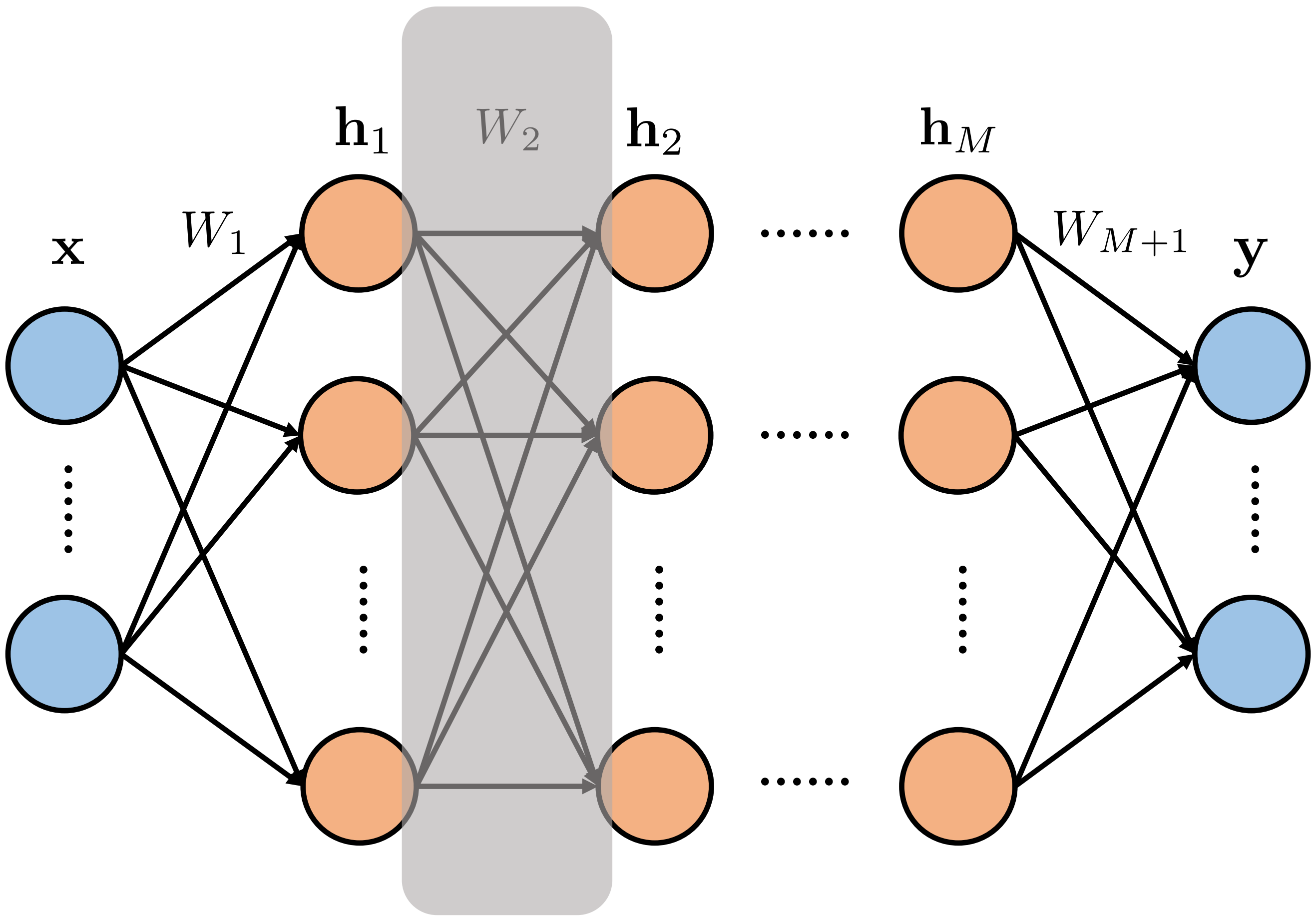
$\mathbf{z}_2 = W_2 \mathbf{h}_1$

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

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

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

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

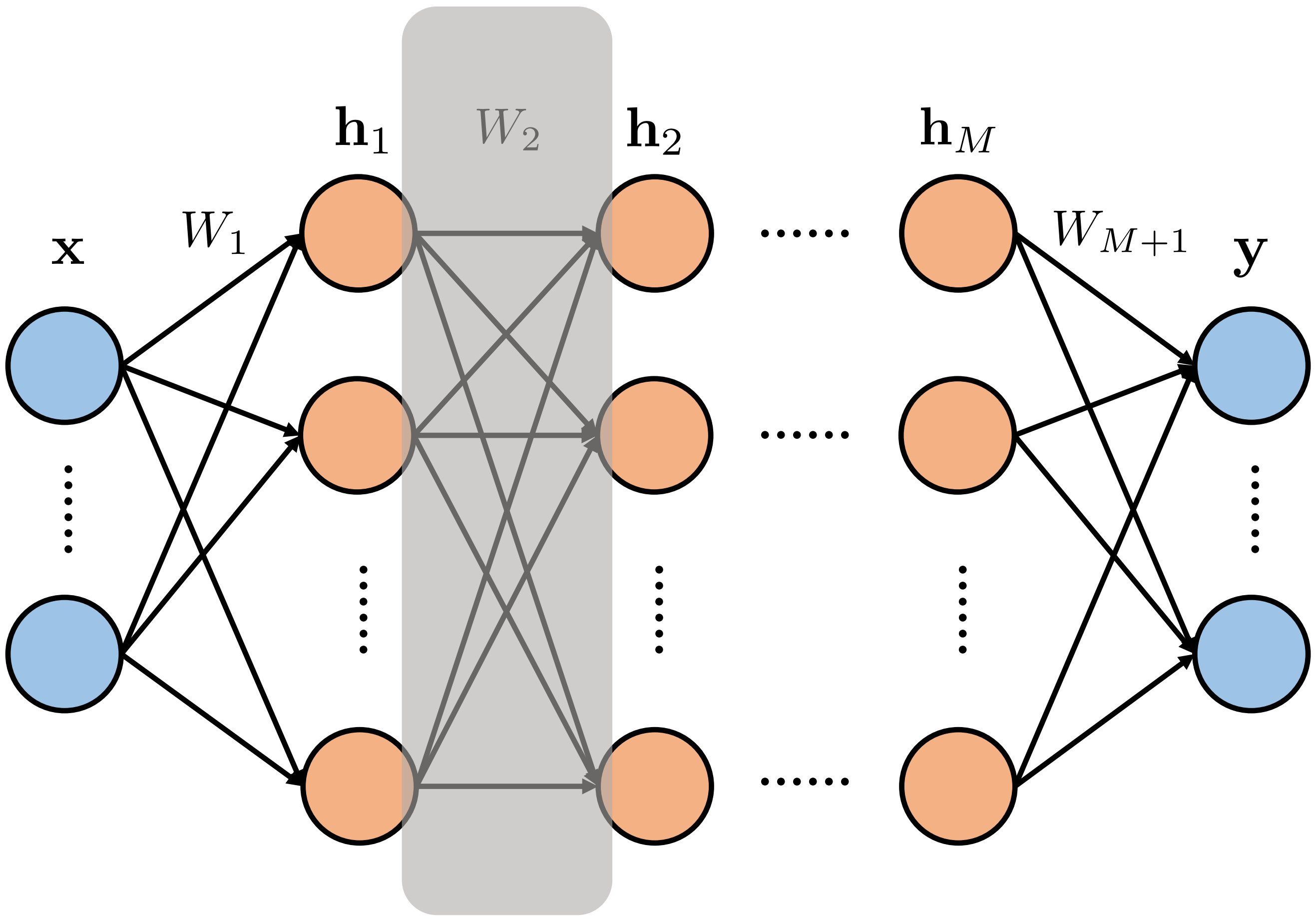
Note $\mathbf{z}_2[i] = \sum_j W_2[i,j] \mathbf{h}_1[j]$

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

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

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

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

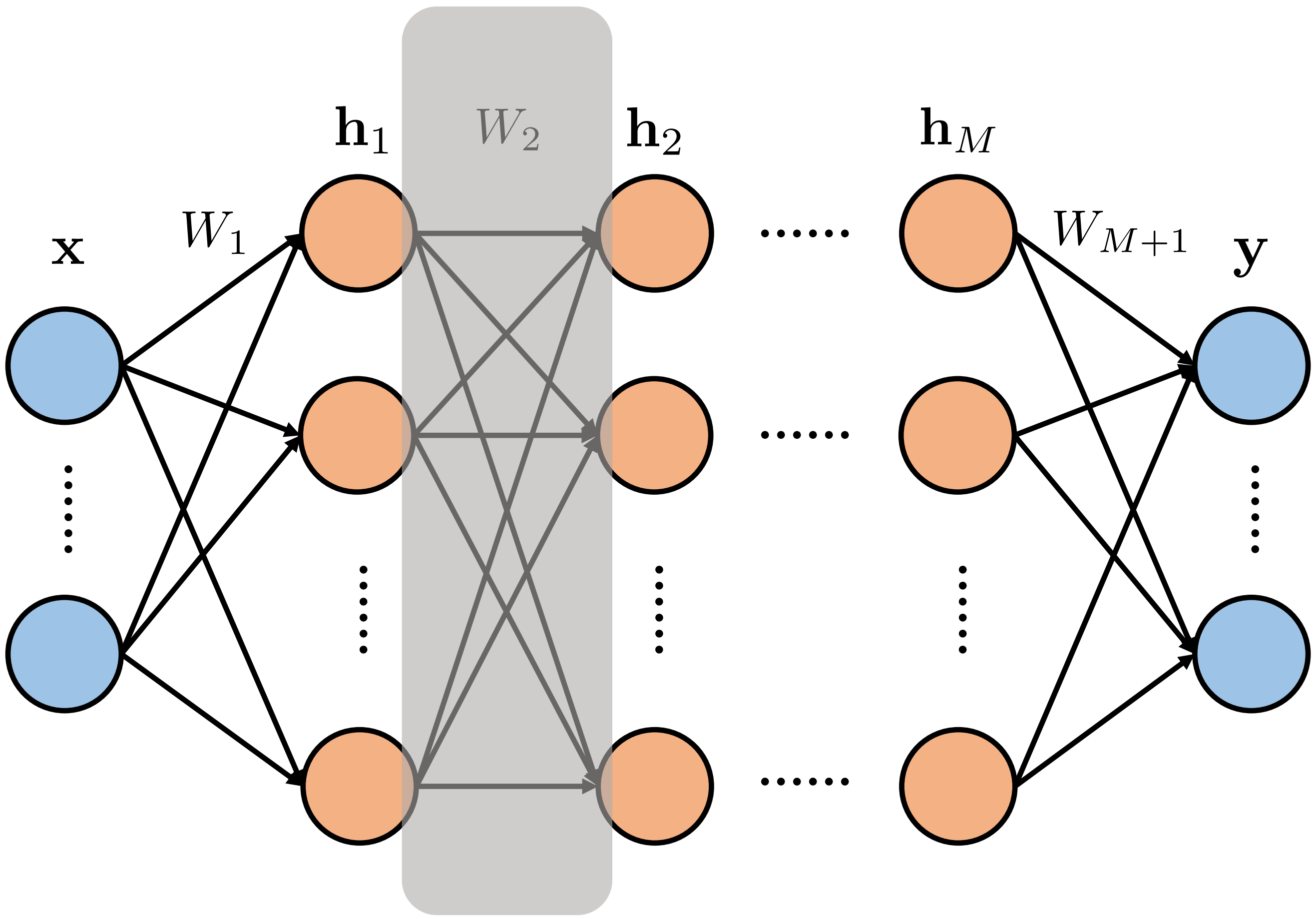
Similarly as before, we have

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

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

Backpropagation

During the backward pass:



In summary:

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

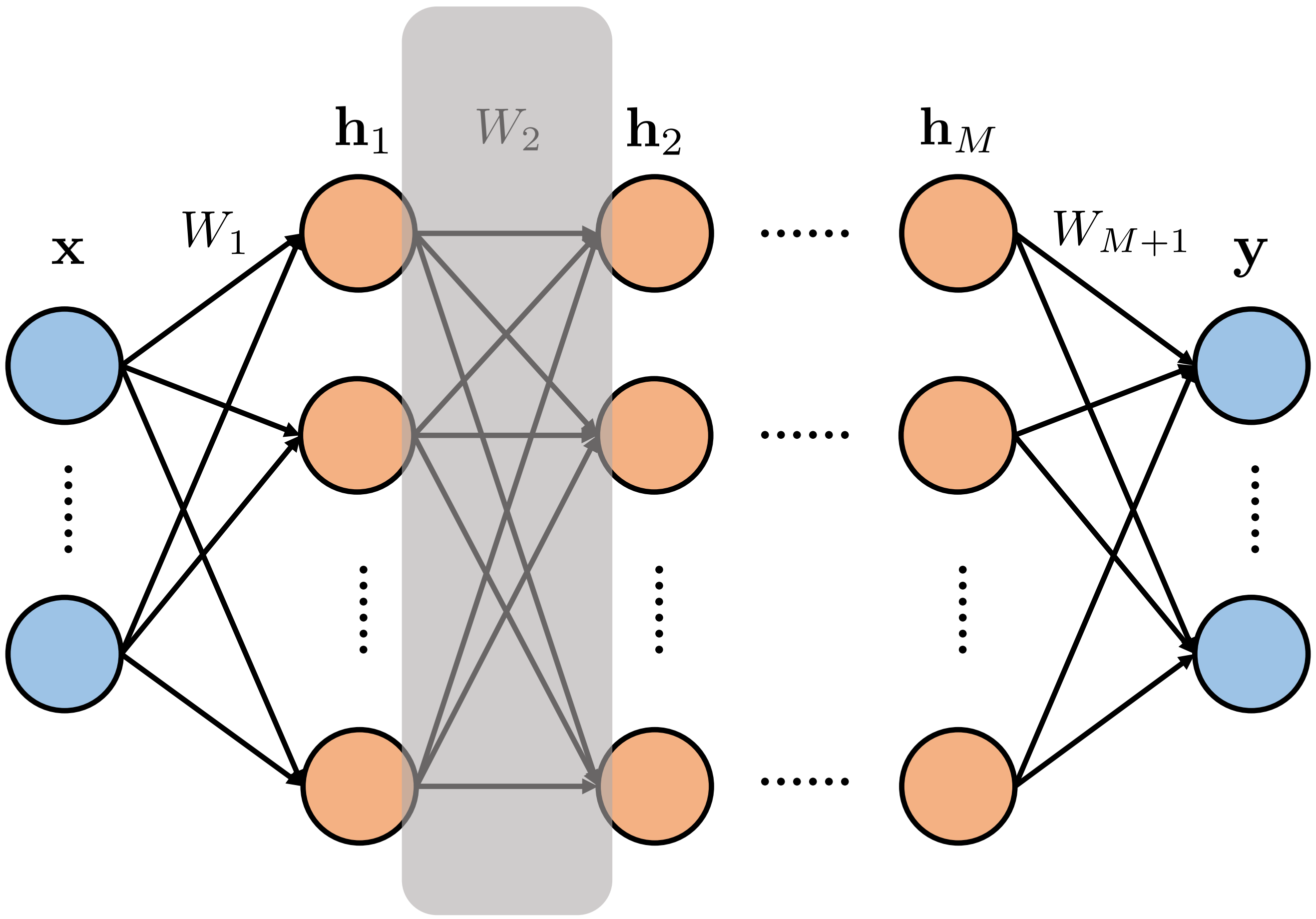
$$\mathbf{J}_i = \mathrm{diag}(\sigma'(\mathbf{z}_i)) W_i$$

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

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

Backpropagation

During the backward pass:



In summary:

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

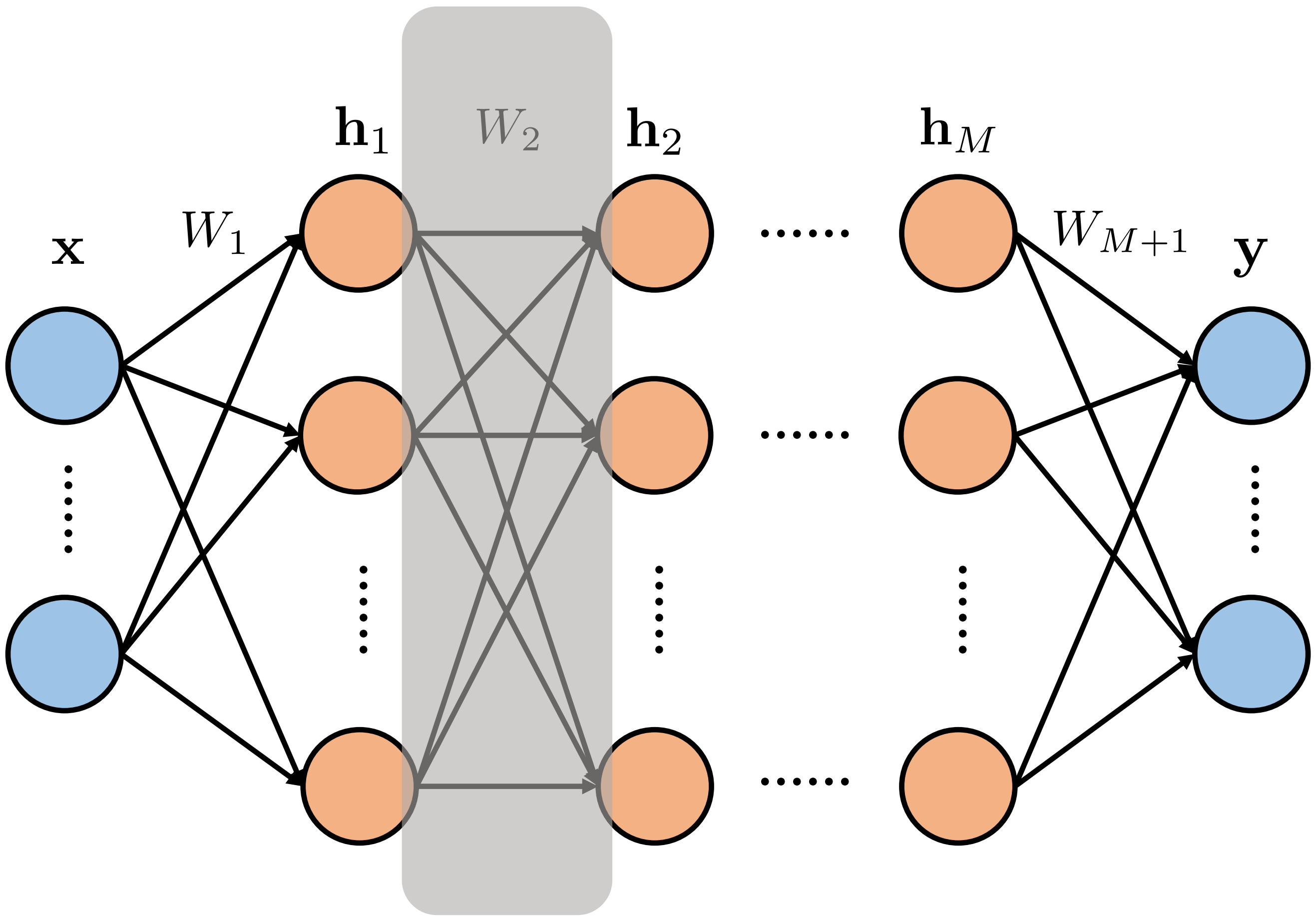
$$\mathbf{J}_i = \mathrm{diag}(\sigma'(\mathbf{z}_i))W_i$$

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

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

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

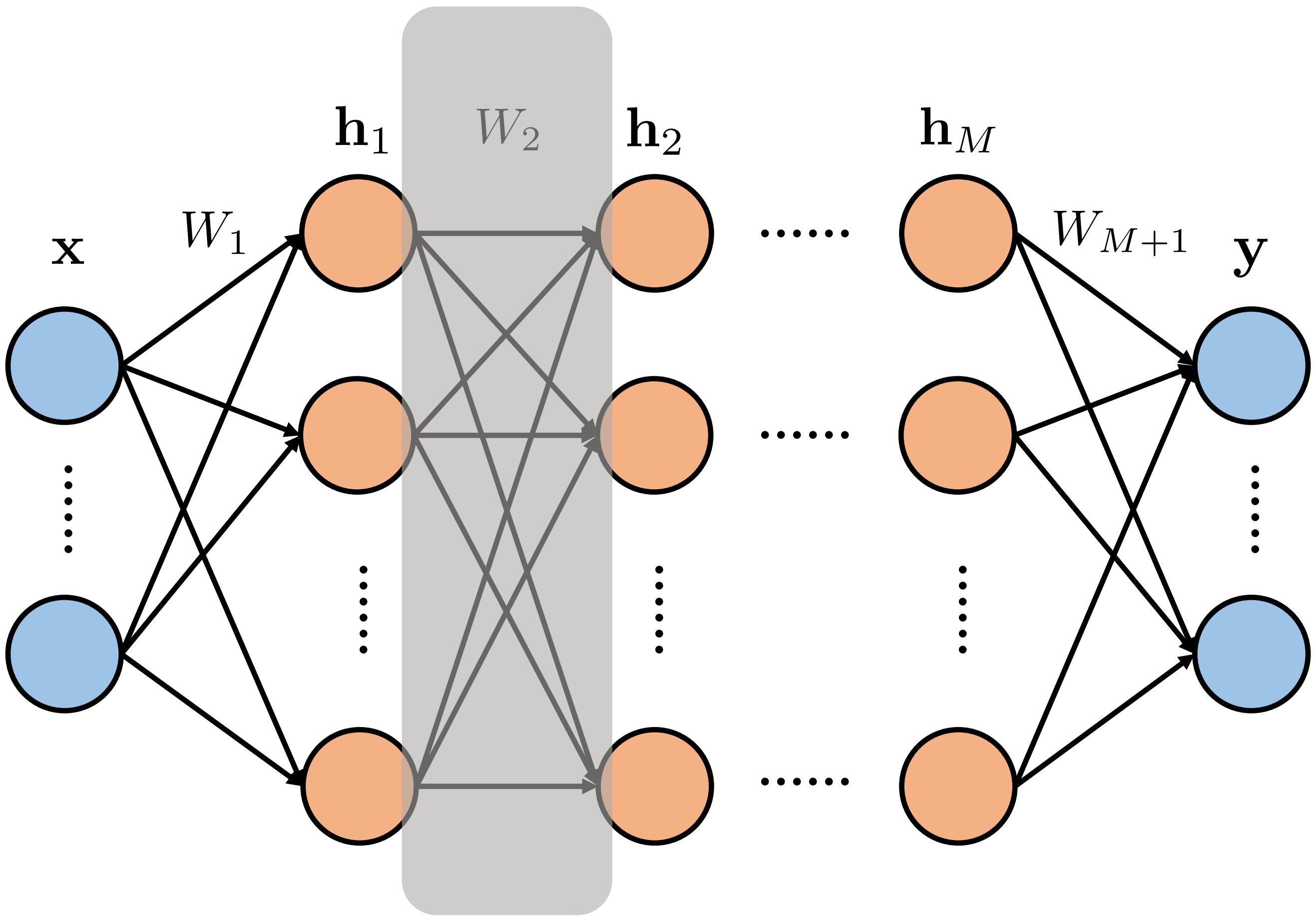
Forward vs. Backward



Computation in Forward Pass:

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

Forward vs. Backward



Computation in Forward Pass:

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

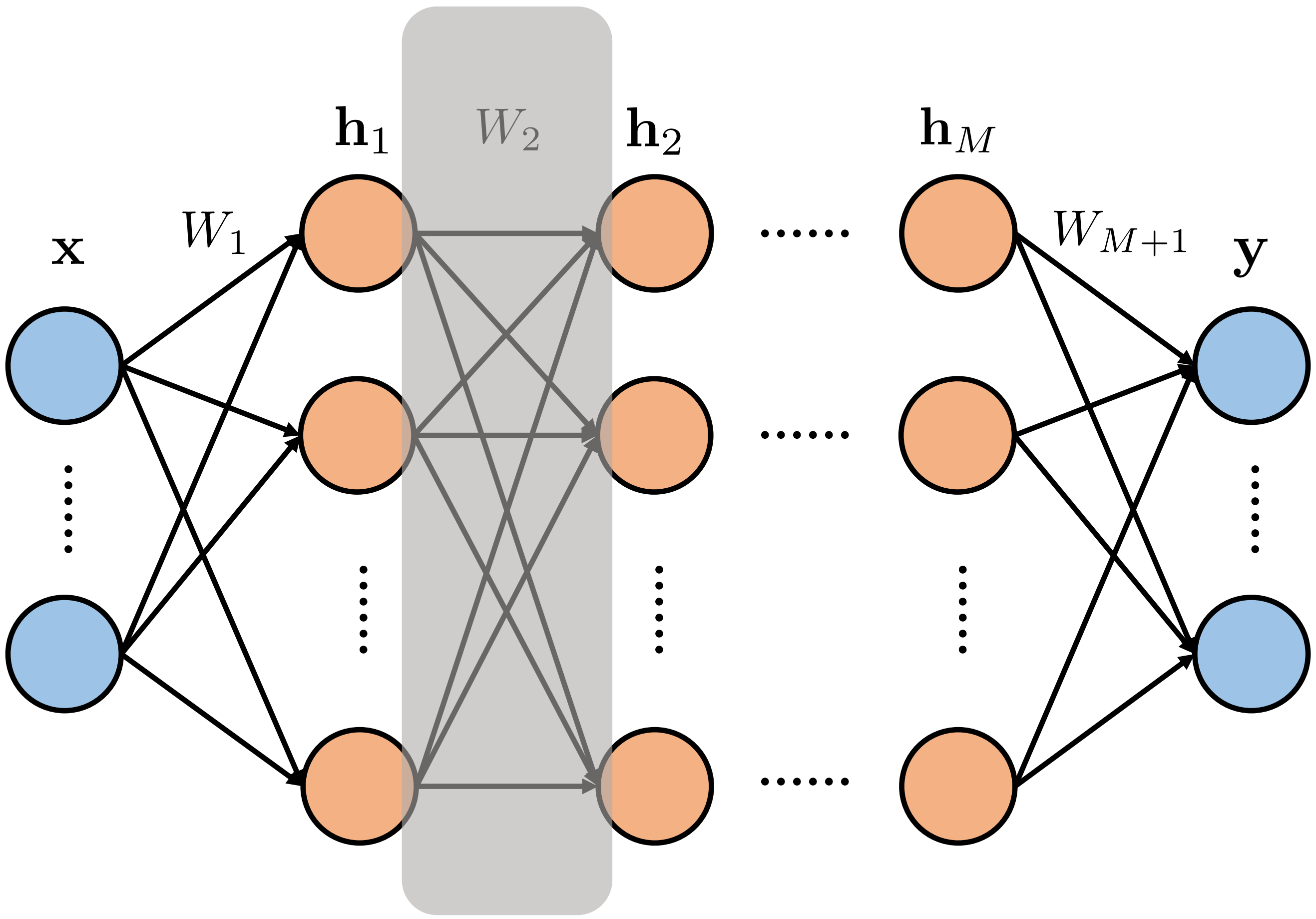
Computation in Backward Pass:

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

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

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

Forward vs. Backward



Computation in Forward Pass:

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

Computation in Backward Pass:

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

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

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

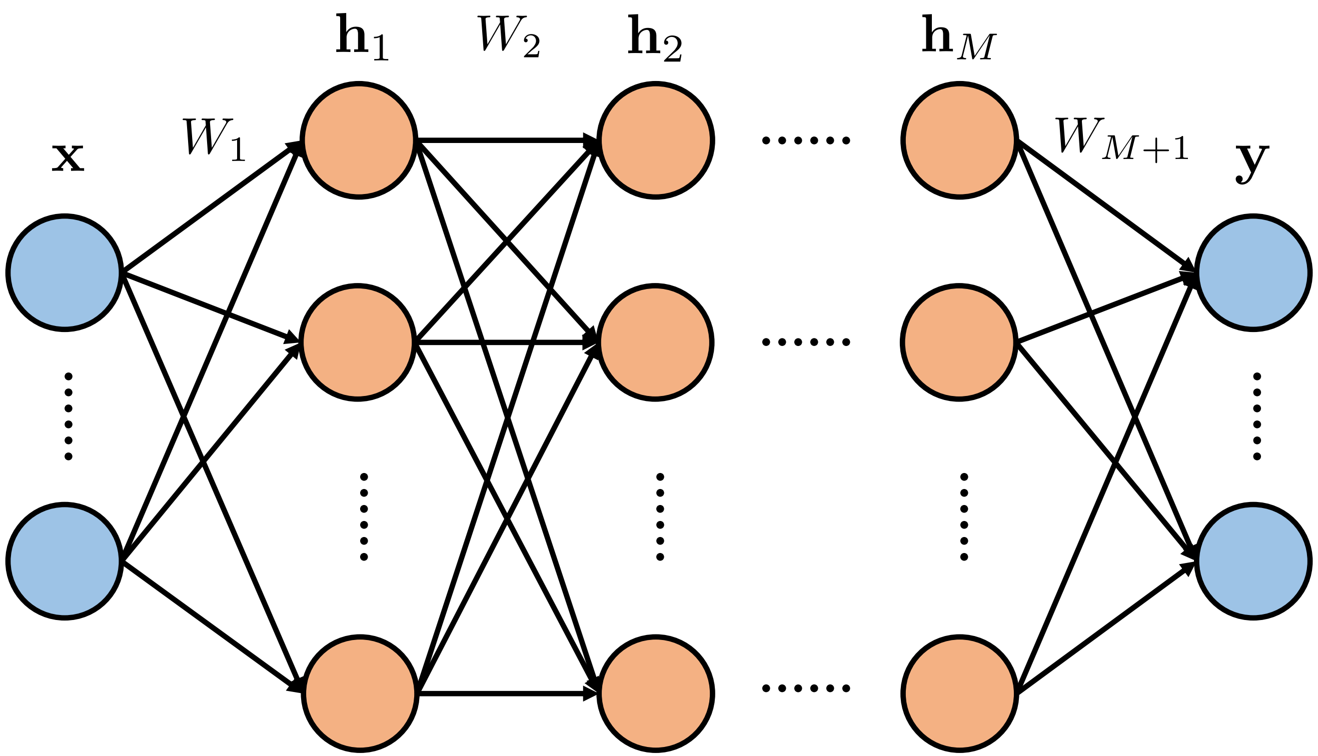
Backward pass is roughly twice as computationally expensive as Forward pass!

Outline

- Learning Algorithm for Feedforward Neural Networks:
  - Backpropagation
  - **Weight Initialization**
  - Learning Rate & Momentum & Adam
  - Weight Decay & Early Stopping

Initialization

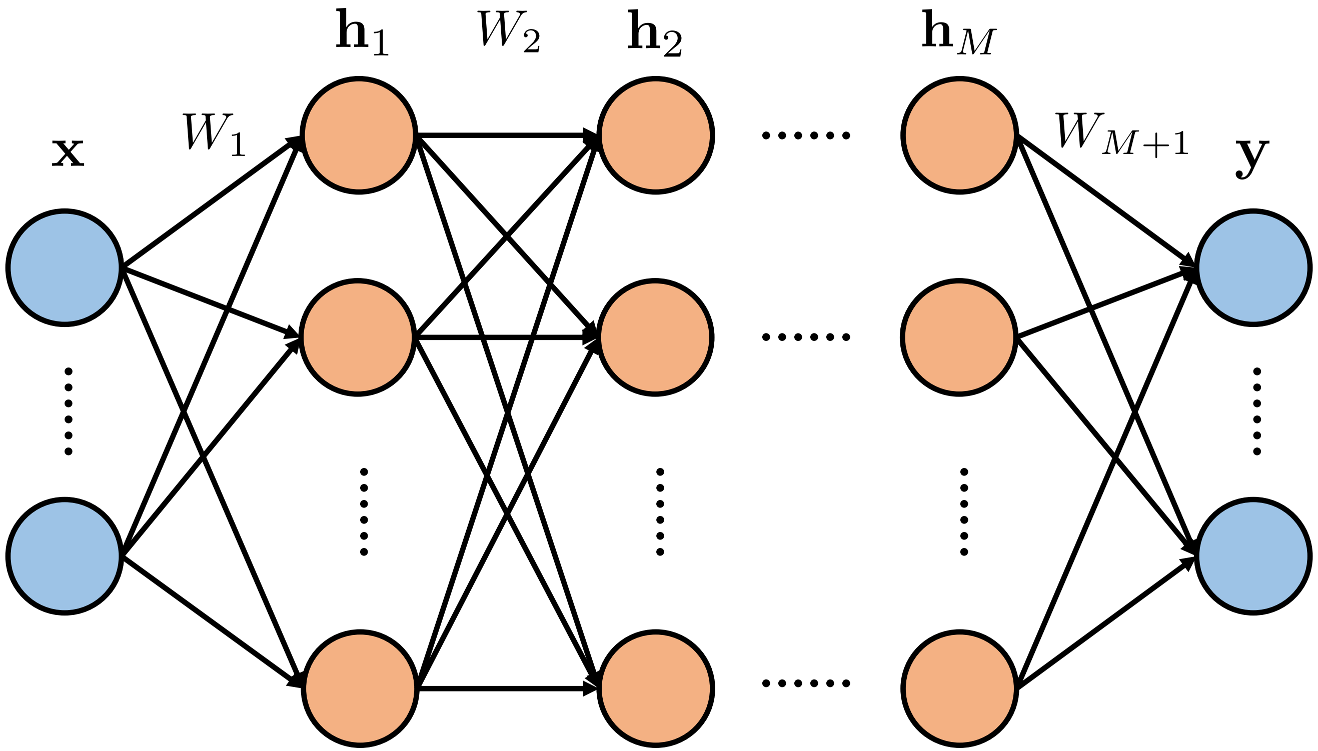
How should we initialize the neural network?



Initialization

How should we initialize the neural network?

A lot of criterions for good initialization exist, e.g., *be close to some local optima*, *have a higher chance to reach a global optima*.

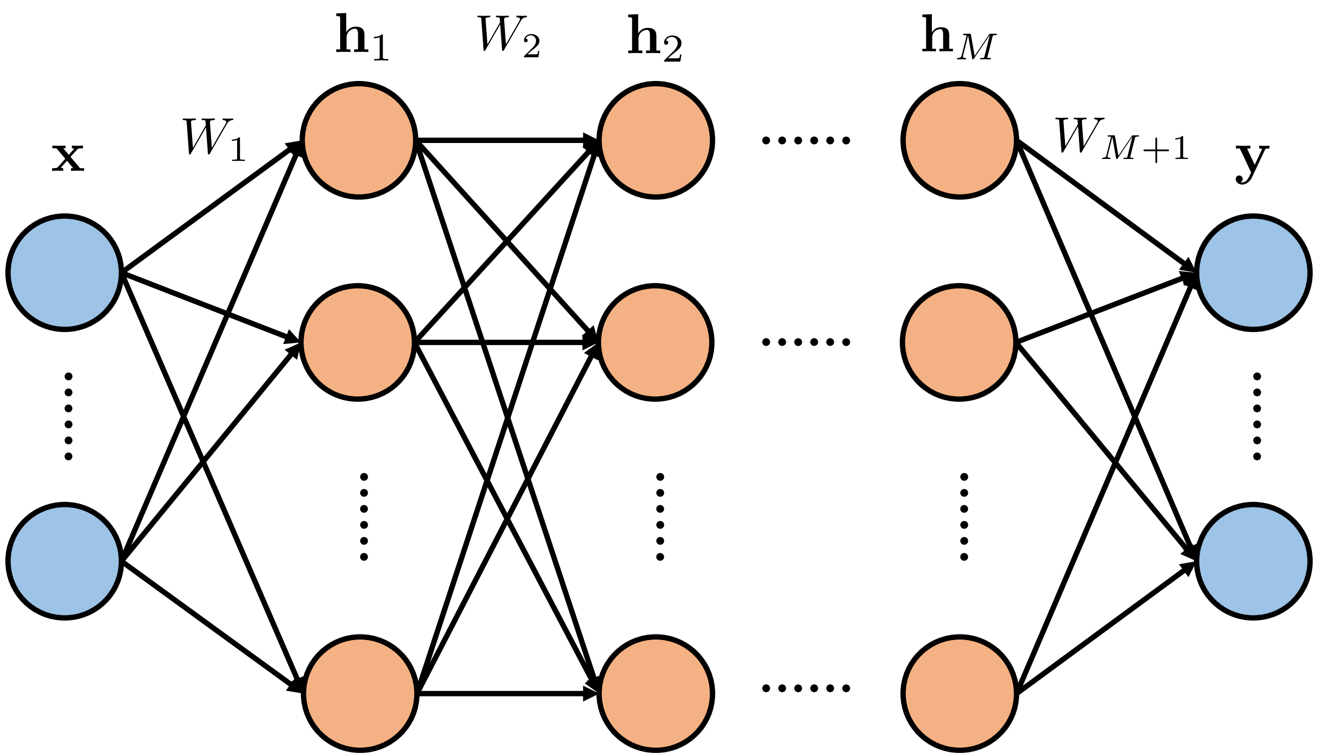


Initialization

How should we initialize the neural network?

A lot of criterions for good initialization exist, e.g., *be close to some local optima*, *have a higher chance to reach a global optima*.

But most of them are not easily computable before training the neural networks!

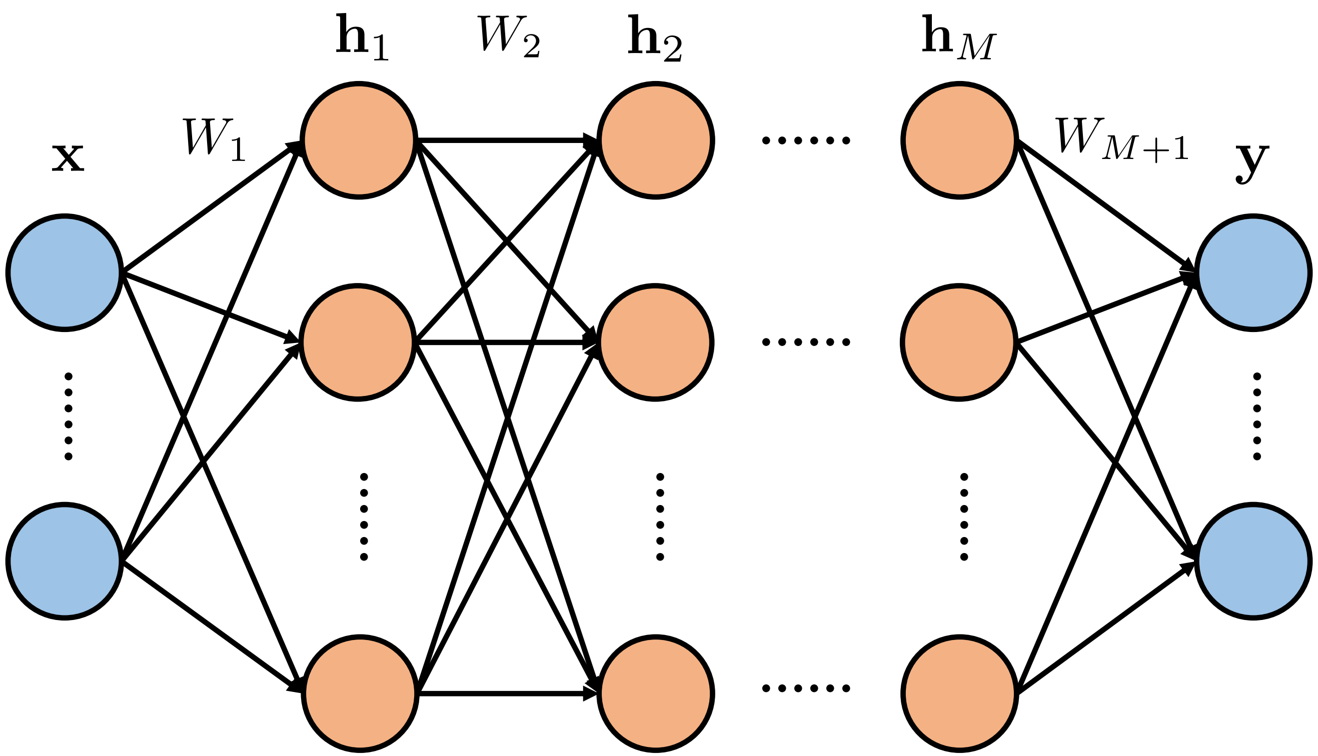


Initialization

How should we initialize the neural network?

A lot of criterions for good initialization exist, e.g., *be close to some local optima*, *have a higher chance to reach a global optima*.

But most of them are not easily computable before training the neural networks!



One computable criterion is:

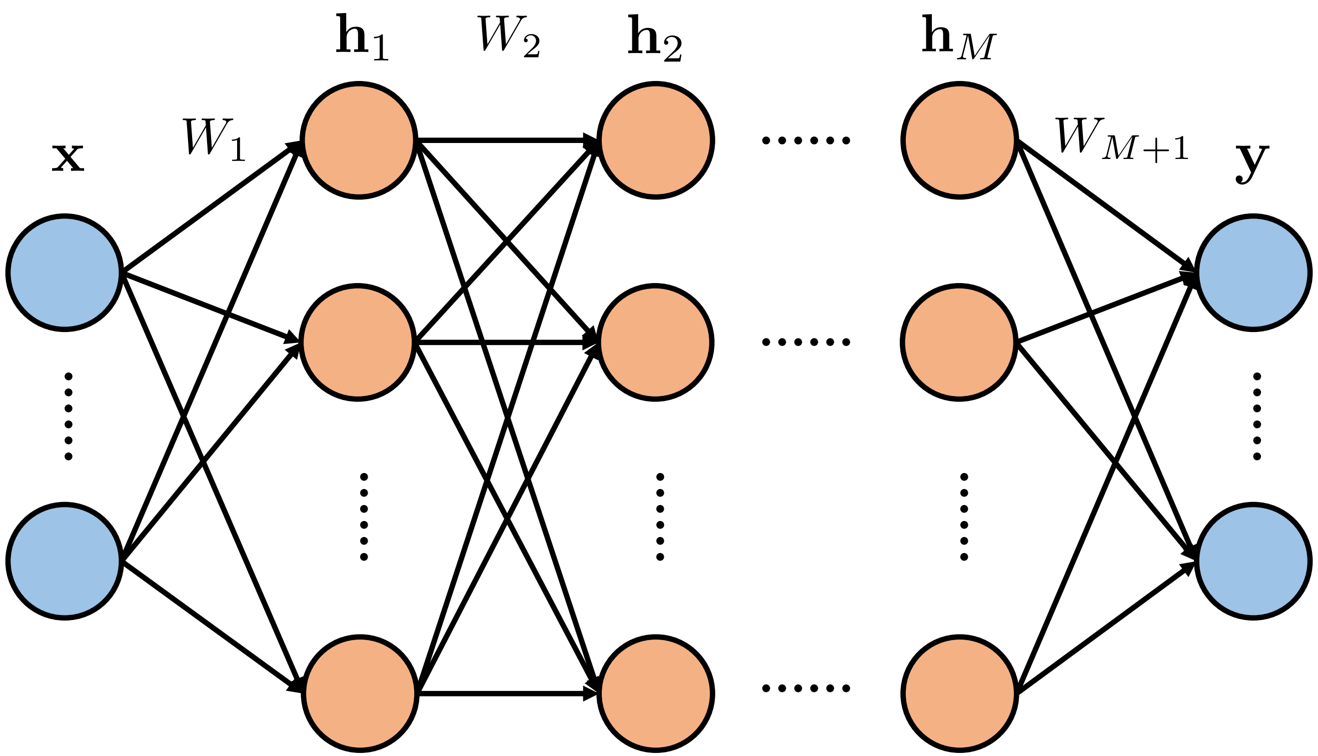
We want to start with some **stable** initial neural network!

Initialization

How should we initialize the neural network?

A lot of criterions for good initialization exist, e.g., *be close to some local optima*, *have a higher chance to reach a global optima*.

But most of them are not easily computable before training the neural networks!



One computable criterion is:

We want to start with some **stable** initial neural network!

There are also many stability notions. Let us look at the variance of the activations and gradients.

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

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

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

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

$$\begin{aligned}
\mathbb{E}[xy] &= \mathbb{E}[x]\mathbb{E}[y] \\
\mathbb{E}[x^2y^2] &= \mathbb{E}[x^2]\mathbb{E}[y^2]
\end{aligned}$$

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

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

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

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

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

Initialization

Let us recap some basic facts about variance

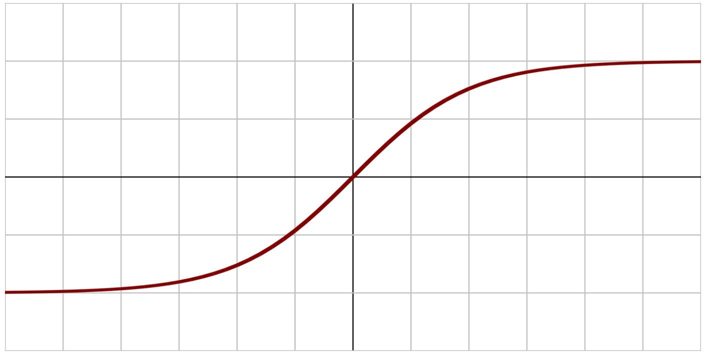
$$\mathbb{V}[x] = \mathbb{E}\left[(x - \mathbb{E}[x])^2\right] = \mathbb{E}[x^2] - (\mathbb{E}[x])^2$$

In summary, for two independent random variables, we have

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

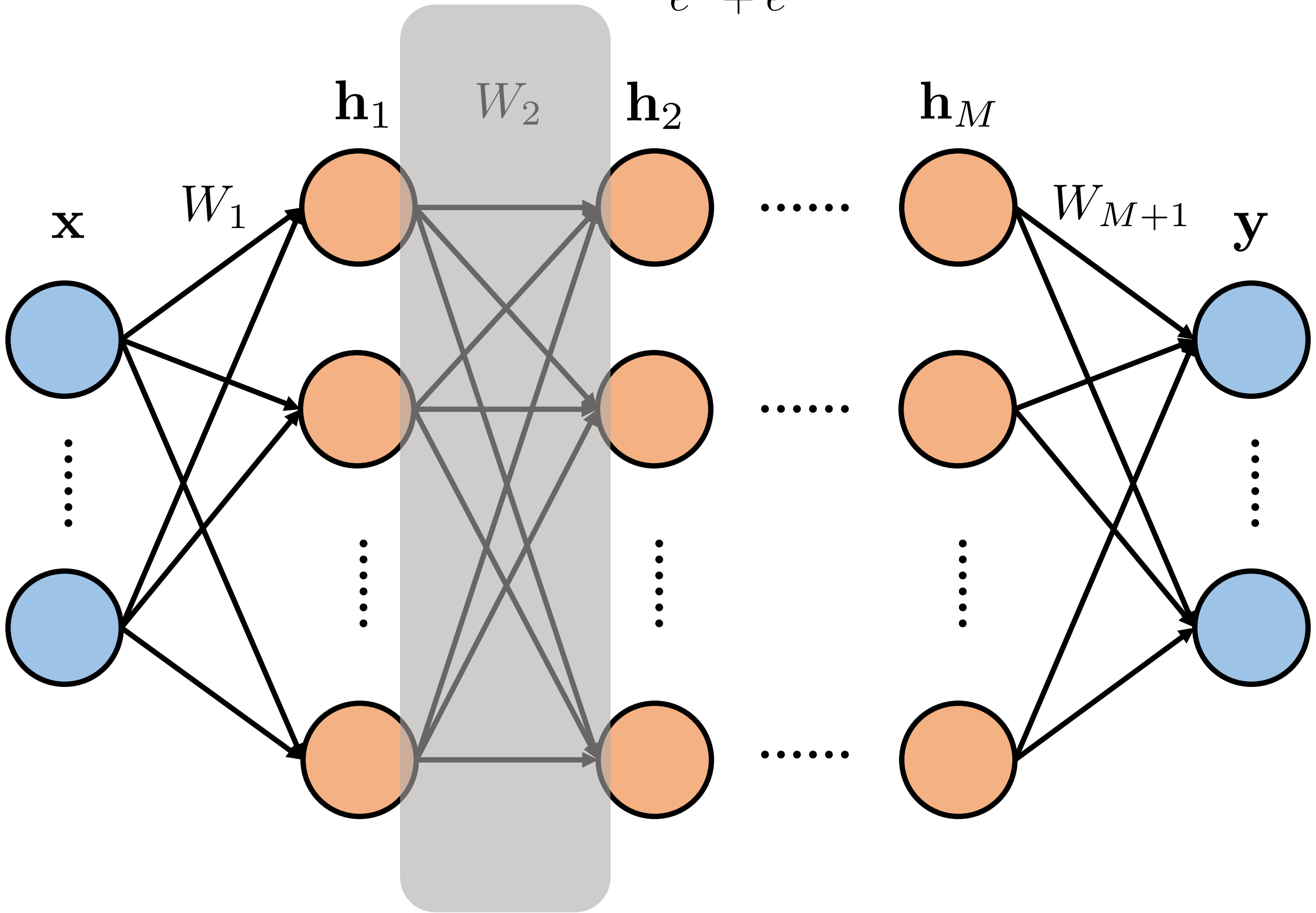
If $\mathbb{E}[x] = \mathbb{E}[y] = 0$, then $\mathbb{V}[xy] = \mathbb{V}[x]\mathbb{V}[y]$

Initialization: Forward Analysis

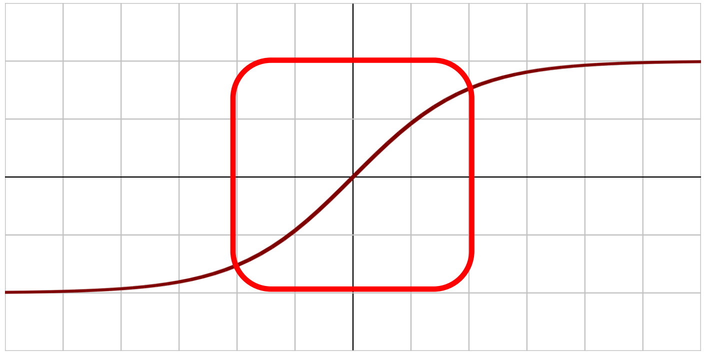


Recall $\mathbf{h}_2 = \sigma\left(W_2 \mathbf{h}_1\right)$

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

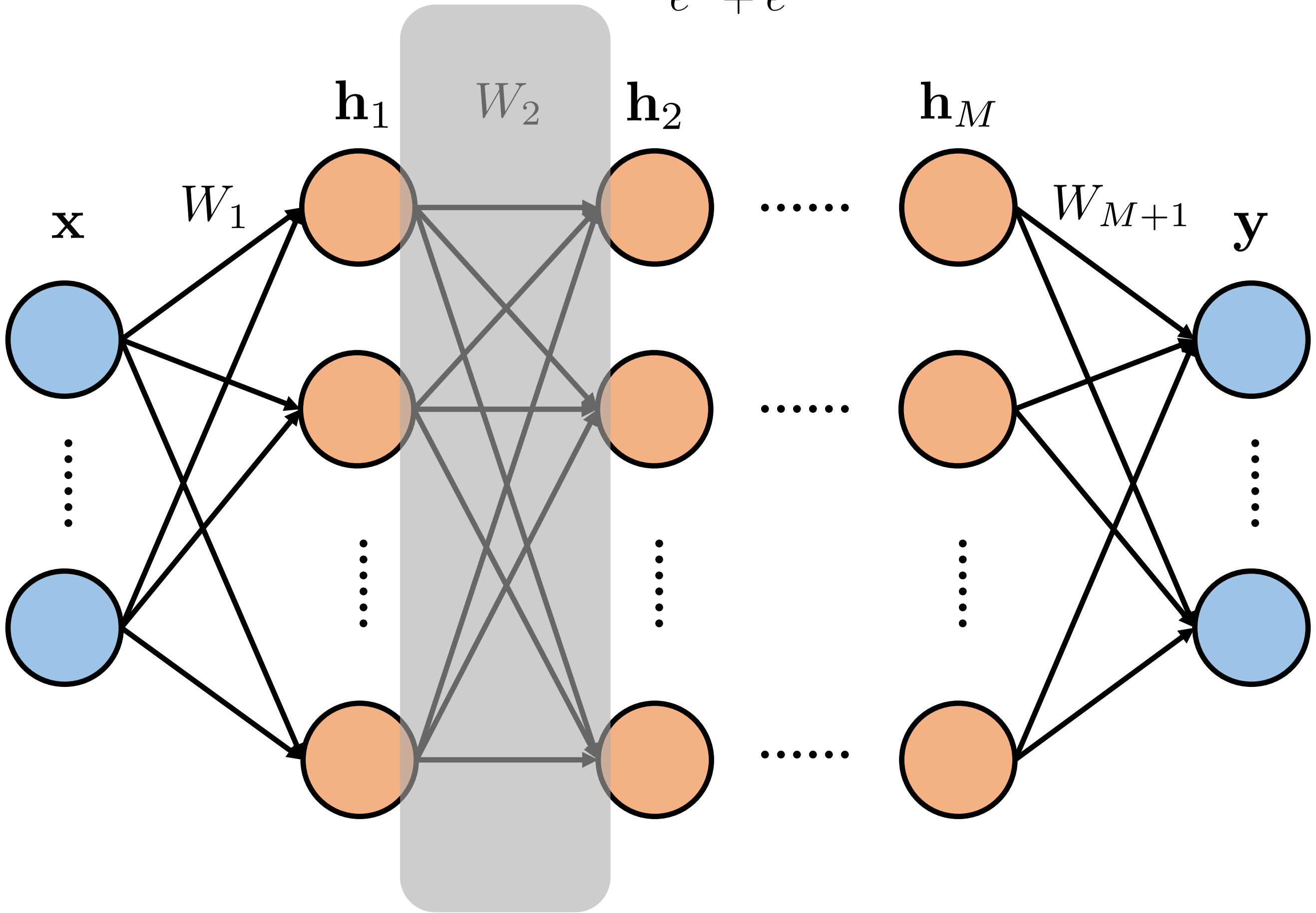


Initialization: Forward Analysis



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

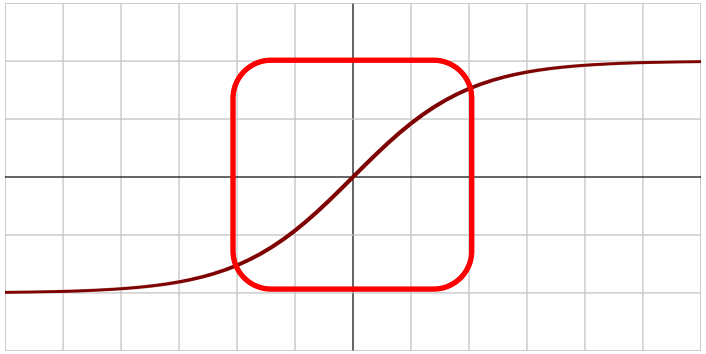
If we assume Tanh activation $\sigma(x) = \dfrac{e^x - e^{-x}}{e^x + e^{-x}}$ and we are in the linear regime



Initialization: Forward Analysis

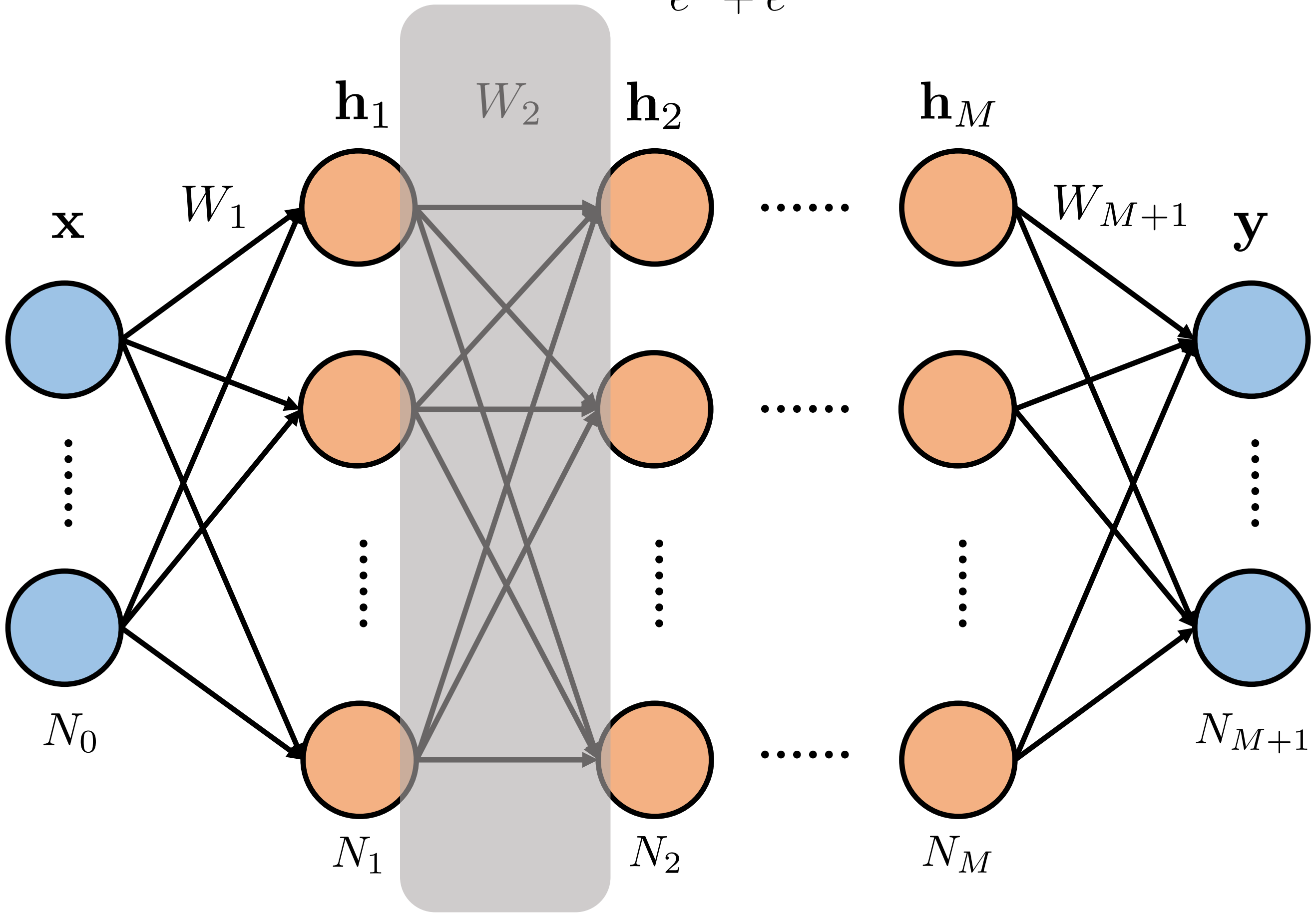
Recall $\mathbf{h}_2 = \sigma\left(W_2 \mathbf{h}_1\right)$

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



Then we have

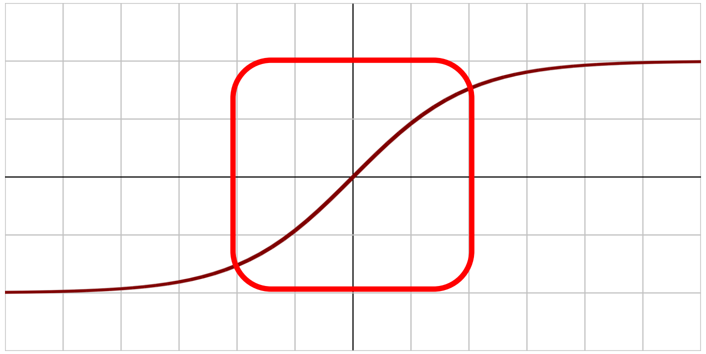
$$\mathbf{h}_2[i] = \sigma\Big(\sum_{j}^{N_1} W_2[i,j]\mathbf{h}_1[j]\Big) \approx \sum_{j=1}^{N_1} W_2[i,j]\mathbf{h}_1[j]$$

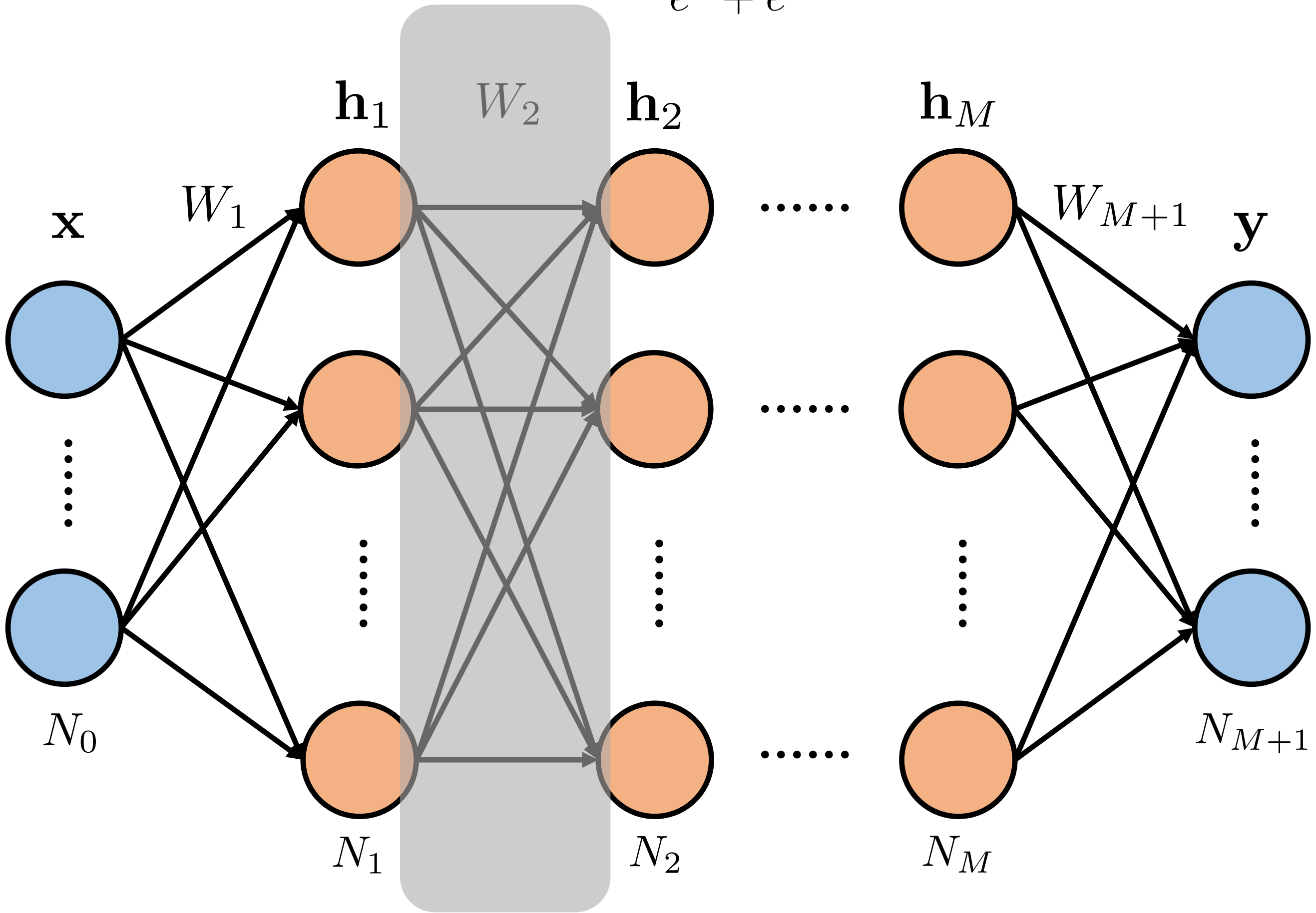


Initialization: Forward Analysis

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





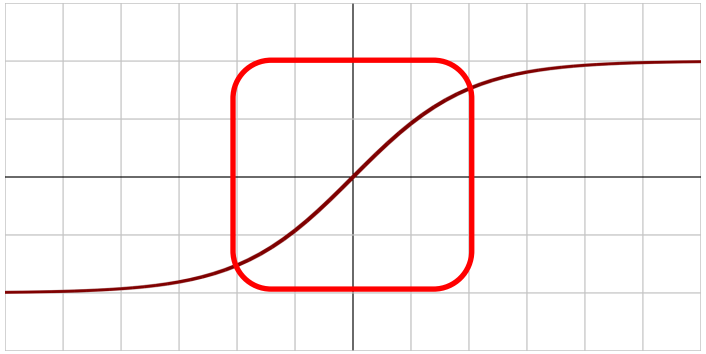
Then we have

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

We further assume i.i.d. and zero mean with same variance for all activations $\mathbf{h}_1[j]$ and all weights $W_2[i,j]$ at all layers, then

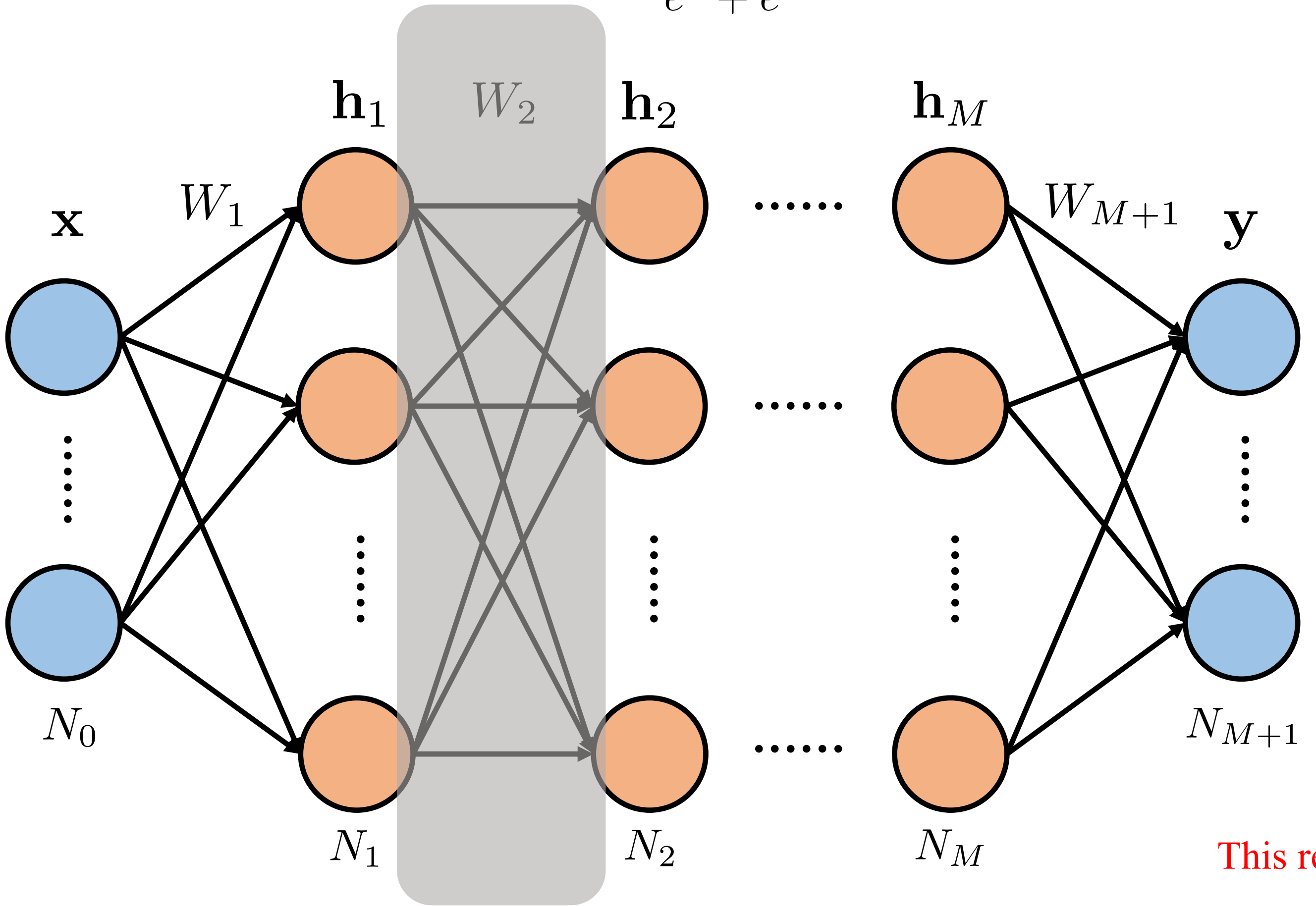
$$\begin{aligned}
\mathbb{V}\left[\mathbf{h}_2[i]\right] &\approx \mathbb{V}\left[\sum_{j=1}^{N_1} W_2[i,j]\mathbf{h}_1[j]\right] \\
&= \sum_{j=1}^{N_1} \mathbb{V}\left[W_2[i,j]\mathbf{h}_1[j]\right] \\
&= \sum_{j=1}^{N_1} \mathbb{V}\left[W_2[i,j]\right]\mathbb{V}\left[\mathbf{h}_1[j]\right] \\
&= N_1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

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



Then we have

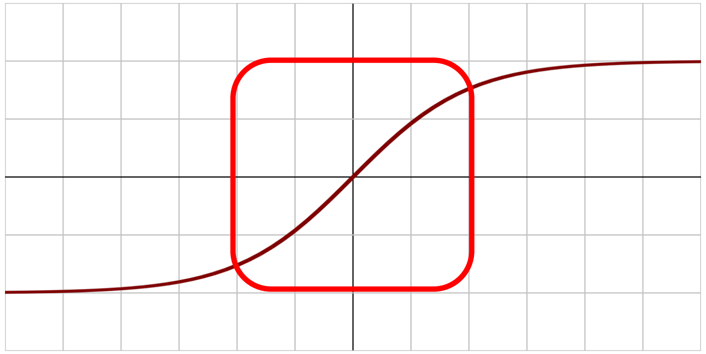
$$\mathbf{h}_2[i] = \sigma\Big(\sum_{j}^{N_1} W_2[i,j]\mathbf{h}_1[j]\Big) \approx \sum_{j=1}^{N_1} W_2[i,j]\mathbf{h}_1[j]$$

We further assume i.i.d. and zero mean with same variance for all activations $\mathbf{h}_1[j]$ and all weights $W_2[i,j]$ at all layers, then

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

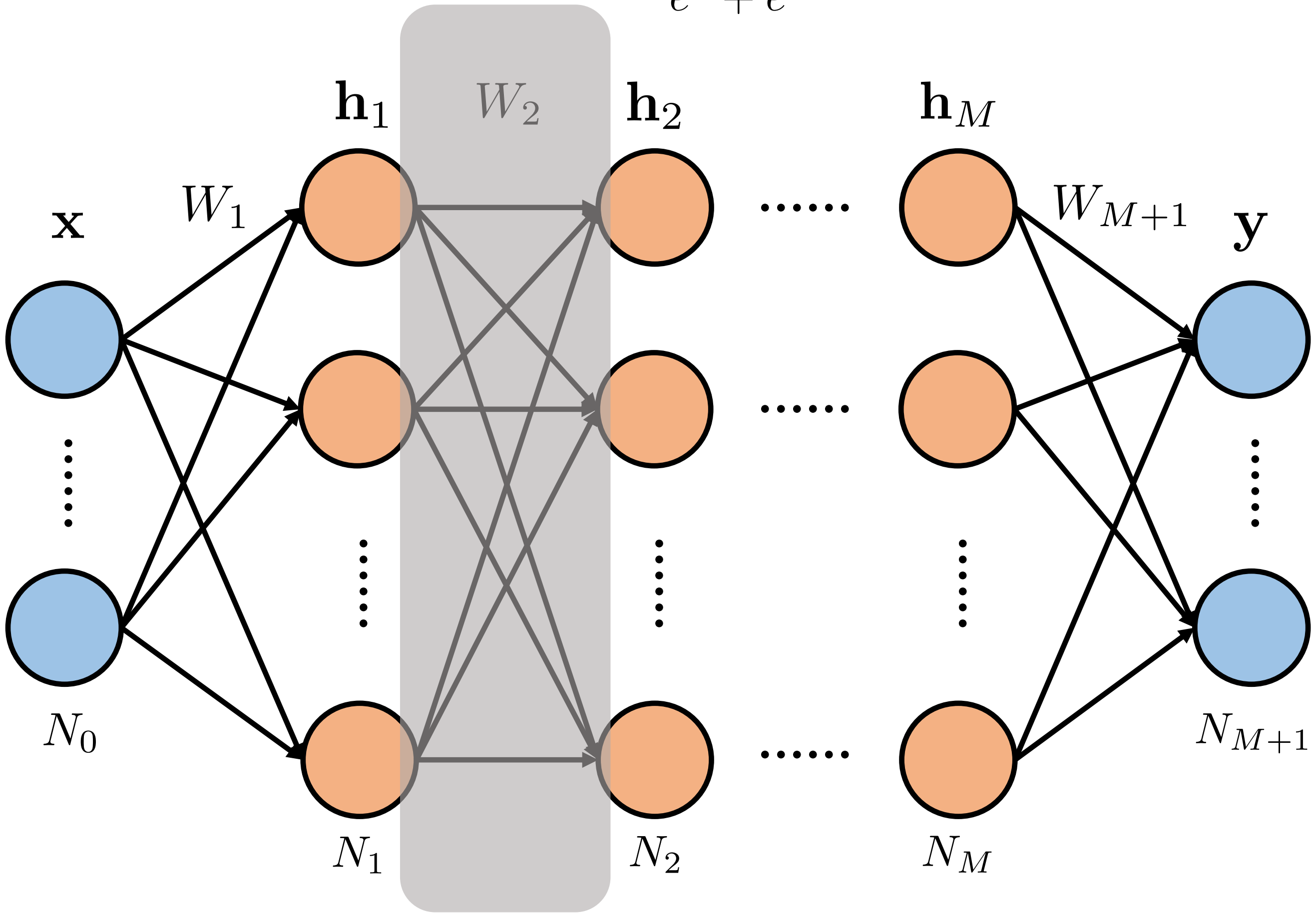
This relation holds for all layers given the assumptions!

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

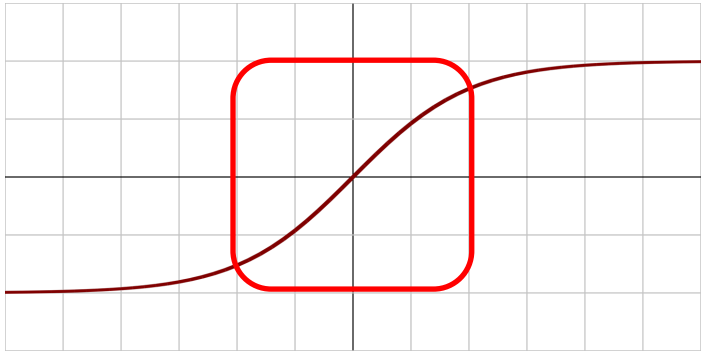


Then unroll the recursion, we have

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

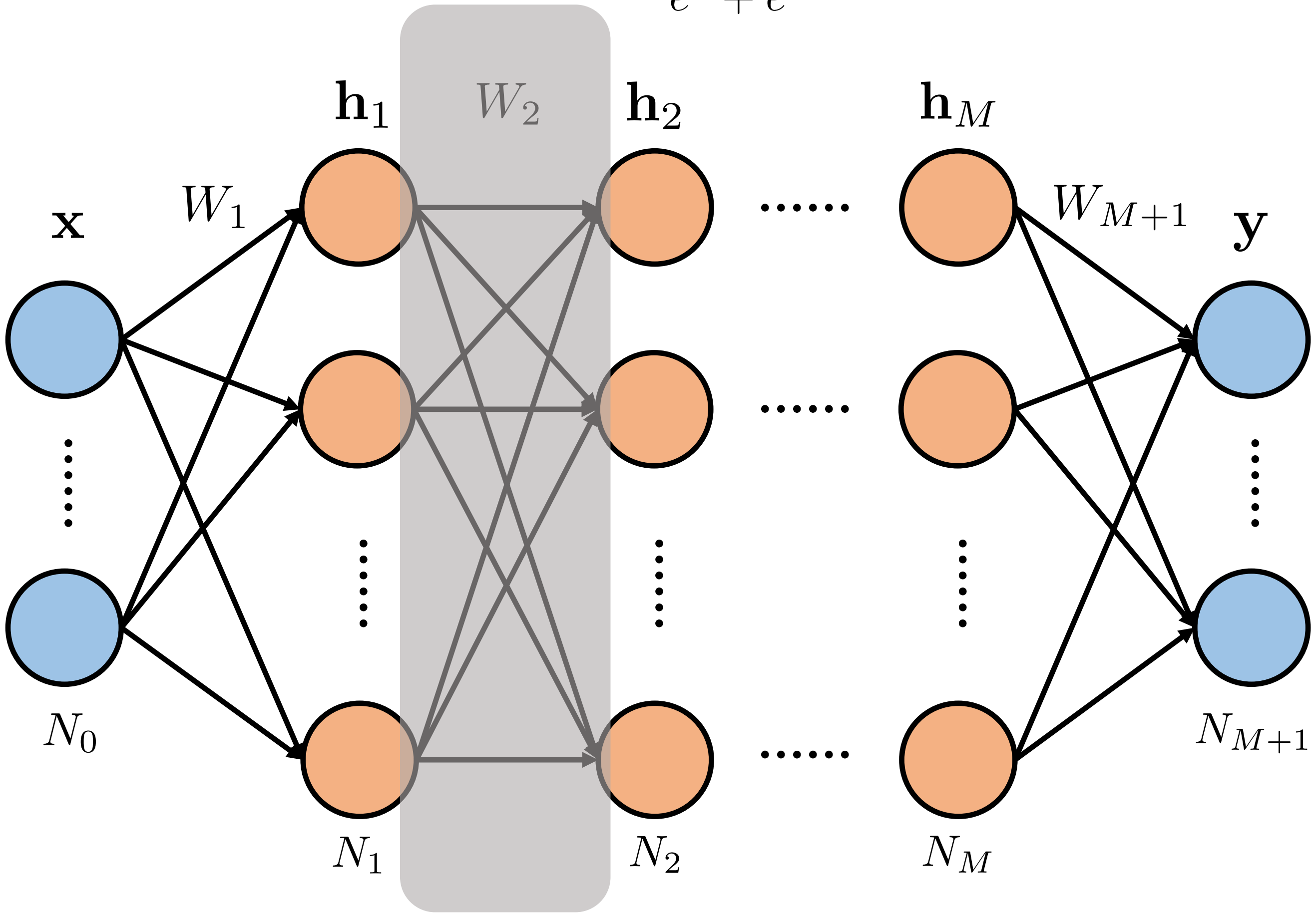
Note i and j here are arbitrary since we assume all activations have the same variance and all weights have the same variance as well!

Initialization: Forward Analysis



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

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



Then unroll the recursion, we have

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

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!

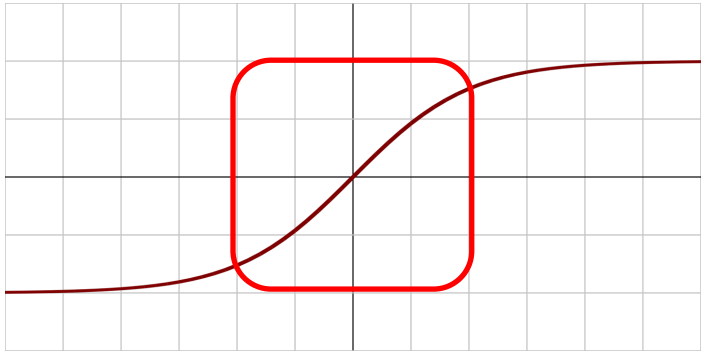
To preserve the variance of activations through forward pass, i.e.,

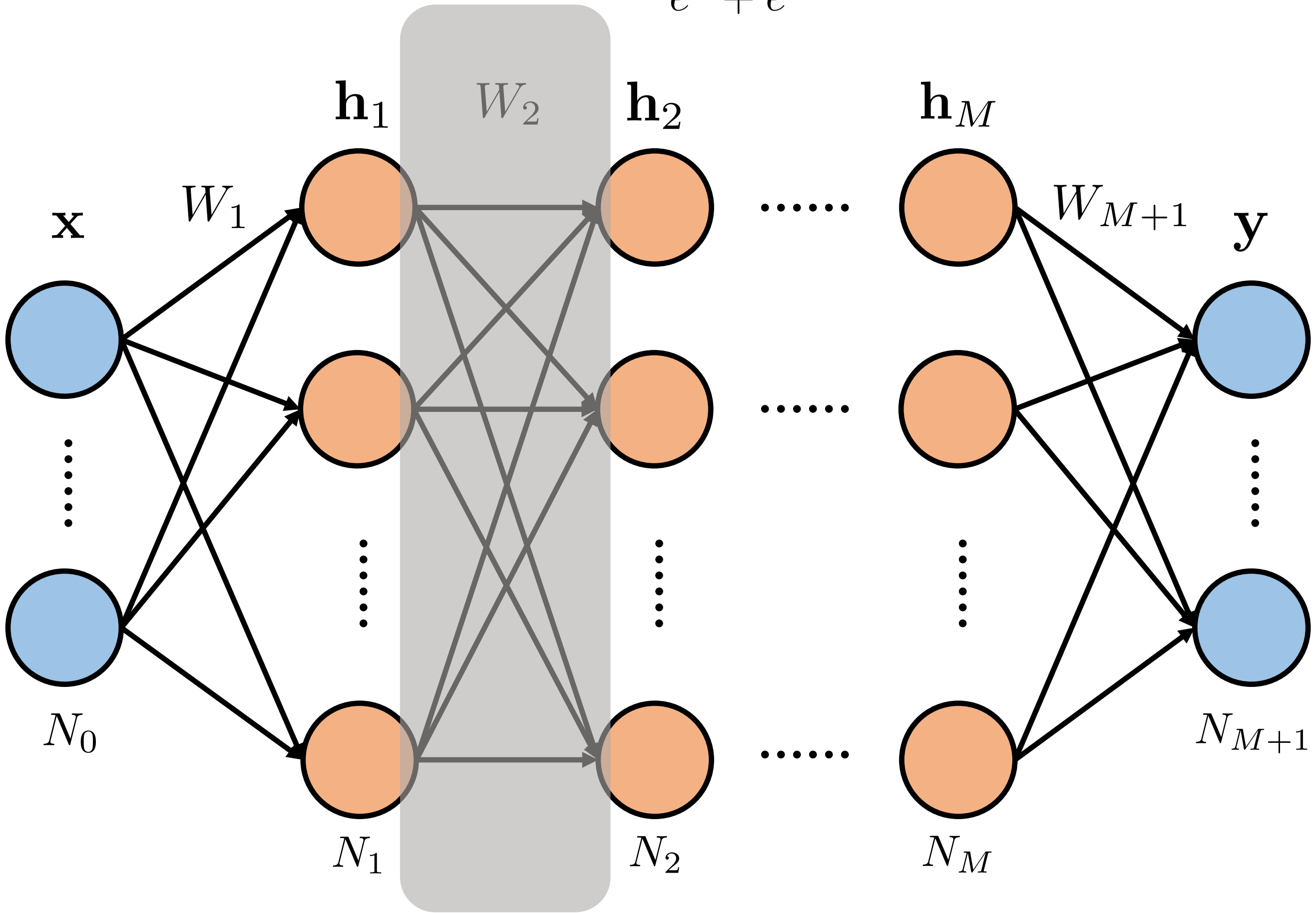
$$\mathbb{V}\left[\mathbf{h}_l[i]\right] \approx \mathbb{V}\left[\mathbf{x}[j]\right]$$

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





Then unroll the recursion, we have

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

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!

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

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

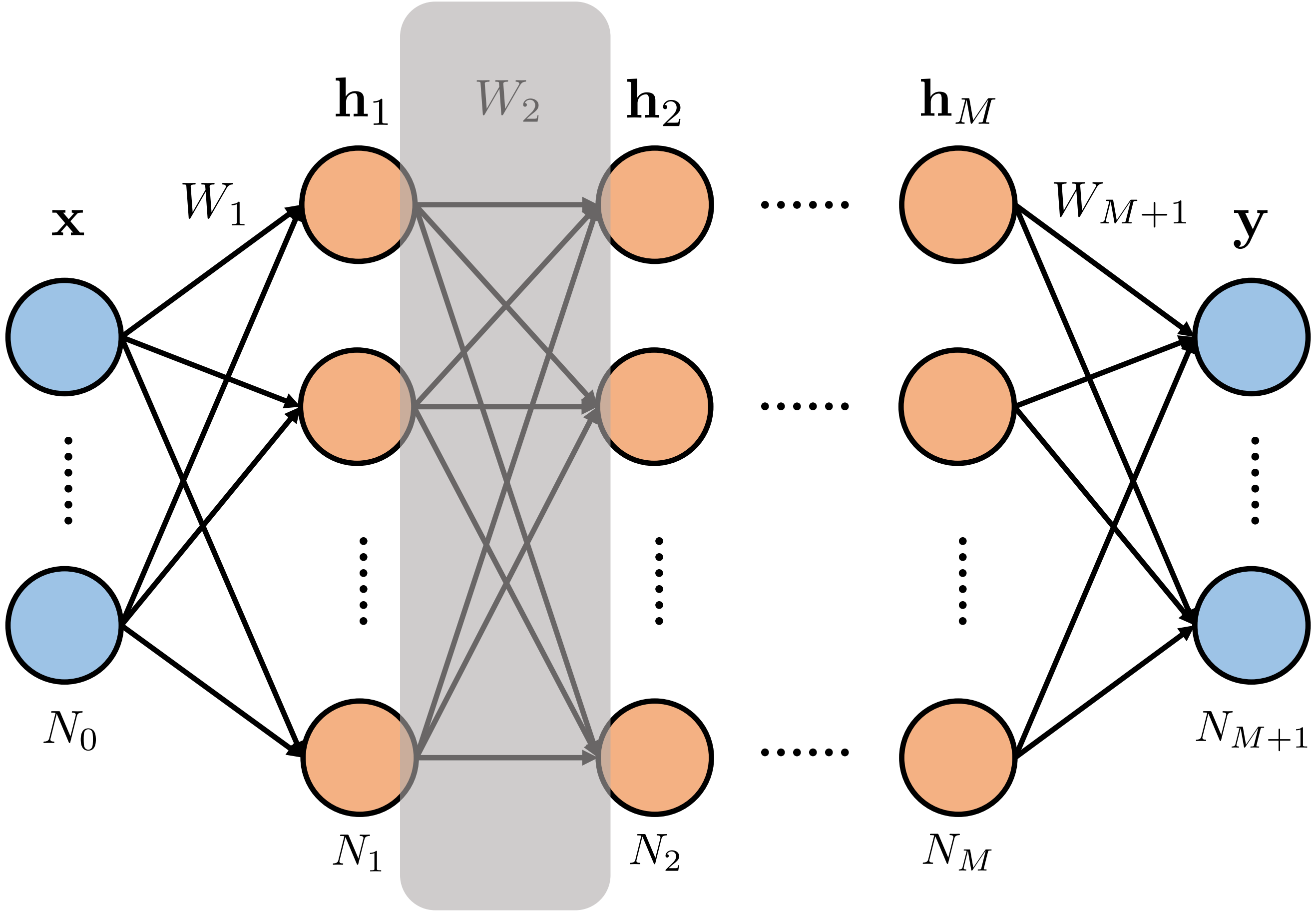
we can simply set:

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

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



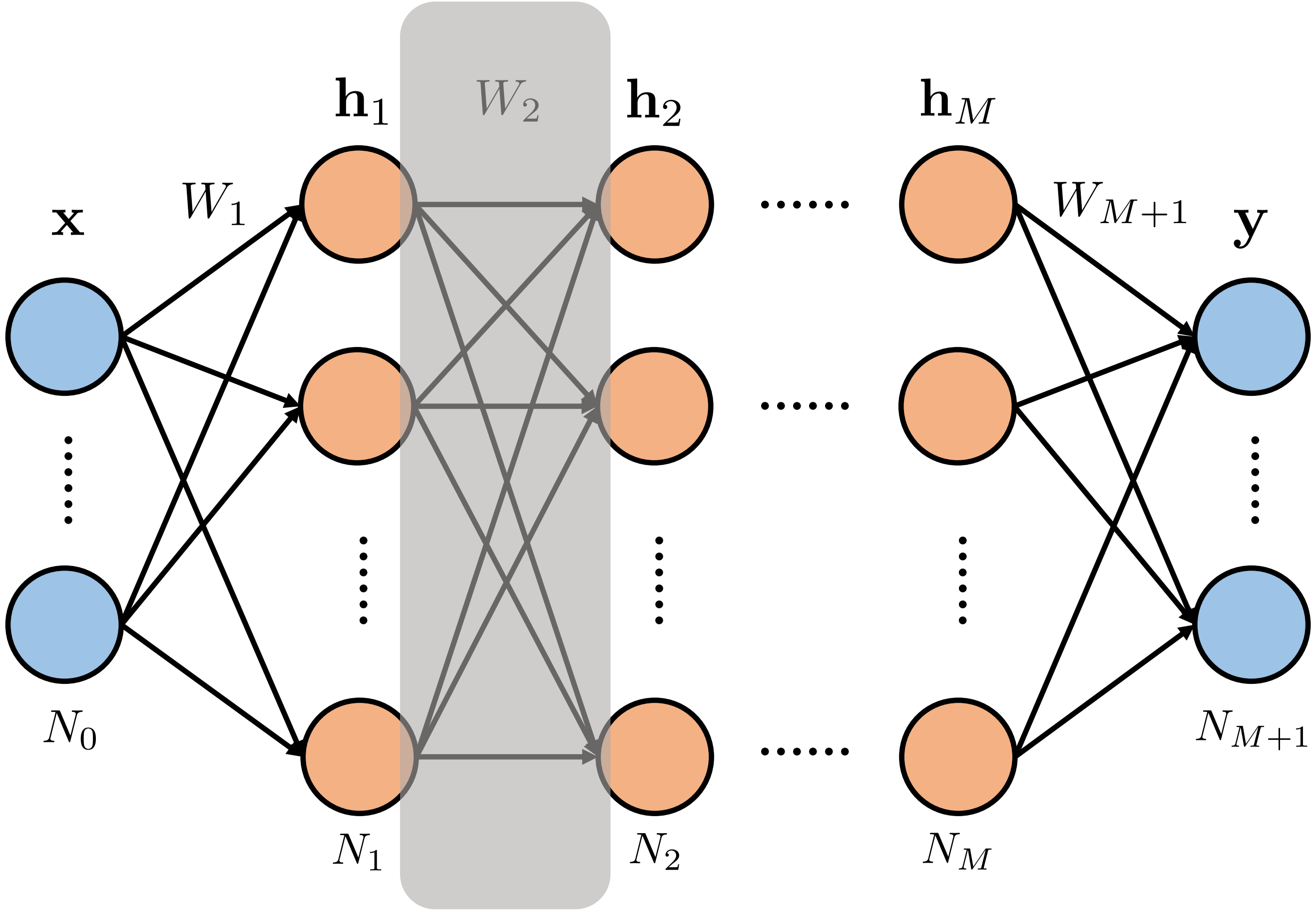
Initialization: Backward Analysis

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

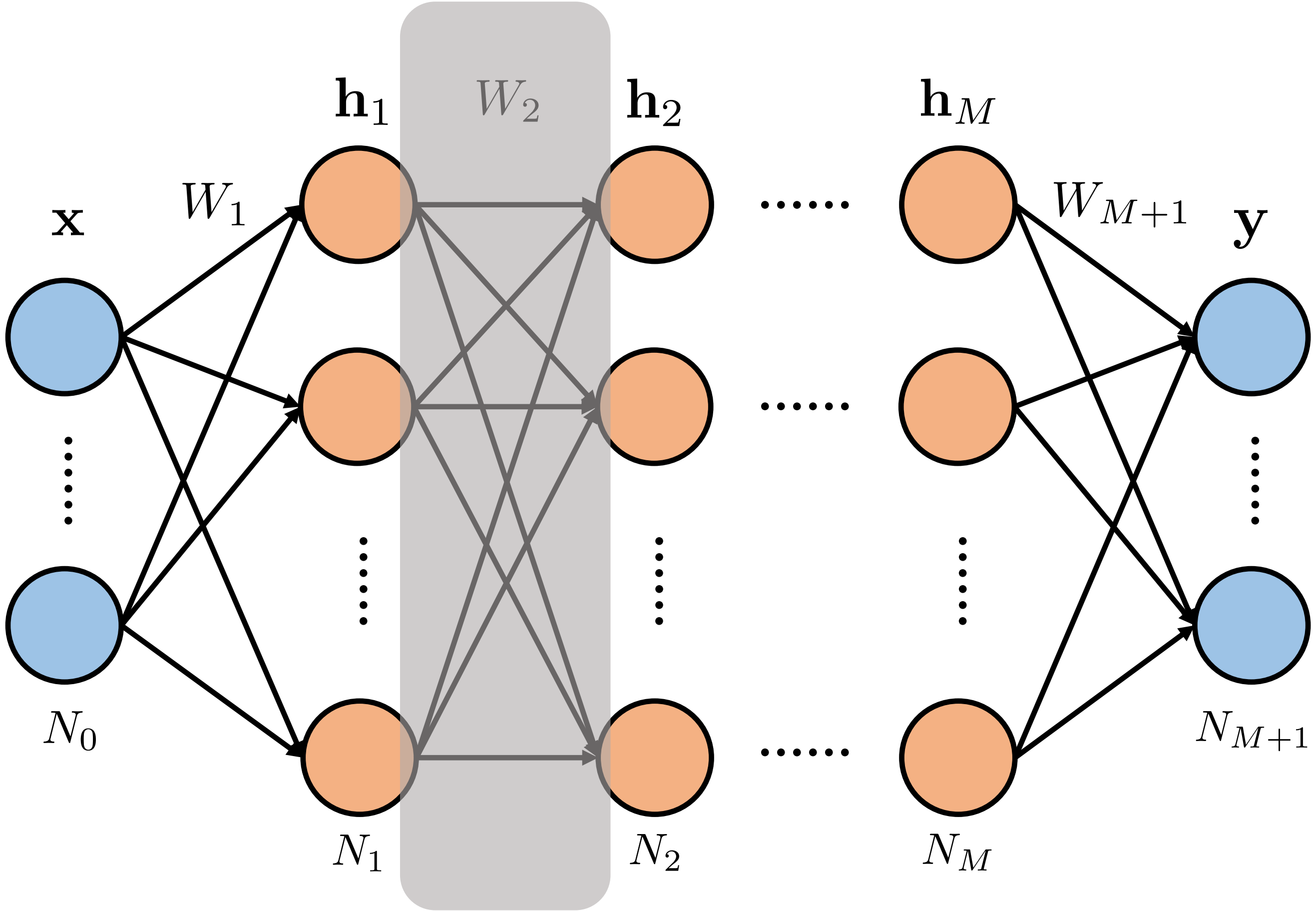
$$\frac{\partial L}{\partial \mathbf{h}_1[j]} = \sum_{i=1}^{N_2} \frac{\partial L}{\partial \mathbf{h}_2[i]} \frac{\partial \mathbf{h}_2[i]}{\partial \mathbf{h}_1[j]} \approx \sum_{i=1}^{N_2} \frac{\partial L}{\partial \mathbf{h}_2[i]} W_2[i,j]$$



Initialization: Backward Analysis

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

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



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

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

Let us again assume i.i.d. and zero mean with

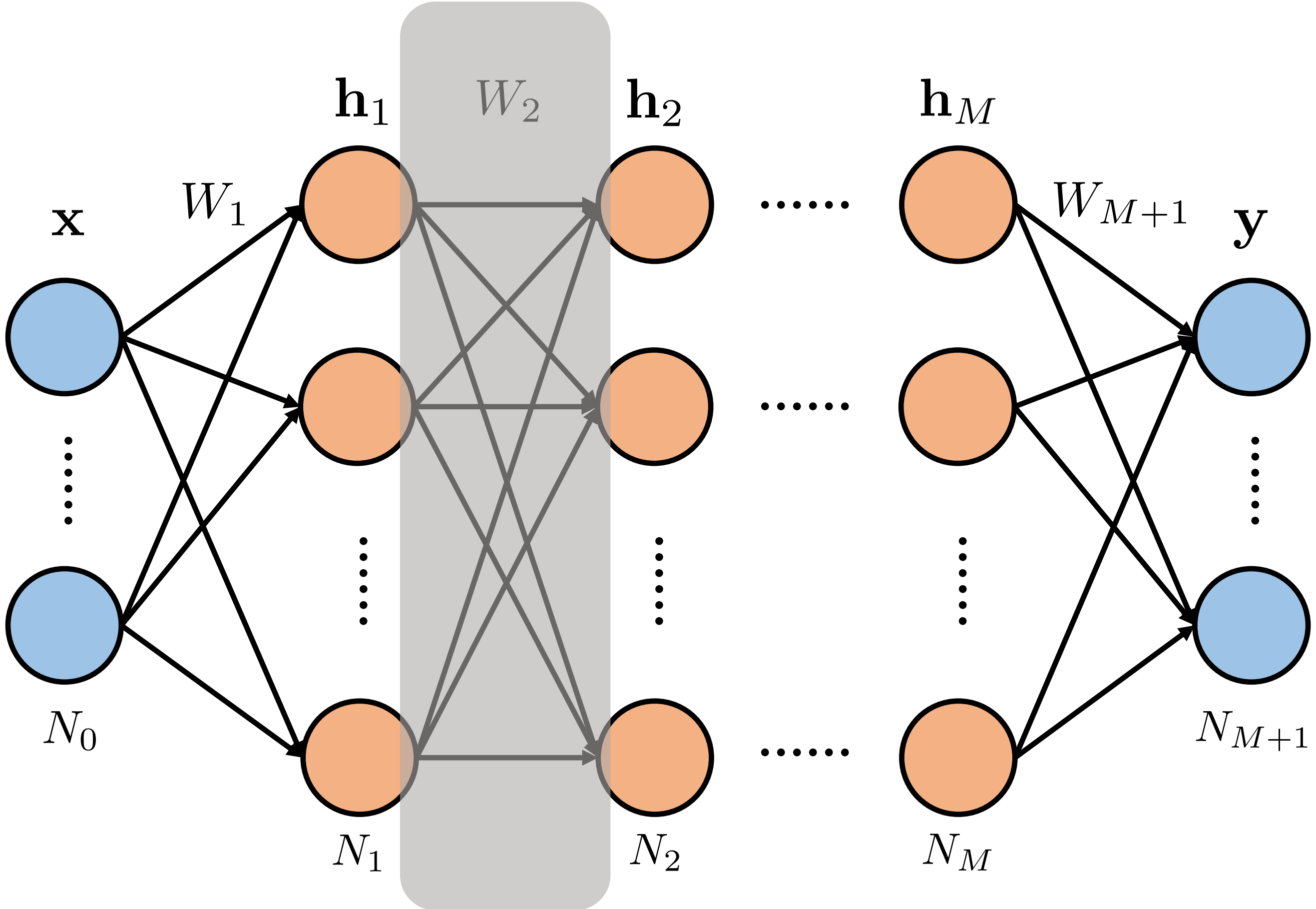
same variance for all $\dfrac{\partial L}{\partial \mathbf{h}_2[i]}$, $W_2[i,j]$, and $\mathbf{h}_2[i]$ at all layers, then

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

Initialization: Backward Analysis

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

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



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

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

Let us again assume i.i.d. and zero mean with

same variance for all $\frac{\partial L}{\partial \mathbf{h}_2[i]}$, $W_2[i,j]$, and $\mathbf{h}_2[i]$ at all layers, then

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

Setting $\mathbb{V}\left[W_k[i,j]\right] = \frac{1}{N_k}$ preserves the

variance of gradients w.r.t. activations!

Initialization: Backward Analysis

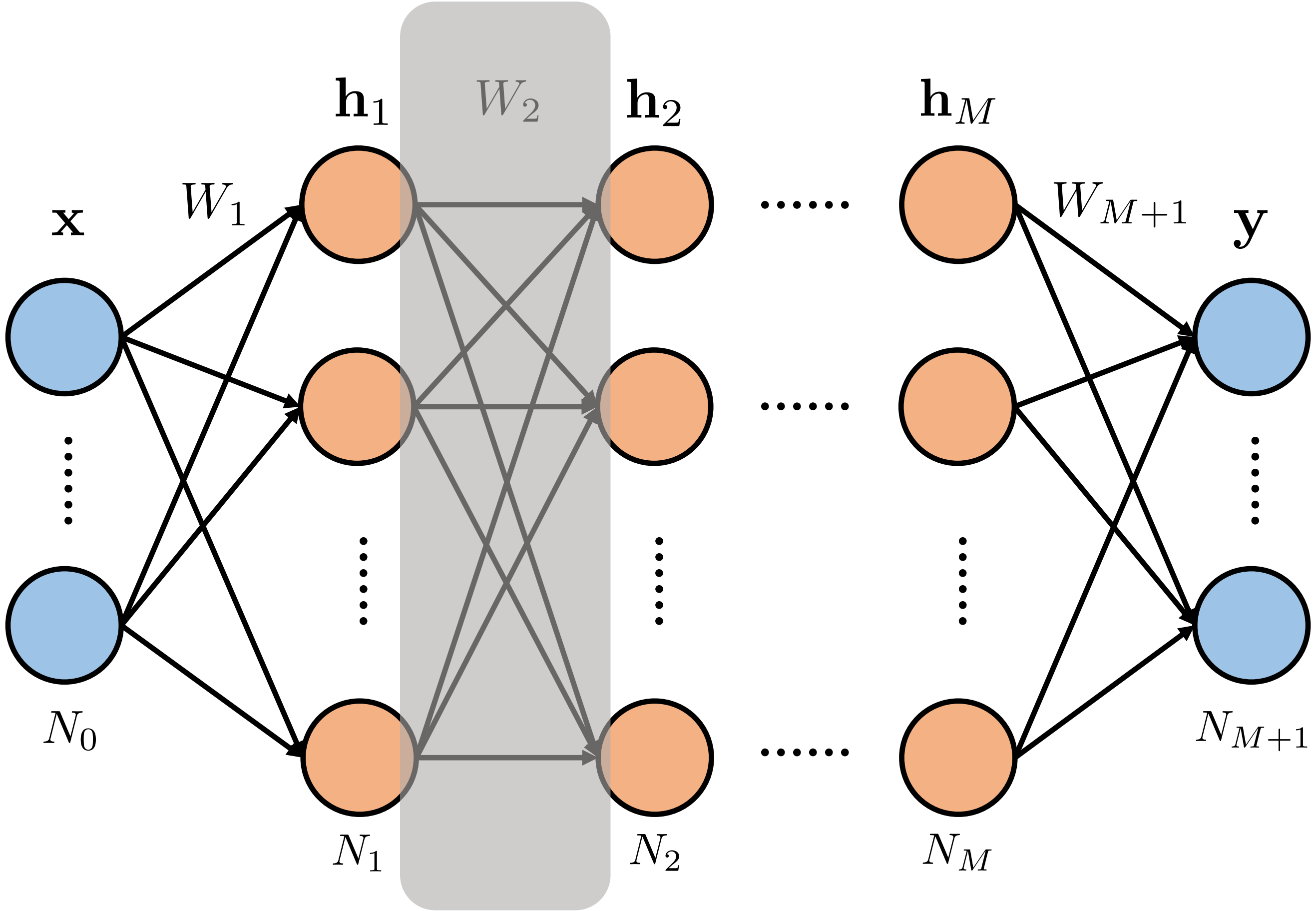
Assuming Tanh activation $\sigma(x) = \dfrac{e^x - e^{-x}}{e^x + e^{-x}}$ and we are in the linear regime

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

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

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

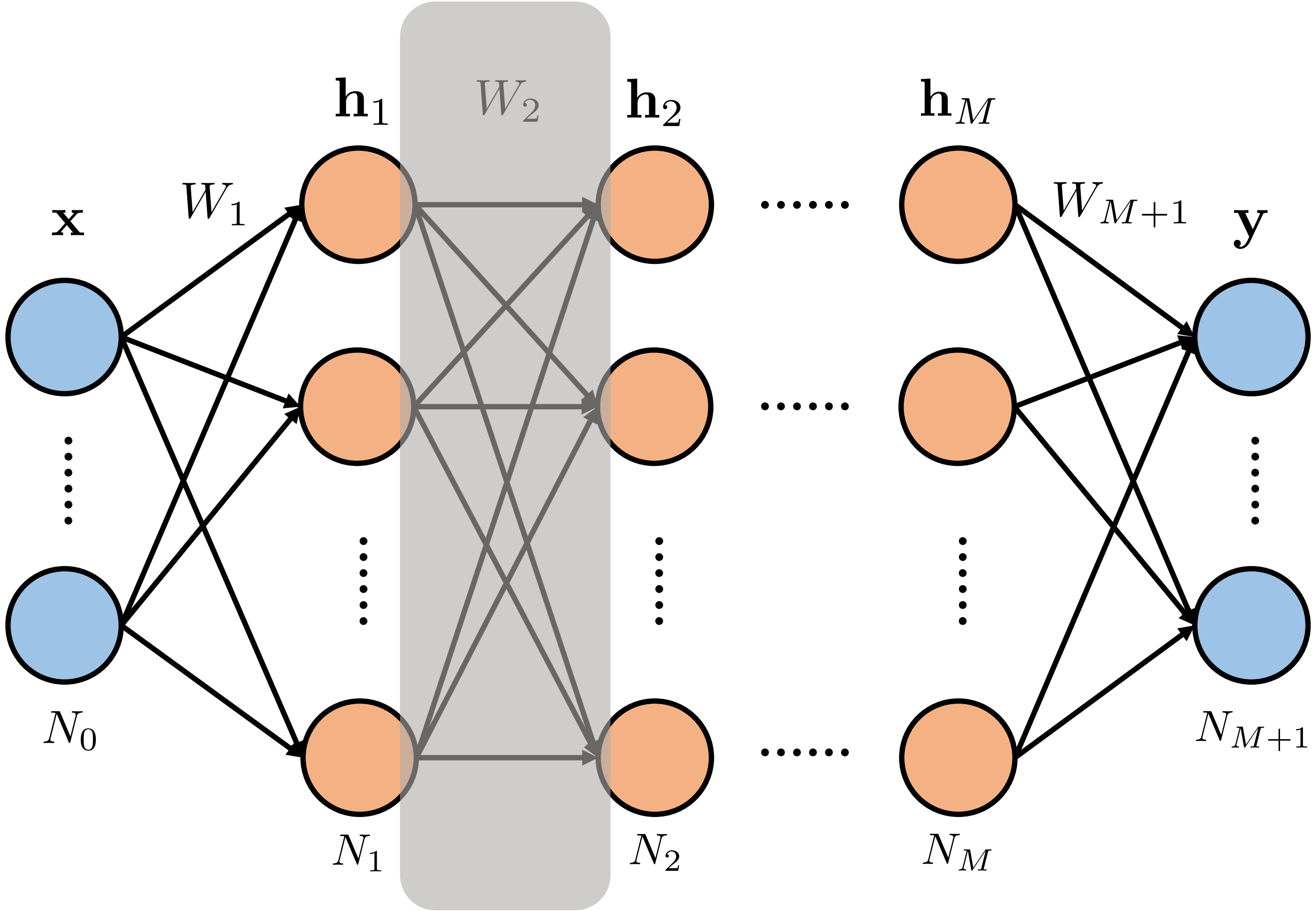
What about the gradients w.r.t. weights?



Initialization: Backward Analysis

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

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



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

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

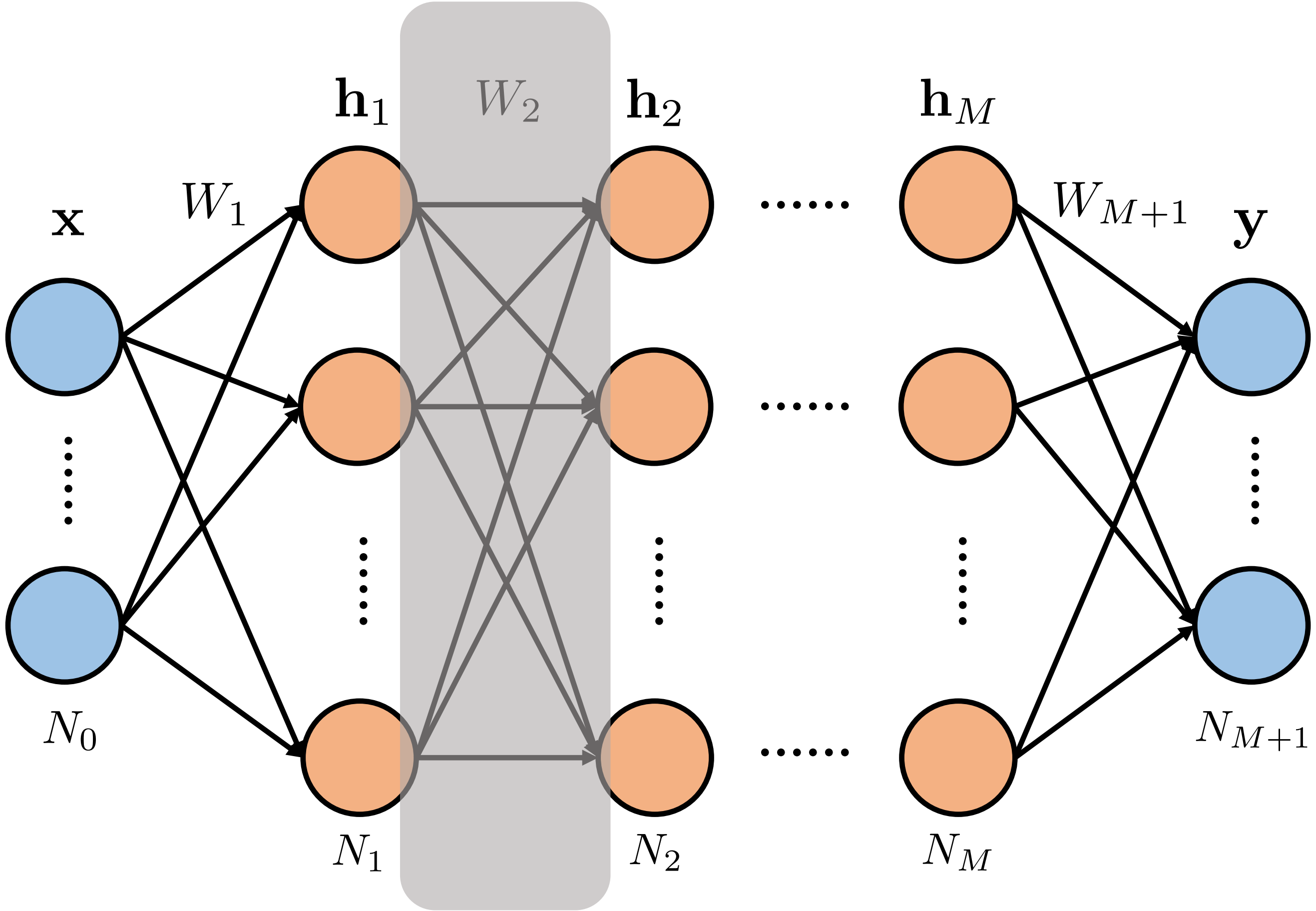
What about the gradients w.r.t. weights?

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

Initialization: Backward Analysis

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

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



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

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

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

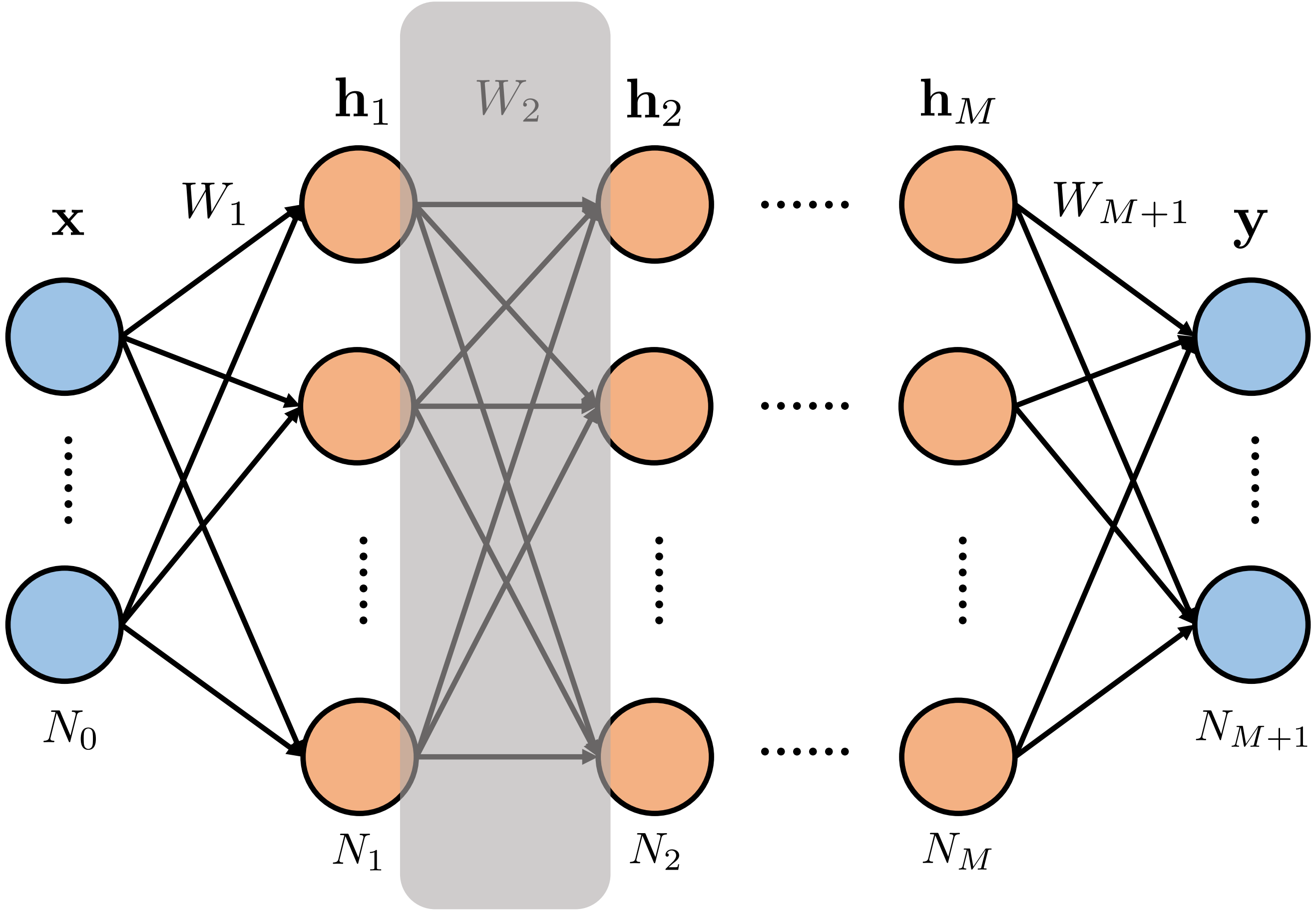
$$\frac{\partial L}{\partial W_2[i,j]} = \frac{\partial L}{\partial \mathbf{h}_2[i]} \frac{\partial \mathbf{h}_2[i]}{\partial W_2[i,j]} \approx \frac{\partial L}{\partial \mathbf{h}_2[i]} \mathbf{h}_1[j]$$

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

Initialization: Backward Analysis

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

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



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

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

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

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

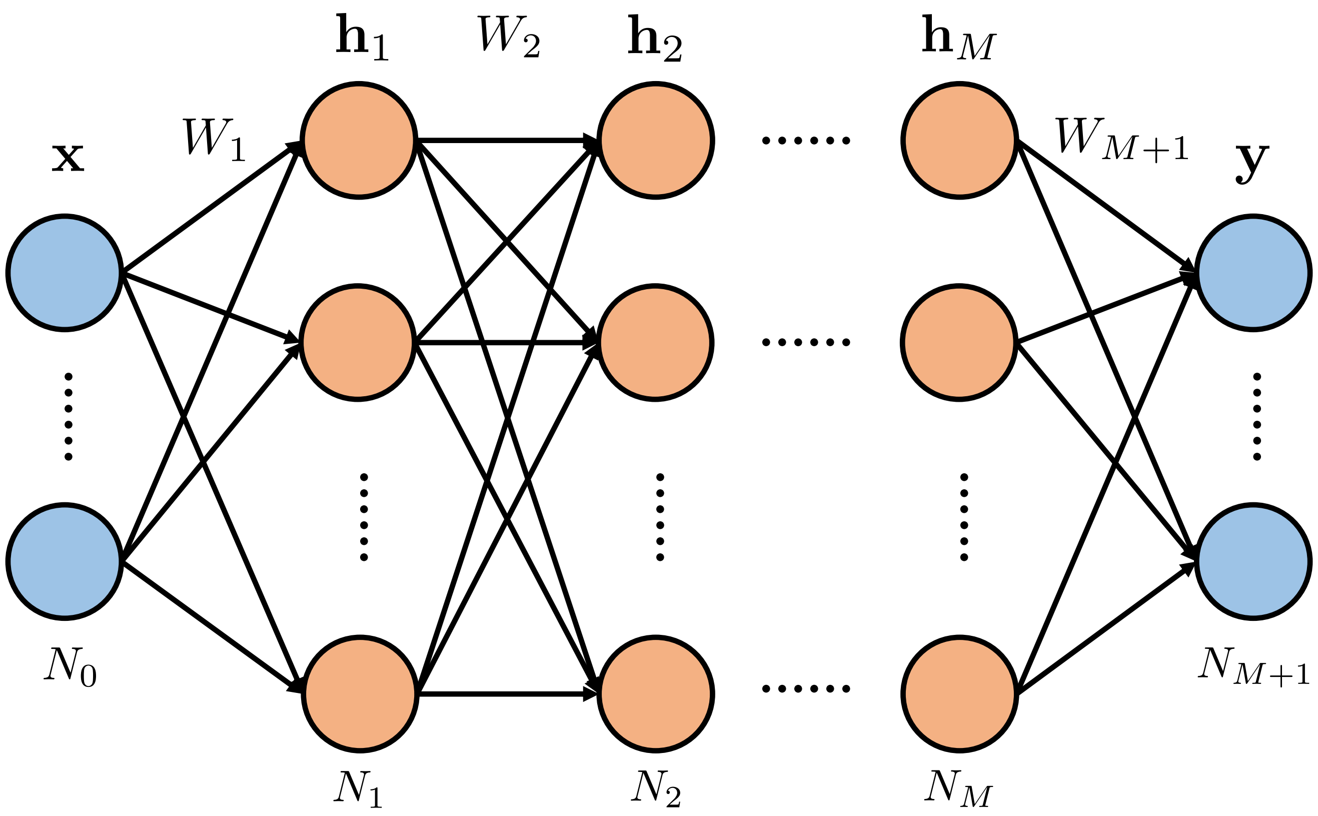
$$\begin{aligned}
\mathbb{V}\left[\frac{\partial L}{\partial W_2[i,j]}\right] &\approx \mathbb{V}\left[\frac{\partial L}{\partial \mathbf{h}_2[i]}\right]\mathbb{V}\left[\mathbf{h}_1[j]\right] \\
&= \left(\mathbb{V}\left[\frac{\partial L}{\partial \mathbf{y}[i]}\right]\prod_{k=3}^{M+1} N_k \mathbb{V}\left[W_k[i,j]\right]\right)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

Setting $\mathbb{V}\left[W_k[i,j]\right] = \dfrac{1}{N_k}$ makes the variance of gradients w.r.t. weights behave reasonably (e.g., no exploding or vanishing)!

Initialization

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

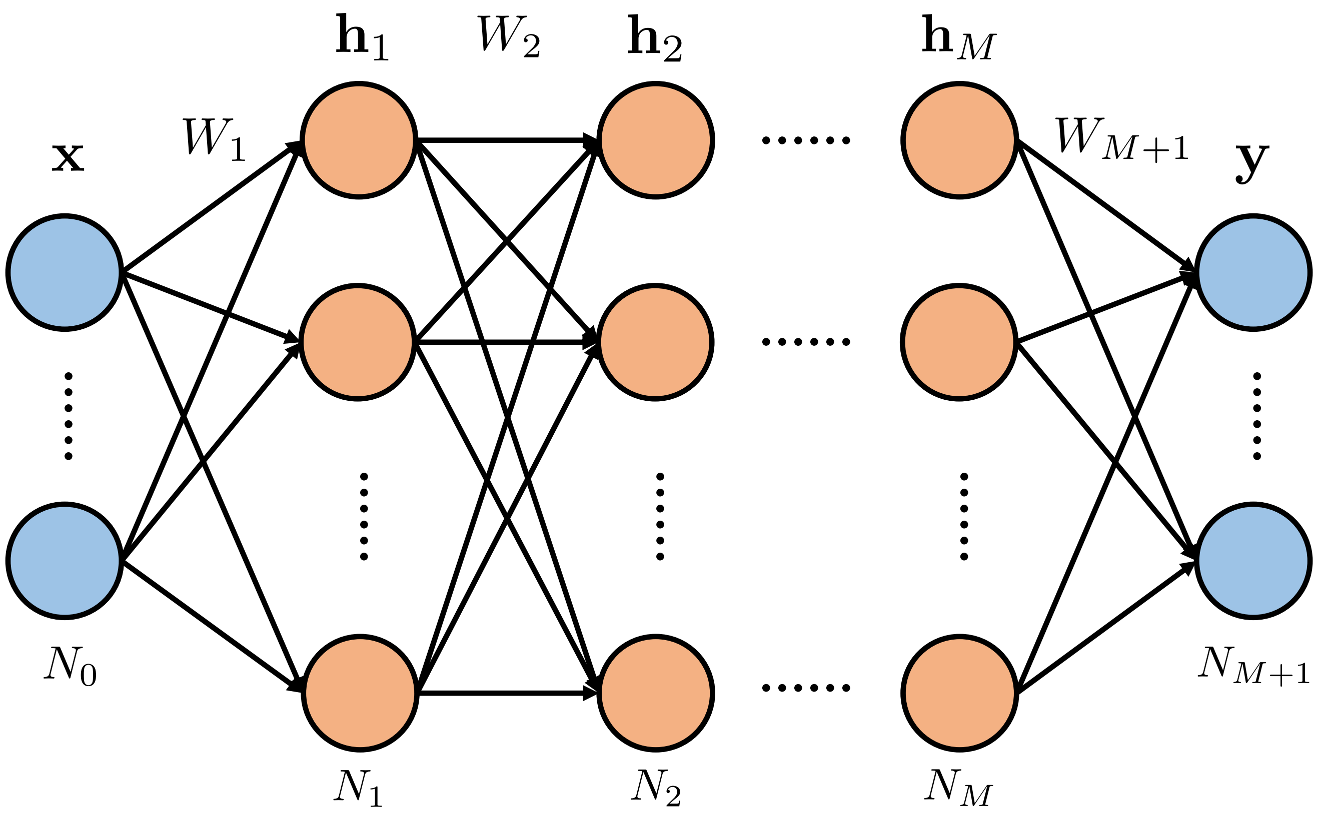
to preserve the variance of gradients w.r.t. activations, we set $\mathbb{V}\left[W_k[i,j]\right] = \dfrac{1}{N_k}$



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 $\mathbb{V}\left[W_k[i,j]\right] = \frac{1}{N_k}$



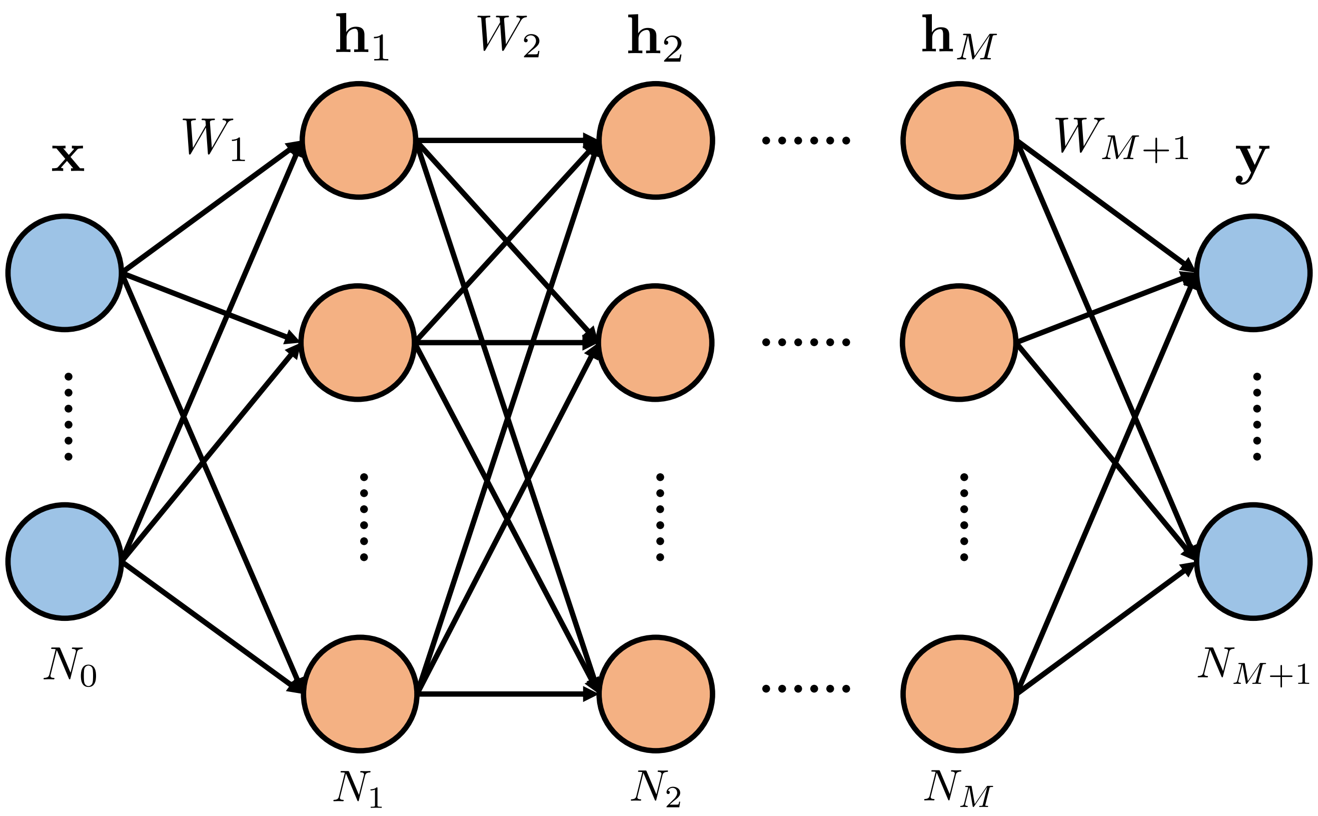
To compromise between two goals, we can take the mean of the denominators:

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

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:

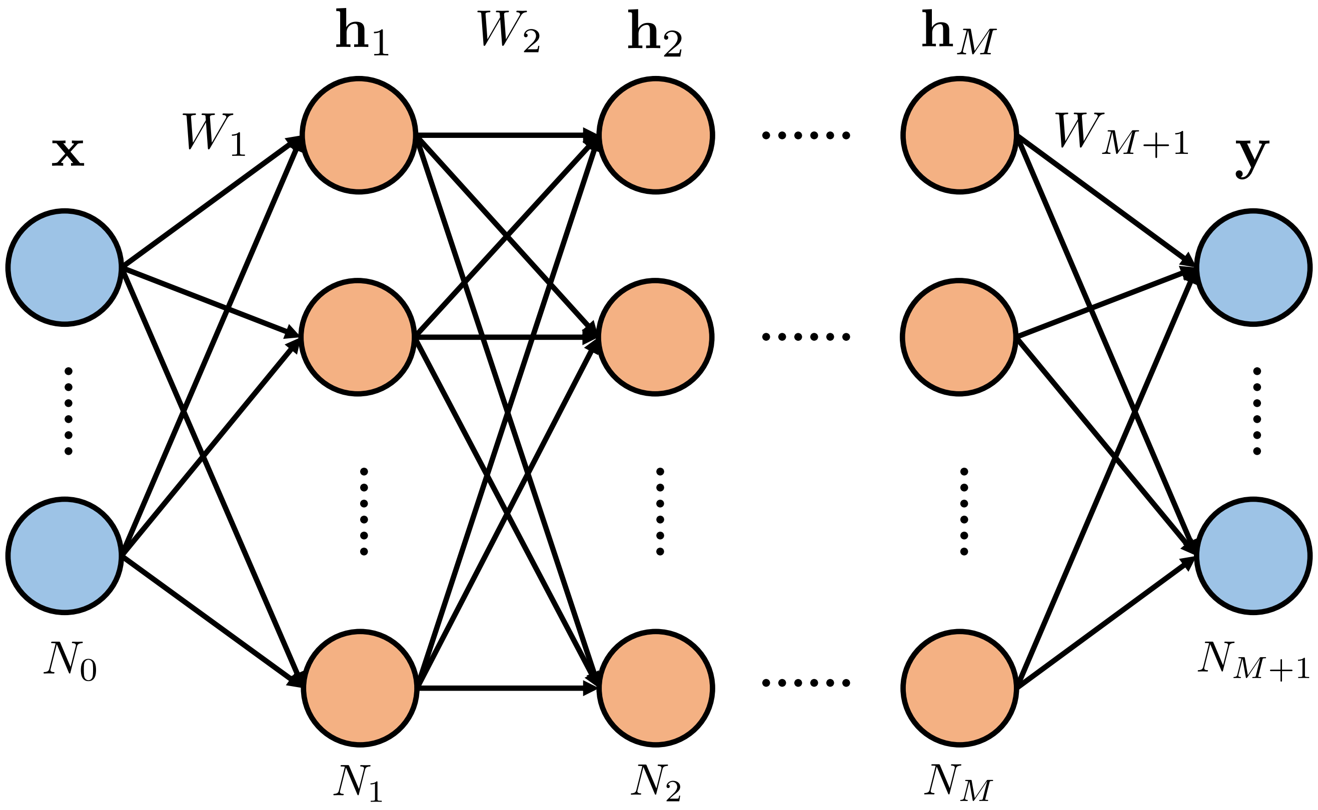
$$\mathbb{V}\left[W_k[i,j]\right] = \frac{1}{\frac{N_k+N_{k-1}}{2}} = \frac{2}{N_k + N_{k-1}}$$

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

Initialization

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]!

By considering the effect of ReLU, we can similarly derive "Kaiming Initialization" [4]!

References

[1] Robbins, H., & Monro, S. (1951). A stochastic approximation method. The annals of mathematical statistics, 400-407.

[2] Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. nature, 323(6088), 533-536.

[3] Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the thirteenth international conference on artificial intelligence and statistics (pp. 249-256).

[4] He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the IEEE international conference on computer vision (pp. 1026-1034).

Questions?