Part 2: Trainable Networks

•••

Mikhail Romanov

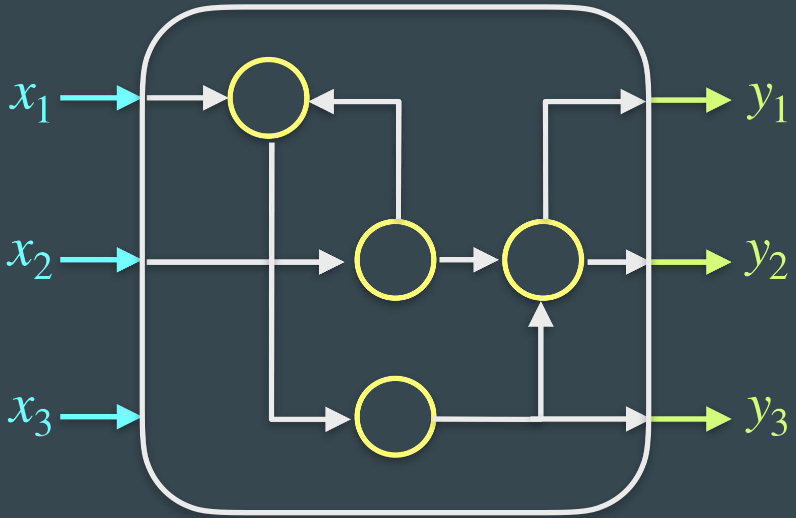
Training Philosophy

Training Philosophy

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \dots \\ x_N \end{bmatrix}$$

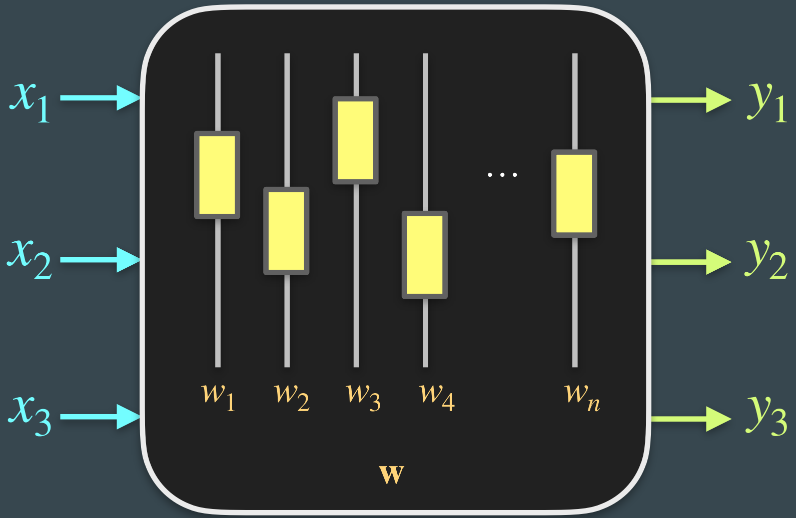
Training Philosophy

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \dots \\ x_N \end{bmatrix}$$



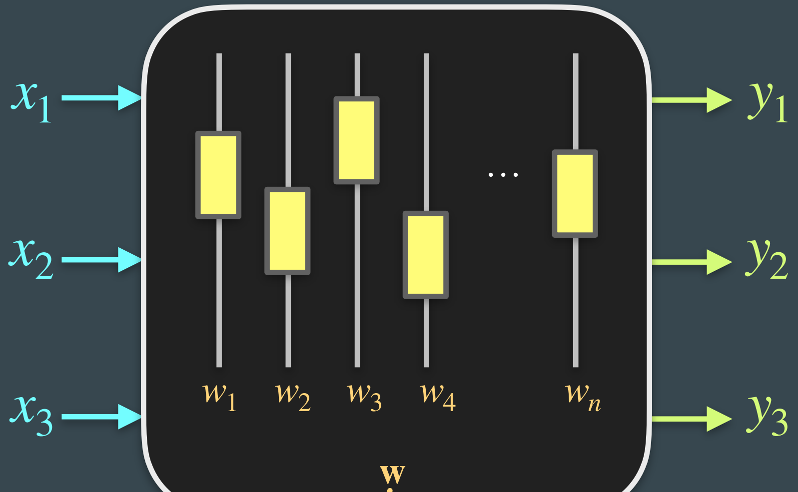
Training Philosophy

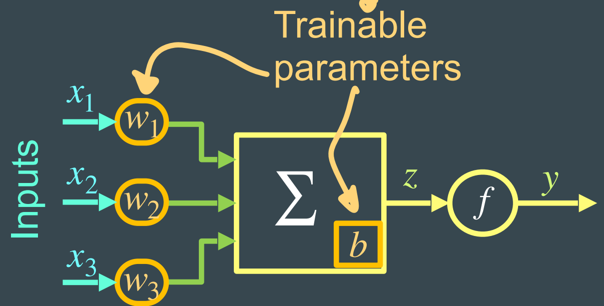
$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \dots \\ x_N \end{bmatrix}$$



Training Philosophy

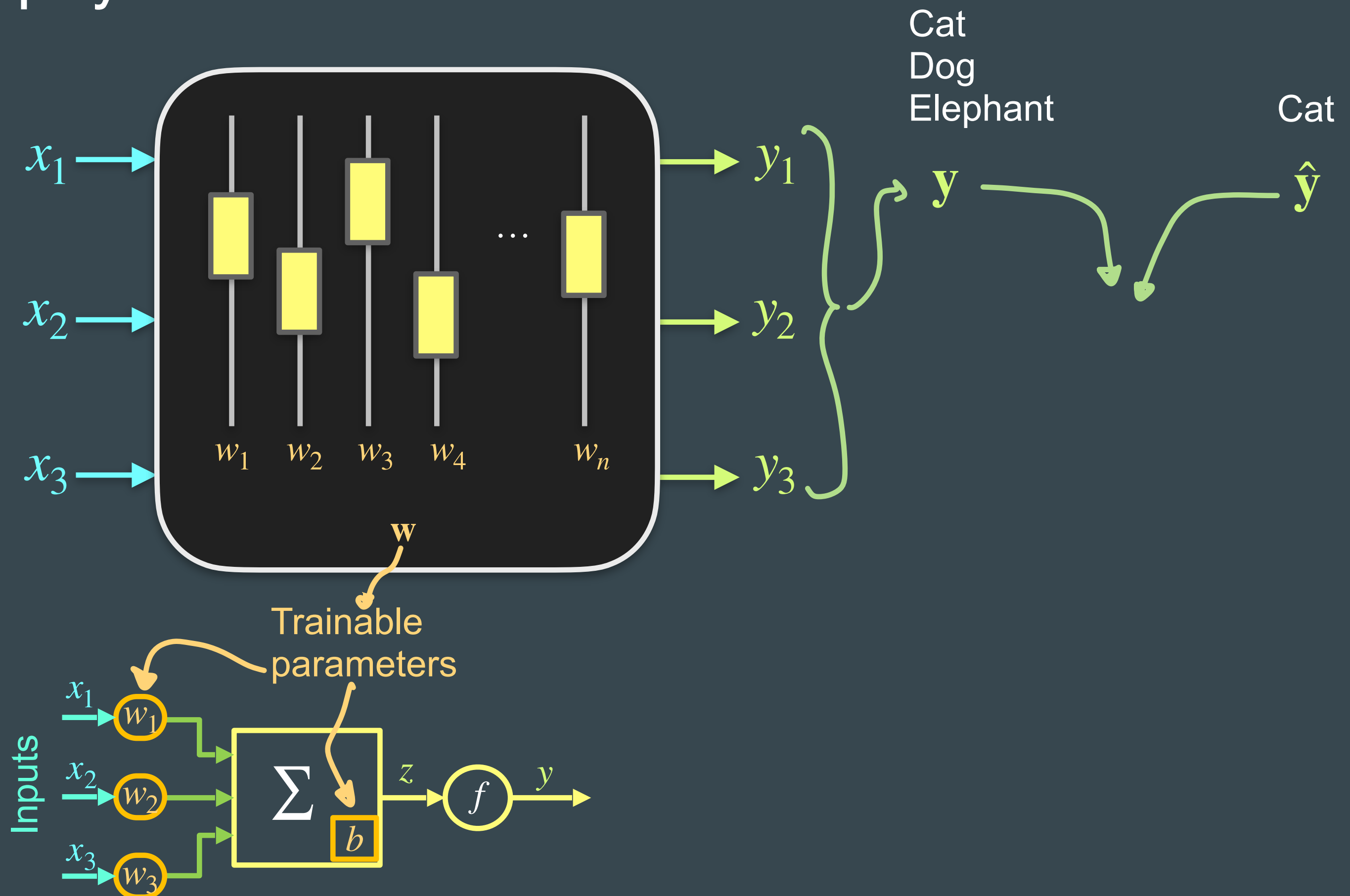
$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \dots \\ x_N \end{bmatrix}$$





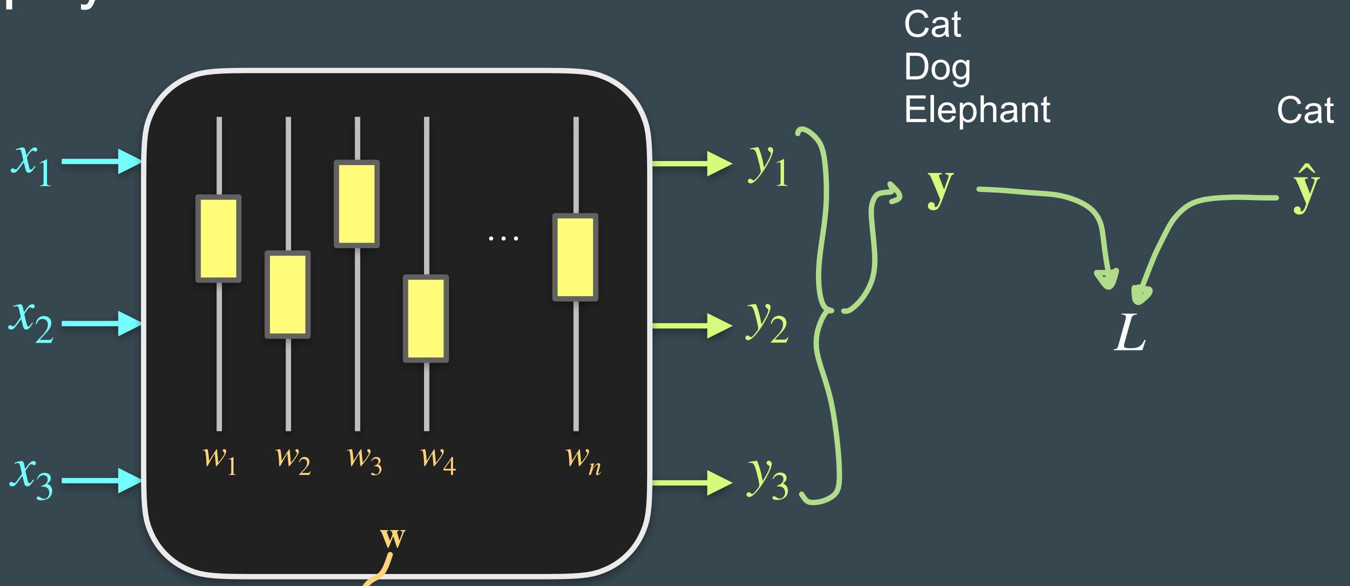
Training Philosophy

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \dots \\ x_N \end{bmatrix}$$

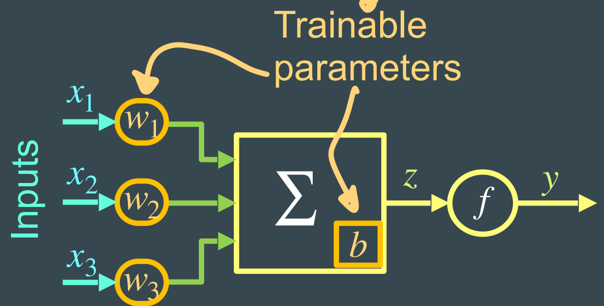


Training Philosophy

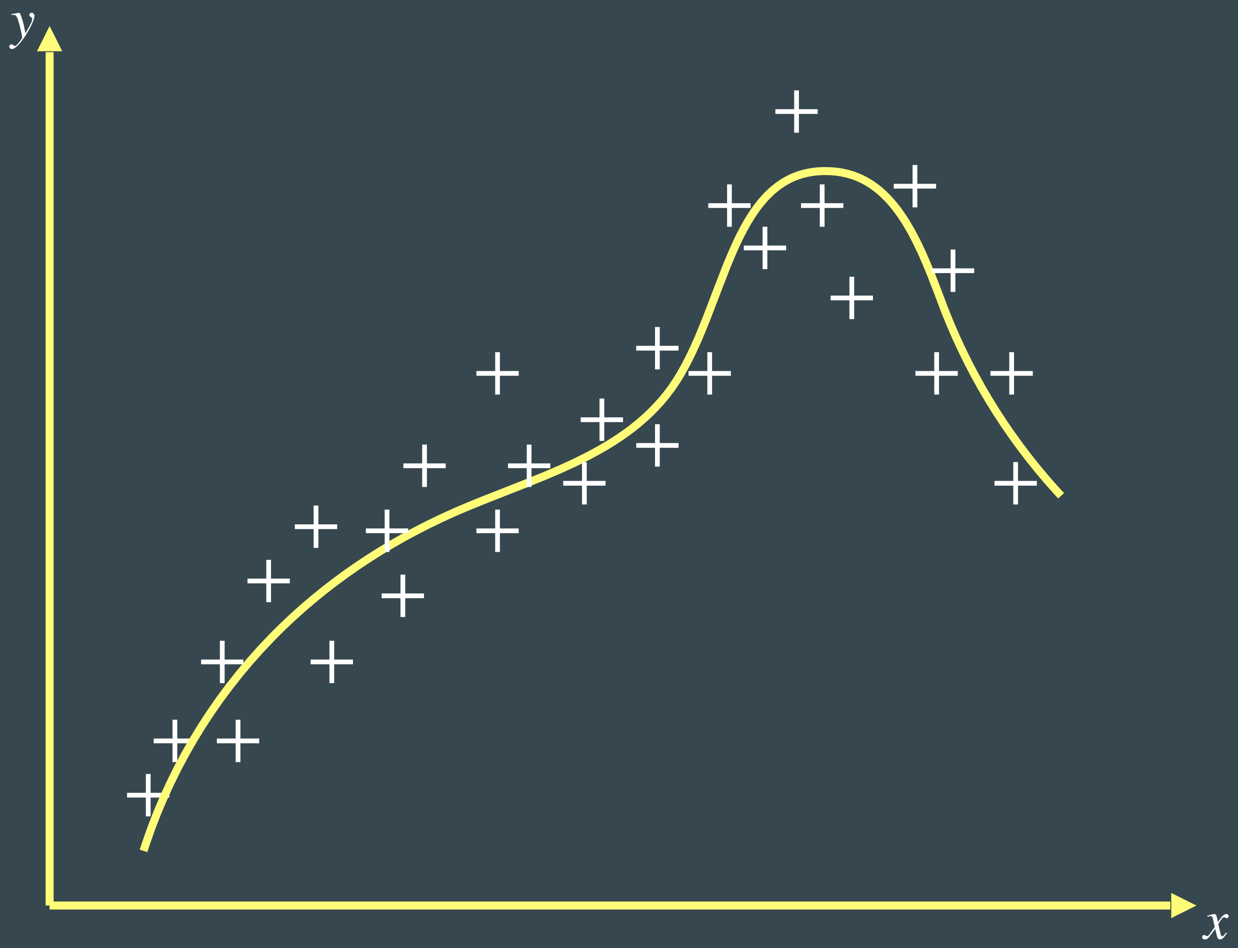
$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \dots \\ x_N \end{bmatrix}$$



$$\mathbf{w} = \underset{\mathbf{w}}{\text{argmin}}\, L(Net_{\mathbf{w}}(x), t)$$



Dependency Reconstruction



$$y_i = f(x_i) + \epsilon_i$$

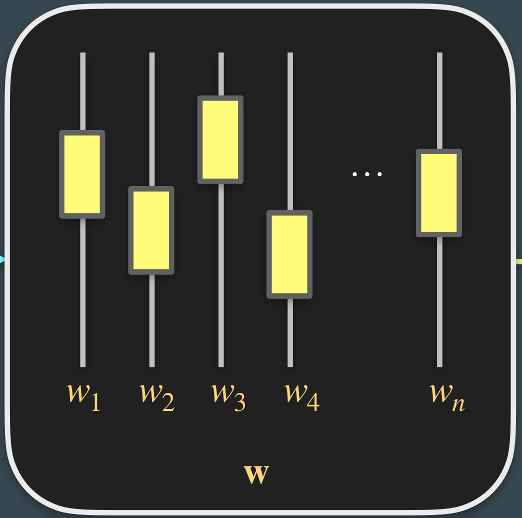
ϵ_i	Noise
$f(x_i)$	Hidden Dependency
x_i	Circumstances of Measurement
y_i	Result of measurement

Tuning the network

Features	Labels
$\mathbf{x}_1$	$\mathbf{y}_1$
$\mathbf{x}_2$	$\mathbf{y}_2$
$\mathbf{x}_3$	$\mathbf{y}_3$
$\mathbf{x}_4$	$\mathbf{y}_4$
$\mathbf{x}_5$	$\mathbf{y}_5$
$\mathbf{x}_6$	$\mathbf{y}_6$
$\dots$	$\dots$
$\mathbf{x}_S$	$\mathbf{y}_S$

$$\mathbf{x} = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \dots \\ x_N \end{bmatrix}$$

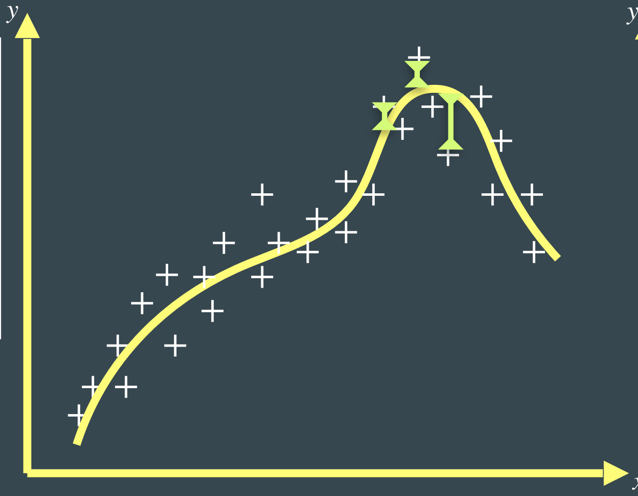
$\mathbf{x}$

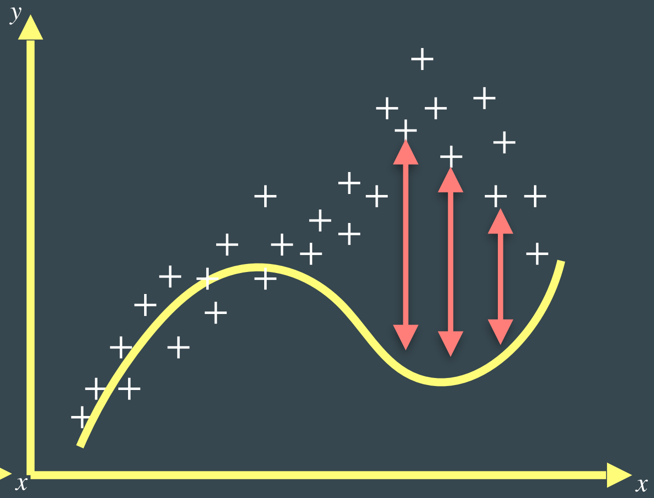


$\hat{\mathbf{y}}$

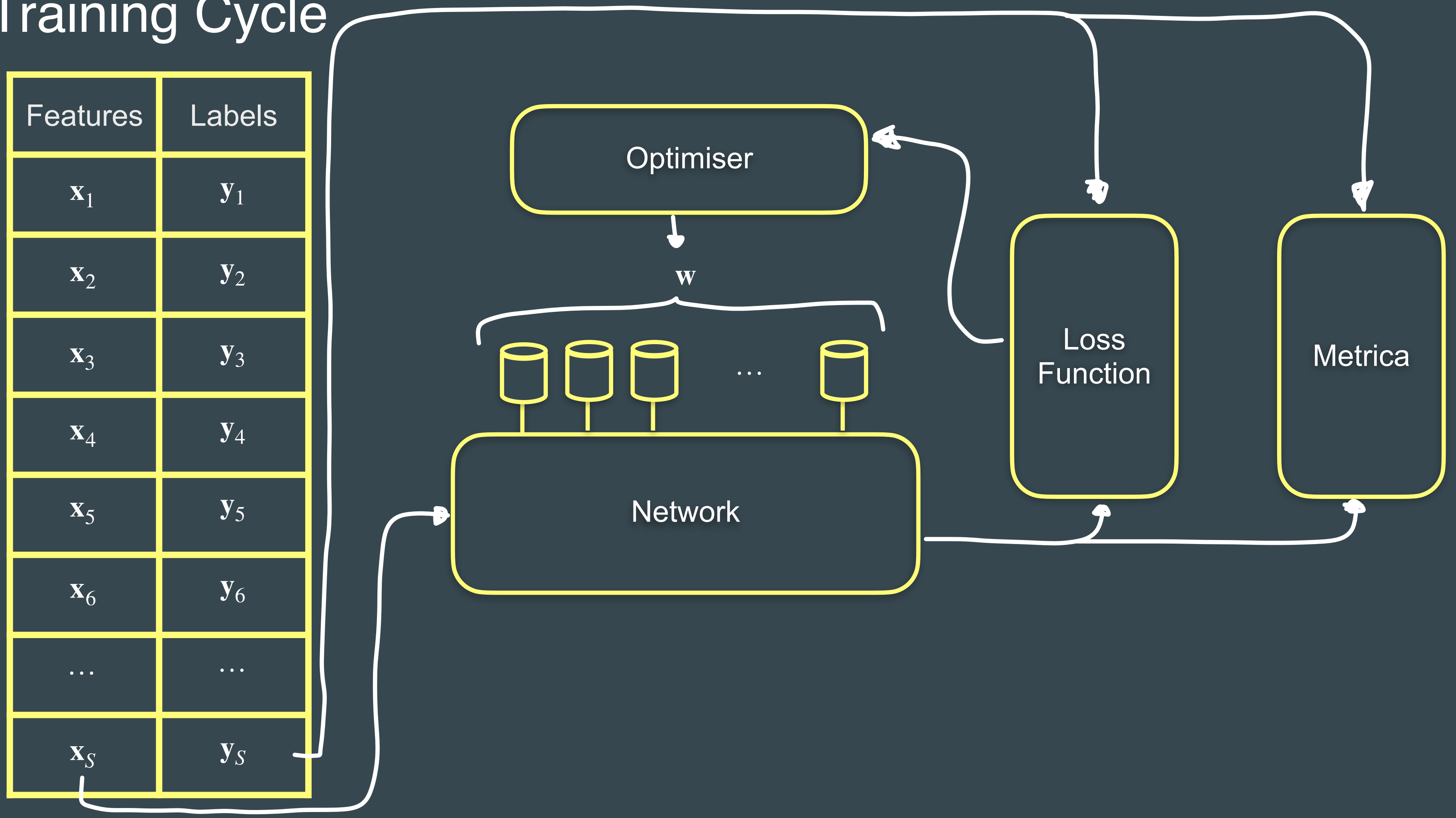
$$L_{total} = \sum_{s=1}^{S} L\left(Net_{\mathbf{w}}(x_s), y_s\right)$$

$$\mathbf{y} = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \dots \\ y_N \end{bmatrix}$$





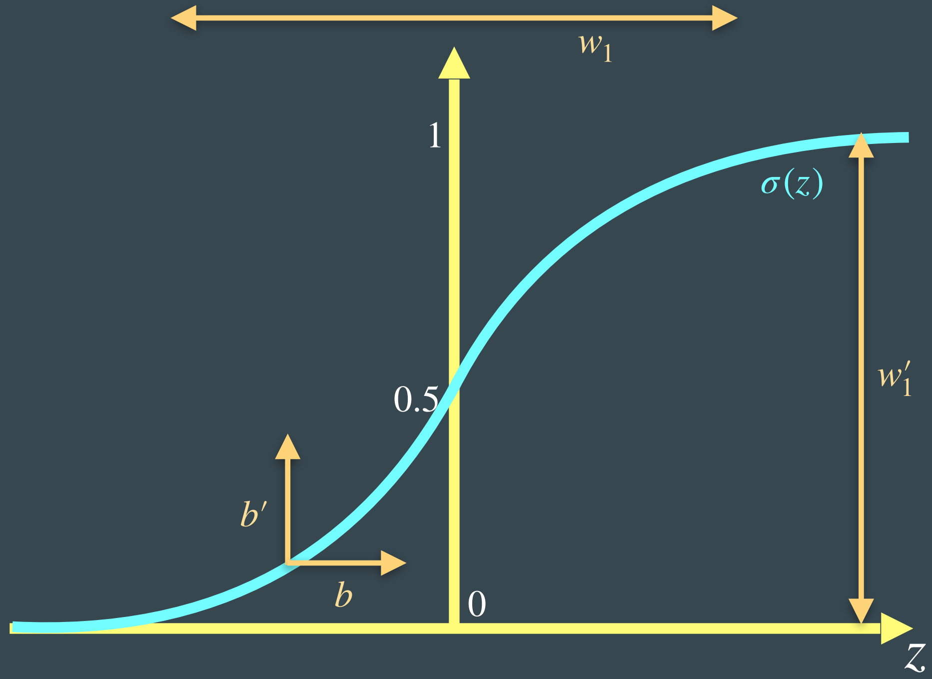
Training Cycle



Shaping the sigmoid

Sigmoid

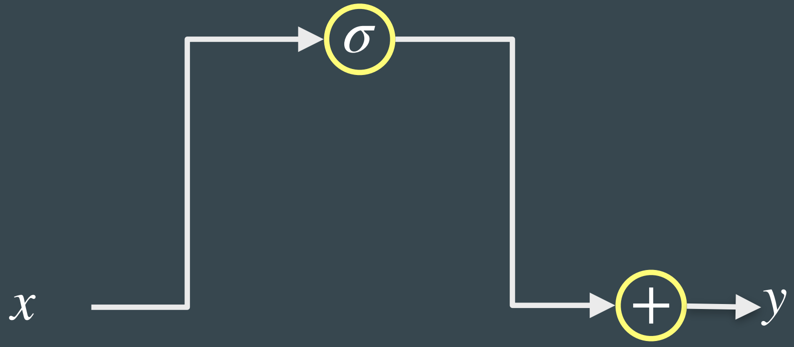
$$\sigma(z) = \frac{1}{1 + \exp(-z)} \qquad y = w'\sigma(wx+b)+b'$$

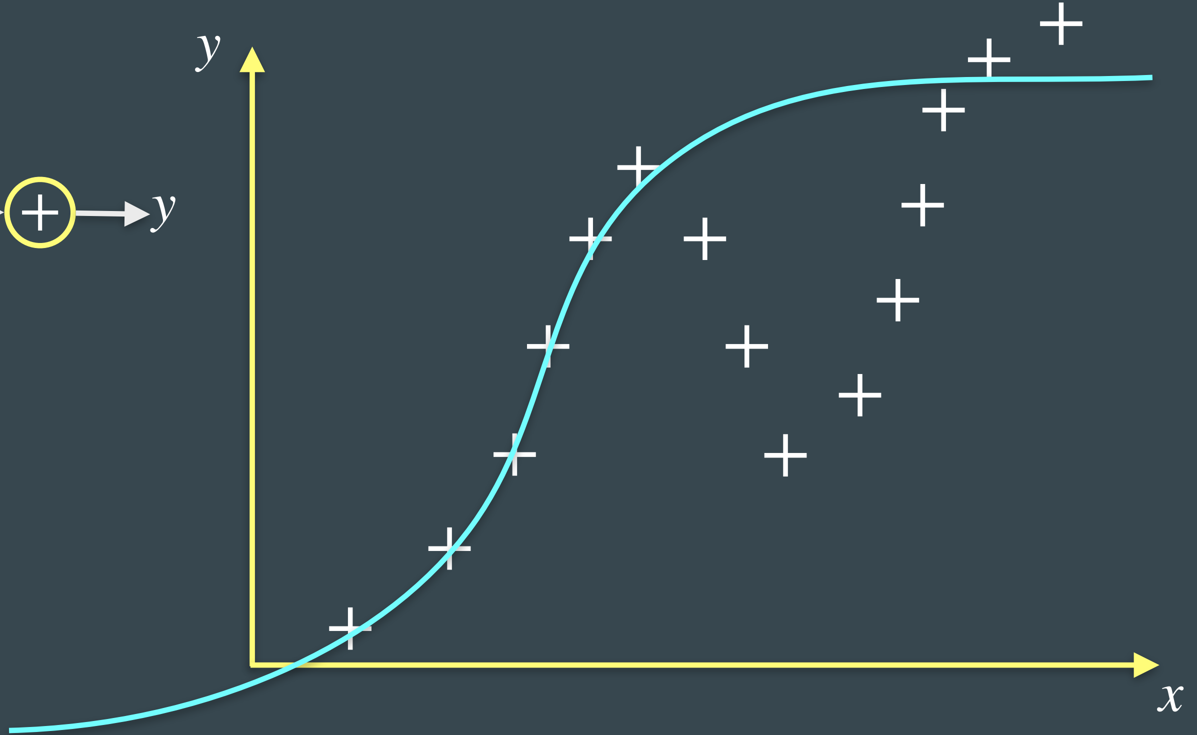


Two Layer Neural Net

$$y = w_1'\sigma(w_1 x+b_1)+b_1'$$

$$\hat{y} = b^{out}+\sum_{i=1}^{N} w_i^{out}\sigma(w^i x+b^i)$$



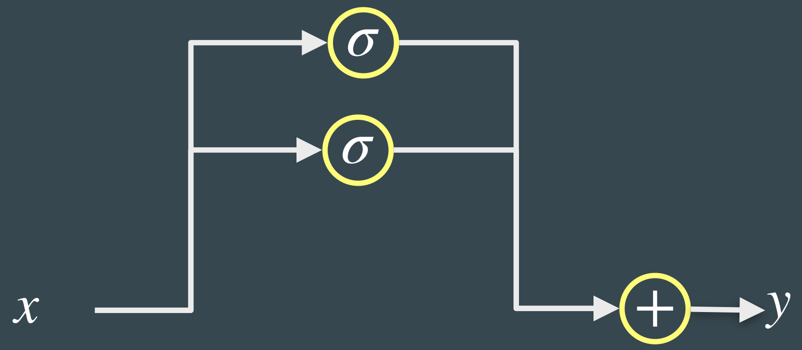


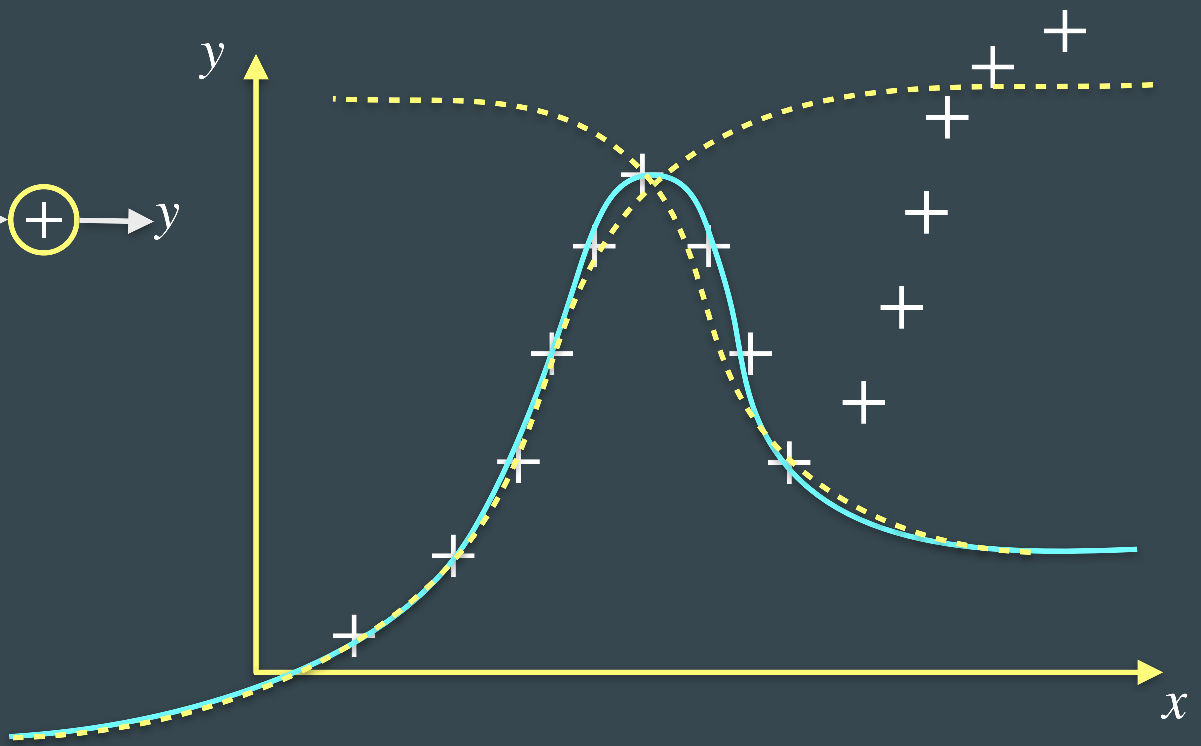
Two Layer Neural Net

$$y = w_1'\sigma(w_1 x + b_1) + b_1'$$

$$+ w_2'\sigma(w_2 x + b_2) + b_2'$$

$$\hat{y} = b^{out} + \sum_{i=1}^{N} w_i^{out}\sigma(w^i x + b^i)$$





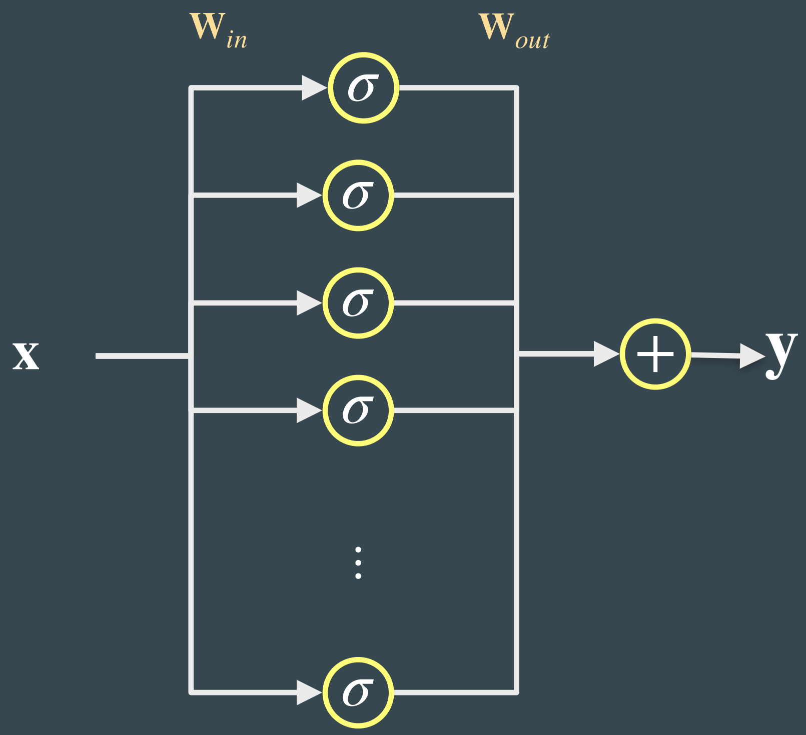
Two Layer Neural Net

$$y = w_1'\sigma(w_1 x + b_1) + b_1'$$

$$+ w_2'\sigma(w_2 x + b_2) + b_2'$$

$$+ w_3'\sigma(w_3 x + b_3) + b_3'$$

$$\hat{y} = b^{out} + \sum_{i=1}^{N} w_i^{out}\sigma(w^i x + b^i)$$



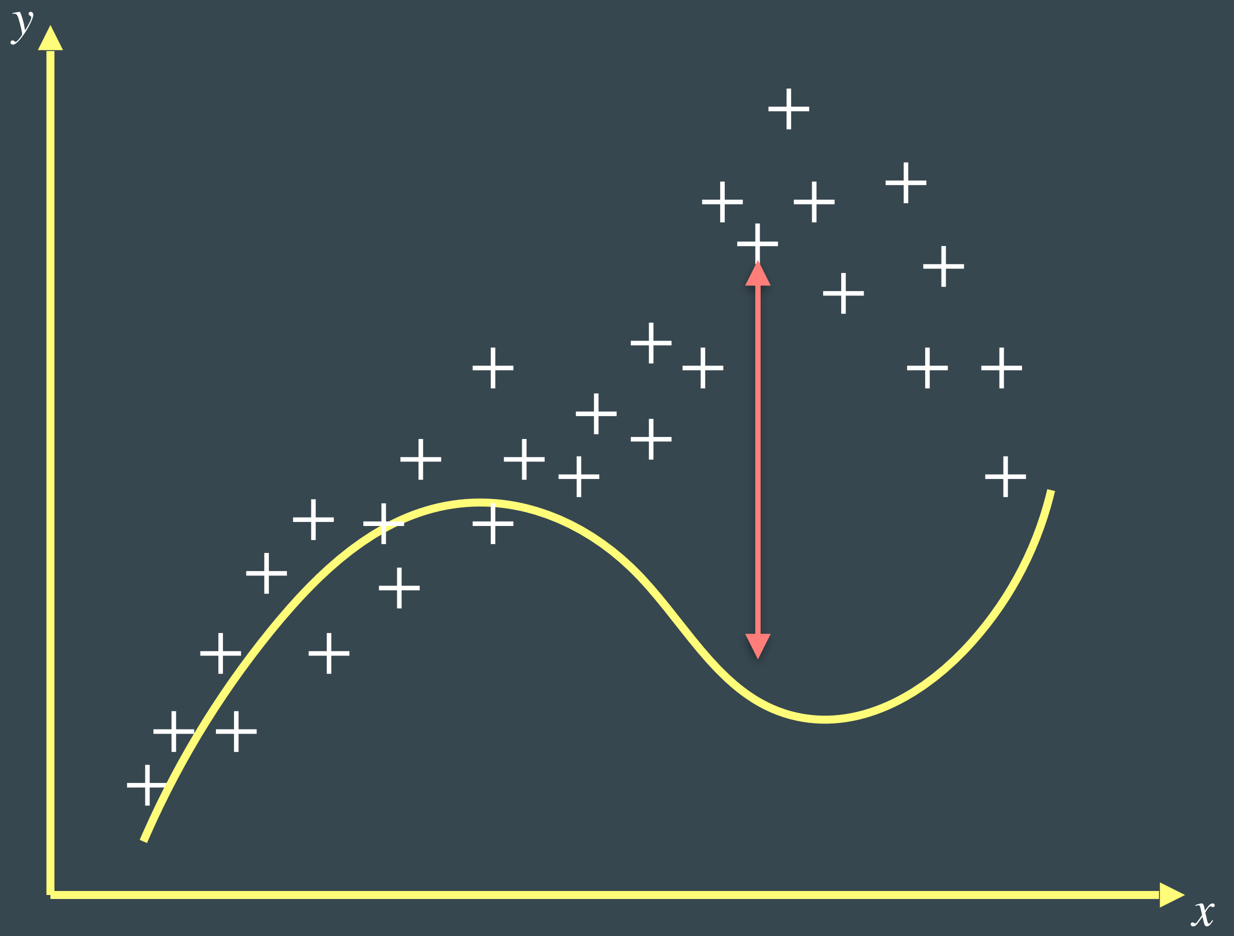
$$\hat{\mathbf{y}} = \mathbf{b}_{out} + \sum_{i=1}^{N} \mathbf{W}_{out}\sigma(\mathbf{W_{in}}\mathbf{X} + \mathbf{b}_{in})$$

Linear Combination of Sigmoids is Full System!

Loss function

The simplest loss function is
Mean Squared Error
(MSE)

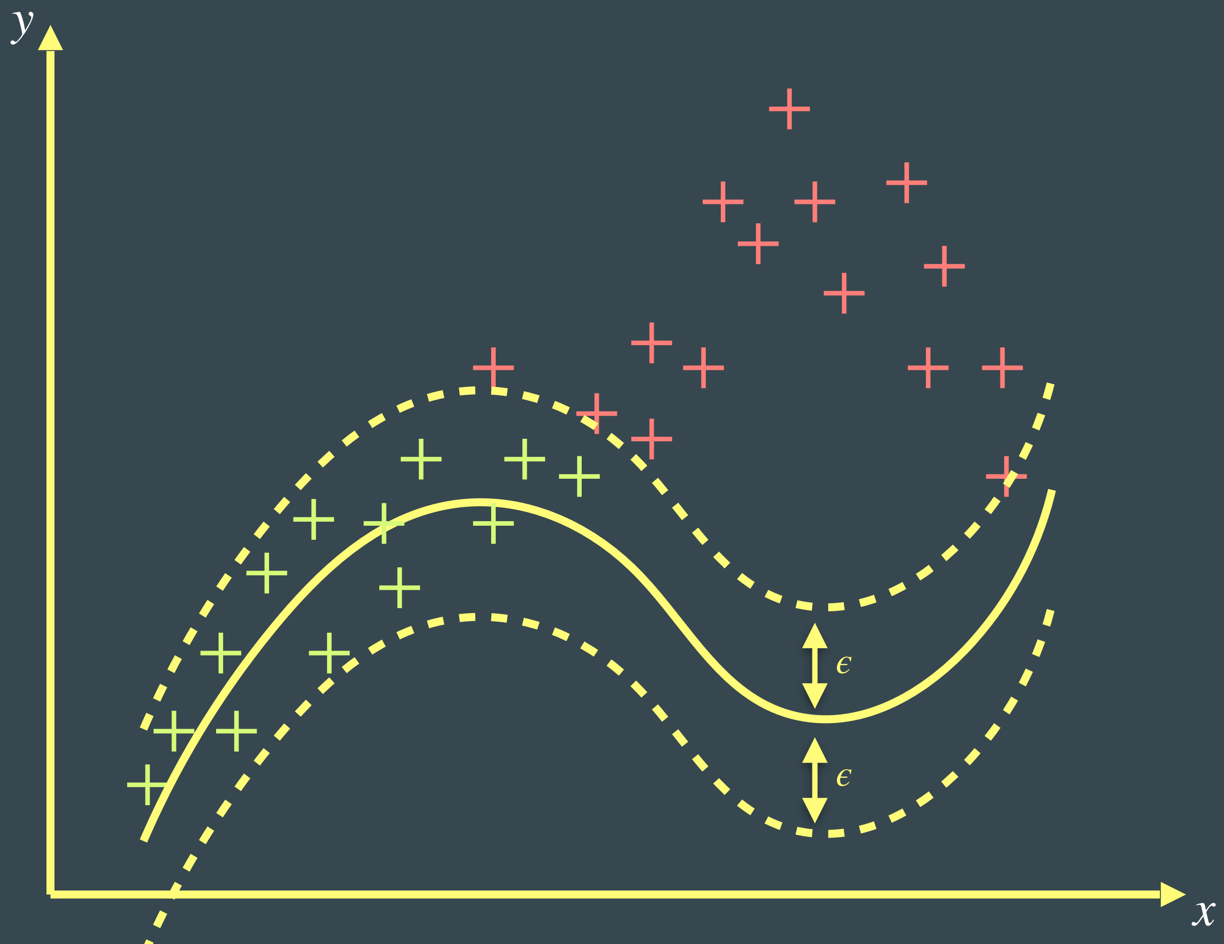
$$MSE(y, \hat{y}) = \frac{1}{N} \sum_{s=1}^{S} (y_s - \hat{y}_s)^2$$



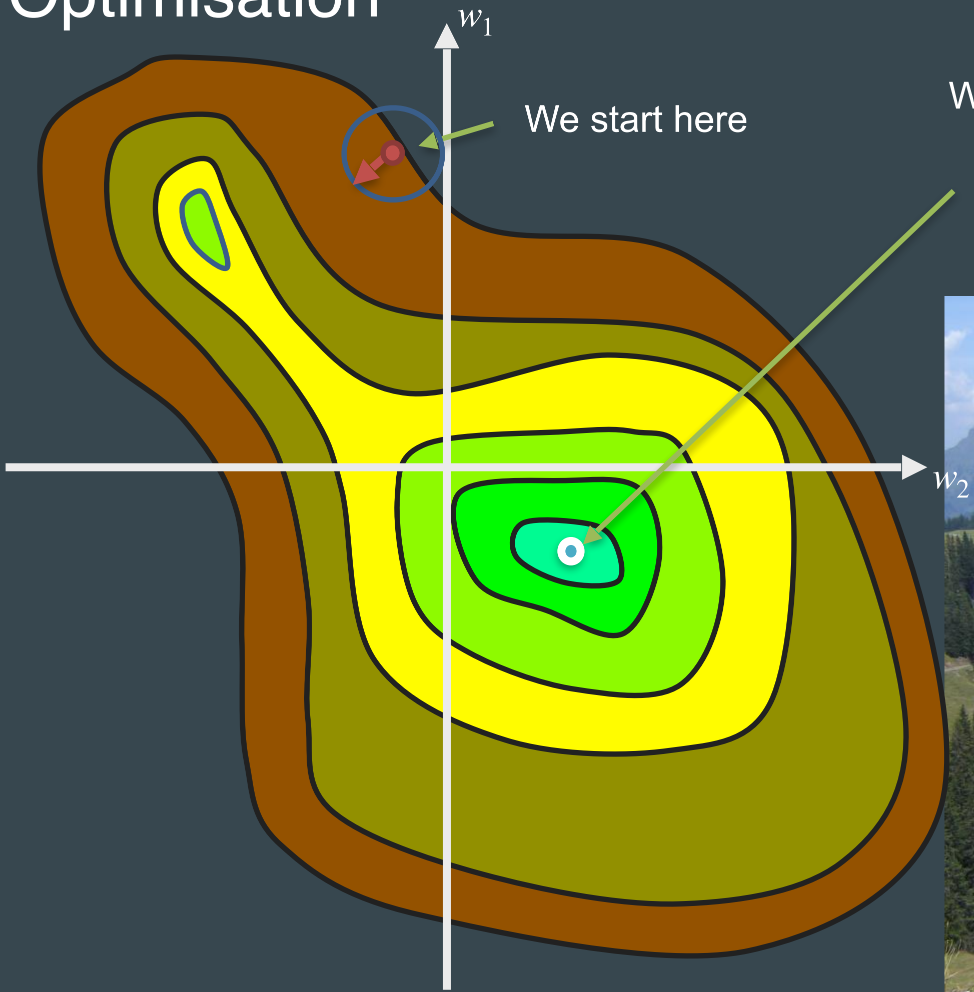
Metrica

Example:
Epsilon-precision

$$Acc_{\epsilon} = \frac{1}{N} \sum_{i=1}^{N} \left[\;\; y_i - \hat{y}_i \;\;\; < \epsilon \right]$$



Optimisation

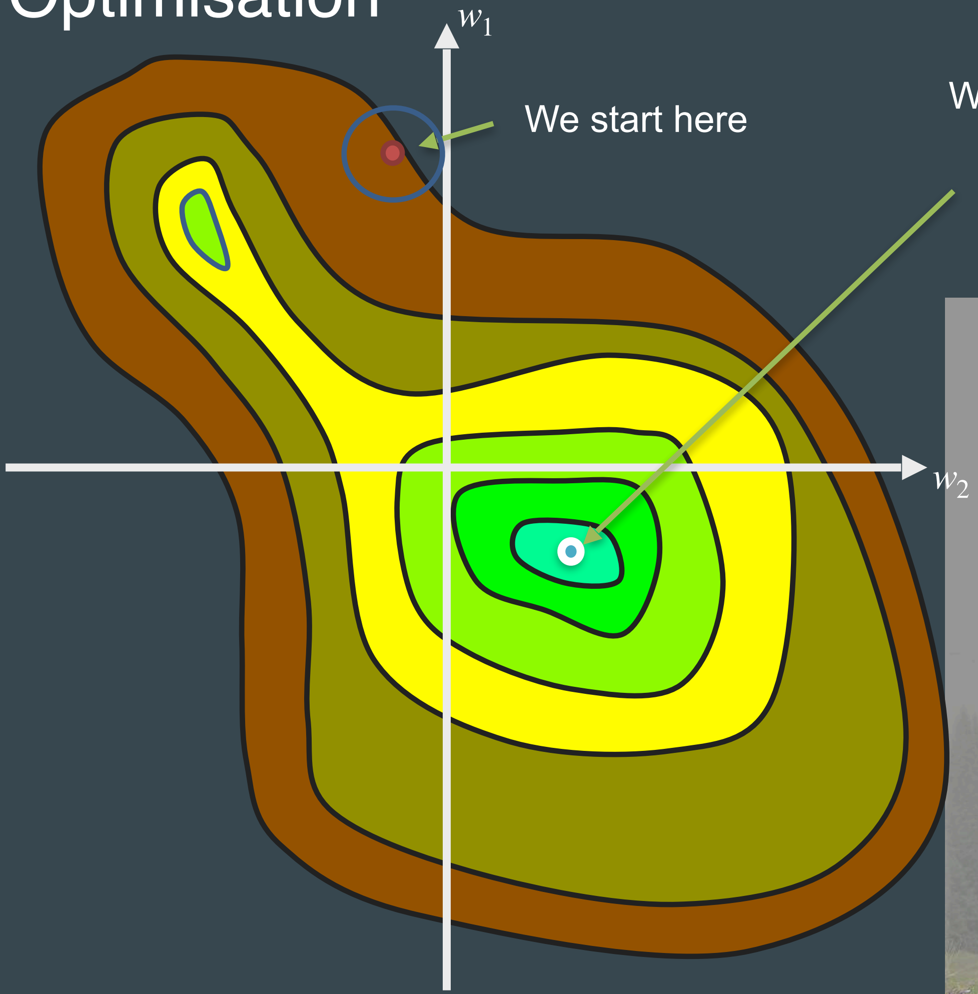


We start here

We want to arrive here

We are in “Switzerland”

Optimisation

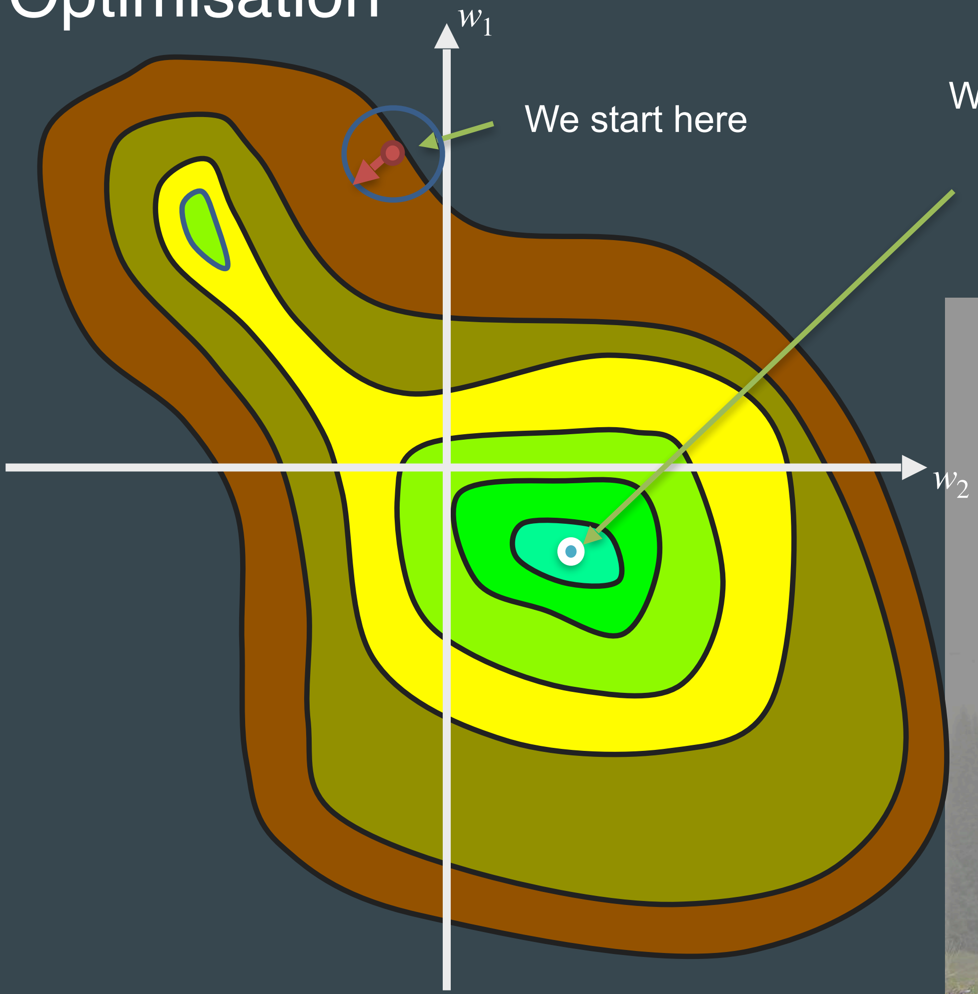


We start here

We want to arrive here

We are in “Switzerland”
In the mist
What would you do?

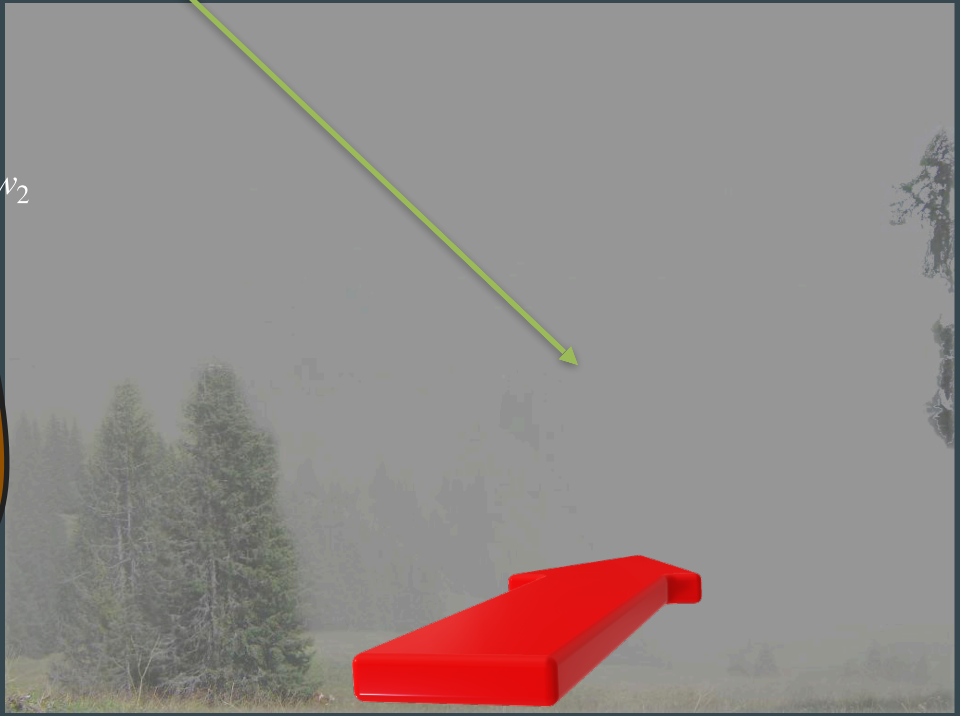
Optimisation



We start here

We want to arrive here

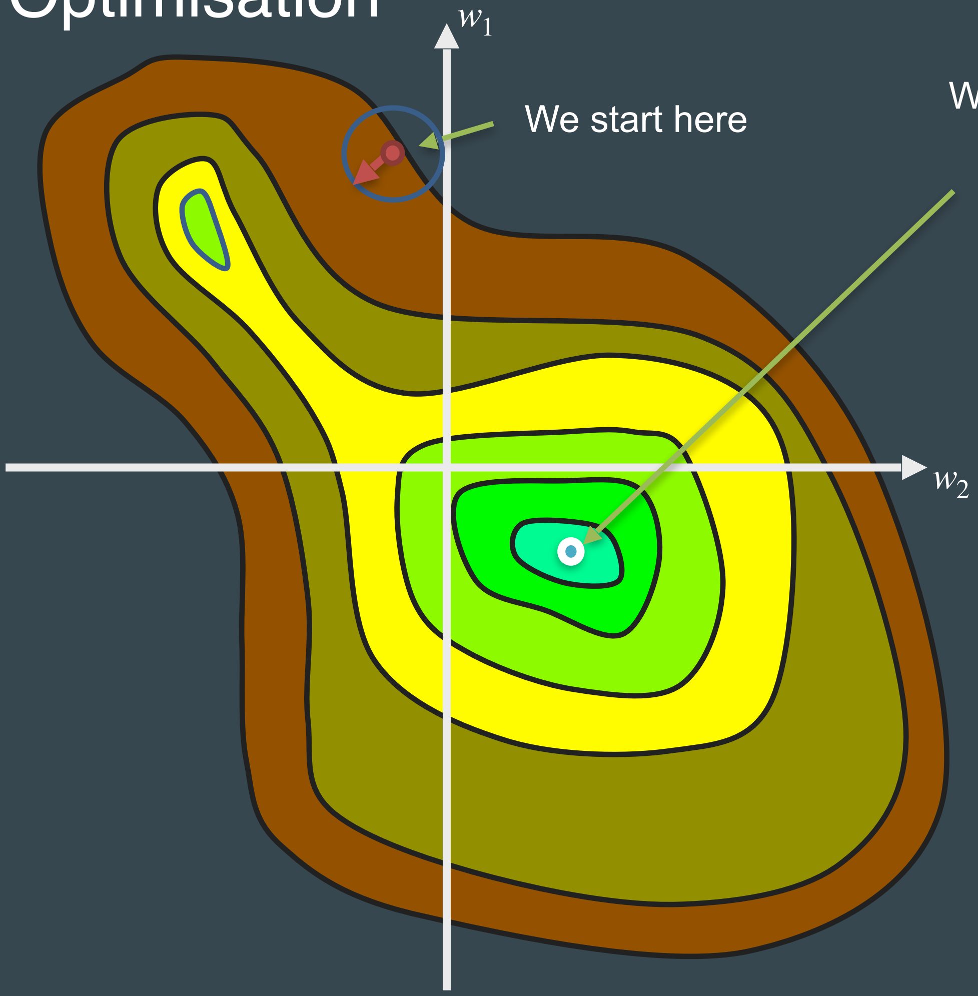
We are in “Switzerland”
In the mist
What would you do?



Optimisation

$$\mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_p \end{bmatrix} \qquad \nabla_{\mathbf{w}} L = \begin{bmatrix} \frac{\partial L}{\partial w_0} \\ \frac{\partial L}{\partial w_1} \\ \vdots \\ \frac{\partial L}{\partial w_p} \end{bmatrix}$$

Optimisation



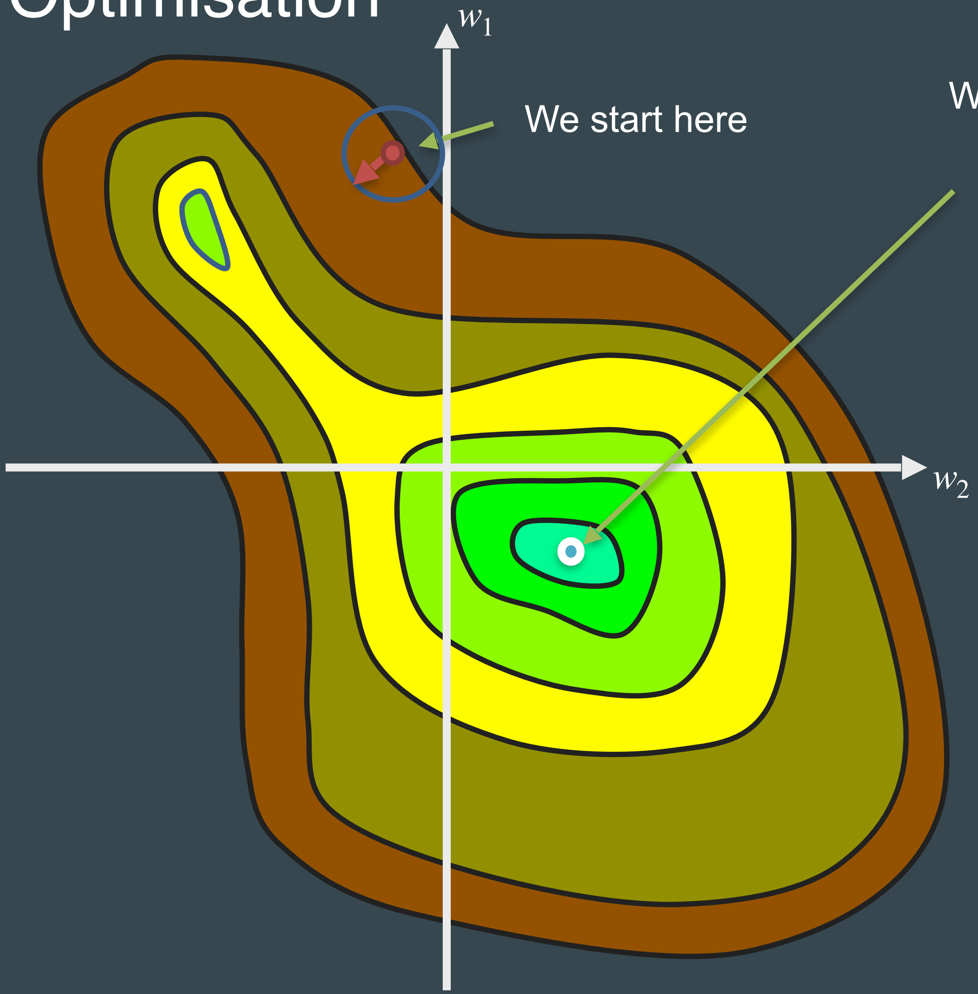
We want to arrive here

$$\mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_p \end{bmatrix} \qquad \nabla_{\mathbf{w}} L = \begin{bmatrix} \frac{\partial L}{\partial w_0} \\ \frac{\partial L}{\partial w_1} \\ \vdots \\ \frac{\partial L}{\partial w_p} \end{bmatrix}$$

$$\mathbf{w}^{t=1} = \mathbf{w}^{t=0} - \alpha \nabla_{\mathbf{w}} L_{total}^{t=0}$$

$$L_{total}^{t=0} = \frac{1}{S} \sum_{s=1}^{S} L(Net_{\mathbf{w^{t=0}}}(x_s), y_s)$$

Optimisation



We want to arrive here

$$\mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_p \end{bmatrix} \qquad \nabla_{\mathbf{w}} L = \begin{bmatrix} \frac{\partial L}{\partial w_0} \\ \frac{\partial L}{\partial w_1} \\ \vdots \\ \frac{\partial L}{\partial w_p} \end{bmatrix}$$

$$\mathbf{w}^{t=1} = \mathbf{w}^{t=0} - \alpha \nabla_{\mathbf{w}} L_{total}^{t=0}$$

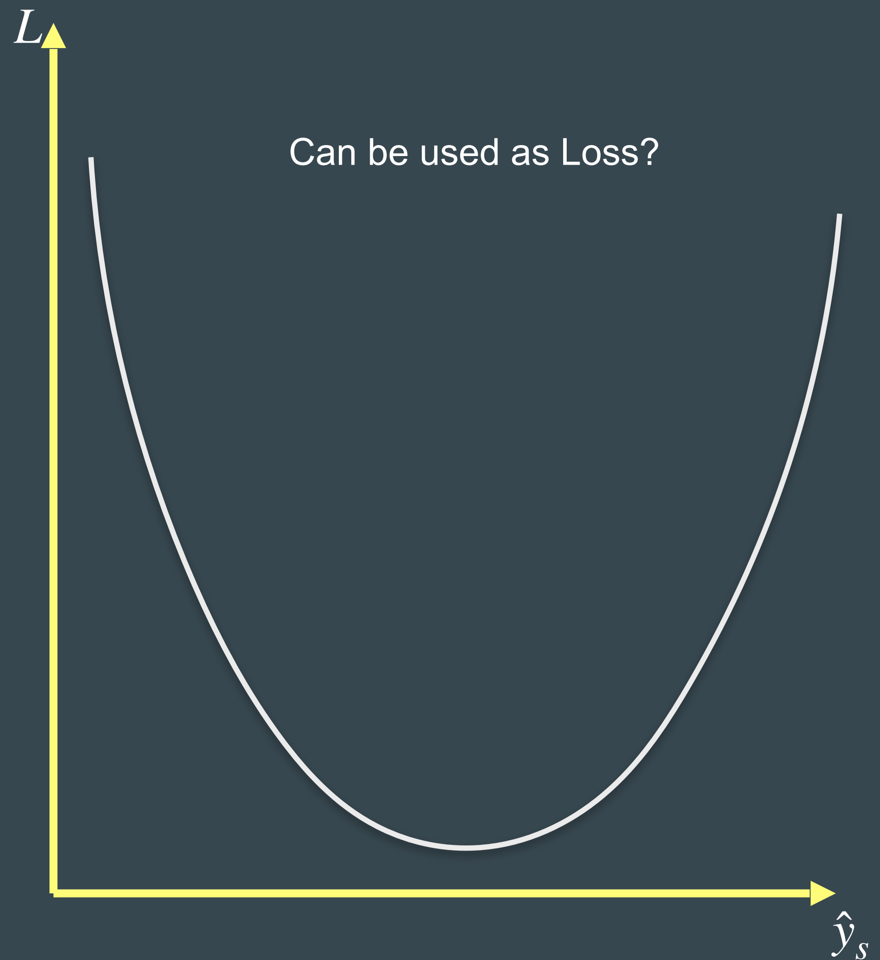
$$\mathbf{w}^{t=2} = \mathbf{w}^{t=1} - \alpha \nabla_{\mathbf{w}} L_{total}^{t=1}$$

$$\dots$$

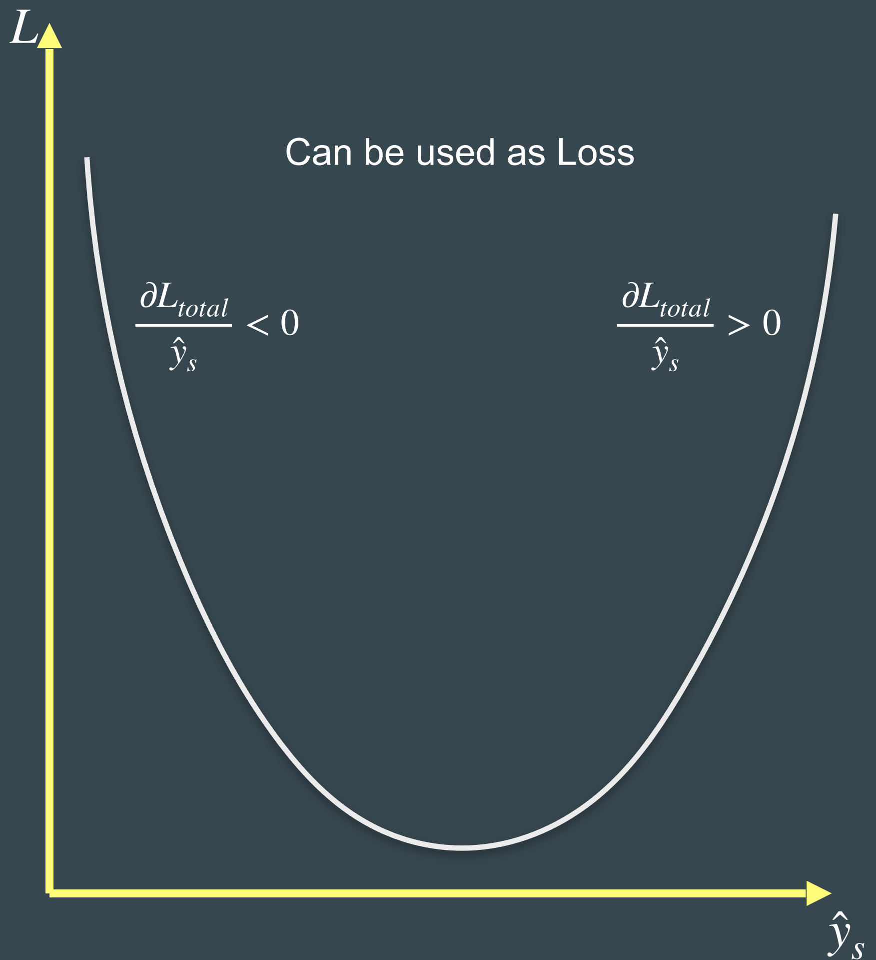
$$\mathbf{w}^{t+1} = \mathbf{w}^{t} - \alpha \nabla_{\mathbf{w}} L_{total}^{t}$$

$$L_{total}^{t=0} = \frac{1}{S} \sum_{s=1}^{S} L(Net_{\mathbf{w^{t=0}}}(x_s), y_s)$$

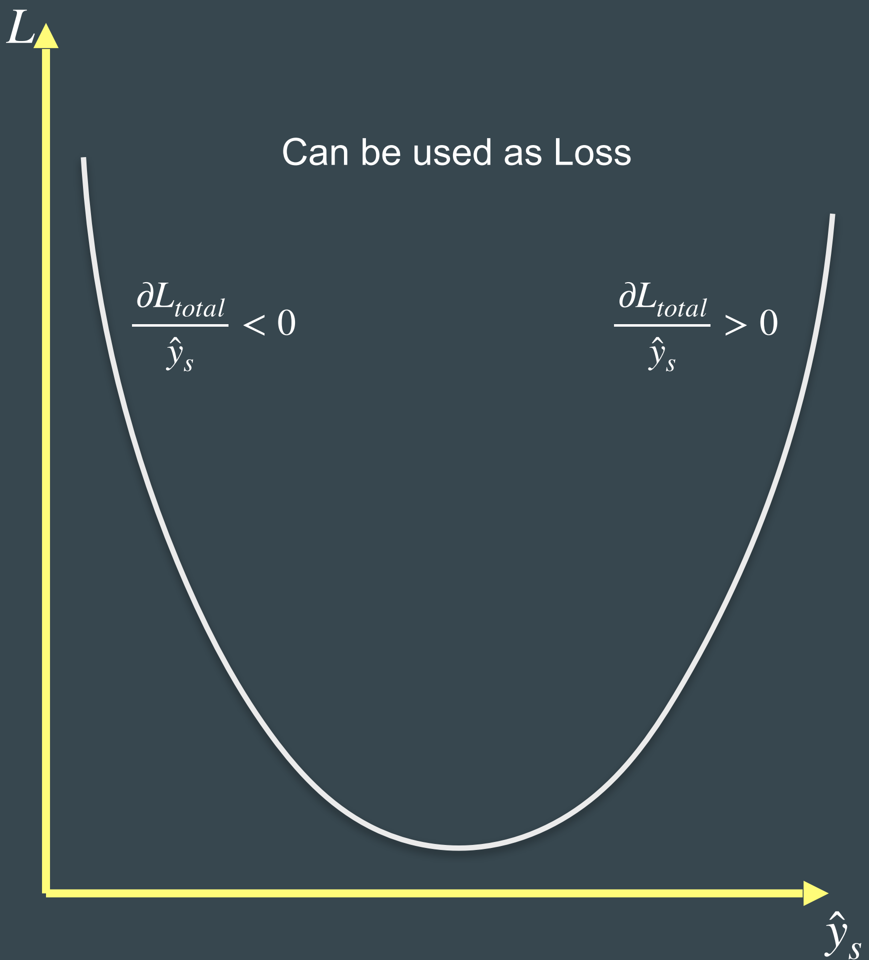
What can be used as Loss Function

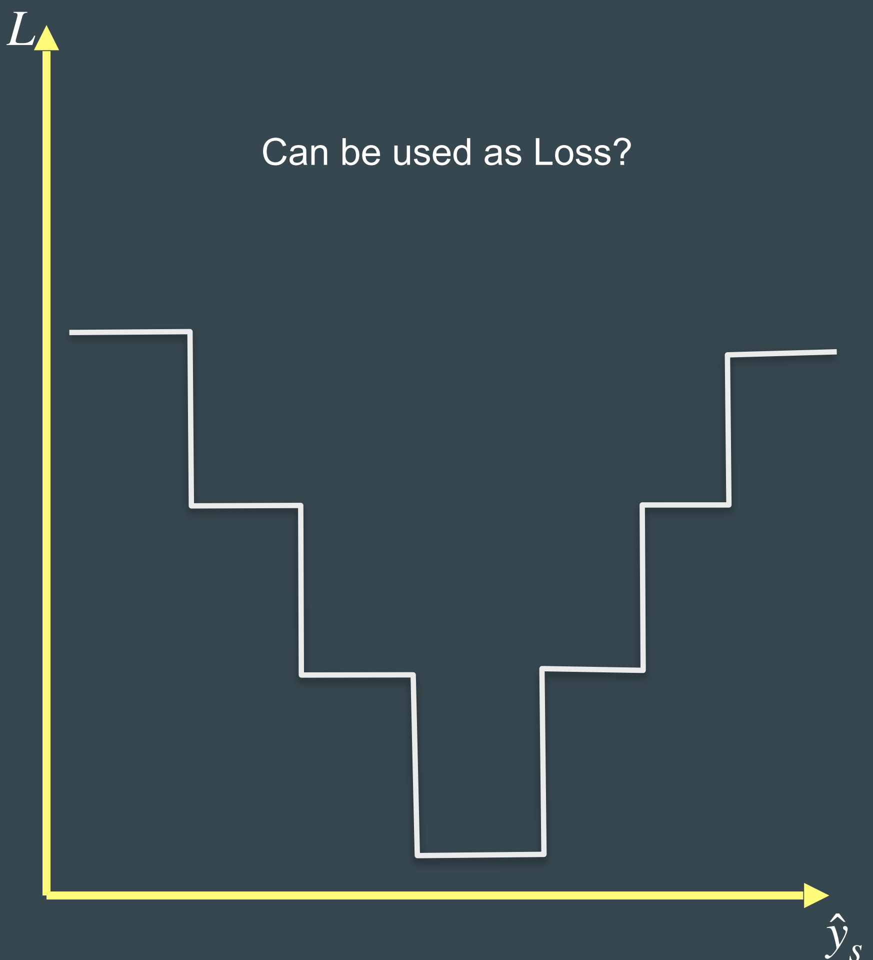


What can be used as Loss Function

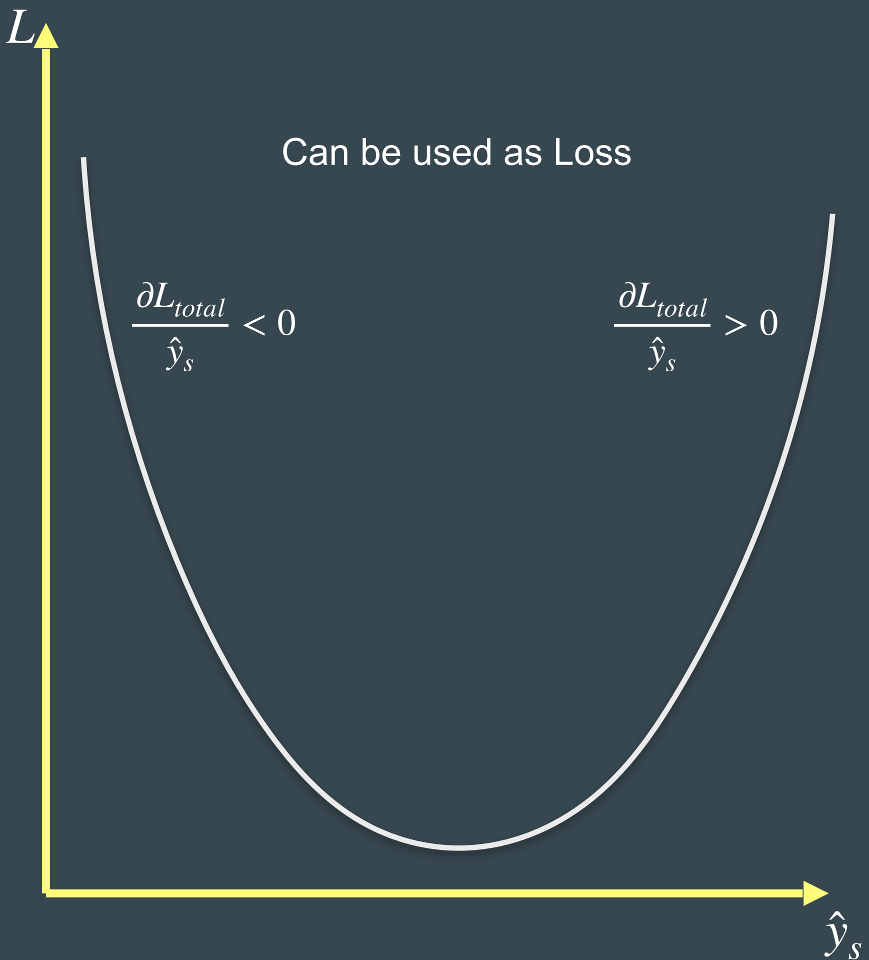


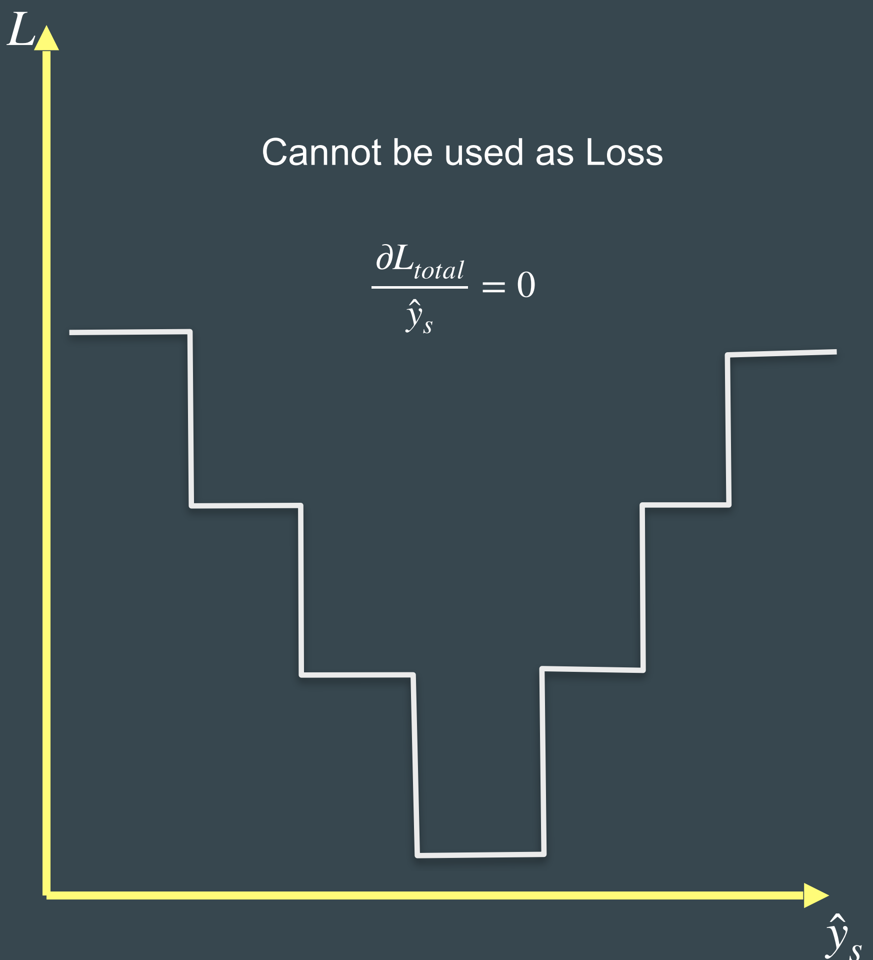
What can be used as Loss Function





What can be used as Loss Function





What is missing?

$$\frac{\partial L_{total}}{\hat{y}_s} \neq 0$$

$$\mathbf{w}^{t+1} = \mathbf{w}^t - \alpha \nabla_{\mathbf{w}} L_{total}^t$$

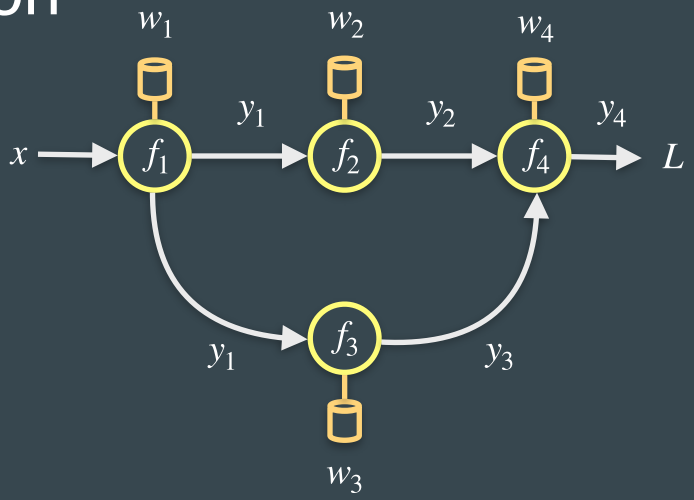
$$\nabla_{\mathbf{w}} L = \begin{bmatrix} \frac{\partial L}{\partial w_1} \\ \frac{\partial L}{\partial w_1} \\ \vdots \\ \frac{\partial L}{\partial w_p} \end{bmatrix}$$

Backpropagation

$$\frac{\partial L_{total}}{\hat{y}_s} \neq 0$$

$$\mathbf{w}^{t+1} = \mathbf{w}^t - \alpha \nabla_{\mathbf{w}} L^t_{total}$$

$$\nabla_{\mathbf{w}} L = \begin{bmatrix} \frac{\partial L}{\partial w_1} \\ \frac{\partial L}{\partial w_1} \\ \vdots \\ \frac{\partial L}{\partial w_p} \end{bmatrix}$$

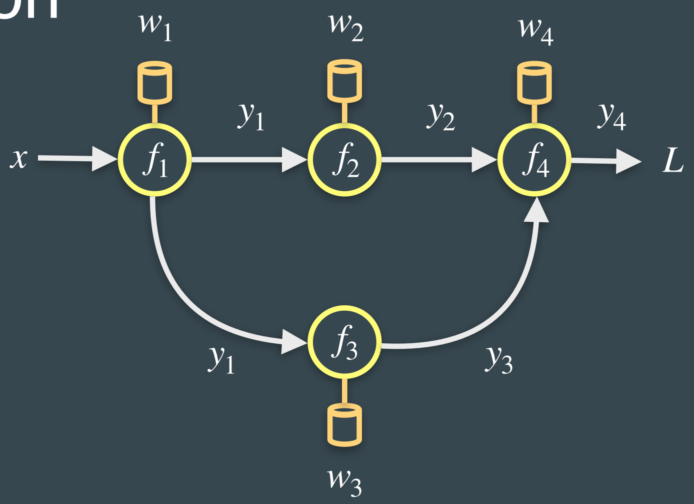


Backpropagation

$$\frac{\partial L_{total}}{\hat{y}_s} \neq 0$$

$$\mathbf{w}^{t+1} = \mathbf{w}^t - \alpha \nabla_{\mathbf{w}} L^t_{total}$$

$$\nabla_{\mathbf{w}} L = \begin{bmatrix} \frac{\partial L}{\partial w_1} \\ \frac{\partial L}{\partial w_1} \\ \vdots \\ \frac{\partial L}{\partial w_p} \end{bmatrix}$$



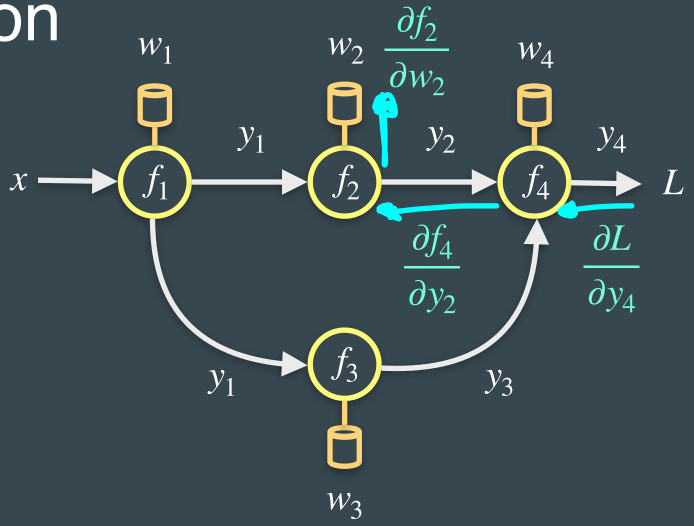
$$L\left(f_4\left(f_3\left(f_1(x)\right), f_2\left(f_1(x)\right)\right)\right)$$

Backpropagation

$$\frac{\partial L_{total}}{\hat{y}_s} \neq 0$$

$$\mathbf{w}^{t+1} = \mathbf{w}^t - \alpha \nabla_{\mathbf{w}} L^t_{total}$$

$$\nabla_{\mathbf{w}} L = \begin{bmatrix} \frac{\partial L}{\partial w_1} \\ \frac{\partial L}{\partial w_1} \\ \vdots \\ \frac{\partial L}{\partial w_p} \end{bmatrix}$$



$$L\left(f_4\left(f_3\left(f_1(x)\right), f_2\left(f_1(x)\right)\right)\right)$$

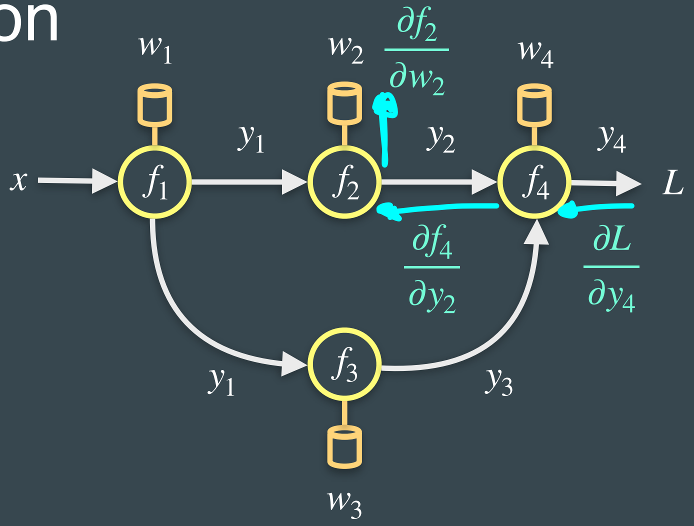
$$\frac{\partial L}{\partial w_2}$$

Backpropagation

$$\frac{\partial L_{total}}{\hat{y}_s} \neq 0$$

$$\mathbf{w}^{t+1} = \mathbf{w}^{t} - \alpha \nabla_{\mathbf{w}} L^{t}_{total}$$

$$\nabla_{\mathbf{w}} L = \begin{bmatrix} \frac{\partial L}{\partial w_1} \\ \frac{\partial L}{\partial w_1} \\ \vdots \\ \frac{\partial L}{\partial w_p} \end{bmatrix}$$



$$L\left(f_4\left(f_3\left(f_1(x)\right), f_2\left(f_1(x)\right)\right)\right)$$

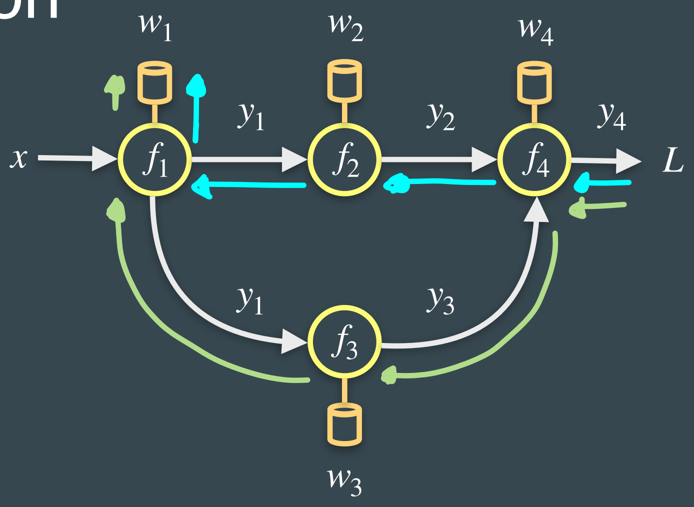
$$\frac{\partial L}{\partial w_2} = \frac{\partial L}{\partial y_4}\frac{\partial f_4}{\partial y_2}\frac{\partial f_2}{\partial w_2}$$

Backpropagation

$$\frac{\partial L_{total}}{\hat{y}_s} \neq 0$$

$$\mathbf{w}^{t+1} = \mathbf{w}^{t} - \alpha \nabla_{\mathbf{w}} L_{total}^{t}$$

$$\nabla_{\mathbf{w}} L = \begin{bmatrix} \frac{\partial L}{\partial w_1} \\ \frac{\partial L}{\partial w_1} \\ \vdots \\ \frac{\partial L}{\partial w_p} \end{bmatrix}$$



$$L\left(f_4\left(f_3\left(f_1(x)\right), f_2\left(f_1(x)\right)\right)\right)$$

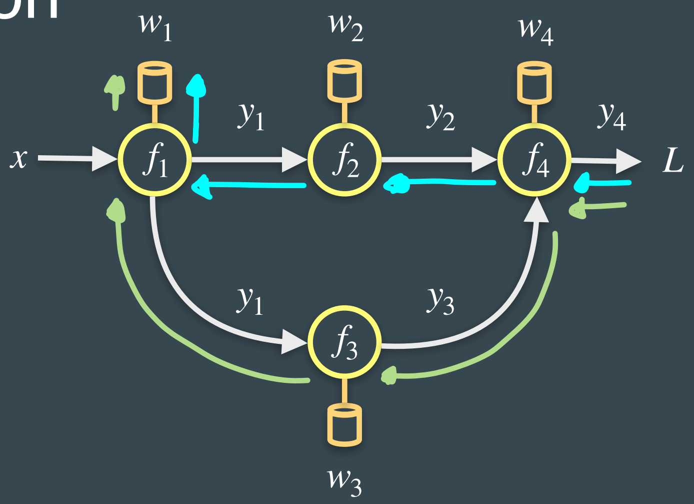
$$\frac{\partial L}{\partial w_1}$$

Backpropagation

$$\frac{\partial L_{total}}{\hat{y}_s} \neq 0$$

$$\mathbf{w}^{t+1} = \mathbf{w}^t - \alpha \nabla_{\mathbf{w}} L^t_{total}$$

$$\nabla_{\mathbf{w}} L = \begin{bmatrix} \frac{\partial L}{\partial w_1} \\ \frac{\partial L}{\partial w_1} \\ \vdots \\ \frac{\partial L}{\partial w_p} \end{bmatrix}$$



$$L\left(f_4\left(f_3\left(f_1(x)\right), f_2\left(f_1(x)\right)\right)\right)$$

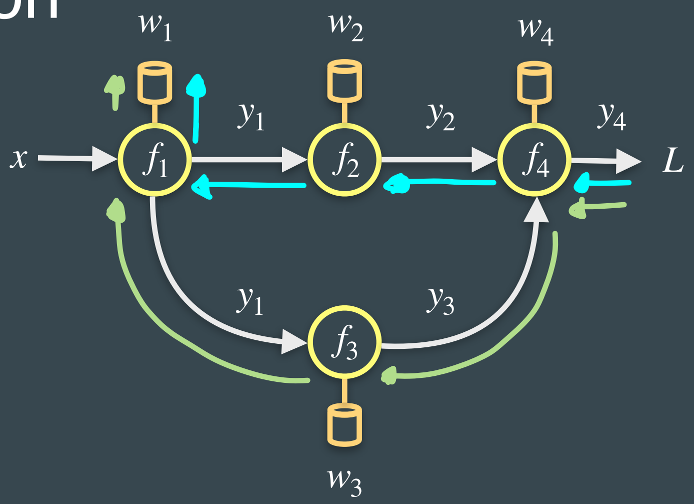
$$\frac{\partial L}{\partial w_1} = \underbrace{\frac{\partial L}{\partial y_4}\frac{\partial f_4}{\partial y_2}\frac{\partial f_2}{\partial y_1}\frac{\partial f_1}{\partial w_1}} + \underbrace{\frac{\partial L}{\partial y_4}\frac{\partial f_4}{\partial y_3}\frac{\partial f_3}{\partial y_1}\frac{\partial f_1}{\partial w_1}}$$

Backpropagation

$$\frac{\partial L_{total}}{\hat{y}_s} \neq 0$$

$$\mathbf{w}^{t+1} = \mathbf{w}^t - \alpha \nabla_{\mathbf{w}} L^t_{total}$$

$$\nabla_{\mathbf{w}} L = \begin{bmatrix} \frac{\partial L}{\partial w_1} \\ \frac{\partial L}{\partial w_1} \\ \vdots \\ \frac{\partial L}{\partial w_p} \end{bmatrix}$$



$$L\left(f_4\left(f_3\left(f_1(x)\right), f_2\left(f_1(x)\right)\right)\right)$$

$$\frac{\partial L}{\partial w_1} = \underbrace{\frac{\partial L}{\partial y_4}\frac{\partial f_4}{\partial y_2}\frac{\partial f_2}{\partial y_1}\frac{\partial f_1}{\partial w_1}} + \underbrace{\frac{\partial L}{\partial y_4}\frac{\partial f_4}{\partial y_3}\frac{\partial f_3}{\partial y_1}\frac{\partial f_1}{\partial w_1}}$$

$$\frac{\partial L}{\partial w_1} = \frac{\partial L}{\partial y_4}\left(\frac{\partial f_4}{\partial y_2}\frac{\partial f_2}{\partial y_1} + \frac{\partial f_4}{\partial y_3}\frac{\partial f_3}{\partial y_1}\right)\frac{\partial f_1}{\partial w_1}$$

Summary

- Example: dependency reconstruction
- Training philosophy
- Training cycle
- Two-layer FCNN and its awesomeness! — ARCHITECTURE
- MSE — LOSS
- Accuracy — METRICS
- Gradient Descent— OPTIMISER
- Backpropagation —Gradient calculation method