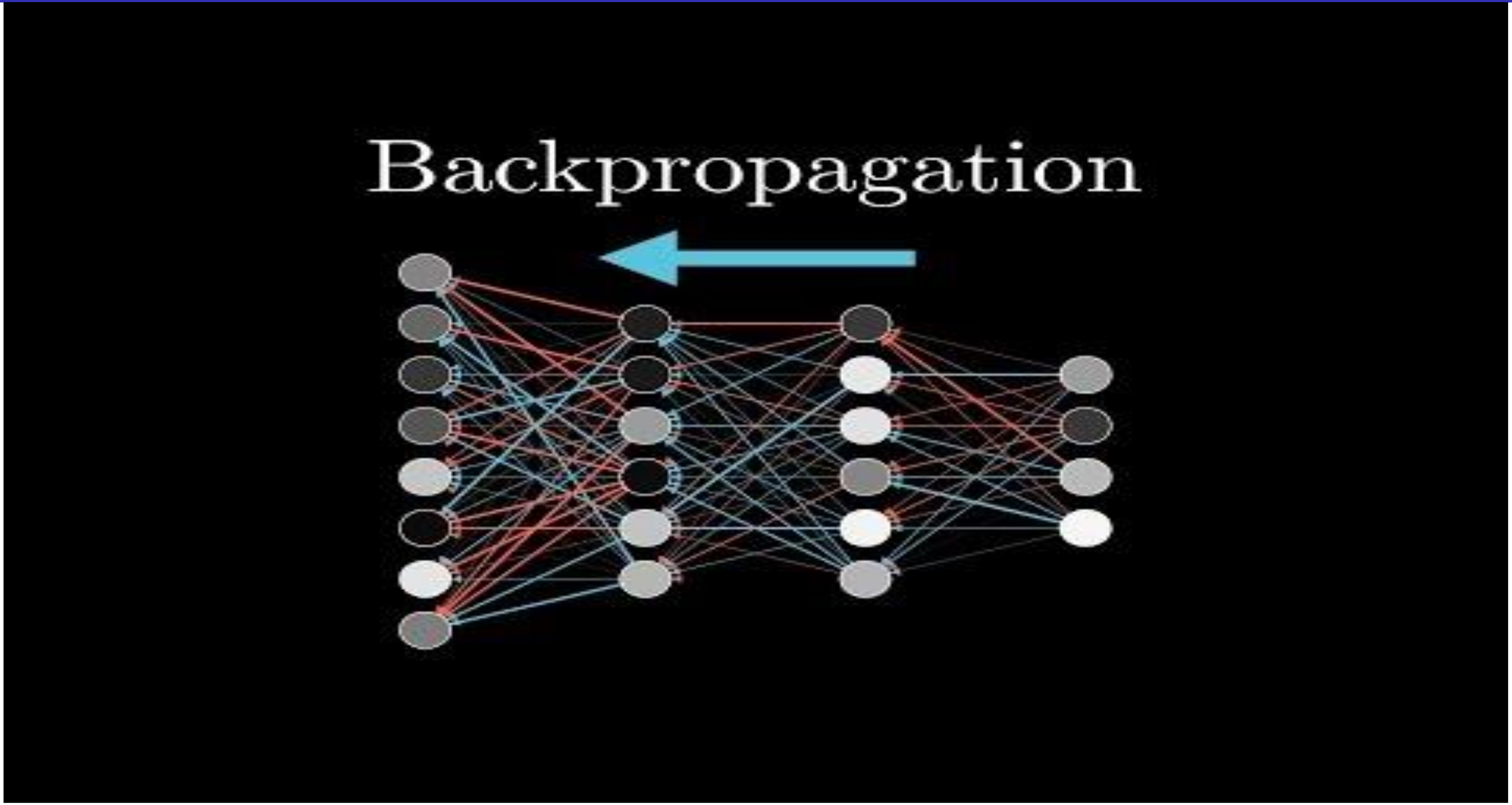
MLP and BP

C. V. Jawahar

IIIT Hyderabad

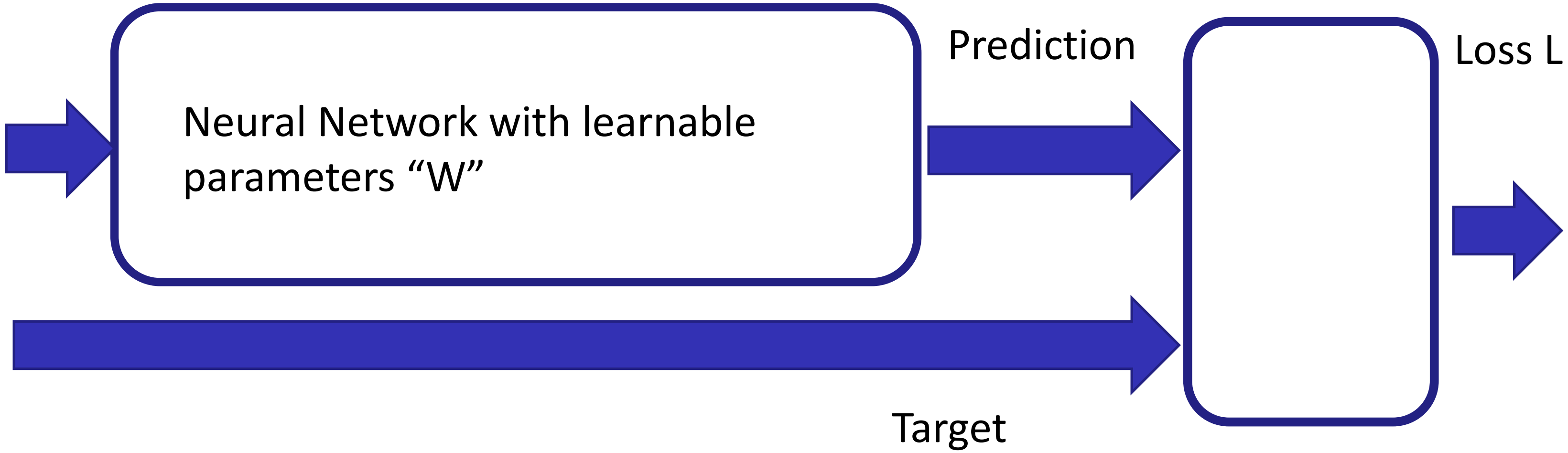
28 Mar, 2025

Video



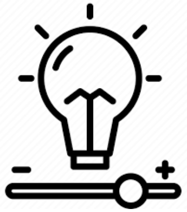
Link to the video: https://www.youtube.com/watch?v=Ilg3gGewQ5U

Basic Problem



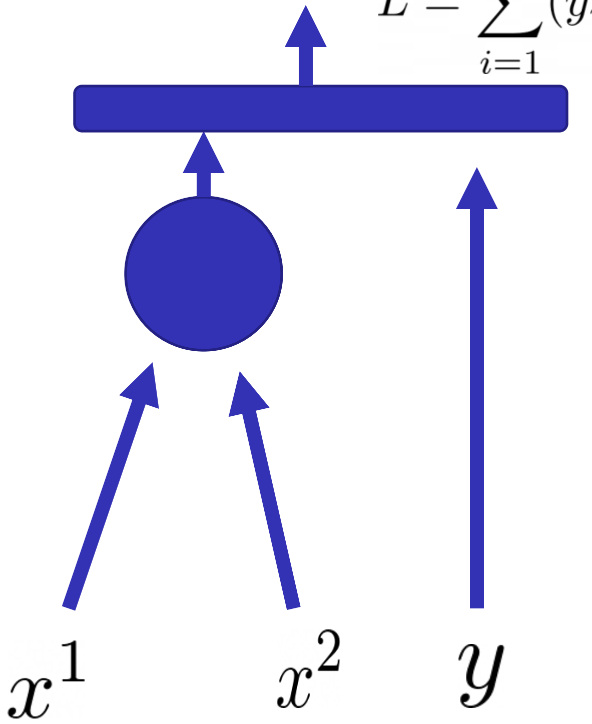
Goal: Minimize Loss L by adjusting/updating the weights W of the neural network.

Motivating Analogy



Many many Knobs (~Millions)
You can increase or decrease each one.
Goal: Control the intensity of the bulb

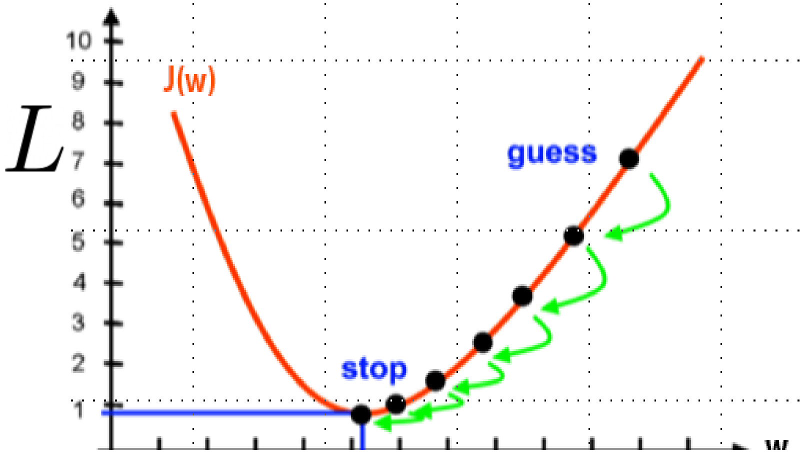
Gradient Descent



$$L = \sum_{i=1}^{N}(y_i - o_i)^2 = \sum_{i=1}^{N}(y_i - (w_1 x_i^1 + w_2 x_i^2))^2$$

$$\frac{\partial L}{\partial w_1} = \sum_{i=1}^{N} 2(y_i - o_i)(-1)(x_i^1)$$

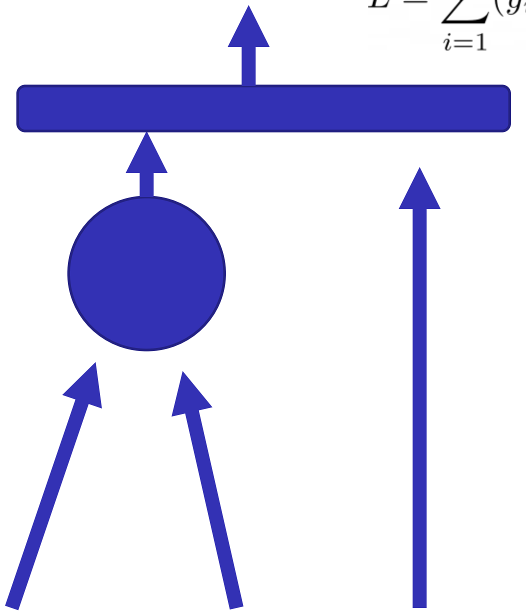
$$\frac{\partial L}{\partial w_2} = \sum_{i=1}^{N} 2(y_i - o_i)(-1)(x_i^2)$$



$$w_1^{n+1} = w_1^n - \eta \frac{\partial L}{\partial w_1}$$

$$w_2^{n+1} = w_2^n - \eta \frac{\partial L}{\partial w_2}$$

Gradient Descent: Simple Vector Notation

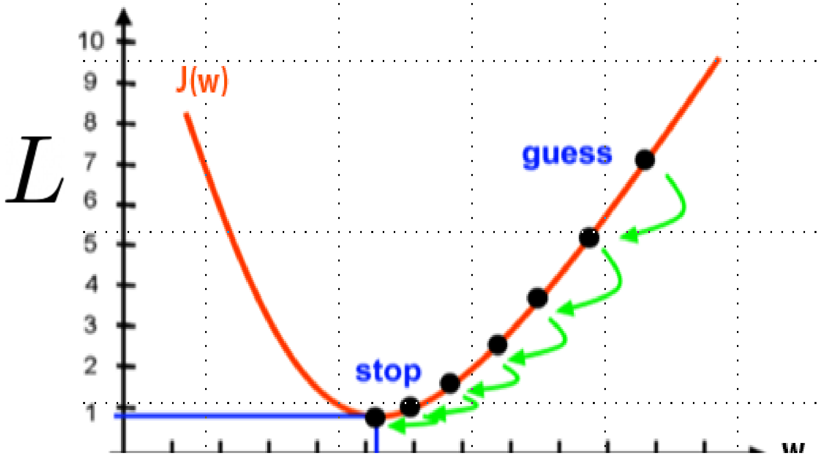


$$L = \sum_{i=1}^{N}(y_i - o_i)^2 = \sum_{i=1}^{N}(y_i - \mathbf{w}^T\mathbf{x})^2$$

$$\frac{\partial L}{\partial \mathbf{w}} = \sum_{i=1}^{N} 2(y_i - o_i)(-1)\mathbf{x}$$

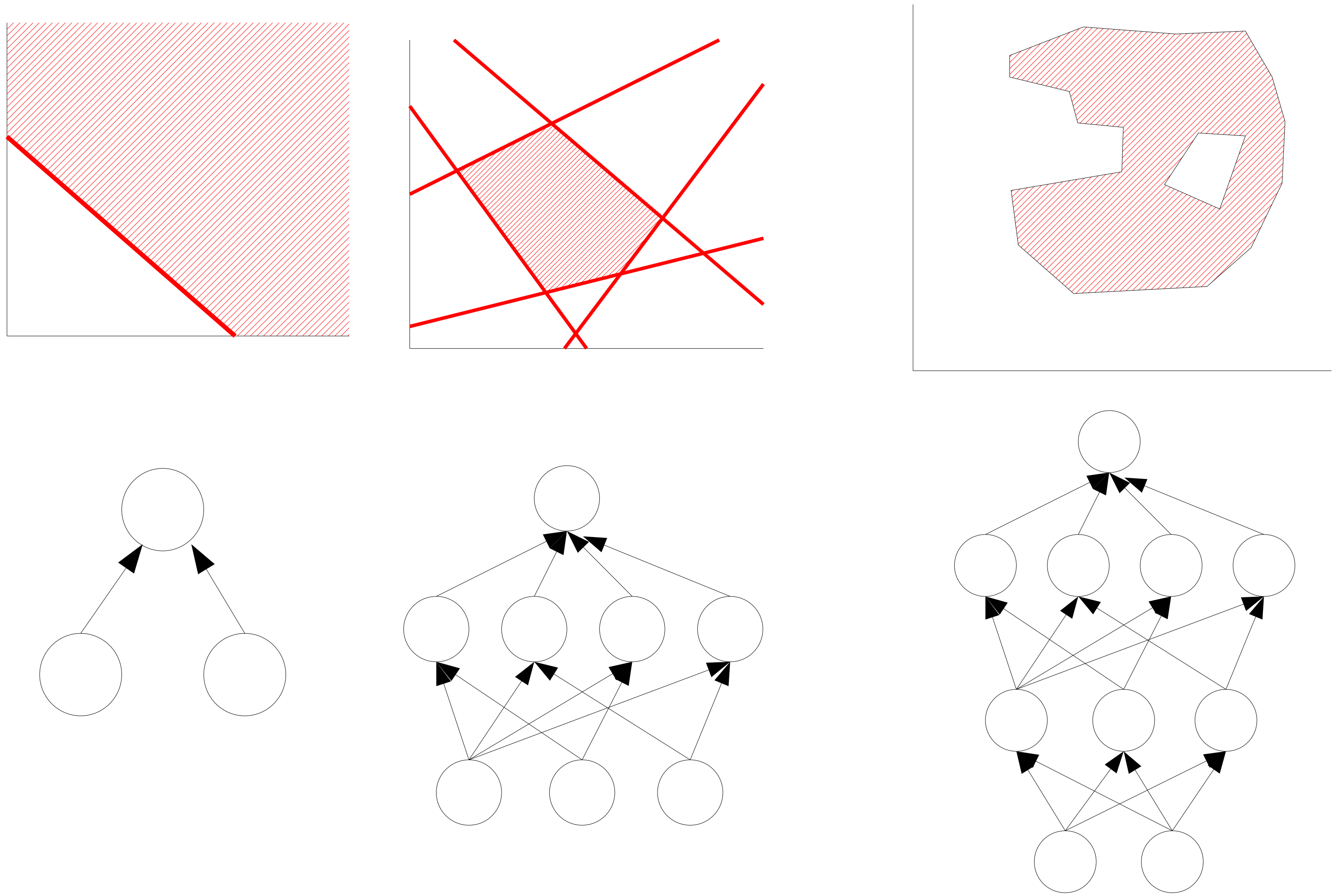
$x^1 \; x^2 \quad y$

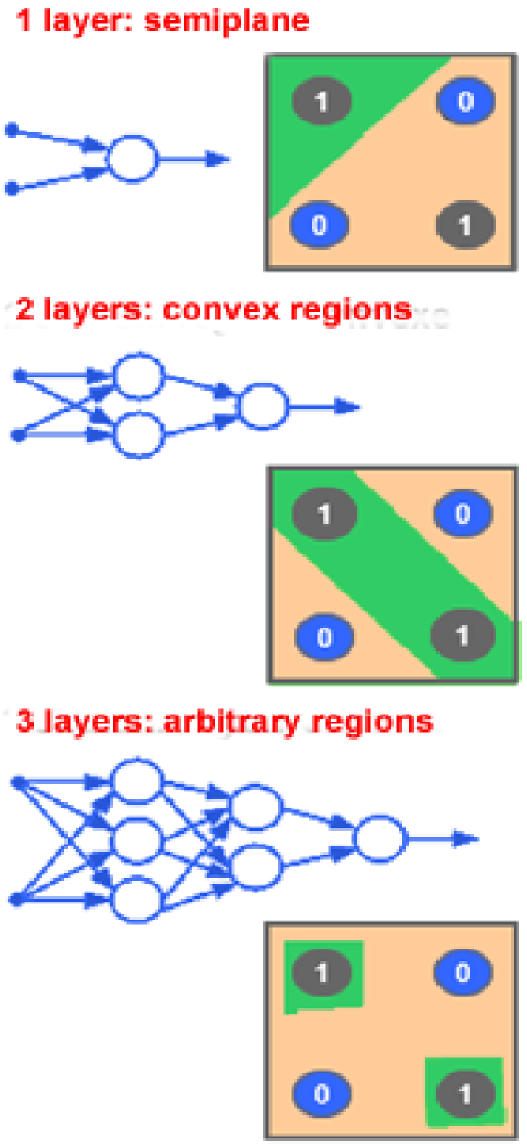
$$\mathbf{x} = [x^1, x^2]^T$$



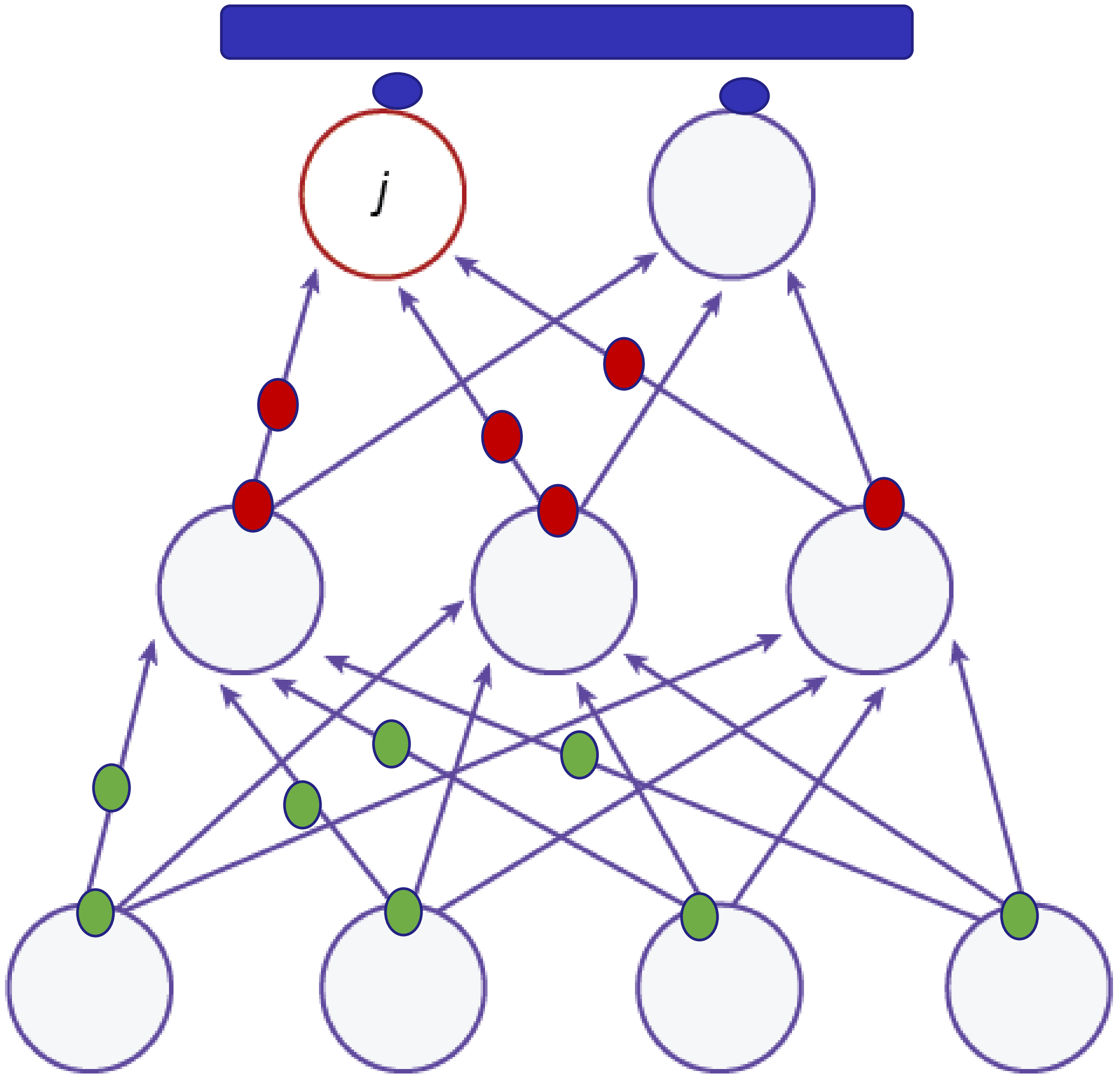
$$\mathbf{w}^{n+1} = \mathbf{w}^n - \eta \frac{\partial L}{\partial \mathbf{w}}$$

Deeper Networks





Error depends on many weights



Compute Loss/Error; Loss Layer

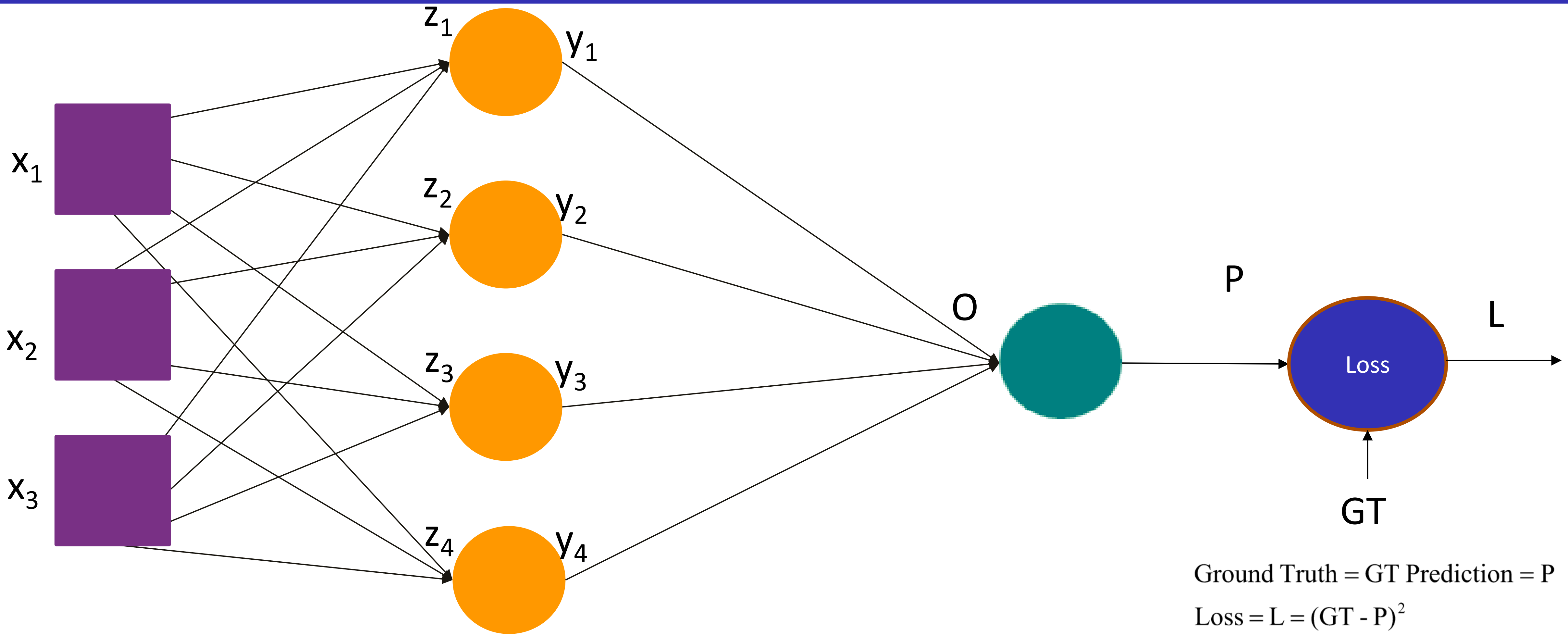
Error depends on two terms/outputs

Each one depends on three weights and three intermediate values

Each one of these intermediate values depends on four weights and four inputs

To minimize error/loss, we need to adjust/vary all the 18 weights.!!

Back Propagation



Ground Truth = GT Prediction = P

Loss = L = $(GT - P)^2$

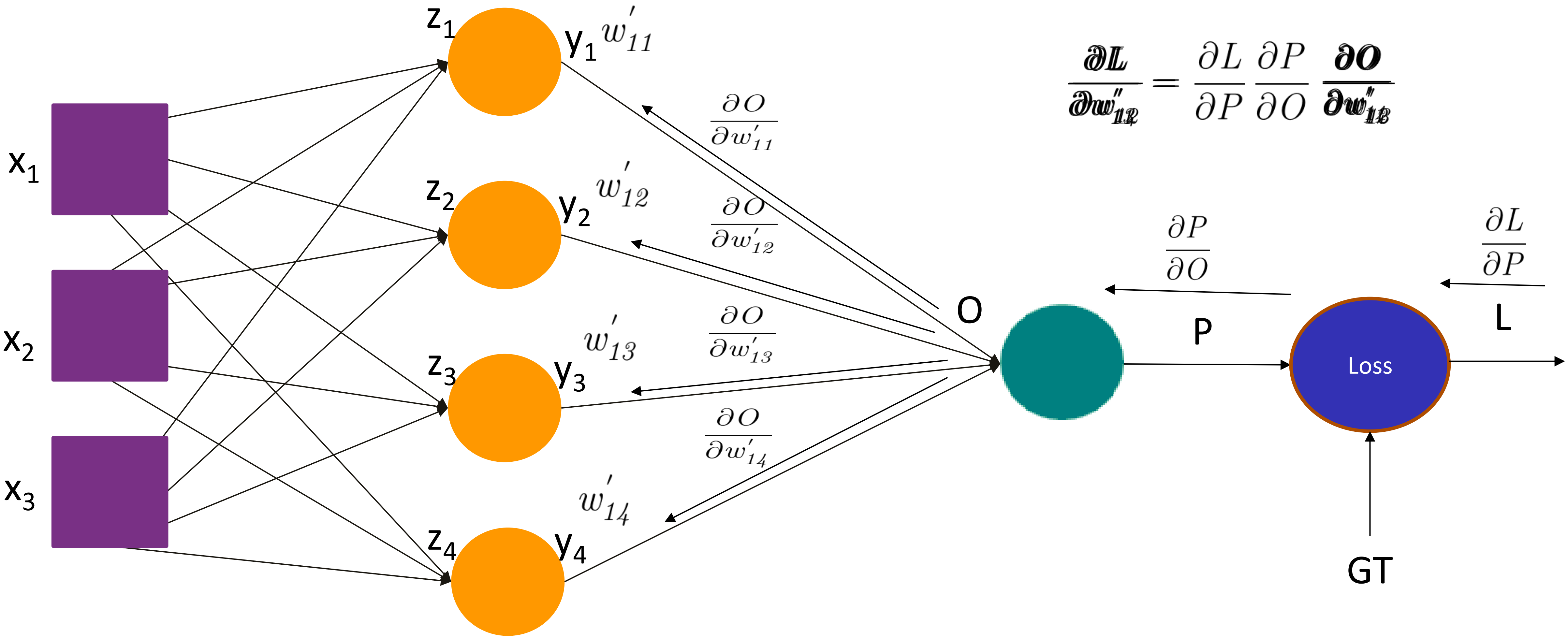
$$\begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} = \begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \\ w_{41} & w_{42} & w_{43} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} \phi(z_1) \\ \phi(z_2) \\ \phi(z_3) \\ \phi(z_4) \end{bmatrix}$$

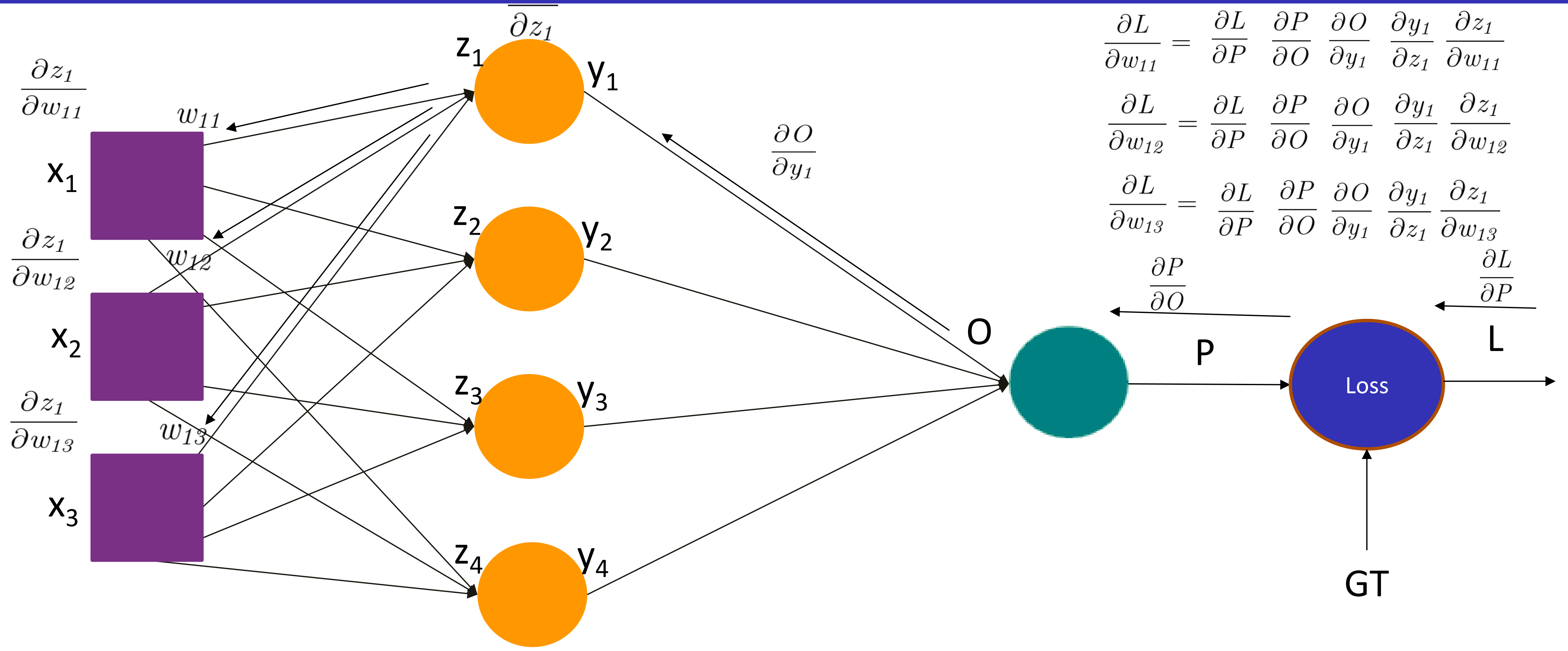
$$\mathbf{O} = \begin{bmatrix} w'_{11} & w'_{12} & w'_{13} & w'_{14} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}$$

$$\mathbf{P} = \phi(\mathbf{O})$$

Back Propagation



Back Propagation



$$\frac{\partial L}{\partial w_{11}} = \frac{\partial L}{\partial P}\frac{\partial P}{\partial O}\frac{\partial O}{\partial y_1}\frac{\partial y_1}{\partial z_1}\frac{\partial z_1}{\partial w_{11}}$$

$$\frac{\partial L}{\partial w_{12}} = \frac{\partial L}{\partial P}\frac{\partial P}{\partial O}\frac{\partial O}{\partial y_1}\frac{\partial y_1}{\partial z_1}\frac{\partial z_1}{\partial w_{12}}$$

$$\frac{\partial L}{\partial w_{13}} = \frac{\partial L}{\partial P}\frac{\partial P}{\partial O}\frac{\partial O}{\partial y_1}\frac{\partial y_1}{\partial z_1}\frac{\partial z_1}{\partial w_{13}}$$

Two Typical Blocks (with and without weights)

$$\mathbf{z} = \mathbf{W}\mathbf{x} \qquad \mathbf{y} = \phi(\mathbf{z})$$

$$\mathbf{x} \rightarrow \boxed{W} \rightarrow \mathbf{z} \rightarrow \boxed{\phi} \rightarrow \mathbf{y}$$

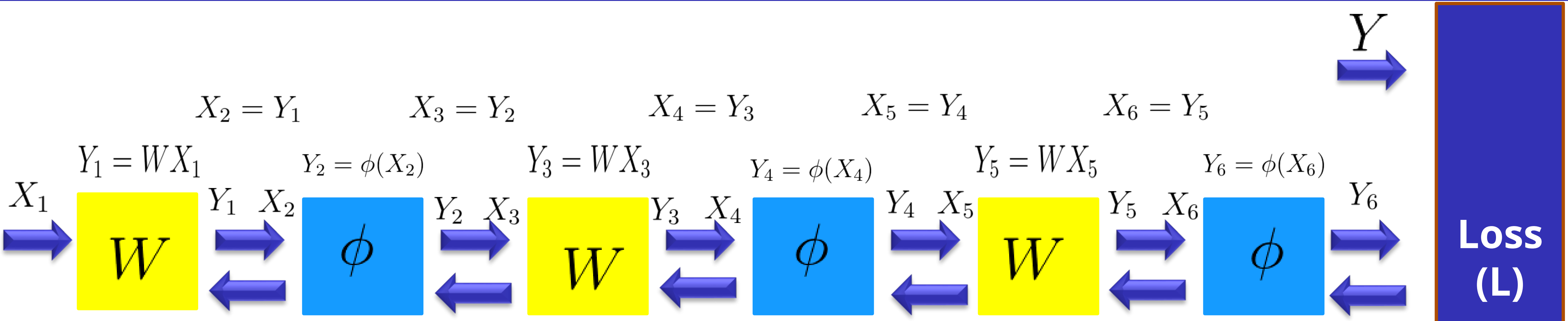
$$\frac{\partial \mathbf{z}}{\partial \mathbf{x}} = \mathbf{W} \qquad \frac{\partial \mathbf{y}}{\partial \mathbf{z}} = \phi'(\mathbf{z})$$

Eg. Sigmoid

$$y = \phi(z) = \frac{1}{1 + e^{-z}}$$

$$\phi'(z) = \phi(z) \cdot (1 - \phi(z))$$

Back Propagation:



$\frac{\partial L}{\partial Y_1}$ $\frac{\partial L}{\partial Y_2}$ $\frac{\partial L}{\partial Y_3}$ $\frac{\partial L}{\partial Y_4}$ $\frac{\partial L}{\partial Y_5}$ $\frac{\partial L}{\partial Y_6}$

$$\frac{\partial L}{\partial Y_1} = \frac{\partial L}{\partial X_2} = \frac{\partial L}{\partial Y_2} \cdot \frac{\partial Y_2}{\partial X_2}$$

$$\frac{\partial L}{\partial Y_3} = \frac{\partial L}{\partial X_4} = \frac{\partial L}{\partial Y_4} \cdot \frac{\partial Y_4}{\partial X_4}$$

$$\frac{\partial L}{\partial Y_5} = \frac{\partial L}{\partial X_6} = \frac{\partial L}{\partial Y_6} \cdot \frac{\partial Y_6}{\partial X_6}$$

$$\frac{\partial L}{\partial Y_2} = \frac{\partial L}{\partial X_3} = \frac{\partial L}{\partial Y_3} \cdot \frac{\partial Y_3}{\partial X_3}$$

$$\frac{\partial L}{\partial Y_4} = \frac{\partial L}{\partial X_5} = \frac{\partial L}{\partial Y_5} \cdot \frac{\partial Y_5}{\partial X_5}$$

$$\frac{\partial L}{\partial W} = \frac{\partial L}{\partial Y_1} \cdot \frac{\partial Y_1}{\partial W}$$

$$\frac{\partial L}{\partial W} = \frac{\partial L}{\partial Y_3} \cdot \frac{\partial Y_3}{\partial W}$$

$$\frac{\partial L}{\partial W} = \frac{\partial L}{\partial Y_5} \cdot \frac{\partial Y_5}{\partial W}$$

$$W^{n+1} = W^n - \eta \frac{dL}{dW}$$

Basic Algorithm

Forward Pass

Take a batch of samples; Compute outputs; Compute Loss.

Backward Pass

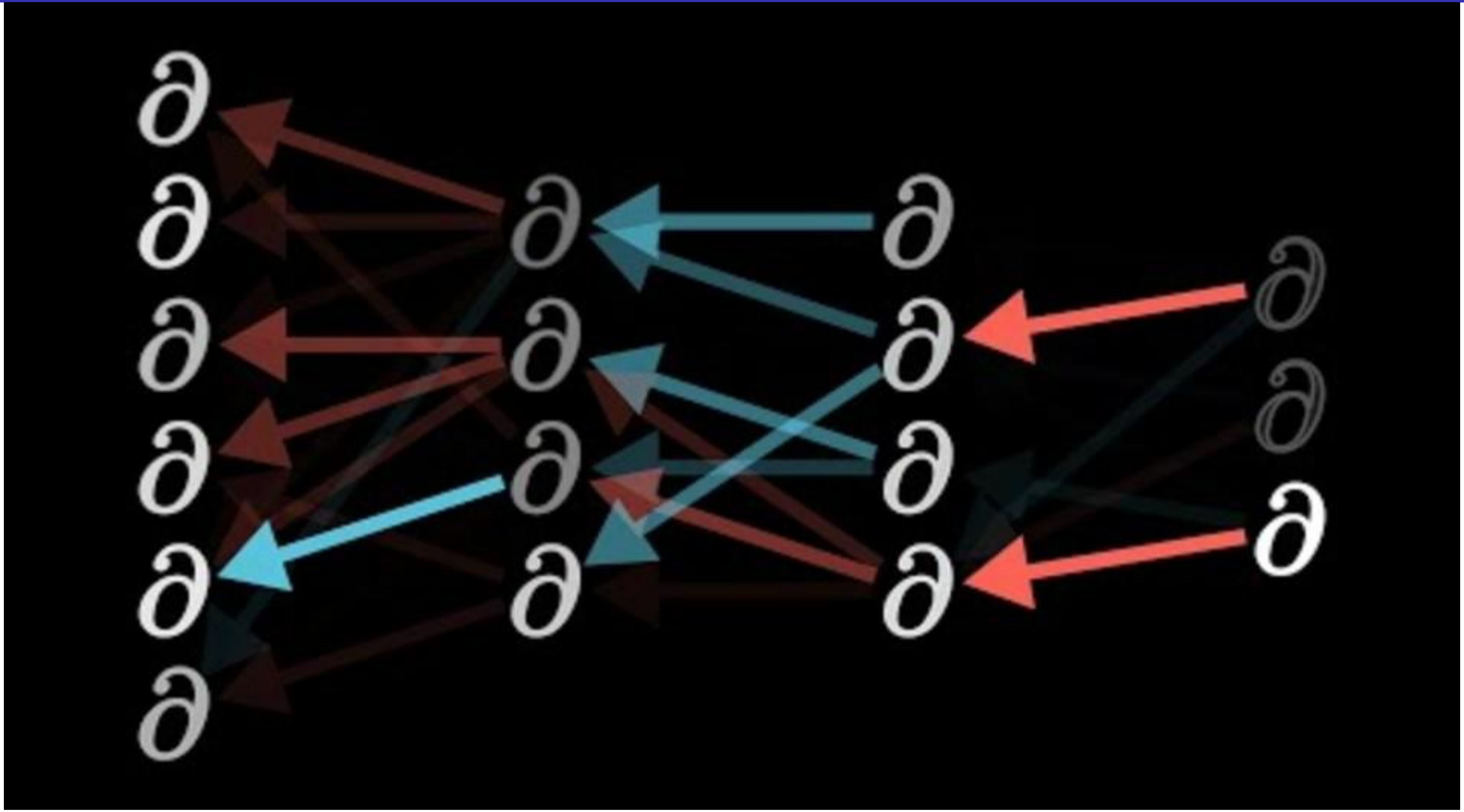
Update all the weights (backwards) and reduce the errors.

Repeat the steps; Until?

Improvements

Basic; Good to Discuss	Worth; Insightful	Advanced	
Mini Batch, Stochastic, SGD	Activation Fns, Learning Rate.	Better Initialization (Xavier, He)	**Design or Initial.**
Momentum (Nesterov)	Numerical/Parallelism (including Vanishing Exploding Gradients)	Better Update (Adam, Adagrad, RMSProp etc.)	**Update**
Early Stopping	Regularization (L1, L2)	Regularization (Dropout, Data Augmentation etc.)	**Regularize**

Video



Link to the video: https://www.youtube.com/watch?v=tIeHLnjs5U8

Questions?