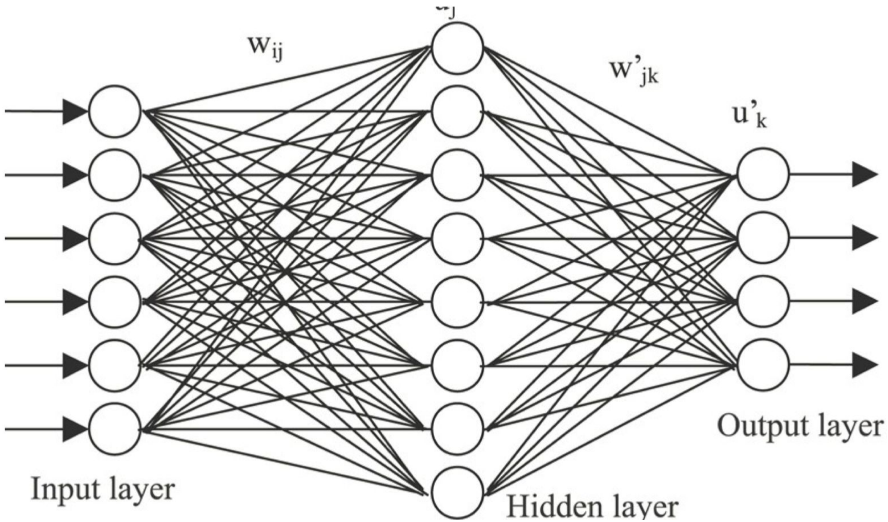
Lecture - 3

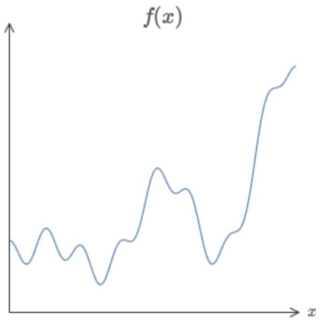
Training a neural Network - 1

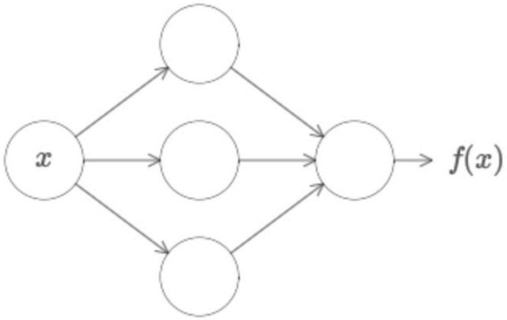
So far …

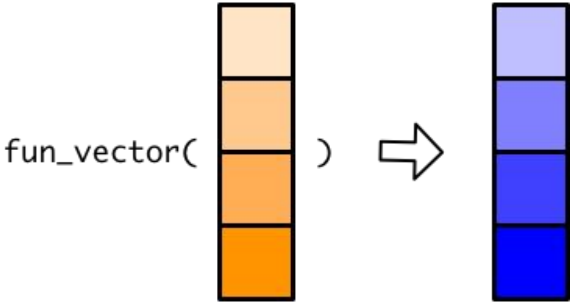


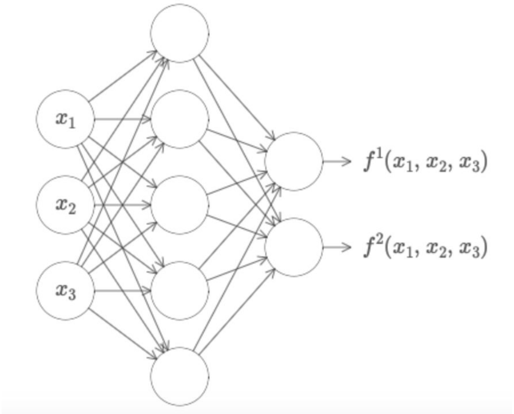
- Neural network (Network of Perceptron/MLP) can:
  - model any Boolean function
  - Model any decision boundary
  - Model any continuous valued function (how?)

Neural Network can compute any func.



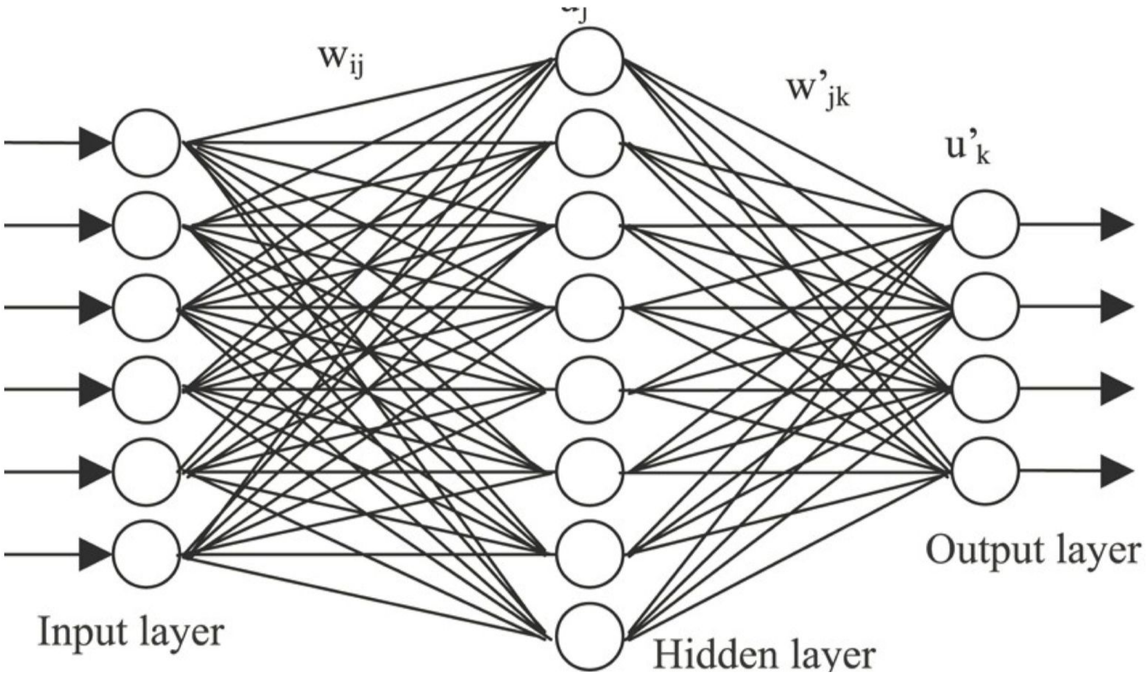






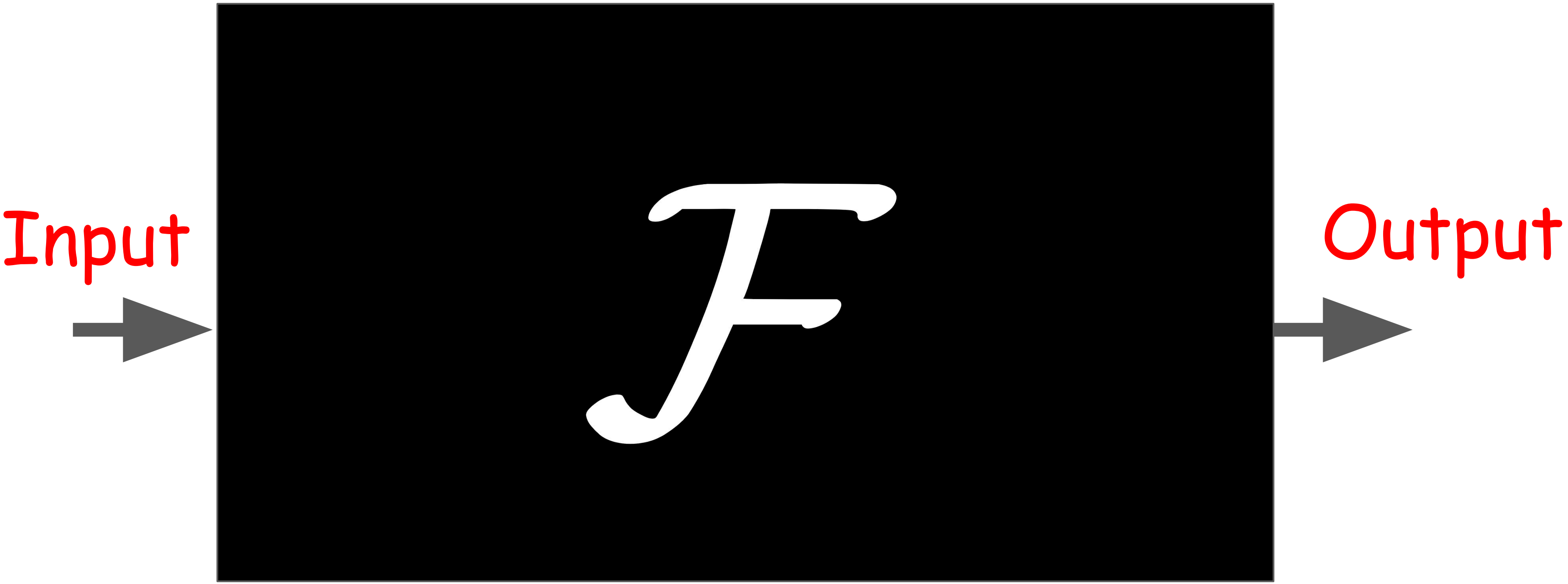
Neural Network

Input

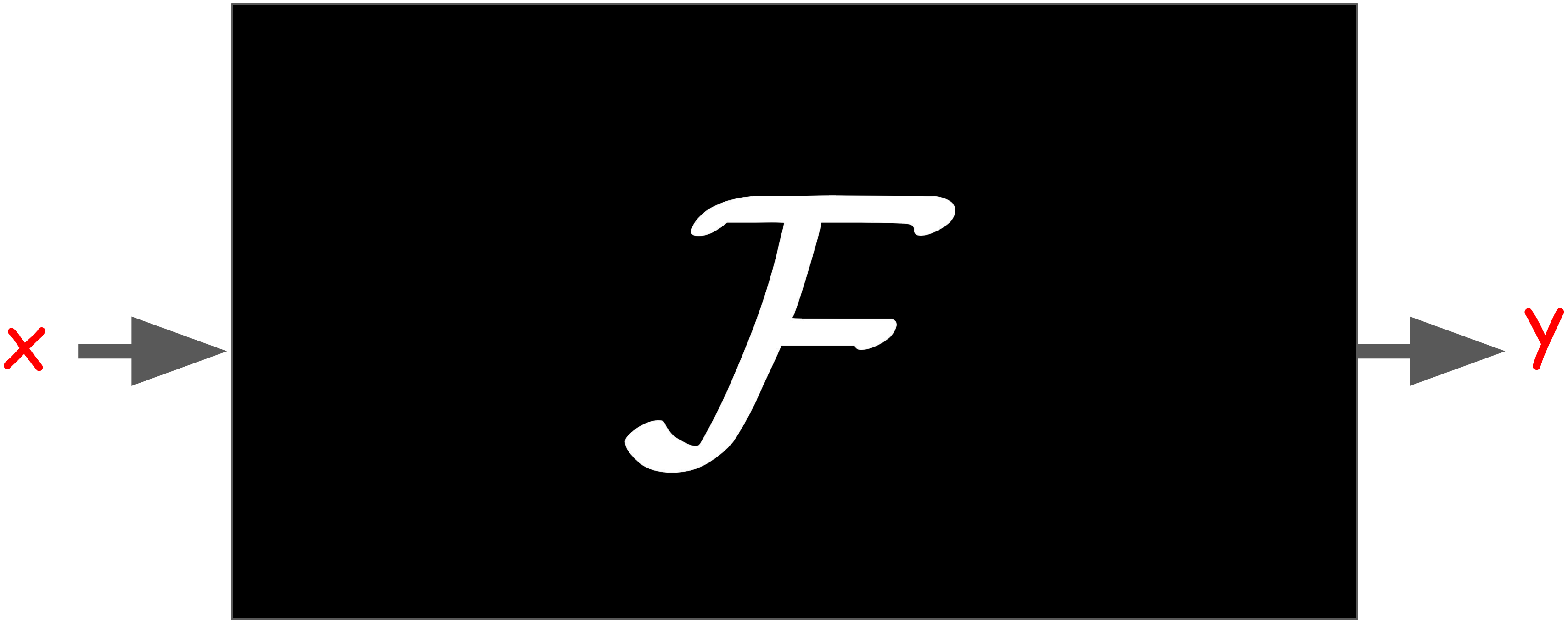


Output

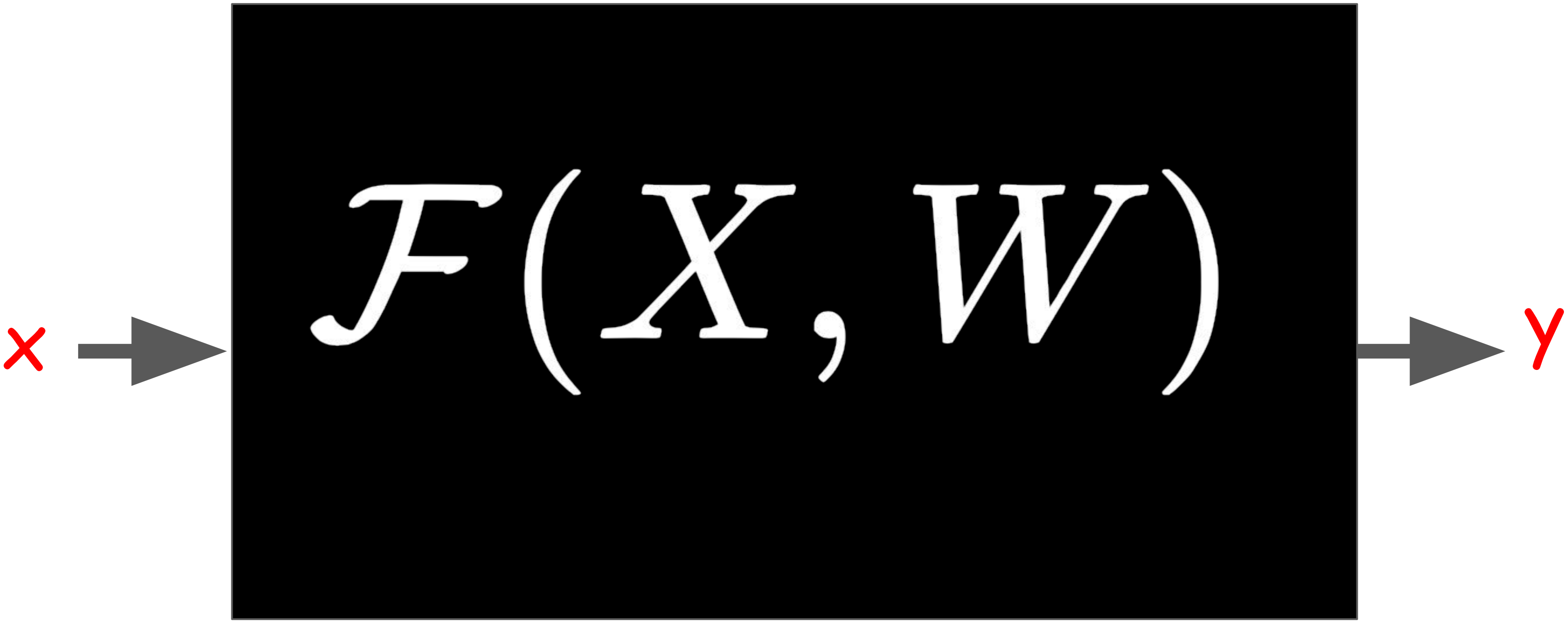
Neural Network as a function



Neural Network as a function



Neural Network as a function

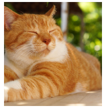


Neural Network as a function

$$\mathcal{F}(X, W)$$

dog

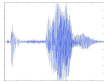
Neural Network as a function



$$\mathcal{F}(X, W)$$

cat

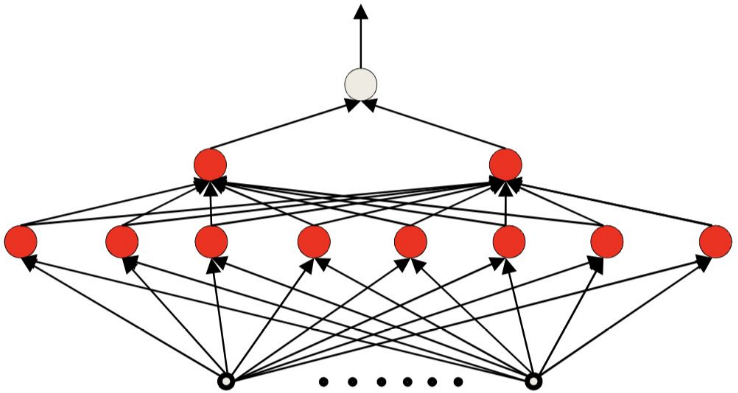
Neural Network as a function

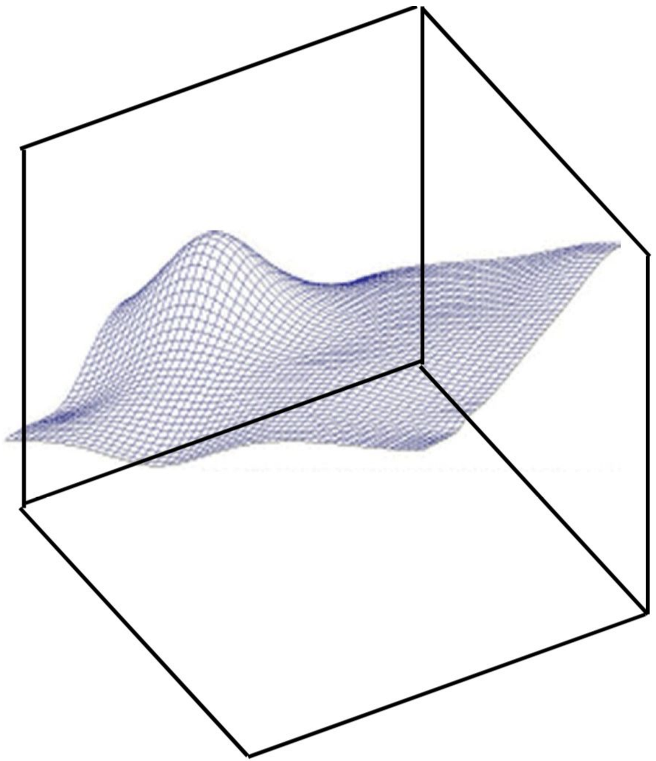


$$\mathcal{F}(X, W)$$

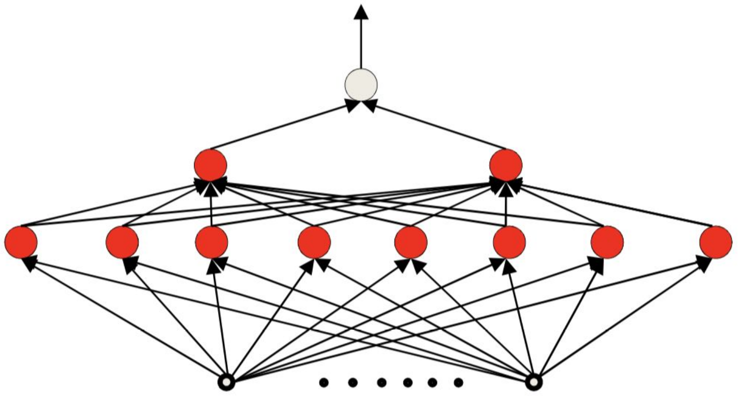
I love
Neural
Network

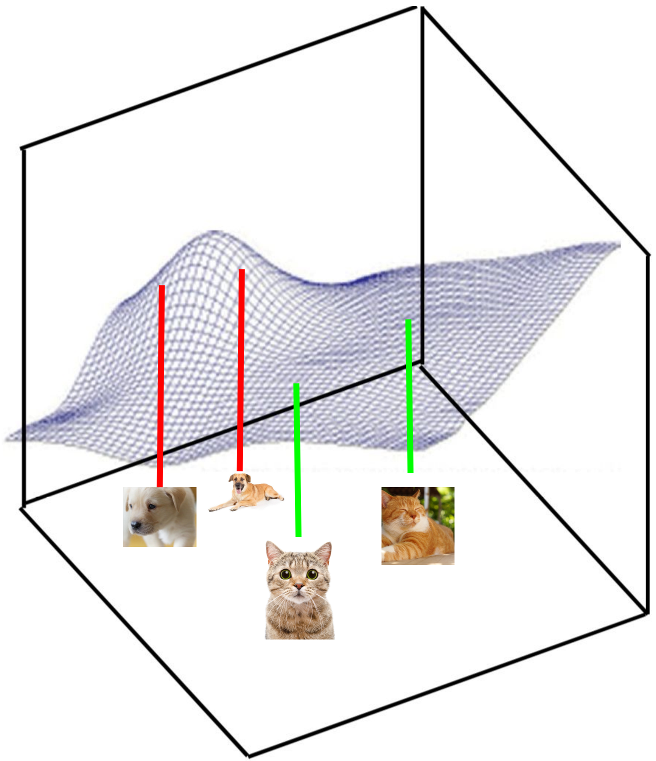
MLP Can represent anything





MLP Can represent anything





MLP Can represent anything

Option 1: Follow Lecture 2 and construct by hand

Option 2: Automatic estimation of an MLP

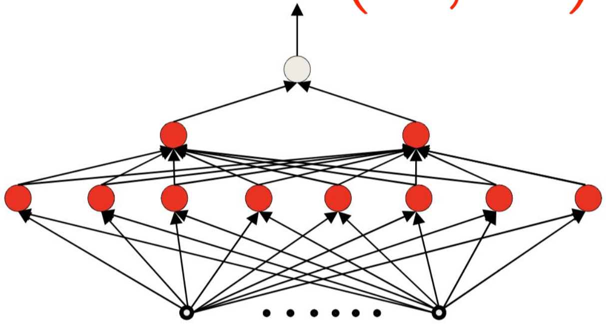
MLP Can represent anything

~~Option 1: Follow Lecture 2 and construct by hand~~

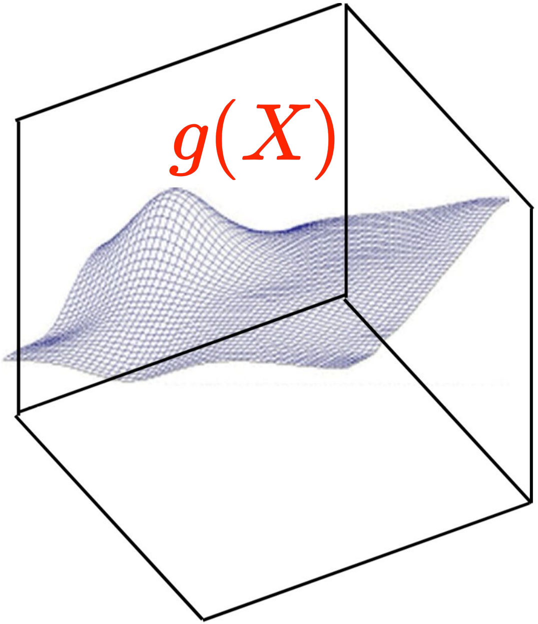
Option 2: Automatic estimation of an MLP

Automatic estimation of MLP

$$Y = \mathcal{F}(X, W)$$

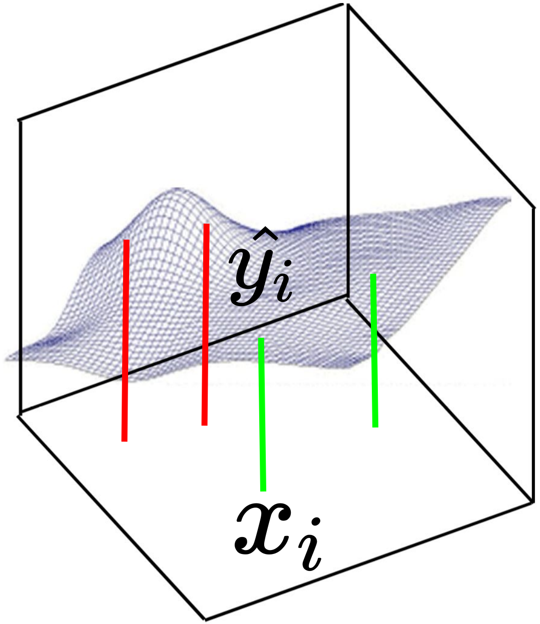


X

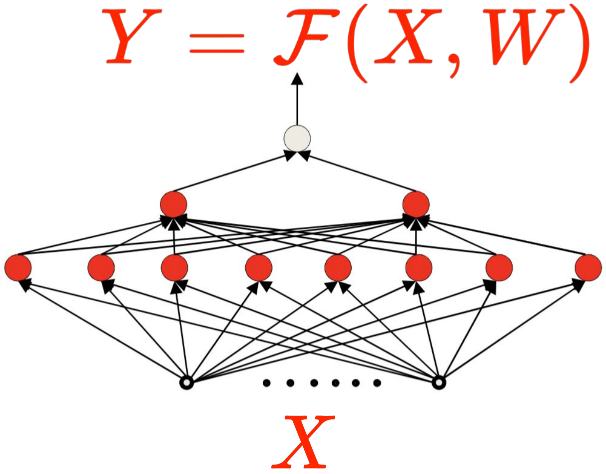


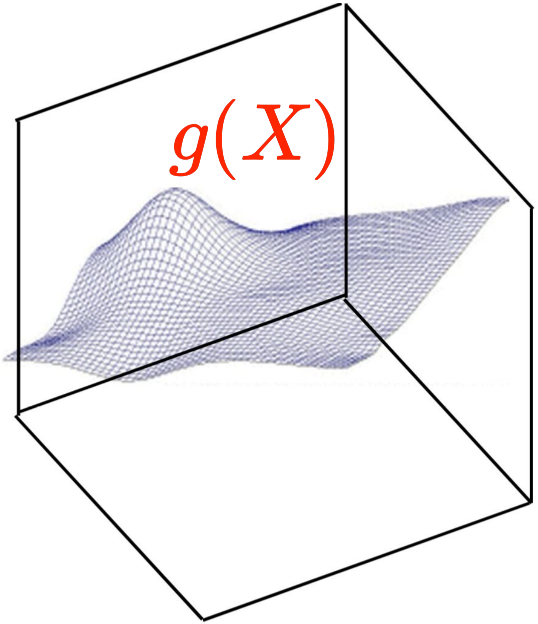
$$W^* = \arg\min_W \int_X \mathcal{DIV}(\mathcal{F}(X, W), g(X))$$

Problem: g(x) is not known everywhere



Automatic estimation of MLP





$$W^* = \arg\min_W \sum_{i=1}^{n} \mathcal{DIV}\left(\mathcal{F}(x_i, W), \hat{y}_i\right)$$

Module 1: Learning Algorithm

Supervised Learning Setup

Data: $\mathcal{D} = \{\mathbf{x}_i, \hat{y}_i\}_{i=1}^{n}$, *where* $\mathbf{x}_i = \{x_{i1}, x_{i2}, \cdots, x_{im}\}$ is a sample, and $\hat{y}_i \in \{+1, -1\}$ is a class label.

Supervised Learning Setup

Data: $\mathcal{D} = \{\mathbf{x}_i, \hat{y}_i\}_{i=1}^{n}$, $where\ \mathbf{x}_i = \{x_{i1}, x_{i2}, \cdots, x_{im}\}$ is a sample, and $\hat{y}_i \in \{+1, -1\}$ is a class label.

$$y = \mathcal{F}(\mathbf{x}, \mathbf{w})$$

Supervised Learning Setup

Data: $\mathcal{D} = \{\mathbf{x}_i, \hat{y}_i\}_{i=1}^{n}$, *where* $\mathbf{x}_i = \{x_{i1}, x_{i2}, \cdots, x_{im}\}$ is a sample, and $\hat{y}_i \in \{+1, -1\}$ is a class label.

$$y = \mathcal{F}(\mathbf{x}, \mathbf{w})$$

Supervised Learning Setup

Data: $\mathcal{D} = \{\mathbf{x}_i, \hat{y}_i\}_{i=1}^{n}$, *where* $\mathbf{x}_i = \{x_{i1}, x_{i2}, \cdots, x_{im}\}$ is a sample, and $\hat{y}_i \in \{+1, -1\}$ is a class label.

Model: $y = \mathcal{F}(\mathbf{x}, \mathbf{w})$ where $\mathcal{F}$ is a neural network.

Parameters: $\mathbf{w}$ needs to be learnt.

Learning Algo: Perceptron learning algo., gradient descent

Loss function: To guide learning algorithm

The problem

$$W^* = \arg\min_W \sum_{i=1}^{n} \mathcal{DIV}\big(\mathcal{F}(x_i, W), \hat{y}_i\big)$$

The problem

$$\mathcal{L}(W) = \boxed{\mathcal{DIV}}(\mathcal{F}(W, x), \hat{y})$$

Divergence (intuitively a function f(a,b) which has lower value when a = b)

Example of Divergence function

- Least Mean Square

$$DIV(y, \hat{y}) = \frac{1}{2}(y - \hat{y})^2$$

- Cross-entropy

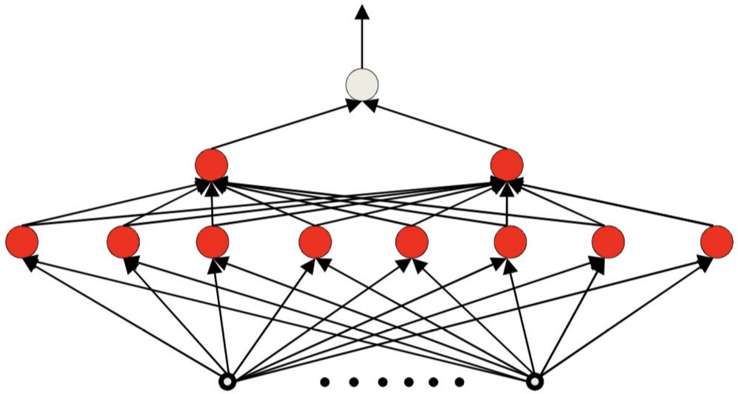
$$\mathcal{DIV}(y, \hat{y}) = -ylog(\hat{y}) - (1 - y)log(1 - \hat{y}))$$

The Problem

$$\mathcal{L}(W) = \mathcal{DIV}(\boxed{\mathcal{F}}(W, x), \hat{y})$$

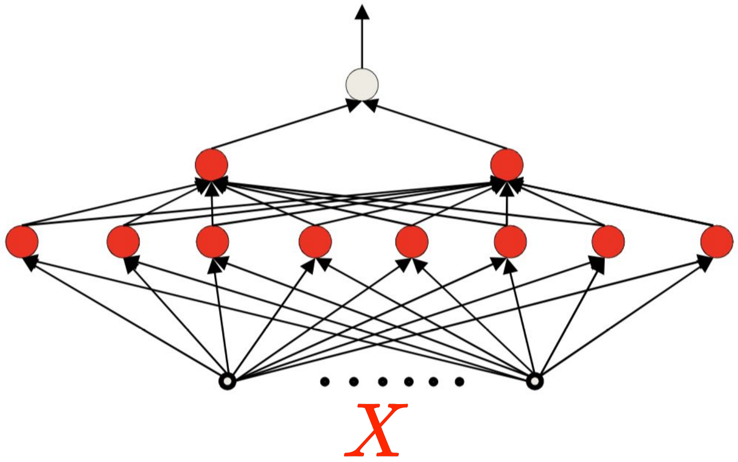
Is a neural network parameterized by W and takes input x

What is F?



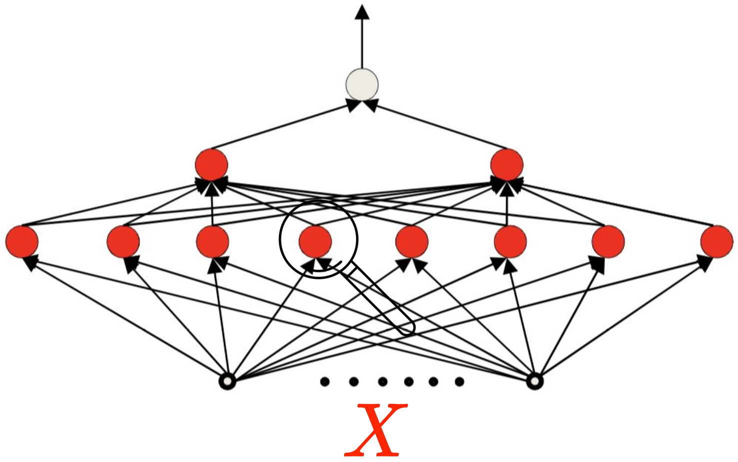
What is F?

$$Y = \mathcal{F}(X, W)$$

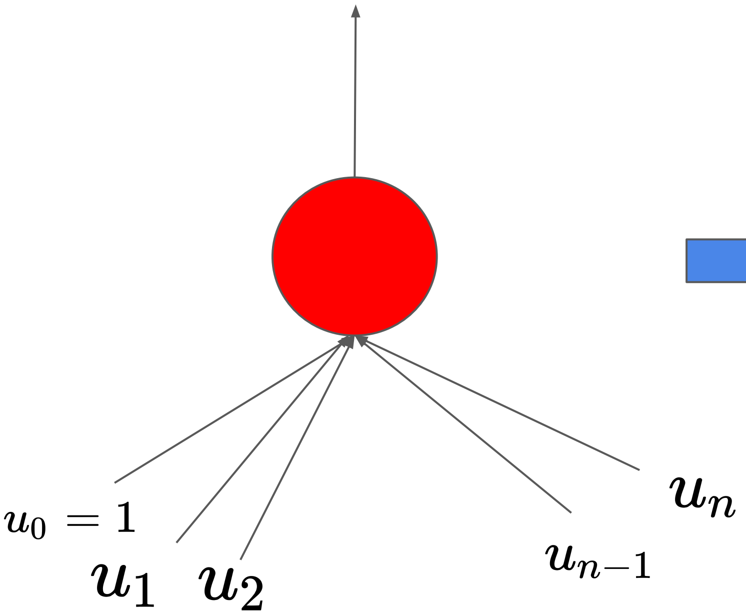


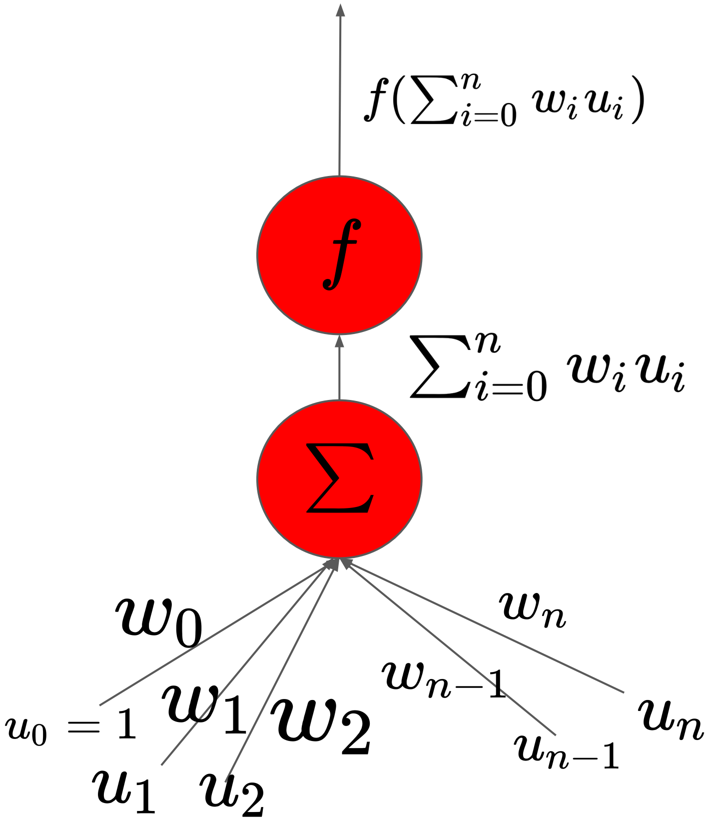
What is F?

$$Y = \mathcal{F}(X, W)$$

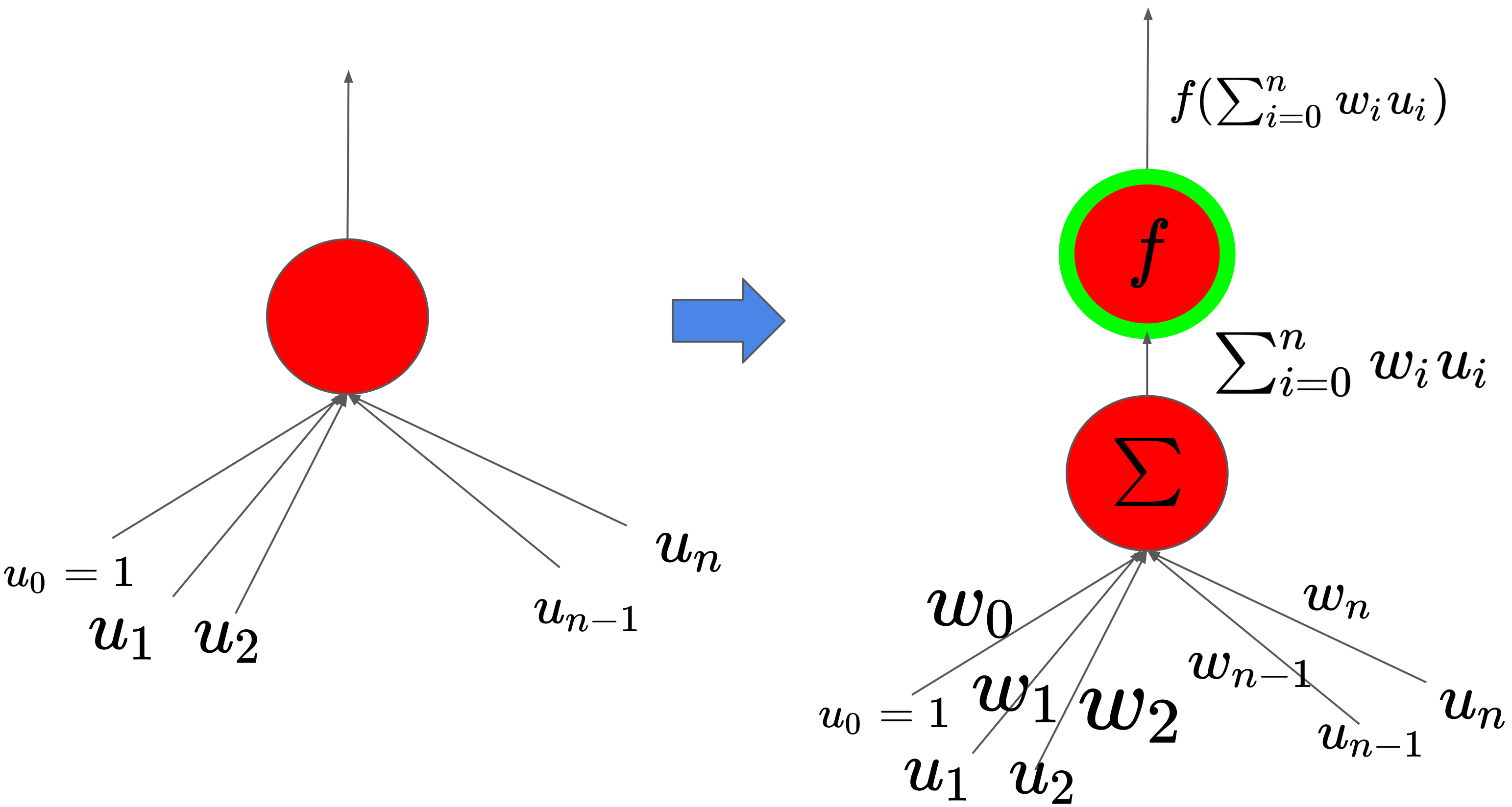


How does each neuron look?

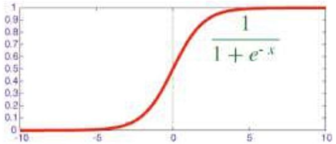




How does each neuron look?



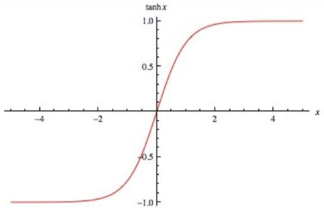
What is f ?



$$f(z) = \frac{1}{1+\exp(-z)}$$

$$f'(z) = f(z)(1 - f(z))$$

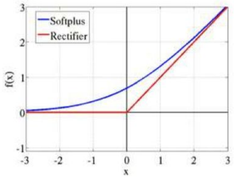
Sigmoid



$$f(z) = \tanh(z)$$

$$f'(z) = (1 - f^2(z))$$

tanh



$$f(z) = \begin{cases} 0, & z < 0 \\ z, & z \geq 0 \end{cases}$$

[*] $$f'(z) = \begin{cases} 1, z \geq 0 \\ 0, z < 0 \end{cases}$$

ReLU

$$f(z) = \log(1 + \exp(z))$$

$$f'(z) = \frac{1}{1+\exp(-z)}$$

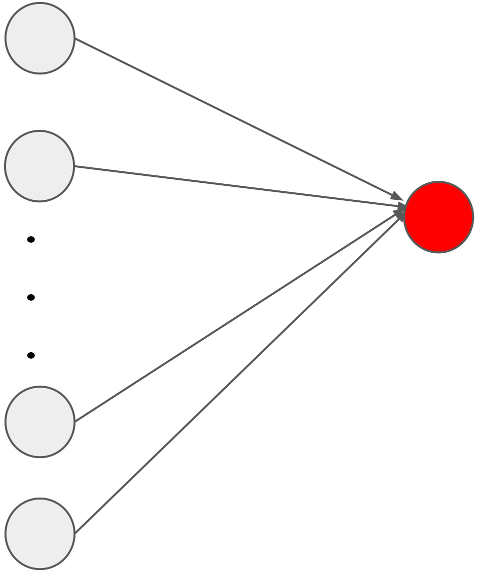
Log likelihood

Back to the problem ...

$$\mathcal{L}(W) = \mathcal{DIV}(\mathcal{F}(W, x), \boxed{\hat{y}})$$

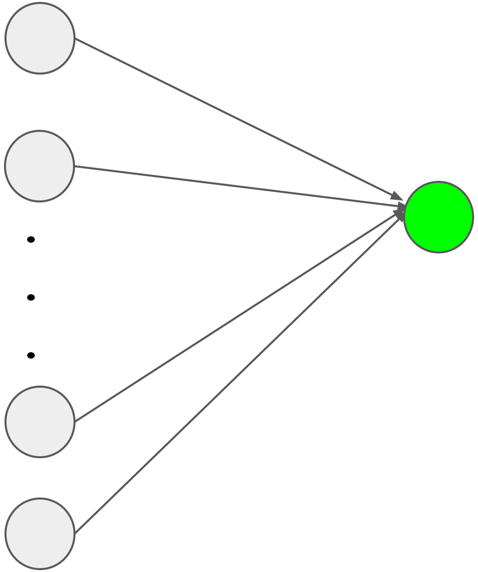
Groundtruth label

Example of $\hat{y}$: Cat vs dog Classifier



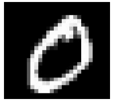
$$\hat{y} = 0$$

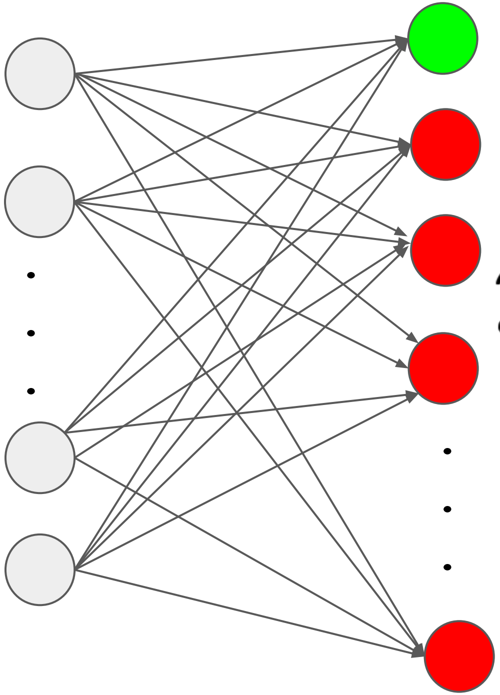
Example of $\hat{y}$: Cat vs dog Classifier



$$\hat{y} = 1$$

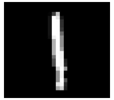
Example of $\hat{y}$: Digit classification

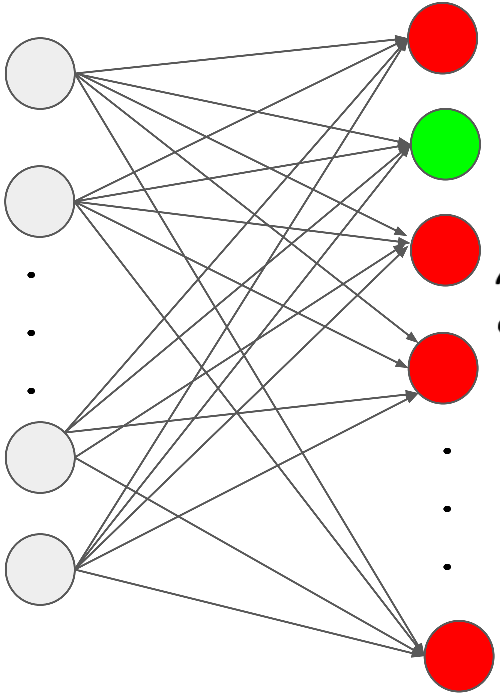




$$\hat{y} = [1, 0, 0, 0, \cdots, 0]$$

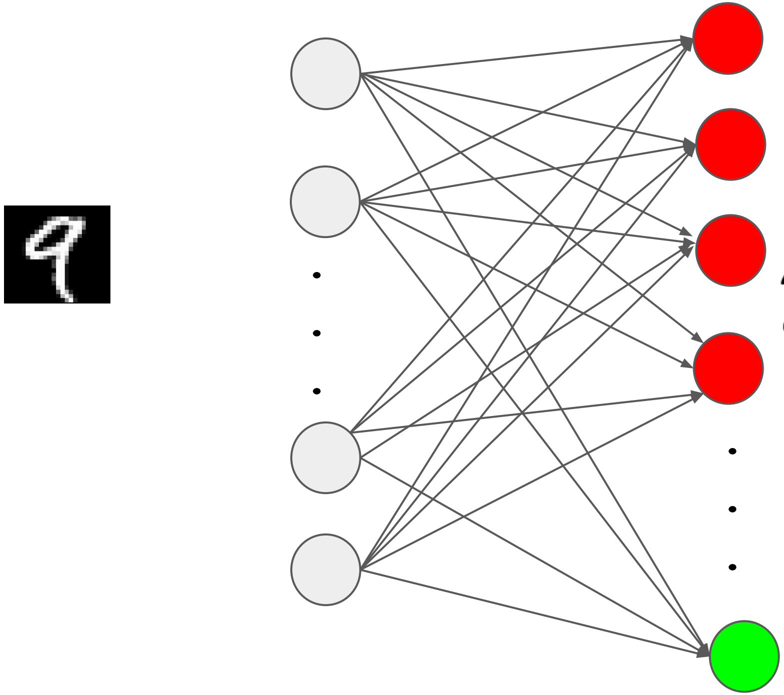
Example of $\hat{y}$: Digit classification





$$\hat{y} = [0, 1, 0, 0, \cdots, 0]$$

Example of $\hat{y}$: Digit classification



$$\hat{y} = [0, 0, 0, 0, \cdots, 1]$$

The problem

$$W^* = \arg\min_{W} \sum_{i=1}^{n} \mathcal{DIV}\big(\mathcal{F}(x_i, W), \hat{y}_i\big)$$

Back to high school ...

Let us understand?

1. What is a derivative?
2. What is gradient?
3. How to minimize a function?
4. What does direction of the gradient says?

On paper and pen!

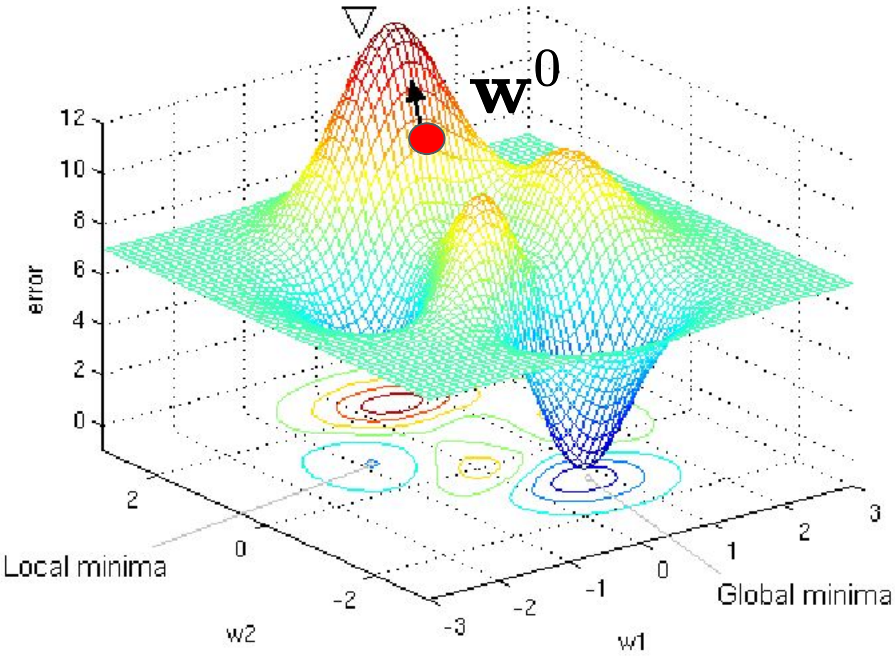
Gradient Descent

1. Initialize $iteration(t) \leftarrow 0, \eta, \mathbf{w}^0$
2. While $|f(\mathbf{w}^t) - f(\mathbf{w}^{t+1})| > \epsilon$

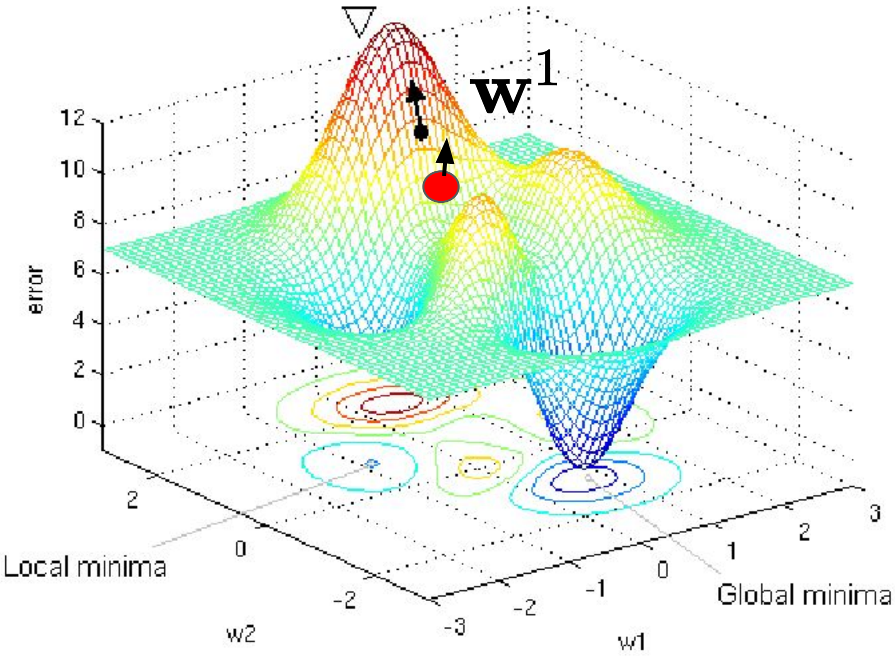
   (i) $\mathbf{w}^{t+1} = \mathbf{w}^t - \eta \nabla f(\mathbf{w^t})^T$

   (ii) $t = t + 1$

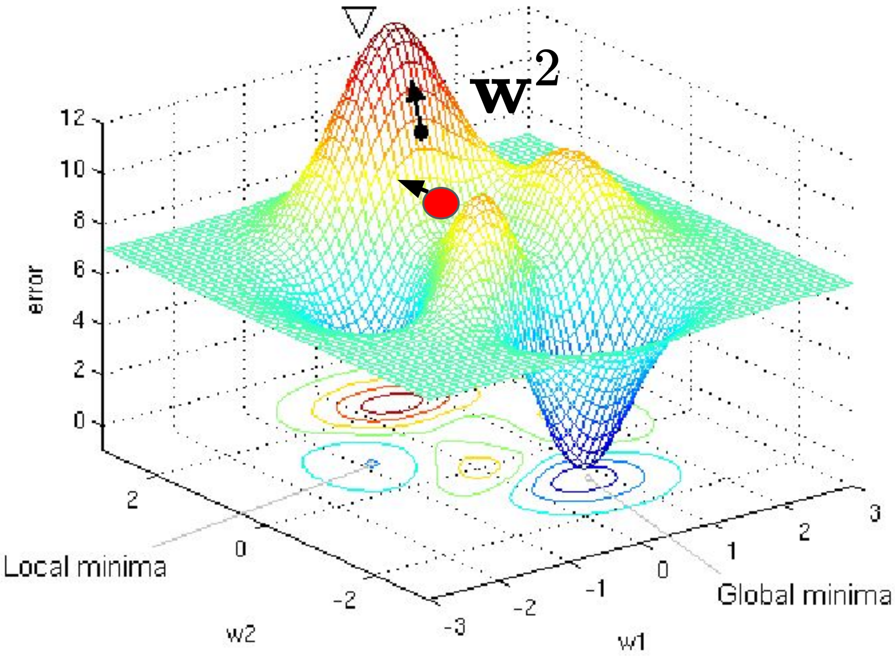
Gradient Descent in action



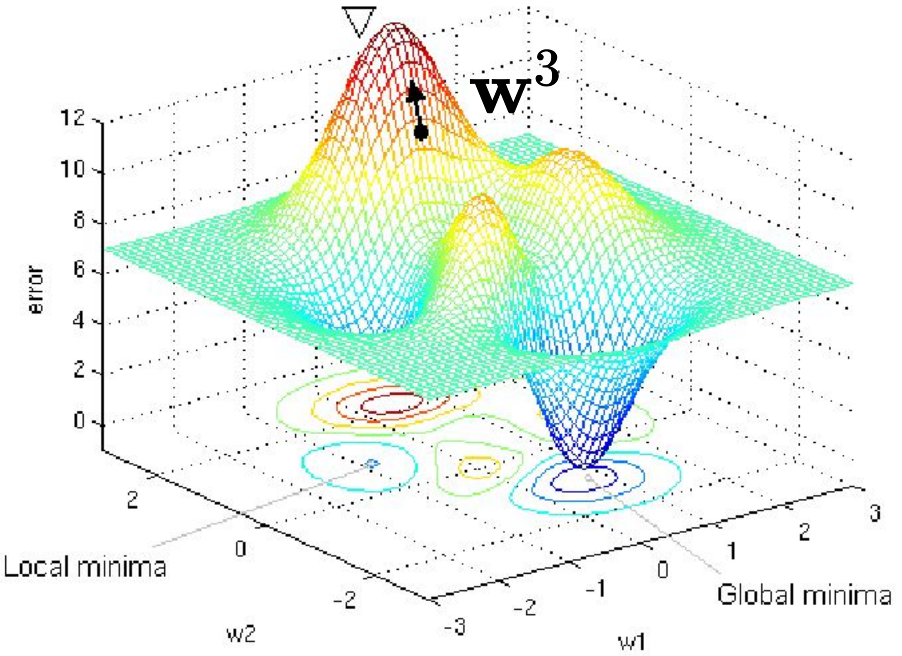
Gradient Descent in action



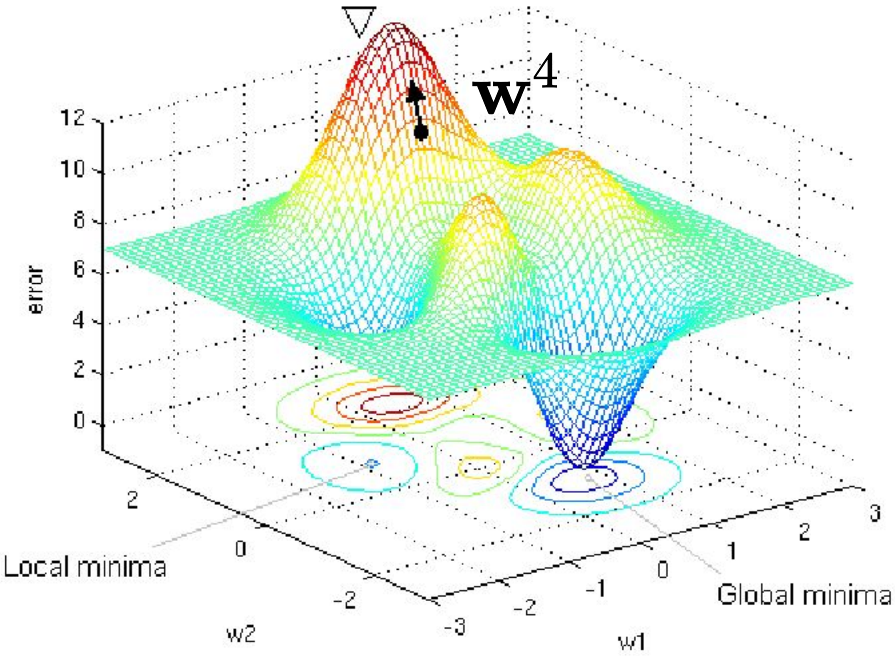
Gradient Descent in action



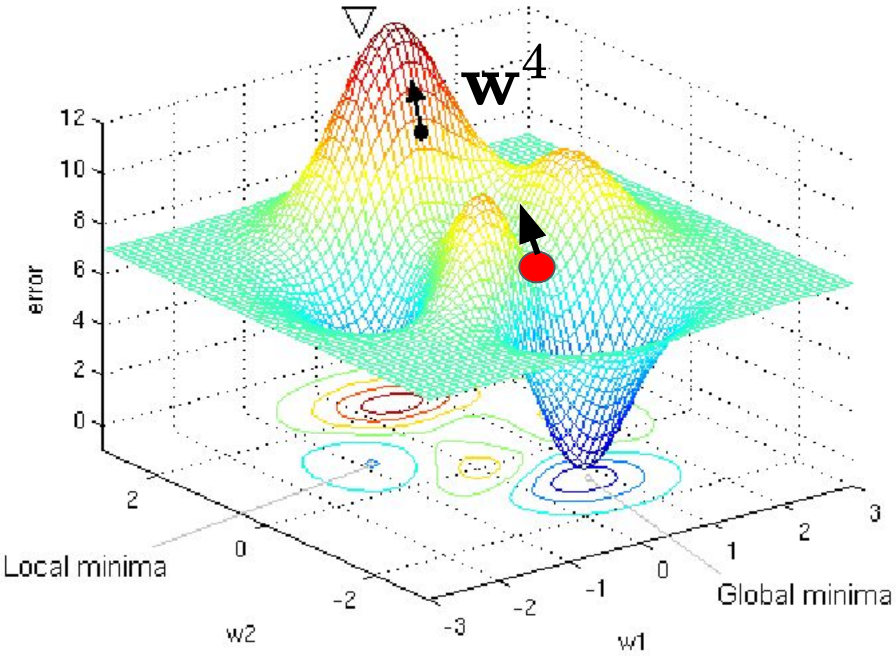
Gradient Descent in action



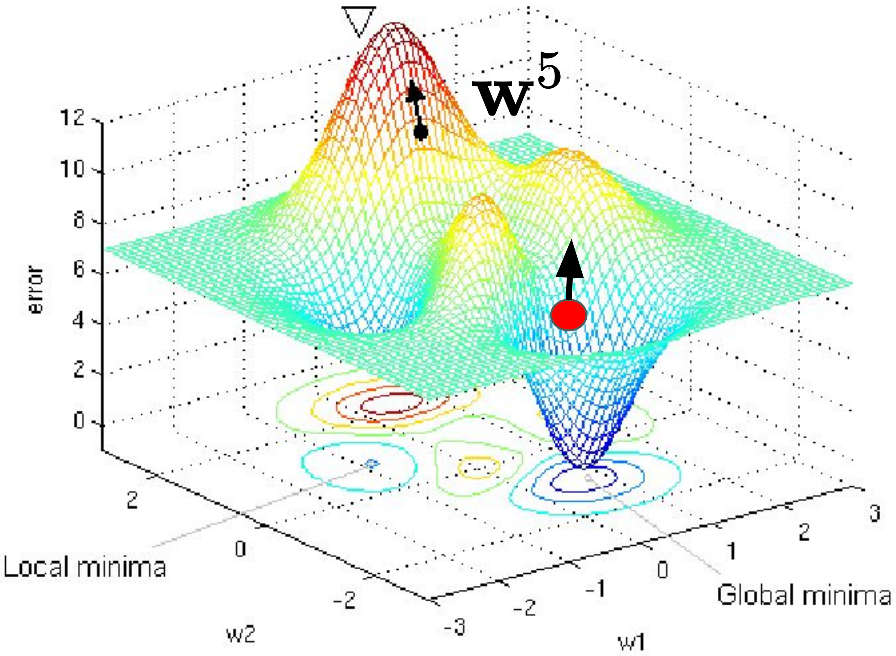
Gradient Descent in action



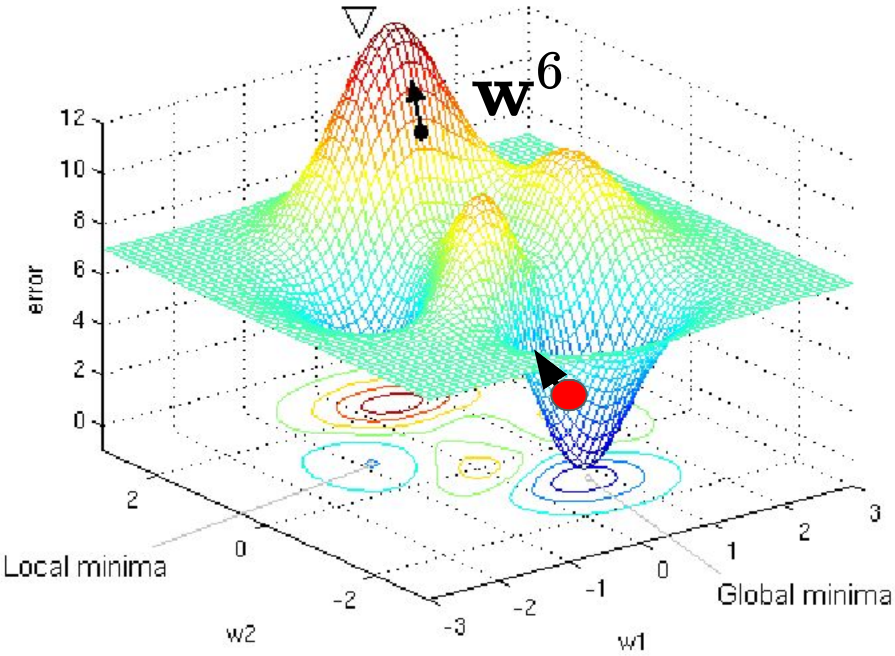
Gradient Descent in action



Gradient Descent in action



Gradient Descent in action



Gradient Descent in action