

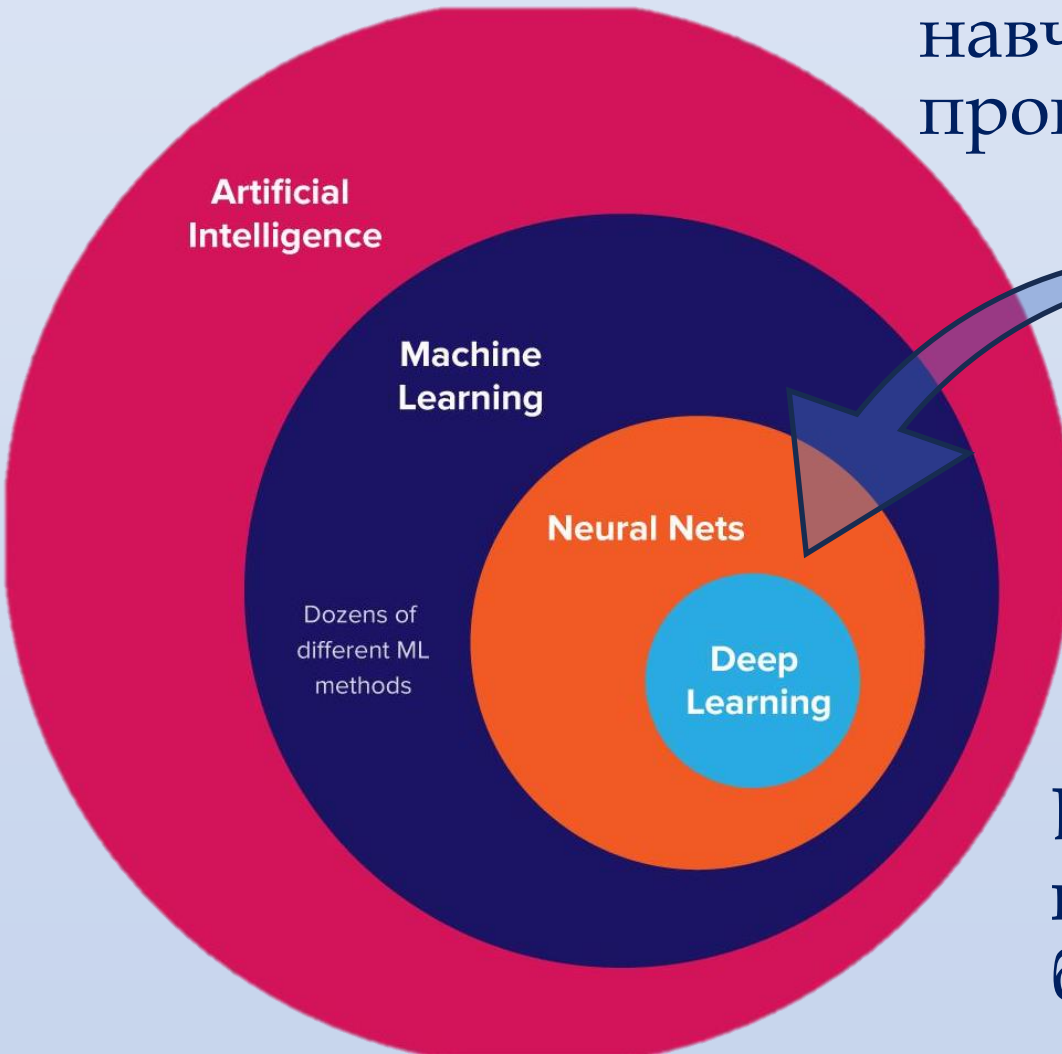
# **ОСНОВИ СИСТЕМ ШТУЧНОГО ІНТЕЛЕКТУ, НЕЙРОННИХ МЕРЕЖ та ГЛИБОКОГО НАВЧАННЯ**

## **Модуль 5. Глибоке навчання**

### **Лекція 5.2. Багатошаровий персептрон**

# Ареал ШІ

**ML – машинне навчання –**  
підрозділ ШІ, де системи  
навчаються без явного  
програмування



**NN – нейронна  
мережа –**  
математична модель,  
що імітує роботу  
людського мозку

**DL – глибоке  
навчання –** навчання  
багатошарових NN

# Багатошаровий персептрон (MLP)

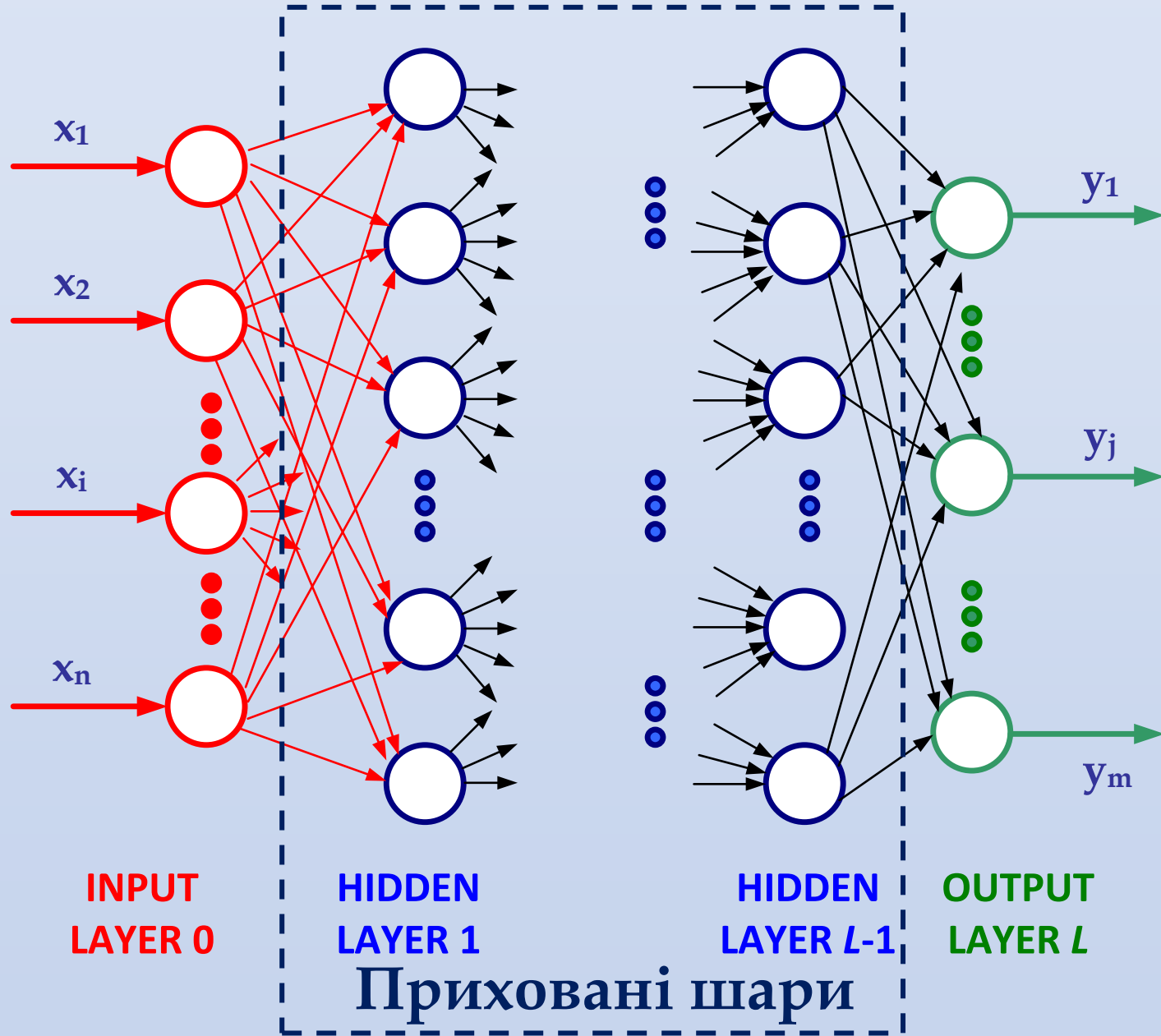
Багатошаровий персептрон (Multilayered perceptron) — повнозв'язаний клас штучної нейронної мережі із прямою передачею даних.

MLP складається щонайменше з трьох рівнів вузлів: вхідного, прихованого та вихідного. За винятком вхідних вузлів, кожен вузол є нейроном, який використовує деяку нелінійну функцію активації.

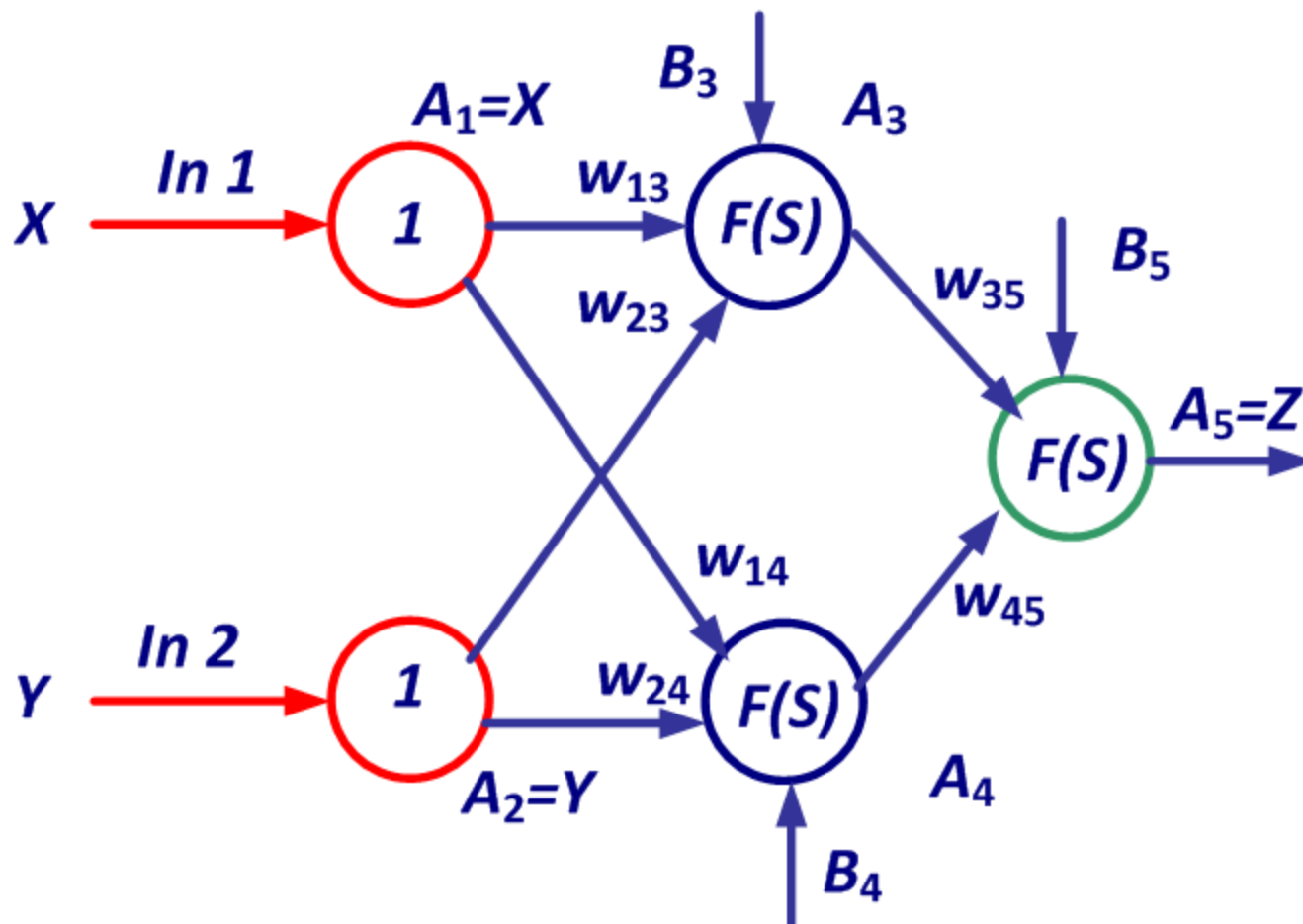
MLP використовує методику навчання з вчителем, з використанням зворотного поширення похибки для навчання.

Багатошаровість та нелінійна активація відрізняють MLP від лінійного персептрона.

# Багатошаровий персептрон (MLP)



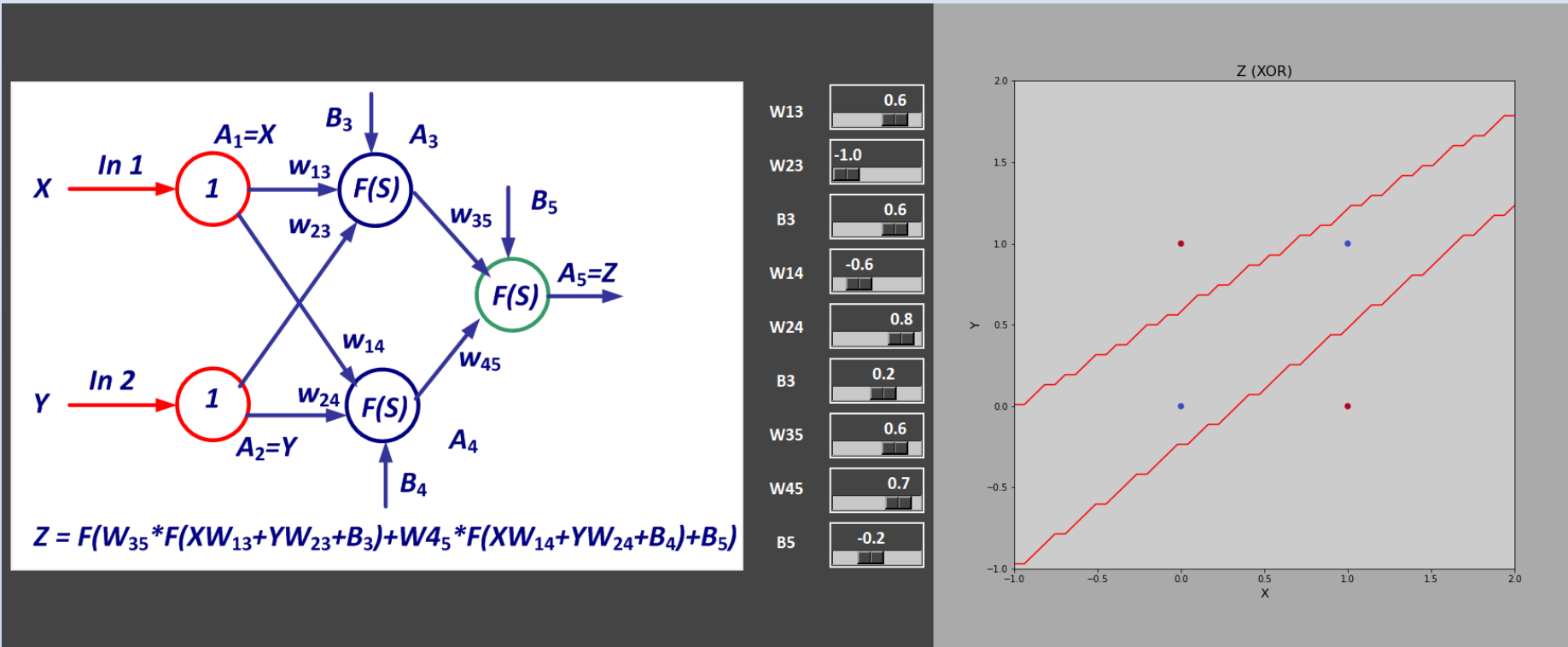
# Найпростіша багат шарова NN



$$Z = F(W_{35} * F(XW_{13} + YW_{23} + B_3) + W_{45} * F(XW_{14} + YW_{24} + B_4) + B_5)$$

# Найпростіша багат шарова NN

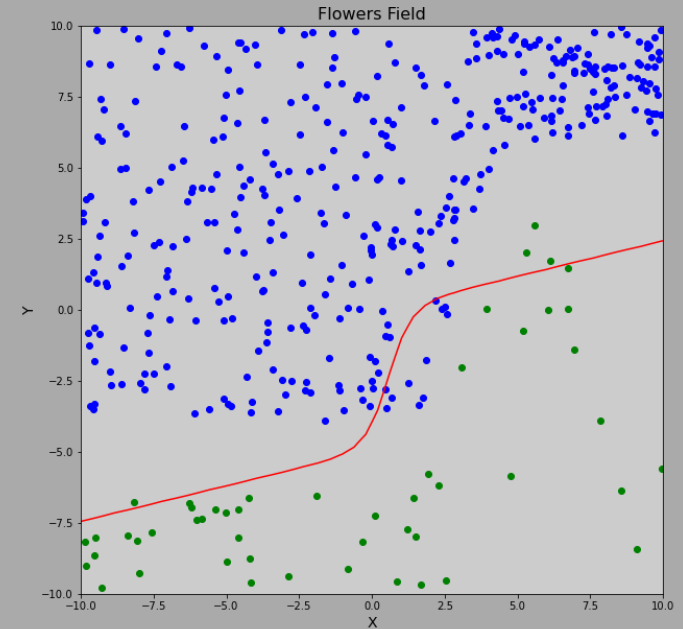
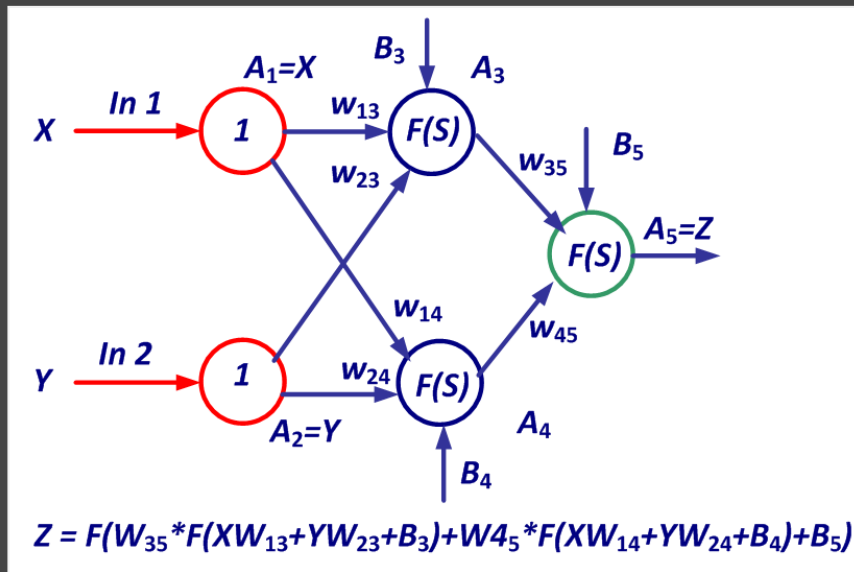
## Модель логічного елементу XOR



Дивись приклад 1 до лекції (скрипт Python)

# Найпростіша багат шарова NN

## Бінарний класифікатор



Дивись приклад 2 до лекції (скрипт Python)

# MLP. Формально

ШАРИ  $q = 0, 1, \dots, Q$

ШАР із номером  $q$ :

Складається з багатьох нейронів. Позначимо кількість нейронів в шарі  $q$  дорівнює  $N^{(q)}$ . Вхідний шар ( $q = 0$ ) має  $N^{(0)}$  і дорівнює розміру вхідного вектору ознак.

Номер нейрону в шарі  $q$  визначимо як  $i^{(q)}$ .

Попередній шар складається з  $N^{(q-1)}$  нейронів.

Таким чином кожен нейрон шару  $q$  має  $N^{(q-1)}$  входів. Номер входу означимо як  $j$  що змінюється від 0 до  $N^{(q-1)}$ .

Наступний шар складається з нейронів  $N^{(q+1)}$

Таким чином кожен нейрон шару  $q$  має  $N^{(q+1)}$  виходів.



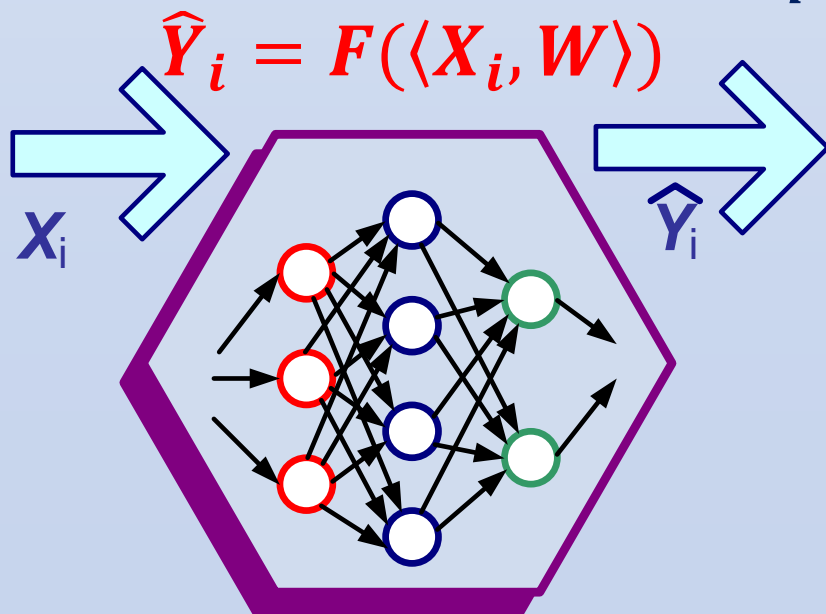
# MLP. Формально

Ваги нейрону  $i^{(q)}$  в шарі  $q$  означимо як  
 $w_{j^{(q-1)}, i^{(q)}}, q = 1, \dots, Q, ;$

$$i^{(q)} = 0, \dots, N^{(q)}; j^{(q)-1} = 0, \dots, N^{(q-1)},$$

Тобто загальна кількість ваг

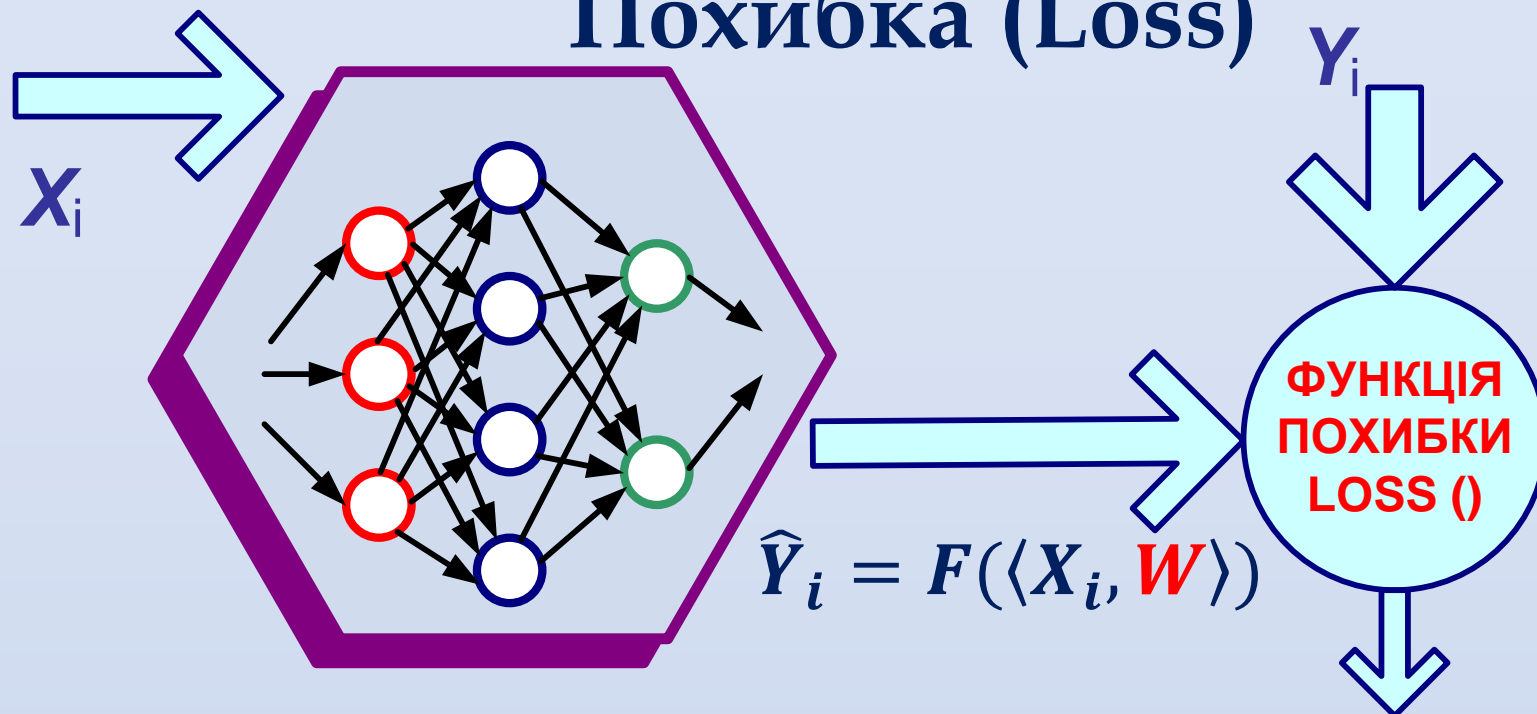
$$W \rightarrow \sum_{q=1}^Q N^{(q-1)} N^{(q)}$$



Тренувальні дані:  
множина  $\{X_i, Y_i\}$ ,

$\hat{Y}_i$  – обчислений  
вихід для  $i$ -го  
тренувального  
зразка.

# Похибка (Loss)



$Y_i$  – мітка, очікуваний вихід для  $i$ -го зразка.  
 $\hat{Y}_i$  – обчислений вихід для  $i$ -го тренувального зразка

$$L(\hat{Y}_i, Y_i) = L(F(\langle X_i, W \rangle), Y_i)$$

$L(W)$  – похибка, для заданих ознак та міток, залежність тільки від  $W$

**Завдання: знайти такі  $W$ , щоб похибка  $L(W)$  була мінімальною.**

# Функції похибки | Losses Functions

## Класифікація

- Log Loss( )
- Entropy Loss( )
- Focal Loss( )
- Expo Loss( )
- Hinge Loss( )

## Регрессія

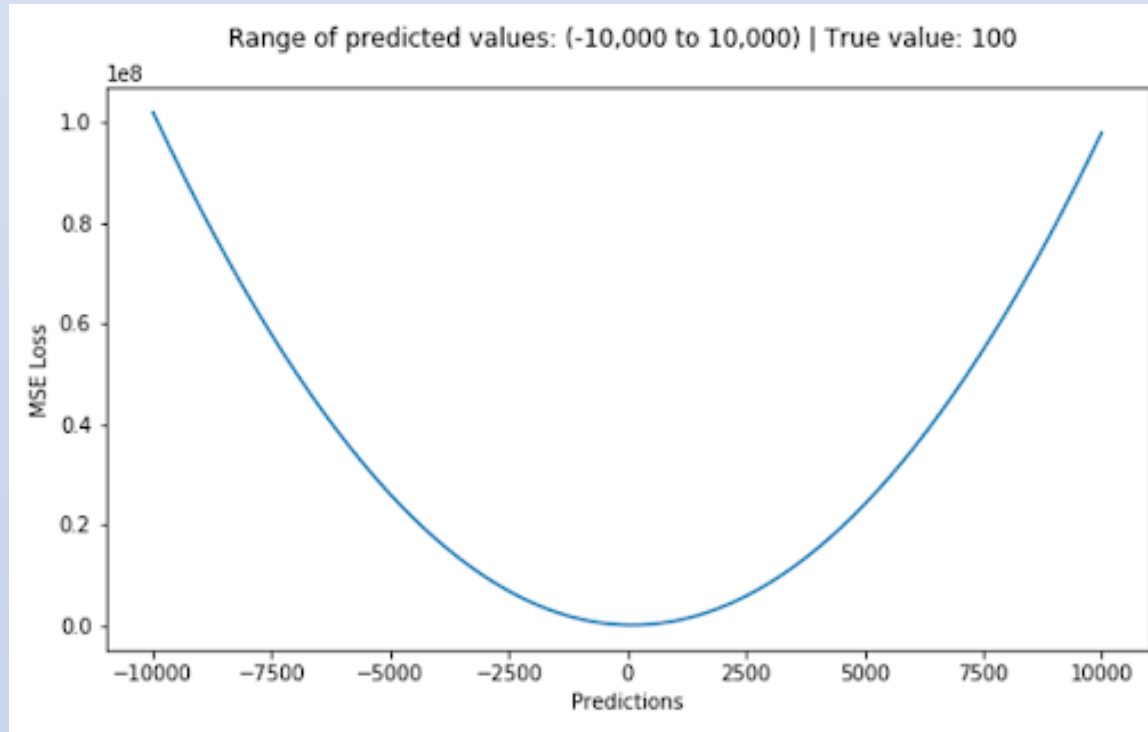
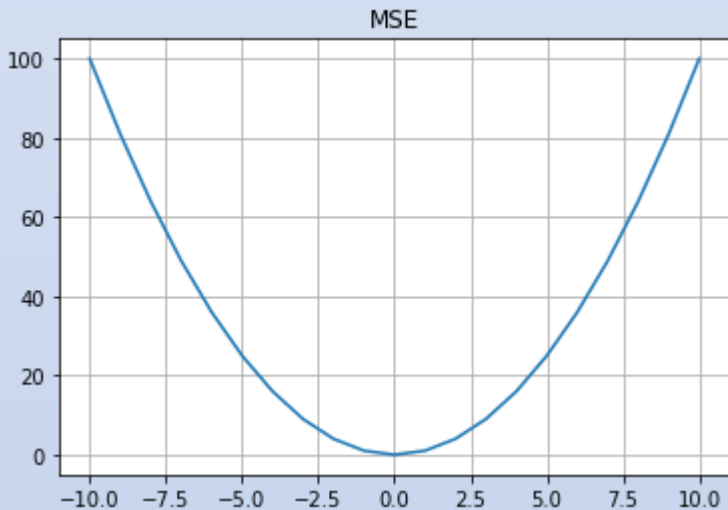
- MSE()
- MAE( )
- Huber Loss( )
- Log Loss( )
- Quantile Loss( )

# Функції похибки | Losses Functions

## Regression Losses

### Mean Square Error (MSE)/Quadratic Loss/L2 Loss

$$MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - \widehat{Y}_i)^2$$



# Функції похибки | Losses Functions

## Regression Losses

### Root Mean Square Error (RMSE)

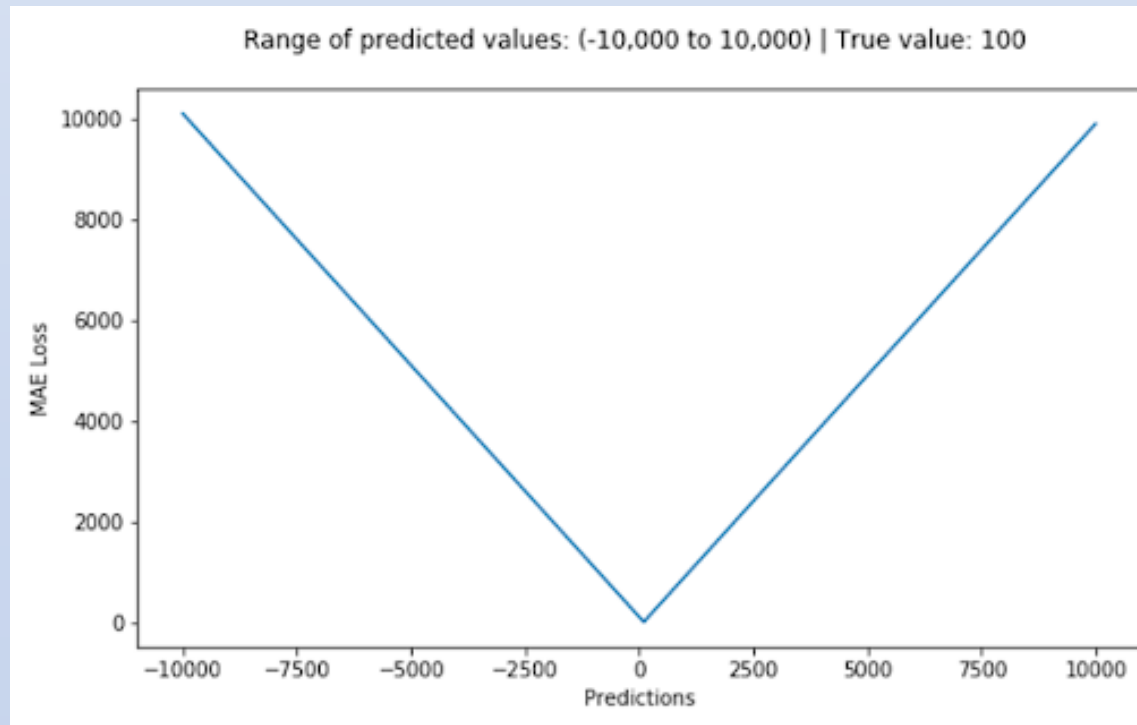
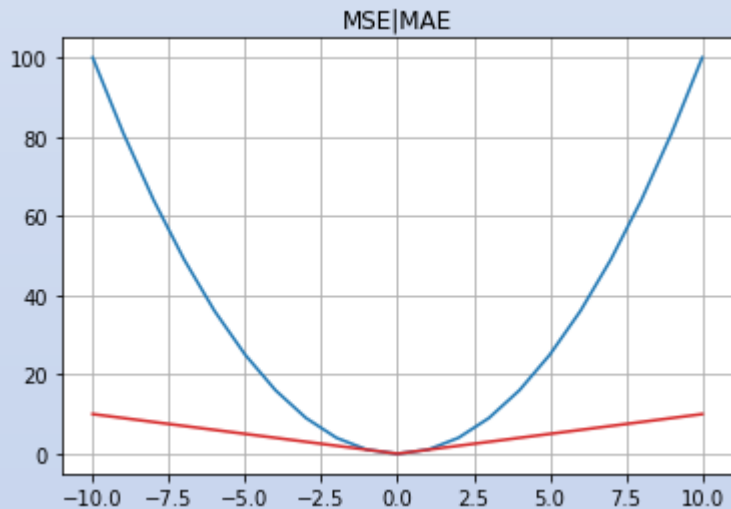
$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (Y_i - \widehat{Y}_i)^2}$$

# Функції похибки | Losses Functions

## Regression Losses

### Mean Absolute Error (MAE) / L1 Loss

$$MAE = \frac{1}{n} \sum_{i=1}^n |Y_i - \hat{Y}_i|$$

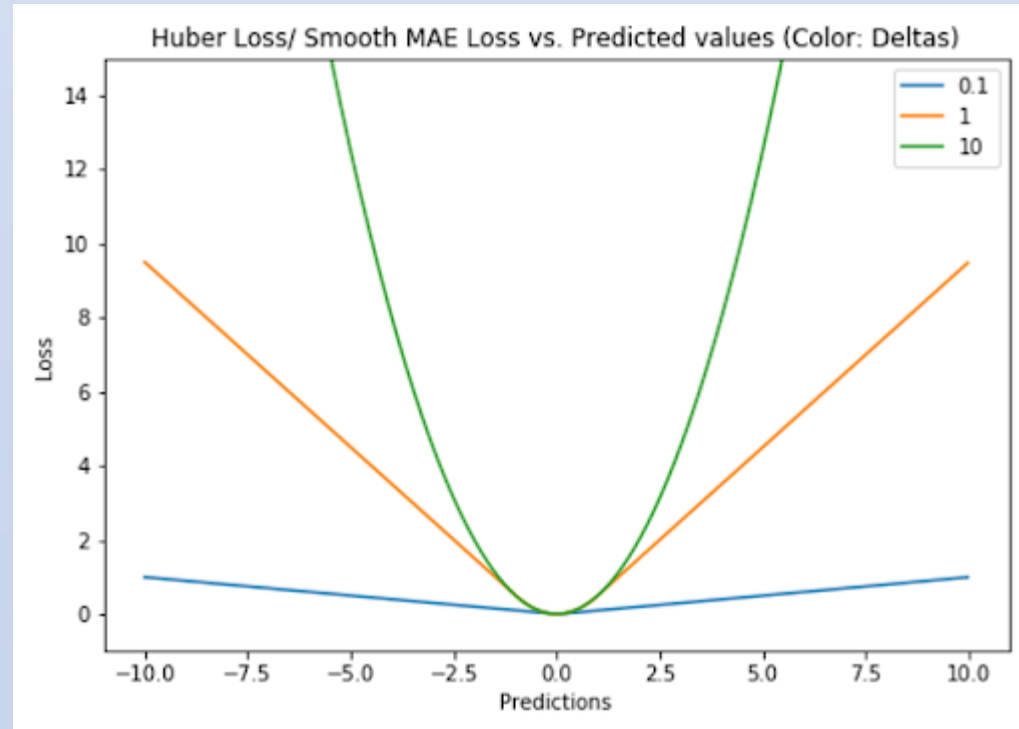
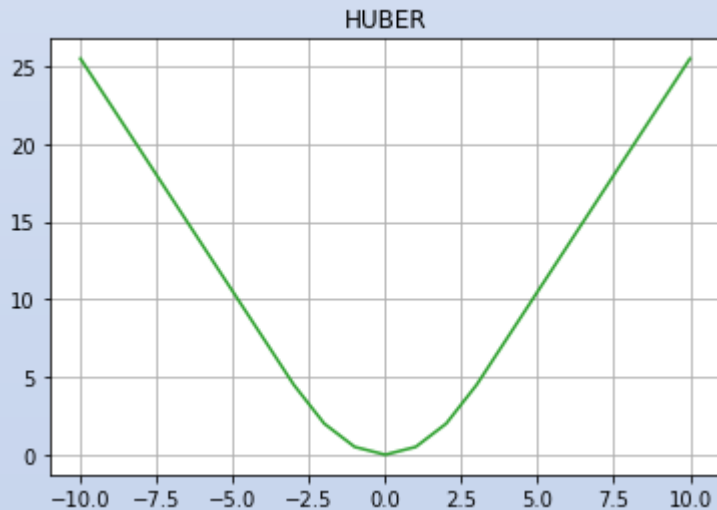


# Функції похибки | Losses Functions

Huber Loss / Smooth Mean Absolute Error / Soft MAE (Комбінація L2 та L1)

$$a = \frac{1}{n} \sum_{i=1}^n (Y_i - \widehat{Y}_i)$$

$$L_{\delta}(a) = \begin{cases} 0.5 * a^2, & |a| \leq \delta \\ \delta * (|a| - 0.5)\delta, & |a| > \delta \end{cases}$$

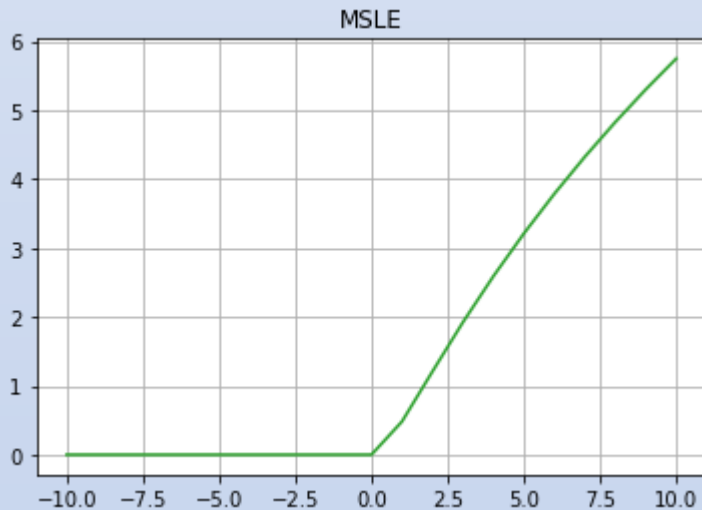


# Функції похибки | Losses Functions

## Classification Losses

### Mean Squared Logarithmic Error (MSLE)

$$MSLE = \frac{1}{n} \sum_{i=1}^n (\log(Y_i) - \log(\hat{Y}_i))^2$$





# Функції похибки | Losses Functions

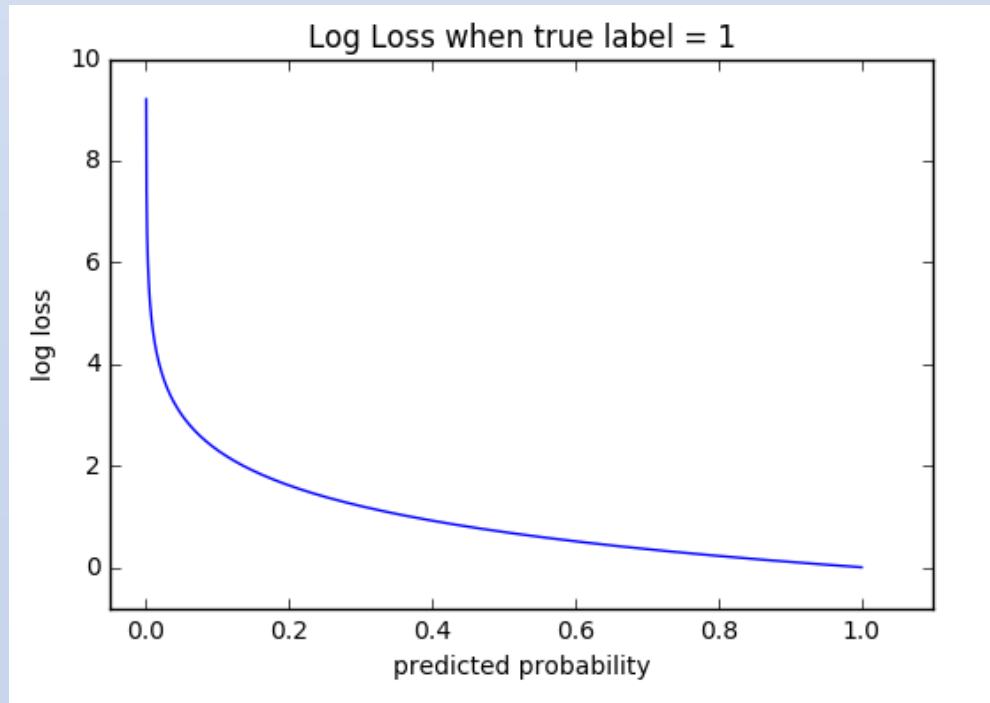
## Classification Losses

Cross entropy loss (logarithmic loss, log loss, logistic loss).

***Loss*** =

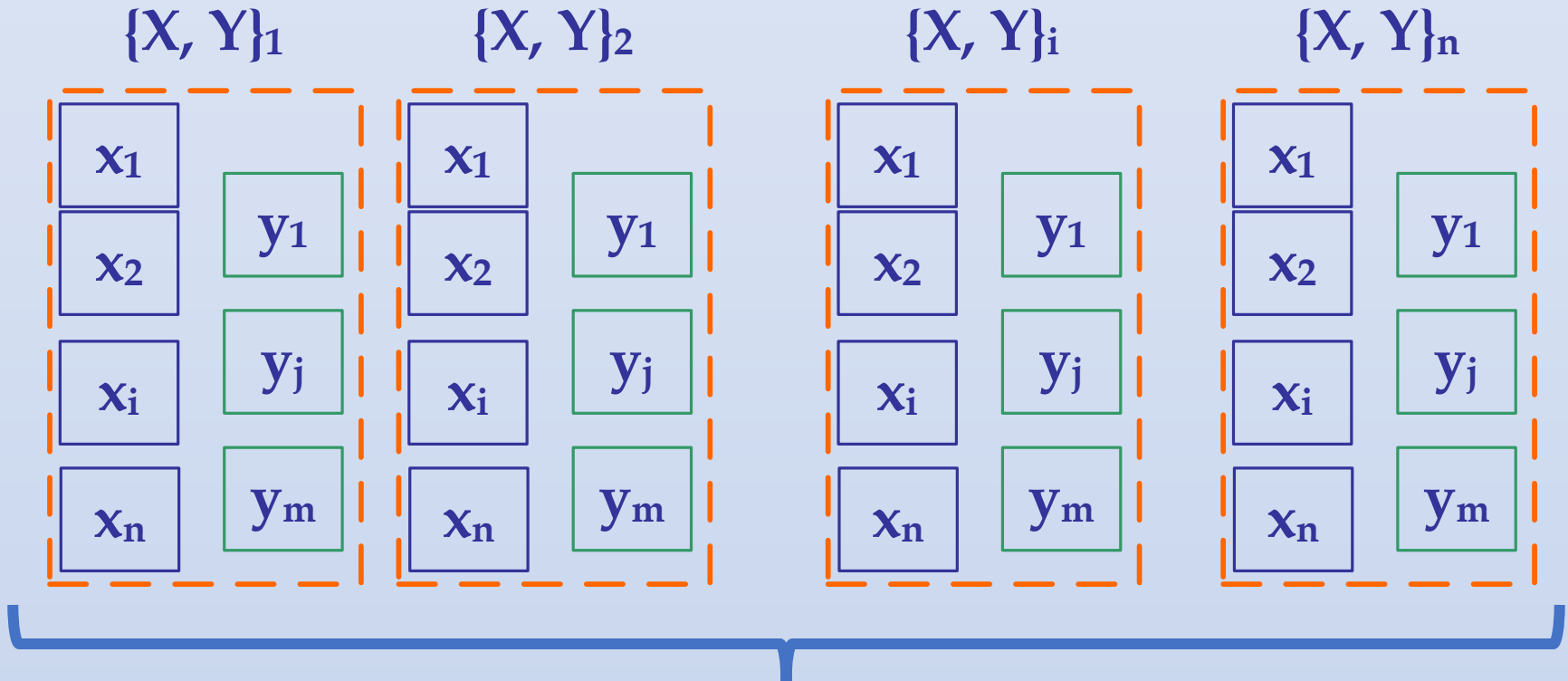
$$-\frac{1}{n} \sum_{i=1}^n [Y_i \log(\hat{Y}_i) + (1 - Y_i) \log(1 - \hat{Y}_i)]^2$$

Випадок  
бінарної  
класифікації  
Тут  $\hat{Y}$  -  
ймовірність  
(вихід softmax())



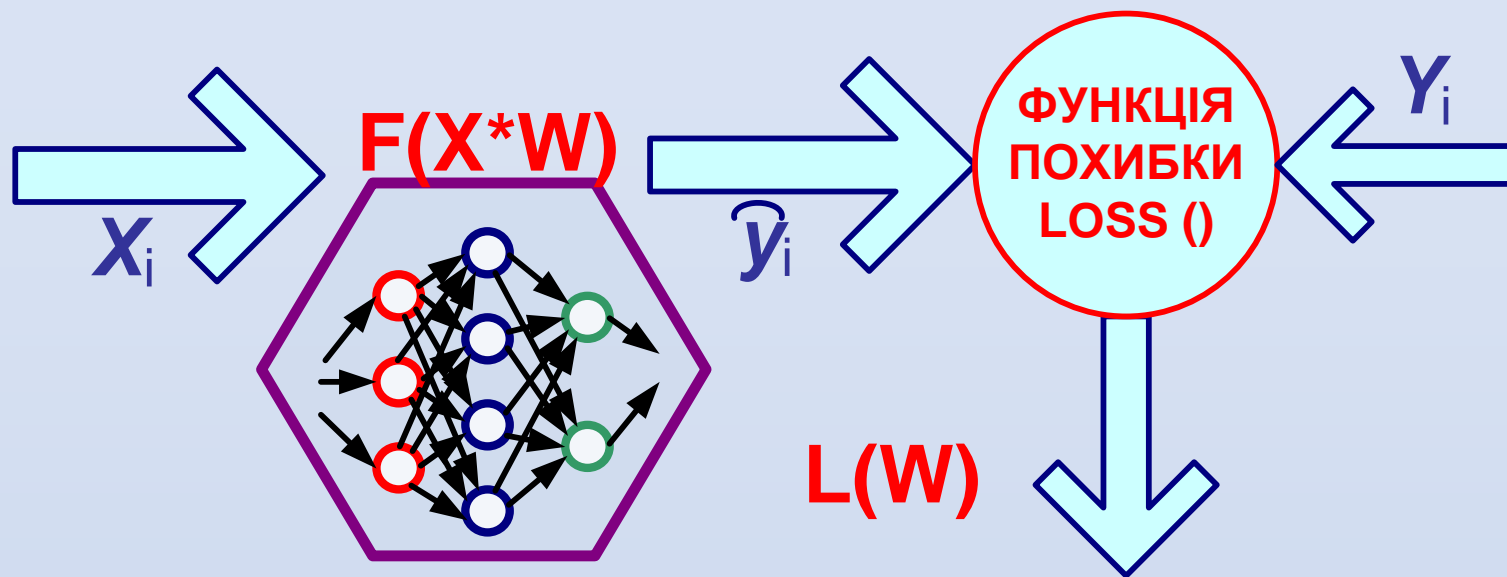
# Тренувальні дані

Визначена множина  $\{X_i, Y_i\}, \quad i = 1, 2, \dots, N$



Тренувальна множина екземплярів

# Навчання



$y_i$  – мітка (очікуваний вихід) для  $i$ -го екземпляру.

$\hat{y}_i$  – обчислений вихід для  $i$ -го екземпляру.

$L(W)$  – похибка, для заданих ознак та міток, як залежність від  $W$

**ЗНАЙТИ  $W$  , щоб похибка  $L(W)$  була мінімальною (бажано 0).**

# Загальний підхід до навчання

```
W ← 0 ; b ← 0 - ініціалізація
for epoche = 1 to max_epoche:
  for all [X, Y]:
    s ← W*X+b
    a ← F(s)
     $\hat{Y}$  ← softmax(a)
    L ← cross entropy (Y,  $\hat{Y}$ )
    GrW, Grb ←  $\partial L / \partial W$  ;  $\partial L / \partial b$ 
    W ← W - lean_rate * GrW
    b ← b - lean_rate * Grb
  end for
end for
```

# MLP. Позначення

ШАРИ  $q = 0, 1, \dots, Q$

Кількість нейронів в шарі  $q \rightarrow N^{(q)}$ .

Вхідний шар ( $q = 0$ ) має  $N^{(0)}$  нейронів і дорівнює розміру вхідного вектору ознак  $n$ .

Номер нейрону в шарі  $q \rightarrow i^{(q)}, i^{(q)} = 1, 2, \dots, N^{(q)}$ .

**ПОВНОЗВ'ЯЗНОСТЬ:**

Кожен  $i^{(q)}$  нейрон шару  $q$  має  $N^{(q-1)}$  входів.

Кожен  $i^{(q)}$  нейрон шару  $q$  має  $N^{(q+1)}$  виходів.

Вихід  $i^{(q)}$  нейрону  $\rightarrow a_{i^{(q)}}$

Ваги нейрону  $i^{(q)}$  в шарі  $q$

$$W_{j^{(q-1)}, i^{(q)}} ,$$

$$i^{(q)} = 1, \dots, N^{(q)}; j^{(q-1)} = 1, \dots, N^{(q-1)},$$

# Загальний алгоритм навчання

```
W ← 0 ; b ← 0 #ініціалізація  
for epoche = 1 to max_epoche:  
    for all [X, Y]:
```

```
        #пряме розповсюдження  
         $\hat{Y} \leftarrow F(W * X + b)$  #по всіх шарах та нейронах  
        #накопичення похибки  
         $Lsum(W) += Loss(Y, \hat{Y})$ 
```

```
    #похибка  
     $L(W) = Lsum(W) / N$   
     $L(W)$  мале ? → breack
```

```
    #зворотне розповсюдження  
     $GrW, Grb \leftarrow \partial L / \partial W ; \partial L / \partial b$   
    # градієнтна мінімізація похибки  
     $W \leftarrow W - learning\_rate * GrW$   
     $b \leftarrow b - learning\_rate * Grb$ 
```

# Загальний алгоритм навчання ПРЯМЕ РОЗПОВСЮДЖЕННЯ

```
# Для кожного екземпляру
for  $q = 0$ :
     $a_{j^{(0)}} = x_j$  #вихід шару 0
for  $q = 1, 2, \dots, Q$ : #попередньо для всіх шарів

    for  $i^{(q)} = 1, 2, \dots, N^{(q)}$  :
         $z_{i^{(q)}} = 0$ 
        for all  $j^{(q-1)} = 1, \dots, N^{(q-1)}$ :
             $z_{i^{(q)}} += a_{j^{(q-1)}} w_{j^{(q-1)}, i^{(q)}}$ 
         $a_{i^{(q)}} = F(z_{i^{(q)}})$ 

 $\hat{Y} = a_{i^{(Q)}}$  #вихід останнього шару - передбачення
#накопичення похибки
 $Lsum(W) += Loss(Y, \hat{Y})$ 
```

# Загальний алгоритм навчання

## ПОХИБКА

*#Осереднюється сума похибок для кожного екземпляру датасету*

$$L(W) = Lsum(W) / N$$

*#Якщо похибка менше деякого визначеного значення (tolerance) обчислення завершуються*

*If  $L(W) \leq tolerance$ :  
    breck*

За правило, попереднє визначається максимальні кількість епох  $i$ , якщо для цієї епохи похибка не досягає *tolerance*, обчислення також завершуються



# Загальний алгоритм навчання ЗВОРОТНЕ РОЗПОВСЮДЖЕННЯ

#Для кожного  $w_{j^{(q-1)},i^{(q)}}$  визначаються компоненти градієнту

$$\nabla L_{j^{(q-1)},i^{(q)}} = \frac{\partial L(W)}{\partial w_{j^{(q-1)},i^{(q)}}}$$

# кожна вага змінюється

$$w_{j^{(q-1)},i^{(q)}}^{(eph+1)} = w_{j^{(q-1)},i^{(q)}}^{(eph)} - learning\_rate * \nabla L_{j^{(q-1)},i^{(q)}}$$

Оновлені значення ваг використовуються для прямого розповсюдження на наступній епосі.

## **Рекомендована ЛІТЕРАТУРА**

**Литвин В. В., Пелешак Р. М., Висоцька В. А.**  
Глибинне навчання : навч. посіб. – Львів : Вид-во  
Львівської Політехніки, 2011. – 264 с.

**Тимощук П.В., Лобур М. В.** Principles of Artificial  
Neural Networks and Their Applications :: Принципи  
штучних нейронних мереж та їх застосування : навч.  
посіб. – Львів : Вид-во Львівської Політехніки, 2011. –  
292 с.

**Тимощук, П.В.** Штучні нейронні мережі : навч.  
посіб. – Львів : Вид-во Львівської Політехніки, 2011. –  
444 с.

## Рекомендована ЛІТЕРАТУРА

**Beyeler M.** Machine Learning for OpenCV . — Packt Publishing Ltd., 2017 . — 350 p.

**Sarkar D., Bali R., Sharma T.** Practical Machine Learning with Python . — APress, 2018. — 530p.

**Raschka S., Mirjalili V.** Python Machine Learning. Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow 2 .- 3rd Edition, Packt Publishing, 2019 .- 859 p.

**The END**

**Модуль 5. Лекція 02.**