ОСНОВИ СИСТЕМ ШТУЧНОГО ІНТЕЛЕКТУ, НЕЙРОННИХ МЕРЕЖ та ГЛИБОКОГО НАВЧАННЯ

Модуль 5. Глибоке навчання

Лекція 5.4. Найпростіша MLP вирішення задач класифікації

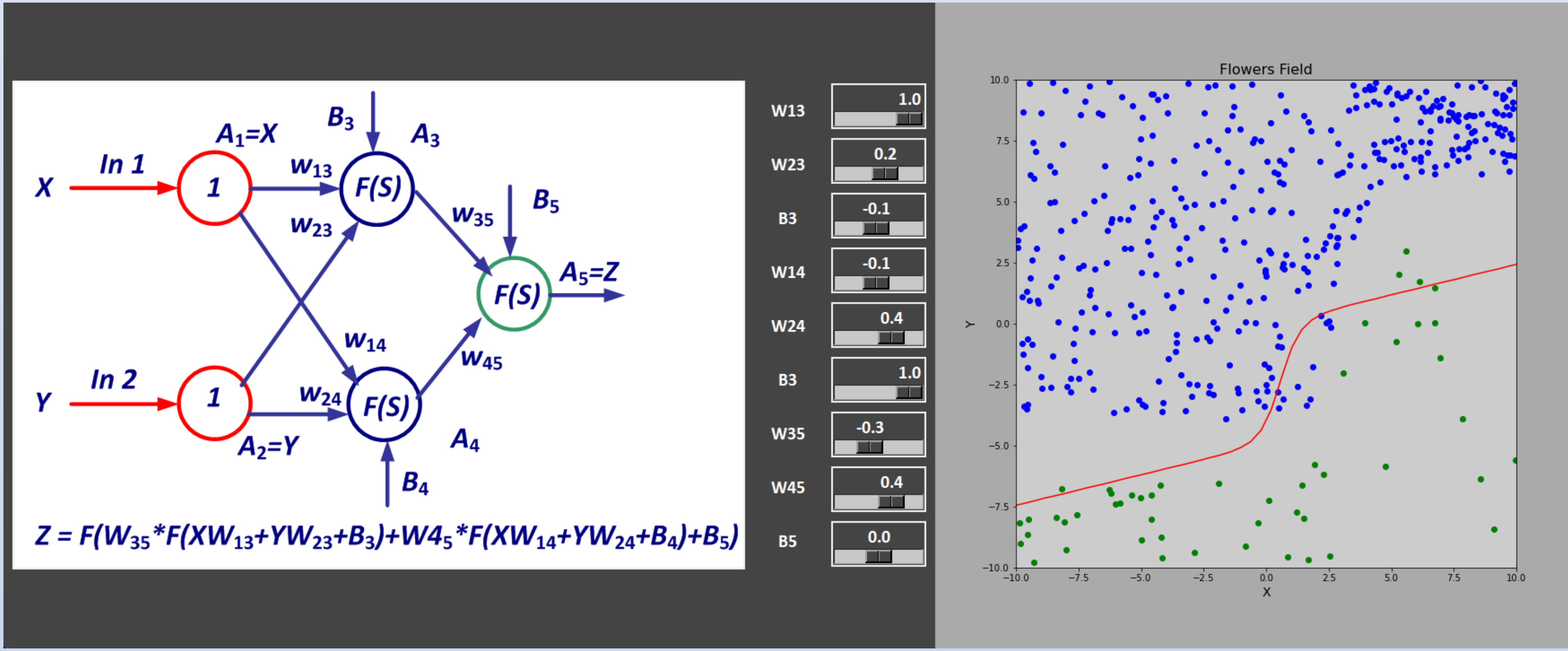
Бінарна класифікація

Маємо множину $\mathbb{O}$ об'єктів $o^{(j)}$, $j = 1,2,\ldots,M$.
Маємо два класи $\mathrm{c}^{(0)}$, $\mathrm{c}^{(1)}$, $\mathrm{K} = 2$.
Припустимо, що кожен об'єкт $o^{(j)}$ має тільки одну ознаку $x^{(j)} \in \mathbb{R}$, множини $\mathbb{X}$.

Необхідно знайти $\boldsymbol{F}(\boldsymbol{x})$, натреновану на множені $\mathbb{O}$, що повертає 0 або 1 з дякою (мінімальною) похибкою.

Бінарна класифікація

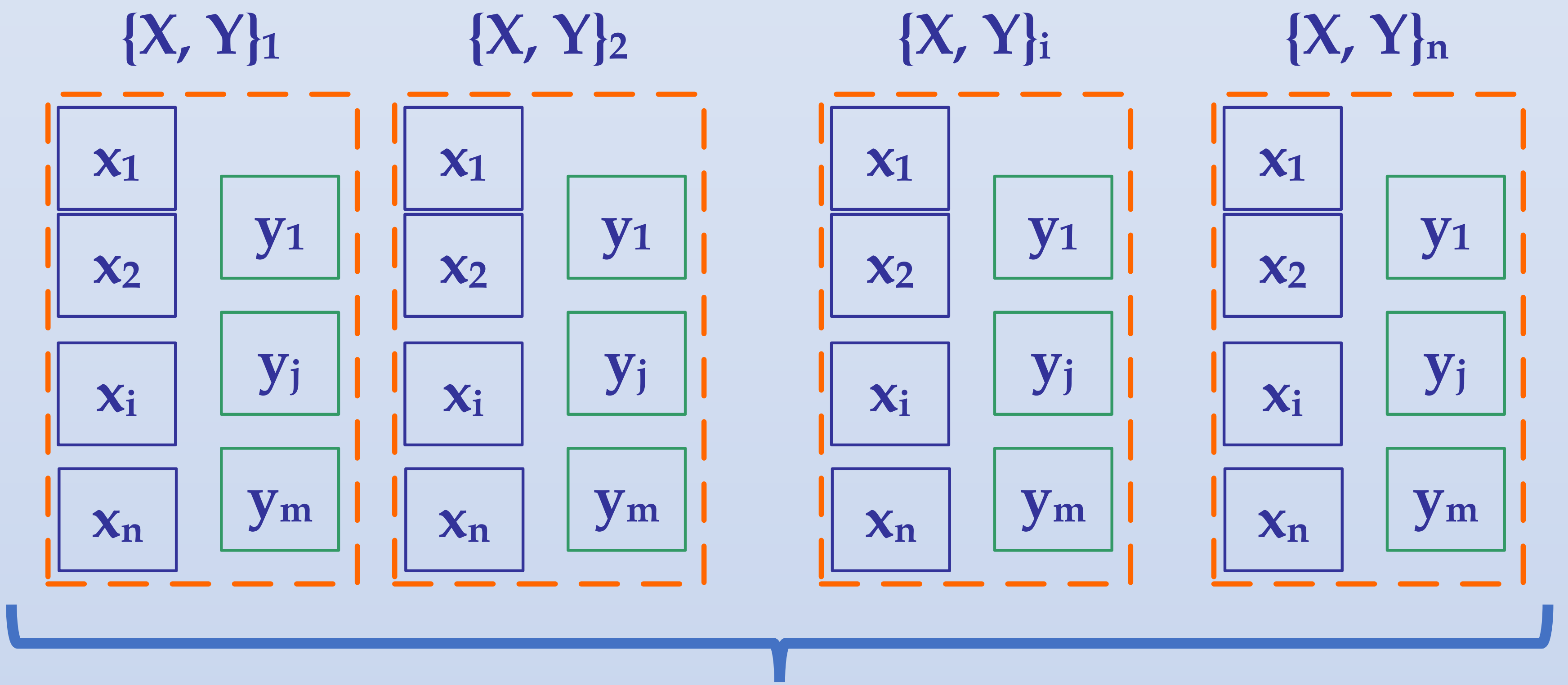
Бінарний класифікатор



Дивись **приклад 2 до лекції 2 (скрипт Python)**

Тренувальні дані

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



Тренувальна множина екземплярів

Стандартний (GD) градієнтний спуск

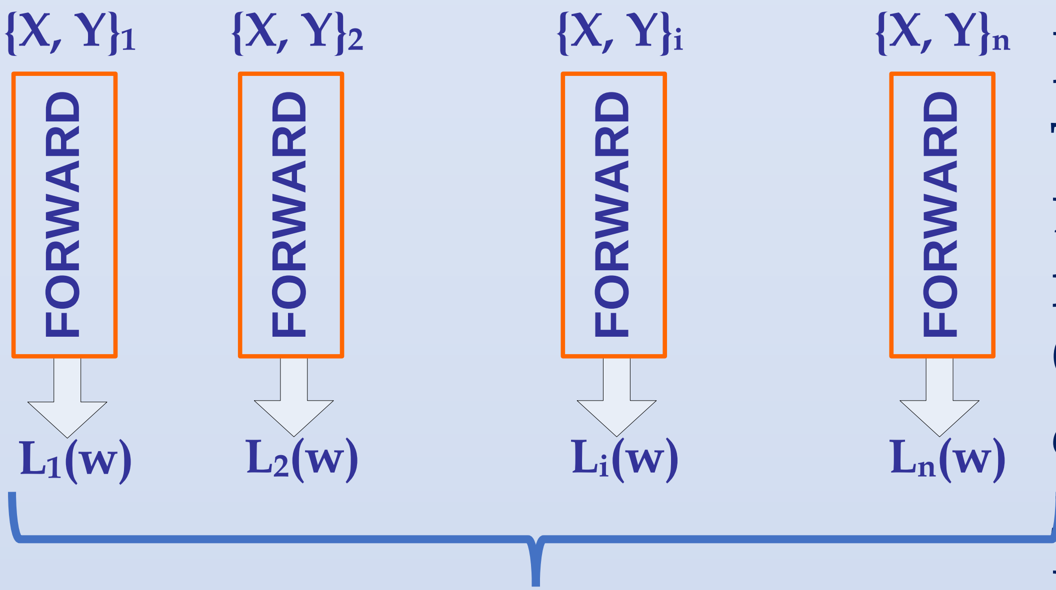
$$\nabla L_{ac}(w) = \sum \nabla L_i(w) \text{ – накопичення змін}$$

$$\nabla L(w) = \frac{\nabla L_{ac}(w)}{n}$$

Розділяються накопичувальні змінні ваг на кількість навчальних прикладів. Це дає середні градієнти для всіх ваг. Далі оновлюються всі ваги

Цей процес повторюється від початку до кінця протягом певної кількості ітерацій.
Це означає, що для ОДНІЄЇ ітерації GD повторюються всі навчальні приклади, обчислюються градієнти, а потім оновлюються ваги.
Виконується певна кількість ітерацій GD.

Градієнтний спуск (GD, Batch-GD)



Послідовно по тренувальним векторам (i = 1, 2, … n) : Обчислюється функція втрат Loss для i-го навчального прикладу

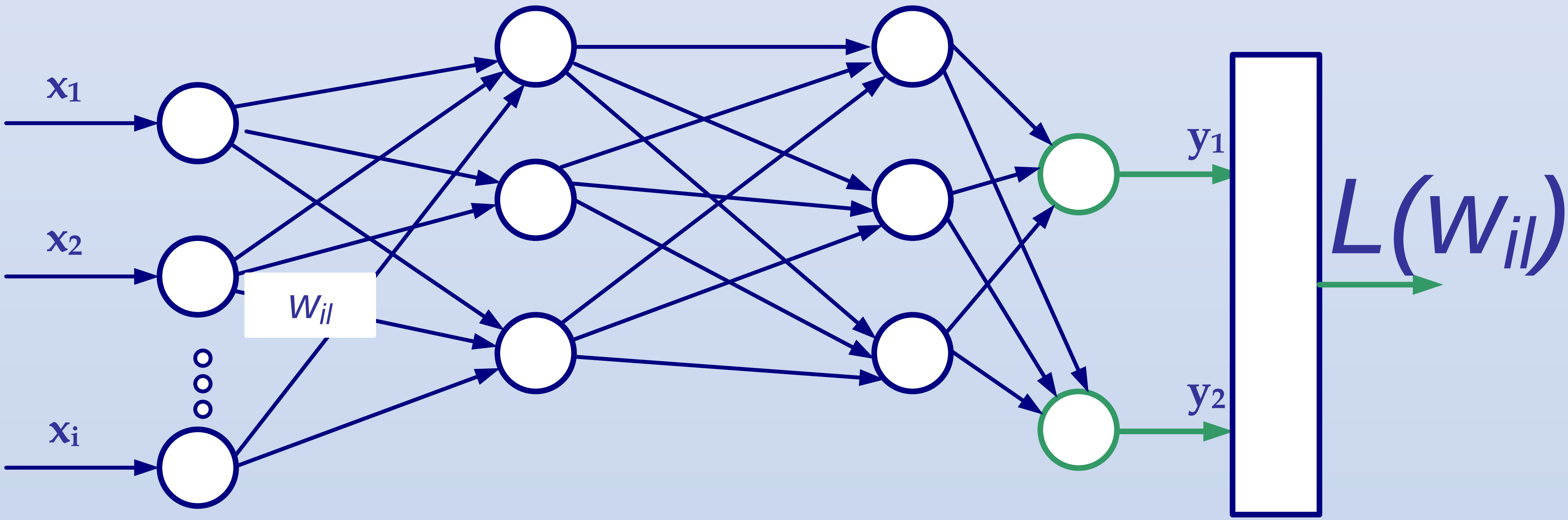
$$L_{ac}(W) = \sum L_i(W);\ L(W) = \frac{L_{ac}(W)}{M}$$

Визначається середнє значення втрат (зазвичай означається як Cost).

Далі розраховуються компоненти вектору градієнту $\nabla L(W)$ відносно кожної ваги мережі.

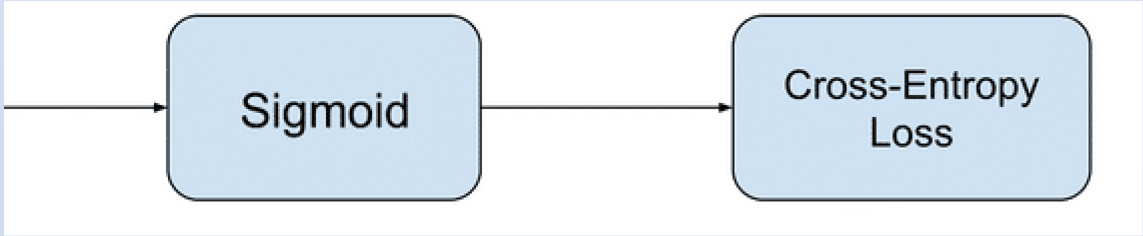
Проблема обчислення градієнту

Необхідно обчислити градієнт $\Delta w_t = \nabla L(w_t)$ з урахуванням великої кількості ваг та складної залежності похибки від ваг.



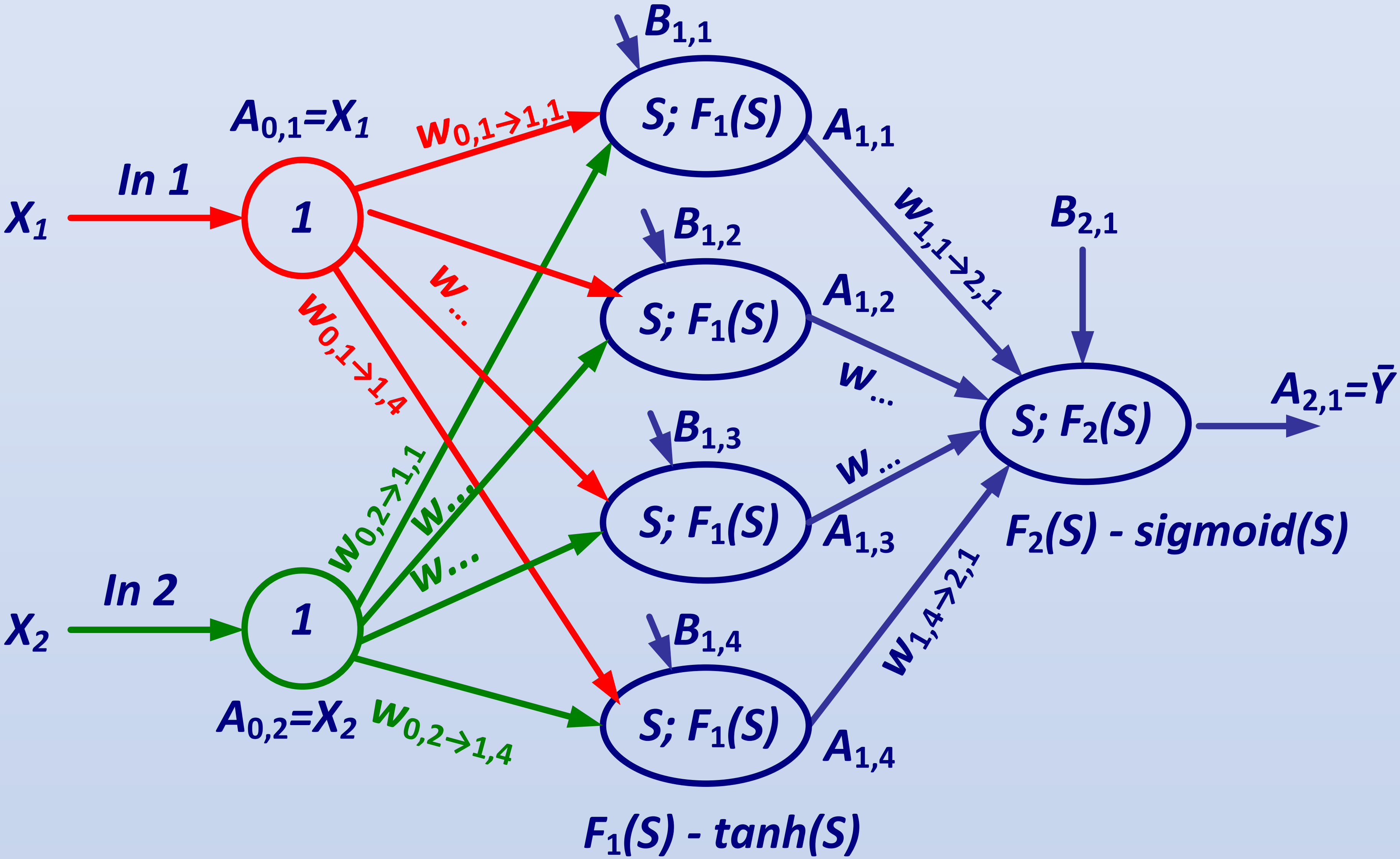
Бінарна класифікація

Завдання → передбачення ймовірності того, що екземпляр належить першому класу, наприклад, класу, мітка якого 1 (ціле), тоді як мітка іншого класу 0.

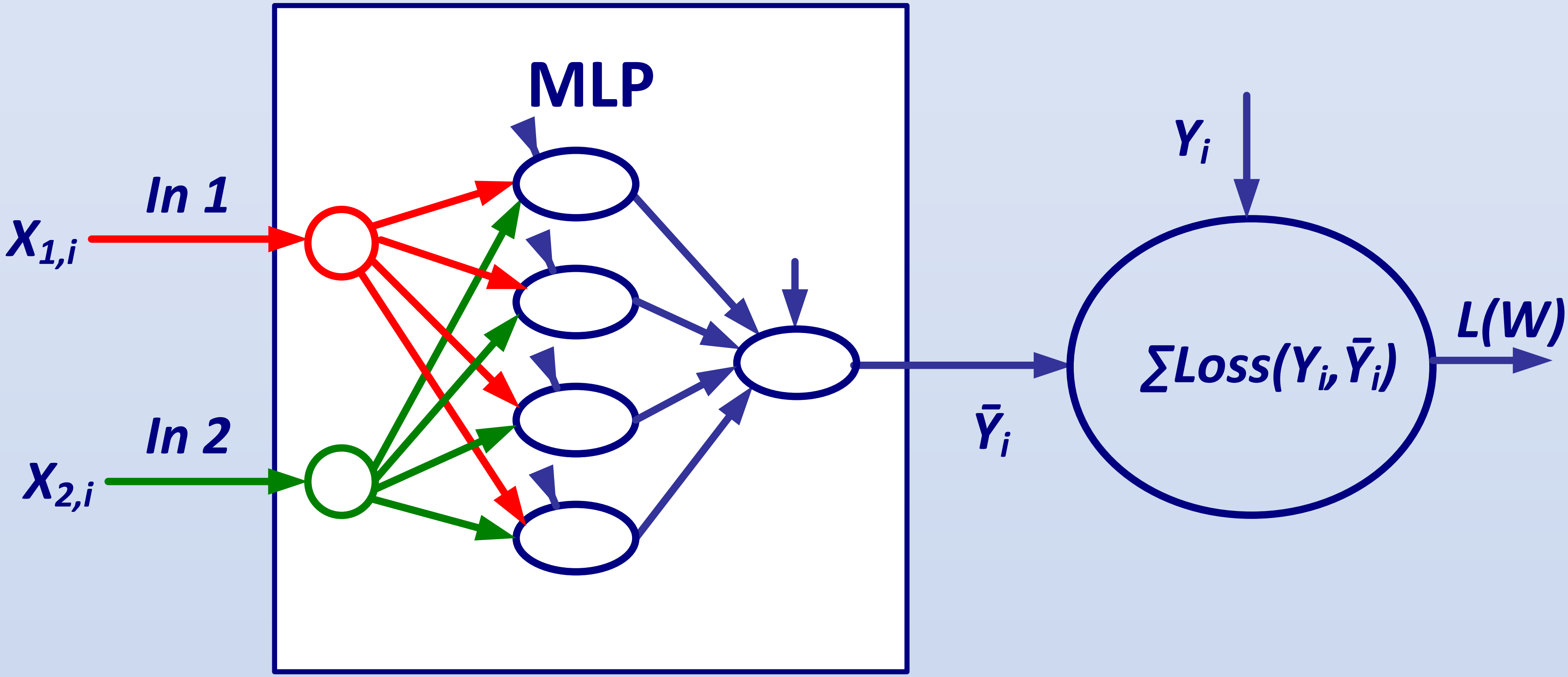


- **Конфігурація вихідного шару: один вузол із сигмоподібною активаційною функцією.**
- **Функція втрат: крос-ентропія (також називається логарифмічною функцією втрат).**

Приклад. Проста MLP



Приклад. Проста MLP



- Конфігурація вихідного шару: один вузол із сигмоподібною активаційною функцією.
- Функція втрат: крос-ентропія (логістична функція втрат).

Приклад. Проста MLP

Обчислення похибки (крос ентропія для бінарної класифікації)

$$Loss = -Ylog\left(\hat{Y}\right) + (1 - Y)log(1 - \hat{Y} \quad)$$

Функція активації вихідного нейрону

$$\hat{Y} = F(S) = {}^{1}\!/_{1 + e^{-S}} =$$

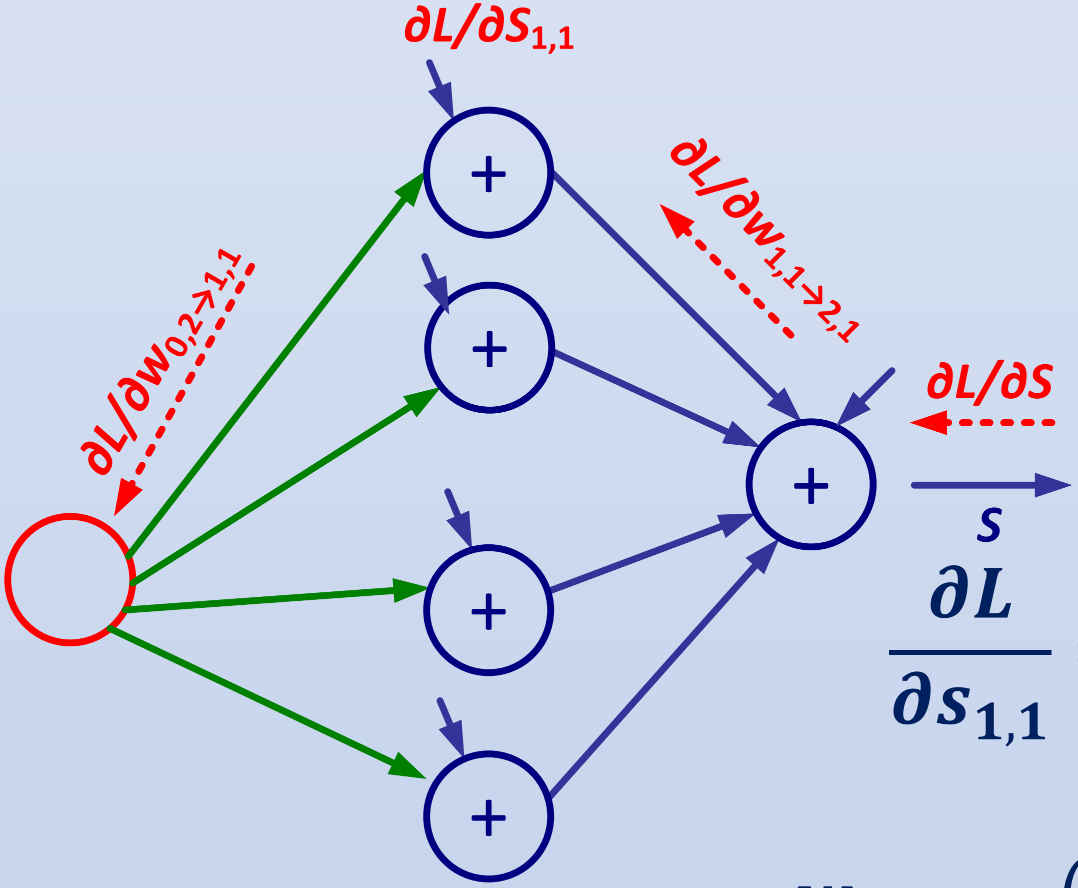
! Градієнт відносно S вихідного нейрону

$$^{\partial Loss}\!/_{\partial S} = \left(\hat{Y} - Y\right)$$

Приклад. Проста MLP

Обчислення компонентів градієнту

$$\frac{\partial L}{\partial w_{1,1-2,1}} = \frac{\partial L}{\partial S} * \frac{\partial S}{\partial w_{1,1-2,1}} = \left(\widehat{Y} - Y\right) * A_{1,1}$$



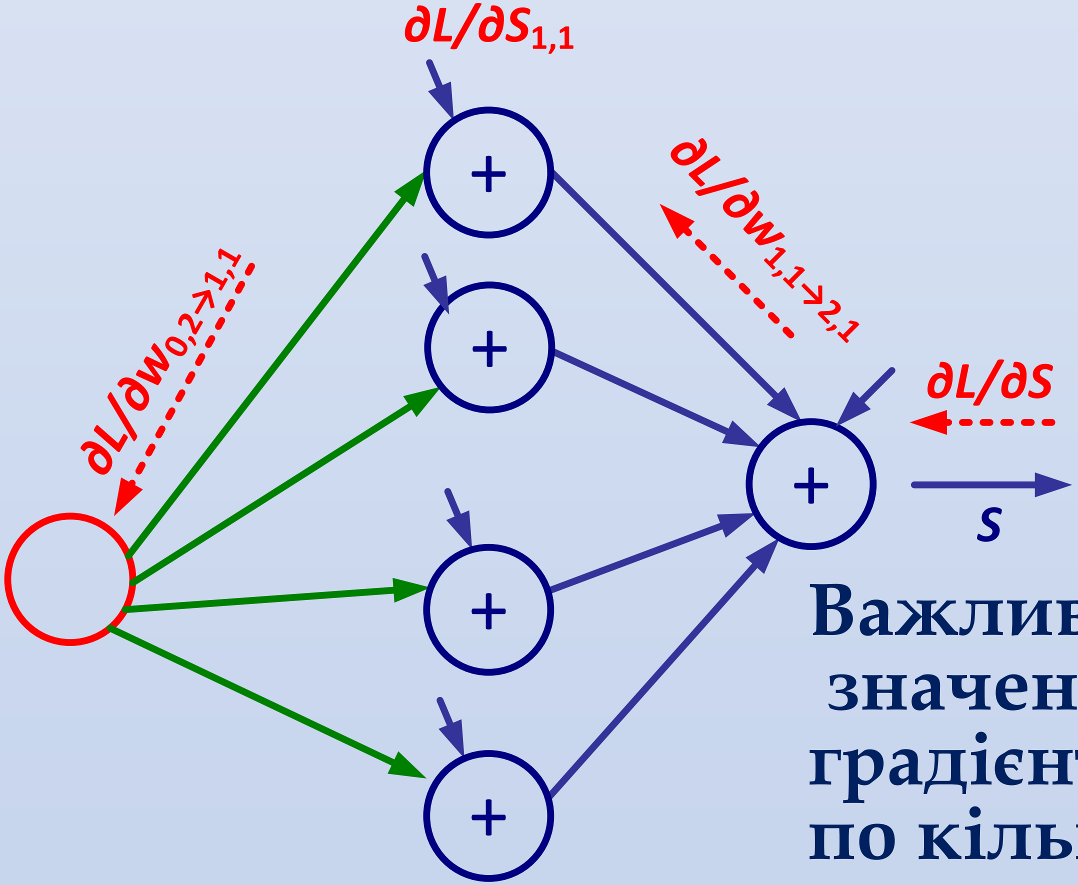
$$\frac{\partial L}{\partial s_{1,1}} = \frac{\partial L}{\partial A_{1,1}} * \frac{\partial A_{1,1}}{\partial s_{1,1}}$$

$$= w_{1,1\rightarrow 2,1}\left(\widehat{Y} - Y\right) * (1 - A_{1,1})^2$$

Приклад. Проста MLP

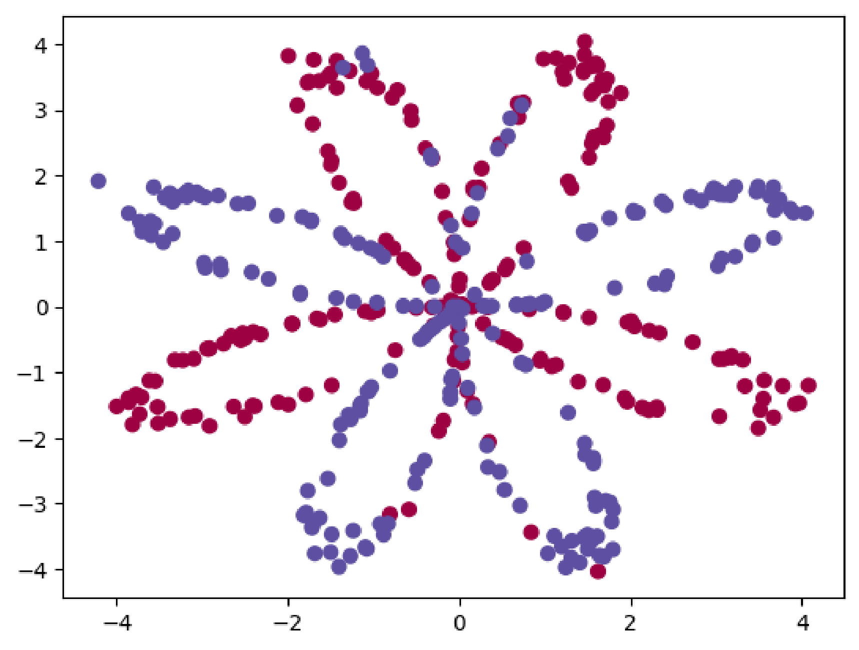
Обчислення компонентів градієнту

$$\frac{\partial L}{\partial w_{0,2-1,1}} = \frac{\partial L}{\partial s_{1,1}} * \frac{\partial s_{1,1}}{\partial w_{0,2-1,1}} = \frac{\partial L}{\partial s_{1,1}} * x_2$$

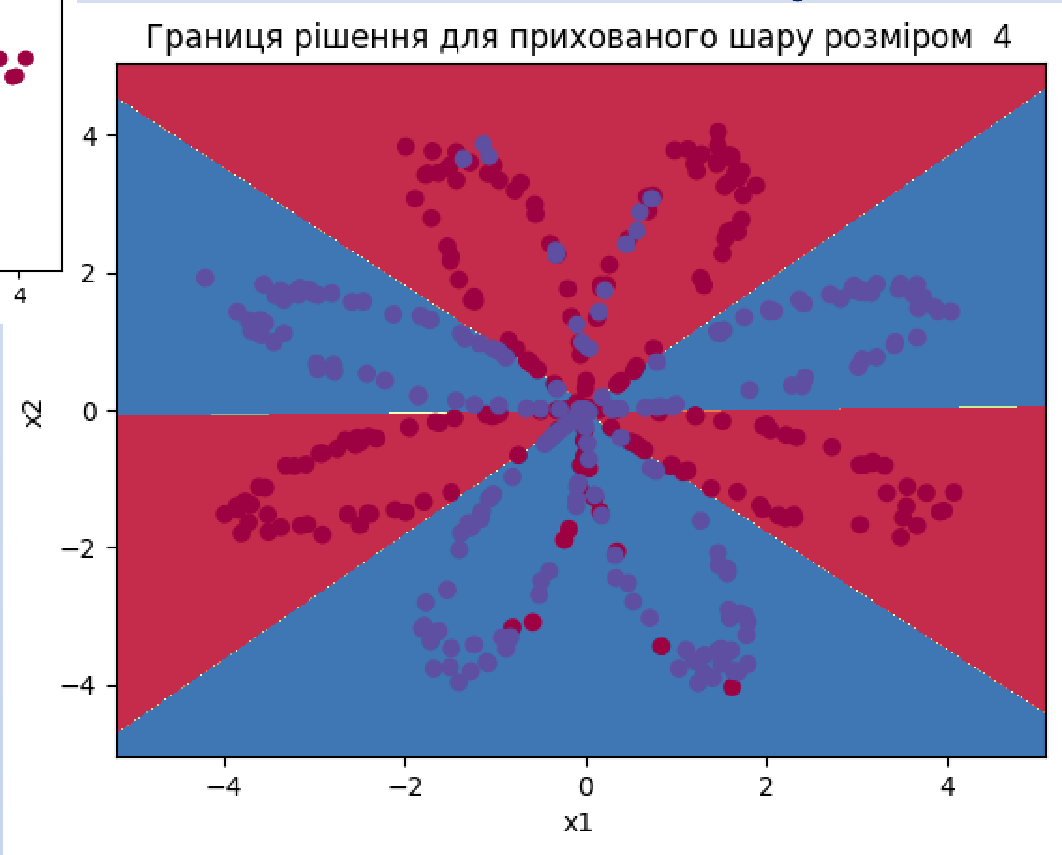


Важливо:
значення компонент градієнтів осереднюється по кількості тренувальних екземплярів.

Приклад. Проста MLP



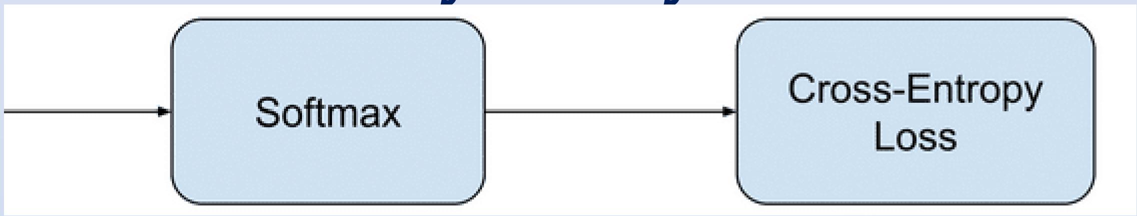
Accuracy: 90%



Дивись: lec_05_04_Exmpl_1

Мульті-класова класифікація

Завдання → класифікувати екземпляр як один із кількох класів. Завдання сформульована як прогноз ймовірності того, що приклад належить кожному класу.

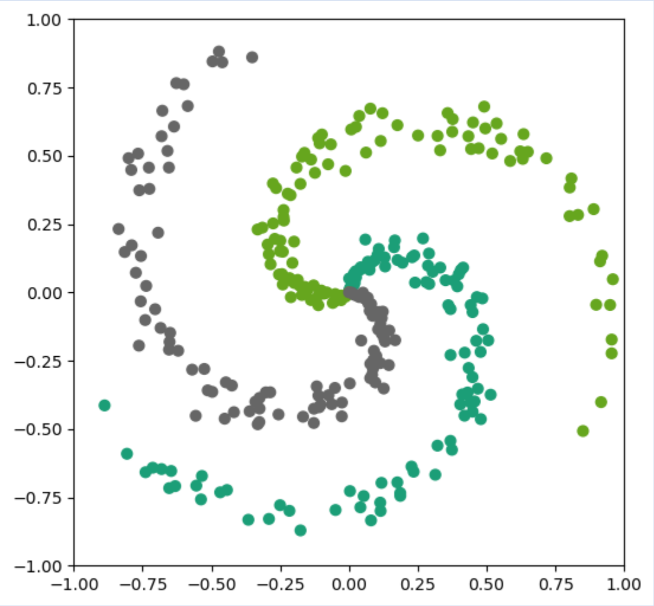


- **Конфігурація вихідного шару: один вузол для кожного класу з використанням функції активації softmax.**
- **Функція втрат: крос-ентропія.**

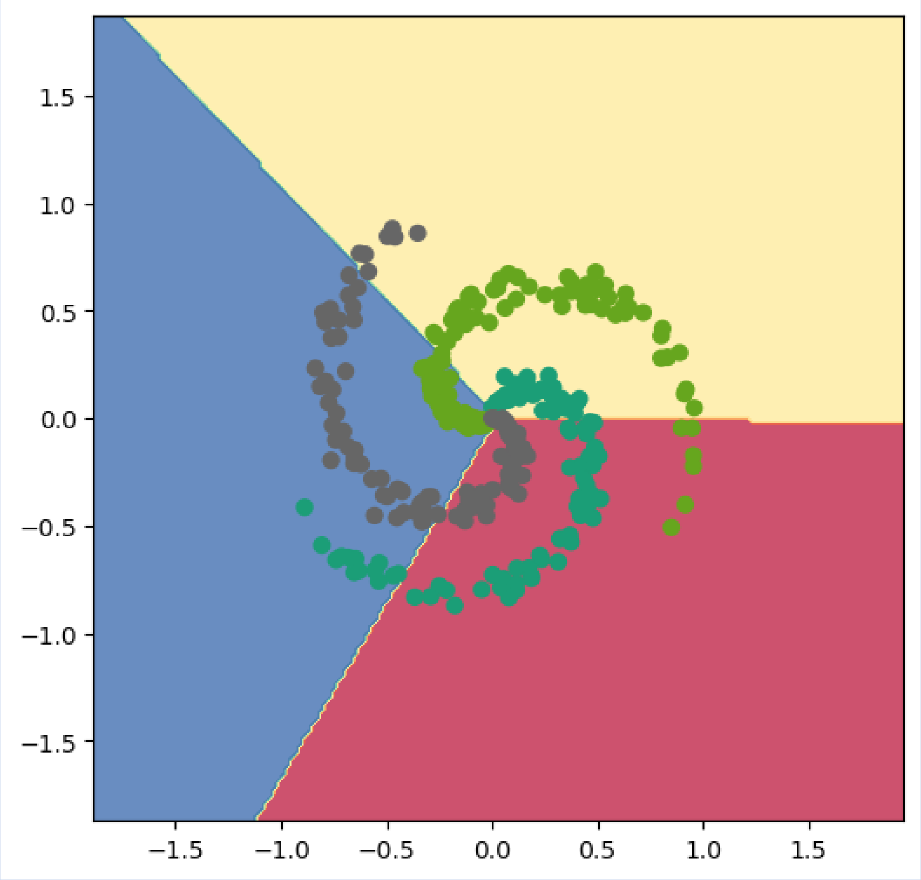
$$L\ \ (W) = \frac{1}{M}\sum_i \sum_j Y_i log(F(X_j))$$

***i* = *1,2,…,M* (екземпляри)**
***j* = *1,2,…,N* (класи)**

Приклад. Проста MLP.

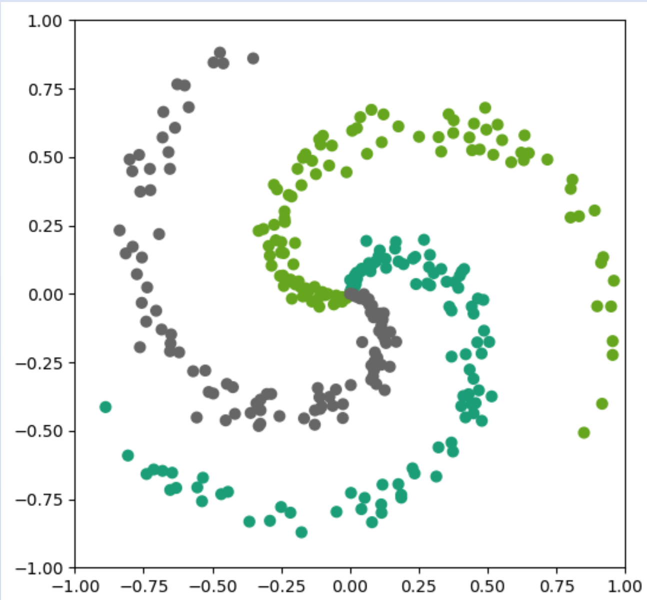


Звичайний лінійний класифікатор
Accuracy: 49%

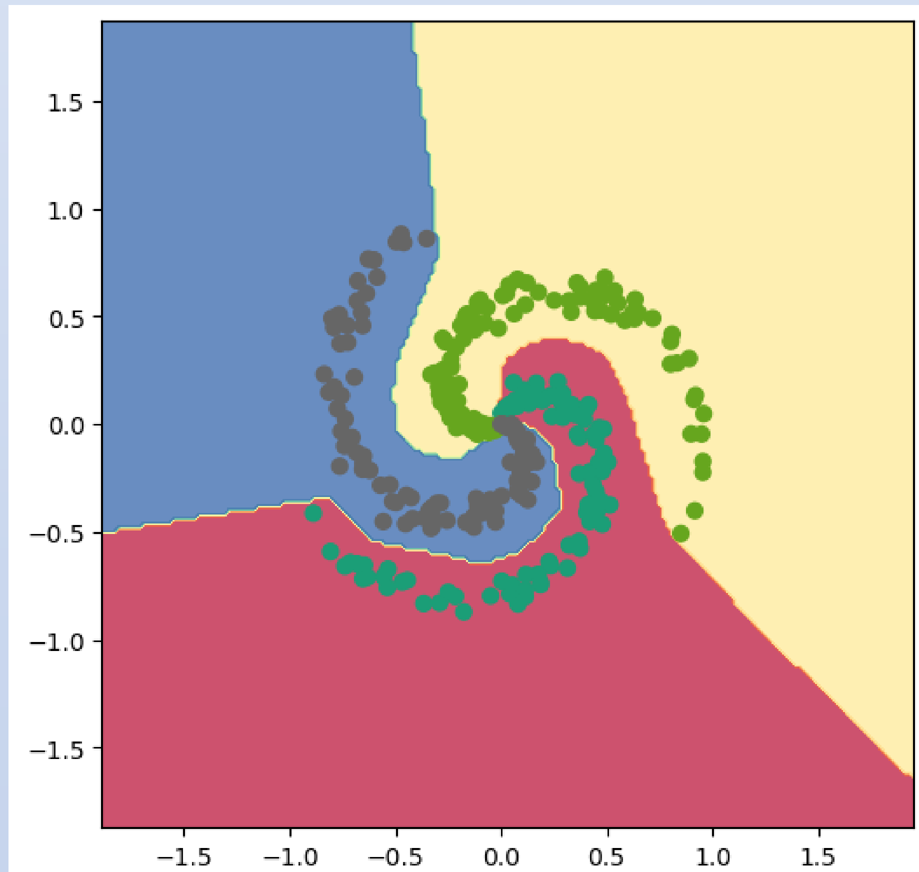


Дивись: lec_05_04_Exmpl_2

Приклад. Проста MLP.



MLP класифікатор
Accuracy: 98%



Дивись: lec_05_04_Exmpl_2

Рекомендована ЛІТЕРАТУРА

- **Anirudh Koul, Siddha Ganju, Meher Kasam. Practical Deep Learning for Clous, Mobile, and Edge / O'Reilly Media, Inc., 2019 . - 620 p.**
- **Chollet F. Deep learning with Python. - Second Ed., Manning Publications Co., 2021 . - 504 p.**
- **Trask A. W. Grokking Deep Learning. — MANNING Shelter Island, 2019. — 336 p.**
- **Ramsundar B., Eastman P., Walters P., Pande V. Deep Learning for the Life Sciences . — O’Reilly Media, 2019 . — 321 p.**

The END
Модуль 5. Лекція 04.