

# НЕЙРОННА МЕРЕЖА

*Лекція 4*

# План

1. Машинне навчання
2. Machine learning algorithms
3. Штучна нейронна мережа
4. The Concept of Artificial Neurons in Neural Networks
5. Функція активації
6. Структура штучної нейронної мережі
7. Типи архітектури нейронної мережі
8. Глибоке навчання

# МАШИННЕ НАВЧАННЯ

■ **Контрольоване навчання:** дерева рішень, ансамблі випадкові ліси), k-NN, лінійна регресія, наївний Байєс, нейронні мережі, логістична регресія, SVM

- Класифікація
- Регресія (прогноз)

■ **Неконтрольоване навчання:** k-середні, c- середні, ієрархічна кластеризація, DBSCAN

- Кластеризація
- Зниження розмірності (PCA, LDA, факторний аналіз, t-SNE)
- Правила асоціації (ринковий аналіз)

■ **Навчання з підкріпленням**

- Динамічне програмування

■ **Нейронні мережі:** глибоке навчання, багатошаровий перцептрон, рекурентна нейронна мережа (RNN), згорткова нейронна мережа (CNN), генеративна змагальна мережа (GAN)

# MACHINE LEARNING ALGORITHMS



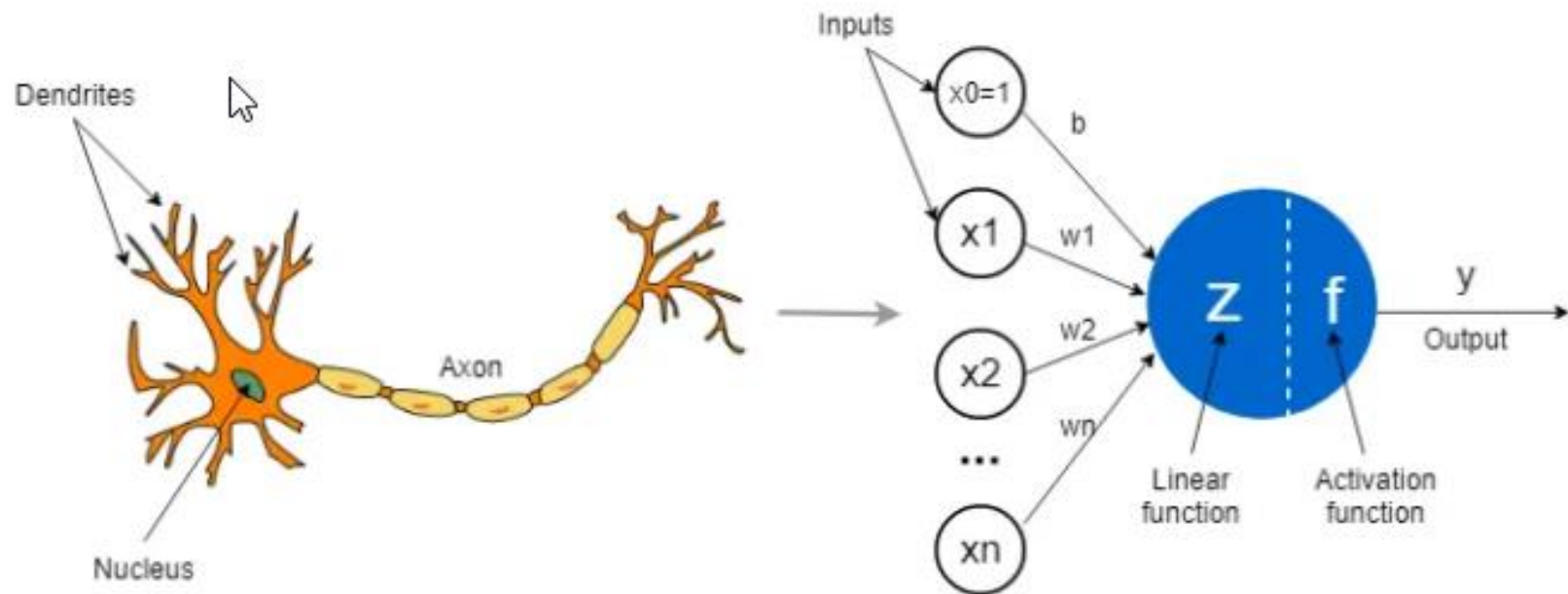
# **ШТУЧНА НЕЙРОННА МЕРЕЖА**

Нейронні мережі – один із напрямків штучного інтелекту, мета якого змоделювати аналітичні механізми, що здійснюються людським мозком.

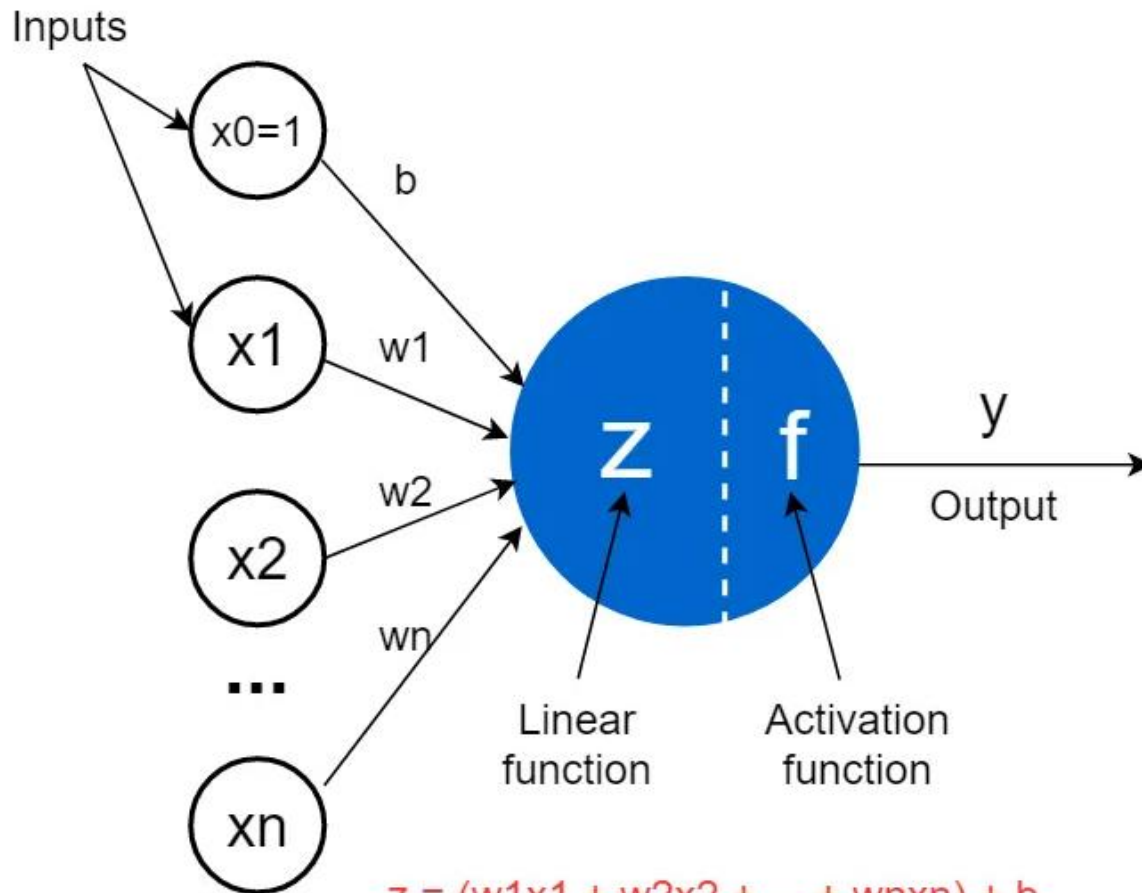
Завдання, які вирішує типова нейронна мережа – класифікація, прогноз та розпізнавання. Нейронні мережі здатні самотійно навчатися і розвиватися, будуючи свій досвід на помилках.

Нейронні мережі це послідовність нейронів, з'єднаних між собою. Структура нейронної мережі прийшла у світ програмування прямо з біології. Завдяки такій структурі машина знаходить здатність аналізувати і навіть запам'ятовувати різну інформацію.

# THE CONCEPT OF ARTIFICIAL NEURONS IN NEURAL NETWORKS



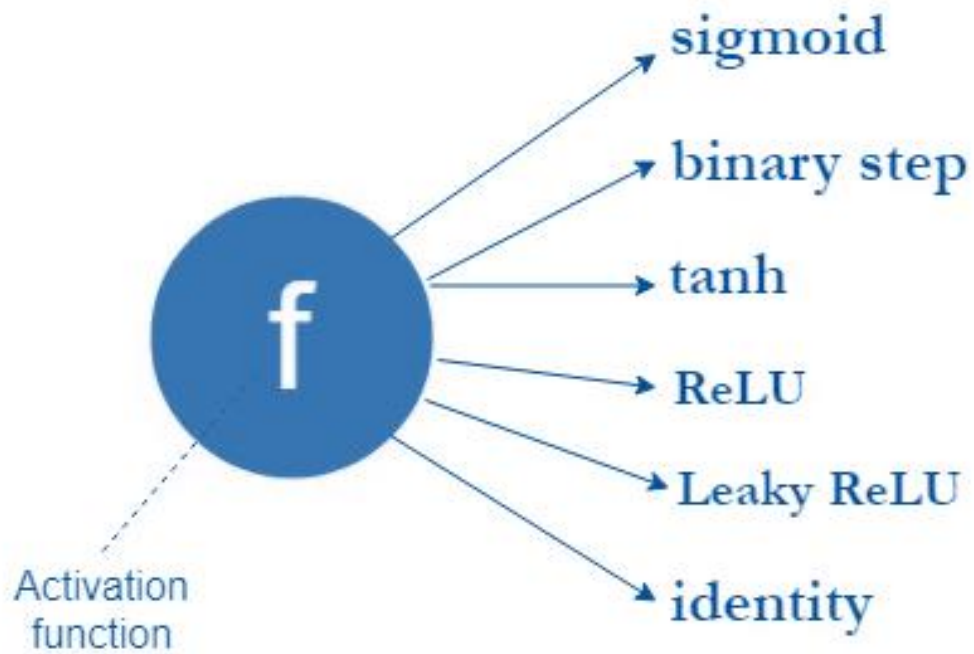
# Будова перцептрона



$$z = (w_1x_1 + w_2x_2 + \dots + w_nx_n) + b$$

$$y = f(z)$$

# ФУНКЦІЯ АКТИВАЦІЇ





# Приклад

Уявіть, що ми маємо 3 входи з наступними значеннями:  $x_1=2$  ,  $x_2=3$ ,  $x_3=1$

Оскільки у нас є 3 входи, ми також маємо 3 ваги, які контролюють рівень важливості кожного входу. Припустимо що  $w_1=0,5$  ,  $w_2=0,2$  і  $w_3=10$

Значення для одиниці зсуву:  $b=2$

Обчислимо лінійну функцію  $z$  :  $z = (0,5*2 + 0,2*3 + 10*1) + 2 = 13,6$

Функція активації приймає вихідні дані  $z$  (13.6) як вхідні дані та обчислює  $y$  на основі функції активації(наприклад, функція *сигмоїдної* активації).

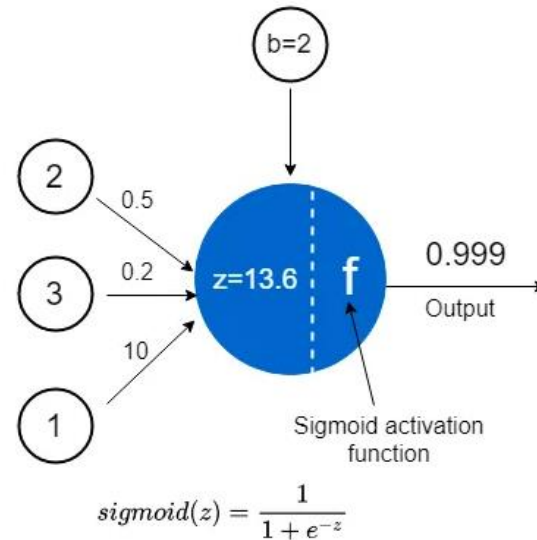
$$\text{sigmoid}(z) = \frac{1}{1 + e^{-z}}$$

Функція активації Sigmoid

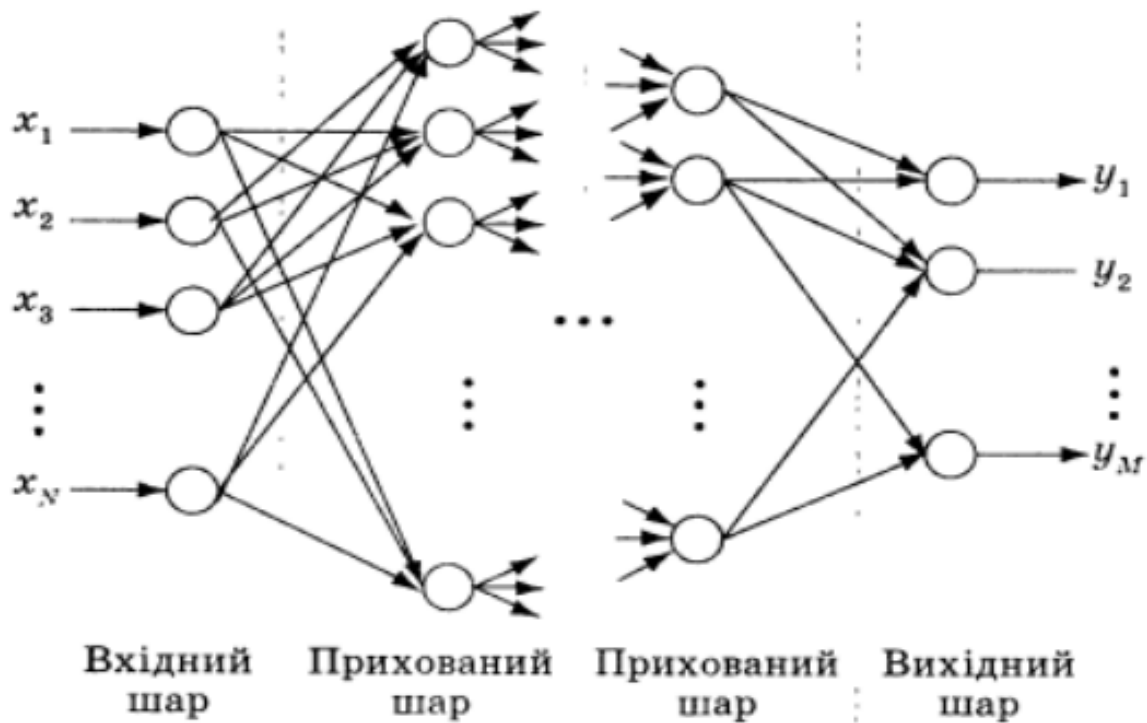
$y = \text{sigmoid}(13.6)$ ,

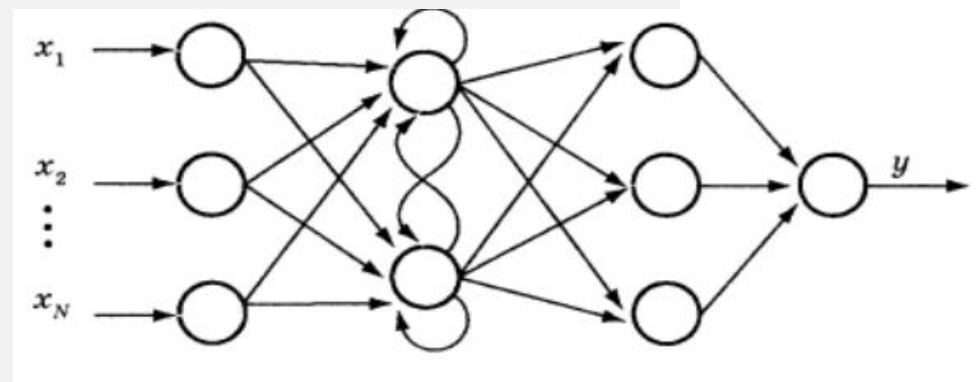
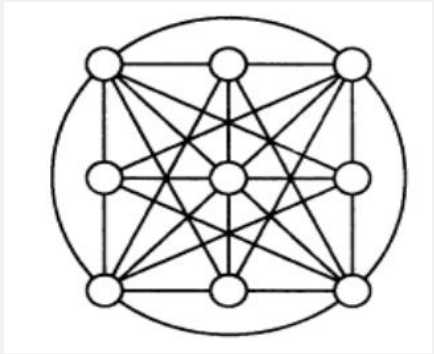
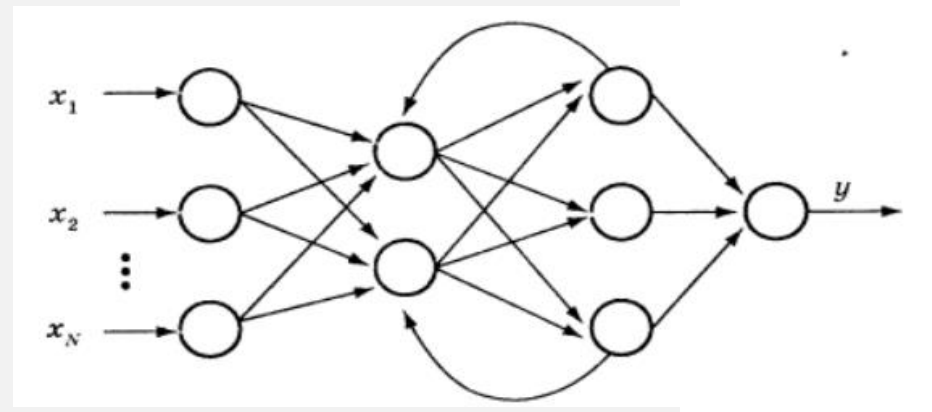
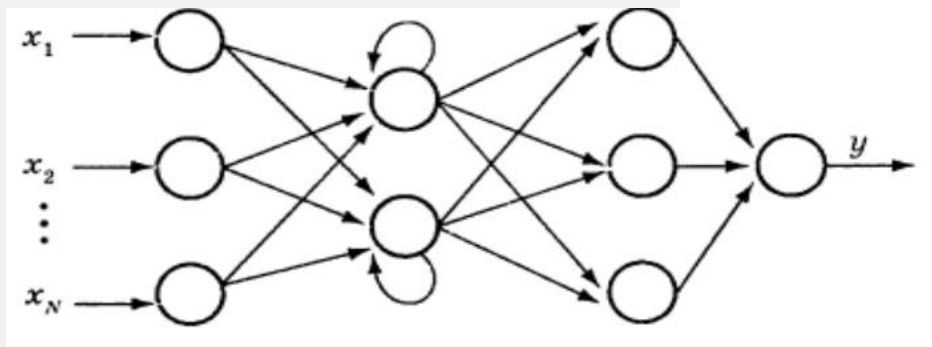
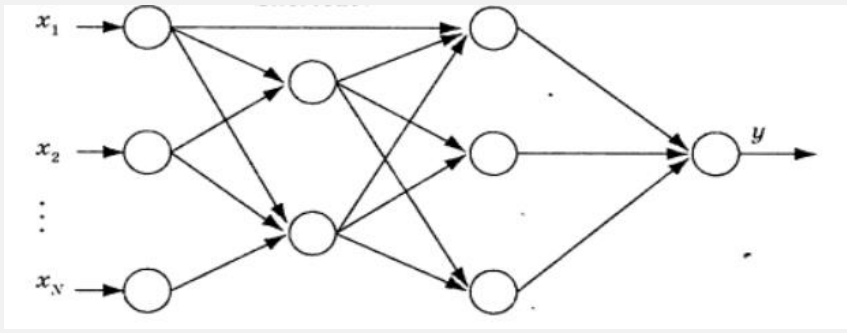
$y = 0,999$ ,

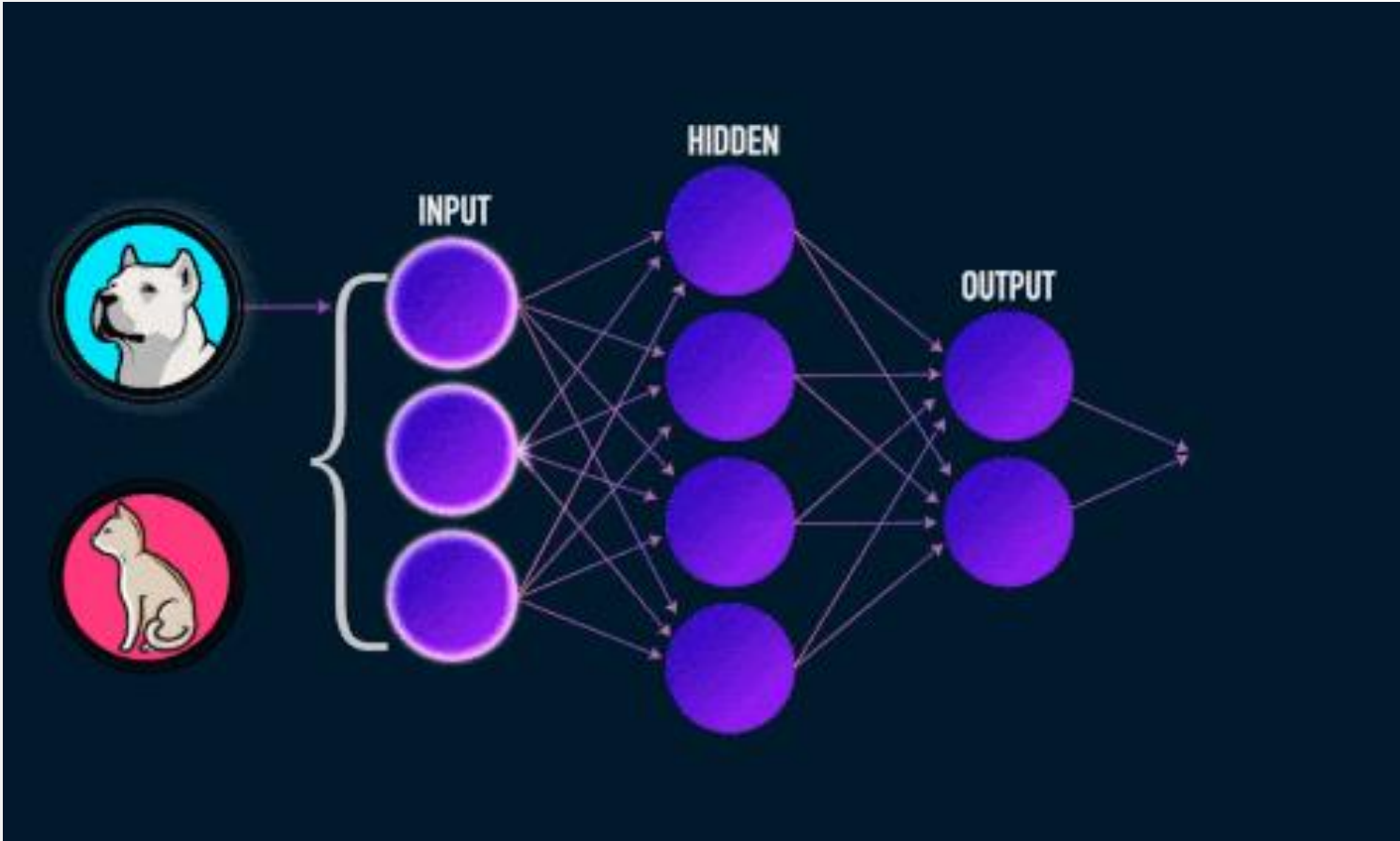
$y \sim 1$



# СТРУКТУРА ШТУЧНОЇ НЕЙРОННОЇ МЕРЕЖІ







# Приклад

У нейронній мережі прямого розповсюдження кожен вихід нейрону стає входом для наступного шару нейронів.

1) Вхід до прихованого шару обчислюється за формулою:

$$Z_{np} = \sum_{i=1}^N x_i w_i + b_1$$

де  $N$  - кількість нейронів у вхідному шарі (дорівнює кількості ознак)

$x_i$  - вхідні дані

$w_i$  - ваги

$b_1$  – біас (зсув)

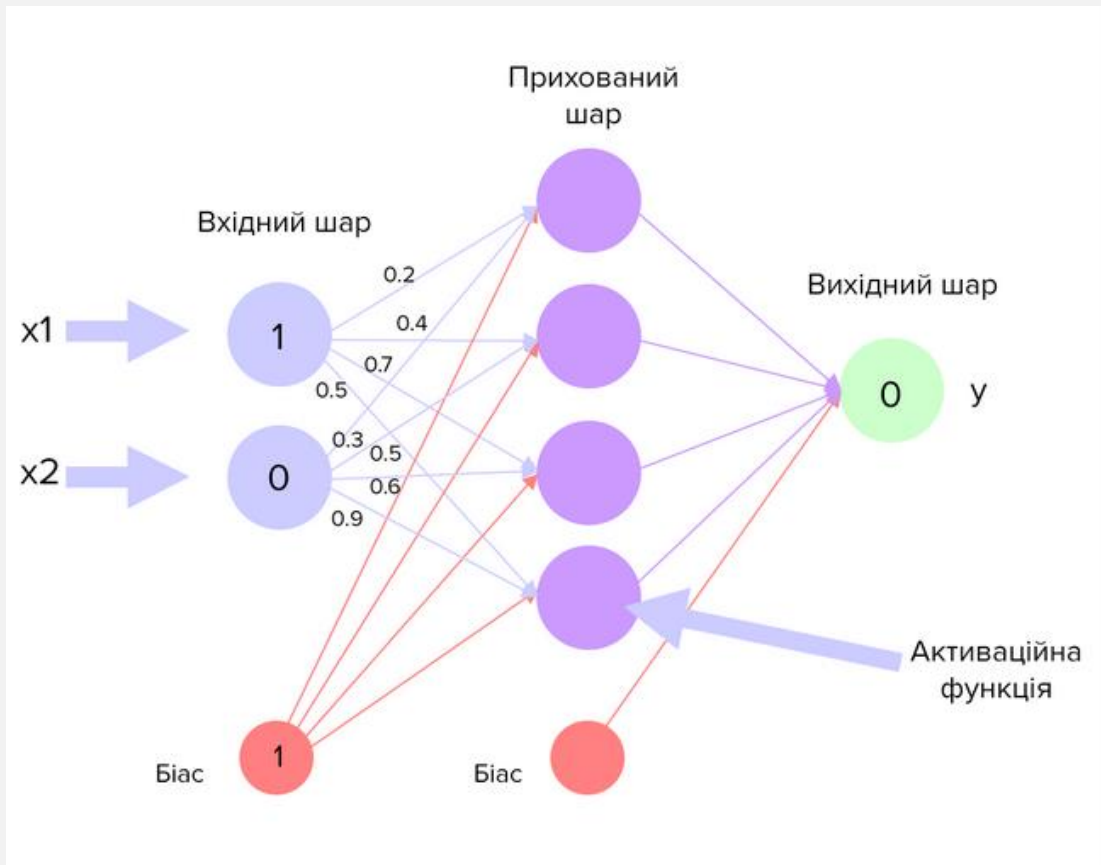
Нехай ваги та біас мають наступні значення:

$w_1=0.2, w_5=0.3, b_1=1.$

$w_2=0.4, w_6=0.5,$

$w_3=0.7, w_7=0.6,$

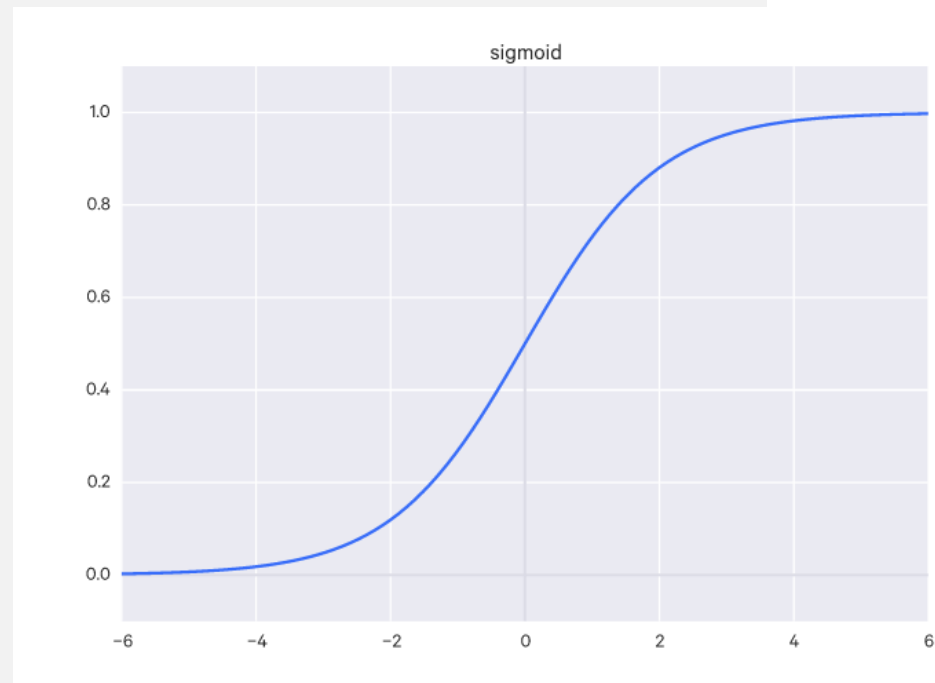
$w_4=0.5, w_8=0.9,$



Обчислимо вхід до першого, другого, третього та четвертого нейронів прихованого шару

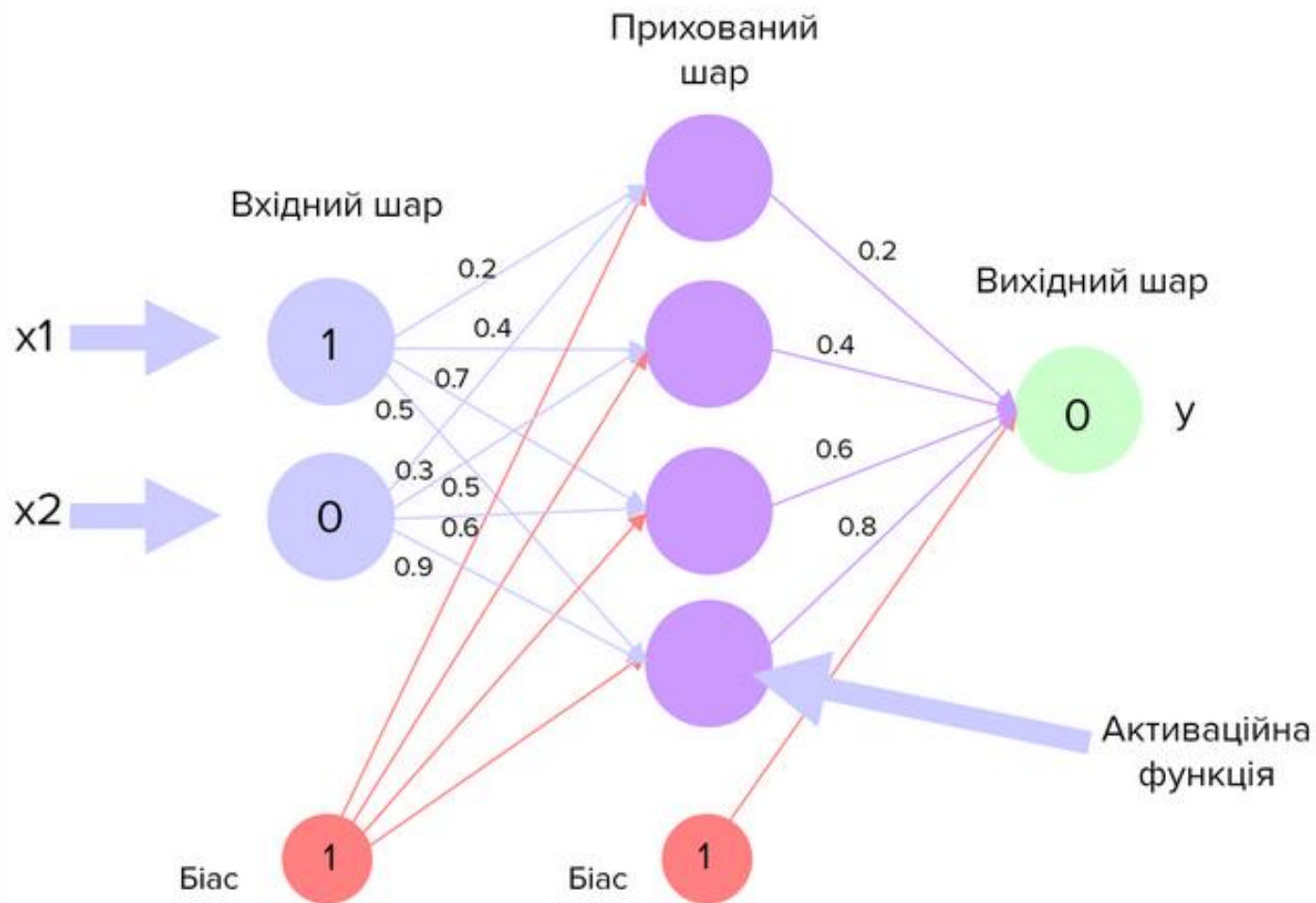
Кожен нейрон прихованого шару містить активаційну функцію. Вона може бути будь-якою, але будемо використовувати сігмоїд-функцію: Ця функція набуває значень на проміжку  $[0;1]$ .

$$y_{np} = \frac{1}{1 + e^{-z_{np}}}$$



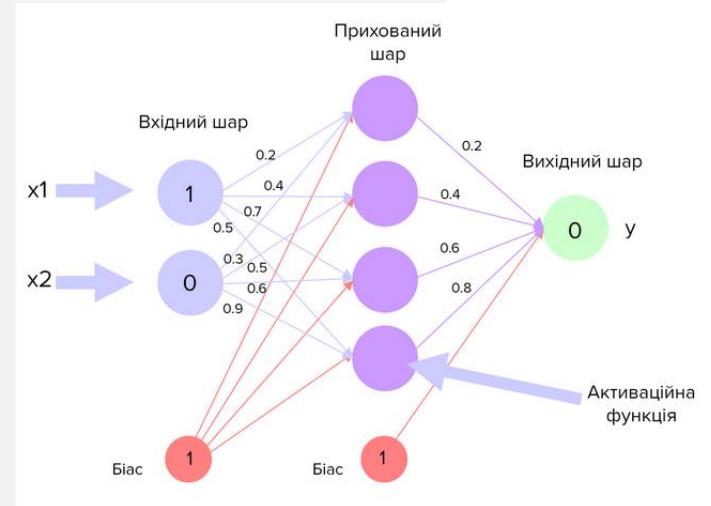
Обчислити результат роботи активаційної функції для першого нейрону прихованого шару.

Обчислити, чому дорівнює загальний вихід нейронної мережі





Обчислити, чому дорівнює загальний вихід нейронної мережі



	A	B	C	D	E	F
нейрони прихованого шару		вхідні дані	застосування функції активації	дані, які отримує вихідний шар	$\Sigma$	$y$
1	1	1,2	0,768524783	0,153704957		
2	2	1,4	0,802183889	0,320873555		
3	3	1,7	0,845534735	0,507320841	2,635958934	0,93314
4	4	1,5	0,817574476	0,654059581		

	A	B	C	D	E	F
нейрони прихованого шару		вхідні дані	застосування функції активації	дані, які отримує вихідний шар	$\Sigma$	$y$
1	1	$=1*0,2+0*0,3+1$	$=1/(1+EXP(-B2))$	$=C2*0,2$		
2	2	$=1*0,4+0*0,5+1$	$=1/(1+EXP(-B3))$	$=C3*0,4$		
3	3	$=1*0,7+0*0,6+1$	$=1/(1+EXP(-B4))$	$=C4*0,6$	$=SUM(D2:D5)+1$	$=1/(1+EXP(-E4))$
4	4	$=1*0,5+0*0,9+1$	$=1/(1+EXP(-B5))$	$=C5*0,8$		

# ТИПИ АРХІТЕКТУРИ НЕЙРОННОЇ МЕРЕЖІ

- Багатошаровий персептрон
- Згорткова нейронна мережа (CNN)
- Рекурентна нейронна мережа (RNN)
  - Нейронна мережа довготривалої короткочасної пам'яті (LSTM)
- Генеративно-змагальні нейронні мережі (GAN)
- Мережа Хопфілда
- Мережа Хеммінга
- ....

# Приклад

застосування генеративно-змагальної нейронної мережі

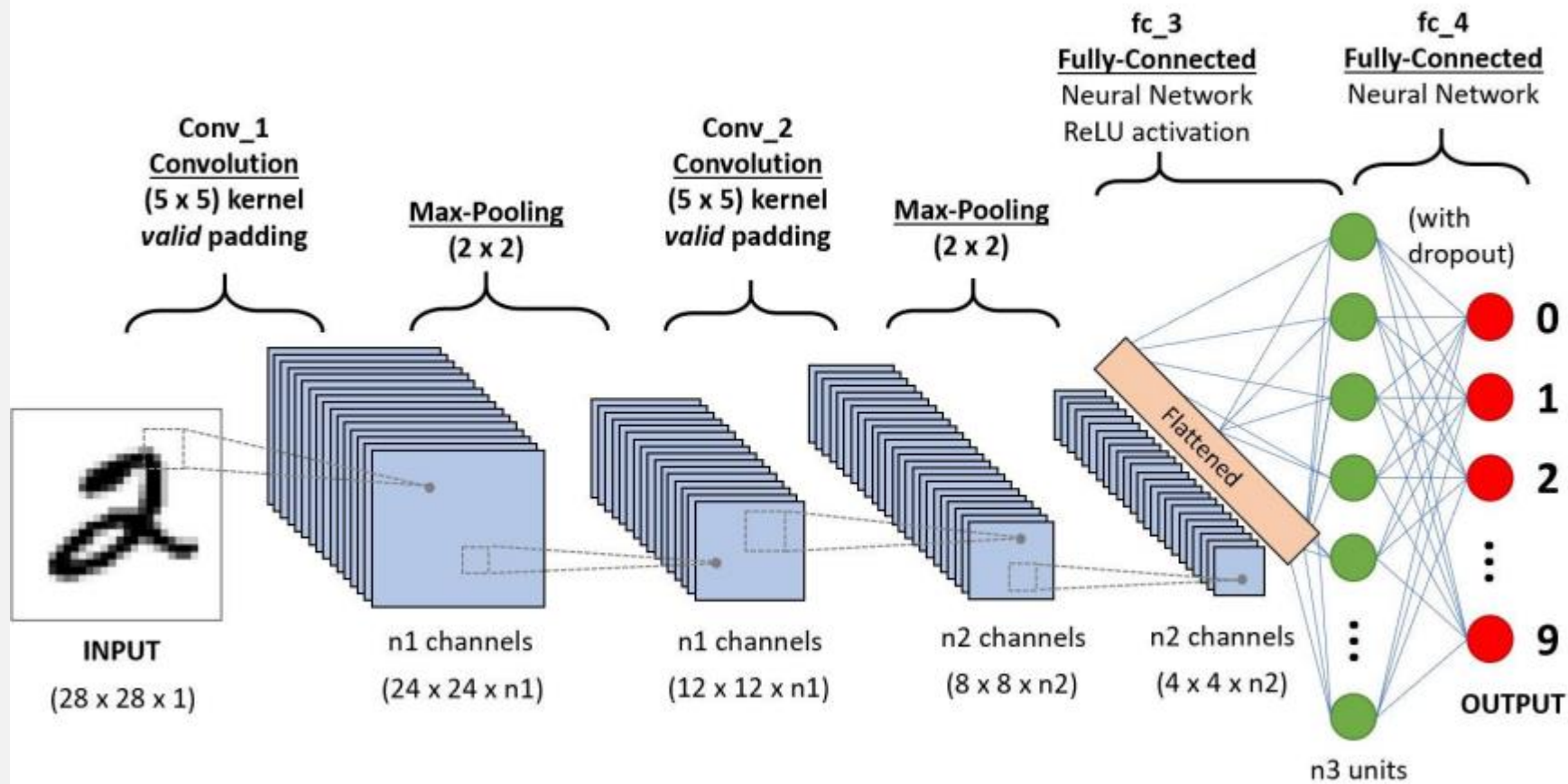


# **ЗГОРТКОВА НЕЙРОННА МЕРЕЖА**

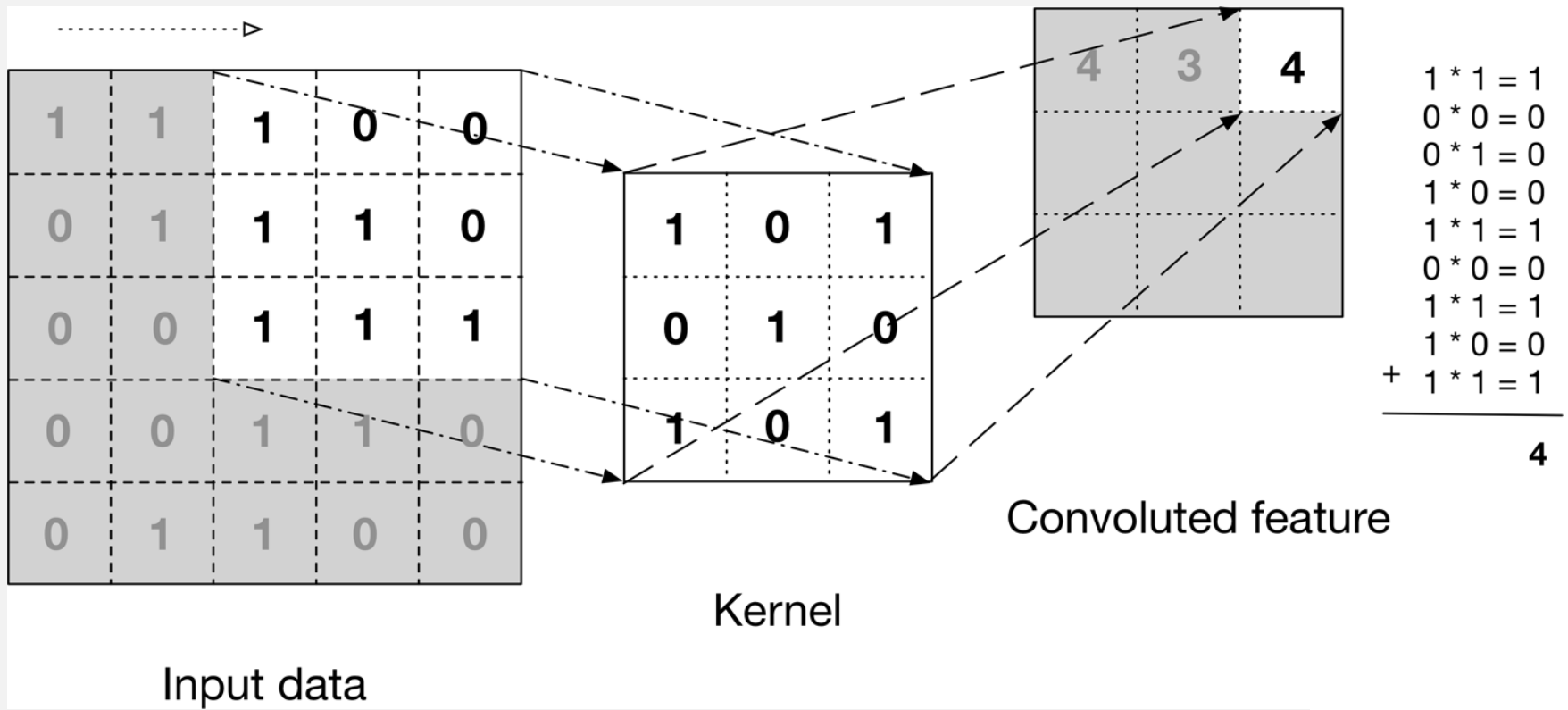
Згорткова нейронна мережа (ЗНМ, CNN) – спеціальна архітектура штучних нейронних мереж, запропонована Яном Лекуном і націлена на ефективне розпізнавання образів. Даній архітектурі вдається набагато точніше розпізнавати об'єкти на зображеннях, оскільки, на відміну багат шарового персептрона, враховується двомірна топологія зображення. При цьому згортковій мережі стійкі до невеликих зсувів, змін масштабу та поворотів об'єктів на вхідних зображеннях.

Архітектури, засновані на згорткових мережах, досі займають перші місця у змаганнях з розпізнавання образів.

Згорткова нейронна мережа є основним інструментом для класифікації та розпізнавання об'єктів, осіб на фотографіях, розпізнавання мови. Є безліч варіантів застосування CNN, такі як Deep Convolutional Neural Network (DCNN), Region-CNN (R-CNN), Fully Convolutional Neural Networks (FCNN), Mask R-CNN та інші.



# Convolution Layer



# Convolution Layer

1 <sub>x1</sub>	1 <sub>x0</sub>	1 <sub>x1</sub>	0	0
0 <sub>x0</sub>	1 <sub>x1</sub>	1 <sub>x0</sub>	1	0
0 <sub>x1</sub>	0 <sub>x0</sub>	1 <sub>x1</sub>	1	1
0	0	1	1	0
0	1	1	0	0

Image

4		

Convolved  
Feature

# Convolution Layer

0	0	0	0	0	0	...
0	156	155	156	158	158	...
0	153	154	157	159	159	...
0	149	151	155	158	159	...
0	146	146	149	153	158	...
0	145	143	143	148	158	...
...	...	...	...	...	...	...

Input Channel #1 (Red)

0	0	0	0	0	0	...
0	167	166	167	169	169	...
0	164	165	168	170	170	...
0	160	162	166	169	170	...
0	156	156	159	163	168	...
0	155	153	153	158	168	...
...	...	...	...	...	...	...

Input Channel #2 (Green)

0	0	0	0	0	0	...
0	163	162	163	165	165	...
0	160	161	164	166	166	...
0	156	158	162	165	166	...
0	155	155	158	162	167	...
0	154	152	152	157	167	...
...	...	...	...	...	...	...

Input Channel #3 (Blue)

-1	-1	1
0	1	-1
0	1	1

Kernel Channel #1



308

+

1	0	0
1	-1	-1
1	0	-1

Kernel Channel #2



-498

+

0	1	1
0	1	0
1	-1	1

Kernel Channel #3



164

+ 1 = -25

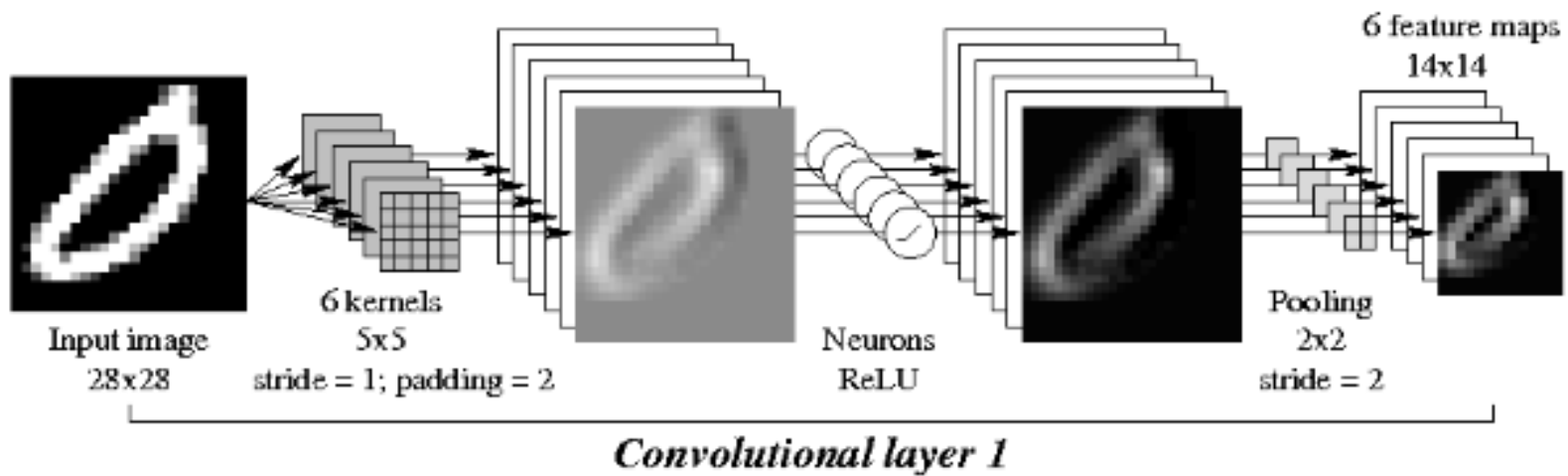


Bias = 1

Output

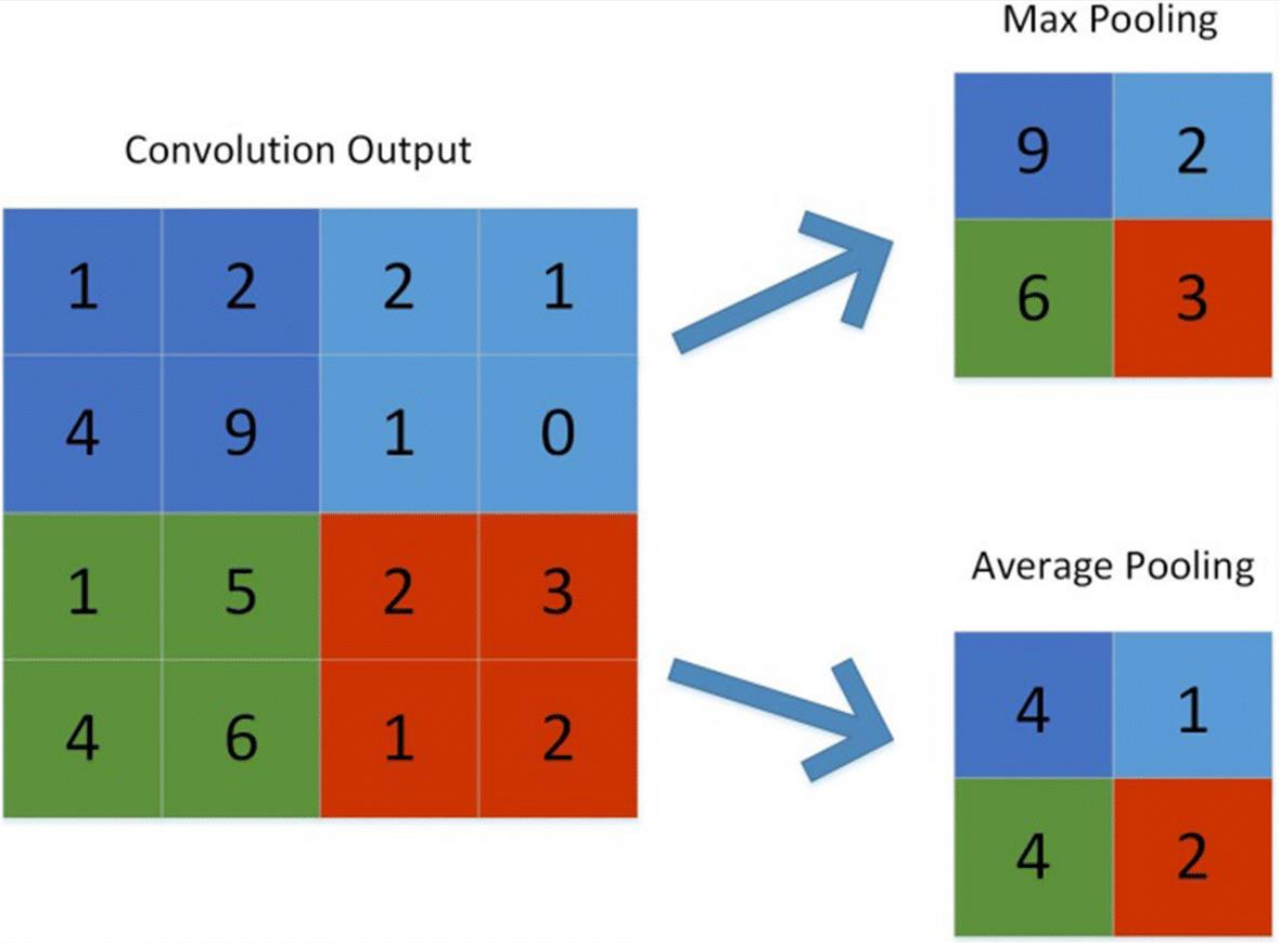
-25				...
				...
				...
				...
...	...	...	...	...



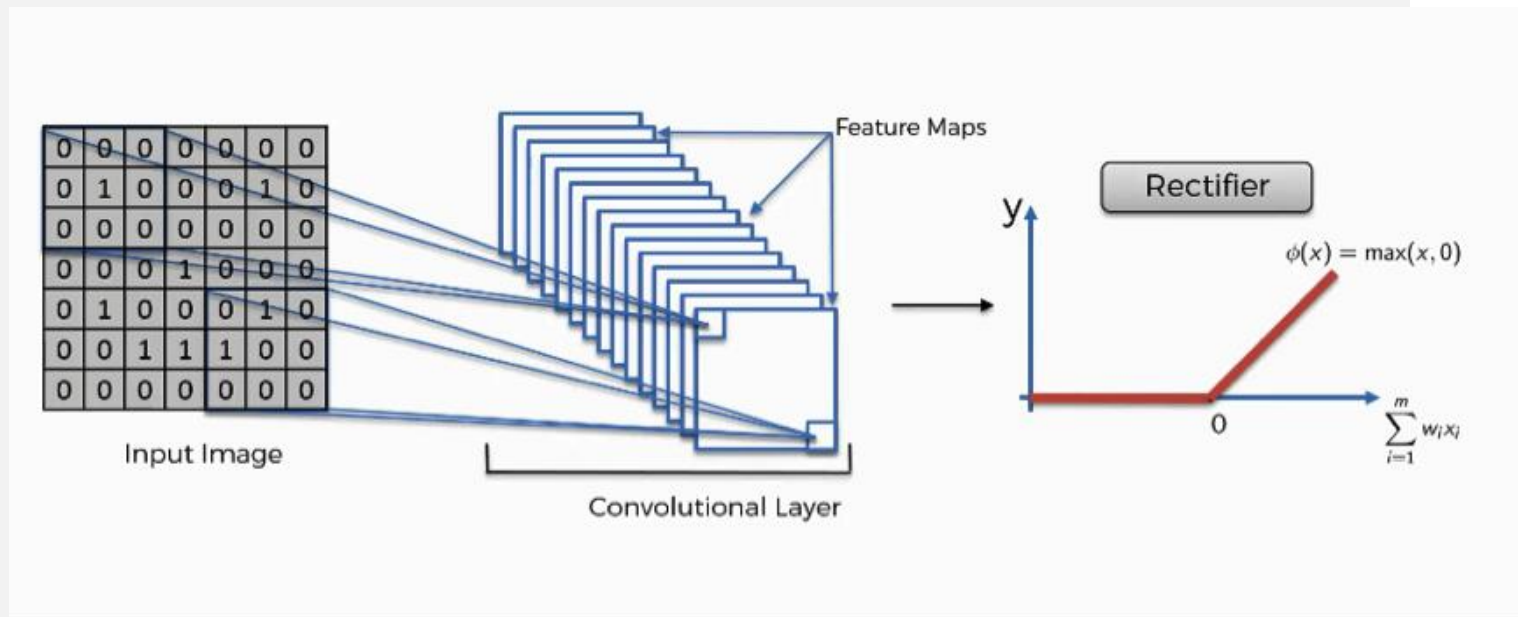


Згорткова нейронна мережа для задачі MNIST: детальний вигляд першого згорткового шару.

# Pooling Layer



# ReLU

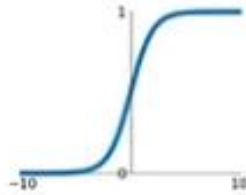


Математичне визначення функції активації ReLU -  $f(x) = \max(0.0, x)$

# Activation Functions

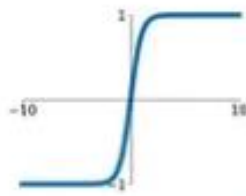
**Sigmoid**

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



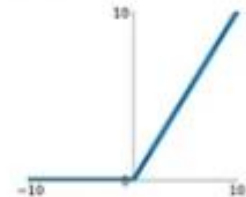
**tanh**

$$\tanh(x)$$



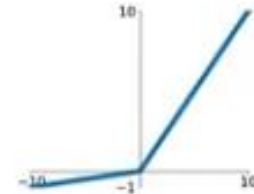
**ReLU**

$$\max(0, x)$$



**Leaky ReLU**

$$\max(0.1x, x)$$

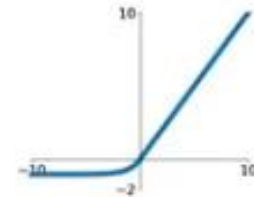


**Maxout**

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

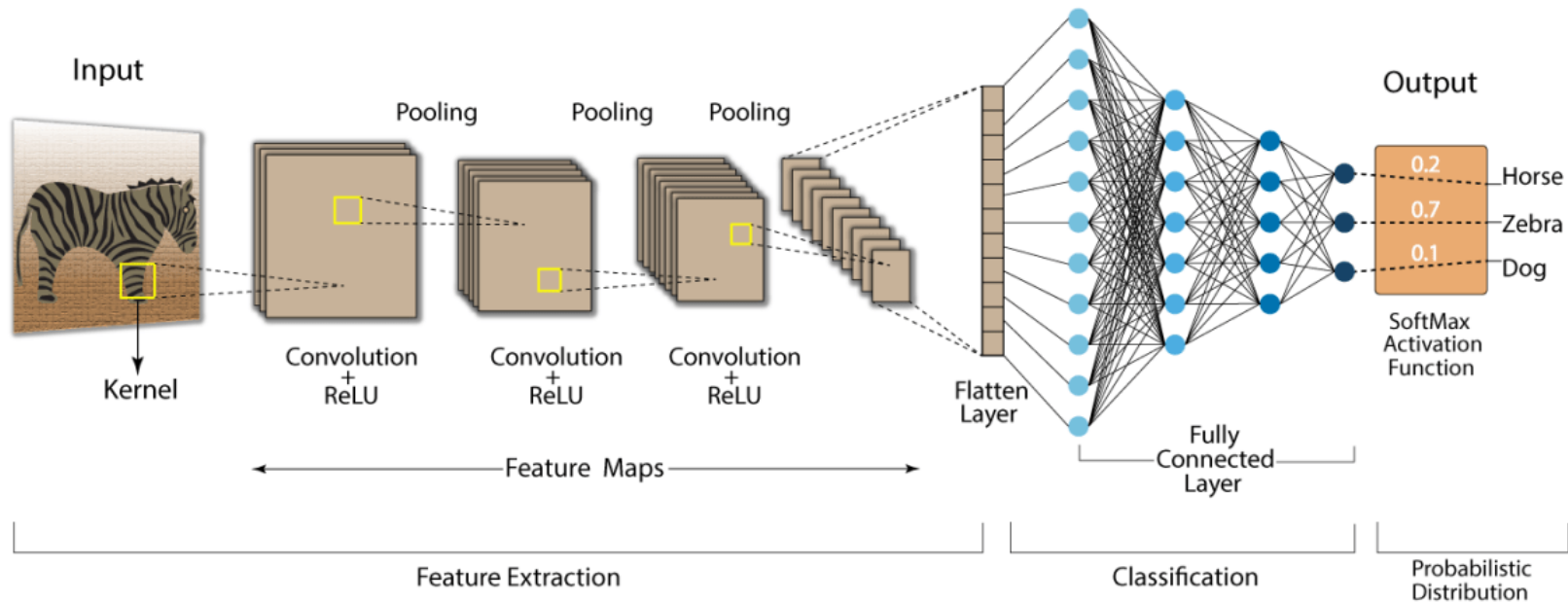
**ELU**

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



Різні функції активації та їх графіки

## Convolution Neural Network (CNN)



# ГЛИБОКЕ НАВЧАННЯ

**Глибке навчання** (також відоме як глибке структуроване навчання, ієрархічне навчання або глибке машинне навчання) — це гілка машинного навчання, заснована на наборі алгоритмів, які намагаються змодельовати високорівневі абстракції в даних за допомогою кількох рівнів обробки зі складними структур або іншим чином, що складається з кількох нелінійних перетворень

# НАВЧАННЯ ГЛИБОКОЇ МЕРЕЖІ

Навчання нейромережі – найскладніша частина глибокого навчання.

- Потрібний великий набір даних.
- Потрібна велика кількість обчислювальної потужності.

Для навчання мережі потрібно подати до неї підготовлені дані та порівняти згенеровані їй вихідні результати з результатами тестового набору даних. Оскільки мережа ще не навчена, результати будуть неправильними.

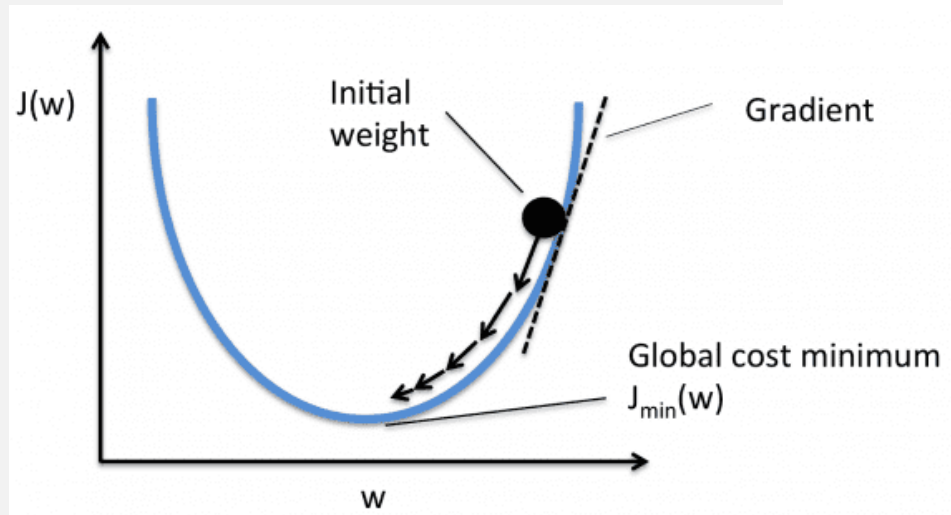
Після пропуску всіх даних можна визначити функцію, яка показуватиме, наскільки результати роботи алгоритму відрізняються від реальних даних. Ця функція називається **функцією втрат**.

В ідеалі потрібно, щоб функція втрат дорівнювала нулю. Це значить, що результати роботи мережі повністю збігаються з результатами тестового набору даних.

# ФУНКЦІЯ ВТРАТ

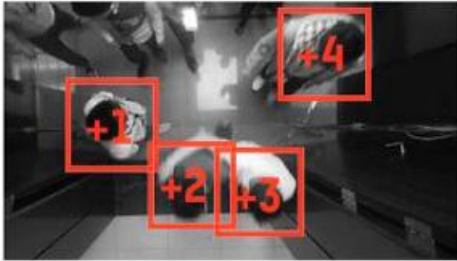
Потрібно змінювати ваги між нейронами. Можна робити це випадковим чином доти, поки функція втрат стане рівної нулю, але це не дуже ефективно.

Можна використовувати метод градієнтного спуску. Градієнтний спуск - це метод, який дозволяє знайти мінімум функції (мінімум функцій втрат).





HIKVISION

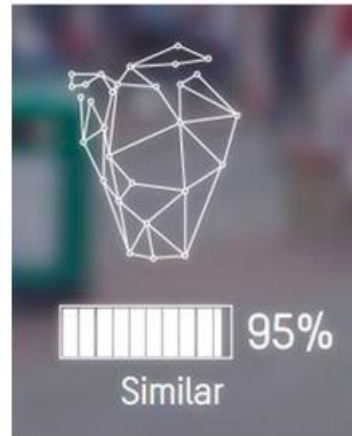


## Технологія глибокого навчання Deep Learning

Виявлення

Розпізнавання

Аналіз



При створенні нейронної мережі необхідно чітко уявляти, яке завдання виконуватиме.

Процес створення складається з кількох етапів:

1. Вибір вхідних та вихідних сигналів нейронної мережі, їх типу та діапазону зміни.
2. Вибір топології нейронної мережі.
3. Підбір структури нейронної мережі.
4. Навчання нейронної мережі.
5. Оцінка якості роботи нейронної мережі.

Якщо після закінчення навчання нейронна мережа не дає задовільного результату, необхідно внести зміни та повторити алгоритм.

# ЗОБРАЖЕННЯ НАБОРУ ДАНИХ CIFAR-10

## Аналіз набору даних:

Набір даних CIFAR-10 містить 60 000 кольорових зображень розміром  $32 \times 32$ , кожне з яких містить один із десяти класів об'єктів, по 6000 зображень на клас. Він складається з 50 000 кольорових навчальних зображень розміром  $32 \times 32$ , розділених на десять категорій, і 10 000 тестових зображень.

## Присутні різні класи:

['літак', 'автомобіль', 'птах', 'кішка', 'олень', 'собака', 'жаба', 'кінь', 'корабель', 'вантажівка']

Кількість зразків різних класів у кожному наборі даних тестування та навчання

```
train_dataset class: {'frog': 5000, 'truck': 5000, 'deer': 5000, 'automobile': 5000, 'bird': 5000, 'horse': 5000, 'ship': 5000, 'cat': 5000, 'dog': 5000, 'airplane': 5000}
```

```
test_dataset class: {'cat': 1000, 'ship': 1000, 'airplane': 1000, 'frog': 1000, 'automobile': 1000, 'truck': 1000, 'dog': 1000, 'horse': 1000, «олень»: 1000, «птах»: 1000}
```

# САМОСТІЙНА РОБОТА

1. Опрацювання теоретичного матеріалу
2. Лабораторна робота