Лекція 1. Нейронні мережі прямого поширення сигналу

кафедра програмної інженерії ЗНУ

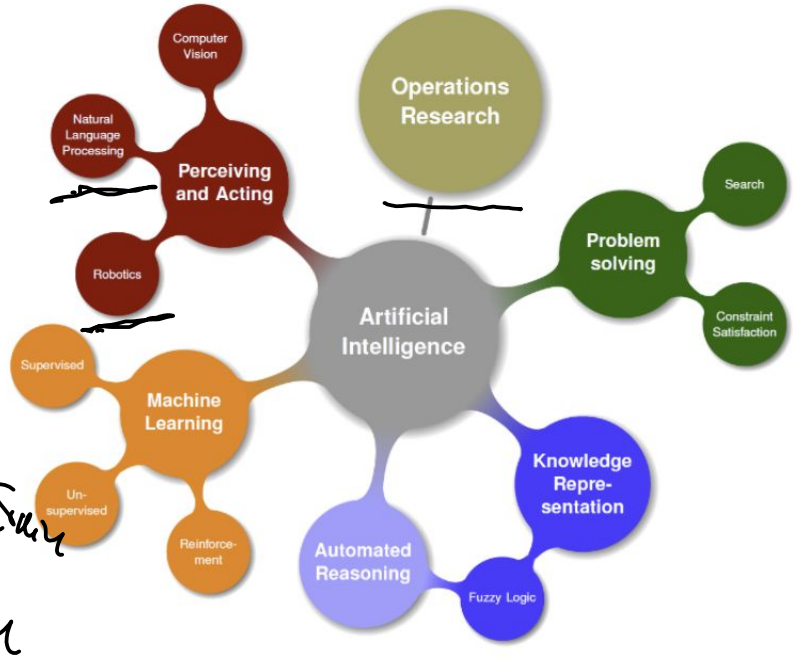
Штучний інтелект

“Artificial intelligence (AI) is an area of computer science that emphasizes the creation of intelligent machines that work and reacts like humans.”

https://levity.ai/blog/general-ai-vs-narrow-ai

https://towardsdatascience.com/what-is-explainable-ai-xai-afc56938d513

https://arxiv.org/search/?query=AI%3A+A+Survey+of+the+State+of+the+Art&searchtype=all&abstracts=show&order=-announced_date_first&size=50



Агенти (для життя)

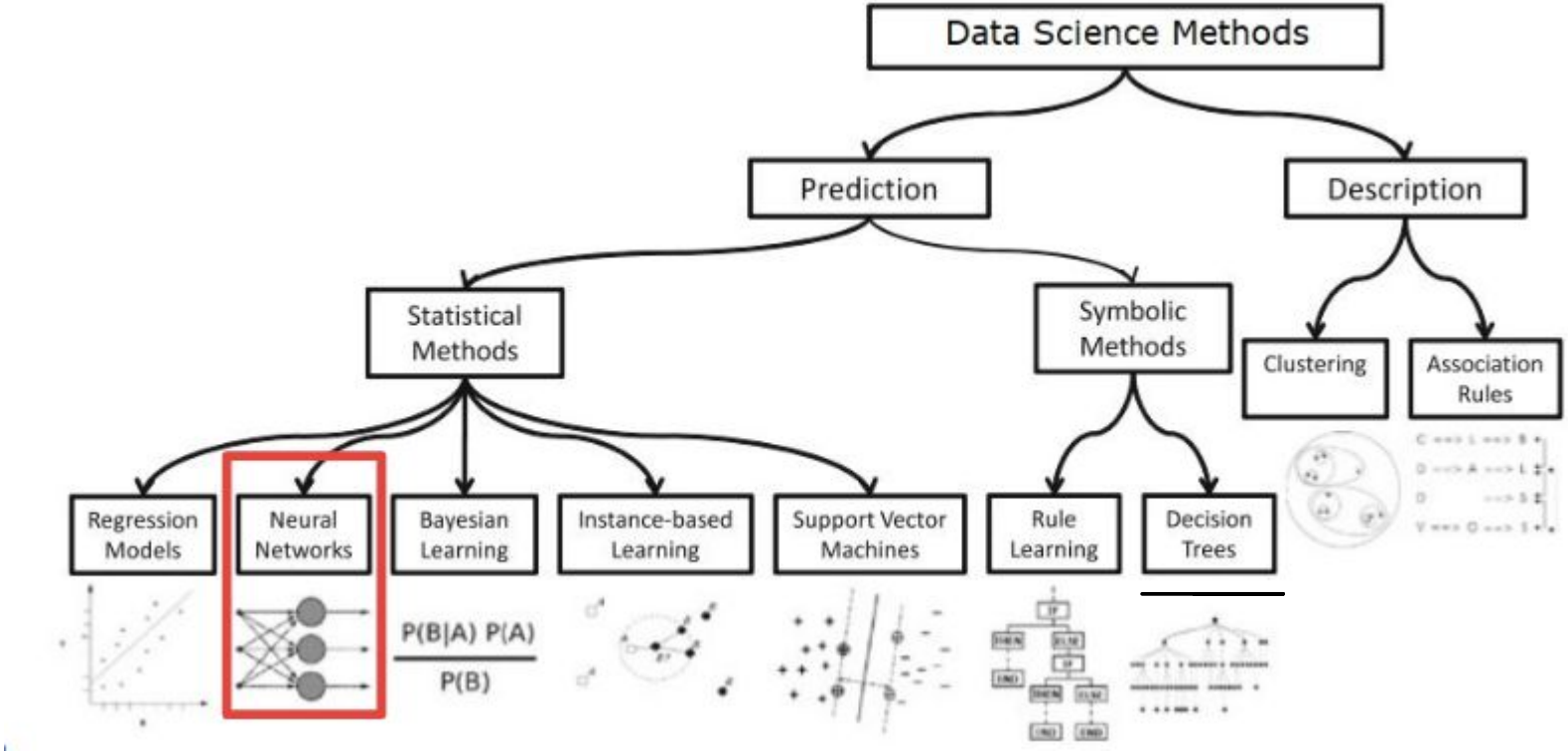
- Еволюційні алгоритми
- Генетичні алгоритми

Машинне навчання vs традиційне програмування

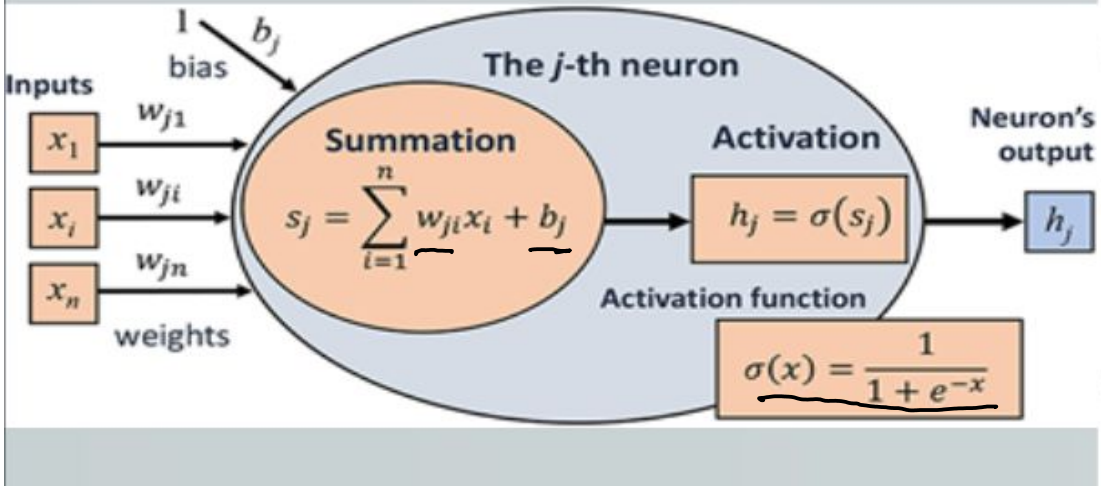
Особливості застосування нейронних мереж

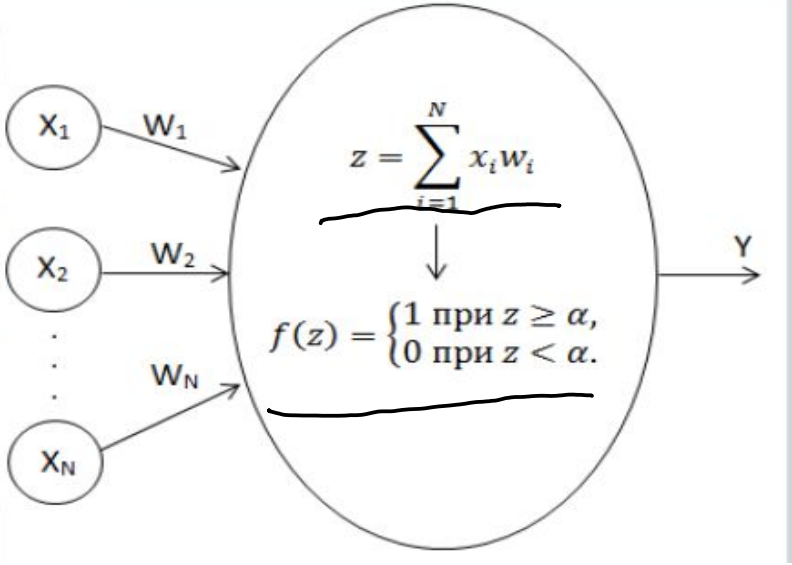
1. Наявність великої кількості відкритих даних (табличних та мультимедійних)
2. Обчислювальні можливості (GPU, TPU)
3. Наявність великої кількості відкритих бібліотек та фреймворків (каркасів розробки) (**Python**, Java)

Типи нейронних мереж



http://pzs.dstu.dp.ua/DataMining/

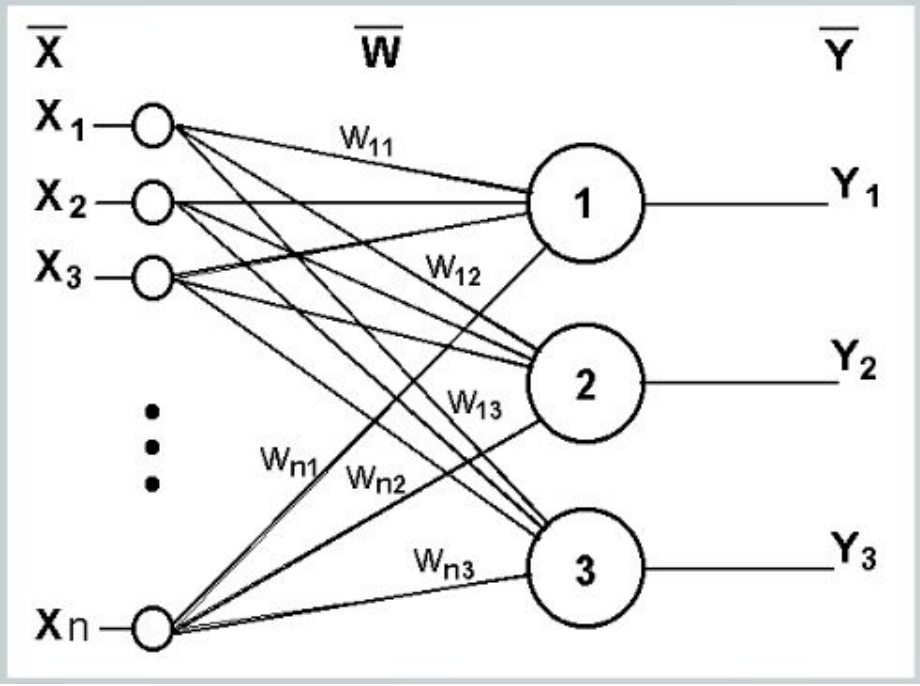


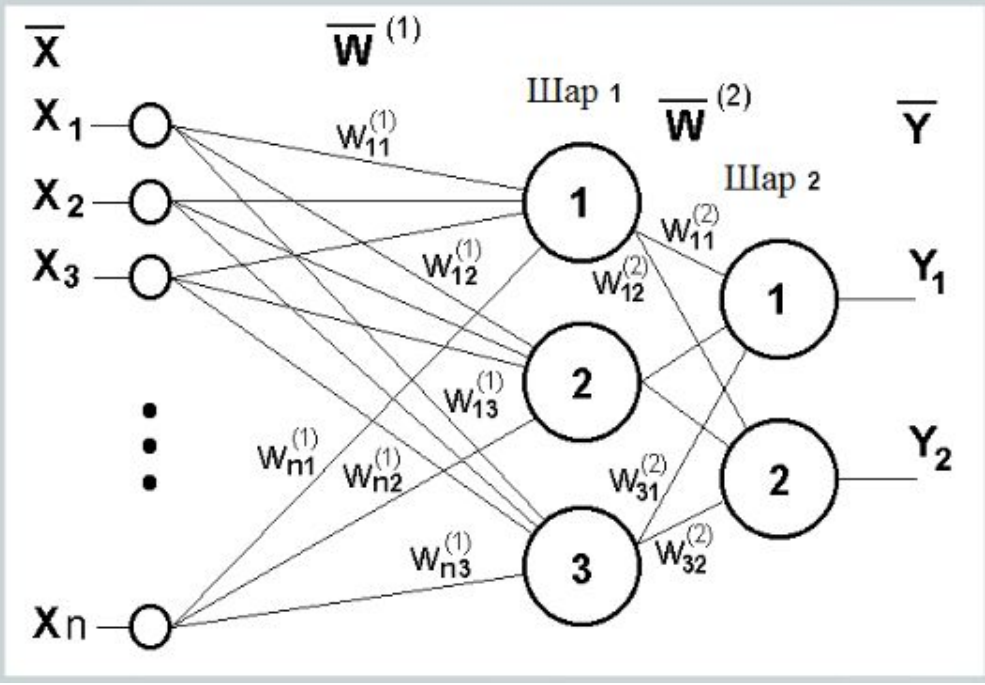


Штучний нейрон (перцептрон) формує вихідний сигнал в залежності від сигналу на вході

Багатошарова нейронна мережа

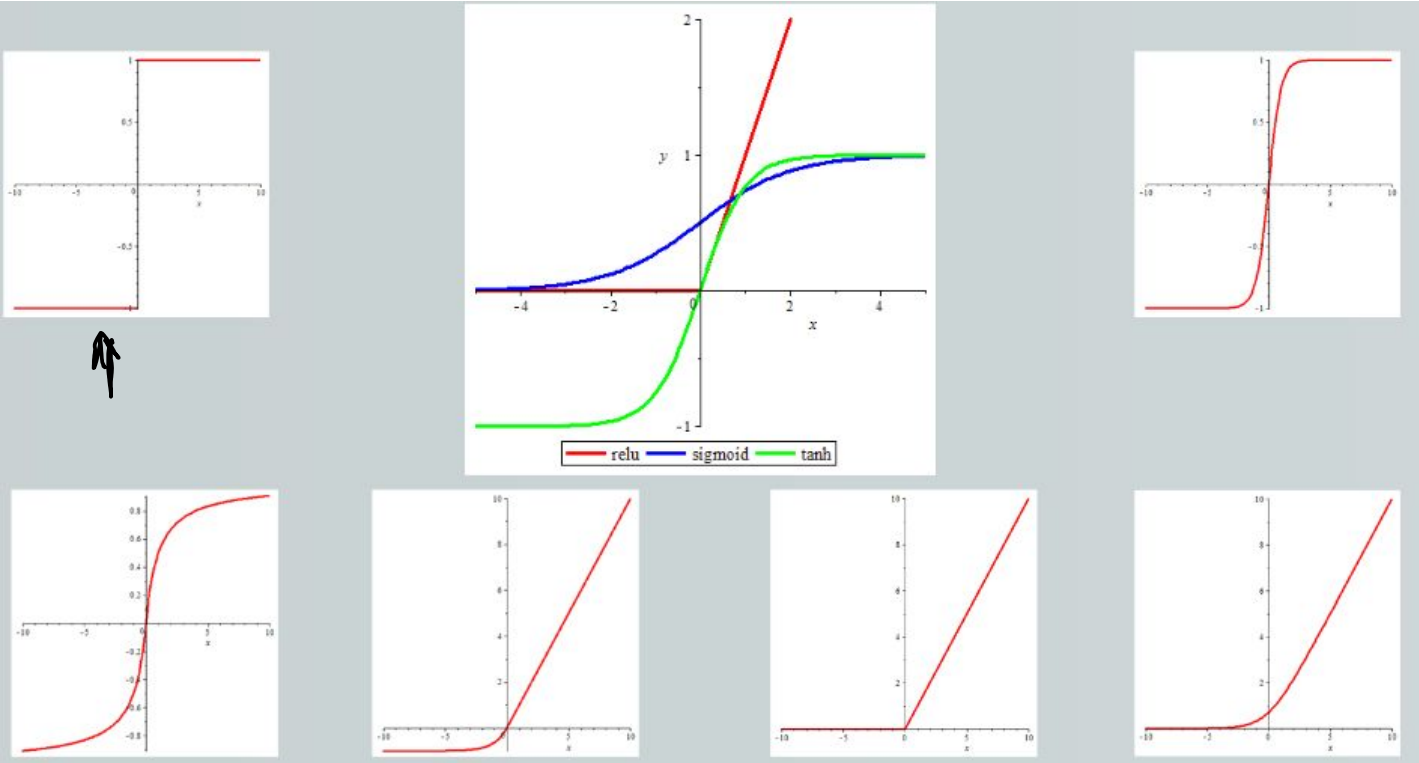
прямого поширення сигнал
багатошар. перцептрон MLP





http://www.asimovinstitute.org/neural-network-zoo/

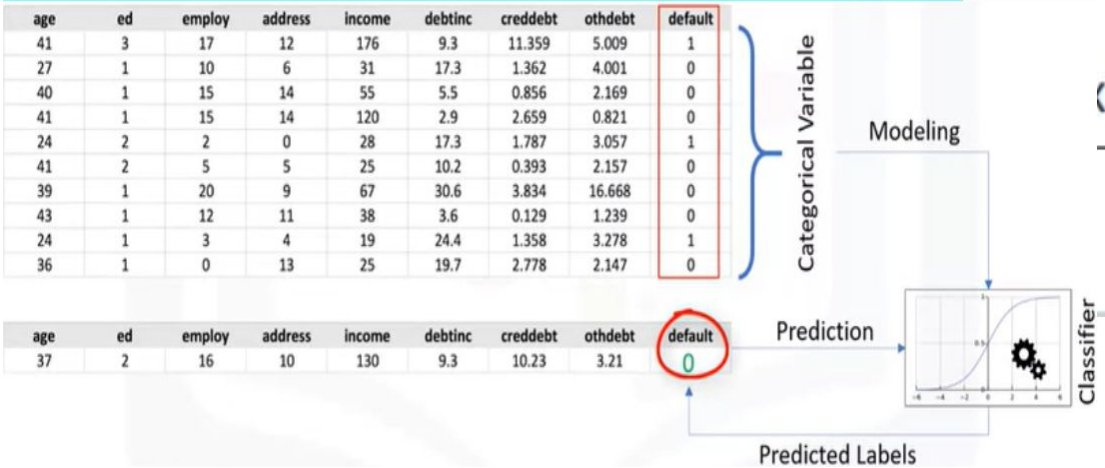
Функції активації

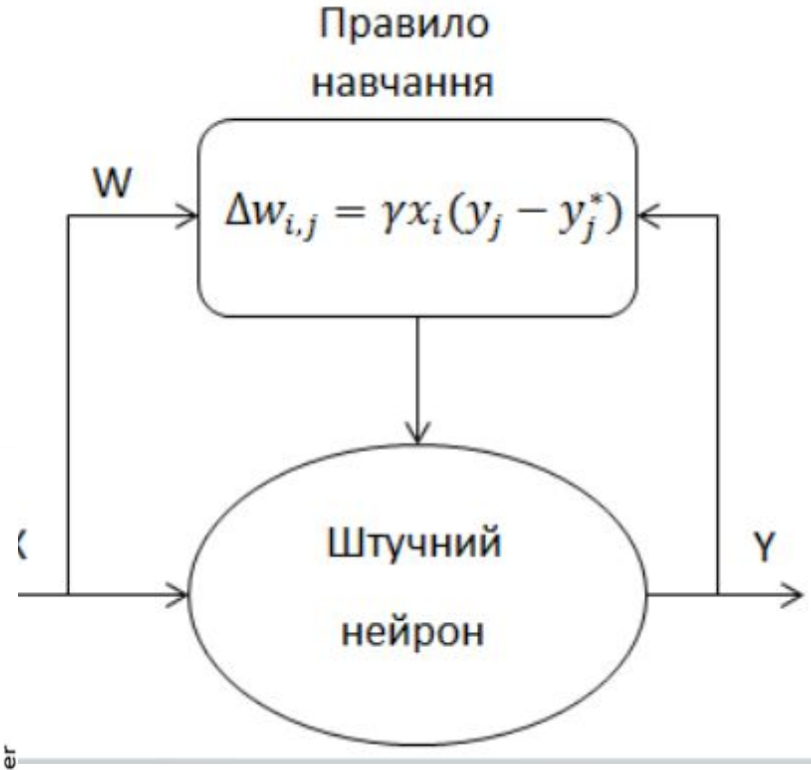


Навчання нейронних мереж

$w_{i,j} = w_{i,j} + delta\_w_{i,j}$

Правило Гебба: при одночасній активації двох нейронів вага їхнього зв'язку зростає



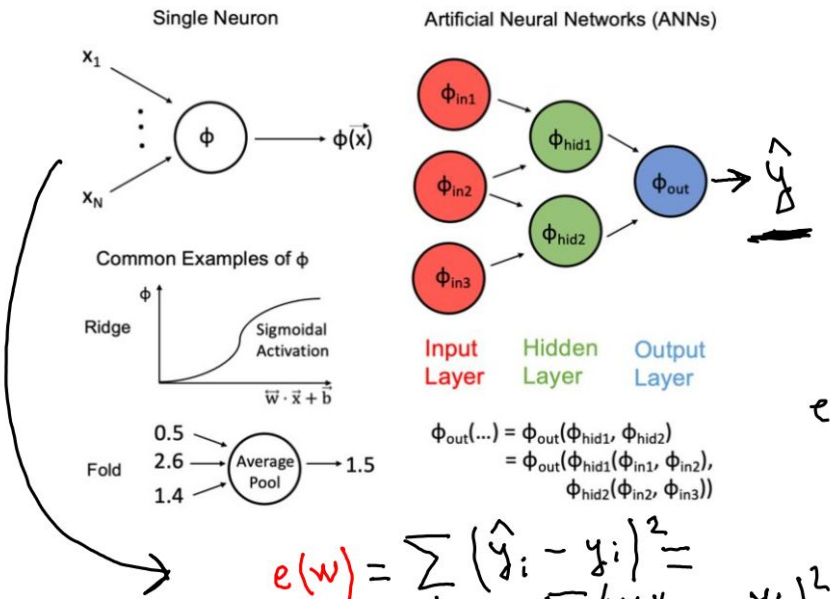


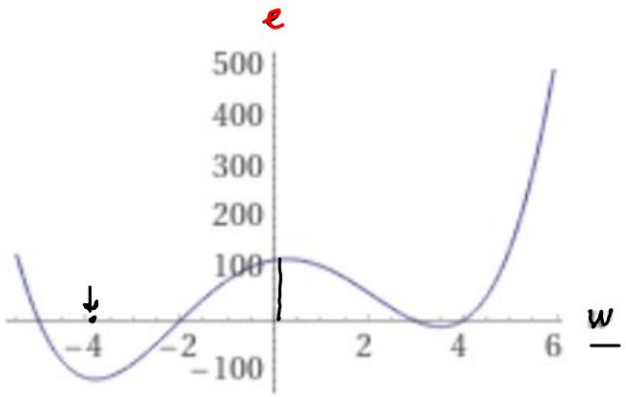
Навчання нейронних мереж

Оптимізація параметрів мережі

Навчальна вибірка

X	$x_1^{(1)} \dots x_m^{(1)}$	$x_1^{(2)} \dots x_m^{(2)}$	$x_1^{(3)} \dots x_m^{(3)}$	…
y	y_1	y_2	y_3	…





$e = w^4 - 27w^2 + 14w + 120$

помилка прогнозування e мінімізується під час навчання

$$e(w) = \sum_i (\hat{y}_i - y_i)^2 = \sum_i (w_i x_i - y_i)^2 \to \min$$

Навчання нейронних мереж

Метод градієнтного спуску оптимізації функції $e(w)$

$e = w^4 - 27w^2 + 14w + 120$

$\frac{de}{dw} = 4w^3 - 54w + 14 \equiv de$

$w_0 = random(\dots)$

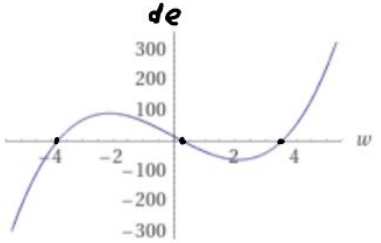
Основна формула градієнтного спуску:

$$w_{n+1} = w_n - \lambda de(w_n)$$

$w_1 = w_0 - \lambda de(w_0)$

де λ – швидкість навчання

n – кількість ітерацій алгоритму (кількість епох навчання)



критичні точки функції $de(w)$:

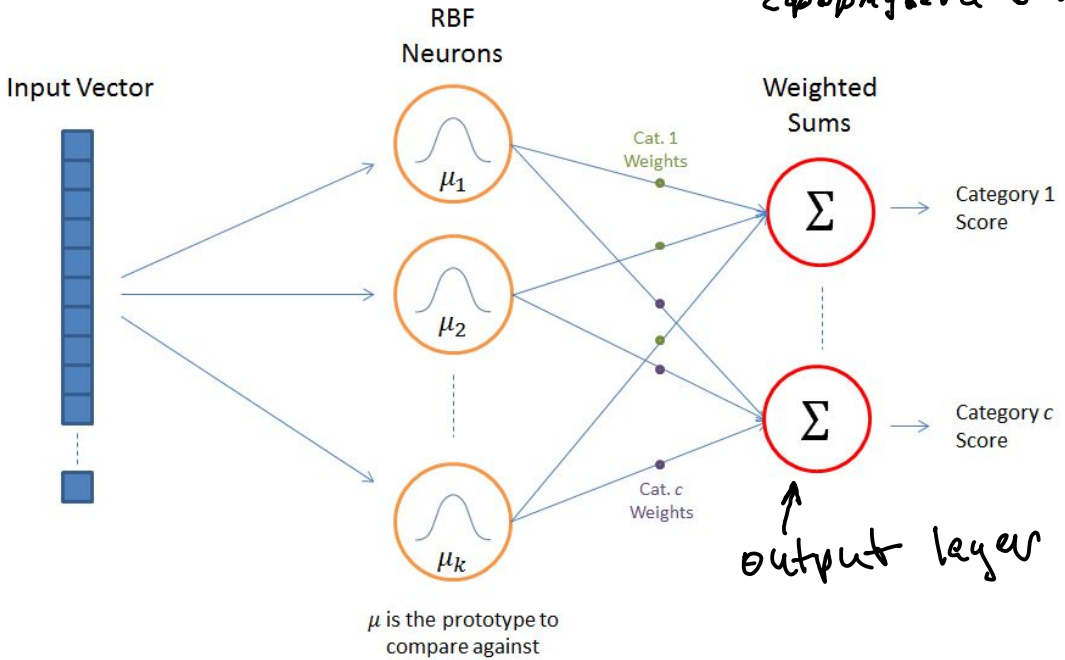
$w \approx -3,7976$ ←

$w \approx 0,2606$

$w \approx 3,5370$

https://colab.research.google.com/drive/1Smr751jgNi4M4qHxkPiNAZ6KSS7TzcFT?usp=sharing
https://uk.wikipedia.org/wiki/%D0%93%D1%80%D0%B0%D0%B4%D1%96%D1%94%D0%BD%D1%82
https://uk.wikipedia.org/wiki/%D0%9F%D0%BE%D1%85%D1%96%D0%B4%D0%BD%D0%B0

Навчання нейронних мереж



- Задача output layer — сформувати вихідний сигнал певної форми
- Суттєва к обчислень не output layer вже не виконується

output layer

https://mccormickml.com/2013/08/15/radial-basis-function-network-rbfn-tutorial/

Особливості RBF мережі

1. Типова структура: 1 input layer – 1 hidden layer – 1 output layer
2. Використання „вектора прототипів" (prototype vector)
3. Мережі прямого поширення та RBF мережі є універсальними апроксиматорами ф-цій

Фреймворк Keras

```
############ Додавання повнозв'язного шару
#model.add(Flatten())
model.add(Dense(20, input_dim =4  , activation = activation))
model.add(Dropout(0.4))
model.add(Dense(15, activation = activation))
model.add(Dropout(0.4))
model.add(Dense(10, activation = activation))
model.add(Dropout(0.4))
model.add(Dense(5, activation = activation))
model.add(Dense(3,kernel_initializer=kernel_initializer,
                bias_initializer=bias_initializer, activation = "softmax"))
############ Компіляція моделі
optimizer = optimizers.Nadam(lr=alpha_zero, beta_1=0.9, beta_2=0.999,
                             epsilon=None, schedule_decay=0.004)
model.compile(loss = "categorical_crossentropy", optimizer = optimizer,
              metrics = ["accuracy"])
history = model.fit(irisX_train, irisY_train, batch_size = batch_size,
                    epochs = nb_epoch, verbose=2, validation_data = (irisX_test, irisY_tes
score = model.evaluate(irisX_test, irisY_test,verbose = 0)
print("score for MLP is ", score[1])
model.summary()
```

Забезпечує можливість будувати нейронну мережу по шарах.

Гіперпараметри нейронних мереж

- Кількість шарів.
- Кількість нейронів в шарах.
- Вид функцій активацій
- Вид оптимізатора
- Тип нейронної мережі
- Тип шару

Для визначення нейронної мережі необхідно

Визначити тип нейромережі

Визначити гіперпараметри

Визначити функцію втрат

Підготувати дані

Приклади

https://www.kaggle.com/code/avk256/neural-network-approach-to-iris-dataset

Задача №1

Варіант 1.
Розглядається один штучний нейрон. Вага зв'язку w_1=0.3; зсув b_1=0.2. Функція активації - ReLu. Вихід нейрону - 0.38. Визначити вхідний сигнал нейрону.

Варіант 2.
Розглядається один штучний нейрон. Вхід нейрону 1.2; вага зв' язку w_1=0.4; зсув b_1=0.1. Функція активації - tanh. Визначити вихідний сигнал нейрону.

Варіант 3.
Розглядається один штучний нейрон. Вхід нейрону 2.3. Вихід нейрону 1. Функція активації - Heaviside step function. Яке значення ваги та зсуву є допустимим?