

ЛАБОРАТОРНА РОБОТА №2

Тривалість роботи – 2 години.

Функції налаштування шарів нейронів

Функції даної групи є допоміжними при роботі з деякими розглянутими функціями навчання НМ (наприклад, **trainwb**, **trainwb1**, **adaptwb**), а також використовуються при налаштуваннях одношарових НМ (персептронів, шарів Кохонена і т. п.).

[dB,LS] = learncon(B,P,Z,N,A,T,E,gW,gA,D,LP,LS) – функція налаштування вагів з введенням «відчуття справедливості».

Аргументи:

B – $S \times 1$ вектор зсувів;

P – $1 \times Q$ вхідний вектор;

Z – $S \times Q$ матриця зважених входів;

N – $S \times Q$ матриця входів;

A – $S \times Q$ матриця вихідних векторів;

T – $S \times Q$ матриця цільових векторів;

E – $S \times Q$ матриця помилок;

gW – $S \times R$ градієнт критерію ефективності по відношенню до вектора вагів;

gA – $S \times Q$ градієнт критерію ефективності по відношенню до вектора виходу;

D – $S \times S$ матриця відстаней між нейронами;

LP – параметр навчання, $LP = []$;

LS – стан навчання, на початку - $[]$;

Величини, що повертаються:

dB – $S \times 1$ вектор змін вагів (або зсувів);

LS – новий стан навчання.

Функція у формі **learncon(code)** повертає наступну інформацію:

при аргументі '**pnames**' – імена параметрів навчання,

при '**pdefaults**' – значення параметрів за умовчанням,

при '**needg**' – 1, якщо ця функція використовує **gW** або **gA**.

Алгоритм виконання функції спочатку обчислює «відчуття справедливості» нейрона по виразу $c = (1-lr)*c + lr*a$, а вже потім коректує вагу відповідно до формули

$$b = \exp(1-\log(c)) - b.$$

Приклад

```
» a = rand(3,1);
```

```
» b = rand(3,1);
```

```
» lr=0.5; % Задання параметру навчання
```

```
» dW = learncon(b,[],[],[],a,[],[],[],[],[],lr,[])
```

```
dW =
```

```
0.3449
```

0.7657

0.5405

learngd – функція корекції вагів і зсувів, яка реалізує градієнтний алгоритм оптимізації.

Запис: $[dW,LS] = \text{learngd}(W,P,Z,N,A,T,E,gW,gA,D,LP,LS)$ $[db,LS]=$
 $= \text{learngd}(b,\text{ones}(1,Q),Z,N,A,T,E,gW,gA,D,LP,LS)$
info = **learngd**(code)

Опис. Аргументи функції: **W** – матриця вагів або вектор зсуву, решта аргументів – як у попередньої функції.

Learngdm – функція практично аналогічна попередній, але використано інший алгоритм оптимізації – градієнтний метод з інерційною складовою.

$[dW,LS] = \text{learnh}(W,P,Z,N,A,T,E,gW,gA,D,LP,LS)$ – функція корекції вагів, що використовує правило Хебба.

$[dW,LS] = \text{learnhd}(W,P,Z,N,A,T,E,gW,gA,D,LP,LS)$ – функція реалізує модифікацію правила Хебба.

$[dW,LS] = \text{learnis}(W,P,Z,N,A,T,E,gW,gA,D,LP,LS)$ – функція підстроювання вагів «вхідної зірки» (нейрона Гроссберга).

$[dW,LS] = \text{learnk}(W,P,Z,N,A,T,E,gW,gA,D,LP,LS)$ – функція підстроювання вагів Кохонена.

$[dW,LS] = \text{learnlv1}(W,P,Z,N,A,T,E,gW,gA,D,LP,LS)$ і

$[dW,LS] = \text{learnlv2}(W,P,Z,N,A,T,E,gW,gA,D,LP,LS)$ – функції налаштування мереж зустрічного розповсюдження.

$[dW,LS] = \text{learnos}(W,P,Z,N,A,T,E,gW,gA,D,LP,LS)$ – функція налаштування нейрона типу «вихідна зірка».

$[dW,LS] = \text{learnp}(W,P,Z,N,A,T,E,gW,gA,D,LP,LS)$ – функція, що реалізує алгоритм навчання перцептрона.

$[dW,LS] = \text{learnpn}(W,P,Z,N,A,T,E,gW,gA,D,LP,LS)$ – те ж, що і попередня функція, але з нормалізацією входів. Ефективніша при великих амплітудних змінах вхідних сигналів.

$[dW,LS] = \text{learnsom}(W,P,Z,N,A,T,E,gW,gA,D,LP,LS)$ – функція навчання карт, що самоорганізуються.

$[dW,LS] = \text{learnwh}(W,P,Z,N,A,T,E,gW,gA,D,LP,LS)$ – функція навчання, що реалізує так званий алгоритм Уїдроу-Хоффа (Widrow-Hoff).

Завдання

1. Реалізувати у *Neural Networks Toolbox* наведені у прикладах функції навчання шарів нейронів НМ.
2. Згідно завдання викладача реалізувати 3 функції навчання шарів нейронів НМ.

Функції одновимірної оптимізації та ініціалізації шарів і зсувів Функції цієї групи можна розглядати як допоміжні для функцій навчання нейронних мереж. Щодо оптимізації, то вони реалізують ряд алгоритмів одновимірного пошуку.

srchbac – функція реалізує так званий алгоритм перебору з поверненнями (backtracking).

srchbre – функція реалізує комбінований метод оптимізації, який об'єднує метод золотого перетину і квадратичну інтерполяцію.

srchcha – функція реалізує різновид методу оптимізації із застосуванням кубічної інтерполяції.

srchgol – функція реалізує метод золотого перетину.

srchhyb – функція реалізує комбінований метод оптимізації, який об'єднує метод дихотомії і кубічну інтерполяцію.

Для багатомірних нейронних мереж етапом, що передує процедурі їх навчання, є етап ініціалізації – задання деяких, зазвичай вибраних випадковим чином, вагів і зсувів мережі. Така ініціалізація виконується за допомогою функцій відповідної групи команд.

initcon(S,PR) – функція, що встановлює зсуви нейронів залежно від середнього виходу нейрона. *Аргументи:* *s* – кількість нейронів, *PR* = [*P*_{min} *P*_{max}] – матриця (з двома стовпцями) мінімальних і максимальних значень входів, за умовчанням [1 1]. Повертає вектор зсувів. Використовується спільно із командою з командою *learncon*.

Приклад

» ***b* = initcon(3)**

***b* =**

8.1548

8.1548

8.1548

initzero – функція задання нульових початкових значень вагам або зсувам. Аргументи ті ж, що і у попередньої команди.

midpoint(S,PR) – функція ініціалізації, що встановлює ваги відповідно до середніх значень входів.

randnc(S,R) – функція задання матриці вагів. Повертає матрицю розміру *S*×*R* з випадковими елементами, нормалізовану за стовпцями (вектори-стовпці мають одиничну довжину).

randnr(S,R) – те ж, що попередня функція, але повертає матрицю вагів, нормалізовану за рядками.

rands – функція ініціалізації вагів/зсувів заданням їх випадкових значень з діапазону [-1, 1].

Запис:

***W* = rands(S,PR)**

***M* = rands(S,R)**

***v* = rands(S)**

Опис. Аргументи ті ж, що і для функції **initcon**; значення *R* за умовчанням - 1. Повертається матриця відповідного розміру.

Завдання

1. Реалізувати у *Neural Networks Toolbox* наведений приклад встановлення зсувів.
2. Згідно завдання викладача реалізувати 3 функції оптимізації/ініціалізації шарів НМ.