Michał Przysucha

Kacper Radzikowski

WMH – projekt

PB2. Zastosowanie sieci neuronowej w zadaniu aproksymacyjnym

Wstępne ustalenia:

1. Język programowania: Python

2. Dane uczące i testujące: wygenerowane z nietrywialnej funkcji

3. Etapy:

- implementacja sieci w różnych strukturach (liczba warstw ukrytych oraz liczba neuronów w warstwie)

- uczenie sieci, w tym zbadanie procesu uczenia się sieci

- określenie optymalnej struktury

Oznaczenia:

- liczba wartsw w sieci neuronowej

- indeks warstw trenujących

- liczba węzłów w l-tej warstwie sieci neuronowej, w szczególności:

- liczba węzłów w 0-wej warstwie (warstwie wejściowej) sieci neuronowej (liczba cech danych trenujących – w naszym zadaniu )

- liczba danych trenujących

- indeks danych trenujących

- wartość na wyjściu l-tej warstwy (po funkcji aktywacji) dla i-tej danej trenującej. W rzeczywistości implementacja będzie opierać się na macierzach, zatem skorzystamy z notacji jak niżej:

- macierz wyjść l-tej warstwy dla wszystkich danych trenujących

- wartość na wyjściu wielomianiu w l-tej warstwie sieci neuronowej (przed funkcją aktywacji) dla i-tej danej trenującej. Tu również implementacja będzie używać macierzy, zatem skorzystamy z poniższej notacji:

- macierz wyjść wielomianiu w l-tej warstwie sieci neuronowej (przed funkcją aktywacji) dla wszystkich danych trenujących

- funkcja aktywacji w l-tej warstwie ()

- odpowiednio: wagi i wyraz wolny (bias) w l-tej warstwie sieci neuronowej (współczynniki wielomianu wykorzystane do policzenia)

- wartość na wejściu sieci neuronowej dla i-tej danej trenującej  
- wartość na wejściu sieci neuronowej (macierz wejść wszystkich danych trenujących lub testowych)

- wyjście L-tej warstwy (wyjściowej) sieci neuronowej (predykcja) dla i-tej danej trenującej

- wartość na wyjściu L-tej warstwy (wyjściowej) sieci neuronowej (predykcja), jest to macierz wyjść wszystkich danych trenujących.

Wymiary macierzy i wektorów:

, - liczba wierszy jest taka sama jak liczba węzłów w l-tej warstwie (wymiar wektora wyjściowego z l-tej warstwy dla pojedynczej danej trenującej), liczba kolumn jest taka sama jak liczba danych trenujących.

Implementacja: obie zmienne będą zatem macierzami o wymiarach:.

- macierz współczynników, liczba wierszy jest taka sama jak liczba węzłów w l-tej warstwie (wymiar wektora wyjściowego z l-tej warstwy dla pojedynczej danej trenującej), liczba kolumn jest taka sama jak liczba węzłów w (l-1)-tej warstwie (wymiar wektora wyjściowego z (l-1)-tej warstwy / wejściowego do l-tej warstwy).

Implementacja: będzie słownikiem / tablicą asocjacyjną, gdzie kluczem będzie(numer warstwy sieci neuronowej), a wartością macierz .

- wektor wyrazów wolnych, liczba wierszy jest taka sama jak liczba węzłów w l-tej warstwie (wymiar wektora wyjściowego z l-tej warstwy)

Implementacja: będzie słownikiem / tablicą asocjacyjną, gdzie kluczem będzie(numer warstwy sieci neuronowej), a wartością wektor .

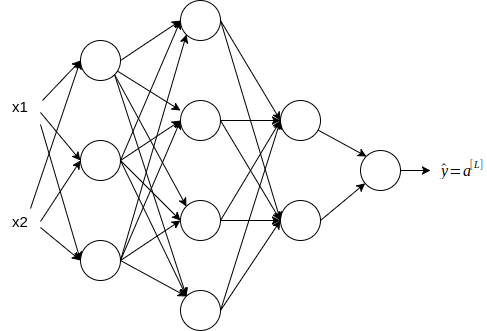
- w niniejszym zadaniu pojedynczy przykład danej trenującej jest wektorem dwuwymiarowym (dwuelementowym), natomiast będzie macierzą, gdzie wiersze to cechy zmiennej wejściowej, a kolumny to kolejne przykłady danych trenujących, czyli będzie to macierz o wymiarach: .

- macierz o 1 wierszu (dla pojedynczej danej wejściowej, wyjście sieci neuronowej w naszym zadaniu to liczba rzeczywista) oraz o kolumnach, czyli: .

- wyjście wielomianu w l-tej warstwie (przed funkcją aktywacji) dla wszystkich danych trenujących, liczba wierszy jest taka sama jak liczba węzłów w l-tej warstwie (wymiar wektora wyjściowego z l-tej warstwy dla pojedynczej danej trenującej lub testującej), a liczba kolumn jest taka sama jak liczba danych trenujących lub testujących: .

- wyjście funkcji aktywacji w l-tej warstwie, liczba wierszy jest taka sama jak liczba węzłów w l-tej warstwie, a liczba kolumn odpowiada liczbie danych trenujących lub testujących: .

Przykład:



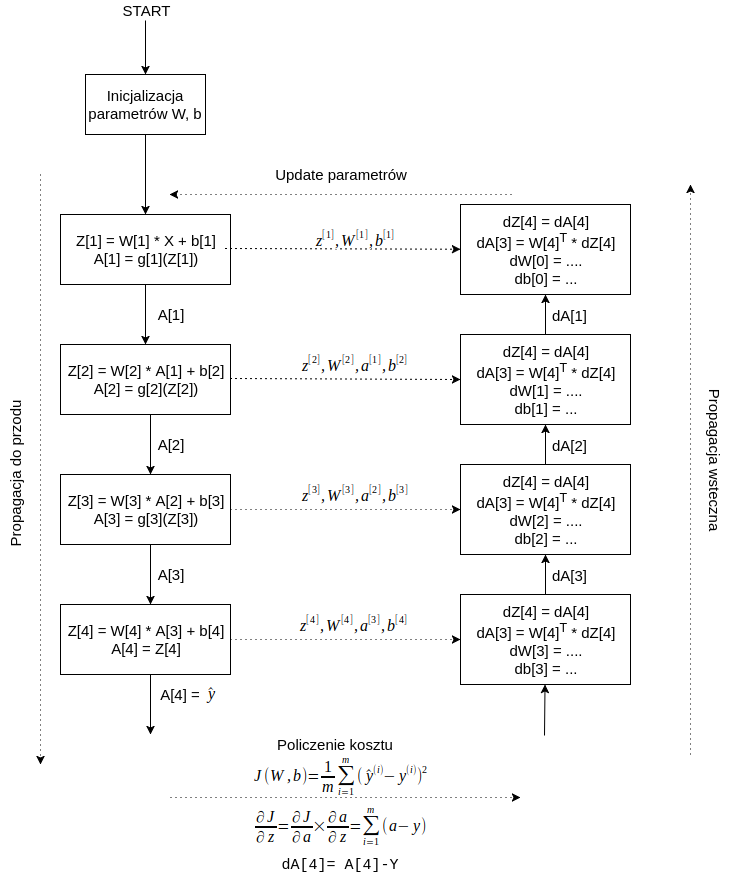
W powyższej sieci neuronowej mamy 3 warstwy ukryte oraz warstę wyjściową, zatem L = 4 (warstwy wejściowej nie liczymy). W poszczególnych warstwach mamy:

Zmienna wejściowa jest dwuwymiarowa (dwuelementowy wektor), zatem .

Wymiary macierzy ze współczynnikami:

Wymiary wektorów z wyrazami wolnymi:

Implementacja:



dA[4] = A[4]-Y