

Data oddania: _____

Ocena: _____

Mateusz Walczak 216911

Konrad Kajszczyk 216790

Zadanie 2: Sieć neuronowa służąca do korygowania pomiaru systemu lokalizacji

Wprowadzenie

Celem zadania było zaprojektowanie i zaimplementowanie sieci neuronowej, która pozwoli na korygowanie błędów uzyskanych z systemu pomiarowego. Projektując sieć neuronową należało odpowiednio dobrać [1]:

- liczbę warstw,
- liczebność neuronów w poszczególnych warstwach,
- funkcje aktywacji,
- liczbę próbek z poprzednich chwil czasowych.

1. Opis architektury sieci neuronowej

W tym rozdziale rozpoczniemy od opisu działania naszego programu oraz drogi jaką przebyliśmy, aby znaleźć taką sieć neuronową, która pozwoli na skuteczne korygowanie błędów uzyskanych z systemu pomiarowego. Następnie skoncentrujemy się na opisie architektury tej sieci neuronowej, która okazała się najskuteczniejsza.

1.1. Historia wyboru odpowiedniej architektury sieci neuronowej

Nasz program został napisany w języku *Java*, bez wykorzystania wysokopoziomowych bibliotek do tworzenia sieci neuronowych. Program został napisany w taki sposób, aby w zależności od ustawień, tworzyć, inicjować a

następnie przeprowadzać proces nauki dla sieci neuronowych o różnych liczbach neuronów w poszczególnych warstwach a także różnych liczbach próbek z poprzednich chwil czasowych.

Po wielu nieudanych próbach poprawy błędów uzyskanych z systemu pomiarowego, z wykorzystaniem sieci 2-warstwowych (n neuronów w 1 warstwie i 2 neurony w warstwie wyjściowej), zdecydowano się na implementację 3-warstwowej sieci neuronowej.

Nasza aplikacja buduje 3-warstwową sieć neuronową na podstawie trzech parametrów, które na potrzeby tego sprawozdania będziemy nazywać n_1 , n_2 oraz p . Kolejno, stanowią one:

- n_1 - liczba neuronów w pierwszej warstwie sieci,
- n_2 - liczba neuronów w drugiej warstwie sieci,
- p - liczbę próbek z poprzednich chwil czasowych wykorzystywanych przez sieć neuronową.

W tym miejscu warto wspomnieć o tym, że trzecia warstwa za każdym razem składała się z 2 neuronów, ponieważ nasza sieć musi mieć 2 wyjścia, aby spełniała warunki zadania (powinna zwracać współrzędną x-ową i y-ową dla danej próbki pomiarowej).

Warstwa trzecia (wyjściowa) posiadała identycznościową funkcję aktywacji. Warstwy pierwsza i druga zaś funkcję hiperboliczną (tangens hiperboliczny), określoną wzorem:

$$\operatorname{tgh} x = \frac{\sinh x}{\cosh x} = \frac{e^x - e^{-x}}{e^x + e^{-x}}. \quad (1)$$

Wszystkie wagi, dla każdego z neuronów we wszystkich 3 warstwach, inicjalizowano wartością losową, należącą do przedziału $\langle -1; 1 \rangle$.

Program uruchamiano dla różnych kombinacji parametrów n_1 , n_2 oraz p . W każdej z kombinacji powtarzano eksperymenty kilkakrotnie, w poszukiwaniu najbardziej optymalnego rozwiązania. Wartości osiągnęte przez poszczególne parametry każdorazowo należały do zbiorów liczb całkowitych spełniających warunki, odpowiednio:

$$2 \leq n_1 \leq 20, \quad (2)$$

$$2 \leq n_2 \leq 20, \quad (3)$$

$$1 \leq p \leq 10. \quad (4)$$

Na podstawie tysięcy iteracji programu, przeprowadzonych na przestrzeni kilkudziesięciu godzin, wyciągnięto wnioski dotyczące tego, jakie wartości

parametrów n_1 , n_2 oraz p są optymalne dla zadanego problemu. Okazało się, że:

- najbardziej optymalne liczby neuronów zarówno w warstwie pierwszej jak i drugiej to te, należące do przedziału

$$n_1, n_2 \in \langle 6; 9 \rangle. \quad (5)$$

- liczby próbek z poprzednich chwil czasowych, dających najlepsze rezultaty są następujące:

$$p = 3 \vee p = 4 \vee p = 5 \quad (6)$$

W ten sposób bardzo zawęziliśmy różnorodność sieci neuronowych, wykorzystywanych do naszych eksperymentów. W następnym etapie badań, wykorzystywano już tylko takie sieci neuronowe, które stosowały się do powyższych wniosków. Program ponownie uruchomiono wiele razy. Tym razem jednak, regularnie udawało się osiągnąć zamierzony cel - nauczona sieć neuronowa znacząco korygowała błędy uzyskane z systemu pomiarowego.

Architektura sieci neuronowej, z wykorzystaniem której uzyskano najlepsze rezultaty została omówiona w następnym podrozdziale.

1.2. Najskuteczniejsza sieć neuronowa - szczegóły architektury

W wielu iteracjach, z wykorzystaniem różnych konfiguracji sieci (spełniających warunki (5) oraz (6)), udawało się nam korygować błędy uzyskane z systemu pomiarowego, a co za tym idzie poprawić dystrybucję błędów pomiarowych, a także zmniejszyć średni błąd pomiaru dla zbioru testowego.

Najlepsze wyniki zarówno dystrybuanty jak i średniego błędu pomiarowego uzyskano dla 3-warstwowej sieci neuronowej, której parametry prezentują się następująco:

- liczebność neuronów w poszczególnych warstwach: $n_1 = 6$ i $n_2 = 7$,
- liczba próbek z poprzednich chwil czasowych $p = 3$
- funkcje aktywacji: 1 i 2 warstwa - funkcja hiperboliczna, 3 warstwa - funkcja identycznościowa.

Omawiana sieć neuronowa posiada 8 wejść do każdego neuronu - przyjmuje 4 próbki - aktualną oraz 3 poprzednie (każda próbka składa się z dwóch wartości - x-owej i y-owej). Sieć nie posiada biasu, a co za tym idzie w pierwszej warstwie każdy neuron ma dokładnie 8 wag. Na podstawie powyższych rozważań należy zauważyć, że funkcja realizowana przez omawianą sieć jest funkcją 8 zmiennych, zwracającą w wyniku 2 wartości.

Zestaw wag nauczonej sieci neuronowej zestawiono w poniższych tabelach.

2. Opis algorytmu uczenia sieci neuronowej

3. Porównanie dystrybuant błędu pomiaru

4. Kod źródłowy programu

Literatura

[1] Treść zadania drugiego

<https://ftims.edu.p.lodz.pl/mod/page/view.php?id=73137>.