

# Sprawozdanie

## Modelowanie metodą sztucznych sieci neuronowych

Amadeusz Filipek

Laboratorium komputerowe WFILS AGH

### 1. Wstęp

Celem ćwiczenia jest wykonanie modelowych obliczeń przebiegu czasowego stężenia trytu w Dunaju za pomocą sztucznych sieci neuronowych.

Sztuczna sieć neuronowa to rodzaj matematycznego modelu, który ma za zadanie procesować informacje. Sieci neuronowe wchodzi w skład dziedziny nauki określonej jako nauczanie maszynowe (machine learning). Sieć neuronowa zbudowana jest z neuronów ułożonych w kolejne warstwy i połączonych ze sobą. Każdy neuron sumuje wartości wejściowe z odpowiednimi wagami, przepuszcza je przez funkcję przejścia i wynik przekazuje dalej. Sieć neuronowa wykorzystana w ćwiczeniu posiada trzy warstwy: wejściową o ilości neuronów odpowiadającej ilości danych wejściowych, ukrytą z 5 neuronami oraz wyjściową z jednym neuronem. Pierwsza warstwa sieci posiada liniową funkcję przejścia. Druga warstwa posiada tansigmoidalną funkcję przejścia:

$$y(x) = \frac{2}{1 + \exp(-2x)} - 1$$

Neuron w ostatniej warstwie posiada logsigmoidalną funkcję przejścia postaci:

$$y(x) = \frac{1}{1 + \exp(-x)}$$

Krytycznym punktem pracy sieci neuronowych jest proces nauczania sieci. W tym celu korzysta się z algorytmów, które dopasowują współczynniki wagowe każdego neuronu tak aby wartości wyjściowe całej sieci były równe danym na których sieci są trenowane. Pracę sieci można zweryfikować korzystając z danych, których sieć nie widziała w procesie trenowania. Jako miarę błędu pracy sieci można wykorzystać parametr RMSE (round mean squared error):

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=0}^N (y_{out}^{calc}(t_i) - y_{out}^{exp}(t_i))^2}$$

Gdzie N – stanowi ilość porównywanych danych,  $y_{out}^{calc}$  – stanowi wartość obliczoną przez sieć oraz  $y_{out}^{exp}$  – wartość oczekiwaną.

Dane wejściowe do sieci stanowią dane wykorzystane w poprzednim ćwiczeniu dotyczącym modelowania stężenia trytu w Dunaju metodą pudełkową. Dane zostały podzielone na zestaw uczący oraz testujący sieć w stosunku 70:30 %. Dane do sieci podawane są w postaci wektora o określonej długości.

## 2. Algorytm sieci neuronowej

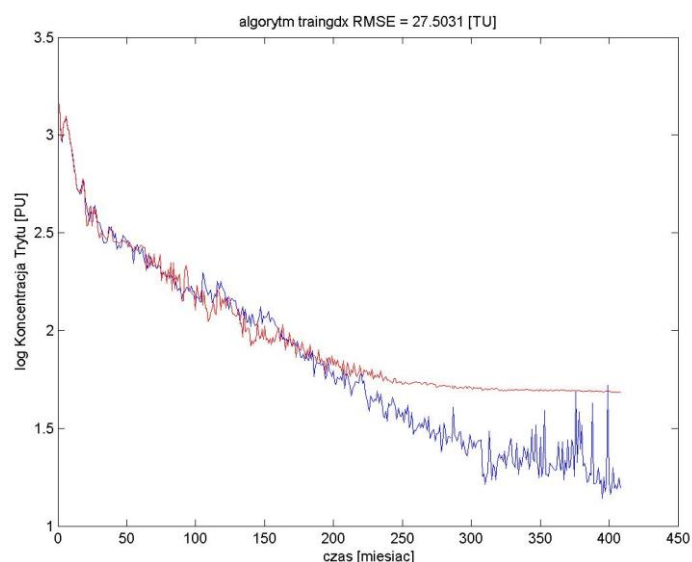
Zbadałem jak różni się wynik sieci neuronowej w zależności od wykorzystanego parametru oraz długości wektora danych wejściowych. W ćwiczeniu porównywane są dwa algorytmy traingdx oraz trainlm. Uzyskane wartości błędów dla obu metod przedstawione są w poniższej tabeli:

*Tabela 1. Uzyskane wartości błędu dla obu algorytmów przy różnych szerokościach wejściowych*

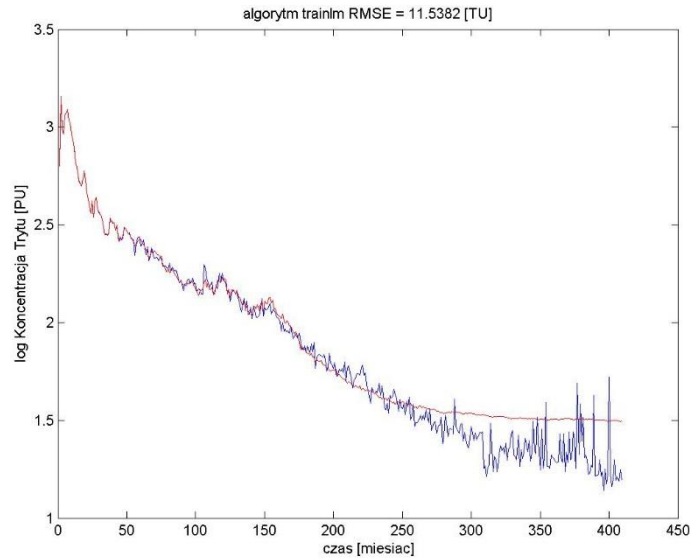
Szerokość danych wejściowych [miesiąc]	48	60	72	80	84	96	120	132	144
traingdx RMSE [TU]	34	32.8	40.35	31.51	35	32.3	27.5	27.6	30.1
trainlm RMSE [TU]	12.8	11.5	11.53						

Dla algorytmu trainlm uzyskałem tylko trzy pierwsze wartości błędu ze względu na ograniczony zasób pamięci komputera. Algorytm ten wykorzystuje bardzo dużo pamięci i dla większej sieci wartość ta rośnie. Jednakże algorytm ten pracuje dużo lepiej i osiąga mniejsze wartości błędu. Dodatkowo, potrzebuje on tylko 15 iteracji do osiągnięcia przedstawionych rezultatów podczas gdy algorytm traingdx wymaga około 1000 iteracji. Minimalna wartość błędu dla algorytmu traingdx odpowiada szerokości 120 danych wejściowych, natomiast dla algorytmu trainlm szerokość ta równa jest 60. Należy zaznaczyć, że uzyskane wartości nieznacznie się różnią za każdym wywołaniem algorytmu ze względu na losowe wartości startowe współczynników wagowych neuronów.

Poniżej przedstawiam zestawienie obliczonych przez sieć stężeń trytu z danymi pomiarowymi dla optymalnych wyników obu algorytmów:



*Wykres 1. Rozkład czasowy stężenia trytu w Dunaju, kolor niebieski - dane doświadczalne, kolor czerwony - wynik sieci, algorytm traingdx, szerokość okna 120 miesięcy*



Wykres 2. Rozkład czasowy stężenia trytu w Dunaju, kolor niebieski - dane doświadczalne, kolor czerwony - wynik sieci, algorytm trainlm, szerokość okna 60 miesięcy

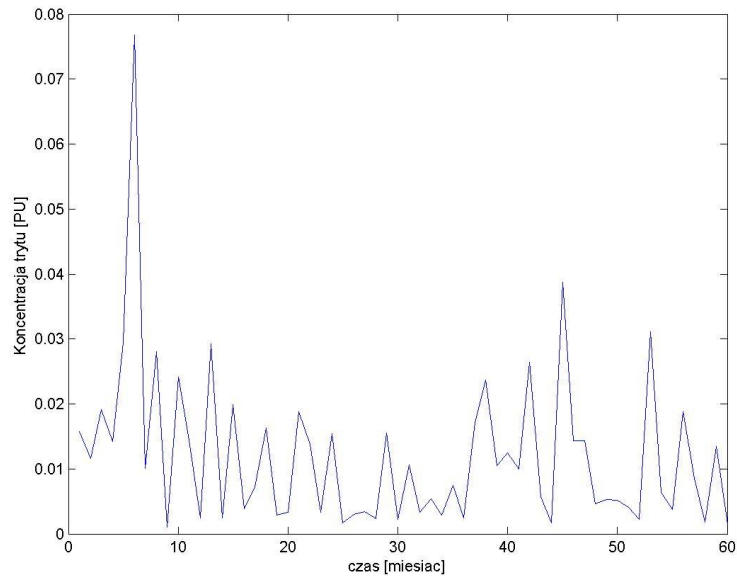
Oba wyniki dobrze odzwierciedlają dane uczące ale dla danych testowych wykazują pewne rozbieżności. Sieć uczona przez algorytm trainlm lepiej odwzorowuje dane testowe. Jednym z problemów sieci jest fakt, że dane uczące mają inny zakres wartości niż dane testowe. Dane zostały podzielone w sposób liniowy.

### 3. Średni czas przebywania wody w zlewni

Na podstawie skonstruowanej i wyuczonej sieci neuronowej możemy policzyć średni czas  $\tau$  przebywania wody w modelowanym obiekcie. Zgodnie z definicją  $\tau$  dla odpowiedzi układu na wymuszenie impulsowe wyrażone jest wzorem:

$$\tau = \frac{\int_0^{\infty} tC(t)dt}{\int_0^{\infty} C(t)dt}$$

Licznik w powyższym wzorze stanowi pierwszy moment rozkładu czasowego  $C(t)$ , natomiast mianownik stanowi normę rozkładu czasowego. Wartość  $\tau$  możemy wyznaczyć zadając na wejście sieci macierz jednostkową o wymiarach odpowiadających ilości neuronów wejściowych. Do obliczeń wykorzystałem sieć wytrenowaną algorytmem trainlm o szerokości okna równej 60 miesięcy. Rozkład stężenia trytu będący odpowiedzią układu na macierz jednostkową wygląda następująco:



Wykres 3. Rozkład czasowy koncentracji trytu w odpowiedzi na wymuszenie jednostkowe

Uzyskany na podstawie powyższej odpowiedzi średni czas przebywania wody w Dunaju wynosi:

$$\tau = 26.1 \text{ miesięcy}$$

#### 4. Podsumowanie

W ramach ćwiczenia skonstruowałem sztuczną sieć neuronową i wykonałem dla niej obliczenia stężenia trytu w zlewni Dunaju. Zbadałem jak szerokość okna danych wejściowych oraz algorytm wpływa na błąd pracy sieci. Zauważyłem, że algorytm `trainlm` Levenberga-Marquardta jest dużo bardziej skuteczny od algorytmu `traindx` jednakże wymaga dużo większej pamięci obliczeniowej komputera. Dla optymalnie wytrenowanej sieci wykonałem obliczenia średniego czasu przebywania wody w zlewni zadając na wejście sieci macierz jednostkową.