

| | | | |
|---|--|--------|--------------|
|  | AKADEMIA GÓRNICZO – HUTNICZA KRAKÓW | | Tomasz Kąkol |
| | Modelowanie metodą sztucznych sieci neuronowych | | |
| Modelowanie procesów fizycznych | | | |
| Data wykonania ćwiczenia: 04.06.2018 | Data złożenia sprawozdania: 09.06.2018 | Ocena: | |

Wprowadzenie

W przypadku tego laboratorium, analogicznie jak w laboratorium nr 7, należało obliczyć przebieg czasowy stężenia trytu w Dunaju. W tym jednak przypadku należało skorzystać z możliwości jakie udostępnia technika sztucznych sieci neuronowych. . Znacznikiem (podobnie jak w we wspomnianym już laboratorium) stosowanym w modelowaniu jest tryt (^3H).

Model

W zastosowanym modelowaniu dla naszego typu danych zdecydowałem się na zastosowanie sieci regresji grnn (ang. ‘generalized regression neural network’). Uogólnione sieci neuronowe regresji *grnn* są rodzajem radialnej sieci bazowej. Omawiany typ sieci cechuje dwuwarstwowość sieć. Dla tego typu zaprojektowanej sieci należy poszukiwać optymalnej wartości parametru ‘*spread*’ (Rozprzestrzenianie podstawowych funkcji radialnych).

Cechą charakterystyczną parametru ‘*spread*’ jest własność, że im wartość jest większa, tym bardziej gładka jest aproksymacja funkcji. Domyślnie wartość parametru jest ustawiona na 1. Wartość tą należy dobierać większą od zera. Wobec tego dobrałem różne wartości ‘*spread*’ od 1 do 40. Zgodnie z teorią, po znalezieniu najlepszego przez nas rozwiązania do parametru ‘*spread*’, w celu wykonania próby znalezienia minimum globalnego, należy zagęścić obszar poszukiwać w obrębie tej wartości. Ja już nie wykonałem poszukiwania minimum globalnego. Celem tego laboratorium nie było wykonanie optymalnej topologii i wartości parametrów sieci neuronowej.

Skutkiem zbyt małej wartości parametru ‘*spread*’ jest to, że zbyt wiele neuronów jest wymaganych do płynnego dopasowania funkcji.

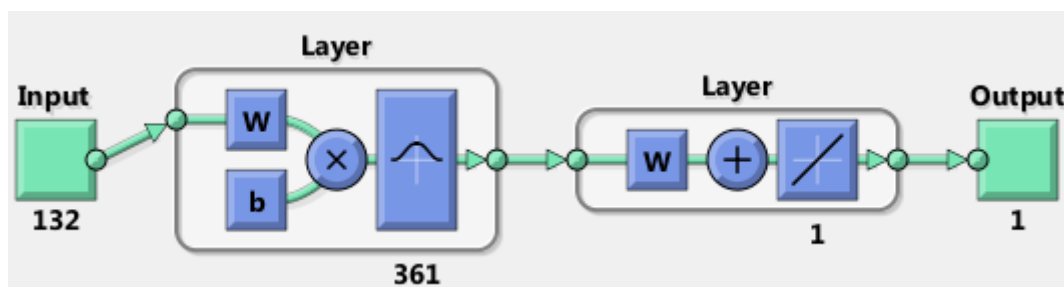
W przypadku poszukiwania optymalnej topologii sieci i parametrów radialnych funkcji bazowych, należy zastosować większą próbę przypadków ‘jednostkowych’. Następnie należy wybrać najlepiej wytrenowaną sieć lub należy uśrednić wyniki uzyskane w tych przypadkach ‘jednostkowych’. W załączonym programie wartość zmiennej iteracyjnej ustawiłem na maksymalną wartość 1, aby przy uruchomieniu nie oczekiwać ‘długo’ na przykładowe rezultaty symulacji (natomiast dla przedstawionych wyników liczba iteracji była równa 20).

Symulację wykonałem dzieląc dane na dane wyjściowe (oczekiwane) i dane wejściowe. Te drugie podzieliłem na dane trenujące i testujące. Wykonałem 9 podziałów zgodnie z wartością zmiennej współczynnika D (od 0.1 do 0.9 z krokiem 0.1) – oznaczającego proporcję udziału danych trenujących do wszystkich danych. Wartość '1-D', analogicznie, oznacza proporcję udziału danych testujących do wszystkich danych.

Czas wykonania obliczeń w porównaniu z topologią sieci RBF (*newrb*) jest znacznie szybszy, co jest niewątpliwie znaczącą zaletą modelu GRNN.

Rozwiązanie

Z naszych danych trenujących wytrenowana topologia sieci jest następująca:



Rysunek 1 Topologia przykładowej sieci neuronowej GRNN

Topologie należy rozumieć tak:

Input 132 – są to nasze dane wejściowe dla 132 miesięcy. Wspomniane dwie warstwy, pierwsza złożona z 361 neuronów opartych o radialne funkcje bazowe, w której wyznaczane są wagi połączeń, druga warstwa posiada liniową funkcję aktywacji. Na podstawie wykonanych obliczeń zostaje wyznaczona wartość na wyjściu sieci (wektor z 1 kolumną wartości).

Przykładowy rezultat otrzymanych wyników w 'Command Window' po wykonaniu programu:

```
Najlepsze lokalne dopasowanie dla: 48 miesięcy
Dane testujące: Średni wiek wody: 24.50, error: 4.62893, spread: 20.00 ,współczynnik D: 0.9
Najlepsze lokalne dopasowanie dla: 60 miesięcy
Dane testujące: Średni wiek wody: 30.50, error: 2.49252, spread: 10.00 ,współczynnik D: 0.9
Najlepsze lokalne dopasowanie dla: 72 miesięcy
Dane testujące: Średni wiek wody: 36.50, error: 0.65205, spread: 5.00 ,współczynnik D: 0.9
Najlepsze lokalne dopasowanie dla: 80 miesięcy
Dane testujące: Średni wiek wody: 40.51, error: 0.84236, spread: 15.00 ,współczynnik D: 0.9
Najlepsze lokalne dopasowanie dla: 84 miesięcy
Dane testujące: Średni wiek wody: 42.50, error: 0.46050, spread: 30.00 ,współczynnik D: 0.9
Najlepsze lokalne dopasowanie dla: 96 miesięcy
Dane testujące: Średni wiek wody: 48.50, error: 0.42717, spread: 30.00 ,współczynnik D: 0.9
Najlepsze lokalne dopasowanie dla: 120 miesięcy
Dane testujące: Średni wiek wody: 60.50, error: 0.38085, spread: 5.00 ,współczynnik D: 0.9
```

Najlepsze lokalne dopasowanie dla: 132 miesięcy

Dane testujące: Średni wiek wody: 66.50, error: 0.36460, spread: 10.00 ,współczynnik D: 0.9

Najlepsze lokalne dopasowanie dla: 144 miesięcy

Dane testujące: Średni wiek wody: 72.50, error: 0.31085, spread: 20.00 ,współczynnik D: 0.9

Najlepsze dopasowanie dla danych testujących:

Wyniki: Średni wiek wody: 72.50, Error: 0.31085 , Miesiące: 144, Spread: 20.00, Współczynnik D: 0.9

Jak wynika z zamieszczonych rezultatów, w naszym przypadku wykonanie próby znalezienia minimum globalnego błędu nie jest łatwym zadaniem, ponieważ zakres wartości dla najlepszych rozwiązań nie jest systematyczny.

Na podstawie przedstawionych wyników należy wnioskować, że najlepsze wytrenowanie sieci otrzymaliśmy, gdy liczba danych trenujących wynosiła 90%. Dla tej wartości nie wystąpiło jeszcze zjawisko przeuczenia sieci.

Wykonany program jest całkowicie uniwersalny, można edytować wartości współczynnika D na inny zakres (np. 0.05:0.05:0.95), i w ten sposób możemy zwiększyć szanse znalezienia optymalnej topologii i wartości parametrów naszej sieci.

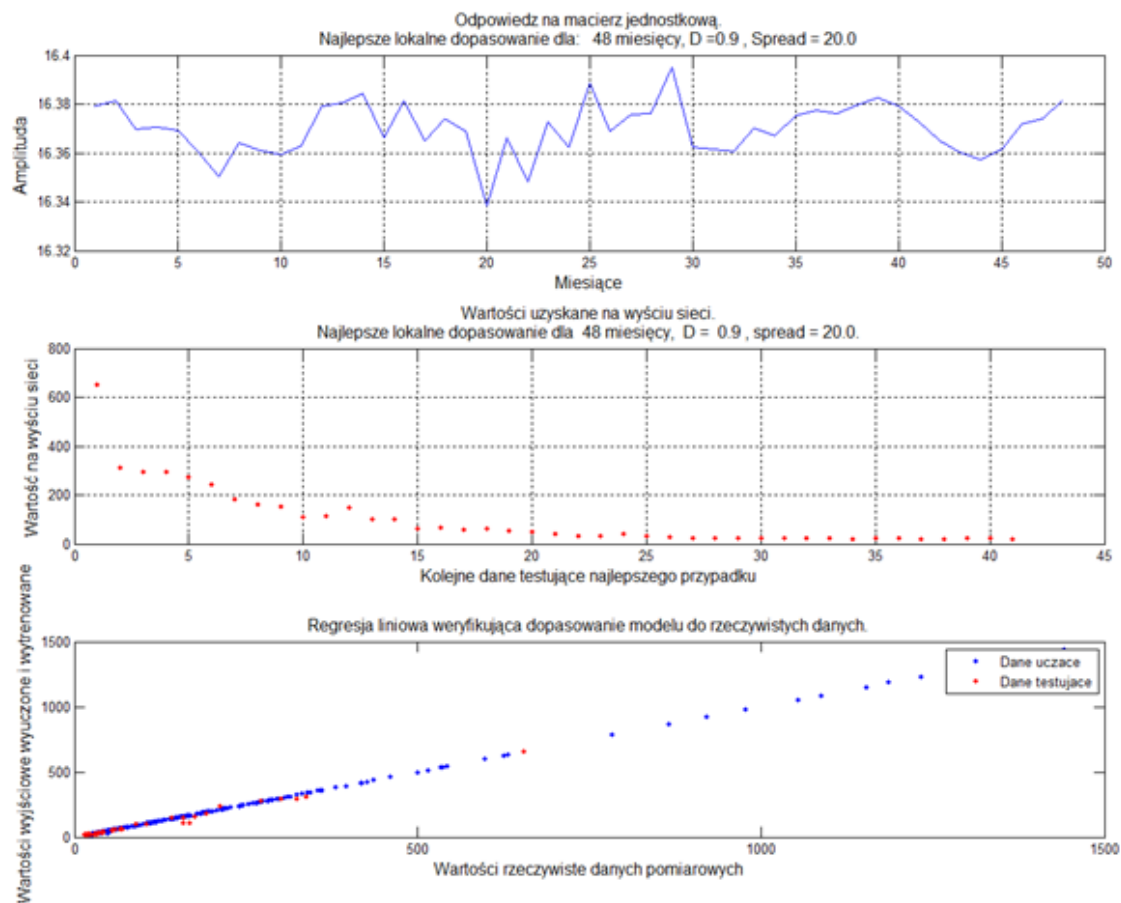
Wartość średniego wieku wody również jest zmienna w zależności od przyjętego zestawu danych. Do wyrażenia średniego błędu wprowadzono poprawkę, aby uwzględnić zmienność wielkości zbioru danych.

Ze standartowej: $\text{error} = \text{mean}((Y_{\text{test}} - D_{\text{test}}') \cdot (Y_{\text{test}} - D_{\text{test}}'))$

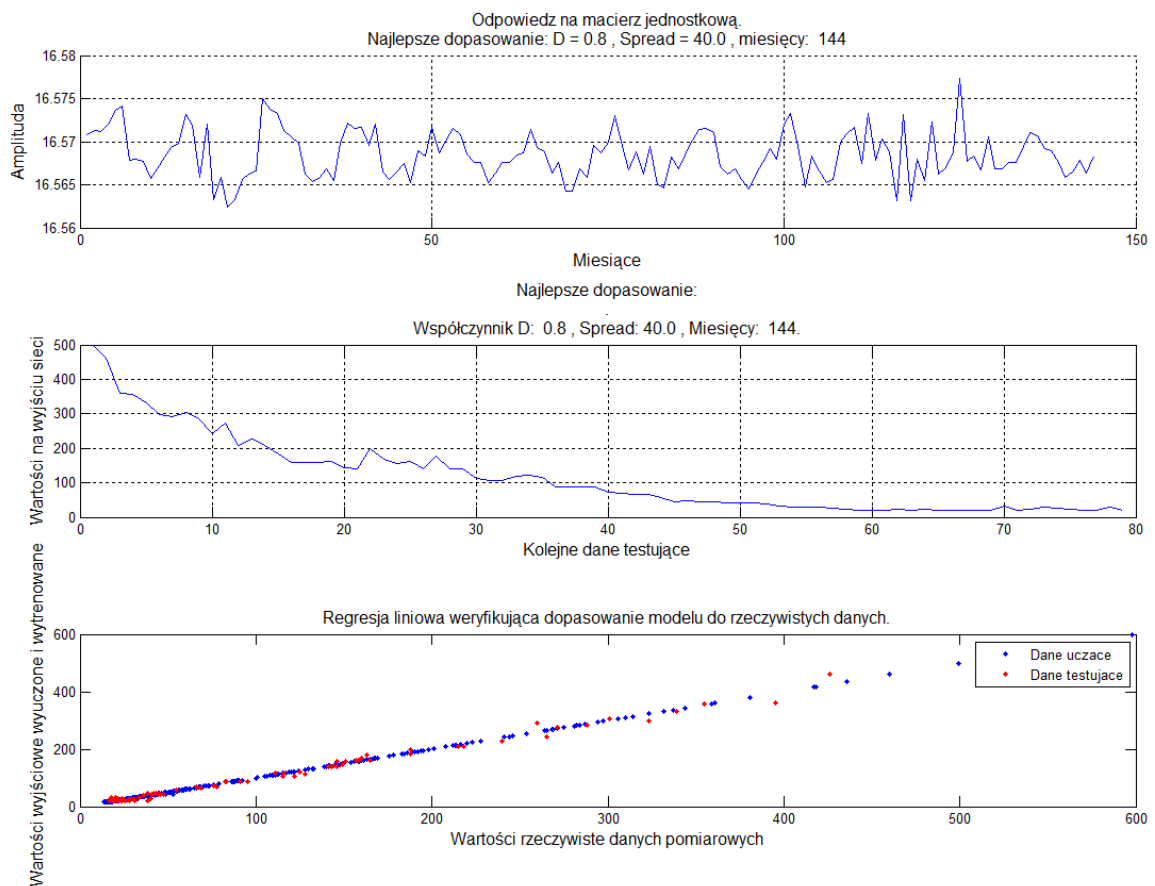
Na następującą: $\text{error} = \text{mean}((Y_{\text{test}} - D_{\text{test}}') \cdot (Y_{\text{test}} - D_{\text{test}}')) / \text{size}(\text{inputData}, 2)$;

Brak tej poprawki powoduje 'zafałszowanie' rzeczywistego stanu, preferując zbiory o najmniejszej liczbie danych. Wobec tego parametr 'error' w naszej symulacji nie oznacza 'do końca' MSE dla zbioru danych trenujących, lecz uwzględnia przy tym również wielkość zbioru danych.

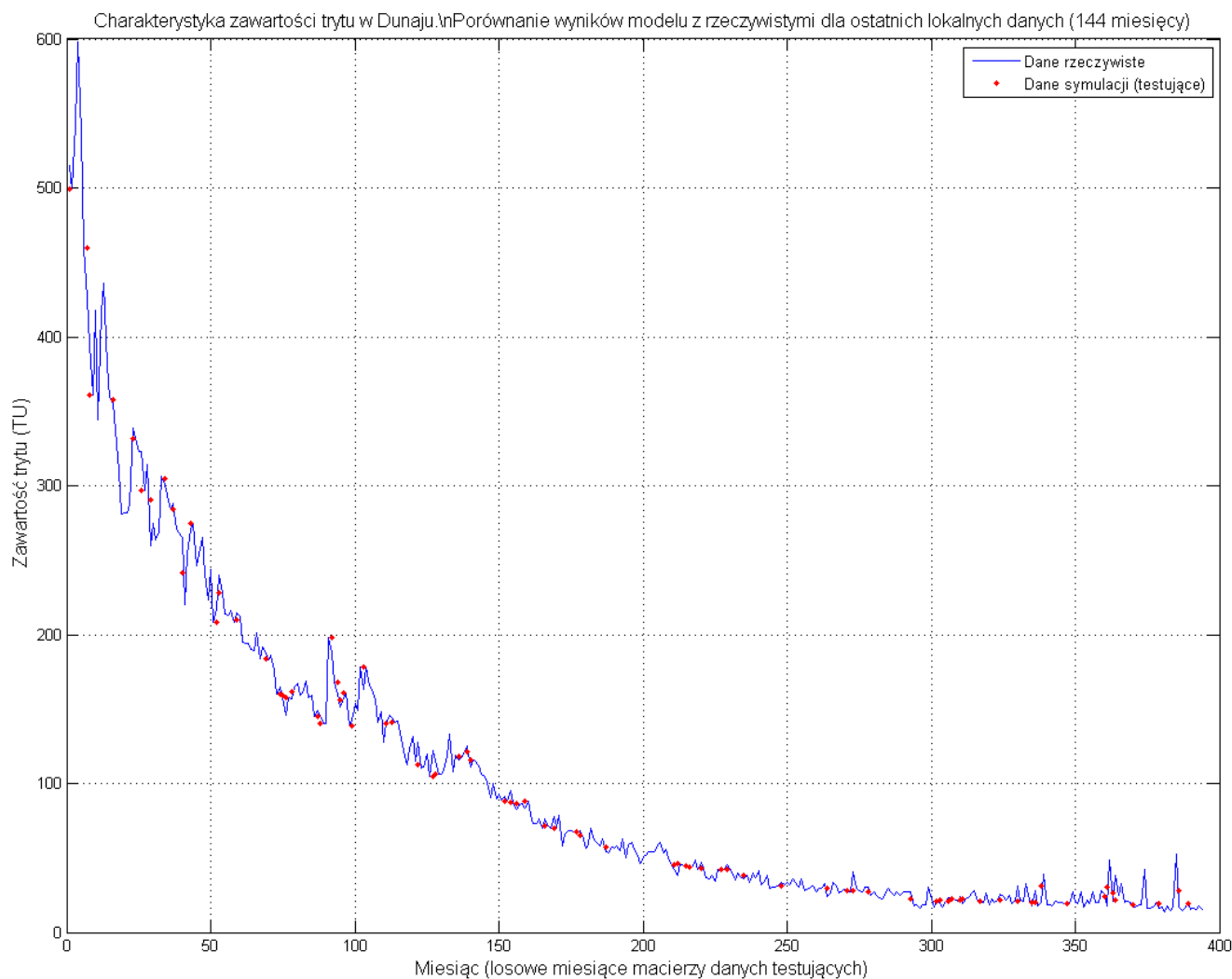
Poniżej przedstawiono przykładowe rezultaty odpowiedzi modelu na macierz jednostkową (tzw. funkcję przejścia), wartości uzyskane na wyjściu naszej sieci, charakterystykę – regresję liniową weryfikującą dopasowanie modelu do rzeczywistych danych, a także przykładową charakterystykę zawartości trytu w Dunaju dla danych z 144 miesięcy.



Rysunek 2 Przykładowe charakterystyki otrzymane dla danych z 48 miesięcy



Rysunek 3 Przykładowe charakterystyki otrzymane dla danych z 144 miesięcy



Rysunek 4 Przykładowa charakterystyka zawartości trytu (TU) w zależności od czasu (144 miesiące)

Na podstawie przedstawionych rezultatów, widoczne jest jak relatywnie dobrze nasz model naśladuje rzeczywiste wartości, mimo tego, że nie staraliśmy się 'z całych sił i posiadanej wiedzy' znaleźć minimum globalnego. Jestem relatywnie zadowolony z uzyskanych wyników.

Podsumowanie

W porównaniu z laboratorium 7, w tym przypadku otrzymaliśmy lepsze dopasowanie modelu do rzeczywistego stanu wartości zawartości tryfu w Dunaju. Wartości otrzymanych błędów są również bardziej korzystne dla zastosowanej topologii SSN, niż w przypadku modeli empirycznych, dyspersyjnych i im podobnym.

Ważnym podkreśleniem jest prostota, a zarazem skuteczność nauczania zastosowanego przeze mnie typu sieci neuronowej.