

210200
<i>numer albumu</i>
Justyna Hubert
<i>imię i nazwisko</i>

210294
<i>numer albumu</i>
Karol Podlewski
<i>imię i nazwisko</i>

Data	20 czerwca 2018
Kierunek	Informatyka
Rok akademicki	2017/18
Semestr	IV
Grupa Dziekańska	I

Laboratorium inteligentnej analizy danych

Zadanie 3: Sieć RBF - aproksymacja

Sieć RBF - wstęp

Zadanie polegało na implementacji oraz zastosowaniu sieci RBF, czyli sieci neuronowej z radialnymi funkcjami bazowymi. Składa się ona z dwóch warstw:

- Pierwsza warstwa to warstwa radialna z gaussowską funkcją radialną dla neuronów. Neurony obliczają odległość między wektorem wejściowym a wektorem reprezentującym centrum funkcji radialnej. Następnie odległość tę, przemnożoną przez współczynnik skalujący, przekazują jako argument do funkcji gaussowskiej.
- Druga warstwa składa się z tradycyjnych neuronów liniowych z identycznościową funkcją aktywacji.

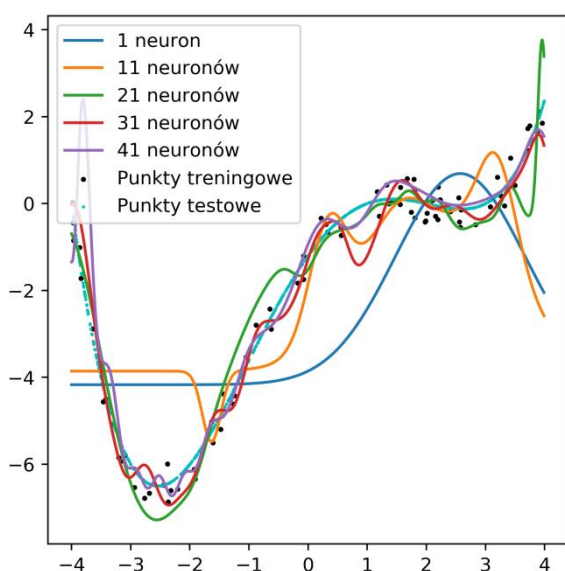
Skupiliśmy się na wykorzystaniu sieci RBF do aproksymacji funkcji. Centra losowane były z wektora treningowego, zaś współczynnik skalujący σ obliczany był ze wzoru:

$$\sigma = \frac{2 \cdot \text{mnożnik}}{\text{odległość}}$$

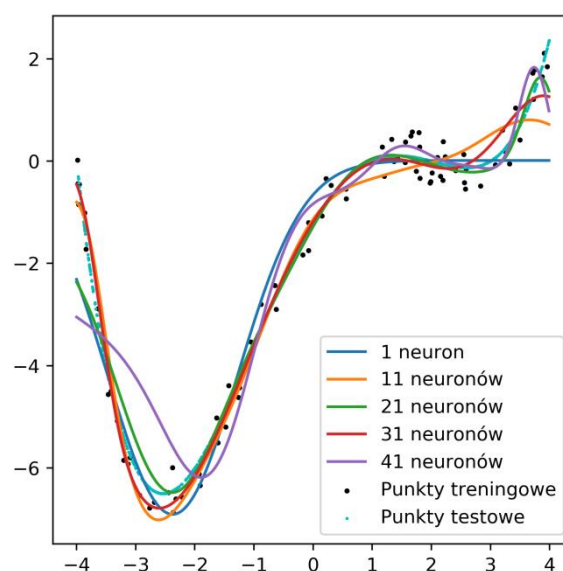
gdzie mnożnik był wprowadzany przez nas (dla optymalnego współczynnika skalującego był on równy 1), zaś odległość była równa średniej odległości pomiędzy najbliższymi neuronami z obu stron. Ze względu na niemożliwość skorzystania ze wzoru dla 1 neuronu, wartość współczynnika skalującego ustawiana była na sztywno i wynosiła ona 1.1.

1. Ilość neuronów a jakość aproksymacji

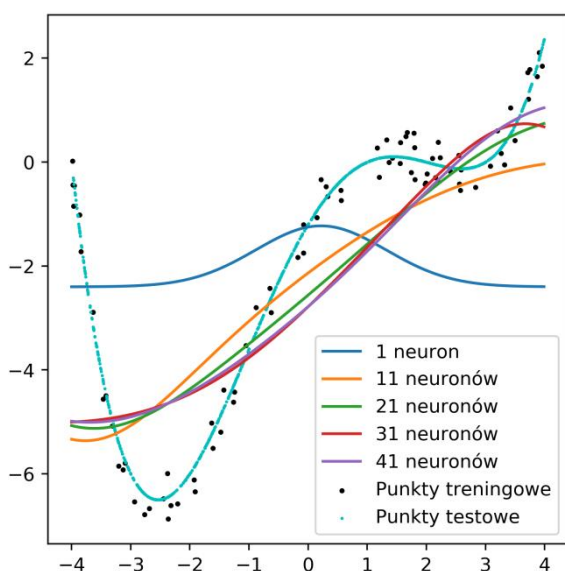
Na samym początku skupiliśmy się na zbadaniu ilości neuronów oraz wysokości współczynnika skalującego potrzebnych do zadowalającej nauki sieci przy 25 iteracjach.



Rys1. Zbyt mały współczynnik skalujący



Rys2. Odpowiedni współczynnik skalujący



Rys3. Zbyt duży współczynnik skalujący

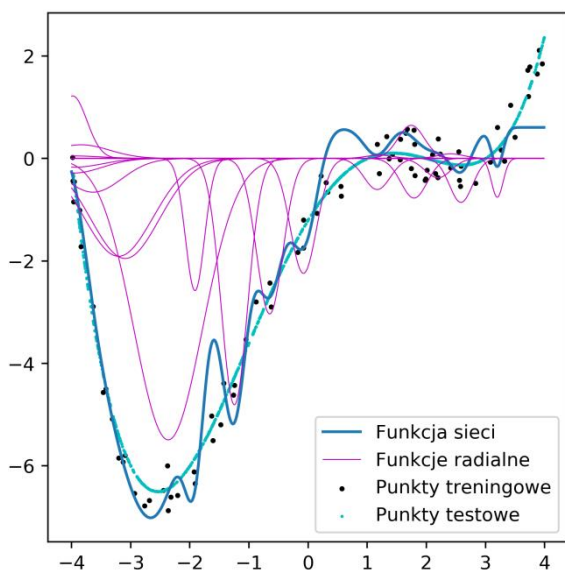
Na Rys1 wyraźnie widać, że przy źle dobranym współczynniku skalującym nawet 41 neuronów nie jest w stanie sobie poradzić z aproksymacją do zadanych danych treningowych. Wykres staje się „szpiczasty”, funkcja ciągle wychyla się w stronę punktów treningowych, przez co aproksymacja nie jest dobra.

W przypadku odpowiednio dobranego współczynnika skalującego (Rys2) funkcja już całkiem nieźle radzi sobie nawet przy 11 neuronach, zaś różnica między 21 a 41 neuronami są naprawdę znikome – mniejsza liczba ma jednak tę przewagę, że sieć się nie przeucza.

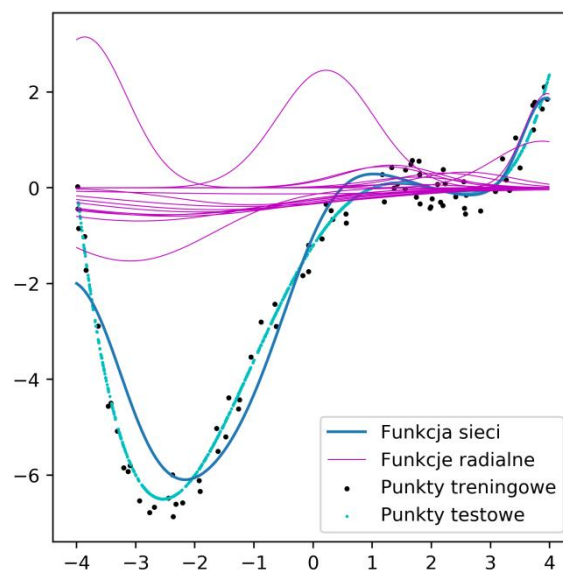
Dla zbyt dużego współczynnika skalującego, widocznego na Rys3, funkcja jest zbyt wypłaskowana, trudno tutaj mówić o dobrym dopasowaniu do wzorca.

2. Funkcje radialne neuronów radialnych

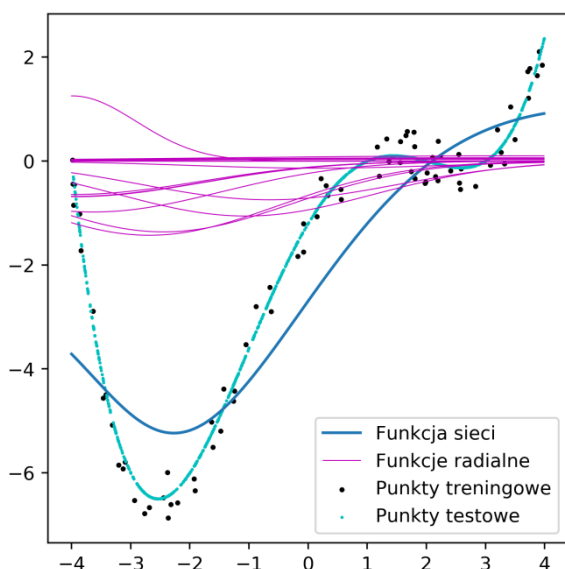
Wiedząc już, że 21 neuronów to optymalna liczba w przypadku tych danych treningowych, zajęliśmy się sprawdzeniem wpływu poszczególnych funkcji radialnych na całą funkcję sieci. Sieć uczyła się przez 25 iteracji.



Rys4. Zbyt mały współczynnik skalujący



Rys5. Odpowiedni współczynnik skalujący



Rys6. Zbyt duży współczynnik skalujący

Patrząc na uzyskane wykresy pierwszym nasuwającym się wnioskiem jest to, że każda funkcja radialna ma wpływ na końcową funkcję. Zadaniem współczynnika skalującego jest zrównoważyć wpływ poszczególnych neuronów.

W przypadku zbyt małego współczynnika skalującego (Rys4) poszczególne neurony potrafią mieć za wysoki wpływ na daną funkcję – funkcje łatwo osiągają duże wartości w miejscach punktów treningowych przez co wykres jest bardzo chaotyczny.

Dla zbyt dużego współczynnika skalującego poszczególne funkcje mają za mały wpływ na końcowy wynik.

3. Ilość neuronów a błąd średniokwadratowy

Następnie zajęliśmy zbadaniem wpływu ilości neuronów na błąd. W każdym wypadku sieć uczyła się przez 25 iteracji.

Tab1. Wyniki nauki dla sieci z jedną figurą geometryczną

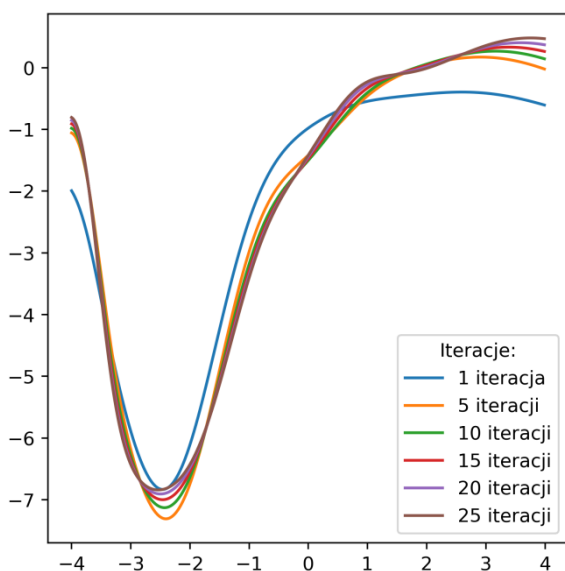
Liczba neuronów	Zbiór treningowy		Zbiór testowy	
	Średni błąd	Odchylenie standardowe	Średni błąd	Odchylenie standardowe
1	2,72047	0,65800	2,24440	0,63422
6	0,50840	0,29363	0,34954	0,21173
11	0,29692	0,19551	0,17405	0,12539
16	0,21191	0,14663	0,13508	0,09895
21	0,14007	0,09401	0,07560	0,05677
26	0,17570	0,15621	0,08955	0,08352
31	0,19256	0,18863	0,09851	0,09866
36	0,20033	0,18295	0,09533	0,08964
41	0,25767	0,20170	0,13971	0,11588

Uzyskane rezultaty pokrywają się z wcześniejszymi eksperymentami – sieć najlepiej się uczy przy 21 neuronach. Mniejsza ilość neuronów sprawia, że sieć nie jest w stanie nauczyć się wystarczająco dobrze. Większa ilość neuronów powoduje za to delikatne przeuczenie sieci – w przypadku nieprecyzyjnych danych treningowych może to prowadzić do błędów.

Warto pamiętać, że uzyskany przez nas rezultat pod żadnym pozorem nie jest optymalną liczbą neuronów dla sieci RBF. Za każdym razem liczbę tę należy dobrać eksperymentalnie.

4. Ilość iteracji potrzebna do dobrej nauki

Ostatnią rzeczą jaką się zajęliśmy było jakie rezultaty uzyskuje sieć w trakcie nauki. Jak zwykle skorzystaliśmy z sieci z 21 neuronami, zaś nauka trwała 25 iteracji.



Rys7. Zbyt duży współczynnik skalujący

Patrząc na uzyskane wyniki możemy stwierdzić, że sieć RBF bardzo szybko uczy się na początku (wynik przy 5 iteracji prawie się nie różni od tego dla 25) jednak dokładna nauka zajmie zdecydowanie więcej czasu – co jest jak najbardziej spodziewanym rezultatem.

Przy nieprecyzyjnych wartościach, szczególnie na początku bądź końcu zbioru danych treningowych sieć może potrzebować więcej iteracji w celu dobrej aproksymacji funkcji.