

| | | |
|------------------------|----------------|-------|
| ----- | 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 | 4 kwietnia 2018 |
| Kierunek | Informatyka |
| Rok akademicki | 2017/18 |
| Semestr | IV |
| Grupa Dziekańska | I |

Laboratorium inteligentnej analizy danych

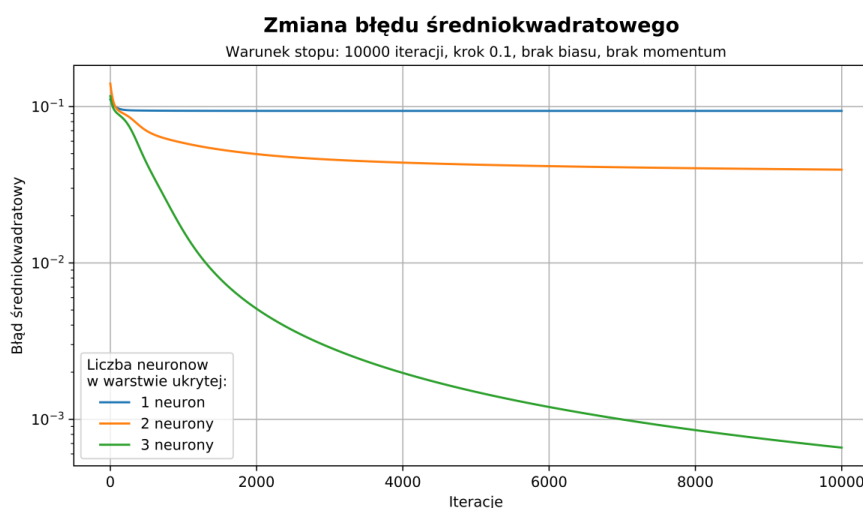
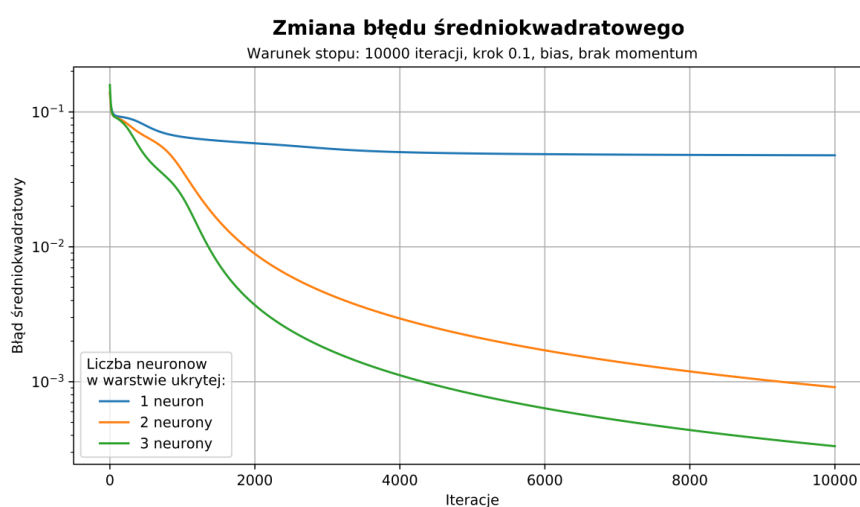
Zadanie 1: Transformacja i aproksymacja

Streszczenie - Transformacja

Celem ćwiczenia było stworzenie sieci neuronowej o 4 wejściach oraz 4 wyjściach z jedną warstwą ukrytą, która powinna nauczyć się odtwarzać otrzymane dane. Wszystkie neurony charakteryzuje sigmoidalna funkcja aktywacji.

1. Zmiana błędu średniokwadratowego

Na początku sprawdziliśmy zmianę błędu średniokwadratowego przy 10 000 epokach nauki, dla odpowiednio 1, 2 lub 3 neuronów. Krok nauki był stały i wynosił 0.1, zmiana wag nie uwzględniała momentum. Rozważyliśmy dwa przypadki: odpowiednio z biasem oraz bez niego.



Przy 1 neuronie w warstwie ukrytej sieć szybko przestawała się uczyć. Zmiana błędu stawała się wręcz niezauważalna, szczególnie w przypadku gdy sieć była pozbawiona biasu. Obecność wagi zerowej znacząco poprawiała rezultaty sieci z 2 neuronami, która dzięki temu cały czas efektywnie się uczyła. Dla sieci z 3 neuronami obecność biasu miała mniejszy wpływ na prędkość nauki, jednak i w tym przypadku końcowy obliczony błąd okazał się mniejszy. Żadnym zaskoczeniem nie był fakt, że zwiększanie liczby neuronów znacząco poprawia uzyskany wynik, jednak warto zaznaczyć wpływ biasu, szczególnie przy małej liczbie neuronów w warstwie ukrytej.

2. Uśredniony czas nauki

Kolejną rzeczą było sprawdzenie wpływu parametrów nauki na jej tempo. Sieć posiadała dwa neurony w warstwie ukrytej oraz bias, zaś parametry nauki były zmienne. Nauka miała trwać do momentu aż wartość błędu średniokwadratowego będzie mniejsza niż 0.0001. Dla każdej pary wartości kroku nauki oraz kroku momentu przeprowadziliśmy 5 prób, z których wyciągnęliśmy średnią ilość epok potrzebnych do osiągnięcia oczekiwanego rezultatu. Wyniki prezentujemy w tabeli poniżej:

| Krok nauki | Krok momentum | Średnie iteracje | Odchylenie standardowe |
|------------|---------------|------------------|------------------------|
| 0.01 | 0 | 899910 | 0.01414 |
| 0.01 | 0.01 | 745202 | 0.01414 |
| 0.01 | 0.05 | 706306 | 0.01414 |
| 0.01 | 0.1 | 739255 | 0.01414 |
| 0.01 | 0.5 | 421202 | 0.01414 |
| 0.05 | 0 | 173155 | 0.01414 |
| 0.05 | 0.01 | 145187 | 0.01414 |
| 0.05 | 0.05 | 140687 | 0.01414 |
| 0.05 | 0.1 | 132720 | 0.01414 |
| 0.05 | 0.5 | 74096 | 0.01414 |
| 0.1 | 0 | 73936 | 0.01414 |
| 0.1 | 0.01 | 72617 | 0.01414 |
| 0.1 | 0.05 | 94450 | 0.01414 |
| 0.1 | 0.1 | 66155 | 0.01414 |
| 0.1 | 0.5 | 37389 | 0.01414 |
| 0.5 | 0 | 17690 | 0.01414 |
| 0.5 | 0.01 | 14558 | 0.01414 |
| 0.5 | 0.05 | 15510 | 0.01414 |
| 0.5 | 0.1 | 13261 | 0.01414 |
| 0.5 | 0.5 | 8329 | 0.01414 |
| 1 | 0 | 7886 | 0.01414 |
| 1 | 0.01 | 8699 | 0.01414 |
| 1 | 0.05 | 7982 | 0.01414 |
| 1 | 0.1 | 6664 | 0.01414 |
| 1 | 0.5 | 3684 | 0.01414 |

Pierwszym wnioskiem po spojrzeniu na średnią ilość iteracji jest to, że momentum zazwyczaj przyspiesza naukę. Szczególnie duże wartości kroku (0.1 oraz 0.5) pozytywnie wpływają na tempo nauki. Niższe wartości przy nieodpowiednim wylosowaniu wag mogą wydłużyć czas potrzebny do osiągnięcia oczekiwanego rezultatu (względem braku momentum), zamiast go przyspieszyć. Bardzo dobrze widać dla kroku nauki 0.1 oraz kroku momentum 0.05:

| Krok nauki | Krok momentum | Iteracje | Odchylenie standardowe |
|------------|---------------|----------|------------------------|
| 0.1 | 0.05 | 70988 | 0.01414 |
| 0.1 | 0.05 | 130057 | 0.01414 |
| 0.1 | 0.05 | 71083 | 0.01414 |
| 0.1 | 0.05 | 130488 | 0.01414 |
| 0.1 | 0.05 | 69632 | 0.01414 |

Podobnie jak z momentum, tak i z krokiem nauki – większy krok powoduje przyspieszenie nauki. Początkowe wartości błędu potrafią jednak „skakać”, co nie jest pożądanym zdarzeniem. W przypadku zbyt dużych wartości kroków sieć może nawet nigdy nie doprowadzić nauki do końca.

3. Odpowiedzi z warstwy ukrytej

Na samym końcu zajęliśmy się interpretacją odpowiedzi z neuronów warstwy ukrytej dla każdej danej wejściowej. Rozważana sieć posiadała dwa neurony z warstwie ukrytej. kroki nauki 0.1. bez momentum. Nauka trwała 1 000 000 iteracji.

| Sieć z biasem | | |
|---------------|-------------------|-------------------|
| Wejście | Wyjście neuronu 1 | Wyjście neuronu 2 |
| 1 0 0 0 | 0.0054 | 0.0063 |
| 0 1 0 0 | 0.9946 | 0.9943 |
| 0 0 1 0 | 0.0051 | 0.9839 |
| 0 0 0 1 | 0.9816 | 0.0045 |

| Sieć bez biasu | | |
|----------------|-------------------|-------------------|
| Wejście | Wyjście neuronu 1 | Wyjście neuronu 2 |
| 1 0 0 0 | 0.0518 | 0.0518 |
| 0 1 0 0 | 0.0517 | 0.0517 |
| 0 0 1 0 | 0.0036 | 0.9995 |
| 0 0 0 1 | 0.9995 | 0.0034 |

Wyniki dla obu sieci różnią się w przypadku dwóch pierwszych wejść. zaś są niemal identyczne dla dwóch ostatnich. Pamiętając wyniki z 1 części tego zadania. można przypuszczać że nawet dla miliona operacji sieć z 2 neuronami w warstwie ukrytej. ale bez biasu nie będzie zwracać wystarczająco poprawnych wyników – te dla wejść 1000 oraz 0100 są niemalże identyczne.

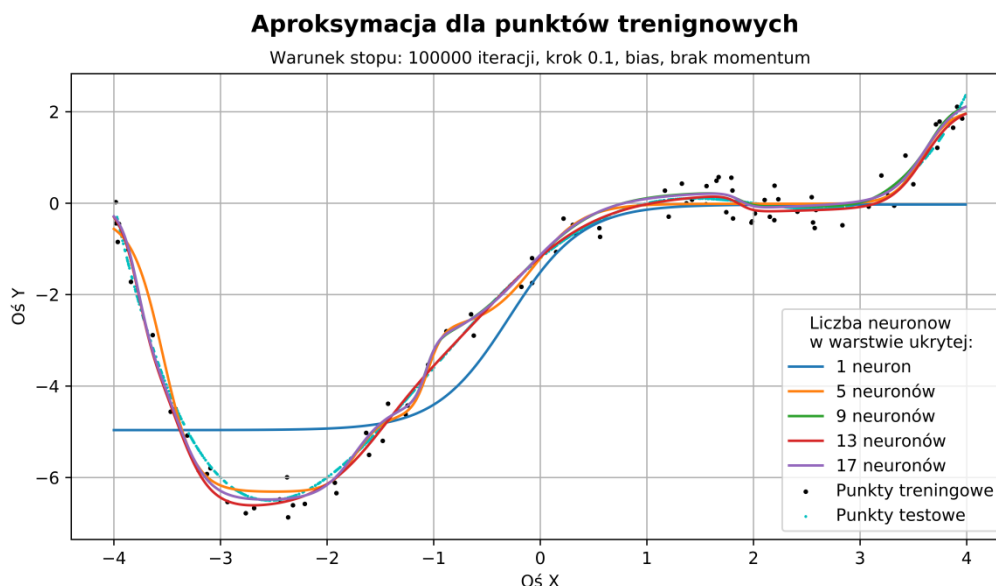
Dla sieci z biasem wyniki te się od siebie różnią (dążą odpowiednio do 1 bądź 0) i można już z nich coś wywnioskować. Wyjścia neuronów z sieci ukrytej działają jak włączniki neuronów w warstwie wyjściowej – by dany neuron miał na wyjściu 1. „włączniki” w warstwie ukrytej muszą być odpowiednio ustawione.

Streszczenie - Aproksymacja

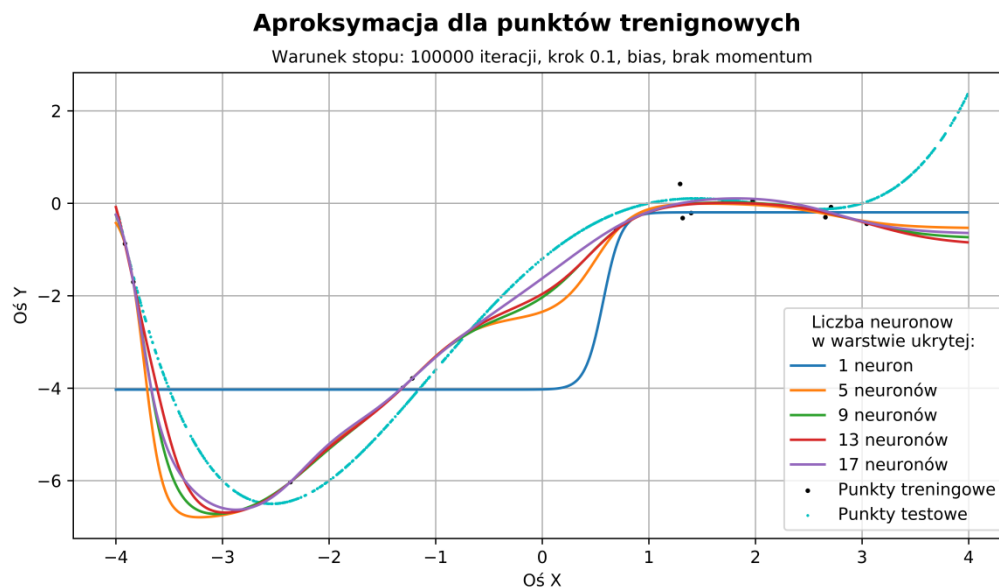
Celem ćwiczenia było stworzenie sieci neuronowej o 1 wejściu oraz 1 wyjściu z jedną warstwą ukrytą. Neurony w warstwie ukrytej charakteryzuje sigmoidalna funkcja aktywacji. zaś w warstwie wyjściowej – liniowa. Sieć miała się uczyć na jednych danych. zaś jakość aproksymacji sprawdzać na innych.

1. Aproksymacja dla różnych danych

Na samym początku sprawdziliśmy jakość aproksymacji w zależności od liczby neuronów w warstwie ukrytej dla obu zbiorów danych treningowych. Sieć uczyła się do osiągnięcia 100 000 epoki. dodatkowo charakteryzowały ją obecność biasu, brak momentum oraz krok nauki 0.1. Wyniki dla danych 1 prezentują się następująco:



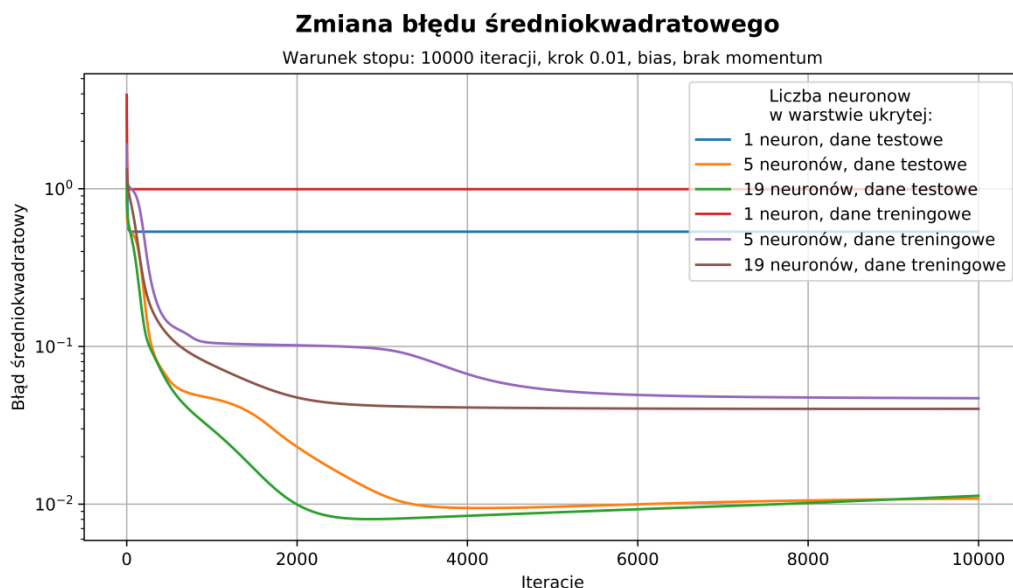
Dla danych 2 wyglądają zaś następująco:



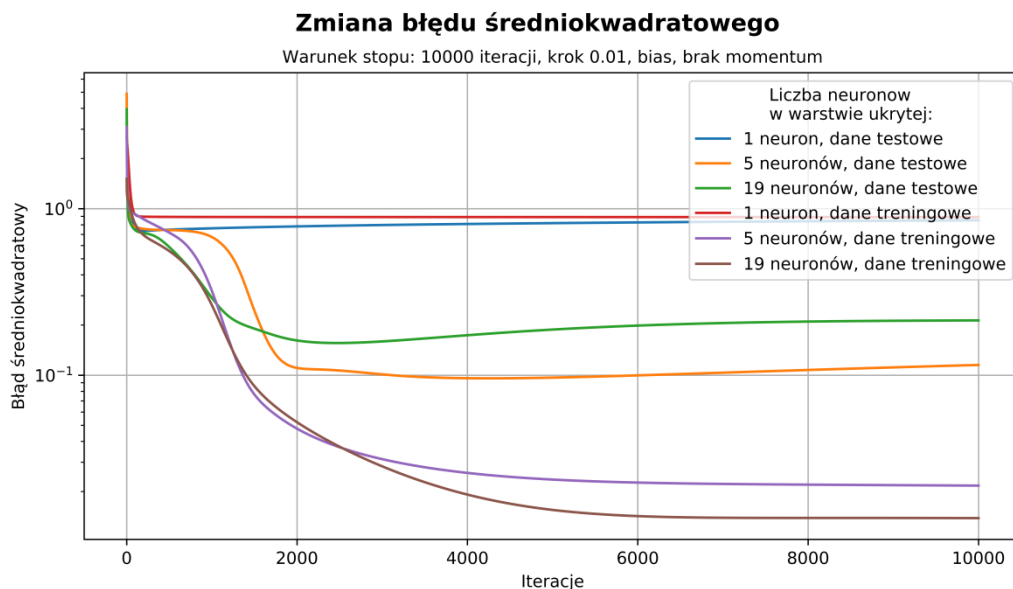
Ze względu na większą ilość punktów treningowych, sieć zdecydowanie lepiej aproksymuje dla pierwszych danych. W przypadku mniejszej ilości danych zdecydowanie większe znaczenie ma ilość neuronów w warstwie ukrytej, jednak wyniki wciąż nie są zadowalające. Dla pierwszych danych już 5 neuronów w warstwie ukrytej aproksymuje wystarczająco dobrze. Warto też zauważyć, że dla obu danych treningowych, 1 neuron zdecydowanie sobie nie radzi z sobie aproksymowaniem funkcji, gdyż jest w stanie zwrócić jedynie wartości sigmoidy, które nie mają prawa być bliskie oczekiwanym wartościom.

2. Zmiana błędu

Później porównaliśmy zmianę błędu w zależności od ilości iteracji, neuronów w warstwie ukrytej oraz danych, za pomocą których obliczaliśmy błąd średniokwadratowy. Nauka trwała do osiągnięcia 10 000 iteracji, sieć charakteryzowała się krokiem 0.01, biasem oraz brakiem momentum. Wykres dla pierwszych danych treningowych wygląda następująco:



Żaś dla drugich danych treningowych:



Jeden neuron w warstwie ukrytej ponownie sobie nie poradził z obliczeniami, zaś wyniki dla pięciu neuronów w warstwie ukrytej znowu zadawałające. Większa ilość punktów treningowych pozytywnie wpłynęła na wielkość błędu. W tym zadaniu wyraźnie widać wpływ zbyt wysokiego kroku nauki, który powoduje ogromną niestabilność podczas wyliczania błędu na początku. Co ciekawe, dla drugich danych treningowych to 5 neuronów uzyskało lepsze wyniki przy porównywaniu z danymi testowymi niż 19 neuronów – jest to spowodowane lepszym przystosowywaniem się sieci do danych treningowych, które nie pokrywają się z danymi testowymi, przez co błąd jest minimalnie większy (by lepiej pokazać różnicę, oś rzędnych przedstawiona jest w skali logarytmicznej).

3. Błąd średniokwadratowy i odchylenie standardowe.

Następnie zajęliśmy się sprawdzeniem wartości błędu średniokwadratowego (dla danych testowych) oraz odchyłeń standardowych (oznaczone jako OS) dla obu danych w zależności od ilości neuronów w warstwie ukrytej. Sieć uczyła się do 1000 epoki. z krokiem 0.1. biasem oraz bez momentum.

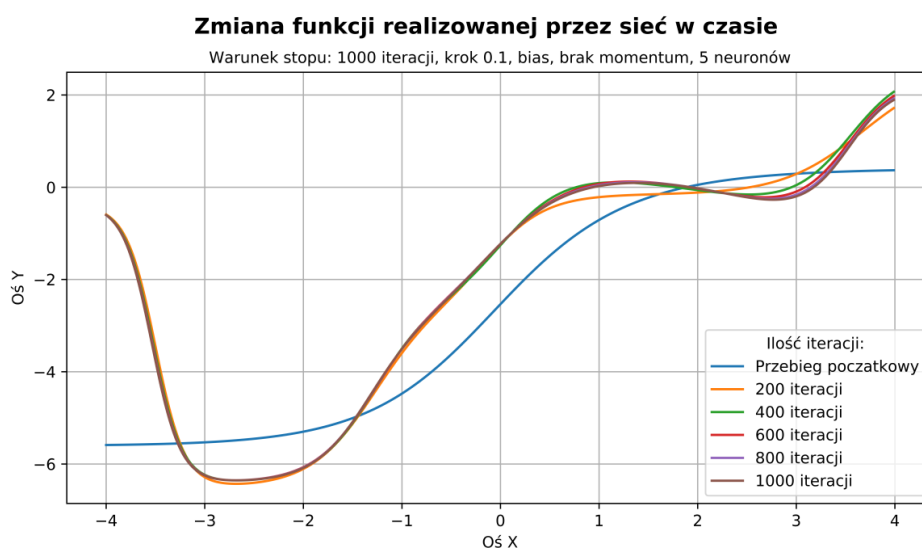
| Pierwsze dane treningowe | | | | |
|--------------------------|---------------------------|----------------------------|---------------------------|-------------------------|
| Ilość neuronów | Błąd dla danych testowych | OS dla danych treningowych | Błąd dla danych testowych | OS dla danych testowych |
| 1 | 1.0994 | 1.4827 | 0.5970 | 1.0926 |
| 2 | 0.9757 | 1.3969 | 0.5192 | 1.0190 |
| 3 | 0.2376 | 0.5617 | 0.1201 | 0.3956 |
| 4 | 0.0553 | 0.3326 | 0.0235 | 0.2166 |
| 5 | 0.0527 | 0.3244 | 0.0215 | 0.2063 |
| 6 | 0.0488 | 0.3124 | 0.0192 | 0.1940 |
| 7 | 0.0528 | 0.3244 | 0.0217 | 0.2050 |
| 8 | 0.0480 | 0.3096 | 0.0155 | 0.1754 |
| 9 | 0.0445 | 0.2983 | 0.0129 | 0.1606 |
| 10 | 0.0471 | 0.3069 | 0.0168 | 0.1816 |
| 11 | 0.0501 | 0.3160 | 0.0184 | 0.1905 |
| 12 | 0.0484 | 0.3110 | 0.0186 | 0.1914 |
| 13 | 0.0475 | 0.3081 | 0.0172 | 0.1838 |
| 14 | 0.0500 | 0.3159 | 0.0177 | 0.1864 |
| 15 | 0.0464 | 0.3047 | 0.0140 | 0.1674 |
| 16 | 0.0486 | 0.3117 | 0.0160 | 0.1778 |
| 17 | 0.0481 | 0.3101 | 0.0154 | 0.1746 |
| 18 | 0.0508 | 0.3186 | 0.0175 | 0.1868 |
| 19 | 0.0486 | 0.3116 | 0.0162 | 0.1791 |
| 20 | 0.0510 | 0.3190 | 0.0176 | 0.1870 |

| Drugie dane treningowe | | | | |
|------------------------|------------------------------|----------------------------|---------------------------|-------------------------|
| Ilość neuronów | Błąd dla danych treningowych | OS dla danych treningowych | Błąd dla danych testowych | OS dla danych testowych |
| 1 | 0.9009 | 1.3423 | 0.9050 | 1.3453 |
| 2 | 0.5598 | 0.9142 | 0.5191 | 0.9444 |
| 3 | 0.3873 | 0.6923 | 0.3818 | 0.8188 |
| 4 | 0.0289 | 0.2400 | 0.2184 | 0.6591 |
| 5 | 0.0266 | 0.2298 | 0.2114 | 0.6444 |
| 6 | 0.0251 | 0.2239 | 0.2863 | 0.7505 |
| 7 | 0.0256 | 0.2261 | 0.2707 | 0.7329 |
| 8 | 0.0257 | 0.2265 | 0.2669 | 0.7305 |
| 9 | 0.0278 | 0.2356 | 0.2652 | 0.7275 |
| 10 | 0.0302 | 0.2459 | 0.2616 | 0.7231 |
| 11 | 0.0313 | 0.2498 | 0.2695 | 0.7331 |
| 12 | 0.0358 | 0.2663 | 0.2985 | 0.7704 |
| 13 | 0.0385 | 0.2775 | 0.2986 | 0.7719 |
| 14 | 0.0539 | 0.3249 | 0.3078 | 0.7812 |
| 15 | 0.0422 | 0.2855 | 0.3035 | 0.7786 |
| 16 | 0.0678 | 0.3596 | 0.3476 | 0.8281 |
| 17 | 0.0912 | 0.4229 | 0.3795 | 0.8682 |
| 18 | 0.0566 | 0.3276 | 0.2983 | 0.7721 |
| 19 | 0.0489 | 0.3085 | 0.2986 | 0.7722 |
| 20 | 0.0633 | 0.3419 | 0.3142 | 0.7919 |

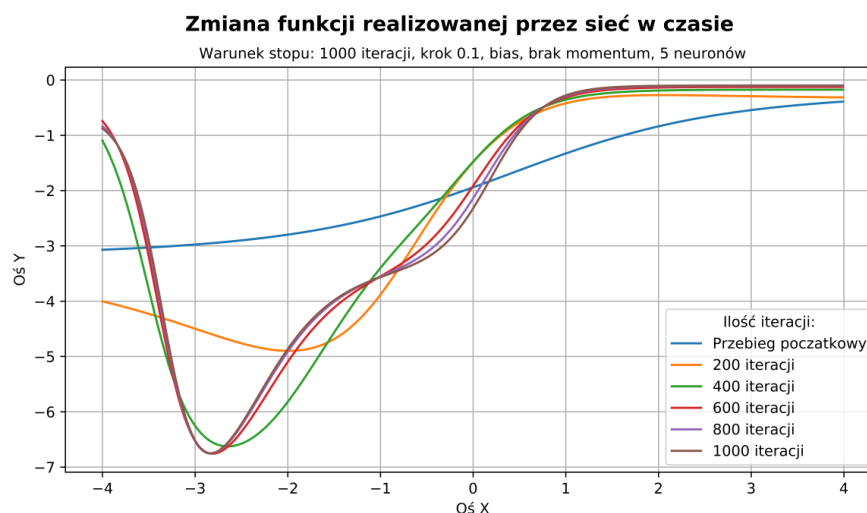
Po raz kolejny okazuje się, że dla pierwszych danych treningowych 5 neuronów osiąga już wystarczająco zadowalające wyniki. by nie było potrzeby dawać sieci więcej neuronów w warstwie ukrytej. Co ciekawe, dla drugich danych testowych, więcej neuronów niż 5 powoduje już zbyt dużą rozbieżność pomiędzy oczekiwanym a otrzymanym rezultatem.

4. Zmiana realizowanej funkcji

Na samym końcu zajęliśmy się obserwacją, w jaki sposób realizowana funkcja zmienia się podczas nauki. Sieć uczyła się do osiągnięcia 1000 epoki, posiadała 5 neuronów w warstwie ukrytej, krok nauki 0.1 oraz brak momentum. Na wykresie widoczna jest funkcja z 6 okresów czasu równo oddalonych od siebie. Dla pierwszych danych treningowych wykres wygląda następująco:



Zaś dla drugich danych treningowych:



W przypadku pierwszych danych treningowych, już po 200 epokach wyniki są całkiem zadowalające, i nie zmieniają się znacznie nawet gdy dojdą do epoki 1000. Przy drugich danych sieć zdecydowanie wolniej się uczy, i dopiero podczas ostatnich dwóch okresów nauki wyniki są już do siebie względnie zbliżone, co znowu pokazuje jak ważna jest ilość danych treningowych.