Współczesne techniki heurystyczne

Sprawozdanie końcowe

PB2 Zastosowanie sieci neuronowej w zadaniu aproksymacyjnym

Rafał Koguciuk, Julita Ołtusek

1. Cel projektu

Celem projektu było zaprojektowanie sztucznej sieci neuronowej do aproksymacji ciągłej funkcji dwuwymiarowej. Sieć ta powinna nauczyć się określonej liczby punktów, a następnie prawidłowo (z akceptowalnym błędem) aproksymować punkty z drugiego zbioru (testowego). Projekt powinien obejmować wypróbowanie różnych struktur sieci (liczba warstw ukrytych oraz liczba neuronów w warstwie) oraz określenie struktury optymalnej dla problemu.

2. Opis problemu

Aproksymacja jest problemem przybliżania funkcji, polegającym na wyznaczeniu dla danej funkcji F(x) takich funkcji f(x), które w określonym sensie najlepiej przybliżają funkcję F(x) dla danego zbioru wejściowego. Stosuje się ją po to, by skomplikowane byty zastępować prostszymi, z pewnym akceptowalnym błędem przybliżenia.
Jednym ze sposobów na aproksymację funkcji jest wykorzystanie narzędzia obliczeniowego jakim jest sztuczna sieć neuronowa. Imituje ona w sposób uproszczony działanie ludzkiego mózgu, którego neurony były inspiracją do stworzenia sieci neuronowych.
W zadaniu aproksymacji funkcji mamy do czynienia z regresją, czyli metodą statystyczną, która polega na przewidywaniu nieznanych wartości pewnej zmiennej na podstawie znanych wartości innych zmiennych. Jest to jedna z dwóch głównych metod używanych do uczenia nadzorowanego. Różni się od drugiej z nich, tym, że przewidywana zmienna jest typu ciągłego, a nie dyskretnego, co ma miejsce w przypadku klasyfikacji. Jest to bardzo istotne, ponieważ ocenianie modelu regresyjnego wygląda zupełnie inaczej, niż klasyfikacyjnego. W naszym przypadku musieliśmy użyć takiej metryki jakości rozwiązania, która pozwoliła ocenić odległość poszczególnych przewidzianych wartości od rozwiązania prawidłowego, gdyż nie było możliwe dokładne trafienie w odpowiednią wartość, kiedy możliwych wartości jest nieskończenie wiele.
Zgodnie z uniwersalnym twierdzeniem o aproksymacji, jednokierunkowa sieć neuronowa posiadająca jedną ukrytą warstwę, zawierającą skończoną liczbę neuronów, jest w stanie aproksymować dowolną funkcję ciągłą, przy założeniach, że funkcja aktywacji jest rosnąca i ograniczona. Nie ma natomiast pewności, że

znalezienie takiej sieci i jej trening są możliwe w rozsądnym czasie. Z tego względu jest to dość trudny problem, który nie w każdym przypadku udaje się rozwiązać.

3. Założenia

Ustaliliśmy, że trójwymiarowa funkcja, która została przybliżona za pomocą sztucznej sieci neuronowej jest postaci:

$$z = 5 * sin(0.1 * x) * cos(0.5y)$$

Wynikiem projektu powinno być znalezienie odpowiedniej struktury sieci neuronowej, która będzie rozwiązywała przedstawiony problem z najniższym błędem.

4. Szczegółowy opis rozwiązania

4.1. Dane

4.1.1. Generowanie danych

Dane zostały wygenerowane automatycznie, przy użyciu prostego skryptu Pythonowego, obliczającego wartość funkcji $z = f(x,y)$ dla 10000 punktów. Zakresy wartości poszczególnych współrzędnych powierzchni są następujące:

$$x \in [-5;\ 4.9]$$
$$y \in [-5\ ;\ 4.9]$$
$$z \in [-2.4\ ;\ 2.4]$$

Po wygenerowaniu dane zostały zapisane w postaci obiektu DataFrame z biblioteki pandas do pliku csv, dzięki czemu można je obejrzeć w Excelu, a także w łatwy sposób załadować do programu w Pythonie.
Do aproksymacji zostały również wykorzystane dane zaszumione. Szum był generowany poprzez dodanie do sygnału pierwotnego składnika wyznaczonego za pomocą rozkładu normalnego o średniej 0 i odchyleniu standardowym 0.2.

4.1.2. Przygotowanie danych

Na początku warto było dokonać pierwszych spostrzeżeń, co do danych, np tego jaki jest ich rozkład, maksymalna, minimalna, średnia wartość każdego parametru i innych istotnych miar statystycznych.
Ponieważ mamy do czynienia z kompletnymi danymi numerycznymi o równomiernym rozkładzie, nie była konieczna wstępna obróbka danych przed podaniem ich do modelu. Dane były więc wczytane z pliku csv do obiektu Pandas.DataFrame, na którym były wykonywane dalsze operacje.

4.1.3. Zbiory danych

Sieć neuronowa jako zbiór danych do treningu dostała 70% losowo wybranych punktów z wygenerowanego wcześniej zbioru. Zbiór ten został dodatkowo podzielony na zbiór treningowy i zbiór walidacyjny w proporcji 8:2. Drugi z nich posłużył do testowania podczas strojenia hiperparametrów sieci. Po zbudowaniu zadowalającej nas architektury sieci, czyli takiej, która dała odpowiednio niski poziom błędu wyznaczonego metryką MSE, model został oceniony na podstawie skuteczności przewidywania wartości punktów z pozostałej (30%) części zbioru, która posłużyła jako zbiór testowy. Taki zabieg miał na celu ocenę faktyczną modelu, gdyż dane te nie są znane modelowi wcześniej, w przeciwieństwie do zbioru treningowego i walidacyjnego, do których model mógł się dopasować podczas treningu. W przypadku, kiedy sieć nie byłaby dobrze regularyzowana, błąd na zbiorze testowym znacznie przewyższałby te obliczone na zbiorze treningowym oraz walidacyjnym.

4.2. Architektura rozwiązania

Wykorzystana została klasyczna, jednokierunkowa, płytka sieć neuronowa, ze wsteczną propagacją błędu, przy użyciu metody największego spadku. Sieć miała postać perceptronu wielowarstwowego, a funkcja aktywacji neuronów w warstwach ukrytych była nieliniową funkcją tangensa hiperbolicznego. Liczbę warstw dopasowaliśmy w trakcie budowy modelu. Kolejnymi parametrami, które zostały dopasowane, są m.in. liczba neuronów w warstwie, wskaźnik uczenia (eng. learning rate, rozmiar kroku do optymalizacji) oraz liczba epok uczenia.

4.2.1. Liczba warstw sieci

Liczba warstw zależy głównie od charakteru danych wejściowych. Zwykle jest to mała wartość, ze względu na wydłużający się czas obliczeń. Zbyt wiele warstw może skutkować pogorszeniu procesu uczenia.

4.2.2. Liczba neuronów w warstwie

Liczba neuronów podobnie jak liczba warstw z reguł zależy od liczby i charakteru danych wejściowych. Zbyt duża ich liczba może skutkować tym, że czas potrzebny na obliczenia znacznie się zwiększy, ale również może doprowadzić do wystąpienia zjawiska przeuczenia, które polega na tym, że sieć za dobrze dopasuje się do danych wejściowych lecz utraci zdolność uogólniania na zbiorze testującym. Zbyt mała ilość neuronów może powodować słabą zdolność do dopasowywania się modelu do funkcji.

4.2.3. Wskaźnik uczenia

Zazwyczaj przyjmuje on wartość z zakresu 0.0001~0.1 i bezpośrednio wpływa na szybkość zbliżania się do rozwiązania optymalnego. Źle dobrana wartość może skutkować tym, że sieć będzie zbyt wolno dochodzić do rozwiązania, z drugiej strony może także gwałtownie zmieniać parametry sieci, przez co trudno będzie znaleźć optimum.

4.2.4. Momentum

Współczynnik ten nadaje procesowi uczenia pewną bezwładność, aby uchronić się od gwałtownych zmian wartości wagowych, niekorzystnych dla uczenia. Dzięki niemu zmiana wag w bieżącym cyklu zależy również od zmiany jaka nastąpiła w cyklu poprzednim i pośrednio w cyklach jeszcze wcześniejszych. Reguluje on wpływ zmiany wag na proces uczenia. Pomaga także w opuszczeniu niegłębokich minimów lokalnych i poprawia efektywność uczenia w monotonicznych przedziałach funkcji energetycznej. Przyjmuje on wartości z przedziału 0 - 1, przy czym najczęściej stosowane to 0.5 - 1.

4.2.5. Liczba epok

Parametr ten wskazuje, ile iteracji algorytmu uczenia sieć ma wykonać. W teorii im większa liczba epok tym lepiej, gdyż sieć może się uczyć coraz lepiej. Zazwyczaj jednak jest tak, że po pewnej liczbie iteracji, skuteczność uczenia się nie wzrasta, lub jest na tyle mała, że nie warto prowadzić więcej iteracji algorytmu.

4.2.6. Funkcja aktywacji

Bardzo ważnym elementem sieci neuronowej jest funkcja aktywacji, według której obliczana jest wartość wyjścia neuronów sieci neuronowej. Determinuje ona czy dany neuron jest aktywowany czy nie, innymi słowy czy wyjście neurony jest brane pod uwagę w następnych warstwach czy nie.

4.3. Metryki skuteczności rozwiązania

Skuteczność rozwiązania była mierzona za pomocą błędu średniokwadratowego. Jest on wartością oczekiwaną kwadratu błędu, czyli różnicy pomiędzy estymatorem i wartością estymowaną. Dzięki niemu, było można ocenić dopasowanie modelu, powstałego w wyniku estymacji, do danych rzeczywistych. Z uwagi na fakt, że wykorzystuje się średnią błędu estymacji, metryka ta jest odporna na liczbę próbek, na podstawie których model jest aproksymowany.

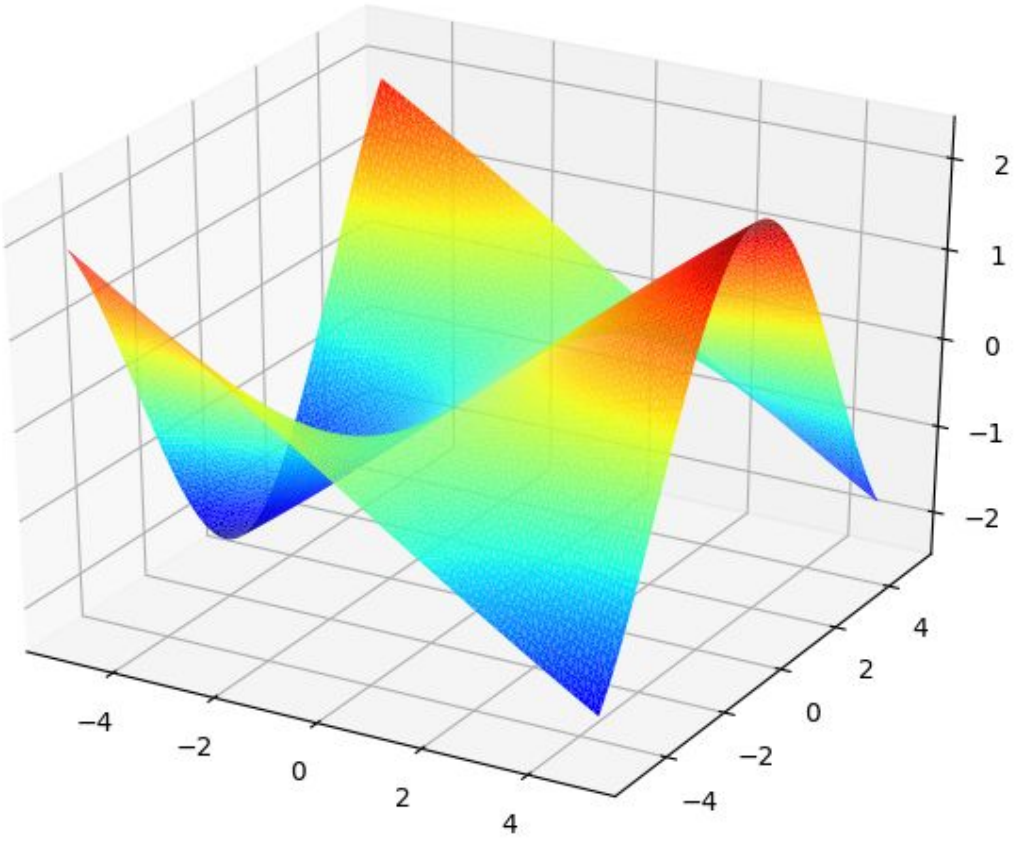
5. Wyniki badań eksperymentalnych

Uczenie sieci zostało przeprowadzone dla dwóch przypadków, dla danych testowych oryginalnych oraz dla zaszumionych danych testowych. Zostały przeprowadzone liczne eksperymenty, w wyniku których na podstawie miary błędu, niezależnej od liczby punktów, zostały znalezione hiperparametry, dla których sieć w najlepszy sposób zaaproksymowała funkcję. Model był trenowany przy użyciu każdego z zestawów parametrów niezależnie 10 razy, a następnie wyniki z 10 prób zostały uśrednione. Oprócz średniego błędu zmierzone zostały odchylenia standardowe. Model ma trzy warstwy neuronów. Dwie pierwsze warstwy (wejściowa i ukryta) mogą posiadać więcej niż jeden neuron, a ostania trzecia (wyjściowa) warstwa zawiera tylko jeden neuron. Założyliśmy, że taka ilość warstw wystarcza, by zbudować dokładny model dla tego problemu, co zostało potwierdzone eksperymentalnie. Uśrednione wartości funkcji straty dla wszystkich zbiorów okazały się zadowalające i nie było

konieczności dodawać kolejnych warstw ukrytych. Warto wspomnieć, że modele były trenowane przy użyciu optymalizatora SGD (Stochastic Gradient Descent), który wspiera użycie takich parametrów jak współczynnik uczenia i momentum. Jednorazowo sieć przyjmowała partie danych o rozmiarze 10, co przyspieszyło proces trenowania, a także powstrzymało sieć przed wahaniami spowodowanymi silnym dopasowywaniem się do pojedynczych przykładów.

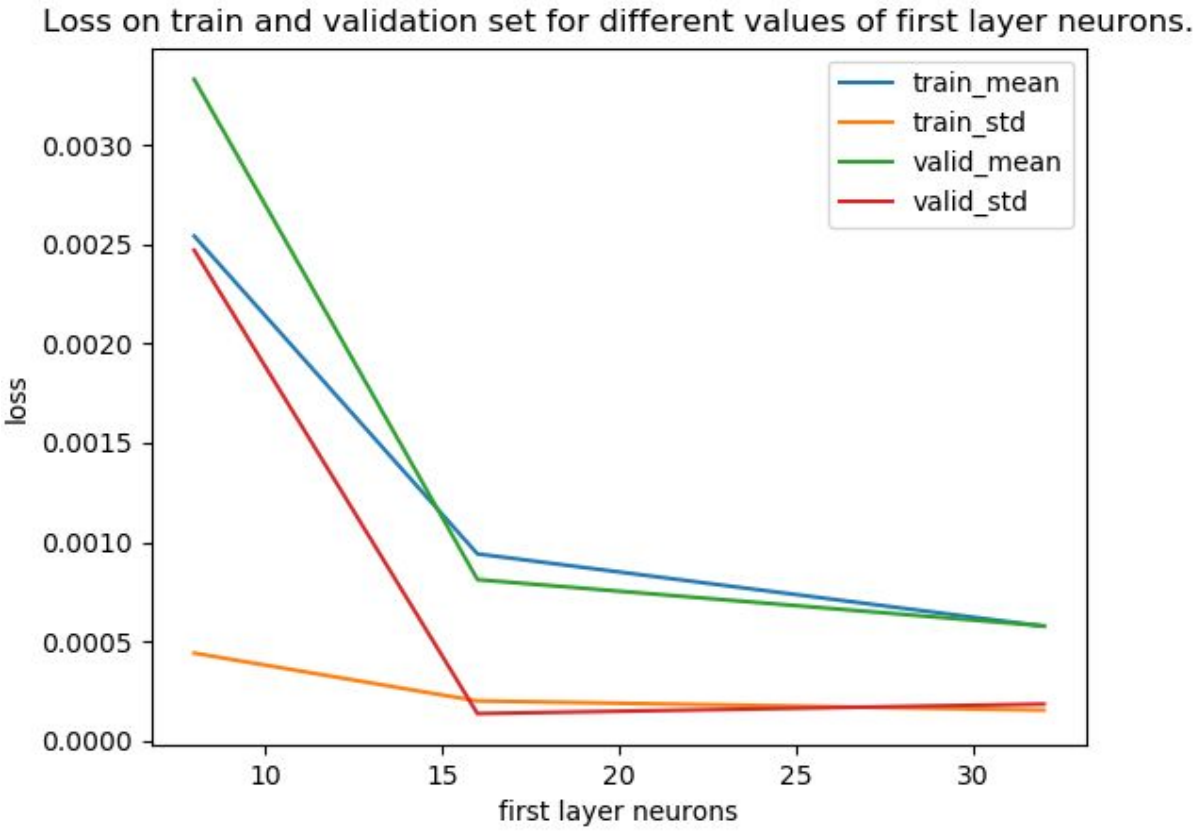
5.1. Dopasowanie hiperparametrów dla danych oryginalnych

Powierzchnia, tworzona przez dane, użyte podczas eksperymentów została zaprezentowana na rysunku poniżej.



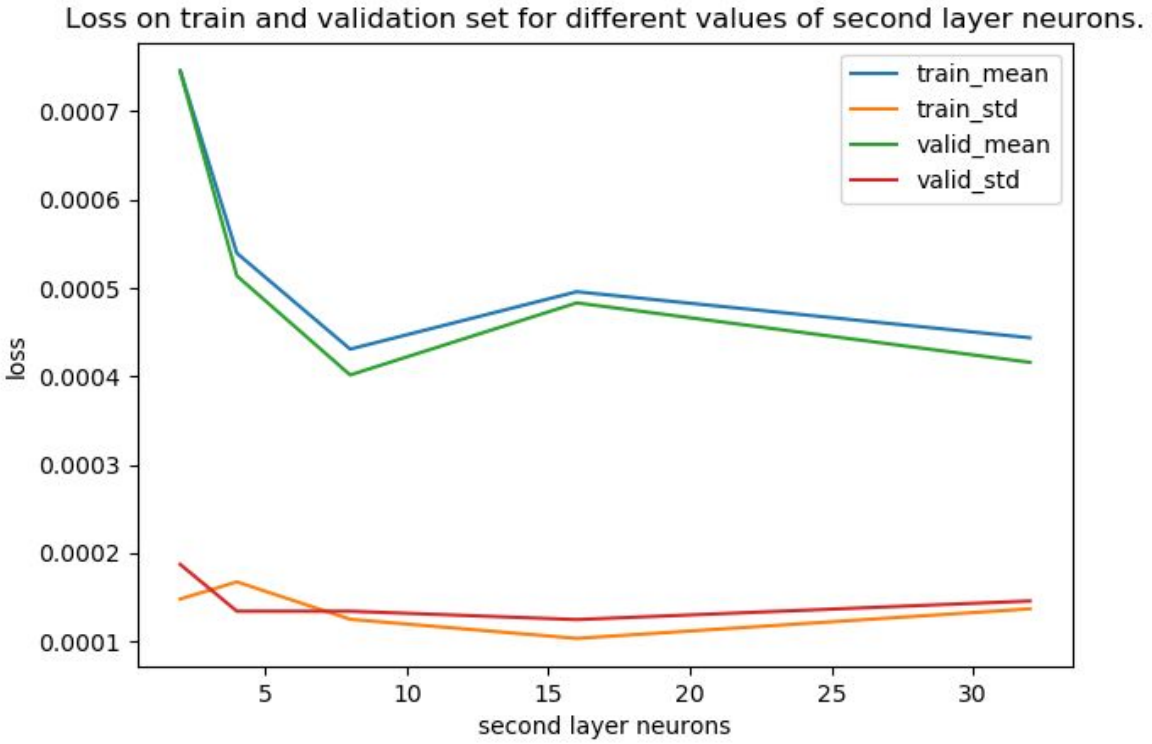
Dane te zgodnie z opisem, zostały podzielone losowo na trzy zbiory: testowy, treningowy i walidacyjny. Następnie przystąpiono do eksperymentów, które polegały na wyznaczeniu najlepszych parametrów sieci uczącej. Dla każdego parametru zbadano zależność jego wartości od wartości średnich błędów i ich odchyleń standardowych. Wyniki przedstawiono za pomocą wykresów.

5.1.1. Liczba neuronów w pierwszej warstwie



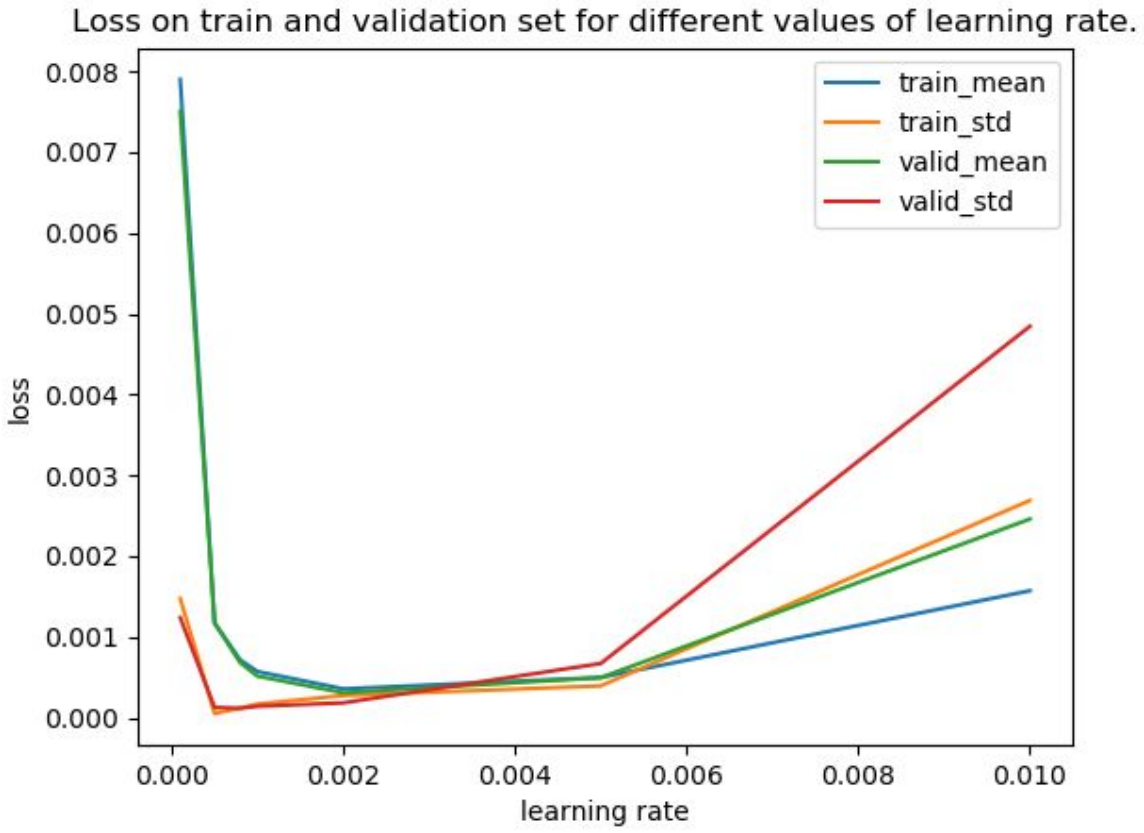
Powyższy wykres przedstawia zależność wartości średniego błędu aproksymacji i jego odchylenia standardowego do ilości neuronów w pierwszej, wejściowej warstwie modelu. Z wykresu wynika, że zwiększanie liczby neuronów w pierwszej warstwie (wejściowej) przynosi pozytywny rezultat zmniejszenia średniego błędu aproksymacji, lecz do pewnego momentu. Możemy zaobserwować, że dla liczby neuronów większej niż 16, zysk ze zwiększenia liczby neuronów nieco maleje. Uznaliśmy, że 32 jest najlepszą liczbą neurónów w pierwszej warstwie, gdyż liczba neuronów w sieci nadal nie jest aż tak duża, żeby powodować przetrenowanie bądź znaczące wydłużenie czasu treningu. Nie zaobserwowaliśmy jednak zjawiska przeuczenia sieci, tj utraty zdolności uogólniania na zbiorze testującym, być może udałoby się zaobserwować dla dalszego wzrostu liczby neuronów w warstwie.

5.1.2. Liczba neuronów w drugiej warstwie



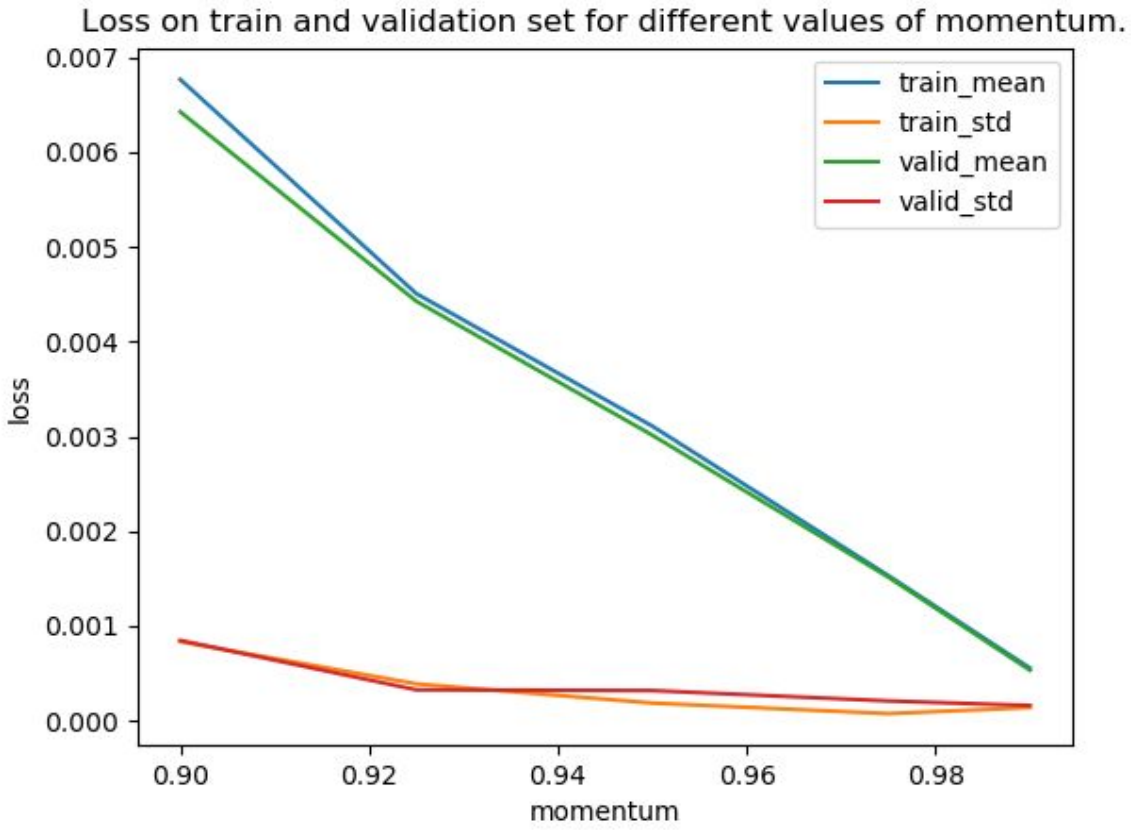
Wykres przedstawiony powyżej, ilustruje zależność wartości średniego błędu aproksymacji i jego odchylenia standardowego na zbiorze trenującym i walidacyjnym od liczby neuronów w drugiej warstwie modelu (pierwszej i jedynej warstwie ukrytej). Możemy zaobserwować, że na początku, zwiększanie liczby neuronów w warstwie przynosi pozytywny efekt tj. spadek średniego błędu i odchylenia standardowego. Później jednak, dla większych wartości liczby neuronów w warstwie, nie odnotowaliśmy spadku wartości średniego błędu, a wzrost odchylenia standardowego na obu zbiorach. Za wartość optymalną uznaliśmy 8.

5.1.3. Wskaźnik uczenia



Wykres powyżej przedstawia zależność wartości średniego błędu i jego odchylenia standardowego zbiorów trenującego i walidacyjnego od wartości wskaźnika współczynnika uczenia. Dla bardzo małych wartości wskaźnika uczenia sieć wolno się uczy, przez co błędy są duże. Z kolei duża wartość wskaźnika powoduje gwałtowne zmiany parametrów sieci, co prowadzi do zmienności działania sieci i trudności w znalezieniu optimum.
Optymalnymi wartościami wskaźnika są te z przedziału (0.0005; 0.005). To dla tych wartości zanotowaliśmy najlepszą skuteczność aproksymacji funkcji.

5.1.4. Momentum



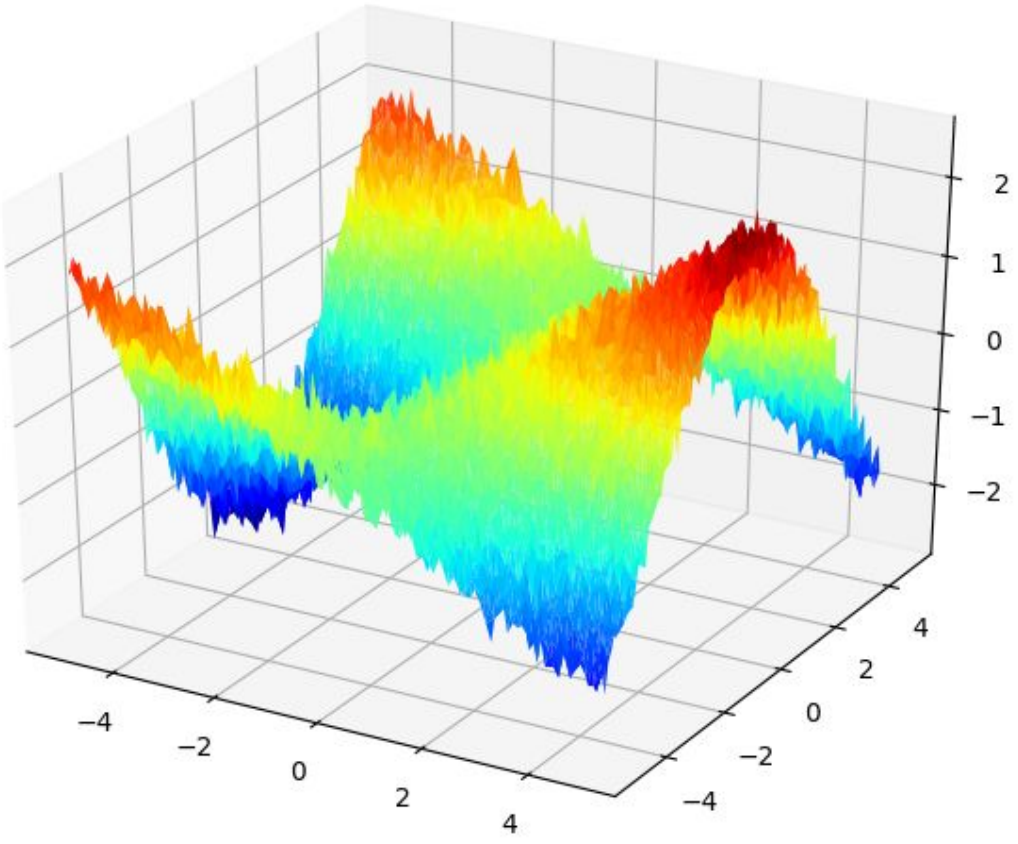
Wykres przedstawiony powyżej zawiera informacje na temat zależności wartości średnich błędów aproksymacji i ich odchyleń standardowych od wartości wskaźnika momentum. Możemy zaobserwować, że wraz ze wzrostem wartości wskaźnika, średni błąd maleje jednoznacznie, a odchylenie standardowe utrzymuje się na stosunkowo niskim poziomie. Wskaźnik ten nadaje procesowi uczenia pewną bezwładność, dzięki temu model nie zmienia tak gwałtownie swoich wartości wag. Za optymalną wartość wskaźnika momentum uznaliśmy 0.99, ponieważ dla tej wartości średni błąd okazał się być najmniejszy.

5.1.5. Liczba epok

Wykres przedstawiony powyżej, ilustruje zależność średniego błędu aproksymacji i odchylenie standardowe na zbiorach trenującym i walidacyjnym od liczby epok. Zgodnie z przypuszczeniami, możemy zaobserwować monotoniczny spadek średniego błędu i jego odchylenia wraz ze wzrostem liczby epok. Zwiększanie liczby epok w nieskończoność w teorii mogłoby przynieść znaczną poprawę jakości aproksymacji, ale również zwiększenia czasu wykonania algorytmu. Jako najlepszą wartość uznaliśmy liczbę epok równą 20, pozwala to na uzyskanie niskiego średniego błędu w akceptowalnym czasie wykonania programu.

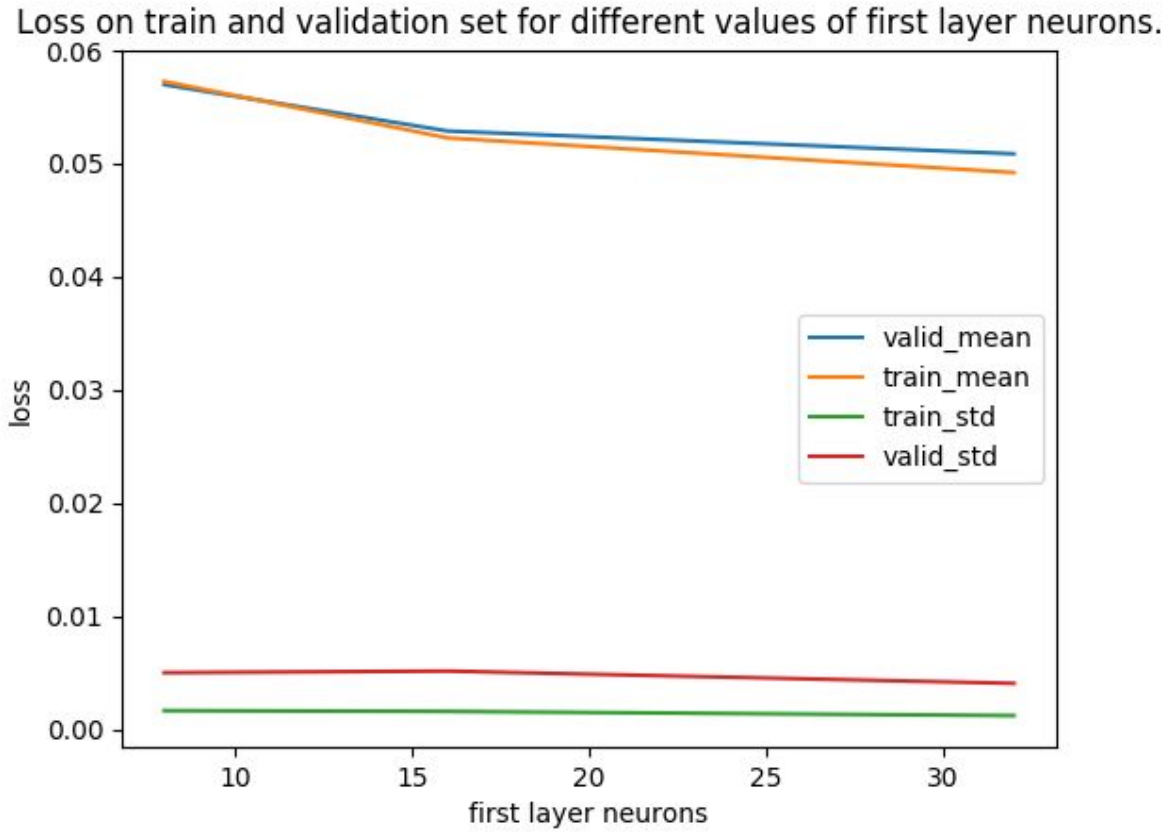
5.2. Dopasowanie hiperparametrów do danych zaszumionych

Powierzchnia, tworzona przez dane zaszumione, użyte podczas eksperymentów zostały zaprezentowane na rysunku poniżej.



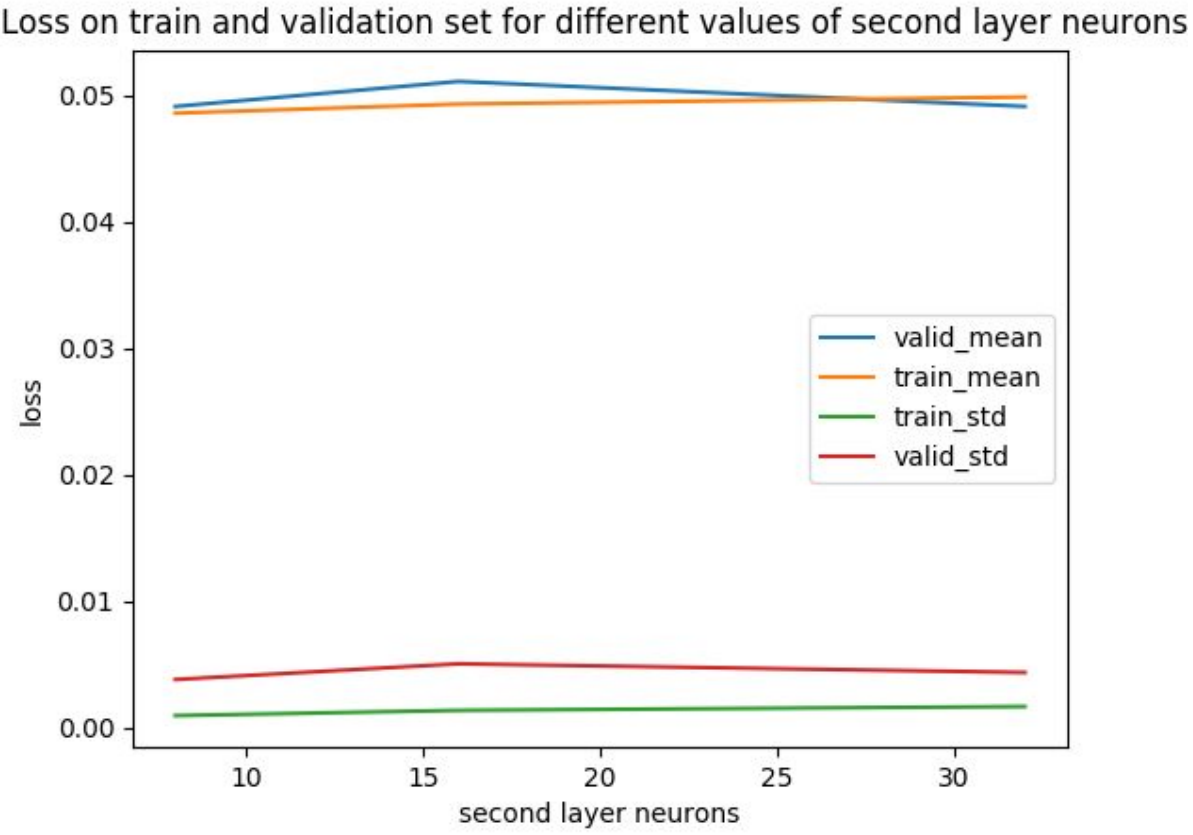
Dane te, tak samo jak w przypadku danych oryginalnych - niezaszumionych, zostały podzielone losowo na trzy zbiory. Przystąpiono później do prób dobrania optymalnych parametrów sieci neuronowej, by uzyskać jak najbardziej dokładny model, który będzie aproksymował powyższą powierzchnię. Wyniki eksperymentów zostały przedstawione poniżej.

5.2.1. Liczba neuronów w pierwszej warstwie



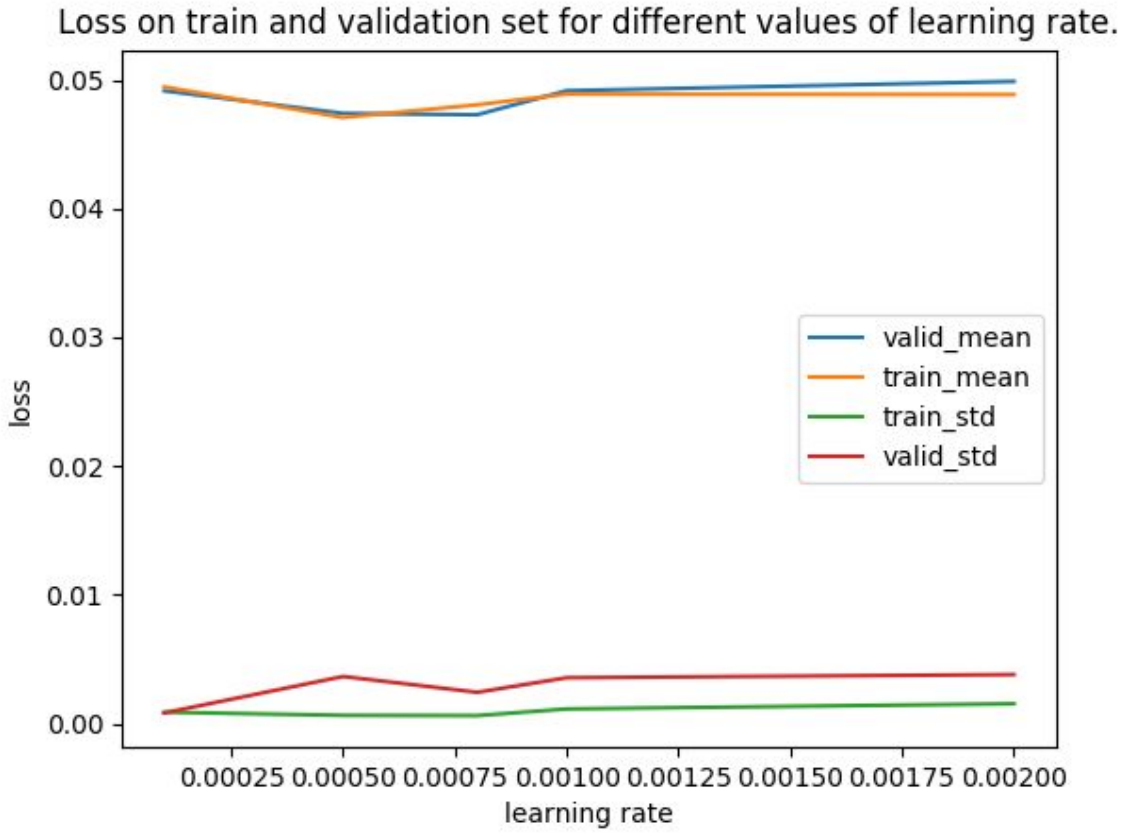
Powyższy wykres przedstawia zależność wartości średniego błędu aproksymacji i jego odchylenia standardowego do ilości neuronów w pierwszej, wejściowej warstwie modelu. Z wykresu wynika, że zwiększanie liczby neuronów w pierwszej warstwie (wejściowej) nie przynosi znaczącej poprawy jakości aproksymacji przez model. Możemy zaobserwować jedynie bardzo delikatną poprawę.

5.2.2. Liczba neuronów w drugiej warstwie



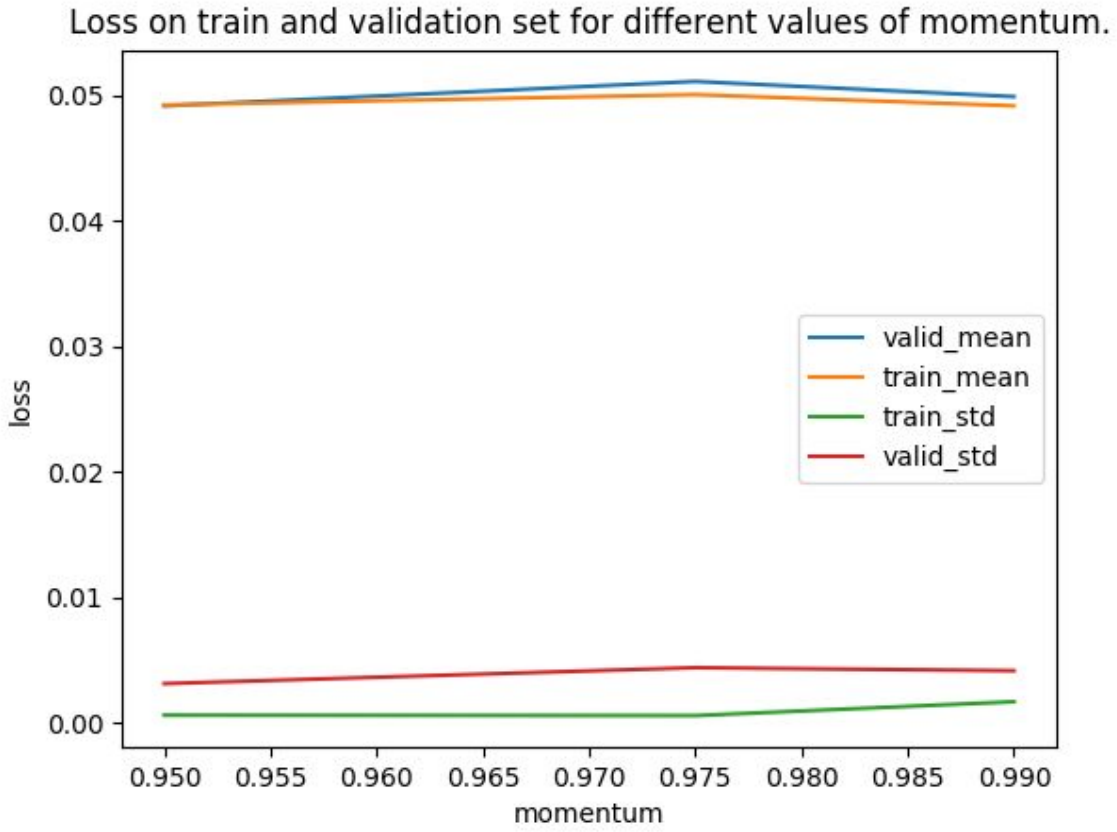
Wykres przedstawiony powyżej, ilustruje zależność wartości średniego błędu aproksymacji i jego odchylenia standardowego na zbiorze trenującym i walidacyjnym od liczby neuronów w drugiej warstwie modelu (pierwszej i jedynej warstwie ukrytej). W tym przypadku zwiększanie liczby neuronów w warstwie, podobnie jak w przypadku liczby w pierwszej warstwie, nie przynosi poprawy. Średni błąd kwadratowy aproksymacji utrzymuje się mniej więcej na takim samym poziomie dla różnej liczby neuronów w warstwie ukrytej.

5.2.3. Wskaźnik uczenia



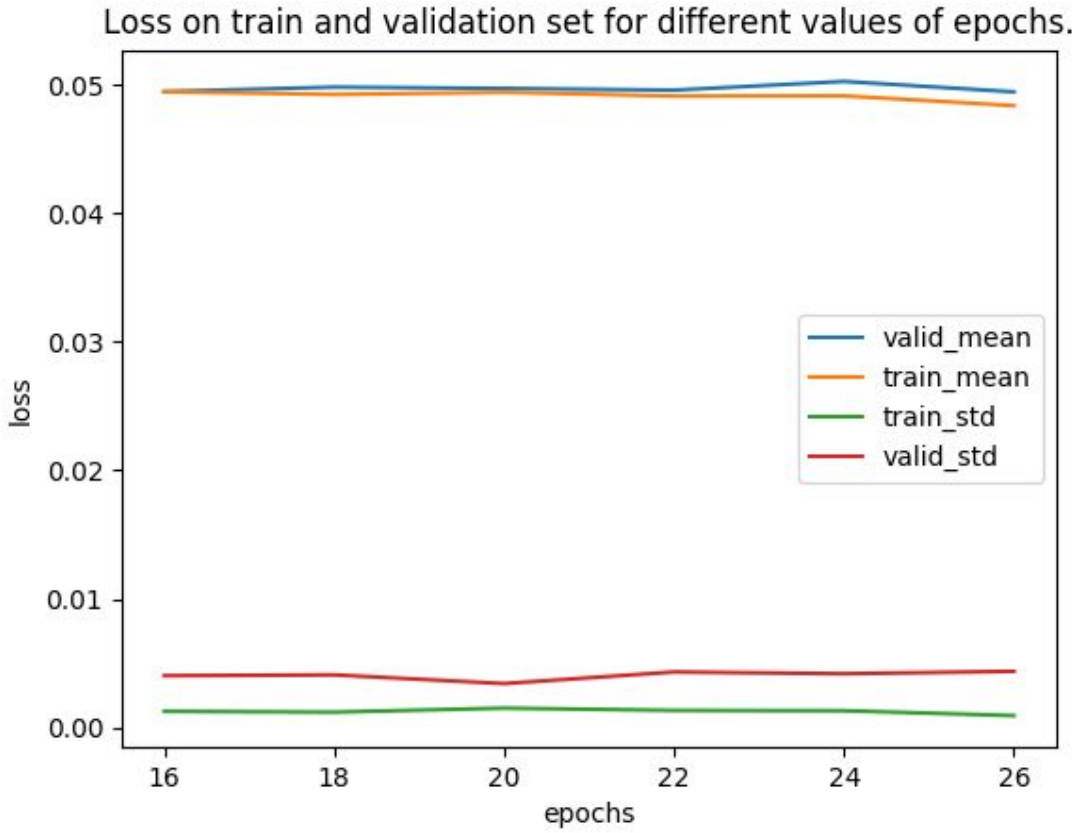
Wykres powyżej przedstawia zależność wartości średniego błędu i jego odchylenia standardowego zbiorów trenującego i walidacyjnego od wartości wskaźnika współczynnika uczenia. W tym przypadku również możemy zaobserwować jedynie delikatną poprawę średniego błędu aproksymacji zwiększając wartość wskaźnika uczenia. Odchylenie standardowe błędu na zbiorze walidacyjnym zdecydowanie wzrosło w porównaniu do odchylenia standardowego błędu na zbiorze trenującym.

5.2.4. Momentum



Wykres przedstawiony powyżej zawiera informacje na temat zależności wartości średnich błędów aproksymacji i ich odchyleń standardowych od wartości wskaźnika momentum. Zgodnie z wykresami poprzednio zaprezentowanymi, zmiana parametru momentum nie przynosi poprawy średniego błędu aproksymacji wykorzystując do uczenia dane zaszumione. Możemy zaobserwować, że odchylenie standardowe błędu zbioru danych walidacyjnych jest zdecydowanie większe od tego odchylenia standardowego błędu zbioru trenującego.

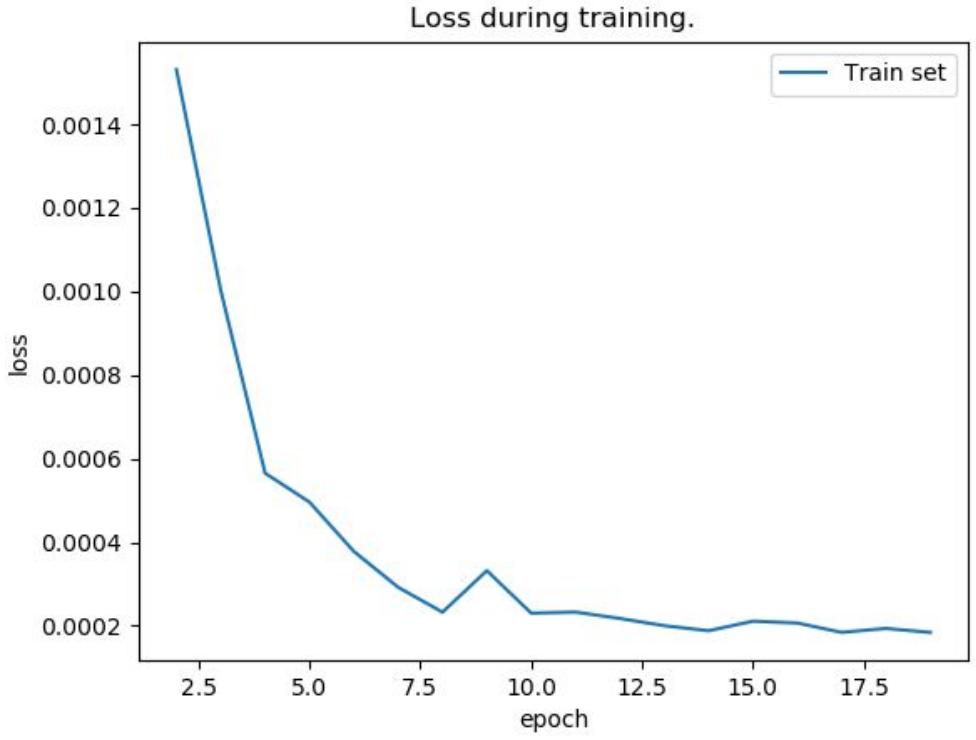
5.2.5. Liczba epok

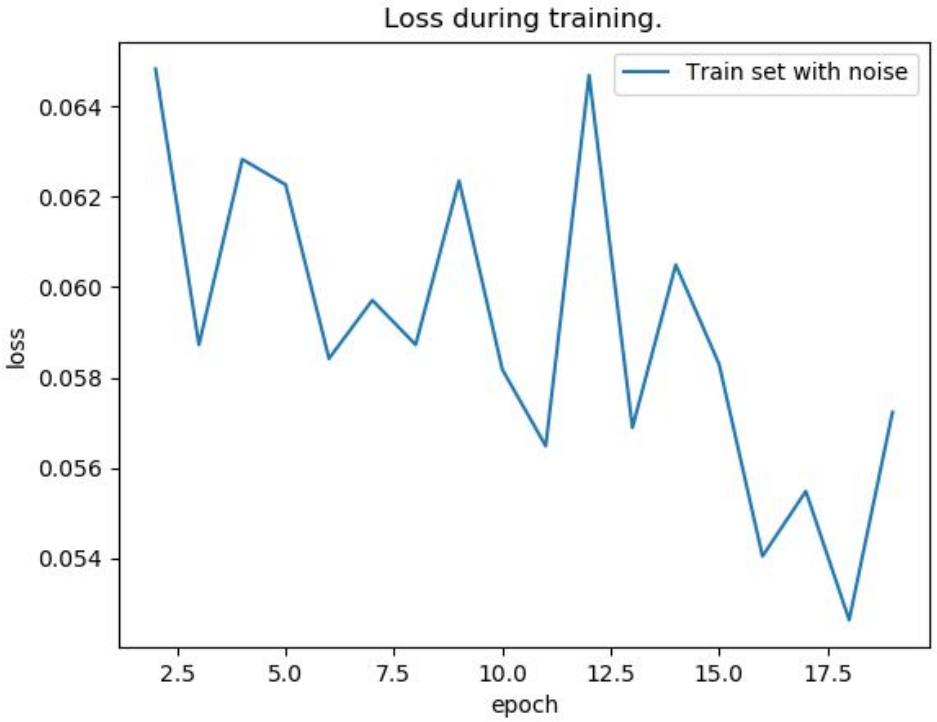


Wykres przedstawiony powyżej, ilustruje zależność średniego błędu aproksymacji i odchylenie standardowe na zbiorach trenującym i walidacyjnym od liczby epok. Zgodnie z przypuszczeniami, możemy zaobserwować brak poprawy błędu aproksymacji wraz ze zmianą liczby epok oraz wyższą wartość odchylenia standardowego błędu aproksymacji obliczonego dla zbioru walidacyjnego niż dla zbioru trenującego.

5.3. Analiza procesu uczenia przy użyciu najlepszych parametrów.

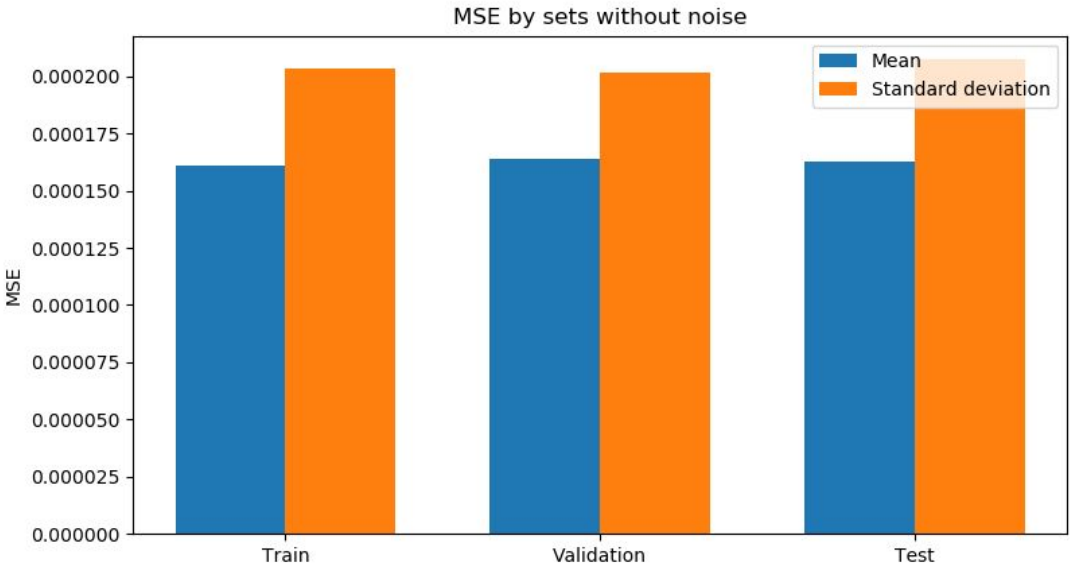
Po eksperymentach związanych z dopasowywaniem hiperparametrów sieci nadeszła pora na analizę procesu uczenia modelu, który zbudowaliśmy. Na poniższych wykresach widoczne są zmiany wartości funkcji straty dla zbioru treningowego bez szumu i z szumem, w zależności od numeru epoki. Można zauważyć, że na zbiorze zaszumionym są bardzo duże wahania średniej wartości błędów, podczas gdy sieć uczona na zbiorze pierwotnym zachowuje się zgodnie z oczekiwaniami i dopasowuje coraz lepiej do funkcji, którą ma a zadanie aproksymować. Zapisaliśmy te wartości na dwóch różnych wykresach, ponieważ rząd wielkości bardzo się różni i niemożliwe było uchwycenie tego na jednym wykresie w taki sposób, aby móc wyciągać interesujące wnioski.

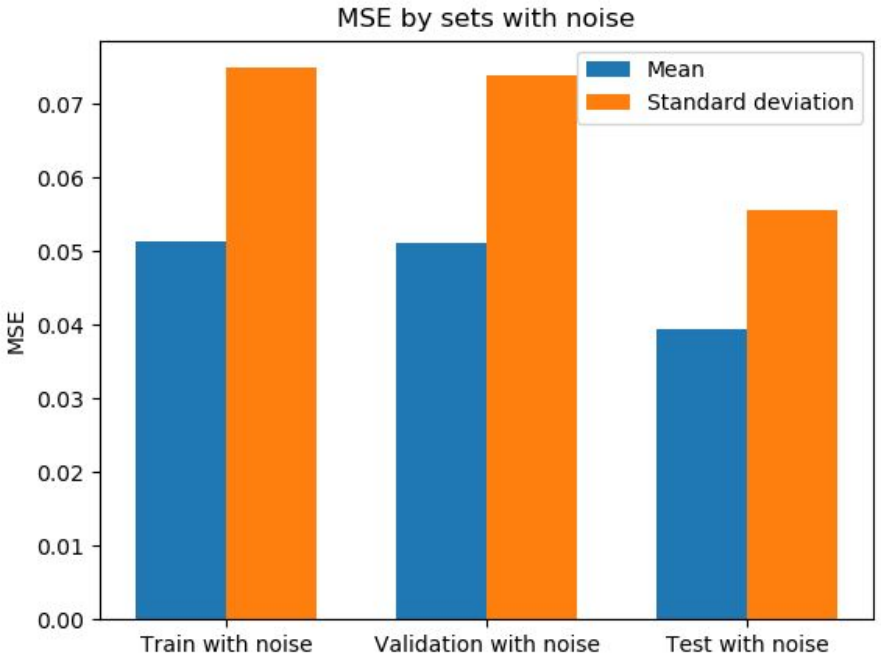




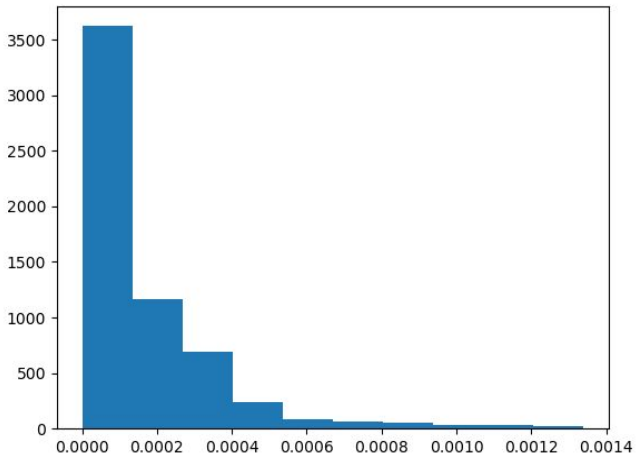
5.4. Wyniki predykcji modeli dla wszystkich zbiorów

Na rysunkach poniżej widoczne są wyniki predykcji modeli dla danych ze wszystkich zbiorów. Warto zauważyć, że znowu nie udało się ująć tego na jednym wykresie, ze względu na dużą różnicę w rzędzie wielkości. Błąd średniokwadratowy dla danych bez szumu wynosi około 1.7e-4, natomiast dla danych zaszumionych jest to około 5e-2, zatem w przybliżeniu 500 razy wyższy. Również odchylenie standardowe dla danych zaszumionych jest odpowiednio wyższe, co oznacza że sieć jest mniej stabilna. Ewidentnie sieć ma problem z przystosowaniem do danych zaszumionych i o wiele łatwiej jest jej się nauczyć aproksymacji rzeczywistej funkcji. Ciekawą obserwacją jest sporo niższa wartość błędu na zbiorze testowym z szumem. Ciężko powiedzieć z czego może to wynikać, prawdopodobnie jest to przypadek, że akurat do zbioru testowego trafiły punkty trochę mniej zaszumione i łatwiej było sieci przewidzieć ich wartości.

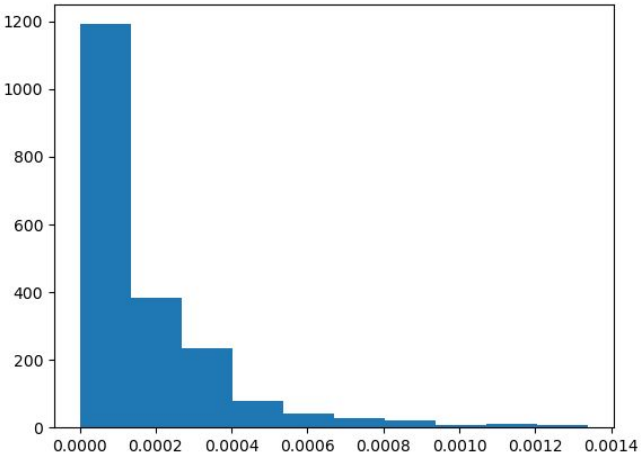




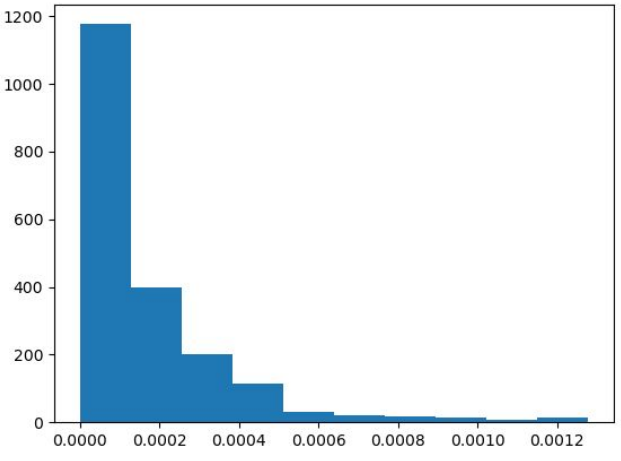
Dodatkowo prezentujemy histogramy wartości błędu średniokwadratowego dla przewidzianych przez sieć próbek ze wszystkich zbiorów. Widać wyraźnie, że sieć w większości przypadków dość dobrze aproksymuje zadaną funkcję, gdyż liczba znikomych błędów znacznie przewyższa te poważniejsze pomyłki. Dzieje się tak w przypadku wszystkich zbiorów danych. Warto zauważyć, że błąd nigdy nie przekracza 0.002 dla danych niezaszumionych i 0.8 dla danych zaszumionych.



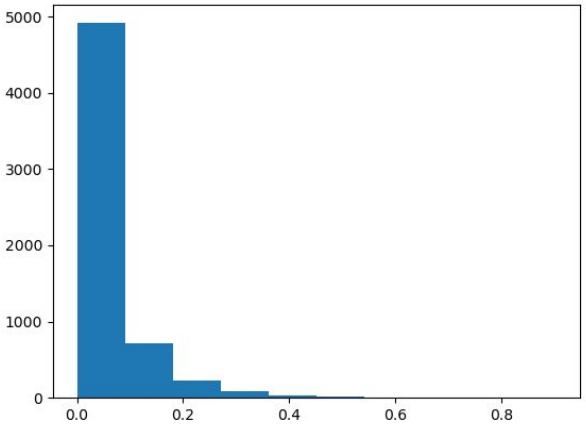
Błąd dla zbioru treningowego bez szumu.



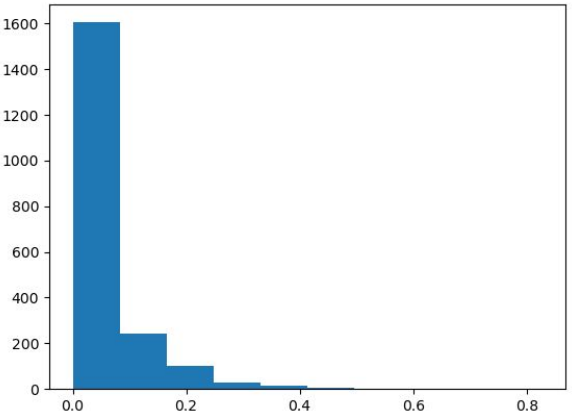
Błąd dla zbioru walidacyjnego bez szumu.



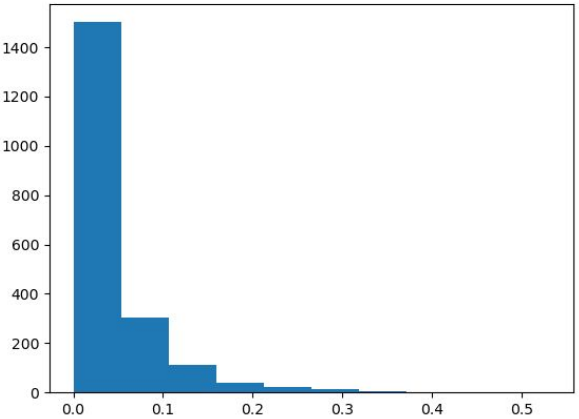
Błąd dla zbioru testowego bez szumu.



Błąd dla zbioru treningowego z szumem.



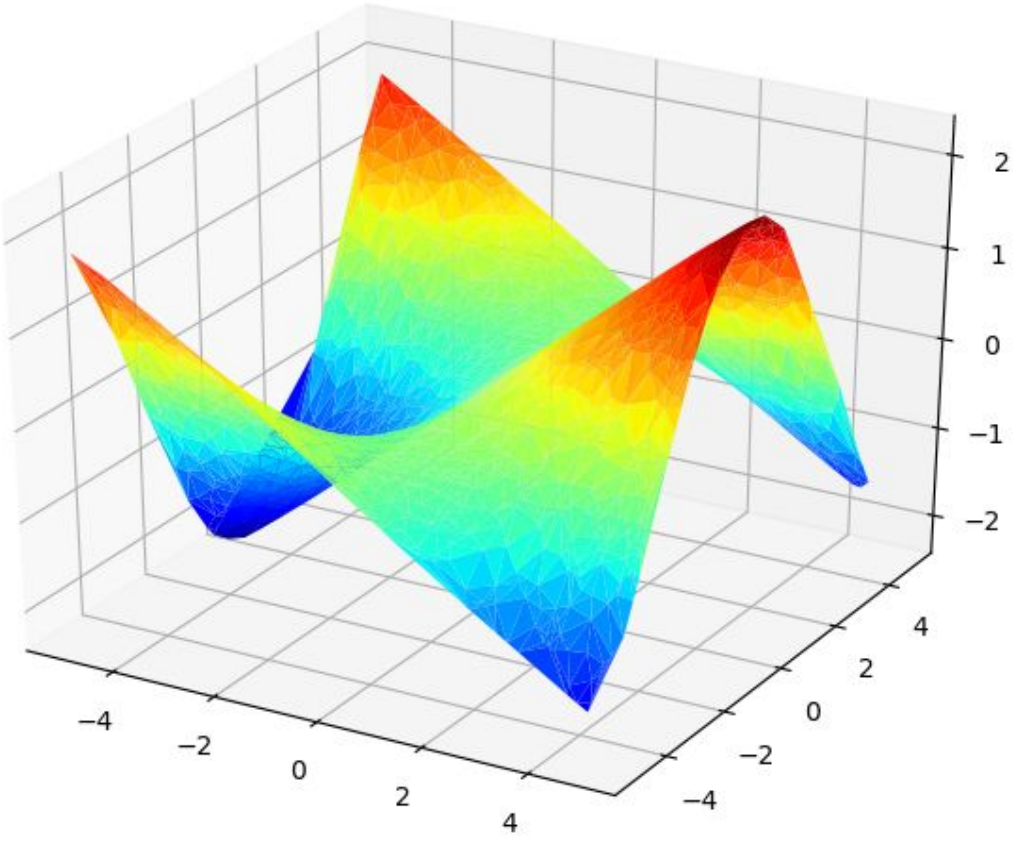
Błąd dla zbioru walidacyjnego z szumem.



Błąd dla zbioru testowego z szumem.

6. Wnioski

Cel projektu udało się osiągnąć. Zaprojektowana przez nas sieć neuronowa z dobrym przybliżeniem potrafi zaproksymować zdefiniowaną funkcję ciągłą dwuwymiarową. Za optymalną strukturę sieci dla tego zadania przyjęliśmy perceptron wielowarstwowy z jedną ukrytą warstwą neuronów, który posiada wiele neuronów w warstwie wejściowej i warstwie ukrytej. Wykres funkcji uzyskanej w wyniku symulacji na modelu ostatecznym został przedstawiony poniżej, a jego parametry są następujące: 32 neuronów w warstwie wejściowej, 8 neuronów warstwy ukrytej, wskaźnik uczenia 0.001, momentum równe 0.99 a liczba epok to 20.



O ile dla danych niezaszumionych model potrafił zaproksymować funkcję ze stosunkowo niskim błędem, o tyle dobranie parametrów modelu dla danych z szumem okazało się bardzo trudnym zadaniem. W tym przypadku jakakolwiek manipulacja parametrami modelu nie przynosiła zadowalającej poprawy błędu aproksymacji. Znaczące błędy aproksymacji na podstawie danych z szumem wynikają głównie z tego, że model mimo wszystko próbuje dopasować się do zadanej funkcji reprezentowanej przez próbki danych zaszumionych. W ten sposób wygładza dane pierwotne w wyniku czego model uzyskuje wyniki zbliżone do funkcji pierwotnej, niezaszumionej. Poniżej przedstawiono wykres funkcji uzyskanej w wyniku aproksymacji poprzez model wyznaczony na podstawie danych zaszumionych.

Parametry tego modelu są następujące: 32 neurony w warstwie wejściowej, 8 neuronów w warstwie ukrytej, współczynnik uczenia równy 0.0008, momentum równe 0.95 a liczba epok to 20.