

Krzysztof Barden 210139 210139@edu.p.lodz.pl

## Zadanie 2.: Sieć neuronowa - Perceptron wielowarstwowy

### 1. Cel

Zadanie polega na napisaniu programu implementującego sieć neuronową typu wielowarstwowy perceptron, nauczaną metodą wstecznej propagacji błędów. Sieć neuronowa ma zostać nauczona obliczania pierwiastka drugiego stopnia liczby przez użycie wejściowego wektora liczb losowych z zakresu 1-100/

### 2. Wprowadzenie

Perceptron wielowarstwowy – najpopularniejszy typ sztucznych sieci neuronowych. Sieć tego typu składa się zwykle z jednej warstwy wejściowej, kilku warstw ukrytych oraz jednej warstwy wyjściowej.

Perceptron wielowarstwowy w przeciwieństwie do perceptronu jednowarstwowego może być wykorzystywany do klasyfikowania zbiorów, które nie są liniowo separowalne. Ogólny wzór opisujący perceptrony:

$$f_w : \mathbb{R}^n \rightarrow \mathbb{R}^m \quad (1)$$

gdzie  $n$  to wejścia,  $w$  to wagi,  $m$  to wyjścia

W tym zadaniu perceptron wielowarstwowy jest uczony metodą wstecznej propagacji.

### 3. Opis implementacji

Do wykonania zadania został użyty język Python.

Sieć neuronowa (MLP) przyjmuje jako parametry ilość wejść (w tym zadaniu zawsze 1), ilość neuronów w warstwie ukrytej, ilość wyjść (w tym zadaniu zawsze 1), współczynnik nauki, współczynnik momentum, wybór czy używać biasu, ilość epok oraz wartość próbkowania błędów.

Wartości wag są inicjalizowane w przedziale  $<-0.5;0.5>$ .

Funkcją aktywacyjną jest funkcja sigmoidalna.

Sekwencja czynności, która zostaje wykonana przy nauce MLP: wzorzec treningowy podawany jest na wejścia sieci, następnie odbywa się jego propagacja wprzód, dalej na podstawie wartości odpowiedzi wygenerowanej przez sieć oraz wartości pożądanego wzorca odpowiedzi następuje wyznaczenie błędów, po czym propagowane są one wstecz, na koniec zaś ma miejsce wprowadzenie poprawek na wagi.

Sekwencja czynności przy testowaniu MLP: wzorzec treningowy podawany jest na wejścia sieci, następnie odbywa się jego propagacja wprzód, a na koniec na podstawie wartości odpowiedzi wygenerowanej przez sieć oraz wartości pożądanego wzorca odpowiedzi następuje wyznaczenie błędów.

## 4. Materiały i metody

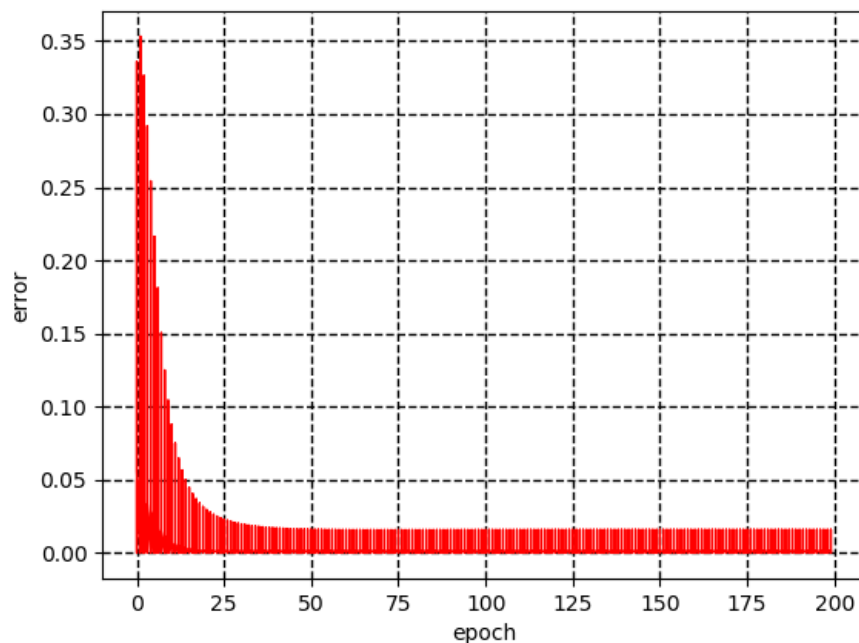
Na wejściu podawany losowo przetasowany znormalizowany wektor liczb od 1 do 100. Oczekiwane wyjście to wartości pierwiastka kwadratowego liczb z wejściowego wektora.

Do ostatniego eksperymentu po nauczaniu sieci zadano wektor liczb od 1 do 150 w celu określenia możliwości sieci do ekstrapolacji.

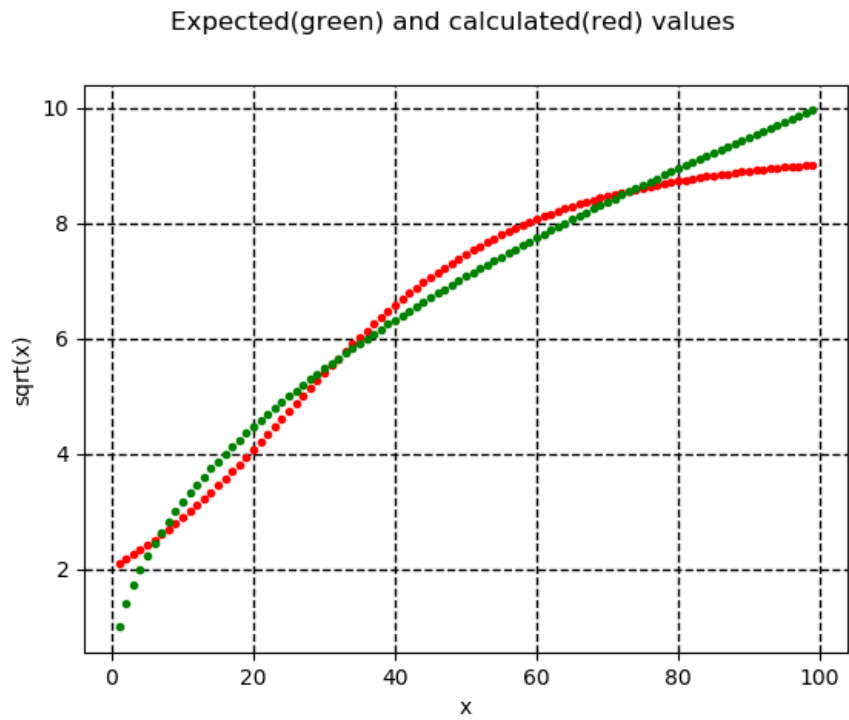
## 5. Wyniki

### Podpunkt 1.1

Error plot hidden nodes= 5 | learning rate= 0.2 | momentum= 0.9  
epochs= 200 | bias: False



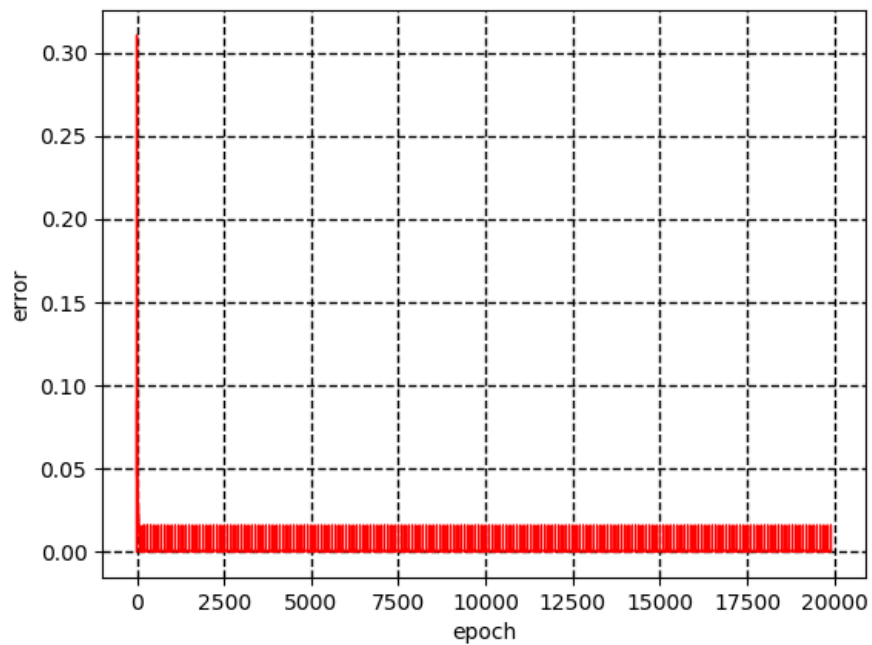
### Podpunkt 1.2



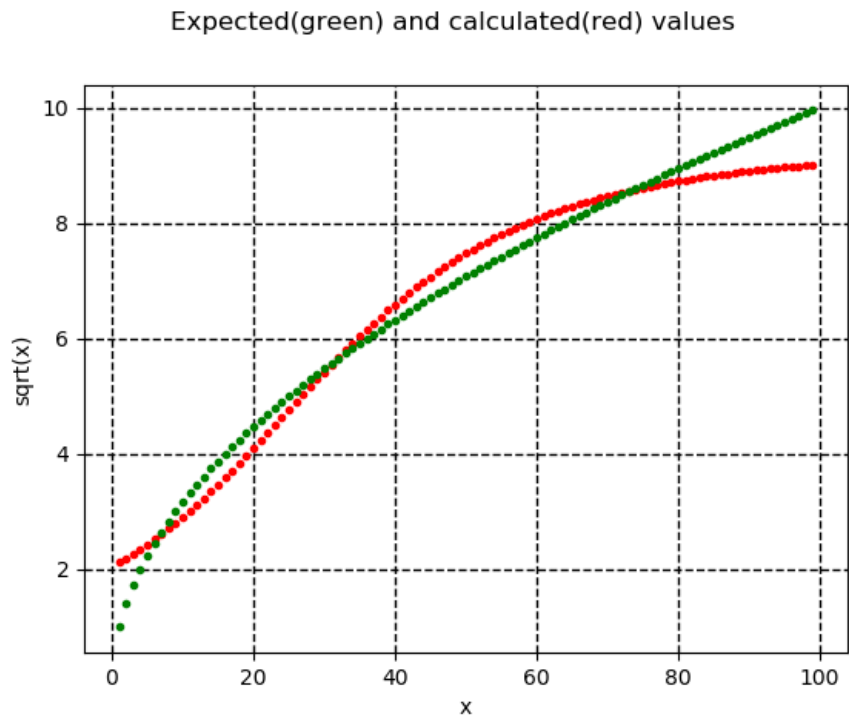
## Eksperyment 2

### Podpunkt 2.1

Error plot hidden nodes= 5 | learning rate= 0.2 | momentum= 0.9  
epochs= 20000 | bias: False



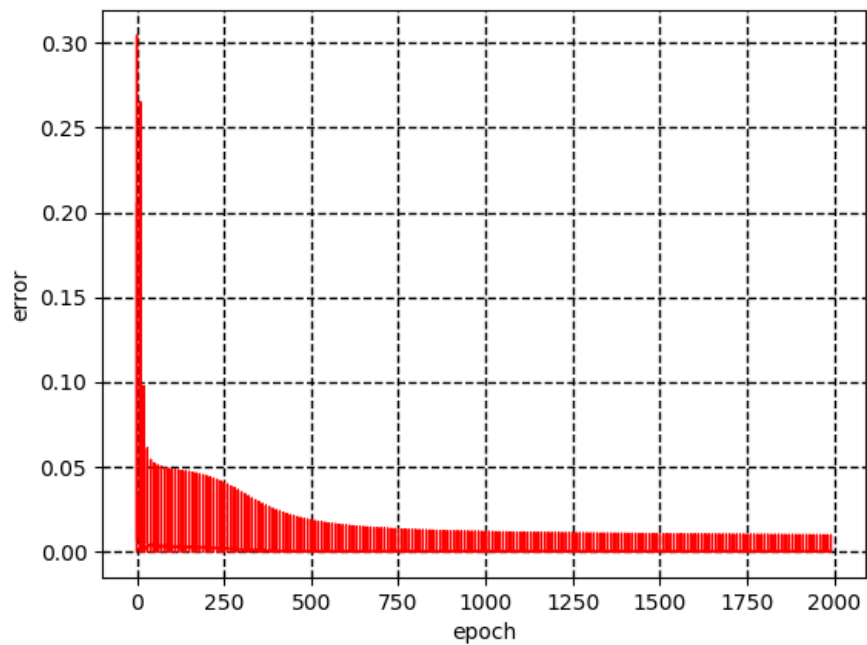
### Podpunkt 2.2



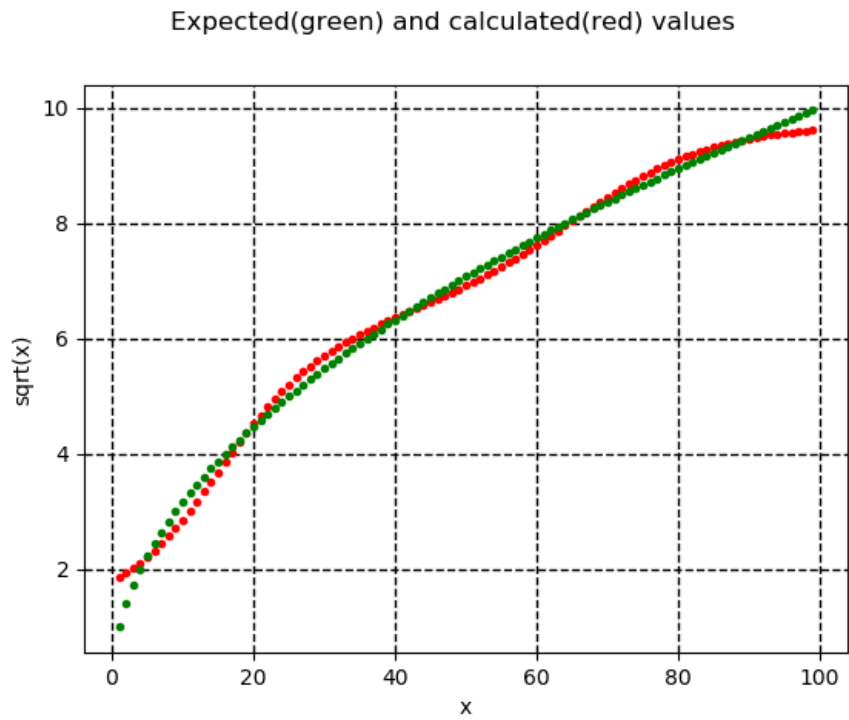
### Eksperyment 3

#### Podpunkt 3.1

Error plot hidden nodes= 1 | learning rate= 0.2 | momentum= 0.9  
epochs= 2000 | bias: True

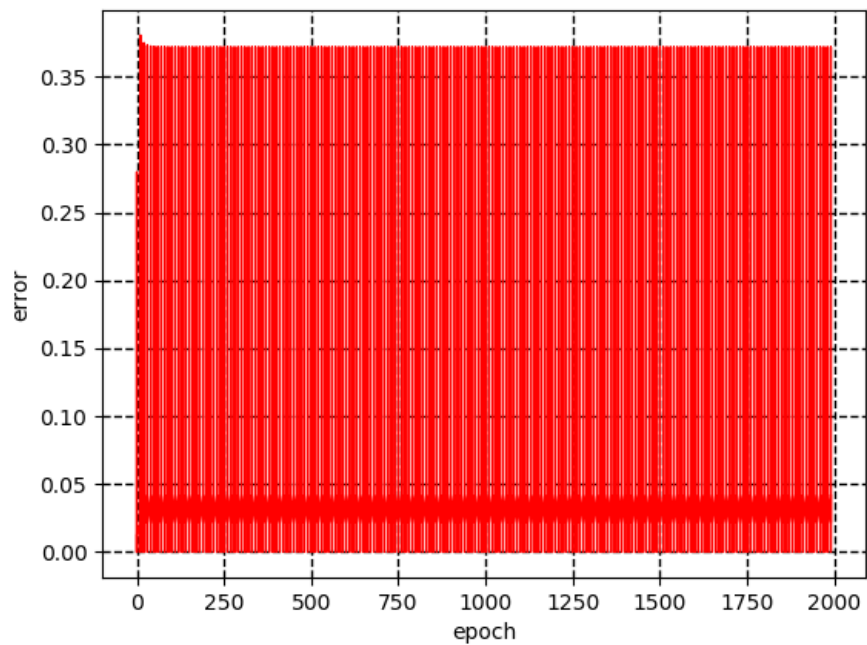


#### Podpunkt 3.2

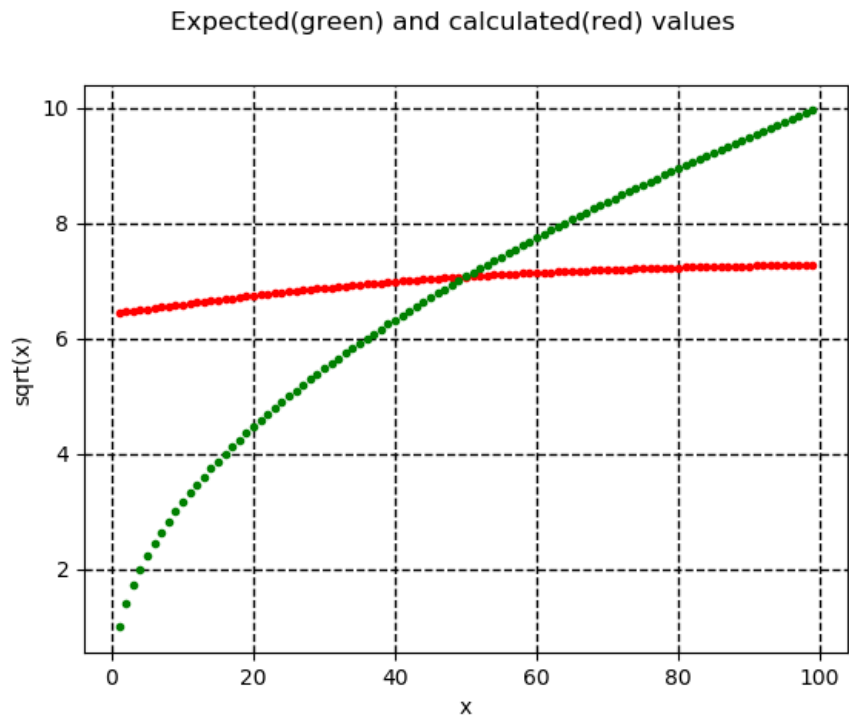


Eksperyment 4  
Podpunkt 4.1

Error plot hidden nodes= 1 | learning rate= 0.2 | momentum= 0.9  
epochs= 2000 | bias: False



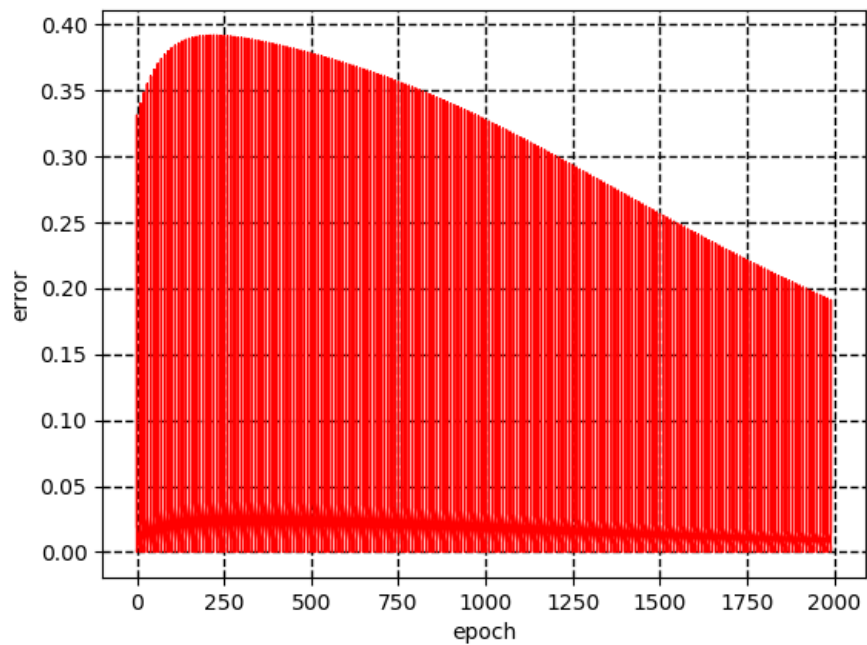
Podpunkt 4.2



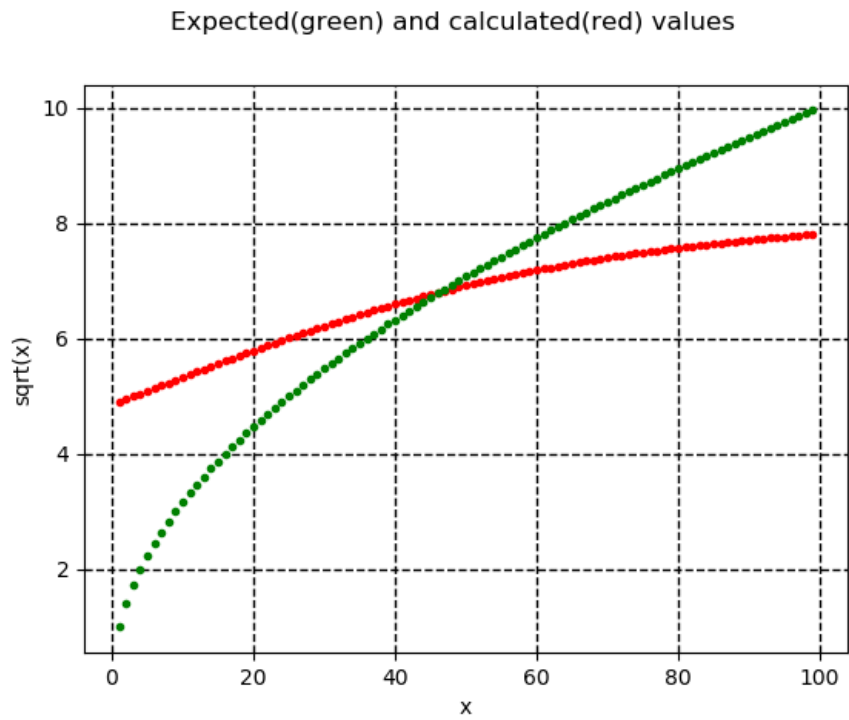
## Eksperyment 5

### Podpunkt 5.1

Error plot hidden nodes= 5 | learning rate= 0.001 | momentum= 0.9  
epochs= 2000 | bias: False

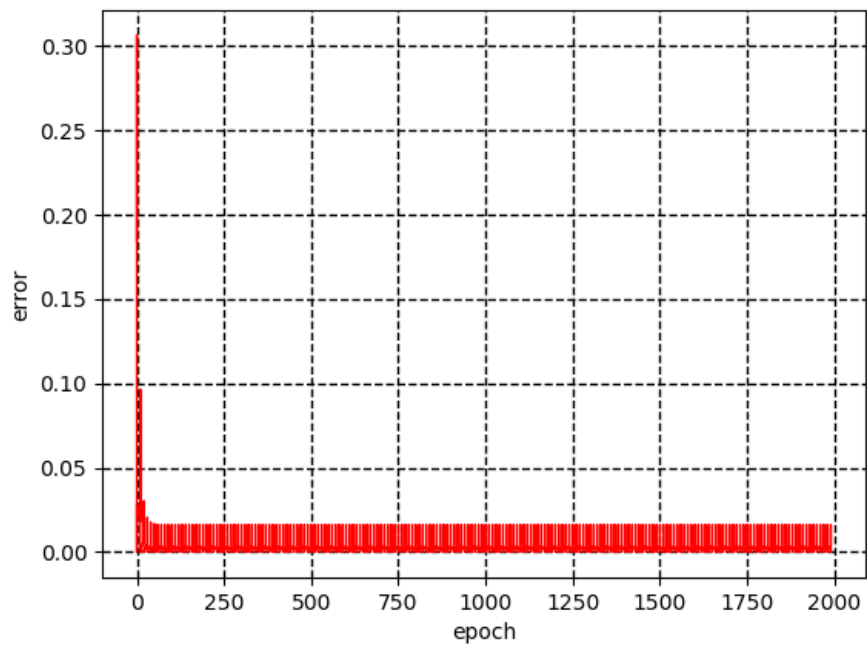


### Podpunkt 5.2

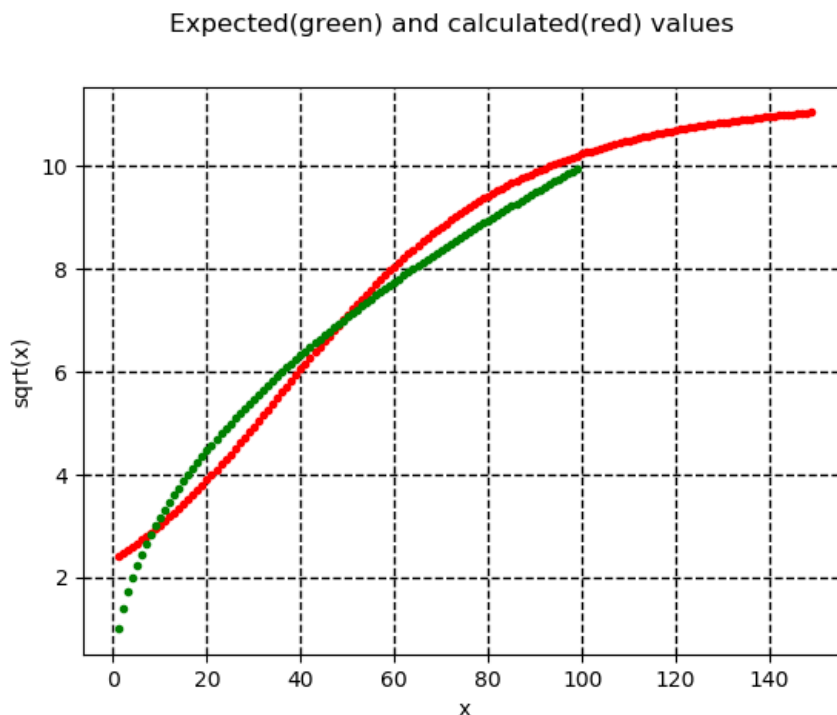


Eksperyment 6  
Podpunkt 6.1

Error plot hidden nodes= 5 | learning rate= 0.2 | momentum= 0.9  
epochs= 2000 | bias: False



Podpunkt 6.2



## 6. Dyskusja

Zwiększając ilość epok nie zawsze się zwiększa dokładność, może wystąpić przetrenowanie lub zmiany będą zbyt małe do zauważenia.

Dobranie learning rate ma duże znaczenie. Zbyt duże wartości mogą doprowadzać do powstawania większych błędów, ale zbyt małe spowalniają proces nauki.

Zbyt mała ilość neuronów w warstwie ukrytej może spowodować brak nauki sieci.

Przy małej ilości neuronów w warstwie ukrytej do poprawnej nauki sieci potrzebny jest bias.

Ekstrapolacja w przypadku tej sieci neuronowej wydaje się być dokładna.

## 7. Wnioski

- W nauce MLP czynnik losowy może mieć duże znaczenie
- W nauce MLP kluczowe jest dobranie odpowiednich wartości learning rate i momentum - za duże wartości mogą generować błędy, ale za małe spowalniać proces nauki
- Bias ma wpływ na sieć przy małej ilości neuronów w warstwie ukrytej
- Należy uważać na zjawisko przetrenowania
- Duża ilość epok może czasami nie mieć sensu gdyż zmiana błędu po czasie może być znikoma



## Literatura

- [1] T. Oetiker, H. Partl, I. Hyna, E. Schlegl. *Nie za krótkie wprowadzenie do systemu L<sup>A</sup>T<sub>E</sub>X2e*, 2007, dostępny online.
- [2] [https://pl.wikipedia.org/wiki/Perceptron\\_wielowarstwowy](https://pl.wikipedia.org/wiki/Perceptron_wielowarstwowy)