

POLITECHNIKA ŚLĄSKA W GLIWICACH

BIOLOGICALLY INSPIRED ARTIFICIAL INTELLIGENCE

---

## Przewidywanie kursu walutowego

---

AUTORZY:

Forczmański Mateusz

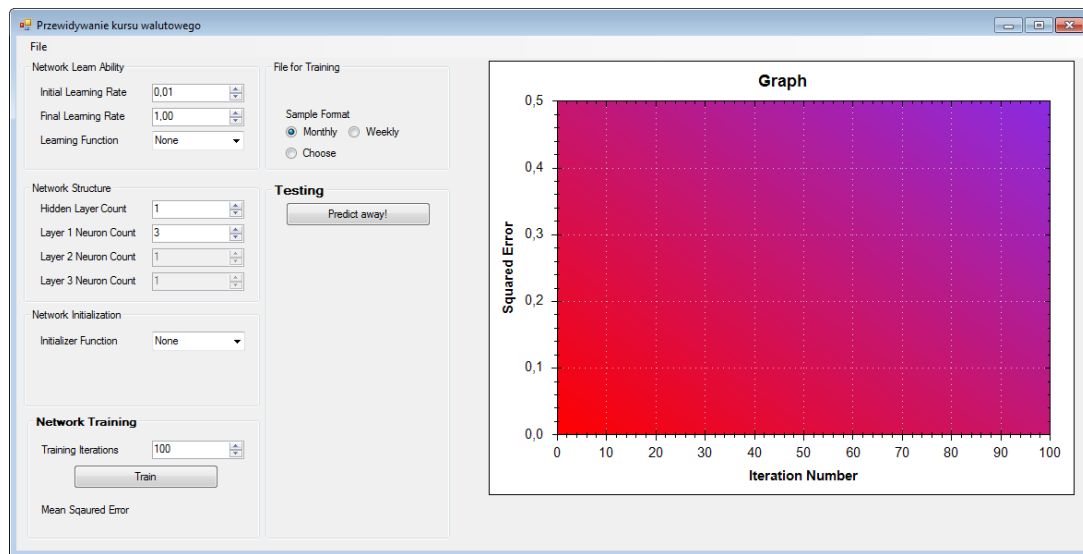
Szukała Patryk

Informatyka, semestr VI  
Rok akademicki 2014/2015  
Grupa GKiO3

21 czerwca 2015

# Specyfikacja zewnętrzna

## 1 Interfejs graficzny



Rysunek 1: Interfejs po uruchomieniu programu

Przed przystąpieniem do trenowania sieci można ją skonfigurować:

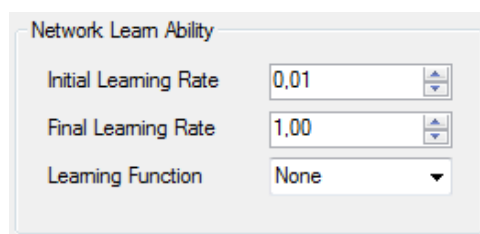
### 1.1 Struktura sieci

This image is a close-up of the 'Network Structure' panel. It contains four spinners for configuring the neural network:   
- **Hidden Layer Count:** Set to 1.   
- **Layer 1 Neuron Count:** Set to 3.   
- **Layer 2 Neuron Count:** Set to 1.   
- **Layer 3 Neuron Count:** Set to 1.   
Each spinner has up and down arrow buttons for adjustment.

Rysunek 2: Panel do konfiguracji struktury sieci z domyślnymi wartościami

W tym panelu definiuje się budowę wewnętrzną sieci neuronowej - liczbę warstw pośrednich oraz liczbę neuronów w każdej z warstw. Maksymalnie można utworzyć 3 warstwy z minimum 1 neuronem. W zależności od wartości w polu *Hidden Layer Count*, odpowiednie pola *Layer Neuron Count* są dostępne lub nie.

## 1.2 Zdolność nauki

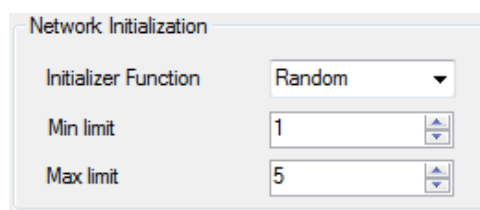


Rysunek 3: Panel do konfiguracji zdolności nauki z domyślnymi wartościami

Ten panel znajduje się w górnym lewym rogu interfejsu. Umożliwia konfigurację trzech parametrów:

- *Initial Learning Rate* - wartość startowa zdolności nauki, z nią uruchamiany jest trening sieci.
- *Final Learning Rate* - wartość zdolności nauki, którą sieć będzie miała, gdy trening się skończy.
- *Learning Function* - specjalizowana funkcja, jakiej sieć backpropagacji będzie wykorzystywać do nauki. Są cztery możliwości:
  - *None* - standardowa funkcja
  - *Exponential* - funkcja eksponencjalna
  - *Hyperbolic* - funkcja hiperboliczna
  - *Linear* - funkcja liniowa

## 1.3 Inicjalizacja funkcji

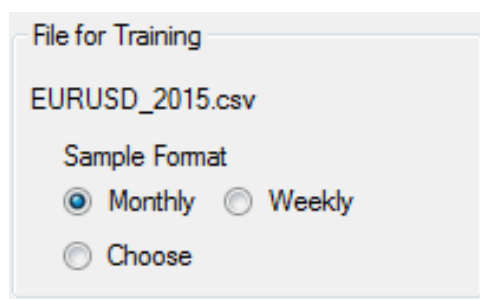


Rysunek 4: Panel do konfiguracji inicjalizacji sieci z przykładowymi ustawieniami

W tej części (lewa strona interfejsu) można ustawić inicjalizację sieci, czyli początkową wartość wag neuronów we wszystkich warstwach. Istnieje sześć możliwości:

- *None* - wartości domyślne.
- *Constant* - wskazana wartość stała.
- *NguyenWidrow* - sparametryzowana funkcja *NGuyen Widrow*.
- *NormalizedRandom* - znormalizowana wartość losowa.
- *Random* - wartość losowa ze wskazanego zakresu.
- *Zero* - funkcja typu Zero (wszystkie wartości są zbliżone do zera).

## 1.4 Analiza plików

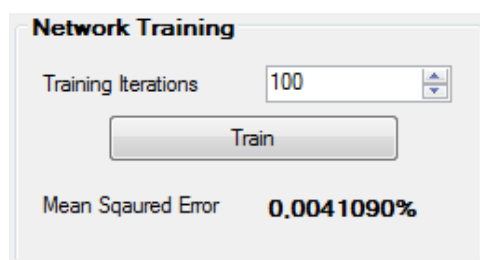


Rysunek 5: Panel z wczytanym plikiem i sposobem podziału

Ten panel znajduje się w górnej środkowej części interfejsu. Umożliwia pogląd załadowanego pliku do treningu sieci (tutaj *EURUSD\_2015.csv*) oraz sposób w jaki dane będą grupowane:

- *Monthly* - co miesiąc.
- *Weekly* - co tydzień.
- *Choose* - wg wskazanej przez użytkownika liczby dni.

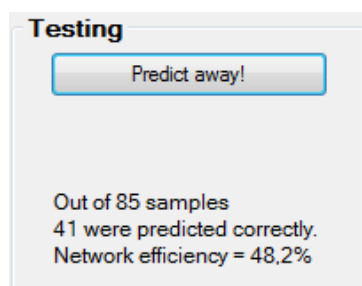
## 1.5 Trening sieci



Rysunek 6: Przykładowy wynik trenowania sieci po 100 iteracjach

Ten panel znajduje się w dolnym prawym rogu. Wymaga od użytkownika wprowadzania liczby iteracji treningowych, jakie wykona sieć (domyślnie jest to wartość 100). Po wciśnięciu przycisku *Train* sieć rozpocznie swój trening. Gdy zostanie on zakończony, na dole pojawia się *Mean Squared Error* (średni kwadratowy błąd) sieci w procentach. Oznacza on jakie wg obliczeń jest prawdopodobieństwo błędu sieci, czyli złego przewidzenia linii trendu kursu waluty. Program poświęca 70% próbek (zaokrąglonych w górę) do testów.

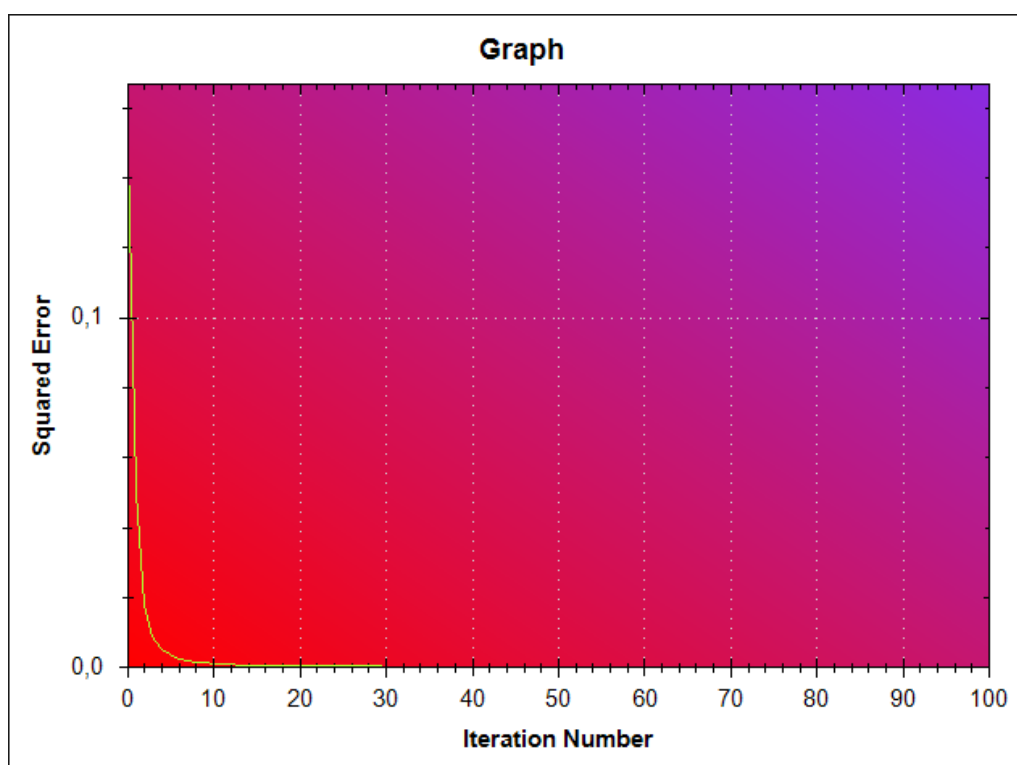
## 1.6 Trening sieci



Rysunek 7: Przykładowy wynik testowania sieci

Na środku interfejsu znajduje się przycisk *Predict away!*. Gdy sieć została wytrenowana i skonfigurowana, jego wciśnięcie powoduje uruchomienie testowania sieci i zmierzenie jej efektywności. Po wykonaniu testów informacje zostaną wyświetlone w tym samym panelu. Program poświęca 30% próbek w celu zbadania jakości działania sieci.

## 1.7 Graf błędu



Rysunek 8: Przykładowy przebieg nauki sieci

W czasie treningu sieci jest generowany graf zależności jakości sieci od liczby iteracji. Przedstawia on jak prawdopodobieństwo błędu zmieniało się wraz z kolejnymi iteracjami. Można powiedzieć, że obrazuje on, w jaki sposób sieć się uczy.

Maksymalną wartością na osi poziomej jest liczba iteracji, z kolei na osi pionowej jest to maksymalny średni błąd kwadratowy (wyliczany dynamicznie w trakcie działania programu).