

**AGH**

**MSI**

**Sieci Neuronowe**

Krzysztof Romanowski  
grudzień 2012

# Klasyfikacja

Do klasyfikacji użyłem zestawienia jakości wina wraz z jego podstawowymi parametrami (zawartość alkoholu, PH itp). Mój klasyfikator miał za zadanie sprawdzić czy dane wino ma jakość większą niż 5.

Przed użyciem dane (zarówno wejściowe) zostały znormalizowane wg wzoru

$$v_{norm} = \frac{v - v_{min}}{v_{max} - v_{min}}$$
 dającego zawsze wartości w przedziale  $[0,1]$ . Dane wyjściowe to wartości binarne (0.0 oraz 1.0) w zależności od klasy wina.

Opis sieci:

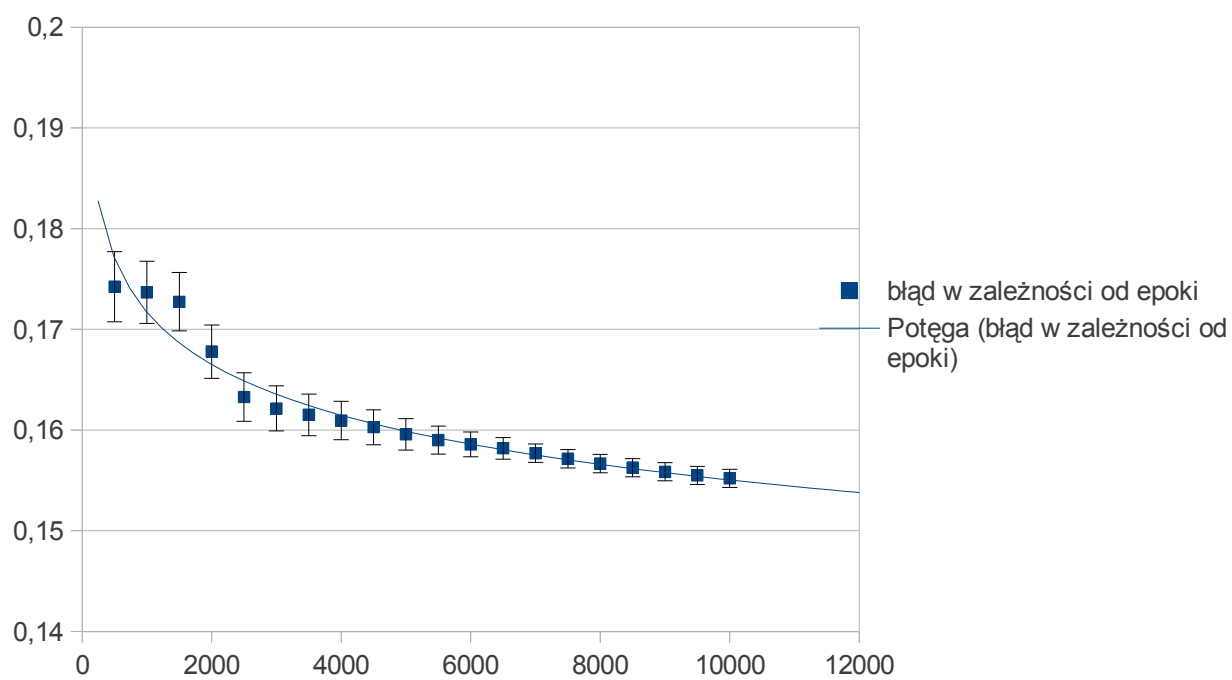
```
net.start <- newff(n.neurons=c(11,6,1),  
                  learning.rate.global=1e-3,  
                  momentum.global=0.8,  
                  error.criterium="LMLS",  
                  Stao=NA, hidden.layer="tansig",  
                  output.layer="tansig",  
                  method="ADAPTgdwm")
```

Procedura trenująca:

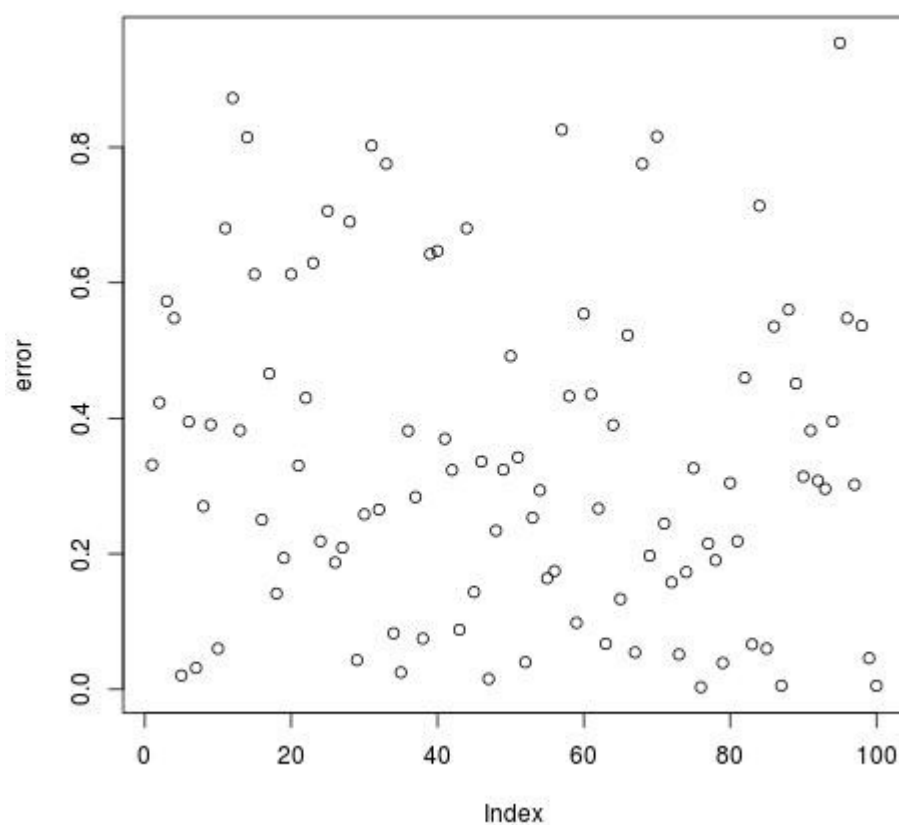
```
result <- train(net.start, data,  
               res, error.criterium="LMS",  
               report=TRUE, show.step=500, n.shows=20)
```

Wyniki:

Skuteczność klasyfikacji (średnia pozytywnie zakwalifikowanych) wahała się od (15% do 28%)



*Ilustracja 1: Błąd w zależności od epoki*



*Ilustracja 2: Przykładowy rozkład błędów*

# Predykcja realnego ciągu czasowego

Do predykcji posłużyły dane o kursie obligacji żadu niemieckiego (FGBL) w ujęciu minutowym (średni kurs obligacji w danej minucie handlu).

Przed użyciem dane (zarówno wejściowe jak i wyjściowe) zostały znormalizowane wg wzoru

$$v_{norm} = \frac{v - v_{min}}{v_{max} - v_{min}} \quad \text{dającego zawsze wartości w przedziale } [0,1] \text{ .}$$

Wyniki będą przedstawiały dla różnej liczby parametrów wejściowych (8, 10, 12 oraz 16).

Dane wejściowe : X poprzednich kursów

Dane wyjściowe: aktualny kurs

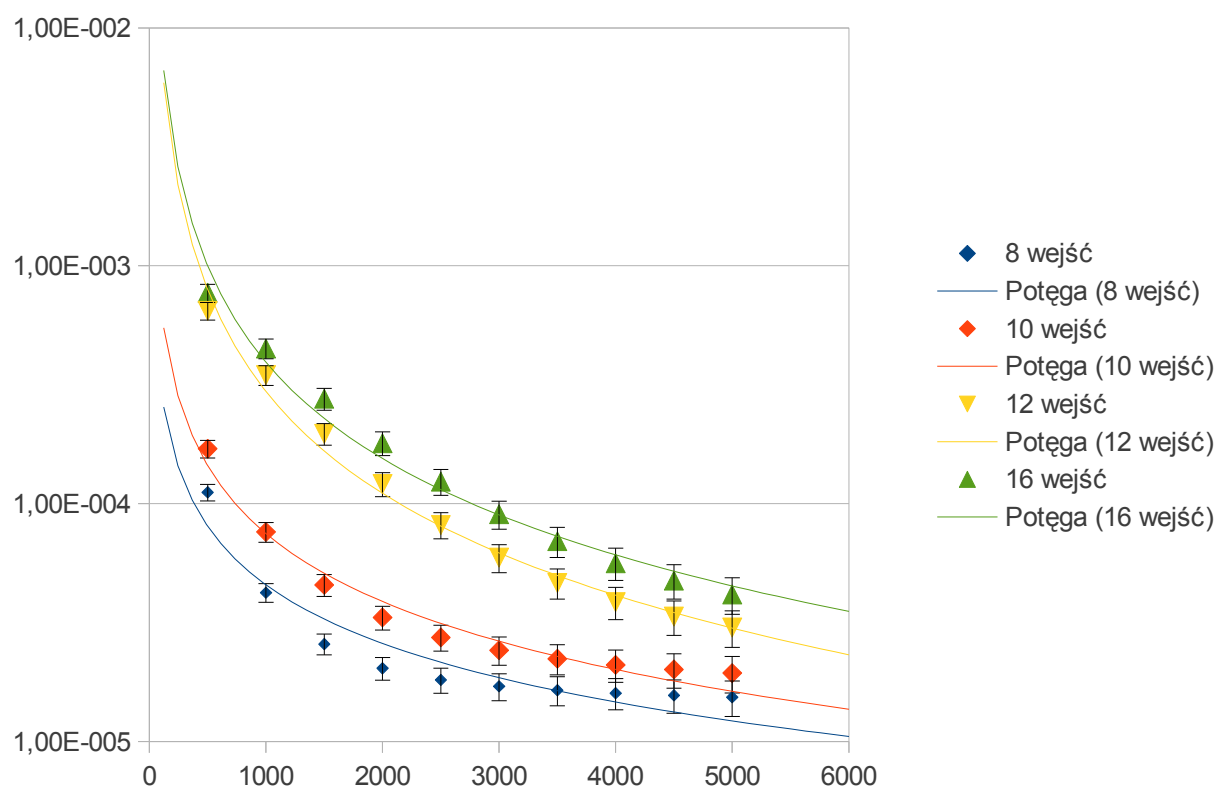
Opis sieci:

```
net.start <- newff(n.neurons=c(X,X/2+1,1),
                  learning.rate.global=1e-3,
                  momentum.global=0.8,
                  error.criterium="LMLS",
                  Stao=NA, hidden.layer="tansig",
                  output.layer="tansig",
                  method="ADAPTgdwm")
```

Procedura trenująca:

```
result <- train(net.start, data, res,
               error.criterium="LMS", report=TRUE,
               show.step=500, n.shows=10)
```

Wyniki:

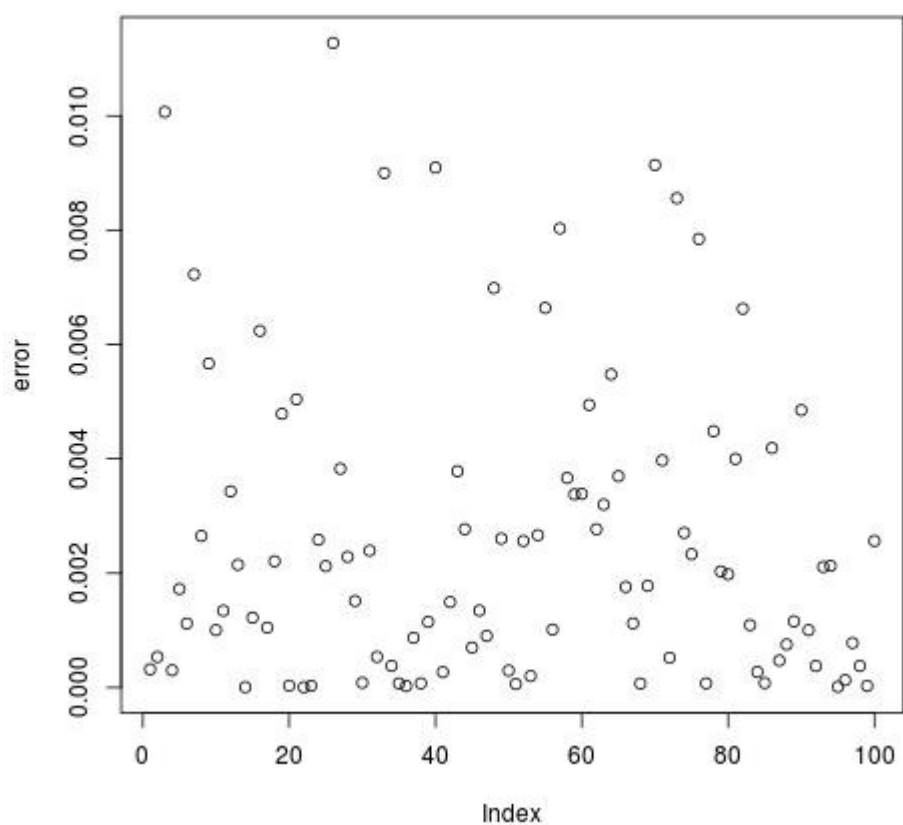


*Ilustracja 3: Wykres średniego błędu w zależności numeru epoki dla różnej liczby wejść*

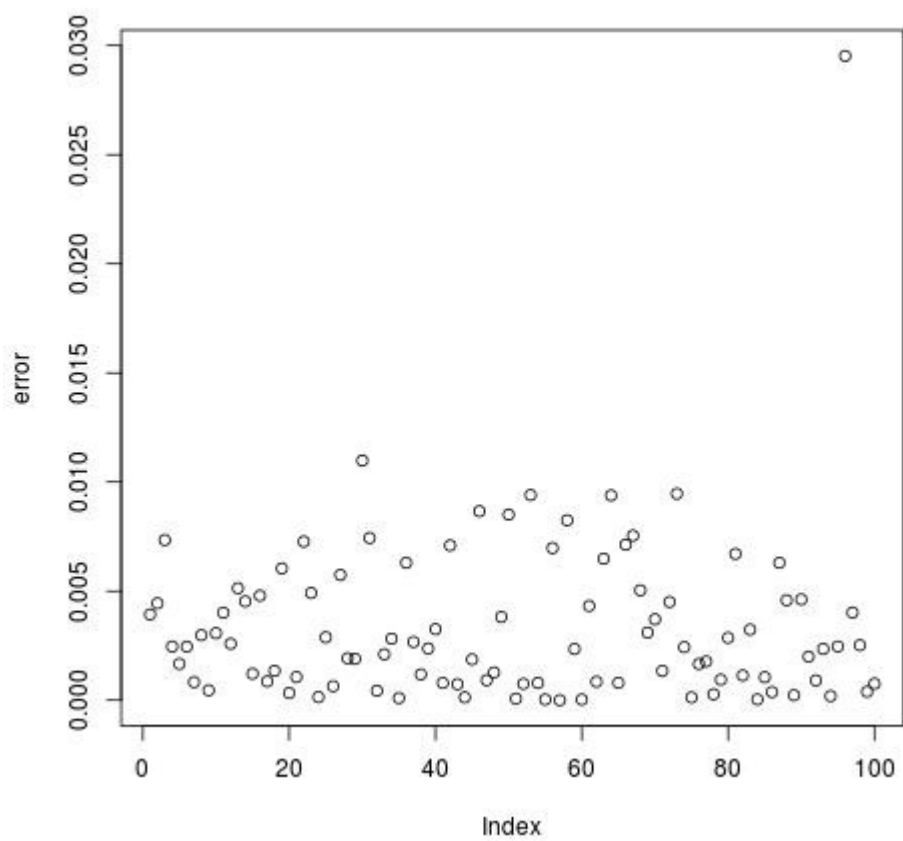
Dane testowe: Były to losowo wybierane momenty czasowe, wykresy obrazują wglądny błąd dla poszczególnych danych.





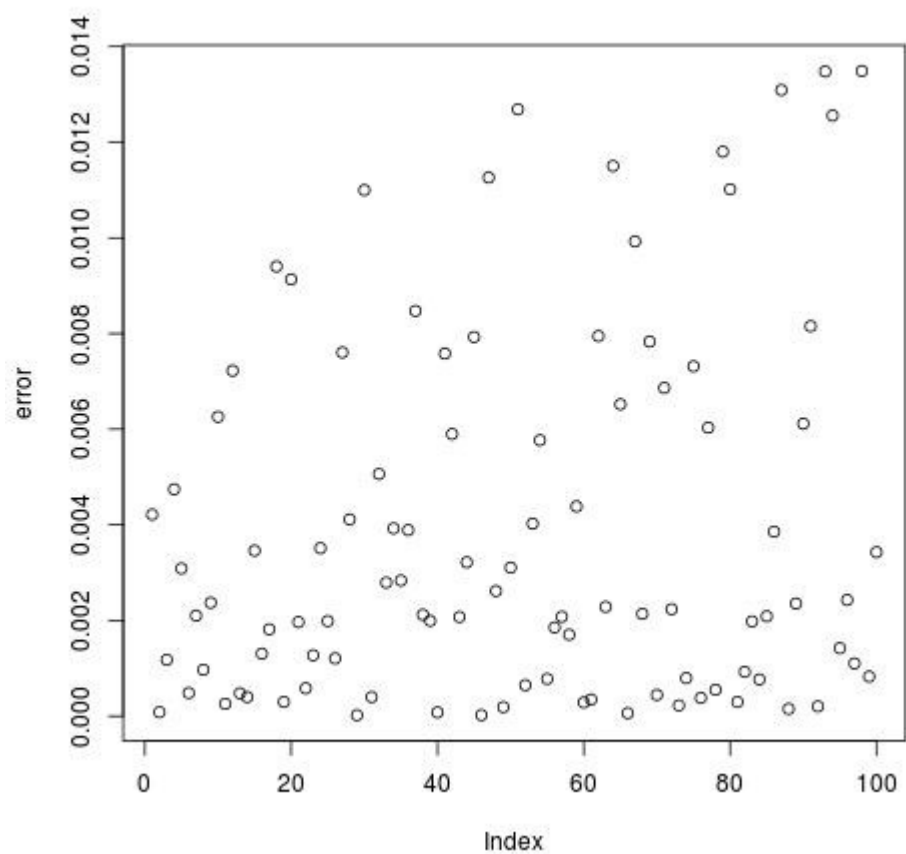


*Ilustracja 4: Dla 8 wejść*



*Ilustracja 5: Dla 10 wejść*





*Ilustracja 6: Dla 12 wejść*

*Ilustracja 7: Dla 16 wejść*

