AITECH®
Artificial Intelligence Laboratory

ZINTEGROWANY PAKIET SZTUCZNEJ INTELIGENCJI SPHINX® 4.0

KRZYSZTOF MICHALIK

NEURONIX 4.0

dla Windows 9x/NT/2000

SYMULATOR SZTUCZNYCH SIECI NEURONOWYCH

PODRĘCZNIK UŻYTKOWNIKA

KATOWICE 2003

AITECH®
Artificial Intelligence Laboratory

ul. Kossutha 7, 40-844 KATOWICE
tel./fax: tel.: (0-32) 254-41-01 w. 374
tel. kom. 0 502-99-27-28
e-mail: aitech@aitech.com.pl
WWW: http://www.aitech.com.pl

Copyright ©1990-2003 AITECH & Krzysztof Michalik

AITECH, **Sphinx**, **CAKE** oraz **Neuronix**
są prawnie zastrzeżonymi znakami towarowymi firmy
AITECH, ARTIFICIAL INTELLIGENCE LABORATORY

SPIS TREŚCI

Rozdział 1. **CZYM JEST SYSTEM NEURONIX ? 1-1**
UŻYTKOWNICY SYSTEMU NEURONIX..... 1-3
CO UMOŻLIWIA SYSTEM NEURONIX UŻYTKOWNIKOWI ? 1-4
Rozdział 2. **ZANIM ROZPOCZNIESZ PRACĘ Z SYSTEMEM NEURONIX..... 2-1**
SIEĆ NEURONOWA A APLIKACJE HYBRYDOWE 2-3
WSPÓŁPRACA Z INNYMI APLIKACJAMI PAKIETU SPHINX 2-3
Przykład wywołania sieci neuronowej z poziomu bazy wiedzy 2-4
URUCHOMIENIE SIECI Z POZIOMU PROGRAMU WYKONYWALNEGO2-5
RODZAJE PLIKÓW UŻYWANE PRZEZ SYSTEM NEURONIX 2-7
PROJEKT SIECI NEURONOWEJ..... 2-7
Rozdział 3. **MENU SYSTEMU NEURONIX 3-1**
GRUPA PLIK 3-3
GRUPA POMOC..... 3-4
EDYCJA 3-5
Wytnij, Kopiuj, Wstaw 3-5
Idź do..... 3-5
Sortuj 3-5
Znajdź..... 3-6
Zastąp 3-6
EDYTOR 3-7
WSTAW 3-7
SIEĆ..... 3-7
NARZĘDZIA 3-8
Wielozadaniowy system uczenia 3-8
Kreator pliku uczącego i testowego 3-8
Parametry sieci 3-9
Translator..... 3-9
ZASOBY PROJEKTU 3-9
OPCJE SIECI NEURONOWEJ..... 3-10
Parametry strukturalne sieci..... 3-10
Parametry procesu uczenia 3-11
Parametry procesu testowania..... 3-12
Warunki zakończenia procesu uczenia 3-13
UCZENIE SIECI - INFORMACJE TEORETYCZNE 3-14
Wskaźniki informujące o bieżącym stanie sieci 3-14
Błąd RMS 3-14
TOLERANCJA..... 3-15
ILOŚĆ WZORCÓW POZA TOLERANCJĄ 3-15
MONITORING PROCESU UCZENIA I TESTOWANIA..... 3-16
UCZENIE SIECI 3-18
Uruchamianie sieci z poziomu systemu Neuronix 3-19
Rozdział 4. **CZYM WŁAŚCIWIE JEST SZTUCZNA SIEĆ NEURONOWA ? 4-1**
HISTORIA 4-3
BUDOWA SZTUCZNEGO NEURONU 4-3
JAK TO ROZWIĄZAŁA NATURA ? 4-4
WYBRANE RODZAJE SIECI NEURONOWYCH..... 4-5
Sieci jednowarstwowe jednokierunkowe..... 4-5
Sieć wielowarstwowa jednokierunkowa 4-5

W CZYM SIECI NEURONOWE SĄ MISTRZAMI ? 4-6
Rozdział 5. **DANE W PROCESIE UCZENIA I URUCHAMIANIA SIECI 5-1**
PLIK UCZĄCY I TESTOWY 5-3
Rozdział 6. **ARKUSZE KALKULACYJNE....6-1**
CO ZAWIERA SKOROSZYT ? 6-3
INFORMACJE OGÓLNE....6-3
Klawisze....6-3
Mysz....6-3
Zawartość komórek....6-3
FUNKCJE....6-4
Wspólne funkcje i własności arkuszy programu Neuronix 6-5
MENU PODRĘCZNE ARKUSZY 6-5
ARKUSZ DANYCH 6-11
ARKUSZ PREZENTACJI 6-12
ARKUSZ RAPORTU 6-13
FUNKCJE NEURONOWE ARKUSZY 6-15
Rozdział 7. **WYKRESY....7-1**
JAK UTWORZYĆ NOWY WYKRES ? 7-3
OTWIERANIE WYKRESU JUŻ ISTNIEJĄCEGO 7-6
PARAMETRY WYKRESU DOSTĘPNE DLA UŻYTKOWNIKA....7-6
Połączenie z arkuszem 7-6
Parametry wykresu....7-6
Kreator wykresu....7-7
SIATKA DANYCH 7-7
Rozdział 8. **OBSŁUGA WYDRUKÓW W SYSTEMIE NEURONIX 8-1**
DRUKOWANIE ZAWARTOŚCI ARKUSZY 8-3
DRUKOWANIE WYKRESÓW 8-4
DRUKOWANIE TOPOLOGII SIECI NEURONOWEJ 8-5
Rozdział 9. **CO POTRAFI SIEĆ NEURONOWA ? 9-1**
KLASYFIKACJA I ROZPOZNAWANIE WZORCÓW....9-3
Funkcja AND....9-3
Funkcja XOR 9-4
Kodowanie....9-7
Aproksymacja 9-8
Odtwarzanie "zaszumionych" wzorców....9-8
Rozdział 10. **W JAKI SPOSÓB UTWORZYĆ MODEL ZJAWISKA ? 10-1**
KLASY MODELI....10-3
PODZIAŁ MODELI....10-4
JAK ZBUDOWAĆ MODEL ?....10-4
MODELE UOGÓLNIAJĄCE I NIE UOGÓLNIAJĄCE 10-4
Model uogólniający....10-5
Model nie uogólniający....10-6
Wnioski....10-7
Rozdział 11. **IDENTYFIKACJA MODELI 11-1**
CO SKŁADA SIĘ NA IDENTYFIKACJĘ MODELU ? 11-3
MODEL SZEREGU CZASOWEGO 11-3
PRZYGOTOWANIE DANYCH DLA PROGNOZY 11-4
Tendencja rozwojowa 11-4
Wahania okresowe 11-4
Wahania koniunkturalne 11-4
WAHANIA PRZYPADKOWE 11-4
IDENTYFIKACJA MODELU SZEREGU CZASOWEGO 11-5
Układ mechaniczny....11-6
Układ elektryczny 11-7
Wygenerowanie szeregu czasowego wyjścia modelu (symulacja)....11-7
Identyfikacja parametrów modelu szeregu czasowego 11-8

IDENTYFIKACJA MODELU ALTMANA 11-8
Rozdział 12. **KREATOR PROGNOZ 12-1**
PODSTAWOWE OPCJE KREATORA 12-3
FILTRACJA SZEREGU CZASOWEGO 12-4
JAK FILTRY DZIAŁAJĄ W PRAKTYCE ? 12-5
Filtr dolnoprzepustowy 12-5
Filtr górnoprzepustowy 12-5
Filtr pasmowoprzepustowy 12-5
Co widać na rozkładzie prawdopodobieństwa ? 12-6
PRZYGOTOWANIE DANYCH ZA POMOCĄ KREATORA 12-7
Panel formatowania danych 12-7
Panel wykresu szeregu czasowego 12-7
Panel histogramu szeregu czasowego 12-8
Panel filtracji szeregu czasowego 12-10
PRZYKŁAD PROGNOZY 12-10
Źródło danych 12-11
Rozdział 13. **PRZYKŁADY DZIEDZINOWE 13-1**
MEDYCYNA 13-3
RYNKI KAPITAŁOWE 13-3
OCENA RYZYKA INWESTYCJI 13-4
ANALIZA FINANSOWA 13-6
Modele w postaci szarej i czarnej skrzynki 13-7
Sposoby oceny 13-8
REDUKCJA ZAKŁÓCEŃ 13-10
DYNAMICZNY MODEL STRATEGII REKLAMOWEJ 13-11
Rozdział 14. **BŁĘDY SYGNALIZOWANE W SYSTEMIE NEURONIX 14-1**
SZABLON BŁĘDU 14-3
WYKAZ BŁĘDÓW 14-3
Rozdział 15. **AUTOMATYZACJA OLE 15-1**
AUTOMATYZACJA OLE 15-3
FUNKCJE SERWERA 15-3
PRZYKŁAD AUTOMATYZACJI OLE 15-4

1

Czym jest system Neuronix ?

W rozdziale:

- Użytkownicy systemu Neuronix
- Co umożliwia system Neuronix Użytkownikowi
- Inne możliwości systemu Neuronix

Cenne informacje

Ćwiczenia z komputerem

Koniecznie przeczytaj

System Neuronix jest narzędziem służącym do wszechstronnej analizy danych, potencjalne pole jego zastosowań jest nieograniczone. Zbudowany został na bazie rozwijanej od lat pięćdziesiątych teorii sztucznych sieci neuronowych, dzięki temu pozwala na symulowanie procesów przybliżonego rozumowania człowieka. Ideą stosowania sztucznych sieci neuronowych jest naśladownictwo złożonego i jak dotychczas nie wyjaśnionego procesu zdobywania wiedzy o otaczającym świecie przez człowieka, a następnie uogólnienie tej wiedzy i indukcję nowych zachowań nie mieszczących się w zbiorze zachowań wyuczonych. Jest to zdolność do abstrakcyjnego myślenia przyporządkowana wyłącznie człowiekowi. Sztuczna sieć neuronowa jest tylko nieudolną próbą naśladowania tych procesów, jednak obecny poziom możliwości technicznych jakimi dysponuje człowiek w dziedzinie symulowania takich procesów uległ zwiększeniu od lat pięćdziesiątych na tyle, że stało się możliwe dostarczenie komercyjnego produktu w postaci systemu Neuronix, przeznaczonego dla szerokiego kręgu obecnych i przyszłych klientów firmy AITECH.

Tak więc Drogi Użytkowniku znalazłeś się w gronie osób stosujących na co dzień najnowocześniejsze rozwiązania informatyczne z zakresu sztucznej inteligencji, będące jeszcze wczoraj na etapie badań laboratoryjnych, a dziś oferujące Tobie więcej niż mógłbyś od nich oczekiwać.

UŻYTKOWNICY SYSTEMU NEURONIX

System Neuronix może być wykorzystywany przez:

1. Menadżerów zarządzających przedsiębiorstwami w zakresie:

   - taktycznego i strategicznego planowania finansowego,
   - zarządzania finansami przedsiębiorstw,
   - podejmowania decyzji strategicznych i operacyjnych,
   - prognoz strategicznych i operacyjnych,
   - badań operacyjnych

2. Banki oraz instytucje finansowe o zbliżonym profilu działalności w zakresie:

   - oceny kondycji finansowej kredytobiorcy,
   - oceny ryzyka kredytowego dla banku.

3. Biura maklerskie w zakresie:

   - tworzenia modeli wybranych procesów socjologicznych na rynkach finansowych,
   - tworzenia modeli wyceny instrumentów rynku kapitałowego,
   - typowej prognozy krótko i długoterminowej dla instrumentów rynku kapitałowego.

4. Otwarte i zamknięte fundusze inwestycyjne i emerytalne w zakresie:

   - alokacji jednostek,
   - ocenie ryzyka alokacji,
   - wstępnego doboru instrumentów dla analizy portfelowej,
   - prognoz wskaźników finansowych,
   - tworzenia modeli

5. Działy logistyki i zaopatrzenia przedsiębiorstw w zakresie:

   - analizy stanu materiałów,
   - zarządzania produkcją w systemie Just In Time,
   - planowaniu zapotrzebowania na materiały w procesie produkcyjnym,

6. Działy produkcji i eksploatacji w zakresie:

   - podejmowania decyzji o wyłączeniu obiektu z ruchu,
   - podejmowaniu bieżących decyzji remontowych.

7. Kliniki i szpitale w zakresie:

   - efektywnej analizy danych o przebiegu procesu leczenia pacjenta,
   - tworzenia skutecznych modeli procesów fizjologicznych i patologicznych.

8. Towarzystwa ubezpieczeniowe w zakresie:

   - alokacji składek,
   - ocenie ryzyka i modelowania struktury portfela,
   - ocenie możliwości reasekuracji.

9. Wyższe uczelnie o profilach ekonomicznych, technicznych i społecznych na etapie wspomagania dydaktyki zarówno z przedmiotów dziedzinowych, jak również dotyczących zagadnień sztucznej inteligencji.

10. We wszystkich zagadnieniach, w których nie jest możliwe utworzenia prostego modelu danego procesu i jego rozwiązanie za pomocą metod klasycznych, lub gdy rozwiązanie można otrzymać w sposób prostszy przy użyciu wyłącznie systemu Neuronix.

11. W przypadkach gdy konieczne jest utworzenie modelu jakościowego, operującego **pojęciami lingwistycznymi**, np. mały, średni, duży. Modele takie mogą znaleźć zastosowanie w dziedzinach obejmujących szeroko rozumiane problemy społeczne-ekonomiczne, gdzie nie jest możliwe utworzenie modelu obejmującego wszystkie czynniki mające wpływ na proces.

CO UMOŻLIWIA SYSTEM NEURONIX UŻYTKOWNIKOWI ?

System Neuronix umożliwia wszechstronną analizę danych poprzez tworzenie modeli różnorodnych procesów spotykanych w ekonomii i technice. Zastosowanie sieci neuronowej pozwala na automatyczne utworzenie modelu bez konieczności głębokiej znajomości modelowanego procesu. Przykładowo, jeżeli Użytkownik chciałby utworzyć model wyceny akcji na giełdzie musi zgromadzić takie dane, które jego zdaniem mają wpływ bezpośredni lub pośredni na cenę akcji. Użytkownik może włączyć do analizy dane, które mogą nie mieć żadnego wpływu na cenę akcji, w tym wypadku sieć neuronowa cechuje się zdolnością do rozróżniania danych istotnych od mniej istotnych i sama dokona wyboru. Cały mechanizm budowy modelu sprowadza się więc do zapisania danych w **arkuszu kalkulacyjnym,** podobnie jak to czynimy na przykład w programie Excel w postaci tabeli, w której określimy wejścia i wyjścia jako cenę akcji. Do przykładowych wejść modelu można zaliczyć wycenę innych akcji, poziomy indeksów giełdy, zmiany stóp procentowych i inne czynniki wpływające na kształtowanie się ceny akcji. Znaczącym osiągnięciem twórców pakietu Neuronix jest opracowanie modułu kodującego, który pozwala tworzyć tzw. **modele jakościowe.** W modelu jakościowym użytkownik może pracować bezpośrednio na danych w postaci symbolicznej, są to tzw. **zmienne lingwistyczne**. Oznacza to w praktyce, że jeżeli chciałby opisać aktualny stan giełdy może użyć zmiennej wejściowej, której jedną z możliwych wartości jest "umiarkowana hossa". Pojęcia tego nie można precyzyjnie wyrazić w postaci konkretnej liczby. W podobny sposób trudno byłoby sprecyzować nastrój inwestorów. Czy można go określić za pomocą jakiegoś wskaźnika ? Oczywiście jest to możliwe, ale konstrukcja takiego wskaźnika byłaby bardzo złożona i powstałby problem jak to zrobić ? Dużo prościej jest taką daną wprowadzić na wejście sieci i pozwolić jej podjąć samodzielną decyzję o włączeniu jej do budowanego modelu.

2

Zanim rozpoczniesz pracę z systemem Neuronix

W rozdziale:

- Sieć neuronowa a aplikacje hybrydowe
- Współpraca z innymi aplikacjami pakietu SPHINX
- Rodzaje plików używanych przez system Neuronix

Cenne informacje

Ćwiczenia z komputerem

Koniecznie przeczytaj

SIEĆ NEURONOWA A APLIKACJE HYBRYDOWE

Pod pojęciem **aplikacji hybrydowej** możliwej do utworzenia w pakiecie sztucznej inteligencji SPHINX, należy rozumieć połączenie dowolnej liczby **źródeł wiedzy** w postaci bazy wiedzy systemu ekspertowego lub sieci neuronowej w ramach **tablicowej architektury** systemu ekspertowego.

WSPÓŁPRACA Z INNYMI APLIKACJAMI PAKIETU SPHINX

Możliwe jest uruchomienie sieci z poziomu programu napisanego przy użyciu języka reprezentacji wiedzy „Sphinx". Uruchamianie sieci z poziomu bloku „control" bazy wiedzy umożliwia budowanie aplikacji hybrydowych, które wykorzystują system ekspertowy i sieć neuronową. W systemie ekspertowym PC-Shell można wykorzystywać wszystkie instrukcje odwołujące się do wyuczonej sieci neuronowej.

W celu uruchomienia sieci neuronowej muszą istnieć następujące pliki:
- plik projektu: rozszerzenie .npr (np. sinus.npr),
- plik aktualnych wartości parametrów sieci neuronowej: rozszerzenie .cpv (np. sinus.cpv),
- plik skali: rozszerzenie ._nn (np. sinus._nn),
- plik słownika symboli (jeżeli plik uczący zawierał kolumny wejściowe lub wyjściowe z wartościami symbolicznymi): rozszerzenie ._dc (np. sinus._dc).

Pliki binarne typu ._nn oraz ._dc są plikami wygenerowanymi przez symulator Neuronix. Aby stworzyć wspomniane zbiory, należy uruchomić uczenie sieci lub uruchomić generację plików binarnych.

W języku Sphinx został zdefiniowany rekord **NeuralNet**, który służy do opisu wejść i wyjść sieci:

```
record NeuralNet
begin
    char Name;
    double DValue;
    char Symbol;
end;
```

W bloku „sources" można zadeklarować kilka sieci neuronowych, które zostały wcześniej wyuczone. Jako źródło wiedzy traktuje się tu plik projektu (rozszerzenie .npr). Inicjalizacja sieci odbywa się poprzez instrukcję **initNetwork**, której parametrem jest nazwa źródła wiedzy. Uruchomienie sieci uzyskuje się instrukcją **runNetwork**, której pierwszym parametrem jest źródło wiedzy, drugim tablica wejściowa a trzecim tablica wyjściowa. Zarówno tablica wejściowa jak i wyjściowa są tablicami rekordów **NeuralNet**. W obu przypadkach konieczne jest przypisanie prawidłowych wartości polu „Name", które oznacza nazwę wejścia lub wyjścia sieci. W tablicy wejściowej należy również podstawić odpowiednie wartości w polu „DValue" lub „Symbol" w zależności od rodzaju danych jakie podawane są sieci. Jeżeli wejście operuje na wartościach numerycznych, to wartość liczbową przypisuje się polu „DValue". Jeżeli wejście sieci to jedna z wartości symbolicznych to podstawia się ją pod pole „Symbol". Kolejność wejść i wyjść w tablicy nie jest ważna. Po uruchomieniu sieci następuje przepisanie wartości wyjściowych do pól rekordów tablicy wyjściowej. Jeżeli wyjście jest numeryczne to wypełniane jest pole „DValue" a jeżeli symboliczne to „Symbol". Usunięcie sieci następuje po wywołaniu instrukcji **delNetwork**, której jedynym parametrem jest źródło wiedzy. W bloku „control" można zadeklarować **dowolną ilość sieci neuronowych** i używać ich jednocześnie bez konieczności usuwania sieci po jej użyciu. W ten sposób wyjście jednej sieci może posłużyć jako wejście innej, dając tym samym sposobność do kaskadowego łączenia kilku sieci neuronowych.

PRZYKŁAD WYWOŁANIA SIECI NEURONOWEJ Z POZIOMU BAZY WIEDZY

```
knowledge base sinus

  sources
    netSin :
        type neural_net
        file "c:\\neuronix\sinus.npr";
  end;

 control
    char NetSin;

    record NeuralNet SinWe[3];
    record NeuralNet SinWy[3];

    SinWe[0].Name := "y-1";
    SinWe[0].DValue := 0.5877;
    SinWe[1].Name := "y-2";
    SinWe[1].DValue := 0.9510;
    SinWe[2].Name := "y-3";
    SinWe[2].DValue := 0.9510;

    SinWy[0].Name := "y";
    SinWy[1].Name := "#ćwiartka";
    SinWy[2].Name := ">moduł";

    initNetwork ( netSin );
    runNetwork ( netSin, SinWe, SinWy );
    delNetwork ( netSin);

    char SSinWy;
    precision ( 10,4 );
    ntos ( SinWy[0].DValue, SSinWy );

    messageBox( 0, 0, "Wartość wyjścia y =", SSinWy );
    messageBox( 0, 0, "Wartość wyjścia ćwiartka =", SinWy[1].Symbol );
    messageBox( 0,0,"Wartość wyjścia moduł =",SinWy[2].Symbol );
 end;
end;
```

Program w języku reprezentacji wiedzy odwołuje się poprzez plik projektu do pliku wag. Należy pamiętać o tym, że przed uruchomieniem sieci z poziomu języka Sphinx musi być uaktualniona wersja pliku wag. Aby zapisać aktualne wartości wag trzeba posłużyć się przyciskiem „Zapis wag” w oknie uczenia lub przyciskiem „Koniec”, który spowoduje automatyczny zapis wag do pliku.

URUCHOMIENIE SIECI Z POZIOMU PROGRAMU WYKONYWALNEGO

Do programu Neuronix dołączana jest biblioteka importowa bckp.lib dla bckp.dll. Przy użyciu języków programowania np. Borland C++, Builder, Delphi można korzystać z funkcji, jakie zostały udostępnione w bibliotece dynamicznej.

Sieć neuronowa może być uruchomiona, jeżeli istnieją następujące pliki:
- plik projektu: rozszerzenie .npr (np. sinus.npr),
- plik aktualnych wartości parametrów sieci neuronowej: rozszerzenie .cpv (np. sinus.cpv),
- plik skali: rozszerzenie .nn_ (np. sinus.nn_),
- plik słownika symboli (jeżeli plik uczący zawierał kolumny wejściowe lub wyjściowe z wartościami symbolicznymi): rozszerzenie .dc_ (np. sinus.dc_).

Pliki binarne typu **.nn_** oraz **.dc_** są plikami wygenerowanymi przez symulator Neuronix. Aby stworzyć wspomniane zbiory, należy uruchomić uczenie sieci lub uruchomić generację plików binarnych.

Pojedyncza wartość wejściowa lub wyjściowa reprezentowana jest przez strukturę **TNNValue**:

```
struct TNNValue
{
   // nazwa wejścia
   char far pszOName [ 255 ];
   // wartość numeryczna wejścia lub wyjścia
   double lfOValue;
   // wartość symboliczna wejścia lub wyjścia
   char far pszOValue [ 17 ];
};
```

int far INITNETWORK (char far* pszFullNprName)

Zainicjowanie sieci neuronowej **pszFullNprName** jest pełną ścieżką dostępu do pliku projektu. Funkcja zwraca wartość nieujemną (numer nadawany inicjowanej sieci) jeżeli inicjacja sieci powiodła się lub wartość ujemną, jeżeli podczas inicjowania wystąpił błąd. Nazwę błędu można uzyskać po wywołaniu funkcji GETNRXERROR, której parametrem jest wartość zwrócona przez funkcję.

int far RUNNETWORK (int iNet, TNNValue huge* pInput, int iInputs, TNNValue huge* pOutput, int iOutputs)

Uruchomienie wskazanej sieci
iNet - numer zainicjowanej sieci,
pInput - wskaźnik na tablicę wejściową struktur zawierających nazwy wejść sieci i ich wartości,
iInputs - ilość wejść sieci,
pOutput - wskaźnik na tablicę wyjściową struktur zawierających nazwy wyjść,
iOutputs - ilość wyjść sieci,

Funkcja zwraca wartość ujemną jeżeli wystąpił błąd. Nazwę błędu można uzyskać po wywołaniu funkcji GETNRXERROR, której parametrem jest wartość zwrócona przez funkcję. Jeżeli funkcja nie zwróciła błędu to do tablicy wyjściowej przepisane zostały wartości na wyjściu sieci. Nie jest wymagane podanie wszystkich wyjść sieci. Po wywołaniu funkcji zostaną wypełnione tylko wskazane wyjścia.

int far DELETENETWORK (int iNet)

Usunięcie wskazanej sieci
iNet - numer zainicjowanej sieci,
Funkcja zwraca wartość ujemną jeżeli wystąpił błąd. Nazwę błędu można uzyskać po wywołaniu funkcji GETNRXERROR, której parametrem jest wartość zwrócona przez funkcję.

void GETNRXERROR (int iErr, char far* pszBuffer, int iBufferLen)

Pobranie opisu błędu, jaki pojawił się w trakcie wykonywania jednej z funkcji neuronowych
iErr - kod błędu (ujemny),
pszBuffer - wskaźnik na bufor, do którego będzie zwrócony opis,
iBufferLen - wielkość bufora,

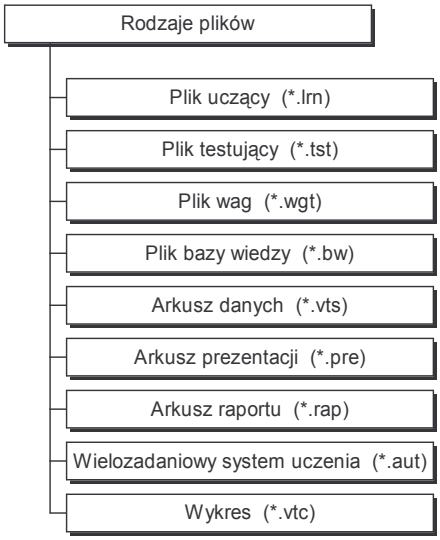
Przykładowe wywołanie sieci neuronowej z poziomu języka C++

```
void Error ( int iErr )
{
   char pszBuffer [ 256 ];
   GETNRXERROR ( iErr, pszBuffer, 255 );
   MessageBox ( NULL, pszBuffer, „Error”, MB_OK );
}

void RunNetwork ()
{
   TNNValue huge pInput [3];
   TNNValue huge pOutput [3];

   // wypełnienie tablicy wejściowej
   strcpy ( (pInput [0]).pszOName, „y-1” );
   (pInput[0]).lfOValue = 0.5877;
   strcpy ( (pInput [1]).pszOName, „y-2” );
   (pInput[1]).lfOValue = 0.9510;
   strcpy ( (pInput [2]).pszOName, „y-3” );
   (pInput[2]).lfOValue = 0.9510;

   // wypełnienie tablicy wyjściowej
   strcpy ( (pOutput [0]).pszOName, „y” );
   strcpy ( (pOutput [1]).pszOName, „#ćwiartka” );
   strcpy ( (pOutput [2]).pszOName, „>moduł” );


   int iNet = INITNETWORK („c:\\sphinx 4.0\\projekt\\sinus.npr” );

   if ( iNet >= 0 )
   {
      int iResult = RUNNETWORK ( iNet, pInput, 3,pOutput, 3 );
      if ( iResult >= 0 )
      {
        char pszMsg [ 256 ];
        sprintf ( pszMsg, „y = %lf, #ćwiartka = %s, >moduł
   = %s”,
        (pOutput[0]).lfOValue,
        (pOutput[1]).pszOValue,
        (pOutput[2]).pszOValue );

        MessageBox ( NULL, pszMsg, „Result”, MB_OK );
      }
```

```
      else
        Error ( iResult );

      DELETENETWORK ( iNet );
   }
   else
      Error ( iNet );
}
```

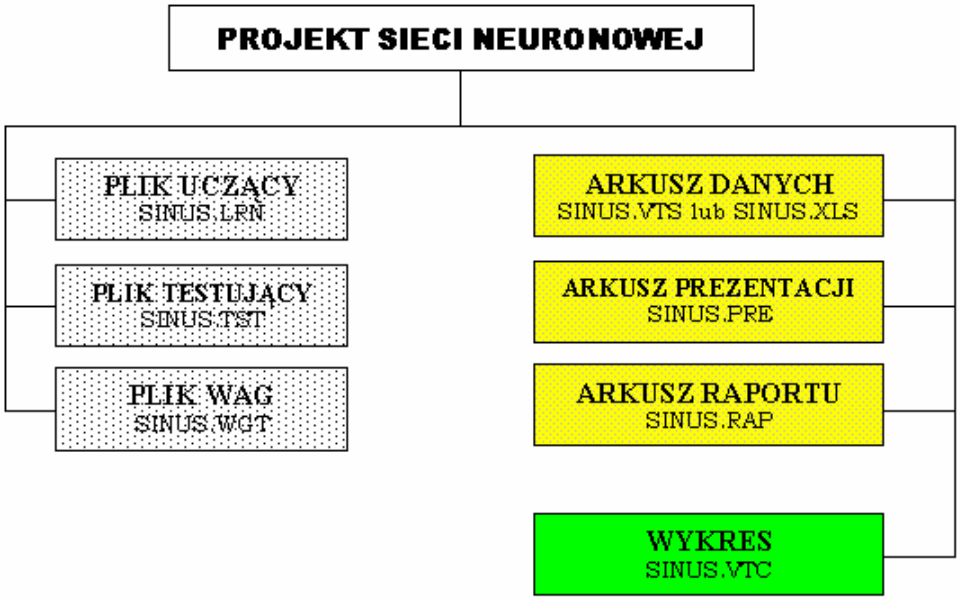
RODZAJE PLIKÓW UŻYWANE PRZEZ SYSTEM NEURONIX

STRUKTURA PLIKÓW APLIKACJI NEURONIX 2.3



Rys. 1. Pliki używane przez z sytem Neuronix

PROJEKT SIECI NEURONOWEJ



Rys. 2. Schemat struktury projektu neuronowego

3

Menu systemu Neuronix

W rozdziale:

- Grupa Plik
- Grupa Edycja
- Grupa Wstaw
- Grupa Sieć
- Grupa Narzędzia
- Grupa Pomoc
- Zasoby projektu
- Opcje sieci neuronowej
- Uczenie sieci - informacje teoretyczne
- Monitoring procesu uczenia i testowania
- Uczenie sieci

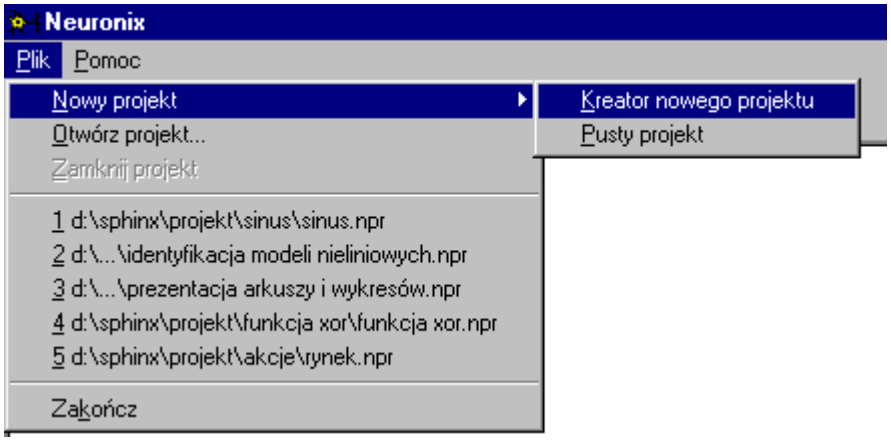
Cenne informacje

Ćwiczenia z komputerem

Koniecznie przeczytaj

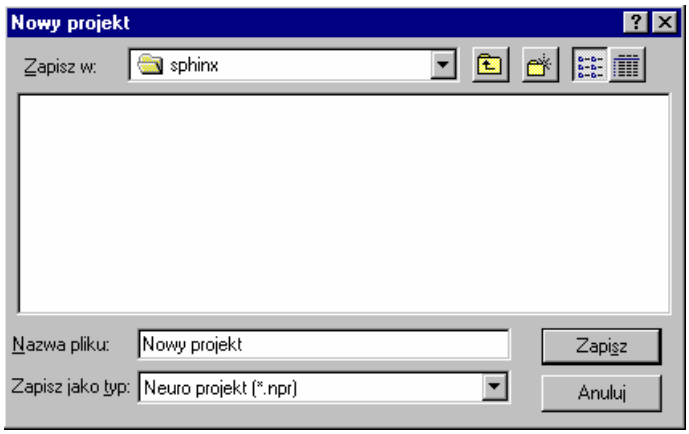
GRUPA PLIK

Grupa **Plik** pozwala na obsługę plików w których są przechowywane dane niezbędne dla prawidłowej pracy systemu Neuronix. Ponadto zawiera opcje służące drukowaniu zawartości arkuszy i wykresów. Podstawowymi opcjami są **Nowy Projekt** oraz **Otwórz Projekt,** które za pomocą Kreatora Nowego Projektu lub bez niego pozwalają na utworzenie nowego projektu sieci neuronowej lub wczytanie z dysku już istniejącego. Należy podkreślić istotną funkcję, jaką pełni projekt. Jest nią zebranie wszystkich potrzebnych i czasami bardzo różnorodnych danych, niezbędnych w procesie uczenia i uruchamiania sieci neuronowej w pewną zintegrowaną i logiczną całość. Każdy projekt sieci neuronowej pozwala na utworzenie tylko jednej sieci dla której jest możliwe stosowanie zamiennie plików z danymi uczącymi, testowymi i innymi. W ramach projektu zdefiniowane są również parametry topologii sieci oraz parametry procesu uczenia i testowania. Rys. 3 przedstawia wygląd grupy poleceń **Plik** po uruchomieniu aplikacji Neuronix lub też po zamknięciu bieżącego projektu.

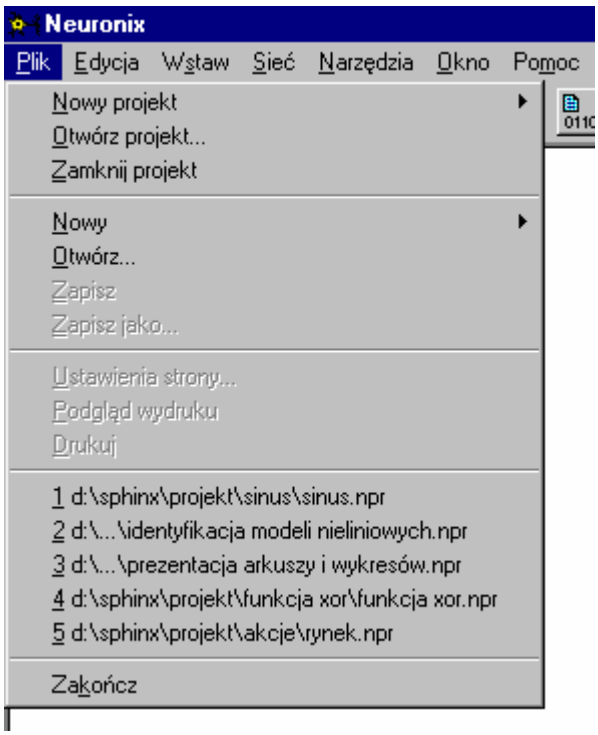


Rys. 3. Podstawowe opcje grupy **Plik**

Po utworzeniu lub wczytaniu nowego projektu Użytkownik ma dostęp do bardziej rozbudowanego menu, w skład którego wchodzą wspomniane wcześniej: **Podgląd wydruku**, **Ustawianie strony** oraz **Drukowanie strony**. Pełne informacje dotyczące tych opcji znajdują się w rozdziale 8 podręcznika pod tytułem **Obsługa wydruków w systemie Neuronix.** Na samym dole grupy **Plik** znajdują się nazwy projektów wcześniej otwartych. Wybranie nazwy pozwala na automatyczne wczytanie już istniejącego projektu z dysku. Widok pełnej grupy **Plik** wraz z opcjami dostępnymi **tylko po otwarciu** projektu sieci neuronowej pokazano na rys. 5.

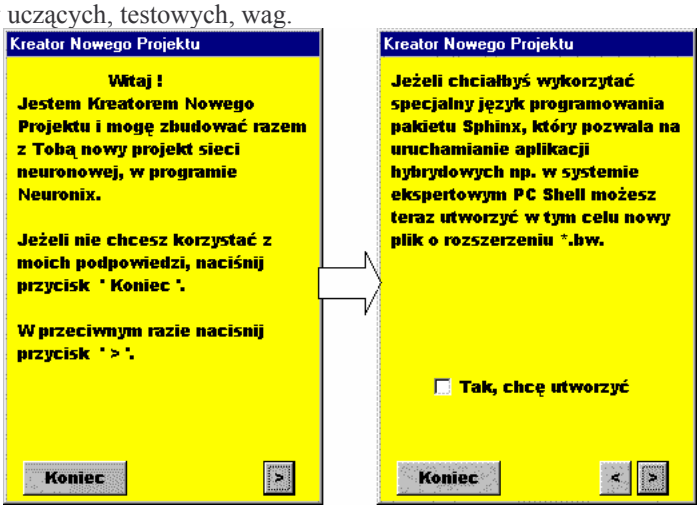


Rys. 4. Dialog obsługi pliku dla nowego projektu



Rys. 5. Pełna grupa **Plik**, po wczytaniu projektu

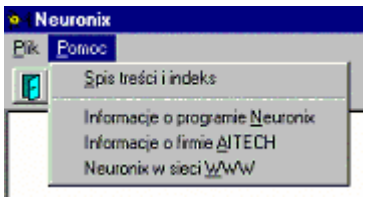
Użycie opcji **Nowy Projekt | Kreator Nowego Projektu** pozwala na utworzenie nowego projektu sieci neuronowej wspomagane kreatorem. Użycie kreatora przyśpiesza proces tworzenia oraz pozwala na zbudowanie poprawnego projektu sieci neuronowej nawet początkującemu Użytkownikowi. Poszczególne etapy tworzenia projektu nie zawsze są intuicyjne i wymagają pewnych założeń dotyczących przyszłych właściwości sieci, dlatego podpowiedzi kreatora mogą stać się w pewnych wypadkach pomocne. Zaleca się również korzystanie z kreatora podczas tworzenia projektów powtarzalnych, celem wyeliminowania mało twórczych działań takich jak na przykład zakładanie nowych plików uczących, testowych, wag.



Rys. 6. Przykładowe komunikaty kreatora **Nowego projektu**

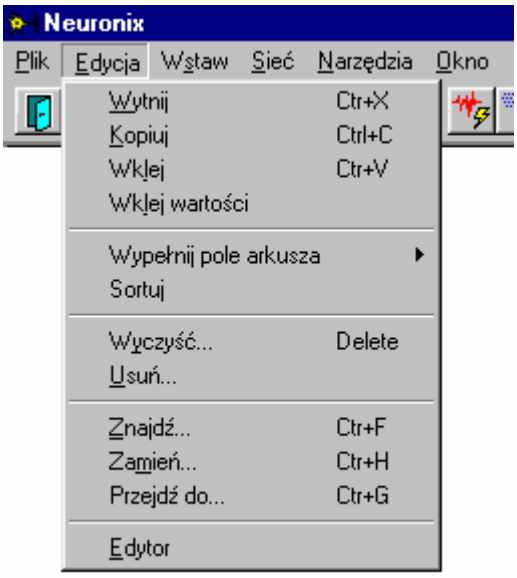
GRUPA POMOC

Grupa **Pomoc** umożliwia uzyskanie szczegółowych wyjaśnień odnośnie obsługi Neuronix'a oraz pozwala na połączenie z witryną internetową, na której znajdują się najnowsze informacje dotyczące produktów firmy AITECH oraz przykłady i uwagi Użytkowników.



Rys. 7. Grupa Pomoc

EDYCJA



Rys. 8. Grupa Edycja

WYTNIJ, KOPIUJ, WSTAW

Opcja pozwala na pracę z zaznaczonymi obszarami arkusza kalkulacyjnego, poprzez ich wycinanie, kopiowanie i wstawianie w dowolny inny obszar tego samego arkusza lub innego.

IDŹ DO

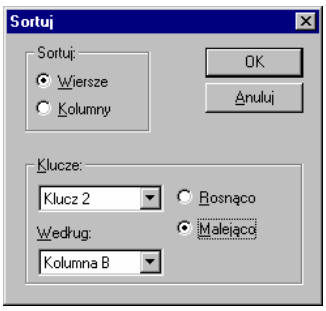
Przesunięcie arkusza do podanego celu. Celem może być pojedyncza komórka (np. C23), zakres komórek (np. A3:B7), nazwa komórki lub cały arkusz.

SORTUJ

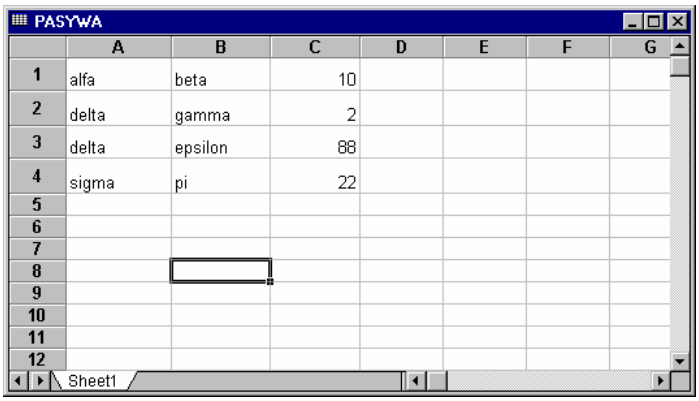
Sortowanie wierszy lub kolumn w zaznaczonym zakresie. Jeśli sortujemy wiersze, każdy wiersz w zaznaczonym zakresie stanowi jeden rekord, sortowany w całości (analogicznie przy sortowaniu kolumn pojedynczym rekordem jest kolumna). Sortowanie może odbywać się według maksymalnie 3 kluczy (pól rekordu, czyli komórek wiersza lub kolumny). Najpierw porównywane są klucze nr 1, w przypadku ich równości klucze nr 2, na końcu nr 3 (jeśli wszystkie są ustawione). Ustawienia kluczy dokonujemy poprzez wybranie numeru klucza w polu *Klucze* (np. Klucz 2), a następnie przypisania mu pożądanej kolumny (lub wiersza przy sortowaniu kolumn) w polu *Według* (np. Kolumna E). Dla każdego klucza możemy określić, czy sortowanie ma być według rosnących, czy też malejących wartości tego klucza.

Przykład: 3-1

Sortujemy wierszami zakres A1C4; kluczem 1 jest kolumna A (rosnąco), kluczem 2 kolumna B (malejąco)



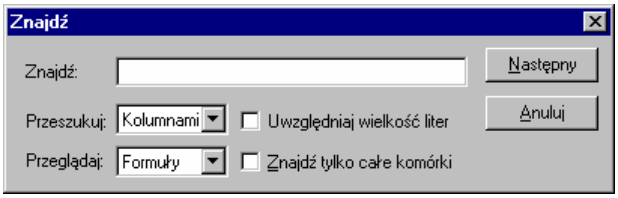
Rys. 9. Opcja sortowania



Rys. 10. Arkusz po operacji sortowania

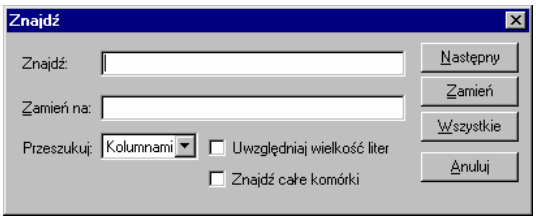
ZNAJDŹ

Odszukiwanie komórek o zadanym wzorcu. Przeszukiwanie może odbywać się kolumnami lub wierszami, poszukiwany wzorzec może być interpretowany jako formuła lub wartość komórki (opcja *Przeglądaj*). W poszukiwaniu może być uwzględniana wielkość liter oraz dokładne dopasowanie wzorca do zawartości komórki. Poszukiwanie może być kontynuowane poprzez naciśnięcie przycisku *Następny*.



Rys. 11. Opcja **Znajdź** określone wyrażenie

ZASTĄP

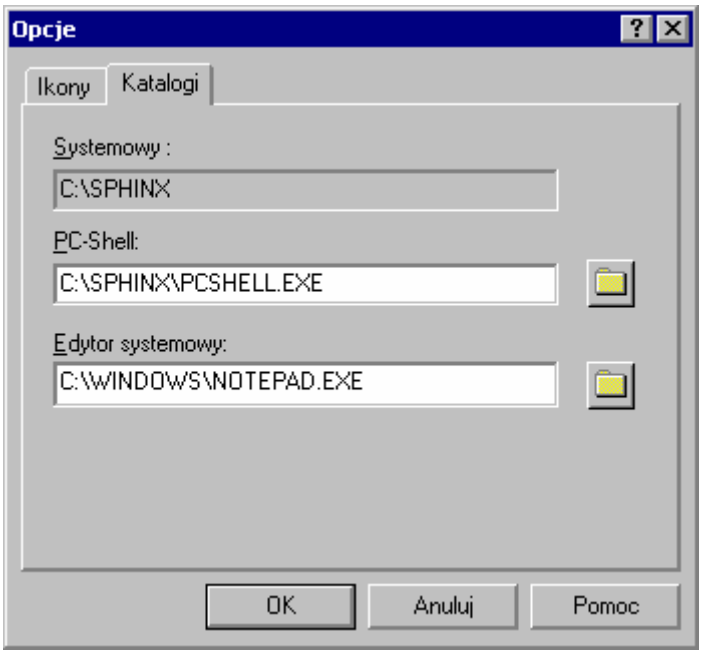


Rys. 12. Opcja zastępowania dowolnego wyrażenia innym

Zastępowanie jednego wzorca innym. Przeszukiwanie może odbywać się kolumnami lub wierszami. W poszukiwaniu może być uwzględniana wielkość liter oraz dokładne dopasowanie poszukiwanego wzorca do zawartości komórki. Zastąpienie wzorca może być wykonywane pojedynczo dla poszczególnych wystąpień wzorca (przycisk *Zastąp*) lub automatycznie dla wszystkich wystąpień (przycisk *Wszystkie*). Przycisk *Następny* odszukuje następne wystąpienie wzorca

EDYTOR

Opcja uruchamia dowolny edytor tekstu zdefiniowany za pomocą dialogu konfiguracyjnego, który jest dostępny w menu głównym **Narzędzia | Konfiguracja**. Przykład definicji Użytkownika, stosującego edytor systemowy Windows 9x/NT przedstawia rys. 13. Definicja składa się z określonej ścieżki dostępu, która pozwoli na wywołanie w systemie Neuronix pliku wykonywalnego *.exe edytora.



Rys. 13. Przykład definicji ścieżki dostępu do edytora tekstu

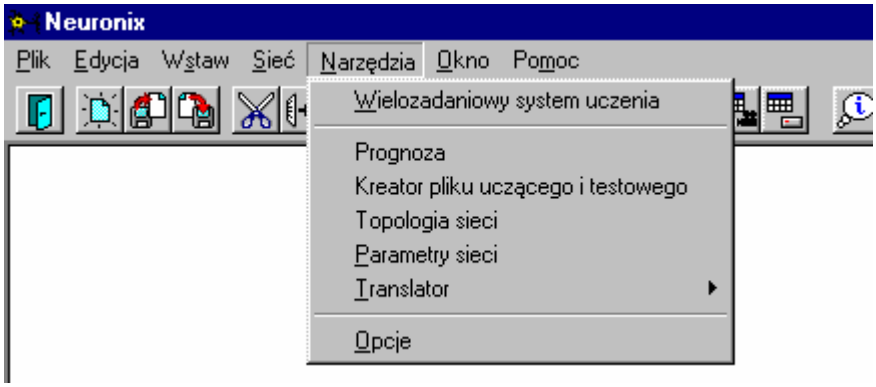
WSTAW

Rys. 14. Grupa Wstaw

SIEĆ

Rys. 15. Grupa **Sieć**

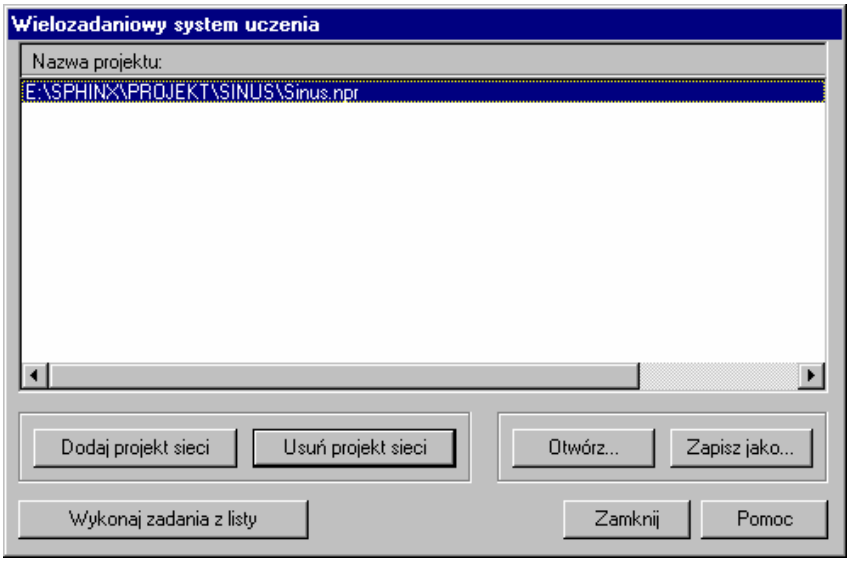
NARZĘDZIA



Rys. 16. Grupa **Narzędzia**

WIELOZADANIOWY SYSTEM UCZENIA

Wielozadaniowy system uczenia wspomaga proces uczenia wielu sieci wchodzących w skład różnych projektów. Typowym zadaniem dla takiego systemu może być potrzeba wyuczenia dużej liczby sieci z których każda prognozuje przebieg wybranego kursu akcji, a całość służy dla prognozy notowań giełdowych w ramach aplikacji wykonanej w pakiecie Sphinx. Rozpoczęcie procesu uczenia sieci dokonuje się poprzez naciśnięcie przycisku **Wykonaj zadania z listy**. Proces uczenia polega na otwarciu projektu znajdującego się na liście (kolejność od góry do dołu), uruchomieniu procesu uczenia oraz zamknięciu projektu i automatycznym przejściu do następnego projektu z listy. Aktualnie otwarty projekt jest wskazywany za pomocą podświetlenia.

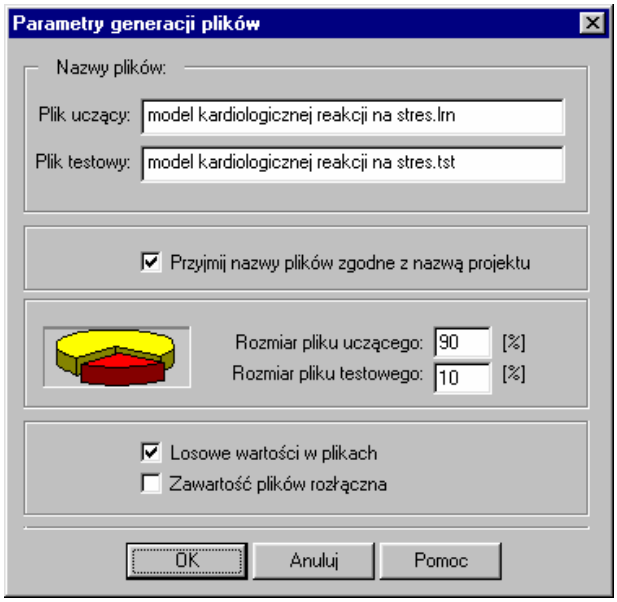


Rys. 17. Wielozadaniowy system uczenia

KREATOR PLIKU UCZĄCEGO I TESTOWEGO

Kreator wspomaga procedurę tworzenia zbioru testowego i uczącego, które są niezbędne dla poprawnego działania sieci neuronowej. Wyręcza Użytkownika od uciążliwych operacji edytorskich oraz pozwala zautomatyzować poszczególne czynności. Do podstawowych opcji okna dialogowego należą **Nazwa generowanego pliku uczącego** i **Nazwa generowanego pliku testowego**. Pozwalają one nadać nazwy dla generowanych plików, domyślenie nazwy są zgodne z nazwą projektu sieci neuronowej. Kolejnymi opcjami są rozmiary plików, które wyrażone są w **procentach** rozmiaru pliku źródłowego. Poprzez rozmiar należy rozumieć **liczbę wierszy** (wzorców uczących) znajdujących się w poszczególnych arkuszach. Zakłada się, że zawartość arkusza źródłowego stanowi 100%. Łączna wartość rozmiaru pliku uczącego i testowego może być dopełnieniem 100%, lub przewyższać tę zawartość. Wartość większa niż 100% oznacza, że niektóre fragmenty pliku źródłowego powtarzają się w plikach uczącym i testowym. Dla wartości 200% plik uczący i testowy jest kopią pliku źródłowego. Przekraczanie wartości 100% jest możliwe po zaznaczeniu pola

Zawartość plików rozłączna. O tym, która linia pliku źródłowego znajdzie się w pliku uczącym, a która w pliku testowym decyduje kolejność wierszy arkusza pliku źródłowego lub po zaznaczeniu pola **Losowe wartości w plikach** specjalna funkcja wybierająca wiersze w kolejności losowej.



Rys. 18. Kreator pliku uczącego i testowego

Skorzystanie z Kreatora pliku uczącego i testowego jest możliwe jeżeli Użytkownik wskaże wcześniej arkusz danych, zawierający sformatowane dane źródłowe niezbędne dla uczenia sieci (zbiór danych).

PARAMETRY SIECI

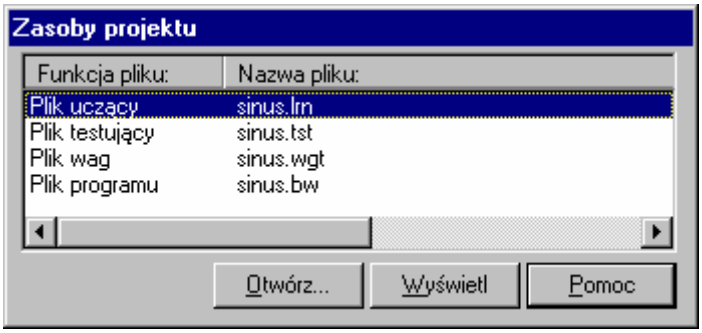
Dokładne sposoby doboru i określania parametrów sieci neuronowych zawiera podrozdział **Opcje sieci neuronowej**.

TRANSLATOR

Opcja pozwala na uruchamianie aplikacji pakietu Sphinx.

ZASOBY PROJEKTU

Okienko Zasoby projektu jest widoczne po otwarciu w systemie Neuronix nowego projektu.



Rys. 19. Zasoby projektu

- **plik uczący**

Parametr zawierający ścieżkę dostępu do pliku zawierającego wzorce uczące. Jego ustawienie jest niezbędne do rozpoczęcia uczenia sieci neuronowej. Dwukrotne naciśnięcie klawisza myszki powoduje otwarcie arkusza danych, wraz z otwarciem pliku uczącego, jeżeli jakikolwiek został wskazany.

- **plik testowy**

Parametr zawierający ścieżkę dostępu do pliku zawierającego plik testowy. Jego ustawienie jest opcjonalne, przy czym jeżeli wartość parametru '*częstość testowania*' widoczna w dialogu **Opcje sieci neuronowej** na zakładce **Parametry procesu testowania** jest większa od zera wtedy plik musi zostać wskazany obligatoryjnie. Dwukrotne naciśnięcie klawisza myszki powoduje otwarcie arkusza danych wraz z otwarciem pliku testowego, jeśli jakikolwiek został wskazany.

- **plik wag**

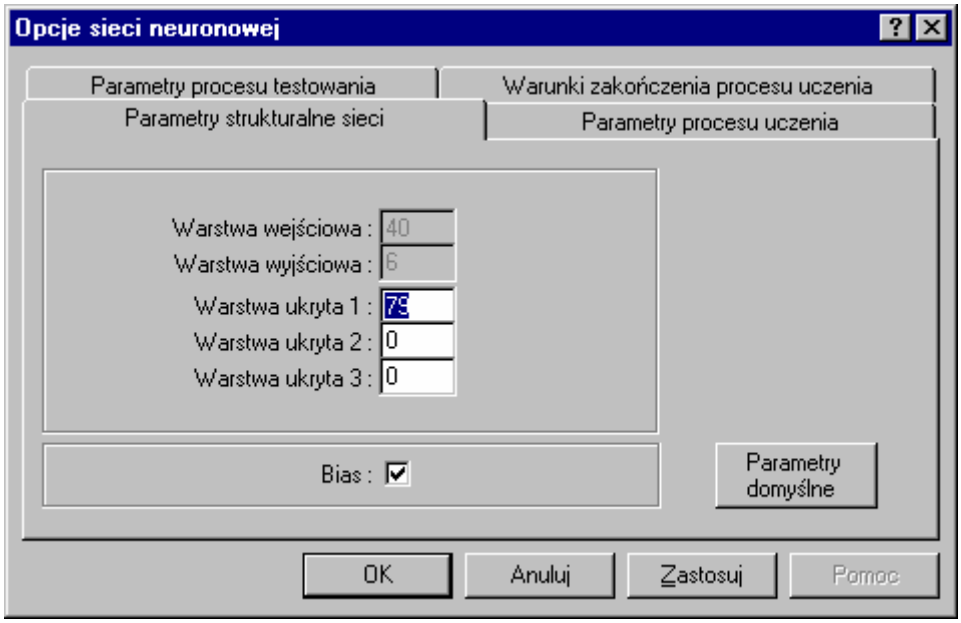
Parametr zawierający ścieżkę dostępu do pliku zawierającego pliku wag. Jego ustawienie jest niezbędne do rozpoczęcia uczenia sieci neuronowej. W przypadku nie istniejącego pliku można podać nową nazwę. Dwukrotne naciśnięcie klawisza myszki powoduje otwarcie edytora tekstu wraz z otwarciem pliku wag, jeśli jakikolwiek został wskazany.

- **plik programu**

Parametr zawierający ścieżkę dostępu do pliku zawierającego plik programu (bazy wiedzy) napisanego w języku reprezentacji wiedzy Sphinx. Dwukrotne naciśnięcie klawisza myszki powoduje otwarcie edytora tekstu wraz z otwarciem pliku programu, jeśli jakikolwiek został wskazany.

OPCJE SIECI NEURONOWEJ

PARAMETRY STRUKTURALNE SIECI



Rys. 20. Zakładka: Parametry strukturalne sieci

Warstwa wejściowa

Określa ilość neuronów wejściowych. Parametr może przyjmować wartości od 1 do 10000. Parametr ten jest automatycznie wypełniany na etapie generacji binarnej plików. Wtedy Neuronix analizuje plik uczący i na podstawie jego zawartości ustala liczbę neuronów wejściowych i wyjściowych. Z tego powodu użytkownik nie ma bezpośredniej możliwości ingerowania w ten parametr.

W przypadku wejściowych kolumn kodowanych następuje niejawne dodanie wejść sieci w celu odpowiedniego zakodowania symboli. Dlatego też może zaistnieć sytuacja, że ilość kolumn wejściowych nie jest równa temu parametrowi.

Warstwa ukryta 1

Określa liczbę neuronów w pierwszej (licząc od warstwy wejściowej) warstwie ukrytej. Parametr może przyjmować wartości od 0 do 10000. Zmiana tego parametru powoduje zainicjowanie losowymi wartościami wag sieci.

Warstwa ukryta 2

Określa liczbę neuronów w drugiej (licząc od warstwy wejściowej) warstwie ukrytej. Parametr może przyjmować wartości od 0 do 10000. Zmiana tego parametru powoduje zainicjowanie losowymi wartościami wag sieci.

Warstwa ukryta 3

Określa liczbę neuronów w trzeciej (licząc od warstwy wejściowej) warstwie ukrytej. Parametr może przyjmować wartości od 0 do 10000. Teoretycznie dwie warstwy ukryte wystarczają do rozwiązania dowolnego problemu. Trzecia warstwa ukryta została wprowadzona tylko ze względów edukacyjnych i badawczych. Zmiana tego parametru powoduje zainicjowanie losowymi wartościami wag sieci.

Warstwa wyjściowa

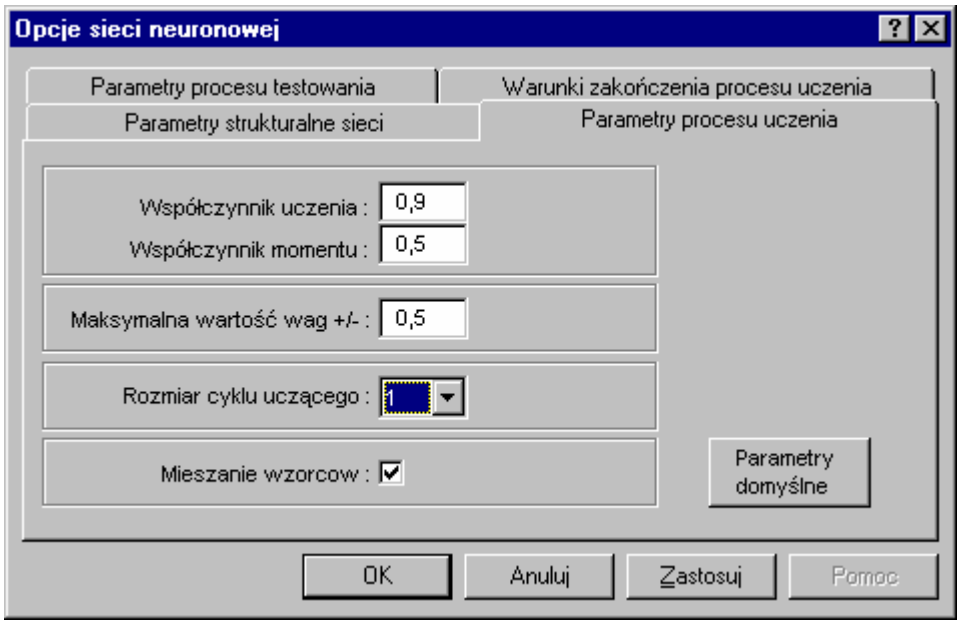
Określa ilość neuronów wejściowych. Parametr może przyjmować wartości od 1 do 10000. Parametr ten jest automatycznie wypełniany wraz z parametrem **warstwa wejściowa** na etapie generacji binarnej plików.

W przypadku wyjściowych kolumn kodowanych następuje niejawne dodanie wyjść sieci w celu odpowiedniego zakodowania symboli. Dlatego też może zaistnieć sytuacja, że ilość kolumn wyjściowych nie jest równa temu parametrowi.

Bias

Określa czy ma być stosowany dodatkowy neuron, którego wyjście jest równe 1. W przypadku wartości „tak” wszystkie neurony w warstwie ukrytej i wyjściowej będą połączone z tym dodatkowym neuronem. Rozwiązanie to powoduje lepszą stabilność w trakcie uczenia i jest klasycznym przykładem na poprawienie osiągów sieci. Możliwość odłączenia tego neuronu udostępniono tylko ze względów dydaktycznych i badawczych.

PARAMETRY PROCESU UCZENIA



Rys. 21. Zakładka: Parametry procesu uczenia

Współczynnik uczenia

Parametr określa wielkość współczynnika uczenia. Im większa wartość tego parametru tym szybsze staje się przeszukiwanie przestrzeni rozwiązań. Zbyt duża wartość może doprowadzić do przeoczenia najlepszego rozwiązania. Wartość tego współczynnika może należeć do przedziału < 0,1 >.

Współczynnik momentu

Parametr określa wielkość współczynnika momentu decydującego o stabilności procesu uczenia. Im większa wartość współczynnika momentu tym większa staje się bezwładność procesu poszukiwania przestrzeni rozwiązań. Wartość tego współczynnika może należeć do przedziału < 0,1).

Maksymalna wartość wag

Określa przedział liczbowy, z którego losowane są wartości wag sieci podczas jej inicjalizacji. Przedział ten zawiera się od **-max** do **max**, gdzie zmienna max jest równa maksymalnej wartości wag. Rozkład wag po inicjalizacji jest w przybliżeniu normalny o średniej równej 0.

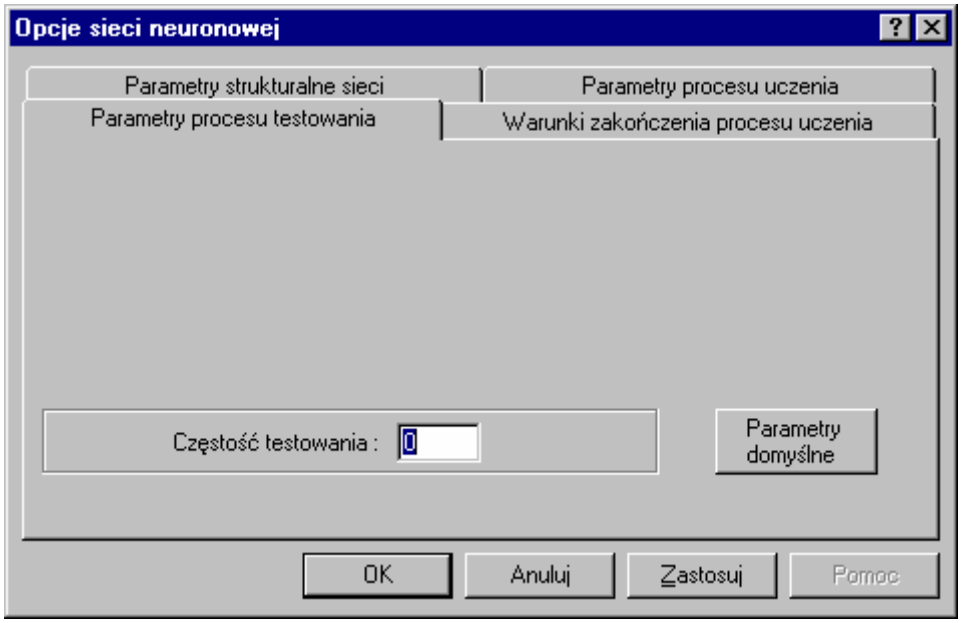
Rozmiar cyklu uczącego

Określa ilość wzorców jakie będą prezentowane sieci neuronowej, po których nastąpi zmiana wag sieci, czyli etap uczenia. Zgodnie z algorytmem metody wstecznej propagacji błędu wyróżnia się dwa zasadnicze etapy tego sposobu pozyskiwania wiedzy. Pierwszy z nich polega na zaprezentowaniu sieci wybranego wzorca i aktywację wszystkich neuronów sieci od wejść w kierunku wyjść. Drugi obejmuje obliczenie powstałych błędów i modyfikację wag od wyjść w kierunku wejść. Ilość wzorców sieci jaka prezentowana jest w pierwszym etapie jest omawianym parametrem. W obecnej wersji symulatora dostępne są dwie wartości tego parametru: „1” - co oznacza, że po każdej prezentacji wzorca następuje modyfikacja wag oraz „n” - co będzie powodowało, że etap modyfikacji wag zostanie przeprowadzony po prezentacji wszystkich wzorców ze zbioru uczącego.

Mieszanie wzorców

Parametr ten decyduje o tym, czy w trakcie uczenia wzorce uczące będą podawane w losowej kolejności w ramach jednej epoki. Mieszanie wzorców jest niezwykle ważne w przypadku, gdy rozmiar cyklu uczącego jest równe „1”.

PARAMETRY PROCESU TESTOWANIA

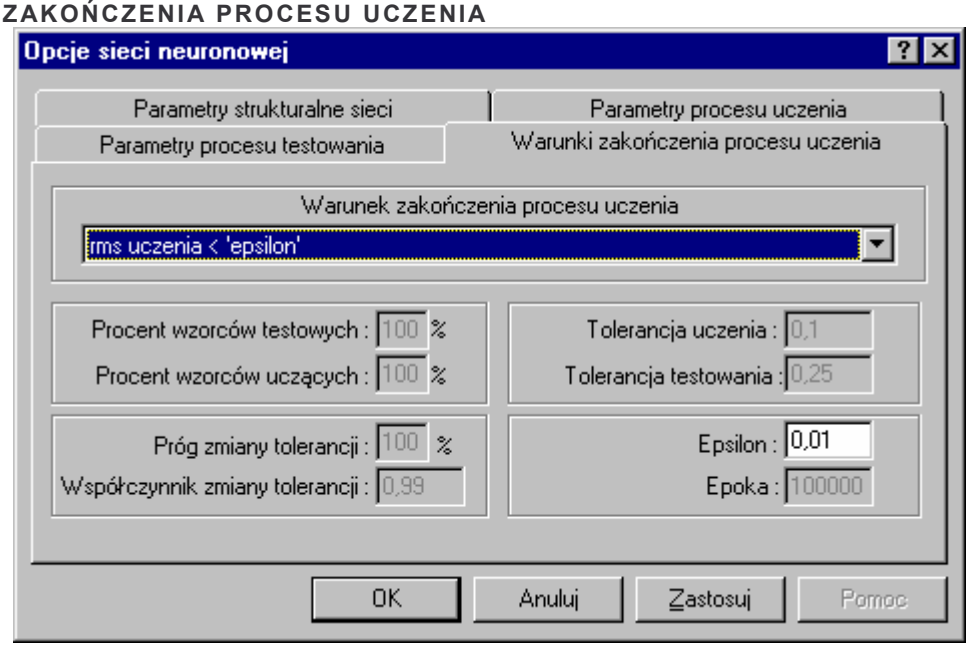


Rys. 22. Zakładka: Parametry procesu testowania

Częstość testowania

Parametr decydujący o częstości automatycznego testowania. Jego wartość określa ilość epok uczących po których nastąpi prezentacja wzorców testowych ze zbioru testowego. Jeżeli częstość testowania jest równa 0, to automatyczne testowanie nie będzie przeprowadzane. W przypadku wartości niezerowej wymagane jest aby w parametrze ‘plik testowy’ podana była ścieżka do pliku zawierającego wzorce testowe.

WARUNKI ZAKOŃCZENIA PROCESU UCZENIA



Rys. 23. Zakładka: Warunki zakończenia procesu uczenia

Warunek zakończenia procesu uczenia

Parametr ten określa warunek, jaki musi spełnić sieć neuronowa, aby został zakończony proces uczenia. W obecnej wersji użytkownik ma do dyspozycji 7 różnych możliwości:

'% wzorców uczących' mieści się w tolerancji

Jeżeli liczba wzorców uczących, które mieszczą się w tolerancji uczenia osiągnie pożądany próg procentowy, to uczenie zostanie przerwane. Próg procentowy określa się w parametrze '*%wzorców uczących*'.

'% wzorców testowych' mieści się w tolerancji

Jeżeli liczba wzorców testowych, które mieszczą się w tolerancji uczenia osiągnie pożądany próg procentowy, to uczenie zostanie przerwane. Próg procentowy określa się w parametrze '*%wzorców testowych*'.

epoka większa od 'epoka uczenia'

Jeżeli epoka uczenia osiągnie wartość progową, to uczenie zostanie przerwane. Próg liczbowy określa się w parametrze '*epoka uczenia*'.

RMS uczenia < 'epsilon'

Jeżeli błąd RMS uczenia spadnie poniżej pewnego progu liczbowego, to uczenie zostanie przerwane. Próg ten określa się w parametrze 'epsilon'.

RMS testowania < 'epsilon'

Jeżeli błąd RMS testowania spadnie poniżej pewnego progu liczbowego, to uczenie zostanie przerwane. Próg ten określa się w parametrze 'epsilon'.

tol. ucz. < 'epsilon'

Jeżeli aktualna wartość tolerancji uczenia spadnie poniżej pewnego progu liczbowego, to uczenie zostanie przerwane. Próg ten określa się w parametrze 'epsilon'.

tol. test. < 'epsilon'

Jeżeli aktualna wartość tolerancji testowania spadnie poniżej pewnego progu liczbowego, to uczenie zostanie przerwane. Próg ten określa się w parametrze
'epsilon'.

procent wzorców uczących

Parametr określa próg procentowy dla warunku zakończenia [**'%wzorców uczących' mieści się w tolerancji].**

procent wzorców testowych
Parametr określa próg procentowy dla warunku zakończenia [**'%wzorców testowych' mieści się w tolerancji].**

epoka uczenia
Parametr określa próg liczbowy dla warunku zakończenia [**epoka większa od 'epoka uczenia'].**

Epsilon
Parametr określający próg liczbowy dla następujących warunków zakończenia:
rms uczenia < 'epsilon'
rms testowania < 'epsilon'
tol. ucz. < 'epsilon'
tol. test. < 'epsilon'

Tolerancja uczenia
Parametr określający początkową wartość tolerancji uczenia. Idea tolerancji uczenia została wyjaśniona w rozdziale poświęconym uczeniu sieci neuronowej.

Tolerancja testowania
Parametr określający początkową wartość tolerancji testowania. Idea tolerancji testowania została wyjaśniona w rozdziale poświęconym uczeniu sieci neuronowej.

Próg zmiany tolerancji
Parametr określający procentową ilość wzorców uczących, które mieszczą się w granicach tolerancji. Jeżeli osiągnięty zostanie ten próg, to tolerancja uczenia zostanie pomnożona przez wartość parametru '*współczynnik zmiany tolerancji*'. Jeżeli parametr '*%próg zmiany tolerancji'* będzie równy 100 oznaczać to będzie, że jeżeli wszystkie wzorce uczące mieszczą się w tolerancji, to jej wartość ulegnie pomnożeniu przez współczynnik zmiany tolerancji.

Współczynnik zmiany tolerancji
Parametr decydujący o wielkości zmiany tolerancji. Jeżeli jego wartość będzie równa 1 to tolerancja nie będzie ulegała zmianie, niezależnie od wartości parametru '*%próg zmiany tolerancji*'. Dopuszczalna wartość parametru zawiera się w przedziale od 0 do 1.

UCZENIE SIECI - INFORMACJE TEORETYCZNE

Sieć neuronowa zdobywa wiedzę podczas procesu uczenia. System Neuronix pozwala na uruchomienie uczenia, śledzenie wskaźników dających wgląd w jej bieżący stan, sterowanie uczeniem w trybie pracy krokowej oraz na dodatkowe czynności związane z bezpieczeństwem, do których należą automatyczne zapisywanie wag w trakcie uczenia.

WSKAŹNIKI INFORMUJĄCE O BIEŻĄCYM STANIE SIECI

W obecnej wersji symulatora Neuronix, wykorzystuje się **trzy wskaźniki**, które pozwalają oszacować jakość wiedzy, jaką sieć zdobyła na etapie uczenia. Podstawowym miernikiem jest błąd średniokwadratowy RMS. Dodatkowe informacje takie jak tolerancja pozwalają stwierdzić, na ile pojedyncze wyjście sieci różni się od pożądanej dla niego wartości. Użytkownik ma możliwość śledzenia wartości tych wielkości, co pozwala mu na podjęcie decyzji o zakończeniu uczenia. Jeżeli stwierdza, że proces uczenia przebiega sprawnie, może ustawić dla wskaźników pewne wartości progowe przy których nastąpi natychmiastowe zakończenie tego procesu.

BŁĄD RMS

Podstawową miarą błędu stosowaną w systemie Neuronix jest błąd RMS. Podstawą do jego obliczenia stanowi błąd średniokwadratowy MSE. Oblicza się go jako sumę kwadratów różnic wartości rzeczywistej i oczekiwanej, podzieloną przez ilość wyrazów, które składają się na sumę. Ta miara błędu zyskała dużą popularność wśród metod statystycznych, gdyż w dużym stopniu upraszcza optymalizację, a przy tym jest intuicyjnie zrozumiały. Pewnym mankamentem miary jest fakt, że nie odzwierciedla ona błędu pojedynczego wyjścia sieci, lecz przede wszystkim pozwala na stwierdzenie średniej odchyłki w ramach całej epoki.

Błąd średniokwadratowy MSE dla jednej prezentacji wyraża się wzorem:

(3-1) $$MSE' = \frac{1}{n}\sum_{i=0}^{n-1}(T_i - O_i)$$

gdzie:
n - liczba wyjść sieci,

T_i - wartość oczekiwana i-tego wyjścia sieci,
O_i - wartość rzeczywista i-tego wyjścia sieci.

Zakładając, że w jednej epoce zaprezentowanych zostanie m - wzorców, to błąd MSE dla jednej epoki wynosi:

(3-2) $$MSE = \frac{1}{m}\sum_{j=0}^{m-1} MSE'_j$$

Mając tak określony błąd średniokwadratowy można zdefiniować błąd RMS jako pierwiastek kwadratowy z błędu RMS, zatem:

(3-3) $$RMS = \sqrt{MSE}$$

W trakcie uczenia należy oczekiwać, że wartość błędu RMS będzie malała w miarę kolejnych epok uczących, choć drobne wahania są dopuszczalne.

TOLERANCJA

Wskaźnik tolerancji służy do określenia dopuszczalnego błędu na pojedynczym wyjściu sieci. Wartość tolerancji zawiera się w przedziale od 0 do 1. Przedział ten podyktowany jest logistyczną funkcją przejścia, której wartości należą do tego samego przedziału.

Przykład 3-1

Sieć neuronowa ma 3 wyjścia. Tolerancja ustawiona została na 0,3.

Wartość prawidłowa wyjść jest:

0 1 1

Wartość rzeczywista wyjść jest:

0,32 0,71 0,99

Jak widać, pierwsze wyjście nie mieści się w granicach tolerancji, gdyż jego wartość odbiega od prawidłowego wyjścia o wartość większą niż dopuszczalna tolerancja. Wyjście drugie i trzecie zawierają się w dopuszczalnym przedziale.

Wzorzec uczący, którego chociaż jedno wyjście nie mieści się w granicach tolerancji uznawany zostaje jako nie mieszczący się w granicach tolerancji. Powoduje to zwiększenie licznika ilości wzorców będących poza tolerancją. W miarę rosnącego numeru epoki, ilość wzorców znajdujących się poza tolerancją powinna ulegać zmniejszeniu. Spowodowane to jest tym, iż w procesie dostosowywania wag uczestniczą tylko wzorce nie mieszczące się w granicach tolerancji. W ten sposób sieć modyfikuje swe wagi, aby zmniejszyć błąd na wyjściu nieprawidłowych wzorców. Szczególnie zalecaną praktyką jest ustawienie tolerancji uczenia na dużą wartość na wstępie i zmniejszaniu jej w miarę postępów w nauce. Takie podejście gwarantuje, że modyfikacje wag sieci nie będą tak drastyczne jak w przypadku modyfikacji wag w odpowiedzi na każdy prezentowany wzorzec. To z kolei zapobiega gwałtownym zwrotom wokół prawidłowego rozwiązania. Sieć powoli przyswaja sobie wiedzę, zmniejszając w ten sposób ryzyko zapominania istotnych informacji. Jak wykazały doświadczenia, dostrajanie tolerancji w trakcie uczenia może doprowadzić w dłuższym horyzoncie czasowym do osiągnięcia mniejszego błędu RMS.

ILOŚĆ WZORCÓW POZA TOLERANCJĄ

Wskaźnik ten pokazuje ilość wzorców, które należą do grupy wzorców nie mieszczących się w bieżącej wartości tolerancji dla wyjść sieci. Wartość tego wskaźnika powinna zmniejszać się w miarę wzrostu epoki uczenia. Jeśli ilość takich wzorców osiągnie wartość 0 to można powiedzieć, że sieć neuronowa odpowiada na każdy wzorzec uczący w granicach zadanej tolerancji.

Warunki niezbędne do uruchomienia uczenia. Przed uruchomieniem uczenia należy przygotować:
- plik uczący przy użyciu arkusza danych lub dowolnego edytora tekstowego,
- plik testowy jeżeli częstość testowania jest większa od 0.

Ponadto konieczne jest:
- wskazanie pliku wag, do którego będą zapisywane aktualne wagi sieci,

- ustawienie wartości parametrów, jeżeli mają być inne niż domyślne.

MONITORING PROCESU UCZENIA I TESTOWANIA

Okno dialogowe **Monitoring uczenia i testowania** służy do kontrolowania procesu uczenia i testowania sieci neuronowej. Użytkownik ma możliwość śledzenia wskaźników sieci i decydowania o dodatkowych działaniach podejmowanych przez system Neuronix do których należą: zapisywanie najlepszego ustawienia sieci oraz automatyczny zapis wag.

Rys. 24. Monitoring uczenia i testowania

W górnej części okna uczenia znajdują się trzy podstawowe informacje.

Epoka
Informuje o aktualnej liczbie epok uczących jakie upłynęły od rozpoczęcia uczenia

Iteracja
Informuje o aktualnej liczbie iteracji (prezentacji wzorca uczącego) jakie zostały wykonane w ramach aktualnej epoki.

Stan sieci
Informuje o rodzaju podejmowanego działania. „NAUKA" oznacza proces prezentacji wzorców uczących a „TEST" oznacza proces prezentacji wzorców testowych.

Środkową część okna wypełniają opisane wcześniej wskaźniki stanu sieci:

Błąd RMS
Uaktualnia się tylko po zmianie epoki, niezależnie od rozmiaru cyklu uczącego.

Tolerancja
Informuje o bieżącej wartości tolerancji.

Poza tolerancją
Informuje o ilości wzorców, dla których wyjścia sieci nie mieszczą się w granicach tolerancji. Dolna część okna zawiera przyciski, do których przypisane zostały następujące akcje:

Start
Uruchamia proces uczenia, który trwa tak długo, aż zostanie spełniony warunek zakończenia, który ustawia się w oknie parametrów sieci. Jeżeli użytkownik zdecyduje się na wcześniejsze przerwanie nauki to może to nacisnąć przycisk „*Stop*"

Stop
Zatrzymuje uruchomiony proces uczenia lub testowania.

Koniec
Zamyka okno uczenia wraz z automatycznym zapisem wag do pliku, jaki został wskazany w oknie parametrów sieci.

Inicjalizacja
Inicjalizacja struktur sieci losowymi wartościami z przedziału określonym parametrem '*maksymalna wartość wag*'.

Test

Przeprowadza prezentację wzorców testowych, niezależnie od parametru '*częstość testowania*'. Warunkiem wykonania tego polecenia jest wskazanie pliku testowego w parametrze '*plik testowy*'. Po wykonaniu wszystkich prezentacji obliczany jest błąd RMS testowania wraz z ustawieniem jego wartości w polu 'Błąd RMS' okna uczenia.

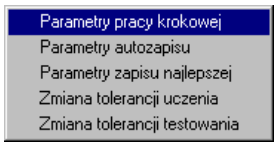
Zapis wag

Zapis wag sieci neuronowej do pliku wag, który został wskazany w parametrze '*plik wag*' w oknie parametrów sieci neuronowej.

Odczyt wag

Odczyt wag sieci neuronowej z pliku wag, który został wskazany w parametrze '*plik wag*' w oknie parametrów sieci neuronowej.

Naciśnięcie prawego klawisza myszki na oknie uczenia powoduje otwarcie się podręcznego menu:



Rys. 25. Podręczne menu okna uczenia

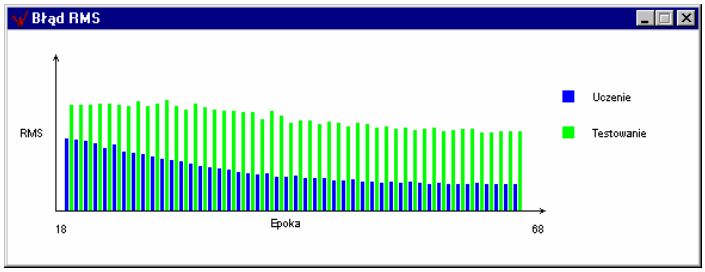
Prawa część okna służy do włączania dodatkowych opcji:

Prezentacja statystyki

Jeżeli opcja jest włączona, to okno uczenia odświeża swoje pola w trakcie procesu uczenia. Użytkownik może obserwować wartości poszczególnych wskaźników co powoduje jednak spowolnienie obliczeń. Jeżeli opcja jest wyłączona obliczenia przebiegają szybciej.

Prezentacja wykresu RMS

Jeżeli opcja jest włączona następuje prezentacja błędu RMS w postaci wykresu słupkowego. Ma on charakter czysto poglądowy, dający przejrzysty obraz błędu RMS uczenia na tle błędu RMS testowania. Wyłączenie tej opcji kilkakrotnie przyspiesza obliczenia.



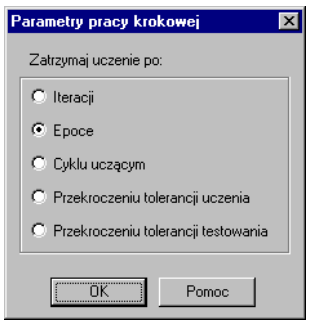
Rys. 26. Prezentacja błędu RMS

Wizualizacja struktury sieci neuronowej

Opcja pozwala na graficzne przedstawienie struktury sieci neuronowej, ma to duże znaczenie w przypadku dydaktyki i jest pomocne na etapie zapoznawania się z teoretycznymi i praktycznymi aspektami sieci neuronowych.

Praca krokowa

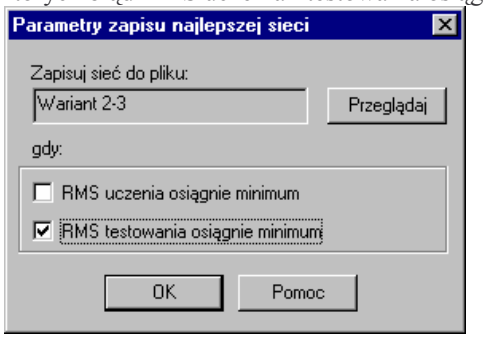
Praca krokowa umożliwia zatrzymywanie procesu uczenia lub testowania po spełnieniu określonych warunków. Opcja ta jest przydatna, gdy użytkownik chce dostrzec zmiany, jakie zachodzą we wskaźnikach w ramach pewnych odcinków czasowych. Dodatkowo praca krokowa może być ustawiona na wychwytywanie wzorców, dla których wyjścia sieci nie mieszczą się w granicach tolerancji. Wraz z funkcjami arkusza pozwala to na wychwycenie dodatkowych informacji o kłopotliwym wzorcu lub uruchomienie sieci i sprawdzenie jej wyników. Parametry pracy krokowej dostępne są na prawym przycisku myszy, której kursor musi znaleźć się w obrębie okna uczenia.



Rys. 27. Parametry pracy krokowej

Zapis najlepszej

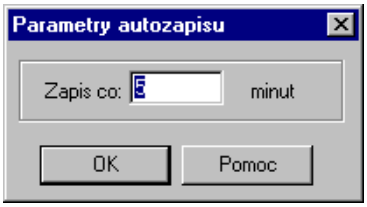
W trakcie uczenia można również ustawić opcję automatycznego zapisywania najlepszej sieci. Wskaźnikiem jakości sieci jest tu wartość błędu RMS uczenia i testowania. Jeżeli któryś z nich osiągnie wartość minimalną to wagi sieci zapisywane są do pliku najlepszych sieci, który wskazywany jest przed rozpoczęciem uczenia. Trzeba liczyć się z faktem, że włączenie tej opcji zwalnia cały proces obliczeniowy, lecz zyskuje się pewność, że w pliku wag najlepszej sieci zapisane są wagi, dla których błąd RMS uczenia i testowania osiągnął najmniejszą wartość.



Rys. 28. Parametry zapisu najlepszej sieci

Autozapis

Autozapis jest opcją, której włączenie powoduje, że wagi sieci zapisywane są do pliku wag, co pewien określony czas. Dzięki temu użytkownik ma pewność, że zanik zasilania lub pojawienie się błędu systemowego nie spowoduje utraty wielu godzin pracy.



Rys. 29. Parametry autozapisu

UCZENIE SIECI

Program Neuronix pozwala na bardzo elastyczne posługiwanie się wyuczoną siecią neuronową. Wyuczona sieć neuronowa może być uruchamiana na trzy różne sposoby:

- z poziomu symulatora Neuronix,
- z poziomu programu napisanego w języku reprezentacji wiedzy Sphinx i uruchamianego w jednym z programów pakietu Sphinx, co sprawia, że możliwe jest współdziałanie systemu ekspertowego i sieci neuronowej,
- z poziomu programu wykonywalnego, który może korzystać z bibliotek dynamicznych dll.

W poniższych punktach posłużono się zbiorem uczącym, który jest wykorzystywany w projekcie **../Projekt/Sinus/sinus.npr.**

We	we	We	wy	wy	wy	-	-
y-3	y-2	y-1	y	#ćwiartka	>moduł	x	numer
0	0,587785	0,951057	0,951057	druga	cDuży	1,884956	jeden
0,587785	0,951057	0,951057	0,587785	druga	bŚredni	2,513274	dwa
0,951057	0,951057	0,587785	1,23E-16	druga	aMały	3,141593	trzy
0,951057	0,587785	1,23E-16	-0,58779	trzecia	aMały	3,769911	cztery
0,587785	1,23E-16	-0,58779	-0,95106	trzecia	cDuży	4,39823	pięć
1,23E-16	-0,58779	-0,95106	-0,95106	czwarta	cDuży	5,026548	sześć
-0,58779	-0,95106	-0,95106	-0,58779	czwarta	BŚredni	5,654867	siedem
-0,95106	-0,95106	-0,58779	-2,5E-16	czwarta	AMały	6,283185	osiem

URUCHAMIANIE SIECI Z POZIOMU SYSTEMU NEURONIX

W celu uruchomienia sieci neuronowej z poziomu Neuronix'a można skorzystać z opcji menu głównego **Sieć | Uruchomienie sieci**. Dla omawianego przykładu spowoduje to otworzenie arkusza danych wraz z wypełnionym nagłówkiem, w którym znajdą się tylko kolumny wejściowe i wyjściowe:

	A	B	C	D	E	F
Typ	we	we	we	wy	wy	wy
Nazwa	t - 2	t - 1	t	t + 1	#ćwiartka	>moduł
3						
4						
5						

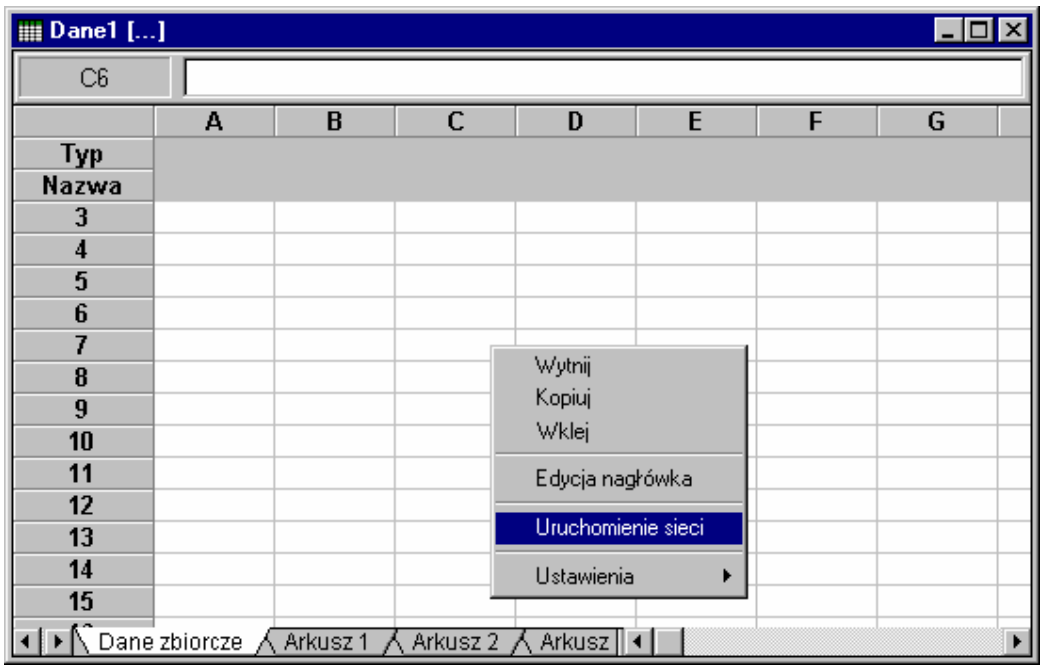
Rys. 30. Pusty arkusz danych po użyciu opcji **Uruchamianie sieci**

W ten sposób użytkownik zostaje uwolniony od własnoręcznego przygotowania nagłówka arkusza danych, gdyż zostaje on skonstruowany w oparciu o plik uczący. W celu uruchomienia sieć dla konkretnych wartości wejść należy najpierw wypełnić komórki należące do kolumn wejściowych:

	A	B	C	D	E	F	G
Typ	we	we	we	wy	wy	wy	-
Nazwa	t - 2	t - 1	t	t + 1	#ćwiartka	>moduł	t + 1
3	0,5875275	0,9508595	0,9508595				
4	0,9508595	0,9508595	=D3				
5	0,9508595	0,5888156	0,0015927				

Rys. 31. Wartości wejściowe w arkuszu uruchamiania sieci

Następnie należy wybrać opcję „Uruchomienie sieci" z podręcznego menu uruchamianego prawym klawiszem myszki. Działanie to spowoduje podanie sieci neuronowej wartości wejściowych z arkusza, pobranie wartości wyjściowych z sieci i przedstawienie ich w komórkach kolumn wyjściowych.



Rys. 32. Opcja uruchamiania sieci w podręcznym menu

	A	B	C	D	E	F
Typ	we	we	we	wy	wy	wy
Nazwa	t - 2	t - 1	t	t + 1	#ćwiartka	>moduł
3	0	0,5875275	0,9508595	0,9166875	druga	cDuży
4	0,5875275	0,9508595	0,9508595	0,5965648	druga	bŚredni
5	0,9508595	0,9508595	0,5888156	0,0145102	druga	aMały

Rys. 33. Arkusz uruchamiania sieci wypełniony odpowiedziami sieci

Opcja podręcznego menu „Uruchomienie sieci” nie działa w trakcie uczenia. Należy zatrzymać uczenie klawiszem „Stop”, a następnie przeprowadzić uruchomienie. Opcja podręcznego menu „Uruchomienie sieci” jest dostępna w każdym arkuszu danych. Dopuszczalne jest odczytanie pliku uczącego do arkusza danych i uruchomienie sieci. Spowoduje to wypełnienie kolumn wyjściowych. Jeżeli użytkownik pragnie porównać wartości wyjściowe w pliku uczącym z wartościami wyjściowymi sieci, to może skopiować kolumny wyjściowe w inne miejsce arkusza i usunąć z nich typ kolumny. W ten sposób zostaną zachowane prawidłowe wartości, które mogą być porównywane z odpowiedzią sieci.

	A	B	C	D	E	F	G	H	I
Typ	we	we	we	wy	wy	wy	-	-	-
Nazwa	t - 2	t - 1	t	t + 1	#ćwiartka	>moduł	t + 1	ćwiartka	moduł
3	0	0,5875275	0,9508595	1,0110598	druga	cDuży	0,9508595	druga	cDuży
4	0,5875275	0,9508595	0,9508595	0,5563987	druga	bŚredni	0,5888156	druga	bŚredni
5	0,9508595	0,9508595	0,5888156	0,0195371	druga	aMały	0,0015927	druga	aMały

Rys. 34. Arkusz uruchamiania sieci (porównanie odpowiedzi sieci z wartościami oczekiwanymi)

Kolejność kolumn w arkuszu danych jest dowolna. Wpisywanie wartości wyjściowych sieci odbywa się od pierwszego wiersza do ostatniego. Daje to możliwość cyklicznego uruchomienia sieci tzn. wykorzystanie wartości wyjściowych sieci jako wartości wejściowych w następnym wierszu. Uzyskuje się to poprzez zastosowanie odwołania do innej komórki arkusza.

4

Czym właściwie jest sztuczna sieć neuronowa ?

W rozdziale:

- Historia
- Budowa sztucznego neuronu
- Jak to rozwiązała natura ?
- Wybrane rodzaje sieci neuronowych
- Jak działa sieć neuronowa ?
- W czym sieci neuronowe są mistrzami ?
- Dlaczego sieć uczy się ?

Cenne informacje

Ćwiczenia z komputerem

Koniecznie przeczytaj

HISTORIA

Już od czasów starożytnych człowiek starał się zgłębić tajemnice własnego mózgu. Dzięki rozwojowi cywilizacyjnemu w początkach lat 40-tych naszego wieku stało się możliwe modelowanie pewnych zachowań tego tajemniczego tworu. W roku 1943 **McCulloch** i **Pitts** opracowali matematyczny model sztucznego neuronu, następstwem tego stały się próby struktur składających się z kilku takich neuronów. Podobnie jak w neuronie biologicznym (rys. 36) dochodziło do niego wiele połączeń i wychodziło jedno, które uaktywniało się po przekroczeniu pewnej wartości progowej. Zespół takich komórek, pomimo ich małej złożoności, potrafi być potężnym i jak dotychczas nie do końca zrozumiałym organem. Szacuje się, że w mózgu ludzkim występuje około 10^{11} neuronów. Sprawność mózgu determinowana jest przede wszystkim liczbą neuronów i wzajemnych połączeń międzyneuronowych, a nie szybkością przetwarzania informacji w neuronach. Sieci neuronowe są więc przedstawicielami równoległych metod przetwarzania informacji. Kolejnym krokiem na drodze rozwoju sieci neuronowych było wyjaśnienie w roku 1949 przez **Hebb'a** mechanizmu pamiętania informacji w biologicznych sieciach neuronowych. Pierwsza sztuczna sieć neuronowa nazwana perceptronem, powstała w roku 1958 i została wykonana przez **Rosenblatt'a**. Wykonana w technice mieszanej, korzystała poza elementami elektromechanicznymi również z elementów elektronicznych. Sieć służyła do rozpoznawania znaków alfanumerycznych. Metodą programowania sieci było uczenie się na podstawie prezentowanych wzorców. Ponadto **Rosenblatt** udowodnił tzw. twierdzenie o zbieżności perceptronu, gwarantujące zbieżność algorytmu procesu uczenia (skończona liczba iteracji algorytmu), dla zbioru wag, które pozwalają uzyskać prawidłową odpowiedź sieci dla każdego wzorca ze zbioru uczącego. Wadą perceptronu jest możliwość rozwiązywania jedynie problemów klasyfikacji separowalnych liniowo w warstwie wyjściowej. Perceptron wydawał się wówczas receptą na opracowanie sztucznego modelu ludzkiego mózgu, a stąd byłby już tylko krok do stworzenia układu przejawiającego cechy ludzkiej inteligencji. Tragiczna śmierć **Rosenblatt'a** (1971) oraz opublikowanie książki **Minsky'ego** i **Papert'a** (1969) zatytułowanej "Perceptrons", w której udowodniono liczne ograniczenia sieci jednowarstwowej (perceptronu), spowodowały zaprzestanie poważniejszych badań naukowych w zakresie sieci neuronowych aż do połowy lata osiemdziesiątych. Obecnie sieci neuronowe realizowane są jako algorytmy implementowane najczęściej w językach wysokiego poziomu (np. C , C++, Pascal), lub sprzętowo jako elektroniczne układy VLSI. Cechą charakterystyczną sieci neuronowej jest odporność na przypadkowe uszkodzenia np. wyłączenie na drodze programowej neuronu lub odpowiadające temu mechaniczne przerwanie połączenia pomiędzy tranzystorami w układzie elektronicznym lub optycznym symulującym neuron, nie powoduje całkowitego uszkodzenia układu. Sieć w takich wypadkach potrafi się adaptować i działać nadal poprawnie. Jest to pewna analogia do systemów biologicznych, w których funkcje uszkodzonego organu w wielu przypadkach przejmują częściowo inne organy.

BUDOWA SZTUCZNEGO NEURONU

Obecne badania naukowe nastawione są nie tyle na modelowanie funkcji mózgu, ile przede wszystkim starają się wykorzystać unikatowe cechy sieci neuronowych w takich zagadnieniach jak sterowanie, diagnostyka, rozpoznawanie obrazów i prognozowanie. Ze względów dydaktycznych dobrym modelem neuronu jest model McCullocha i Pittsa, gdyż charakteryzuje się prostotą i przejrzystością. Na jego przykładzie omówione zostaną podstawowe elementy budowy sztucznego neuronu, które będą odgrywały istotne znaczenie przy późniejszej budowie sieci. Jak widać na rys. 35 sztuczny neuron składa się z następujących elementów:

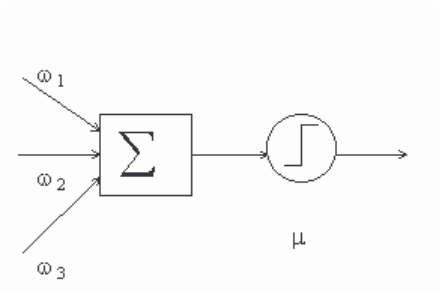
- połączeń wejściowych, z którymi skojarzona jest pewna liczba zwana **wagą połączenia**,
- elementu sumującego pobudzenie pochodzące od połączeń wejściowych,
- elementu progowego, który daje na wyjściu stan wysoki lub niski w zależności od wartości sumy.

Element progowy jest szczególnym rodzajem **funkcji przejścia,** zwanej także **funkcją aktywacji**, której argumentem jest sumaryczne pobudzenie, a na wyjściu podaje wartości z określonego przedziału (najczęściej od 0 do 1). Najczęściej spotykaną funkcją aktywacji jest **funkcja logistyczna** określona wzorem:

(4-1) $$f(x) = \frac{1}{1+e^{-z}},$$

gdzie:

(4-2) $$z = \sum_i x_i \omega_i .$$



Rys. 35. Schemat sztucznego neuronu

Przykład 4-1:

Załóżmy, że neuron posiada 3 wejścia:
$x_1 = 0.5$,
$x_2 = 0.3$,
$x_3 = 0.1$.
Odpowiadają im połączenia wejściowe, których wagi mają wartości:
w1 = 0.01,
w2 = 0.2,
w3 = 0.9

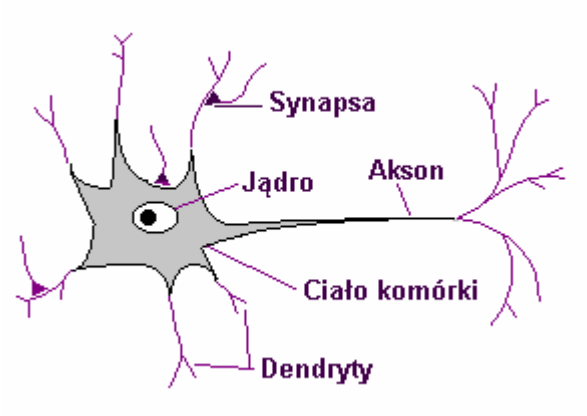
Dla podanych wyżej wartości, sumaryczne pobudzenie wynosi $z = 0.5 \cdot 0.01 + 0.3 \cdot 0.2 + 0.1 \cdot 0.9 = 0.155$,
a wartość aktywacji dla funkcji logistycznej:

$$f(z) = \frac{1}{1 + e^{-0.155}} \approx 0.5387$$

Wartość wyjścia neuronu wynosi więc w przybliżeniu 0.5387.

JAK TO ROZWIĄZAŁA NATURA ?

W organizmie człowieka znajduje się około 10 miliardów **komórek nerwowych**, czyli **neuronów**. Większość z nich skupiona jest w ośrodkowym układzie nerwowym, pozostałe tworzą zwoje nerwowe nerwów czaszkowych, nerwów rdzeniowych oraz zwoje układu autonomicznego. Niezależnie od lokalizacji neurony charakteryzują się podobną budową. Komórka składa się z: ciała komórki, wypustek krótkich (**dendrytów)** oraz **aksonu**. Bodziec działając na błonę komórkową neuronu zmienia jej właściwości, co wywołuje potencjał czynnościowy. Do wnętrza neuronu przemieszczają się **jony Na**. Następuje wyrównanie ładunków elektrycznych pomiędzy wnętrzem i otoczeniem. Jest to tzw. depolaryzacja błony komórkowej. Depolaryzacja szybko przemieszcza się wzdłuż błony, w tym również aksonów. W ten sposób powstaje **impuls nerwowy** transportowany do **synapsy**. Błonę komórkową neuronu przekazującego impuls nazywamy błoną presynaptyczną, a odbierającego – błoną postsynaptyczną. W zakończeniach **aksonów**, w obrębie synaps, wydzielane są **przekaźniki chemiczne** tzw. transmittery, które zmieniają właściwości błony postsynaptycznej, w której powstaje postsynaptyczny potencjał, odpowiednio, zależnie od rodzaju przekaźnika, o charakterze pobudzającym lub hamującym. Synapsy pokrywają około 40% powierzchni ciała neuronu i dendrytów. Pomiędzy błoną presynaptyczną i postsynaptyczną znajduje się przestrzeń synaptyczna. W obrębie synapsy transmittery zlokalizowane są w pęcherzykach synaptycznych. W przestrzeni synaptycznej przekaźniki uwalniane są z pęcherzyków, a następnie łączą się z receptorami w błonie postsynaptycznej, powodując w niej zmiany potencjału i tym samym impuls rozprzestrzenia się na następny neuron. Przykładem przekaźnika pobudzającego jest m.in.: acetylocholina, dopamina, adrenalina, noradrenalina, serotonina, histamina. Przekaźnikiem hamującym może być przykładowo: glicyna, kwas gamma-aminomasłowy (GABA).



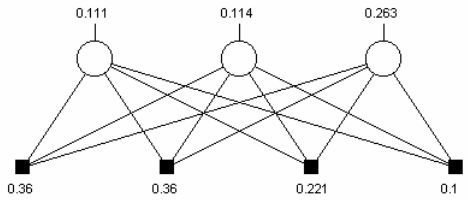
Rys. 36. Neuron biologiczny

WYBRANE RODZAJE SIECI NEURONOWYCH

Neurony łączy się zazwyczaj w struktury zwane potocznie sieciami neuronowymi. Dla pewnego uproszczenia przyjęło się gromadzić neurony w tak zwane warstwy. Ze względu na ilość warstw rozróżnia się sieci jedno i wielowarstwowe. Połączenia pomiędzy kolejnymi warstwami mogą być pełne tzn., że każdy neuron z warstwy wyższej połączony jest z każdym neuronem warstwy niższej. Warstwę, która przyjmuje bezpośrednie wartości wejść, nazywa się potocznie **warstwą wejściową**. Z **warstwy wyjściowej** odczytuje się odpowiedź sieci na wzorzec podany na jej wejście. W sieciach wielowarstwowych istnieją **warstwy ukryte**, które leżą pomiędzy warstwą wejściową i wyjściową. Warstwa wejściowa nie przetwarza informacji podanej na wejście sieci przesyłając ją do warstw ukrytych.

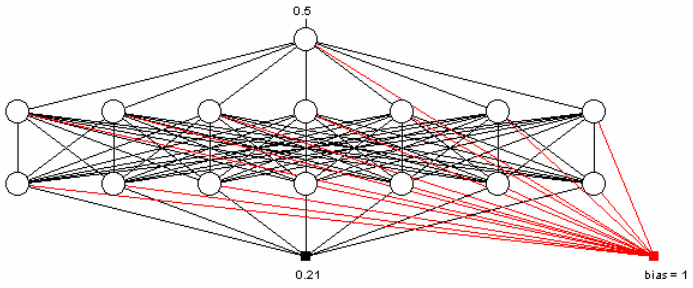
SIECI JEDNOWARSTWOWE JEDNOKIERUNKOWE

Sieć jednowarstwowa jest najprostszą sztuczną siecią neuronową. Składa się w warstwy wejściowej nie przetwarzającej sygnałów, oraz warstwy wyjściowej połączonej bezpośrednio z warstwą wejściową.



Rys. 37. Sieć jednowarstwowa

SIEĆ WIELOWARSTWOWA JEDNOKIERUNKOWA



Rys. 38. Sieć wielowarstwowa.

Sieć wielowarstwowa posiada dodatkowe warstwy ukryte, które pozwalają zwiększyć jej możliwości rozwiązywania problemów o zagadnienia klasyfikacji nie separowalne liniowo, przez co rozumie się zdolność do oddzielenia dwóch zbiorów przy pomocy linii innej niż prosta. Przyjmuje się, że jedna warstwa ukryta jest wystarczająca dla rozwiązania większości stawianych przed siecią zadań.

W CZYM SIECI NEURONOWE SĄ MISTRZAMI ?

Sieci neuronowe wykazują szereg unikatowych własności, dzięki którym budzą one żywe zainteresowanie szerokich kręgów badaczy z wielu dziedzin życia. Podstawową własnością sieci neuronowych jest ich **zdolność do przyswajania wiedzy** na podstawie **prezentowanych przykładów**. Nie trzeba precyzować sposobu rozwiązywania danego problemu, lecz wystarczy zgromadzić odpowiednio duży i reprezentatywny zbiór próbek. Sieci neuronowe potrafią w naturalny sposób przetworzyć podaną w ten sposób informację i wyciągnąć z niej najistotniejsze elementy. Szczególnie ważne jest to w problemach klasyfikacyjnych oraz wszelkiego rodzaju prognozach. Łączy się to w naturalny sposób z kolejną ważną własnością, jaką jest **zdolność do generalizacji,** czyli **uogólniania**. Sprowadza się to do dawania sensownej odpowiedzi w sytuacjach, z którymi sieć nigdy się wcześniej nie zetknęła. Jako przykład można podać możliwość rozróżniania liter, których czcionka nie występowała w zbiorze uczącym. Sieć neuronowa potrafi poradzić sobie z danymi niekompletnymi i zakłóconymi. Ponadto w realizacjach sprzętowych charakteryzuje się dużą odpornością na uszkodzenia oraz bardzo dużą szybkością działania, co kwalifikuje ją do zastosowań w **systemach czasu rzeczywistego**. Z racji swoich własności sieci neuronowe nazywane są czasami metodą „superregresji". Ich zdolność do aproksymacji jest tym większa, że w ich naturze tkwi **silna nieliniowość**. Jakość ich aproksymacji jest porównywalna z jakością aproksymacji najlepszych metod regresji nieliniowej. Odpowiednio zastosowana technika obliczeń neuronowych zwykle okazuje się być lepsza od metod bazujących na metodzie regresji liniowej. Różnica ta jest szczególnie widoczna przy dużej liczbie parametrów wejściowych i wyjściowych. Sieci neuronowe bardzo dobrze potrafią analizować wszystkie problemy nieliniowe, a więc takie z którymi najczęściej (prawie wyłącznie) można spotkać się w rzeczywistych warunkach.

5

Dane w procesie uczenia i uruchamiania sieci

W rozdziale:

- Plik uczący i testowy
- Funkcja przejścia (aktywacji)
- Arkusze kalkulacyjne

Cenne informacje

Ćwiczenia z komputerem

Koniecznie przeczytaj

PLIK UCZĄCY I TESTOWY

Każda sieć neuronowa potrzebuje **zbioru uczącego** na etapie pozyskiwania wiedzy. W przypadku sieci wielowarstwowej uczonej **metodą wstecznej propagacji błędu** konieczne jest zgromadzenie danych, w których zawarte będą pewne parametry wejściowe oraz odpowiadające im wartości wyjściowe. Użytkownik może taki zbiór przygotować w dowolnym edytorze tekstowym lub skorzystać z rozwiązania zaproponowanego w programie Neuronix. Oprócz pliku zawierającego dane uczące, można również posługiwać się tak zwanym zbiorem testowym, który zawiera te same funkcjonalne parametry wejściowe i wyjściowe, przy czym ich wartości nie powinny pokrywać się ze zbiorem uczącym. Jak sama nazwa wskazuje plik testowy służy do testowania sieci, a zatem do kontrolowania poprawności jej działania. Sieć neuronowa może na etapie uczenia osiągnąć zbyt dużą dokładność odwzorowywania danych znajdujących się w pliku uczącym. Traci w ten sposób zdolność do uogólniania, czyli jej zdolność generalizacji jest mała. Najczęściej dąży się do rozwiązania pośredniego, które daje w wyniku sieć neuronową, której odpowiedzi są sensowne dla wartości wejść, które nigdy wcześniej nie były prezentowane. Idea pliku uczącego i testowego została przedstawiona na poniższym przykładzie.

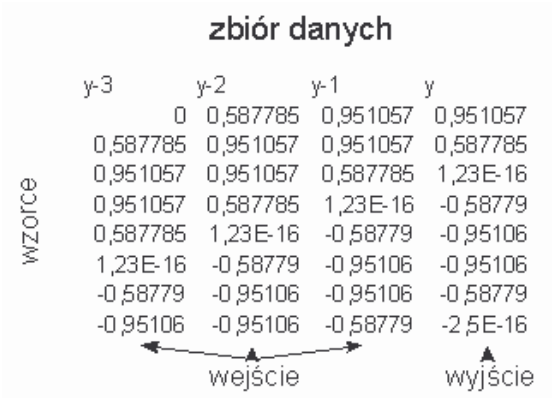
Funkcja sinus

Omawiany przykładowy projekt znajduje się w katalogu **../Projekt/Sinus/sinus.npr**. Celem sieci neuronowej będzie nauczenie się przebiegu funkcji sinus w przedziale od 0 do 2π. Jeżeli przedział ten podzielony zostanie na 10 równych odcinków to otrzyma się następujące wartości funkcji sinus.

X	sin(x)
0	0
0,628319	0,587785
1,256637	0,951057
1,884956	0,951057
2,513274	0,587785
3,141593	1,23E-16
3,769911	-0,58779
4,39823	-0,95106
5,026548	-0,95106
5,654867	-0,58779
6,283185	-2,5E-16

Rys. 39. Wartości funkcji sinus

Można założyć, że zbiór uczący będzie konstruowany w oparciu o cztery wartości funkcji y = sin(x). Pierwsze trzy będą stanowiły wartości wejściowe, a ostatnia czwarta będzie odpowiadającą im wartością wyjściową. W ten sposób tworzy się zbiór danych, w których składowe wejściowe oznaczone będą jako: y-1, y-2, y-3, a wartość wyjściowa oznaczona zostaje jako y. Zilustrowano to na rys. 40.



Rys. 40. Zbiór danych i jego poszczególne elementy

Utworzonych zostało osiem wzorców, które mogą posłużyć do budowy zbioru uczącego i ewentualnie testowego. Przykładowy zbiór uczący mógłby składać się ze wzorców od numeru 1 do numeru 6. Natomiast zbiór testowy powinien być rozłączny w stosunku do uczącego, a zatem będzie zawierał wzorce numer 7 i 8. Jak widać, zarówno zbiór uczący jak i testowy ma taką samą budowę, chociaż zawierają inne wartości. Pliki uczące i testowe w

symulatorze Neuronix mają format tekstowy. Poszczególne kolumny tych zbiorów odpowiadają wejściom lub wyjściom sieci. Każdy wiersz zawiera wartości dla konkretnego wejścia lub wyjścia sieci.
Symulator wymaga, aby każda kolumna, która jest wejściem była oznaczona symbolem „**we**”, natomiast każda kolumna reprezentująca wyjście powinna być oznaczona symbolem „**wy**”. Drugi wiersz zawiera nazwy wejść. Kolejne wartości w wierszu powinny być oddzielone **tabulatorem**. Dopuszczalne są również kolumny z dodatkowymi opisami wierszy, które nie będą bezpośrednio wykorzystywane jako wejście lub wyjście. W takim przypadku kolumna powinna być oznaczona znakiem „**-**”. Ostatecznie plik uczący **sinus.lrn** mógłby wyglądać następująco:

we	we	we	wy	-	-
y-3	y-2	y-1	y	x	numer
0	0,587785	0,951057	0,951057	1,884956	jeden
0,587785	0,951057	0,951057	0,587785	2,513274	dwa
0,951057	0,951057	0,587785	1,23E-16	3,141593	trzy
0,951057	0,587785	1,23E-16	-0,58779	3,769911	cztery
0,587785	1,23E-16	-0,58779	-0,95106	4,39823	pięć
1,23E-16	-0,58779	-0,95106	-0,95106	5,026548	sześć

Oprócz kolumn wejściowych i jednej kolumny wyjściowej plik uczący zawiera dwie kolumny opisowe, które nie będą bezpośrednio wykorzystywane w procesie uczenia sieci neuronowej. Plik testowy **sinus.tst** zawiera:

we	we	we	wy	-	-
y-3	y-2	y-1	y	x	numer
-0,58779	-0,95106	-0,95106	-0,58779	5,654867	siedem
-0,95106	-0,95106	-0,58779	-2,5E-16	6,283185	osiem

Symulator Neuronix umożliwia operowanie na następujących wartościach:
- liczbowych rzeczywistych,
- symbolicznych wyliczeniowych,
- symbolicznych hierarchicznych.

Wartości liczbowe podwójnej precyzji mogą należeć do przedziału od $1.7*10^{-308}$ do $1.7*10^{+308}$. Ich rozpiętość to przedział od $-1.7*10^{+308}$ do $1.7*10^{+308}$. **Wartości symboliczne** zawierać mogą dowolny ciąg znaków, nie dłuższy niż **32** pozycje. Wartości **symboliczne wyliczeniowe** powinny reprezentować pewne kategorie, pomiędzy którymi nie zachodzi żadna relacja większości. Przykładem mogą być nazwy miesięcy w roku lub płeć. Jeżeli kolumna pliku uczącego zawiera wartości symboliczne wyliczeniowe, to jej nazwa powinna rozpoczynać się od znaku specjalnego „#”. Wartości **symboliczne hierarchiczne** reprezentują kategorie, pomiędzy którymi zachodzi relacja większości. Jako przykład można podać określenie wzrostu: wysoki, średni, niski. Jeżeli kolumna pliku uczącego zawiera wartości symboliczne hierarchiczne, to jej nazwa powinna rozpoczynać się od znaku specjalnego „>”. Aby była zachowana relacja większości należy w odpowiedni sposób dobierać nazwy symboliczne. Przyjęto, że symbol, którego wartość w hierarchii jest najmniejsza jest również pierwszym alfabetycznie symbolem w całym zbiorze dostępnych wartości. Przykładowo, aby zachować relację większości w zbiorze: { wysoki, średni, niski }, należałoby przekształcić go w zbiór wartości { aNiski, bŚredni, cWysoki }. W ten sposób zachowuje się zrozumiałość poszczególnych symboli wraz z prawidłowym ułożeniem alfabetycznym. Jeżeli plik uczący zostanie rozszerzony o dodatkowe dwie symboliczne kolumny wyjściowe jego zawartość będzie następująca:

we	we	we	wy	wy	wy	-	-
y-3	y-2	y-1	y	#ćwiartka	>moduł	x	numer
0	0,587785	0,951057	0,951057	druga	cDuży	1,884956	jeden
0,587785	0,951057	0,951057	0,587785	druga	bŚredni	2,513274	dwa
0,951057	0,951057	0,587785	1,23E-16	druga	aMały	3,141593	trzy
0,951057	0,587785	1,23E-16	-0,58779	trzecia	aMały	3,769911	cztery
0,587785	1,23E-16	-0,58779	-0,95106	trzecia	cDuży	4,39823	pięć
1,23E-16	-0,58779	-0,95106	-0,95106	czwarta	cDuży	5,026548	sześć
-0,58779	-0,95106	-0,95106	-0,58779	czwarta	bŚredni	5,654867	siedem
-0,95106	-0,95106	-0,58779	-2,5E-16	czwarta	aMały	6,283185	osiem

Plik uczący zawiera trzy kolumny wejściowe o wartościach rzeczywistych, jedną kolumnę wyjściową o wartościach rzeczywistych, jedną kolumnę wyjściową o wartościach symbolicznych wyliczeniowych oraz jedną

kolumnę wyjściową o wartościach symbolicznych hierarchicznych. Ponadto plik zawiera dwie kolumny z dodatkowym opisem wierszy, który nie jest bezpośrednio wykorzystywany w trakcie uczenia.

Symulator neuronowy Neuronix posługuje się plikami uczącymi i testowymi zarówno w postaci tekstowej jak i binarnej. Dzięki translacji plików tekstowych na specyficzne pliki binarne uzyskuje się dwie zasadnicze korzyści:

- sieć neuronowa w trakcie uczenia wykorzystuje wstępnie przetworzone dane, co znacznie przyspiesza jej pracę,
- proces uczenia może być wspomagany przez dodatkowe mechanizmy prezentacji, które mogą pobierać w znacznie szybszy sposób wstępnie przetworzone informacje z plików binarnych.

Etapy generacji binarnej składa się z dwóch zasadniczych etapów:

1. Zamiana pliku tekstowego na postać binarną bez skalowania wartości liczbowych.
2. Zamiana pliku binarnego na postać przeskalowaną, która jest wykorzystywana przez sieć neuronową.

Etap **skalowania wartości liczbowych** jest o tyle ważny, iż sieć neuronowa, która została zaimplementowana w systemie Neuronix używa **logistycznej funkcji przejścia**. Zatem wielkości podawane na jej wejście a w szczególności na wyjście powinny być z zakresu od 0 do 1. Dlatego też, w systemie Neuronix wykorzystano liniowe przeskalowanie wartości przypadających na jedno wejście (lub wyjście). Dodatkowo skorzystano z normalizacji w oparciu o wartość średnią i odchylenie standardowe. Ostatecznie wzór na przekształcenie wartości rzeczywistej na przeskalowaną ma postać:

(5-1) $$X_s = 0{,}1 + 0{,}24 * (\frac{X_R - \mu}{\sigma} + 1{,}66)\ ,$$

gdzie:

X_S - wartość przeskalowana,

X_R - wartość rzeczywista,

μ - średnia wszystkich wartości *X,*

σ - odchylenie standardowe wszystkich wartości *X*.

Odpowiedź sieci skalowana jest według wzoru:

(5-2) $$X_R = \mu + \sigma * (\frac{X_S - 0{,}1}{0{,}24} - 1{,}66)\ ,$$

a następnie zostaje przedstawiona użytkownikowi.

Ważnym etapem jest również proces **kodowania wartości symbolicznych**. W przypadku wartości symbolicznych wyliczeniowych następuje kodowanie **1 z n**, gdzie n oznacza ilość wartości w całym zbiorze. Symbole hierarchiczne kodowane są metodą **k z n**, zwaną także termometrem. Istotną informacją jest fakt, że w przypadku zmiennych symbolicznych następuje niejawne dodanie wejść lub wyjść sieci. Nazwa pojedynczego wejścia lub wyjścia tworzona jest z nazwy podstawowej i nazwy symbolicznej. Należy pamiętać o kontrolowaniu liczby nazw symbolicznych w ramach pojedynczej kolumny, gdyż może dojść do sytuacji, gdy w pliku uczącym występuje niewiele kolumn wejściowych o wartościach symbolicznych, natomiast całkowita liczba wejść sieci będzie duża. Na etapie tworzenia binarnego pliku uczącego nastąpi przekodowanie tych dwóch kolumn do następującej postaci:

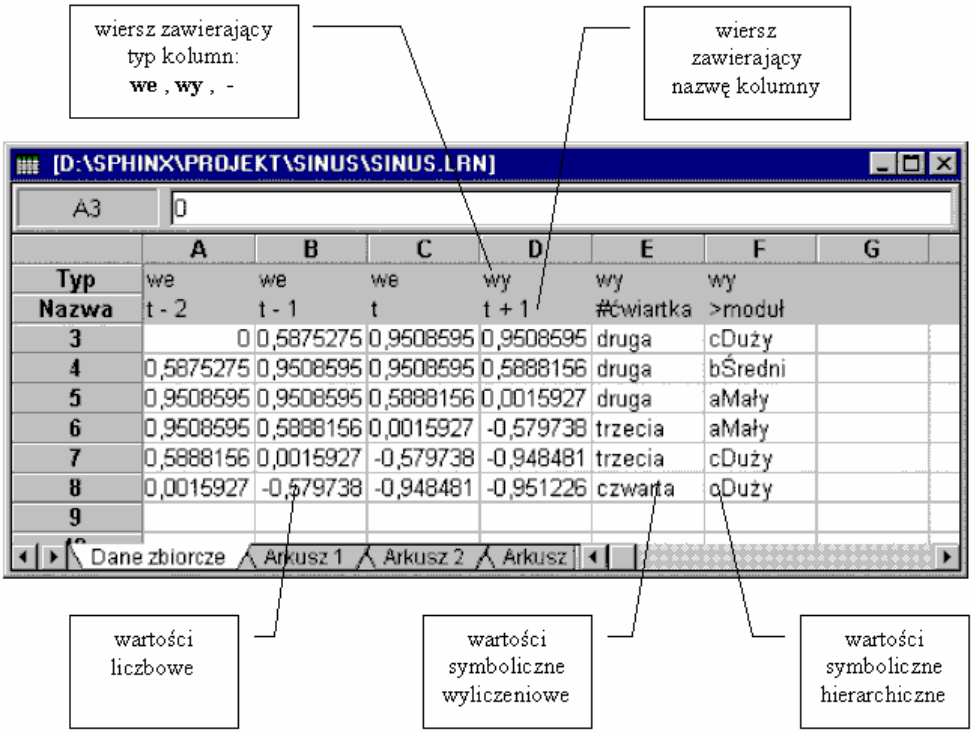
Przed kodowaniem Po kodowaniu

wy	wy	wy	wy
#ćwiartka	#ćwiartka_druga	#ćwiartka_trzecia	#ćwiartka_czwarta
druga	1	0	0
druga	1	0	0
druga	1	0	0
trzecia	0	1	0
trzecia	0	1	0
czwarta	0	1	0
czwarta	0	0	1
czwarta	0	0	1

wy	wy	wy	wy
>moduł	>moduł_aMały	>moduł_bŚredni	>moduł_cDuży
cDuży	1	1	1
bŚredni	1	1	0
aMały	1	0	0
aMały	1	0	0

cDuży	1	1	1
cDuży	1	1	1
bŚredni	1	1	0
aMały	1	0	0

Jak widać pojedyncza kolumna wyjściowa została rozwinięta w obu przypadkach do trzech kolumn wyjściowych. W efekcie tego sieć miałaby 3 wejścia i 7 wyjść. W przypadku zmiennych symbolicznych należy zwrócić szczególną uwagę na wyjścia symboliczne. Proces przekodowania wyjść sieci na odpowiedni symbol odbywa się metodą najmniejszej odległości euklidesowej pomiędzy dwoma wektorami. Należy zdawać sobie sprawę z niebezpieczeństwa złego przekodowania odpowiedzi sieci. Dlatego też zalecane jest nie stosować wyjść symbolicznych. Lepszym rozwiązaniem będzie zastosowanie kodowania „**1 z n**” lub „**k z n**” bezpośrednio w zbiorach uczącym i testowym. Umożliwi to śledzenie wyjść każdego z neuronów wyjściowych, co umożliwia bezpieczniejszą interpretację wyjść. Pliki tekstowe można utworzyć za pomocą dowolnego edytora tekstowego, należy pamiętać o określeniu rodzaju kolumny jako „we” lub „wy”, o wpisaniu w drugim wierszu nazwy kolumny oraz o oddzielaniu wartości znakiem tabulatora. Neuronix dysponuje specjalizowanym arkuszem kalkulacyjnym, który pomaga użytkownikowi w przygotowaniu zbiorów uczących i testowych. Arkusz nazywa się **arkuszem danych**. Dostępny jest w menu głównym **Plik** pod opcją „**Nowy | Arkusz danych**” oraz pasku narzędziowym. **Arkusz danych** stanowi jeden z trzech wyspecjalizowanych arkuszy kalkulacyjnych. Kolejne dwa są przydatne w procesie uczenia sieci. **Arkusz prezentacji** pozwala z kolei na śledzenie zmiennych opisujących stan sieci. Podobną rolę pełni również **Arkusz raportu**, który zapisuje wyniki do wskazanego tekstowego pliku raportu.



Rys. 41. Arkusz danych z utworzonym plikiem uczącym

Arkusze kalkulacyjne odgrywają szczególną rolę w symulatorze neuronowym Neuronix. Wszystkie są zgodne z Excel 5.0/7.0, dzięki czemu możliwe jest przenoszenie gotowych rozwiązań wprost z tego programu. Dodatkowo zastosowano mechanizm współdziałania okna uczącego z Arkuszem Prezentacji i Arkuszem Raportu. Przygotowanie danych uczących stanowi poważny krok w procesie wykorzystania sieci neuronowych. Kolejnym etapem jest nadzorowanie uczenia w sposób pozwalający na śledzenie krytycznych zmiennych i szybkie reagowanie na źle zachowującą się sieć neuronową. Podstawowym oknem, które pozwala na wgląd w najważniejsze wartości jest okno uczenia opatrzone tytułem: **Monitoring uczenia i testowania**. Dzięki niemu można sterować uczeniem sieci neuronowej. Po uruchomieniu uczenia następuje przeglądanie zbioru uczącego i testowego oraz modyfikacja wag tak długo, aż sieć osiągnie wymaganą dokładność na wyjściu. Użytkownik może śledzić następujące wielkości:

- numer epoki,
- numer iteracji,
- rodzaj podejmowanej przez sieć akcji,
- błąd RMS uczenia i testowania,
- tolerancja uczenia i testowania,

- liczbę wzorców uczących i testowych znajdujących się poza tolerancją.

Parametry, jakie umieszczono w oknie uczenia nie są przypadkowe, gdyż ważą one w dużej mierze na decyzji o zakończeniu procesu uczenia. Użytkownik może mieć jednak potrzebę śledzenia większej liczby zmiennych, które według niego mogą mieć istotny wpływ na późniejszą pracę sieci. Dlatego też arkusze kalkulacyjne zostały wyposażone w dodatkowe funkcje, których zadaniem jest pobieranie wszystkich istotnych wielkości z sieci neuronowej. Dodatkowo funkcje te mogą pobierać wskazane dane z plików binarnych. Wstawiając wybraną formułę w komórkę arkusza i odświeżając go powoduje się, że sięga ona do sieci neuronowej lub pliku binarnego i pobiera pożądane informacje. Każdy arkusz może być odświeżany ręcznie poprzez naciśnięcie klawisza >F9< lub automatycznie przez okno uczące. Jednym z parametrów Arkusza Prezentacji i Arkusza Raportu jest częstość odświeżania. Jeżeli użytkownik zdecyduje się na cykliczne przeliczanie to uruchomienie uczenia spowoduje generację impulsów taktujących i natychmiastowe wyprowadzenie wyników. Zastosowanie automatycznie odświeżanych wykresów umożliwia śledzenie wskazanych elementów arkusza w postaci graficznej i ułatwia analizę wyników.

Wśród dodatkowych funkcji arkuszy kalkulacyjnych znajdują się takie, które dają sposobność do:
- pobierania wartości wagi z dowolnej warstwy sieci,
- pobierania wartości wejściowych i wyjściowych sieci,
- pobierania wartości symbolicznych, liczbowych oraz dodatkowych opisów z plików uczących i testowych,
- pobierania poprawnej wartości na wyjściu sieci.

Zastosowanie arkuszy kalkulacyjnych w procesie uczenia jest pożyteczne z punktu widzenia dydaktyki. Dzięki arkuszowi kalkulacyjnemu można śledzić na bieżąco jak zmienia się wartość dowolnej wagi sieci i obserwować wielkość jej zmian po każdym cyklu uczącym. Ponadto można zdefiniować swoje miary błędu i obliczać jego wartość w dowolnej chwili w trakcie uczenia.

Arkusz raportu umożliwia zapisywanie całej historii zmian wybranej wartości. Ma to duże znaczenie przy analizie całego procesu uczenia i wychwytywaniu charakterystycznych etapów w procesie uczenia sieci.

Arkusz prezentacji wraz z pracą krokową ustawioną na wychwytywanie wzorców, które nie mieszczą się w tolerancji pozwala na wgląd w aktualne wejście, wyjście, pożądane wyjście oraz dodatkowe informacje o tym wzorcu. W sytuacji, gdy zbiór uczący zawiera bardzo dużo wzorców uczących, dodatkowa informacja w zbiorach uczących może posłużyć do znalezienia przyczyny błędnego działania sieci, bez żmudnego analizowania całego pliku. Przykładowo zbiór uczący zawiera zbiór ekspertyz 5 ekspertów finansowych. Każdy z nich mając do dyspozycji pewną grupę czynników określał kondycję finansową przedsiębiorstwa w postaci pojedynczej liczby z zakresu od 0 do 100. Zbiór uczący mógłby zawierać wszystkie czynniki oraz odpowiadające im ekspertyzy. Dodatkowo do zbioru uczącego można by wprowadzić informację o numerze eksperta, który wydawał określony osąd. Jeśli decyzje jednego z nich znacznie odbiegały od reszty, to sieć neuronowa będzie miała do czynienia ze sprzecznymi informacjami, a zatem jej wyuczenie może być trudne. Jeśli w takiej sytuacji zastosuje się arkusz kalkulacyjny wraz z pracą krokową to w momencie zatrzymania sieci na wzorcu wychodzącym poza tolerancję będzie można zidentyfikować numer eksperta, który wydał trudną do przyswojenia przez sieć ekspertyzę.

Arkusz danych spełnia rolę okna uruchamiania sieci neuronowej. Wyniki uruchomienia sieci wpisywane są bezpośrednio w komórki arkusza, które znajdują się w kolumnie wyjściowej. Takie rozwiązanie umożliwia elastyczne uruchamianie sieci dla większej ilości wzorców a także natychmiastową obróbkę rezultatów wyprowadzonych przez sieć.

6

Arkusze kalkulacyjne

W rozdziale:

- Informacje ogólne
- Co zawiera skoroszyt ?
- Formatowanie arkuszy
- Arkusz danych
- Arkusz prezentacji
- Arkusz raportu

Cenne informacje

Ćwiczenia z komputerem

Koniecznie przeczytaj

Program Neuronix dysponuje trzema rodzajami **wyspecjalizowanych arkuszy kalkulacyjnych**. Każdy z nich jest zgodny z arkuszem kalkulacyjnym Excel 7.0, pozwala zatem na operowanie na pojedynczych komórkach, wpisywanie formuł obliczeniowych, formatowanie zawartości komórek i ustawianie dodatkowych atrybutów takich jak np. ochrona arkusza.

CO ZAWIERA SKOROSZYT ?

Pod pojęciem **skoroszytu** należy rozumieć **zespół arkuszy** połączony w jedną spójną całość za pomocą okna obsługującego. W systemie Neuronix pojęcia **arkusza** i **skoroszytu** są używane zamiennie.

INFORMACJE OGÓLNE

Poniżej przedstawiono podstawowe informacje dotyczące obsługi arkuszy. Pełne informacje można zasięgnąć w instrukcji obsługi Excel 7.0.

KLAWISZE

Enetr - zatwierdzenie zawartości komórki,
Tab - przejście do następnej komórki,
F2 - przejście w tryb edycyjny,
F9 - natychmiastowe przeliczenie arkusza,
Del - wyczyszczenie zawartości komórek,
Esc - zaniechanie edycji,

Shift + dowolny klawisz ruchu - poszerzenie selekcji.

MYSZ

Lewy klawisz - wskazanie aktywnej komórki
Prawy klawisz - uruchomienie podręcznego menu
Lewy klawisz wraz z przesunięciem kursora - zaznaczenie sekcji komórek
Ctrl + lewy klawisz wraz z przesunięciem kursora - zaznaczenie nowej sekcji komórek,

ZAWARTOŚĆ KOMÓREK

Komórki mogą zawierać dwa typy danych: **stałe** oraz **formuły obliczeniowe**.
Do **stałych** zalicza się:

Liczby	składają się z cyfr od 0 do 9 oraz znaków specjalnych takich jak: +, -, /, *, (,), E, e, %, $,
Data i czas	np. 28/09/97, 17-Mar, 21:55,
Nazwy błędów	#N/A, #VALUE!, #REF!, #NULL!, #DIV/0!, #NUM!, #NAME?,
Tekst	dowolny ciąg znaków

Formuły obliczeniowe to ciągi znaków poprzedzone znakiem równości:

Odwołania do innych komórek	np. =A1, =$A1, =$A$1, =A$1, =Dane1!A1
Nazwy	np. =Wiek, =Miesiąc, =Zadłużenie,
Funkcje	np. =SIN(A1), =PI(),

Operatory:

Arytmetyczne	+ - dodawanie, - - odejmowanie, / - dzielenie, * - mnożenie, % - procent, ^ - potęgowanie,
Tekstowe	& - połączenie,
Relacji	= - równe > - większe niż, < - mniejsze niż, >= - większe lub równe, <= - mniejsze lub równe, <> - różne,

Adresowania	zakres, np. A1:C10 to obszar prostokątny o lewym górnym rogu w A1 oraz prawym dolnym w C10, spacja - część wspólna dwóch obszarów np. A1:C10 C10:F23 daje w wyniku C10, połączenie dwóch obszarów np. A1:C10,C10:F23

Adresacje:

Względna	Odwołanie się do komórki, której położenie określa się względem bieżącej komórki. Gdy komórka zawiera adres względny to po skopiowaniu jej lub przesunięciu następuje automatyczne dopasowanie się adresu tak, aby względne przesunięcie było identyczne do tego sprzed operacji. Np. A1 - względna adresacja komórki A1.
Bezwzględna	Odwołanie się do komórki, której położenie jest określone dokładnie. Gdy komórka zawiera adres bezwzględny to po skopiowaniu jej lub przesunięciu adres nie ulega zmianie. Adresacja bezwzględna oznaczana jest znakiem $ przed kolumną lub wierszem, do których następuje odwołanie bezwzględne. Np. $A$1 - bezwzględna adresacja komórki A1.

Adresacje mogą być częściowo względne i częściowo bezwzględne: np. $A1 jest bezwzględna adresacja kolumny wraz ze względną adresacją wiersza, A$1 jest względna adresacja kolumny wraz z bezwzględną adresacją wiersza.

Adresowanie komórek w innych arkuszach.

Aby odwołać się do komórek innego arkusza niż bieżący należy skorzystać z operatora zewnętrznego adresowania tzn. !.

przykładowe adresacje:

Dane1!A1
Prezentacja2!$A$1
Raport5!$A1

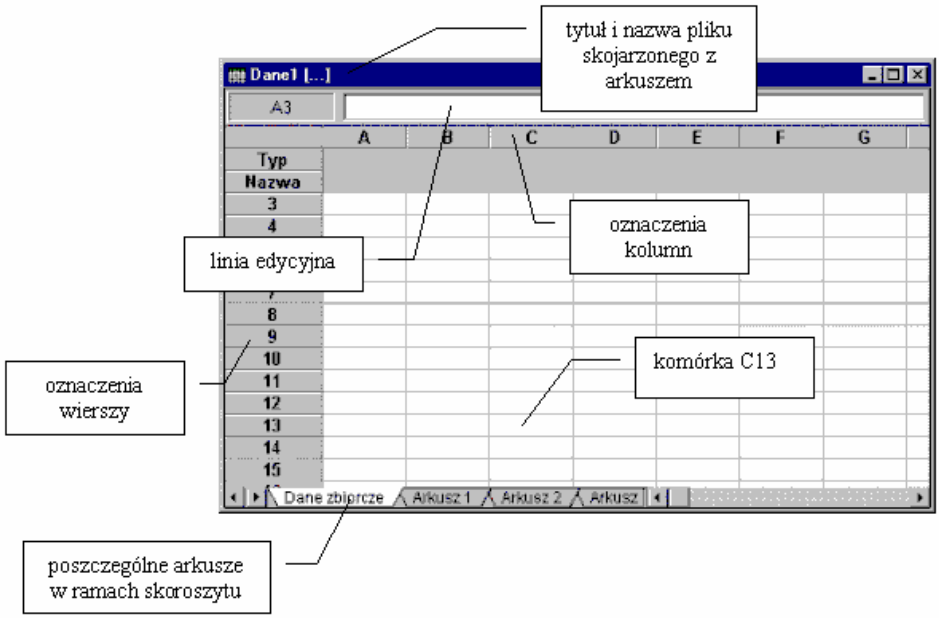
Nazwy

W celu łatwiejszego i bardziej przejrzystego posługiwania się komórkami wprowadzono możliwość definiowania nazw wskazanych komórek, zakresów i stałych. Przykładowo, zamiast posługiwać się formułą obliczeniową = A1 - F13, można zdefiniować nazwę „Przychód” dla A1 oraz „Rozchód” dla F13. W ten sposób zapis: **operacja = Przychód - Rozchód** staje się bardziej zrozumiały.

FUNKCJE

Arkusze kalkulacyjne dysponują dużym zbiorem predefiniowanych funkcji z takich zakresów jak: finanse, matematyka, statystyka i inne. Jako parametr funkcji można przekazywać zarówno stałe jak i adresy komórek zawierające prawidłowe wartości. Dopuszczalne jest również stosowanie wywołań funkcji w miejscu parametrów. Warunkiem jest, aby funkcja podstawiana jako parametr zwracała dla niego poprawną wartość.

WSPÓLNE FUNKCJE I WŁASNOŚCI ARKUSZY PROGRAMU NEURONIX



Rys. 42. Budowa arkusza kalkulacyjnego.

W programie Neuronix występują trzy typy arkuszy, które zostały nazwane:

- **arkusze danych**
- **arkusze prezentacji**
- **arkusze raportu**

Wraz z otwarciem arkusza zostaje mu nadany tymczasowy tytuł, który składa się z nazwy typu arkusza i unikatowego numeru. Każdy z arkuszy może zostać skojarzony z plikiem dyskowym. Fakt ten ma miejsce po odczycie pliku do arkusza lub po zapisie danych z arkusza do wskazanego pliku. Jeżeli arkusz jest skojarzony z plikiem to w nawiasach kwadratowych w tytule okna znajduje się jego ścieżka dostępu. Jeżeli brak owego skojarzenia to w nawiasach kwadratowych pojawiają się trzy kropki.

Każdy z nich spełnia swoje odrębne funkcje, chociaż wiele funkcji i własności jest wspólnych. Należą do nich:

- ogólne zasady korzystania z arkuszy kalkulacyjnych,
- większość funkcji podręcznego menu,
- funkcje z grupy menu głównego **Edycja** służące do kopiowania, usuwania i wklejania zawartości komórek.

MENU PODRĘCZNE ARKUSZY

Naciśnięcie prawego klawisza myszki w obrębie arkusza powoduje otwarcie się podręcznego menu. Poniżej zostaną omówione poszczególne opcje.

Ustawienia | Komórek

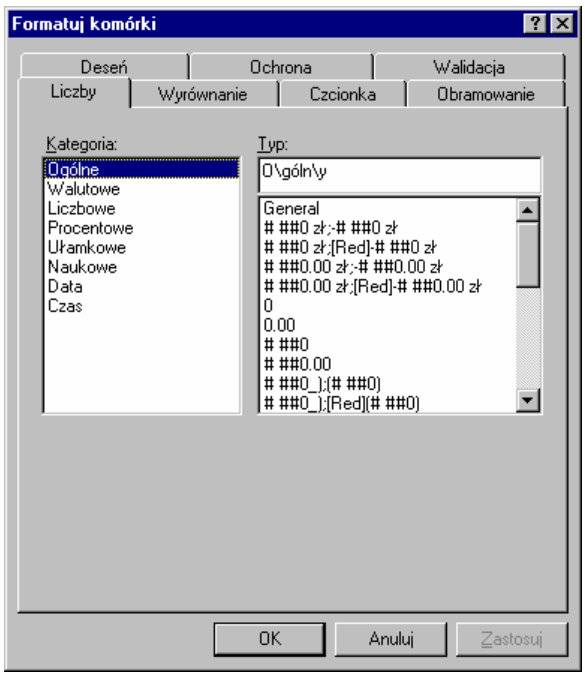
Ustawienie właściwości zaznaczonych komórek.

Liczby – format (sposób wyświetlania) liczb w komórce

Symbol formatu	Opis
General	Wyświetlanie liczb w formacie ogólnym.
0	Znacznik pozycji. Jeśli liczba zawiera mniej cyfr niż jest znaczników w formacie, to zostaje ona uzupełniona zerami. Jeśli na prawo od przecinka jest więcej cyfr niż znaczników, to część ułamkowa zostaje zaokrąglona do tylu miejsc po przecinku, ile jest znaczników w formacie. Nadmiarowe cyfry na lewo od przecinka zostają

	zachowane.
#	Znacznik pozycji. Funkcjonuje podobnie jak znacznik 0, z tym że liczba nie jest uzupełniana 0 jeśli zawiera mniej cyfr niż jest znaczników w formacie.
?	Znacznik pozycji. Działa podobnie jak znacznik 0, przy czym uzupełnia brakujące pozycje spacjami.
. (kropka)	Przecinek dziesiętny. Ustala liczbę cyfr (zer lub #) wyświetlanych po każdej stronie przecinka dziesiętnego. Jeśli format po lewej stronie przecinka zawiera tylko #, liczby mniejsze od 1 zaczynają się od przecinka. Jeśli format zawiera 0 na lewo od przecinka, liczby mniejsze od 1 zaczynają się od 0 przed przecinkiem.
%	Wyświetla liczbę jako procent. Liczba jest mnożona przez 100 oraz dopisywany jest znak %.
, (przecinek)	Separator tysięcy. Jeśli format zawiera przecinek między # lub 0, liczba jest wyświetlana z podziałem na tysiące. Jeśli po przecinku występuje spacja, to liczba jest skalowana przez 1000. Np. format 0, skaluje przez 1000 (10 000 zostanie wyświetlone jako 10).
E- E+ e- e+	Wyświetla liczbę w zapisie naukowym. Jeśli format zawiera symbol zapisu naukowego na lewo od 0 lub #, to liczba jest wyświetlana w zapisie naukowym z dodatkiem E lub e. Liczba 0 i # na prawo od przecinka dziesiętnego określa liczbę cyfr wykładnika. E- i e- umieszczają znak minusa przed ujemnymi wykładnikami. E+ i e+ umieszczają znak minusa przed ujemnymi wykładnikami oraz znak plusa przed dodatnimi wykładnikami.
$ - + / ( ) : spacja	Wyświetla dany znak. Aby wyświetlić znak inny niż wymienione należy poprzedzić go znakiem \ lub umieścić w cudzysłowach (" "). Można również użyć / dla formatów ułamkowych.
\	Wyświetla następny znak. \ nie jest wyświetlany. Znaki lub napisy można również wyświetlić poprzez umieszczenie w cudzysłowach (" "). \ jest wstawiany automatycznie dla następujących znaków: ! ^ & ` (lewy pojedynczy cudzysłów) ' (prawy pojedynczy cudzysłów) ~ { } = < >
* (gwiazdka)	Powtarza następny znak aż do wypełnienia szerokości kolumny. W jednej sekcji formatu może być tylko jedna gwiazdka.
_ (podkreślenie)	Przeskakuje szerokość następnego znaku. Np. aby liczby ujemne otoczone ( ) były wyrównane z liczbami dodatnimi, możesz wstawić format _) dla liczb dodatnich, żeby przeskoczyć szerokość nawiasu.
"tekst"	Wyświetla teks wewnątrz cudzysłowów.
@	Znacznik pozycji tekstu. Jeśli w komórce jest tekst, zastępuje on znak @.
m	Numer miesiąca. Wyświetla miesiące jako liczby bez wiodących 0 (np. 1-12). Może również reprezentować minuty, jeśli użyte z formatami h lub hh.
mm	Numer miesiąca. Wyświetla miesiące jako liczby z wiodącymi 0 (np. 01-12). Może również reprezentować minuty, jeśli użyte z formatami h lub hh.
mmm	Skrót miesiąca. Wyświetla miesiące w postaci

	skrótu (np. , sty-gru).
mmmm	Nazwa miesiąca. Wyświetla miesiące w postaci pełnej nazwy (np. styczeń-grudzień).
d	Numer dnia. Wyświetla dzień jako liczbę bez wiodącego 0 (np. 1-2).
dd	Numer dnia. Wyświetla dzień jako liczbę z wiodącym 0 (np. 01-02).
ddd	Skrót dnia. Wyświetla dzień w postaci skrótu (np. Pn-N).
dddd	Nazwa dnia. Wyświetla dni w postaci pełnej nazwy (np. poniedziałek-niedziela).
yy	Numer roku. Wyświetla rok jako liczbę dwucyfrową (np. 00-99).
yyyy	Numer roku. Wyświetla rok jako liczbę czterocyfrową (np. 1900-2078).
h	Liczba godzin. Wyświetla godzinę jako liczbę bez wiodącego 0 (np. 1-23). Jeśli format zawiera któryś z formatów AM/PM, wykorzystywany jest zegar 12-godzinny, w przeciwnym razie 24-godzinny.
hh	Liczba godzin. Wyświetla godzinę jako liczbę z wiodącym 0 (np. 01-23). Jeśli format zawiera któryś z formatów AM/PM, wykorzystywany jest zegar 12-godzinny, w przeciwnym razie 24-godzinny.
m	Liczba minut. Wyświetla minuty jako liczbę bez wiodącego 0 (np. 0-59). Format m musi wystąpić bezpośrednio po symbolu h lub hh. W przeciwnym razie jest interpretowany jako numer miesiąca.
mm	Liczba minut. Wyświetla minuty jako liczbę bez wiodącego 0 (np. 00-59). Format m musi wystąpić bezpośrednio po symbolu h lub hh. W przeciwnym razie jest interpretowany jako numer miesiąca.
s	Liczba sekund. Wyświetla sekundy jako liczbę bez wiodącego 0 (np. 0-59).
ss	Liczba sekund. Wyświetla sekundy jako liczbę z wiodącym 0 (np. 00-59).
AM/PM am/pm A/P a/p	12-godzinny zegar. Wyświetla AM, am, A, lub a dla czasów między północą a południem; PM, pm, P, lub p dla czasów od południa do północy.
[h]	Wypisuje całkowitą liczbę godzin.
[m]	Wypisuje całkowitą liczbę minut.
[s]	Wypisuje całkowitą liczbę sekund.
s.0, s.00, s.000, ss.0, ss.00, ss.000	Wypisuje ułamkową część sekund.
[Black]	Wyświetla tekst komórki na czarno.
[Blue]	Wyświetla tekst komórki na niebiesko.
[Cyan]	Wyświetla tekst komórki w kolorze cyjanowym.
[Green]	Wyświetla tekst komórki na zielono.
[Magenta]	Wyświetla tekst komórki w kolorze magenty.
[Red]	Wyświetla tekst komórki na czerwono.
[White]	Wyświetla tekst komórki na biało.
[Yellow]	Wyświetla tekst komórki na żółto.
[COLOR n]	Wyświetla tekst komórki w kolorze o zadanym numerze n w palecie kolorów.



Rys. 43. Opcje formatowania komórki

[wartość warunkowa] Każdy format może składać się z 4 sekcji: dla liczb dodatnich, ujemnych, zer oraz dla tekstu. Używając nawiasów wartości warunkowej ([ ]), możesz wyznaczyć różne warunki dla każdej sekcji. Np. liczby dodatnie możesz wyświetlać na czarno, ujemne na czerwono, a zera na niebiesko. Jest to realizowane następującym formatem: [>0] [Black]General; [<0] [Red]General; [Blue]General

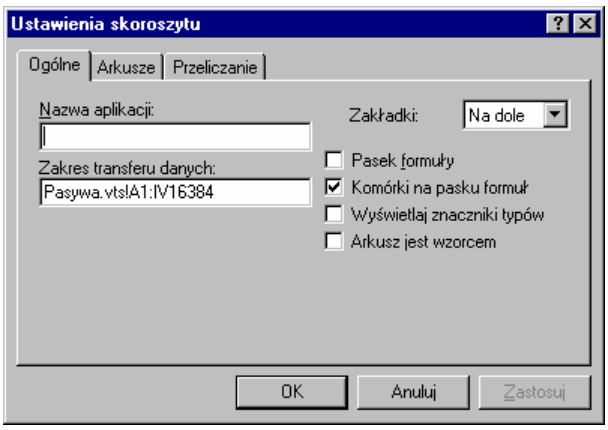
Przykłady

Format	Zawartość komórki	Wynik
#,##0.00	32002,00	32 002,00
0.00%	0,32	32,00%
# ??/??	2/26	1/13
0.00E+00	2/26	7,69E-02
ddd-mmm	1900-1-1	pon-sty
h:mm A/P	23:03	11:03 P

Wyrównanie – wyrównanie tekstu w komórce w pionie i w poziomie, możliwe włączenie opcji zawijania tekstu (tekst szerszy od kolumny nie przesłania następnej kolumny lecz jest przenoszony do następnego wiersza)
Czcionka – określenie czcionki, jej stylu i wielkości, efektów typu podkreślenie, przekreślenie, koloru
Obramowanie – rodzaj i kolor linii obramowania komórki (dla grup komórek również linie wewnętrzne), które fragmenty obramowania (linii wewnętrznych) są widoczne
Deseń – tło komórki
Ochrona – włączenie blokady komórki (uniemożliwienie zmiany zawartości) i / lub ukrycie komórki
Walidacja – reguła kontrolująca poprawność zawartości komórki

Ustawienia | Skoroszytu

Ustawienie właściwości skoroszytu.
Ogólne – nazwa aplikacji, zakres transferu danych, położenie zakładek arkuszy, wyświetlanie paska formuły, adresu aktywnej komórki, stosowanie znaczników (markerów) typu (komórki o tym samym typie, np. zawierające liczby, mają obramowanie w tym samym kolorze), traktowanie arkusza jako wzorzec
Arkusze – właściwości arkuszy wchodzących w skład skoroszytu: górna lewa komórka arkusza, aktywna komórka, wybrany zakres, włączenie ochrony

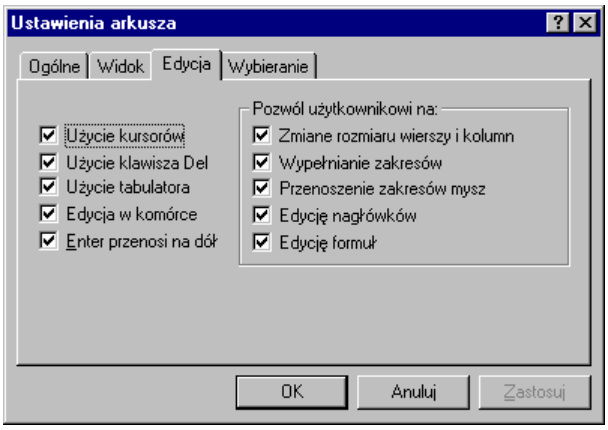


Rys. 44. Opcje ustawiania skoroszytu

Przeliczanie – włączenie automatycznego przeliczania, włączenie przeliczania iteracyjnego – komórki w arkuszu są przeliczane liczbę razy podaną przez parametr *Maksymalna iteracja* lub aż do chwili, gdy wszystkie komórki nie zmieniają się bardziej niż *Maksymalna zmiana.* Zaznaczenie opcji *Precyzja jak wyświetlana* powoduje przechowywanie wartości z dokładnością podaną w formacie komórki

Ustawienia | Arkusza | Właściwości

Ogólne - określenie nazwy arkusza (nazwę arkusza można również zmienić dwukrotnie klikając myszą na zakładce arkusza)



Rys. 45. Opcje ustawienia arkusza

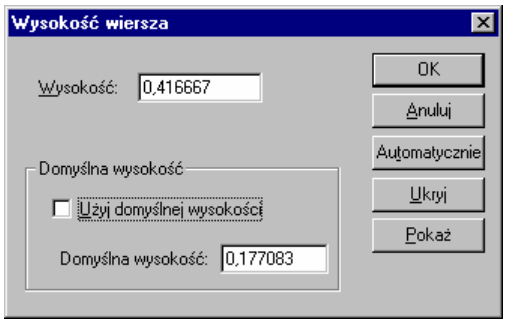
Widok - określenie widocznej części arkusza (min. i maks. wiersz, kolumna), wyświetlanie: formuł zamiast ich wartości, linii siatki, wartości zerowych (jeśli ta pozycja nie jest zaznaczona, komórka jest wyświetlana jako pusta, jeśli zawiera 0) nagłówków wierszy, kolumn; określenie, które suwaki są widoczne; wybranie skali widoku; opcja *Zakresy* umożliwia wyłączenie pokazywania wybranego zakresu komórek, włączenie lub pracę automatyczną.

Edycja – określa możliwość korzystania z wybranych klawiszy (kursory, Del, tabulacja), zakres uprawnień użytkownika: zmiana rozmiarów wierszy, kolumn; wypełnianie zakresów; przenoszenie zakresów myszą; edycja nagłówków; edycja formuł.

Wybieranie – określenie czy użytkownik może wybierać zakres komórkę; czy może wybierać obiekty; włączenie trybu wierszowego – po zaznaczeniu tej opcji wskazanie pojedynczej komórki zaznacza cały wiersz (kolumnę, jeśli wskażemy nagłówek kolumny).

Ustawienia | Arkusza | Wstaw nowy
Dodanie nowego arkusza do aktywnego skoroszytu.
Ustawienia | Arkusza | Usuń aktywny arkusz
Usunięcie aktywnego arkusza ze skoroszytu.

Ustawienia | Wierszy | Wysokość wiersza



Rys. 46. Opcja ustawiania wysokości wiersza

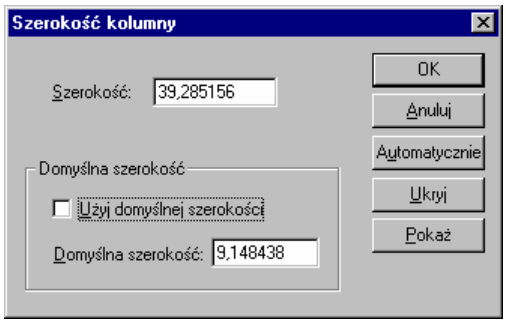
Ustalenie bieżącej wysokości wiersza. Jeśli zaznaczymy pole *Użyj domyślnej wysokości* to wybrana zostanie domyślna wysokości wiersza (możemy też wtedy zdefiniować wysokości domyślną w polu *Domyślna wysokości*).

Ustawienia | Wierszy | Ukryj
Ukrywa wiersz, w którym znajduje się aktywna komórka (lub wiersze, jeśli zaznaczony jest jakiś obszar).

Ustawienia | Wierszy | Odkryj
Odkrywa ukryte uprzednio wiersze.

Ustawienia | Wierszy | Domyślna wysokość
Ustalenie wysokości wiersza aktywnej komórki (lub wierszy zaznaczonego obszaru) na domyślną wysokość.

Ustawienia | Kolumn | Szerokość kolumny



Rys. 47. Opcja ustawiania szerokości kolumny

Ustalenie bieżącej szerokości kolumny. Jeśli zaznaczymy pole *Użyj domyślnej szerokości* to wybrana zostanie domyślna szerokość kolumny (możemy też wtedy zdefiniować szerokość domyślną w polu *Domyślna szerokość*).

Ustawienia | Kolumn | Autodopasowanie
Ustalenie szerokości kolumny tak, by zawartość dowolnej komórki w tej kolumnie mieściła się w całości, w jednym wierszu. To samo można uzyskać klikając dwukrotnie myszą na linii oddzielającej nagłówek wybranej kolumny od sąsiedniej.

Ustawienia | Kolumn | Ukryj
Ukrywa kolumnę, w której znajduje się aktywna komórka (lub kolumny, jeśli zaznaczony jest jakiś obszar).

Ustawienia | Kolumn | Odkryj
Odkrywa ukryte uprzednio kolumny.

Ustawienia | Kolumn | Domyślna szerokość
Ustalenie szerokości kolumny aktywnej komórki (lub kolumn zaznaczonego obszaru) na domyślną szerokość.

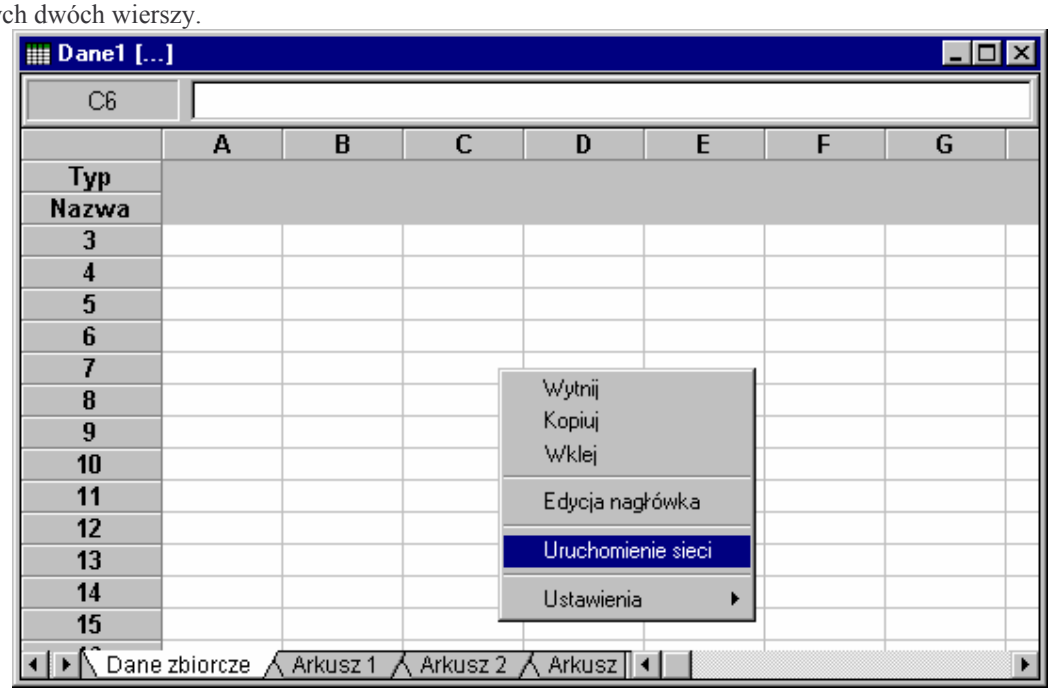
ARKUSZ DANYCH

Arkusz Danych służy jako okno przygotowywania danych uczących i testowych, a także wykorzystywany jest do uruchamiania wyuczonej sieci. Pierwsze dwa wiersze arkusza są przeznaczone do opisu kolumny.

Wiersz nazwany „Typ” służy do wprowadzania symbolu kolumny. Symbol ten może być jednym ze słów:
- „we” - co oznacza, że kolumna odpowiada wejściu sieci,
- „wy” - co oznacza, że kolumna odpowiada wyjściu sieci.

Jeżeli żadne z tych słów nie jest wprowadzone do wiersza „Typ”, to kolumna traktowana jest jako dodatkowy opis, który nie uczestniczy bezpośrednio w procesie uczenia. Wiersz „Nazwa” służy do wprowadzania nazwy wejścia sieci. Nazwa każdego z wejść sieci musi być niepowtarzalna. Kolejność kolumn jest dowolna. Jeżeli nazwa kolumny rozpoczyna się od znaku specjalnego, to jej wartości traktowane są jak symboliczne. Zastrzeżone są następujące znaki:
- # - oznacza, że symbole kolumny zostaną zamienione na kod „**1 z n**”,
- > - oznacza, że symbole kolumny zostaną zamienione na kod „**k z n**” .

Wiersz „Typ” oraz „Nazwa” są stałe, co oznacza, że ich położenie nie zmienia się przy przewijaniu arkusza w pionie. Pociąga to za sobą konieczność posługiwania się opcją „Edycja nagłówka”, która uaktywnia komórkę w obrębie tych dwóch wierszy.

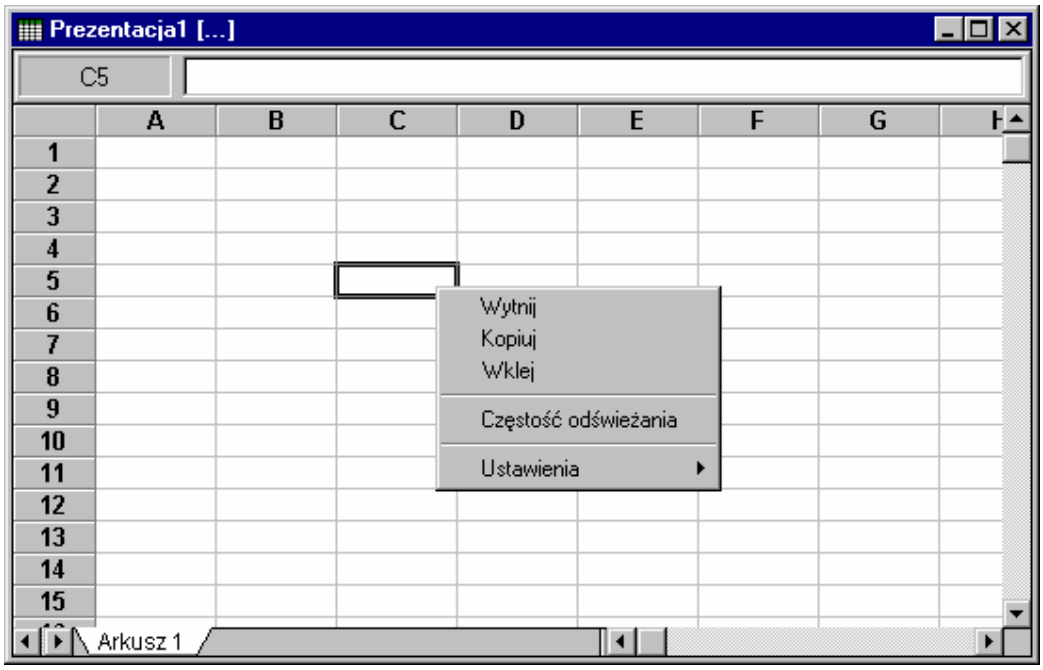


Rys. 48. Arkusz danych wraz z podręcznym menu

Ostatnim charakterystycznym elementem arkusza danych jest możliwość uruchamiania sieci. Jeżeli nagłówek (dwa pierwsze wiersze) zostaną wypełnione prawidłowo tzn. ilość, nazwy i typy wartości w kolumnach wejściowych i wyjściowych będą zgodne z plikiem uczącym to program Neuronix wypełni kolumny wyjściowe wartościami, jakie pojawią się na wyjściu sieci. Kolejność kolumn nie musi być taka sama jak w pliku uczącym. Wykorzystanie arkusza jako okno uruchamiania sieci daje sposobność przygotowania formuł obliczeniowych, które zostaną automatycznie uruchomione po wpisaniu wartości wyjściowych. W ten sposób można przygotować cały zestaw dodatkowych wskaźników opisujących jakość sieci i co pewien czas przeprowadzać uruchomienie obserwując jak zmieniają swoją wartość.

ARKUSZ PREZENTACJI

Arkusz prezentacji służy do dynamicznego obserwowania zmiennych charakteryzujących stan sieci. W trakcie, gdy użytkownik wykorzystuje okno uczące, arkusz prezentacji otrzymuje z niego impulsy taktujące. Jeżeli częstość odświeżania arkusza prezentacji jest ustawiona na konkretną wartość np. po każdej epoce, to będzie wywoływane przeliczanie arkusza po nadejściu wskazanego impulsu taktującego.

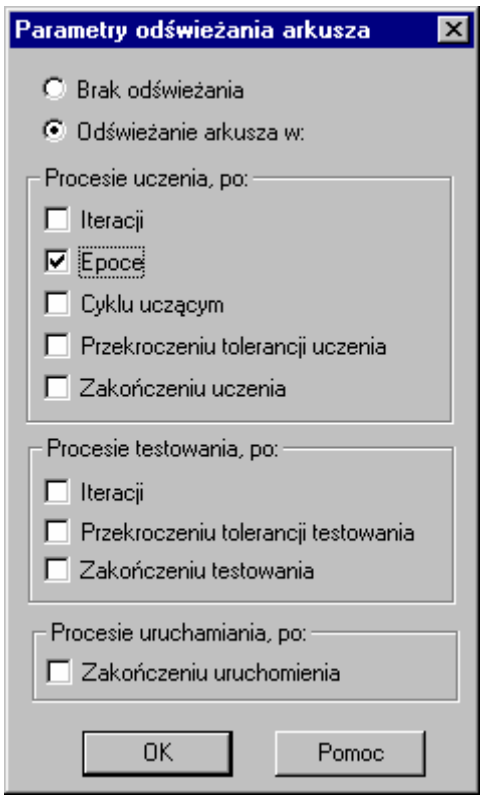


Rys. 49. Arkusz prezentacji

Bezpośredni dostęp do sieci neuronowej, pliku uczącego i testowego uzyskuje się przez grupę funkcji neuronowych, które zostały omówione wcześniej. Jeżeli odświeżenie arkusza nastąpi w ściśle określonym czasie to funkcja neuronowa sięgnie do sieci neuronowej, odczyta pożądaną wartość i przedstawi ją w komórce arkusza. Do użytkownika należy wykorzystanie informacji, jaka pojawia się w tym arkuszu.

Wstawienie, chociaż jednego arkusza prezentacji powoduje spowolnienie uczenia. Dlatego, jeśli nie jest to konieczne nie powinno się otwierać tego typu okna.

Okno ustalania częstości odświeżania pozwala na elastyczny dobór impulsów taktujących.

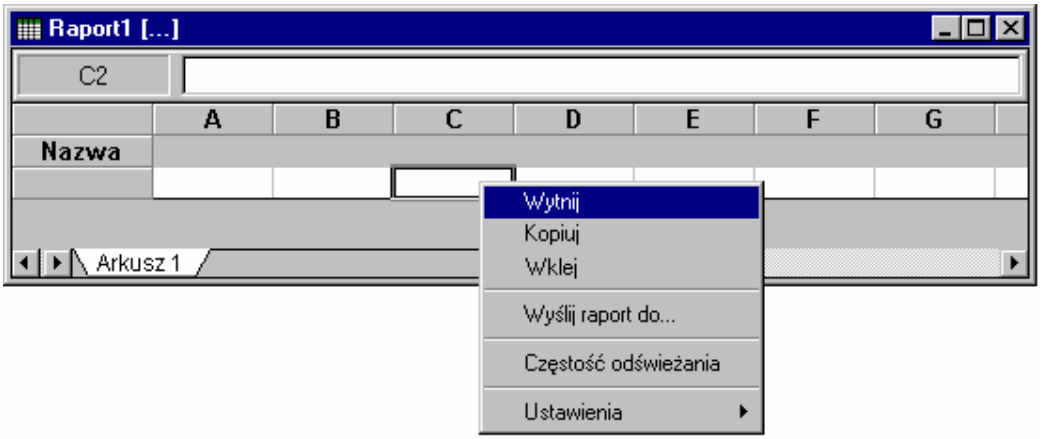


Rys. 50. Parametry odświeżania arkusza

Arkusz prezentacji może być odświeżany po nadejściu kilku rodzajów impulsów, zarówno w trakcie uczenia jak również testowania i uruchamiania sieci.

ARKUSZ RAPORTU

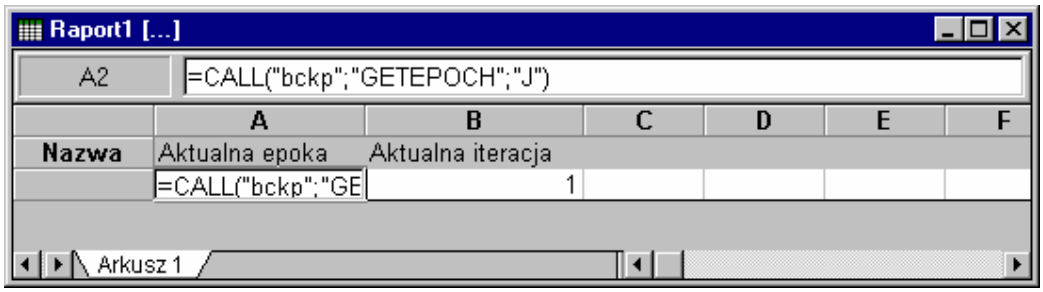
Arkusz raportu pozwala na zapis zmian wybranych wielkości charakteryzujących proces uczenia oraz parametrów sieci na przestrzeni czasu. Podobnie jak arkusz prezentacji może być odświeżany w takt impulsów nadchodzących od okna uczenia lub okna uruchomienia sieci, czyli arkusza danych. Ze względu na ograniczoną liczbę wierszy (16384) arkusza kalkulacyjnego, zapis historii zmian dokonuje się do pliku tekstowego, który może być potem importowany do dowolnego edytora tekstowego.



Rys. 51. Arkusz raportu z menu podręcznym

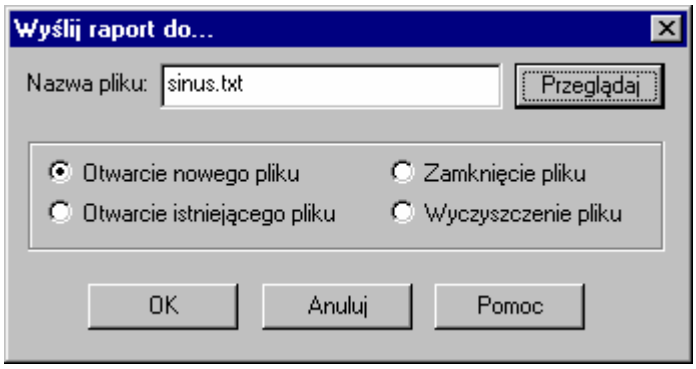
Arkusz raportu składa się z dwóch wierszy. Pierwszy z nich to „Nazwa”, gdzie umieszcza się nagłówek, jaki pojawi się w pliku tekstowym. Drugi wiersz powinien zawierać formuły obliczeniowe, których wyniki będą dopisywane do pliku tekstowego w miarę jak arkusz odświeża się w takt impulsów. Można wyobrazić sobie arkusz, który będzie miał

częstość odświeżania ustawioną na „Odświeżanie w procesie uczenia: po epoce” oraz wypełnione w następujący sposób komórki:



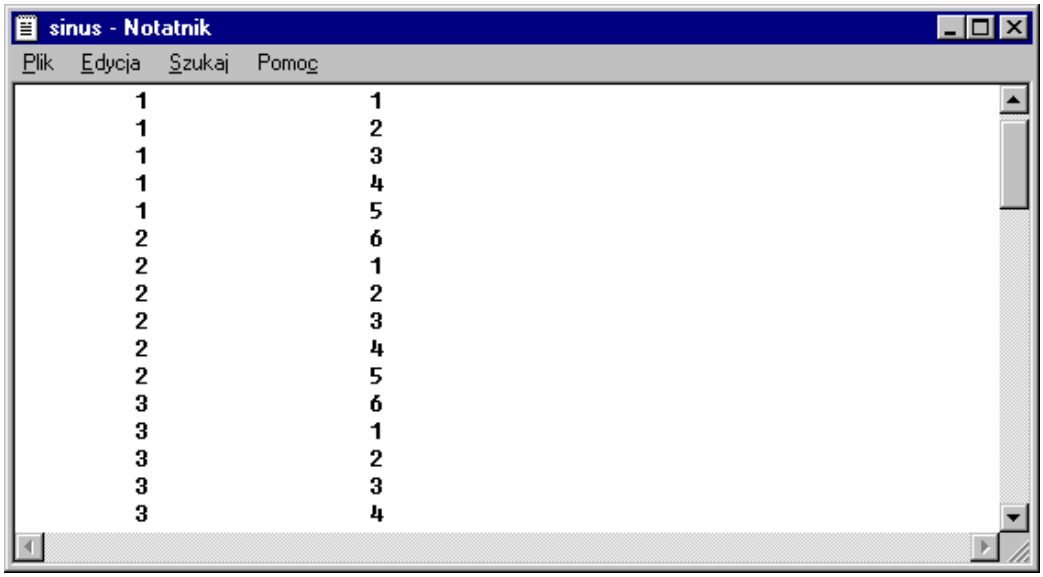
Rys. 52. Przygotowanie danych w arkuszu raportu

Zanim arkusz będzie mógł zapisywać wyniki do pliku powinno nastąpić otwarcie pliku raportu, co realizuje się poprzez wybranie opcji „Otwarcie pliku raportu”
z podręcznego menu.



Rys. 53. Opcja tworzenia pliku tekstowego raportu

Jeżeli otwarcie pliku zostanie przeprowadzone pomyślnie to opcja „Otwarcie pliku raportu” będzie zaznaczona daszkiem. Uruchomienie uczenia i zatrzymanie po 3 epoce spowoduje wygenerowanie pliku o zawartości jak poniżej:



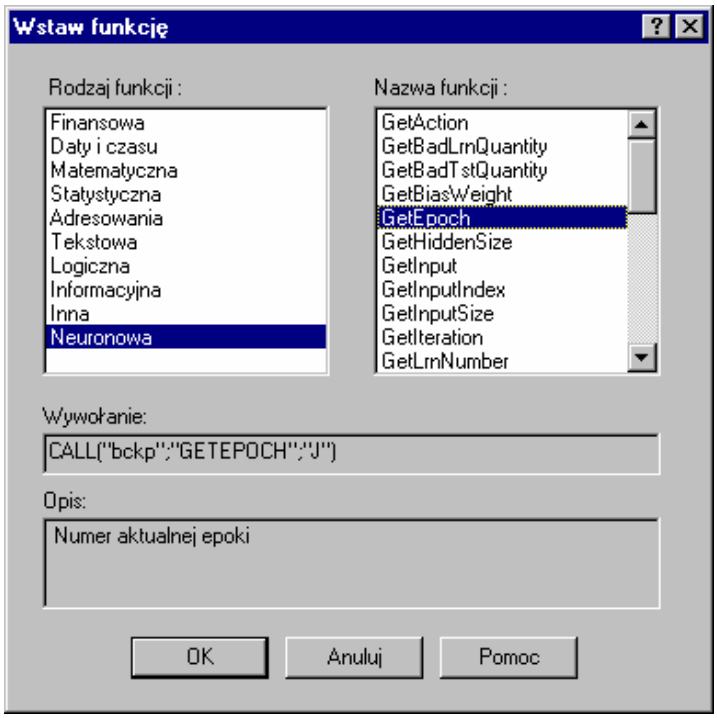
Rys. 54. Zawartość pliku tekstowego sinus.txt

Gdy arkusz raportu ma być usunięty, należy zamknąć uprzednio otwarty plik raportu.

Wstawienie, chociaż jednego arkusza raportu powoduje spowolnienie uczenia. Dlatego, jeśli nie jest to konieczne nie powinno się otwierać tego typu okna.

FUNKCJE NEURONOWE ARKUSZY

Arkusze kalkulacyjne dysponują wbudowanymi funkcjami, które mogą wydatnie wspomóc analizę danych. Oprócz standardowych funkcji istnieje również możliwość korzystania z funkcji przygotowanych w postaci biblioteki dynamicznej dll. Symulator Neuronix pozwala na wykorzystanie zaproponowanych procedur w celu śledzenia stanu sieci i wglądu w pliki uczące i testowe.



Rys. 55. Dialog pozwalający wstawiać funkcję do arkuszy

Okno, które służy do wprowadzania funkcji jest uruchamiane opcją **Wstawienie funkcji** z podręcznego menu. Każda z procedur neuronowych korzysta z funkcji CALL, której parametrami są nazwa biblioteki, nazwa funkcji oraz parametry przekazywane do jej wnętrza. Po wybraniu jednej z funkcji następuje skopiowanie wywołania funkcji wraz z opisem parametrów. Użytkownik musi postawić znak równości przed wywołaniem formuły i wstawić odpowiednie parametry. Poniżej omówiono każdą z funkcji neuronowych:

Jeżeli wejście operuje na wartościach symbolicznych należy pamiętać, że pełna nazwa wejścia jest złożeniem nazwy podstawowej oraz nazwy wartości symbolicznej np. jeżeli wejście nazywa się #płeć to sieć neuronowa będzie miała dwa wejścia: #płeć_kobieta oraz #płeć_mężczyzna

GetAction	
Wywołanie:	CALL("bckp";"GetAction";"I")
Parametry:	brak
Funkcja zwraca numer działania jaki aktualnie podejmowany jest przez sieć.	
Rezultat:	0 - brak działania 1 - uczenie 2 - testowanie 3 - uruchamianie

GetBadLrnQuantity	
Wywołanie:	CALL("bckp";"GetBadLrnQuantity";"J")
Parametry:	Brak
Funkcja zwraca ilość wzorców uczących dla których wyjście sieci nie mieści się w granicy tolerancji uczenia.	
Rezultat:	Ilość wzorców jako liczba *long*

GetBadTstQuantity	
Wywołanie:	CALL("bckp";"GetBadTstQuantity";"J")
Parametry:	brak
Funkcja zwraca ilość wzorców testowych dla których wyjście sieci nie mieści się w granicy tolerancji testowania.	
Rezultat:	Ilość wzorców jako liczba *long*.

GetBiasWeight	
Wywołanie:	CALL("bckp";"GetBiasWeight";"EII"; neuron; warstwa)
Parametry:	warstwa - numer warstwy sieci; warstwa wejściowa ma numer 0 a następne odpowiednio wyższe numery; dopuszczalna wartość to 1,2,3,4 neuron - numer neuronu w podanej warstwie; pierwszy neuron ma numer 0 a ostatni *n-1*, gdzie *n* jest ilością neuronów w warstwie *warstwa*.
Funkcja zwraca wartość wagi pomiędzy neuronem ze wskazanej warstwy a elementem *bias*.	
Rezultat:	Wartość wagi jako liczba *double*.

GetEpoch	
Wywołanie:	CALL("bckp";"GetEpoch";"J")
Parametry:	Brak
Funkcja zwraca numer bieżącej epoki uczącej.	
Rezultat:	Numer epoki jako liczba *long*

GetHiddenSize	
Wywołanie:	CALL("bckp";"GetHiddenSize";"II";nr_ukrytej)
Parametry:	nr_ukrytej - numer warstwy ukrytej; dopuszczalne wartości to 1,2,3
Funkcja zwraca rozmiar wskazanej warstwy ukrytej.	
Rezultat:	Rozmiar jako liczba *int*.

GetInput	
Wywołanie:	CALL("bckp";"GetInput";"EI";nr_wejścia)
Parametry:	nr_wejścia – numer wejścia sieci; pierwsze wejście ma numer 0
Funkcja zwraca wartość wskazanego wejścia.	
Rezultat:	Wartość jako liczba *double*

GetInputIndex	
Wywołanie:	CALL("bckp";"GetInputIndex";"MC";nazwa_wejścia)
Parametry:	nazwa_wejścia - pełna nazwa wejścia sieci UWAGA: jeżeli wyjście operuje na wartościach symbolicznych należy pamiętać, że pełna nazwa wyjścia jest złożeniem nazwy podstawowej oraz nazwy wartości symbolicznej np. jeżeli wyjście nazywa się #płeć to sieć neuronowa będzie miała dwa wyjścia: #płeć_kobieta oraz #płeć_mężczyzna
Funkcja zwraca numer wejścia, którego nazwę przekazuje się jako parametr.	
Rezultat:	Numer wejścia jako liczba *int*.

GetInputSize	
Wywołanie:	CALL("bckp";"GetInputSize";"I")
Parametry:	brak
Funkcja zwraca rozmiar wejścia sieci.	
Rezultat:	Rozmiar jako liczba *int*.

GetIteration	
Wywołanie:	CALL("bckp";"GetIteration";"J")
Parametry:	brak
Funkcja zwraca numer bieżącej iteracji.	
Rezultat:	Numer iteracji jako liczba *long*.

GetLrnNumber	
Wywołanie:	CALL("bckp";"GetLrnNumber";"EIC";numer_wiersza;nazwa_kolumny)
Parametry:	numer_wiersza - numer wiersza pliku uczącego; Numer wiersza jest jednocześnie numerem wzorca podawanego na wejście sieci; dopuszczalna wartość to liczba całkowita większa od 0 nazwa_kolumny - nazwa kolumny w pliku uczącym
Funkcja zwraca wartość numeryczną, która znajduje się w podanym wierszu i kolumnie pliku uczącego sieci neuronowej. Ponieważ numer wiersza jest jednocześnie numerem wzorca to można jako *numer_wiersza* wstawić funkcję GetPatternNumber.	
Rezultat:	Wartość numeryczna typu *double*

GetLrnQuantity	
Wywołanie:	CALL("bckp";"GetLrnQuantity";"J")
Parametry:	Brak
Funkcja zwraca ilość wzorców uczących w pliku uczącym.	
Rezultat:	Liczba wzorców jako liczba *long*.

GetLrnRate	
Wywołanie:	CALL("bckp";"GetLrnRate";"E")
Parametry:	Brak
Funkcja zwraca wartość współczynnika uczenia	
Rezultat:	Wartość liczbowa typu double.

GetLrnRMS	
Wywołanie:	CALL("bckp";"GetLrnRMS";"E")
Parametry:	brak
Funkcja zwraca bieżącą wartość błędu RMS uczenia.	
Rezultat:	Wartość błędu jako liczba *double*.

GetLrnSymbol	
Wywołanie:	CALL("bckp";"GetLrnSymbol";"CIC";numer_wiersza;nazwa_kolumny)
Parametry:	numer_wiersza - numer wiersza pliku uczącego; numer wiersza jest jednocześnie numerem wzorca podawanego na wejście sieci; dopuszczalna wartość to liczba całkowita większa od 0 nazwa_kolumny - nazwa kolumny w pliku uczącym

Funkcja zwraca wartość symboliczną, która znajduje się w podanym wierszu i kolumnie pliku uczącego sieci neuronowej. Ponieważ numer wiersza jest jednocześnie numerem wzorca to można jako *numer_wiersza* wstawić funkcję GetPatternNumber.	
Rezultat:	Wartość symboliczna typu *char**.

GetLrnTol	
Wywołanie:	CALL("bckp";"GetLrnTol";"E")
Parametry:	brak
Funkcja zwraca bieżącą wartość tolerancji uczenia.	
Rezultat:	Wartość tolerancji jako liczba *double*.

GetLrnTolMult	
Wywołanie:	CALL("bckp";"GetLrnTolMult";"E")
Parametry:	brak
Funkcja zwraca wartość współczynnika, przez który mnożona jest tolerancja uczenia.	
Rezultat:	Wartość współczynnika typu *double*.

GetLrnTolTresh	
Wywołanie:	CALL("bckp";"GetLrnTolTresh";"I")
Parametry:	brak
Funkcja zwraca procentowy próg przy którym następuje zmiana tolerancji uczenia.	
Rezultat:	Próg typu *int*.

GetMomentum	
Wywołanie:	CALL("bckp";"GetMomentum";"E")
Parametry:	brak
Funkcja zwraca wartość współczynnika momentu.	
Rezultat:	Wartość współczynnika typu *double*.

GetOutput	
Wywołanie:	CALL("bckp";"GetOutput";"EI";nr_wyjścia)
Parametry:	nr_wyjścia - numer wyjścia sieci; pierwsze wyjście ma wartość 0
Funkcja zwraca bieżącą wartość wskazanego wyjścia sieci.	
Rezultat:	Wyjście sieci typu *double*.

GetOutputIndex	
Wywołanie:	CALL("bckp";"GetOutputIndex";"MC";nazwa_wyjścia)
Parametry:	nazwa_wyjścia - pełna nazwa wyjścia sieci UWAGA: jeżeli wyjście operuje na wartościach symbolicznych należy pamiętać, że pełna nazwa wyjścia jest złożeniem nazwy podstawowej oraz nazwy wartości symbolicznej np. jeżeli wyjście nazywa się #płeć to sieć neuronowa będzie miała dwa wyjścia: #płeć_kobieta oraz #płeć_mężczyzna
Funkcja zwraca numer wyjścia, którego nazwę przekazuje się jako parametr.	
Rezultat:	Numer wyjścia jako liczba *int*.

GetOutputSize	
Wywołanie:	CALL("bckp";"GetOutputSize";"I")
Parametry:	brak
Funkcja zwraca rozmiar wyjścia sieci.	
Rezultat:	Rozmiar wyjścia jako liczba *int*.

GetPatternNumber	
Wywołanie:	CALL("bckp";"GetPatternNumber";"J")
Parametry:	brak
Funkcja zwraca numer wzorca, który podawany jest na wejście sieci.	
Rezultat:	Numer wzorca jako liczba *long*.

GetTarget	
Wywołanie:	CALL("bckp";"GetTarget";"EI";nr_wyjścia)
Parametry:	nr_wyjścia - numer wyjścia sieci
Funkcja zwraca wartość oczekiwaną wskazanego wyjścia sieci.	
Rezultat:	Wartość oczekiwana jako liczba *double*.

GetTstNumber	
Wywołanie:	CALL("bckp";"GetTstNumber";"EIC";numer_wiersza;nazwa_kolumny)
Parametry:	numer_wiersza - numer wiersza pliku testowego; numer wiersza jest jednocześnie numerem wzorca podawanego na wejście sieci, dopuszczalna wartość to liczba całkowita większa od 0 nazwa_kolumny - nazwa kolumny w pliku testowym
Funkcja zwraca wartość numeryczną, która znajduje się w podanym wierszu i kolumnie pliku testowego sieci neuronowej. Ponieważ numer wiersza jest jednocześnie numerem wzorca to można jako *numer_wiersza* wstawić funkcję GetPatternNumber.	
Rezultat:	Wartość numeryczna typu *double*.

GetTstQuantity	
Wywołanie:	CALL("bckp";"GetTstQuantity";"J")
Parametry:	brak
Funkcja zwraca ilość wzorców testowych w pliku testowym.	
Rezultat:	Liczba wzorców jako liczba *long*.

GetTstRMS	
Wywołanie:	CALL("bckp";"GetTstRMS";"E")
Parametry:	brak
Funkcja zwraca bieżącą wartość błędu RMS testowania.	
Rezultat:	Wartość błędu jako liczba *double*.

GetTstSymbol	
Wywołanie:	CALL("bckp";"GetTstSymbol";"CIC";numer_wiersza;nazwa_kolumny)
Parametry:	numer_wiersza - numer wiersza pliku testowego; numer wiersza jest jednocześnie numerem wzorca podawanego na wejście sieci; dopuszczalna wartość to liczba całkowita większa od 0 nazwa_kolumny - nazwa kolumny w pliku testowym
Funkcja zwraca wartość symboliczną, która znajduje się w podanym wierszu i kolumnie pliku testowego sieci neuronowej. Ponieważ numer wiersza jest jednocześnie numerem wzorca to można jako *numer_wiersza* wstawić funkcję GetPatternNumber.	
Rezultat:	Wartość symboliczna typu *char**.

GetTstTol	
Wywołanie:	CALL("bckp";"GetTstTol";"E")
Parametry:	brak
Funkcja zwraca bieżącą wartość tolerancji testowania.	
Rezultat:	Wartość tolerancji jako liczba *double*.

GetWeight	
Wywołanie:	CALL("bckp";"GetWeight";"EIII"; nr_dol_neur; nr_gór_neur; warstwa)
Parametry:	warstwa - numer warstwy w której znajduje się neuron o numerze *nr_gór_neur* nr_gór_neur - numer neuronu górnego, który znajduje się w warstwie *warstwa* nr_dol_neur - numer neuronu dolnego, który znajduje się w warstwie o poprzedzającej warstwę *warstwa*.
Funkcja zwraca wartość wagi pomiędzy dwoma neuronami. Pierwszy z nich zlokalizowany jest w warstwie *warstwa,* natomiast drugi znajduje się w warstwie niższej.	
Rezultat:	Wartość wagi jako liczba *double*.

7

Wykresy

W rozdziale:

- Jak utworzyć nowy wykres
- Otwieranie wykresu już istniejącego
- Parametry wykresu dostępne dla użytkownika

Cenne informacje

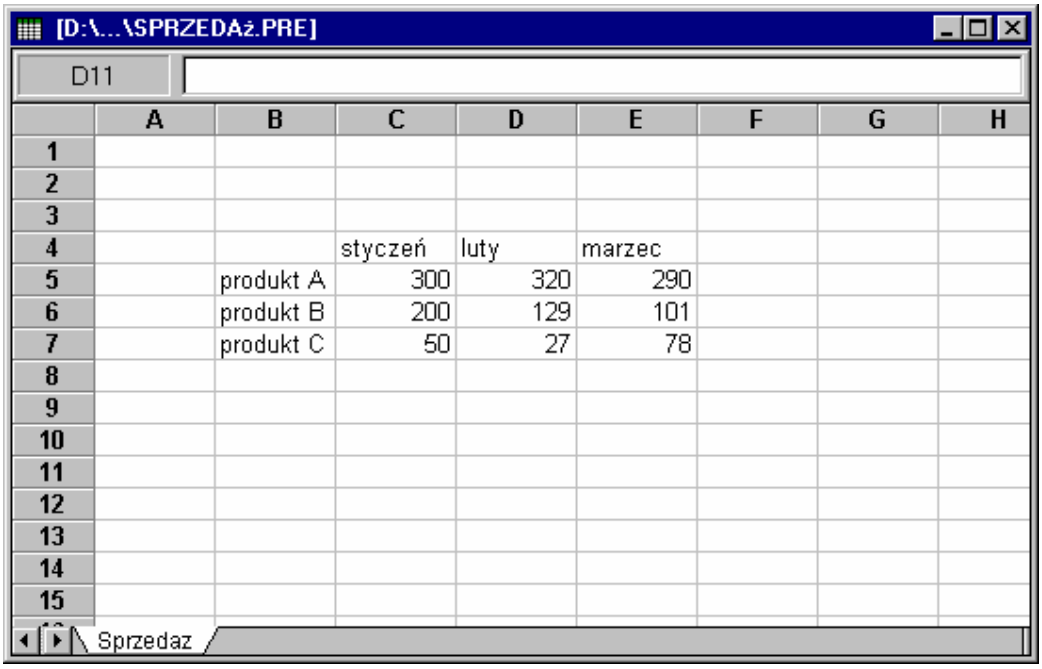
Ćwiczenia z komputerem

Koniecznie przeczytaj

JAK UTWORZYĆ NOWY WYKRES ?

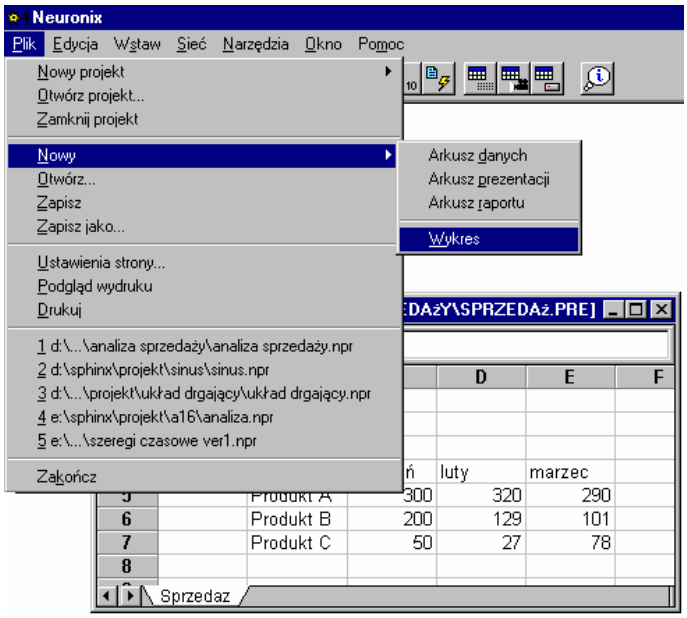
W celu utworzenia wykresu należy wpisać do arkusza kalkulacyjnego dane, które mają zostać zobrazowane. W przykładzie zostanie utworzony wykres trójwymiarowy, reprezentujący miesięczną sprzedaż wybranej grupy produktów.

Przykład 7-1



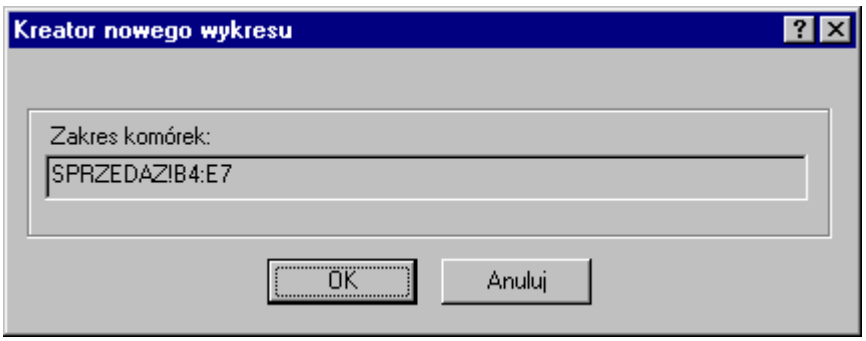
Rys. 56. Dane dla wykresu

Po wpisaniu przykładowych danych w arkusz prezentacji, należy zaznaczyć za pomocą myszki lub klawiatury zakres do przedstawienia na wykresie. Po zaznaczeniu obszaru należy odszukać w menu głównym w grupie **Plik** opcję **Nowy wykres**, co pozwoli na uruchomienie kreatora nowych wykresów.



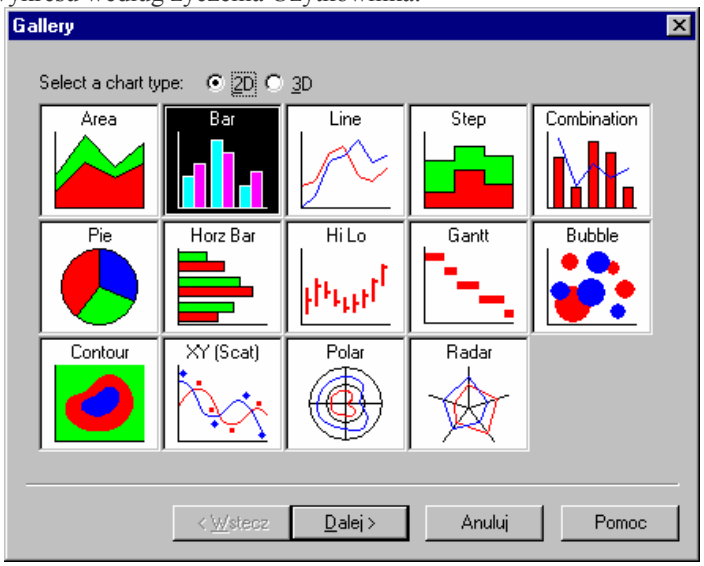
Rys. 57. Opcja tworzenia nowego wykresu

Po uruchomieniu kreatora pojawia się dialog informacyjny (rys. 58), zawierający nazwę arkusza źródłowego oraz zakres danych.

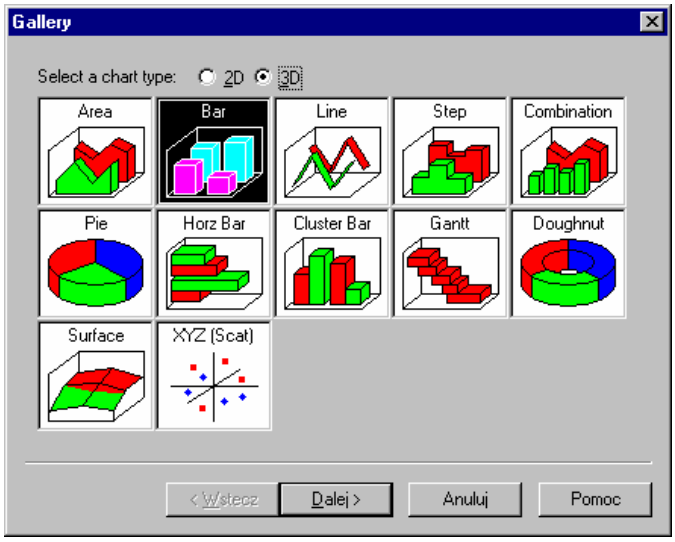


Rys. 58. Informacja o połączeniu wykresu z arkuszem

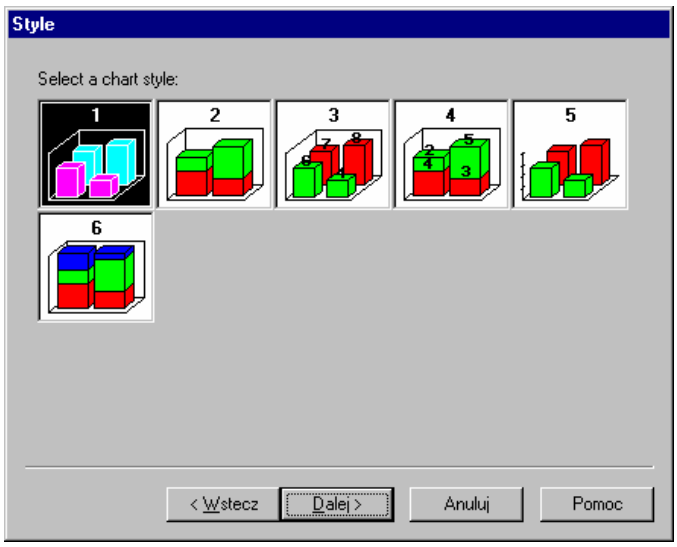
Po zatwierdzeniu zakresu danych, który pojawia się w dialogu informacyjnym, użytkownik ma do wyboru kilka typów wykresów dwuwymiarowych i trójwymiarowych (rys. 59). Kolejne rysunki przedstawiają kroki w celu wybrania i dostosowania wykresu według życzenia Użytkownika.



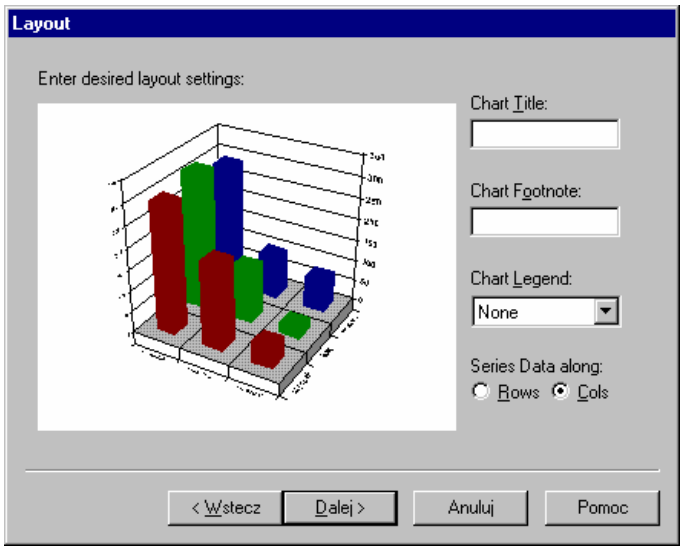
Rys. 59. Kreator wykresu 2-d



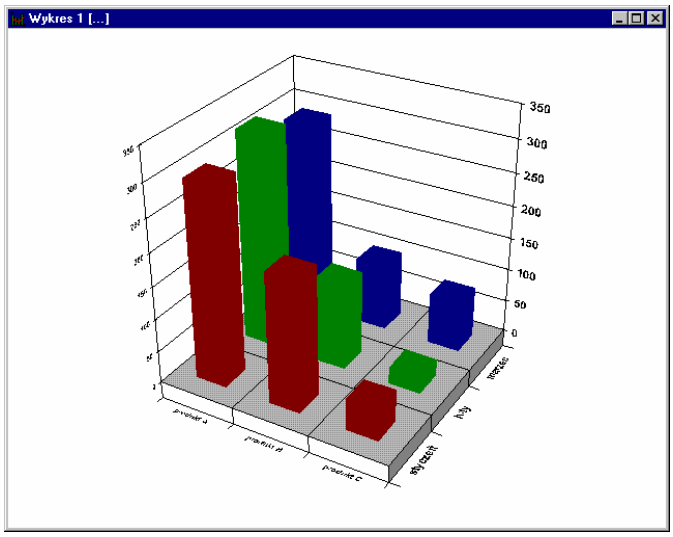
Rys. 60. Kreator wykresu 3-d



Rys. 61. Przykładowy wybór stylów dla trójwymiarowego wykresu słupkowego



Rys. 62. Orientacyjny podgląd wykresu w dialogu kreatora



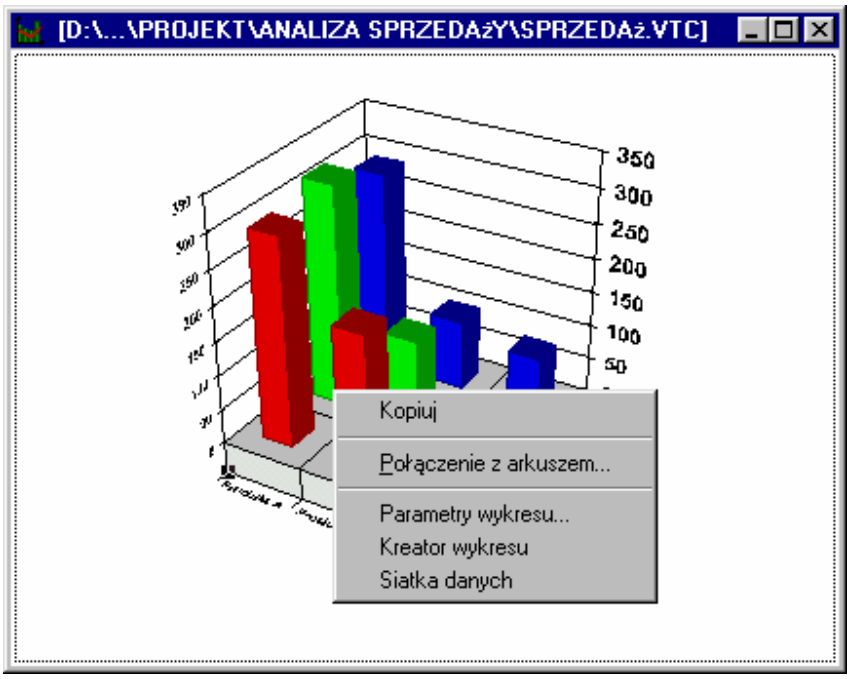
Rys. 63. Ostateczny wygląd wykresu

OTWIERANIE WYKRESU JUŻ ISTNIEJĄCEGO

W celu otwarcia wykresu utworzonego wcześniej i zapisanego na dysku z rozszerzeniem ***.vtc** należy w menu głównym **Plik** użyć opcji **Otwórz** i następnie wybrać w dialogu obsługi plików rozszerzenie odpowiadające wykresom ***.vtc**.

PARAMETRY WYKRESU DOSTĘPNE DLA UŻYTKOWNIKA

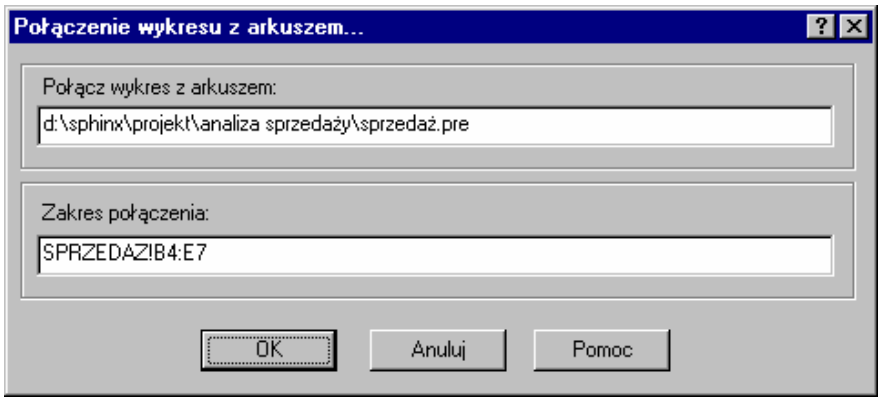
Dla wykresu, który został nowo utworzony lub otwarty za pomocą opcji **Otwórz** można przeprowadzić korektę ustawień głównych parametrów wykresu.



Rys. 64. Menu podręczne wykresu

POŁĄCZENIE Z ARKUSZEM

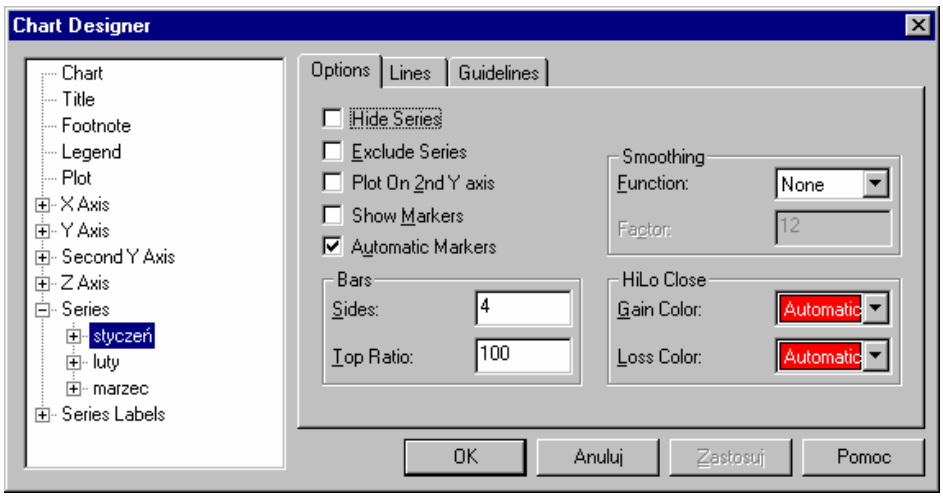
Opcja pozwala na kontrolę poprawności oraz edycję parametrów połączenia pomiędzy arkuszem źródłowym i wykresem, który wizualizuje pewną część danych zawartych w tym arkuszu.



Rys. 65. Dialog dla opcji **Połączenie wykresu z arkuszem**

PARAMETRY WYKRESU

Opcja pozwala na dowolne kształtowanie parametrów wykresu oraz uzupełnianie ustawień domyślnych, które zostały wygenerowane za pomocą **Kreatora nowego wykresu**.



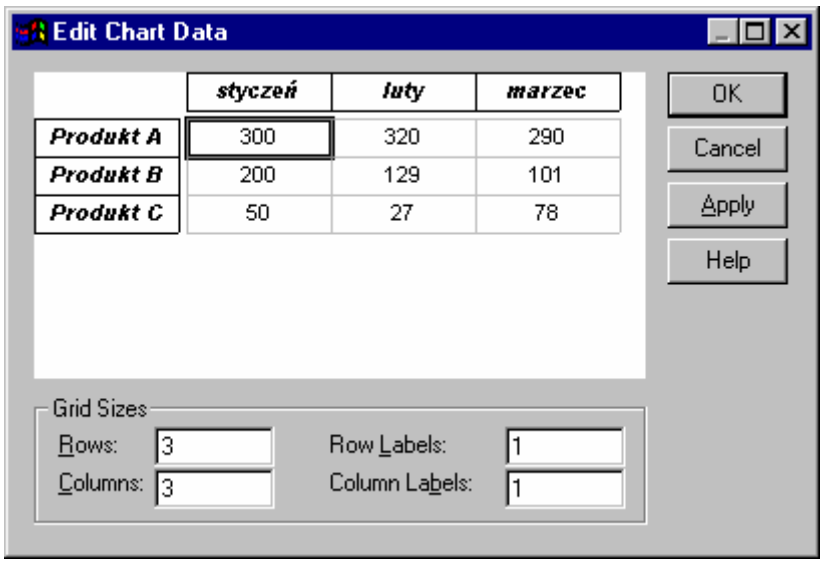
Rys. 66. Dialog kontroli wszystkich parametrów wykresu

KREATOR WYKRESU

Opcja uruchamia dialog budowy nowego wykresu. Zasadniczą różnicą w porównaniu z opcją z menu głównego **Kreator nowego wykresu** jest możliwość zachowania fizycznej struktury wykresu na dysku pod tą sama nazwą i dla tych samych danych w arkuszu źródłowym. Opcja zalecana dla Użytkowników mniej doświadczonych, który mieliby problemy konfiguracji wykresu za pomocą opcji **Parametry wykresu**.

SIATKA DANYCH

Siatka danych zawiera wartości komórek w postaci reprezentowanej na wykresie wraz z nazwami osi układu współrzędnych. Istnieje możliwość skopiowania siatki do arkusza opcją **Kopiuj** z menu podręcznego wykresu. Dane zawarte w siatce można edytować i zmieniać ich wartość. Możliwe jest utworzenie wykresu tylko na podstawie danych zawartych w siatce. Ilość kolumn oraz wierszy siatki ogranicza zakres danych widocznych na wykresie.



Rys. 67. Dialog edycji danych zawartych w siatce

8

Obsługa wydruków w systemie Neuronix

W rozdziale:

- Drukowanie zawartości arkuszy
- Drukowanie wykresów
- Drukowanie topologii sieci neuronowej

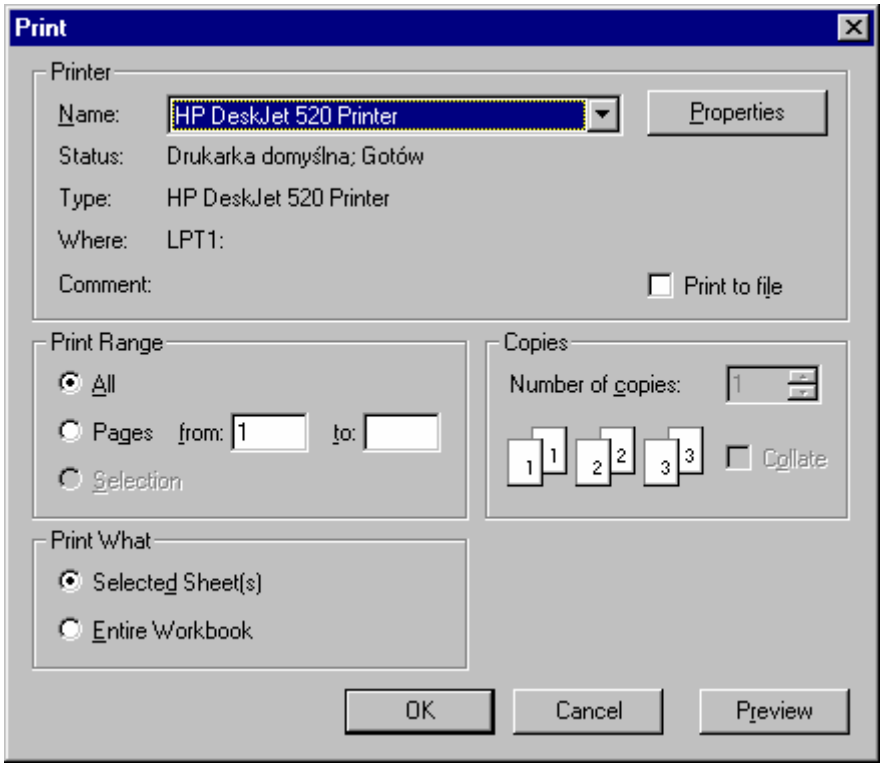
Cenne informacje

Ćwiczenia z komputerem

Koniecznie przeczytaj

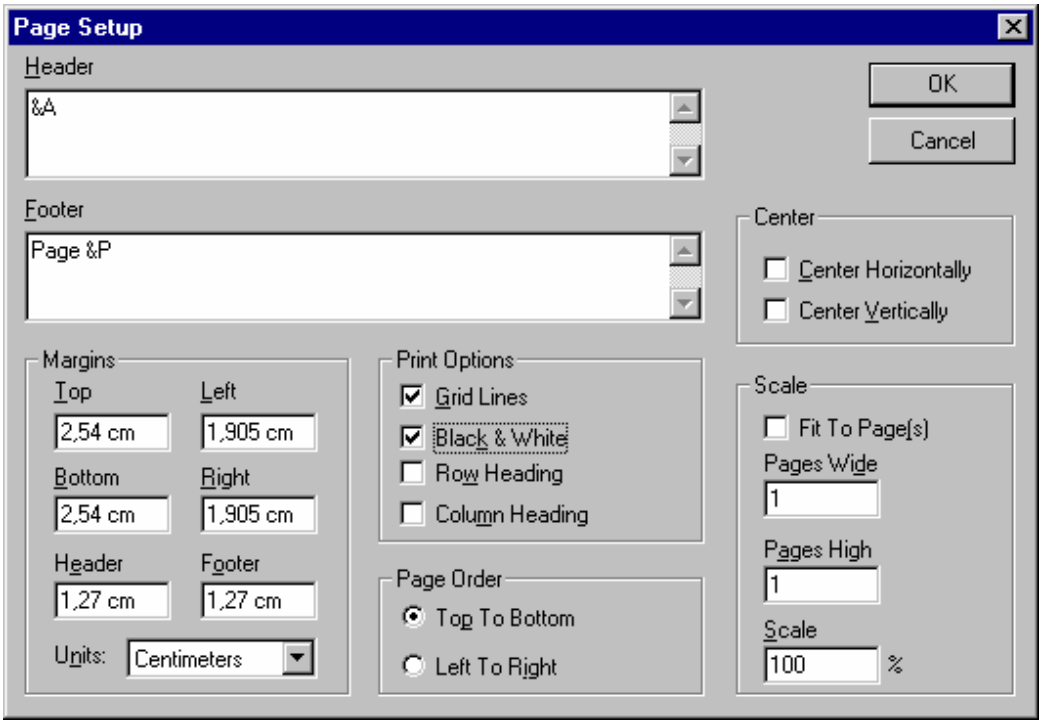
DRUKOWANIE ZAWARTOŚCI ARKUSZY

W celu wydrukowania zawartości arkusza należy wybrać opcję **Drukuj** dostępną w menu głównym, w grupie **Plik**. Po wybraniu opcji pojawia się dialog konfiguracyjny wydruku zawierający podstawowe parametry związane z obsługą drukarki.



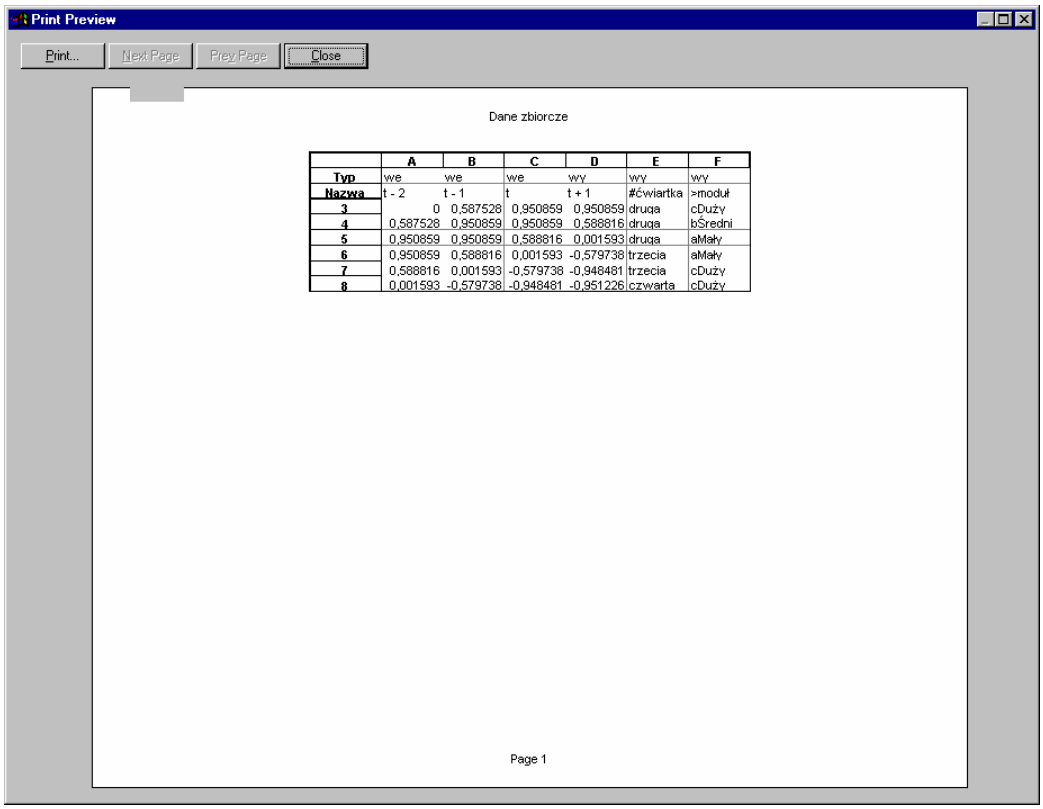
Rys. 68. Dialog obsługi wydruku arkusza kalkulacyjnego

W celu określenia nietypowych ustawień parametrów drukowanego arkusza kalkulacyjnego konieczne jest użycie opcji **Ustawienia strony** z grupy **Plik** menu głównego.



Rys. 69. Dialog obsługi ustawień arkusza kalkulacyjnego

Arkusz kalkulacyjny przeznaczony do wydruku można podglądać za pomocą opcji **Podgląd strony** z menu głównego **Plik.**

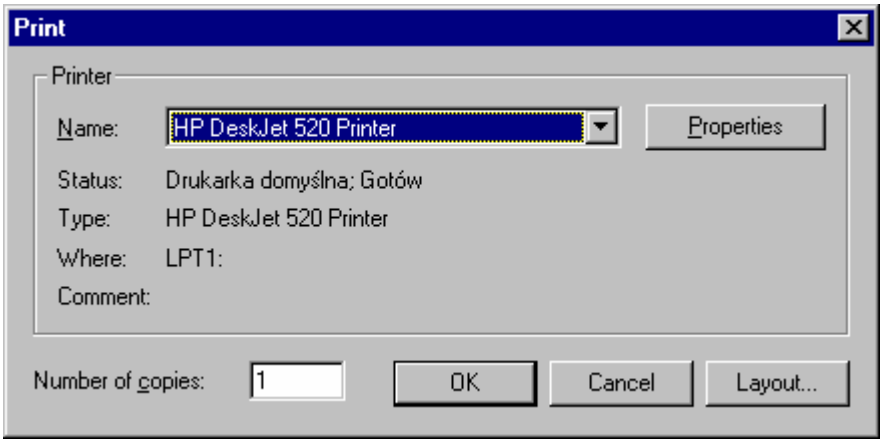


	A	B	C	D	E	F
Typ	we	we	we	wy	wy	wy
Nazwa	t - 2	t - 1	t	t + 1	#ćwiartka	>moduł
3	0	0,587528	0,950859	0,950859	druga	cDuży
4	0,587528	0,950859	0,950859	0,588816	druga	bŚredni
5	0,950859	0,950859	0,588816	0,001593	druga	aMały
6	0,950859	0,588816	0,001593	-0,579738	trzecia	aMały
7	0,588816	0,001593	-0,579738	-0,948481	trzecia	cDuży
8	0,001593	-0,579738	-0,948481	-0,951226	czwarta	cDuży

Rys. 70. Podgląd strony dla arkusza kalkulacyjnego

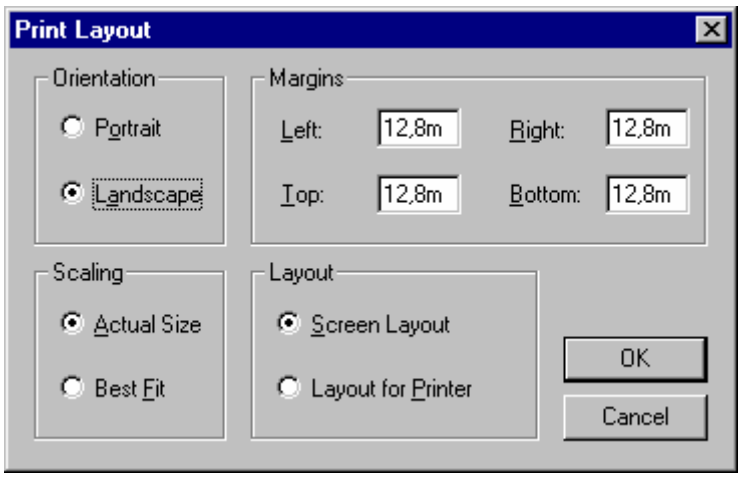
DRUKOWANIE WYKRESÓW

W celu wydrukowania wykresu należy wybrać opcję **Drukuj** dostępną w menu głównym, w grupie **Plik**. Po wybraniu opcji pojawia się dialog konfiguracyjny wydruku zawierający podstawowe parametry związane z obsługą drukarki.



Rys. 71. Dialog obsługi wydruku wykresu

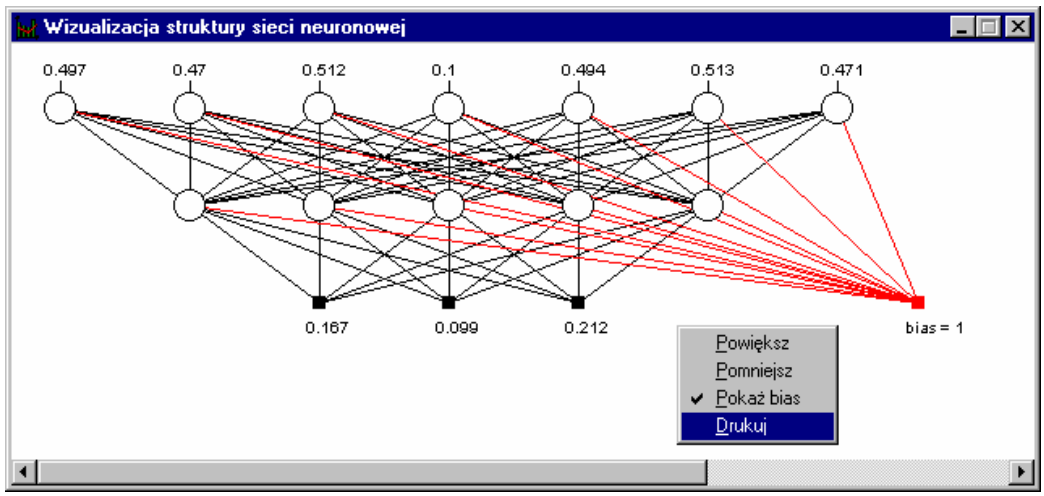
W celu określenia nietypowych ustawień parametrów drukowanego wykresu konieczne jest użycie opcji **Ustawienia strony** z grupy **Plik** menu głównego.



Rys. 72. Dialog obsługi ustawień wykresu

DRUKOWANIE TOPOLOGII SIECI NEURONOWEJ

W celu wydrukowania topologii sieci należy wybrać opcję **Drukuj** dostępną w menu podręcznym, które można wyświetlić poprzez kliknięcie prawym przyciskiem myszy na oknie **Wizualizacja topologii sieci**. Po wybraniu tej opcji następuje proces automatycznego drukowania zakresu obrazu widocznego w oknie. W celu wydrukowania dowolnego zakresu topologii sieci neuronowej należy użyć opcji **Powiększ** lub **Pomniejsz** z menu podręcznego oraz suwaków poziomego i pionowego, dla dostosowania widocznego fragmentu obrazu.



Rys. 73. Drukowanie struktury sieci neuronowej

9

Co potrafi sieć neuronowa ?

W rozdziale:

- Klasyfikacja i rozpoznawanie wzorców obrazów
- Aproksymacja
- Kodowanie
- Odtwarzanie "zaszumionych" wzorców

Cenne informacje

Ćwiczenia z komputerem

Koniecznie przeczytaj

KLASYFIKACJA I ROZPOZNAWANIE WZORCÓW

Zdolność sieci do klasyfikacji można przedstawić na przykładzie funkcji AND oraz XOR. Dwa przykłady pozwalają zorientować się również w podstawowych problemach i trudnościach stosowania sieci neuronowej do rozwiązywania **zadań klasyfikacji**. Omawiane przykłady znajdują się w katalogach **../Projekt/AND/and.npr** oraz **../Projekt/XOR/xor.npr.**

FUNKCJA AND

Naszym zadaniem będzie wyuczenie sieci rozpoznawania wzorca zero-jedynkowego dla różnych jego kombinacji. Tabelka przejścia dla funkcji AND przedstawia się następująco:

we1	we2	wy
0	0	0
0	1	0
1	0	0
1	1	1

Rys. 74. Tablica przejścia funkcji AND

Na podstawie tabeli wartości wejścia-wyjścia funkcji AND przedstawionej na rys. 76, tworzymy zbiór uczący w arkuszu danych. Po utworzeniu zbioru uczącego, który powinien znajdować się w pliku o rozszerzeniu *.lrn, należy przystąpić do uczenia sieci. Zależność, która opisuje wyjście sieci dla naszego przykładu jest następująca:

(9-1) $$y = w_{1,1} \cdot x_1 + w_{1,2} \cdot x_2 + b \, .$$

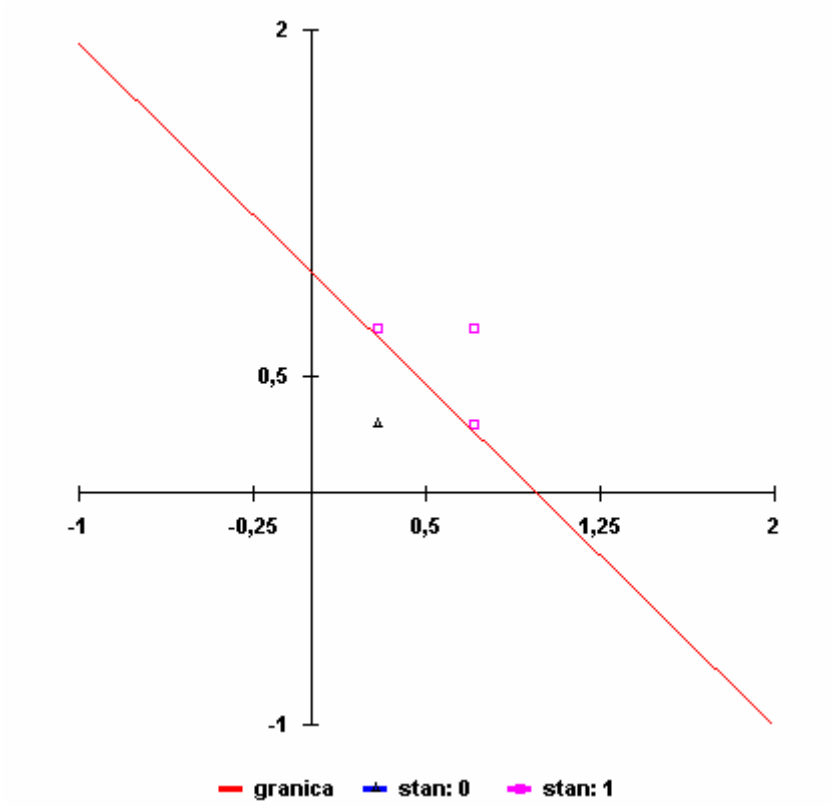
Równanie służące obliczeniu wartości wyjścia sieci dla dwóch wejść, jest równaniem prostej na płaszczyźnie, które przedstawia się następująco:

(9-2) $$A \cdot x + B \cdot y + C = 0 \, .$$

W celu wykreślenia prostej w układzie kartezjańskim, czyli po prostu zwykłym układzie współrzędnych xy, należy dokonać pewnych przekształceń współczynników,

(9-3) $$y = m \cdot x + k$$
$$y = -\frac{w_{1,1}}{w_{1,2}} \cdot x - \frac{b}{w_{1,2}} \, .$$

Teraz możemy wykreślić prostą na wykresie dostępnym w systemie Neuronix. Prosta ta wyznacza granicę podziału zbioru uczącego na dwa podzbiory. Pamiętamy, że wartości bezpośrednio na wejściach sieci neuronowej są **odwzorowaniem** wartości, które wpisane są do arkusza danych, ale nie są im równe. Sytuacja taka wynika, z zastosowania w systemie Neuronix **automatycznego układu skalującego** wartości wejściowe i wyjściowe sieci do przedziału <0,1> i dodatkowo narzucenia ograniczenia w celu uniknięcia przesterowania wejść neuronów do wartości <0.297, 0.781>. Warości wejścia w arkuszu danych równej 0 odpowiada wartość bezpośrednio na wejściu sieci równa 0.291, a wartości 1 odpowiada wartość 0.781. Dlatego takie wartości zostały wykorzystane w przykładzie i znajdują się w arkuszu prezentacji w celu wizualizacji punktów w układzie współrzędnych.

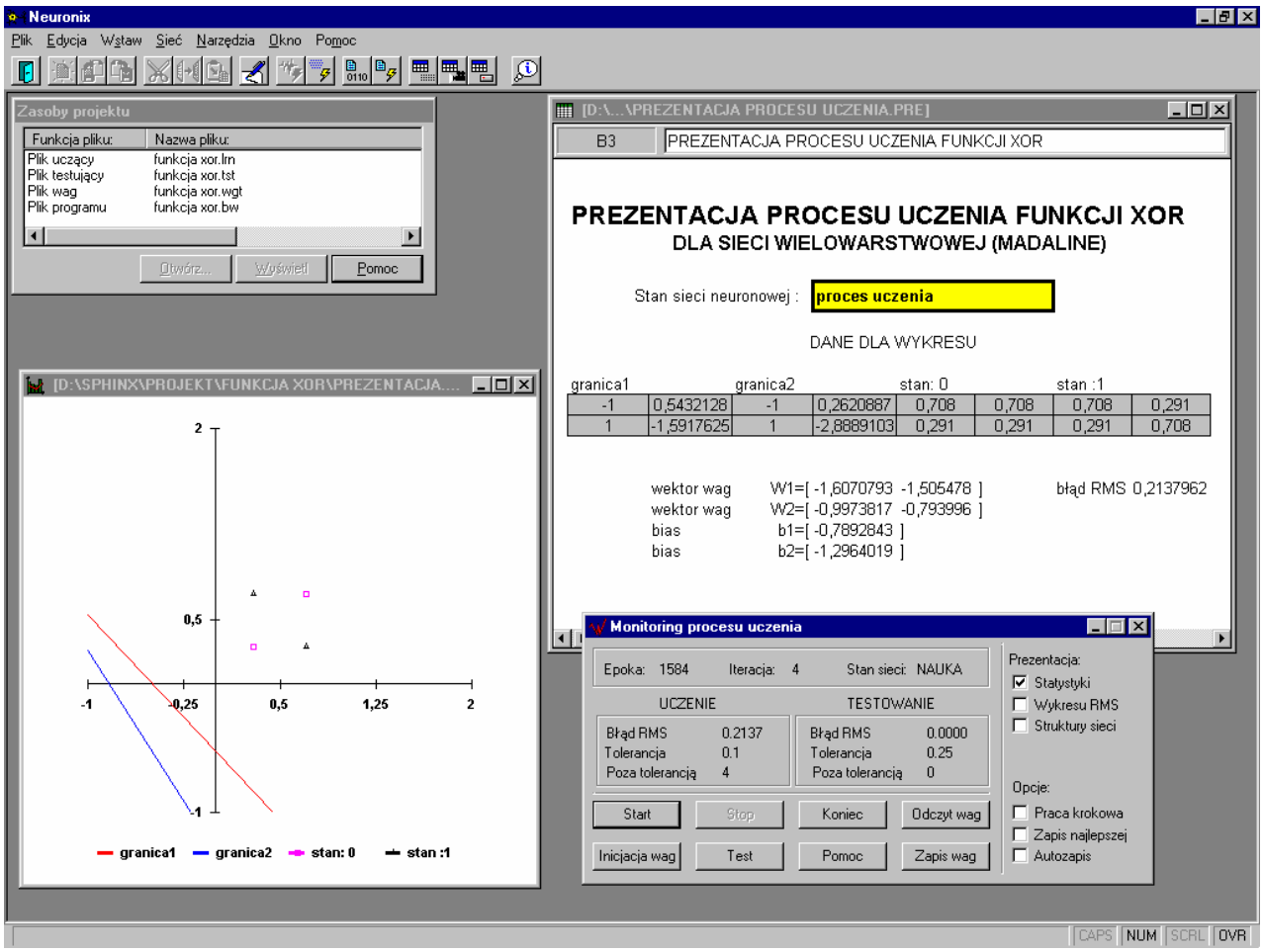


FUNKCJA XOR

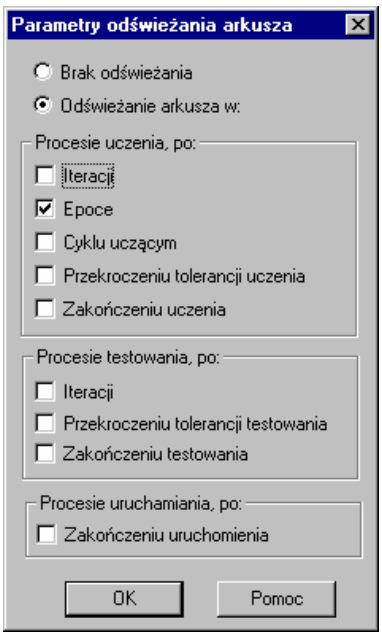
Przykład funkcji XOR jest zbliżony do poprzedniego przykładu, rożnica polega na konieczności użycia sieci wielowarstwowej, która jest w stanie rozwiązać problem klasyfikacji dla funkcji XOR. Uczenie sieci neuronowej funkcji XOR jest możliwe wtedy, gdy sieć posiada przynajmniej jedną warstwę ukrytą.

we1	we2	wy
0	0	0
0	1	1
1	0	1
1	1	0

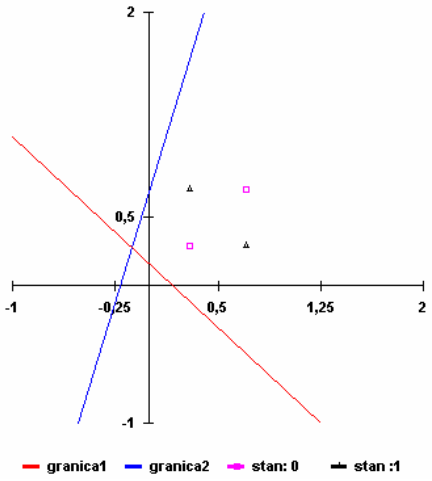
Rys. 75. Tablica przejścia funkcji XOR



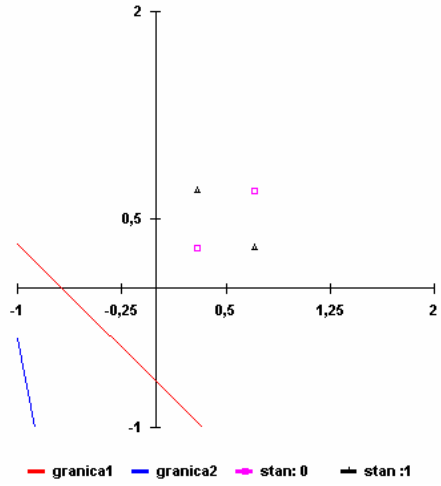
Rys. 76. Projekt neuronowy funkcji XOR



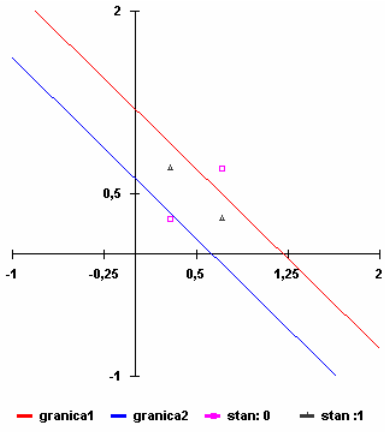
Rys. 77. Ustawienia parametrów odświeżania arkusza prezentacji



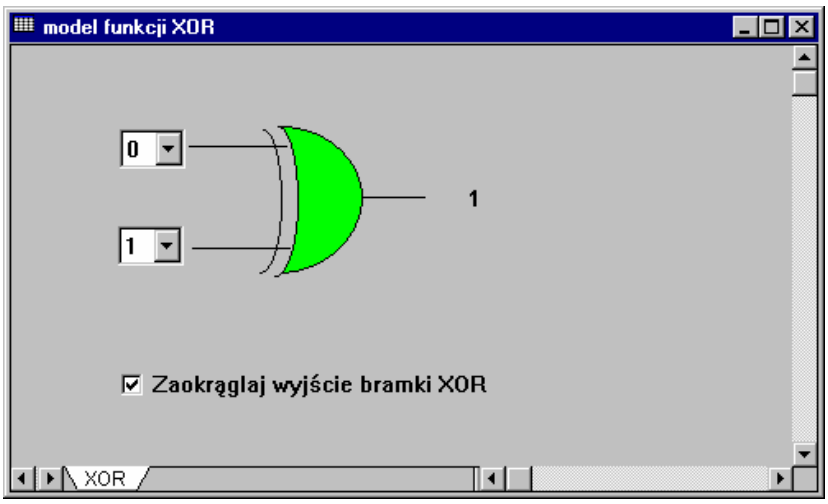
Rys. 78. Początkowy etap uczenia sieci



Rys. 79. Sieć podczas procesu uczenia



Rys. 80. Sieć wyuczona

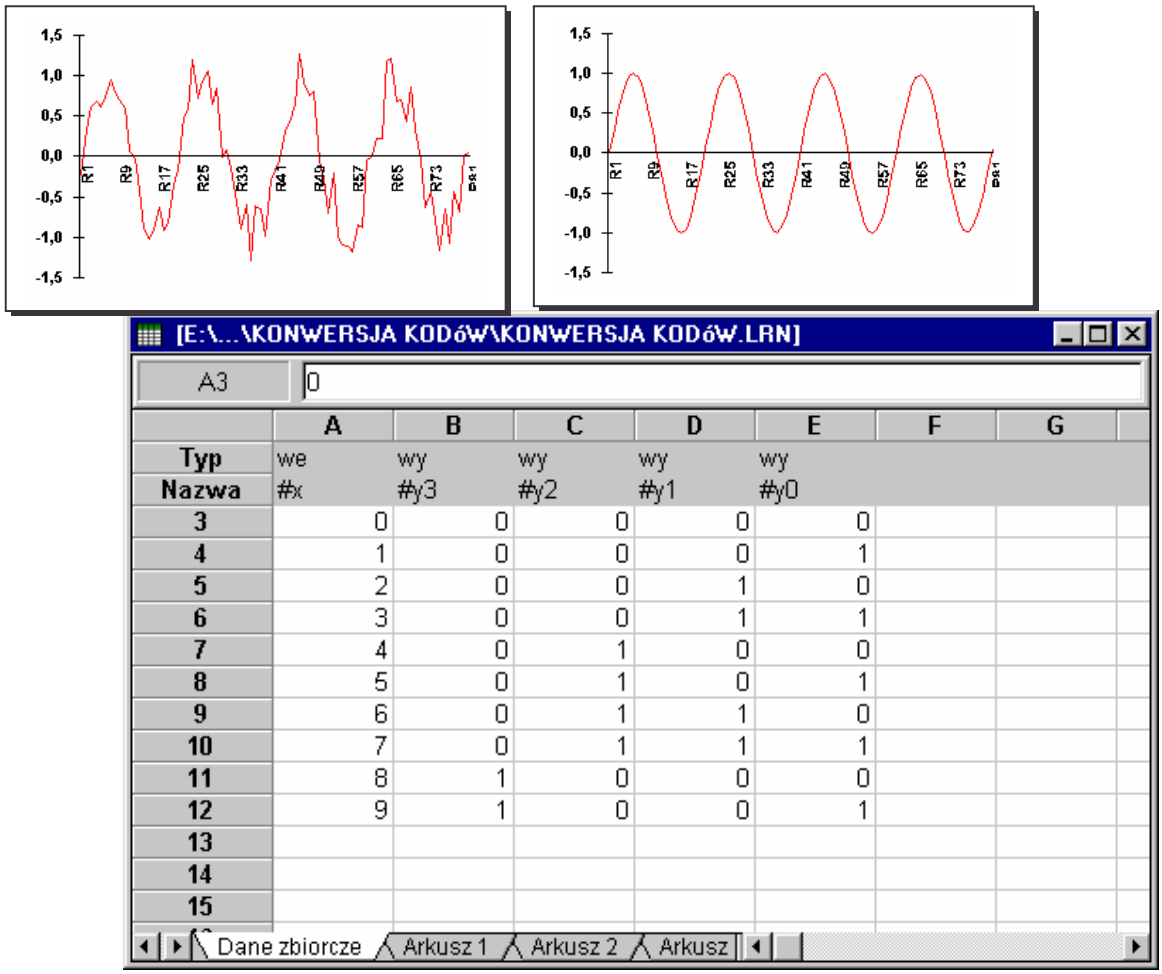


Rys. 81. Wizualizacja modelu funkcji XOR

Rys. 82. Opcja uaktualniania wyjścia modelu

KODOWANIE

Konwerter kodu symuluje działanie rzeczywistego układu, który mógłby zostać wykonany np. w technice cyfrowej w postaci układ elektroniczego. Jego zadaniem jest poprawne przekształcenie kodu dziesiętnego, na równoważny kod w systemie dwójkowym. W celu realizacji tego zadania sieć została wyuczona na zbiorze uczącym zawierającym wartości elementów kodu w postaci danych symbolicznych. Gotowy projekt jest dostępny w katalogu **../Projekt/Konwerter kodów/kody.npr**. Arkusz z danymi przedstawia rys. 83.



Rys. 83. Arkusz z konwertowanymi kodami

APROKSYMACJA

Sieć neuronowa jest uniwersalnym aproksymatorem funkcji wielu zmiennych. Większość zagadnień modelowania, można sprowadzić do problemu aproksymacji funkcji w przestrzeni wielowymiarowej. Możliwe jest poprzez odpowiednie wyuczenie sieci uzyskanie układu interpolującego funkcję nieliniową w przestrzeni wielowymiarowej.

ODTWARZANIE "ZASZUMIONYCH" WZORCÓW

Transmisja sygnałów na większe odległości z reguły narażona jest na różnego rodzaju zakłócenia. W takich przypadkach, aby uzyskać interesujący nas sygnał użyteczny i jednocześnie poprawnie odtworzyć przesyłane w ten sposób informacje konieczne jest stosowanie filtrów umieszczonych przed odbiornikiem.

Rys. 84. Koncepcja przesyłu sygnałów

Proponowane przez nas filtry opierają się na sieciach neuronowych o odpowiedniej strukturze. Takie właśnie rozwiązanie uwalnia nas od skomplikowanych obliczeń (między innymi estymacji parametrów przesyłanego sygnału), zwiększa szybkość przetwarzania i powoduje znaczne uproszczenie problemu sprowadzając nasz filtr do postaci

„czarnej skrzynki”. W praktyce największe zastosowanie mają sygnały prostokątny, trójkątny (piłokształtny) i sinusoidalny, zatem przedstawione poniżej przykłady dotyczą tylko tych trzech przypadków. W przypadku **przebiegu prostokątnego** została zaprojektowana i wyuczona sieć neuronowa o jednym wejściu i jednym wyjściu. Żeby z niej skorzystać należy otworzyć projekt o nazwie: **..Projekt/Prostokąt/prost.npr**, zainicjować sieć, odczytać wagi, wprowadzić dane do arkusza jak na rys.85.

W tej kolumnie wpisujemy dane wejściowe

A3 -0.2248

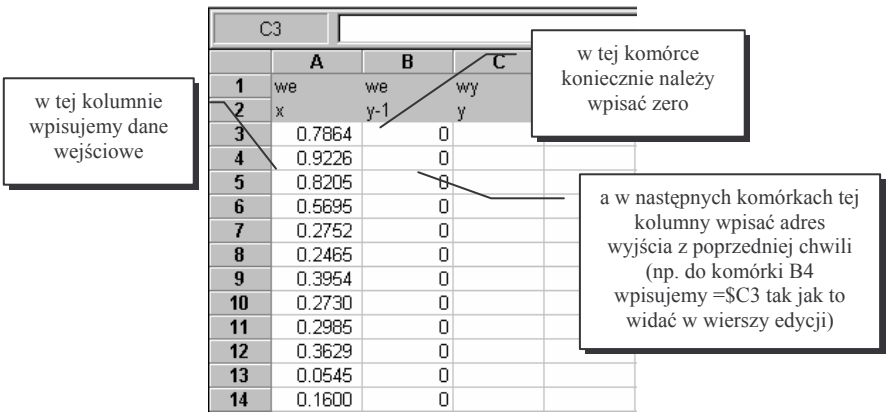
	A	B	C
1	we	wy	
2	x	y	
3	-0.2248		
4	0.3328		
5	0.6093		
6	0.6716		
7	0.6092		
8	0.7332		
9	0.9462		
10	0.7806		
11	0.6914		
12	0.5858		
13	0.0573		
14	-0.0354		

Rys. 85. Arkusz wypełniony danymi wejściowymi

Następnie należy uruchomić sieć korzystając z podręcznego menu.

Rys. 86. Przykład filtracji przebiegu prostokątnego

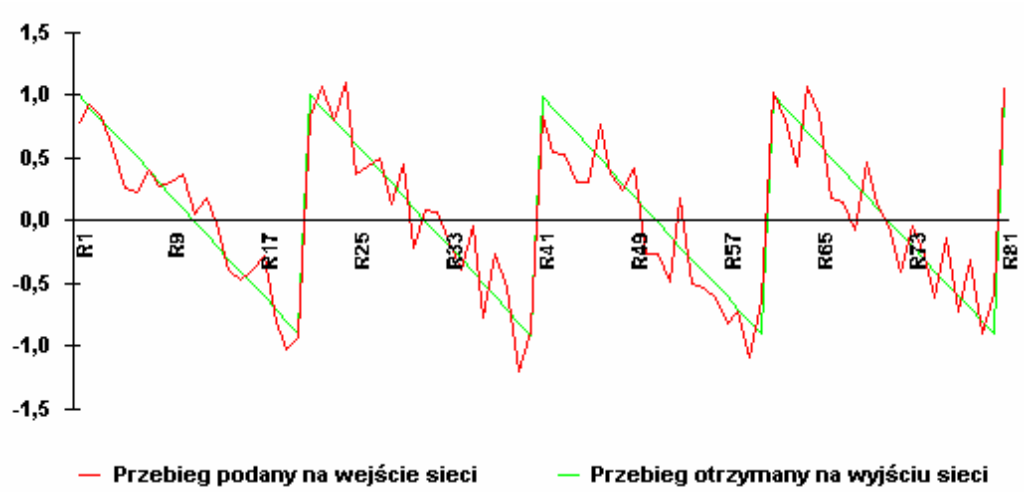
W przypadku przebiegu trójkątnego została zaprojektowana i wyuczona sieć neuronowa o dwóch wejściach i jednym wyjściu. Przy czym na jedno z wejść podajemy zaszumiony sygnał, a na drugie sygnał wyjściowy z poprzedniej chwili. Aby skorzystać z tej sieci należy otworzyć projekt o nazwie **../Projekt/Trójkąt/trojkat.npr**, zainicjować sieć, odczytać wagi, wprowadzić dane do arkusza jak na rys. 87.



	A	B	C
1	we	we	wy
2	x	y-1	y
3	0.7864	0	
4	0.9226	0	
5	0.8205	0	
6	0.5695	0	
7	0.2752	0	
8	0.2465	0	
9	0.3954	0	
10	0.2730	0	
11	0.2985	0	
12	0.3629	0	
13	0.0545	0	
14	0.1600	0	

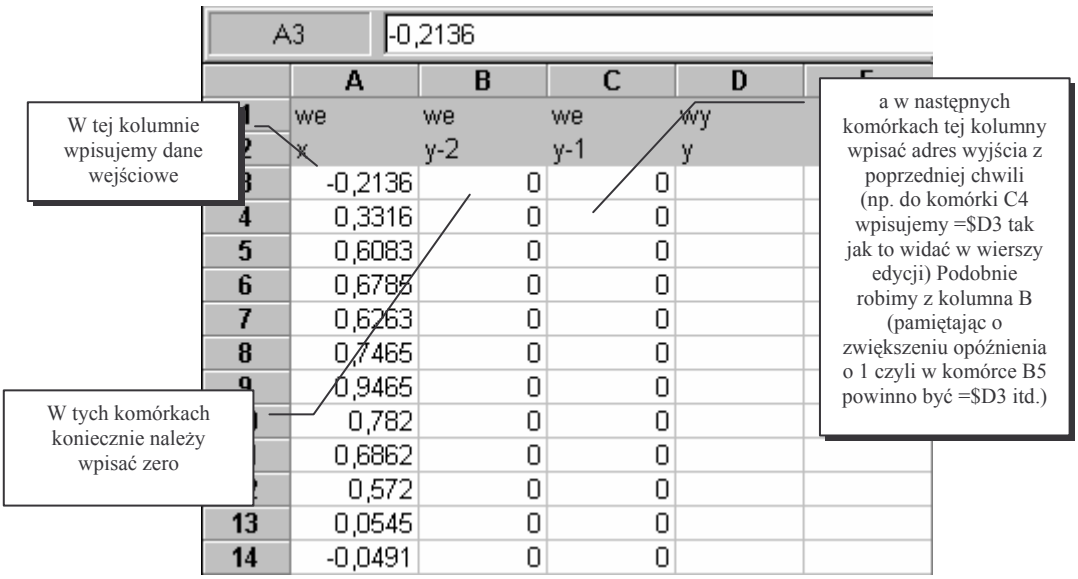
Rys. 87. Arkusz z danymi

Po wprowadzeniu danych do kolumny „x” oraz adresów do kolumny „y-1” (obie kolumny powinny zawierać tyle samo elementów) należy z podręcznego menu uruchomić sieć. Po tej czynności w kolumnie oznaczonej „y” otrzymamy sygnał z wyjścia sieci, a w kolumnie „y-1” pojawi się ten sam sygnał, ale opóźniony o jedną iterację.



Rys. 88. Przykład filtracji przebiegu trójkątnego

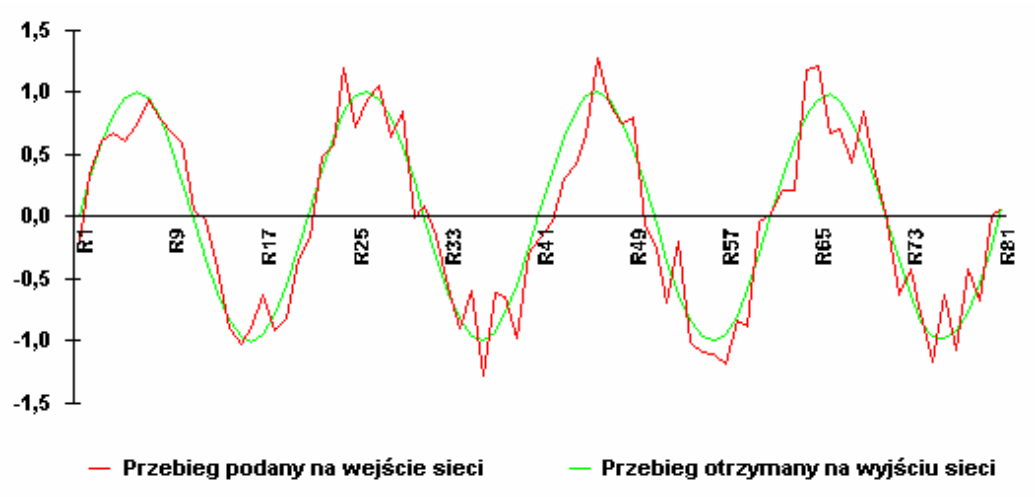
W przypadku przebiegu sinusoidalnego została zaprojektowana i wyuczona sieć neuronowa o trzech wejściach i jednym wyjściu. Tutaj na jedno z wejść podajemy zaszumiony sygnał, na drugie sygnał wyjściowy z poprzedniej chwili (opóźniony o jedną iterację) natomiast na wejście trzecie sygnał wyjściowy opóźniony o dwie iteracje. Aby skorzystać z tej sieci należy otworzyć projekt o nazwie sinus, zainicjować sieć, odczytać wagi, wprowadzić dane do arkusza w następujący sposób:



	A	B	C	D
1	we	we	we	wy
2	x	y-2	y-1	y
3	-0,2136	0	0	
4	0,3316	0	0	
5	0,6083	0	0	
6	0,6785	0	0	
7	0,6263	0	0	
8	0,7465	0	0	
9	0,9465	0	0	
10	0,782	0	0	
11	0,6862	0	0	
12	0,572	0	0	
13	0,0545	0	0	
14	-0,0491	0	0	

Rys. 89. Arkusz z danymi

Po wprowadzeniu danych do kolumny „x” oraz adresów do kolumny „y-1” i „y-2” (wszystkie kolumny powinny zawierać tyle samo elementów) należy z podręcznego menu uruchomić sieć. Po tej czynności w kolumnie oznaczonej „y” otrzymamy sygnał z wyjścia sieci, w kolumnie „y-1” pojawi się ten sam sygnał, ale opóźniony o jedną iterację, natomiast w kolumnie „y-2” pojawi się sygnał opóźniony o dwie iteracje. Przykład filtracji przebiegu sinusoidalnego przedstawia rys. 90.



Rys. 90. Przykład filtracji przebiegu sinusoidalnego

10

W jaki sposób utworzyć model zjawiska ?

W rozdziale:

- Klasy modeli
- Podział modeli
- Jak zbudować model ?
- Modele uogólniające

Cenne informacje

Ćwiczenia z komputerem

Koniecznie przeczytaj

KLASY MODELI

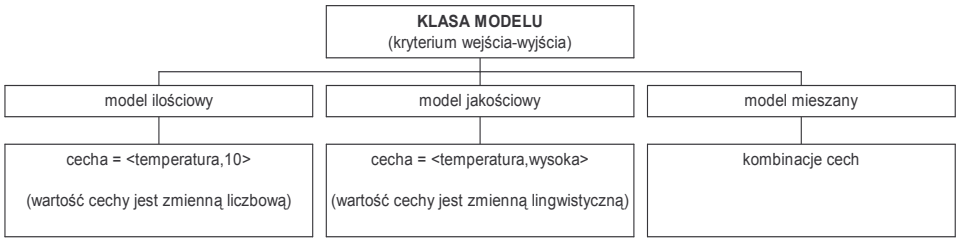
W celu analizy dowolnego systemu, obiektu lub procesu (np. ekonomicznego, społecznego, technicznego) można utworzyć jego model, który stanowi próbę formalizacji zależności, jakie występują pomiędzy jego **wejściami i wyjściami** Istnieje wiele metod opisu tych zależności, wszystkie wywodzą się z trzech możliwych podejść:

- podejścia **ilościowego** wykorzystującego cechy ilościowe,
- podejścia **jakościowego**, wykorzystującego cechy jakościowe,
- podejścia **mieszanego**, łączącego w sobie metody ilościowe i jakościowe opisu zjawisk.

Należy podkreślić, że podejście jakościowe pozwala na zbudowanie dużo lepszych modeli. Wydaje się to pewną sprzecznością, ale należy podkreślić, że w niektórych przypadkach np. analizie zjawisk społecznych, budowa modeli ilościowych prowadzi do znacznego skomplikowania zapisu zależności pomiędzy wejściem i wyjściem modelu. Poszczególne podejścia określają klasę modelu. Dla lepszego zobrazowania różnic pomiędzy klasami modeli należy przypomnieć podstawowe pojęcie cechy:

cecha = < **nazwa atrybutu**, **wartość** >,

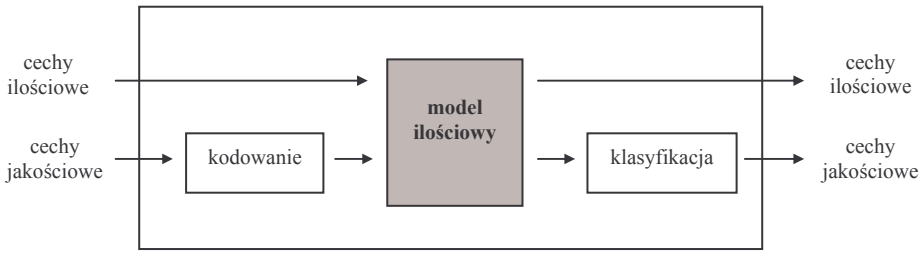
przez pojęcie cechy rozumiemy atrybut przysługujący danemu obiektowi.



Rys. 91. Klasy modeli

Ze względu na **cechy jakościowe,** do budowy modeli jakościowych lub mieszanych mogą zostać użyte metody zbiorów przybliżonych, rozmytych, sieci neuronowych wspomaganych układami kodującymi i klasyfikującymi, regresji uwzględniającej cechy jakościowe.

Modele mieszane mogą posiadać złożoną strukturę, przykład takiego rozwiązania, na którym opiera się koncepcja Neuronix'a przedstawia rys. 92. Rozwiązanie takie zapewnia lepszą jakość analizy złożonych problemów. W układzie kodowania stosuje w zależności od sterowania, kodowanie "**1 z n**" lub "**k z n**".



Rys. 92. Model mieszany

PODZIAŁ MODELI

Można rozpatrywać modele **liniowe** i **nieliniowe. Modele liniowe** są pewnym uproszczeniem rzeczywistości, jednak ze względu na proste (w porównaniu z nieliniowymi) metody ich syntezy są najczęściej używane. Istniej bardzo wiele metod estymacji parametrów modeli liniowych. **Modele nieliniowe** są bardziej złożone i wymagają wykorzystania złożonego aparatu matematycznego celem identyfikacji ich parametrów, pozwalają jednak w sposób bardziej dokładny opisać modelowany obiekt lub proces. Często stosuje się podejście upraszczające zakładając nieliniowy model i linearyzując fragment jego powierzchni, dla którego jest to możliwe. Wśród modeli występują modele **deterministyczne** i **stochastyczne**. W przypadku tych ostatnich, przebieg procesu jest uzależniony od składowej losowej (zmienna losowa). Ponadto istnieją modele **ciągłe** i **dyskretne.**

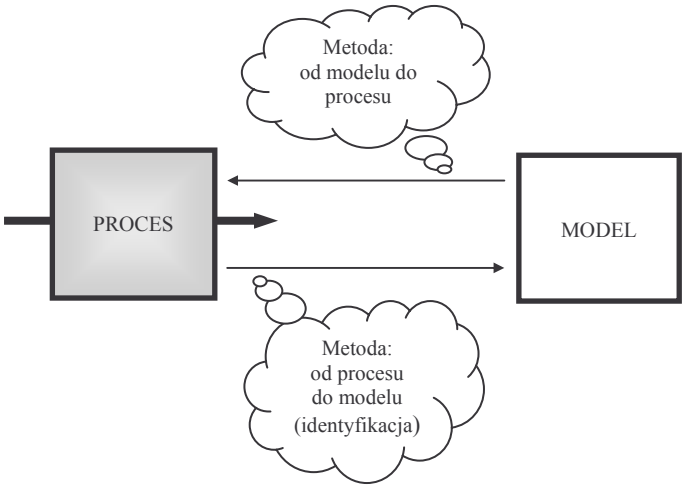
JAK ZBUDOWAĆ MODEL ?

Istnieją dwa sposoby utworzenia modelu.

- Budowa modelu poprzez opis za pomocą równań matematycznych, jeżeli znane są prawa rządzące zachowaniem obiektu lub procesu.
- Identyfikacja modelu, jeżeli prawa rządzące nie są znane. Wtedy wystarczające są informacje o wejściu i wyjściu.

Model może występować w różnych postaciach, które są uwarunkowane zastosowaną metodologią objaśniania procesu. Najczęściej modele są przedstawiane:

- w postaci konwencjonalnej: funkcji, równań, układów równań,
- niekonwencjonalnej: reguł, wag połączeń międzyneuronowych.



Rys. 93. Budowanie modelu

MODELE UOGÓLNIAJĄCE I NIE UOGÓLNIAJĄCE

Specjalym podziałem modeli może być kryterium ogólności wiedzy bardzo charakterystyczne dla sieci neuronowej. Można wyróżnić **modele uogólniajace**, które pozwalają zastosować wiedzę jaką dysponuje sieć neuronowa do klasyfikacji przypadków, który nie zawierały się w zbiorze uczącym i testowym. Osiągnięcie odpowiedniej ogólności wiedzy jest związane z doborem takiej wartości błędu podczas etapu uczenia sieci, który pozwala na wypracowanie wag połączeń międzyneuronowych o odpowiedniej jakości. **Modele nie uogólniające** są w zasadzie używane wyłącznie dla przypadków, kiedy należy zbudować model dokładnie odpowiadający procesowi i używany w celu jego np. symulacji.

Przykład

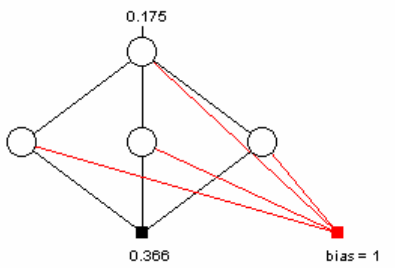
Przykład umożliwia porównanie wyników pracy sieci neuronowej mającej cechy modelu uogólniającego z siecią o cechach modelu nieuogólniajacego oraz modelami pośrednimi. Dla prezentacji zagadnienia wybrano dowolną funkcję opisującą proces ekonomiczny, określoną na zbiorze liczb naturalnych. Fragment przebiegu tej funkcji prezentuje rys. 97.

MODEL UOGÓLNIAJĄCY

W celu zbudowania modelu uogólniającego należy utworzyć sieć neuronową, która będzie posiadała odpowiednie właściwości. Podstawowym warunkiem uzyskania modelu uogólniającego jest zachowanie poniższej proporcji:

liczba próbek zbioru uczącego >> liczba parametrów modelu

Oznacza to, że jeżeli zbiór uczący zawiera 100 próbek, to liczba parametrów modelu, którą jest w przypadku sieci neuronowej liczba wag połączeń międzyneuronowych powinna być kilkakrotnie mniejsza, np. powinna wynosić 20 parametrów. Dla naszego przykładu wybraliśmy sieć neuronową składającą się czterech neuronów, sieć została przedstawiona na rys. 94.



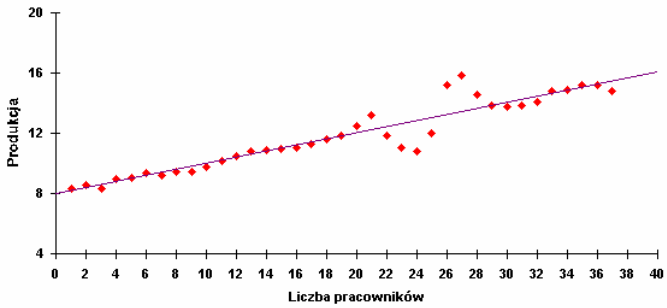
Rys. 94. Struktura sieci dla modelu uogólniającego

Wyznaczmy dla niej liczbę parametrów

$$liczba\ parametrów = 1 \cdot 0 + 3(1+1) + 1(3+1) = 10 \qquad (10\text{-}1)$$

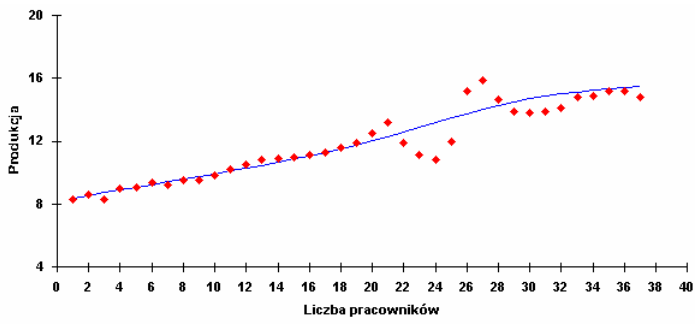
Ponieważ liczba próbek zbioru uczącego zawiera 37 wierszy, na które składają się wejścia i wyjścia sieci, model można uznać za uogólniający.

Obserwacja 10-1



Rys. 95. Aproksymacja danych empirycznych (regresja liniowa)

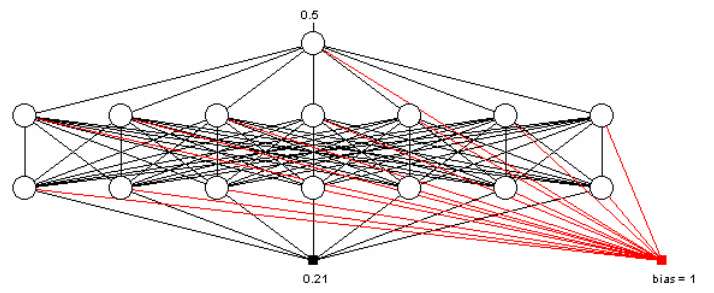
Obserwacja 10-2



Rys. 96. Model uogólniający

MODEL NIE UOGÓLNIAJĄCY

W modelu nie uogólniającym sieć neuronowa zachowuje się jak układ interpolujący. Takie właściwości sieci są w ogólnym przypadku niekorzystne. Dla takiego modelu przyjęto strukturę jak na rys. 97.

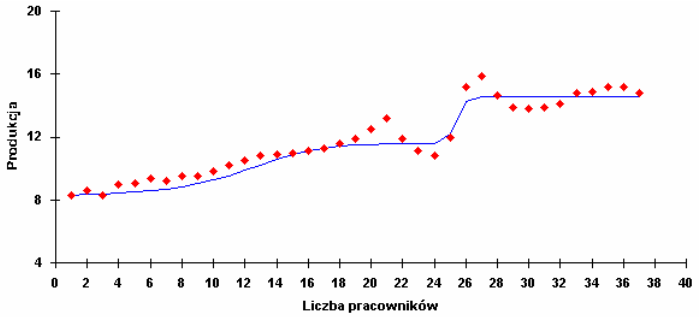


Rys. 97. Struktura sieci dla modelu dokładnego

Wyznaczmy dla niej liczbę parametrów

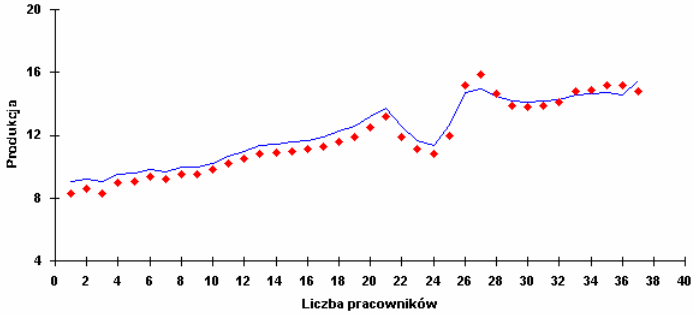
(10-2) $$\textit{liczba parametrów} = 1 \cdot 0 + 7(1+1) + 7(7+1) = 70$$

Obserwacja 10-3



Rys. 98. Model częściowo uogólniający

Obserwacja 10-4



Rys. 99. Model nie uogólniający

WNIOSKI

Jeżeli naszym celem jest wyuczenie sieci i **uogólnienie** wiedzy zawartej w wagach połączeniach międzyneuronowych, (parametry modelu), to należy przyjąć strukturę sieci (liczbę i rozmieszczenie neuronów), w której ich liczba jest **kilkakrotnie mniejsza** od liczby danych (próbek) w zbiorze uczącym. W innym przypadku, np. jeżeli liczba wag połączeń międzyneuronowych będzie **równa lub większa** od liczby danych uczących uzyskamy efekt konwersji jednej postaci danych w inną. Zaletą sieci neuronowych jest możliwość aproksymacji wszelkich nieliniowości. Na etapie uczenia sieci, istnieje możliwość takiego sterowania procesem uczenia sieci żeby uzyskać model odpowiedniej jakości. Istotą stosowania sieci neuronowych jest wyuczenie jej na danych uczących w celu pozyskania wiedzy, która jest przechowywana w postaci wag połączeń między neuronowych. W procesie uczenia powinno nastąpić uogólnienie wiedzy, w celu późniejszej weryfikacji w oparciu o przypadki nie uwzględnione w procesie uczenia. Dla modeli procesów tworzonych za pomocą sieci neuronowej należy zidentyfikować, czyli wyznaczyć parametry (wartości wag) połączeń międzyneuronowych. Zastąpienie danych empirycznych modelem nieliniowym zostało zrealizowane dla dwóch przypadków.

11

Identyfikacja modeli

W rozdziale:

- Co składa się na identyfikację modelu ?
- Model szeregu czasowego
- Przygotowanie danych dla prognozy
- Identyfikacja modelu szeregu czasowego
- Identyfikacja modelu Altmana

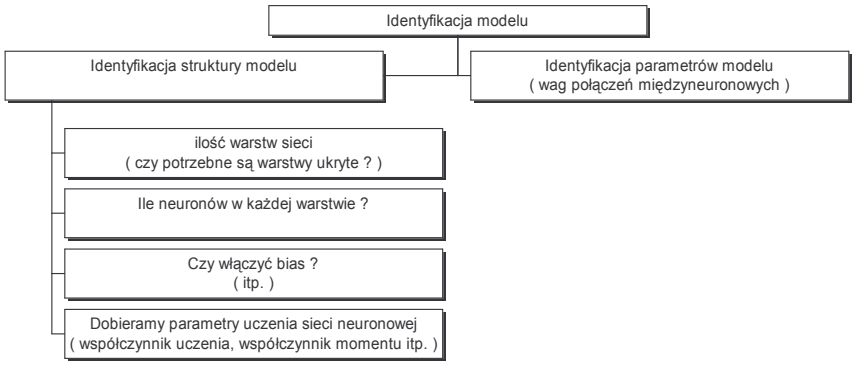
Cenne informacje

Ćwiczenia z komputerem

Koniecznie przeczytaj

CO SKŁADA SIĘ NA IDENTYFIKACJĘ MODELU ?

Identyfikacja modelu składa się z dwóch podstawowych czynności polegających na określeniu i doborze na podstawie najczęściej wiedzy eksperckiej i w zależności od rozwiązywanego problemu klasy modelu, rodzaju modelu, który może wynikać z modelowanego procesu.



Rys. 100. Identyfikacja modelu

Identyfikacja rozumiana w ogólnie jest procesem iteracyjnym obejmującym poza identyfikacją właściwą również testowanie i weryfikację modelu. W tym znaczeniu na identyfikację składają się następujące etapy:

- Wybór modelu.
- Identyfikacja struktury i parametrów wybranego modelu.
- Testowanie modelu
- Weryfikacja modelu.
- Jeżeli model zweryfikowany negatywnie to korekta.

Identyfikacji modeli szeregów czasowych jest możliwa za pomocą **kreatora**, który w sposób automatyczny generuje plik z danymi przygotowanymi specjalnie dla sieci neuronowej. Zastosowanie **kreatora plików uczących i testowych** umożliwia w następnej kolejności otrzymanie zbioru uczącego i testowego.

MODEL SZEREGU CZASOWEGO

Stochastyczny model szeregu czasowego można przedstawić jako **ciąg niezależnych impulsów**, które są realizacjami zmiennych losowych o ustalonym rozkładzie, który zwykle przyjmuje się jako **rozkład normalny.** Proces stochastyczny opisuje wartość średnia μ (11-1) wokół której proces oscyluje oraz wariancja σ^2 (11-3) będąca miarą rozrzutu procesu w stosunku do poziomu średniej. Wartości mogą być estymowane bezpośrednio z ciągu czasowego będącego realizacją procesu za pomocą sum skończonych odpowiednio (11-2) i (11-4). Zakłada się, że wartości elementów tego szeregu są silnie zależne. Proces białego szumu e_i zostaje przekształcony w proces y_i za pomocą filtru liniowego. Operacja filtracji polega na uwzględnianiu sum ważonych poprzednich obserwacji.

(11-1) $$\mu = E(y) = \int_{-\infty}^{\infty} y p(y) dy ,$$

(11-2) $$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i ,$$

(11-3) $$\sigma^2 = E\left[(y-\mu)^2\right] = \int_{-\infty}^{\infty} (y-\mu)^2 p(y) dy ,$$

(11-4) $$\sigma^2 = \frac{1}{n}\sum_{i=1} (y-\bar{y})^2$$

Rys. 101. Model szeregu czasowego

PRZYGOTOWANIE DANYCH DLA PROGNOZY

W celu przeprowadzenia identyfikacji modelu szeregu czasowego należy w odpowiedni sposób przygotować dane dla celów prognozy. Zależnie od budowanego modelu prognostycznego konieczna jest analiza szeregu metodami eksperckimi (ocena postaci graficznej szeregu), jak również przeprowadzenie obliczeń sprowadzających postać szeregu do warunków dogodnych dla wybranego rodzaju prognozy. Istnieje możliwość pewnej automatyzacji przeprowadzania oceny eksperckiej za pomocą specjalnie opracowanych algorytmów, które spróbują wstępnie rozpoznać parametry strukturalne modelu szeregu. Na prawidłową identyfikację parametrów strukturalnych modelu szeregu czasowego są bardzo wrażliwe metody statystyczne. Najmniejszą wrażliwością charakteryzują się modele neuronowe, dla których jest możliwa bezpośrednia estymacja parametrów modelu kosztem jednak zachowania dokładności prognozy. Sposób taki nie jest optymalny i dlatego zaleca się przygotowanie danych dla prognozy poprzez ich odpowiednie opracowanie. Dla zaprezentowania podstawowych problemów, jakie mogą się pojawić podczas przygotowywania danych posłużymy się typowym modelem ekonometrycznym. Podstawowymi składnikami szeregu czasowego w ujęciu modelu ekonometrycznego są: tendencja rozwojowa (trend), wahania okresowe, wahania koniunkturalne, wahania przypadkowe.

TENDENCJA ROZWOJOWA

Tendencja rozwojowa, czyli trend jest własnością szeregu czasowego ujawniającą się poprzez systematyczne, jednokierunkowe zmiany (wzrost lub spadek) poziomu badanego zjawiska, zachodzące w długim okresie. Przyczyna takiego rozwoju w procesie jest stałe i systematyczne oddziaływanie pewnych czynników.

WAHANIA OKRESOWE

Wahania okresowe są rytmicznymi zmianami w realizacji procesu o określonym cyklu (okresie przebiegu). Ich przyczyną są najczęściej czynniki przyrodnicze. Przyjmowane są okresy roczne, kwartalne, miesięczne i inne.

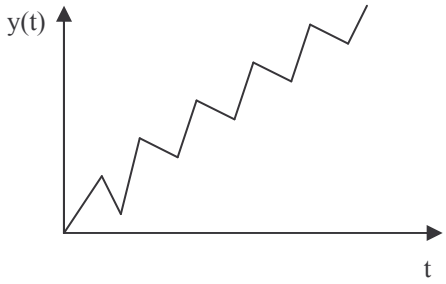
WAHANIA KONIUNKTURALNE

Wahania koniunkturalne są typowe dla procesów gospodarczych.

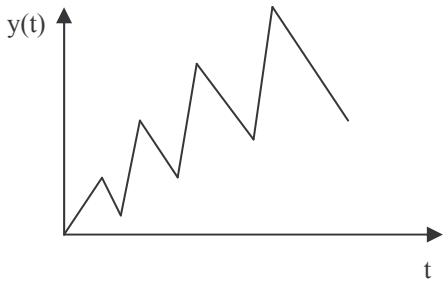
WAHANIA PRZYPADKOWE

Wahania przypadkowe są realizacją zmiennej losowej o określonym rozkładzie.

Na podstawie opisu ekonometrycznego jest możliwe wyróżnienie dwóch modeli: **modelu addytywnego,** dla którego występuje liniowy trend oraz stała amplituda wahań okresowych i **modelu multiplikatywnego** z liniowym trendem, na który są nałożone wahania okresowe o amplitudzie proporcjonalnej do poziomu procesu.



Rys. 102. Model addytywny



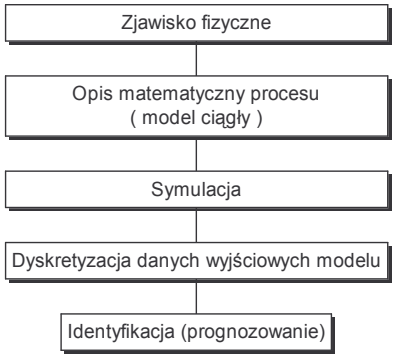
Rys. 103. Model multiplikatywny

Prawidłowe przygotowanie danych w zależności od metody prognozowania powinno określić cel prognozy w sensie kryterium czasu. Dla **prognoz krótkoterminowych** należy dążyć do uwydatnienia lokalnych cech szeregu czasowego w postaci drgań szybkozmiennych i niewielkich amplitudach. Jest to możliwe poprzez eliminację z szeregu czasowego składników tendencji rozwojowej oraz wahań okresowych. W tym celu posługiwać należy się metodami statystycznymi w postaci średnich ruchomych, wygładzania wykładniczego, lub sieci wyuczonej dla odfiltrowania składowej długookresowej. **Prognozy długoterminowe** wymagają użycia metod filtracji górnoprzepustowej w celu wytłumienia lokalnych wahań i wyodrębnienia trendu. Celem przygotowania danych do identyfikacji modelu szeregu czasowego dla celów prognozy krótkookresowej przy pomocy sieci neuronowej należy spełnić kilka warunków. **Ocenić histogram próbki szeregu** (lub całego szeregu) w celu oszacowania wielkości asymetrii tego szeregu oraz subiektywnego określenia rodzaju rozkładu. Należy podjąć czynności zmierzające do usunięcia pewnych składników szeregu, jeżeli jego rozkład znacząco odbiega od rozkładu oczekiwanego (np. normalnego), lub zmniejszyć ich wagę w końcowej ocenie szeregu. Określić **rodzaj procesu** reprezentowanego przez szereg (stacjonarny lub niestacjonarny) i zastosować metody eliminacji niestacjonarności (np. różnicowanie).

IDENTYFIKACJA MODELU SZEREGU CZASOWEGO

Przykład

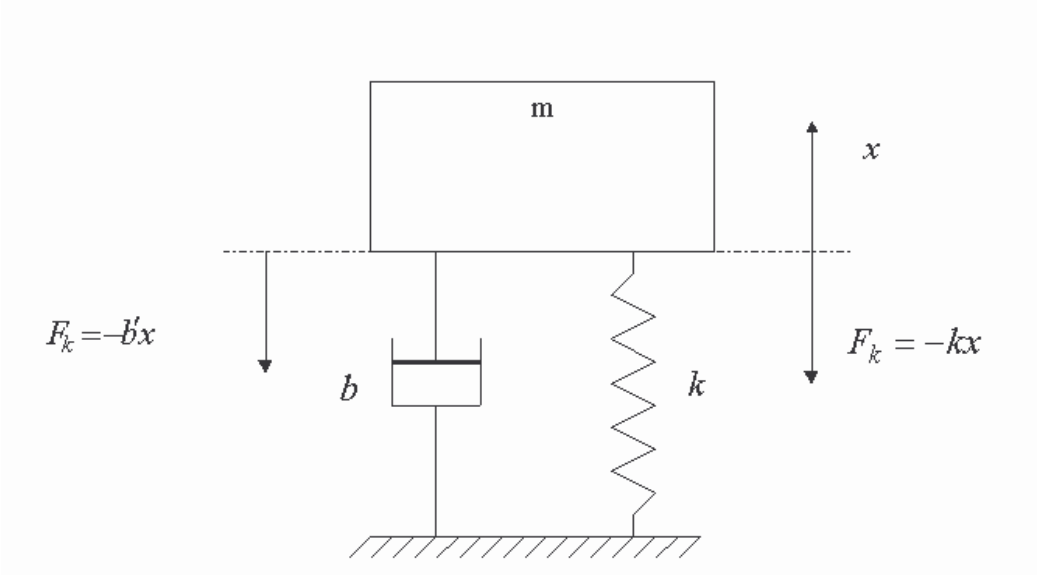
Przykład prezentuje możliwości syntezy **modelu szeregu czasowego drgań** poprzez posłużenie się modelem matematycznym drgań i zbudowaniu na jego podstawie modelu szeregu czasowego tych drgań.



Rys. 104. Czynności w przykładzie

Typowym zjawiskiem, na które można natrafić w większości układów technicznych zbudowanych przez człowieka są różne odmiany drgań. Drgania występują w układach mechanicznych, hydraulicznych, elektrycznych i innych. Dla pokazania przykładowej możliwości analizy takich drgań zbudowano dwa modele matematyczne: dla przykładowego układu mechanicznego oraz elektrycznego. Matematyczny zapis modelu jest taki sam, różnice występują jedynie w zjawiskach fizycznych wywołujących drgania.

UKŁAD MECHANICZNY



Rys. 105. Modelowany układ mechaniczny

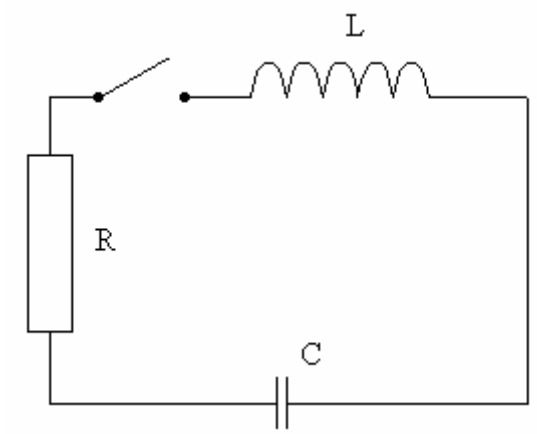
Modelowane są drgania masy m, połączonej z podporą za pomocą elementu tłumiącego o danym współczynniku b oraz elementu sprężystego o współczynniku równym k. Założono brak wymuszenia w postaci siły zewnętrznej. Równanie (11-1) opisuje matematycznie, poprzez wykorzystanie równania różniczkowego zachowanie drgającej masy

(11-5) $$\frac{d^2x(t)}{dt^2} + \frac{1}{\tau}\frac{dx(t)}{dt} + \omega_0^2 x(t) = 0 .$$

Przyjęto następujące oznaczenia i interpretację fizyczną:
b - współczynnik zależny od ośrodka i kształtu ciała, związany z właściwościami tłumiącymi ośrodka (siła oporu hamująca ruch masy m),
$\frac{1}{\tau} = 2\beta = \frac{b}{m}$ - stała,
$\omega_0^2 = \frac{k}{m}$ - kwadrat częstości drgań własnych układu.
Rozwiązaniem powyższego równania różniczkowego jest zależność

$$x(t) = A_0 \cdot e^{-\frac{t}{2\tau}} \cdot \sin(\omega\, t + \varphi) , \qquad (11\text{-}6)$$

UKŁAD ELEKTRYCZNY



Rys. 106. Modelowany układ elektryczny

Modelowane są drgania w obwodzie elektrycznym składającym się z oporu, kondensatora oraz cewki (obwód RLC). Dla opisanego obwodu można wyprowadzić równanie różniczkowe opisujące wartość ładunku na kondensatorze w funkcji czasu.

(11-7) $$\frac{d^2Q(t)}{dt^2}+\frac{R}{L}\frac{dQ(t)}{dt}+\frac{1}{LC}Q(t)=0\,.$$

Przyjęto następujące oznaczenia i interpretację fizyczną:

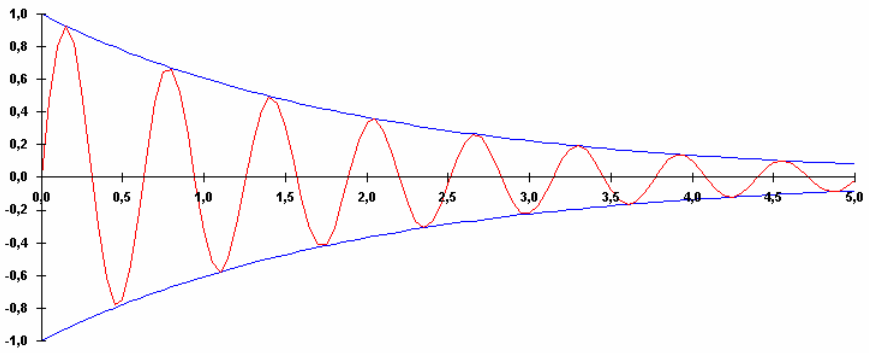
$\tau=\frac{L}{R}$ - stała o wymiarze czasu.

Rozwiązaniem powyższego równania różniczkowego jest zależność

(11-8) $$Q(t)=Q_0\cdot e^{-\frac{R}{2L}t}\cdot\cos(\omega t)\,.$$

WYGENEROWANIE SZEREGU CZASOWEGO WYJŚCIA MODELU (SYMULACJA)

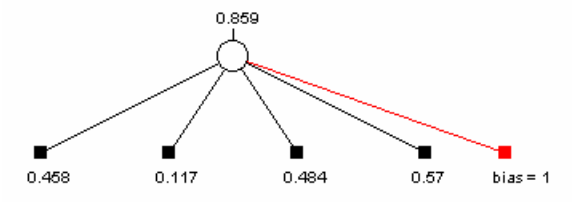
Dla założonych parametrów $\lambda=0.5$, $\omega_0=10$ wygenerowano wartości wyjścia modelu matematycznego oraz zobrazowano wykresem na rys. 107. Parametry modelu matematycznego zostały dobrane w celu ogólnego przestawienia problemu i mogą być sprzeczne z właściwościami fizycznymi modelowanych układów.



Rys. 107. Wyjście modelu matematycznego

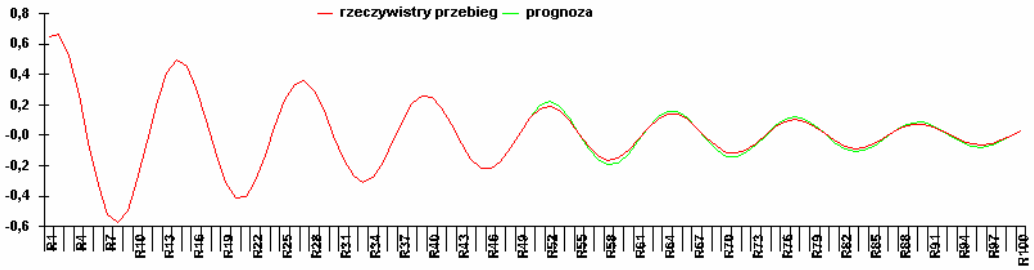
IDENTYFIKACJA PARAMETRÓW MODELU SZEREGU CZASOWEGO

W celu wyuczenia sieci neuronowej zadanego szeregu czasowego, podzielono wygenerowany zbiór na dwa podzbiory równo liczne, podzbiór **uczący** oraz **testujący.** Uczenie przeprowadzono dla 500 epok, osiągając błąd RMS = 0.0183. Strukturę sieci przedstawia rys. 108.

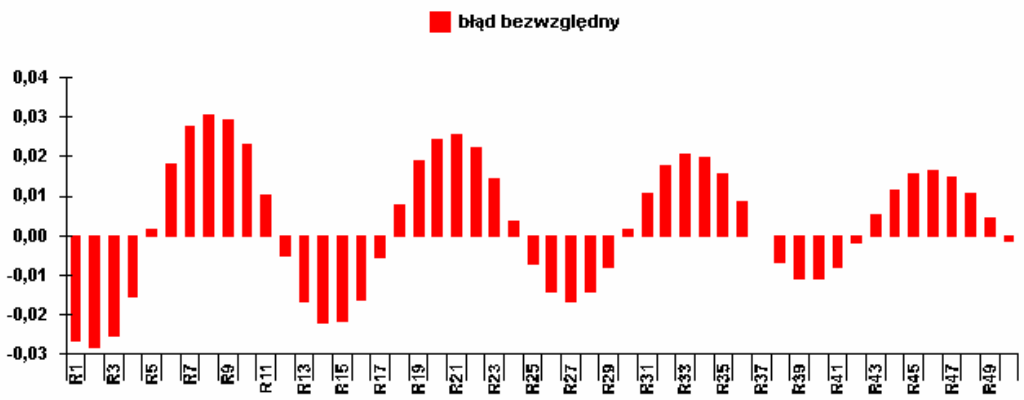


Rys. 108. Struktura sieci neuronowej

Dla podanej struktury sieci neuronowej, utworzono model uogólniający o liczbie parametrów $1 \cdot 0 + 1(4+1) = 5$.



Rys. 109. Prognoza



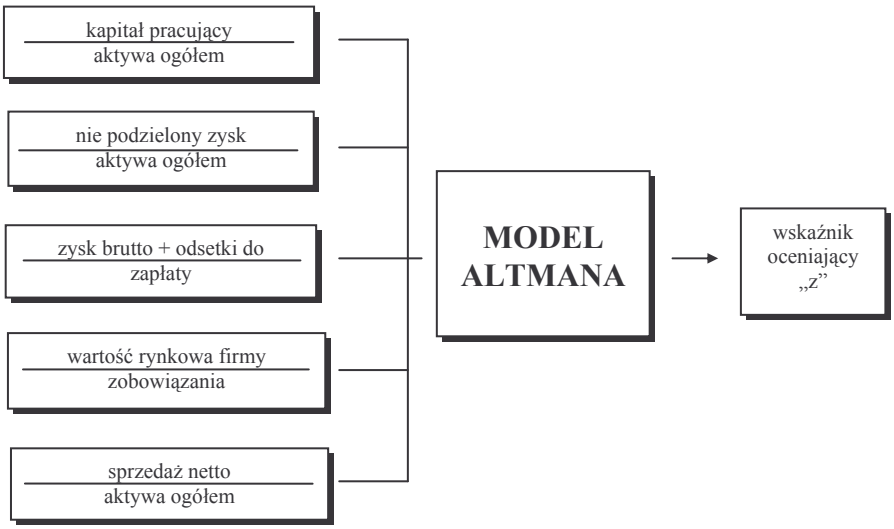
Rys. 110. Błąd bezwzględny prognozy (dane testujące)

IDENTYFIKACJA MODELU ALTMANA

Ideą zastosowania sieci neuronowej do oceny płynności finansowej przedsiębiorstwa jest uzyskanie narzędzia o cechach naśladujących postępowanie eksperta. Zasadniczą cechą takiego podejścia jest próba formalizacji pewnych zależności wyprowadzonych na podstawie subiektywnej oceny eksperta, badań statystycznych, lub innych źródeł. Dla przedstawienia typowego podejścia w modelowaniu problemów oceny jakości finansów firmy, zostanie zaprezentowany popularny **model Altmana** oceny kondycji finansowej firmy. Jest to model liniowy ze względu na współczynniki. Matematyczny zapis modelu Altmana jest następujący:

(11-9) $$z = 1{,}2 \cdot x_1 + 1{,}4 \cdot x_2 + 3{,}3 \cdot x_3 + 0{,}6 \cdot x_4 + 1{,}0 \cdot x_5 ,$$

gdzie zmienne x_1, x_2, x_3, x_4, x_5 oznaczają zdefiniowane wskaźniki kondycji finansowej firmy.

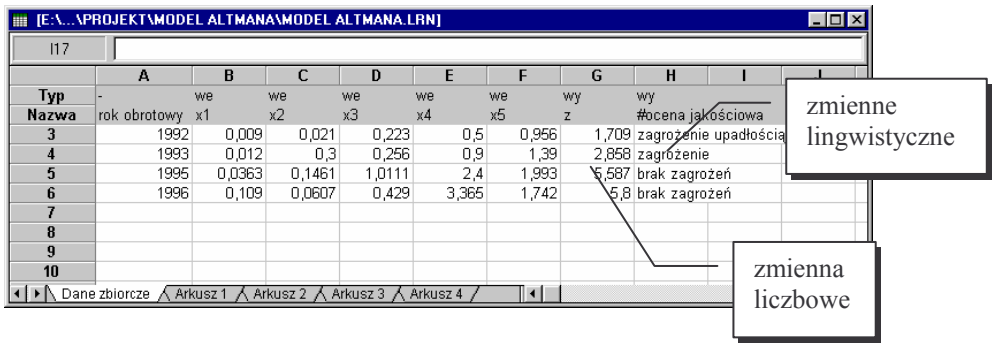


Rys. 111. Schemat modelu Altmana

Ocena zmiennej wyjściowej jest prosta i polega na wskazaniu klasy kondycji finansowej według zależności:

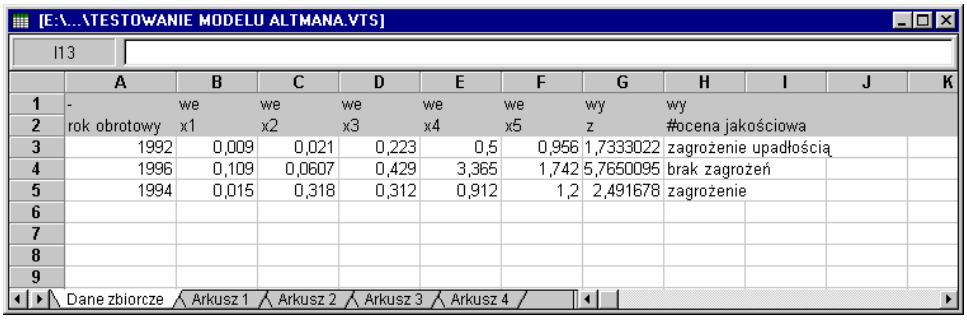
$$\text{kondycja finasowa firmy } (\textit{aspekt zagrożenia upadością}) = \begin{cases} \text{duże zagrożenie,} & z \le 1{,}8 \\ \text{zagrożenie,} & 1{,}8 < z < 2{,}99 \\ \text{brak zagrożenia,} & z \ge 2{,}99 \end{cases} \tag{11-10}$$

Powyższa klasyfikacja może zostać przedstawiona jako problem poszukiwania rozwiązania w układzie pięciowymiarowym. Problem klasyfikacji za pomocą sieci neuronowej kondycji finansowej firmy do jednego ze stanów zagrożenia bankructwem sprowadza się do estymacji parametrów modelu poprzez aproksymacje funkcji $z(x_1, x_2, x_3, x_4, x_5)$ w przestrzeni wielowymiarowej. Celem prezentacji budowy modelu ekonomicznego utworzono zbiór uczący z danych wejściowych wymaganych przez model Altmana oraz wyjściowych, czyli oceny ilościowej i jakościowej kondycji finansowej firmy. Dla oceny przydatności modelu wyłączono ze zbioru próbkę dotyczącą 1994 roku. Oznacza to, że sieć nie będzie posiadała wiedzy odnoście tego okresu. Sieć musi uogólnić posiadaną wiedzę z poprzednich okresów.



	A	B	C	D	E	F	G	H	I	J
Typ	-	we	we	we	we	we	wy	wy		
Nazwa	rok obrotowy	x1	x2	x3	x4	x5	z	#ocena jakościowa		
3	1992	0,009	0,021	0,223	0,5	0,956	1,709	zagrożenie upadłością		
4	1993	0,012	0,3	0,256	0,9	1,39	2,858	zagrożenie		
5	1995	0,0363	0,1461	1,0111	2,4	1,993	5,587	brak zagrożeń		
6	1996	0,109	0,0607	0,429	3,365	1,742	5,8	brak zagrożeń		
7										
8										
9										
10										

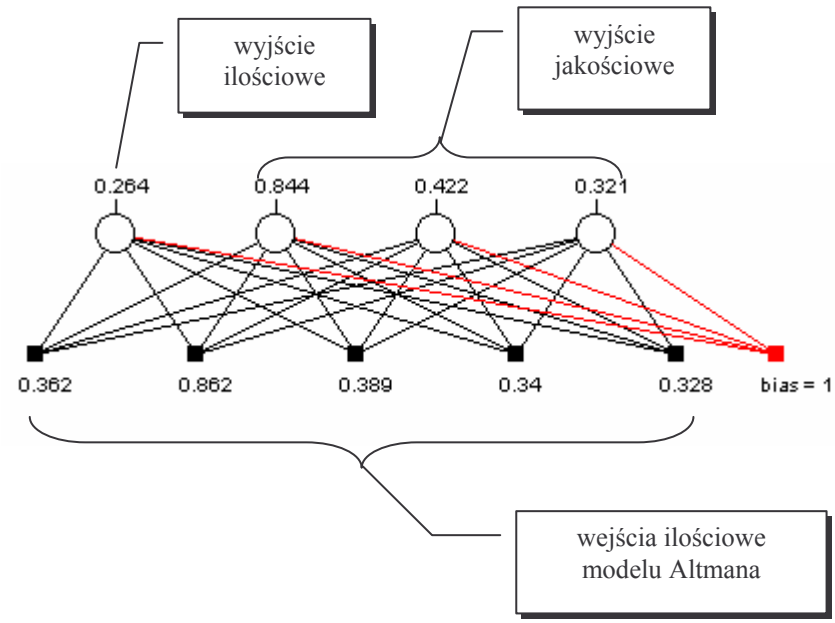
Rys. 112. Dane uczące w modelu Altmana



	A	B	C	D	E	F	G	H	I	J	K
1	-	we	we	we	we	we	wy	wy			
2	rok obrotowy	x1	x2	x3	x4	x5	z	#ocena jakościowa			
3	1992	0,009	0,021	0,223	0,5	0,956	1,7333022	zagrożenie upadłością			
4	1996	0,109	0,0607	0,429	3,365	1,742	5,7650095	brak zagrożeń			
5	1994	0,015	0,318	0,312	0,912	1,2	2,491678	zagrożenie			
6											
7											
8											
9											

Rys. 113. Testowanie modelu Altmana

Podczas testowania możemy zobaczyć różnice w wynikach pracy sieci neuronowej oraz wynikach uzyskanych metodami tradycyjnymi poprzez podstawienie do wzoru reprezentującego model Altmana. Dla wzorców dotyczących roku obrotowego 1992 oraz 1996, które zawierały się w zbiorze uczącym różnice są niewielkie (pomijalnie małe), natomiast dla roku 1994, w którym sieć musiała uogólnić pozyskaną wiedzę w procesie uczenia różnica jest znaczna. W jaki sposób tłumaczyć taką rozbieżność ? Należy mieć na uwadze fakt, że model Altamana jest modelem liniowym. Model Altamana utworzony za pomocą sieci neuronowej jest w ogólnym przypadku nieliniowy. Dlatego sieć może wybrać wartości wyjścia inne niż w modelu liniowym, dla którego w roku 1994 wyjściowy wskaźnik kondycji finansowej powinien wynosić 2,573. Wizualizacja topologii sieci użytej do utworzenia modelu Altmana jest przedstawiona na rys. 114.



Rys. 114. Topologia sieci dla modelu Altmana

12

Kreator prognoz

W rozdziale:

- Podstawowe opcje kreatora
- Filtracja szeregu czasowego
- Jak filtry działają w praktyce ?
- Co widać na rozkładzie prawdopodobieństwa ?
- Przygotowanie danych za pomocą kreatora
- Przegląd metod przygotowania danych na przykładach

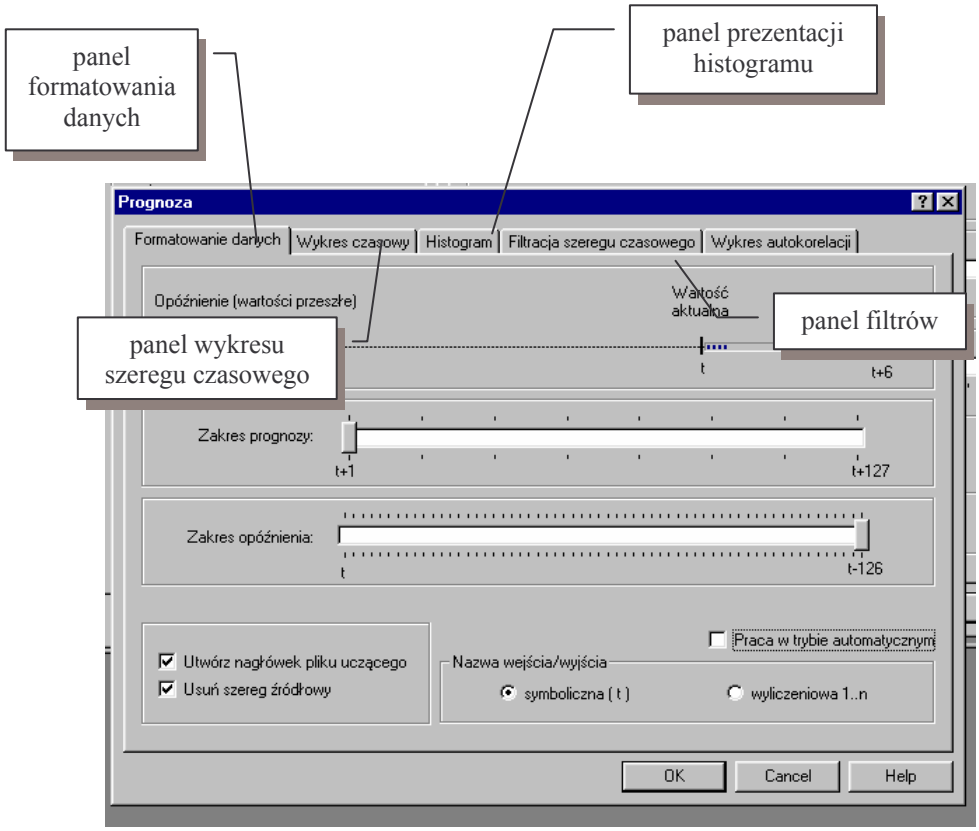
Cenne informacje

Ćwiczenia z komputerem

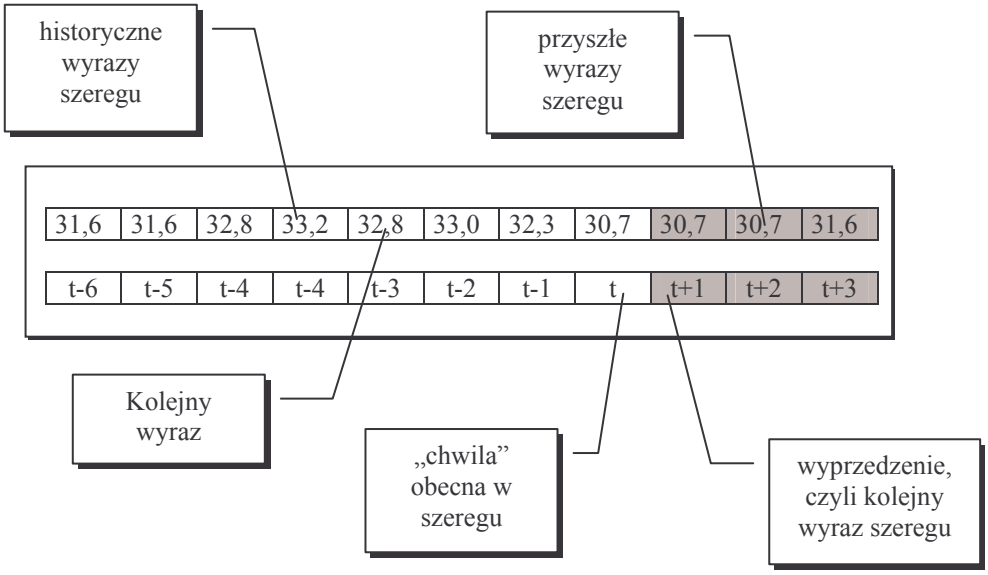
Koniecznie przeczytaj

PODSTAWOWE OPCJE KREATORA

Kreator składa się z zakładek, na których znajdują się pogrupowane opcje. Każda zakładka dotyczy określonego zakresu przygotowania danych dla uczenia sieci w celu uzyskania prognozy.



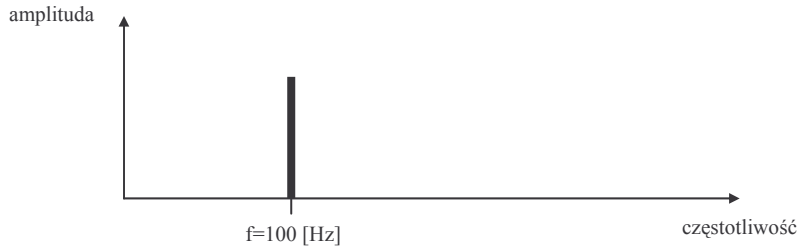
Kreator posiada dodatkowe udogodnienia w postaci opcji **Pracy w trybie automatycznym**. Opcja umożliwia efektywną pracę Użytkownikom nie posiadającym dostatecznej wiedzy na temat prognozowania.



Rys. 116. Schemat szeregu (ciągu) czasowego

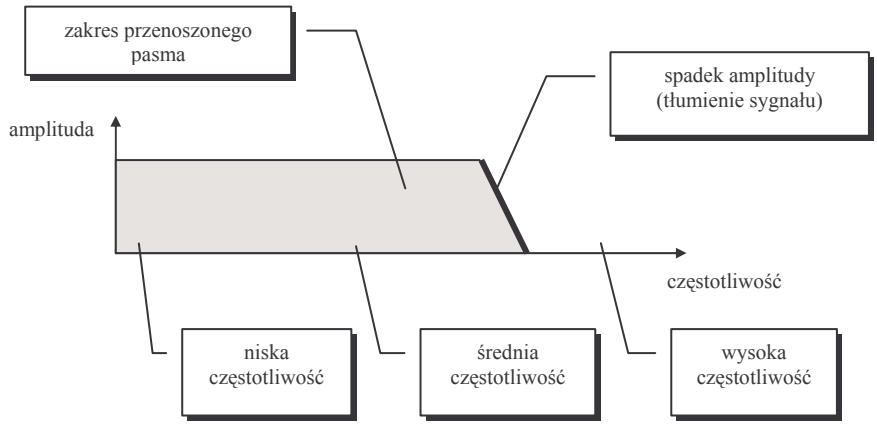
FILTRACJA SZEREGU CZASOWEGO

W uproszczony sposób, wystarczający dla przeprowadzenia odpowiedniej **korekcji szeregu czasowego,** można określić filtr szeregu czasowego jako metodę pozwalająca zmienić wartości wyrazów w szeregu, zgodnie z funkcją przejścia (na przykład transmitancją). Ponadto należy wprowadzić pojęcie **widma sygnału**. Widmo jest **obrazem sygnału w dziedzinie częstotliwości**, podobnie jak **wykres czasowy w dziedzinie czasu**.

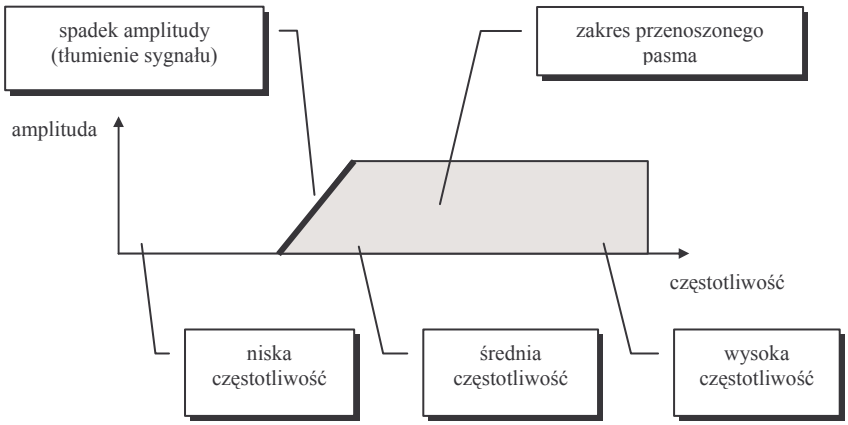


Rys. 117. Widmo idealnej sinusoidy

Powyższy rysunek przedstawia widmo jakie uzyskamy dla sygnału y=sin(x). Dla większości rzeczywistych sygnałów widmo będzie ciągłe. Można wtedy mówić o filtrowaniu sygnału, czyli usuwaniu (tłumieniu) pewnych zakresów widma sygnału. Poniższe rysunki 118 i 119 przedstawiają zasadę działania filtru dolnoprzepustowego oraz górnoprzepustowego.



Rys. 118. Charakterystyka filtru dolnoprzepustowego

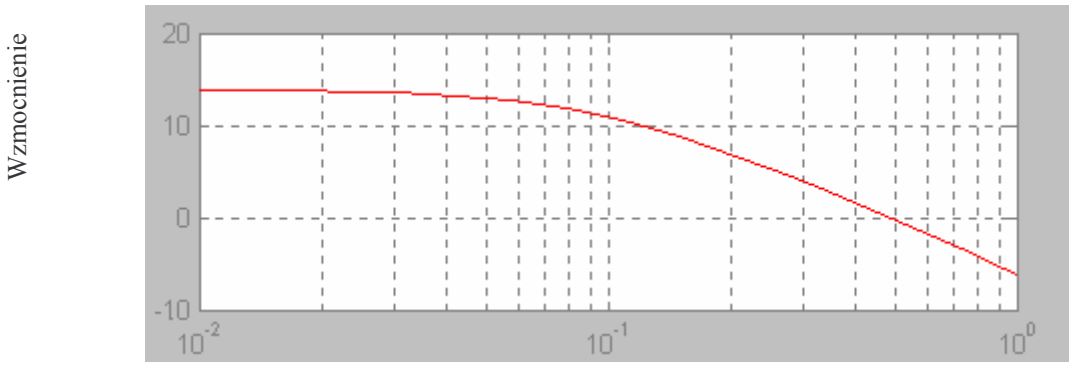


Rys. 119. Charakterystyka przenoszenia filtru górnoprzepustowego

JAK FILTRY DZIAŁAJĄ W PRAKTYCE ?

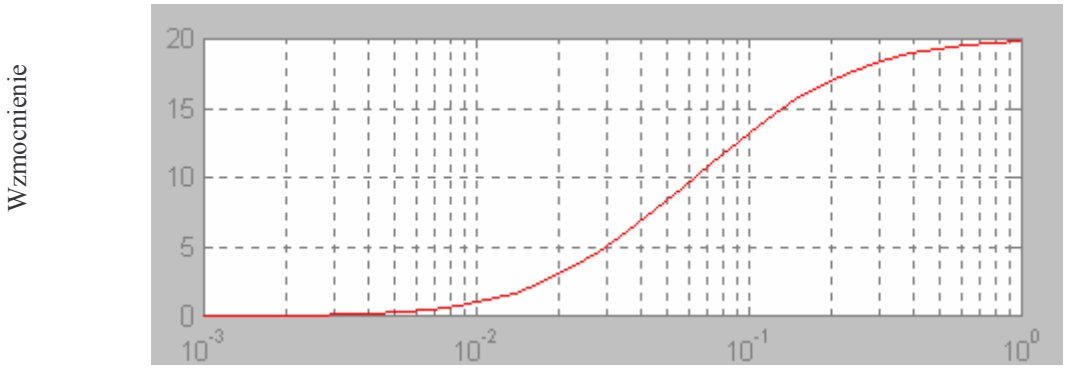
Bardzo często zdarza się, że sygnał (przebieg), który analizujemy jest złożeniem przebiegów sinusoidalnych o różnych amplitudach i częstotliwościach. Jeśli sygnał jest złożeniem przebiegów sinusoidalnych z całego zakresu częstotliwości to taki sygnał nazywamy białym szumem. W praktyce uzyskanie białego szumu jest niemożliwe, zdarzają się losowe sygnały do niego podobne. Jeśli przebieg losowy ma jakikolwiek wpływ na dany proces to jest on zakłóceniem. W przypadku sygnału, który jest złożeniem skończonej liczby przebiegów sinusoidalnych o różnych częstotliwościach każdy z sygnałów składowych nosi nazwę harmonicznej. I tak sygnał najbardziej znaczący jest pierwszą harmoniczną, następny w kolejności drugą harmoniczną itd. Badany sygnał może być również złożeniem sygnału użytecznego i zakłócenia, przy czym sygnał użyteczny może być złożeniem kilku harmonicznych. Niezależnie od przypadku, jeśli do analizy potrzebny jest nam tylko sygnał użyteczny lub sygnały z określonego zakresu częstotliwości należy stosować odpowiednie filtry.

FILTR DOLNOPRZEPUSTOWY



Na przedstawionej powyżej charakterystyce amplitudowej widać ze dla niskich częstotliwości wzmocnienie filtru jest duże a dla wyższych częstotliwości bardzo małe – filtr taki będzie przenosił bez zniekształceń przebiegi o niskich częstotliwościach natomiast wytłumi przebiegi o wysokich częstotliwościach

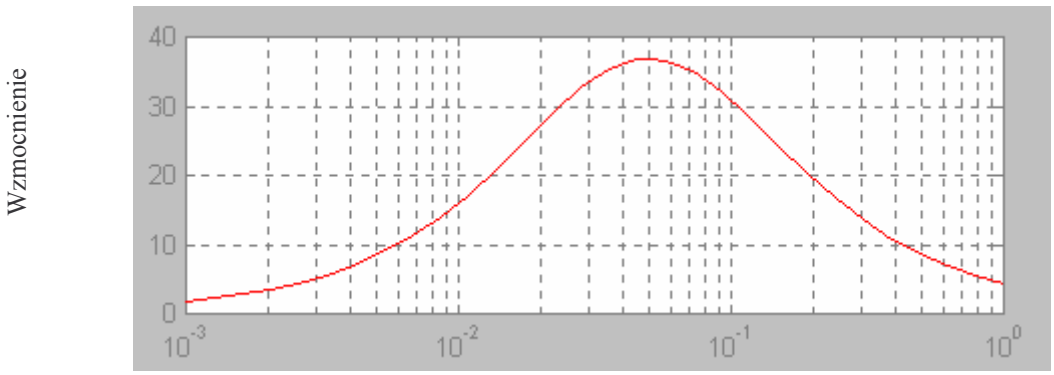
FILTR GÓRNOPRZEPUSTOWY



Na przedstawionej powyżej charakterystyce amplitudowej widać że dla wysokich częstotliwości wzmocnienie filtru jest duże a dla niskich częstotliwości bardzo małe – filtr taki będzie przenosił bez zniekształceń przebiegi o wysokich częstotliwościach natomiast wytłumi przebiegi o niskich częstotliwościach

FILTR PASMOWOPRZEPUSTOWY

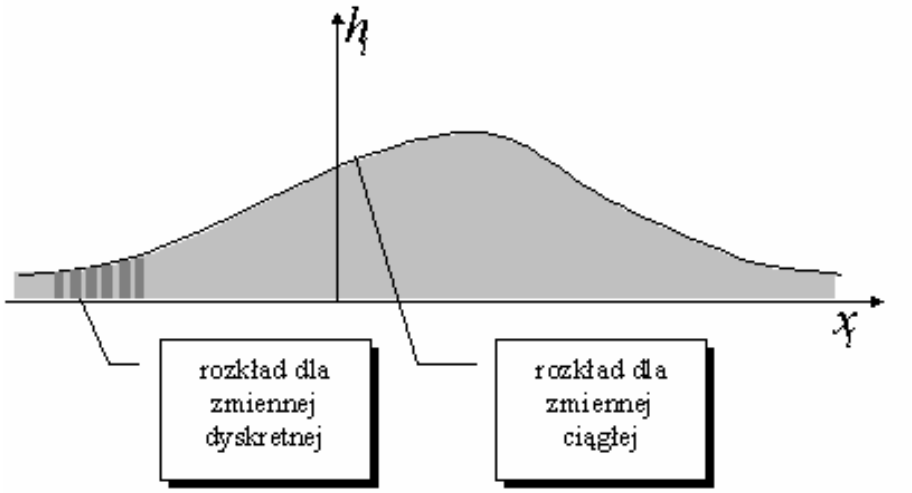
Do wyselekcjonowania pewnego zakresu (pasma) częstotliwości a zatem odfiltrowania wszystkich częstotliwości leżących poza pasmem stosujemy filtr **pasmowoprzepustowy**.



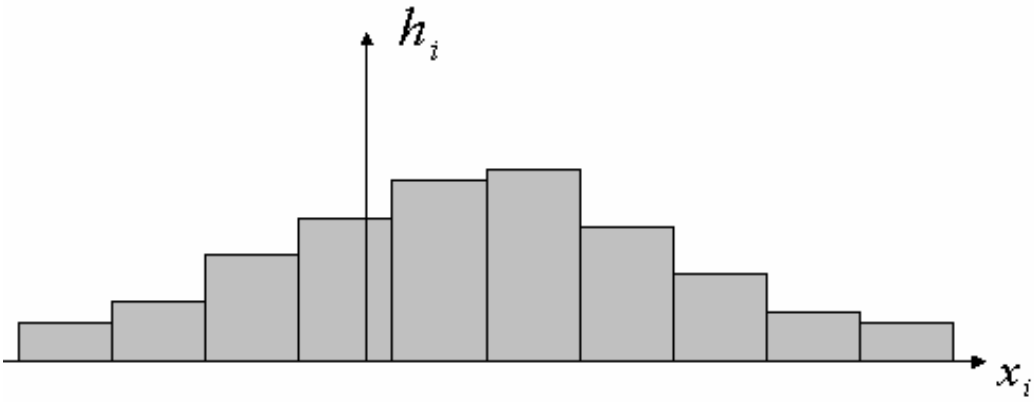
Na przedstawionej powyżej charakterystyce amplitudowej widać ze dla częstotliwości lezących w interesującym nas przedziale wzmocnienie filtru jest duże a dla niskich i wysokich częstotliwości bardzo małe – filtr taki będzie przenosił bez zniekształceń przebiegi o częstotliwościach leżących w paśmie w którym wzmocnienie jest duże natomiast wytłumi przebiegi o niskich i wysokich częstotliwościach

CO WIDAĆ NA ROZKŁADZIE PRAWDOPODOBIEŃSTWA ?

Rozkład prawdopodobieństwa umożliwia nam ocenę ważnych cech szeregu czasowego. Rozkład jest w zasadzie budowany dla zmiennej ciągłej. W przypadku, gdy dysponujemy danymi empirycznymi wykreślenie rozkładu prawdopodobieństwa jest możliwe dla odpowiedniej licznej populacji. Alternatywą rozkładu jest histogram wykreślany dla zmiennej dyskretnej (grupowanej przedziałami). Użycie histogramu jest możliwe dla mniej licznej próbki całej populacji. Wykres rozkładu zbudowany jest z osi x, na której odkładane są wartości występujące w populacji oraz osi h, przedstawiającej najczęściej gęstość prawdopodobieństwa. Oznacza to dla wybranej wartości zmiennej ciągłej z osi x można określić prawdopodobieństwo jej wystąpienia w populacji. Histogram wykreślony dla danych empirycznych będących próbką z całej populacji pozwala na oszacowanie rozkładu prawdopodobieństwa tej populacji. W histogramie przedstawiana jest graficznie ilość wystąpień (częstość) jednostki lub ich grupy (przedział) w próbce populacji.



Rys. 120. Rozkład prawdopodobieństwa



Rys. 121. Rozkład dla zmiennej uśrednianej przedziałami

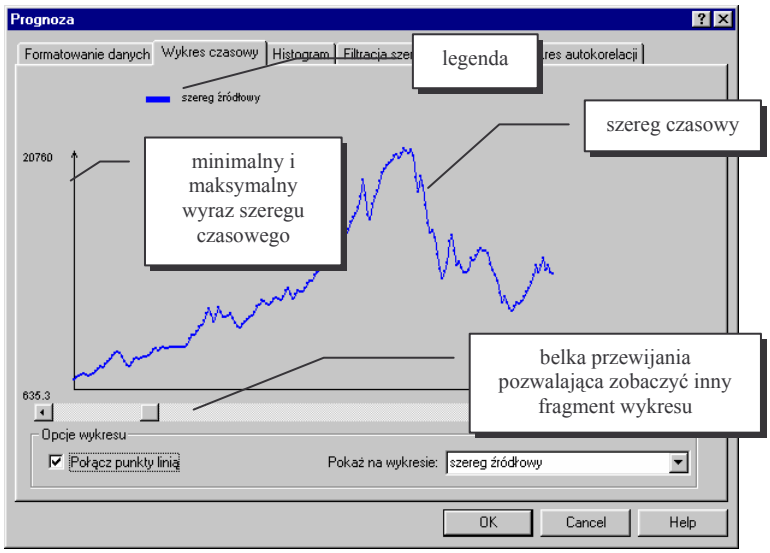
Wykreślając histogram rozkładu zmiennej ciągłej, można dobrać skalę na osi rzędnych, aby łączna powierzchnia prostokątów, która reprezentuje całą populację była równa 1. Wtedy pola prostokątów są równe częstością jednostek w poszczególnych przedziałach. Można w ten sposób określić **funkcję gęstości** opisującą rozkład zmiennej ciągłej.

PRZYGOTOWANIE DANYCH ZA POMOCĄ KREATORA

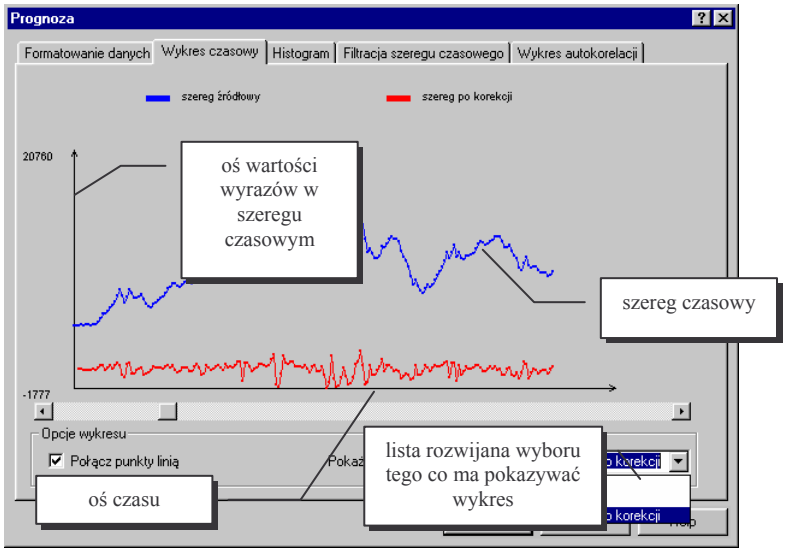
PANEL FORMATOWANIA DANYCH

Panel formatowania danych pozwala na przygotowanie pliku uczącego, który jest niezbędny dla przeprowadzenia procesu uczenia sieci neuronowej. Panel dysponuje opcjami, które pozwalają na określenie podstawowych parametrów prognozy. Do tych podstawowych parametrów należy zaliczyć zakres danych historycznych pobranych z szeregu czasowego na podstawie których będą prognozowane przyszłe wartości szeregu.

PANEL WYKRESU SZEREGU CZASOWEGO

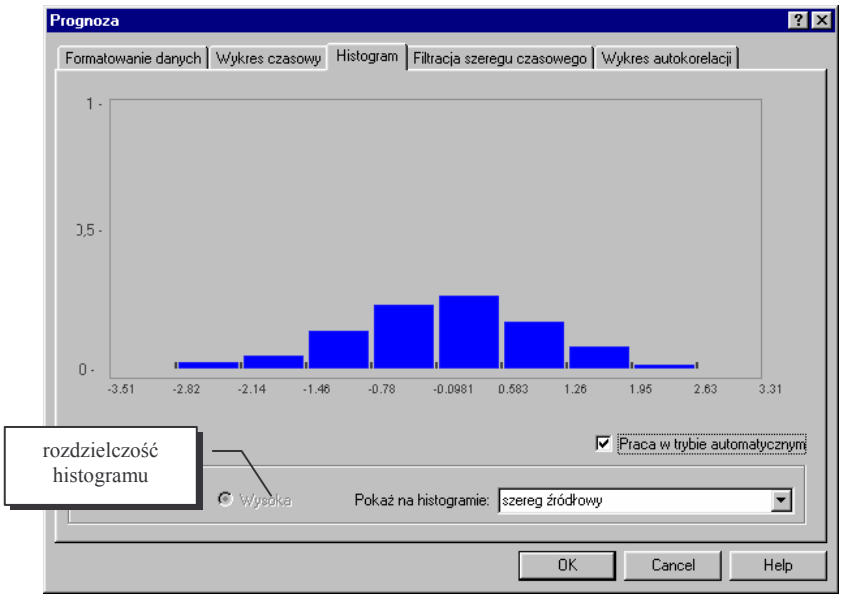


Rys. 122. Panel wykresu szeregu czasowego



Rys. 123. Panel wykresu szeregu czasowego z dwoma przebiegami

PANEL HISTOGRAMU SZEREGU CZASOWEGO



Rys. 124. Panel histogramu

W celu zaprezentowania właściwości panelu wykorzystano wygenerowane szeregi czasowe szumów o różnych rozkładach. Każdy szereg liczy 1000 wyrazów. Histogram pozwala ocenić w sposób przybliżony rozkład zmiennej losowej.

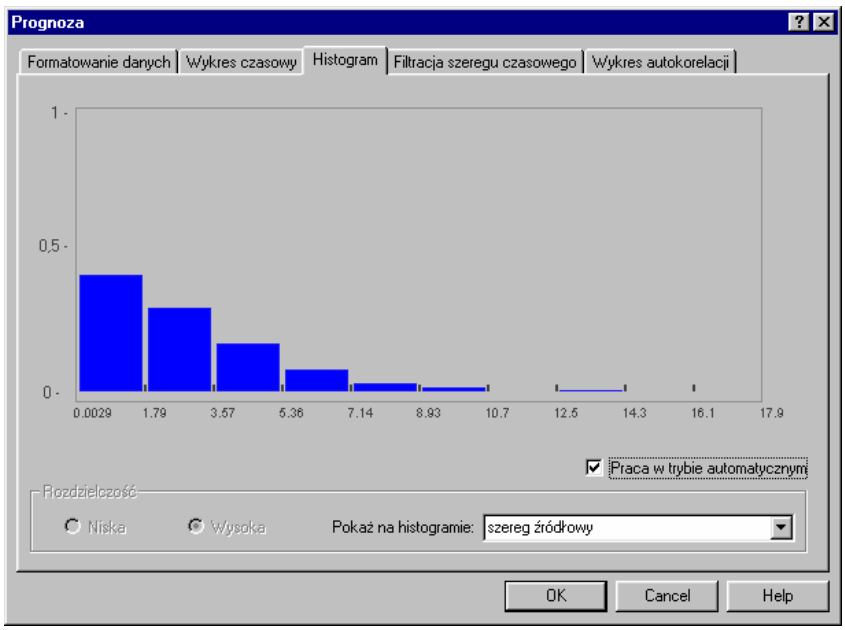
[E:\SPHINX\PROJEKT\PROGNOZY\SZUMY TESTOWE.VTS]

D8 1,5125

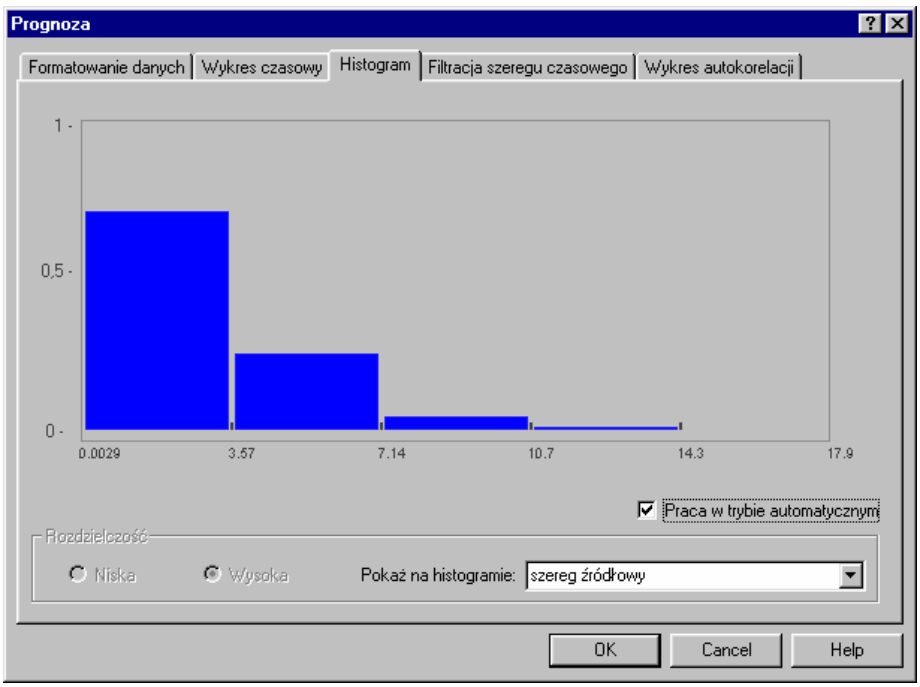
	A	B	C	D
1	szum biały	rozkład normalny	rozkład Beta	rozkład Chi-square
2	(szum A)	(szum B)	(szum C)	(szum D)
3	1,253827684	1,6461	0,7285	1,2025
4	1,081888542	-0,7351	0,4933	0,493
5	3,30072679	0,2711	0,3843	3,3469
6	3,31515374	-0,0351	0,5283	12,011
7	1,788908643	1,2335	0,2604	3,4893
8	2,187556488	0,7131	0,4055	1,5125
9	2,261887017	-0,1093	0,8013	5,9365
10	3,53509183	-2,0857	0,3312	3,6619
11	2,288390189	0,9207	0,824	3,3445
12	0,9739000057	-0,1391	0,2841	6,6074
13	0,3373953776	0,2935	0,6207	2,0787
14	2,604111563	0,8132	0,3002	0,8539
15	3,30303538	1,4174	0,6598	0,7582

Dane zbiorcze | Arkusz 1 | Arkusz 2 | Arkusz 3 | Arkusz 4

Rys. 125. Arkusz zawierający wygenerowane szeregi czasowe szumów



Rys. 126. Histogram szumu o rozkładzie Chi-square



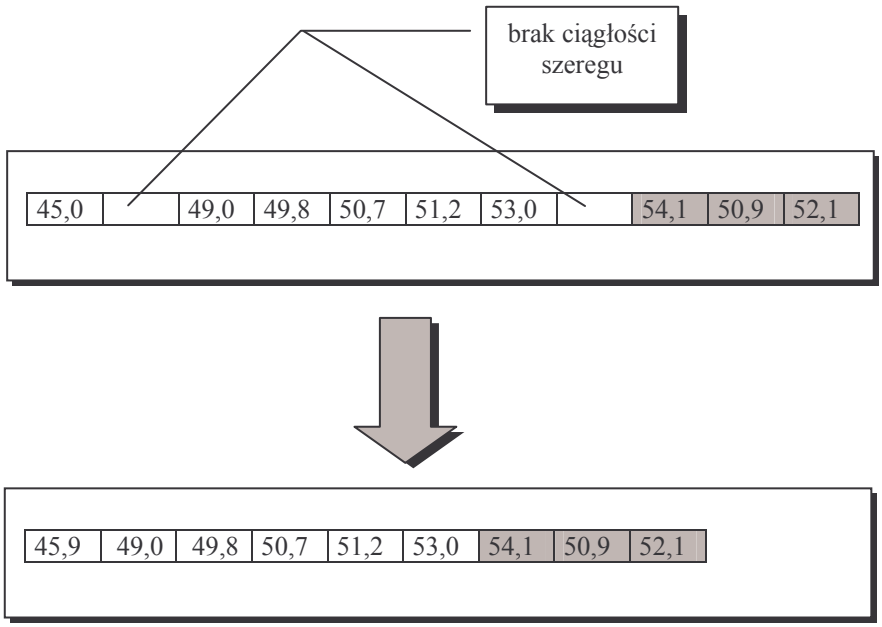
Rys. 127. Histogram szeregu w niskiej rozdzielczości

PANEL FILTRACJI SZEREGU CZASOWEGO

Zespół prostych metod korekcyjnych użytych w kreatorze prognoz umożliwia modyfikacje szeregu czasowego, czego efektem jest wzrost poprawności uzyskanych wyników prognozy oraz lepsze dopasowanie danych do wymagań sieci neuronowej. W panelu zastosowano podstawowe metody filtracji w zależności od rodzaju prognozy. Dla prognoz długoterminowych dostępna jest średnia ruchoma MA (ang. moving average) o regulowanej liczbie okresów. Dla prognoz krótkookresowych jest możliwe usunięcie trendu z szeregu za pomocą metody najmniejszych kwadratów i punktów końcowych. Ponadto jest dostępna metoda jednokrotnego różnicowania szeregu.

PRZYKŁAD PROGNOZY

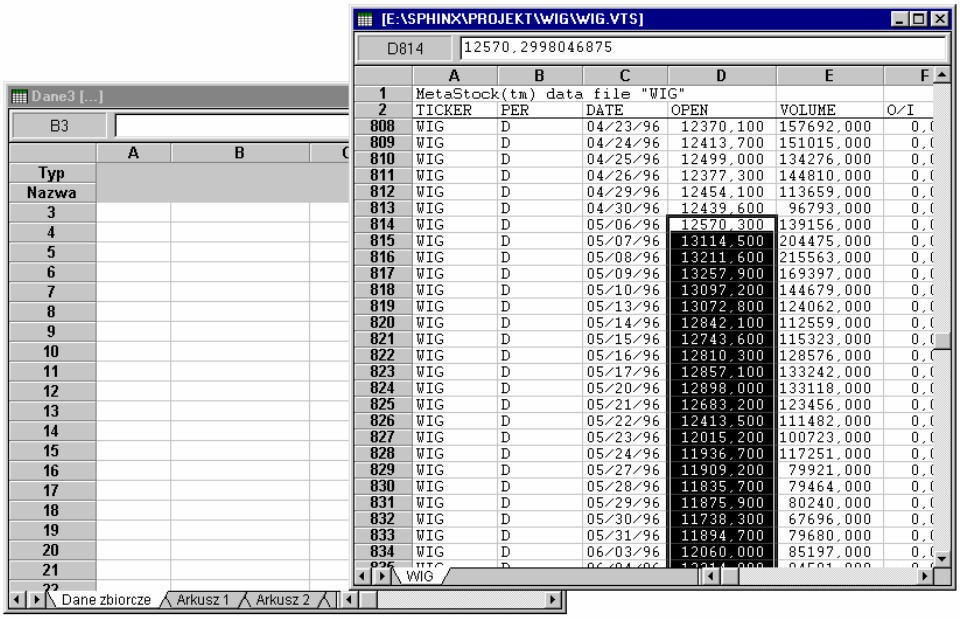
Dla wyuczenia sieci wybieramy próbkę ze zbioru wartości indeksu giełdowego WIG (Warszawski Indeks Giełdowy), który składa się dziennych zapisów wartości indeksu od roku 1991 do roku 1998. Należy pamiętać, że w początkowym okresie (1991) notowania odbywały się co 7 dni, obecnie codziennie z wyjątkiem sobót i niedziel. Zakładając w naszym przykładzie ciągłość notowań, można złączyć poszczególne notowania w jeden szereg (ciąg) czasowy. Próbka wybrana dla wyuczenia sieci liczy około 240 wartości indeksu i składa się w przybliżeniu na 1 rok notowań. Inna metoda stosowaną w szczególnych przypadkach może być próba ustalenia brakującej wartości w szeregu za pomocą metod statystycznych (na przykład średnia arytmetyczna).



Rys. 128. Usuwanie nieciągłości w szeregu czasowym

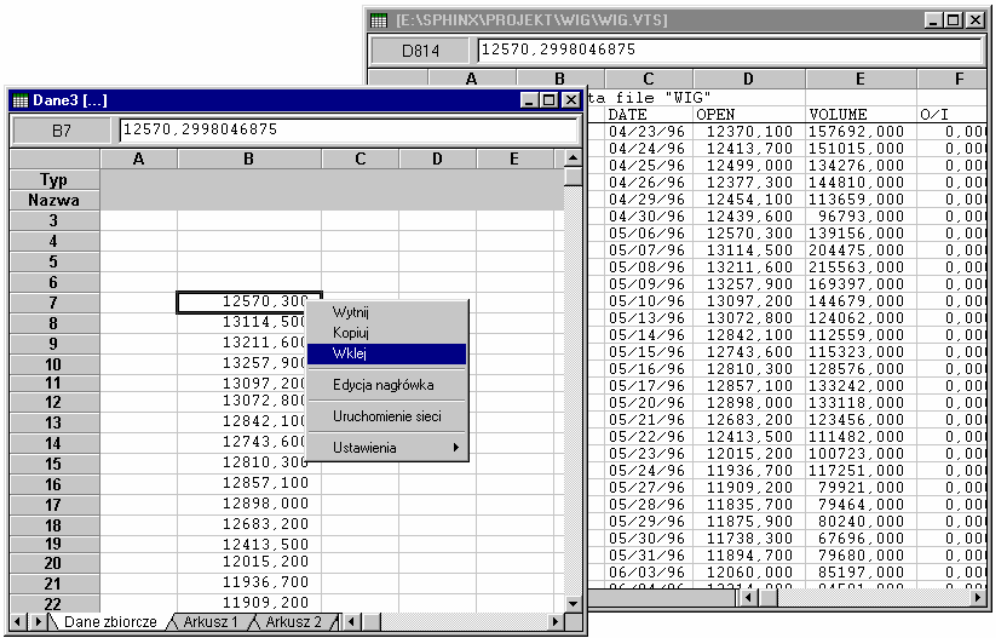
ŹRÓDŁO DANYCH

1. Jako źródło danych należy wybrać arkusz kalkulacyjny zawierający przynajmniej jedną kolumnę lub jeden wiersz danych. Arkusz ten może być w standardzie Excel'a (*.xls), Neuronix'a (*vtc). Może to być również plik tekstowy formatowany za pomocą tabulacji zawierający kolumny lub wiersze danych z dowolnym rozszerzeniem (*.*). Wymienione czynności można wykonać poprzez grupę **Plik** w menu głównym korzystając z opcji **Otwórz**.
2. Następnie otwieramy nowy arkusz, który będzie zawierał dane uczące sieci neuronowej opcją **Nowy | Arkusz danych** w grupie menu **Plik**.

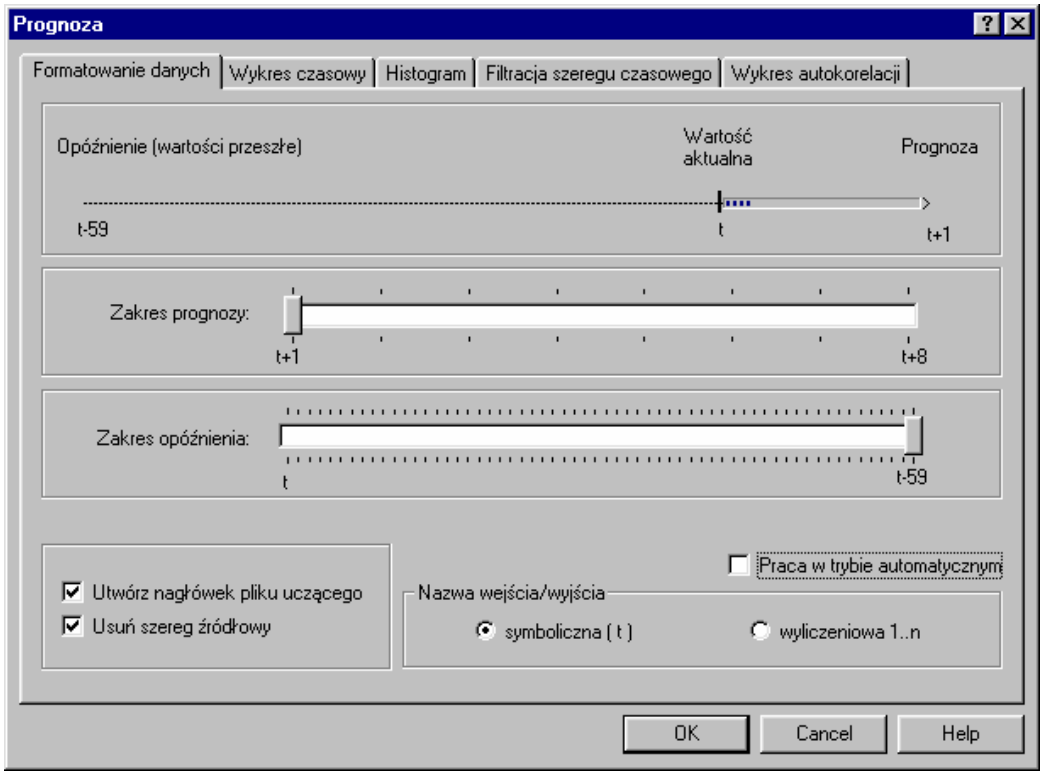


Rys. 129. Kopiowanie danych z arkusza źródłowego

3. Po otwarciu arkuszy zaznaczamy wiersz lub kolumnę i kopiujemy do arkusza docelowego, po zaznaczeniu wybranego zakresu danych uruchamiamy **Kreatora prognoz** w grupie **Narzędzia** menu głównego. Pojawia się wtedy okienko dialogowe z zakładkami.
4. Na zakładce **Formatowanie danych** ustawiamy podstawowe parametry prognozy: zakres danych historycznych, na podstawie, których prognozujemy, wyprzedzenie prognozy, czyli liczbę elementów szeregu czasowego, którą chcemy znać w przyszłości. Ponadto określamy inne parametry.



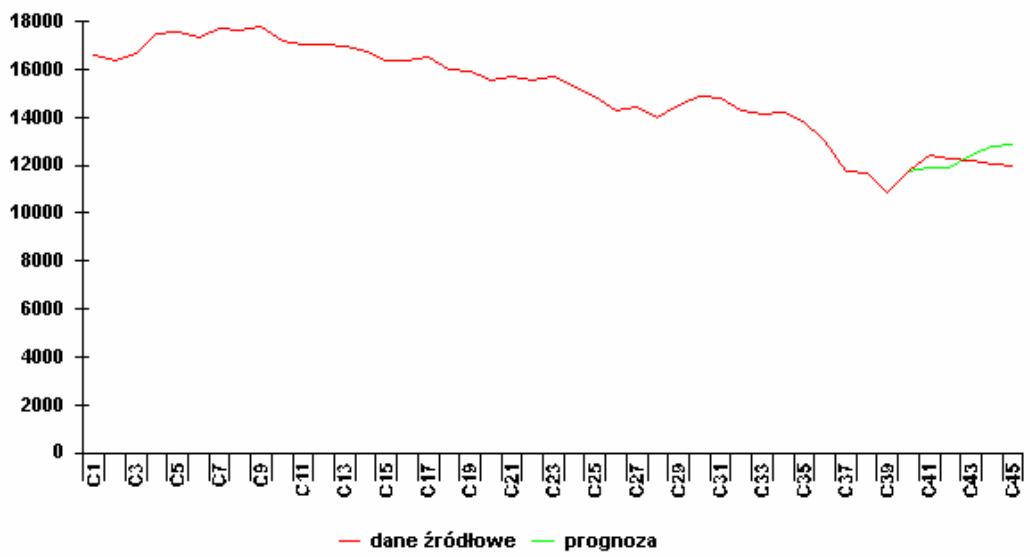
Rys. 130. Arkusze po operacji wklejania

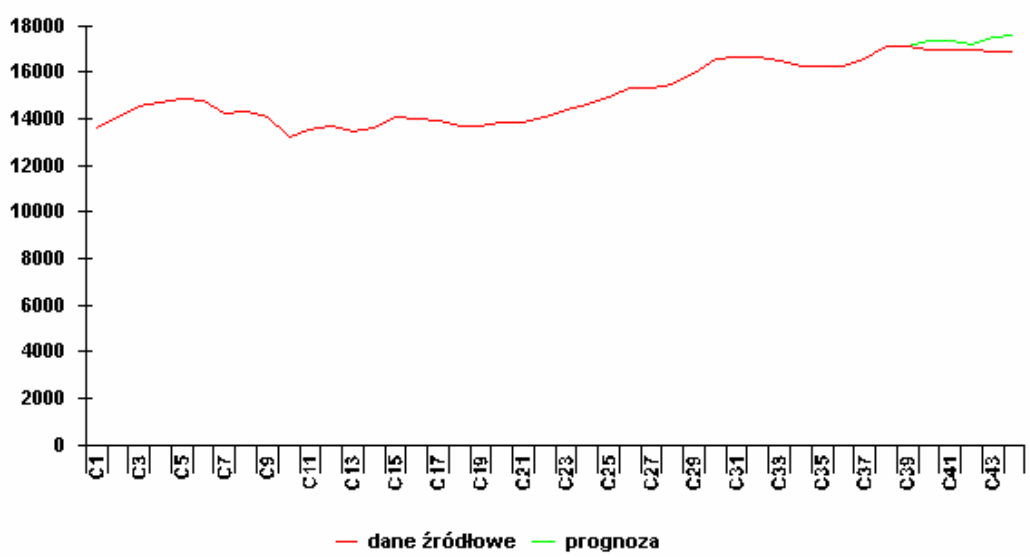


Rys. 131. Podstawowe parametry prognozy

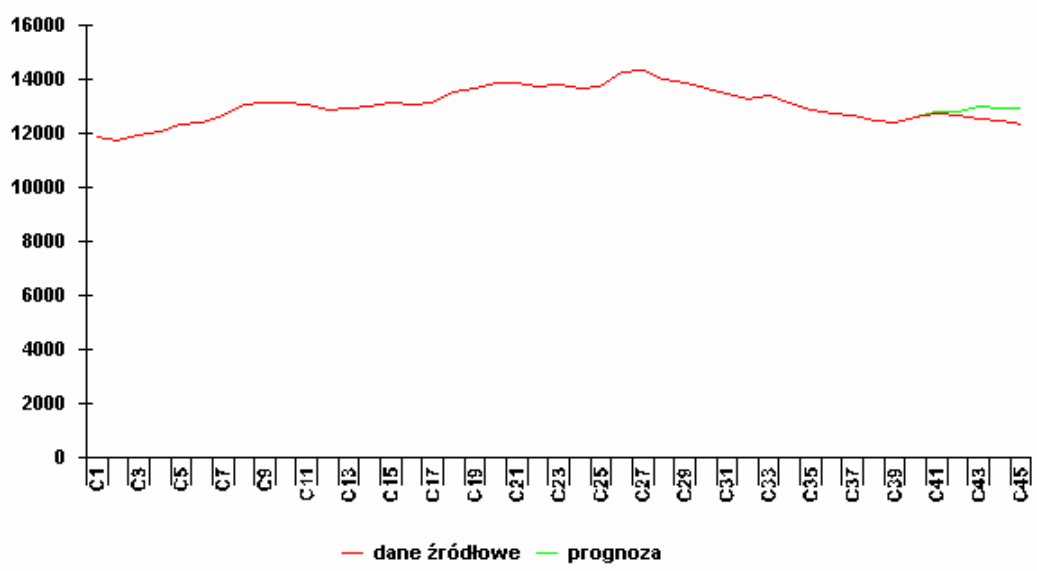
Po wyuczeniu sieci wykonano kilka przykładów prognoz dla różnych zakresów danych pochodzących ze zbioru wartości indeksu WIG. Wybrane próbki pochodzą zarówno z danych, które zostały wytypowane do uczenia sieci neuronowej, jak również z poza tego zakresu. Skuteczność prognoz można ocenić metodą jakościową oraz ilościową. W przypadku oceny jakościowej należy dokonać obserwacji wykresów źródłowego oraz prognozowanego i ustalić

subiektywnie czy wyprzedzenie prognozy (tutaj na przykład 5 dniowe) jest wystarczające dla podjęcia decyzji o zakupie bądź sprzedaży akcji, w przypadku, gdy WIG traktujemy jako barometr nastrojów panujących na giełdzie. Dla metody ilościowej najprostszym sposobem jest ocena korelacji liniowej szeregu źródłowego i prognozowanego.





Rys. 132. Dane historyczne nie zawierające się w zbiorze uczącym



Rys. 133. Dane historyczne zawierające się w zbiorze uczącym

13

Przykłady dziedzinowe

W rozdziale:

- Medycyna
- Rynki kapitałowe
- Analiza finansowa
- Redukcja zakłóceń
- Dynamiczny model strategii reklamowej

Cenne informacje

Ćwiczenia z komputerem

Koniecznie przeczytaj

MEDYCYNA

Przedstawiony model obrazuje wpływ stresu (w tym wypadku związanego z uczestnictwem mężczyzny w porodzie rodzinnym) na układ krążenia. Badane parametry to: częstość akcji serca, skurczowe i rozkurczowe ciśnienie tętnicze oraz przebieg odcinka ST w elektrokardiogramie rejestrowanym metodą Holtera (24 godzinny zapis ekg). Podstawowym elementem powodującym zmiany czynności układu krążenia w sytuacji stresowej jest część współczulna układu autonomicznego. Podczas wysiłku fizycznego sprawcą zmian jest przede wszystkim noradrenalina, w sytuacji interesującego nas stresu psychicznego jest nim adrenalina, oddziaływująca na receptory β_1, β_2 i α zlokalizowane w sercu i naczyniach tętniczych. Na stopień tego oddziaływania wpływa między innymi:

- wiek badanych, stan zdrowia (dobry, średni, zły),
- ewentualnie stosowanie leków (trakwilizatory),
- stopień stresu (w 10 stopniowej skali subiektywnej),
- współczynnik BMI wyrażający masę ciała:

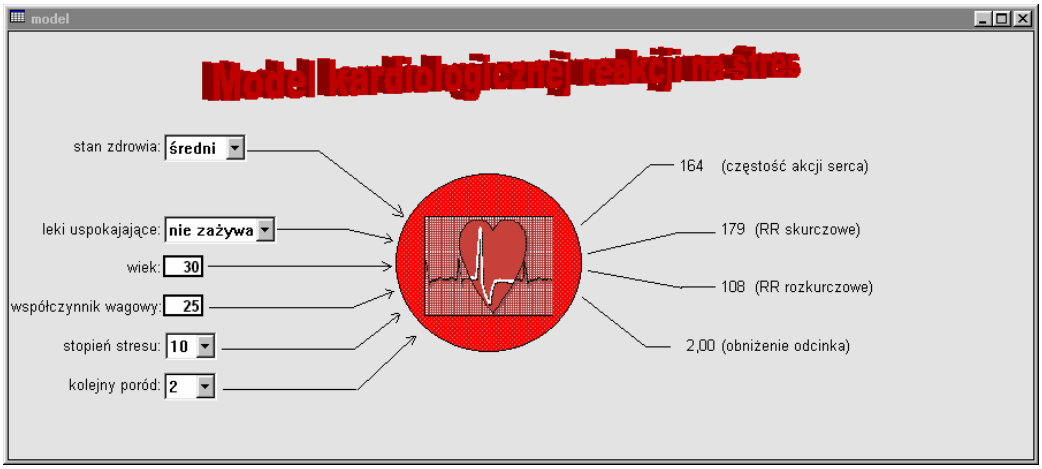
(13-1) $$BMI = \frac{M}{h^2},$$

gdzie:
M - masa ciała w kg,
h - wzrost w metrach,

Interpretacja współczynnika BMI:

- BMI < 25 - normalna waga ciała,
- 25<BMI ≤ 30 - nadwaga,
- BMI> 30 - otyłość.

Odcinek ST elektrokardiogramu jest odcinkiem przebiegającym fizjologicznie w linii izoelektrycznej (w ekg obrazuje okres pomiędzy depolaryzacją i repolaryzacją komór). Uniesienie lub obniżenie tego odcinka o 1mm (0,1 mV) w stosunku do tej linii uważane jest za nieprawidłowe i może odpowiadać zmianom niedokrwiennym, które pojawiają się lub pogłębiają w czasie stresu.

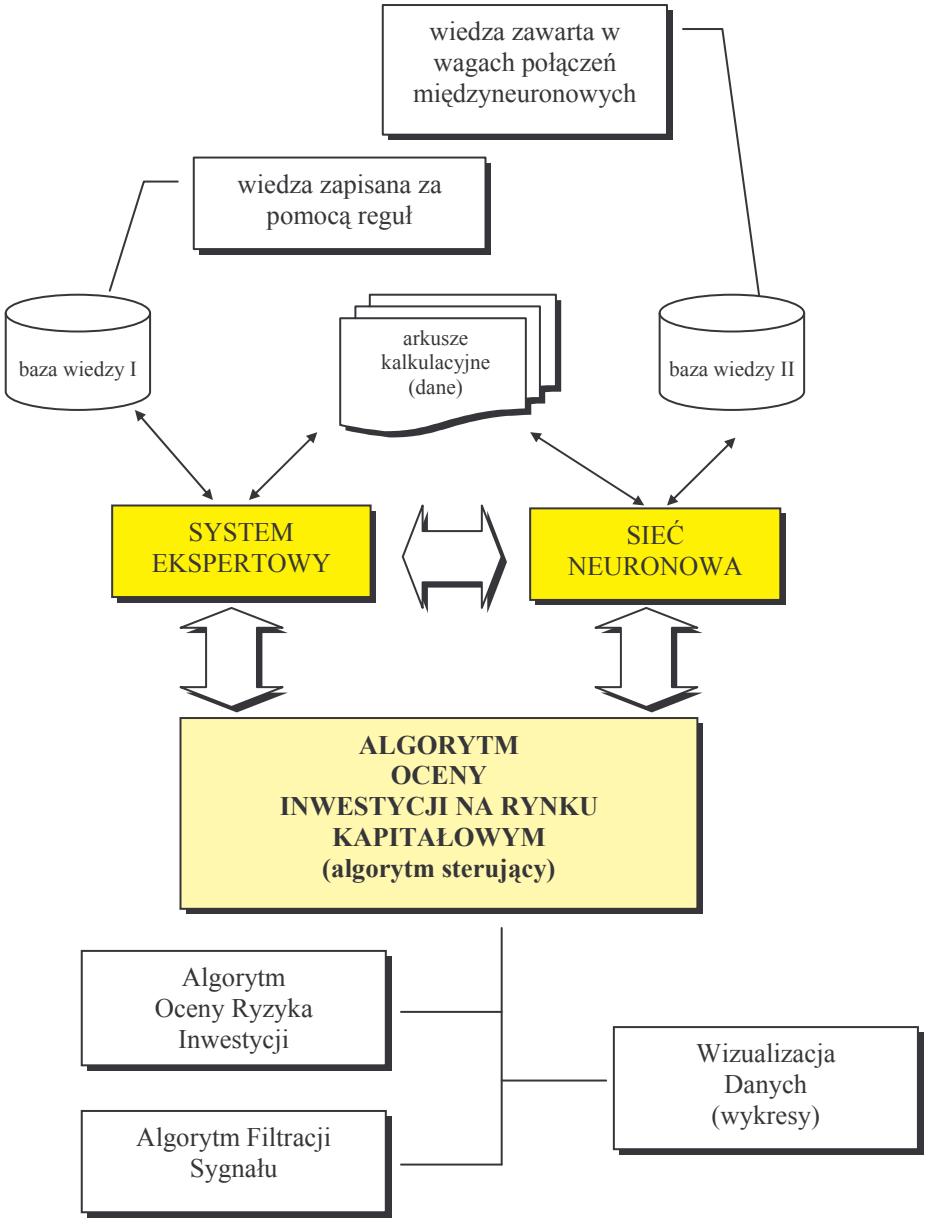


Rys. 134. Kardiologiczny model reakcji na stres

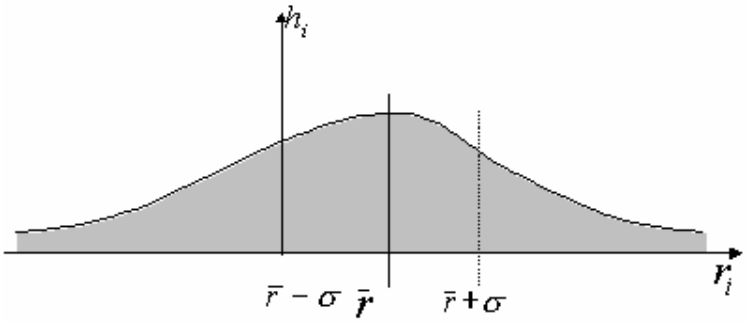
RYNKI KAPITAŁOWE

Aplikacja demonstracyjna „Doradca kapitałowy" pozwala na zaprezentowanie istotnej cechy pakietu Sphinx jaką jest **hybrydowość**. Poprzez pojęcie hybrydowości będziemy rozumieć współpracę poszczególnych komponentów sztucznej inteligencji, na przykład: systemu ekspertowego i sieci neuronowej. Współpraca będzie polegać na **lokalnym rozwiązywaniu problemów** (w obrębie poszczególnych komponentów) oraz na wypracowaniu wspólnego **globalnego rozwiązania problemu**. Podczas takiej współpracy niezbędna jest efektywna komunikacja komponentów pomiędzy sobą, umożliwia to język makroprogramowania pakietu Sphinx. W ramach aplikacji hybrydowej, poza komponentami sztucznej inteligencji mogą współpracować tradycyjne komponenty w postaci modułów zawierających

wysoko specjalizowane **algorytmy** rozwiązujące cząstkowe problemy lub generujące rozwiązania lokalne alternatywne dla metod sztucznej inteligencji.



OCENA RYZYKA INWESTYCJI



Rys. 135. Rozkład normalny a ryzyko inwestycji

W celu oszacowania ryzyka inwestycji kapitałowej można ryzyko utożsamić z odchyleniem standardowym. Pozwoli to na zdefiniowanie pola o szerokości $\bar{r}-\sigma$ i $\bar{r}+\sigma$. Ryzyko wyznaczamy najczęściej dla stopy zwrotu z inwestycji o wybranym okresie najczęściej miesięcznym lub rocznym. Definicja stopy zwrotu przedstawia się nastepująco

(13-2) $$r=\frac{\sum K_k-\sum K_p}{\sum K_p}\cdot 100\%,$$

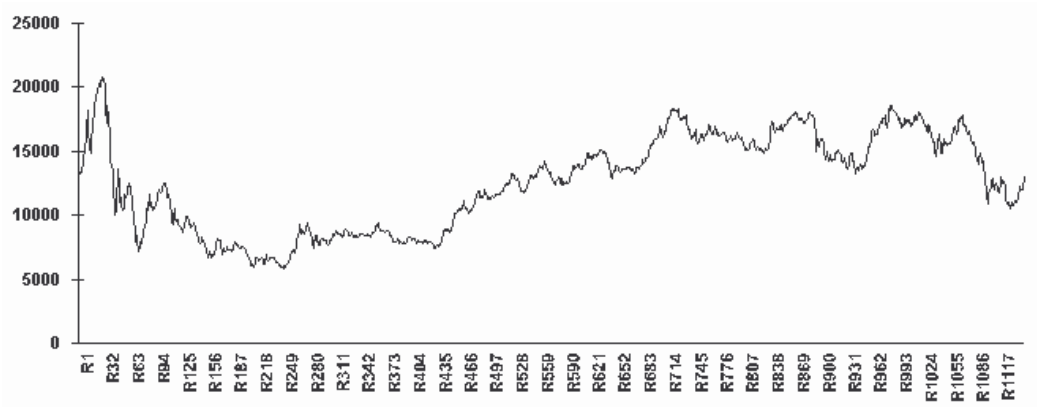
przez $\sum K_p$ oznaczamy sumę kapitałów na początku okresu inwestycji, natomiast $\sum K_k$ oznacza sumę kapitaów na końcu okresu.

Średnia stóp zwrotu za określony okres można obliczać na dwa sposoby. Jako średnią arytmetyczną, bardziej przydatną dla inwestycji krótkoterminowych oraz jako średnią geometryczną zalecana dla inestycji długoterminowych

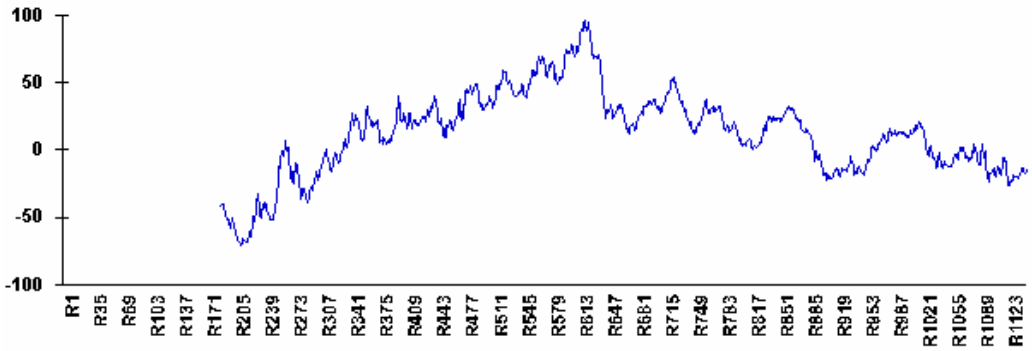
(13-3) $$\bar{r}=\frac{\sum_{i=1}^{n} r_i}{n},$$

(13-4) $$\bar{r}=\sqrt[n]{\prod_{i=1}^{n}(1+r_i)}-1$$

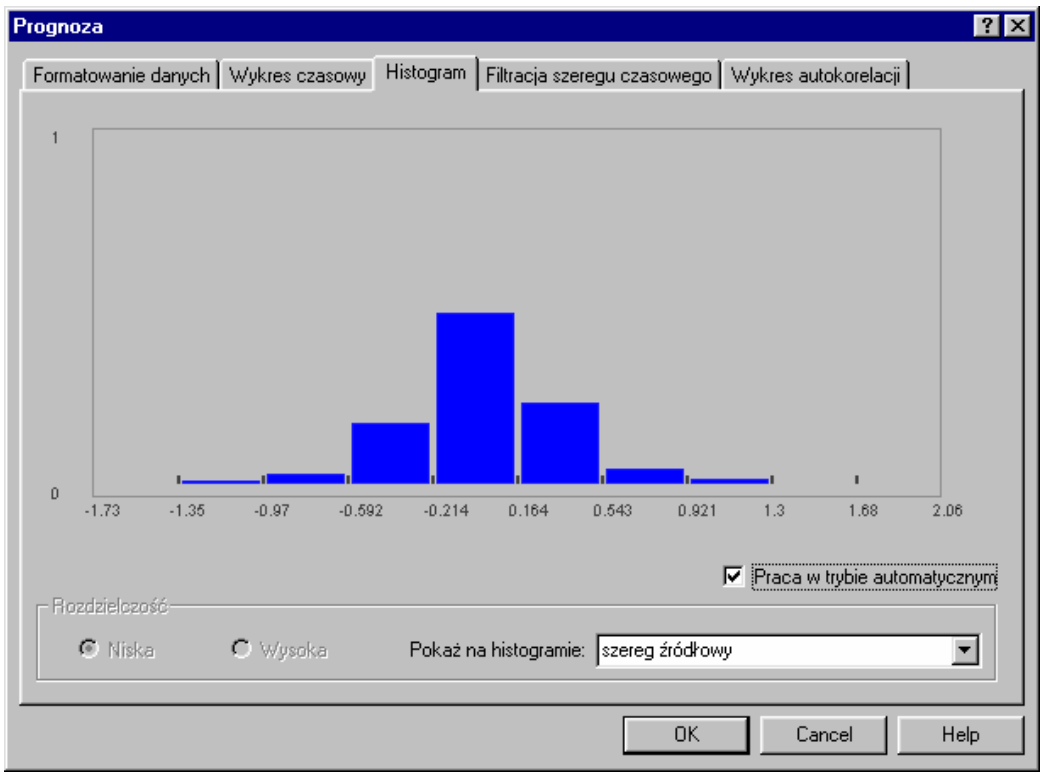
Zachodzi pytanie, czy można przyjąć, że rozkład stóp zwrotu z inwestycji kapitałowych na giełdach jest normalny ? Otóż jak dowiodły testy warunek ten jest spełniony dla większości rynków giełdowych na świecie.



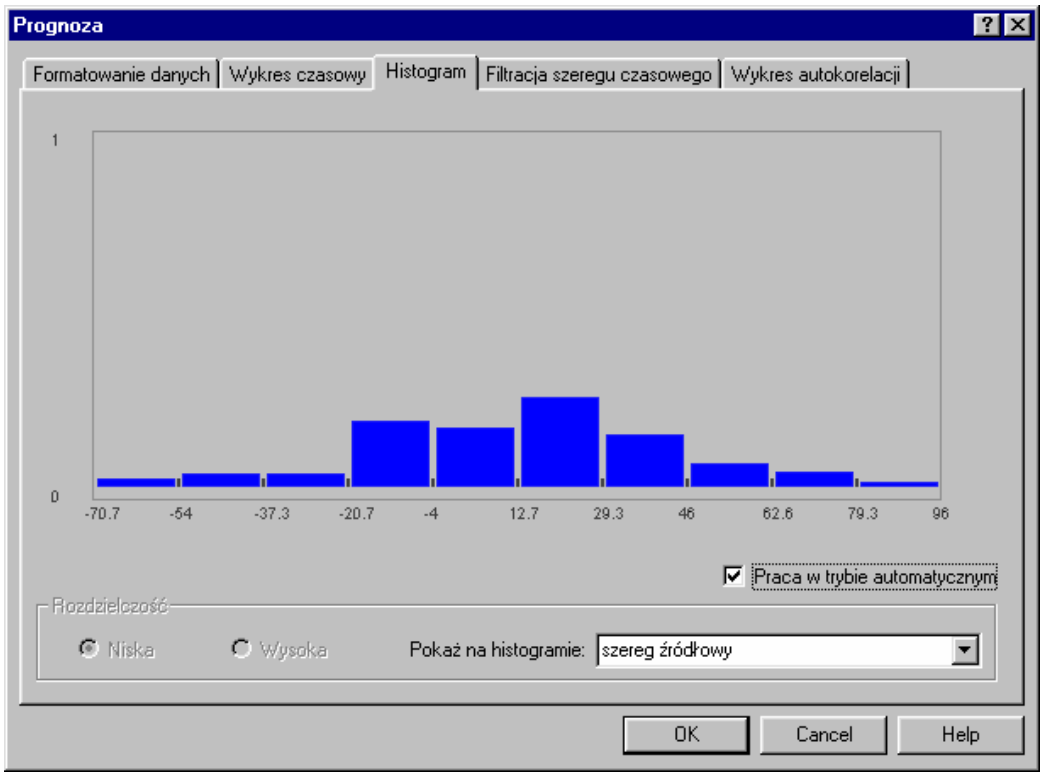
Rys. 136. Wykres wartości szeregu czasowego indeksu WIG



Rys. 137. Wykres szeregu czasowego rocznej stopy zwrotu z indeksu WIG



Rys. 138. Rozkład miesięcznych stóp zwrotu z indeksu WIG

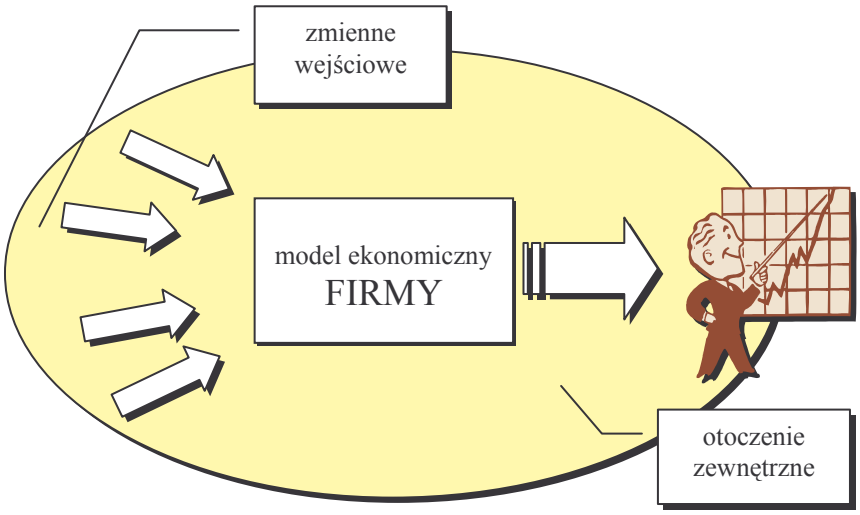


Rys. 139. Rozkład rocznych stóp zwrotu z indeksu WIG

ANALIZA FINANSOWA

Zastosowanie sieci neuronowej dla szybkiej oceny kondycji finansowej przedsiębiorstwa jest jednym z istotnych problemów dziedzinowych. Ze względu na duże znaczenie problemu utrzymania właściwej jakości wskaźników oceny płynności, zostanie opisany przykład aplikacji stosującej dwa narzędzia: system ekspertowy oraz sieć neuronową dla oceny zagrożeń utraty płynności przez firmę. W ramach serii **Doradcy** znajduje się aplikacja demonstracyjna:

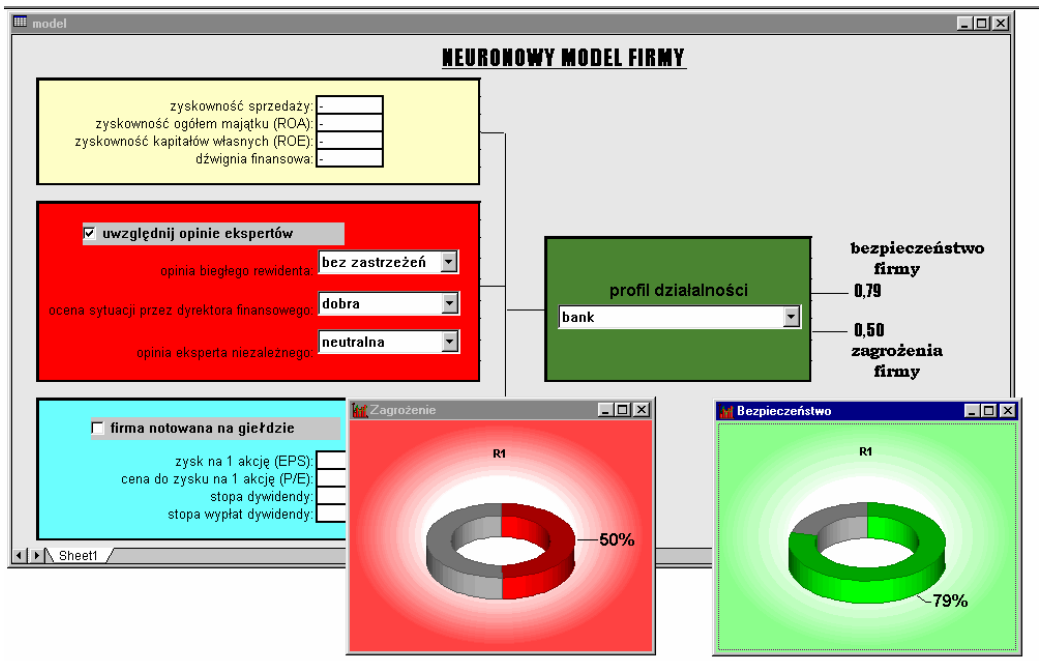
Doradca finansowy, która zawiera wspomniane wcześniej narzędzia analizujące płynność finansową firmy. Prezentowany przykład umożliwia przyszłe teoretyczne i praktyczne uogólnienie metodologii postępowania na całościową **analizę ekonomiczną** firmy. Dla firm pozostających w niestabilnym i niepewnym otoczeniu zewnętrznym informacja przedstawiona w postaci syntetycznego wskaźnika jakości kondycji finansowej firmy może mieć duże znaczenie. Szybka diagnoza stanów niebezpiecznych pozwoli zasygnalizować konieczność przeprowadzenia bardziej szczegółowych ekspertyz.



Rys. 140. Idea szybkiej oceny symptomów stanu firmy

MODELE W POSTACI SZAREJ I CZARNEJ SKRZYNKI

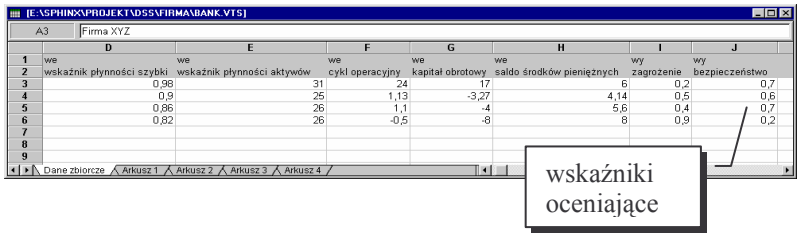
W przypadku budowy modelu finansowego firmy (w ogólnym przypadku ekonomicznego) możliwe są różne postacie tych modeli. Związane jest to ze sposobem estymacji parametrów modelu. W przypadku problemów finansowych można budować modele w postaci jawnej, typowym przedstawicielem tej grupy są modele tworzone z myślą o systemach ekspertowych. Identyfikacja parametrów w tych modelach nie jest potrzebna, gdyż wiedza zapisana jest w postaci reguł, dla których z góry określono pewne wartości parametrów, a Użytkownik wprowadza tylko pewne zmienne w celu uzyskania rozwiązania. W przypadku modeli w postaci „**czarnej skrzynki”** sytuacja jest odwrotna, Użytkownik dysponuje zbiorem danych, lecz nie posiada informacji o zależnościach występujących pomiędzy nimi. Jest to typowe zadanie dla sieci neuronowej. Po wyznaczeniu struktury modelu, czyli zdefiniowaniu wnętrza „**czarnej skrzynki**” następuje proces odpytywania sieci dla konkretnych zmiennych wejściowych. W szczególnych przypadkach można mówić o modelach w postaci „**szarej skrzynki**”, które posiadają częściowo wyznaczoną strukturę lub parametry. Dla budowy tego typu modeli najlepiej nadają się systemy hybrydowe w których sieć „zajmuje się szarą częścią” modelu, natomiast system ekspertowy pracuje podczas poszukiwania rozwiązania finalnego.



Rys. 141. Fragment modelu w postaci „czarnej skrzynki”

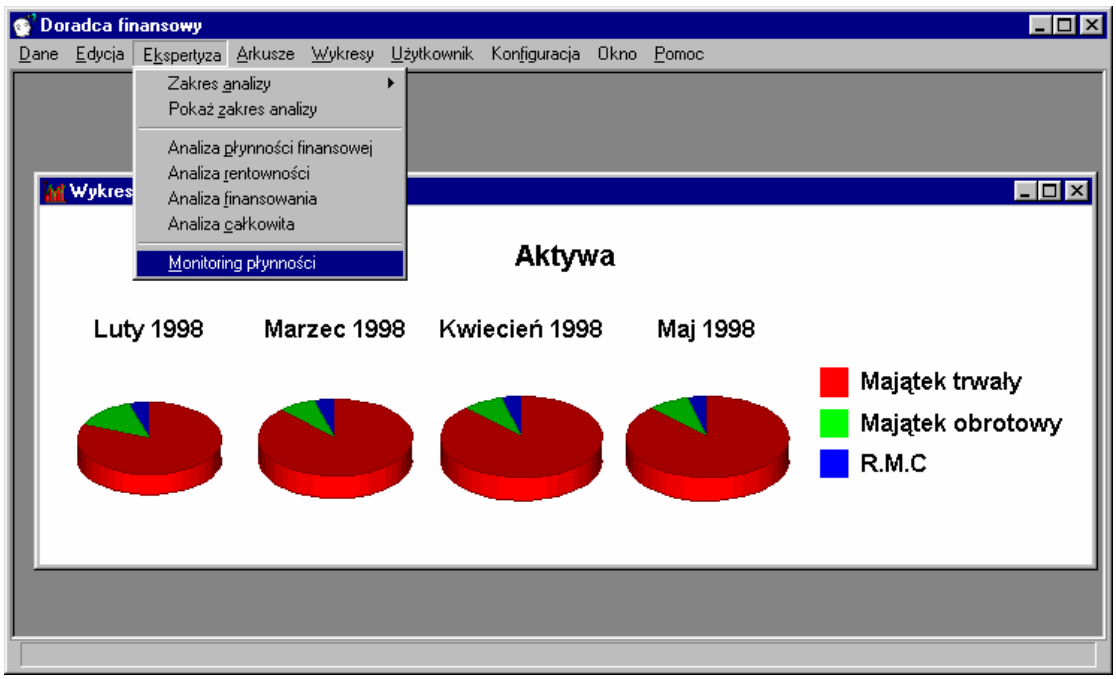
SPOSOBY OCENY

Istnieje wiele sposobów oceny sytuacji finansowej firmy. W przypadku aplikacji demonstracyjnej „Doradca finansowy” zdecydowano się na użycie dwóch niezależnych wskaźników w postaci liczb z zakresu 0..1 oceniających zagrożenie oraz bezpieczeństwo. Ocenie poddano płynność finansową firmy.

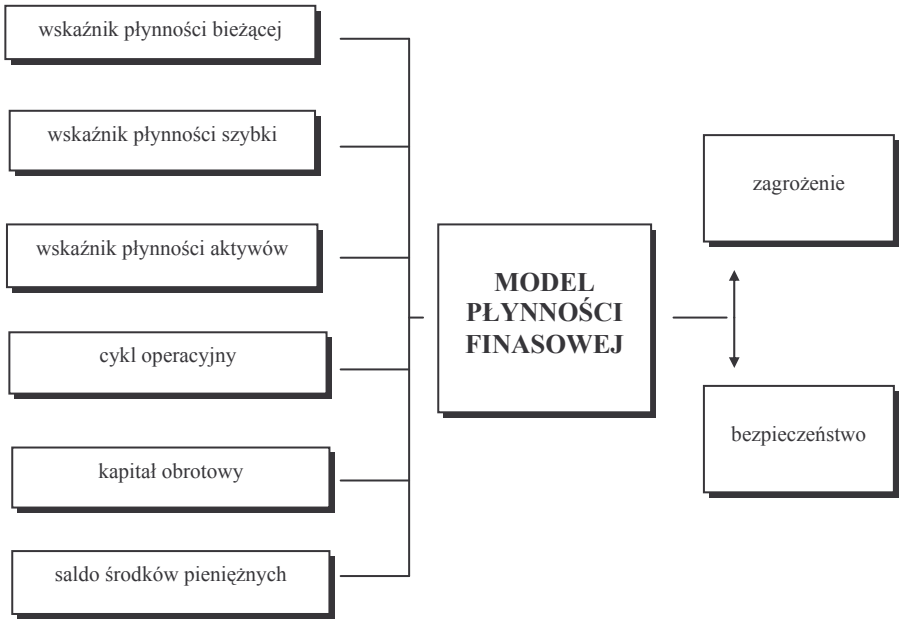


Rys. 142. Dane uczące

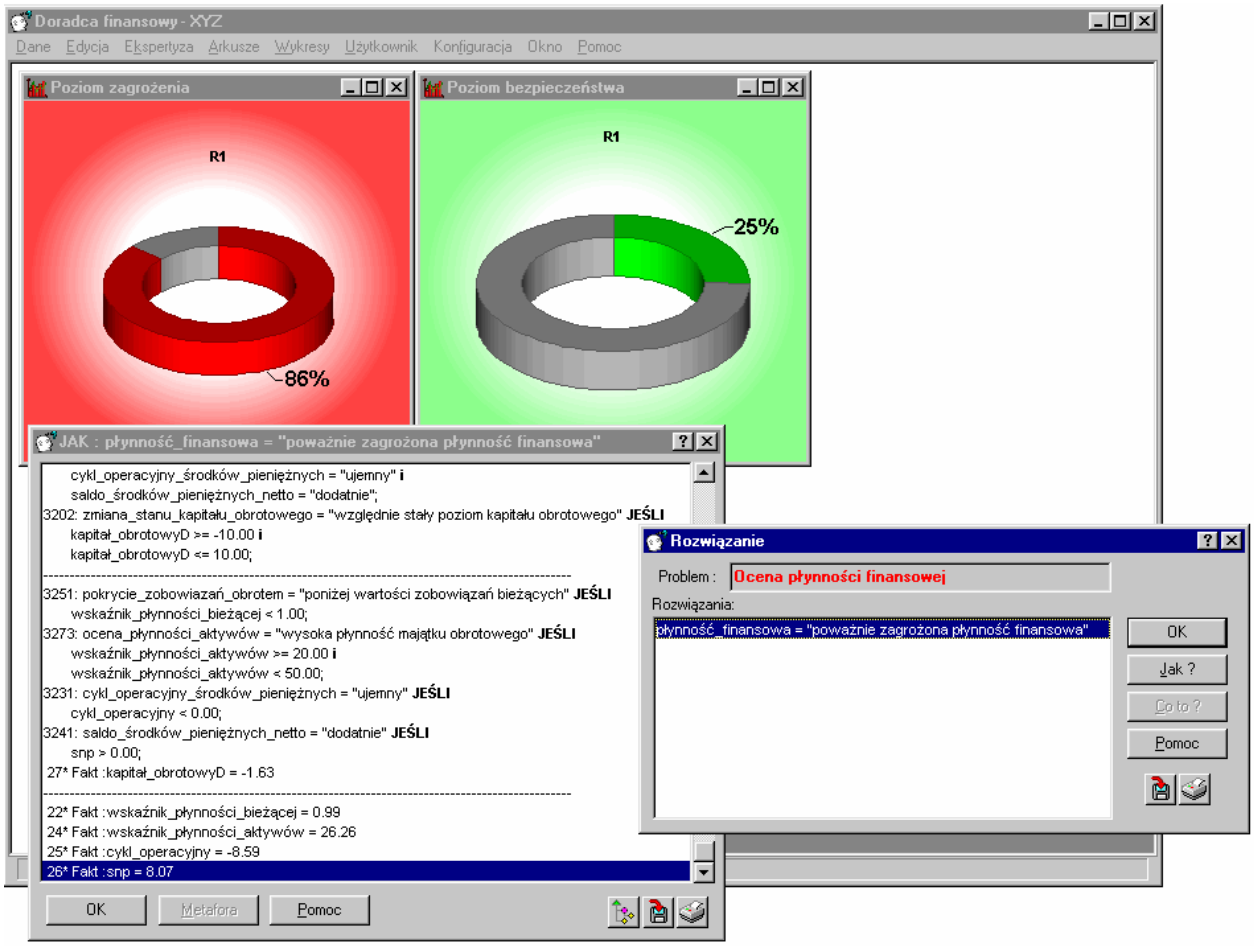
Wartości wskaźników dobiera ekspert dziedzinowy przed rozpoczęciem procesu uczenia sieci neuronowej. Dla wybranego problemu w ujęciu praktycznym, na przykład dla sygnalizacji zagrożenia płynności wystarczający jest pojedynczy wskaźnik. Dostęp do wyuczonej sieci odbywa się za pomocą języka Sphinx. Użytkownik otrzymuje tylko dostęp do opcji uruchamiającej proces monitoringu płynności w określonych odstępach czasu.



Rys. 143. Opcja uruchamiająca monitoring



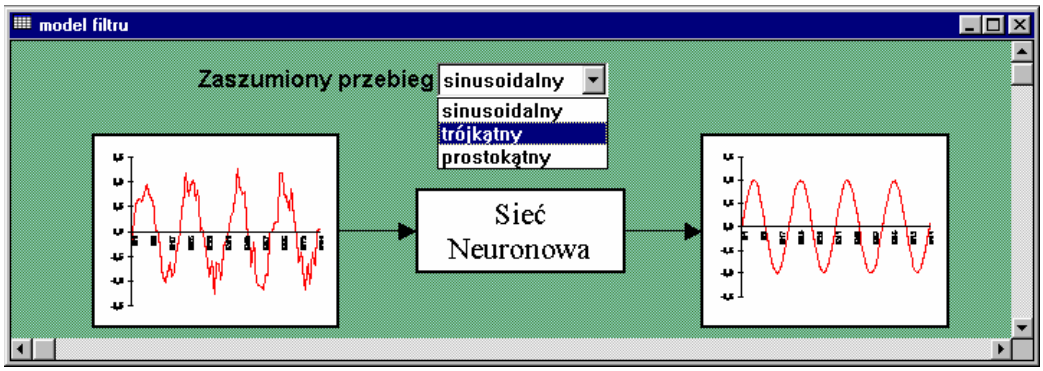
Rys. 144. Schemat modelu oceny płynności finansowej



Rys. 145. Dwie metody oceny płynności firmy

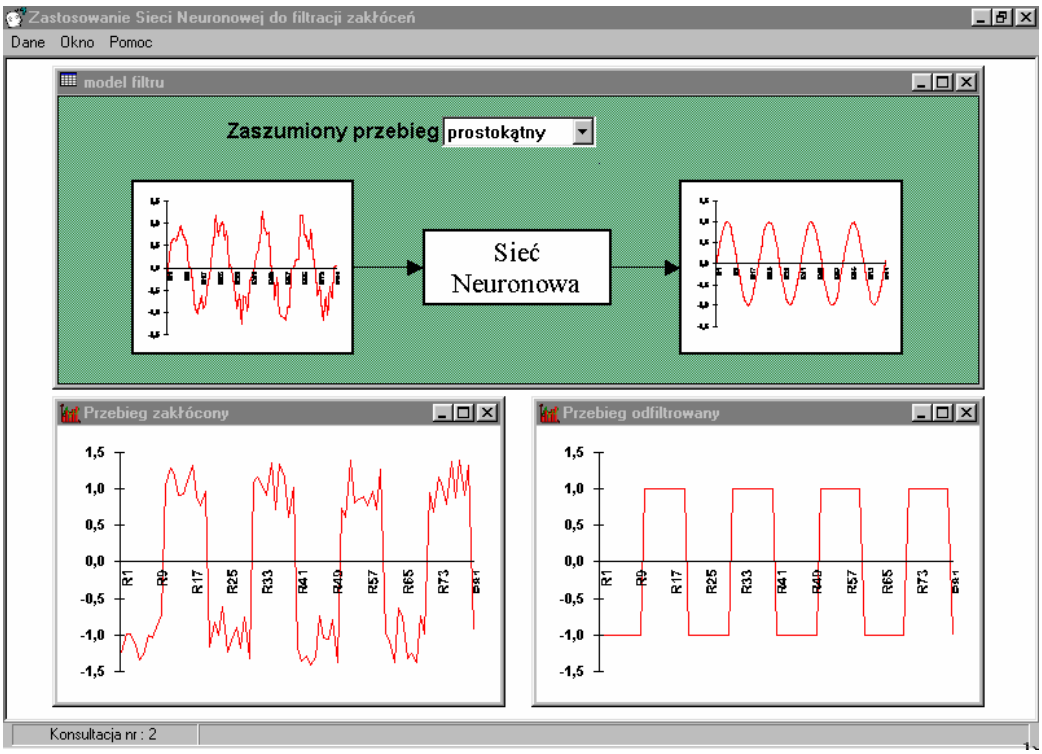
REDUKCJA ZAKŁÓCEŃ

Aplikacja napisana została w środowisku PC-Shell. Celem jest prezentacja możliwości zastosowania sieci neuronowych do filtracji zakłóceń nałożonych na najczęściej wykorzystywane w technice sygnały



Rys. 146. Symulacja filtru

Aby uruchomić aplikację należy wybrać interesujący nas przebieg z listy (tak jak pokazano na powyższym rysunku). Następnie z głównego menu należy wybrać opcję **Dane | Odfiltruj przebieg**

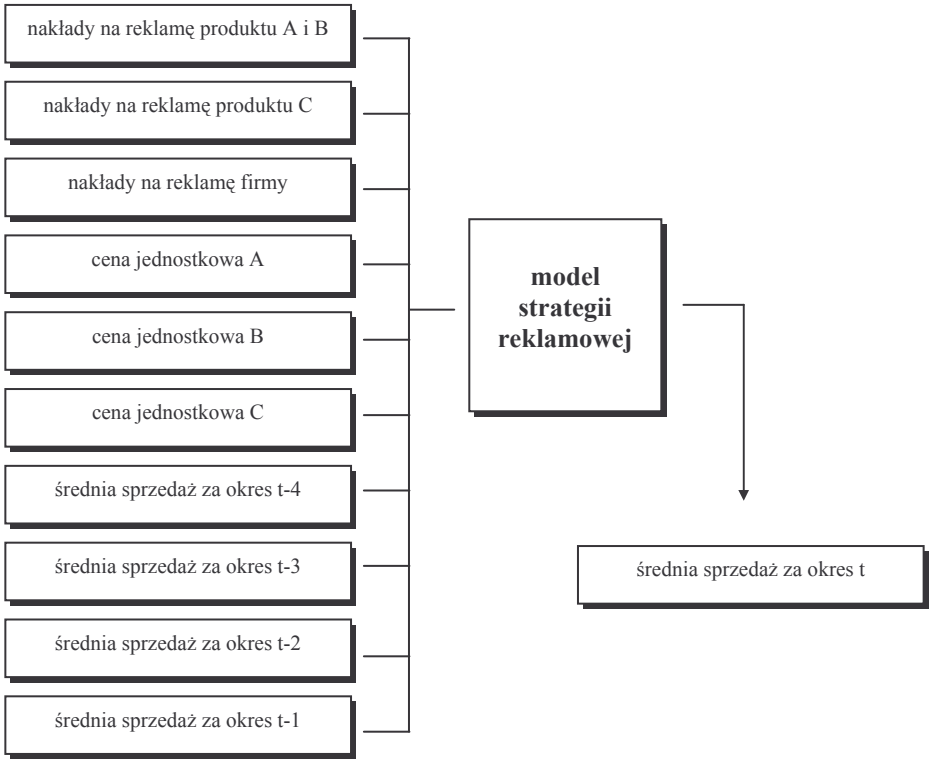


Rys. 147. Demonstracyjna aplikacja filtrująca zakłócenia

Po uruchomieniu na wykresach pojawiają się przebiegi ilustrujące skuteczność filtracji zakłóceń.

DYNAMICZNY MODEL STRATEGII REKLAMOWEJ

W celu przeanalizowania wpływu wydatków reklamowych na skuteczność reklamy zbudowano uproszczony model strategii reklamowej polegającej na poszukiwaniu dodatkowych rynków zbytu i poszerzenie kręgu klientów za pomocą reklamy w określonym typie mediów.



Rys. 148. Model strategii reklamowej

Dla oceny strategii wybrano trzy charakterystyczne produkty firmy A, B i C. Produkty A i B charakteryzują się podobnymi cechami użytkowymi oraz ceną i są przeznaczone dla typowego klienta. Produkt C jest skierowany do bardzo wymagających użytkowników i charakteryzuje się znacznie wyższą ceną i wysoką jakością. Dla powyższych produktów zgodnie z założeniami modelu zgromadzono potrzebne dane.

[E:\SPHINX\PROJEKT\ANALIZA WRAżLIWOśCI\DANE.VTS]

B23

	A	B	C	D	E	F
1	-	we	we	we	-	we
2	okres	nakłady na reklamę firmy	nakłady na reklamę produktu A i B	nakłady na reklamę produktu C	sprzedaż produktu A [szt.]	cena jednostkowa A
3	01.98	100	0	0	608	2
4	02.98	0	20	40	769	2
5	03.98	0	30	60	889	3
6	04.98	50	50	50	1208	3
7	05.98	0	0	0	1500	3
8	06.98	0	0	0	1306	3
9	07.98	100	0	0	1289	3
10	08.98	0	10	20	1023	3
11	09.98	0	10	10	1277	3
12	10.98	30	0	0	1389	3
13	11.98	10	0	0	1316	3
14	12.98	0	20	20	1190	3
15	01.99	0	20	20	1203	3
16	02.99	0	0	0	1178	3
17	03.99	0	0	0	1156	3
18	04.99	0	0	0	1107	3
19	05.99	0	0	0	1043	3
20						
21						

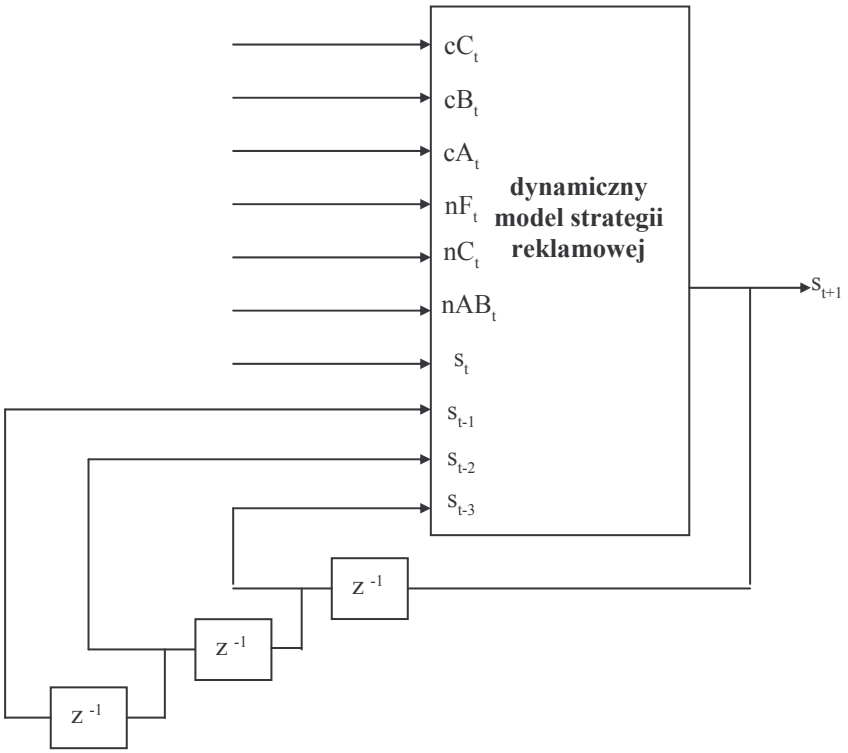
Dane zbiorcze / Arkusz 1 / Arkusz 2 / Arkusz 3 / Arkusz 4

Rys. 149. Zgromadzone dane uczące

Model można opisać równaniem różnicowym:

$$s_{t+1} = s_t + s_{t-1} + s_{t-2} + s_{t-3} + nAB_t + nC_t + nF_t + cA_t + cB_t + cC_t,$$

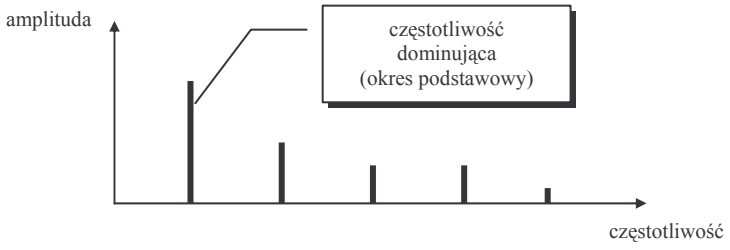
w którym przyjęto, że wartość średnia sprzedaży oznaczona jako s, w przyszłym okresie t+1 zależy od poprzednich wartości średniej sprzedaży w okresach: t, t-1, t-2, t-3 oraz od nakładów na reklamę nAB, nC, nF w okresie bieżącym t. Ponadto zależy od ceny jednostkowej produktu cA, cB, cC również w okresie bieżącym t.



Rys. 150. Schemat modelu

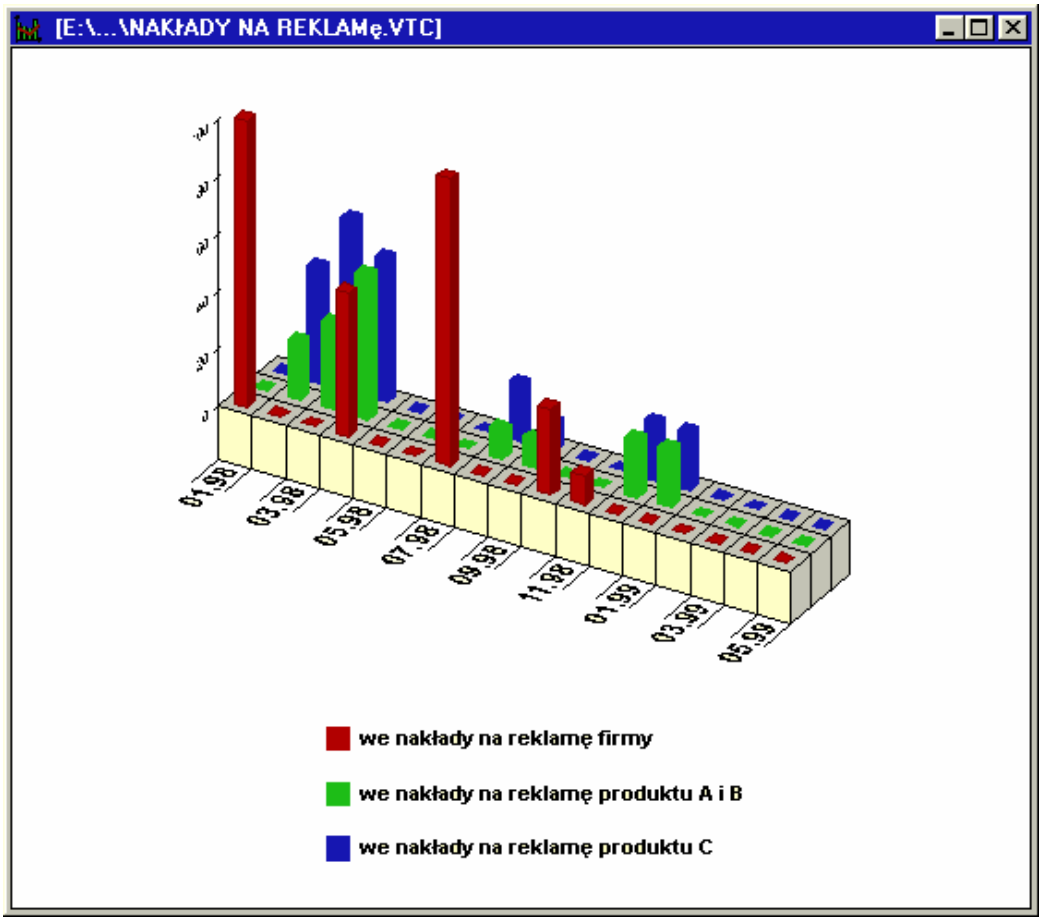
Model pozwoli ocenić, w jakim stopniu opłacalne jest zwiększanie nakładów na reklamę w poszczególnych segmentach oferowanych produktów (A, B i C) oraz na ogólny wizerunek firmy. Cechą szczególną modelu jest **wprowadzenie opóźnienia** o pewną liczbę okresów podanego w postaci **sygnału sprzężenia zwrotnego** na wejście modelu. Wartość opóźnienia można oszacować subiektywnie lub poprzez zastosowanie specjalnych metod, na

przykład **transformaty Fouriera**, która pozwala na podstawie **widma sygnału** (obrazu w dziedzinie częstotliwości) ocenić występujące w nim okresowości. Arbitralnie przyjęto opóźnienie równe okresowi składającemu się z czterech jednostek ewidencyjnych (miesięcy). Przykład identyfikowania okresowości w sygnale przedstawia rys. xxx. W rzeczywistych sygnałach w zależności od zastosowanej metody wizualizacji widma (tzw. okna) występują czasami problemy z jednoznacznym określeniem okresu podstawowego sygnału.

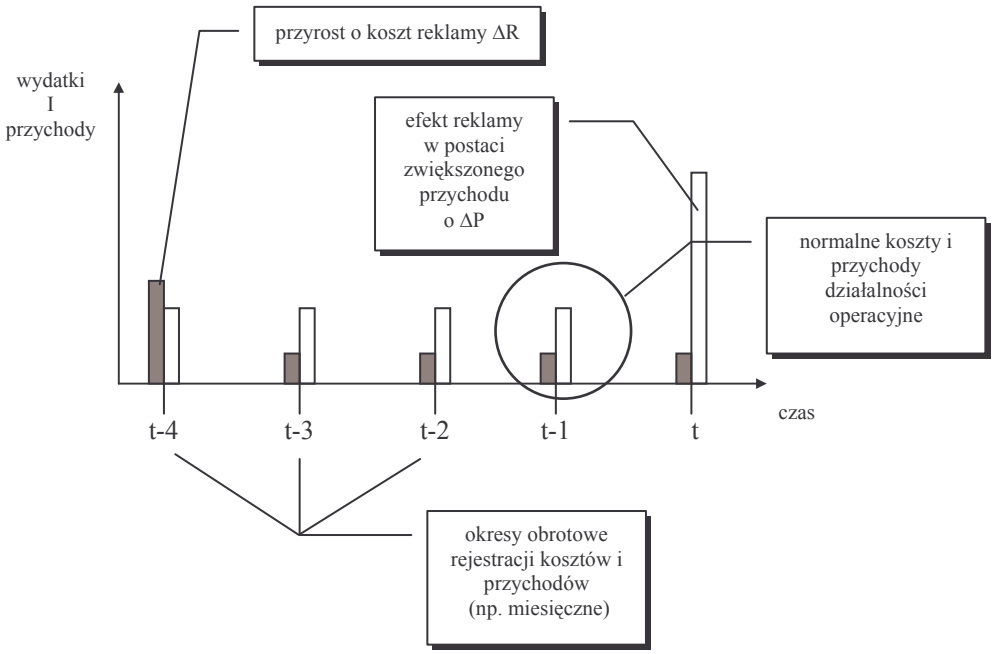


Rys. 151. Wyznaczanie okresu podstawowego dla sygnału harmonicznego

W rozważanym modelu mogą występować duże, zależne od sposobu reklamy opóźnienia czasu reakcji wzrostu wartości przychodu na źródło jego zwiększenia. Sytuację taką przedstawia rys. 153, w rzeczywistych warunkach jest ona spowodowana opóźnieniami w ukazywaniu się reklam na przykład o czas druku i przygotowania.



Rys. 152. Nakłady na reklamę w poszczególnych okresach ewidencyjnych



Rys. 153. Opóźnienie podczas realizacji strategii marketingowej

14

Błędy sygnalizowane w systemie Neuronix

W rozdziale:

- szablon błędu
- wykaz błędów

SZABLON BŁĘDU

Program Neuronix informuje użytkownika o zaistniałych nieprawidłowych sytuacjach przy użyciu okien komunikatów. Format komunikatu o błędzie ma następującą postać:

numer_ błędu: opis_ błędu [dodatkowy_ opis]

gdzie:
numer_błędu - wewnętrznym oznaczeniem sytuacji nieprawidłowej,
opis_błędu - słowny opis nieprawidłowości,
dodatkowy_opis - informacje dodatkowe, które pozwalają zlokalizować miejsce wystąpienia błędu; opis ten pojawią się tylko przy niektórych rodzajach błędów np. przekroczeniu dopuszczalnej długości symbolu w jednej z kolumn wejściowych lub wyjściowych

WYKAZ BŁĘDÓW

Lp	*Opis*	*Dodatkowy opis*
1	*Znak końca pliku w niedozwolonym miejscu!* Błąd zwracany w sytuacji, gdy znak końca pliku pojawia się tam, gdzie oczekiwana jest inna wartość. Przykładowo może pojawić się gdy w ostatnim wierszu pliku uczącego nie istnieje wartość dla ostatniej kolumny. Może się również pojawić w przypadku trudności odczytu innych plików.	
2	*Zbyt długi tekst!* Błąd zwracany jest w sytuacji, gdy długość tekstu w jednej z kolumn przekroczył maksymalną dopuszczalną długość 255 znaków. Pojawia się gdy: - wprowadzono tekst dłuższy niż 255 znaków w pliku uczącym lub testowym, - łączna długość nazwy kolumny i jednego z jej symboli przekracza 255 znaków	*[wiersz, kolumna]* Miejsce wystąpienia błędu w pliku uczącym lub testowym.
3	*Ilość kolumn w pierwszym wierszu jest mniejsza niż w wierszach następnych!* Błąd zwracany w sytuacji, gdy pierwszy wiersz pliku uczącego lub testowego mniej wartości niż w wierszach następnych. Symulator Neuronix wymaga, aby liczba wartości w każdym wierszu była stała, co umożliwia mu przypisanie ich do odpowiednich kolumn.	*[wiersz, kolumna]* Miejsce wystąpienia błędu w pliku uczącym lub testowym.
4	*Ilość kolumn w pierwszym wierszu jest większa niż w wierszach następnych!* Błąd zwracany w sytuacji, gdy pierwszy wiersz pliku uczącego lub testowego zawiera więcej wartości niż w wierszach następnych. Symulator Neuronix wymaga, aby liczba wartości w każdym wierszu była stała, co umożliwia mu przypisanie ich do odpowiednich kolumn.	*[wiersz, kolumna]* Miejsce wystąpienia błędu w pliku uczącym lub testowym.

5	*Kolumna zawiera wartości różnego typu - numeryczne i symboliczne!* Błąd zwracany w sytuacji, gdy w ramach tej samej kolumny występują wartości numeryczne i symboliczne. Symulator Neuronix przyjmuje, że typ wartości w kolumnie jest taki jak typ pierwszej występującej w niej wartości.	*[wiersz, kolumna]* Miejsce wystąpienia błędu w pliku uczącym lub testowym.
6	*Długość wartości symbolicznej przekracza dopuszczalną długość!* Błąd zwracany w sytuacji, gdy długość jednej z wartości symbolicznych pliku uczącego lub testowego jest dłuższa niż 16 znaków.	*[wiersz, kolumna]* Miejsce wystąpienia błędu w pliku uczącym lub testowym.
7	*Błąd zapisu wartości!* Błąd zwracany w sytuacji, gdy wartość kolumny nie została poprawnie zapisana do pliku binarnego.	
8	*Błąd otwarcia pliku!* Błąd pojawiający się w sytuacji, gdy nie powiedzie się próba otwarcia jednego z plików. Powodem może być brak dostępu lub zezwolenia na jedną z operacji plikowych.	
9	*Zarezerwowany.* Błąd zarezerwowany dla następnych wersji programu Neuronix.	*[wiersz, kolumna]* Miejsce wystąpienia błędu w pliku uczącym lub testowym.
10	*Nazwa kolumny wejściowej lub wyjściowej nie jest unikalna!* Błąd pojawiający się w sytuacji, gdy nazwa kolumny występuje więcej niż jednokrotnie. Program Neuronix wymaga, aby nazwa każdej z kolumn pliku uczącego i testowego była niepowtarzalna, co umożliwia mu lokalizowanie jej wartości.	*[wiersz, kolumna]* Miejsce wystąpienia błędu w pliku uczącym lub testowym.
11	*Jedna z kolumn pliku uczącego nie została odnaleziona!* Błąd pojawiający się w sytuacji, gdy program Neuronix nie potrafi zlokalizować kolumny, której nazwa występowała w pliku uczącym. Występuje: - gdy plik testowy nie zwiera kolumny, która znajduje się w pliku uczącym, - gdy w trakcie uruchamiania sieci wykorzystywana jest wartość symboliczna, która nie była wprowadzona w pliku uczącym - Neuronix poszukuje wtedy kolumny, która jest złożeniem nazwy kolumny i nieznanego symbolu.	*[nazwa_kolumny]* *Nazwa kolumny, w której pojawił się błąd*

12	*Plik testowy nie został wskazany!* Błąd zwracany, gdy użytkownik nie wskazał pliku testowego w oknie parametrów sieci neuronowej. Najczęściej pojawia się w sytuacji, gdy „Częstość testowania” jest różna od zera lecz parametr „Plik testowy” jest pusty.	
13	*Plik uczący nie został wskazany!* Błąd zwracany gdy użytkownik próbuje uruchomić uczenie bez wskazania pliku uczącego w parametrze „Plik uczący”	
14	*Plik testowy nie istnieje!* Błąd zwracany w sytuacji, gdy wskazany plik testowy nie istnieje lub podana ścieżka dostępu jest nieprawidłowa.	
15	*Plik uczący nie istnieje!* Błąd zwracany w sytuacji, gdy wskazany plik uczący nie istnieje lub podana ścieżka dostępu jest nieprawidłowa.	
16	*Ilość kolumn wejściowych i wyjściowych w pliku uczącym i testowym nie jest taka sama!* Błąd zwracany w sytuacji, gdy liczba kolumn wejściowych i wyjściowych jest inna w pliku uczącym i pliku testowym. Program Neuronix wymaga aby plik uczący i testowy zawierały identyczne kolumny wejściowe i wyjściowe - ta sama ilość, te same nazwy i ten sam typ wartości w ramach kolumny. Powyższy warunek nie dotyczy kolumn opisowych, które nie są ani wejściowe ani wyjściowe.	
17	*Musi istnieć przynajmniej jedno wejście!* Błąd zwracany w sytuacji, gdy plik uczący lub testowy nie zawiera żadnej kolumny wejściowej.	
18	*Musi istnieć przynajmniej jedno wyjście!* Błąd zwracany w sytuacji, gdy plik uczący lub testowy nie zawiera żadnej kolumny wyjściowej.	
19	*Niedopuszczalna wartość wejścia lub wyjścia!* Błąd zwracany w sytuacji, gdy w kolumnie numerycznej znajdują się wartości symboliczne. Kolumna, która zawiera wartości symboliczne powinna mieć nazwę rozpoczynającą się od jednego z dwóch znaków specjalnych: # lub <. Jeżeli nazwa nie rozpoczyna się znakiem specjalnym to symulator Neuronix zakłada, że kolumna zawiera tylko wartości numeryczne.	*[wiersz, kolumna]* Miejsce wystąpienia błędu w pliku uczącym lub testowym.

20	*Opis kolumny nie może być dodany!* Błąd wewnętrzny generowany w sytuacji, gdy nie może być utworzona struktura opisująca jedną z kolumn pliku uczącego lub testowego.	
21	*Zbyt mało pamięci dla opisu kolumn!* Błąd wewnętrzny generowany w sytuacji, gdy nie może być utworzona struktura opisująca jedną z kolumn pliku uczącego lub testowego. Powodem jest zbyt mała ilość pamięci operacyjnej.	
22	*Zbyt mało pamięci dla słownika symboli!* Błąd wewnętrzny generowany w sytuacji, gdy nie może być utworzona struktura opisująca pojedynczy symbol. Powodem jest zbyt mała ilość pamięci operacyjnej.	
23	*Błąd podczas zapisu słownika!* Błąd wewnętrzny generowany w sytuacji, gdy jeden z symboli nie może być zapisany.	
24	*Zapis słownika symboli został przerwany!* Komunikat generowany po przerywaniu procesu zapisu słownika symboli poprzez naciśnięcie klawisza „Anuluj”.	
25	*Generacja słownika symboli została przerwana!* Komunikat generowany po przerwaniu procesu generacji słownika symboli poprzez naciśnięcie klawisz „Anuluj”.	*[wiersz, kolumna]* Miejsce wystąpienia błędu w pliku uczącym lub testowym.
26	*Zapis pliku binarnego został przerwany!* Komunikat generowany po przerwaniu procesu zapisu pliku binarnego poprzez naciśnięcie klawisz „Anuluj”.	*[wiersz, kolumna]* Miejsce wystąpienia błędu w pliku uczącym lub testowym.
27	*Zbyt mało pamięci podczas generacji pliku skali!* Błąd wewnętrzny generowany w sytuacji, gdy ilość pamięci potrzebna do generacji pliku skali jest zbyt mała.	
28	*Błąd w czasie odczytu słownika symboli!*	
29	*Opis słownika nie może być dodany!* Błąd wewnętrzny generowany w sytuacji, gdy nie może być utworzona struktura opisująca jedną z wartości symbolicznych.	
30	*Błąd w czasie odczytu pliku binarnego!*	

31	*Kolumna o podanej nazwie nie została odnaleziona - nieprawidłowa wartość symboliczna!* Błąd występujący w sytuacji, gdy w trakcie odczytu pliku binarnego nie została odnaleziona kolumna o szukanej nazwie. Błąd może pojawić się jeżeli w jednej z kolumn podany został symbol, który nie występował w pliku uczącym.	*[nazwa kolumny]* Nazwa kolumny, w której pojawił się błąd.
32	*Wartość binarna nie ma określonego typu dla wejścia lub wyjścia!* Błąd wewnętrzny występujący w sytuacji gdy struktura opisująca wartość binarną nie zawiera żadnej z dopuszczalnych wartości.	
33	*Błąd podczas zapisu pliku skali!*	
34	*Błąd podczas odczytu pliku skali!*	
35	*Opis skali nie może być dodany!* Błąd wewnętrzny występujący w sytuacji, gdy jedna ze struktur opisująca przeskalowaną wartość numeryczną nie może zostać utworzona.	
36	*Zapis pliku typu NNN został przerwany!* Komunikat jaki pojawia się po przerwaniu zapisu pliku skali poprzez naciśnięcie klawisza „Przerwij”	*[wiersz, kolumna]* Miejsce wystąpienia błędu w pliku uczącym lub testowym.
37	*Zapis pliku typu NNx został przerwany!* Komunikat generowany po przerwaniu przez użytkownika procesu zapisu plików typ NNx. Pliki typu NNx są to przeskalowane, binarne pliki uczące i testowe. Należą do nich pliki o rozszerzeniu ._nl oraz _nt.	*[wiersz, kolumna]* Miejsce wystąpienia błędu w pliku uczącym lub testowym.
38	*Błąd zapisu pliku typu NNx!*	
39	*Błąd wewnętrzny!* Błąd wewnętrzny, który nie powinien wystąpić w trakcie działania symulatora Neuronix. W razie pojawienia się tego komunikatu należy skontaktować się z producentem oprogramowania.	
40	*Plik wag nie został wskazany!* Błąd pojawiający się w sytuacji, gdy użytkownik próbuje uruchomić uczenie bez wskazania pliku wag sieci. Występuje również w sytuacji, gdy użytkownik inicjuje sieć bez wskazania pliku wag.	
41	*Zbyt mało pamięci dla struktur sieci!* Błąd pojawiający się w sytuacji, gdy ilość wolnej pamięci operacyjnej jest zbyt mała aby utworzyć struktury sieci neuronowej.	

42	*Zbyt mało pamięci dla bufora plikowego!* Błąd pojawiający się w sytuacji, gdy ilość wolnej pamięci operacyjnej jest zbyt mała aby utworzyć pomocnicze bufory dla plików uczącego i testowego.	
43	*Błąd podczas ustalania pozycji w pliku typu NNl!* Błąd wewnętrzny podczas ustalania pozycji kursora w binarnym, przeskalowanym pliku uczącym.	
44	*Błąd podczas ustalania pozycji w pliku typu NNt!* Błąd wewnętrzny podczas ustalania pozycji kursora w binarnym, przeskalowanym pliku testowym.	
45	*Zbyt mało pamięci podczas operacji mieszania wzorców!* Błąd pojawiający się w sytuacji, gdy ilość wolnej pamięci operacyjnej jest zbyt mała dla przeprowadzenia mieszania wzorców.	
46	*Błąd w trakcie odczytu bufora plikowego!*	
47	*Struktury sieci nie zostały zainicjowane!* Błąd wewnętrzny pojawiający się w sytuacji, gdy następuje próba rozpoczęcia uczenia bez wcześniejszego zainicjowania struktur sieci neuronowej.	
48	*Struktury sieci nie istnieją!* Błąd pojawiający się w sytuacji, gdy użytkownik próbuje uruchomić sieć neuronową bez jej wcześniejszej inicjalizacji. Inicjalizację dokonuje się z poziomu głównego menu lub okna uczenia.	
49	*Zbyt mało pamięci podczas uruchamiania sieci!* Błąd sygnalizowany w sytuacji, gdy ilość pamięci operacyjnej jest zbyt mała aby przeprowadzić uruchomienie sieci neuronowej.	
50	*Zbyt mała ilość kolumn wejściowych w procesie uruchamiania sieci!* Błąd sygnalizowany w sytuacji, gdy w ilość kolumn wejściowych w arkuszu jest inna niż ilość kolumn wejściowych w pliku uczącym.	
51	*Błędny typ wartości w kolumnie wejściowej!* Błąd występujący w przypadku gdy w kolumnie wejściowej znajduje się niedozwolony typ wartości np. tekst w kolumnie numerycznej.	*[wiersz, kolumna]* Miejsce wystąpienia błędu w pliku uczącym lub testowym.

52	*Plik skali nie istnieje!* Błąd sygnalizowany w sytuacji gdy symulator Neuronix nie potrafi zlokalizować pliku skali, który potrzebny jest do uruchomienia sieci z poziomu arkusza lub zewnętrznego programu.	
53	*Plik słownika symboli nie istnieje!* Błąd sygnalizowany w sytuacji gdy symulator Neuronix nie potrafi zlokalizować pliku słownika symboli, który potrzebny jest do uruchomienia sieci z poziomu arkusza lub zewnętrznego programu. Plik słownika musi być dostępny w przypadku, gdy plik uczący zawierał kolumny z wartościami symbolicznymi.	
54	*Kolumna o podanej nazwie nie istnieje!* Błąd pojawiający się w sytuacji gdy symulator Neuronix nie może odnaleźć kolumny o podanej nazwie. Komunikat generowany jest w trakcie uruchamiania sieci jeżeli nazwa kolumny lub jedna z wartości symbolicznych nie występowała w pliku uczącym.	*[wiersz, kolumna]* Miejsce wystąpienia błędu w pliku uczącym lub testowym.
55	*Błąd w trakcie odczytu pliku wag!*	
56	*Przepełnienie operacji matematycznej exp!* Błąd przepełnienia operacji matematycznej. Może pojawić się w sytuacji, gdy wartość wejściowa sieci w trakcie jej uruchamiania znacznie odbiega od skrajnych wartości z pliku uczącego.	
57	*Podany plik projektu nie istnieje!* Błąd zwracany w przypadku gdy wskazany plik projektu nie może być zlokalizowany. Zwracany podczas uruchamiania sieci z poziomu zewnętrznego programu napisanego w języku reprezentacji wiedzy Sphinx.	
58	*Błąd w trakcie odczytu pliku projektu!* Zwracany podczas uruchamiania sieci z poziomu zewnętrznego programu napisanego w języku reprezentacji wiedzy Sphinx.	
59	*Plik parametrów nie istnieje!* Błąd zwracany w sytuacji, gdy następuje próba inicjowania sieci neuronowej na podstawie pliku projektu, do którego dostęp nie jest możliwy.	
60	*Nieprawidłowy rozmiar warstwy!* Błąd zwracany w sytuacji, gdy w pliku parametrów sieci neuronowej (rozszerzenie .cpv) występuje błędny rozmiar warstwy - np. rozmiar wejścia lub wyjścia nie jest wartością numeryczną z przedziału od 0 do 10000)	

61	*Nieprawidłowy typ funkcji przejścia!* Błąd zwracany w sytuacji, gdy w pliku parametrów sieci neuronowej (rozszerzenie .cpv) występuje błędny typ funkcji przejścia - w obecnej wersji dostępna jest tylko logistyczna funkcja przejścia.	
62	*Plik wag nie istnieje!* Błąd zwracany w sytuacji, gdy w pliku parametrów sieci neuronowej (rozszerzenie .cpv) występuje błędna ścieżka dostępu do pliku wag.	
63	*Numer sieci przekroczył maksymalną wartość!* Błąd zwracany w sytuacji, gdy numer sieci przekroczył wartość INT_MAX równą 32767.	
64	*Nieprawidłowy numer sieci!* Błąd zwracany w sytuacji, gdy numer sieci jest niepoprawny np. równy jest liczbie ujemnej lub jest większy od największego numeru istniejącej sieci.	
65	*Sieć o podanym numerze nie istnieje!* Błąd zwracany w sytuacji, gdy użytkownik próbuje odwołać się do sieci, która nie została utworzona lub została już usunięta z pamięci operacyjnej.	
66	*Nieprawidłowy rozmiar wzorca wejściowego lub wyjściowego* Błąd zwracany w sytuacji, gdy rozmiar wzorca wejściowego lub wyjściowego sieci nie jest zgodny z rozmiarem warstwy wejściowej lub wyjściowej.	*[wiersz, kolumna]* Miejsce wystąpienia błędu w pliku uczącym lub testowym.
67	*Nieprawidłowa wartość parametru 'bias'* Błąd zwracany w sytuacji, gdy wartość parametru ‘bias’ w pliku parametrów sieci neuronowej nie jest prawidłowa. Dopuszczalne wartości to ‘tak’ i ‘nie’.	
68	*Nieprawidłowy warunek zapisu najlepszej sieci - brak testowania!* Błąd zwracany w sytuacji, gdy użytkownik błędnie określił warunek zapisywania najlepszych ustawień sieci. Jednym z tych warunków jest osiągnięcie przez sieć neuronową najmniejszej dotychczasowej wartości błędu testowania. W przypadku, gdy testowanie sieci nie jest włączone traci on tym samym sens.	
69	*Błąd w trakcie otwarcia pliku wag!* Błąd jaki pojawia się w przypadku trudności z odczytem pliku wag. Powodem może być brak dostępu do pliku.	

70	*Opis arkusza prezentacji nie może być dodany!* Błąd wewnętrzny, który pojawia się w sytuacji, gdy nie może zostać utworzona struktura opisująca arkusz prezentacji.	
71	*Zbyt mało pamięci przy dodawaniu opisu arkusza prezentacji!* Błąd wewnętrzny, jaki zachodzi w przypadku istnienia zbyt małej ilości pamięci operacyjnej podczas tworzenia struktur opisującej arkusz prezentacji.	
72	*Błąd w trakcie usuwania arkusza prezentacji - nie odnaleziony!* Błąd wewnętrzny, jaki występuje w przypadku próby usunięcia arkusza prezentacji, którego numer identyfikacyjny nie został odnaleziony.	
73	*Plik typu NNN nie jest dostępny!* Błąd jaki pojawia się w sytuacji, gdy program Neuronix nie potrafi zlokalizować pliku skali (rozszerzenie ._nn).	
74	*Binarny plik uczący nie istnieje!* Błąd jaki pojawia się w sytuacji, gdy program Neuronix nie potrafi zlokalizować binarnego pliku uczącego (rozszerzenie ._bl).	
75	*Binarny plik uczący nie jest dostępny!* Błąd jaki pojawia się w sytuacji, gdy program Neuronix nie ma dostępu do binarnego pliku uczącego (rozszerzenie ._nl).	
76	*Tekstowy plik uczący nie jest dostępny!* Błąd jaki pojawia się w sytuacji, gdy program Neuronix nie ma dostępu do tekstowego pliku uczącego (rozszerzenie .lrn).	
77	*Błąd w trakcie odczytu zmiennej z binarnego pliku uczącego!* Błąd wewnętrzny jaki pojawia się w sytuacji, gdy program Neuronix nie może odczytać jednej ze zmiennych zawartych w binarnym pliku uczącym (rozszerzenie ._bl).	
78	*Błąd w trakcie odczytu pliku binarnego!*	
79	*Plik testowy nie jest dostępny!*	
80	*Binarny plik testowy nie istnieje!* Błąd jaki pojawia się w sytuacji, gdy program Neuronix nie potrafi zlokalizować binarnego pliku testowego (rozszerzenie ._bt).	

81	*Binarny plik testowy nie jest dostępny!* Błąd jaki pojawia się w sytuacji, gdy program Neuronix nie ma dostępu do binarnego pliku uczącego (rozszerzenie ._nl).	
82	*Błąd w trakcie odczytu binarnego pliku testowego!*	
83	*Błąd w trakcie odczytu zmiennej z binarnego pliku testowego!* Błąd wewnętrzny jaki pojawia się w sytuacji, gdy program Neuronix nie może odczytać jednej ze zmiennych zawartych w binarnym pliku testowym (rozszerzenie._bt).	
84	*Numer wiersza poniżej dopuszczalnej wartości!* Błąd, jaki ma miejsce w sytuacji,gdy następuje próba odczytu jednej z wartości binarnych, której położenie określone jest nieprawidłową wartością wiersza np. wartością ujemną.	
85	*Plik programu nie został wskazany!* Błąd pojawiający się w sytuacji, gdy użytkownik próbuje uruchomić translację programu (bazy wiedzy) bez podania do niej ścieżki dostępu.	
86	*Zawartość pliku skali nie dotyczy wskazanej sieci neuronowej!* Błąd, jaki pojawia się w przypadku, gdy rozmiar i rodzaj wejść uruchamianej sieci nie jest zgodny z plikiem skali (rozszerzenie._nn).	
87	*Plik skali zawiera niedozwolone kolumny symboliczne!* Błąd generowany w sytuacji, gdy użytkownik uruchamia sieć funkcją RUNNETWORK2, przy czym plik skali dla uruchamianej sieci zawiera wartości symboliczne.	

15

Automatyzacja OLE

W rozdziale:

- Wykorzystanie Neuronixa jako serwera automatyzacji OLE

AUTOMATYZACJA OLE

Począwszy od wersji 4.0 program Neuronix został rozbudowany o możliwości pracy aplikacji jako serwera automatyzacji OLE (Object Linking And Embedding). Oznacza to, że sieci neuronowe zbudowane i wyuczone mogą być wywoływane z dowolnego środowiska programowego obsługującego automatyzację OLE np. Delphi, VC++, Visual Basic, Visual Basic for Application (język makr pakietu Microsoft Office) i inne. Możliwe jest również uruchamianie sieci neuronowej w sieci Internet.

FUNKCJE SERWERA

System Neuronix udostępniany jest jako klasa ”Sphinx.Neuronix” i udostępnia cztery metody.

```
Funkcja InitNetwork ( BSTR NNPath ) int
```

Funkcja powoduje otwarcie i zainicjowanie sieci neuronowej wskazywanej w parametrze wywołania. Po prawidłowym zainicjowaniu sieci neuronowej zwracany jest jej unikalny identyfikator używany w kolejnych procedurach.

```
Procedura DeleteNetwork ( long nnID )
```

Procedura usuwa siec neuronową z pamięci. Powinna być wywoływana po zakończeniu pracy z siecią.

```
Procedura RunNetwork ( long nnID, Variant *Input,
                       Variant *Output, Variant *Values )
```

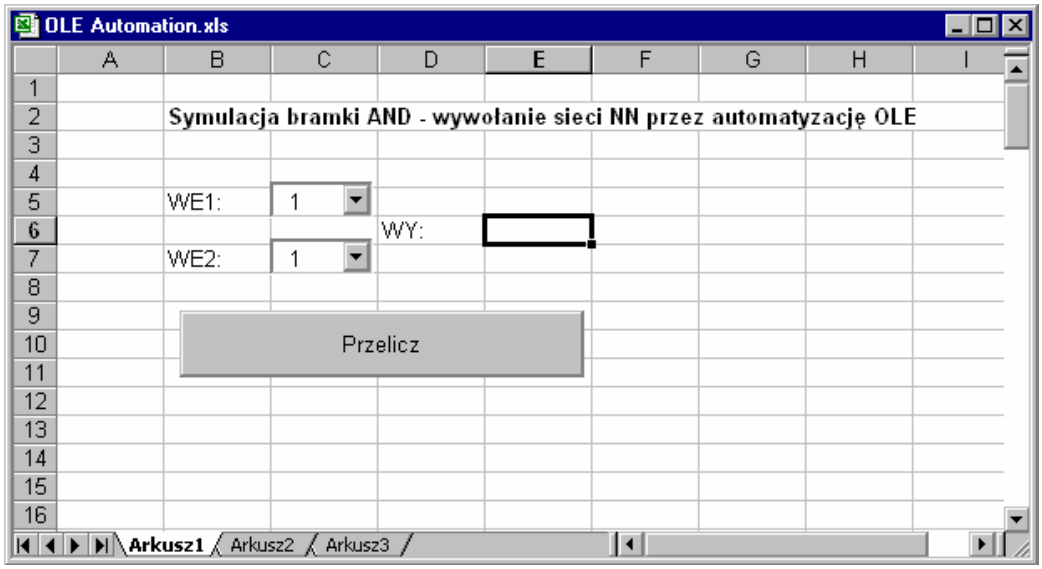
Procedura uruchamiająca sieć neuronową identyfikowaną przez identyfikator *nnID*. Parametr *Input* wskazuje na tablicę typu Variant zawierającą **pełną** listę wejść sieci neuronowej w postaci „we=wartość”. Parametr *Output* zawiera tablicę elementów wyjściowych (symbolicznych nazw wyjść), natomiast ostatni parametr jest zwrotną tablicą do której zwracane są wartości wygenerowane przez sieć neuronową.

```
Procedura GetNrxError ( int iErrorId, Variant Text )
```

Procedura służy pobraniu treści komunikatu błędu zwróconego przez *InitNetwork* lub *RunNetwork*.

PRZYKŁAD AUTOMATYZACJI OLE

Poniżej przedstawiamy przykładowe makro wywołujące sieć neuronową z poziomu arkusza kalkulacyjnego Microsoft Excel. Na rysunku poniżej przedstawiono arkusz kalkulacyjny do uruchamiania sieci neuronowej projektu Funkcja AND.



Makro podpięte pod przycisk *Przelicz* zdefiniowane jest w następujący sposób:

```
Private Sub CommandButton1_Click()
  '
  Dim NN As Object

  Set NN = CreateObject("Sphinx.Neuronix")
  Dim WE(2) As Variant
  Dim WY_Name(1) As Variant
  Dim WY As Variant

  'pobranie wartości (neuronów) wejściowych
  WE(0) = "x1=" + Format(Range("C5").Text)
  WE(1) = "x2=" + Format(Range("C7").Text)

  ' wyjście
  WY_Name(0) = "y"

  'Identyfikator sieci neuronowej
  Dim nnID As Integer

  nnID = NN.InitNetwork("C:\Sphinx 4.0\Projekt\Funkcja AND\AND.npr")

  i = NN.RunNetwork(nnID, WE, WY_Name, WY)

  'zapisz wyjście do komórki E6
  Range("E6").Select
  ActiveCell.FormulaR1C1 = WY(0)

  NN.DeleteNetwork (nnID)

End Sub
```