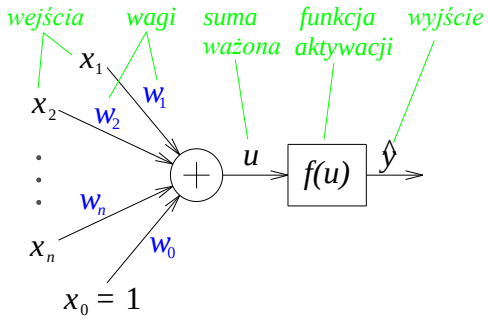
Prognozowanie szeregów czasowych za pomocą sieci neuronowych

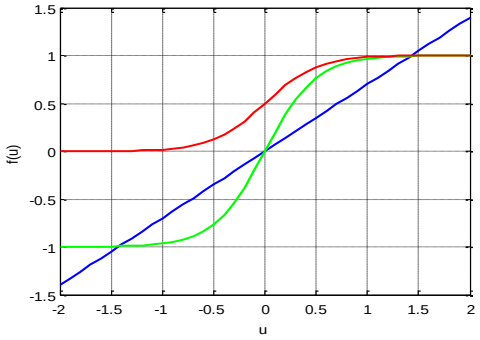
Model neuronu



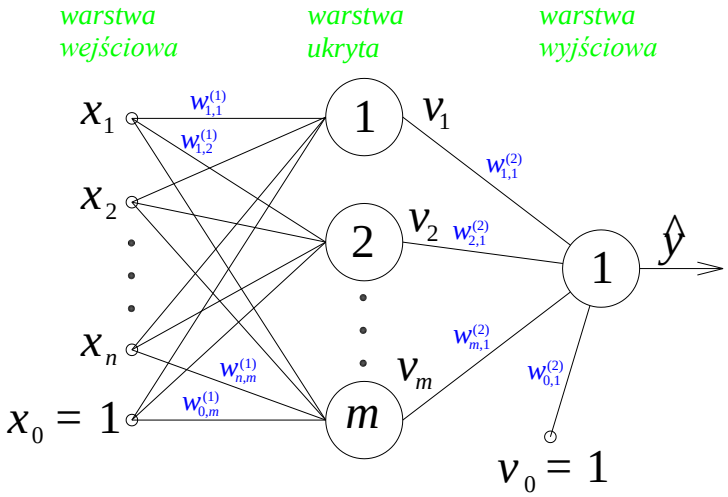
Neuron realizuje funkcję:

$$\hat{y} = f\left(\sum_{j=0}^{n} x_j w_j\right)$$

Funkcje aktywacji neuronu f(.)

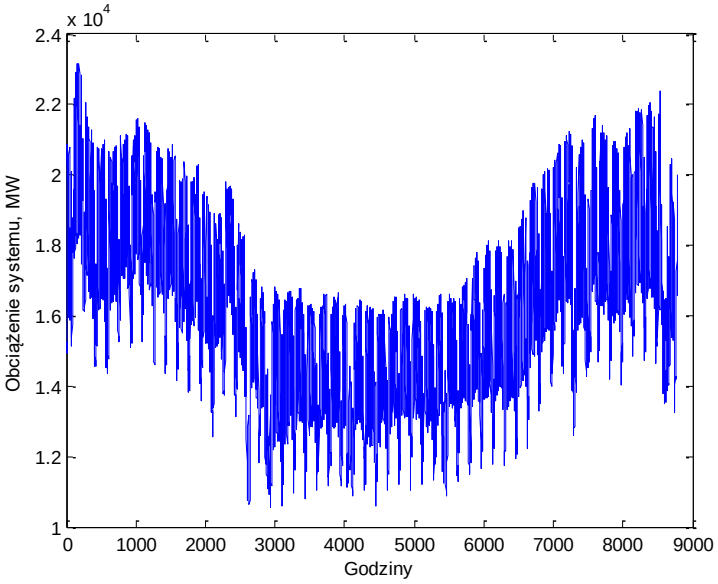


Model sieci neuronowej

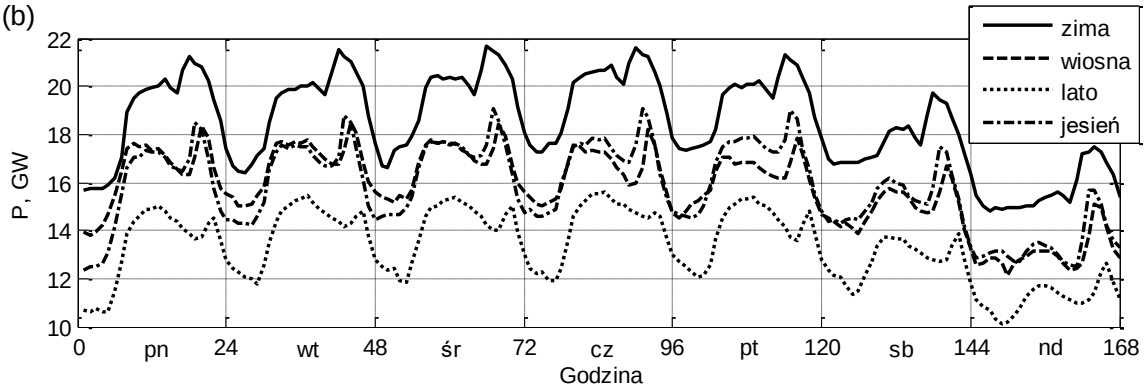


Zadanie

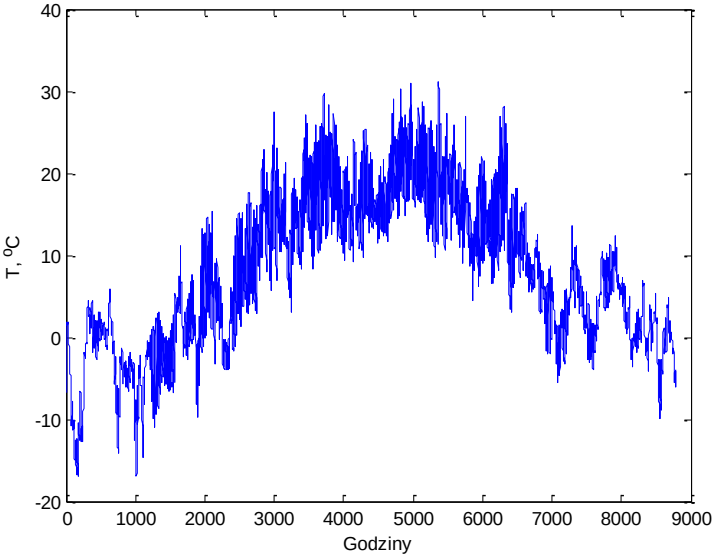
1. Dane jest obciążenie systemu w kolejnych godzinach roku:



(b)



2. Dane są godzinowe temperatury atmosferyczne z tego samego okresu co obciążenie:



3. Zaprognozować obciążenia w kolejnych godzinach doby na podstawie:
    a. obciążeń w poprzednich godzinach
       P(t) = f(P(t-1), P(t-2), …, P(t-d)), d – długość ciągu wejściowego
    b. temperatur w poprzednich godzinach
       P(t) = f(T(t-1), T(t-2), …, T(t-d))
    c. obciążeń i temperatur w poprzednich godzinach
       P(t) = f(P(t-1), P(t-2), …, P(t-d), T(t-1), T(t-2), T(t-d))

4. Z wykorzystaniem skryptu zamieszczonego poniżej przeprowadź eksperymenty dla różnych ustawień:
    - war = 1, 2 lub 3,
    - opóźnienie d = var w zakresie od 1 do 24,
    - liczba neuronów l_neur = var w zakresie od 1 do 10.

    Odnotuj wyniki i wskaż najlepsze ustawienia.
5. Zapoznaj się z narzędziem ntstool. Wykonaj za jego pomocą kilka eksperymentów prognostycznych j.w. Odnotuj i skomentuj wyniki i wykresy.

```
load dane1;

d=1; %opóźnienie (od 1 do 24)
l_neur=1; %liczba neuronów w warstwie ukrytej
war=3;  %wariant: 1 - P(t) = f(P(t-1), P(t-2), …, P(t-d))
%2 - P(t) = f(T(t-1), T(t-2), …, T(t-d))
%3 - P(t) = f(P(t-1), P(t-2), …,P(t-d), T(t-1), T(t-2), T(t-d))

for j=1:1
    ll=length(P);
    TT=[]; PP=[];
    Py = P(d+1:end);
    [P,vp]=mapminmax(P);
    [T,vt]=mapminmax(T);
    for i=1:ll-d
        TT = [TT T(i:i+d-1)'];
        PP = [PP P(i:i+d-1)'];
    end
    y = P(d+1:end);

    if war == 1
        x=PP;
    elseif war == 2
        x=TT;
    elseif war == 3
        x=[PP;TT];
    end
end


%trening sieci
net=newff(minmax(x),[l_neur 1],{'tansig','purelin'},'trainscg');
%utworzenie sieci; pierwszy argument reprezentuje zakresy danych
wejściowych,
```

```
%drugi - liczbę neuronów w warstwie ukrytej i wyjściowej, trzeci - typy
funkcji aktywacji w tych warstwach, czwarty - metodę uczenia sieci
%funkcja zwraca obiekt sieci net
net.trainParam.epochs = 100; %liczba epok uczenia
[net,tr]=train(net,x,y); %trening sieci

%symulacja nauczonej sieci dla zbioru uczącego
yy=sim(net,x);
Pp = mapminmax('reverse',yy,vp);
b_trn=abs((Pp-Py)./Py)*100; %błędy aproksymacji w procentach
disp(sprintf('Błąd średni prognozy MAPE:  %f%%',mean(b_trn))); %błąd średni
figure(1);
plot(Py); hold on; plot(Pp,'r'); hold off;
xlabel('Godziny'); ylabel('Moc, MW');

figure(2);
hist(b_trn,40);
xlabel('Błąd MAPE'); ylabel('Liczba przypadków');
```