

*Andrzej Złobecki**, *Ryszard Macura***, *Magdalena Michalczyk***

**Katedra Inżynierii Mechanicznej i Agrofizyki*

***Katedra Chłodnictwa i Koncentratów Spożywczych
Akademia Rolnicza w Krakowie*

MODELOWANIE PROCESU OMŁOTU PRZY WYKORZYSTANIU SZTUCZNYCH SIECI NEURONOWYCH

Streszczenie

Celem pracy było wykorzystanie sieci neuronowych jako modelu symulacyjno-decyzyjnego procesu omłotu w kombajnie zbożowym. Proces modelowania polegał na znalezieniu najlepszej sieci neuronowej o najmniejszym błędzie przetwarzania danych kolejno dla niedomłotu i uszkodzeń ziarna. Po przeprowadzeniu procesu uczenia i testowania określono najlepsza sieć tzn. taką, która generuje wielkość strat w postaci niedomłotu i uszkodzeń ziarna z najmniejszym błędem przetwarzania. Kryterium wyboru sieci był najmniejszy błąd dla zbioru walidacyjnego. Zmiennymi prezentowanymi na wejściu sieci były parametry regulacyjne zespołu młocącego (tj. obroty bębna młocącego, wielkość zasilania regulowana prędkością jazdy kombajnu, rozmiary szczeliny roboczej) oraz wilgotność młóconego zboża. Na wyjściu obu sieci uzyskiwano wielkości strat lub niedomłotu w zależności od wartości zmiennych wejściowych. Proces modelowania wykonano w programie „Statistica Neural Network” firmy Statsoft. Uzyskane modele zostały przetestowane, uruchomione i zapisane. Modele te można wykorzystać w procesie automatyzacji zbioru kombajnowego. Powstałe sieci neuronowe mogą być zastosowane w komputerze pokładowym kombajnu, który ustawiałby parametry regulacyjne w zależności od przeznaczenia ziarna tak aby niedomłot lub straty były najmniejsze.

Słowa kluczowe: kombajn zbożowy, proces omłotu, straty, niedomłot, uszkodzenia, sztuczne sieci neuronowe

Wstęp

W czasie zbioru zbóż kombajnami do obniżenia plonów przyczyniają się zarówno straty jakościowe jak i straty ilościowe. Straty jakościowe mają szczególne znaczenie w przypadku zbioru materiału siewnego, gdyż uszkodzenia mogą znacznie

obniżyć jakość ziarniaków. Straty jakościowe mogą powodować wzrost strat ilościowych, gdyż wzrasta proces oddychania w uszkodzonych ziarnach, co poprzez samozagrzewanie się ziarna, pogarsza jego wartość użytkową lub całkowite zepsucie. Wielkość tych strat nabiera szczególnego znaczenia, jeśli rozpatrzy się je w skali kraju [Gieroba, Dreszer 1993].

Mimo tego iż wiele czynników wpływa na wielkość powstawania strat, nadal nie do końca wiadomo w jaki sposób należy podejść do problemu zmniejszenia strat podczas zbioru kombajnowego [Złobecki 1995].

W warunkach polowych jedynym sposobem zmniejszenia strat jest właściwy dobór odmian zbóż o odpowiednich właściwościach, dobór terminu zbioru oraz parametrów regulacyjnych kombajnu. Problem strat można rozwiązać opracowując model optymalizacji procesu zbioru, co może mieć wpływ na dalszy rozwój konstrukcji kombajnów zbożowych, i może umożliwić opracowanie systemu automatyzacji procesu zbioru. Taki system sterujący można by umieścić w komputerze pokładowym kombajnu, komputer pobierałby dane z czujników i automatycznie ustawiał parametry regulacyjne aby zoptymalizować proces zbioru [Mikucki 1993].

Cel i zakres pracy

Trudności związane z matematycznym opisem procesu powstawania strat ilościowych i jakościowych w czasie zbioru kombajnem zbożowym, stwarzają konieczność oparcia się o dane empiryczne przy strukturalnym tworzeniu modelu. Problem ten rozwiązuje możliwość zastosowania sieci neuronowych jako narzędzia do odzwierciedlenia zależności zachodzących w modelowanym procesie. Tak więc celem pracy jest wykorzystanie sieci neuronowych jako modelu symulacyjno-decyzyjnego procesu omlotu w kombajnie zbożowym.

Material i metody

Do opracowania modelu przebiegu procesu omlotu wykorzystane były wyniki badań kombajnu zbożowego Bizon Z056, które przeprowadzono w czasie kombajnowego zbioru ziarna siewnego. Badania były przeprowadzone na pszenicy ozimej Begra, przy wilgotności ziarna zmieniającej się w zakresie 16–24%.

Po zliczeniu ziaren pozostałych w wymłóconych kłosach określano procentowe wartości niedomłotu, uwzględniając wielkość plonu, średnią ilość źdźbeł na m², oraz średnią ilość ziaren w kłosie. Straty w postaci uszkodzeń ziarna były określane przez zdolność kiełkowania, która została oznaczona zgodnie z obowiązującymi normami. Parametrem opisującym wielkość strat spowodowanych uszkodzeniami była liczba ziarniaków nie skielkowanych w danej próbce.

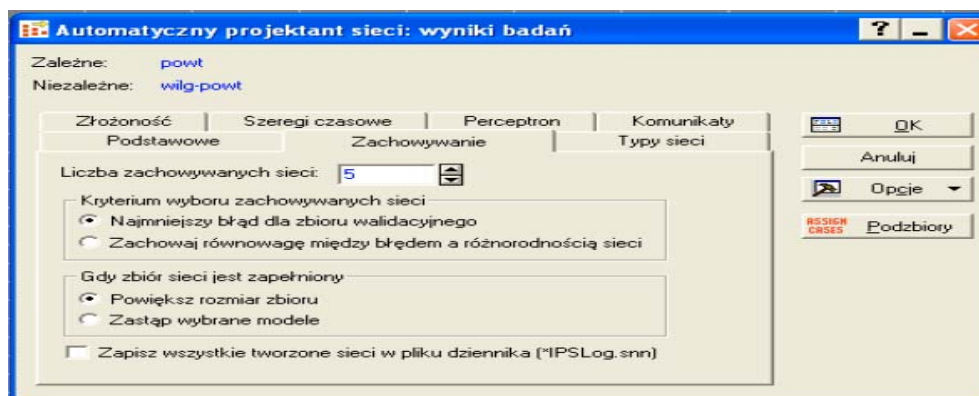
Model odzwierciedlający zależności zachodzące w zespole młócającym kombajnu zbożowego utworzono w oparciu o strukturę sieci neuronowej, modelowanie przebiega w programie „Statistica Neural Networks”.

Ze względu na dużą czasochłonność projektowania i testowania jakości sieci skorzystano z funkcji „Automatycznego Projektanta Sieci”. Jego praca polega na wyborze z kilkuset rodzajów sieci takiej, której jakość oraz błąd walidacyjny jest najmniejszy. Uruchamiając pracę programu na wejściu sieci zadano takie zmienne, jakie ustalono w doświadczeniu polowym:

- wielkość zasilania [kg/s] – regulowaną przez zmianę prędkości jazdy,
- kombajnu w zakresie 0,3–0,9 [m/s],
- liczbę obrotów bębna młócającego w zakresie 850–1050 [obr/min],
- wielkość szczeliny roboczej w zakresie [3–16] mm,
- wilgotność ziarna [16–24] %,

Uzyskano w ten sposób 300 poziomów wartości zmiennych traktowanych jako straty w postaci procentowych wartości ziarna nie wymłóconego i uszkodzonego w czterech powtórzeniach.

Po określeniu typu zmiennych otwierając kartę *zmienne wyjściowe/wejściowe* określono rodzaj zmiennych. Zmiennymi wyjściowymi były w pierwszym przypadku uszkodzenia a w drugim niedomłot. Na wejściu sieci wybrano wilgotność, prędkość, obroty, szczelinę i powtórzenie.



Rys. 1. Widok okna do wyboru opcji zachowywanych sieci

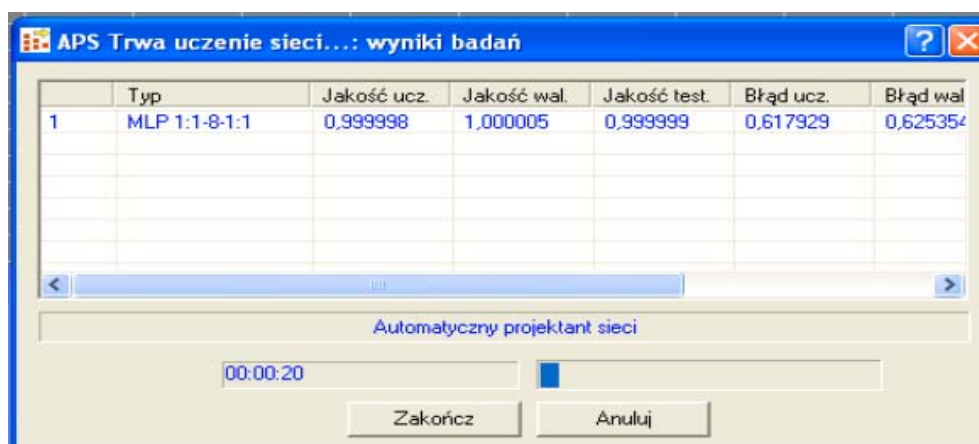
Fig. 1. Screen of saved networks selection option

Mając wczytane wyniki badań, zadeklarowany typ i rodzaj zmiennych uruchomiono Automatycznego Projektanta Sieci. W otworzonym oknie projektanta znajduje się wiele zakładek które umożliwiają dokładniejsze sprecyzowanie poszukiwań sieci.

Karta *zachowywanie* umożliwiła wyznaczenie, ile sieci program ma pokazać jako najlepsze. Do dalszych etapów jest potrzebna jedna najlepsza sieć, ale zaznaczając więcej mamy możliwość podglądu jak wiele różnią się one między sobą. W tym oknie musimy też zaznaczyć kryterium wyboru sieci, zaznaczona opcja „najmniejszy błąd dla zbioru walidacyjnego” jest najczęściej wykorzystywana do porównywania sieci. Umożliwia ona Automatycznemu Projektantowi Sieci utworzenie pętli w której jest uczona i testowana dana sieć aż do momentu uzyskania najmniejszego błędu uczenia dla zbioru walidacyjnego. Automatyczny Projektant wybiera losowo dane uczące i testujące sprawdzając czy błąd walidacyjny maleje, jeśli nie to zapisuje sieć i sprawdza inne sieci.

W zakładce *podstawowe* można było wybrać ilość testowanych sieci lub czas na testowanie sieci. Przeprowadzone eksperymentalne próby na tych samych danych (wyniki badań kombajnu) wykazały, że powyżej zadanych do testu 800 sieci program generuje jako najlepszą tą samą sieć.

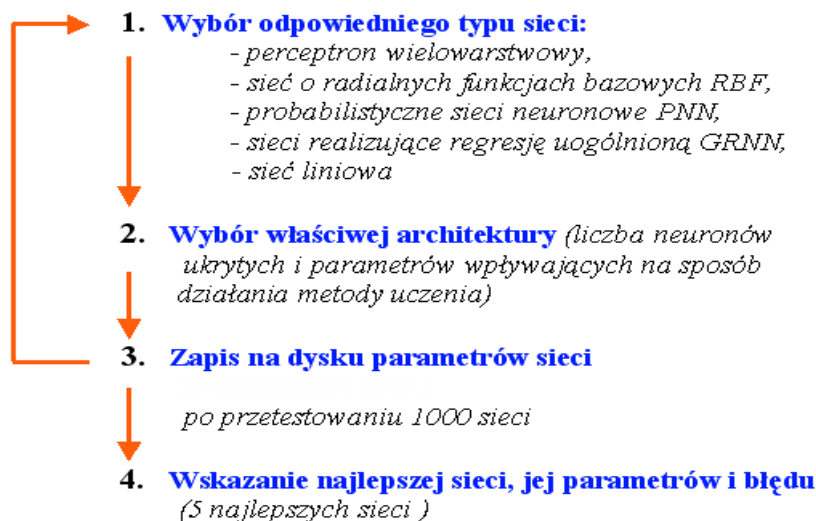
Po zatwierdzeniu zmian program rozpoczyna testowanie poszczególnych sieci (rys. 2).



Rys. 2. Okno procesu uczenia sieci
Fig. 2. Network learning process window

Tworzenie sieci można było w rezultacie podzielić na następujące etapy:

Etapy działania Automatycznego Projektanta Sieci



Po przeprowadzeniu uczenia sieci program wygenerował raport modelowania (rys. 3).

Raport podsumowania modelu (wyniki badań)							
Typ	Jakość ucz.	Jakość wal.	Jakość test.	Błąd ucz.	Błąd walid.	Błąd test.	
MLP 3:3-10-4-1:1	0,019695	0,022213	0,019198	0,004444	0,004972	0,004364	
MLP 3:3-2-1:1	0,019368	0,021805	0,019397	0,004370	0,004877	0,004412	
MLP 3:3-4-1:1	0,019445	0,021722	0,019525	0,004387	0,004860	0,004438	
MLP 4:4-10-2-1:1	0,017287	0,019488	0,016296	0,003901	0,004360	0,003706	
MLP 4:4-10-7-1:1	0,013476	0,016241	0,013916	0,003041	0,003633	0,003162	

Rys. 3. Raport o wygenerowanych sieciach (dla niedomlotu)

Fig. 3. Generated network report (for non-threshing)

W wyświetlonym raporcie jest podane wiele parametrów. Wielkość zbioru uczącego zazwyczaj osiąga poziom ok. 50% danych wejściowych. Jakość sieci dla podzbioru uczącego nie jest jednak wiarygodnym wskaźnikiem ogólnej jakości sieci. Dla zbioru uczącego dość łatwo uzyskać dowolnie mały błąd na zasadzie przeuczenia sieci. Również wartości dla podzbioru testowego do wybierania modelu, nie są obiektywne gdyż zaprzecza to istocie podzbioru testowego, który powinien pozostać do niezależnej oceny końcowej modelu [Osowski 1996].

Jakość sieci mierzy się różnie dla różnych typów sieci. Dla ciągłych zmiennych (sieci regresyjne) jako jakość sieci podaje się iloraz odchyłeń standardowych co w tym przypadku jest parametrem weryfikującym działanie sieci i jednocześnie decydującym o jej wyborze. Dla zmiennych nominalnych (sieci klasyfikujące) jako jakość podaje się proporcję poprawnie zaklasyfikowanych przypadków. Trudniejszym w interpretacji jest Błąd (dla zbioru uczącego/walidacyjnego/testowego). Ta wielkość jest trudniejsza w interpretacji niż jakość sieci. Jednak właśnie błąd podlega minimalizacji w trakcie uczenia sieci (dla podzbioru uczącego). Jest to błąd sieci wyznaczony na podstawie wartości błędów jednostkowych obliczonych za pomocą przyjętej funkcji błędu. W raporcie znajduje się też skrótowy zapis stosowanych algorytmów uczących. Podane są kody algorytmów i liczby epok, a także, opcjonalnie kod sposobu wybrania najlepszej sieci i tak dla wybranej sieci odczytany kod to: BP100,CG20,CG35b gdzie BP oznacza że używany był algorytm wstecznej propagacji błędów dla 100 epok, potem nastąpiło 20 epok gradientów sprzężonych a po nich 35 gradientów sprzężonych, kiedy to sieć zaczęła się przeuczać (b).

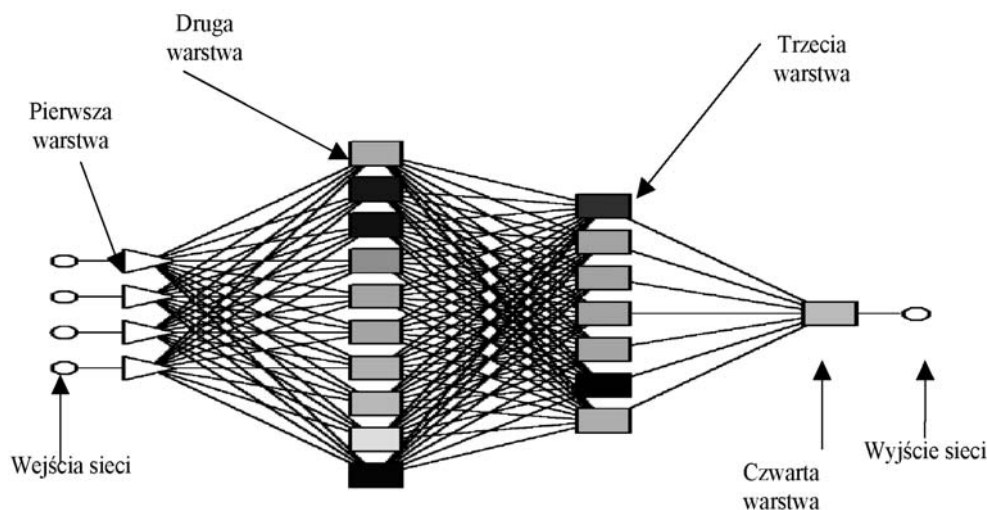
Najlepsza sieć została zapisana na dysku komputera, co jest etapem koniecznym do jej predykcji a więc uruchomienia. Została wybrana sieć o oznaczeniu: MLP 4:4-10-7-1:1

Skrót MLP oznacza perceptron wielowarstwowy, sieć ma 4 wejścia, liczba neuronów w pierwszej warstwie wynosi 4, w drugiej warstwie 10, w trzeciej warstwie 7, w czwartej jeden neuron i ostatnia liczba określa ilość wyjść sieci - jedno. Należy jednak zauważyć, że prezentowana sieć jest przewymiarowana (dwie warstwy ukryte) i rezultaty uzyskane przez Automatycznego Projektanta Sieci stanowią jedynie dane wyjściowe do projektowania modelu.

Program generuje widok sieci (rys. 4), kolorem zaznaczono aktualnie aktywowane neurony (w zależności od zadanych parametrów wejściowych będą się one zmieniać).

W drugim przypadku proces modelowania przebiegał identycznie, z tym że, zmienną wyjściową były uszkodzenia ziarna.

Raport błędów (rys. 5) pokazuje min. jakości i błędy oraz kod sposobu wybierania danej sieci, Dla wygenerowanej sieci jest to: BP100,CG20,CG55b gdzie BP oznacza że używany był algorytm wstecznej propagacji błędów dla 100 epok, potem nastąpiło 20 epok gradientów sprzężonych a po nich 55 gradientów sprzężonych, kiedy to sieć zaczęła się przeuczać (b).



Rys. 4. Widok wygenerowanej sieci dla niedomlotu
 Fig. 4. Screen of generated network for non-threshing

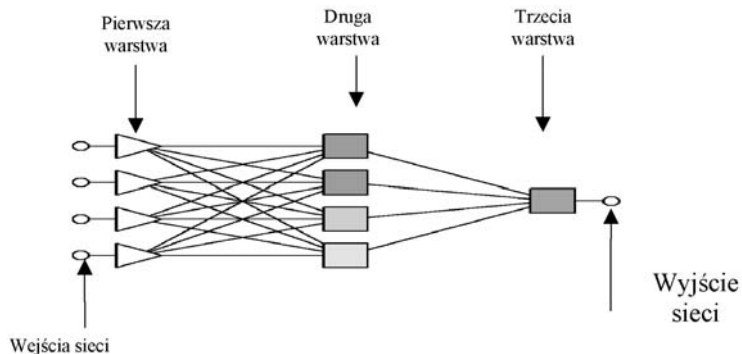
Typ	Jakość ucz.	Jakość wal.	Jakość test.	Błąd ucz.	Błąd walid.	Błąd test.
MLP 4:4-4-1:1	0,013678	0,014186	0,012969	0,003533	0,003649	0,003387

Rys. 5. Raport o wygenerowanych sieciach (dla uszkodzeń)
 Fig. 5. Generated network report (for damages)

Wygenerowana przez Automatycznego Projektanta sieć to: MLP 4:4-4-1:1

Skrót MLP oznacza perceptron wielowarstwowy, sieć ma 4 wejścia, liczba neuronów w pierwszej warstwie wynosi 4, w drugiej warstwie 4, w trzeciej warstwie 4 i ostatnia liczba określa ilość wyjść sieci, czyli jedno.

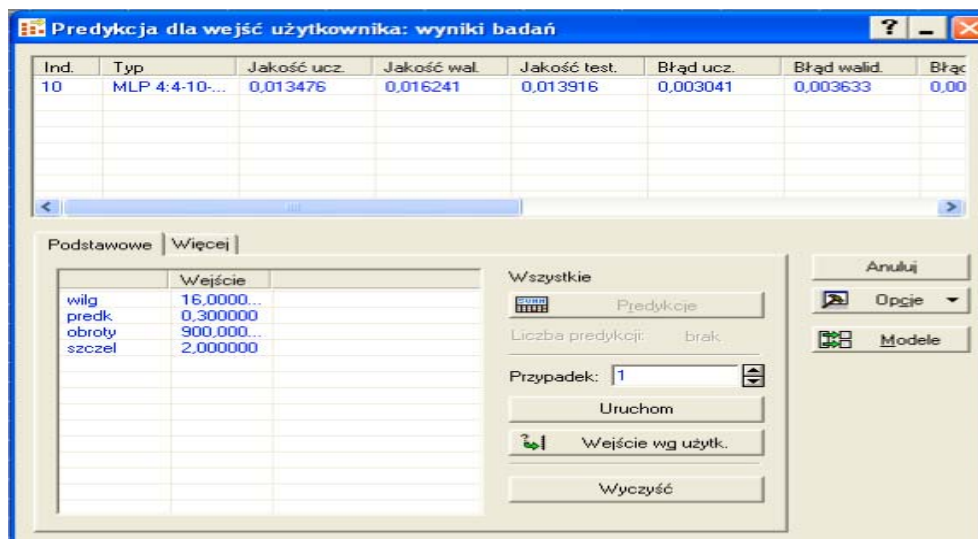
Program generuje widok sieci (rys. 6), kolorem zaznaczono aktualnie aktywowane neurony (w zależności od zadanych parametrów wejściowych będą się one zmieniać).



Rys. 6. Widok wygenerowanej sieci dla uszkodzeń.
 Fig. 6. Screen of generated network for damages

Wyniki

Po zapisaniu sieci można uruchomić sieć poleceniem *uruchom istniejący model*. Uruchamiając kartę *wejscie według użytkownika* pojawia się okno arkusza Excel umożliwiające wpisanie własnych wartości dla wilgotności, prędkości, obrotów i szczeliny. Dane wpisywane na wejście sieci muszą zawierać się w zakresie wyników badań.



Rys. 7. Widok okna predykcji
 Fig. 7. Prediction screen

Zaznaczając kartę *wszystkie* sieć w arkuszu Excel (rys. 8) wyświetla wartości dla kolejnych pomiarów w kolumnie niedomłot są wartości uzyskane z badań a w kolumnie niedomłot 10 znajdują się wartości wygenerowane przez sieć.

	Predykcja (10) (wyniki badań)					
	wilg	predk	obroty	szczel	nied	nied.10
1	16,00000	0,300000	900,000	2,000000	0,049841	0,050142
2	16,00000	0,300000	900,000	2,000000	0,049841	0,050142
3	16,00000	0,300000	900,000	2,000000	0,049841	0,050142
4	16,00000	0,300000	900,000	2,000000	0,049841	0,050142
5	16,00000	0,300000	950,000	2,000000	0,043320	0,043480
6	16,00000	0,300000	950,000	2,000000	0,043320	0,043480
7	16,00000	0,300000	950,000	2,000000	0,043320	0,043480
8	16,00000	0,300000	950,000	2,000000	0,043320	0,043480
9	16,00000	0,300000	1000,000	2,000000	0,037663	0,037600
10	16,00000	0,300000	1000,000	2,000000	0,037663	0,037600
11	16,00000	0,300000	1000,000	2,000000	0,037663	0,037600
12	16,00000	0,300000	1000,000	2,000000	0,037663	0,037600
13	16,00000	0,300000	1050,000	2,000000	0,032860	0,032321
14	16,00000	0,300000	1050,000	2,000000	0,032860	0,032321
15	16,00000	0,300000	1050,000	2,000000	0,032860	0,032321
16	16,00000	0,300000	1050,000	2,000000	0,032860	0,032321
17	16,00000	0,300000	900,000	4,000000	0,049497	0,049864
18	16,00000	0,300000	900,000	4,000000	0,049497	0,049864
19	16,00000	0,300000	900,000	4,000000	0,049497	0,049864

Rys. 8. Widok okna z wynikami predykcji dla niedomłotu

Fig. 8. Screen of prediction results for non-threshing

Wnioski

1. Na podstawie przeprowadzonych testów sieci nie jest możliwe utworzenie jednej wspólnej sieci dla niedomłotu i uszkodzeń, gdyż błąd przetwarzania takiej sieci jest zbyt duży.
2. Z różnic pomiędzy sieciami wygenerowanymi dla strat w postaci uszkodzeń i niedomłotu wynika, że większa prostota sieci dla uszkodzeń spowodowana jest łatwiejszą do przewidzenia wielkością zmian tej funkcji, a co za tym idzie bardziej losowym kształtowaniem się wartości niedomłotu.
3. Na podstawie modeli wygenerowanych przez wyuczoną sieć możliwe jest przewidywanie wielkości strat ilościowych i jakościowych dla różnych parametrów regulacyjnych bez konieczności prowadzenia pracochłonnych badań empirycznych.

Bibliografia

Gieroba J., Dreszer K. 1993. Wybrane zagadnienia procesu omlotu i wydzielania ziarna z masy zbożowej. Probl. Inż. Roln., z. 1.

Mikucki K. 1993. Model matematyczny procesu powstawania strat ziarna w kombajnie zbożowym. Zesz. Probl. Post. Nauk Roln., z. 408.

Osowski T. 1996. Sieci neuronowe w ujęciu algorytmicznym. WNT, Warszawa. 1.

Złobecki A. 1995. Zależność strat jakościowych ziarna pszenicy od warunków zbioru kombajnem. Zesz. Probl. Post. Nauk Roln., z. 423.

MODELING THE THRESHING PROCESS WHEN USING ARTIFICIAL NEURAL NETWORKS

Summary

The purpose of the work was the use of neural networks as a simulation – decision-making model of the threshing process in the harvester. The modeling process involved finding the best neural network with the smallest data processing error for non-threshing and damage of grain, respectively. After conducting a learning and testing process best network was specified i.e. such that generated loss value in form of non-threshing and damage of grain with the smallest processing error. As a criterion for selecting the network the smallest error for the validation set was used. Variables presented at the network input were the control parameters of the threshing unit (i.e. rotations of the threshing drum, power supply, adjustable harvester speed, dimensions of working slot) and moisture of threshed corn. At the output of both networks loss or non-threshing values were obtained depending on the input variable values. The modeling process was performed using "Statistica Neural Network" program by Statsoft. The obtained models were tested, launched and saved. These models can be used in the harvester cropping automation process. The created neural networks can be used on the harvester on-board computer, which would set the control parameters according to the grain designation to minimize non-threshing or losses.

Key words: harvester, threshing process, losses, non-threshing, damages, artificial neural networks