PIOTR BONIECKI, WOJCIECH MUELLER
Instytut Inżynierii Rolniczej
Akademia Rolnicza im. A. Cieszkowskiego w Poznaniu

THE NEURONAL CLASSIFYING TECHNIQUES IN PROBLEMS OF IDENTIFICATION OF AGRICULTURAL ENGINEERING

Summary

*The aim of the work was discussion of basic classifying techniques in context of their utilisation in investigative problems of agricultural engineering. The chosen topology of artificial neural networks were showed as effective classifying tools. Creation of the computer system "The neuronal nets - Perceptron " was the additional effect of the conducted analysis, helping the process of education. The aim of the created computer program is to classify the data obtained from the area of agricultural engineering. The program acts on the basis of many-layered network of perceptron type - MLP (**M**ultiLayer **P**erceptron).*

NEURONOWE TECHNIKI KLASYFIKACYJNE W PROBLEMACH IDENTYFIKACYJNYCH INŻYNIERII ROLNICZEJ

Streszczenie

*Celem pracy było omówienie podstawowych technik klasyfikacyjnych w kontekście wykorzystania ich w problemach badawczych inżynierii rolniczej. Wskazano wybrane topologie sztucznych sieci neuronowych jako efektywne narzędzia klasyfikacyjne. Dodatkowym efektem przeprowadzonej analizy było wytworzenie systemu informatycznego „Sieci neuronowe - Perceptron" wspomagającego proces edukacji. Wytworzony program komputerowy ma za zadanie klasyfikować dane zaczerpnięte z obszaru inżynierii rolniczej. Program działa w oparciu o sieć wielowarstwową typu perceptron - MLP (**M**ulti**L**ayer **P**erceptron).*

Wstęp

W ostatnich latach daje się zauważyć dynamiczny rozwój nowych technologii informacyjnych, mających zastosowanie również w rolnictwie. Zwłaszcza szeroko rozumiana informatyka służy coraz częściej do wspomagania, bądź kontrolowania nowoczesnej techniki rolniczej. Należy podkreślić, że coraz częściej w inżynierii rolniczej z powodzeniem stosowane są metody sztucznych sieci neuronowych. Sieci neuronowe bez najmniejszych problemów odwzorowują złożone zbiory danych empirycznych, posiadają zdolność uczenia i przystosowania się do zmiennych warunków, jak również potrafią uogólniać zdobytą wiedzę.

W pracy pokrótce omówiono wybrane problemy klasyfikacyjne w kontekście ich realizacji przez symulacyjne modele neuronowe. Wskazano przykładowe zastosowania neuronowych technik klasyfikacyjnych w inżynierii rolniczej, akcentując użyteczność sztucznych sieci neuronowych w badaniach prowadzonych w obszarze inżynierii rolniczej. Dodatkowym celem pracy było wytworzenie, weryfikacja oraz przetestowanie systemu informatycznego symulującego działanie sieci neuronowych typu *MLP* oraz *RBF* w kontekście ich wykorzystania w zagadnieniach klasyfikacyjnych, a w szczególności jako narzędzi identyfikacyjnych [1].

Krótka charakterystyka problemów klasyfikacyjnych

Klasyfikacje można rozumieć jako podział dowolnego zbioru elementów na grupy według określonych wcześniej kryteriów. Do grup zaliczamy elementy podobne i różniące się, które reprezentują własności charakteryzujące daną grupę. Elementy zbioru przypisane do jednej grupy zdefiniowane są klasą, a element klasy obiektem. W zależności od posiadanych informacji dotyczących struktury zbioru, klasyfikację dzielimy na:

- wzorcową: konstrukcja idei jest wiadoma, posiada ona charakterystykę klasy, pochodzącą od obiektów. Jest to przypadek uczenia sieci neuronowych (SN) z nauczycielem bądź pod nadzorem;
- bezwzorcową: jest to badanie skupień (klasteryzacja), występuje tu uczenie sieci neuronowych (SN) bez nauczyciela.

Poniżej przedstawiono przykładowy zbór przypadków w następującej postaci:

$$\left\{x_i, A^{(i)}\right\} \quad i = 1...n, \tag{1}$$

gdzie:

x_i - jest wektorem *n*-wymiarowym,

$A^{(i)} \in A_1, A_2, ..., A_L$ - jest etykietą klasy.

Pod postacią etykiety może występować ciąg znaków będący np. nazwą klasy, jak również unikatowym numerem klasy. Etykieta ciągu znaków oznaczona jest przez symbol *A*. Natomiast etykieta pod postacią numeru klasy, opisana jest przez symbol *d*. Stan etykiety klasy uwarunkowany jest od algorytmu, który stosowany jest w klasyfikacji danych. Niekiedy algorytmy posiadają etykiety o charakterze nazw (np. drzewa decyzyjne), natomiast inne posiadają jednostkę numeryczną (np. większość sieci neuronowych). Odpowiednie składowe wektora *x* określają obiekt i nazwane są cechami.

Metody klasyfikujące (klasyfikatory), prowadzą mapowanie wektorów wejściowych *x* na etykietę klasy *A*. Mapowanie sprowadza się do adaptacyjnego doboru atrybutów *W* określonego typu, który winien spełniać zależność:

$$y(x_i;W) = A^{(i)} \quad i = 1...n, \tag{2}$$

gdzie:
y - jest funkcją macierzową,
x_i - jest wektorem n-wymiarowym,
W - jest zbiorem parametrów klasyfikatora,
$A^{(i)}$ - jest etykietą klasy,
i - jest indeksem,
n - jest liczbą naturalną.

Dostosowanie atrybutów to proces adaptacyjny, który w przypadku sztucznych sieci neuronowych zdefiniowany jest jako uczenie lub trenowanie sieci. Natomiast dane na podstawie których nastąpi określenie wartości stałej nazywamy danymi treningowymi lub zbiorem uczącym.

Efektem poprawnego uczenia sztucznej sieci neuronowej, oprócz trafności przystosowania się SN do danych treningowych, jest zdolność do generalizacji modelu, czyli wartość klasyfikacji na danych, które nie były stosowane w trakcie uczenia modelu (dane testowe).
System klasyfikacyjny powinien być uczony na przykładzie adekwatnym tzn. na takim, który spełnia następujące założenia:

1. element przykładu (zbioru) powinien być uzyskany systemem losowym z określonych klas (populacji),
2. próbkę należy pobrać tak, aby była wystarczająco liczebna.

Kryterium zawarte w punkcie drugim jest często trudne do spełnienia m.in. ze względu na jego rozmyty charakter. Jest ono zależne od wielu parametrów, takich jak np. skala nakładania na siebie klas, skala zaszumienia danych, grupy cech określających obiekty w przebiegu uczącym. Stwierdza się, że im zadanie jest większe tym bardziej udane jest odzwierciedlenie cech określonych klas.

Przestrzenie decyzyjne w problemach klasyfikacyjnych

Z geometrycznego punktu, wektorowi x z przebiegu ćwiczebnego można przypisać miejsca o ustalonych odpowiednio obszarach cech (obserwacji) o rozmiarze równym rozmiarowi wektora uczącego. Funkcja klasyfikująca tworzy granice dla powierzchni cech na przestrzeniach decyzyjnych. Punktom w tej przestrzeni jest przypisana taka sam klasa (decyzja). Tworzące się granice między tymi przestrzeniami zdefiniowane są jako granice decyzyjne bądź hiperpowierzchnie decyzyjne. Granice decyzyjne są „terenem" w przestrzeni cech, gdzie niektóre funkcje kryterialne (funkcje przynależności do klasy) posiadają takie same walory [3].
Postaci obszaru decyzji są zależne od struktury danych pierwotnych oraz stosowanego modelu klasyfikacyjnego. Odpowiednia granica decyzyjna tworzona jest na podstawie prawdopodobieństwa *a posteriori*. Prawdopodobieństwo *a priori* jest klasycznym prawdopodobieństwem obliczanym przed wykonaniem doświadczenia. Natomiast prawdopodobieństwo *a posteriori* jest obliczane po dokonaniu doświadczenia. Idealny system klasyfikujący dobiera klasę, w której prawdopodobieństwo *a posteriori* jest zwiększone. Wektor x klasyfikowany jest do klasy A_k wtedy, gdy realizowana jest poniższa deklaracja:

$$p(A_k \mid x) > p(A_j \mid x) \quad j \neq k, \tag{3}$$

gdzie:
$p(A_j \mid x)$ - to prawdopodobieństwo *a posteriori*, ale przy wzorcu x znajdującym się w klasie A_j,
$p(A_k \mid x)$ - to prawdopodobieństwo *a posteriori* przy wzorcu x znajdującym się w klasie A_k.

Stosując klasyczną zależność *Bayesa* w postaci:

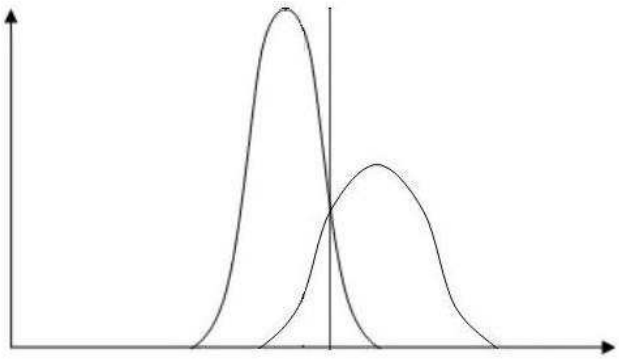
$$p(A_k \mid x) = \frac{p(x \mid A_k)\, p(A_k)}{p(x)}, \tag{4}$$

można uzyskać:

$$p(x \mid A_k)\, p(A_k) > p(x \mid A_j)\, p(A_j) \quad k \neq j, \tag{5}$$

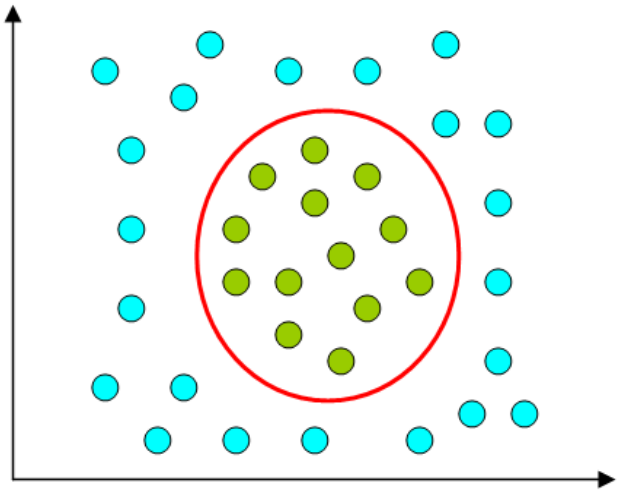
gdzie:
$p(x \mid A_k)$ - to prawdopodobieństwo *a priori*, że obiekt x należy do populacji A_k,
$p(A_k)$ - jest prawdopodobieństwem *a priori* w klasie A_k,
$p(x)$ - to obserwowane prawdopodobieństwo x,
$p(x \mid A_j)$ - to prawdopodobieństwo *a priori*, że obiekt x należy do populacji A_k,
$p(A_j)$ - jest prawdopodobieństwem *a priori* w klasie A_j.

Zależność (5) ustala granice decyzyjna pomiędzy klasami znajdującymi się na styku tych klas. Rys. 1. przedstawia przykład jednowymiarowy.



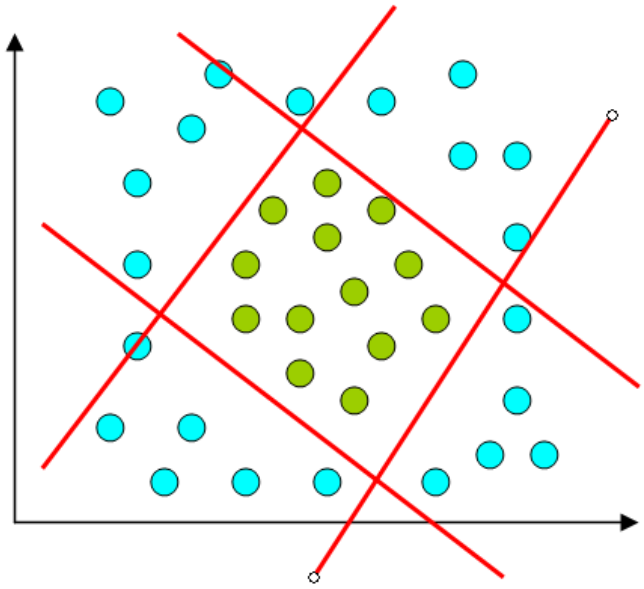
Rys. 1. *Bayesowska* granica decyzyjna
Fig. 1. Decision border of Bayes

Najczęściej kształt obszaru decyzyjnego występującego w eksplorowanych danych nie jest znany. Dlatego stosuje się różne modele, adekwatne dla posiadanych danych i bada ich generalizację rozumianą jako rozszerzenie i uogólnienie danego zagadnienia. Lepszym modelem (w kontekście jakości predykcji) jest ten, w którym występuje największa generalizacja. W razie gdyby generalizacja była taka sama dla kilku modeli, za wygraną zostaje wybrana ta, która posiada dodatkowe pożądane cechy, takie jak np. najprostsza budowa, zdolność do podejmowana decyzji bądź też jest łatwa w użytkowaniu. Przykładowe obszary decyzyjne pokazano na rys. 2.



Rys. 2. Optymalna granica decyzyjna
Fig. 2. Optimal decision border

Optymalnym sposobem odizolowania dwóch klas dla przypadku przedstawionego na rys. 2. jest stworzenie granicy w postaci okręgu (klasyfikacja jądrowa lub lokalna). W przypadku, gdy próbujemy dokonać klasyfikacji w obszarze tego samego zbioru za pomocą linii prostych (klasyfikacja globalna), uzyskany wynik może być daleki od optymalnego [4] (rys. 3.).



Rys. 3. Nieoptymalna granica decyzyjna
Fig. 3. Unoptimal decision border

Wyznaczony obszar decyzyjny jest stworzony z wykorzystaniem czterech prostych. Ilość użytych parametrów jest znaczna, w porównaniu z poprzednim modelem przedstawionym na rys.2, na którym obszar decyzyjny wyznaczony jest przy pomocy okręgu. Stosując nieoptymalną granice decyzyjną jesteśmy narażeni m.in. na dłuższy czas uczenia [4].

Ocena jakości klasyfikacji

Istnieje wiele metod oceny jakości klasyfikacji. Wybierane są one w zależności od czasu uczenia, wielkości zbioru uczącego i od tego czy w strukturze danych znajduje się zbiór testowy.

Przy braku występowania zbioru testującego, uogólnienia dokonuje się za pomocą nieplanowego podziału danych na dane treningowe i testowe. Najczęściej wykorzystywane są trzy metody podziału: kroswalidacja, kroswalidacja Monte Carlo oraz bootstrapping.

Kroswalidacja

Przypuśćmy, że posiadamy zbiór ćwiczebny C składający się z rozdzielnych podzbiorów S_i o podobnej ilości

$$C = \left\{ \bigcup_i S_i ; \underset{m \neq p}{\forall} S_m \cap S_p = 0, \underset{m \neq p}{\forall} |S_m| \approx |S_p| \right\},$$

w którym: $i = 1...n$.
Klasyfikator będzie poddany uczeniu n-krotnej kroswalidacji na zbiorze S_k^{CR} takim, gdzie $S_k^{CR} = \left\{ \bigcup_{l \neq k} S_l \right\}$ oraz sprawdzaniu na zbiorze $S_k^{CR} = \{S_k\}$ nie pojawiającym się w S_k^{CR}. Testowanie i uczenie odbywa się n-krotnie, dla indeksu $k = 1...n$. Ostateczna jakość klasyfikacji obliczana jest przy pomocy przeciętnych cech dla wszystkich k. Odchylenie od średniej dokładności można ocenić przy pomocy wariancji wyników.

Kroswalidacja Monte Carlo

W metodzie kroswalidacji Monte Carlo zbiór dzielony jest w sposób losowy. Najczęściej jest to podział 2 do 3. Zbiór dzieli się na zbiór treningowy i testowy. Gdy model zostaje nauczony przechodzi testowanie. Natomiast wartość generalizacji jest zapamiętywana. Przebieg ten jest powielany wielokrotnie. Średnia z cech częściowych określana jest poprzez finalną wartość generalizacji. Późniejsze podziały nie są zależne od rozdzielnych podzbiorów.

Bootstrapping

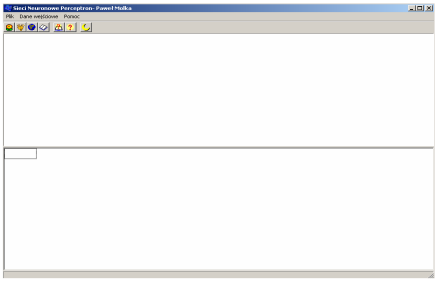
W tym przebiegu występuje dobieranie z podstawianiem. Dla określonej liczby zdarzeń, dokonywany jest systemem nieplanowany wybór takiej samej liczby wektorów z ciągu treningowego. Wybór ten jest dokonywany z powtórkami. Po wyuczeniu następuje testowanie algorytmu dla kolejnej części danych. Tak samo jak w kroswalidacji Monte Carlo podziały te nie są uzależnione od rozdzielnych podzbiorów.

Sieci neuronowe w zagadnieniach klasyfikacyjnych

Głównym celem stawianym siecią neuronowym w zagadnieniu klasyfikacji, jest odpowiednie przypisanie danych wejściowych do wybranej klasy. Do klasyfikacji najczęściej wykorzystuje się następujące sieci: perceptron wielowarstwowy, sieci o radialnych funkcjach bazowych oraz sieci *Kohonena* [1, 3]. Klasyfikacyjne problemy w sieciach neuronowych można podzielić na dwie kategorie. Pierwszym, a zarazem najprostszym, jest problem dwuklasowy, do którego wykorzystuje się jeden neuron wyjściowy. Zadana wartość wyjściowa może być w tym przypadku równa 1 lub 0, co charakteryzuje przynależność do jednej z dwóch klas. Bardziej złożonym zagadnieniem jest problem wieloklasowy. W tym przypadku neuronowy klasyfikator wykorzystuje w swoim działaniu więcej niż jeden neuron na wyjściu.

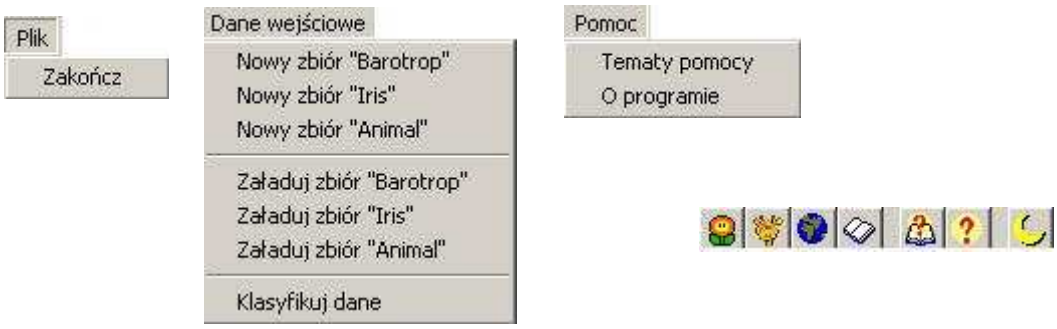
Opis wytworzonej aplikacji

W ramach prowadzonej analizy powstał system informatyczny, którego celem była klasyfikacja danych empirycznych zaczerpniętych z obszaru szeroko rozumianego rolnictwa. Wytworzony oraz przetestowany program „Sieć neuronowa - perceptron" służy do klasyfikacji pozyskanych danych w oparciu o sztuczną sieć neuronową typu MLP (*Multilayer Perceptron*) [3]. Po zainstalowaniu, a następnie uruchomieniu programu, pojawia się główny interfejs użytkownika, który jest przedstawiony na rys. 4.



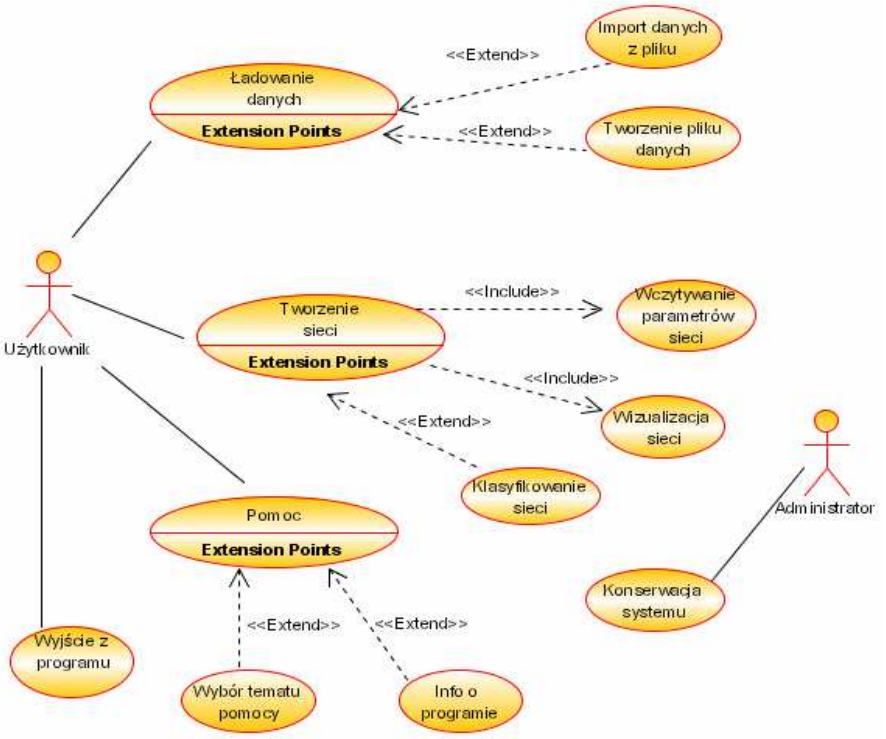
Rys. 4. Interfejs użytkownika programu
Fig.4. User's interface of the program

Menu aplikacji składa się z następujących elementów:



Rys. 5. Paski menu i pasek narzędzi
Fig. 5. Menu belts and belt of tools

Wymagania funkcjonalne stawiane wytworzonej aplikacji obrazuje diagram przypadków użycia dla systemu „Sieci neuronowe - Perceptron", przedstawiony na rys. 6.



Rys. 6. Diagram przypadków użycia dla systemu „Sieci neuronowe - Perceptron"
Fig. 6. Diagram of use cases for the "Neural Networks - Perceptron" system

System informatyczny „Siec neuronowa – Perceptron" klasyfikuje dane w oparciu o sieci typu MLP, wygenerowane oraz wyuczone w programie *Sieci Neuronowe*, stanowiącym moduł pakietu STATISTICA v.6.0. W przypadku błędnej odpowiedzi sieci przy klasyfikacji pojawia się opis (błąd). Program wyposażony jest w plik pomocy, która umożliwia zapoznanie się z podstawowymi informacjami na temat sieci neuronowych oraz zawiera instrukcję obsługi aplikacji.

Testowanie oprogramowania

Po uruchomieniu programu na platformie sprzętowej priorytetem było poddanie go ocenie i weryfikacji w aspekcie jego poprawności funkcjonowania oraz zgodności z wymogami stawianymi w poszczególnych fazach cyklu życia oprogramowania. Wytworzone oprogramowanie zastało przetestowane zarówno pod względem specyfikacji wymagań jak również jego funkcjonalności. Na początku zostały ustalone najistotniejsze detale systemu, bez których oprogramowanie nie mogłoby poprawnie pracować. Dalej zostały one przetestowane pod względem ich bezbłędności działania. Kolejnym etapem testowania była weryfikacja interfejsu użytkownika. Testy polegały na sprawdzeniu czy badany interfejs jest stosowany właściwie, czy jest zrozumiały, czy nie znajdą się niezgodności w stosunku do wytycznych systemu. Zbadano również, czy określone części interfejsu zachowują się jednolicie w kontekście swego przeznaczenia. Bezbłędność kodu była sprawdzana w momencie planowania i wykryte błędy zostały wyeliminowane. Należy podkreślić, że wytworzone oprogramowanie zostało zaimplementowane w środowisku obiektowym na platformie firmy *Borland*, co ułatwia późniejszy rozwój oprogramowania a także pozwala na samodzielną prace programu na różnorodnych platformach sprzętowych.

Uwagi końcowe

W oparciu o przeprowadzone rozważania, oraz biorąc pod uwagę wnioski jakie nasuwają się podczas eksploatacji wytworzonego systemu informatycznego „Sieć neuronowa – Perceptron", można sformułować następujące uwagi końcowe:

1. Na podstawie analizy technik neuronowych stwierdzono, że modele neuronowe są adekwatnym narzędziem klasyfikacyjnym, mogącym mieć zastosowanie w obszarze inżynierii rolniczej.
2. Wytworzony system informatyczny „Sieć neuronowa – Perceptron" spełnia przyjęte założenia. System klasyfikuje dane zaczerpnięte z obszaru inżynierii rolniczej, które importowane są z pliku bądź wprowadzane bezpośrednio przez użytkownika. Przeprowadzony cykl weryfikacyjny oraz testowy pozwala stwierdzić, że system działa poprawnie.
3. Zaprojektowane oraz przetestowane oprogramowanie spełnia oczekiwania funkcjonalne założone w fazie określenia wymagań. Przedstawiony diagram użycia przypadków ukazuje zależności miedzy systemem a użytkownikiem.
4. System informatyczny „Sieć neuronowa – Perceptron" może stanowić poglądowe wsparcie w procesie edukacji w zakresie wykorzystania sztucznych sieci neuronowych jako narzędzia klasyfikacyjnego.

Literatura

[1] Rutkowska D., Piliński M., Rutkowski L. (1997). Sieci neuronowe, algorytmy genetyczne i systemy rozmyte: Wydawnictwo Naukowe PWN, Warszawa-Łódź

[2] Osowski S. (2000). Sieci neuronowe do przetwarzania informacji: Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa

[3] Hertz J., Krogh A., Palmer R. G. (1993). Wstęp do teorii obliczeń neuronowych: WNT Warszawa

[4] Boniecki P. (2004). Sieci neuronowe typu MLP oraz RGB jako komplementarne modele aproksymacyjne w procesie predykcji plonu pszenżyta: Journal of Research and Applications in Agricultural Engineering, Poznań, 1'2004, Vol. 49(1), str. 28-33.