

Marcin Pelka

Uniwersytet Ekonomiczny we Wrocławiu

SIECI NEURONOWE DLA DANYCH SYMBOLICZNYCH: PERCEPTRON WIELOWARSTWOWY

1. Wstęp

Jedną z najczęściej stosowanych struktur sieci neuropodobnych jest wielowarstwowa sieć perceptronowa [Duda, Hart, Stork 2001, s. 282]. Celem artykułu jest zaprezentowanie metod transformacji zmiennych symbolicznych na potrzeby perceptronu wielowarstwowego (MLP) [Rossi, Conan-Guez 2002] oraz porównanie wpływu wyboru metody transformacji na błąd klasyfikacji na przykładzie zbiorów ze zmiennymi zakłócającymi i obserwacjami odstającymi. Perceptron wielowarstwowy dla danych symbolicznych wymaga, aby danymi wejściowymi były obiekty symboliczne opisywane wyłącznie przez zmienne w postaci przedziałów liczbowych. W artykule przedstawiono sposoby zamiany przedziałów liczbowych (transformacji) w liczby rzeczywiste na potrzeby perceptronu wielowarstwowego dla danych w ujęciu klasycznym.

W części empirycznej artykułu porównano metody pozwalające na zastosowanie przedziałów liczbowych w MLP na przykładowych danych symbolicznych. Wykorzystano sztucznie wygenerowane zbiory obiektów opisywane przez zmienne symboliczne w postaci przedziałów liczbowych (z wykorzystaniem procedury `cluster.Gen` z pakietu `clusterSim` dla programu R), w tym także zbiory ze zmiennymi zakłócającymi i obserwacjami odstającymi (*outliers*). Do uczenia sieci neuronowej (perceptronu wielowarstwowego) wykorzystano pakiet `neural` dla programu R (procedury `mlp-trainimlp`). Do oceny wyników wykorzystano błąd średniokwadratowy (MSE).

2. Typy zmiennych w analizie danych symbolicznych

W przypadku obiektów symbolicznych możemy mieć do czynienia z rodzajami zmiennych, takimi jak [*Analysis of symbolic...* 2000, s. 2-3]:

- 1) ilorazowe, przedziałowe, porządkowe, nominalne;
- 2) kategorie, np. biały, zielony;

3) przedziały liczbowe, rozłączne lub nierozłączne, np. ilość spalanej benzyny na 100 km w pewnym samochodzie (6 litrów; 13 litrów);

4) lista kategorii – tu przykładem może być typ nadwozia samochodu pewnej marki: sedan, hatchback, minivan, coupé, co oznacza, że dostępny jest on w czterech różnych wariantach nadwozia;

5) lista kategorii z wagami (prawdopodobieństwami), gdzie oprócz listy kategorii występują wagi (prawdopodobieństwa), z jakimi obiekt posiada wybraną kategorię, np. jeżeli wybrać zmienną wybrane kolory nadwozia dla pewnej marki i modelu samochodu: biały (0,45), zielony (0,30), czarny (0,15), to oznacza to, że możemy kupić samochód w kolorze białym i zielonym, natomiast kolor czarny jest o wiele mniej spotykany. Sytuacja taka może wynikać m.in. z polityki producenta czy popularności kolorów;

6) zmienne strukturalne [*Analysis of symbolic...* 2000, s. 33-37] – w literaturze przedmiotu wyróżnia się oprócz wyżej wymienionych typów zmiennych także zmienne strukturalne:

a) zmienne o zależności funkcyjnej lub logicznej pomiędzy poszczególnymi zmiennymi, gdzie *a priori* ustalono reguły funkcyjne lub logiczne decydujące o tym, jaką wartość przyjmie dana zmienna,

b) zmienne hierarchiczne, w których *a priori* ustalono warunki, od których zależy, czy zmienna dotyczy danego obiektu, czy też nie,

c) zmienne taksonomiczne, w których *a priori* ustalono systematykę, według której przyjmuje ona swoje realizacje.

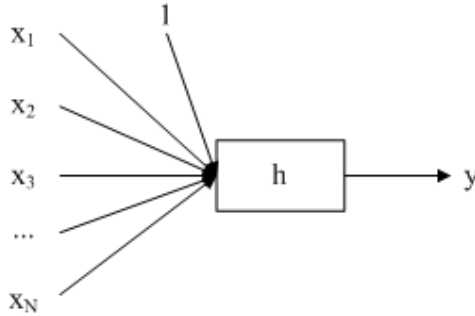
W artykule Rossiego i Conana-Gueza [2002] zaproponowano rozwiązanie dla perceptronu wielowarstwowego, które umożliwia analizowanie obiektów symbolicznych opisywanych jedynie przez zmienne w postaci przedziałów liczbowych [Rossi, Conan-Guez 2002, s. 427].

3. Perceptron wielowarstwowy

Perceptron wielowarstwowo, który jest jednym z rodzajów sieci neuronowych [Duda, Hart, Stork 2001, s. 324-326], jest siecią jednokierunkową (*feed-forward neural network*), która dobrze sprawdza się w klasyfikacji czy dyskryminacji zbiorów o niezbyt dobrze wyodrębnionej strukturze klas. Z punktu widzenia statystyki sieć neuronowa to pewna klasa modeli liniowych lub nieliniowych, zależnie od struktury sieci [Gatnar 2008, s. 46].

Dla zrozumienia działania perceptronu wielowarstwowego konieczne jest przedstawienie opisu działania pojedynczego neuronu (zob. rys. 1).

Sygnal wejściowy $\mathbf{x}_i = [x_{i1}, \dots, x_{iL}]$ jest przekształcany w sygnał wyjściowy za pomocą funkcji aktywacji (*activation function*) – oznaczonej na rys. 1 jako h – oraz wektora wag $\mathbf{w} = [w_0, w_1, \dots, w_L]$. Waga w_0 oznacza wagę przypisaną elementowi 1 na rys. 1. Jest to dodatkowy sygnał wejściowy, który oznacza próg aktywacji [Gatnar 2008, s. 46; Duda, Hart, Stork 2001, s. 285].



Rys. 1. Model pojedynczego neuronu

Źródło: [Gatnar 2008, s. 46].

Dla każdego z wejść neuronu dokonywane jest przekształcenie sygnału wejściowego zgodnie ze wzorem [Gatnar 2008, s. 46]:

$$\hat{y} = h \left(w_0 + \sum_{l=1}^L w_l x_{il} \right), \quad (1)$$

gdzie: w_0 – waga przypisana dodatkowemu sygnałowi wejściowemu, w_i – i -ta waga, x_{il} – i -ty sygnał wejściowy dla l -tej zmiennej, $l = 1, \dots, L$ – numer zmiennej.

Typowymi funkcjami aktywacji, których dobór wpływa na właściwość uczenia się sieci, są funkcje [Gatnar 2008, s. 46-47; Duda, Hart, Stork 2001, s. 285, 308-309]:

1. Progowa (unipolarna, skoku jednostkowego):

$$h(u) = \begin{cases} 1 & \text{gdy } u > 0 \\ 0 & \text{gdy } u \leq 0 \end{cases}, \quad (2)$$

gdzie: $u = \left(w_0 + \sum_{l=1}^L w_l x_{il} \right)$.

2. Bipolarna (signum):

$$h(u) = \begin{cases} 1 & \text{gdy } u > 0 \\ -1 & \text{gdy } u \leq 0 \end{cases}, \quad (3)$$

gdzie: oznaczenia jak we wzorze 2.

3. Identyfnościowa:

$$h(u) = u. \quad (4)$$

4. Liniowa:

$$h(u) = ku, \quad (5)$$

gdzie: k – jest stałą, różną od zera.

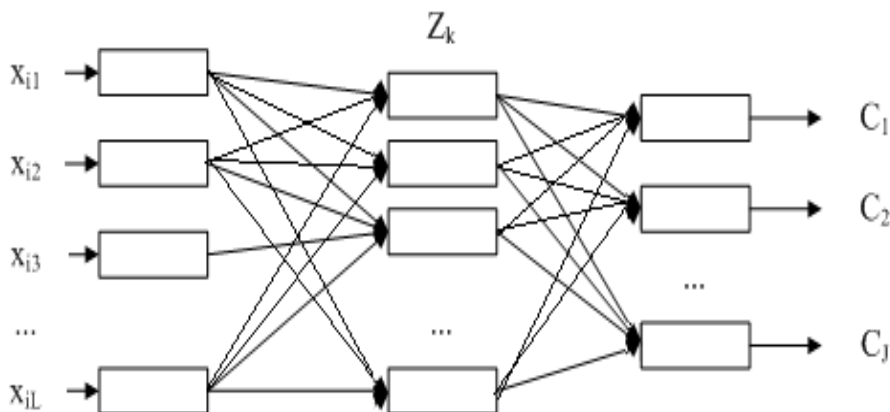
5. Sigmoidalna:

$$h(u) = \frac{1}{1 + \exp(-u)}. \quad (6)$$

6. Tangens hiperboliczny:

$$h(u) = \operatorname{tgh}\left(\frac{u}{2}\right) = \frac{1 - \exp(u)}{1 + \exp(u)}. \quad (7)$$

Na rysunku 2 przedstawiono model perceptronu z jedną warstwą ukrytą.



Rys. 2. Model perceptronu z jedną warstwą ukrytą

Źródło: [Gatnar 2008, s. 49; Duda, Hart, Stork 2001, s. 290].

Perceptron wielowarstwowy jest siecią jednokierunkową, w której nie występują sprzężenia zwrotne. Zmienne Z_k nie są bezpośrednio obserwowalne i tworzą tzw. warstwę ukrytą (*hidden layer*). Algorytm uczenia perceptronu wielowarstwowego wykorzystuje tzw. wsteczną propagację błędów (*error backpropagation*) [Gatnar 2008, s. 49; Duda, Hart, Stork 2001, s. 288].

Ogólnie wsteczną propagację błędów można ująć w dwóch krokach [Gatnar 2008, s. 52]:

1. Krok w przód (*forward pass*), który polega na obliczeniu wartości teoretycznych zgodnie z funkcjami aktywacji dla obserwacji ze zbioru uczącego.

2. Krok wstecz (*backward pass*), w którym oblicza się wartość błędu dla każdego wiersza ze zbioru uczącego, a następnie dokonywana jest minimalizacja funkcji błędu [Gatnar 2008, s. 50]:

$$Q(\mathbf{w}, \mathbf{v}) = \sum_{i=1}^N Q_i(\mathbf{w}, \mathbf{v}), \quad (8)$$

gdzie: $Q(\mathbf{w}, \mathbf{v})$ – miara błędu dopasowania,

\mathbf{w}, \mathbf{v} – wektory parametrów dla neuronów w warstwach sieci.

Następnie minimalizuje się funkcję błędu dopasowania metodą gradientową. Otrzymaone gradienty pozwalają na modyfikację wag synaptycznych zgodnie ze wzorami [Gatnar 2008, s. 50]:

$$w_{kl}^{(r+1)} = w_{kl}^{(r)} - \eta \sum_{i=1}^N \frac{\partial Q_i(\mathbf{w}, \mathbf{v})}{\partial w_{kl}^{(r)}}, \quad (9)$$

$$v_{jk}^{(r+1)} = v_{jk}^{(r)} - \eta \sum_{i=1}^N \frac{\partial Q_i(\mathbf{w}, \mathbf{v})}{\partial v_{jk}^{(r)}}, \quad (10)$$

gdzie: $w_{kl}^{(r+1)}, w_{kl}^{(r)}, v_{jk}^{(r+1)}, v_{jk}^{(r)}$ – wagi synaptyczne w warstwach sieci, odpowiednio o wymiarach $L \times K$ oraz $K \times J$,

η – współczynnik szybkości uczenia, parametr z przedziału (0; 1).

4. Adaptacja zmiennych symbolicznych w postaci przedziałów liczbowych na potrzeby perceptronu wielowarstwowego

Adaptacja zmiennych symbolicznych w postaci przedziałów liczbowych na potrzeby perceptronu wielowarstwowego sprowadza się do przekodowania (transformacji) zmiennych symbolicznych w klasyczne. W artykule Rossiego i Conana-Gueza [2002] zaproponowano trzy podejścia do zamiany przedziałów liczbowych zmiennej symbolicznej w zmienne klasyczne [Rossi, Conan-Guez 2002, s. 428-429], takie jak:

1. Metoda ekstremów – w metodzie tej przedział liczbowy zmiennej zastępowany jest przez dwie zmienne klasyczne po jednej dla każdego z krańców przedziału liczbowego, np. zmienna pojemność silnika V_1 (1200; 1800) zostanie zastąpiona dwiema zmiennymi: $V_{1\min} = 1200$ i $V_{1\max} = 1800$. Problematyczne wydaje się w tym przypadku sztuczne zwiększenie liczby zmiennych.

2. Metoda środków – przedział liczbowy zmiennej zastępowany jest środkiem tego przedziału, np. spalanie paliwa na 100 km V_1 (6; 8) zostanie zastąpiona zmienną $V_{1\text{środ}} = 7$. W tym przypadku istnieje możliwość zastąpienia przedziałów o różnej rozpiętości, lecz o tym samym środku tą samą liczbą.

3. Metoda próbkowania (losowania) – w metodzie tej przedział liczbowy zmiennej jest zastępowany zmienną losową o rozkładzie jednostajnym o krańcach równych krańcom zmiennej symbolicznej. Na przykład zmienna preferowana cena samochodu V_1 (25 000; 36 000) zostanie zastąpiona zmienną $V_{1\text{prob}} = 26\,373,88$. W metodzie tej autorzy nie podają, ilu losowań należy dokonać. Na potrzeby badań empirycznych przyjęto wynik jednego losowania.

5. Badania symulacyjne

Na potrzeby porównania zamiany zmiennych symbolicznych w postaci przedziału liczbowego, które zaproponowano w artykule Rossiego i Conana-Gueza, przygotowano pięć modeli o znanej strukturze klas (do generowania modeli wykorzystano polecenie `cluster.Gen` z pakietu `clusterSim` dla **R**):

1. **Model 1** – 400 obiektów podzielonych na dwie dobrze separowalne klasy o wydłużonym kształcie. Obiekty opisywane są przez dwie zmienne symboliczne w postaci przedziałów liczbowych. Zmienne te są losowane niezależnie z dwuwymiarowego rozkładu normalnego o średnich $(0, 0)$, $(1, 5)$ oraz macierzy kowariancji $\sum(\sigma_{jj} = 1, \sigma_{jl} = -0,9)$.

Brak zmiennych zakłócających czy obserwacji odstających.

2. **Model 2** – model 1, do którego dodano 15 obserwacji odstających.

3. **Model 3** – model 1, do którego dodana została jedna zmienna zakłócająca oraz 15 obiektów odstających.

4. **Model 4** – 400 obiektów podzielonych na pięć niezbyt dobrze separowalnych klas. Obiekty są opisywane przez trzy zmienne symboliczne w postaci przedziałów liczbowych, zmienne są losowane niezależnie z wielowymiarowego rozkładu normalnego o średnich $(5, 5, 5)$, $(-3, 3, -3)$, $(3, -3, 3)$ oraz macierzy kowariancji, gdzie $\sigma_{jj} = 1$ ($1 \leq j \leq 3$), $\sigma_{jl} = 0,9$ ($1 \leq j \neq l \leq 3$). Brak zmiennych zakłócających czy obiektów odstających.

5. **Model 5** – model 4, do którego dodano 20 obiektów odstających.

W każdym z modeli zbiór uczący to 300 obiektów, a zbiór testowy to 100 obiektów.

Dla każdego z modeli dokonano dziewięciu różnych symulacji różniących się: liczbą iteracji, liczbą warstw i neuronów w perceptronie oraz współczynnikiem uczenia. Charakterystykę poszczególnych symulacji zawarto w tab. 1.

Tabela 1. Charakterystyka ścieżek symulacji

Numer symulacji	I	II	III	IV	V	VI	VII	VIII	IX
Liczba iteracji (w tys.)	3	3	30	30	3	30	3	30	30
Liczba warstw	1	1	1	1	2	2	3	3	3
Liczba neuronów w warstwach	2	2	3	4	2	4	2	2	2, 1, 2
Współczynnik uczenia	0,2	0,6	0,2	0,2	0,2	0,2	0,2	0,6	0,2
Funkcje aktywacji	sigmoidalne								

Źródło: opracowanie własne.

Dodatkowo w ramach poszczególnych symulacji porównano trzy metody zamiany przedziałowych zmiennych symbolicznych w klasyczne: metodę ekstremów, metodę środków, metodę próbkowania.

Tabela 2. Wyniki badań symulacyjnych

Numer modelu Symulacja	1	2	3	4	5
I	E) 0,09 S) 0,16 L) 0,13	E) 0,25 S) 0,12 L) 0,09	E) 2,42 S) 2,40 L) 2,35	E) 0,51 S) 0,05 L) 0,06	E) 0,30 S) 0,35 L) 0,47
II	E) 0,23 S) 0,06 L) 0,32	E) 0,22 S) 0,08 L) 0,09	E) 2,54 S) 2,41 L) 2,97	E) 0,47 S) 0,03 L) 0,08	E) 0,15 S) 0,34 L) 0,20
III	E) 0,13 S) 0,23 L) 0,04	E) 0,18 S) 0,14 L) 0,16	E) 2,24 S) 2,35 L) 1,05	E) 0,003 S) 0,002 L) 0,004	E) 0,24 S) 0,38 L) 0,35
IV	E) 0,05 S) 0,13 L) 0,05	E) 1,15 S) 1,08 L) 0,51	E) 0,92 S) 1,09 L) 1,16	E) 0,001 S) 0,001 L) 0,002	E) 0,27 S) 0,24 L) 0,16
V	E) 0,39 S) 0,31 L) 0,32	E) 0,25 S) 0,12 L) 0,10	E) 2,60 S) 2,14 L) 2,62	E) 0,41 S) 0,01 L) 0,02	E) 0,18 S) 0,34 L) 0,45
VI	E) 0,15 S) 0,16 L) 0,13	E) 0,16 S) 0,38 L) 0,96	E) 1,48 S) 1,29 L) 1,44	E) 0,002 S) 0,0003 L) 0,0002	E) 0,09 S) 0,12 L) 0,10
VII	E) 0,45 S) 0,56 L) 0,49	E) 0,26 S) 0,29 L) 0,48	E) 2,49 S) 2,51 L) 2,71	E) 0,97 S) 0,004 L) 0,01	E) 0,18 S) 0,35 L) 0,36
VIII	E) 0,23 S) 0,44 L) 0,38	E) 0,10 S) 0,12 L) 0,07	E) 1,51 S) 1,46 L) 1,56	E) 0,28 S) 0,09 L) 0,14	E) 0,11 S) 0,28 L) 0,27
IX	E) 0,38 S) 0,47 L) 0,29	E) 0,18 S) 0,09 L) 0,10	E) 1,59 S) 1,35 L) 1,01	E) 0,27 S) 0,01 L) 0,02	E) 0,21 S) 0,14 L) 0,22

E – metoda ekstremów, S – metoda środków, L – metoda próbkowania.

Źródło: obliczenia własne z wykorzystaniem procedur `mlptrain` i `mlp` z pakietu `neural` dla **R**.

Wyniki uzyskane za pomocą metod zamiany przedziałowych zmiennych symbolicznych w zmienne klasyczne oceniono, biorąc pod uwagę wszystkie modele i symulacje jednocześnie. Najlepsza z metod to ta, dla której MSE (błąd średniokwadratowy) jest najniższy. Wyniki dotyczące wartości MSE dla poszczególnych modeli i symulacji zawarto w tab. 2.

6. Wnioski

Za najlepszą metodę zamiany przedziałowych zmiennych symbolicznych na klasyczne na użytek perceptronu wielowarstwowego w ujęciu wszystkich modeli i

symulacji należy uznać metodę próbkowania. Kolejne miejsce zajmuje metoda ekstremów, natomiast najgorsze wyniki przyniosła metoda średnich.

Perceptron wielowarstwowy pozwala dość trafnie klasyfikować dane symboliczne o wyraźnej i niezbyt dobrze rozróżnialnej strukturze klas bez zmiennych zakłócających i obiektów odstających. Równie dobrze rozróżnia on klasy, gdy w zbiorze zmiennych znajdują się zmienne zakłócające lub gdy mamy do czynienia z obserwacjami odstającymi. Niestety, gdy mamy do czynienia zarówno ze zmiennymi zakłócającymi, jak i z obserwacjami odstającymi, perceptron wielowarstwowy nie zawsze pozwala dokładnie wykryć przynależność obiektów do klas.

Dalszym kierunkiem badań nad perceptronem wielowarstwowym dla danych symbolicznych powinno stać się podejście wielomodelowe.

Literatura

Analysis of symbolic data. Explanatory methods for extracting statistical information from complex data (2000), red. H.-H. Bock, E. Diday, Springer-Verlag, Berlin-Heidelberg.

Duda R.O., Hart P.E., Stork D.G. (2001), *Pattern Classification*, John Wiley and Sons, Indianapolis.

Gatnar E. (2008), *Podejście wielomodelowe w zagadnieniach dyskryminacji i regresji*, Wydawnictwo Naukowe PWN, Warszawa.

Rossi F., Conan-Guez B. (2002), *Multi-layer perceptron on interval data*, [w:] *Classification, clustering and data analysis*, red. K. Jajuga, A. Sokołowski, H.-H. Bock, Springer-Verlag, Berlin-Heidelberg, s. 427-436.

NEURAL NETWORKS FOR SYMBOLIC DATA: MULTILAYER PERCEPTRON

Summary

The aim of this paper is to present and compare methods that allow to classify symbolic data with application of multilayer perceptron. In the article, basic terms of neural networks and transformation methods of symbolic interval-valued variables are presented.

The article presents and compares in the empirical part simulation experiment results based on artificial data generated with application of `cluster.Gen` procedure from `cluster.Sim` package for **R** software.