Tadeusz Kwater, Paweł Krutys, Marek Bolanowski

Interfejs graficzny do badań identyfikacji bramek logicznych z zastosowaniem sztucznych sieci neuronowych

Edukacja - Technika - Informatyka nr 3(13), 333-338

2015

Tadeusz KWATER, Paweł KRUTYS
Uniwersytet Rzeszowski, Polska

Marek BOLANOWSKI
Politechnika Rzeszowska, Polska

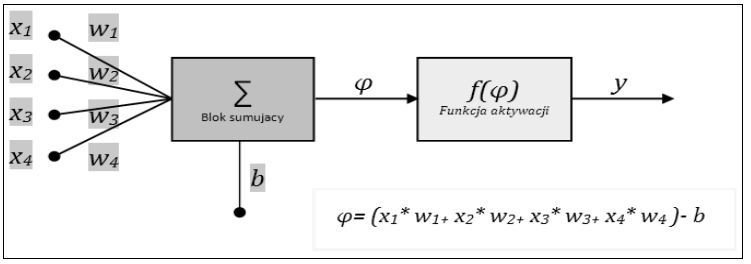
Interfejs graficzny do badań identyfikacji bramek logicznych z zastosowaniem sztucznych sieci neuronowych

Wstęp

Elementy sztucznej inteligencji w dobie nowoczesnej techniki stanowią przydatne narzędzie w wielu zastosowaniach. W celu zwiększenia przystępności i łatwości obsługi takich zaawansowanych narzędzi stosuje się dedykowane interfejsy. W artykule przedstawiono opracowanie interfejsu graficznego z aplikacją wykorzystującą sztuczne sieci neuronowe. W szczególności zaprojektowano główne elementy interfejsu z możliwością dokonywania zmian wartości parametrów oraz możliwością wyboru architektury sztucznych sieci neuronowych do aproksymacji funkcji logicznych, a także przeprowadzenia procesu uczenia sztucznych sieci neuronowych i badań symulacyjnych. Cecha charakterystyczna tego interfejsu to prostota obsługi i łatwość zastosowania sztucznej inteligencji do aproksymacji różnych funkcji logicznych.

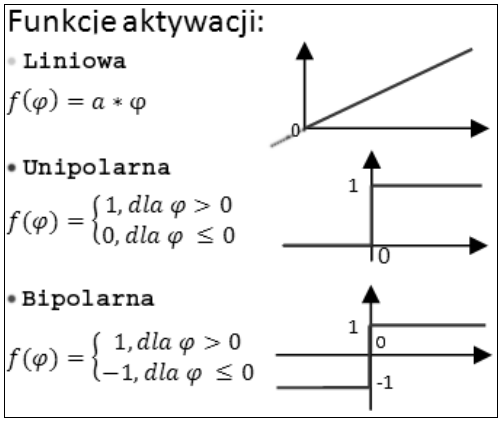
Matematyczny model neuronu

Sztuczny neuron można rozpatrywać jako specyficzny przetwornik sygnałów działający według następującej zasady. Na wejście przetwornika podawane są sygnały wejściowe, które następnie są mnożone przez odpowiednie współczynniki wag (odpowiadające „sile" połączeń synaptycznych w biologicznym neuronie). „Ważone" sygnały wejściowe są następnie sumowane i na tej podstawie wyznacza się aktywność neuronu [*Wprowadzenie do sztucznych...*].



Rys. 1. Schemat sztucznego neuronu

Model sztucznego neuronu składa się z bloku sumowania ∑ i bloku aktywacji *f(φ)*. W pewnym przybliżeniu blok sumowania odpowiada biologicznemu ciału komórki, w której realizowane jest algebraiczne sumowanie ważonych sygnałów wejściowych oraz generowany jest sygnał wyjściowy, który może być traktowany jako potencjał membranowy komórki.



Rys. 2. Przykładowe funkcje aktywacji

Sygnał φ poddawany jest przetwarzaniu przez blok aktywacji *f(φ)*, który w zależności od potrzeb może być opisany różnymi funkcjami [Walkowiak 2006].

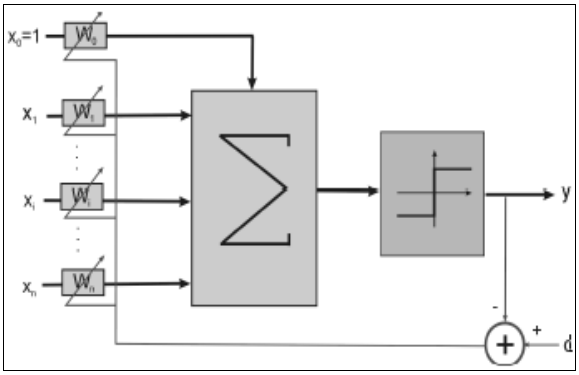
Proces uczenia

Uczenie polega na automatycznym dobraniu takich wartości wag sygnałów wejściowych, przy których sieć będzie generować sygnały zgodne z podanym wzorcem. Jedną z najprostszych metod uczenia neuronu jest modyfikacja wektora wag o ustaloną wartość proporcjonalną do popełnionego błędu:

$$\Delta w_i(t) = \eta \delta_\mu(t) x^\mu, \qquad \delta_\mu(t) = d_\mu - y_\mu(t), \tag{1}$$

gdzie: t – numer kroku algorytmu, η – współczynnik uczenia, $d\mu$ – żądana odpowiedź (wzorzec) ma μ-ty wektor wejściowy x_μ, y_μ – aktualna odpowiedź na μ-ty wektor wejściowy x_μ, δ – błąd wynikający z dobranych wag.

Proces ucznia odbywa się iteracyjnie. Na wejściu pobierane są dane. Następnie w bloku sumującym odbywa się sumowanie. Kolejnym etapem jest przejście przez wybraną funkcję aktywacji. Ostatni etap to sprawdzenie odpowiedzi układu z wzorcem. Zależnie od wyniku tego testu proces uczenia się kończy lub przechodzi do samego początku z odpowiednią korektą wag. Pętla powtarza cały algorytm *n* razy. Wartość *n* to liczba epok uczenia danego neuronu. Przy niewłaściwym doborze liczby epok uczenia sieć może utknąć i się nie nauczyć.



Rozróżniamy dwa rodzaje uczenia neuronu:
nadzorowane:
– reguła Delta,
– reguła perceptronowa,
– reguła Widrowa-Hoffa,

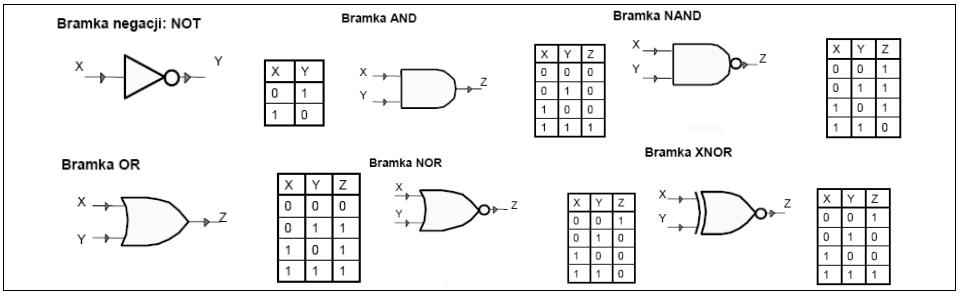
nienadzorowane
– reguła Hebba,
– uczenie typu konkurencyj-

Rys. 3. Model procesu uczenia neuronu

Uczenie nadzorowane stosuje się tylko wówczas, gdy istnieje możliwość zweryfikowania poprawności odpowiedzi udzielanych przez sieć. Oznacza to, że dla każdego wektora wejściowego musi być znana dokładna postać wektora wyjściowego (pożądana odpowiedź). Uczenie nienadzorowane stosuje się wówczas, gdy nie znamy oczekiwanych odpowiedzi na zadany wzorzec [Tadeusiewicz 1993].

Funkcje logiczne

Teoretyczną podstawę techniki cyfrowej stanowi algebra Boole'a. Jest to dział matematyki zajmujący się działaniami, dla których zarówno argumenty, jak i wyniki mogą przybierać tylko dwie wartości: 0 lub 1.

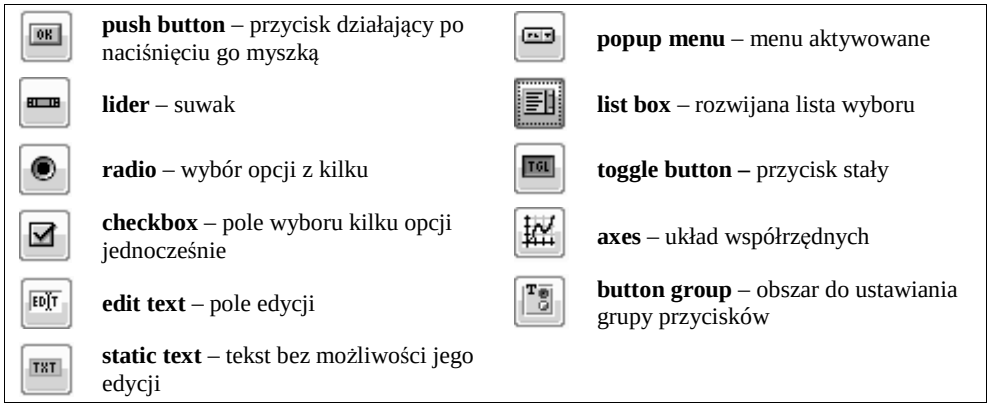


Rys. 4. Symbole bramek oraz tablice prawdy

Teoria i funkcjonowanie bramek logicznych stanowi postawę do konstruowania bardziej złożonych układów logicznych [Kręciejewski 1988]. Bramki te reprezentują prawa de Morgana i wchodzą w skład algebry Boole'a stosowanej w projektowaniu układów logicznych i sterowaniu z wykorzystaniem teorii zbiorów rozmytych.

Interfejs graficzny i badania symulacyjne

MATLAB jest interakcyjnym środowiskiem do wykonywania obliczeń naukowych i inżynierskich. Umożliwia testowanie algorytmów, modelowanie i symulację, analizę oraz wizualizację danych, sygnałów, a także wyników obliczeń. Jest wykorzystywany w różnych dziedzinach nauki związanych z techniką. Tworzenie interfejsu użytkownika podzielono na kilka etapów. Do najważniejszych należą: projektowanie, ustawianie elementów, właściwości elementów, programowanie, testowanie. Realizację interfejsu użytkownika rozpoczęto od konfiguracji narzędzia Guide, dokonując odpowiedniego wyboru przycisków akcji.



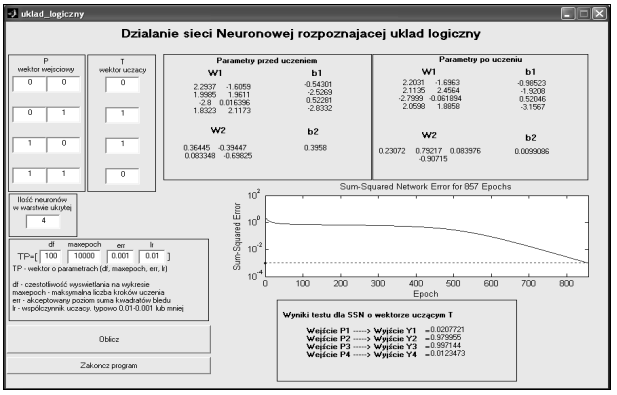
Rys. 5. Przyciski okna GUI

Każdy z obiektów graficznych (przycisków) interfejsu ma szereg ustawień dodatkowych. Aby z nich skorzystać, należy wybrać z menu podręcznego odpowiednie właściwości za pomocą Property Inspector, zaznaczając dany element [*Środowisko graficzne*...].

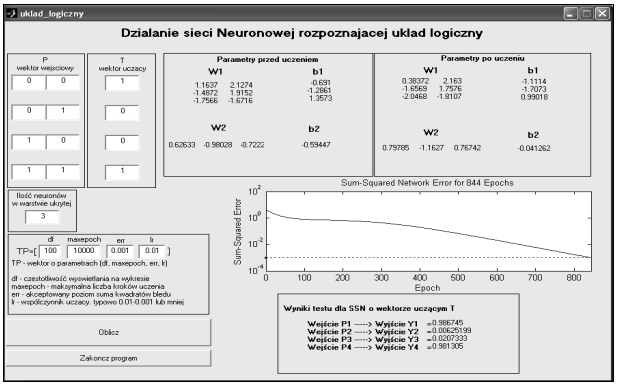
Po zaprojektowaniu interfejsu użytkownika zostają automatycznie wygenerowane dwa pliki:

– plik *.fig zawierający rozmieszczenie poszczególnych elementów,
– plik *.m zawierający kod programu.

W pliku zawierającym kod programy będą umieszczone wszystkie funkcje zapewniające obsługę wszystkich zdarzeń związanych z wywoływaniem interfejsu. Programiście pozostaje teraz zdefiniować wszystkie funkcje związane z interakcją użytkownika z projektowanym interfejsem i programem. Aby utworzyć taki lub podobny program, pracę rozpoczynamy od uruchomienia nakładki Guide. W nowym oknie przeciągamy kontrolki obiektów, które będą nam potrzebne, i umieszczamy je zgodnie z wytycznymi. Widok przykładowych eksperymentów przedstawiono na rys. 6–8.

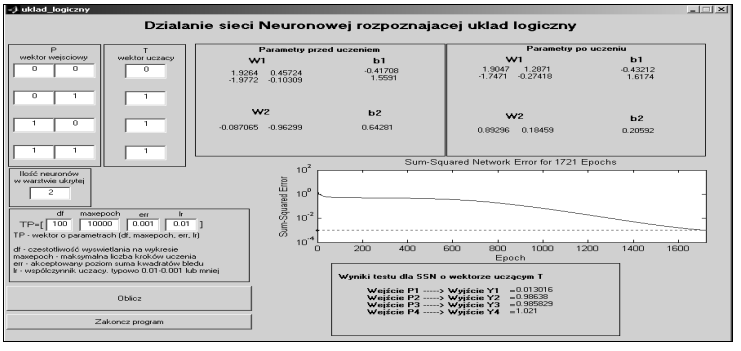


Rys. 6. Okno aplikacji dla bramki XOR



Rys. 7. Okno aplikacji dla bramki NXOR

W interfejsie zaprezentowano przebieg symulacji dla badanej funkcji. Obliczenia wszystkich parametrów zostają przedstawione w formie wykresu procesu uczenia oraz wyświetlanej odpowiedzi.



Rys. 8. Okno aplikacji dla bramki OR

Działanie programu to zainicjowanie sieci neuronowej potrzebnej do symulacji, nauczenie tej sieci oraz testowanie. Istnieje możliwość ustalenia architektury oraz parametrów samego uczenia. W tym celu tworzymy wektor 4-elementowy, w którym ustala się częstotliwość wyświetlania wyników na wykresie, maksymalną liczbę epok uczenia, akceptowalny poziom sumy kwadratów błędu oraz współczynnik uczący.

Podsumowanie

Sieci neuronowe znajdują coraz to szersze zastosowania w różnych dziedzinach życia. Zaprojektowane rozwiązanie ma na celu łatwiejsze zapoznanie się ze sztucznymi sieciami neuronowymi i pokazanie w prosty sposób działania oraz efektów pracy sieci neuronowych. Rezultat końcowy badań to stworzenie programu (inerfejsu), który wykorzystując SSN, będzie potrafił rozpoznawać funkcje logiczne. Program po wprowadzaniu danych wejściowych realizuje uczenie sieci oraz sprawdza, czy sieć została poprawnie nauczona, po czym wyświetli wynik końcowy testu uczenia.

Literatura

Kręciejewski M. (1988): *Układy cyfrowe*, Warszawa.
Środowisko graficzne Guide, http://www.ftj.agh.edu.pl/~stegowski/gui.
Tadeusiewicz R. (1993): *Sieci neuronowe*, Warszawa.
Walkowiak T. (2006): *Sieci neuronowe. Wprowadzenie*, Wrocław.
Wprowadzenie do sztucznych sieci neuronowych, http://aragorn.pb.bialystok.pl/~gkret/SNN.

Streszczenie

Zaprojektowany i wykonany interfejs graficzny do zastosowania w sztucznych sieciach neuronowych pozwala na lepsze zrozumienie zagadnień sztucznej inteligencji oraz przyszłych innych zastosowań. Dalsze prace mogą być prowadzone jako rozwiniecie tej idei.

Słowa kluczowe: bramki logiczne, sieci nauronowe, interfejs.

Graphic Interface to Research Identification Logic Gates by Using Artificial Neural Networks

Abstract

Designed and manufactured graphical interface to be used in artificial neural networks allows for a better understanding of the issues of artificial intelligence and the future of other applications. Further work can be carried out as you develop this idea.

Keywords: logic gates, naural network, interface.