

ZASTOSOWANIE SZTUCZNYCH SIECI NEURONOWYCH DO ROZPOZNAWANIA OBRAZÓW

Wojciech Dobrosielski

Uniwersytet Kazimierza Wielkiego
Instytut Techniki
ul. Chodkiewicza 30/p.215, 85-064 Bydgoszcz
e-mail: wdobroseilski@ukw.edu.pl

Streszczenie: *Praca zawiera informacje dotyczące sztucznej sieci neuronowej. Przedstawiony jest sposób działania wejść, wag, bloku sumarycznego, bloku aktywacji i wyjść, z których składa się sztuczny neuron. Omówiona jest struktura sieci z jakiego rodzaju funkcji aktywacji i ilości warstw się składa oraz metody procesu uczenia sieci neuronowej. Zaprezentowany model neuronu typu Adaline. Praca zawiera także informacje dotyczące obrazu cyfrowego, z czego się składa oraz modeli barw na drodze algebraicznej. W części badawczej zostało zaprezentowane w jaki sposób sieć zbudowana z neuronów typu Adaline rozpoznaje wzorce pisma grafiki rastrowej. Omówiony jest sposób przygotowania materiałów testujących oraz tworzenie zbioru uczącego i struktury sieci w zaprojektowanym programie. Przedstawiony jest proces uczenia sieci, jak również rodzaje przykładowych odpowiedzi jakich udziela sieć podczas testowania. Część badawcza zawiera eksperyment w którym ukazane są ważne czynniki jakie wpływają na poprawne działanie sieci.*

Słowa kluczowe: *Sztuczne sieci neuronowe, Adaline, Obraz cyfrowy*

Using artificial neural networks in image recognition

Abstarct: *The paper contains information on artificial neural network. Is shown how the inputs, weights, summary block, block the activation and output, which consists of an artificial neuron. In part of the research was presented on how the network is built with a Adaline neurons recognize patterns letters graphics.*

Keywords: *Artificial neural networks, Adaline, Digital image*

1. WSTĘP

Na podstawie obserwacji człowiek od wielu lat podejmuje próbę poznania ludzkiego mózgu. Prace prowadzone przez naukowców scharakteryzowały mózg jako zespół połączonych komórek, których zadaniem jest przetwarzanie informacji dostarczonych przez zmysły.

Porównanie ludzkiego mózgu z komputerem, algorytmem dostarczyło zróżnicowanych odpowiedzi. Klasyczny komputer okazał się jednostką szybszą, wydajniejszą. Jednak są takie zadania, w których mózg góruje

bezdyskusyjnie nad komputerem. Chodzi przede wszystkim o wszelakiego rodzaju rozpoznawanie i kojarzenie.

Dotychczasowa wiedza, którą człowiek zdobył obserwując mózg pozwoliła na wprowadzenie sztucznych sieci neuronowych. Sztuczne sieci neuronowe są wykorzystywane do wszelkiego rodzaju rozpoznawania, kojarzenia, przewidywania i sterowania. Różniącą sztuczne sieci neuronowe od klasycznego algorytmu komputerowego jest sposób przetwarzania danych.

Algorytm potrzebuje zgromadzenia i bieżącego dostępu do całej wiedzy na temat zagadnienia. Sztuczne sieci neuronowe wymagają jednorazowego przyswojenia wiedzy, która może być niekompletna a i tak otrzymamy trafną

odpowiedź, co więcej może zauważyć takie rozwiązania, o których programista nawet nie przypuszczał.

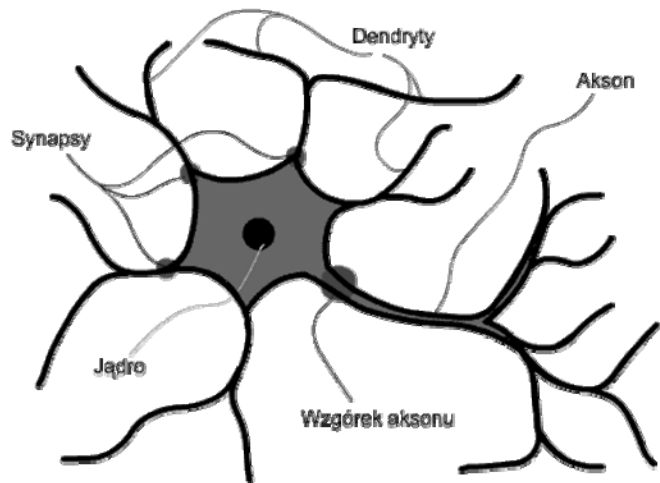
Percepcja człowieka umożliwia mu odbieranie i analizowanie skomplikowanych obrazów z otaczającego go świata. Obraz daje człowiekowi najwięcej informacji o otoczeniu. Z kolei mózg błyskawicznie przetwarza informacje przedstawione w postaci obrazu. Przetwarzanie informacji dzieje się tak szybko i jest to tak naturalne, że nie widać w tym nic dziwnego. Po prostu rozpoznawane są twarze, miejsca czy litery. Jednak tych wszystkich kształtów mózg wcześniej się nauczył, aby później nie określił ich za nieznane.

Człowiek potrafi uogólnić zdobytą wiedzę. W podobny sposób działają sztuczne sieci neuronowe, które uogólniają wiedzę dla nowych danych nieznanymi wcześniej, czyli nie prezentowanych w trakcie uczenia. W taki o to sposób sieć po nauczaniu się np. koloru zielonego czy niebieskiego potrafi rozpoznać również ciemno zielony i granatowy czy blade zielony i błękitny. Można określić taką umiejętność także jako zdolność do aproksymacji wartości funkcji wielu zmiennych.

2. SZTUCZNE SIECI NEURONOWE

Implementacja sztucznych sieci neuronowych w oprogramowaniu nadała inny wymiar relacji człowieka – maszyna. Oprogramowanie umożliwiło rozwiązanie dużego zakresu problemów, które dotychczas były zbyt abstrakcyjne. Ze względu na specyficzne cechy i niepodważalne zalety, obszar zastosowań sieci neuronowych jest rozległy. Głównym obszarem wykorzystania sztucznych sieci neuronowych są zagadnienia związane z rozpoznawaniem i klasyfikacją wzorców.

Inspiracją do budowy sztucznych sieci neuronowych stał się ludzki mózg, elementarną jednostką sztucznego mózgu jest neuron. Jego schemat został opracowany przez McCullocha i Pittsa w 1943 roku i oparty jest na budowie biologicznej komórki nerwowej. Schemat takiego neuronu przedstawiony jest na rysunku 2.1.



Rysunek. 2.1 Schemat neuronu biologicznego [1]

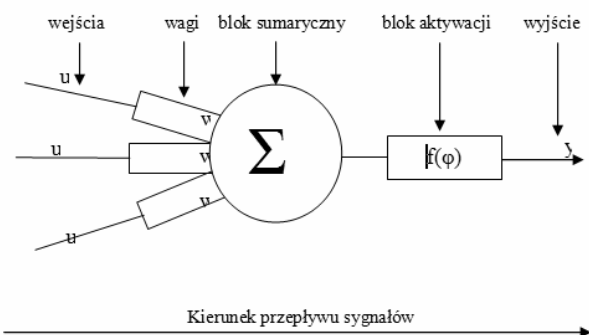
Mózg składa się z około 10^{11} neuronów wielu różnych typów. Każdy neuron może przysyłać i przetwarzać złożone sygnały elektrochemiczne. Neuron składa się z ciała komórki, w którym znajduje się jądro oraz otaczające je dwa rodzaje wypustek:

- Dendryty – posiadają rozgałęzioną strukturę i przenoszą sygnały otrzymane od innych neuronów przez synapsy do ciała komórki.

- Akson – jest pojedynczym, długim włóknem, które rozgałęzia się na mniejsze wypustki, wyprowadzając informacje do kolejnych neuronów.

Funkcję przekaźników informacji pomiędzy aksonem przez dendryty do ciała komórki, spełniają złącza nerwowe zwane synapsami. Transmisja sygnału odbywa się na drodze skomplikowanych procesów chemiczno – elektrycznych, podczas którego są uwalniane specyficzne substancje (tzw. transmitery). W rezultacie rośnie lub maleje potencjał elektryczny ciała komórki. Jeżeli potencjał osiągnie wartość progową, to sygnał na wyjściu neuronu jest przesyłany poprzez akson do innych neuronów.

Oczywiście w biologicznym mózgu występuje szereg bardziej złożonych mechanizmów przetwarzania informacji. Interesujące jest jednak to, że sztuczne sieci neuronowe mogą osiągnąć bardzo znaczące rezultaty praktyczne, korzystając z niesłychanie uproszczonego modelu, czyli przetwornika sygnału, który ukazany jest na rysunku 2.2.



Rysunek. 2.2 Model sztucznego neuronu [2]

„Na wejściu przetwornika podawane są sygnały wejściowe, następnie mnożone przez odpowiednie współczynniki wag (współczynniki te odpowiadają „sile” połączeń synaptycznych w biologicznym neuronie). „Ważone” sygnały wejściowe są następnie sumowane i na tej podstawie wyznacza się aktywność neuronu.”[3] Na rysunku 2.2 przedstawiono model sztucznego neuronu składający się z dwóch bloków: sumowania i aktywacji.

Blok sumowania odpowiada biologicznemu ciału komórki, w której wykonywane jest algebraiczne sumowanie ważonych sygnałów wejściowych, natomiast generowany sygnał wyjściowy φ może być traktowany jako potencjał membranowy komórki. Potencjał φ możemy wyznaczyć według wzoru:

$$\varphi = \sum_{i=1}^m w_i * u_i = w^T * u \quad (2.1)$$

gdzie:

- w – wektor współczynników wag;
- u – wektor sygnałów wejściowych;
- (.)^T – operator transponowania wektora lub macierzy;
- m – liczba wejść neuronu.

Sygnał φ poddawany jest przetwarzaniu przez blok aktywacji F, który w zależności od potrzeb może być opisany różnymi funkcjami.

Sposób działania sieci neuronowej polega głównie na zadawaniu pytania i otrzymywaniu odpowiedzi. Owe pytanie to odpowiednio przetworzone dane (np. obraz, dźwięk, dane giełdowe, warunki pogodowe) podawane na wejściu sieci. Po otrzymaniu pytania sieć generuje odpowiedź w postaci pewnej danej, które projektant sieci potrafi przełożyć na konkretną informację. Na wejście trafiają sygnały w postaci liczb opisujące zadanie, które neuron ma rozwiązać. Są również wagi posiadające pewne wartości. Sygnały wejściowe są mnożone przez wartości wag. Wyniki owej operacji są dodawane do siebie w bloku

sumarycznym. W ten sposób powstaje konkretna liczba, którą określa się jako potencjał membranowy. Jest ona wysyłana do bloku aktywacji, gdzie może zostać dodatkowo przetworzona. Po ewentualnej operacji wykonanej w bloku aktywacji otrzymujemy gotową odpowiedź neuronu na sygnały wejściowe. Bloku aktywacji jest pewną funkcją, której argumentem jest potencjał membranowy. Innymi słowy: „jeśli potencjał membranowy oznaczymy jako φ , odpowiedź, czyli wyjście neuronu jako y, a funkcję aktywacji jako f, to operację wykonywaną w bloku aktywacji można zapisać tak:

$$y = f(\varphi) \quad (2.2)$$

„Zatem wartość owej funkcji, czyli odpowiedź neuronu ($f(\varphi)$), będzie miała zawsze tę samą wartość, co potencjał membranowy (φ). Neuron z taką funkcją aktywacji nazywany jest neuronem liniowym.”[4] Należy jeszcze wyjaśnić, do czego właściwie służą wagi.

Każde z wejść ma swoją wagę, czyli informację o tym, jak duże znaczenie dla neuronu ma sygnał podany na to wejście. Kiedy wartość wagi jest mała, co do wartości bezwzględnej to dane wejście jest mało ważne dla neuronu. Natomiast duża wartość dodatnia oznacza, iż dane wejście ma większe znaczenie „pozytywne”, zaś duża, co do wartości bezwzględnej wartość ujemna to znak, że wejście ma większe znaczenie „negatywne”.

Wybór struktury sieci neuronowej jest pierwszym problemem, przed którym staje projektant sieci. Zależy to od rodzaju zadania, jaką ma sieć wykonywać. Jak już wiadomo wartość funkcji aktywacji jest sygnałem wyjściowym neuronu. Funkcja aktywacji może mieć postać liniową. Sieci, które wykorzystują funkcje liniową są najprostsze i nazywają się Madaline.[5] Sieci typu Madaline (ang. Many adaline) mimo swej prostoty mają duże znaczenie praktyczne zwłaszcza w budowie systemów rozpoznających, pamięci asocjacyjnych, filtrów adaptacyjnych, etc. Elementy tworzące daną sieć nazywają się Adaline.[6] Model neuronu typu Adaline (ADaptive LInear NEuron) opracowany został przez B. Widrowa.

Podstawą działania sztucznych sieci neuronowych jest ich uczenie. Wyróżnia się sieci samoorganizujące, sieci uczone są przez nauczyciela oraz sieci, w których proces uczenia przebiega z pomocą krytyka, każdy rodzaj sieci ma właściwy dla siebie algorytm uczenia i możliwe są jej zastosowania tylko dla określonych problemów.

Sieci samoorganizujące oparte są na algorytmie uczenia bez nadzoru lub bez nauczyciela. Jest to taka sieć bez informacji zwrotnej korygującej jej działanie z zewnątrz, sieć sama wypracowuje funkcje przetwarzania danych, np. uporządkowywania, klasyfikacji lub kodowania i innych. Nauka sieci samoorganizujących polega na konkurencji

neuronów. W konkurencyjnej metodzie uczenia sieci, w stanie aktywacji może się znajdować tylko jeden element wyjściowy. Nazywany jest on zwycięzcą, ponieważ schemat takiej reguły aktywacji neuronów określany jest mianem „zwycięzca bierze wszystko” (ang. Winner Takes All – WTA) [5].

Sieci uczone pod nadzorem nauczyciela stosowane są do problemów dobrze znanych przez projektanta sieci. Programista wie, czego chce sieć nauczyć i dysponuje wiedzą na temat tego, jaka ma być odpowiedź sieci na określone wymuszenie. W uczeniu pod nadzorem każdemu wektorowi wejściowemu

$$x(k) = (x_1(k), x_2(k), \dots, x_n(k))^T \quad (2.9)$$

odpowiada zadany wektor wyjściowy

$$d(k) = (d_1(k), d_2(k), \dots, d_n(k))^T. \quad (2.10)$$

Dane przeznaczone do nauczenia sieci są podawane w postaci par

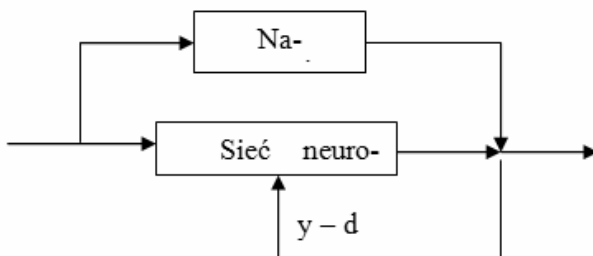
$$(x(k), d(k)) \text{ dla } k = 1, 2, \dots, p \quad (2.11)$$

gdzie:

p – oznacza liczbę wzorców uczących.

Uczenie z nauczycielem polega na takiej modyfikacji wag, aby jak najbardziej przybliżyć odpowiedzi sieci $y(k)$ na określone wymuszenie $x(k)$ do wartości żądanych $d(k)$. Realizowane jest to poprzez minimalizację odpowiednio zdefiniowanej funkcji celu. Najbardziej znanym przykładem algorytmu uczenia pod nadzorem jest algorytm wstecznej propagacji błędów.

Algorytm wstecznej propagacji błędu - BP (ang. BackPropagation) określa strategię doboru wag w sieci wielowarstwowej przy wykorzystaniu gradientowych metod optymalizacji. Podczas procesu uczenia sieci dokonuje się prezentacji pewnej ilości zestawów uczących. Uczenie polega na takim doborze wag neuronów by w efekcie końcowym błąd popełniany przez sieć był mniejszy od zadanego [7].



Rysunek. 2.3 Schemat sieci neuronowej z nauczycielem [9]

Uczenie z krytykiem stosowane jest w problemach, w których programista wie, do czego dąży, jest w stanie określić ostateczny wynik, jaką sieć ma osiągnąć. Jednak nie ma on konkretnych informacji, w jaki sposób ma postępować, aby otrzymać żadaną odpowiedź. Progra-mista może jedynie obserwować wynik poczynań sieci i odpowiednio korygować wartość wag, tak, aby uzyskać ostateczny cel. „Jeśli działania podjęte przez układ uczący dają wyniki pozytywny, to następuje wzmocnienie tendencji do właściwego zachowania się systemu w podobnych sytuacjach w przyszłości. W przeciwnym wypadku, jeśli wynik jest negatywny, następuje osłabienie tendencji takiego działania systemu.” [9]

3. MODEL ADALINE

Model neuronu typu Adaline (ang. ADAPtive LInear NEuron) jest najprostszym elementem sieci, która wykorzystuje funkcje liniową i nazywa się Madaline. Model neuronu tego typu został opracowany przez B. Widrowa. Budowa neuronu jest podobna do modelu perceptronu. Sposób wyznaczania sygnału wyjściowego jest identyczna jak przy perceptronie, a jedyna różnica dotyczy algorytmu uczenia. „Jednak w przypadku neuronu typu Adaline porównuje się sygnał wzorcowy d z sygnałem φ na wyjściu części liniowej neuronu. W ten oto sposób otrzymujemy błąd dany wzorem” [10]

$$\delta = d - \varphi \quad (2.17)$$

Wagi podczas uczenia neuronu, modyfikuje się zgodnie ze wzorem:

$$w_i(k+1) = w_i(t) + \eta \delta x_i. \quad (2.18)$$

W danej metodzie uczenia miarą błędu jest wynik minimalizacji kwadratowego kryterium błędu i określa się go ze wzoru:

$$Q(w) = \frac{1}{2} \delta^2 = \frac{1}{2} \left[d - \left(\sum_{i=0}^n w_i x_i \right) \right]^2. \quad (2.19)$$

Powyższa reguła nosi nazwę *reguły delta* i jest to najpopularniejsza metoda uczenia z nauczycielem. Reguła delta jest algorytmem działającym w sposób skumulowany, gdyż w każdej epoce, przy uwzględnieniu wszystkich przypadków uczących, wyznaczana jest średnia wartość gradientu błędu. Następnie w końcowej fazie epoki jednokrotnie modyfikowane są wagi sieci. Współczynnik uczenia dla każdej wagi jest indywidualny i zmienia się w każdej epoce.

Reguła delta posiada trzy parametry:

- „początkowy współczynnik uczenia, – który stosowany jest dla wszystkich wag w trakcie pierwszej epoki,
- współczynnik przyrostu, – który dodawany jest współczynnika uczenia w przypadku, gdy znak pochodnej nie ulega zmianie,
- współczynnik określający tempo zaniku, – przez który przemnażany jest współczynnik uczenia, jeśli pochodna zmienia znak.”[11]

Zastosowanie liniowego wzrostu oraz wykładniczego spadku współczynnika uczenia przyczynia się do zachowania stabilności metody uczenia neuronu typu Adaline.

4. OBRAZ CYFROWY

Obraz cyfrowy jest tworzony przez próbkowanie i kwantowanie sygnału wizyjnego i jego najmniejszym elementem jest piksel. Piksel ma przypisany poziom jasności różnych kolorów, który może być odmienny od elementu sąsiedniego. Jest kilka sposobów określania barw danym pikselom, np.:

- model RGB, który jest oparty na wiązce światła i składa się z R – czerwonego (ang. Red), G – zielonego (ang. Green) i B – niebieskiego (ang. Blue), mieszanych ze sobą w pewnych proporcjach;

- model HSB, który również oparty jest na wiązce światła i składa się z H – częstotliwości światła (ang. Hue), S – nasycenia koloru (ang. Saturation) i B – moc światła białego (ang. Brightness);

- model CMYK oparty jest na pigmentach, które odbijają światło i wykorzystywane jest w drukowaniu kolorowych obrazów. Składa się on z czterech podstawowych farb C – cyjan (ang. Cyan), M – magenta (ang. Magenta), Y – żółty (ang. Yellow) i K – czarny (ang. Black);

- model Lab, który jest nie zależny od światła i pigmentów a składa się z L – Luminacji (jasności pikseli), parametrów a – zakresie kolorów od zielonego do różowego i b – zakresie kolorów od niebieskiego do ugoru.

Obraz cyfrowy ma postać tablicy prostokątnej, która ma N wierszy i M kolumn a każdy jej element jest pikselem obrazu cyfrowego. Taką tablicę można zapisać jako macierz o wymiarach N na M elementów, co umożliwia wykonanie:

- działań matematycznych – dodawania, odejmowania, mnożenia, dzielenia, nakładania obrazów oraz operacji XOR;
- przekształceń geometrycznych – przesuwania, skalowania, obracania obrazu;

- transformacji dotyczącej klas – przekształceń klasy niższej w wyższą (segmentacja, binaryzacja, konturyzacja, szkieletonowanie), przekształcenia klasy wyższej w niższą (interporacja, aproksymacja, wypuklanie, cieniowanie, polepszanie, rekonstrukcja) oraz przekształcenia wewnątrzklasowe (kompresja, filtracja, obroty, przesunięcia obrazu).

W obróbce cyfrowej obrazu wykorzystuje się wiele różnych filtrów do poprawy jakości obrazu między innymi:

- splot, który jest bardzo prostą operacją i oblicza ona nowe wartości piksela na podstawie sąsiednich pikseli;
- filtr dolnoprzepustowy powoduje redukcję szumu jedno – albo dwupikselowy w obrazie;
- filtr górnoprzepustowy zwiększa ostrość obrazu;

- filtr uwypuklający i wykrywający krawędzie redukuje obraz jedynie do zawartych w nim krawędzi i tłumi elementy obrazu o niskiej częstotliwości;

- filtr medianowy usuwa zakłócenia losowe.

Filtry te mogą być wykorzystywane w transformacji Fouriera w postaci maski do poprawy jakości obrazu.

Wszystkie te procesy, działania i operacje są nieodzownym elementem obróbki cyfrowej, które są wykorzystywane w różnych programach graficznych przez zwykłych użytkowników. Graficy komputerowi wykorzystują różnego rodzaju przekształceń w programach graficznych do zaprezentowania prac z zakresu efektów wizualnych w obróbce cyfrowej obrazu, a programiści w wszelkiego rodzaju programach czy np. efektywnych grach komputerowych.

5. PRACA BADAWCZA

Praca badawcza polega na dobraniu odpowiedniej sztucznej sieci neuronowej do konkretnego problemu. Sieć będzie rozpoznawać wzorce pisma podanego w postaci grafiki rastrowej. Z rozdziału o sztucznych sieciach neuronowych wiadomo, iż sztuczny neuron składa się z wejść, na które podawane są pewne dane. W badaniu tym danymi podawanymi na wejście neuronu będą obrazy w postaci tablic rejestracyjnych, czyli sieć będzie przetwarzać poziom jasności pikseli w obrazie.

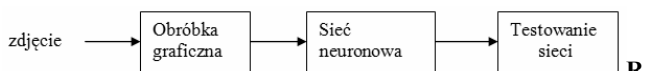
Chcąc zastosować sztuczną sieć należy określić, czego chcemy ją nauczyć. W pracy tej sieć będzie nauczona rozpoznawania liter i liczb podanych w postaci obrazu. Problem rozpoznawania liter i liczb w formie obrazu jest

stosunkowo prostym zadaniem do wykonania dla sieci, dlatego w badaniu tym zostanie wykorzystany model neuronu typu Adaline.

Proces rozpoznawania przez sieć neuronową dzieli się na trzy fazy:

1. Obróbka graficzna, w której przygotowuje się materiały podawane na wejście sieci.
2. Sieć neuronowa, w której tworzy się zbiór uczący i przeprowadza proces uczenia.

Testowanie, w którym sprawdza się czy sieć nauczyła się rozpoznawania wzorców liter i liczb podawanych w formie obrazu.



rysunek. 5.1 Schemat blokowy procesu rozpoznawania obrazu. [źródło własne]

Wykorzystywany w sieci neuron Adaline jest uproszczonym modelem biologicznego neuronu, posiada wiele wejść i jedno bądź kilka wyjść oraz działa jednokierunkowo. Sieć zbudowana z neuronów typu Adaline uczy się pod nadzorem nauczyciela, ponieważ w pracy tej określenie odpowiedzi na dane wymuszenie nie stanowi większego problemu.

Zadaniem sieci neuronowej jest rozpoznawanie. Rodzaj danych wejściowych został określony, a jest to obraz, który przedstawia wzorce liter i liczb w postaci tablic rejestracyjnych. Chcąc nauczyć sieć rozpoznawania należy najpierw przygotować materiały do przeprowadzenia procesu uczenia. Jak wiadomo z rozdziału o sieciach neuronowych, dane wykorzystywane w procesie uczenia z nauczycielem podawane są w parach, dlatego w pracy tej wektorem wejściowym jest plik w formacie obrazu wyglądający następująco:

Rysunek. 5.2 Wygląd tablicy rejestracyjnej podawanej na wejście sieci. [źródło własne]

natomiast wektorem wyjściowym jest nazwa pliku zawierająca znaki w grafice rastrowej takie jak:

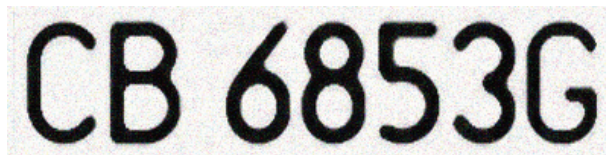
CB6853G.bmp (4.1)

Rysunek 5.1 przedstawia obrazek, który jest podawany w formie pliku na wejście sieci. Jednak, aby taki rysunek osiągnąć trzeba wykonać obróbkę rysunków tablic rejestracyjnych.

Jest wiele sposobów obróbki rysunków, jednak w pracy tej został wykorzystany pro-gram Adobe Photoshop, który służy do obróbki grafiki rastrowej. Przygotowanie materiałów na wejście sieci w tym programie wygląda następująco:

1. Ze zdjęcia, które zawiera tablice rejestracyjną, zostaje zaznaczona tablica. Zaznaczony fragment rysunku zawiera tylko wzorce liter i liczb, które są istotne. Zaznaczenie zostaje kopiowane do nowego pliku w celu pozbycia się niepotrzebnych danych.

2. W nowym dokumencie ustawia się dokładne wymiary rysunku 400 x 100 pikseli. Dokładny wymiar stanowi tu istotną funkcję, ponieważ parametry wysokości i szerokości obrazka będą potrzebne później do projektowania sieci neuronowej.



Rysunek. 5.3. Tablica rejestracyjna w nowym dokumencie. [źródło własne]

3. Jak wiadomo z rozdziału o obrazie cyfrowym rysunek składa się z pikseli, który tworzy macierz. Każdy piksel ma przypisany kolor i poziom jasności, gdzie parametry te są określane przez różne modele barw. Obraz, który jest wykorzystywany do tego badania występuje w modelu RGB. Ten model barw zawiera, aż 16 milionów różnych kolorów. Taka ilość kolorów może sprawić pewne utrudnienie podczas projektowania sieci neuronowej, dlatego została przeprowadzona segmentacja. Powoduje ona, iż obraz stał się czarno - biały i zawiera kolory w skali szarości.

4. Obraz tablicy rejestracyjnej posiada jeszcze pewne zakłócenia w formie kurzu i szumu. Chcąc poprawić ostrość obrazu trzeba przeprowadzić filtrację. Najlepiej do takich zadań służy filtr medianowy, którego dokładne działanie zostało opisane w rozdziale o obróbce cyfrowej.



Rysunek. 5.4 Obrazek po zastosowaniu filtru medianowego. [źródło własne]

5. Obraz zostaje poddany jeszcze jednej operacji. Spowoduje on, iż liczby i litery staną się wyraźniejsze. Do tego służy operacja dodawania skalar powodująca wyostrzenie zarysu danych.



Rysunek. 5.5 Obrazek po operacji dodawania. [źródło własne]

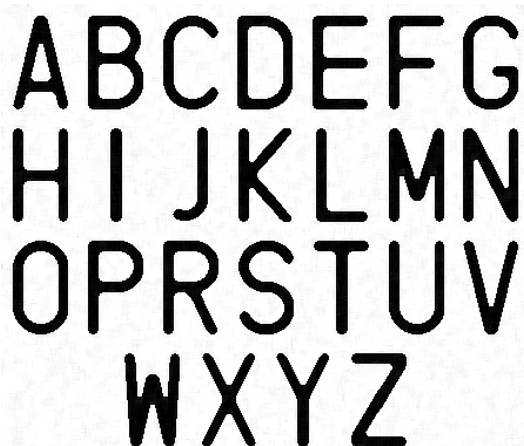
6. Tak przygotowany rysunek zostaje poddany jeszcze jednej transformacji, którą jest binaryzacja. Polega ona na zamianie odcieni szarości na dwa kolory, czarny i biały. W Photoshopie proces ten wykonuje się przez funkcję Indeksowania. BINARYZACJA spowoduje, że sieć będzie w jeszcze szybszy sposób przetwarzać dane.



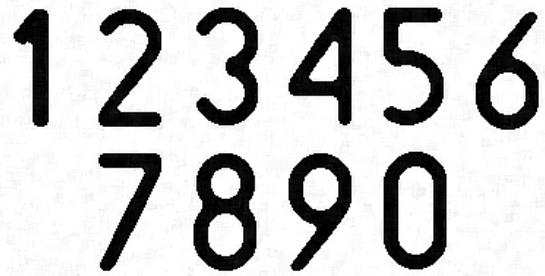
Rysunek. 5.6 Efekt końcowy po obróbce obrazka. [źródło własne]

Pozostaje jeszcze zapisanie obrazka pod odpowiednią nazwą i w formacie mapy bitowej. Sieć neuronowa zaprojektowana w programie Borland C++ Builder bez żadnych komplikacji potrafi przetworzyć informacje z mapy bitowej.

Sieć neuronowa ma rozpoznawać następujące znaki:



Rysunek. 5.7 Litery wykorzystywane w tablicach rejestracyjnych. [źródło własne]

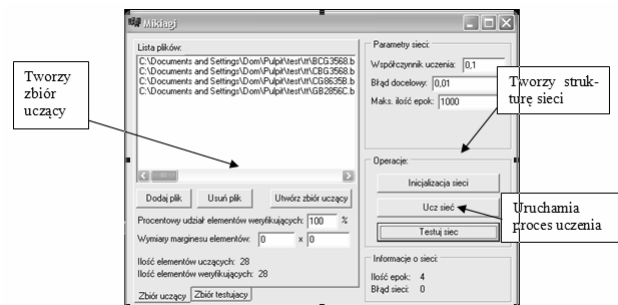


Rysunek. 5.8 Cyfry wykorzystywane w tablicach rejestracyjnych [źródło własne]

Ponieważ tablice rejestracyjne zawierają tylko siedem znaków, dlatego trzeba przygotować ich tyle, aby zawierały kombinacje wszystkich liter i cyfr.

6. OPROGRAMOWANIE

Po przygotowaniu materiałów podawanych na wejście sieci, można przejść do programu, który będzie rozpoznawać znaki w postaci tablic rejestracyjnych.



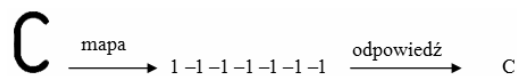
Rysunek. 6.1 Panel sterowania sieci do tworzenia zbioru uczącego [źródło własne]

Pobierane są pliki do programu, które zawierają tablice rejestracyjne, przyciskiem Dodaj plik. Program wykonuje filtrację plików tak, aby można było wybierać tylko pliki w formacie mapy bitowej. Wykonuje również sprawdzenie czy wybrany plik już istnieje na liście i uniemożliwia ponowne wpisanie. Sprawdzenie to powoduje, że do zbioru uczącego są podawane obrazki, które się nie powtarzają. Przy braku powtórzeń sieć na pewno uczy się sprawiedliwiej, gdyż występowanie znaków jest bardziej zrównoważone i nie ma faworyzowania danej tablicy. Przy powtarzaniu dany zbiór powodowałby szybsze nauczenie zestawu znaków, który zawiera, ale reszta pozostawałaby i tak do nauczenia, i w przypadku, gdyby tych znaków nie

było w zbiorze weryfikującym, to uczenie mogłoby być nawet dłuższe.

Chcąc przeprowadzić proces uczenia trzeba najpierw utworzyć zbiór uczący. Do tego służy przycisk Utwórz zbiór uczący widoczny na rysunku 6.1. Po wybraniu tego przycisku program sprawdza listę plików, czy znajdują się na niej obrazki, oraz czy te pliki zawierają 8-mio bitową głęboką kolorów. Następnie dzieli obrazek tablicy rejestracyjnej na poszczególne znaki, przez co określana jest liczba elementów uczących sieć neuronową.

Odpowiedzi sieci, które podawane w formie nazwy pliku są odpowiednio dzielone i wstawiane do tabeli odpowiedzi. Z tabel elementów i odpowiedzi jest tworzona mapa odpowiedzi, z której sieć neuronowa będzie korzystała podczas uczenia się. Mapa odpowiedzi jest identyfikacją danego elementu do konkretnej odpowiedzi np.:



Rysunek 6.2 Identyfikacja znaku „C” [źródło własne]

Po utworzeniu zbioru uczącego można przejść do utworzenia sieci neuronowej. Tworzenie sieci wykonywane jest przez przycisk Inicjalizacja sieci widoczny na rysunku 6.1. Przycisk ten tworzy strukturę sieci, a jest ona zbudowana z:

- warstwy wejściowej – ilość neuronów w warstwie zależna jest od największej długości elementu ze zbioru uczącego.
- warstwy wyjściowej – ilość neuronów w warstwie zależna jest od ilości odpowiedzi określające dane elementy.

Sieć jest tak zbudowana, że wszystkie wyjścia z warstwy wejściowej połączone są z wszystkimi wejściami warstwy wyjściowej tak jak jest to widoczne na rysunku 6.3. W warstwie wyjściowej neurony posiadają skokową funkcję aktywacji. Funkcja ta przyjmuje wartości 1 oraz -1 i jest bipolarną funkcją aktywacji sieci neuronowej.

W programie jest możliwość podania parametru współczynnika uczenia przez użytkownika i przyczynia się on do zachowania stabilności metody uczenia sieci. Współczynnik ten powinien zawierać się w przedziale (0,1] i stosowany jest dla wszystkich wag w trakcie pierwszej epoki.

Tworzenie struktury sieci w programie odbywa się przez utworzenie obiektu klasy „KlasaSiec” i przypisanie go pod wskaźnik o nazwie „siec”. Konstruktor pobiera cztery dane, którymi są:

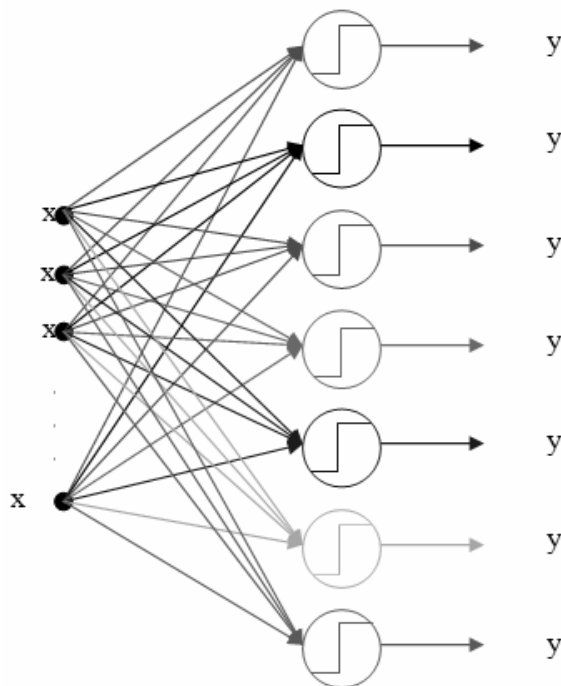
- ilość wejść sieci, która jest równa długości elementów, wektorów wejściowych w zbiorze uczącym.

Zwracana jest ona przez metodę DlugoscElementu() obiektu zbiorUczacy.

- ilość wyjść sieci, która jest równa ilości unikalnych odpowiedzi dla wszystkich par uczących. Zakładana jest sygnalizacja wystąpienia danego znaku, będącej odpowiedzią, poprzez pojawieniu się stanu aktywnego dla neuronu, który odpowiada temu znakowi, oraz nieaktywności pozostałych neuronów. Ilość zwracana jest po-przez wywołanie metody loscOdpowiedzi() obiektu zbiorUczacy.

- wskaźnik do funkcji aktywacji używanej przez neurony warstwy wyjściowej, która została wcześniej odpowiednio zdefiniowana.

- współczynnik uczenia w postaci liczby rzeczywistej z przedziału (0,1] uzyskanej poprzez konwersję z postaci tekstowej przy pomocy funkcji StrToFloatDef, ciągu znaków zawartego w odpowiednim oknie umieszczonym na formacie będącym graficzną reprezentacją komponentu Współczynnik Uczenia.



Rysunek 6.3 Budowa wykorzystanej sieci [źródło własne]

Kiedy materiały podawane na wejście sieci są przygotowane, to można przejść do uczenia sieci neuronowej. Jak wiadomo z rozdziału o sieciach neuronowych najpierw sieć należy nauczyć rozpoznawania. Proces uczenia widoczny na schemacie blokowym przebiega w następujący sposób:

1. Wagi początkowe są wybierane w sposób losowy.
2. Wprowadzany jest współczynnik uczenia η i zawiera się w przedziale $(0,1]$.
3. Ze zbioru uczącego n - elementowego losowanych jest m - wektorów zbioru weryfikującego, gdzie $m \leq n$.
4. Na wejście neuronu podawany jest wektor uczący x , zgodnie ze wzorem

$$x(k) = (x_1(k), x_2(k), \dots, x_n(k))^T \quad (4.4)$$

gdzie $k = 1, 2, \dots$

5. Wyznaczany jest sygnał $\varphi(k)$ ze wzoru

$$\varphi(k) = f \sum_{i=1}^N (x(i) * w(i, j)) + \theta \quad (4.5)$$

gdzie:

θ – bias wyrażone wzorem ,
 N – długość wektora wejściowego,
 w – macierz wag.

6. Zostaje wykonana modyfikacja wag według wzoru

$$w(k+1) = w(k) + \eta \delta x(k) \quad (4.6)$$

gdzie $\delta = d(k) - \varphi(k)$

7. Wracamy do punktu 4 procesu uczenia.

Algorytm powtarza się tak długo, aż dla wszystkich wektorów wejściowych wchodzących w skład ciągu weryfikującego błąd na wyjściu będzie mniejszy od założonej tolerancji.

7. EKSPERYMENT BADAWCZY

Eksperyment badawczy polega na porównaniu procentowej zawartości błędnych odpowiedzi popełnionych podczas testowania sieci, która została nauczona na dwóch różnych zbiorach uczących.

Pierwszy zbiór uczący przedstawiony jest w formie tablic rejestracyjnych, które zawierają kombinację wszystkich liter i liczb. Znaki te występują w różnego rodzaju zniekształceń.

Kolejny zbiór zawiera tablice rejestracyjne z uporządkowanymi znakami liter i liczb. W każdej pojedynczej tablicy znajduje się jeden idealny symbol oraz sześć elementów z pewnymi ubytkami.

Ilość tablic rejestracyjnych w jednym i drugim zbiorze uczącym jest 35 sztuk, w sumie jest 245 elementów uczących.

Zbiór testujący na którym będzie sprawdzana procentowa zawartość błędnych odpowiedzi składa się z 20 tablic rejestracyjnych, gdzie 70 elementów nie występuje w zbiorze uczącym i taka sama ilość elementów wystąpiła w zbiorze uczącym.

Podczas przeprowadzenia procesu uczenia będą się zmieniać takie parametry jak:

- wartość współczynnika uczenia,
- wartość błędu docelowego,
- procentowy udział elementów weryfikujących.

Eksperyment wykonano na zbiorze uczącym, który zawiera kombinację wszystkich liter i liczb w postaci tablic rejestracyjnych.

Poniższa tabele zawierają procentową zawartość błędnych odpowiedzi podczas testu na zbiorze testującym:

Tabela 7.1 Zmiana współczynnika uczenia.

Procentowy udział elementów weryfikacyjnych	Wartość błędu docelowego	Wartość współczynnika uczenia	Procentowa zawartość błędnych odpowiedzi
100 %	0,01	0,1	13 %
100 %	0,01	0,2	14 %
100 %	0,01	0,3	16 %
100 %	0,01	0,4	17 %
100 %	0,01	0,5	20 %
100 %	0,01	0,6	16 %
100 %	0,01	0,7	14 %
100 %	0,01	0,8	11 %
100 %	0,01	0,9	16 %
100 %	0,01	1	18 %

Jak widać w tabeli sieć, której wartość błędu zmierza do zera i liczba elementów weryfikujących zawiera 100 % zbioru uczącego, ilość popełnionych błędów zawarta w procentach najmniejsza jest, gdy współczynnik ma wartość 0,8.

Tabela 7.2 Zmiana wartości błędu docelowego.

Procentowy udział elementów weryfikacyjnych	Wartość błędu docelowego	Wartość współczynnika uczenia	Procentowa zawartość błędnych odpowiedzi
100 %	0,01	0,8	11 %
100 %	0,1	0,8	12 %
100 %	0,5	0,8	13 %
100 %	1	0,8	17 %
100 %	2	0,8	20 %
100 %	5	0,8	25 %
100 %	10	0,8	33 %

Sieć w której zwiększa się wartość błędu docelowego ma tendencje do coraz większej ilości popełniania błędnych odpowiedzi.

Podczas testowania sieci gdzie zbiór testujący składa się w połowie z elementów, które nie powtarzają się w zbiorze uczącym popełnia błędy w granicach 21%. Jednak na drugiej połowie liczbie elementów zbioru testującego, gdzie symbole wystąpiły w zbiorze uczącym sieć popełnia błąd w granicach 1,5%.

Tabela 7.3 Zmiana procentowego udziału elementów weryfikujących.

Procentowy udział elementów weryfikacyjnych	Wartość błędu docelowego	Wartość współczynnika uczenia	Procentowa zawartość błędnych odpowiedzi
100 %	0,01	0,8	11 %
75 %	0,01	0,8	14 %
50%	0,01	0,8	17 %
25 %	0,01	0,8	19 %
2 %	0,01	0,8	22 %
100 %	10	0,8	33 %
75 %	10	0,8	36 %
50 %	10	0,8	38 %
25 %	10	0,8	45 %
2 %	10	0,8	51 %

Tabela powyżej przedstawia sieć w której zmienia się procentowy udział elementów weryfikacyjnych przy dwóch rodzajach wartości błędów docelowych. Parametr ten określa liczbę elementów zbioru uczącego jaka jest wykorzystywana podczas procesu uczenia. W przypadku, gdy maleje liczba elementów weryfikacyjnych rośnie liczba popełnianych błędów dla sieci. Jednak gdy wartość błędu docelowego jest duża, potęguje to w ilości popełnianych błędnych odpowiedzi sieci dla takiej samej liczby elementów w zbiorze testującym.

8. WNIOSKI

Po przeprowadzonym eksperymencie można zauważyć, że ważny jest współczynnik uczenia sieci, który powoduje że sieć jest stabilniejsza i popełnia mniej błędów, oraz błąd docelowy do którego zmierza sieć w procesie uczenia.

Istotne jest również sposób przygotowania zbioru uczącego, ponieważ nie może zawierać elementów o podobnym wyglądzie a różnych odpowiedziach w procesie uczenia, gdyż sieć jest wówczas źle uczona co powoduje zwiększenie ilości popełnianych błędów.

Sieć neuronowa udziela mniej błędnych odpowiedzi kiedy zbiór uczący składa się z uporządkowanych symboli liter i liczb niż z tablic rejestracyjnych z przypadkowo ułożonych elementów

9. PODSUMOWANIE

Praca zawiera wiedzę dotyczącą sztucznych sieci neuronowych. Sposobu jej działania, struktury z jakiej może się składać oraz rodzajów procesów uczenia. Przedstawiona jest również wiedza dotycząca obrazu cyfrowego. Modeli kolorów z jakich się składa, filtrów jakim można poddać obraz, oraz przekształceń wykorzystywanych podczas różnych transformacji. Zaprezentowane są czynności jakim poddawany jest obraz w postaci tablicy rejestracyjne w procesie rozpoznawania przez sieć neuronową.

Przedstawiony został eksperyment na sieci, która zbudowana jest z dwóch warstw neuronowych, gdzie neurony są połączone na zasadzie „każdy z każdym” i bipolarnej funkcji aktywacji. Badanie wykazuje, iż ważnymi czynnikami na poprawne działanie sieci są:

- sposób przygotowania elementów do zbioru uczącego,
- parametr współczynnika uczenia,
- wartość błędu docelowego

w procesie uczenia sieci pod nadzorem nauczyciela.

Podczas testowania sieci można zauważyć, iż nie sprawia jej większego problemu w rozpoznawaniu symboli liter i liczb, gdy występują braki w zbiorze uczącym lub kiedy tablice rejestracyjne są przypadkowo zaburzone.

Sztuczna sieć o modelu neuronu Adaline pozwala skutecznie rozpoznawać wzorce pisma podanego w postaci grafiki rastrowej.

Kierunkiem dalszych badań tej pracy może być porównanie skuteczności rozpoznawania symboli liter i liczb z tablic rejestracyjnych przez inne sieci jak na przykład sieć Kohonena czy RBF.

Literatura

1. A. Stateczny, „Sztuczne sieci neuronowe w rozpoznawaniu obiektów morskich”, GTN, Gdynia, 2002, s.12.
2. A. Stateczny, „Sztuczne sieci neuronowe w rozpoznawaniu obiektów morskich”, GTN, Gdynia, 2002, s. 14.
3. A. Stateczny, „Sztuczne sieci neuronowe w rozpoznawaniu obiektów morskich”, GTN, Gdynia, 2002, s. 13.
4. dzieło cytowane B. Barkowski.
5. J. Żurada, M. Barski, W. Jędruch, „Sztuczne sieci neuronowe. Podstawy teorii i zastosowania”, Wydawnictwo PWN, Warszawa, 1996.
6. L. Rutkowski, „Metody i techniki sztucznej inteligencji”, Wydawnictwo PWN, Warszawa, 2005.

7. cytowane Copyright StatSoft, Inc., 1984-2003 „Sieci neuronowe. Algorytm wstecznej propagacji błędu”.
8. A. Stateczny, „Sztuczne sieci neuronowe w rozpoznawaniu obiektów morskich”, GTN, Gdynia, 2002, s.18.
9. Katedra Inżynierii Komputerowej Politechniki Częstochowskiej, „Sieci Neuronowe”, <http://www.kik.pcz.czest.pl/nn/uczenie.php?art=5>, 20.11.2006.
10. Katedra Inżynierii Komputerowej Politechniki Częstochowskiej, „Sieci Neuronowe”, <http://www.kik.pcz.czest.pl/nn/uczenie.php?art=5>, 20.11.2006. s. 167
11. dzieło cytowane Copyright StatSoft, Inc., 1984-2003 „Sieci neuronowe. Algorytm Delta-bar-Delta”.