

Studia i Materiały Informatyki Stosowanej

Studia i Materiały Informatyki Stosowanej

czasopismo młodych pracowników naukowych, doktorantów i studentów

patronat: Polskie Towarzystwo Informatyczne



Przewodniczący Rady Naukowej

prof. dr hab. inż. czł. koresp. PAN Janusz Kacprzyk, IBS PAN

Redaktorzy Naczelni

dr inż. Jacek Czerniak, UKW, IBS PAN
dr inż. Marek Macko, UKW

Sekretarz redakcji

dr Iwona Filipowicz, UKW

Komitet Redakcyjny

dr inż. Mariusz Dramski, US
dr inż. Piotr Dziurzański, ZUT
dr Włodzimierz Masierak, UKW
dr Piotr Prokopowicz, UKW

Rada Naukowa

	dr hab. Stanisław	Ambroszkiewicz	Instytut Podstaw Informatyki PAN
	dr inż. Rafał	Angryk	Montana State University, USA
	dr hab. Zenon	Biniek	Wyższa Szkoła Technologii Informatycznych
prof. dr hab. inż.	Ryszard	Budziński	Zachodniopomorski Uniwersytet Technologiczny
	dr inż. Joanna	Chimiak-Opoka	University of Innsbruck, Austria
prof. dr hab. inż.	Ryszard	Choraś	Uniwersytet Technologiczno-Przyrodniczy
	dr inż. Grzegorz	Domek	Uniwersytet Kazimierza Wielkiego
	dr hab. Petro	Filevych	Lviv National University of Veterinary and Biotechnologies, Ukraina
	dr hab. inż. Piotr	Gajewski	Wojskowa Akademia Techniczna
	dr Marek	Holyński	Prezes Polskiego Towarzystwa Informatycznego
prof. dr hab. inż. czł. rzeczn. PAN	Janusz	Kacprzyk	Instytut Badań Systemowych PAN
	dr hab. inż. Andrzej	Kobyliński	Szkoła Główna Handlowa
prof. dr hab. inż. czł. koresp. PAN	Peter	Kopacek	Vienna University of Technology, Austria
	Józef	Korbicz	Uniwersytet Zielonogórski
prof. dr hab. inż. czł. koresp. PAN	Jacek	Koronacki	Instytut Podstaw Informatyki PAN
	Witold	Kosiński	Uniwersytet Kazimierza Wielkiego, PJWSTK
prof. dr hab. inż.	Marek	Kurzyński	Politechnika Wrocławska
prof. dr hab. inż.	Halina	Kwaśnicka	Politechnika Wrocławska
	prof. dr Miroslaw	Majewski	New York Institute of Technology, United Arab Emirates
	dr hab. Andrzej	Marciniak	Politechnika Poznańska
	dr Marcin	Paprzycki	Instytut Badań Systemowych PAN
prof. dr hab. inż.	Witold	Pedrycz	University of Alberta, Canada
prof. dr hab. inż.	Andrzej	Piegat	Zachodniopomorski Uniwersytet Technologiczny
prof. dr hab. inż.	Andrzej	Polański	Politechnika Śląska
prof. dr hab. inż.	Orest	Popov	Zachodniopomorski Uniwersytet Technologiczny
	dr inż. Izabela	Rojek	Uniwersytet Kazimierza Wielkiego
prof. dr hab. inż.	Danuta	Rutkowska	Politechnika Częstochowska
prof. dr hab. inż.	Leszek	Rutkowski	Politechnika Częstochowska
prof. dr hab. inż. czł. koresp. PAN	Roman	Słowiński	Instytut Badań Systemowych PAN, Politechnika Poznańska
prof. dr hab. inż.	Włodzimierz	Sosnowski	Uniwersytet Kazimierza Wielkiego, IPPT PAN
prof. dr hab. inż.	Andrzej	Stateczny	Akademia Morska w Szczecinie
	dr hab. Janusz	Szczepański	Uniwersytet Kazimierza Wielkiego, IPPT PAN
prof. dr hab. inż. czł. koresp. PAN	Ryszard	Tadeusiewicz	Akademia Górniczo-Hutnicza
prof. zw. dr hab. inż. czł. rzeczn. PAN	Jan	Węglarz	Instytut Chemii Bioorganicznej PAN, Politechnika Poznańska
	prof. dr hab. inż. Sławomir	Wierchoń	Instytut Podstaw Informatyki PAN
	dr hab. inż. Antoni	Wiliński	Zachodniopomorski Uniwersytet Technologiczny
	dr hab. inż. Andrzej	Wiśniewski	Akademia Podlaska
	dr hab. inż. Ryszard	Wojtyna	Uniwersytet Technologiczno-Przyrodniczy
	dr hab. Sławomir	Zadrozny	Instytut Badań Systemowych PAN
	dr hab. Zenon	Zwierzewicz	Akademia Morska w Szczecinie

**Studies and Materials
in
Applied Computer
Science**

Studies and Materials in Applied Computer Science

Journal of young researchers, PhD students and students

Endorsed by Polish Information Processing Society

Chairman of Editorial Board

Janusz Kacprzyk Systems Research Institute, PAS

Editors-in-Chief

Jacek Czerniak Kazimierz Wielki University in Bydgoszcz, SRI PAS
Marek Macko Kazimierz Wielki University in Bydgoszcz

Secretary

Iwona Filipowicz Kazimierz Wielki University in Bydgoszcz

Editorial Office

Mariusz Dramski University of Szczecin
Piotr Dziurzański West Pomeranian University of Technology
Włodzimierz Masierak Kazimierz Wielki University in Bydgoszcz
Piotr Prokopowicz Kazimierz Wielki University in Bydgoszcz

Editorial Board

Stanisław	Ambroszkiewicz	Institute of Computer Science, PAS
Rafał	Angryk	Montana State University, USA
Zenon	Biniek	University of Information Technology
Ryszard	Budziński	University of Szczecin
Joanna	Chimiak-Opoka	University of Innsbruck, Austria
Ryszard	Choraś	University of Technology and Life Sciences in Bydgoszcz
Grzegorz	Domek	Kazimierz Wielki University in Bydgoszcz
Petro	Filevych	Lviv National University of Veterinary and Biotechnologies, Ukraina
Piotr	Gajewski	Military University of Technology
Marek	Holyński	President of Polish Information Processing Society
Janusz	Kacprzyk	Systems Research Institute, PAS
Andrzej	Kobyliński	Warsaw School of Economics
Peter	Kopacek	Vienna University of Technology, Austria
Józef	Korbicz	University of Zielona Góra
Jacek	Koronacki	Institute of Computer Science, PAS
Witold	Kosiński	Kazimierz Wielki University in Bydgoszcz, PJIIT
Marek	Kurzyński	Wrocław University of Technology
Halina	Kwaśnicka	Wrocław University of Technology
Mirosław	Majewski	New York Institute of Technology, United Arab Emirates
Andrzej	Marciniak	Poznań University of Technology
Marcin	Paprzycki	Systems Research Institute, PAS
Witold	Pedrycz	Systems Research Institute, PAS
Andrzej	Piegat	West Pomeranian University of Technology
Andrzej	Polański	Silesian University of Technology
Orest	Popov	West Pomeranian University of Technology
Izabela	Rojek	Kazimierz Wielki University in Bydgoszcz
Danuta	Rutkowska	Częstochowa University of Technology
Leszek	Rutkowski	Częstochowa University of Technology
Roman	Słowiński	Systems Research Institute, PAS, Poznań University of Technology
Włodzimierz	Sosnowski	Kazimierz Wielki University in Bydgoszcz, IFTR PAS
Andrzej	Stateczny	Maritime University of Szczecin
Janusz	Szczepański	Kazimierz Wielki University in Bydgoszcz, IFTR PAS
Ryszard	Tadeusiewicz	AGH University of Science and Technology
Jan	Węglarz	Institute of Bioorganic Chemistry, PAS, Poznań University of Technology
Sławomir	Wierchoń	The Institute of Computer Science, PAS
Antoni	Wiliński	West Pomeranian University of Technology
Andrzej	Wiśniewski	University of Podlasie
Ryszard	Wojtyna	University of Technology and Life Sciences in Bydgoszcz
Sławomir	Zadrozny	Systems Research Institute, PAS
Zenon	Zwierzewicz	Maritime University of Szczecin

Studia i Materiały Informatyki Stosowanej

czasopismo młodych pracowników
naukowych, doktorantów i studentów

Tom 2, Nr 3, 2010

Bydgoszcz 2010

Studia i Materiały Informatyki Stosowanej
czasopismo młodych pracowników naukowych, doktorantów i
studentów

© Copyright 2010 by Uniwersytet Kazimierza Wielkiego

**Autorzy dziękują Fundacji Rozwoju Mechatroniki za przyznanie im *stypendium
wydawniczego*, które pokryło całość kosztów wydawniczych umożliwiając publikację.**

Sponsor wydania:

Fundacja Rozwoju Mechatroniki
ul. Jeżynowa 19
85-343 Bydgoszcz
tel. +48 505-44-77-53
fax. +48 525-81-22-51
email: biuro@mechatronika.org.pl
www.mechatronika.org.pl
NIP: 9671334879, REGON: 340787784, KRS: 0000300630

ISSN 1689-6300
ISBN 978-83-932977-1-9

Projekt okładki: Łukasz Zawadzki
DTP: Sebastian Szczepański

Wydawca:

Wydział Matematyki, Fizyki i Techniki
Uniwersytet Kazimierza Wielkiego
ul. Chodkiewicza 30
85-064 Bydgoszcz
tel. (052) 34-19-331
fax. (052) 34-01-978
e-mail: simis@ukw.edu.pl

Kontakt:

dr inż. Jacek Czerniak
dr inż. Marek Macko
Uniwersytet Kazimierza Wielkiego
ul. Chodkiewicza 30
85-064 Bydgoszcz
*e-mail: jczerniak@ukw.edu.pl
mackomar@ukw.edu.pl*

Druk (ze środków sponsora):
Oficyna Wydawnicza MW

Nakład 510 egz.

Bydgoszcz 2010

SPIS TREŚCI

Słowo wstępne	11
Nadesłane Artykuły	13
Metody oceny jakości dźwięku w telefonii Voice over IP	
ŁUKASZ APIECIONEK	13
Zastosowanie sztucznych sieci neuronowych do rozpoznawania metali na podstawie wykresu statycznej próby rozciągania	
DAWID EWALD	21
Koncepcja bazy danych jako podstawowej części programu generującego oceny opisowe w nauczaniu wczesnoszkolnym	
ARLETA TRZECIAKOWSKA	31
Projektowanie mechanizmów i układów androidalnych na podstawie sterowanego komputerowo manipulatora AndroArm002	
ZBIGNIEW SZCZEPAŃSKI	39
Zastosowanie sztucznych sieci neuronowych do rozpoznawania obrazów	
WOJCIECH DOBROSIELSKI	45
Zaproszenie do Publikowania.....	57

Contents

Editorial	11
Submitted Articles	13
ŁUKASZ APIECIONEK	13
Quality evaluation methods of a sound in Voice over IP telephony	
DAWID EWALD	21
Application of artificial neural networks for recognition of metals on the basis of static tensile test chart	
ARLETA TRZECIAKOWSKA	31
Conception of a database as fundamental part of the program generating the descriptive grades in elementary schools	
ZBIGNIEW SZCZEPAŃSKI	39
Projecting and constructing of a robot arm mechanisms controlled by a computer module - project AndroArm002	
WOJCIECH DOBROSIELSKI	45
Using artificial neural networks in image recognition.....	45
Call for papers	59

SŁOWO WSTĘPNE

Szanowni Czytelnicy,

Dziękujemy bardzo serdecznie wszystkim, którzy włączyli się do promowania SiMIS wśród swoich środowisk, szczególne podziękowania kierujemy do Członków Rady Naukowej, dostajemy z różnych stron kraju wiadomości, że informacje o SiMIS pojawiają się w gablotach i podczas spotkań z młodzieżą akademicką. Szczególnie dziękujemy Dziekanom i Kierownikom Studiów Doktoranckich za propagowanie SiMIS wśród swoich młodych współpracowników. Miło nam poinformować, że numery SiMIS wydawane w nadchodzącym roku 2011 ukażą się w języku angielskim. Stanie się tak głównie dzięki dotacji z MNiSW w ramach programu IndexPlus, która pozwoli pokryć zwiększone koszty edycji anglojęzycznej. Cieszymy się bardzo, że dzięki temu SiMIS będzie dostępny dla szerokiej grupy odbiorców. Przypominamy, że dzięki staraniom grona redakcyjnego zwieńczonym pozyskaniem hojnych sponsorów druk artykułów pozostaje darmowy dla autorów. Zapraszamy też PT sponsorów, którzy chcieliby wesprzeć nasze wspólne czasopismo.

W nadchodzącym Nowym Roku 2011 życzymy wszystkim Autorom, Czytelnikom oraz środowiskom akademickim, z których Państwo pochodzą wielu śmiałych planów i skutecznych realizacji.

Redaktorzy Naczelní SiMIS,
dr inż. Jacek Czerniak,
dr inż. Marek Macko

NADEŚLANE ARTYKUŁY

METODY OCENY JAKOŚCI DŹWIĘKU W TELEFONII VOICE OVER IP

Łukasz Apiecionek

*Uniwersytet Kazimierza Wielkiego
Instytut Mechaniki i Informatyki Stosowanej
doktorant
ul. Chodkiewicza 30, 85-064 Bydgoszcz
e-mail: lukaszapi@gmail.com*

Streszczenie: *Algorytmy oceny jakości dźwięku pracują z sygnałem wzorcowym bądź bez niego. Metody charakteryzują się różnymi wymaganiami na moc obliczeniową procesora oraz pamięć danych. Determinuje to możliwość zastosowania metod w trybie monitorowania stanu pracy sieci VoIP. Niniejszy artykuł przedstawia opis metod oceny jakości dźwięku, które można próbować zastosować dla telefonii VoIP.*

Słowa kluczowe: *VoIP, jakość dźwięku, PESQ, E-model*

Quality evaluation methods of a sound in Voice over IP telephony

Abstarct: *Voice quality measurement algorithms could work with or without reference signal. They have different needs for CPU performance and data memory. This needs could disturb possibility of using algorithms in real VoIP network. This paper contains descriptions of quality measurement algorithms in case of using its in VoIP networks.*

Keywords: *VoIP, voice quality, PESQ, E-model*

1. WSTĘP

Metody oceny jakości dźwięku można podzielić na:

- metody subiektywne,
- metody obiektywne.

Metody subiektywne polegają na ocenie systemu poprzez jego użytkownika w laboratorium testowym. Metody obiektywne oceniają wartość jakości przez porównanie próbek mowy nadanej z mową u odbiorcy.

Ważną grupę stanowią metody estymowane, w których obliczana jest wartość MOS (przewidująca wartość obiektywną) na podstawie metryk różnorodnych

parametrów np. opóźnienia, jitter, utraty pakietów dźwięku, czy zastosowanego sposobu jego kodowania.

Zarówno metody subiektywne jak i obiektywne mogą być realizowane w trybie odsłuchowym jak i konwersacyjnym. Metody odsłuchowe polegają na transmisji jednokierunkowej dźwięku, podczas gdy metody konwersacyjne wykorzystują dźwięk w przeprowadzanej rozmowie.

Uwzględniając sposób pomiaru jakości sygnału możemy rozpatrywać następujące tryby pomiaru:

- na obu końcach systemu (ang. double-ended),
- na jednym końcu systemu (ang. single-ended),
- na podstawie parametrów systemu,
- hybrydowo - wykorzystując parametry systemu jak i pomiar sygnałów.

Ze względu na algorytm działania metody oceny jakości można również podzielić na:

- ingerujące w strukturę badanego systemu,
- nie ingerujące w badany system (ang. *non-intrusive*) [1] [2].

Metody oceny jakości dźwięku, których algorytm działania ingeruje w badany system, potrzebują sygnału wzorcowego – sygnału odniesienia, który w warunkach laboratoryjnych można w dość prosty sposób dostarczyć do systemu pomiarowego, podczas gdy w systemie rzeczywistym, wymaga to specjalnych rejestratorów dźwięku, które po podłączeniu do systemu pomiarowego muszą być zsynchronizowane w czasie, ze względu na opóźnienia w wykorzystywanej sieci. Bazują one na porównaniu różnic w sygnale, który został przesłany przez system z sygnałem, który został do badanego systemu przyłożony. Na podstawie tych różnic określają wynik pomiaru jakości transmisji sygnału w systemie. Metody oceny jakości dźwięku, które polegają tylko na „podśluchu” badanego systemu, nie ingerują w niego, pozwalając na jego normalne funkcjonowanie, przez co spełniają wymagania na metody służące do nadzorowania pracy systemu. Bazują one na parametryzacji sygnału, obliczaniu jego cech charakterystycznych, a następnie na ich podstawie określają wynik pomiaru.

2. PRZEGLĄD METOD OCENY JAKOŚCI DŹWIĘKU

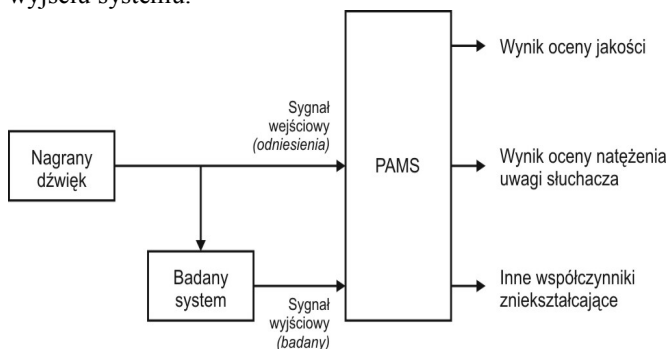
I. PAMS

Intencją opracowania metody PAMS (ang. Perceptual Analysis Measurement System) było znalezienie metody pomiaru jakości dźwięku w sieciach VoIP w przypadku występowania takich zdarzeń jak:

- czasowe przerwy w sygnale,
- straty pakietów,
- jitter,
- zniekształcenia wniesione poprzez różne metody kodowania dźwięku.

Metoda została przedstawiona w 1998 roku poprzez firmę PsyTechnics i pracuje z sygnałem odniesienia [3]. Oznacza to, że system może być badany:

- w warunkach laboratoryjnych,
- poprzez nagrywanie próbek dźwięku na wejściu i wyjściu systemu.



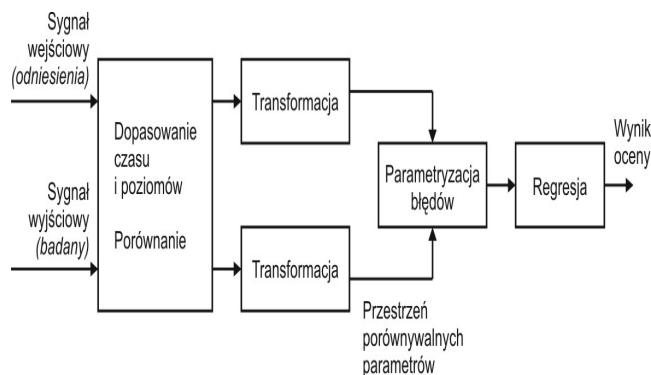
Rysunek. 1 Schemat pomiaru za pomocą metody PAMS (ang. Perceptual Analysis Measurement System)

Metoda poza wynikiem oceny jakości przesłanego dźwięku dostarcza informacji na temat prawdopodobnego wymaganego natężenia uwagi słuchacza oraz innych współczynników określających zniekształcenia dźwięku w badanym sygnale. Wynik oceny jakości przesłanego dźwięku dokonywany metodą PAMS przedstawiony jest w skali MOS. Metoda PAMS jest w stanie zmierzyć wpływ jitter'a, przesunięcia w czasie i poziomów sygnału oraz poziom szumów, natomiast nie uwzględnia opóźnienia w transmisji sygnału [3].

Warunkami brzegowymi użycia metody PAMS jest:

- brak dużych poziomów szumów,
- stały poziom mocy sygnału badanego,
- brak zbyt dużego poziomu jitter'a.

Krytycznym elementem metody jest odwzorowywanie wyników na skalę MOS [4].



Rysunek. 2 Schemat strukturalny metody PAMS

II. PESQ

Metoda PESQ (ang. Perceptual Evaluation of Speech Quality) została opracowana i zaakceptowana przez ITU w 2001 roku jako rekomendacja P.862 i szybko stała się szeroko używaną metodą do oceny jakości systemów [5]. Ocenia jakość dźwięku z uwzględnieniem:

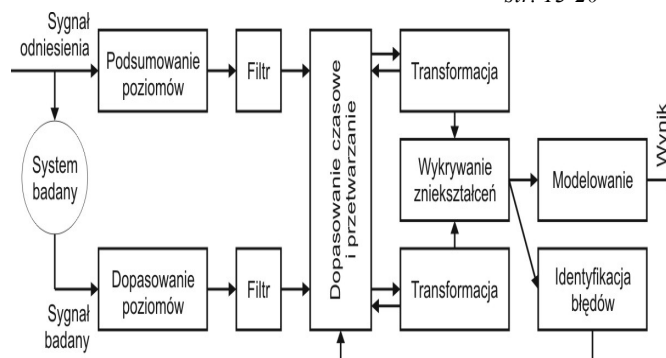
- czasowych przerw w sygnale,
- straty pakietów,
- efektów filtracji sygnału,
- jitter'a,
- zniekształceń wniesionych poprzez kanał transmisyjny,
- transkodowania,
- szumów dodanych przez system transmisyjny.

Metoda ta nie uwzględnia i nie nadaje się do pomiarów [3]:

- opóźnień, utraty poziomu sygnału jak i całkowitego poziomu szumów, co spowodowane jest wprowadzeniem bloków dopasowujących sygnał w czasie i w jego poziomie,

- wpływu jednoczesnej rozmowy w dwóch kierunkach,
- niedopasowania kodowania sygnału,
- szumów w sygnale wejściowym,
- muzyki w sygnale wejściowym,
- dźwięku po kompresji o przepływnościach mniejszych od 4 kbs,
- echa,
- sprzężenia zwrotnego sygnałów.

Braki w implementacji wymienionych efektów w metodzie PESQ powodują, iż czasami wynik oceny jakości sygnału może być dobry (wysoki wskaźnik jakości MOS) podczas gdy sygnał faktycznie jest bardzo słabej jakości.



Rysunek. 3 Schemat algorytmu metody PESQ (ang. Perceptual Evaluation of Speech Quality)

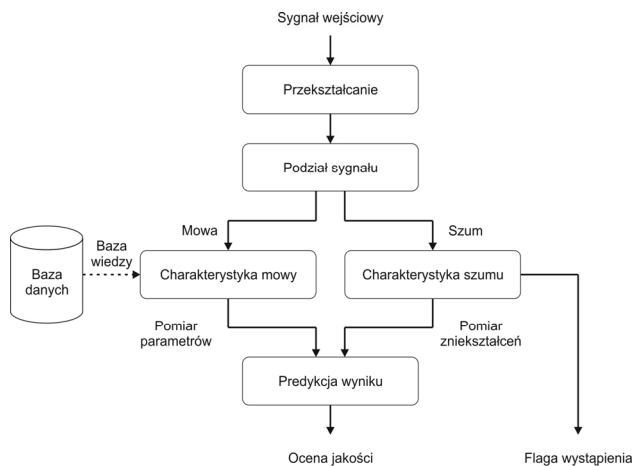
III. PSQM

Metoda PSQM (ang. Perceptual Single ended Objective Measure) została zarekomendowana przez ITU-T w 1996 roku. Zniekształcenia sygnału są mierzone na podstawie porównań i odniesień do statystycznych wyników zawartych w bazie danych. Dysponowanie odpowiednią bazą danych jest ograniczeniem i główną wadą metody PSQM. Metoda PSQM bazuje na analizie parametrów w dziedzinie czasowej i częstotliwościowej. Sygnał mowy jest oddzielany od szumu, a następnie oceniany z wykorzystaniem statystycznej bazy danych. Pomiar poziomu zniekształceń jest dokonywany na podstawie pomiaru szumów. Korelacja poziomu zniekształceń wraz z parametrami mowy pozwala na predykcję wyniku jakości sygnału mowy w systemie. Wynik metody PSQM jest wykorzystywany do obliczenia wyniku skali MOS według wzoru [6]:

$$MOS = \frac{4}{1 + e^{0,66 PSQM - 2,2}} + 1$$

gdzie:

- $PSQM$ - wynik oceny jakości dźwięku przez metodę PSQM.



Rysunek. 4 Algorytm działania metody PSQM (ang. Perceptual Single ended Objective Measure)
IV. E-model

E-model został zaproponowany przez ETSI (Europejski Instytut Standardów Telekomunikacji) [9]. ITU wydało stosowne rekomendacje w 1998 roku [7][8]. Zadaniem modelu było dostarczenie dla operatorów i projektantów narzędzia do projektowania sieci IP uwzględniającego czynniki obiektywne, wynikające z konfiguracji sieci, metody transmisji, czy stosowanych kodeków. E-model jest modelem matematycznym, który zakłada, że pomiędzy ustami osoby mówiącej, a uchem słuchacza mamy do czynienia z kanałem telekomunikacyjnym składającym się z wielu torów i elementów, które jako całość tworzą łącze telefoniczne. Jakość transmitowanej mowy jest zależna od następujących parametrów i czynników:

- rodzaju elementów końcowych, mianowicie telefony analogowe i cyfrowe:
 - stacjonarne,
 - bezprzewodowe,
 - mobilne,
- rodzaju elementów połączeń, analogowe i cyfrowe centrale:
 - abonenckie,
 - międzymiastowe,
- rodzaju elementów transmisji, łącza:
 - przewodowe,
 - światłowodowe,
 - cyfrowe,
- konfiguracji połączenia:
 - długości połączenia,
 - typu połączenia,
 - rodzaju składników sieci,
- parametrów mowy i słuchu człowieka,

- tłumienia transmisji mowy między mówcą a słuchaczem,
- tłumienia linii abonenckiej,
- tłumienia obwodów 4-przewodowych,
- efektu lokalny echa mówcy i słuchacza,
- echa,
- stabilności systemu,
- czasu transmisji – opóźnienia,
- szumu i zniekształceń kwantyzacji,
- przesłuchów.

Ważnym ogniwem jest też występowanie urządzeń specjalnych, takich jak kodery mowy o małej przepływności binarnej, systemy mobilne, systemy wykrywania mowy VAD (ang. Voice Active Detection) czy systemy eliminacji echa za pomocą procesorów DSP. W E-modelu zakłada się, że pomiar jakości mowy przesyłanej poprzez łącze telekomunikacyjne odbywa się w słuchawce telefonicznej słuchacza dla pasm częstotliwości od 300 Hz do 3400 Hz. Wynik oceny jakości mowy w E-modelu wyznacza się za pomocą współczynnika jakości transmitowanego dźwięku R , który określa zadowolenie bądź niezadowolenie użytkowników z jakości mowy w badanym systemie. Należy podkreślić, że stopień zadowolenia bądź jego braku jest zależny zarówno od sposobu mówienia mówcy jak i koncentracji osoby słuchającej.

Współczynnik R wyrażany jest wzorem:

$$R = R_O - I_s - I_d - I_e + A$$

gdzie:

- R_O - podstawowy współczynnik sygnał/szum SNR (ang. Signal Noise Ratio) transmisji mowy w punkcie odniesienia 0 dBr;
- I_s – współczynnik pogorszenia jakości transmitowanej mowy, reprezentujący zniekształcenia mowy podobne jak przy odbiorze zbyt głośnego sygnału mowy, przy nieoptymalnym efekcie lokalnym echa lub na skutek zniekształceń kwantyzacji;
- I_d – współczynnik pogorszenia jakości transmitowanej mowy, wywołanego echem odnoszącym się do sygnału mowy, tj. tłumienności głośności echa mówcy TELR (ang. Talker Echo Loudness Rating) i ważonej straty efektu echa podczas transmisji w systemie WEPL (ang. Weighted Echo Path Lost) oraz trudności w porozumiewaniu się, spowodowane zbyt dużym bezwzględnym opóźnieniem transmitowanych próbek dźwięku T_a ;
- I_e – współczynnik pogorszenia jakości transmitowanej mowy, spowodowanego zastosowaniem w łączu specjalnych urządzeń, tj. koderów o małej przepływności binarnej, DCME (ang. Digital Circuit Multiplication

Equipment), VPE (ang. Voice Packeting Equipment) i innych; wpływ tych urządzeń na jakość mowy jest bardzo złożony i trudno jest go analizować, biorąc pod uwagę poszczególne parametry;

- A – współczynnik oczekiwania jakości dźwięku w systemie telekomunikacyjnym, będący wielkością dodatnią, reprezentującą korzyści wynikające z używania różnych systemów transmisji mowy. Np. w niektórych okolicznościach systemy radiowe mają znaczną przewagę nad systemami kablowymi, pomimo gorszej jakości transmisji mowy. Dotyczy to rozwoju telefonii mobilnej w trudno dostępnych regionach, gdzie względy ekonomiczne okazały się ważniejsze niż jakość połączenia [9]. Skrócony opis modelu E pokazuje jego przydatność i uniwersalność do oceny różnorodnych projektów sieci i występujących w nich możliwych połączeniach [9]. Wykorzystując E-model można przewidywać jakość mowy w budowanym systemie, jednak konieczne jest odpowiednie dobranie współczynników do wzorów obliczeniowych. Model ten jest dobrym modelem do badań w środowiskach testowych, gdzie w idealnych warunkach można dokonywać pomiarów wymaganych parametrów. Natomiast nie nadaje się do użycia w działającej sieci oferującej usługę VoIP, ponieważ nie ma wówczas możliwości pomiaru wszelkich niezbędnych do prawidłowego działania E-modelu parametrów.

V. P.563

Metoda P.563 ta została zatwierdzona przez ITU-T w maju 2005 roku [10]. Wydana rekomendacja opisuje metodę, nie ingerującą w badane środowisko, do predykcji subiektywnej oceny jakości przesyłanej mowy w wąskopasmowej telefonii (pasmo 3,1 kHz). Należy tutaj podkreślić, iż algorytm metody jest aplikacją przewidującą jakość systemu bez użycia sygnału odniesienia. ITU zaleca tą metodę do monitorowania sieci przy użyciu nieznanego źródła dźwięku pochodzącego z dalekiego miejsca w sieci. Jest to pierwsze podejście ITU do metod pomiaru jakości bez sygnału odniesienia. Algorytm był testowany z różnorodnymi źródłami dźwięku w laboratoriach różnych firm, co skutkowało publikacjami dokumentów i wyników tych badań [11]. Metoda P.563 została opracowana w taki sposób, żeby istniała możliwość pomiaru takich cech

w transmitowanym dźwięku jak:

- charakterystyka akustyczna środowiska,
- szum środowiska po stronie nadawczej,
- charakterystyka akustyczna interfejsu nadawczego,
- zniekształcenia spowodowane elektroniką po stronie nadawczej,

- charakterystyka kodowania,
 - poziom sygnału nadawanego,
 - błędy w kanale transmisyjnym,
 - straty pakietów i ukryte straty w wyniku stosowania kodowania CELP (ang. Code Excited Linear Prediction),
 - przepływność standardu kodowania próbek dźwięku jeśli posiada więcej niż jedną,
 - zmiana sposobów kodowania próbek dźwięku,
 - efekt jitter w odniesieniu do testów ACR,
 - krótkotrwałe wypaczenia w dźwięku,
 - długotrwałe wypaczenia w dźwięku,
 - systemy transmisyjne eliminacji echa i szumów w warunkach rozmowy jednokierunkowej w odniesieniu do testów ACR,
 - różne technologie kodowania, np. G.711; G.726; G.727, G.729, G.723.1, G.728,
- Metodę P.563 rekomenduje się do użycia w następujących środowiskach:

- monitorowanie stanu pracy sieci w czasie rzeczywistym stosując interfejs cyfrowy bądź analogowy,
- testowanie całościowe (od nadawcy do odbiorcy) systemu transmisji w czasie rzeczywistym,
- testowanie całościowe (od nadawcy do odbiorcy) systemu transmisji w czasie rzeczywistym przy braku znajomości sygnału nadawanego.

W dokumentacji metody P.563 [10] podane są efekty, które wpływają na niewłaściwe oszacowanie wyniku:

- poziom słyszanego dźwięku,
- straty poziomu sygnału,
- efekt opóźnienia w teście rozmowy dwukierunkowej,
- echo,
- muzyka w sygnale wejściowym,
- kodowanie według standardu LPC (kodowanie z predykcją liniową, ang. Linear Predictive Coding) o przepływności mniejszej niż 4 kbit/s,
- niedopasowanie przepływności standardu kodowania dźwięku na wejściu i wyjściu systemu,
- stosowanie syntetyzowanej mowy.

Algorytm metody P.563 wymaga obliczenia parametrów do oszacowania różnorodnych zjawisk w transmisji dźwięku (ludzkiej mowy) i na ich podstawie dokonywane jest modelowanie jakości dźwięku. W metodzie P.563 można wydzielić trzy główne klasy pomiarowe:

- analiza wokalna traktu oraz nienaturalności dźwięku (mowy):

- głos mężczyzny,
- głos kobiety,
- głos syntetyzowany,

Klasa ta jest głównym elementem testów występowania nienaturalności w transmitowanej mowie. Zawiera komponenty wykrywania w sygnale danych, które mogą

być interpretowane jako głos oraz osobno wykrywają dane reprezentujące szумы. Ponadto odpowiada za wykrywanie powtarzających się próbek w dźwięku. Jednym z zadań elementów tej klasy jest porównanie sygnału z wygenerowanym sztucznie sygnałem odniesienia.

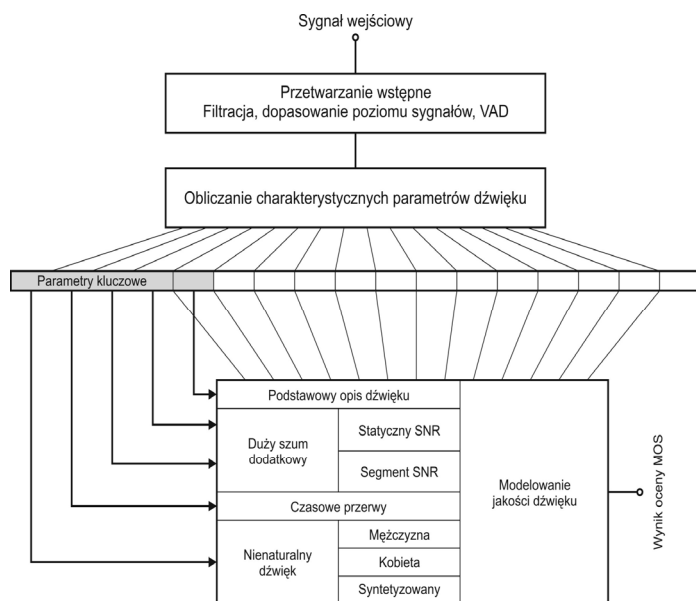
- analiza dodatkowego szumu:

- statyczny pomiar SNR,
- segment pomiaru SNR,

Zadaniem elementów w tej klasie jest ocena charakterystyk różnych szumów, które mogą wystąpić w transmitowanym dźwięku. Oszacowują również czy występujący szum jest zdarzeniem lokalnym, czy globalnie ma wpływ na ogólną jakość sygnału.

- przerwy, cisza i czasowe zaniki dźwięku

Zadaniem elementów tej klasy jest wykrywanie przerw w sygnale, które mogą być spowodowane stosowaniem mechanizmów VAD, DCME albo zakłóceń w kanale transmisyjnym. Algorytm stosowany w metodzie zaprojektowany jest do wykrywania zarówno normalnego zakończenia słów jak i ich urwania.



Rysunek. 5 Algorytm metody P.563

Ponadto metoda P.563 stosuje deskryptory opisujące poziom dźwięku, aktywność mowy i jej zmiany. Wykorzystywane są funkcje wykrywania głosu VAD i analiza danych.

VI. K-factor

Metoda ta jest wykorzystywana przez firmę Cisco w rozwiązaniach do budowy sieci oferujących usługę

VoIP. Telefony firmy Cisco potrafią na bieżąco informować użytkownika o wartości wskaźnika MOS przeprowadzanej rozmowy. Ponadto są w stanie podawać bieżącą wartość opóźnienia oraz jitter. Uwzględniając informacje o sposobie kodowania dźwięku, jaki jest używany w przeprowadzanej rozmowie, telefon umożliwia uzyskanie dość dużej ilości danych na temat stanu pracy sieci. Informacje o wartości wskaźnika MOS są przechowywane w bazach danych Cisco Call Managera, który pełni rolę „centrali” dla urządzeń firmy Cisco. Według informacji od producenta metoda K-factor estymuje wartość wskaźnika MOS na podstawie:

- stosowanej metody kodowania dźwięku,
- liczby utraconych pakietów.

Metoda nie uwzględnia w swoich obliczeniach żadnych parametrów „kanału” związanego choćby z opóźnieniem pakietów transmitowanych w sieci. Algorytm metody nie jest znany, przez co trudno mieć zaufanie do wyników jakie przedstawia

3. PODSUMOWANIE I WNIOSKI

Analizując metody oceny jakości transmisji dźwięku postanowiono dokonać ich oceny pod kątem przydatności do zastosowań w monitorowaniu stanu pracy sieci oferujących usługę VoIP.

E-Model

Głównym celem opracowania E-modelu była konieczność posiadania narzędzia do projektowania sieci telefonicznych. Stosując wzory obliczeniowe przy ekstrakcji parametrów sieci rzeczywistej można pokusić się pod pewnymi warunkami o predykcję jakości. Otóż algorytm musi posiadać parametry bazowe – zmierzone w środowisku pracy terminala VoIP czy parametry samego terminala VoIP. Ponieważ w sieci nie można określić i przewidzieć wszystkich warunków pracy, nie jest możliwe przygotowanie parametrów do obliczeń. Ponadto trudno jest przewidzieć jakie parametry będzie miał sam terminal i nie można określić czy jego parametry nie uległy zmianie w stosunku do parametrów zakładanych. Gdyby zbudować sieć firmową tylko z urządzeń jednego producenta, to i tak zastosowanie modelu E byłoby trudne.

K-factor

Metoda bazująca na wskaźniku K jest metodą opracowaną przez producenta urządzeń i może być wykorzystana tylko

przy pracy z urządzeniami pochodzącymi od niego. Gdy sieć oferująca usługę VoIP nie jest siecią zamkniętą – tj. jednego producenta, gdy łączy się z sieciami publicznymi, określanie jakości dźwięku przy wykorzystaniu metody K-factor jest już ograniczone. Brak pełnej dokumentacji metody sprowadza ją do roli wskaźnika w zamkniętym systemie telefonów wymienionej firmy. Ponadto, jak już wspomniano, metoda nie uwzględnia w swoich obliczeniach żadnych parametrów związanych choćby z opóźnieniem pakietów transmitowanych w sieci, mimo, że terminale posiadają takie informacje.

PSQM

Wadą zastosowania metody PSQM jest konieczność posiadania statystycznej bazy danych służącej do określania wpływu parametrów na jakość sygnału. Metodę można by zastosować w sieci z usługą VoIP, gdyby parametry dźwięku były zbierane w punktach sieci, a następnie przesyłane do stacji zarządzającej. Dopiero stacja zarządzająca posiadając odpowiednie zasoby na bazę danych dokonywała by oceny jakości przeprowadzanej rozmowy. Takie rozwiązanie stawia jednak duże wymagania na moc obliczeniową stacji zarządzania. Zamiast tylko zbierać informacje i przekazywać je operatorowi sieci, zajmowała by się nieustannie dokonywaniem obliczeń i predykcją wyników. Konieczna moc obliczeniowa rosła by wraz ze wzrostem ilości użytkowników w sieci co wiąże się z ilością przeprowadzanych rozmów. Podejście to powoduje konieczność budowy skomplikowanej stacji zarządzającej. Nie będzie wówczas możliwości korzystania z oprogramowania do monitorowania stanu pracy sieci z dowolnego terminala w sieci IP.

Metoda PAMS i PESQ

Metody te nie nadają się do zastosowania w monitorowaniu stanu pracy sieci oferujących usługę VoIP ze względu na konieczność posiadania sygnału wzorcowego. Niemożliwe wręcz jest opracowanie sposobów na dostarczenie do każdego badanego punktu w sieci sygnału odniesienia, przeprowadzenie rozmowy a następnie dokonanie oceny jej jakości. Obecnie metoda PESQ jest najszerzej stosowaną metodą w laboratoriach. Posiada jednak swoje wady. Przeprowadzono testy metody PESQ, których wyniki były bardzo zaskakujące [12]. Otóż przygotowano dwa fragmenty dźwięku poprzez modyfikację wzorca. W pierwszym podejściu wzorzec zmiksowano z szumem białym o niskim poziomie mocy. Po odsłuchu uzyskanego nagrania nie stwierdzono

problemów z jakością. Natomiast drugie nagranie do badań uzyskano poprzez poddanie wzorca obróbce polegającej na zmniejszeniu ilości składowych harmonicznych o małej i wysokiej częstotliwości. Odsłuchiwane nagranie znacznie odbiega od oryginału i bez problemów można stwierdzić utratę jakości. Wyniki pomiaru jakości nie wykazały różnic.

Metoda P.563

Wadą metody P.563 jest stosunkowo duży błąd w przypadku krótkotrwałej utraty dźwięku. Przeprowadzone przez autora badania wykazały, że metoda w takich przypadkach przedstawia błędne wyniki oceny jakości (wynik jest zaniżony bądź zawyżony), w związku z czym nie spełnia postawionych przed nią wymagań

Podsumowując przegląd przedstawionych w artykule metod oceny jakości transmitowanego dźwięku, można stwierdzić, że nie ma dostatecznie dobrej metody, która nadawała by się do zastosowania w sieci telefonicznej VoIP. W związku z czym, istnieje uzasadniona konieczność prac nad algorytmami oceny jakości transmisji dźwięku dla sieci telefonicznej VoIP.

Literatura

1. J. Davidson, J. Peters, VOICE OVER IP PODSTAWY, Mikom, 2005
2. D. Picovici, A. E. Mahdi, Towards Non-intrusive Speech Quality Assessment for Modern Telecommunications, First Joint IEI/IEEE Symposium on Telecommunications Systems Research, Dublin, 2001
3. E. Bernex, A. Gatineau, Quality of service in VoIP environments, White Paper, www.neotip.com
4. M.P. Hollier, PAMS: Measuring speech quality over networks, as the customers hear it, Psytechnics white paper, 2001
5. J. Anderson, Methods for measuring perceptual speech quality, Agilent Technologies white paper, 2000
6. R. Dai, A Technical White Paper on Sage's PSQM Test, White paper, 2000
7. ITU-T Recommendation G.107, The E-model, a computational model for use in transmission planning, grudzień 1998
8. ITU-T Recommendation G.109, Definition of categories of speech transmission quality, wrzesień 1999
9. M.J. Trzaskowska, Matematyczny model do oceny jakości mowy przesyłanej w sieciach

telekomunikacyjnych, Telekomunikacja i techniki informacyjne, nr 1, 2001

10. ITU T Recommendation P.563, Single-ended method for objective speech quality assessment in narrow-band telephony applications, maj 2004

11. Opticom GmbH, P.563 – Single Sided Speech Quality Measure, www.opticom.de, 2006

12. P. Ordas, B. Fox, Perceptual Evaluation of Speech Quality (PESQ), Microtronix Systems Ltd, 2004, <http://www.microtronix.ca/pesq-disc.html>

ZASTOSOWANIE SZTUCZNYCH SIECI NEURONOWYCH DO ROZPOZNAWANIA METALI NA PODSTAWIE WYKRESU STATYCZNEJ PRÓBY ROZCIĄGANIA

Dawid Ewald

Uniwersytet Kazimierza Wielkiego
Instytut Techniki
doktorant
ul. Chodkiewicza 30/ p.215, 85-064 Bydgoszcz
e-mail: dawidewald@gmail.com

Streszczenie: *W artykule przedstawia zagadnienie sztucznych sieci neuronowych oraz ich wykorzystania w klasyfikacji metali na podstawie wykresu statycznej próby rozciągania. W pracy opisano działanie sieci neuronowych oraz sposób ich wykorzystania..*

Słowa kluczowe: *Sztuczne sieci neuronowe, FANN, Delphi*

Application of artificial neural networks for recognition of metals on the basis of static tensile test chart

Abstarct: *In this article presents the issue of artificial neural networks and their use in the classification of metals on the basis of the static tensile test chart. This paper describes the operation of neural networks and how to use them.*

Keywords: *Artificial neural networks, FANN, Delphi*

1. WSTĘP

Współczesny naukowiec ma do dyspozycji wiele narzędzi ułatwiających pracę naukową. Wiele z tych narzędzi posiada ogromną bazę danych pozwalającą użytkownikowi skracać procesy badawcze oraz wygodniej i szybciej zarządzać zgromadzonymi materiałami. Narzędzia takie często opierają się na algorytmach działania co ogranicza ich skuteczność. Techniki sieci neuronowych pozwalają nam tworzyć narzędzia pozbawione tego ograniczenia a zarazem szybko rozwijać nowe sposoby ich zastosowania w tych sytuacjach, w których mamy do czynienia z nieliniowymi problemami.

2. BIOLOGICZNE PODŁOŻE BUDOWY SZTUCZNEGO NEURONU

Komórka nerwowa zwana potocznie neuronem, stanowi podstawowy element systemu nerwowego organizmu. Neuron składa się z ciała zawierającego jądro oraz odchodzących od niego licznych wypustek, te o strukturze drzewiastej zwane dendrytami (greckiego słowa "déndron", czyli drzewo) oraz grubsze rozwidlający się na końcu aksony.

W neuronie biologicznym sygnał wejściowy doprowadzany jest za pośrednictwem dendrytów, zaś sygnał wyjściowy odprowadzany jest za pomocą aksonu. Upraszczając system działania tak by zilustrować przepływ sygnałów można przyjąć, że przekazywanie impulsu od jednego neuronu do następnego odbywa się za pośrednictwem neuroprzekaźnika

wydzielanego pod wpływem bodźców pochodzących od synaps.

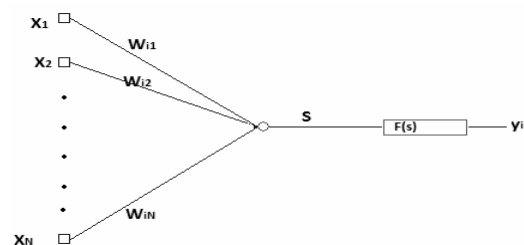
Analogicznie sztuczny neuron zbudowany jest z wejść ($x_1, x_2 \dots x_n$) (biologiczne dendryty), odpowiednim im współczynnikom wagowym ($w_{i1}, w_{i2} \dots w_{in}$) oraz elementu sumującego co przedstawiono na rysunku 2.1.

Sposób przetwarzania informacji wejściowych x na informację wyjściową y uzależniony jest od mogących się zmieniać podczas uczenia współczynników wagowych. Sztuczny neuron mnoży poszczególne sygnały wejściowy x przez odpowiadające im współczynniki wag w a następnie je sumuje. Sygnał wyjściowy może przyjmować 0 lub 1 zgodnie z zależnością:

$$y_i(s_i) = \begin{cases} 1, s_i > 0 \\ 0, s_i \leq 0 \end{cases} \quad (1)$$

$$s_i = \sum_{i=1}^m w_i X_i \quad (2)$$

Gdzie: S_i to sygnał sumatora opisywany wzorem 2.



Rysunek. 2.1 Schemat budowy sztucznego neuronu.

3. JAK DZIAŁA SIEĆ NEURONOWA

Jak wynika z budowy pojedynczego neuronu posiada on swoją pewną wewnętrzną pamięć reprezentowaną poprzez wartość wag i progę. Jednak jego możliwości przetwarzania sygnału są bardzo ograniczone co powoduje małe zapotrzebowanie na moc obliczeniową. Tworząc z pojedynczych neuronów sieć można zbudować system realizujący bardzo złożone zadania. Dlatego sieć neuronowa musi działać jako całość. Działanie sieci polega na przesyłaniu między neuronami kolejnych warstw sygnałów od wejścia pierwszej warstwy do wyjścia ostatniej. Sygnał w poszczególnych neuronach jest przetwarzany i przekazywany do kolejnego.

Najważniejsza cecha sieci neuronowych jest możliwość uczenia. Istnieje wiele sposobów uczenia zależnych od

modelu sieci oraz problemu, który przednią stawiamy. Najczęściej stosowany sposób to uczenie z nauczycielem.

Polega ono na podaniu konkretnego sygnału wejściowego i wyjściowego, pokazując w ten sposób jaka jest wymagana odpowiedź. Dane te zwane są danymi uczącymi i muszą zostać odpowiedni dobrane aby skrócić proces uczenia sieci.

4. CHARAKTERYSTYKA STATYCZNEJ PRÓBY ROZCIĄGANIA

Próba statycznego rozciągania jest podstawową i najczęściej stosowaną próbą wytrzymałościową, jednak trzeba mieć na uwadze fakt, że wielkości charakterystyczne uzyskane podczas przeprowadzania tej próby nie powinny być uznawane jako odzwierciedlenie ogólnego zachowania się konstrukcji pod obciążeniem. Dlatego niektóre elementy tj. liny, których obciążenie robocze to rozciąganie, poddaje się próbie rozciągania w całości.

Statyczna próba rozciągania metali ujęta jest normą polega na poddaniu działaniu siły rozciągającej w kierunku osiowym na odpowiednio przygotowaną próbkę aż do jej zerwania. Mimo iż próbę rozciągania nazywa się statyczną, to jednak obciążenie wolno narasta z określoną prędkością. Przy czym zakłada się, że odpowiadające w stanie spoczynku badanym naprężeniem odkształcenia, pojawiają się natychmiast po przyłożeniu obciążenia, tzn., że istnieje w każdej chwili przeprowadzania badania równowaga naprężenia i odkształcenia. Takie założenie jest słuszne dla odkształceń sprężystych, natomiast w zakresie odkształceń plastycznych dla wielu materiałów przyjęcie takiego założenia jest niezgodne z rzeczywistością. Maksymalny przyrost naprężeń jest określony normą i nie powinien przekraczać 30 MPa/s oraz narastanie obciążeń powinno być ciągłe do swojej maksymalnej wartości.

Maszyna służąca do przeprowadzania próby rozciągania zwana jest zrywarką. Próbkę do przeprowadzania badania posiadają część pomiarową o stałym przekroju i są zakończone główkami o zwiększonych wymiarach. Długość próbki musi być dobrana zgodnie z normą tak by przejściu do główek było łagodne co pozwala przyjąć, że stan odkształcenia i naprężenia w każdym punkcie części pomiarowej próbki jest jednorodny. W tak przyjętych warunkach można z pomiarów odkształceń na powierzchni ciała wyciągać wnioski o odkształceniach wewnątrz ciała, a z pomiarów siły można obliczyć naprężenia wewnątrz próbki.

W normie zostały zawarte następujące wielkości określające własności plastyczne i mechaniczne materiału:

- Średnica początkowa próbki (d_0 [mm]) - średnica próbki na jej długości roboczej mierzona przed rozerwaniem.
- Średnica końcowa próbki (d_u [mm]) - średnica najmniejszego przekroju próbki w miejscu rozerwania.
- Średnica próbki do wyznaczania wydłużenia równomiernego (d_r [mm]) - średnica próbki po rozerwaniu mierzona na dłuższej części próbki w połowie odległości od miejsca jej rozerwania do końca długości pomiarowej.
- Długość pomiarowa początkowa (L_0 [mm]) - długość odcinka badanej próbki, na której określa się wydłużenie.
- Długość próbki (L_t [mm]) - całkowita długość próbki.
- Długość pomiarowa końcowa (L_u [mm]) - długość pomiarowa próbki po rozerwaniu.
- Powierzchnia przekroju początkowego próbki (S_0 [mm²]) - powierzchnia przekroju poprzecznego próbki na długości pomiarowej mierzona przed rozerwaniem.
- Powierzchnia przekroju końcowego (S_u [mm²]) - powierzchnia przekroju poprzecznego próbki w miejscu rozerwania.
- Bezwzględne wydłużenie próbki po rozerwaniu (ΔL [mm]) obliczane na podstawie wzoru 4.1.

$$\Delta L = L_u - L_0 \text{ [mm]} \quad (4.1)$$

- Względne wydłużenie próbki proporcjonalnej po rozerwaniu (A_p [%]) obliczana ze wzoru 4.2:

$$A_p = \frac{\Delta L}{L_0} 100 \text{ [%]} \quad (4.2)$$

gdzie:

p - wskaźnik wielokrotności średnicy d_0 lub wielokrotności $\sqrt{S_0}$.

- Względne wydłużenie równomierne próbki okrągłej (A_r [%]).

$$A_r = \frac{d_0^2 - d_r^2}{d_r^2} 100 \text{ [%]} \quad (4.3)$$

- Względne przewężenie próbki (Z [%]).

$$Z = \frac{d_0^2 - d_u^2}{d_r^2} 100 \text{ [%]} \quad (4.4)$$

- Względne przewężenie próbki okrągłej

$$Z = \frac{S_0 - S_u}{S_0} 100 \text{ [%]} \quad (4.5)$$

- Siła rozciągająca (F [N]) - siła działająca na próbkę w określonej chwili badania.
- Napężenie rozciągające (R [MPa]) - napężenie wyrażone stosunkiem siły F , do przekroju początkowego próbki S_0 .
- Umowna granica sprężystości ($R_{0,05}$ [MPa]) - napężenie odpowiadające działaniu siły rozciągającej, wywołującej w próbce umowne wydłużenie trwałe x wynoszące 0.05% długości pomiarowej L_e wyrażane wzorem 4.6, w technicznie uzasadnionych przypadkach dopuszcza się określenie granicy sprężystości przy wydłużeniach trwałych mniejszych niż 0.05%.

$$R_{0,05} = \frac{F_{0,05}}{S_0} \text{ [MPa]} \quad (4.6)$$

- Umowna granica plastyczności ($R_{0,2}$ [MPa]) - napężenie odpowiadające działaniu siły rozciągającej, wywołującej w próbce umowne wydłużenie trwałe x wynoszące 0.2% długości pomiarowej L_e wyrażane wzorem 4.7, w technicznie uzasadnionych przypadkach dopuszcza się określenie umownej granicy plastyczności przy innych wydłużeniach trwałych w granicach 0.05 - 0.5%.

$$R_{0,2} = \frac{F_{0,2}}{S_0} \text{ [MPa]} \quad (4.7)$$

- Siła odpowiadająca wyraźnej granicy plastyczności (F_e [N]) - siła przy której występuje wyraźny wzrost wydłużenia rozciąganej próbki; dla określonych materiałów rozróżnia się siłę F_{eH} odpowiadającą górnej granicy plastyczności oraz siłę F_{eL} odpowiadającą dolnej granicy plastyczności.
- Wyrażna granica plastyczności (R_e [MPa]) - napężenia odpowiadające działaniu siły F_e obliczane ze wzoru 4.8:

$$R_e = \frac{F_e}{S_0} \text{ [MPa]} \quad (4.8)$$

Rozróżnia się górną granicę plastyczności R_{eH} , w której napężenie odpowiada pierwszemu szczytowi obciążenia, zarejestrowanemu przy badaniu materiału oraz dolną granicę plastyczności R_{eL} odpowiadającą najmniejszej wielkości napężenia przy wyraźnym wzroście wydłużenia; w przypadku, gdy występuje więcej niż jedno minimum pierwszego z nich nie bierze się pod uwagę.

- Największa siła (F_m [N]) - największa siła rozciągająca działająca na próbkę.

- Wytrzymałość na rozciąganie (R_m [MPa]) - naprężenie odpowiadające działaniu siły F_m .

$$R_m = \frac{F_m}{S_0} \text{ [MPa]} \quad (4.9)$$

- Siła rozerwania (F_u [N]) - siła rozciągająca w chwili rozerwania próbki.
- Naprężenie rozrywające (R_u [MPa]) - naprężenie odpowiadające działaniu siły F_m .

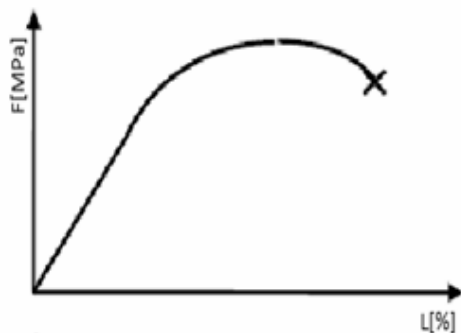
$$R_u = \frac{F_u}{S_0} \text{ [MPa]} \quad (4.10)$$

- Współczynnik sprężystości wzdłużnej (E [MPa]) - stosunek naprężenia R do odpowiadającego mu wydłużenia względnego A_p w zakresie, w którym krzywa rozciągania jest linią prostą.
- Podatność maszyny (K [mm/N]) - stosunek zmiany odległości między uchwytami maszyny wytrzymałościowej do zmiany siły obciążającej.
- Powiększenie skali wydłużeń (α) - stosunek Δl odczytanego na wykresie do rzeczywistego Δl próbki.

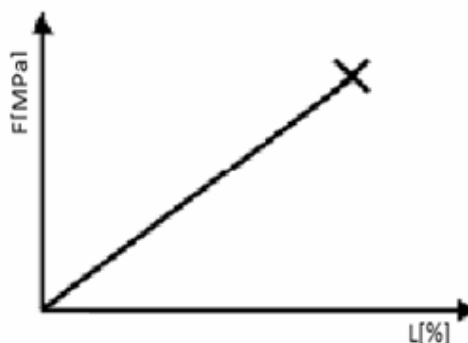
W wyniku przeprowadzania próby uzyskuje się wykres którego kształt zależy od cech plastycznych materiału. Można więc wyróżnić trzy podstawowe kształty wykresy:



Rysunek. 4.1 Materiał z wyraźną granicą plastyczności.



Rysunek. 4.2 Materiały bez wyraźnej granicy plastyczności.



Rysunek. 4.2 Materiał kruchy.

5. OPIS PROGRAMU ROZPOZNAJĄCEGO METAL NA PODSTAWIE WYKRESU STATYCZNEJ PRÓBY ROZCIĄGANIA

Sztuczne sieci neuronowe będą wykorzystywały darmową bibliotekę FANN w wersji 2.0 tworzoną przez Steffen Nissen. Biblioteka ta pozwala w prosty sposób implementować wielowarstwowe sieci neuronowe. Dostarcza również kilka metod uczenia oraz gotowy komponent możliwy do wykorzystania w kompilatorze Borland Delphi 7. Program będzie wykorzystywał cztery nauczone sztuczne sieci neuronowe. Pierwsza z nich zawierać będzie 122500 wejść oraz 3 wyjścia. Zadaniem tej sieci będzie rozpoznanie jednego z trzech kształtów wykresu. Danymi uczącymi były obrazy w formacie BMP o rozdzielczości 350 x 350 pikseli w tonacji monochromatycznej. Obrazy przedstawiają kształty wykresów statycznej próby rozciągania pozbawione osi. Podczas działania program zmienia obraz na postać 0 dla pikseli białych i 1 dla czarnych co znacznie ułatwia przetwarzanie go przez sieć. Tak przygotowane dane interpretowane są przez pierwszą sieć. Trzy kolejne sieci będą rozpoznawać metale podzielane na trzy grupy tj. pierwsza z sieci - metale z wyraźną granicą plastyczności, druga - metale bez wyraźnej granicy plastyczności oraz trzecia - metale kruche. Danymi uczącymi będą liczby odczytane z wykresów statycznej próby rozciągania. Uruchomienie odpowiedniej sieci będzie zależało od wartości na wyjściu sieci pierwszej rozpoznającej kształt wykresu. Obrazy zawierające kształty wykresów jak i same liczby opisujące te wykresy zostały stworzone do potrzeb eksperymentu.

Czas na kod programu!

Cały projekt składa się z trzech programów. Dlaczego aż trzech? Odpowiedź jest prosta. Ponieważ dwa pierwsze programy tworzą i trenują sieci neuronowe, które następnie wykorzystane są w trzeciej aplikacji.

Aplikacja „FANN - interpretacja wykresów”.

Zadaniem tej aplikacji jest stworzenie i wytrenowanie sztucznej sieci neuronowej rozpoznającej kształt wykresu statycznej próby rozciągania. Danymi uczącymi są obrazy w formacie 350 x 350. Aby możliwa była interpretacja przez sieć takich obrazów trzeba te obrazy zamienić na dane „rozumiane” przez sieć. W tym celu musimy obraz zamienić na tablicę zawierającą 0 i 1, gdzie 0 to kolor biały, a 1 to kolor czarny piksela.

Deklarujemy tablicę dwuwymiarową oraz jednowymiarową :

```
sgA: array [0..349, 0..349] of single;  
sgB: array [0..122499] of single;
```

następnie funkcję `zamienObrazNaTablice` realizującą „dekodowanie wykresu” oraz zapisanie wyniku do tablicy:

```
var  
i,j:integer;  
begin  
for i:=0 to 349 do  
begin  
for j:=0 to 349 do  
begin  
if  
GetRValue (image1.Canvas.Pixels[j,i])=0  
then  
begin  
sgA[i,j]:=1;  
end else  
sgA[i,j]:=0;  
end;  
end;  
end;  
end;
```

Kiedy mamy już wypełnioną tablicę dwuwymiarową danymi musimy ją przepisać do tablicy jedno wymiarowej ponieważ biblioteka FANN tego od nas wymaga zadanie to realizuje następująca funkcja `przepiszTablice`:

```
var  
i,j,m:integer;  
begin
```

```
m:=-1;  
for i:=0 to 349 do  
begin  
for j:=0 to 349 do  
begin  
m:=m+1;  
sgb[m]:=sga[j,i];  
end;  
end;  
end;
```

Kolejnym korkiem jest wygenerowanie pliku uczącego zawierające wszystkie dane uczące. Pierwszym krokiem jest stworzenie pliku na dysku z informacjami o ilości danych uczących oraz ilością wejść i wyjść sieci:

```
var  
TF : TextFile;  
begin  
AssignFile(TF,  
nazwapliku+'grupaMaterialow.data');  
If not  
FileExists(nazwapliku+'grupaMaterialow.d  
ata') then  
// sprawdzenie, czy plik istnieje  
begin  
ReWrite(TF);  
// jeżeli nie - stwórz  
Writeln(TF, edit1.text);  
//dane do zapisu  
end else  
begin  
ReWrite(TF); // jeżeli nie - stwórz  
Writeln(TF, edit1.text);  
//dane do zapisu  
end;  
CloseFile(TF)  
end;
```

Po utworzeniu pliku czas na zapisanie wszystkich danych uczących:

```
var  
i, j, k: Integer;  
Plik: TextFile;  
begin  
AssignFile(Plik,  
nazwapliku+'grupaMaterialow.data');  
Append(Plik);
```

```

for k:=0 to listBox1.Count-1 do // w
listbox'i zapisane są ścieżki do obrazów
wykresów
begin
image1.Picture.LoadFromFile(listbox1.Items[k]);
//otwieramy każdy obraz
zamienObrazNaTablice;
//"dekodujemy"
przepiszTablice;
//przepisujemy do tablicy
jednowymiarowej
for i:= 0 to 122499 do
//pętla realizuje zapisanie tablicy
sgb[] do pliku
begin

Write(Plik,floattostr(sgb[i])+ ' ');
end;
Writeln(Plik,#13);// przejście do
następczej linii
Writeln(Plik,listBox2.Items[k]);
end;
CloseFile(Plik);
end;

```

Aplikacja jest już prawie kompletna brakuje tylko fragmentu kodu odpowiadającego za tworzenie oraz uczenie sieci całość opisana jest poniżej:
 Tworzymy oraz uczymy sieć neuronową wykorzystując plik utworzony wcześniej:

```

var
mse,dost: single;
begin
fn.Build; // buduj sieć
fn.TrainOnFile(nazwapliku+'grupaMaterialow.data',100,0.0000001);
//trenuj sieć
mse:=fn.MSE;

lblMse.Caption:=Format('%%.16f',[mse]);
// wyświetl błąd w odpowiedzi sieci
end;

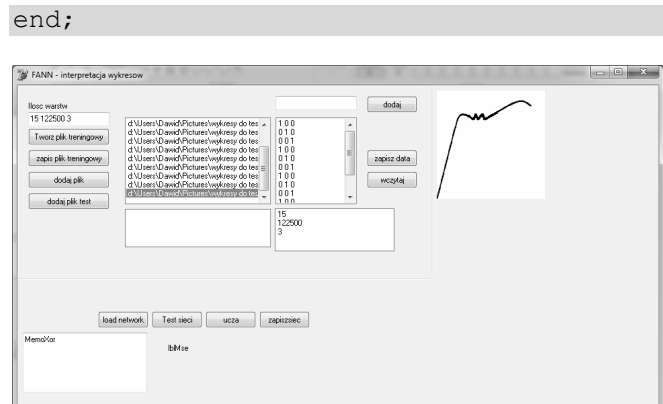
```

Kiedy wynik uczeni sieci jest optymalny zapisujemy sieć do pliku:

```

begin
fn.SaveToFile(nazwapliku+'FanngrupaMaterialow.dat')

```



Rysunek. 5.1 Aplikacja „FANN - interpretacja wykresów”.

Aplikacja „FANN-materialy”

Aplikacja ta tworzy trzy różne sieci neuronowe jednak opiszę przykład tworzenie pierwsze sieci ponieważ reszta odbywa się analogicznie. W przypadku tego programu danymi uczącymi są liczby odczytane z wykresu statycznej próby rozciągania dlatego nie zachodzi potrzeba „dekodowania” danych do postaci „rozumianej” przez sieć. Pamiętajmy, że komponent FANN 2.0 wykorzystuje inne zmienne środowiskowe niż system Windows z Polską lokalizacją, dlatego należy w opcjach regionu systemu symbol dziesiętny zmienić z przecinka na kropkę. Unikniemy w ten sposób problemów z kompilacją i działaniem programu.
 Pierwszą czynnością jest wczytanie danych uczących do komponentu StrinGrid realizuje tą czynność procedura Zpliku2StringGrid:

```

Procedure Zpliku2StringGrid(StringGrid:
TStringGrid; FileName: String);
var
File: TextFile;
Tile, i, j: Integer;
Tdane: string;
begin
AssignFile(File, FileName);
Reset(File);
Readln(File, Tile);
StringGrid.ColCount:= Tile;
Readln(File, Tile);
StringGrid.RowCount:= Tile;
for i:=0 to StringGrid.ColCount-1 do
for j:=0 to StringGrid.RowCount-1 do
begin
Readln(File, Tdane);

```

```
StringGrid.Cells[i,j]:= Tdane;
end;
CloseFile(File);
end;
```

poniżej procedura zapisująca dane do pliku:

```
procedure StringGrid2plik(StringGrid:
TStringGrid;FileName: String);
var
File: TextFile;
i, j: Integer;
begin
AssignFile(File, FileName);
Rewrite(File);
Writeln(File, StringGrid.ColCount);
Writeln(File, StringGrid.RowCount);
for i:=0 to StringGrid.ColCount-1 do
for j:=0 to StringGrid.RowCount-1 do
Writeln(File,
StringGrid.Cells[i,j]);
CloseFile(File);
end;
```

Kolejnym korkiem jest wygenerowanie pliku uczącego zawierające wszystkie dane uczące. Pierwszym krokiem jest stworzenie pliku na dysku z informacjami o ilości danych uczących oraz ilością wejść i wyjść sieci. Realizowane jest to tak samo jak w poprzedniej aplikacji dlatego nie będę już tego opisywał, różnica pojawia się podczas zapisu danych uczących do wcześniej utworzonego pliku:

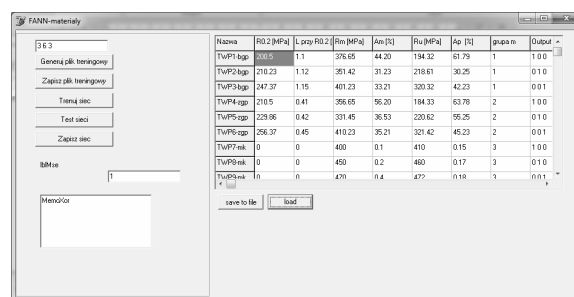
```
var
i, j, k: Integer;
Plik: TextFile;
begin
AssignFile(Plik,
nazwapliku+'material1.data');
Append(Plik);
for k:=1 to 3 do
begin
for i:= 1 to 6 do
begin
Write(Plik,stringgrid1.Cells[i,k]+' ');
end;
Writeln(Plik,#13);

Writeln(Plik,stringgrid1.Cells[8,k]);
end;
CloseFile(Plik);
```

```
end;
```

Tworzenie oraz uczenie sieci neuronowej wygląda tak samo jak w przypadku wcześniejszej aplikacji jednak dla uniwersalności programu zapis odbywa się za pomocą komponentu Savedialog i wygląda następująco:

```
begin
if savedialog1.Execute then
Fannmaterialy.SaveToFile(savedialog1.FileName);
end;
```



Rysunek. 5.2 Aplikacja „FANN-materialy”.

Aplikacja „FANN=>identification of the material”

Zajmiemy się teraz sednem tego artykułu, czyli aplikacją wykorzystującą wcześniej stworzone i wytrenowane sztucznie sieci neuronowe do rozpoznawania niektórych metali na podstawie wykresu statycznej próby rozciągania. Aplikacja wykorzystuje fragmenty kodów z wcześniej omówionych programów z tego względu poniżej opisane zostaną tylko nowe części kodu.

Najpierw tworzymy i wczytujemy nauczoną sieć:

```
begin
fn.Build;
fn.LoadFromFile(nazwapliku+'FanngrupaMaterialow.dat');
end;
```

rozpoczynamy interpretacji kształtu wykresu:

```
var
wejście: array [0..122499] of single;
output: array [0..2] of single;
i,m:integer;
begin
result.Clear; //czyścimy memo
```

```
result.Lines.Add('Result of the
identification chart: '); // wstawiamy
tekst
for m:=0 to 122499 do
begin
wejscie[m]:=sgb[m]; //przepisujemy
tablice zawierającą 0 i 1 po dekodowaniu
obrazu
end;
fn.Run(wejscie,output); // uruchamiamy
nauczoną sieć
for i:=0 to 2 do
begin
result.Lines.Add(Format('%.9f',[Output[i]
])); // wyświetlamy wynik
Identify2.Enabled:=true;
end;
// poniżej lista warunków po spełnieniu
jednego z nich zostaje uruchomiona
odpowiednia
// sieć neuronowa oraz zmienna
rodzajMaterialu i kategoria.
if (Output[0]>= 0.8) and
(Output[0]>Output[1])and
(Output[0]>Output[2]) then
begin
fn2.Build;
rodzajMaterialu:='bez wyraznej granicy
plastycznosci';
fn2.LoadFromFile(nazwapliku+'FannMateria
ly1.dat');
kategoria:='1';
end;
if (Output[1]>= 0.8) and
(Output[1]>Output[0])and
(Output[1]>Output[2]) then
begin
fn2.Build;
rodzajMaterialu:='kruchych';
fn2.LoadFromFile(nazwapliku+'FannMateria
ly3.dat');
kategoria:='2';
end;
if (Output[2]>= 0.8) and
(Output[2]>Output[0])and
(Output[2]>Output[1]) then
begin
fn2.Build;
rodzajMaterialu:='z wyrazna granica
plastycznosci';
```

```
fn2.LoadFromFile(nazwapliku+'FannMateria
ly2.dat');
kategoria:='3';
end;
end;
```

Po zidentyfikowaniu wykresu i wybrani odpowiedniej sieci rozpoznającej materiały następuje interpretacja danych odczytanych z wykresu:

```
var
wejscie: array [0..5] of single;
output: array [0..2] of single;

i,m:integer;
begin

result.Clear;
result.Lines.Add('Result of the
identification chart: ');
for m:=0 to 5 do
begin
wejscie[m]:=strtofloat(stringgrid1.Cells
[m+1,ComboBox1.ItemIndex+1]);
end;
fn2.Run(wejscie,output);
for i:=0 to 2 do
begin
result.Lines.Add(Format('%.9f',[Output[i]
]));
end;
memo1.Clear;
if (Output[0]>= 0.8) and
(Output[0]>Output[1])and
(Output[0]>Output[2]) then
begin
Material:='1';
end;
if (Output[1]>= 0.8) and
(Output[1]>Output[0])and
(Output[1]>Output[2]) then
begin
Material:='2';
end;
if (Output[2]>= 0.8) and
(Output[2]>Output[0])and
(Output[2]>Output[1]) then
begin
Material:='3';
end;
```




Rysunek. 6.1 Przykład wykresów - dane uczące zastosowane podczas uczenia sieci.

Nazwa	R0.2 [MPa]	L przy R0.2 [%]	Rm [MPa]	Am [%]	Rlu [MPa]	Ap [%]	Output
TWP1	200.5	0.33	376.65	44.20	194.32	61.79	1 0 0
TWP2	49.85	0.4	321.42	22.23	218.61	29.25	1 0 0
TWP3	247.37	1.1	401.23	33.21	320.32	42.23	1 0 0
TWP4	200.5	0.33	376.65	44.20	194.32	61.79	0 1 0
TWP5	49.86	0.4	321.42	22.23	218.61	29.25	0 1 0
TWP6	247.37	1.1	401.23	33.21	320.32	42.23	0 1 0
TWP7	0	0	400	0.1	410	0.15	0 0 1
TWP8	0	0	450	0.2	460	0.17	0 0 1
TWP9	0	0	470	0.4	472	0.18	0 0 1

Rysunek. 6.2 Przykład danych odczytanych z wykresu - dane uczące zastosowane podczas uczenia sieci.

Algorytm uczenia.

Dobór metody uczenia sieci neuronowej jest ważnym elementem, który ma istotny wpływ na działanie sieci. Wybrana metoda została dobrana doświadczalnie i jest jedną z wielu dostępnych w bibliotece FANN. Metoda QUICKPROP powoduje znaczne przyspieszenie procesu uczenia jednocześnie zabezpiecza on przed „wpadnięciem” w płytkie minimum lokalne.

7. PODSUMOWANIE

Taka budowa programu daje nam możliwość inteligentnego rozpoznawania metalu na podstawie wykresy statycznej próby rozciągania. Skuteczność programu zależy od ilości danych uczących jednak istnieje możliwość dalszego uczenia sieci co zwiększa możliwości tego narzędzia. Jak widać zastosowanie kilku sieci pozwala porównać program do działania bardzo prostego modelu sztucznego mózgu ponieważ każda z sieci realizuje oddzielne zadanie, jednak wynik końcowy zależy od kolektywnego działania wszystkich sieci.

Jak widać sztuczne sieci neuronowe można z powodzeniem wykorzystać do klasyfikacji danych eksperymentalnych.

Literatura

1. J. Żurada, M. Barski, W. Jędruch: Sztuczne sieci neuronowe. Podstawy teorii i zastosowania, Wydawnictwo PWN, Warszawa 1996.
2. A. Bochenek, M. Reicher: Anatomia Człowieka, tom IV (Układ nerwowy ośrodkowy), PZWL, Warszawa 2000.
3. J. Korbicz, A. Obuchowicz, D. Uciński: Sztuczne sieci neuronowe - Podstawy i zastosowania, Akademicka Oficyna Wydawnicza PLJ, Warszawa 1994.
4. T. Master: Sieci neuronowe w praktyce, Wydawnictwa Naukowo-Techniczne, Warszawa 1996.
5. PN – EN 10002 - 1:1998. Próba statyczna rozciągania metali.
6. Opis biblioteki FANN 2.0 [online]. [dostęp: 25 marzec 2011].
7. Dostępny w Internecie <http://leenissen.dk/fann/wp/>
7. Uczenie sieci neuronowych [online]. [dostęp: 25 kwiecień 2011]. Dostępny w Internecie http://kopernik.matfiz.polsl.pl/www/Marcin.Wozniak/inne%5Csieci_neuronowe/uczenie

KONCEPCJA BAZY DANYCH JAKO PODSTAWOWEJ CZĘŚCI PROGRAMU GENERUJĄCEGO OCENY OPISOWE W NAUCZANIU WCZESNOSZKOLNYM

Arleta Trzeciakowska

Uniwersytet Kazimierza Wielkiego
Instytut Techniki
II rok MU Edukacja Techniczno-Informatyczna
ul. Chodkiewicza 30, 85-064 Bydgoszcz
e-mail: arletatrzo2.pl

Streszczenie: Szeroki dostęp do Internetu, istnienie ogromnej ilości tekstów w wersji elektronicznej powoduje konieczność rozwoju nauki określanej jako inżynieria lingwistyczna. Zajmuje się ona szeroko pojętym przetwarzaniem danych lingwistycznych. Jednym z aspektów przetwarzania tego rodzaju danych jest generowanie tekstów w języku naturalnym. Ponieważ przeważająca ilość powstających tekstów dostępna jest w wersji elektronicznej, istnieje bardzo duże zapotrzebowanie na programy przetwarzające je. Głównym celem powstania tego artykułu jest przedstawienie koncepcji relacyjnej bazy danych będącej podstawą eksperymentalnego programu automatycznie generującego oceny opisowe w nauczaniu wczesnoszkolnym.

Słowa kluczowe: Komputerowe przetwarzanie tekstów, NLP, generowanie tekstów w języku naturalnym, ocena opisowa

Conception of a database as fundamental part of the program generating the descriptive grades in elementary schools

Abstract: Common access to the Internet and huge number of the texts in numeric version causes necessity of progress of the science known as linguistic engineering. It researches the wide implied natural language processing. One of the aspects of processing that kind of data is generating the texts in the natural language. Because the most of the nascent texts are available in numeric version, there is large demand for the programs processing them. The main point of that article is to present the conception of a database that is the fundamental part of the experimental program automatically generating descriptive grades in elementary schools.

Keywords: NLP, natural language processing, generating texts in natural language, descriptive grades.

1. WPROWADZENIE

Komputerowe generowanie tekstów w języku naturalnym jest jednym z zagadnień, którymi zajmuje się inżynieria lingwistyczna LE (ang. linguistic engineering). Niemal równoważnie funkcjonują inne określenia tej dziedziny, jak komputerowe przetwarzanie języka naturalnego NLP (ang. natural language processing), lub technologia języka LT (ang. language technology).

Definicja:

Inżynieria lingwistyczna to opracowanie metod i konstruowanie narzędzi umożliwiających automatyczne przetwarzanie języka naturalnego.

2. NLP JAKO INTERDYSCYPLINARNA DZIEDZINA NAUKI

NLP jest interdyscyplinarną dziedziną nauki. Łączy zagadnienia z zakresu m.in. informatyki, sztucznej

inteligencji, matematyki, filozofii, psychologii, nauki o języku.

Językoznawstwo (lingwistyka) zawiera wiele dziedzin, np.:

- **syntaktykę, która zajmuje się formatem, regułami budowy i formalną składnią,**
- **morfologię, zajmującą się morfemami-elementarnymi jednostkami morfologii, formami odmiennej części mowy (fleksja) oraz słotwórstwem,**
- **semantykę, której zadaniem jest określenie znaczenia wyrażen i konstrukcji języka, zrozumiałego często tylko w określonym kontekście,**
- **gramatykę – naukę o zasadach budowy i odmiany wyrazów oraz o regułach składni danego języka,**
- **wiedzę pragmatyczną, dotyczącą znaczenia w szerokim kontekście wiedzy o świecie.**

3. ROZWÓJ INŻYNIERII LINGWISTYCZNEJ

O ile rozwój informatyki w dziedzinie przetwarzania danych liczbowych, czy zbiorów danych ustrukturalizowanych następuje bardzo dynamicznie i przynosi bardzo dobre rezultaty, to efekty prac nad przetwarzaniem języka naturalnego są wciąż dalekie od oczekiwanych.

Wynika to stąd, że możliwości systemów komputerowych są wciąż bardzo niewielkie wobec oczekiwań, aby maszyna posługiwała się językiem naturalnym jak człowiek.

Problem stanowią cechy języka naturalnego, złożoność procesu inteligentnego posługiwania się nim, co w znacznym stopniu utrudnia lub w niektórych aspektach uniemożliwia efektywne przetwarzanie komputerowe. Wiele trudności wynika ze specyficznych cech języka polskiego takich jak :

- **wieloznaczność (to samo słowo może mieć wiele znaczeń),**
- **niejednoznaczności (ten sam tekst może mieć wiele interpretacji),**
- **możliwość sformułowania tych samych informacji wieloma sposobami, przy użyciu różnych konstrukcji zdaniowych, stosowanie wyrazów bliskoznacznych,**
- **istnienie różnych relacji (powiązań) między wyrazami,**
- **konieczność wykorzystania bardzo rozległej wiedzy lingwistycznej,**

- **bogactwo języka, ciągła ewolucja, nieostre reguły poprawności.**

Wciąż nierozwiązanym problemem jest konieczność korzystania z ogólnej wiedzy o świecie i szczegółowych informacji o danej dziedzinie, zwłaszcza w fazie komputerowej analizy tekstu, w celu jego „rozumienia” przez maszynę.

Formalny opis poszczególnych aspektów języka jest bardzo szeroki i złożony. Z tego względu korzystanie z niego przy budowanie programów komputerowych jest znacznie utrudnione i powoduje ich mniejszą efektywność. Dlatego twórcy programów stosują często bardzo pobieżną analizę językową.

Inną metodą stosowaną w przetwarzaniu języka naturalnego jest korzystanie z modeli języka opartych na danych statystycznych. Również ten sposób jest niedoskonały, ponieważ bardzo trudne jest zgromadzenie dużych zbiorów do budowy modeli statystycznych. Modele te nie biorą pod uwagę wszystkich cech języka, co wiąże się z niewystarczająco dobrymi efektami.

4. PRZYKŁADY PROGRAMÓW PRZETWARZAJĄCYCH JĘZYK NATURALNY

Pomimo trudności powstało wiele programów w dziedzinie NLP. Są wśród nich programy mające znaczenie eksperymentalne, ale także wykorzystywane na co dzień. Są to np.:

- **edytory tekstów,**
- **systemy dialogowe,**
- **wyszukiwarki internetowe,**
- **korektory pisowni,**
- **systemy wspomagające tłumaczenie,**
- **wyszukiwanie i ekstrakcja informacji,**
- **przetwarzanie tekstu na mowę,**
- **automatyczne generowanie streszczeń tekstów,**
- **wnioskowanie na podstawie danych tekstowych,**
- **klasyfikacja dokumentów tekstowych**

5. KOMPUTEROWE GENEROWANIE TEKSTÓW

Jednym z elementów komputerowego przetwarzania tekstów jest generowanie tekstów w języku naturalnym. O ile analiza tekstów napotyka na wiele trudności w związku ze złożonością tego procesu i wciąż jeszcze niewystarczającym postępem nauki w tej dziedzinie, to

generowanie tekstów w wielu przypadkach okazuje się zadaniem o wiele prostszym.

Stopień trudności zależy tu do przeznaczenia programu, stopnia złożoności i dynamiczności mającego powstać tekstu, rozległości dziedziny, której dotyczy.

Istnieje szereg zastosowań, dla których złożoność stosowanych rozwiązań i stopień trudności tworzenia aplikacji jest stosunkowo niewielki. W tego typu programach tworzonych zwłaszcza dla konkretnych, niezbyt szerokich zastosowań nie jest ważna kompletność słownika, jak w przypadku analizy tekstu. Podstawowym problemem jest wybór odpowiedniego zasobu słów i możliwych do zastosowania konstrukcji językowych.

Ważnym etapem pracy nad generatorem tekstu jest odpowiednie zaplanowanie treści wypowiedzi oraz jej postaci. Wiąże się to z dobraniem właściwych konstrukcji językowych, wyborem odpowiednich słów i powiązań.

Możliwe jest zastosowanie różnych rozwiązań w zależności od oczekiwanych efektów działania konkretnego programu. Podstawą jego działania mogą być gotowe fragmenty tekstów, wzorce z miejscami do uzupełniania, budowanie zdań z pojedynczych słów, tworzenie tekstu z gotowych zdań.

Tak więc stopień trudności w tworzeniu programów generujących teksty jest bardzo zróżnicowany.

6. OCENA OPISOWA W NAUCZANIU Wczesnoszkolnym

W nauczaniu wczesnoszkolnym (kl. 1-3 szkoły podstawowej) ocena końcoworoczna ma formę opisową. Składa się na nią ocena stopnia opanowania wiadomości i umiejętności w zakresie poszczególnych treści programowych oraz ocena zachowania. Jest ona formułowana indywidualnie dla każdego ucznia.

Po przeanalizowaniu ocen zredagowanych przez wielu nauczycieli nasuwa się wniosek, że pomimo indywidualizacji, zdania oceniające opanowanie wiedzy i umiejętności powtarzają się często w bardzo podobnej, lub takiej samej formie.

Wynika to między innymi z tego, że w klasie zwykle są grupy uczniów osiągających bardzo zbliżone wyniki w danej dziedzinie. Poza tym ocena opisowa w pewnym stopniu przybiera postać dokumentu, który musi być zredagowany za pomocą krótkich, w miarę możliwości prostych zdań dotyczących oceny w zakresie konkretnej dziedziny objętej programem nauczania.

W znacznym stopniu jest to konsekwencją ograniczonego miejsca przeznaczonego na ocenę na świadectwie szkolnym. Jest to kilkanaście linii na ocenę stopnia

opanowania wiadomości i umiejętności z zakresu języka polskiego, matematyki, środowiska społeczno-przyrodniczego, wychowania fizycznego, plastyki, techniki, muzyki, informatyki i języka obcego. Na ocenę zachowania przeznaczono kilka linii.

7. KONCEPCJA PROGRAMU AUTOMATYCZNIE GENERUJĄCEGO OCENY OPISOWE

Opisane wyżej uwarunkowania sprawiają, że nauczyciele podczas redagowania ocen często korzystają z przygotowanych wcześniej sformułowań, gotowych fraz, lub całych zdań uwzględniających osiągnięcia uczniów w ramach każdej z dziedzin.

Stanowi to podstawę pomysłu, aby zautomatyzować ten proces za pomocą programu komputerowego.

Istnieją już programy komputerowe wspomagające pisanie ocen. Ułatwienia, jakie ich twórcy proponują nauczycielom, to przede wszystkim zbiory gotowych zdań, pogrupowane w odniesieniu do każdej umiejętności, którą dziecko powinno opanować.

Przykłady zbiorów zdań z tego typu programów:

Ocena wypowiedzi w zakresie języka polskiego:

- ☐ Posiada ubogi zasób słownictwa.
- ☐ Potrafi opowiadać o wydarzeniach zawartych w tekstach literackich.
- ☐ Potrafi wypowiadać się na podstawie obrazka, w uporządkowanej formie.
- ☐ Samodzielne wypowiedzi są nieuporządkowane.
- ☐ Stosuje poprawne formy gramatyczne.
- ☐ W kilku zdaniach wypowiada się na konkretny temat.
- ☐ wersja demo.
- ☐ Wypowiada się poprawnie, ale najchętniej na swobodne tematy.
- ☐ Wypowiedzi są kilkuzdaniowe i poprawnie zbudowane.
- ☐ Wypowiedzi są zwięzłe i komunikatywne.
- ☐ Zapomina o stosowaniu poprawnych form gramatycznych.

Rysunek. 1 Przykładowy formularz programu wspomagającego ocenę opisową w zakresie wypowiedzi.

Ocena umiejętności z zakresu wychowania fizycznego:

Arleta Trzeciakowska, Koncepcja bazy danych jako podstawowej części programu generującego oceny opisowe w nauczaniu wczesnoszkolnym

☐	Chętnie uczestniczy w zespołowych grach sportowych.
☐	Doskonale radzi sobie w grach z piłką i z wszystkimi ćwiczeniami sprawnościowymi.
☐	Dziecko jest sprawnie fizycznie, ćwiczenia wykonuje z dużym zapalem.
☐	Jest zwinne i poprawnie wykonuje ćwiczenia gimnastyczne.
☐	Lubi zajęcia sportowe, jednak mało dokładnie wykonuje ćwiczenia gimnastyczne.
☐	wersja demo.
☐	Precyzyjnie i celowo wykonuje ruchy.
☐	Trudność sprawia mu wykonywanie podstawowych ćwiczeń gimnastycznych.
☐	W czasie zajęć sportowych jest zdyscyplinowane, dba o bezpieczeństwo kolegów i koleżanek.

Rysunek. 2 Przykładowy formularz programu wspomagającego ocenę opisową w zakresie wychowania fizycznego.

Ocena umiejętności w zakresie czytania i pracy z tekstem:

☐	SM ma trudności z rozumieniem przeczytanych wyrazów i poleceń.
☐	SM nie potrafi odpowiedzieć na proste pytania nauczyciela związane z tekstem.
☐	Pomimo pomocy nauczyciela SM nie potrafi uporządkować wydarzeń, które zostały opisane w tekście.
☐	SM nie czyta ze zrozumieniem.
☐	SM ma duże trudności w rozumieniu czytanej instrukcji.
☐	SM ma trudności z odpowiedzią na proste pytania związane z przeczytanym tekstem. Nie rozumie czytanych przez siebie poleceń.
☐	SM nie potrafi czytać wykresów i map, nie rozumie ich znaczenia.
☐	SM ma kłopoty ze zrozumieniem czytanego tekstu.
☐	SM czyta głośno ze zrozumieniem tekstu.
☐	SM czyta ze zrozumieniem po cichu i głośno wyrazi, polecenia oraz różnorodne proste teksty w postaci prostych zdań.
☐	SM potrafi poprawnie odpowiedzieć na proste pytania dotyczące przeczytanego tekstu.
☐	SM rozumie przeczytany tekst.
☐	SM potrafi na ogół czytać ze zrozumieniem.
☐	SM umie ocenić, czy przeczytany utwór jest poważny, żartobliwy czy smutny.
☐	SM potrafi wymienić postacie występujące w utworze.
☐	SM potrafi przedstawić treść opowiadania zgodnie z kolejnością wydarzeń.
☐	SM potrafi odpowiedzieć na pytania dotyczące przeczytanego samodziennie tekstu.
☐	SM czyta po cichu i głośno teksty oraz kierowany pytaniami nauczyciela potrafi skonstruować kilka zdań na temat przeczytanego tekstu.
☐	SM czyta cicho ze zrozumieniem.
☐	SM czyta ze zrozumieniem.
☐	SM samodzielnie układa pytania i odpowiedzi dotyczące przeczytanego tekstu.
☐	SM kierowan[ą/y] pytaniami nauczyciela potrafi wypowiedzieć kilka zdań na temat przeczytanego tekstu.
☐	SM samodzielnie potrafi ułożyć pytania do przeczytanego cicho tekstu.
☐	SM po przeczytaniu krótkiego tekstu potrafi wypowiedzieć się na jego temat.
☐	SM potrafi samodzielnie ocenić postępowanie bohaterów tekstu literackiego.
☐	SM czyta ze zrozumieniem zarówno głośno jak i cicho.
☐	SM potrafi interpretować dane przedstawione za pomocą wykresów i tabel.

Rysunek. 3 Przykładowy zestaw fraz w programie wspomagającym ocenę opisową w zakresie czytania i pracy z tekstem.

Zaletą tego typu programów jest nie tylko możliwość wyboru zdań zróżnicowanych pod względem stylistycznym, ale również bardziej precyzyjna ocena.

Często jednak teksty powstające z tak szczegółowych zdań nie mieszczą się na świadectwie szkolnym.

Przeanalizowanie istniejących programów skłoniło mnie to podjęcia próby stworzenia eksperymentalnego programu całkowicie automatycznie generującego ocenę opisową dla klasy pierwszej. Niewątpliwie bardzo pomocne okazało się moje wieloletnie doświadczenie zawodowe jako nauczyciela w nauczaniu wczesnoszkolnym.

Rozwiązanie praktyczne zastosowane w generatorze ocen opisowych opiera się na formularzach utworzonych w języku HTML. Kod programu został utworzony w PHP, a fragmenty kodu obsługujące bazę danych (MySQL) w języku SQL.

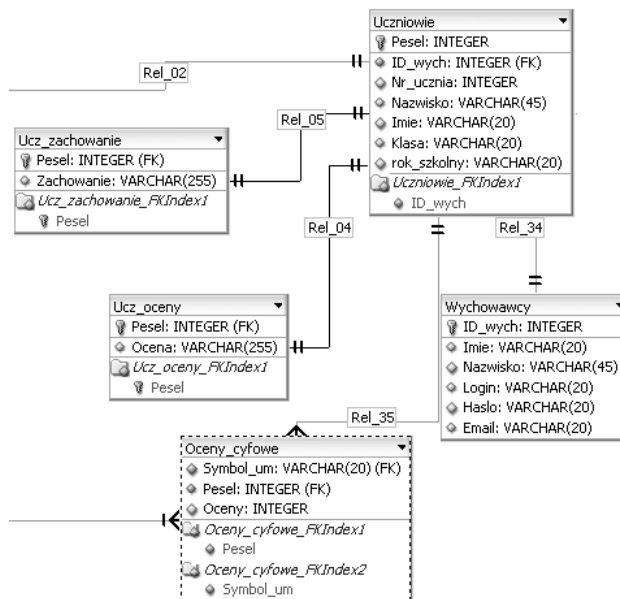
Na formularzach wyszczególnione są wiadomości i umiejętności, które podlegają ocenie. Obok każdej z nich nauczyciel powinien wpisać ocenę cyfrową w skali 1-6.

Taka skala została przyjęta, ponieważ w taki sposób odnotowywane są oceny cząstkowe w dzienniku szkolnym. Podobnie jest w przypadku oceny zachowania. Na formularzu wymienione są najważniejsze oceniane elementy zachowania:

wykonywanie podjętych zadań
aktywność podczas lekcji
stosunek do obowiązków szkolnych
umiejętność współdziałania w grupie
przestrzeganie zasad bezpieczeństwa

8. BAZA DANYCH

Podstawą kreatora ocen jest relacyjna baza danych. Tworzą ją tabele połączone zgodnie z zasadami budowy tego typu baz. Kilka tabel zawiera dane o wychowawcach i uczniach. W tabeli *Wychowawcy* znajdują się dane umożliwiające logowanie zarejestrowanych wychowawców, którzy mają dostęp wyłącznie do informacji o swojej klasie i tylko w tym zakresie mogą dokonywać zmian i wprowadzać dane.



Rysunek. 4 Fragment relacyjnej bazy danych.

Powiązane są z nimi tabele, które służą do przechowywania gotowych ocen opisowych osiągnięć dydaktycznych i zachowania każdego ucznia (*Ucz_ocen*, *Ucz_zachowanie*). Informacje o poszczególnych przedmiotach znajdują się w tabelach:

- *Przedmioty*, która przechowuje nazwy przedmiotów i ich identyfikatory (klucz główny),

ID_przedmiotu	Nazwa
jp	j_polski
mat	matematyka
srod	środowisko
pl_tech	plastyka_technika
muz	muzyka
wf	wych_fizyczne
jo	j_obcy
inf	informatyka
zach	zachowanie

Rysunek. 5 Tabela *Przedmioty*.

- *Lista_przedmiotów* wiążąca tabelę *Przedmioty* z tabelą *Uczniowie* (przechowuje klucze główne tych tabel)
- *Przedmiot_umiejetnosci* (wyszczególnione są tu określenia poszczególnych umiejętności, które wyświetlane są na formularzu i podlegają ocenie nauczyciela).

Symbol_um	ID_przedmiotu	Opis_um
jp_w	jp	wypowiedzi
jp_cz	jp	czytanie
jp_pis	jp	pisanie
mat_l	mat	dodawanie i odejmowanie w zakresie 10
mat_z	mat	rozwiązywanie zadań tekstowych
srod_przyr	srod	znajomość otaczającego środowiska przyrodnicze...
srod_obs	srod	umiejętność prowadzenia obserwacji środowiska
srod_ochr	srod	znajomość form ochrony środowiska
plast_tech_tw	plast_tech	twórcze realizowanie zadań plastycznych i techn...
plast_tech_techn	plast_tech	stosowanie różnorodnych technik
muz_spiew	muz	śpiew
muz_rytm	muz	poczucie rytmu
wf_sprawn	wf	sprawność w wykonywaniu ćwiczeń
wf_udzial	wf	zaangażowanie w zabawy ruchowe

jo_um	jo	umiejętności w zakresie języka angielskiego
inf_um	inf	umiejętności w zakresie posługiwania się komputerem
zach_1	zach	podejmowanie zadań i obowiązków
zach_2	zach	wykonywanie podjętych zadań
zach_3	zach	aktywność podczas lekcji
zach_4	zach	stosunek do obowiązków szkolnych
zach_5	zach	umiejętność współdziałania w grupie
zach_6	zach	przestrzeganie zasad bezpieczeństwa

Rysunek. 6 Tabela *Przedmiot_umiejetnosci*.

Ponieważ język polski obejmuje znacznie więcej elementów, które należy ocenić niż pozostałe przedmioty, dlatego powstała osobna tabela *Jp_umiejetnosci* z określeniami umiejętności ocenianych w zakresie wypowiedzi, czytania i pisania.

ID_um	Symbol_um	Opis
jp_w2	jp_w	forma wypowiedzi
jp_w1	jp_w	zasób słownictwa
jp_pis6	jp_pis	samodzielne pisanie prostych zdań
jp_pis5	jp_pis	pisanie odpowiedzi na proste pytania
jp_pis4	jp_pis	pisanie wyrazów z pamięci
jp_pis3	jp_pis	Przepisywanie
jp_pis2	jp_pis	estetyka pisma
jp_pis1	jp_pis	poprawność pisania liter
jp_cz6	jp_cz	czytanie wyraziste
jp_cz5	jp_cz	czytanie biegle
jp_cz4	jp_cz	czytanie płynne
jp_cz3	jp_cz	czytanie wyrazów i prostych zdań
jp_cz2	jp_cz	czytanie sylab
jp_cz1	jp_cz	rozpoznawanie liter

Rysunek. 7 Tabela *Jp_umiejetnosci*.

W bazie danych jest grupa tabel, w których przechowywane są słowa, frazy i całe zdania, które po wystawieniu ocen cyfrowych służą do wygenerowania oceny opisowej.

Trzy spośród nich, *Wypowiedzi*, *Pisanie* i *Czytanie* zawierają słownictwo do redagowania oceny z języka polskiego.

Pozostałe przedmioty obejmują mniej elementów ocenianych, dlatego dla każdego z nich istnieje jedna tabela ze słownictwem. Są to tabele: *Matematyka*, *Srodowisko*, *Język_obcy*, *Informatyka*, *Zachowanie*.

Arlęta Trzeciakowska, *Koncepcja bazy danych jako podstawowej części programu generującego oceny opisowe w nauczaniu wczesnoszkolnym*

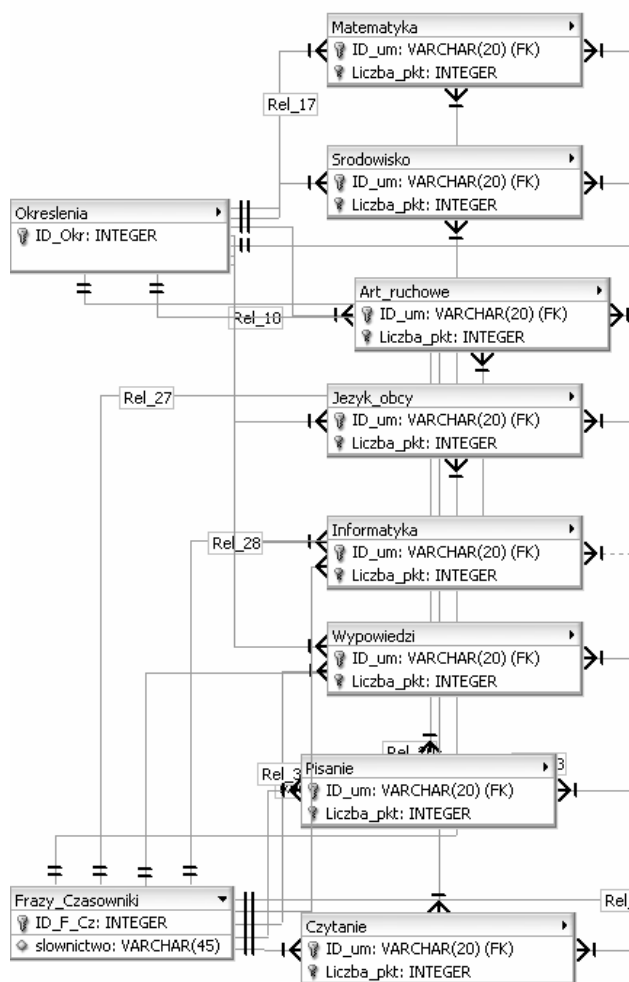
Słownictwo potrzebne do zredagowania oceny umiejętności z zakresu plastyki, muzyki, techniki i wychowania fizycznego umieszczone jest w jednej tabeli: *Art_ruchowe*.

Podczas dokonywania wyboru odpowiedniego słownictwa i konstrukcji zdań dla generatora ocen, okazało się, że wiele wyrazów i całych fragmentów zdań powtarza się. Dotyczy to głównie nazw czynności, wyrazów określających stopień opanowania umiejętności (głównie przymiotników i przysłówków) oraz różnej długości fraz. Aby uniknąć zbyt dużej nadmiarowości bazy danych wprowadzone zostały dwie dodatkowe tabele zawierające to słownictwo: *Okreslenia* oraz *Frazy_Czasowniki*.

ID	F	Cz	słownictwo
1			Posługuje się
2			posługuje się
3			Wypowiada się w formie
4			Wypowiada się
5			czyta
6			Litery i cyfry pisze
7			Bezbłędnie przepisuje
8			dodaje i odejmuje w zakresie 10.
9			rozwiązuje zadania tekstowe
10			środowisku przyrodniczym.
11			środowisko przyrodnicze.
12			ochrony środowiska
13			Prace plastyczno-techniczne wykonuje
14			Śpiewa poznane piosenki
15			sprawnność ruchową
16			uczestniczy w zabawach ruchowych.
17			wyniki w zakresie języka angielskiego.
18			Posługuje się komputerem
19			podejmuje zadanie i obowiązki,
20			je wykonuje.
21			bierze udział w lekcji.
22			wywiązuje się z obowiązków szkolnych.
23			współpracuje w grupie.
24			przestrzega zasad bezpieczeństwa.

Rysunek. 8 Tabela *Frazy_Czasowniki*.

Te dwie dodatkowe tabele powiązane są z każdą z tabel zawierających słownictwo do redagowania zdań oceniających w zakresie poszczególnych przedmiotów. Klucz główny w tabeli *Frazy_Czasowniki* (*ID_F_Cz*) i w tabeli *Okreslenia* (*ID_Okr*) są kluczami obcymi w tabelach odnoszących się do poszczególnych przedmiotów (np.: *Matematyka*, *Srodowisko*, *Czytanie*).



Rysunek. 9 Fragment relacyjnej bazy danych, powiązania między tabelami przechowującymi słownictwo.

Tabele takie jak np. *Czytanie* mają dwie kolumny przeznaczone na przechowywanie słownictwa, tak aby słowa i frazy z dodatkowych tabel mogły być wstawiane nie tylko na początku i na końcu zdania, ale również w środku. Sposób budowania zdań z użyciem wyrazów z tabel *Okreslenia* i *Frazy_Czasowniki* przedstawione zostaną na podstawie wybranej z tabeli *Jp_umijetnosci* umiejętności o kluczu głównym *jp_cz2* (*czytanie sylab*):

ID_um	Symbol_um	Opis
jp_cz2	jp_cz	czytanie sylab

Następnie na podstawie tabeli *Czytanie*, w zależności od wystawionej oceny cyfrowej znajdującej się w kolumnie *Liczba_pkt* wybrane zostaje odpowiednie słownictwo dla każdej z ocen.

ID_um	Liczba_pkt	ID_F_Cz	ID_Okr	Ocena opis 1	Ocena opis 2
jp_cz2	6543	0	0		
jp_cz2	2	5	0	Z trudnością	syllabami.
jp_cz2	1	0	3	Nie czyta	prostych wyrazów i zdań.

Rysuek. 10 Część tabeli *Czytanie*.

Jeśli wystawiona zostaje ocena 6, 5, 4, lub 3 zdanie nie jest redagowane, ponieważ oznacza to, że dziecko posiada umiejętności na wyższym poziomie, np. czyta całymi wyrazami lub zdaniami. Odpowiednie zdanie oceniające będzie wygenerowane przy okazji umiejętności na wyższym poziomie(jp_cz3, jp_cz4, jp_cz5, jp_cz6,):

ID_um	Symbol_um	Opis
jp_cz3	jp_cz	czytanie wyrazów i prostych zdań
jp_cz4	jp_cz	czytanie płynne
jp_cz5	jp_cz	czytanie biegle
jp_cz6	jp_cz	czytanie wyraziste

Rysuek. 11 Część tabeli *Czytanie*.

Gdy *Liczba_pkt* wynosi 2 (oceną będzie 2), *ID_F_Cz* w tym wierszu to 5(jest to identyfikator słowa *czyta*). Zdanie oceniające będzie następujące:

Z trudnością czyta sylabami.

Gdy *Liczba_pkt* wynosi 1 (oceną będzie 1), *ID_Okr* w tym wierszu to 3(identyfikator wyrażenia *na ogół*). Zdanie oceniające będzie następujące:

Nie czyta na ogół prostych wyrazów zdań.

Przy pomocy języka SQL i skryptów PHP podobnie buduje się pozostałe zdania.

9. ODSUMOWANIE

W realizacji takiej koncepcji generatora ocen opisowych budowa bazy danych i odpowiednia organizacja słownictwa ma podstawowe znaczenie. Wybrana w tym przypadku opcja budowania zdań zarówno z pojedynczych wyrazów, fraz i całych zdań jest kompromisem pomiędzy nadmiarowością bazy danych, a zbyt dużym stopniem

skomplikowania kodu i konieczności wykorzystania złożonej wiedzy lingwistycznej.

Pomimo intensywnego rozwoju inżynierii lingwistycznej, stopień jej rozwoju jest wciąż zbyt niski w porównaniu z potrzebami i oczekiwaniami. Dlatego warto tworzyć proste programy użytkowe ułatwiające i przyspieszające pracę nad redagowaniem tekstów w języku naturalnym mające zastosowanie w różnych dziedzinach życia.

Literatura

1. Mykowiecka A., "Inżynieria lingwistyczna, Komputerowe przetwarzanie tekstów w języku naturalnym", Wydawnictwo PJWSTK, Warszawa 2007
2. Bolc L., Cichy M., Różańska L., „Przetwarzanie języka naturalnego”, Wydawnictwa Naukowo-Techniczne, Warszawa 1982
3. Rojek-Mikołajczak I., "Bazy danych, kurs podstawowy dla inżynierów informatyków", Wydawnictwo Akademii Bydgoskiej im. Kazimierza Wielkiego, Bydgoszcz 2004
4. Jakubowski A., "Podstawy SQL, ćwiczenia praktyczne", Helion 2001

Arleta Trzeciakowska, Koncepcja bazy danych jako podstawowej części programu generującego oceny opisowe w nauczaniu wczesnoszkolnym

PROJEKTOWANIE MECHANIZMÓW I UKŁADÓW ANDROIDALNYCH NA PODSTAWIE STEROWANEGO KOMPUTEROWO MANIPULATORA ANDROARM002

Zbigniew Szczepański

Uniwersytet Kazimierza Wielkiego
Instytut Mechaniki i Informatyki Stosowanej
ul. Kopernika 1 p.207a, 85-074 Bydgoszcz
e-mail: zszczep@ukw.edu.pl

Streszczenie: *Streszczenie: W pracy zaprezentowano konstrukcję manipulatora, który ma stanowić alternatywne rozwiązanie dla kosztownych manipulatorów, znanych z komercyjnych zastosowań. Zwykle mechanizmy jako podzespoły robotów służą do jednego, ściśle określonego celu, jakim może być np. malowanie, spawanie, szycie, montaż, mycie, itp. Niektóre roboty, aby mogły wykonywać pewne operacje, muszą być specjalnie konstruowane lub przekonstruowane. W rzeczywistości dobiera się najbardziej korzystne rozwiązanie dla wykonywania danej funkcji. Niestety, kiedy funkcja ta przestaje być przydatna, robot staje się bezużyteczny. Autor pracy zakłada, że szersze stosowanie manipulatorów androidalnych, podobnych w swej budowie do opisywanego w pracy manipulatora androarm002 (rys poniżej), może przyczynić się do uniknięcia tego typu problemów. Manipulatory androidalne mogą stanowić w niedalekiej przyszłości jedną z ważniejszych gałęzi robotyki.*

Słowa kluczowe: *Projektowanie, robotyka, androarm, CAD, SolidWorks, Sterowanie, Manipulator*

Projecting and constructing of a robot arm mechanisms controlled by a computer module - project AndroArm002

Abstarct: *Abstract: In this article the robot arm construction was presented, this is an alternative idea for expensive robots known from their commercial uses. Usually this type of standard robots are used to a closely definite aim, e.g. painting, welding, sewing, assembly, etc Standard robots aimed to do some special operations have to be reconstructed by a special way. In real, the most advantageous solution for a special function is chosen. When the functions becomes less useful, the robot is useless too. Author assumes that wide application of robot arms with construction, similar to (mentioned in the article) construction androarm002 (fig. below) is able to help avoid similar problems. In near future constructions similar to project - AndroArm002 can be important section of robotics.*

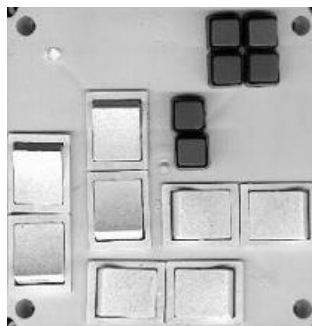
Keywords: *Keywords: Design, Robotics, androarm, CAD, SolidWorks, Control, Manipulator*

1. WSTĘP

Zaproponowano w ramach pracy zaprojektowanie i skonstruowanie robota, w którego skład wchodzi: komputer klasy PC, moduł kontrolny i manipulator androidalny, przypominający wyglądem i funkcjami ludzkie ramię. Stosowanie w wielu sytuacjach dnia codziennego podstawowych funkcji, które realizuje „mechanizm” ludzkiej ręki, jest konieczne i uzasadnione. Człowiek nie zawsze jest w gotowości, aby wykonać niebezpieczne zadanie, bądź pracować w szkodliwych warunkach. Dlatego stworzenie zastępczego systemu imitującego ruch niektórych kończyn człowieka i wprowadzenie ich w ruch stało się tematem niniejszej pracy. Zastępowanie ich mechanizmami wzorowanymi na ruchach ludzkich kończyn decyduje o ułatwieniu i poprawie efektywności pracy, które musiałaby wykonać ludzka ręka, bądź całe ramię. Istotnymi aspektami brany pod uwagę przy budowie robota systemu androarm powinny być użyteczność i funkcjonalność ludzkich kończyn, których budowa i funkcje są wynikiem wielu milionów lat ewolucji. Dodatkowym argumentem za podjęciem pracy nad robotem androidalnym jest, oczywiście poza jego użytecznością, łatwość obsługi maszyny, która wyglądem i budową przypomina górną kończynę człowieka. Sposób kontroli, czy to automatyczny (komputerowy), czy też manualny, powinien być intuicyjny i przyjazny.

2. MODUŁ RĘCZNEGO STEROWANIA

Na potrzeby prezentowanego projektu Androarm002 został skonstruowany panel kontrolny (rys.1), który steruje pracą ramienia (rys.2). Panel sterujący manipulatorem został zaprojektowany z myślą o zasadach ergonomii. Nauka obsługi panelu kontrolnego nie powinna zająć wiele czasu, gdyż rozmieszczenie przełączników wynika z budowy ramienia. Budowa ma ułatwić korzystanie z manipulatora. Obsługujący ramię, intuicyjnie wyobraża sobie, jak wygląda ruch danego segmentu „oczyma robota”.



Rysunek.1 Panel kontrolny manipulatora.

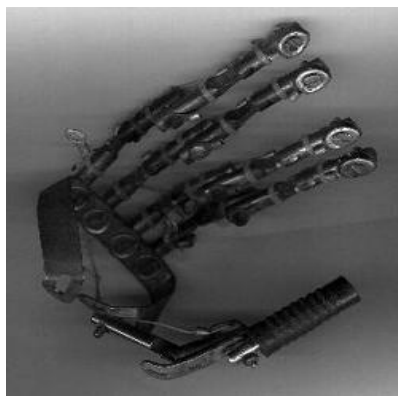


Rysunek.2 Widok z boku głównych elementów nośnych ramienia.



Rysunek.3 Mocowanie chwytaka.

Na rys.3 i rys.4 przedstawiono widok i sposób mocowania chwytaka.



Rysunek.4 Widok chwytaka z dołu.

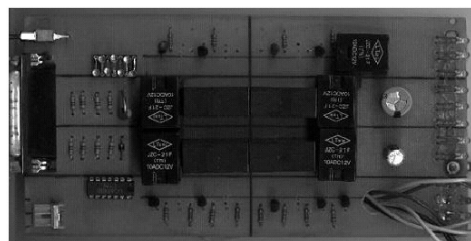
3. UKŁAD STEROWANIA RUCHEM RAMIENIA

Do budowy ramienia Adroarm002 użyte zostały głównie podzespoły samochodowe, co zapewnia stabilne i długotrwałe ich funkcjonowanie, a ponadto wypełnia oczekiwania, co do ceny. Sterowanie odbywa się poprzez odwracanie polaryzacji napięciowej za pomocą układu sterującego silnikami, które są dostatecznie wydajne, aby mogły podnieść ciężar równoznaczny kilkudziesięciu kN. Ruch pojedynczego stawu jest realizowany poprzez dwustopniową przekładnię: ślimakową i zębatą. Budowa tej przekładni jest dość specyficzna, gdyż ostatnim szczeblem przekładni jest czwarta część koła zębatego bezpośrednio przymocowanego do elementu poruszanego. Uszkodzenie przekładni jest praktycznie niemożliwe, gdyż na końcach znajdują się elastyczne ograniczniki uniemożliwiające dalszy ruch. Silniki stosowane w przekładni ramienia i przedramienia są odporne na płynące przez nie prądy, nawet, gdy opory mechaniczne spowodują zatrzymanie mechanizmu.

4. ELEKTRONICZNY MODUŁ STERUJĄCY

Moduł stanowi interfejs pomiędzy komputerem klasy PC, a urządzeniami robotycznymi różnych kształtów i rozmiarów. Układ ten został zaprojektowany jako element badawczo – rozwojowy. W celu ułatwienia oprogramowania moduł został wyposażony w elektroniczny system kontrolny zbudowany z elementów logicznych (rys.6). Część elektroniczna została zaprojektowana tak, aby w momencie popełnienia błędu przez programistę nie nastąpiło całkowite niszczące zwarcie układu powodujące przeciążenie i uszkodzenie modułu. Układ ten został zaprojektowany z

myślą o sterowaniu wybranymi segmentami ramienia androidalnego, jednocześnie może on stanowić moduł kontrolny innego układu robotycznego, np. małego mobilnego robota. Jego zadaniem jest sterowanie za pośrednictwem komputera czterema silnikami, które pracują w sposób niezależny od siebie i są w pełni programowalne.



Rysunek.5 Widok ogólny modułu sterującego.

Zakłada się, że w przyszłości podobne układy mogłyby swobodnie sterować bardziej zaawansowanymi technicznie manipulatorami. W powiązaniu z „widzeniem” maszynowym i interpretowaniem przez maszynę prostych kształtów (poprzez kamerę zintegrowaną z systemem) układy tego typu mogłyby wykonywać wiele czynności niebezpiecznych, bądź szkodliwych dla organizmu człowieka. Maszyny tego typu byłyby wielozadaniowe i stanowiłyby alternatywę dla zastosowań kończyn człowieka.

5. ZASADU I SPOSÓB PROGRAMOWANIA MODUŁU STERUJĄCEGO

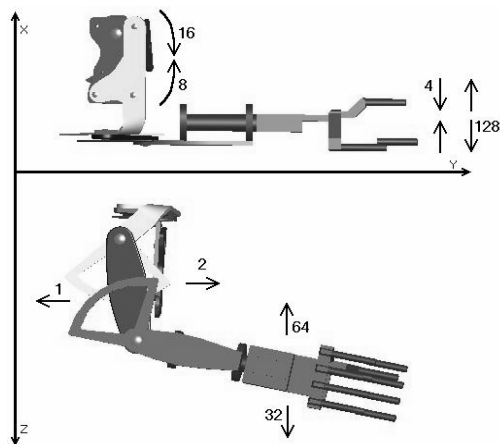
Opracowano w ramach pracy schemat sterowania ramieniem za pośrednictwem portów komputera. Na rys. 6 przedstawiono ideowy opis rozmieszczenia programowalnych wartości pin w porcie LPT1 odpowiedzialnych za działanie wybranego segmentu. Podczas oprogramowywania modułu przestrzegano kryteriów:

- pin, do którego się odwołujemy
- pin ma na celu uruchomić silnik, który zadziała w odpowiednim kierunku symulując pracę poszczególnego mięśnia;
- na jaki czas ma zadziałać nasz symulowany mięsień.

Przykładowo, aby podnieść przedramię, a ruch był wykonywany w czasie jednej sekundy, musimy odwołać się do pinu 64 portu LPT1 i zadeklarować opóźnienie („delay”) równe 1000 (czas podawany w milisekundach). Obracając ramię w prawo ruchem trwającym dwie sekundy, odwołujemy się do pinu 8 portu LPT1 i deklarujemy

opóźnienie równe 2000. Możliwa jest współzależność dwóch i więcej działających napędów jednocześnie. Maksymalnie możemy zadziałać czterema pinami z portu LPT1 jednocześnie. Więc, aby wysunąć ramię do przodu odwołujemy się do pinów 2 i 32 portu LPT1 na czas opóźnienia dostosowany do realizowanej czynności.

Programy realizowane przez system stanowiący komputer, moduł kontrolujący, oraz ramię androidalne, mogą mieć charakter: a - sztywny; b - dynamiczny; c - adaptacyjny.



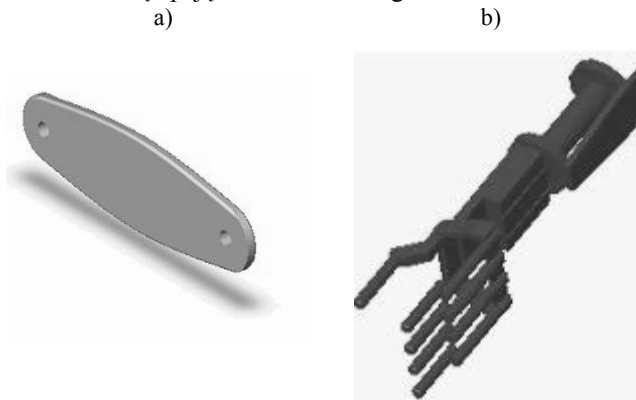
Rysunek.6 Opis portów LPT1 odpowiedzialnych za poszczególne ruchy ramienia.

6. WYKORZYSTANIE ŚRODOWISKA SOLIDWORKS, NA POTRZEBY PROJEKTOWANIA MANIPULATORA

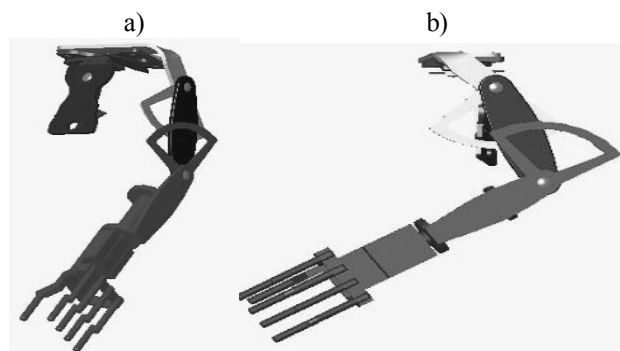
Program SolidWorks jest komputerowym narzędziem wspomagającym projektowanie elementów trójwymiarowych. Bazuje na grafice wektorowej, więc jest bardzo dokładny w odwzorowywaniu szczegółów konstrukcyjnych, (podobnie jak inne programy: auto CAD i 3D Studio Max), co czyni go bardzo przydatnym narzędziem. Złożenia elementów dokonane za pomocą programu SolidWorks zasadniczo można podzielić na dwie grupy: połączenie ruchome, połączenie sztywne.

Aby uzyskać zdolność ruchu i móc lepiej przewidzieć działanie manipulatora, niezbędnym okazało się użycie złożenia za pomocą programu SolidWorks z aktywną opcją połączeń ruchomych (rys.7, rys.8). Efekt ruchomego złożenia był możliwy poprzez nałożenie na siebie dwóch elementów posiadających otwory o tej samej średnicy. Do połączenia tych elementów został użyty nit, który umieszczono w otworach i spełnia tam jednocześnie rolę ogranicznika ruchu. Element ten pozawala na ruch w

dwuwymiarowej osi współrzędnych pomiędzy dwiema częściami manipulatora. W rzeczywistości przypomina to osadzenie czopa w panewce łożyska ślizgowego z zablokowaną opcją ruchu wzdłużnego.



Rysunek.7 Widok elementu ramienia – a) i chwytaka – b) wykonanych w środowisku SolidWorks.



Rysunek.8 Widok ramienia z przodu – a), z boku – b).

7. PODSUMOWANIE I WNIOSKI

Efektem prowadzonych prac jest wykonanie projektu modułu komputerowego i manipulatora Androarm002. Podstawowe cele projektu zostały zrealizowane. Analiza badań prowadzonych pod kątem oceny mechanizmów androidalnych oraz komputerowych systemów kontroli stanowi niezwykle interesujące pole doświadczeń. Zastosowanie tych technologii może się okazać w przyszłości wiodącą gałęzią prac badawczych i konstruktorskich. Pomijając wykorzystanie techniczne można tę technologię z powodzeniem stosować w protetyce lub w budowie mechanizmów posiadających sztuczną

inteligencję. Tego typu „roboty” można byłoby wykorzystać podczas np. urzeczywistniania misji kosmicznych, w których owe urządzenie spełniałoby te same funkcje, co człowiek. Z pewnością następne urządzenia tego typu będą bardziej dopracowane technologicznie. Praca nad mechanizmami androidalnymi jest dowodem na to, że dysponując ograniczonymi środkami, oraz niewielkim zapleczem badawczym, można podjąć próbę „naśladowania” najlepszego konstruktora, którym okazała się ewolucja. W niedalekiej przyszłości tego typu instalacje będą wyposażone w liczne czujniki oraz będą potrafiły postrzegać i interpretować otoczenie, za pomocą wbudowanej sztucznej inteligencji. Zakłada się, że roboty bardziej zaawansowane będą potrafiły poprzez własną sieć neuronową rozpoznawać kończyny należące do siebie samych i odróżniać je od tła. Przydatne stanie się widzenie stereoskopowe (dwie kamery przekazujące obraz pod innym kątem). Takie rozwiązanie umożliwi maszynie interpretowanie przestrzenne i ocenę odległości. Realizowane funkcje widzenia będą przypominały widzenie z perspektywy widzenia człowieka. Zbudowany system komputerowego sterowania wraz z manipulatorem podczas testów okazał się wystarczający do realizowania prostych czynności, które w innym przypadku musiałyby wykonywać ludzka kończyna.

Literatura

1. Szczepański Z.: Problematyka mechanizmów i układów androidalnych rozpatrywana na podstawie sterowanego komputerowo MANIPULATORA ANDROARM002
- praca magisterska pod kierunkiem dr. hab. M. Nieraszkowskiego, dr. inż. M. Macko. Bydgoszcz 2005.
2. Chlebus E.: Techniki komputerowe CAX w inżynierii produkcji, WNT, W-wa 2000.

ZASTOSOWANIE SZTUCZNYCH SIECI NEURONOWYCH DO ROZPOZNAWANIA OBRAZÓW

Wojciech Dobrosielski

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

Abstract: 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

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

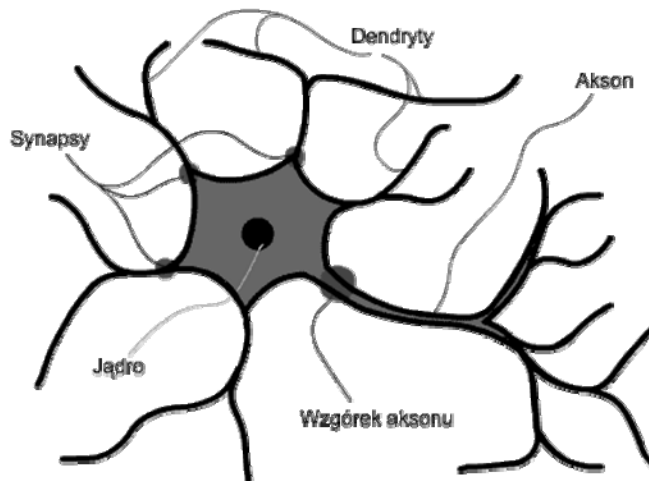
Percepcja człowieka umożliwia mu odbieranie i analizowanie skomplikowanych obrazów z otaczającego go świata. Obraz daje człowiekowi najwięcej informacji o otoczeniu. Z kolei mózg błyskawicznie przetwarza informacje przedstawione w postaci obrazu. Przetwarzanie informacji dzieje się tak szybko i jest 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 uogólniać 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 blado 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.



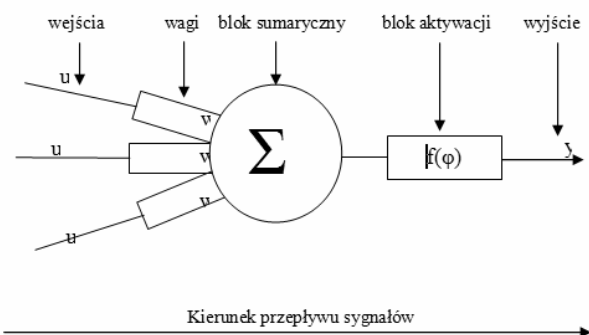
Rysunek. 2.1 Schemat neuronu biologicznego [1]

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 niesłyszanie 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:

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

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 oznaczmy jako ϕ , odpowiedź, czyli wyjście neuronu jako y , a funkcję aktywacji jako f , to operację wykonywaną w bloku aktywacji można zapisać tak:

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

„Zatem wartość owej funkcji, czyli odpowiedź neuronu ($f(\phi)$), 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 nauczania sieci są podawane w postaci par

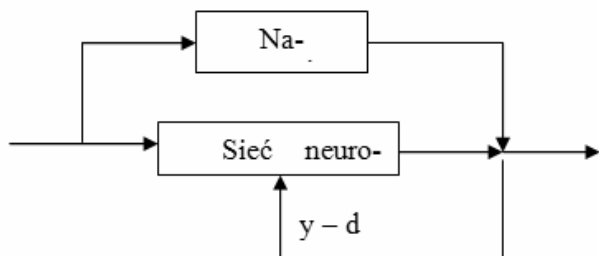
$$(x(k), d(k)) \text{ dla } k = 1, 2, \dots, p \quad (2.11)$$

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. Programista może jedynie obserwować wynik poczyniń sieci i odpowiednio korygować wartość wag, tak, aby uzyskać ostateczny cel. „Jeśli działania podjęte przez układ uczący dają wyniki pozytywne, 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 jedyną różnicą 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]

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

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 pikselem, 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 niezależny od światła i pigmentów a składa się z L – Luminacji (jasności pikseli), parametrów a – zakresie kolorów od zielonego do czerwonego i b – zakresie kolorów od niebieskiego do żółtego.
- 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 – przekształcenie klasy niższej w wyższą (segmentacja, binaryzacja, konturyzacja, szkielestowanie), przekształcenia klasy wyższej w niższą (interpolacja, 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 tłumia 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.

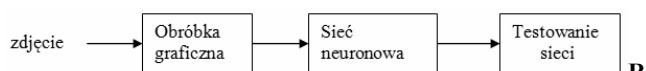
Chcąc zastosować sztuczną sieć należy określić, czego chcemy ją nauczyć. W pracy tej sieć będzie nauczona rozpoznawania liter i liczb podanych w postaci obrazu. Problem rozpoznawania liter i liczb w formie obrazu jest

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:

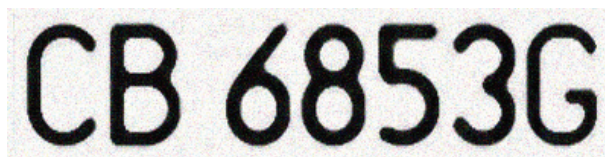
CB6853G.bmp (4.1)

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.

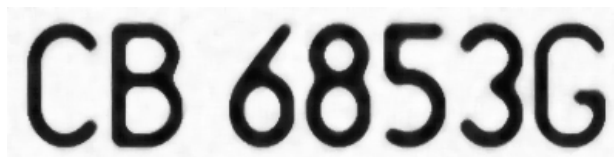
2. W nowym dokumencie ustawia się dokładne wymiary rysunku 400 x 100 pikseli. Dokładny wymiar stanowi tu istotną funkcję, ponieważ parametry wysokości i szerokości obrazka będą potrzebne później do projektowania sieci neuronowej.



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 filtra 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 wyostrenie zarysu danych.



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

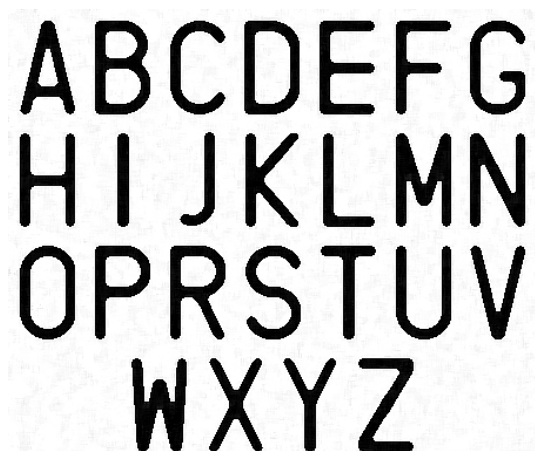
6. Tak przygotowany rysunek zostaje poddany jeszcze jednej transformacji, którą jest binaryzacja. Polega ona na zamianie odcieni szarości na dwa kolory, czarny i biały. W Photoshopie proces ten wykonuje się przez funkcję Indeksowania. Binarizacja spowoduje, że sieć będzie w jeszcze szybszy sposób przetwarzać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



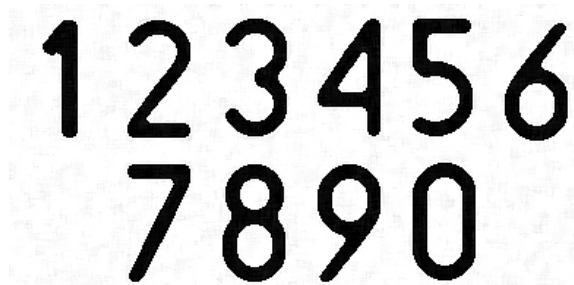
Rysunek. 5.6 Efekt końcowy po obróbce obrazka. [źródło własne]

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]

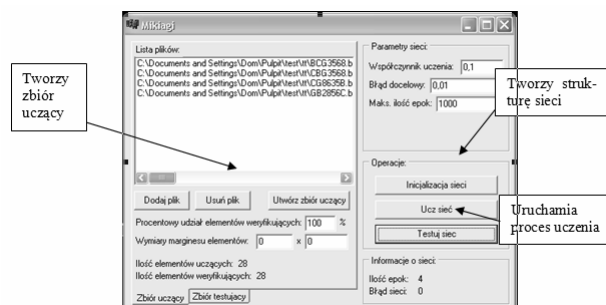


Rysunek. 5.8 Cyfry wykorzystywane w tablicach rejestracyjnych [źródło własne]

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.



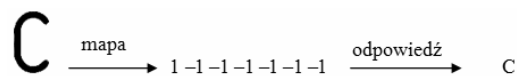
Rysunek. 6.1 Panel sterowania sieci do tworzenia zbioru uczącego [źródło własne]

Pobierane są pliki do programu, które zawierające 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 głębie 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 korzystać 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. Two-rzenie 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:

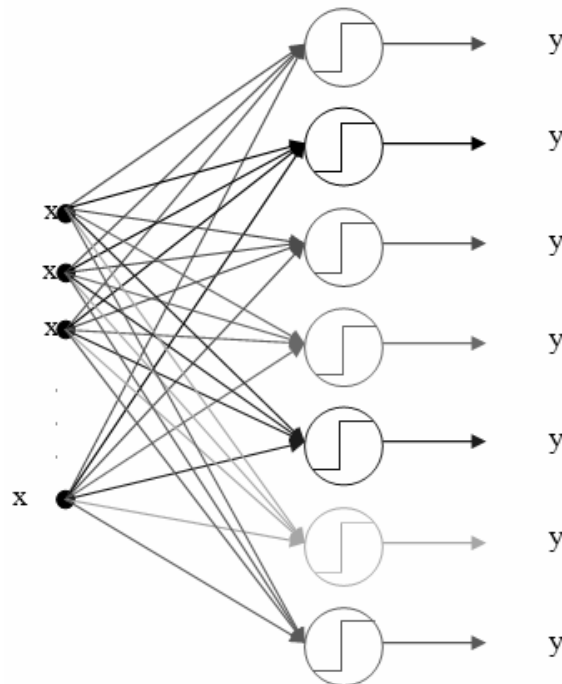
- ilość wejść sieci, która jest równa długości elementów, wektorów wejściowych w zbiorze uczącym.

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 $\phi(k)$ ze wzoru

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

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 $\delta = d(k) - \phi(k)$

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 rejestracyjnej 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”.



UNIERSYTET KAZIMIERZA WIELKIEGO

Wydział Matematyki, Fizyki i Techniki
ul. Chodkiewicza 30

fax.: (+4852) 34-01-978
tel.: (+4852) 34-19-264



www.simis.ukw.eu.pl

Studia i Materiały Informatyki Stosowanej

Czasopismo młodych pracowników naukowych, doktorantów i studentów

Wprowadzenie

W dzisiejszej technice i naukach ścisłych trudno obejść się bez wspomagania procesów badawczych i technologicznych metodami i narzędziami informatycznymi. Nowe czasopismo *Studia i Materiały Informatyki Stosowanej* skierowane jest do naukowców wykorzystujących szeroko rozumiane metody informatyczne w swoich pracach badawczych na polu różnych dyscyplin technicznych. Pozwoli ona skonsolidować różne w swoich obszarach badawczych dyscypliny techniczne z punktu widzenia podobnych metod komputerowych wspomagających te dziedziny. Do aktywnego udziału zaproszeni są wszyscy pracownicy, doktoranci i studenci, szczególnie ci reprezentujący szeroko rozumianą technikę, którym bliska naukowo jest dziedzina Informatyki Stosowanej. Szczególne zaproszenie kierujemy do młodych pracowników naukowych, do osób prowadzących koła naukowe. Naszym pragnieniem jest aby SiMIS stało się forum gdzie doktoranci, czy studenci-dyplomanci opublikują swoje pierwsze publikacje, często z promotorem, i wkroczą na drogę działalności naukowej.

Zakres tematyczny

- Różne dziedziny Informatyki (*Sztuczna Inteligencja, Kryptologia, Inżynieria Oprogramowania, Algorytmika, Programowanie Równoległe, Sieci komputerowe, itp.*),
- Komputerowe wspomaganie projektowania CAX,
- Symulowanie komputerowe procesów w technice,
- Symulacje i modelowanie w naukach ścisłych,
- Przetwarzanie danych pomiarowych,
- Mechatronika,
- Wszelkie zastosowania technik komputerowych w nauce i technice.

Ważne daty

30.06.2011 Zamknięcie numeru 4/2011

20.12.2011 Zamknięcie numeru 5/2011



UNIERSYTET KAZIMIERZA WIELKIEGO

Wydział Matematyki, Fizyki i Techniki
ul. Chodkiewicza 30

fax.: (+4852) 34-01-978
tel.: (+4852) 34-19-264



www.simis.ukw.eu.pl

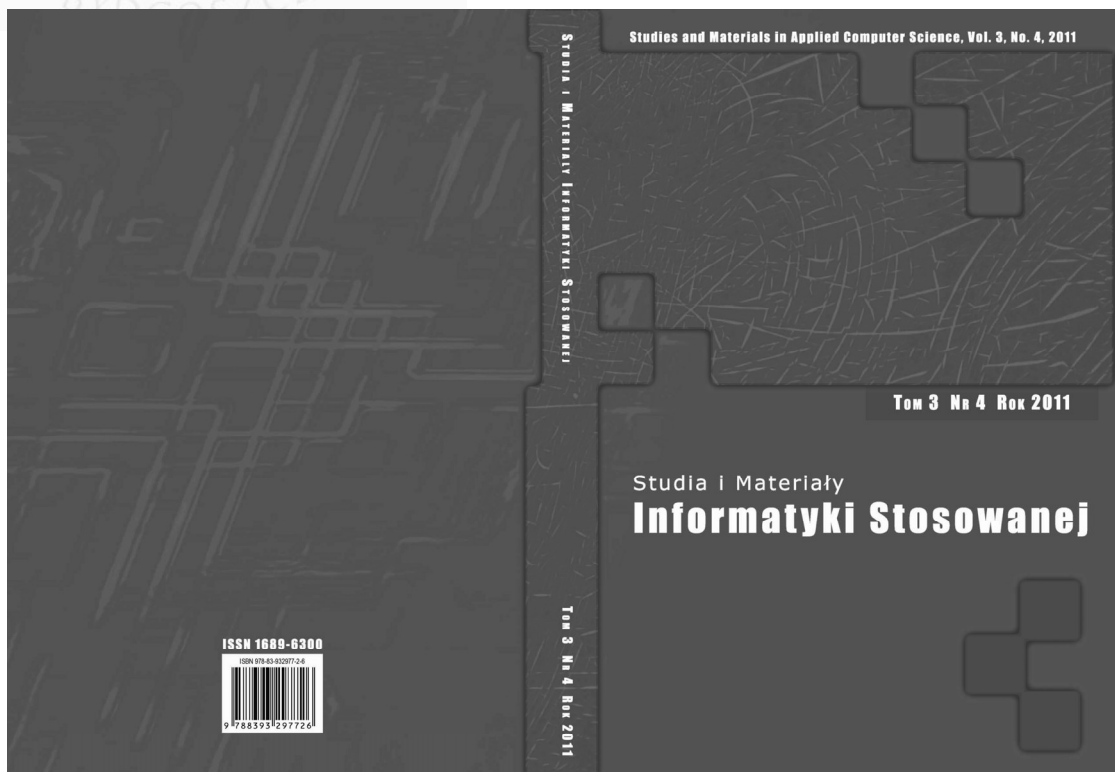
Dla autorów

Studia i Materiały Informatyki Stosowanej są czasopismem ukazującym się co pół roku. Możliwe jest wyłącznie nadsyłanie artykułów, których wcześniej nie publikowano, wszystkie artykuły są recenzowane.

- Studia i Materiały Informatyki Stosowanej są wydawnictwem recenzowanym
- Autorem może być nauczyciel akademicki, doktorant lub student oraz każda inna osoba zainteresowana tematyką.
- Format artykułu w załączniku, grafika będzie konwertowana do skali szarości
- Język publikacji: polski z obowiązkowym streszczeniem w j.angielskim, lub całość artykułu w j.angielskim.
- Zgłoszenie artykułu następuje przez wysłanie maila z wypełnionym plikiem „SiMIS2010 zgłoszenie” na adres: jczerniak@ukw.edu.pl
- Artykuły napisane z użyciem szablonu „SiMIS2010 szablon” prosimy przesłać na adres: jczerniak@ukw.edu.pl

Kontakt

- dr inż. Jacek Czerniak, kontakt: tel. (052) 341 92 64, p.216A, jczerniak@ukw.edu.pl
- dr inż. Marek Macko, kontakt: tel. (052) 341 93 30, p.04, mackomar@ukw.edu.pl





Faculty of Mathematics, Physics
and Technical Science
ul. Chodkiewicza 30

KAZIMIERZ WIELKI UNIVERSITY IN BYDGOSZCZ

fax.: (+4852) 34-01-978
tel.: (+4852) 34-19-264



www.smacs.ukw.edu.pl

Studies and Materials in Applied Computer Science

Journal of young researchers, PhD students and students

Introduction

Modern engineering and science cannot do without computer-aided research and technologic processes. The new journal, Studies and Materials in Applied Computer Science, is aimed at scientists who, in their research work on diverse engineering branches, use computer science methods in their broad sense. With that journal, it will be possible to consolidate different engineering branches that are aided by common computer-related methods. We would like all employees, PhD students, students and specially those who represent engineering in its broad sense and whose research works are closely related to Applied Computer Science to feel invited to actively cooperate with the journal. We address that invitation especially to young researches and to leaders of scientific circles. We would like SMACS to become a forum where PhD students or MSc. candidates publish their first papers, often together with their thesis supervisor, and thus they enter the world of scientific activity.

The editorial staff is going to publish original papers of experts of world renown as well as young scientists both in Polish and in English language.

An important part of our publishing programme will be special editions prepared by world-renown authors.

We also plan to publish review papers presenting the state of knowledge in a given branch of science as well as special papers written by most outstanding Polish and foreign scientist, who present original, sometimes controversial attitude towards challenges that are put out to computer science in its broad sense by the modern world. Those specially ordered papers will be published in a series called "Meetings with Masters". We particularly want young computer science specialists who enter the world of science or begin their professional career to learn opinion of well-known experts of their branch. Hence the "master-student" relationship that has been present in science and in culture since the dawn of time will be continued by our journal.

Range of topics

- Different Computer Science branches (*Artificial intelligence, Cryptology, Software Engineering, Theoretical Computer Science, Concurrent programming, Computer networks, etc.*),
- Computer Aided Design CAx,
- Computer Aided technological process simulation,
- Simulation and modelling in science,



- Measurement data processing,
- Mechatronics,
- Any other computer technology applications in science and engineering.

Important deadlines

- Submissions to No. 4/2011: **June 30th**
- Submissions to No. 5/2011: **December 20th**

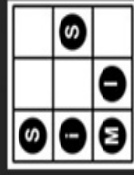
To authors

Studies and Materials in Computer Science is a journal that comes out twice a year. Papers sent to us by their authors cannot be published before. All the articles will be subject to review. By now, research studies were published under the name Applied Computer Science Symposium, a compact publication in monograph form and it included papers presented during meetings and symposiums.

- Studies and Materials in Applied Computer Science is a publication subject to review
- The author of a paper can be an employee, a PhD student, a student or any other person interested in this subject.
- See the attachment with the format of the paper. Be aware that graphics will be converted into greyscale.
- Language of a publication: Polish plus mandatory summary in English or an entire paper in English.
- Papers should be submitted by sending an e-mail including completed file „SMACS2009_request” to the address: jczerniak@ukw.edu.pl
- Papers written using the „SMACS2009_template” template should be sent to the address: jczerniak@ukw.edu.pl

Contact:

- Jacek Czerniak, PhD. contact: Phone: (+4852)341-92-64, room 216A, jczerniak@ukw.edu.pl
- Marek Macko, PhD contact: Phone: (+4852)341-93-30, room 04, mackomar@ukw.edu.pl



Studia i Materiały Informatyki Stosowanej



Ministerstwo Nauki
i Szkolnictwa Wyższego

W ZWIĄZKU Z OTRZYMANYM GRANTEM MNISW NA
UMIĘDZYNARODOWIENIE CZASOPISMA „STUDIA I MATERIAŁY
INFORMATYKI STOSOWANEJ”, AKTUALNE NUMERY (WYDANE W ROKU
2011) BĘDĄ TŁUMACZONE NA JĘZYK ANGIELSKI

ZAPEWNIAMY:

- ☒ BEZPŁATNE, PROFESJONALNE TŁUMACZENIE NADEŚŁANYCH
W JĘZYKU POLSKIM TEKSTÓW NA JĘZYK ANGIELSKI
- ☒ STYPENDIA WYDAWNICZE W WYSOKOŚCI PEŁNEJ OPŁATY
WYDAWNICZEJ DLA

KAŻDEGO ARTYKUŁU, KTÓRY SPEŁNIA WYMOGI RECENZENTÓW

W 2011 ZA **DARMO** WYDRUK I **DARMOWE TŁUMACZENIE** NA

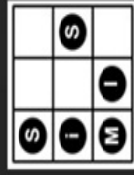
Bliższe informacje na stronie:

www.simis.ukw.edu.pl



* W MOMENCIE PRZYGOTOWANIA ANKIETY EWALUACYJNEJ CZASOPISMO
WYPEŁNIA KRYTERIA MNISW, PRZYZNAJĄCE 2 PUNKTY ZA PUBLIKACJĘ NAUKOWĄ





Studia i Materiały Informatyki Stosowanej



Ministerstwo Nauki
i Szkolnictwa Wyższego

W ZWIĄZKU Z OTRZYMANYM GRANTEM MNISW NA
UMIĘDZYNARODOWIENIE CZASOPISMA „STUDIA I MATERIAŁY
INFORMATYKI STOSOWANEJ”, AKTUALNE NUMERY (WYDANE W ROKU
2011) BĘDĄ TŁUMACZONE NA JĘZYK ANGIELSKI

ZAPEWNIAMY:

- ☒ BEZPŁATNE, PROFESJONALNE TŁUMACZENIE NADESŁANYCH
W JĘZYKU POLSKIM TEKSTÓW NA JĘZYK ANGIELSKI
- ☒ STYPENDIA WYDAWNICZE W WYSOKOŚCI PEŁNEJ OPŁATY
WYDAWNICZEJ DLA

KAŻDEGO ARTYKUŁU, KTÓRY SPEŁNIA WYMOGI RECENZENTÓW

W 2011 ZA **DARMO** WYDRUK I **DARMOWE TŁUMACZENIE** NA

Bliższe informacje na stronie:

www.simis.ukw.edu.pl



* W MOMENCIE PRZYGOTOWANIA ANKIETY EWALUACYJNEJ CZASOPISMO
WYPEŁNIA KRYTERIA MNISW, PRZYZNAJĄCE 2 PUNKTY ZA PUBLIKACJĘ NAUKOWĄ

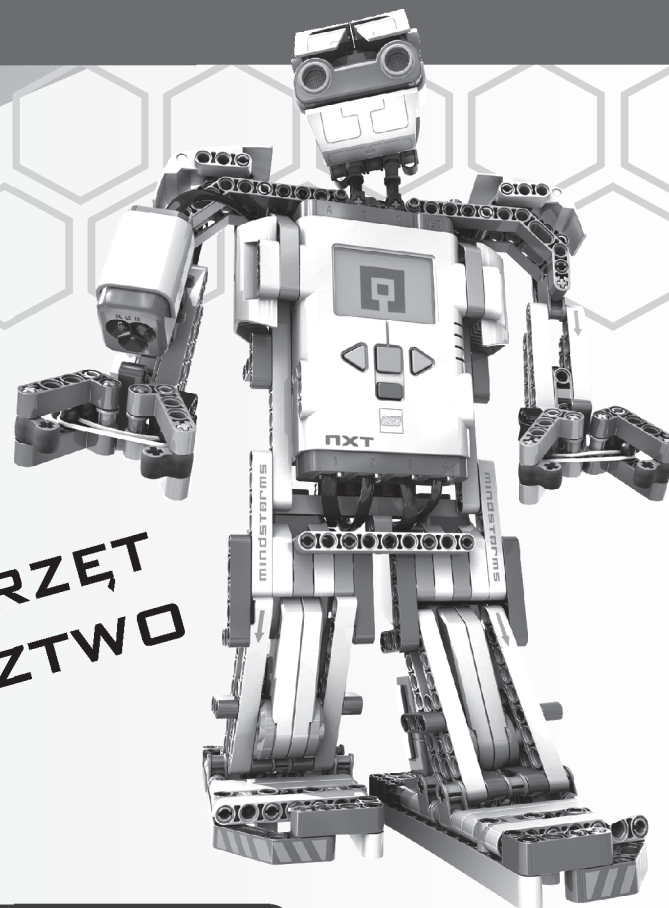




LIDER WŚRÓD DOSTAWCÓW
SPRZĘTU MECHATRONICZNEGO



- ZAAWANSOWANY SPRZĘT
- FACHOWE DORADZTWO

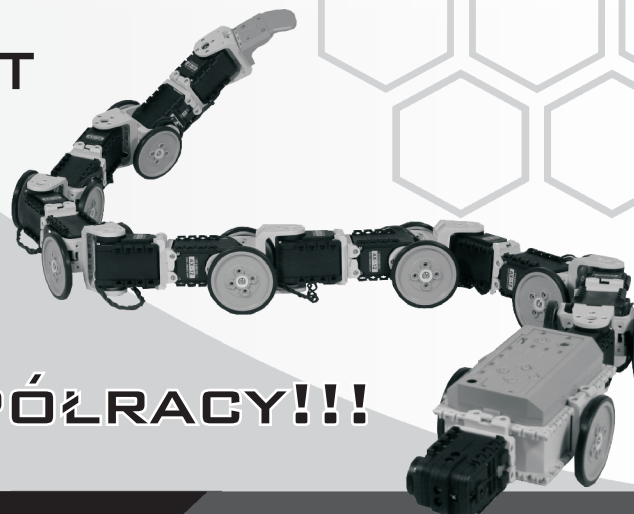


**WYPOSAŻYMY KAŻDE
LABORATORIUM MECHATRONIKI!**

SZCZEGÓLNIIE POLECAMY ZESTAWY EDUKACYJNE:

- BIOLOID
- LEGO MINDSTORMS NXT
- QUANSER
- LYNXMOTION
- FISHERTECHNIK

ZAPRASZAMY DO WSPÓŁRACY!!!



RoboNET Sp. z o.o.
ul. Trzy lipy 3 | 80-172 Gdańsk
tel./fax (58) 739 61 61 | biuro@roboshop.pl
www.roboshop.pl

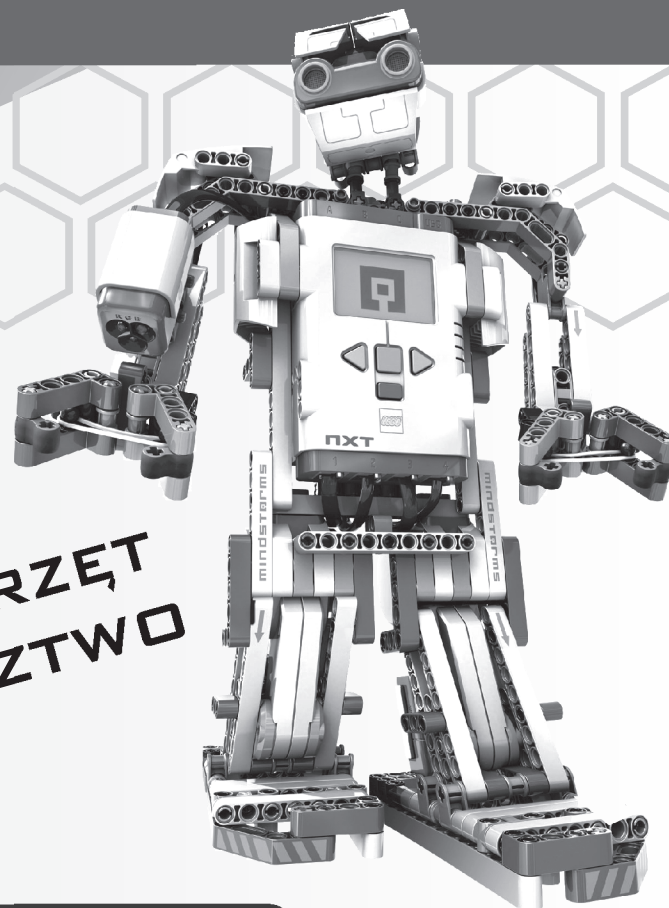




LIDER WŚRÓD DOSTAWCÓW
SPRZĘTU MECHATRONICZNEGO



- ZAAWANSOWANY SPRZĘT
- FACHOWE DORADZTWO

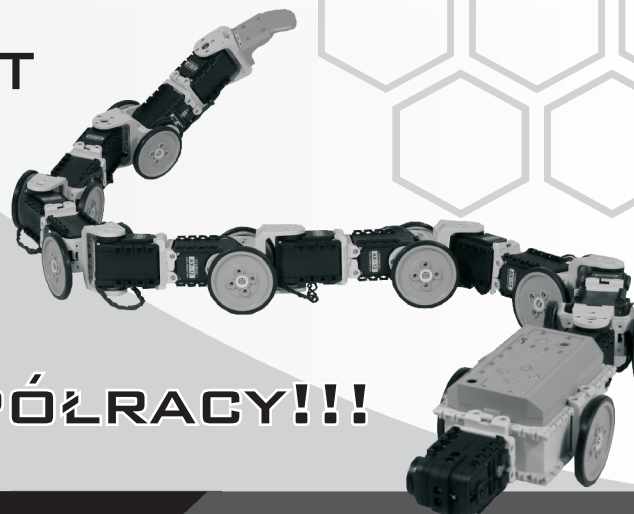


**WYPOSAŻYMY KAŻDE
LABORATORIUM MECHATRONIKI!**

SZCZEGÓLNIIE POLECAMY ZESTAWY EDUKACYJNE:

- BIOLOID
- LEGO MINDSTORMS NXT
- QUANSER
- LYNXMOTION
- FISHERTECHNIK

ZAPRASZAMY DO WSPÓŁRACY!!!



RoboNET Sp. z o.o.
ul. Trzy lipy 3 | 80-172 Gdańsk
tel./fax (58) 739 61 61 | biuro@roboshop.pl
www.roboshop.pl

