

POLITECHNIKA ŚLĄSKA
WYDZIAŁ INŻYNIERII BIOMEDYCZNEJ



PROJEKT
BIOCYBERNETYKA

Rozpoznawanie zwierząt przy pomocy sieci neuronowej
uczonej algorytmem wstecznej propagacji błędu przyspieszoną
metodą momentum.

Autorzy:

Agnieszka DOMAŃSKA

Katarzyna RADECKA

Nadzorca:

Mgr inż. Beata CHOROBA

Gliwice, 14 stycznia 2018

Spis treści

<i>1. Sprawozdanie</i>	3
1.1 Cel pracy	3
1.2 Wprowadzenie	3
1.3 Specyfikacja programu	5
1.3.1 Dane aplikacji	5
1.3.2 Proces uczenia	7
1.4 Wyniki	8
1.5 Podsumowanie	9

1. Sprawozdanie

1.1 Cel pracy

Głównym celem projektu było utworzenie aplikacji w środowisku *MatLab*, która przy użyciu sztucznej sieci neuronowej pozwoliłaby na rozpoznanie gromady zwierzęcia na podstawie zestawu 16 cech danych wejściowych. Sztuczną sieć neuronową uczono algorytmem wstecznej propagacji błędu przyspieszoną metodą momentum.

1.2 Wprowadzenie

Sztuczne sieci neuronowe należą do szybko rozwijającej się dziedziny inżynierii biomedycznej. Zostały zaprojektowane tak, aby sposobem działania i budową naśladowały to co natura i ewolucja stworzyły i rozwijały przez miliony lat. Opierają się na uproszczonym, a jednocześnie matematycznie i algorytmicznie zaawansowanym modelu rzeczywistego, biologicznego systemu nerwowego. Analogicznie do układu nerwowego, sztuczna sieć neuronowa zbudowana jest z kilkuset, a nawet kilku tysięcy sztucznych neuronów, których budowa i sposób działania przypominają rzeczywistą komórkę nerwową.

Określenie *uczyć* w kontekście sieci neuronowych rozumie się jako sposób dostosowywania odpowiedzi programu na określone dane wejściowe do założonych wymagań. W dziedzinie uczenia sztucznych sieci neuronowych wyróżnia się dwa podstawowe mechanizmy uczenia:

- z nauczycielem,
- bez nauczyciela.

Uczenie sieci neuronowej metodą wstecznej propagacji błędu jest jedną z metod uczenia z nadzorem. Podczas tego procesu dokonuje się prezentacji pewnej liczby zestawów uczących (tzn. wektorów wejściowych oraz odpowiadających im wektorów sygnałów wzorcowych wyjściowych). Należy tak dobrać wagi neuronów aby w efekcie końcowym błąd popełniany przez sieć był mniejszy od zadanego. Uczenie w przypadku zastosowania tego algorytmu jest procesem trójetapowym w trakcie którego wykonywane są kolejno kroki:

1. Podanie na wejścia sieci wektora sygnałów \mathbf{X} oraz uzyskanie sygnałów wynikowych na wyjściach sieci dla wszystkich neuronów – w szczególności ostatniej warstwy;
2. Obliczenie błędów uzyskanych na wyjściach neuronów ostatniej warstwy względem wektora oczekiwanych wartości \mathbf{Z} ;
3. Wykonanie propagacji błędów, która odbywa się warstwa po warstwie.

Szczególnym przypadkiem uczenia metodą wstecznej propagacji błędu jest zastosowanie zjawiska bezwładności procesu uczenia – *Momentum*. Metoda ta polega na dodaniu składnika bezwładności do poprawki każdej wagi.

1.3 Specyfikacja programu

1.3.1 Dane aplikacji

Dane dotyczące osobników rozpoznawanych przez program zostały zaimportowane ze strony <http://archive.ics.uci.edu/ml/datasets/zoo>. Na podstawie danych cech utworzono wektory wejściowe określające zwierzęta. W każdym wektorze znajdowała się informacja na temat:

- gatunku zwierzęcia,
- posiadania włosów,
- posiadania opierzenia,
- składania jaj,
- produkcji mleka,
- lotności,
- życia w środowisku wodnym,
- drapieżnictwa,
- posiadania uzębienia,
- posiadania kręgosłupa,
- oddychania,
- jadowitości,
- posiadania łusek,
- ilości posiadanych nóg,

- posiadania ogona,
- udomowienia,
- wielkości zbliżonej do wielkości kota,
- przynależności do gromady.

Wszystkie informacje poza nazwą gatunku, ilości posiadanych nóg (typ numeryczny) oraz nazwą gromady były typem danych bool.

girl,1,0,0,1,0,0,1,1,1,1,0,0,2,0,1,1,1

Rys. 1.1: Przykładowy wektor danych wejściowych – w tym wypadku dziewczynka.

Na podstawie powyższych cech dokonano klasyfikacji zwierząt do jednej z siedmiu następujących grup:

1. Ssaki,
2. Ptaki,
3. Gady,
4. Ryby,
5. Płazy,
6. Owady,
7. Mięczaki.

Posiadając zestaw 101 zwierząt oraz ich cech przygotowano odpowiednie wektory przystosowane do sprawnego działania aplikacji. W tym celu z wektorów pobranych ze wspomnianej wcześniej strony usunięto wartości określające nazwę gatunku oraz numer gromady. Jednocześnie numery określające typ gromady pełniły funkcję nauczyciela sieci jako osobno utworzony w tym celu wektor.

1.3.2 Proces uczenia

Początkowo sieć neuronowa była uczona przy pomocy wszystkich wektorów zwierząt („na pamięć”), aby dostosować parametry uczenia takie jak:

- wartość składnika bezwładności (*momentum*),
- cel wydajności,
- liczba epok uczenia,
- liczba warstw sieci,
- liczba neuronów w warstwie.

Następnie na podstawie optymalnego rozwiązania uczenia „na pamięć” ograniczono ciąg uczący do 68 osobników, a wektor rozpoznawanych zwierząt do 38. W ciągu uczącym zastosowano *oversampling* gadów, które w trakcie uczenia na pamięć okazały się być dla sieci najbardziej problematyczne do rozpoznania. Dokonano weryfikacji dostosowanych parametrów oraz ich optymalizacji dla ograniczonego zbioru.

1.4 Wyniki

Celem dokonania wyboru optymalnych parametrów dokonano kilkunastu prób, które przedstawiono w poniższych tabelach.

Numer próby	Liczba warstw oraz neuronów w warstwie	Liczba epok	Cel wydajności	Wartość składnika bezwładności	Liczba błędnie rozpoznanych zwierząt
1	[1 1 1]	1000	10^{-10}	0.9	78
2	[3 3 3]	1000	10^{-10}	0.9	40
3	[5 5 5]	1000	10^{-10}	0.8	46
4	[5 5 5]	2000	10^{-10}	0.7	41
5	[3 3 3]	3000	10^{-10}	0.9	69
6	[15 10 5]	3000	10^{-10}	0.7	31
7	[25 20 15]	3000	10^{-10}	0.5	11
8	[25 25 25]	3000	10^{-5}	0.3	2

Rys. 1.2: Tabela wyników testowania parametrów dla grupy 101 zwierząt.

Numer próby	Liczba warstw oraz neuronów w warstwie	Liczba epok	Cel wydajności	Wartość składnika bezwładności	Liczba błędnie rozpoznanych zwierząt
1	[25 25 25]	3000	10^{-5}	0.3	8
2	[25 25 25]	3000	10^{-5}	0.3	5
3	[25 25 25]	3000	10^{-5}	0.3	6
4	[25 25 25]	3000	10^{-5}	0.3	4
5	[25 25 25]	3000	10^{-5}	0.3	4
6	[25 25 25]	3000	10^{-5}	0.3	6
7	[25 25 25]	3000	10^{-10}	0.3	3
8	[25 25 25]	3000	10^{-10}	0.2	5
9	[25 25 25]	3000	10^{-15}	0.2	2

Rys. 1.3: Tabela wyników testowania parametrów dla grupy 38 rozpoznawanych zwierząt.

Numer próby	Liczba warstw oraz neuronów w warstwie	Liczba epok	Cel wydajności	Wartość składnika bezwładności	Liczba błędnie rozpoznanych zwierząt
1	[25 25 25]	3000	10^{-15}	0.2	7
2	[25 25 25]	3000	10^{-15}	0.2	4
3	[25 25 25]	3000	10^{-15}	0.2	5
4	[25 25 25]	3000	10^{-15}	0.2	4
5	[25 25 25]	3000	10^{-15}	0.2	3
6	[25 25 25]	3000	10^{-15}	0.1	3
7	[25 25 25]	3000	10^{-15}	0.1	2

Rys. 1.4: Tabela wyników testowania parametrów dla grupy 38 rozpoznawanych zwierząt, z zastosowanym oversampling’iem gadów w ciągu uczącym.

1.5 Podsumowanie

Właściwy dobór współczynnika uczenia oraz współczynnika momentum ma duży wpływ na tempo oraz zbieżność procesu uczenia. Optymalne wartości obu tych współczynników mogą być różne, nie tylko dla poszczególnych iteracji ale także dla każdej z wag. Wartości przyjęte na początku procesu uczenia sieci mogą okazać się niewłaściwe, co znacznie zmniejsza efektywność uczenia, a w skrajnym przypadku może prowadzić do jego rozbieżności. Stosowanie współczynnika momentu pozwala jednak wyraźnie przyspieszyć proces uczenia, bez zaburzenia stabilności algorytmu – jednak metoda ta nie jest doskonała. Zauważono, iż przy małych ilościach poszczególnego typu danych (jakimi są np. płazy czy gady) sztuczna sieć neuronowa nie jest w stanie ich rozpoznać i przydziela przedstawicieli tych grup do najliczniej występujących typów zwierząt.

Bibliografia

- [1] <http://archive.ics.uci.edu/ml/datasets/zoo> data dostępu: 14.01.2018
- [2] Dr Zdzisław Stęgowski *Sztuczne Sieci Neuronowe* Kernel. –2004
- [3] Ryszard Tadeusiewicz *Sieci Neuronowe* Akademicka Oficyna Wydawnicza 1993
s.122-124, s.45
- [4] Ryszard Tadeusiewicz *Problemy Biocybernetyki* Wydawnictwo Naukowe PWN, Wydanie II, 1994 s.139