

POLITECHNIKA ŚLĄSKA W GLIWICACH
WYDZIAŁ INŻYNIERII BIOMEDYCZNEJ

Projekt

Biocybernetyka
Uczenie sztucznych sieci neuronowych - przegląd
metod

Daniel Imiołek

Zabrze, 20 stycznia 2018

Spis treści

1. Wprowadzenie	2
1.1 Sztuczna Sieć neuronowa - co to właściwie jest?	2
1.2 Historia	2
1.3 Typy Sieci Neuronowych	3
2. Sztuczne Sieci Neuronowe	4
2.1 Metoda reguły delta	4
2.2 Metoda wstecznej propagacji błędów	5
2.3 Sieć Hopfielda	6
2.3.1 Maszyna Boltzmanna	7
2.4 Sieci Kohonena	7
3. Podsumowanie	11
4. Bibliografia	12

1. Wprowadzenie

1.1 Sztuczna Sieć neuronowa - co to właściwie jest?

Definicją sztucznej sieci neuronowej jest zbiór prostych jednostek obliczeniowych przetwarzających dane, komunikujących się ze sobą i pracujących równolegle. Pierwowzorem wszystkich sieci neuronowych jest ludzki mózg, dlatego można powiedzieć, że sztuczna sieć neuronowa jest bardzo uproszczonym modelem mózgu.

1.2 Historia

Początki sieci neuronowych datuje się na rok 1943 wraz z wydaniem historycznej pracy McCulloch'a i Pitts'a, w której po raz pierwszy przedstawiono matematyczny opis komórki nerwowej oraz powiązanie go z problemem przetwarzania danych. W 1949 roku Donald Hebb, odkrył, że informacja może być przechowywana w strukturze połączeń pomiędzy neuronami i jako pierwszy zaproponował metodę uczenia sieci polegającą na zmianach wag połączeń między neuronami. W latach 50-tych budowano pierwsze sieci neuronowe. Pierwszym szeroko znanym przykładem zbudowanej i ciekawej działającej sieci neuronopodobnej jest perceptron wymyślony przez Rosenblatta. Sieć ta była przedstawiona jako układ częściowo elektromechaniczny, częściowo elektroniczny. Zbudowana została ona w Cornell Aeronautical Laboratory. Zadaniem tej sieci było rozpoznawanie znaków. Po wielu próbach okazało się, że sieć nie potrafiła rozpoznać złożonych znaków i była wrażliwa na zmianę skali obiektów i ich położenie. Zaletą była zdolność do zachowania poprawnego działania nawet po uszkodzeniu pewnej części elementów. Po ogłoszeniu wyników przez twórców nastąpił gwałtowny rozwój tego typu sieci neuronowych na całym świecie. Kolejnym przełomem było zbudowanie sieci elektrochemicznej uczącej się z elementów Adaline w 1960r. przez Bernarda Widrowa. Sieć ta składała się z elementów Adaline, które powielone oraz połączone dały układ Madaline (Many Adaline). Sieć Madaline wykorzystywana jest w radarach, sonarach, modemach i liniach telefonicznych. Rozwój sieci neuronowych został gwałtownie zahamowany na początku lat 70-tych, gdy Minsky i Papert dowiedli, że sieci jednowarstwowe mają bardzo ograniczony zakres zastosowań. Taki stan utrzymywał się blisko 15 lat, aż do ukazania publikacji, że sieci nieliniowe są wolne od ograniczeń, które wykazali Minsky i Papert w swoich pracach. W 1982r. Kohonen opracował sieci do wydobywania cech, uczące się bez nauczyciela. W latach 80-tych pojawiły się pierwsze sieci ze sprzężeniem zwrotnym, w których rozwiązywanie zadań polegało na poszukiwaniu

przez sieć stanu równowagi w długim iteracyjnym procesie dynamicznym. Przykładem jest opracowana w 1982 roku przez Johna Hopfielda teoria komiwojażera. Badania sieci neuronowych i ich rozwój jest mocno rozwijany do dziś.

1.3 Typy Sieci Neuronowych

Wyróżniamy 3 typy sieci neuronowych

- Sieci Jednokierunkowe
- Sieci Rekurencyjne
- Samoorganizujące się mapy

Sieci możemy też podzielić ze względu na metody uczenia sieci neuronowych

- Sieci z nauczycielem (Uczenie nadzorowane)
- Sieci bez nauczyciela (Uczenie nienadzorowane)

2. Sztuczne Sieci Neuronowe

2.1 Metoda reguły delta

Reguła Delta została opracowana przez Widrowa i Hoffa, znalazła ona zastosowanie do uczenia elementów liniowych i nieliniowych. Reguła delta jest regułą uczenia z nauczycielem. Polega ona na tym, że każdy neuron po otrzymaniu na swoich wejściach określone sygnały (z wejść sieci albo od innych neuronów, stanowiących wcześniejsze piętra przetwarzania informacji) wyznacza swój sygnał wyjściowy wykorzystując posiadaną wiedzę w postaci wcześniej ustalonych wartości współczynników wzmocnienia wag wszystkich wejść oraz ewentualnie progu. Wartość sygnału wyjściowego, wyznaczonego przez neuron na danym kroku procesu uczenia porównywana jest z odpowiedzią wzorcową podaną przez nauczyciela w ciągu uczącym. Jeśli występuje rozbieżność - neuron wyznacza różnicę pomiędzy swoim sygnałem wyjściowym a tą wartością sygnału, która była by - według nauczyciela prawidłowa. Ta różnica oznaczana jest zwykle symbolem greckiej litery delta i stąd nazwa opisywanej metody.

Sygnał błędu wykorzystywany jest przez neuron do korygowania swoich współczynników wagowych:

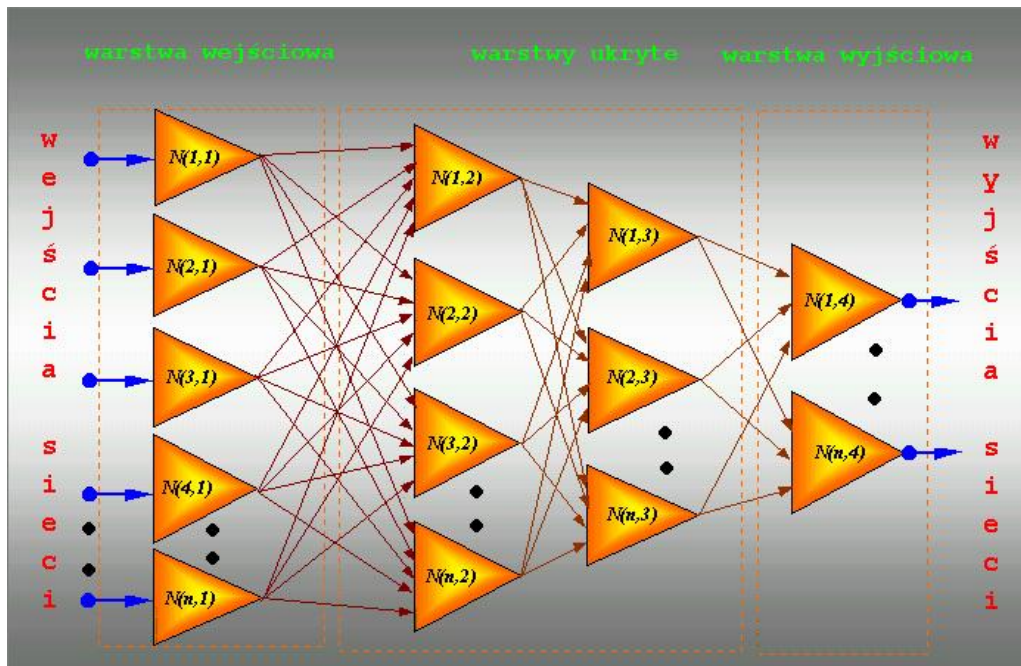
- wagi zmieniane są tym silniej, im większy jest błąd
- wagi związane z tymi wejściami, na których występowały duże wartości sygnałów wejściowych zmieniane są bardziej niż wagi wejść, na których sygnał wejściowy był niewielki

Znając błąd popełniony przez neuron oraz jego wagi wejściowe możemy łatwo przewidzieć jak będą się zmieniać jego wagi.

$$y = \sum_{i=1}^n w_i x_i = \mathbf{W}^T \mathbf{X},$$

gdzie:

$$\mathbf{X} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix}, \quad \mathbf{W} = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_n \end{bmatrix}$$



Rys. 2.1: Schemat sieci

2.2 Metoda wstecznej propagacji błędów

Przez wiele lat nie znaleziono skutecznej metody uczenia sieci wielowarstwowych, dopiero w latach 80-tych zaproponowany został algorytm wstecznej propagacji błędów polegający na tym, że mając wyznaczony błąd $\delta_m^{(j)}$ występujący podczas realizacji j-tego kroku procesu uczenia w neuronie o numerze m można podawać ten błąd wstecz do wszystkich tych neuronów, których sygnały stanowiły wejścia dla m-tego neuronu.

Uczenie odbywa się przez minimalizację odpowiednio zdefiniowanej funkcji celu $Q(W)$, przy czym wektor W reprezentuje wagi sieci poddawane optymalizacji. Najprostsza funkcja celu ma postać błędu średniokwadratowego. Zastosowanie różniczkowalnej funkcji aktywacji umożliwia minimalizację funkcji celu metodami gradientowymi.

Wzór na wyznaczenie błędu sieci

$$Error_i = Output_i(1 - Output_i)(Actual_i - Output_i)$$

Gdzie:

$Error_i$ jest błędem osiągniętym w węźle i-tym

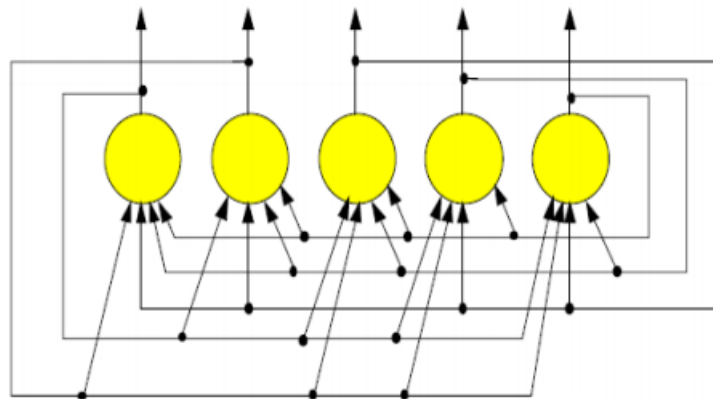
$Output_i$ jest wartością przewidzianą przez sieć

$Actual_i$ jest wartością rzeczywistą, której sieć powinna się nauczyć

Najprościej mówiąc celem tej metody jest zoptymalizowanie wag, aby sieć neuronowa mogła się nauczyć poprawnie mapować dowolne wejścia na wyjścia.

Schemat działania metody wstecznej propagacji błędów.

- 1 Wyznaczenie odpowiedzi neuronów warstwy wyjściowej oraz warstw ukrytych na zadany sygnał wejściowy.



Rys. 2.2: Schemat sieci Hopfielda

- 2 Wyznaczenie błędu popełnianego przez neurony znajdujące się w warstwie wyjściowej i przesłanie go w kierunku warstwy wejściowej.
- 3 Adaptacja wag.

2.3 Sieć Hopfielda

<http://th-www.if.uj.edu.pl/~erichter/dydaktyka/Dydaktyka2012/SieciNN-2012/NN-wyklad>

<https://www.mimuw.edu.pl/~rlatkows/publications/latkowski1999sieci.pdf>

Pseudoinwersja z instrukcji

Sieć Hopfielda jest najbardziej znaną siecią, w której kierunek przepływu sygnałów jest odwrócony, posiada sprzężenia zwrotne typu każdy z każdym, jest prostym przykładem sieci rekurencyjnej i często jest nazywana autoasocjatorem, a w ramach tego sprzężenia każdy neuron jest połączony z jednym z wejść oraz z własnym wyjściem.

W sieciach hopfielda wykorzystujemy uczenie oparte na pseudoinwersji macierzy. Tak dobieramy wagi, aby uzyskać na wyjściu takie same wzorce jakie podajemy na wejściu.

$$WX = X \quad (2.1)$$

gdzie \mathbf{W} to macierz wag o wymiarze $n \times n$, a \mathbf{X} to macierz wzorców o wymiarze $n \times p$ złożoną z p wektorów uczących.

Jednym z ważniejszych osiągnięć pracy Hopfielda jest pojęcie funkcji energetycznej w sieciach neuronowych. Najważniejszą własnością funkcji energetycznej jest to, że zawsze maleje lub pozostaje stała, gdy układ ewoluuje zgodnie z regułą

$$Z_i^n = \text{sign}\left(Z_i^n + \frac{1}{N} \sum_{j=1}^N \sum_{m \neq n}^p Z_i^m Z_j^m Z_i^n\right)$$

W ten sposób wzorce leżą w minimach lokalnych powierzchni funkcji energetycznej. Dla sieci neuronowych funkcja energetyczna istnieje, gdy wagi połączeń są symetryczne.

2.3.1 Maszyna Boltzmanna

Z siecią Hopfielda kojarzona jest zwykle Maszyna Boltzmanna. Koncepcja tej maszyny oparta jest na założeniu, że stan (sygnał wyjściowy) każdego neuronu może się zmieniać w sposób losowy z określonym prawdopodobieństwem. Prawdopodobieństwo to zależy od "energii" i "temperatury" sieci podobnie jak w systemach termodynamicznych, w których gęstość prawdopodobieństwa energii systemu związana jest z temperaturą. Przekładając te informacje bardziej w informatyczny sposób, możemy na każdym kroku j związać z neuronem o numerze m energię $E_m^{(j)}$ wyrażającą nadwyżkę jego łącznego pobudzenia $e_m^{(j)}$ ponad progiem pobudzenia $w_0^{(m)}$.

$$E_m^{(j)} = e_m^{(j)} - w_0^{(m)} \quad (2.2)$$

Następnie w oparciu o tą energię wyznaczane jest prawdopodobieństwo zgodnie z regułą będącą uogólnieniem prawa Boltzmanna.

$$p_m^{(j)} = 1 / [1 + \exp(-\delta E_m^{(j)} / T^{(j)})]$$

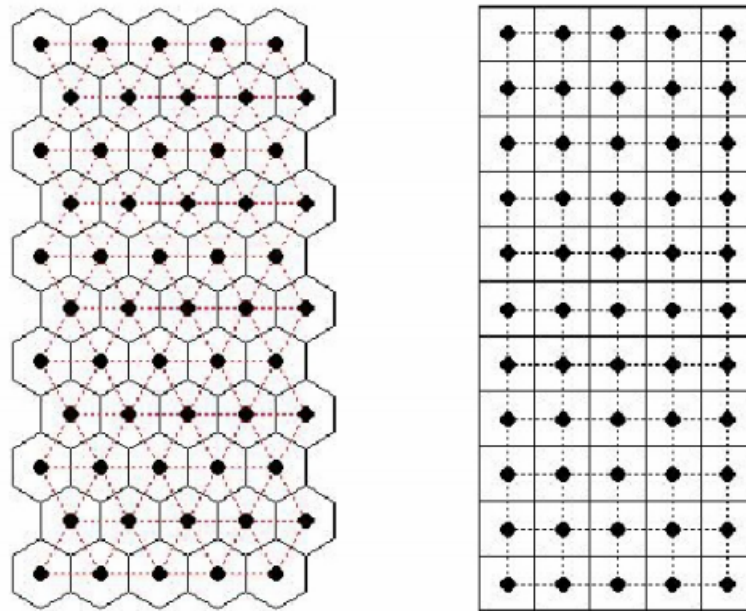
gdzie δ jest pewną arbitralnie dobieraną stałą, a $T^{(j)}$ reprezentuje symulowaną w j -tym kroku "temperaturę" sieci.

2.4 Sieci Kohonena

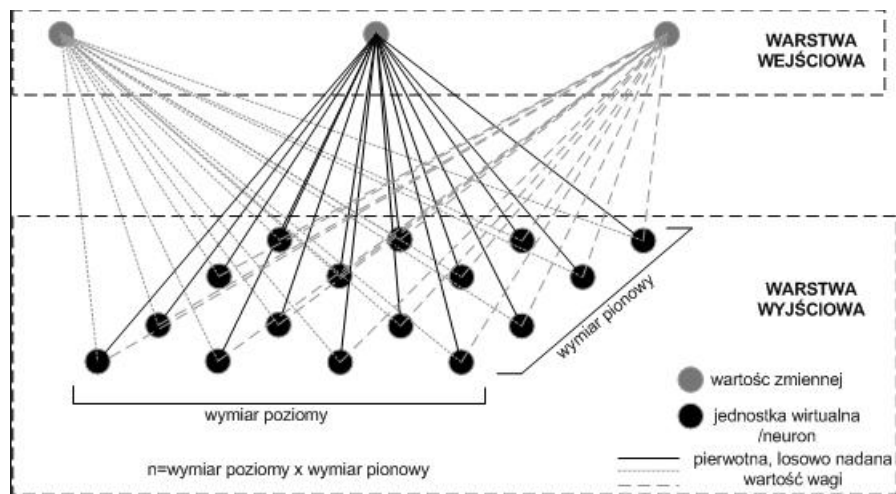
Sieć Kohonena jest przykładem sieci samoorganizującej się i nienadzorowanej - bez nauczyciela. Składa się z 2 warstw. Warstwa wejściowej - wektorów wejściowych i warstwa wyjściowej - mapy topologicznej. Sieć Kohonena jest siecią jednokierunkową. Sieć ta wyróżnia się tym od innych sieci, że zachowuje odwzorowanie sąsiedztwa przestrzeni wejściowej. Wynikiem działania sieci jest klasyfikacja przestrzeni w sposób grupujący zarówno przypadki ze zbioru uczącego, jak i wszystkie inne wprowadzenia po procesie uczenia. Najczęściej używaną mapą wynikową jest mapa dwuwymiarowa, analizowane dane mają być na niej odwzorowane.

Zasady działania sieci Kohonena:

- Wejścia (tyle, iloma parametrami opisano obiekty) połączone są ze wszystkimi węzłami sieci
- Każdy węzeł przechowuje wektor wag o wymiarze identycznym z wektorami wejściowymi
- Każdy węzeł oblicza swój poziom aktywacji jako iloczyn skalarny wektora wag i wektora wejściowego (podobnie jak w zwykłym neuronie)

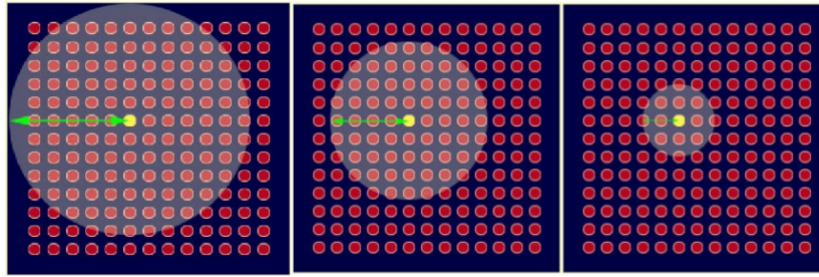


Rys. 2.3: Przykładowe mapy dwuwymiarowe - siatka heksagonalna i prostokątna



Rys. 2.4: Struktura sieci Kohonena

- Ten węzeł, który dla danego wektora wejściowego ma najwyższy poziom aktywacji, zostaje zwycięzcą i jest uaktywniony
- Wzmacniamy podobieństwo węzła-zwycięzcy do aktualnych danych wejściowych poprzez dodanie do wektora wag wektora wejściowego (z pewnym współczynnikiem uczenia)
- Każdy węzeł może być stowarzyszony z pewnymi innymi, sąsiednimi węzłami - wówczas te węzły również zostają zmodyfikowane, jednak w mniejszym stopniu.



Rys. 2.5: Zmniejszanie się zasięgu sąsiedztwa, wraz z kolejnymi iteracjami

$$w_c(k+1) = w_c(k) + \eta(k) h_{cm}(k) [x(k) - w_c(k)], \quad m = 1, \dots, M$$

We wzorze powyższym:

- $\eta(k)$ oznacza współczynnik uczenia,
- c oznacza numer wektora-zwycięzcy, tj. numer wektora w znajdującemu się najbliżej prezentowanego w k -tym kroku wektora $x(k)$,
- wartość funkcji $h_c(k)$ określa, w jakim stopniu należy uwzględnić przynależność neuronu i do sąsiedztwa zwycięskiego neuronu o numerze $c = c(k)$.

Rys. 2.6: Pierwszy wzór i opis uczenia się sieci kohonena

- Neurony w poszczególnych warstwach nie mogą się komunikować między sobą.

Tak jak wcześniej wspomniano, przy uczeniu się mapy w sieci Kohonena ważną rolę odgrywa sąsiedztwo neuronów. Wyznacza się je według położenia wektorów referencyjnych na mapie. Sąsiedztwo neuronu oznaczonego numerem c Będziemy oznaczać N_c .

Ważnym pojęciem jest również zasięg sąsiedztwa, który zazwyczaj zmniejsza się podczas uczenia.

Inicjalizacja wag sieci Kohonena jest losowa. Wektory wejściowe stanowią próbę uczącą, podobnie jak w przypadku zwykłych sieci rozpatrywaną w pętli podczas budowy mapy. Wykorzystanie utworzonej w ten sposób mapy polega na tym, że zbiór obiektów umieszczamy na wejściu sieci i obserwujemy, które węzły sieci się uaktywniają. Obiekty podobne powinny trafiać w podobne miejsca mapy. Na ogół uczenie przebiega w dwóch fazach. Najpierw przyjmuje się duży promień sąsiedztwa i duży współczynnik uczenia. W drugiej fazie obydwie te wielkości ulegają zmniejszeniu, w szczególności promień sąsiedztwa spada do zera. Pierwsza faza przebiega według zasady Winner Takes Most, gdzie zwycięski neuron zostaje najbardziej zmodyfikowany, ale udział w modyfikacji biorą także jego sąsiedzi (zależnie od odległości od zwycięzcy). Zmiana wag w 1 fazie następuje według wzoru 2.6. W drugiej fazie uczenia obowiązuje zasada Winner Takes All, gdzie zwycięzca bierze wszystko, a sąsiedzi nie są modyfikowani, ponieważ promień sąsiedztwa zmalał do zera, zmienia się tylko wektor wagowy w_c . Wzór: (2.7).

$$w_{\alpha}(k+1) = w_{\alpha}(k) + \eta(k) [x(k) - w_{\alpha}(k)],$$

Rys. 2.7: Drugi wzór uczenia się sieci kohonena

3. Podsumowanie

SOM Self Organizing Maps sieć uczy się bez nadzoru sieć składa się z dwóch warstw o wyraźnie rozdzielonych funkcjach uporządkowane neurony wyjściowe uczony głównie neuron zwycięski ważna rola sąsiedztwa w wyniku uczenia powstaje mapa topologiczna aprioryczna interpretacja wartości wyjściowych jest niemożliwa

Sieci jednokierunkowe Zbudowane z 1 lub kilku warstw Przepływ sygnału w tego typu sieciach przebiega zawsze w ściśle określonym kierunku: od warstwy wejściowej do warstwy wyjściowej. Każda dodatkowa warstwa pomiędzy warstwami wejściową i wyjściową nazywana jest warstwą ukrytą z uwagi na to, że jej działalność nie może być obserwowana bezpośrednio ani na wejściu sieci ani na jej wyjściu Zwykle wszystkie neurony warstwy poprzedniej połączone są ze wszystkimi neuronami warstwy następnej. Do uczenia perceptronów wielowarstwowych stosuje się algorytmy spadku gradientowego, między innymi algorytm propagacji wstecznej.

Sieci rekurencyjne

Połączenia między neuronami stanowią graf z cyklami (obieg zamknięty) tzn. sygnały z warstwy wyjściowej sieci podawane są z powrotem do warstwy wejściowej.

4. Bibliografia

R. Tadeusiewicz, Sieci neuronowe. Akademicka Oficyna Wydawnicza RM, Warszawa, 1993,

http://www.dbc.wroc.pl/Content/1908/Rusiecki_Algorytmy_PhD.pdf

https://platforma.polsl.pl/rib/pluginfile.php/2498/mod_resource/content/2/Laborator

http://www.neurosoft.edu.pl/media/pdf/tkwater/sztuczna_inteligencja/2_alg_ucz_ssn.p

<http://zsi.tech.us.edu.pl/~nowak/wi/som.pdf>