Sieci neuronowe ♠

Piotr Grabowski ♣ ♣

21 stycznia 2015

http://iisi.pcz.pl/nn/podstawy.php

Spis treści

1 **Wykład 1 - wprowadzenie ♡** 2

2 **Wykład 2 - wstęp do neurofizjologii ♡ ♡** 3

3 **Wykład 3 - liniowe i nieliniowe sieci neuronowe ♡ ♡ ♡** 5

4 **Wykład 4 - Sieci Kohonena uczenie bez nauczyciela ♡ ♡ ♡ ♡** 8

5 **Wykład 5 - sieci CP i LVQ ♡ ♡ ♡ ♡ ♡** 10

6 **Algorytm wstecznej propagacji błędów ♡ ♡ ♡ ♡ ♡ ♡** 13

7 **Wykład 7 - sieci RBF (Radial Basis Functions)** 16

8 **Wykład 8 - sieci rezonansowe** 19

9 **Wykład 9 - sieci rekurencyjne** 20

1 Wykład 1 - wprowadzenie ♡

Podstawowe cechy sieci neuronowych

1. adaptacja i samoorganizacja
2. zmniejszona wrażliwość na uszkodzenia elementów
3. równoległa praca
4. wygoda programowania przez uczenie

Zastosowania

1. klasyfikacja obrazów
2. klasteryzacja / kategoryzacja
3. aproksymacja funkcji
4. predykcja / prognozowanie
5. optymalizacja
6. odtwarzanie
7. sterowanie

Przepływ sygnału w NN

1. warstwa wejściowa
2. warstwa ukryta
3. warstwa wyjściowa

Warstwa wejściowa nie może się uczyć. (jak jest warstwa wejściowa, 2 ukryte i wyjściowa to jest to sieć trójwarstwowa (wejściowa się nie liczy))

W warstwie ukrytej są neurony ukryte. Mogą być nieliniowe. Warstwa wyjściowa nie różni się niczym od ukrytej poza tym, że jest jedna i wyjścia z niej są wyjściami z sieci.

2 Wykład 2 - wstęp do neurofizjologii ♡ ♡

Mózg waga $1250 - 1400$ g, 100 mld neuronów, dzienny ubytek 10 tyś., liczba połączeń 100 tyś mld, szerokość synaps ($20 - 25$ nanometrów - szerokość $\frac{1}{600}$ włosa. Zużycie glukozy 20% większe niż inne narządy, zużycie tlenu 20%, intensywność ukrwienia 750 ml/min, energii 10 WAT.

Neuron składa się z dendrytów , jądra komórki (na niej ziarnistości - soma) i aksonu. (na koniec kolaterale końcowe) Wejścia presynaptyczne (akson) do jądra (neuronu postsynaptycznego).

Rodzaje neuronu

1. jednobiegunowy, dwubiegunowy, gwiaździsty
2. purkinjego, kłębuszkowy, golgiego

Na końcu aksonu jest synapsa (kolbka synaptyczna). Dotyka ciała komórki odbiorczej. Synapsa składa się z neurogibryli , mitochondrium, aparatu Golgiego, pęcherzyków synaptycznych i błony pre i post synaptycznej.

Typowy kierunek przemieszczania się sygnału jest od dentrytu do aksonu, ale może następować odwrotny przepływ sygnału. pojedyncze neurony mogą wysyłać sygnały nawet przy braku elektrycznej stymulacji w obrębie ciała komórki czy dendrytów. Procesy pamięciowe komórek zachodzą w aksonach, a potencjał iglicowy jest generowany dalej (bliżej końca aksonu). Aksony komunikują się ze sobą.

Schemat funkcjonowania synapsy

1. neurotransmiter
2. pęcherzyk synaptycznych
3. enzym rozkładający
4. receptor presynaptyczny
5. receptor postsynaptyczny

Neurotransmitery

1. serotonina
2. acetylocholina
3. adrenalina
4. dopamina

Polaryzacja błony komórkowej

1. potencjał spoczynkowy -70mV
2. potencjał czynnościowy -59mV (od synapsy pobudzającej)
3. hiperpolaryzacja (od synapsy hamującej) -80mV

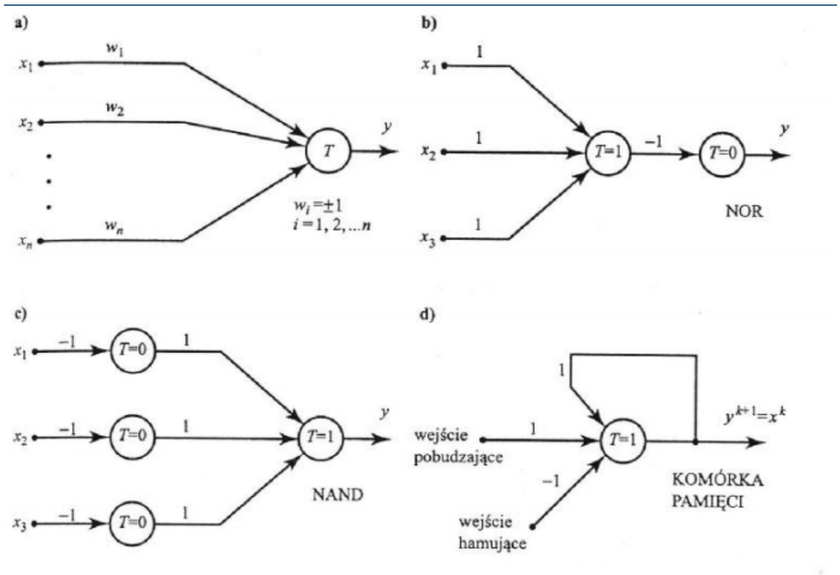
Istnieje mechanizm sygnalizacji wstecznej - endokanabinoidy działają wstecz, przemieszczają się z neuronu postsynaptycznego do presynaptycznego. Hamuje wytwarzanie się neuroprzekaźnika GABA.

Model neuronu McCullocha-Pittsa 1943

$$y^{k+1} = \begin{cases} 1, & \text{if } \sum_{i=1}^{n} w_i x_i^k \geqslant T \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

x_i^k - sygnały wejściowe, mówią, czy w chwili k pojawił się sygnał czy nie, y - wyjście neuronu. w_i może być dodatni (synapsa pobudzająca) i ujemny (hamująca), T jest wartością progową.

Wierny model pojedynczego neuronu 32000 równań różniczkowych, 19200 parametrów do oszacowania.



Rys. 2.3. Model neuronu McCullocha–Pittsa i elementarne układy logiczne: a) schemat neuronu, b) bramka NOR, c) bramka NAND, d) komórka pamięci

3 Wykład 3 - liniowe i nieliniowe sieci neuronowe ♡ ♡ ♡

1. jądro - centrum obliczeniowe neuronu, to są kluczowe procesy
2. akson - wyjście neuronu, ma tylko jedno wyjście
3. wzgórek aksonu - sumowanie przychodzących sygnałów i generowanie potencjałów czynnościowych, które wędrują przez akson
4. dendryt - wejście neuronu, może być wiele, biologiczne neurony mają ich tysiące
5. synapsa - jeśli dendryt jest wejsciem neuronu to synapsa jest furtką. ma wpływ na moc sygnału napływającego poprzez akson.

Ogólna definicja neuronu

$$y = f(\phi(x_i w_i)) \tag{2}$$

1. ϕ - Post synaptic potential (potencjał postsynaptyczny)
2. f - funkcja aktywacji
3. x_i - wejścia
4. w_i - wagi
5. y - wyjście

Definicja neuronu

$$y = f(\sum(x_i w_i)) \tag{3}$$

Zamiast PSP blok sumujący.

Neuron liniowy

$$y = \sum(x_i w_i) \tag{4}$$

Brak funkcji aktywacji.

Neuron z funkcją skokową (nieliniowy neuronik) $f(x)$ jest takie, że jeśli x jest większe od progu to robi skok (podaje 1), inaczej 0.

BIAS (przesunięcie) oraz PRÓG (threshold)

$$e = b + \sum_{i=1}^{n} x_i w_i \tag{5}$$

Można przyjąć, że $w_0 = b$ i wtedy bias jest jak każda inna waga, z tym, że sygnał x_0 jest równy 1. Wtedy:

$$e = \sum_{i=0}^{n} x_i w_i \tag{6}$$

Funkcję aktywacji można wtedy przyjąć jako:

$$f(e) = \begin{cases} 1, & \text{if } e \geqslant 0 \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

Możemy uznać, że bias jest stałym progiem wtedy:

$$e = \sum_{i=1}^{n} x_i w_i \tag{8}$$

oraz

$$f(e) = \begin{cases} 1, & \text{if } e \geqslant \theta \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

gdzie θ to próg.

Pierwszy wniosek jest taki, że $\sum x_i w_i$ jest prostą, płaszczyzną, i hiperpłaszczyzną (zależy od wymiaru).

Druga różnica jest taka, że przy założeniu, że w_2 jest różne od 0 wzory na granicę pomiędzy 0, a 1 na wyjściu są takie: (dla 2 wymiarów)

$$x_2 = -\frac{w_1}{w_2} - \frac{b}{w_2} \tag{10}$$

oraz

$$x_2 = -\frac{w_1}{w_2} + \frac{\theta}{w_2} \tag{11}$$

Jak jest próg stały to uwaga... podczas nauki się nie zmienia! A jak nie jest stały to uwaga.. się zmienia! I to czy ma być stały czy nie zależy od problemu. Bo raz zadziała, raz nie... heheszki.

Dyskryminacja liniowa - wykres przedstawiający linię oddzielającą 0 od 1 dla różnych x (np. w 2d)

Powierzchnia odpowiedzi - dodajemy 3 wykres - wartość wyjścia - dyskryminacja liniowa jest rzutem na powierzchnię odpowiedzi od góry na iksy

Neuron liniowy - screeny od Tadka Wejścia i wyjścia w neuronie mogą być znormalizowane (patrz Tadeusiewicz, ale lepiej nie) od -1 do 1.

I można zapisać wzory wektorowo:

$$y = W^T X \tag{12}$$

co jest najzwyczajniejszym na świecie mnożeniem iksy przez wartości. Jednak jeśli x_i i w_i są znormalizowane to

$$y = cos\phi \tag{13}$$

gdzie ϕ jest kątem pomiędzy wektorami W oraz X. Prosty wniosek, że sygnał będzie tym większy im bardziej kąt będzie mniejszy (czyli wektory będą skierowane bardziej w tą samą stronę).

Można też zapisać to macierzowo.

W sieci liniowej nie robi się warstw ukrytych z zasady, bo nie wzbogacą zachowania sieci. (algebra)

ADALINE - Adaptive Linear Element - Reguła Widrow-Hoffa Regułę robimy następująco:

$$W' = W + \mu\delta X \tag{14}$$

albo za pomocą wzoru

$$W' = ZX^{-1} \tag{15}$$

$mu \geqslant 0$. Jak $\delta = z - y$ jest $\geqslant 0$ to znaczy, że $z > y$ czyli odpowiedź sieci była za mała. A jak $\delta < 0$ to $z < y$ czyli odpowiedź była za duża.

Jak odpowiedź była za mała to W' się zwiększa, jak za duża to W' się zmniejsza.

Można udowodnić, że reguła ta optymalizuje wagi względem funkcji celu:

$$Q = \frac{1}{2}\sum_{i=1}^{N}\left(z^j - y^j\right)^2 \tag{16}$$

za pomocą poszukiwania minimum metodą gradientową.

MADALINE to jest Wiele ADALINE i Reguła Widrow-Hoffa również działa.

4 Wykład 4 - Sieci Kohonena uczenie bez nauczyciela ♡ ♡ ♡ ♡

Reguła Hebba

$$W' = W + nyX \tag{17}$$

Analogiczna do Widrow-Hoffa, ale zamiast błędu mamy po prostu sygnał wyjściowy.
Wady Hebba:

1. niska efektywność uczenia
2. przemnożony wpływ początkowych wart. wag
3. możliwość pomijania niektórych klas nauczonej sieci
4. powstawanie reduntantnych nadinterpretacji klas

Uczenia polega na tym, że wektory id do sygnału wejściowego uczącego. Zasada owczego pędu - idą wszystkie. Może być za duży albo za mały współczynnik uczenia. Czas nauki jest istotny. Bardzo duża liczba neuronów może nie pomagać. Sieć może zapomnieć klasy (wektory zostały *ukradzione*)

Warianty metod samouczenia Metoda przyrostowego samouczenia (differential hebbian learning)

$$w_{ki}^{j+1} = w_{ki}^{j} + \mu(x_i^j - x_i^(j-1))(y_k^j - y_k^(j-1)) \tag{18}$$

Metoda *Gwiazdy wejść* (Instar learning) - najczęściej wybierana w analizie skupień

$$w_{ki}^{j+1} = w_{ki}^{j} + \mu^j(x_i^j - w_{ki}^j) \tag{19}$$

$$\mu^j = 0.1 - \lambda \tag{20}$$

Miarą podobieństwa wektora wejściowego i wektora wag jest cosinus kąta unormowanych wektorów.
Metoda *Gwiazd wyjść* (Outstar learning)

$$w_{ki}^{j+1} = w_{ki}^{j} + \mu^j(y_k^j - w_{ki}^j) \tag{21}$$

Samouczenie sieci metodą Kohonena Po pojawieniu się sygnału wejściowego wszystkie sygnały wyjściowe są porównywane i wybierany jest **zwycięzca** o numerze k, którego sygnał y_k^j ma **największą** wartość. Wówczas zmieniamy współczynnik w następujący sposób:

$$w_{ki}^{j+1} = w_{ki}^{j} + \mu(x_i^j - w_{ki}^j) \tag{22}$$

Można zdefiniować pojęcie sąsiedztwa:

$$w_{mi}^{j+1} = w_{mi}^{j} + \mu h(m,k)(x_i^j - w_{mi}^j) \quad (23)$$

funkcja $h(m,k)$ jest malejącą funkcją odległości między neuronem m, a zwycięzcą k.
Zasady rywalizacji:

1. ZWYCIĘZCA BIERZE WSZYSTKO (WINNER TAKES ALL)
2. ZWYCIĘZCA BIERZE WIĘKSZOŚĆ (WINNER TAKES MOST)

Mapa topologiczna w korze mózgowej w obszarze czucia somatycznego oraz sterowania jest o podobnej strukturze, co sieci Kohonena. Istnieje zjawisko skręcenia się sieci Kohonena np. w postać krzyża. SOM może być wykorzystywana do przedstawiania skomplikowanych korelacji w danych statystycznych. Sieć Kohonena może służyć do rzutowania wielowymiarowego zbioru danych do przestrzeni o małej wymiarowości.

Mechanizm sumienia Istnieje możliwość wprowadzenia sumienia, wtedy niwelujemy możliwość dłuższego braku nauki jakiegoś neuronu.

Podsumowanie

1. sieć uczy się bez nauczyciela
2. ma dwie warstwy o wyraźnie rozdzielonych funkcjach (wejściowa i topologiczna)
3. uporządkowane neurony wyjściowe
4. ważną rolę odgrywa sąsiedztwo
5. w wyniku uczenia powstaje mapa topologiczna
6. apriorytyczna interpretacja wartości wyjściowych jest niemożliwa
7. po uczeniu można ustalić jakie znaczenie mają poszczególne rejony mapy topologicznej - ale tylko na podstawie analizy konkretnych danych wejściowych

Zastosowania

1. rozpoznawanie obrazów
2. klasyfikacja
3. zgłębianie danych
4. tworzenie modeli - modelu świata zewnętrznego w mózgu robota, modelu uczciwego przedsiębiorcy albo działania procesu

5 Wykład 5 - sieci CP i LVQ ♡ ♡ ♡ ♡ ♡

Uczenie z forsowaniem

$$w_{ki}^{(}j+1) = w_{ik}^{j} + \mu z_k^j x_i^j \tag{24}$$

Maksymalna liczba możliwych do zapamiętania wzorów w sieci o k neuronach

$$N_{max} = \frac{k}{2logk} \tag{25}$$

Efekty uczenia (przy założeniu, że wszystkie wektory wejściowe są ortonormalne) sieć:

1. uczy się wiernie odtwarzać wymagane reakcje na wszystkie rozważane sygnały wejściowe
2. potrafi uśredniać wejściowe sygnały i odtwarzać idealny wzorzec serii przypadkowo zniekształconych obserwacji

Architektura sieci LVQ Sieć składa się z warstwy wyjściowej, warstwy kohonena oraz warstwy wejściowej.

Sieć LVQ służy do klasyfikacji sygnałów wejściowych i jest przykładem uczenia z forsowaniem. Warstwa wyjściowa przypisuje wektory wyjściowej do jednej z kilku klas. Główną częścią sieci jest warstwa kohonena dokonująca klasyfikacji. LVQ daje **jednakową liczbę neuronów** przypisanych do danej klasyfikacji

Podklasy w danej grupie nie muszą być podobne.

Podczas uczenia obliczana jest odległość wektora wejściowego od wszystkich neuronów warstwy i wyłaniany jest najbliższy zwycięzca. Jeśli wygrywający wektor należy do klasy sygnału, który pojawił się na wejściu to jego wagi są modyfikowane tak, aby zbliżyć się do prezentowanego sygnału. Jeśli nie należy to wektor ten jest odsywany co jest określane jako *odpychanie*. Podczas procesu uczenia neuron przypisany do danej klasy wędruje do obszaru związanego z tą kategorią. W trybie testowania (klasyfikacji) obliczana jest odległość prezentowanego neuronu wejściowego do każdego neuronu i leżący najbliżej zostaje zwycięzca. Przynależność do klasy tego sygnału wskazuje ten zwycięski neuron.

$$d_i = ||w_i - x|| = \sqrt{\sum_{j=1}^{N}(w_{ij} - x_j)^2} \tag{26}$$

Modyfikacja wag zwycięskiego neuronu

$$W' = \begin{cases} w + \alpha(x - w), & \text{if neuron należy do właściwej klasy} \\ w - \gamma(x - w) & \text{otherwise} \end{cases} \tag{27}$$

Wariant LVQ #1 Istnieje możliwość wprowadzenia pojęcia *sumienia*. Jeśli neuron wygrywa zbyt często oddaje je innemu neuronowi. Realizowane jest to poprzez bias:

$$d_i' = d_i + b_i \tag{28}$$

$$d_i = ||w_i - x|| = \left(\sum_{j=1}^{N}(w_{ij} - x_j)^2\right)^2 \tag{29}$$

$$b_i = \mu d_{i_{max}}(1 - Np_i) \tag{30}$$

1. $d_{i_{max}}$ największa odległość wyestymowana wewnętrznie (?)
2. μ - stała, która zwalnia proces uczenia
3. p_i oblicza częstotliwość wygrywania, na początku $\frac{1}{N}$
4. liczba neuronów kohonena na klasę
5. ϕ stała do aktualizacji częstotliwości wygrywania

$$W' = \begin{cases} (1-\phi)p_i, & \text{if i nie jest zwycięzcą w klasie} \\ (1-\phi)p_i + \phi, & \text{otherwise} \end{cases} \tag{31}$$

Wybiera się globalnego zwycięzce obliczając odległość d_i i lokalnego zwycięzce, ale z tej klasy biorąc pod uwagę d'_i. Wagi lokalnego zwycięzcy są modyfikowane w następujący sposób:

$$W' = \begin{cases} w + \alpha(x - w), & \text{if neuron jest globalnym i lokalnym zwycięzcą} \\ w - \beta(x - w) & \text{otherwise} \end{cases} \tag{32}$$

podczas, gdy globalny zwycięzca jest odrzucany od wektora wejściowego zgodnie ze wzorem:

$$w' = w - \gamma(x - w) \tag{33}$$

jeśli globalny zwycięzca nie jest we właściwej klasie

Wariant LVQ #2 Jeśli w sieci jest neuron zwycięzca z wektorem wag w_1, który nie wskazuje na klasę sygnału wejściowego, a drugi następny w kolejności o wagach w_2 właśnie z tej klasy to w takiej wersji LVQ zwycięzca jest odpychany od sygnału wejściowego, drugi natomiast jest traktowany jak zwycięzca pod warunkiem, że odległość wektora wejściowego od obu wybranych neuronów jest podobna

$$w_1 = w_1 - \alpha(x - w) \tag{34}$$

$$w_2 = w_2 - \alpha(x - w) \tag{35}$$

Strategia uczenia sieci LVQ Podstawowy proces uczenia sieci LVQ polega na obliczeniu odległości w sensie metryki euklidesowskiej pomiędzy wektorami w_i (wektor wagi), a wektorem x (norma średniokwadratowa):

Sieci Counter Propagation zaproponowane przez Robarta Hecht-Nielsens, kompilacja sieci Kohonena i Grosberga, szybciej się uczą (w przeciwieństwie do sieci ze wsteczną propagacją). Przy pomocy CP można szybko weryfikować hipotezy robocze

$$||x|| = 1 \tag{36}$$

Normalizacja wektorów wejściowych

$$x_i' = \frac{x_i}{\sqrt{\sum_{j=1}^{n} x_j^2}} \tag{37}$$

Pierwsza warstwa realizuje algorytm Kohonena:

$$e_j = W_j^T X \tag{38}$$

$$k_j = \begin{cases} 1, & \text{if } \forall_{j \neq i} e_j > e_i \\ 0, & \text{otherwise} \end{cases} \tag{39}$$

Jak działa warstwa druga? Realizuje algorytm Outstar Grossberga.

$$Y = VK \tag{40}$$

Uczenie pierwszej warstwy W danym kroku uczenia korekcie wag podlega tylko zwycięzca

$$\Delta W = \mu_1(X - W) \tag{41}$$

Początkowo $w_{ij} = \sqrt{\frac{1}{n}}$, zamiast x podaje się na wejście x':

$$x_i^{k\prime} = \mu_2(k) x_i^k + (1 - \mu_2(k))\sqrt{\frac{1}{n}} \tag{42}$$

Uczenie drugiej warstwy Według reguły Widrow-Hoffa

$$v_{ij}^{k+1} = v_{ij}^k + \mu_3(z_i - y_i)k_j \tag{43}$$

Podczas uczenia warstwy Grossberga polega na wpisywaniu do tablicy *look up table* właściwych wartości, które mają być odpowiednią reakcją na pewną grupę sygnałów pojawiających się na wejściu sieci, a którą identyfikuje pewien neuron warstwy Kohonena.

6 Algorytm wstecznej propagacji błędów ♡ ♡ ♡ ♡ ♡ ♡

Liniowa funkcja PSP wyznacza ważoną sumę wszystkich wartości wejściowych. Ta suma następnie zostaje zmodyfikowana w taki sposób, że odejmuje się od niej wartość progową. W terminologii wektorowej można powiedzieć, że rozważana funkcja PSP jest to iloczyn skalarny wektora wag i wektora wejściowego - minus wartość progu. Neurony z liniową funkcją PSP generują liniowe funkcje dyskryminacyjne. Oznacza to, że identyczne wartości sygnału wyjściowego otrzymuje się dla sygnałów wejściowych znajdujących się po tej samej stronie hiperpłaszczyzny w przestrzeni wzorców. Położenie tej hiperpłaszczyzny w przestrzeni sygnałów wejściowych determinowane jest przez parametry neuronu (współczynniki wagowe i próg). Obserwując zachowanie neuronów z liniową funkcją PSP można stwierdzić, że próbują one rozwiązać stawiane im zadania poprzez odpowiednie manipulowanie wspomnianą hiperpłaszczyzną. Na przykład często podejmowane zadanie rozpoznawania wejściowych sygnałów neurony te usiłują zrealizować optymalizując klasyfikację wejściowych sygnałów poprzez stosowane podzielenie na części całej przestrzeni sygnałów wejściowych (na podstawie odpowiednich wzorców) za pomocą systemu przecinających się hiperpłaszczyzn.

$$y = f(\sum(x_i w_i)) \tag{44}$$

Radialna Neurony wyposażone w radialną funkcję PSP wyznaczają kwadrat odległości pomiędzy dwoma punktami w N wymiarowej przestrzeni (gdzie N jest liczbą wejść). Punkty pomiędzy którymi wyznacza się odległość reprezentują odpowiednio wektor opisujący sygnał wejściowy oraz wektor wag neuronu. Neurony posiadające radialną funkcję PSP wytwarzają identyczne wartości wyjściowe dla wszystkich sygnałów wejściowych leżących na hipersferach wyznaczonych w przestrzeni tych sygnałów wejściowych. Środki tych hipersfer ulokowane są w punktach odpowiadających wektorom wag neuronów. Wektory te pełnią rolę wzorców sygnałów, na które dana sieć powinna szczególnie reagować. Neurony radialne próbują więc zrealizować klasyfikację wejściowych sygnałów poprzez pomiar odległości reprezentowanych przez nie punktów od wyznaczonych wzorców, które przechowywane są w postaci wektorów wag neuronów. Kwadrat odległości wyznaczany przez neurony radialne mnożony jest przez wartość progową (która w neuronach radialnych pełni rolę miary wartości dopuszczalnego odchylenia); w ten sposób wyznaczana jest wartość wejściowa rozważanego neuronu. (różnica wektorowa)

$$y = f(||x - w||) \tag{45}$$

Ilorazowa Ten typ funkcji PSP został specjalnie zaprojektowany dla sieci regresyjnych i nie powinien być stosowany w innych przypadkach. W neuronach stosujących ten typ funkcji PSP oczekuje się, że waga skojarzona z jednym wejściem będzie równa $+1$, waga skojarzona z innym wejściem będzie równa -1, zaś wszystkie pozostałe wagi przyjmują wartość zero. Wartością generowaną przez tę funkcję jest wartość powstająca w ten sposób, że wartość sygnału na wejściu odpowiadającym wadze $+1$ podzielona jest przez wartość sygnału na wejściu o wadze -1.

Funkcje nieliniowe Przykłady: liniowa, threshold, sigmoid, limited linear, funkcja skokowa (binary step function) - dla x mniejszych od θ jest np. 0 a, dla większych 1 (θ - próg).

Funkcja sigmoida unipolarna

$$f(x) = \frac{1}{1 + \exp -\tau x} \tag{46}$$

Sigmoida bipolarna

$$g(x) = 2f(x) - 1 = \frac{1 - \exp -\tau x}{1 + \exp -\tau x} \tag{47}$$

Tangens hiperboliczny

$$h(x) = \frac{\exp x - \exp -x}{\exp x + \exp -x} \tag{48}$$

Funkcja Gaussa

$$f(x) = \exp -x^2 \tag{49}$$

1. liniowa
2. logistyczna $\frac{1}{1+\exp -x}$
3. wykładnicza
4. softmax $\frac{e^x}{\sum_i e^{x_i}}$
5. pierwiastek
6. sinus
7. liniowa z nasyceniem
8. progowa

Uczenie sieci nieliniowych jednowarstwowych Cel - uzyskanie jak największej zgodności pomiędzy odpowiedzią neuronu, a wymaganą wartością na wyjściu Metoda - minimalizacja funkcji kryterialnej

W efekcie otrzymujemy - regułę Delta.

$$W' = W + \mu\delta x^T \tag{50}$$

dla sieci jednowarstwowych.

Błąd na wejściu nieliniowego neuronu

$$\delta_k = (z_k - y_k)f'(e_k) \tag{51}$$

Błąd na wyjściu

$$d_{wyj} = y_k - z_k \tag{52}$$

Reguła delta dla sieci nieliniowych

$$w_{ik}^{j+1} = w_{ik}^j + \mu f'(e_k^j)(x_k^j - y_k^j)x_i^j \tag{53}$$

$$y_k^j = f(e_k^j) = f(\sum_{l=0}^{L} w_{lk}^j x_l^j \tag{54}$$

Backpropagation Algorytm uczenia sieci nieliniowych backpropagation czyli metoda wstecznej propagacji błędów polega na odtwarzaniu przypuszczalnej wartości błędów głębszych warstw sieci (do których nie ma bezpośredniego dostępu) na podstawie rzutowania wstecz błędów wykrytych na wyjściu. Rozważając pojedynczy neuron warstwy ukrytej bierze się pod uwagę błędy wszystkich tych neuronów, do których wysłał swój sygnał wyjściowy, sumuje się je uwzględniając wagi

Istnieje reguła delta dla sieci nielinowych wielowarstwowywch

Korekta Wag **sposób przyrostowy -** aktualizacja wag następuje bespośrednio po podaniu każdej pary uczącej. Funkcja błędu zmienia się w każdym kolejnym kroku. Jeżeli pary uczące podawane są w losowej kolejności to scieżka w przestrzeni wag jest stochastyczna, co pozwala lepiej wykorzystać powierzchnię błędu.

sposób grupowy – obliczany jest gradient błędu łącznego. Korekta wag następuje po podaniu całego zestawu uczącego. Ten sam efekt można uzyskać obliczając poprawki wag dla każdej pary uczącej, ale bez dokonywania jej aktualizacji. Zmiana wagi następuje po prezentacji wszytkich par uczących poprzez dodanie wszytkich poprawek.

Błąd w metodzie backpropagation Błąd średniokwadratowy z wyjścia i zadanego wyjścia oraz suma kwadratów błędów w epoce

$$\text{tss} = \sum_{k=1}^{K} \sum_{i=1}^{N} (z_i - y_i)^2 \tag{55}$$

Metoda momentum

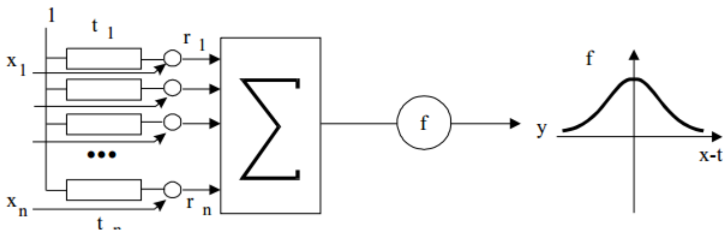
$$w_{ik}^{mj} = w_{ik}^{mj} + \mu_1 \delta_k^{mj} x_i^{mj} + \mu_2 \Delta w_{ik}^{m(j-1)} \tag{56}$$

Współczynnik μ (momentum) jest miarą bezwładności procesu uczenia, chroniąca algorytm przed niestabilnym działaniem w warunkach sielnie niemonotonicznej charakterystyki hiperpowierzchni błędu. W związku z tym wzrost wartości tego współczynnika prowadzi do wygładzania lokalnych oscylazji zmian współczynników wagowych i zwiększa prawdopodobieństwo osiągnięcia globalnego minimum funkcji błędu mimo obecności pasożytniczych atraktorów w formie drobnych, ale głębokich minimów lokalnych tej funkcji.

Zależność współczynnika uczenia od kształtu powierzchni funkcji błędu Dla płaskowyżu duża wartość powoduje większe poprawki wag, które powodują szybsze przesuwanie się w stronę minimum, zaś mała niewielkie zmiany w poszczególnych krokach i wolny przebieg procesu minimalizacji. Dla wąwozu mała wartość spowoduje, że trajektoria będzie spadać dokładnie po linii najmniejszego spadku, duża może powodować oscylacje między dwiema ścianami wąwozu.

Zależność współczynnika momentum od kształtu powierzchni funkcji błędu Dla płaskowyżu stosunkowo duża wartość nadaje procesowi minimalizację dodatkowego pędu i zwiększenie efektywnego tempa uczenia, mała zaś ma jedynie znikomy wpływ na poprawę efektywnego tempa uczenia. Dla wąwozu niewielka powoduje tłumienie oscylacji (rola filtru LP dla zmian składowych gradientu), zbyt duża powoduje odejście trajektorii zbyt daleko od linii najszybszego spadku - możliwy wzrostu funkcji błędu w kolejnych krokach.

7 Wykład 7 - sieci RBF (Radial Basis Functions)



Neuron radialny - Agregacja sygnałów wejściowych w tym typie neuronu polega na obliczaniu odległości pomiędzy obecnym wektorem wejściowym X a ustalonym podczas uczenia centroidem pewnego podzbioru T.

Również nieliniowa funkcja przejścia w tych neuronach ma odmienną formę - „dzwonu” gaussoidy - czyli jest funkcją niemonotoniczną.

Sieć typu RBF w zastosowaniu do klasyfikacji (wykrywa i sygnalizuje skupiska danych wejściowych)

Neuron radialny jest zdefiniowany przez swoje centrum i parametr - promień. Obliczana jest odległość wektora wag i wektora sygnałów. Promień jest przechowywany w neuronie jako wartość progowa.

Powtórne próbkowanie Metoda ta polega na tym, że wybrane w sposób losowy elementy ze zbioru uczącego kopiowane są do neuronów radialnych (jako występujące w tych neuronach zestawy wag). Ponieważ podlegające kopiowaniu sygnały wejściowe wybrane zostały w sposób losowy, więc żeprezentująóne (w sensie statystycznym) rozkład wszystkich danych uczących. Jednakże, jeśli liczba neuronów radialnych nie jest duża, to neurony te mogą stanowić w rzeczywistości złą reprezentację (Haykin, 1994).

k-średnich Algorytm ten (Bishop, 1995) próbuje ustalić optymalny zbiór punktów, które stanowić będą centra skupień występujących w danych uczących. Mając K neuronów radialnych musimy wytworzyć K niezależnych centrów, w taki sposób, by reprezentowały one charakterystyczne skupiska wejściowych danych. Centra te ustala się w procesie iteracyjnym, w którym powtarzane są następujące czynności

1. Każdy element zbioru uczącego przypisywany jest do tego centrum skupienia, które położone jest bliżej danego elementu niż wszystkie pozostałe centra

2. Każde centrum skupienia wyznaczane jest jako wektor średnich wartości zmiennych wyznaczony dla wszystkich punktów należących do danego skupienia

Wybór promienia (odchylenia) Definiowanie przez użytkownika. Użytkownik samodzielnie określa wielkość odchylenia. Równomierny przydział odchyleń. Odchylenie (identyczne dla wszystkich neuronów) jest określane za pomocą pewnej reguły heurystycznej, uwzględniającej liczbę centrów oraz wielkość zajmowanej przez nie przestrzeni (Haykin, 1994). Przydział metodą k-najbliższych sąsiadów. Odchylenie dla każdego neuronu jest określane indywidualnie jako średnia odległość do jego k najbliższych sąsiadów (przypadków ze zbioru danych). Stąd odchylenia są mniejsze w mocno zagęszczonym obszarze danych, co umożliwia zachowanie drobnych szczegółów i większe w obszarze, w którym dane występują rzadko (umożliwia to lepszą interpolację).

Zastosowanie RBF (zamiast MLP) spowoduje, że sieć neuronowa znajdzie aproksymację lepiej dopasowaną do lokalnych właściwości zbioru danych, ale gorzej ekstrapolującą. **Sieci RBF bywają nadmiernie wrażliwe na nawet nieliczne błędy w danych uczących**

Sieć RBF Ma ona strukturę dwuwarstwową, warstwa ukryta realizuje odwzorowanie nieliniowe realizowane przez neurony radialnej funkcji bazowej. Neuron wyjściowy jest liniowy, a jego rolą jest sumowanie wagowe sygnałów pochodzących od neuronów warstwy ukrytej.

Sieć radialna, a sieć sigmoidalna Sieci neuronowe o radialnych funkcjach bazowych znalazły zastosowanie zarówno w rozwiązywaniu problemów klasyfikacyjnych, zadaniach aproksymacji funkcji wielu zmiennych, jak i zagadnieniach predykcji tych obszarach zastosowań gdzie funkcje sigmoidalne mają ugruntowaną pozycję. W stosunku do sieci wielowarstwowych o sigmoidalnych funkcjach aktywacji wyróżniają się pewnymi właściwościami szczególnymi, umożliwiającymi lepsze odwzorowanie cech charakterystycznych modelowanego procesu.

funkcje radialne Sieci typu radialnego stanowią naturalne uzupełnienie sieci sigmoidalnych. Neuron sigmoidalny reprezentował w przestrzeni wielowymiarowej hiperplaszczyznę separującą tą przestrzeń na dwie kategorie (klasy), w których był spełniony odpowiedni warunek, albo $W_{ij}x_j > 0$ albo $W_{ij}x_j < 0$. Neuron radialny z kolei reprezentuje hipersferę, dokonującą podziału kołowego wokół punktu centralnego.

Sieć radialna a sieć sigmoidalna **Sieć sigmoidalna** Działanie funkcji rozciąga się od określonego punktu w przestrzeni aż do nieskończoności, reprezentuje aproksymację globalną funkcji zadanej. Nie ma niemożności fizycznego powiązania obszaru aktywności neuronu z odpowiednim obszarem danych uczących, trudności z określeniem optymalnego punktu startowego z procesie uczenia. **Sieć radialna** Bazuje na funkcjach mających wartość niezerową jedynie w określonej przestrzeni tylko wokół centrów, realizuje aproksymację typu lokalnego, której zasięg działania jest bardzo ograniczony. Można się spodziewać że zdolności do uogólniania są gorsze niż dla sieci sigmoidalnych. Łatwość powiązania parametrów funkcji bazowych z fizycznym rozmieszczeniem danych w obszarze parametrów. Łatwość uzyskania dobrych wartości startowych w procesie uczenia pod nadzorem.

Przestrzenie decyzyjne tworzone w sieciach radialnych są stosunkowo proste i w sposób naturalny kształtowane. Sieć dostarcza nie tylko informacji do jakiej klasy należy wzorzec testujący, ale wskazuje również na ewentualną możliwość utworzenia oddzielnej klasy. Na ogół uważa się, że sieci radialne lepiej niż sieci sigmoidalne nadają się do takich żądań klasyfikacyjnych jak wykrywanie uszkodzeń w różnego rodzaju systemach, rozpoznawanie wzorców, itp. Znaczną zaletą sieci radialnych jest znacznie uproszczony algorytm uczenia. Przy istnieniu tylko jednej warstwy ukrytej i ścisłym powiązaniu aktywności neuronu z odpowiednim obszarem przestrzeni danych uczących, punkt startowy uczenia jest znacznie bliżej rozwiązania optymalnego, niż jest to możliwe w sieciach wielowarstwowych.

Dodatkowo, możliwe jest odseparowanie etapu doboru parametrów funkcji bazowych od doboru wartości wag sieci (algorytm hybrydowy), co może przyśpieszyć i uprościć proces uczenia. Przy zastosowaniu ortogonalizacji proces optymalnego kształtowania struktury sieci jest stałym fragmentem uczenia, nie wymagającym żadnego dodatkowego wysiłku. Liczba neuronów ukrytych decyduje w dużym stopniu o dokładności odwzorowania i zdolnościach uogólniania sieci. W przypadku sieci radialnej problem doboru liczby neuronów ukrytych jest o wiele prostszy niż w sieciach sigmoidalnych, ze względu na lokalny charakter aproksymacji reprezentowany przez poszczególne funkcje bazowe.

Sieć RBF może być na różne sposoby hybrydyzowana - może być na przykład uczona algorytmem Kohonena i LVQ, co jest alternatywą do przypisywania centrów odzwierciedlającego rozkład danych. Warstwa wyjściowa (liniowa lub nie) może być uczona którymkolwiek z iteracyjnych algorytmów dla warstw z iloczynem skalarnym.

Dobór parametrów funkcji radialnych Znacznie lepsze rezultaty można uzyskać przez zastosowanie samoorganizujacego się procesu podziału danych uczących na klastry w jednej z jego licznych odmian. Centrum klastra jest utożsamiane z centrum odpowiedniej funkcji radialnej. Liczba tych funkcji jest równa liczbie klastrów i może być korygowana przez algorytm samoorganizacji. Proces podziału danych na klastry może być przeprowadzany metodą K-uśrednień. Aparat matematyczny zaangażowany w tą procedurę jest dość skomplikowany.

Sieć klasy GRNN Warstwa wejściowa, radialna, regresyjna oraz wyjściowa.

1. Wejściowe wektory uczące dzielone są na skupienia - w szczególnym przypadku każdy wektor tworzy oddzielne skupienie
2. Dla każdego skupienia znana jest wartość zmiennej objaśnianej (wyjście sieci)
3. wartość zmiennej objaśnianej dla dowolnego wektora wejściowego szacowana jest jako średnia ważona liczona z wartości tej zmiennej dla skupień - wagi uzależnione są od odległości wejścia od centrów skupień

8 Wykład 8 - sieci rezonansowe

Wagi w dół - pamięć krótkotrwała Wagi w górę - pamięć długotrwała

`http://iisi.pcz.pl/nn/rezonnsowe.php?art=5`

Głównym powodem dla którego w tym wykładzie nic nie ma jest przekroczenie poziomu chujowości wykładu ponad mój już i tak bardzo tolerancyjny próg. W związku z tym, że sieci rezonansowe w linku są ładnie opisane odsyłam do powyższego.

9 Wykład 9 - sieci rekurencyjne

Sprzężenie zwrotne w neuronie liniowym Po jednorazowym impulsie na wejściu na wyjściu otrzymamy długotrwały proces, w którym sygnał wyjściowy zmienia się wielokrotnie, aż osiągnie stan równowagi (jeśli go osiągnie). Równowaga w sieci może być osiągnięta (bez działającego sygnału wejściowego) jedynie w taki sposób, że sygnał wyjściowy po przemnożeniu przez wagę sprzężenia zwrotnego daje taki sam sygnał. Taki sygnał nazywamy ATRAKTOREM. Położenie atraktora jest związane z parametrami sieci. Dla współczynnika wagowego sprzężenia zwrotnego o wadze 1 każdy punkt jest atraktorem, natomiast dla dowolnej sieci stan równowagi uzyskujemy tylko wtedy, gdy sygnał wyjściowy ma wartość 0.

Jeśli wartość współczynnika wagi synaptycznej w obwodzie sprzężenia zwrotnego jest dodatnia to przebiegi mają charakter aperiodyczny (nie mają oscylacji). Wartości nie zmieniają znaku oraz są monotoniczne (rosną dla dodatnich, maleją dla ujemnych).

Jeśli wartości współczynnika wagi synaptycznej są ujemne to system ma charakter periodyczny.

Istnieje pojęcie niestabilności sieci. Jeśli sieci są stabilne to dążą do stanu równowagi.

Wartość sygnału wejściowego może być podawana cały czas.

Wnioski Przebieg sygnałów wyjściowych w sieci ze sprzężeniem zwrotnym może wykazywać dwojakiego rodzaju zmienność. W przypadku nieliniowego neuronu możliwe byłyby formy zachowania systemu char. dla systemów nielinowych to jest chaotyczne błądzenie *ze wszystkimi cudeńkami współczestnej teorii chaosu - efekt motyka, dziwne atraktory, fraktale, zbiory Mandelbrotta.*

Trzy warunki stabilności sieci Hopfielda

1. wprowadzono bardzo regularną strukturę wewnętrzną sieci - neurony są łączone każdy z każdym
2. zabroniono sprzężeń zwrotnych obejmujących jeden neuron
3. wprowadzone współczynniki wagowe muszą być symetryczne - jak x do y ma wagę w to y do x ma wagę w

Istnieje pojęcie funkcji energetycznej dla sieci Hopfielda. Wynika z niej, że zmiana y ma znak identyczny ze znakiem łącznego pobudzenia. Zmiana energii podczas aktualizacji wyjść jest zawsze niedodatnia.

METODY WYKORZYSTUJĄCE JEDNOKROTNĄ PREZENTACJĘ WZORCÓW
Metoda Hebba

$$t_{ij}^s = \begin{cases} 0, & \text{if} i = j \\ \frac{1}{N}\sum_{s=1}^{M} x_i^s x_j^s & \text{otherwise} \end{cases} \tag{57}$$

1. N - liczba bitów w obrazie wzorcowym
2. M - liczba wektorów wzorcowych
3. t_{ji}^s waga połączenia wyjścia j-tego neuronu z wejściem i-tego neuronu przy prezentacji s-tego obrazu wzorcowego

Istnieję wersje metody hebba dla macierzy, wzorców unipolarnych oraz wersja iteracyjna. Metoda wzajemnych ograniczeń - reguła Hebba ze składnikiem λ *odpychającym*

$$t_{ij}^s = \begin{cases} 0, & \text{if} i \neq j \\ \sum_{s=1}^{M} x_i^s x_j^s - \lambda \sum_{p \neq s}^{M} x_i^p x_j^s & \text{otherwise} \end{cases} \tag{58}$$

Istnieje również reguła rzutowania Δ oraz zmodyfikowana reguła perceptronu. Zmodyfikowana reguła perceptronu różni się od hebba, że w procesie uczenia dodano składnik bieżącej korekty błędów.

SIEĆ HOPFIELDA JAKO PAMIĘĆ SKOJARZENIOWA Tego typu sieci mogą działać jako pamięć autoasocjacyjna, czyli rozpoznają wzorce, którymi były uczone. Wykorzystanie takiej pamięci polega na tym, że potrafi ona odtworzyć obraz na podstawie obrazu silnie zniekształconego lub zakłóconego.

Ciekawostka Rozwiązywanie problemu TSP przy wykorzystaniu sieci Hopfielda jest mało efektywne. Nie istnieją reguły dopasowania parametrów sieci a ich dobór jest czasochłonny.