Sieci Neuronowe

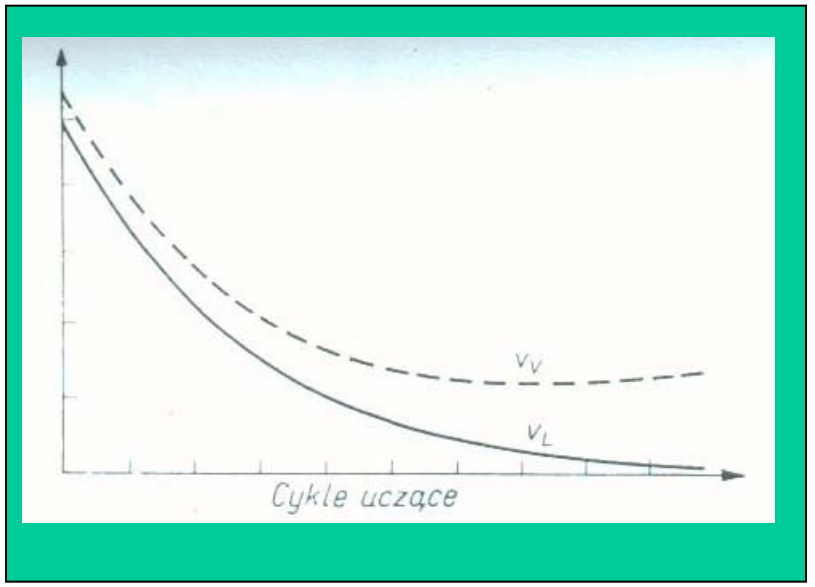
Wykład 9
Dobór optymalnej architektury i próbek uczących

wykład przygotowany na podstawie.
S. Osowski, “Sieci Neuronowe w ujęciu algorytmicznym ”, Rozdz. 3, PWNT, Warszawa 1996.

Zdolności uogólniania sieci neuronowej

Wpływ sposobu i czasu uczenia na zdolności uogólniania

W ogólnym przypadku wraz z upływem czasu uczenia błąd uczenia $v_L(W)$ maleje i błąd testowania $v_V(W)$ również (przy ustalonej wartości liczby próbek uczących p oraz miary *VCdim*). Od pewnego momentu błąd testowania pozostaje stały, natomiast błąd uczenia nadal maleje. W ostatnich fazach procesu uczenia nieregularności w danych odbiegające od cech charakterystycznych danego procesu zaczynają odgrywać role i powodują wzrost błędu testowania.



Tendencje te (przeuczenie) jest tym silniejsze im większe nadmiarowości wag występują w sieci. Te “niepotrzebne” wagi dopasowywują się do nieregularności danych uczących, traktując je jako cechę główną. Ważne jest aby kontrolować jak daleko jest zaawansowany proces uczenia, poprzez przeplatanie go z procesem testowania.

Zdolności uogólniania sieci neuronowej

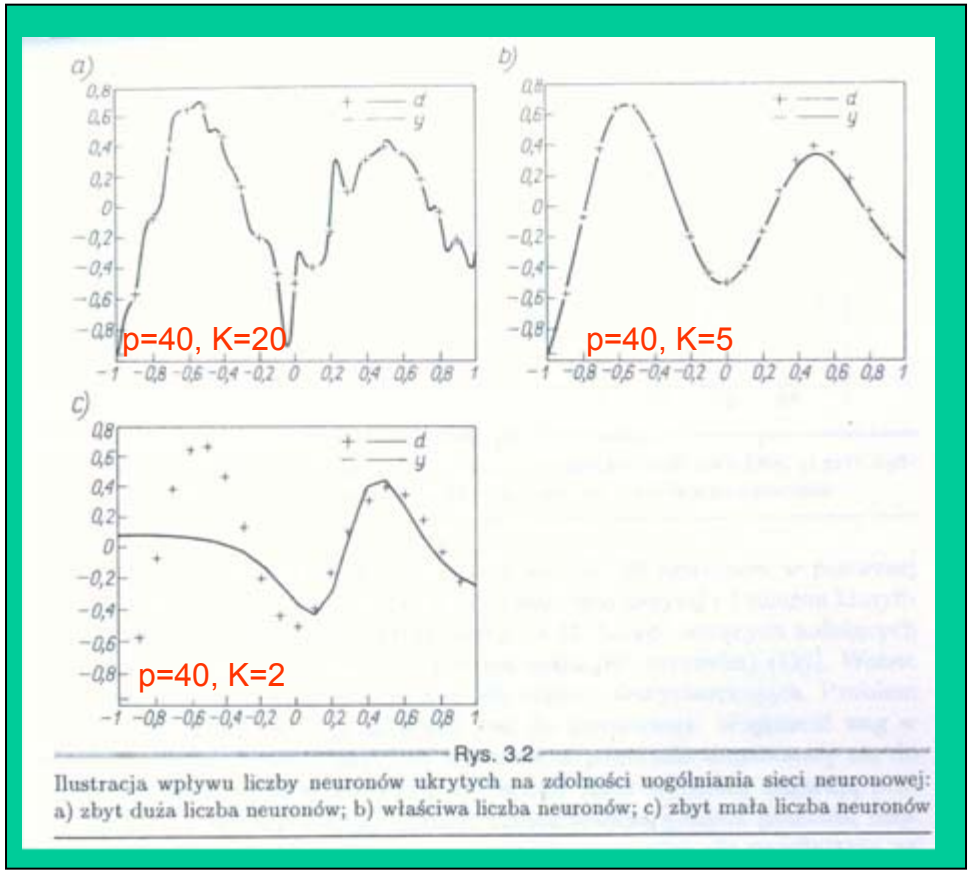
Błąd uogólniania może być oszacowany na podstawie błędu uczenia $v_L(W)$ oraz tzw. przedziału ufności ε_1

$$v_G(W) \leq v_L(W) + \varepsilon_1 (p/K, v_L)$$

Mała liczba próbek uczących przy ustalonej wartości h oznacza bardzo dobre dopasowanie sieci do próbek uczących ale złe uogólnienie bo w procesie uczenia nastąpił nadmiar parametrów dobieranych.

Zadanie aproksymacji zostało niejako sprowadzone do zagadnienia interpolacji.

p-ilosc probek, K – liczba neuronow w warstwie ukrytej



Rys. 3.2
Ilustracja wpływu liczby neuronów ukrytych na zdolności uogólniania sieci neuronowej: a) zbyt duża liczba neuronów; b) właściwa liczba neuronów; c) zbyt mała liczba neuronów

Rosądnym rozwiązaniem jest wówczas redukcja stopnia złożoności sieci prowadzaca do zmniejszenia miary *VCdim*.

Zdolności uogólniania sieci neuronowej

Sam proces uczenia powinien być powiązany ze sprawdzaniem zdolności do uogólniania, a więc powinien zawierać “fazę uczącą” i “fazę sprawdzającą”. Proces uczenia kontynuuje się do chwili uzyskania minimum funkcji celu lub dopóki błąd testowania nie zacznie wzrastać (wskazując na przeuczenie).

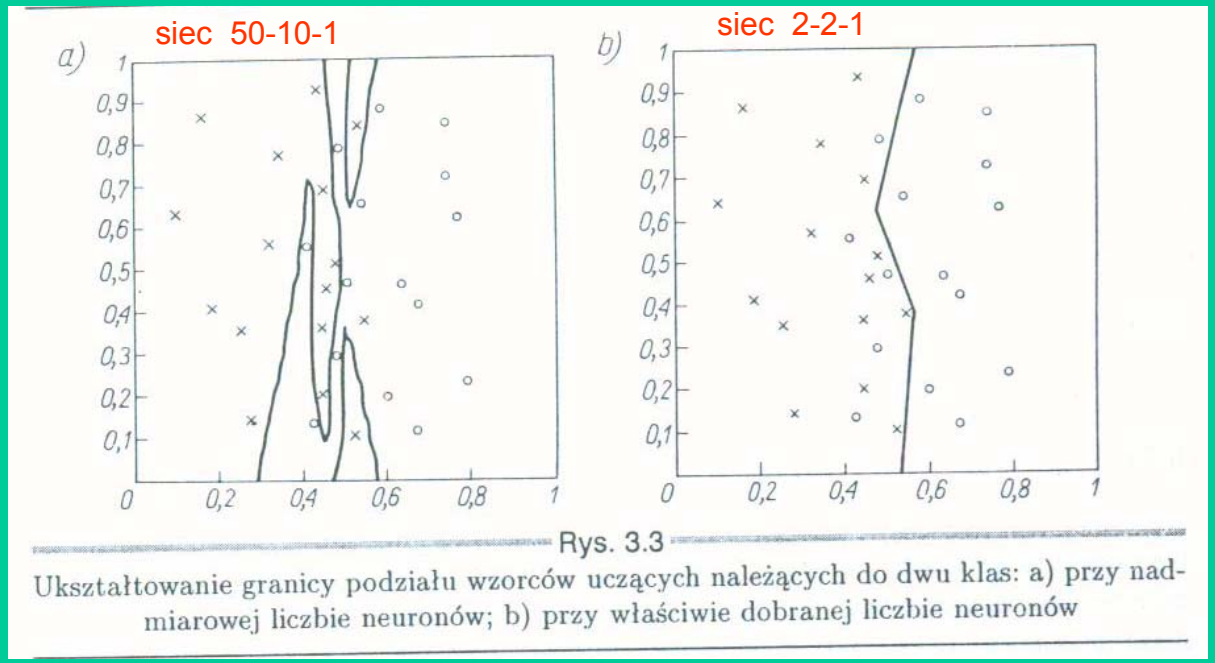
Dobór liczby neuronów w warstwie (warstwach) ukrytych jest kluczowym zagadnieniem, decydującym o właściwościach uogólniających sieci.
Są możliwe dwa kierunki działań:

→ **Zakłada się wstępną liczbę neuronów ukrytych**, opartą bądź na teorii Kolmogorowa, bądź na dotychczasowych doświadczeniach, a następnie przeprowadza się redukcję w trakcie uczenia sieci.

→ **Startuje się z minimalną liczbą neuronów ukrytych** i stopniowo następuje proces ich dodawania aż do uzyskania dobrego stopnia wytrenowania na zbiorze uczącym. Proces dodawania jest zazwyczaj połączony ze sprawdzaniem zdolności do uogólniania sieci na podzbiorze V.

Metody redukcji sieci

Zadaniem redukcji sieci jest zmniejszanie liczby neuronów ukrytych oraz powiązań miedzy neuronowych. Uzyskuje się w ten sposób poprawę zdolności uogólniania.



Rys. 3.3
Ukształtowanie granicy podziału wzorców uczących należących do dwu klas: a) przy nadmiarowej liczbie neuronów; b) przy właściwie dobranej liczbie neuronów

Wektor wejsciowy (x1, x2). Siec 50-10-1, 671 wag, 31 danych. Zbyt mala ilosc danych uczacych. W procesie uczenia wiekszosc wag dobrana dowolnie, przypadkowe dopasowanie do nieistotnych szczegolow.

Metody redukcji sieci

Podstawę redukcji sieci (*pruning*) stanowią algorytmy podejmujące decyzje co do obcięcia wagi lub redukcji neuronów w trakcie procesu uczenia. Większość stosowanych obecnie algorytmów może być zakwalifikowana do dwóch grup:

→ **Szacuje się wrażliwość funkcji względem wagi lub neuronu**. Wagi o najmniejszej wrażliwości, wpływając najmniej na funkcje celu, są usuwane, a proces uczenia kontynuowany na tak zredukowanej sieci.

→ **Modyfikuje się funkcję celu wprowadzając kary** za nieefektywną strukturę. Najczęściej do definicji funkcji celu wprowadza się składniki faworyzujące małe amplitudy wag, zmuszając algorytm uczący w trakcie uczenia do ich ciągłej redukcji. Metoda ta jest mniej efektywna niż pierwsza, bo małe wartości wag niekoniecznie muszą oznaczać mały ich wpływ na działanie sieci.

Metody wrażliwosciowe redukcji

Parametrem, na podstawie którego podejmuje się decyzje co do eliminacji wagi (redukcji złożoności sieci) jest wrażliwość funkcji celu na dane połączenie synaptyczne.
Do określenia wrażliwości neuronu wprowadzamy współczynnik α_i dla każdej wagi. Wyjściowy sygnał –tego neuronu określa się na podstawie zmodyfikowanej zależności

$$y_i = f\left(\sum_j W_{ij}\, \alpha_j\, y_j \right)$$

w której W_{ij} jest waga od j-tego do i-tego neuronu, y_i oraz y_j oznaczają sygnały wyjściowe odpowiednich neuronów a f() oznacza funkcje aktywacji. Przy wartości $\alpha_i = 0$ nie ma połączenia W_{ij}, przy $\alpha_j = 1$ występuje stan normalny pracy sieci.

Metody wrażliwosciowe redukcji

Ważność połączenia synaptycznego opisanego wagą W_{ij} , jest oceniana na podstawie wrażliwości bezwzględnej funkcji celu E względem współczynnika α_j.

$$\rho_j = - \partial E / \partial \alpha_j$$

dla wartości $\alpha_j = 1$. Jest to równoznaczne wyznaczeniu składnika gradientu funkcji celu względem wagi $W_{ij,}$ określanym zwykłą metodą propagacji wstecznej. Waga $W_{ij,}$ jest obcinana jeżeli wartość ρ_j zmniejszy się poniżej określonego progu.

Metody wrażliwosciowe redukcji

Zwykle, dla zapewnienia stabilności procesu wartość współczynnika ρ_j w k-tym cyklu oblicza się w postaci skumulowanej, korzystając ze wzoru

$$\rho_j(k) = 0.8\ \rho_j(k-1) + 0.2\ \partial E / \partial \alpha_j$$

Inna metoda redukcji wrażliwości, przyjmuje miarę półwzględną współczynnika wrażliwości S_{ij}, zdefiniowaną w postaci

$$S_{ij} = -\left(E(W_f) - E(0)\right) / \left(W_{ij,f} - W_{ij,0}\right) W_{ij,f}$$

gdzie W_f oznacza wektor końcowy wag sieci (po zakończeniu procesu uczenia), $W_{ij,0}$ jej zerową wartość po usunięciu z sieci, $E(W_f)$ jest oznaczeniem wartości funkcji celu po zakończeniu procesu uczenia, a $E(0)$ wartością funkcji celu po zakończeniu procesu uczenia i usunięciu wagi W_{ij}.

Metody wrażliwosciowe redukcji

Zamiast dodatkowych obliczeń wymaganych do wyznaczenia funkcji wrażliwości S_{ij}, stosuje się jej aproksymacje, uwzględniającą wszystkie zmiany wagi w procesie uczenia. Przybliżona wartość S_{ij} jest określana ze wzoru

$$S_{ij} \approx - \sum_{k=1}^{n_c} \partial E / \partial W_{ij} \; \Delta W_{ij}(k) \, / (W_{ij,f} - W_{ij,0})$$

Po przeprowadzeniu procesu uczenia sieci każda waga W_{ij} ma określoną skumulowaną wartość wrażliwości S_{ij}. Połączenia synaptyczne o najmniejszych wartościach S_{ij} są usuwane, a sieć po redukcji podlega powtórnemu douczeniu.

W obu przedstawionych metodach jest możliwe usunięcie neuronu z warstwy, jeśli wszystkie wagi dochodzące lub odchodzące od niego zastaną wyeliminowane.

Metoda ODB (Optimal Brain Damage)

Punktem wyjścia jest rozwinięcie funkcji celu w szereg Taylora w otoczeniu aktualnego rozwiązania.

$$\Delta E = \Sigma g_i \Delta W_i + \tfrac{1}{2} [\Sigma h_{ii} [\Delta W_{ii}]^2 + \sum_{i \neq j} h_{ij} \Delta W_i \Delta W_j] + O(|| \Delta W ||^2)$$

w którym ΔW_i oznacza perturbacje wagi i-tej, g_i – i ty wskaźnik wektora gradientu względem tej wagi, $g_i = \partial E / \partial W_i$, $h_{ij} = \partial^2 E / \partial W_i \partial W_j$.

Ponieważ obcinanie wag dotyczy sieci już wytrenowanej, składowe gradientu są bliskie zeru (wytrenowanie oznacza że minimum funkcji celu zostało osiągnięte) i mogą zostać pominięte w rozwinięciu. Ostatni składnik również może zostać pominięty. Otrzymujemy więc przybliżony wzór

$$\Delta E \approx \tfrac{1}{2} [\Sigma h_{ii} [\Delta W_{ii}]^2 + \sum_{i \neq j} h_{ij} \Delta W_i \Delta W_j]$$

Metoda ODB (Optimal Brain Damage)

Dla uproszczenia przyjmuje się że tylko diagonalne elementy h_{ji} są istotne. Miarą ważności danego połączenia synaptycznego pozostaje współczynnik $S_{ij} = ½ \partial^2 E / \partial W^2_{ij} W^2_{ij}$.

Obcięciu podlegają wagi o najmniejszej wartości tego współczynnika.

Procedura ODB

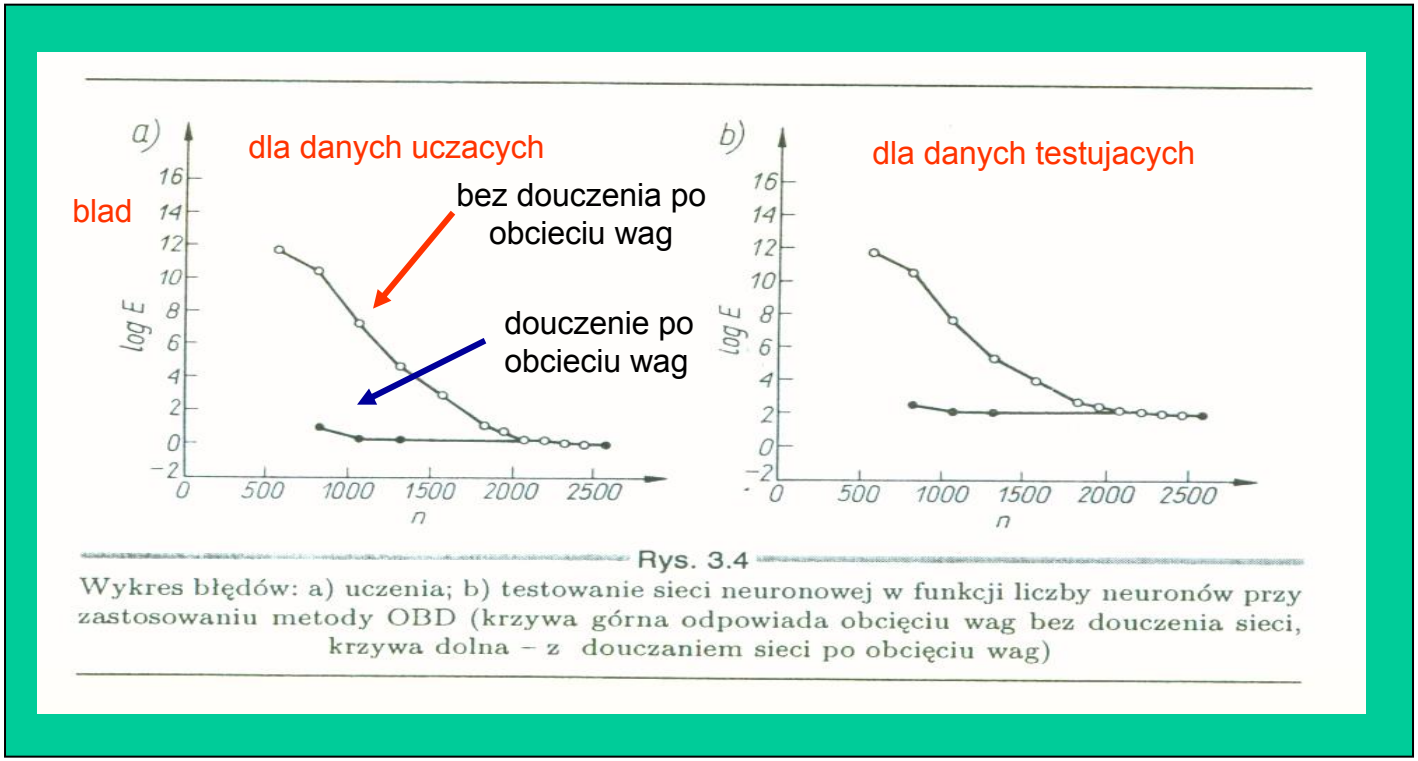
Procedurę ODB redukcji sieci można przedstawić następująco.

1. Selekcja wstępna struktury sieci neuronowej (wstępny wybór liczby neuronów w poszczególnych warstwach).
2. Przeprowadzenie programu uczenia tak dobranej sieci przy zastosowaniu dowolnej metody gradientowej uczenia.
3. Określenie elementów diagonalnych $h_{kk} = \Sigma\, \partial^2 E / \partial W^2_{ij}$ odpowiadających każdej wadze W_{ij} sieci (sumowanie po wszystkich połączeniach synaptycznych którym przypisana jest waga W_{ij}, “shared weight”).
4. Obliczenie parametru $S_{ij} = ½\, h_{kk}\, W_{ij}$ określającego znaczenie danego połączenia synaptycznego dla działania sieci.
5. Posortowanie wag wg. przypisanych im parametrów S_{ij} i obcięcie tych których wartości są najmniejsze.
6. Kilkakrotne powtórzenie procedury 2-5.

Metoda ODB uważana jest za jedną z najlepszych metod redukcji sieci spośród metod wrażliwosciowych.

Procedura ODB

Przykład zastosowania procedury ODB dla sieci rozpoznającej ręcznie pisane kody pocztowe.
Sieć miała 10^5 połączeń synaptycznych, którym zostało przypisane 2578 różnych wag (część wag była wspólna).



Rys. 3.4
Wykres błędów: a) uczenia; b) testowanie sieci neuronowej w funkcji liczby neuronów przy zastosowaniu metody OBD (krzywa górna odpowiada obcięciu wag bez douczenia sieci, krzywa dolna – z douczaniem sieci po obcięciu wag)

Procedura ODB

Przy zastosowaniu metody ODB uzyskuje się bardzo dobre własności uogólniające sieci, niewiele odbiegające od błędu uczenia. Szczególnie dobre wyniki uzyskuje się dzięki powtórzeniu douczenia sieci po obcięciu najmniej znaczących wag.

Ulepszeniem metody ODB jest opracowana 3-lata później *metoda OBS (Optimal Brain Surgeon).* Punktem wyjścia jest rozwinięcie w szereg Taylora (podobnie jak w metodzie ODB) .
Podstawowa różnica metody OBS w stosunku do ODB jest inna definicja współczynnika asymetrii (który służy do podjęcia decyzji o eleminacji danego połączenia synaptycznego) oraz korekta wag sieci po wyeliminowaniu wagi o najmniejszym znaczeniu. Osiągnięte uprzednio minimum zostaje zachowane.
Metoda ta ma znacznie większą złożoność obliczeniową.

Metody funkcji kary

W odróżnieniu od metod, kiedy obcięcie wag następowało w wyniku określenia wrażliwości funkcji celu na brak danej wagi, w metodach funkcji kary modyfikuje się samą funkcje celu w taki sposób, aby proces uczenia samoczynnie eliminował najmniej potrzebne wagi.

Eliminacja następuje przez stopniowe zmniejszanie wartości wag, aż do osiągnięcia pewnego progu, poniżej którego przyjmuje się wartość wagi równą zero.
Najprostszą metodą modyfikacji funkcji celu jest dodanie do niej składnika kary za duże wartości wagi.

$$E(W) = E^0(W) + \gamma \Sigma W^2_{ij}$$

W tym wzorze $E^0(W)$ oznacza standartową definicję funkcji celu, np. $E(W) = \frac{1}{2} \Sigma (y_i^{(k)} - d_i^{(k)})^2$, a γ oznacza współczynnik kary za osiągnięcie dużych wartości przez wagi.

Metody funkcji kary

Uczenie składa się więc z dwóch etapów:

→ minimalizacji wartości funkcji $E^0(W)$ standartową metodą propagacji wstecznej

→ korekcji wartości wag, wynikającej z czynnika modyfikującego

Jeśli przez W^0_{ij} oznaczamy wartości wagi W_{ij} po etapie pierwszym, to po korekcji waga ta zostanie zmodyfikowana według metody gradientowej największego spadku zgodnie ze wzorem

$$W_{ij} = W_{ij} (1 - \eta\gamma)$$

gdzie η oznacza stałą uczenia.

Tak zdefiniowana funkcja kary wymusza zmniejszenie wszystkich wartości wag, nawet wówczas, gdy ze względu na specyfikę problemu pewne wagi powinny osiągać duże wartości. Poziom wartości, przy których eliminuje się daną wagę musi być starannie dobrany na podstawie wielu eksperymentów, wskazujących przy jakim progu obcięcia proces uczenia sieci jest najmniej zakłócany.

Metody funkcji kary

Lepsze efekty, nie powodujące obniżenia poziomu wartości wszystkich wag, można uzyskać modyfikując definicję funkcji celu do postaci

$$E(W) = E^0(W) + 1/2\gamma \Sigma W^2_{ij}/(1+W^2_{ij})$$

W takim wypadku uzyskuje się korekcję wagi w postaci

$$W_{ij} = W_{ij}^0 (1 - \eta\gamma/[1+W^{0}_{ij}{}^2]^2)$$

Przy małych wartościach wag W_{ij} ($W_{ij} << 1$) oba wzory na korekcje są równoważne.

Przy dużych wartościach wag czynnik modyfikujący wagi jest pomijalnie mały i modyfikacja funkcji, praktycznie biorąc, ma niewielki wpływ na dobór wag.

Metody funkcji kary

Podobną metodę można zastosować do usuwania niepotrzebnych neuronów w warstwie. Za takie uważa się neurony, których sygnał wyjściowy jest bliski zeru. Do eliminacji liczby neuronów stosuje się modyfikację funkcji celu w postaci

$$E(W) = E^0(W) + 1/2\gamma \sum_{i,j} W^2_{ij}/(1+ \sum_{k} W^2_{ik})$$

Minimalizacja tej funkcji redukuje nie tylko powiązania między neuronowe, ale prowadzi również do eliminacji tych neuronów, dla których $\sum_{k} W^2_{ik}$ jest bliska zeru. Można udowodnić, że reguła korekcyjna wagi może być wyrażona wzorem

$$W_{ij} = W_{ij}^0 \left(1 - \eta\gamma \ (1+ \sum_{k\neq j} W^2_{ik}{}^{(0)}) /[(1+ \sum_{k} W^2_{ik}{}^{(0)})]^2 \right)$$

Przy małych wartościach wag W_{ik} prowadzących do i-tego neuronu następuje dalsze zmniejszenie ich wartości i w efekcie eliminacja neuronu z sieci.

Metody funkcji kary

Inny sposób redukcji neuronów zakłada taką modyfikację funkcji celu która eliminuje neurony ukryte o najmniejszej zmianie aktywności w procesie uczenia. Przyjmuje się tu założenie, że jeśli sygnał wyjściowy określonego neuronu dla wszystkich wzorców uczących jest niezmienny (wartość sygnału wyjściowego stale na poziomie zera lub 1), to jego obecność w sieci jest niepotrzebna. Przy dużej aktywności neuronu przyjmuje się że jego działalność wnosi istotną informację.
Chauvin zaproponował następującą modyfikację funkcji celu:

$$E(W) = E^0(W) + \mu \Sigma \Sigma e(\Delta^2_{i,j})$$

w której $\Delta^2_{i,j}$ oznacza zmianę wartości sygnału neuronu wyjściowego neuronu i-tego, przy j-tej próbie uczącej, a $e(\Delta^2_{i,j})$ stanowi czynnik korekcyjny funkcji celu. Postać funkcji korekcyjnej dobiera się tak, aby korekta była mała przy dużej aktywności neuronu, np. $e=1/(1+ \Delta^2_{i,j})^n$. Następnie eliminuje się neurony najmniej aktywne.

Metody funkcji kary

Obie metody redukcji sieci, opierające się zarówno na metodach wrażliwościowych jak i modyfikacjach funkcji celu, prowadzą do zmniejszenia liczby wag i neuronów w sieci, zmniejszając w ten sposób stopień jej złożoności i poprawiając relacje pomiędzy liczbą próbek uczących a miarą VCdim.

W efekcie eliminacji neuronów wzrasta zdolność uogólniania sieci.

Przyjmuje się że metody wrażliwościowe, zwłaszcza wykorzystujące drugą pochodną funkcji celu, są doskonalszym narzędziem redukcji sieci, gdyż o eliminacji decyduje nie wartość wagi, ale jej stopień ważności dla danej sieci, wyrażony miarą współczynnika asymetrii.

Niemniej jednak nawet w tym przypadku małe wartości wag są bardziej podatne na eliminacje. Wynika to z samej idei metod QBD i QBS o eliminacji w pierwszej kolejności małych wag.

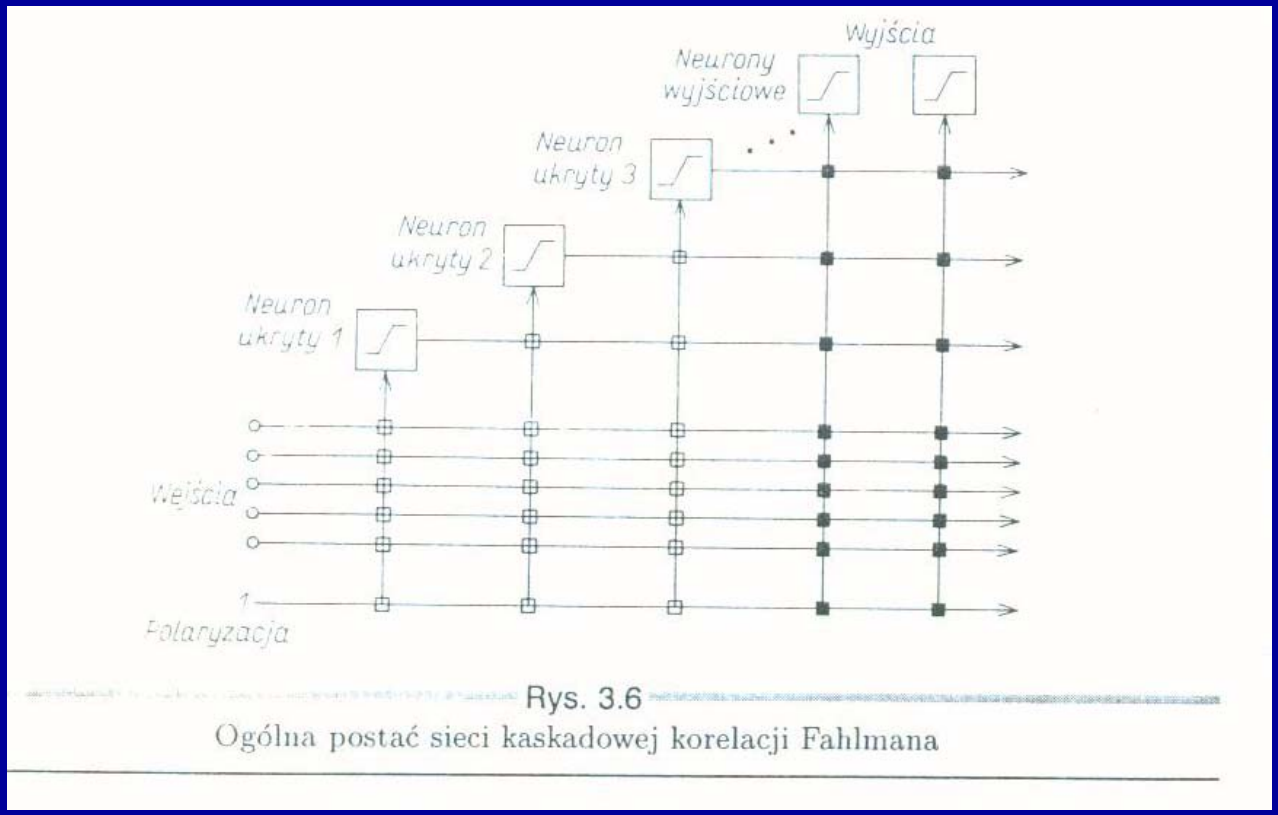
Algorytm kaskadowej korelacji Fahlmana

Algorytmy redukcji zakładały nadmiarową architekturę sieci, która w procesie uczenia lub po jego zakończeniu ulegała uproszczeniu bądź przez obcięcie mniej znaczących wag, bądź przez eliminację neuronów ukrytych.
Innym podejściem jest metoda uczenia która zakłada na wstępie małą liczbę neuronów ukrytych (często zerową), która w miarę postępów w uczeniu podlega stopniowemu zwiększaniu.
Istotną cechą *algorytmu kaskadowej korelacji Fahlmana* jest

→ kaskadowa architektura sieci neuronowej, w której połączenia neuronów są w postaci rozwijającej się kaskady połączeń wagowych. Kolejno dokładany neuron ma połączenia w węzłami wejściowymi i wszystkimi już istniejącymi neuronami ukrytymi;

→ metodzie doboru wag każdego kolejno dodawanego neuronu ukrytego, polegającej na maksymalizacji korelacji między jego sygnałem wyjściowym a residuum błędu odwzorowania przez sieć sygnałów zadanych.

Algorytm kaskadowej korelacji Fahlmana



Rys. 3.6
Ogólna postać sieci kaskadowej korelacji Fahlmana

Każdy, kolejno dokładany neuron ma połączenia z węzłami wejściowymi i wszystkimi już istniejącymi neuronami ukrytymi. Wyjścia wszystkich neuronów ukrytych i węzły wejściowe sieci zasilają bezpośrednio neurony wyjściowe.

Algorytm kaskadowej korelacji Fahlmana

Powstał jako wyzwanie na dwa podstawowe problemy: "poruszającego się celu" i "stada" , które pojawiają się przy typowej architekturze sieci neuronowej:

→ efekt *"poruszającego się celu"*

zmiany wag neuronów zachodzą równolegle, bez koordynacji na podstawie stale zmieniających się wzorców uczących. Każdy neuron widzi tylko strzęp informacji: przetworzony fragment informacji wejściowej i sygnał błędu propagacji wstecznej. Zamiast szybkiej zmiany wag w jednym kierunku obserwuje się "nieskoordynowane zamieszanie" aby dopasować się do zmieniającego się celu.

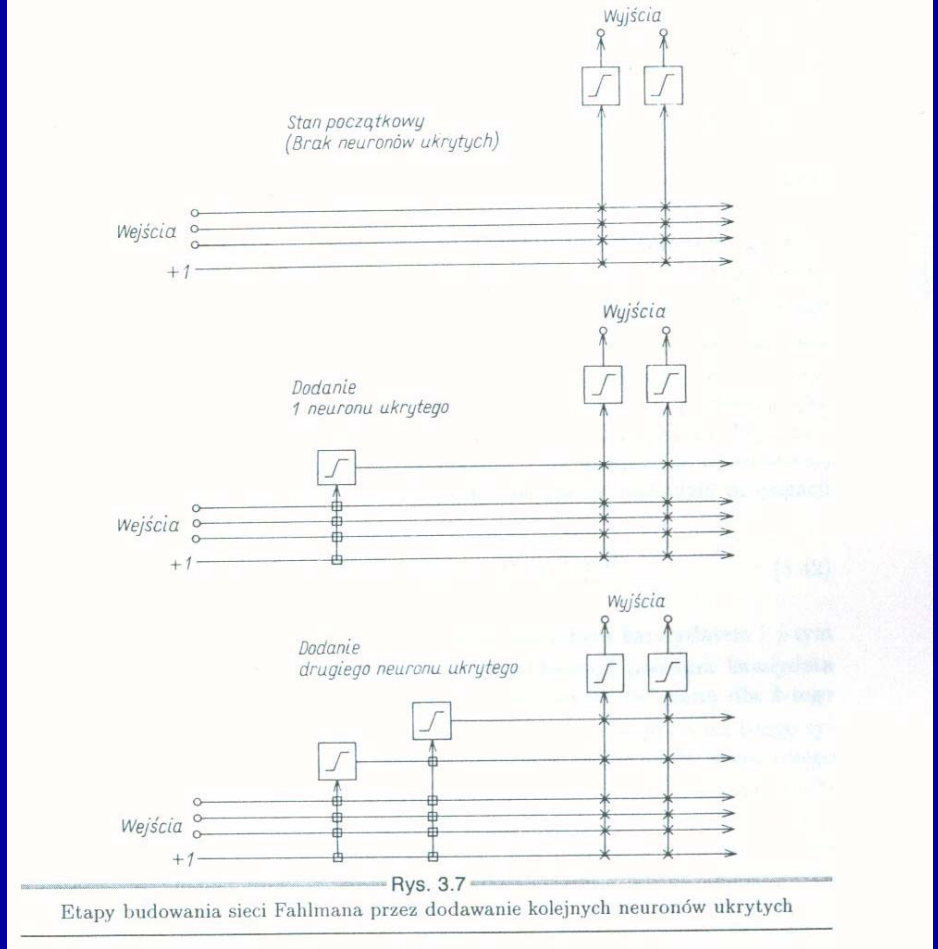
→ efekt *"stada"*

zakłada że są do wykonania zadania A i B które może wykonać każdy neuron. Najpierw neurony skupiają się na trudniejszy zadaniu (wszystkie) a potem usiłują się dopasować do łatwiejszego, psując rozwiązanie dla trudniejszego. Ostatecznie dzielą się na grupę, jedna dopasowana do A a druga do B, podział ten jest nieoptymalny.

Algorytm kaskadowej korelacji Fahlmana

Jedną z metod przeciwdziałania efektom *"stada"* i *"poruszającego się celu"* jest przypisanie na każdym etapie uczenia aktywnej roli tylko niektórym neuronom i połączeniom wagowym. Pozostałe wagi nie podlegają uczeniu.

Algorytm kaskadowej korelacji Fahlmana realizuje te strategie w ten sposób, że na każdym etapie uczenia tylko jeden neuron ukryty podlega zmianom. Połączenia oznaczone kwadratem ulęgają zamrożeniu, oznaczone krzyżykiem są uczone.



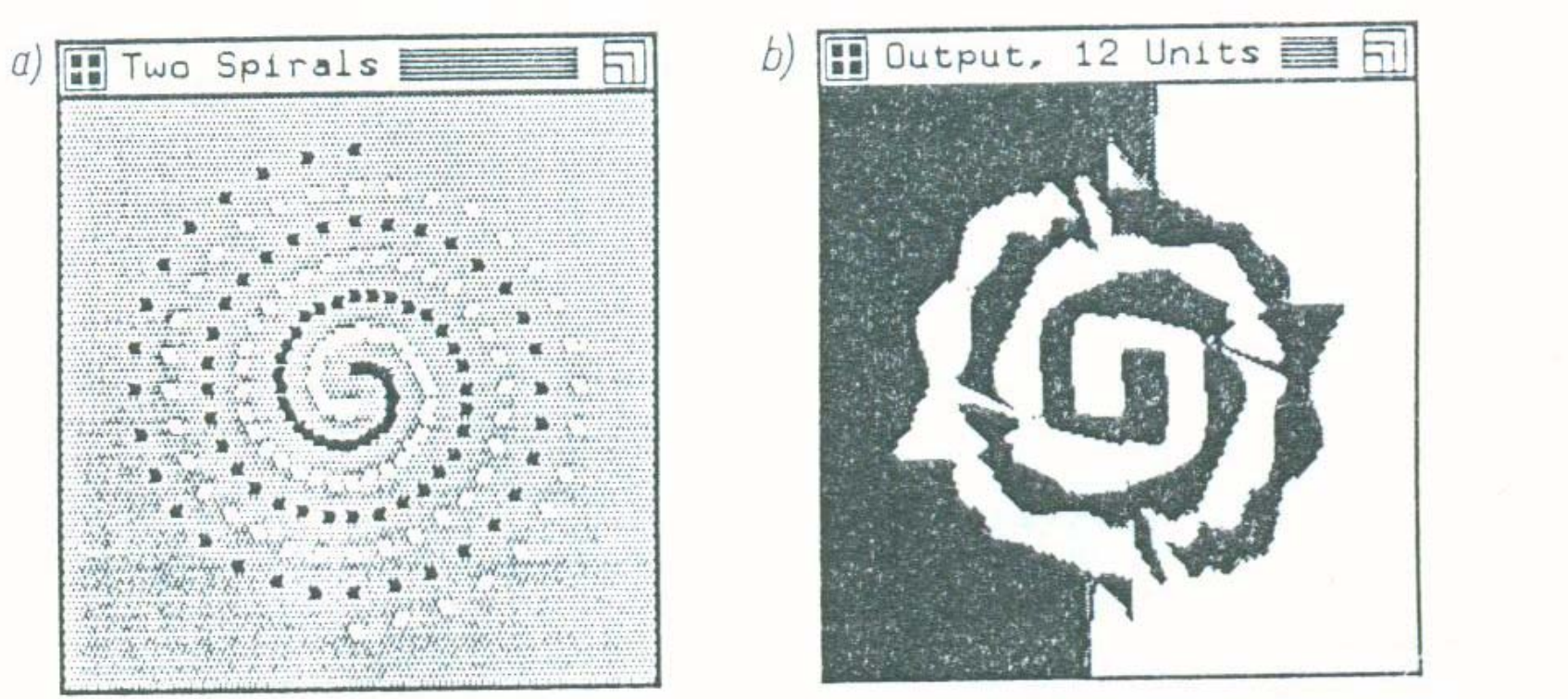
Rys. 3.7
Etapy budowania sieci Fahlmana przez dodawanie kolejnych neuronów ukrytych

Algorytm kaskadowej korelacji Fahlmana

→ Na etapie początkowym zakłada się sieć zawierającą jedynie wejścia i neurony wyjściowe.
→ Liczba wejść/wyjść zależy od specyfikacji problemu, nie podlega modyfikacji.
→ Każde wejście jest połączone z każdym wyjściem za pomocą wagi podlegającej uczeniu.
→ Następuje proces uczenia przy braku neuronów ukrytych.
→ Po określonej liczbie cykli uczących bez poprawy wyniku (liczba cykli definiowana jako “patience”), następuje dodanie jednego neuronu warstwy ukrytej którego wagi są zamrożone.
→ Kolejna iteracja uczenia sieci , uczeniu podlegają tylko wagi neuronów wejściowych i wyjściowy
→ Specjalna procedurę przygotowywania zamrożonych wag dodawanych neuronów ukrytych. Neuron jest uczony poza systemem, jako odrębna jednostka, zalecane uczenie kilku neuronów na raz z różnymi funkcjami aktywacji i wybranie najlepszego.

Algorytm kaskadowej korelacji Fahlmana

Bardzo dobry wynik



Rys. 3.8

Rezultaty działania sieci Fahlmana na przykładzie problemu 2 spiral: a) rozkład danych uczących należących do 2 klas; b) ukształtowanie się granic podziału przestrzeni danych

Sieć neuronowa z rozszerzeniem funkcyjnym

W odwzorowaniu danych uczących poprzez sieć wielowarstwową neurony warstw ukrytych odgrywają rolę elementów poszerzających informacje o położeniu konkretnego wzorca uczącego w przestrzeni parametrów i umożliwiają podjęcie decyzji przez neurony wyjściowe. Taki model przetwarzania informacji jest wystarczający w przypadku, gdy wymiar N wektora wejściowego jest w odpowiedniej proporcji do wymiaru M wektora wyjściowego sieci.

W przypadku gdy N << M, powstaje problem odwróconej piramidy, skąpa informacja wejściowa jest niewystarczająca do odtworzenia informacji wyjściowej mimo istnienia układu pośredniczącego w postaci warstw ukrytych. Zadanie staje się źle uwarunkowane, zdolności do uogólniania takiej sieci są zwykle małe.

Sieć Pao

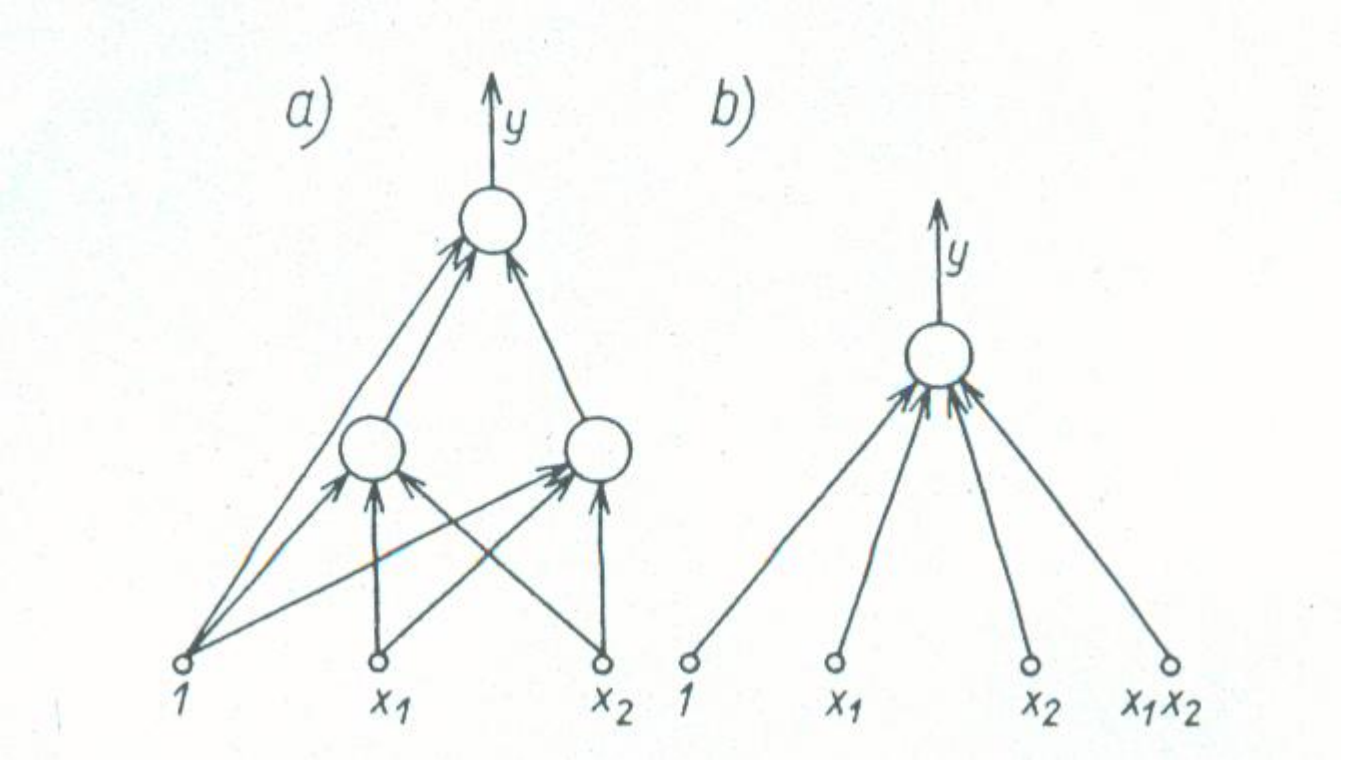
Drugi problem to stopień złożoności sieci. Istnienie wielu warstw ukrytych o pełnych powiązaniach między neuronowych może prowadzić nawet przy dobrym uwarunkowaniu zadania do złego uogólniania, a przy tym proces uczenia znacznie wydłuża się ze względu na dużą liczbę trenowanych wag.
Pewne rozwiązanie problemu to stosowanie
sieci neuronowej z rozszerzeniem funkcyjnym (functional link net)
w której sygnały wejściowe sieci x_i są zdublowane przez dodatkowe wprowadzenie ich odpowiedników funkcyjnych. Sieci takie noszą nazwę sieci wyższego rzędu i zostały wprowadzone przez Pao.
Stosowane są dwa rodzaje rozszerzeń funkcyjnych:
→ bazujące na funkcjach pojedynczych sygnałów x_i
→ bazujace na iloczynach tensorowych składowych wektora wejściowego.

Sieć Pao

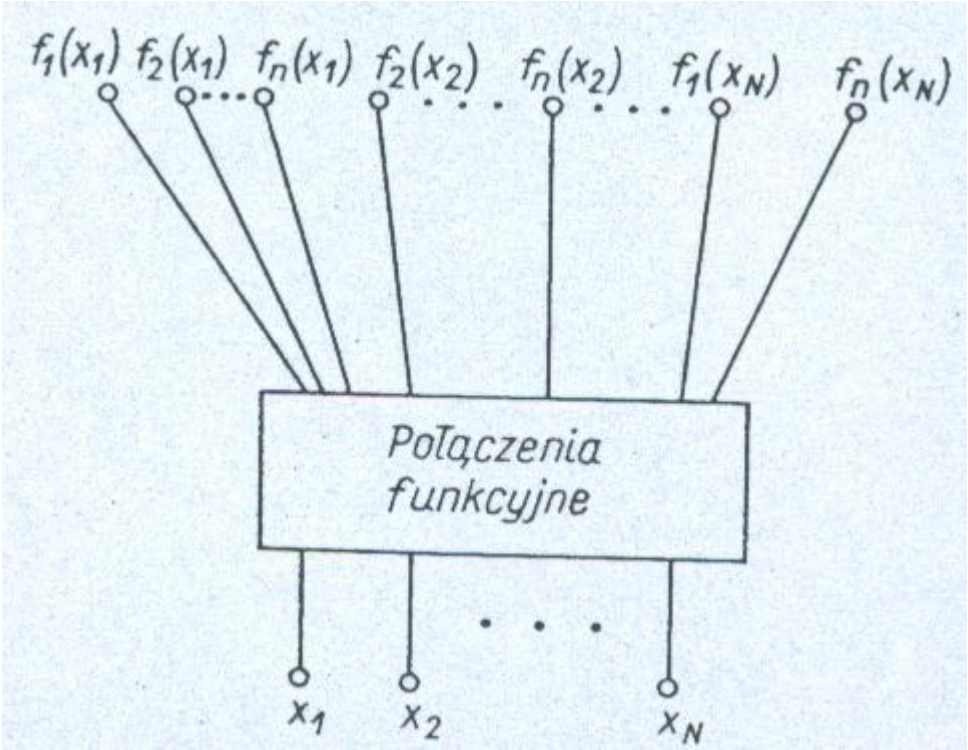
Sieci bazujące na funkcjach pojedynczych sygnałów x_i

Reprezentacja rozszerzona składa się ze zbioru oryginalnego x_i oraz zbioru funkcji pojedynczych elementów x_i. Przykładami funkcji rozszerzających może być funkcja wielomianowa, funkcje ortogonalne: $\sin\pi x$, $\cos\pi x$, $\sin 2\pi x$, $\cos 2\pi x$, itp.
Efektem takiego rozszerzenia jest rzutowanie wzorców wejściowych z przestrzeni N-wymiarowej w przestrzeń o większych wymiarach. Nie wprowadza to nowej informacji ale wzbogaca istniejącą.

Sieć Pao



Sieć Pao


$f_1(x_1)$ $f_2(x_1)$ $f_n(x_1)$ $f_2(x_2)$ $f_n(x_2)$ $f_1(x_N)$ $f_n(x_N)$
Połączenia funkcyjne
x_1 x_2 x_N

Sieć Pao

Sieci bazujace na rozszerzeniu iloczynowym

Podobny efekt uzyskuje się przez zastosowanie rozszerzenia iloczynowego działającego na kilku składowych wektora **X** jednocześnie. W modelu tym reprezentacja

$\{x_i\}$

jest rozszerzona przez uwzględnienie kolejnych iloczynów

$\{x_i, x_i x_j, x_i x_j x_k, ...\}$

Uwzględnienie łącznego oddziaływania wielu składowych wektora x jednocześnie umożliwia wzbogacenie informacji o procesach zachodzących w sieci. Proces iloczynowego wzbogacania liczby wejść ma tendencje narastania lawinowego.

Można temu przeciwdziałać eliminując te składniki dla których $\Sigma(x_i x_j)_k \rightarrow 0$. W praktyce rozszerzenie iloczynowe stosuje się zwykle do drugiego lub trzeciego rzędu, ograniczając lawinowe narastanie składników rozszerzonego wektora wejściowego.

Sieć Pao

Efektem wzbogacenia informacji wejściowej dostarczanej sieci jest zwykle znaczne przyspieszenie procesu uczenia, a przy tym uproszczenie architektury. W wielu wypadkach wystarczy użycie jednej warstwy neuronów.

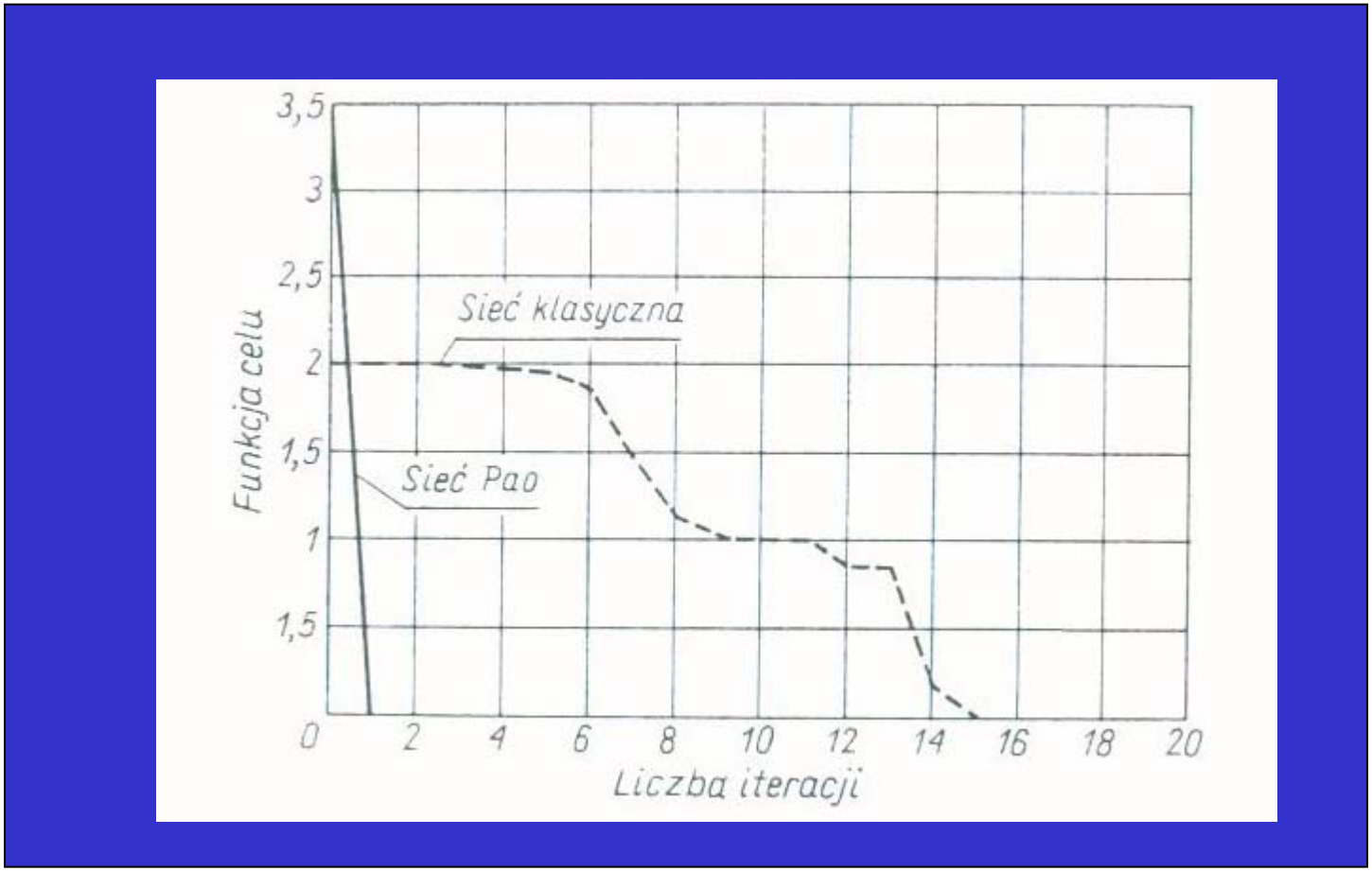
Przyklad: problem XOR

Dane uczące w postaci par (x,d):

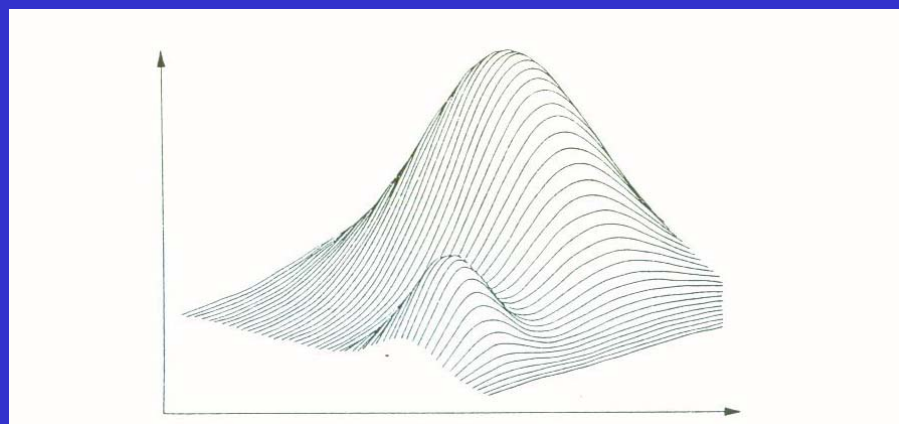
([-1,-1],-1), ([-1,-1], 1), ([1,-1], 1) oraz ([1, 1],-1),

w standartowym rozwiazaniu nie mogą być odwzorowywane przez sieć jednowarstwową i jest wymagane użycie jednej warstwy ukrytej. Stosując rozszerzenie funkcyjne typu iloczynowego uzyskuje się rozwiązanie problemu bez stosowania warstwy ukrytej z jednym neuronem typu sigmoidalnego na wyjściu. Liczba wag sieci została zredukowana z 9 (w klasycznym rozwiazaniu) do 4 w rozwiazaniu Pao. Jeszcze większą różnicę można zaobserwować w procesie uczenia.

Sieć Pao

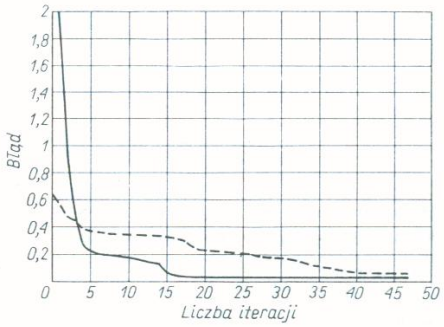

3,5
3
2,5
2
1,5
1
1,5
0
2
4
6
8
10
12
14
16
18
20
Funkcja celu
Liczba iteracji
Sieć klasyczna
Sieć Pao

Sieć Pao



Rys. 3.12

Postać funkcji trójwymiarowej poddanej odwzorowaniu za pomocą sieci neuronowej trójwarstwowej i sieci jednowarstwowej Pao



Rys. 3.13

Krzywe uczenia sieci przy odwzorowaniu funkcji z rys. 3.12 (linia ciągła – sieć Pao, linia przerywana – sieć trójwarstwowa)

Funkcja dwóch zmiennych (x1, x2) poddana odwzorowaniu za pomocą standartowej sieci dwuwejściowej o jednej warstwie ukrytej zawierającej 15 neuronów oraz sieci jednowarstwowej (bez warstwy ukrytej), ale w zamian z zastosowaniem rozszerzenia funkcyjnego.

Sieć zawiera tylko jeden neuron liniowy w warstwie wyjsciowej.

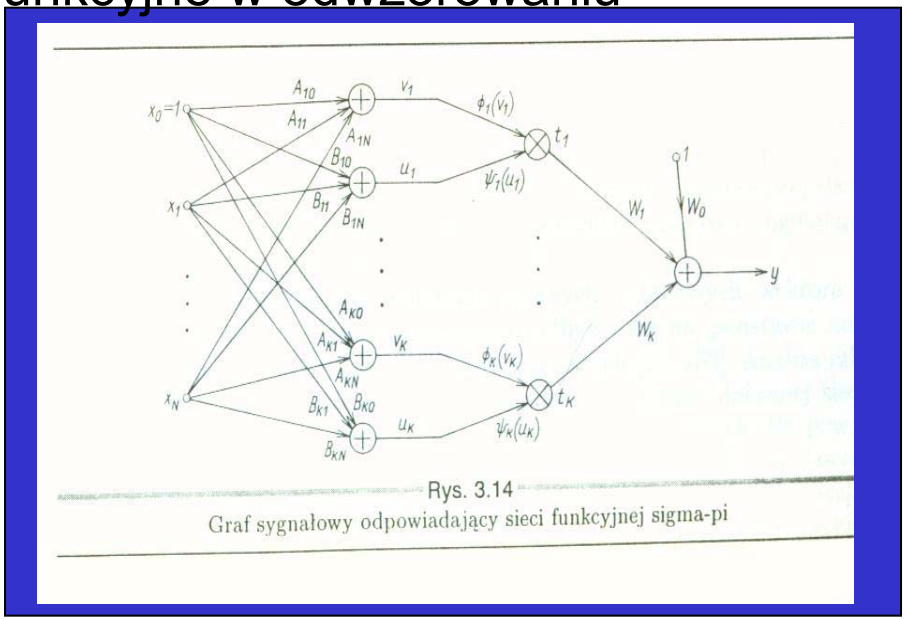
Wejście sieci stanowi rozszerzony wektor 19-elementowy o skladnikach:

x_1, $\sin(\pi x_1)$, $\cos(\pi x_1)$, $\sin(2\pi x_1)$, $\cos(2\pi x_1)$, $\sin(3\pi x_1)$, $\cos(3\pi x_1)$, x_2, $\sin(\pi x_2)$, $\cos(\pi x_2)$, $\sin(2\pi x_2)$, $\cos(2\pi x_2)$, $\sin(3\pi x_2)$, $\cos(3\pi x_2)$, $x_1 x_2$, $x_1 \sin(\pi x_2)$, $x_1 \cos(\pi x_2)$, ...

Sieć sigma-pi

Odmianą sieci neuronowej z rozszerzeniem funkcyjnym jest sieć sigma-pi, wykorzystującą składniki funkcyjne w odwzorowaniu danych wejściowych.

Sieć zawiera 2 warstwy neuronów: warstwę ukrytą funkcyjną oraz warstwę wyjściową liniową. Każdy neuron warstwy ukrytej zawiera dwa sumatory, dwie funkcje aktywacji oraz jeden mnożnik.



Rys. 3.14
Graf sygnałowy odpowiadający sieci funkcyjnej sigma-pi

Wprowadzenie połączeń synaptycznych wyższego rzędu do struktury sieci umożliwia uproszczenie i przyspieszenie procesu uczenia z jednoczesnym zwiększeniem zdolności klasyfikacyjnych i aproksymacyjnych sieci.

Analiza wrażliwościowa danych uczących

Ograniczenie liczby powiązań neuronowych i neuronów sprowadza się w klasycznym podejściu do redukcji wag warstwy ukrytej i wyjściowej, bez uwzględnienia warstwy wejściowej. Tymczasem każda składowa wektora wejściowego x zawiera różną dawkę informacji, w zróżnicowany sposób wpływającą na wartość sygnałów wejściowych sieci.
Podjęcie decyzji co do eliminacji pewnych składowych wektora x i zmniejszenia liczby danych wejściowych odbywa się na podstawie analizy wrażliwościowej sieci względem danych uczących.
Przez wrażliwość rozumie się

$$S_{ki}^{(p)} = \partial y_k / \partial x_i$$

W oparciu o wyznaczoną wrażliwość można podjąć decyzję o eliminacji składnika(ków) wektora x.

Dobór próbek uczących sieci

Ze względu na działanie układu sieć neuronowa może być rozpatrywana jako klasyfikator wektorowy, wyznaczający przynależność danego wektora wejściowego x do określonej grupy. Każda warstwa neuronów pełni przy tym inną funkcję w sieci.

→ neurony należące do pierwszej warstwy ukrytej kształtują hiperpłaszczyzny separujące N-wymiarowa przestrzeń danych (N-liczba wejść sieci) na regiony zawierające dane należące do tej samej grupy.

→neurony warstwy wyjściowej (lub drugiej warstwy ukrytej) reprezentują zbiór danych tworzących pewną grupę. Położenie tych danych względem hiperpłaszczyzn jest bardzo ważne.

→ Istnieją pewne reguły dotyczące możliwości reprezentacji hiperpłaszczyzn i regionów przez liczbę neuronów których uwzględnianie jest istotne przy doborze liczby próbek uczących.

Wtrącanie szumu do danych uczących

→ Metody przestawione poprzednio realizowały zwiększenie zdolności uogólniania poprzez oddziaływanie na samą architekturę sieci. Jest to podstawowa metoda umożliwiająca uzyskanie dobrych właściwości uogólniających sieci.

→ Przy ustalonej minimalnej architekturze sieci jest możliwa dalsza poprawa poprzez odpowiednie przygotowanie zbioru danych uczących. *Przy dobrze wytrenowanej sieci* podstawowa jest zasada mówiąca że sygnały wyjściowe sieci powinny być niewrażliwe na zmianę wielkości wejściowych dopóty, dopóki zmiany te są zawarte w pewnych dopuszczalnych granicach przy założeniu, że sieć realizuje odwzorowanie gładkie. Podobne sygnały powinny generować podobne odpowiedzi nawet jeżeli nie wchodziły w skład wzorców uczących. *Uzasadnienie matematyczne* tego twierdzenia przebiega następująco.

Wtrącanie szumu do danych uczących

Rozważamy sieć dwuwarstwową zawierającą warstwę ukrytą o K neuronach i warstwę wyjściową o M neuronach. Liczbę wejść sieci przyjmuje się jako N. Wektor wag sieci oznacza się przez W.

$$y = f(W,x)$$

Wektor wejściowy uczący zaznaczamy x_u, testujący x_t.
Rozwiązanie problemu uczenia definiujemy jako minimalizację funkcji celu:

$$E = \tfrac{1}{2} \Sigma \, || \, d^{(k)} - f(x_t^{(k)}) \, ||$$

prowadzi to do wartości wag optymalnych z punktu widzenia wzorców uczących.
Minimalizacja tej funkcji niekoniecznie zapewnia właściwą odpowiedź sieci na wymuszenie w postaci wektora x_t nie będącego składnikiem danych uczących.

Wtrącanie szumu do danych uczących

Dla zbadania wrażliwości sieci na niewielką zmianę wektora uczącego x_t założono, że testowany wektor x_t różni się niewiele od wektora uczącego x_u. Przyjęto że:

$$x_t^{(k)} = x_u^{(k)} + s,$$

przy czym

$$s=[s_1, s_2, ..., s_N]^T$$

oznacza *wektor szumu*, składający się ze zmiennych losowych o małej amplitudzie.
Wektor testowy można traktować jako wektor zaszumiony, który wywołuje perturbacje sygnału wyjściowego $y^{(k)}$, zapisaną w postaci

$$\Delta y^{(k)} = f(x_u^{(k)} + s) – f(x_u^{(k)}) \approx \partial f(x_u^{(k)})/ \partial x \; s$$

przy czym $\partial f(x_u^{(k)})/\partial x$ oznacza jakobian funkcji wektorowej $f(x)$.

Wtrącanie szumu do danych uczących

Wrażliwość względna sieci jest definiowana jako

$$R(W) = (\Sigma < || \Delta y^{(k)} ||^2 >) / <||s||^2>$$

gdzie wektor szumu s ma wartość średnią $<s> = 0$ i wariancje $<ss^T> = \sigma$

Przy powyższych założeniach dotyczących szumu, można przekształcić do postaci

$$R(W) = \Sigma\ 1/N\ || \partial f(x_u^{(k)})/\partial x ||^2$$

Im mniejsza wrażliwość sieci, tym sieć mniej czuła na “zakłócenia” wektora wejściowego w stosunku do wektora uczącego, czyli lepsza zdolność uogólniająca.

Wtrącanie szumu do danych uczących

Uwzględnianie czynnika wrażliwościowego w uczeniu sieci jest możliwe przez modyfikację definicji funkcji celu. Oznacza się ją w postaci odpowiedniej sumy ważonej

$$L(W) = E(W) + \alpha\, R(W)$$

przy czym α jest współczynnikiem wagowym, otrzymuje się

$$L(W) = \Sigma \{ || d^{(k)} - f(W, x_u^{(k)} ||2 + \alpha/N || \partial f(x_u^{(k)})/\partial x ||^2 \}$$

Jeżeli przyjąć ze α/N oznacza wariancje pewnego szumu tworzącego wektor $n = [n_1, n_2, ..., n_N]^T$ o zerowej wartości oczekiwanej $<n>=0$ oraz $<nn^T> = \varepsilon = 1$. Można wówczas funkcje celu $L(W)$ przekształcić w postać

$$L(W) = \Sigma \{ || d^{(k)} - f(W, x_u^{(k)}) ||^2 + \varepsilon || \partial f(x_u^{(k)})/\partial x ||^2 \}$$
$$\approx < \Sigma || d^{(k)} - f(x_u^{(k)} + n) ||^2 >$$

równanie definiujące zmodyfikowaną funkcję celu ma postać identyczną ze standartową postacią z ta różnicą, że zamiast wektora wejściowego x_u używa się zaszumionego wektora x_t.

Wtrącanie szumu do danych uczących

Taką funkcję celu można interpretować jako *proces uczenia na wzorcach zaszumionych*, co przy ustalonej architekturze sieci powinno prowadzić do lepszych własności uogólniających.

Oddzielny problem to *dobór wariancji szumu*, umożliwiający rzeczywistą poprawę zdolności uogólniania sieci. Jest to zagadnienie teoretycznie trudne do rozwiązania. Natomiast dość proste empirycznie. Empirycznie stwierdzono, że ta wariancja powinna być skorelowana z prawidłowym rozkładem różnicy między danymi uczącymi (niezaszumionymi) a danymi testującymi, stanowiąc niewielki procent tej różnicy.

Zestaw pytań do testu

1. Czym charakteryzuje się metoda wrażliwościowa redukcji sieci.
2. Na czym polega sieć Pao.
3. Co to jest rozszerzenie funkcyjne.
4. Co to jest sieć sigma-pi.
5. Jaki jest cel wtrącania szumu do danych uczących.