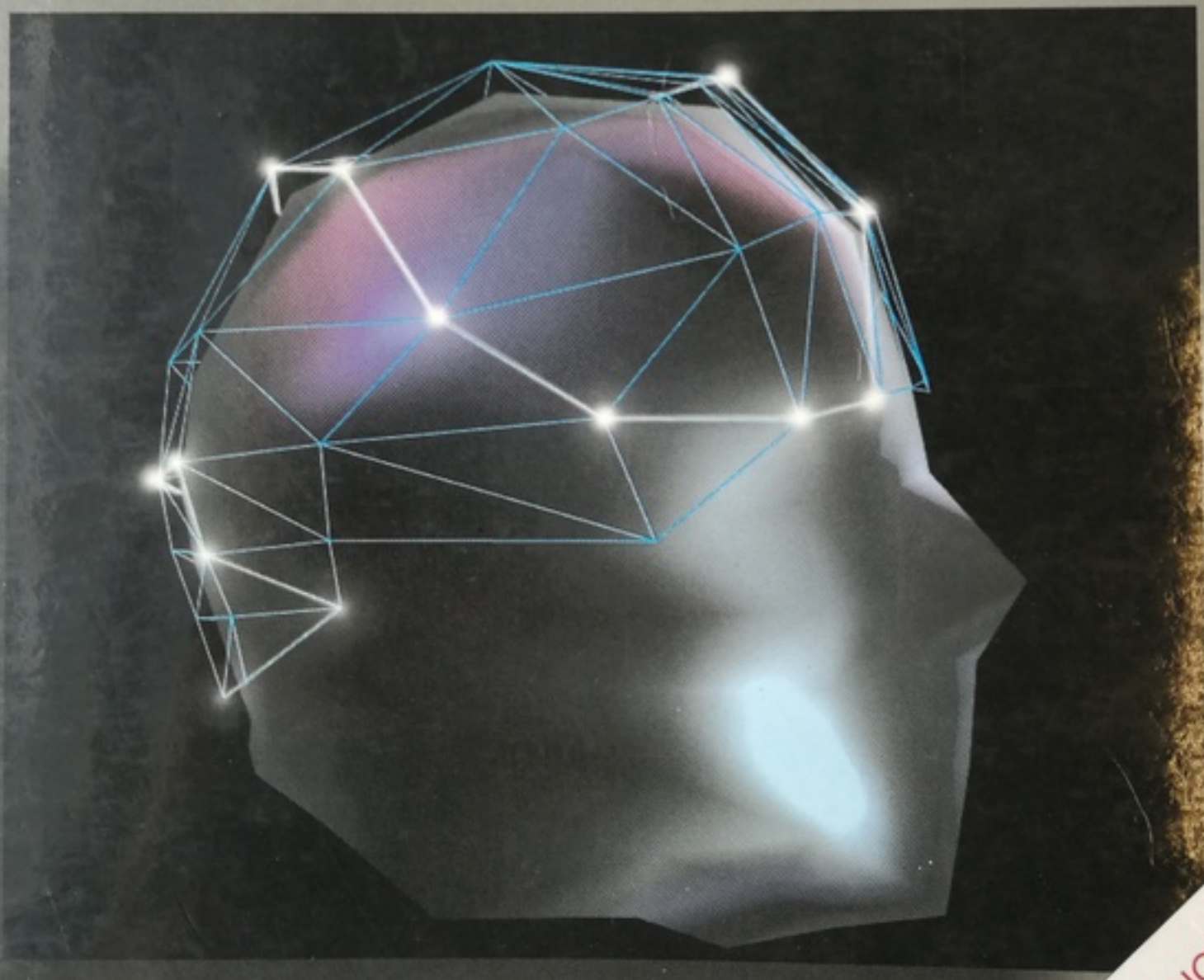
WSTĘP DO TEORII OBLICZEŃ NEURONOWYCH

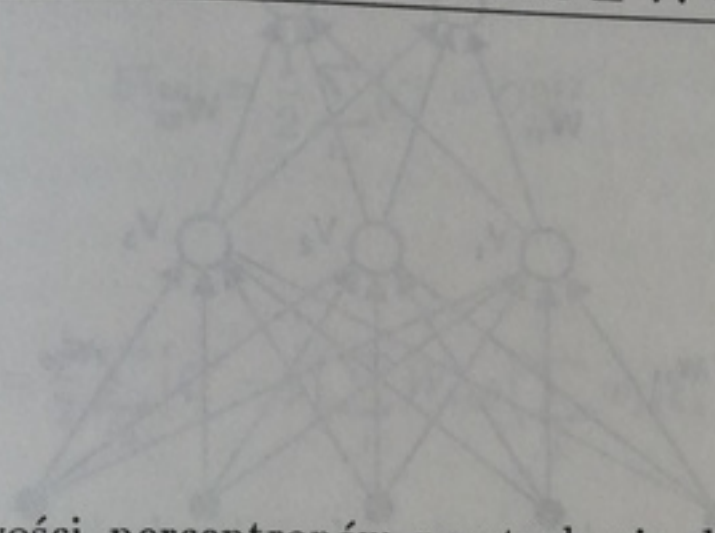
John Hertz
Anders Krogh
Richard G. Palmer



Wydawnictwa
Naukowo-Techniczne

Wydanie drugie

6. SIECI WIELOWARSTWOWE



Ograniczenia możliwości perceptronów prostych nie dotyczą sieci jednokierunkowych z warstwami pośrednimi, czyli „ukrytymi" między warstwą wejściową a wyjściową. Rzeczywiście, jak zobaczymy później, sieć z jedną warstwą ukrytą może reprezentować dowolną funkcję boolowską (łącznie, np. z funkcją XOR). Chociaż z większych możliwości sieci wielowarstwowych zdawano sobie sprawę już od dawna, dopiero ostatnio pokazano, jak należy uczyć je konkretnych funkcji za pomocą „propagacji wstecznej" lub innych metod. Brak reguły uczenia wraz z argumentacją Minsky'ego i Paperta [1969], że tylko funkcje liniowo separowalne mogły być realizowane przez perceptrony proste, doprowadziły do zaniku zainteresowania sieciami warstwowymi na wiele lat.

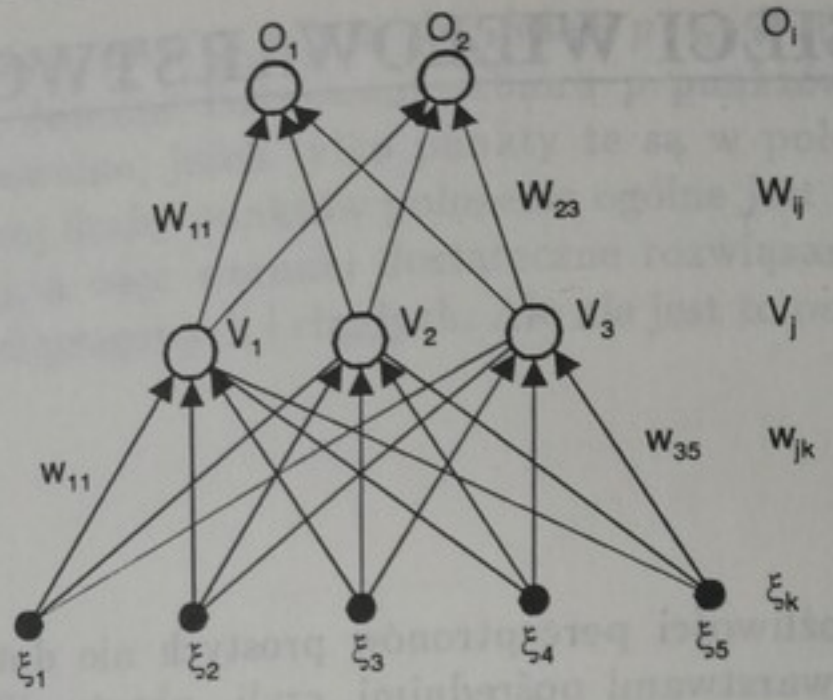
W tym rozdziale, jak w poprzednim, rozważamy tylko sieci *jednokierunkowe*. Bardziej ogólne sieci są omówione w następnym rozdziale.

6.1. PROPAGACJA WSTECZNA BŁĘDU

Algorytm propagacji wstecznej błędu jest podstawą większości bieżących prac na temat uczenia sieci neuronowych. Został on odkryty niezależnie, kilka razy, przez Brysona i Ho [1969], Werbosa [1974], Parkera [1985] i Rumelharta, Hintona i Williamsa [1986a, b]. Ściśle pokrewny sposób podał Le Cun [1985]. Algorytm podaje przepis na zmianę wag w_{pq} w dowolnej sieci jednokierunkowej, która ma nauczyć się zbioru treningowego par wejścia-wyjścia $\{\xi_k^\mu, \zeta_i^\mu\}$. Podstawą jest po prostu spadek gradientu, jak opisano w p. 5.4 (liniowy) i p. 5.5 (nieliniowy) dla perceptronu prostego.

Najpierw rozważymy sieć dwuwymiarową (rys. 6.1). Nasza konwencja zapisu jest pokazana na rysunku; jednostki wyjściowe są oznakowane jako O_i, jednostki ukryte jako V_j, a końcówki wejściowe jako ξ_k. Istnieją połączenia w_{jk} od wejść do jednostek ukrytych i W_{ij} od jednostek ukrytych do wyjściowych. Zwracamy uwagę, że indeks i zawsze dotyczy jednostki wyjściowej, j — ukrytej, a k — końcówki wejściowej.

Sygnały wejściowe są zawsze stałe i równe określonym wartościom. Jak w rozdziale poprzednim, różne wzorce oznaczono indeksem μ, a więc wejście



Rys. 6.1. Dwuwarstwowa sieć jednokierunkowa; pokazano oznaczenia jednostek i wag

k-te jest równe ξ_k^μ, gdy jest prezentowany wzorzec μ-ty. Wielkości ξ_k^μ mogą być dwuwartościowe (0/1 lub ± 1) albo ciągłe. Stosujemy N na oznaczenie liczby jednostek wejściowych i p, jak poprzednio, na oznaczenie liczby wzorców wejściowych ($\mu = 1, 2, \ldots, p$).

Dla danego μ-tego wzorca j-ta jednostka ukryta otrzymuje sygnał pobudzenia sieciowego

$$h_j^\mu = \sum_k w_{jk} \xi_k^\mu \tag{6.1}$$

i wytwarza sygnał wyjściowy

$$V_j^\mu = g(h_j^\mu) = g\Big(\sum_k w_{jk} \xi_k^\mu\Big) \tag{6.2}$$

Jednostka wyjściowa i-ta otrzymuje zatem sygnał

$$h_i^\mu = \sum_j W_{ij} V_j^\mu = \sum_j W_{ij} g\Big(\sum_k w_{jk} \xi_k^\mu\Big) \tag{6.3}$$

i ostatecznie wytwarza wyjściowy

$$O_i^\mu = g(h_i^\mu) = g\Big(\sum_j W_{ij} V_j^\mu\Big) = g\Big(\sum_j W_{ij} g\Big(\sum_k w_{jk} \xi_k^\mu\Big)\Big) \tag{6.4}$$

Podobnie jak w poprzednim rozdziale, pomijamy progi; można je uwzględnić wprowadzając, jak zwykle, dodatkową jednostkę wejściową o stałym sygnale wejściowym — 1 i połączoną ze wszystkimi jednostkami sieci.

Nasza zwykła miara błędu, czyli funkcja kosztu

$$E[w] = \frac{1}{2}\sum_{i\mu}[\zeta_i^\mu - O_i^\mu]^2 \tag{6.5}$$

ma teraz postać

$$E[w] = \frac{1}{2}\sum_{i\mu}\Big[\zeta_i^\mu - g\Big(\sum_j W_{ij} g\Big(\sum_k w_{jk}\xi_k^\mu\Big)\Big)\Big]^2 \tag{6.6}$$

Jest ona oczywiście ciągłą i różniczkowalną funkcją wszystkich wag, możemy więc zastosować algorytm spadku gradientu do uczenia właściwych wag. W pewnym sensie to wszystko, o co chodzi w propagacji wstecznej, ale postać wynikowa reguł modyfikacji wag ma wielkie znaczenie praktyczne.

Dla połączeń między jednostkami ukrytymi a wyjściowymi reguła spadku gradientu daje

$$\Delta W_{ij} = -\eta\frac{\partial E}{\partial W_{ij}} = \eta\sum_\mu[\zeta_i^\mu - O_i^\mu]g'(h_i^\mu)V_j^\mu =$$

$$= \eta\sum_\mu \delta_i^\mu V_j^\mu \tag{6.7}$$

gdzie określiliśmy

$$\delta_i^\mu = g'(h_i^\mu)[\zeta_i^\mu - O_i^\mu] \tag{6.8}$$

Wynik ten jest oczywiście identyczny z otrzymanym wcześniej (równania (5.50) i (5.51)) dla perceptronu jednowarstwowego, gdzie na wyjściu jednostki ukrytej pojawia się V_j^μ, odgrywające teraz rolę wejścia perceptronu.

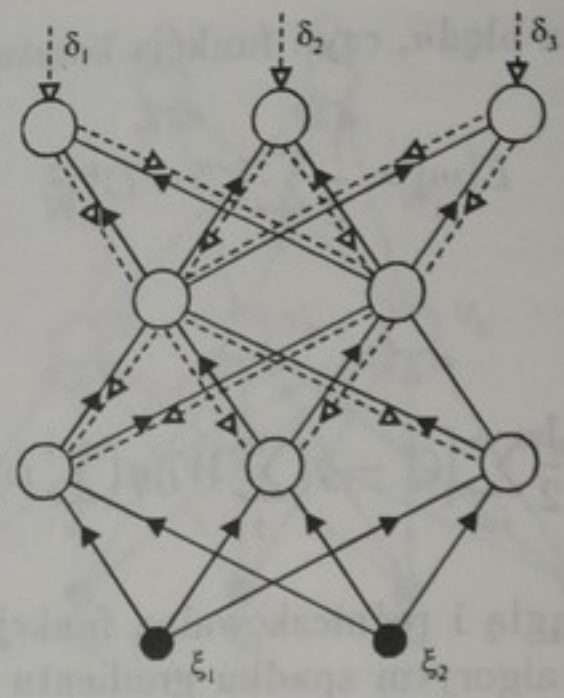
Aby obliczyć zmianę Δw_{jk} połączeń między jednostkami wejściowymi a ukrytymi, musimy wykonać różniczkowanie względem wag w_{jk}, w nawiasach we wzorze (6.6). Stosując regułę łańcuchową otrzymujemy

$$\Delta w_{jk} = -\eta\frac{\partial E}{\partial w_{jk}} = -\eta\sum_\mu \frac{\partial E}{\partial V_j^\mu}\frac{\partial V_j^\mu}{\partial w_{jk}} =$$

$$= \eta\sum_{\mu i}[\zeta_i^\mu - O_i^\mu]g'(h_i^\mu)W_{ij}g'(h_j^\mu)\xi_k^\mu =$$

$$= \eta\sum_{\mu i}\delta_i^\mu W_{ij}g'(h_j^\mu)\xi_k^\mu =$$

$$= \eta\sum_\mu \delta_j^\mu \xi_k^\mu \tag{6.9}$$



Rys. 6.2. Propagacja wsteczna w sieci trójwarstwowej. Linie ciągłe oznaczają propagację sygnału do przodu, a linie przerywane — propagację wsteczną błędów (δ)

przy czym

$$\delta_j^\mu = g'(h_j^\mu) \sum_i W_{ij} \delta_i^\mu \tag{6.10}$$

Zauważmy, że wzór (6.9) ma taką samą postać jak (6.7), ale przy innej definicji δ. W zasadzie dla dowolnej liczby warstw reguła modyfikacji metodą propagacji wstecznej ma zawsze postać

$$\Delta w_{pq} = \eta \sum_{wzorce} \delta_{wyjście} \times V_{wejście} \tag{6.11}$$

przy czym *wyjście* i *wejście* odnosi się do dwóch końców p i q omówionego połączenia, a V oznacza odpowiednią aktywację końca wejściowego pochodzącą od jednostki ukrytej lub rzeczywistego wejścia. Znaczenie δ zależy od rozpatrywanej warstwy; dla ostatniej warstwy połączeń δ jest określone wzorem (6.8), a dla wszystkich pozostałych warstw jest dane przez równanie typu (6.10). Wyprowadzenie uogólnionego wyniku (6.11) dla wielu warstw można łatwo przeprowadzić stosując po prostu kolejno regułę łańcuchową.

Równanie (6.10) umożliwia nam wyznaczenie błędu δ dla danej jednostki ukrytej V_j jako funkcji błędów jednostek O_i, które są przez nią pobudzone. Współczynnikami są zwyczajne wagi „do przodu" W_{ij}, ale w tym miejscu przenoszą one błędy (δ) wstecz zamiast sygnałów do przodu: stąd nazwa **propagacja wsteczna błędu** lub po prostu **propagacja wsteczna**. Możemy wobec tego skorzystać z tej samej sieci — czy raczej jej dwukierunkowej wersji — do obliczania zarówno wartości wyjściowych, jak i błędów (δ). Ideę tę pokazano na rys. 6.2 dla sieci trójwymiarowej.

Chociaż napisaliśmy reguły modyfikacji (6.7) i (6.9) w postaci sum po wzorcach μ, są one zazwyczaj stosowane przyrostowo: wzorzec μ-ty jest

podawany na wejście, a następnie wszystkie wagi są modyfikowane, zanim zostanie uwzględniony następny wzorzec. Jest oczywiste, że dzięki temu funkcja kosztów (dla dostatecznie małego η) zmniejsza się w każdym kroku, a kolejne kroki mogą być wykonane w kierunku lokalnego gradientu. Jeżeli wzorce są podawane w losowej kolejności, to ścieżka w przestrzeni wag jest stochastyczna, dzięki czemu można lepiej wykorzystać powierzchnię funkcji kosztu. Alternatywny sposób grupowy — przyjąć dosłownie (6.7) i (6.9) i modyfikować po zakończeniu podawania wszystkich wzorców — wymaga dodatkowej pamięci lokalnej dla każdego połączenia. Efektywność względna obu sposobów zależy od danego problemu, ale podejście przyrostowe wydaje się mieć przewagę w większości przypadków, zwłaszcza gdy zbiór trenujący jest bardzo regularny lub nadmiarowy.

Fakt, że odpowiednie pochodne funkcji kosztu mogą być obliczane metodą propagacji wstecznej błędu jest oczywiście atrakcyjny. Ale ma to dwie istotne konsekwencje:

- Reguła modyfikacji (6.11) jest *lokalna*. Aby obliczyć zmianę wagi danego połączenia trzeba tylko wielkości dostępnych (po propagacji wstecznej błędów) na obu końcach tego połączenia. Dzięki temu reguła propagacji wstecznej nadaje się do obliczeń równoległych. Może ona też mieć pośredni związek z neurobiologią.[30)]
- Złożoność obliczeniowa jest mniejsza, niż moglibyśmy oczekiwać. Jeżeli liczba wszystkich połączeń jest równa n, to obliczenie funkcji kosztu (6.6) wymaga wykonania liczby operacji rzędu n, a obliczenia bezpośrednio n pochodnych wymagałoby liczby operacji rzędu n^2. Schemat propagacji wstecznej umożliwia zaś obliczenie wszystkich pochodnych wykonując liczbę operacji rzędu n.

Zastosowanie funkcji sigmoidalnej jako funkcji aktywacji $g(h)$ jest naturalne. Funkcja ta musi być oczywiście różniczkowalna i na ogół chcemy, żeby miała nasycenie z obu stron. Oba zakresy zmian 0/1 i ± 1 mogą być wykorzystane, przy czym funkcje aktywacji mają odpowiednio postać

$$g(h) = f_\beta(h) = \frac{1}{1 + \exp(-2\beta h)} \tag{6.12}$$

oraz

$$g(h) = \operatorname{tgh} \beta h \tag{6.13}$$

30) Lokalność jest konieczna z punktu widzenia biologii, ale nie jest dostateczna. Dwukierunkowe, dwufunkcyjne połączenia nie mają sensu biologicznego [Grossberg, 1987b], ale można ich uniknąć przez hipotetyczne realizacje neurobiologiczne [Hecht-Nielsen, 1989]. Jednak propagacja wsteczna wydaje się być odmienna od biologicznych mechanizmów uczenia [Crick, 1989].

Parametr stromości β często ma wartość 1 lub 1/2 w równaniu (6.12). Jak zauważyliśmy w rozdz. 5, pochodne tych funkcji można wyrazić bezpośrednio przez te funkcje, jako $g'(h) = 2\beta g(1-g)$ dla (6.12) i $g'(h) = \beta(1-g^2)$ dla (6.13). Często zatem spotyka się wzór (6.8) zapisany na przykład jako

$$\delta_i^\mu = O_i^\mu(1-O_i^\mu)(\zeta_i^\mu - O_i^\mu) \tag{6.14}$$

dla jednostek 0/1, gdy $\beta = 1/2$.

Ponieważ propagacja wsteczna jest tak bardzo ważna, podsumujemy wyniki opisując procedurę krok po kroku, biorąc jeden μ-ty wzorzec na raz (tj. modyfikację przyrostową). Omawiamy sieć M-warstwową, $m = 1,2,\ldots,M$ i używamy V_i^m jako wyjścia jednostki i-tej w warstwie m-tej. V_i^0 będzie synonimem ξ_i, wejścia i-tego. Zwracamy uwagę, że górny indeks oznacza m-tą warstwę, a nie wzorzec. Wielkość w_{ij}^m jest połączeniem od V_j^{m-1} do V_i^m. Procedura propagacji wstecznej jest zatem następująca:

1. Początkowe wartości wag wybierz jako małe liczby losowe.
2. Wybierz wzorzec ξ_k^μ i podaj go na wejście (warstwa $m = 0$) tak, aby

$$V_k^0 = \xi_k^\mu \quad \text{dla każdego } k \tag{6.15}$$

3. Przepuść sygnał przez sieć do przodu za pomocą

$$V_i^m = g(h_i^m) = g\left(\sum_j w_{ij}^m V_j^{m-1}\right) \tag{6.16}$$

dla każdego i oraz m, dopóki nie obliczysz wszystkich wyjść V_i^M.

4. Oblicz błędy δ w warstwie wyjściowej

$$\delta_i^M = g'(h_i^M)[\zeta_i^\mu - V_i^M] \tag{6.17}$$

porównując wyjścia otrzymane V_i^M z pożądanym ζ_i^μ dla wzorca μ-tego, który został podany.

5. Oblicz błędy δ w poprzednich warstwach za pomocą propagacji wstecznej błędu

$$\delta_i^{m-1} = g'(h_i^{m-1}) \sum_j w_{ji}^m \delta_j^m \tag{6.18}$$

dla $m = M, M-1, \ldots, 2$, dopóki nie obliczysz błędów dla wszystkich jednostek.

6. Zastosuj

$$\Delta w_{ij}^m = \eta \delta_i^m V_j^{m-1} \tag{6.19}$$

do modyfikacji wszystkich połączeń według zależności $w_{ij}^{nowe} = w_{ij}^{stare} + \Delta w_{ij}$.

7. Wróć do punktu 2 i powtórz procedurę dla następnego wzorca.

Można w prosty sposób uogólnić propagację wsteczną na inne rodzaje sieci, w których połączenia przeskakują jedną lub kilka warstw, na przykład bezpośrednie połączenia wejście-wyjście, (rys. 6.5b). Otrzymamy taki sam rodzaj schematu propagacji wstecznej, jeżeli sieć jest *jednokierunkowa*, bez połączeń wstecznych ani bocznych (lateralnych).

6.2. WARIACJE NA TEMAT PROPAGACJI WSTECZNEJ

Propagacja wsteczna była badana w czasie ostatnich kilku lat i rozpatrywano jej liczne modyfikacje i rozszerzenia. Algorytm podstawowy wyżej opisany jest wolnozbieżny w sieci wielowarstwowej i liczne wariacje proponują, jak je przyspieszyć. Inne cele modyfikacji dotyczą unikania minimów lokalnych i poprawy zdolności generalizacji. Obecnie omawiamy przede wszystkim problem szybkości metody.

Warto wspomnieć, że porównania szybkości osiąganych za pomocą różnych technik nie zawsze są jasne. Różni autorzy badają różne problemy przy różnych kryteriach stopu, różnych miarach prędkości obliczeń i różnym podejściu do uśredniania. Jako jeden przykład trudności omówmy problem, co robić, gdy próba nie kończy się pomyślnie. Na ogół problem rozwiązujemy wiele razy (np. poczynając od różnych losowych zbiorów wag i posługując się losowymi sekwencjami modyfikacji) i niekiedy algorytm utknie w minimum lokalnym lub na bardzo płaskiej wyżynie. Po pewnym maksymalnym czasie T eksperymentator musi się poddać, ale jak to uwzględnić w obliczeniu średniego czasu $\langle t \rangle$ na jedną symulację? Niektóre stosowane rozwiązania są następujące:

- Policz ten czas jako próbę udaną z czasem T.
- Odrzuć ten czas, uwzględniając osobno próby nieudane.
- Nie zaliczaj tej próby do udanych, ale dodaj czas T do czasu następnej próby, toteż uśrednione czasy są to czasy całkowite między udanymi wynikami.
- Uśrednij $1/t$ zamiast t, przyjmując $1/t = 0$ dla prób nieudanych.

Fahlman [1989] szeroko omawia te i inne aspekty.

Istnieje wiele parametrów, które można rozpatrywać w zakresie usprawnienia ogólnej metody propagacji wstecznej, włącznie z architekturą (liczbą warstw, liczbą jednostek w warstwie), wielkością i naturą zbioru treningowego i regułą modyfikacji wag. W tym podrozdziale koncentrujemy się głównie na regule modyfikacji, utrzymując architekturę bez zmian. Inne aspekty i inne architektury są omówione później.