Metody treningu ANN

dr hab. inż. Jerzy Balicki, prof. nadzw.

Plan

1. Perceptron
2. Perceptron wielowarstwowy
3. Algorytm wstecznej propagacji błędów
4. Metody minimalizacji funkcji błędu

Perceptron 1958

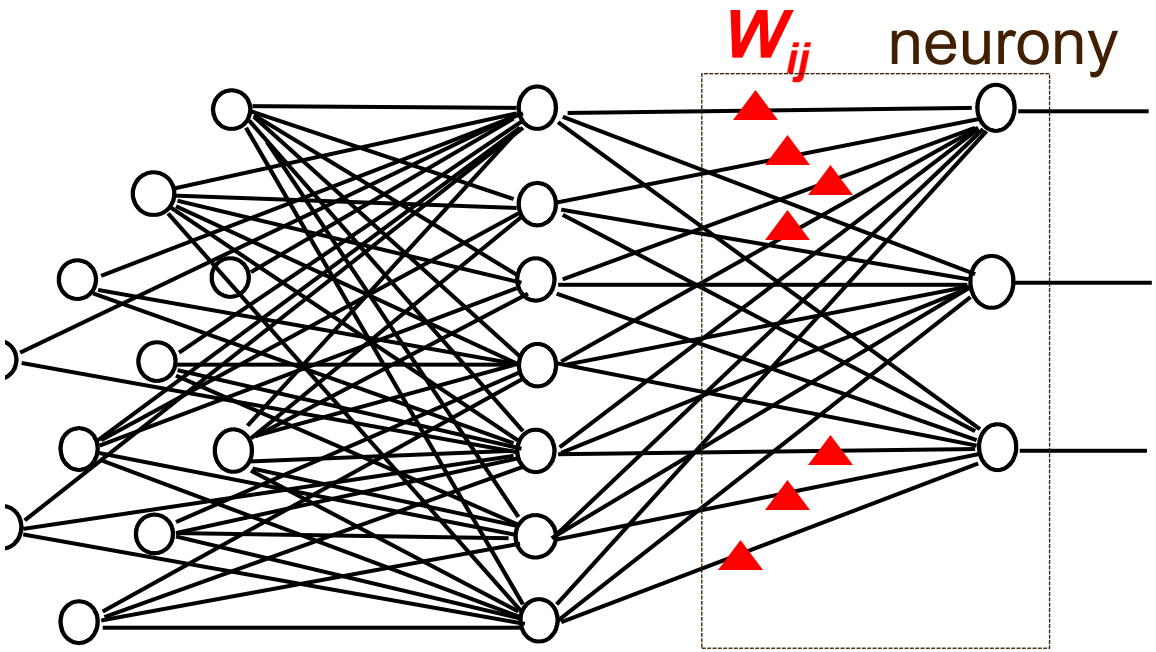
- **Frank Rosenblatt (Cornell Univ.) 1960, klasyfikator neuronowy Mark I wzorowany na biologicznej percepcji**

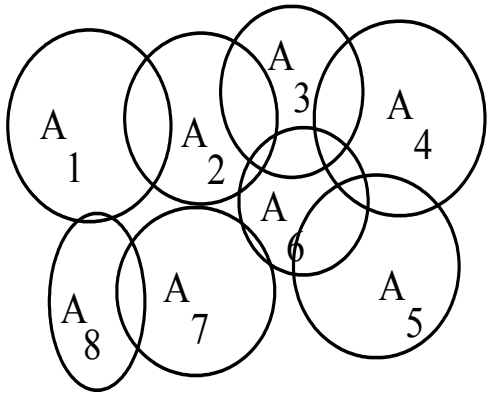
- **Trzy warstwy, elementy:**
  - **wejściowe (S), np. fotokomórki 12 x 20**
  - **asocjacyjne (A), zbierające dane z większych obszarów, 512**
  - **wyjściowe (R), 8**

- **Identyfikacja figur, znaków, eksperymenty psychologiczne, szybkość uczenia, błędy.**
- **Jakich klasyfikacji dokonać może perceptron?**
- **Jak można go uczyć?**

Perceptron jednowarstwowy

$S^m = \pm 1;$ $Y^m = \pm 1;$





Działanie perceptronu

S_j = –1 lub +1

Połączenia C_{ij} = 0, ±1 od S_j do A_i

$$A_i = g\left(\sum_j C_{ij} S_j - \theta_i\right)$$

$g(\ )$ – funkcja bipolarna

A_i = +1 powyżej progu, A_i = –1 poniżej.

Działanie perceptronu

Pary treningowe (S^m, Y^m), $Y^m = \pm 1$;

Sygnał wyjściowy R^m :

$$\begin{cases} R^{\mu} = +1 \text{ dla } I^{\mu} = \sum_j W_{ij} A_j{}^{\mu} > N\kappa \\ R^{\mu} = -1 \text{ dla } I^{\mu} = \sum_j W_{ij} A_j{}^{\mu} < -N\kappa \\ R^{\mu} = 0 \text{ w pozostałych przypadkach} \end{cases}$$

dla $\kappa > 0$ (κ próg wyjściowy)

Wyznaczanie wag – trening z nauczycielem

Empiryczny błąd systemu (ryzyko empiryczne, błąd uczenia):

$$E_{emp}(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - d_i)^2 \underset{\mathbf{w}}{=\min}\sum_{i=1}^{n}\left[f(\mathbf{x}_i, \mathbf{w}) - d_i\right]^2$$

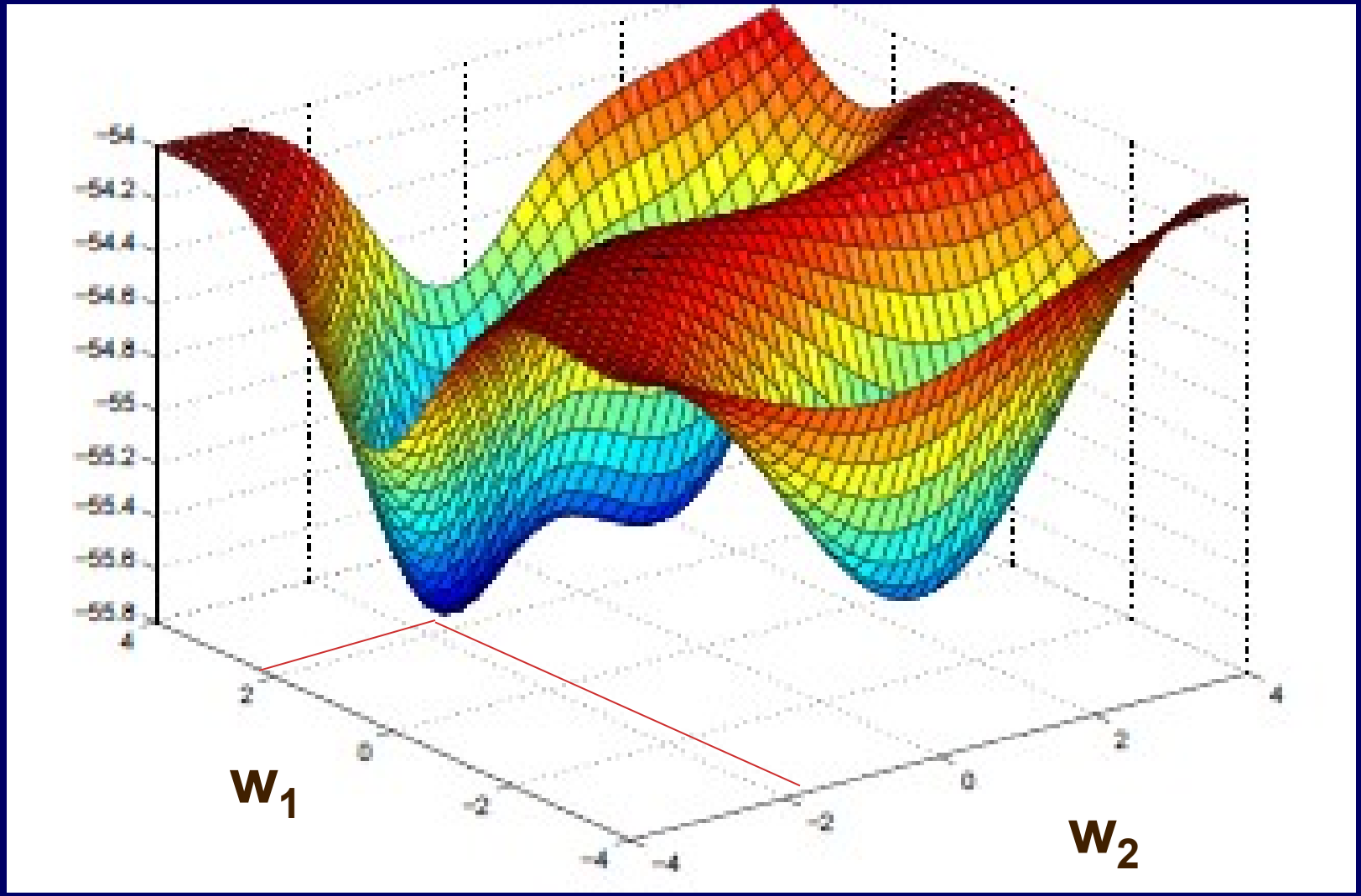
Zastosowano *n* par uczących (x_i, d_i)

Dla wejścia do sieci x_i powinna być wyliczona przez sieć odpowiedź d_i, ale jest y_i .

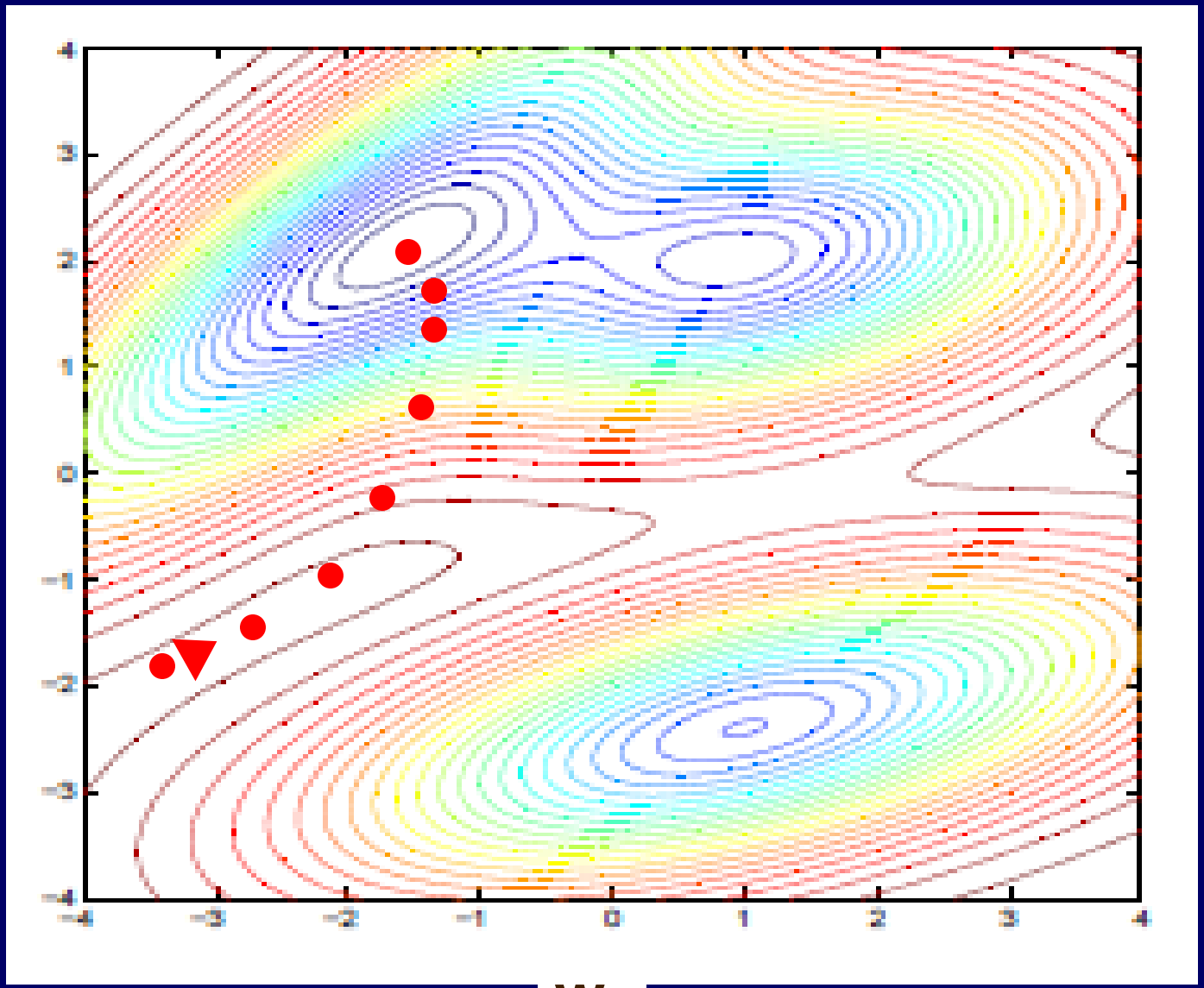
x_i , d_i , y_i są wektorami

Aby odpowiedzi systemu *y* były prawidłowe (równe *d*), minimalizuje się wartość funkcji *E(w)*, gdzie zmiennymi decyzyjnymi są *x*

Wyznaczanie wag *w** – minimalizacja *E(w)*



Wyznaczanie wag *w** – minimalizacja *E(w)*



Metody gradientowe minimalizacji *E(w)*
Metoda najszybszego spadku

1. Losowy wybór punktu startowego *w*

2. Obliczenie gradientu funkcji *E*(*w*) w punkcie *w*

$$\nabla E(\mathbf{w}) = \begin{bmatrix} \frac{\partial E}{\partial w_1} \\ \frac{\partial E}{\partial w_2} \\ \vdots \\ \frac{\partial E}{\partial w_n} \end{bmatrix}$$

3. Wyznaczenie nowego punktu ***w'***

$$\mathbf{w}' = \mathbf{w} - \eta \nabla E(\mathbf{w})$$

η - arbitralnie wybrana stała (współczynnik korekcji)

4. Sprawdzenie warunku stopu (numer kroku lub wartość funkcji *E(w)*). Jeżeli warunek nie jest spełniony to skok do kroku 2.

Metody gradientowe minimalizacji *E(w)*

1. Do metod gradientowych należą także:

a) metoda propagacji wstecznej błędu uczenia ANN

b) metoda doboru funkcji przynależności systemu wnioskowania rozmytego

Gdy nie jest znany wzór funkcji, a funkcja dana jest przez wartości, to można zastosować przybliżenie:

$$\nabla E(\mathbf{w}) \approx \frac{1}{h}\begin{bmatrix} E(\mathbf{w}+h\mathbf{e}_1)-E(\mathbf{w}) \\ E(\mathbf{w}+h\mathbf{e}_2)-E(\mathbf{w}) \\ \vdots \\ E(\mathbf{w}+h\mathbf{e}_n)-E(\mathbf{w}) \end{bmatrix}$$

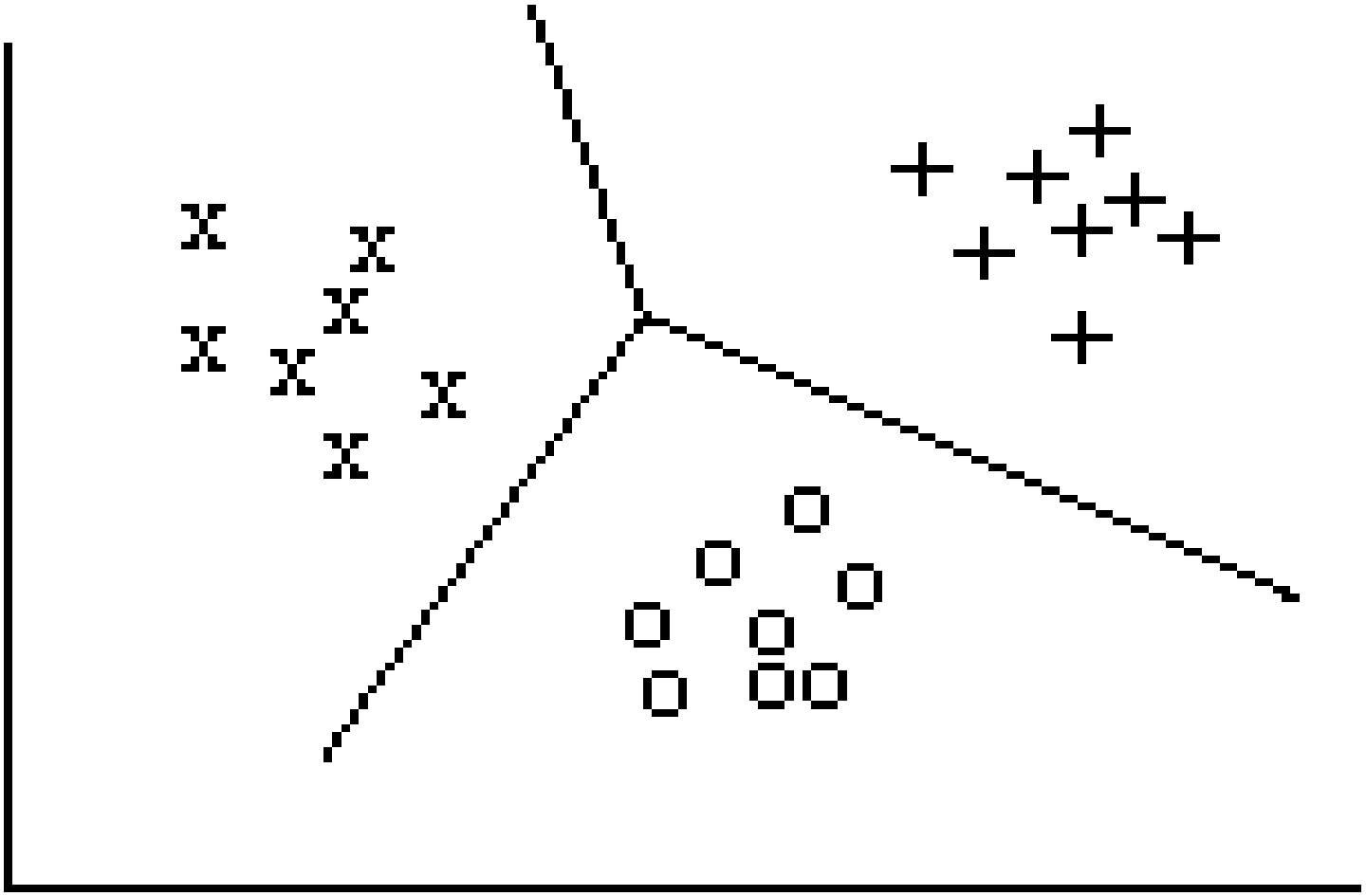
gdzie h jest arbitralnie wybraną niewielką liczbą,

$\mathbf{e}_i$, i = 1, 2, . . . , n oznacza wektor jednostkowy w kierunku i.

Metody gradientowe mogą zatrzymywać się w ekstremach lokalnych.

Perceptron jako klasyfikator dla *M* klas

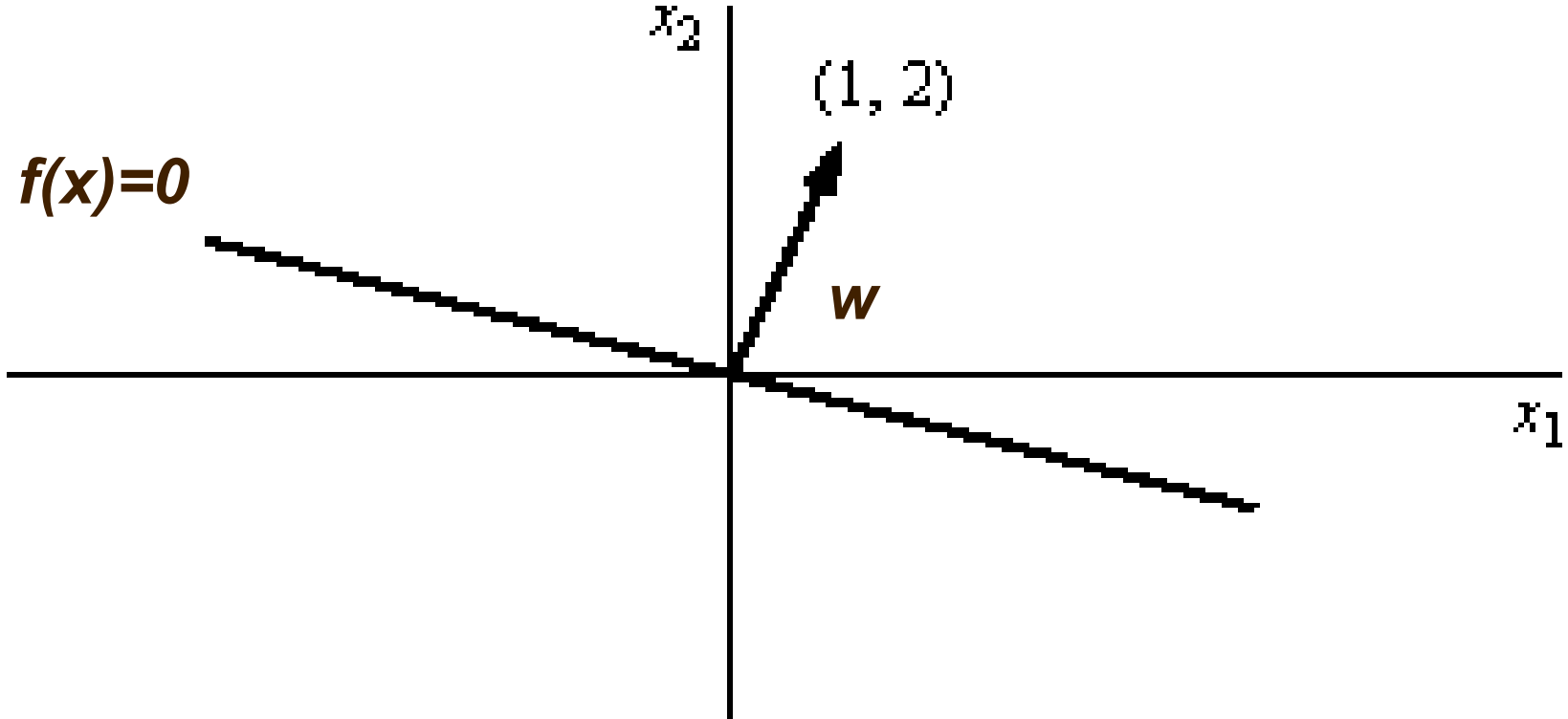
Granice decyzji perceptronu dla 3 klas.



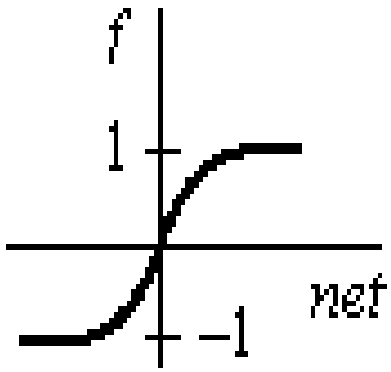
Hiperpowierzchnia decyzyjna neuronu

$$f(I) = f\left(\sum_i W_i X_i\right) = f(\mathbf{W}\cdot\mathbf{X}) = f(net)$$

I*(*X*), *net **– poziom aktywacji;** ***f*(*I*)** **– funkcja aktywacji**

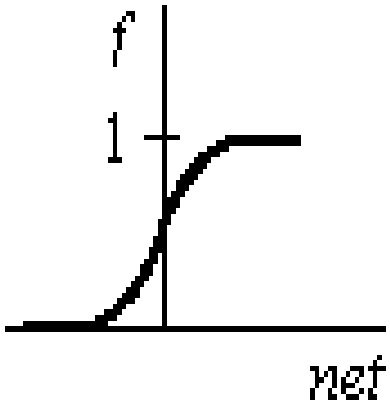


Sigmoidalne funkcje aktywacji



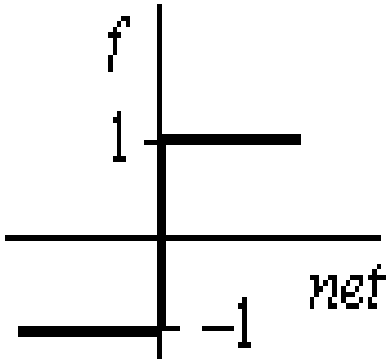
$f = \tanh(\alpha net)$

tanh



$$f = \frac{1}{1 + \exp(-\alpha net)}$$

logistic

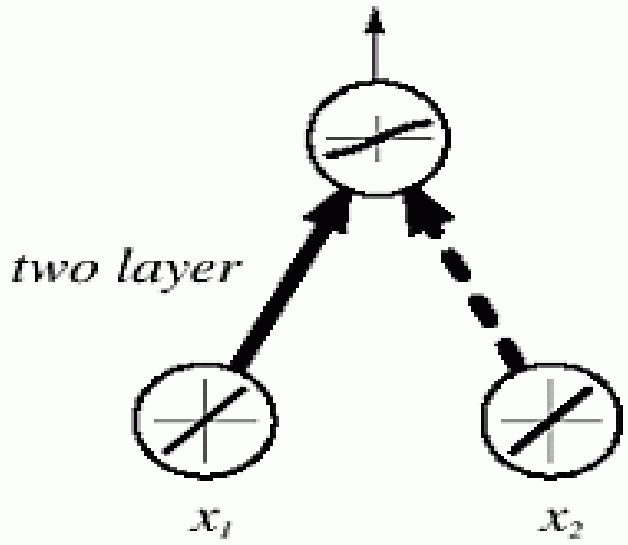


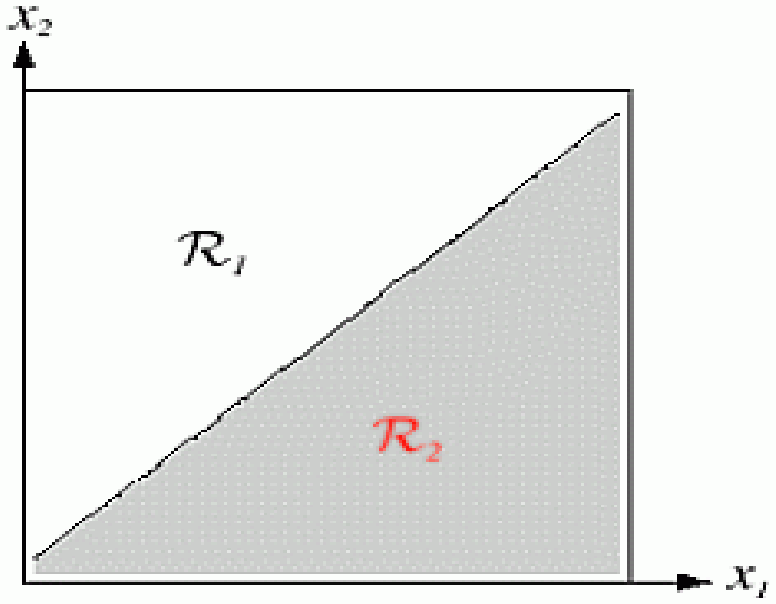
$$f = \begin{cases} 1 & net > 0 \\ -1 & net < 0 \end{cases}$$

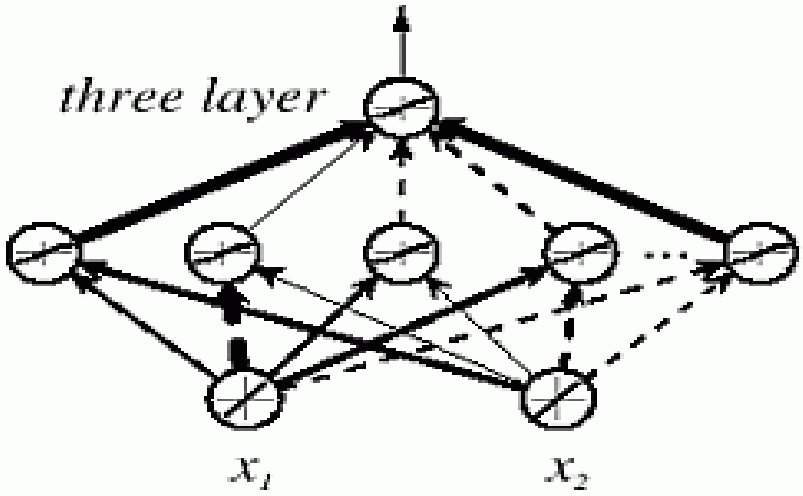
threshold

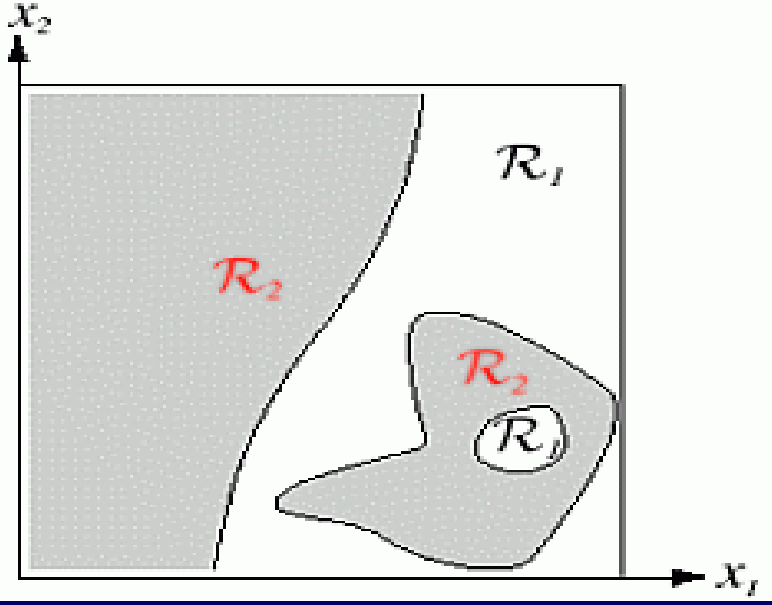
Bipolarne i unipolarne funkcje sigmoidalne.

Warstwa ukryta i granice decyzji

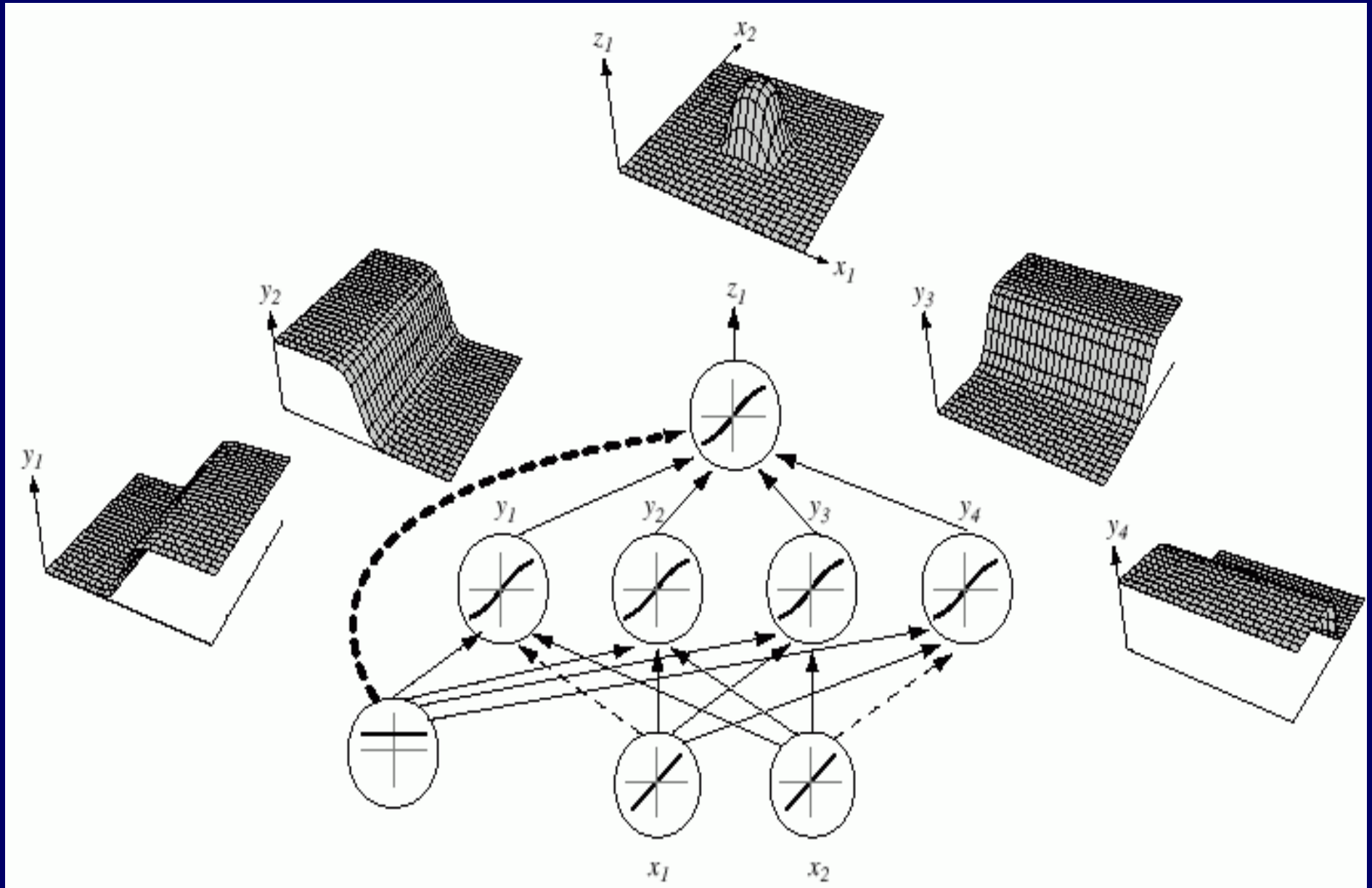




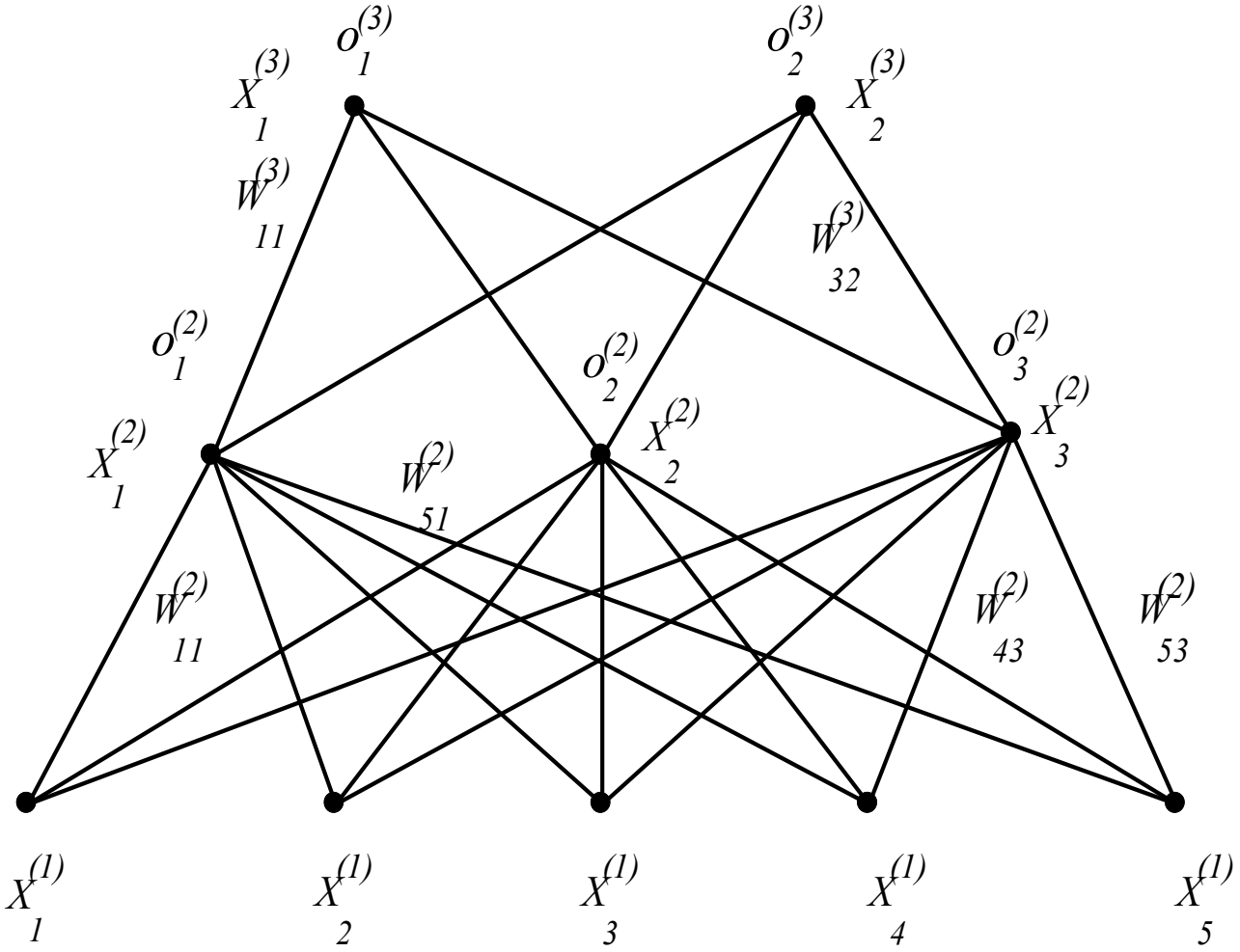




Sieć *Multilayer Perceptron MLP* 2-4-1



MLP Multilayer Perceptron



Problemy z MLP

- **Minimalizacja funkcji błędu jest zagadnieniem NP-trudnym**
- **Trudno jest dobrać optymalne parametry (liczba neuronów, warstw, wartości początkowe wag)**
- **Metody gradientowe wpadają w lokalne minima i zwalniają na plateau**
- **Zbieżność może być powolna**
- **Wyniki zależą od kolejności prezentacji danych**

Sigmoidy

Logistyczna funkcja aktywacji:

$$o_i = \sigma_i\left(X_i\right) = \frac{1}{\left(1+\exp\left[-\left(X_i-\theta_i\right)/T_i\right]\right)} = \frac{1}{\left(1+\exp\left[-\sum_j\left(W_{ij}o_j-\theta_i\right)/T_i\right]\right)}$$

Próg θ, nachylenie T

Pochodna osiąga max dla o=0,5

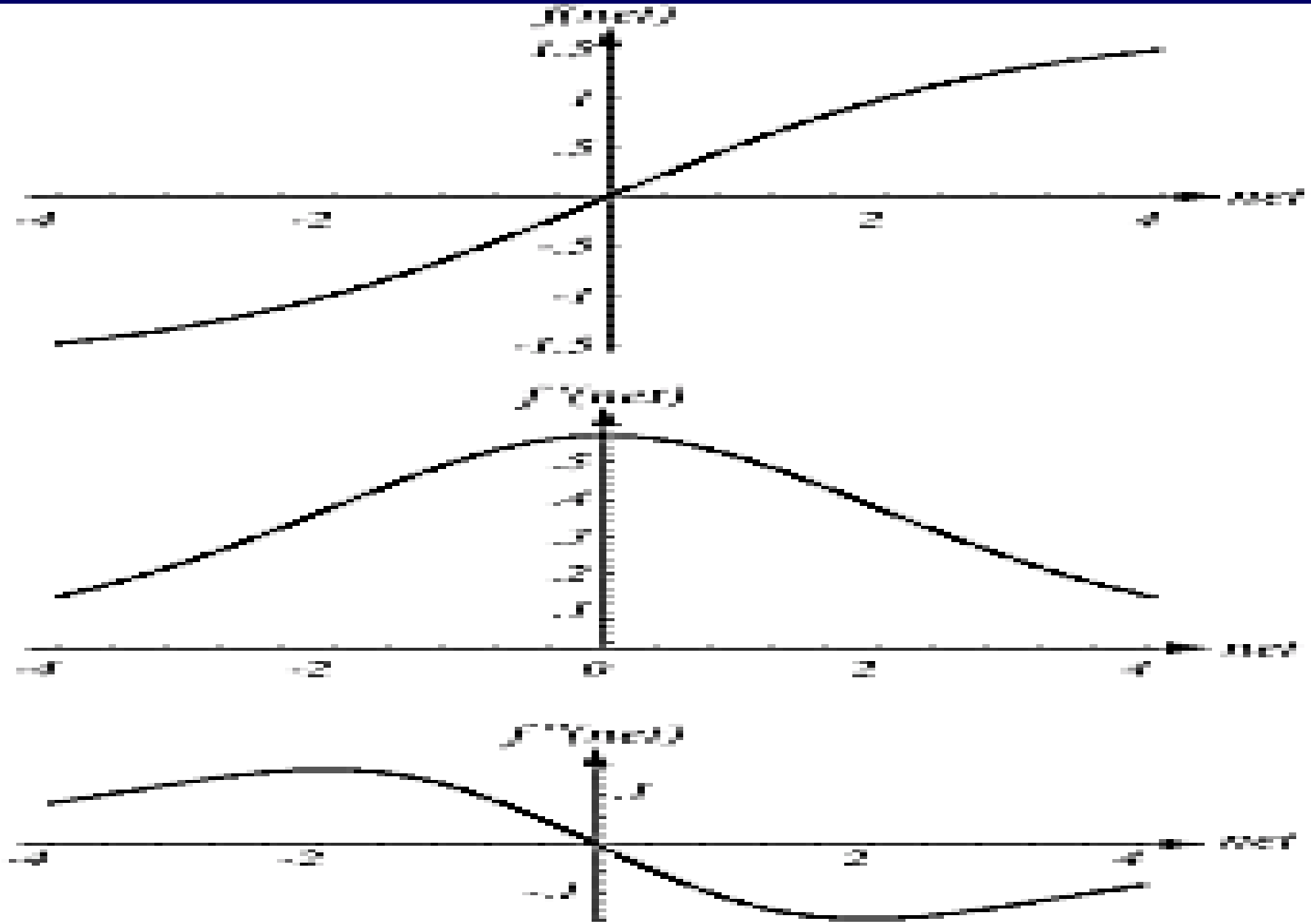
$$\sigma'\left(X_i\right) = \frac{\partial o_i}{\partial X_i} = o_i\left(1-o_i\right)$$

Błąd wyjściowego elementu:

$$\delta_i^{(M)} = o_i\left(1-o_i\right)\left(Y_i - F_i\left(X;W\right)\right)$$

$$\delta_j^{(l)} = o_j\left(1-o_j\right)\sum_i \delta_i^{(l+1)} W_{ij}^{(l)}$$

Sigmoidy



Problemy

- **Niewłaściwie dobrana architektura sieci**
- **Minima lokalne i plateau, wąskie „rynny”**
- **Wpływ nowych wzorców na już nauczone – zapominanie**
- **Szybkość uczenia – zagadnienie jest NP-trudne**
- **Schematy adaptacji dla stałej uczenia:**
  - **zwiększać η o $a=const$ dla malejącego błędu**
  - **zmniejszać o $-\eta b$ dla rosnącego błędu;**
  - **duże kroki na powierzchni gładkiej**
  - **drobne kroki na skomplikowanej powierzchni f. błędu.**

Ulepszenia MLP

- **Szybsze procedury minimalizacji błędu**
- **Modyfikacje schematu wstecznej propagacji**
- **Unikanie minimów lokalnych**
- **Funkcje celu, niekoniecznie MSE**
- **Inicjalizacja parametrów, lepszy start**
- **Regularyzacja i zwiększenie zdolność do generalizacji sieci - wybór modelu o odpowiedniej złożoności**
- **Funkcje transferu (aktywacji), nie tylko sigmoidy**

Minimalizacja funkcji błędu

Metody gradientowe 2 rzędu

$$E(X;W) = W_0 + (W - W_0)^T \cdot \nabla E(X;W)_{|W=W_0} + \frac{1}{2}(W - W_0)^T H (W - W_0)$$

$$H_{ij} = \frac{\partial^2 E(X;W)}{\partial W_i \partial W_j}$$ Hessjan - macierz drugich pochodnych

Metoda Newtona - w minimum gradient znika, więc rozwinięcie:

$$\nabla E(X;W) = \nabla E(X;W_0) + H \cdot (W - W_0) = 0$$

$$W = W_0 - H^{-1} \cdot \nabla E(X;W_0)$$

Wada: kosztowne odwracanie macierzy $O(n^3)$

Minimalizacja - metody liniowe

Wada metod 2-rzędu:

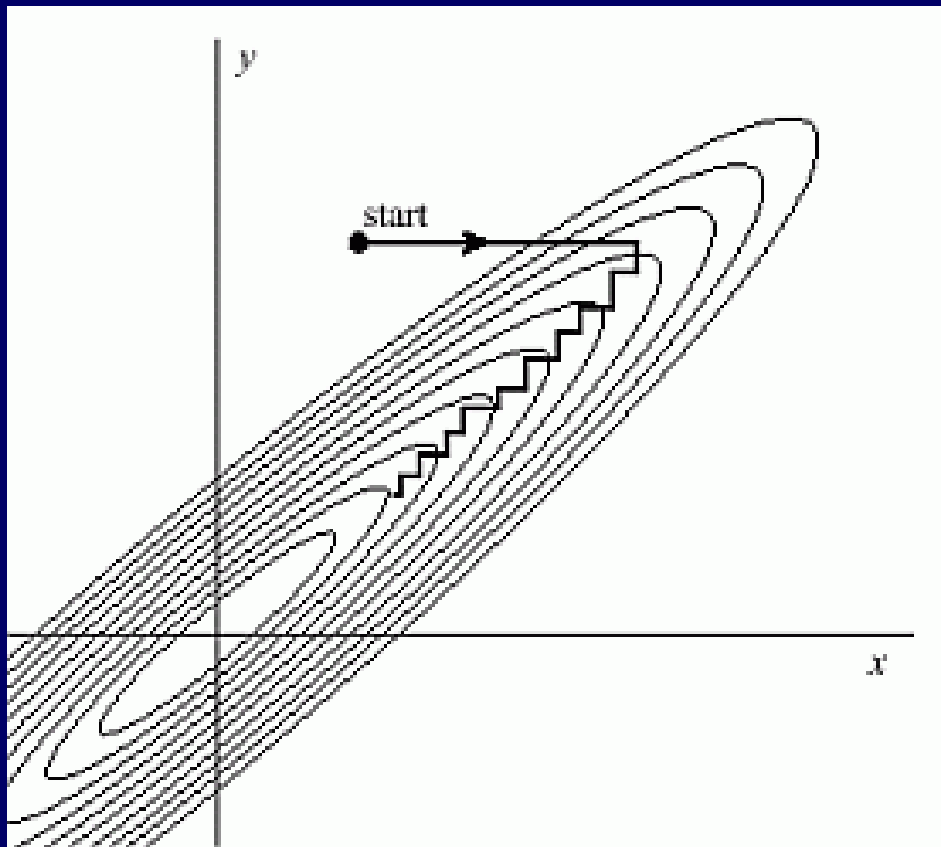
kosztowne odwracanie macierzy $O(n^3)$

Metoda najszybszego spadku: podążaj wzdłuż gradientu aż znajdziesz minimum w danym kierunku:

$$W = W_0 + \lambda K$$

minimalizacja $E(X;W(\lambda))$

oblicz gradient w punkcie $W(\lambda)$ jest prostopadły do poprzedniego



Quickprop

Quickprop (Fahlman 1988)

Jeśli wszystkie wagi są niezależne a powierzchnia błędu wyrażona funkcją kwadratową, to można dopasować parabolę.

Quickprop używa w tym celu 2 punkty + gradienty.

$$W(m+1) = W(m) - \frac{\nabla E\left(W = W_m\right)}{\nabla E\left(W = W_m\right) - \nabla E\left(W = W_{m-1}\right)} \Delta W_m$$

Wagi mają niezależne szybkości uczenia;

Złożoność jest O(n^2)

Rprop

Resilent BP (Riedmiller, Braun 1992)

Problemy ze zbyt małymi i dużymi gradientami.

Tylko znak gradientu jest używany do obliczenia poprawki:

$$\Delta W_{ij}(t) = -\eta_{ij}(t)\,\mathrm{sgn}\left(\frac{\partial E(\mathbf{W}(t))}{\partial W_{ij}}\right) = -\eta_{ij}(t)\,\mathrm{sgn}\left(\nabla_{ij}(t)\right)$$

Gradient używany jest do obliczenia współczynnika η

$$\eta_{ij}(t) = \begin{cases} \min\left(a\eta_{ij}(t-1), \eta_{\max}\right) \text{ dla } \nabla_{ij}(t)\nabla_{ij}(t-1) > 0 \\ \max\left(b\eta_{ij}(t-1), \eta_{\min}\right) \text{ dla } \nabla_{ij}(t)\nabla_{ij}(t-1) < 0 \\ \eta_{ij}(t-1) \text{ w poz. przyp.} \end{cases}$$

Wzrost η jeśli znak się nie zmienia, małe jeśli zmiana (oscylacje).

Np. a=1.2, b=0.5

Minimalizacja - CG

Metoda sprzężonych gradientów (ang. *conjugated gradients*):
dla form kwadratowych: startuj wzdłuż gradientu, potem wybierz nowy kierunek jako prostopadły do starego.

$$K^n = \beta K^s - \nabla E(X;W)$$

$$K^s \cdot \nabla E(X;W) = K^s \cdot \nabla E(X;W_0 + \lambda K^n) = 0$$

Po rozwinięciu gradientu

$$K^s \cdot \nabla E(X;W_0) + \lambda K^s \cdot H \cdot K^n = \lambda K^s \cdot H \cdot K^n = 0$$

Reguła Fletchera-Reevesa

$$\beta = \left(\nabla E^n\right)^2 \Big/ \left(\nabla E^s\right)^2$$

Polaka-Ribiera:

$$\beta = \left(\nabla E^n - \nabla E^s\right) \cdot \nabla E^n \Big/ \left(\nabla E^s\right)^2$$

Minimalizacja - CG

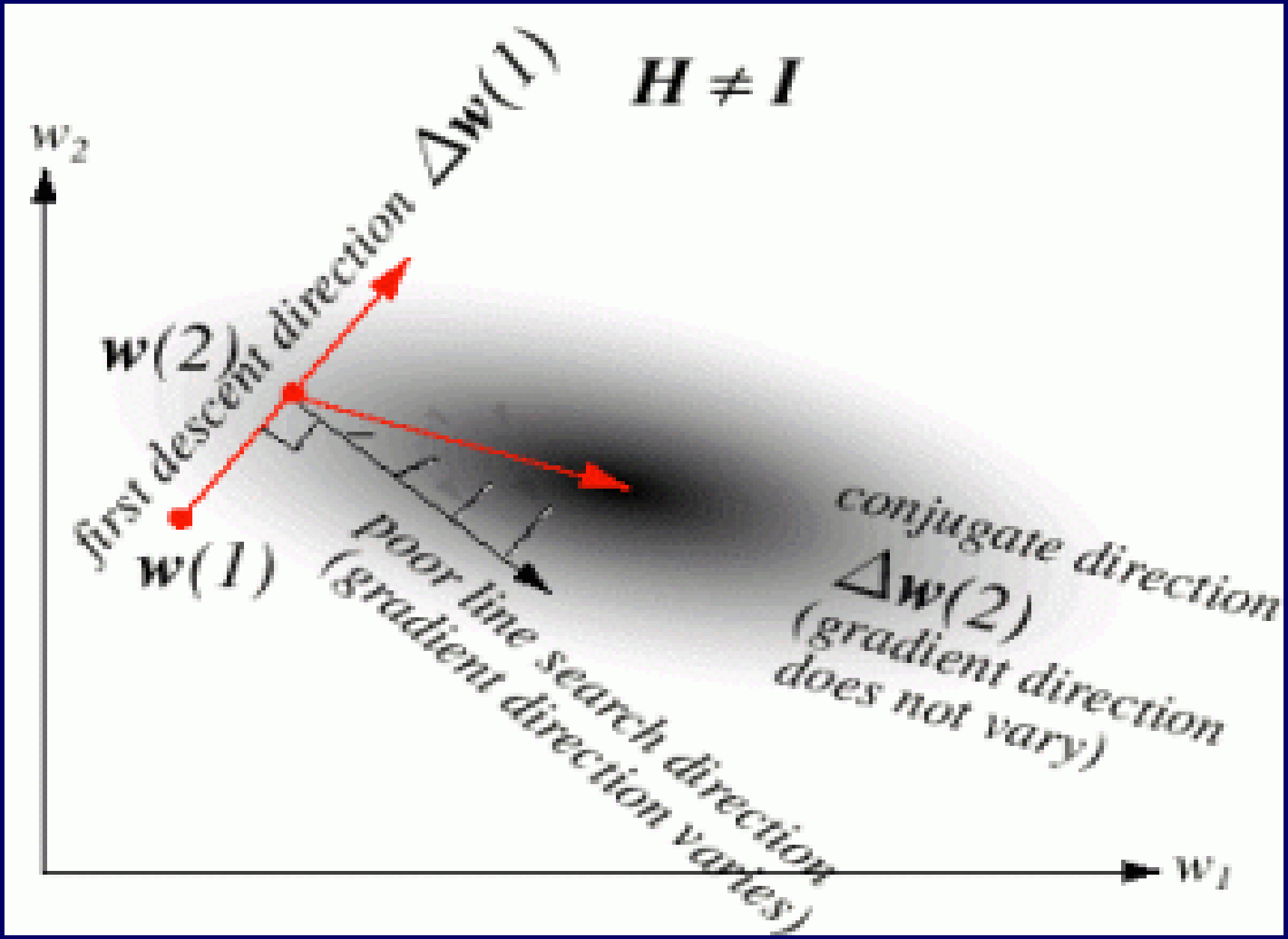
Wektory własne Hesjanu tworzą zbiór wektorów sprzężonych

Dla kwadratowej funkcji *E*(W) w *n*-wymiarach metoda CG osiąga minimum w *n* krokach; zbieżność kwadratowa

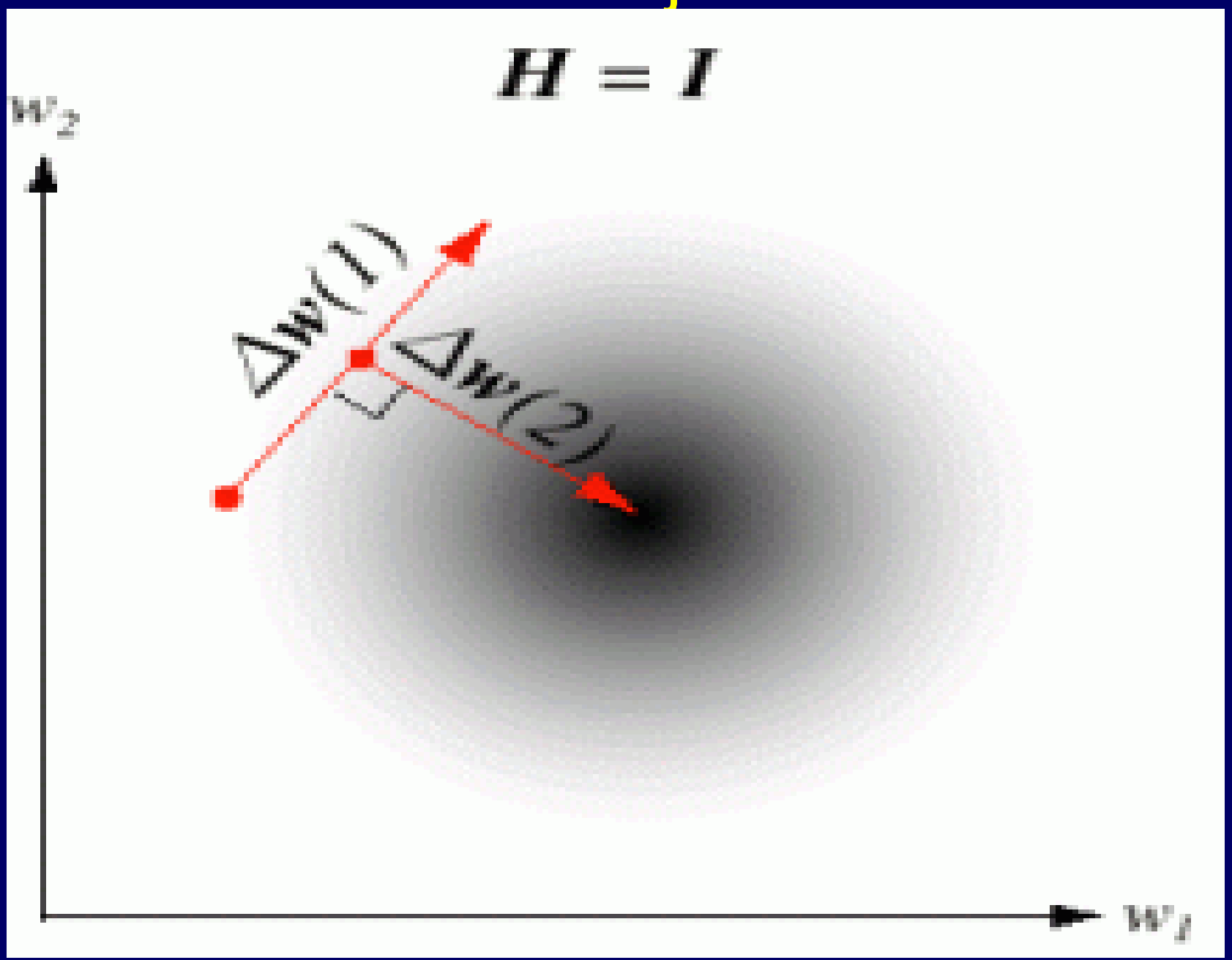
Metoda najszybszego spadku jest znacznie wolniejsza

SCG, Skalowana Metoda Sprz. Gradientów - szybka metoda szukania minimów wzdłuż prostej

Minimalizacja - CG



Minimalizacja - CG



Metody kwadratowe

Przybliżenia do Hesjanu:

zaniedbanie pozadiagonalnych elementów - metoda Newtona dla każdej wagi niezależnie.

$$W = W_0 - H^{-1} \cdot \nabla E\left(X; W_0\right)$$

$$\nabla W_{ij} = -\frac{\partial E}{\partial W_{ij}} \Bigg/ \frac{\partial^2 E}{\partial^2 W_{ij}}$$

Metoda zmiennej metryki - przybliżenie do H^{-1} oraz iteracyjna metoda Newtona, kwadratowo zbieżna.

Dwie wersje: DFP (Davidon-Fletcher-Power),
Broyden-Fletcher-Goldfarb-Shanno (BFGS).

Metoda Levenberg-Marquardta oparta na przybliżeniu Gaussa-Newtona.

Levenberg-Marquardt

Korzystamy z Jakobianu, który dla funkcji kwadratowej:

$$J_{ij} = \left[\frac{\partial E(X)}{\partial W_{ij}} \right]; \quad \Delta E(X) = \mathbf{J}^T E(X)$$

Jakobian można policzyć korzystając z wstecznej propagacji.
Przybliżenie do Hesjanu:

$$\mathbf{H} = \mathbf{J}^T \mathbf{J}$$

Parametry obliczamy korzystając z:

$$W_{k+1} = W_k - \left(\mathbf{J}^T \mathbf{J} - \mu \mathbf{I} \right)^{-1} \mathbf{J}^T E(X)$$

Dla $\mu = 0$ mamy metodę Newtona, a dla dużego największego spadku z małym krokiem; LM używa metod Newtona w pobliżu minimum, zmniejszając μ.

Globalna minimalizacja

•

Najprostsza metoda: wielokrotne starty.

Monte Carlo, symulowane wyżarzanie, metody multisympleksowe, tabu search, ...

Algorytmy genetyczne z sieciami MLP

Zalety: globalne, proste w realizacji, niektóre nie potrzebują gradientu, inne łączą zalety gradientowych z globalnymi.

Wady: zwykle kosztowne.

Szum dodawany do wag lub do danych pozwala uciec z płytszych minimów.

Inicjalizacja wag

Duże wagi => duża wariancja wyników, ale możliwe stają się dobre nieliniowe rozwiązania.

Za duże wartości wag: nasycone wartości sigmoid, małe gradienty => wolne uczenie.

Małe przypadkowe wagi, dające aktywacje rzędu 0,5 => szybkie uczenie i gładka aproksymacja => dobra generalizacja.

Zalecenia empiryczne $W_{ij} = \pm 0.78$

Battou $\pm a/\sqrt{N}$, a =2.38 by osiągnąć największą wariancję.

Inne próby inicjalizacji:

hiperpłaszczyzny z pojedynczych perceptronów;

wstępna klasteryzacja i płaszczyzny oddzielające klastry;

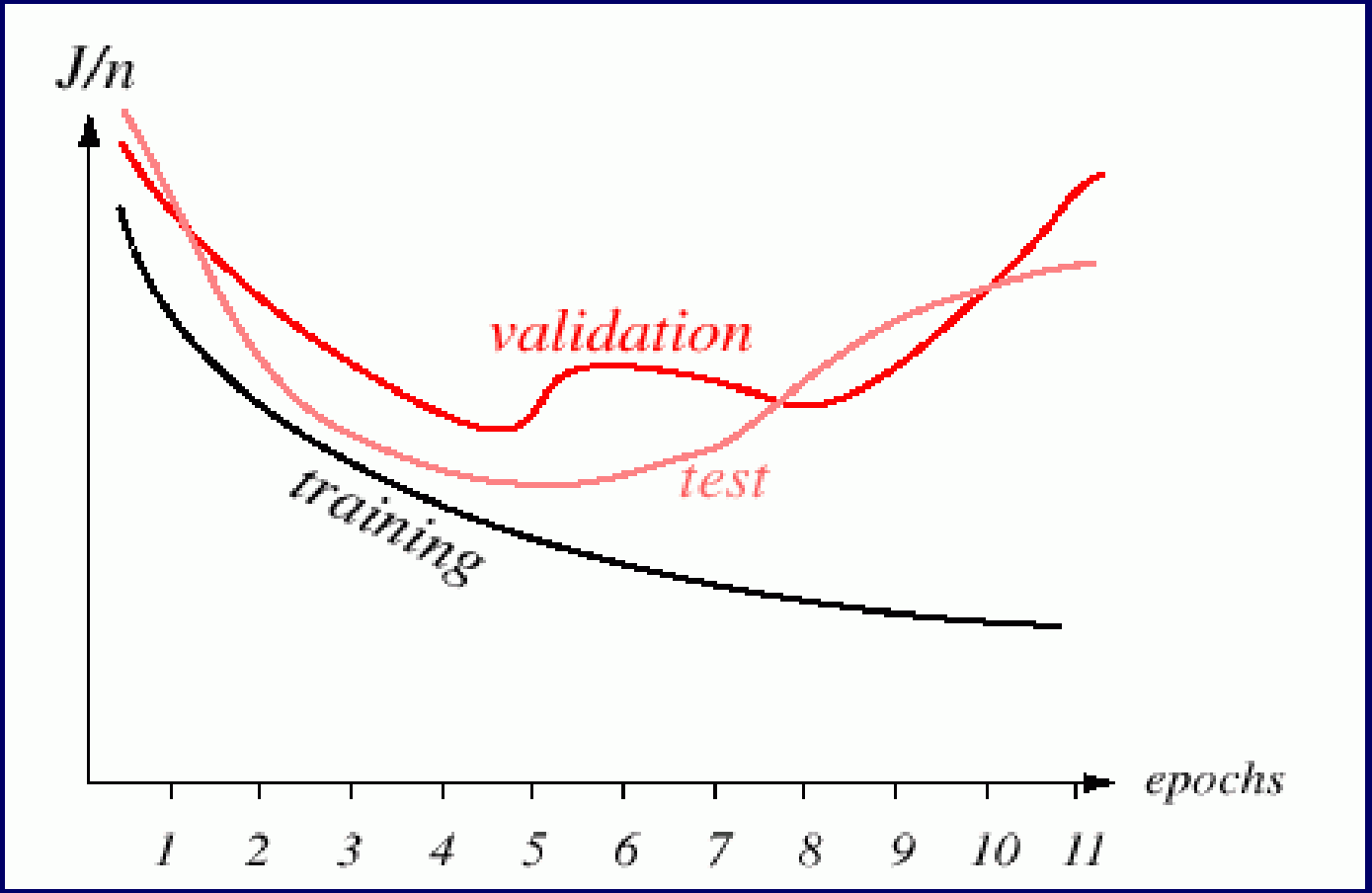
klasteryzacja w przestrzeni unormowanych wektorów.

Generalizacja

Wyniki na zbiorze treningowym mogą osiągnąć 100%

Celem jest osiągnięcie najlepszego wyniku dla nowych przypadków, nie pokazywanych wcześniej sieci.

Zbiór walidacyjny: pozwala na ocenę błędu generalizacji; oczekujemy korelacji wyników na zbiorze walidacyjnym i testowym.



Regularyzacja

Brzytwa Ockhama: najprostsze rozwiązania są najlepsze.
Zbyt złożona sieć - za dużo parametrów - marna generalizacja
Trudności w analizie funkcji realizowanej przez sieć.

Zalety małych wag: „gładka” funkcja.

$$E(X;W) = E_0(X;W) + \frac{1}{2}\gamma \sum_{ij} W_{ij}^{\ 2}$$

To jest równoważne dodatkowej zmianie wag:

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

Tu zanikają głownie duże wagi, a chodzi o zerowanie mniejszych.

Regularyzacja

Zmodyfikowany człon kary:

$$E(X;W) = E_0(X;W) + \frac{1}{2}\gamma \sum_{ij} \frac{W_{ij}^{\ 2}}{1 + W_{ij}^{\ 2}}$$

Równoważne dodatkowej zmianie wag:

$$W_{ij}^{\ n} = \left(1 - \frac{\gamma\eta}{\left(1 + W_{ij}^{\ s2}\right)^2} \right) W_{ij}^{\ s}$$

Małe wagi można usunąć i sieć dalej przetrenować - automatyczna selekcja cech.

Metoda „optimal brain damage” - upraszczanie sieci.

Rozpad synaps w mózgu - potrzebny do regularyzacji?

Szybkość zbieżności

Problem Title	Problem Type	Network Structure	Error Goal
SIN	Function Approx.	1-5-1	0.002
PARITY	Pattern Recog.	3-10-10-1	0.001
ENGINE	Function Approx.	2-30-2	0.005
CANCER	Pattern Recog.	9-5-5-2	0.012
CHOLESTEROL	Function Approx.	21-15-3	0.027
DIABETES	Pattern Recog.	8-15-15-2	0.05

Testy robione pakietem nnet z Matlaba:

Szybkość zbieżności

Metoda Levenberga-Marquardta dobra w aproksymacji dla sieci <1000 parametrów, ale słabsza w klasyfikacji, dużo RAM.

Rprop – dobry w klasyfikacji, słabszy w aproksymacji, mała pamięć.

SCG – szybka zbieżność jak w LM, mała pamięć.