POLSKA AKADEMIA NAUK
Instytut Badań Systemowych

WSPOMAGANIE DECYZJI
SYSTEMY EKSPERCKIE

pod redakcją

Romana Kulikowskiego i Lucyny Bogdan

Warszawa 1995

WSPOMAGANIE DECYZJI

SYSTEMY EKSPERCKIE

pod redakcją

Romana Kulikowskiego i Lucyny Bogdan

Warszawa 1995

Wydano z wykorzystaniem dotacji
KOMITETU BADAŃ NAUKOWYCH

Materiały konferencji: "Analiza Decyzyjna, Systemy Eksperckie, Zastosowania Systemów Komputerowych", Warszawa, 25-27 maja 1994r.

Komitet Programowy Konferencji:

Andrzej Ameljańczyk, Zdzisław Bubnicki, Wiesław Grudzewski, Olgierd Hryniewicz, Janusz Kacprzyk, Lech Kruś, Roman Kulikowski (przewodniczący), Kazimierz Mańczak, Ireneusz Nykowski, Zdzisław Pawlak, Roman Słowiński, Andrzej Straszak, Andrzej Weryński, Andrzej Wierzbicki.

Wykonano z oryginałów tekstowych dostarczonych przez autorów

ISBN 83-85847-85-5

Metody subgradientowe dla zadań klasyfikacji w sieciach neuronowych

Krzysztof C. Kiwiel, Piotr Kowalski i Bożena Łopuch
Instytut Badań Systemowych PAN, Warszawa

Streszczenie. Zadanie klasyfikacji sprowadza się w przypadku rozdzielnym liniowo do układu nierówności liniowych. Stosowana zwykle metoda perceptronu jest wolno zbieżna. Proponujemy wykorzystanie metod z rzutowaniem na półprzestrzenie adaptacyjnie agregowanych nierówności. Metody te nadają się do obliczeń równoległych. Dla przypadku nierozdzielnego liniowo rozważamy konstrukcję drzewa neuronowego, którego kolejne rozgałęzienie odpowiada hiperpłaszczyźnie rozdzielającej podzbiór dotychczas niesklasyfikowany w danej gałęzi. W odróżnieniu od klasycznych metod, nie wymaga się aby hiperpłaszczyzna ta klasyfikowała prawidłowo choć jeden punkt. Daje to dużą swobodę w konstrukcji hiperpłaszczyzn, niezależnie od możliwej degeneracji.

1 Wprowadzenie

Jednym z podstawowych problemów rozwiązywanych w sieciach neuronowych jest zadanie klasyfikacji (zob. [BeM92a, BeM92b, HKP91, Man94, MSW90, Tad93]).

Definicja 1.1. Niech $\mathcal{A}$ i $\mathcal{B}$ oznaczają dwa zadane, niepuste, skończone i rozłączne zbiory w przestrzeni euklidesowej $\mathbb{R}^N$ o wymiarze N. *Zadanie klasyfikacji* wymaga wyznaczenia *funkcji rozdzielającej* $\phi : \mathbb{R}^N \to \mathbb{R}$ takiej, że $\phi(\mathcal{A}) < 0$ i $\phi(\mathcal{B}) > 0$, tj.

$$\phi(a) < 0 \quad \forall a \in \mathcal{A} \qquad \text{oraz} \qquad \phi(b) > 0 \quad \forall b \in \mathcal{B}. \tag{1}$$

Zbiory $\mathcal{A}$ i $\mathcal{B}$ są *rozdzielne liniowo*, jeżeli posiadają one liniową (afiniczną) funkcję rozdzielającą postaci $\phi(\cdot) = \langle \cdot, w \rangle + \gamma$, gdzie $w \in \mathbb{R}^N$, $\gamma \in \mathbb{R}$, zaś $\langle \cdot, \cdot \rangle$ oznacza standardowy iloczyn skalarny w $\mathbb{R}^N$.

W przypadku rozdzielnym liniowo, parametry (w, γ) funkcji rozdzielającej ϕ postaci $\phi(\cdot) = \langle \cdot, w \rangle + \gamma$ wyznacza się zwykle metodą perceptronu (zob. [HKP91, Tad93]), rozwiązując układ nierówności liniowych względem (w, γ) postaci

$$\langle a, w \rangle + \gamma \leq -1, \ \forall a \in \mathcal{A}, \quad \langle b, w \rangle + \gamma \geq 1, \ \forall b \in \mathcal{B}. \tag{2}$$

Metoda perceptronu jest bardzo prosta, ale jest ona (w ogólnym przypadku) wolno zbieżna. W pracy niniejszej proponujemy wykorzystywać dla zadań klasyfikacji metody z rzutowaniem na półprzestrzenie adaptacyjnie agregowanych nierówności układu (2). Współczynniki agregacji dobiera się minimalizując oszacowanie górne odległości bieżącego przybliżenia rozwiązania od zbioru rozwiązań. Metody te wprowadzono w [Kiw94, Kiw93] dla ogólnych wypukłych problemów dopuszczalnych. Są one, ogólnie biorąc, znacznie szybciej zbieżne od prostych metod relaksacyjnych, takich jak metoda perceptronu. Ponadto ich implementacje mogą wykorzystywać obliczenia równoległe [BDG+93].

W przypadku nierozdzielnym liniowo, rozważane w literaturze modyfikacje metody perceptronu nie gwarantują wyznaczenia funkcji separującej (por. [HKP91]). Ponadto uczenie nieliniowej sieci neuronowej metodą wstecznej propagacji błędów (backpropagation) może być mniej efektywne niż wyznaczanie kawałkami liniowej funkcji rozdzielającej przez rozwiązanie ciągu zadań programowania liniowego [BeM92a, BeM92b]. Z kolei metody oparte na programowaniu liniowym [BeM92a, BeM92b] wymagają bardzo kosztownych procedur antydegeneracyjnych w celu zapobieżenia pętleniu się algorytmu. W związku z tym, w pracy wprowadzamy konstrukcję drzewa neuronowego, którego kolejne rozgałęzienie odpowiada hiperpłaszczyźnie rozdzielającej podzbiór dotychczas niesklasyfikowany w danej gałęzi. W odróżnieniu od klasycznych metod, nie wymaga się aby hiperpłaszczyzna ta klasyfikowała prawidłowo choć jeden punkt. Daje to dużą swobodę w konstrukcji hiperpłaszczyzn, niezależnie od możliwej degeneracji.

2 Klasyfikacja w przypadku rozdzielnym liniowo

W rozdziale tym rozważamy wykorzystanie metod rzutowych z agregacją z [Kiw94] do wyznaczenia parametrów funkcji klasyfikującej przez rozwiązanie układu nierówności liniowych (2).

Zacznijmy od wprowadzenia niezbędnych oznaczeń. Dla $x, y \in \mathbb{R}^N$, $\langle x, y \rangle = \sum_{i=1}^N x_i y_i$ i $|x| = \langle x, x \rangle^{1/2}$ definiują iloczyn skalarny i normę w $\mathbb{R}^N$. $B(x, r) = \{y : |y - x| \leq r\}$ oznacza kulę o środku w punkcie x i promieniu $r \geq 0$. Niech C będzie zbiorem domkniętym i wypukłym w $\mathbb{R}^N$. $P_C(\cdot) = \arg\min_{x \in C} |\cdot - x|$ oznacza operator *rzutu* metrycznego na C, zaś $d_C(\cdot) = |\cdot - P_C(\cdot)|$ jest funkcją *odległości* od C; wygodnie jest przyjąć, że $P_C(\cdot) \equiv \cdot$ i $d_C(\cdot) \equiv \infty$ gdy $C = \emptyset$. Przypomnijmy, że dla hiperpłaszczyzny $H = \{x : \langle a, x \rangle \leq b\}$, gdzie $a \in \mathbb{R}^N$ i $b \in \mathbb{R}$, wyznaczenie rzutu jest bardzo proste: $P_H(x) = x - (\langle a, x \rangle - b)_+ a/|a|^2$, gdzie $(\cdot)_+ = \max\{\cdot, 0\}$.

Niech $T = [t_{\min}, t_{\max}] \subset (0,2)$ oznacza zbiór *dopuszczalnych współczynników kroku*. Operator *relaksacji* dla C z krokiem $t \in T$

$$R_{C,t}(x) = x + t[P_C(x) - x] \tag{3}$$

ma własność *zwężenie Fejérowskiego* (zob. np. [Kiw94])

$$|c - R_{C,t}(x)|^2 \leq |c - x|^2 - t(2-t)|x - P_C(x)|^2 \quad \forall c \in C. \tag{4}$$

Definicja 2.1. Niech C będzie zbiorem domkniętym i wypukłym w $\mathbb{R}^N$. Niech $\mathbf{H}_C$ oznacza rodzinę półprzestrzeni domkniętych zawierających C, wraz z $\mathbb{R}^N$, oraz $\emptyset$ gdy $C = \emptyset$. Odwzorowanie $\mathcal{H}_C : \mathbb{R}^N \to 2^{\mathbf{H}_C}$ jest *cięciem* dla C gdy $x \mapsto \delta_{\mathcal{H}_C}(x) = \inf_{H_C(x) \in \mathcal{H}_C(x)} d_{H_C(x)}(x)$ jest *separatorem* dla C, tj. $z^\infty \in C$ gdy $z^k \to z^\infty$ i $\delta_{\mathcal{H}_C}(z^k) \to 0$ przy $k \to \infty$.

Rozważmy następujący *wypukły problem dopuszczalny*: dla zadanej skończonej rodziny $\{C_i\}_{i=1}^m$ zbiorów domkniętych i wypukłych w $\mathbb{R}^N$, należy wyznaczyć (o ile jest to możliwe) punkt x leżący w przecięciu $\cap_{i=1}^m C_i$. W interesującym nas zastosowaniu, zbiory C_i odpowiadają kolejnym półprzestrzeniom nierówności liniowych układu (2).

Przedstawimy teraz metodę z [Kiw94] do rozwiązywania wypukłych zadań dopuszczalnych.

Algorytm 2.2 (poszukujący punktu w $\bigcap_{i=1}^m C_i$).
Krok 0 (*Inicjacja*). Wybierz punkt początkowy $x^1 \in \mathbb{R}^N$, tolerancję dopuszczalności $\epsilon_f \geq 0$ i początkowy promień lokalizacji $r_1 \geq d_C(x^1)$ ($r_1 = \infty$ gdy nie można oszacować $d_C(x^1)$). Ustaw licznik iteracji $k = 1$.

Krok 1 (*Wybór zbioru roboczego*). Wybierz $\emptyset \neq I^k \subset I$ tak, aby $|\{k : i \in I^k\}| = \infty$ dla $i = 1{:}m$ (tzn. każdy indeks i powinien wchodzić do zbioru I^k nieskończenie często gdy $k \to \infty$).

Krok 2 (*Przybliżony rzut*). Wybierz półprzestrzeń cięcia $\check{H}^k \in \check{\mathcal{H}}_{C^k}(x^k)$ dla $C^k = \cap_{i \in I^k} C_i$. Jeżeli $\check{H}^k = \emptyset$, ogłoś "$\cap_i C_i = \emptyset$" i STOP.

Krok 3 (*Kryterium stopu*). Jeśli $I^k = I$ i $|x^k - P_{\check{H}^k}(x^k)| \leq \epsilon_f$, to STOP.

Krok 4 (*Relaksacja*). Wybierz krok $t_k \in T$ i podstaw (por. (3) i (4))

$$x^{k+1} = R_{\check{H}^k, t_k}(x^k) = x^k + t_k[P_{\check{H}^k}(x^k) - x^k],$$

$$\sigma_k = t_k(2 - t_k) d^2_{\check{H}^k}(x^k).$$

Krok 5 (*Wykrywanie sprzeczności*). Jeżeli $r_k^2 < \sigma_k$, ogłoś "$\cap_i C_i = \emptyset$" i STOP.

Krok 6 (*Poprawa lokalizacji*). Podstaw $r_{k+1} = (r_k^2 - \sigma_k)^{1/2}$, $k \leftarrow k+1$ i wróć do kroku 1.

Test wykrywający niedopuszczalność w kroku 5 jest uzasadniony następującym wynikiem z [Kiw94].

Lemat 2.3. *Jeżeli* $\cap_{i=1}^m C_i \neq \emptyset$ *i* $r_1 \geq d_{\cap_i C_i}(x^1)$ *to* $\emptyset \neq \bigcap_{i=1}^m C_i \cap B(x^k, r_k) \subset \bigcap_{i=1}^m C_i \cap B(x^{k+1}, r_{k+1})$ *dla wszystkich* k.

Algorytm 2.2 jest *zbieżny globalnie* [Kiw94].

Twierdzenie 2.4. *Jeżeli* $\cap_{i=1}^m C_i \neq \emptyset$ *to ciąg* $\{x^k\}$ *zbiega do punktu w* $\cap_{i=1}^m C_i$.

Podamy teraz przykłady cięć dla $C = \{x : \langle a^j, x\rangle \leq b_j, j \in J\}$, gdzie $|J| < \infty$. Dla $\lambda_{\min} \in (0, \frac{1}{|J|}]$ i $\hat{x} \notin C$, niech $H_C(\hat{x}) = \{x : \langle \hat{a}, x\rangle \leq \hat{b}\}$ odpowiada *nierówności zastępczej* (*zagregowanej*) z *wagami* $\lambda_j \geq 0$, $\sum_j \lambda_j = 1$, $(\hat{a}, \hat{b}) = \sum_j \lambda_j (a^j, b_j)$ i

$$d_{H_C(\hat{x})}(\hat{x}) = \frac{\langle \hat{a}, \hat{x}\rangle - \hat{b}}{|\hat{a}|} \geq \lambda_{\min} \frac{\max_{j\in J}(\langle a^j, \hat{x}\rangle - b_j)}{\max_{j\in J}|a^j|}, \tag{5}$$

np. $\lambda_{\hat{\jmath}} \geq \lambda_{\min}$ dla pewnego $\hat{\jmath} \in \operatorname{Arg\,max}_j(\langle a^j, \hat{x}\rangle - b_j)$ i $\lambda_j = 0$ gdy $\langle a^j, \hat{x}\rangle < b_j$, lub $\lambda_j = (\langle a^j, \hat{x}\rangle - b_j)_+^\gamma / \sum_i (\langle a^i, \hat{x}\rangle - b_i)_+^\gamma$, $j \in J$, przy $\gamma \geq 0$; czy też $\lambda_j = [(\langle a^j, \hat{x}\rangle - b_j)_+ / |a^j|^2] / \sum_i [(\langle a^i, \hat{x}\rangle - b_i)_+ / |a^i|^2]$ $\forall j \in J$.

Najlepsza półprzestrzeń zastępcza odpowiada wagom maksymalizującym odległość $d_{H_C(\hat{x})}(\hat{x})$, tj.

$$\check{\lambda} \in \operatorname{Arg\,max} \left\{ \frac{\sum_{j\in J} \lambda_j (\langle a^j, \hat{x}\rangle - b_i)}{|\sum_{j\in J} \lambda_j a^j|} : \lambda_j \geq 0, j \in J, \sum_j \lambda_j = 1 \right\}.$$

Wyznaczając mnożnik Lagrange'a zadania rzutowego

$$P_C(\hat{x}) = \arg\min\{\, |x - \hat{x}|^2/2 : \langle a^j, x\rangle \leq b_j, j \in J \,\},$$

czyli rozwiązanie zadania dualnego

$$\hat{\lambda} \in \operatorname{Arg\,min} \left\{ |\sum_{j\in J} \lambda_j a^j|^2/2 + \sum_{j\in J} \lambda_j (b_j - \langle a^j, \hat{x}\rangle) : \lambda_j \geq 0, j \in J \right\}, \tag{6}$$

można podstawić $\check{\lambda} = \hat{\lambda} / \sum_j \hat{\lambda}_j$. Odpowiednie procedury programowania kwadratowego [Kiw94] pozwalają w dwóch iteracjach wyznaczyć przybliżone rozwiązanie zadania (6), dla którego obowiązuje oszacowanie (5) z $\lambda_{\min} = 1$.

Wersja blokowo-równoległa metody dla $C_i = \{x : \langle a^i, x\rangle \leq b_i\}$ wygląda następująco. Niech $I^k = \cup_{j=1}^{m_k} I_j^k$. Dla każdego bloku $j = 1{:}m_k$, wybierz wagi $\bar{w}_i^{jk} \geq 0$, $i \in I_j^k$, $\sum_{i\in I_j^k} \bar{w}_i^{jk} = 1$, dla *zagregowanych półprzestrzeni blokowych*

$$\bar{H}_j^k = \{x : \langle \bar{a}^{jk}, x\rangle \leq \bar{b}_{jk}\}, \quad (\bar{a}^{jk}, \bar{b}_{jk}) = \sum_{i\in I_j^k} \bar{w}_i^{jk}(a^i, b_i)$$

z $d_{\bar{H}_j^k}(x^k) \geq w_{\min} \max_{i\in I_j^k} d_{C_i}(x^k)$, gdzie $w_{\min} > 0$. Następnie wybierz wagi $\hat{w}_j^k \geq 0$, $j = 1{:}m_k$, $\sum_{j=1}^{m_k} \hat{w}_j^k = 1$, dla półprzestrzeni zagregowanej

$$\check{H}^k = \{x : \langle \hat{a}^k, x\rangle \leq \hat{b}_k\}, \quad (\hat{a}^k, \hat{b}_k) = \sum_{j=1}^{m_k} \hat{w}_j^k(\bar{a}^{jk}, \bar{b}_{jk})$$

z $d_{\check{H}^k}(x^k) \geq w_{\min} \max_{j=1:m_k} d_{\bar{H}_j^k}(x^k)$. Wyznaczanie ograniczeń zastępczych dla poszczególnych bloków może przebiegać równolegle. Dla zwiększenia szybkości zbieżności, agregacja odrębnych bloków ograniczeń może wykorzystywać ograniczenia zastępcze innych bloków z poprzednich iteracji.

Metody omawiane dotychczas znajdują rozwiązanie układu (2) zależne od ich punktu startowego. *Perceptron optymalny* daje rozdzielenie centralne funkcją $\phi(\cdot) = \langle \hat{w}, \cdot\rangle + \hat{\gamma}$ przy

$$\hat{w} = \arg\min\{\max_{a\in\mathcal{A}} \langle a, w\rangle - \min_{b\in\mathcal{B}} \langle b, w\rangle : |w| = 1\}, \tag{7}$$

$$\hat{\gamma} = (\max_{a\in\mathcal{A}} \langle a, \hat{w}\rangle - \min_{b\in\mathcal{B}} \langle b, \hat{w}\rangle)/2.$$

Dla $f(x) = \max_{a\in\mathcal{A}, b\in\mathcal{B}} \langle a - b, x\rangle$ i $S = \{x : |x| \leq 1\}$, odpowiada to zadaniu wypukłemu

$$f^* = \min\{f(x) : x \in S\}.$$

Opracowane ostatnio metody subgradientowe [Kiw93] dla tego zadania mogą wykorzystywać obliczenia równoległe.

3 Klasyfikacja metodą drzewa neuronowego

W przypadku nierozdzielnym liniowo, zadanie pomocnicze (2) nie ma rozwiązań, co znacznie komplikuje działanie metod opartych na programowaniu liniowym [BeM92a, BeM92b, Man94, MSW90]. W kroku $l \geq 0$ metody z [MSW90], oznaczając przez $\mathcal{A}^l$ i $\mathcal{B}^l$ niesklasyfikowane dotychczas podzbiory $\mathcal{A}$ i $\mathcal{B}$, znajduje się kierunek hiperpłaszczyzny odcinającej (por. (7))

$$\hat{w}^l \in \mathrm{Arg}\max\{\min_{a\in\mathcal{A}^l} \langle a, w\rangle - \max_{b\in\mathcal{B}^l} \langle b, w\rangle : \|w\|_\infty = 1\}, \tag{8}$$

gdzie $\|w\|_\infty = \max_{i=1:n} |w_i|$. Wymaga to rozwiązania $2n$ zadań programowania liniowego (z ograniczeniami postaci $-1 \leq w_j \leq 1$, $j \neq i$, $w_i = \pm 1$, dla $i = 1{:}n$). Zamiast przeszukiwać $2n$ ścian hipersześcianu, proponujemy przeglądanie $n+1$ ścian sympleksu. Niech $\{\Sigma^j\}_{j=1}^{n+1}$ będą ścianami pewnego sympleksu n-wymiarowego, którego wnętrze zawiera 0. Zastąpienie (8) przez

$$\hat{w}^l \in \operatorname{Arg\,max}\{\min_{a\in\mathcal{A}^l} \langle a, w\rangle - \max_{b\in\mathcal{B}^l} \langle b, w\rangle : w \in \bigcup_{j=1}^{n+1} \Sigma^j\} \tag{9}$$

wymaga rozwiązania $n+1$ zadań programowania liniowego, dając prawie dwukrotne zmniejszenie nakładu obliczeń.

Ponieważ konstrukcje (8) i (9) nie gwarantują odcięcia co najmniej jednego z punktów $\mathcal{A}$ lub $\mathcal{B}$, metody takie wymagają w ogólności używania potencjalnie kosztownych procedur antydegeneracyjnych w celu zapobiegania pętleniu się metody [MSW90]. Podobne trudności występują w wielowarstwowych metodach perceptronowych [HKP91, §6.6]. Ponadto metody perceptronowe mogą nie znaleźć odcięcia nawet w przypadkach niezdegenerowanych. Do usunięcia tych trudności proponujemy wykorzystywać odpowiednie *drzewo neuronowe*.

Algorytm 3.1 (budujący klasyfikujące drzewo neuronowe)

(a) *Inicjacja*: Podstaw poziom drzewa $i = 0$, $\mathcal{A}^{01} = \mathcal{A}$, $\mathcal{B}^{01} = \mathcal{B}$.

(b) *Na poziomie drzewa i*: Dla $j = 1, 2, \ldots, 2^i$:

- Stop jeżeli $\mathcal{A}^{ij} = \emptyset$ lub $\mathcal{B}^{ij} = \emptyset$ dla wszystkich j.
- Dla $\mathcal{A}^{ij} \neq \emptyset$ i $\mathcal{B}^{ij} \neq \emptyset$ wyznacz $w^{ij} \neq 0$ i γ^{ij} takie, że $\{c \in \mathcal{A}^{ij} \cup \mathcal{B}^{ij} : \langle c, w^{ij}\rangle \leq \gamma^{ij}\} \neq \mathcal{A}^{ij} \cup \mathcal{B}^{ij}$.
- Dla $\mathcal{A}^{ij} \neq \emptyset$ i $\mathcal{B}^{ij} \neq \emptyset$ wyznacz potomków

$$\mathcal{A}^{(i+1)(2j-1)} = \{a \in \mathcal{A}^{ij} : \langle a, w^{ij}\rangle \leq \gamma^{ij}\},$$

$$\mathcal{B}^{(i+1)(2j-1)} = \{b \in \mathcal{B}^{ij} : \langle b, w^{ij}\rangle \leq \gamma^{ij}\},$$

$$\mathcal{A}^{(i+1)(2j)} = \{a \in \mathcal{A}^{ij} : \langle a, w^{ij}\rangle > \gamma^{ij}\},$$

$$\mathcal{B}^{(i+1)(2j)} = \{b \in \mathcal{B}^{ij} : \langle b, w^{ij}\rangle > \gamma^{ij}\}.$$

(c) *Zwiększ poziom drzewa*: $i \leftarrow i+1$ i wróć do (b).

Kolejne rozgałęzienie tego drzewa w kroku (b) odpowiada hiperpłaszczyźnie rozdzielającej podzbiór $\mathcal{A}^{ij} \cup \mathcal{B}^{ij}$, dotychczas niesklasyfikowany w danej gałęzi. Najprostsza (aczkolwiek najmniej efektywna) implementacja kroku (b) może bazować na fakcie, że przy $a \in \mathcal{A}^{ij}$, $b \in \mathcal{B}^{ij}$, $a \neq b$, kładąc $w^{ij} = (b-a)/|b-a|$ i $\gamma^{ij} = -|b-a|/2$ zapewnia się $\langle a, w^{ij}\rangle < \gamma^{ij} < \langle b, w^{ij}\rangle$. Oczywiście

bardziej wyszukany dobór hiperpłaszczyzn może skrócić gałęzie, zwiększając efektywność klasyfikacji. W odróżnieniu od klasycznych metod, nie wymaga się aby hiperpłaszczyzny takie klasyfikowały prawidłowo choć jeden punkt (poza trywialnym przypadkiem podzbioru dwuelementowego). Daje to dużą swobodę w konstrukcji hiperpłaszczyzn, niezależnie od możliwej degeneracji. Można do tego wykorzystywać metody programowania liniowego [BeM92a, BeM92b], lub znajdować rozwiązania przybliżone pomocniczych zadań optymalizacji metodami subgradientowymi [Kiw93], które łatwiej zaprogramować w wersji równoległej. Ponadto, niezależnie od sposobu wyznaczania hiperpłaszczyzn rozdzielających, konstrukcja drzewa neuronowego polega w istocie na rekursywnym dzieleniu problemu wyjściowego na mniejsze podzadania, co umożliwia efektywne wykorzystanie obliczeń równoległych.

Literatura

[BDG+93] A. Beguelin, J. Dongarra, A. Geist, R. Manchek and V. Sunderam, *A users' guide to* PVM*: Parallel virtual machine*, Tech. Report 11826, Mathematical Sciences Section, Oak Ridge National Laboratory, Oak Ridge, TN, 1993.

[BeM92a] K. P. Bennet and O. L. Mangasarian, *Neural network training via linear programming*, in Advances in Optimization and Parallel Computing, P. M. Pardalos, ed., North-Holland, Amsterdam, 1992, pp. 56–67.

[BeM92b] ———, *Robust linear programming discrimination of two linearly inseparable sets*, Optimization Methods & Software **1** (1992) 23–34.

[HKP91] J. Hertz, A. Krogh and R. G. Palmer, *Introduction to the Theory of Neural Computation*, Addison-Wesley, Reading, MA, 1991.

[Kiw93] K. C. Kiwiel, *The efficiency of subgradient projection methods for convex nondifferentiable optimization*, Research Report 1845, INRIA, Rocquencourt, 1993.

[Kiw94] ———, *Block-iterative surrogate projection methods for convex feasibility problems*, Linear Algebra Appl. ? (1994). To appear.

[Man94] O. L. Mangasarian, *Mathematical programming in neural networks*, ORSA J. Comput. **5** (1994) 349–360.

[MSW90] O. L. Mangasarian, R. Setiono and W. H. Wolberg, *Pattern recognition via linear programming: Theory and application to medical diagnosis*, in Large-Scale Numerical Optimization, T. F. Coleman and Y. Li, eds., SIAM Publications, Philadelphia, 1990, pp. 22–31.

[Tad93] R. Tadeusiewicz, *Sieci neuronowe*, Akademicka Oficyna Wydawnicza, Warszawa, 1993.

ISBN 83-85847-85-5

W celu uzyskania bliższych informacji i zakupu dodatkowych egzemplarzy prosimy o kontakt
z Instytutem Badań Systemowych PAN
ul. Newelska 6, 01-447 Warszawa
tel. 36-19-01 w. 241 e-mail: kotuszew@ibspan.waw.pl