

{plik w6pca.tex 20.11.2008, popr. 30.03.2009, 20.11.09}

6 Uczenie wg. zasady Hebba, Składowe główne

Przechodzimy teraz do omawiania uczenia nie nadzorowanego (ang. *unsupervised learning*). Sieci neuronowe nie mają przedstawianych wzorców, których mają się nauczyć, lecz same organizują przedstawiane im wektory danych.

6.1 Zasada Hebba i siła powiązań między neuronami

W neuro-fizjologii zauważono, że siła (waga) powiązań między dwoma neuronami wzrasta przy jednoczesnym pobudzeniu obu neuronów, w przeciwnym przypadku maleje. Zaobserwowano również, że im częściej jakiś bodziec dochodzi do neuronu, tym silniejsza jest odpowiedź neuronu na ten bodziec. W konsekwencji najczęściej powtarzające się mają największy wpływ na samo-adaptację wag.

W sieciach neuronowych neurony 'uczą się' rozpoznawać bodźce (w naszym rozumieniu: dane podawane na wejście sieci), czyli dane. Zmiany aktualnych wag powinny iść w kierunku dostosowywania wag do najczęściej przedstawianych wzorców, jakimi są wektory danych. Sygnały napływające do neuronu noszą również nazwę sygnałów *presynaptycznych*. Sygnały wytwarzane przez neuron to sygnały *postsynaptyczne*.

Podczas procesu uczenia, w miarę napływających wzorców (wektorów danych w naszym rozumieniu) sieć 'uczy się' tych danych, czyli adaptuje sekwencyjnie swoje wagi – aby umieć prawidłowo rozpoznawać te dane. Adaptacja ta jest iteracyjna, tzn. po przedstawieniu k -tego wzorca $\mathbf{x}(k)$ wytworzone już wagi \mathbf{w} zmieniają się według ogólnej zasady

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \Delta\mathbf{w}(k). \quad (6.1)$$

Jak wyrazić zmianę wektora wag, czyli wielkość $\Delta\mathbf{w}(k)$?

Zagadnienie to było przedmiotem intensywnych dyskusji, w rezultacie których pojawiło się kilka propozycji, które przetrwały do dzisiaj, a nawet są dalej rozwijane [1].

W latach 1949 ukazała się książka Hebba dyskutująca to zagadnienie¹. Z książki tej pochodzą dwie zasady, znane dzisiaj pod nazwą ogólnej i prostej reguły Hebba.

Ogólna reguła Hebba mówi, że zmiany wag powinny odbywać się według reguły

$$\Delta\mathbf{w}(k) = F(\mathbf{x}(k), y(k)),$$

czyli ogólnie, że przyrost wag $\Delta\mathbf{w}(k)$ powinien zależeć zarówno od wielkości wzorca presynaptycznego $\mathbf{x}(k)$ jak i od wytworzonego wzorca postsynaptycznego $y(k)$.

Jak wyrazić funkcję F ? Jedną z możliwości podaje tzw. prosta reguła Hebba. Upraszcza ona funkcję F do funkcji iloczynowej:

¹D.O. Hebb, *The organization of behavior: a neuropsychological theory*. Wiley, New York, 1949

Prosta reguła Hebba: (stała η oznacza tu współczynnik proporcjonalności):

$$\Delta \mathbf{w}(k) = \eta \cdot \mathbf{x}(k) \cdot y(k).$$

Jednak ta reguła była zbyt uproszczona. Przy wielokrotnej prezentacji tego samego wzorca wagi wzrastały wykładniczo. Po każdej prezentacji wzorca $\mathbf{x}(k)$ wektor wag był przesuwany w kierunku tego wzorca, mogło to powodować raptowne i dość znaczne zmiany wyuczonego dotychczas wektora wag.

Reguła Sejnowskiego² proponuje, że zmiana wag ma być proporcjonalna do wektora kowariancji sygnałów presynaptycznych \mathbf{x} z sygnałem postsynaptycznym y .

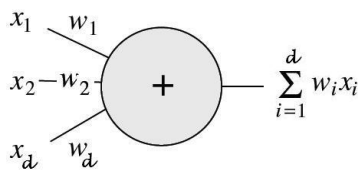
Reguła zapominania. Przy użyciu funkcji iloczynowej użytej w prostej regule Hebba wagi wzrastają wykładniczo z postępowaniem uczenia i wielokrotną prezentacją tego samego wzorca, co jest zjawiskiem bardzo niepożądanym. Aby temu zapobiec, wprowadzono reguły ograniczające przyrosty wag. Jedną taką modyfikacją jest reguła zapominania [6], str 27.

Reguła Oji. Inną ważną modyfikację wprowadził Oja³, który zaproponował, aby prostą regułą Hebba stosować na sygnale pre-synaptycznym skorygowanym wg następującego wzoru {Oja}:

$$\Delta \mathbf{w}(k) = \eta y(k) [\mathbf{x}^T(k) - y(k) \mathbf{w}(k)].$$
 (6.2)

Tutaj wpływ sygnału presynaptycznego \mathbf{x} jest zmniejszany (proporcjonalnie) o wielkość rzutu wektora \mathbf{x} na wektor \mathbf{w} .⁴ Ponadto, ze wzoru (6.2) wynika, że w kolejnych krokach uczenia wagi ulegają zmniejszeniu proporcjonalnie do kwadratu wytwarzanego sygnału postsynaptycznego y .

Generalnie rzecz biorąc, chodzi o zbudowanie tzw. liniowego asocjatora pokazanego na rys. 6.1. Należy wyznaczyć wektor wag w ten sposób, aby sygnał wyjściowy y reprezentował wszystkie sygnały wejściowe (x_1, \dots, x_d) , przy czym reprezentacja ta wyraża się jako liniowa kombinacja zmiennych wejściowych



Rysunek 6.1: Liniowy asocjator, patrz Rojas [7], Fig. 6.5.0: Linear associator.

²Terence Sejnowski, praca w J. Math. Biology, 1977, cytowane za [6], str. 27

³E. Oja, Simplified neuron model as a principal component analyser, J. Math. Biology, 1982, Vol. 15, s. 267–273, cytowane za [6]

⁴Zauważmy, że

$$y = \langle \vec{\mathbf{x}}, \vec{\mathbf{w}} \rangle = \|\vec{\mathbf{w}}\| \underbrace{\|\vec{\mathbf{x}}\| \cos(\alpha)}_{x_r} \propto x_r,$$

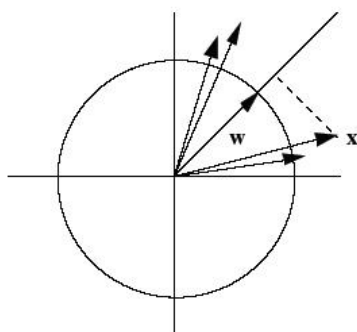
gdzie α jest kątem między wektorami $\vec{\mathbf{x}}$ i $\vec{\mathbf{w}}$, a $x_r = \|\vec{\mathbf{x}}\| \cos(\alpha)$ jest rzutem wektora $\vec{\mathbf{x}}$ na wektor $\vec{\mathbf{w}}$

Pseudokod obliczania pierwszej składowej głównej (Rojas, Alg. 5.3.1, str 116)

start Let $X_{n \times d}$ be the data matrix
 Initialize randomly \mathbf{w} , the weight vector, $\mathbf{w} \neq \mathbf{0}$
 Set $t:=0$; % iteration no.
 Define a declining function $\eta(t)$, ($0 \leq \eta(t) \leq 1$)

update $t:=t+1$;
 Select randomly a row vector \mathbf{x} from X
 Compute the scalar product $\Phi = \mathbf{x} \cdot \mathbf{w}$
 Update $\mathbf{w} := \mathbf{w} + \eta(t) \Phi (\mathbf{x} - \Phi \mathbf{w})$
 If not STOPcondition, go to update, end

Jeżeli dane są wielowymiarowe, to po obliczeniu pierwszej składowej głównej \mathbf{w}_1 możemy od każdego wektora uczącego \mathbf{x}_i odjąć jego rzut na \mathbf{w}_1 , a następnie – na powstałych w ten sposób danych – szukać następnego kierunku \mathbf{w}_2 . Drugi szukany wektor \mathbf{w}_2 powinien być ortogonalny do \mathbf{w}_1 . Przy używaniu jednej składowej punkt z R^d jest reprezentowany jedną liczbą, mianowicie długością jego rzutu na wektor \mathbf{w} .



Rysunek 6.2: Uczenie neuronu w celu wyznaczenia 1-ej składowej głównej. Rysunek hebb2. Autor rysunku: Rojas [7], Fig. 5.17: Cluster of vectors and principal component

Oja pokazał, że przy uczeniu określonym wzorem (6.2) wektor \mathbf{w} w trakcie uczenia stabilizuje się i dąży do wartości jednostkowej (por. Dodatek na końcu tego rozdziału).

Oja pokazał również, że wektor wagowy \mathbf{w} w granicy uczenia spełnia następujące równanie macierzowe {sekularne}

$$\mathbf{C}\mathbf{w} = \lambda\mathbf{w}, \quad \text{gdzie } \lambda = \mathbf{w}^T \mathbf{C}\mathbf{w}. \quad (6.3)$$

W innych słowach można powiedzieć, że wektor \mathbf{w} jest wektorem własnym macierzy kowariancji \mathbf{C} . Generalnie, równanie sekularne ma d rozwiązań, gdzie d jest rzędem macierzy \mathbf{C} .

Jeżeli sieć neuronowa ma d neuronów, a i -temu neuronowi jest przypisany wektor wag \mathbf{w}_i , a ponadto na występujące wektory wagowe nałożymy warunek ortogonalności, to wszystkie wektory wagowe \mathbf{w}_i po ustabilizowaniu się procesu uczenia powinny spełniać równanie 6.3, ale z różnymi wartościami własnymi. Jeżeli sieć ma tylko $H < d$ neuronów,

to kolejne kolejne wektory wag są wektorami własnymi macierzy \mathbf{C} odpowiadającymi H największym wartościom własnym tej macierzy.

Oja, a później Sanger, podali efektywne wzory, jak znajdować metodą sekwencyjną kolejne wektory $\mathbf{w}_1, \mathbf{w}_2, \dots$, itd. Umożliwiło to skonstruowanie odpowiednich sieci neuronowych które potrafią same – na podstawie prezentacji danych – znaleźć te kierunki.

Jednak, nauka taka trwa długo, i bardzo często łatwiej jest znaleźć kierunki główne metodami algebry numerycznej, o których piszemy w następnym podrozdziale.

6.2 Metoda PCA – podejście statystyczne

Metoda składowych głównych jest jedną z bardziej rozpowszechnionych metod analizy danych wielowymiarowych. Służy ona przede wszystkim: a) redukcji wymiarowości danych (ze wzgl"du na liczbę cech); b) wizualizacji rozważanych danych na płaszczyźnie, c) odsumianiu obserwowanych danych. Dobry i zrozumiały opis tej metody znajduje się w książkach Morrisona [5] oraz Johnsona i Wicherna [4]; istnieje również świetna monografia Iana Jolliffe'a [3].

W środowisku statystycznym metoda składowych głównych jest wiązana z nazwiskiem Hotellinga⁵; w środowisku technicznym (przetwarzanie sygnałów, sieci neuronowe) metoda ta jest określana mianem transformaty Karhunen–Loevego.

Przedstawiana dalej metoda obliczeniowa wyznacza składowe główne z kwadratowej macierzy kowariancji, korelacji lub iloczynów skalarnych rozmiaru $d \times d$ (d - liczba składowych wektora danych charakteryzującego tzw. "osobnika" w rozumieniu statystycznym). Metoda ta wynika z następującego zagadnienia optymalizacyjnego: Dla danej **scenowanej** tablicy danych $\mathbf{X} = (x_{ij})$, $i = 1, \dots, N$; $j = 1, \dots, d$ wyznaczyć kombinację liniową $\mathbf{a} = [a_1, \dots, a_p]^T$, $\mathbf{a}^T \mathbf{a} = 1$, transformującą tablicę \mathbf{X} do postaci {transf}

$$\mathbf{y} = \mathbf{X} \mathbf{a} \quad (6.4)$$

tak, żeby utworzony nowy wektor miał maksymalną wariancję, czyli żeby kryterium K

$$K(\mathbf{a}) = \sum_{n=1}^N (y_n)^2 = \mathbf{y}^T \mathbf{y} = (\mathbf{X}\mathbf{a})^T (\mathbf{X}\mathbf{a}) \quad (6.5)$$

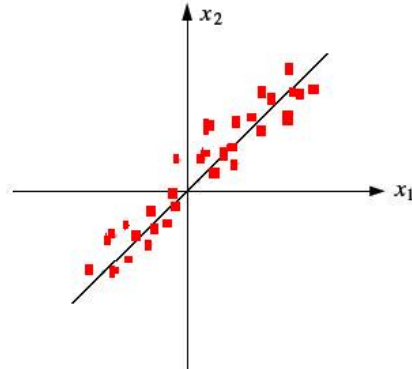
przyjmowało wartość maksymalną⁶.

Pokazuje się (patrz Dodatek 2), że zagadnienie to ma d rozwiązań. Są nimi wektory \mathbf{a} spełniające równanie macierzowe (6.6).

Rysunek 6.3 ilustruje znajdowanie pierwszej składowej głównej dla dwóch o dwóch zmiennych wejściowych x_1 i x_2 . Chcemy otrzymać nową zmienną która będzie możliwie dobrze reprezentować dla danego punktu obydwie jego składowe. Takie wartości możemy otrzymać przez rotację układu współrzędnych (x_1, x_2) .

⁵H. Hotelling, Simplified calculation of principal components. Psychometrika Vol. 1., 1936, s. 1–27. Cytowane za [5].

⁶Kryterium $K(\mathbf{a})$ jest proporcjonalne do wariancji wektora \mathbf{y} , wynika to ze scenowania macierzy \mathbf{X} ; każda wartość y_i stanowi rzut punktu \mathbf{x}_i na jednostkowy wektor \mathbf{a}



Rysunek 6.3: Kierunek pierwszej składowej głównej – rysunek hebb3a. Autor rysunku: Rojas [7], Fig. 5.16: Distribution of input data

Przypuśćmy, że mamy daną tablicę \mathbf{X} o wymiarach $[N \times d]$. Wektory-wiersze tej tablicy mogą być interpretowane jako punkty $\in R^d$. Przyjmijmy – dla wprowadzającej ilustracji – że $d = 2$. Wtedy wektory-wiersze tablicy \mathbf{X} można zobrazować na płaszczyźnie $\langle x_1, x_2 \rangle$. Mówimy, że wykreślone punkty tworzą *chmurę punktów indywidualnych*. Na ogół, dla danych statystycznych spotykanych w praktyce, tzn. *nie* wygenerowanych za pomocą specjalnych modeli, chmura punktów indywidualnych w przypadku $d = 2$ przypomina elipsę, w przypadku $d = 3$ elipsoidę, a w przypadku $d > 3$ hiper-elipsoidę.

Kierunki osi głównych tej elipsy lub (hiper) elipsoidy wyznacza się z równania macierzewego $\{\text{eigS}\}$

$$(\mathbf{S}_X - \lambda \mathbf{I})\mathbf{a} = \mathbf{0}, \quad \text{lub} \quad \mathbf{S}_X \mathbf{a} = \lambda \mathbf{a}, \quad (6.6)$$

gdzie $\mathbf{a} = [a_1, \dots, a_d]^T$ jest jednym z kierunków głównych, a \mathbf{S}_X jest macierzą kowariancji danych zapisanych w tablicy \mathbf{X} . Dla jednoznaczności rozwiązania równania (6.6) przyjmujemy, że wektor \mathbf{a} jest wektorem o długości 1, co oznacza że $\mathbf{a}^T \mathbf{a} = 1$.

Wektory \mathbf{a} spełniające równanie (6.6) i wyznaczające kierunki główne są najczęściej znajduwane sekwencyjnie (por. Morrison [5]).

Najpierw szuka się wektora \mathbf{a}_1 wyznaczającego najdłuższą oś elipsy lub (hiper) elipsoidy. Wektor \mathbf{a}_1 musi spełniać równanie macierzowe (6.6). Okazuje się, że równanie to ma d rozwiązań, odpowiadających d wartościom własnym macierzy \mathbf{S} . Tak więc, rozwiązując równanie (6.6) otrzymujemy d wartości własnych. Porządkując otrzymane wartości własne w ciąg nierosnący otrzymujemy

$$\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_d \geq 0.$$

Z każdą z tych wartości własnych jest związany odpowiadający jej wektor własny

$$\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_d.$$

Można wykazać, że kierunek najdłuższej osi elipsoidy jest identyczny z kierunkiem wyznaczanym przez wektor \mathbf{a}_1 , a wariancja rzutów punktów indywidualnych na kierunek wyznaczony przez \mathbf{a}_1 wynosi $s_{y[1]}^2 = \lambda_1 = \mathbf{a}_1^T \mathbf{S}_X \mathbf{a}_1$. Dalsze osie elipsoidy są wyznaczone przez dalsze wektory własne po wprowadzeniu dodatkowych warunków ubocznych $\mathbf{a}_i^T \mathbf{S}_X \mathbf{a}_j = 0$, $i \neq j$.

Wszystkie wektory własne możemy zestawić w macierz \mathbf{A} :

$$\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_d].$$

Każdy z wektorów własnych \mathbf{a}_i , $i = 1, \dots, d$ spełnia równanie macierzowe (6.6). Wobec tego {eigS1}

$$\mathbf{S}_X \mathbf{A} = \mathbf{A} \mathbf{\Lambda}, \quad \text{gdzie } \mathbf{\Lambda} = \text{diag}(\lambda_1, \dots, \lambda_d). \quad (6.7)$$

Można wykazać, że macierz \mathbf{A} wyznacza transformację będącą rotacją układu współrzędnych zmiennych $\langle x_1, \dots, x_d \rangle$ (zaczepionego w punkcie zero) do układu współrzędnych wyznaczonych przez wektory $\langle \mathbf{a}_1, \dots, \mathbf{a}_d \rangle$ zaczepionych również w punkcie zero i wyznaczających osie główne elipsoidy.

Przypomnijmy, że \mathbf{X} jest tablicą scentrowaną na $\mathbf{0}$ oraz że macierz kowariancji zmiennych \mathbf{X} wynosi \mathbf{S} : $\text{Var}(\mathbf{X}) = \mathbf{S}$.

Rotacja określona macierzą \mathbf{A} może być uważana jako liniowa transformacja macierzy \mathbf{X} . Po wykonaniu tej rotacji otrzymujemy {tabY}:

$$\mathbf{Y} = \mathbf{X} \mathbf{A}. \quad (6.8)$$

Tablica \mathbf{Y} , jako powstała ze scentrowanej tablicy \mathbf{X} , jest również scentrowana na zero. Mamy: $\mathbf{Y} = [\mathbf{y}_{[1]}, \dots, \mathbf{y}_{[d]}]$, gdzie $\mathbf{y}_{[j]}$ oznacza j -tą kolumnę tablicy \mathbf{Y} . Tak więc

$$\mathbf{Y} = [\mathbf{y}_{[1]}, \dots, \mathbf{y}_{[d]}] = \mathbf{X} * [\mathbf{a}_1, \dots, \mathbf{a}_d].$$

Otrzymana w ten sposób tablica \mathbf{Y} przedstawia te same dane co tablica \mathbf{X} , ale wyrażone w innym układzie współrzędnych. Ponieważ nowe zmienne zostały otrzymane w wyniku liniowej transformacji zmiennych \mathbf{X} , wariancja nowych zmiennych ($\text{Var}(\mathbf{Y})$) otrzymuje się z macierzy kowariancji \mathbf{S}_X jako formę kwadratową wyznaczoną przez macierz starych zmiennych: $\text{Var}(\mathbf{Y}) = \mathbf{A}^T \mathbf{S}_X \mathbf{A}$.

Powtórzmy jeszcze raz: wzór (6.8) określa **nowe zmienne**: Każda kolumna tablicy \mathbf{Y} przedstawia kombinację liniową 'starych' zmiennych (identyfikowanych kolumnami tablicy \mathbf{X}). W niektórych przypadkach, uwzględniając wartości elementów a_{ij} , $j = 1, \dots, d$ tworzących nową zmienną $\mathbf{y}_{[i]}$, możemy przypisać 'nowej' zmiennej szczególne znaczenie (np. ogólna wielkość osobnika, jego sylwetka, itp., patrz przykłady w [5]).

Definicja. Elementy tablicy \mathbf{Y} określonej wzorem (6.8) są nazywane **współrzędnymi głównymi** lub **składowymi głównymi** tablicy danych \mathbf{X} (terminy angielskie: *principal coordinates* i *principal components*).

Zauważmy, że każda współrzędna główna (składowa główna) jest otrzymywana jako kombinacja liniowa *wszystkich* zmiennych wyjściowych ($\mathbf{y}_{[j]}$ oznacza j -tą kolumnę tablicy \mathbf{Y}) {newCoord}:

$$\mathbf{y}_{[j]} = \mathbf{X}\mathbf{a}_j. \quad (6.9)$$

Macierz kowariancyjna 'nowych' danych \mathbf{Y} równa się (korzystamy z wzoru (6.7)):

$$\text{Var}(\mathbf{Y}) = (1/n)\mathbf{Y}^T\mathbf{Y} = (1/n)(\mathbf{A}^T\mathbf{X}^T\mathbf{X}\mathbf{A}) = \mathbf{A}^T \underbrace{\mathbf{S}_X\mathbf{A}} = \mathbf{A}^T\mathbf{A}\mathbf{\Lambda} = \mathbf{\Lambda}.$$

Tak więc: $\{\mathbf{S}_Y\}$

$$\mathbf{S}_Y = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & \lambda_p \end{bmatrix}. \quad (6.10)$$

Współrzędne główne (przetransformowane dane) mają następujące **własności**:

1. Wariancje wartości zapamiętanych w kolejnych kolumnach tablicy \mathbf{Y} wynoszą odpowiednio $\lambda_1, \lambda_2, \dots, \lambda_p$,
2. Pary kolumn tablicy \mathbf{Y} przedstawiają wartości zmiennych nieskorelowanych (albo-
wiem elementy $s_{ij}(Y)$ dla $i \neq j$ są równe zero).
3. Zachowanie śladów macierzy \mathbf{X} i \mathbf{Y} . Suma wariancji wartości 'nowych zmiennych' powstałych jako transformacja $\mathbf{Y}=\mathbf{X}\mathbf{A}$ równa się sumie wariancji 'starych zmiennych' danych w tablicy \mathbf{X} : danych współrzędnych głównych $\mathbf{y}_{[1]}, \dots, \mathbf{y}_{[d]}$ zapisanych w tablicy \mathbf{Y} równa się sumie wariancji zmiennych oryginalnych zapisanych w tablicy \mathbf{X} . Tak więc:

$$\begin{aligned} \text{trace}(\mathbf{S}_Y) &= \text{trace}((1/n)\mathbf{Y}^T\mathbf{Y}) = \text{trace}((1/n)\mathbf{X}\mathbf{A}^T\mathbf{X}\mathbf{A}) \\ &= \text{trace}((1/n)\mathbf{A}^T\mathbf{X}^T\mathbf{X}\mathbf{A}) = \text{trace}(\mathbf{A}^T\mathbf{S}_X\mathbf{A})\text{trace}(\mathbf{S}_X\mathbf{A}\mathbf{A}^T) = \text{trace}(\mathbf{S}_X). \end{aligned}$$

Stąd suma wariancji zmiennych oryginalnych (X) jest równa sumie wariancji nowych zmiennych (Y) $\{\text{trace}\}$

$$\text{trace}(\mathbf{S}_X) = \sum_{i=1}^d \underbrace{s_{ii}}_{\text{elementy przek.}} = \sum_{i=1}^d \underbrace{s_i^2}_{\text{wariancje } x_1, \dots, x_d} = \sum_{i=1}^d \lambda_i = \text{trace}(\mathbf{S}_Y). \quad (6.11)$$

Jeżeli suma dwóch lub trzech pierwszych wartości własnych jest duża w stosunku do pozostałych (np. stanowi 90% lub więcej całkowitej ich sumy), to wynika stąd że wariancje wyjściowych zmiennych X_1, \dots, X_p są odtwarzane w znacznym stopniu przez nowe (przetransformowane) zmienne, czyli przez współrzędne główne. Można stąd dedukować, że elipsoida w R^d obejmująca dane \mathbf{X} jest faktycznie mocno spłaszczona, wobec czego jest możliwe i uzasadnione posługiwanie się reprezentacją tych danych w niższej przestrzeni R^r wyznaczonej przez r pierwszych składowych głównych.

4. Twierdzenie o odtwarzaniu macierzy kowariancji za pomocą wartości własnych i wektorów własnych

Macierz kowariancji \mathbf{S} może być odtworzona z jej wartości własnych i wektorów własnych za pomocą wzoru: {odtwarzS} ⁷

$$\mathbf{S} = \sum_{j=1}^d \lambda_j \mathbf{a}_j \mathbf{a}_j^T. \quad (6.12)$$

W szczególności możemy skoncentrować się na odtwarzaniu przekątnej. Dla przyjętego q ($1 \leq q < d$) możemy wtedy ustalić, jak dobrze poszczególne zmienne są określane (wyjaśniane, ang. *reproduced*) przez q pierwszych składowych.

5. Twierdzenie o odtwarzaniu tablicy danych za pomocą składowych głównych i wektorów własnych ⁸

$$\mathbf{X} = \mathbf{Y}\mathbf{A}^T = \sum_{j=1}^d \mathbf{y}_{[j]} \mathbf{a}_j^T. \quad (6.13)$$

Wskazówki praktyczne i podsumowanie

Przy obliczaniu współrzędnych głównych (składowych głównych) należy obejrzeć koniecznie wartości własne odpowiadające tym składowym.

Tylko wtedy, gdy udział pierwszych wartości własnych w stosunku do sumy wszystkich wartości własnych jest duży, mamy teoretyczne podstawy (i uzasadnienie, jesteśmy uprawnieni) do zastąpienia tablicy danych \mathbf{X} o d kolumnach zredukowaną macierzą $\mathbf{Y}^{(r)}$ o r kolumnach ($r < d$).

Obliczenia składowych głównych są zrealizowane w wielu pakietach poświęconych sieciom neuronowym. Są one również zrealizowane – jako algorytmy algebry numerycznej – w pakietach Netlab i Somtoolbox (ten ostatni jest poświęcony sieciom Kohonena).

W dalszym ciągu omówimy algorytmy obliczania PCA zaimplementowane w pakietach Netlab i Somtoolbox [8].

⁷Dla macierzy \mathbf{S} pełnego rzędu mamy: $\mathbf{S}\mathbf{A} = \mathbf{A}\mathbf{\Lambda} \Leftrightarrow \mathbf{S}\mathbf{A}\mathbf{A}^T = \mathbf{A}\mathbf{\Lambda}\mathbf{A}^T$.

Stosując twierdzenie o mnożeniu macierzy kolumnowej przez macierz wierszową otrzymujemy wzór szukany wzór (6.12).

⁸Wzór (6.13) wynika z definicji (6.8) przyjmującej, że $\mathbf{Y} = \mathbf{X}\mathbf{A}$. Mnożąc tę równość prawostronnie przez $\mathbf{A}^{-1} = \mathbf{A}^T$ otrzymujemy $\mathbf{X} = \mathbf{Y}\mathbf{A}^T$. Teraz należy zastosować twierdzenie o mnożeniu macierzy kolumnowej przez macierz wierszową.

6.3 Obliczanie PCA za pomocą funkcji Matlaba

6.3.1 Obliczanie składowych głównych za pomocą funkcji 'eig' Matlaba

Wszystkie podane metody obliczają (jawnie lub niejawnie) wartości własne z kwadratowej macierzy korelacji, kowariancji lub iloczynów skalarnych rozmiaru $d \times d$, obliczone z tablicy danych X . Macierz wektorów własnych (A) oraz wektor kolumnowy (λ) zawierający wartości własne uporządkowane malejąco (nie-rosnąco) i dla macierzy kowariancji lub korelacji tablicy danych możemy otrzymać bezpośrednio z funkcji `eig` podstawowego Matlaba za pomocą następującego kodu:

```
[A,lam]=eig(cov(X)); % [A,lam]=eig(corrcoef(X));
lam=diag(lam); % wektor kolumnowy, wartosci własne rosnące
lam=lam1(end:-1:1); A=A(:,end:-1:1); % posortowane malejaco
```

Niech \tilde{X} oznacza odpowiednio scentrowaną (gdy procedura `eig` była wywoływana z macierzą kowariancji) lub wystandaryzowaną statystycznie (gdy procedura `eig` była wywoływana z macierzą korelacji) tablicę danych X .

Wtedy składowe główne (Principal components), współrzędne rzutów punktów indywidualnych danych otrzymujemy ze wzoru: $Y = \tilde{X}A$.

W przypadku danych o bardzo dużej liczbie cech możemy ominąć obliczanie macierzy kowariancji/korelacji używając funkcji `svd` Matlaba.

6.3.2 Obliczanie PCA za pomocą pakietu NETLAB

Obliczanie wartości własnych i wektorów własnych z danych

`[evals, evecs] = pca(data, K)`; liczy K wartości i K wektorów własnych.

`evals = pca(data, K)`; Liczy tylko K wartości własnych.

`[evals, evecs] = pca(data)`; liczy wszystkie wartości i wektory własne.

`evals = pca(data)`; Liczy tylko wartości własne.

Liczba K nie może być większa niż liczba kolumn (cech) w danych: $K \leq \text{size}(\text{data}, 2)$

Funkcja `pca` w trakcie swego działania oblicza najpierw dla zadeklarowanych danych `data` macierz kowariancji S , dla której zostają obliczone wartości własne i wektory własne.

Tablica `evals` jest jednokolumnowa i zawiera wartości własne w uporządkowaniu od największej.

Tablica `evecs` zawiera w kolejnych kolumnach kolejne wektory własne. Wektory te są ortonormalne.

Obliczanie wartości własnych i wektorów własnych z macierzy kowariancji

`[evals, evecs] = eigdec(S, K)`; liczy K wartości i wektorów własnych macierzy kowariancji S .

Podobnie jak funkcja `pca`, również funkcja `eigdec` może mieć tylko jeden argument, tzn. macierz S , oraz może liczyć tylko wartości własne bez wektorów własnych.

Funkcja `cov` oblicza macierz kowariancji z danych.

Funkcja `corrcoef` oblicza macierz korelacji z danych.

Jeżeli dane są wystandaryzowane na jednostkowe wariancje, to obliczana macierz kowariancji jest jednocześnie macierzą korelacji.

Aby otrzymać rzuty (czyli współrzędne główne), wystarczy po obliczeniu wektorów własnych wykonać projekcję na odpowiednio scentrowanych danych:

`y = data*evecs;` – jeśli dane `data` są już scentrowane;

`mu = mean(data);`

`y = (data - ones(size(data,1),1)*mu)*evecs;` – jeśli dane nie są scentrowane.

Jeśli wykonywaliśmy obliczenia na wystandaryzowanych danych, to przy rzutowaniu dane też powinny być wystandaryzowane.

6.3.3 Obliczenia PCA za pomocą pakietu `somtoolbox`

Obliczenia składowych głównych odbywają się za pomocą funkcji `pcaproj`. Podstawowe wywołania tej funkcji:

`[P, V, mu, lam] = pcaproj(X, odim);` Funkcję tę wywołujemy z parametrami:

X – tablica danych wymiaru $N \times d$. Musimy sami zapewnić właściwą normalizację tej tablicy (np. normalizację na średnią=0);

`odim` – output dimension; *ile* składowych chcemy policzyć.

Wynikami funkcji mogą być:

`P`, tablica rzutów (Projection), ma wymiar $N \times odim$,

`V`, tablica wektorów własnych, ma wymiar $d \times odim$;

`mu`, wektor średnich tablicy X , ma wymiar $1 \times d$;

`lam`, tablica wymiaru $1 \times odim$, pokazuje udział kolejnych `lam` w odtwarzaniu całkowitej zmienności (wariancji) układu definiowanej jako $trace\{(\mathbf{X}^T \mathbf{X})/N\}$.

Obliczone wartości `V` i `mu` mogą służyć do zrzutowania innego zbioru danych (np. zbioru umieszczonego w tablicy `X1`) do poprzednio wyznaczonego układu składowych głównych. Możemy to zrobić przez wywołanie funkcji `pcaproj` z następującymi parametrami:

`P1 = pcaproj(X1, V, mu);`

Podkreślmy wyraźnie: Tablica `X1` musi mieć tę samą liczbę kolumn co tablica `X` i musi być wystandaryzowana według tych samych zasad, jakie były zastosowane do tablicy `X` (a dokładniej, do tej tablicy, z której obliczono operatory rzutowania znajdujące się w tablicy `V`).

Zamiast tablic `X` i `X1` możemy posłużyć się również tablicami danych znajdujących się w strukturach `sD` lub `sM` specyficznych dla pakietu `somtoolbox`. Struktury te zapewniają odpowiednio ujednoliconą standaryzację. Tworzenie tych struktur będzie omawiane w następnych rozdziałach.

Obliczenia procedury `pcaproj` przy pierwszym, standardowym wywołaniu przebiegają w następujących krokach:

Najpierw – dla zadeklarowanej tablicy \mathbf{X} – oblicza się wektor średnich μ oraz macierz kowariancji \mathbf{S}_X .

Następnie wyznacza się wszystkie wartości własne i wektory własne macierzy \mathbf{S}_X . Obliczone wartości własne – wraz z odpowiadającymi im wektorami własnymi – są porządkowane malejąco.

Wreszcie oblicza się – według wzoru (6.9) – żądanych *odim* współrzędnych głównych, które umieszczają się jako wynik w tablicy \mathbf{P} o wymiarach $[N \times \textit{odim}]$.

Oprócz tego w wynikach mogą być umieszczone:

- ◊ \mathbf{V} o wymiarach $[d \times \textit{odim}]$ – wektory własne użyte do projekcji (są to pierwsze kolumny macierzy \mathbf{A} w naszym opisie metody),
- ◊ μ o wymiarach $[1 \times d]$ – wektor średnich,
- ◊ \textit{lam} o wymiarach $[1 \times \textit{odim}]$ – frakcje $|\lambda_j| / \sum_j |\lambda_j|$, $j = 1, \dots, \textit{odim}$.

6.4 Przykłady zastosowań metody składowych głównych

Przedstawiona metoda jest nazywana analizą składowych głównych (PCA, Principal Component Analysis).

Generalnie, metoda PCA ma duże znaczenie przy redukcji wymiarowości: pozwala zastąpić dane o dużej liczbie cech znacznie mniejszą liczbą skonstruowanych nowych cech, bez istotnej utraty informacji. Wtedy np. wektory danych $\mathbf{x} = [x_1, \dots, x_d]^T$ o dużej liczbie wymiarów można wtedy zastąpić wektorami $\mathbf{z} = [z_1, \dots, z_k]^T$, ($k < d$) o znacznie mniejszej liczbie cech (na ogół).

W teorii sygnałów przekształcenie $\mathbf{x} \rightarrow \mathbf{z}$ jest nazywane transformacją Karhunen-Loevego.

Osowski (str 235) pokazuje przykład zastosowania tej techniki do zapamiętywania i odtwarzania obrazów; pokazuje m.in. obraz przedstawiający głowę mężczyzny, odtworzony za pomocą 2 pierwszych składowych głównych.

Nabney pokazuje dwa przykłady: Jeden, to badanie frakcji ropy w rurociągu za pomocą promieniowania gamma, drugi, to identyfikacja cyfry '5' z bazy MNIST (każdy wzorec zapamiętany jako tablica 8 bit gray level matrix of dimension 28×28).

Szczegółowy przykład obliczeń dla danych *iris*

Pokażemy obliczenia dla danych *iris*. Tablica danych ma tutaj wymiar 150×4 ; każdy wiersz zawiera $d = 4$ pomiary kwiatka irysa. Wiersze 1–50 odnoszą się do gatunku *Setosa* (Se), wiersze 51 do 100 – do gatunku *Versicolor* (Ve), a wiersze 101 do 150 – do gatunku *Virginica* (Vi).

Obliczenia wykonano na danych znormalizowanych na średnią = 0 i wariancję = 1. Oznacza to, że wartości własne i wektory własne wykonano na macierzy korelacji.

Frakcje odtworzonej sumy $\textit{Slam} = \sum_{i=1}^4 \lambda_i$ przez kolejne składowe wynoszą odpowiednio

λ_1/\textit{Slam}	λ_2/\textit{Slam}	λ_3/\textit{Slam}	znaczenie
0.7272	0.2303	0,0368	udziały kolejnych λ , skumulowane sumy tych udziałów.
0.7272	0.9575	0.9943	

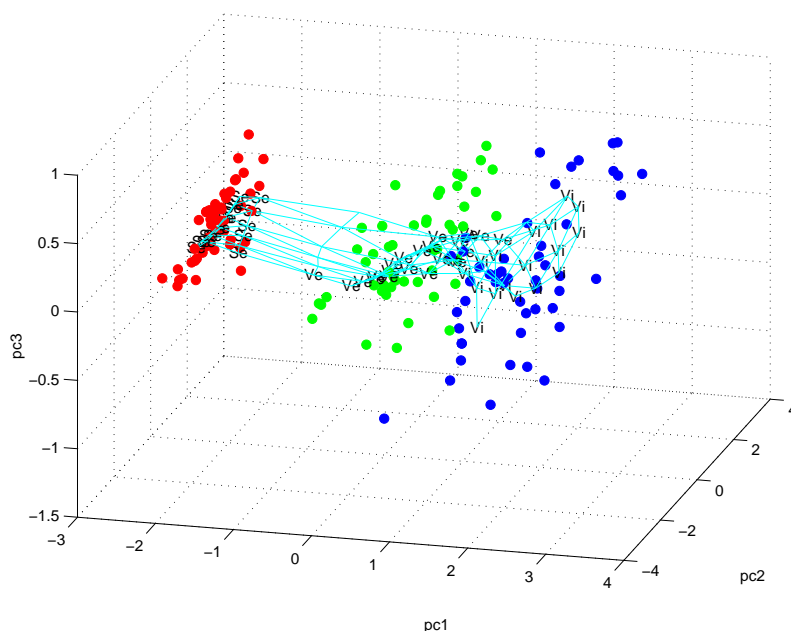
Jak widzimy, *dwie* pierwsze składowe główne wyrażają przeszło 95% całkowitej zmienności całego układu punktów.

Natomiast *trzy* pierwsze składowe odtwarzają już 99% całkowitej zmienności.

Oznacza to, że praktycznie wszystkie związki między rozważanymi wektorami danych możemy pokazać na płaszczyźnie lub za pomocą zobrazowania trójwymiarowego wykonywanego za pomocą funkcji `plot3`.

Na wykresach tych możemy punkty odpowiadające różnym gatunkom irysa zaznaczyć różnymi kolorami. Oglądając wykresy możemy przekonać się naocznie, czy zbiory danych zawierające irysy z różnych gatunków są rozłączne, czy też nachodzą częściowo na siebie.

Wykres taki jest pokazany na rysunku 5.2. Punkty-kwiaty są tam zaznaczone wypełnionymi okręgami o różnych kolorach. Kolor czerwony oznacza irysy Setosa (Se), zielony – Versicolor (Ve), a niebieski – Virginica (Vi).



Rysunek 6.4: Dane iris: ● – *iris Setosa*, ● – *iris Versicolor*, ● – *iris Virginica*, oraz prototypy znalezione techniką SOM Kohonena (punkty połączone siatką) zrzutowane do układu trzech pierwszych składowych głównych $pca1$, $pca2$, $pca3$. Obliczenia wykonano na znormalizowanych danych; wykres został sporządzony za pomocą funkcji `som_grid` pakietu `somtoolbox {iris.eps}`

Literatura

- [1] C. Fyfe, (2005) *Hebbian learning and negative feedback*. Springer, Berlin-Heidelberg.
- [2] J. Hertz, A. Krogh, R.G. Palmer, *Wstęp do teorii obliczeń neuronowych*. Tłum. z ang., wyd. II, WNT W-wa 1993.
- [3] I.T. Jolliffe, *Principal Component Analysis*, Springer, New York 2002. 2nd Edition.
- [4] R.A. Johnson, D.W. Wichern (1998). *Applied Multivariate Statistical Analysis*. Prentice Hall Inc., Englewood Cliffs, N.J. (4-th Edition).
- [5] D.F. Morrison (1990). *Wielowymiarowa analiza danych statystycznych*. PWN Warszawa 1990. Tłum. z ang. *Multivariate Statistical Methods*, 2nd ed., McGraw-Hill 1981.
- [6] S. Osowski, *Sieci neuronowe w ujęciu algorytmicznym*. WNT W-wa 1996.
- [7] Raul Rojas, *Neural Networks – A systematic Introduction*. Springer 1996.
- [8] J. Vesanto, J. Himberg, E. Alhoniemi, J. Parhankangas, *SOM Toolbox for Matlab 5*. Som Toolbox team, Helsinki University of Technology, Finland, Libella Oy, Espoo 2000, 1–54. <http://www.cis.hut.fi/projects/somtoolbox/>

Dodatek 1. Reguła Oji – interpretacja wag

Reguła Oji, przedstawiona wzorem [6.3] mówi, że po zaprezentowaniu sieci dużej liczby wzorców \mathbf{x} pochodzących z tej samej populacji \mathcal{P} wagi sieci powinny ustabilizować się w sensie statystycznym i wskazywać kierunki odpowiadające wektorom własnym macierzy kowariancji wyznaczonej dla zmiennej losowej \mathbf{x} charakteryzującej populację \mathcal{P} .

Przedstawiany szkic dowodu jest oparty na książkach: Hertz i in. [2], i Osowski [6].

Po wielokrotnej (powtórzonej kilkaset lub kilka tysięcy razy) prezentacji próbek danych wartości *średnie* lub też *wartości oczekiwane* przyrostów wag powinny wynosić zero. Oznacza to, że po ustaleniu punktu równowagi *wartości oczekiwane* \mathcal{E}_x tych przyrostów⁹ powinny być równe zeru:

$$\mathcal{E}_x\{\Delta \mathbf{w}_i(k)\} = \mathcal{E}_x\{\eta \cdot y_i(k)[\mathbf{x}(k) - y_i(k)\mathbf{w}_i(k)]\} = \mathbf{0},$$

gdzie $\mathbf{x}(k)$ jest wektorem kolumnowym, oznaczającym jeden wiersz tablicy danych \mathbf{X} .

Opuszczając współczynnik η oraz wskaźniki (k) pokazujące zależności poszczególnych wielkości od numeru iteracji k , mamy:

$$\mathcal{E}_x\{\mathbf{x}y_i\} - \mathcal{E}_x\{y_i y_i \mathbf{w}_i\} = \mathbf{0}.$$

Pamiętając, że $y_i = \mathbf{w}_i^T \mathbf{x} = \mathbf{x}^T \mathbf{w}_i$, oraz że operator \mathcal{E}_x odnosi się do zmiennych \mathbf{x} , otrzymujemy:

$$\mathcal{E}_x\{\mathbf{x}\mathbf{x}^T \mathbf{w}_i\} - \mathcal{E}_x\{\underbrace{\mathbf{w}_i^T \mathbf{x}^T \mathbf{x} \mathbf{w}_i}_{\lambda_i} \mathbf{w}_i\} = \mathbf{0}.$$

⁹wartość oczekiwana jest tu liczona względem rozkładu próbek \mathbf{x}

Oznaczmy¹⁰

$$\mathcal{E}_x\{\mathbf{x}^T\mathbf{x}\} = \mathbf{C}, \quad \text{oraz} \quad \mathbf{w}_i^T \mathbf{C} \mathbf{w}_i = \lambda_i. \quad (6.14)$$

Po wykonaniu podstawienia określającego macierz \mathbf{C} i skalar λ_i otrzymujemy:

$$\mathbf{C} \mathbf{w}_i = \lambda_i \mathbf{w}_i, \quad \text{lub inaczej} \quad (\mathbf{C} - \lambda_i \mathbf{I}) \mathbf{w}_i = \mathbf{0}. \quad (6.15)$$

Jak widać, wektor \mathbf{w}_i w stanie równowagi przedstawia sobą wektor własny macierzy \mathbf{C} związany z wartością własną λ_i tej macierzy. Można pokazać, że \mathbf{w}_i jest wektorem o długości 1: mnożąc (6.15) lewostronnie przez \mathbf{w}_i^T i korzystając z tożsamości $\mathbf{w}_i^T \mathbf{C} \mathbf{w}_i = \lambda_i$ (wzór 6.14) otrzymujemy: $\mathbf{w}_i^T \mathbf{C} \mathbf{w}_i = \lambda_i \mathbf{w}_i^T \mathbf{w}_i$, skąd wynika $\lambda_i = \lambda_i \|\mathbf{w}_i\|^2$, co jest prawdziwe tylko wtedy, gdy $\|\mathbf{w}_i\| = 1$.

Dodatek 2. PCA – Rachunek algebraiczny

Mamy tablicę danych $\mathbf{X}_{N \times d}$. **Tablica \mathbf{X} jest scentrowana na $\mathbf{0}$.** Chcemy utworzyć nową kolumnę danych, $\mathbf{y} = \mathbf{y}_{N \times 1}$, będącą kombinacją liniową kolumn tablicy \mathbf{X} i taką, żeby miała maksymalną wariancję:

$$\mathbf{y} = \mathbf{X} \mathbf{a},$$

$$s_y^2 = (1/n) \mathbf{y}^T \mathbf{y} = \max_{\mathbf{a} \in R^d, \|\mathbf{a}\|=1} s^2(\mathbf{a})$$

Wariancja nowo utworzonej zmiennej zapisuje się jako

$$s_y^2 = (1/n) (\mathbf{X} \mathbf{a})^T \mathbf{X} \mathbf{a} = (1/n) \mathbf{a}^T \underbrace{\mathbf{X}^T \mathbf{X}}_{\mathbf{S}} \mathbf{a} = \mathbf{a}^T \mathbf{S} \mathbf{a},$$

ponieważ $(1/n) \mathbf{X}^T \mathbf{X} = \mathbf{S}$ jest macierzą kowariancji rozpatrywanych danych.

Tym samym nasze zagadnienie sprowadza się do maksymalizacji formy kwadratowej utworzonej z wektora \mathbf{a} – za pomocą macierzy \mathbf{S} .

$$\max_{\mathbf{a} \in R^d, \|\mathbf{a}\|=1} s^2(\mathbf{a}) = \max_{\mathbf{a} \in R^d, \|\mathbf{a}\|=1} \mathbf{a}^T \mathbf{S} \mathbf{a}.$$

Uwzględniając warunki uboczne i metodę mnożników Lagrange'a należy maximizować funkcję

$$F(\mathbf{a}) = \mathbf{a}^T \mathbf{S} \mathbf{a} + \lambda(1 - \mathbf{a}^T \mathbf{a})$$

Obliczając pochodne cząstkowe względem \mathbf{a} otrzymujemy¹¹:

$$2\mathbf{S} \mathbf{a} - 2\lambda \mathbf{a} = \mathbf{0}$$

skąd

¹⁰W zależności od przeskalowania (normalizacji) danych macierz \mathbf{C} jest nazywana macierzą iloczynów skalarnych (iloczynów mieszanych), macierzą kowariancji lub macierzą korelacji (danych)

¹¹ponieważ $\frac{\partial \mathbf{x}^T \mathbf{A} \mathbf{x}}{\partial \mathbf{x}} = 2\mathbf{A} \mathbf{x}$

$$\boxed{(\mathbf{S} - \lambda\mathbf{I})\mathbf{a} = \mathbf{0} \quad \star} \quad \text{lub} \quad \boxed{\mathbf{S}\mathbf{a} = \lambda\mathbf{a} \quad \star\star}$$

Rozwiązaniem równania macierzowego (\star) lub ($\star\star$) są wartości własne macierzy \mathbf{S} i odpowiadające tym wartościom wektory własne:

$$\lambda_1 \geq \lambda_2 \geq \dots \lambda_d \geq 0$$

$$\mathbf{a}_1, \mathbf{a}_2 \dots \mathbf{a}_d$$

Generalnie rozwiązań jest d . Każdy wektor \mathbf{a}_i może posłużyć do utworzenia nowej zmiennej \mathbf{y}_i . Wartości własne mogą być zerowe (oznacza to współliniowość niektórych kolumn tablicy \mathbf{X} – wtedy odpowiadające tym wartościom (zerowym) wektory własne są niejednoznaczne).

Które z rozwiązań daje największą wariancję nowej zmiennej?

Każda para rozwiązań λ_i, \mathbf{a}_i musi spełniać ($\star\star$), tzn.

$$\mathbf{S}\mathbf{a}_i = \lambda_i\mathbf{a}_i.$$

Mnożąc tę równość lewostronnie przez \mathbf{a}_i^T otrzymamy

$$\mathbf{a}_i^T \mathbf{S}\mathbf{a}_i = \mathbf{a}_i^T \lambda_i \mathbf{a}_i$$

Skąd (ponieważ $\mathbf{a}_i^T \mathbf{a}_i = 1$)

$$\mathbf{a}_i^T \mathbf{S}\mathbf{a}_i = \lambda_i.$$

Tak więc λ_i jest wariancją nowej zmiennej: $\lambda_i = s_{\mathbf{y}_i}^2$, gdzie $\mathbf{y}_i = \mathbf{X}\mathbf{a}_i$.

Zauważmy, że nowe zmienne - utworzone z wektorów własnych macierzy \mathbf{S} noszą nazwę \mathbf{y}_i to są składowe główne Hotellinga, oznaczane w literaturze również jako PC_i .

Spis treści

6	Uczenie wg. zasady Hebba, Składowe główne	1
6.1	Zasada Hebba i siła powiązań między neuronami	1
6.2	Metoda PCA – podejście statystyczne	4
6.3	Obliczanie PCA za pomocą funkcji Matlaba	9
6.3.1	Obliczanie składowych głównych za pomocą funkcji 'eig' Matlaba	9
6.3.2	Obliczanie PCA za pomocą pakietu NETLAB	9
6.3.3	Obliczenia PCA za pomocą pakietu somtoolbox	10
6.4	Przykłady zastosowań metody składowych głównych	11
	Dodatek 1. Reguła Oji – interpretacja wag	13
	Dodatek 2. PCA – Rachunek algebraiczny	14
	Dodatek 2. Skrypt: Uczenie sieci według reguły Oji	16

Dodatek 2. Uczenie sieci według reguły Oji. Skrypt i wyniki

Dalej opisujemy przeprowadzone eksperymenty mające na celu pokazanie, jak uczy się sieć zaprogramowana według wzoru Oji (6.2). Badania przeprowadzono na danych dwuwymiarowych generowanych z dwuwymiarowego rozkładu normalnego o zerowym wektorze średnich i danej macierzy kowariancji \mathbf{C} (skrypt zadania znajduje się w Dodatku 2).

Interesowało nas dopasowywanie się pierwszego wektora własnego (oznaczonego umownie \mathbf{w}) do przedstawianej sieci próbki. Wyniki każdego eksperymentu pokazano na dwóch planszach: pierwsza z nich przedstawia wykres punktów indywidualnych (tzn. wylosowane punkty–próbki, na podstawie sieć się uczyła), z naniesioną trajektorią końca wektora \mathbf{w} oraz zaznaczonym końcowym wektorem \mathbf{w} . Druga plansza pokazuje na 2 subplotach, jak się zmieniały składowe wektora \mathbf{w} w trakcie uczenia. Można tu zauważyć, że w trakcie uczenia składowe wektora \mathbf{w} oscylują dookoła wartości teoretycznej.

Przeprowadzone eksperymenty na niezbyt dużych danych (liczebności próbek) rzędu $n=30, 50, 150$) wykazały, że wektor wag może być rozbieżny, jeżeli współczynnik uczenia jest większy od 0.1.

```
% skrypt oja.m ilustruje uczenie sieci wg. reguły Oji
% potrzebuje Netlab-a do gsamp
w=randn(1,2), eta=0.05; n=50;
mu=[0, 0]'; Sigma=[1 0.8; 0.8 1];% wektor srednich i m. kowariancji
x=gsamp(mu,Sigma,n); % probka z r. normalnego 2-wymiarowego
[lambda,w2]=pca(x); % metoda algebraiczna, daje wektory kolumnowe
%if (w2(1,1)<=0)|(w2(2,1)<=0), w2=w2*(-1); end
nk=5; kk=nk*n; % liczba prezentacji wylosowanej probki
ww=w; % disp(['w init= ', num2str(w)])

for h=1:nk, perm=randperm(n);
for k=1:n
x1=x(perm(k),:); y=w*x1'; %prezentowany wzorzec i wynik sumowania
delta=y*(x1-y*w); w=w+ eta*delta;
ww=[ww;w];
end % k zaprezentowano wszystkie elementy probki
end % h zaprezentowano h-ty raz probke
disp(['norma w =', num2str(w*w')]) %norm(ww(end,:), 'fro')
figure(1), clf,
subplot(3,1,1), plot(ww(:,1),ww(:,2));%, 'linewidth', 2);
hold on, plot(ww(end,1),ww(end,2),'.r', 'MarkerSize',24,...
'linewidth',3)
plot(x(:,1),x(:,2), 'g.' , 'MarkerSize',6);%'MarkerFaceColor','c');%
axis([-3, 3, -3,3])
f1=line([0 w(1,1)], [0 w(1,2)], 'linewidth',2,'color','m',...
'LineStyle', '--');
%set(f1,'color','k')
f2=line([0 w2(1,1)], [0 w2(2,1)], 'linewidth',2,'color','r');
Title(['Oja, \eta =', num2str(eta), ' n=', int2str(n),...
'nrep=', int2str(nk), ' c_{(1,2)} = ',num2str(Sigma(1,2))])
hold off
```



```

subplot(3,1,2), len=size(ww,1);
plot(ww(:,1)); hold on,
axis([0, kk, -1.5, 1.5]),
y1=w2(1,1);
line([1 1;len len],[y1 -y1; y1 -y1],...
      'linewidth',1,'color','r')
ylabel('1. skladowa w','FontSize',12)
subplot(3,1,3),plot(ww(:,2)); hold on,
axis([0, kk, -1.5, 1.5])
line([1 1;len len],[w2(1,1) -w2(1,1);w2(1,1) -w2(1,1)],...
      'color',[1,0,0])
ylabel('2. skladowa w','FontSize',12)
%for p=1:30, oja, pause(2),end % Mozna zrobic petle po skrypcie Oja1

```

Rysunek 6.5: Wyniki eksperymentu pokazującego uczenie się sieci kierunku głównego dwu-wymiarowego zbioru danych według reguły Oji. Współczynnik $\eta = 0.2$, kowariancja $c_{1,2} = 0.5$. Wyznaczony przez sieć kierunek główny w zasadzie pokrywa się z kierunkiem głównym wyznaczonym metodą algebraiczną. W trakcie uczenia składowe wektora \mathbf{w} oscylują dookoła wartości teoretycznej. {oja1 i oja2.eps}

Rysunek 6.6: {oja5.eps}

Rysunek 6.7: Jak rysunek 5.3. Współczynnik $\eta = 0.2$, kowariancja $c_{1,2} = 0.8$. Wyznaczany przez sieć wektor własny w trakcie uczenia zmienił nieoczekiwanie zwrot. Końcowy wektor \mathbf{w} dość znacznie różni się od teoretycznego wektora \mathbf{w} wyznaczonego metodą algebraiczną. {oja5, oja6.eps}