

## N E U R A L N E T W O R K S 2008/2009

### 1 LIST NB. 1. WORKING WITH MATLAB

symbol '↵' means 'press the ENTER key'

Ex. 1.1. At the Matlab prompt (in the Matlab Working Window) open the scripts `brain.m` and `iris4.m` by typing  
`edit brain_`  
`edit iris4 ↵`,  
`edit wine_↵`

See, how the scripts are prepared, notice the comments, and the declarations of the variables labels and of the data matrix.

Notice, whether the workspace of Matlab contains any variable, by typing  
`whos↵`

Ex. 1.2. At the Matlab prompt, execute the scripts `brain.m`, `wine.m` and `iris4.m` by typing  
`brain_↵`  
`iris4↵`  
Next type again  
`whos↵`.

As the answer, you should now obtain the information, that the workspace contains the variables `brain`, `iris`, `wine` and labels of their columns, e.g. `irisCLabs`, `irisLabs`. What is the difference of the content of these variables?

Ex. 1.3. Displaying the means of the data tables `brain`, `iris`, `wine` from the workspace.  
Taking as example the `iris` data. The matrix `iris` of size  $150 \times 4$  contains 150 rows (flowers) and 4 columns (variables, attributes).  
The `iris` flowers belong to 3 groups (species of irises).  
Rows 1–50 contain irises from the species *Iris setosa*,  
rows 51–100 contain irises from species *Iris versicolor*,  
rows 101–150 contain species *Iris virginica*.

- Calculate the means (using Matlab function `mean`) and standard deviations (using Matlab function `std`) first for the whole group of all irises, and next for each species separately. *Hint:* `m1=mean(iris(1:50,:)); s1=std(iris(1:50,:));`

Say, you calculated the vectors `mm`, `m1`, `m2`, `m3`.

- Put the vectors together into one matrix `M=[mm;m1;m2;m3]`

Notice, each row of `M` contains one vector of means.

- Plot the vectors of means in one scatter-plot issuing the command

`h=plot(M')↵`

Notice: You got the handle `h` to the plot. The handle has 4 subhandles, corresponding to the plots of the four vectors of means. The handles allow to change some properties of the plot(s), markers, linewidth etc.

*Hint: you may consult macro-file `jobann1.m` in this respect.*

Ex. 1.4. To make a display of scatterplots for each pair of the four columns (attributes) of the iris flowers, issue the command

```
h=plotmatrix(iris)↔
```

Look at the plot and find out, whether there are some clusters or outliers in the data. If yes, then for which pairs of attributes?

Ex. 1.5. An alternate function for making a scatterplot matrix. Type the command: (text of the M-file `scatm` is given at the end of this list)

```
genes=iris; comp_nam=irisCLabs; scatm ↔
```

Ex. 1.6. Construct a scatterplot matrix from the iris data, Using only the subgroups 'Iris versicolor' and 'Iris setosa'.

Do you see any clusters in the scatterplots? If yes, then for which pairs?

Ex. 1.7. Construct a scatterplot matrix from the iris data, taking only variables no. 2, 3, 4.

*Remark.* You may use the Matlab function `h=plotmatrix(X)` or modify the script `scatm` shown below.

Ex. 1.8. Construct a scatterplot matrix from the brain data.

*Remark.* You may use the Matlab function `h=plotmatrix(X)` or modify the script `scatm` shown below.

```
% ----- jobANN1 -----
% file jobANN1, simultaneous display of several vectors of means from iris
mm=mean(iris); % entire data
m1=mean(iris(1:50,:)); m2=mean(iris(51:100,:)); m3=mean(iris(101:150,:));
M=[mm;m1;m2;m3]
h=plot(M') % h is 'handle' to the displayed plot
legend(h,'All','Setosa','Versicolor','Virginica')
%% 'h' is vectorized, contains handles to displays of the 4 means
xlabel('No. of the variable'), ylabel('Group means'),
title('Iris n=150 Grand Mean and Subgroup Means')
set(h,{'LineStyle'},{'-';'-.'; '--'; ':'})
set(h,{'Color'},{'m';'r';'g';'b'})
set(h,{'Marker'},{'o';'s';'d'; '+'})
set(h(1),'linewidth',2)
set(h(1),'color','m','MarkerFacecolor','m'); % green, 'g' or [0,1,0]
% command get(h(1)) prints all properties of the graph displaying 'mm'
% ----- end jobANN1 -----

% ----- scatm -----
%% scatmg - drawing scatterplot matrix for data named genes size nxd
%%          comp_nam={'x1','x2' ...} contains labels of variables
%%          displaying variables c1 through c2, default 1 to d
%%          the 'genes' and 'comp_nam' variables should be in workspace

%fh1=figure; % [n,d]=size(genes); c2=d;
c1=1; c2=size(genes,2); % can be changed
HH=[c1:c2]; % [1 2 3 4 5]; % [11 12 13 10]; % y-variables in the plot
```

```

cc= HH;% [4 5 6 7 8 9 ]; % x-variables in the plot
k=1; p=length(HH); c=length(cc);
% comp_nam={ 'x1' 'x2' 'x3' 'x4' 'x5'
% 'angle1', 'angle2', 'angle3' ...
%           '%x1' 'y1' 'x2' 'y2' 'x3' 'y3' 'length' ...
%           % 'rho1' 'rho2' 'rho3'
% };
for i=HH, % horizontally
  for j=[cc], % vertically
    if i==j,
      subplot(p,c,k); hist(genes(:,i));
      xlabel(comp_nam{i},'fontweight','bold') %title
    % elseif i<j,
  else subplot(p,c,k);
    plot(genes(:,j),genes(:,i),'m.','Markersize',12) % specify groups
    if i==HH(1), title(comp_nam(j),'fontweight','bold'), end;
    if i==HH(p), xlabel(comp_nam(j),'fontweight','bold'), end;
    %xlabel(comp_nam{i})
    % ylabel(comp_nam{j})
    if j==cc(1), ylabel(comp_nam(i),'fontweight','bold'), end;
  end
  k=k+1;
end
end;
drawnow; %colormap hsv
% ----- end scatm -----

```

## 2 LIST NB. 2. MULTILAYER PERCEPTRON (MLP)

symbol '↔' means 'press the ENTER key'

Ex. 2.1. Netlab demo demmlp1 for a simple **regression task**.

- Read the demo into your personal Matlab editor by typing in the command window:

```
edit demmlp1↔
```

Save the file to your personal working directory.

- Run the demo by typing in the command window

```
demmlp1↔
```

- Change in your copy of the `demmlp1.m` file the sinus function to another one. Run the demo again.

Ex. 2.2. Read from ROBIPR (article of Robi Policar in the *Wiley Encyclopedia of PR*, a manuscript downloadable from

<http://users.rowan.edu/~polikar/RESEARCH/PUBLICATIONS/> )

the Section 4.2.1 on Bayes classifiers, pp. 8-9 from the manuscript . This is necessary to understand the Netlab demonstration demmlp2.

Ex. 2.3. Run the netlab demo on MLP performing a **classification task** by running the file `demmlp2`.

The task is the following: We generate 3 groups of data, out of with we retain the first group of data as it is (it is considered as it is and called GROUP ONE). The generated groups 2 and 3 are fused together and constitute in the following a joint group called GROUP TWO.

The networks task is to build a decision boundary between GROUP ONE and GROUP TWO.

The quality of the calculated decision boundary is checked on test data and evaluated by so called **confusion matrix**.

The results yielded by the networks are compared with those yielded by the optimal Bayesian rule.

Memorize, what is a confusion matrix and how to construct it.

```
function fh=config(y, t); function [C,rate]=confmat(Y,T);
```

```
function plotmat(matrix, textcolour, gridcolour, fontsize);
```

Ex. 2.4. Open the `demmlp2` file in your personal editor. Try to change the parameters (centers) of the generated data, making the classification task easier or more difficult for the network; for each trail notice the overall percentage of correct classification made by both methods.

Summarize the results of your trials writing them down together making a small report of your experiments.

Ex. 2.5. The logistic activation function is defined as:

$$f_{logi}(x) = \frac{1}{1 + \exp(-\beta x)}, \quad -\infty < x < \infty, \quad \beta > 0 \quad (1)$$

Plot the logistic activation function for various values of the parameter  $\beta$ . (if you are not familiar with Matlab, you may get hints how to do it - by inspecting the `demomlp1` file).

Ex. 2.6. \* **Important!** Calculate the first derivative of the logistic activation function

$$\frac{\partial f_{logi}(x)}{\partial x} = \beta f_{logi}(x)(1 - f_{logi}(x)) \quad (2)$$

Ex. 2.7. \* The *tangens hiperbolicus* (**tanh**) activation function is defined as:

$$f_{tanh}(x) = \frac{e^{\beta x} - e^{-\beta x}}{e^{\beta x} + e^{-\beta x}}, \quad -\infty < a < \infty, \quad \beta > 0 \quad (3)$$

Plot the **tanh** activation function using various values of the parameter  $\beta$ .

Show that the **tanh** function eq.(3) can be expressed by the logistic  $f_{logi}$  eq.(2) function as follows

$$f_{tanh}(x) = 2f_{logi}(x) - 1.$$

Ex. 2.8. \*\* (book by Osowski 1996 p. 39) Show that the derivative of the **tanh** activation function is equal to

$$\frac{\partial f_{tanh}(x)}{\partial x} = \beta(1 - f_{tanh}(x))$$

or, alternatively, this derivative may be expressed through the logistic activation function as

$$\frac{\partial f_{tanh}(x)}{\partial x} = 2\beta f_{logi}(x)(1 - f_{logi}(x)).$$

Ex. 2.9. \*\* Say, we consider only the two first attributes of the *iris* flowers (i.e. the first two columns of the data matrix *iris*).

Define an additional data vector **ind** of size 150 (=n, the total nb of rows in the *iris* data. The vector should take only values 1, 2, 3 – depending on the group the *i*th flower belongs to:

$$ind(i)_{i=1,\dots,n} = \begin{cases} 1, & \text{if the } i\text{th flower is Setosa} \\ 2, & \text{if the } i\text{th flower is Versicolor} \\ 3, & \text{if the } i\text{th flower is Virginica} \\ 0, & \text{otherwise.} \end{cases}$$

Using the **plot** function, make a scatterplot displaying each iris flower – characterized by its first two attributes – with a marker and color specific for the group of irises, the subsequent flower belongs to.

Ex. 2.11 Run the demo: **nnd4db** – decision boundary. Explain the principles of its work.

Ex. 2.12 Run the demo: **nnd10lc** – learning classification. Explain the principles of its work.

Ex. 2.13 Run the demo: **nnd10nc** – noise cancellation. Explain the principles of its work.

Ex. 2.15 **Matlab functions for dealing with MLP**

The functions for creating, training and working with MLP are:

```
function net = mlp(nin, nhidden, nout, outfunc);
```

Creates a network called 'net' with one hidden layer. Input parameters:  
nin – number of inputs, denoted by us as 'd'

nhidden – number of hidden units denoted by us as 'H'

nout – number of outputs, denoted by us as 'c'

outfunc – transfer function used by the output layer ('linear', 'logistic', softmax')

`function [net, options] = netopt(net, options, x, t, alg);` Function for training the created 'net' network with the training data 'x' (of size  $n1 \times d$ ) and target data 't' of size  $n1 \times nout$ . Options for training and kind output are given in the vector `options(1:18)` — consult the function `glm` for the meaning of individual elements of `options`. The function 'netopt' accepts as 'alg' the following algorithms: 'conjgrad', 'quasineu', 'scg'.

`function [y] = mlpfwd(net, x);` Function using the learned network 'net' for performing the desired task from the domain of regression (predictions) and classification. The task is performed for data 'x' of size  $n \times d$ . The output is 'y' of size  $n \times c = nout$ . We may wish to obtain in the output additional information about the work of the hidden layer. Then we call `[y,z,a] = mlpfwd(net, x);` The network output matrix 'z' of size  $n \times H$  contains the (transformed by *tanh*) output of the hidden layer – forwarded to the output layer. The network output matrix 'a' of size  $n \times c$  contains the activations of the output neurons – before applying to them the transfer function 'outfun' attached to the 'net' network.

Ex. 2.15 Calculate a decision boundary for the iris flowers belonging to the subgroups 'setosa' and 'versicolor'. Calculations for 2 variables only.

Create from the available data `iris(1:100, 1:2)` a training set and a test set. Create an appropriate network, train it using the established training data.

Examine the structure of the Matlab struct 'net'.

Ex. 2.16 Check the performance of the network, using the established 'test data'.

Visualize graphically the obtained results and write a comment on the behavior of the trained network.

Ex. 2.17 \* Confusion matrix. Use the functions

`fh=conffig(y, t)` or function `[C,rate]=confmat(Y,T)`.

for visualization of the results obtained in Exercise 2.15 (quality of discrimination between the two species of iris) yielded by the network 'net'.

### 3 PERCEPTRON PROSTY. DYSKRYMINACJA DLA 2 GRUP

Data: 4 listopada 2008

Będziemy posługiwać się danymi zebranymi przez prof. Seunga z MIT. Dane zostały trochę przeorganizowane. Oryginalne dane znajdują się pod adresem ([http://hebb.mit.edu/courses/9.641/assignments/perceptron\\_delta.m](http://hebb.mit.edu/courses/9.641/assignments/perceptron_delta.m)).

Przeorganizowane dane są dostępne na serwerze studenckim w folderze **abaFiles**, znajdującym się w s. 107 na dysku twardym D na ścieżce `D:/Matlab/toolbox/abaFiles`. Interesują nas pliki

`seungMNIST.mat`, `seungFaces.mat`

oraz funkcję `[w,b]=seung(train, trainLabs,options)`. Funkcja `seung` oblicza wagi właściwe  $w$  oraz bias  $b$  potrzebne do zbudowania logistycznej funkcji dyskryminacyjnej dla dwóch grup danych.

Pliki z danymi o rozszerzeniu `.mat` wczytujemy za pomocą instrukcji `load`.

cyferki			twarze		
<code>seungMNIST.mat</code>			<code>seungFaces</code>		
<code>mnistTrain</code>	784x5000	uint8	<code>facesTrain</code>	361x6977	double
<code>mnistTrainLabs</code>	5000x1	uint8	<code>facesTrainLabs</code>	6977x1	double
<code>mnistTest</code>	784x1000	uint8	<code>facesTest</code>	361x2000	double
<code>mnistTestLabs</code>	1000x1	uint8	<code>facesTestLabs</code>	2000x1	double

Otrzymane zmienne zawierają zarówno pliki uczące (`Train`) jak i pliki do testowania (`Test`).

Identyfikacja cyferek znajduje się w wektorach `mnistTrainLabs` i `mnistTestLabs`: składowe tych wektorów zawierają znaki  $0, 1, 2, \dots, 9$  odpowiednio.

Identyfikacja twarzy jest określona binarnie: wartość 1 oznacza, że dana kolumna przedstawia twarz, wartość 0, że dana kolumna nie przedstawia twarzy.

### Rozpoznawanie cyferek

Lab. 3.1. Wczytaj plik `seungMnist.mat` i zobacz jakie zmienne pojawiły się w przestrzeni roboczej. Jakie wartości przyjmują te zmienne.

Wykreśl niektóre kolumny za pomocą instrukcji `imagesc`.

Lab. 3.2. Przyjmij dwie cyferki,  $c1$  i  $c2$  (np.  $c1 = 5$ ,  $c2 = 9$ ).

- (a) Wyznacz funkcję dyskryminacyjną która potrafi rozpoznawać między cyferkami  $c1$  i  $c2$ .
- (b) Określ macierz pomieszania (confusion matrix).
- (d) Napisz makro wyświetlające dynamicznie źle zaklasyfikowane cyferki (po wyświetleniu każdej cyferki ma nastąpić krótkie zatrzymanie obrazu).
- (e) Zmodyfikuj makro w punkcie (d) tak, żeby różne cyferki były wyświetlane za pomocą różnych palet (`colormap`)
- (c) Zbudowaną funkcję dyskryminacyjną przetestuj na pliku testowym. Wyznacz również macierz pomieszania (confusion matrix).

Lab. 3.3. Wykonaj analizę danych z twarzami w podobny sposób jak analiza danych cyferkowych.

UWAGA. Dane cyferkowe są w pojedynczej precyzji. **Do obliczeń numerycznych trzeba zamienić dane uint8 na podwójną precyzję** (instrukcja `double`). Dane z twarzami są już w podwójnej precyzji.

## 4 FUNKCJE SEUNG (SEUNG1) DO RÓŻNICOWANIA DWÓCH ZBIORÓW GRAFICZNYCH

Data: 13 listopada 2008

Wybrać z danych seungMnist dwie cyferki, np. c1 i c2.

Zbadać, jak dobrze funkcja `seung` lub `seung1` potrafi zróżnicować te literki.

Wyniki obliczeń sporządzone jako Raport html (opcja Edytora Matlaba) powinny być do mnie dostarczone do końca tygodnia, tj. do 16 listopada b.r. Proszę o przysłanie skryptu podstawowego z rozszerzeniem `.m` oraz skryptu w formacie `.html` wyprodukowanym przez Edytor Matlaba.

Wyniki powinny zawierać następujące punkty:

1. Temat raportu, nazwisko autora (autorów).
2. Cel pracy, na jaką cyferkę ma wskazywać szukany wektor wag, przygotowanie danych do trenowania i testowania.
3. Trenowanie sieci za pomocą funkcji `seung/seung1`.
4. Ocena jakości wyznaczonej funkcji dyskryminacyjnej na zbiorze treningowym.
5. Ocena jakości wyznaczonej funkcji dyskryminacyjnej na zbiorze testowym.
6. Podsumowanie wyników, ewtl. dyskusja na temat zastosowanej metody i otrzymanych wyników.

## 5 UCZENIE BEZ NADZORU. SKŁADOWE GŁÓWNE

Data: 20 listopada i 27 listopada 2008

Zbadaj dowolny zbiór danych (znajdujący się w kartotece `abaFiles` lub gdzieindziej) na jego wymiarowość. Niech będzie to tablica  $X$  o rozmiarze  $n \times d$ .

Uwaga! Wybrane dane powinny być odpowiednio wystandaryzowane.

Odpowiedz na pytania:

1. Ile składowych głównych potrzeba, aby odtworzyć 95 procent zmienności (inercji) badanego zbioru danych. Przypuśćmy, że potrzeba ich  $K$ . Oczywiście  $K$  powinno być mniejsze niż  $n$ .

2. Wyznacz  $K$  składowych głównych i odtwórz na ich podstawie analizowane dane.

Niech  $XX$  o rozmiarze  $n \times d$  oznacza odtworzoną tablicę danych, zrekonstruowaną na podstawie  $K$  pierwszych składowych głównych.

3. Wybierz jakiś mały podzbiór numerów ze zbioru  $1, \dots, n$ . Przypuśćmy że zbiór wybranych numerów zapamiętano w tablicy `nr` o liczebności  $n1$  (np.  $n1 = 5$ ).

Sporządź wykres profilowy osobników o numerach `nr` znajdujących się w tablicach  $X$  i  $XX$ .

Tym samym dla każdego osobnika otrzymamy dwa profile: na podstawie obserwowanych i aproksymowanych danych. Wybrać tak kolory/markery/rodzaj linii łączącej, aby pary profili dla jednego osobnika były dobrze widoczne. (Ewentualnie zastosuj technikę *subplotów*).

4. Opcyjnie zaproponuj inne metody porównywania macierzy  $X$  i  $XX$ .



## 6 MAPY KOHONENA – KLASTERYZACJA DANYCH

Data: 4 grudnia i 11 grudnia 2008

*Uwagi wstępne.* Software: Matlab SomToolbox autorstwa Vesanto i inn., Helsinki.

O możliwościach tego pakietu można się dowiedzieć z jego modułów demonstracyjnych `som_demo1`, `som_demo2`, `som_demo3`, `som_demo4`.

Obliczenia proszę wykonać na danych `lettersUCI` (skrypt `lettersUCI.m`). Skrypt zawiera około 20 tys. literek alfabetu angielskiego zakodowanych jako wektory o 17 składowych. Pierwsza składowa zawiera symbol danej literki (A:Z), a następne składowe są wartościami 16 cech morfologicznych charakteryzujących te literki. Konstrukcja map Kohonena ma się odbywać na podstawie danych obejmujących dwie literki z posiadanej bazy. Klucz wyboru literki jest następujący:

Każdy student zapisany na pracownię ma swój numer identyfikacyjny (1 do 25) na liście zapisanych na pracownię. Nazwijmy ten numer  $J$ . Należy z ciągu Liter A, B, ... Z wybrać literę o numerze  $J$ , oraz następującą po niej, czyli zapisaną na miejscu  $J+1$ . Tak więc opracowywane dane będą zawierały litery występujące w angielskim alfabecie na miejscach  $J$  i  $J+1$ .

Proszę makra do obydwu zadań napisać w jednym skrypcie w konwencji Raportu Matlabowskiego w formacie html. Być może będę chciała zobaczyć to makro w czasie oddawania zadania.

**Zadanie 6.1** (a) Dla swoich danych skonstruuji 2-wymiarową mapę Kohonena o jednostkach hexagonalnych.

(b) Wykreśl na oddzielnym rysunku profile wektorów kodowych trzech wektorów kodowych (codebook vectors) znajdujących się w czterech narożnikach mapy. Wszystkie 12 profili mają być na tym samym rysunku; jednak trójki powinny być wyrysowane tak samo (tj. tym samym kolorem).

(c) Oblicz wskaźniki błędu kwantyzacji i błędu topologicznego.

Wykonaj obliczenia (a) i (c) dwa razy: dla danych standaryzowanych i nie-standaryzowanych. Porównaj wyniki.

Skomentuj wyniki i zapisz je do raportu.

**Zadanie 6.2** Wykreśl na oddzielnym rysunku dwie mapy Kohonena o tym samym rozmiarze i kształcie, co w zadaniu 6.1.

Umieść na mapach informacje o obsadzie poszczególnych heksagonów przez wybrane dwie literki ( $'J'$  i  $'J+1'$ ).

Skomentuj wyniki, tj. zapisz je za pomocą odpowiedniego tekstu w przygotowywanym Raporcie.

**Uwagi dodatkowe:** Wymienione dwa laboratoria są poświęcone mapom Kohonena. Obowiązują obydwu zadania. Na zadanie 6.1 jest przewidziany termin 4 grudnia, a na zadanie 6.2 termin 11 grudnia. Można oddać obydwu zadania w jednym terminie.

Laboratorium 18 grudnia jest przeznaczone na indywidualną pracę własną do projektu oddawanego w czasie 'egzaminu'. W tym samym terminie można oddać zaległe zadania potrzebne do zaliczenia pracowni.

---

## 7 Terminy referowania 'projektów'

*Uwaga.* Utracenie terminu jest równoważne z deklaracją przystąpienia do normalnego egzaminu z całości materiału. Studenci mogą się wymieniać terminami między sobą. Na jeden termin przypada maksymalnie 8 osób. Czas referowania: godz. 16:15 do 20:00.

W mies. grudniu można – w miarę możliwości – zmieniać tematy.

### 8 stycznia 2009 referują:

nr kolejny	id	nazwisko	temat
1	24	Szczepański	T22
2	11	Kimel	
3	7	Faleński	T1
4	13	Kowaluk	T23
5	8	Fuss	T35
6	16	Kuliński	T34
7	22	Such	
8	2	Brzeziński	T25

### 15 stycznia 2009 referują:

nr kolejny	id	nazwisko	temat
1	6	Teler	T27
2	5	Dolecki	T2
3	18	Milewski	T2
4	14	Kraszewski	T29,T30
5	21	Styrczula	T29,T30
6	10	Kercz	T24
7	9	Imos	T31
8	3	Cecki	T32

### 22 stycznia 2009 referują:

nr kolejny	id	nazwisko	temat
1	23	Szampera	T21 ?
2	4	Dąbrowski	T20
3	20	Olszewska	T28
4	1	Adamus	
5	15	Kudelko	T3
6	17	Michaluk	T4
7	12	Klimek	T8
8			

---

## 8 Tematy indywidualnych projektów na egzamin z Sieci Neur.

*Uwagi.* Tematy powinny być zreferowane w czasie 'normalnych' zajęć w miesiącu styczniu 2009, podczas których jest obowiązkowa obecność wszystkich studentów. Optymalna forma referowania: część teoretyczna projektu na wykładzie, część praktyczna na pracowni. Terminy referowania zostaną ustalone na najbliższych zajęciach. Niezreferowanie w wyznaczonym terminie (lub otrzymanie oceny niezadawalającej z referowanego tematu) powoduje, że student przystępuje do normalnego egzaminu z sieci neuronowych.

1. Pocket algorithm. Zaimplementować w Matlabie i pokać jego działanie. **FALEŃSKI**
2. Algorytm backpropagation. Implementacja w Matlabie dla MLP z jedną warstwą ukrytą. **DOLECKI, MILEWSKI**
3. Znajdywanie jednorodnych grup w danych 'AminokwasyDNA'. Metoda kmeans\_cluster zaimplementowana w Matlab SomToolbox. Ewntl. pominąć cechy 5,9,13,18 **MICHALUK**
4. Znajdywanie jednorodnych grup w danych 'AminokwasyDNA'. Metoda GMM (Gaussian mixture models) zaimplementowana w Netlabie. **KUDEŁKO**
5. Wyznaczanie prototypów metodą gazu neuronowego (SomToolbox). Odwzorować otrzymane prototypy (np. techniką hitów) na mapie Kohonena skonstruowanej z analizowanych danych.
6. Budowa histogramu na mapie Kohonena - metoda SDH (Smoothed Data Histogram) Pampalka.
7. Metoda Oji wyznaczania pierwszej składowej głównej; należy napisać implementację w Matlabie. Pokazać jak to działa na różnych danych.
8. Przedstawić algorytm kompresji i dekompresji danych za pomocą sieci MLP. Porównać wyniki z wynikami otrzymanymi algebraiczną metodą PCA. **KLIMEK**
9. Rozpoznawanie twarzy i nie-twarzy. Porównać metodę Seunga z wynikami liniowej funkcji dyskryminacyjnej o współczynnikach otrzymanych za pomocą Matlabowskiej funkcji mldivide nazywanej 'left divide' (jesli  $\mathbf{A}\mathbf{w}=\mathbf{y}$ , to wektor  $\mathbf{w}$  otrzymuje się jako  $\mathbf{w}=\mathbf{A}\backslash\mathbf{y}$ ).
10. Porównanie wyników klasyfikacji kilka grup danych za pomocą funkcji LVQ1 i LVQ3 (somtoolbox) z wynikami otrzymanymi metodą MLP.
11. Mamy dane składające się z kilku podgrup (np. wine lub iris). Wyznaczyć centra metodą LVQ1 (LVQ3), sklasyfikować dane, obliczyć macierz pomieszenia.
12. Kanoniczna analiza dyskryminacyjna na podstawie zmienności międzygrupowej i wewnątrzgrupowej (metoda R.A. Fishera). Zilustrować dla dwóch grup danych.
13. Kanoniczna analiza dyskryminacyjna na podstawie zmienności międzygrupowej i wewnątrzgrupowej (Metoda Fishera). Zilustrować dla kilku grup danych.
14. Sporządzić jedną 1-wymiarową mapę Kohonena dla 2 grup danych wybranych z danych iris i wine. Przedstawić wyniki graficznie, zaznaczając wyraźnie prototypy i punkty danych odmiennymi markarami i kolorami.
15. Zilustrować proces tworzenia mapy dwuwymiarowej na płaszczyźnie na podstawie danych o rozkładzie jednostajnym w kwadracie:  $[0,1] \times [0,1]$ .
16. Zilustrować proces tworzenia mapy jednowymiarowej z danych o rozkładzie jednostajnym w obszarze przedstawiającym trójkąt.

17. Tworzenie mapy dwuwymiarowej z danych ułożonych w jednostkowej kostce przestrzennej.
  18. Wygenerować dwie grupy danych dwu-wymiarowych. Trenujemy perceptron prosty tylko za pomocą jednej grupy danych. Czy taki perceptron nauczy się rozpoznawać obydwie grupy danych? Zilustrować graficznie.
  19. Wpływ losowych zakłóceń danych na wyniki klasyfikacji – na przykładzie danych seungFaces.
  20. Wpływ losowych zakłóceń danych na wyniki klasyfikacji – na przykładzie danych dwóch literek pobranych z lettersUCI. **DĄBROWSKI**
- DO NASTĘPNYCH TRZECH TEMATÓW POLECAM STRONĘ

[http://bossa.pl/analizy/techniczna/elementarz/sieci\\_neuronowe/](http://bossa.pl/analizy/techniczna/elementarz/sieci_neuronowe/)

21. Analiza notowań danych finansowych. Zagadnienie 1. **SZAMPERA**
  22. Analiza notowań danych finansowych. Zagadnienie 2. **SZCZEPAŃSKI**
  23. Analiza notowań danych finansowych. Zagadnienie 3. **KOWALUK**
  24. Praca do zreferowania: H. A. Rowley, S. Baluha, T. Kanade: Neural Network-Based Face Detection. IEEE PAMI, January 1998 **KERCZ**
  25. Praca do zreferowania: J.S. Rani, D. Devaraj, R. Sukanesh: Robust Face Recognition using Wavelet Transform and Autoassociative Neural Network. IJBM 2008, no. 2. **ZAJĘTY BRZEZIŃSKI**
  26. Praca do zreferowania: B. Yegnanarayana, S.P. Kishore: AANN: an alternative to GMM for pattern recognition. Neural Networks 15 (2002), 459–469.
  27. Praca do zreferowania: M. La Rocca, C. Perna: Modelling complex structures by artificial neural networks. KNEMO Anacapri 2006. **TELER**
  28. Praca do zreferowania: Feng Chu, Lipo Wang: Applying RBF Neural Networks to Cancer Classification Based on Gene Expression. IJCNN Vancouver 2006. **OLSZEWSKA**
  29. Praca do zreferowania: Robi Polikar i inn., An ensemble based data fusion approach for early diagnosis of Alzheimers disease. Information Fusion 9 (2008) 8395. **STYRCZULA-KRASZEWSKI**
  30. Praca do zreferowania: Robi Polikar i inn. : Comparative multiresolution wavelet analysis of ERP spectral bands using an ensemble of classifiers approach for early diagnosis of Alzheimers disease. Computers in Biology and Medicine 37 (2007) 542–558. **STYRCZULA-KRASZEWSKI**
  31. Praca do zreferowania: Matthias Schonlau, William DuMouchel, i inn. : Computer intrusion: Detecting Masquerades. Statistical Science, 2001, Vol. 16, no. 1, 1–17. **ZAJĘTY IMOS-CECKI**
  32. Praca do zreferowania: Praca do zreferowania: Han-Sung Kim, Sung-Deok Cha: Empirical evaluation of SVM-based masquerade detection using UNIX commands. Computers & Security 24, 160–168. **CECKI-IMOS**
  33. Praca do zreferowania: Shian-Chang Huang: Combining extended Kalman Filters and Support Vector Machines (SVM) for online option price forecasting. Konferencja JCIS 2005, str 1-4. ATLANTIS PRESS scientific publishing.
  34. Praca do zreferowania: Analiza wyników EEG przy pomocy probabilistycznych sieci neuronowych. **KULIŃSKI**
  35. Praca do zreferowania: Recognizing emotions expressed by body pose: a biological inspired neural model. **FUSS**
-