MatrikelNr:

GOETHE UNIVERSITÄT FRANKFURT AM MAIN

Klausur zur Vorlesung Adaptive Systeme
Wintersemester 2011/2012

Datum: 16.02.2012

Vorname:
Name:
Matrikelnummer:
Geburtsdatum:
Studiengang:

Als BSc bearbeiten Sie bitte den Teil der Aufgaben, der mit „AS-1“ gekennzeichnet ist. Als MSc bearbeiten Sie den“AS-1“ und/oder den „AS-2“-Teil. Im AS-1-Teil sollen 80 Punkte (5CP) aus den Aufgaben 1-5 erreicht werden. Im AS-2-Teil sollen aus den Aufgaben 5-10 ebenfalls 80 Punkte (5CP) erreicht werden.

Die Punktzahl einer Aufgabe entspricht ungefähr der maximalen Bearbeitungsdauer der Aufgabe in Minuten.

Durch die Übungspunkte können maximal 10% der Klausurleistung erbracht werden. Als Hilfsmittel ist ein Taschenrechner erlaubt. Bitte benutzen Sie für Notizen die Rückseiten der Aufgabenblätter.

Viel Erfolg!

Wird vom Prüfer ausgefüllt:

1	2	3	4	5	6	7	8	9	10	Σ
/10	/25	/30	/15	/40	/20	/10	/15	/5	/10	

Punkte Klausur:
Punkte Übungen:
Punkte Gesamt:
Note:

MatrikelNr:

AS-1.1 Grundlagen 10 Punkte

a) Wie und warum kann man die Multiplikation einer Matrix mit einem Vektor mittels eines neuronalen Netzes durchführen? (5P)

b) Was sind die Vor- und Nachteile von Online- bzw. Offline-Learning? (2P)

c) Welche Maßnahmen können durchgeführt werden, um Overfitting zu verhindern? (3P)

a) Ein einzelnes, lineares Neuron implementiert das Skalarprodukt $y = \mathbf{w}^T\mathbf{x}$. *Geben wir die Eingabe* $\mathbf{x}$ *an* n *lineare Neuronen, so ist das Tupel aller Ausgaben* $\mathbf{y} = (y_1,\ldots,y_n) = (\mathbf{w}_1^T\mathbf{x},\ldots, \mathbf{w}_n^T\mathbf{x}) = (\mathbf{w}_1,\ldots,\mathbf{w}_n)^T\mathbf{x} = \mathbf{W}\mathbf{x}$ *was einer Matrix-Vektor-Multiplikation entspricht.*

b) ***Offline-Learning****: Nachteil: nur begrenzte Anzahl von Trainingsmustern, die nicht unbedingt die echte Verteilung widerspiegeln.*
Vorteil: Alle Trainingsmuster liegen gleichzeitig vor und erlauben eine globale Optimierung.
Online-Learning*: Vorteil: Im Laufe der Zeit können sehr viele Trainingsmuster verwendet werden und damit eine gute Schätzung der Verteilung erzielt werden. Selbst eine nicht-stationäre Informationsquelle kann man kurzzeit-approximieren.*
Nachteil: Eine Verallgemeinerung kann nicht über alle Trainingsmuster erzielt werden, sondern nur über die bisher gesehenen.

c) Overfitting vermeidet man, indem man eine Generalisierung beim lernenden System erzwingt. Dies kann durch Verwendung nur weniger Parameter (z.B. wenige Neuronen, kleine Netze) oder durch Abbruch des Lernens bei steigendem Testfehler (stopped training) erreichen.

AS-1.2 Lernregeln für Fehlerminimierung 25 Pkte

a) Erklären Sie die Unterschiede zwischen der Lernregel des Perzeptron und der von Widrow-Hoff. Die Widrow-Hoff Gleichung lautet (2P)

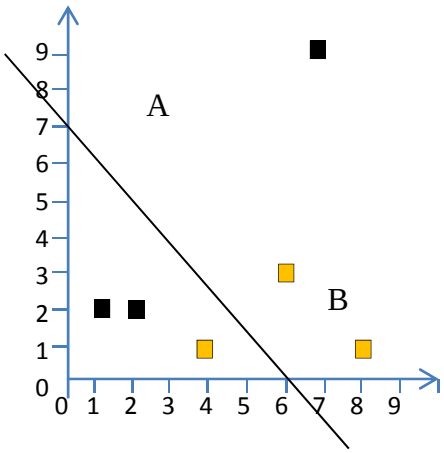
$$\mathbf{w}(t+1) = \mathbf{w}(t) + \gamma(L - \mathbf{w}^T\mathbf{x})\,\mathbf{x}\,/|\mathbf{x}|^2$$

Führen Sie die Widrow-Hoff Lerngleichung für folgende Muster für eine Iteration aus:
Klasse A: (1,2), (7,9), (2,2), Klasse B: (6,3), (8,1), (4,1)

b) Beginnen Sie dafür mit der eingezeichneten Klassengrenze als initialem Wert. Welchem initialen Gewichtsvektor **w**(0) entspricht sie? (5P)

c) Verwenden Sie ein geeignetes Muster einer Klasse zur Iteration der Klassengrenze und schreiben Sie für den Lernschritt bei γ=0.4 die Werte der Eingabe, der Ausgabe und des Gewichtsvektors **w**(1) hin. *Hinweis*: Kodieren Sie Klasse A mit L=-1, B mit L=+1. (15P)

d) Zeichnen Sie nach dem Lernschritt im Diagramm die Lage der Geraden der aktuellen Klassentrennung ein. Was stellen Sie fest? Wie verändert sich die Gerade? (3P)



***Lösung*:**

*a) Der wesentliche Unterschied zwischen beiden Gleichungen liegt im Vergleichsterm (L-y) beim Perzeptron bzw. (L-z) bei Widrow-Hoff: y ist die nichtlineare Ausgabe S(z), während z=**wx** die lineare Aktivität bedeutet. Außerdem ist die Widrow-Hoff Eingabe x mit dem Term x^2 normiert.*

*b) Die Lage der initialen Klassengrenze ist mit y=ax+b gegeben, wobei a=-7/6 und b=7 ist.Damit ist 0 = -7/6x-y+7 = $w_1x_1+w_2x_2+w_3$ oder **w**(0) = (-7/6, -1, 7)*

*c) der größte Fehler ist durch das Muster (7,9) der Klasse A gegeben. Die Aktivität ist somit **wx** = (-7/6,-1,7)(7,9,1)^T = -10,167. Die Lehrervorgabe dafür war L(A) = -1, so dass der Verbesserungsterm lautet 0.4·(-1+10.167)·(7,9,1)/131 = (0.196, 0.252, 0.028).*

*d) Damit ist die neue Klassengrenze **w**(1) = (-0.97, -0.75, 7.028). Die Koeffizienten der Geraden ergeben sich zu a=-0.97/0.75=-1.3 und b=7.028/0.75=9.4 Die Gerade wird deutlich steiler und schneidet die y-Achse bei 9,4 und die x-Achse bei 7,24. Damit nähert sie sich der exakten Trennline, die eine positive Steigung aufweist.*

AS-1.3 PCA **30 Punkte**

a) Welcher Zweck wird mit der PCA verfolgt? (1P)

Mit PCA sollen die „intrinsischen Variablen“ ermittelt werden, also die Variablen, die das Problem hauptsächlich charakterisieren. Dies sind hier die Richtungen größter Varianz, also größter Signalenergie.

b) Erklären Sie die Vorgehensweise der PCA. (1P)

Dazu wird die Kovarianzmatrix der Signale gebildet und ihre Eigenvektoren und Eigenwerte bestimmt.

c) Geben Sie einen Algorithmus sowie eine dazu benötigte Lerngleichung für die PCA an. (3P)

Beispielsweise kann man den Eigenvektor mit dem größten Eigenwert durch die beschränkte Hebbsche Lernregel

$$\boldsymbol{w}_1(t+1) = \boldsymbol{w}_1(t) + g\, y_1 \boldsymbol{x} \qquad \text{mit } |\boldsymbol{w}_1| = 1,\ y_1 = \boldsymbol{w}_1 \boldsymbol{x}$$

gewinnen. Transformieren wir den Eingaberaum {x} durch $\boldsymbol{x}' = \boldsymbol{x} - y_1 \boldsymbol{w}_1$, so läßt sich aus obiger Gleichung auch der zweite Eigenvektor $\boldsymbol{w}_2$ bestimmen, und so fort.

d) Gegeben sei eine Datenmenge der 2-dim Gestalt wie in Abb. 1.3 gezeigt. Zeichnen Sie die Eigenvektoren darin ein und bezeichnen Sie diese.(1P)

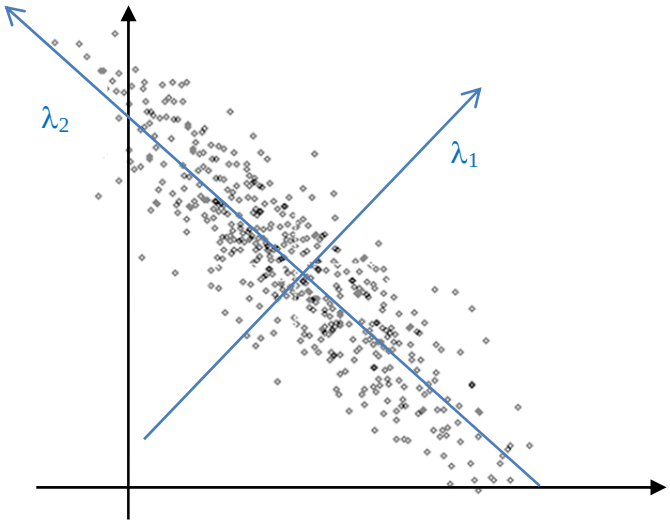


Abb. 1.3

e) Geben Sie den Index der Eigenwerte in absteigender Reihenfolge an. (1P)

λ_2, λ_1

f) Führen Sie eine PCA für die vier Punkte $(3,\sqrt{3})$, $(-\sqrt{3},-3)$, $(-3,-\sqrt{3})$, $(\sqrt{3},3)$ durch. Verwenden Sie dazu entweder die char. Gleichung oder drei Schritte der Oja-Lernregel für jeden der beiden Eigenvektoren. (**20 Pkte**)

Mit dem Mittelwert (0,0) der vier Punkte ergibt sich die Kovarianzmatrix als Autokorrelationsmatrix $A_{ij} = \langle x_i x_j \rangle$, $i,j = 1,2$ zu

A12 = A21 = (4·3√3)/4 = 3√3
A11 = (9+3+9+3)/4 = 6, A22 = (3+9+3+9)/4 = 6
mit dem char.Polynom (6-λ)(6-λ) –27 = 0 = λ² –12λ +9 oder λ₁,₂ = 6 ± 3√3
Die Eigenvektoren sind mit der ersten Komponente x1 der char Gleichung
(6 + 3√3)x₁ = 6x₁ + 3√3 x₂ oder x₁ = x₂ so dass ***e***$_1$ *= (1,1)*
(6 – 3√3)x₁ = 6x₁ + 3√3 x₂ oder x₁ = –x₂ so dass ***e***$_2$ *= (–1,1)*

AS-1.4 Self-Organizing-Maps (SOM) 15 Pkte

Nehmen Sie an, dass Sie 16 RBF-Neuronen im $\Re^2$ haben. Formulieren Sie die SOM-Gleichungen, um die RBF-Neuronen zu trainieren. Verwenden Sie bei der Auswahlregel die RBF-Aktivitäten. Die Nachbarschaftsbeziehung darf dazu frei von Ihnen gewählt werden.

Bei der SOM gilt die Auswahlregel: $|\boldsymbol{w}_c-\boldsymbol{x}| = \min_i |\boldsymbol{w}_i-\boldsymbol{x}|$ *Dies impliziert, dass das dem Eingabemuster x nächste Neuron aktiviert wird. Verwenden wir stattdessen die RBF-Aktivitäten S(wᵢ), so muss die Regel lauten: S(w_c) = max S(wᵢ) da mit w=RBF-Zentrum dies der Definition des RBF-Neurons entspricht.*
Für das Training definieren wir uns ein 1-dim Raster, in dem die RBF-Neuronen angeordnet sind. Damit ist eine Trainingsregel beispielsweise
$w_c(t+1) = w_c(t) +(x-w_c(t))$
und für die Nachbarn
$w_{c-1}(t+1) = w_{c-1}(t) +0{,}5(x-w_{c-1}(t))$
$w_{c+1}(t+1) = w_{c+1}(t) +0{,}5(x-w_{c+1}(t))$
$w(t+1) = w(t)$ *sonstig.*

So werden nur die Nachbarn des gewählten RBF-Neurons trainiert und nicht die anderen.

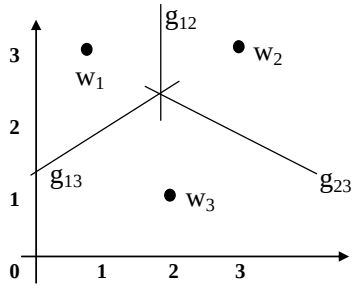
Beginn Teil AS-1+AS-2

AS-2.5 Klassifizierung 40 Pkte

a) Gegeben seien drei Klassenprototypen mit den Koordinaten $w_1 = (1,3)$, $w_2 = (3,3)$ und $w_3 = (2,1)$. Geben Sie die Geradengleichungen an für die drei linearen Klassengrenzen, die jeweils durch $g_{ik} = \{\mathbf{x} / |\mathbf{x}-\mathbf{w}_i|=|\mathbf{x}-\mathbf{w}_k|\}$ zwischen Klasse i und k definiert seien.

Wie lautet die Koordinate des Schnittpunkts aller drei Grenzen? (10 Pkte)

Lösung Aufgabe 1.5 a) ***10 Punkte***



$\boldsymbol{w}_1 = (1,3)^T$, $\boldsymbol{w}_2 = (3,3)^T$, $\boldsymbol{w}_3 = (2,1)^T$

Die Diskriminanzfunktionen sind :

$g_{12} = \{\boldsymbol{x} | \; x_1=2, \; x_2 \in \Re\} = \{\boldsymbol{x} | \; 1 \cdot x_1 + 0 \cdot x_2 - 2 = 0\}$

$g_{13} = \{\boldsymbol{x} | \; x_2 - a_1 x_1 - b_1 = 0\}$:= $\boldsymbol{h}$ *ist senkrecht auf der Verbindungsgerade zwischen* $\boldsymbol{w}_1$ *und* $\boldsymbol{w}_2$, *also* $\boldsymbol{h} \cdot (\boldsymbol{w}_1 - \boldsymbol{w}_2) = 0$, *so dass* $-h_1 + 2h_2 = 0$ *oder* $2h_2 = h_1$. *Dies entspricht einer Steigung von* $a_1 = 1/2$. *Einsetzen vom bekannten Punkt zwischen beiden Zentren auf (1.5, 2) ergibt* $b_1 = 1.25$. *Also ist*

$$g_{13} = \{\boldsymbol{x} | \; x_2 - \tfrac{1}{2} x_1 - 1{,}25 = 0\} = \{\boldsymbol{x} | \; -2x_1 + 4x_2 - 5 = 0\}$$

Das gleiche gilt für g_{23}. *Hier ist* $a_2 = -1/2$ *und* $b_2 = 3{,}25$, *so dass*

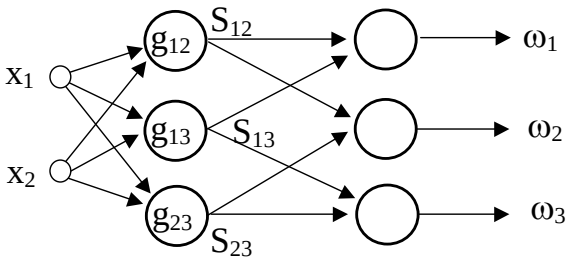
$$g_{23} = \{\boldsymbol{x} | \; x_2 + \tfrac{1}{2} x_1 - 3{,}25 = 0\} = \{\boldsymbol{x} | 2x_1 + 4x_2 - 13 = 0\}.$$

Der gemeinsame Schnittpunkt $\boldsymbol{P}$ *ist bei* $g_{12} = g_{13}$ *oder bei* $P_1 = 2$, $P_2 = 2{,}25$.

b) Bei der Eingabe von $\mathbf{x} = (x_1, x_2)$ soll als Ausgabe der Index der Klasse aus Aufgabenteil a), in dessen Bereich $\mathbf{x}$ fällt, mit Hilfe formaler Neuronen ermittelt werden. Verwenden Sie zwei Schichten aus binären Neuronen. Die erste Schicht implementiert die Diskriminanzfunktion. Die zweite Schicht hat als Ausgabe für jede Klasse genau ein Ausgabeneuron, das genau dann 1 wird, wenn die Klasse vorliegt, und sonst null ist. Wie lauten dabei die Gewichte der Neuronen? Nutzen Sie dabei die Ergebnisse aus a). Wie ändert sich das Netz, wenn in der ersten Schicht Distanzneuronen verwendet werden? (15 Pkte)

Lösung Aufgabe 1.5b) ***15 Punkte***

Das neuronale Netz aus zwei Schichten sieht folgendermaßen aus:



Aus den drei Gleichungen für die Grenzen folgen direkt die drei Gewichte jedes Neurons.

g_{12} : $w_1 = +1$, $w_2 = 0$, $w_3 = -2$

g_{13} : $w_1 = -2$, $w_2 = 4$, $w_3 = -5$

g_{23} : $w_1 = +2$, $w_2 = 4$, $w_3 = -13$

Es ergibt sich also folgende Funktionstabelle:

	g_{12}	g_{13}	g_{23}
ω_1	< 0	> 0	*
ω_2	> 0	*	> 0

ω_3	*	< 0	< 0

Die binäre Ausgabe $S_{bin}(\mathbf{w},\mathbf{x})$ der drei Neuronen ist =1, wenn $g(x)\geq 0$ und sonst 0. Sei $S_{ij}=S(g_{ij})$, so bedeutet die obige Tabelle für die binäre Ausgabe

	S_{12}	S_{13}	S_{23}	*Klasse liegt vor, wenn...*
ω_1	0	1	*	$NOT(S_{12})$ AND S_{13}
ω_2	1	*	1	S_{12} AND S_{13}
ω_3	*	0	0	$NOT(S_{13})$ AND $NOT(S_{23})$

Bekanntlich kann man jede logische Funktion mit einem binären Neuron und entsprechenden Gewichten implementieren. Die Gewichte der zweiten Schicht lassen sich direkt aus den obigen logischen Ausdrücken ablesen.

ω_1: $w_1=-1, w_2=+1, w_3=1,$ ω_2: $w_1=+1, w_2=+1, w_3=2,$ ω_3: $w_1=-1, w_2=-1, w_3=0$

Mit Distanzneuronen werden in der ersten Schicht die Eingaben direkt mit den Klassenprototypen verglichen. Das Ergebnis (der Abstand) muss dann in der zweiten Schicht untereinander verglichen werden. Das Ergebnis dieser drei Vergleiche muss dann in einer weiteren Schicht wie oben ausgewertet werden.

c) Man implementiere das Klassifizierungsnetz als Programm und gebe den Pseudocode an. (15 Pkte)

Lösung Aufgabe 1.5 c) ***15 Punkte***

Ein Programm in Pseudocode, das das obige neuronale Netz implementiert, kann folgendermaßen geschrieben sein:

```
Program  Klassifikation;
(* implementiert eine Klassifikation von drei Klassen *)

CONST n:=3; M:=3;
TYPE  Vector =    ARRAY[1..n] OF REAL;
VAR    w1,w2:     ARRAY[1..M]OF Vector; (* weights 1.+2.layer *)
       x1,x2:     Vector;                (* input 1.+2.layer *)

Function S(w,x:Vector): INTEGER;
(* Neuronenfunktion *)
BEGIN Sum:=0.0;
   FOR j:=1 TO n DO               (* Skalarprodukt*)
      Sum:=Sum+(w[j]*x[j])
   END FOR
   IF Sum>=0 THEN S:=1 ELSE S:=0; (* Binäre Ausgabe *)
END

LOOP
   Input x1[1],x1[2]; x1[3]:=1;
   FOR i:=1 TO M DO          (*Aktivität 1.Schicht *)
      x2[i]= S(w1[i],x1);
   ENDFOR
   FOR i:=1 TO M DO          (*Aktivität 2.Schicht *)
      IF S(w2[i],x2)=1 THEN WriteLn(„Klasse“,i,“liegt vor.“);
   ENDFOR
ENDLOOP

BEGIN (*main*)
 w1:=((1,0,–2),(–2,4,–5),(2,4,–13));
 w2:=((1,1,1),(1,1,2),(–1,–1,0))
END
```

Man beachte, dass jede Schicht modular für sich funktioniert und damit Zeitparallelität für alle Neuronen einer Schicht simuliert wird. Die Eingabedaten und die Gewichte sind als Datenstrukturen getrennt, um die sich ändernden und die konstanten Daten zu trennen.

Ende des Teils AS-1+AS-2

Beginn Teil AS-2

AS-2.6 Grundlagen **20 Punkte**

a) Erklären Sie den „Fluch der Dimensionen“ und geben Sie Maßnahmen an, die durchgeführt werden können, um dieses Problem abzumildern. (5P)

Die Komplexität der Lösung eines Problems steigt exponentiell mit der Anzahl der Quellen. Um für einen d dimensionalen Raum eine hinreichend genaue Lösung zu erzielen, sind n^d Beispiele notwendig. Leo Breiman hat n=10 für ein Intervall von 0 bis 1 als hinreichend definiert. Schon für drei Dimensionen wären somit 1.000 Trainingsmuster notwendig. In der Praxis sind meist jedoch deutlich weniger Beispiele anzutreffen, die meist hochdimensionale Daten beinhalten.
Als Lösungen eigenen sich folgende Vorgehensweisen:
** Anzahl der Dimensionen für das zu lösende Problem reduzieren -> z.B. PCA zur Identifizierung der Dimensionen mit dem größten Eigenwert*
** SOMs, um anhand eines Trainingsmusters mehrere Neuronen / Nachbarschaftsbeziehungen zu trainieren*
** Trainingsdaten künstlich erweitern*
** RBF / Fuzzy, um mittels Reduktion von Regelsets die bedeutendsten Regeln und somit Dimensionen zu identifizieren, die für die praktische Anwendung hinreichend sind*

b) Beschreiben Sie das Vorgehen des Backpropagation sowie Vor- und Nachteile gegenüber dem schichtenorientierten Training etwa bei RBF-Netzen. (5P)

Beim Backpropagation-Netzwerk wird ein mehrschichtiges Netzwerk aufgebaut, bei dem die Architektur aus einer Eingabeschicht und einer Ausgabeschicht sowie mindestens einer Zwischenschicht, dem hiddenlayer, besteht. Die Eingabe wird vom hiddenlayer verarbeitet, der daraufhin eine Aktivität berechnet und diese an die Ausgabeschicht weiter leitet. Die Ausgabeschicht erzeugt daraufhin eine Ausgabeaktivität. Die Ausgabeaktivität wird mit der gewünschten Ausgabeaktivität verglichen und ein Fehler für die Ausgabeschicht berechnet. Anschleißend wird für den hiddenlayer ein Fehler berechnet und zurück propagiert. In der Vorlesung wurde der stochastische Gradientenabstieg vorgestellt.

***Vorteile**: Bei hinreichend vielen Trainingsdaten kann jede beliebige Funktion mit den zwei Schichten (hiddenlayer und Ausgabeschicht) erlernt werden. Auch eine komplexere Anzahl an Klassen (zur Erinnerung: NETtalk hat über 18.000 Gewichte) lassen sich recht schnell lernen mit wenig Aufwand lernen.*

***Nachteile**: Backpropagation neigt u.a. bei wachsender Anzahl an Neuronen im hiddenlayer zu Overfitting. Bei Eingaben mit sehr vielen Dimensionen ist die Konvergenz von Backpropagation sehr langsam. RBF Netze sind hierfür besser geeignet. Auch ist für das Clustern von unbekannten Eingabedaten ein Backpropagation-Netz nicht geeignet. Eine Lehrervorgabe zur Berechnung des Fehlers wird erwartet. Ein Backpropagation Netzwerk kann nicht ohne weiteres dynamisch erweitert werden.*

c) Woran liegt es, dass die Reihenfolge der Quellen bei der ICA unbestimmt ist? (2P)

Das Ziel der ICA ist die Unabhängigkeit der Ausgabekomponenten mit $P(\mathbf{y}) = P(y_1) \cdot P(y_2) \cdots P(y_n)$. Da das Produkt kommutativ ist, ist jede Permutation der Produktreihgenfolge auch eine Lösung. Damit ist eine eindeutige Reihenfolge der Komponenten nicht möglich.

Was wird bei der ICA im Gegensatz zur PCA bestimmt? (1P)

Die PCA (Dekorrelation) ermittelt die Richtung der größten Varianz. Die ICA (Unabhängigkeitsanalyse) ermittelt die Richtungen unabhängiger Variablen/Ausgaben.

d) Warum wird bei der ICA die Varianz jeder Quelle mit $\sigma=1$ vorgegeben? Warum kann man die Varianz nicht bestimmen? (1P)

Aus der beobachteten Varianz der Mischung kann die Varianz einer Quelle nicht eindeutig bestimmt werden. Deshalb ist per Definition die Varianz aller entmischten Quellen 1 (Einheitsmatrix mit allen Hauptdiagonaleinträgen gleich eins). //siehe auch S.162f. im Skript

e) Was ist der Unterschied zwischen stochastischer Abhängigkeit und Korrelation? (2P)

Durch die Korrelation wird die Beziehung zwischen Merkmalen beschrieben. Merkmale können stochastische Abhängigkeit aufweisen, was bedeutet, dass das Auftreten eines Merkmals die Wahrscheinlichkeit des Auftretens eines anderen Merkmaltyps beeinflusst. Sind Merkmale stochastisch unabhängig, so sind sie dekorreliert.
(Andere Antwortmöglichkeit: Beide Begrifflichkeiten untersuchen den gleichen Sachverhalt, aber aus Abhängigkeit folgt Korrelation, was umgekehrt nicht gilt.)

f) Wann ist ein Fixpunkt stabil und wann ist er labil? Woran kann man das erkennen und wie lautet die Bedingung dafür? (2P)

Ein Fixpunkt ist stabil, wenn gilt: $|f'(x)| \leq 1$ mit $x(t+1) = f(x(t))$
Ein Fixpunkt ist instabil, wenn gilt: $|f'(x)| > 1$

g) Was ist eine Ljapunov-Funktion und wozu wird sie verwendet? (2P)

Bei einer Ljapunov-Funktion handelt es sich um eine Funktion, für die für alle t gilt: $f(t+1) \leq f(t)$ und es ex. eine untere Schranke $f_{min} \leq f(t)$. Diese Funktion wurde in der Vorlesung z.B. benutzt, um die Konvergenz einer Lerngleichung (S.68 im Skript) herzuleiten.

AS-2.7 Lagrangeoptimierung 10 Punkte

Gegeben seien *n* Wahrscheinlichkeiten P_i für *n* Zustände X eines Systems. Welche Wahrscheinlichkeiten müssen die *n* Einzelwahrscheinlichkeiten annehmen für eine maximale Entropie ?

$$H(X) = -\sum_i P_i(x) \log P_i(x)$$

Verwenden Sie für die Maximierung die Lagrange-Funktion und denken Sie an die Nebenbedingung für die Wahrscheinlichkeiten.

Lösung: *Die maximale Entropie wird angenommen, wenn sowohl H maximal ist als auch die Summe aller P_i gleich eins ist. Es lässt sich also die Lagrangefunktion aufstellen*

$$L(H, \mu) = -\sum_i P_i(x) \log P_i(x) + \mu \left(\sum_i P_i - 1\right)$$

und die Bedingung für maximales L ist

$$\partial L/\partial P_i = 0 = \partial/\partial P_i \ (P_i \log P_i + \mu P_i) = \log P_i + 1 + \mu = 0 \text{ oder } \mu = -\log P_i - 1$$

Dies gilt auch für ein beliebig anderes P_k, so dass auch hier gilt $\mu = -\log P_k - 1$

Also ist $-\log P_i - 1 = \mu = -\log P_k - 1$

oder $P_i = P_k$

Alle Wahrscheinlichkeiten sind gleich groß; die uniforme Verteilung hat maximale mittlere Information.

AS-2.8 PCA 15 Punkte

Führen Sie die PCA für folgende drei Punkte mittels der Fixpunktgleichung durch: (1,0), (2,2), (3,1).
Führen Sie dabei die Iteration dreimal für den ersten für den zweiten Eigenvektor. Nehmen Sie für w den Startwert (0,1) an. Wie wird der zweite Eigenvektor aussehen?

Erwartungswert: $d_1=2$, $d_2=1$
- ⇨ *Punkte‘: (-1,-1), (0,1), (1,0)*
- ⇨ *Varianz: $d_1=2/3$, $d_2=2/3$*
- ⇨ *Punkte“: (-1.2247,- 1.2247), (0, 1.2247), (1.2247,0)*
- ⇨ *Covarianzmatrix: $C_{11}=1$, $C_{12}=0.5$, $C_{21}=0.5$, $C_{22}=1$*
- ⇨ *1. w(1) = (0.4472, 0.8944)*
- ⇨ *2. w(2) = (0.6247, 0.7809)*
- ⇨ *3. w(3) = (0.6805, 0.7328)*

Der zweite Eigenvektor muss orthogonal auf dem ersten stehen und ist somit beispielsweise (-0.7328, 0.6805).

MatrikelNr:

AS-2.9 ROC-Kurven 5 Punkte

a) Was ist eine ROC-Kurve? (1P)

Die ROC-Kurve visualisiert die Abhängigkeit von Sensivität und Spezifität eines Klassifizierers.

b) Erklären Sie mit kurzen Stichworten, wie man die ROC-Kurve für ein gegebenes Diagnosesystem D(x) ermittelt. (1P)

Seien die auftretenden Ereignisse x mit der Diagnose D(x) in ihrer Art bestimmt, so erhält man für alle Ereignisse X = {x} je einen Wert für das Tupel S = (Sensitivität, Spezifität). Da D(x) = D(p,x) mindestens von einem Parameter p abhängig ist, kann man durch Veränderung von p für jeden Wert von p auch ein davon abhängenden Wert S für die Diagnose D(X) erhalten. Die Menge aller Ergebnispunkte {S} von unterschiedlichen Parameterwerten bildet approximiert eine Kurve, die ROC-Kurve.

c) Wie sieht die best-möglichste ROC-Kurve eines Diagnosesystems aus? (1P)

Die ideale ROC-Kurve ist ein Rechteck (sofern das Koordinatensystem gemäß der Vorlesung verwendet wird).

d) In welchem Fall sagt die ROC-Kurve, dass ein System ungeeignet für die Klassifikation von Mustern ist? (1P)

Wenn die ROC-Kurve eine Diagonale ist, was gleichbedeutend damit ist, dass ein Muster mit 50%iger Wahrscheinlichkeit der richtigen Klasse zugeordnet wird. Dies entspricht dem Zufall. (Weiterhin darf die Erkennung nicht unter 50% liegen, aber dann könnte die Entscheidung umgedreht werden, damit eine Wahrscheinlichkeit über 50% erreicht wird)

e) Wenn zwei Systeme verglichen werden und für System 1 ist bekannt, dass die Wahrscheinlichkeit für die korrekte Klassifikation p_1 ist mit $p_1 > p_2$. Wann ist das zweite System dennoch besser als das erste? (1P)

Beispielsweise, wenn der AUC Wert des zweiten Systems höher ist als der des ersten Systems.

MatrikelNr:

AS-2.10 Bayes-Klassifikation 10 Punkte

Sei eine Datenquelle gegeben, die vier Muster $x_1, .. ,x_4$ aus drei Klassen ω_1, ω_2 ω_3 hervorbringt. Die Wahrscheinlichkeiten sind in der Tabelle aufgeführt.

$P(x_i\|\omega_k)$	ω_1	ω_2	ω_3	$P(x_i)$
x1	0,3	0,7	0	0,41
x2	0,2	0,1	0,3	0,18
x3	0,2	0,2	0,3	0,23
x4	0,3	0	0,4	0,18
$P(\omega_k)$	0,2	0,5	0,3	

Wie lautet der jeweils beste Klassifizierer für $x_1, .. ,x_4$?

***Lösung**: Es ergibt sich folgende Tabelle für die Likelihood-Werte $P(\omega_i|x_j)$:*

P(wi|xj)	w1	w2	w3
x1	0,15	0,85	0,00
x2	0,22	0,28	0,50
x3	0,17	0,43	0,39
x4	0,33	0,00	0,67

Also ist der günstigste Klassifizierer für x_1 und x_3 die Klasse 2 und für x_2 sowie x_4 die Klasse 3. Die Klasse 1 sollte nie gewählt werden.