1. Übungsblatt

Aufgabe 1 Vektoren und Winkel

Berechnen Sie den Betrag des Vektors sowie die Winkel zwischen Vektor und Koordinatenachsen:

a) $\mathbf{a} = (6, -2, -3)$,

b) $\mathbf{b} = (-2, 11, -10)$,

c) $\mathbf{c} = \overline{AB}$ mit $A(1, -2, -3)$ und $B(4, 2, 9)$.

Aufgabe 2 Punkte auf einer Ebene

Eine Ebene durch den Punkt P mit dem Ortsvektor $\mathbf{c}$, die senkrecht zum Vektor $\mathbf{n}$ verläuft, hat die Gleichung $(\mathbf{r} - \mathbf{r}_0)^T\mathbf{n} = 0$. Gegeben sind $\mathbf{r}_0 = (1, -1, 2)$ und $\mathbf{n} = (1, 2, -3)$. Welche der folgenden Punkte liegen in der Ebene? $P_1(-2, -1, 1), P_2(1, -1, 2), P_3(2, -2, 1)$

Aufgabe 3 Abstand eines Punktes von einer Ebene

Ermitteln Sie eine Formel zur Berechnung des Abstandes eines Punktes P mit dem Ortsvektor $\mathbf{c}$ von einer Ebene $(\mathbf{r} - \mathbf{a})^T\mathbf{b} = 0$. Berechnen Sie mit dieser Formel den Abstand für folgendes Zahlenbeispiel: $\mathbf{a} = (-1, -1, -1), \mathbf{b} = (4, -2, 3), \mathbf{c} = (3, 14, -6)$.

Aufgabe 4 Hesse'sche Normalform

Bringen Sie folgende Lineargleichungen auf die Hesse'sche Normalform $\mathbf{r} \cdot \mathbf{n}_0 - d = 0$:

a) $3x - 4y - 20 = 0$,

b) $x + y + 3 = 0$,

c) $y = mx + n$ mit $n < 0$.

Aufgabe 5 Abstände zu Geraden

a) Welchen Abstand hat der Ursprung von der Geraden $12x - 5y + 39 = 0$?

b) Welchen Abstand hat $P_1(4, 3)$ von der Geraden, welche die Koordinatenachsen bei $x = \frac{10}{3}$ und $y = 2.5$ schneidet?

c) Welchen Abstand haben die Parallelen $2x - 3y = 6$ und $4x - 6y = 25$ voneinander?

Aufgabe 6 Ebenengleichung

Welche Ebene durch $\mathbf{r}_0 = (-3, 0, 2)$ ist senkrecht zur Geraden $\mathbf{r} = (-1, -2, 0) + t(1, 1, -1)$ für $t \in \mathbb{R}$?

Aufgabe 7 Gradientenverfahren

Suchen Sie das Minimum der Funktion $f(x_1, x_2) = 2x_1^2 - 2x_1 + x_2^2 - x_2$ mithilfe des Gradientenverfahrens. Nutzen Sie die Anfangsnäherung $(x_1, x_2) = (0, 0)$ und die Schrittweite $\gamma = 0.2$.

2. Übungsblatt

Aufgabe 8 Schwellenwertelemente/Perzeptren

Bestimmen Sie die Parameter von Schwellenwertelementen, sodass diese die folgenden Boole'schen Funktionen berechnen:

a) $y = x_1 \vee x_2$,

b) $y = \neg x_1 \wedge x_2$.

Hinweis: Ein Schwellenwertelement berechnet, auf welcher Seite einer (Hyper-)Ebene ein Eingabevektor liegt.

Aufgabe 9 Schwellenwertelemente/Perzeptren

Versuchen Sie die Parameter eines Schwellenwertelementes zu bestimmen, sodass es das exklusive Oder (geschrieben $x_1 \dot{\vee} x_2$ oder $x_1 \oplus x_2$) berechnet! Welches Problem tritt auf? Wie kann man dieses Problem lösen?

Hinweis: Erinnern sie sich an die in der Vorlesung behandelte Lösung des Biimplikationsproblems.

Aufgabe 10 Schwellenwertelemente/Perzeptren

Bestimmen Sie die Parameter von *einzelnen* Schwellenwertelementen, sodass diese die folgenden Boole'schen Funktionen berechnen:

a) $y = x_1 \wedge \neg x_2 \wedge x_3$ (oder kurz: $x_1\overline{x_2}x_3$)

b) $y = (x_1 \wedge \neg x_2) \vee (x_1 \wedge x_3)$ (oder kurz: $x_1\overline{x_2} \vee x_1x_3$)

c) $y = (x_1 \wedge x_2) \vee (\neg x_2 \wedge x_3)$ (oder kurz: $x_1x_2 \vee \overline{x_2}x_3$)

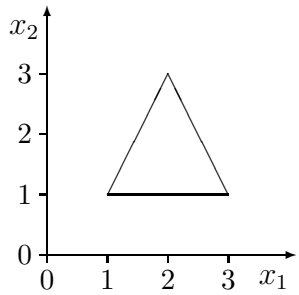
d) $y = (x_1 \wedge x_2) \vee \neg x_3$ (oder kurz: $x_1x_2 \vee \overline{x_3}$)

Hinweis: Ein Schwellenwertelement berechnet, auf welcher Seite einer (Hyper-)Ebene ein Eingabevektor liegt.

3. Übungsblatt

Aufgabe 11 Netze von Schwellenwertelementen

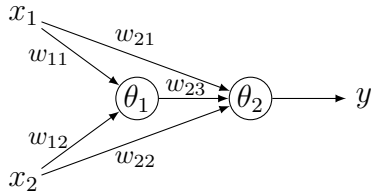
Geben Sie ein neuronales Netz aus Schwellenwertelementen an, das für Punkte (x_1, x_2) innerhalb des in der nebenstehenden Skizze gezeigten Dreiecks den Wert 1 und für Punkte außerhalb den Wert 0 liefert!



Hinweis: Erinnern Sie sich an das in der Vorlesung behandelte neuronale Netz zur Lösung des Biimplikationsproblems und interpretieren Sie die Berechnungen der Schwellenwertelemente der ersten Schicht als eine Koordinatentransformation.

Aufgabe 12 Netze von Schwellenwertelementen

Bestimmen Sie die Parameter w_{ji} und θ_j des in der nebenstehenden Skizze gezeigten neuronalen Netzes, sodass dieses Netz das exklusive Oder der Boole'schen Variablen x_1 und x_2 berechnet (d.h. $y = x_1 \dot{\vee} x_2$ bzw. $y = x_1 \oplus x_2$)!



Hinweis: Gehen Sie von einer geometrischen Interpretation der Berechnung im Eingaberaum des rechten Neurons aus und überlegen Sie, wie Sie die Ausgabe des linken Neurons verwenden können, um die Punkte (x_1, x_2), für die 1 bzw. 0 geliefert werden soll, so anzuordnen, dass sie durch eine Ebene trennbar werden.

Aufgabe 13 Darstellung Boole'scher Funktionen

Geben Sie einen Algorithmus an, der zu einer beliebigen gegebenen Boole'schen Funktion $y = f(x_1, \ldots, x_n)$ ein neuronales Netz aus Schwellenwertelementen mit *nur zwei Schichten* liefert, das diese Funktion berechnet! Das neuronale Netz soll konstruiert, nicht durch Beispiele trainiert werden!

Hinweis: Boole'sche Funktionen schreibt man oft als bestimmte Normalformen.

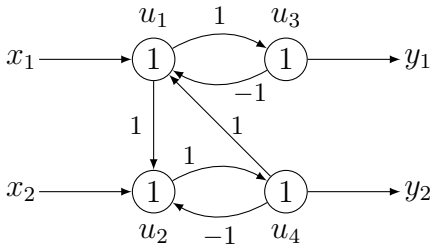
Aufgabe 14 Trainieren von Schwellenwertelementen

Geben Sie den Ablauf des Lernvorgangs (Delta-Regel) eines Schwellenwertelementes für die Boole'sche Funktion $x_1 \rightarrow x_2$ an! Am besten mit Hilfe einer Tabelle, die Spalten für die Werte von x_1, x_2, $d = x_1 \rightarrow x_2$, $\mathbf{x} \cdot \mathbf{w}$, y, e, $\Delta\theta$, Δw_1, Δw_2, θ, w_1 und w_2 enthält. Verwenden Sie als Anfangsbelegung des (erweiterten) Gewichtsvektors $\mathbf{w} = (0, 0, 0)$ und als Lernrate 1. Geben Sie eine geometrische Interpretation des Lernergebnisses an!

4. Übungsblatt

Aufgabe 15 Aktualisierungsreihenfolge

Gegeben sei das folgende Netz aus Schwellenwertelementen:



Zeigen Sie, dass es von der Aktivierungsreihenfolge der Schwellenwertelemente abhängt, ob das Netz in einen stabilen Zustand gelangt, wenn die Eingaben $x_1 = 0$ und $x_2 = 1$ angelegt werden!

Aufgabe 16 Funktionsapproximation

a) Geben Sie ein mehrschichtiges Perzeptron mit ca. 10 Neuronen an, das die Funktion $y = x^2$ im Intervall $[0.5, 4.5]$ durch eine Treppenfunktion annähert.

b) Wie kann man diese Näherung verbessern? (Geben Sie zwei Möglichkeiten an.)

Aufgabe 17 Funktionsapproximation

Wir betrachten die Indikatorfunktion der rationalen Zahlen über der Menge der reellen Zahlen (auch als Dirichlet-Funktion bekannt), d.h. die Funktion

$$f : \mathbb{R} \to \{0, 1\}, \qquad x \mapsto \begin{cases} 1, & \text{falls } x \in \mathbb{Q}, \\ 0, & \text{sonst.} \end{cases}$$

a) Kann diese Funktion durch ein neuronales Netz (mehrschichtiges Perzeptron) beliebig genau angenähert werden?

b) Was zeigt das Ergebnis der Teilaufgabe a) über die Berechnungsfähigkeiten neuronaler Netze?

5. Übungsblatt

Aufgabe 18 Methode der kleinsten Quadrate/Regression

Bestimmen Sie Ausgleichsgeraden $y = a + bx$ (Regressionsgeraden) für die folgenden beiden Datensätze mit Hilfe der Methode der kleinsten Quadrate:

a) $(-2, 0), (0, 1), (1, 3), (2, 5)$

b) $(-1, 3), (1, 2), (2, 0), (4, -2)$

Zeichnen Sie die Datenpunkte und die Ausgleichsgeraden!

Aufgabe 19 Methode der kleinsten Quadrate/Regression

Bestimmen und zeichnen Sie für den Datensatz der Aufgabe 18 a) eine Ausgleichsparabel $y = a + bx + cx^2$!

Aufgabe 20 Methode der kleinsten Quadrate/Regression

Radioaktive Substanzen zerfallen nach dem Gesetz $N(t) = N_0 e^{-\lambda t}$, wobei t die Zeit, λ ein subtanzabhängiger Zerfallsparameter, N_0 die zu Beginn und $N(t)$ die zum Zeitpunkt t noch vorhandenen Teilchen der radioaktiven Substanz sind. Mit Hilfe eines Geiger-Müller-Zählers werden bei einer kleinen Probe eines radioaktiven Materials die folgenden Anzahlen n der bis zu den Zeitpunkten t ausgesandten α-Teilchen gemessen:

t (in s)	0	30	60	90	120	150	180	210	240
n	0	306	552	655	768	863	901	919	956

Jedes gezählte α-Teilchen zeigt den Zerfall eines Teilchens der radioaktiven Substanz an. Bestimmen Sie die Halbwertszeit der radioaktiven Substanz!

Vorgehen: Legen Sie durch die Datenpunkte eine Ausgleichskurve $n = n_0(1 - e^{a+bt})$!

(Hinweis: Sie müssen dazu eine Transformation finden, durch die das Problem auf die Bestimmung einer Ausgleichsgeraden zurückgeführt wird; $n_0 = 1000$.) Obwohl man so einen Wert für a erhält, der von 0 verschieden ist, wird man $-b$ als Näherungswert für den Zerfallsparameter λ ansehen dürfen, aus dem man die Halbwertszeit leicht berechnen kann.

Aufgabe 21 Logistische Regression

Die folgende Tabelle zeigt die Anzahl der amerikanischen ballistischen Interkontinentalraketen (intercontinental ballistic missiles, ICBMs) in den sechziger Jahren:

Jahr, x	1960	1961	1962	1963	1964	1965	1966	1967	1968	1969
Anzahl, y	18	63	294	424	834	854	904	1054	1054	1054

Finden Sie eine Ausgleichskurve mit Hilfe logistischer Regression ($Y = 1060$)! Zeichnen Sie die Originaldaten und skizzieren Sie die Kurve $y = \frac{1060}{1+e^{a+bx}}$!

6. Übungsblatt

Aufgabe 22 Aktivierungsfunktionen

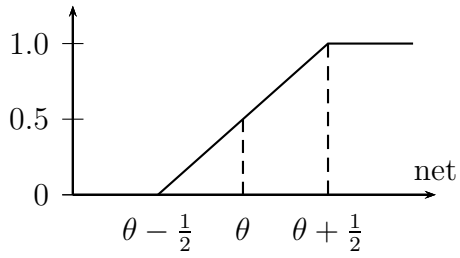
Es ist wichtig, dass mindestens eine Schicht eines mehrschichtigen Perzeptrons nichtlineare Aktivierungsfunktionen benutzt, da sonst seine Fähigkeiten stark eingeschränkt sind: Zeigen Sie, dass ein mehrschichtiges Perzeptron mit beliebig vielen Schichten, in dem die Netzeingabefunktion jedes Neurons die gewichtete Summe seiner Eingänge (abzüglich eines Biaswertes θ) und sowohl die Aktivierungsfunktion als auch die Ausgabefunktion jedes Neurons lineare Funktionen sind, äquivalent ist zu einem zweischichtigen Perzeptron mit den gleichen Eigenschaften!

Hinweis: Die Struktur eines neuronalen Netzes wird manchmal auch, wie in der Vorlesung erwähnt, durch eine Gewichtsmatrix dargestellt.

Aufgabe 23 Gradientenabstieg

Gegeben sei ein zweischichtiges Perzeptron mit n Eingängen und einem Ausgang, in dem das Ausgabeneuron die gewichtete Summe der Eingänge als Netzeingabefunktion, eine semilineare Funktion

$$f_{\text{act}}(\text{net}, \theta) = \begin{cases} 1, & \text{wenn net} > \theta + \frac{1}{2}, \\ 0, & \text{wenn net} < \theta - \frac{1}{2}, \\ (\text{net} - \theta) + \frac{1}{2}, & \text{sonst}, \end{cases}$$



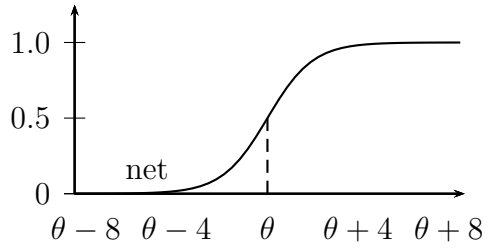
(siehe Zeichnung) als Aktivierungsfunktion und die Identität als Ausgabefunktion besitzt.

Leiten Sie die Änderungsregel für die Gewichte ab, die sich aus einem Ansatz mit Gradientenabstieg ergibt, wenn der Netzfehler als Summe (über alle Muster) der quadrierten Differenzen von gewünschter und tatsächlicher Ausgabe berechnet wird!

Aufgabe 24 Gradientenabstieg

Gegeben sei ein zweischichtiges Perzeptron mit n Eingängen und einem Ausgang, in dem das Ausgabeneuron die gewichtete Summe der Eingänge als Netzeingabefunktion, eine logistische Funktion

$$f_{\text{act}}(\text{net}, \theta) = \frac{1}{1 + e^{-(\text{net} - \theta)}}$$



(siehe Diagramm) als Aktivierungsfunktion und die Identität als Ausgabefunktion besitzt.

Leiten Sie die Änderungsregel für die Gewichte für ein einzelnes Trainingsmuster ab, die sich aus einem Ansatz mit Gradientenabstieg ergibt, wenn der Netzfehler als absoluter Betrag der Differenz von gewünschter und tatsächlicher Ausgabe berechnet wird! Welche Änderungen müsste man (bei mehrschichtigen Perzeptren) an dem Rückpropagationsverfahren vornehmen?

Aufgabe 25 Fehlerrückpropagation

Betrachten Sie ein mehrschichtiges Perzeptron mit 4 Eingängen, 5 Neuronen in der versteckten Schicht und 3 Ausgabeneuronen, das ein 4-dimensionales Dreiklassenproblem lösen soll. Für das Training dieses Netzes soll die Fehlerrückpropagation verwendet werden. Des Weiteren wird angenommen, dass

- die Ausgabeneuronen den *tangens hyperbolicus* und
- die Neuronen der versteckten Schicht die logistische Funktion

als Aktivierungsfunktion benutzen.

a) Definieren Sie für das Netz sinnvolle Ausgabewerte für die Trainingsdaten für jede der drei Klassen.

b) Entwickeln Sie die Formel für die Gewichtsänderung der Verbindung des 3. versteckten Neurons zum 2. Ausgabeneuron. Seien Sie dabei so genau wie möglich.

c) Entwickeln Sie die Formel für die Gewichtsänderung der Verbindung des 1. Eingabeneurons zum 3. versteckten Neuron.

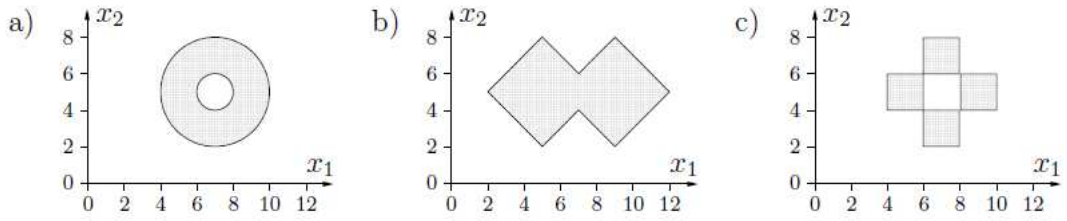
7. Übungsblatt

Aufgabe 26 Radiale-Basisfunktionen-Netze

Bestimmen Sie die Parameter (Gewichte $\mathbf{w}_u$ und Radien σ_u) von Radiale-Basisfunktionen-Netzen mit der Aktivierungsfunktion

$$f_{\text{act}}^{(u)}(\text{net}_u, \sigma_u) = \begin{cases} 1, & \text{wenn } \text{net}_u \leq \sigma_u, \\ 0, & \text{sonst,} \end{cases}$$

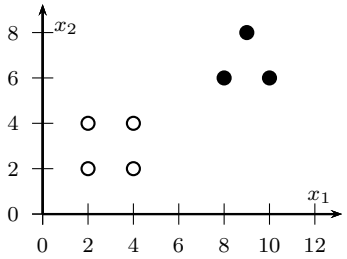
für die Neuronen der versteckten Schicht, die für Punkte innerhalb der grauen Flächen, die in den unten gezeigten Diagrammen dargestellt sind, den Wert 1 und für Punkte außerhalb den Wert 0 liefern! Ob die Netze für Punkte auf den Rändern der Flächen den Wert 0 oder den Wert 1 liefern, ist gleichgültig. Sie sollten jedoch sicherstellen, dass für jeden Punkt der x_1-x_2-Ebene *entweder* der Wert 0 *oder* der Wert 1 berechnet wird.



Aufgabe 27 Radiale-Basisfunktionen-Netze

Die Bestimmung der Gewichte der Verbindungen von den Eingabeneuronen zu den Neuronen der versteckten Schicht — also die Bestimmung der Zentren der radialen Basisfunktionen — und die Bestimmung der Radien gehören zu den Hauptproblemen des Lernens von Radiale-Basisfunktionen-Netzen. Bei Klassifikationsaufgaben verwendet man manchmal statistische Schätzfunktionen, um geeignete (Startwerte für die) Zentren und Radien zu berechnen, jedenfalls dann, wenn zu erwarten ist, dass eine radiale Basisfunktion je Klasse ausreicht. Man fasst dazu die radiale Basisfunktion als skalierte Wahrscheinlichkeitsdichte auf und bestimmt den Erwartungswert und die Standardabweichung der Verteilung z.B. mit einer Maximum-Likelihood-Schätzung.

Als Beispiel betrachten wir ein Radiale-Basisfunktionen-Netz mit zwei Eingängen, zwei versteckten Neuronen und zwei Ausgabeneuronen, das den rechts gezeigten Datensatz klassifizieren soll. Die versteckten Neuronen mögen den Euklidischen Abstand als Netzeingabefunktion und die Gaußfunktion $f_{\text{act}}(\text{net}, \sigma) = e^{-\frac{\text{net}^2}{2\sigma^2}}$ als Aktivierungsfunktion verwenden.



Bestimmen Sie geeignete Zentren $\mathbf{w}$ und Radien σ für die beiden Klassen mit Hilfe einer Maximum-Likelihood-Schätzung! Was müssen Sie bei der Bestimmung der Radien beachten?

Aufgabe 28 Funktionsapproximation

a) Geben Sie ein Radiale-Basisfunktionen-Netz mit ca. 10 Neuronen an, das die Funktion $y = x^2$ im Intervall $[0.5, 4.5]$ durch eine Treppenfunktion annähert.
b) Wie kann man diese Näherung verbessern? (Geben Sie zwei Möglichkeiten an.)

8. Übungsblatt

Aufgabe 29 Fehlerrückpropagation

a) Implementieren Sie die Fehlerrückpropagation (die generalisierte Delta/Regel) für ein mehrschichtiges Perzeptron mit einer versteckten Schicht. Nutzen Sie den *tangens hyperbolicus* als Aktivierungsfunktion. Erweitern Sie Ihre Implementation um die Möglichkeit, einen Momentterm für das Training zu verwenden. Beachten Sie, dass sämtliche Variablen parametrisierbar sein sollten.

b) Testen Sie Ihre Implementierung des neuronalen Netzes, um das Problem der Biimplikation zu lösen.

c) Experimentieren Sie mit dem 4-dimensionalen Zweiklassenproblem (verfügbar auf der Homepage). Teilen Sie den Datensatz durch zufälliges Auswählen in Trainings- (70%), Test- (15%) und Validierungsdaten (15%). Führen Sie mithilfe der Trainings- und Testdaten wenigstens die folgenden Experimente durch:

- Variieren Sie die Anzahl der versteckten Neuronen, um Aussagen über die Laufzeit und die Genauigkeit treffen zu können. Plotten Sie die Länge der Laufzeit, die Höhen des Trainings- und des Testfehlers in Abhängigkeit von der Anzahl der versteckten Neuronen.
- Variieren Sie die Lernrate und dann das Moment des Momentterms. Plotten Sie die Länge der Laufzeit, die Höhen des Trainings- und des Testfehlers in Abhängigkeit von der Lernrate (mit/ohne Momentterm).
- Wiederholen Sie alle o.g. Experimente mehrmals mit unterschiedlich zufällig gewählten Initialisierungen der Gewichte.
- Plotten Sie den Fehler jeder Lernepoche während des Trainings (gemittelt über die Anzahl der Initialisierungen).

Beantworten Sie abschließend folgende zwei Fragen: Wie viele versteckte Neuronen sind bei welcher Lernrate (mit oder ohne Momentterm) optimal in Bezug auf den Testfehler? Wie groß ist der Fehler Ihres Netzes auf den Validierungsdaten?

Aufgabe 30 Radiale-Basisfunktionen-Netze

Bestimmen Sie die Parameter (Gewichte $\mathbf{w}_u$ und Biaswert θ_u) eines einfachen Radiale-Basisfunktionen-Netzes, das die Implikation $x_1 \rightarrow x_2$ berechnet! Alle Basisfunktionen sollen den Radius $\frac{3}{2}$ haben. Die versteckten Neuronen sollen den Maximumabstand als Netzeingabefunktion und eine Dreiecksfunktion

$$f_{\text{act}}(\text{net}_u, \sigma_u) = \begin{cases} 0, & \text{wenn } \text{net}_u > \sigma_u, \\ 1 - \frac{\text{net}_u}{\sigma_u}, & \text{sonst.} \end{cases}$$

als Aktivierungsfunktion besitzen.

Aufgabe 31 Radiale-Basisfunktionen-Netze

Bestimmen Sie mit Hilfe der Methode der Pseudoinversen die Parameter (Gewichte $\mathbf{w}_u$ und Biaswert θ_u) von Radiale-Basisfunktionen-Netzen, die die Konjunktion $x_1 \wedge x_2$ berechnen! Verwenden Sie

a) zwei radiale Basisfunktionen mit Zentren $(0, 0)$ und $(1, 1)$,
b) eine radiale Basisfunktion mit Zentrum $(1, 1)$.

Alle Basisfunktionen sollen den Radius $\frac{1}{2}$ haben. Die versteckten Neuronen sollen den euklidischen Abstand als Netzeingabefunktion und eine Gauß'sche Aktivierungsfunktion

$$f_{\text{act}}(\text{net}_u, \sigma_u) = e^{-\frac{\text{net}_u^2}{2\sigma_u^2}}$$

besitzen. Berechnen Sie die tatsächlichen Ausgaben der beiden Netze und vergleichen Sie sie mit den gewünschten Ausgaben! Warum erhält man in Teilaufgabe a) eine perfekte Lösung des Lernproblems?

Aufgabe 32 Radiale-Basisfunktionen-Netze

Bestimmen Sie mit Hilfe der Methode der Pseudoinversen die Parameter (Gewichte $\mathbf{w}_u$ und Biaswert θ_u) eines Radiale-Basisfunktionen-Netzes, das das Exklusive Oder $x_1 \dot{\vee} x_2$ (bzw. $x_1 \oplus x_2$) berechnet. Verwenden Sie

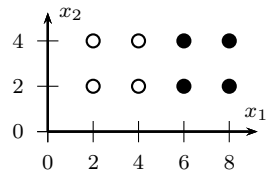
a) zwei radiale Basisfunktionen mit Zentren $(0, 0)$ und $(1, 1)$,
b) eine radiale Basisfunktion mit Zentrum $(1, 1)$.

Alle Basisfunktionen sollen den Radius $\frac{5}{4}$ haben. Die versteckten Neuronen sollen den City-Block-Abstand (auch Manhattanabstand genannt) als Netzeingabefunktion und eine Dreiecksfunktion (siehe Aufgabe 30) als Aktivierungsfunktion besitzen. Berechnen Sie die tatsächlichen Ausgaben der beiden Netze und vergleichen Sie sie mit den gewünschten Ausgaben!

9. Übungsblatt

Aufgabe 33 Wettbewerbslernen / Lernende Vektorquantisierung

Gegeben seien die rechts gezeigten acht Trainingsmuster, die zwei Klassen A (leere Kreise) und B (ausgefüllte Kreise) angehören. Dieser Mustersatz soll mit Hilfe von zwei Referenzvektoren unter Verwendung des euklidischen Abstands quantisiert werden. Welche Endposition werden die Referenzvektoren im Idealfall einnehmen, wenn



a) nur die „Anziehungsregel“ (Muster gleicher Klasse ziehen Referenzvektoren an),

b) sowohl die „Anziehungsregel“ als auch die „Abstoßungsregel“ (Muster anderer Klasse stoßen Referenzvektoren ab)

zur Änderung der Positionen der Referenzvektoren verwendet werden?

Hinweis: Sie brauchen das Verfahren nicht im Detail durchzurechnen. Die Lösung kann direkt aus der Struktur der Trainingsmuster abgelesen werden.

Aufgabe 34 Wettbewerbslernen / Lernende Vektorquantisierung

Das unter der URL `http://www.borgelt.net/lvqd.html` verfügbare Programm visualisiert die lernende Vektorquantisierung für zweidimensionale Daten. (Es können aber auch höherdimensionale Daten geladen werden. Das Programm bietet die Möglichkeit, zwei Eingabevariablen auszuwählen, die verwendet werden sollen — siehe Menüpunkt `Settings > Attributes`.) Wenden Sie dieses Programm auf die Irisdaten[1] mit den Eingabegrößen Blütenblattlänge (petal_length) und Blütenblattbreite (petal_width) an und lassen Sie drei Referenzvektoren bestimmen! Lassen Sie einmal die Klasseninformation berücksichtigen (ein Referenzvektor je Klasse) und einmal nicht. Vergleichen Sie die Ergebnisse!

Aufgabe 35 Selbstorganisierende Karten

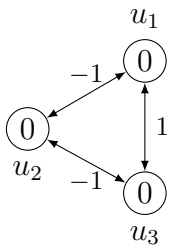
Das unter der URL `http://www.borgelt.net/somd.html` verfügbare Programm visualisiert die Entwicklung einer selbstorganisierenden Karte, die mit zufällig erzeugten Datenpunkten aus einer zweidimensionalen Figur trainiert wird. Führen Sie mit diesem Programm einige Trainingsläufe durch! Variieren Sie dabei die Parameter, speziell die Form der Fläche, aus der die Datenpunkte gewählt werden, die Lernrate und den Radius der Nachbarschaft. Was geschieht, wenn Sie die Lernrate oder den Nachbarschaftsradius zu klein wählen oder zu schnell abnehmen lassen? (Die Lernrate nimmt nach der Formel $\eta(t) = \eta_0 t^{-\kappa}$ ab. Die Werte für η_0 (initial learning rate) und κ (learning rate decay exponent) können in der Parameterdialogbox eingegeben werden. Der Nachbarschaftsradius nimmt nach einem gleichartigen Gesetz ab.)

[1] Die Irisdaten sind auf der Vorlesungsseite verfügbar.

10. Übungsblatt

Aufgabe 36 Hopfield-Netze

Die nebenstehende Abbildung zeigt ein einfaches Hopfield-Netz. Bestimmen Sie für dieses Netz ausgehend vom Anfangszustand $(\text{act}_{u_1}, \text{act}_{u_2}, \text{act}_{u_3}) = (-1, -1, -1)$ den bzw. die Endzustände.



Hinweis: Verwenden Sie einen Zustandsübergangsgraphen. Markieren Sie in diesem Graphen Start- und Endzustand/Endzustände. Beachten Sie, dass es kein explizites Startneuron gibt und somit alle möglichen Folgezustände zu berücksichtigen sind.

Aufgabe 37 Hopfield-Netze

Bestimmen Sie die Energiefunktion des Hopfield-Netzes aus Aufgabe 36! Berechnen Sie mit Hilfe dieser Energiefunktion die Energien der einzelnen Zustände. Ordnen Sie dann die Zustände nach dieser Energie an, d.h., zeichnen Sie einen neuen Zustandsübergangsgraphen, in dem die Position der Zustände ihre Energie angibt.

Aufgabe 38 Hopfield-Netze: Mustererkennung

In einem Hopfield-Netz mit vier Neuronen sollen die beiden Muster $(-1, +1, -1, +1)$ und $(+1, -1, -1, +1)$ gespeichert werden, d.h., diese Muster sollen stabile Zustände des Netzes sein, die durch eine Aktualisierung beliebiger Neuronen nicht verlassen werden.

a) Berechnen Sie die Verbindungsgewichte und die Schwellenwerte der Neuronen eines Hopfield-Netzes, das die genannten Muster speichert!

b) Wie viele weitere Muster können in diesem Netz noch gespeichert werden?

c) Finden Sie zwei weitere Muster, die man zusätzlich in dem von Ihnen konstruierten Netz speichern könnte, ohne dass die alten Muster vergessen werden!
(Um diese Muster tatsächlich zu speichern, müssen natürlich eventuell die Verbindungsgewichte geändert werden.)

Aufgabe 39 Hopfield-Netze: Lösen von Optimierungsproblemen

Gegeben sei eine Folge $F = (a_1, a_2, \ldots, a_n)$ ganzer Zahlen. Wir nehmen vereinfachend an, dass mindestens eine dieser Zahlen nicht negativ ist. Gesucht ist die maximale Teilsumme dieser Folge, d.h. das Maximum der Summen von Teilfolgen der Folge F, wobei wir unter einer Teilfolge der Folge F eine Folge $F_{ij} = (a_i, a_{i+1}, \ldots, a_j)$ mit $1 \leq i \leq j \leq n$ verstehen. Wir suchen also

$$\text{mts}(F) = \max_{1 \leq i \leq j \leq n} \sum_{k=i}^{j} a_k.$$

Konstruieren Sie ein Hopfield-Netz zur Lösung dieses Optimierungsproblems!

Hinweis: Sie müssen eine geeignete Energiefunktion finden.

11. Übungsblatt

Aufgabe 40 Vapnik-Chervonenkis-Dimension

a) Informieren Sie sich über die Vapnik-Chervonenkis-Dimension (VC-Dimension). Was besagt diese Größe und warum spielt sie für das Erlernen einer Funktion aus Datenpunkte (z.B. durch ein neuronales Netz) eine Rolle?

b) Wie groß ist die VC-Dimension einer Hyperebene im Raum $\mathbb{R}^n$? Verdeutlichen Sie Ihre Überlegungen mit Beispielen für $n = 2$.

c) Zeigen Sie, dass die Menge der Funktionen $\{I(\sin(\alpha x) > 0)\}$ die folgenden Punkte im eindimensionalen Raum für beliebige l trennen kann:

$$z_1 = 10^{-1}, \ldots, z_l = 10^{-l}.$$

Das heißt, zeigen Sie, dass die VC-Dimension dieses Klasse von Funktionen unendlich ist.

Aufgabe 41 Optimaler Hyperebenenklassifikator

a) Zeigen Sie, dass für zwei linear-trennbare Klassen von Punkten im Raum $\mathbb{R}^n$ die Maximierung des Abstandes zw. diesen Klassen (engl. *margin*) identisch ist mit der Minimierung der Norm $\|w\|$ des Normalenvektors w der Trennebenen $\{x \in \mathbb{R}^n : x^T w - b = 0\}$.

b) Wie kann dieses Minimierungsproblem angepasst werden, um auch sich überlappende Klassen voneinander zu trennen?

Aufgabe 42 Kernel-Trick

a) Informieren Sie sich über den sogenannten Kernel-Trick. Was besagt er und wo kommt er zum Einsatz?

b) Wie kann das Optimierungsproblem einer linearen SVM angepasst werden, um auch nichtlineare Lösungen zu finden?

Aufgabe 43 Support-Vektor-Maschine

a) Zeigen Sie, dass eine Support-Vector-Maschinen (SVM) mit Gauß-Kern die gleiche Funktion berechnet wie ein RBF-Netz.

b) Zeigen Sie, dass eine SVM mit Sigmoid-Kern einem MLP mit einer versteckter Schicht und einem Ausgabeneuron entspricht.

12. Übungsblatt

Die folgenden Übungsaufgaben werden im Labor G29-035 durchgeführt. Sie können sich gern darauf vorbereiten. Benötigtes Material wird in der Übung verteilt. Machen Sie sich mit dem *Stuttgart Neural Network Simulator (SNNS)* vertraut. Er ist für verschiedene Betriebssysteme unter `http://www.ra.cs.uni-tuebingen.de/SNNS/` erhältlich. Hinweise zum SNNS finden Sie auch auf der Website dieser Lehrveranstaltung.

Aufgabe 44 SNNS: Biimplikation

a) Legen Sie eine SNNS-Musterdatei für die Biimplikation an. (Orientieren Sie sich dabei an der Datei `examples/xor.pat`.) Legen Sie mit dem SNNS ein neuronales Netz mit zwei Neuronen in der Eingabeschicht, zwei Neuronen in der versteckten Schicht und einem Neuron in der Ausgabeschicht an. Initialisieren Sie das Netz mit Zufallswerten (Schalter `init` im `snns-control`-Fenster) und testen Sie es im Einzelschritt (mit dem Schalter `test` im `snns-control`-Fenster). Lassen Sie sich die Verbindungsgewichte und den Schwellenwert anzeigen und deuten Sie diese geometrisch. (Beachten Sie, dass die Neuronen statt einer scharfen Schwellenwertfunktion eine logistische Aktivierungsfunktion verwenden. Sie können die Position der Neuronen im `display`-Fenster verändern, wenn Sie aus den Menüs, die sich bei Drücken der linken Maustaste zusammen mit der Control-Taste öffnen, zuerst `Units` und dann `Move` auswählen.)

b) Wählen Sie als Lernfunktion (Schalter `sel.func.`) `BackpropMomentum`, trainieren Sie das Netz bis sich der Fehler nicht mehr bzw. kaum noch verringert (Fenster `graph`, es kann einige tausend Epochen dauern), lassen Sie sich wieder die Verbindungsgewichte und den Schwellenwert anzeigen und fertigen Sie eine neue graphische Darstellung an. Wiederholen Sie den Lernvorgang mehrmals. Wird immer das gleiche Ergebnis erreicht? Wenn nein, warum nicht?

c) In der Zeile `learn` des `snns-control`-Fensters können Sie Parameter des Lernverfahrens eingeben. Der erste Parameter ist die Lernrate, der zweite ein Momentterm (Genaueres zu diesem Term in der Vorlesung). Experimentieren Sie mit verschiedenen Werten für diese beiden Parameter! Welche Wirkung lässt sich erzielen?

Aufgabe 45 SNNS: MLP zur Analyse der Irisdaten

Die von R. A. Fisher gesammelten Irisdaten sind wahrscheinlich der bekannteste Datensatz im Bereich der Mustererkennung. Er enthält zu jeder der drei Irisarten Iris Setosa, Iris Versicolor und Iris Virginica Beschreibungen von 50 Beispielen, insgesamt also 150 Beispiele. Für jedes Beispiel sind die Länge und Breite der Kelchblätter (sepal length/width, in cm), die Länge und Breite der Blütenblätter (petal length/width, in cm) und die Irisart angegeben. (Passende Eingabedatei namens `iris.train.pat` `iris.test.pat` erhalten Sie während der Übung.)

a) Versuchen Sie, ein neuronales Netz als Klassifikator zu trainieren. D.h. das neuronale Netz soll aus den Eingabedaten die Irisart vorhersagen können.

b) Laden Sie die Trainingsmuster `iris.train` und die Testmuster `iris.test` und trainieren und testen Sie mit diesen das erzeugte neuronale Netz.

c) Experimentieren Sie mit der Zahl der Einheiten in der versteckten Schicht. Welche Fehlerraten lassen sich mit welcher Zahl versteckter Einheiten erzielen?

d) Experimentieren Sie mit dem Weglassen von Eingabewerten. Wählen Sie dazu beim Anlegen des Netzwerks *nicht* `full connection`, sondern legen Sie die Verbindungen per Hand an und lassen Sie dabei einige Eingabeeinheiten einfach unverbunden. (Eine Anleitung finden Sie im SNNS-Handbuch unter "Link Editor".) Auf welche Eingabewerte kann man am leichtesten verzichten?

Aufgabe 46 SNNS: RBF-Netz zur Analyse Irisdaten

a) In Aufgabe 45 haben wir mit dem SNNS ein mehrschichtiges Perzeptron zur Klassifikation der Irisdaten trainiert. Trainieren Sie nun, ein Radiale-Basisfunktionen-Netz für die gleiche Lernaufgabe! Verwenden Sie allerdings eine veränderte Musterdatei, die Sie in der Übung erhalten werden.

b) Lassen sie sich die gelernten radialen Basisfunktionen und die Ausgaben des Netzes im `projection`-Fenster anzeigen. (Sie können auch das gelernte Netz speichern und die radialen Basisfunktionen anhand der Netzparameter skizzieren).

Hinweise: Sie müssen die Aktivierungsfunktionen der Neuronen, die Initialisierungsmethode für die Gewichte und das Lernverfahren ändern. Wählen Sie als Aktivierungsfunktionen für die Neuronen der versteckten Schicht `Act_RBF_Gaussian` und für die Ausgabeneuronen `Act_IdentityPlusBias`. Wählen Sie als Initialisierungsmethode `RBF_Weights` und als Lernverfahren `RadialBasisLearning`. Beachten Sie, dass Sie einige Parameter setzen müssen, z.B. den Biaswert $b = \frac{1}{2\sigma^2}$ der Basisfunktionen im vierten Feld der Zeile `INIT` des `snns-control`-Fensters, sowie die Lernparameter in der Zeile `LEARN` dieses Fensters. Weitere Informationen finden Sie im HTML-Handbuch zum SNNS.

Aufgabe 47 SNNS: SOM zur Analyse der Irisdaten

In dieser Aufgabe wenden wir uns noch einmal den Irisdaten zu, für die wir diesmal mit dem SNNS eine selbstorganisierende Karte trainieren wollen. (Eine passende Eingabedatei erhalten Sie während der Übung.)

a) Legen Sie eine selbstorganisierende Karte mit ca. 10×10 Neuronen in der Ausgabeschicht an (`BIGNET > Kohonen`)! Trainieren Sie diese Karte mit den Irisdaten für einige hundert Epochen (cycles)! Wählen Sie dabei `shuffle` im `snns-control`-Fenster, um die Trainingsmuster zufällig zu mischen. Beachten Sie, dass Sie einige Lernparameter setzen müssen (siehe Abschnitt "Parameters of the Learning Functions" im SNNS-Handbuch). Setzen Sie außerdem als Aktivierungsfunktion der Ausgabeneuronen `Act_Euclid`.

b) Testen Sie die trainierte Karte durch Drücken des Knopfes `test` im `snns-control`-Fenster für die einzelnen Trainingsmuster. (Dazu ist es günstig, die Gitterweite der Anzeige im `snns-display`-Fenster auf 16 Pixel zu setzen und die Beschriftungen abzuschalten.) Wie werden die Eingabemuster den verschiedenen Ausgabeneuronen zugeordnet? (Die ersten

50 Trainingsmuster gehören zur Klasse *Iris Setosa*, die nächsten 50 zur Klasse *Iris Versicolor* und die restlichen 50 zur Klasse *Iris Virginica*.)

c) Initialisieren und trainieren Sie die Karte erneut! Erhalten Sie wieder die gleichen Zuordnungen?