Wintersemester 2002/03 Abgabe am 28.10.2002
Prof. Dr. S. Posch, Dr. A. Gerisch

# 1. Übung „Algorithmen der Bioinformatik II“

**Aufgabe 1.** Zeigen Sie:
(a) Für beliebige Ereignisse $A_1, A_2, \ldots, A_n$ mit $P(A_1, A_2, \ldots, A_{n-1}) > 0$ gilt

$$P(A_1, A_2, \ldots, A_n) = P(A_1)P(A_2|A_1)P(A_3|A_1, A_2) \cdots P(A_n|A_1, A_2, \ldots, A_{n-1}) .$$

(b) Sind $A$ und $B$ unabhängig, so sind auch $A$ und $\overline{B}$ unabhängig.

**Aufgabe 2.** Wir betrachten die Situation eines AIDS-Tests mit empirischen Parametern. Das Ereignis eines positives bzw. negativen Testergebnisses wird mit „$+$“ bzw. „$-$“ bezeichnet. Das Ereignis, dass eine Person HIV-infiziert ist, wird mit „$HIV$“ und das entsprechende Komplementärereignis mit „$\overline{HIV}$“ bezeichnet. Der AIDS-Test liefert bei einer HIV-infizierten Person immer ein positives Testergebnis. Bei nicht HIV-infizierten Personen schlägt der Test mit Wahrscheinlichkeit $0.002$ an. Weiterhin stehen folgende Daten zur Verfügung:

| | männlich und weiblich | nur weiblich |
|---|---|---|
| Einwohnerzahl | $8.2 \cdot 10^7$ | $4.2 \cdot 10^7$ |
| davon HIV-infiziert | $4 \cdot 10^4$ | $8.3 \cdot 10^3$ |

Wie groß ist die Wahrscheinlichkeit, dass eine Person mit positivem Testergebnis nicht HIV-infiziert ist? Wie groß ist diese Wahrscheinlichkeit für weibliche Personen?

**Aufgabe 3.** Beschreiben Sie einen Zufallsvorgang aus dem täglichen Leben oder den Naturwissenschaften, der durch eine normalverteilte Zufallsgröße beschrieben werden kann. Geben Sie benutzte Literaturquellen an.

**Aufgabe 4.** Unter
`http://www.informatik.uni-halle.de/~posch/LEHRE/AdB-II-02/DNASequenz.txt`
ist eine hypothetische DNA-Sequenz $D$ mit 1000 Basen als String gegeben. Jedes Zeichen im String wurde, unabhängig von den anderen Zeichen, mit dem gleichen DNA-Würfel gewürfelt. Dabei hatte jede Seite $X \in \{A, C, G, T\}$ des Würfels die gleiche Wahrscheinlichkeit $p_X = 1/4$. Wir bezeichnen dieses Würfelmodell mit $M_0$.
(a) Bestimmen Sie anhand der ersten $L$ Basen der DNA-Sequenz mögliche Modellparameter $\{p_A^L, p_C^L, p_G^L, p_T^L\}$ durch Auswertung der relativen Häufigkeiten von A, C, G, T für Werte $L \in \{25, 50, 100\}$. Die zugehörigen Modelle seien mit $M_L$ bezeichnet.
(b) Bezeichne $D_n$ die ersten $n$ Basen der gegebenen DNA-Sequenz. Bestimmen Sie für die unter (a) erhaltenen Modelle $M_L$ sowie für das Modell $M_0$ den dekadischen Logarithmus der Likelihood-Werte der Daten $D_n$ für $n \in \{10, 20, \ldots, 90, 100, 200, \ldots, 1000\}$. Die Kurven $\log_{10} P(D_n|M_L)$ sollen als Funktionen von $n$ in ein Diagramm gezeichnet werden. Diskutieren Sie das Diagramm. Warum nimmt $\log_{10} P(D_n|M_L)$ mit wachsendem $n$ ab (bei festem Modell $M_L$)?
(c) Betrachten Sie im folgenden Modelle $M(p_C)$ wobei $p_A = 1/4$, $p_C \in [0, 1/2]$, $p_G = 1/2 - p_C$ und $p_T = 1/4$. Es existiert also ein freier Parameter ($p_C$) im Modell. Wählen Sie ein Gitter für diesen Parameter und berechnen Sie für alle Parameterwerte die Likelihood-Werte $P(D_n|M(p_C))$ für $n \in \{50, 100, 500, 1000\}$. Für jeden Wert $n$ normalisieren Sie die erhaltenen Funktionswerte $P(D_n|M(p_C))$, so dass das Maximum eins beträgt, und zeichnen die normalisierten Werte als Funktion von $p_C$ in ein Diagramm. Diskutieren Sie das Diagramm.

Wintersemester 2002/03 Abgabe am 4.11.2002
Prof. Dr. S. Posch, Dr. A. Gerisch

# 2. Übung „Algorithmen der Bioinformatik II“

**Aufgabe 1.** Zeigen Sie, dass die Familie der Dirichlet-verteilten Zufallsgrößen zur Familie der multinomialverteilten Zufallsgrößen konjugiert ist. Sei also eine Familie von Modellen $M(\theta)$ mit Parametern $\theta \in \mathbb{R}^k$ gegeben, so dass die Priorwahrscheinlichkeiten Dirichlet-verteilt sind mit Parametern $\alpha \in \mathbb{R}$ und $Q = (q_1, q_2, \ldots, q_k) \in \mathbb{R}^k$. Ferner seien die Daten $D = (n_1, n_2, \ldots, n_k) \in \mathbb{N}^k$ gegeben, so dass die Likelihood $P(D|M(\theta))$ multinomialverteilt ist. Nun ist zu zeigen, dass die Posteriorwahrscheinlichkeit $P(M(\theta)|D)$ ebenfalls Dirchlet-verteilt ist (mit neuen Parametern $\beta$ und $R$).

**Aufgabe 2.** Betrachten Sie einen String der Länge $n \in \mathbb{N}$, in dem jeder Buchstabe unabhängig von den anderen und entsprechend der gleichen Verteilung ausgewürfelt wurde.
(a) Sei $B_k$ das Ereignis, dass ein beliebiger, von nun an fester, Buchstabe im String genau $k$-mal auftritt. Welcher Verteilung mit Parameter $p$ gehorchen die zugeordneten Wahrscheinlichkeiten $P(B_k)$, $k = 0, 1, 2, \ldots, n$?
(b) Für den Parameter $p$ aus Teilaufgabe (a) nehmen wir eine stetige Verteilung als Priorwahrscheinlichkeit an, konkret eine Betaverteilung mit den Parametern $(\alpha, \beta)$. Berechnen Sie die Dichte der Posteriorwahrscheinlichkeit. Ist das wiederum die Dichte einer Betaverteilung? Was heißt das in Bezug auf konjugierte Verteilungen?
(c) Was ergibt sich als Maximum-a-posteriori(MAP)-Schätzung für den Parameter $p$?
(d) Was ergibt sich als Mean-a-posteriori(MP)-Schätzung für den Parameter $p$?
(e) Stellen Sie einen Zusammenhang zwischen der Beta- und der Dirichletverteilung her!

**Aufgabe 3.** Seien $D_1$ und $D_2$ unabhängige Beobachtungen eines Versuches, der mit einem Modell $M(p)$ mit Parametervektor $p$ beschrieben wird. Die Datenlikelihood $P(D|M(p))$ und die Priorwahrscheinlichkeit $P(M(p))$ für die Modelle mit Parametern $p$ seien gegeben.
Zeigen Sie, dass die Posteriorwahrscheinlichkeit für die Parameter $p$ nach Beobachtung von $D_1$ und $D_2$ unabhängig davon ist, ob $D_1$ und $D_2$ „gleichzeitig“ in das Modell einbezogen werden oder sukzessive integriert werden (also erst $D_1$ einbeziehen, das führt zu neuem Prior, in welchen dann die Daten $D_2$ einbezogen werden, um schließlich zur gesuchten Posteriorwahrscheinlichkeit für die Parameter zu gelangen).

Wintersemester 2002/03 Abgabe am 11.11.2002
Prof. Dr. S. Posch, Dr. A. Gerisch

# 3. Übung „Algorithmen der Bioinformatik II“

**Aufgabe 1.** Beschäftigen Sie sich etwas mit Matlab (im Windows-Pool installiert) bzw. Octave (auf Linux-Maschinen, z.B. linpc22, installiert). Schreiben Sie eine Matlab-Funktion

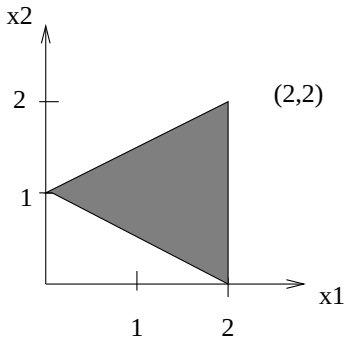
**function** [ prob ] = DirichletDistribution (alpha , Q, P)

die die Dichtefunktion der Dirichlet-Verteilung mit den Parametern $\alpha \in \mathbb{R}$ und $Q \in \mathbb{R}^n$ an der Stelle $P \in \mathbb{R}^n$ auswertet und das Ergebnis zurückgibt. Wie sieht eine Funktion aus, die $P \in \mathbb{R}^{n,k}$, also $k$ Eingaben gleichzeitig, akzeptiert und einen $k$-elementigen Vektor zurückgibt?

**Aufgabe 2.** Drei Schwellwertneuronen erhalten einen identischen zweidimensionalen Eingabevektor $\vec{x} = (x_1, x_2)$. Die Ausgabe des Neurons $i$ ist gegeben durch $y_i = \Theta((1, x_1, x_2) \cdot \vec{w}_i)$, wobei $w_{i0}$ der Schwellwert des $i$-ten Neutrons ist.

1. Bestimmen Sie geeignete Gewichtsvektoren $\vec{w}_i$, so dass genau für Eingabevektoren $\vec{x}$ innerhalb des grauen Dreiecks alle drei Neuronen aktiv sind, für alle anderen Eingaben mindestens ein Neuron inaktiv ist.

2. Fügen Sie ein weiteres Schwellwertneuron hinzu, dessen drei Eingabewerte die Aktivitäten $y_i, i = 1, 2, 3$ sind. Wie muss dessen Gewichtsvektor gewählt werden, damit es genau dann reagiert, wenn der Eingabevektor $\vec{x}$ innerhalb des grauen Dreiecks liegt?




**Aufgabe 3.** Die Dichtefunktion der eindimensionalen Normalverteilung ist gegeben durch

$$p_1(x) = \frac{1}{\sqrt{2\pi}\sigma}\, e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \quad x \in \mathbb{R},$$

wobei $\mu \in \mathbb{R}$ der Erwartungswert und $\sigma^2 \in \mathbb{R}$ die Varianz sind. Dies verallgemeinert sich zur Dichtefunktion der $n$-dimensionalen Normalverteilung

$$p_n(\vec{x}) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}}\, e^{-\frac{1}{2}(\vec{x}-\mu)^\mathsf{T}\Sigma^{-1}(\vec{x}-\mu)}, \quad \vec{x} \in \mathbb{R}^n,$$

mit Erwartungswertvektor $\mu \in \mathbb{R}^n$ und Kovarianzmatrix (symmetrisch und positiv definit) $\Sigma \in \mathbb{R}^{n,n}$.

(a) Wir betrachten den bivariaten Fall ($n = 2$) und schreiben

$$\Sigma = \begin{pmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{12} & \sigma_{22} \end{pmatrix} = \begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix},$$

wobei $\rho := \frac{\sigma_{12}}{\sigma_1\sigma_2}$ der Korrelationskoeffizient ist. Schreiben Sie die Formel für $p_2(\vec{x})$ explizit auf und plotten Sie die Dichtefunktion für $\mu = (1, 1)$ und die folgenden drei Kovarianzmatrizen:

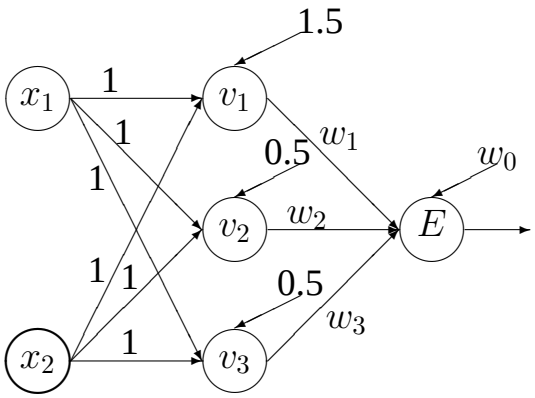
$$\Sigma_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \Sigma_2 = \begin{pmatrix} 1 & 0 \\ 0 & 3 \end{pmatrix}, \quad \Sigma_3 = \begin{pmatrix} 1 & 1 \\ 1 & 3 \end{pmatrix}.$$

(b) In der Vorlesung haben wir diskutiert, dass wir nach Definition einer Fehlerfunktion für die Daten mit deren Hilfe auch eine Likelihood für die Daten definieren können. Welche Fehlerfunktion führt bei dieser Vorgehensweise zur $n$-dimensionalen Normalverteilung? Da $\Sigma$ symmetrisch und positiv definit ist, können wir $\Sigma = V^\mathsf{T}\Lambda V = V^\mathsf{T}\Lambda^{1/2}\Lambda^{1/2}V$ schreiben, wobei $V^\mathsf{T}V = I$ und $\Lambda$ die Diagonalmatrix mit den Eigenwerten von $\Sigma$ ist. Unter Benutzung dieser Zerlegung interpretiere man die Fehlerfunktion, die auch als Mahalanobisabstand bezeichnet wird.

Wintersemester 2002/03 Abgabe am 18.11.2002
Prof. Dr. S. Posch, Dr. A. Gerisch

# 4. Übung „Algorithmen der Bioinformatik II“

**Aufgabe 1.** Betrachten Sie die XOR-Funktion in $x_1$ und $x_2$ und die gegebene Architektur eines Multiple Layer Perceptron.




(a) Welche Funktionen von $x_1$ und $x_2$ berechnen die drei Schwellwertneuronen in der versteckten Schicht?
(b) Betrachten Sie jetzt die Ausgabe dieser drei Neuronen als dreidimensionalen Eingabevektor für das Ausgabeneuron $E$ und veranschaulichen Sie sich die Lage der zu $(x_1, x_2)$ gehörenden Punkte $(v_1, v_2, v_3)$ in $\mathbb{R}^3$.
Geben Sie eine Trennebene an, die diese vier Punkte entsprechend der XOR-Funktion in die Klassen $0$ und $1$ teilt.

**Aufgabe 2.** (a) Implementieren Sie das Gradientenabstiegsverfahren zur Funktionsminimierung einer Funktion $f : \mathbb{R}^n \to \mathbb{R}$ in einer Programmiersprache Ihrer Wahl.
Ausgehend vom Startwert $x_0$ sollen also die Iterierten $x_{k+1} = x_k - \mu_k g(x_k)$, $k = 0, 1, 2, \ldots$, berechnet werden, wobei die Funktion $g$ der Gradient von $f$ sei und $\mu_k$ die Lernschrittweite im Schritt $k$ bezeichne.
Die Iteration soll abgebrochen und die erhaltene Approximation $x_k$ als Minimum akzeptiert werden, wenn die euklidische Norm von $g(x_k) \leq 10^{-3}$ ist.
Die Lernschrittweite soll mit $\mu_0 = 1$ initialisiert werden. Wenn der momentane Wert $\mu_k$ zu $f(x_{k+1}) \geq f(x_k)$ führt, dann wird $x_{k+1}$ nicht akzeptiert und $\mu_k$ wird solange gedrittelt bis die Ungleichung nicht mehr erfüllt ist. Dieses so erhaltene $\mu_k$ wird zur Berechnung von $x_{k+1}$ benutzt. Wir setzen danach $\mu_{k+1} = 2\mu_k$.
(b) Was ist die Gradientenfunktion $g(x)$ zur Funktion

$$f(x) = 1 - e^{-x^{\mathsf{T}} M x}\,, \quad x \in \mathbb{R}^n\,,\ M \in \mathbb{R}^{n,n}\,?$$

(c) Testen Sie Ihre Implementierung an der Funktion $f$ aus Teil (b) mit $M = \begin{pmatrix} 1 & 0 \\ 0 & 10 \end{pmatrix}$. Verwenden Sie dabei folgende Startvektoren: $x_0^1 = (0.6, 0)\,, x_0^2 = (0.6, 0.01)$ und $x_0^3 = (0.6, 0.2)$. Wieviele Iterationsschritte werden bis zum Abbruch benötigt? Was für Werte ergeben sich, wenn die $10$ in $M$ durch eine $20$ ersetzt wird? Gibt es eine Erklärung für die langsame Konvergenz (mal graphisch auf die Folge der Iterierten schauen)?

Wintersemester 2002/03 Abgabe am 25.11.2002
Prof. Dr. S. Posch, Dr. A. Gerisch

# 5. Übung „Algorithmen der Bioinformatik II“

**Aufgabe 1.** In der Vorlesung haben wir gehört, dass ein Schwellwertneuron nur Trenngeraden bzw. Trenn-(hyper)ebenen realisieren kann. Zeigen Sie, dass auch ein Multiple Layer Perceptron mit einer versteckten Schicht mit Neuronen mit linearer Aktivierungsfunktion und einem Schwellwertausgabeneuron nicht mächtiger ist. Was passiert, wenn wir mehrere versteckte Schichten mit linearen Aktivierungsfunktionen haben?

**Aufgabe 2.** Stellen wir uns vor, dass wir die XOR-Funktion per Backpropagation lernen wollen.
(a) Warum können wir nicht die Schwellwertfunktion als Aktivierungsfunktion verwenden?
(b) Wenn wir nun die logistische Aktivierungsfunktion nehmen, sollten wir besser nicht $0$ und $1$ als Zielwerte verwenden. Warum würde das Probleme geben? Was ist ein möglicher Ausweg?

**Aufgabe 3.** Neuronale Netze können jede „vernünftige“ Funktion mit beliebiger Genauigkeit approximieren. Wir wollen das für den Fall einer stetigen Funktion $f : [0,1] \to \mathbb{R}$ beweisen. Wir suchen also ein Neuronales Netz, dass zu gegebener Funktion $f$ und gegebener Genauigkeit $\varepsilon > 0$ für alle Eingaben $x$ eine Ausgabe $y(x)$ erzeugt, so dass $|f(x) - y(x)| \leq \varepsilon$ gilt.
Nach Voraussetzung ist $f$ stetig auf dem kompakten Intervall $[0,1]$ und deshalb gleichmäßig stetig. Das bedeutet, dass es zu jedem $\varepsilon > 0$ ein $n = n(\varepsilon) \in \mathbb{N}$ gibt, so dass für alle $x_1, x_2 \in [0,1]$ gilt

$$|x_1 - x_2| \leq \frac{1}{n} \quad \Rightarrow \quad |f(x_1) - f(x_2)| \leq \varepsilon\,.$$

Wir definieren uns nun eine Treppenfunktion $g(x)$ durch $g(0) = f(0)$ und $g(x) = f(k/n)$ für alle $x \in ((k-1)/n, k/n]$ für alle $k = 1, 2, \ldots, n$.
(a) Zeigen Sie, dass ein neuronales Netz, welches die Funktion $g$ exakt realisiert, auch die Funktion $f$ mit Genauigkeit $\varepsilon$ approximiert.
(b) Geben Sie ein neuronales Netz an, das die Funktion $g$ realisiert. Das Netz soll eine Eingabeeinheit zur Darstellung von $x$ haben, dazu $n+1$ versteckten Schwellwertneuronen, welche jeweils mit der Eingabe verbunden sind, sowie eine Ausgabeeinheit mit linearer Aktivierungsfunktion, die die Ausgaben der versteckten Schwellwertneuronen als Eingabe erhält. Die Gewichte an den Verbindungen von der Eingabe zu den $n+1$ versteckten Einheiten seien $1$. Wie müssen die Schwellwerte in den versteckten Einheiten gewählt werden und wie die Gewichte an den Verbindungen zur Ausgabeeinheit? Zeigen Sie, dass das Netzwerk wirklich die Funktion $g$ realisiert.
(c) Wie müsste das Netzwerk erweitert/modifiziert werden, um stetige Funktionen $f : [0,1] \to \mathbb{R}^m$ mit Genauigkeit $\varepsilon$ zu approximieren? Wählen Sie eine für Sie günstige Norm.
(d) Wie müsste das Netzwerk erweitert/modifiziert werden, um stetige Funktionen $f : [a,b] \subset \mathbb{R} \to \mathbb{R}$ mit Genauigkeit $\varepsilon$ zu approximieren?

Wintersemester 2002/03 Abgabe am 9.12.2002
Prof. Dr. S. Posch, Dr. A. Gerisch

# 6. Übung „Algorithmen der Bioinformatik II“

**Aufgabe 1.** Beschreiben Sie mit Worten und in erläuterten Formeln die Funktionsweise eines Schwellwertneurons (Anzahl und Typ der Eingangs- und Ergebnisgrößen, Form der Aktivierungsfunktion, Sinn und Wirkungsweise des Bias/Schwellwertes). Gibt es Nachteile von Schwellwertneuronen gegenüber Neuronen mit anderer Aktivierungsfunktion?

**Aufgabe 2.** Gewichtsteilung (*weight sharing*) ist eine Technik für die Zuweisung von Gewichten an Neuronen in einem MLP. Dabei werden für bestimmte Gruppen von Neuronen einer Schicht identische Gewichte für die eingehenden Verbindungen gewählt. Diese Technik kann sinnvoll sein, wenn die zu bearbeitenden Probleme durch eine Art von Translationsinvarianz gekennzeichnet sind. Das heißt, für Probleme, für die eine bestimmte Operation, wie zum Beispiel die Extraktion charakteristischer Merkmale, auf verschiedene Bereiche der Eingabe angewendet werden muss. Die resultierenden Netzwerke werden auch als Konvolutionsnetzwerke bezeichnet (*convolutional networks*).
Weight sharing kann gut mit dem Backpropagation-Algorithmus kombiniert werden. Wir gehen von einem voll vernetzten MLP mit $L$ Schichten und $M^l$ Neuronen in Schicht $l$ aus. Schicht $L$ ist die Ausgabeschicht. $w_{ij}^l$ ist das Gewicht der Verbindung vom Neuron $j$ der Schicht $(l-1)$ zum Neuron $i$ der Schicht $l$. Die Eingabe des Neurons $i$ der Schicht $l$ ist $x^l \equiv (x_1^l, x_2^l, \ldots, x_{M^{l-1}}^l) = y^{l-1} \equiv (y_1^{l-1}, y_2^{l-1}, \ldots, y_{M^{l-1}}^{l-1})$ und dessen Ausgabe ist $y_i^l$. Die Aktivierungsfunktion sei $\sigma(v_i^l)$, wobei $v_i^l = \sum_{j=0}^{M^{l-1}} w_{ij}^l x_j^l$ die synaptische Summe der Eingaben ist (mit $x_0^l = -1$ als Eingabe für den Bias $w_{i0}^l$).
Wir betrachten das Netzwerk im Online-Lernmodus mit Daten $D = (\vec{d}, \vec{t}\,)$. Hierbei ist $\vec{t} = (t_1, t_2 \ldots, t_K)$ die gewünschte Ausgabe zur Eingabe $\vec{d}$. Die Fehlerfunktion für das Netzwerk habe die Form $E_{total}(D, W) = \sum_{k=1}^{K} E(y_k^L(\vec{d}, W), t_k)$ , wobei $W$ der Gewichtsvektor ist.
(a) Wir betrachten den Fall des *weight sharing* für Gewichte zwischen der Schicht $L-1$ und der Ausgabeschicht. Sei also $\mathcal{I}_L \subset \{1, 2, \ldots, K\}$ eine Gruppe von Ausgabeneuronen, die ein Gewicht teilen. Es gibt also für alle $i \in \mathcal{I}_L$ ein $j = j(i) \in \{0, 1, \ldots, M^{L-1}\}$, so dass die Gewichte $w_{i,j(i)}^L$ für alle $i \in \mathcal{I}_L$ gleich einem Gewicht $= w_{\mathcal{I}_L}^L$ sind. Mit $W$ bezeichnen wir weiterhin den Gewichtsvektor, in dem alle Gewichte unabhängig sind, und mit $W_{shared}$ denjenigen, den wir durch *weight sharing* erhalten. Zeigen Sie, dass dann gilt

$$\frac{\partial E_{total}(D, W_{shared})}{\partial w_{\mathcal{I}_L}^L} = \sum_{i \in \mathcal{I}_L} \frac{\partial E_{total}(D, W)}{\partial w_{i,j(i)}^L} ,$$

also die Ableitung der Fehlerfunktion nach einem gemeinsam genutzten Gewicht ist gleich der Summe der Ableitungen der Fehlerfunktion nach den (unabhängigen) Gewichten, die gleich dem gemeinsam genutzten sein sollen.
(b) Betrachte nun den Fall, dass Gewichte zwischen den Schichten $L-2$ und $L-1$ gemeinsam genutzt werden (unabhängig davon, ob Gewichte zwischen Schicht $L-1$ und der Ausgabeschicht gemeinsam genutzt werden). $\mathcal{I}_{L-1}$ sei eine Menge von Neuronen der Schicht $L-1$, die ein Gewicht teilen. Wie sieht die Ableitung der Fehlerfunktion $E_{total}(D, W_{shared})$ nach einem solchen gemeinsam genutzten Gewicht $w_{\mathcal{I}_{L-1}}^{L-1}$ aus und wie hängt diese Ableitung mit den Ableitungen der Fehlerfunktion $E_{total}(D, W)$ nach den als unabhängig angenommenen Gewichten zusammen?

Wintersemester 2002/03 Abgabe am 16.12.2002
Prof. Dr. S. Posch, Dr. A. Gerisch

# 7. Übung „Algorithmen der Bioinformatik II“

**Aufgabe 1.** Wir wollen den $\mathbb{R}^n$ durch eine Hyperebene $H$ in zwei Klassen einteilen. Die Hyperebene wird durch $n-1$ linear unabhängige Vektoren $g_1, g_2, \ldots g_{n-1} \in \mathbb{R}^n$ aufgespannt. Sei $x_0 \in H$ ein beliebiger, von nun an fester, Punkt in $H$. Jeder Punkt $x \in \mathbb{R}^n$ in $H$ kann dann durch

$$x = x_0 + \alpha_1 g_1 + \alpha_2 g_2 + \cdots + \alpha_{n-1} g_{n-1}$$

mit reellen Parametern $\alpha_i$ beschrieben werden.
(a) Sei $h \in \mathbb{R}^n$ ein Vektor, der auf allen Vektoren $g_1, g_2, \ldots, g_{n-1}$ senkrecht steht. Zeigen Sie, dass für jeden Punkt $x \in H$ dann gilt $(x - x_0) \cdot h = 0$. Die Umkehrung gilt auch. (Zusatz: Zeigen Sie, dass auch die Umkehrung gilt.)
(b) Wir wollen nun die Klassifikationsaufgabe mit Hilfe eines Schwellwertneurons lösen. Die Schwellwertfunktion $\sigma(v)$ sei gleich Eins für alle $v > 0$ und Null sonst. Gegeben seien der Punkt $x_0$ und der in Teilaufgabe (a) bestimmte Vektor $h$, also die Trennhyperebene in der Form $(x - x_0) \cdot h = 0$, sowie ein Punkt $\tilde{x} \in \mathbb{R}^n$, $\tilde{x} \notin H$, der in die Klasse Eins klassifiziert werden soll. Geben Sie einen Algorithmus an, der aus diesen Daten die Gewichte $w_1, w_2, \ldots, w_n$ und den Bias $w_0$ des Schwellwertneurons berechnet. Zeigen Sie die Korrektheit Ihres Algorithmus. In welche Klasse werden die Punkte der Hyperebene $H$ klassifiziert?
(c) Mit Hilfe des unter (b) entwickelten Algorithmus gebe man die Gewichte und Biaswerte für Schwellwertneuronen an, die anhand folgender Hyperebenen klassifizieren:

$(i)$ $x = \begin{pmatrix}1\\1\end{pmatrix} + \alpha_1 \begin{pmatrix}8\\2\end{pmatrix}$ $(ii)$ $x = \begin{pmatrix}1\\1\\1\end{pmatrix} + \alpha_1 \begin{pmatrix}1\\2\\8\end{pmatrix} + \alpha_2 \begin{pmatrix}2\\8\\5\end{pmatrix}$ $(iii)$ $x = \begin{pmatrix}1\\1\\1\\1\end{pmatrix} + \alpha_1 \begin{pmatrix}2\\1\\5\\9\end{pmatrix} + \alpha_2 \begin{pmatrix}6\\9\\7\\0\end{pmatrix} + \alpha_3 \begin{pmatrix}9\\2\\6\\5\end{pmatrix}$

Der Ursprung des Koordinatensystems soll jeweils nach Klasse Eins klassifiziert werden.

Wintersemester 2002/03 Abgabe am 13. Januar 2003
Prof. Dr. S. Posch, Dr. A. Gerisch

# 8. Übung „Algorithmen der Bioinformatik II“

**Aufgabe 1.** Im MLP-Modell von Brunak, Engelbrecht & Knudsen (siehe Vorlesung) zur Vorhersage von Donor- und Akzeptorstellen in DNA-Sequenzen wird zur Erkennung des Übergangs von kodierenden zu nicht kodierenden Abschnitten folgender Algorithmus angewendet.

> Sei $L$ die Länge der betrachteten DNA-Sequenz und sei $c_i$, $i = 1, 2, \ldots, L$, die Ausgabe des MLPs zur Klassifikation der $i$. Base als kodierend oder nicht kodierend. Ferner sei $K \in \mathbb{N}$ gegeben. Dann wird für jede Stelle $i = K+1, K+2, \ldots, L-K$, der DNA-Sequenz der Wert
>
> $$d_i = \frac{\sum_{j=1}^{K} c_{i+j} - \sum_{j=1}^{K} c_{i-j}}{2K}$$
>
> berechnet. (Für die ersten und letzten $K$ Basen werden die $d_i$ anderweitig berechnet.)

(a) Zeigen Sie, dass die Werte $d_i$ im Fall $K = 1$ als Ableitungen der Werte $c_i$ interpretiert werden können. Dazu betrachte man die $c_i$ als Funktionswerte der 2-mal stetig differenzierbaren Funktion $c(x)$ für $x = i$, also $c_i = c(i)$. Die Funktion $c$ wird also auf einem Gitter mit Gitterweite $h = 1$ ausgewertet und wir können anstelle obiger Formel schreiben

$$d_i = \frac{c(i+h) - c(i-h)}{2h}\,.$$

(Der Quotient heißt auch *zentraler Differenzenquotient 2. Ordnung* von $c$ an der Stelle $i$ mit Gitterweite $h$.) Nun zeige man, dass für feiner werdendes Gitter der Quotient gegen $c'(i)$ strebt:

$$\lim_{h \to 0} \frac{c(i+h) - c(i-h)}{2h} = c'(i)\,.$$

(b) Zeigen Sie, dass die Werte $d_i$ für $K \geq 1$ als Approximation an $\frac{K+1}{2} c'(i)$, also an ein von $K$ abhängiges Vielfaches der Ableitung von $c$ an der Stelle $i$, interpretiert werden können.

**Aufgabe 2.** Zusammenhang zwischen HMMs und dem *Single Die Model*.
(a) Beschreiben Sie ein HMM, das zum unendlichen *Single Die Model* äquivalent ist. Das HMM soll also eine unendlich Folge von Zeichen aus einem Alphabet $X = \{x_1, x_2, \ldots, x_n\}$ ausgeben, wobei jedes Zeichen unabhängig von den anderen gewählt wird und Zeichen $x_i$ mit Wahrscheinlichkeit $p_i \geq 0$ auftritt ($\sum_{i=1}^{n} p_i = 1$).
(b) Beschreiben Sie ein HMM an, das zum endlichen *Single Die Model* äquivalent ist, also nur eine Folge der Länge $N$ als Ausgabe erzeugt. Wie muss dieses HMM modifiziert werden, so dass die Ausgabelänge kleiner oder gleich $N$ ist? Welche neue Parameter müssen für dieses Modell bereitgestellt werden?
(c) Nehmen Sie für das unter (a) beschriebene Modell an, dass $X = \{A, B\}$ mit $p_1 = p \in [0, 1]$ und $p_2 = 1 - p$. Sei $Y$ die Zufallsgröße, dass der Buchstabe $A$ genau $k$-mal hintereinander ausgegeben und dann von einem $B$ gefolgt wird. Wie groß ist die Wahrscheinlichkeit $P_Y(k) \equiv P(Y = k)$, dass also $Y = k$ eintritt, für $k = 0, 1, 2, \ldots$? Geben Sie die Verteilungsfunktion $F_Y(k) \equiv P(Y \leq k)$ für $Y$ an.
(Bemerkung: Wir können $A$ als Symbol für das erfolgreiche Ausführen eines Vorgangs interpretieren und $B$ als fehlerhaftes Ausführen. Das Ausführen des Vorgangs sei immer unabhängig von den bereits ausgeführten Vorgängen. Dann ist $P_Y(k)$ also die Wahrscheinlichkeit dafür, dass wir den Vorgang $k$-mal erfolgreich Ausführen und danach sofort ein Fehler auftritt.)

Wintersemester 2002/03 Abgabe am 20. Januar 2003
Prof. Dr. S. Posch, Dr. A. Gerisch

# 9. Übung „Algorithmen der Bioinformatik II“

**Aufgabe 1.** Finden Sie mit Hilfe der Methode der Lagrangeschen Multiplikatoren alle stationären Punkte der Funktion $f(x, y, z, t) = x + y + z + t$ unter der Nebenbedingung $g(x, y, z, t) \equiv xyzt - c^4 = 0$. Der Wertebereich der Variablen $x, y, z, t$ sei dabei auf $x, y, z, t > 0$ beschränkt.

**Aufgabe 2.**
(a) Was sollen die Worte „hidden“ und „Markov“ im Term *Hidden Markov Modells* ausdrücken? Was heißt, ein System besitzt die Markov-Eigenschaft?
(b) Die Menge der Zustände eines HMM sei $\mathcal{S} = \{S_1, S_2, \ldots, S_N\}$. Die Übergangsmatrix $T \in \mathbb{R}^{N,N}$ enthalte Einträge $t_{S_j,S_i}$, die die Übergangswahrscheinlichkeit vom Zustand $S_i$ zum Zustand $S_j$ des HMMs angeben. Welche Normierungsbedingungen und Beschränkungen müssen die Elemente von $T$ erfüllen?
(c) Sei $\mathcal{E} = \{E_1, E_2, \ldots, E_M\} \subset \mathcal{S}$ die Teilmenge der emittierenden Zustände eines HMM und $\Sigma$ die Menge der Ausgabesymbole (Alphabet). Es gebe $e_{E_i,X}$ die Ausgabewahrscheinlichkeit des Symbols $X \in \Sigma$ im Zustand $E_i$ an. Welche Normierungsbedingungen und Beschränkungen müssen die Elemente von $E \in \mathbb{R}^{M,|\Sigma|}$ erfüllen?

**Aufgabe 3.** Wiederholen Sie die Standardarchitektur für HMM’s für biologische Anwendungen, die den linearen Aspekten von Sequenzen (oft) gut angepaßt ist. Warum spricht man hier von einer Links-Rechts-Architektur? Wieviele Wahrscheinlichkeitsparameter hat die Standardarchitektur mit $K$ Hauptzuständen. Geben Sie die Anzahl getrennt nach Emissions- und Übergangsparametern als Funktion von $K$ an (von $K$ unabhängige, also konstante, Terme können vernachlässigt werden).

**Aufgabe 4.** Zeigen Sie, dass die Größen $\alpha_i(t)$ im Forward-Algorithmus für die Standardarchitektur wohldefiniert sind. Dazu überlege man sich zuerst, wie die Werte $\alpha_i(0)$ für alle Zustände $S_i \in \mathcal{S}$ zu wählen sind. Hier ist eine Unterscheidung von emittierenden Zuständen $S_i \in \mathcal{E} \subset \mathcal{S}$ und nichtemittierenden Zuständen $S_i \notin \mathcal{E}$ nötig. Nun muss gezeigt werden, dass aus den Werten $\alpha_i(t)$ und den Parametern des HMM die Werte $\alpha_i(t + 1)$ berechnet werden können (das heißt, die Rekursion ist wohldefiniert). Überlegen Sie sich dies zuerst wieder für emittierende Zustände $S_i \in \mathcal{E}$. Danach betrachtet man die nichtemittierenden Zustände. Stumme Pfade zwischen welchen Zuständen werden in der Rekursion benötigt? Wieviele gibt es davon und wie berechnet sich ihre Wahrscheinlichkeit? Mit diesem Wissen kann jetzt auch die Wohldefiniertheit der $\alpha_i(t + 1)$ mit $S_i \notin \mathcal{E}$ gezeigt werden.

Wintersemester 2002/03 Abgabe am 27. Januar 2002
Prof. Dr. S. Posch, Dr. A. Gerisch

# 10. Übung „Algorithmen der Bioinformatik II“

**Aufgabe 1.** Sei $O$ eine beliebige Ausgabe des HMMs $M(W)$ mit Parametern $W$. Der wahrscheinlichste Pfad im HMM zur Erzeugung von $O$ kann mit dem Viterbi-Algorithmus berechnet werden.
Ein Pfad $\pi_i(t)$ heißt *Präfixpfad* von $O$, wenn der Pfad im Zustand $S_i \in \mathcal{S}$ endet und die Zeichen $o_1 o_2 \cdots o_t$ emittiert werden. Wir definieren $\delta_i(t)$ als die Wkt. für den wahrscheinlichsten Pfad im HMM, der die ersten $t$ Zeichen von $O$ ausgibt und im Zustand $S_i$ endet, also

$$\delta_i(t) = \max_{\pi_i(t)} P(\pi_i(t), o_1 o_2 \cdots o_t | W) \,.$$

Initialisiert werden die Werte mit $\delta_i(0) = \max_{i=1,2,\ldots,N} t^{D_{max}}_{S_i,S_1}$. Hierbei bezeichnet $t^{D_{max}}_{S_i,S_j}$ die *maximale* Wahrscheinlichkeit aller stummen Pfade von $S_j$ nach $S_i$ im Gegensatz zu $t^{D}_{S_i,S_j}$ aus dem Forward-Algorithmus, welches die *Summe* der Wahrscheinlichkeiten aller stummen Pfade von $S_j$ nach $S_i$ bezeichnet. Die Aktualisierung dieser Wahrscheinlichkeiten erfolgt analog zum Forward-Algorithmus, wobei allerdings Summen durch Maximumbildung zu ersetzen sind ($\mathcal{E} \subset \mathcal{S}$ bezeichnet die Menge der emittierenden Zustände des HMM und $N^-(S_i)$ die Menge der direkten Vorgängerzustände vom Zustand $S_i$):

$$\delta_i(t+1) = e_{S_i,o_{t+1}} \cdot \max_{\{j:S_j \in N^-(S_i)\}} \delta_j(t) t_{S_i,S_j} \qquad \text{wenn } S_i \in \mathcal{E} \,,$$

$$\delta_i(t+1) = \max_{\{j:S_j \in \mathcal{E}\}} \delta_j(t+1) t^{D_{max}}_{S_i,S_j} \qquad \text{wenn } S_i \notin \mathcal{E} \,.$$

(a) (Wohldefiniertheit der $t^{D_{max}}_{S_i,S_j}$)
Zeigen Sie, dass für beliebige Zustände $S_i \notin \mathcal{E}, S_j \in \mathcal{E}$ die Wahrscheinlichkeit $t^{D_{max}}_{S_i,S_j}$ immer definiert ist. Damit sind auch die Viterbi-Variablen $\delta_i(t)$ wohldefiniert.
(b) (Konstruktion optimaler Pfade)
Für jedes Paar $(S_i, S_j)$, $S_i \notin \mathcal{E}, S_j \in \mathcal{E}$, mit $t^{D_{max}}_{S_i,S_j} > 0$ sei ein stummer Pfad $\pi_{S_i,S_j}$ maximaler Wahrscheinlichkeit von $S_j$ nach $S_i$ gegeben. Mit Hilfe diese Pfade und den Größen $\delta_i(t)$ gebe man einen Algorithmus an, der einer Pfad vom Startzustand $S_1$ zum Endzustand $S_N$ mit größter Wahrscheinlichkeit der Ausgabe $O$ ermittelt. (Hinweis: Zu jedem $\delta_i(t)$ merke man sich den vorhergehenden, optimalen Zustand, also jenes $j$, für das das Maximum angenommen wird.)
(c) (Komplexität des Viterbi-Algorithmus für Standardarchitektur)
Wir betrachten ein HMM mit Standardarchitektur und $N$ Hauptzuständen. Damit ist die Länge der Ausgaben dieses HMM von der Ordnung $N$. Leiten Sie einen Ausdruck für die Komplexität des Viterbi-Algorithmus, also der Berechnung der $\delta_i(t)$ für dieses HMM, her.

Wintersemester 2002/03 Abgabe am 3. Februar 2002
Prof. Dr. S. Posch, Dr. A. Gerisch

# 11. Übung „Algorithmen der Bioinformatik II“

**Aufgabe 1.** Wir betrachten ein HMM mit $M$ emittierenden Zuständen $\mathcal{E} = \{E_1, E_2, \ldots, E_M\}$ und Alphabet $\Sigma$, in dem Emissionswahrscheinlichgkeiten $e_{E_i,X}$ geteilt werden. Das heißt, dass eine Teilmenge der emittierenden Zustände die gleichen Emissionswahrscheinlichkeiten haben soll. Sei eine Teilmenge $\mathcal{I} \subset \{1, 2, \ldots, M\}$ gegeben und gelte für alle $i \in \mathcal{I}$

$$e_{E_i,X} = e_{\mathcal{I},X} \geq 0 \quad \text{für alle} \quad X \in \Sigma \quad \text{und} \quad \sum_{X\in\Sigma} e_{\mathcal{I},X} = 1\,.$$

Die $e_{\mathcal{I},X}$, $X \in \Sigma$ bezeichnen also die geteilten Emissionswahrscheinlichkeiten in der durch $\mathcal{I}$ definierten Teilmenge von $\mathcal{E}$.
Ziel der Aufgabe ist es, eine Formel für die Maximum-Likelihood Schätzung der geteilten Parameter des HMMs zu ermitteln. Diese können dann in einem leicht modifizierten Baum-Welsh-Algorithmus verwendet werden. Wir betrachten den Fall des Online-Lernens. Es ist also eine Ausgabe $O$ gegeben. Wie üblich bezeichne $W$ die Menge aller freien Parameter des HMM.
(a) (Aufstellen der Lagrange-Funktion)
Sei $\lambda_{\mathcal{I}}$ der Lagrange-Multiplikator zur Nebenbedingung $\sum_{X\in\Sigma} e_{\mathcal{I},X} = 1$ (die anderen Multiplikatoren können wie in der Vorlesung als $\lambda_i$ und $\mu_j$ bezeichnet werden). Geben Sie die Lagrangefunktion $\mathcal{L}(W, \lambda, \mu)$ für die ML-Schätzung an.

(b) (Vorbereitung für die Ableitung der Lagrange-Funktion)
Sei $f(x, y)$ eine differenzierbare Funktion und seien $x = x(t)$ sowie $y = y(t)$ ebenfalls differenzierbar. $f(x(t), y(t))$ heißt *mittelbare Funktion*. Geben Sie die Ableitung von $f(x(t), y(t))$ nach $t$ an. Was ist die Ableitung von $f(x(t), y(t))$ nach $t$ wenn $x(t) = y(t) = t$ gilt?

(c) (Berechnung der Ableitung der Lagrangefunktion)
Wir suchen stationäre (kritische) Punkte der unter (a) erhaltenen Lagrange-Funktion. Zeigen Sie, dass für die Ableitung von $\mathcal{L}(W, \lambda, \mu)$ nach dem geteilten Parameter $e_{\mathcal{I},X}$ gilt

$$\frac{\partial\mathcal{L}(W, \lambda, \mu)}{\partial e_{\mathcal{I},X}} = \sum_{\pi} \sum_{i\in\mathcal{I}} n(i, X|\pi, O) \frac{P(\pi, O|W)}{e_{\mathcal{I},X}} - \lambda_{\mathcal{I}}\,.$$

(d) (Berechnung des stationären Punkts)
Setzen Sie die in (c) erhaltene Ableitung gleich Null und summieren Sie über alle Zeichen des Alphabets $\Sigma$ auf. Leiten Sie daraus einen Ausdruck für den Multiplikator $\lambda_{\mathcal{I}}$ ab. Nutzen Sie diesen, um zu zeigen, dass für einen stationären Punkt der Lagrange-Funktion gilt

$$e_{E_{\mathcal{I}},X} = \frac{\sum_{i\in\mathcal{I}} n_{i,X}}{\sum_{i\in\mathcal{I}} n_i}\,.$$

(e) (Anpassung des Baum-Welsh-Algorithmus an den Fall geteilter Emissionswahrscheinlichkeiten)
Geben Sie die nötigen Änderungen am Baum-Welsh-Algorithmus an, so dass er auch mit geteilten Emissionswahrscheinlichkeiten verwendet werden kann.