Asja Fischer, Christian Igel

# Markov-Random-Fields und Boltzmann Maschinen

Vorlesungsnotizen

Version 0.1.5

Institut für Neuroinformatik
Ruhr-Universität Bochum

# Vorwort

"Wenn ich groß bin, werde ich ein Vorlesungsskript."

# Inhaltsverzeichnis

# 1

# Markov-Random-Fields

Markov-Random-Fields (MRFs) nutzen ungerichtete Graphen zur Modellierung der Abhängigkeiten von Zufallsvariablen.

Ein ungerichteter Graph ist ein Tupel $G = (V, E)$, wobei $V$ eine endliche Menge von Knoten und $E$ eine Menge von ungerichteten Kanten bezeichnet. Eine ungerichtete Kante zwischen zwei Knoten $v, w \in V$ existiert genau dann, wenn $\{v, w\} \in E$ gilt. In diesem Fall werden $v$ und $w$ auch als *benachbart* bezeichnet und die *Nachbarschaft* $\mathcal{N}_v$ eines Knotens $v$ ist entsprechend gegeben durch

$$\mathcal{N}_v = \{w \in V : \{w, v\} \in E\} \ .$$

Ein Beispiel eines ungerichteten Graphen ist in Abbildung 1.1 zu sehen. Hier bilden die Knoten $v_2$, $v_5$ und $v_6$ die Nachbarschaft von $v_4$.

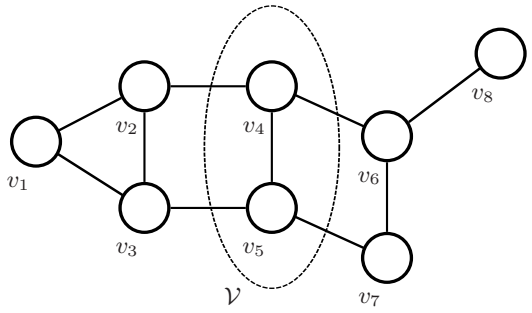


**Abb. 1.1.** Ein Beispiel eines ungerichteten Graphen. Die Knoten $v_1$ und $v_8$ werden von der Menge $\mathcal{V} = \{v_4, v_5\}$ separiert.

Als einen *Pfad* von einem Knoten $v_1$ zu einen Konten $v_m$ bezeichnet man eine Folge von sich unterscheidenden Knoten $v_1, v_2, \ldots, v_m \in V$, mit $\{v_i, v_{i+1}\} \in E$ für $i = 1, \ldots, m-1$. Eine Menge $\mathcal{V} \subset V$ *separiert* zwei Knoten

$v \notin \mathcal{V}$ und $w \notin \mathcal{V}$, wenn jeder Pfad von $v$ nach $w$ wenigstens einen Knoten aus $\mathcal{V}$ enthält (Abbildung 1.1 zeigt ein Beispiel).

Ein weiteres Konzept aus der Graphentheorie, welches für MRFs von Bedeutung ist, ist das einer *Clique*. Eine Clique ist eine Teilmenge von Knoten aus $V$, die sich dadurch auszeichnet, dass Kanten zwischen allen Knoten der Teilmenge existieren, dass diese also *vollvernetzt* ist. Eine Clique wird *maximal* genannt, wenn es nicht möglich ist, andere Knoten des Graphen mit in die Untermenge einzuschließen, ohne dass sie dadurch aufhört, vollvernetzt zu sein. Im Graph in Abbildung 1.1 bilden z. B. die Knoten $v_1$, $v_2$ und $v_3$ eine maximale Clique.

In *gewichteten* Graphen ist jede Kante des Graphen mit einem *Gewicht* assoziiert. Diese Gewichte sind meistens reelle Zahlen.

## 1.1 Definition eines MRFs und die Markov-Eigenschaften

Basierend auf einem ungerichteten Graphen lässt sich ein MRF nun wie folgt definieren:

**Definition 1.** *Sei $G = (V, E)$ ein ungerichteter Graph und jedem Knoten $v \in V$ sei eine Zufallsvariable $X_v$ mit endlichem Zustandsraum $\Lambda$ zugeordnet. Dann wird $\mathbf{X} = (X_v)_{v \in V}$ als* Markov-Random-Field *bezüglich $G$ bezeichnet, wenn für alle $v \in V$ gilt, dass $X_v$ und $(X_w)_{w \in V \setminus (\mathcal{N}_v \cup v)}$ bedingt unabhängig gegeben $(X_w)_{w \in \mathcal{N}_v}$, sind, d. h. wenn für alle $v \in V, \mathbf{x} \in \Lambda^{|V|}$ gilt*

$$P\left(x_v | (x_w)_{w \in V \setminus v}\right) = P\left(x_v | (x_w)_{w \in \mathcal{N}_v}\right) \quad , \tag{1.1}$$

*wobei $\mathcal{N}_v$ die Nachbarschaft des Knotens $v$ bezeichnet.*

Die Eigenschaft, welche durch (1.1) beschrieben ist, wird als *lokale Markov-Eigenschaft* eines MRFs bezeichnet. In Worten besagt sie, dass die Wahrscheinlichkeit, dass eine Zufallsvariable $X_v$ einen bestimmten Wert annimmt, allein von den Werten der Zufallsvariablen $X_w$ abhängig ist, deren zugehöriger Knoten $w$ zur Nachbarschaft des Knotens $v$ gehören.

Es lassen sich zwei weitere Markov-Eigenschaften formulieren. Seien $v, w \in V$ zwei nicht benachbarte Knoten, also $\{v, w\} \notin E$. So sind $X_v$ und $X_w$ bedingt unabhängig gegeben alle anderen Zufallsvariablen $(X_t)_{t \in V \setminus \{v,w\}}$. Es gilt also für alle $\mathbf{x} \in \Lambda^{|V|}$

$$P\left(x_v, x_w | (x_t)_{t \in V \setminus \{v,w\}}\right) = P\left(x_v | (x_t)_{t \in V \setminus \{v,w\}}\right) P\left(x_w | (x_t)_{t \in V \setminus \{v,w\}}\right) \quad . \tag{1.2}$$

Diese Identität wird als *paarweise Markov-Eigenschaft* bezeichnet. Seien $\mathcal{A}, \mathcal{B}, \mathcal{S}$ drei disjunkte Teilmengen von $V$, so dass die Knoten in $\mathcal{A}$ und $\mathcal{B}$ durch $\mathcal{S}$ voneinander separiert sind, so sind $(X_a)_{a \in \mathcal{A}}$ und $(X_b)_{b \in \mathcal{B}}$ bedingt unabhängig, gegeben die Zufallsvariablen $(X_s)_{s \in \mathcal{S}}$. Es gilt also für alle $\mathbf{x} \in \Lambda^{|V|}$

$$P\left((x_a)_{a\in\mathcal{A}}|(x_t)_{t\in\mathcal{S}\cup\mathcal{B}}\right) = P\left((x_a)_{a\in\mathcal{A}}|(x_t)_{t\in\mathcal{S}}\right) \quad . \tag{1.3}$$

Dies ist die *globale Markov-Eigenschaft*.

Es ist leicht zu zeigen (siehe z. B. [1], S. 33), dass aus der globalen die lokale Markov-Eigenschaft und aus der lokalen die paarweise Markov-Eigenschaft folgt. Ein MRF besitzt also immer auch die paarweise Markov-Eigenschaft. Die globale Markov-Eigenschaft besitzt ein MRF aber nur dann, wenn es eine 'passende' gemeinsame Wahrscheinlichkeitsverteilung aufweist (siehe Abschnitt 1.2). Die globale Eigenschaft ist deshalb von Bedeutung, weil sie eine generelle Regel liefert, mit der durch Analyse des Graphen entschieden werden kann, ob zwei Mengen von Zufallsvariablen im MRF, $(X_a)_{a\in\mathcal{A}}$ und $(X_b)_{b\in\mathcal{B}}$, gegeben eine dritte Menge von Zufallsvariablen $(X_s)_{s\in\mathcal{S}}$, voneinander unabhängig sind: Dies ist genau dann der Fall, wenn jeder Pfad von einem Knoten aus $\mathcal{A}$ zu einem Knoten aus $\mathcal{B}$ durch wenigstens einen Knoten in $\mathcal{S}$ führt.

Es gibt ungerichtete graphische Modelle, die bedingte Abhängigkeiten und Unahängigkeiten abbilden, die von gerichteten graphischen Modellen (Bayes'schen Netzwerken) nicht abgebildet werden können. Ein Beispiel gibt Abbildung 1.2.

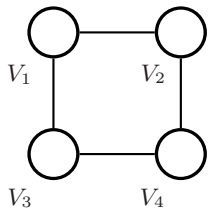


**Abb. 1.2.** Die bedingten Abhängigkeiten zwischen den Zufallsvariablen $V_1$ bis $V_4$, die durch dieses ungerichtete graphische Modell beschrieben werden, können nicht vollständig durch ein Bayes'sches Netzwerk abgebildet werden.

## 1.2 Faktorisierungseigenschaften eines MRFs und die Gibbsverteilung

Die bedingte Wahrscheinlichkeitsverteilung einer Zufallsvariablen $X_v$, gegeben alle übrigen Zufallsvariablen $(X_w)_{w\in V\setminus v}$ eines MRFs $\mathbf{X} = (X_v)_{v\in V}$, kann nun durch

$$p\left(x_v|(x_w)_{w\in V\setminus v}\right) := P\left(x_v|(x_w)_{w\in\mathcal{N}_v}\right) \tag{1.4}$$

ausgedrückt werden. Sie wird auch als *lokale Charakteristik* des MRF 'an der Stelle' $v$ bezeichnet.

Während die Menge aller lokalen Charakteristiken $\{p\left(x_v|(x_w)_{w\in V\setminus v}\right)|v\in V\}$ durch die gemeinsame Wahrscheinlichkeitsverteilung $p$ von $\mathbf{X}$ eindeutig bestimmt ist, führen beliebige lokale Charakteristiken nicht zu einer wohldefinierten gemeinsamen Wahrscheinlichkeitsverteilung. Es stellt sich jedoch heraus, dass $p$ eindeutig durch $\{p\left(x_v|(x_w)_{w\in V\setminus v}\right)|v\in V\}$ bestimmt ist, wenn die *Positivitäts-Bedingung* erfüllt ist. Ein Beweis dieser Aussage kann etwa im Lehrbuch von Brémaud ([2], S. 255 f.) gefunden werden. Eine Verteilung $p$ über einem Zustandsraum $\Lambda^n$ erfüllt die *Positivitäts-Bedingung* wenn $\forall\mathbf{x}\in\Lambda^n$ gilt, dass $p(\mathbf{x})>0$.[1] Die Positivitäts-Bedingung stellt also sicher, dass alle Werte, die von den Zufallsvariablen $X_v$ alleine angenommen werden können, auch in beliebiger Kombination unter der gemeinsamen Wahrscheinlichkeitsverteilung von $\mathbf{X}=(X_v)_{v\in V}$ angenommen werden können.

Da ein enger Zusammenhang zwischen bedingter Unabhängigkeit von Umgebungsvariablen (wie sie z. B. durch die lokale Markov-Eigenschaft eines MRFs spezifiziert ist) und der Faktorisierung der gemeinsamen Wahrscheinlichkeitsfunktion dieser Zufallsvariablen existiert, stellt sich die Frage, ob es möglich ist, eine generelle faktorisierte Form für gemeinsame Wahrscheinlichkeitsverteilungen von MRFs zu finden, d. h. ob es eine generelle Regel gibt, nach der jede mögliche gemeinsame Verteilung $p$ eines MRFs, welche die durch einen Graphen spezifizierte lokale Markov-Eigenschaft und die Positivitäts-Bedingung erfüllt, als ein Produkt von Funktionen über Teilmengen der Zufallsvariablen ausgedrückt werden kann. Erste Überlegungen können davon ausgehen, dass für $p$ die paarweise Markov-Eigenschaft gelten muss. Um Gleichung (1.2) zu erfüllen, muss eine Faktorisierung derart sein, dass für nicht benachbarte Knoten $v,w\in V$ die Zustandswerte $x_v$ und $x_w$ nicht im selben Faktor auftauchen. Ausgehend von diesem Gedanken liegt es nahe, die gemeinsame Verteilung in Funktionen von Zufallsvariablen zu zerlegen, deren zugehörige Knoten Cliquen bilden. Dazu muss man sich ins Gedächtnis rufen, dass Cliquen gerade die Teilmengen der Knoten eines Graphens darstellen, in denen alle Knoten miteinander benachbart sind. Zwischen den zugehörigen Zufallsvariablen tritt also keine bedingte Unabhängigkeit auf. In Gibbsverteilungen ist eine solche Faktorisierung wiederzufinden:

[1] Die Positivitätsbedingung kann abgeschwächt werden. Sei $V=\{1,2,...,n\}$, $p$ die gemeinsame Wahrscheinlichkeitsverteilung von $\mathbf{X}=(X_1,X_2,....X_n)$ auf dem endlichen Zustandsraum $\Lambda^n$ und die Marginalverteilung der Zufallsvariablen $X_v$ durch $p_v(x_v)$ gegeben. Dann erfüllt $p$ die *Positivitäts-Bedingung*, wenn $\forall\mathbf{x}=(x_1,x_2,...,x_n)\in\Lambda^n$ gilt:

$$p_v(x_v)>0 \text{ für } v=1,2,...,n \Rightarrow p\left((x_1,x_2,...,x_n)\right)>0\ . \tag{1.5}$$

Es ist möglich, die Positivitäts-Bedingung in einer anderen Form anzugeben ([2], S. 255)). Dass diese und die hier angegebene Form äquivalent sind, folgt, weil jeweils auch die Rückrichtungen der Aussagen gelten. Dies wird deutlich, wenn betrachtet wird, wie die Marginalverteilungen aus der gemeinsamen Verteilung hervorgehen (z. B. $p_1(x_1)=\sum_{x_2\in\Lambda}\cdots\sum_{x_n\in\Lambda}p((x_1,x_2,\ldots,x_n))$.

**Definition 2.** *Gegeben sei ein Graph $G = (V, E)$ und die Menge $\mathcal{C}$ aller Cliquen in $V$. Eine Menge $\{\Psi_C\}_{C \subset \mathcal{C}}$ von Funktionen $\Psi_C : \Lambda^{|V|} \to \mathbb{R} \cup \{+\infty\}$ mit*

$$\forall \mathbf{x}, \mathbf{x}' \in \Lambda^{|V|} : (x_c)_{c \in C} = (x_c)_{c \in C} \Rightarrow \Psi_C(\mathbf{x}) = \Psi_C(\mathbf{x}')$$

*wird als* (Gibbs-)Potential *bezeichnet. Eine Wahrscheinlichkeitsverteilung $p$ auf $\Lambda^{|V|}$ wird* Gibbsverteilung *genannt, wenn gilt*

$$p(\mathbf{x}) = \frac{1}{Z} e^{-\frac{1}{T} E(\mathbf{x})} \quad , \tag{1.6}$$

*wobei*

$$Z = \sum_{\mathbf{x} \in \Lambda^{|V|}} e^{-\frac{1}{T} E(\mathbf{x})} \tag{1.7}$$

*eine Normalisierungskonstante darstellt und als* Zustandssumme *(im Englischen* partition function*) bezeichnet wird und $T$ eine Konstante ist. Die Funktion $E$ ist durch*

$$E(\mathbf{x}) = \sum_{C \subset \mathcal{C}} \Psi_C(\mathbf{x}) \tag{1.8}$$

*gegeben und wird* Energie *genannt.*

In physikalischen Anwendungen bezeichnet die Konstante $T$ die *Temperatur*. Im Zusammenhang mit MRFs ist jedoch häufig $T = 1$ gewählt. Die Bezeichnung von $E$ als Energie stammt ebenfalls aus der Physik. Hier beschreibt $p(\mathbf{x})$ die Wahrscheinlichkeit, dass ein System den Zustand $\mathbf{x}$ annimmt, und diese ist um so größer, je geringer die Energie des Zustandes ist.

Die Funktionen $\Psi_C$ sind nicht eindeutig bestimmt. So können Gruppen von Funktionen auf verschiedene Weise zusammengefasst und aufgesplittet werden. Ohne Einschränkung der Allgemeinheit ist es möglich, in der Summe in (1.8) nur maximale Cliquen zu berücksichtigen, da alle anderen Cliquen Teilmengen der maximalen Cliquen sind. Damit können Funktionen über Cliquen, die Teilmengen einer maximalen Clique sind, einfach zu einer Funktion dieser maximalen Clique zusammengefasst werden.

Es ist leicht nachzuprüfen, dass eine Gibbsverteilung die lokale Markov-Eigenschaft und sogar die globale Markov-Eigenschaft bezüglich des zugehörigen Graphen erfüllt (entsprechende Beweise sind in [2], S. 262. f., bzw. [1], S. 35, zu finden). Das heißt also, dass eine Menge von Zufallsvariablen, die einem Graphen zugeordnet sind und eine Gibbsverteilung bezüglich des Graphen besitzen, ein MRF bilden. Überraschender ist, dass auch die Umkehrung gilt: Die Wahrscheinlichkeitsverteilung eines beliebigen MRF bezüglich eines Graphens, welche die Positivitäts-Bedingung erfüllt, ist eine Gibbsverteilung. Die Gibbsverteilung stellt also die generelle faktorisierte Form dar, welche wir gesucht haben. Das Resultat wurde von mehreren Autoren in verschiedenen Formen entdeckt [3, 4, 5, 6], wurde aber zuerst von Hammersley und Clifford [7] in diskreter Form formuliert:

**Theorem 1 (Satz von Hammersley und Clifford).** *Sei $p$ die Verteilung eines MRFs bezüglich eines Graphens $G = (V, E)$ und erfülle die Positivitäts-Bedingung, so gilt*

$$p(\mathbf{x}) = \frac{1}{Z} e^{-E(\mathbf{x})}$$

*für eine Energiefunktion $E$, welche auf einem Gibbs-Potential bezüglich $G$ basiert.*

Ein Beweis dieses Satzes kann in Brémaud ([2], S. 262 f.) nachgeschlagen werden.

Mit der Definition $\psi_C(\mathbf{x}) = e^{-\frac{1}{T}\Psi_C(\mathbf{x})}$ gilt

$$p(\mathbf{x}) = \frac{1}{Z} e^{-\frac{1}{T}E(\mathbf{x})} = \frac{1}{Z} \prod_{C \subset \mathcal{C}} \psi_C(\mathbf{x}) \ . \tag{1.9}$$

Die Funktion $\psi_C$ wird als Potentialfunktion der Clique $C$ bezeichnet. Potentialfunktionen messen die Kompatibilität einer Beobachtung $\mathbf{x}$ mit der repräsentierten Verteilung. Die Messung durch $\psi_C$ kann als das Urteil eines Experten für die Zufallsvariablen in $C$ interpretiert werden und daher ist für die Faktorisierung (1.9) die Bezeichnung *Produkt von Experten* (*product of experts*) geläufig.

Die Präsenz der Normalisierungskonstante $Z$ in der Gibbsverteilung stellt einen großen Nachteil von MRFs dar, da, um $Z$ zu berechnen, eine Summe über alle möglichen Konfigurationen der Zufallsvariablen $X_v, v \in V$ gebildet werden muss. Enthält der Zustandsraum $\Lambda$ nun $K$ Elemente, so besteht die Summe aus $K^{|V|}$ Summanden und ist somit exponentiell abhängig von der Anzahl der Zufallsvariablen und damit selbst für relativ kleine Modelle nicht berechenbar. Dieses Problem taucht auch beim Parameter-Lernen in MRFs auf, wie im nächsten Abschnitt genauer erläutert wird. Bei der Berechnung lokaler Charakteristika ist es jedoch nicht nötig, den Wert der Normalisierungskonstante zu bestimmen. Dies kann man sich dadurch klar machen, dass die bedingte Wahrscheinlichkeitsverteilung berechnet werden kann, indem man eine Marginalverteilung durch eine andere dividiert, wobei $Z$ herausgekürzt wird.

Es ist anzumerken, dass die Definition und die Eigenschaften eines MRF nicht voraussetzen, dass die Zufallsvariablen denselben endlichen Ergebnisraum besitzen. Da dies aber in vielen Anwendungen (so auch bei binären RBMs, auf die hier der Fokus liegt) der Fall ist und zu einer einfacheren Notation führt, wurde diese Darstellung hier gewählt. Ein entsprechender Beweis des Satzes von Hammersley und Clifford für MRFs mit Zufallsvariablen auf reellen endlichdimensionalen Wahrscheinlichkeitsräumen ist z. B. bei Lauritzen ([1], S. 36) zu finden.

## 1.3 MRFs mit latenten Variablen

Es kann sinnvoll sein, zusätzliche Variablen, welche als *latente* oder auch *versteckte* Variablen bezeichnet werden, einzuführen. Die übrigen Knoten des Modells werden dann als *sichtbare* Variablen oder auch *beobachtete* Variablen bzeichnet. Das Modell definiert nun die gemeinsame Wahrscheinlichkeitsverteilung aller Variablen, und die Verteilung über die sichtbaren Variablen kann durch Marginalisierung gewonnen werden. Dadurch ist es möglich, relativ komplexe Marginalverteilungen der sichtbaren Variablen durch eine einfach handhabbare gemeinsame Wahrscheinlichkeitsverteilung über den erweiterten Raum der sichtbaren und latenten Variablen darzustellen. Die Einführung versteckter Variablen macht es also möglich, komplizierte Verteilungen durch einfachere Komponenten auszudrücken. In Graphischen Modellen sind die sichtbaren Variablen oft als schattierte Knoten und die versteckten Variablen als nicht schattierte Knoten dargestellt.

Seien im Folgenden nun die Zufallsvariablen $\mathbf{X}$ eines MRFs in sichtbare Variablen $\mathbf{V} = (V_1, ..., V_m)$ und versteckte Variablen $\mathbf{H} = (H_1, ..., H_n)$ unterteilt, und der Ergebnisraum aller Zufallsvariablen sei durch die endliche Menge $\Lambda$ gegeben. Die gemeinsame Wahrscheinlichkeitsverteilung von $\mathbf{X} = (\mathbf{V}, \mathbf{H})$ ist, wie oben beschrieben (wenn sie die Positivitäts-Bedingung erfüllt), eine Gibbsverteilung, d. h.

$$p(\mathbf{v}, \mathbf{h}) = \frac{1}{Z} e^{-E(\mathbf{v},\mathbf{h})} \quad , \tag{1.10}$$

wobei in

$$Z = \sum_{\mathbf{v}', \mathbf{h}'} e^{-E(\mathbf{v}',\mathbf{h}')} \tag{1.11}$$

über alle möglichen Werte $\mathbf{h}' \in \Lambda^n$ von $\mathbf{H}$ und $\mathbf{v}' \in \Lambda^m$ von $\mathbf{V}$ summiert wird. Die Marginalverteilung der sichtbaren Variablen ergibt sich nun wie folgt:

$$p(\mathbf{v}) = \sum_{\mathbf{h}} p(\mathbf{v}, \mathbf{h}) = \frac{1}{Z} \sum_{\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})} \quad . \tag{1.12}$$

Da die Summe über alle möglichen Werte von $\mathbf{H}$ aus $|\Lambda|^n$ Summanden besteht, ist diese für größere Modelle nicht unbedingt berechenbar. In dem Fall, in dem sich die Energie als Summe von Termen schreiben lässt, welche jeweils mit höchstens einer versteckten Variable assoziiert sind, kann eine geschickte Faktorisierung genutzt werden. Die Funktionen des Gibbs-Potentials seien also derart, dass die Energie durch

$$E(\mathbf{v}, \mathbf{h}) = -\beta(\mathbf{v}) - \sum_{i=1}^{n} \alpha_i(\mathbf{v}, h_i) \tag{1.13}$$

gegeben ist, wobei $\beta$ und $\alpha_i$, für $i = 1, ..., n$ beliebige Funktionen sind. Dann kann $\sum_{\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})}$ wie folgt faktorisiert werden:

$$\begin{aligned}\sum_{\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})} &= \sum_{h_1}\sum_{h_2}\dots\sum_{h_n} e^{\beta(\mathbf{v})+\sum_{i=1}^{n}\alpha_i(\mathbf{v},h_i)} \\ &= \sum_{h_1}\sum_{h_2}\dots\sum_{h_n} e^{\beta(\mathbf{v})}\prod_{i=1}^{n} e^{\alpha_i(\mathbf{v},h_i)} \\ &= e^{\beta(\mathbf{v})}\sum_{h_1} e^{\alpha_i(\mathbf{v},h_1)}\sum_{h_2} e^{\alpha_i(\mathbf{v},h_2)}\dots\sum_{h_n} e^{\alpha_i(\mathbf{v},h_n)} \\ &= e^{\beta(\mathbf{v})}\prod_{i=1}^{n}\sum_{h_i} e^{\alpha_i(\mathbf{v},h_i)} \quad . \end{aligned} \tag{1.14}$$

Durch die Faktorisierung ergibt sich also ein Produkt von $n$ Summen über jeweils $|\Lambda|$ Summanden und damit eine lineare statt einer exponentiellen Abhängigkeit des Rechenaufwands von der Anzahl der versteckten Variablen.

## 1.4 Parameter-Lernen in MRFs

Eine zentrales Problem im Bereich des maschinellen Lernens ist es, ein Modell einer Wahrscheinlichkeitsverteilung ausgehend von einer endlichen Menge von Stichproben, welche von dieser Verteilung gezogen wurden, zu konstruieren. Bezogen auf graphische Modelle kann dieses Problem im Lernen der Struktur des Graphen bestehen und/oder darin, bei bekannter Graphen-Struktur und Form der Wahrscheinlichkeitsverteilung die konkreten Parameter der Verteilung zu lernen. Im Folgenden betrachten wir das Lernen der Parameter bei vorgegebener Graphstruktur.

Beim Lernen der Parameter eines MRFs wird davon ausgegangen, dass der Graph bekannt und die gemeinsame Wahrscheinlichkeitsfunktion eine Gibbsverteilung mit bekannter Form ist. Das bedeutet, dass das Gibbs-Potential $\{\Psi_C\}_{C\subset\mathcal{C}}$ und damit die Energiefunktion $E$ bis auf konkrete zu lernende Parameter $\boldsymbol{\theta}$ spezifiziert sind. Diese Parameter sollen nun ausgehend von einer endlichen Menge beobachteter Daten bestimmt werden. Eine verbreitete, aus der Statistik stammende Methode zur Bestimmung der Parameter ist die *Maximum-Likelihood-Methode*.

Basierend auf einem beobachteten Wert $\mathbf{x}$ der Zufallsvariablen, ist die *Likelihood* der MRF-Modellparameter $\boldsymbol{\theta}$ durch

$$L(\boldsymbol{\theta}|\mathbf{x}) := p(\mathbf{x}) = \frac{1}{Z}e^{-E(\mathbf{x})} \tag{1.15}$$

gegeben, wobei $E$ von $\boldsymbol{\theta}$ abhängig ist. Unter der *Likelihood-Funktion* versteht man die Funktion $L : \Theta \to \mathbb{R}$, welche Parametern $\boldsymbol{\theta}$ aus einer Menge $\Theta$ möglicher Parameter die Likelihood $L(\boldsymbol{\theta}|\mathbf{x})$ zuordnet. Die *Log-Likelihood-Funktion* ist auf entsprechende Weise durch die *Log-Likelihood* $\log L(\boldsymbol{\theta}|\mathbf{x})$ gegeben. Basierend auf einer Menge $S = (\mathbf{x}_1, ..., \mathbf{x}_N)$ von Stichproben, welche unabhängig

voneinander gezogen wurden, ist die Likelihood durch $\prod_{i=1}^{N} L(\boldsymbol{\theta}|\mathbf{x}_i)$ und die Log-Likelihood damit durch $\sum_{i=1}^{N} \log L(\boldsymbol{\theta}|\mathbf{x}_i)$ gegeben. Ausgehend von der Stichprobenmenge $S$ werden nun diejenigen Parameter als *Maximum-Likelihood-Schätzer* bezeichnet, welche die Wahrscheinlichkeit, die Stichproben unter dem Modell zu erhalten, maximieren. Dies sind eben die Parameter, für welche die Likelihood-Funktion (und da der Logarithmus eine monotone Funktion ist, damit auch die Log-Likelihood-Funktion) ihr Maximum annimmt. Für die Gibbsverteilung eines MRF ist es nicht möglich, die Maximum-Likelihood-Schätzer durch einfache Extremwertermittelung der Likelihood-Funktion analytisch zu bestimmen, weshalb numerische Approximationsverfahren genutzt werden, um lokale Extremata zu approximieren.

Aus dem Blickwinkel der Maximum-Likelihood-Methode besteht das Lernproblem also nun darin, die Parameter so zu verändern, dass die Log-Likelihood der beobachteten Daten unter dem Modell möglichst groß wird. Dies kann als ein mathematisches Optimierungsproblem formuliert werden. In einer alternativen Betrachtungsweise, die aber zu dem selben Optimierungsproblem führt, besteht das Lernproblem darin, die Wahrscheinlichkeitsverteilung des MRF an die Wahrscheinlichkeitsverteilung, aus welcher die Trainingsdaten gezogen wurden, anzunähern, indem der Abstand zwischen den beiden Verteilungen minimiert wird. Dabei dient die *Kullback-Leibler-Divergenz* (KL-Divergenz) als Abstandsmaß. Die KL-Divergenz zwischen zwei Wahrscheinlichkeitsverteilungen $p$ und $q$ auf einem endlichen Zustandsraum $\Omega$ ist wie folgt definiert:

$$\mathrm{KL}(q||p) = \sum_{\mathbf{x}\in\Omega} q(\mathbf{x}) \log \frac{q(\mathbf{x})}{p(\mathbf{x})} \tag{1.16}$$

Die KL-Divergenz ist nie negativ und genau dann gleich Null, wenn $p = q$ gilt. Sie ist nicht symmetrisch und damit keine Metrik. Bezeichne nun $p$ die Verteilung des MRFs und $q$ die empirische Verteilung, welche modelliert werden soll, dann ist es leicht zu sehen, dass ein Optimierungsproblem, welches die Minimierung der KL-Divergenz zum Ziel hat, äquivalent zu einem Optimierungsproblem ist, welches die Maximierung der Log-Likelihood erzielt. Denn die KL-Divergenz lässt sich wie folgt umformen:

$$\mathrm{KL}(q||p) = \sum_{\mathbf{x}\in\Omega} q(\mathbf{x}) \log q(\mathbf{x}) - \sum_{\mathbf{x}\in\Omega} q(\mathbf{x}) \log p(\mathbf{x}) \ . \tag{1.17}$$

Nun entspricht der erste Term der negativen Entropie[2] der empirischen Verteilung, welche konstant ist, und der Gesamtausdruck wird minimal, wenn

[2] Die Entropie einer Zufallsvariablen misst, umgangssprachlich ausgedrückt, die Information, die im Mittel dadurch gewonnen wird, wenn die Variable beobachtet wird.

der der zweite Term maximal ist. Basierend auf der Menge der Stichproben $S$ kann der Erwartungswert $\sum_{\mathbf{x}\in\Omega} q(\mathbf{x}) \log p(\mathbf{x})$ durch $\frac{1}{N}\sum_{i=1}^{N} \log p(\mathbf{x}_i) = \frac{1}{N}\sum_{i=1}^{N} \log L(\boldsymbol{\theta}|\mathbf{x}_i)$ approximiert werden. Es wird also deutlich, dass die KL-Divergenz genau dann minimiert wird, indem die Log-Likelihood basierend auf den Trainingsbeispielen maximiert wird.

Zur Maximierung der Log-Likelihood, bzw. zur Minimierung der KL-Divergenz, und damit zur Optimierung der Parameter, kommen häufig gradientenbasierte Optimierungsmethoden, wie *Gradientenauftieg* bzw. -*abstieg*, zum Einsatz. Gradientenaufstieg, auch bekannt als *Verfahren des steilsten Aufstiegs*, ist ein iteratives Verfahren, welches auf dem Gradienten der Funktion beruht, die maximiert werden soll. Da sich die Extremata durch einen verschwindenden Gradienten auszeichnen, und der Gradient an einer Stelle die Richtung der größten Zunahme des Funktionswerts angibt (bzgl. des Euklid'schen Metrik für einen infinitesimal kleinen Schritt), werden die Parameter schrittweise in Richtung des Gradienten verschoben, um ein Maximum der Funktion zu erreichen. Auf entsprechende Weise ist Gradientenabstieg eine auf dem negativen Gradienten basierende Methode zur Approximation eines Minimums einer Funktion.

Ein Schritt eines Gradientenaufstiegs auf der Log-Likelihood basierend auf der Trainingsmenge $S = \{\mathbf{x}_i, \ldots, \mathbf{x}_N\}$ kann durch die folgende Gleichung beschrieben werden, welche angibt, wie sich durch den Gradienten der Log-Likelihood-Funktion die neuen Parameter $\boldsymbol{\theta}^{(t+1)}$ aus den alten Parametern $\boldsymbol{\theta}^{(t)}$ ergeben:

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - \eta \frac{\partial}{\partial \boldsymbol{\theta}^{(t)}} \left( \sum_{i=1}^{N} \log L(\boldsymbol{\theta}^{(t)}|\mathbf{x}_i) \right) \tag{1.18}$$

Dabei bezeichnet $\eta > 0$ die Lernrate. In (1.18) wird der Gradient der Log-Likelihood für alle Trainingsbeispiele approximiert und aufsummiert, bevor die Parameter geändert werden. Diese Methode wird als *Batch*-Lernen bezeichnet. Ein alternativer Ansatz ist *Online*-Lernen, bei dem durch die Trainingsmenge iteriert wird und ein Parameter-Update jeweils nur auf einem (zufällig gewählten) Trainingsbeispiel basiert. Diese Variante des Gradientenaufstiegs wird auch als *stochastischer Gradientenaufstieg* bezeichnet und hat sich in der Praxis beim Trainieren neuronaler Netze mit großen Trainingsmengen bewährt [8]. Es ist auch möglich einen Mittelweg zwischen Batch- und Online-Szenario zu wählen, und basierend auf einigen (aber nicht allen) Trainingsbeispielen zu lernen, was häufig als *Minibatch*-Lernen bezeichnet wird. Ein Vergleich der Vor- und Nachteile von Batch- und Online-Lernen kann in [9] gefunden werden.

Im Folgenden soll der Log-Likelihood-Gradient für MRF mit latenten Variablen genauer betrachtet werden. Wie beschrieben, werden beobachtete Daten in einem MRF mit latenten Variablen nur durch die sichtbaren Zufalls-

variablen repräsentiert. Seien Trainingsdaten $S = \{\mathbf{v}_1, \ldots, \mathbf{v}_N\}$, d. h. exemplarische Beobachtungen der sichtbaren Variablen, gegeben. Dann ist das Ziel die Maximierung der Log-Likelihood der Parameter $\boldsymbol{\theta}$

$$\log L(\boldsymbol{\theta}|S) = \sum_{\mathbf{v}\in S} \log L(\boldsymbol{\theta}|\mathbf{v}) \ . \tag{1.19}$$

Wir betrachten nun die Log-Likelihood bei einem beobachteten Wert $\mathbf{v}$ der sichtbaren Zufallsvariablen. Diese ist durch

$$\begin{aligned} \log L(\boldsymbol{\theta}|\mathbf{v}) = \log p(\mathbf{v}) &= \log \frac{1}{Z} \sum_{\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})} \\ &= \log \sum_{\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})} - \log \sum_{\mathbf{v}',\mathbf{h}'} e^{-E(\mathbf{v}',\mathbf{h}')} \end{aligned} \tag{1.20}$$

gegeben.

Der Log-Likelihood-Gradient ergibt sich unter Anwendung von Summen- und Kettenregel wie folgt:

$$\begin{aligned} \frac{\partial \log L(\boldsymbol{\theta}|\mathbf{v})}{\partial \boldsymbol{\theta}} &= \frac{\partial}{\partial \boldsymbol{\theta}} \left( \log \sum_{\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})} \right) - \frac{\partial}{\partial \boldsymbol{\theta}} \left( \log \sum_{\mathbf{v}',\mathbf{h}'} e^{-E(\mathbf{v}',\mathbf{h}')} \right) \\ = -\frac{1}{\sum\limits_{\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})}} \sum_{\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})} \frac{\partial E(\mathbf{v},\mathbf{h})}{\partial \boldsymbol{\theta}} &+ \frac{1}{\sum\limits_{\mathbf{v}',\mathbf{h}'} e^{-E(\mathbf{v}',\mathbf{h}')}} \sum_{\mathbf{v}',\mathbf{h}'} e^{-E(\mathbf{v}',\mathbf{h}')} \frac{\partial E(\mathbf{v}',\mathbf{h}')}{\partial \boldsymbol{\theta}} \\ &= -\sum_{\mathbf{h}} p(\mathbf{h}|\mathbf{v}) \frac{\partial E(\mathbf{v},\mathbf{h})}{\partial \boldsymbol{\theta}} + \sum_{\mathbf{v}',\mathbf{h}'} p(\mathbf{v}',\mathbf{h}') \frac{\partial E(\mathbf{v}',\mathbf{h}')}{\partial \boldsymbol{\theta}} \\ &= -\sum_{\mathbf{h}} p(\mathbf{h}|\mathbf{v}) \frac{\partial E(\mathbf{v},\mathbf{h})}{\partial \boldsymbol{\theta}} + \sum_{\mathbf{v}'} p(\mathbf{v}') \sum_{\mathbf{h}'} p(\mathbf{h}'|\mathbf{v}') \frac{\partial E(\mathbf{v}',\mathbf{h}')}{\partial \boldsymbol{\theta}} \ . \end{aligned} \tag{1.21}$$

Dabei wurde im vorletzten Schritt benutzt, dass die bedingte Verteilung $p(\mathbf{h}|\mathbf{v})$ durch

$$p(\mathbf{h}|\mathbf{v}) = \frac{p(\mathbf{h},\mathbf{v})}{p(\mathbf{v})} = \frac{\frac{1}{Z} e^{-E(\mathbf{v},\mathbf{h})}}{\frac{1}{Z} \sum\limits_{\mathbf{h}'} e^{-E(\mathbf{v},\mathbf{h}')}} = \frac{e^{-E(\mathbf{v},\mathbf{h})}}{\sum\limits_{\mathbf{h}'} e^{-E(\mathbf{v},\mathbf{h}')}} \tag{1.22}$$

gegeben ist. Soll deutlich gemacht werden, dass die beiden Terme in (1.21) Erwartungswerte sind, kann der Log-Likelihood-Gradienten auch als

$$\begin{aligned} \frac{\partial \log L(\boldsymbol{\theta}|\mathbf{v})}{\partial \boldsymbol{\theta}} &= -E_{p(\mathbf{h}|\mathbf{v})} \left[ \frac{\partial E(\mathbf{v},\mathbf{h})}{\partial \boldsymbol{\theta}} \right] + E_{p(\mathbf{h}',\mathbf{v}')} \left[ \frac{\partial E(\mathbf{v}',\mathbf{h}')}{\partial \boldsymbol{\theta}} \right] \\ &= -E_{p(\mathbf{h}|\mathbf{v})} \left[ \frac{\partial E(\mathbf{v},\mathbf{h})}{\partial \boldsymbol{\theta}} \right] + E_{p(\mathbf{v})} \left[ E_{p(\mathbf{h}'|\mathbf{v}')} \left[ \frac{\partial E(\mathbf{v}',\mathbf{h}')}{\partial \boldsymbol{\theta}} \right] \right] \end{aligned} \tag{1.23}$$

geschrieben werden. Ist die im Abschnitt 1.3 beschriebene Faktorisierung anwendbar, so kann die Summe im ersten Term von (1.21) effizient berechnet werden. In diesem Fall kann die Faktorisierung auch genutzt werden, um die innere Summe im zweiten Term zu berechnen. Bei der vollständigen Berechnung des Gradienten, stellt nun aber, wie oben beschrieben, die Normalisierungskonstante (bzw. die Ableitung der Normalisierungskonstanten, welche durch den zweiten Term in (1.21) gegeben ist) trotzdem ein Problem dar, weil diese die Summe über alle möglichen Konfigurationen der sichtbaren Variablen $\mathbf{V}$ beinhaltet. Gibt es eine Möglichkeit, Stichproben aus der Verteilung der sichtbaren Variablen zu ziehen, so kann dieser Erwartungswert über $\mathbf{V}$ jedoch approximativ bestimmt werden.

Ist es nicht möglich, die Summe über die versteckten Variablen zu faktorisieren, so kann eine Monte-Carlo-Approximation für beide Terme genutzt werden: der bedingte Erwartungswert im ersten Term kann durch Stichproben der bedingten Verteilung von $\mathbf{H}$, gegeben die beobachteten Werte der sichtbaren Variablen, und der Erwartungswert im zweiten Term durch Stichproben der gemeinsamen Wahrscheinlichkeitsverteilung von $\mathbf{V}$ und $\mathbf{H}$ näherungsweise bestimmt werden.

Im Zusammenhang mit Boltzmann Maschinen, welche eine spezielle Art von MRFs darstellen (siehe Abschnitt 2), wurde durch Hinton et al. [10, 11, 12] für die beiden benötigten Sampling-Prozeduren die folgende Terminologie eingeführt: die *positive Phase* bezeichnet die Prozedur, in der die sichtbaren Variablen $\mathbf{V}$ auf die beobachteten Werte $\mathbf{v}$ gesetzt und Stichproben aus der Verteilung der latenten Variablen $\mathbf{H}$, gegeben $\mathbf{v}$, gezogen werden. Das Generieren von Stichproben aus der gemeinsamen Verteilung von $\mathbf{V}$ und $\mathbf{H}$ wird als *negative Phase* bezeichnet.

Markov-Chain-Monte-Carlo-Verfahren, zu denen auch das *Gibbs-Sampling* [13, 14] zählt, bieten die Möglichkeit, die benötigten Stichproben zu erzeugen.

## 1.5 Übungsaufgaben

**Übung 1.** Welche bedingten Abhängigkeiten und Unabhängigkeiten werden durch den Graph in Abbildung 1.2 festgelegt?

**Übung 2.** Zeigen Sie, dass aus der globalen Markov-Eigenschaft die lokale folgt und diese wiederum die paarweise Markov-Eigenschaft impliziert.

**Übung 3.** Zeigen Sie, dass eine Gibbsverteilung die lokale Markov-Eigenschaft und sogar die globale Markov-Eigenschaft bezüglich des zugehörigen Graphen erfüllt.

**Übung 4.** Ein bekanntes Beispiel für den Einsatz von Markov-Random-Fields ist das Entfernen von Rauschen aus Bildern [15, 16]. Wir betrachten den einfachen Fall eines binären Bildes $\mathbf{y}$ mit Pixeln $y_i \in \{-1, 1\}$. Wir nehmen an, dass dem beobachteten Bild $\mathbf{y}$ ein rauschfreies Bild $\mathbf{x}$ zu Grunde liegt und

**y** aus **x** dadurch enstanden ist, dass sich das Vorzeichen von einigen Pixeln zufällig geändert haben.

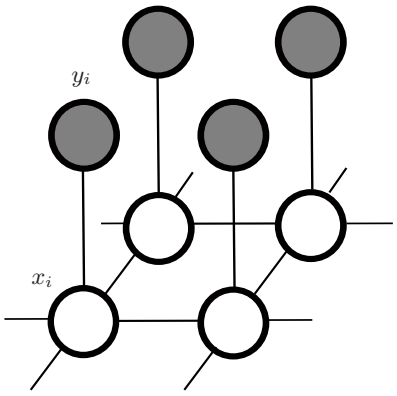


**Abb. 1.3.** Markov-Random-Field zum Entfernen von Bildrauschen, siehe Übungsaufgabe 4.

Wir setzen ein nicht zu geringes Signal-Rausch-Verhältnis voraus. Daher sind $x_i$ und $y_i$ korreliert. Ferner gehen wir davon aus, dass benachbarte Pixel korreliert sind. Dises Vorwissen kann durch ein Markov-Random-Field, wie es in Abbildung 1.3 gezeigt ist, kodiert werden. Die Energiefunktion schreiben wir wie folgt:

$$E(\mathbf{x}, \mathbf{y}) = h \sum_i x_i - \beta \sum_{\{i,j\}} x_i x_j - \eta \sum_i x_i y_i \tag{1.24}$$

Die nichtnegativen reellwertigen Parameter $\beta$ und $\eta$ wichten den Einfluss der Korrelationen mit den Nachbarn bzw. mit dem beobachteten Bild. Über den ersten, mit $h$ gewichteten Term können generell Bilder mit vielen oder wenigen auf 1 gesetzten Pixeln bevorzugt werden.

1. Beschreiben sie die Cliquen in dem Graph.
2. Wie ändert sich die Energie, wenn ein Pixel $x_i$ sein Vorzeichen ändert?
3. Schreiben Sie ein Programm, das ein Bild einliest, dieses wie oben beschrieben verrauscht und dann mit Hilfe des beschriebenen Markov-Random-Fields das Rauschen reduziert. Um (1.24) zu minimieren, können Sie eine gierige, iterative Heuristik verwenden. Wir initialisieren mit $\mathbf{x} = \mathbf{y}$. In jeder Iteration werden alle Pixel $x_i$ hintereinander einzeln betrachtet. Das Vorzeichen des jeweiligen Pixel wird geändert. Nimmt die Energie ab, so wird die Änderung beibehalten, nimmt sie zu, wird sie rückgängig gemacht. Dies wird solange wiederholt, bis in einer Iteration

keines der Pixel $x_i$ seine Wert ändert. Dieses Vorgehen, das auch als *iterated conditional modes* (ICM) bezeichnet wird [15], führt zu einem lokalen, nicht notwendigerweise globalen Optimum.

Weitere Informationen, insbesondere eine Referenz auf eine Strategie zum Finden des globalen Optimums, finden sich in [16].

# 2

# Boltzmann Maschinen

Eine Boltzmann Maschine (BM) [11, 12, 10] ist ein MRF mit latenten Variablen, das sich durch eine spezifische Graphenstruktur auszeichnet und dessen gemeinsame Wahrscheinlichkeitsverteilung durch eine Gibbsverteilung mit einer spezifischen Energiefunktion gegeben ist.

Der Graph einer BM ist gewichtet und legt eine Darstellung in zwei Schichten nahe. Die Knoten der einen Schicht repräsentieren sichtbare Variablen $\mathbf{V} = (V_1, ..., V_m)$, also Zufallsvariablen, deren Zustandswerte beobachtbaren Zuständen der Umwelt entsprechen, und werden im Folgenden als sichtbare Knoten bezeichnet. Die Knoten in der anderen Schicht entsprechen versteckten Variablen $\mathbf{H} = (H_1, ..., H_n)$. Sie dienen dazu, Abhängigkeiten in den sichtbaren Variablen zu modellieren und werden als versteckte Knoten bezeichnet. In der ursprünglichen und gebräuchlichsten Form von BMs sind alle Zufallsvariablen binär, d. h. sie nehmen Werte aus der Menge $\{0, 1\}$ an. In diesem Fall stellt eine BM mit $m$ sichtbaren Variablen also ein Modell einer Wahrscheinlichkeitsverteilung auf dem elementaren Grundraum $\Omega = \{0, 1\}^m$ dar. Alle Kanten des Graphen sind symmetrisch gewichtet und alle Knoten sind mit einem *Bias-Term* assoziiert. Das Gewicht zwischen dem $j$-ten versteckten und dem $i$-ten sichtbaren Knoten sei nun mit $w_{ij}$, das Gewicht zwischen zwei sichtbaren Knoten, die mit den Variablen $V_k$ und $V_l$ (mit $l < k$) assoziiert sind, mit $u_{kl}$ und das Gewicht zwischen zwei versteckten Knoten, die mit den Variablen $H_k$ und $H_l$ (mit $l < k$) assoziiert sind, mit $y_{kl}$ bezeichnet. Weiterhin bezeichne $b_j$ den Bias-Term des $j$-ten sichtbaren und $c_i$ den Bias-Term des $i$-ten versteckten Knotens.

Basierend auf diesen Parametern ist nun die für BMs spezifische Energiefunktion wie folgt definiert:

$$E(\mathbf{v},\mathbf{h}) = \\ -\sum_{i=1}^{n}\sum_{j=1}^{m} h_i w_{ij} v_j - \sum_{k=1}^{m}\sum_{l=1}^{k-1} v_k u_{kl} v_l - \sum_{k=1}^{n}\sum_{l=1}^{k-1} h_k y_{kl} h_l - \sum_{j=1}^{m} b_j v_j - \sum_{i=1}^{n} c_i h_i \tag{2.1}$$

Durch die Energiefunktion ist auch die durch die BM modellierte Wahrscheinlichkeitsverteilung, die durch die zugehörige Gibbsverteilung gegeben ist, vollständig beschrieben. In einigen Darstellungen werden die Bias-Terme in Vektoren $\mathbf{b}$ und $\mathbf{c}$, die Gewichte zwischen den Schichten in einer Matrix $\mathbf{W}$ und die Gewichte innerhalb der Schichten in symmetrischen Matrizen $\mathbf{U}$ und $\mathbf{Y}$ (welche auf der Diagonalen Nullen enthalten) zusammengefasst, wodurch die Energiefunktion als

$$E(\mathbf{v},\mathbf{h}) = -\mathbf{h}^T\mathbf{W}\mathbf{v} - \frac{1}{2}\mathbf{v}^T\mathbf{U}\mathbf{v} - \frac{1}{2}\mathbf{h}^T\mathbf{Y}\mathbf{h} - \mathbf{b}^T\mathbf{v} - \mathbf{c}^T\mathbf{h} \tag{2.2}$$

angegeben werden kann. Dabei wird der Faktor $\frac{1}{2}$ oft weggelassen, da die Konstante in die Gewichtsmatrizen hineingezogen werden kann.

Mit dem dritten Term, welcher mit den Kanten innerhalb der Schicht der versteckten Variablen in Verbindung gebracht werden kann, enthält die Energiefunktion Ausdrücke, die mehr als eine versteckte Variable enthalten. Dies hat zur Folge, dass die in Abschnitt 1.3 beschriebene Faktorisierung nicht angewendet werden kann. Möchte man also die Parameter einer BM lernen und in diesem Zusammenhang den Gradienten der Log-Likelihood, welcher in (1.21) gegeben ist, approximieren, müssen dafür, wie in Abschnitt 1.4 im Allgemeinen für MRFs erläutert, zwei unabhängige Gibbs-Sampling-Prozeduren (eine in der positiven und eine in der negativen Phase) durchgeführt werden.

Die spezielle Form der Energiefunktion führt zu vergleichsweise einfachen Log-Likelihood-Gradienten (1.23). Zum Beispiel gilt für ein Gewicht $w_{ij}$ und Trainigsdaten $S = \{\mathbf{v}_1, \ldots, \mathbf{v}_N\}$

$$\begin{aligned}\sum_{\mathbf{v}\in S} \frac{\partial \log L(\boldsymbol{\theta}|\mathbf{v})}{\partial w_{ij}} &= \sum_{\mathbf{v}\in S}\left[-E_{p(\mathbf{h}|\mathbf{v})}\left[\frac{\partial E(\mathbf{v},\mathbf{h})}{\partial w_{ij}}\right] + E_{p(\mathbf{h}',\mathbf{v}')}\left[\frac{\partial E(\mathbf{v}',\mathbf{h}')}{\partial w_{ij}}\right]\right] \\ &= \sum_{\mathbf{v}\in S}\left[E_{p(\mathbf{h}|\mathbf{v})}\left[v_i h_j\right] - E_{p(\mathbf{h}',\mathbf{v}')}\left[v_i' h_j'\right]\right] \\ &= N\left[\langle v_i h_j\rangle_{p(\mathbf{h}|\mathbf{v})p_e(\mathbf{v})} - \langle v_i h_j\rangle_{p(\mathbf{h},\mathbf{v})}\right] \quad . \end{aligned} \tag{2.3}$$

Hier bezeichnet $p_e$ die empirischen Verteilung. Der Term $\langle v_i h_j\rangle_{p(\mathbf{h}|\mathbf{v})p_e(\mathbf{v})} = \frac{1}{N}\sum_{\mathbf{v}\in S}\sum_{\mathbf{h}} P(\mathbf{h}|\mathbf{v}) v_i h_i$ misst die Korrelation zwischen der $i$-ten sichtbaren und der $j$-ten latenten Variablen, wenn die sichtbaren Variablen auf die Trainingsdatenvektoren gesetzt werden und die verstecken Variablen ihre Werte gemäß der entsprechenden bedingten Wahrscheinlichkeitsverteilung annehmen, und wird häufig mit $\langle v_i h_j\rangle_{\text{data}}$ bezeichnet. Der Term $\langle v_i h_j\rangle_{p(\mathbf{h},\mathbf{v})} = \frac{1}{N}\sum_{\mathbf{v},\mathbf{h}} p(\mathbf{v},\mathbf{h}) v_i h_i$ misst entsprechend die Korrelation zwischen $v_i$ und $h_j$

im aktuellen Modell mit Parametern $\boldsymbol{\theta}$. Er wird häufig mit $\langle v_i h_j \rangle_{\text{model}}$ oder $\left\langle v_i^{(\infty)} h_j^{(\infty)} \right\rangle$ bezeichnet, wobei die zweite Schreibweise andeutet, dass die Modellverteilung durch $\infty$-schrittiges Gibbs-Sampling exakt realisiert wird. Damit ergibt sich die intuitive Lernvorschrift

$$\sum_{\mathbf{v} \in S} \frac{\partial \log L(\boldsymbol{\theta}|\mathbf{v})}{\partial w_{ij}} \propto \langle v_i h_j \rangle_{\text{data}} - \langle v_i h_j \rangle_{\text{model}} \quad . \tag{2.4}$$

Da in den einzelnen Gibbs-Sampling-Schritten die lokalen Charakteristiken, bzw. die bedingten Wahrscheinlichkeitsverteilungen einzelner Zufallsvariablen gegeben die übrigen Zufallsvariablen, eine wichtige Rolle spielen, stellt sich nun die Frage, wie diese für BMs spezifiziert werden können. Im Folgenden soll beispielhaft (in Anlehnung an die Darstellungen in [17] die bedingte Verteilung einer beliebigen sichtbaren Zufallsvariable $V_l$, gegeben die übrigen Zufallsvariablen, hergeleitet werden (die lokale Charakteristik bezüglich einer versteckten Variablen kann man auf analoge Weise angeben). Sei dazu zunächst $\mathbf{V}_{-l}$ der Vektor, der alle sichtbaren Variablen bis auf $V_l$ enthält. Entsprechend ist die lokale Charakteristik bezüglich $V_l$ dann durch

$$p(v_l|\mathbf{v}_{-l}, \mathbf{h}) = P(V_l = v_l|\mathbf{V}_{-l} = \mathbf{v}_{-l}, \mathbf{H} = \mathbf{h}) = \frac{e^{-E(v_l, \mathbf{v}_{-l}, \mathbf{h})}}{\sum\limits_{v_l' \in \{0,1\}} e^{-E(v_l', \mathbf{v}_{-l}, \mathbf{h})}} \tag{2.5}$$

gegeben. Um eine vereinfachte Schreibweise zu erhalten, werden nun

$$\alpha(\mathbf{v}_{-l}, \mathbf{h}) = -\sum_{i=1}^{n} w_{il} h_i - \sum_{j=1}^{l-1} u_{lj} v_j - \sum_{j=l+1}^{m} u_{jl} v_j - b_l \quad , \tag{2.6}$$

definiert und durch $v_l \alpha(\mathbf{v}_{-l}, \mathbf{h})$ die Terme der Energiefunktion zusammengefasst, welche $v_l$ enthalten. Und die übrigen Terme der Energiefunktion sind durch

$$\begin{aligned} \beta(\mathbf{v}_{-l}, \mathbf{h}) = &-\sum_{i=1}^{n} \sum_{j=1, j\neq l}^{m} h_i w_{ij} v_j - \sum_{k=1, k\neq l}^{m} \sum_{r=1, r\neq l}^{k-1} v_k u_{kr} v_r \\ &- \sum_{k=1}^{n} \sum_{r=1}^{k-1} h_k y_{kr} h_r - \sum_{j=1, j\neq l}^{m} b_i v_i - \sum_{i=1}^{n} c_i h_i \end{aligned} \tag{2.7}$$

gegeben. Dann ergibt sich für $V_l = 1$

$$
\begin{aligned}
&p(v_l = 1 | \mathbf{v}_{-l}, \mathbf{h}) \\
&= \frac{\exp(-\beta(\mathbf{v}_{-l}, \mathbf{h}) - 1 \cdot \alpha(\mathbf{v}_{-l}, \mathbf{h}))}{\exp(-\beta(\mathbf{v}_{-l}, \mathbf{h}) - 1 \cdot \alpha(\mathbf{v}_{-l}, \mathbf{h})) + \exp(-\beta(\mathbf{v}_{-l}, \mathbf{h}) - 0 \cdot \alpha(\mathbf{v}_{-l}), \mathbf{h})} \\
&= \frac{\exp(-\beta(\mathbf{v}_{-l}, \mathbf{h})) \cdot \exp(-\alpha(\mathbf{v}_{-l}, \mathbf{h}))}{\exp(-\beta(\mathbf{v}_{-l}, \mathbf{h})) \cdot \exp(-\alpha(\mathbf{v}_{-l}, \mathbf{h})) + \exp(-\beta(\mathbf{v}_{-l}, \mathbf{h}))} \\
&= \frac{\exp(-\beta(\mathbf{v}_{-l}, \mathbf{h})) \cdot \exp(-\alpha(\mathbf{v}_{-l}, \mathbf{h}))}{\exp(-\beta(\mathbf{v}_{-l}, \mathbf{h})) \cdot (\exp(-\alpha(\mathbf{v}_{-l}, \mathbf{h})) + 1)} \\
&= \frac{\exp(-\alpha(\mathbf{v}_{-l}, \mathbf{h}))}{\exp(-\alpha(\mathbf{v}_{-l}, \mathbf{h})) + 1} = \frac{\frac{1}{\exp(\alpha(\mathbf{v}_{-l}, \mathbf{h}))}}{\frac{1}{\exp(\alpha(\mathbf{v}_{-l}, \mathbf{h}))} + 1} \\
&= \frac{1}{1 + \exp(\alpha(\mathbf{v}_{-l}, \mathbf{h}))} = \mathrm{sig}(-\alpha(\mathbf{v}_{-l}, \mathbf{h})) \ . \qquad (2.8)
\end{aligned}
$$

Dies ist gerade eine häufige Aktivierungsfunktion in künstlichen neuronalen Netzen, also die Gleichung, welche genutzt wird, um die Ausgabe eines Neurons in Abhängigkeit von der Eingabe, den aktuellen Werten der angrenzenden Neuronen, zu berechnen (siehe z. B. [18] und [16]). Aus diesem Grund kann eine BM auch als ein zweischichtiges neuronales Netz angesehen werden, werden die Knoten auch als Neuronen bezeichnet und eignen sich BMs besonders gut als Bausteine in tiefen neuronalen Architekturen.

Die lokale Charakteristik hat mit (2.8) eine Form, die im Gibbs-Sampling einfach berechnet werden kann. Da beim Parameter-Lernen für jedes Trainingsbeispiel jedoch zwei Sampling-Prozeduren benötigt werden, um den Log-Likelihood-Gradienten zu approximieren, ist der fürs Lernen benötigte Rechenaufwand trotzdem extrem hoch.

Die im Folgenden beschriebenen Restricted Boltzmann Maschinen bieten gegenüber allgemeinen BM die Vorteile, dass Gibbs-Sampling vereinfacht wird, dass generell nur eine Sampling-Prozedur zur Approximation des Log-Likelihood-Gradienten benötigt wird und dass darüber hinaus in neuerer Zeit effiziente Algorithmen zur Gradienten-Approximation entwickelt wurden.

## 2.1 Restricted Boltzmann Maschinen

Restricted Boltzmann Maschinen zeichnen sich durch eine eingeschränkte (daher ihr Name) Konnektivität zwischen den Knoten des zugehörigen Graphen aus. In einer Restricted Boltzmann Maschine (RBM) existieren zwar Kanten zwischen der Schicht der sichtbaren und der Schicht der versteckten Knoten, aber keine innerhalb der Schichten. Der bipartite ungerichtete Graph einer RBM ist in Abbildung 2.1 zu sehen.

Die RBM wurde zuerst 1986 unter dem Namen 'Harmonium' beschrieben [19]. Das Fehlen von lateralen Verbindungen innerhalb der sichtbaren und der versteckten Schicht hat zur Folge, dass die Energiefunktion eine einfachere Form ohne quadratische Terme hat:

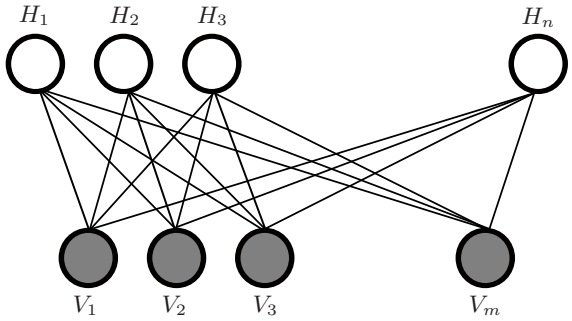


**Abb. 2.1.** Ungerichteter Graph einer RBM. Die sichtbaren Variablen sind durch grau ausgefüllte Knoten, die versteckten Variablen durch unausgefüllte Knoten dargestellt.

$$E(\mathbf{v}, \mathbf{h}) = -\sum_{i=1}^{n}\sum_{j=1}^{m} h_i w_{ij} v_j - \sum_{j=1}^{m} v_j b_j - \sum_{i=1}^{n} h_i c_i \ , \tag{2.9}$$

wobei $w_{ij}$ wiederum das Gewicht zwischen dem $j$-ten verdeckten und dem $i$-ten sichtbaren Neuron, $b_j$ den Bias-Term $j$-ten sichtbaren Knotens und $c_i$ den Bias-Term des $i$-ten versteckten Knotens bezeichnet. Mit $\beta(\mathbf{v}) = \sum_{j=1}^{m} v_j b_j$ und $\alpha_i(\mathbf{v}, h_i) = h_i(\sum_{j=1}^{m} w_{ij} v_j + c_i)$ kann die Energiefunktion in die Darstellung (1.13) überführt werden, und damit ist es möglich, die in Abschnitt 1.3 beschriebene Faktorisierung anzuwenden. Dies hat einerseits zur Folge, dass zur Approximation des Log-Likelihood-Gradienten nur ein Erwartungswert approximiert werden muss, wie in Abschnitt 1.4 allgemein für MRFs erläutert wurde und in Abschnitt 2.3 für den für RBMs spezifischen Gradienten konkretisiert wird. Andererseits führt die Faktorisierung auch dazu, dass die bedingte Wahrscheinlichkeitsverteilung von $\mathbf{H}$ gegeben $\mathbf{V}$ effizient berechnet werden kann:

$$
\begin{aligned}
p(\mathbf{h}|\mathbf{v}) &= \frac{\exp(\sum_{i=1}^{n}\sum_{j=1}^{m} h_i w_{ij} v_j + \sum_{j=1}^{m} v_j b_j + \sum_{i=1}^{n} h_i c_i)}{\sum_{\mathbf{h}'} \exp(\sum_{i=1}^{n}\sum_{j=1}^{m} h'_i w_{ij} v_j + \sum_{j=1}^{m} v_j b_j + \sum_{i=1}^{n} h'_i c_i)} \\
&= \frac{\exp(\sum_{j=1}^{m} v_j b_j) \prod_{i=1}^{n} \exp(\sum_{j=1}^{m} h_i w_{ij} v_j + h_i c_i)}{\exp(\sum_{j=1}^{m} v_j b_j) \prod_{i=1}^{n} \sum_{h'_i} \exp(\sum_{j=1}^{m} h'_i w_{ij} v_j + h'_i c_i)} \\
&= \frac{\prod_{i=1}^{n} \exp(h_i(\sum_{j=1}^{m} w_{ij} v_j + c_i))}{\prod_{i=1}^{n} \sum_{h'_i} \exp(h'_i(\sum_{j=1}^{m} w_{ij} v_j + c_i))} = \prod_{i=1}^{n} p(h_i|\mathbf{v})
\end{aligned} \tag{2.10}
$$

Dass diese Gleichung für die bedingte Verteilung gelten muss, kann auch aus der Struktur des Graphen einer RBM geschlossen werden. Das Fehlen von Kanten innerhalb der verdeckten Schicht führt dazu, dass je zwei versteckte Neuronen durch die sichtbaren Neuronen separiert werden. Mit der globalen Markov-Eigenschaft von MRFs mit Gibbsverteilung folgt nun dass die versteckten Knoten bedingt unabhängig gegeben die Neuronen der sichtbaren Schicht sind, was mathematisch eben durch $p(\mathbf{h}|\mathbf{v}) = \prod_{i=1}^{n} p(h_i|\mathbf{v})$ ausgedrückt wird.

Sowohl aus der Tatsache, dass es innerhalb der sichtbaren Schicht keine Kanten gibt, als auch daraus, dass $\mathbf{x}$ und $\mathbf{h}$ symmetrische Rollen in der Energiefunktion spielen, folgt, dass die bedingte Wahrscheinlichkeitsverteilung von $\mathbf{V}$ gegeben $\mathbf{H}$ ebenfalls faktorisiert werden kann, dass also gilt

$$
p(\mathbf{v}|\mathbf{h}) = \prod_{j=1}^{m} p(v_j|\mathbf{h}) \ . \tag{2.11}
$$

Im allgemeinen Fall einer RBM mit binären Einheiten (zu anderen Formen von RBMs siehe Abschnitt 2.5) können die lokalen Charakteristiken auf analoge Weise wie für generelle BMs bestimmt werden. So ist $p(h_i|\mathbf{v})$ durch

$$
p(h_i = 1|\mathbf{v}) = P(H_i = 1|\mathbf{V} = \mathbf{v}) = \mathrm{sig}(\sum_{j=1}^{m} w_{ij} v_j + c_i) \tag{2.12}
$$

und $p(v_j|\mathbf{h})$ durch

$$
p(v_j = 1|\mathbf{h}) = P(V_j = 1|\mathbf{H} = \mathbf{h}) = \mathrm{sig}(\sum_{i=1}^{n} w_{ij} h_i + b_j) \tag{2.13}
$$

beschrieben. Es wird also deutlich, dass die bedingte Wahrscheinlichkeitsverteilung über eine Schicht des RBM gegeben die andere Schicht auf sehr effiziente Weise berechnet werden kann.

Die bedingte Unabhängigkeit der Variablen einer Schicht bewirkt außerdem, dass das Erzeugen von Stichproben der gemeinsamen Wahrscheinlichkeitsverteilung einer RBM durch Gibbs-Sampling viel weniger Rechenaufwand benötigt als bei generellen BM: Anstatt für alle Zufallsvariablen neue Werte in einzelnen Zwischenschritten zu ziehen, kann Gibbs-Sampling in RBMs mit nur zwei Zwischenschritten durchgeführt werden: Im ersten Schritt werden neue Stichproben für alle latenten Zufallsvariablen $\mathbf{H}$ basierend auf dem aktuellen Wert $\mathbf{v}$ gezogen, im zweiten Schritt neue Werte für die sichtbaren Zufallsvariablen $\mathbf{V}$ basierend auf dem aktuellen Wert $\mathbf{h}$ erzeugt. Ausgehend von einem Wert $\mathbf{v}^{(0)}$ hat das Gibbs-Sampling mit $k$ Sampling-Schritten dann folgende Form:

$$\begin{aligned}
\mathbf{h}^{(0)} &\sim p(\mathbf{h}|\mathbf{v}^{(0)}) \\
\mathbf{v}^{(1)} &\sim p(\mathbf{v}|\mathbf{h}^{(0)}) \\
\mathbf{h}^{(1)} &\sim p(\mathbf{h}|\mathbf{v}^{(1)}) \\
&\dots \\
\mathbf{v}^{(k)} &\sim p(\mathbf{v}|\mathbf{h}^{(k-1)})
\end{aligned}$$

Führt man das Gibbs-Sampling zur Approximation des Gradienten beim Parameter-Lernen durch, so macht es Sinn, die Markov-Kette ausgehend von einem Trainingsbeispiel $\mathbf{v}^{(0)}$ zu starten bzw. $\mathbf{v}^{(0)}$ aus der zu modellierenden Verteilung zu ziehen. Denn während des Trainings wird die Modell-Verteilung, welche ja die stationäre Verteilung der Markov-Kette ist, der Trainingsverteilung immer mehr angeglichen. Das bedeutet, dass die Verteilung der Trainingsbeispiele im Verlauf des Trainings schon relativ 'nah' an der stationären Verteilung der Markov-Kette liegen sollte. Damit sollte die Kette schneller konvergieren, wenn man die Trainingsverteilung als Ausgangsverteilung wählt, wie auch ein Vergleich mit der Konvergenzrate des allgemeinen Gibbs-Samplers vor Augen führt [2].

## 2.2 Universalität von RBMs*

Es stellt sich nun die Frage, wie gut eine beliebige Wahrscheinlichkeitsverteilung auf $\{0,1\}^n$ durch eine RBM approximiert werden kann, bzw. ob nur bestimmte Wahrscheinlichkeitsverteilungen durch RBMs modelliert werden können.

Freund und Haussler (1994) haben gezeigt, dass theoretisch für jedes $n$ jede Verteilung auf $\{0,1\}^n$ beliebig gut durch eine RBM mit $n$ sichtbaren und $2^n$ verdeckten Neuronen approximiert werden kann, dass RBMs also universelle Approximatoren sind. Le Roux und Bengio (2008) haben diese Aussage konkretisiert: Die Anzahl der benötigten Neuronen hängt von der Anzahl der Vektoren aus $\{0,1\}^n$ ab, welchen die Verteilung eine positive Wahrscheinlichkeit zuordnet, wie folgendem Theorem zu entnehmen ist:

**Theorem 2.** *Gegeben sei eine Wahrscheinlichkeitsfunktion $q$ auf $\{0,1\}^n$. Sei $N$ die Anzahl der Vektoren $\mathbf{x} \in \{0,1\}^n$ mit $q(\mathbf{x}) > 0$. Dann existiert eine RBM mit $N+1$ versteckten Neuronen, mit welcher $q$ beliebig gut (im Sinne der KL Divergenz) approximiert werden kann.*

Außerdem wurde gezeigt, dass durch Hinzufügen einer versteckten Variable mit 'passendem' Bias-Term und Gewichten in jedem Fall eine Erhöhung der Log-Likelihood der Trainingsdaten unter dem Modell (bzw. eine Verkleinerung der KL-Divergenz zwischen der empirischen Verteilung und der Modellverteilung) erzielt wird, wenn durch die RBM nicht bereits ein perfektes Modell gegeben ist [20].

**Theorem 3.** *Sei $q$ eine beliebige Verteilung auf $\{0,1\}^n$ und $R_p$ ein beliebige RBM mit einer Marginalverteilung $p$ der sichtbaren Neuronen, für die $\mathrm{KL}(q||p) > 0$ gilt. So existiert eine RBM $R_{p_{w,c}}$, welche aus $R_p$ und einem zusätzlichen versteckten Neuronen mit Gewichtsvektor $w$ und Bias-Term $c$ besteht und für deren Marginalverteilung $p_{w,c}$ der sichtbaren Neuronen gilt: $\mathrm{KL}(q||p_{w,c}) < \mathrm{KL}(q||p)$ .*

## 2.3 Parameter-Lernen in RBMs

Um die Parameter in einer RBM zu lernen, werden, wie in Abschnitt 1.4 für MRFs im Allgemeinen erklärt, Optimierungsmethoden eingesetzt, welche auf dem Gradienten der Log-Likelihood basieren. Für die Gibbsverteilung eines MRFs mit Parameter-Vektor $\boldsymbol{\theta}$ und bzgl. eines Trainingsbeispiels $\mathbf{v}$ ist dieser, wie in (1.21) gezeigt wurde, durch

$$\frac{\partial \log L(\boldsymbol{\theta}|\mathbf{v})}{\partial \boldsymbol{\theta}} = -\sum_{\mathbf{h}} p(\mathbf{h}|\mathbf{v}) \frac{\partial E(\mathbf{v},\mathbf{h})}{\partial \boldsymbol{\theta}} + \sum_{\mathbf{v}'} p(\mathbf{v}') \sum_{\mathbf{h}'} p(\mathbf{h}'|\mathbf{v}') \frac{\partial E(\mathbf{v}',\mathbf{h}')}{\partial \boldsymbol{\theta}} \tag{2.14}$$

gegeben.

Um in konkreterer Form den Log-Likelihood-Gradienten für die Gibbsverteilung einer RBM anzugeben, muss die für RBM spezifische Energiefunktion und Parametrisierung berücksichtigt werden. Es werden nun nacheinander die Ableitungen der Log-Likelihood in Richtung der verschiedenen Parameter (Gewichte der Kanten zwischen sichtbaren und versteckten Neuronen, Bias-Terme der sichtbaren Neuronen und Bias-Terme der versteckten Neuronen) betrachtet.

Die Ableitung der Energiefunktion in Richtung des Gewichtes $w_{ij}$ des $i$-ten versteckten und des $j$-ten sichtbaren Neuronen ist durch

$$\frac{\partial E(\mathbf{v},\mathbf{h})}{\partial w_{ij}} = \frac{\partial}{\partial w_{ij}} \left( -\sum_{i=1}^{n}\sum_{j=1}^{m} h_i w_{ij} v_j - \sum_{j=1}^{m} v_j b_j - \sum_{i=1}^{n} h_i c_i \right) = -h_i v_j \tag{2.15}$$

gegeben. Damit lässt sich die Ableitung der Log-Likelihood in Richtung des Gewichtes $w_{ij}$ wie folgt angeben:

$$\frac{\partial \log L(\mathbf{W}, \mathbf{b}, \mathbf{c}|\mathbf{v})}{\partial w_{ij}} = \sum_{\mathbf{h}} p(\mathbf{h}|\mathbf{v}) h_i v_j - \sum_{\mathbf{v}'} p(\mathbf{v}') \sum_{\mathbf{h}'} p(\mathbf{h}'|\mathbf{v}') h'_i v'_j \ , \quad (2.16)$$

wobei die Matrix $\mathbf{W}$ alle Gewichte enthält und in den Vektoren $\mathbf{b}$ und $\mathbf{c}$ die Bias-Terme zusammengefasst sind. Der erste Term der Ableitung lässt sich vereinfachen, indem man die Faktorisierung der bedingten Wahrscheinlichkeitsverteilung der versteckten Variablen gegeben die sichtbaren Variablen nutzt (siehe (1.14)) und erneut faktorisiert:

$$\begin{aligned}
\sum_{\mathbf{h}} p(\mathbf{h}|\mathbf{v}) h_i v_j &= \sum_{\mathbf{h}} \prod_{i=1}^{n} p(h_i|\mathbf{v}) h_i v_j \\
&= \sum_{h_1} \sum_{h_2} ... \sum_{h_n} \prod_{k=1}^{n} p(h_k|\mathbf{v}) h_i v_j \\
&= \prod_{k=1, k\neq i}^{n} \left( \sum_{h_k} p(h_k|\mathbf{v}) \right) \cdot \sum_{h_i} p(h_i|\mathbf{v}) h_i v_j \\
&= \sum_{h_i} p(h_i|\mathbf{v}) h_i v_j \\
&= p(h_i = 1|\mathbf{v}) v_j \\
&= \text{sig}(\sum_{j=1}^{m} w_{ij} v_j + c_i) v_j \ . \quad (2.17)
\end{aligned}$$

Im vorletzten Schritt wurde hier genutzt, dass $p(h_k|\mathbf{v})$ eine Wahrscheinlichkeitsverteilung ist, und somit $\sum_{h_k} p(h_k|\mathbf{v}) = 1$ gilt. Der letzte Schritt gilt nur für binäre RBMs, d. h. wenn $h_i \in \{0, 1\}$.

Die innere Summe im zweiten Term der Ableitung lässt sich auf identische Weise vereinfachen, wodurch sich die Ableitung der Log-Likelihood in Richtung $w_{ij}$ wie folgt angeben lässt:

$$\begin{aligned}
\frac{\partial \log L(\mathbf{W}, \mathbf{b}, \mathbf{c}|\mathbf{v})}{\partial w_{ij}} &= p(h_i = 1|\mathbf{v}) v_j - \sum_{\mathbf{v}'} p(\mathbf{v}') p(h_i = 1|\mathbf{v}') v'_j \\
&= p(h_i = 1|\mathbf{v}) v_j - E_{p(\mathbf{v}')} \left[ p(h_i = 1|\mathbf{v}') v'_j \right] \quad (2.18)
\end{aligned}$$

Auf analoge Weise ergibt sich für die Ableitung der Log-Likelihood in Richtung Bias-Terms $c_i$ des $i$-ten versteckten Knotens

$$\begin{aligned}\frac{\partial \log L(\mathbf{W},\mathbf{b},\mathbf{c}|\mathbf{v})}{\partial c_i} &= -\sum_{\mathbf{h}} p(\mathbf{h}|\mathbf{v})\frac{\partial E(\mathbf{v},\mathbf{h})}{\partial c_i} + \sum_{\mathbf{v}'} p(\mathbf{v}')\sum_{\mathbf{h}'} p(\mathbf{h}'|\mathbf{v}')\frac{\partial E(\mathbf{v}',\mathbf{h}')}{\partial c_i} \\ &= \sum_{\mathbf{h}} p(\mathbf{h}|\mathbf{v})h_i - \sum_{\mathbf{v}'} p(\mathbf{v}')\sum_{\mathbf{h}'} p(\mathbf{h}'|\mathbf{v}')h_i' \\ &= p(h_i=1|\mathbf{v}) - \sum_{\mathbf{v}'} p(\mathbf{v}')p(h_i=1|\mathbf{v}') \\ &= p(h_i=1|\mathbf{v}) - E_{p(\mathbf{v}')}\left[p(h_i=1|\mathbf{v}')\right] \quad . \end{aligned} \tag{2.19}$$

Und für die Ableitungen in Richtung des Bias-Terms $b_j$ des $j$-ten sichtbaren Knotens ergibt sich

$$\begin{aligned}\frac{\partial \log L(\mathbf{W},\mathbf{b},\mathbf{c}|\mathbf{v})}{\partial b_j} &= -\sum_{\mathbf{h}} p(\mathbf{h}|\mathbf{v})\frac{\partial E(\mathbf{v},\mathbf{h})}{\partial b_j} + \sum_{\mathbf{v}'} p(\mathbf{v}')\sum_{\mathbf{h}'} p(\mathbf{h}'|\mathbf{v}')\frac{\partial E(\mathbf{v}',\mathbf{h}')}{\partial b_j} \\ &= \sum_{\mathbf{h}} p(\mathbf{h}|\mathbf{v})v_j - \sum_{\mathbf{v}'} p(\mathbf{v}')\sum_{\mathbf{h}'} p(\mathbf{h}'|\mathbf{v}')v_j' \\ &= v_j\sum_{\mathbf{h}} p(\mathbf{h}|\mathbf{v}) - \sum_{\mathbf{v}'} p(\mathbf{v}')v_j'\sum_{\mathbf{h}'} p(\mathbf{h}'|\mathbf{v}') \\ &= v_j - \sum_{\mathbf{v}'} p(\mathbf{v}')v_j' \\ &= v_j - E_{p(\mathbf{v}')}\left[v_j'\right] \quad , \end{aligned} \tag{2.20}$$

wobei im vierten Schritt genutzt wird, dass die Summe $\sum_{\mathbf{h}} p(\mathbf{h}|\mathbf{v})$ eins ergibt. Die jeweiligen Erwartungswerte im zweiten Term der Ableitungen lassen sich nun durch Stichproben der Verteilung $p(\mathbf{v})$, welche durch Gibbs-Sampling näherungsweise erzeugt werden können, approximieren. Möchte man einen Gradientenaufstieg auf der Log-Likelihood durchführen, ist es möglich, die Approximation des Gradienten der Log-Likelihood bezüglich eines Trainingsbeispiels $\mathbf{v}^{(0)}$ durch eine weitere Näherung weniger rechenaufwändig zu machen: anstatt eine Reihe von Stichproben $\mathbf{v}_1^{(\infty)}.....\mathbf{v}_N^{(\infty)}$ zu erzeugen (indem man $N$ Markov-Ketten ausgehend von $\mathbf{v}^{(0)}$ laufen lässt bis diese die stationäre Verteilung erreicht haben) und den Erwartungswert $E\left[\frac{\partial E(\mathbf{v},\mathbf{h})}{\partial \theta}\right]$ durch den Mittelwert $\frac{1}{N}\sum_{i=1}^{N}\frac{\partial E(\mathbf{v}_i^{(\infty)},\mathbf{h})}{\partial \theta}$ zu approximieren, kann man den Erwartungswert durch eine einzelne Stichprobe $\mathbf{v}^{(\infty)}$ ersetzen. Die Approximation des Gradienten ist dann durch

$$-\sum_{\mathbf{h}^{(0)}} p(\mathbf{h}^{(0)}|\mathbf{v}^{(0)})\frac{\partial E(\mathbf{v}^{(0)},\mathbf{h}^{(0)})}{\partial \boldsymbol{\theta}} + \sum_{\mathbf{h}^{(\infty)}} p(\mathbf{h}^{(\infty)}|\mathbf{v}^{(\infty)})\frac{\partial E(\mathbf{v}^{(\infty)},\mathbf{h}^{(\infty)})}{\partial \boldsymbol{\theta}} \tag{2.21}$$

gegeben. Da die Parameter in vielen Schritten in Richtung des Log-Likelihood-Gradienten bezüglich eines Trainingsbeispiels (beim Online-Lernen), bzw. einiger oder aller Trainingsbeispiele (beim Minibatch- oder Batch-Lernen),

verschoben werden, findet dennoch im Laufe des Prozesses eine Art Mittelwertbildung statt. Durch die Näherung wird zwar zusätzliche Varianz in der Gradienten-Approximation erzeugt, aber es muss nur eine anstatt einer großen Anzahl von Markov-Ketten simuliert werden, was den Rechenaufwand verkleinert. Da die Markov-Kette jedoch aus vielen Schritten bestehen muss, um sicher zu gehen, dass die stationäre Verteilung erreicht wird, sind die Kosten dennoch zu groß. Die in neuerer Zeit entwickelten Algorithmen zum Parameter-Lernen in RBMs (siehe Abschnitt 2.4) führen aus diesem Grund eine weitere Näherung in der Approximation des Gradienten ein, die es unnötig macht, in jedem Schritt eine lange Markov-Kette zu erzeugen.

## 2.4 Lernen in RBMs basierend auf Approximationen des Log-Likelihood-Gradienten

Erste speziell für RBMs entwickelte Lernalgorithmen gab es bereits 1994 [21], wirklich effiziente Algorithmen wurden aber erst in jüngerer Zeit entwickelt. Diese Algorithmen haben gemeinsam, dass sie einen Gradientenaufstieg auf einer Approximation der Log-Likelihood durchführen und dass diese Approximation lange Markov-Ketten unnötig macht.

Die Idee von $k$-Schritt Contrastive Divergence (CD-$k$) [22] ist einfach: Anstatt den zweiten Term des Gradienten durch eine Stichprobe $\mathbf{v}^{(\infty)}$ der Modellverteilung zu approximieren (wozu die Konvergenz der Markov-Kette gegen die stationären Verteilung nötig wäre), lässt man die Markov-Kette ausgehend von einem Trainingsbeispiel $\mathbf{v}^{(0)}$ nur $k$-Schritte laufen, und betrachtet die Stichprobe $\mathbf{v}^{(k)}$ als Näherung von $\mathbf{v}^{(\infty)}$. Der Gradient (1.21) wird dann durch

$$\begin{aligned}&\mathrm{CD}_k(\boldsymbol{\theta}, \mathbf{v}^{(0)}) = \\ &-\sum_{\mathbf{h}^{(0)}} p(\mathbf{h}^{(0)}|\mathbf{v}^{(0)}) \frac{\partial E(\mathbf{v}^{(0)}, \mathbf{h}^{(0)})}{\partial \boldsymbol{\theta}} + \sum_{\mathbf{h}^{(k)}} p(\mathbf{h}^{(k)}|\mathbf{v}^{(k)}) \frac{\partial E(\mathbf{v}^{(k)}, \mathbf{h}^{(k)})}{\partial \boldsymbol{\theta}}\end{aligned} \tag{2.22}$$

approximiert (vergleiche (2.21)). Die Ableitungen in Richtung der einzelnen Parameter sind dann wie in Abschnitt 2.3 gegeben, wobei der Erwartungswert über $p(\mathbf{v})$ jeweils durch die konkrete Stichprobe $\mathbf{v}^{(k)}$ ersetzt wird. Durch die CD-$k$-Approximation entsteht i. d. R. ein Fehler, der mit $k \to \infty$ verschwindet. Dies wird dadurch deutlich, dass CD-$k$ eigentlich nicht die Likelihood maximiert, was, wie in Abschnitt 1.4 dargestellt, einer Minimierung der KL-Divergenz zwischen der empirischen Verteilung $p_e$ und der Modellverteilung $p_\infty$ entspricht, sondern die Differenz zweier KL-Divergenzen minimiert [22]:

$$\mathrm{KL}(p_e \| p_\infty) - \mathrm{KL}(p_k \| p_\infty) \ , \tag{2.23}$$

wobei $p_k$ die Verteilung über den sichtbaren Variablen nach $k$ Schritten der Markov-Kette ist. Ist $k$ so groß, dass bereits die stationäre Verteilung der

Markov-Kette erreicht ist, so gilt $p_k = p_\infty$ und damit $\mathrm{KL}(p_k|p_\infty) = 0$, wodurch das Bias der CD-k-Approximation verschwindet.

Der Fehler durch die $k$-Schritt-Approximation kann zur Folge haben, dass die Maximum-Likelihood-Schätzer der Modellparameter nicht gefunden werden. So hat MacKay [23] Beispiele für Energiefunktionen und durch Übergangsmatrizen spezifizierte Markov-Ketten gegeben, bei denen die CD-Approximation des Likelihood-Gradienten nicht zur Konvergenz gegen die Maximum-Likelihood-Schätzer führt.

In Anlehnung an Resultate für stochastische Approximations-Methoden konnten Bedingungen für eine Konvergenz von CD-$k$ gegen eine optimale Lösung formuliert werden [24]. Um nachprüfen zu können, ob diese erfüllt sind, ist jedoch vorausgesetzt, dass man die optimale Lösung bereits kennt, und dass die $m$-Schritt-Übergangsmatrix angegeben werden kann. Diese beiden Voraussetzungen sind für 'größere' RBMs nicht erfüllt, da weder die Maximum-Likelihood-Schätzer noch alle $m$-Schritt-Übergangswahrscheinlichkeiten effizient berechnet werden können. Schon die 1-Schritt-Übergangswahrscheinlichkeit von einem sichtbaren Vektor $\mathbf{v}_1$ zu einem sichtbaren Vektor $\mathbf{v}_2$, die durch

$$P(\mathbf{v}_2|\mathbf{v}_1) = \sum_{\mathbf{h}} p(\mathbf{v}_2|\mathbf{h})p(\mathbf{h}|\mathbf{v}_1) \tag{2.24}$$

gegeben ist, enthält bei $n$ versteckten Neuronen $2^n$ Summanden, und bei $m$ sichtbaren Neuronen sind $(2^m)^2$ verschiedene Übergänge möglich sind.

Ein erstaunliches empirisches Resultat ist, dass trotz des Fehlers der CD-$k$-Approximation häufig sogar $k = 1$ gute Lernresultate liefert. Experimente, welche den Lernerfolg nach dem Training kleiner RBMs mit CD-$k$ und dem wahren Log-Likelihood-Gradienten vergleichen, sind in [25, 26] zu finden. Dabei hat man sich zu Nutze gemacht, dass der Log-Likelihood-Gradient exakt berechnen werden kann, wenn die Anzahl der sichtbaren oder (da bei RBMs mit weniger versteckten als sichtbaren Neuronen eine (1.14)) entsprechende Faktorisierung über die sichtbaren Neuronen durchgeführt werden kann) der versteckten Neuronen klein genug ist. In diesen Experimenten werden kleine RBMs mit einfachen empirischen Verteilungen trainiert, und es zeigt sich, dass die Optima, welche durch CD-Lernen gefunden werden nicht unbedingt den Optima entsprechen, welche man bei exakter Berechnung des Gradienten erhält. Bei den untersuchten Lernproblemen führt das Bias von CD-1 häufig zu einer geringeren Likelihood der Trainingsdaten unter der Modellverteilung. Der Unterschied zu den auf dem exakten Gradienten basierenden Optima ist jedoch gering [25]. Je größer der Wert $k$ gewählt wird, desto bessere Ergebnisse werden durch CD-$k$ erzielt [26].

Die theoretischen Resultate in [26] liefern die Grundlage für ein besseres Verständnis der CD-$k$-Approximation und des von CD gemachten Approximationsfehlers. Bengio und Dellaleau zeigen, dass der Log-Likelihood-Gradient basierend auf einer Markov-Kette als eine Summe dargestellt werden kann, welche die $k$-te Stichprobe der Markov-Kette enthält.

**Theorem 4.** *Gegeben sei eine konvergierende Markov-Kette* $\mathbf{v}^{(0)} \Rightarrow \mathbf{h}^{(0)} \Rightarrow \mathbf{v}^{(1)} \Rightarrow \mathbf{h}^{(1)} \dots$ *welche von einem Trainingsbeispiel* $\mathbf{v}^{(0)}$ *ausgeht. Dann kann der Log-Likelihood-Gradient wie folgt geschrieben werden:*

$$\begin{aligned}\frac{\partial \log L(\boldsymbol{\theta}|\mathbf{v}^{(0)})}{\partial \boldsymbol{\theta}} = &- \sum_{\mathbf{h}^{(0)}} p(\mathbf{h}^{(0)}|\mathbf{v}^{(0)}) \frac{\partial E(\mathbf{v}^{(0)}, \mathbf{h}^{(0)})}{\partial \boldsymbol{\theta}} \\ &+ E_{p(\mathbf{v}^{(k)}|\mathbf{v}^{(0)})} \left[ \sum_{\mathbf{h}^{(k)}} p(\mathbf{h}^{(k)}|\mathbf{v}^{(k)}) \frac{\partial E(\mathbf{v}^{(k)}, \mathbf{h}^{(k)})}{\partial \boldsymbol{\theta}} \right] \\ &+ E_{p(\mathbf{v}^{(k)}|\mathbf{v}^{(0)})} \left[ \frac{\partial \log p(\mathbf{v}^{(k)})}{\partial \boldsymbol{\theta}} \right] \quad .\end{aligned}$$

*Der letzte Term konvergiert gegen Null, wenn k gegen unendlich geht.*

Dabei bezeichnet $p(\mathbf{v}^{(k)}|\mathbf{v}^{(0)})$ die Wahrscheinlichkeitsverteilung über die sichtbaren Neuronen im $k$-ten Schritt der Markov-Kette. Diese ist natürlich von $\mathbf{v}^{(0)}$ abhängig.

Da der letzte Term der Summen-Entwicklung des Log-Likelihood-Gradienten mit größerem $k$ klein wird, scheint es gerechtfertigt, die Markov-Kette nur $k$ Schritte laufen zu lassen, und den Gradienten nur durch die ersten beiden Terme

$$-\sum_{\mathbf{h}^{(0)}} p(\mathbf{h}^{(0)}|\mathbf{v}^{(0)}) \frac{\partial E(\mathbf{v}^{(0)}, \mathbf{h}^{(0)})}{\partial \boldsymbol{\theta}} + E_{p(\mathbf{v}^{(k)}|\mathbf{v}^{(0)})} \left[ \sum_{\mathbf{h}^{(k)}} p(\mathbf{h}^{(k)}|\mathbf{v}^{(k)}) \frac{\partial E(\mathbf{v}^{(k)}, \mathbf{h}^{(k)})}{\partial \boldsymbol{\theta}} \right] \tag{2.25}$$

zu approximieren. Der Erwartungswert im zweiten Term kann durch eine Stichprobe von $p(\mathbf{v}^{(k)})$ ersetzt werden, was zu einer stochastischen Gradienten-Approximation führt, deren Erwartungswert wieder (2.25) ist. Dies entspricht gerade der CD-$k$-Approximation. Dadurch wird deutlich, dass der letzte Term der Summen-Entwicklung der CD-Fehler ist, und mit Theorem 2.5.1 ist damit auch eine theoretische Grundlage für die empirischen Beobachtungen gegeben, dass ein größeres $k$ zu einem kleineren Bias und besseren Ergebnissen führt.

Der Fehler der CD-Approximation hängt jedoch nicht nur von $k$ sondern auch von der Geschwindigkeit der Konvergenz gegen die stationäre Verteilung ab. Dies ist intuitiv einleuchtend: Ist eine schnelle Konvergenz gegen die stationäre Verteilung der Markov-Kette gegeben, sollten weniger Schritte der Markov-Kette 'in die Nähe' der stationären Verteilung führen als bei einer kleinen Konvergenzrate. Die Geschwindigkeit der Konvergenz steht in Verbindung mit der Mischungsrate der Markov-Kette, also damit, wie schnell oder langsam die Markov-Kette die verschiedenen Zustände des Zustandsraums 'durchwandert'. Bei einer Markov-Kette, welche die Zustände einer RBM beschreibt, ist die Mischungsrate von der Größe der Modellparameter, genau genommen von der Größe des Betrags der Modellparameter, abhängig. Dies wurde bereits beschrieben [22, 25], und der Zusammenhang wird klar, wenn bedacht wird, dass die Wahrscheinlichkeiten $P(H_i = 1|\mathbf{v})$ und $P(V_j = 1|\mathbf{h})$

über die Sigmoid-Funktion von den Parametern des Modells abhängen. Im Betrag große Parameter führen nun dazu, dass die bedingten Wahrscheinlichkeiten entweder sehr groß oder sehr klein sind, also nahe an 0 oder 1 liegen, was wiederum dazu führt, dass es zunehmend vorhersagbar wird, welche Werte aus den Verteilungen $p(h_i|\mathbf{v})$ und $p(v_j|\mathbf{h})$ gezogen werden. Die Markov-Kette ändert ihren Zustand also langsam und die Mischungsrate ist klein. Bengio und Dellaleau [26] haben den Zusammenhang von betragsmäßig großen Gewichten und dem Fehler von CD-$k$ empirisch untersucht. Außerdem ist es ihnen gelungen, eine Abschätzung des Fehler-Terms anzugeben, welche die Abhängigkeit deutlich mach: Eine obere Schranke für den bei der Approximation der Ableitung in Richtung eines Parameters $\theta$ (also für eine einzelne Komponente der Gradienten-Aproximation) gemachten Fehler ist durch

$$\left| E_{p(\mathbf{v}^{(k)}|\mathbf{v}^{(0)})} \left[ \frac{\partial \log p(\mathbf{v}^{(k)})}{\partial \theta} \right] \right| \leq 2^m (1 - 2^m 2^n \operatorname{sig}(-\alpha)^m \operatorname{sig}(-\beta)^n)^{k-1} \quad (2.26)$$

gegeben, wobei $\alpha = \max_j \sum_i |w_{ij}| + |b_j|$ und $\beta = \max_i \sum_j |w_{ij}| + |c_i|$ gilt.

Sind die Modellparameter auf Null gesetzt, ist die Schranke für $k \geq 2$ gleich 0, aber mit wachsendem Betrag der Parameter wird sie sehr schnell schwach und nähert sich der Anzahl $2^m$ der möglichen Konfigurationen der sichtbaren Variablen. Die Experimente, welche den exakten Log-Likelihood-Gradienten und den Erwartungswert von CD-$k$ (in kleinen RBMs, wo beide Ausdrücke berechnet werden können) vergleichen, zeigen jedoch auch, dass die Qualität von CD-$k$ als Approximation des Gradienten mit (betragsmäßig) wachsenden Parametern abnimmt. Trotzdem führt das Training der RBMs zu einem guten Modell der 'einfachen' empirischen Verteilungen. Zusätzliche Ergebnisse dieser Untersuchungen sind, dass die Verzerrung von CD-$k$ mit einer größeren Dimension der Trainingsdaten (und das bedeutet mit einer größeren Anzahl von sichtbaren Neuronen) zunimmt, dass der Anteil der Parameter-Updates, bei welchen der Log-Likelihood-Gradient und der Erwartungswert der CD-Approximation sich im Vorzeichen unterscheiden, jedoch klein bleibt.

## 2.5 RBMs mit stetigen Variablen*

Bei Problemen des unüberwachten Lernens, die in der Praxis auftauchen, sollen in der Regel Verteilungen über reellwertige und nicht binären Eingabedaten (z. B. Grauwerten von Pixeln eines Bildes) gelernt werden. Restricted Boltzmann Machines mit binären Neuronen mit einem 'Trick' auch dazu genutzt werden, stetige Verteilungen auf reellwertigem Input zu modellieren. Dazu werden die Eingabedaten so skaliert, dass sie in dem Intervall $(0, 1)$ liegen. Stichproben der sichtbaren Neuronen können dann im RBM erzeugt werden, indem die Wahrscheinlichkeiten, dass die im ursprünglichem Sinne binären Neuronen den Wert 1 annehmen (welche durch $p(v_i = 1|\mathbf{h}), i = 1, \ldots, n$ gegeben sind), einfach als reellwertige Zustände der sichtbaren Variablen angesehen werden. Wie Hintons Implementation zu entnehmen ist, wurden auf diese

Weise die RBMs eines DBNs auf die Graustufen-Bilder von handgeschriebenen Zahlen aus der MNIST Datenbank trainiert (wobei in der RBM der ersten Schicht die Grauwerte der Bilder die Eingabe darstellten, und in den höheren Schichten der Erwartungswert $E[p(\mathbf{h}|\mathbf{x})] = p(\mathbf{h} = 1|\mathbf{x})$ der versteckten Neuronen aus der darunter gelegenen Schicht als Eingabe diente). Ein alternativer Ansatz, die MNIST Daten zu lernen, ist es, die Grauwerte als Wahrscheinlichkeiten einer Bernoulliverteilung anzusehen, und binäre Daten aus dieser Verteilung zu ziehen [27].

Es besteht aber auch die Möglichkeit, Zufallsvariablen, welche Werte in einem Intervall $I$ in $\mathbb{R}$ annehmen können, durch RBMs mit linearer Energiefunktion $E$ zu modellieren, indem die Menge der erlaubten Zustände wirklich von $\{0, 1\}$ auf $I$ erweitert wird. Dies hat Auswirkungen auf die Form der lokalen Charakteristiken und damit auch auf die Sampling-Prozeduren, wie im Folgenden erläutert werden soll. Sei $V_l$ eine beliebige Zufallsvariable in der sichtbaren Schicht des RBMs, und die übrigen sichtbaren Variablen seinen durch den Vektor $\mathbf{V}_{-l}$ gegeben. Die Terme der Energiefunktion $E$, die $v_l$ enthalten, können durch $v_l\alpha(\mathbf{h})$ zusammenfasst werden, wobei $\alpha(\mathbf{h}) = -\sum_{j=1}^{m} h_i w_{il} - b_l$ gilt. Die übrigen Terme von $E$ sind durch $\beta(\mathbf{v}_{-l}, \mathbf{h}) = -\sum_{i=1, i\neq l}^{n}\sum_{j=1}^{m} h_i w_{ij} v_j - \sum_{i=1}^{n} h_i c_i$ gegeben. Damit lässt sich die bedingte Verteilung von $V_l$, gegeben die versteckten Variablen, in einem stetigen Analogon zu (2.8) wie folgt bestimmen:

$$\begin{aligned}
p(v_l|\mathbf{h}) &= \frac{e^{-E(v_l,\mathbf{v}_{-l},\mathbf{h})}\mathbf{1}_{v_l\in I}}{\int_{v_l'} e^{-E(v_l',\mathbf{v}_{-l},\mathbf{h})}\mathbf{1}_{v_l\in I}dv_l'} \\
&= \frac{\exp(-\beta(\mathbf{v}_{-l},\mathbf{h}) - v_l\alpha(\mathbf{h}))\mathbf{1}_{v_l\in I}}{\int_{v_l'}\exp(-\beta(\mathbf{v}_{-l},\mathbf{h}) - v_l'\alpha(\mathbf{h}))\mathbf{1}_{v_l\in I}dv_l'} \\
&= \frac{\exp(-\beta(\mathbf{v}_{-l},\mathbf{h}))\cdot\exp(-v_l\alpha(\mathbf{h}))\mathbf{1}_{v_l\in I}}{\int_{v_l'}\exp(-\beta(\mathbf{v}_{-l},\mathbf{h}))\cdot\exp(-v_l'\alpha(\mathbf{h}))\mathbf{1}_{v_l\in I}dv_l'} \\
&= \frac{\exp(-\beta(\mathbf{v}_{-l},\mathbf{h}))\cdot\exp(-v_l\alpha(\mathbf{h}))\mathbf{1}_{v_l\in I}}{\exp(-\beta(\mathbf{v}_{-l},\mathbf{h}))\cdot\int_{v_l'}\exp(-v_l'\alpha(\mathbf{h}))\mathbf{1}_{v_l\in I}dv_l'} \\
&= \frac{\exp(-v_l\alpha(\mathbf{h}))\mathbf{1}_{v_l\in I}}{\int_{v_l'}\exp(-v_l'\alpha(\mathbf{h}))\mathbf{1}_{v_l\in I}dv_l'} \ . \qquad (2.27)
\end{aligned}$$

Mit $I = [0, \infty)$ ist dies die Exponentialverteilung mit dem Parameter $\alpha(\mathbf{h})$, denn es gilt

$$\int_0^\infty \exp(-v_l'\alpha(\mathbf{h}))\mathbf{1}_{v_l\in I}dv_l' = \left[-\frac{1}{\alpha(\mathbf{h})}\exp(-v_l'\alpha(\mathbf{h}))\right]_0^\infty = \frac{1}{\alpha(\mathbf{h})} \qquad (2.28)$$

und damit $p(v_l|\mathbf{h}) = \alpha(\mathbf{h})e^{-\alpha(\mathbf{h})v_l}$. Es ist also möglich, Stichproben der einzelnen sichtbaren Zufallsvariablen, und auf analoge Weise auch der einzelnen versteckten Zufallsvariablen, (gegeben die Variablen der anderen Schicht) einfach zu erzeugen, indem aus dieser Exponentialverteilung gezogen wird.

Ist $I$ hingegen ein abgeschlossenes Intervall, z. B. $I = [0, 1]$, so sind die lokalen Charakteristiken durch gestutzte Exponentialverteilungen gegeben. Um eine Stichprobe einer solchen Verteilung zu erzeugen, kann die inverse Transformationsmethode, welche z. B. in [28] erklärt ist, benutzt werden.

Sowohl wenn $I = [0, \infty)$ gilt als auch wenn $I$ abgeschlossen ist, bleibt die Approximation des Log-Likelihood-Gradienten die selbe wie für RBMs mit binären Neuronen, da diese ja nur auf den Ableitungen der Energiefunktion in Richtung der Parameter beruht.

Wird die Struktur des ungerichteten Graphen einer RBM beibehalten, aber andere Energiefunktionen zugelassen, können RBMs definiert werden, deren Zufallsvariablen durch andere Verteilungen beschrieben sind. Eine generelle Formulierung von RBMs, durch die für $p(v_j|\mathbf{h})$ und $p(h_i|\mathbf{v})$ jede beliebige Verteilung der Exponentiellen Familie gewählt werden kann, ist in [29] gegeben. Einheiten, deren lokale Charakteristiken Normalverteilungen entsprechen, können z. B. dadurch erzeugt werden, dass die lineare Energiefunktion durch quadratische Terme $\sum_{i=1}^{m} d_j v_j^2$, bzw. $\sum_{j=1}^{n} a_i h_i^2$ mit positivem Parametern $d_j$ und $a_i$ erweitert wird. Dass diese Erweiterung zu einer Normalverteilung führt, soll am Beispiel der bedingten Verteilung eines sichtbares Neurons $V_l$, gegeben die versteckten Neuronen, gezeigt werden. Dazu wird $\alpha(\mathbf{h})$ wie oben definiert, wodurch sich die Terme der Energiefunktion, welche $v_l$ enthalten, als $d_l v_l^2 + v_l \alpha(\mathbf{h})$ angeben lassen. Damit ergibt sich für die bedingte Verteilung

$$
\begin{aligned}
p(v_l|\mathbf{h}) = \frac{e^{-E(v_l, \mathbf{v}_{-l}, \mathbf{h})}}{\int_{-\infty}^{+\infty} e^{-E(v_l', \mathbf{v}_{-l}, \mathbf{h})} dv_l'} &= \frac{\exp(-d_l v_l^2 - v_l \alpha(\mathbf{h}))}{\int_{-\infty}^{+\infty} \exp(-d_l v_l'^2 - v_l' \alpha(\mathbf{h})) dv_l'} \\
&= \frac{\exp(-d_l(v_l^2 + \frac{1}{d_l} v_l \alpha(\mathbf{h})))}{\int_{-\infty}^{+\infty} \exp(-d_l(v_l'^2 + \frac{1}{d_l} v_l' \alpha(\mathbf{h}))) dv_l'} \\
&= \frac{\exp(-d_l(v_l + \frac{\alpha(\mathbf{h})}{2d_l})^2) * \exp(\frac{\alpha(\mathbf{h})^2}{4})}{\int_{-\infty}^{+\infty} \exp(-d_l(v_l' + \frac{\alpha(\mathbf{h})}{2d_l})^2) * \exp(\frac{\alpha(\mathbf{h})^2}{4}) dv_l'} \\
&= \frac{\exp(-\frac{1}{2\sigma^2}(v_l + \sigma^2 \alpha(\mathbf{h}))^2)}{\int_{-\infty}^{+\infty} \exp(-\frac{1}{2\sigma^2}(v_l' + \sigma^2 \alpha(\mathbf{h}))^2) dv_l'} \\
&= \frac{\exp(-\frac{1}{2\sigma^2}(v_l + \sigma^2 \alpha(\mathbf{h}))^2)}{\sqrt{2\pi\sigma^2} \int_{-\infty}^{+\infty} \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{1}{2\sigma^2}(v_l' - (-\sigma^2 \alpha(\mathbf{h})))^2) dv_l'} \\
&= \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{(v_l - (-\sigma^2 \alpha(\mathbf{h})))^2}{2\sigma^2}) \ . \qquad (2.29)
\end{aligned}
$$

Dabei wurden zunächst zu (2.27) analoge Umformungen durchgeführt. Dann wurde eine passende quadratische Ergänzung gewählt, $d = \frac{1}{2\sigma^2}$ gesetzt und schließlich genutzt, dass das Integral der Normalverteilung 1 ergibt. Die bedingte Verteilung ist also normalverteilt mit Erwartungswert

$$\sigma_{vl}^2 \left( \sum_{j=1}^{m} h_i w_{il} + b_l \right) \tag{2.30}$$

gegeben und die Varianz ergibt sich in Abhängigkeit von $d_l$ als $\sigma^2 = \frac{1}{2d_l}$.

Die Ableitung der Log-Likelihood in Richtung der linearen Parameter bleibt die selbe wie für binäre Neuronen, und die Ableitungen in Richtung der quadratischen Parameter $d_j$ und $a_i$ sind durch $2d_j v_j^2$ und $2a_i h_i^2$ gegeben.

# Literaturverzeichnis


1. Lauritzen, S.L.: Graphical Models. Oxford University Press (1996)
2. Brémaud, P.: Markov Chains: Gibbs Fields, Monte Carlo Simulation, and Queues. Springer (1999)
3. Besag, J.: Spatial interaction and the statistical analysis of lattice systems. Journal of the Royal Statistical Society. Series B (Methodological) **36**(2) (1974) 192–236
4. Grimmett, G.R.: A theorem about random fields. Bulletin of the London Mathematical Society **5** (1973) 721–741
5. Preston, C.: Generalized Gibbs states and Markov random fields. Advances in Applied Probability **5** (1973) 242–261
6. Sherman, S.: Markov random fields and Gibbs ensembles. Israeli Journal of Mathematics **14** (1973) 92–103
7. Hammersley, J.M., Clifford, P.: Markov fields on finite graphs and lattices. Unveröffentlichtes Manuskript (1968)
8. Lecun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., Jackel, L.D.: Backpropagation applied to handwritten zip code recognition. Neural Computation **1**(4) (1989) 541–551
9. LeCun, Y., Bottou, L., Orr, G., Müller, K.: Efficient backprop. In Orr, G., K., M., eds.: Neural Networks: Tricks of the Trade, Springer (1998)
10. Ackley, D.H., Hinton, G.E., Sejnowski, T.J.: A learning algorithm for Boltzmann machines. Cognitive Science **9** (1985) 147–169
11. Hinton, G.E., Sejnowski, T.J., Ackley, D.H.: Boltzmann machines: Constraint satisfaction networks that learn. Technical Report CMS-CS-84-119, CMU Computer Science Department (1984)
12. Hinton, G.E., Sejnowski, T.J.: Learning and relearning in Boltzmann machines. In Rumelhart, D.E., McClelland, J.L., eds.: Parallel distributed processing: explorations in the microstructure of cognition, vol. 1: Foundations. MIT Press (1986) 282–317
13. Geman, S., Geman, D.: Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. IEEE Transactions on Pattern Analysis and Machine Intelligence **6** (1984) 721–741
14. Andrieu, C., de Freitas, N., Doucet, A., Jordan, M.I.: An introduction to MCMC for machine learning. Machine Learning **50**(1) (2003) 5–43

15. Besag, J.: On the statistical analysis of dirty pictures. Journal of the Royal Statistical Society. Series B (Methodological) **48**(3) (1986) 259–302
16. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer (2006)
17. Bengio, Y.: Learning deep architectures for AI. Foundations and Trends in Machine Learning **2** (2009) 1–127
18. Haykin, S.: Neural Networks and Learning Machines. 3rd edition edn. Prentice Hall (2009)
19. Smolensky, P.: Information processing in dynamical systems: Foundations of harmony theory. In Rumelhart, D.E., McClelland, J.L., eds.: Parallel distributed processing: explorations in the microstructure of cognition, vol. 1: Foundations. MIT Press (1986) 194–281
20. Le Roux, N., Bengio, Y.: Representational power of restricted Boltzmann machines and deep belief networks. Neural Computation **20**(6) (2008) 1631–1649
21. Freund, Y., Haussler, D.: Unsupervised learning of distributions on binary vectors using two layer networks. Technical report, Baskin Center for Computer Engeneering & Information Sciences, University of California, Santa Cruz, USA (1994)
22. Hinton, G.E.: Training products of experts by minimizing contrastive divergence. Neural Computation **14**(8) (2002) 1771–1800
23. MacKay, D.J.C.: Failures of the one-step learning algorithm. Cavendish Laboratory, Madingley Road, Cambridge CB3 0HE, UK. http://www.cs.toronto.edu/ mackay/gbm.pdf (2001)
24. Yuille, A.: The convergence of contrastive divergence. In: Advances in Neural Processing Systems (NIPS 17). (2004) 1593–1600
25. Carreira-Perpiñán, M.Á., Hinton, G.E.: On contrastive divergence learning. Proceeding of the Tenth International Conference on Artificial Intelligence and Statistics (AISTATS 2005) (2005) 59–66
26. Bengio, Y., Delalleau, O.: Justifying and generalizing contrastive divergence. Neural Computation **21**(6) (2009) 1601–1621
27. Tieleman, T.: Training restricted Boltzmann machines using approximations to the likelihood gradient. In: International Conference on Machine learning (ICML). (2008) 1064–1071
28. Devroye, L.: Non-Uniform Random Variate Generation. Springer, New York (1986)
29. Welling, M., Rosen-Zvi, M., Hinton, G.: Exponential family harmoniums with an application to information retrieval. In Saul, L.K., Weiss, Y., Bottou, L., eds.: Advances in Neural Information Processing Systems (NIPS 17). MIT Press, Cambridge, MA (2005) 1481–1488