ЛЕКЦИЯ 2. Машины Больцмана и сети доверия

Энергетические модели

Многие интересные теоретические результаты о неориентированных моделях основаны на предположении о том, что $\forall x\, \tilde{p}(x) > 0$. Удобный способ наложить такое ограничение – воспользоваться энергетической моделью (energy-based model – EBM), в которой

$$\tilde{p}(x) = exp(-E(x)), \tag{1a}$$

а $E(x)$ называется функцией энергии. Поскольку $exp(z)$ положительно для всех z, гарантируется, что при любой функции энергии все состояния x будут иметь ненулевую вероятность. Возможность выбирать абсолютно произвольную функцию энергии упрощает обучение. Если бы мы обучали потенциалы клик непосредственно, то пришлось бы использовать ограниченную оптимизацию, чтобы задать неизвестно как выбранное минимальное значение вероятности. А обучая функцию энергии, мы можем пользоваться неограниченной оптимизацией. В энергетической модели вероятности могут быть сколь угодно близки к нулю, но никогда не обращаются в нуль.

Распределение вида (1a) называется распределением Больцмана. Поэтому многие энергетические модели называются машинами Больцмана. Не существует общепринятого соглашения о том, когда называть модель энергетической, а когда – машиной Больцмана. Термин «машина Больцмана» был введен для обозначения модели, в которой все переменные бинарные, но сегодня так называются многие модели с вещественными переменными, например ограниченные машины Больцмана с усреднением и ковариацией. Хотя в первоначальном определении машины Больцмана охватывали модели с латентными переменными и без них, в наши дни этот термин чаще всего относится к моделям с латентными переменными, а машины Больцмана без латентных переменных обычно называют случайными марковскими полями, или лог-линейными моделями.

Клики в неориентированном графе соответствуют факторам ненормированной функции вероятности. Поскольку $exp(a)exp(b) = exp(a + b)$, это означает, что разные клики соответствуют разным членам функции энергии. Иными словами, энергетическая модель – это просто частный случай марковской сети: операция потенцирования сопоставляет каждому члену функции энергии фактор отдельной клики. На рис. 1a показано, как получить функцию энергии по неориентированному графу. Энергетическую модель с функцией энергии, содержащей несколько членов, можно рассматривать как произведение экспертов. Каждый член функции энергии соответствует фактору распределения вероятности, его можно считать «экспертом», который определяет,

удовлетворяется ли некоторое мягкое ограничение. Каждый эксперт может налагать только одно ограничение, относящееся лишь к проекции случайных величин на пространство низкой размерности, но в сочетании с произведением вероятностей сообщество экспертов налагает сложное ограничение высокой размерности.

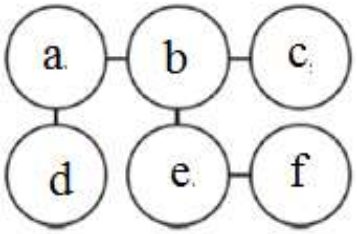

Рис. 1а. Из этого графа следует, что $E(a,b,c,d,e,f)$ можно записать в виде $Ea,b(a,b)+Eb,c(b,c)+Ea,d(a,d)+Eb,e(b,e)+Ee,f(e,f)$ при подходящем выборе функции энергии для каждой клики.

Энергетическую модель с функцией энергии, содержащей несколько членов, можно рассматривать как произведение экспертов. Каждый член функции энергии соответствует фактору распределения вероятности, его можно считать «экспертом», который определяет, удовлетворяется ли некоторое мягкое ограничение. Каждый эксперт может налагать только одно ограничение, относящееся лишь к проекции случайных величин на пространство низкой размерности, но в сочетании с произведением вероятностей сообщество экспертов налагает сложное ограничение высокой размерности.

В определении энергетической модели есть часть, не несущая никакой полезной функции с точки зрения машинного обучения: знак – (минус) в формуле (1а). Его можно было бы включить в определение E, поскольку во многих случаях алгоритм обучения вправе выбрать любой знак энергии. Знак – присутствует главным образом для того, чтобы сохранить совместимость между литературой по машинному обучению и по физике. Многими своими достижениями вероятностное моделирование обязано статистической физике, где E обозначает физическую энергию и не может иметь произвольного знака. Термины «энергия» и «статистическая сумма» заимствованы именно оттуда, хотя их математическое применение шире, чем в том физическом контексте, в котором они возникли. Некоторые специалисты по машинному обучению убирали знак минус, но это не стало стандартным соглашением.

Во многих алгоритмах для работы с вероятностными моделями нужно вычислять не $p_{model}(x)$, а только $\log\tilde{p}_{model}(x)$. В энергетических моделях с латентными переменными h такие алгоритмы иногда формулируются в терминах этой величины со знаком минус, называемой свободной энергией:

$$\mathcal{F}(x) = -\log \sum_h exp\,(-E(x,h)) \qquad (2.a)$$

Можно использовать более общую формулировку: $\log \tilde{p}_{model}(x)$.

Машины Больцмана

Машины Больцмана первоначально были предложены как общий «коннекционистский» подход к обучению произвольных распределений вероятности бинарных векторов.

Коннекционизм моделирует мыслительные или поведенческие явления процессами становления в сетях из связанных между собой простых элементов. Существует много различных форм коннекционизма, но наиболее общие используют нейросетевые модели.

Варианты машин Больцмана со случайными величинами других видов по популярности давно превзошли оригинал. Кратко вспомним и рассмотрим бинарную машину Больцмана и обсудим, какие проблемы возникают при попытке обучить модель и выполнить с ее помощью вывод.

Мы определим машину Больцмана над d-мерным бинарным случайным вектором $x \in \{0,1\}^d$. Машина Больцмана – это энергетическая модель, т. е. совместное распределение вероятности определяется с помощью функции энергии:

$$P(x) = \frac{\exp(-E(x))}{Z} \qquad (1)$$

где $E(x)$ – функция энергии, а Z – статистическая сумма, гарантирующая, что $\Sigma_x P(x) = 1$. Функция энергии машины Больцмана имеет вид

$$E(x) = -x^\top U x - b^\top x \qquad (2)$$

где U – матрица «весов», содержащая параметры модели, а b – вектор смещений.

Для машины Больцмана общего вида мы имеем набор n-мерных обучающих примеров, а формула (1) описывает совместное распределение вероятности наблюдаемых переменных. Ситуация вполне рабочая, но виды взаимодействий между наблюдаемыми переменными ограничены матрицей весов. Точнее, вероятность, что некоторый блок включен, определяется линейной моделью (логистической регрессией) по значениям других блоков.

Мощность машины Больцмана возрастает, если не все переменные наблюдаемые. В таком случае латентные переменные могут действовать подобно скрытым блокам в многослойном перцептроне и моделировать взаимодействия высшего порядка между видимыми блоками. Напомним, что добавление скрытых блоков в модель логистической регрессии приводит к многослойному перцептрону (МСП), который является универсальным аппроксиматором

функций. Точно так же машина Больцмана со скрытыми блоками может использоваться уже не только для моделирования линейных связей между переменными, а становится универсальным аппроксиматором функций вероятности для дискретных случайных величин.

Формально говоря, мы разбиваем множество блоков x на два подмножества: видимые v и скрытые h. Функция энергии принимает вид

$$E(v,h) = -v^\top Rv - v^\top Wh - h^\top Sh - b^\top v - c^\top h \quad (3)$$

Обучение машины Больцмана. Алгоритмы обучения машин Больцмана обычно основаны на критерии максимального правдоподобия. Если правила обучения основаны на максимальном правдоподобии, то у машин Больцмана появляется интересное свойство: обновление веса связи между двумя блоками зависит только от статистик этих двух блоков, собираемых относительно двух разных распределений: $P_{model}(v)$ и $P_{data}(v)P_{model}(h|v)$. Вся остальная сеть участвует в формировании этих статистик, но для обновления веса не нужно ничего знать ни об остальной сети, ни о том, как собиралась статистика. Следовательно, правило обучения «локально», что придает обучению машины Больцмана некоторое биологическое правдоподобие. Можно предположить, что если бы каждый нейрон был случайной величиной в машине Больцмана, то аксоны и дендриты, соединяющие две случайные величины, могли бы обучаться только путем наблюдения закономерностей возбуждения клеток, с которыми у них имеется физический контакт. В частности, в положительной фазе усиливается связь между двумя блоками, которые часто активируются вместе. Это пример правила обучения Хебба которое иногда выражают в виде мфразы: между нейронами, которые возбуждаются вместе, устанавливается связь. Правила обучения Хебба принадлежат к числу самых старых гипотетических объяснений обучения в биологических системах и сохраняют актуальность по сей день.

Другие алгоритмы обучения, в которых используется больше информации, чем просто локальная статистика, похоже, требуют гипотез о наличии дополнительных механизмов. Например, чтобы мозг мог реализовать обратное распространение, как в многослойном перцептроне, кажется необходимым поддержание вторичной коммуникационной сети для передачи информации о градиенте назад по сети. Предложения биологически правдоподобных реализаций (и аппроксимаций) обратного распространения выдвигались, но пока не проверены.

Ограниченные машины Больцмана

Ограниченная машина Больцмана, названная гармониумом, сейчас является самым распространенным строительным блоком глубоких вероятностных

моделей. Ограниченная машина Больцмана (ОМБ) представляет собой неориентированную вероятностную графическую модель, содержащую слой наблюдаемых переменных и единственный слой латентных переменных. ОМБ можно собирать в стек (одна поверх другой) для формирования более глубоких моделей.

На рис. 1 приведено несколько примеров. Так, на рис. 1а изображена графовая структура самой ОМБ. Это двудольный граф, в котором запрещены связи внутри слоя наблюдаемых переменных и внутри слоя латентных переменных.

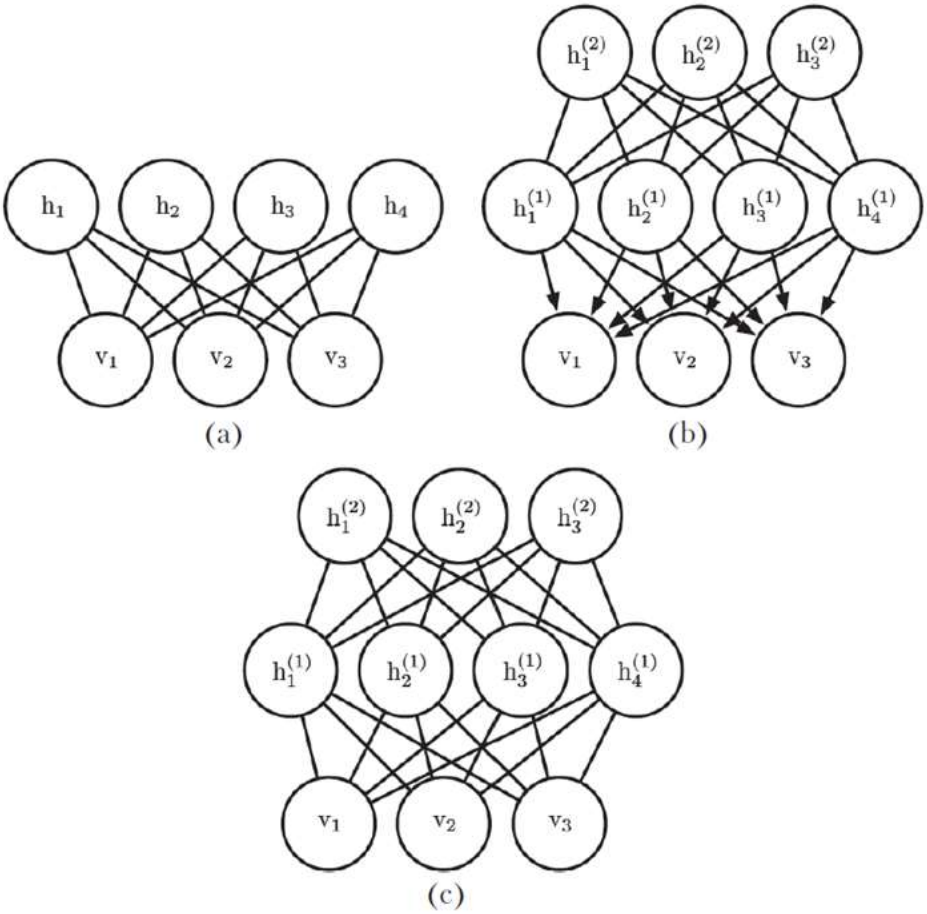

Рис. 1 Примеры моделей, построенных из ограниченных машин Больцмана. (a) Сама ограниченная машина Больцмана – это неориентированная графическая модель, основанная на двудольном графе, в одной доле которого находятся видимые блоки, а в другой – скрытые блоки. Между видимыми блоками нет никаких связей – так же, как между скрытыми. Обычно каждый видимый блок связан с каждым скрытым, но встречаются и ОМБ с разреженными связями, например сверточные ОМБ. (b) Глубокая сеть доверия (ГСД, англ. DBN)) – гибридная графическая модель, включающая как ориентированные, так и неориентированные связи. Как и в ОМБ, в ней нет внутрислойных связей. Однако

в ГСД несколько скрытых слоев, поэтому возможны связи между скрытыми блоками на разных уровнях. Все локальные условные распределения вероятности, необходимые глубокой сети доверия, копируются непосредственно из локальных условных распределений вероятности, составляющих сеть ОМБ. Можно было бы вместо этого представить глубокую сеть доверия полностью неориентированным графом, но тогда потребовались бы внутрислойные связи для улавливания зависимостей между родителями. (с) Глубокая машина Больцмана (ГМБ, англ. DBM) – это неориентированная графическая модель с несколькими слоями латентных переменных. У ГМБ, как и у ОМБ и ГСД, нет внутрислойных связей. ГМБ не так тесно связаны с ОМБ, как ГСД. Если ГМБ инициализируется стеком ОМБ, то параметры ОМБ необходимо немного модифицировать. Некоторые виды ГМБ можно обучать без предварительного обучения набора ОМБ

Начнем с бинарной версии ограниченной машины Больцмана, но существуют обобщения на другие типы видимых и скрытых блоков. Формально говоря, предположим, что наблюдаемый слой состоит из множества n_v бинарных случайных величин, которое будем обозначать вектором v. А скрытый слой, состоящий из n_h бинарных случайных величин, обозначим h.

Как и общая машина Больцмана, ограниченная машина Больцмана – это энергетическая модель, в которой совместное распределение вероятности описывается функцией энергии:

$$P(\mathrm{v} = v, \mathrm{h} = h) = (1/Z)exp(-E(v,h)) \quad (4)$$

Функция энергии ОМБ имеет вид

$$E(v,h) = -b^\top v - c^\top h - v^\top Wh \quad (5)$$

а Z – нормировочная постоянная, называемая статистической суммой:

$$Z = \sum_v \sum_h exp\{-E(v,h)\}$$

Из определения статистической суммы Z ясно, что наивный метод ее вычисления (суммирование по всем состояниям) может оказаться вычислительно неразрешимым. Формально доказано, что статистическая сумма Z неразрешима. это означает, что неразрешимым является также совместное распределение $P(v)$.

Условные распределения

Хотя $P(v)$ неразрешима, у двудольного графа, описывающего структуру ОМБ, естьспециальное свойство: условные распределения $P(h|v)$ и $P(v|h)$ факторные, допускающие сравнительно простое вычисление и выборку.

Вывести условные распределения из совместного просто:

$$P(h|v) = \frac{P(h,v)}{P(v)} \quad (7)$$

$$= \frac{1}{P(v)} \frac{1}{Z} exp\{b^\top v + c^\top h + v^\top W h\} \quad (8)$$

$$= \frac{1}{Z'} exp\{c^\top h + v^\top W h\} \quad (9)$$

$$= \frac{1}{Z'} exp\left\{\sum_{j=1}^{n_h} c_j h_j + \sum_{j=1}^{n_h} v^\top W_{:,j} h_j\right\} \quad (10)$$

$$= \frac{1}{Z'} \prod_{j=1}^{n_h} exp\{c_j h_j + v^\top W_{:,j} h_j\} \quad (11)$$

$W_{:,j} - j$-й столбец матрицы W

Поскольку в условии находятся видимые блоки v, можно рассматривать их как постоянные относительно распределения $P(h|v)$. Факторная природа условного распределения $P(h|v)$ сразу же следует из возможности записать совместное распределение вектора h в виде произведения (ненормированных) распределений отдельных элементов h_j. Осталось только нормировать распределения индивидуальных бинарных h_j.

$$P(h_j = 1|v) = \frac{\tilde{P}(h_j = 1|v)}{\tilde{P}(h_j = 0|v) + \tilde{P}(h_j = 1|v)} \quad (12)$$

$$= \frac{exp\{c_j + v^\top W_{:,j}\}}{exp\{0\} + exp\{c_j + v^\top W_{:,j}\}} \quad (13)$$

$$= \sigma(c_j + v^\top W_{:,j}) \quad (14)$$

Теперь можно выразить полное условное распределение скрытого слоя в виде факторного распределения:

$$P(h|v) = \prod_{j=1}^{n_h} \sigma\left((2h - 1) \odot (c + W^\top v)\right) \quad (15)$$

Точно так же можно показать, что и другое условное распределение, $P(v|h)$, является факторным:

$$P(v|h) = \prod_{j=1}^{n_v} \sigma((2v - 1) \odot (b + Wh)) \quad (16)$$

$\odot$ – произведение Адамара известное как поэлементное произведение/

Обучение ограниченных машин Больцмана

Поскольку ОМБ (ограниченные машины Больцмана) допускает эффективное вычисление и дифференцирование $\tilde{P}(v)$, а также эффективную МСМС (Монте-Карло по схеме марковских цепей)-выборку в виде блочной выборки по Гиббсу (алгоритм для генерации выборки совместного распределения множества случайных величин), то ее легко обучить методов обучения моделей с неразрешимыми статистическими суммами: CD (алгоритм сопоставительного расхождения (contrastive divergence)), PCD (устойчивое сопоставительное расхождение (persistent contrastive divergence), сопоставление отношений и т. д.

Концептуально простой и эффективной способ построения марковской цепи, которая производит выборку из $p_{model}(x)$, дает выборка по Гиббсу, когда

выборка из повторного обновления $T(x'|x)$ производится путем выбора одной величины x_i и выборки ее значений из p_{model} при условии соседей в неориентированном графе $\mathcal{G}$, определяющем структуру энергетической модели. Можно также одновременно производить выборку нескольких величин, если только они условно независимы при условии всех своих соседей. Из всех скрытых блоков можно производить выборку одновременно, потому что они условно независимы друг от друга при условии всех видимых блоков. И точно так же можно одновременно производить выборку из всех видимых блоков, потому что они условно независимы друг от друга при условии всех скрытых блоков. Если одновременно обновляется несколько величин, то говорят о блочной выборке по Гиббсу.

Алгоритм сопоставительного расхождения (contrastive divergence) (CD, или CDk, чтобы показать, что это алгоритм CD с k шагами выборки по Гиббсу) инициализирует марковскую цепь на каждом шаге примерами, выбранными из распределения данных. Эта идея представлена в алгоритме 1.

Алгоритм 1. Алгоритм сопоставительного расхождения, в котором в качестве процедуры оптимизации используется градиентное восхождение

Установить размер шага ε равным малому положительному числу.

Установить число шагов выборки по Гиббсу k достаточно большим для того, чтобы выборка по схеме марковской цепи из $p(x;\theta)$ перемешивалась при инициализации из p_{data}. Для обучения ОМБ на небольшом фрагменте изображения можно взять значение от 1 до 20.

while не сошелся **do**

Выбрать мини-пакет m примеров $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ из обучающего набора

$g \leftarrow \frac{1}{m}\sum_{i=1}^{m} \nabla_\theta \log \tilde{p}\,(x^{(i)};\theta)$.

for $i = 1$ to m **do**

$\tilde{\boldsymbol{x}}(i) \leftarrow \boldsymbol{x}(i)$

end for

for $i = 1\ to\ k$ **do**

  for $j = 1\ to\ m$ **do**

    $\tilde{\boldsymbol{x}}^{(j)} \leftarrow gibbs\_update(\tilde{\boldsymbol{x}}^{(j)})$

  end for

end for

  $g \leftarrow g - \frac{1}{m}\sum_{i=1}^{m} \nabla_\theta \log \tilde{p}\,(\tilde{x}^{(i)};\theta)$

  $\theta \leftarrow \theta + \varepsilon g$

end while

Получение примеров из распределения данных ничего не стоит, потому что они уже присутствуют в наборе данных. Первоначально распределение данных сильно отличается от модельного, поэтому отрицательная фаза не очень точна. Но, к счастью, положительная фаза все-таки может верно увеличивать вероятность данных в модели. Если дать положительной фазе поработать некоторое время, то модельное распределение окажется ближе к распределению данных, и верность негативной фазы начнет расти. Пример положительной и отрицательной фазы представлены на рис.2.

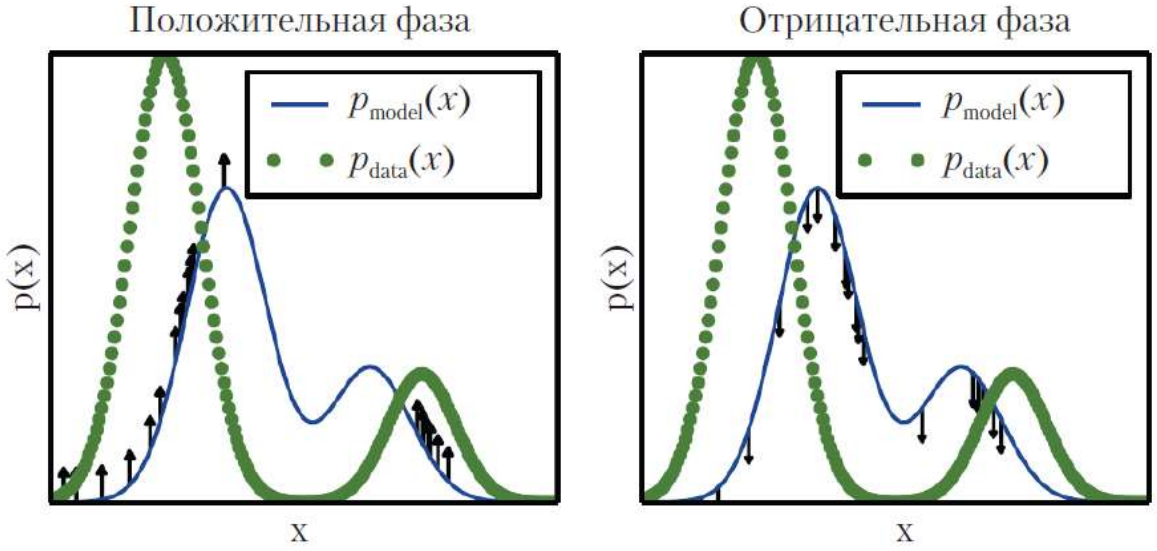

Рис. 2. Положительная и отрицательная фаза.

(Слева) В положительной фазе мы выбираем точки из распределения данных и толкаем вверх их ненормированную вероятность. Это означает, что точки, которые с вероятностью принадлежат данным, проталкиваются выше вверх. (Справа) В отрицательной фазе мы выбираем точки из модельного распределения и толкаем вниз их ненормированную вероятность. Это противодействует стремлению положительной фазы просто всюду прибавить большую постоянную к ненормированной вероятности. Если распределение данных и модельное распределение совпадают, то у положительной фазы такие же шансы поднять точку вверх, как у отрицательной – опустить вниз. Если такое происходит, то градиент математического ожидания обнуляется, и обучение следует остановить.

Разумеется, алгоритм CD по-прежнему является лишь приближением к правильной отрицательной фазе. Основная причина, по которой CD качественно не справляется с реализацией отрицательной фазы, заключается в невозможности подавить области высокой вероятности, далекие от реальных обучающих примеров. Такие области, в которых вероятность в модели высокая, а в истинном порождающем данные распределении низкая, называются паразитными модами.

На рис. 3 показано, почему это происходит. Дело в том, что моды модельного распределения, далекие от распределения данных, посещаются марковскими цепями, инициализированными в обучающих точках, только если k очень велико.

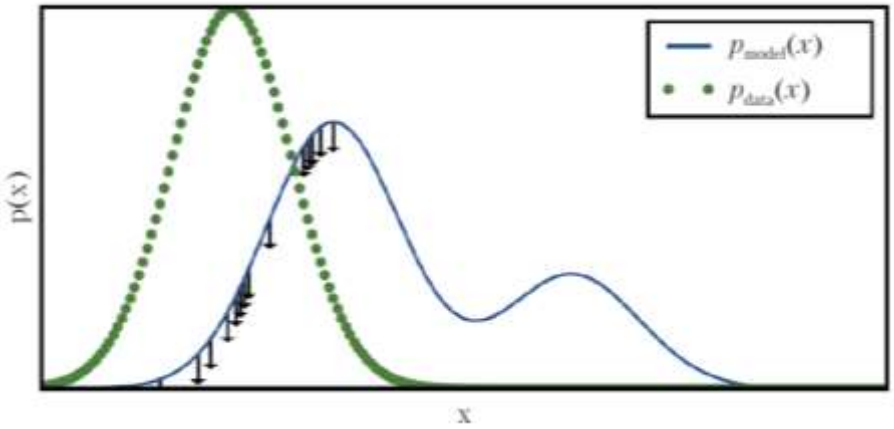

Рис. 1 Паразитная мода.

Иллюстрация того, как отрицательная фаза сопоставительного расхождения (алгоритм 1) не справляется с подавлением паразитных мод. Паразитной называется мода, присутствующая в модельном распределении, но отсутствующая в истинном распределении данных. Поскольку в алгоритме сопоставительного расхождения марковские цепи инициализируются по точкам из распределения данных и работают всего несколько шагов, то маловероятно, что они посетят моды модели, далеко отстоящие от данных. Это означает, что при выборке из модели мы иногда будем получать примеры, не похожие на данные. Кроме того, из-за расходования части массы вероятности на эти моды модель будет испытывать трудности с размещением областей высокой вероятности в правильных модах. Для наглядности на этом рисунке используется несколько упрощенное понятие расстояния – паразитная мода далеко отстоит от правильной моды вдоль горизонтальной оси в $\mathbb{R}$ (множестве вещественных чисел). Это соответствует марковской цепи, которая производит локальные перемещения с единственной случайной величиной x из $\mathbb{R}$. В большинстве глубоких вероятностных моделей марковские цепи основаны на выборке по Гиббсу и могут нелокально перемещать любую величину, но не все сразу. Для таких задач обычно лучше рассматривать не евклидово, а редакторское расстояние между модами. Однако редакторское расстояние (это минимальное количество букв, которые нужно вставить, удалить или заменить, чтобы получить из одного слова другое) в многомерном пространстве трудно изобразить на двумерном рисунке

Алгоритм CD полезен для обучения мелких моделей типа ОМБ. Собрав несколько таких моделей, можно инициализировать более глубокие модели, например глубокие сети доверия или глубокие машины Больцмана. Но CD мало

чем может помочь в непосредственном обучении более глубоких моделей. Все дело в трудности получения примеров скрытых блоков при наличии примеров видимых блоков. Поскольку скрытые блоки не включаются в данные, инициализация по обучающим примерам не решает проблему. Даже если видимые блоки инициализированы на основе данных, мы все равно должны приработать марковскую цепь, чтобы получить выборку из распределения скрытых блоков при условии видимых примеров.

Можно считать, что алгоритм CD штрафует модель за наличие марковской цепи, которая быстро изменяет вход, если тот поступает из данных. Это означает, что обучение с помощью CD чем-то напоминает обучение автокодировщика. Несмотря на то что смещение CD больше, чем у некоторых других методов обучения, этот алгоритм может быть полезен для предобучения мелких моделей, которые впоследствии собираются в стек. Объясняется это тем, что предшествующие модели в стеке копируют больше информации в свои латентные переменные, делая ее доступной последующим моделям. Это следует рассматривать скорее как часто эксплуатируемый побочный эффект обучения с помощью CD, нежели как принципиальную особенность, заложенную в проект.

Другая стратегия, решающая многие проблемы, присущие CD, – инициализировать марковские цепи на каждом шаге градиентного спуска состояниями с предыдущего шага. Впервые этот подход получил распространение под названием стохастической максимизации правдоподобия (СМП) в прикладной математике и статистике, а впоследствии был независимо открыт в сообществе глубокого обучения под названием устойчивое сопоставительное расхождение (persistent contrastive divergence – PCD или PCD-k, чтобы показать, что используется k шагов выборки по Гиббсу на каждое обновление). См. алгоритм 2.

Алгоритм 2. Алгоритм стохастической максимизации правдоподобия (устойчивого сопоставительного расхождения), в котором в качестве процедуры оптимизации используется градиентное восхождение

Установить размер шага ε равным малому положительному числу.

Установить число шагов выборки по Гиббсу k достаточно большим для того, чтобы выборка по схеме марковской цепи из $p(x; \theta + \varepsilon g)$ приработалась, начав с примеров из $p(x; \theta)$. Для обучения ОМБ на небольшом фрагменте изображения можно взять значение 1, для более сложной модели, например глубокой сети доверия, – от 5 до 50.

Инициализировать набор m примеров $\{\tilde{\boldsymbol{x}}^{(1)}, \ldots, \tilde{\boldsymbol{x}}^{(m)}\}$ случайными значениями (выбранными, например, из равномерного или нормального распределения).

while не сошелся **do**

Выбрать мини-пакет m примеров $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ из обучающего набора

$g \leftarrow \frac{1}{m}\sum_{i=1}^{m} \nabla_\theta \log \tilde{p}\,(x^{(i)}; \theta)$.

for $i = 1\ to\ k$ **do**

for $j = 1\ to\ m$ **do**

$\tilde{\boldsymbol{x}}^{(j)} \leftarrow gibbs\_update(\tilde{\boldsymbol{x}}^{(j)})$

end for

$g \leftarrow g - \frac{1}{m}\sum_{i=1}^{m} \nabla_\theta \log \tilde{p}\,(\tilde{x}^{(i)}; \theta)$

$\theta \leftarrow \theta + \varepsilon g$

end while

Основная идея состоит в том, что коль скоро шаги алгоритма стохастического градиентного спуска малы, модели, построенные на двух соседних шагах, будут похожи. Отсюда следует, что примеры, выбранные из распределения предыдущей модели, будут очень близки к настоящим примерам из распределения текущей модели, так что марковская цепь, инициализированная этими примерами, не потребует много времени для приработки.

Поскольку все марковские цепи непрерывно обновляются на протяжении всего процесса обучения, а не перезапускаются на каждом шаге вычисления градиента, то они могут забрести достаточно далеко, чтобы обнаружить все моды модели. Поэтому СМП оказывается гораздо устойчивее к формированию моделей с паразитными модами, чем CD. Кроме того, благодаря возможности запоминать состояние всех переменных, из которых производится выборка, как видимых, так и латентных, СМП поставляет начальные данные для скрытых и видимых блоков. CD может обеспечить инициализацию только видимых блоков, поэтому в глубоких моделях требуется приработка. СМП способен обучать глубокие модели более эффективно.

СМП может утратить верность, если стохастический градиентный алгоритм перемещает модель настолько быстро, что марковская цепь не успевает прирабатываться между шагами. Это может случиться, если k слишком мало или ε слишком велико. К сожалению, допустимый диапазон значений сильно зависит от задачи. Неизвестно, как можно формально проверить, успешно ли прирабатывается цепь между шагами. Субъективно, если скорость обучения слишком высока для выбранного числа шагов выборки по Гиббсу, то оператор-

человек будет наблюдать, что дисперсия примеров в отрицательной фазе гораздо больше между шагами вычисления градиента, чем между различными марковскими цепями.

По сравнению с другими неориентированными моделями, используемыми в глубоком обучении, ОМБ обучается относительно просто, поскольку мы можем точно вычислить $P(h|v)$ в замкнутой форме. В других глубоких моделях, в частности в глубокой машине Больцмана, проблема неразрешимой статистической суммы сочетается с проблемой неразрешимого вывода.

Глубокие сети доверия

Глубокие сети доверия (ГСД) были одними из первых несверточных моделей, которые удалось успешно обучить в контексте глубоких архитектур. До недавнего времени считалось, что глубокие модели с трудом поддаются оптимизации. Усилия исследователей были в основном направлены на изучение ядерных методов с выпуклыми целевыми функциями. Глубокие сети доверия продемонстрировали, что глубокая архитектура может по качеству превзойти метод опорных векторов с ядром на наборе MNIST (объёмная база данных образцов рукописного написания цифр). Сегодня глубокие сети доверия вышли из моды и применяются редко даже по сравнению с другими порождающими или обучаемыми без учителя алгоритмами, но их роль в истории глубокого обучения достойна всяческого уважения.

Глубокие сети доверия – это порождающие модели с несколькими слоями латентных переменных. Латентные переменные обычно бинарные, хотя видимые блоки могут быть как бинарными, так и вещественными. Внутри слоев нет никаких связей. Обычно каждый блок любого слоя связан с каждым блоком соседних слоев, хотя можно строить и ГСД с более разреженными связями. Связи между двумя верхними слоями неориентированные. Связи между всеми остальными слоями ориентированные, причем стрелки направлены в сторону слоя, ближайшего к данным. Пример показан на рис. 1b

ГСД с l скрытыми слоями содержит l матриц весов $W^{(1)}, \dots, W^{(l)}$, а также $l + 1$ векторов смещений $b^{(0)}, \dots, b^{(l)}$, где $b^{(0)}$ – смещения для видимого слоя. Распределение вероятности, представляемое ГСД, имеет вид:

$$P(h^{(l)}, h^{(l-1)}) \propto exp(b^{(l)\top} h^{(l)} + b^{(l-1)\top} h^{l-1} + h^{(l-1)\top} W^{(l)} h^{(l)}) \quad (17)$$

$$P(h_i^{(k)} = 1 | h^{(k+1)}) = \sigma(b_i^{(k)} + W_{:;i}^{(k+1)\top} h^{(k+1)}) \forall i, \forall k \in 1, \dots, l-2 \quad (18)$$

$$P(v_i = 1 \mid h^{(1)}) = \sigma(b_i^{(0)} + W_{:;i}^{(1)\top} h^{(1)}) \forall i \quad (19)$$

В случае вещественных видимых блоков подставляем

$$v \sim \mathcal{N}(v;\ b^{(0)} + W^{(1)\top} h^{(1)}, \beta^{-1}) \quad (20)$$

где β – диагональная матрица, иначе вычисления становятся слишком сложными.

Обобщение на другие экспоненциальные семейства видимых блоков не вызывают трудностей, по крайней мере в теории. ГСД с единственным скрытым слоем – это просто ОМБ.

Чтобы произвести выборку из ГСД, мы сначала выполняем несколько шагов выборки по Гиббсу для двух верхних скрытых слоев. На этом этапе, по существу, производится выборка из ОМБ, определенной этими двумя слоями. Затем можно применить один проход предковой выборки к остальной части модели, чтобы произвести выборку из видимых блоков.

Глубокие сети доверия подвержены многим проблемам, присущим как ориентированным, так и неориентированным моделям.

Вывод в глубокой сети доверия вычислительно неразрешим из-за эффекта оправдания внутри каждого ориентированного слоя и взаимодействия между двумя скрытыми слоями с неориентированными связями. Вычисление или максимизация стандартной нижней границы свидетельств для логарифмического правдоподобия также неразрешимы, поскольку в этой нижней границе участвует математическое ожидание клик, размер которых равен ширине сети. При вычислении или максимизации логарифмического правдоподобия приходится сталкиваться не только с проблемой неразрешимого вывода для исключения латентных переменных, но и с проблемой неразрешимой статистической суммы в неориентированной модели двух верхних слоев.

Обучение глубокой сети доверия начинается с того, что мы обучаем ОМБ максимизировать математическое ожидание $\log p(v)$ при заданном распределении p_{data} $\mathbb{E}_{v\sim p_{data}} \log p(v)$ ($v \sim p_{data}$ случайная величина v имеет распределение p_{data}) с помощью алгоритма сопоставительного расхождения или стохастической максимизации правдоподобия. Полученные параметры ОМБ определяют параметры первого слоя ГСД. Далее мы обучаем вторую ОМБ приближенно максимизировать выражение

$$\mathbb{E}_{v\sim p_{data}} \mathbb{E}_{h^{(1)}\sim p^{(1)}(h^{(1)}|v)}\ log\ p^{(2)}(h^{(1)}), \tag{21}$$

где $p^{(1)}$ – распределение вероятности, представленное первой ОМБ, а $p^{(2)}$ – распределение, представленное второй ОМБ. Иными словами, вторая ОМБ обучается моделировать распределение, определенное выборкой из скрытых блоков первой ОМБ, тогда как первая ОМБ управляется данными. Эту процедуру можно повторять бесконечно, добавляя в ГСД столько слоев, сколько необходимо, при этом каждая новая ОМБ будет моделировать выборку из

предыдущей. Каждая ОМБ определяет очередной слой ГСД. Эту процедуру можно обосновать как увеличение вариационной нижней границы логарифмического правдоподобия данных в модели ГСД.