ЛЕКЦИЯ 4. Виды машины Больцмана

Машины Больцмана для вещественных данных

Первоначально машины Больцмана разрабатывались для бинарных данных, но во многих приложениях, в т. ч. при моделировании изображений и звука, необходимо представлять распределения вероятности вещественных значений. В некоторых случаях вещественные данные на отрезке [0,1] можно рассматривать как представление математического ожидания бинарной случайной величины. Например, полутоновые изображения в обучающем наборе рассматриваются как определение вероятностей из диапазона [0,1]. Каждый пиксель определяет вероятность того, что бинарная величина принимает значение 1, и все бинарные пиксели выбираются независимо друг от друга. Это распространенная процедура вычисления бинарных моделей для наборов полутоновых изображений. Тем не менее с теоретической точки зрения этот подход не слишком хорош, а независимо выбранные таким способом бинарные изображения напоминают шум. Рассмотрим машины Больцмана, определяющие плотность вероятности вещественных данных.

ОМБ Гаусса–Бернулли

Ограниченные машины Больцмана можно разработать для многих экспоненциальных семейств условных распределений. Наиболее распространены ОМБ с бинарными скрытыми и вещественными видимыми блоками, и нормальным условным распределением видимых блоков, среднее значение которого является функцией скрытых блоков. Существует много способов параметризации ОМБ Гаусса–Бернулли. В частности, можно выбрать, использовать для нормального распределения ковариационную матрицу или матрицу точности. Ниже будет описана формулировка с матрицей точности. Переформулирование с ковариационной матрицей не составляет труда. Мы хотим иметь условное распределение

$$p(v \mid h) = \mathcal{N}(v;\, Wh, \beta^{-1}) \qquad (1)$$

Мы можем найти, какие члены следует прибавить к функции энергии, раскрыв ненормированное логарифмическое условное распределение:

$$\log \mathcal{N}(v;\, Wh, \beta^{-1}) = 1/2(\mathrm{v} - \mathrm{Wh})^{\top}\beta(\mathrm{v} - \mathrm{Wh}) + \mathrm{f}(\beta) \qquad (2)$$

Здесь f инкапсулирует все члены, являющиеся функцией только параметров, а не случайных величин модели. Мы можем отбросить f, поскольку ее единственная роль – нормировать распределение, но эту роль сыграет статистическая сумма выбранной нами функции энергии.

Если мы включим все содержащие v члены (с противоположным знаком) уравнения (2) в нашу функцию энергии и не будем прибавлять никаких других

членов, содержащих v, то функция энергии будет представлять желаемое условное распределение $p(v|h)$.

В выборе другого условного распределения $p(h|v)$ нам предоставлена некоторая свобода. Заметим, что в уравнении (2) имеется член

$$1/2h^\top W^\top \beta W h \tag{3}$$

Этот член нельзя включить целиком, поскольку он содержит члены вида $h_i h_j$, соответствующие ребрам между скрытыми блоками. Если бы мы включили эти члены, то получили бы линейную факторную модель, а не ограниченную машину Больцмана. При проектировании машины Больцмана мы просто опускаем эти попарные произведения. При этом условное распределение $p(v|h)$ не изменяется, так что уравнение (2) по-прежнему справедливо. Однако мы можем решить, следует ли включать члены, содержащие единственный элемент h_i. Если взять диагональную матрицу точности, то окажется, что для каждого скрытого блока h_i имеется член

$$\frac{1}{2} h_i \sum_j \beta_j W_{j,i}^2 \tag{4}$$

Здесь мы воспользовались тем фактом, что $h_i^2 = h_i$, потому что $h_i \in \{0,1\}$. Если включить этот член (с противоположным знаком) в функцию энергии, то у h_i появится естественная тенденция (выражаемая смещением) к выключению, когда велики веса связей этого блока с видимыми блоками высокой точности. Решение о том, включать этот член смещения или нет, не влияет на семейство распределений, представимых моделью (в предположении, что включены параметры смещения для скрытых блоков), но влияет на динамику обучения модели. Благодаря его включению активации скрытых блоков, возможно, останутся разумными даже тогда, когда абсолютные величины весов быстро возрастают.

Вот одно из возможных определений функции энергии для ОМБ Гаусса–Бернулли ($\beta \odot v$ поэлементное произведение матриц β и v (произведение Адамара)):

$$E(v,h) = 1/2v^\top(\beta \odot v) - (v \odot \beta)^\top W h - b^\top h, \tag{5}$$

но можно также добавить дополнительные члены или параметризовать энергию в терминах дисперсии, а не точности.

В этом выводе не включен член смещения для видимых блоков, но его легко добавить. И последний способ модифицировать параметризацию ОМБ Гаусса–Бернулли – решить, как трактовать матрицу точности. Она может быть фиксированной (скажем, взять оценку на основе маргинальной точности данных) или обучаемой.

Она может быть равна произведению тождественной матрицы на скаляр или произвольной диагональной матрицей. Обычно мы не используем в этом контексте недиагональных матриц точности, потому что некоторые операции над нормальным распределением требуют обращения матрицы, а диагональная матрица обращается тривиально. Другие виды машин Больцмана допускают моделирование ковариационной структуры с применением различных приемов, позволяющих избежать обращения матрицы точности.

Неориентированные модели условной ковариации

Ковариация – мера взаимосвязи двух случайных величин, измеряющая общее отклонение двух случайных величин от их ожидаемых значений. Метрика оценивает, в какой степени переменные изменяются вместе.

Хотя гауссова ОМБ всегда была канонической энергетической моделью для вещественных данных, но индуктивное смещение гауссовой ОМБ плохо соответствует статистическим вариациям, присутствующим в некоторых типах вещественных данных, особенно в естественных изображениях. Проблема в том, что значительная часть информационного содержимого естественных изображений заключена в ковариации между пикселями, а не в самих значениях пикселей. Другими словами, именно связи между пикселями, а не их абсолютные значения определяют полезную информацию, присутствующую в изображении. Поскольку гауссова ОМБ моделирует только условное среднее входных данных при условии скрытых блоков, она не способна уловить информацию об условной ковариации. В ответ на эту критику были предложены альтернативные модели, пытающиеся лучше учесть ковариацию вещественных данных. К их числу относится ОМБ со средним и ковариацией (mean and covariance RBM – mcRBM), модель среднего произведения t-распределения Стьюдента (mPoT) и ОМБ типа Spike and Slab RBM (ssRBM).

ОМБ со средним и ковариацией.

В модели mcRBM скрытые блоки используются для независимого кодирования условного среднего и ковариации всех наблюдаемых блоков. Скрытый слой mcRBM разбит на две группы блоков: блоки среднего и блоки ковариации. Группа, моделирующая условное среднее, – это просто гауссова ОМБ. Вторая половина – ковариационная ОМБ, которую часто называют cRBM; ее компоненты моделируют структуру условной ковариации, как описано ниже.

Точнее говоря, если бинарные блоки среднего обозначить $h^{(m)}$, а бинарные блоки ковариации $h^{(c)}$, то модель mcRBM определяется как комбинация двух функций энергии:

$$E_{mc}(x,\ h^{(m)}, h^{(c)})\ =\ E_m(x,\ h^{(m)})\ +\ E_c(x, h^{(c)}\ ), \qquad (6)$$

где E_m – стандартная функция энергии ОМБ Гаусса–Бернулли (Этот вариант функции энергии ОМБ Гаусса–Бернулли предполагает, что в данных изображения среднее всех пикселей равно нулю. В модель можно легко добавить пиксельные смещения, чтобы учесть ненулевые средние)

$$E_m\left(x,\ h^{(m)}\right) = \frac{1}{2}x^\top x - \sum_j x^\top W_{:,j} h_j^{(m)} - \sum_j b_j^{(m)}\, h_j^{(m)} \qquad (7)$$

а E_c – функция энергии cRBM, моделирующая информацию об условной ковариации:

$$E_c\left(x,\ h^{(c)}\right) = \frac{1}{2}\sum_j h_j^{(c)} \left(x^\top r^{(j)}\right)^2 - \sum_j b_j^{(c)}\, h_j^{(c)} \qquad (8)$$

Параметр $r^{(j)}$ соответствует вектору весов ковариации, ассоциированному с $h_j^{(c)}$, а $b^{(c)}$ – вектор смещений ковариации. Объединенная функция энергии определяет совместное распределение

$$p_{mc}(x,\ h^{(m)}\ ,\ h^{(c)}) \ =\ (1/Z)exp\{-E_{mc}(x,\ h^{(m)}, h^{(c)})\} \qquad (9)$$

и соответствующее условное распределение наблюдений при условии $h^{(m)}$ и $h^{(c)}$ в виде многомерного нормального распределения:

$$p_{mc}(x|\ h^{(m)}\ ,\ h^{(c)}) \ = \mathcal{N}\left(x; C_{x|h}^{mc}\left(\sum_j W_{:,j} h_j^{(m)}\right), C_{x|h}^{mc}\right) \qquad (9)$$

Отметим, что ковариационная матрица $C_{x|h}^{mc} = \left(\sum_j h_j^{(c)}, r^{(j)}, r^{(j)\top} + I\right)^{-1}$ не является диагональной и что W – матрица весов, ассоциированная с моделированием условных средних с помощью гауссовой ОМБ. Обучить mcRBM методами сопоставительного расхождения или устойчивого сопоставительного расхождения трудно из-за недиагональной условной ковариационной матрицы. В методах CD (сопоставительного расхождения) и PCD (устойчивое сопоставительное расхождение) требуется производить выборку из совместного распределения $x,\ h^{(m)}\ ,\ h^{(c)}$, что в стандартной ОМБ достигается путем выборки по Гиббсу из условных распределений. Однако в mcRBM (ОМБ со средним и ковариацией) для выборки из $p_{mc}(x|\ h^{(m)}\ ,\ h^{(c)})$ необходимо вычислять $(C^{mc})^{-1}$ на каждой итерации обучения. Для больших объемов наблюдений это может оказаться неподъемной вычислительной задачей.

Среднее произведение t-распределения Стьюдента

Модель среднего произведения t-распределения Стьюдента (mPoT) обобщает модель PoT (произведение t-распределения Стьюдента) примерно так же, как mcRBM обобщает cRBM (ОМБ с ковариацией). Достигается это путем включения ненулевых гауссовых средних за счет добавления скрытых блоков, как в гауссовой ОМБ. Подобно mcRBM (ОМБ со средним и ковариацией), условное распределение наблюдений PoT является многомерным нормальным

распределением (с недиагональной ковариационной матрицей), но, в отличие от mcRBM, дополнительное условное распределение скрытых блоков описывается условными независимыми гамма-распределениями.

Гамма-распределение $\mathcal{G}(k, \theta)$ – это распределение вероятности положительных вещественных чисел со средним $k\theta$. Для понимания основных идей модели mPoT знакомство с деталями гамма-распределения необязательно.

Функция энергии в модели mPoT имеет вид

$$E_{mPoT}\left(x,\ h^{(m)}\ ,\ h^{(c)}\right) = E_m\left(x,\ h^{(m)}\right) + \sum_j \left(h_j^{(c)} \left(1 + \frac{1}{2}\left(r^{(j)\top} x\right)^2\right) + \left(1 - \gamma_j\right) \log h_j^{(c)}\right) \quad (10)$$

где $r^{(j)}$ – вектор весов ковариации, ассоциированный с блоком $h_j^{(c)}$, а функция $E_m\left(x,\ h^{(m)}\right)$ определена, как в (7).

Как и в случае mcRBM, функция энергии в модели mPoT определяет многомерное нормальное распределение – такое, что условное распределение x имеет недиагональную ковариационную матрицу. Обучение модели mPoT – как и mcRBM – осложняется невозможностью выборки из нормального условного распределения с недиагональной ковариационной матрицей $E_{mPoT}\left(x,\ h^{(m)}\ ,\ h^{(c)}\right)$, поэтому также предлагается прямая выборка из p(x) гамильтоновым (гибридным) методом Монте-Карло.

Ограниченные машины Больцмана типа Spike and Slab.

Spike and Slab (Шип и Плита)

Ограниченные машины Больцмана типа Spike and Slab, или ssRBM, – еще один способ моделирования ковариационной структуры вещественных данных. По сравнению с mcRBM (ОМБ со средним и ковариацией), они обладают тем преимуществом, что не нуждаются ни в обращении матрицы, ни в гамильтоновых методах Монте-Карло. Подобно mcRBM и mPoT (среднее произведение t-распределения Стьюдента), в бинарных скрытых блоках ssRBM (ограниченные машины Больцмана типа Spike and Slab) закодирована условная ковариация между пикселями посредством использования вспомогательных вещественных переменных.

В ОМБ типа Spike and Slab есть два набора скрытых блоков: бинарные spike-блоки h и вещественные slab-блоки s. Среднее значение видимых блоков при условии скрытых блоков равно $(h \odot s)W^\top$. Иначе говоря, каждый столбец $W_{:,\mathrm{i}}$ определяет компоненту, которая может встречаться во входных данных, когда $h_i = 1$. Соответствующая spike-переменная h_i определяет, присутствует ли эта компонента вообще. А соответствующая slab-переменная определяет

интенсивность этой компоненты, если она присутствует. Когда spike-переменная активна, соответствующая slab-переменная добавляет дисперсию к входным данным вдоль оси, определенной столбцом $W_{:,\mathrm{i}}$. Это позволяет моделировать ковариацию между входами. По счастью, методы CD (сопоставительного расхождения) и PCD (устойчивое сопоставительное расхождение) с выборкой по Гиббсу по-прежнему применимы. Обращать матрицы не нужно.

Формально модель ssRBM определяется функцией энергии:

$$E_{\mathrm{ss}}(x,s,h) = -\sum_i x^\top s_i h_i + \frac{1}{2}x^\top(\Lambda + \sum_i \Phi_i h_i)x + \frac{1}{2}\sum_i \alpha_i s_i^2 - \sum_i \alpha_i \mu_i s_i h_i - \sum_i b_i h_i - \sum_i \alpha_i \mu_i^2 h_i \quad (11)$$

где b_i – смещение spike-блока h_i, Λ – диагональная матрица точности для наблюдений x, $\alpha_i > 0$ – скалярный параметр точности вещественной slab-переменной s_i, а Φ_i – неотрицательная диагональная матрица, определяющая h-модулированный квадратичный штраф на x. Параметр μ_i задает среднее slab-переменной s_i.

В случае, когда совместное распределение определено функцией энергии, вывести условные распределения в модели ssRBM сравнительно просто. Например, исключая slab-переменные s, получаем, что условное распределение наблюдений при условии бинарных spike-переменных h имеет вид

$$p_{ss} = (x \mid h) = \frac{1}{P(h)}\frac{1}{Z}\int \exp\{-E(x,s,h)\}\,ds = \mathcal{N}\left(x; C^{ss}_{x\mid h}\sum_i W_{:,\mathrm{i}}\,\mu_i h_i, C^{ss}_{x\mid h}\right) \quad (12)$$

где $C^{ss}_{x\mid h} = \left(\Lambda + \sum_i \Phi_i h_i - \sum_i \alpha_i^{-1} h_i\, W_{:,\mathrm{i}}\, W_{:,i}^\top\right)$. Последнее равенство имеет место, только если ковариационная матрица $C^{ss}_{x\mid h}$ положительно определенная.

Фильтрация по spike-переменным означает, что истинное маргинальное распределение $h \odot s$ разреженное. Это не то же самое, что разреженное кодирование, где выборки из модели «почти никогда» (в смысле теории меры) не содержат нулей в коде и требуется, чтобы MAP-вывод индуцировал разреженность.

Если сравнить ssRBM (ограниченные машины Больцмана типа Spike and Slab) с mcRBM (ОМБ со средним и ковариацией) и mPoT (среднее произведение t-распределения Стьюдента), то окажется, что ssRBM параметризует условную ковариацию между наблюдениями совершенно иначе. И mcRBM, и mPoT моделируют структуру в виде $\left(\sum_j h_j^{(c)} r^{(j)} r^{(j)\top} + I\right)^{-1}$ используя активацию скрытых блоков $h_j > 0$, чтобы наложить ограничения на условную ковариацию в направлении $r^{(j)}$. Что же касается ssRBM, то она задает условную ковариацию между наблюдениями с помощью скрытых spike-активаций $h_i = 1$, чтобы стянуть матрицу активации вдоль направления, определяемого соответствующим весовым

вектором. Условная ковариация в модели ssRBM похожа на даваемую другой моделью: анализом произведения вероятностных главных компонент (product of probabilistic principal components analysis – PoPPCA). В сверхполной конфигурации разреженная активация с ssRBM-параметризацией допускает значительную дисперсию (выше номинальной, определяемой матрицей Λ^{-1}) только в избранных направлениях разреженно активированных h_i. В моделях mcRBM и mPoT сверхполное представление означало бы, что для улавливания вариативности в конкретном направлении в пространстве наблюдений потенциально пришлось бы удалить все ограничения с положительной проекцией на это направление. Отсюда следует, что эти модели хуже приспособлены к сверхполной конфигурации.

Основной недостаток ограниченной машины Больцмана типа Spike and Slab – в том, что при некоторых конфигурациях параметров получающаяся ковариационная матрица не является положительно определенной. В этом случае значения, далекие от среднего, получают большую ненормированную вероятность, так что интеграл по всем возможным исходам расходится. Обычно этой проблемы можно избежать с помощью простых эвристических приемов. Теоретически строгого решения пока не найдено. Применить ограниченную оптимизацию, чтобы явно избежать областей, где вероятность не определена, трудно, не впадая в грех чрезмерной консервативности, из-за чего может случиться так, что модель никогда не попадет в области пространства параметров, где достигается хорошее качество.

Качественно сверточные варианты ssRBM дают хорошие примеры естественных изображений. У ssRBM есть несколько обобщений. Если включить взаимодействия высшего порядка и пулинг с усреднением по slab-переменным, то модель сможет обучиться отличным признакам для классификатора в случае, когда помеченных данных мало. Добавление в функцию энергии члена, предотвращающего неопределенность статистической суммы, приводит к модели разреженного кодирования типа Spike and Slab, или S3C (spike and slab sparse coding – разреженное кодирование шипов и плит).

Сверточные машины Больцмана

Входные данные очень высокой размерности, например, изображения, предъявляют жесткие требования к моделям машинного обучения с точки зрения объема вычислений, потребной памяти и статистических свойств. Замена умножения матриц дискретной сверткой с небольшим ядром – стандартный способ решения этих проблем для входных данных с пространственной или

временной структурой, инвариантной относительно параллельных переносов. Этот подход хорошо работает в применении к ОМБ.

В глубоких сверточных сетях обычно требуется операция пулинга, так что пространственный размер каждого последующего слоя меньше размера предыдущего. В сверточных сетях прямого распространения часто в качестве функции пулинга берут, например, максимум агрегируемых элементов. Не ясно, как обобщить эту идею на энергетические модели. Можно было бы ввести бинарный блок пулинга p по n бинарным детекторным блокам d и гарантировать, что $p = max_i d_i$, полагая функцию энергии равной ∞ всюду, где это ограничение нарушается. Это решение плохо масштабируется, поскольку требует рассмотрения 2^n конфигураций энергии, чтобы вычислить нормировочную постоянную. Для небольшой области пулинга размера 3×3 придется выполнить $2^9 = 512$ вычислений функции энергии на один блок пулинга.

Для решения этой проблемы разработан метод вероятностного max-пулинга (не путайте со «стохастическим пулингом» – методом неявного построения ансамблей сверточных сетей прямого распространения). Стратегия заключается в том, чтобы наложить ограничение на детекторные блоки: не более одного активного в каждый момент времени. Это означает, что всего имеется лишь $n + 1$ состояний (по одному для случаев, когда включен один из n детекторных блоков, плюс дополнительное состояние, в котором все детекторные блоки выключены). Блок пулинга включен тогда и только тогда, когда включен один из детекторных блоков. Состоянию, в котором все блоки выключены, назначается нулевая энергия. Можно считать это описанием модели с одной переменной, имеющей $n + 1$ состояний, или, эквивалентно, модели с $n + 1$ переменными, которая назначает энергию ∞ всем совместным комбинациям переменных, кроме $n + 1$.

При всей своей эффективности вероятностный max-пулинг делает детекторные блоки взаимно исключающими, что в одних контекстах может считаться полезным регуляризирующим ограничением, а в других вредным ограничением на емкость модели. Этот метод не поддерживает пересекающихся областей пулинга, которые обычно нужны для достижения оптимального качества сверточных сетей прямого распространения, так что это ограничение, вероятно, сильно снижает качество сверточных машин Больцмана.

Декларируется, что вероятностный max-пулинг можно было бы использовать для построения сверточных машин Больцмана. Эта модель умеет выполнять такие операции, как восполнение отсутствующих частей данных. Несмотря на интеллектуальную привлекательность, работать с этой моделью на

практике трудно, и обычно в роли классификатора она показывает худшие результаты, чем традиционные сверточные сети, обученные с учителем.

Многие сверточные модели одинаково хорошо работают с входными данными разного пространственного размера. Для машин Больцмана изменить размер входа сложно по нескольким причинам. При изменении размера входа меняется статистическая сумма. Кроме того, во многих сверточных сетях инвариантность относительно размера достигается путем увеличения размера областей пулинга пропорционально размеру входа, но масштабировать области пулинга в машине Больцмана неудобно. В традиционных сверточных нейронных сетях можно использовать фиксированное число блоков пулинга и динамически увеличивать их размер. В машинах Больцмана большие области пулинга обходятся слишком дорого при наивном подходе. Сделать детекторные блоки в одной области пулинга взаимноисключающими – решает вычислительные проблемы, но все равно не позволяет иметь области пулинга переменного размера. Предположим, к примеру, что мы обучаем модель детекторных блоков, обучающихся обнаружению границ с вероятностным max-пулингом по области 2×2. Это налагает ограничение: в каждой области 2×2 может встречаться только одна граница. Если мы затем увеличим размер входного изображения на 50% в каждом направлении, то естественно ожидать, что число границ соответственно возрастет. Если же мы вместо этого увеличим на 50% размер областей пулинга в каждом направлении до 3×3, то ограничение взаимного исключения теперь говорит, что в каждой области размера 3×3 может присутствовать не более одной границы. По мере увеличения входного изображения модель генерирует границы с меньшей плотностью. Разумеется, такие проблемы возникают, только когда модель вынуждена использовать переменный размер области пулинга, чтобы выходной вектор имел фиксированный размер. Модели с вероятностным max-пулингом все же могут принимать изображения переменного размера, при условии, что карта признаков на выходе модели может масштабироваться пропорционально размеру входного изображения.

Пиксели на границе изображения тоже представляют сложность, усугубляющуюся тем фактом, что связи в машине Больцмана симметричны. Если мы не будем неявно дополнять вход нулями, то скрытых блоков будет меньше, чем видимых, и видимые блоки на границе изображения будут моделироваться плохо, потому что принадлежат рецептивному полю меньшего числа скрытых блоков. Но если производить неявное дополнение нулями, то скрытые блоки на границе будут управляться меньшим числом входных пикселей, так что активация может не произойти, когда необходимо.

Машины Больцмана для структурных и последовательных выходов

В случае структурного выхода необходимо обучить модель, умеющую отображать вход x на выход y, так что различные элементы y связаны друг с другом и должны подчиняться некоторым ограничениям. Например, в задаче синтеза речи y – звуковой сигнал, и полный выходной сигнал должен звучать как связная речь.

Естественный способ представить связи между элементами y – воспользоваться распределением вероятности $p(y|x)$. Такую вероятностную модель могут предложить машины Больцмана, обобщенные на моделирование условных распределений.

Тот же инструментарий условного моделирования с помощью машины Больцмана можно применить не только к задаче структурного вывода, но и для моделирования последовательностей. В этом случае вместо отображения входа x на выход y модель должна оценить распределение вероятности последовательности переменных $p(x^{(1)}, \dots, x^{(\tau)})$. Для решения этой задачи условные машины Больцмана могут представить факторы виды $p(x^{(t)}| x^{(1)}, \dots, x^{(t-1)})$.

Важная для киноиндустрии и видеоигр задача – смоделировать последовательности углов сочленения суставов в скелетах, используемых для отрисовки трехмерных персонажей. Эти последовательности часто запоминаются системами захвата движения при регистрации движений актеров. Вероятностная модель движения персонажа позволяет генерировать новые, не встречавшиеся ранее, но реалистичные движения. Для решения этой задачи предложено моделирование условной ОМБ $p(x^{(t)}|x^{(t-1)}, \dots, x^{(t-m)})$ для малых m. Модель представляет собой ОМБ над распределением $p(x^{(t)})$, в которой параметры смещения – линейная функция от предыдущих m значений x. При обусловливании разными значениями $x^{(t-1)}$ и более ранних переменных получаем новую ОМБ над x. Веса в ОМБ над x никогда не меняются, но за счет обусловливания по различным прошлым значениям можно изменять вероятность активности различных скрытых блоков ОМБ. Активируя и деактивируя различные подмножества скрытых блоков, мы можем вносить значительные изменения в индуцированное распределение вероятности x. Возможны и другие варианты условной ОМБ и моделирования последовательностей с помощью условных ОМБ.

Еще одна задача – моделирование распределения последовательностей музыкальных нот для создания песен. Предложена модель последовательности

РНС-ОМБ (рекуррентная нейронная сеть – ограниченная машина Больцмана) (англ. RNN-RBM), которая применена к этой задаче. Это порождающая модель последовательности кадров $x^{(t)}$, состоящая из РНС, которая порождает параметры ОМБ для каждого временного шага. В отличие от предыдущих подходов, где от шага к шагу варьировались только параметры смещения ОМБ, РНС, используемая в этой модели, порождает все параметры ОМБ, включая и веса. Для обучения модели мы должны выполнить обратное распространение градиента функции потерь по РНС. Функция потерь применяется не напрямую к выходам РНС, а к ОМБ. Это означает, что мы должны приближенно продифференцировать потерю по параметрам ОМБ, применив метод сопоставительного расхождения или другой похожий алгоритм. Затем приближенный градиент можно обратно распространить по РНС, применив обычный алгоритм обратного распространения во времени.

Другие машины Больцмана

Существует еще много вариантов машин Больцмана.

Для обобщения машин Больцмана можно применять различные критерии обучения. Мы сосредоточили внимание на машинах Больцмана, обучаемых приближенно максимизировать порождающий критерий $\log p(v)$. Можно вместо этого обучить дискриминантную ОМБ, нацеленную на максимизацию $\log p(y|v)$. Этот подход часто дает наилучшие результаты, если используется линейная комбинация порождающего и дискриминантного критериев. К сожалению, ОМБ не так хорошо обучаются с учителем, как МСП, по крайней мере с применением существующих технологий.

В большинстве практически используемых машин Больцмана функция энергии включает только взаимодействия второго порядка, т. е. представляет собой суммы большого числа членов, каждый из которых является произведением всего двух случайных величин, например $v_i W_{i,j} h_j$. Можно обучить и машины Больцмана более высокого порядка, для которых члены функции энергии являются произведениями многих величин. Трехсторонние взаимодействия между скрытым блоком и двумя разными изображениями могут моделировать пространственное преобразование между текущим и следующим кадрами видео. Умножение на унитарную переменную класса может изменить связь между видимым и текущим блоками в зависимости от того, бит какого класса поднят. Недавний пример использования взаимодействий высшего порядка дает машина Больцмана с двумя группами скрытых блоков, одна из которых взаимодействует с видимыми блоками v и меткой класса y, а другая – только с входными значениями v. Это можно интерпретировать как поощрение некоторых скрытых

блоков обучаться моделированию входа с использованием признаков, релевантных классу; кроме того, дополнительные скрытые блоки обучаются объяснять мелкие детали, необходимые для придания реалистичности примерам v, не определяя класса примера. Еще одно использование взаимодействий высшего порядка – пропускание части признаков. Машина Больцмана с взаимодействиями третьего порядка и бинарными масочными переменными, ассоциированными с каждым видимым блоком. Если масочная переменная равна нулю, то она устраняет влияние соответствующего видимого блока на скрытые. Это позволяет убирать видимые блоки, не релевантные задаче классификации, из пути вывода, на котором оценивается класс.

Вообще, инфраструктура машин Больцмана – богатое поле для исследований, где возможных структур моделей гораздо больше, чем изучено до сих пор. Для разработки нового вида машин Больцмана требуется больше тщатнльности и изобретательности, чем для разработки нового слоя нейронной сети, поскольку зачастую трудно подобрать функцию энергии, допускающую обсчет всевозможных условных распределений, которые необходимы для использования модели. Несмотря на требуемые немалые усилия, эта область остается открытой для инноваций.

Обратное распространение через случайные операции

В традиционных нейронных сетях реализовано детерминированное преобразование некоторых входных переменных x. Но при разработке порождающих моделей часто желательно наделить нейронную сеть способностью к стохастическим преобразованиям x. Один из способов добиться этой цели – пополнить нейронную сеть дополнительными входами z, выбранными из какого-нибудь простого распределения, например равномерного или нормального. Тогда на внутреннем уровне сеть будет и дальше выполнять детерминированные вычисления, но наблюдателю, не имеющему доступа к z, функция $f(x, z)$ будет казаться стохастической. При условии что f непрерывна и дифференцируема, мы можем как обычно вычислить градиенты, необходимые для обучения методом обратного распространения.

В качестве примера рассмотрим операцию, состоящую из выборки примеров y из нормального распределения со средним μ и дисперсией σ^2:

$$y \sim \mathcal{N}(\mu, \sigma^2) \tag{13}$$

Поскольку отдельный пример y порождается не функцией, а процессом выборки, выдающим новый результат при каждом запросе, взятие производных y по параметрам распределения μ и σ^2 может показаться противоречащим интуиции. Однако мы можем переформулировать процесс выборки как

преобразование случайной величины $z \sim N(z; 0,1)$ для получения примера из желаемого распределения:

$$y = \mu + \sigma z \tag{14}$$

Теперь можно выполнить обратное распространение через операцию выборки, рассматривая ее как детерминированную операцию с дополнительным входом z. Важно, что дополнительный вход – это случайная величина, распределение которой не является функцией от любой из величин, чьи производные мы хотим вычислять. Этот результат говорит, как бесконечно малое изменение μ или σ отразилось бы на выходе, если бы мы могли повторить операцию выборки с тем же значением z.

Зная, как выполнить обратное распространение через эту операцию выборки, мы можем включить ее в объемлющий граф. Можно строить элементы графа на базе выхода выборочного распределения. Например, мы можем вычислять производные некоторой функции потери $J(y)$. Можно также строить элементы графа, выходы которых являются входами или параметрами операции выборки. Например, можно было бы построить больший граф с $\mu = f(x; \theta)$ и $\sigma = g(x; \theta)$. В этом пополненном графе можно воспользоваться обратным распространением через эти функции для вычисления $\nabla_\theta J(y)$.

Принцип, использованный в этом примере выборки из нормального распределения, применим и в более общей ситуации. Можно выразить любое распределение вероятности вида $p(y; \theta)$ или $p(y|x; \theta)$ как $p(y|\omega)$, где ω – переменная, содержащая как параметры θ, так и (если это осмыслено) входы x. Зная значение y, выбранное из распределения $p(y|\omega)$, где ω может, в свою очередь, быть функцией от других переменных, мы можем переписать

$$y \sim p(y|\omega) \tag{15}$$

в виде

$$y = f(z; \omega), \tag{16}$$

где z – источник случайности. Затем можно вычислить производные y по ω с помощью традиционных средств, например алгоритма обратного распространения в применении к f в предположении, что f непрерывна и дифференцируема почти всюду. Важно, что ω не должна быть функцией z, а z не должна быть функцией ω. Эту технику часто называют перепараметризацией, стохастическим обратным распространением или методом малых возмущений.

Из требования о непрерывности и дифференцируемости f, конечно, вытекает, что y должна быть непрерывна. Если мы хотим выполнять обратное распространение через процесс выборки, порождающий дискретные примеры, то

все же возможно оценить градиент по ω, применяя алгоритмы обучения с подкреплением, например варианты алгоритма REINFORCE

В приложениях нейронных сетей мы обычно выбираем z из какого-нибудь простого распределения, например равномерного или нормального, а чтобы получить более сложные распределения, разрешаем детерминированной части сети изменять форму входа.

Идея распространения градиентов или оптимизации посредством стохастических операций восходит еще к середине XX столетия и впервые была применена к машинному обучению в контексте обучения с подкреплением. Она применялась к вариационным аппроксимациям и к стохастическим и порождающим нейронным сетям. Многие сети, в т. ч. шумоподавляющие автокодировщики и сети, регуляризируемые методом прореживания, также естественно проектируются для приема шума на входе, не требуя специальной перепараметризации, для того чтобы сделать шум независимым от модели.

Обратное распространение через дискретные стохастические операции

Если модель выдает на выходе дискретную переменную y, то перепараметризация неприменима. Предположим, что модель принимает входы x и параметры θ, инкапсулированные вектором ω, и объединяет их со случайным шумом z для порождения y:

$$y = f(z; \omega), \tag{17}$$

Поскольку y дискретна, f должна быть кусочно-постоянной функцией. Производные такой функции бесполезны во всех точках. В точках разрыва производная не определена, но это еще меньшая из бед. Настоящая беда в том, что производная равна нулю на участках постоянства, т. е. почти всюду. Поэтому производные любой функции стоимости $J(y)$ ничего не говорят о том, как обновлять параметры модели θ.

Алгоритм REINFORCE (REward Increment = nonnegative Factor × Offset Reinforcement × Characteristic Eligibility (Приращение вознаграждения = неотрицательный коэффициент × компенсационное подкрепление × характеристика соответствия критериям)) предлагает инфраструктуру для определения семейства простых, но очень эффективных решений. Основная идея заключается в том, что хотя $J(f(z; \omega))$ – кусочно-постоянная функция с бесполезными производными, ожидаемая стоимость $\mathbb{E}_{z \sim p(z)} J(f(z; \omega))$ часто является гладкой функцией, пригодной для градиентного спуска.

Хотя это математическое ожидание обычно вычислительно неразрешимо, если размерность y велика (или y является результатом композиции большого числа дискретных стохастических решений), для него можно получить

несмещенную оценку, вычислив среднее методом Монте-Карло. Стохастическую оценку градиента можно использовать совместно с алгоритмом СГС (стохастического градиентного спуска) или другим методом стохастической градиентной оптимизации.

Простейший вариант алгоритма REINFORCE получается, если просто продифференцировать ожидаемую стоимость:

$$\mathbb{E}_z[J(y)] = \sum_y J(y)\, p(y) \tag{18}$$

$$\frac{\partial \mathbb{E}[J(y)]}{\partial \omega} = \sum_y J(y) \frac{\partial p(y)}{\partial \omega} = \sum_y J(y) \frac{\partial \log p(y)}{\partial \omega} \approx \frac{1}{m} \sum_{y^{(i)} - p(y), i=1}^{m} J(y^{(i)}) \frac{\partial \log p(y^{(i)})}{\partial \omega} \tag{19}$$

Уравнение (17) опирается на предположение, что J не ссылается на ω напрямую. Ослабить это предположение и тем самым обобщить решение очень просто. В уравнении (18) использовано правило дифференцирования логарифма $\frac{\partial \log p(y)}{\partial \omega} = \frac{1}{p(y)} \frac{\partial p(y)}{\partial \omega}$. Уравнение (19) дает несмещенную оценку градиента методом Монте-Карло.

Всюду, где встречается $p(y)$, можно было бы с тем же успехом написать $p(y|x)$, поскольку $p(y)$ параметризовано ω, а ω содержит θ и x, если x вообще присутствует.

Эта простая оценка по алгоритму REINFORCE обладает одним недостатком – очень высокой дисперсией, поэтому для получения хорошей оценки градиента нужно выбрать много примеров y. Иначе говоря, если выбрать только один пример, то алгоритм СГС (стохастического градиентного спуска) будет сходиться очень медленно и потребуется уменьшать скорость обучения. Дисперсию оценки можно значительно снизить, воспользовавшись методами снижения дисперсии. Идея в том, чтобы модифицировать оценку таким образом, что математическое ожидание остается неизменным, а дисперсия уменьшается. В контексте REINFORCE предложенные методы снижения дисперсии включают вычисление базового значения, которое используется для смещения $J(y)$. Отметим, что любое смещение $b(\omega)$, не зависящее от y, не изменяет математического ожидания оценки градиента, потому что

$$E_{p(y)}\left[\frac{\partial \log p(y)}{\partial \omega}\right] = \sum_y p(y)\left[\frac{\partial \log p(y)}{\partial \omega}\right] = \sum_y \frac{\partial p(y)}{\partial \omega} = \frac{\partial}{\partial \omega} \sum_y p(y) = \frac{\partial}{\partial \omega} 1 = 0 \tag{20}$$

а это означает, что

$$E_{p(y)}\left[(J(y) - b(\omega)) \frac{\partial \log p(y)}{\partial \omega}\right] = E_{p(y)}\left[J(y) \frac{\partial \log p(y)}{\partial \omega}\right] - b(\omega) E_{p(y)}\left[\frac{\partial \log p(y)}{\partial \omega}\right] =$$

$$E_{p(y)}\left[J(y) \frac{\partial \log p(y)}{\partial \omega}\right] \tag{21}$$

Далее, мы можем получить оптимальное значение $b(\omega)$, вычислив дисперсию $(J(y) - b(\omega)) \frac{\partial \log p(y)}{\partial \omega}$ относительно распределения $p(y)$ и

минимизировав его относительно $b(\omega)$. В результате мы найдем, что оптимальные базовые значения $b^*(\omega)_i$ различаются для всех элементов ω_i вектора ω:

$$b^*(\omega)_i = \frac{E_{p(y)}\left[J(y)\frac{\partial \log p(y)^2}{\partial \omega_i}\right]}{E_{p(y)}\left[\frac{\partial \log p(y)^2}{\partial \omega_i}\right]} \tag{22}$$

Таким образом, оценка градиента по ω_i принимает вид

$$(J(y) - b(\omega)_i)\frac{\partial \log p(y)}{\partial \omega_i} \tag{23}$$

где $b(\omega)_i$ оценивает приведенное выше значение $b^*(\omega)_i$. Обычно оценку b получают, добавляя новые выходы в нейронную сеть и обучая их оценивать величины $E_{p(y)}\left[J(y)\frac{\partial \log p(y)^2}{\partial \omega_i}\right]$ и $E_{p(y)}\left[\frac{\partial \log p(y)^2}{\partial \omega_i}\right]$ для каждого элемента ω. Эти дополнительные выходы можно обучить, взяв в качестве целевой функции среднеквадратическую ошибку и используя соответственно $J(y)\frac{\partial \log p(y)^2}{\partial \omega_i}$ и $\frac{\partial \log p(y)^2}{\partial \omega_i}$ в качестве целей, когда y выбирается из $p(y)$ для заданного ω. Тогда оценку b можно восстановить, подставив эти оценки в уравнение (22). В некоторых работах отдают предпочтение использованию одного разделяемого (между всеми элементами ω_i) выхода, обученного с меткой $J(y)$, а в качестве базового значения берется $b(\omega) \approx E_{p(y)}[J(y)]$.

Методы снижения дисперсии предложены в контексте обучения с подкреплением, как обобщение предшествующей работы для случая бинарного вознаграждения. Существует достаточно примеров современного использования алгоритма REINFORCE со сниженной дисперсией в контексте глубокого обучения. Помимо использования, зависящего от входа базового значения $b(\omega)$, установлено, что масштаб $(J(y) - b(\omega))$ можно регулировать во время обучения путем деления на его стандартное отклонение, оцененное с помощью скользящего среднего; получается своего рода адаптивная скорость обучения, которая противостоит эффекту важных вариаций абсолютной величины этого значения, имеющих место в процессе обучения. Авторы назвали эту технику эвристической нормировкой дисперсии.

Основанные на алгоритме REINFORCE оценки можно интерпретировать как оценивание градиента путем коррелирования выбора y с соответствующими значениями $J(y)$. Если хорошее значение y при текущей параметризации маловероятно, то может потребоваться много времени на то, чтобы случайно получить его и необходимый сигнал о том, что эту конфигурацию следует подкрепить.