

ЛЕКЦИЯ 1. Введение генеративное (порождающее) моделирование

В общих чертах генеративную модель можно определить так:

Генеративная (порождающая) модель описывает, как генерируется набор данных, с точки зрения вероятностной модели. Используя эту модель, можно генерировать новые данные.

Допустим, есть коллекция изображений объекта и на ее основе нужно построить модель, способную генерировать новые изображения объектов, которых никогда не существовало, но которые выглядят реальными, тому что модель выучила общие правила, определяющие внешний вид объекта. Эту задачу можно решить с помощью генеративного моделирования. Краткое описание типичного процесса генеративного моделирования показано на рис. 1.

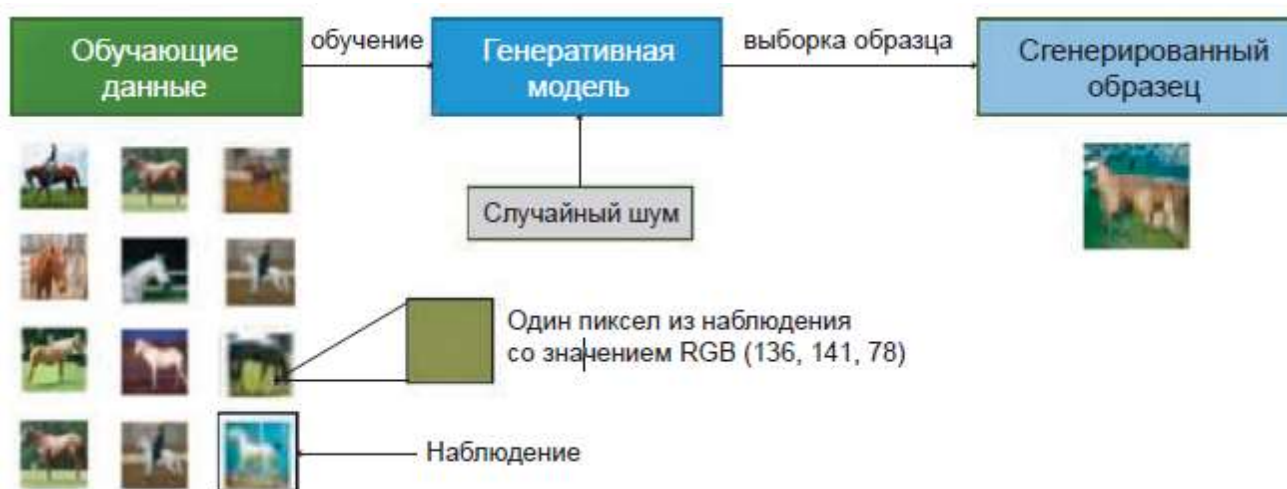


Рис. 1. Процесс генеративного моделирования

Прежде всего, необходим набор данных, состоящий из множества образцов сущности, которую нужно сгенерировать. Этот набор данных называется обучающим набором, а один образец данных в наборе называется наблюдением

Каждое наблюдение состоит из множества признаков — в задачах генерации изображений роль признаков обычно играют отдельные пиксели. Наша цель — создать модель, способную генерировать новые наборы признаков, которые выглядят так, будто созданы с использованием тех же правил, что и исходные данные. Концептуально генерация изображений — невероятно сложная задача, учитывая огромное количество способов выбора значений для отдельных пикселей и относительно крошечное число вариантов такого их расположения, когда получается изображение, похожее на моделируемый объект.

Генеративная модель также должна быть вероятностной, а не детерминированной. Если модель просто представляет фиксированные вычисления, например, выбирает среднее значение каждого пиксела в наборе данных, то она не будет генеративной, потому что каждый раз будет давать один

и тот же результат. Модель должна включать стохастический (случайный) элемент, который влияет на отдельные выборки, генерируемые моделью.

Другими словами, мы можем представить, что существует какое-то неизвестное вероятностное распределение, объясняющее, почему одни изображения могли бы присутствовать в обучающем наборе, а другие нет. Наша задача — создать модель, максимально точно имитирующую это распределение, а затем произвести выборку из нее, чтобы сгенерировать новые наблюдения, которые выглядят так, будто могли бы иметься в исходном обучающем наборе.

Генеративное и дискриминативное моделирование

Чтобы по-настоящему понять цель и важность генеративного моделирования, полезно сравнить его со своим аналогом, дискриминативным моделированием. Знакомые с машинным обучением знают, что большинство задач, с которыми вы столкнетесь, скорее всего, носят дискриминативный характер. Рассмотрим пример, чтобы понять разницу.

Предположим, у нас есть набор данных с коллекцией картин, часть которых написаны Ван Гогом, а часть — другими художниками. Имея достаточный объем данных, мы сможем обучить дискриминативную модель, способную предсказать, была ли данная картина написана Ван Гогом. Наша модель может выучить, какие цвета, формы и текстуры с большей вероятностью будут указывать на принадлежность картины кисти голландского мастера, и в соответствии с этими характеристиками оценивать свой прогноз. На рис. 2 показан процесс дискриминативного моделирования — обратите внимание, как он отличается от процесса генеративного моделирования, изображенного на рис. 1.



Рис. 2. Процесс дискриминативного моделирования

Одно из ключевых отличий состоит в том, что при выполнении дискриминативного моделирования каждое наблюдение в обучающих данных имеет метку. Для задачи бинарной классификации, такой как определение принадлежности картины, картины Ван Гога будут помечены меткой 1, а картины

других художников — меткой 0. Исследовав этот набор, наша модель научится различать эти две группы и выведет вероятность, что новое наблюдение имеет метку 1, то есть что эта картина нарисована Ван Гогом.

По этой причине дискриминативное моделирование часто называют обучением с учителем или определением функции отображения входных данных в выходные с использованием маркированного набора данных. Для построения генеративных моделей обычно используются наборы данных без меток (то есть это форма обучения без учителя), хотя также можно использовать наборы данных с метками, чтобы узнать, как генерировать наблюдения из каждого отдельного класса.

Давайте рассмотрим некоторые математические обозначения, помогающие описать разницу между генеративным и дискриминативным моделированием.

Дискриминативная модель оценивает $p(y|x)$ — вероятность метки y для данного наблюдения x .

Генеративная модель оценивает $p(x)$ — вероятность получения наблюдения x . Если набор данных содержит метки, то можно построить генеративную модель, оценивающую распределение $p(x|y)$.

Иначе говоря, дискриминативная модель оценивает вероятность того, что наблюдение x относится к категории y . Генеративная модель не учитывает метки наблюдений и оценивает вероятность того, что сгенерированное наблюдение похоже на остальные наблюдения. Важно отметить, что даже если бы мы были в состоянии создать идеальную дискриминативную модель для идентификации картин Ван Гога, то она все равно не смогла бы создать картину, похожую на картины Ван Гога. Она сможет лишь определять вероятности для существующих изображений, потому что ее учили именно этому. Нам же нужно обучить генеративную модель, которая сможет генерировать наборы пикселей, с высокой вероятностью принадлежащие исходному набору обучающих данных.

Если рассматривать классификаторы (рис. 3)

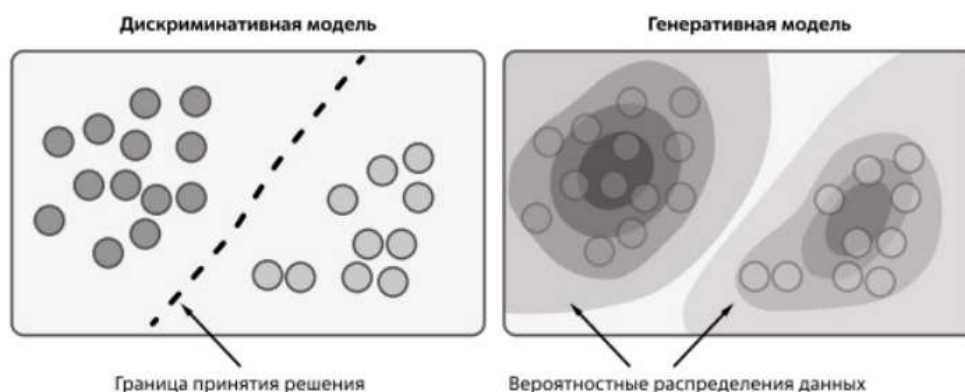


Рис. 3. Сравнение дискриминативного и генеративного моделирования.

Дискриминативные классификаторы пытаются определить, какие признаки на входных данных наиболее полезны для классификации. Модель данного типа пытается определить границу, отделяющую один класс от другого. При этом дискриминативные модели классифицируют точки данных, не задаваясь целью определить, как эти точки были созданы.

Генеративные классификаторы пытаются моделировать класс. Когда этим классификаторам дается новое наблюдение, они пытаются предсказать, какой класс, скорее всего, породил данное наблюдение.

Как правило, дискриминативный классификатор более точен, поскольку пытается непосредственно решить задачу классификации, а не более общую задачу (в качестве промежуточного шага), как это делают генеративные.

Генеративные модели основаны на использовании методов неконтролируемого машинного обучения, в процессе которого осуществляется автоматическое обнаружение и изучение закономерностей во входных данных таким образом, чтобы созданную модель можно было использовать для генерации новых примеров, которые, вероятно, могли быть получены из исходного набора данных.

Достижения в машинном обучении

Чтобы понять, почему генеративное моделирование можно считать следующим рубежом машинного обучения, сначала нужно разобраться, почему дискриминативное моделирование послужило движущей силой для большинства достижений в методологии машинного обучения за последние два десятка лет, как в науке, так и в промышленности. С академической точки зрения прогресс в дискриминативном моделировании легко проследить, так как есть возможность измерить показатели производительности по определенным задачам классификации и выявить лучшую методологию в своем классе. Оценить генеративные модели труднее, особенно когда оценка качества получаемых результатов в значительной степени субъективна. Поэтому в последние годы большое внимание уделялось обучению дискриминативных моделей для достижения надежности классификации изображений или текста, сравнимой с человеческой или даже превосходящей ее.

Наряду с простотой публикации измеримых результатов в академической среде, дискриминативное моделирование исторически проще применить для решения практических задач, чем генеративное моделирование. При решении практических задач нас, как правило, не волнует, как были сгенерированы данные, нас интересует лишь, как классифицировать или оценить новый образец. Например:

- при изучении спутникового снимка сотрудника министерства обороны будет интересовать только вероятность присутствия на нем вражеских подразделений, а не вероятность того, что это изображение является спутниковым снимком;
- менеджеру по работе с клиентами будет интереснее узнать, какой эмоциональный настрой имеет полученное им электронное письмо, положительный или отрицательный, и он едва ли посчитает полезной генеративную модель, которая может генерировать образцы электронных писем клиентов, которых не существует в действительности;
- врачу полезнее будет знать вероятность наличия признаков глаукомы на данном снимке сетчатки глаза, а не возможность генерировать новые изображения, похожие на снимки задней стенки глазного яблока.

Поскольку большинство решений, имеющих практическое применение, относится к области дискриминативного моделирования, наблюдается рост числа инструментов вида машинное обучение как услуга нацеленных на коммерциализацию дискриминативного моделирования в промышленности и автоматизацию процессов сборки, проверки и мониторинга, являющихся универсальными для почти всех задач дискриминативного моделирования

Появление генеративного моделирования

Несмотря на то что прогресс в области машинного обучения до сих пор в значительной степени обеспечивало дискриминативное моделирование, наиболее интересные достижения в этой области в последние 3–5 лет стали результатом применения методов глубокого обучения для задач генеративного моделирования.

В частности, возросло внимание средств массовой информации к таким проектам генеративного моделирования, как StyleGAN от NVIDIA, который способен создавать очень реалистичные изображения человеческих лиц, и языковая модель GPT-2 от OpenAI, которая может завершить отрывок текста по короткому вступительному абзацу.

Достижения в генеративном моделировании с успехом могут использоваться в таких отраслях, как разработка компьютерных игр и кинематография, и в этих же областях наверняка найдут практическое применение усовершенствованные модели автоматического создания музыки. Еще неизвестно, будем ли мы в обозримом будущем читать новостные статьи или романы, написанные генеративной моделью, но последние достижения в этой области ошеломляют и позволяют надеяться, что этот день когда-нибудь настанет. Однако, несмотря на восхищение от новых достижений, возникают

также этические вопросы, связанные с распространением фальшивого контента в интернете, а это означает, что доверять всему, что поступает к нам по общедоступным каналам связи, становится все труднее.

Помимо практического применения (многие варианты которого пока не обнаружены), существуют еще три причины, по которым генеративное моделирование можно считать ключом к гораздо более сложным формам искусственного интеллекта, недоступным для дискриминативного моделирования.

Во-первых, с чисто теоретической точки зрения мы не должны довольствоваться успехами в области классификации данных, но должны стремиться к более полному пониманию, как эти данные были созданы. Это, несомненно, более трудная задача из-за высокой размерности пространства возможных выходных данных и относительно небольшого числа творений, которые мы классифицировали бы как принадлежащие набору данных. При этом, многие из методов, которые привели к развитию дискриминативного моделирования, такие как глубокое обучение, могут использоваться и в генеративных моделях.

Во-вторых, весьма вероятно, что генеративное моделирование окажет существенное влияние на выбор направления будущих разработок в других областях машинного обучения, таких как обучение с подкреплением (когда обучаемый агент получает возможность оптимизировать цели методом проб и ошибок). Например, обучение с подкреплением можно использовать, чтобы обучить робота ходить по данной местности. Для этого можно построить компьютерную имитацию местности, а затем провести множество экспериментов, дав агенту возможность опробовать разные стратегии. Со временем агент выявит наиболее успешные стратегии и, следовательно, будет постепенно совершенствоваться. Типичная проблема этого подхода состоит в том, что физика окружающей среды часто очень сложна и ее необходимо рассчитывать на каждом временном шаге, чтобы передать информацию агенту для принятия решения о его следующем шаге. Однако если агент сможет моделировать свое окружение с помощью генеративной модели, то ему не придется проверять стратегию в компьютерной модели или в реальном мире, поскольку он сможет учиться в своем воображаемом окружении.

Наконец, если мы действительно зададимся целью создать машину, обладающую интеллектом, сравнимым с человеческим, то нам определенно потребуется использовать приемы генеративного моделирования. Один из ярких примеров генеративной модели в природе — человек. Проведите мысленный

эксперимент, который поможет вам понять, насколько невероятной генеративной моделью вы являетесь. Вы можете закрыть глаза и представить, как будет выглядеть слон под любыми возможными углами зрения. Вы можете вообразить ряд возможных концовок вашего любимого телешоу и спланировать свою неделю наперед, прорабатывая разные варианты будущего в своем воображении. Современная нейробиологическая теория предполагает, что наше восприятие реальности — это не сложная дискриминативная модель, получающая информацию от органов чувств и производящая предсказания на ее основе, а генеративная модель, которая с рождения обучается моделированию нашего окружения, точно соответствующего будущему. Некоторые теории даже предполагают, что результатом этой генеративной модели является наше непосредственное восприятие реальности. Очевидно, что глубокое понимание того, как создавать машины, обладающие этими способностями, будет иметь ключевое значение для нашего дальнейшего понимания работы мозга в частности и искусственного интеллекта в целом.

Для начала рассмотрим простейшие примеры генеративных моделей, некоторые мы уже изучали.

Основа для генеративного моделирования

Рассмотрим генеративное моделирование в двух измерениях. Выберем правило, согласно которому был сгенерирован набор точек X , изображенных на рис. 4. Назовем это правило p_{data} . Задача состоит в том, чтобы выбрать другую точку $x = (x_1, x_2)$ в пространстве, которая выглядит так, будто она сгенерирована тем же правилом.

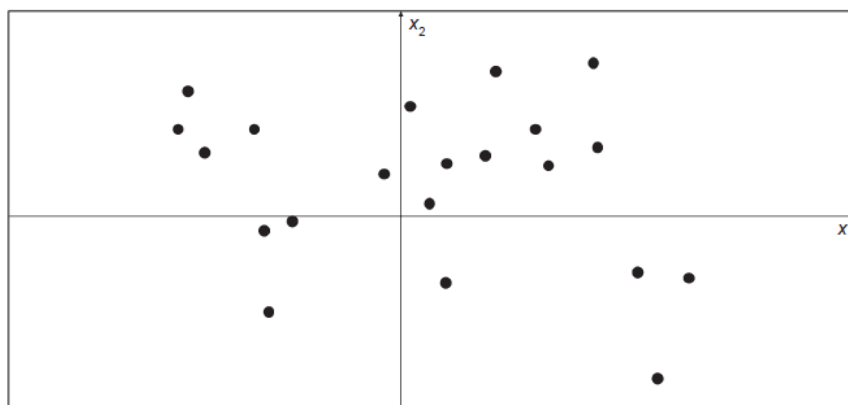


Рис. 4. Набор точек на двумерной плоскости, сгенерированный с использованием неизвестного правила p_{data}

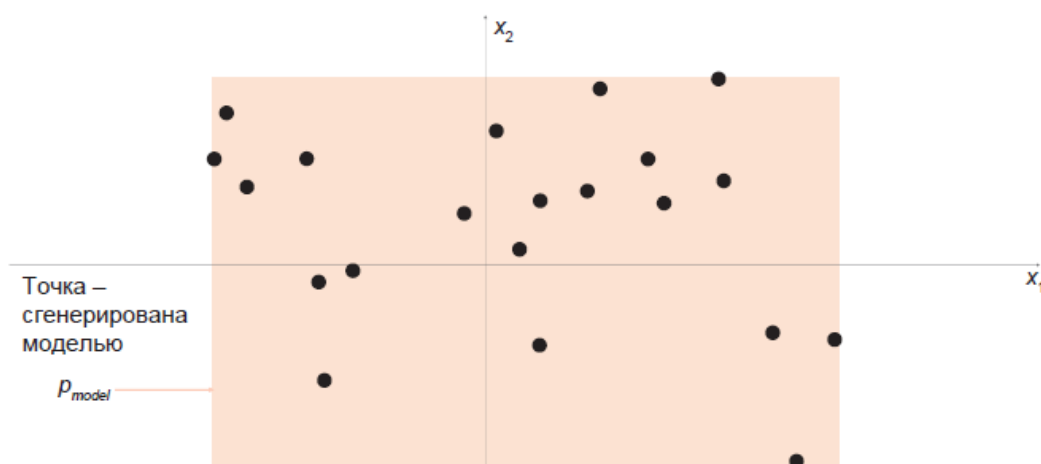


Рис. 5. Оранжевый прямоугольник p_{model} — это оценка фактического распределения p_{data}

Используя свои знания о существующих точках, чтобы построить ментальную модель p_{model} и определить, где в пространстве должна находиться эта точка. В этом отношении p_{model} является оценкой p_{data} . Возможно, что p_{model} должна выглядеть, как показано на рис. 5, — прямоугольное поле, в котором могут находиться точки, и область за границами прямоугольника, где точки находиться не должны. Чтобы сгенерировать новое наблюдение, можно просто выбрать случайную точку внутри поля или, выражаясь более формальным языком, выбрать образец из распределения p_{model} . Это может рассматриваться как генеративная модель.

Это не самый сложный пример, но его можно использовать, чтобы понять цель генеративного моделирования. Следующие базовые принципы определяют наши мотивы.

Теперь раскроем истинное распределение данных p_{data} и посмотрим, как базовые принципы применяются к этому примеру. Как показано на рис. 6, правило, на основе которого получены данные, — это просто равномерное распределение точек по земной суше.

Очевидно, что наша модель p_{model} — это упрощенное представление p_{data} .

Точки A , B и C соответствуют трем наблюдениям, сгенерированным моделью p_{model} с разной степенью успеха:

Точка A нарушает правило 1 базовых принципов генеративного моделирования — ясно видно, что она не принадлежит распределению p_{data} , поскольку находится посреди моря.

Точка B настолько близко расположена от точки из исходного набора данных, что мы не впечатлены способностью модели создать такую точку. Если все образцы, сгенерированные моделью, будут расположены так же близко к

точкам из исходного набора, то это можно считать нарушением правила 2 базовых принципов генеративного моделирования.

Точку C можно рассматривать как успех, потому что она вполне могла быть получена из распределения p_{data} и существенно отличается от всех точек в исходном наборе.

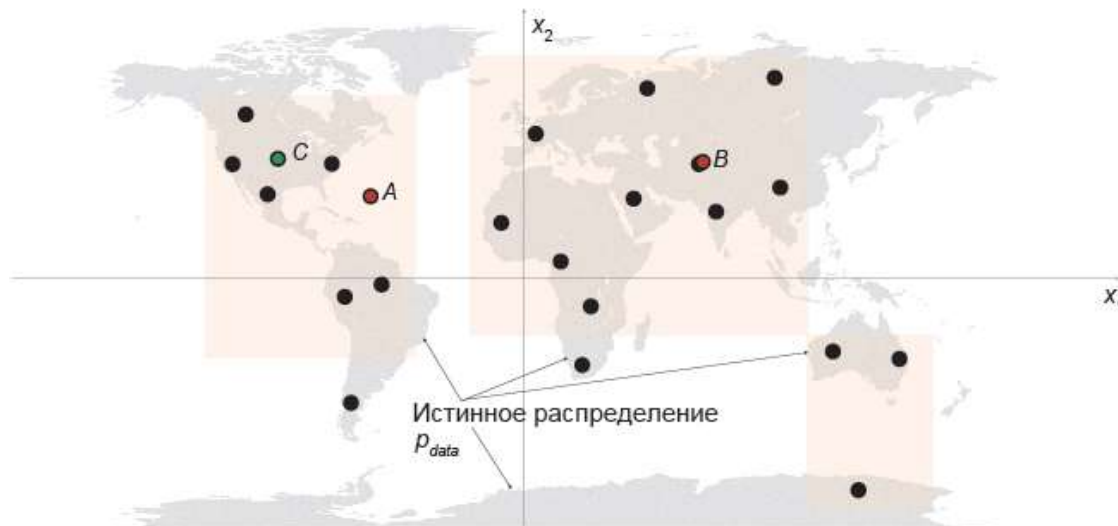


Рис. 6. Оранжевые прямоугольники p_{model} — это оценка истинного распределения p_{data} (серые области)

На основе рассмотренного примера рассмотрим базовые принципы генеративного моделирования.

Есть набор данных с наблюдениями X .

- Предполагается, что наблюдения сгенерированы в соответствии с некоторым неизвестным распределением p_{data} .
- Генеративная модель p_{model} пытается имитировать p_{data} . Правильно подобрав модель p_{model} , мы сможем с ее помощью генерировать наблюдения, которые выглядят так, будто были получены из p_{data} .
- Модель p_{model} впечатлит нас, если:

Правило 1: она сможет генерировать образцы, которые выглядят так, будто получены из p_{data} .

Правило 2: она сможет генерировать образцы, отличающиеся от наблюдений в X . То есть модель не должна просто воспроизводить уже известные ей наблюдения.

Вероятностные генеративные модели

Чтобы получить полное представление об задаче порождающего моделирования стоит попробовать создать генеративную модель, которая опирается не на глубокое обучение, а исключительно на теорию вероятностей. Это поможет заложить основы для понимания всех генеративных моделей,

основанных и не основанных на глубоком обучении, с одной и той же вероятностной точки зрения.

Вспомним и определим четыре ключевых термина:

ФУНКЦИЯ ПЛОТНОСТИ ВЕРОЯТНОСТИ

Функция плотности вероятности (или просто функция плотности), $p(x)$, отображает точку x из выборочного пространства в число от 0 до 1. Сумма функции плотности по всем точкам в выборочном пространстве должна быть равна 1, то есть это четко определенное распределение вероятностей. В примере с картой мира функция плотности нашей модели равна 0 за пределами оранжевого прямоугольника и постоянна внутри него.

ПАРАМЕТРИЧЕСКОЕ МОДЕЛИРОВАНИЕ

Параметрическая модель $p_\theta(x)$ — это семейство функций плотности, которое можно описать с использованием конечного числа параметров θ . Семейство всех возможных прямоугольников, которые можно было бы нарисовать на рис. 6, является примером параметрической модели. В данном случае мы имеем четыре параметра: координаты левого нижнего (θ_1, θ_2) и правого верхнего (θ_3, θ_4) углов прямоугольника. Таким образом, каждая функция плотности $p_\theta(x)$ в этой параметрической модели (то есть каждый прямоугольник) может быть уникально представлена четырьмя числами: $\theta = (\theta_1, \theta_2, \theta_3, \theta_4)$.

ПРАВДОПОДОБИЕ

Вероятность $\mathcal{L}(\theta | x)$ набора параметров θ является функцией, которая измеряет правдоподобие θ с учетом некоторой наблюдаемой точки x .

Определяется она как

$$\mathcal{L}(\theta | x) = p_\theta(x)$$

То есть правдоподобие θ некоторой наблюдаемой точки x определяется как значение функции плотности, параметризованной θ , в точке x . При наличии полного набора данных X независимых наблюдений можно написать:

$$\mathcal{L}(\theta | X) = \prod_{x \in X} p_\theta(x)$$

Поскольку с вычислительной точки зрения найти это произведение довольно сложно, вместо него часто используется логарифм правдоподобия ℓ :

$$\ell(\theta | X) = \sum_{x \in X} p_\theta(x)$$

Есть вполне определенные статистические причины, объясняющие, почему правдоподобие определяется именно так. Во многих приложениях необходимо найти максимум функции правдоподобия, что связано с вычислением производной. Логарифм — монотонно возрастающая функция, поэтому логарифм

от функции достигнет максимума в той же точке, что и сама функция. С другой стороны, логарифм произведения является суммой, что упрощает дифференцирование. Поэтому для практических вычислений предпочитают использовать логарифм функции правдоподобия. Но нам в принципе достаточно будет интуитивного понимания. Тогда определяем правдоподобие того, что набор параметров θ равен вероятности увидеть данные в рамках модели, параметризованной θ . В примере с картой мира оранжевый прямоугольник, покрывающий только левую половину карты, имел бы правдоподобие 0 — он не сможет сгенерировать набор данных, потому что мы наблюдаем точки и в правой половине карты. Оранжевый прямоугольник на рис. 6 имеет положительное правдоподобие, так как функция плотности положительна для всех точек данных в этой модели.

Существует только одна истинная функция плотности p_{data} , которая, как предполагается, сгенерировала наблюдаемый набор данных, и бесконечное число функций плотности p_{model} , которые можно использовать для оценки p_{data} . Чтобы структурировать подход к поиску подходящей $p_{model}(X)$, мы можем использовать метод, известный как параметрическое моделирование. Параметрическое моделирование — это моделирование с использованием параметров элементов модели и соотношений между этими параметрами.

Поэтому интуитивно понятно, что цель параметрического моделирования — поиск оптимального значения набора параметров, которое максимизирует вероятность наблюдения набора данных X . Этот метод вполне уместно называют оценкой максимального правдоподобия.

ОЦЕНКА МАКСИМАЛЬНОГО ПРАВДОПОДОБИЯ

Оценка максимального правдоподобия — это метод, позволяющий оценить (набор параметров θ функции плотности $p_{\theta}(x)$), которые наиболее вероятно объясняют некоторые наблюдаемые данные X .

Более формально:

$$\hat{\theta} = \arg \max_{\theta} \mathcal{L}(\theta | X)$$

$\hat{\theta}$ также называют оценкой максимального правдоподобия (Maximum Likelihood Estimate, MLE).

Пример реализации генеративной модели.

Необходимо создать новые стили моды, похожие на те, что уже существуют, но не идентичные им. Предоставляется набор данных с 50 наблюдениями моды (рис. 7). Создавая шедевры (10 образов), можно

экспериментировать с прическами, цветом волос, очками, типом и цветом одежды.



Рис. 7. 50 портретов

Набор данных, состоит из $N = 50$ наблюдений модных стилей. Каждое наблюдение можно описать пятью признаками (тип аксессуаров, цвет одежды, тип одежды, цвет волос, прическа), как показано в табл. 1.

№ п/п	Аксессуар	Цвет одежды	Тип одежды	Цвет волос	Прическа
1	Круглые очки	Белый	Футболка с круглым вырезом	Рыжий	Короткая стрижка, прямые волосы
2	Круглые очки	Белый	Комбинезон	Серебристо- серый	Короткая стрижка, курчавые волосы
3	Солнцезащитные очки	Белый	Футболка с круглым вырезом	Блонд	Короткая стрижка, прямые волосы
4	Круглые очки	Белый	Футболка с круглым вырезом	Рыжий	Длинная стрижка, прямые волосы
5	Круглые очки	Белый	Комбинезон	Серебристо- серый	Нет волос
6	Нет	Белый	Комбинезон	Черный	Длинная стрижка, прямые волосы
7	Солнцезащитные	Белый	Комбинезон	Серебристо-	Длинная

	очки			серый	стрижка, прямые волосы
8	Круглые очки	Белый	Футболка с круглым вырезом	Серебристо- серый	Длинная стрижка, прямые волосы
9	Круглые очки	Розовый	Худи	Серебристо- серый	Длинная стрижка, прямые волосы
10	Круглые очки	Пастельный оранжевый	Футболка с круглым вырезом	Блонд	Длинная стрижка, прямые волосы

Каждый признак может иметь следующие возможные значения:

- 7 причесок:
 - нет волос;
 - длинные волосы, собранные в пучок;
 - длинная стрижка, волнистые волосы;
 - длинная стрижка, прямые волосы;
 - короткая стрижка, волнистые волосы;
 - короткая стрижка, прямые волосы;
 - короткая стрижка, курчавые волосы.
- 6 цветов волос:
 - черный;
 - блонд;
 - каштановый;
 - пастельный розовый;
 - рыжий;
 - серебристо-серый.
- 3 вида очков (аксессуаров):
 - нет очков;
 - круглые очки;
 - солнцезащитные очки.
- 4 типа одежды:
 - худи;
 - комбинезон;
 - футболка с круглым вырезом;
 - футболка с V-образным вырезом.

8 цветов одежды:

- черный;
- синий;
- серый;
- пастельный зеленый;
- пастельный оранжевый;
- розовый;
- красный;
- белый.

Итого возможно $7 \times 6 \times 3 \times 4 \times 8 = 4032$ разные комбинации этих признаков, то есть выборочное пространство насчитывает 4032 точки.

Можно представить, что наш набор данных сгенерирован некоторым распределением p_{data} , в котором одни значения признаков пользуются большим предпочтением, чем другие. Например, как можно видеть на рис. 7, белая одежда, кажется, пользуется большой популярностью, так же как серебристо-серые волосы и футболки с круглым вырезом.

Проблема в том, что истинное распределение p_{data} неизвестно. Нам доступна только выборка наблюдений X , сгенерированных распределением p_{data} . Целью генеративного моделирования является построение модели p_{model} с использованием этих наблюдений, которая может точно имитировать наблюдения, производимые p_{data} . Чтобы достичь этой цели, можно просто назначить вероятность каждой возможной комбинации признаков, основываясь на имеющихся данных. Такая параметрическая модель будет иметь $d = 4031$ параметр — по одному на каждую точку в выборочном пространстве — минус один, потому что значение последнего параметра будет выбираться так, чтобы общая сумма была равна 1. Значит, мы должны оценить параметры модели $(\theta_1, \dots, \theta_{4031})$. Этот конкретный класс параметрических моделей известен как полиномиальное (мультиномиальное) распределение, а оценка максимального правдоподобия каждого параметра определяется как,

$$\hat{\theta}_j = \frac{n_j}{N}$$

где n_j — число наблюдений с j -й комбинацией в наборе данных, а $N = 50$ — общее число наблюдений.

Иначе говоря, оценка каждого параметра — это просто доля от общего количества наблюдений, в которой имела место соответствующая комбинация. Например, следующая комбинация (назовем ее комбинацией 1) дважды появляется в наборе данных (длинная стрижка, прямые волосы; рыжий; круглые очки; футболка с круглым вырезом; белый).

То есть

$$\hat{\theta}_1 = \frac{2}{50} = 0,04$$

Другой пример — комбинация (назовем ее комбинацией 2), которая ни разу не появляется в наборе данных (длинная стрижка, прямые волосы; рыжий; круглые очки; футболка с круглым вырезом; синий).

То есть

$$\hat{\theta}_2 = \frac{0}{50} = 0$$

Таким способом можно вычислить все значения и определить распределение по имеющемуся выборочному пространству. Так как из этого распределения можно выбирать образцы, наш список мог бы называться генеративной моделью. Однако этой модели не хватает главного: она не может генерировать того, что еще не видела, так как для любой комбинации, отсутствующей в исходном наборе данных X , $\hat{\theta}_j = 0$.

Чтобы решить эту проблему, можно назначить дополнительный псевдоотсчет от 1 для каждой возможной комбинации признаков. Этот прием известен как аддитивное сглаживание, в соответствии с которым оценка максимального правдоподобия (MLE) для параметров будет определяться так:

$$\hat{\theta}_j = \frac{n_j + 1}{N + d}$$

Теперь каждая комбинация имеет ненулевую вероятность быть выбранной, включая и те, которые отсутствовали в исходном наборе данных, но даже с этим дополнением модель не является удовлетворительной генеративной моделью, потому что вероятность наблюдения точки не из исходного набора данных является простой константой. Если попытаться использовать такую модель для создания картин Пикассо, то она бы придала одинаковый вес и случайной коллекции красочных пикселей, и копии картины Пикассо, лишь незначительно отличающейся от настоящей. В идеале хотелось бы, чтобы генеративная модель подчеркивала области выборочного пространства, которые, по ее мнению, более вероятны, согласно некоторой внутренней структуре обучающих данных, а не просто присваивала вероятностные веса точкам, присутствующим в наборе данных. Для этого нужно выбрать другую параметрическую модель.

Наивная байесовская параметрическая модель

Наивная байесовская параметрическая модель использует простое предположение, чтобы резко уменьшить количество параметров для оценки. Мы наивно предполагаем, что каждый признак x_j не зависит от любого другого

признака x_k . В отношении набора данных, это означает, например, что выбор цвета волос не влияет на выбор типа одежды, а выбор типа очков не влияет на выбор прически. Более формально, для всех признаков x_j, x_k :

$$p(x_j|x_k) = p(x_j)$$

Это известно, как наивное байесовское предположение. Чтобы применить его, сначала используется цепное правило вероятности для записи функции плотности в виде произведения условных вероятностей:

$$\begin{aligned} p(X) &= p(x_1 \dots x_k) = p(x_2 \dots x_k|x_1)p(x_1) = p(x_3 \dots x_k|x_2, x_1)p(x_2|x_1)p(x_1) \\ &= \prod_{k=1}^K p(x_k|x_1 \dots x_{k-1}) \end{aligned}$$

где K — общее число признаков (то есть пять в примере с модой).

Теперь применим наивное байесовское предположение, чтобы упростить последнюю строку:

$$p(X) = \prod_{k=1}^K p(x_k)$$

Это — наивная байесовская модель. Задача сводится к оценке параметров $\theta_{kl} = p(x_k = l)$ для каждого признака в отдельности и их перемножению для определения вероятности любой возможной комбинации.

Тогда каждого признака нужно оценить параметр для каждого значения, которое может принять этот признак. Следовательно, в примере эта модель определяется всего $7 + 6 + 3 + 4 + 8 - 5 = 23$ параметрами (последний член в выражении -5 отражает тот факт, что последний параметр для каждого признака подбирается так, чтобы сумма его параметров была равна 1).

Оценка максимального правдоподобия $\hat{\theta}_{kl}$ вычисляется как

$$\hat{\theta}_{kl} = \frac{n_{kl}}{N}$$

где $\hat{\theta}_{kl}$ — число раз, когда признак k принимает значение l в наборе данных, а $N = 50$ — общее число наблюдений.

В табл. 2 показаны вычисленные параметры для набора данных.

Таблица 2. Оценки максимального правдоподобия для параметров в наивной байесовской модели

Прическа	n	$\hat{\theta}$
нет волос	7	0,14
длинные волосы,	0	0,00

Цвет волос	n	$\hat{\theta}$
черный	7	0,14
блонд	6	0,12

собранные в пучок		
длинная стрижка, волнистые волосы	1	0,02
длинная стрижка, прямые волосы	23	0,46
короткая стрижка, волнистые волосы	1	0,02
короткая стрижка, прямые волосы	11	0,22
короткая стрижка, курчавые волосы	7	0,14
Всего	50	1,00

каштановый	2	0,04
пастельный розовый	3	0,06
рыжий	8	0,16
серебристо- серый	24	0,48
Всего	50	1,00

Цвет одежды	n	$\hat{\theta}$
черный	0	0,00
синий	4	0,08
серый	10	0,20
пастельный зеленый	5	0,10
пастельный оранжевый	2	0,04
розовый	4	0,08
красный	3	0,06
белый	22	0,44
Всего	50	1,00

Вид очков	n	$\hat{\theta}$
нет	11	0,22
круглые	22	0,44
солнцезащитные	17	0,34
Всего	50	1,00

Тип одежды	n	$\hat{\theta}$
Худи	7	0,14
комбинезон	18	0,36
футболка с круглым вырезом	19	0,38
футболка с V- образным вырезом	6	0,12
Всего	50	1,00

Чтобы найти вероятность, с которой модель сгенерирует некоторое наблюдение x , достаточно перемножить вероятности отдельных признаков. Например:

p (длинная стрижка, прямые волосы; рыжий; круглые очки;

футболка с круглым вырезом; белый)

$$\begin{aligned}
 &= p(\text{длинная стрижка, прямые волосы}) \times p(\text{рыжий}) \\
 &\times p(\text{круглые очки}) \times p(\text{футболка с круглым вырезом}) \\
 &\times p(\text{белый}) = 0,46 \times 0,16 \times 0,44 \times 0,38 \times 0,44 = 0,0054
 \end{aligned}$$

Обратите внимание: эта комбинация отсутствует в исходном наборе данных, но наша модель определяет для нее ненулевую вероятность, а значит, вполне может сгенерировать ее. Кроме того, вероятность этой комбинации выше, чем, например, (длинная стрижка, прямые волосы; рыжий; круглые очки; футболка с круглым вырезом; синий), потому что белый цвет одежды появляется в наборе наблюдений чаще, чем синий. То есть наивная байесовская модель способна выявить некоторую структуру данных и использовать ее для создания новых образцов, отсутствующих в исходном наборе. Модель оценила вероятность встретить каждое значение признака независимо от других, поэтому при использовании наивного байесовского предположения можно перемножить эти вероятности, чтобы построить полную функцию плотности, $p_{\theta}(x)$.

Для этой простой задачи наивное байесовское предположение о независимости признаков является разумным и, следовательно, дает хорошую генеративную модель. На рис.8 представлено 10 образов.



Рис. 8. Десять новых стилей, сгенерированных наивной байесовской моделью

Теперь возьмем набор данных с незамысловатым названием Planet Pixel, который был предоставлен, не содержит пяти высокоуровневых признаков, которые мы применяли (цвет волос, тип аксессуара и т. д.), а только значения 32×32 пикселей, составляющих каждое изображение. То есть каждое наблюдение теперь имеет $32 \times 32 = 1024$ признака и каждый признак может принимать любое из 256 значений (отдельные цвета в палитре).

Изображения из нового набора данных показаны на рис. 9, а выборка значений пикселей для первых десяти наблюдений показана в табл. 3.



Рис. 9. Образцы моды в Planet Pixel

Таблица 3. Значения пикселей 458–467 для первых десяти наблюдений в Planet Pixel

№ образца	px_458	px_459	px_460	px_461	px_462	px_463	px_464	px_465	px_466	px_467
0	49	14	14	19	7	5	5	12	19	14
1	43	10	10	17	9	3	3	18	17	10
2	37	12	12	14	11	4	4	6	14	12
3	54	9	9	14	10	4	4	16	14	9
4	2	2	5	2	4	4	4	4	2	5
5	44	15	15	21	14	3	3	4	21	15
6	12	9	2	31	16	3	3	16	31	2
7	36	9	9	13	11	4	4	12	13	9
8	54	11	11	16	10	4	4	19	16	11
9	49	17	17	19	12	6	6	22	19	17

Решим попробовать наивную байесовскую модель еще раз, на этот раз обученную на наборе данных пикселей. Модель оценит параметры максимального правдоподобия, определяющие распределение цвета каждого пикселя, чтобы на основе этого распределения сгенерировать новые наблюдения. Однако, закончив модель, вы понимаете, что что-то пошло не так. Вместо новых образцов моды модель вывела десять похожих друг на друга изображений, на которых нельзя различить ни аксессуары, ни четкие признаки прически или одежды (рис. 10).



Рис. 10. Десять новых стилей, сгенерированных наивной байесовской моделью на основе набора данных Planet Pixel

Почему так случилось?

Во-первых, поскольку наивная байесовская модель выбирает пиксели независимо друг от друга, она не понимает, что два смежных пиксела обычно имеют похожие оттенки, потому что, например, являются частью одного и того же предмета одежды. Модель может сгенерировать цвет лица и рта, поскольку все эти пиксели в обучающем наборе имеют примерно одинаковый оттенок во всех наблюдениях; однако цвет пикселей для футболки выбирается случайно из множества цветов, присутствующих в обучающем наборе, независимо от цвета соседних пикселей. Кроме того, отсутствует механизм формирования пикселей в области вокруг глаз, позволяющий получить круглые очки, или в верхней части изображения, чтобы получить конкретную прическу.

Во-вторых, на этот раз в выборочном пространстве имеется непостижимо большое количество возможных наблюдений. Лишь небольшая часть из них — узнаваемые лица, а еще меньшее подмножество — лица, которые придерживаются правил моды в Planet Pixel. Следовательно, если наивная байесовская модель обучается на сильно коррелированных значениях пикселей, то вероятность найти удовлетворительную комбинацию значений оказывается очень мала.

В первом случае мы имели независимые признаки и относительно небольшое выборочное пространство, поэтому наивная байесовская модель показала неплохие результаты. Во втором случае, когда данные были представлены наборами пикселей, предположение о независимости значений пикселей оказалось несостоятельным. Значения пикселей сильно коррелированы, а выборочное пространство огромно, поэтому получить правильное изображение путем независимой выборки пикселей практически невозможно. Это объясняет, почему наивные байесовские модели плохо работают с необработанными изображениями.

Рассмотренный нами пример высветил две ключевые проблемы, которые должна преодолеть генеративная модель, чтобы добиться успеха.

- Как модель может справиться с высокой условной взаимозависимостью признаков?
- Как модель может отыскивать одну из крошечных пропорций, чтобы получить удовлетворительное наблюдение в многомерном выборочном пространстве?

Ключом к решению обеих этих проблем является глубокое обучение.

Нам нужна модель, способная выявить релевантную структуру в данных, которая не требует делать каких-либо предположений заранее. Именно с этим прекрасно справляется глубокое обучение, и именно поэтому этот метод стал движущей силой последних достижений в генеративном моделировании. Тот факт, что глубокое обучение может формировать свои признаки в пространстве более низкой размерности, означает, что это — форма обучения представлению.

Обучение представлению

Основная идея обучения представлению заключается в том, чтобы вместо моделирования многомерного выборочного пространства непосредственно попытаться описать каждое наблюдение в обучающем наборе с использованием некоторого скрытого пространства меньшей размерности, а затем определить функцию отображения, которая может взять точку из скрытого пространства и отобразить ее в точку в исходном пространстве. Другими словами, каждая точка в скрытом пространстве является представлением некоторого многомерного изображения. Что это означает с практической точки зрения? Предположим, у нас есть обучающий набор, состоящий из черно-белых изображений форм тортов (рис. 11).

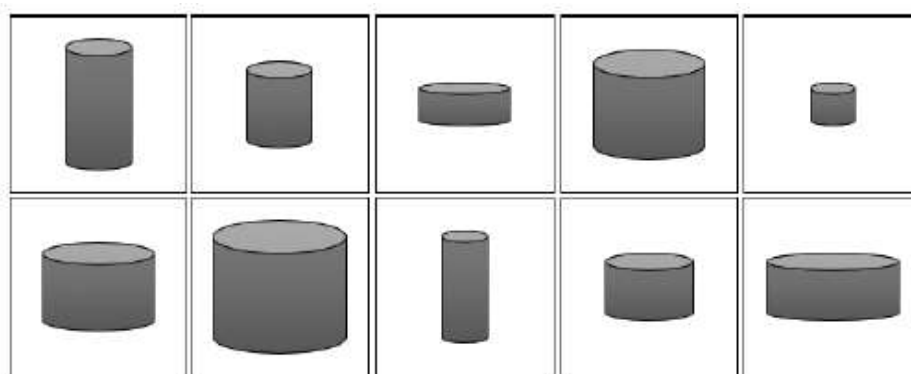


Рис. 11. Набор изображений форм тортов

Совершенно очевидно, что уникально идентифицировать формы тортов можно по двум признакам: высоте и ширине. Зная высоту и ширину, мы сможем нарисовать соответствующую форму, даже если ее изображение отсутствовало в

обучающем наборе. Однако для компьютера это не так просто — сначала он должен понять, что высота и ширина являются двумя измерениями скрытого пространства, которые наилучшим образом описывают этот набор данных, а затем сконструировать функцию отображения f , которая сможет выбрать точку в этом пространстве и привести ее в соответствие с черно-белым изображением формы. Пример такого скрытого пространства форм тортов и процесс генерации показаны на рис. 12.

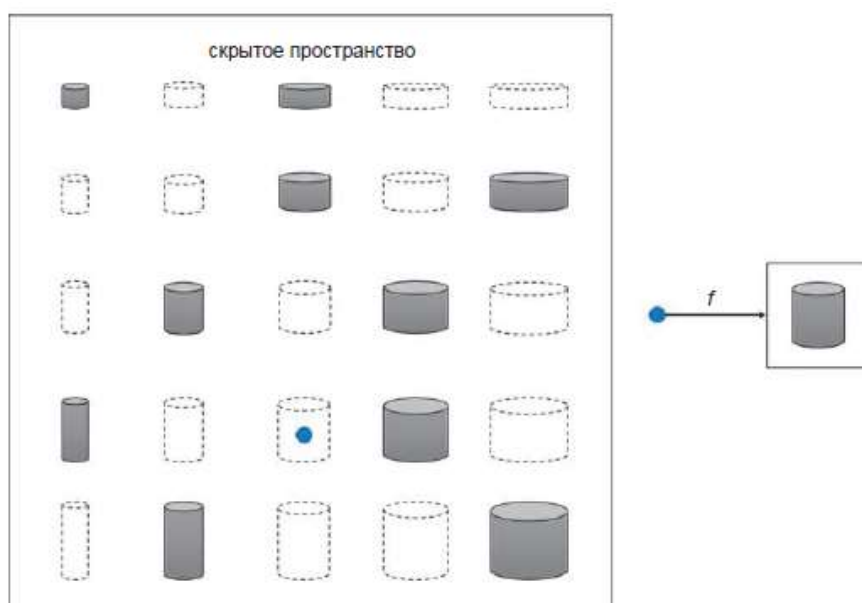


Рис. 12. Скрытое пространство свойств форм тортов и функция f , отображающая точку из скрытого пространства в пространство исходных изображений

Глубокое обучение дает нам возможность разными способами выявлять порой очень сложные функции отображения f .

Таким образом, одно из преимуществ обучения представлению заключается в возможности выполнять операции, влияющие на высокоуровневые свойства изображений, в более управляемом скрытом пространстве. В исходном пространстве изображений неясно, как сгенерировать каждый пиксел, чтобы получить более высокое изображение торта. Однако в скрытом пространстве для этого достаточно увеличить на единицу скрытое измерение, отвечающее за высоту, а затем применить функцию отображения для перехода обратно в пространство изображений. Пример такого подхода применительно не к формам тортов, а к граням.

Обучение представлению для нас, людей, настолько естественно, что вы, возможно, никогда не задумывались о том, с какой удивительной легкостью мы делаем это. Представьте, что вы решили описать свою внешность кому-то, кто должен отыскать вас в толпе и не знает, как вы выглядите. Вы едва ли стали бы описывать цвет всех пикселей, формирующих ваше изображение. Вместо этого

вы предположили бы, что ваш визави имеет представление о том, как выглядит обычный человек, и заменили описание пикселей высокоуровневыми признаками, соответствующими группам пикселей, например: у меня очень светлые волосы или я ношу очки. Если вы назовете с десятков таких признаков, то человек сможет отобразить описание обратно в пиксели и создать ваш образ в своей голове. Конечно, образ может получиться далеко не идеальным, но достаточно близким к вашему фактическому внешнему виду, чтобы вас можно было отыскать среди сотен других людей, даже если вас никогда не видели раньше.

Подчеркнем: обучение представлению не просто присваивает значения определенному набору признаков, таких как светлые волосы, рост и т. д. Сила обучения представлению заключается в том, что этот процесс позволяет выявить в исходных данных наиболее важные признаки, позволяющие описать имеющиеся наблюдения. Говоря математическим языком, цель обучения представлению состоит в том, чтобы найти высоконелинейное многообразие, на котором лежат данные, а затем установить измерения, необходимые для полного описания этого пространства (рис. 13).

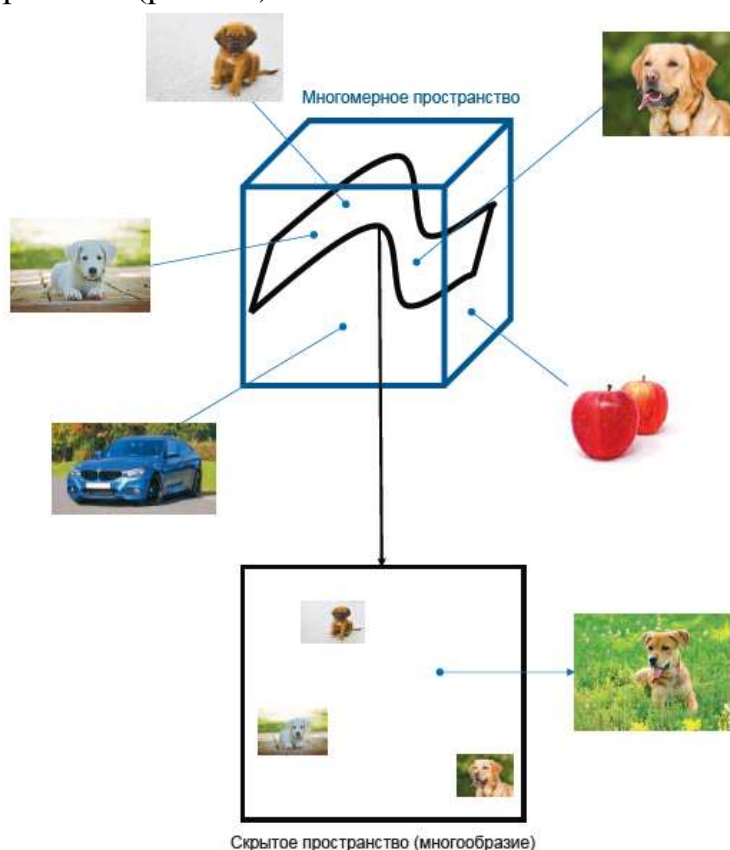


Рис. 1.13. Куб представляет многомерное пространство всех изображений; обучение представлению пытается найти скрытое подпространство, или многообразие, с меньшим числом измерений, на котором располагаются определенные виды изображений (например, многообразие собак)

Таким образом, обучение представлению выявляет наиболее важные высокоуровневые признаки, описывающие группы пикселей, поэтому любая точка в скрытом пространстве вполне может представлять правильно сформированное изображение. Изменяя значения признаков в скрытом пространстве, можно создавать новые представления, которые при отображении в пространство исходных изображений имеют гораздо больше шансов выглядеть похожими на реальные изображения, чем если бы мы пытались работать с отдельными пикселями.