Практическая работа № 4

“Полносвязные нейронные сети”

Цель: изучить основы нейронных сетей. Понять алгоритм обратного распространения ошибки.

Теоретический материал

Нейронные сети

Глубокие сети прямого распространения, которые называют также нейронными сетями прямого распространения, или многослойными перцептронами (МСП), – самые типичные примеры моделей глубокого обучения. Цель сети прямого распространения – аппроксимировать некоторую функцию f\*. Например, в случае классификатора y = f \*(x) отображает вход x в категорию y. Сеть прямого распространения определяет отображение y = f(x; θ) и путем обучения находит значения параметров θ, дающие наилучшую аппроксимацию

Слова «прямое распространение» означают, что распространение информации начинается с x, проходит через промежуточные вычисления, необходимые для определения f, и заканчивается выходом y. Не существует обратных связей, по которым выходы модели подаются на ее вход. Обобщенные нейронные сети, включающие такие обратные связи, называются рекуррентными и рассматриваются в главе 10.

Сети прямого распространения исключительно важны для практического применения машинного обучения. Они лежат в основе многих важных коммерческих приложений. Например, сверхточные сети, используемые для распознавания объектов на фотографиях, – это частный случай сетей прямого распространения. Сети прямого распространения – концептуальная веха на пути к рекуррентным сетям, стоящим за многими приложениями в области естественных языков. Многими приложениями в области естественных языков.

Нейронные сети прямого распространения называются сетями, потому что они, как правило, образованы композицией многих различных функций. С моделью ассоциирован ориентированный ациклический граф, описывающий композицию. Например, можно связать три функции f1, f2 и f3 в цепочку f(x) = f3(f2(f1(x))). Такие цепные структуры чаще всего используются в нейронных сетях. В данном случае f(1) называется первым слоем сети, f(2) – вторым слоем и т. д. Общая длина цепочки определяет глубину модели. Название «глубокое обучение» непосредственно связано с этой терминологией. Последний слой сети прямого распространения называется выходным. В ходе обучения нейронной сети мы стремимся приблизить f(x) к f\*(x). Обучающие данные – это зашумленные приближенные примеры f\*(x), вычисленные в различных точках. Каждый пример x сопровождается меткой y ≈ f\*(x). Обучающие примеры напрямую указывают, что в выходном слое должно соответствовать каждой точке x, – это должно быть значение, близкое к y. Поведение остальных слоев напрямую обучающими данными не определяется. Алгоритм обучения должен решить, как использовать эти слои для порождения желаемого выхода, но обучающие данные ничего не говорят о том, что должен делать каждый слой. Алгоритму обучения предстоит самостоятельно решить, как с помощью этих слоев добиться наилучшей аппроксимации f \* . Поскольку обучающие данные не определяют выходов каждого из этих слоев, они называются скрытыми слоями.

Наконец, сети называются нейронными, потому что их идея заимствована из нейробиологии. Каждый скрытый слой сети обычно вырабатывает векторные значения. Размерность скрытых слоев определяет ширину модели. Каждый элемент вектора можно интерпретировать как нейрон. Вместо того чтобы рассматривать слой как представление функции с векторными аргументами и векторными значениями, мы можем считать, что слой состоит из многих блоков, работающих параллельно, и что каждый такой блок представляет функцию, отображающую вектор в скаляр. Каждый блок напоминает нейрон в том смысле, что получает данные от многих других блоков и вычисляет собственное значение активации. Идея использования многих слоев векторных представлений пришла из нейробиологии. Выбор функций f(i) (x), используемых для вычисления этих представлений, также изначально навеян экспериментально полученными фактами о функциях, вычисляемых биологическими нейронами. Но в основе современных исследований по нейронным сетям лежат математика и инженерные дисциплины, а перед нейронной сетью не ставится цели точно смоделировать работу мозга. Лучше рассматривать сети прямого распространения не как модели функционирования мозга, а как машины для аппроксимации функций, спроектированные с целью статистического обобщения и иногда использующие наши знания о мозге.

Один из способов разобраться в сетях прямого распространения состоит в том, чтобы начать с линейных моделей и подумать, как преодолеть их ограничения. Линейные модели, в т. ч. логистическая регрессия и линейная регрессия, так привлекательны, потому что дают эффективную и надежную аппроксимацию – в замкнутой форме или посредством выпуклой оптимизации. Но у линейных моделей есть очевидный недостаток – емкость модели ограничена линейными функциями, поэтому модель неспособна понять произвольную связь между двумя величинами.

Чтобы обобщить линейную модель на представление нелинейных функций от x, мы можем применить ее не к самому x, а к результату вычисления ϕ(x), где ϕ – нелинейное преобразование. Или, что то же самое, применить трюк с ядром, описанный в разделе 5.7.2, для получения нелинейного алгоритма обучения, основанного на неявном применении преобразования ϕ. Можно считать, что ϕ дает набор признаков, описывающих x, или новое представление x.

Тогда вопрос сводится к выбору отображения ϕ.

1. Один из вариантов – взять очень общее ϕ, например бесконечномерное, неявно используемое в ядерных методах, основанных на радиально-базисном ядре. Если размерность ϕ(x) достаточна велика, то емкости модели хватит для аппроксимации обучающего набора, но обобщаемость на тестовый часто оставляет желать лучшего. Очень общие отображения признаков обычно основаны на принципе локальной гладкости, и закодированной в них априорной информации недостаточно для решения сложных задач.
2. Другой вариант – спроектировать ϕ вручную. До наступления эры глубокого обучения так в основном и поступали. Но для каждой задачи требуются десятилетия человеческого труда и специалисты в соответствующей предметной области, например распознавания речи или компьютерного зрения, а передачи знаний между разными областями почти нет.
3. Стратегия глубокого обучения состоит в обучении ϕ. При таком подходе имеется модель y = f(x; θ, w) = ϕ(x; θ)⏉w. Теперь у нас есть параметры θ, используемые для обучения ϕ, выбираемой из широкого класса функций, и параметры w, отображающие ϕ(x) в желаемый выход. Это пример глубокой сети прямого распространения, где ϕ определяет скрытый слой. Это единственный из трех подходов, который порывает с предположением о выпуклости задачи обучения, но его достоинства перевешивают недостатки. В этом случае мы параметризуем представление в виде ϕ(x; θ) и применяем алгоритм оптимизации для нахождения отображения ϕ, которому соответствует хорошее представление. Если мы пожелаем, то у этого подхода будут все преимущества общности первого – для этого нужно только взять очень широкое семейство функций ϕ(x; θ). Глубокое обучение может также воспользоваться достоинствами второго подхода. Исследователь может включить в модель свои знания, спроектировав семейство функций, которое, по его мнению, должно хорошо обобщаться. Преимущество в том, что человеку нужно только отыскать подходящее семейство функций, а не одну конкретную функцию.

Общий принцип улучшения моделей путем обучения признаков выходит за рамки описываемых в этой главе сетей прямого распространения. Эта тема снова и снова возникает в глубоком обучении и относится ко всем видам моделей, рассматриваемым в книге. Сети прямого распространения – применение этого принципа к обучению детерминированных отображений x в y без обратных связей. Ниже будут представлены другие модели, в которых тот же принцип применяется к обучению стохастических отображений, функций с обратной связью и распределений вероятности одиночного вектора.

Мы начнем эту главу с простого примера сети прямого распространения. Затем мы рассмотрим все проектные решения, принимаемые при развертывании таких сетей. Во-первых, при обучении сети прямого распространения приходится думать о тех же вещах, что для линейных моделей: выборе оптимизатора, функции стоимости и вида выходных блоков. Мы рассмотрим эти основы градиентного обучения, а затем перейдем к решениям, характерным только для сетей прямого распространения. Поскольку в таких сетях есть скрытые слои, то мы должны выбрать функции активации, используемые для вычисления вырабатываемых ими значений. Кроме того, нужно спроектировать архитектуру сети: сколько в ней скрытых слоев, как эти слои связаны между собой, сколько блоков в каждом слое. Обучение в контексте глубоких нейронных сетей подразумевает вычисление градиентов сложных функций. Мы опишем алгоритм обратного распространения и его современные реализации, позволяющие эффективно вычислять градиенты.