Оглавление

4 Введение в нейронное обучение: градиентный спуск

В этой главе

- ✓ Способны ли нейронные сети делать точные прогнозы?
- ✓ Зачем измерять ошибку?
- ✓ Обучение методом «холодно/горячо».
- ✓ Вычисление направления и величины из ошибки.
- ✓ Градиентный спуск.
- ✓ Обучение просто уменьшает ошибку.
- ✓ Производные и как их использовать для обучения.
- ✓ Расхождение и альфа-коэффициент.

Единственный надежный способ проверить гипотезу — сравнить ее прогноз с экспериментальными данными.

Милтон Фридман (Milton Friedman),
Essays in Positive Economics
(издательство Чикагского университета, 1953)

Предсказание, сравнение и обучение

В главе 3 вы познакомились с парадигмой «предсказание, сравнение, обучение» и углубились в изучение первого шага: *предсказания*. Попутно вы узнали много нового, включая основные компоненты нейронных сетей (узлы и веса), как организуется соответствие наборов данных и сети (количество точек данных, одновременно подаваемых на вход) и как нейронная сеть получает прогноз.

Возможно, в процессе чтения у вас возник вопрос: «Как выбрать значения весов так, чтобы сеть получала точные прогнозы?» Ответ на этот вопрос является главной темой этой главы, и здесь мы рассмотрим следующие два шага парадигмы: *сравнение* и *обучение*.

Сравнение

Сравнение позволяет оценить, насколько прогноз «промахнулся»

Следующий шаг после получения прогноза — оценка его качества. Это может показаться простым делом, но, как вы убедитесь сами, выбор хорошего способа измерения ошибки — одна из самых сложных и важных задач в глубоком обучении.

Есть много разных подходов к измерению ошибок, которые вы наверняка использовали в своей жизни, даже не подозревая об этом. Например, вы (или ваш знакомый) могли преувеличивать большие ошибки и игнорировать мелкие. В этой главе вы познакомитесь с математическим аппаратом, который поможет научить сеть это делать. Вы также узнаете, что ошибка всегда положительна! В качестве аналогии возьмем стрельбу из лука по мишени: если стрела попала в мишень на дюйм выше или ниже, в обоих случаях ошибка составит 1 дюйм. На этапе *сравнения* результатов, полученных от нейронной сети, необходимо учитывать это при оценке ошибки.

Сразу отмечу, что в этой главе мы будем оценивать только один простой способ измерения ошибки: вычисление *среднеквадратической ошибки*. Это лишь один из способов оценки точности нейронной сети.

Этот шаг поможет вам получить представление, насколько вы промахнулись, однако этого недостаточно для обучения. Результатом логики *сравнения* является сигнал «горячо/холодно». Мера ошибки, вычисленная по результатам прогноза, сообщит вам, «насколько сильно» вы промахнулись, но она ничего

не скажет, почему случился промах, в какую сторону вы промахнулись или что нужно сделать, чтобы исправить ошибку, — она лишь скажет «сильно промахнулись», «мало промахнулись» или «попали точно в цель». Исправление ошибки — это уже задача следующего этапа: *обучения*.

Обучение

Процесс обучения определяет, как изменить каждый вес, чтобы уменьшить ошибку

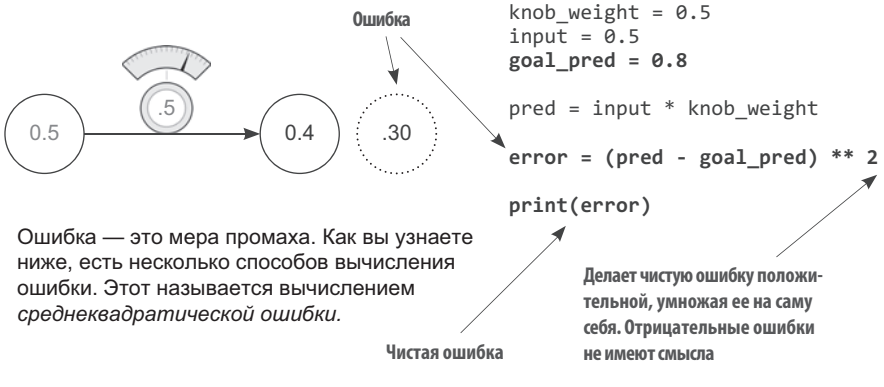
Обучение — это процесс *определения причин ошибок*, или искусство выяснения вклада каждого веса в общую ошибку. Это главная задача глубокого обучения. В этой главе мы много времени уделим изучению одного из самых популярных способов решения этой задачи: *градиентного спуска*.

Этот способ позволяет вычислить для каждого веса некоторое число, определяющее, насколько этот вес должен быть выше или ниже, чтобы уменьшить ошибку. После этого вам останется только изменить вес на это число, и дело в шляпе.

Сравнение: способны ли нейронные сети делать точные прогнозы?

Измерим ошибку и узнаем!

Выполните следующий код в своем блокноте Jupyter Notebook. Он должен вывести `0.3025`:



ЧТО ЭТО ЗА ПЕРЕМЕННАЯ GOAL_PRED?

Переменная `goal_pred`, так же как `input`, хранит число, полученное в реальном мире путем наблюдений, иногда очень сложных, как, например, «процент людей, *надевших* теплую одежду» при данной температуре воздуха; или «*попал ли* отбивающий в хоум-ран»[1].

ПОЧЕМУ ОШИБКА ВОЗВОДИТСЯ В КВАДРАТ?

Представьте лучника, стреляющего в мишень. Допустим, стрела попала в мишень на 2 дюйма выше центра. Насколько промахнулся лучник? А если на 2 дюйма ниже? В обоих случаях лучник промахнулся на 2 дюйма. Основная причина *возведения в квадрат* «величины промаха» заключается в получении *положительного* числа. Выражение `(pred - goal_pred)` может дать отрицательный результат, *в отличие от фактической ошибки*.

РАЗВЕ ВОЗВЕДЕНИЕ В КВАДРАТ НЕ УВЕЛИЧИВАЕТ БОЛЬШИЕ ОШИБКИ (>1) И НЕ УМЕНЬШАЕТ МАЛЕНЬКИЕ (<1)?

Да... Это немного странный способ измерения ошибки, но, как оказывается, *преувеличение* больших ошибок и *преуменьшение* маленьких — это нормально. Позднее вы будете использовать эту ошибку для обучения сети, поэтому лучше сосредоточить внимание на больших ошибках и игнорировать маленькие. Так же поступают хорошие родители: они не замечают мелких ошибок своих детей (например, сломанный грифель карандаша), но могут взорваться в случае большой ошибки (например, если сын или дочь разбили автомобиль). Теперь понимаете, почему возведение в квадрат может быть полезным?

Зачем измерять ошибку?

Измерение ошибки упрощает задачу

Цель обучения нейронной сети — получение достоверных прогнозов. Это наше желание. И в нашем прагматичном мире (как отмечалось в предыдущей главе) хотелось бы иметь сеть, принимающую входные данные, которые легко получить (например, сегодняшние цены на акции), и предсказывающую что-то, что трудно вычислить (завтрашние цены на акции). Это то, что делает нейронные сети полезными.

---

[1] https://ru.wikipedia.org/wiki/Бейсбольная_терминология.