Обучение нейронных сетей: функция потерь

Нейрон дважды ошибся в одной точке, но второй раз был ближе к правде. Как измерить ошибку так, чтобы заметить это улучшение?

Ранее

На карте робота мы различили выход нейрона, вероятностную оценку и выбранный класс. Теперь сравним этот выход с известной меткой места.

← Классификация и вероятности классов

В прошлом сюжете робот оценивал безопасность одного места. Мы могли менять веса и видеть, как уверенно неверный ответ становится менее уверенным, а затем правильным. Но программе нужен точный критерий: какая настройка лучше? Будем сравнивать выходы с известными метками **на той же карте**.

Для безопасного места желаемый выход равен +1, для опасного — −1. Пусть в безопасной точке нейрон сначала выдал −0,8, а потом −0,2. Оба раза выбран опасный класс: знак ещё отрицательный. Но второй выход ближе к правильному ответу. Простое «верно / неверно» пока не замечает улучшения.

1. Ошибка в одном месте

Обозначим выход нейрона буквой a, известную метку — y. Возьмём **функцию потерь**, которая используется в нашем Playground:

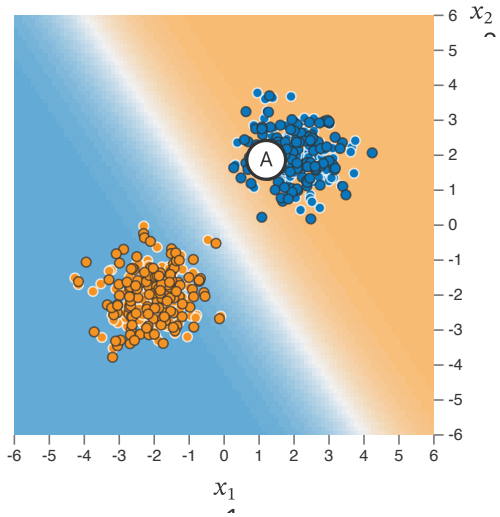
$$\ell = \frac{(a - y)^2}{2}.$$

Сначала находим расхождение, затем возводим его в квадрат и делим на два. Деление на два — часть выбранного правила; оно не меняет того, какой вариант лучше. Нулевая ошибка означает совпадение числового выхода с меткой. Чем дальше выход от метки, тем больше ошибка.

Для безопасной точки $y = 1$. При $a = -0{,}8$ ошибка равна 1,62, при $a = -0{,}2$ — 0,72. Правильный класс ещё не выбран, но функция потерь уже различает два состояния. При $a = 0{,}8$ ошибка становится 0,02: теперь и знак правильный, и выход близок к метке.

Попробуйте. Выберите точку A. Меняйте первый вес небольшими движениями и наблюдайте сразу три вещи: цвет фона у маркера, выход a и ошибку ℓ. Найдите два положения, где выбранный класс одинаков, а ошибка различается.

От ответа точки к ошибке модели



Два облака. Синий класс — безопасно, оранжевый — опасно.

Состояние опыта: 0 проходов. Скрытых слоёв нет.

Веса: -0,6; -0,4. Смещение: 0.

Для каждого пикселя считаем сумму: $s = w_1 x_1 + w_2 x_2 + b$
s = w1x1 + w2x2 + b

Точка A: (1,203; 1,866). Известный класс: безопасно. Прогноз: опасно.

$a \approx -0{,}899,\ y = 1,\ \ell = \frac{(a-y)^2}{2} \approx 1{,}804$a ≈ −0,899, y = 1, ℓ = 2(a−y)2 ≈ 1,804

Средняя ошибка на A, B, C: 1,836. На всех обучающих точках: 1,901.

A, B, C — три обучающие точки. Для выбранной точки показан расчёт ошибки; ниже — среднее по трём точкам и по всему обучающему набору.

Интерактивный эксперимент — в электронной версии урока.

Метка точки при движении ползунка не меняется. Именно она позволяет проверить прогноз. Веса общие для всех мест: улучшение ответа A может изменить ответы B и C. Посмотрите на них, прежде чем объявлять найденную настройку лучшей.

> Опасная точка имеет метку $y = -1$, а модель выдала $a = 1$. Чему равна ошибка $\ell = (a - y)^2/2$?

2. Ошибка на всей карте

Чтобы выбрать общие веса, мало смотреть на одну точку. Вычислим ошибку для каждого обучающего примера и возьмём **среднее**. Его обозначим L. Каждый пример

участвует с одинаковым весом: складываем отдельные ошибки и делим на число примеров.

Для трёх точек с ошибками 0,02, 0,18 и 0,08 получаем:

$$L = \frac{0{,}02 + 0{,}18 + 0{,}08}{3} \approx 0{,}093.$$

Средняя ошибка на A, B, C в тренажёре нужна для наглядного сравнения. Полное обучающее множество больше: число рядом с ним считается по всем его точкам. Эти два средних могут различаться. Нельзя выдавать три выбранных примера за результат на всей карте.

Попробуйте. Установите оба веса равными 1, смещение оставьте на 0. Запишите среднюю ошибку на трёх точках. Затем меняйте только смещение. Бывает ли, что выбранной точке становится лучше, а среднему — хуже? Если в вашем диапазоне такого не произошло, выберите другую точку или направление изменения.

Среднее удобно для настройки, но оно скрывает отдельные случаи. На большой карте модель может хорошо отвечать в большинстве мест и систематически ошибаться в небольшой области. Поэтому вместе с общим числом рассматриваем и расположение неверных ответов.

Нельзя просто сложить знаковые разности без квадрата: отклонения +1 и −1 погасились бы, хотя оба прогноза неточны. Квадрат устраняет это погашение и сильнее штрафует большие отклонения.

> Ошибки на двух точках карты равны 0,2 и 0,8. Какова средняя ошибка?

Заметка

Ошибка в этом уроке — половина квадрата разности выхода и метки. Это не процент неверных классов; у разных функций потерь разные шкалы.

3. Обучение и независимая проверка

В опыте половина точек предназначена для обучения, половина — для проверки. Алгоритм подбирает параметры по обучающей части. Проверочную часть используем, чтобы увидеть, как правило работает на примерах, которые не участвовали в этих обновлениях.

Если обучающая ошибка падает, а проверочная устойчиво растёт, модель может подстраиваться под особенности известных мест вместо полезного общего правила. Это признак возможного **переобучения**. Одна удачная точка или один удачный запуск не доказывают качество на новых данных.

В ручных опытах мы видели все точки и могли учитывать их при выборе весов. Поэтому это учебная демонстрация вычислений, а не независимое исследование качества.

В полноценном эксперименте итоговый проверочный набор не используют для постоянного подбора настроек.

Метка — правильный ответ из данных, а не пожелание модели. Нельзя уменьшать ошибку, перекрашивая неудобные точки под текущую границу. Можно исправить ошибку разметки после проверки, но тогда изменился набор данных, и сравнение нужно провести заново на одинаковых условиях.

> Обучающая ошибка падает, а ошибка на отложенных точках растёт. Какой вывод обоснован?
>
> ○ Отложенные ответы нужно подогнать под модель
>
> ○ Следует сообщать только обучающую ошибку
>
> ○ Возможно переобучение; улучшение на обучении не означает улучшения переноса
>
> ○ Модель обязательно лучше на любых данных

Заметка

Для каждого места входами остаются две координаты. Во время обучения меняются общие параметры модели, а известные координаты и метки сохраняются.

4. Функция потерь и доля верных классов

Счётчик «Верно» в ручном тренажёре проверяет совпадение классов. Он меняется, когда граница пересекает известную точку. Функция потерь учитывает величину выхода и может меняться ещё до пересечения. Поэтому высокая доля верных ответов и малая ошибка связаны, но это разные числа.

Например, в безопасном месте выходы 0,2 и 0,8 дают одинаковую метку. Ошибки при этом равны 0,32 и 0,02. Для обучения второй вариант лучше по выбранному правилу. В опасном месте с меткой -1 те же выходы дадут ошибки 0,72 и 1,62: более уверенное неверное решение получит больший штраф.

В прошлом сюжете мы переводили выход в вероятность безопасности: $p = (a + 1)/2$. При нашем квадратичном правиле и безопасной метке ошибка равна $2(1 - p)^2$. Значит, вероятностное описание и числовая ошибка относятся к одному ответу модели, а не к двум разным задачам.

Для классификации часто выбирают другую функцию — **кросс-энтропию**. Она тоже сильнее штрафует малую вероятность правильного класса. Например, 80% для известного безопасного места лучше, чем 20%. Это другой способ оценки: его числа нельзя напрямую сравнивать с числами нашей квадратичной функции. В тренажёрах этого раздела сохраняем исходное правило Playground, чтобы можно было проверить каждый расчёт.

Следующий шаг

Теперь мы знаем, какое число хотим уменьшать. В следующем сюжете поручим алгоритму менять веса на этой же карте и разберём направление и размер обновления.

Функция потерь сравнивает выход нейрона с известной меткой; на карте используем $\ell = (a - y)^2/2$.

Средняя ошибка объединяет примеры, а доля верных классов отвечает на другой вопрос.

Подбор параметров и независимая проверка требуют разных данных.

Домашняя работа

1. Найдите две настройки, при которых точка A классифицирована верно, но имеет разную ошибку. Проверьте B и C: улучшились ли все три ответа? Запишите веса, смещение и средние ошибки.

2. > Безопасная точка имеет метку 1, а выход нейрона равен 0. Чему равна половина квадрата разности?

Источники

- Оригинальный тренажёр взят из TensorFlow Playground (https://playground.tensorflow.org/). Авторы: Daniel Smilkov и Shan Carter.
- Функция потерь классификатора (https://d2l.ai/chapter_linear-classification/softmax-regression.html)

Повторить эту тему