

Оглавление

1	Введение	2
1.1	Машинное обучение	2
1.2	Виды задач	3
1.3	Выбор модели, переобучение	3
2	Классическое обучение с учителем	4
2.1	Линейные модели	4
3	Основные алгоритмы	5
3.1	Backpropagation	5
3.1.1	Вступление	5
3.1.2	Почему вообще нужно задумываться о том, как работает ВР?	5
3.1.3	Простейший пример	5
3.1.4	Слой $L - 1$	6
3.1.5	Слой $L - 2$	11

Глава 1

Введение

1.1 Машинное обучение

Машинное обучение – это наука, изучающая алгоритмы, автоматически улучшающиеся благодаря опыту.

Большинство решений задач можно представить в виде функции:

$$\text{Примеры (samples)} \rightarrow \text{Предсказания (targets)}.$$

Данная функция – **модель**, а набор примеров – **обучающая выборка (dataset)**.

Обучающая выборка = Объекты + Ответы.

Качество таких предсказаний измеряют **метриками** – функциями, которые показывают насколько полученные предсказания похожи на правильные ответы. Примером метрики является **среднее абсолютное отклонение**:

$$MAE(f, X, y) = L(f, X, y) = \frac{1}{N} \sum_{i=1}^N |f(x_i) - y_i|.$$

На сегодня достаточно знать два типа моделей – **градиентный бустинг на решающих деревьях** и **нейросетевые модели**.

Метрику, которую используют при поиске оптимальной модели – **функция потерь, лосс-функцией (loss)**.

1.2 Виды задач

Определенные выше задачи – **обучение с учителем (supervised learning)**, так как правильные ответы были даны заранее. Виды таких обучений:

- $\mathbb{Y} = \mathbb{R}$ или $\mathbb{Y} = \mathbb{R}^M$ – **регрессия**;
- $\mathbb{Y} = \{0, 1\}$ – **бинарная классификация**;
- $\mathbb{Y} = \{1, \dots, K\}$ – **многоклассовая (multiclass) классификация**;
- $\mathbb{Y} = \{0, 1\}^K$ – **многоклассовая классификация с пересекающимися классами (multilabel classification)**;
- \mathbb{Y} – конечное упорядоченное множество – **ранжирование**.

Имеется другой класс задач – **обучение без учителя (unsupervised learning)** – для которой известны только данные, а ответы отсутствуют. Одним из примеров является *кластеризация* – задача разделения объектов на группы, обладающие некоторыми свойствами.

1.3 Выбор модели, переобучение

Обобщающаяся способность модели – способность модели учить общие закономерности и давать адекватные предсказания. Выборку для этого делят на две части: **обучающая выборка** и **тестовая выборка (train и test)**.

Такой подход позволяет отделить модели, которые просто удачно подстроились к обучающим данным, от моделей, в которых произошла **генерализация (generalization)**, то есть от таких, которые на самом деле кое-что поняли о том, как устроены данные, и могут выдавать полезные предсказания для объектов, которые не видели.

Переобученный алгоритм – алгоритм, избыточно подстроившийся под данные.

Глава 2

Классическое обучение с учителем

2.1 Линейные модели

Глава 3

Основные алгоритмы

3.1 Backpropagation

3.1.1 Вступление

Алгоритмически описать данный процесс достаточно просто:

- Вычислить функцию потерь $C(w)$
- Вычислить градиент функции $C(w)$, учитывая все веса (w) и биасы (b) в нейронной сети
- Изменить веса (w) и биасы (b) пропорционально значению их градиента

Намного сложнее разобраться в том, как в действительности вычисляются эти градиенты.

3.1.2 Почему вообще нужно задумываться о том, как работает ВР?

3.1.3 Простейший пример

Рассмотрим следующую нейронную сеть:

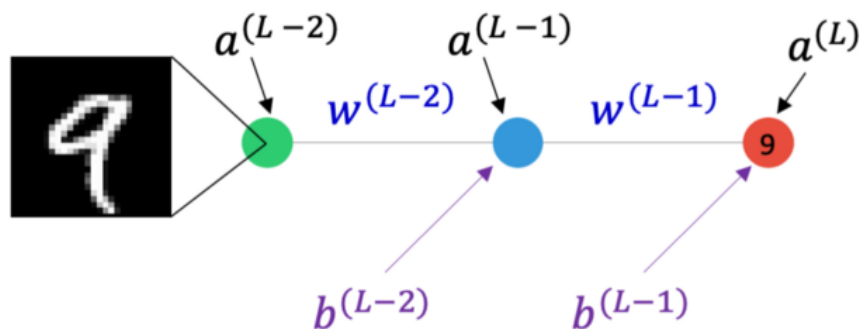


Рис. 3.1: простейшая нейронная сеть

На рисунке 3.1 изображена простейшая нейронная сеть, где зелёный, синий и красный нейрон отвечают за входной, скрытый и выходной нейрон (слой, если говорить в терминах более крупных нейронных сетей) соответственно. Обозначим функцию активации последнего нейрона как $a^{(L)}$, функцию активации предпоследнего слоя как $a^{(L-1)}$, а функцию активации первого слоя как $a^{(L-2)}$, где L - количество слоёв в нашей нейронной сети (в нашем случае $L = 3$). Аналогично определим веса и биасы между слоем $L - 1$ и L как $w^{(L-1)}$ и $b^{(L-1)}$.

Представим, что наша картинка цифры 9 состоит из одного пикселя (понятно, что это не возможно; сделаем это для упрощения вычислений). Теперь, пусть после того, как этот один пиксель прошёл через все три слоя нашей простейшей нейронной сети (стадия прямого распространения), на выходе мы получили число 0.68 (так как мы решаем задачу классификации, то выходные значения должны находиться в диапазоне от 0 до 1, что можно интерпретировать как вероятность; для этого можно в качестве функции активации последнего слоя (в нашем случае нейрона) использовать функцию Softmax (добавить сюда потом ссылку)). Но наше желаемое выходное значение — это 1 (т.е. мы хотим (после завершения обучения), чтобы наша сеть была уверена на 100%, что данная картинка (в нашем случае один пиксель) — это число 9). Теперь, пусть мы будем использовать средне квадратичную ошибку в качестве функции потерь. Т.е. мы получаем, что потеря C для одного тренировочного образца (т.е. цифры 9) равняется $(0.68 - 1)^2$

3.1.4 Слой $L - 1$

Рассчитаем градиент нашей функции потерь C учитывая вес, соединяющий нейроны в слое L со слоем $L - 1$. Чтобы посмотреть, как бы мы это сделали, развернём последний слой нашей простейшей нейронной сети.

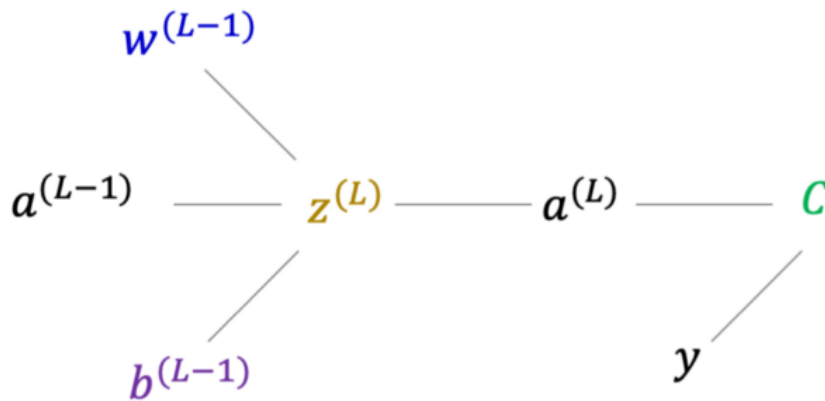


Рис. 3.2: Последний слой

Фигура 3.2 показывает, как каждый параметер последних двух слоёв влияет на потерю C . Чтобы получить взвешенную сумму $z^{(L)}$ для последнего слоя, мы умножаем активацию слоя (т.е. выход предпоследнего слоя, к которому была применена соответствующая функция активации для данного слоя) $L - 1$ на вес $w^{(L-1)}$, соединяющий два слоя, а затем мы прибавляем биас $b^{(L-1)}$. И наконец, мы пропускаем взвешенную сумму через нелинейную функцию (функцию активации) $\sigma(z^{(L)})$, чтобы посчитать $a^{(L)}$.

$$z^{(L)} = w^{(L-1)} a^{(L-1)} + b^{(L-1)}$$

$$a^{(L)} = \sigma(z^{(L)})$$

Рис. 3.3: Функция активации

Теперь мы хотим посмотреть, как сильно изменится C , если мы изменим

$w^{(L-1)}$. Т.е. мы хотим посчитать $\frac{\partial C}{\partial w^{(L-1)}}$.

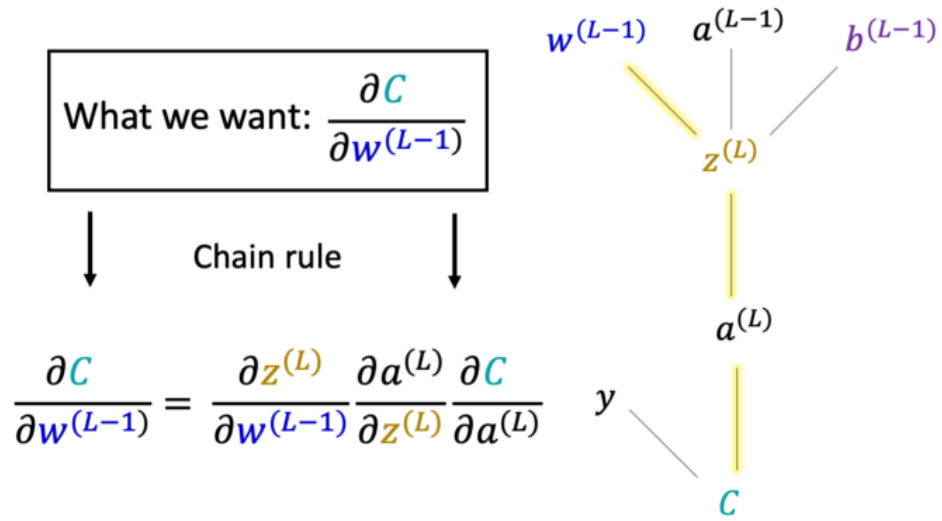


Рис. 3.4: Дифференцирование

Из картинки 3.4 можно понять, как вес $w^{(L-1)}$ влияет на C . Сначала вес $w^{(L-1)}$ даёт вклад во взвешенную сумму $z^{(L)}$, которая используется для того, чтобы посчитать потерю C . Найдём частную производную $\frac{\partial C}{\partial w^{(L-1)}}$ явно.

$$\left. \begin{aligned} \frac{\partial C}{\partial a^{(L)}} &= 2(a^{(L)} - y) \\ \frac{\partial a^{(L)}}{\partial z^{(L)}} &= \sigma'(z^{(L)}) \\ \frac{\partial z^{(L)}}{\partial w^{(L-1)}} &= a^{(L-1)} \end{aligned} \right\} \quad \boxed{\frac{\partial C}{\partial w^{(L-1)}} = \frac{\partial z^{(L)}}{\partial w^{(L-1)}} \frac{\partial a^{(L)}}{\partial z^{(L)}} \frac{\partial C}{\partial a^{(L)}} = a^{(L-1)} \sigma'(z^{(L)}) 2(a^{(L)} - y)}$$

Рис. 3.5: Находим частную производную явно

То, как получены последние два выражения в левой части - очевидно. Рассмотрим первое. Так как для подсчёта потери мы используем средне квадратичную ошибку (MSE, которая считается как $(a^{(L)} - y)^2$), то если взять от этого производную по $\partial a^{(L)}$, то мы как раз получим $2(a^{(L)} - y)$.

Заметим, что мы не нашли производную от нашей функции активации (которую мы обозначили как σ) явно. Если бы мы использовали Sigmoid в качестве нашей функции активации:

$$\frac{1}{1 + e^{-z}}$$

Тогда её производная равнялась бы:

$$\frac{e^{-z}}{(1 + e^{-z})^2}$$

Замечание: вот почему важно использовать в качестве функций активации всюду дифференцируемые функции, в противном случае мы не сможем рассчитать наши градиенты, однако функцию ReLu всё же можно использовать (добавить потом сюда ссылку).

Также мы хотим знать, как изменится значение нашей функции потерь, если мы изменим биас $b^{(L-1)}$.

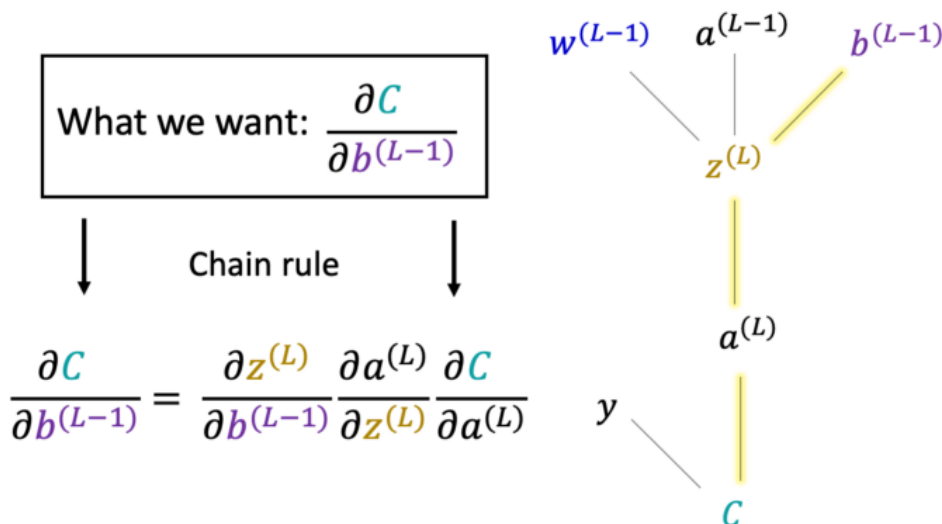


Рис. 3.6: Считаем производную для биаса

Как можно заметить, поменялся только первый множитель. Аналогично, найдём явную производную.

$$\left. \begin{aligned} \frac{\partial \mathcal{C}}{\partial a^{(L)}} &= 2(a^{(L)} - y) \\ \frac{\partial a^{(L)}}{\partial z^{(L)}} &= \sigma'(z^{(L)}) \\ \frac{\partial z^{(L)}}{\partial b^{(L-1)}} &= 1 \end{aligned} \right\} \boxed{\frac{\partial \mathcal{C}}{\partial b^{(L-1)}} = \frac{\partial z^{(L)}}{\partial b^{(L-1)}} \frac{\partial a^{(L)}}{\partial z^{(L)}} \frac{\partial \mathcal{C}}{\partial a^{(L)}} = \sigma'(z^{(L)}) 2(a^{(L)} - y)}$$

Рис. 3.7: Считаем производную для биаса

Отлично, теперь нейронная сеть может использовать полученные выражения для обновления весов и биасов по следующему принципу.

$$\begin{aligned} w^{(L-1)}(t+1) &= w^{(L-1)}(t) - \eta \frac{\partial \mathcal{C}}{\partial w^{(L-1)}} \\ b^{(L-1)}(t+1) &= b^{(L-1)}(t) - \eta \frac{\partial \mathcal{C}}{\partial b^{(L-1)}} \end{aligned}$$

Рис. 3.8: Обновление весов

На картинке 3.8 η — это learning rate.

Несмотря на то, что нейронная сеть не имеет прямого контроля над $a^{(L-1)}$, скоро будет показано, что что нам будет это нужно, чтобы дальше двигаться по сети и обновлять параметры, поэтому проделаем предыдущие шаги, только для $a^{(L-1)}$.

Получаем:

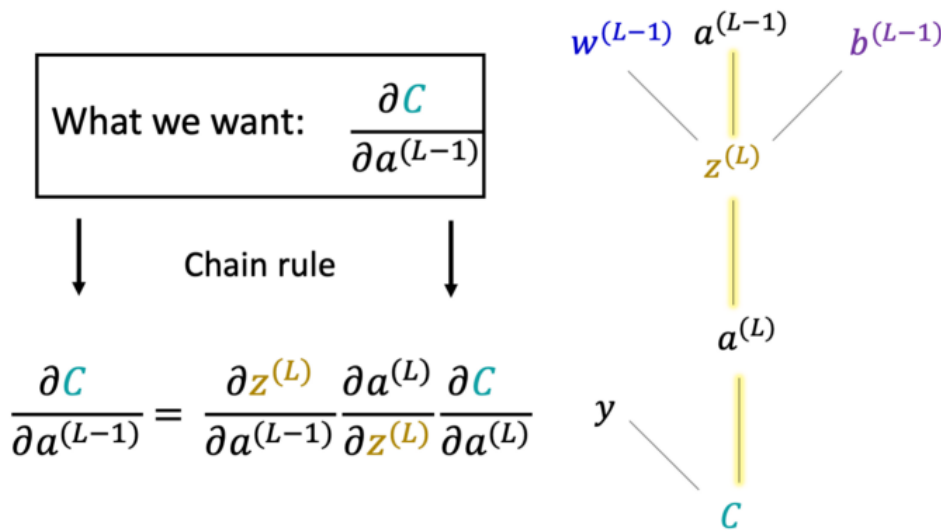


Рис. 3.9: Считаем градиент относительно функции активации предпоследнего слоя

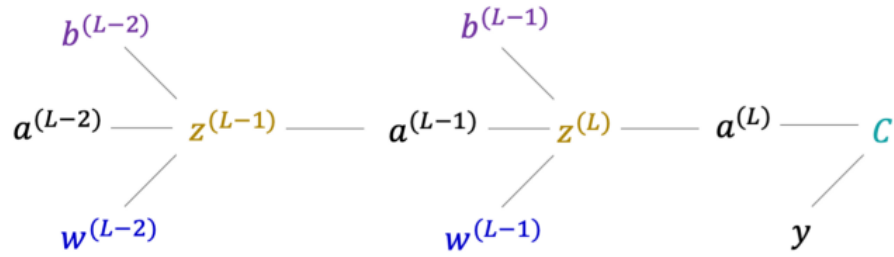
Находим производную явно:

$$\left. \begin{aligned} \frac{\partial C}{\partial a^{(L)}} &= 2(a^{(L)} - y) \\ \frac{\partial a^{(L)}}{\partial z^{(L)}} &= \sigma'(z^{(L)}) \\ \frac{\partial z^{(L)}}{\partial a^{(L-1)}} &= w^{(L-1)} \end{aligned} \right\} \quad \boxed{\frac{\partial C}{\partial a^{(L-1)}} = \frac{\partial z^{(L)}}{\partial a^{(L-1)}} \frac{\partial a^{(L)}}{\partial z^{(L)}} \frac{\partial C}{\partial a^{(L)}} = w^{(L-1)} \sigma'(z^{(L)}) 2(a^{(L)} - y)}$$

Рис. 3.10: Находим производную явно

3.1.5 Слой $L - 2$

Теперь давайте посмотрим, как будет меняться значение функции потерь C , если изменять параметры в слое $L - 2$. Как и раньше, нарисуем граф, чтобы наглядно продемонстрировать, как $w^{(L-2)}$ и $b^{(L-2)}$ косвенно влияют на значение потери C .

Рис. 3.11: Слои $L-1$ и $L-2$

Аналогично, используя цепное правило, посмотрим, как изменение веса $w^{(L-2)}$ влияет на значение функции потерь C .

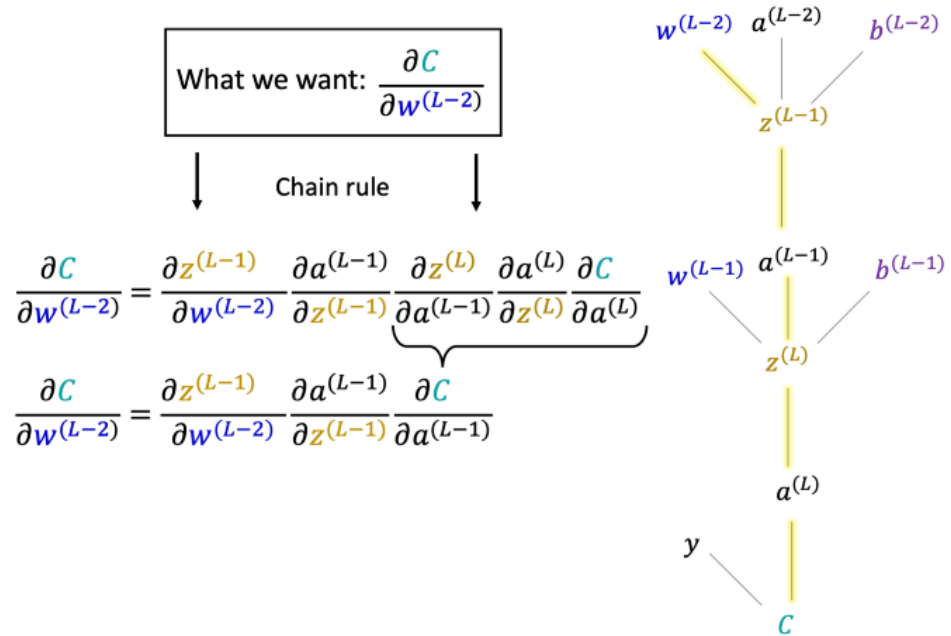


Рис. 3.12: Частная производная для веса первого слоя

Теперь найдём производную явно:

Аналогичным образом можно посчитать градиенты для всех весов и биасов. Этот алгоритм называется **обратным распространением ошибки**.

$$\left. \begin{aligned} \frac{\partial a^{(L-1)}}{\partial z^{(L-1)}} &= \sigma'(z^{(L-1)}) \\ \frac{\partial z^{(L-1)}}{\partial w^{(L-2)}} &= a^{(L-2)} \end{aligned} \right\} \boxed{\frac{\partial C}{\partial w^{(L-2)}} = \frac{\partial z^{(L-1)}}{\partial w^{(L-2)}} \frac{\partial a^{(L-1)}}{\partial z^{(L-1)}} \frac{\partial C}{\partial a^{(L-1)}} = a^{(L-2)} \sigma'(z^{(L-1)}) \frac{\partial C}{\partial a^{(L-1)}}}$$

Рис. 3.13: Производная в явном виде