Deep Learning. Улучшение нейронных сетей

Урок 2.5

Егор Конягин

Содержание

Повторение

Задача оптимизации. Повторение

Основными проблемами, с которыми сталкиваются алгоритмы оптимизации при обучении нейронных сетей - это

- седловые точки;
- немасштабированные данные.

Для решения первой проблемы применяют алгоритмы с использованием скользящим средним градиентов предыдущих шагов (Momentum GD, AdaM), для решения второй: используют адаптивные алгоритмы (RMSProp, AdaM). Заметим, что AdaM является комбинацией адаптивных алгоритмов и алгоритмов со скользящим средним.

Важно! Для подсчета градиентов (не обновления параметров) используется SGD или mini-batch GD.

Проблема переобучения. Регуляризация

Проблема переобучения

Переобучение - это чрезмерное постраивание под данные обучающей выборки, при котором ухудшается качество работы модели.

Рис. 1: Проблема переобучения

Переобучение или недообучение

Важно понимать, насколько хорошо данная модель обучена в текущей момент. Возможны три состояния:

1. оптимальное обучение;
2. недообучение (underfitting);
3. переобучение (overfitting).

Underfitting заключается в следующем: мы имеем похожее качество на train и на test - выборке, однако это качество нас не устраивает.

Overfitting характеризуется тем, что качество на train-выборке очень высокое (вплоть до 100%), в то время как на test-выборке оно может быть существенно ниже

Кривая обучения

Кривая обучения - это график зависимости функции потерь от номера итерации обучения.

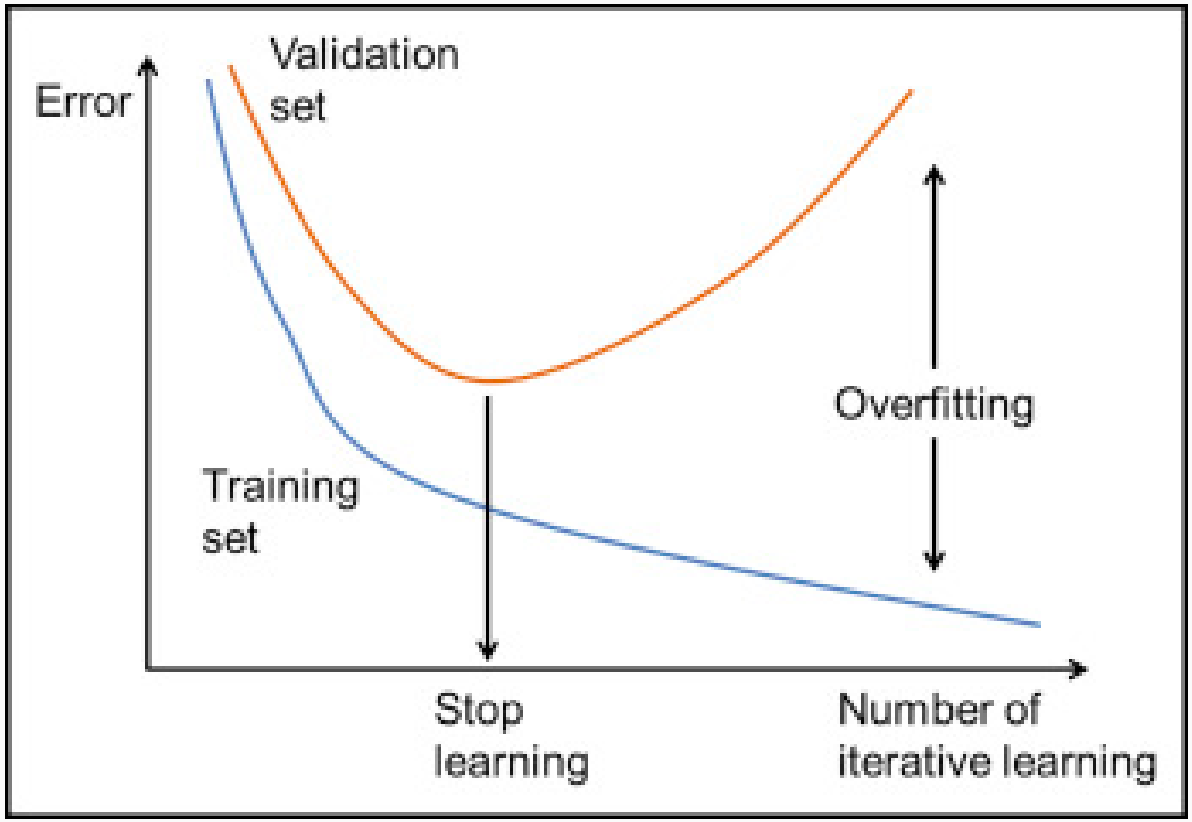

Рис. 2: Кривая обучения

Методы регуляризации

Регуляризация. обзор

Регуляризация - это наложение неких ограничений на параметры w, b с тем, чтобы избежать переобучение. Сегодня мы рассмотрим следующие алгоритмы регуляризации:

1. L2 - регуляризация (ridge regularization);
2. L1 - регуляризация (Tikhonov, lasso regularization);
3. Dropout;

Weight decay

При проведенной нормализации входных данных (т.е. приведение их к виду $[-1, 1]$) одним из симптомов переобучения являются большие по абсолютному значению веса. Идея метода weight decay: увеличивать функцию потерь, если значения весов велики.

L2-регуляризация:

$$L(y, \hat{y})^{reg} = \sum \mathcal{L}(y^{(i)}, \hat{y}^{(i)}) + \frac{\lambda}{2} \sum_{l} ||W^{[l]}||_F^2 \tag{1}$$

L1-регуляризация (weight decay):

$$L(y, \hat{y})^{reg} = \sum \mathcal{L}(y^{(i)}, \hat{y}^{(i)}) + \lambda \sum_{l} \sum_{i,j} ||W_i^{[l]j}||_F \tag{2}$$

Теперь градиент функции ошибки по параметрам изменится:

$$\frac{\partial L}{\partial W^{[l]}} + \frac{\lambda}{m} W^{[l]}. \tag{3}$$

Dropout

Dropout - это техника случайного отключения заданного количества нейронов с тем, чтобы увеличить стабильность работы нейросети.

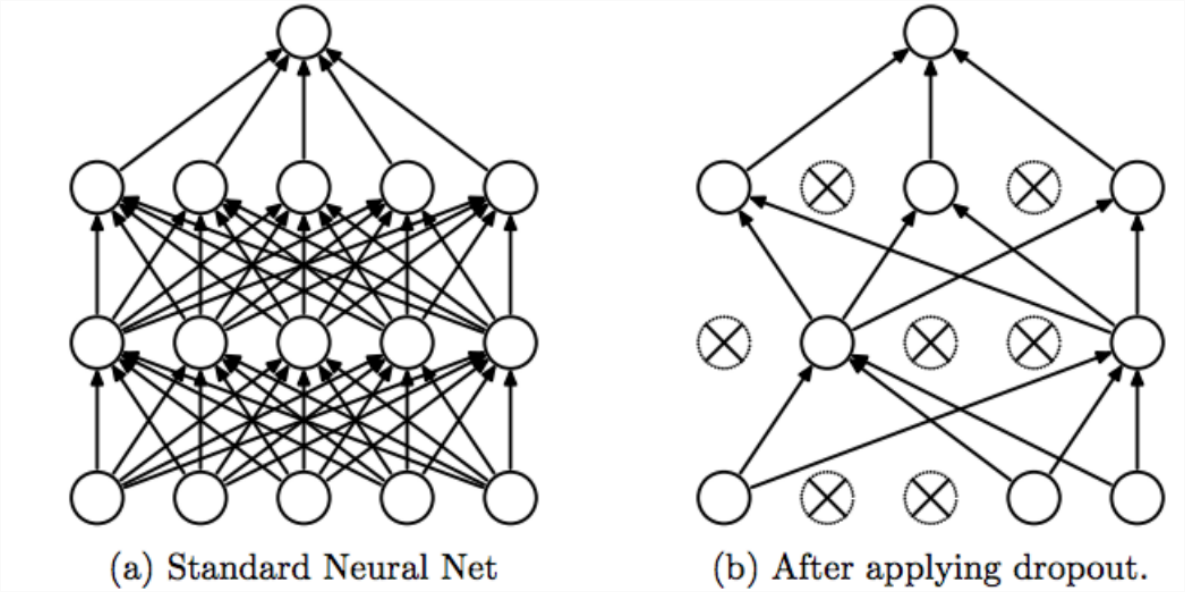

Рис. 3: Применение dropout. Источник: Stanford University

Dropout. Forward propagation

Рассмотрим, как же можно в формульном виде записать Dropout. Раньше forward propagation описывался так:

$$a^{[L]} = \sigma(W^{[L]}a^{[L-1]} + b^{[L]}) \tag{4}$$

Теперь надо выход каждого слоя поэлементно умножать на вектор, состоящий только из 1 и 0:

$$a^{[L]*} = a^{[L]} * D^{[L]}. \tag{5}$$

В итоге, в полученном выходе какие-то координаты будут равны нулю, а какие-то останутся без изменения.

Dropout характеризуется одним параметром - вероятностью.

Dropout. Backward propagation

Здесь также необходимо внести изменения: при подсчете $dA^{[L]}$ необходимо в конце этот вектор поэлементно умножить на маску $D^{[L]}$. Для этого надо не забыть сохранить ее значение, полученное в ходе forward propagation.

Помимо этого, вектор градиента надо разделить на параметр dropout-a, чтобы среднее значение выхода этого слоя оставалось неизменным.

ВАЖНО!!! Dropout применятся ТОЛЬКО на стадии обучения модели. Если модель уже обучена, dropout применять не нужно.

Проблема переобучения

Переобучение - это чрезмерное подстраивание модели под обучающие данные, при котором ухудшается качество работы модели.

Рис. 4: Проблема переобучения

Переобучение

Переобучение можно по-разному трактовать. Одна из возможных трактовок - это попытка нейросети ”запомнить”правильные ответы, а не пытаться подстроиться под общую закономерность.

В результате, на тренировочных данных модель запоминает правильные ответы, и качество стремится к 100%. На тестовых данных модель, игнорируя реальные физические закономерности, выдает ответы, близкие к случайным, или просто с очень большой ошибкой.

Кривая обучения

Кривая обучения - это график зависимости функции потерь от номера итерации обучения.

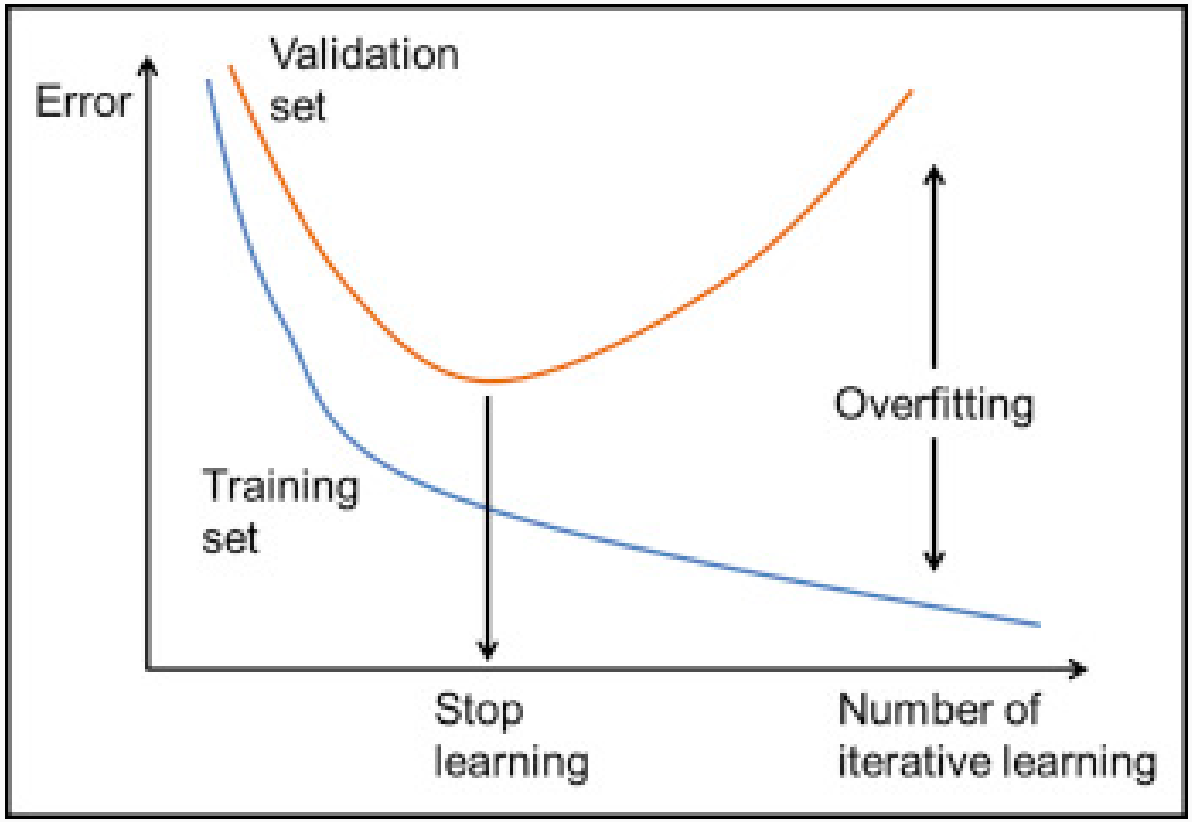

Рис. 5: Кривая обучения

Dropout

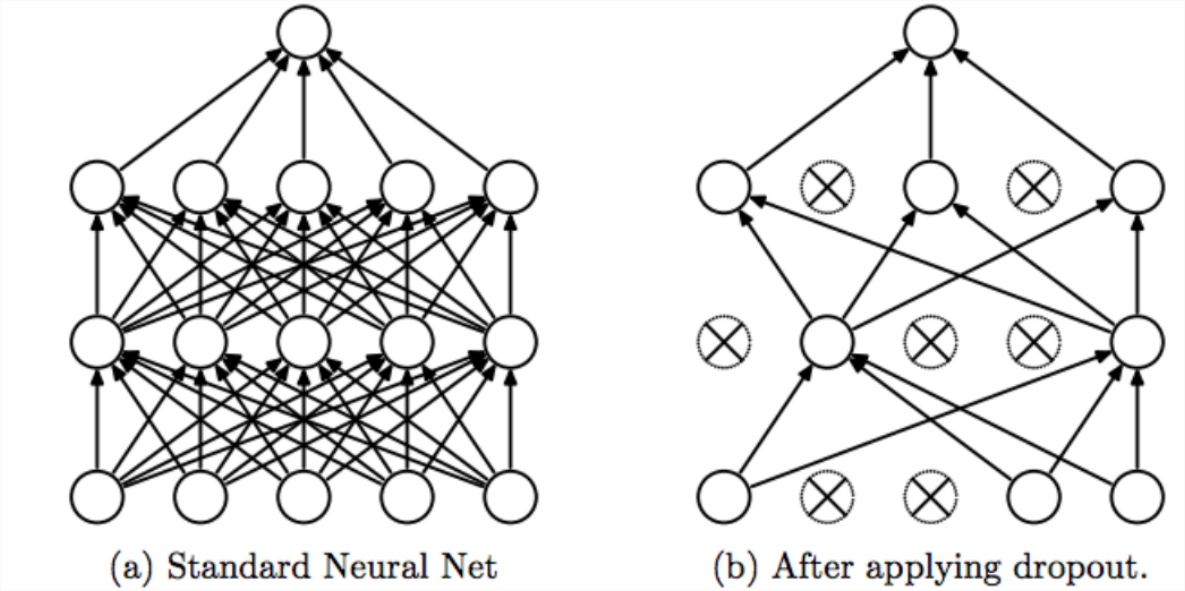

Рис. 6: Применение dropout. Источник: Stanford University

Борьба с переобучением. Продолжение

Борьба с переобучением

Как мы ранее выяснили, борьба с переобучением осуществляется с помощью регуляризации, то есть в наложении дополнительных ограничений на функцию потерь.

Вопрос Может ли увеличение объема датасета побороться с переобучением?

А с недообучением?

Борьба с переобучением - II

Увеличение объема датасета иногда помогает бороться с переобучением.

Проблема не решается, если обучение алгоритма проходит на хороших данных, а работать нейросети приходится на худших данных (и наоборот). Это вызывает ”перекосы”в распределении данных.

Затухание/взрыв градиентов

Затухание/взрыв градиентов

Рассмотрим глубокую нейросеть (L » 1):

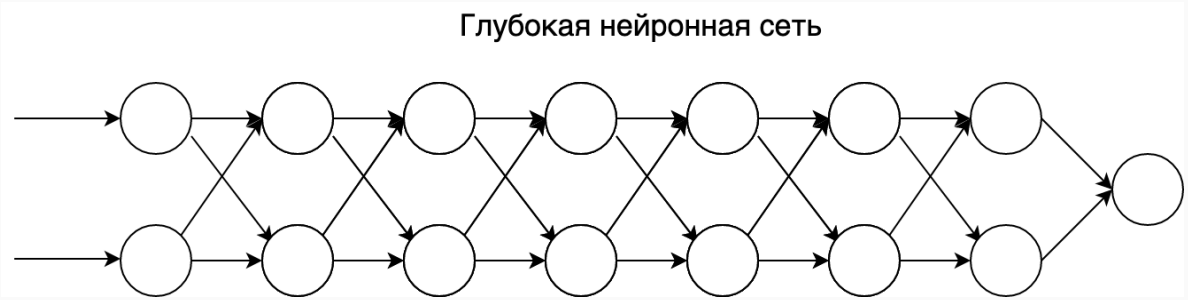

Рис. 7: Глубокая нейронная сеть

Положим, что все веса при инициализации являются положительными и меньше единицы. Выберем функцию активацию такую, что она удовлетворяет следующему разложению вблизи малого значения аргумента: $\sigma(x) = x + o(x)$.

Затухание/взрыв градиентов

Если это так, то forward propagation можно представить следующим способом:

$$\hat{y} = w^{[L]} \cdot w^{[L-1]} \dots w^{[1]} \cdot x. \tag{6}$$

По предположению веса инициализированы положительными малыми числами. Тогда

$$|y| \leq ||w||^{L}_{max} \cdot |x|_{max} \tag{7}$$

Поскольку веса малы, то $||w|| \to 0|_{L \to +\infty}$. Таким образом, у будет очень малым значением, обучение будет происходить крайне долго. Если веса, наоборот, будут больше единицы, то у будет экспоненциально возрастать как функция от L.

Затухание/взрыв градиентов

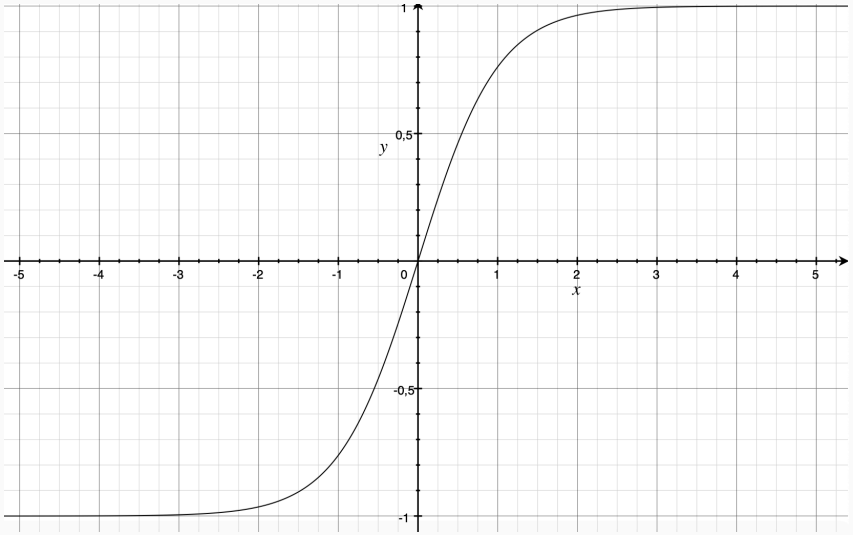

Рис. 8: График гиперболического тангенса

Как видно,

$$\tanh(x)|_{x>>1} \to 1; \qquad \frac{d\tanh(x)}{dx}|_{x>>1} \to 0. \tag{8}$$

Таким образом, градиент такой функции активации в случае большого значения аргумента будет равен нулю. Обучение также будет происходить крайне долго.

Борьба с взрывом и затуханием градиентов. Инициализация весов

Инициализировать веса можно следующим образом (He initialization):

$$w^{[n_l]} \sim \mathcal{N}(0,1) \cdot \sqrt{\frac{1}{n_{l-1}}} \quad - sigmoid; \tag{9}$$

$$w^{[n_l]} \sim \mathcal{N}(0,1) \cdot \sqrt{\frac{2}{n_{l-1}}} \quad - ReLU. \tag{10}$$

Модификация

$$w^{[n_l]} \sim \mathcal{U}niform(-1,1) \cdot \sqrt{\frac{2}{n_{l-1}+n_l}} \tag{11}$$

называется Xavier Initialization.

Batch normalization

Другой способ борьбы с взрывом и затуханием градиентов - это batch normalization (batch norm). Идея состоит в следующем: отнормировать данные на каждом слое:

$$\mu^{[l]} = \frac{1}{m}\sum_{i=1}^{m} z^{[l](i)} \tag{12}$$

$$\sigma^{[l]} = \frac{1}{m}\sum_{i=1}^{m} (z^{[l](i)} - \mu^{[l]}) \tag{13}$$

$$z^{[l]}_{norm} = \frac{z^{[l]} - \mu^{[l]}}{\sqrt{\sigma^2 + \varepsilon}} \tag{14}$$

Можно еще сильнее модифицировать z:

$$\hat{z}^{[l]} = \gamma^{[l]} z^{[l]}_{norm} + \beta^{[l]} \tag{15}$$

Summary

Сегодня мы

- поговорили про переобучение;
- познакомились с проблемой затухания градиентов;
- научились бороться с этой проблемой двумя методами
  1. правильной инициализацией весов;
  2. с помощью метода batch norm;