Deep Learning. Оптимизация и настройка нейронных сетей - II

Урок 5

Егор Конягин

МФТИ & АО ”ЦОСиВТ”

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

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

Рис. 1: Проблема переобучения

Кривая обучения

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

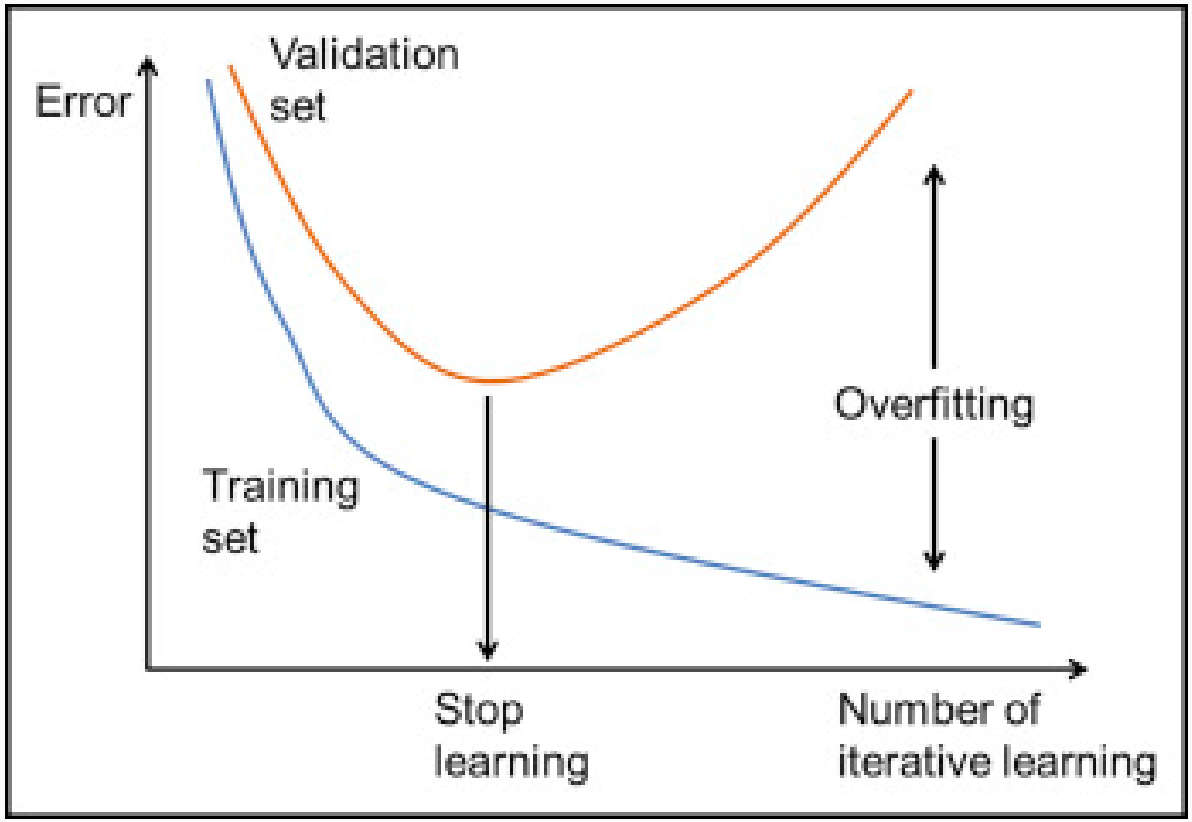


Рис. 2: Кривая обучения

Dropout

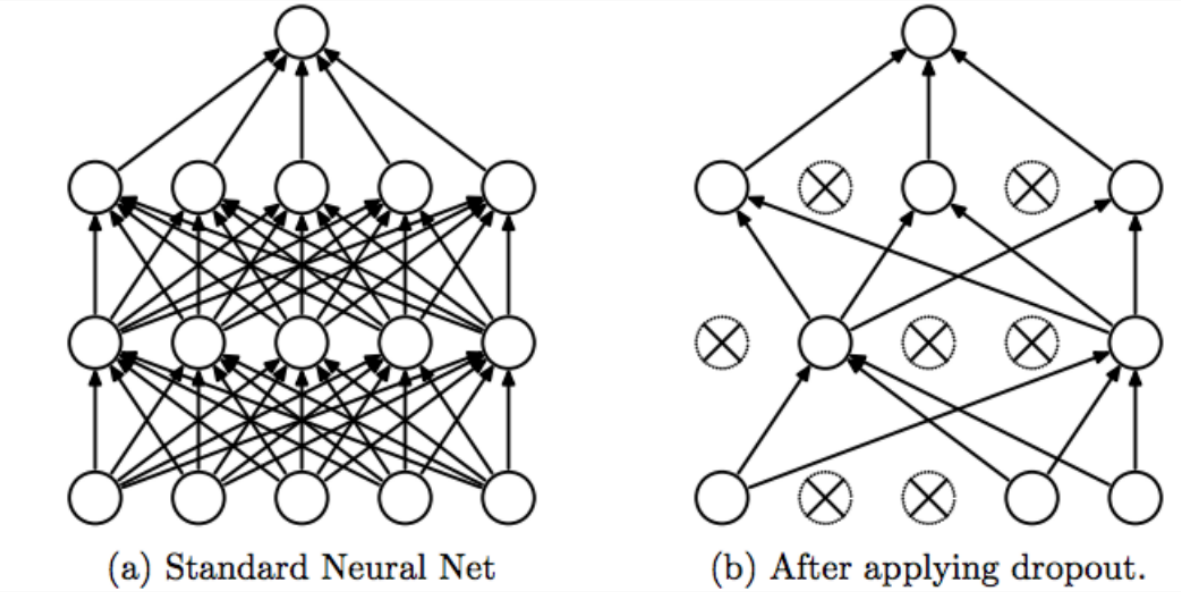


Рис. 3: Применение dropout. Источник: Stanford University

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 действительно помогает бороться с переобучением, поскольку с увеличением его объема распределение данных в обоих датасетах становятся более похожими.

Проблема иногда случается, что обучение алгоритма проходит на хороших данных, а работать нейросети приходится на худших данных. Это вызывает проблемы в качестве, которые сложно непосредственно отнести к переобучению.

Затухание/взрыв градиентов

Затухание/взрыв градиентов

Рассмотрим глубокую нейросеть (L » 1):

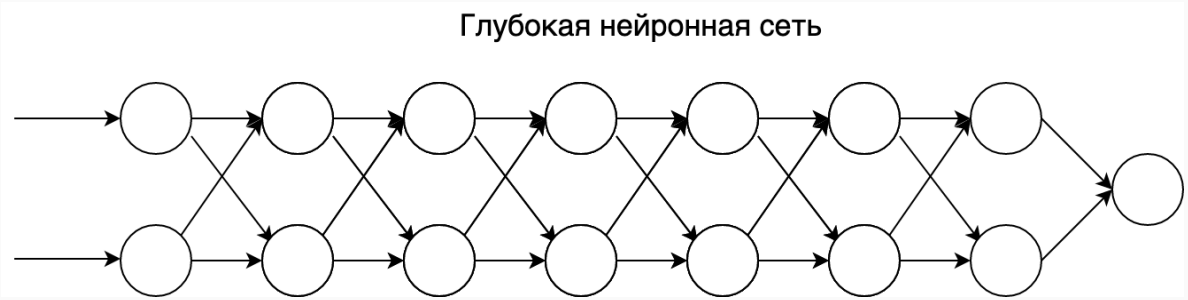


Рис. 4: Глубокая нейронная сеть

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:

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

По предположению веса инициализированы положительными малыми числами. Тогда

$$|y| \leq ||w||^{L}_{max} \cdot |x|_{max} \tag{2}$$

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

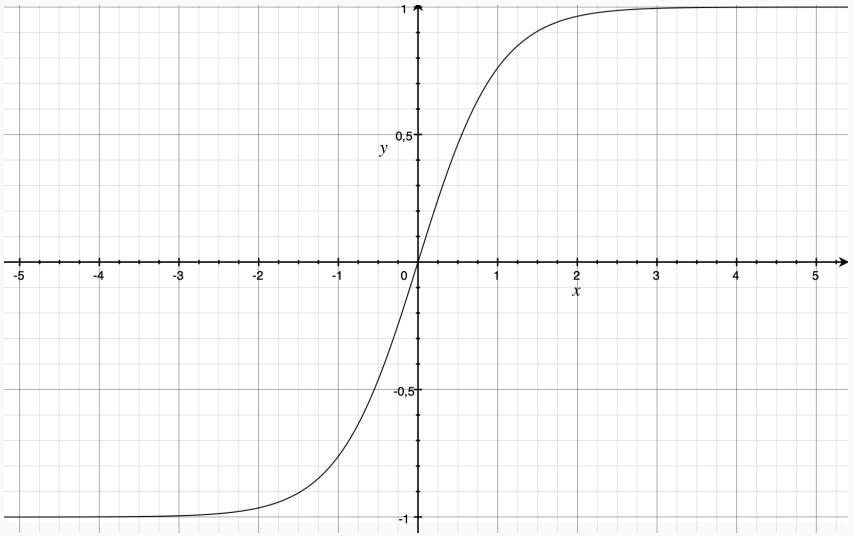


Рис. 5: График гиперболического тангенса

Как видно,

$$\tanh(x)|_{x>>1} \to 1; \qquad \frac{d\tanh(x)}{dx}|_{x>>1} \to 0. \tag{3}$$

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):

$$w^{[n_l]} \sim \mathcal{N}(0,1) \cdot \sqrt{\frac{1}{n_{l-1}}} \quad - sigmoid; \tag{4}$$

$$w^{[n_l]} \sim \mathcal{N}(0,1) \cdot \sqrt{\frac{2}{n_{l-1}}} \quad - ReLU. \tag{5}$$

Модификация

$$w^{[n_l]} \sim \mathcal{N}(0,1) \cdot \sqrt{\frac{2}{n_{l-1} + n_l}} \tag{6}$$

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:

$$\mu^{[l]} = \frac{1}{m}\sum_{i=1}^{m} z^{[l](i)} \tag{7}$$

$$\sigma^{[l]} = \frac{1}{m}\sum_{i=1}^{m} (z^{[l](i)} - \mu^{[l]}) \tag{8}$$

$$z^{[l]}_{norm} = \frac{z^{[l]} - \mu^{[l]}}{\sqrt{\sigma^2 + \varepsilon}}. \tag{9}$$

Можно еще сильнее модифицировать z:

$$\hat{z}^{[l]} = \gamma^{[l]} z^{[l]}_{norm} + \beta^{[l]} \tag{10}$$

Метод главных компонент (principal component analysis)

Метод главных компонент (principal component analysis)

Метод главных компонент - это метод понижения размерности признакового пространства. PCA непосредственно не относится к глубокому обучению, однако его можно применять и в задачах глубокого обучения в рамках этапа предобработки данных.

Сингулярное разложение:

$$\mathbf{M} = U\Sigma V^*, \tag{11}$$

где U—ортогональная (унитарная) матрица, Σ—диагональная матрица, причем кол-во ненулевых элементов на диагонали равно рангу матрицы $\mathbf{M}$, а V—тоже ортогональная (унитарная) матрица (звездочка означает эрмитово сопряжение).

PCA. Реализация

1. Нормализация:

$$X^* = \frac{X - \mu}{\sigma} \tag{12}$$

2. поиск первой главной компоненты (поиск направления, вдоль которого дисперсия максимальна):

$$\frac{1}{m}\sum_{i=1}^{m}(ux^{(i)})^2 = \frac{1}{m}\sum_{i=1}^{m}(u^T x^{(i)T} x^{(i)} u) = u^T(X^T X)u; \tag{13}$$

3. Поиск остальных главных компонент:

$$\hat{X}_k = X - \sum_{s=1}^{k-1} X u_{(s)} u_{(s)}^T; \tag{14}$$

$$u_k = \underset{u}{\text{argmax}}(u^T \hat{X}_k^T \hat{X}_k u). \tag{15}$$

PCA. Пример

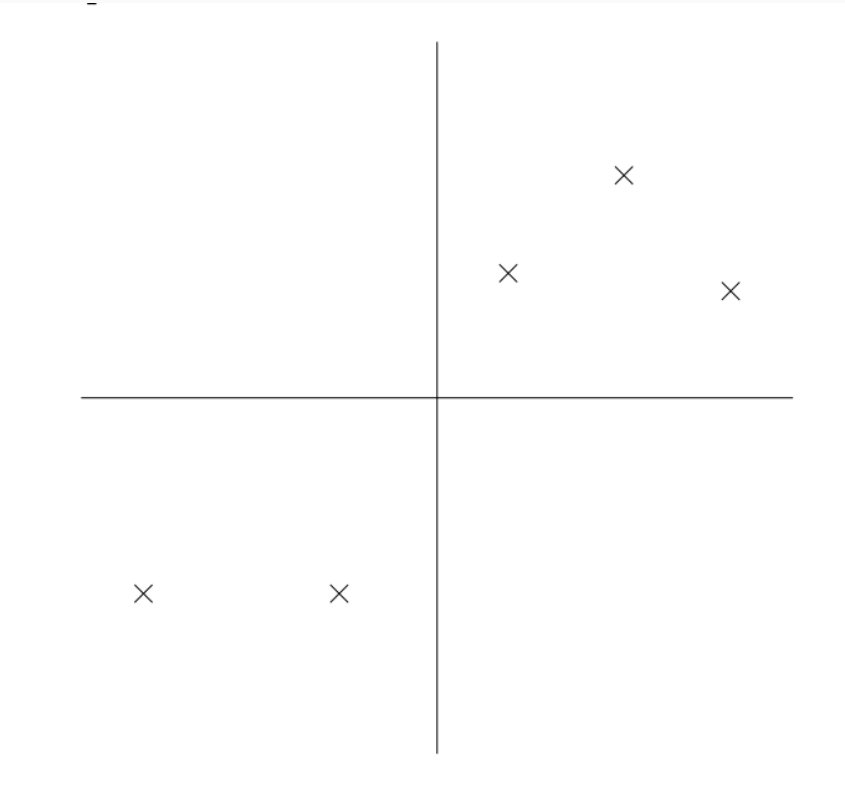
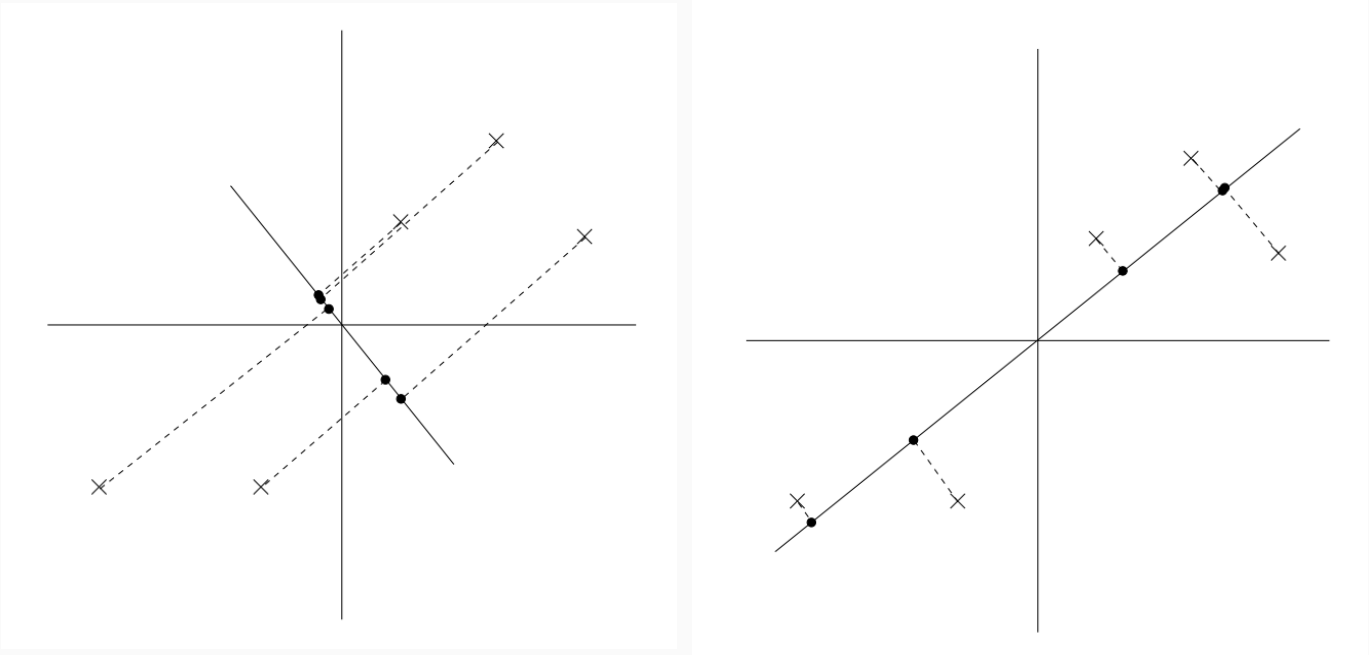


Рис. 6: Двумерный датасет

PCA. Пример



a) b)

Рис. 7: Неверное и верное главное направление

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;
- обсудили метод PCA для предобратботки данных.