Другие методы обучения нейросетей

Вручтель Серафима. Группа М05-895б.

6 декабря 2018 г.

1 Введение

Самым популярным методом обучения нейронных сетей является Back Propogation. Основная его «фишка» — это chain rule, благодаря которому и происходит и обратное распространение ошибки.

Но помимо Back Propogation существуют также и другие методы обучения нейросетей. В этой статье мы попробуем разобраться, какие ещё есть способы обучения искусственных нейросетей, и в чём всё-таки их превзошёл Back Propogation.

2 Обзор методов

2.1 Метод Хебба

Метод Хебба был создан в 1949 году на основе физиологических и психологических исследований и стал первым способом обучения нейронных сетей.

К 1949 году Хебб сформулировал «универсальный нейрофизиологический постулат», суть которого заключается в следующем: «Если нейрон А находится достаточно близко к нейрону В и часто принимает участие в его возбуждении, то можно наблюдать процесс роста или метаболических изменений в одном или в обоих нейронах, ведущий к увеличению эффективности А, как одного из нейронов, возбуждающих В».

Обычно для обучения нейросетей, основываясь на методе Хебба, используют следующие правила корректировки (предполагается, что каждый нейрон на выходе выдаёт либо логический ноль, либо логическую единицу):

Первое правило: Если сигнал на выходе из нейрона неверен и при этом равен нулю, **увеличить** веса тех входов, на которые была подана единица.

Второе правило: Если сигнал на выходе из нейрона неверен и при этом равен единице, **уменьшить** веса тех входов, на которые была подана единица.

Правила последовательно применяются для всех элементов из обучающей выборки.

2.2 Генетические алгоритмы

Алгоритмы этого вида являются стохастическими и, можно сказать, используют теорию эволюции.

Пусть есть множество алгоритмов, которое мы назовём популяцией. Для каждой «особи» p в популяции можно вычислить значение некоторой функции ошибки $E(p)$. «Особи» могут размножаться, причём вероятность их размножения зависит от значения $E(p)$ — чем меньше ошибка, тем выше вероятность размножения. Например, если ошибка некоторого алгоритма достаточно велика, соответствующая «особь» с большой вероятностью может погибнуть не произведя на свет «потомства» (Размножаться особи могут как скрещиванием, так и делением, в зависимости от выбранного подхода).

Потомство может мутировать. Это осуществляется с помощью случайных модификаций весов нейросети. Наиболее удачные мутации, дающие наименьшую ошибку, будут сохраняться, а наименее удачные - уничтожаться.

В итоге может быть получен алгоритм с минимальным значением ошибки E.

2.3 Ньютоновский метод

Метод Ньютона похож на метод обратного распространения ошибки с некоторым существенным изменением. Его суть заключается в поиске лучшего направления обучения с использованием вторых производных функции потерь (гессиана).

Пусть w_i — вектор весов нейросети на шаге обучения i. Тогда шаг метода Ньютона осуществляется следующим образом:

$$w_{i+1} = w_i - (H_i^{-1} g_i) \cdot \eta_i$$

где $H_i = H(f(w_i))$ — матрица Гессе, $g_i = \nabla f(w_i)$ — производная функции потерь на i-й итерации, η_i — learning rate.

Во многих случаях метод Ньютона сходится быстро, но требует больших затрат из-за вычисления гессиана и обратной матрицы. Пытаясь избежать вычисления матрицы Гессе, изобретают различные квазиньютоновские методы.

Квазиньютоновский метод

Суть этого метода заключается в приближённом вычислении обратной матрицы Гессе на каждой итерации работы алгоритма, причём для вычисления приближённой обратной матрицы необходимы только первые производные функции потерь. Гессиан аппроксимируется некоторой матрицей G, и шаг метода выглядит следующим образом:

$$w_{i+1} = w_i - (G_i \cdot g_i) \cdot \eta_i$$

Приближённую обратную матрицу Гессе G можно вычислять разными способами. Два самых популярных варианта: формула Давиона-Флетчера-Пауэлла и формула Бройдена-Флетчера-Гольдфарба-Шанно.

2.4 Моменты

Моменты применяются совместно с градиентным спуском. Они позволяют придать векторам градиентов «ускорение» в нужном направлении, благодаря чему метод сходится быстрее. На самом деле, именно этот способ чаще всего и используется для оптимизации современных нейросетей (см., например, Nesterov Momentum, SGD Momentum и пр.).

Пусть у нас есть зашумлённая последовательность чисел S_i. Попытаемся понять, как выглядела исходная последовательность без шума. Для этого построим новую последовательность чисел V_i:

$$V_i = \beta V_{i-1} + (1 - \beta)S_i,\ \beta \in [0, \dots, 1]$$

Здесь β выступает в качестве гиперпараметра.

Так мы получим, что i-е значение последовательности V_i зависит от всех предыдущих значений оригинальной последовательности. При этом наиболее «недавним» значениям S_i придаётся больший вес.

Момент получается применением записанной выше формулы с использованием шагов градиента функции потерь в качестве исходной последовательности. Таким образом момент позволяет регулировать направление и скорость градиентного спуска, придавая ему некоторую инерцию. Например, SGD momentum выглядит следующим образом:

$$V_i = \beta V_{i-1} + (1 - \beta)\nabla L_W(W, X, y)$$

$$W = W - \alpha V_i$$

Здесь L — функция потерь, а α — learning rate.

3 Заключение

Как можно видеть, существует много различных способов обучения нейронных сетей. Но чем же всё-таки оказался так хорош Back Propogation?

В отличие от остальных алгоритмов, таких как, например, генетический алгоритм или ньютоновский метод, Back Propogation требует меньше ресурсов для вычислений. Также он имеет существенное преимущество в скорости по сравнению с, например, методом Хебба или же, снова, генетическим алгоритмом. С применением метода обратного распространения ошибки нейросеть обучается намного быстрее.

Ну и, как говорится, we user back propogation because
(1) we don't have anything better than back propogation,
(2) it is working.