АЛГОРИТМЫ СТОХАСТИЧЕСКОГО ГРАДИЕНТНОГО СПУСКА ОБУЧЕНИЯ И ТРЕНИРОВКИ НЕЙРОННЫХ СЕТЕЙ

Белошедов Е. С., Гуринович А. Б., Гаруля Д. В., Архипенко Я. С.
Кафедра информационных технологий автоматизированных систем, Белорусский государственный университет информатики и радиоэлектроники
Минск, Республика Беларусь
E-mail: 2595118@mail.ru

В работе исследуется возможность повышения эффективности обучения нейронных сетей за счёт алгоритма оптимизации под названием стохастический градиентный спуск, показывающий необходимость изменять вес и смещение для минимизации потерь. Предложенный алгоритм позволяет ускорить процесс обучения и снизить количество корректировок параметров нейронной сети.

Введение

Нейронные сети способны решать широкий круг задач машинного обучения – прогнозирование временных рядов [1], распознавание речи [2], компьютерное зрение [3] и т. д. Актуальность проблемы обучения нейронных сетей связана с увеличением объемов данных, а так же с разнообразием архитектур сетей. Перед применением нейронной сети на практике, её необходимо обучить. Этот процесс является крайне требовательным к вычислительным мощностям процессора, а также видеокарты. Основная задача становится не просто обучение, а нахождение нейронной сети, наилучшим образом решающей поставленную прикладную задачу. Существует семейство методов нахождения такой сети, основанных на эмпирическом исследовании. Использование этих методов предполагает, что обучение является операцией решения задачи оптимизации структуры сети. Следовательно потребность в быстром обучении еще больше возрастает. Таким образом, для эффективного обучения нейронных сетей необходим новый алгоритм обучения.

I. Архитектура нейронной сети

Базовые компоненты нейронной сети – нейроны. Нейрон представляет собой единицу обработки информации в нейронной сети. Нейрон принимает вводные данные, выполняет с ними определенные математические операции, а затем выводит результат (см. рис. 1).

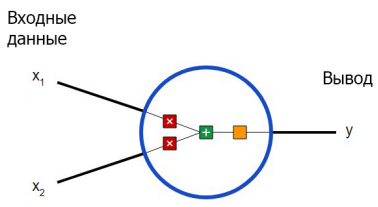

Рис. 1 – Нейрон с двумя входными данными

В первую очередь каждый вход умножается на вес: $x_{1,2} -> x_{1,2} * w_{1,2}$. Затем все взвешенные входы складываются вместе со смещением b: $x_1 * w_1 + x_2 * w_2 + b$. Затем сумма передается через функцию активации: $y = f(x_1 * w_1 + x_2 * w_2 + b)$. Функция активации используется для подключения несвязанных входных данных с выводом, у которого простая и предсказуемая форма. Как правило, в качестве используемой функцией активации берется функция сигмоида: $\sigma(x) = \frac{1}{1+e^{-x}}$ (см. рис. 2).

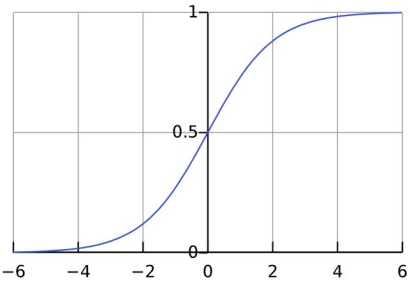

Рис. 2 – Функция сигмоида

II. Метод стохастического градиентного спуска

Персептрон — простейший вид нейронных сетей. В основе лежит математическая модель восприятия информации мозгом, состоящая из сенсоров, ассоциативных и реагирующих элементов. Нейронные сети часто обучаются стохастически, то есть на разных итерациях используются разные части данных. Это определяется, как минимум, двумя причинами: во-первых, наборы данных, используемые для обучения, часто очень большие, чтобы хранить их полностью в оперативной памяти и/или производить вычисления эффективно; во-вторых, оптимизируемая функция обычно невыпуклая. Таким образом, использование разных частей данных на каждой итерации может помочь от застревания модели в локальном минимуме. Кроме того, обучение нейронных сетей обычно производится с помощью градиентных методов первого порядка,

так как из-за большого количества параметров в нейронной сети невозможно эффективно применять методы более высоких порядков. Стандартным методом обучения нейронных сетей является метод стохастического градиентного спуска (SGD). Однако он может расходиться или сходиться очень медленно, если шаг обучения настроен недостаточно аккуратно. Поэтому существует много альтернативных методов, с целью ускорения сходимости обучения и избавить пользователя от необходимости тщательной настройки гиперпараметров [4]. Эти методы часто более эффективно вычисляют градиенты и адаптивно изменяют шаг обучения по итерациям. Суть данного алгоритма отражается в следующем уравнении: $\omega_1 \leftarrow \omega_1 - \eta \frac{\partial L}{\partial \omega_1}$, где η является константой, которая называется оценкой обучения, которая контролирует скорость обучения. Процесс тренировки нейронной сети на основе стохастического градиентного спуска будет выглядеть следующим образом:

1. Выбирается один пункт из набора данных;
2. Подсчитываются все частные производные потери по весу или смещению;
3. Применяется уравнение SGD, для обновления каждого веса и смещения;
4. Возврат к первому пункту.

У данного алгоритма можно выделить ряд преимуществ:

- Ошибка на каждом шаге считается быстро, веса меняются сразу же, что очень сильно ускоряет обучение. Обучение может сойтись ещё до того, как был выполнен единственный проход по всем тренировочным примерам. К тому же не надо хранить всю обучающую выборку в памяти;
- Стохастическии градиентныи спуск работает «более случаино», чем обычныи, и поэтому можно надеяться, что он не остановится в маленьких локальных минимумах. Пакетный же спуск хорош для строго выпуклых функций, потому что уверенно стремится к минимуму глобальному или локальному;
- Подходит для онлайн-обучения, т. е. в случаях, когда обучающая выборка постоянно обновляется.

В данном методе присутствует недостаток — обновлять веса модели после каждого тренировочного примера может быть накладно, поэтому можно скрестить два этих варианта, получив мини-пакетный (mini-batch) спуск, который за раз обрабатывает, к примеру, 100 элементов, а не все или один. За счёт возможности распараллеривания это всё равно быстрее, чем в случае с пакетным спуском, а результат даёт даже лучше (см. рис. 3) [5].

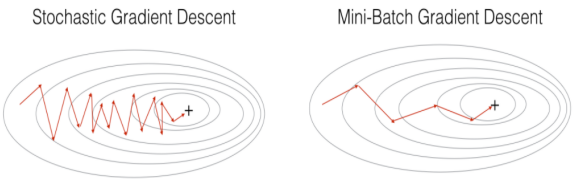

Рис. 3 – Сравнение методов градиентного спуска

Идеи иннерционных методов применяются для стохастического градиентного спуска и на практике часто дают прирост, в теории же обычно считается, что асимптотическая скорость сходимости не меняется из-за того, что основная погрешность в стохастическом градиентном спуске обусловлена дисперсией.

III. Заключение

При обучении нейросети типа «персептрон» требуется изменять весовые коэффициенты сети так, чтобы минимизировать среднюю ошибку на выходе нейронной сети при подаче на вход последовательности обучающих входных данных. Формально, чтобы сделать всего один шаг по методу градиентного спуска (сделать всего одно изменение параметров сети), необходимо подать на вход сети последовательно абсолютно весь набор обучающих данных, для каждого объекта обучающих данных вычислить ошибку и рассчитать необходимую коррекцию коэффициентов сети (но не делать эту коррекцию), и уже после подачи всех данных рассчитать сумму в корректировке каждого коэффициента сети (сумма градиентов) и произвести коррекцию коэффициентов «на один шаг». Показано, что при большом наборе обучающих данных алгоритм будет работать крайне медленно, поэтому на практике часто производят корректировку коэффициентов сети после каждого элемента обучения. Здесь значение градиента аппроксимируются градиентом функции стоимости, вычисленном только на одном элементе обучения показанного стохастического градиентного спуска является одной из форм стохастического приближения. Теория стохастических приближений даёт условия сходимости метода стохастического градиентного спуска.

IV. Список литературы

1. Kolarik, Thomas, and Gottfried Rudorfer. "Time series forecasting using neural networks."ACM Sigapl Apl Quote Quad. Vol. 25. No. 1. ACM, 1994.
2. Waibel, Alexander, et al. "Phoneme recognition using time-delay neural networks."Acoustics,Speech and Signal Processing, IEEE Transactions on 37.3 (1989): 328-339.
3. Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks."Advances in neural information processing systems. 2012
4. Mnih, Volodymyr, Nicolas Heess, and Alex Graves. "Recurrent models of visual attention."Advances in Neural Information Processing Systems. 2014.
5. Stochastic Gradient Descent - Mini-batch and more. (2017, March 30). Adventures in Machine Learning.