УДК 621.391

ВОЗМОЖНОСТИ МЕТОДА ОТЖИГА В ЗАДАЧЕ ОБУЧЕНИЯ НЕЙРОННЫХ СЕТЕЙ

В.В. МАЦКЕВИЧ

Белорусский государственный университет, Республика Беларусь

Поступила в редакцию 22 февраля 2022

Аннотация. В работе рассматривается проблема обучения нейронных сетей. Предложен алгоритм обучения на основе метода отжига. Показано, что разработанный алгоритм обладает гораздо большей эффективностью, чем существующие градиентные методы.

Ключевые слова: метод отжига, метод градиентного спуска, обучение, нейронная сеть.

Введение

В настоящее время происходит стремительное развитие цифровых технологий. Объемы получаемой информации из различных источников стремительно растут [1]. Возникает множество проблем, связанных с большими объемами данных: передача, хранение и обработка. Для решения данных проблем были разработаны различные алгоритмы.

Из-за широкого разнообразия обрабатываемой информации наибольшее распространение получили универсальные нейросетевые алгоритмы обработки. В отличие от классической концепции, где класс обрабатываемых данных зафиксирован, нейросетевой подход предлагает адаптируемость к обрабатываемой информации [2]. Процесс настройки такого алгоритма под конкретный класс данных называется обучением. Эффективность полученного решения напрямую зависит от результата обучения [3].

Несмотря на достигнутые в данном направлении результаты, проблема обучения по-прежнему является актуальной.

Существуют два основных подхода к обучению нейронных сетей. Один базируется на методологии градиентного спуска, а другой – на случайном поиске. Наиболее популярными являются градиентные методы, которые на практике обучают нейронные сети за приемлемое время. Представители другого подхода – методы отжига, обеспечивают хорошее качество, но работают существенно медленнее.

В работе сравнивается эффективность метода отжига и градиентного спуска для обучения нейронной сети на примере решения задачи сжатия цветных изображений.

Анализ проблемы

Обучение нейронных сетей представляет из себя процесс настройки отдельных параметров сети для достижения наилучшего решения. Эффективность результата оценивается с помощью заданного для решаемой задачи функционала качества.

Таким образом, обучение нейронной сети является оптимизационной задачей, где решением являются конкретные значения параметров нейронной сети, а целевой функцией – функционал качества, описывающий решаемую задачу.

Для решения оптимизационных задач используются итерационные методы, которые, в свою очередь, подразделяются на два класса – направленные и ненаправленные.

При направленном поиске каждое последующее приближение вычисляется по строго определенному правилу на основе текущего приближенного решения. К направленным относятся все методы, основанные на вычислении градиентов: метод простого градиентного

спуска, метод секущих, метод Ньютона и т.д. Ненаправленные методы (методы случайного поиска) характеризуются случайностью выбора последующего приближения. К ним можно отнести метод дифференциальной эволюции, метод отжига и т.д.

Методы случайного поиска имеют практически неограниченное пространство для поиска решения, в то время как направленные методы ограничены строгими правилами выбора последующего решения, что существенно сужает пространство поиска решения. Этот факт дает интуитивное преимущество случайного поиска в качестве полученного решения над направленными методами. Таким образом, ненаправленные методы также имеют определенную перспективу.

Для сравнения этих двух принципиально разных подходов к решению оптимизационных задач рассмотрим два сильнейших представителя из каждого класса. Из ненаправленных методов – метод отжига, из направленных методов – метод адаптивного градиентного спуска.

Алгоритм обучения на основе метода отжига

Любая нейронная сеть представима в виде набора параметров – компонент. Например, многослойный персептрон задается двумя наборами параметров: множеством весов W и набором смещений нейронов B. Такое представление сети необходимо для точного описания разработанного подхода к обучению. Алгоритм реализуется в два основных этапа.

Предварительный этап. На данном этапе производится инициализация (задание начальных значений) параметров сети и начальной температуры T_0.

Основной этап. На данном этапе обучения реализуется процедура последовательного обновления значений параметров с использованием некоторого функционала качества.

Опишем более подробно процедуру обновления параметров.

Процедура обновления. Для простоты изложения рассмотрим ее на примере множества параметров W. Для других множеств эта процедура полностью совпадает.

Множеству параметров W поставим в соответствие отрезок L_w длины l. После этого каждый элемент множества W последовательно помещаем в центр заданного отрезка. Для определения направления изменения значений параметров генерируем случайное число от 0 до 1. Если полученная реализация числа больше 0,5, то значение параметра увеличивается, в противном случае – уменьшается.

Новое значение параметра определяется в результате реализации равномерно распределенной случайной величины на отрезке, концами которого являются текущее значение параметра, и конец отрезка, в сторону которого осуществляется изменение.

Аналогично действия последовательно выполняются для других параметров сети.

Для вновь полученных значений параметров вычисляется функционал качества.

Далее принимается решение о переходе в новое состояние:

– если значение функционала больше текущего, то переходим в новое состояние с вероятностью:

$$P\left(y\middle|x\right)=\exp\left(-\frac{F\left(y\right)-F\left(x\right)}{T_i}\right), \tag{1}$$

где x – текущее состояние, y – выбранное для перехода состояние, F – минимизируемая целевая функция, T_i – температура i-ой итерации.

– в противном случае происходит безусловный переход в новое состояние.

– в случае смены состояния происходит охлаждение по правилу:

$$T_{k+1}=\frac{T_0}{\ln\left(k+1\right)}, \tag{2}$$

где k – количество совершенных итераций.

– в противном случае температура не изменяется.

После охлаждения производится проверка полученного решения на оптимальность:

решение является оптимальным, если в течение последних S итераций не проводился переход в новое состояние, либо время, отведенное на обучение, истекло.

Если полученное решение оптимально, то:

– останов алгоритма,

– в противном случае переход на следующую итерацию.

Конец алгоритма. Главным достоинством данного метода является гарантированная сходимость к точке глобального минимума, т.е. к оптимальному решению. К недостаткам можно отнести логарифмическую скорость сходимости метода.

Метод градиентного спуска

На каждой итерации градиентного спуска производится вычисление производной от целевой функции по каждому из настраиваемых параметров нейронной сети, с последующим их обновлением в противоположном градиенту направлении.

Данный подход имеет существенный недостаток: рассматриваемый метод, как правило, завершается в точках, где градиент равен нулю. Это приводит к получению решения равному локальному минимуму значения функционала качества. Как правило, значение локального минимума значительно больше значения глобального минимума.

Различные модификации метода градиентного спуска отличаются лишь правилом пересчета корректировок параметров на основе статистики изменений на предыдущих итерациях.

В проводимых нами экспериментах использовалась следующая модификация градиентного метода – метод адаптивного градиента [4]. В литературных источниках она считается наилучшей модификацией градиентного метода обучения, применяется для обучения глубоких нейронных сетей сложной архитектуры [4, 5].

Основная идея модификации заключается в «стабилизации» градиентов. Метод осуществляет вычисление корректировок с помощью специальных статистик корректировок из предыдущих итераций. Это обеспечивает снижение разности по модулю между корректировками соседних итераций. Данное свойство частично решает проблему локальных минимумов. А с другой стороны путем сглаживания предотвращаются резкие изменения параметров, что приводит к более «тонкой» настройке параметров.

К преимуществам данного подхода можно отнести линейную скорость сходимости.

Эксперименты

Для сравнения эффективности алгоритмов обучения нейронных сетей на основе методов градиентного спуска и отжига будем решать задачу сжатия изображений на выборках CIFAR-10 [6] и STL-10 [7].

Для всех степеней сжатия построена нейронная сеть из набора ограниченных машин Больцмана типа Гаусс-Бернулли.

При 32-кратном сжатии входной слой отдельной машины состоит из 48 нейронов, а выходной из 12. При этом каждая машина принимает в качестве входных данных фрагмент изображения размером 4 на 4 пикселя, но так у цветных изображений 3 цветовых канала то фрагмент изображения задается 48 значениями.

При 64-кратном сжатии входной слой отдельной машины состоит из 96 нейронов, а выходной из 12. При этом каждая машина принимает в качестве входных данных фрагмент изображения размером 4 на 8 пикселей, но так у цветных изображений 3 цветовых канала то фрагмент изображения задается 96 значениями.

При 128-кратном сжатии входной слой отдельной машины состоит из 192 нейронов, а выходной из 12. При этом каждая машина принимает в качестве входных данных фрагмент изображения размером 8 на 8 пикселей, но так у цветных изображений 3 цветовых канала то фрагмент изображения задается 192 значениями.

Следует отметить, что при более низких степенях сжатия классические алгоритмы сжатия, как правило, не уступают нейросетевому подходу, поэтому использовать более низкие степени сжатия в экспериментах для нейронных сетей нецелесообразно. С другой стороны, при более высоких степенях сжатия происходит практически полная потеря исходных изображений и сжатие в таком случае не имеет смысла.

Экспериментально было замечено, что меньшие размеры нейронных сетей приводят к более плохим результатам. Потому, что при высоких степенях сжатия (свыше 8 кратного) необходимо обобщать большие фрагменты входных данных, чтобы обеспечить качественное сжатие высокой степени. Архитектуры с большим количеством нейронов требуют гораздо большего времени на обучение, что усложняет проведение экспериментов, а также требуется большой объем входных данных для настройки большого количества параметров, что создает проблему нехватки данных для обучения.

Существует множество градиентных алгоритмов обучения ограниченных машин Больцмана: CD-k, PCD, N-PT_k. Они отличаются друг от друга способом оценки градиента функции правдоподобия.

PCD - быстрый и достаточно эффективный метод обучения. Однако, при большом количестве итераций обучения его эффективность снижается, и он уступает даже CD-1 [8]. В то же время алгоритм обучения CD-10 по эффективности примерно равен 10-PT5, однако требует гораздо меньшего объема вычислений [9]. При этом алгоритм CD-100 более эффективный, чем CD-10 [10].

Поэтому для сравнения эффективности обучения нейронных сетей в экспериментах использован описанный выше алгоритм на основе метода отжига и сравнительного расхождения (CD-100).

Для реализации процесса обучения использовалось 10000 изображений (3000 непосредственно для обучения и 7000 для валидации) для тестирования качества сжатия использовались остальные 50000 и 90000 изображений выборки соответственно.

Время, отводимое на обучение, определялось индивидуально для каждого алгоритма. Это было сделано для того, чтобы обеспечить сходимость алгоритма обучения и напрасно не расходовать вычислительные ресурсы.

Эксперименты проводились на системе Lubuntu 20.04 с процессором intel i7-4770k, видеокартой nvidia 1070 ti.

Как показали результаты экспериментов (см. табл. 1, 2) теоретическая сходимость метода отжига на практике обеспечила огромное превосходство в качестве сжатия по сравнению с градиентным методом (более чем в 2,7 раза).

Табл. 1. **Результаты обучения методом градиентного спуска**

Выборка	CIFAR-10			STL-10		
Степень сжатия, бит/пиксель	0,75	0,375	0,1875	0,75	0,375	0,1875
MSE	1898	2385	3230	1728	2220	2949
PSNR	15,5	14,6	13,3	15,9	14,8	13,6
PSNR-HVS	15,6	14,7	13,4	16	14,9	13,8
SSIM	0,383	0,27	0,16	0,307	0,255	0,225
Время обучения, ч	0,142	0,08	0,064	1,27	0,714	0,554

Табл. 2. **Результаты обучения методом отжига**

Выборка	CIFAR-10			STL-10		
Степень сжатия, бит/пиксель	0,75	0,375	0,1875	0,75	0,375	0,1875
MSE	649	855	1104	626	835	1061
PSNR	20,1	18,9	17,8	20,3	19	18
PSNR-HVS	20,3	19,1	18	20,5	19,2	18,1
SSIM	0,684	0,602	0,498	0,562	0,469	0,38
Время обучения, ч	1	1	1	8	8	8

В то же время теоретическая логарифмическая скорость сходимости отжига на практике оказалась существенно медленнее линейной градиента. Обучение методом отжига оказалось в среднем примерно в 12 раз медленнее градиентного метода. По количеству итераций

необходимых для сходимости алгоритма разница еще более заметная. Для сходимости отжига понадобилось порядка 10^6 итераций в то время как для градиентного метода порядка 10^3.

Заключение

Экспериментально было показано, что метод градиентного спуска застревает в точках локального минимума и полученное решение существенно хуже оптимального. В тоже время разработанный, на основе метода отжига, алгоритм на практике показал высокое качество обучения, что подтверждает теоретические результаты о сходимости метода к оптимальному решению. Разработанный нами алгоритм был ограничен только временем, отведенным на обучение. Это означает, что при росте вычислительной мощности компьютера возрастает число совершенных итераций отжига за отведенное время и тем самым повышается качество обучения. В то время как у градиентного алгоритма число итераций с ростом мощности компьютера не увеличивается, а лишь уменьшается время обучения.

Из полученных результатов можно сделать вывод, что метод отжига имеет в перспективе определенный потенциал в решении задачи обучения нейронных сетей. Он может существенно повысить качество обучения нейронных сетей.

ANNEALING METHOD OPPORTUNITIES IN NEURAL NETWORKS TRAINING PROBLEM

V.V. MATSKEVICH

Abstract. The paper deals with a neural network training problem. Training algorithm based on annealing method is proposed. It is shown that constructed algorithm has significantly higher efficiency that existed gradient decay methods.

Keywords: annealing method, gradient decay method, training, neural network.

Список литературы

1. Краснопрошин В.В., Мацкевич В.В. // Вестник Брестского государственного технического университета. Серия - физика математика, информатика. 2018. № 5 (113). С. 15–18.
2. Zhengxue C., Heming S., Masaru T., Jiro K. // IEEE Transactions on multimedia 2019. Vol. 14. No. 8.
3. Krasnoproshin V.V., Matskevich V.V. // Research Papers Collection “Open Semantic Technologies for Intelligent Systems” 2019. Iss. 3. P. 265–268.
4. Kingma D.P., Ba J.L. // Proc. of the 3rd International Conference on Learning Representations. 2015. P. 1–15.
5. Hamis S., Zaharia T., Rousseau O. // IEEE 23rd International Symposium on Consumer Technologies (ISCT), P. 128–133.
6. Выборка CIFAR-10 [Электронный ресурс]: – Режим доступа: https://www.cs.toronto.edu/~kriz/cifar.html – Дата доступа: 04.03.2020.
7. Выборка STL-10 [Электронный ресурс]: – Режим доступа: https://web.archive.org/web/20110803194852/http://www.stanford.edu/~acoates//stl10/ – Дата доступа: 24.04.2019.
8. Oswin K., Fischer A., Igel Ch. // Pattern Recognition Letters. 2018. Vol. 102. P. 1–7.
9. Brugge K., Fischer A., Igel Ch. // Machine Learning. 2013. Vol. 93. Iss. 1. P. 53–69.
10. Li X., Gao X., Wang Ch. // IEEE ACCESS. 2021. Vol. 9. P. 21939–21950.