СЛУЧАЙНЫЙ ПОИСК В ЗАДАЧЕ ОБУЧЕНИЯ НЕЙРОННЫХ СЕТЕЙ

В. В. Мацкевич

Белорусский государственный университет, г. Минск;

matskevich1997@gmail.com;

науч. рук. – В. В. Краснопрошин, д.т.н., проф.

В работе рассматривается актуальная проблема, связанная с обучением нейронных сетей. На сегодняшний день широко распространены градиентные алгоритмы обучения. Разработано множество модификаций градиента для повышения качества обучения. Наилучшей модификацией считается метод адаптивного момента [1]. Методы обучения на основе случайного поиска, к сожалению, мало изучены. Это связано, как принято считать, с медленной сходимостью подобных методов. Из-за чего они ранее считались трудно применимыми на практике.

Целью данной работы является разработка алгоритма обучения нейронных сетей на основе метода отжига. Представленный в работе алгоритм гарантированно сходится к оптимальному решению. Показано, что с ростом производительности компьютеров появляются большие перспективы для его практического использования. Для проверки эффективности разработанного алгоритма проводится его сравнение (на примере решения задачи сжатия цветных изображений) с методом адаптивного момента [1].

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

АНАЛИЗ ПРОБЛЕМЫ

Алгоритмы обучения на основе метода градиентного спуска обладают высокой скоростью сходимости, следовательно, не требуют большого объема вычислений и времени для обучения. Однако данные алгоритмы обладают рядом недостатков: целевая функция, должна быть дифференцируемой, что сужает класс решаемых задач; алгоритм сходится в точках, где величина градиента равна нулю, это точки перегиба либо локальных минимумов. Все это означает, что полученное алгоритмом решение, как правило, далеко от оптимально.

Для устранения указанных недостатков предлагается использовать алгоритмы обучения на основе случайного поиска, в которых на каждой итерации новое решение генерируется случайным образом на основе текущего. Это существенно расширяет пространство поиска, что в итоге повышает вероятность нахождения оптимального решения. Единственным недостатком такого подхода является медленная

сходимость алгоритма. На этапе становления нейронных сетей вычислительная мощность компьютеров была недостаточной для применения данного подхода, из-за чего сейчас повсеместно используются градиентные алгоритмы обучения.

АЛГОРИТМ ОБУЧЕНИЯ

На сегодняшний день существует два основных метода, основанные на идее случайного поиска: генетический алгоритм и метод отжига. Если градиентные алгоритмы невозможно применить, то используются, как правило, генетические алгоритмы. Однако, на практике они не гарантируют получения оптимального решения.

Метод отжига, напротив, при определенных условиях, обладает свойством сходимости к оптимальному решению, причем из любого начального приближения [2]. Единственным недостатком метода является, как принято считать, чрезмерно медленная сходимость, из-за чего он не получил широкого распространения.

Рассмотрим вначале архитектуру ограниченной машины Больцмана (ОМБ) типа Гаусс-Бернулли. Именно этот тип сети будет использоваться в экспериментах. Архитектура состоит из двух слоев нейронов – входного и скрытого. Они связаны между собой весами, образуя двудольный граф. Конкретная ОМБ задается фиксацией значений всех ее параметров – четырех групп параметров. Весов w , смещений входного и скрытого слоев (b, fb) и параметра «сигма» входного слоя σ . Таким образом, ОМБ задается следующим образом $RBM=(w, \sigma, fb, b)$. Алгоритм минимизации целевой функции F является итерационным.

Опишем теперь предлагаемый алгоритм, реализующий метод отжига.

Предварительный шаг. Инициализация начального состояния нейронной сети $RBM_0=(w_0, \sigma_0, fb_0, b_0)$ и последовательности температур, связанных следующим соотношением:

$$T_k = \frac{T_0}{\ln(k+3)}, k > 0, \quad (1)$$

где T_0 – заранее заданное значение.

Общий k -ый шаг. Шаг 1.

Генерация четырех равномерно распределенных дискретных случайных величин a, b, c, d на отрезке от нуля до количества параметров в группе. Генерация четырех случайных перестановок длиной, равной количеству параметров в группе. Первые a, b, c, d элементов перестановок задают индексы изменяемых параметров в каждой группе параметров соответственно.

Шаг 2. Генерация нового решения. Для каждого изменяемого параметра генерируется две равномерно распределенные случайные величины e, g на отрезках $[0;1], [0;l/2]$. Величина l зависит от того, какой группе принадлежит изменяемый параметр и равна l_a, l_b, l_c, l_d соответственно. Значения l для каждой группы задаются как параметры алгоритма. Пусть x_i – изменяемый параметр, а его новое значение x_i' , тогда:

$$x_i' = \begin{cases} x_i + g, e \leq 0,5 \\ x_i - g, e > 0,5 \end{cases}$$

Шаг 3. Принятие решения о переходе. Пусть x текущее решение, y – новое, сгенерированное на шаге 2 решение. Тогда решение x' на следующей итерации определяется следующим образом:

$$P(x' = y | x) = \min \left\{ 1, \exp \left(\frac{F(x) - F(y)}{T_k} \right) \right\} \quad (2)$$

Шаг 4. Проверка критерия остановки. Если время на обучение нейронной сети истекло, то алгоритм завершается. В противном случае производится переход на следующую итерацию.

Стоит отметить, с учетом разработанного генератора новых решений, использования правила перехода (2) и формулы охлаждения (1), выполняется критерий сходимости [2], а значит, алгоритм сходится к оптимальному решению, причем из любого начального приближения.

ЭКСПЕРИМЕНТЫ

Для экспериментов была выбрана выборка CIFAR-10 [3]. Для экспериментов были зафиксированы три различные степени сжатия: 16-кратное, 32-кратное и 64-кратное. Данные варианты сжатия были выбраны из двух соображений. Более высокие степени сжатия приводят к слишком большим потерям, а более низкие не представляют большого интереса. Т.к. существует достаточно большое количество эффективных алгоритмов для низких степеней сжатия, в том числе и без потерь.

Для сжатия изображений использовался ансамбль из ОМБ. Количество нейронов во входном и скрытом слое отдельной машины зависит от степени сжатия изображений. Каждая отдельная ОМБ обрабатывала отдельный фрагмент изображения. При этом фрагменты не пересекаются.

Для 16-кратного сжатия в каждой машине было установлено 48 нейронов во входном слое и 24 в скрытом. Для 32-кратного сжатия было установлено 48 нейронов во входном слое и 12 в скрытом по аналогичной

логике. Для 64-кратного сжатия было установлено 96 нейронов во входном слое и 12 в скрытом.

Для оценки потерь при сжатии использовались наиболее распространенные функционалы: корень среднеквадратической ошибки (MSER), соотношение максимального сигнала к шуму (PSNR), в том числе с поправкой на зрение человека (PSNR_HVS), структурная схожесть изображений (SSIM).

Результаты обучения

Алгоритм обучения	Метод отжига			Метод градиентного спуска		
	1,5	0,75	0,375	1,5	0,75	0,375
Степень сжатия, бит/пиксель						
MSE	413	684	855	2221	1898	2385
PSNR	22,1	19,9	18,9	14,8	15,5	14,6
PSNR_HVS	22,3	20,1	19,1	14,9	15,6	14,7
SSIM	0,767	0,668	0,602	0,307	0,383	0,27
Время обучения, ч	3	3	1	0,041	0,142	0,08

Из полученных результатов (см. Таблица) видно, что разработанный алгоритм более, чем в 2,5 раза превосходит градиентные методы по качеству обучения. Однако, разработанный алгоритм более, чем в 10 медленнее градиентов. Т.к. время обучения для метода отжига является приемлемым и нейронная сеть обучается лишь один раз, а дальше имеет значение скорость функционирования обученной сети, то медлительность алгоритма обучения в данном случае не является критичной. Стоит также отметить, не смотря на то, что обучение производилось на видеокарте nvidia 1070 ti, т.е. средней мощности, разработанный алгоритм оказался применим на практике.

С ростом вычислительной мощности компьютеров растет качество обучения и при этом существенно сокращается время. Градиентным алгоритмам рост вычислительной мощности не дает преимущества. Незначительное сокращение времени обучения не повлияет на конечный результат, а качество обучения от этого не меняется.

Из полученных результатов можно сделать вывод, что разработанный алгоритм обладает большой перспективой в обучении нейронных сетей.

Библиографические ссылки

1. Kingma D.P. Adam: A Method for Stochastic Optimization / D.P. Kingma, J.L. Ba, // Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015), pp. 1-15.
2. Hajek B. Cooling schedules for optimal annealing / B. Hajek // Mathematics of operations research, vol. 13, iss. 2, 1988.
3. Выборка CIFAR-10 [Электронный ресурс]: – Режим доступа: <https://www.cs.toronto.edu/~kriz/cifar.html> – Дата доступа: 04.03.2020.