СПИСОК ЛИТЕРАТУРЫ

1. Юшкова Е.А., Лебедев В.А. Потоки энергии и эксергии // Молодой ученый. – 2017. – №12 (146). – С. 17-19.
2. Шаргут Я., Петела Р. Эксергия. – М.: Энергия. – 1968. – 280 с.
3. Dincer I, Cengel YA Energy, entropy and exergy concepts and their roles in thermal engineering. Entropy. – 2001. – №3(3). pp 116-149
4. Гохштейн Д.П. Современные методы термодинамического анализа энергетических установок. / Д.П. Гохштейн. – М.: Энергия, – 1969. – 368 с.
5. Лебедев В.А., Юшкова Е.А. Устройство для измерения эксергии // Патент России № 2702701, 26.11.2018.

SRP

ТЕХНИЧЕСКИЕ НАУКИ
TECHNICAL SCIENCES

УДК 51

Ядута Анна Зауровна,
к.т.н., доцент, Белгородский государственный национальный исследовательский университет, г. Белгород
Yaduta Anna Zaurovna,
Belgorod National Research University, Belgorod

Гурьянова Ирина Владимировна,
ст.пр., Белгородский государственный национальный исследовательский университет, г. Белгород
Guryanova Irina Vladimirovna,
Belgorod National Research University, Belgorod

Попов Дмитрий Вадимович,
Белгородский государственный национальный исследовательский университет, г. Белгород
Popov Dmitriy Vadimovich,
Belgorod National Research University, Belgorod

ВЫЧИСЛИТЕЛЬНАЯ МАТЕМАТИКА, ИСПОЛЬЗУЕМАЯ В НЕЙРОННЫХ СЕТЯХ: ГРАДИЕНТНЫЙ СПУСК

COMPUTATIONAL MATHEMATICS USED IN NEURAL NETWORKS: GRADIENT DESCENT

Аннотация: с каждым годом искусственные нейронные сети становятся неотъемлемой частью повседневной жизни. Они интегрируются во многие электронные устройства, необходимые человеку. В данной статье рассмотрим одним из наиболее используемых методов обучения – искусственных нейронных сетей.

Abstract: every year, artificial neural networks are becoming an integral part of everyday life. They integrate into many electronic devices needed by man. In this article, we will consider one of the most used training methods – artificial neural networks.

Ключевые слова: Нейронные сети, градиентный спуск, матрица, машинное обучение.

Keywords: Neural networks, gradient descent, matrix, machine learning.

Градиентный спуск – метод нахождения локального экстремума (минимума или максимума) функции с помощью движения вдоль градиента. Для минимизации функции в направлении градиента используются методы одномерной оптимизации, например, метод золотого сечения. Также можно искать не наилучшую точку в направлении градиента, а какую-либо лучше текущей.

В нейронных сетях градиентный спуск известен как метод обратного распространения ошибки, который является самым популярным методом обучения искусственных нейронных сетей без учителя. Данным методом хорошо обучаются самые популярные нейронные сети, такие как персептрон и глубокие нейронные сети [1].

Целью данной работы является изучение и анализ эффективности метода градиентного спуска с точки зрения математики.

При рассмотрении примера вычисления линейной регрессии с помощью метода наименьших квадратов, параметры были найдены аналитически:

$$\Theta = A^{+} Y,$$

где Θ – вектор параметров, Y – вектор, состоящий из значений зависимой переменной, A^{+} – псевдообратная матрица. Это решение наглядное, точное и короткое, но есть проблема, которую можно решить численно.

Градиентный спуск – метод численной оптимизации, который может быть использован во многих алгоритмах, где требуется найти экстремум функции – нейронные сети, SVM (англ. support vector machine – метод опорных векторов), k-средних, регрессии, однако проще его воспринять в чистом виде.

Проблема заключается в том, что вычисление псевдообратной матрицы очень затратно: два умножения по $O(n^3)$, нахождение обратной матрицы – $O(n^3)$, умножение матрицы вектор $O(np)$, где *n* – количество элементов в матрице *A* (количество признаков * количество элементов в тестовой выборке). Если количество элементов в матрице A велико (> 10^6 – например), то даже если в наличии 10000 ядер, нахождение решения аналитически может затянуться. Также первая производная может оказаться нелинейной, что усложнит решение, аналитического решения может не оказаться вовсе или данные могут просто не влезть в память и потребуется итеративный алгоритм [4]. Бывает, что в плюсы записывают и то, что численное решение не идеально точное, в связи с чем функцию стоимости минимизируют численными методами. Задачу нахождения экстремума называют задачей оптимизации [3].

Обозначим функцию потерь *S (Θ)* как

$$(\Theta) = \sum_{i=0}^{n} (y_i - \acute{y}_i)^2$$

Градиент – это вектор вида

$$\nabla f = \frac{\partial f}{\partial x_1} e_1 + \cdots + \frac{\partial f}{\partial x_n} e_n ,$$

Где

$$\frac{\partial f}{\partial x_n}$$

– это частная производная. Одним из свойств градиента является то, что он указывает направление, в котором некоторая функция *f* возрастает больше всего, что следует из определения производной. Возьмем некоторую точку *a* – в окрестности этой точки функция F должна быть определена и дифференцируема, тогда вектор антиградиента будет указывать на направление, в котором функция F убывает быстрее всего. Отсюда делается вывод, что в некоторой точке *b*, равной $b = a - a\nabla F(a)$, при некотором малом *α* значение функции будет меньше либо равным значению в точке *a*. Можно представить это, как движение вниз по холму – сделав шаг вниз, текущая позиция будет ниже, чем предыдущая. Таким образом, на каждом следующем шаге высота будет как минимум не увеличиваться. Получим формулу для нахождения неизвестных параметров:

$$\theta_j = \theta_j - \alpha \frac{\partial S(\Theta)}{\partial \theta_j}$$

α – это шаг метода. В задачах машинного обучения его называют скоростью обучения.

Метод очень прост и интуитивен, возьмем простой двумерный пример для демонстрации:

Необходимо минимизировать функцию вида y = $(\theta - 5)^2$. Минимизировать значит найти при каком значении θ функция f (θ)

принимает минимальное значение. Определим последовательность действий:

1) Необходима производная по

$$\theta : \frac{\partial y}{\partial \theta} = 2(\theta - 5)$$

2) Установим начальное значение $\theta = 0$

3) Установим размер шага – попробуем несколько значений – 0.1, 0.9, 1.2, чтобы посмотреть, как этот выбор повлияет на сходимость.

4) 25 раз подряд выполним указанную выше формулу:

$$\theta^n = \theta^{n-1} - \alpha \frac{\partial y}{\partial \theta}$$

Так как неизвестный параметр только один, то и формула только одна. При значении, равном 1.2, метод расходится, на каждом шаге опускаясь не ниже, а наоборот выше, устремляясь в бесконечность. Шаг в простой реализации подбирается вручную и его размер зависит от данных – на каких-нибудь ненормализованных значениях с большим градиентом и 0.0001 может приводить к расхождению [2].

Рассмотрим работу алгоритма на трехмерном графике. Часто рисуют только изолинии какой-то фигуры. Взят простой параболоид вращения: $z = 4x^2 + 16y^2$.

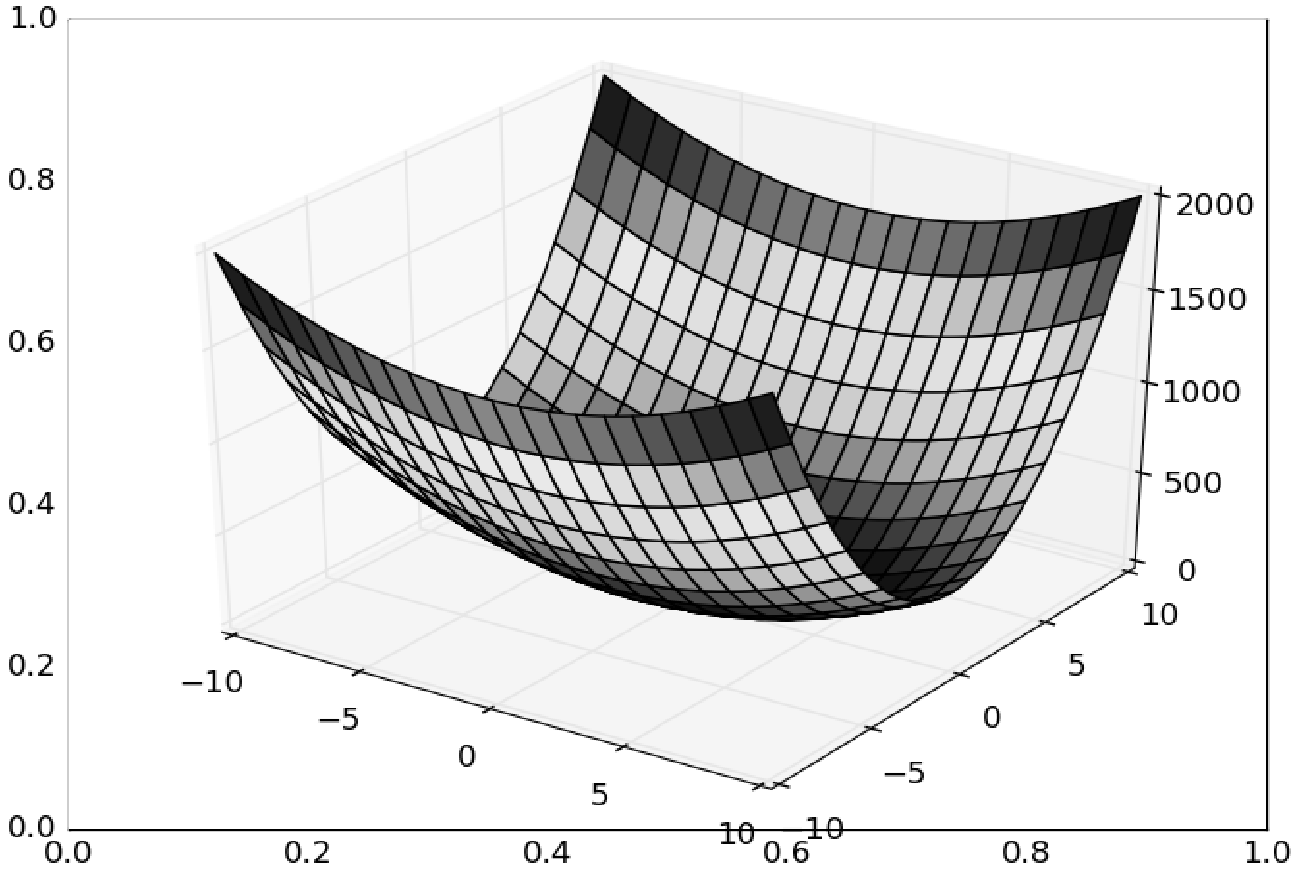

Рисунок 1 – График с изолиниями – «вид сверху»

Все линии градиента направлены перпендикулярно изолиниям, поэтому двигаясь в сторону антиградиента, не получится за один шаг сразу же прыгнуть в минимум – градиент указывает совсем не туда.

После графического пояснения, найдем формулу для вычисления неизвестных параметров θ_j линейной регрессии с методом наименьших квадратов.

$$\frac{\partial S(\Theta)}{\partial \theta_j} = \frac{\partial}{\partial \theta_j}\left(y - \sum_{i=0}^{p} \theta_i x_i\right)^2 = -2\left(y - \sum_{i=0}^{p} \theta_i x_i\right) x_j$$

Если бы количество элементов в тестовой выборке было равно единице, то формулу можно было бы так и оставить, и считать. В случае, когда в наличии *n* элементов алгоритм выглядит так:

Повторять *v* раз

$$\theta_j = \theta_j + a \sum_{i=0}^{n} \left(y^{(i)} - \acute{y}^{(i)}\right) x_j^{\{i\}}$$

для каждого *j* одновременно, где *n* – количество элементов в обучающей выборке, v – количество итераций [2].

Требование одновременности означает, что производная должна быть вычислена со старыми значениями θ, не стоит вычислять сначала отдельно первый параметр, затем второй и т.д., потому что после изменения первого параметра отдельно, производная также изменит свое значение, а если вычислять значения параметров по одиночке, то это уже будет не градиентный спуск.

Заключение: В данной статье представлен и рассмотрен метод вычислительной математики – градиентный спуск. Метод является наиболее используемым в глубоком обучении самых популярных типов искусственных нейронных сетей.

СПИСОК ЛИТЕРАТУРЫ

1. Кутузов А.А. Градиентные методы в задачах математической физики. Методы функционального анализа и их приложения. Самара, 1994.

2. Бобылев Н. А., Булатов А. В., Кутузов А. А., Коровин С. К. Некоторые свойства математических моделей нейронных сетей. // Автоматика и телемеханика. №. 4, 1997.

3. Дементьев С.Н., Кутузов А.А. Градиентные методы в задачах управления распределёнными системами. XXVI Воронежская зимняя математическая школа. Воронеж, 1994.

4. Рассел С. Искусственный интеллект. Современный подход / С. Рассел, П. Норвиг. – 2-е изд. – М.: Вильямс, 2007. – 1408 с.