МОСКОВСКИЙ ФИЗИКО-ТЕХНИЧЕСКИЙ
ИНСТИТУТ

Проектная работа

Adam: A method for stochastic optimization

Выполнили: Крючков Матвей, Кондакова Алина
Швамонян Эдгар, Кожевников Савелий

Долгопрудный, 2019 г.

Содержание

1 Введение

Adam - метод оптимизации, который требует только вычисления градиентов первого порядка и небольшого объема памяти. Был представлен Diederik P. Kingma и Jimmy Ba 22-ого декабря 2014 года, последняя версия датируется 30-ым января 2017 года. Его можно использовать вместо классической процедуры стохастического градиентного спуска (SGD) для итеративного обновления весов сети на основе обучающих данных.

Преимущества использования Adam в невыпуклых задачах оптимизации:

- Прост в реализации
- Эффективен в вычислениях
- Требует мало памяти
- Инвариантен к изменению масштаба градиента по диагонали
- Эффективен для задач с большим объемом данных и/или параметров
- Подходит для нестационарных целей
- Подходит для задач с очень шумными и/или редкими градиентами
- Интуитивно понятен, из-за чего прост в настройке параметров

В третьей секции мы рассмотрим псевдокод данного алгоритма, который уже сам по себе проиллюстрирует некоторые преимущества.

2 Сравнение с другими алгоритмами

Он комбинирует преимущества двух ранее популярных методов - *AdaGrad* и *RMSProp*.

AdaGrad - Адаптивный Градиентный Алгоритм. Поддерживает скорость обучения по одному параметру, которая улучшает производительность при проблемах с разреженными градиентами.

RMSProp - Среднеквадратичное распространение. Поддерживает скорость обучения по каждому параметру, адаптированные на основе среднего значения последних величин градиентов для веса, то есть эффективно справляется с онлайн и нестационарными задачами.

Adam использует среднее значение вторых моментов градиентов, а также средний первый момент, как в RMSProp. В частности, алгоритм вычисляет экспоненциальную скользящую среднюю градиента и квадрата градиента, а также имеет два параметра для управления скоростями их затухания.

Алгоритм Adam чаще всего используется по умолчанию в задачах оптимизации для приложений глубокого обучения из-за преимуществ выше и коррекции смещения, позволяющей быстрее работать к концу оптимизации по сравнению с другими алгоритмами (AdaGrad, RMSProp, Adadelta и прочие). Но все же иногда наряду с данным алгоритмом рекомендуется использование SGD + Nesterov Momentum.

3 Алгоритм Адама

3.1 Псевдокод

Замечание 1: Все операции над векторами в проекте - поэлементные.

Вход:

Шаг алгоритма: α

Параметры для управления скоростями затухания экспоненциальных скользящих: $\beta_1, \beta_2 \in [0, 1)$

Функция от случайной величины θ: $f(\theta)$

Начальное значение вектора θ: θ_0

Инициализация:

$m_0 = 0$ - первый момент вектора

$v_0 = 0$ - второй момент вектора

$t = 0$ - временная метка

Алгоритм:

while θ_t not converged **do**:

$t \mathrel{+}= 1$

$g_t = \nabla_\theta f_t(\theta_{t-1})$

$m_t = \beta m_{t-1} + (1 - \beta_1) g_t$

$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$

$\hat{m}_t = m_t / (1 - (\beta_1)^t)$

$\hat{v}_t = v_t / (1 - (\beta_2)^t)$

$\theta_t = \theta_{t-1} - \alpha \hat{m}_t / (\sqrt{\hat{v}_t} + \epsilon)$

end while

Замечание 2: В оригинальной статье рекомендуются следующие параметры: $\alpha = 0.001$, $\beta_1 = 0.999$, $\beta_2 = 0.99$, $\epsilon = 10^{-8}$. Они считаются оптимальными для большинства задач.

3.2 Описание

Пусть в псевдокоде $f(\theta)$ - шумная: дифференцируемая функция от случайной величины θ. Задача - минимизировать значение ее математического ожидания.

Пусть $f_1(\theta)$, ..., $f_T(\theta)$ - ее значения в момент времени $1, ..., T$, а $g_t = \nabla_\theta f_t(\theta_{t-1})$ - градиенты этих значений.

Алгоритм обновляет экспоненциальную скользящую градиента и квадрата градиента, где параметры $\beta_1, \beta_2 \in [0, 1)$ контролируют изменение. Эта скользящая - оценка параметров 1-ого (главного) момента и 2-ого (нецентрального) момента. Изначально оценки инициализированы нулями, а параметры берутся близкими к единице.

Начальная инициализация может быть улучшена с помощью правила обновнения Adam'a (Adam's update rule). Об этом будет написано далее.

Заметим, что эффективность данного псевдокода может быть увеличена переменой порядка вычисления, как-то: заменой последних трех строк следующими:

$$\alpha_t = \alpha\sqrt{1-\beta_2^t}/(1-\beta_1^t)$$

$$\theta_t = \theta_{t-1} - \alpha_t m_t/(\sqrt{v_t} + \hat{\epsilon})$$

3.3 Правило обновления

Важной частью данного правила является аккуратный выбор stepsize. При условии, что $\epsilon = 0$, эффективный шаг при временной отметке t есть $\Delta_t = \alpha\hat{m}_t/\sqrt{\hat{v}_t}$. У эффективного шага есть две границы:

$$|\Delta_t| \leq \alpha(1-\beta_1)/\sqrt{1-\beta_2}, \text{ при условии, что} (1-\beta_1) > \sqrt{1-\beta_2}$$

$|\Delta_t| \leq \alpha$, в противном случае.

Первый случай имеет место в очень редких случаях: когда градиент был нулевым в любой момент времени, кроме текущего. Для менее редких случаев эффективный размер шага будет меньше. Когда $(1-\beta_1) = \sqrt{1-\beta_2}$, имеем: $|\hat{m}_t/\sqrt{\hat{v}_t}| < 1$, для этого $|\Delta_t| < \alpha$.

В более общих сценариях, будем иметь, что $\hat{m}_t/\sqrt{\hat{v}_t} \approx \pm 1$, поскольку $|\mathbb{E}[g]/\sqrt{\mathbb{E}[g^2]}| \leq 1$. Эффективная величина шагов, предпринимаемых на каждой временной отметке, приблизительно ограничена настройкой шага α, то есть $|\Delta_t| \lessapprox \alpha$.

Это можно понимать как установление доверительного интервала вокруг текущего значения параметра, за пределами которого текущая оценка градиента не обеспечивает достаточной информации. Это обычно позволяет относительно легко узнать правильный масштаб α заранее.

К примеру, во многих моделях машинного обучения мы часто заранее знаем, в каком доверительном интервале находится наше значение; также мы нередко знаем предварительное распределение параметра. Поскольку α устанавливает (верхнюю границу) величину шага, мы часто можем вывести правильный порядок величины , такой, что оптимальное значение параметра может быть достигнуто из θ_0 в течение некоторого числа итераций.

Проще говоря, если назвать величину $\hat{m}_t/\sqrt{\hat{v}_t}$ signal-to-noise (SNR). Чем меньше этот сигнал, тем значение stepsize Δ_t будет ближе к 0. Также чем меньше сигнал, тем с меньшей уверенностью мы можем сказать о сонаправленности вектора $\hat{m}_t$ и вектора градиента.

Например, сигнал обычно становится близок к 0 при приближении к оптимальному значению, что приходит к меньшей эффективности одного шага работы.

Заметим, что эффективный stepsize Δ_t инвариантен относительно величины градиента: изменяя величину градиента g в c раз, мы тем самым изменяем $\hat{m}_t$ в c раз и $\hat{v}_t$ в c^2 раз, что в итоге сокращается: $(c\hat{m}_t)/(\sqrt{c^2\hat{v}_t}) = \hat{m}_t/\sqrt{\hat{v}_t}$

4 Корректировка шага

В этой главе мы выведем условие для оценки второго момента, вычисление для первого момента аналогично. Пусть g градиент случайной функции f. Мы хотим оценить второй момент (нецентральную дисперсию), используя экспоненциальную скользящую квадрата градиента с параметром β_2.
Пусть $g_1, ..., g_T$ градиенты в моменты времени $1, ..., T$, каждый из которых вычисляется с помощью распределения градиента $g_t \sim p(g_t)$. Инициализируем экспоненциальную скользящую как $v_0 = 0$ (вектор нулей). Для начала заметим, что выражение для обновления экспоненциальной скользящей в момент времени t, т.е. $v_t = \beta_2 v_{t-1} + (1-\beta_2)g_t^2$ может быть переписано как функция от градиентов всех предыдущих моментов времени:

$$v_t = (1-\beta_2)\sum_{i=1}^{t}\beta_2^{t-i}g_i^2$$

.

Мы хотим узнать как $\mathbb{E}[v_t]$ связано со вторым моментом $\mathbb{E}[g_t^2]$, чтобы мы смогли выразить одно через другое. Для этого возьмем математическое ожидание от обеих частей этого выражения:

$$\mathbb{E}[v_t] = \mathbb{E}[(1-\beta_2)\sum_{i=1}^{t}\beta_2^{t-i}g_i^2] =$$

$$= \mathbb{E}[g_t^2](1-\beta_2)\sum_{i=1}^{t}\beta_2^{t-i} + \xi =$$

$$= \mathbb{E}[g_t^2](1-(\beta_2)^t) + \xi$$

Где $\xi = 0$, если второй момент $\mathbb{E}[g_i^2]$ равен константе. Иначе ξ можно положить маленькой константе, т.к. параметр затухания β_1 должен быть выбран таким образом, что экспоненциальная скользящая средняя присваивает малые веса градиентам на временной метке $t = \infty$.
В условии осталось $(1-(\beta_2)^t)$, потому что мы изначально инициализировали скользящую вектором нулей. Из-за этого в алгоритме мы делим на $(1-(\beta_2)^t)$, чтобы скорректировать обновление.

В случае разреженных градиентов, для достоверной оценки второго момента нужно усреднять по градиентам, выбирая малые значения β_2. Таким образом, в случае малых β_2, начальные шаги работы алгоритма становятся больше.

5 Анализ сходимости

Проанализируем сходимость алгоритма Адама.
Пусть дана произвольная неизвестная последовательность выпуклых функций $f_1(\theta), ..., f_T(\theta)$. В каждый момент времени t мы хотим предсказать значение параметра θ_t и оценить его с помощью f_t. Так как мы ничего не знаем про последовательность функций, мы оцениваем наш алгоритм с отклонением равным сумме предыдущих разностей между предсказанным значением $f_t(\theta_t)$ и

значением $f_t(\theta^*)$ - значением в лучшей точке θ^* из множества $\mathcal{X}$ для всех предыдущих шагов. Таким образом

$$R(T) = \sum_{t=1}^{T}[f_t(\theta_t) - f_t(\theta^*)]$$

где $\theta^* = \arg\min_{\theta\in\mathcal{X}} \sum_{t=1}^{T}[f_t(\theta)]$. Покажем, что отклонение имеет ассимптотику $O(\sqrt{T})$. Этот результат является одним из лучших для данной задачи выпуклой оптимизации.

Введем некоторые обозначения. Положим $g_t = \nabla f_t(\theta_t)$, а $g_{t,i}$ - i-й элемент вектора g_t. Определим $g_{1:t,i} \in \mathbb{R}^t$ как вектор, содержащий i-е координаты градиентов от 1 до t, т.е. $g_{1:t,i} = [g_{1,i}, g_{2,i}, ..., g_{t,i}]$. Также положим $\gamma = \frac{\beta_1^2}{\sqrt{\beta_2}}$.

Замечание 2: Следующая теорема выполняется, когда α_t убывает со скоростью $t^{-\frac{1}{2}}$ и параметр затухания $\beta_{1,t}$ убывает экспоненциально с параметром λ, т.е. достаточно близок к 1.

Теорема 1.
Пусть функция f_t имеет ограниченные градиенты $\|\nabla f_t(\theta)\|_\infty \leq G_\infty$ для всех $\theta \in \mathbb{R}^d$. Пусть также расстояние между любыми θ_t в алгоритме Адама ограничено, т.е. $\|\theta_n - \theta_m\|_2 \leq D$ и $\|\theta_m - \theta_n\|_\infty \leq D_\infty$ для любых $n, m \in \{1, ..., T\}$, а $\beta_1, \beta_2 \in [0, 1)$ удовлетворяют неравенству $\frac{\beta_1^2}{\sqrt{\beta_2}} < 1$. Положим $\alpha_t = \frac{\alpha}{\sqrt{t}}$ и $\beta_{1,t} = \beta_1(\lambda_t)^{t-1}$, $\lambda \in (-1, 1)$.
Тогда $\forall T \geq 1$ в алгоритме Адама верно

$$R(T) \leq \frac{D^2}{2\alpha(1-\beta_1)} \sum_{i=1}^{d} \sqrt{T\hat{v}_{T,i}} + \frac{\alpha(1+\beta_1)G_\infty}{(1-\beta_1)\sqrt{1-\beta_2}(1-\gamma)^2} \sum_{i=1}^{d} \|g_{1:T,i}\|_2 + \sum_{i=1}^{d} \frac{D_\infty^2 G_\infty \sqrt{1-\beta_2}}{2\alpha(1-\beta_1)(1-\lambda)^2}$$

Доказательство. Оставим это в качестве тривиального упражнения для пытливого читателя. □

Вышеописанная теорема упрощается при разреженных данных с ограниченными градиентами. В таком случае, сумма становится намного меньше своей верхней грани

$$\sum_{i=1}^{d} \|g_{1:T,i}\|_2 << dG_\infty\sqrt{T}$$

и

$$\sum_{i=1}^{d} \sqrt{T\hat{v}_{T,i}} << dG_\infty\sqrt{T}$$

В частности, адаптивные методы типа Adam и Adagrad могут достигать ассимптотики $O(\log_2 d\sqrt{T})$, а неадаптивные - $O(\sqrt{dT})$.

Замечание 3: В оригинальной статье утверждается, что стремление $\beta_{1,t} \to 0$ важно в анализе и также оно совпадает с предыдущими эмпирическими вычислениями. Например, ранее ученые предлагали уменьшать коэффициент момента в конце обучения, чтобы улучшить сходимость.

Теорема 2. *Пусть функция f_t имеет ограниченные градиенты $\|\nabla f_t(\theta)\|_\infty \leq G_\infty$ для всех $\theta \in \mathbb{R}^d$. Пусть также расстояние между любыми θ_t в алгоритме Адама ограничено, т.е. $\|\theta_n - \theta_m\|_2 \leq D$ и $\|\theta_m - \theta_n\|_\infty \leq D_\infty$ для любых $n, m \in \{1, ..., T\}$.*
Тогда $\forall T \geq 1$ в алгоритме Адама верно:

$$\frac{R(T)}{T} = O(\frac{1}{\sqrt{T}})$$

Доказательство. Этот результат тривиально получается, используя **Теорему 1** и условие $\sum_{i=1}^{d} \|g_{1:T,i}\|_2 << dG_\infty\sqrt{T}$, откуда $\lim_{T\to\infty} \frac{R(T)}{T} = 0$ □

Список литературы

[1] Kingma, Diederik P., and Jimmy Ba. "Adam: A method for stochastic optimization."
arXiv preprint arXiv:1412.6980, 2014

[2] Sebastian Ruder. "An overview of gradient descent optimization algorithms."
arXiv preprint arXiv:1609.04747, 2017

[3] Stochastic gradient descent
from Wikipedia, the free encyclopedia

[4] Polyak, Boris T and Juditsky, Anatoli B. "Acceleration of stochastic approximation by averaging."
SIAM Journal on Control and Optimization, 30(4):838-855, 1992

[5] Zeiler, Matthew D. Adadelta. "An adaptive learning rate method."
arXiv preprint arXiv:1212.5701, 2012.