Домашнее задание 7

**1.В чем идея и зачем нужно ускорение (accelerate):**

**Машинное обучение, если быть точным, суть контролируемого обучения состоит в том, чтобы дать серию обучающих выборок (, ) и попытаться изучить соотношение отображения x-> y, так что, учитывая новый x, даже если этого x нет в обучающей выборке, это может сделать вывод модели y^, максимально приближенный к реальному Y.**

**Лосс функция(loss function): Функция потерь используется для оценки разрыва между выходным значением y^ модели и истинным значением y и для определения направления оптимизации модели.**

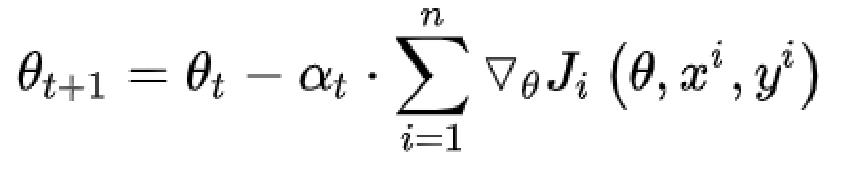
**Оптимизатор(optimizer): это алгоритм, который минимизирует (максимизирует) функцию потерь путем обучения параметров оптимизации. В процессе оптимизации модели он динамически регулирует размер и направление градиента, чтобы модель могла сходиться в лучшем положении или сходиться с большей скоростью.**

**Оптимизаторы часто играют важную роль в машинном обучении и глубоком обучении. Из-за выбора разных оптимизаторов для одной и той же модели производительность может сильно различаться, и даже некоторые модели не поддаются обучению.**

**Сейчас сушествуют многие алгоритмы оптимайзеров, например BGD, SGD, SGD+Momentum,MBGD,Adagrad,Adadelta,Adam и т д. Давайте кратко представим различные часто используемые оптимизаторы**

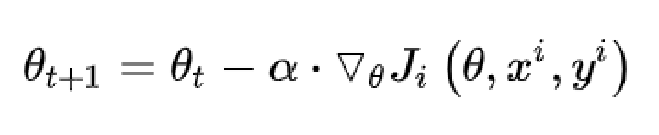
**2.Первый тип оптимизаторов(основываться на Gradient Descent):**

**a)** **BGD: Предположим, что общее количество обучающих выборок равно n, а выборки равны {(x1, y1),.... (xn, yn)}, параметр модели равен θ, функция потерь равна J (θ), а скорость обучения равна α, затем используйте BGD для обновления параметра как：**

****

**Плюс:** **Количество итераций относительно невелико Недостатки:** **При каждом обновлении параметра используются все обучающие данные (например, если их m, используются m). Если обучающих данных много, это отнимает очень много времени.**

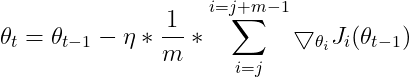
**b)SGD:** **Самое большое отличие от BGD заключается в том, что при обновлении параметров не учитываются все обучающие выборки, а затем суммируются и делятся на общее число, но берется любая точка выборки, а затем эта точка выборки используется для обновления.Выражение параметра обновления является:**

****

**Плюс:** **Градиент вычисления выполняется быстро, и SGD может хорошо сходиться при малом шуме.Для крупномасштабных данных обучение происходит очень быстро. Возьмите большое количество выборок из данных, чтобы вычислить градиент и обновить параметры.**

**Недостатки:** **Здесь больше шума, чем в BGD, и могут быть ошибки в направлении обновления веса, так что SGD не движется в направлении общей оптимизации на каждой итерации.**

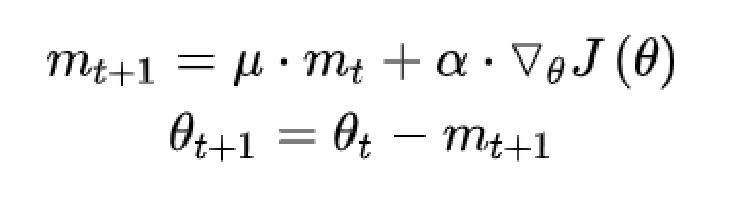
**с)MBGD: это компромисс, который сочетает в себе BGD и SGD. Для набора данных, содержащего n обучающих выборок, при каждом обновлении параметров выберите мини-пакетную выборку данных размером m (m<n) для вычисления ее градиента. Формула обновления параметров выглядит следующим образом, где j - начало партия:**

****

**Плюс:** **Может сходиться очень быстро и обладает определенной способностью избавляться от локальной оптимальности**

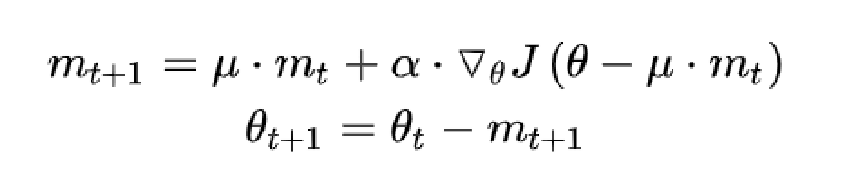
**Недостатки:1)** **Случайный выбор градиентов в то же время приведет к появлению шума, так что направление обновления веса может быть неправильным.2)** **Не удается решить проблему локального оптимального решения**

**d)SGD+Momentum:** **Когда параметры обновляются, ранее обновленное направление сохраняется в определенной степени, и в то же время градиент текущего пакета используется для точной настройки окончательного направления обновления. Короче говоря, текущий градиент ускоряется за счет накопления предыдущего импульса.Предполагая, что mt представляет импульс в момент времени t, а µ представляет коэффициент импульса, обычно со значением 0,9 или приблизительным значением, и импульс увеличивается на основе SGD, формула обновления параметра выглядит следующим образом:**

****

**Плюс:** **Это может ускорить конвергенцию SGD и подавить колебания.**

**e)NAG:** **SGD+Momentum сохраняет градиент предыдущего времени J(θ) и не вносит в него никаких изменений. NAG - это улучшение импульса. Внесите коррекцию при обновлении градиента. Конкретный подход заключается в добавлении импульса предыдущего времени µ\* mt к текущему градиентуJ(θ), и градиент изменяется наJ(θ-u\*mt)**

****

**Плюс:** **Дополнительная скорость сходимости ошибок улучшена с O() (после k шагов) до O(), что быстрее стремиться и менее колеблющееся, чем SGD+Momentum.**

**3.Второй тип оптимизаторы(само-адаптивное обучение):**

**В машинном обучении скорость обучения является очень важным гиперпараметром, но скорость обучения очень трудно определить. Хотя надлежащая скорость обучения может быть определена с помощью нескольких тренингов, как правило, не известно, сколько тренингов может обеспечить оптимальную скорость обучения. Это метафизическое событие и требует относительно высокого уровня человеческого опыт, так что есть ли какие-то стратегии для адаптивной корректировки размера скорости обучения, чтобы увеличить скорость обучения.**

**a)AdaGrad:** **Он может регулировать различные скорости обучения для каждого отдельного параметра, обновлять часто изменяющиеся параметры с меньшими шагами и разрежать параметры с большими шагами.Формула:**

****

****

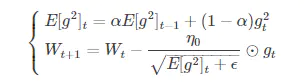
**где -диагональная матрица,(i,i)-это сумма квадратов градиента параметра θ\_i в время t.**

****

**Плюс:** **Это уменьшает необходимость ручной настройки скорости обучения и обычно занимает 0,01 секунды для скорости обучения.**

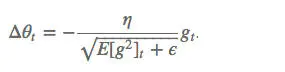
**Недостатки:** **Знаменатель будет продолжать накапливаться, так что скорость обучения будет сокращаться и в конечном итоге станет очень маленькой**

**b)RMSprop:** **Градиентное накопление AdaGrad модифицировано до экспоненциально взвешенного скользящего среднего, чтобы оно лучше работало в невыпуклых условиях.**

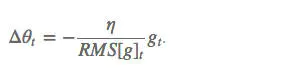
****

**Плюс:** **Может преодолеть проблему резкого уменьшения градиента AdaGrad,** **Особенно при нестабильных (нестационарных) целевых функциях, он работает лучше, чем базовые SGD, Momentum и AdaGrad.**

**c)Adadelta:** **Этот алгоритм является улучшением Adagrad. По сравнению с Adagrad, G знаменателя заменен средним значением затухания квадрата градиента в прошлом и средним значением экспоненциального затухания**

****

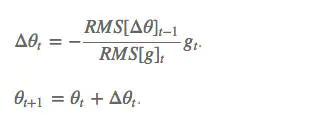
**Этот знаменатель эквивалентен среднеквадратичному значению градиента** root mean squared (RMS) :

****

**Где формула расчета E выглядит следующим образом, время t зависит от среднего значения предыдущего времени и текущего градиента：**

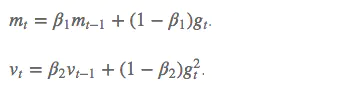
****

**Кроме того, скорость обучения η также заменяется среднеквадратичным значением [Δθ], в этом случае нам даже не нужно заранее устанавливать скорость обучения：**

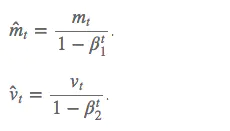
****

**Плюс:** **нам даже не нужно устанавливать скорость обучения по умолчанию, поскольку она была исключена из правила обновления.**

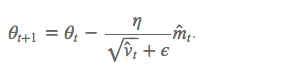
**d)Adam:** **Этот алгоритм является еще одним методом расчета адаптивной скорости обучения для каждого параметра.Это эквивалентно RMSProp + Momentum.** **Adam - едва ли не самый широко используемый адаптивный оптимизатор скорости обучения, и у него есть свои уникальные особенности.В Adam импульс непосредственно включается в вычисление момента градиента первого порядка.Ключом к Adam является коррекция смещения, которая корректирует момент первого порядка (член импульса) и (нецентральную) оценку момента второго порядка, В дополнение к хранению среднего значения экспоненциального ослабления квадрата vt прошлого градиента, такого как Adadelta и RMSProp, он также сохраняет среднее значение экспоненциального ослабления прошлого градиента mt, такого как momentum.：**

****

**Если m\_t и v\_t инициализированы векторами 0, то они будут смещены в сторону 0, поэтому выполняется коррекция отклонения, и эти отклонения компенсируются путем вычисления mt и vt после коррекции отклонения.：**

****

**Правила обновления градиента:**

****

**m\_t и v\_t имеют импульс первого порядка и импульс второго порядка, соответственно, установленные значения гиперпараметра: β1 = 0,9, β2 = 0,999, β = 10e-8**

**Плюс:** **После коррекции смещения скорость обучения каждой итерации имеет определенный диапазон, что делает параметры относительно стабильными.Adam сочетает в себе преимущества способности Adagrad обрабатывать разреженные градиенты и способности RMSProp обрабатывать нестационарные цели**