

УДК 004.855.5

УСКОРЕНИЕ ОБУЧЕНИЯ НЕЙРОННОЙ СЕТИ

М.Ю. ВАШКЕВИЧ*(Представлено: канд. физ.-мат. наук, доц. О.Н. ПЕТРОВИЧ)*

В статье рассматривается возможность ускорения обучения нейронной сети с использованием библиотеки Batch Normalization, а также возможные минусы данного ускорения.

Введение. В современном мире нейронные сети находят себе всё больше применений в различных областях науки и бизнеса. Причем чем сложнее задача, тем более сложной получается нейросеть.

Обучение сложных нейронных сетей иногда может занимать дни и недели только для одной конфигурации. А чтобы подобрать оптимальную конфигурацию для конкретной задачи, требуется запустить обучение несколько раз — это может занять месяцы вычислений даже на действительно мощной машине.

В какой-то момент, знакомясь с представленным в 2015 году методом Batch Normalization от компании Google, для решения задачи связанной с распознаванием лиц, удалось существенно улучшить скорость работы нейросети.[1]

Основной раздел. В современном мире нейронные сети находят себе всё больше применений в различных областях науки и бизнеса. Причем чем сложнее задача, тем более сложной получается нейросеть.

Обучение сложных нейронных сетей иногда может занимать дни и недели только для одной конфигурации. А чтобы подобрать оптимальную конфигурацию для конкретной задачи, требуется запустить обучение несколько раз — это может занять месяцы вычислений даже на действительно мощной машине.

В какой-то момент, знакомясь с представленным в 2015 году методом Batch Normalization от компании Google, для решения задачи связанной с распознаванием лиц, удалось существенно улучшить скорость работы нейросети.

В данной статье будет представлена возможность совмещения дух актуальных на сегодняшний день задач — это задача компьютерного зрения и машинного обучения. В качестве проектировки архитектуры нейронной сети, будет использоваться Batch Normalization для ускорения обучения нейронной сети. Обучение же нейронной сети (написанной с использованием популярной в рамках computer vision библиотеки Caffe) проводилось на базе из 3 миллионов лиц 14 тысяч различных людей.

В данной задаче необходимо произвести классификацию на 14700 классов. Известна точность классификации на тестовой выборке: 94,5%. При этом для этого потребовалось 420 тысяч итераций обучения — а это 95 часов на видеокарте NVidia Titan X. Изначально для данной нейросети использовались некоторые стандартные способы ускорения обучения:

- 1) Увеличение learning rate
- 2) Уменьшение число параметров сети
- 3) Изменение learning rate в процессе обучения особым способом

Все методы были применены к данной сети, однако в процессе применения данных способов была обнаружена потеря точности обучения.

Рассмотрим классическую нейронную сеть с несколькими слоями. Каждый слой имеет множество входов и множество выходов. Сеть обучается методом обратного распространения ошибки, по батчам, то есть ошибка считается по какому-то подмножеству обучающей выборки.

Стандартный способ нормировки — для каждого k рассмотрим распределение элементов батча. Вычтем среднее и поделим на дисперсию выборки, получив распределение с центром в 0 и дисперсией 1. Такое распределение позволит сети быстрее обучаться, т.к. все числа получатся одного порядка. Но ещё лучше ввести две переменные для каждого признака, обобщив нормализацию следующим образом (рисунки).

Получим среднее, дисперсию. Эти параметры будут входить в алгоритм обратного распространения ошибки. Тем самым получаем batch normalization слой с $2*k$ параметрами, который и будем добавлять в архитектуру предложенной сети для распознавания лиц.

На вход в задаче подаётся черно-белое изображение лица человека размером 50×50 пикселей. На выходе имеем 14000 вероятностей классов. Класс с максимальной вероятностью считается результатом предсказания.

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1\dots m}\}$;	
Parameters to be learned: γ, β	
Output: $\{y_i = \text{BN}_{\gamma,\beta}(x_i)\}$	
$\mu_{\mathcal{B}} \leftarrow \frac{1}{m} \sum_{i=1}^m x_i$	// mini-batch mean
$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m} \sum_{i=1}^m (x_i - \mu_{\mathcal{B}})^2$	// mini-batch variance
$\hat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}}$	// normalize
$y_i \leftarrow \gamma \hat{x}_i + \beta \equiv \text{BN}_{\gamma,\beta}(x_i)$	// scale and shift

Рисунок. – Обобщение нормализации

Используется 8 свёрточных слоёв, каждый размером 3x3. После каждой свёртки, используется ReLU: $\max(x, 0)$. После блока из двух свёрток идёт max-pooling с размером ячейки 2x2 (без перекрытия ячеек). Последний pooling слой имеет размер ячейки 7x7, который усредняет значения, а не берёт максимум. В итоге получается массив 1x1x320, который и подаётся на полносвязный слой.

В новой архитектуре каждый блок из двух свёрток содержит слой Batch Normalization между ними. Также пришлось удалить один свёрточный слой из второго блока, так как добавление новых слоёв увеличило расход памяти графической карты.

При этом был убран Dropout в соответствии с рекомендациями по применению BN авторов оригинальной статьи. Экспериментальная оценка.

Основная метрика — точность, делим количество правильно классифицированных изображений на количество всех изображений в тестовой выборке.

Основная сложность в оптимизации нейросети с помощью слоя Batch Normalization — подобрать learning rate и правильно его изменять в процессе обучения сети. Чтобы сеть сходилась быстрее, начальный learning rate должен быть больше, а потом снижаться, чтобы результат был точнее.

Было протестировано несколько вариантов изменения learning rate (рисунок 2).

Имя	Формула изменения learning rate	Итераций до точности 80%	Итераций до полной сходимости	Максимальная точность
<u>original</u>	$0.01 * 0.1^{\lfloor \frac{\#iter - 1}{50000} \rfloor}$	64000	420000	94,5%
<u>short step</u>	$0.055 * 0.7^{\lfloor \frac{\#iter}{11000} \rfloor}$	45000	180000	86,7%
<u>multistep without dropout</u>	$0.055 * 0.7^{Nsteps}$	45000	230000	91,3%

[x] – целая часть; #iter – номер итерации; Nsteps – шаги, заданные вручную на итерациях: 14000, 28000, 42000, 120000(x4), 160000(x4), 175000, 190000, 210000

Рисунок 2. – Сравнение скорости и точности обучения

График, демонстрирующий процесс обучения: точность на тестовой выборке в зависимости от количества выполненных итераций обучения нейронной сети:

Оригинальная сеть сходится за 420000 итераций, при этом learning rate за всё время изменяется только 2 раза на 150000-ой итерации и на 300000-ой. Такой подход был предложен автором оригинальной сети, и мои эксперименты с этой сетью показали, что этот подход оптимален.

Но если присутствует слой Batch Normalization, такой подход даёт плохие результаты — график long_step. В результате проведения тестов выяснилось, что лучших показателей можно добиться если изменять learning rate плавно, а потом сделать несколько скачков (график multistep_no_dropout). График short_step показывает, что просто плавное изменение learning rate работает хуже.

В итоге экспериментов был получен результат, что ускорить обучение можно, но точность в любом случае будет немного хуже. Если стоит задача как можно быстрее получить приемлимую точность, то BN как раз может помочь: чтобы достичь точности 80% требуется в 1,4 раза меньше времени по сравнению с оригинальной сетью (45000 итераций против 64000). Это можно использовать, например, для проектирования новых сетей и подбора параметров.

Заключение. Обучение с использованием Batch Normalization позволяет существенно сократить количество итераций и как следствие количество времени на обучение нейросети. Но так же это ведёт к понижению точности обучения. Как следствие необходимо выбирать подходит ли этот алгоритм улучшения для поставленной задачи или нет.

ЛИТЕРАТУРА

1. Нейросеть [Электронный ресурс]. – Режим доступа <https://geektimes.ru/post/101789/DES>. – Дата обращения: 24.09.2019.
2. Описание нейросетей [Электронный ресурс]. – Режим доступа <https://ru.wikipedia.org/wiki/>. – Дата обращения: 24.09.2019.
3. Описание нейросетей [Электронный ресурс]. – Режим доступа http://elibr.psuti.ru/Gorozhanina_Nejronnye_seti_uchebnoe_posobie.pdf. – Дата обращения 24.09.2019.