УДК 681.327.12.001.362

ДОРОГИЙ Я.Ю.

УСКОРЕННЫЙ АЛГОРИТМ ОБУЧЕНИЯ СВЕРТОЧНЫХ НЕЙРОННЫХ СЕТЕЙ

В статье рассмотрен вопрос построения нового алгоритма обучения для сверточных нейронных сетей. В ходе исследования предложено новое правило для вычисления ошибки нейрона.

In the article the question of designing of new training algorithm for the convolution neural networks. The study found the new rule for calculation of neuron's error.

Введение в проблему

Обучение нейронных сетей является одной из важных проблем при использовании их при решении различных проблем. Наличие качественного алгоритма обучения позволяет улучшить получаемые результаты, снизить ошибку построенной сети.

Анализ существующих решений

В 1981 году нейробиологи Торстен Визел и Девид Хабел исследовали зрительную кору головного мозга кошки и выявили, что существуют так называемые простые клетки, которые особенно сильно реагируют на прямые линии под разными углами и сложные клетки, что реагируют на движение линий в одном направлении.

Позднее Ян Лекун предложил использовать так называемые сверточные нейронные сети, как аналог зрительной коры головного мозга для распознавания изображений [1-3].

Данный тип сетей хорошо зарекомендовал себя для решения проблемы распознавания человека по фотопортрету, но задача разработки качественного алгоритма обучения для такого вида сетей является актуальной проблемой.

Цель работы

Целью данной работы является разработка ускоренного алгоритма обучения для сверточных нейронных сетей на базе классического алгоритма обратного распространения ошибки.

Архитектура классической сверточной нейронной сети

Идея сверточных нейронных сетей заключается в чередовании сверточных слоев (C-layers), субдискретизирующих слоев (S-layers) и наличии полносвязных (F-layers) слоев на выходе.

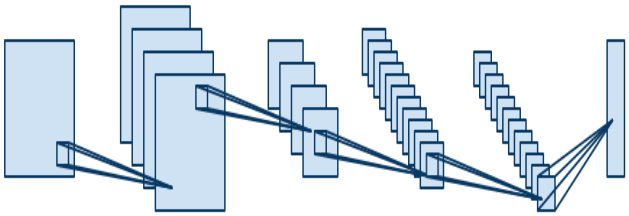

Рис. 1. Структура СНС

Такая архитектура заключает в себе 3 основных парадигмы:

- локальное восприятие;
- разделяемые веса;
- субдискретизация.

Локальное восприятие подразумевает, что на вход одного нейрона подается не всё изображение (или выходы предыдущего слоя), а лишь некоторая его область. Такой подход позволяет сохранять топологию изображения от слоя к слою.

Концепция разделяемых весов предполагает, что для большого количества связей используется небольшой набор весов. Т.е. если у нас имеется на входе изображение 32x32 пикселя, то каждый из нейронов следующего слоя примет на вход только небольшой участок этого изображения размером, к примеру, 5x5, причем каждый из фрагментов будет обработан одним и тем же набором. Важно понимать, что самих наборов весов может быть много, но каждый из них будет применен ко всему изображению. Такие наборы часто называют ядрами (kernels).

Большинство систем распознавания изображений строятся на основе двумерных фильтров. Фильтр представляет собой матрицу коэффициентов, обычно заданную вручную. Эта матрица применяется к изображению с помощью математической операции, называемой сверткой. Суть этой операции в том, что каждый фрагмент изображения умножается на матрицу (ядро) свертки поэлементно и результат суммируется и записывается в аналогичную позицию

выходного изображения. Основное свойство таких фильтров заключается в том, что значение их выхода тем больше, чем больше фрагмент изображения похож на сам фильтр. Таким образом, изображение, свернутое с неким ядром даст нам другое изображение, каждый пиксель которого будет означать степень похожести фрагмента изображения на фильтр.

Разберем более подробно процесс распространения сигнала в С-слое.

Каждый фрагмент изображения поэлементно умножается на небольшую матрицу весов (ядро), результат суммируется. Эта сумма является пикселем выходного изображения, которые и формируют карту признаков. Следует сказать, что в идеале не разные фрагменты проходят последовательно через ядро, а параллельно всё изображение проходит через идентичные ядра. Кроме того, количество ядер (наборов весов) определяется разработчиком и зависит от того какое количество признаков необходимо выделить. Еще одна особенность сверточного слоя в том, что он немного уменьшает изображение за счет краевых эффектов. Суть субдискретизации и S-слоев заключается в уменьшении пространственной размерности изображения. Т.е. входное изображение грубо (усреднением) уменьшается в заданное количество раз. Чаще всего в 2 раза, хотя может быть и не равномерное изменение, например, в 2 раза по вертикали и в 3 раза по горизонтали.

Использование субдискретизации необходимо для обеспечения инвариантности к масштабу.

Чередование слоев позволяет составлять карты признаков из карт признаков, что на практике означает способность распознавания сложных иерархий признаков.

Обычно после прохождения нескольких слоев карта признаков вырождается в вектор или даже скаляр, но таких карт признаков становится сотни. В таком виде они подаются на один-два слоя полносвязной сети. Выходной слой такой сети может иметь различные функции активации. В простейшем случае это может быть тангенциальная функция, также успешно используются радиальные базисные функции.

Адаптация алгоритма обратного распространения ошибки для сверточных нейронных сетей

Рассмотрим алгоритм обратного распространения ошибки. В алгоритме можно выделить несколько основных шагов процесса обучения. Обычно именно эти шаги выделяют при практической реализации НС и данного алгоритма обучения.

Рассматриваемая НС имеет сигмоидальную активационную функцию (1).

$$f(u) = \frac{1}{1+e^{-u}}, \quad (1)$$

где

$$u = w_0 + w_1 x_1 + ... + w_n x_n \quad (2)$$

Введем следующие обозначения: множество учебных данных $X_s = \{x_i \mid i = 1,..., N\}$ и соответствующие им ожидаемые значения $Y_s = \{y_i \mid i = 1,..., M\}$, где N – число входов НС, M – число выходов НС, $s = 1,..,S$, а S – размерность обучающей выборки. Обозначим множество значений вывода нейронов НС через $F_i = \{f_j^i\}$, где i – номер слоя, а j – номер нейрона в слое. Для входного слоя $j = 1,2,...,N$, для первого скрытого слоя – $j = 1,2,...,C_1$, второго – $j = 1,2,...,C_2$ и т.д., для последнего скрытого слоя – $j = 1,2,...,M$. Общее число слоев НС равно К и число нейронов во всех скрытых слоях будет равно $C = \{C_0, C_1, ...C_K\}$, где $C_0 = N, C_K = M$.

Алгоритм обратного распространения ошибки можно представить следующим образом:

1) Для входного слоя установим значение каждого элемента, равное соответствующему элементу входного вектора. Значение вывода каждого элемента установим равным вводу.

$$F_0 = X_1 \quad (3)$$

2) Для элементов первого слоя вычисляем совокупный ввод и вывод:

$$u_j^1 = w_{j,0}^1 + \sum_{i=1}^{N} f_i^0 w_{j,i}^1, \quad (4)$$

где

$$f_j^1 = \frac{1}{1+e^{-u_j^1}} \quad (5)$$

где $j = 1,2,..., N_1$ – номер нейрона первого скрытого слоя, $w_{j,i}^l$ – i-й весовой коэффициент j-го нейрона l–слоя НС, f_i^0 – значение i-го элемента входного слоя.

3) Повторим первый шаг для всех скрытых слоев НС, включая выходной слой нейронов, причем формулы (4) и (5) немного изменятся, и будут иметь вид:

$$u_j^k = w_{j,0}^k + \sum_{i=1}^{C_k} f_i^{c-1} w_{j,i}^c, \quad (6)$$

где

$$f_j^c = \frac{1}{1+e^{-u_j^c}}, \quad (7)$$

где $c = 2,...,K$.

4) Сравниваем значения векторов Y_0 и F_K, если разница между вектором - образцом Y_0 и реальным выводом НС F_K находится в допустимом пределе, то переходим к шагу 5, если нет, то переходим к шагу 6.

5) Для входного вектора устанавливаем значение каждого элемента, равное соответствующему элементу $X_2 (F_0 = X_2)$ и переходим вновь к шагу 2. Если же на вход НС подан последний вектор X_S, то в зависимости от того, требовалось ли на этой эпохе обучение, либо вновь подаются на вход все обучающие векторы, начиная с первого, либо считается, что НС обучена и обучение заканчивается.

6) Для каждого нейрона выходного слоя вычисляется его ошибка. Поскольку рассматривается НС с сигмоидальной активационной функцией, то значение ошибки равно

$$\delta_j^K = f_j^K (1 - f_j^K)(y_j^0 - f_j^K), \quad (8)$$

где $j = 1,2,...,M$.

7) Для предпоследнего слоя вычисляется ошибка каждого нейрона

$$\delta_i^{K-1} = f_i^{K-1}(1 - f_i^{K-1}) \sum_{j=1}^{C_K} \delta_j^K w_{i,j}^{K-1}, \quad (9)$$

где δ_j^K – значение ошибки j-го элемента *K-го* слоя, $w_{i,j}^{K-1}$ – вес связи

между i-м элементом *K-1* слоя и j-м элементом *K*-го слоя, f_i^{K-1} – значение i-го элемента *K-1*-го слоя.

8) Для всех остальных скрытых слоев ошибка вычисляется по формуле (9).

9) Далее для всех слоев обновляются значения весовых коэффициентов каждого нейрона

$$\Delta w_{i,j}^c (t+1) = \eta(\delta_i^c f_j^{c-1}) + \alpha \Delta w_{i,j}^c (t), \quad (10)$$

где η – скорость обучения, α – инерция, или влияние значения предыдущего изменения, обычно берется значение $\alpha < 1$, t – номер итерации. И тогда устанавливаются новые значения весовых коэффициентов, равные

$$w_{i,j}^c = w_{i,j}^c + \Delta w_{i,j}^c (t). \quad (11)$$

10) Для входного вектора устанавливается значение каждого нейрона, равное соответствующему элементу $X_2 (F_0 = X_2)$ и выполняется переход вновь на шаг 2. Если уже подан последний вектор X_S на вход НС, то вновь подается первый обучающий вектор.

Замечание: Если НС является сверточной (СНС), то необходимо учесть факт совместного использования весов множеством межнейронных связей. Такие связи используются в СНС между парами слоев свертки-субдискретизации, а также внутри этих пар. Для учета этих особенностей необходимо привести формулу расчета весов (10) к виду (12):

$$\Delta w_{i,j}^c (t+1) = \frac{\eta(\delta_i^c f_j^{c-1})}{N_w} + \alpha \Delta w_{i,j}^c (t), \quad (12)$$

где N_w- количество связей, что совместно используют весовой коэффициент w.

Модификация алгоритма обучения

Теперь рассмотрим значения ошибки для нейронов последнего слоя, вычисляемые на шаге 6 алгоритма обратного распространения ошибки в соответствии с формулой (8). Зависимость значения ошибки нейрона δ_j^{K-1} последнего слоя от выходного значения этого нейрона f_j^{K-1} для различных ожидаемых значений, представлена на рис. 1-3. Необходимо отметить, что форма графика будет меняться при различных ожидаемых значениях y_j^0.

В соответствии с формулой (8), значение y_j^0 может находиться в диапазоне от 0 до 1, так как сигмоидальная функция определена именно на этом диапазоне. Рассмотрим три графика для значений $y_j^0 = 1$ (рис.1), $y_j^0 = 0$ (рис.2) и $y_j^0 = 0.5$ (рис.3).

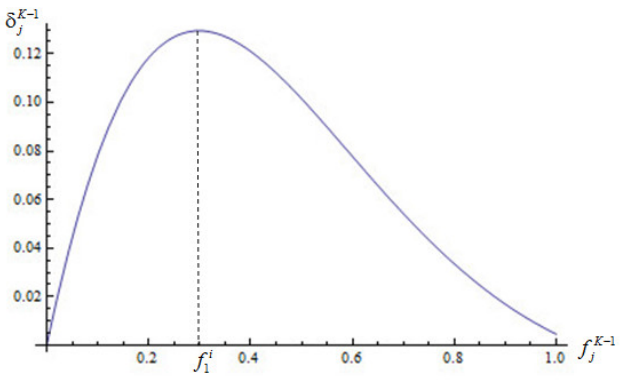

Рис.1. График ошибки при $y_j^0 = 1$

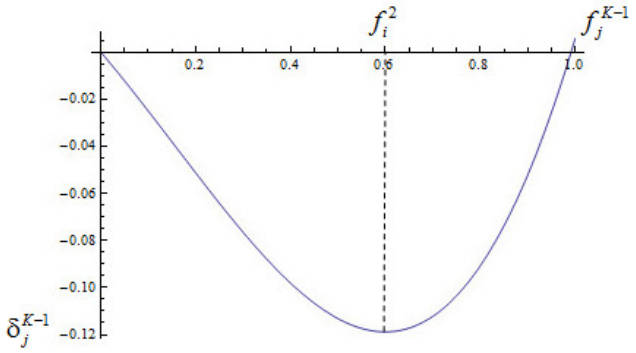

Рис.2. График ошибки при $y_j^0 = 0$

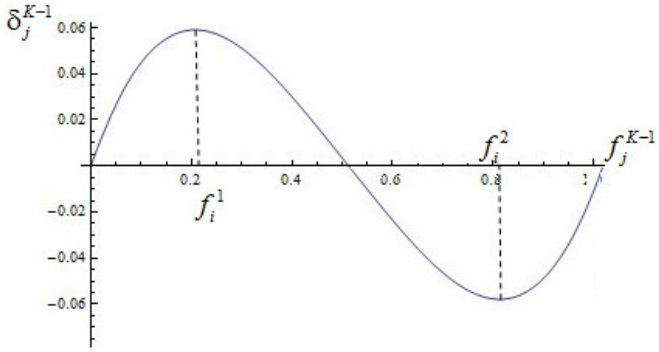

Рис.3. График ошибки при $y_j^0 = 0.5$

По графикам видно, что значение ошибки стремится к 0 в трех случаях, когда:

- значение f_j^0, полученное от НС, близко к ожидаемому y_j^0;
- значение f_j^0, полученное от НС, близко к 0;
- значение f_j^0, полученное от НС, близко к 1.

Это значит, что в случаях, когда значение f_j^0 приближается к 0 или 1, а не к ожидаемому y_j^0, процесс обучения останавливается и изменений весовых коэффициентов НС уже не происходит или происходят, но весьма незначительные. Таким образом, получается, что одна итерация обучения оказывается «потерянной» и никак не влияет на общий ход обучения.

Если вернуться к логике функционирования НС, то этот эффект в изменении значения ошибки можно объяснить. Значение ошибки, стремящееся к 0 в граничных точках графика, объясняет ее возможность обучаться даже на зашумленных данных, когда по ошибке некоторые учебные образы могли быть отнесены к неверному классу. Но необходимо помнить, что для сигмоидальной функции выполняется условие:

$$\frac{1}{1+e^{-u}} \to 0 \text{, при } u \to -\infty \qquad (13)$$

и в то же время

$$\frac{1}{1+e^{-u}} \to 0 \text{, при } u \to +\infty \qquad (14)$$

Из этого следует, что значение ошибки может быть сколь угодно малым, но не станет равным 0. Следовательно, остается возможность «доучить» НС и на этот учебный образ. Однако чрезвычайно малые значения, которые может принимать ошибка в этих точках, означают, что на обучение может потребоваться значительное время, а в зависимости от размеров НС это могут быть часы или даже дни.

Для исключения подобной ситуации предлагается произвести модернизацию алгоритма, ускорив процесс обучения. Для этого изменим функцию (8) определения ошибки нейронов последнего слоя. Теперь значение ошибки δ_i будет стремиться к 0 только в том случае, когда полученное на выходе нейрона значение будет близко к ожидаемому.

Точки локальных экстремумов на диапазоне [0,1] можно найти как значения первой производной по функции (8) на этом диапазоне. Это будут точки:

$$f_i^{'1} = \frac{1}{3} + \frac{1}{3}y_i - \frac{1}{3}\sqrt{1 - y_i + y_i^2}, \qquad (15)$$

$$f_i^{'2} = \frac{1}{3} + \frac{1}{3}y_i + \frac{1}{3}\sqrt{1 - y_i + y_i^2}, \qquad (16)$$

где y_i – ожидаемое значение от i-го нейрона последнего слоя.

Теперь формула (8), применяемая на шестом шаге обучения НС, для вычисления ошибки нейрона изменит свой вид и примет следующую форму:

$$\delta_i = \begin{cases} f_i^{'1}(1 - f_i^{'1})(y_i - f_i^{'1}) \, при \, f_i < f_i^{'1} \\ f_i(1 - f_i)(y_i - f_i) \, при \, f_i^{'2} < f_i < f_i^{'1} \\ f_i^{'2}(1 - f_i^{'2})(y_i - f_i^{'2}) \, при \, f_i > f_i^{'2} \end{cases}, \qquad (17)$$

где f_i – полученное значение j-го элемента выходного слоя; $f_i^{'1}, f_i^{'2}$ – точки локальных экстремумов для ожидаемого значения y_i на диапазоне [0,1].

Эксперименты и результаты

Проведенные эксперименты показали, что при использовании данного модифицированного алгоритма время, необходимое для обучения НС, уменьшается на 24%. Это связано с тем, что количество выполняемых элементарных операций данного алгоритма увеличивается на 17 для каждого исходящего нейрона, а число эпох обучения, необходимых для тренировки сети, сокращается на 32%.

Выводы

В ходе исследования был разработан новый алгоритм обучения для сверточных нейронных сетей. Также, проведен ряд экспериментов для определения работоспособности предложенного алгоритма и получено значительное повышение скорости обучения нейронных сетей.

Список литературы

1. LeCun Y. A theoretical framework for backpropagation // Proc. of IEEE. - 1998. - P.21-28.
2. LeCun Y., Bottou L., Bengio Y., Haffne P. Gradient-Based Learning Applied to Document Recognition // Proc. IEEE. - 1998. - P.59-67.
3. Дорогий Я.Ю. Модифицированный алгоритм обучения сверточных нейронных сетей. //Сборник материалов. VII Международная научно-практическая конференция “перспективы развития информационных технологий”. 18 апреля 2012, г. Новосибирск: Издательство НГТУ. – с 34-38.