Нейросетевое распознавание рукописных символов на изображениях низкого качества

С. А. Комков (МГУ имени М. В. Ломоносова, Москва)

В данной работе решена задача построения сверточной нейронной сети, способной распознавать рукописные символы на сильно зашумленных изображениях с точностью, сопоставимой с человеческой. При этом обучение классификатора происходит по размеченной базе сильно зашумленных изображений, в которой 5% обучающих примеров размечено неправильно.

Ключевые слова: сверточные нейронные сети, распознавание изображений, машинное обучение, обучение с учителем.

Введение.

Стандартная искусственная нейронная сеть прямого распространения — это система, состоящая из нескольких слоев взаимосвязанных искусственных нейронов. Каждый нейрон принимает вектор выходных сигналов всех нейронов предшествующего слоя и скалярно умножает его на собственный вектор весов. К полученному числу в нейроне применяется функция активации, после чего результат поступает на входы ко всем нейронам следующего слоя. Таким образом, входной слой сети обнаруживает набор примитивных шаблонов поступающих данных, второй слой обнаруживает закономерности шаблонов и т.д.

Сверточная нейронная сеть — это особый вид искусственных нейронных сетей. Она состоит из одного или нескольких сверточных слоев (иногда со слоями подвыборки), за которыми следуют полносвязные слои как в обычной нейронной сети. Архитектура сверточных нейронных сетей мотивирована открытием механизма работы визуальной коры головного мозга. В коре содержится много клеток-рецепторов, которые

отвечают за детектирование света в маленьких перекрывающихся областях визуального поля, а более сложные клетки обрабатывают сигналы, поступающие с этих рецепторов.

Сверточные нейронные сети показывают отличные результаты при обработке данных с пространственной структурой по нескольким причинам:

- устойчивость к сдвигам и поворотам объекта на изображении, а так же устойчивость к шумам;
- учет пространственной структуры входных признаков;
- меньшее количество оптимизируемых параметров относительно классических полносвязных сетей;
- более быстрое и качественное обучение относительно обучения полносвязных сетей.

Наиболее известной классической сверточной нейронной сетью является LeNet-5 французского информатика Яна ЛеКуна [7]. Данная сеть обучалась и тестировалась по базе качественных рукописных изображений MNIST [8]. На тестовой выборке сеть верно классифицировала более 99% символов, что сравнимо с человеческой точностью.

Слои сверточной нейронной сети.

Сверточный слой. Входом слоя являются D матриц размера $N \times M$. Сверточный слой может быть как входным слоем сети, так и скрытым слоем. В случае, если сверточный слой является входным, то $N \times M$ — размер изображения, а D — количество цветовых каналов изображения. Входные импульсы сворачиваются T ядрами размера $k \times k \times D$ каждое. Свертка слоя одним ядром производит один выходной признак. Начиная с левого верхнего угла, ядро перемещается по изображению, пока не дойдет до правой границы. Тогда начальное положение ядра смещается вниз, и ядро снова начинает движение вправо. Таким образом, на выходе слоя образуются T матриц размера $(N - k + 1) \times (M - k + 1)$, где значение на месте с координатой (i, j) матрицы под номером k — это результат свертки k-го ядра с входным изображением в ситуации, когда левый верхний угол ядра имеет координаты (i, j). Полученные значения могут быть поданы на вход следующего сверточного слоя.

Слой подвыборки. На слое подвыборки каждый канал входа разбивается на непересекающиеся квадраты размера r на r. Из всех значений каждого квадрата на следующий слой подается только максимальное значение. Таким образом, если вход слоя подвыборки состоит из D матриц $N \times M$, то на выходе будет D матриц размера $(N/r) \times (M/r)$. Данный слой делает сеть более устойчивой к шуму и уменьшает количество весовых коэффициентов для оптимизации. Использование слоя подвыборки мотивировано тем, что для сети важно само наличие признака, а не его точное положение.

Нелинейный слой активации. На данных слоях внутри сети ко всем значениям входа применяется нелинейная функция активации, и результат подается на выход. Таким образом, слой активации не меняет размер входа. Наиболее популярной нелинейной функцией активации является ReLU функция: $\mathrm{ReLU}(x) = \max(0, x)$. В качестве функции активации в работе использовалась модернизированная ReLU функция: $\mathrm{PReLU}(x) = \max(a, x)$, где параметр a автоматически подбирается при обучении модели. Данный подход позволяет улучшить результаты сети за счет подбора оптимальной функции активации для каждого нейрона [3].

Выходной слой построенной сети состоит из 67 нейронов, по количеству возможных классов. На этом слое применяется функция активации Softmax. Данная функция преобразовывает выход j-го нейрона равный z_j по формуле $\sigma_j(z) = \frac{e^{z_j}}{\sum\limits_{k=1}^{67} e^{z_k}}$. Таким образом, на выходе сети получаются оценки вероятности того, что был подан соответствующий класс.

Полносвязный слой. В полносвязном слое на вход каждому нейрону подаются все выходы предшествующего слоя. Соответствующие веса для входов в каждом нейроне, величина сдвига и параметр a в функции активации PReLU подбираются автоматически при обучении сети.

Дропаут слой. Дропаут слой задается параметром p, который равен вероятности, с которой каждый вход не будет передан на выход в течение одной итерации обучения сети. Таким образом, при обучении на каждой итерации часть нейронов выключается из процесса, и веса меняются только у оставшихся нейронов. При распознавании с помощью сети в работе участвуют уже все нейроны. Так как выход слоя при обучении имел меньший размер, то при распознавании все значения входа

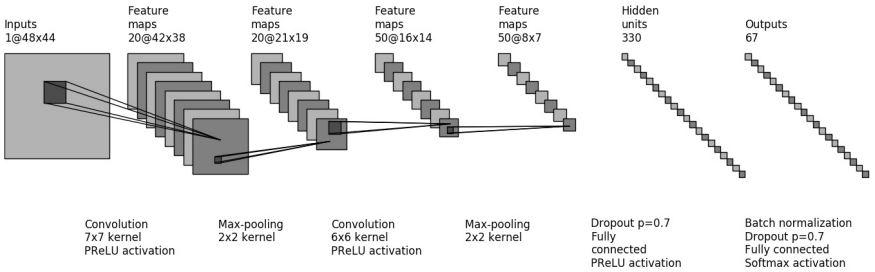

Рис. 1: Архитектура сети.

дропаут слоя умножаются на число $(1 - p)$ и переходят на выход. Данный слой уменьшает время одной эпохи обучения в связи с меньшим число оптимизируемых параметров, а также позволяет лучше бороться с переобучением сети относительно стандартных методов регуляризации [10].

Нормализующий слой. На данном слое от всех входов отнимается выборочное среднее значений на входах, и результат делится на корень выборочной дисперсии. Выборочные величины вычисляются с учетом значений на входах данного слоя на предыдущих итерациях обучения. Данный подход позволяет увеличить скорость обучения сети и улучшить итоговый результат [4].

Архитектура сети.

Архитектура построенной сети представлена на рис. 1.

Исходное монохромное изображение имеет размеры 48 на 44. Первым слоем идет свертка изображения 20 ядрами размера $7\times7\times1$ и применение функции PReLU к получившимся значениям. Получаем 20 изображений 42 на 38.

Далее слой подвыборки разбивает каждое из 20 изображений на непересекающиеся квадраты 2 на 2 и оставляет только максимальное значение из каждого квадрата. Таким образом, на выходе первого слоя подвыборки получаются 20 изображений 21 на 19.

Затем идет свертка полученных изображений 50 ядрами размера $6 \times 6 \times 20$ и применение к полученным значениям функции PReLU. На выходе получается 50 изображений 16 на 14.

Второй слой подвыборки аналогично первому возвращает 50 изображений 8 на 7.

После, значения подаются на дропаут слой с параметром $p = 0.7$. После дропаут слоя идет полносвязный слой из 330 нейронов с функцией активацией PReLU.

Все выходы первого полносвязного слоя подаются на нормализующий слой, а после него на дропаут слой с параметром $p = 0.7$. В конце идет полносвязный слой с 67 нейронами, по количеству возможных классов, каждый из которых распознает определенный класс. К выходам последнего полносвязного слоя применяется функция активации Softmax. Итоговым классом для изображения предсказывается тот класс, у которого оценка вероятности наибольшая.

База изображений.

Имеется размеченная база сильно зашумленных монохромных изображений рукописных символов [5] со следующими свойствами:

- размер изображений — 48 на 44 пикселей;
- символы на изображениях принадлежат одному из 67 классов: 33 класса, соответствующие символам кириллицы (прописные и строчные буквы определяются в один класс), 30 классов, соответствующие числам с 1 по 30, 4 класса, соответствующие запятым, точкам, пробелам и символам процента;
- тренировочная подвыборка состоит из 3600 изображений;
- тестирование построенной модели проводится по 900 изображениям, не участвовавшим в обучении;
- изображения различных классов равномерно распределены по тренировочной и тестовой подвыборкам;
- 5% изображений тренировочной подвыборки размечены неправильно;
- на изображениях присутствуют артефакты в виде границ ячеек, клякс или утерянной части изображения, часть символов выходит за пределы изображения;

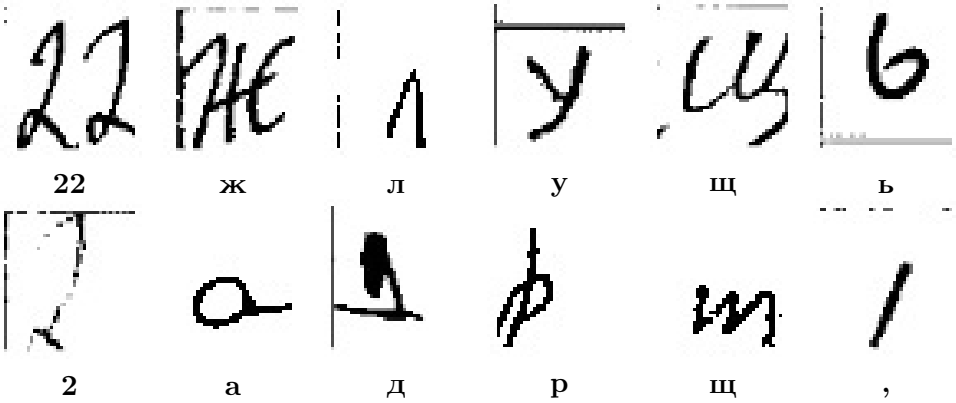

Рис. 2: Изображения тренировочной подвыборки и их классы.

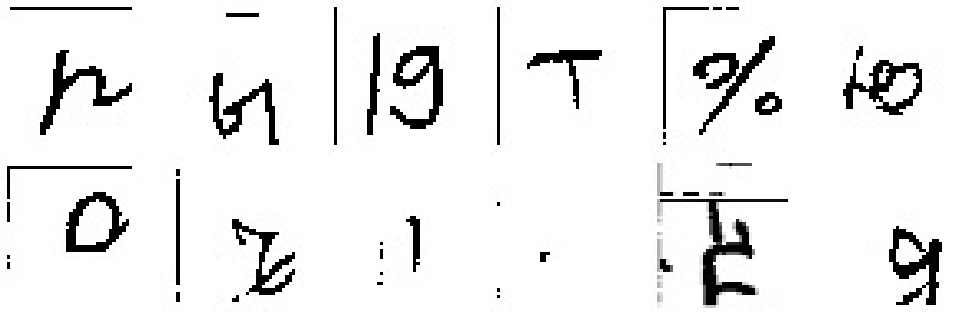

Рис. 3: Изображения тестовой подвыборки.

- примеры изображений из тренировочной и тестовой подвыборок представлены на рис. 2 и рис. 3.

Таким образом, исследуется способность сверточных нейронных сетей обобщать образы классов при обучении по выборке низкого качества с выбросами.

Обучение нейронной сети.

Современные методы обучения нейронных сетей базируются на классическом методе обратного распространения ошибки (параграф 4.3 из [1]). Основная идея этого метода состоит в распространении сигналов ошибки от выходов сети к её входам, в направлении, обратном прямому распространению сигналов в обычном режиме работы. Для этого вво-

дится функция потерь, зависящая, в частности, от всех весов нейронной сети. Таким образом, на каждом слое вычисляется градиент функции потерь, и веса данного слоя меняются в направлении противоположным градиенту.

Для мультиклассовой классификации при обучении сети в качестве функции потерь в методе обратного распространения ошибки используется категориальная кросс-энтропия. В работе веса представленной нейронной сети при обучении меняются каждый раз после обработки 30 изображений. В качестве метода обучения в работе используется адаптивный метод Nadam [2], полученный добавлением ускоренного градиента Нестерова [9] в адаптивный метод обучения Adam [6].

Ускоренный градиент Нестерова вычисляет по следующей формуле:

$$g_t := \gamma g_{t-1} + \alpha \bigtriangledown_\theta f(\theta - \gamma g_{t-1}),$$

где g_t — градиент Нестерова в момент времени t, θ — вектор весов нейронной сети, α — скорость обучения, γ — импульс обучения, а $f(\cdot)$ — функция потерь. Данный прием придает импульс процессу обучения нейронной сети, что позволяет меньше застревать в точках локального минимума.

Идея адаптивных методов заключается в понижении скорости обучения только тех весов нейронной сети, которые обучаются интенсивнее всего. Для этого для каждого параметра нейронной сети вычисляет некоторое число, характеризующее интенсивность обучения этого параметра. Преимущество метода Adam перед другими адаптивными методами заключается в универсальной начальной инициализации параметров этого метода, которая показывает отличные результаты на нейронных сетях различных архитектур. Прочие методы обучения требуют большего количества экспериментов и более чувствительны к изменениям архитектуры сети. Изменение весов методом Adam при рекомендуемой инициализации задается следующими формулами:

$$g_t := \bigtriangledown_\theta f(\theta_{t-1}),$$

$$m_t := 0.9 m_{t-1} + 0.1 g_t,$$

$$v_t^i := 0.999 v_{t-1}^i + 0.001 {g_t^i}^2,$$

$$\hat{m_t} := m_t / (1 - 0.9^t),$$

$$\hat{v_t} := v_t / (1 - 0.999^t),$$

$$\theta_t^i := \theta_{t-1}^i - \alpha \hat{m}_t{}^i / (\sqrt{\hat{v}_t{}^i} + 10^{-8}),$$

где $f(\cdot)$ — функция потерь, θ_t — вектор весов нейронной сети в момент времени t, g_t — градиент функции потерь в момент времени t, m_t — импульс движения в момент времени t, v_t — вектор интенсивности обучения весов нейронной сети в момент времени t, а α — скорость обучения. Таким образом, для весов нейронной сети, для которых на предшествующих итерациях соответствующее значение градиента было велико, будет уменьшаться скорость обучения.

Результаты и выводы.

Результаты тестирования построенной нейронной сети представлены в табл. 1. Так же по описанной базе изображений была обучена и протестирована сверточная нейронная сеть LeNet-5, о которой говорилось в введении, и LeNet-5 с функцией активации ReLU вместо сигмоиды. Дополнительно, тестовая подвыборка была полностью размечена человеком.

Классификатор	**Процент совпадений**
LeNet-5	61.667
LeNet-5 + ReLU activation	67.556
Представленная сверточная нейронная сеть	80.556
Человек	84.889

Таблица 1: Результаты тестирования классификаторов.

Таким образом, построенная сверточная нейронная сеть показывает значительно лучшие результаты по сравнению с каноничной архитектурой сверточных нейронных сетей. При этом результаты классификации сравнимы с человеческой классификацией символов на изображениях. Видно, что предложенные методы по улучшению качества нейросетевого распознавания вносят ощутимый вклад в способность нейронной сети предсказывать верные значения.

Список литературы

[1] Хайкин С. Нейронные сети: полный курс, 2-е издание. – Издательский дом Вильямс, 2008.

[2] Dozat T. Incorporating Nesterov momentum into Adam. – Stanford University, Tech. Rep., 2015.[Online]. Available: http://cs229. stanford. edu/proj2015/054 report. pdf, 2015.

[3] He K. et al. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification //Proceedings of the IEEE international conference on computer vision. – 2015. – С. 1026-1034.

[4] Ioffe S., Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift //arXiv preprint arXiv:1502.03167. – 2015.

[5] Kaggle in Class [Электронный ресурс] : Handwritten symbols recognition (CMF). — Электрон. дан. (4 файла). — San Francisco : 2010 — Режим доступа: https://inclass.kaggle.com/c/handwritten-symbols-recognition-cmf, свободный. — Загл. с экрана

[6] Kingma D., Ba J. Adam: A method for stochastic optimization //arXiv preprint arXiv:1412.6980. – 2014.

[7] LeCun Y. et al. Gradient-based learning applied to document recognition //Proceedings of the IEEE. – 1998. – Т. 86. – №. 11. – С. 2278-2324.

[8] LeCun Y., Cortes C., Burges C. J. C. The MNIST database of handwritten digits. – 1998.

[9] Nesterov Y. A method of solving a convex programming problem with convergence rate O (1/k2) //Soviet Mathematics Doklady. – 1983. – Т. 27. – №. 2. – С. 372-376.

[10] Srivastava N. et al. Dropout: a simple way to prevent neural networks from overfitting //Journal of Machine Learning Research. – 2014. – Т. 15. – №. 1. – С. 1929-1958.