

УДК 004.415.25

DOI: 10.18503/2306-2053-2020-8-1-15-22

АНАЛИЗ ПОНЯТИЙ НЕЙРОННАЯ СЕТЬ И СВЕРТОЧНАЯ НЕЙРОННАЯ СЕТЬ, ОБУЧЕНИЕ СВЕРТОЧНОЙ НЕЙРОСЕТИ ПРИ ПОМОЩИ МОДУЛЯ TENSORFLOW*Багаев И.И.*

Аннотация. В статье проведен анализ перехода внимания научного сообщества от основного средства для распознавания образов при помощи методов машинного обучения без применения нейронных сетей к сверточным нейронным сетям как к наилучшему способу для классификации объектов на изображениях. Рассмотрены подробно понятия нейронной сети и сверточной нейронной сети, приведены примеры однослойной, двухслойной нейронных сетей, показан принцип, при помощи которого можно создавать нейронную сеть любой глубины, проведен анализ отличительных особенностей сверточных нейросетей, разобраны типы слоев, которые может иметь сверточная нейросеть. Проанализирован процесс обучения нейронной сети на наборе данных Fashion MNIST при использовании программного пакета TensorFlow от Google, выбрана структура собственной сверточной нейронной сети, нейросеть обучена для распознавания различных видов одежды. Результат работы модели на тестовом наборе данных Fashion MNIST составил 90%. Анализ пакета TensorFlow показал простоту API высокого уровня, предоставляемое модулем, позволяющее строить, конфигурировать и обучать нейросети любой сложности, включая сверточные нейронные сети, что делает TensorFlow удобным для интеграции и использования в собственных разработках.

Ключевые слова: нейронная сеть, сверточная нейронная сеть, GPU, CPU, нейрон, функция активации, TensorFlow, функция потерь, метрика качества.

Введение

Сверточная нейронная сеть (англ. – CNN или ConvNet) является частным случаем искусственных нейронных сетей глубокого обучения (англ. – deep learning) [1]. Архитектура сверточных сетей была предложена Яном Лекуном [2] в 1988 году с целью повышения эффективности распознавания образов, но, как и другие искусственные нейронные сети, требовала вычислительной мощности, которой на момент появления обеспечить не представлялось возможным, вследствие чего интерес научного сообщества к нейронным сетям был снижен вплоть до появления реализаций на графическом процессоре (англ. graphics processing unit или GPU) [3].

Помимо использования особенностей архитектуры GPU, ориентированной на многопоточное исполнение [4], в качестве основы для нейронных сетей требовалось также эффективная реализация, которая показала бы преимущество перед центральными процессорами (англ. central processing unit или CPU), мощности которых долгое время безальтернативно использовались при работе с нейронными сетями: в силу вычислительных и архитектурных ограничений CPU не давал необходимо уровня производительности. Сравнительное противопоставление с графическими ускорителями произошло много позднее. В работе О. К. и Юнга К. [3] была показана имплементация искусственной нейронной сети на графическом ускорителе с итоговым быстродействием, превышающим аналогичную на центральном процессоре в 20 раз, что вновь повысило внимание к нейронным сетям.

Тем не менее в распознавании образов вплоть до победы сверточной нейронной сети AlexNet в конкурсе ImageNet 2012 года [5], преимущественно использовались классические методы машинного обучения без использования нейронных сетей. В последующие годы лидерство как в соревнованиях ImageNet, так и в целом в компьютерном зрении захватили сверточные нейронные сети глубокого обучения.

Таким образом при постановке задач классификация объектов на изображении эффективней всего использовать сверточные нейронные сети. Необходимо учитывать, что перед использованием искусственной нейронной сети, требуется решить проблему конфигурации и настроить её гиперпараметры: количество слоев, количество нейронов в каждом из слоев, а также выбрать функции активации, в которые будет подаваться результат слоя перед передачей на следующий слой.

Рассмотрим, чем является нейронная сеть в общем понимании, а также особенности сверточных нейронных сетей.

Понятие нейронной сети

Нейронная сеть – универсальная модель, способная аппроксимировать любые поверхности. Соответствующую теорему сформулировал в 1957 году Андрей Николаевич Колмогоров [6]: каждая непрерывная функция $a(x)$, заданная на единичном кубе d -мерного пространства, представима в виде:

$$a(x) = \sum_{i=1}^{2d+1} \sigma_i \left(\sum_{j=1}^d f_{ij}(x_j) \right), \tag{1}$$

где $x = [x_1, \dots, x_d]$ – вектор описания объекта, функции $\sigma_i(\cdot)$ и $f_{ij}(\cdot)$ являются непрерывными, f_{ij} не зависят от выбора a .

Согласно тексту теоремы, функция $a(x)$ определена только на единичном кубе, следовательно, все элементы выборки должны лежать в нем: признаки всегда можно отмасштабировать так, чтобы на обучающей выборке каждый из них принимал значения из отрезка $[0; 1]$. Важно отметить, что в теореме не указан вид функций σ_i и f_{ij} . Проблема выбора гиперпараметров нейронной сети все еще остается сложной задачей, и для достижения высокого уровня метрик качества требует ручного подхода к каждой конкретной выборке. С понятиями функции активации и виды функций активации можно ознакомиться в работе [7].

Однослойная нейронная сеть представляет собой нейрон и определяется следующим выражением:

$$a(x, w) = \sigma(w^T x) = \sigma \left(\sum_{j=1}^d w_j^{(1)} x_j + w_0^{(1)} \right), \tag{2}$$

где σ – функция активации, w – вектор параметров (весов), x – вектор описания объекта. Функция активации должна быть непрерывной, монотонной и, желательно, дифференцируемой функцией. Для удобства $x \in R^{d+1}$ дополнен постоянным на всех объектах признаком $x_0 = 1$. Соответствующий вклад в скалярное произведение $w^T x$ равен w_0 .

Нейрон можно изобразить в виде вершины графа (рис. 1): он характеризуется своей функцией активации, имеет множество входов и один выход.

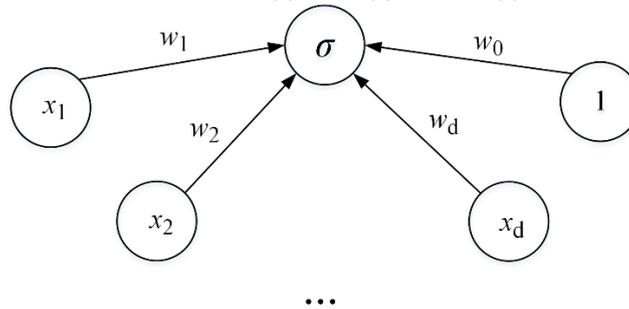


Рис. 1. Однослойная нейронная сеть как единственный нейрон

Пределом применимости однослойных нейронных является линейно разделимые выборки.

Двухслойная нейронная сеть – это линейная комбинация из D -нейронов (однослойных нейронных сетей):

$$a(x, w) = \sigma^{(2)} \left(\sum_{i=1}^D w_i^{(2)} * \sigma^{(1)} \left(\sum_{j=1}^d w_{ji}^{(1)} x_j + w_{0i}^{(1)} \right) + w_0^{(2)} \right) \tag{3}$$

Это выражение можно представить в векторном виде:

$$a(x, w) = \sigma^{(2)} \left(w^{T(2)} \sigma^{(1)} \left(\left[w^{T(1)} x, \dots, w_D^{T(1)} x \right] \right) \right) \quad (4)$$

Вектор w параметров нейронной сети получается соединением всех параметров нейронной сети на всех слоях:

$$w = \{W^{(1)}, w^{(2)}\} \quad (5)$$

где:

$$\begin{aligned} W^{(1)} &= [w_0^{(1)}, w_1^{(1)}, \dots, w_d^{(1)}]^T \in R^{(d+1)*D}, \\ w_i^{(1)} &= [w_{0i}^{(1)}, w_{1i}^{(1)}, \dots, w_{di}^{(1)}]^T \in R^{d+1}, \\ w^{(2)} &= [w_0^{(2)}, w_1^{(2)}, \dots, w_d^{(2)}]^T \in R^{D+1}. \end{aligned}$$

На рис. 2 приведено изображение двухслойной нейронной сети с помощью направленного графа.

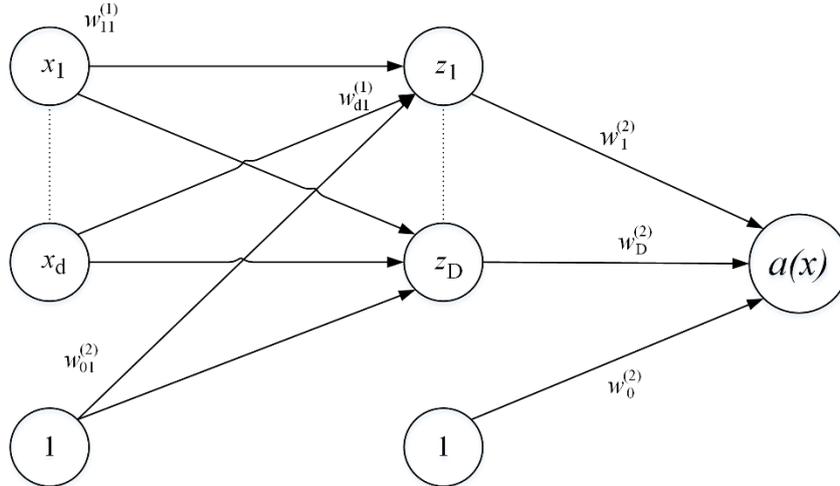


Рис. 2. Граф двухслойной нейронной сети

Таким же образом можно получить нейронные сети, имеющие большее количество слоев.

Сверточные нейронные сети (CNN)

Сверточные нейронные сети созданы для работы с изображениями, в более редких случаях используются для других видов информации [8, 9]. Сверточные нейросети являются частью технологий глубокого обучения. Работа сверточных нейросетей имеет схожие черты с функциональностью коры больших полушарий головного мозга, используемой для обработки визуальных образов [10]: в зрительной коре присутствуют простые клетки, которые реагируют на границы определенной ориентации, т.е. на прямые линии, расположенные под разными углами: горизонтально, вертикально и по диагонали, а также – сложные клетки, которые включаются в работу после активации определенного набора простых клеток. Устройство сверточной нейронной сети имеет схожую структуру: сверточные слои (англ. – convolutional layer) чередуются со слоями подвыборки или, другими словами, избирательными слоями (англ. – pooling layers). В нейронной сети AlexNet, о которой говорилось выше, применялась также операция, известная как dropout [11].

Рассмотрим структуру сверточной нейронной сети и приведенные выше понятия.

В качестве примера возьмем сверточную нейронную сеть с пятью слоями (рис. 3), где два сверточных слоя чередуются с двумя слоями подвыборки, последним слоем является полносвязный слой.

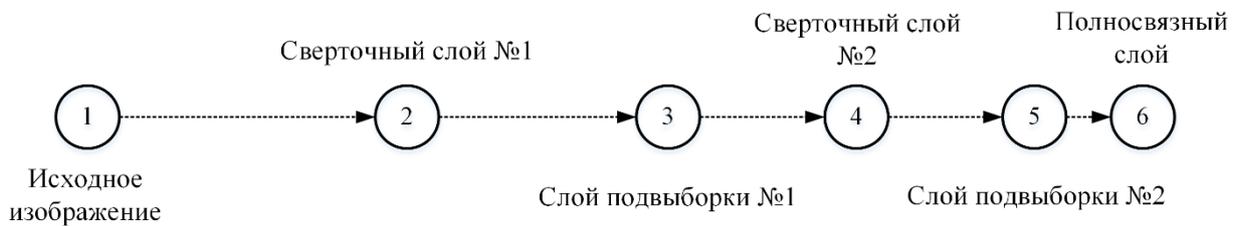
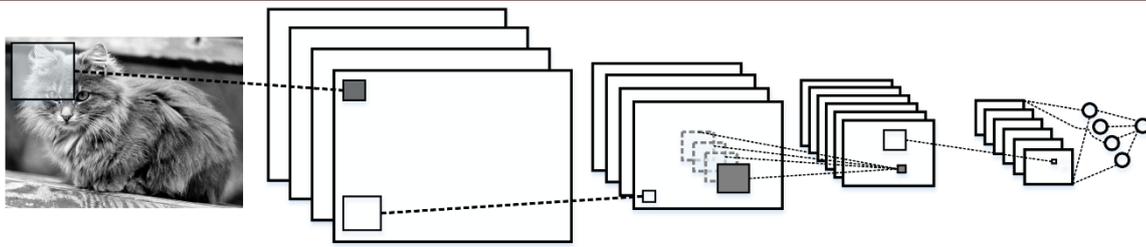


Рис. 3. Структура сверточной нейронной сети

Сверточный слой – это слой, на котором при помощи операции математической свертки заданное количество ядер или окон фильтров (на рис. 3 на первом сверточном слое находятся четыре фильтра) в виде матриц также заданного размера проходят построчно по входящей матрице, формируя новую матрицу. Ядро фильтра – матрица весов, которая путем свертки извлекает признаки из пришедших данных. Весовые коэффициенты ядра свертки устанавливаются в процессе обучения и изначально не определены. Фильтр – это объединение нескольких ядер.

Количество фильтров на сверточном слое задается на основе исходной задачи и требований к её решению: чем фильтров больше, тем выше качество распознавание и ниже быстроедействие.

Обработка границ исходной матрицы может осуществляться следующими способами (рис. 4): результирующая матрица может быть меньше исходной матрицы (рис. 4, 1), результирующая матрица может быть такого же размера (рис. 4, 2), либо результирующая матрица может быть большего размера (рис. 4, 3). Метод обработки выбирается при построении нейронной сети.

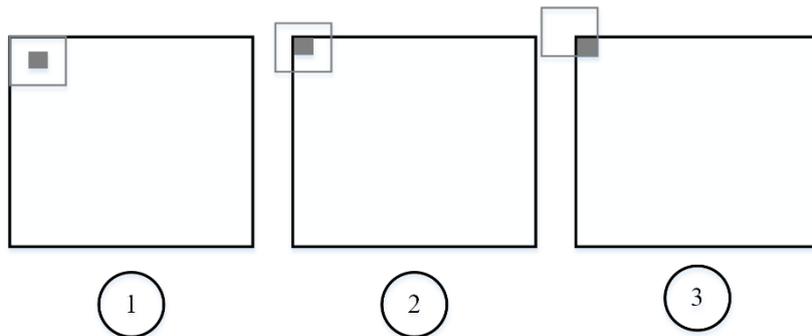


Рис. 4. Виды свертки матрицы

Подвыборочный слой – это слой, который имеет тоже самое количество фильтров, что и предыдущий сверточный слой. Задача слоя – уменьшить размерность фильтров предыдущего слоя. Это делается для того, чтобы не анализировать одну и ту же информацию повторно: ядра фильтров сверточного слоя уже учли некоторую часть информации в своих весовых коэффициентах, поэтому далее следует анализировать более абстрактные признаки следующего уровня.

Ядра фильтров подвыборочного слоя проходят по входящей матрице похожим со сверткой образом за исключением того, что области, по которым движется ядро, не пересекаются между собой и применяется не операция свертки, а, к примеру, функция максимума (англ. – max pooling) [12]: выбирается максимальный элемент в той части исходной матрицы, на которое направлено в данный момент ядро фильтра (рис. 5).

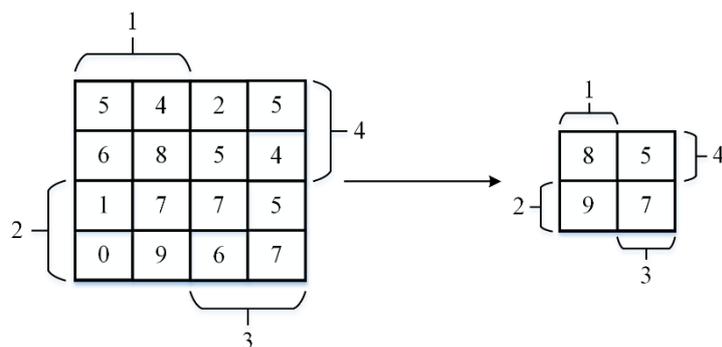


Рис. 5. Операция max pooling

Последний слой – это полносвязная нейронная сеть или персептрон [13], который также может состоять из нескольких слоев. Слой предназначен для классификации: сколько классов изображений закладывается разработчиком нейросети, столько нейронов на последнем слое этого слоя необходимо задать.

После проведенного анализа понятий нейронная сеть и сверточная нейросеть следует перейти к обучению последней.

Обучение сверточной нейронной сети при помощи программного модуля TensorFlow от Google

Для оценки порога вхождения, который необходим разработчику нейронной сети, и сложности API, предоставляемого TensorFlow, требуется провести анализ примеров, которые находятся на официальном сайте проекта, и далее обучить собственную сверточную нейронную сеть.

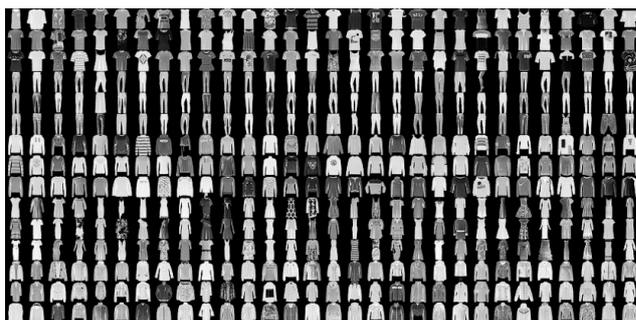


Рис. 7. Часть изображений, входящих в выборку Fashion MNIST

Для того что бы воспользоваться TensorFlow необходимо иметь python, ipython-notebook и, соответственно, установленный модуль TensorFlow.

Для формирования обучающей и тестовой выборки используется набор данных Fashion MNIST (рис. 7), который содержит 70 тысяч изображений размером 28 на 28 пикселей в десяти категориях одежды.

Для формирования обучающей и тестовой выборки используется набор данных Fashion MNIST (рис. 7), который содержит 70 тысяч изображений размером 28 на 28 пикселей в десяти категориях одежды.

Для обучающей выборки подготовлено 60 тысяч изображений, для тестовой выборки, при помощи которой проверяется качество обучения, предоставлено 10 тысяч изображений. Доступ к исходному набору данных Fashion MNIST получается путем вызова метода `datasets.fashion_mnist` в `ipython-notebook`, воспользовавшись подмодулем TensorFlow под названием `keras`.

После загрузки данной выборки имеется четыре массива: обучающие данные и метки классов и, соответственно, тестовые данные и метки классов для тестовых данных. Всего классов в выборке имеется десять классов (рис. 8).

Изображения одежды перед обучением нейросети нужно подготовить: каждое изображение в выборке представлено в оттенках серого цвета (рис. 9) на отрезке яркости [0; 255], поэтому перед тем, как отправлять данные в нейросеть, необходимо изменить масштаб, поделив исходную яркость пикселя на 255, для приведения значения точки в отрезок [0; 1].

Когда данные подготовлены, необходимо настроить слои – сверточная нейросеть со структурой, которая была приведена на рис. 3, взята за основу за исключением того, что изменено кол-во

| Label | Class |
|-------|-------------|
| 0 | T-shirt/top |
| 1 | Trouser |
| 2 | Pullover |
| 3 | Dress |
| 4 | Coat |
| 5 | Sandal |
| 6 | Shirt |
| 7 | Sneaker |
| 8 | Bag |
| 9 | Ankle boot |

Рис. 8. Метки классов набора данных Fashion MNIST

фильтров на сверточных слоях для уменьшения итоговой ошибки классификации. Таким образом будет использовано пять основных слоев и один дополнительный вспомогательный слой. Описание слоев приведено в табл. 1.

Таблица 1

Структура сверточной нейронной сети для распознавания видов одежды

| № слоя | Тип слоя | Кол-во фильтров/нейронов | Размер ядра | Функция активации/преобразования |
|--------|----------------|--------------------------|-------------|----------------------------------|
| 1 | Сверточный | 64 | 5x5 | Relu |
| 2 | Подвыборки | – | 2x2 | MaxPooling |
| 3 | Сверточный | 128 | 5x5 | Relu |
| 4 | Подвыборки | – | 2x2 | MaxPooling |
| 5 | Преобразование | – | – | – |
| 6 | Полносвязный | 10 | – | Softmax |

После выбора структуры модели в TensorFlow необходимо указать функцию потерь, минимизация которой будет происходить в процессе обучения, выбрать оптимизатор, который будет указывать, каким образом следует обновляться модели на основе исходных данных и функции потерь. Также требуется указать метрику качества, по которой можно будет судить о результате обучения модели.

Перед обучением собственной сверточной нейронной сети в качестве функции потерь и оптимизатора выбрано то, что предложено в официальных примерах, а именно функция потерь `categorical_crossentropy` [14] и оптимизатор `adam` [16]. За метрику качества будет взята метрика `assuagasy` – доля правильных ответов классификации.

После компиляции модели с указанными выше параметрами необходимо запустить обучение модели, для это следует указать данные обучающей выборки и количество эпох или количество итераций, проходя которые нейросеть будет улучшать точность распознавания, опираясь на результат предыдущей итерации. Для обучения выбрано стандартное число эпох, а именно десять.

Спустя десять итераций обучения доля правильных ответов на обучающей выборке составила 0.9602 или 96%. После окончания обучения необходимо проверить работу модели на заранее отделенной тестовой выборке: точность на проверочных данных составила 0.90 или 90%. Более низкая доля правильных ответов на тестовой выборке говорит о том, что модель переобучилась (англ. *overfitting*). Переобучение возникает в случаях, когда модель показывает худший результат на данных, которые не были использованы в обучении. Существуют методы борьбы с переобучением [12], но они не рассматриваются в рамках данной работы.

Заключение

При работе над данной статьей были проанализированы многочисленные статьи и интернет-источники, которые способствовали более глубокому пониманию понятий нейронная сеть и сверточная нейронная сеть. Полученные знания позволили выстроить общую структуру представления, что такое нейронная сеть, как проектировать структуру нейронной сети, каким образом предобрабатывать исходную выборку, чтобы впоследствии обучить нейронную сеть, какие выбирать параметры модели для обучения.

Проанализирован процесс обучения нейронной сети в программном модуле TensorFlow для python: для людей, которые имеют представления о нейронных сетях, о том, как готовить исходную выборку, в каком соотношении разделять исходную выборку на обучающую и тестовую, какими способами можно бороться с переобучением, знающие, что такое функция активации и имеющие другие познания, не составит труда работать с TensorFlow, так как модуль имеет высокоуровневое API, понятное для людей со знанием контекста.

В результате обучения собственной сверточной нейронной сети, за основу которой были взяты предоставляемые TensorFlow обучающие материалы, качество работы сети составило 90%, что является относительно хорошим показателем, но поскольку изначально стояла задача изучить возможности программного пакета, и не ставилась задача оптимизации отдельно скорости или отдельно качества распознавания или баланса этих двух показателей, оценить строго данный показатель нельзя. TensorFlow – это гибкий инструмент, который

позволяет строить нейронную сеть любой сложности без наличия знаний о том, как работают нейросети на более низких уровнях абстракции исходного кода, которые присутствуют в других программных пакетах.

Список использованных источников

1. He, K. Deep Residual Learning for Image Recognition / K. He, X. Zhang, S. Ren, J. Sun // IEEE Conference on Computer Vision and Pattern Recognition (CVPR). – 2016. – P. 770-778.
2. LeCun, Y. Backpropagation Applied to Handwritten Zip Code Recognition / Y. LeCun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, Hubbard W., Jackel L.D. // Neural Computation. – 1989. – №1(4). – P. 541-551.
3. KS, O. GPU Implementation of Neural Networks / O. KS, J. K // Pattern Recognition. – 2004. – №37(6). – P. 1311-1314.
4. Ghorpade, J. GPGPU Processing in Cuda Architecture / J. Ghorpade, J. Parande, M. Kulkarni, A. Bawaskar // Advanced Computing an International Journal. – 2012. – V. 3. – №1. – P. 105-120.
5. Krizhevsky, A. ImageNet Classification with Deep Convolutional Neural Networks / A. Krizhevsky, I. Sutskever, G.E. Hinton // Communications of the ACM. – 2012. – №60(6). – P. 84-90.
6. Колмогоров, А.Н. О представлении непрерывных функций нескольких переменных суперпозициями непрерывных функций меньшего числа переменных / А.Н. Колмогоров // Известия АН СССР. – 1956. – №108. – С. 179-182.
7. Галушкин А.И. Нейронные сети. Основы теории. М.: Горячая Линия - Телеком, 2010. – 496 с.
8. Doerfler, M. Inside the Spectrogram Convolutional Neural Networks in Audio Processing / M. Doerfler, T. Grill // International Conference on Sampling Theory and Applications (SampTA). 2017. – P. 152-155.
9. Wyse L. Audio Spectrogram Representations for Processing with Convolutional Neural Networks / L. Wyse // Proceedings of the First International Workshop on Deep Learning and Music joint with IJCNN. – 2017. – №1(1). – P. 37-41.
10. Masakazu, M. Subject Independent Facial Expression Recognition with Robust Face Detection Using A Convolutional Neural Network / M. Masakazu, K. Mori, Y. Mitari, Y. Kaneda // Neural Networks. – 2003. – V. 16. – №5. P. 555-559.
11. Hinton, G.E. Improving Neural Networks by Preventing Co-Adaptation of Feature Detectors / G.E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, R. R. Salakhutdinov // Computer Vision and Pattern Recognition. – 2012. P. 1-15.
12. Versloot S. What Are Max Pooling, Average Pooling, Global Max Pooling and Global Average Pooling? [Электронный ресурс]. – Режим доступа: <https://www.machinecurve.com/index.php/2020/01/30/what-are-max-pooling-average-pooling-global-max-pooling-and-global-average-pooling>. Дата доступа: 18.12.2020.
13. Rosenblatt, F Frank Rosenblatt: Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms. Spartan Books, 1962. – 616 с.
14. Verma S. Understanding Different Loss Functions for Neural Networks [Электронный ресурс]. – Режим доступа: <https://towardsdatascience.com/understanding-different-loss-functions-for-neural-networks-dd1ed0274718>. Дата доступа: 18.12.2020.
15. Kingma, D.P. Adam: A Method for Stochastic Optimization / D.P. Kingma, J. Ba // 3rd International Conference for Learning Representations. – 2014. – P. 1-15.

Материал поступил в редакцию: 11.10.2020

Материал принят к публикации: 01.12.2020

INFORMATION ABOUT THE PAPER IN ENGLISH

CONCEPT ANALYSIS NEURAL NETWORK AND CONVENTIONAL NEURAL NETWORK, CONVENTIONAL NEURAL NETWORK TRAINING USING THE TENSORFLOW MODULE

Vagaev I.I.

Abstract. The article consists analyzes the transition of the scientific community's attention from the main tool for pattern recognition using machine learning methods without using neural networks to convolutional neural networks as the best way to classify objects in images. The concepts of a neural network and a convolutional neural network are considered in detail, examples of one-layer, two-layer neural networks are given, a principle is shown that can be used to create a neural network of any depth, the distinctive features of convolutional neural networks has been analyzed, the types of layers that a convolutional neural network can have are analyzed. The process of training a neural network on the Fashion MNIST dataset using the TensorFlow software package from Google has been analyzed, the structure of its own convolutional neural network has been selected, the neural network has been trained to recognize various types of clothing. The result of the model's work on the Fashion MNIST test dataset was 90%. Analysis of the Tensor-Flow package showed the simplicity of the high-level API provided by the module, which allows you to build, configure and train neural networks of any complexity, including convolutional neural networks, which makes TensorFlow easy to integrate and use in your own development.

Keywords: neural network, convolutional neural network, GPU, CPU, neuron, activation function, TensorFlow, loss function, quality metric.

References

1. He, K. *Deep Residual Learning for Image* IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770-778.

2. LeCun, Y. *Backpropagation Applied to Handwritten Zip Code Recognition*, *Neural Computation*, 1989, no. 1(4), pp. 541-551.
3. KS, O. *GPU Implementation of Neural Networks*, *Advanced Computing an International Journal*, 2004, no. 37(6), pp. 1311-1314.
4. Ghorpade, J. *GPGPU Processing in Cuda*, *Advanced Computing an International Journal*, 2012, vol. 3, no. 1, pp. 105-120.
5. Krizhevsky, A. *ImageNet Classification with Deep Convolutional Neural*, *Communications of the ACM*, 2012, no. 60(6), pp. 84-90.
6. Kolmogorov A.N. *O predstavlenii nepreryvnykh funktsiy neskol'kih peremennykh superpozitsiyami nepreryvnykh funktsiy men'shego chisla peremennykh*, 1956, no. 108, pp. 179-182.
7. Galushkin A.I. *Nejronnye seti. Osnovy teorii*, 2010. – 496 с.
8. Doerfler M. *Inside the Spectrogram Convolutional Neural Networks in Audio Processing*, *International Conference on Sampling Theory and Applications (SampTA)*, 2017, pp. 152-155.
9. Wyse L. *Audio Spectrogram Representations for Processing with Convolutional Neural Networks*, *Proceedings of the First International Workshop on Deep Learning and Music joint with IJCNN*, 2017, no. 1(1), pp. 37-41.
10. Masakazu M. *Subject Independent Facial Expression Recognition with Robust Face Detection Using A Convolutional Neural Network*, *Neural Networks*, 2003, vol. 16, no. 5, pp. 555-559.
11. Hinton, G.E. *Improving Neural Networks by Preventing Co-Adaptation of Feature Detectors*, *Computer Vision and Pattern Recognition*, 2012, pp. 1-15.
12. Versloot S. *What Are Max Pooling, Average Pooling, Global Max Pooling and Global Average Pooling*, 2020
13. Rosenblatt, F Frank Rosenblatt: *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*. Spartan Books, 1962.
14. Verma S. *Understanding Different Loss Functions for Neural Networks*, 2020.
15. Kingma, D.P. Adam: *A Method for Stochastic Optimization*, *3rd International Conference for Learning Representations*, 2014, pp. 1-15.

ОБ АВТОРАХ:

Багаев Иван Игоревич – ведущий инженер-программист АО «НПО «Андронидная техника».
E-mail: InPROfess@yandex.ru.

ОБРАЗЕЦ ДЛЯ ЦИТИРОВАНИЯ:

Багаев, И.И. Анализ понятий нейронная сеть и сверточная нейронная сеть, обучение сверточной нейросети ПРИ помощи модуля TensorFlow / И.И. Багаев // Математическое и программное обеспечение систем в промышленной и социальной сферах. – 2020. – Т.8. – № 1. – С. 15-22. DOI: 10.18503/2306-2053-2020-8-1-15-22.

Bagaev I.I. (2020) Concept analysis neural network and conventional neural network, conventional neural network training using the tensorflow module. Software of systems in the industrial and social fields, 8 (1): 15-22.

DOI: 10.18503/2306-2053-2020-8-1-15-22.
