

УДК 004.8

*Ле Мань Ха*

Московский физико-технический институт (государственный университет)

## Свёрточная нейронная сеть для решения задачи классификации

В данной работе рассмотрена свёрточная нейронная сеть для решения задачи классификации, ее структуру, метод обучения и преимущества перед полносвязной нейронной сетью.

**Ключевые слова:** нейронная сеть, глубинное обучение, свёртка, субдискретизация, классификация.

*Le Manh Ha*

Moscow Institute of Physics and Technology (State University)

## Convolutional neural networks for classification

In this paper, the main purpose is to consider convolutional neural networks for classification, its structure, the learning method and advantages over common neural network.

**Key words:** neural network, deep learning, convolution, pooling, classification.

### 1. Введение

Сегодня искусственная нейронная сеть широко используется для решения различных задач в реальной жизни. Такие проблемы, как распознавание речи или оптическое распознавание символов, теперь решают с использованием нейронной сети с высокой точностью. В данной работе рассматривается одна специальная архитектура нейронной сети – свёрточная нейронная сеть, а также её структура и применение для классификации различных типов данных.

### 2. Скрытые признаки и глубинное обучение

Один из самых распространенных способов классификации – способ на основе описаний объектов с использованием признаков, в котором каждый объект характеризуется набором числовых или нечисловых признаков. Однако для некоторых типов данных открытые признаки не дают хорошей точности классификации, например, цвет точек изображений или цифровой звуковой сигнал. Классификация изображений собак и самолетов является простой для человека, но очень сложной для машины. В чем причина? Эти типы данных содержат «скрытые признаки», которые человек может легко воспринимать, а машина нет, например, у собаки есть морда, глаза и нос, а у самолета – крылья [1].

Глубинное обучение (англ. deep learning) представляет собой набор алгоритмов машинного обучения, которые пытаются моделировать высокоуровневые абстракции в данных, иными словами, выделять из данных «скрытые признаки» [2].

### 3. Автоэнкодер

Один из алгоритмов глубинного обучения – автоэнкодер (англ. autoencoder), это алгоритм обучения без учителя, выходной вектор которого равен входному вектору признаков [3]. Одной из самых распространенных архитектур автоэнкодера является нейронная сеть прямого распространения без обратных связей, содержащая входной, скрытый и выходной слои. В отличие от персептрона выходной слой автоэнкодера должен содержать столько же нейронов, сколько и входной слой. Данные на входном слое сжимаются на скрытом слое и восстанавливаются на выходном слое, таким образом выделяются «скрытые признаки».

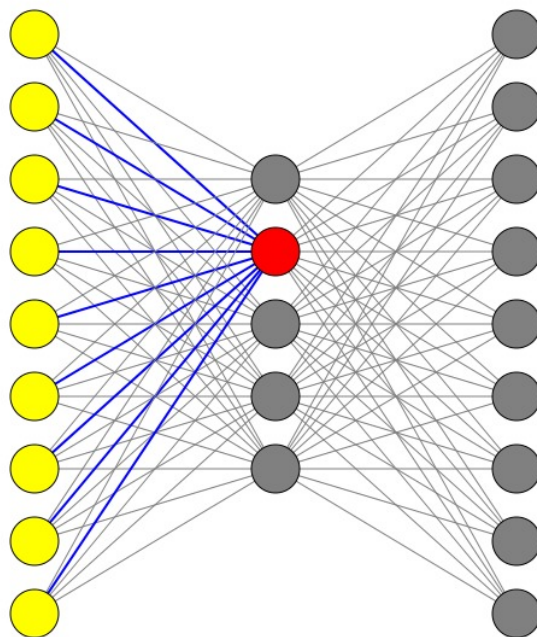


Рис. 1. Автоэнкодер

Цель автоэнкодера состоит в том, чтобы выход нейронной сети был наиболее близким к входному вектору. Для того чтобы решение этой задачи было нетривиальным, на топологию сети накладываются особые условия:

- 1) количество нейронов скрытого слоя должно быть меньше, чем размерность входных данных;
- 2) количество неактивных нейронов в скрытом слое должно значительно превышать количество активных нейронов.

Первое ограничение позволяет получить сжатие данных при передаче входного сигнала на выход сети. Такое сжатие возможно, если в данных есть скрытые взаимосвязи, корреляция признаков или структура. Второе ограничение – требование большого количества неактивных нейронов скрытого слоя – позволяет получать нетривиальные результаты, даже когда количество нейронов скрытого слоя превышает размерность входных данных, другими словами, цель автоэнкодера – получать только самые особые признаки. Будем считать нейрон активным, когда значение его функции передачи близко к единице, и, наоборот, неактивным, если значение его функции передачи близко к нулю.

Эти ограничения заставляют автоэнкодер искать обобщения и корреляцию в поступающих на вход данных, выполнять их сжатие. Таким образом, нейронная сеть автоматически

обучается выделять из входных данных общие признаки, которые кодируются в значениях весов сети. Необходимо, чтобы среднее значение функции передачи каждого скрытого нейрона приняло значение, наиболее близкое к заданному разреженному параметру (порядка 0.05). Для этого был добавлен в каждый нейрон скрытого слоя параметр разреженности  $\rho$ :

$$\hat{\rho}_j = \frac{1}{m} \sum_{i=1}^m \left[ a_j^{(2)}(x^{(i)}) \right]. \quad (1)$$

Необходимо, чтобы среднее значение функции передачи каждого скрытого нейрона приняло значение, наиболее близкое к  $\rho$ :

$$\hat{\rho}_j = \rho. \quad (2)$$

Штрафная функция:

$$S = \sum_{j=1}^{s_2} KL(\rho|\hat{\rho}_j), \quad (3)$$

$$(4)$$

где

$$KL(\rho|\hat{\rho}_j) = \rho \log \frac{\rho}{\hat{\rho}_j} + (1 - \rho) \log \frac{1 - \rho}{1 - \hat{\rho}_j}. \quad (5)$$

Интересное свойство штрафной функции – ее производная:

$$\frac{\partial KL(\rho|\hat{\rho}_j)}{\partial \rho_j} = -\frac{\rho}{\hat{\rho}_j} + \frac{1 - \rho}{1 - \hat{\rho}_j}. \quad (6)$$

#### 4. Свёрточная нейронная сеть

Автоэнкодер подходит для процесса обучения на данных небольшого размера, а при больших данных количество параметров (количество весов между входным и скрытым слоями, а также количество весов между скрытым и выходным слоями) существенно вырастет. Например, если при классификации картинок размером  $m \times n = 8 \times 8$ , мы хотим выделять  $k = 50$  признаков, тогда входной и выходной слои имеют  $m \times n = 8 \times 8 = 64$  нейронов, а скрытый слой имеет  $k = 50$  нейронов, количество весов равно  $2 \times m \times n \times k = 2 \times 8 \times 8 \times 50 = 6400$ . Но если картинки имеют размер  $m \times n = 100 \times 100$ , то количество весов будет  $2 \times 100 \times 100 \times 50 = 1\,000\,000$ .

Чтобы размер нейронной сети не был большим и не зависел от размера обучаемых данных, будем использовать специальную структуру нейронной сети, называемую *свёрточной нейронной сетью* (англ. Convolutional Neural Network – CNN) [4]. При обучении для каждого образца на вход свёрточной нейронной сети попадает не вся картинка  $m \times n$ , а только ее часть  $r \times c$ , например  $10 \times 10$ , эта часть называется *фильтром* или *ядром* и она сдвигается по большой картинке. Такая свёрточная нейронная сеть имеет фиксированное количество нейронов на входном и выходном слоях  $r \times c = 10 \times 10 = 100$ , а количество весов  $2 \times r \times c \times k = 2 \times 10 \times 10 \times 50 = 10\,000$ .

Для каждой большой картинки  $m \times n$  берем маленькие картинки  $r \times c$  для обучения, их количество равно  $(m - r + 1) \times (n - c + 1)$ .

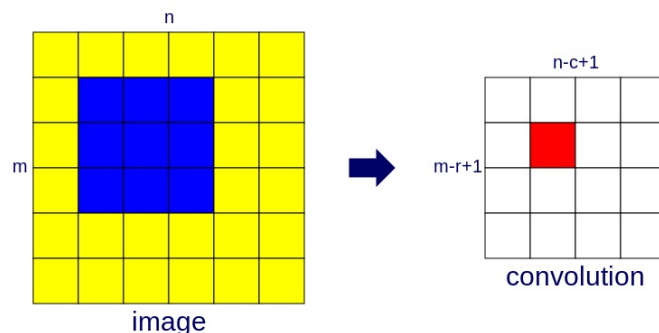


Рис. 2. Свёртка

## 5. Субдискретизация

При использовании свёрточной нейронной сети возникает проблема – большое количество выделенных признаков. Для одной большой картинке  $m \times n$  будет  $(m-r+1) \times (n-c+1)$  маленьких картинок  $r \times c$ , и количество выделенных признаков равно  $k \times (m-r+1) \times (n-c+1) = 50 \times (100-10+1) \times (100-10+1) = 414\,050$ . Использование такого огромного количества признаков для классификации оказалось неэффективно.

Для уменьшения размера пространства признаков проводим субдискретизацию (англ. pooling), разделив карту признаков, полученных от свёрточной нейронной сети, на фиксированное количество частей  $p$ , см. рис. 3, на этом рисунке  $p = p_m \times p_n = 2 \times 2 = 4$  и на каждой части вычисляется ее максимальное значение (англ. max pooling) или среднее значение (англ. mean pooling).

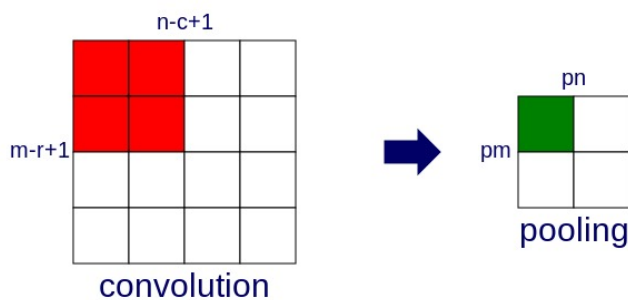


Рис. 3. Субдискретизация

## 6. Преимущества свёрточной нейронной сети

- 1) Уменьшение количества обучаемых параметров и повышение скорости обучения по сравнению с полносвязной нейронной сетью.
- 2) Возможность распараллеливания вычислений и реализации алгоритмов обучения сети на графических процессорах (GPU).
- 3) Устойчивость к сдвигу позиции объекта во входных данных. При обучении свёрточная нейронная сеть сдвигается по частям объекта. Поэтому обучаемые признаки не

зависят от позиции «важных частей», например голова собаки (см. рис. 4, свёрточная нейронная сеть выделяет одинаковые признаки для двух картинок, хотя позиции собак в этих картинках разные). Таким образом, это свойство свёрточной нейронной сети помогает повышать качество классификации [6].

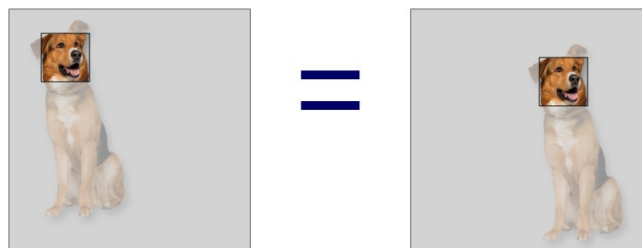


Рис. 4. Устойчивость к сдвигу позиции объекта во входных данных

## 7. Применение свёрточной нейронной сети в классификации текстов

### 7.1. Метод векторного представления слов

Известно, что метод векторного представления слов широко используется для решения задачи классификации текстов. В этом методе каждое слово представляется как числовой вектор из  $\mathbb{R}^n$ .

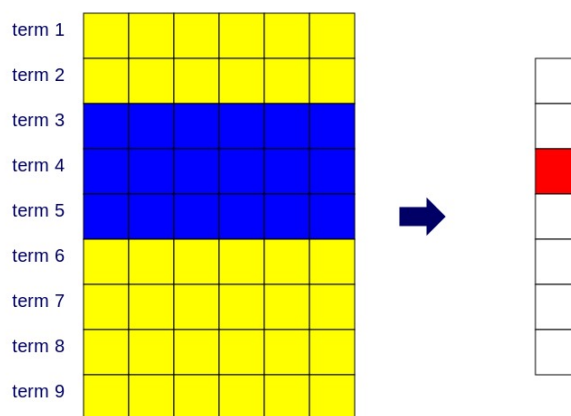


Рис. 5. Векторное представление слов и свёртка

### 7.2. При классификации текстов с использованием свёрточной нейронной сети

- 1) фильтр имеет такую же ширину  $m$ , как матрица (на вход нейронной сети поступают целые слова);
- 2) можно применять одновременно разные фильтры с разными высотами для выделения различных признаков.

Свёрточная нейронная сеть является очень эффективным методом представления текстов. Из-за сложности вычисления и хранения данных большинство  $n$ -грам корпусов, в том числе и Google, не представляет  $n$ -грам корпусы с  $n$  больше, чем 5. Например, корпус

n-gram Google из 1,024,908,267,229 слов, 95,119,665,584 предложений, 13,588,391 uni-grams, 314,843,401 bi-grams, 977,069,902 tri-grams, 1,313,818,354 four-grams, 1,176,470,663 five-grams после сжатия в формате gzip имеет размер 24 GB [7]. А свёрточная нейронная сеть может обучаться методом представления текста без хранения слов.

### 7.3. Тестирование:

Было проведено тестирование классификации документов с применением свёрточной нейронной сети на корпусе Wikinews на английском языке и сравнение его скорости, а также точности по сравнению с другими алгоритмами классификации текстов. Параметры использованной свёрточной нейронной сети: по 128 фильтрам высотой 3, 4 и 5, размер векторов слов  $n = 128$ , число эпох = 200. Обучение и тестирование были реализованы на основе работы Denny Britz [8] с использованием библиотеки TensorFlow ([www.tensorflow.org](http://www.tensorflow.org)).

Т а б л и ц а 1

Классификация текстов корпуса Wikinews на английском языке

Алгоритм	Точность (%)
KNN	87.3
Оптимизированный KNN	88.9
CNN	92.6

## 8. Заключение

Итак, свёрточную нейронную сеть можно применять для решения различных типов задач: классификации изображений, распознавания речи, классификации текстов и т.п. При использовании свёрточной нейронной сети можно существенно уменьшать количество обучаемых параметров и получать высокое качество классификации.

## Литература

1. Lee H., Grosse R., Ranganath R., Ng A.Y. Convolutional Deep Belief Networks for Scalable Unsupervised Learning of Hierarchical Representations. Proceedings of the 26th Annual International Conference on Machine Learning. 2009.
2. Bengio Y. Learning deep architectures for AI // Foundations and Trends in Machine Learning, 2009.
3. Hinton G.E., Salakhutdinov R.R. Reducing the dimensionality of data with neural networks // Science. 2006.
4. LeCun Y. LeNet-5, convolutional neural networks. Retrieved 16 November 2013.
5. Bengio Y., Courville A., Vincent P. Representation learning: A review and new perspectives. Pattern Analysis and Machine Intelligence // IEEE Transactions. 2013.
6. Bengio Y., LeCun Y. Scaling learning algorithms towards AI / eds. L. Bottou, O. Chapelle, D. DeCoste, J. Weston. Large Scale Kernel Machines. MIT Press, 2007.
7. Thorsten B., Franz A. Web 1T 5-gram Version 1 LDC2006T13. DVD. Philadelphia: Linguistic Data Consortium, 2006.
8. Britz D. Implementing a CNN for Text Classification in TensorFlow, 2015.

## References

1. *Lee H., Grosse R., Ranganath R., Ng A.Y.* Convolutional Deep Belief Networks for Scalable Unsupervised Learning of Hierarchical Representations. Proceedings of the 26th Annual International Conference on Machine Learning. 2009.
2. *Bengio Y.* Learning deep architectures for AI. Foundations and Trends in Machine Learning, 2009.
3. *Hinton G.E., Salakhutdinov R.R.* Reducing the dimensionality of data with neural networks. Science. 2006.
4. *LeCun Y.* LeNet-5, convolutional neural networks. Retrieved 16 November 2013.
5. *Bengio Y., Courville A., Vincent P.* Representation learning: A review and new perspectives. Pattern Analysis and Machine Intelligence. IEEE Transactions. 2013.
6. *Bengio Y., LeCun Y.* Scaling learning algorithms towards AI. eds. L. Bottou, O. Chapelle, D. DeCoste, J. Weston. Large Scale Kernel Machines. MIT Press, 2007.
7. *Thorsten B., Franz A.* Web 1T 5-gram Version 1 LDC2006T13. DVD. Philadelphia: Linguistic Data Consortium, 2006.
8. *Britz D.* Implementing a CNN for Text Classification in TensorFlow, 2015.

Поступила в редакцию 30.06.2016