

Рис. 8. Вхідні і вихідні дані фільтра з кроком 1

На Рис. 9 зображено фільтр з кроком 2, використання якого призводить до зменшення даних на виході. Як правило, крок фільтру збільшують для зменшення перекриття рецептивних полів та зменшення кількості даних на виході фільтра.

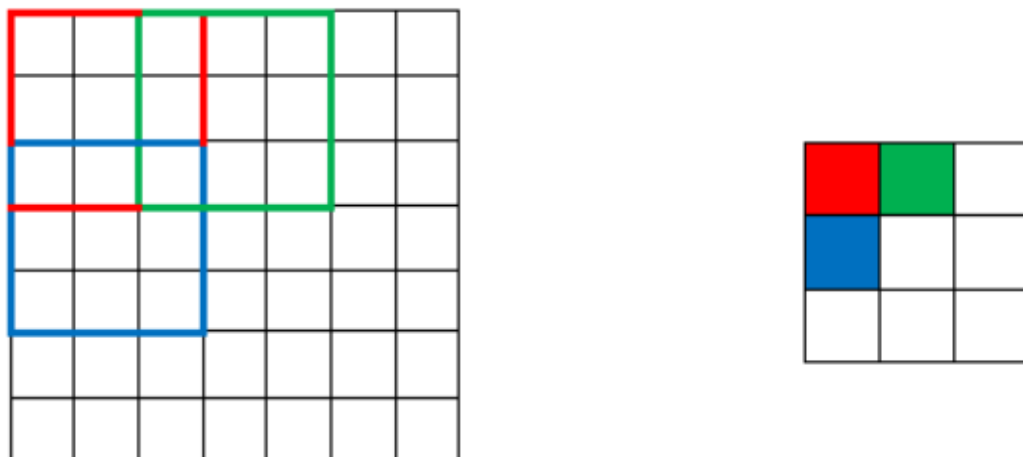


Рис. 9. Вхідні і вихідні дані фільтра з кроком 2

Тепер давайте поглянемо на відступ. Що відбудеться, якщо застосувати три фільтри $5 \times 5 \times 3$ до вхідних даних розміром $32 \times 32 \times 3$? Розмір вихідних буде $28 \times 28 \times 3$. Оскільки ЗНМ побудована на застосуванні ряду згорткових шарів, то розмір вихідних даних після застосування кожного з них зменшуватиметься швидше, ніж хотілося б. У ранніх шарах мережі ми хочемо зберегти максимальну кількість інформації про вхідні дані. Скажімо, ми хочемо застосувати один і той же згортковий шар, але хочемо, щоб обсяг вихідних даних залишався $32 \times 32 \times 3$. Для цього ми можемо застосувати нульовий відступ розміру 2 до цього шару. Нульовий відступ збільшує розмір матриці вхідних даних за допомогою додавання нулів «навколо».

На Рис. 10 зображено вхідні дані, представлені матрицею з розміром $32 \times 32 \times 3$. Якщо додати нулі «навколо», ми отримаємо матрицю з розміром $36 \times 36 \times 3$. Після застосування згорткового шару з 3-ма фільтрами $5 \times 5 \times 3$ і кроком 1, отримаємо вихідну матрицю все того ж розміру – $32 \times 32 \times 3$. [3]

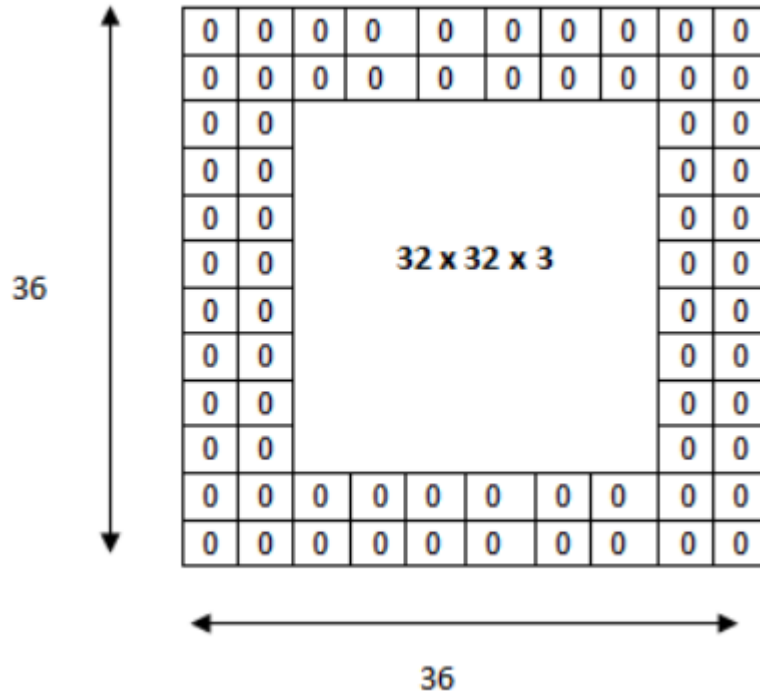


Рис. 10. Вхідні дані розміром 32x32x3 з відступом – 2

Якщо розмір кроку – 1 і розмір відступу задати за формулою:

$$P = \frac{K - 1}{2}, \quad (3)$$

де K – розмір фільтру. Розмір матриці вхідних даних завжди дорівнювати розміру матриці вихідних даних.

Формула для обрахунку розмірів вихідних даних для будь-якого згорткового шару має наступний вигляд:

$$O = \frac{W - K + 2 \cdot P}{S} + 1, \quad (4)$$

де O – вихідна висота або ширина, W – вхідна висота або ширина, K – розмір фільтру, P – відступ, S – крок.

Вибір параметрів навчання

Як визначити, скільки шарів слід використовувати, скільки згорткових шарів, які розміри фільтрів та значення для кроку і відступу? Це не тривіальні питання і немає встановленого стандарту, який використовують усі дослідники. Це пояснюється тим, що мережа значною мірою залежить від типу даних, з якими вона працює. Дані можуть відрізнятися за розміром, складністю зображення, типом завдання обробки зображень тощо. Один із способів обрати параметри – це знайти правильну комбінацію, яка створює абстракції зображення у належному масштабі.

ReLU (Rectified Linear Units) шари

Після кожного згорткового шару прийнято застосовувати нелінійний шар (або шар активації). Метою цього шару є введення нелінійності в систему, яка в основному просто обраховує лінійні операції під час застосування згорткових шарів (множення і додавання елементів). В минулому застосовувалися нелінійні функції, такі як \tanh і sigmoid , але дослідники з'ясували що ReLU шари працюють значно краще, то му що мережа має змогу навчатися значно швидше (через обчислювальну ефективність) без істотної різниці в точності. Це також допомагає полегшити проблему зникаючого градієнта, що є проблемою, коли нижні шари мережі навчаються дуже повільно, через градієнт що експоненційно зменшується. Шар ReLU застосовує функцію $f(x) = \max(0, x)$ до всіх вхідних значень. Коротко кажучи, ця функція просто замінює всі негативні значення на 0. ReLU шар збільшує нелінійні властивості моделі та мережі в загальному, не впливаючи на рецептивні поля згорткового шару. [5]

Шари об'єднання (Pooling layers)

Після деяких шарів ReLU розробники можуть застосувати шар об'єднання. Його також називають шаром зменшення. В цій категорії існує також кілька варіантів, найпопулярнішим є макспулінг (maxpooling, рис. 11). Такий підхід застосовує фільтр (зазвичай розміром 2×2) з кроком такого ж розміру. Фільтр застосовується до вхідних даних і залишає лише максимальне значення в кожній з областей згорнутих фільтром.

Інші підходи до об'єднання – це середнє значення та *L2-norm*, яка визначається за формулою:

$$|x| = \sqrt{\sum_{k=1}^n |x_k|^2}. \quad (5)$$

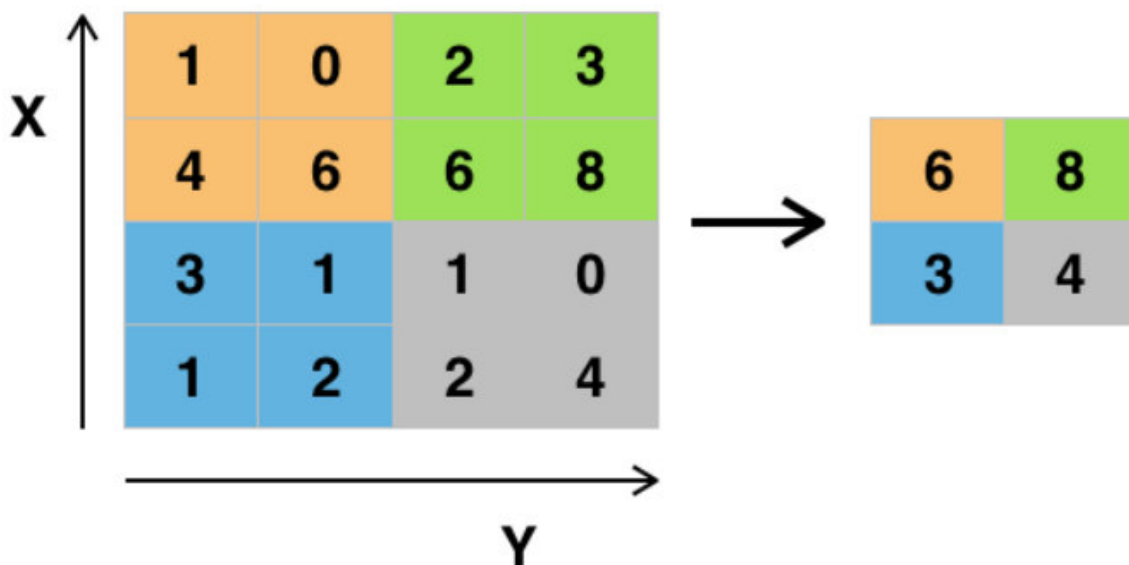


Рис. 11. Макспулінг з фільтром розміром 2×2 і кроком 2

Коли ми дізнаємося, що специфічна характеристика є у вхідних даних (в цій області буде високе значення активації), її точне розташування не так важливе як відносне розташування відносно інших характеристик. ReLU шар дуже швидко зменшує розміри (довжину і ширину, але не глибину) вхідних даних. Такі

трансформації мають 2 основні цілі. Перша – зменшення кількості параметрів або ваг на 75%, що зменшує складність обчислень. Друга – контроль **перенавчання (overfitting)**. Це поняття описує процес при якому модель настільки налаштована на навчальні приклади, що не в змозі узагальнити дані для перевірки на тестових даних. Симптомом перенавчання є наявність моделі, яка видає 100% або 99% результат на навчальних даних, але лише 50% результат на тестових даних.

Шари відсівання (Dropout Layers)

Шари відсівання мають дуже специфічну функцію в нейронних мережах. Цей шар «відсіває» випадкові набори активацій, встановлюючи їх значення рівним нулю. Які ж переваги такого простого і, здавалося б, непотрібного і неінтуїтивного процесу? Такий підхід змушує мережу забезпечувати правильну класифікацію результату для конкретного прикладу, навіть якщо деякі з активацій відкинуті. Це гарантує, що мережа не стане занадто «пристосованою» до навчальних даних, а отже, знижує ймовірність виникнення проблеми перенавчання (перенасичення). Варто зазначити, цей шар використовується лише під час навчання, але не під час тестування. [11]

Шар мережі в мережі (Network in Network Layers)

Шар мережі в мережі посилається на згортковий шар з фільтром розміру 1×1 . Виникає питання, чому цей шар є корисним, оскільки рецептивні поля зазвичай більші за простір, на який вони відображаються. Однак, згортка розміром 1×1 охоплює певну глибину, тому таку згортку необхідно розглядати як згортку розміром $1 \times 1 \times N$, де N – кількість фільтрів застосованих в цьому шарі. Фактично цей шар виконує N -D елементарне множення, де N – глибина вхідних даних шару. [10]

Класифікація, локалізація, виявлення, сегментація

Класифікація зображень - це процес отримання вхідного зображення і виведення номеру або назви класу з набору категорій. Але для задачі **локалізації** об'єктів потрібно не лише визначити клас, а і створити обмежувальну рамку, яка описує, де об'єкт знаходиться на зображенні.

Також є завдання **виявлення об'єктів**, де локалізація повинна здійснюватися на всіх об'єктах зображення (рис. 12, 13). Таким чином з'являється кілька обмежувальних полів і декілька класів.



Рис. 12. Класифікація – визначення що зображення це собака



Рис. 13. Локалізація – визначення класу та знаходження об'єкта на зображенні

Трансферне навчання (TransferLearning)

Поширена помилка у спільноті машинного навчання полягає в тому, що без величезних об'ємів даних (таких як в Google), ми не маємо можливості створити ефективні моделі глибокого навчання. Хоча дані є важливою частиною створення мережі, ідея трансферного навчання допомогла зменшити вимоги до даних. **Трансферне навчання** – це процес взяття попередньо навченої моделі (ваги та параметри мережі, яка була навчена на великій кількості даних кимось іншим) і «тонкого налаштування» моделі з нашим власним набором даних. Ідея полягає в тому, що ця попередньо навчена модель буде виступати в якості екстрактора ознак. Ми вилучаємо останній шар мережі і заміняємо його власним класифікатором (залежно від області задачі). Далі ми заморожуємо ваги всіх інших шарів і навчаємо мережу стандартним способом (замороження означає заборону змін ваг під час оптимізації).



Рис. 14. Виявлення об'єктів – локалізація кількох об'єктів

Завдання **сегментації об'єктів** – це визначення класу та контуру кожного об'єкта у вхідному зображенні (рис. 15). [4]



Рис. 15. Сегментація об'єктів – визначення класу об'єкта та його виділення на вхідному зображенні

Розглянемо, чому це працює. Припустимо, попередньо підготовлена модель була навчена на ImageNet (ImageNet – набір даних, що містить 14 мільйонів зображень з більш ніж 1000 класами) [2]. Коли ми говоримо про низькорівневі шари мережі, ми знаємо, що вони будуть виявляти такі риси, як ребра та криві. Тепер, якщо в нас не дуже унікальний простір проблем і набір даних, мережі потрібно буде виявити криві і ребра. Замість того, щоб навчати мережу через ініціалізацію ваг випадковими значеннями, ми можемо використати ваги вже навченої моделі і зосередитися на більш важливих для навчання шарах. Якщо ж наш набір даних абсолютно інший ніж той що в ImageNet, тоді необхідно збільшити кількість шарів і заморозити лише кілька нижніх. [9]

Методи збільшення даних

Розглянемо, як можна збільшити існуючий набір даних лише за допомогою кількох легких перетворень. Як вже згадувалося раніше, коли комп'ютер отримує на вхід зображення, він отримує його у виді масиву чисел, що представляють пікселі. Припустимо, що все зображення зміщується вліво на один піксель. Для людини ця зміна непомітна. Однак для комп'ютера цей зсув може бути досить значним, оскільки клас зображення не змінюється, а масив пікселів – так. Підходи, що змінюють навчальні дані способами, які модифікують подання масиву, зберігаючи клас зображення, відомі як **методи збільшення даних**. Вони є способом штучного розширення набору даних. Також популярними підходами є використання чорно-білих зображень, випадкові обрізання, обертання та багато іншого. Застосувавши лише кілька цих перетворень до навчальних даних, можна легко подвоїти або потроїти кількість навчальних прикладів.

Висновки

У статті розглянуто поняття згорткових нейронних мереж, їх зв'язок з біологією та структуру. Далі більш детально розглянуто основні шари, що є в ЗНМ, описано їх роботу за допомогою математики. Кожен з шарів розглянуто окремо та у взаємодії з іншими шарами нейронної мережі.

Показано як нейронні мережі працюють, та як їх правильно навчати. Розглянуто кілька підходів до навчання нейронних мереж та прості способи вирішення поширених проблем, що виникають при навчанні: збільшення об'єму даних для навчання та

використання вже навчених мереж як бази для вирішення вузькоспеціалізованих задач. Наведено параметри, завдяки яким можна змінювати процес навчання мережі.

Розглянуто кілька найпоширеніших задач, які розв'язують за допомогою згорткових нейронних мереж: класифікація, локалізація, виявлення, сегментація.

Список використаної літератури:

1. C. Zhang, and Z. Zhang, "Improving multiview face detection with multi-task deep convolutional neural networks," IEEE Winter Conference on Applications of Computer Vision, 2014, pp. 1036-1041.
2. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). ImageNet: A Large-Scale Hierarchical Image Database. In CVPR09.
3. Dumitru Erhan, Christian Szegedy, Alexander Toshev, Dragomir Anguelov, Scalable Object Detection using Deep Neural Networks, arXiv preprint arXiv: arXiv:1312.4400, 2014.
4. E. Borenstein; J. Malik, "Shape Guided Object Segmentation" in IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2006.
5. Geoffrey E. Hinton, Vinod Nair. "Rectified Linear Units Improve Restricted Boltzmann Machines", Proceedings of International Conference on Machine Learning (ICML), 2010, pp 807-814.
6. H. Li, Z. Lin, X. Shen, J. Brandt, and G. Hua, "A convolutional neural network cascade for face detection," in IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 5325-5334.
7. Hubel, D. H., & Wiesel, T N. (1979). Brain mechanisms of vision. Scientific American, 241, 150–162.
8. J. Yan, Z. Lei, L. Wen, and S. Li, "The fastest deformable part model for object detection," in IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 2497-2504.
9. Jason Yosinski, Jeff Clune, Yoshua Bengio, Hod Lipson, How transferable are features in deep neural networks?, arXiv preprint arXiv: arXiv: 1411.1792v1, 2014.
10. Min Lin, Qiang Chen, Shuicheng Yan, Network In Network, arXiv preprint arXiv: arXiv: 1312.2249, 2013.
11. Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, Ruslan Salakhutdinov, Dropout: A Simple Way to Prevent Neural Networks from Overfitting, Journal of Machine Learning Research 15, 2014, pp. 1929-1958.

Bibliography:

1. C. Zhang, and Z. Zhang, "Improving multiview face detection with multi-task deep convolutional neural networks," IEEE Winter Conference on Applications of Computer Vision, 2014, pp. 1036-1041.
2. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). ImageNet: A Large-Scale Hierarchical Image Database. In CVPR09.
3. Dumitru Erhan, Christian Szegedy, Alexander Toshev, Dragomir Anguelov, Scalable Object Detection using Deep Neural Networks, arXiv preprint arXiv: arXiv:1312.4400, 2014.
4. E. Borenstein; J. Malik, "Shape Guided Object Segmentation" in IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2006.
5. Geoffrey E. Hinton, Vinod Nair. "Rectified Linear Units Improve Restricted Boltzmann Machines", Proceedings of International Conference on Machine Learning (ICML), 2010, pp 807-814.
6. H. Li, Z. Lin, X. Shen, J. Brandt, and G. Hua, "A convolutional neural network cascade for face detection," in IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 5325-5334.
7. Hubel, D. H., & Wiesel, T N. (1979). Brain mechanisms of vision. Scientific American, 241, 150–162.
8. J. Yan, Z. Lei, L. Wen, and S. Li, "The fastest deformable part model for object detection," in IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 2497-2504.
9. Jason Yosinski, Jeff Clune, Yoshua Bengio, Hod Lipson, How transferable are features in deep neural networks?, arXiv preprint arXiv: arXiv: 1411.1792v1, 2014.
10. Min Lin, Qiang Chen, Shuicheng Yan, Network In Network, arXiv preprint arXiv: arXiv: 1312.2249, 2013.
11. Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, Ruslan Salakhutdinov, Dropout: A Simple Way to Prevent Neural Networks from Overfitting, Journal of Machine Learning Research 15, 2014, pp. 1929-1958.

KUCHER Oleksandr,

PhD-student, The Bohdan Khmelnytsky National University of Cherkasy

FEATURES OF APPLICATION OF CONVENSIONAL NEURAL NETWORKS FOR IMAGE PROCESSING PROBLEMS

Summary. Introduction. Convolutional Neural Networks (CNN) are one of the most influential innovations in the field of computer vision. For the first time neural networks attracted attention in 2012 at the ImageNet competition. With the help of CNN, a new record of classification errors was set

- 15% (the previous value was 26%). Today, many well-known companies use deep learning. But the classic and most popular way to use neural networks is through image processing. Consider how CNNs are used to classify images.

Purpose. The task of image classification is to process the image and determine the class to which it belongs (car, animal, etc.) or the group of classes that best characterize it.

Results. The article gives the basic concept and characteristics of convolutional neural networks, describes the structure and mathematical approaches to the implementation of this type of neural network. Provided description of the main layers of convolutional neural networks, mentioned parameters that allow changing the process of training the network. The process of training each of the neural network's layer is considered in detail. In addition, described techniques that allow you to increase the data set and improve the training process of the network only by several transformations.

Conclusion. The concept of convolutional neural networks, their connection with biology and structure has been considered in the article. The main layers that are found in CNN are discussed in detail, described their work through mathematics. Each of the layers is considered individually and in conjunction with the other layers of the neural network.

Described how neural networks work and how to train them properly. Several approaches to learning neural networks are discussed, as well as simple ways to solve common learning problems: increasing the amount of data you need to train and using already trained networks as a base for solving specialized tasks. Here are some options for changing your network's learning process.

The article discusses some of the most common tasks that are solved using convolutional neural networks: classification, localization, detection, and segmentation.

Keywords: deep learning, machine vision, neural networks, face recognition, convolutional neural networks, receptive fields, deep learning, neural networks, convolutional neural networks, classification, localization, detection, segmentation, rectified linear units.

Одержано редакцією 11.04.2018 р.
Прийнято до публікації 19.09.2018 р.

УДК 519.688

DOI 10.31651/2076-5886-2019-1-75-85

PACS 02.60.-x, 02.60.Pn

КУЦЕНКО Олександр Анатолійович
студент спеціальності «Прикладна
математика» Черкаського національного
університету імені Богдана
Хмельницького
e-mail: alexkutsenko1995@gmail.com
ORCID 0000-0002-6220-6034

СЕРДЮК Олександр Анатолійович
кандидат економічних наук, старший
викладач кафедри прикладної математики
та інформатики Черкаського
національного університету
ім. Б. Хмельницького
e-mail: serdyuk4labs@ukr.net
ORCID 0000-0002-3919-4661

ОГЛЯД АЛГОРИТМІВ ПОШУКУ ПЛАГІАТУ У ПРОГРАМНОМУ КОДІ

У статті розглянуто поняття плагіату, його класифікацію та загальнозживані алгоритми пошуку плагіату, а саме: метод ідентифікаційних міток, алгоритм Хескела, метод вирівнювання рядків та метод жадібного рядкового заміщення. З розвитком технологій та можливістю використання інтернету студент, не прикладаючи багато зусиль, може видавати матеріали іншої персоні за свої. Тому дані методи та алгоритми часто