

Нейронные сети. Свёрточные нейронные сети

Алексей Колесов

Белорусский государственный университет

2 декабря 2018 г.

Свёрточные нейронные сети

- в CNN имеет обучаемые параметры (веса и bias), вычисляет скалярное произведение и применяет нелинейность
- CNN — дифференцируемая функция
- Свёрточные сети явно предполагает, что на входе изображение

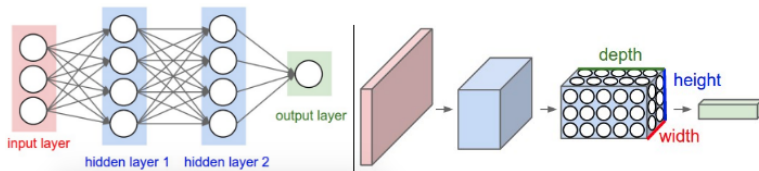
Плохость fc-сетей

- пусть изображение $32 \times 32 \times 3$
- нейрон первого слоя имеет $3072 = 32 \cdot 32 \cdot 3$ параметра
- если больше одного нейрона или большее изображение — количество параметров растёт быстро

Решение

- входные данные — изображения (ширина, высота, глубина)
- CIFAR-10 — $32 \times 32 \times 3$ (RGB)
- на выходе $1 \times 1 \times 10$ (10 выходных классов)
- каждая картинка (срез по глубине) — «feature map»
- каждый нейрон «смотрит» только на небольшой участок изображения

Сравнение



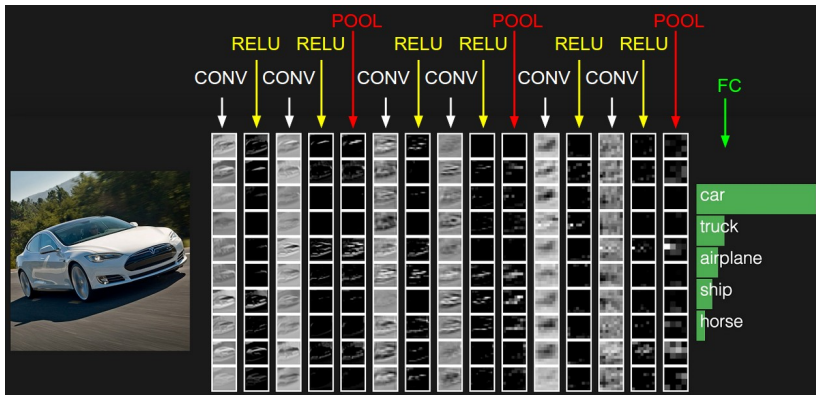
Слои для построения CNN

Для построения свёрточных сетей используются компоненты:

- `input` — входное изображение 3D
- `conv` — свёрточный слой (локальное скалярное произведение); выход — трёхмерный массив
- `relu` — или любая другая нелинейность; выход — трёхмерный тензор
- `pooling` — слой для сокращения размерности, без параметров; выход — трёхмерный тензор
- `bn` — слой нормализации по мини-батчам
- `fc` — полносвязный слой

Послойная визуализация

Tiny VGG-net

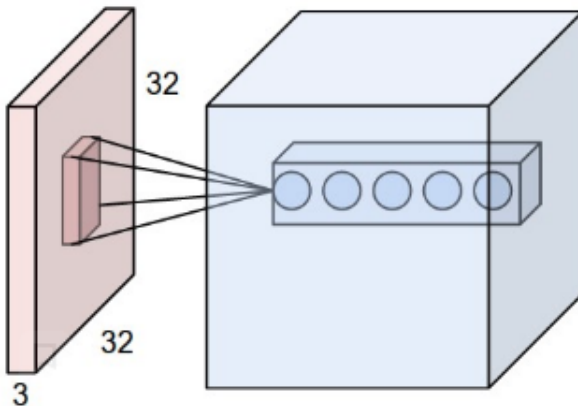


CONV-слой

- трёхмерная матрица параметров — набор обучаемых «фильтров»
- малые размеры по пространственным параметрам; полный размер по глубине
- вычисляем значение фильтра в каждой точке, получаем изображение
- соединяем изображения — получаем выход
- стандартный размер: 3×3

CONV-слой: локальная связность

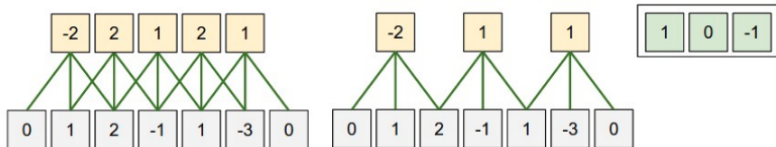
- не будем соединять все нейроны прошлого слоя с нейронами текущего слоя
- смотрим на небольшой участок по ширине и высоте, на весь по глубине



CONV-слой: выходной слой

На размер выходного слоя влияют:

- глубина (количество фильтров)
- stride (шаг, с которым вычисляем свёртку)
- padding



CONV-слой: разделение параметров

- каждый нейрон имеет значительное количество параметров
- нам неважно, есть ли сигнал в (x_1, y_1) или в (x_2, y_2)
- давайте каждый нейрон будет использовать одни и те же параметры!
- следствие: обратный проход — тоже свёртка

CONV-слой: примеры фильтров



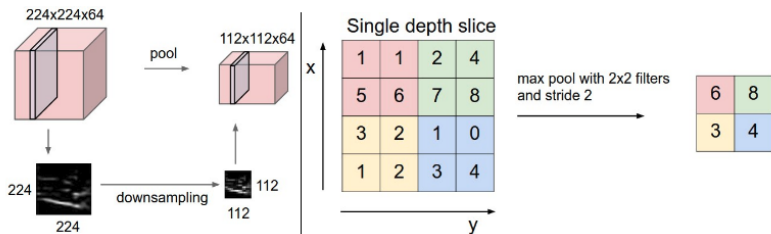
CONV-слой: замечания

- можно реализовать как матричное умножение
- обратный проход — тоже свёртка
- 1×1 свёртки
- dilated (расширенная) свёртка — шаг в суммировании больше 1

Pooling-слой

- задача — уменьшить размерность
- чаще всего — max в окошке 2×2 с шагом 2 (реже 3×3 с шагом 2)
- больше размер — слишком много информации теряется
- другие функции — применяются, но обычно хуже
- обратный ход — тривиальный
- есть работы по избавлению от pooling

Pooling-слой



Rules of thumb

- размер входного изображения должен делиться на большую степень двойки
- размер фильтров должен быть небольшим (3×3 , 5×5)
- используйте zero-padding и stride равный 1
- используйте batchnorm!

Важные архитектуры

- LeNet (Ян ЛеКун, 1990-е) — распознавание штрих-кодов
- AlexNet — нейронная сеть, которая в 2012 году выиграла ImageNet (как LeNet, но глубже, несколько Conv слоёв подряд)
- GoogLeNet — Inception module
- VGGNet — очень глубокая (16 слоёв, 140M параметров)
- ResNet — residual connection
- Densenets — принимаем на вход feature map со всех предыдущих слоёв!

Больше правил!

- используйте претренированные модели
- используйте separable convolutions ($3 \times 3 \Rightarrow 3 \times 1 \cdot 1 \times 3$)
- используйте residual connections

Итоги

- рассмотрели устройство свёрточных нейронных сетей

Литература

- С. Николенко, А. Кадури, Е. Архангельская — Глубокое обучение (главы 3-4)