Содержание

1 NN

Дано

$X \in \mathbb{R}^{n \times p}$ — множество объектов.
$Y \in \mathbb{R}^{n}$ — множество ответов.

Оптимизационная задача

$$Q(a, X^n) = \frac{1}{n} \sum_{j=1}^{n} \mathcal{L}(a, x_i, y_i) \to \min_{\omega}.$$

Алгоритм (модель)

$$a(x, \omega) = \sigma\left(< \omega, x >\right) \sigma\left(\sum_{j=1}^{p} \omega_j f_j(x) - \omega_0\right),$$

$\omega = (\omega_1, \dots, \omega_p) \in \mathbb{R}^p$ — вектор параметров.
$\sigma : \mathbb{R} \to \mathbb{R}$ — функция активации.
ω_0 — порог активации.

Если $\sigma(z) = sign(z)$, то $a(x, \omega)$ — просто линейный классификатор. Уравнение $< \omega, x >= 0$ задает гиперплоскость, разделяющие классы в пространстве $\mathbb{R}^p$. Если вектор x находится по одну сторону гиперплоскости с ее направляющим вектором ω, то объект x относится к классу $+1$, иначе к классу -1.

Задача регрессии

$\mathbf{y} = \mathbb{R}$.
$a(x_j, \omega) = \sigma(< \omega, x_j >)$.

$$Q(\omega; X^n) = \sum_{j=1}^{n} \mathcal{L}(< \omega, x_j >, y_j) = \sum_{j=1}^{n} \left(\sigma(< \omega, x_j >) - y_j\right)^2 \to \min_{\omega}.$$

При $\sigma(z) = z$ получаем многомерную линейную регрессию.

Задача классификации

$\mathbf{y} = \{-1, 1\}$.
$a(x_j, \omega) = sign < \omega, x_j >$.

$$Q(\omega; X^n) = \sum_{j=1}^{n} \mathcal{L}(< \omega, x_j >, y_j) = \sum_{j=1}^{n} \left[y_j < \omega, x_j >< 0\right]^2 \to \min_{\omega}.$$

Функции активации

- логистическая (сигмоидная) функция: $\sigma(z) = \dfrac{1}{1 + e^{-az}}, a \in \mathbb{R}$;

- гиперболический тангенс: $\sigma(z) = \dfrac{e^{az} - e^{-az}}{e^{az} + e^{-az}}, a \in \mathbb{R}$;
- softmax: $SM_i(z) = \dfrac{e^{z_i}}{\sum\limits_{k=1}^{K} e^{z_k}}$;
- выпрямитель: $ReLU(p) = \max(0, p)$.

Функции потерь

- MSE
- CE (кросс энтропия)
- BCE (бинарная кросс-энтропия)

Расчет весов

Backprropagation (метод обратного распространения ошибки), стохастический градиентный спуск.

Выбор гиперпараметров

- число слоев
- число нейронов
- число связей для каждого нейрона
- и т.д. (число эпох, batch size)
- динамическое добавление нейронов
- удаление избыточных связей (OBD)

2 CNN

2.1 Особенности построения нейронных сетей для изображений (Convolutional Neural Networks)

- локальная скореллированность пикселей
- распределенность признака
- возможность трансформации шаблонов (поворот, смещение, масштабирование)

Входной сигнал изображения подается на вход нейрона только в пределах ограниченной области, как правило, квадратной, например, 3x3 пикселей. Затем, эта область смещается вправо на заданный шаг, допустим, 1 пиксель и входы подаются уже на второй нейрон. Так происходит сканирование всего изображения. Причем, весовые коэффициенты для всех нейронов этой группы – одинаковые.

После этого сканирование изображения повторяется, но с другим набором весовых коэффициентов. Получаем вторую группу нейронов. Затем, третью, четвертую и в общем случае имеем n различных групп. Так формируется первый скрытый слой нейронов сверточной НС.

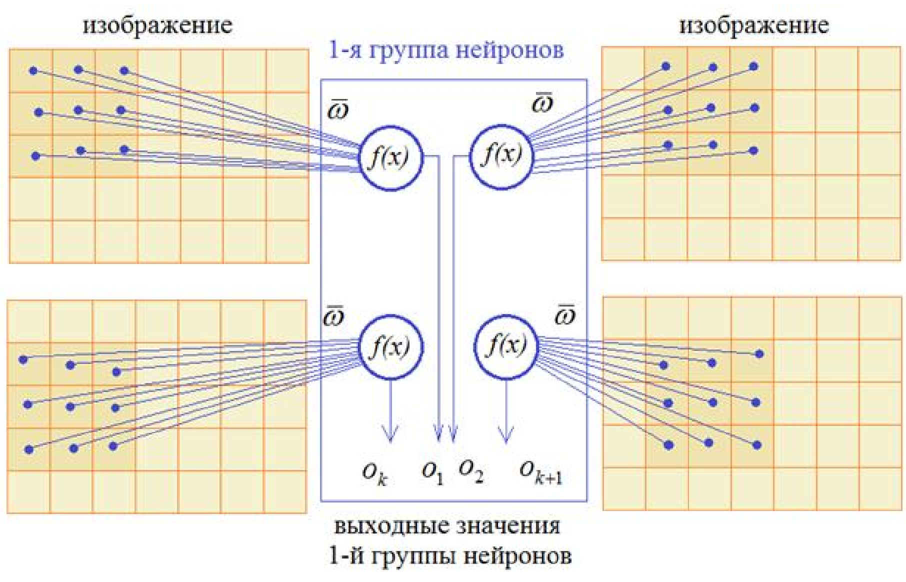

Рис. входной сигнал

2.2 Свертка

Свёртка, конволюция — операция в функциональном анализе, которая при применении к двум функциям f и g возвращает третью функцию, соответствующую взаимнокорреляционной функции $f(x)$ и $g(-x)$. Операцию свёртки можно интерпретировать как «схожесть» одной функции с отражённой и сдвинутой копией другой. Понятие свёртки обобщается для функций, определённых на произвольных измеримых пространствах, и может рассматриваться как особый вид интегрального преобразования. В дискретном случае свёртка соответствует сумме значений f с коэффициентами, соответствующими смещённым значениям g , то есть

$$(f * g)(x) = f(1)g(x-1) + f(2)g(x-2) + f(3)g(x-3) + \dots$$

Определение свертки

Пусть $f, g : \mathbb{R}^n \to \mathbb{R}$ — две функции, интегрируемые относительно меры Лебега на пространстве $\mathbb{R}^n$. Тогда их свёрткой называется функция $f * g : \mathbb{R}^n \to \mathbb{R}$, определённая формулой

$$(f * g)(x) \stackrel{\text{def}}{=} \int\limits_{\mathbb{R}^n} f(y)\, g(x-y)\, dy = \int\limits_{\mathbb{R}^n} f(x-y)\, g(y)\, dy.$$

2.3 Карта признаков (feature map / activation map)

Пусть на вход нейросети подается изображение размером $M \times N$. Обозначим его как отображение $x : Z \times Z \to \mathbb{R}^3$. Введем ядро свертки, представляющее собой набор весов $W : Z \times Z \to \mathbb{R}^3$. Рассмотрим следующее отображение:

$$f(i,j) : Z \times Z \to \mathbb{R};$$

$$f(i,j) = \sigma((W \cdot x)(i,j) + b);$$

где b — смещение, σ — нелинейная функция. Вектор из таких отображений, построенный на разных ядрах W_k и смещениях b_k называется **feature map**.

2.4 Ядро фильтра

Ядро фильтра — набор коэффициентов ω размера, например, 3×3.

$$W = \begin{matrix} \omega_{11} & \omega_{12} & \omega_{13} \\ \omega_{21} & \omega_{22} & \omega_{23} \\ \omega_{31} & \omega_{32} & \omega_{33} \end{matrix}$$

Мы имеем $3 \cdot 3 + 1 = 10$ настраиваемых параметров.

Матрица данных (изображение)

$$\begin{matrix} x_{11} & x_{12} & x_{13} & x_{14} & x_{15} \\ x_{21} & x_{22} & x_{23} & x_{24} & x_{25} \\ x_{31} & x_{32} & x_{33} & x_{34} & x_{35} \\ x_{41} & x_{42} & x_{43} & x_{44} & x_{45} \\ x_{51} & x_{52} & x_{53} & x_{54} & x_{55} \end{matrix}$$

Свертка общая формула
$v_{k,m} = \sum_{i=1}^{3} \sum_{j=1}^{3} x_{i+k,j+m} \cdot \omega_{ij} + \omega_0,\ k,m = 0,\ldots,n$

Свертка пример
$v_{0,0} = \sum_{i=1}^{3} \sum_{j=1}^{3} x_{i,j} \cdot \omega_{ij} + \omega_0$

$v_{0,1} = \sum_{i=1}^{3} \sum_{j=1}^{3} x_{i,j+1} \cdot \omega_{ij} + \omega_0$

Берем часть изображения. Если сумма произведений значения каждого пикселя на весовой коэффициент превышает некоторое пороговое значение, это значит что мы выявили паттерн, функция активации срабатывает и сигнал передается на следующий слой.

2.5 Пример паттерна

Пусть у нас есть фильтр, способный определять вертикальную линию, где 0 — белый цвет, 1 — черный цвет (либо наоборот).

$$W = \begin{matrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{matrix}$$

Тогда если у нас будет следующее изображение

$$x = \begin{matrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{matrix}$$

то $v_{0,0} = 1 \cdot 1 + 1 \cdot 1 + 1 \cdot 1$
Если же у нас будет длинная вертикальная линия

$$x = \begin{matrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ \vdots & \vdots & \vdots \\ 0 & 1 & 0 \end{matrix}$$

то наши коэффициенты при каждом сдвиге будут следующими

$$\begin{aligned} v_{0,0} &= 1 \cdot 1 + 1 \cdot 1 + 1 \cdot 1 \\ v_{1,0} &= 1 \cdot 1 + 1 \cdot 1 + 1 \cdot 1 \\ &\vdots \\ v_{n,0} &= 1 \cdot 1 + 1 \cdot 1 + 1 \cdot 1 \end{aligned}$$

На каждом шаге мы будем получать большие коэффициентые и сигнал будет переходить на следующих слой. Т.е. фильтр позволяет выделять характерные участки на изображении в соответствии с конфигурацией весовых коэффициентов.

Благодаря такому подходу нейроны каждой группы активируются тогда, когда на участке изображения появляется фрагмент, подходящий под их ядра.

2.6 Пример изображения

Давайте представим, что у нас имеется схематичное изображение дома и мы пропустим его через вот такие ядра:

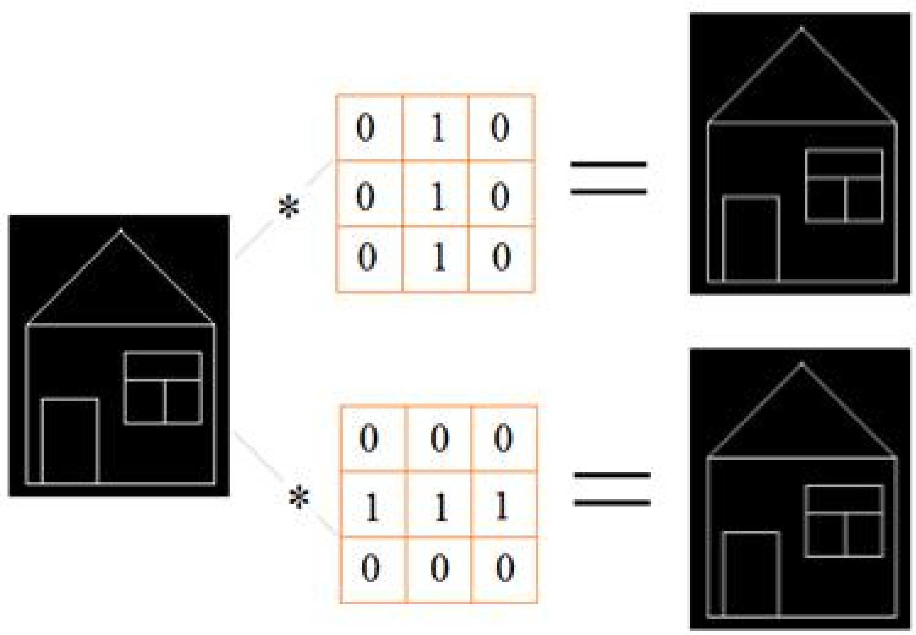

Рис. пример

На выходе получаем отчетливые вертикальные линии в первом случае и горизонтальные – во втором случае. Все остальные линии стали более бледными. То есть, фильтр позволяет выделять характерные участки на изображении в соответствии с конфигурацией весовых коэффициентов. Благодаря такому подходу, нейроны каждой группы активизируются, когда на участке изображения появляется фрагмент, подходящий под их ядра:

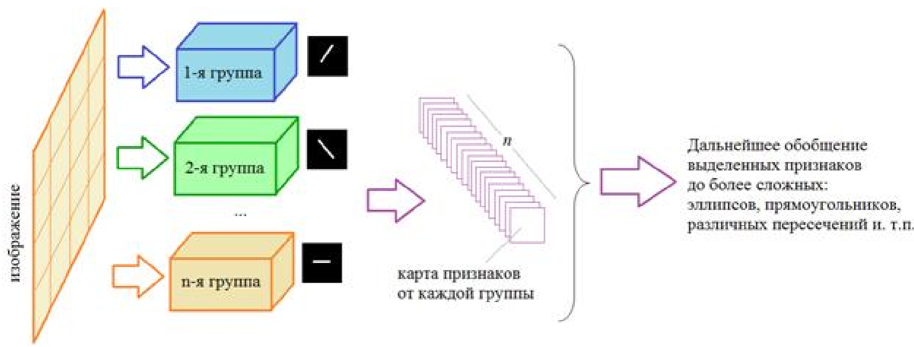

Рис. каналы

И на выходе формируется набор карт признаков, которые называются каналами. Значимые величины в каждой карте показывают наличие признака в строго определенном месте изображения. Если таких признаков будет несколько (на разных участках изображения), то на выходе будут активироваться несколько нейронов, связанных с этими областями. Благодаря этому, следующие слои сверток могут обобщать найденные особенности до более сложных, например, эллипсов, прямоугольников, различных пересечений линий и т.п.

Разумеется, значения карт признаков – это выходы функций активации нейронов, то есть, здесь, все как обычно: сумма (свертка) проходит через функцию активации и формируются выходные значения:

$$v_{k,m} = \sum_{i=1}^{3} \sum_{j=1}^{3} x_{i+k,j+m} \cdot \omega_{ij} + \omega_0,\ k,m = 0,\dots,n$$

2.7 Полноцветное изображение

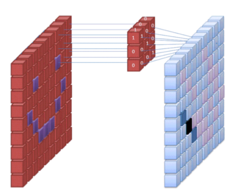

Рис. одноцветное

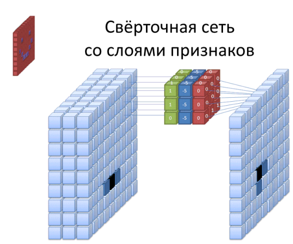

Рис. полноцветное

Но мы рассмотрели простейший вариант, когда на вход подавалось одноканальное изображение, например, в градациях серого. Если обрабатывается полноцветное изображение, представленное, например, тремя цветовыми компонентами RGB, то каждая цветовая компонента сначала преобразовывается своим отдельным, независимым ядром, затем, вычисленные карты признаков, складываются, к ним добавляется смещение и формируется единая итоговая матрица признаков, которая проходит через функцию активации нейронов и получаются выходные значения на соответствующем канале.

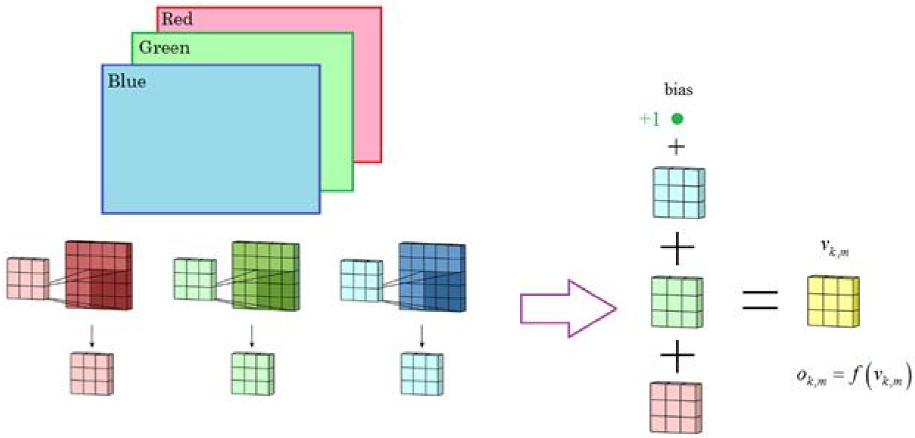

Рис. полноцветное изображение

2.8 Функции активации

После того, как карта признаков изображения была создана, значения, представляющие изображение, передаются через функцию активации или слой активации, например

- ReLU

2.9 MaxPooling

Далее размерность карт сокращается

- MaxPooling – отбор наибольших значений;
- MinPooling – отбор наименьших значений;
- AveragePooling – отбор средних значений.

Давайте представим, что мы хотим вдвое уменьшить линейные размеры карты признаков. В этом случае ее можно покрыть непересекающимися блоками 2х2 пиксела и в каждом блоке оставить только максимальные значения:

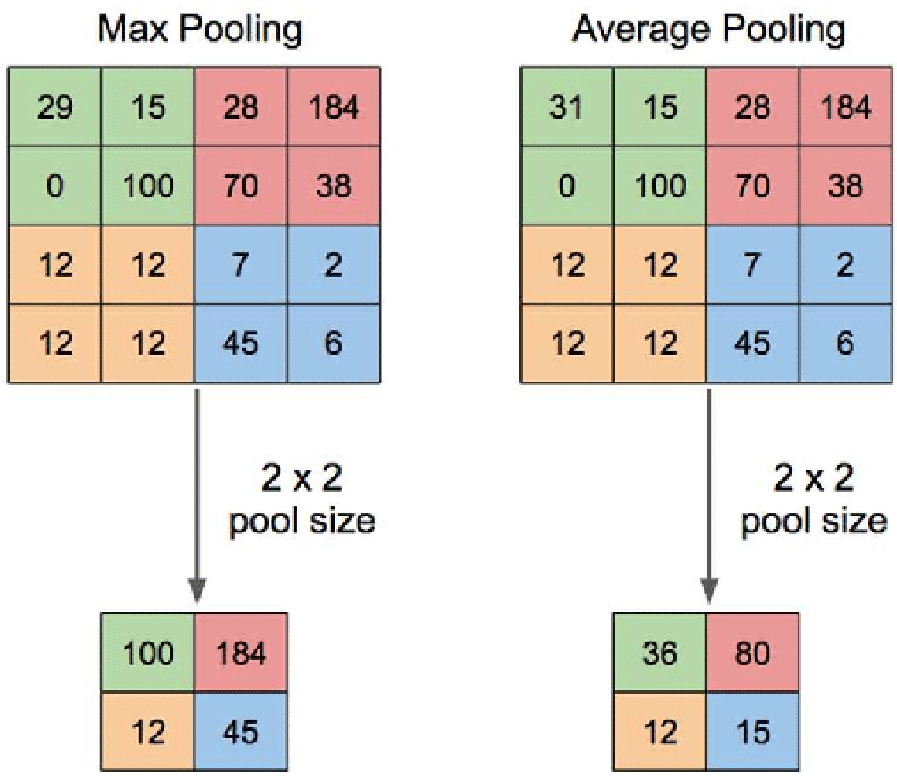

Рис. maxpooling & averagepooling

320x320xRGB — 307 200 байт

Сжатие в 100 раз JPEG (3.08Kb)

Рис. сжатие в 100 раз

2.10 Итоговая архитекрутра

Финальная архитектура будет выглядеть следующим образом: на вход подается изображение 32 на 32 пикселя. В первом слое мы получаем набор из 10 карт признаков и сжимаем их при помощи MaxPooling. Далее те же операции повторяются во втором скрытом слое. Число слоев можно увеличивать и далее. В конце вычисленные карты признаков подаются на вход обычной полносвязной NN. Конечный этап сверточной NN в задачах классификации, обычно, завершается полносвязной NN, на выходе которой получаем вероятности принадлежности к тому или иному классу.

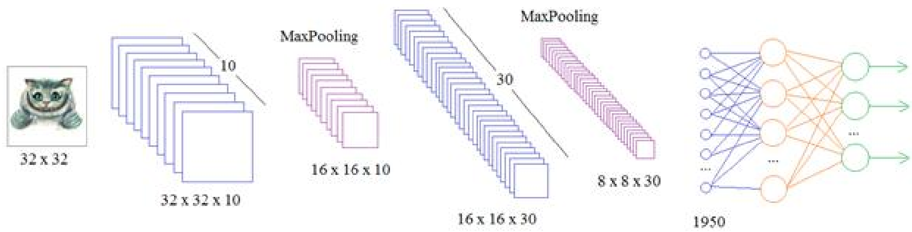

Рис. итоговая архитектура