

Анализ изображений. Нейронные сети.

Шевченко Александр

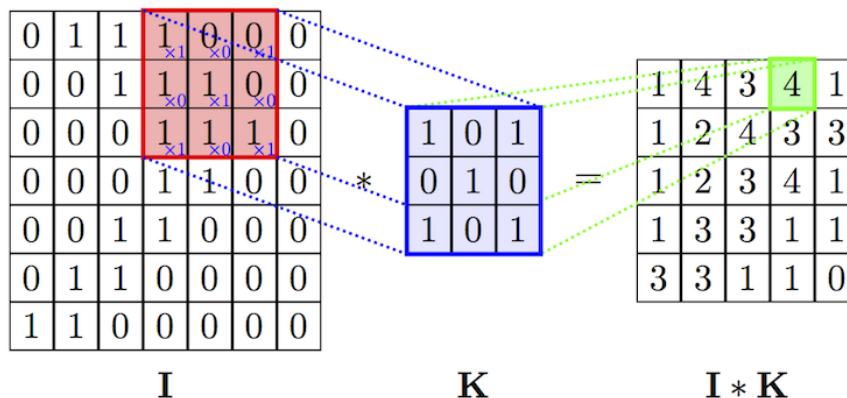
Июль 2017

В 2012 с ростом вычислительных мощностей графических ускорителей широкое распространение для анализа изображений получили сверточные нейронные сети, названные в честь ключевого слоя используемого в них - сверточного.

Устройство и основной принцип работы сверток.

Рассмотрим фильтр, который представляет собой матрицу размера $m \times n$, для определенности касательно примера рассмотрим операцию матричной свертки с фильтром размера 3×3 .

Рис. 1: Матричная свертка



Суть операции заключается в последовательном "применении фильтра"к областям размера 3×3 входной матрицы, под применением понимается покомпонентное умножение матрицы области и фильтра с последующим суммированием элементов.

Необходимо отметить два подхода к применению фильтров в граничных областях. Первый способ заключается в дополнении входной матрицы числами (например нулями, такой подход называется zero padding) по контуру

так, чтобы было возможно применить свертку, к примеру рассмотрим крайний правый верхний элемент, чтобы операция свертки с центром в данном элементе была применима необходимо дополнить справа и сверху область так, чтобы получилась область 3×3 . Другой подход заключается в игнорировании областей к которым свертка не применима, в данном примере в соответствии с данным подходом на выходе получим матрицу размера на 2 меньшего по каждой оси, чем входная.

Так же необходимо сделать важное замечание что размер фильтра в подавляющем большинстве случаев выбирается нечетным по осям, что в частности обосновывается центрированием фильтра при применении к области.

Еще одно важное замечание в разобранный выше примере "вход" являлся матрицей, однако это в большинстве случаев не так, к примеру изображение в формате RGB представляет собой тензор - матрица с третьим измерением толщиной. В таком случае фильтр будет так же являться тензором размера $3 \times 3 \times$ толщина и свертка применяется аналогично только по всей толщине входного тензора. Кроме того обычно фильтр не единственный, а к входу применяется несколько для определенности k , тогда результат применения сверток будет являться тензором толщины k .

Активации

Матричная свертка является линейной операцией, таким образом просто применяя свертки ничего лучше чем линейная модель получить не удастся. Поэтому к выходу свертки обычно применяют нелинейную операцию к примеру сигмоиду. Пусть x результат свертки, тогда применение сигмоиды значит ровно следующее:

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

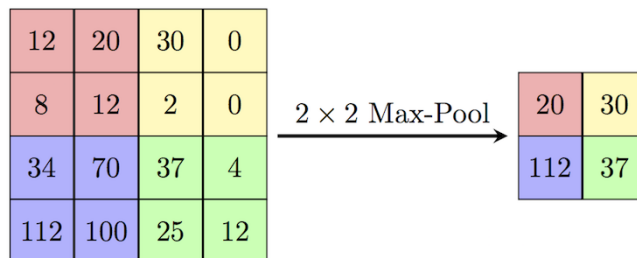
Здесь и далее результат применения свертки и затем нелинейности ко всему входному тензору будем называть feature map, объяснить это можно тем, что фактически выход является набором признаков выделенных из исходного тензора.

Pooling

При свертке и активации применима следующая интуиция: полученный выход отвечает некоторой области исходной feature map. Кроме того можно воспользоваться следующей эвристикой, поскольку во время сверток уже были выделены некоторые признаки, то для дальнейшей обработки столь детальное изображение уже не нужно и его можно уменьшить, например из каждой непересекающейся области 2×2 исходного изображения, брать только пиксель с наибольшим значением, такая операция называется max-rolling с размером соответственно 2×2 . Стоит заметить, что в отличие от сверток, толщина тензора не меняется.

Полносвязный слой

Рис. 2: Max-polling



Применяя свертки, активации и max-pooling можно извлечь из изображения признаки, которые затем можно использовать. Разберем пример с классификацией, для этого необходимо дать определение полносвязного слоя. Пусть после применения вышеописанных операций получен тензор признаков. Полносвязный слой размера k - это слой состоящий из k нейронов каждый из которых связан со всеми нейронами предыдущего слоя с определенными весами, при этом выход нейрона с индексом n без учета активации:

$$\sum_{i=1}^t w_i^n * a_i$$

Где t - количество нейронов в предыдущем слое, a_i - активация i -го нейрона предыдущего слоя, а w_i - соответствующие веса связей.

Таким образом, если классификация изображений происходит на m классов, то на выходе сети хотелось бы иметь вероятности каждого класса, которые суммируются в 1. Следовательно можно сделать несколько полносвязных слоев после сверточных, последний из которых будет иметь количество нейронов отвечающее количеству классов.

Стоит заметить что активации последнего слоя должны суммироваться в 1, для этих целей используется активация soft-max.

Интуиция про обучение

По аналогии с линейными моделями для обучения сетей необходим оптимизируемый функционал, в соответствии с которым обновляются веса в полносвязных слоях и свертки по аналогии с градиентным спуском, однако в этом случае градиент оценивается по одному примеру или малой подвыборке (batch).

Кроме того при обучении по аналогии с линейными моделями изображения лучше отнормировать, поскольку в таком случае существенно снизится разброс градиента по параметрам, что улучшит конечный результат и повысит скорость обучения (впрочем не панацея).