

АВТОЭНКОДЕРЫ И СИАМСКИЕ СЕТИ

Неделя 9

Святослав Елизаров, Борис Коваленко, Артем Грачев

26 января 2018

Высшая школа экономики

АВТОЭНКОДЕРЫ

- Представим себе множество всех возможных изображений разрешения 256×256

- Представим себе множество всех возможных изображений разрешения 256x256
- Всего возможно $255^{256 \cdot 256 \cdot 3} = 255^{196608} = 10^{473145}$
- Очевидно, что большая часть таких «изображений» это просто шум, не несущий никакой полезной информации

- Представим себе множество всех возможных изображений разрешения 256x256
- Всего возможно $255^{256 \cdot 256 \cdot 3} = 255^{196608} = 10^{473145}$
- Очевидно, что большая часть таких «изображений» это просто шум, не несущий никакой полезной информации
- Можно предположить, что внутри этого пространства существует некоторая структура, описывающая осмысленные изображения

- Такая структура называется многообразием
- Очевидно, что элементов, принадлежащих многообразию, значительно меньше чем всех возможных изображений
- Следовательно изображения могут быть закодированы более эффективно

Как нам получить это более оптимальное кодирование?

$$E : X \mapsto H$$

$$D : H \mapsto X$$

Где X – пространство кодируемых объектов

H – скрытое представление меньшей размерности

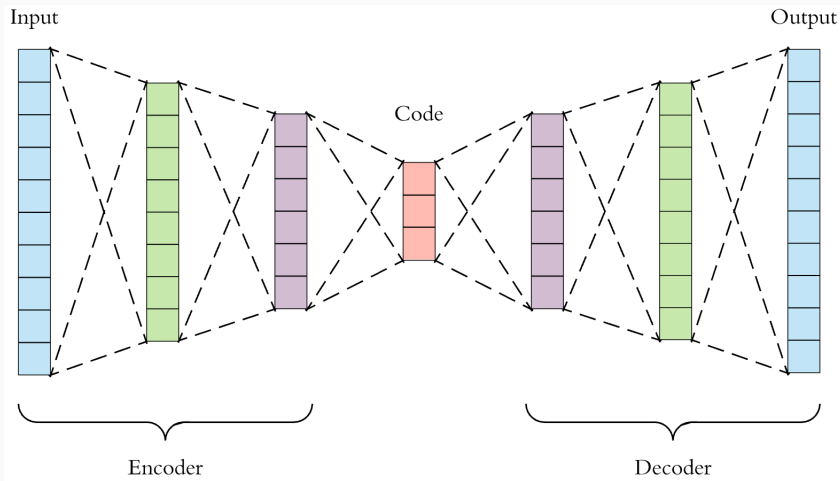
Функции E и D называются encoder и decoder соответственно.

Определим функцию потерь следующим образом:

$$L(x) = d(x, D(E(x)))$$

Где d – метрика в пространстве объектов

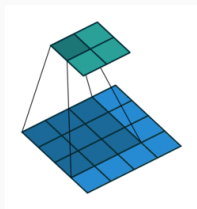
Такая модель называется autoencoder



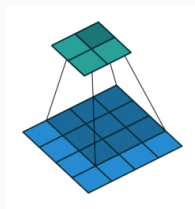
Пример архитектуры encoder-decoder

- Такая модель подразумевает обучение без учителя
- Мы ожидаем, что она найдёт оптимальное представление
- Как писать декодер?

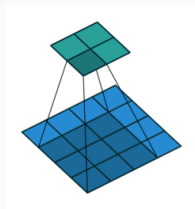
Рассмотрим простой пример свёртки:



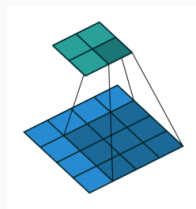
(a) шаг 1



(b) шаг 2



(c) шаг 3



(d) шаг 4

Легко угадать параметры:

- Ядро 3×3
- $stride = 1$
- $padding = 0$

Обозначим ядро:

$$K = \begin{pmatrix} k_{0,0} & k_{0,1} & k_{0,2} \\ k_{1,0} & k_{1,1} & k_{1,2} \\ k_{2,0} & k_{2,1} & k_{2,2} \end{pmatrix}$$

- Операция свёртки может быть представлена, как умножение матриц
- Для этого разматываем оригинал $I_{4 \times 4}$ (аналог операции Flatten в Keras), запишем как I_{flat}
- Запишем свёртку, как матрицу специального вида

ТРАНСПОНИРОВАННЫЕ СВЁРТКИ

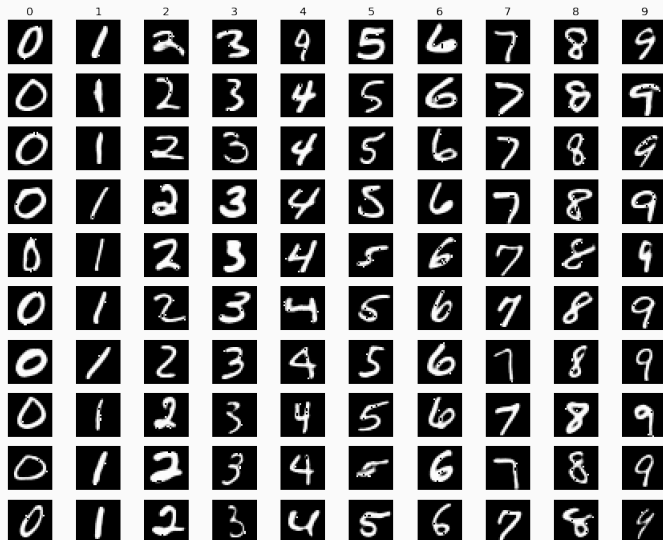
$$C = \begin{pmatrix} k_{0,0} & k_{0,1} & k_{0,2} & 0 & k_{1,0} & k_{1,1} & k_{1,2} & 0 & k_{2,0} & k_{2,1} & k_{2,2} & 0 & 0 & 0 & 0 & 0 \\ 0 & k_{0,0} & k_{0,1} & k_{0,2} & 0 & k_{1,0} & k_{1,1} & k_{1,2} & 0 & k_{2,0} & k_{2,1} & k_{2,2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & k_{0,0} & k_{0,1} & k_{0,2} & 0 & k_{1,0} & k_{1,1} & k_{1,2} & 0 & k_{2,0} & k_{2,1} & k_{2,2} & 0 \\ 0 & 0 & 0 & 0 & 0 & k_{0,0} & k_{0,1} & k_{0,2} & 0 & k_{1,0} & k_{1,1} & k_{1,2} & 0 & k_{2,0} & k_{2,1} & k_{2,2} \end{pmatrix}$$

Теперь мы можем вычислить операцию свёртки следующим образом:

$$C \cdot I_{flat} = O_{flat}$$

Где O_{flat} является плоской версией карты активации

- **Transposed convolution** расширяет карту активации
- Может быть использовано для генерации изображений



ШАХМАТНАЯ ДОСКА

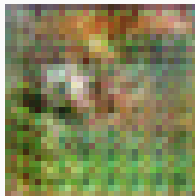
Изучая результаты работы разных алгоритмов, или, возможно, самостоятельно обучая модели, вы могли столкнуться так называемым эффектом шахматной доски (checkerboard artifacts)



(e) птица



(f) лодка



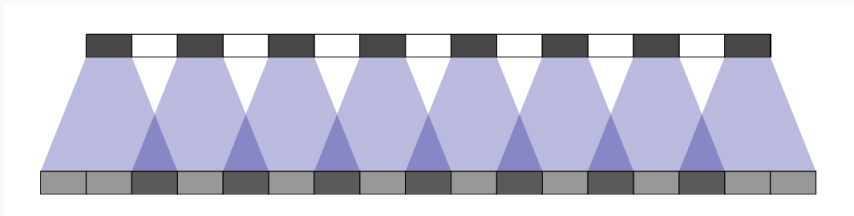
(g) олень



(h) самолёт

Откуда они берутся?

Odena, et al., "Deconvolution and Checkerboard Artifacts Distill, 2016.



Мы видим, что результаты применения каждой транспонированной свёртки «наслаиваются». Отсюда и возникает эффект шахматной доски.

- Существует несколько способов избежать эффекта шахматной доски в изображениях, сгенерированных нейронной сетью

- Существует несколько способов избежать эффекта шахматной доски в изображениях, сгенерированных нейронной сетью
- Размер ядра свёртки должен быть кратен stride
- Использовать up-sampling (nearest neighbors) и обычный свёртки

ЗАЧЕМ НУЖНЫ АВТОЭНКОДЕРЫ?

Зачем нужны автоэнкодеры?

ЗАЧЕМ НУЖНЫ АВТОЭНКОДЕРЫ?

Зачем нужны автоэнкодеры?

- Извлечение признаков
- Подавление шума
- Обучение расстояния
- Восстановление распределения

Предобучим автоэнкодер, а потом используем его в другой модели

$$L(x) = d(x, D(E(x + \epsilon)))$$

Где ϵ – шум, например Гауссовский

При помощи архитектуры энкодер-декодер можно решать задачу обучения расстояния (distance learning)

1. Дано множество объектов X
2. Дана разметка по сходству для пар объектов. $\{0, 1\}$
3. Необходимо построить метрику на объектах таких образом, чтобы её значения соответствовали представленной разметке

$$L(x_1, x_2, y) = \|d(E(x_1), E(x_2)) - y\|_2$$