

Глубокое обучение и вообще

Соловей Влад и Шигапова Фирюза

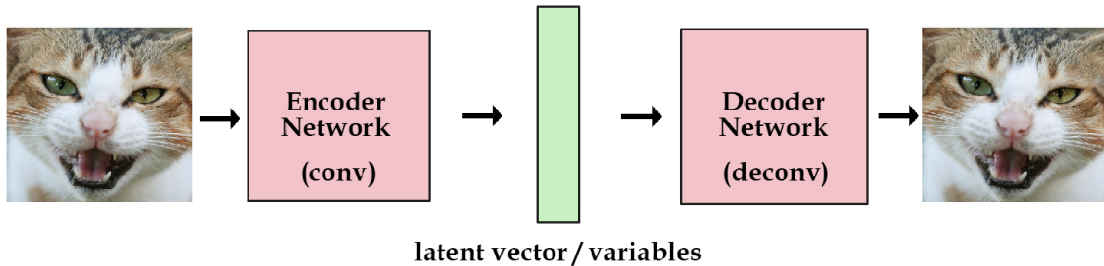
11 февраля 2021 г.

Посиделка 16: Автоэнкодеры

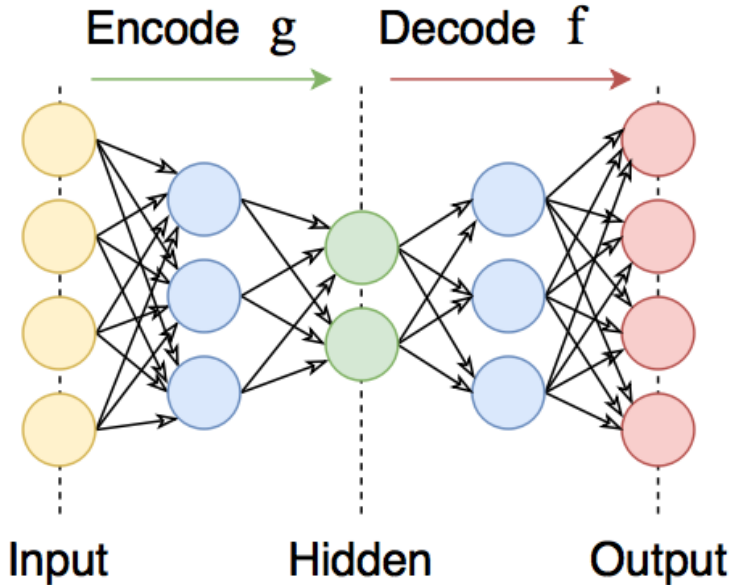
Autoencoder

1. Когда данных много, мы хотим понизить их размерность. Классическое машинное обучение позволяете делать это с помощью метода главных компонент, tsne и других методов. Нейросети также позволяют решать подобную задачу сжатия с минимальными потерями.
2. Понижение размерности — задача обучения без учителя
3. Давайте превратим её в обучение с учителем!

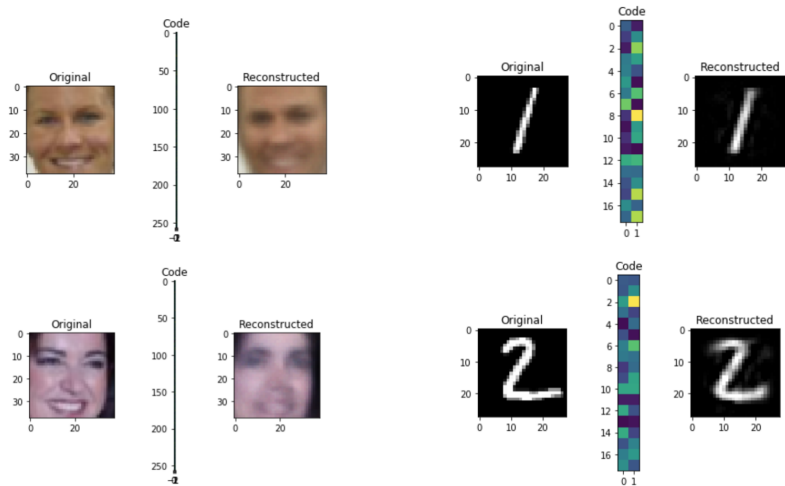
Autoencoder



Autoencoder



Примеры сжатия



Как используют

1. Для предобучения сетей. Именно так в 2005 началась революция.
2. Скрытое представление признаков можно использовать в других моделях в качестве фичей.
3. Генерация похожих объектов

Типы автокодировщиков

Denoise автокодировщик

В обычном автокодировщике восстанавливаем следующую последовательность: $x = decoder(encoder(x))$, изменяя 2 функции $encoder(x)$ и $decoder(x)$. Loss в нашем случае -

$$L(x, decoder(encoder(x)))$$

Тождественно не можем выучить из-за искусственного ограничения количества нейронов в середине.

1. Однослойной автокодировщик по своему действию совпадает с PCA
2. Если задаем многослойный автокодировщик может находить достаточно сложные особенности в данных (при правильной архитектуре любые)
3. Можно делать сетки любых типов (сверточные, рекуррентные, полносвязные)

Denoise автокодировщик

Мы пытаемся не просто восстановить выход по входу, но и ещё искусственно добавляем шум к входным данным. По сути мы пытаемся решить следующую задачу $x = \text{decoder}(\text{encoder}(\hat{x}))$, где \hat{x} зашумленные входные данные. Обычно зашумляют данные через добавление гауссовского шума, но можно делать разное. Например, для картинок можно затемнять какую-либо часть изображения. Остальной процесс обучения совпадает с обычным автокодировщиком.

Разреженный автокодировщик

Теперь мы к нашему лосу добавляем регуляризатор

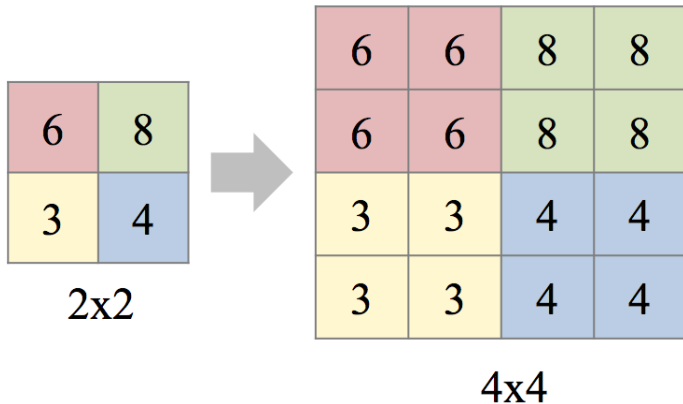
$$L(x, \text{decoder}(\text{encoder}(x))) + \omega(h),$$

где $\text{decoder}(h)$ -выход декодера, $h = \text{encoder}(x)$ - выход энкодера. Обычно ограничение накладывают на энкодер и это L1 или L2 норма.

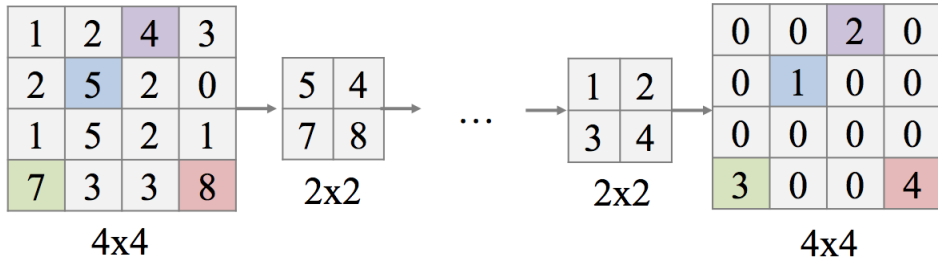
Такой автокодировщик не сможет полностью выучить картинку из-за штрафа при любой архитектуре. Он может расширяться к выходу, пытаясь разложить сигнал на множество статистически независимых сигналов. Используют его также как и обычный автокодировщик, если требуется чтобы получающиеся латентные векторы были более линейно-независимые. Из-за того, что он пытается разложить один сигнал на множество, иногда его для разложения сигнала на составляющие - аналог вейвлет преобразований для аудио.

Разворачивание картинки назад

Nearest neighbor unpooling

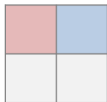


Max unpooling

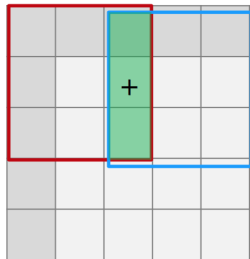


Learnable unpooling: Transpose convolution

Input: 2x2



Input gives
weight for
filter



Stride: 2

Output: 4x4

- Каждую клетку надо распаковать в 4 клетки \Rightarrow свёртка 3×3 со сдвигом 2

Пример:

