

Кластеризация изображений по визуальному подобию с помощью вариационных автоэнкодеров

А. С. Коваленко

ЮФУ

Институт математики, механики и компьютерных наук им. И. И. Воровича
научный руководитель: доцент кафедры ПМП, к.ф.-м.н. Я. М. Демяненко

20 апреля 2018 г.

1 Введение

- Постановка задачи
- Обзор подходящих алгоритмов

2 Автоэнкодеры

- Принцип работы
- Разряженные автоэнкодеры
- Вариационные автоэнкодеры

3 Применение вариационных автоэнкодеров к задаче

- Архитектура построенного автоэнкодера
- Скорость обучения
- Пример работы
- Кластеризация
- Поиск похожих по визуальному подобию

4 Список литературы

Введение

Постановка задачи

Разметка данных всегда трудоемкий процесс. Поэтому алгоритмы машинного обучения не требующие данного этапа актуальны. Класс вариационных автоэнкодеров позволяет решить данную проблему.

Поставим задачу:

Пусть имеется набор изображений:

$$X = \{I_m\}_{m=1}^N \quad (1)$$

Требуется разбить данный набор на K классов и ввести операцию сравнения.

Для произвольного изображения найдем M схожих объектов из множества X , отсортированных по убыванию визуального подобия:

$$I_{input} \notin X, S = \{I_{m_k}\}_{k=1}^M$$

Для решения поставленной задачи могут подойти следующие подходы неконтролируемого обучения и обучения с подкреплением:

- Сиамские нейронные сети
- Ключевые точки (не является алгоритмом машинного обучения)
- Автоэнкодеры

Для решения поставленной задачи могут подойти следующие подходы неконтролируемого обучения и обучения с подкреплением:

- Сиамские нейронные сети
- Ключевые точки (не является алгоритмом машинного обучения)

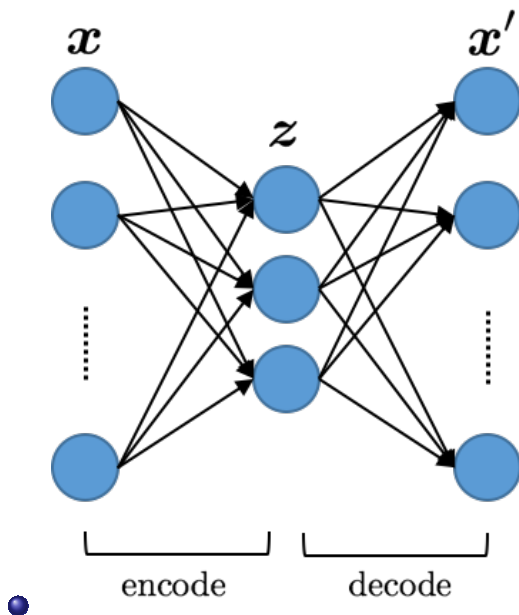
Автоэнкодеры

Автоэнкодеры

Принцип работы

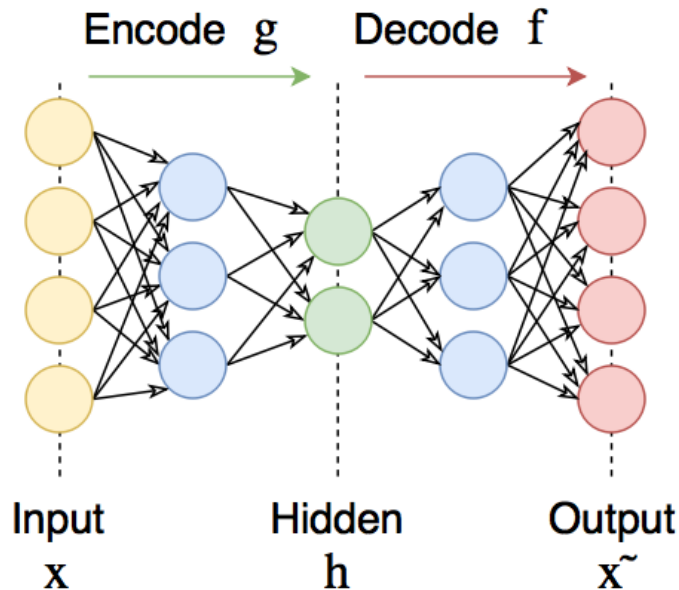
Определение:

Автоэнкодеры — это нейронные сети прямого распространения, которые восстанавливают входной сигнал на выходе.



Автоэнкодеры

Принцип работы



g – энкодер, f – декодер,

x – сигнал, h – код сигнала

$$h = g(x), \quad x = f(h)$$

$$x = f(g(x)) \quad (2)$$

Автоэнкодеры

Разряженные автоэнкодеры

Автоэнкодер при обучении стремится аппроксимировать функцию (2):

$$x = f(g(x)),$$

минимизируя заданный функционал ошибки:

$$\min_{\omega} L(x, f(g(x))), \quad (3)$$

где ω - параметры автоэнкодера

Определение:

Разряженным называется автоэнкодер, для которого критерий обучения включает также минимизацию разреженности $\Omega(h)$ на кодовом слове h :

$$\min_{\omega} L(x, f(g(x))) + \Omega(h),$$

где $\Omega(h)$ - обычный регуляризатор (пусть L_1), $\Omega(h) = \lambda * \|h\|$

Рассмотрим работу декодера как некоторый процесс генерации данных X , зависящий от скрытых переменных Z - случайных величин.

Пусть:

- $P(X)$ - вероятностное распределение изображений
- $P(Z)$ - вероятностное распределение скрытых переменных
- $P(Z|X)$ - вероятностное распределение скрытых параметров при заданном изображении X , рассматривается как энкодер
- $P(X|Z)$ - вероятностное распределение изображений при заданных скрытых переменных Z , рассматривается как декодер

Справедливо:

$$P(X) = \int_Z P(X|Z)P(Z)dZ \quad (4)$$

Пространство может быть высокоразмерным, поэтому напрямую хорошо приблизить интеграл не получится. Воспользуемся тем, что для заданного X соответствует небольшое подмножество Z , а для остальных вероятность $P(X|Z) \rightarrow 0$. Также при приближении будем семплировать из “оптимальных” Z .

Чтобы понимать какие Z “оптимальные” вводится распределение $Q(Z|X)$, которое будет показывать для X распределение $Z \sim Q$, которое приводит к этому X .

Пусть $Q(Z|X)$ будет нормальным распределением:

$$Q(Z|X) = N(\mu(X), \Sigma(X)), \quad (5)$$

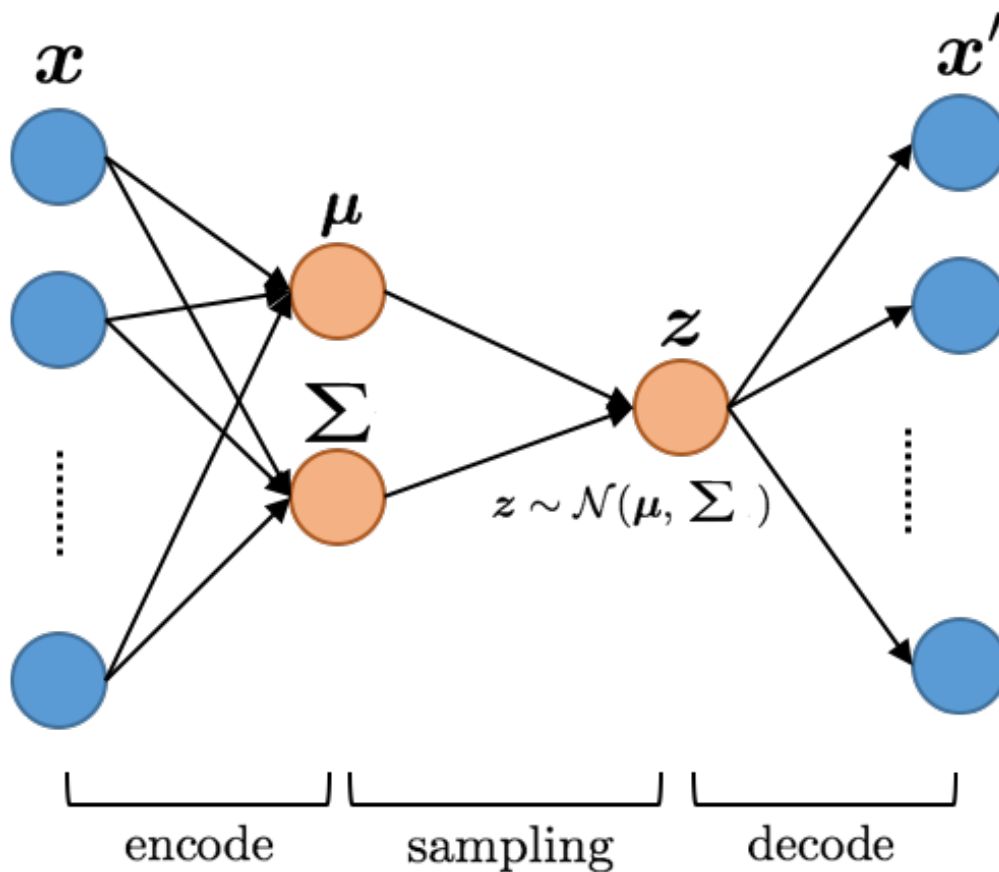
где μ и Σ - среднее и матрица ковариации для нормального распределения.

Таким образом семплирование скрытых параметров идет из нормального распределения с параметрами μ , Σ . $P(Z|X)$ также является нормальным распределением.

Функционал ошибки для обучения (3) выводится из рассмотрения расстояния Кульбака-Лейблера между $Q(Z|X)$ и $P(Z|X)$.

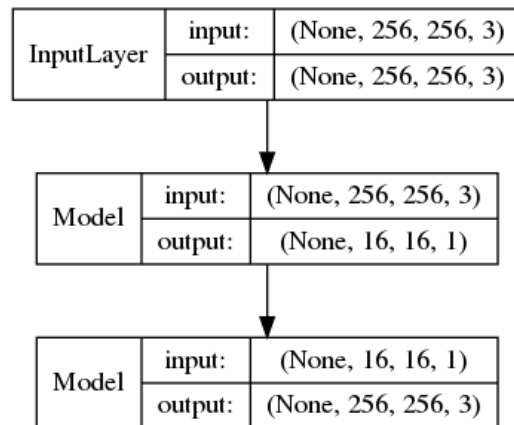
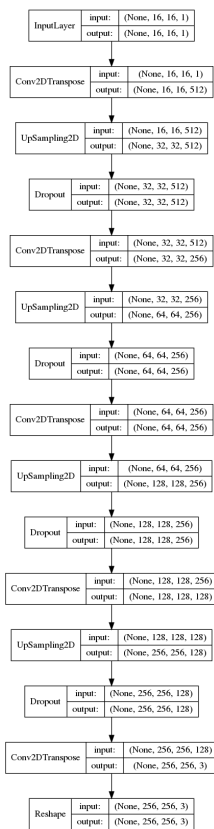
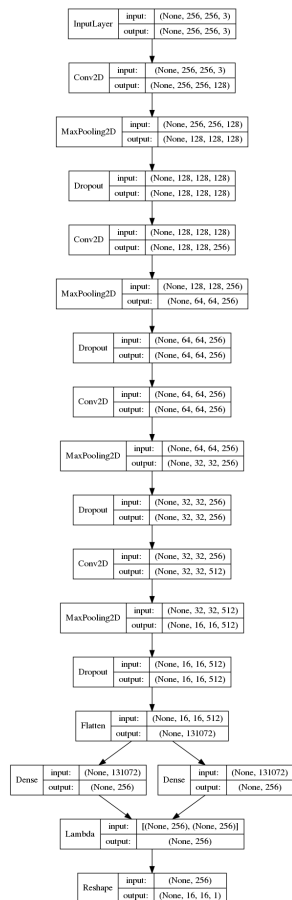
Автоэнкодеры

Вариационные автоэнкодеры



Применение вариационных автоэнкодеров к задаче

Архитектура построенного автоэнкодера



Применение вариационных автоэнкодеров к задаче

Скорость обучения

Layer (type)	Output Shape	Param #
input_3 (InputLayer)	(None, 256, 256, 3)	0
Encoder (Model)	(None, 16, 16, 1)	74221184
Decoder (Model)	(None, 256, 256, 3)	18326659
Total params: 92,547,843		
Trainable params: 92,547,843		
Non-trainable params: 0		

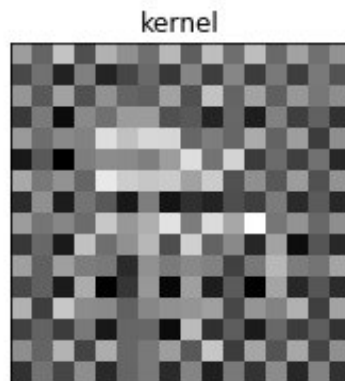
Используемые вычислительные и программные ресурсы:

- CPU: Intel i7-4770K, RAM: 16 Gb
- GPU: Nvidia GTX 1080Ti 11 Gb
- System: Linux Mint 18.2 Sonya x86_64
- Realisation: Keras(2.1.3) + Tensorflow(1.1.0) + CuDNN(7.1)

Время обучения одной эпохи на наборе из 154 644 изображений составляет 5 часов 7 минуты 23 секунды. Оптимальное количество эпох - от 3 (15 часов).

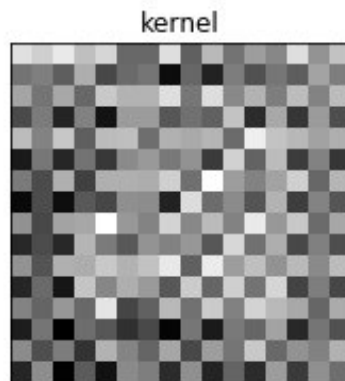
Применение вариационных автоэнкодеров к задаче

Пример работы вариационного автоэнкодера



Применение вариационных автоэнкодеров к задаче

Пример работы вариационного автоэнкодера



Применение вариационных автоэнкодеров к задаче Кластеризация

Построим множество H следующим образом:

$$H = \{g(x) | x \in X\}, \quad (6)$$

где X - множество (1), g - вариационный автоэнкодер.

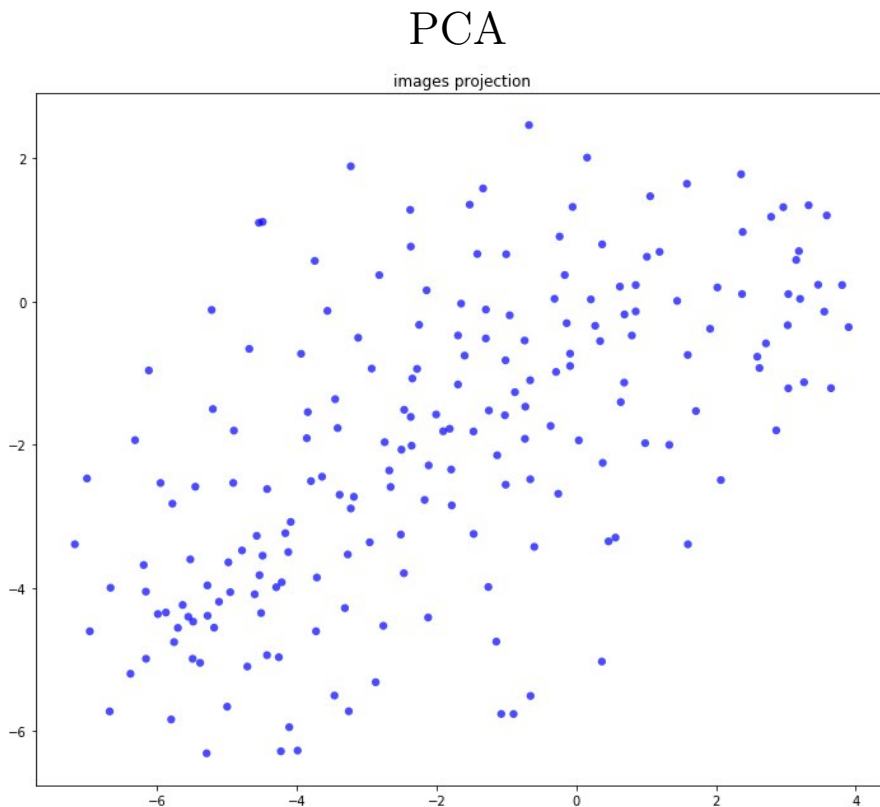
Теперь понизим размерность множества H с помощью метода главных компонент (PCA) или алгоритма распределенного стохастического выделения соседей (t-SNE):

$$\hat{H} = \text{t-SNE}(H), \quad \forall h \in \hat{H} \Rightarrow \dim(h) = 2 \quad (7)$$

К множеству \hat{H} можно применять классические алгоритмы кластеризации данных, к примеру k -средних, предварительно выбрав количество классов.

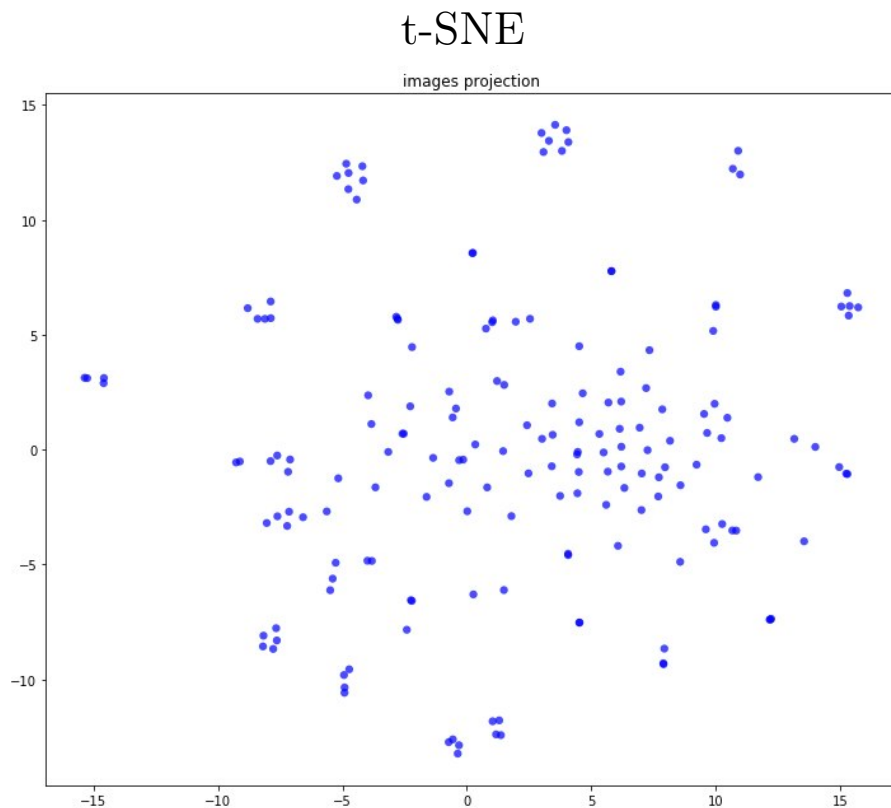
Применение вариационных автоэнкодеров к задаче Кластеризация

Сравнение PCA и t-SNE



Применение вариационных автоэнкодеров к задаче Кластеризация

Сравнение PCA и t-SNE



Применение вариационных автоэнкодеров к задаче

Поиск похожих по визуальному подобию

Пусть H , как и в (6):

$$H = \{g(x) | x \in X\}$$

Поделиствуем энкодером на входное изображение:

$$h_{input} = g(I_{input})$$

Теперь найдем элементы множества S , это будут M -ближайших элемента из множества H по евклидовой метрике.

P. Flash.

Machine learning: The art and science of algorithms that make sense of data.

2018.

I. Goodfellow.

Deep learning.

2018.

S. Haykin.

Neural networks. a comprehensive foundation.

2016.