Вариационные автоэнкодеры

Шешукова Марина

16 ноября 2021

Определение

Уменьшение размерности - это процесс уменьшения количества признаков, описывающих некоторые данные.

Уменьшение размерности

Определение

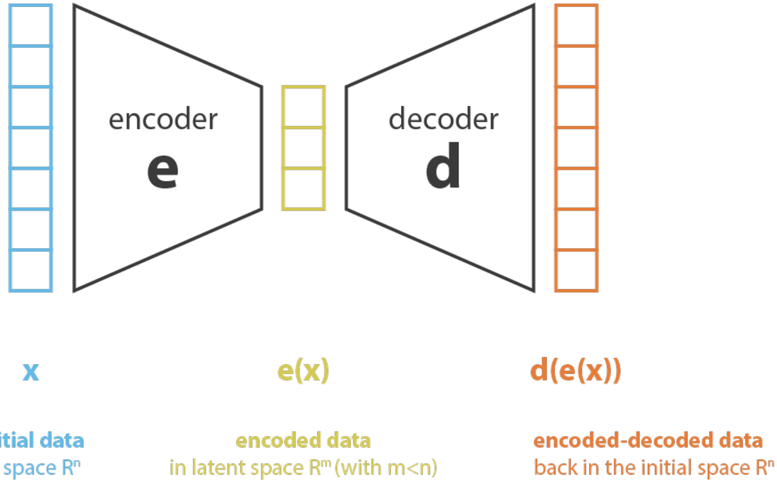
Уменьшение размерности - это процесс уменьшения количества признаков, описывающих некоторые данные.

Основная цель метода уменьшения размерности - найти лучшую пару кодировщик/декодировщик среди заданного семейства.

$$(e*, d*) = arg \min_{e \in F, d \in \mathcal{D}} l(x, d(e(x)))$$

$l(x, y)$– мера ошибки между входными и декодированными данными

Уменьшение размерности

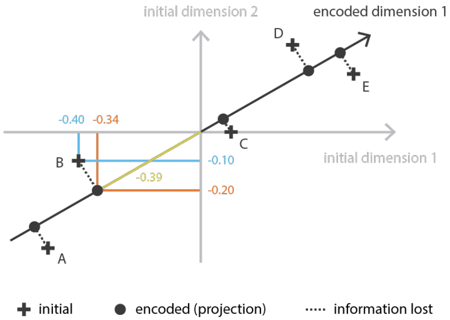


Метод главных компонент

Метод главных компонент ищет наилучшее линейное подпространство исходного пространства таким образом, чтобы ошибка аппроксимации данных их проекциями на это подпространство была как можно меньше.

Метод главных компонент

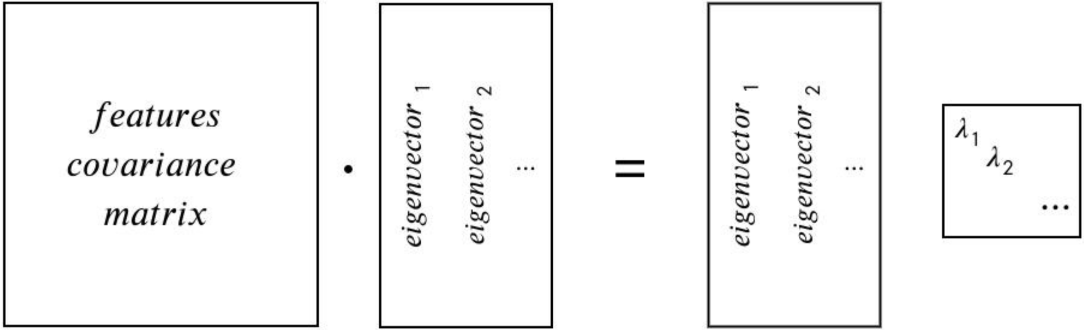
Метод главных компонент ищет наилучшее линейное подпространство исходного пространства таким образом, чтобы ошибка аппроксимации данных их проекциями на это подпространство была как можно меньше.



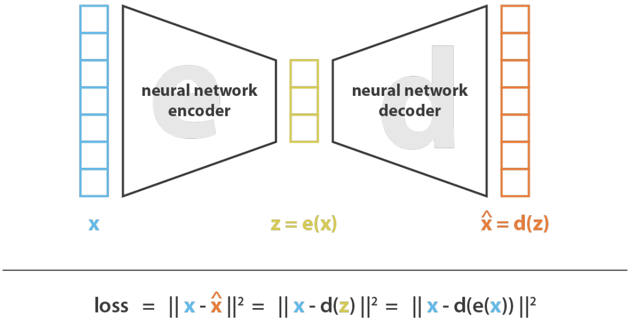
Point	Initial	Encoded	Decoded
A	(-0.50, -0.40)	-0.63	(-0.54, -0.33)
B	(-0.40, -0.10)	-0.39	(-0.34, -0.20)
C	(0.10, 0.00)	0.09	(0.07 0.04)
D	(0.30, 0.30)	0.41	(0.35, 0.21)
E	(0.50, 0.20)	0.53	(0.46, 0.27)

Метод главных компонент

Мы ищем кодировщик в семействе матриц размера $e \times d$, причем его строки ортонормированные. Декодировщик ищется соответственно в семействе матриц $d \times e$.



В терминах метода уменьшения размерности, мы имеем E- архитектура нейросети кодировщика, D - архитектура нейросети декодировщика. В качестве функции потерь используют среднеквадратичную ошибку, а обучают обычно градиентным спуском.



Автоэнкодер

Предположим, что кодировщик и декодировщик состоят только из одного линейного слоя. Тогда, как и в методе главных компонент автоэнкодер ищет лучшее линейное подпространство для проецирования данных с минимальной потерей информации при этом.

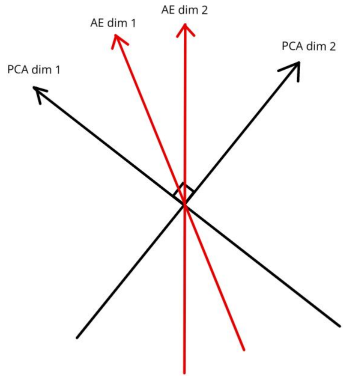


Рис.: Автоэнкодер и метод главных компонент получают разный базис в латентном пространстве.

Автоэнкодер

Предположим, что и кодировщик и декодировщик являются глубокими нейросетями. Интуитивно, кодировщик теоретически мог бы взять наши N изначальных точек данных и кодировать их как целое число m на действительной оси, а соответствующий декодировщик может выполнить обратное преобразование без потери информации во время процесса.

Проблема: латентное пространство может быть не непрерывным.

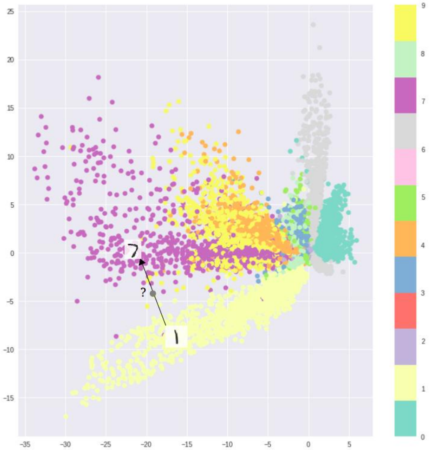


Рис.: Обучение автоэнкодера на датасете MNIST.

Вариационный автоэнкодер

- Теперь мы кодируем ввод не как одну точку, а как распределение по латентному пространству.

Вариационный автоэнкодер

- Теперь мы кодируем ввод не как одну точку, а как распределение по латентному пространству.
- После этого мы выбираем точку из получившегося распределения, считаем что эта точка из латентного пространства.

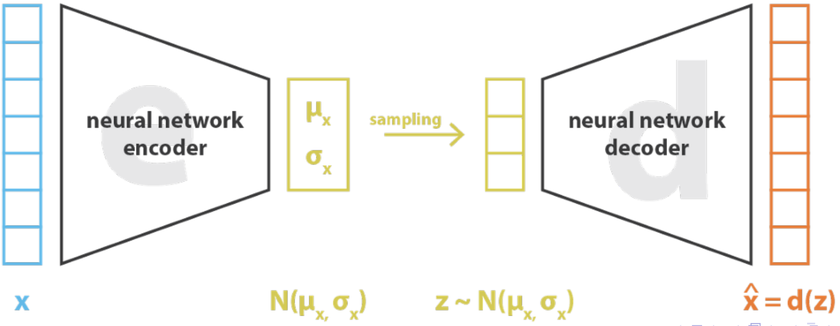
Вариационный автоэнкодер

- Теперь мы кодируем ввод не как одну точку, а как распределение по латентному пространству.
- После этого мы выбираем точку из получившегося распределения, считаем что эта точка из латентного пространства.
- Точка из латентного пространства декодируется и считается функция потерь.

Вариационный автоэнкодер

- Теперь мы кодируем ввод не как одну точку, а как распределение по латентному пространству.
- После этого мы выбираем точку из получившегося распределения, считаем что эта точка из латентного пространства.
- Точка из латентного пространства декодируется и считается функция потерь.

$$loss = ||x - d(z)||2 + KL(\mathcal{N}(\mu_x, \sigma_x), \mathcal{N}(0, I))$$



Интуиция регуляризации

Чтобы генерировать новые данные нам нужно, чтобы латентное пространство обладало следующими свойствами:

Интуиция регуляризации

Чтобы генерировать новые данные нам нужно, чтобы латентное пространство обладало следующими свойствами:

- Непрерывность, то есть две близкие точки в латентном пространстве не должны давать два совершенно разных результата после декодирования.

Интуиция регуляризации

Чтобы генерировать новые данные нам нужно, чтобы латентное пространство обладало следующими свойствами:

- Непрерывность, то есть две близкие точки в латентном пространстве не должны давать два совершенно разных результата после декодирования.
- Полнота, то есть точка, полученная из латентного распределения, после декодирования должна нести значимое содержание.

Интуиция регуляризации

Чтобы генерировать новые данные нам нужно, чтобы латентное пространство обладало следующими свойствами:

- Непрерывность, то есть две близкие точки в латентном пространстве не должны давать два совершенно разных результата после декодирования.
- Полнота, то есть точка, полученная из латентного распределения, после декодирования должна нести значимое содержание.

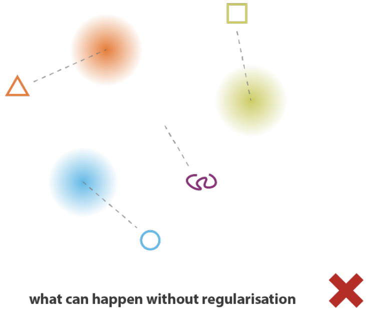


Рис.: Что может случиться без регуляризации.

Регуляризация

Идея: нужно регуляризовать как вектор средних значений так и ковариационную матрицу в латентном пространстве.

Реализация на практике: делаем распределения в латентном пространстве близкими к стандартному гауссовскому распределению. Так, среднее значение будет близко к 0, что предотвращает большое расстояние между закодированными распределениями, а ковариационные матрицы будут близки к единичной, что предотвращает точечные распределения.

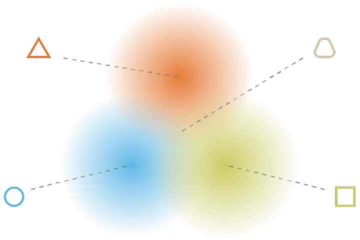
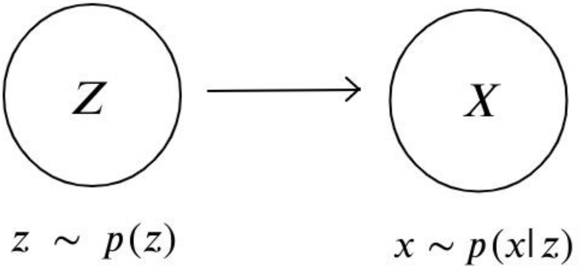


Рис.: Регуляризация.

Некоторые предположения:

- Пусть x – переменная, которая представляет наши данные.
- Предполагаем, что x генерируется из условного распределения $p(x|z)$, при условии латентной переменной z, которая имеет априорное распределение $p(z)$.



Еще немного математики

Некоторые предположения:

- Декодировщик определен распределением $p(x|z)$ – которое описывает распределение декодированной переменной по закодированной.
- Кодировщик определен распределением $p(z|x)$ – которое описывает распределение закодированной переменной по декодированной.

Утверждение

Вспомним формулу Байеса, которое связывает априорное распределение $p(z)$, апостериорное $p(z|x)$ и $p(x|z)$.

$$p(z|x) = \frac{p(x|z)p(z)}{p(x)} = \frac{p(x|z)p(z)}{\int p(x|u)p(u)du}$$

Предполагаем, что $p(x) \sim \mathcal{N}(0, I)$ и $p(x|z) \sim \mathcal{N}(f(z), cI)$

Замечание

Если $f(z) = z$, то $p(z|x)$ тоже гауссовское. Поэтому мы пытаемся аппроксимировать $p(z|x)$ гауссовским распределением $p_x(z) \equiv \mathcal{N}(g(x), h(x))$

И еще математики

Предполагаем, что $p(x) \sim \mathcal{N}(0, I)$ и $p(x|z) \sim \mathcal{N}(f(z), cI)$

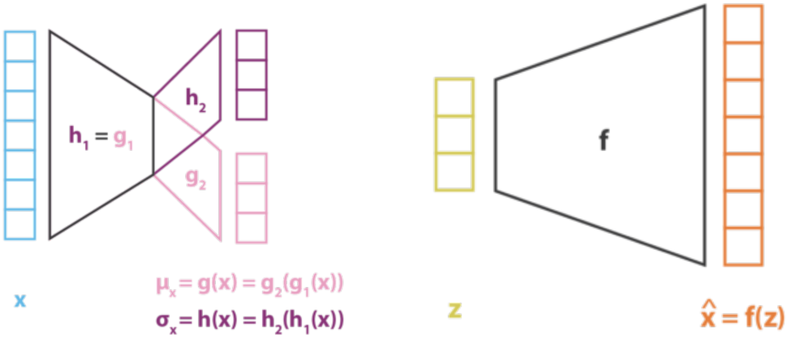
Замечание

Если $f(z) = z$, то $p(z|x)$ тоже гауссовское. Поэтому мы пытаемся аппроксимировать $p(z|x)$ гауссовским распределением $p_x(z) \equiv \mathcal{N}(g(x), h(x))$

$$\begin{aligned}
(g^*, h^*) &= \underset{(g,h)\in G\times H}{\arg\min}\ KL(q_x(z), p(z|x)) \\
&= \underset{(g,h)\in G\times H}{\arg\min}\ \left(\mathbb{E}_{z\sim q_x}(\log q_x(z)) - \mathbb{E}_{z\sim q_x}\left(\log \frac{p(x|z)p(z)}{p(x)}\right)\right) \\
&= \underset{(g,h)\in G\times H}{\arg\min}\ (\mathbb{E}_{z\sim q_x}(\log q_x(z)) - \mathbb{E}_{z\sim q_x}(\log p(z)) - \mathbb{E}_{z\sim q_x}(\log p(x|z)) + \mathbb{E}_{z\sim q_x}(\log p(x))) \\
&= \underset{(g,h)\in G\times H}{\arg\max}\ (\mathbb{E}_{z\sim q_x}(\log p(x|z)) - KL(q_x(z), p(z))) \\
&= \underset{(g,h)\in G\times H}{\arg\max}\ \left(\mathbb{E}_{z\sim q_x}\left(-\frac{||x - f(z)||^2}{2c}\right) - KL(q_x(z), p(z))\right)
\end{aligned}$$

Итоговая формула

$$(f^*, g^*, h^*) = \underset{(f,g,h)\in F\times G\times H}{\arg\max} \left(\mathbb{E}_{z\sim q_x}\left(-\frac{||x - f(z)||^2}{2c} \right) - KL(q_x(z), p(z)) \right)$$



$$\xi \sim \mathcal{N}(0,1); z = h(x)\cdot\xi + g(x)$$

Список литературы.

- https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73
- https://neurohive.io/ru/osnovy-data-science/variacionnyj-avtojenkoder-vae/