

Глубокое обучение для обработки изображений

Лекция 6

Обучение с учителем и без учителя

Обучение с учителем:

Есть входные данные x и
лейблы/метки y

Цель: обучить функцию $f: x \rightarrow y$

Классификация, регрессия,
детекция объектов,
семантическая сегментация и т.д.

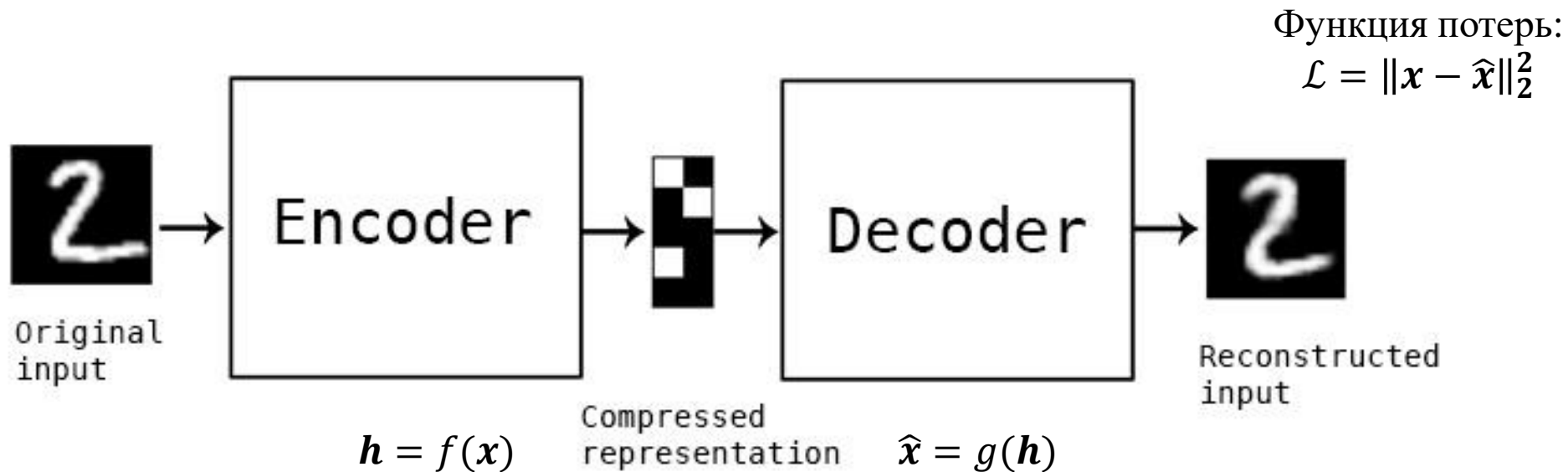
Обучение без учителя:

Есть только входные данные x

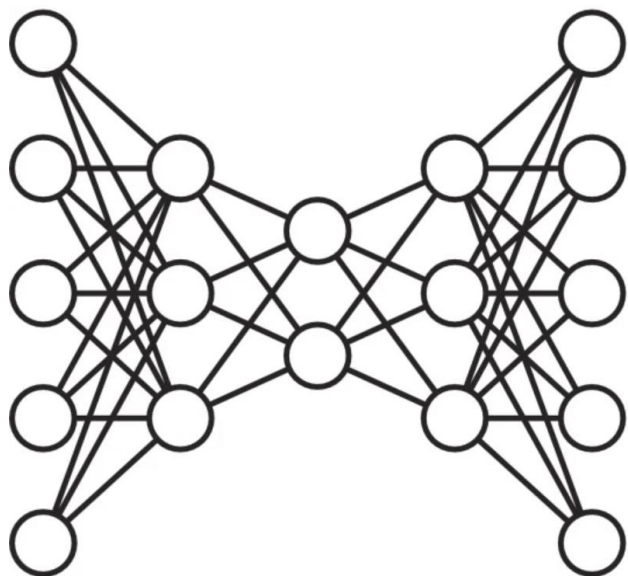
Цель: найти некоторые
особенности, структуру данных

Кластеризация, понижение
размерности и т.д.

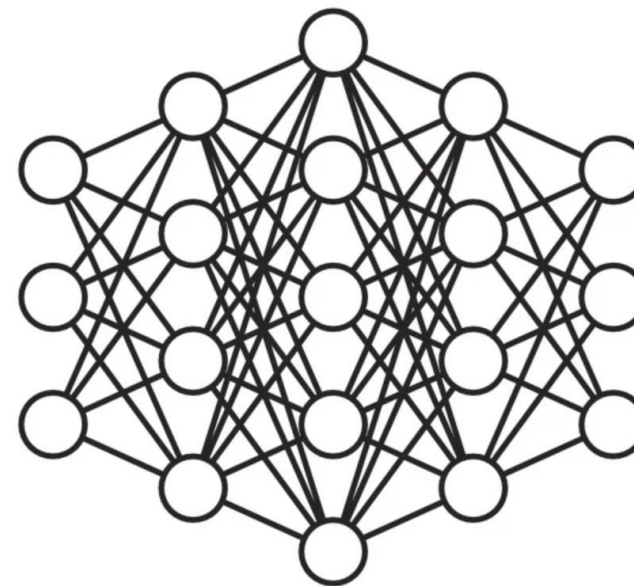
Автокодировщик (Autoencoder)



Понижающие и повышающие автокодировщики



С помощью архитектуры **понижающего (undercomplete) автокодировщика** мы пытаемся наилучшим образом сжать данные, чтобы затем их возможно было восстановить



Повышающий автокодировщик (overcomplete)
Вместо того, чтобы ограничивать емкость модели, стремясь сделать кодировщик и декодер мелкими, а размер кода малым, используют функцию потерь с регуляризацией, которая побуждает модель приобретать полезные свойства

Регуляризированные автокодировщики

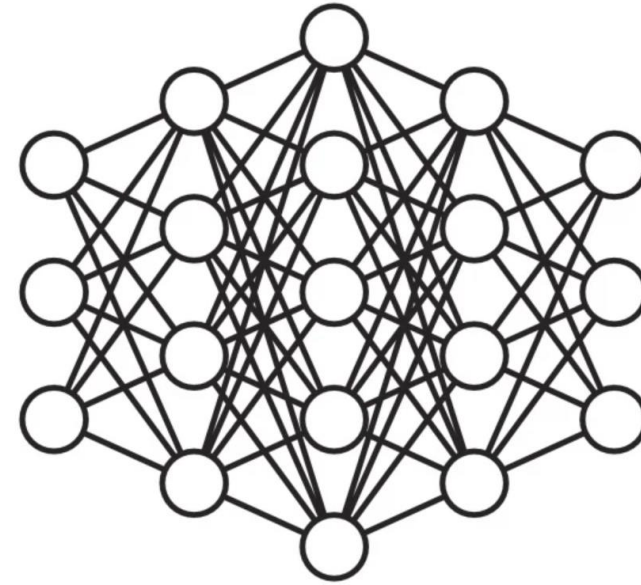
Для *разреженного* (sparse autoencoder) и *сжимающего* (contractive autoencoder, CAE) автокодировщиков в функцию потерь добавляется штрафное слагаемое:

$$\mathcal{L}(\mathbf{x}, g(f(\mathbf{x}))) + \Omega(\mathbf{h}, \mathbf{x}).$$

Например, для CAE штраф имеет вид:

$$\Omega(\mathbf{h}, \mathbf{x}) = \lambda \sum_i \|\nabla h_i\|^2,$$

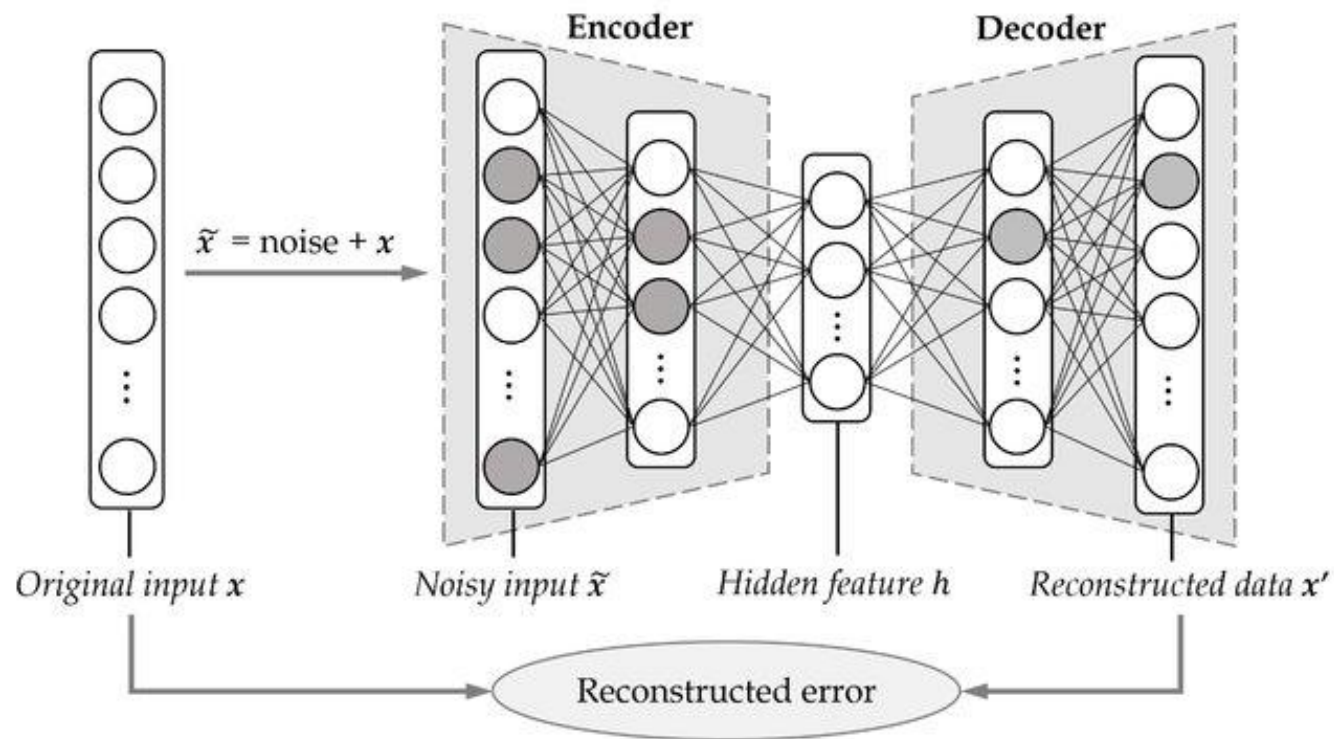
который способствует обучению функции, которая слабо изменяется при малом изменении \mathbf{x} .

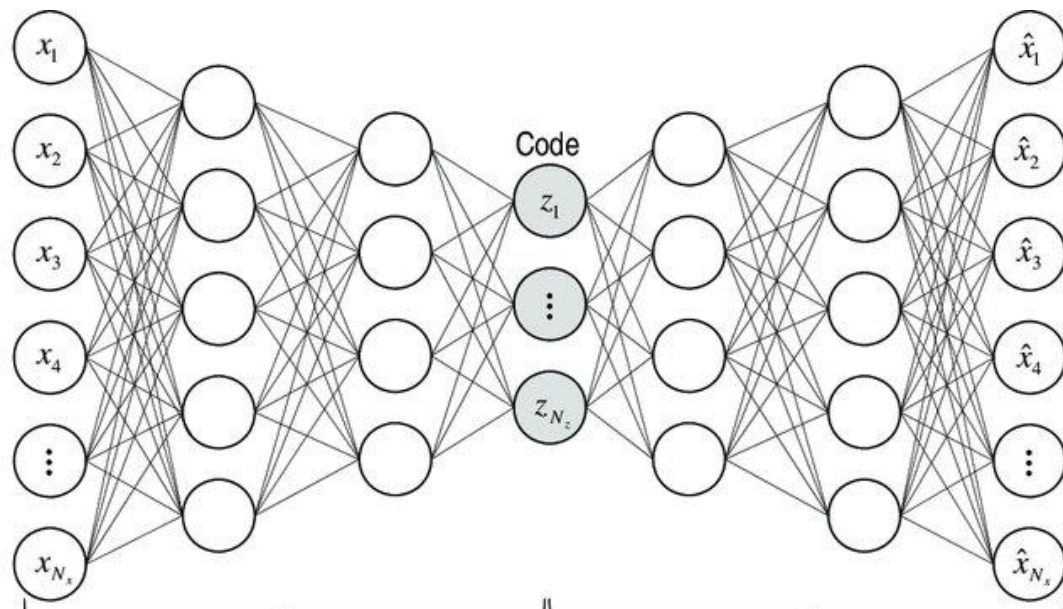


Регуляризированные автокодировщики

Заставлять модель выучивать важные и интересные признаки можно и без добавления регуляризационного члена.

В **шумоподавляющем автоэнкодере** (denoising autoencoder, DAE) мы минимизируем следующую функцию потерь:

$$\mathcal{L}(x, g(f(\tilde{x})))$$




В архитектуре можно использовать и линейные, и сверточные слои.

