DEEP LEARNING WITH KERAS WORKSHOP

MUHAMMAD RAJABINASAB @ TARBIAT MODARES UNIVERSITY

Chapter 6.5 :

Autoencoders

INTRODUCTION

- In this chapter, we will learn about autoencoders
- First we learn what autoencoders are
- The are many different types of autoencoders
- We will cover code examples for each type

AUTOENCODERS

- "Autoencoding" is a data compression algorithm where the compression and decompression functions are 1) data-specific, 2) lossy, and 3) learned automatically
- The compression and decompression functions are implemented with neural networks

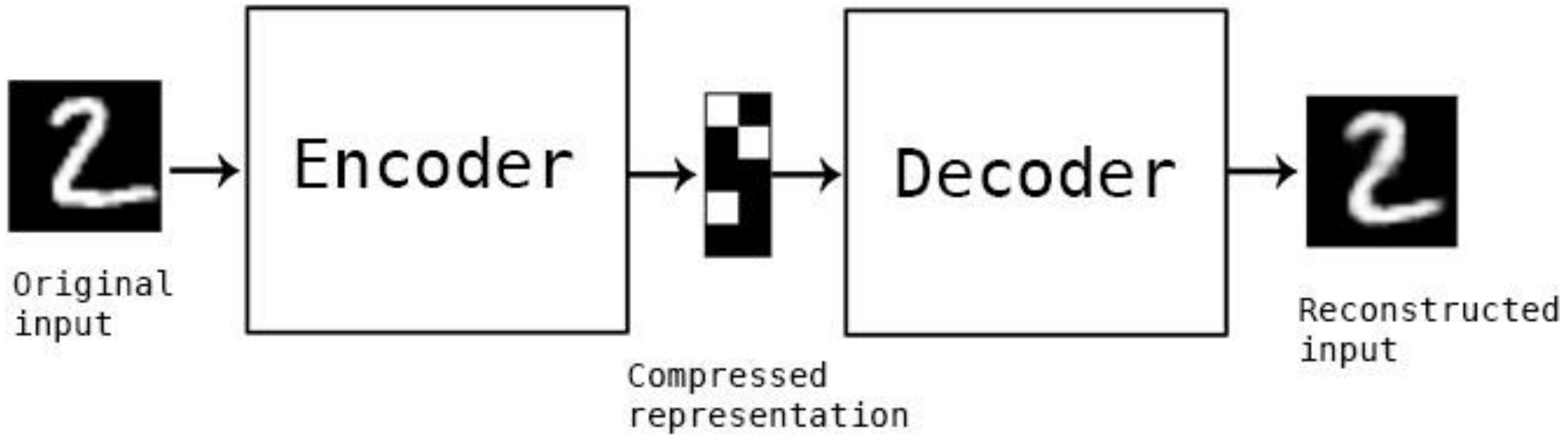
AUTOENCODERS

- Autoencoders are data-specific, which means that they will only be able to compress data similar to what they have been trained on
- Autoencoders are lossy, which means that the decompressed outputs will be degraded compared to the original inputs
- Autoencoders are learned automatically from data examples

AUTOENCODERS

- To build an autoencoder, you need three things
- An encoding function
- A decoding function
- A distance function between the amount of information loss between the compressed representation of your data and the decompressed representation

AUTOENCODERS



AUTOENCODERS

- Autoencoders are not good for data compression
- They are used for data denoising
- Dimensionality reduction
- Unsupervised learning problems

EXERCISE 6.501

- In this exercise we will build a single fully-connected neural layer as encoder and as decoder
- It is the simplest possible autoencoder
- Then we will add activity regularizer to our dense layer
- We call this regularizer sparsity constraint and the resulting autoencoder is the sparse autoencoder

EXERCISE 6.502

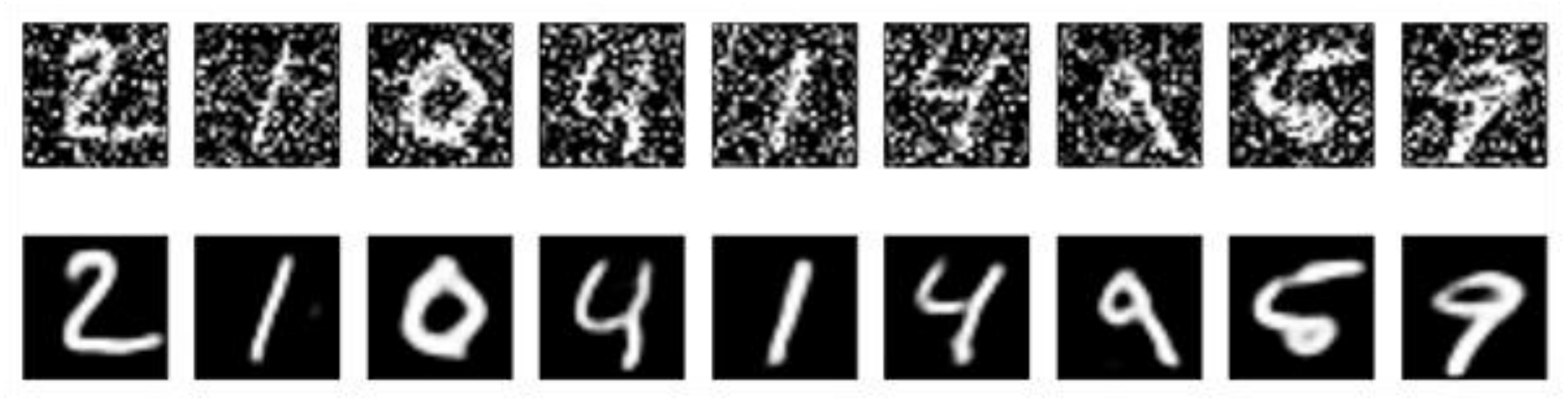
- We do not have to limit ourselves to a single layer as encoder or decoder
- We could instead use a stack of layers
- We call this autoencoder a deep autoencoder

OTHER TYPES OF AUTOENCODERS

- As we mentioned earlier, we have many different types of autoencoders such as convolutional autoencoders, denoising autoencoders and sequence-to-sequence autoencoders
- Building them is very similar to what we have learned so far, but we do not cover them here as we did not discuss convolutional and recurrent neural networks yet

OTHER TYPES OF AUTOENCODERS

- For example, a denoising autoencoder is able to denoise data very well:



VARIATIONAL AUTOENCODER

- As the final topic of this chapter, we will discuss variational autoencoders
- Variational autoencoders are a slightly more modern and interesting take on autoencoding
- It's a type of autoencoder with added constraints on the encoded representations being learned
- More precisely, it is an autoencoder that learns a latent variable model for its input data

VARIATIONAL AUTOENCODER

- So instead of letting your neural network learn an arbitrary function, you are learning the parameters of a probability distribution modeling your data
- If you sample points from this distribution, you can generate new input data samples
- But how does it work?

VARIATIONAL AUTOENCODER

- First, an encoder network turns the input samples x into two parameters in a latent space, which we will note z_mean and z_log_sigma
- Then, we randomly sample similar points z from the latent normal distribution that is assumed to generate the data, via z = z_mean + exp(z_log_sigma) * epsilon, where epsilon is a random normal tensor
- Finally, a decoder network maps these latent space points back to the original input data

VARIATIONAL AUTOENCODER

- The parameters of the model are trained via two loss functions: a reconstruction loss forcing the decoded samples to match the initial inputs and the KL divergence between the learned latent distribution and the prior distribution acting as a regularization term
- You could actually get rid of this latter term entirely, although it does help in learning well-formed latent spaces and reducing overfitting to the training data

KL DIVERGENCE

- The Kullback-Leibler divergence or the KL divergence is a measure of how one probability distribution is different from a second, reference probability distribution
- It is also called relative entropy
- The calculation of KL divergence will be different for discrete and continuous probability distributions

KL DIVERGENCE

- For discrete probability distributions:

$$D_{\mathrm{KL}}(P \parallel Q) = \sum_{x \in \mathcal{X}} P(x) \log\left(\frac{P(x)}{Q(x)}\right)$$

- For continuous probability distributions:

$$D_{\mathrm{KL}}(P \parallel Q) = \int_{-\infty}^{\infty} p(x) \log\left(\frac{p(x)}{q(x)}\right) dx$$

EXERCISE 6.503

- In this exercise we will build a variational autoencoder
- The procedure is a little different from previous exercises

SUMMARY

- In this chapter, we learned about autoencoders
- We saw that there are many different types of autoencoders
- We learned to build simple fully-connected autoencoders, deep autoencoders and variational autoencoders using keras
- Other types of autoencoders will be discussed in the next chapters

THANK YOU FOR YOUR ATTENTION!