Assignment 3.B: AutoEncoders

Wojtek Kowalczyk

05.05.2020

Introduction. The main purpose of this assignment is to gain some practical skills with implementing and using several variants of autoencoders that are discussed (and partially implemented) in the blog:

https://blog.keras.io/building-autoencoders-in-keras.html

Your key task is to complete (or re-implement) the code snippets provided in this blog and, after testing it on the MNIST data (the results should be similar to those reported in the blog), and applying it to the Fashion MNIST data. We estimate that completing this assignment shouldn't cost you more than 1-2 days of work.

The code snippets, (and the blog) were published in 2016, so they require some modifications to run on TF 2.x. There are two strategies: (recommended) either rewrite most of the code using the new syntax (as in the course textbook), or just stick to the old syntax, modifying it only where needed. In both cases you will learn how to "read" the old code examples and get them running with TF2.x.

In contrast to other assignments, you are not supposed to produce any report. Instead, you should submit just one jupyter notebook (or a single python program) that demonstrates your work. Your submission will not be graded! Instead, it will have to pass our test. If your notebook doesn't work, you will be required to fix it.

Details. Start with studying the first part of chapter 17 (pages 567-591) of the textbook. Then study the blog and complete/rewrite the program snippets that are included in this blog, so you can reproduce the results (including the images) of experiments that are described in the blog. In particular, implement:

1) A simple autoencoder (784:32:784).
2) A simple sparse autoencoder (also 784:32:784).
3) A simple deep autoencoder (784:128:64:32:64:128:784).
4) A convolutional autoencoder (encoder: 3 convolutional layers; each followed by MaxPooling; decoder: symmetric deconvolutions; each followed by UpSampling). Skip the part that demonstrates the use of tensorboard.
5) An application of convolutional autoencoder to image denoising.
6) A variational autoencoder and its application for 2-d visualization of 10 classes of images and VAE as a generative model for 10 classes of images (the last 2 images from the blog).
7) Once you are finished with the MNIST data, rerun your notebook on the Fashion MNIST.

Hint: In all cases replace the optimizer 'adadelta' by 'adam' – then the results should become more similar to what is reported in the blog.

Deadline: 24 May, 23:59.