

Project 1: Deep Neural Networks and Auto Encoders

DUE DATE: 02/05/2018

For this project, you are free to use any language or any infrastructure that you want. However, I strongly suggest using iPython, Keras with TensorFlow.

Problem 1 [40 points]

- Download the MNIST dataset. It's available in many places, including [Kaggle](#)!
- Create a modified training set
 - For each image in the train set, generate two images where the probability of a pixel value being flipped is 0.03
 - Your new training set size should be 120K
- Review the blog on keras and autoencoders, <https://blog.keras.io/building-autoencoders-in-keras.html>
- Implement a convolutional autoencoder (four times)
 - Vary the CNN architecture in four ways
 - Optimize as best as possible during training

Plot the test set error (mean L2 error) as a function of the number of samples used to train each autoencoder

Plot the best- and worst set of digits as measured by error (original image, reconstructed image)

Problem 2 [60 points]

Download the imagenet collection from <http://image-net.org/>. The imagenet collection is very big for the scope of the class. We will pick few categories so that we can keep the computations tractable. As you can see <http://image-net.org/explore> the categories are not balanced. For the scope of this project, we will use the images from:

- [Plant, flora, plant life \(1271\)](#)
 - [Geological formation \(1808\)](#)
 - [Fungus \(1207\)](#)
 - [Sport \(1888\)](#)
 - [Person \(1242\)](#)
 - [Animal \(1571\)](#)
- Implement a convolutional autoencoder (four times)
 - Vary the CNN architecture in four ways
 - Optimize as best as possible during training

Plot the test set error (mean L2 error) as a function of the number of samples used to train each autoencoder

Plot the best- and worst set of digits as measured by error (original image, reconstructed image)

Deliverables. Present a detailed report on your experiment that explains your design choices and results, shows examples of outputs and inputs, etc. Are the results reasonable? How did you test to assure so? Reports should include your code, preferably in python or ipython notebooks.