Lab 10
Neural Networks and Deep Learning
Advanced Machine Learning
DATA 442/642

Exercise 1

This exercise focuses on the original GAN, which is trained on the MNIST database to learn to generate artificial hand-written characters. To implement and train the network, use the TensorFlow framework. You are encouraged to use the Keras high-level API for simplicity.

(a) Load the data set of 60000 to be used for training. Normalize image values so that all values are in the $[-1, 1]$ interval. In Keras, you may use the functions provided by the tensorflow.keras.datasets.mnist module.

(b) The generator is fed to its input with a noise vector and outputs an image. The dimension of the input noise vector is 100, and its elements are i.i.d. sampled from a normal distribution of zero mean and unit variance. The generator consists of three fully connected layers with 256, 512, and 1024 neurons, respectively. The activation function used for the neurons in the hidden layers is the leaky ReLU with parameter $\alpha = 0.2$. The output layer comprises 784 nodes and the tanh is employed as the respective activation function.

(c) The discriminator takes as input a 1×784 vector, corresponding to the vectorized form of the 28×28 MNIST images. The discriminator consists of three fully connected layers with 1024, 512, and 256 neurons, respectively. The leaky ReLU activation function is also employed, with $\alpha = 0.2$. During training, use the dropout regularization method, with probability of discarding nodes equal to 0.3. The output layer consist of a single node with a sigmoid activation function.

(d) To train the implemented network, use the two-class (binary) cross-entropy loss function. Adopt the Adam minimizer with step size (learning rate) equal to 2×10^{-3}, $\beta_1 = 0.5$, and $\beta_2 = 0.999$ as parameters for the optimizer. The recommended batch size is 100. Train the network for 400 epochs as follows. For each training loop, (a) generate a random set of input noise and images, (b) generate fake images via the generator, (c) train only the discriminator, and (d) then train only the generator, according to Algorithm 18.5. Play with the number of iterations, associated with the discriminator training.

(e) During training and every 20 epochs, visualize the generated images created by the generator and comment on the evolution of the learning process.

(f) Play with all the various parameters that have been suggested above and see the effect on the training.