CSE455/555 - Intro to Pattern Recognition
Problem Set 5: Generative Models

Due Date: Friday, April 23, 2021 11:59PM

A trend in deep learning is developing generative machine learning models that can make the predictions explainable and accountable. In this problem set, we will train several classic generative models — restricted Boltzmann machine, variational auto-encoder and generative adversarial networks — with Tensorflow/Keras to generate MNIST images. There are many tutorials in public domain on exactly MNIST + Tensorflow/Keras + RBM/VAE/GAN, so (a) start to study those tutorials right away, and (b) derive the algorithms + write the code by yourself instead of copying from the Internet (UBLearns has a mechanism to check for plagiarism, so do not even think of copying something from somewhere without being noticed).

1 Task-1

For this task, you are free to choose whether you want to use binary nodes or floating point nodes, but the derivation has to match the implementation. You are free to choose any parameters that are not specified here.
(a) Derive the restricted Boltzmann machine algorithm that you will implement, and explain your derivation.
(b) Implement the training and inference algorithms for RBM. Train RBMs with 20, 100 and 500 hidden nodes to generate MNIST images using the training data set.
(c) Generate MNIST images from the ones in the testing data set that have 20%, 50% and 80% pixels missing/removed. It means you are going to train the RBM with those degraded images.

2 Task-2

For this task, you are free to choose any parameters that are not specified here.
(a) Derive the variational autoencoder algorithm that you will implement, and explain your derivation.
(b) Implement the training and inference algorithms for VAE. Train VAE with 2, 8 and 16 code units to encode MNIST images using the training data set. Code units are the nodes in the bottleneck layer in the middle. Check this page for more information: https://towardsdatascience.com/introduction-to-autoencoders-7a47cf4ef14b. The neural network will be 784 input → 256 hidden → 2/8/16 code → 256 hidden → 784 output.
(c) Then use the 2 code → 256 hidden → 784 output part of the trained network to generate

images by varying each of the 2 code units from -3 to 3. It means you are going to choose two random values in the range of [-3,3] and pass those two values to the 2 code units. Try three different pair of values and output the results. You do not have to find the best result.

3 Task-3[Optional]

Implement VAE to generate MNIST images, where you use convolutional neural network from encoder and deconvolutional neural network for decoder. This one should be super-easy for you to catch up with the total score because there are many tutorials on the Internet and you have already solved task-2.

4 Task-4[Optional]

Derive and explain the GAN algorithm. Implement GAN and train it from MNIST training data set to generate digits. Show images generated from GAN.

5 Submission

Submit your solutions as a single ipynb file through UBlearn. You can use Google Colab:
https://colab.research.google.com/notebooks/intro.ipynb
https://towardsdatascience.com/getting-started-with-google-colab-f2fff97f594c.
The ipynb file should include your code, execution results, any explanations and answers to the questions. Use text cells to answer questions and add explanations.
Markdown guide for text cells:
https://colab.research.google.com/notebooks/markdown_guide.ipynb#scrollTo=Lhfnlq1Surtk
https://colab.research.google.com/notebooks/basic_features_overview.ipynb#scrollTo=4hfV37gxpP_c
You can also add math to text cells using LaTeX. Just place the statement within a pair of $ signs. Please typeset your mathematics. Do not upload pictures of handwriting math formulas. Math typesetting help: https://www.codecogs.com/latex/eqneditor.php

6 Libraries

Allowed:
Tensorflow/Keras/Pytorch are allowed.
Basic libraries are allowed, such as gzip, pickle, math, numpy, scipy, matplotlib, etc.

Not allowed:
Do not use any Python libraries/toolboxes, built-in functions, or external tools/libraries that directly perform training and generating.
Do not use sklearn/scikit-learn.

7 Rubric

Total: 10 points + 2 bonus points

Task-1:
5 points: (a) 1 point, (b) 2 points, (c) 2 points.
Task-2:
5 points: (a) 1 point, (b) 2 points, (c) 2 points.
Task-3:
1 bonus point.
Task-4:
1 bonus point.

8 Late Penalty

5% late penalty if you submit it within three days after the deadline (before April 26, 11:59 pm). 10% late penalty for submissions after three days but within a week after the deadline (before April 30, 11: 59 pm). Submissions more than a week late will not be graded.

9 Acknowledgement

By submitting this paper, you agree: (1) that you are submitting your paper to be used and stored as part of the SafeAssign™ services in accordance with the Blackboard Privacy Policy; (2) that your institution may use your paper in accordance with your institution's policies; (3) that your use of SafeAssign will be without recourse against Blackboard Inc. and its affiliates.

10 Academic Integrity

Academic integrity is a fundamental university value. Any violation will be reported to the University and will result in penalties in grades.
Do not share your answers with other students. This is an individual assignment. You are not allowed to work in groups. Working in groups and submitting similar answers is considered a violation of academic integrity.
Do not plagiarise someone else's words, ideas, or data you find online. Always cite your sources.