Supplementary Material: Unsupervised Open Set Recognition using Adversarial Autoencoders

Mehadi Hassen
School of Computing
Florida Institute of Technology
Melbourne, FL 32901
mhassen2005@my.fit.edu

Philip K. Chan
School of Computing
Florida Institute of Technology
Melbourne, FL 32901
pkc@cs.fit.edu

October 8, 2020

A Network Architectures

Microsoft Challenge Dataset For the Microsoft Malware Challenge Dataset, we used two non-linear layers of 250 hidden units with ReLU activation function for the encoder, decoder, gaussian discriminator, and categorical discriminator networks. For the output layer of the decoder, we used Sigmoid activation function. We used 30 linear units for the z_ii-layer and z-layer. The dimensionality of the y-layer was set to 30 (i.e., 30 clusters). Batch normalization was used in all layers and dropout was not used in any layer. We used Adam Optimizer for the reconstruction loss, adversarial losses, and ii-loss. The learning rate of 0.001 and beta1 of 0.9 was used for the first 1000 training iterations of the first stage after which we change the learning rate to 0.0001. We ran the first training stage for 3000 iterations and the second stage for 2000 iterations. We used a batch size of 512.

For the evaluation of the kmeans-based approach we used the same number of clusters (i.e., 30 clusters). For all the evaluated approaches we used a contamination ratio of 0.05 for the outlier threshold estimation.

Android Dataset For the Android Genome Project Dataset, we used a similar architecture as the Microsoft dataset with the following differences. We used a single hidden layer with 100 units and a ReLU activation function for the encoder, decoder, gaussian discriminator, and categorical discriminator networks. We used 14 clusters and 12 linear units for the z_ii-layer and z-layer. We used learning rate of 0.001 for the first 3000 training iterations of the first stage after which we change the learning rate to 0.0001. The beta1 of the Adam Optimizer for reconstruction loss was fixed at 0.9 while it was fixed at 0.1 for the other loss functions. We ran the first training stage for 3000 iterations and the second stage for 1000 more iterations. We used a batch size of 256.

For the evaluation of the kmeans-based approach we used the same number of clusters (i.e., 10 clusters). For all the evaluated approaches we used a contamination ratio of 0.01 for the outlier threshold estimation.

MNIST For the MNIST dataset, we used a convolutional network in the encoder. We first pad the input images of size (28,28) to get an input layer size (32,32) with 1 channel. Following the input are two convolutional layers with 32 and 64 units, (4,4) kernel size with a stride of (1,1) and SAME padding. We use max-polling of (3,3) kernel, (2,2) stride size, and SAME padding after each convolutional layer. The decoder network uses the same architecture with deconvolutional layers of 64 and 32 units. A linear layer was used for the output layer of the decoder. For the Gaussian and Categorical discriminator networks, we used two hidden layers with 1000 units. We use ReLU activation function for all non-linear layer. The dimensionality of the y-layer was set to 6 (i.e., 6 clusters) and 15 linear units were used in the z_ii-layer and z-layer. Batch normalization was used in all layers and dropout was not used.

Adam Optimizer with a learning rate of 0.001 was used for the first 3000 training iterations of the first stage after which the learning rate is changed to 0.0001. The beta1 of the Adam Optimizer for the reconstruction loss was fixed at 0.9 while it was fixed at 0.1 for the other losses. We ran the first training stage for 3000 iterations and the second stage for an additional 7000 iterations. We used a batch size of 512.