Appendix: Revisiting Deep Archetypal Analysis for Phenotype Discovery in High Content Imaging

Mario Wieser, Daniel Siegismund, Stephan Steigele
Genedata AG
{mario.wieser, daniel.siegismund, stephan.steigele}@genedata.com

1. Architecture and Training Details

MNIST. For our model, we define an encoder network with three fully-connected hidden layers with 784, 512 and 256 dimensions, respectively. Subsequently, we set our latent dimensional representation to be two-dimensional. Similar to the encoder network, we define the decoder to be a fully-connected network with 256, 512 and 784 nodes and employ a mean squared error loss for reconstruction. We train our model in 250 epochs where the first 10 epochs are used as burn-in phase similar to Markov Chain Monte Carlo methods before the start the alternating optimization procedure. In addition, we use a batch size of 256 and the Adam optimizer with an initial learning rate of 0.001 and a reduce on plateau learning rate scheduler for optimizing L_2 loss function and an Adam optimizer to minimize the L_1 loss function with learning rate 5×10^{-4}. We set the number of archetypes to three due to prior knowledge and use ReLU activation functions.

COOS. To perform this experiment, we propose an encoder network with two convolutional layers having 16 and 32 feature maps with kernel size 3, stride 2 and padding 1 and batch normalization. This is followed by a Flatten layer and two fully-connected layers with 256 and 128 dimensions, respectively. The latent dimension of our layer is set to 51 dimensions which we found sufficient to reconstruct the HCI images reasonably well. For the decoder network, we define fully-connected network with 512 and 2048 nodes followed by two deconvolutional layers 16 and 1 feature maps using a kernel size of 3, stride 2 padding 1 and output padding 1 as well as batch normalization. For the archetype part of our model, we use a fully-connected network with 64 and 3 nodes as well as a weight matrix for the archetypes with dimensions 51 and 3 that is optimized as a free parameter. For all layers, we assume ReLu activation functions and employ a mean squared error loss for reconstruction. We train our model in 3000 epochs where the first 1000 epochs are used as burn-in phase similar before the start the alternating optimization procedure. In addition, we use a batch size of 3096 and the Adam optimizer with an initial learning rate of 0.001 and a reduce on plateau learning rate scheduler for optimizing L_2 loss function and an Adam optimizer to minimize the L_1 loss function with learning rate 5×10^{-4}. We set the number of archetypes to three due to prior knowledge.

Neurite Assay. To perform this experiment, we propose an encoder network with two convolutional layers having 32 and 64 feature maps with kernel size 3, stride 2 and padding 1. This is followed by a Flatten layer and two fully-connected layers with 512 and 100 dimensions, respectively. The latent dimension of our layer is set to 9 dimensions which we found sufficient to reconstruct the HCI images reasonably well. For the decoder network, we define fully-connected network with 512 and 50176 nodes followed by two deconvolutional layers 32 and 1 feature maps using a kernel size of 3, stride 2 padding 1 and output padding 1. For the archetype part of our model, we use a fully-connected network with 32 and 3 nodes as well as a weight matrix for the archetypes with dimensions 9 and 3 that is optimized as a free parameter. For all layers, we assume ReLu activation functions and employ a mean squared error loss for reconstruction. We train our model in 3000 epochs where the first 1000 epochs are used as burn-in phase similar before the start the alternating optimization procedure. In addition, we use a batch size of 128, the Adam optimizer with an initial learning rate of 0.001 and a reduce on plateau learning rate scheduler for optimizing L_2 loss function and an Adam optimizer to minimize the L_1 loss function with learning rate 5×10^{-4}. We set the number of archetypes to three due to prior knowledge. In order to assess the quality of the trained model, we split the dataset in a 80/20 percent training/test split.