

## A SUPPLEMENTARY

### A.1 IMPLEMENTATION DETAILS

The implementation details of the neural architecture used in the experimentation are described here. We design the architecture to facilitate training for the semi-supervised and unsupervised embedding with the neural architecture of the same representational capability. The implementation has three distinct components, (1) encoder, (2) classifier, and (3) decoder. The encoder architecture takes the data input and maps to a reduced dimension. The classification framework performs classification tasks on the embedding produced by the encoder framework. The decoder network produces the approximate inverse map of the encoder network. Training of the unsupervised embedding uses encoder and decoder components only, while semi-supervised embedding uses all three components. In semi-supervised scenario gradient of classification and decoder layer is combined linearly at the embedding layer, and which is used to train the weights of the encoder layer. The architecture and the details of all the ML models for different data-sets are described below.

1. Input to Embedding network (I2E)
  - Input layer
  - Multiple layers of
    - Dense layer, ReLU
    - Batch normalization
  - Dense layer, ReLU (Embedding layer)
2. Embedding to Classifier network (E2C)
  - Input - Embedding layer
  - Multiple layers of
    - Dense layer, ReLU
    - Batch normalization
  - Dense layer, Softmax (Classification Output)
3. Embedding to Decoder network (E2D)
  - Input - Embedding layer
  - Multiple layers of
    - Dense layer, ReLU
    - Batch normalization
  - Dense layer, ReLU (Decoder Output)

For the unsupervised auto-encoder model we train the input to embedding network + embedding to decoder network with mean square error as the loss function. For semi-supervised auto-encoder we train the full model with a linear combination of categorical cross-entropy and mean square error as the loss function.

For categorical variable we use embedding layer. The output of the embedding layer is concatenated and then used for further processing in the encoder layer. The decoder layer similarly contains an extra layer to decode the reconstructed embedding. During the training process the embedding and decoding layer for the categorical variables are trained jointly.

#### A.1.1 TRAINING DETAILS

We have performed extensive search over the various neural layer combination to derive the models with best out sample performance. The final architecture for different data-sets are mentioned in Table 4. Data-sets are scaled with min-max scalar and divided into 80%-20% training-testing splits. Models are then trained using Adam optimizer with learning rate 0.01. The models are run for various epoch length till convergence.

<b>Data-set</b>	<b>I2E layers</b>	<b>E2C layers</b>	<b>E2D layers</b>	<b>epochs</b>
German Credit	[16, 8, 4]	[2]	[8, 16, 20]	300
COMPAS	[8, 4, 4]	[2]	[4, 8, 10]	200
Adult Income	[24, 12, 6]	[2]	[12, 24, 12]	150
PIMA	[4, 4, 2]	[2]	[4, 4, 8]	60
Cancer	[24, 16, 8]	[2]	[16, 24, 30]	50

Table 4: Training details