Kunal Kwatra

SUM UP

TITLE: INTRUSION DETECTION USING CVAE

My contribution in the project is implementation of Conditional Variational Autoencoder using TensorFlow and the processing of data both in its dimension and type to be fed to the model for training.

The implementation of model is divided into two parts:

1. **Encoder**
2. **Decoder**

The Encoder takes the features of the training data as input, whereas the Decoder takes the output of Encoder along with the 5 labels associated with training data as input.

For Encoder the out is the latent variables (dimension = (25,)), and in case of Decoder the output are the reconstructed features of the training dataset.

For the structure of Encoder Layers are defined as:

1. **Input -> Taking 116 features of training data as input.**
2. **he1 -> Takes the input layer and provides an output set of 500 neurons with the 'relu' activation function.**
3. **he2 -> Takes the the output of he1 and provides an output set of 500 neurons with the 'relu' activation function.**
4. **Z_mean -> Takes he2 and produce a set of dimension (25,) for sampling.**
5. **Z_log_sigma - > Takes he2 and produce a set of dimension (25,) for sampling.**

After encoder, sampling is performed to get latent variables taking z_mean and z_log_sigma into consideration.

For the structure of Decoder:

1. **Label_input -> Takes input as the comination of the output attained after sampling along with the 5 unique class labels.**
2. **hd1 -> Takes output of sampling as input .**
3. **hd2-> Takes hd1 along with label_input and produce the output with relu function.**
4. **Output -> Provides the output of cvae model.**

After defining, loss is calculated as combination of KL_Loss and reconstruction loss.

The implemented model is fully functional and works without any ladder.

Summary of CVA models

dense (Dense)	(None, 500)	58500	['input_1[0][0]']
dense_1 (Dense)	(None, 500)	250500	['dense[0][0]']
dense_2 (Dense)	(None, 25)	12525	['dense_1[0][0]']
dense_3 (Dense)	(None, 25)	12525	['dense_1[0][0]']
lambda (Lambda)	(None, 25)	0	['dense_2[0][0]', 'dense_3[0][0]']
dense_4 (Dense)	(None, 495)	12870	['lambda[0][0]']
label_5 (InputLayer)	[(None, 5)]	0	[]
concatenate (Concatenate)	(None, 500)	0	['dense_4[0][0]', 'label_5[0][0]']
dense_5 (Dense)	(None, 500)	250500	['concatenate[0][0]']
dense_6 (Dense)	(None, 116)	58116	['dense_5[0][0]']