Deep Learning inlämningsuppgift #1

Publishing: 15th April 2024

Deadline 4th Maj 2024 KL 23:59

G: 75+ , VG : 90+

About project:

As we noticed in deep learning, we need to build our own network semi-manually and now it is your turn to make a difference and automate it to the other Python AI developer. So, we would like to build our own class call MyANN that will help developer to get the model and some more info as next description:

1- **Main class MyANN**:
   - MyAnn is a class to build multilayer Perceptrons classifier or regressor by reading in and validation of parameters.

2- **MyAnn Class Parameters descriptions**: **(60 Points)**

parameters_name	Descriptions	Points
data_set **Type = String**	**Valid absolute path to .csv dataset file,** Data cleaned and ready to process in csv dataset form.	**4**
target **Type = string**	**Valid** name to the Labels' column.	**4**
hidden_layer_sizes. **type = Tuple**	A Tuple like to describe the shape of hidden layers, default= (100,) The number in each tuple field represents the number of neurons in the related hidden layer. In case **it was negative, so it is dropout layer** and the value represent the percent. Example: hidden_layer_sizes = (50,25, -0.25,12) so the model has 4 hidden layers,	**8**

	with 50 neurons in the first hidden layer, and 25 in the second and drop out with 25% and fourth has 12.	
activation **Type = string**	**Valid** Activation function for the hidden layer default is ‘relu’ other options ‘sigmoid’, ‘softmax’, ‘tanh’.	**4**
loss **Type = string**	**Valid** Loss function, values = ‘mse’, 'binary_crossentropy' or 'categorical_crossentropy'	**4**
optimizer **Type = string**	**Valid** Optimizer for the output layer, the default is ‘adam’ other options for this class is ‘rmsprop’ and ‘sgd’	**4**
batch_size **type = int**	Integer or None. Number of samples per gradient update. If unspecified, batch_size will default to 32	**4**
epochs **Type = int**	Number of epochs to train the model, default is 1 epoch	**4**
monitor **Type = string**	Result to be monitored, default = ‘val_loss’, other option ‘accuracy’	**4**
patience **Type = int**	Number of epochs with no improvement after which training will be stopped, default = as value in epochs.	**4**
mode **Type = string**	One of {"auto", "min", "max"}. In min mode, training will stop when the quantity monitored has stopped decreasing; in "max" mode it will stop when the quantity monitored has stopped increasing; in "auto" mode, the direction is automatically inferred from the name of the monitored quantity.	**4**

verbose	0, 1, or 2. Verbosity mode. 0 = silent, 1 = progress bar, 2 = one line per epoch	**4**
use_multiprocessing: Boolean	If True, use process-based threading. If unspecified, use_multiprocessing will default as False	**8**

3- Supporting Methods: **(25 Points)**

Methods name	Descriptions	Points
model_loss	Method to return Pandas DataFrame for all loss, val_loss , accuracy and val_accuracy values.	**2**
plot_model_loss	Method Show the plotting of model's loss and val_loss against number of epochs	**2**
model_predict	Take in a row of data set and return a predict. Ps. Check needed processing for the input to get the right prediction	**2**
save_original keras model	Save Keras model (NOT MyANN model) in .h5 format	**2**
Load_ original keras model	load saved Keras model (NOT ANN model) from .h5 format	**2**
For Classfication cases Only	The next 2 methods you should allow for classification case , Otherwise print needed info that not applicable by regressions	
plot_model_accuracy	Method Show the plotting of model's accuracy and val_accuracy	**2**

print_classfication_report	Print to terminal classification report that shows: precision, recall, f1-score per each class and global accuracy.	**2**
For regression cases only	The next 3 methods you should allow for classification case , Otherwise print needed info that not applicable by classifiers	
plot_ predictions_scatter	Show scatter plot of real values in target columns against predict value.	**2**
rmse	Return sqrt of the mean squared error	**2**
mae	Return mean absolute error	**2**
Plot_residual_error	Shows distplot for residual error. Where Residual errors are the difference between real label and predict label values.	**5**

4- MyAnn Class attributes that we can get from object(Not protected) of myANN class: **(15 Points)**

attributes	descriptions	points
classes_ **type : list of strings**	Class label(s)	**2**
loss_ **type : float**	The last loss computed with the loss function of the model	**2**
best_loss **type : float**	The minimum loss reached by the model throughout fitting	**3**
features_	List of features columns name	**2**

type : list of strings		
n_layers **type : int**	Number of layers in this model	**2**
n_outputs_ **type : int**	Number of outputs	**2**
out_activation_ **type string**	Name of the output activation function.	**2**

Good Luck!