- Step1: Data Preparation: Get the data from csv file to dataframe

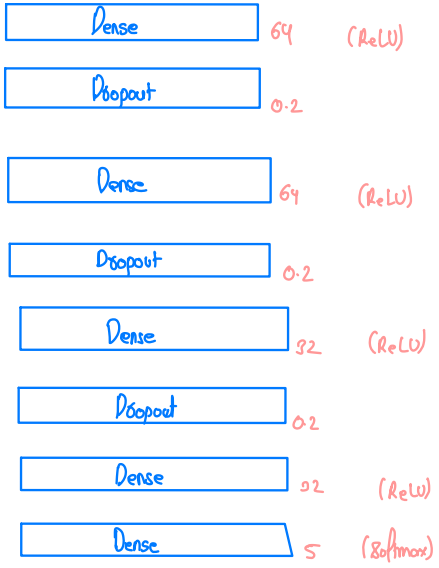
  Train data: (87553, 188)
  test data: (21891, 188)

- Step2: Our data is not in categorical form so we first have to convert it to categorical form

```
from keras.utils import to_categorical
y_train = to_categorical(y_train)
y_test = to_categorical(y_test)
```

(A) FOR DNN

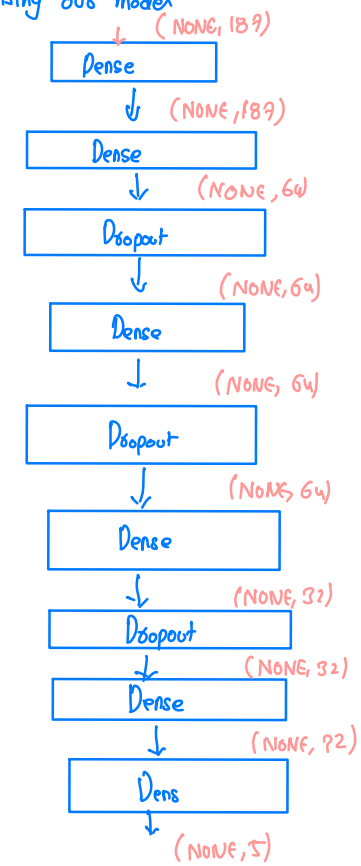
- Step3: Lets create our model : Sequential API



- Step4: To prevent from overfitting & for better Coverage of loss function we will use EarlyStopping & ReduceLROnPlateau

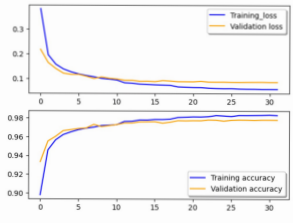
```
es = EarlyStopping(monitor='val_loss', mode='min', verbose=1, patience=5)
learning_rate_reduction = ReduceLROnPlateau(monitor='val_accuracy', patience=3, verbose=1,
                                            factor=0.5, min_lr=0.000
```

- Step5: Visualising our model



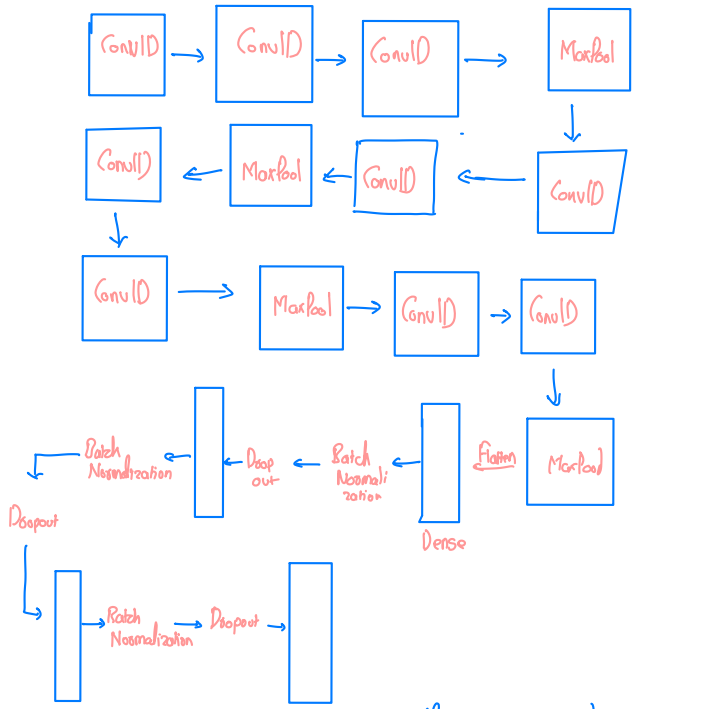
- Step 6: fit
  - batch_size = 128
  - epochs = 200
  - verbose = 1
  - validation_data = (x_test, y_test)
  - Callbacks
    - ReduceLROnPlateau
    - Early Stopping

- Step7: We can plot the output using history['loss'], history['val_loss']
  history['accuracy'], history['val_accuracy']



⑰ for CNN

- lets visualise our model



- Compile with → optimizer = Adam (learning_rate = 0.001)
  → loss = "categorical_crossentropy"
  → metrics = ["accuracy"]

- fit the model with → epochs = 100
  → verbose = 1
  → batch_size = 128
  → validation_data = (x_test, y_test)
  → callbacks → learning_rate Reduction Callback
  → Early Stopping

8