1)

model\_simple = tf.keras.Sequential([

tf.keras.layers.Dense(128, activation='relu', input\_shape=(input\_size,)),

tf.keras.layers.Dense(64, activation='relu'),

tf.keras.layers.Dense(32, activation='relu'),

tf.keras.layers.Dense(16, activation='relu'),

tf.keras.layers.Dense(8, activation='relu'),

tf.keras.layers.Dense(1, activation='sigmoid')

])

2)

import tensorflow as tf

model\_dropout = tf.keras.Sequential([

tf.keras.layers.Dense(128, activation='relu', input\_shape=(input\_size,)),

tf.keras.layers.Dropout(0.5),

tf.keras.layers.Dense(64, activation='relu'),

tf.keras.layers.Dropout(0.5),

tf.keras.layers.Dense(32, activation='relu'),

tf.keras.layers.Dropout(0.5),

tf.keras.layers.Dense(16, activation='relu'),

tf.keras.layers.Dropout(0.5),

tf.keras.layers.Dense(8, activation='relu'),

tf.keras.layers.Dropout(0.5),

tf.keras.layers.Dense(1, activation='sigmoid')

])

3)

import tensorflow as tf

model\_batch\_norm = tf.keras.Sequential([

tf.keras.layers.Dense(128, activation='relu', input\_shape=(input\_size,)),

tf.keras.layers.BatchNormalization(),

tf.keras.layers.Dense(64, activation='relu'),

tf.keras.layers.BatchNormalization(),

tf.keras.layers.Dense(32, activation='relu'),

tf.keras.layers.BatchNormalization(),

tf.keras.layers.Dense(16, activation='relu'),

tf.keras.layers.BatchNormalization(),

tf.keras.layers.Dense(8, activation='relu'),

tf.keras.layers.BatchNormalization(),

tf.keras.layers.Dense(1, activation='sigmoid')

])

4)

import tensorflow as tf

model\_activation = tf.keras.Sequential([

tf.keras.layers.Dense(128, activation='relu', input\_shape=(input\_size,)),

tf.keras.layers.Dense(64, activation='tanh'),

tf.keras.layers.Dense(32, activation='sigmoid'),

tf.keras.layers.Dense(16, activation='elu'),

tf.keras.layers.Dense(8, activation='selu'),

tf.keras.layers.Dense(1, activation='sigmoid')

])

5)

import tensorflow as tf

model\_varied\_neurons = tf.keras.Sequential([

tf.keras.layers.Dense(256, activation='relu', input\_shape=(input\_size,)),

tf.keras.layers.Dense(128, activation='relu'),

tf.keras.layers.Dense(64, activation='relu'),

tf.keras.layers.Dense(32, activation='relu'),

tf.keras.layers.Dense(16, activation='relu'),

tf.keras.layers.Dense(8, activation='relu'),

tf.keras.layers.Dense(1, activation='sigmoid')

])