tf.convert\_to\_tensor ---- conert array to tensor ds

tf.keras.activations.softmax(inputs) -- convert to softmax

tf.reduce\_sum(outputs) -- sum of all he values

np.argmax(outputs) -- return index with maximum values

-------------------

model = keras.Sequential([ **---- seq of layers**

  keras.layers.Flatten(input\_shape=(28,28)), **-- covert to 1d**

  keras.layers.Dense(128, activation=tf.nn.relu), **-- layers of neuron**

  keras.layers.Dense(10, activation=tf.nn.softmax),

])

model.summary()

model.compile(

optimizer = tf.optimizers.Adam(), **reducing loss and improve accuracy**

loss='sparse\_categorical\_crossentropy',

metrics=['accuracy']) - **judge the performance of your model**

model.fit(training\_images, training\_labels, epochs=5

)