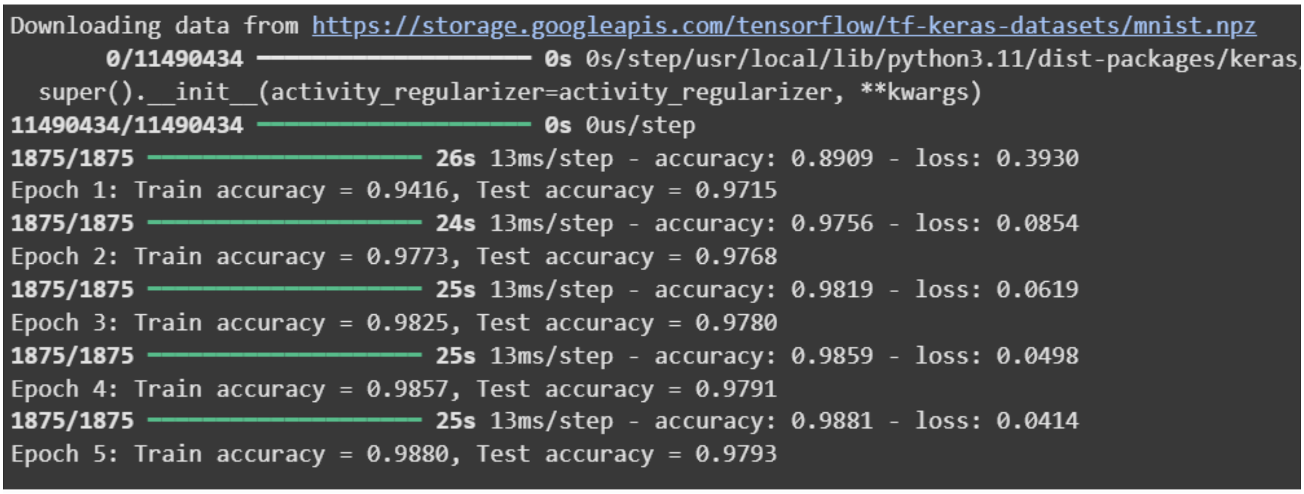
```
import tensorflow as tf
from mpi4py import MPI
import numpy as np

comm = MPI.COMM_WORLD
rank = comm.Get_rank()
size = comm.Get_size()

tf.random.set_seed(42)
np.random.seed(42)

model = tf.keras.models.Sequential([
    tf.keras.layers.Conv2D(32, (3, 3), activation='relu', input_shape=(28, 28, 1)),
    tf.keras.layers.MaxPooling2D((2, 2)),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(10, activation='softmax')
])

mnist = tf.keras.datasets.mnist
(x_train, y_train), (x_test, y_test) = mnist.load_data()
x_train, x_test = x_train / 255.0, x_test / 255.0
x_train = x_train.reshape(-1, 28, 28, 1)
x_test = x_test.reshape(-1, 28, 28, 1)

if rank == 0:
    weights = model.get_weights()
else:
    weights = None
weights = comm.bcast(weights, root=0)
model.set_weights(weights)
```

```
def train(model, x_train, y_train, rank, size):
    n = len(x_train)
    chunk_size = n // size
    start = rank * chunk_size
    end = (rank + 1) * chunk_size if rank < size - 1 else n
    x_train_chunk = x_train[start:end]
    y_train_chunk = y_train[start:end]
    model.compile(optimizer='adam', loss='sparse_categorical_crossentropy',
metrics=['accuracy'])
    model.fit(x_train_chunk, y_train_chunk, epochs=1, batch_size=32, verbose=0)
    train_loss, train_acc = model.evaluate(x_train_chunk, y_train_chunk, verbose=0)
    train_acc = comm.allreduce(train_acc, op=MPI.SUM)
    return train_acc / size
epochs = 5
for epoch in range(epochs):
    train_acc = train(model, x_train, y_train, rank, size)
    test_loss, test_acc = model.evaluate(x_test, y_test, verbose=0)
    test_acc = comm.allreduce(test_acc, op=MPI.SUM)
    if rank == 0:
        print(f"Epoch {epoch + 1}: Train accuracy = {train_acc:.4f}, Test accuracy = {test_acc /
size:.4f}")
comm.Barrier()
```



```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz
       0/11490434 ━━━━━━━━━━━━━━━━━━━━ 0s 0s/step/usr/local/lib/python3.11/dist-packages/keras
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
11490434/11490434 ━━━━━━━━━━━━━━━━━━━━ 0s 0us/step
1875/1875 ━━━━━━━━━━━━━━━━━━━━ 26s 13ms/step - accuracy: 0.8909 - loss: 0.3930
Epoch 1: Train accuracy = 0.9416, Test accuracy = 0.9715
1875/1875 ━━━━━━━━━━━━━━━━━━━━ 24s 13ms/step - accuracy: 0.9756 - loss: 0.0854
Epoch 2: Train accuracy = 0.9773, Test accuracy = 0.9768
1875/1875 ━━━━━━━━━━━━━━━━━━━━ 25s 13ms/step - accuracy: 0.9819 - loss: 0.0619
Epoch 3: Train accuracy = 0.9825, Test accuracy = 0.9780
1875/1875 ━━━━━━━━━━━━━━━━━━━━ 25s 13ms/step - accuracy: 0.9859 - loss: 0.0498
Epoch 4: Train accuracy = 0.9857, Test accuracy = 0.9791
1875/1875 ━━━━━━━━━━━━━━━━━━━━ 25s 13ms/step - accuracy: 0.9881 - loss: 0.0414
Epoch 5: Train accuracy = 0.9880, Test accuracy = 0.9793
```