Gradient Descent is the most fundamental optimization algorithm and commonly used in training deep learning models. It calculates the negative gradient of the loss function with respect to the weights and updates the weights in the direction of the negative gradient.

# code:

import tensorflow as tf

import numpy as np

train\_data = np.random.rand(1000, 10)

train\_labels = np.random.rand(1000, 1)

model = tf.keras.Sequential([

tf.keras.layers.Dense(64, activation='relu', input\_shape=(10,)),

tf.keras.layers.Dense(1)

])

optimizer = tf.keras.optimizers.SGD(learning\_rate=0.01)

model.compile(optimizer=optimizer, loss='mse', metrics=['mae'])

model.fit(train\_data, train\_labels, epochs=10, batch\_size=32)