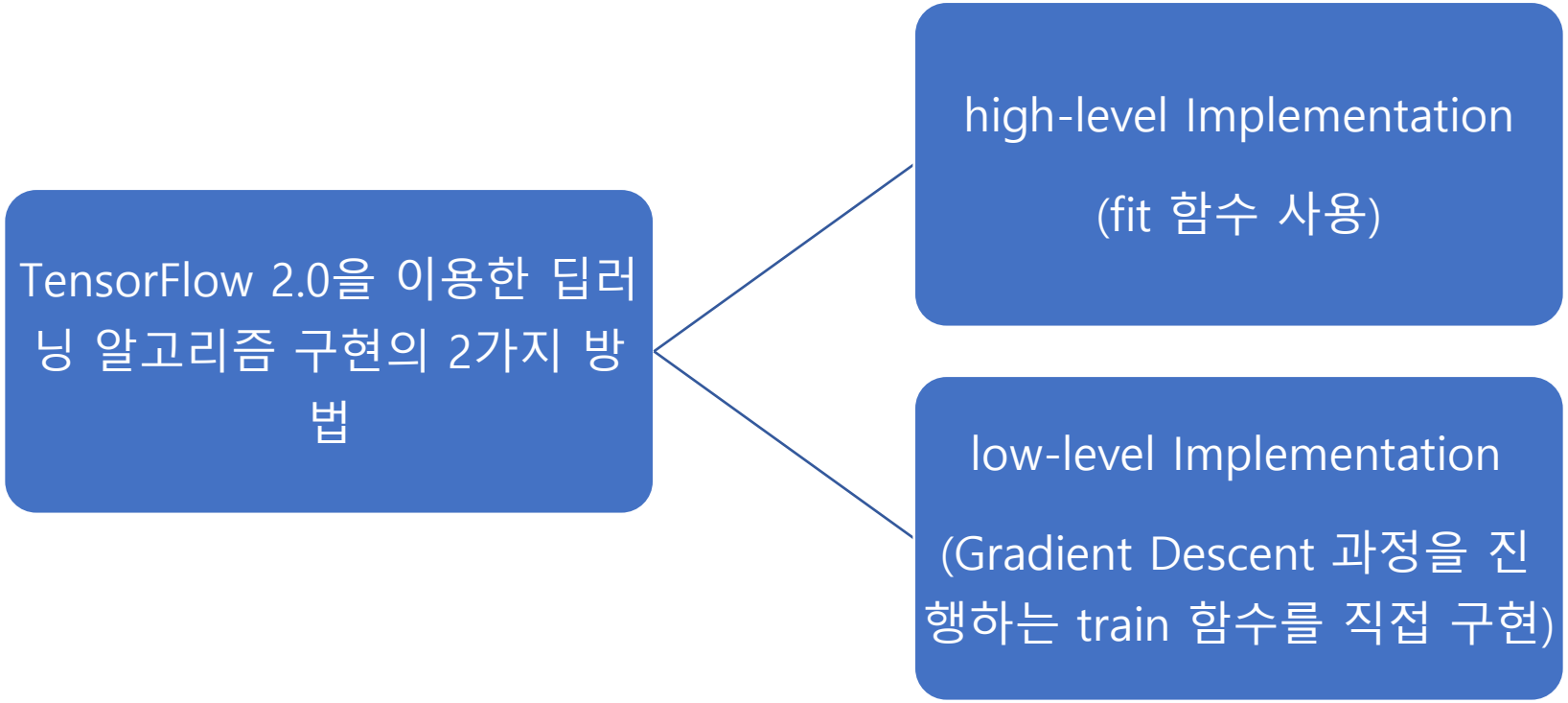
## TensorFlow 2.0을 이용한 딥러닝 알고리즘 구현의 2가지 방법



# TensorFlow 2.0을 이용한 딥러닝 알고리즘 구현의 2가지 방법

- https://www.tensorflow.org/overview/

### For beginners

The best place to start is with the user-friendly Sequential API. You can create models by plugging together building blocks. Run the "Hello World" example below, then visit the **tutorials** to learn more.

To learn ML, check out our **education page**. Begin with curated curriculums to improve your skills in foundational ML areas.

```
import tensorflow as tf
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

model = tf.keras.models.Sequential([
  tf.keras.layers.Flatten(input_shape=(28, 28)),
  tf.keras.layers.Dense(128, activation='relu'),
  tf.keras.layers.Dropout(0.2),
  tf.keras.layers.Dense(10, activation='softmax')
])

model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])

model.fit(x_train, y_train, epochs=5)
model.evaluate(x_test, y_test)
```

### For experts

The Subclassing API provides a define-by-run interface for advanced research. Create a class for your model, then write the forward pass imperatively. Easily author custom layers, activations, and training loops. Run the "Hello World" example below, then visit the **tutorials** to learn more.

```
class MyModel(tf.keras.Model):
  def __init__(self):
    super(MyModel, self).__init__()
    self.conv1 = Conv2D(32, 3, activation='relu')
    self.flatten = Flatten()
    self.d1 = Dense(128, activation='relu')
    self.d2 = Dense(10, activation='softmax')

  def call(self, x):
    x = self.conv1(x)
    x = self.flatten(x)
    x = self.d1(x)
    return self.d2(x)
model = MyModel()

with tf.GradientTape() as tape:
  logits = model(images)
  loss_value = loss(logits, labels)
grads = tape.gradient(loss_value, model.trainable_variable
optimizer.apply_gradients(zip(grads, model.trainable_varia
```

## TensorFlow 2.0을 이용한 예측 모델 구성방법 – beginners style

- TensorFlow 2.0을 이용해서 예측모델을 생성하는 방법은 다음과 같습니다.

## Beginner style을 이용한 MNSIT 숫자 분류기 구현

- Beginner Style을 이용해서 MNIST 숫자 분류기를 구현해봅시다.
- https://colab.research.google.com/drive/1bdyNIJPKVQZG7Y-rlrgQ8gBKTH75xI_n?usp=sharing

# Thank you!