```
یق است که توسط گوگل توسعه داده شده است. این کتابخانه به شما این امکان را میدهد که مدلهای یادگیری عمیق را بسازید و آموزش دهید TensorFlow#
!pip install tensorflow==2.15.0
#!pip install tensorflow==2.15.0 protobuf==3.20.3
!pip install tensorflow-quantum==0.7.3
import importlib, pkg_resources
importlib.reload(pkg resources)
     Show hidden output
وارد كردن كتابخانه هاى لازم #
import tensorflow as tf # كتابخانه TensorFlow مميق عميق
برای یادگیری ماشین کوانتومی TensorFlow Quantum کتابخانه # TensorFlow Quantum عدالت
براى آزمایش صدل Fashion MNIST مجموعه داده # Fashion MNIST مجموعه
from tensorflow.keras import regularizers # ابزارهای منظم سازی در
import cirq # کتابخانه برای کار با مدارهای کوانتومی
import sympy # كتابخانه براى محاسبات نمادين
import numpy as np # کتابخانه برای کار با آرایهها و محاسبات عددی
معيارهاي ارزيابي صال # from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
برای نمایش نمودارها به صورت آنلاین Jupyter Notebook تنظیمات #
%matplotlib inline
import matplotlib.pyplot as plt # کتابخانه برای رسم نمودارها
from cirq.contrib.svg import SVGCircuit \# صورت \# صورت \# SVG
# Fashion MNIST بارگذاری مجموعه داده
(train_images, train_labels), (test_images, test_labels) = fashion_mnist.load_data()
نرمالسازی تصاوی #
train_images = train_images / 255.0 # [1,0] # بازه (0, 1 | 1 سازی مقادیر پیکسلها به بازه (1,0)
(تصاویر خاکستری: یک کانال) QCNN افزودن بعد کانال برای سازگاری با خط لوله #
افزودن بعد جدید به انتهای آرایه تصاویر آموزشی # train_images = train_images[..., np.newaxis]
test_images = test_images[..., np.newaxis] # انجام همان كار براى تصاوير آزمايشى
چاپ شکل داده های آموزشی و آزمایشی قبل از ایجاد زیرمجموعه #
print("Training Data Shape (Before Subset):", train_images.shape)
print("Testing Data Shape (Before Subset):", test_images.shape)
Downloading data from <a href="https://storage.googleapis.com/tensorflow/tf-keras-datasets/train-labels-idx1-ubyte.gz">https://storage.googleapis.com/tensorflow/tf-keras-datasets/train-labels-idx1-ubyte.gz</a>
     29515/29515 [============ ] - Os Ous/step
     Downloading data from <a href="https://storage.googleapis.com/tensorflow/tf-keras-datasets/train-images-idx3-ubyte.gz">https://storage.googleapis.com/tensorflow/tf-keras-datasets/train-images-idx3-ubyte.gz</a>
     26421880/26421880 [=========] - 0s Ous/step
     Downloading data from <a href="https://storage.googleapis.com/tensorflow/tf-keras-datasets/t10k-labels-idx1-ubyte.gz">https://storage.googleapis.com/tensorflow/tf-keras-datasets/t10k-labels-idx1-ubyte.gz</a>
     5148/5148 [=========== ] - Os Ous/step
     Downloading data from <a href="https://storage.googleapis.com/tensorflow/tf-keras-datasets/t10k-images-idx3-ubyte.gz">https://storage.googleapis.com/tensorflow/tf-keras-datasets/t10k-images-idx3-ubyte.gz</a>
     4422102/4422102 [=========== ] - Os Ous/step
     Training Data Shape (Before Subset): (60000, 28, 28, 1)
     Testing Data Shape (Before Subset): (10000, 28, 28, 1)
،تعداد نمونههایی که در زیرمجموعهها خواهند بود #
حفظ نسبت اصلی 6:1 از دادههای آموزشی به آزمایشی #
تعداد كل نمونه هاى آموزشي # num_train = 6000
تعداد كل نمونههاى آزمايشى # num_test = 1000
تعداد نمونه ها در هر کلاس (10 کلاس) #
num train per class = num train // 10 # تعداد نمونه ها در هر کلاس آموزشی
num_test_per_class = num_test // 10 # تعداد نمونه ها در هر کلاس آزمایشی
لیست برای زیرمجموعهها #
train_images_subset = []
train_labels_subset = []
test_images_subset = []
test_labels_subset = []
تکرار بر روی هر کلاس و نمونهگیری برابر #
for i in range(10):
    در دادههای آموزشی i پیدا کربن ایندکسهای کلاس # [0] class_indices_train = np.where(train_labels == i
    در دادههای آزمایشی i پیدا کردن ایندکسهای کلاس # class_indices_test = np.where(test_labels == i)[0]
    نمونهگیری تصادفی از تصاویر در هر کلاس #
    sampled train indices = np.random.choice(class indices train, num train per class, replace=False) نمونه گيری بدون جليگزيني #
```

```
train_images_subset.append(train_images[sampled_train_indices]) # افزودن تصاوير آموزشی نمونهگيری شده
    train_labels_subset.append(train_labels[sampled_train_indices]) # افازودن برچسبهای آموزشی نمونهگیری شده
    test_images_subset.append(test_images[sampled_test_indices]) # افازودن تصاوير أزمايشي نمونهگيري شده
    test_labels_subset.append(test_labels[sampled_test_indices])
                                                                           افزودن برجسبهای آزمایشی نمونهگیری شده #
تبديل ليستها به آرايهها #
ترکیب لیست تصاویر آموزشی به یک آرایه # (train_images_subset = np.concatenate(train_images_subset)
تركيب ليست برچسبهاى أموزشي به يک أرايه # train_labels_subset = np.concatenate(train_labels_subset)
test_images_subset = np.concatenate(test_images_subset)
                                                                 تركيب ليست تصاوير آزمايشي به يک آرايه #
test_labels_subset = np.concatenate(test_labels_subset)
                                                                 تركيب ليست برچسبهاى آزمايشى به يک آرايه #
چاپ شکل دادههای آموزشی و آزمایشی پس از ایجاد زیرمجموعه #
print("Training Data Shape (After Subset):", train_images_subset.shape)
print("Testing Data Shape (After Subset):", test_images_subset.shape)
چاپ کلاسهای منحصر به فرد برای تأیید اینکه همه 10 کلاس شامل شدهاند #
print("Unique Train Labels:", np.unique(train_labels_subset))
print("Unique Test Labels:", np.unique(test_labels_subset))
Training Data Shape (After Subset): (6000, 28, 28, 1)
     Testing Data Shape (After Subset): (1000, 28, 28, 1)
     Unique Train Labels: [0 1 2 3 4 5 6 7 8 9]
     Unique Test Labels: [0 1 2 3 4 5 6 7 8 9]
تابعی برای نمایش تصاویر نمونه #
def display_images(images, labels, num_images=3):
    Selected_classes = [0, 1, 2] # کلاسهای منتخب برای نمایش تصاویر
    لیستهای خالی برای نخیره تصاویر و برچسبها #
    selected_images = []
    selected_labels = []
    انتخاب یک تصویر از هر کلاس منتخب #
    for class_label in selected_classes:
        class_indices = np.where(labels == class_label)[0] # پیدا کردن ایندکسهای کلاس
         selected_images.append(images[class_indices[0]]) # التَخاب اولين تَصوير از كلاس
         selected\_labels.append(labels[class\_indices[0]]) # نخيره برچسب مربوطه
    تبديل ليستها به آرايهها #
    selected_images = np.array(selected_images)
    selected_labels = np.array(selected_labels)
    ایجاد یک شکل برای نمایش تصاویر #
    fig, axes = plt.subplots(1, num_images, figsize=(10, 4)) # ايجاد زيرنمودارها
    for i, ax in enumerate(axes):
        ax.imshow(selected_images[i].squeeze(), cmap='gray') # مايش تصوير با حذف بعد كاتال #
        ax.set_title(f'Label: {selected_labels[i]}') # عنوان بر اساس برچسب
        ax.axis('off') # خاموش کردن محور
    plt.tight_layout() # تنظیمات نهایی
    نمایش تصاویر # plt.show()
نمایش دادههای نمونه از مجموعه داده پاک #
display_images(train_images_subset, train_labels_subset, num_images=3)
<del>_</del>__
                                                                                                                 Label: 2
                       Label: 0
                                                                    Label: 1
```

sampled_test_indices = np.random.choice(class_indices_test, num_test_per_class, replace=False)

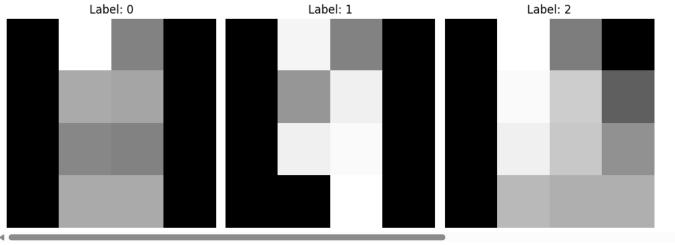
افزودن به لیست #

نمونهگیری بدون جایگزینی #

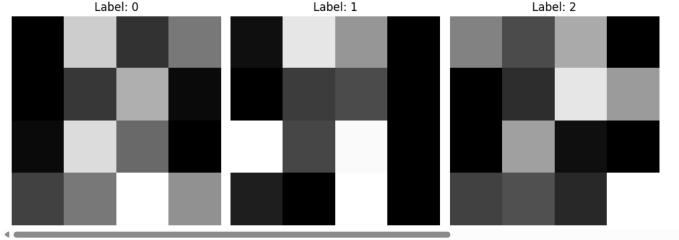
```
# تابعي براى تغيير الدازه تصاوير به الدازه مدف و تبديل به آرايه # Numpy

# تعيير الدازه المدف و تبديل به آرايه # آرايه # آرايه # آرايه الدازه تصاوير أموزشي و آزمايشي # تغيير الدازه تصاوير أموزشي # آرمايشي # تغيير الدازه تصاوير آرمايشي # آرمايشي بس از تغيير الدازه تصاوير آزمايشي بس از تغيير الدازه # تغيير الدازه تصاوير آزمايشي بس از تغيير الدازه الد
```

Resized Training Data Shape: (6000, 4, 4, 1)
Resized Testing Data Shape: (1000, 4, 4, 1)



```
تابعی برای افزودن نویز گوسی به تصاویر #
def add_gaussian_noise(images, mean=0.0, severity=1):
    سطوح شدت نويز # severity_levels = [0.08, 0.12, 0.18, 0.26, 0.38]
    دریافت انحراف استاندارد بر اساس شدت (1 تا 5) # (5 تا 7) stddev = severity_levels
    noise = np.random.normal(mean, stddev, images.shape) # توليد نويز گوسى
    noisy_images = images + noise # افزودن نویز به تصاویر
    اطمينان از اينكه مقادير پيكسلها در بازه [0.0, 0.0] هستند # (1.0 noisy_images = np.clip(noisy_images, 0.0, 1.0)
    return noisy_images
افزودن نویز به دادههای آموزشی و آزمایشی #
train_data_noisy = add_gaussian_noise(train_data_clean, severity=5) # افزودن نويز به دادههای آموزشی
test_data_noisy = add_gaussian_noise(test_data_clean, severity=5) # افزودن نویز به دادههای آزمایشی
چاپ شکل داده های آموزشی و آزمایشی با نویز #
print("Noisy Training Data Shape:", train_data_noisy.shape)
print("Noisy Testing Data Shape:", test_data_noisy.shape)
نمایش دادههای نمونه از دادههای آموزشی با نویز #
display_images(train_data_noisy, train_labels_subset, num_images=3)
```



```
تعریف آستانه برای فعالسازی حالت کوبیتها #
THRESHOLD = 0.5
ایجاد کدگذاری باینری برای دادههای آموزشی و آزمایشی #
x_train_bin = np.array(train_data_clean > THRESHOLD, dtype=np.float32) # كدگذاری بلانری برای دادههای آموزشی پاک #
x_train_noisy_bin = np.array(train_data_noisy > THRESHOLD, dtype=np.float32) تدگذاری بلانری برای دادههای آموزشی با نویز # (x_train_noisy_bin = np.array(train_data_noisy > THRESHOLD, dtype=np.float32)
تابعی برای تبدیل تصاویر کلاسیک به دادههای کوانتومی #
def convert_to_circuit(image):
   values = np.ndarray.flatten(image) # تبدیل تصویر به یک آرایه یک بعدی
    qubits = cirq.GridQubit.rect(4, 4) # 4 در 4 # 4 بجاد یک شبکه از کوبیتها به ابعاد 4 در 4
    ایجاد یک مدار کوانتومی خالی # (circuit = cirq.Circuit)
    for i, value in enumerate(values):
        if value: # باشد برابر با 1 (فعال) باشد
           به کوبیت مربوطه X اضافه کردن گیت # (circuit.append(cirq.X(qubits[i]))
    return circuit # مدار کوانتومی
تبدیل تصاویر باینری تمیز به مدارهای کوانتومی #
برای دادههای آموزشی تمیز # train_clean_circ = [convert_to_circuit(x) for x in x_train_bin] برای دادههای آموزشی
برای دادههای آزمیشی تعیز # test_clean_circ = [convert_to_circuit(x) for x in x_test_bin]
تبدیل تصاویر باینری با نویز به مدارهای کوانتومی #
برای دادههای آموزشی با نویز # [convert_to_circuit(x) for x in x_train_noisy_bin] برای دادههای آموزشی
test_noisy_circ = [convert_to_circuit(x) for x in x_test_noisy_bin] # برای دادههای آزمایشی با نویز
نمایش کوبیت هایی که روی آن ها گیت قرار گرفته است #
نمایش مدار کوانتومی اولین تصویر تمیز # ([0] SVGCircuit(train_clean_circ
```

```
WARNING:matplotlib.font_manager:findfont: Font family 'Arial' not found.
        WARNING:matplotlib.font_manager:findfont: Font family 'Arial' not found. WARNING:matplotlib.font_manager:findfont: Font family 'Arial' not found.
            (0, 1):
                              Χ
            (1, 1): -
                              Χ
            (1, 2): -
                              Χ
            (2, 1): -
            (3, 1):
            (3, 2):
                              Χ
TensorFlow Quantum به تتسورهای cirq تبدیل مدارهای #
تبنيل مدارهای آموزشی نمیز به نتسور # (train_clean_tfcirc = tfq.convert_to_tensor(train_clean_circ
test_clean_tfcirc = tfq.convert_to_tensor(test_clean_circ)
                                                                                                               تبدیل مدارهای آزمایشی تمیز به نتسور #
تبنیل مدارهای آموزشی با نویز به نتسور # (train_noisy_tfcirc = tfq.convert_to_tensor(train_noisy_circ
تبدیل مدارهای آزملیشی با نویز به تنسور # (test_noisy_tfcirc = tfq.convert_to_tensor(test_noisy_circ
كلاس براى ايجاد لايههاى مدار كوانتومى #
class CircuitLayerBuilder:
       def __init__(self, data_qubits, readout):
              self.data_qubits = data_qubits # کوبیتهای داده
              self.readout = readout
                                                                             روش خوانش #
       def add_layer(self, circuit, prefix):
               باینری QCNN مشابه مدل ZZ و XX اعمال در همتنیدگی #
              for i in range(len(self.data_qubits) - 1):
                      circuit.append(cirq.XX(self.data_qubits[i], self.data_qubits[i + 1])) درهېتَدگی # (XX
                      zz درهېټيدگۍ # (circuit.append(cirq.ZZ(self.data_qubits[i], self.data_qubits[i + 1])) درهېټيدگۍ #
              اعمال گیتهای کوانتومی پارامتری به هر کوبیت #
              for i, qubit in enumerate(self.data_qubits):
                      تعریف پارامترها و افزودن تنظیمکننده #
                      #lambda_ = sympy.Symbol(f'{prefix}_lambda_{i}', regularizer=regularizers.12(0.01)) # براى استفادههاى آبِنده
                      theta = sympy. Symbol(f'\{prefix\}_theta_{\{i\}'}, regularizer=regularizers. 12(0.01)) \  \  \, \forall theta = sympy. Symbol(f'\{prefix\}_theta_{\{i\}'}, regularizer=regularizers. 12(0.01)) \  \  \, \forall theta = sympy. Symbol(f'(sympholes)) \  \  \, \forall theta = sympy. Symbol(f'(
                      phi = sympy.Symbol(f'{prefix}_phi_{i}', regularizer=regularizers.12(0.01))
                      افزودن گیتهای چرخشی پارامتری #
                      #circuit.append(cirq.rx(lambda_).on(qubit)) # گيت # RX (غيرفعال)
                      circuit.append(cirq.ry(theta).on(qubit)) # گیت # theta
                      circuit.append(cirq.rz(phi).on(qubit)) # گیت # RZ با پارامتر RZ
                      circuit.append(cirq.ry(theta).on(qubit)) # دوباره گیت # theta
                      circuit.append(cirq.rz(phi).on(qubit))
                                                                                                phi با پارامتر RZ دوباره گیت #
              تجمع مبتتی بر اندازهگیری (غیرفعال) #
              استفاده نمیشود زیرا دقت را کاهش میدهد زیرا اندازه دادهها کوچک است #
              #for i in range(0, len(self.data_qubits), 4):
                      #circuit.append(cirq.CNOT(self.data qubits[i], self.data qubits[i + 1]))
                      #circuit.append(cirq.measure(self.data_qubits[i + 1]))
              بهروزرسانی کوبیتها برای نشان دادن تجمع (غیرفعال) #
              #self.data_qubits = [self.data_qubits[i] for i in range(0, len(self.data_qubits), 2)]
تعریف کوبیتها و کوبیت خوانش #
data_qubits = cirq.GridQubit.rect(2, 2) # 2 اليجاد يک شبکه 2 x2
readout = cirq.GridQubit(-1, -1)
                                                                          تعریف کوبیت خوانش #
```

ابجاد یک مدار نمونه #

```
demo_builder = CircuitLayerBuilder(data_qubits=data_qubits, readout=readout) ابجاد یک شی از کلاس # (CircuitLayerBuilder
اليجاد يک مدار کوانتومي خالي # circuit = cirq.Circuit()
افزودن لابهها به مدار #
demo_builder.add_layer(circuit, prefix="") # افزودن لايه به مدار
نمایش مدار #
نمایش مدار کوانتومی به صورت گرافیکی # SVGCircuit(circuit)
WARNING:matplotlib.font_manager:findfont: Font family 'Arial' not found.
     WARNING: matplotlib.font_manager: findfont: Font family 'Arial' not found.
     WARNING:matplotlib.font_manager:findfont: Font family 'Arial' not found.
     WARNING: matplotlib.font manager: findfont: Font family 'Arial' not found.
     WARNING:matplotlib.font_manager:findfont: Font family 'Arial' not found.
     WARNING:matplotlib.font manager:findfont: Font family 'Arial' not found.
     WARNING:matplotlib.font_manager:findfont: Font family 'Arial' not found.
     WARNING: matplotlib.font_manager: findfont: Font family 'Arial' not found.
     WARNING:matplotlib.font_manager:findfont: Font family 'Arial' not found.
     WARNING: matplotlib.font manager: findfont: Font family 'Arial' not found.
     WARNING:matplotlib.font_manager:findfont: Font family 'Arial' not found.
       (0,0): -
                 XX
                         ZZ
                                Ry(_theta_0)
                                                Rz(_phi_0)
                                                             Ry(_theta_0)
                                                                             Rz(_phi_0)
                 т
                         т
                         ZZ
                                                             Ry(_theta_1)
                                                                             Rz(_phi_1)
                                                                                          Ry(_theta_1)
                                                                                                         Rz(_phi_1)
                 XX
                                     XX
                                                   ZZ
       (0, 1):
       (1, 0):
                                     XX
                                                   ZZ
                                                                 XX
                                                                                ZZ
                                                                                          Ry(_theta_2)
                                                                                                         Rz(_phi_2)
                                                                                                                       Ry(_theta_2)
                                                                                                                                      Rz(_phi_2)
       (1, 1): -
                                                                 XX
                                                                                77
                                                                                          Ry(_theta_3)
                                                                                                         Rz(_phi_3)
                                                                                                                       Ry(_theta_3)
                                                                                                                                      Rz(_phi_3)
و عملیات خوانش QCNN تابعی برای ایجاد مدل کوانتومی #
def create_quantum_model():
    data_qubits = cirq.GridQubit.rect(4, 4) # 4 البجاد یک شبکه 4 x4 برای کوبیتهای داده x4ابجاد یک شبکه
    readout = cirq.GridQubit(-1, -1)
                                               تعریف کوبیت خوانش در موقعیت [-1,-1] #
    circuit = cirq.Circuit()
                                                 ایجاد یک مدار کوانتومی خالی #
    افزودن لایههای گیت به مدار کوانتومی #
    builder = CircuitLayerBuilder(data_qubits=data_qubits, readout=readout) البجاد شي از كلاس # CircuitLayerBuilder
    builder.add_layer(circuit, prefix="layer1") # افزودن لايه اول #
    builder.add_layer(circuit, prefix="layer2") # افزودن لایه دوم
    آمادهسازي كوبيت خوانش #
    circuit.append(cirq.H(readout)) # خوانش خوانش هادامارد به کوبیت خوانش
    circuit.append(cirq.measure(readout)) # اندازهگیری کوبیت خوانش
    اندازهگیری تمام کوبیتها برای طبقهبندی چندکلاسی #
    measurements = [cirq.Z(qubit) for qubit in data_qubits] # انداز مگیری کوبیتهای داده
    return circuit, measurements # بازگشت مدار و نتایج اندازهگیری
ایجاد مدار کوانتومی و عملیات خوانش #
model_circuit, model_readout = create_quantum_model()
ساخت مدل تميز #
clean_model = tf.keras.Sequential([
    tf.keras.layers.Input(shape=(), dtype=tf.dtypes.string), # مدل به صورت رشته
```

```
با مدار کوانتومی و خوانش PQC لایه # مدار کوانتومی و خوانش PQC الایه # با مدار کوانتومی و خوانش PQC با کوانتومی PQC با کوانتومی و خوانش PQC با کوانتوم
                                 tf.keras.layers.Dense(8, activation='relu', kernel_regularizer=regularizers.l2(0.01)), # با 8 نورون و تلبع فعال الله # ReLU
                                 tf.keras.layers.Dense(10, activation='softmax', kernel_regularizer=regularizers.12(0.01)) # 4/ Dense (الله # Dense كلاسي 10 كلاس
 ])
كامپايل كردن مدل #
clean_model.compile(optimizer='adam', # استفاده از بهینهساز Adam
                                                                                                                                                          loss=tf.keras.losses.SparseCategoricalCrossentropy(), # تلبع هزينه براى طبقهبندى چندكلاسى
                                                                                                                                                         metrics=['accuracy']) # معيار دقت
```

چاپ خلاصه مدل #

print(clean_model.summary()) # نمایش ساختار مدل

→ Model: "sequential"

Layer (type)	Output Shape	Param #
pqc (PQC)	(None, 16)	64
dense (Dense)	(None, 8)	136
dense_1 (Dense)	(None, 10)	90

Total params: 290 (1.13 KB) Trainable params: 290 (1.13 KB) Non-trainable params: 0 (0.00 Byte)

:در اینجا نمایش بصری مدل مدار کوانتومی آمده است# SVGCircuit(model_circuit)

```
WARNING:matplotlib.font_manager:findfont: Font family 'Arial' not found.
        WARNING:matplotlib.font manager:findfont: Font family 'Arial' not found.
        WARNING:matplotlib.font_manager:findfont: Font family 'Arial' not found.
        {\tt WARNING:matplotlib.font\_manager:findfont:} \ \ {\tt Font\ family\ 'Arial'\ not\ found.}
        WARNING:matplotlib.font_manager:findfont: Font family 'Arial' not found.
        WARNING:matplotlib.font manager:findfont: Font family 'Arial' not found.
        {\tt WARNING:matplotlib.font\_manager:findfont:} \ \ {\tt Font\ family\ 'Arial'\ not\ found.}
        WARNING:matplotlib.font_manager:findfont: Font family 'Arial' not found.
        WARNING: matplotlib.font manager: findfont: Font family 'Arial' not found.
        WARNING:matplotlib.font_manager:findfont: Font family 'Arial' not found.
        WARNING:matplotlib.font_manager:findfont: Font family 'Arial' not found.
        WARNING: matplotlib.font_manager: findfont: Font family 'Arial' not found.
        WARNING:matplotlib.font manager:findfont: Font family 'Arial' not found.
        WARNING:matplotlib.font_manager:findfont: Font family 'Arial' not found.
        WARNING: matplotlib.font manager: findfont: Font family 'Arial' not found.
        WARNING:matplotlib.font_manager:findfont: Font family 'Arial' not found.
        WARNING:matplotlib.font_manager:findfont: Font family 'Arial' not found.
        WARNING:matplotlib.font_manager:findfont: Font family 'Arial' not found.
        WARNING:matplotlib.font manager:findfont: Font family 'Arial' not found.
        WARNING:matplotlib.font_manager:findfont: Font family 'Arial' not found.
        WARNING:matplotlib.font_manager:findfont: Font family 'Arial' not found.
آموزش مدل تمیز بر روی مجموعه دادههای تمیز #
history_clean = clean_model.fit(
          دادههای آموزشی و برچسبها # train_clean_tfcirc, train_labels_subset,
          لدازه بچ 32 # 32 اندازه بچ
          تعداد دورههای آموزشی # epochs=10,
          نمایش جزئیات آموزش # verbose=1,
          دادههای اعتبارسنجی # validation_data=(test_clean_tfcirc, test_labels_subset)
)
ارزیابی مدل بر روی دادههای آزمایشی #
test_loss_clean, test_acc_clean = clean_model.evaluate(test_clean_tfcirc, test_labels_subset, verbose=2)
حاب دقت و از دست دادن آزمون #
print(f"Clean Data - Test Accuracy: {test_acc_clean:.4f}, Test Loss: {test_loss_clean:.4f}")
print(f"Clean Data - Training Accuracy: {history_clean.history['accuracy'][-1]:.4f}")
₩ABNANG/matplotlib.font_manager:findfont: Font family 'Arial' not found.
        \daggreen \dagg
        世後日本 | Paraget | Paraget
        Laborn 3/10 matplotlib_font_manager:findfont: -Fgqtsfqgj}YepArialssmot.fgqtgd. accuracy: 0.3472 - val_loss: 1.9077 - val_accuracy: 0.372€
₩ABNING;qqtplotlib.font_manager:findfont: Font family 'Arial' not found.
        WARNING; motplotlib.font_manager:findfont: Font family
        HABNING:matelotlib_fent_manager:findfent: -Fgntsfmrild:schot. 88岁gd- accuracy: 0.3687 - val_loss: 1.8307 - val_accuracy: 0.3930
HABNING:matelotlib.font_manager:findfont: Font family 'Arial' not found.
        ظABNANG/matplotlib.font_manager:findfont: Font family
```

```
WARNING: matelotlib.font_manager:findfont: F924sfamily 'Arial' not found.

WARNING: matelotlib.font_manager:findfont: F024sfamily 'Arial' not found.
    WARNING; Matplotlib.font_manager:findfont: Font family 'Arial' not found.
    WARNING: #atplotlib_font_manager:findfont: F97tsfamily "Arial" not found.
WARNING: #atplotlib.font_manager:findfont: F97tsfamily "Arial" not found.
WARNING: #atplotlib.font_manager:findfont: Font family "Arial" not found.
    WARNING:matplotlib.font_manager:findfont: Font family 'Arial' not found.
تابعی برای نمایش منحنی های آموزش و اعتبار سنجی #
def plot_metrics(history, metric_name, title, ylim=1, xlim=10):
   عنوان نمودار # plt.title(title)
   plt.ylim(0, ylim) # تظیم محدودیت محور عمودی
   تنظيم محدوديت محور افقى # plt.xlim(0, xlim)
   plt.plot(history.history[metric_name], color='blue', label='Train') # نمودار دقت أموزش
   plt.plot(history.history['val_' + metric_name], color='green', label='Validation') " نمودار دقت اعتبارسنجي
   مكان قرارگيرى افسانه # (loc='lower right') مكان قرارگيرى
   plt.xlabel("Epoch") # برچسب محور افقى
   plt.ylabel("Accuracy") # برچسب محور عمودی
رسم منحنی های دقت برای مدل تمیز #
plot_metrics(history_clean, "accuracy", "QCNN Accuracy on Clean Data - 4x4 Images, 10 Classes")
              · _ . _ . _ . _ . _ . _ . _ . _
                            QCNN Accuracy on Clean Data - 4x4 Images, 10 Classes
        1.0
       0.8
       0.6
     Accuracy
        0.4
        0.2
                                                           Train
                                                           Validation
        0.0
                                     Epoch
    WARRING, MIGCPTOCITE, FORCE MIGHAGET . I THATONC. FORCE FAMILY ALICE HOC FOUND.
    WARNING:matplotlib.font manager:findfont: Font family 'Arial' not found.
ساخت و کامپایل یک مدل دیگر برای دادههای نویزی #
noisy_model = tf.keras.Sequential([
   tf.keras.layers.Input(shape=(), dtype=tf.dtypes.string), # ورودی مدل به صورت رشته
   با مدار كوانتومي و خوانش PQC لايه # با مدار كوانتومي و خوانش PQC لايه # با مدار كوانتومي و خوانش PQC
   tf.keras.layers.Dense(8, activation='relu', kernel_regularizer=regularizers.12(0.01)), # با 8 نورون و تابع فعالسازی Dense لایه # ReLU
   tf.keras.layers.Dense(10, activation='softmax', kernel_regularizer=regularizers.l2(0.01)) # براى طبقبندى چندکلاسي (10 کلاس) Dense لایه #
])
كامپايل كردن مدل #
noisy_model.compile(optimizer='adam', # استفاده از بهینهساز Adam
                تلبع هزينه برای طبقهبندی چندکلاسی # (), # اloss=tf.keras.losses.SparseCategoricalCrossentropy
                metrics=['accuracy']) # معيار دقت
چاپ خلاصه مدل #
print(noisy_model.summary()) # نمایش ساختار مدل
    WAKNING:matplotlib.tont_manager:findfont: Font family 'Arial' not found.
    WARRING: Madptotiab_font_manager: findfont: Font family 'Arial' not found.
    WARNING:matplotlib.font_manager:findfont: Font_family 'Arial' not found.
    WARMENG(MXRB)otlib.font managertrundsbare Font family 'RATAT'#not found.
    WARNING:matplotlib:font_manager:findfont:=Font=family=*Arial*=not found.
    WARNING(RQ€)lotlib.font_manageNortnd+6At: Font family 'A4ial' not found.
    WARNING:matplotlib.font_manager:findfont: Font family 'Arial' not found.
    WARNING?m&@pT6€lib.font_manageNoff⊈nd&dnt: Font family 'AP€al' not found.
```

```
WARNING:matplotlib.font manager:findfont: Font family 'Arial' not found.
        WaBNING3mépphot}ib.font_managénofièpdf0nt: Font family '90ial' not found.
        {\tt WARNING:matplotlib.font\_manager:findfont:} \ \ {\tt Font\ family\ 'Arial'\ not\ found.}
        WARNING:matplotlib=font_manager:findfont:=Font=family='Arial'=not found.
        WARNINGamamplo190b(font_mb)ager:findfont: Font family 'Arial' not found.
        WABNAN61mapprankib290nt1maBagBn; findfont: Font family 'Arial' not found.
        NARNINGimatheoparamsono_monogeByfendfont: Font family 'Arial' not found.
        WARNING:matplotlib.font_manager:findfont: Font family 'Arial' not found.
        WARNING:matplotlib.font_manager:findfont: Font family 'Arial' not found.
        WABNIU6cmàtplbtþýbh6nBt1@addagerpátkdgentkeFontsfomihýtiAdiadrshóntifómhdzers.py:120: UserWarning: The initializer RandomUniform is ເ
        WARNEnmestplophlib.font_manager:findfont: Font family 'Arial' not found.
        WARNING:matplotlib.font_manager:findfont: Font family 'Arial' not found. WARNING:matplotlib.font_manager:findfont: Font family 'Arial' not found.
آموزش مدل نویزی بر روی مجموعه داده های نویزی #
history_noisy = noisy_model.fit(
           دادههای آموزشی و برچسبها # train_noisy_tfcirc, train_labels_subset,
          لندازه بچ 32 # 32 اندازه بچ
          تعداد دورههای آموزشی # epochs=10,
          نمایش جزئیات آموزش # verbose=1,
          validation_data=(test_noisy_tfcirc, test_labels_subset) # دادههای اعتبار سنجی
)
ارزیابی مدل بر روی دادههای آزمایشی #
test_loss_noisy, test_acc_noisy = noisy_model.evaluate(test_noisy_tfcirc, test_labels_subset, verbose=2)
چاپ دقت و از دست دادن آزمون #
print(f"Noisy Data - Test Accuracy: {test_acc_noisy:.4f}, Test Loss: {test_loss_noisy:.4f}")
print(f"Noisy Data - Training Accuracy: {history_noisy.history['accuracy'][-1]:.4f}")
        WARNING/motplotlib.font_manager:findfont: Font family 'Arial' not found.
         ₩ĠBŊĬŊŚ:matplotlib±font_manager:findfont:-Fontsfam;l¥epAriædssnot.fond. accuracy: 0.1238 - val_loss: 2.4016 - val_accuracy: 0.1670
        WARNING, matplotlib.font_manager:findfont: Font family 'Arial' not found.
        ₩ÅBŊŦŊŚ:『patplotlib_font_manager:findfont: Font family 'Arial' not found.

\#ÅBŊŦŊŚ: patplotlib.font_manager:findfont: Font family 'Arial' not found.
        \lambda8\Pi86 matplotlib_foot_manager:findfoot: -F9ntsfagjsYepArialssnot.fag9d4 accuracy: 0.2095 - val_loss: 2.1619 - val_accuracy: 0.219(
፵/BNENG/matplotlib.font_manager:findfont: Font family 'Arial' not found.
        ₩88/186: matplotlib_font_manager:findfont:-Fontsfam;lyepArialssnot.found- accuracy: 0.2248 - val_loss: 2.0726 - val_accuracy: 0.2380
        MARNING/matplotlib.font_manager:findfont: Font family 'Arial' not found.
        \delta be accuracy: 0.2852 - val_loss: 1.9816 - val_accuracy: 0.311{
\delta be accuracy: 0.2852 - val_loss: 1.9816 - val_accuracy: 0.311{
\delta be accuracy: 0.10 - val_accuracy: 0.311{
\delta be accuracy: 0.2852 - val_loss: 1.9816 - val_accuracy: 0.311{
\delta be accuracy: 0.311{
\delta 
        ሥልፀሃዝዋና ጥልቷበቀ፤ ከታ ያወደ መጠብጀውድ findfort: - Fonts fagj sych ለናነዊ of sound.
ሦሳፅሃዝዋና ጥልቲያ of sound of so
        WABNING: matplotlib.font_manager:findfont: Font family 'Arial' not found.

WABNINGOMADPLOTLIB font_manager:findfont: Font family 'Arial' not found.
        ₩88V¥86:matplotlib_foot_manager:findfoot:-Fontsfay}skepAriabssnot.699Dd- accuracy: 0.3057 - val_loss: 1.9489 - val_accuracy: 0.3170
        ŊġŖŊĬŇĠ:ġġtplosłip.fogismanagerufiacyfoot315ont famidiochAciałomgostifound.
WARNJNGiratploslibafeOfargragesifoodfest:L5egt famidy 'Arial' not found.
        NOISY Data - Test Accuracy: -0:3176; Test Loss: 1:9400
WASBJN6a@atplopslihifignac@anasge:5igd5gnt: Font family 'Arial' not found.
WARNING:matplotlib.font manager:findfont: Font family 'Arial' not found.
رسم معیارهای مربوط به مدل نویزی #
plot_metrics(history_noisy, "accuracy", "QCNN Accuracy on Noisy Data - 4x4 Images, 10 Classes")
        WARNING:matplotlib.font_manager:findfont: Font family 'Arial' not found.
        {\tt WARNING:matplotlib.font\_manager:findfont:} \ \ {\tt Font\ family\ 'Arial'\ not\ found.}
        WARNING:matplotlib.font_manager:findfont: Font family 'Arial' not found.
        WARNING:matplotlib.font_manager:findfont: Font family 'Arial' not found.
        WARNING: matplotlib.font manager: findfont: Font family 'Arial' not found.
```