#QML-HEP GSoC 2023 Task Solutions

##Installing Required Package

```
!pip install cirq
!pip install qiskit
!pip install pylatexenc
!pip install pennylane
!pip install -U tensorflow-addons
# # !pip install -q tensorflow==2.3.1
# !pip install -q tensorflow-quantum
```

```
Looking in indexes: https://pypi.org/simple, https://us-
python.pkg.dev/colab-wheels/public/simple/
Collecting cirq
  Downloading cirq-1.1.0-py3-none-any.whl (7.7 kB)
Collecting cirq-aqt==1.1.0
  Downloading cirq_aqt-1.1.0-py3-none-any.whl (27 kB)
Collecting cirq-rigetti==1.1.0
  Downloading cirq_rigetti-1.1.0-py3-none-any.whl (66 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 66.4/66.4 KB 3.9 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 577.4/577.4 KB 17.7 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 57.6/57.6 KB 7.0 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 594.6/594.6 KB 48.4 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.8/1.8 MB 66.9 MB/s eta
0:00:00
ent already satisfied: requests~=2.18 in
/usr/local/lib/python3.9/dist-packages (from cirq-aqt==1.1.0->cirq)
(2.27.1)
Requirement already satisfied: numpy<1.24,>=1.16 in
/usr/local/lib/python3.9/dist-packages (from cirq-core==1.1.0->cirq)
(1.22.4)
Requirement already satisfied: sortedcontainers~=2.0 in
/usr/local/lib/python3.9/dist-packages (from cirq-core==1.1.0->cirq)
(2.4.0)
Requirement already satisfied: matplotlib~=3.0 in
/usr/local/lib/python3.9/dist-packages (from cirq-core==1.1.0->cirq)
(3.7.1)
Requirement already satisfied: tqdm in /usr/local/lib/python3.9/dist-
packages (from cirq-core==1.1.0->cirq) (4.65.0)
Collecting duet~=0.2.7
  Downloading duet-0.2.7-py3-none-any.whl (28 kB)
Requirement already satisfied: pandas in
/usr/local/lib/python3.9/dist-packages (from cirq-core==1.1.0->cirq)
(1.4.4)
Requirement already satisfied: sympy in /usr/local/lib/python3.9/dist-
```

```
packages (from cirq-core==1.1.0->cirq) (1.11.1)
Collecting networkx~=2.4
  Downloading networkx-2.8.8-py3-none-any.whl (2.0 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.0/2.0 MB 66.8 MB/s eta
0:00:00
ent already satisfied: typing-extensions in
/usr/local/lib/python3.9/dist-packages (from cirq-core==1.1.0->cirq)
(4.5.0)
Requirement already satisfied: scipy in /usr/local/lib/python3.9/dist-
packages (from cirq-core==1.1.0->cirq) (1.10.1)
Requirement already satisfied: protobuf<4,>=3.15.0 in
/usr/local/lib/python3.9/dist-packages (from cirq-google==1.1.0->cirq)
(3.20.3)
Collecting google-api-core[grpc]<2.0.0dev,>=1.14.0
  Downloading google_api_core-1.34.0-py3-none-any.whl (120 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 120.2/120.2 KB 16.8 MB/s eta
0:00:00
ent already satisfied: proto-plus>=1.20.0 in
/usr/local/lib/python3.9/dist-packages (from cirq-google==1.1.0->cirq)
(1.22.2)
Collecting pyquil>=3.2.0
  Downloading pyquil-3.3.4-py3-none-any.whl (221 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 221.6/221.6 KB 28.3 MB/s eta
0:00:00
ent already satisfied: googleapis-common-protos<2.0dev,>=1.56.2 in
/usr/local/lib/python3.9/dist-packages (from google-api-
core[grpc]<2.0.0dev,>=1.14.0->cirq-google==1.1.0->cirq) (1.59.0)
Requirement already satisfied: google-auth<3.0dev,>=1.25.0 in
/usr/local/lib/python3.9/dist-packages (from google-api-
core[grpc]<2.0.0dev,>=1.14.0->cirq-google==1.1.0->cirq) (2.17.0)
Requirement already satisfied: grpcio-status<2.0dev,>=1.33.2 in
/usr/local/lib/python3.9/dist-packages (from google-api-
core[grpc]<2.0.0dev,>=1.14.0->cirq-google==1.1.0->cirq) (1.48.2)
Requirement already satisfied: grpcio<2.0dev,>=1.33.2 in
/usr/local/lib/python3.9/dist-packages (from google-api-
core[grpc]<2.0.0dev,>=1.14.0->cirq-google==1.1.0->cirq) (1.53.0)
Requirement already satisfied: pillow>=6.2.0 in
/usr/local/lib/python3.9/dist-packages (from matplotlib~=3.0->cirq-
core==1.1.0->cirq) (8.4.0)
Requirement already satisfied: importlib-resources>=3.2.0 in
/usr/local/lib/python3.9/dist-packages (from matplotlib~=3.0->cirq-
core==1.1.0->cirq) (5.12.0)
Requirement already satisfied: python-dateutil>=2.7 in
/usr/local/lib/python3.9/dist-packages (from matplotlib~=3.0->cirq-
core==1.1.0->cirq) (2.8.2)
Requirement already satisfied: packaging>=20.0 in
/usr/local/lib/python3.9/dist-packages (from matplotlib~=3.0->cirq-
core==1.1.0->cirq) (23.0)
Requirement already satisfied: contourpy>=1.0.1 in
/usr/local/lib/python3.9/dist-packages (from matplotlib~=3.0->cirq-
```

```
core==1.1.0->cirq) (1.0.7)
Requirement already satisfied: pyparsing>=2.3.1 in
/usr/local/lib/python3.9/dist-packages (from matplotlib~=3.0->cirq-
core==1.1.0->cirq) (3.0.9)
Requirement already satisfied: fonttools>=4.22.0 in
/usr/local/lib/python3.9/dist-packages (from matplotlib~=3.0->cirq-
core==1.1.0->cirq) (4.39.3)
Requirement already satisfied: kiwisolver>=1.0.1 in
/usr/local/lib/python3.9/dist-packages (from matplotlib~=3.0->cirq-
core==1.1.0->cirq) (1.4.4)
Requirement already satisfied: cycler>=0.10 in
/usr/local/lib/python3.9/dist-packages (from matplotlib~=3.0->cirq-
core==1.1.0->cirq) (0.11.0)
Collecting lark<0.12.0,>=0.11.1
  Downloading lark-0.11.3.tar.gz (229 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 229.9/229.9 KB 26.0 MB/s eta
0:00:00
etadata (setup.py) ... ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━
147.4/147.4 KB 13.3 MB/s eta 0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 45.6/45.6 KB 4.8 MB/s eta
0:00:00
etadata (setup.py) ... ent already satisfied: idna<4,>=2.5 in
/usr/local/lib/python3.9/dist-packages (from requests~=2.18->cirq-
aqt==1.1.0->cirq) (3.4)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in
/usr/local/lib/python3.9/dist-packages (from requests~=2.18->cirq-
aqt==1.1.0->cirq) (1.26.15)
Requirement already satisfied: charset-normalizer~=2.0.0 in
/usr/local/lib/python3.9/dist-packages (from requests~=2.18->cirq-
aqt==1.1.0->cirq) (2.0.12)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.9/dist-packages (from requests~=2.18->cirq-
aqt==1.1.0->cirq) (2022.12.7)
Requirement already satisfied: pytz>=2020.1 in
/usr/local/lib/python3.9/dist-packages (from pandas->cirq-core==1.1.0-
>cirq) (2022.7.1)
Requirement already satisfied: mpmath>=0.19 in
/usr/local/lib/python3.9/dist-packages (from sympy->cirq-core==1.1.0-
>cirq) (1.3.0)
Requirement already satisfied: six>=1.9.0 in
/usr/local/lib/python3.9/dist-packages (from google-
auth<3.0dev,>=1.25.0->google-api-core[grpc]<2.0.0dev,>=1.14.0->cirq-
google==1.1.0->cirq) (1.16.0)
Requirement already satisfied: rsa<5,>=3.1.4 in
/usr/local/lib/python3.9/dist-packages (from google-
auth<3.0dev,>=1.25.0->google-api-core[grpc]<2.0.0dev,>=1.14.0->cirq-
google==1.1.0->cirq) (4.9)
Requirement already satisfied: cachetools<6.0,>=2.0.0 in
/usr/local/lib/python3.9/dist-packages (from google-
auth<3.0dev,>=1.25.0->google-api-core[grpc]<2.0.0dev,>=1.14.0->cirq-
```

```
google==1.1.0->cirq) (5.3.0)
Requirement already satisfied: pyasn1-modules>=0.2.1 in
/usr/local/lib/python3.9/dist-packages (from google-
auth<3.0dev,>=1.25.0->google-api-core[grpc]<2.0.0dev,>=1.14.0->cirq-
google==1.1.0->cirq) (0.2.8)
Requirement already satisfied: zipp>=3.1.0 in
/usr/local/lib/python3.9/dist-packages (from importlib-
resources>=3.2.0->matplotlib~=3.0->cirq-core==1.1.0->cirq) (3.15.0)
Requirement already satisfied: toml<0.11.0,>=0.10.2 in
/usr/local/lib/python3.9/dist-packages (from qcs-api-
client<0.22.0,>=0.21.0->pyquil>=3.2.0->cirq-rigetti==1.1.0->cirq)
(0.10.2)
Collecting PyJWT<3.0.0,>=2.4.0
  Downloading PyJWT-2.6.0-py3-none-any.whl (20 kB)
Collecting attrs<22.0.0,>=21.3.0
  Downloading attrs-21.4.0-py2.py3-none-any.whl (60 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 60.6/60.6 KB 8.2 MB/s eta
0:00:00
ent already satisfied: pydantic<2.0.0,>=1.7.2 in
/usr/local/lib/python3.9/dist-packages (from qcs-api-
client<0.22.0,>=0.21.0->pyquil>=3.2.0->cirq-rigetti==1.1.0->cirq)
(1.10.7)
Collecting rfc3339<7.0,>=6.2
  Downloading rfc3339-6.2-py3-none-any.whl (5.5 kB)
Collecting httpx<0.24.0,>=0.23.0
  Downloading httpx-0.23.3-py3-none-any.whl (71 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 71.5/71.5 KB 10.7 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 98.7/98.7 KB 11.0 MB/s eta
0:00:00
ent already satisfied: decorator>=3.4.2 in
/usr/local/lib/python3.9/dist-packages (from retry<0.10.0,>=0.9.2-
>pyquil>=3.2.0->cirq-rigetti==1.1.0->cirq) (4.4.2)
Requirement already satisfied: msgpack<2.0,>=0.6 in
/usr/local/lib/python3.9/dist-packages (from rpcq<4.0.0,>=3.10.0-
>pyquil>=3.2.0->cirq-rigetti==1.1.0->cirq) (1.0.5)
Collecting python-rapidjson
  Downloading python_rapidjson-1.10-cp39-cp39-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl (1.7 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.7/1.7 MB 68.0 MB/s eta
0:00:00
ent already satisfied: pyzmq>=17 in /usr/local/lib/python3.9/dist-
packages (from rpcq<4.0.0,>=3.10.0->pyquil>=3.2.0->cirq-
rigetti==1.1.0->cirq) (23.2.1)
Collecting ruamel.yaml
  Downloading ruamel.yaml-0.17.21-py3-none-any.whl (109 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 109.5/109.5 KB 11.3 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 69.6/69.6 KB 8.8 MB/s eta
0:00:00
```

```
ent already satisfied: pyasn1<0.5.0,>=0.4.6 in
/usr/local/lib/python3.9/dist-packages (from pyasn1-modules>=0.2.1-
>google-auth<3.0dev,>=1.25.0->google-api-core[grpc]<2.0.0dev,>=1.14.0-
>cirq-google==1.1.0->cirq) (0.4.8)
Collecting ruamel.yaml.clib>=0.2.6
  Downloading ruamel.yaml.clib-0.2.7-cp39-cp39-
manylinux_2_17_x86_64.manylinux2014_x86_64.manylinux_2_24_x86_64.whl
(519 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 519.4/519.4 KB 38.8 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 58.3/58.3 KB 5.4 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 80.6/80.6 KB 7.3 MB/s eta
0:00:00
e=lark-0.11.3-py2.py3-none-any.whl size=99646
sha256=3fbdf5a3c50fb279ede331680bb10c13be520ad53fc8a3f5d4fa94c64566a4c
3
  Stored in directory:
/root/.cache/pip/wheels/ec/6a/24/f8eeaf52fee56bfe54309621b59c41bb7f1df
56f4bfbcdb0ce
  Building wheel for rpcq (setup.py) ... e=rpcq-3.11.0-py3-none-
any.whl size=45985
sha256=809c584a2b129cd3ca635d99b95ecbbfe80158d7c9f044bfb38492ef836c513
4
  Stored in directory:
/root/.cache/pip/wheels/a6/c4/42/34581dfe489802146924ad802b13aa7fe3820
f9e8c15f67afc
Successfully built lark rpcq
Installing collected packages: types-retry, types-python-dateutil,
rfc3986, rfc3339, lark, sniffio, ruamel.yaml.clib, retrying, python-
rapidjson, PyJWT, py, networkx, iso8601, h11, duet, attrs,
ruamel.yaml, retry, anyio, rpcq, httpcore, google-api-core, cirq-core,
httpx, cirq-web, cirq-pasqal, cirq-ionq, cirq-aqt, qcs-api-client,
cirq-google, pyquil, cirq-rigetti, cirq
  Attempting uninstall: networkx
    Found existing installation: networkx 3.0
    Uninstalling networkx-3.0:
      Successfully uninstalled networkx-3.0
  Attempting uninstall: attrs
    Found existing installation: attrs 22.2.0
    Uninstalling attrs-22.2.0:
      Successfully uninstalled attrs-22.2.0
  Attempting uninstall: google-api-core
    Found existing installation: google-api-core 2.11.0
    Uninstalling google-api-core-2.11.0:
      Successfully uninstalled google-api-core-2.11.0
Successfully installed PyJWT-2.6.0 anyio-3.6.2 attrs-21.4.0 cirq-1.1.0
cirq-aqt-1.1.0 cirq-core-1.1.0 cirq-google-1.1.0 cirq-ionq-1.1.0 cirq-
pasqal-1.1.0 cirq-rigetti-1.1.0 cirq-web-1.1.0 duet-0.2.7 google-api-
core-1.34.0 h11-0.14.0 httpcore-0.16.3 httpx-0.23.3 iso8601-1.1.0
```

```
lark-0.11.3 networkx-2.8.8 py-1.11.0 pyquil-3.3.4 python-rapidjson-
1.10 qcs-api-client-0.21.3 retry-0.9.2 retrying-1.3.4 rfc3339-6.2
rfc3986-1.5.0 rpcq-3.11.0 ruamel.yaml-0.17.21 ruamel.yaml.clib-0.2.7
sniffio-1.3.0 types-python-dateutil-2.8.19.11 types-retry-0.9.9.3
```

{"pip_warning":{"packages":["google"]}}

```
Looking in indexes: https://pypi.org/simple, https://us-
python.pkg.dev/colab-wheels/public/simple/
Collecting qiskit
  Downloading qiskit-0.42.1.tar.gz (14 kB)
  Preparing metadata (setup.py) ...
anylinux_2_17_x86_64.manylinux2014_x86_64.whl (5.1 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 5.1/5.1 MB 33.8 MB/s eta
0:00:00
anylinux_2_17_x86_64.manylinux2014_x86_64.whl (12.8 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 12.8/12.8 MB 46.0 MB/s eta
0:00:00
q-provider==0.20.2
  Downloading qiskit_ibmq_provider-0.20.2-py3-none-any.whl (241 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 241.5/241.5 KB 23.2 MB/s eta
0:00:00
ent already satisfied: scipy>=1.0 in /usr/local/lib/python3.9/dist-
packages (from qiskit-aer==0.12.0->qiskit) (1.10.1)
Requirement already satisfied: numpy>=1.16.3 in
/usr/local/lib/python3.9/dist-packages (from qiskit-aer==0.12.0-
>qiskit) (1.22.4)
Requirement already satisfied: python-dateutil>=2.8.0 in
/usr/local/lib/python3.9/dist-packages (from qiskit-ibmq-
provider==0.20.2->qiskit) (2.8.2)
Requirement already satisfied: requests>=2.19 in
/usr/local/lib/python3.9/dist-packages (from qiskit-ibmq-
provider==0.20.2->qiskit) (2.27.1)
Collecting websockets>=10.0
  Downloading websockets-11.0-cp39-cp39-
manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_17_x86_64.manylinux
2014_x86_64.whl (129 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 129.5/129.5 KB 6.6 MB/s eta
0:00:00
ent already satisfied: urllib3>=1.21.1 in
/usr/local/lib/python3.9/dist-packages (from qiskit-ibmq-
provider==0.20.2->qiskit) (1.26.15)
Collecting websocket-client>=1.5.1
  Downloading websocket_client-1.5.1-py3-none-any.whl (55 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 55.9/55.9 KB 6.4 MB/s eta
0:00:00
<=1.1.0
  Downloading requests_ntlm-1.1.0-py2.py3-none-any.whl (5.7 kB)
Collecting symengine>=0.9
  Downloading symengine-0.10.0-cp39-cp39-
manylinux_2_12_x86_64.manylinux2010_x86_64.whl (37.5 MB)
```

```
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 37.5/37.5 MB 12.0 MB/s eta
0:00:00
anylinux_2_17_x86_64.manylinux2014_x86_64.whl (1.9 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.9/1.9 MB 13.7 MB/s eta
0:00:00
ent already satisfied: sympy>=1.3 in /usr/local/lib/python3.9/dist-
packages (from qiskit-terra==0.23.3->qiskit) (1.11.1)
Collecting dill>=0.3
  Downloading dill-0.3.6-py3-none-any.whl (110 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 110.5/110.5 KB 7.7 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 49.6/49.6 KB 5.4 MB/s eta
0:00:00
ent already satisfied: psutil>=5 in /usr/local/lib/python3.9/dist-
packages (from qiskit-terra==0.23.3->qiskit) (5.9.4)
Collecting ply>=3.10
  Downloading ply-3.11-py2.py3-none-any.whl (49 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 49.6/49.6 KB 2.1 MB/s eta
0:00:00
ent already satisfied: six>=1.5 in /usr/local/lib/python3.9/dist-
packages (from python-dateutil>=2.8.0->qiskit-ibmq-provider==0.20.2-
>qiskit) (1.16.0)
Requirement already satisfied: charset-normalizer~=2.0.0 in
/usr/local/lib/python3.9/dist-packages (from requests>=2.19->qiskit-
ibmq-provider==0.20.2->qiskit) (2.0.12)
Requirement already satisfied: idna<4,>=2.5 in
/usr/local/lib/python3.9/dist-packages (from requests>=2.19->qiskit-
ibmq-provider==0.20.2->qiskit) (3.4)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.9/dist-packages (from requests>=2.19->qiskit-
ibmq-provider==0.20.2->qiskit) (2022.12.7)
Requirement already satisfied: cryptography>=1.3 in
/usr/local/lib/python3.9/dist-packages (from requests-ntlm<=1.1.0-
>qiskit-ibmq-provider==0.20.2->qiskit) (40.0.1)
Collecting ntlm-auth>=1.0.2
  Downloading ntlm_auth-1.5.0-py2.py3-none-any.whl (29 kB)
Collecting pbr!=2.1.0,>=2.0.0
  Downloading pbr-5.11.1-py2.py3-none-any.whl (112 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 112.7/112.7 KB 14.1 MB/s eta
0:00:00
ent already satisfied: mpmath>=0.19 in /usr/local/lib/python3.9/dist-
packages (from sympy>=1.3->qiskit-terra==0.23.3->qiskit) (1.3.0)
Requirement already satisfied: cffi>=1.12 in
/usr/local/lib/python3.9/dist-packages (from cryptography>=1.3-
>requests-ntlm<=1.1.0->qiskit-ibmq-provider==0.20.2->qiskit) (1.15.1)
Requirement already satisfied: pycparser in
/usr/local/lib/python3.9/dist-packages (from cffi>=1.12-
>cryptography>=1.3->requests-ntlm<=1.1.0->qiskit-ibmq-
provider==0.20.2->qiskit) (2.21)
Building wheels for collected packages: qiskit
```

```
  Building wheel for qiskit (setup.py) ... e=qiskit-0.42.1-py3-none-
any.whl size=12938
sha256=727022bb6132caec7f0462c7106119f07cf8d8628e4794f8476eae0d7ada4b2
0
  Stored in directory:
/root/.cache/pip/wheels/40/64/74/29c046bda04fd60f3f6b2e244fa85b70f219e
363fc3373f541
Successfully built qiskit
Installing collected packages: ply, websockets, websocket-client,
symengine, rustworkx, pbr, ntlm-auth, dill, stevedore, requests-ntlm,
qiskit-terra, qiskit-ibmq-provider, qiskit-aer, qiskit
Successfully installed dill-0.3.6 ntlm-auth-1.5.0 pbr-5.11.1 ply-3.11
qiskit-0.42.1 qiskit-aer-0.12.0 qiskit-ibmq-provider-0.20.2 qiskit-
terra-0.23.3 requests-ntlm-1.1.0 rustworkx-0.12.1 stevedore-5.0.0
symengine-0.10.0 websocket-client-1.5.1 websockets-11.0
Looking in indexes: https://pypi.org/simple, https://us-
python.pkg.dev/colab-wheels/public/simple/
Collecting pylatexenc
  Downloading pylatexenc-2.10.tar.gz (162 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 162.6/162.6 KB 5.0 MB/s eta
0:00:00
etadata (setup.py) ... e=pylatexenc-2.10-py3-none-any.whl size=136831
sha256=00a6990882fd4dfb2054e571f5bf5d8203f74229e3a0b4068c07f6a2893a8a7
c
  Stored in directory:
/root/.cache/pip/wheels/a3/68/66/2f15abd0673d83c02f354115feedeb89c3dae
d2ac319b11090
Successfully built pylatexenc
Installing collected packages: pylatexenc
Successfully installed pylatexenc-2.10
Looking in indexes: https://pypi.org/simple, https://us-
python.pkg.dev/colab-wheels/public/simple/
Collecting pennylane
  Downloading PennyLane-0.29.1-py3-none-any.whl (1.3 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.3/1.3 MB 17.2 MB/s eta
0:00:00
ent already satisfied: networkx in /usr/local/lib/python3.9/dist-
packages (from pennylane) (2.8.8)
Collecting semantic-version>=2.7
  Downloading semantic_version-2.10.0-py2.py3-none-any.whl (15 kB)
Requirement already satisfied: appdirs in
/usr/local/lib/python3.9/dist-packages (from pennylane) (1.4.4)
Requirement already satisfied: toml in /usr/local/lib/python3.9/dist-
packages (from pennylane) (0.10.2)
Collecting autoray>=0.3.1
  Downloading autoray-0.6.3-py3-none-any.whl (48 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 48.3/48.3 KB 6.3 MB/s eta
0:00:00
anylinux_2_17_x86_64.manylinux2014_x86_64.whl (16.5 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 16.5/16.5 MB 60.2 MB/s eta
```

```
0:00:00
ent already satisfied: autograd in /usr/local/lib/python3.9/dist-
packages (from pennylane) (1.5)
Requirement already satisfied: cachetools in
/usr/local/lib/python3.9/dist-packages (from pennylane) (5.3.0)
Requirement already satisfied: scipy in /usr/local/lib/python3.9/dist-
packages (from pennylane) (1.10.1)
Requirement already satisfied: numpy<1.24 in
/usr/local/lib/python3.9/dist-packages (from pennylane) (1.22.4)
Requirement already satisfied: requests in
/usr/local/lib/python3.9/dist-packages (from pennylane) (2.27.1)
Requirement already satisfied: future>=0.15.2 in
/usr/local/lib/python3.9/dist-packages (from autograd->pennylane)
(0.18.3)
Requirement already satisfied: charset-normalizer~=2.0.0 in
/usr/local/lib/python3.9/dist-packages (from requests->pennylane)
(2.0.12)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in
/usr/local/lib/python3.9/dist-packages (from requests->pennylane)
(1.26.15)
Requirement already satisfied: idna<4,>=2.5 in
/usr/local/lib/python3.9/dist-packages (from requests->pennylane)
(3.4)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.9/dist-packages (from requests->pennylane)
(2022.12.7)
Requirement already satisfied: rustworkx==0.12.1 in
/usr/local/lib/python3.9/dist-packages (from retworkx->pennylane)
(0.12.1)
Installing collected packages: semantic-version, autoray, retworkx,
pennylane-lightning, pennylane
Successfully installed autoray-0.6.3 pennylane-0.29.1 pennylane-
lightning-0.29.0 retworkx-0.12.1 semantic-version-2.10.0
Looking in indexes: https://pypi.org/simple, https://us-
python.pkg.dev/colab-wheels/public/simple/
Collecting tensorflow-addons
  Downloading tensorflow_addons-0.19.0-cp39-cp39-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl (1.1 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.1/1.1 MB 14.8 MB/s eta
0:00:00
ent already satisfied: packaging in /usr/local/lib/python3.9/dist-
packages (from tensorflow-addons) (23.0)
Requirement already satisfied: typing-extensions>=4.4.0 in
/usr/local/lib/python3.9/dist-packages (from typeguard>=2.7-
>tensorflow-addons) (4.5.0)
Requirement already satisfied: importlib-metadata>=3.6 in
/usr/local/lib/python3.9/dist-packages (from typeguard>=2.7-
>tensorflow-addons) (6.1.0)
Requirement already satisfied: zipp>=0.5 in
/usr/local/lib/python3.9/dist-packages (from importlib-metadata>=3.6-
```

```
>typeguard>=2.7->tensorflow-addons) (3.15.0)
Installing collected packages: typeguard, tensorflow-addons
Successfully installed tensorflow-addons-0.19.0 typeguard-3.0.2
```

##Task VIII: Vision transformer/Quantum Vision Transformer

Implement a classical Vision transformer and apply it to MNIST. Show its performance on the test data. Comment on potential ideas to extend this classical vision transformer architecture to a quantum vision transformer and sketch out the architecture in detail.

###Vision Transformer

In this notebook, I will build a vision transformer model to classify images from the MNIST dataset. The MNIST dataset contains images of handwritten digits, and I will resize these images to 72x72 and extract patches of size 6x6. The vision transformer model will consist of multiple transformer layers, each of which will have multi-head self-attention and feedforward neural network units. The model will have a final dense layer for classification.

I will use TensorFlow and TensorFlow Addons libraries to build the model. I will also define some hyperparameters such as learning rate, weight decay, batch size, number of epochs, projection dimension, number of transformer layers, size of the dense layers, etc.

```
!pip install -U tensorflow-addons
```

```
Looking in indexes: https://pypi.org/simple, https://us-
python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: tensorflow-addons in
/usr/local/lib/python3.9/dist-packages (0.19.0)
Requirement already satisfied: packaging in
/usr/local/lib/python3.9/dist-packages (from tensorflow-addons) (23.0)
Requirement already satisfied: typeguard>=2.7 in
/usr/local/lib/python3.9/dist-packages (from tensorflow-addons)
(3.0.2)
Requirement already satisfied: importlib-metadata>=3.6 in
/usr/local/lib/python3.9/dist-packages (from typeguard>=2.7-
>tensorflow-addons) (6.1.0)
Requirement already satisfied: typing-extensions>=4.4.0 in
/usr/local/lib/python3.9/dist-packages (from typeguard>=2.7-
>tensorflow-addons) (4.5.0)
Requirement already satisfied: zipp>=0.5 in
/usr/local/lib/python3.9/dist-packages (from importlib-metadata>=3.6-
>typeguard>=2.7->tensorflow-addons) (3.15.0)
```

```
import numpy as np
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
import tensorflow_addons as tfa
```

```
/usr/local/lib/python3.9/dist-packages/tensorflow_addons/utils/
ensure_tf_install.py:53: UserWarning: Tensorflow Addons supports using
```

```
Python ops for all Tensorflow versions above or equal to 2.9.0 and
strictly below 2.12.0 (nightly versions are not supported).
 The versions of TensorFlow you are currently using is 2.12.0 and is
not supported.
Some things might work, some things might not.
If you were to encounter a bug, do not file an issue.
If you want to make sure you're using a tested and supported
configuration, either change the TensorFlow version or the TensorFlow
Addons's version.
You can find the compatibility matrix in TensorFlow Addon's readme:
https://github.com/tensorflow/addons
  warnings.warn(
```

I will start by preparing the data by loading the MNIST dataset and printing its shapes. Then, I will configure the hyperparameters for the model. Finally, I will build the vision transformer model using TensorFlow and train it on the MNIST dataset.

```
num_classes = 100
input_shape = (28, 28, 1)

(x_train, y_train), (x_test, y_test) =
keras.datasets.mnist.load_data()

x_train = x_train.reshape(-1, 28, 28, 1)
x_test = x_test.reshape(-1, 28, 28, 1)

print(f"x_train shape: {x_train.shape} - y_train shape:
{y_train.shape}")
print(f"x_test shape: {x_test.shape} - y_test shape: {y_test.shape}")
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-
keras-datasets/mnist.npz
11490434/11490434 [==============================] - 0s 0us/step
x_train shape: (60000, 28, 28, 1) - y_train shape: (60000,)
x_test shape: (10000, 28, 28, 1) - y_test shape: (10000,)
```

```
learning_rate = 0.001
weight_decay = 0.0001
batch_size = 256
num_epochs = 10
image_size = 28  # resizing input images to this size
patch_size = 4  # Size of the patches to be extract from the input
images
num_patches = (image_size // patch_size) ** 2
projection_dim = 64
num_heads = 4
transformer_units = [
    projection_dim * 2,
    projection_dim,
]  # Size of the transformer layers
```

```
transformer_layers = 8
mlp_head_units = [2048, 1024]  # Size of the dense layers of the final
classifier

data_augmentation = keras.Sequential(
    [
        layers.Normalization(),
        layers.Resizing(image_size, image_size),
        layers.RandomFlip("horizontal"),
        layers.RandomRotation(factor=0.02),
        layers.RandomZoom(
            height_factor=0.2, width_factor=0.2
        ),
    ],
    name="data_augmentation",
)
# Compute the mean and the variance of the training data for
normalization.
data_augmentation.layers[0].adapt(x_train)

def mlp(x, hidden_units, dropout_rate):
    """
    Builds a multi-layer perceptron (MLP) model.

    Args:
    x (tensorflow.Tensor): Input tensor.
    hidden_units (list[int]): A list of integers specifying the number
of hidden units for each layer.
    dropout_rate (float): A float value specifying the dropout rate
for each layer.

    Returns:
    tensorflow.Tensor: Output tensor of the MLP model.

    The function constructs a feedforward neural network with multiple
layers, where each layer is a fully connected layer
    followed by a dropout layer to prevent overfitting. The activation
function used in each hidden layer is the GeLU
    activation function.
    """
    for units in hidden_units:
        x = layers.Dense(units, activation=tf.nn.gelu)(x)
        x = layers.Dropout(dropout_rate)(x)
    return x
```

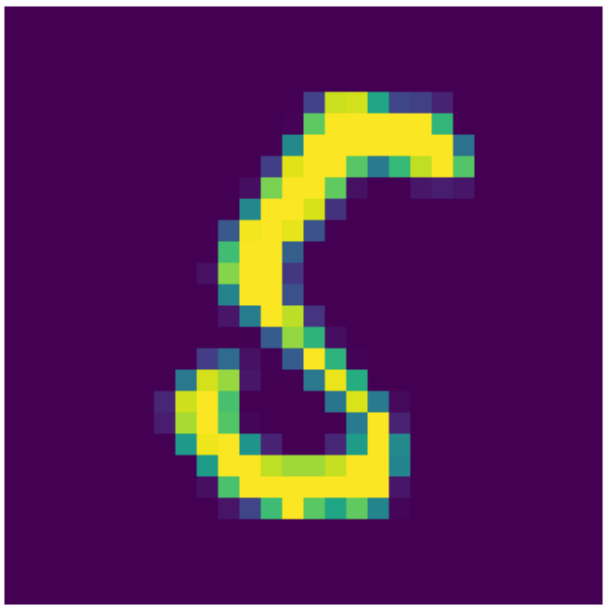
Creating patches for form the images

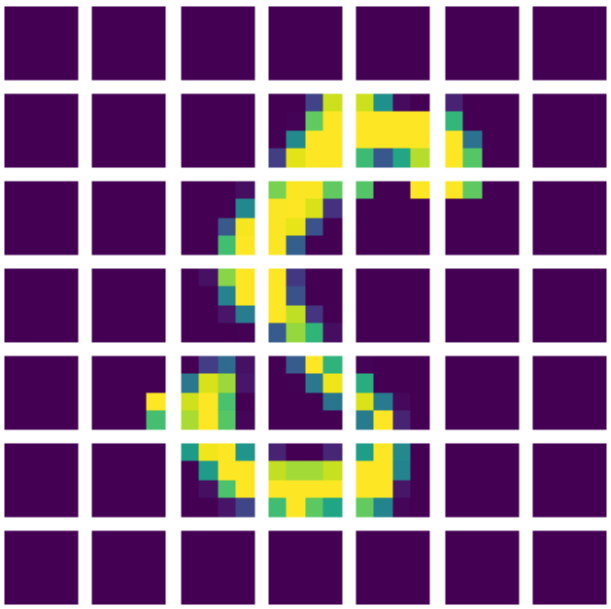
```
class Patches(layers.Layer):
    """
    A layer that extracts patches from input images.
```

```
    This layer takes an input tensor of images and extracts patches of
a specified
    size from each image. The output tensor is a batch of flattened
patches.

    Args:
        patch_size: An integer representing the size of each patch.

    Returns:
        A tensor representing the batch of flattened patches extracted
from the input images.
    """

    def __init__(self, patch_size):
        """
        Initializes the Patches layer.

        Args:
            patch_size: An integer representing the size of each
patch.
        """
        super().__init__()
        self.patch_size = patch_size

    def call(self, images):
        """
        Computes the output of the Patches layer.

        Args:
            images: A tensor representing a batch of input images.

        Returns:
            A tensor representing the batch of flattened patches
extracted from the input images.
        """
        batch_size = tf.shape(images)[0]
        patches = tf.image.extract_patches(
            images=images,
            sizes=[1, self.patch_size, self.patch_size, 1],
            strides=[1, self.patch_size, self.patch_size, 1],
            rates=[1, 1, 1, 1],
            padding="VALID",
        )
        patch_dims = patches.shape[-1]
        patches = tf.reshape(patches, [batch_size, -1, patch_dims])
        return patches

import matplotlib.pyplot as plt

plt.figure(figsize=(4, 4))
```

```
image = x_train[np.random.choice(range(x_train.shape[0]))]
plt.imshow(image.astype("uint8"))
plt.axis("off")

resized_image = tf.convert_to_tensor([image])
patches = Patches(patch_size)(resized_image)
print(f"Image size: {image_size} X {image_size}")
print(f"Patch size: {patch_size} X {patch_size}")
print(f"Patches per image: {patches.shape[1]}")
print(f"Elements per patch: {patches.shape[-1]}")

n = int(np.sqrt(patches.shape[1]))
plt.figure(figsize=(4, 4))
for i, patch in enumerate(patches[0]):
    ax = plt.subplot(n, n, i + 1)
    patch_img = tf.reshape(patch, (patch_size, patch_size, 1))
    plt.imshow(patch_img.numpy().astype("uint8"))
    plt.axis("off")
```

```
Image size: 28 X 28
Patch size: 4 X 4
Patches per image: 49
Elements per patch: 16
```





```
class PatchEncoder(layers.Layer):
    """
    Layer that encodes a sequence of image patches using a dense
projection and
    position embeddings.

    Args:
        num_patches (int): The number of patches to be encoded in the
sequence.
        projection_dim (int): The number of units in the dense
projection layer.

    Attributes:
        num_patches (int): The number of patches in the sequence.
        projection (Dense layer): A dense layer that maps each patch
to a higher
            dimensional space.
        position_embedding (Embedding layer): An embedding layer that
maps each
            patch position to a learned embedding vector of the same
dimension as
            the projection output.

    Methods:
        call(patch): Encodes a sequence of patches given as input, by
applying the
            projection layer and adding the corresponding position
embeddings.

    Returns:
```

```
        A tensor of shape (batch_size, num_patches, projection_dim),
representing
        the encoded sequence of image patches.
    """
    def __init__(self, num_patches, projection_dim):
        super().__init__()
        self.num_patches = num_patches
        self.projection = layers.Dense(units=projection_dim)
        self.position_embedding = layers.Embedding(
            input_dim=num_patches, output_dim=projection_dim
        )

    def call(self, patch):
        """
        Encodes a sequence of patches given as input, by applying the
projection layer
        and adding the corresponding position embeddings.

        Args:
            patch (tensor): A tensor of shape (batch_size,
num_patches, patch_dim),
                representing the input sequence of image patches.

        Returns:
            A tensor of shape (batch_size, num_patches,
projection_dim), representing
            the encoded sequence of image patches.
        """
        positions = tf.range(start=0, limit=self.num_patches, delta=1)
        encoded = self.projection(patch) +
self.position_embedding(positions)
        return encoded

def create_vit_classifier(input_shape=(72, 72, 3), num_classes=10,
patch_size=16, projection_dim=64,
                          num_heads=4, transformer_layers=8,
transformer_units=256, mlp_head_units=256):
    """
    Creates a Vision Transformer (ViT) model for image classification.

    Args:
        input_shape (tuple): shape of input image, e.g., (224, 224,
3).
        num_classes (int): number of output classes.
        patch_size (int): size of patches to extract from the input
image.
        projection_dim (int): dimension of the projected feature
space.
        num_heads (int): number of heads in the multi-head attention
layer.
```

```
        transformer_layers (int): number of Transformer blocks to
stack.
        transformer_units (int): number of units in the Transformer
block MLP.
        mlp_head_units (int): number of units in the MLP used for
classification.

    Returns:
        A Keras model for ViT image classification.

    """
    # Define the input layer.
    inputs = layers.Input(shape=input_shape)

    # Data augmentation.
    augmented_inputs = data_augmentation(inputs)

    # Extract patches from the augmented input.
    patches = Patches(patch_size)(augmented_inputs)

    # Encode the patches.
    encoded_patches = PatchEncoder(num_patches, projection_dim)
(patches)

    # Stack multiple Transformer blocks.
    for _ in range(transformer_layers):
        # Layer normalization before the multi-head attention layer.
        x1 = layers.LayerNormalization(epsilon=1e-6)(encoded_patches)

        # Multi-head attention layer.
        attention_output = layers.MultiHeadAttention(
            num_heads=num_heads, key_dim=projection_dim, dropout=0.1
        )(x1, x1)

        # Add skip connection.
        x2 = layers.Add()([attention_output, encoded_patches])

        # Layer normalization before the MLP.
        x3 = layers.LayerNormalization(epsilon=1e-6)(x2)

        # MLP.
        x3 = mlp(x3, hidden_units=transformer_units, dropout_rate=0.1)

        # Add skip connection.
        encoded_patches = layers.Add()([x3, x2])

    # Layer normalization after the Transformer blocks.
    representation = layers.LayerNormalization(epsilon=1e-6)
(encoded_patches)
```

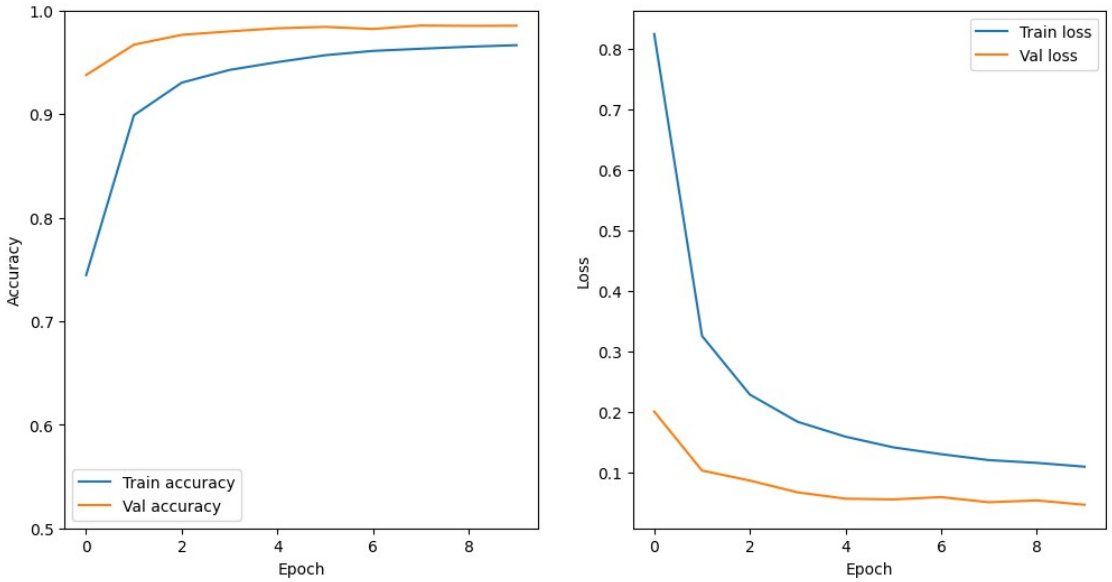
```
    # Flatten the feature tensor.
    representation = layers.Flatten()(representation)

    # Add dropout.
    representation = layers.Dropout(0.5)(representation)

    # Add MLP for classification.
    features = mlp(representation, hidden_units=mlp_head_units,
dropout_rate=0.5)

    # Add output layer.
    logits = layers.Dense(num_classes)(features)

    # Create the Keras model.
    model = keras.Model(inputs=inputs, outputs=logits)

    return model

import matplotlib.pyplot as plt

def plot_history(history):
    """
    Plots the training and validation accuracy and loss history of a
neural network.

    Parameters:
    history: A keras History object containing the training history of
a model.

    Returns:
    None

    """
    fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 6))

    ax1.plot(history.history["accuracy"], label="Train accuracy")
    ax1.plot(history.history["val_accuracy"], label="Val accuracy")
    ax1.set_xlabel("Epoch")
    ax1.set_ylabel("Accuracy")
    ax1.set_ylim([0.5, 1])
    ax1.legend()

    ax2.plot(history.history["loss"], label="Train loss")
    ax2.plot(history.history["val_loss"], label="Val loss")
    ax2.set_xlabel("Epoch")
    ax2.set_ylabel("Loss")
    ax2.legend()
```

```
    plt.show()


def run_experiment(model, x_train, y_train, x_test, y_test,
learning_rate, weight_decay, batch_size, num_epochs):
    """
    Trains a model on the given data and evaluates its performance on
a test set.

    Args:
        model: A Keras model to be trained.
        x_train: The input training data.
        y_train: The target training data.
        x_test: The input test data.
        y_test: The target test data.
        learning_rate: The learning rate for the optimizer.
        weight_decay: The weight decay for the optimizer.
        batch_size: The batch size for training.
        num_epochs: The number of epochs to train for.

    Returns:
        The training history of the model.
    """

    # Define the optimizer
    optimizer = tfa.optimizers.AdamW(
        learning_rate=learning_rate,
        weight_decay=weight_decay
    )

    # Compile the model with the optimizer and loss function
    model.compile(
        optimizer=optimizer,

loss=keras.losses.SparseCategoricalCrossentropy(from_logits=True),
        metrics=[
            keras.metrics.SparseCategoricalAccuracy(name="accuracy"),
            keras.metrics.SparseTopKCategoricalAccuracy(5,
name="top_5_accuracy"),
        ],
    )

    # Define a callback to save the best weights during training
    checkpoint_filepath = "/tmp/checkpoint"
    checkpoint_callback = keras.callbacks.ModelCheckpoint(
        checkpoint_filepath,
        monitor="val_accuracy",
        save_best_only=True,
        save_weights_only=True,
    )
```

```
    # Train the model and save the best weights
    history = model.fit(
        x=x_train,
        y=y_train,
        batch_size=batch_size,
        epochs=num_epochs,
        validation_split=0.1,
        callbacks=[checkpoint_callback],
    )
    model.load_weights(checkpoint_filepath)

    # Evaluate the model on the test set
    _, accuracy, top_5_accuracy = model.evaluate(x_test, y_test)
    print(f"Test accuracy: {round(accuracy * 100, 2)}%")
    print(f"Test top 5 accuracy: {round(top_5_accuracy * 100, 2)}%")

    # Return the training history
    return history



vit_classifier = create_vit_classifier()
history = run_experiment(vit_classifier, x_train, y_train, x_test,
y_test,learning_rate, weight_decay, batch_size, num_epochs)
plot_history(history)
```

```
Epoch 1/10
211/211 [==============================] - 988s 5s/step - loss: 0.8240
- accuracy: 0.7444 - top-5-accuracy: 0.9656 - val_loss: 0.2007 -
val_accuracy: 0.9377 - val_top-5-accuracy: 0.9985
Epoch 2/10
211/211 [==============================] - 929s 4s/step - loss: 0.3255
- accuracy: 0.8989 - top-5-accuracy: 0.9951 - val_loss: 0.1033 -
val_accuracy: 0.9670 - val_top-5-accuracy: 0.9988
Epoch 3/10
211/211 [==============================] - 1068s 5s/step - loss:
0.2287 - accuracy: 0.9303 - top-5-accuracy: 0.9974 - val_loss: 0.0868
- val_accuracy: 0.9765 - val_top-5-accuracy: 0.9988
Epoch 4/10
211/211 [==============================] - 1047s 5s/step - loss:
0.1837 - accuracy: 0.9426 - top-5-accuracy: 0.9981 - val_loss: 0.0673
- val_accuracy: 0.9798 - val_top-5-accuracy: 0.9987
Epoch 5/10
211/211 [==============================] - 960s 5s/step - loss: 0.1592
- accuracy: 0.9502 - top-5-accuracy: 0.9987 - val_loss: 0.0568 -
val_accuracy: 0.9828 - val_top-5-accuracy: 0.9993
Epoch 6/10
211/211 [==============================] - 943s 4s/step - loss: 0.1416
- accuracy: 0.9568 - top-5-accuracy: 0.9987 - val_loss: 0.0556 -
```

```
val_accuracy: 0.9842 - val_top-5-accuracy: 0.9997
Epoch 7/10
211/211 [==============================] - 927s 4s/step - loss: 0.1303
- accuracy: 0.9610 - top-5-accuracy: 0.9990 - val_loss: 0.0594 -
val_accuracy: 0.9822 - val_top-5-accuracy: 0.9993
Epoch 8/10
211/211 [==============================] - 958s 5s/step - loss: 0.1205
- accuracy: 0.9631 - top-5-accuracy: 0.9993 - val_loss: 0.0510 -
val_accuracy: 0.9855 - val_top-5-accuracy: 0.9995
Epoch 9/10
211/211 [==============================] - 970s 5s/step - loss: 0.1159
- accuracy: 0.9650 - top-5-accuracy: 0.9992 - val_loss: 0.0540 -
val_accuracy: 0.9852 - val_top-5-accuracy: 0.9990
Epoch 10/10
211/211 [==============================] - 968s 5s/step - loss: 0.1097
- accuracy: 0.9665 - top-5-accuracy: 0.9993 - val_loss: 0.0468 -
val_accuracy: 0.9853 - val_top-5-accuracy: 0.9992
313/313 [==============================] - 56s 180ms/step - loss:
0.0496 - accuracy: 0.9841 - top-5-accuracy: 0.9995
Test accuracy: 98.41%
Test top 5 accuracy: 99.95%
```



###Quantum Vision TransFormer

Extending the classical vision transformer architecture to a quantum vision transformer requires adapting the architecture to operate on quantum data and exploit the advantages of quantum computing. In general, quantum computing has the potential to perform certain computations exponentially faster than classical computing, and it is expected to offer significant advantages for image recognition tasks.

Here are some potential ideas to extend the classical vision transformer architecture to a quantum vision transformer:

1. **Quantum Encoding:** Instead of encoding the input image as a sequence of vectors as in classical transformers, the image can be encoded as a quantum state. One approach could be to use a quantum circuit to encode the image as a quantum state, and then use quantum gates to perform operations on the state to extract features. The output can then be measured to obtain a classical result.

2. **Quantum Attention:** Quantum attention can be used to replace the classical attention mechanism used in the transformer architecture. In a quantum attention mechanism, the key, query, and value vectors are represented as quantum states, and quantum gates are used to perform the attention operation. This approach has the potential to provide exponential speedup over classical attention mechanisms.

3. **Quantum Filters:** Quantum filters can be used to perform convolutional operations on the input image. A quantum filter consists of a quantum circuit that acts as a filter on the quantum state representing the image. This approach has the potential to provide exponential speedup over classical convolutional filters.

4. **Quantum Pooling:** Quantum pooling can be used to downsample the quantum state representing the image. Quantum pooling can be achieved using quantum circuits that perform measurements on the quantum state and discard certain components to reduce the dimensionality of the quantum state.