Name : Devdeep Shetranjiwala
Email ID : devdeep0702@gmail.com

Specific Task : Vision Transformers

Description:

> **Train a Transformer model of your choice on the dataset below to achieve the performance closest to your CNN model's performance in Task 1. </br> Discuss the resulting performance of the 2 chosen architectures.**
>
> ## (The discussion is done at the end of the code)

Datasets (Same as in Task 1): </br> https://cernbox.cern.ch/index.php/s/AtBT8y4MiQYFcgc **(Photons) </br>** https://cernbox.cern.ch/index.php/s/FbXw3V4XNyYB3oA **(Electrons)**

In [ ]:

```
import numpy as np
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
import h5py
from sklearn.model_selection import train_test_split
```

In [ ]:

```
print("Num of GPUs Available: ", len(tf.config.list_physical_devices('GPU')))
```

```
Num of GPUs Available:  1
```

In [ ]:

```
import requests
url = 'https://cernbox.cern.ch/remote.php/dav/public-files/AtBT8y4MiQYFcgc/SinglePhotonPt
50_IMGCROPS_n249k_RHv1.hdf5'
r = requests.get(url, allow_redirects=True)
open('photons.hdf5', 'wb').write(r.content)
url = 'https://cernbox.cern.ch/remote.php/dav/public-files/FbXw3V4XNyYB3oA/SingleElectron
Pt50_IMGCROPS_n249k_RHv1.hdf5'
r = requests.get(url, allow_redirects=True)
open('electrons.hdf5', 'wb').write(r.content)
```

Out[ ]:

```
128927319
```

In [ ]:

```
file_electron = "electrons.hdf5"
file_photon = "photons.hdf5"

with h5py.File(file_electron, "r") as f1:
    X_elec = np.array(f1['X'][:])
    y_elec = np.array(f1['y'][:])
with h5py.File(file_photon, "r") as f2:
    X_phot = np.array(f2['X'][:])
    y_phot = np.array(f2['y'][:])
print(X_elec.shape)
print(X_phot.shape)
del(file_electron,file_photon)
```

```
(249000, 32, 32, 2)
(249000, 32, 32, 2)
```

In [ ]:

```
num_classes = 2
input_shape = (32, 32, 2)
X = np.append(X_elec, X_phot, axis=0)
y = np.append(y_elec, y_phot)
X.shape
del(X_elec,X_phot,y_elec,y_phot)
```

In [ ]:

```
x_train, x_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42
, stratify=y)
del(X,y)
```

In [ ]:

```
y_train = keras.utils.to_categorical(y_train, num_classes=2)
y_test = keras.utils.to_categorical(y_test, num_classes=2)
```

In [ ]:

```
learning_rate = 0.0001
batch_size = 256
num_epochs = 50
patch_size = 2
num_patches = (input_shape[0]//patch_size)**2
projection_dim = 64
num_heads = 2
transformer_units = [
    projection_dim * 2,
    projection_dim,
]
transformer_layers = 2
mlp_head_units = [512, 256]
```

In [ ]:

```
def mlp(x, hidden_units):
    for units in hidden_units:
        x = layers.Dense(units, activation=tf.nn.gelu)(x)
    return x
```

In [ ]:

```
class Patches(layers.Layer):
    def __init__(self, patch_size):
        super(Patches, self).__init__()
        self.patch_size = patch_size

    def call(self, 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
```

In [ ]:

```
class PatchEncoder(layers.Layer):
    def __init__(self, num_patches, projection_dim):
        super(PatchEncoder, self).__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):
        positions = tf.range(start=0, limit=self.num_patches, delta=1)
        encoded = self.projection(patch) + self.position_embedding(positions)
        return encoded
```

In [ ]:

```
def create_vit_classifier():
    inputs = layers.Input(shape=input_shape)
    patches = Patches(patch_size)(inputs)
    encoded_patches = PatchEncoder(num_patches, projection_dim)(patches)

    for _ in range(transformer_layers):
        x1 = layers.LayerNormalization(epsilon=1e-6)(encoded_patches)
        attention_output = layers.MultiHeadAttention(
            num_heads=num_heads, key_dim=projection_dim, dropout=0
        )(x1, x1)
        x2 = layers.Add()([attention_output, encoded_patches])
        x3 = layers.LayerNormalization(epsilon=1e-6)(x2)
        x3 = mlp(x3, hidden_units=transformer_units)
        encoded_patches = layers.Add()([x3, x2])
    representation = layers.LayerNormalization(epsilon=1e-6)(encoded_patches)
    representation = layers.Flatten()(representation)
    features = mlp(representation, hidden_units=mlp_head_units)
    outputs = layers.Dense(num_classes, activation='softmax')(features)
    model = keras.Model(inputs=inputs, outputs=outputs)
    return model
```

In [ ]:

```
def run_experiment(model):
    model.compile(optimizer=tf.optimizers.Adam(learning_rate=learning_rate), loss='categ
orical_crossentropy', metrics=[tf.keras.metrics.AUC()])

    reduce_lr = tf.keras.callbacks.ReduceLROnPlateau(monitor='val_auc', factor=0.2,
                              patience=10, min_lr=1e-10   , verbose=1)
    history = model.fit(
        x=x_train,
        y=y_train,
        batch_size=batch_size,
        epochs=num_epochs,
        validation_split=0.1,
        callbacks=[reduce_lr]
    )
    return model, history
```

In [ ]:

```
vit_classifier = create_vit_classifier()
model, history = run_experiment(vit_classifier)
```

```
Epoch 1/50
1401/1401 [==============================] - 172s 113ms/step - loss: 0.6733 - auc: 0.6385
- val_loss: 0.6478 - val_auc: 0.6729 - lr: 1.0000e-04
Epoch 2/50
1401/1401 [==============================] - 163s 116ms/step - loss: 0.6318 - auc: 0.6976
- val_loss: 0.6121 - val_auc: 0.7287 - lr: 1.0000e-04
Epoch 3/50
1401/1401 [==============================] - 163s 116ms/step - loss: 0.6153 - auc: 0.7218
- val_loss: 0.6269 - val_auc: 0.7157 - lr: 1.0000e-04
Epoch 4/50
1401/1401 [==============================] - 163s 116ms/step - loss: 0.6047 - auc: 0.7355
- val_loss: 0.6086 - val_auc: 0.7398 - lr: 1.0000e-04
Epoch 5/50
1401/1401 [==============================] - 163s 116ms/step - loss: 0.5992 - auc: 0.7421
- val_loss: 0.5999 - val_auc: 0.7424 - lr: 1.0000e-04
Epoch 6/50
```

```
Epoch 6/50
1401/1401 [==============================] - 163s 116ms/step - loss: 0.5926 - auc: 0.7495
- val_loss: 0.5970 - val_auc: 0.7497 - lr: 1.0000e-04
Epoch 7/50
1401/1401 [==============================] - 163s 116ms/step - loss: 0.5877 - auc: 0.7549
- val_loss: 0.5846 - val_auc: 0.7589 - lr: 1.0000e-04
Epoch 8/50
1401/1401 [==============================] - 163s 116ms/step - loss: 0.5815 - auc: 0.7618
- val_loss: 0.5804 - val_auc: 0.7631 - lr: 1.0000e-04
Epoch 9/50
1401/1401 [==============================] - 163s 116ms/step - loss: 0.5772 - auc: 0.7664
- val_loss: 0.5825 - val_auc: 0.7609 - lr: 1.0000e-04
Epoch 10/50
1401/1401 [==============================] - 159s 114ms/step - loss: 0.5734 - auc: 0.7705
- val_loss: 0.5732 - val_auc: 0.7715 - lr: 1.0000e-04
Epoch 11/50
1401/1401 [==============================] - ETA: 0s - loss: 0.5694 - auc: 0.7747
Epoch 11: ReduceLROnPlateau reducing learning rate to 1.9999999494757503e-05.
1401/1401 [==============================] - 163s 116ms/step - loss: 0.5694 - auc: 0.7747
- val_loss: 0.5851 - val_auc: 0.7607 - lr: 1.0000e-04
Epoch 12/50
1401/1401 [==============================] - 163s 116ms/step - loss: 0.5551 - auc: 0.7890
- val_loss: 0.5681 - val_auc: 0.7759 - lr: 2.0000e-05
Epoch 13/50
1401/1401 [==============================] - 159s 114ms/step - loss: 0.5531 - auc: 0.7909
- val_loss: 0.5667 - val_auc: 0.7781 - lr: 2.0000e-05
Epoch 14/50
1401/1401 [==============================] - 159s 113ms/step - loss: 0.5516 - auc: 0.7923
- val_loss: 0.5656 - val_auc: 0.7787 - lr: 2.0000e-05
Epoch 15/50
1401/1401 [==============================] - 163s 116ms/step - loss: 0.5499 - auc: 0.7939
- val_loss: 0.5662 - val_auc: 0.7780 - lr: 2.0000e-05
Epoch 16/50
1401/1401 [==============================] - 161s 115ms/step - loss: 0.5483 - auc: 0.7954
- val_loss: 0.5657 - val_auc: 0.7786 - lr: 2.0000e-05
Epoch 17/50
1401/1401 [==============================] - 157s 112ms/step - loss: 0.5470 - auc: 0.7966
- val_loss: 0.5663 - val_auc: 0.7782 - lr: 2.0000e-05
Epoch 18/50
1401/1401 [==============================] - 160s 114ms/step - loss: 0.5454 - auc: 0.7981
- val_loss: 0.5658 - val_auc: 0.7792 - lr: 2.0000e-05
Epoch 19/50
1401/1401 [==============================] - 160s 114ms/step - loss: 0.5434 - auc: 0.8000
- val_loss: 0.5680 - val_auc: 0.7776 - lr: 2.0000e-05
Epoch 20/50
1401/1401 [==============================] - 160s 114ms/step - loss: 0.5419 - auc: 0.8014
- val_loss: 0.5671 - val_auc: 0.7773 - lr: 2.0000e-05
Epoch 21/50
1401/1401 [==============================] - ETA: 0s - loss: 0.5403 - auc: 0.8028
Epoch 21: ReduceLROnPlateau reducing learning rate to 3.999999898951501e-06.
1401/1401 [==============================] - 156s 111ms/step - loss: 0.5403 - auc: 0.8028
- val_loss: 0.5667 - val_auc: 0.7783 - lr: 2.0000e-05
Epoch 22/50
1401/1401 [==============================] - 160s 114ms/step - loss: 0.5345 - auc: 0.8081
- val_loss: 0.5653 - val_auc: 0.7794 - lr: 4.0000e-06
Epoch 23/50
1401/1401 [==============================] - 160s 114ms/step - loss: 0.5337 - auc: 0.8088
- val_loss: 0.5659 - val_auc: 0.7794 - lr: 4.0000e-06
Epoch 24/50
1401/1401 [==============================] - 160s 114ms/step - loss: 0.5332 - auc: 0.8093
- val_loss: 0.5666 - val_auc: 0.7782 - lr: 4.0000e-06
Epoch 25/50
1401/1401 [==============================] - 160s 114ms/step - loss: 0.5326 - auc: 0.8098
- val_loss: 0.5658 - val_auc: 0.7795 - lr: 4.0000e-06
Epoch 26/50
1401/1401 [==============================] - 160s 114ms/step - loss: 0.5321 - auc: 0.8102
- val_loss: 0.5662 - val_auc: 0.7793 - lr: 4.0000e-06
Epoch 27/50
1401/1401 [==============================] - 160s 114ms/step - loss: 0.5315 - auc: 0.8108
- val_loss: 0.5666 - val_auc: 0.7788 - lr: 4.0000e-06
Epoch 28/50
1401/1401 [==============================] - 160s 114ms/step - loss: 0.5311 - auc: 0.8111
- val_loss: 0.5665 - val_auc: 0.7784 - lr: 4.0000e-06
```

```
Epoch 29/50
1401/1401 [==============================] - 160s 114ms/step - loss: 0.5305 - auc: 0.8116
- val_loss: 0.5667 - val_auc: 0.7780 - lr: 4.0000e-06
Epoch 30/50
1401/1401 [==============================] - 160s 114ms/step - loss: 0.5299 - auc: 0.8122
- val_loss: 0.5668 - val_auc: 0.7789 - lr: 4.0000e-06
Epoch 31/50
1401/1401 [==============================] - ETA: 0s - loss: 0.5295 - auc: 0.8125
Epoch 31: ReduceLROnPlateau reducing learning rate to 7.999999979801942e-07.
1401/1401 [==============================] - 160s 114ms/step - loss: 0.5295 - auc: 0.8125
- val_loss: 0.5670 - val_auc: 0.7786 - lr: 4.0000e-06
Epoch 32/50
1401/1401 [==============================] - 162s 115ms/step - loss: 0.5278 - auc: 0.8140
- val_loss: 0.5667 - val_auc: 0.7792 - lr: 8.0000e-07
Epoch 33/50
1401/1401 [==============================] - 161s 115ms/step - loss: 0.5277 - auc: 0.8141
- val_loss: 0.5666 - val_auc: 0.7788 - lr: 8.0000e-07
Epoch 34/50
1401/1401 [==============================] - 162s 115ms/step - loss: 0.5275 - auc: 0.8142
- val_loss: 0.5669 - val_auc: 0.7789 - lr: 8.0000e-07
Epoch 35/50
1401/1401 [==============================] - 162s 116ms/step - loss: 0.5274 - auc: 0.8143
- val_loss: 0.5667 - val_auc: 0.7786 - lr: 8.0000e-07
Epoch 36/50
1401/1401 [==============================] - 161s 115ms/step - loss: 0.5273 - auc: 0.8144
- val_loss: 0.5671 - val_auc: 0.7786 - lr: 8.0000e-07
Epoch 37/50
1401/1401 [==============================] - 161s 115ms/step - loss: 0.5272 - auc: 0.8145
- val_loss: 0.5670 - val_auc: 0.7788 - lr: 8.0000e-07
Epoch 38/50
1401/1401 [==============================] - 161s 115ms/step - loss: 0.5271 - auc: 0.8146
- val_loss: 0.5670 - val_auc: 0.7790 - lr: 8.0000e-07
Epoch 39/50
1401/1401 [==============================] - 161s 115ms/step - loss: 0.5269 - auc: 0.8147
- val_loss: 0.5669 - val_auc: 0.7786 - lr: 8.0000e-07
Epoch 40/50
1401/1401 [==============================] - 161s 115ms/step - loss: 0.5268 - auc: 0.8148
- val_loss: 0.5669 - val_auc: 0.7787 - lr: 8.0000e-07
Epoch 41/50
1401/1401 [==============================] - ETA: 0s - loss: 0.5267 - auc: 0.8149
Epoch 41: ReduceLROnPlateau reducing learning rate to 1.600000018697756e-07.
1401/1401 [==============================] - 157s 112ms/step - loss: 0.5267 - auc: 0.8149
- val_loss: 0.5668 - val_auc: 0.7785 - lr: 8.0000e-07
Epoch 42/50
1401/1401 [==============================] - 161s 115ms/step - loss: 0.5263 - auc: 0.8153
- val_loss: 0.5670 - val_auc: 0.7787 - lr: 1.6000e-07
Epoch 43/50
1401/1401 [==============================] - 161s 115ms/step - loss: 0.5263 - auc: 0.8153
- val_loss: 0.5669 - val_auc: 0.7788 - lr: 1.6000e-07
Epoch 44/50
1401/1401 [==============================] - 161s 115ms/step - loss: 0.5263 - auc: 0.8153
- val_loss: 0.5670 - val_auc: 0.7787 - lr: 1.6000e-07
Epoch 45/50
1401/1401 [==============================] - 157s 112ms/step - loss: 0.5262 - auc: 0.8153
- val_loss: 0.5670 - val_auc: 0.7788 - lr: 1.6000e-07
Epoch 46/50
1401/1401 [==============================] - 157s 112ms/step - loss: 0.5262 - auc: 0.8154
- val_loss: 0.5670 - val_auc: 0.7787 - lr: 1.6000e-07
Epoch 47/50
1401/1401 [==============================] - 161s 115ms/step - loss: 0.5262 - auc: 0.8154
- val_loss: 0.5670 - val_auc: 0.7788 - lr: 1.6000e-07
Epoch 48/50
1401/1401 [==============================] - 157s 112ms/step - loss: 0.5262 - auc: 0.8154
- val_loss: 0.5670 - val_auc: 0.7788 - lr: 1.6000e-07
Epoch 49/50
1401/1401 [==============================] - 161s 115ms/step - loss: 0.5261 - auc: 0.8154
- val_loss: 0.5670 - val_auc: 0.7788 - lr: 1.6000e-07
Epoch 50/50
1401/1401 [==============================] - 161s 115ms/step - loss: 0.5261 - auc: 0.8154
- val_loss: 0.5670 - val_auc: 0.7787 - lr: 1.6000e-07
```

In [ ]:

```
model.evaluate(x_test,y_test)
```

```
3113/3113 [==============================] - 30s 10ms/step - loss: 0.5670 - auc: 0.7786
```

Out[ ]:

```
[0.5669955611228943, 0.7785612940788269]
```

Discuss the resulting performance of the 2 chosen architectures.

> **The AUC score is low because of lower no. of epochs still it is comperable to keras and pytorch score of 0.8.**