```python
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.optimizers import Adam



# Generate toy dataset
from sklearn.datasets import make_classification
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler


X, y = make_classification(n_samples=1000, n_features=20, n_classes=2)
X = StandardScaler().fit_transform(X)
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2)


# Define model
model = Sequential([
    Dense(16, activation='relu', input_shape=(X.shape[1],)),
    Dense(8, activation='relu'),
    Dense(1, activation='sigmoid')  # Binary classification
])
```

```
/usr/local/lib/python3.11/dist-packages/keras/src/layers/core/
dense.py:87: UserWarning: Do not pass an `input_shape`/`input_dim`
argument to a layer. When using Sequential models, prefer using an
`Input(shape)` object as the first layer in the model instead.
  super().__init__(activity_regularizer=activity_regularizer,
**kwargs)
```

```python
model.compile(optimizer=Adam(), loss='binary_crossentropy',
metrics=['accuracy'])

# Train
model.fit(X_train, y_train, epochs=20, batch_size=32,
validation_data=(X_test, y_test))
```

```
Epoch 1/20
25/25 ━━━━━━━━━━━━━━━━━━━━ 2s 20ms/step - accuracy: 0.4591 - loss:
0.7816 - val_accuracy: 0.4350 - val_loss: 0.7274
Epoch 2/20
25/25 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.4783 - loss:
0.7254 - val_accuracy: 0.5650 - val_loss: 0.6857
Epoch 3/20
25/25 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - accuracy: 0.5589 - loss:
0.6823 - val_accuracy: 0.6050 - val_loss: 0.6511
Epoch 4/20
```

```
25/25 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.6599 - loss:
0.6339 - val_accuracy: 0.6850 - val_loss: 0.6147
Epoch 5/20
25/25 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.7166 - loss:
0.5948 - val_accuracy: 0.7350 - val_loss: 0.5731
Epoch 6/20
25/25 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.7678 - loss:
0.5516 - val_accuracy: 0.7750 - val_loss: 0.5253
Epoch 7/20
25/25 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.8354 - loss:
0.4735 - val_accuracy: 0.7950 - val_loss: 0.4739
Epoch 8/20
25/25 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.8436 - loss:
0.4282 - val_accuracy: 0.8200 - val_loss: 0.4286
Epoch 9/20
25/25 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.8480 - loss:
0.3917 - val_accuracy: 0.8450 - val_loss: 0.3889
Epoch 10/20
25/25 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.8875 - loss:
0.3313 - val_accuracy: 0.8500 - val_loss: 0.3575
Epoch 11/20
25/25 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9102 - loss:
0.2805 - val_accuracy: 0.8500 - val_loss: 0.3351
Epoch 12/20
25/25 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.9204 - loss:
0.2458 - val_accuracy: 0.8600 - val_loss: 0.3178
Epoch 13/20
25/25 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.9233 - loss:
0.2462 - val_accuracy: 0.8650 - val_loss: 0.3055
Epoch 14/20
25/25 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.9221 - loss:
0.2282 - val_accuracy: 0.8700 - val_loss: 0.2959
Epoch 15/20
25/25 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9342 - loss:
0.2084 - val_accuracy: 0.8800 - val_loss: 0.2898
Epoch 16/20
25/25 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.9188 - loss:
0.2107 - val_accuracy: 0.8800 - val_loss: 0.2868
Epoch 17/20
25/25 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9320 - loss:
0.1898 - val_accuracy: 0.8800 - val_loss: 0.2849
Epoch 18/20
25/25 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9391 - loss:
0.1939 - val_accuracy: 0.8850 - val_loss: 0.2845
Epoch 19/20
25/25 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9256 - loss:
0.1835 - val_accuracy: 0.8850 - val_loss: 0.2825
Epoch 20/20
```

```
25/25 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9424 - loss:
0.1697 - val_accuracy: 0.8800 - val_loss: 0.2850

<keras.src.callbacks.history.History at 0x79f626997e90>
```

logistic evaluation

```
import tensorflow as tf
from sklearn.datasets import make_classification
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler


# Dataset
X, y = make_classification(n_samples=1000, n_features=10, n_classes=2)
X = StandardScaler().fit_transform(X)
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2)


# Logistic Regression Model = 1 Dense layer with sigmoid
model = tf.keras.Sequential([
    tf.keras.layers.Dense(1, activation='sigmoid',
input_shape=(X.shape[1],))
])

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

```
/usr/local/lib/python3.11/dist-packages/keras/src/layers/core/
dense.py:87: UserWarning: Do not pass an `input_shape`/`input_dim`
argument to a layer. When using Sequential models, prefer using an
`Input(shape)` object as the first layer in the model instead.
  super().__init__(activity_regularizer=activity_regularizer,
**kwargs)
```

```
# Train
model.fit(X_train, y_train, epochs=10, batch_size=32,
validation_data=(X_test, y_test))
```

```
Epoch 1/10
25/25 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - accuracy: 0.7024 - loss:
0.5644 - val_accuracy: 0.7450 - val_loss: 0.4987
Epoch 2/10
25/25 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - accuracy: 0.7280 - loss:
0.5421 - val_accuracy: 0.7650 - val_loss: 0.4851
```

```
Epoch 3/10
25/25 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.7382 - loss:
0.5228 - val_accuracy: 0.7800 - val_loss: 0.4724
Epoch 4/10
25/25 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.7354 - loss:
0.5494 - val_accuracy: 0.7950 - val_loss: 0.4607
Epoch 5/10
25/25 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.7651 - loss:
0.5057 - val_accuracy: 0.8050 - val_loss: 0.4501
Epoch 6/10
25/25 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - accuracy: 0.7476 - loss:
0.5034 - val_accuracy: 0.8100 - val_loss: 0.4396
Epoch 7/10
25/25 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.7932 - loss:
0.4817 - val_accuracy: 0.8250 - val_loss: 0.4302
Epoch 8/10
25/25 ━━━━━━━━━━━━━━━━━━━━ 0s 14ms/step - accuracy: 0.7688 - loss:
0.5039 - val_accuracy: 0.8450 - val_loss: 0.4215
Epoch 9/10
25/25 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.8067 - loss:
0.4557 - val_accuracy: 0.8550 - val_loss: 0.4138
Epoch 10/10
25/25 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.8248 - loss:
0.4511 - val_accuracy: 0.8700 - val_loss: 0.4066
```

```
<keras.src.callbacks.history.History at 0x79f61ce74fd0>
```

```python
# Evaluate
loss, accuracy = model.evaluate(X_test, y_test)
print(f"Test Accuracy: {accuracy:.4f}")
```

```
7/7 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.8934 - loss: 0.3880
```

```
Test Accuracy: 0.8700
```