```
import tensorflow as tf
import numpy as np
from tensorflow.keras import datasets, layers, models
from keras.models import Sequential
from keras.layers import Dense


(X_train, y_train), (X_test, y_test) = datasets.cifar10.load_data()


X_train.shape
```

```
(50000, 32, 32, 3)
```

```
X_test.shape
```

```
(10000, 32, 32, 3)
```

```
y_train.shape
```

```
(50000, 1)
```

```
y_test.shape
```

```
(10000, 1)
```

```
y_train[:5]
```

```
array([[6],
       [9],
       [9],
       [4],
       [1]], dtype=uint8)
```

```
y_train = y_train.reshape(-1,)
y_train[:5]
```

```
array([6, 9, 9, 4, 1], dtype=uint8)
```

```
y_test = y_test.reshape(-1,)


X_train = X_train / 255.0
X_test = X_test / 255.0
```

## ANN algorithm

```
ann = models.Sequential([
        layers.Flatten(input_shape=(32,32,3)),
        layers.Dense(3000, activation='relu'),
        layers.Dense(1000, activation='relu'),
        layers.Dense(10, activation='softmax')
    ])

ann.compile(optimizer='SGD',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])

ann.fit(X_train, y_train, epochs=100)
```

```
Epoch 1/100
1563/1563 [==============================] - 137s 87ms/step - loss: 1.8129 - accuracy: 0.3523
Epoch 2/100
1563/1563 [==============================] - 130s 83ms/step - loss: 1.6265 - accuracy: 0.4256
Epoch 3/100
1563/1563 [==============================] - 129s 82ms/step - loss: 1.5426 - accuracy: 0.4547
Epoch 4/100
1563/1563 [==============================] - 132s 84ms/step - loss: 1.4830 - accuracy: 0.4765
Epoch 5/100
1563/1563 [==============================] - 129s 83ms/step - loss: 1.4340 - accuracy: 0.4945
Epoch 6/100
1563/1563 [==============================] - 130s 83ms/step - loss: 1.3901 - accuracy: 0.5118
Epoch 7/100
1563/1563 [==============================] - 127s 81ms/step - loss: 1.3516 - accuracy: 0.5261
Epoch 8/100
1563/1563 [==============================] - 130s 83ms/step - loss: 1.3162 - accuracy: 0.5401
Epoch 9/100
```

```
1563/1563 [==============================] - 131s 84ms/step - loss: 1.2838 - accuracy: 0.5496
Epoch 10/100
1563/1563 [==============================] - 129s 83ms/step - loss: 1.2532 - accuracy: 0.5611
Epoch 11/100
1563/1563 [==============================] - 132s 84ms/step - loss: 1.2223 - accuracy: 0.5730
Epoch 12/100
1563/1563 [==============================] - 130s 83ms/step - loss: 1.1970 - accuracy: 0.5822
Epoch 13/100
1563/1563 [==============================] - 130s 83ms/step - loss: 1.1695 - accuracy: 0.5921
Epoch 14/100
1563/1563 [==============================] - 131s 84ms/step - loss: 1.1396 - accuracy: 0.6022
Epoch 15/100
1563/1563 [==============================] - 132s 84ms/step - loss: 1.1149 - accuracy: 0.6132
Epoch 16/100
1563/1563 [==============================] - 130s 83ms/step - loss: 1.0863 - accuracy: 0.6217
Epoch 17/100
1563/1563 [==============================] - 130s 83ms/step - loss: 1.0595 - accuracy: 0.6312
Epoch 18/100
1563/1563 [==============================] - 132s 84ms/step - loss: 1.0356 - accuracy: 0.6396
Epoch 19/100
1563/1563 [==============================] - 130s 83ms/step - loss: 1.0096 - accuracy: 0.6489
Epoch 20/100
1563/1563 [==============================] - 131s 84ms/step - loss: 0.9856 - accuracy: 0.6585
Epoch 21/100
1563/1563 [==============================] - 130s 83ms/step - loss: 0.9592 - accuracy: 0.6682
Epoch 22/100
1563/1563 [==============================] - 130s 83ms/step - loss: 0.9332 - accuracy: 0.6790
Epoch 23/100
1563/1563 [==============================] - 129s 83ms/step - loss: 0.9054 - accuracy: 0.6867
Epoch 24/100
1563/1563 [==============================] - 132s 85ms/step - loss: 0.8801 - accuracy: 0.6957
Epoch 25/100
1563/1563 [==============================] - 131s 84ms/step - loss: 0.8556 - accuracy: 0.7046
Epoch 26/100
1563/1563 [==============================] - 130s 83ms/step - loss: 0.8328 - accuracy: 0.7143
Epoch 27/100
1563/1563 [==============================] - 131s 84ms/step - loss: 0.8056 - accuracy: 0.7228
Epoch 28/100
1563/1563 [==============================] - 131s 84ms/step - loss: 0.7801 - accuracy: 0.7332
Epoch 29/100
1563/1563 [==============================] - 130s 83ms/step - loss: 0.7549 - accuracy: 0.7420
```

```
from sklearn.metrics import confusion_matrix , classification_report
import numpy as np
y_pred = ann.predict(X_test)
y_pred_classes = [np.argmax(element) for element in y_pred]

print("Classification Report: \n", classification_report(y_test, y_pred_classes))
```

```
313/313 [==============================] - 8s 26ms/step
Classification Report:
              precision    recall  f1-score   support

           0       0.68      0.62      0.65      1000
           1       0.69      0.66      0.67      1000
           2       0.50      0.45      0.47      1000
           3       0.42      0.40      0.41      1000
           4       0.51      0.52      0.51      1000
           5       0.48      0.50      0.49      1000
           6       0.61      0.68      0.64      1000
           7       0.64      0.62      0.63      1000
           8       0.67      0.74      0.70      1000
           9       0.59      0.63      0.61      1000

    accuracy                           0.58     10000
   macro avg       0.58      0.58      0.58     10000
weighted avg       0.58      0.58      0.58     10000
```