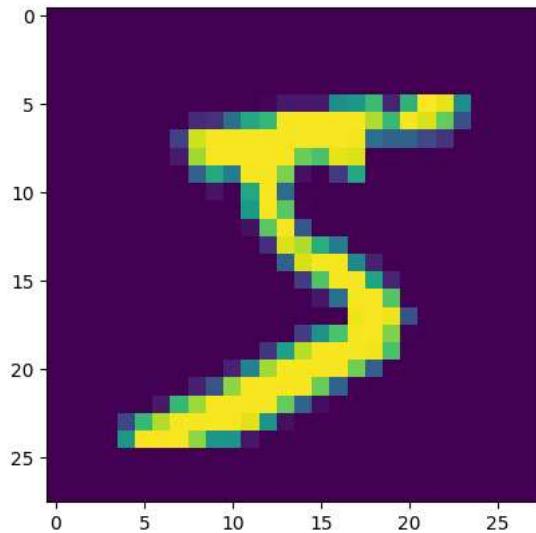


```
1 plt.imshow(X_train[0])  
2  
<matplotlib.image.AxesImage at 0x7dde56bb35b0>
```



```
1 print(X_train.shape) # shape of X_train which has 60000 images with the size of matrix as 28X28  
2 print(y_train.shape)  
3 print(X_test.shape)  
4 print(y_test.shape)  
  
(60000, 28, 28)  
(60000,)  
(10000, 28, 28)  
(10000,)
```

```
1 from sklearn.model_selection import train_test_split # splitting the data with test_size of 20%  
2 X_train, X_val, y_train, y_val = train_test_split(X_train,y_train, test_size=0.20)
```

```
1 X_train  
  
array([[[0, 0, 0, ..., 0, 0, 0],  
       [0, 0, 0, ..., 0, 0, 0],  
       [0, 0, 0, ..., 0, 0, 0],  
       ...,  
       [0, 0, 0, ..., 0, 0, 0],  
       [0, 0, 0, ..., 0, 0, 0],  
       [0, 0, 0, ..., 0, 0, 0]],  
  
      [[0, 0, 0, ..., 0, 0, 0],  
       [0, 0, 0, ..., 0, 0, 0],  
       [0, 0, 0, ..., 0, 0, 0],  
       ...,  
       [0, 0, 0, ..., 0, 0, 0],  
       [0, 0, 0, ..., 0, 0, 0],  
       [0, 0, 0, ..., 0, 0, 0]],  
  
      [[0, 0, 0, ..., 0, 0, 0],  
       [0, 0, 0, ..., 0, 0, 0],  
       [0, 0, 0, ..., 0, 0, 0],  
       ...,  
       [0, 0, 0, ..., 0, 0, 0],  
       [0, 0, 0, ..., 0, 0, 0],  
       [0, 0, 0, ..., 0, 0, 0]]]
```

```

    ...,
    [0, 0, 0, ..., 0, 0, 0],
    [0, 0, 0, ..., 0, 0, 0],
    [0, 0, 0, ..., 0, 0, 0]],

    ...,
    [[0, 0, 0, ..., 0, 0, 0],
     [0, 0, 0, ..., 0, 0, 0],
     [0, 0, 0, ..., 0, 0, 0],
     ...,
     [0, 0, 0, ..., 0, 0, 0],
     [0, 0, 0, ..., 0, 0, 0],
     [0, 0, 0, ..., 0, 0, 0]],

    [[0, 0, 0, ..., 0, 0, 0],
     [0, 0, 0, ..., 0, 0, 0],
     [0, 0, 0, ..., 0, 0, 0],
     ...,
     [0, 0, 0, ..., 0, 0, 0],
     [0, 0, 0, ..., 0, 0, 0],
     [0, 0, 0, ..., 0, 0, 0]],

    ...,
    [[0, 0, 0, ..., 0, 0, 0],
     [0, 0, 0, ..., 0, 0, 0],
     [0, 0, 0, ..., 0, 0, 0],
     ...,
     [0, 0, 0, ..., 0, 0, 0],
     [0, 0, 0, ..., 0, 0, 0],
     [0, 0, 0, ..., 0, 0, 0]]], dtype=uint8)

```

```
1 X_train.shape
```

```
(48000, 28, 28)
```

```

1 def get_quad_features(image):
2 # Feature extraction: average the pixel values in the quadrants in each image to
3 # generate a feature vector of 4 values for each image
4   q1 = image[:14,:14]
5   q2 = image[:14,14:]
6   q3 = image[14:,:14]
7   q4 = image[14:,14:]
8
9   q1_avg = np.mean(q1)
10  q2_avg = np.mean(q2)
11  q3_avg = np.mean(q3)
12  q4_avg = np.mean(q4)
13  return np.array([q1_avg,q2_avg,q3_avg,q4_avg])

```

```
1 X_train_features = np.array([get_quad_features(image) for image in X_train])
```

```
1 X_val_features = np.array([get_quad_features(image) for image in X_val])
2 X_test_features = np.array([get_quad_features(image) for image in X_test])
```

```

1 #Convert the label vectors for all the sets to binary class matrices using
2 #to_categorical() Keras function.
3 y_train_categorical = to_categorical(y_train)
4 y_val_categorical = to_categorical(y_val)
5 y_test_categorical = to_categorical(y_test)

```

```
1 model = Sequential()
```

```
1 X_train_features.shape
```

```
(48000, 4)
```

```

1 from tensorflow.keras.layers import Flatten,Dense
2 '''d) Build, compile, train, and then evaluate:
3     i. Build a neural network with 1 layer that contains 16 nodes using the Keras
4     library.
5     ii. Compile the network. Make sure to select a correct loss function for this
6         classification problem. Use stochastic gradient descent learning (SGD,
7         learning rate of 0.0001). Explain your selection of the loss function.
8     iii. Train the network for 30 epochs and a batch size of 16.
9     iv. Plot the training loss (i.e., the learning curve) for all the epochs.
10    v. Use the evaluate() Keras function to find the training and validation loss and
11        accuracy.'''
12 #model.add(Flatten(input_shape=(28,28)))
13 model.add(Dense(16,activation='relu',input_shape=(4,)))
14 model.add(Dense(10,activation='softmax'))

```

```
1 model.compile(optimizer=SGD(learning_rate=0.0001), loss='categorical_crossentropy', metrics=['accuracy'])

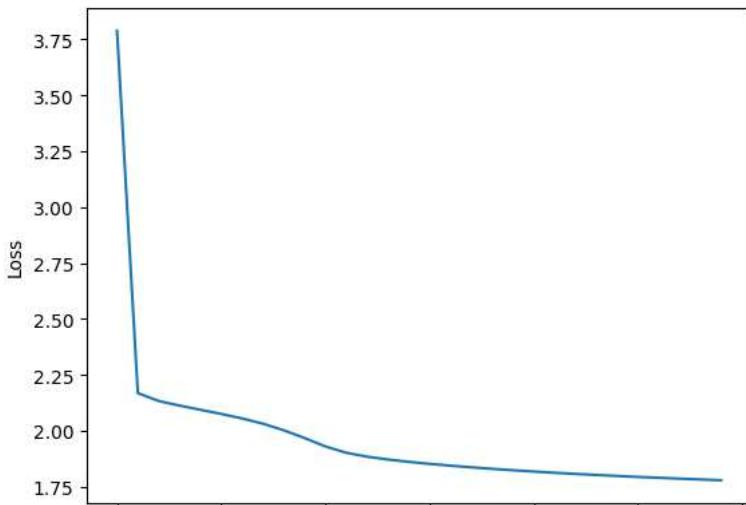
1 #X_train_features = X_train_features.reshape(-1,28,28)
2 #X_val_features = X_val_features.reshape(-1,28,28)
3 #X_test_features = X_test_features.reshape(-1,28,28)

1 y_train_categorical.shape
(48000, 10)

1 X_train_features.shape
(48000, 4)

1 history = model.fit(X_train_features,y_train_categorical, epochs=30, batch_size=16,verbose=1,validation_split=0.2)
2400/2400 [=====] - 9s 4ms/step - loss: 2.1688 - accuracy: 0.2710 - val_loss: 2.1421 - val_accuracy: 0.
Epoch 3/30
2400/2400 [=====] - 4s 2ms/step - loss: 2.1335 - accuracy: 0.2801 - val_loss: 2.1189 - val_accuracy: 0.
Epoch 4/30
2400/2400 [=====] - 6s 2ms/step - loss: 2.1127 - accuracy: 0.2866 - val_loss: 2.1002 - val_accuracy: 0.
Epoch 5/30
2400/2400 [=====] - 4s 2ms/step - loss: 2.0945 - accuracy: 0.2952 - val_loss: 2.0825 - val_accuracy: 0.
Epoch 6/30
2400/2400 [=====] - 5s 2ms/step - loss: 2.0758 - accuracy: 0.3046 - val_loss: 2.0640 - val_accuracy: 0.
Epoch 7/30
2400/2400 [=====] - 5s 2ms/step - loss: 2.0554 - accuracy: 0.3140 - val_loss: 2.0438 - val_accuracy: 0.
Epoch 8/30
2400/2400 [=====] - 5s 2ms/step - loss: 2.0323 - accuracy: 0.3200 - val_loss: 2.0194 - val_accuracy: 0.
Epoch 9/30
2400/2400 [=====] - 5s 2ms/step - loss: 2.0027 - accuracy: 0.3194 - val_loss: 1.9894 - val_accuracy: 0.
Epoch 10/30
2400/2400 [=====] - 4s 2ms/step - loss: 1.9681 - accuracy: 0.3261 - val_loss: 1.9518 - val_accuracy: 0.
Epoch 11/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.9304 - accuracy: 0.3351 - val_loss: 1.9165 - val_accuracy: 0.
Epoch 12/30
2400/2400 [=====] - 6s 2ms/step - loss: 1.9023 - accuracy: 0.3433 - val_loss: 1.8938 - val_accuracy: 0.
Epoch 13/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.8846 - accuracy: 0.3448 - val_loss: 1.8809 - val_accuracy: 0.
Epoch 14/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.8719 - accuracy: 0.3464 - val_loss: 1.8680 - val_accuracy: 0.
Epoch 15/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.8616 - accuracy: 0.3487 - val_loss: 1.8579 - val_accuracy: 0.
Epoch 16/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.8524 - accuracy: 0.3502 - val_loss: 1.8495 - val_accuracy: 0.
Epoch 17/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.8444 - accuracy: 0.3512 - val_loss: 1.8430 - val_accuracy: 0.
Epoch 18/30
2400/2400 [=====] - 4s 2ms/step - loss: 1.8371 - accuracy: 0.3545 - val_loss: 1.8337 - val_accuracy: 0.
Epoch 19/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.8303 - accuracy: 0.3558 - val_loss: 1.8285 - val_accuracy: 0.
Epoch 20/30
2400/2400 [=====] - 6s 3ms/step - loss: 1.8240 - accuracy: 0.3571 - val_loss: 1.8210 - val_accuracy: 0.
Epoch 21/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.8182 - accuracy: 0.3588 - val_loss: 1.8161 - val_accuracy: 0.
Epoch 22/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.8127 - accuracy: 0.3605 - val_loss: 1.8105 - val_accuracy: 0.
Epoch 23/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.8077 - accuracy: 0.3622 - val_loss: 1.8045 - val_accuracy: 0.
Epoch 24/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.8030 - accuracy: 0.3640 - val_loss: 1.7996 - val_accuracy: 0.
Epoch 25/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.7984 - accuracy: 0.3662 - val_loss: 1.7951 - val_accuracy: 0.
Epoch 26/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.7941 - accuracy: 0.3666 - val_loss: 1.7911 - val_accuracy: 0.
Epoch 27/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.7902 - accuracy: 0.3672 - val_loss: 1.7859 - val_accuracy: 0.
Epoch 28/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.7864 - accuracy: 0.3695 - val_loss: 1.7827 - val_accuracy: 0.
Epoch 29/30
2400/2400 [=====] - 4s 2ms/step - loss: 1.7829 - accuracy: 0.3699 - val_loss: 1.7782 - val_accuracy: 0.
Epoch 30/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.7793 - accuracy: 0.3717 - val_loss: 1.7754 - val_accuracy: 0.

1 plt.plot(history.history['loss'])
2 plt.xlabel('Epoch')
3 plt.ylabel('Loss')
4 plt.show()
```



```
1 train_loss, train_accuracy = model.evaluate(X_train_features,y_train_categorical)
2 val_loss, val_accuracy = model.evaluate(X_val_features,y_val_categorical)
```

```
1500/1500 [=====] - 2s 2ms/step - loss: 1.7771 - accuracy: 0.3716
375/375 [=====] - 1s 2ms/step - loss: 1.7700 - accuracy: 0.3711
```

```
1 print(f'Train loss {train_loss}')
2 print(f'Train accuracy {train_accuracy}')
3 print(f'Validation loss {val_loss}')
4 print(f'Validation accuracy {val_accuracy}')
```

```
Train loss 1.7771121263504028
Train accuracy 0.37158334255218506
Validation loss 1.770027995109558
Validation accuracy 0.3710833191871643
```

```
1 # Model 1: 1 layer with 16 nodes
2 model1 = Sequential()
3 model1.add(Dense(16, activation='relu', input_shape=(4,)))
4 model1.add(Dense(10, activation='softmax'))
5 model1.compile(loss='categorical_crossentropy', optimizer=SGD(learning_rate=0.0001), metrics=['accuracy'])
6 model1.fit(X_train_features,y_train_categorical, epochs=30, batch_size=16, verbose=1, validation_split=0.2)
7 train_loss1, train_accuracy1 = model1.evaluate(X_train_features,y_train_categorical)
8 val_loss1, val_accuracy1 = model1.evaluate(X_val_features,y_val_categorical)
9 print(f'Train loss {train_loss1}')
10 print(f'Train accuracy {train_accuracy1}')
11 print(f'Validation loss {val_loss1}')
12 print(f'Validation accuracy {val_accuracy1}')

2400/2400 [=====] - 5s 2ms/step - loss: 1.7313 - accuracy: 0.3932 - val_loss: 1.7252 - val_accuracy: 0.
Epoch 6/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.7150 - accuracy: 0.3988 - val_loss: 1.7082 - val_accuracy: 0.
Epoch 7/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.7011 - accuracy: 0.4077 - val_loss: 1.6911 - val_accuracy: 0.
Epoch 8/30
2400/2400 [=====] - 4s 2ms/step - loss: 1.6883 - accuracy: 0.4127 - val_loss: 1.6800 - val_accuracy: 0.
Epoch 9/30
2400/2400 [=====] - 4s 2ms/step - loss: 1.6773 - accuracy: 0.4193 - val_loss: 1.6709 - val_accuracy: 0.
Epoch 10/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.6675 - accuracy: 0.4247 - val_loss: 1.6593 - val_accuracy: 0.
Epoch 11/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.6584 - accuracy: 0.4259 - val_loss: 1.6474 - val_accuracy: 0.
Epoch 12/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.6501 - accuracy: 0.4289 - val_loss: 1.6402 - val_accuracy: 0.
Epoch 13/30
2400/2400 [=====] - 4s 2ms/step - loss: 1.6426 - accuracy: 0.4313 - val_loss: 1.6353 - val_accuracy: 0.
Epoch 14/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.6355 - accuracy: 0.4341 - val_loss: 1.6262 - val_accuracy: 0.
Epoch 15/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.6291 - accuracy: 0.4359 - val_loss: 1.6183 - val_accuracy: 0.
Epoch 16/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.6237 - accuracy: 0.4370 - val_loss: 1.6123 - val_accuracy: 0.
Epoch 17/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.6182 - accuracy: 0.4399 - val_loss: 1.6080 - val_accuracy: 0.
Epoch 18/30
```

```

2400/2400 [=====] - 5s 2ms/step - loss: 1.5930 - accuracy: 0.4467 - val_loss: 1.5825 - val_accuracy: 0.▲
Epoch 24/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.5896 - accuracy: 0.4457 - val_loss: 1.5780 - val_accuracy: 0.▲
Epoch 25/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.5872 - accuracy: 0.4459 - val_loss: 1.5756 - val_accuracy: 0.▲
Epoch 26/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.5837 - accuracy: 0.4480 - val_loss: 1.5764 - val_accuracy: 0.▲
Epoch 27/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.5811 - accuracy: 0.4460 - val_loss: 1.5687 - val_accuracy: 0.▲
Epoch 28/30
2400/2400 [=====] - 4s 2ms/step - loss: 1.5777 - accuracy: 0.4490 - val_loss: 1.5708 - val_accuracy: 0.▲
Epoch 29/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.5758 - accuracy: 0.4475 - val_loss: 1.5622 - val_accuracy: 0.▲
Epoch 30/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.5729 - accuracy: 0.4496 - val_loss: 1.5611 - val_accuracy: 0.▲
1500/1500 [=====] - 2s 2ms/step - loss: 1.5687 - accuracy: 0.4509
375/375 [=====] - 1s 2ms/step - loss: 1.5590 - accuracy: 0.4505
Train loss 1.5686849355697632
Train accuracy 0.4509166771339417
Validation loss 1.5590364933013916
Validation accuracy 0.4505000114440918

```

```

1 # Model 2: 1 layer with 64 nodes
2 model2 = Sequential()
3 model2.add(Dense(64, activation='relu', input_shape=(4,)))
4 model2.add(Dense(10, activation='softmax'))
5 model2.compile(loss='categorical_crossentropy', optimizer=SGD(learning_rate=0.0001), metrics=['accuracy'])
6 model2.fit(X_train_features,y_train_categorical, epochs=30, batch_size=16, verbose=1, validation_split=0.2)
7 train_loss2, train_accuracy2 = model2.evaluate(X_train_features,y_train_categorical)
8 val_loss2, val_accuracy2 = model2.evaluate(X_val_features,y_val_categorical)
9 print(f'Train loss {train_loss2}')
10 print(f'Train accuracy {train_accuracy2}')
11 print(f'Validation loss {val_loss2}')
12 print(f'Validation accuracy {val_accuracy2}')

2400/2400 [=====] - 5s 2ms/step - loss: 1.5428 - accuracy: 0.4575 - val_loss: 1.5365 - val_accuracy: 0.▲
Epoch 6/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.5303 - accuracy: 0.4589 - val_loss: 1.5170 - val_accuracy: 0.▲
Epoch 7/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.5195 - accuracy: 0.4639 - val_loss: 1.5091 - val_accuracy: 0.▲
Epoch 8/30
2400/2400 [=====] - 4s 2ms/step - loss: 1.5101 - accuracy: 0.4670 - val_loss: 1.4973 - val_accuracy: 0.▲
Epoch 9/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.5024 - accuracy: 0.4680 - val_loss: 1.4923 - val_accuracy: 0.▲
Epoch 10/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.4957 - accuracy: 0.4745 - val_loss: 1.4811 - val_accuracy: 0.▲
Epoch 11/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.4893 - accuracy: 0.4719 - val_loss: 1.4775 - val_accuracy: 0.▲
Epoch 12/30
2400/2400 [=====] - 4s 2ms/step - loss: 1.4849 - accuracy: 0.4747 - val_loss: 1.4707 - val_accuracy: 0.▲
Epoch 13/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.4800 - accuracy: 0.4760 - val_loss: 1.4657 - val_accuracy: 0.▲
Epoch 14/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.4753 - accuracy: 0.4795 - val_loss: 1.4660 - val_accuracy: 0.▲
Epoch 15/30
2400/2400 [=====] - 4s 2ms/step - loss: 1.4718 - accuracy: 0.4803 - val_loss: 1.4575 - val_accuracy: 0.▲
Epoch 16/30
2400/2400 [=====] - 4s 2ms/step - loss: 1.4683 - accuracy: 0.4797 - val_loss: 1.4642 - val_accuracy: 0.▲
Epoch 17/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.4651 - accuracy: 0.4809 - val_loss: 1.4562 - val_accuracy: 0.▲
Epoch 18/30
2400/2400 [=====] - 4s 2ms/step - loss: 1.4621 - accuracy: 0.4832 - val_loss: 1.4541 - val_accuracy: 0.▲
Epoch 19/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.4605 - accuracy: 0.4800 - val_loss: 1.4503 - val_accuracy: 0.▲
Epoch 20/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.4575 - accuracy: 0.4842 - val_loss: 1.4477 - val_accuracy: 0.▲
Epoch 21/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.4544 - accuracy: 0.4852 - val_loss: 1.4507 - val_accuracy: 0.▲
Epoch 22/30
2400/2400 [=====] - 8s 3ms/step - loss: 1.4529 - accuracy: 0.4862 - val_loss: 1.4500 - val_accuracy: 0.▲
Epoch 23/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.4510 - accuracy: 0.4833 - val_loss: 1.4417 - val_accuracy: 0.▲
Epoch 24/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.4492 - accuracy: 0.4874 - val_loss: 1.4384 - val_accuracy: 0.▲
Epoch 25/30
2400/2400 [=====] - 4s 2ms/step - loss: 1.4469 - accuracy: 0.4860 - val_loss: 1.4388 - val_accuracy: 0.▲
Epoch 26/30

```

```
375/375 [=====] - 1s 2ms/step - loss: 1.4253 - accuracy: 0.4948
Train loss 1.4331797361373901
Train accuracy 0.49172917008399963
Validation loss 1.425349235534668
Validation accuracy 0.4948333203792572
```

```
1 # Model 3: 1 layer with 128 nodes
2 model3 = Sequential()
3 model3.add(Dense(128, activation='relu', input_shape=(4,)))
4 model3.add(Dense(10, activation='softmax'))
5 model3.compile(loss='categorical_crossentropy', optimizer=SGD(learning_rate=0.0001), metrics=['accuracy'])
6 model3.fit(X_train_features,y_train_categorical, epochs=30, batch_size=16, verbose=1, validation_split=0.2)
7 train_loss3, train_accuracy3 = model3.evaluate(X_train_features,y_train_categorical)
8 val_loss3, val_accuracy3 = model3.evaluate(X_val_features,y_val_categorical)
9 print(f'Train loss {train_loss3}')
10 print(f'Train accuracy {train_accuracy3}')
11 print(f'Validation loss {val_loss3}')
12 print(f'Validation accuracy {val_accuracy3}')

2400/2400 [=====] - 5s 2ms/step - loss: 1.5288 - accuracy: 0.4611 - val_loss: 1.5084 - val_accuracy: 0.
Epoch 6/30
2400/2400 [=====] - 4s 2ms/step - loss: 1.5183 - accuracy: 0.4638 - val_loss: 1.5011 - val_accuracy: 0.
Epoch 7/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.5107 - accuracy: 0.4645 - val_loss: 1.4943 - val_accuracy: 0.
Epoch 8/30
2400/2400 [=====] - 4s 2ms/step - loss: 1.5036 - accuracy: 0.4690 - val_loss: 1.4913 - val_accuracy: 0.
Epoch 9/30
2400/2400 [=====] - 4s 2ms/step - loss: 1.4984 - accuracy: 0.4687 - val_loss: 1.4857 - val_accuracy: 0.
Epoch 10/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.4919 - accuracy: 0.4712 - val_loss: 1.4752 - val_accuracy: 0.
Epoch 11/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.4874 - accuracy: 0.4711 - val_loss: 1.4777 - val_accuracy: 0.
Epoch 12/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.4816 - accuracy: 0.4745 - val_loss: 1.4654 - val_accuracy: 0.
Epoch 13/30
2400/2400 [=====] - 4s 2ms/step - loss: 1.4768 - accuracy: 0.4774 - val_loss: 1.4701 - val_accuracy: 0.
Epoch 14/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.4739 - accuracy: 0.4768 - val_loss: 1.4601 - val_accuracy: 0.
Epoch 15/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.4702 - accuracy: 0.4790 - val_loss: 1.4616 - val_accuracy: 0.
Epoch 16/30
2400/2400 [=====] - 4s 2ms/step - loss: 1.4667 - accuracy: 0.4809 - val_loss: 1.4585 - val_accuracy: 0.
Epoch 17/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.4647 - accuracy: 0.4804 - val_loss: 1.4543 - val_accuracy: 0.
Epoch 18/30
2400/2400 [=====] - 4s 2ms/step - loss: 1.4615 - accuracy: 0.4802 - val_loss: 1.4492 - val_accuracy: 0.
Epoch 19/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.4593 - accuracy: 0.4811 - val_loss: 1.4607 - val_accuracy: 0.
Epoch 20/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.4573 - accuracy: 0.4821 - val_loss: 1.4420 - val_accuracy: 0.
Epoch 21/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.4549 - accuracy: 0.4811 - val_loss: 1.4548 - val_accuracy: 0.
Epoch 22/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.4526 - accuracy: 0.4840 - val_loss: 1.4408 - val_accuracy: 0.
Epoch 23/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.4513 - accuracy: 0.4826 - val_loss: 1.4413 - val_accuracy: 0.
Epoch 24/30
2400/2400 [=====] - 4s 2ms/step - loss: 1.4492 - accuracy: 0.4842 - val_loss: 1.4383 - val_accuracy: 0.
Epoch 25/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.4471 - accuracy: 0.4842 - val_loss: 1.4428 - val_accuracy: 0.
Epoch 26/30
2400/2400 [=====] - 4s 2ms/step - loss: 1.4458 - accuracy: 0.4859 - val_loss: 1.4346 - val_accuracy: 0.
Epoch 27/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.4447 - accuracy: 0.4853 - val_loss: 1.4338 - val_accuracy: 0.
Epoch 28/30
2400/2400 [=====] - 4s 2ms/step - loss: 1.4433 - accuracy: 0.4855 - val_loss: 1.4302 - val_accuracy: 0.
Epoch 29/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.4417 - accuracy: 0.4872 - val_loss: 1.4292 - val_accuracy: 0.
Epoch 30/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.4413 - accuracy: 0.4853 - val_loss: 1.4257 - val_accuracy: 0.
1500/1500 [=====] - 2s 2ms/step - loss: 1.4321 - accuracy: 0.4906
375/375 [=====] - 1s 2ms/step - loss: 1.4240 - accuracy: 0.4960
Train loss 1.4321058988571167
Train accuracy 0.4906249940395355
Validation loss 1.424037098845825
Validation accuracy 0.4959999918937683
```

```
1 # Model 4: 2 layers with 128 nodes and 16 nodes
2 model4 = Sequential()
3 model4.add(Dense(128, activation='relu', input_shape=(4,)))
4 model4.add(Dense(16, activation='relu'))
5 model4.add(Dense(10, activation='softmax'))
6 model4.compile(loss='categorical_crossentropy', optimizer=SGD(learning_rate=0.0001), metrics=['accuracy'])
7 model4.fit(X_train_features,y_train_categorical, epochs=30, batch_size=16, verbose=1, validation_split=0.2)
8 train_loss4, train_accuracy4 = model4.evaluate(X_train_features,y_train_categorical)
9 val_loss4, val_accuracy4 = model4.evaluate(X_val_features,y_val_categorical)
```

```

> val_loss4, val_accuracy4 = model4.evaluate(X_val_features,y_val_categorical)
10 print(f'Train loss {train_loss4}')
11 print(f'Train accuracy {train_accuracy4}')
12 print(f'Validation loss {val_loss4}')
13 print(f'Validation accuracy {val_accuracy4}')

2400/2400 [=====] - 5s 2ms/step - loss: 1.9303 - accuracy: 0.3173 - val_loss: 1.9164 - val_accuracy: 0.
Epoch 6/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.9004 - accuracy: 0.3253 - val_loss: 1.8867 - val_accuracy: 0.
Epoch 7/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.8725 - accuracy: 0.3332 - val_loss: 1.8599 - val_accuracy: 0.
Epoch 8/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.8493 - accuracy: 0.3398 - val_loss: 1.8407 - val_accuracy: 0.
Epoch 9/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.8302 - accuracy: 0.3446 - val_loss: 1.8188 - val_accuracy: 0.
Epoch 10/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.8131 - accuracy: 0.3481 - val_loss: 1.8028 - val_accuracy: 0.
Epoch 11/30
2400/2400 [=====] - 6s 2ms/step - loss: 1.7977 - accuracy: 0.3492 - val_loss: 1.7879 - val_accuracy: 0.
Epoch 12/30
2400/2400 [=====] - 4s 2ms/step - loss: 1.7835 - accuracy: 0.3499 - val_loss: 1.7735 - val_accuracy: 0.
Epoch 13/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.7697 - accuracy: 0.3514 - val_loss: 1.7596 - val_accuracy: 0.
Epoch 14/30
2400/2400 [=====] - 4s 2ms/step - loss: 1.7565 - accuracy: 0.3529 - val_loss: 1.7464 - val_accuracy: 0.
Epoch 15/30
2400/2400 [=====] - 4s 2ms/step - loss: 1.7433 - accuracy: 0.3553 - val_loss: 1.7334 - val_accuracy: 0.
Epoch 16/30
2400/2400 [=====] - 6s 2ms/step - loss: 1.7309 - accuracy: 0.3581 - val_loss: 1.7210 - val_accuracy: 0.
Epoch 17/30
2400/2400 [=====] - 4s 2ms/step - loss: 1.7175 - accuracy: 0.3667 - val_loss: 1.7069 - val_accuracy: 0.
Epoch 18/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.6991 - accuracy: 0.3954 - val_loss: 1.6826 - val_accuracy: 0.
Epoch 19/30
2400/2400 [=====] - 4s 2ms/step - loss: 1.6774 - accuracy: 0.4182 - val_loss: 1.6607 - val_accuracy: 0.
Epoch 20/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.6572 - accuracy: 0.4249 - val_loss: 1.6420 - val_accuracy: 0.
Epoch 21/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.6390 - accuracy: 0.4279 - val_loss: 1.6245 - val_accuracy: 0.
Epoch 22/30
2400/2400 [=====] - 4s 2ms/step - loss: 1.6228 - accuracy: 0.4287 - val_loss: 1.6116 - val_accuracy: 0.
Epoch 23/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.6093 - accuracy: 0.4280 - val_loss: 1.5963 - val_accuracy: 0.
Epoch 24/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.5975 - accuracy: 0.4301 - val_loss: 1.5915 - val_accuracy: 0.
Epoch 25/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.5874 - accuracy: 0.4292 - val_loss: 1.5759 - val_accuracy: 0.
Epoch 26/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.5795 - accuracy: 0.4321 - val_loss: 1.5686 - val_accuracy: 0.
Epoch 27/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.5714 - accuracy: 0.4324 - val_loss: 1.5614 - val_accuracy: 0.
Epoch 28/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.5653 - accuracy: 0.4353 - val_loss: 1.5586 - val_accuracy: 0.
Epoch 29/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.5592 - accuracy: 0.4380 - val_loss: 1.5489 - val_accuracy: 0.
Epoch 30/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.5538 - accuracy: 0.4407 - val_loss: 1.5468 - val_accuracy: 0.
1500/1500 [=====] - 2s 2ms/step - loss: 1.4321 - accuracy: 0.4906
375/375 [=====] - 1s 2ms/step - loss: 1.4240 - accuracy: 0.4960
Train loss 1.4321058988571167
Train accuracy 0.4906249940395355
Validation loss 1.424037098845825
Validation accuracy 0.4959999918937683

```

```

1 # Model 5: 2 layers with 128 nodes and 64 nodes
2 model5 = Sequential()
3 model5.add(Dense(128, activation='relu', input_shape=(4,)))
4 model5.add(Dense(64, activation='relu'))
5 model5.add(Dense(10, activation='softmax'))
6 model5.compile(loss='categorical_crossentropy', optimizer=SGD(learning_rate=0.0001), metrics=['accuracy'])
7 model5.fit(X_train_features,y_train_categorical, epochs=30, batch_size=16, verbose=1, validation_split=0.2)
8 train_loss5, train_accuracy5 = model5.evaluate(X_train_features,y_train_categorical)
9 val_loss5, val_accuracy5 = model5.evaluate(X_val_features,y_val_categorical)
10 print(f'Train loss {train_loss5}')
11 print(f'Train accuracy {train_accuracy5}')
12 print(f'Validation loss {val_loss5}')
13 print(f'Validation accuracy {val_accuracy5}')

```

```

Epoch 1/30
2400/2400 [=====] - 6s 2ms/step - loss: 2.0536 - accuracy: 0.3323 - val_loss: 1.6703 - val_accuracy: 0.
Epoch 2/30
2400/2400 [=====] - 6s 2ms/step - loss: 1.6186 - accuracy: 0.4359 - val_loss: 1.5708 - val_accuracy: 0.
Epoch 3/30
2400/2400 [=====] - 4s 2ms/step - loss: 1.5516 - accuracy: 0.4573 - val_loss: 1.5276 - val_accuracy: 0.
Epoch 4/30
2400/2400 [=====] - 5s 2ms/step - loss: 1.5208 - accuracy: 0.4631 - val_loss: 1.5000 - val_accuracy: 0.
Epoch 5/30

```

```
2400/2400 [=====] - 4s 2ms/step - loss: 1.5017 - accuracy: 0.4693 - val_loss: 1.4847 - val_accuracy: 0.  
Epoch 6/30  
2400/2400 [=====] - 5s 2ms/step - loss: 1.4887 - accuracy: 0.4733 - val_loss: 1.4847 - val_accuracy: 0.  
Epoch 7/30  
2400/2400 [=====] - 5s 2ms/step - loss: 1.4784 - accuracy: 0.4768 - val_loss: 1.4616 - val_accuracy: 0.  
Epoch 8/30  
2400/2400 [=====] - 4s 2ms/step - loss: 1.4694 - accuracy: 0.4785 - val_loss: 1.4692 - val_accuracy: 0.  
Epoch 9/30  
2400/2400 [=====] - 5s 2ms/step - loss: 1.4632 - accuracy: 0.4770 - val_loss: 1.4464 - val_accuracy: 0.  
Epoch 10/30  
2400/2400 [=====] - 5s 2ms/step - loss: 1.4570 - accuracy: 0.4814 - val_loss: 1.4443 - val_accuracy: 0.  
Epoch 11/30  
2400/2400 [=====] - 5s 2ms/step - loss: 1.4525 - accuracy: 0.4825 - val_loss: 1.4366 - val_accuracy: 0.  
Epoch 12/30  
2400/2400 [=====] - 4s 2ms/step - loss: 1.4463 - accuracy: 0.4830 - val_loss: 1.4507 - val_accuracy: 0.  
Epoch 13/30  
2400/2400 [=====] - 5s 2ms/step - loss: 1.4436 - accuracy: 0.4849 - val_loss: 1.4381 - val_accuracy: 0.  
Epoch 14/30  
2400/2400 [=====] - 5s 2ms/step - loss: 1.4392 - accuracy: 0.4839 - val_loss: 1.4276 - val_accuracy: 0.  
Epoch 15/30  
2400/2400 [=====] - 5s 2ms/step - loss: 1.4357 - accuracy: 0.4876 - val_loss: 1.4211 - val_accuracy: 0.  
Epoch 16/30  
2400/2400 [=====] - 5s 2ms/step - loss: 1.4332 - accuracy: 0.4866 - val_loss: 1.4394 - val_accuracy: 0.  
Epoch 17/30  
2400/2400 [=====] - 5s 2ms/step - loss: 1.4304 - accuracy: 0.4863 - val_loss: 1.4302 - val_accuracy: 0.  
Epoch 18/30  
2400/2400 [=====] - 4s 2ms/step - loss: 1.4269 - accuracy: 0.4878 - val_loss: 1.4249 - val_accuracy: 0.  
Epoch 19/30  
2400/2400 [=====] - 5s 2ms/step - loss: 1.4250 - accuracy: 0.4896 - val_loss: 1.4243 - val_accuracy: 0.  
Epoch 20/30  
2400/2400 [=====] - 4s 2ms/step - loss: 1.4233 - accuracy: 0.4886 - val_loss: 1.4091 - val_accuracy: 0.  
Epoch 21/30  
2400/2400 [=====] - 5s 2ms/step - loss: 1.4201 - accuracy: 0.4899 - val_loss: 1.4200 - val_accuracy: 0.  
Epoch 22/30  
2400/2400 [=====] - 5s 2ms/step - loss: 1.4181 - accuracy: 0.4896 - val_loss: 1.4078 - val_accuracy: 0.  
Epoch 23/30  
2400/2400 [=====] - 5s 2ms/step - loss: 1.4174 - accuracy: 0.4895 - val_loss: 1.4042 - val_accuracy: 0.  
Epoch 24/30  
2400/2400 [=====] - 5s 2ms/step - loss: 1.4154 - accuracy: 0.4911 - val_loss: 1.4067 - val_accuracy: 0.  
Epoch 25/30  
2400/2400 [=====] - 4s 2ms/step - loss: 1.4138 - accuracy: 0.4893 - val_loss: 1.4037 - val_accuracy: 0.  
Epoch 26/30  
2400/2400 [=====] - 5s 2ms/step - loss: 1.4120 - accuracy: 0.4930 - val_loss: 1.4019 - val_accuracy: 0.  
Epoch 27/30  
2400/2400 [=====] - 4s 2ms/step - loss: 1.4107 - accuracy: 0.4906 - val_loss: 1.4005 - val_accuracy: 0.  
Epoch 28/30  
2400/2400 [=====] - 5s 2ms/step - loss: 1.4084 - accuracy: 0.4929 - val_loss: 1.4013 - val_accuracy: 0.
```