Assignment 2: Build a CNN for image recognition.

Due Date: March 29, 11:59PM

Name: Surya Giri

CWID: 10475010

Introduction:

1. In this assignment, you will build Convolutional Neural Network to classify CIFAR-10 Images.
2. You can directly load dataset from many deep learning packages.
3. You can use any deep learning packages such as pytorch, keras or tensorflow for this assignment.

Requirements:

1. You need to load cifar 10 data and split the entire training dataset into training and validation.
2. You will implement a CNN model to classify cifar 10 images with provided structure.
3. You need to plot the training and validation accuracy or loss obtained from above step.
4. Then you can use tuned parameters to train using the entire training dataset.
5. You should report the testing accuracy using the model with complete data.
6. You may try to change the structure (e.g, add BN layer or dropout layer,...) and analyze your findings.

Google Colab

- If you do not have GPU, the training of a CNN can be slow. Google Colab is a good option.

Batch Normalization (BN)

Background:

- Batch Normalization is a technique to speed up training and help make the model more stable.
- In simple words, batch normalization is just another network layer that gets inserted between a hidden layer and the next hidden layer. Its job is to take the outputs from the first hidden layer and normalize them before passing them on as the input of the next hidden layer.
- For more detailed information, you may refer to the original paper: https://arxiv.org/pdf/1502.03167.pdf (https://arxiv.org/pdf/1502.03167.pdf).

BN Algorithm:

- Input: Values of x over a mini-batch: $\mathbf{B}$ = $\{x_1, \ldots, x_m\}$;
- Output: $\{y_i = BN_{\gamma,\beta}(x_i)\}$, γ, β are learnable parameters

Normalization of the Input:

$$\mu_{\mathbf{B}} = \frac{1}{m}\sum_{i=1}^{m} x_i$$

$$\sigma_{\mathbf{B}}^2 = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_{\mathbf{B}})^2$$

$$\hat{x}_i = \frac{x_i - \mu_{\mathbf{B}}}{\sqrt{\sigma_{\mathbf{B}}{}^2 + \epsilon}}$$

Re-scaling and Offsetting:

$$y_i = \gamma \hat{x}_i + \beta = BN_{\gamma,\beta}(x_i)$$

Advantages of BN:

1. Improves gradient flow through the network.
2. Allows use of saturating nonlinearities and higher learning rates.
3. Makes weights easier to initialize.
4. Act as a form of regularization and may reduce the need for dropout.

Implementation:

- The batch normalization layer has already been implemented in many packages. You may simply call the function to build the layer. For example: torch.nn.BatchNorm2d() using pytroch package, keras.layers.BatchNormalization() using keras package.
- The location of BN layer: Please make sure `BatchNormalization` is between a `Conv` / `Dense` layer and an `activation` layer.

1. Data preparation

1.1. Load data

In [1]:
```
# Load Cifar-10 Data
# This is just an example, you may load dataset from other packages.
import keras
import tensorflow as tf
import numpy as np

### If you can not load keras dataset, un-comment these two lines.
#import ssl
#ssl._create_default_https_context = ssl._create_unverified_context

(x_train, y_train), (x_test, y_test) = tf.keras.datasets.cifar10.load_data()

print('shape of x_train: ' + str(x_train.shape))
print('shape of y_train: ' + str(y_train.shape))
print('shape of x_test: ' + str(x_test.shape))
print('shape of y_test: ' + str(y_test.shape))
print('number of classes: ' + str(np.max(y_train) - np.min(y_train) + 1))
```

```
shape of x_train: (50000, 32, 32, 3)
shape of y_train: (50000, 1)
shape of x_test: (10000, 32, 32, 3)
shape of y_test: (10000, 1)
number of classes: 10
```

1.2. One-hot encode the labels (5 points)

In the input, a label is a scalar in $\{0, 1, \cdots, 9\}$. One-hot encode transform such a scalar to a 10-dim vector. E.g., a scalar `y_train[j]=3` is transformed to the vector `y_train_vec[j]=[0, 0, 0, 1, 0, 0, 0, 0, 0, 0]`.

1. Implement a function `to_one_hot` that transforms an $n \times 1$ array to a $n \times 10$ matrix.
2. Apply the function to `y_train` and `y_test`.

In [2]:
```
def to_one_hot(y, num_class=10):
  coded = np.zeros((len(y), num_class))
  for i, y in enumerate(y):
    coded[i, y] = 1
  return coded

y_train_vec = to_one_hot(y_train)
y_test_vec = to_one_hot(y_test)

print('Shape of y_train_vec: ' + str(y_train_vec.shape))
print('Shape of y_test_vec: ' + str(y_test_vec.shape))

print(y_train[0])
print(y_train_vec[0])
```

```
Shape of y_train_vec: (50000, 10)
Shape of y_test_vec: (10000, 10)
[6]
[0. 0. 0. 0. 0. 0. 1. 0. 0. 0.]
```

Remark: the outputs should be

- Shape of y_train_vec: (50000, 10)
- Shape of y_test_vec: (10000, 10)
- [6]
- [0. 0. 0. 0. 0. 0. 1. 0. 0. 0.]

1.3. Randomly partition the training set to training and validation sets (5 points)

Randomly partition the 50K training samples to 2 sets:

- a training set containing 40K samples: x_tr, y_tr

- a validation set containing 10K samples: x_val, y_val

In [3]:
```
rand_indices = np.random.permutation(50000)
train_indices = rand_indices[0:40000]
valid_indices = rand_indices[40000:50000]

x_val = x_train[valid_indices, :]
y_val = y_train_vec[valid_indices, :]

x_tr = x_train[train_indices, :]
y_tr = y_train_vec[train_indices, :]

print('Shape of x_tr: ' + str(x_tr.shape))
print('Shape of y_tr: ' + str(y_tr.shape))
print('Shape of x_val: ' + str(x_val.shape))
print('Shape of y_val: ' + str(y_val.shape))
```

```
Shape of x_tr: (40000, 32, 32, 3)
Shape of y_tr: (40000, 10)
Shape of x_val: (10000, 32, 32, 3)
Shape of y_val: (10000, 10)
```

2. Build a CNN and tune its hyper-parameters (50 points)

- Build a convolutional neural network model using the below structure:
- It should have a structure of: Conv - ReLU - Max Pool - ConV - ReLU - Max Pool - Dense - ReLU - Dense - Softmax
- In the graph 3@32x32 means the dimension of input image, 32@30x30 means it has 32 filters and the dimension now becomes 30x30 after the convolution.
- All convolutional layers (Conv) should have stride = 1 and no padding.
- Max Pooling has a pool size of 2 by 2.

- You may use the validation data to tune the hyper-parameters (e.g., learning rate, and optimization algorithm)
- Do NOT use test data for hyper-parameter tuning!!!
- Try to achieve a validation accuracy as high as possible.

In [4]:
```
from keras.layers import Dense
from keras.layers.pooling import MaxPooling2D
from keras.layers.convolutional import Conv2D
from keras import models, layers

model_cnn = models.Sequential()
model_cnn.add(layers.Conv2D(32, (3, 3), activation='relu', input_shape=(32, 32, 3)))
model_cnn.add(layers.MaxPooling2D((2, 2)))
model_cnn.add(layers.Conv2D(64,(4,4), activation='relu'))
model_cnn.add(layers.MaxPooling2D((2, 2)))
model_cnn.add(layers.Flatten())
model_cnn.add(layers.Dense(256, activation='relu'))
model_cnn.add(layers.Dense(10, activation='softmax'))
```

In [5]:
```
# Checking if the model is built correctly
model_cnn.summary()
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 30, 30, 32)        896
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 15, 15, 32)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 12, 12, 64)        32832
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 6, 6, 64)          0
_________________________________________________________________
flatten (Flatten)            (None, 2304)              0
_________________________________________________________________
dense (Dense)                (None, 256)               590080
_________________________________________________________________
dense_1 (Dense)              (None, 10)                2570
=================================================================
Total params: 626,378
Trainable params: 626,378
Non-trainable params: 0
_________________________________________________________________
```

In [6]:
```
# Define model optimizer and loss function
from keras import optimizers
model_cnn.compile(optimizer='Adadelta', loss='categorical_crossentropy', metrics=['accuracy'])
```

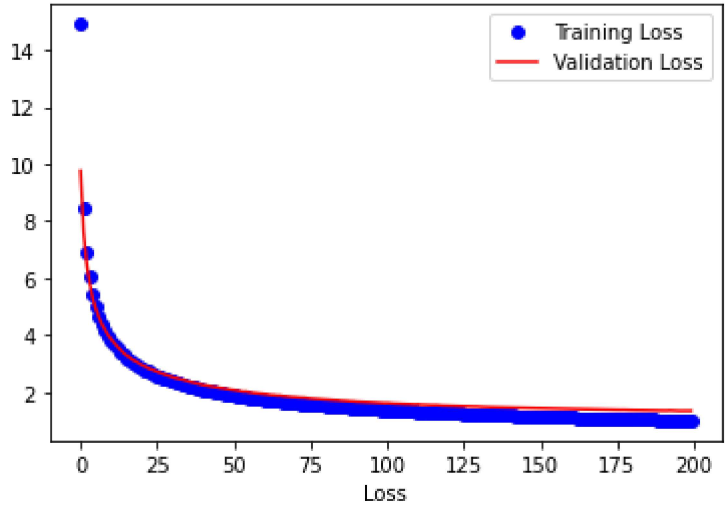
In [7]:
```
# Train the model and store model parameters/loss values
history = model_cnn.fit(x_tr, y_tr, batch_size=16, epochs=200, validation_data=(x_val, y_val))
```

```
30 - val_accuracy: 0.5108
Epoch 120/200
2500/2500 [==============================] - 4s 2ms/step - loss: 1.2827 - accuracy: 0.5783 - val_loss: 1.54
11 - val_accuracy: 0.5135
Epoch 121/200
2500/2500 [==============================] - 4s 2ms/step - loss: 1.2778 - accuracy: 0.5799 - val_loss: 1.54
12 - val_accuracy: 0.5102
Epoch 122/200
2500/2500 [==============================] - 4s 2ms/step - loss: 1.2736 - accuracy: 0.5799 - val_loss: 1.53
11 - val_accuracy: 0.5132
Epoch 123/200
2500/2500 [==============================] - 4s 2ms/step - loss: 1.2685 - accuracy: 0.5817 - val_loss: 1.53
79 - val_accuracy: 0.5124
Epoch 124/200
2500/2500 [==============================] - 4s 2ms/step - loss: 1.2636 - accuracy: 0.5827 - val_loss: 1.52
79 - val_accuracy: 0.5136
Epoch 125/200
2500/2500 [==============================] - 4s 2ms/step - loss: 1.2593 - accuracy: 0.5837 - val_loss: 1.52
40 - val_accuracy: 0.5148
Epoch 126/200
```

3. Plot the training and validation loss curve versus epochs. (5 points)

In [8]:
```
# Plot the loss curve
import matplotlib.pyplot as plt
%matplotlib inline

epochs = range(200)
train_acc = history.history['loss']
valid_acc = history.history['val_loss']
plt.plot(epochs, train_acc, 'bo', label='Training Loss')
plt.plot(epochs, valid_acc, 'r', label='Validation Loss')
plt.xlabel('Epochs')
plt.xlabel('Loss')
plt.legend()
plt.show()
```



4. Train (again) and evaluate the model (5 points)

- To this end, you have found the "best" hyper-parameters.
- Now, fix the hyper-parameters and train the network on the entire training set (all the 50K training samples)
- Evaluate your model on the test set.

Train the model on the entire training set

Why? Previously, you used 40K samples for training; you wasted 10K samples for the sake of hyper-parameter tuning. Now you already know the hyper-parameters, so why not using all the 50K samples for training?

In [9]:
```
#<Compile your model again (using the same hyper-parameters you tuned above)>
from keras import optimizers
model_cnn.compile(optimizer='Adadelta', loss='categorical_crossentropy', metrics=['accuracy'])
```

In [10]:
```
# Evaluate your model performance (testing accuracy) on testing data.
history = model_cnn.fit(x_train, y_train_vec, batch_size=16, epochs=200, validation_data = (x_test, y_test_vec))
```

```
Epoch 1/200
3125/3125 [==============================] - 6s 2ms/step - loss: 1.0849 - accuracy: 0.6387 - val_loss: 1.34
25 - val_accuracy: 0.5675
Epoch 2/200
3125/3125 [==============================] - 5s 2ms/step - loss: 1.0812 - accuracy: 0.6399 - val_loss: 1.34
30 - val_accuracy: 0.5680
Epoch 3/200
3125/3125 [==============================] - 5s 2ms/step - loss: 1.0774 - accuracy: 0.6417 - val_loss: 1.33
40 - val_accuracy: 0.5687
Epoch 4/200
3125/3125 [==============================] - 6s 2ms/step - loss: 1.0744 - accuracy: 0.6425 - val_loss: 1.33
44 - val_accuracy: 0.5680
Epoch 5/200
3125/3125 [==============================] - 5s 2ms/step - loss: 1.0709 - accuracy: 0.6430 - val_loss: 1.33
68 - val_accuracy: 0.5660
Epoch 6/200
3125/3125 [==============================] - 6s 2ms/step - loss: 1.0682 - accuracy: 0.6438 - val_loss: 1.32
96 - val_accuracy: 0.5727
Epoch 7/200
3125/3125 [==============================] - 5s 2ms/step - loss: 1.0643 - accuracy: 0.6460 - val_loss: 1.33
37 - val_accuracy: 0.5669
Epoch 8/200
3125/3125 [==============================] - 5s 2ms/step - loss: 1.0618 - accuracy: 0.6455 - val_loss: 1.32
67 - val_accuracy: 0.5731
Epoch 9/200
3125/3125 [==============================] - 5s 2ms/step - loss: 1.0590 - accuracy: 0.6470 - val_loss: 1.32
37 - val_accuracy: 0.5715
Epoch 10/200
3125/3125 [==============================] - 5s 2ms/step - loss: 1.0557 - accuracy: 0.6476 - val_loss: 1.32
17 - val_accuracy: 0.5728
Epoch 11/200
3125/3125 [==============================] - 6s 2ms/step - loss: 1.0525 - accuracy: 0.6489 - val_loss: 1.31
78 - val_accuracy: 0.5728
Epoch 12/200
3125/3125 [==============================] - 6s 2ms/step - loss: 1.0500 - accuracy: 0.6504 - val_loss: 1.31
59 - val_accuracy: 0.5743
Epoch 13/200
3125/3125 [==============================] - 5s 2ms/step - loss: 1.0469 - accuracy: 0.6508 - val_loss: 1.31
80 - val_accuracy: 0.5719
```

```
Epoch 14/200
3125/3125 [==============================] - 6s 2ms/step - loss: 1.0443 - accuracy: 0.6514 - val_loss: 1.31
30 - val_accuracy: 0.5769
Epoch 15/200
3125/3125 [==============================] - 5s 2ms/step - loss: 1.0410 - accuracy: 0.6527 - val_loss: 1.31
90 - val_accuracy: 0.5764
Epoch 16/200
3125/3125 [==============================] - 5s 2ms/step - loss: 1.0390 - accuracy: 0.6538 - val_loss: 1.31
16 - val_accuracy: 0.5726
Epoch 17/200
3125/3125 [==============================] - 5s 2ms/step - loss: 1.0361 - accuracy: 0.6554 - val_loss: 1.31
20 - val_accuracy: 0.5772
Epoch 18/200
3125/3125 [==============================] - 5s 2ms/step - loss: 1.0333 - accuracy: 0.6556 - val_loss: 1.30
70 - val_accuracy: 0.5735
Epoch 19/200
3125/3125 [==============================] - 5s 2ms/step - loss: 1.0305 - accuracy: 0.6563 - val_loss: 1.30
35 - val_accuracy: 0.5762
Epoch 20/200
3125/3125 [==============================] - 5s 2ms/step - loss: 1.0281 - accuracy: 0.6562 - val_loss: 1.30
02 - val_accuracy: 0.5777
Epoch 21/200
3125/3125 [==============================] - 5s 2ms/step - loss: 1.0252 - accuracy: 0.6576 - val_loss: 1.30
22 - val_accuracy: 0.5783
Epoch 22/200
3125/3125 [==============================] - 5s 2ms/step - loss: 1.0230 - accuracy: 0.6574 - val_loss: 1.30
00 - val_accuracy: 0.5742
Epoch 23/200
3125/3125 [==============================] - 5s 2ms/step - loss: 1.0196 - accuracy: 0.6599 - val_loss: 1.30
31 - val_accuracy: 0.5798
Epoch 24/200
3125/3125 [==============================] - 5s 2ms/step - loss: 1.0176 - accuracy: 0.6599 - val_loss: 1.29
64 - val_accuracy: 0.5812
Epoch 25/200
3125/3125 [==============================] - 5s 2ms/step - loss: 1.0155 - accuracy: 0.6612 - val_loss: 1.29
82 - val_accuracy: 0.5782
Epoch 26/200
3125/3125 [==============================] - 5s 2ms/step - loss: 1.0126 - accuracy: 0.6613 - val_loss: 1.29
24 - val_accuracy: 0.5811
Epoch 27/200
3125/3125 [==============================] - 5s 2ms/step - loss: 1.0100 - accuracy: 0.6626 - val_loss: 1.29
61 - val_accuracy: 0.5818
Epoch 28/200
```

```
3125/3125 [==============================] - 5s 2ms/step - loss: 1.0081 - accuracy: 0.6623 - val_loss: 1.29
04 - val_accuracy: 0.5835
Epoch 29/200
3125/3125 [==============================] - 5s 2ms/step - loss: 1.0054 - accuracy: 0.6632 - val_loss: 1.28
94 - val_accuracy: 0.5808
Epoch 30/200
3125/3125 [==============================] - 5s 2ms/step - loss: 1.0031 - accuracy: 0.6645 - val_loss: 1.28
94 - val_accuracy: 0.5801
Epoch 31/200
3125/3125 [==============================] - 5s 2ms/step - loss: 1.0001 - accuracy: 0.6652 - val_loss: 1.28
57 - val_accuracy: 0.5823
Epoch 32/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.9983 - accuracy: 0.6657 - val_loss: 1.28
56 - val_accuracy: 0.5818
Epoch 33/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.9956 - accuracy: 0.6688 - val_loss: 1.28
18 - val_accuracy: 0.5819
Epoch 34/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.9936 - accuracy: 0.6676 - val_loss: 1.28
17 - val_accuracy: 0.5812
Epoch 35/200
3125/3125 [==============================] - 6s 2ms/step - loss: 0.9912 - accuracy: 0.6684 - val_loss: 1.28
25 - val_accuracy: 0.5838
Epoch 36/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.9892 - accuracy: 0.6692 - val_loss: 1.27
87 - val_accuracy: 0.5843
Epoch 37/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.9865 - accuracy: 0.6700 - val_loss: 1.28
06 - val_accuracy: 0.5868
Epoch 38/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.9844 - accuracy: 0.6701 - val_loss: 1.27
88 - val_accuracy: 0.5844
Epoch 39/200
3125/3125 [==============================] - 6s 2ms/step - loss: 0.9823 - accuracy: 0.6712 - val_loss: 1.27
98 - val_accuracy: 0.5822
Epoch 40/200
3125/3125 [==============================] - 6s 2ms/step - loss: 0.9799 - accuracy: 0.6714 - val_loss: 1.27
87 - val_accuracy: 0.5845
Epoch 41/200
3125/3125 [==============================] - 6s 2ms/step - loss: 0.9776 - accuracy: 0.6719 - val_loss: 1.27
36 - val_accuracy: 0.5867
Epoch 42/200
3125/3125 [==============================] - 6s 2ms/step - loss: 0.9756 - accuracy: 0.6740 - val_loss: 1.27
```

```
42 - val_accuracy: 0.5858
Epoch 43/200
3125/3125 [==============================] - 6s 2ms/step - loss: 0.9730 - accuracy: 0.6738 - val_loss: 1.27
86 - val_accuracy: 0.5835
Epoch 44/200
3125/3125 [==============================] - 6s 2ms/step - loss: 0.9713 - accuracy: 0.6743 - val_loss: 1.27
03 - val_accuracy: 0.5897
Epoch 45/200
3125/3125 [==============================] - 6s 2ms/step - loss: 0.9689 - accuracy: 0.6754 - val_loss: 1.26
99 - val_accuracy: 0.5864
Epoch 46/200
3125/3125 [==============================] - 6s 2ms/step - loss: 0.9672 - accuracy: 0.6759 - val_loss: 1.26
64 - val_accuracy: 0.5882
Epoch 47/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.9647 - accuracy: 0.6756 - val_loss: 1.26
92 - val_accuracy: 0.5872
Epoch 48/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.9628 - accuracy: 0.6780 - val_loss: 1.26
73 - val_accuracy: 0.5881
Epoch 49/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.9606 - accuracy: 0.6790 - val_loss: 1.26
86 - val_accuracy: 0.5859
Epoch 50/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.9586 - accuracy: 0.6786 - val_loss: 1.26
36 - val_accuracy: 0.5881
Epoch 51/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.9565 - accuracy: 0.6805 - val_loss: 1.26
24 - val_accuracy: 0.5902
Epoch 52/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.9544 - accuracy: 0.6804 - val_loss: 1.26
36 - val_accuracy: 0.5870
Epoch 53/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.9517 - accuracy: 0.6807 - val_loss: 1.26
14 - val_accuracy: 0.5924
Epoch 54/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.9505 - accuracy: 0.6825 - val_loss: 1.25
86 - val_accuracy: 0.5908
Epoch 55/200
3125/3125 [==============================] - 6s 2ms/step - loss: 0.9484 - accuracy: 0.6825 - val_loss: 1.26
17 - val_accuracy: 0.5898
Epoch 56/200
3125/3125 [==============================] - 6s 2ms/step - loss: 0.9465 - accuracy: 0.6829 - val_loss: 1.25
```

```
88 - val_accuracy: 0.5918
Epoch 57/200

3125/3125 [==============================] - 6s 2ms/step - loss: 0.9444 - accuracy: 0.6833 - val_loss: 1.25
56 - val_accuracy: 0.5902
Epoch 58/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.9423 - accuracy: 0.6841 - val_loss: 1.25
62 - val_accuracy: 0.5922
Epoch 59/200
3125/3125 [==============================] - 6s 2ms/step - loss: 0.9400 - accuracy: 0.6851 - val_loss: 1.25
86 - val_accuracy: 0.5926
Epoch 60/200
3125/3125 [==============================] - 6s 2ms/step - loss: 0.9384 - accuracy: 0.6852 - val_loss: 1.26
04 - val_accuracy: 0.5942
Epoch 61/200
3125/3125 [==============================] - 6s 2ms/step - loss: 0.9365 - accuracy: 0.6868 - val_loss: 1.25
73 - val_accuracy: 0.5885
Epoch 62/200
3125/3125 [==============================] - 6s 2ms/step - loss: 0.9347 - accuracy: 0.6865 - val_loss: 1.25
47 - val_accuracy: 0.5913
Epoch 63/200
3125/3125 [==============================] - 6s 2ms/step - loss: 0.9325 - accuracy: 0.6870 - val_loss: 1.25
08 - val_accuracy: 0.5911
Epoch 64/200
3125/3125 [==============================] - 6s 2ms/step - loss: 0.9308 - accuracy: 0.6877 - val_loss: 1.25
05 - val_accuracy: 0.5942
Epoch 65/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.9289 - accuracy: 0.6873 - val_loss: 1.25
31 - val_accuracy: 0.5915
Epoch 66/200
3125/3125 [==============================] - 6s 2ms/step - loss: 0.9264 - accuracy: 0.6902 - val_loss: 1.25
09 - val_accuracy: 0.5929
Epoch 67/200
3125/3125 [==============================] - 6s 2ms/step - loss: 0.9252 - accuracy: 0.6901 - val_loss: 1.24
49 - val_accuracy: 0.5971
Epoch 68/200
3125/3125 [==============================] - 6s 2ms/step - loss: 0.9231 - accuracy: 0.6898 - val_loss: 1.24
89 - val_accuracy: 0.5970
Epoch 69/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.9214 - accuracy: 0.6893 - val_loss: 1.24
64 - val_accuracy: 0.5960
Epoch 70/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.9193 - accuracy: 0.6910 - val_loss: 1.24
```

```
29 - val_accuracy: 0.5990
Epoch 71/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.9177 - accuracy: 0.6913 - val_loss: 1.24
40 - val_accuracy: 0.5951
Epoch 72/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.9159 - accuracy: 0.6923 - val_loss: 1.24
45 - val_accuracy: 0.5940
Epoch 73/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.9140 - accuracy: 0.6926 - val_loss: 1.24
07 - val_accuracy: 0.5961
Epoch 74/200
3125/3125 [==============================] - 6s 2ms/step - loss: 0.9126 - accuracy: 0.6934 - val_loss: 1.24
09 - val_accuracy: 0.5954
Epoch 75/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.9107 - accuracy: 0.6944 - val_loss: 1.24
09 - val_accuracy: 0.5960
Epoch 76/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.9087 - accuracy: 0.6953 - val_loss: 1.24
16 - val_accuracy: 0.5976
Epoch 77/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.9072 - accuracy: 0.6951 - val_loss: 1.23
69 - val_accuracy: 0.5982
Epoch 78/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.9052 - accuracy: 0.6963 - val_loss: 1.24
29 - val_accuracy: 0.5968
Epoch 79/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.9036 - accuracy: 0.6977 - val_loss: 1.23
75 - val_accuracy: 0.5984
Epoch 80/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.9019 - accuracy: 0.6974 - val_loss: 1.23
67 - val_accuracy: 0.5977
Epoch 81/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.8998 - accuracy: 0.6978 - val_loss: 1.23
72 - val_accuracy: 0.6015
Epoch 82/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.8981 - accuracy: 0.6987 - val_loss: 1.24
57 - val_accuracy: 0.5989
Epoch 83/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.8966 - accuracy: 0.7001 - val_loss: 1.23
69 - val_accuracy: 0.5961
Epoch 84/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.8946 - accuracy: 0.7006 - val_loss: 1.23
14 - val_accuracy: 0.6009
```

```
Epoch 85/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.8931 - accuracy: 0.7004 - val_loss: 1.23
23 - val_accuracy: 0.6035
Epoch 86/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.8916 - accuracy: 0.7003 - val_loss: 1.22
99 - val_accuracy: 0.6020
Epoch 87/200
3125/3125 [==============================] - 6s 2ms/step - loss: 0.8900 - accuracy: 0.7005 - val_loss: 1.23
00 - val_accuracy: 0.6037
Epoch 88/200
3125/3125 [==============================] - 6s 2ms/step - loss: 0.8877 - accuracy: 0.7022 - val_loss: 1.22
94 - val_accuracy: 0.6002
Epoch 89/200
3125/3125 [==============================] - 6s 2ms/step - loss: 0.8867 - accuracy: 0.7019 - val_loss: 1.23
43 - val_accuracy: 0.6003
Epoch 90/200
3125/3125 [==============================] - 6s 2ms/step - loss: 0.8847 - accuracy: 0.7028 - val_loss: 1.23
23 - val_accuracy: 0.5989
Epoch 91/200
3125/3125 [==============================] - 6s 2ms/step - loss: 0.8831 - accuracy: 0.7038 - val_loss: 1.22
82 - val_accuracy: 0.6017
Epoch 92/200
3125/3125 [==============================] - 6s 2ms/step - loss: 0.8816 - accuracy: 0.7040 - val_loss: 1.22
94 - val_accuracy: 0.6023
Epoch 93/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.8798 - accuracy: 0.7034 - val_loss: 1.23
08 - val_accuracy: 0.5999
Epoch 94/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.8781 - accuracy: 0.7053 - val_loss: 1.22
89 - val_accuracy: 0.6041
Epoch 95/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.8770 - accuracy: 0.7061 - val_loss: 1.22
88 - val_accuracy: 0.6036
Epoch 96/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.8748 - accuracy: 0.7063 - val_loss: 1.22
32 - val_accuracy: 0.6015
Epoch 97/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.8735 - accuracy: 0.7080 - val_loss: 1.22
28 - val_accuracy: 0.6025
Epoch 98/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.8717 - accuracy: 0.7070 - val_loss: 1.22
64 - val_accuracy: 0.6020
Epoch 99/200
```

```
3125/3125 [==============================] - 5s 2ms/step - loss: 0.8701 - accuracy: 0.7084 - val_loss: 1.22
51 - val_accuracy: 0.6027
Epoch 100/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.8687 - accuracy: 0.7097 - val_loss: 1.22
62 - val_accuracy: 0.6039
Epoch 101/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.8670 - accuracy: 0.7094 - val_loss: 1.22
82 - val_accuracy: 0.6038
Epoch 102/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.8656 - accuracy: 0.7094 - val_loss: 1.22
51 - val_accuracy: 0.6039
Epoch 103/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.8641 - accuracy: 0.7104 - val_loss: 1.21
93 - val_accuracy: 0.6049
Epoch 104/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.8620 - accuracy: 0.7120 - val_loss: 1.21
86 - val_accuracy: 0.6040
Epoch 105/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.8606 - accuracy: 0.7121 - val_loss: 1.22
05 - val_accuracy: 0.6068
Epoch 106/200
3125/3125 [==============================] - 6s 2ms/step - loss: 0.8595 - accuracy: 0.7121 - val_loss: 1.22
12 - val_accuracy: 0.6037
Epoch 107/200
3125/3125 [==============================] - 6s 2ms/step - loss: 0.8576 - accuracy: 0.7135 - val_loss: 1.22
17 - val_accuracy: 0.6041
Epoch 108/200
3125/3125 [==============================] - 6s 2ms/step - loss: 0.8563 - accuracy: 0.7139 - val_loss: 1.21
54 - val_accuracy: 0.6050
Epoch 109/200
3125/3125 [==============================] - 6s 2ms/step - loss: 0.8546 - accuracy: 0.7126 - val_loss: 1.21
80 - val_accuracy: 0.6055
Epoch 110/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.8531 - accuracy: 0.7145 - val_loss: 1.21
60 - val_accuracy: 0.6060
Epoch 111/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.8517 - accuracy: 0.7151 - val_loss: 1.21
44 - val_accuracy: 0.6075
Epoch 112/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.8501 - accuracy: 0.7148 - val_loss: 1.21
75 - val_accuracy: 0.6043
Epoch 113/200
```

```
3125/3125 [==============================] - 5s 2ms/step - loss: 0.8484 - accuracy: 0.7154 - val_loss: 1.21
51 - val_accuracy: 0.6074
Epoch 114/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.8472 - accuracy: 0.7165 - val_loss: 1.21
50 - val_accuracy: 0.6073
Epoch 115/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.8458 - accuracy: 0.7172 - val_loss: 1.21
39 - val_accuracy: 0.6112
Epoch 116/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.8445 - accuracy: 0.7180 - val_loss: 1.21
21 - val_accuracy: 0.6074
Epoch 117/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.8426 - accuracy: 0.7174 - val_loss: 1.21
26 - val_accuracy: 0.6077
Epoch 118/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.8415 - accuracy: 0.7187 - val_loss: 1.21
05 - val_accuracy: 0.6096
Epoch 119/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.8397 - accuracy: 0.7192 - val_loss: 1.21
16 - val_accuracy: 0.6073
Epoch 120/200
3125/3125 [==============================] - 6s 2ms/step - loss: 0.8384 - accuracy: 0.7187 - val_loss: 1.21
02 - val_accuracy: 0.6090
Epoch 121/200
3125/3125 [==============================] - 6s 2ms/step - loss: 0.8373 - accuracy: 0.7199 - val_loss: 1.21
02 - val_accuracy: 0.6100
Epoch 122/200
3125/3125 [==============================] - 6s 2ms/step - loss: 0.8355 - accuracy: 0.7204 - val_loss: 1.21
14 - val_accuracy: 0.6097
Epoch 123/200
3125/3125 [==============================] - 6s 2ms/step - loss: 0.8341 - accuracy: 0.7207 - val_loss: 1.21
03 - val_accuracy: 0.6087
Epoch 124/200
3125/3125 [==============================] - 6s 2ms/step - loss: 0.8327 - accuracy: 0.7216 - val_loss: 1.20
89 - val_accuracy: 0.6097
Epoch 125/200
3125/3125 [==============================] - 6s 2ms/step - loss: 0.8313 - accuracy: 0.7223 - val_loss: 1.20
81 - val_accuracy: 0.6105
Epoch 126/200
3125/3125 [==============================] - 6s 2ms/step - loss: 0.8301 - accuracy: 0.7238 - val_loss: 1.21
01 - val_accuracy: 0.6094
Epoch 127/200
3125/3125 [==============================] - 6s 2ms/step - loss: 0.8284 - accuracy: 0.7241 - val_loss: 1.20
```

```
90 - val_accuracy: 0.6122
Epoch 128/200
3125/3125 [==============================] - 6s 2ms/step - loss: 0.8269 - accuracy: 0.7236 - val_loss: 1.20
87 - val_accuracy: 0.6112
Epoch 129/200
3125/3125 [==============================] - 6s 2ms/step - loss: 0.8260 - accuracy: 0.7239 - val_loss: 1.20
57 - val_accuracy: 0.6109
Epoch 130/200
3125/3125 [==============================] - 6s 2ms/step - loss: 0.8245 - accuracy: 0.7241 - val_loss: 1.20
76 - val_accuracy: 0.6110
Epoch 131/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.8228 - accuracy: 0.7254 - val_loss: 1.20
55 - val_accuracy: 0.6113
Epoch 132/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.8215 - accuracy: 0.7259 - val_loss: 1.20
50 - val_accuracy: 0.6113
Epoch 133/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.8200 - accuracy: 0.7261 - val_loss: 1.20
51 - val_accuracy: 0.6124
Epoch 134/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.8188 - accuracy: 0.7272 - val_loss: 1.20
28 - val_accuracy: 0.6118
Epoch 135/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.8177 - accuracy: 0.7277 - val_loss: 1.20
44 - val_accuracy: 0.6108
Epoch 136/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.8157 - accuracy: 0.7279 - val_loss: 1.21
07 - val_accuracy: 0.6105
Epoch 137/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.8147 - accuracy: 0.7276 - val_loss: 1.20
20 - val_accuracy: 0.6124
Epoch 138/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.8132 - accuracy: 0.7288 - val_loss: 1.20
16 - val_accuracy: 0.6133
Epoch 139/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.8122 - accuracy: 0.7285 - val_loss: 1.20
10 - val_accuracy: 0.6135
Epoch 140/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.8106 - accuracy: 0.7302 - val_loss: 1.20
04 - val_accuracy: 0.6138
Epoch 141/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.8092 - accuracy: 0.7306 - val_loss: 1.20
47 - val_accuracy: 0.6125
```

```
Epoch 142/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.8079 - accuracy: 0.7305 - val_loss: 1.20
18 - val_accuracy: 0.6135
Epoch 143/200
3125/3125 [==============================] - 6s 2ms/step - loss: 0.8063 - accuracy: 0.7316 - val_loss: 1.19
86 - val_accuracy: 0.6148
Epoch 144/200
3125/3125 [==============================] - 6s 2ms/step - loss: 0.8051 - accuracy: 0.7314 - val_loss: 1.20
05 - val_accuracy: 0.6132
Epoch 145/200
3125/3125 [==============================] - 6s 2ms/step - loss: 0.8039 - accuracy: 0.7324 - val_loss: 1.20
28 - val_accuracy: 0.6143
Epoch 146/200
3125/3125 [==============================] - 6s 2ms/step - loss: 0.8024 - accuracy: 0.7327 - val_loss: 1.20
06 - val_accuracy: 0.6145
Epoch 147/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.8013 - accuracy: 0.7334 - val_loss: 1.19
78 - val_accuracy: 0.6159
Epoch 148/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.7999 - accuracy: 0.7328 - val_loss: 1.20
74 - val_accuracy: 0.6136
Epoch 149/200
3125/3125 [==============================] - 6s 2ms/step - loss: 0.7988 - accuracy: 0.7338 - val_loss: 1.20
17 - val_accuracy: 0.6147
Epoch 150/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.7974 - accuracy: 0.7350 - val_loss: 1.19
78 - val_accuracy: 0.6160
Epoch 151/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.7962 - accuracy: 0.7343 - val_loss: 1.19
64 - val_accuracy: 0.6168
Epoch 152/200
3125/3125 [==============================] - 6s 2ms/step - loss: 0.7944 - accuracy: 0.7358 - val_loss: 1.20
63 - val_accuracy: 0.6132
Epoch 153/200
3125/3125 [==============================] - 6s 2ms/step - loss: 0.7937 - accuracy: 0.7349 - val_loss: 1.19
79 - val_accuracy: 0.6144
Epoch 154/200
3125/3125 [==============================] - 6s 2ms/step - loss: 0.7923 - accuracy: 0.7365 - val_loss: 1.19
74 - val_accuracy: 0.6147
Epoch 155/200
3125/3125 [==============================] - 6s 2ms/step - loss: 0.7911 - accuracy: 0.7369 - val_loss: 1.19
57 - val_accuracy: 0.6173
Epoch 156/200
```

```
3125/3125 [==============================] - 5s 2ms/step - loss: 0.7898 - accuracy: 0.7362 - val_loss: 1.19
89 - val_accuracy: 0.6146
Epoch 157/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.7884 - accuracy: 0.7370 - val_loss: 1.19
72 - val_accuracy: 0.6172
Epoch 158/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.7872 - accuracy: 0.7376 - val_loss: 1.19
43 - val_accuracy: 0.6178
Epoch 159/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.7861 - accuracy: 0.7382 - val_loss: 1.19
83 - val_accuracy: 0.6156
Epoch 160/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.7847 - accuracy: 0.7400 - val_loss: 1.19
50 - val_accuracy: 0.6172
Epoch 161/200
3125/3125 [==============================] - 6s 2ms/step - loss: 0.7836 - accuracy: 0.7393 - val_loss: 1.19
57 - val_accuracy: 0.6180
Epoch 162/200
3125/3125 [==============================] - 6s 2ms/step - loss: 0.7825 - accuracy: 0.7400 - val_loss: 1.19
46 - val_accuracy: 0.6188
Epoch 163/200
3125/3125 [==============================] - 6s 2ms/step - loss: 0.7809 - accuracy: 0.7407 - val_loss: 1.19
46 - val_accuracy: 0.6191
Epoch 164/200
3125/3125 [==============================] - 6s 2ms/step - loss: 0.7798 - accuracy: 0.7416 - val_loss: 1.19
29 - val_accuracy: 0.6193
Epoch 165/200
3125/3125 [==============================] - 6s 2ms/step - loss: 0.7785 - accuracy: 0.7415 - val_loss: 1.19
92 - val_accuracy: 0.6149
Epoch 166/200
3125/3125 [==============================] - 6s 2ms/step - loss: 0.7773 - accuracy: 0.7427 - val_loss: 1.19
13 - val_accuracy: 0.6189
Epoch 167/200
3125/3125 [==============================] - 6s 2ms/step - loss: 0.7761 - accuracy: 0.7424 - val_loss: 1.19
54 - val_accuracy: 0.6181
Epoch 168/200
3125/3125 [==============================] - 6s 2ms/step - loss: 0.7748 - accuracy: 0.7435 - val_loss: 1.19
42 - val_accuracy: 0.6188

Epoch 169/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.7737 - accuracy: 0.7430 - val_loss: 1.19
16 - val_accuracy: 0.6187
Epoch 170/200
```

```
3125/3125 [==============================] - 5s 2ms/step - loss: 0.7722 - accuracy: 0.7442 - val_loss: 1.19
06 - val_accuracy: 0.6188
Epoch 171/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.7714 - accuracy: 0.7443 - val_loss: 1.19
39 - val_accuracy: 0.6183
Epoch 172/200
3125/3125 [==============================] - 6s 2ms/step - loss: 0.7703 - accuracy: 0.7441 - val_loss: 1.19
25 - val_accuracy: 0.6175
Epoch 173/200
3125/3125 [==============================] - 6s 2ms/step - loss: 0.7688 - accuracy: 0.7452 - val_loss: 1.19
12 - val_accuracy: 0.6176
Epoch 174/200
3125/3125 [==============================] - 6s 2ms/step - loss: 0.7679 - accuracy: 0.7445 - val_loss: 1.18
97 - val_accuracy: 0.6180
Epoch 175/200
3125/3125 [==============================] - 6s 2ms/step - loss: 0.7665 - accuracy: 0.7455 - val_loss: 1.19
40 - val_accuracy: 0.6198
Epoch 176/200
3125/3125 [==============================] - 6s 2ms/step - loss: 0.7649 - accuracy: 0.7453 - val_loss: 1.18
87 - val_accuracy: 0.6216
Epoch 177/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.7643 - accuracy: 0.7456 - val_loss: 1.19
26 - val_accuracy: 0.6181
Epoch 178/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.7631 - accuracy: 0.7473 - val_loss: 1.19
03 - val_accuracy: 0.6178
Epoch 179/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.7623 - accuracy: 0.7478 - val_loss: 1.18
92 - val_accuracy: 0.6207
Epoch 180/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.7607 - accuracy: 0.7478 - val_loss: 1.19
27 - val_accuracy: 0.6184
Epoch 181/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.7595 - accuracy: 0.7482 - val_loss: 1.18
82 - val_accuracy: 0.6209
Epoch 182/200
3125/3125 [==============================] - 6s 2ms/step - loss: 0.7579 - accuracy: 0.7488 - val_loss: 1.18
81 - val_accuracy: 0.6224
Epoch 183/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.7570 - accuracy: 0.7491 - val_loss: 1.19
29 - val_accuracy: 0.6189
Epoch 184/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.7558 - accuracy: 0.7506 - val_loss: 1.18
```

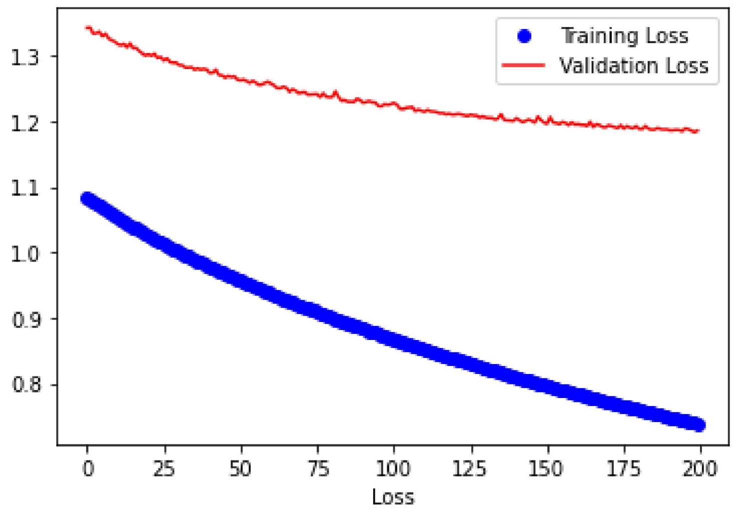
```
88 - val_accuracy: 0.6206
Epoch 185/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.7545 - accuracy: 0.7503 - val_loss: 1.18
71 - val_accuracy: 0.6225
Epoch 186/200
3125/3125 [==============================] - 6s 2ms/step - loss: 0.7537 - accuracy: 0.7508 - val_loss: 1.18
73 - val_accuracy: 0.6222
Epoch 187/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.7526 - accuracy: 0.7506 - val_loss: 1.18
97 - val_accuracy: 0.6199
Epoch 188/200
3125/3125 [==============================] - 6s 2ms/step - loss: 0.7515 - accuracy: 0.7505 - val_loss: 1.18
80 - val_accuracy: 0.6200
Epoch 189/200
3125/3125 [==============================] - 6s 2ms/step - loss: 0.7505 - accuracy: 0.7527 - val_loss: 1.18
73 - val_accuracy: 0.6237
Epoch 190/200
3125/3125 [==============================] - 6s 2ms/step - loss: 0.7491 - accuracy: 0.7528 - val_loss: 1.18
77 - val_accuracy: 0.6191
Epoch 191/200
3125/3125 [==============================] - 6s 2ms/step - loss: 0.7477 - accuracy: 0.7528 - val_loss: 1.18
60 - val_accuracy: 0.6199
Epoch 192/200
3125/3125 [==============================] - 6s 2ms/step - loss: 0.7469 - accuracy: 0.7539 - val_loss: 1.18
62 - val_accuracy: 0.6235
Epoch 193/200
3125/3125 [==============================] - 6s 2ms/step - loss: 0.7459 - accuracy: 0.7540 - val_loss: 1.18
70 - val_accuracy: 0.6204
Epoch 194/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.7448 - accuracy: 0.7540 - val_loss: 1.18
67 - val_accuracy: 0.6202
Epoch 195/200
3125/3125 [==============================] - 6s 2ms/step - loss: 0.7438 - accuracy: 0.7532 - val_loss: 1.18
48 - val_accuracy: 0.6225
Epoch 196/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.7426 - accuracy: 0.7544 - val_loss: 1.18
92 - val_accuracy: 0.6221
Epoch 197/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.7412 - accuracy: 0.7557 - val_loss: 1.18
79 - val_accuracy: 0.6224
Epoch 198/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.7401 - accuracy: 0.7560 - val_loss: 1.18
58 - val_accuracy: 0.6239
```

```
Epoch 199/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.7392 - accuracy: 0.7557 - val_loss: 1.18
35 - val_accuracy: 0.6242
Epoch 200/200
3125/3125 [==============================] - 5s 2ms/step - loss: 0.7378 - accuracy: 0.7570 - val_loss: 1.18
56 - val_accuracy: 0.6245
```

In [11]:

```
# Plot the loss curve
import matplotlib.pyplot as plt
%matplotlib inline

epochs = range(200)
train_acc = history.history['loss']
valid_acc = history.history['val_loss']
plt.plot(epochs, train_acc, 'bo', label='Training Loss')
plt.plot(epochs, valid_acc, 'r', label='Validation Loss')
plt.xlabel('Epochs')
plt.xlabel('Loss')
plt.legend()
plt.show()
```



5. Evaluate the model on the test set (5 points)

Do NOT used the test set until now. Make sure that your model parameters and hyper-parameters are independent of the test set.

In [12]:
```
modvar = model_cnn.evaluate(x_test, y_test_vec)
print('loss = ' + str(modvar[0]))
print('accuracy = ' + str(modvar[1]))
```

```
313/313 [==============================] - 1s 1ms/step - loss: 1.1856 - accuracy: 0.6245
loss = 1.1856324672698975
accuracy = 0.6244999766349792
```

6. Building model with new structure (25 points)

- In this section, you can build your model with adding new layers (e.g, BN layer or dropout layer, ...).
- If you want to regularize a `Conv/Dense layer`, you should place a `Dropout layer` before the `Conv/Dense layer`.
- You can try to compare their loss curve and testing accuracy and analyze your findings.
- You need to try at lease two different model structures.

6.1. Model 1

In [13]:
```
from keras.layers import Dense
from keras.layers.pooling import MaxPooling2D
from keras.layers.convolutional import Conv2D
from keras import models, layers

# Build the network
model_cnn = models.Sequential()
model_cnn.add(layers.Conv2D(32, (3, 3), activation='relu', input_shape=(32, 32, 3)))
model_cnn.add(layers.MaxPooling2D((2, 2)))
model_cnn.add(layers.Conv2D(64,(4,4), activation='relu'))
model_cnn.add(layers.MaxPooling2D((2, 2)))
model_cnn.add(layers.Flatten())
model_cnn.add(layers.Dense(256, activation='relu'))
model_cnn.add(layers.Dropout(0.3))
model_cnn.add(layers.Dense(10, activation='softmax'))

# Define model optimizer and loss function
from keras import optimizers
model_cnn.compile(optimizer='Ftrl', loss='categorical_crossentropy', metrics=['accuracy'])

# Train the model and store model parameters/loss values
history = model_cnn.fit(x_tr, y_tr, batch_size=256, epochs=50, validation_data=(x_val, y_val))
```

```
Epoch 1/50
157/157 [==============================] - 2s 7ms/step - loss: 3.6486 - accuracy: 0.2314 - val_loss: 1.9772
- val_accuracy: 0.2675
Epoch 2/50
157/157 [==============================] - 1s 5ms/step - loss: 1.8793 - accuracy: 0.3239 - val_loss: 1.7669
- val_accuracy: 0.3791
Epoch 3/50
157/157 [==============================] - 1s 5ms/step - loss: 1.7732 - accuracy: 0.3647 - val_loss: 1.7049
- val_accuracy: 0.3952
Epoch 4/50
157/157 [==============================] - 1s 5ms/step - loss: 1.6933 - accuracy: 0.3914 - val_loss: 1.6116
- val_accuracy: 0.4333
Epoch 5/50
157/157 [==============================] - 1s 5ms/step - loss: 1.6460 - accuracy: 0.4128 - val_loss: 1.6012
- val_accuracy: 0.4229
Epoch 6/50
157/157 [==============================] - 1s 5ms/step - loss: 1.6031 - accuracy: 0.4276 - val_loss: 1.5572
- val_accuracy: 0.4468
Epoch 7/50
```

```
157/157 [==============================] - 1s 5ms/step - loss: 1.5740 - accuracy: 0.4367 - val_loss: 1.5000
- val_accuracy: 0.4675
Epoch 8/50
157/157 [==============================] - 1s 5ms/step - loss: 1.5400 - accuracy: 0.4546 - val_loss: 1.4800
- val_accuracy: 0.4723
Epoch 9/50
157/157 [==============================] - 1s 5ms/step - loss: 1.5184 - accuracy: 0.4596 - val_loss: 1.5212
- val_accuracy: 0.4606
Epoch 10/50
157/157 [==============================] - 1s 5ms/step - loss: 1.4926 - accuracy: 0.4690 - val_loss: 1.4450
- val_accuracy: 0.4860
Epoch 11/50
157/157 [==============================] - 1s 5ms/step - loss: 1.4691 - accuracy: 0.4744 - val_loss: 1.4382
- val_accuracy: 0.4946
Epoch 12/50
157/157 [==============================] - 1s 5ms/step - loss: 1.4513 - accuracy: 0.4828 - val_loss: 1.4187
- val_accuracy: 0.4988
Epoch 13/50
157/157 [==============================] - 1s 5ms/step - loss: 1.4290 - accuracy: 0.4940 - val_loss: 1.3778
- val_accuracy: 0.5144
Epoch 14/50
157/157 [==============================] - 1s 5ms/step - loss: 1.4142 - accuracy: 0.4978 - val_loss: 1.3858
- val_accuracy: 0.5113
Epoch 15/50
157/157 [==============================] - 1s 5ms/step - loss: 1.3955 - accuracy: 0.5048 - val_loss: 1.3504
- val_accuracy: 0.5244
Epoch 16/50
157/157 [==============================] - 1s 5ms/step - loss: 1.3814 - accuracy: 0.5092 - val_loss: 1.3589
- val_accuracy: 0.5193
Epoch 17/50
157/157 [==============================] - 1s 5ms/step - loss: 1.3616 - accuracy: 0.5185 - val_loss: 1.3742
- val_accuracy: 0.5134
Epoch 18/50
157/157 [==============================] - 1s 5ms/step - loss: 1.3471 - accuracy: 0.5268 - val_loss: 1.3391
- val_accuracy: 0.5215
Epoch 19/50
157/157 [==============================] - 1s 5ms/step - loss: 1.3379 - accuracy: 0.5264 - val_loss: 1.3612
- val_accuracy: 0.5135
Epoch 20/50
157/157 [==============================] - 1s 5ms/step - loss: 1.3218 - accuracy: 0.5337 - val_loss: 1.3120
- val_accuracy: 0.5329
Epoch 21/50
157/157 [==============================] - 1s 5ms/step - loss: 1.3080 - accuracy: 0.5387 - val_loss: 1.2931
```

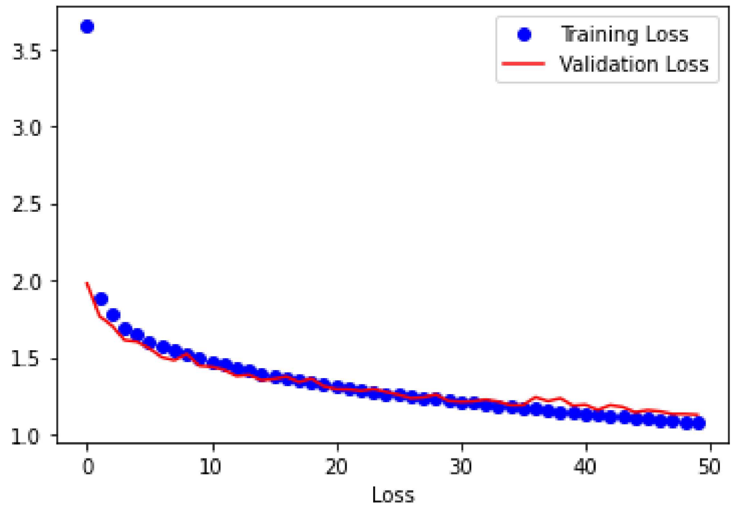
```
- val_accuracy: 0.5475
Epoch 22/50
157/157 [==============================] - 1s 5ms/step - loss: 1.3010 - accuracy: 0.5412 - val_loss: 1.2918
- val_accuracy: 0.5434
Epoch 23/50
157/157 [==============================] - 1s 5ms/step - loss: 1.2889 - accuracy: 0.5448 - val_loss: 1.2824
- val_accuracy: 0.5450
Epoch 24/50
157/157 [==============================] - 1s 5ms/step - loss: 1.2754 - accuracy: 0.5493 - val_loss: 1.2937
- val_accuracy: 0.5437
Epoch 25/50
157/157 [==============================] - 1s 5ms/step - loss: 1.2636 - accuracy: 0.5549 - val_loss: 1.2747
- val_accuracy: 0.5455
Epoch 26/50
157/157 [==============================] - 1s 5ms/step - loss: 1.2575 - accuracy: 0.5574 - val_loss: 1.2570
- val_accuracy: 0.5576
Epoch 27/50
157/157 [==============================] - 1s 5ms/step - loss: 1.2457 - accuracy: 0.5631 - val_loss: 1.2344
- val_accuracy: 0.5665
Epoch 28/50
157/157 [==============================] - 1s 5ms/step - loss: 1.2375 - accuracy: 0.5648 - val_loss: 1.2400
- val_accuracy: 0.5609
Epoch 29/50
157/157 [==============================] - 1s 5ms/step - loss: 1.2310 - accuracy: 0.5688 - val_loss: 1.2576
- val_accuracy: 0.5548
Epoch 30/50
157/157 [==============================] - 1s 5ms/step - loss: 1.2167 - accuracy: 0.5734 - val_loss: 1.2170
- val_accuracy: 0.5694
Epoch 31/50
157/157 [==============================] - 1s 5ms/step - loss: 1.2109 - accuracy: 0.5746 - val_loss: 1.2106
- val_accuracy: 0.5762
Epoch 32/50
157/157 [==============================] - 1s 5ms/step - loss: 1.2022 - accuracy: 0.5795 - val_loss: 1.2139
- val_accuracy: 0.5696
Epoch 33/50
157/157 [==============================] - 1s 5ms/step - loss: 1.1936 - accuracy: 0.5809 - val_loss: 1.2251
- val_accuracy: 0.5623
Epoch 34/50
157/157 [==============================] - 1s 5ms/step - loss: 1.1845 - accuracy: 0.5840 - val_loss: 1.2119
- val_accuracy: 0.5722
Epoch 35/50
157/157 [==============================] - 1s 5ms/step - loss: 1.1758 - accuracy: 0.5893 - val_loss: 1.1890
- val_accuracy: 0.5797
```

```
Epoch 36/50
157/157 [==============================] - 1s 5ms/step - loss: 1.1707 - accuracy: 0.5912 - val_loss: 1.1889
- val_accuracy: 0.5811
Epoch 37/50
157/157 [==============================] - 1s 5ms/step - loss: 1.1630 - accuracy: 0.5932 - val_loss: 1.2412
- val_accuracy: 0.5553
Epoch 38/50
157/157 [==============================] - 1s 5ms/step - loss: 1.1567 - accuracy: 0.5949 - val_loss: 1.2147
- val_accuracy: 0.5711
Epoch 39/50
157/157 [==============================] - 1s 5ms/step - loss: 1.1476 - accuracy: 0.6012 - val_loss: 1.2339
- val_accuracy: 0.5679
Epoch 40/50
157/157 [==============================] - 1s 5ms/step - loss: 1.1401 - accuracy: 0.6016 - val_loss: 1.1858
- val_accuracy: 0.5834
Epoch 41/50
157/157 [==============================] - 1s 5ms/step - loss: 1.1319 - accuracy: 0.6058 - val_loss: 1.1916
- val_accuracy: 0.5767
Epoch 42/50
157/157 [==============================] - 1s 5ms/step - loss: 1.1274 - accuracy: 0.6072 - val_loss: 1.1579
- val_accuracy: 0.5911
Epoch 43/50
157/157 [==============================] - 1s 5ms/step - loss: 1.1226 - accuracy: 0.6088 - val_loss: 1.1891
- val_accuracy: 0.5857
Epoch 44/50
157/157 [==============================] - 1s 5ms/step - loss: 1.1163 - accuracy: 0.6119 - val_loss: 1.1770
- val_accuracy: 0.5829
Epoch 45/50
157/157 [==============================] - 1s 5ms/step - loss: 1.1058 - accuracy: 0.6159 - val_loss: 1.1428
- val_accuracy: 0.5959
Epoch 46/50
157/157 [==============================] - 1s 5ms/step - loss: 1.0997 - accuracy: 0.6199 - val_loss: 1.1562
- val_accuracy: 0.5968
Epoch 47/50
157/157 [==============================] - 1s 5ms/step - loss: 1.0936 - accuracy: 0.6194 - val_loss: 1.1479
- val_accuracy: 0.5999
Epoch 48/50
157/157 [==============================] - 1s 5ms/step - loss: 1.0864 - accuracy: 0.6238 - val_loss: 1.1315
- val_accuracy: 0.6033
Epoch 49/50
157/157 [==============================] - 1s 5ms/step - loss: 1.0823 - accuracy: 0.6242 - val_loss: 1.1317
- val_accuracy: 0.6055
Epoch 50/50
```

```
157/157 [==============================] - 1s 5ms/step - loss: 1.0786 - accuracy: 0.6265 - val_loss: 1.1270
- val_accuracy: 0.6074
```

In [14]:
```
# Plot the loss curve
import matplotlib.pyplot as plt
%matplotlib inline

epochs = range(50)
train_acc = history.history['loss']
valid_acc = history.history['val_loss']
plt.plot(epochs, train_acc, 'bo', label='Training Loss')
plt.plot(epochs, valid_acc, 'r', label='Validation Loss')
plt.xlabel('Epochs')
plt.xlabel('Loss')
plt.legend()
plt.show()
```



Evaluate with Test Set

In [15]:
```
model_cnn.compile(optimizer='Ftrl', loss='categorical_crossentropy', metrics=['accuracy'])

# Train the model and store model parameters/loss values
history = model_cnn.fit(x_train, y_train_vec, batch_size=256, epochs=50, validation_data = (x_test, y_test_vec))
```

```
Epoch 1/50
196/196 [==============================] - 2s 6ms/step - loss: 2.2955 - accuracy: 0.1040 - val_loss: 2.3002 -
val_accuracy: 0.1000
Epoch 2/50
196/196 [==============================] - 1s 5ms/step - loss: 2.2940 - accuracy: 0.1042 - val_loss: 2.2774 -
val_accuracy: 0.1324
Epoch 3/50
196/196 [==============================] - 1s 5ms/step - loss: 2.2127 - accuracy: 0.1540 - val_loss: 2.1569 -
val_accuracy: 0.1706
Epoch 4/50
196/196 [==============================] - 1s 5ms/step - loss: 2.0949 - accuracy: 0.2083 - val_loss: 2.0403 -
val_accuracy: 0.2451
Epoch 5/50
196/196 [==============================] - 1s 5ms/step - loss: 2.0369 - accuracy: 0.2395 - val_loss: 2.0130 -
val_accuracy: 0.2517
Epoch 6/50
196/196 [==============================] - 1s 5ms/step - loss: 1.9997 - accuracy: 0.2573 - val_loss: 1.9597 -
val_accuracy: 0.2793
Epoch 7/50
196/196 [==============================] - 1s 5ms/step - loss: 1.9574 - accuracy: 0.2776 - val_loss: 1.9436 -
val_accuracy: 0.2791
Epoch 8/50
196/196 [==============================] - 1s 5ms/step - loss: 1.8994 - accuracy: 0.2997 - val_loss: 1.8511 -
val_accuracy: 0.3165
Epoch 9/50
196/196 [==============================] - 1s 5ms/step - loss: 1.8390 - accuracy: 0.3245 - val_loss: 1.7873 -
val_accuracy: 0.3491
Epoch 10/50
196/196 [==============================] - 1s 5ms/step - loss: 1.7911 - accuracy: 0.3468 - val_loss: 1.7746 -
val_accuracy: 0.3516
Epoch 11/50
196/196 [==============================] - 1s 5ms/step - loss: 1.7508 - accuracy: 0.3636 - val_loss: 1.7183 -
val_accuracy: 0.3836
Epoch 12/50
196/196 [==============================] - 1s 5ms/step - loss: 1.7202 - accuracy: 0.3747 - val_loss: 1.7055 -
val_accuracy: 0.3851
Epoch 13/50
```

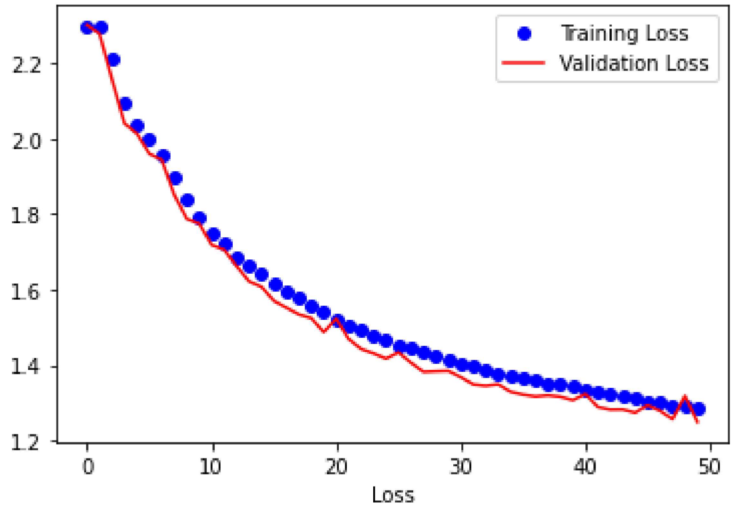
```
196/196 [==============================] - 1s 5ms/step - loss: 1.6871 - accuracy: 0.3866 - val_loss: 1.6623 -
val_accuracy: 0.4073
Epoch 14/50
196/196 [==============================] - 1s 5ms/step - loss: 1.6640 - accuracy: 0.3962 - val_loss: 1.6217 -
val_accuracy: 0.4162
Epoch 15/50
196/196 [==============================] - 1s 5ms/step - loss: 1.6407 - accuracy: 0.4041 - val_loss: 1.6075 -
val_accuracy: 0.4227
Epoch 16/50
196/196 [==============================] - 1s 5ms/step - loss: 1.6177 - accuracy: 0.4136 - val_loss: 1.5701 -
val_accuracy: 0.4328
Epoch 17/50
196/196 [==============================] - 1s 5ms/step - loss: 1.5964 - accuracy: 0.4224 - val_loss: 1.5521 -
val_accuracy: 0.4354
Epoch 18/50
196/196 [==============================] - 1s 5ms/step - loss: 1.5768 - accuracy: 0.4297 - val_loss: 1.5342 -
val_accuracy: 0.4391
Epoch 19/50
196/196 [==============================] - 1s 5ms/step - loss: 1.5549 - accuracy: 0.4337 - val_loss: 1.5244 -
val_accuracy: 0.4437
Epoch 20/50
196/196 [==============================] - 1s 5ms/step - loss: 1.5392 - accuracy: 0.4433 - val_loss: 1.4873 -
val_accuracy: 0.4590
Epoch 21/50
196/196 [==============================] - 1s 5ms/step - loss: 1.5207 - accuracy: 0.4485 - val_loss: 1.5229 -
val_accuracy: 0.4475
Epoch 22/50
196/196 [==============================] - 1s 5ms/step - loss: 1.5045 - accuracy: 0.4548 - val_loss: 1.4691 -
val_accuracy: 0.4684
Epoch 23/50
196/196 [==============================] - 1s 5ms/step - loss: 1.4915 - accuracy: 0.4594 - val_loss: 1.4429 -
val_accuracy: 0.4780
Epoch 24/50
196/196 [==============================] - 1s 5ms/step - loss: 1.4778 - accuracy: 0.4658 - val_loss: 1.4311 -
val_accuracy: 0.4801
Epoch 25/50
196/196 [==============================] - 1s 5ms/step - loss: 1.4659 - accuracy: 0.4703 - val_loss: 1.4173 -
val_accuracy: 0.4871
Epoch 26/50
196/196 [==============================] - 1s 5ms/step - loss: 1.4505 - accuracy: 0.4762 - val_loss: 1.4339 -
val_accuracy: 0.4858
Epoch 27/50
196/196 [==============================] - 1s 5ms/step - loss: 1.4439 - accuracy: 0.4787 - val_loss: 1.4065 -
```

```
val_accuracy: 0.4897
Epoch 28/50
196/196 [==============================] - 1s 5ms/step - loss: 1.4331 - accuracy: 0.4850 - val_loss: 1.3823 -
val_accuracy: 0.5024
Epoch 29/50
196/196 [==============================] - 1s 5ms/step - loss: 1.4224 - accuracy: 0.4894 - val_loss: 1.3838 -
val_accuracy: 0.4978
Epoch 30/50
196/196 [==============================] - 1s 5ms/step - loss: 1.4133 - accuracy: 0.4910 - val_loss: 1.3848 -
val_accuracy: 0.4988
Epoch 31/50
196/196 [==============================] - 1s 5ms/step - loss: 1.4041 - accuracy: 0.4949 - val_loss: 1.3683 -
val_accuracy: 0.5090
Epoch 32/50
196/196 [==============================] - 1s 5ms/step - loss: 1.3955 - accuracy: 0.4980 - val_loss: 1.3489 -
val_accuracy: 0.5184
Epoch 33/50
196/196 [==============================] - 1s 5ms/step - loss: 1.3884 - accuracy: 0.5004 - val_loss: 1.3451 -
val_accuracy: 0.5195
Epoch 34/50
196/196 [==============================] - 1s 5ms/step - loss: 1.3788 - accuracy: 0.5046 - val_loss: 1.3492 -
val_accuracy: 0.5167
Epoch 35/50
196/196 [==============================] - 1s 5ms/step - loss: 1.3740 - accuracy: 0.5068 - val_loss: 1.3294 -
val_accuracy: 0.5256
Epoch 36/50
196/196 [==============================] - 1s 5ms/step - loss: 1.3644 - accuracy: 0.5107 - val_loss: 1.3220 -
val_accuracy: 0.5275
Epoch 37/50
196/196 [==============================] - 1s 5ms/step - loss: 1.3590 - accuracy: 0.5125 - val_loss: 1.3170 -
val_accuracy: 0.5326
Epoch 38/50
196/196 [==============================] - 1s 5ms/step - loss: 1.3514 - accuracy: 0.5153 - val_loss: 1.3199 -
val_accuracy: 0.5306
Epoch 39/50
196/196 [==============================] - 1s 5ms/step - loss: 1.3481 - accuracy: 0.5175 - val_loss: 1.3163 -
val_accuracy: 0.5249
Epoch 40/50
196/196 [==============================] - 1s 5ms/step - loss: 1.3430 - accuracy: 0.5186 - val_loss: 1.3071 -
val_accuracy: 0.5350
Epoch 41/50
196/196 [==============================] - 1s 5ms/step - loss: 1.3332 - accuracy: 0.5234 - val_loss: 1.3237 -
val_accuracy: 0.5226
```

```
Epoch 42/50
196/196 [==============================] - 1s 5ms/step - loss: 1.3275 - accuracy: 0.5250 - val_loss: 1.2891 -
val_accuracy: 0.5406
Epoch 43/50
196/196 [==============================] - 1s 5ms/step - loss: 1.3214 - accuracy: 0.5291 - val_loss: 1.2822 -
val_accuracy: 0.5446
Epoch 44/50
196/196 [==============================] - 1s 5ms/step - loss: 1.3185 - accuracy: 0.5291 - val_loss: 1.2826 -
val_accuracy: 0.5461
Epoch 45/50
196/196 [==============================] - 1s 5ms/step - loss: 1.3105 - accuracy: 0.5320 - val_loss: 1.2740 -
val_accuracy: 0.5463
Epoch 46/50
196/196 [==============================] - 1s 5ms/step - loss: 1.3048 - accuracy: 0.5342 - val_loss: 1.2940 -
val_accuracy: 0.5412
Epoch 47/50
196/196 [==============================] - 1s 5ms/step - loss: 1.3012 - accuracy: 0.5354 - val_loss: 1.2789 -
val_accuracy: 0.5450
Epoch 48/50
196/196 [==============================] - 1s 5ms/step - loss: 1.2939 - accuracy: 0.5379 - val_loss: 1.2569 -
val_accuracy: 0.5545
Epoch 49/50
196/196 [==============================] - 1s 5ms/step - loss: 1.2891 - accuracy: 0.5408 - val_loss: 1.3183 -
val_accuracy: 0.5315
Epoch 50/50
196/196 [==============================] - 1s 5ms/step - loss: 1.2862 - accuracy: 0.5439 - val_loss: 1.2495 -
val_accuracy: 0.5556
```

In [16]:
```
# Plot the loss curve
import matplotlib.pyplot as plt
%matplotlib inline

epochs = range(50)
train_acc = history.history['loss']
valid_acc = history.history['val_loss']
plt.plot(epochs, train_acc, 'bo', label='Training Loss')
plt.plot(epochs, valid_acc, 'r', label='Validation Loss')
plt.xlabel('Epochs')
plt.xlabel('Loss')
plt.legend()
plt.show()
```



6.2. Model 2

In [17]:
```
from keras.layers import Dense
from keras.layers.pooling import MaxPooling2D
from keras.layers.convolutional import Conv2D
from keras import models, layers

# Build the network
model_cnn = models.Sequential()
model_cnn.add(layers.Conv2D(32, (3, 3), activation='relu', input_shape=(32, 32, 3)))
model_cnn.add(layers.MaxPooling2D((2, 2)))
model_cnn.add(layers.Conv2D(64,(4,4), activation='relu'))
model_cnn.add(layers.MaxPooling2D((2, 2)))
model_cnn.add(layers.Flatten())
model_cnn.add(layers.Dropout(0.3))
model_cnn.add(layers.Dense(256, activation='relu'))
model_cnn.add(layers.Dense(10, activation='softmax'))

# Define model optimizer and loss function
from keras import optimizers
model_cnn.compile(optimizer='Ftrl', loss='categorical_crossentropy', metrics=['accuracy'])

# Train the model and store model parameters/loss values
history = model_cnn.fit(x_tr, y_tr, batch_size=256, epochs=50, validation_data=(x_val, y_val))
```

```
Epoch 1/50
157/157 [==============================] - 2s 7ms/step - loss: 3.0316 - accuracy: 0.2910 - val_loss: 1.7132 -
val_accuracy: 0.3972
Epoch 2/50
157/157 [==============================] - 1s 6ms/step - loss: 1.6809 - accuracy: 0.4047 - val_loss: 1.6688 -
val_accuracy: 0.3862
Epoch 3/50
157/157 [==============================] - 1s 6ms/step - loss: 1.5657 - accuracy: 0.4449 - val_loss: 1.5960 -
val_accuracy: 0.4261
Epoch 4/50
157/157 [==============================] - 1s 6ms/step - loss: 1.4882 - accuracy: 0.4729 - val_loss: 1.4221 -
val_accuracy: 0.5006
Epoch 5/50
157/157 [==============================] - 1s 5ms/step - loss: 1.4299 - accuracy: 0.4953 - val_loss: 1.4337 -
val_accuracy: 0.4845
Epoch 6/50
157/157 [==============================] - 1s 5ms/step - loss: 1.3961 - accuracy: 0.5082 - val_loss: 1.3677 -
val_accuracy: 0.5132
Epoch 7/50
```

```
157/157 [==============================] - 1s 5ms/step - loss: 1.3595 - accuracy: 0.5227 - val_loss: 1.4488 -
val_accuracy: 0.4909
Epoch 8/50
157/157 [==============================] - 1s 5ms/step - loss: 1.3351 - accuracy: 0.5298 - val_loss: 1.3371 -
val_accuracy: 0.5347
Epoch 9/50
157/157 [==============================] - 1s 5ms/step - loss: 1.3130 - accuracy: 0.5387 - val_loss: 1.2887 -
val_accuracy: 0.5547
Epoch 10/50
157/157 [==============================] - 1s 5ms/step - loss: 1.2975 - accuracy: 0.5471 - val_loss: 1.3066 -
val_accuracy: 0.5397
Epoch 11/50
157/157 [==============================] - 1s 5ms/step - loss: 1.2755 - accuracy: 0.5533 - val_loss: 1.2968 -
val_accuracy: 0.5422
Epoch 12/50
157/157 [==============================] - 1s 5ms/step - loss: 1.2551 - accuracy: 0.5611 - val_loss: 1.2396 -
val_accuracy: 0.5661
Epoch 13/50
157/157 [==============================] - 1s 5ms/step - loss: 1.2405 - accuracy: 0.5664 - val_loss: 1.2576 -
val_accuracy: 0.5623
Epoch 14/50
157/157 [==============================] - 1s 5ms/step - loss: 1.2303 - accuracy: 0.5711 - val_loss: 1.2203 -
val_accuracy: 0.5702
Epoch 15/50
157/157 [==============================] - 1s 5ms/step - loss: 1.2145 - accuracy: 0.5749 - val_loss: 1.2385 -
val_accuracy: 0.5652
Epoch 16/50
157/157 [==============================] - 1s 5ms/step - loss: 1.2008 - accuracy: 0.5810 - val_loss: 1.2017 -
val_accuracy: 0.5853
Epoch 17/50
157/157 [==============================] - 1s 5ms/step - loss: 1.1882 - accuracy: 0.5875 - val_loss: 1.2061 -
val_accuracy: 0.5739
Epoch 18/50
157/157 [==============================] - 1s 5ms/step - loss: 1.1812 - accuracy: 0.5894 - val_loss: 1.1761 -
val_accuracy: 0.5906
Epoch 19/50
157/157 [==============================] - 1s 5ms/step - loss: 1.1673 - accuracy: 0.5918 - val_loss: 1.1839 -
val_accuracy: 0.5895
Epoch 20/50
157/157 [==============================] - 1s 5ms/step - loss: 1.1567 - accuracy: 0.5969 - val_loss: 1.1642 -
val_accuracy: 0.5988
Epoch 21/50
157/157 [==============================] - 1s 5ms/step - loss: 1.1450 - accuracy: 0.6019 - val_loss: 1.1918 -
```

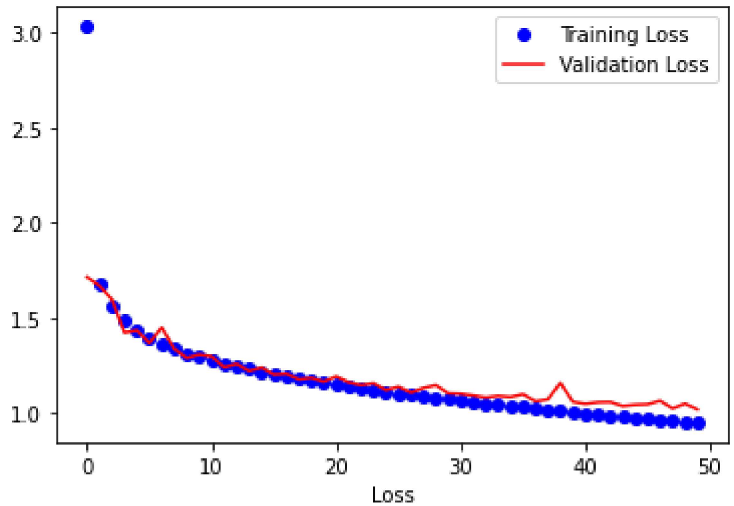
```
val_accuracy: 0.5836
Epoch 22/50
157/157 [==============================] - 1s 5ms/step - loss: 1.1408 - accuracy: 0.6056 - val_loss: 1.1610 -
val_accuracy: 0.5937
Epoch 23/50
157/157 [==============================] - 1s 5ms/step - loss: 1.1316 - accuracy: 0.6088 - val_loss: 1.1461 -
val_accuracy: 0.6046
Epoch 24/50
157/157 [==============================] - 1s 5ms/step - loss: 1.1165 - accuracy: 0.6124 - val_loss: 1.1561 -
val_accuracy: 0.5924
Epoch 25/50
157/157 [==============================] - 1s 5ms/step - loss: 1.1129 - accuracy: 0.6147 - val_loss: 1.1198 -
val_accuracy: 0.6118
Epoch 26/50
157/157 [==============================] - 1s 5ms/step - loss: 1.1020 - accuracy: 0.6197 - val_loss: 1.1375 -
val_accuracy: 0.5997
Epoch 27/50
157/157 [==============================] - 1s 5ms/step - loss: 1.0952 - accuracy: 0.6231 - val_loss: 1.1069 -
val_accuracy: 0.6169
Epoch 28/50
157/157 [==============================] - 1s 5ms/step - loss: 1.0913 - accuracy: 0.6235 - val_loss: 1.1307 -
val_accuracy: 0.6087
Epoch 29/50
157/157 [==============================] - 1s 5ms/step - loss: 1.0761 - accuracy: 0.6299 - val_loss: 1.1458 -
val_accuracy: 0.6015
Epoch 30/50
157/157 [==============================] - 1s 5ms/step - loss: 1.0713 - accuracy: 0.6294 - val_loss: 1.1031 -
val_accuracy: 0.6190
Epoch 31/50
157/157 [==============================] - 1s 5ms/step - loss: 1.0640 - accuracy: 0.6312 - val_loss: 1.1006 -
val_accuracy: 0.6197
Epoch 32/50
157/157 [==============================] - 1s 5ms/step - loss: 1.0536 - accuracy: 0.6358 - val_loss: 1.0914 -
val_accuracy: 0.6209
Epoch 33/50
157/157 [==============================] - 1s 5ms/step - loss: 1.0487 - accuracy: 0.6355 - val_loss: 1.0807 -
val_accuracy: 0.6264
Epoch 34/50
157/157 [==============================] - 1s 5ms/step - loss: 1.0399 - accuracy: 0.6406 - val_loss: 1.0886 -
val_accuracy: 0.6193
Epoch 35/50
157/157 [==============================] - 1s 5ms/step - loss: 1.0337 - accuracy: 0.6440 - val_loss: 1.0831 -
val_accuracy: 0.6244
```

```
Epoch 36/50
157/157 [==============================] - 1s 5ms/step - loss: 1.0299 - accuracy: 0.6453 - val_loss: 1.0981 -
val_accuracy: 0.6204
Epoch 37/50
157/157 [==============================] - 1s 5ms/step - loss: 1.0223 - accuracy: 0.6490 - val_loss: 1.0612 -
val_accuracy: 0.6338
Epoch 38/50
157/157 [==============================] - 1s 5ms/step - loss: 1.0138 - accuracy: 0.6508 - val_loss: 1.0711 -
val_accuracy: 0.6340
Epoch 39/50
157/157 [==============================] - 1s 5ms/step - loss: 1.0087 - accuracy: 0.6507 - val_loss: 1.1574 -
val_accuracy: 0.5891
Epoch 40/50
157/157 [==============================] - 1s 5ms/step - loss: 1.0070 - accuracy: 0.6525 - val_loss: 1.0580 -
val_accuracy: 0.6339
Epoch 41/50
157/157 [==============================] - 1s 5ms/step - loss: 0.9961 - accuracy: 0.6574 - val_loss: 1.0477 -
val_accuracy: 0.6404
Epoch 42/50
157/157 [==============================] - 1s 5ms/step - loss: 0.9958 - accuracy: 0.6577 - val_loss: 1.0558 -
val_accuracy: 0.6306
Epoch 43/50
157/157 [==============================] - 1s 5ms/step - loss: 0.9850 - accuracy: 0.6618 - val_loss: 1.0570 -
val_accuracy: 0.6313
Epoch 44/50
157/157 [==============================] - 1s 5ms/step - loss: 0.9785 - accuracy: 0.6629 - val_loss: 1.0359 -
val_accuracy: 0.6398
Epoch 45/50
157/157 [==============================] - 1s 5ms/step - loss: 0.9738 - accuracy: 0.6658 - val_loss: 1.0426 -
val_accuracy: 0.6404
Epoch 46/50
157/157 [==============================] - 1s 5ms/step - loss: 0.9729 - accuracy: 0.6633 - val_loss: 1.0449 -
val_accuracy: 0.6405
Epoch 47/50
157/157 [==============================] - 1s 5ms/step - loss: 0.9652 - accuracy: 0.6682 - val_loss: 1.0638 -
val_accuracy: 0.6263
Epoch 48/50
157/157 [==============================] - 1s 5ms/step - loss: 0.9612 - accuracy: 0.6700 - val_loss: 1.0232 -
val_accuracy: 0.6466
Epoch 49/50
157/157 [==============================] - 1s 5ms/step - loss: 0.9526 - accuracy: 0.6719 - val_loss: 1.0483 -
val_accuracy: 0.6333
Epoch 50/50
```

```
157/157 [==============================] - 1s 5ms/step - loss: 0.9508 - accuracy: 0.6716 - val_loss: 1.0184 - val_accuracy: 0.6486
```

In [18]:
```
# Plot the loss curve
import matplotlib.pyplot as plt
%matplotlib inline

epochs = range(50)
train_acc = history.history['loss']
valid_acc = history.history['val_loss']
plt.plot(epochs, train_acc, 'bo', label='Training Loss')
plt.plot(epochs, valid_acc, 'r', label='Validation Loss')
plt.xlabel('Epochs')
plt.xlabel('Loss')
plt.legend()
plt.show()
```



Evaluate with Test Set

In [19]:
```
model_cnn.compile(optimizer='Ftrl', loss='categorical_crossentropy', metrics=['accuracy'])

# Train the model and store model parameters/loss values
history = model_cnn.fit(x_train, y_train_vec, batch_size=256, epochs=50, validation_data = (x_test, y_test_vec))
```

```
Epoch 1/50
196/196 [==============================] - 2s 6ms/step - loss: 2.2952 - accuracy: 0.1020 - val_loss: 2.3026 -
val_accuracy: 0.1001
Epoch 2/50
196/196 [==============================] - 1s 5ms/step - loss: 2.3026 - accuracy: 0.1016 - val_loss: 2.3026 -
val_accuracy: 0.0990
Epoch 3/50
196/196 [==============================] - 1s 5ms/step - loss: 2.3026 - accuracy: 0.1001 - val_loss: 2.3026 -
val_accuracy: 0.0995
Epoch 4/50
196/196 [==============================] - 1s 5ms/step - loss: 2.3026 - accuracy: 0.0988 - val_loss: 2.3026 -
val_accuracy: 0.0997
Epoch 5/50
196/196 [==============================] - 1s 5ms/step - loss: 2.3026 - accuracy: 0.1049 - val_loss: 2.3026 -
val_accuracy: 0.0997
Epoch 6/50
196/196 [==============================] - 1s 5ms/step - loss: 2.3026 - accuracy: 0.1019 - val_loss: 2.3025 -
val_accuracy: 0.1000
Epoch 7/50
196/196 [==============================] - 1s 5ms/step - loss: 2.3025 - accuracy: 0.0999 - val_loss: 2.3025 -
val_accuracy: 0.1000
Epoch 8/50
196/196 [==============================] - 1s 5ms/step - loss: 2.3025 - accuracy: 0.1000 - val_loss: 2.3024 -
val_accuracy: 0.1000
Epoch 9/50
196/196 [==============================] - 1s 5ms/step - loss: 2.3024 - accuracy: 0.1000 - val_loss: 2.3022 -
val_accuracy: 0.1000
Epoch 10/50
196/196 [==============================] - 1s 5ms/step - loss: 2.3019 - accuracy: 0.1000 - val_loss: 2.3012 -
val_accuracy: 0.1000
Epoch 11/50
196/196 [==============================] - 1s 5ms/step - loss: 2.2993 - accuracy: 0.1000 - val_loss: 2.2955 -
val_accuracy: 0.1000
Epoch 12/50
196/196 [==============================] - 1s 5ms/step - loss: 2.2886 - accuracy: 0.1003 - val_loss: 2.2755 -
val_accuracy: 0.1116
Epoch 13/50
```

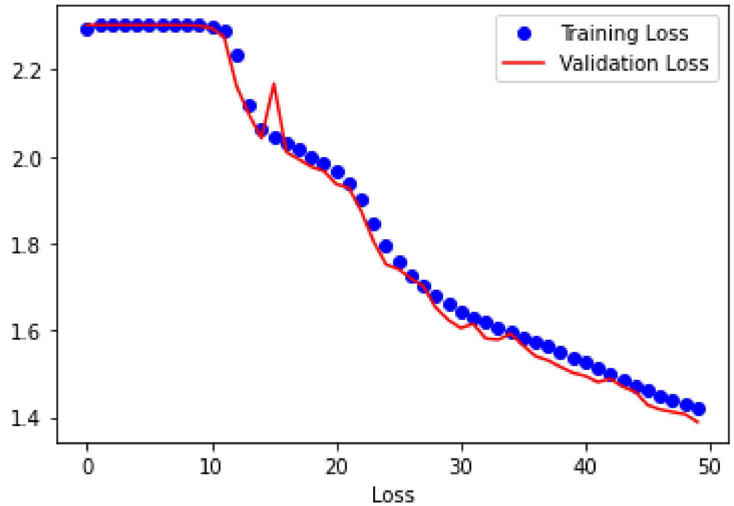
```
196/196 [==============================] - 1s 5ms/step - loss: 2.2342 - accuracy: 0.1197 - val_loss: 2.1623 -
val_accuracy: 0.1506
Epoch 14/50
196/196 [==============================] - 1s 5ms/step - loss: 2.1189 - accuracy: 0.1792 - val_loss: 2.0970 -
val_accuracy: 0.2207
Epoch 15/50
196/196 [==============================] - 1s 5ms/step - loss: 2.0647 - accuracy: 0.2292 - val_loss: 2.0426 -
val_accuracy: 0.2356
Epoch 16/50
196/196 [==============================] - 1s 5ms/step - loss: 2.0451 - accuracy: 0.2382 - val_loss: 2.1684 -
val_accuracy: 0.2040
Epoch 17/50
196/196 [==============================] - 1s 5ms/step - loss: 2.0302 - accuracy: 0.2464 - val_loss: 2.0104 -
val_accuracy: 0.2661
Epoch 18/50
196/196 [==============================] - 1s 5ms/step - loss: 2.0154 - accuracy: 0.2585 - val_loss: 1.9939 -
val_accuracy: 0.2715
Epoch 19/50
196/196 [==============================] - 1s 6ms/step - loss: 2.0002 - accuracy: 0.2675 - val_loss: 1.9768 -
val_accuracy: 0.2815
Epoch 20/50
196/196 [==============================] - 1s 6ms/step - loss: 1.9846 - accuracy: 0.2748 - val_loss: 1.9671 -
val_accuracy: 0.2868
Epoch 21/50
196/196 [==============================] - 1s 5ms/step - loss: 1.9662 - accuracy: 0.2852 - val_loss: 1.9367 -
val_accuracy: 0.2996
Epoch 22/50
196/196 [==============================] - 1s 5ms/step - loss: 1.9383 - accuracy: 0.3000 - val_loss: 1.9289 -
val_accuracy: 0.3074
Epoch 23/50
196/196 [==============================] - 1s 5ms/step - loss: 1.8999 - accuracy: 0.3161 - val_loss: 1.8755 -
val_accuracy: 0.3336
Epoch 24/50
196/196 [==============================] - 1s 5ms/step - loss: 1.8485 - accuracy: 0.3342 - val_loss: 1.8046 -
val_accuracy: 0.3582
Epoch 25/50
196/196 [==============================] - 1s 5ms/step - loss: 1.7967 - accuracy: 0.3543 - val_loss: 1.7520 -
val_accuracy: 0.3743
Epoch 26/50
196/196 [==============================] - 1s 5ms/step - loss: 1.7601 - accuracy: 0.3680 - val_loss: 1.7411 -
val_accuracy: 0.3829
Epoch 27/50
196/196 [==============================] - 1s 5ms/step - loss: 1.7268 - accuracy: 0.3814 - val_loss: 1.7198 -
```

```
val_accuracy: 0.3900
Epoch 28/50
196/196 [==============================] - 1s 5ms/step - loss: 1.7009 - accuracy: 0.3895 - val_loss: 1.7013 -
val_accuracy: 0.3983
Epoch 29/50
196/196 [==============================] - 1s 5ms/step - loss: 1.6789 - accuracy: 0.3964 - val_loss: 1.6523 -
val_accuracy: 0.4072
Epoch 30/50
196/196 [==============================] - 1s 6ms/step - loss: 1.6625 - accuracy: 0.4012 - val_loss: 1.6240 -
val_accuracy: 0.4187
Epoch 31/50
196/196 [==============================] - 1s 5ms/step - loss: 1.6445 - accuracy: 0.4085 - val_loss: 1.6055 -
val_accuracy: 0.4236
Epoch 32/50
196/196 [==============================] - 1s 5ms/step - loss: 1.6296 - accuracy: 0.4147 - val_loss: 1.6151 -
val_accuracy: 0.4211
Epoch 33/50
196/196 [==============================] - 1s 5ms/step - loss: 1.6184 - accuracy: 0.4177 - val_loss: 1.5815 -
val_accuracy: 0.4312
Epoch 34/50
196/196 [==============================] - 1s 5ms/step - loss: 1.6056 - accuracy: 0.4233 - val_loss: 1.5785 -
val_accuracy: 0.4365
Epoch 35/50
196/196 [==============================] - 1s 5ms/step - loss: 1.5956 - accuracy: 0.4262 - val_loss: 1.5922 -
val_accuracy: 0.4368
Epoch 36/50
196/196 [==============================] - 1s 5ms/step - loss: 1.5845 - accuracy: 0.4296 - val_loss: 1.5657 -
val_accuracy: 0.4394
Epoch 37/50
196/196 [==============================] - 1s 5ms/step - loss: 1.5722 - accuracy: 0.4362 - val_loss: 1.5408 -
val_accuracy: 0.4464
Epoch 38/50
196/196 [==============================] - 1s 5ms/step - loss: 1.5623 - accuracy: 0.4367 - val_loss: 1.5313 -
val_accuracy: 0.4486
Epoch 39/50
196/196 [==============================] - 1s 5ms/step - loss: 1.5501 - accuracy: 0.4428 - val_loss: 1.5161 -
val_accuracy: 0.4546
Epoch 40/50
196/196 [==============================] - 1s 5ms/step - loss: 1.5377 - accuracy: 0.4458 - val_loss: 1.5024 -
val_accuracy: 0.4599
Epoch 41/50
196/196 [==============================] - 1s 5ms/step - loss: 1.5279 - accuracy: 0.4505 - val_loss: 1.4952 -
val_accuracy: 0.4615
```

```
Epoch 42/50
196/196 [==============================] - 1s 5ms/step - loss: 1.5135 - accuracy: 0.4566 - val_loss: 1.4811 -
val_accuracy: 0.4676
Epoch 43/50
196/196 [==============================] - 1s 5ms/step - loss: 1.4977 - accuracy: 0.4586 - val_loss: 1.4890 -
val_accuracy: 0.4612
Epoch 44/50
196/196 [==============================] - 1s 5ms/step - loss: 1.4856 - accuracy: 0.4634 - val_loss: 1.4706 -
val_accuracy: 0.4709
Epoch 45/50
196/196 [==============================] - 1s 5ms/step - loss: 1.4727 - accuracy: 0.4714 - val_loss: 1.4593 -
val_accuracy: 0.4745
Epoch 46/50
196/196 [==============================] - 1s 5ms/step - loss: 1.4619 - accuracy: 0.4743 - val_loss: 1.4283 -
val_accuracy: 0.4856
Epoch 47/50
196/196 [==============================] - 1s 5ms/step - loss: 1.4506 - accuracy: 0.4783 - val_loss: 1.4179 -
val_accuracy: 0.4908
Epoch 48/50
196/196 [==============================] - 1s 5ms/step - loss: 1.4385 - accuracy: 0.4845 - val_loss: 1.4123 -
val_accuracy: 0.4955
Epoch 49/50
196/196 [==============================] - 1s 5ms/step - loss: 1.4294 - accuracy: 0.4861 - val_loss: 1.4072 -
val_accuracy: 0.4937
Epoch 50/50
196/196 [==============================] - 1s 5ms/step - loss: 1.4202 - accuracy: 0.4921 - val_loss: 1.3893 -
val_accuracy: 0.5016
```

In [20]:
```
# Plot the loss curve
import matplotlib.pyplot as plt
%matplotlib inline

epochs = range(50)
train_acc = history.history['loss']
valid_acc = history.history['val_loss']
plt.plot(epochs, train_acc, 'bo', label='Training Loss')
plt.plot(epochs, valid_acc, 'r', label='Validation Loss')
plt.xlabel('Epochs')
plt.xlabel('Loss')
plt.legend()
plt.show()
```



6.3. Model 3

In [21]:
```
from keras.layers import Dense
from keras.layers.pooling import MaxPooling2D
from keras.layers.convolutional import Conv2D
from keras import models, layers

# Build the network
model_cnn = models.Sequential()
model_cnn.add(layers.Conv2D(32, (3, 3), activation='relu', input_shape=(32, 32, 3)))
model_cnn.add(layers.MaxPooling2D((2, 2)))
model_cnn.add(layers.Dropout(0.5))
model_cnn.add(layers.Conv2D(64,(4,4), activation='relu'))
model_cnn.add(layers.MaxPooling2D((2, 2)))
model_cnn.add(layers.Flatten())
model_cnn.add(layers.Dense(256, activation='relu'))
model_cnn.add(layers.Dense(10, activation='softmax'))

# Define model optimizer and loss function
from keras import optimizers
model_cnn.compile(optimizer='Ftrl', loss='categorical_crossentropy', metrics=['accuracy'])

# Train the model and store model parameters/loss values
history = model_cnn.fit(x_tr, y_tr, batch_size=256, epochs=50, validation_data=(x_val, y_val))
```

```
Epoch 1/50
157/157 [==============================] - 1s 7ms/step - loss: 5.4296 - accuracy: 0.1515 - val_loss: 2.2143
- val_accuracy: 0.1584
Epoch 2/50
157/157 [==============================] - 1s 5ms/step - loss: 2.1385 - accuracy: 0.2131 - val_loss: 2.3760
- val_accuracy: 0.1146
Epoch 3/50
157/157 [==============================] - 1s 6ms/step - loss: 2.0201 - accuracy: 0.2659 - val_loss: 2.3518
- val_accuracy: 0.1428
Epoch 4/50
157/157 [==============================] - 1s 6ms/step - loss: 1.9244 - accuracy: 0.3073 - val_loss: 2.3389
- val_accuracy: 0.1774
Epoch 5/50
157/157 [==============================] - 1s 6ms/step - loss: 1.8493 - accuracy: 0.3329 - val_loss: 2.0890
- val_accuracy: 0.2449
Epoch 6/50
157/157 [==============================] - 1s 5ms/step - loss: 1.7940 - accuracy: 0.3557 - val_loss: 1.9538
- val_accuracy: 0.2918
Epoch 7/50
```

```
157/157 [==============================] - 1s 6ms/step - loss: 1.7515 - accuracy: 0.3650 - val_loss: 1.8033
- val_accuracy: 0.3526
Epoch 8/50
157/157 [==============================] - 1s 5ms/step - loss: 1.7115 - accuracy: 0.3844 - val_loss: 1.7620
- val_accuracy: 0.3612
Epoch 9/50
157/157 [==============================] - 1s 5ms/step - loss: 1.6781 - accuracy: 0.3963 - val_loss: 1.7419
- val_accuracy: 0.3777
Epoch 10/50
157/157 [==============================] - 1s 5ms/step - loss: 1.6514 - accuracy: 0.4082 - val_loss: 1.8152
- val_accuracy: 0.3341
Epoch 11/50
157/157 [==============================] - 1s 5ms/step - loss: 1.6265 - accuracy: 0.4155 - val_loss: 1.7440
- val_accuracy: 0.3670
Epoch 12/50
157/157 [==============================] - 1s 5ms/step - loss: 1.6090 - accuracy: 0.4214 - val_loss: 1.6658
- val_accuracy: 0.3994
Epoch 13/50
157/157 [==============================] - 1s 6ms/step - loss: 1.5861 - accuracy: 0.4295 - val_loss: 1.6388
- val_accuracy: 0.4014
Epoch 14/50
157/157 [==============================] - 1s 5ms/step - loss: 1.5737 - accuracy: 0.4358 - val_loss: 1.5966
- val_accuracy: 0.4280
Epoch 15/50
157/157 [==============================] - 1s 6ms/step - loss: 1.5629 - accuracy: 0.4403 - val_loss: 1.5773
- val_accuracy: 0.4364
Epoch 16/50
157/157 [==============================] - 1s 5ms/step - loss: 1.5488 - accuracy: 0.4486 - val_loss: 1.6535
- val_accuracy: 0.4101
Epoch 17/50
157/157 [==============================] - 1s 5ms/step - loss: 1.5347 - accuracy: 0.4514 - val_loss: 1.5433
- val_accuracy: 0.4505
Epoch 18/50
157/157 [==============================] - 1s 5ms/step - loss: 1.5261 - accuracy: 0.4539 - val_loss: 1.6349
- val_accuracy: 0.4062
Epoch 19/50
157/157 [==============================] - 1s 5ms/step - loss: 1.5170 - accuracy: 0.4567 - val_loss: 1.5459
- val_accuracy: 0.4470
Epoch 20/50
157/157 [==============================] - 1s 5ms/step - loss: 1.5055 - accuracy: 0.4615 - val_loss: 1.5509
- val_accuracy: 0.4477
Epoch 21/50
157/157 [==============================] - 1s 5ms/step - loss: 1.4950 - accuracy: 0.4675 - val_loss: 1.5346
```

```
- val_accuracy: 0.4477
Epoch 22/50
157/157 [==============================] - 1s 5ms/step - loss: 1.4882 - accuracy: 0.4705 - val_loss: 1.4920
- val_accuracy: 0.4686
Epoch 23/50
157/157 [==============================] - 1s 5ms/step - loss: 1.4796 - accuracy: 0.4733 - val_loss: 1.5055
- val_accuracy: 0.4643
Epoch 24/50
157/157 [==============================] - 1s 5ms/step - loss: 1.4696 - accuracy: 0.4778 - val_loss: 1.5194
- val_accuracy: 0.4599
Epoch 25/50
157/157 [==============================] - 1s 5ms/step - loss: 1.4607 - accuracy: 0.4823 - val_loss: 1.5105
- val_accuracy: 0.4606
Epoch 26/50
157/157 [==============================] - 1s 5ms/step - loss: 1.4555 - accuracy: 0.4824 - val_loss: 1.4793
- val_accuracy: 0.4714
Epoch 27/50
157/157 [==============================] - 1s 5ms/step - loss: 1.4469 - accuracy: 0.4862 - val_loss: 1.4720
- val_accuracy: 0.4771
Epoch 28/50
157/157 [==============================] - 1s 5ms/step - loss: 1.4411 - accuracy: 0.4891 - val_loss: 1.4798
- val_accuracy: 0.4742
Epoch 29/50
157/157 [==============================] - 1s 5ms/step - loss: 1.4291 - accuracy: 0.4931 - val_loss: 1.4939
- val_accuracy: 0.4681
Epoch 30/50
157/157 [==============================] - 1s 5ms/step - loss: 1.4239 - accuracy: 0.4952 - val_loss: 1.4660
- val_accuracy: 0.4804
Epoch 31/50
157/157 [==============================] - 1s 5ms/step - loss: 1.4208 - accuracy: 0.4924 - val_loss: 1.5168
- val_accuracy: 0.4575
Epoch 32/50
157/157 [==============================] - 1s 5ms/step - loss: 1.4164 - accuracy: 0.4983 - val_loss: 1.4848
- val_accuracy: 0.4713
Epoch 33/50
157/157 [==============================] - 1s 5ms/step - loss: 1.4100 - accuracy: 0.5006 - val_loss: 1.4326
- val_accuracy: 0.4932
Epoch 34/50
157/157 [==============================] - 1s 5ms/step - loss: 1.4022 - accuracy: 0.5062 - val_loss: 1.4911
- val_accuracy: 0.4627
Epoch 35/50
157/157 [==============================] - 1s 5ms/step - loss: 1.3952 - accuracy: 0.5060 - val_loss: 1.4711
- val_accuracy: 0.4793
```

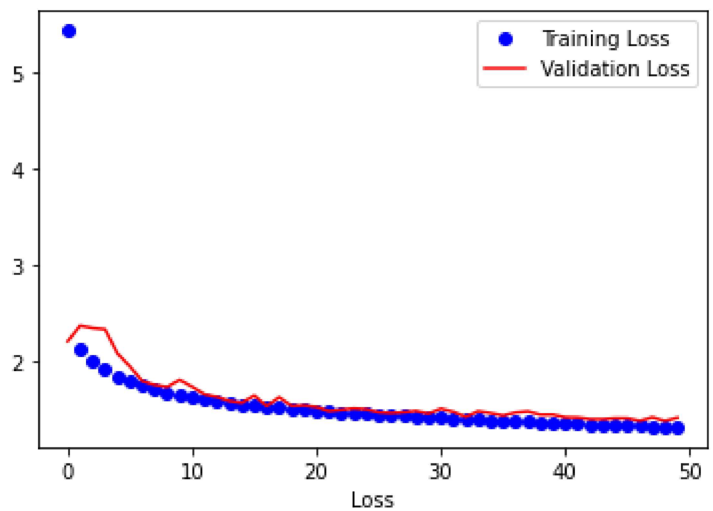
```
Epoch 36/50
157/157 [==============================] - 1s 5ms/step - loss: 1.3909 - accuracy: 0.5106 - val_loss: 1.4462
- val_accuracy: 0.4879
Epoch 37/50
157/157 [==============================] - 1s 5ms/step - loss: 1.3849 - accuracy: 0.5096 - val_loss: 1.4772
- val_accuracy: 0.4794
Epoch 38/50
157/157 [==============================] - 1s 5ms/step - loss: 1.3793 - accuracy: 0.5138 - val_loss: 1.4892
- val_accuracy: 0.4756
Epoch 39/50
157/157 [==============================] - 1s 5ms/step - loss: 1.3764 - accuracy: 0.5147 - val_loss: 1.4542
- val_accuracy: 0.4900
Epoch 40/50
157/157 [==============================] - 1s 5ms/step - loss: 1.3701 - accuracy: 0.5197 - val_loss: 1.4552
- val_accuracy: 0.4850
Epoch 41/50
157/157 [==============================] - 1s 5ms/step - loss: 1.3627 - accuracy: 0.5185 - val_loss: 1.4277
- val_accuracy: 0.4953
Epoch 42/50
157/157 [==============================] - 1s 5ms/step - loss: 1.3608 - accuracy: 0.5259 - val_loss: 1.4254
- val_accuracy: 0.4959
Epoch 43/50
157/157 [==============================] - 1s 5ms/step - loss: 1.3535 - accuracy: 0.5242 - val_loss: 1.4091
- val_accuracy: 0.4990
Epoch 44/50
157/157 [==============================] - 1s 5ms/step - loss: 1.3498 - accuracy: 0.5238 - val_loss: 1.4089
- val_accuracy: 0.4994
Epoch 45/50
157/157 [==============================] - 1s 5ms/step - loss: 1.3470 - accuracy: 0.5258 - val_loss: 1.4181
- val_accuracy: 0.5018
Epoch 46/50
157/157 [==============================] - 1s 5ms/step - loss: 1.3422 - accuracy: 0.5283 - val_loss: 1.4178
- val_accuracy: 0.5039
Epoch 47/50
157/157 [==============================] - 1s 5ms/step - loss: 1.3390 - accuracy: 0.5253 - val_loss: 1.3908
- val_accuracy: 0.5084
Epoch 48/50
157/157 [==============================] - 1s 5ms/step - loss: 1.3327 - accuracy: 0.5325 - val_loss: 1.4279
- val_accuracy: 0.4952
Epoch 49/50
157/157 [==============================] - 1s 5ms/step - loss: 1.3283 - accuracy: 0.5338 - val_loss: 1.3902
- val_accuracy: 0.5094
Epoch 50/50
```

```
157/157 [==============================] - 1s 5ms/step - loss: 1.3268 - accuracy: 0.5330 - val_loss: 1.4231
- val_accuracy: 0.5014
```

In [22]:

```python
# Plot the loss curve
import matplotlib.pyplot as plt
%matplotlib inline

epochs = range(50)
train_acc = history.history['loss']
valid_acc = history.history['val_loss']
plt.plot(epochs, train_acc, 'bo', label='Training Loss')
plt.plot(epochs, valid_acc, 'r', label='Validation Loss')
plt.xlabel('Epochs')
plt.xlabel('Loss')
plt.legend()
plt.show()
```



Evaluate with Test Set

In [23]:
```
model_cnn.compile(optimizer='Ftrl', loss='categorical_crossentropy', metrics=['accuracy'])

# Train the model and store model parameters/loss values
history = model_cnn.fit(x_train, y_train_vec, batch_size=256, epochs=50, validation_data = (x_test, y_test_vec))
```

```
Epoch 1/50
196/196 [==============================] - 2s 6ms/step - loss: 2.2977 - accuracy: 0.1032 - val_loss: 2.3025 -
val_accuracy: 0.1063
Epoch 2/50
196/196 [==============================] - 1s 5ms/step - loss: 2.3025 - accuracy: 0.1065 - val_loss: 2.3025 -
val_accuracy: 0.1001
Epoch 3/50
196/196 [==============================] - 1s 5ms/step - loss: 2.3024 - accuracy: 0.1021 - val_loss: 2.3022 -
val_accuracy: 0.1207
Epoch 4/50
196/196 [==============================] - 1s 5ms/step - loss: 2.3016 - accuracy: 0.1024 - val_loss: 2.3006 -
val_accuracy: 0.1000
Epoch 5/50
196/196 [==============================] - 1s 5ms/step - loss: 2.2937 - accuracy: 0.1001 - val_loss: 2.2782 -
val_accuracy: 0.1008
Epoch 6/50
196/196 [==============================] - 1s 5ms/step - loss: 2.2234 - accuracy: 0.1391 - val_loss: 2.1665 -
val_accuracy: 0.1863
Epoch 7/50
196/196 [==============================] - 1s 5ms/step - loss: 2.0888 - accuracy: 0.2060 - val_loss: 2.0508 -
val_accuracy: 0.2337
Epoch 8/50
196/196 [==============================] - 1s 5ms/step - loss: 2.0403 - accuracy: 0.2342 - val_loss: 2.0202 -
val_accuracy: 0.2373
Epoch 9/50
196/196 [==============================] - 1s 5ms/step - loss: 2.0114 - accuracy: 0.2523 - val_loss: 1.9995 -
val_accuracy: 0.2579
Epoch 10/50
196/196 [==============================] - 1s 5ms/step - loss: 1.9883 - accuracy: 0.2664 - val_loss: 1.9789 -
val_accuracy: 0.2760
Epoch 11/50
196/196 [==============================] - 1s 5ms/step - loss: 1.9691 - accuracy: 0.2756 - val_loss: 1.9507 -
val_accuracy: 0.2849
Epoch 12/50
196/196 [==============================] - 1s 5ms/step - loss: 1.9495 - accuracy: 0.2827 - val_loss: 1.9293 -
val_accuracy: 0.2922
Epoch 13/50
```

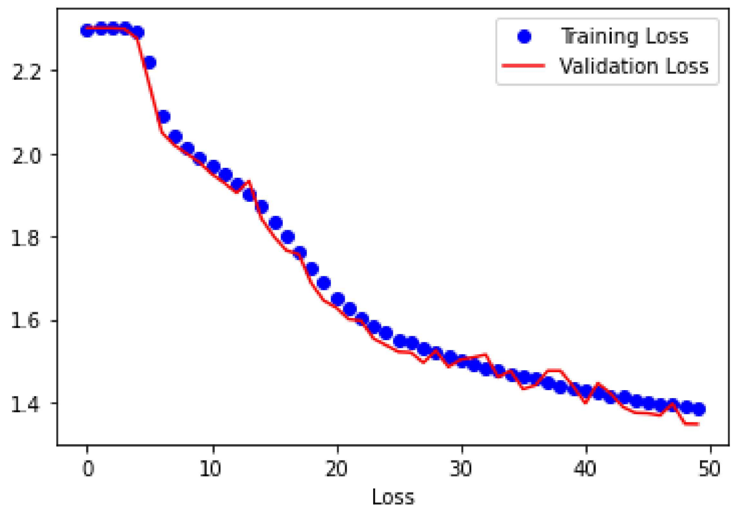
```
196/196 [==============================] - 1s 5ms/step - loss: 1.9283 - accuracy: 0.2915 - val_loss: 1.9048 -
val_accuracy: 0.2991
Epoch 14/50
196/196 [==============================] - 1s 5ms/step - loss: 1.9042 - accuracy: 0.3012 - val_loss: 1.9340 -
val_accuracy: 0.2875
Epoch 15/50
196/196 [==============================] - 1s 5ms/step - loss: 1.8728 - accuracy: 0.3163 - val_loss: 1.8424 -
val_accuracy: 0.3273
Epoch 16/50
196/196 [==============================] - 1s 5ms/step - loss: 1.8368 - accuracy: 0.3292 - val_loss: 1.7996 -
val_accuracy: 0.3422
Epoch 17/50
196/196 [==============================] - 1s 5ms/step - loss: 1.8013 - accuracy: 0.3458 - val_loss: 1.7659 -
val_accuracy: 0.3523
Epoch 18/50
196/196 [==============================] - 1s 5ms/step - loss: 1.7636 - accuracy: 0.3636 - val_loss: 1.7568 -
val_accuracy: 0.3664
Epoch 19/50
196/196 [==============================] - 1s 5ms/step - loss: 1.7231 - accuracy: 0.3802 - val_loss: 1.6871 -
val_accuracy: 0.3875
Epoch 20/50
196/196 [==============================] - 1s 5ms/step - loss: 1.6893 - accuracy: 0.3948 - val_loss: 1.6452 -
val_accuracy: 0.4089
Epoch 21/50
196/196 [==============================] - 1s 5ms/step - loss: 1.6526 - accuracy: 0.4037 - val_loss: 1.6282 -
val_accuracy: 0.4136
Epoch 22/50
196/196 [==============================] - 1s 5ms/step - loss: 1.6259 - accuracy: 0.4133 - val_loss: 1.6001 -
val_accuracy: 0.4267
Epoch 23/50
196/196 [==============================] - 1s 5ms/step - loss: 1.6040 - accuracy: 0.4221 - val_loss: 1.5963 -
val_accuracy: 0.4234
Epoch 24/50
196/196 [==============================] - 1s 5ms/step - loss: 1.5844 - accuracy: 0.4296 - val_loss: 1.5532 -
val_accuracy: 0.4399
Epoch 25/50
196/196 [==============================] - 1s 5ms/step - loss: 1.5685 - accuracy: 0.4366 - val_loss: 1.5373 -
val_accuracy: 0.4487
Epoch 26/50
196/196 [==============================] - 1s 5ms/step - loss: 1.5519 - accuracy: 0.4448 - val_loss: 1.5215 -
val_accuracy: 0.4523
Epoch 27/50
196/196 [==============================] - 1s 5ms/step - loss: 1.5424 - accuracy: 0.4468 - val_loss: 1.5198 -
```

```
val_accuracy: 0.4539
Epoch 28/50
196/196 [==============================] - 1s 5ms/step - loss: 1.5311 - accuracy: 0.4522 - val_loss: 1.4952 -
val_accuracy: 0.4641
Epoch 29/50
196/196 [==============================] - 1s 5ms/step - loss: 1.5194 - accuracy: 0.4536 - val_loss: 1.5245 -
val_accuracy: 0.4532
Epoch 30/50
196/196 [==============================] - 1s 5ms/step - loss: 1.5122 - accuracy: 0.4582 - val_loss: 1.4849 -
val_accuracy: 0.4685
Epoch 31/50
196/196 [==============================] - 1s 5ms/step - loss: 1.5014 - accuracy: 0.4606 - val_loss: 1.5029 -
val_accuracy: 0.4589
Epoch 32/50
196/196 [==============================] - 1s 5ms/step - loss: 1.4929 - accuracy: 0.4640 - val_loss: 1.5079 -
val_accuracy: 0.4569
Epoch 33/50
196/196 [==============================] - 1s 5ms/step - loss: 1.4839 - accuracy: 0.4688 - val_loss: 1.5148 -
val_accuracy: 0.4455
Epoch 34/50
196/196 [==============================] - 1s 5ms/step - loss: 1.4770 - accuracy: 0.4726 - val_loss: 1.4603 -
val_accuracy: 0.4722
Epoch 35/50
196/196 [==============================] - 1s 5ms/step - loss: 1.4680 - accuracy: 0.4729 - val_loss: 1.4756 -
val_accuracy: 0.4639
Epoch 36/50
196/196 [==============================] - 1s 5ms/step - loss: 1.4634 - accuracy: 0.4760 - val_loss: 1.4313 -
val_accuracy: 0.4783
Epoch 37/50
196/196 [==============================] - 1s 5ms/step - loss: 1.4560 - accuracy: 0.4765 - val_loss: 1.4409 -
val_accuracy: 0.4822
Epoch 38/50
196/196 [==============================] - 1s 5ms/step - loss: 1.4470 - accuracy: 0.4824 - val_loss: 1.4759 -
val_accuracy: 0.4686
Epoch 39/50
196/196 [==============================] - 1s 5ms/step - loss: 1.4399 - accuracy: 0.4846 - val_loss: 1.4758 -
val_accuracy: 0.4645
Epoch 40/50
196/196 [==============================] - 1s 5ms/step - loss: 1.4350 - accuracy: 0.4870 - val_loss: 1.4392 -
val_accuracy: 0.4799
Epoch 41/50
196/196 [==============================] - 1s 5ms/step - loss: 1.4302 - accuracy: 0.4861 - val_loss: 1.3970 -
val_accuracy: 0.4991
```

```
Epoch 42/50
196/196 [==============================] - 1s 5ms/step - loss: 1.4237 - accuracy: 0.4919 - val_loss: 1.4457 -
val_accuracy: 0.4783
Epoch 43/50
196/196 [==============================] - 1s 5ms/step - loss: 1.4148 - accuracy: 0.4953 - val_loss: 1.4218 -
val_accuracy: 0.4862
Epoch 44/50
196/196 [==============================] - 1s 5ms/step - loss: 1.4134 - accuracy: 0.4943 - val_loss: 1.3888 -
val_accuracy: 0.5006
Epoch 45/50
196/196 [==============================] - 1s 5ms/step - loss: 1.4067 - accuracy: 0.4985 - val_loss: 1.3741 -
val_accuracy: 0.5038
Epoch 46/50
196/196 [==============================] - 1s 5ms/step - loss: 1.3983 - accuracy: 0.5010 - val_loss: 1.3729 -
val_accuracy: 0.5077
Epoch 47/50
196/196 [==============================] - 1s 5ms/step - loss: 1.3964 - accuracy: 0.5037 - val_loss: 1.3680 -
val_accuracy: 0.5092
Epoch 48/50
196/196 [==============================] - 1s 5ms/step - loss: 1.3936 - accuracy: 0.5035 - val_loss: 1.3959 -
val_accuracy: 0.4954
Epoch 49/50
196/196 [==============================] - 1s 5ms/step - loss: 1.3890 - accuracy: 0.5060 - val_loss: 1.3476 -
val_accuracy: 0.5157
Epoch 50/50
196/196 [==============================] - 1s 5ms/step - loss: 1.3832 - accuracy: 0.5057 - val_loss: 1.3468 -
val_accuracy: 0.5183
```

In [24]:
```
# Plot the loss curve
import matplotlib.pyplot as plt
%matplotlib inline

epochs = range(50)
train_acc = history.history['loss']
valid_acc = history.history['val_loss']
plt.plot(epochs, train_acc, 'bo', label='Training Loss')
plt.plot(epochs, valid_acc, 'r', label='Validation Loss')
plt.xlabel('Epochs')
plt.xlabel('Loss')
plt.legend()
plt.show()
```



6.4. Model 4

In [25]:
```
from keras.layers import Dense
from keras.layers.pooling import MaxPooling2D
from keras.layers.convolutional import Conv2D
from keras import models, layers

# Build the network
model_cnn = models.Sequential()
model_cnn.add(layers.Conv2D(32, (3, 3), activation='relu', input_shape=(32, 32, 3)))
model_cnn.add(layers.MaxPooling2D((2, 2)))
model_cnn.add(layers.Dropout(0.2))
model_cnn.add(layers.Conv2D(64,(4,4), activation='relu'))
model_cnn.add(layers.MaxPooling2D((2, 2)))
model_cnn.add(layers.Dropout(0.2))
model_cnn.add(layers.Flatten())
model_cnn.add(layers.Dense(256, activation='relu'))
model_cnn.add(layers.Dropout(0.2))
model_cnn.add(layers.Dense(10, activation='softmax'))

# Define model optimizer and loss function
from keras import optimizers
model_cnn.compile(optimizer='Ftrl', loss='categorical_crossentropy', metrics=['accuracy'])

# Train the model and store model parameters/loss values
history = model_cnn.fit(x_tr, y_tr, batch_size=256, epochs=50, validation_data=(x_val, y_val))
```

```
Epoch 1/50
157/157 [==============================] - 2s 7ms/step - loss: 3.9939 - accuracy: 0.1728 - val_loss: 2.1619 -
val_accuracy: 0.1534
Epoch 2/50
157/157 [==============================] - 1s 5ms/step - loss: 2.0255 - accuracy: 0.2519 - val_loss: 2.0291 -
val_accuracy: 0.2517
Epoch 3/50
157/157 [==============================] - 1s 5ms/step - loss: 1.9426 - accuracy: 0.2903 - val_loss: 1.9300 -
val_accuracy: 0.2839
Epoch 4/50
157/157 [==============================] - 1s 5ms/step - loss: 1.8833 - accuracy: 0.3166 - val_loss: 1.9494 -
val_accuracy: 0.2786
Epoch 5/50
157/157 [==============================] - 1s 6ms/step - loss: 1.8313 - accuracy: 0.3391 - val_loss: 1.8844 -
val_accuracy: 0.3135
Epoch 6/50
157/157 [==============================] - 1s 5ms/step - loss: 1.7843 - accuracy: 0.3544 - val_loss: 1.8415 -
```

```
val_accuracy: 0.3163
Epoch 7/50
157/157 [==============================] - 1s 6ms/step - loss: 1.7456 - accuracy: 0.3703 - val_loss: 1.7628 -
val_accuracy: 0.3572
Epoch 8/50
157/157 [==============================] - 1s 5ms/step - loss: 1.7106 - accuracy: 0.3826 - val_loss: 1.7232 -
val_accuracy: 0.3704
Epoch 9/50
157/157 [==============================] - 1s 5ms/step - loss: 1.6804 - accuracy: 0.3911 - val_loss: 1.6629 -
val_accuracy: 0.4016
Epoch 10/50
157/157 [==============================] - 1s 5ms/step - loss: 1.6582 - accuracy: 0.3991 - val_loss: 1.7087 -
val_accuracy: 0.3879
Epoch 11/50
157/157 [==============================] - 1s 5ms/step - loss: 1.6366 - accuracy: 0.4099 - val_loss: 1.5929 -
val_accuracy: 0.4380
Epoch 12/50
157/157 [==============================] - 1s 5ms/step - loss: 1.6124 - accuracy: 0.4180 - val_loss: 1.5841 -
val_accuracy: 0.4390
Epoch 13/50
157/157 [==============================] - 1s 5ms/step - loss: 1.5974 - accuracy: 0.4218 - val_loss: 1.5628 -
val_accuracy: 0.4371
Epoch 14/50
157/157 [==============================] - 1s 5ms/step - loss: 1.5850 - accuracy: 0.4293 - val_loss: 1.5411 -
val_accuracy: 0.4482
Epoch 15/50
157/157 [==============================] - 1s 5ms/step - loss: 1.5738 - accuracy: 0.4283 - val_loss: 1.5576 -
val_accuracy: 0.4355
Epoch 16/50
157/157 [==============================] - 1s 5ms/step - loss: 1.5613 - accuracy: 0.4337 - val_loss: 1.5413 -
val_accuracy: 0.4306
Epoch 17/50
157/157 [==============================] - 1s 5ms/step - loss: 1.5468 - accuracy: 0.4403 - val_loss: 1.5155 -
val_accuracy: 0.4555
Epoch 18/50
157/157 [==============================] - 1s 5ms/step - loss: 1.5417 - accuracy: 0.4429 - val_loss: 1.4876 -
val_accuracy: 0.4646
Epoch 19/50
157/157 [==============================] - 1s 6ms/step - loss: 1.5265 - accuracy: 0.4519 - val_loss: 1.5026 -
val_accuracy: 0.4487
Epoch 20/50
157/157 [==============================] - 1s 5ms/step - loss: 1.5195 - accuracy: 0.4516 - val_loss: 1.4948 -
val_accuracy: 0.4602
```

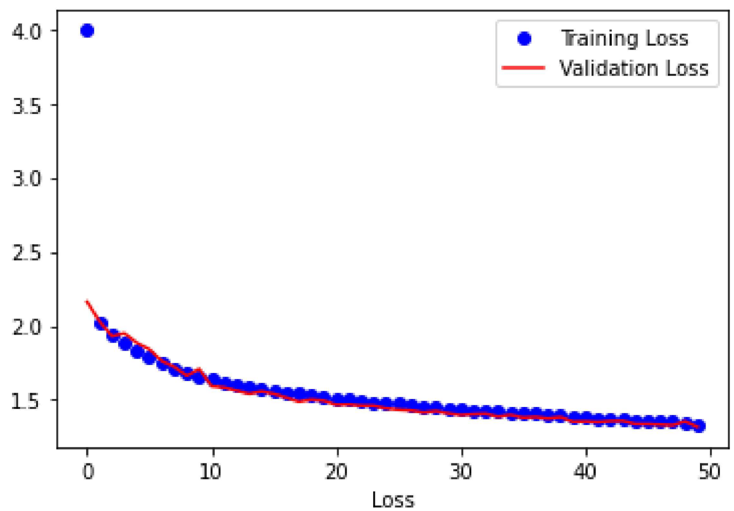
```
Epoch 21/50
157/157 [==============================] - 1s 5ms/step - loss: 1.5116 - accuracy: 0.4535 - val_loss: 1.4672 -
val_accuracy: 0.4657
Epoch 22/50
157/157 [==============================] - 1s 5ms/step - loss: 1.5035 - accuracy: 0.4579 - val_loss: 1.4693 -
val_accuracy: 0.4619
Epoch 23/50
157/157 [==============================] - 1s 6ms/step - loss: 1.4952 - accuracy: 0.4606 - val_loss: 1.4615 -
val_accuracy: 0.4583
Epoch 24/50
157/157 [==============================] - 1s 6ms/step - loss: 1.4853 - accuracy: 0.4633 - val_loss: 1.4608 -
val_accuracy: 0.4618
Epoch 25/50
157/157 [==============================] - 1s 6ms/step - loss: 1.4839 - accuracy: 0.4667 - val_loss: 1.4433 -
val_accuracy: 0.4758
Epoch 26/50
157/157 [==============================] - 1s 6ms/step - loss: 1.4734 - accuracy: 0.4708 - val_loss: 1.4340 -
val_accuracy: 0.4740
Epoch 27/50
157/157 [==============================] - 1s 5ms/step - loss: 1.4659 - accuracy: 0.4726 - val_loss: 1.4292 -
val_accuracy: 0.4768
Epoch 28/50
157/157 [==============================] - 1s 5ms/step - loss: 1.4581 - accuracy: 0.4759 - val_loss: 1.4152 -
val_accuracy: 0.4837
Epoch 29/50
157/157 [==============================] - 1s 5ms/step - loss: 1.4526 - accuracy: 0.4782 - val_loss: 1.4245 -
val_accuracy: 0.4764
Epoch 30/50
157/157 [==============================] - 1s 5ms/step - loss: 1.4440 - accuracy: 0.4811 - val_loss: 1.4096 -
val_accuracy: 0.4848
Epoch 31/50
157/157 [==============================] - 1s 6ms/step - loss: 1.4389 - accuracy: 0.4833 - val_loss: 1.3978 -
val_accuracy: 0.4899
Epoch 32/50
157/157 [==============================] - 1s 5ms/step - loss: 1.4284 - accuracy: 0.4868 - val_loss: 1.4038 -
val_accuracy: 0.4910
Epoch 33/50
157/157 [==============================] - 1s 5ms/step - loss: 1.4311 - accuracy: 0.4863 - val_loss: 1.4070 -
val_accuracy: 0.4831
Epoch 34/50
157/157 [==============================] - 1s 5ms/step - loss: 1.4230 - accuracy: 0.4895 - val_loss: 1.3869 -
val_accuracy: 0.4933
Epoch 35/50
```

```
157/157 [==============================] - 1s 5ms/step - loss: 1.4114 - accuracy: 0.4895 - val_loss: 1.3990 -
val_accuracy: 0.4895
Epoch 36/50
157/157 [==============================] - 1s 5ms/step - loss: 1.4094 - accuracy: 0.4934 - val_loss: 1.3802 -
val_accuracy: 0.4964
Epoch 37/50
157/157 [==============================] - 1s 6ms/step - loss: 1.4057 - accuracy: 0.4970 - val_loss: 1.3850 -
val_accuracy: 0.4962
Epoch 38/50
157/157 [==============================] - 1s 5ms/step - loss: 1.4029 - accuracy: 0.4978 - val_loss: 1.3744 -
val_accuracy: 0.4955
Epoch 39/50
157/157 [==============================] - 1s 5ms/step - loss: 1.3940 - accuracy: 0.4996 - val_loss: 1.3840 -
val_accuracy: 0.4907
Epoch 40/50
157/157 [==============================] - 1s 5ms/step - loss: 1.3888 - accuracy: 0.5040 - val_loss: 1.3534 -
val_accuracy: 0.5090
Epoch 41/50
157/157 [==============================] - 1s 5ms/step - loss: 1.3786 - accuracy: 0.5068 - val_loss: 1.3550 -
val_accuracy: 0.5067
Epoch 42/50
157/157 [==============================] - 1s 5ms/step - loss: 1.3770 - accuracy: 0.5081 - val_loss: 1.3505 -
val_accuracy: 0.5094
Epoch 43/50
157/157 [==============================] - 1s 6ms/step - loss: 1.3737 - accuracy: 0.5092 - val_loss: 1.3573 -
val_accuracy: 0.5059
Epoch 44/50
157/157 [==============================] - 1s 5ms/step - loss: 1.3659 - accuracy: 0.5112 - val_loss: 1.3593 -
val_accuracy: 0.5050
Epoch 45/50
157/157 [==============================] - 1s 5ms/step - loss: 1.3638 - accuracy: 0.5132 - val_loss: 1.3376 -
val_accuracy: 0.5097
Epoch 46/50
157/157 [==============================] - 1s 5ms/step - loss: 1.3580 - accuracy: 0.5125 - val_loss: 1.3372 -
val_accuracy: 0.5078
Epoch 47/50
157/157 [==============================] - 1s 5ms/step - loss: 1.3608 - accuracy: 0.5125 - val_loss: 1.3335 -
val_accuracy: 0.5163
Epoch 48/50
157/157 [==============================] - 1s 5ms/step - loss: 1.3506 - accuracy: 0.5189 - val_loss: 1.3293 -
val_accuracy: 0.5174
Epoch 49/50
157/157 [==============================] - 1s 6ms/step - loss: 1.3480 - accuracy: 0.5177 - val_loss: 1.3551 -
```

```
val_accuracy: 0.5052
Epoch 50/50
157/157 [==============================] - 1s 5ms/step - loss: 1.3365 - accuracy: 0.5214 - val_loss: 1.3176 -
val_accuracy: 0.5182
```

In [26]:
```
# Plot the loss curve
import matplotlib.pyplot as plt
%matplotlib inline

epochs = range(50)
train_acc = history.history['loss']
valid_acc = history.history['val_loss']
plt.plot(epochs, train_acc, 'bo', label='Training Loss')
plt.plot(epochs, valid_acc, 'r', label='Validation Loss')
plt.xlabel('Epochs')
plt.xlabel('Loss')
plt.legend()
plt.show()
```



Evaluate with Test Set

In [27]:
```
model_cnn.compile(optimizer='Ftrl', loss='categorical_crossentropy', metrics=['accuracy'])

# Train the model and store model parameters/loss values
history = model_cnn.fit(x_train, y_train_vec, batch_size=256, epochs=100, validation_data = (x_test, y_test_vec)
```

```
Epoch 1/100
196/196 [==============================] - 2s 6ms/step - loss: 2.2975 - accuracy: 0.1011 - val_loss: 2.3026 -
val_accuracy: 0.1000
Epoch 2/100
196/196 [==============================] - 1s 5ms/step - loss: 2.3026 - accuracy: 0.0965 - val_loss: 2.3026 -
val_accuracy: 0.1000
Epoch 3/100
196/196 [==============================] - 1s 6ms/step - loss: 2.3026 - accuracy: 0.0969 - val_loss: 2.3026 -
val_accuracy: 0.1000
Epoch 4/100
196/196 [==============================] - 1s 5ms/step - loss: 2.3026 - accuracy: 0.0975 - val_loss: 2.3026 -
val_accuracy: 0.1000
Epoch 5/100
196/196 [==============================] - 1s 5ms/step - loss: 2.3026 - accuracy: 0.0978 - val_loss: 2.3026 -
val_accuracy: 0.1000
Epoch 6/100
196/196 [==============================] - 1s 5ms/step - loss: 2.3026 - accuracy: 0.0979 - val_loss: 2.3026 -
val_accuracy: 0.1000
Epoch 7/100
196/196 [==============================] - 1s 5ms/step - loss: 2.3026 - accuracy: 0.0987 - val_loss: 2.3026 -
val_accuracy: 0.1000
Epoch 8/100
196/196 [==============================] - 1s 5ms/step - loss: 2.3026 - accuracy: 0.0961 - val_loss: 2.3026 -
val_accuracy: 0.1000
Epoch 9/100
196/196 [==============================] - 1s 5ms/step - loss: 2.3026 - accuracy: 0.0979 - val_loss: 2.3026 -
val_accuracy: 0.1000
Epoch 10/100
196/196 [==============================] - 1s 5ms/step - loss: 2.3026 - accuracy: 0.0990 - val_loss: 2.3026 -
val_accuracy: 0.1000
Epoch 11/100
196/196 [==============================] - 1s 5ms/step - loss: 2.3026 - accuracy: 0.0971 - val_loss: 2.3026 -
val_accuracy: 0.1000
Epoch 12/100
196/196 [==============================] - 1s 5ms/step - loss: 2.3026 - accuracy: 0.0971 - val_loss: 2.3026 -
val_accuracy: 0.1000
Epoch 13/100
```

```
196/196 [==============================] - 1s 5ms/step - loss: 2.3026 - accuracy: 0.0986 - val_loss: 2.3026 -
val_accuracy: 0.1000
Epoch 14/100
196/196 [==============================] - 1s 5ms/step - loss: 2.3026 - accuracy: 0.0991 - val_loss: 2.3026 -
val_accuracy: 0.1000
Epoch 15/100
196/196 [==============================] - 1s 5ms/step - loss: 2.3026 - accuracy: 0.0984 - val_loss: 2.3026 -
val_accuracy: 0.1000
Epoch 16/100
196/196 [==============================] - 1s 5ms/step - loss: 2.3026 - accuracy: 0.0960 - val_loss: 2.3026 -
val_accuracy: 0.1000
Epoch 17/100
196/196 [==============================] - 1s 5ms/step - loss: 2.3026 - accuracy: 0.0990 - val_loss: 2.3026 -
val_accuracy: 0.1000
Epoch 18/100
196/196 [==============================] - 1s 5ms/step - loss: 2.3026 - accuracy: 0.0978 - val_loss: 2.3026 -
val_accuracy: 0.0990
Epoch 19/100
196/196 [==============================] - 1s 5ms/step - loss: 2.3026 - accuracy: 0.0983 - val_loss: 2.3026 -
val_accuracy: 0.1000
Epoch 20/100
196/196 [==============================] - 1s 5ms/step - loss: 2.3026 - accuracy: 0.0979 - val_loss: 2.3026 -
val_accuracy: 0.1000
Epoch 21/100
196/196 [==============================] - 1s 5ms/step - loss: 2.3026 - accuracy: 0.0983 - val_loss: 2.3026 -
val_accuracy: 0.1000
Epoch 22/100
196/196 [==============================] - 1s 5ms/step - loss: 2.3026 - accuracy: 0.0976 - val_loss: 2.3026 -
val_accuracy: 0.1000
Epoch 23/100
196/196 [==============================] - 1s 5ms/step - loss: 2.3026 - accuracy: 0.0987 - val_loss: 2.3026 -
val_accuracy: 0.1000
Epoch 24/100
196/196 [==============================] - 1s 5ms/step - loss: 2.3026 - accuracy: 0.0971 - val_loss: 2.3026 -
val_accuracy: 0.1000
Epoch 25/100
196/196 [==============================] - 1s 5ms/step - loss: 2.3026 - accuracy: 0.0988 - val_loss: 2.3026 -
val_accuracy: 0.1000
Epoch 26/100
196/196 [==============================] - 1s 5ms/step - loss: 2.3026 - accuracy: 0.0989 - val_loss: 2.3026 -
val_accuracy: 0.1000
Epoch 27/100
196/196 [==============================] - 1s 5ms/step - loss: 2.3026 - accuracy: 0.0985 - val_loss: 2.3026 -
```

```
val_accuracy: 0.1000
Epoch 28/100
196/196 [==============================] - 1s 5ms/step - loss: 2.3026 - accuracy: 0.0990 - val_loss: 2.3026 -
val_accuracy: 0.1000
Epoch 29/100
196/196 [==============================] - 1s 5ms/step - loss: 2.3026 - accuracy: 0.0989 - val_loss: 2.3026 -
val_accuracy: 0.1000
Epoch 30/100
196/196 [==============================] - 1s 5ms/step - loss: 2.3026 - accuracy: 0.1000 - val_loss: 2.3026 -
val_accuracy: 0.1000
Epoch 31/100
196/196 [==============================] - 1s 5ms/step - loss: 2.3026 - accuracy: 0.0996 - val_loss: 2.3026 -
val_accuracy: 0.1000
Epoch 32/100
196/196 [==============================] - 1s 6ms/step - loss: 2.3026 - accuracy: 0.1000 - val_loss: 2.3026 -
val_accuracy: 0.1000
Epoch 33/100
196/196 [==============================] - 1s 5ms/step - loss: 2.3026 - accuracy: 0.0998 - val_loss: 2.3026 -
val_accuracy: 0.1000
Epoch 34/100
196/196 [==============================] - 1s 5ms/step - loss: 2.3026 - accuracy: 0.0994 - val_loss: 2.3026 -
val_accuracy: 0.1000
Epoch 35/100
196/196 [==============================] - 1s 5ms/step - loss: 2.3026 - accuracy: 0.0995 - val_loss: 2.3026 -
val_accuracy: 0.1000
Epoch 36/100
196/196 [==============================] - 1s 5ms/step - loss: 2.3026 - accuracy: 0.1004 - val_loss: 2.3026 -
val_accuracy: 0.1000
Epoch 37/100
196/196 [==============================] - 1s 5ms/step - loss: 2.3026 - accuracy: 0.0997 - val_loss: 2.3026 -
val_accuracy: 0.1000
Epoch 38/100
196/196 [==============================] - 1s 5ms/step - loss: 2.3026 - accuracy: 0.1026 - val_loss: 2.3026 -
val_accuracy: 0.1000
Epoch 39/100
196/196 [==============================] - 1s 5ms/step - loss: 2.3026 - accuracy: 0.0999 - val_loss: 2.3026 -
val_accuracy: 0.1000
Epoch 40/100
196/196 [==============================] - 1s 5ms/step - loss: 2.3026 - accuracy: 0.0996 - val_loss: 2.3026 -
val_accuracy: 0.1000
Epoch 41/100
196/196 [==============================] - 1s 5ms/step - loss: 2.3026 - accuracy: 0.0999 - val_loss: 2.3026 -
val_accuracy: 0.1000
```

```
Epoch 42/100
196/196 [==============================] - 1s 5ms/step - loss: 2.3026 - accuracy: 0.1008 - val_loss: 2.3026 -
val_accuracy: 0.1000
Epoch 43/100
196/196 [==============================] - 1s 5ms/step - loss: 2.3026 - accuracy: 0.1011 - val_loss: 2.3026 -
val_accuracy: 0.1000
Epoch 44/100
196/196 [==============================] - 1s 5ms/step - loss: 2.3026 - accuracy: 0.1049 - val_loss: 2.3026 -
val_accuracy: 0.1000
Epoch 45/100
196/196 [==============================] - 1s 5ms/step - loss: 2.3026 - accuracy: 0.1025 - val_loss: 2.3026 -
val_accuracy: 0.1000
Epoch 46/100
196/196 [==============================] - 1s 5ms/step - loss: 2.3026 - accuracy: 0.0999 - val_loss: 2.3026 -
val_accuracy: 0.1000
Epoch 47/100
196/196 [==============================] - 1s 6ms/step - loss: 2.3026 - accuracy: 0.0999 - val_loss: 2.3026 -
val_accuracy: 0.1000
Epoch 48/100
196/196 [==============================] - 1s 6ms/step - loss: 2.3026 - accuracy: 0.1058 - val_loss: 2.3026 -
val_accuracy: 0.1000
Epoch 49/100
196/196 [==============================] - 1s 5ms/step - loss: 2.3026 - accuracy: 0.1075 - val_loss: 2.3026 -
val_accuracy: 0.1000
Epoch 50/100
196/196 [==============================] - 1s 5ms/step - loss: 2.3026 - accuracy: 0.1045 - val_loss: 2.3026 -
val_accuracy: 0.1000
Epoch 51/100
196/196 [==============================] - 1s 6ms/step - loss: 2.3026 - accuracy: 0.1031 - val_loss: 2.3026 -
val_accuracy: 0.1000
Epoch 52/100
196/196 [==============================] - 1s 5ms/step - loss: 2.3026 - accuracy: 0.1102 - val_loss: 2.3026 -
val_accuracy: 0.1070
Epoch 53/100
196/196 [==============================] - 1s 5ms/step - loss: 2.3026 - accuracy: 0.1117 - val_loss: 2.3026 -
val_accuracy: 0.1246
Epoch 54/100
196/196 [==============================] - 1s 5ms/step - loss: 2.3026 - accuracy: 0.1156 - val_loss: 2.3026 -
val_accuracy: 0.1366
Epoch 55/100
196/196 [==============================] - 1s 6ms/step - loss: 2.3026 - accuracy: 0.1114 - val_loss: 2.3026 -
val_accuracy: 0.1304
Epoch 56/100
```

```
196/196 [==============================] - 1s 6ms/step - loss: 2.3026 - accuracy: 0.1194 - val_loss: 2.3026 -
val_accuracy: 0.1360
Epoch 57/100
196/196 [==============================] - 1s 6ms/step - loss: 2.3026 - accuracy: 0.1156 - val_loss: 2.3026 -
val_accuracy: 0.1217

Epoch 58/100
196/196 [==============================] - 1s 5ms/step - loss: 2.3026 - accuracy: 0.1127 - val_loss: 2.3026
- val_accuracy: 0.1078
Epoch 59/100
196/196 [==============================] - 1s 5ms/step - loss: 2.3026 - accuracy: 0.1154 - val_loss: 2.3026
- val_accuracy: 0.1019
Epoch 60/100
196/196 [==============================] - 1s 5ms/step - loss: 2.3026 - accuracy: 0.1044 - val_loss: 2.3026
- val_accuracy: 0.1003
Epoch 61/100
196/196 [==============================] - 1s 5ms/step - loss: 2.3026 - accuracy: 0.1028 - val_loss: 2.3026
- val_accuracy: 0.1000
Epoch 62/100
196/196 [==============================] - 1s 5ms/step - loss: 2.3025 - accuracy: 0.0999 - val_loss: 2.3025
- val_accuracy: 0.1000
Epoch 63/100
196/196 [==============================] - 1s 5ms/step - loss: 2.3025 - accuracy: 0.1005 - val_loss: 2.3025
- val_accuracy: 0.1003
Epoch 64/100
196/196 [==============================] - 1s 5ms/step - loss: 2.3025 - accuracy: 0.1002 - val_loss: 2.3025
- val_accuracy: 0.0991
Epoch 65/100
196/196 [==============================] - 1s 6ms/step - loss: 2.3024 - accuracy: 0.0991 - val_loss: 2.3024
- val_accuracy: 0.0994
Epoch 66/100
196/196 [==============================] - 1s 5ms/step - loss: 2.3023 - accuracy: 0.0988 - val_loss: 2.3021
- val_accuracy: 0.0972
Epoch 67/100
196/196 [==============================] - 1s 5ms/step - loss: 2.3015 - accuracy: 0.1014 - val_loss: 2.3005
- val_accuracy: 0.1023
Epoch 68/100
196/196 [==============================] - 1s 5ms/step - loss: 2.2976 - accuracy: 0.1023 - val_loss: 2.2930
- val_accuracy: 0.1010
Epoch 69/100
196/196 [==============================] - 1s 5ms/step - loss: 2.2826 - accuracy: 0.1081 - val_loss: 2.2609
- val_accuracy: 0.1379
Epoch 70/100
```

```
196/196 [==============================] - 1s 5ms/step - loss: 2.2264 - accuracy: 0.1517 - val_loss: 2.2300
- val_accuracy: 0.1585
Epoch 71/100
196/196 [==============================] - 1s 5ms/step - loss: 2.2058 - accuracy: 0.1635 - val_loss: 2.1948
- val_accuracy: 0.1710
Epoch 72/100
196/196 [==============================] - 1s 5ms/step - loss: 2.1965 - accuracy: 0.1695 - val_loss: 2.1842
- val_accuracy: 0.1810
Epoch 73/100
196/196 [==============================] - 1s 5ms/step - loss: 2.1841 - accuracy: 0.1777 - val_loss: 2.1696
- val_accuracy: 0.1870
Epoch 74/100
196/196 [==============================] - 1s 5ms/step - loss: 2.1721 - accuracy: 0.1862 - val_loss: 2.1583
- val_accuracy: 0.2032
Epoch 75/100
196/196 [==============================] - 1s 6ms/step - loss: 2.1597 - accuracy: 0.1952 - val_loss: 2.1530
- val_accuracy: 0.2002
Epoch 76/100
196/196 [==============================] - 1s 6ms/step - loss: 2.1443 - accuracy: 0.2021 - val_loss: 2.1202
- val_accuracy: 0.2123
Epoch 77/100
196/196 [==============================] - 1s 6ms/step - loss: 2.1167 - accuracy: 0.2152 - val_loss: 2.0653
- val_accuracy: 0.2389
Epoch 78/100
196/196 [==============================] - 1s 6ms/step - loss: 2.0392 - accuracy: 0.2400 - val_loss: 1.9831
- val_accuracy: 0.2811
Epoch 79/100
196/196 [==============================] - 1s 6ms/step - loss: 1.9895 - accuracy: 0.2679 - val_loss: 1.9459
- val_accuracy: 0.2879
Epoch 80/100
196/196 [==============================] - 1s 6ms/step - loss: 1.9632 - accuracy: 0.2811 - val_loss: 1.9276
- val_accuracy: 0.2943
Epoch 81/100
196/196 [==============================] - 1s 6ms/step - loss: 1.9475 - accuracy: 0.2887 - val_loss: 1.9146
- val_accuracy: 0.3078
Epoch 82/100
196/196 [==============================] - 1s 6ms/step - loss: 1.9321 - accuracy: 0.2958 - val_loss: 1.8961
- val_accuracy: 0.3153
Epoch 83/100
196/196 [==============================] - 1s 6ms/step - loss: 1.9199 - accuracy: 0.3011 - val_loss: 1.8842
- val_accuracy: 0.3200
Epoch 84/100
196/196 [==============================] - 1s 6ms/step - loss: 1.9063 - accuracy: 0.3081 - val_loss: 1.8813
```

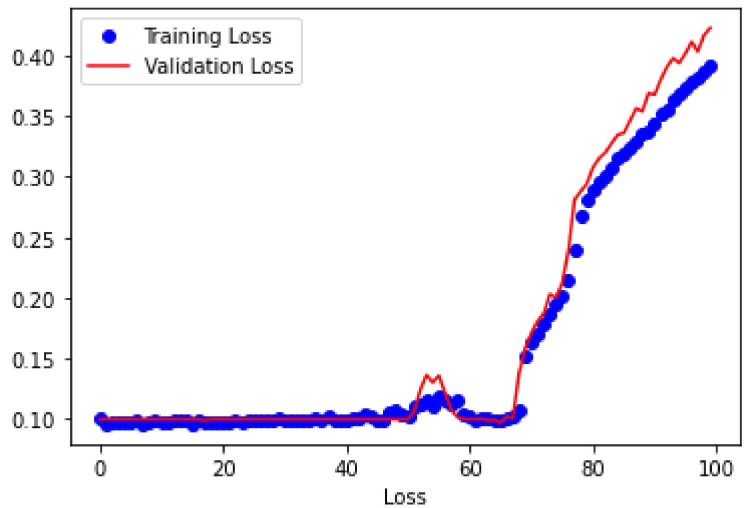
```
- val_accuracy: 0.3275
Epoch 85/100
196/196 [==============================] - 1s 6ms/step - loss: 1.8912 - accuracy: 0.3149 - val_loss: 1.8508
- val_accuracy: 0.3345
Epoch 86/100
196/196 [==============================] - 1s 5ms/step - loss: 1.8792 - accuracy: 0.3190 - val_loss: 1.8432
- val_accuracy: 0.3362
Epoch 87/100
196/196 [==============================] - 1s 5ms/step - loss: 1.8665 - accuracy: 0.3244 - val_loss: 1.8269
- val_accuracy: 0.3462
Epoch 88/100
196/196 [==============================] - 1s 5ms/step - loss: 1.8567 - accuracy: 0.3288 - val_loss: 1.8106
- val_accuracy: 0.3564
Epoch 89/100
196/196 [==============================] - 1s 6ms/step - loss: 1.8432 - accuracy: 0.3348 - val_loss: 1.8039
- val_accuracy: 0.3539
Epoch 90/100
196/196 [==============================] - 1s 5ms/step - loss: 1.8302 - accuracy: 0.3368 - val_loss: 1.7737
- val_accuracy: 0.3689
Epoch 91/100
196/196 [==============================] - 1s 5ms/step - loss: 1.8152 - accuracy: 0.3442 - val_loss: 1.7686
- val_accuracy: 0.3678
Epoch 92/100
196/196 [==============================] - 1s 6ms/step - loss: 1.7974 - accuracy: 0.3513 - val_loss: 1.7443
- val_accuracy: 0.3803
Epoch 93/100
196/196 [==============================] - 1s 6ms/step - loss: 1.7787 - accuracy: 0.3561 - val_loss: 1.7146
- val_accuracy: 0.3907
Epoch 94/100
196/196 [==============================] - 1s 6ms/step - loss: 1.7641 - accuracy: 0.3641 - val_loss: 1.7096
- val_accuracy: 0.3978
Epoch 95/100
196/196 [==============================] - 1s 5ms/step - loss: 1.7511 - accuracy: 0.3687 - val_loss: 1.7040
- val_accuracy: 0.3939
Epoch 96/100
196/196 [==============================] - 1s 6ms/step - loss: 1.7377 - accuracy: 0.3730 - val_loss: 1.6725
- val_accuracy: 0.4018
Epoch 97/100
196/196 [==============================] - 1s 5ms/step - loss: 1.7254 - accuracy: 0.3787 - val_loss: 1.6575
- val_accuracy: 0.4114
Epoch 98/100
196/196 [==============================] - 1s 6ms/step - loss: 1.7151 - accuracy: 0.3825 - val_loss: 1.6834
- val_accuracy: 0.4031
```

```
Epoch 99/100
196/196 [==============================] - 1s 5ms/step - loss: 1.7135 - accuracy: 0.3859 - val_loss: 1.6459
- val_accuracy: 0.4168
Epoch 100/100
196/196 [==============================] - 1s 5ms/step - loss: 1.6976 - accuracy: 0.3911 - val_loss: 1.6251
- val_accuracy: 0.4226
```

In [28]:

```python
# Plot the loss curve
import matplotlib.pyplot as plt
%matplotlib inline

epochs = range(100)
train_acc = history.history['accuracy']
valid_acc = history.history['val_accuracy']
plt.plot(epochs, train_acc, 'bo', label='Training Loss')
plt.plot(epochs, valid_acc, 'r', label='Validation Loss')
plt.xlabel('Epochs')
plt.xlabel('Loss')
plt.legend()
plt.show()
```



In [ ]: