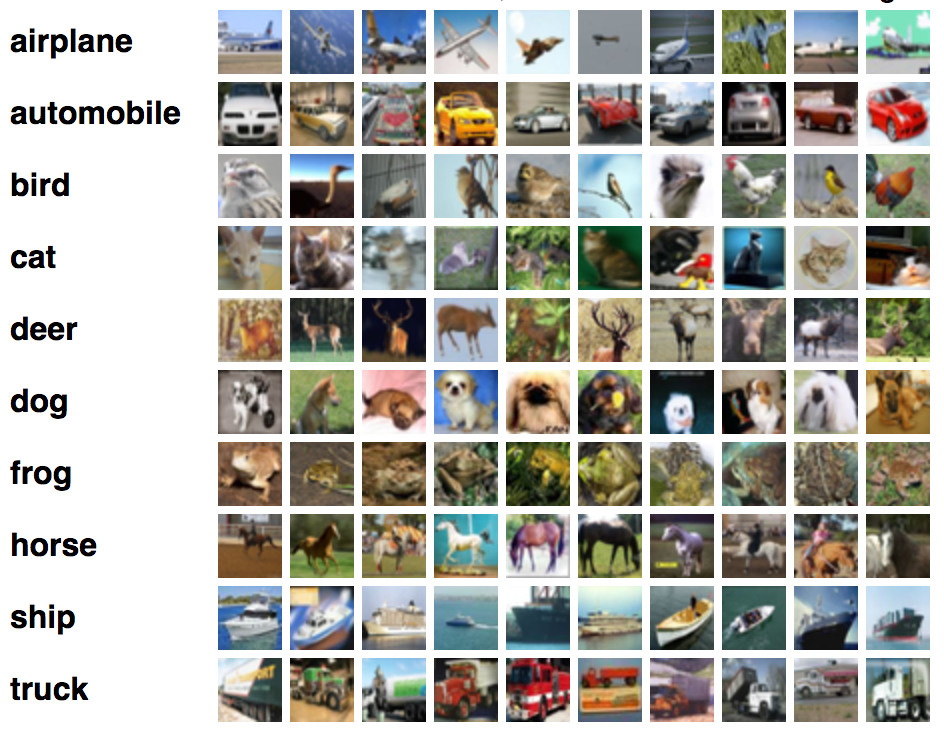
Object Recognition in Cifar-10 using Convolutional Neural Networks



Submitted BY: fARHAN HASSAN

Machine Learning | 11-6-2018

# Object Recognition

Object recognition is a computer vision technique for identifying objects in images or videos. Object recognition is a key output of deep learning and machine learning algorithms. Object recognition — determining what objects are where in a digital image — is a central research topic in computer vision.

Object recognition as a process for identifying or knowing the nature of an object (in an image or video frame). Recognition (applications) can be based on matching, learning, or pattern recognition algorithms with the goal being to label (classify) an object—to ask the question: what is the object? The goal is to teach a computer to do what comes naturally to humans: to gain a level of understanding of what an image contains

## Cifar 10

The CIFAR-10 dataset is a collection of images that are commonly used to train machine learning and computer vision algorithms. It is one of the most widely used datasets for machine learning research.The CIFAR-10 dataset contains 60,000 32x32 color images in 10 different classes. The 10 different classes represent airplanes, cars, birds, cats, deer, dogs, frogs, horses, ships, and trucks. There are 6,000 images of each class.

Here are the classes in the dataset, as well as 10 random images from each:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| airplane | https://www.cs.toronto.edu/~kriz/cifar-10-sample/airplane1.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/airplane2.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/airplane3.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/airplane4.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/airplane5.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/airplane6.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/airplane7.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/airplane8.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/airplane9.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/airplane10.png |
| automobile | https://www.cs.toronto.edu/~kriz/cifar-10-sample/automobile1.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/automobile2.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/automobile3.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/automobile4.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/automobile5.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/automobile6.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/automobile7.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/automobile8.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/automobile9.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/automobile10.png |
| bird | https://www.cs.toronto.edu/~kriz/cifar-10-sample/bird1.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/bird2.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/bird3.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/bird4.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/bird5.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/bird6.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/bird7.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/bird8.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/bird9.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/bird10.png |
| cat | https://www.cs.toronto.edu/~kriz/cifar-10-sample/cat1.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/cat2.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/cat3.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/cat4.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/cat5.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/cat6.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/cat7.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/cat8.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/cat9.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/cat10.png |
| deer | https://www.cs.toronto.edu/~kriz/cifar-10-sample/deer1.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/deer2.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/deer3.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/deer4.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/deer5.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/deer6.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/deer7.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/deer8.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/deer9.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/deer10.png |
| dog | https://www.cs.toronto.edu/~kriz/cifar-10-sample/dog1.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/dog2.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/dog3.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/dog4.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/dog5.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/dog6.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/dog7.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/dog8.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/dog9.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/dog10.png |
| frog | https://www.cs.toronto.edu/~kriz/cifar-10-sample/frog1.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/frog2.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/frog3.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/frog4.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/frog5.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/frog6.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/frog7.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/frog8.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/frog9.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/frog10.png |
| horse | https://www.cs.toronto.edu/~kriz/cifar-10-sample/horse1.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/horse2.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/horse3.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/horse4.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/horse5.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/horse6.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/horse7.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/horse8.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/horse9.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/horse10.png |
| ship | https://www.cs.toronto.edu/~kriz/cifar-10-sample/ship1.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/ship2.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/ship3.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/ship4.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/ship5.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/ship6.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/ship7.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/ship8.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/ship9.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/ship10.png |
| truck | https://www.cs.toronto.edu/~kriz/cifar-10-sample/truck1.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/truck2.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/truck3.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/truck4.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/truck5.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/truck6.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/truck7.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/truck8.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/truck9.png | https://www.cs.toronto.edu/~kriz/cifar-10-sample/truck10.png |

# Libraries:

My programming language of choice for this project was python and major part of the project was done using keras.

# Model Architecture:

For this particular problem I have chosen a very simple yet effective Convolution neural network.

# Convolution:

A convolution is the integral measuring how much two functions overlap as one passes over the other. The Convolutional layer is the core building block of a Convolutional Network. My model contains total four Convolutional layers, each with a filter size of 3x3. Convolutional networks take those filters, slices of the image’s feature space, and map them one by one; that is, they create a map of each place that feature occurs. By learning different portions of a feature space, convolutional nets allow for easily scalable and robust feature engineering.

# Activation Function:

An activation function is just a node that you add to the output end of any neural network. It is used to determine the output of the neural network like yes, no or 0, 1. As we are dealing with a classification problem in which the model needs to output a class, the activation function of choice is ReLu.

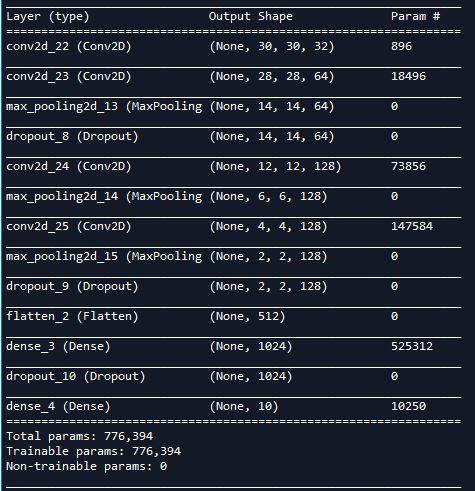
# Pooling:

A Pooling layer is inserted between convolutional layers. Its function is to progressively reduce the spatial size of the representation to reduce the amount of parameters and computation in the network, and hence to also control overfitting. I have applied Maxpooling with a filter size of 2x2

# Dropout:

The main idea behind dropout is to randomly drop units along with their connections from the network during training. It prevents the units from coa-dopting. It is basically used to overcome overfitting problem.

Following is a summary of my CNN model. It shows all the number of layers used and the number of parameters



# Model Parameters:

# Loss:

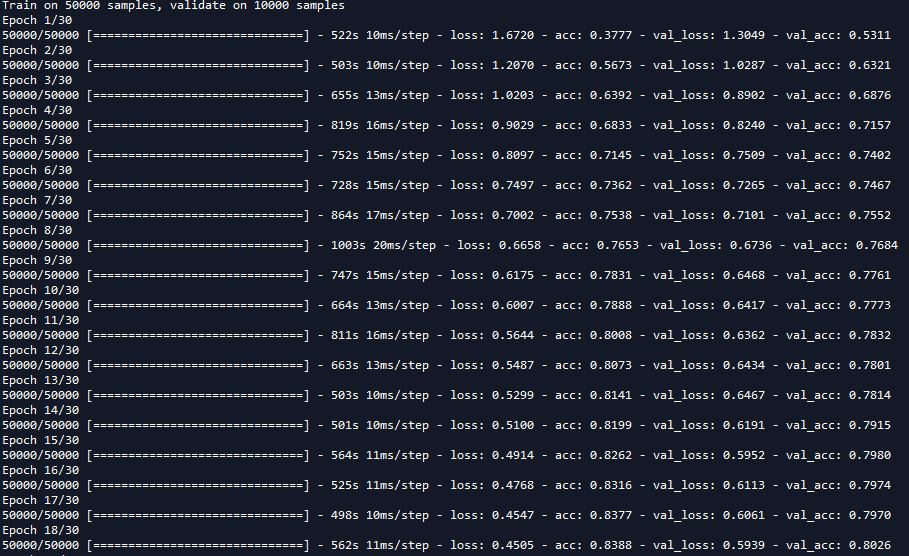
Cross-entropy loss, or log loss, measures the performance of a classification model whose output is a probability value between 0 and 1. As the given problem is a 10 class classification problem therefore the better choice is categorical cross entropy.

# Optimizer:

Adam is an optimization algorithm that can used instead of the classical stochastic gradient descent procedure to update network weights iterative based in training data. A learning rate is maintained for each network weight and separately adapted as learning unfolds. The learning rate I have selected is 0.001.

# Training:

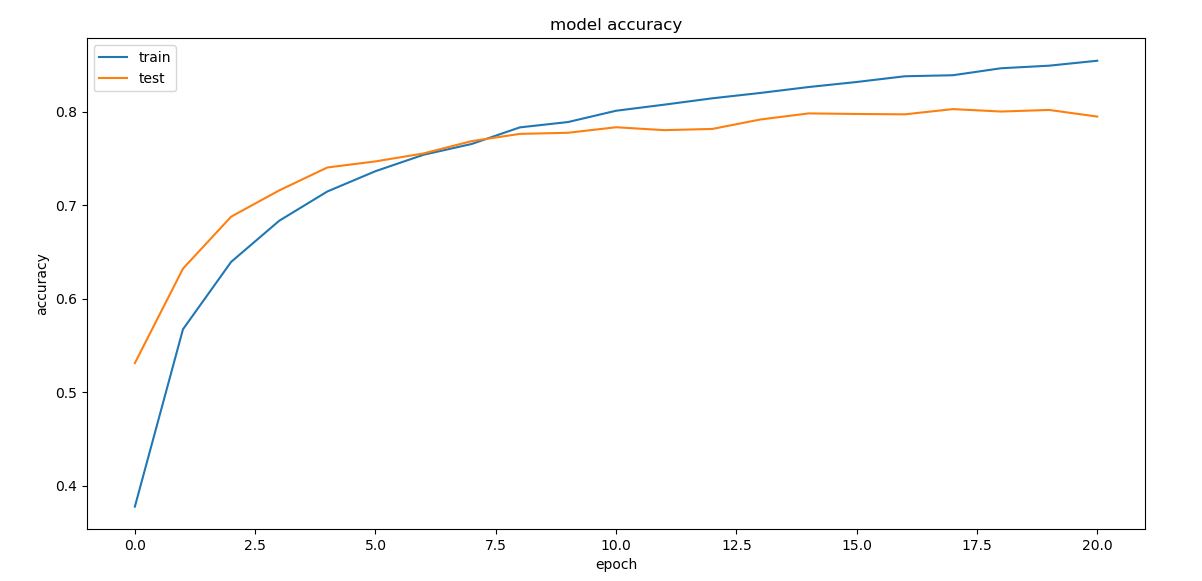
I have trained my model for 30 epochs with a batch size of 128



# Results:

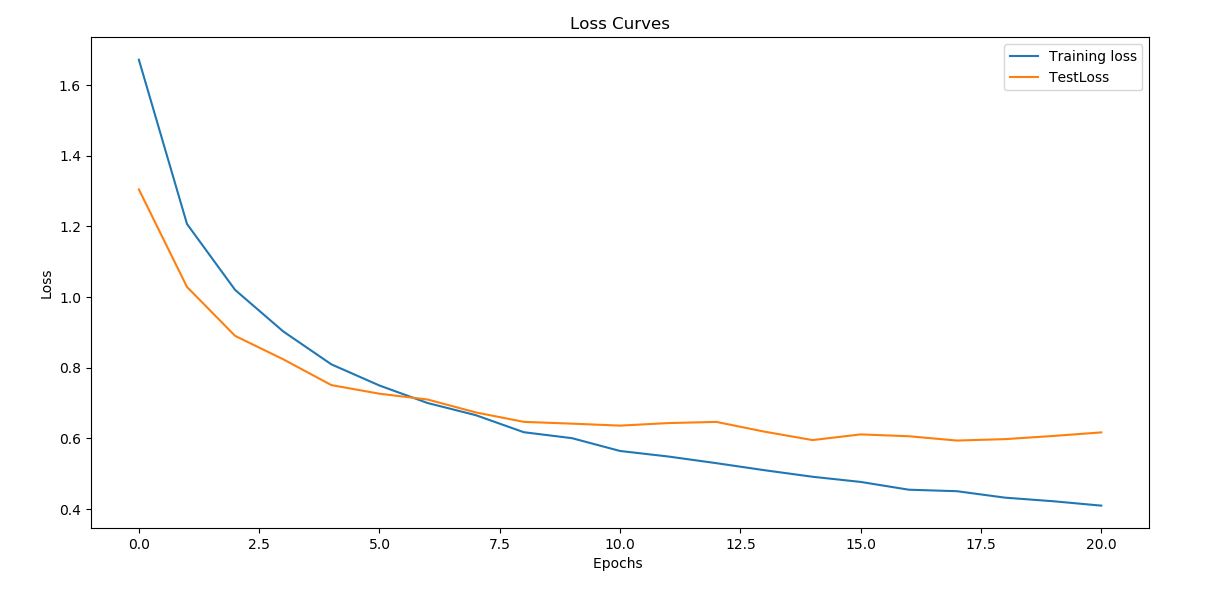
The training starts with the first epoch ending at 0.377 accuracy and a loss of 1.6720. As the epochs continue, both the training and test accuracy increases gradually and the loss decreases

Accuracy



The above plot demonstrates the relationship between accuracy and number of epochs. As the number of epochs increases, both the training and test set accuracy increases.

Loss



This plot shows the relationship between training and test loss and epochs. It shows that the both the losses decreases as the number of epochs increases.



The maximum test set accuracy I have achieved from this model is about 79% and the loss is about 0.617