Abstract

Target image recognition is the technique in which we want to input an image to a classifier model and the output tells whether the input image is the target image or not. We want to train our classifier model over a large set of pair of input image and corresponding label. If we train our classifier model over enough dataset, we can expect our classifier model to classify the input image correctly with the help of learnt parameters. The target image in this project is the image of animal cat. The dataset used in the project is a labelled dataset containing image information and the label of the image. Thus, the learning mechanism is called supervised learning. The classifier model in our project is a Neural Network model to recognize images of animal cats than that of non-cats. The learning algorithm used in the project is a logistic regression which is a binary classifier, i.e. it can classify the input data into one of two classes. In my approach, I have modelled a Four-Layer-Deep Neural Network to learn features of images over the training dataset for a number of iterations and I have tested my Neural Network model on the test dataset achieving an accuracy of 80%.

Keywords: Target image recognition, Classifier model, Supervised learning, Neural network, Logistic regression.

Introduction

1. What is Neural Network?

Neural Network is a network of neurons connected together in order to achieve complex tasks such as image recognition, voice recognition, object detection and so on. Neurons are basically the building blocks of the Neural Networks. They take input from the data, compute some function and provide final output or output to the neurons in the next layer. Many neurons can be stacked together in several layers in order to perform more complex tasks. Such a Neural Network is known as a Deep Neural Network. The process of training a Deep Neural Network is known as Deep Learning. Neural Networks, when trained on a large number of input data and output label pair, are extremely good at coming up with a mapping function which can map input data to the output labels.

2. Parameters of Neural Network

Each neuron in the Neural Network is associated with some parameters, namely weight and bias. These parameters are initially randomly initialized. Using some learning algorithm and some cost function corresponding to it, we can update the parameters such that the cost function is minimized with every iteration of our training process, and at the end of our training, we have our learnt parameters using which we can classify the input data as target data or non-target data. We train our Neural Network for sufficiently large number of iterations until our cost function is minimized. The learning algorithm used in the project is the logistic regression learning algorithm, the cost function used is the logistic regression cost function and the data is the image.

3. **How images are stored in computer?**

   Images are stored in computer as three-dimensional matrices. Each dimension corresponds to one of the three colour channels – Red, Green and Blue, called as RGB channels. Each channel is a two-dimensional matrix of dimensions equal to width and height of the image in pixels. The entry in each matrix cell is the pixel intensity value of that pixel corresponding to the colour channel, ranging from 0 to 255.

4. **Input feature vector X**

   We convert the three-dimensional image matrix into a single column vector of dimension [(height * width * 3), 1] where height and width are image height and width in pixels. For a number of training set images, these column vectors are stacked together column-wise making the input feature matrix X of dimension [nx, m] where "nx" is the dimension of each image feature vector and "m" is the number of training set images.

5. **Standardizing the dataset**

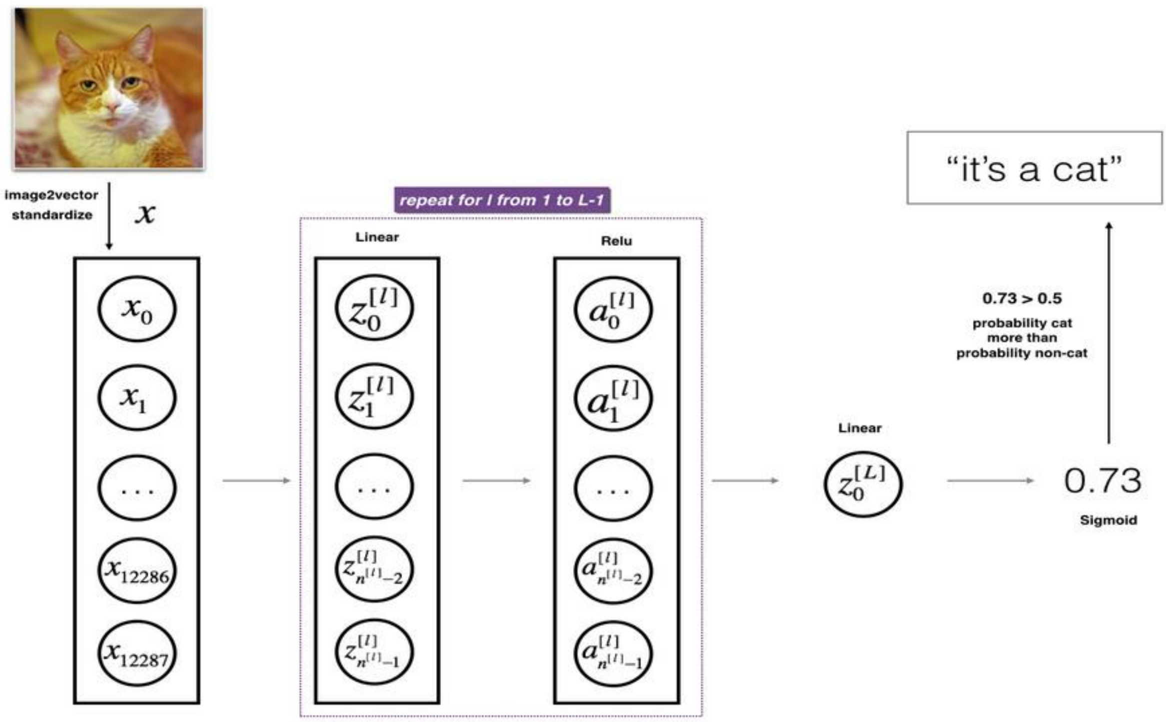
   Standardizing the features around the center and 0 with a standard deviation of 1 is important when we compare measurements that have different units. Variables that are measured at different scales do not contribute equally to the analysis and might end up creating a bias. For example, A variable that ranges between 0 and 1000 will outweigh a variable that ranges between 0 and 1. Using these variables without standardization will give the variable with the larger range weight of 1000 in the analysis. Transforming the data to comparable scales can prevent this problem. Typical data standardization procedures equalize the range and/or data variability.

   To represent colour images, the red, green and blue channels (RGB) must be specified for each pixel, and so the pixel value is actually a vector of three numbers ranging from 0 to 255. One common pre-processing step in machine learning is to center and standardize your dataset, meaning that you subtract the mean of the whole numpy array from each example, and then divide each example by the standard deviation of the whole numpy array. But for the image datasets, it is simpler and more convenient and works almost as well to just divide every row of the dataset by 255 (which is the maximum value of a pixel channel).

6. **Hyperparameters**

   Hyperparameters are the parameters which influence the value of our model parameters. These include – learning rate (α), number of iterations for which we want to train our Neural Network model, number of hidden layers in our Neural Network, number of hidden units in a particular layer, choice of activation function and so on.

7. Architecture of the model



- The input is a (64,64,3) image which is flattened to a vector of size (12288,1).
- The corresponding vector: $[x_0, x_1, ..., x_{12287}]^T$ is then multiplied by the weight matrix $W^{[1]}$ and then you add the intercept $b^{[1]}$. The result is called the linear unit.
- Next, we take the relu of the linear unit. This process could be repeated several times for each ($W^{[l]}$, $b^{[l]}$) depending on the model architecture.
- Finally, we take the sigmoid of the final linear unit. If it is greater than 0.5, we classify it to be a cat.

8. General methodology

1. Initialize parameters / Define hyperparameters.
2. Loop for number of iterations:
   a. Forward propagation
   b. Compute cost function
   c. Backward propagation
   d. Update parameters (using parameters, and grads from backprop)
4. Use trained parameters to predict labels.