Assignment
Building a Convolution Neural Network in R

A. Nagar

CS 6301
Spring 2019

Instructions

- This assignment involves downloading image dataset of your choice from Imagenet image repository http://www.image-net.org/
- You should store your image dataset under your account in the UTD server and your code should run on Google Cloud. Do not submit the images on eLearning,
- You are allowed to work in teams of maximum 4 students. Please write the names and NetIDs of each group member on the cover page.
  Only 1 final submission per team.
- **You have a total of 4 free late days for the entire semester. You can use at most 2 days for any one assignment. After four days have been used up, there will be a penalty of 10% for each late day. The submission for this assignment will be closed 2 days after the due date.**
- Please ask all questions on Piazza, not via email.

1 Getting Data

In this assignment, you will work with the images from the ImageNet project, which is available at `http://www.image-net.org`. You have to select an image category having sufficient number of images.

Good examples are:

- Any valid category under **dog**
`http://www.image-net.org/search?q=dog`
- Any valid category under **house**
`http://www.image-net.org/search?q=house`
- You can also browse through the collection and choose another valid category

Next, you will download the images and split them in the ratio 80:20 for training and test parts. You need to place these images in separate folder called **train** and **test** on your account under the UTD website. You can easily do it by putting data under your public_html folder in the Unix server. Be sure to give view permission for everyone.

2 Creating CNN model using Keras

After getting the data, you can do any pre-processing required, such as making them the same size, etc.

Next you will create a Keras model using alternating Conv2D and maxpool layers. It is up to you to design the network, including things such as input image size, output size, number of layers, etc

You need to tune as many of the following parameters as possibleL:

1. Number of layers
2. Kernel (filter size)
3. Number of filters in each layer
4. How many neurons in the last dense layer
5. Activation function
6. Error function (such as binary_crossentropy, etc)
7. Batch size

An additional important requirement is that your code must run on Google Cloud.

2.1 Output

You need to output the following from your code:

1. History plots showing training and testing accuracy and loss as a function of number of iterations. This is automatically generated by the system.
2. Example of at least 25 images arranged in 5x5 from test dataset, showing the following
   - Image
   - True Label
   - Predicted Label
3. A table containing details of parameter testing and tuning. Example:

Iteration	**Parameters**	**Training and Test Accuracy**
1.	Number of layers = ... Kernel Size Layer 1= .... Activation Function = ...	Train = 80% and Test = 78%
.....	......	....