MIT CSAIL

6.869 Advances in Computer Vision

Fall 2018

Miniplaces Challenge: Part1

Posted: Thursday, November 1, 2018 **Due:** Thursday 23:59, November 8, 2018

Please submit a report named your_kerberos.pdf.

Late Submission Policy: We do not accept late submissions. The submission deadline has a 50-minute soft cut-off; after midnight Thursday, submissions are penalized 2% per minute late.

Collaborators: You are free to discuss problems with other students but all writing must be done individually. Please list all collaborators at the top of your report.

Readings: Pg. 200-220 (Section 6.5) from the Deep Learning Book, freely available at http://www.deeplearningbook.org/contents/mlp.html.

Problem 1 *Convolution Layers*

Convolution layer is the most popular module in computer vision tasks. In this question, you will derive the equations for its forward and backward propagations.

(a) Consider your input x_{in} and output x_{out} are both 1-D signals with the same dimension N, and your kernel W has size k. Find the equation for forward propagation.

(b) Consider the back propagation process, with learning rate η, and the gradients from the last layer is $\frac{\partial C}{\partial x_{out}}$. Find the gradients of the input $\frac{\partial C}{\partial x_{in}}$, and the update rule for the kernel weights W^{i+1}.

(c) Discuss how you handle the boundaries and explain your choice.

Problem 2 *Testing with a pre-trained model*

Pre-training a CNN model on a large dataset is important for many downstream computer vision tasks. In this problem, we will use AlexNet and ResNet, which have achieved great success in computer vision community in the past few years. We have slightly modified them so that they take 128 by 128 images as input. The provided models have been trained on the Miniplace training set and can be found inside the "models" folder of the code base.

(a) Fill in line 73 of "test.py" to complete the code. Show what the AlexNet and ResNet predict for the "sample.jpg" image. Besides, try your own image and check what they predict (Resize your own image to make height and width equal). Please include the one line code, prediction results for "sample.jpg" and your own image in the report.

(b) Download the validation set which includes 10,000 images. Evaluate the prediction accuracy of AlexNet and ResNet on the validation set. Include the accuracy in the report. For those who do not have GPU resources, feed-forwarding 10,000 images one by one takes around 20 minutes for ResNet and 10 minutes for AlexNet on a typical Macbook with only CPU. You might expect similar time consumption in your own laptop.

(c) For a given image, you can extract its feature from ResNet by invoking the "get_features" function, which is around line 134 in "models/ResNet.py". For a pair of images, the Euclidean distance of their features can be used to measure their similarity. Try to find 5 nearest neighbours of your own image from the Miniplace validation set. Include the results in the report.

Problem 3 *Deep Dream (optional)*

Visualizing pre-trained model is a way to understand what the model learns. Instead of visualizing the pre-trained CNN features of a particular layer, we can generate psychedelic-looking images by maximizing the activations of a certain layer. Deep Dream was introduced by Google Research in a blog post. See original gallery for more examples

We can feed a base image to the pre-trained CNN. Then, forward pass is done till a particular layer. Now, to get a sense of what that layer has learned, we need to maximize the activations through that layer.

Choose your favorite deep learning framework(TensorFlow or PyTorch, and we have provide both starter code), fill the missing block under # TODO, run the code and attach the resulting image to your report.

We also upload the code to Google Colab(Google's Jupyter), you can run it within Google Colab if you want to use Google's free GPU.

TensorFlow: https://colab.research.google.com/drive/13r6USsZyEfbdKxpX6Us0DFOcaJ6RDV87
PyTorch: https://colab.research.google.com/drive/1Rn6fTNrAuxtBYVpyA3QGCFfVNRADLJnV