MIT CSAIL

6.869 Advances in Computer Vision

Fall 2021

Miniplaces Challenge: Part 1

Posted: Wednesday, Mar 31, 2021 **Due:** Wednesday, Apr 7, 2021

Note: 6.869 and 6.819 students are expected to finish all problems unless there is an additional instruction.

We provide a python notebook with the code to be completed. You can run it locally or in Colab (upload it to Google Drive and select 'open in colab') to avoid setting up your own environment. Once you finish, run the cells and download the notebook to be submitted.

Submission Instructions: Please submit a .zip file named <your kerberos>.zip containing 1) report named report.pdf including your answers to all required questions with images and/or plots showing your results, and 2) the python notebook provided, with the cells run and the relevant source code. If you include other source code files for a given exercise, please indicate it in the report.

Late Submission Policy: If your pset is submitted within 7 days (rounding up) of the original deadline, you will receive partial credit. Such submissions will be penalized by a multiplicative coefficient that linearly decreases from 1 to 0.5.

The Places dataset[1] is an image dataset of different place categories, like creek or waterfall. Miniplaces is a subset of the Places dataset that we will work with. Next week, the 'challenge' part will be a competition to try to get as high as possible an accuracy on place classification in the Miniplaces dataset. This week, we will start just with an investigation of place recognition using neural networks.

In problems 1 - 3, you will be experimenting with neural networks using PyTorch. In order to run the experiments fast, you will need access to a GPU. We recommend using Colab to run the experiments, since it comes with GPU support. To enable it simply do:

Runtime - Change Runtime type - Hardware accelerator - Gpu.

For those questions include the generated images and relevant code in this report.

Problem 1 *Filter Visualization* (2 pts)

Convolutional kernels in different layers are used to extract information from the input image in different levels. In this problem, we will focus on visualizing the filters of the ResNet network

[1]`http://places2.csail.mit.edu/`

pretrained on Places365 Dataset. You need to download the model and the pretarined weights first.

(a) Normalize the input kernel for better visualization in the *visualize_filters* function.

(b) Visualize filters for another convolutional layer in ResNet.

Problem 2 *Internal Activation Visualization* (3 pts)

In this problem, we will visualize the activations of the internal units in the model. We will chop the fully connected layers of ResNet and visualize the activations of the last convolutional layer into different locations of the input image.

(a) Create a version of the ResNet model without the last two layers to expose the last convolutional layers in *generate_featuremap_unit* function.

(b) As you can see, the unit 300 activates the mountain part of the image. Find other units that detect 1) sky and 2) building in the image.

(c) **(6.869 required; optional for 6.819)** Each unit contributes differently to the final prediction. The contribution weight for each unit is determined by the weights of the fully connected layer (last layer in ResNet). If we deactivate top-5 units (force the kernel to be zeros) that have the maximum weights for the top 1 category of some input image, the prediction for that category will drop dramatically. The code to find the index of those units is given, try to deactivate those units and compare the difference between the original and new top 5 predictions. Show the feature map of that unit.

(d) **(6.869 required; optional for 6.819)** In (c), we deactivate top-5 units and observe dramatic changes in predictions. Try to deactivate less units and report the least amount of units you need to deactivate for changing the predictions.

Problem 3 *Class Activation Map (CAM)* (5 pts)

So far, we know the output of the last convolutional layer activate different parts of the input image. In this problem, we will explore how to visualize which parts of the image are responsible for the final decision. Use the chopped ResNet from problem 2 as the model.

(a) Run the chopped ResNet and you will get a tensor with the size of $(1 \times 2048 \times 8 \times 8)$. 1 is the batch size (b), 2048 is the number of channels (c) and 8×8 is the height (h) and width (w) of the kernel. Convert the output tensor to a new tensor with the size of $(hw \times c)$.

(b) Feed the new tensor from (a) to the fully connected layer of ResNet to compute the weighted average. You will get a output tensor with the size of $(hw \times 365)$, where 365 is the number of classes in the Places365 Dataset.

(c) show the feature maps (combined with input image) of the top 5 predicted categories and explain the relationship between the feature activation and the corresponding category.

(d) *(Bonus, 0.5 points)* Try running this visualization method on another image of a scene. Is the visualization still effective for your image? Explain why or why not.