DEPENDABLE AND SECURE AI-ML (AI60006)

Session 2023-2024

Assignment 1

Submission Deadline: (2 weeks from the date of upload)

You need to be alert to (usually minor) changes that may be made to the assignment statement or to the guidelines after the assignment is first put up. Refresh this frame and re-read the assignment carefully before you make your final submission.

Aim

Understanding anatomy of CNN

CIFAR-10 demo (https://cs.stanford.edu/people/karpathy/convnetjs/demo/cifar10.html)

In this assignment, you'll learn to visualize the internal structures of Convolutional Neural Networks (CNNs) for image data analysis. Specifically, we'll explore the visualization of two-dimensional filters and activation maps, aiding in the understanding of feature detection. The tutorial will guide you through developing visualizations for specific filters and feature maps in a CNN. Additionally, you'll learn a systematic approach to visualize feature maps for each block in a deep CNN, providing insights into the learned representations at different layers. These techniques are invaluable for interpreting and debugging CNN models, enhancing your understanding of the underlying learning processes.

Upon finishing this assignment, you will have acquired the following skills:

1. Proficiency in creating visualizations for specific filters within a Convolutional Neural Network (CNN).
2. Capability to generate visualizations for specific feature maps in a CNN, enhancing your understanding of detected features.
3. A systematic approach to visualizing feature maps for every block in a deep Convolutional Neural Network, allowing for a comprehensive analysis of learned representations at different hierarchical levels.

Reference papers:

Visualizing and Understanding Convolutional Networks (https://arxiv.org/pdf/1311.2901.pdf)

Visualizing and Comparing Convolutional Neural Networks ( <https://ar5iv.labs.arxiv.org/html/1412.6631>)

Interpretable Convolutional Neural Networks (https://openaccess.thecvf.com/content\_cvpr\_2018/papers/Zhang\_Interpretable\_Convolutional\_Neural\_CVPR\_2018\_paper.pdf )

Notebook links

<https://drive.google.com/file/d/1jpSUEZYBD1h2yQrvAJCOao6k9s7vWKDl/view?usp=drive_link>

<https://drive.google.com/file/d/1iW9XJPl5B2mLmXWPHNfe65EXO5JMXfXd/view?usp=drive_link>

Exercises:

1) Applying various filters to the input image:

Gaussian Filter

Laplacian Filter

Gaussian Blur

Edge Detection

Sobel Filter

Canny Edge Detector

2) Generate random 2D vector of size any one of the above filters and train your filter to learn the chosen functionality using gradient descent algorithm. Plot loss curve.

3) Applying filters of varying sizes 5\*5, 7\*7, 9\*9, 11\*11 on input image.

4) Use stride on filters, depending on the filter size and padding.

5) Applying Max pooling, Min pooling, and Average pooling to the applied filter.

6) Visualize the filters of pre trained CNN models( vgg16 or resnet18) and apply them to a new image.

7) Visualize the filters of a pretrained CNN model( vgg19, resnet) applying on new image. Count the number of learned filters layer wise ( example layers 1, 2,... n). Plot a graph among the number of learned filters and layer numbers. Find the number of duplicate filters after visualizing each filter in each model.

6) Prepare a table by applying multiple images to the same model and writing your observations in filter weights (weights are changing in all layers or specific layers). Repeat this experiment for multiple models and multiple input images (see the sample table shown below).

Document

a) The number of filters getting applied to a particular input.

b) For a single MNIST image with a single filter.

c) For a single MNIST image, where each layer affects the filter weights, how can the number of layer-wise learned filter weight changes be calculated? Consider both weight decay and weight gain. Provide a systematic approach to quantifying these changes, taking into account the specific multipliers applied at each layer.

d) For batch processing MNIST images through a model, where each layer influences the filter weights, how can the count of changed filters (either decay or increase in weight) be determined? Additionally, provide a side-by-side comparison of all the filters layer-wise in a document, highlighting the modifications resulting from the applied multipliers.

TPU architecture

<https://cloud.google.com/tpu/docs/system-architecture-tpu-vm>

e) For batch processing MNIST images using a model and considering the Tensor Processing Unit (TPU) architecture, how does the TPU architecture influence computation changes and convolution operations at each layer? Please provide a detailed examination of the altered filters side by side for each layer in a document, taking into account the impact of TPU architecture on the learning process.

|  | Model name | Total number of layers | Number of filters | Present layer number |
| --- | --- | --- | --- | --- |
| input1 |  |  |  |  |
| input2 |  |  |  |  |

Marking guidelines

Submit notebooks and the answers to the questions above in a separate report

(word document). Name the report “ROLLNO\_1.doc. Name the archive “

ROLLNO\_1.zip” .

Assignment marking is to be done only after the deadline expires, as submission gets blocked after the assignment is marked.

You should keep submitting your incomplete assignment from time to time after making some progress, as you can submit any number of times before the deadline expires.

Warning: Cases of copying from internet sources or fellow students will be dealt with seriously and severely, with recommendation to the Dean to de-register the student from the course.