Arthur Liou
NetID: arthurl3
CS445

Final Project Report

Topic: Image Segmentation

Motivation

- From Project Proposal Submission
- "Motivation - I wanted to do something with image classification & segmentation. I hope to learn how use to various Python libraries and tools to understand fundamentally how image classification and segmentation is performed."
    - Addendum: For my final project, I absolutely wanted to do a machine learning oriented project relevant to computational photography, so image classification or segmentation came first to mind. I've never really dove into either topic since it was never needed in my work experience and or in my school work prior to this class.

Approach

- Used Colaboratory as a substitute for Jupyter notebooks, for GPU capabilities
- I started by walking through the tutorials from the notebooks above and the Pytorch official docs & also read online educational resources regarding image classification and segmentation. As I am new to Pytorch and implementing neural networks, this was a good intro experience to the PyTorch library and various functions, while also learning about, building, and training neural networks.
    - Colaboratory Intro What is Colaboratory?
    - Deep Learning for Image Classification - Tutorial
    - PyTorch / Dataloader Tutorial Training a Classifier
    - CNN Tutorial: Neural Networks. Wrote and trained a neural network in pytorch, trained it, classified the CIFAR10 dataset
- With the knowledge from the tutorials, I completed Image Classification. More details in the Implementation Details section below.

Results

- Find my results and analyses inline in the details in the next section.

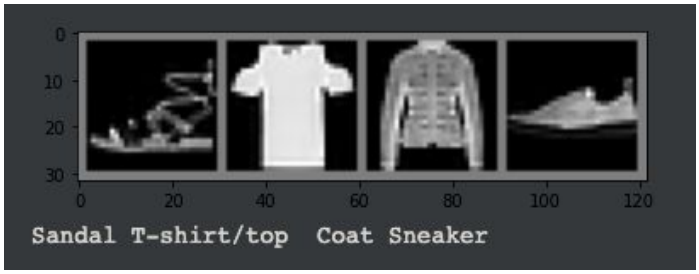
Challenges / innovation

- Plotting Training Loss
- Rebuilding the training model and adapting it for a new data set, training model and test model. (see Loss Function and Optimizer below)
- Catalogue and display images from each class that are correct and incorrect
- Visualize learned filters

Implementation Details

Image Classification

- Hitting my Data Load and CNN milestones from my proposal
    - a) using FashionMINST instead of CIFAR10, and
    - b) re-writing the CNN for the channel requirements.

Dataloader Random Set #1 	Dataloader Random Set #2 
Dataloader Random Set #3 	Dataloader Random Set #4 

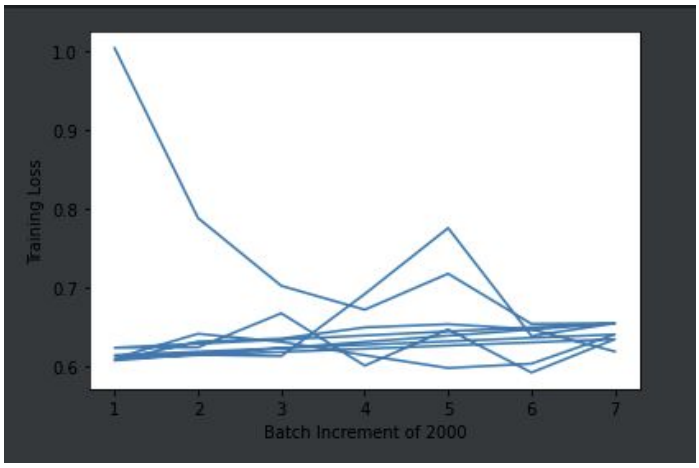
Loss Function and Optimizer:

- I started running into issues / challenges here. I found that I had to go back and adjust the hyperparameters of my CCN to process the images from the FashionMINST dataset correctly so the optimizer can be trained properly.
- After validating I could train my CNN once, I went about the challenge of implementing a training loop until convergence and plotting each loss step
- After overcoming that (mental) milestone, I played around with the various hyperparameters, learning rates (0.001, 0.005, 0.05, 0.02, etc), optimizer functions (ie the original SVG vs current Adam), and my convergence threshold (0.1, 0.05, 0.01, etc). Each adjustment would take some time to complete. Not Project 5 length of run time, but enough that I'd take 5-15 to work on my writeup, get coffee, or go for a short walk while waiting for training to complete.
    - Fun Challenge: Once I went too small of a threshold, went to make lunch, and came back to find my CNN was still running and had completed over a dozen iterations. In the logging, I would see the interactions and training loss increase, so I added a hard stop after 10 iterations after that.
    - Reflection: Given the amount of time and training, I should have marked "training the model until convergence" as its own milestone in my original proposal.
- Setup my classifier to save and show 2 correct & incorrectly classified image sets.
- Learned how to show the CNN's learned convolutional filters

CNN Hyperparamaters

```
Net(
  (conv1): Conv2d(1, 32, kernel_size=(4, 4), stride=(1, 1))
  (conv2): Conv2d(32, 32, kernel_size=(4, 4), stride=(1, 1))
  (pool): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  (pool2): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  (fc1): Linear(in_features=512, out_features=120, bias=True)
  (fc2): Linear(in_features=120, out_features=84, bias=True)
)
```

Plot Training Loss (for each epoch)



Accuracy of CNN on 10k images

```
Accuracy of the network on the 10000 test images: 76 %
```

76% accuracy

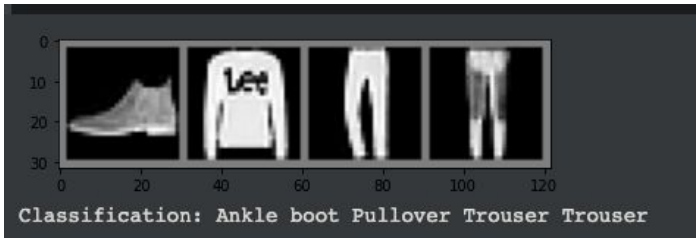
Output: Epochs + Training Loss

```
Device: cpu
[1,  2000] loss: 1.003
[1,  4000] loss: 0.788
[1,  6000] loss: 0.702
[1,  8000] loss: 0.672
[1, 10000] loss: 0.717
[1, 12000] loss: 0.654
[1, 14000] loss: 0.655
[2,  2000] loss: 0.624
[2,  4000] loss: 0.624
[2,  6000] loss: 0.667
[2,  8000] loss: 0.601
[2, 10000] loss: 0.647
[2, 12000] loss: 0.592
[2, 14000] loss: 0.634
[3,  2000] loss: 0.610
[3,  4000] loss: 0.641
[3,  6000] loss: 0.631
[3,  8000] loss: 0.614
[3, 10000] loss: 0.598
[3, 12000] loss: 0.603
[3, 14000] loss: 0.641
[4,  2000] loss: 0.614
[4,  4000] loss: 0.615
[4,  6000] loss: 0.613
[4,  8000] loss: 0.692
[4, 10000] loss: 0.775
[4, 12000] loss: 0.639
[4, 14000] loss: 0.655
[5,  2000] loss: 0.608
[5,  4000] loss: 0.631
[5,  6000] loss: 0.635
[5,  8000] loss: 0.649
[5, 10000] loss: 0.653
[5, 12000] loss: 0.647
[5, 14000] loss: 0.619
Finished Training
```

Per Class Accuracy of CNN

```
Right/Wrong Classification Started
Right/Wrong Classification Started
Accuracy of T-shirt/top : 74 %
Accuracy of Trouser : 90 %
Accuracy of Pullover : 60 %
Accuracy of Dress : 87 %
Accuracy of  Coat : 74 %
Accuracy of Sandal : 93 %
Accuracy of Shirt : 14 %
Accuracy of Sneaker : 81 %
Accuracy of   Bag : 96 %
Accuracy of Ankle boot : 96 %
```

Visualization of A Sample Set of Results and Images



- Results + Significance
  - This CNN Classifier training run came back with 75% accuracy.
  - Shirt and Pullover classes were hardest to classify
  - Trouser, Sandal, Bag, and Ankle boot were the most accurate.
  - During the CNN's training, I found that usually at least 4-6 epochs trained before convergence.
  - Included a learned filter from the CNN that is used to classify image classes

Expected Points Outline

The core assignment is worth 100 points, as per rubric:

- Paper completeness and quality (30 points)
    - Page Length
        - 2 Short Pages + 1 Page for References and Figures *which are in line
        - This 4th page is for expected points / rubric.
    - Motivation and potential impact of topic is clear (5 points)
        - In proposal and copied above
    - Approach is described as clearly as space allows (15 points)
        - Yes at a high level, more granular details in Implementation Details
    - Results and their significance is clearly presented (5 points)
        - Yes, placed inline in the Implementation Details section
    - Implementation details are clearly described (5 points)
        - Yes, expanded on what I did at each step during my approach in a linear matter, shared some challenges and different paths I took. At the very end, I shared or summarized my train loss plot, accuracy images, my image classification results and images to hit my completion / evaluation milestone.
            - "Plot the train and validation accuracy during the training process, visualization of the results and images."
- Implementation (70 points)
    - Completeness (50 points): Based on the submitted paper and code, the implementation addresses the proposed objectives. If implementing an existing paper or known technique, full credit requires that the method is fully implemented with expected results demonstrated
        - Completed - Image Classification. Image Segmentation was B&W
    - Innovation / Challenge (20 points)
        - Project implements a basic technique with low risk, such that most students could design and implement within a few hours (0/20 points)
        - Project implements a moderately complex paper or technique that would be expected to take 10-15 hours to complete. The risk for failure is low, but moderate effort is required. (10 / 20 points)
        - Project proposes and successfully executes a new application or new technique, **expands on an existing technique,** performs experimentation that provides insights not present in the initial paper, or implements a difficult paper, such that the student overcomes potential setbacks or risk of failure to achieve a successful outcome. (20 / 20 points)
        - Assessment: Moderately Complex Technique + expands on this to generate additional insights not present in the tutorial.Up to 20 points

Expected Points: (5+15+5+5 + a missing 5) + (50+20) = 100

*I could not fit all my images/results and annotations in under two pages, so similarly to the projects, I exported my notebook as a PDF to show all outputs generated.