Homework 6
601.482/682 Deep Learning
Fall 2023

Oct 18, 2023

Due 11:59pm on Nov 8, 2023

Please submit 1) a single zip file containing your Jupyter Notebook and PDF of your Jupyter Notebook to "Homework 6 - Notebook" and 2) your written report (LaTeX generated PDF) to "Homework 6 - Report" on Gradescope (Entry Code: BBVDNN)

Important: You mus Ve highly recommend using Google C

Important: If you do Python scripts to Colaboratory and en ttings). Training should converge in le ates in that time, you should re-examin gnment early.

1. *Unsupervised P* copic Instrument Challenge.[1] Y oic frame images (*not* in sequen h RGB frame as an input and p instrument type and backgroun ning method and compare the pe out pre-training. This is relevant shortage of data labels.

   Data Folder egmentation' and '/colorization'. aining purposes.

   The **main goa** umerated on the next page.

   - *Complete the Network structure for segmentation task (a-c).* The network structure we provide is a simplified U-Net, which is a very popular framework in medical image segmentation task. Read and understand the code in `unet.py`, the implementation is missing the last layer and the last activation function. Next, fill in the missing components for 1(a). For the segmentation task, train ONLY with the frames in '/segmentation/train' and validate and test with the frames in '/segmentation/validation' and '/segmentation/test' respectively. The original input image is a $256 \times 320 \times 3$ RGB image. The ground truth label is a grey-scale image that has the same dimension, where different gray values indicate the instrument type or background tissue.
   - *Pre-training by self-supervised colorization (d).* Image colorization training[2] is a common way of self-supervised learning.[3] The idea is to take grey-scale images as an input and predict colorized images, similar to filling in colors in a draft painting. You must

[1] `https://endovissub2017-roboticinstrumentsegmentation.grand-challenge.org`

[2] Larsson, G., Maire, M., & Shakhnarovich, G. (2017). Colorization as a proxy task for visual understanding. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 6874-6883).

[3] Jing, L., & Tian, Y. (2020). Self-supervised visual feature learning with deep neural networks: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence.

use the **same** U-Net structure as above to train a model for this colorization task. Then use the pretrained weights as initialization for the segmentation task.

For the colorization task, you need to train on '/colorization/train_cor' and validate on '/colorization/validation_cor'. You are given `mapping.json` that contains the label of each grey level. We have also provided grey-scale image for each input in each subfolder of '/colorization' for your colorization task input.

Now that you know the broad goals and data sets, please complete the following TODOs.

(a) Train a segmentation network using the frames in the '/segmentation/train' folder. Please complete the DICE score function to evaluate your model, and write from scratch a DICE loss function as your network loss[4]. (*Hint: You need to convert the grey-scale label mask to one-hot encoding of the label and then calculate the DICE score for each label*). Please train the network until convergence (should take around 30 min) using the default provided hyperparameters and provide a figure of training loss and validation loss w.r.t. epochs (in a single figure). Please report your performance (DICE score) on the test dataset, you should expect a DICE score > 0.5. (*Hint: Using BatchNorm might help you achieve better performance.*)

(b) Introduce meaningful data augmentation (e.g. vertical and horizontal flips) and train the netwo (a). Please plot the traini ort test dataset performan

(c) Train on rain_cor' folder. Use hyper xperiments) and mean squa aining loss w.r.t. epochs un the network for the next t

(d) Load the gmentation task using the u are using the same hyp ly state them in your repo ort test dataset performan *Hint: Since this is a relativel in performance.*)

2. *Transfer Learn* sed in HW4 and download the `vision/models.html`).

(a) Randomly odel to learn the Fashion M e? Please report your test $> 85\%$.

(b) Load the pre-trained VGG16 model from torch vision models. Freeze all but the last layer: randomly initialize the last layer of your network and fine-tune this. What accuracy do you get now? Please again report your test accuracy on the test dataset. You should expect an accuracy $> 60\%$.

(c) Now, imagine a scenario in which you want to train the VGG16 model on an entirely new dataset and will fine-tune either the model from (2a) or (2b). Which pre-trained model is the preferred starting point for your new use case?

[4]Read more about the DICE score: `https://medium.com/datadriveninvestor/deep-learning-in-medical-imaging-3c1008431aaf`

[5]The full FashionMNIST dataset can be downloaded from the official website here: `https://github.com/zalandoresearch/fashion-mnist`