Math 6380p: Adv. Top. Deep Learning **31 Oct, 2018**

Project 2. Image Classification and Deep Learning.

Instructor: Yuan Yao *Due: 00:00am Wednesday 14 Nov, 2018*

1 Mini-Project Requirement and Datasets

This project as a warm-up aims to explore feature extractions using existing networks, such as pre-trained deep neural networks and scattering nets, in image classifications with traditional machine learning methods.

1. Pick up ONE (or more if you like) favourite dataset below to work. If you would like to work on a different problem outside the candidates we proposed, please email course instructor about your proposal.
2. Team work: we encourage you to form small team, up to FOUR persons per group, to work on the same problem. Each team just submit ONE report, *with a clear remark on each person's contribution*. The report can be in the format of either Python (Jupyter) Notebooks with a detailed documentation (preferred format), a *technical report within 8 pages*, e.g. NIPS conference style

   `https://nips.cc/Conferences/2016/PaperInformation/StyleFiles`

   or of a *poster*, e.g.

   `https://github.com/yuany-pku/2017_math6380/blob/master/project1/DongLoXia_poster.pptx`
3. In the report, show your proposed scientific questions to explore and main results with a careful analysis supporting the results toward answering your problems. Remember: scientific analysis and reasoning are more important than merely the performance tables. Separate source codes may be submitted through email as a zip file, GitHub link, or as an appendix if it is not large.
4. Submit your report by email or paper version no later than the deadline, to the following address (deeplearning.math@gmail.com) with Title: Math 6380P: Project 1.

2 Challenge

The following proposes three candidates and you are welcome to propose your own research project. Previous challenges are collected in the end and you may pursue a deeper exploration.

2.1 Nexperia Kaggle in-class Contest

Nexperia (`https://www.nexperia.com/`) is one of the biggest Semi-conductor company in the world. They will produce billions of semi-conductors every year. But unfortunately, they are facing a hard problem now which is the yield rate of the semi-conductors. However they have lots of data and hope that the yield rate could be greatly improved by the hot deep learning technics now. So with the data they provide to us, we lunch this in-class Kaggle contest which tries to use various machine learning and deep learning methods to solve this real world problem.
Because this is the first Nexperia image classification contest, we set only 2 classes, one for bad semi-conductor and another for good. The aim of this simplified contest is to predict the type of each semi-conductor based on the image. And we provide 30K and 3000 images for training and testing respectively.

Checking the following kaggle website for more details.

- Kaggle: `https://www.kaggle.com/c/semi-conductor-image-classification-1`

To participate the contest, you need to login your Kaggle account first, then open the following invitation link and accept the Kaggle contest rule to download the data:

`https://www.kaggle.com/t/7002fff75f2c422cb34068731971afcd`

In the future, we may consider to lunch another more complex in-class Kaggle contest. But this project is not limited to classification, you can do various explorations such as visualization 2 and abnormal outlier detection.

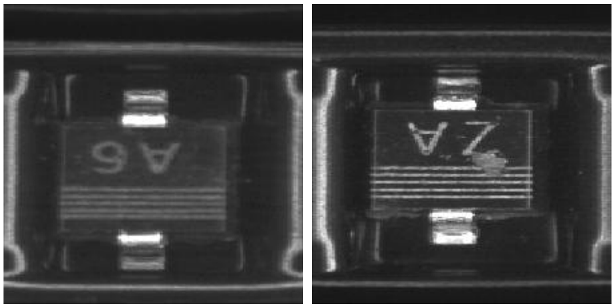

Figure 1: Sample Semi-conductor Image. Left: good Right: bad

2.2 DCF-Net Exploration

This challenge is to implement the DCF-Net in image classification tasks, defined by Xiuyuan Cheng et al. in the following paper

Qiang Qiu, Xiuyuan Cheng, Robert Calderbank, Guillermo Sapiro, *DCFNet: Deep Neural Network with Decomposed Convolutional Filters*, ICML 2018. `arXiv:1802.04145.`

Currently there are two implementations,

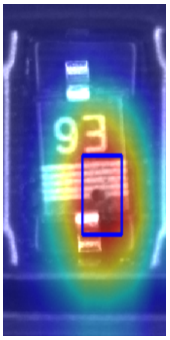

Figure 2: Sample Semi-conductor Visualization.

- Matlab: `https://github.com/xycheng/DCFNet`
- Pytorch: `https://github.com/ZeWang95/DCFNet-Pytorch`

You may train a DCF-Net, e.g. DCF-Net-VGG16 or DCF-Net-ResNet18, on your favorite datasets (e.g. MNIST/Fashion-MNIST/Cifar10), and compare them against pretrained VGG16 and ResNet18 etc. For example, Table 4 of the DCF-Net paper, shows comparison between imagenet-vgg-verydeep-16 and DCFNet based VGG 16. You may either reproduce such an experiment and/or explore new models with new datasets.

2.3 Reproducible Training of CNNs

The following best award paper in ICLR 2017,

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals, Understanding deep learning requires rethinking generalization. `https://arxiv.org/abs/1611.03530`

received lots of attention recently. Reproducibility is indispensable for good research. Can you reproduce some of their key experiments by yourself? The following are for examples.

1. Achieve ZERO training error in standard and randomized experiments. As shown in Figure 3, you need to train some CNNs (e.g. ResNet, over-parametric) with Cifar10 dataset, where the labels are true or randomly permuted, and the pixels are original or random (shuffled, noise, etc.), toward zero training error (misclassification error) as epochs grow. During the training, you might turn on and off various regularization methods to see the effects. If you use loss functions such as cross-entropy or hinge, you may also plots the training loss with respect to the epochs.

2. Non-overfitting of test error and overfitting of test loss when model complexity grows. Train several CNNs (ResNet) of different number of parameters, stop your SGD at certain large enough epochs (e.g. 1000) or zero *training error (misclassification)* is reached. Then compare the *test (validation) error* or *test loss* as model complexity grows to see if you observe similar phenomenon in Figure 4: when *training error* becomes zero, *test error* (misclassification) does not overfit but *test loss* (e.g. cross-entropy, exponential) shows overfitting as model complexity grows. This is for reproducing experiments in the following paper:

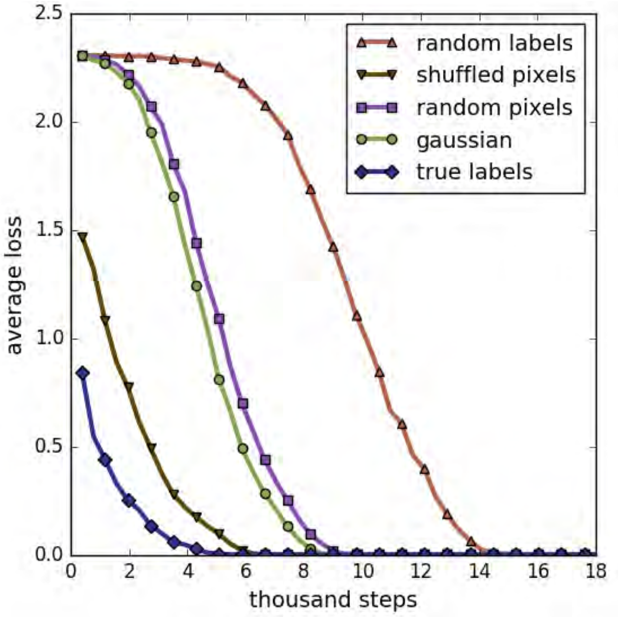

Figure 3: Overparametric models achieve zero *training error* (or near zero *training loss*) as SGD epochs grow, in standard and randomized experiments.

Tomaso Poggio, K. Kawaguchi, Q. Liao, B. Miranda, L. Rosasco, X. Biox, J. Hidary, and H. Mhaskar. Theory of Deep Learning III: the non-overfitting puzzle. Jan 30, 2018. `http://cbmm.mit.edu/publications/theory-deep-learning-iii-explaining-non-overfitting-puzzle`

3. Can you give an analysis on what might be the reasons for the phenomena you observed?

3 Old Challenge

The basic challenge is

- Feature extraction by scattering net with known invariants;
- Feature extraction by pre-trained deep neural networks, e.g. VGG19, and resnet18, etc.;
- Visualize these features using classical unsupervised learning methods, e.g. PCA/MDS, Manifold Learning, t-SNE, etc.;
- Image classifications using traditional supervised learning methods based on the features extracted, e.g. LDA, logistic regression, SVM, random forests, etc.;
- *Train the last layer or fine-tune the deep neural networks in your choice;
- Compare the results you obtained and give your own analysis on explaining the phenomena.

Below are two candidate datasets. Challenge marked by * above is only optional.

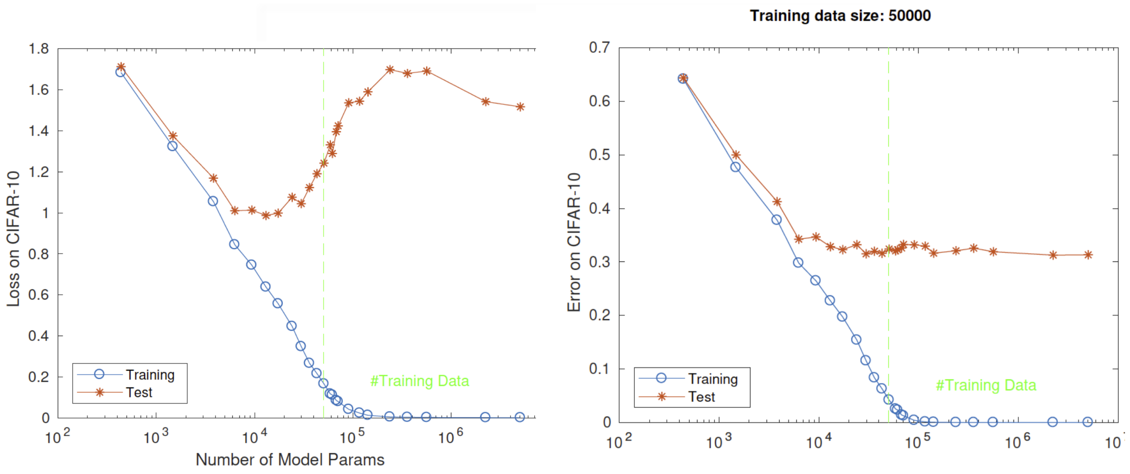

Figure 4: When *training error* becomes zero, *test error* (misclassification) does not increase (resistance to overfitting) but *test loss* (cross-entropy/hinge) increases showing overfitting as model complexity grows.

3.1 MNIST dataset

Yann LeCun's website contains original MNIST dataset of 60,000 training images and 10,000 test images.

`http://yann.lecun.com/exdb/mnist/`

There are various ways to download and parse MNIST files. For example, Python users may refer to the following website:

`https://github.com/datapythonista/mnist`

or MXNET tutorial on mnist

`https://mxnet.incubator.apache.org/tutorials/python/mnist.html`

3.2 Fashion-MNIST dataset

Zalando's Fashion-MNIST dataset of 60,000 training images and 10,000 test images, of size 28-by-28 in grayscale.

`https://github.com/zalandoresearch/fashion-mnist`

As a reference, here is Jason Wu, Peng Xu, and Nayeon Lee's exploration on the dataset in project 1:

`https://deeplearning-math.github.io/slides/Project1_WuXuLee.pdf`

3.3 Cifar10 dataset

The Cifar10 dataset consists of 60,000 color images of size 32x32x3 in 10 classes, with 6000 images per class. It can be found at

`https://www.cs.toronto.edu/~kriz/cifar.html`

3.4 Identification of Raphael's paintings from the forgeries

The following data, provided by Prof. Yang WANG from HKUST,

`https://drive.google.com/folderview?id=0B-yDtwSjhaSCZ2FqN3AxQ3NJNTA&usp=sharing`

contains a 28 digital paintings of Raphael or forgeries. Note that there are both jpeg and tiff files, so be careful with the bit depth in digitization. The following file

`https://docs.google.com/document/d/1tMaaSIrYwNFZZ2cEJdx1DfFscIfERd5Dp2U7K1ekjTI/edit`

contains the labels of such paintings, which are

1 Maybe Raphael - Disputed

2 Raphael

3 Raphael

4 Raphael

5 Raphael

6 Raphael

7 Maybe Raphael - Disputed

8 Raphael

9 Raphael

10 Maybe Raphael - Disputed

11 Not Raphael

12 Not Raphael

13 Not Raphael

14 Not Raphael

15 Not Raphael

16 Not Raphael

17 Not Raphael

18 Not Raphael

19 Not Raphael

20 My Drawing (Raphael?)

21 Raphael

22 Raphael

23 Maybe Raphael - Disputed

24 Raphael

25 Maybe Raphael - Disputed

26 Maybe Raphael - Disputed

27 Raphael

28 Raphael

There are some pictures whose names are ended with alphabet like A's, which are irrelevant for the project.

The challenge of Raphael dataset is: can you exploit the known Raphael vs. Not Raphael data to predict the identity of those 6 disputed paintings (maybe Raphael)? Textures in these drawings may disclose the behaviour movements of artist in his work. One preliminary study in this project can be: *take all the known Raphael and Non-Raphael drawings and use leave-one-out test to predict the identity of the left out image; you may break the images into many small patches and use the known identity as its class.*

The following student poster reports are good explorations

1) Hanlin GU, Yifei HUANG, and Jiaze SUN: `https://github.com/yuany-pku/2017_CSIC5011/blob/master/project3/05.GuHuangSun_poster.pdf`

2) Jianhui ZHANG, Hongming ZHANG, Weizhi ZHU, and Min FAN: `https://deeplearning-math.github.io/slides/Project1_ZhangZhangZhuFan.pdf`,

3) Wei HU, Yuqi ZHAO, Rougang YE, and Ruijian HAN: `https://deeplearning-math.github.io/slides/Project1_HuZhaoYeHan.pdf`.

The following papers by Haixia Liu et al. study art authentication using geometric tight frames and scattering transform, respectively, which might be useful reference for you:

`http://dx.doi.org/10.1016/j.acha.2015.11.005`

`https://www.sciencedirect.com/science/article/pii/S0165168418301105`