Deep Learning: Towards Deeper Understanding **29 Mar, 2018**

Project 2: Midterm

Instructor: Yuan Yao *Due: 12 Apr, 23:59, 2018*

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 6380O: Project 2.

1 Great Challenges of Reproducible Training of CNNs

The following best award paper in ICLR 2017,

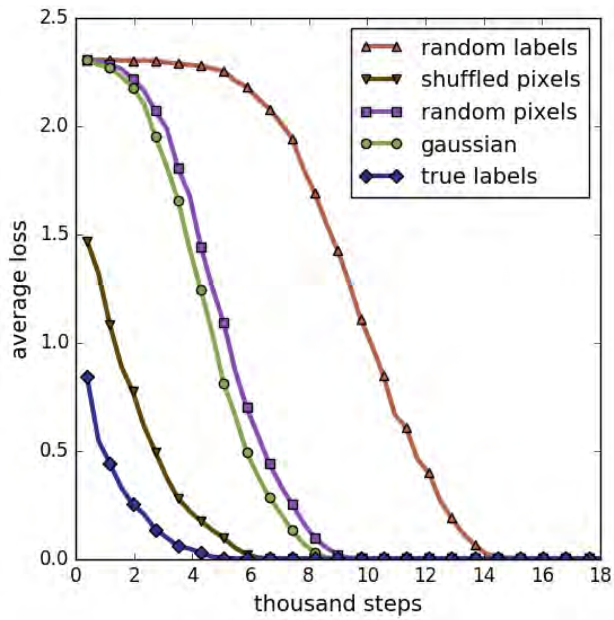

Figure 1: Overparametric models achieve zero *training error* (or near zero *training loss*) as SGD epochs grow, in standard and randomized experiments.

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 1, 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 2: 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:

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?

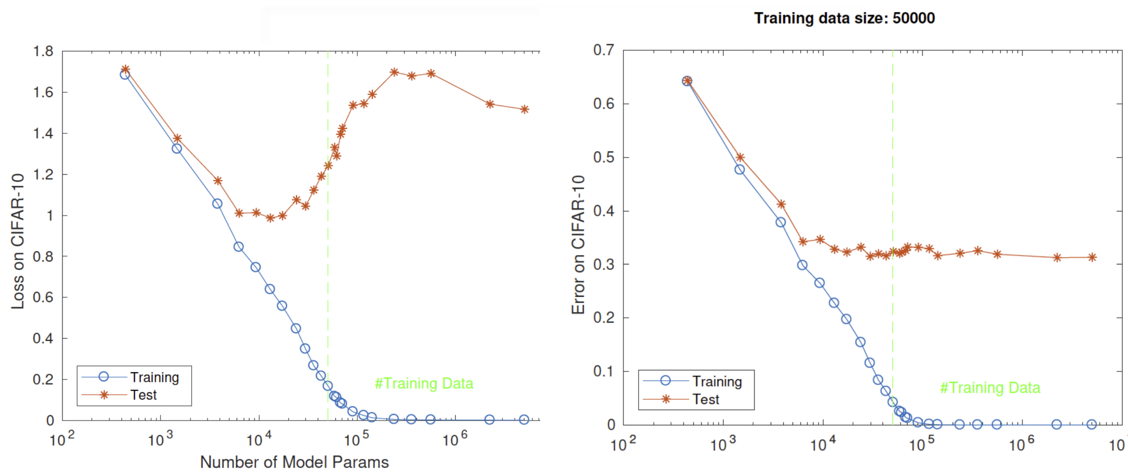

Figure 2: When *training error* becomes zero, *test error* (misclassification) does not increase (resistance to overfitting) but *test loss* (cross-entropy/hinge) increases showing overfitting as model complexity grows.

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`

Attention: training CNNs with such a dataset is time-consuming, so GPU is usually adopted. If you would like an easier dataset without GPUs, perhaps use MNIST or Fashion-MNIST (introduced below).

1.1 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`

2 Kaggle contest in-class: Predictive Maintenance

2.1 Background

Predictive maintenance techniques are designed to help anticipate equipment failures to allow for advance scheduling of corrective maintenance, thereby preventing unexpected equipment downtime, improving service quality for customers, and also reducing the additional cost caused by over-maintenance in preventative maintenance policies. Many types of equipment – e.g., automated teller machines (ATMs), information technology equipment, medical devices, etc. – track run-time status by generating system messages, error events, and log files, which can be used to predict impending failures.

Thanks to Nexperia company for providing the dataset, we launched the Kaggle competition in-class at the following website

`https://www.kaggle.com/c/predictive-maintenance1`

To participate the contest, you need the following Invitation Link:

`https://www.kaggle.com/t/212723063992429cbb66ded8c43f923f`

2.2 Data

The data consists of log message and failure record of 984 days from one machine.

- log message: five basic daily statics below of some 'minor' errors of 26 types occurred during machine running. Each 'minor' error has an ID. These errors are not fatal but may be good predictor of machine failure in next day. So there are $p = 5 \times 26 = 130$ features per day.
  - count: how many times the error occurs in that day.
  - min: tick of the first time the error occurs in that day (seconds).
  - max: tick of the last time the error occurs in that day.
  - mean: mean of tick the error occurs.
  - std: standard deviation of tick.
- failure record: binary variable.
  - 0 : machine is OK in that day.
  - 1 : machine break down in that day.

The test data is constructed from last $n_{test} = 300$ days of log messages by withholding the labels. The training set is the remaining records of $n_{train} = 684$ days.

2.3 Goal

This project aims to predict machine failure in advance. There are several tasks for you to try:

- 1-day in-advance prediction: you may use daily log message as inputs (features), to predict *next day*'s machine failure (1 for break-down and 0 for OK);

- multiple-days in-advance prediction: explore the prediction of a day's failure using historic record in previous days.

For more detail, you may refer to the Kaggle website pages. Make sure **DO NOT** use any information on the same day or after the day been predicted.

3 Image Captioning by Combined CNN/RNN/LSTM

In this project, you're required to implement a RNN to do image captioning. Your work may include the following parts, but not limited to,

- Implement a CNN structure to do feature selection. You may do this by transfer learning, like using Inception, ResNet, etc.
- Implement a (e.g. single hidden layer) fully connected network to do word embedding.
- Implement a RNN structure to do image caption. You may use select one of network structure, like LSTM, BiLSTM, LSTM with Attention, etc.
- Train your network and tune the parameters. Select the best model on validation set.
- Show the caption ability of your model visually. Evaluate your model by BLEU (bilingual evaluation understudy) score on test set.

3.1 Dataset: Flickr8K

You could download Filckr8K dataset, which includes 8,000 images and 5 captions for each, via the following links. `https://forms.illinois.edu/sec/1713398`

The Flickr8K dataset is provided by flicker, an image- and video-hosting website. It's a relatively small dataset in image captioning community. Perhaps it's still too big for CPU computations. If you don't have access to GPU resources, try using dimension reduction on image features and using pre-trained word embedding to help you work this project on your own CPU.

4 Continued Challenges from Project 1

In project 1, the basic challenges are

- 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.

You may continue to improve your previous work. Below are some candidate datasets.

4.1 MNIST dataset – a Warmup

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`

4.2 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 report seems a good exploration

https://github.com/yuany-pku/2017_CSIC5011/blob/master/project3/05.GuHuangSun_poster.pdf

The following paper by Haixia Liu, Raymond Chan, and me studies Van Gogh's paintings which might be a reference for you:

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

In project 1, some explorations can be found here for your reference:

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

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

Moreover, the following report by Shun ZHANG from Fudan University presents a comparison with Neural Style features:

3) https://www.dropbox.com/s/ccver43xxvo14is/ZHANG.Shun_essay.pdf?dl=0.