Deep Learning: Towards Deeper Understanding **28 Nov, 2018**

Final Project. Advanced Topics in Deep Learning

Instructor: Yuan Yao *Due: 23:59 Sunday 15 Dec, 2018*

1 Requirement

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 must submit:

   ONE report, *with a clear remark on each person's contribution.* The report can be in the format of either a *technical report within 8 pages*, e.g. NIPS conference style (preferred format)

   `https://nips.cc/Conferences/2019/PaperInformation/StyleFiles`

   Python (Jupyter) Notebooks with a detailed documentation, or a *poster*, e.g.

   `https://github.com/yuany-pku/2017_math6380/blob/master/project1/DongLoXia_poster.pptx`

   (b) *ONE short presentation video within 10 mins*, e.g. in Youtube or Bilibili link. You may submit your presentation slides together with the video link to help understanding.

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.

2 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 anomaly detection of the semi-conductors. A small number of devices may have defects, e.g. Fig. 1 shows a good example and a bad example. Nexperia accumulated lots of image data using online digital camera and hoped that human based anomaly detection could be greatly improved by the state-of-the-art deep learning technique. So with the data they provide to us, we launch this in-class Kaggle contest in a purpose to test various machine learning methods, esp. deep learning, to solve this real world problem.
Because this is the first Nexperia image classification contest, we start from binary classification with only two classes of labels, 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. 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-first`

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/8cf20e3116d54a749766f9904a9e330a`

In the future, we may consider to launch another in-class Kaggle contest, while the current project is just for a warm-up. Beyond binary classification, you can do various explorations such as visualization in Fig. 1 (right picture) and unsupervised abnormal outlier detection.

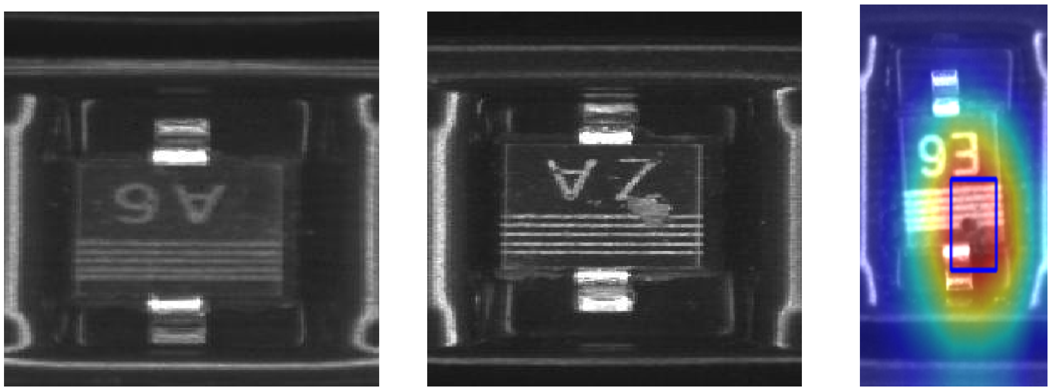

Figure 1: Examples of Nexperia image data. Left: a good example; Middle: a bad example; Right: a visualization based on heatmap

3 China Equity Index Prediction Contest

3.1 Overview

Asset Management is an area where traditional statistics are holding ground firmly and not losing to deep learning. Unlike the previous field such as playing go where too many possibilities and the computing power was the limitation; Asset management, on the contrary, has too little data and therefore easily gets over-fitting. Another argument will be that there are too many players, too many noises and no fundamental theory supporting that there actually is a robust relationship between any features and future asset returns. However, we believe in higher frequency field of asset management, where we have more data and also more micro-structure related features which are more robust, recent development in machine learning like deep neural networks might have its chance to win over traditional statistics. For this reason and also considering easier to start with not too large data set, equity index high frequency data is our choice for this contest. Later on, a much larger data set, equity data, which is thousands of times index data could be offered, if interested.

3.2 Data

The in-sample data we provided is 9 years of equity index snap shot data. Trading time of each day is 4 hours and the frequency of the snap shot is 0.5s. The goal is to predict index return for the next 10 minutes, which is labeled y_{10m} in the csv files provided. We also provided around 80 features we found that have forecasting power, together with our linear prediction labeled yhat1, yhat2 and yhat3. We kept some data as out-of-sample test.

You could download in-sample data here:
SFTP
IP: 120.132.124.224
Port: 22
Username: testuser
Password: testuser

data.csv: All-in-one file with features, y and our linear predictions yhat.
data.tar.gz: Compressed data.csv
IF.csv: Average of our linear prediction
y.csv: Index return to predict
evaluation.py: A script to evaluate your prediction. See Evaluation for details.

3.3 Evaluation

3.3.1 Standard

Assume your forecast is ybar:

- Maximize $corr(ybar, y)$: note we calculate this correlation every 20 days, and then take average.
- $corr(\text{ybar}, \text{our existed yhat}) < 0.3$: hopefully the new ybar could discover something linear model missed.

3.3.2 Script

Run `evaluation.py` after some configurations to automatically check the performance of your predictions. For usage, please check out docstring of `evaluation.py` for details.

4 Reproducible Study of Training and Generalization Performance

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 scientific 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 2, 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 3: 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. There seems a double descent phenomemon in bias-variance trade-off: in traditional statistical learning, bias reduces and variance increases when model complexity grows, that leads to a bell shape of generalization (test) error; but in overparameterized models, generalization error seems always dropping as model complexity grows. For example, Fig. 4 is copied from the following paper:

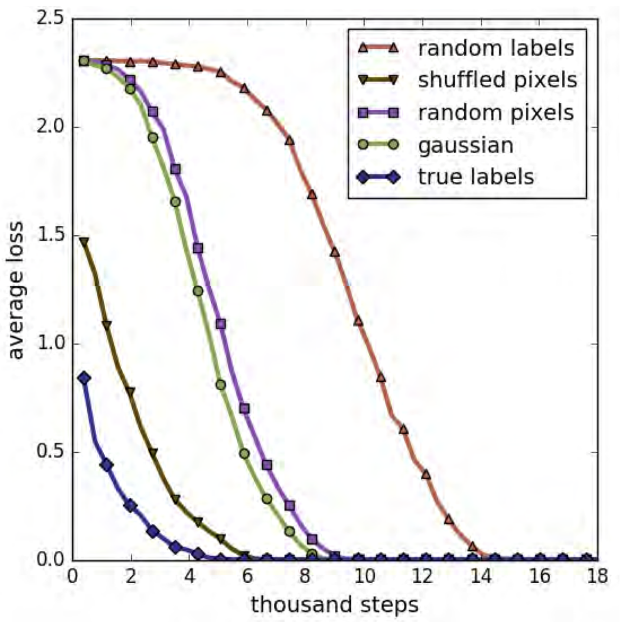

Figure 2: Overparametric models achieve zero *training error* (or near zero *training loss*) as SGD epochs grow, in standard and randomized experiments.

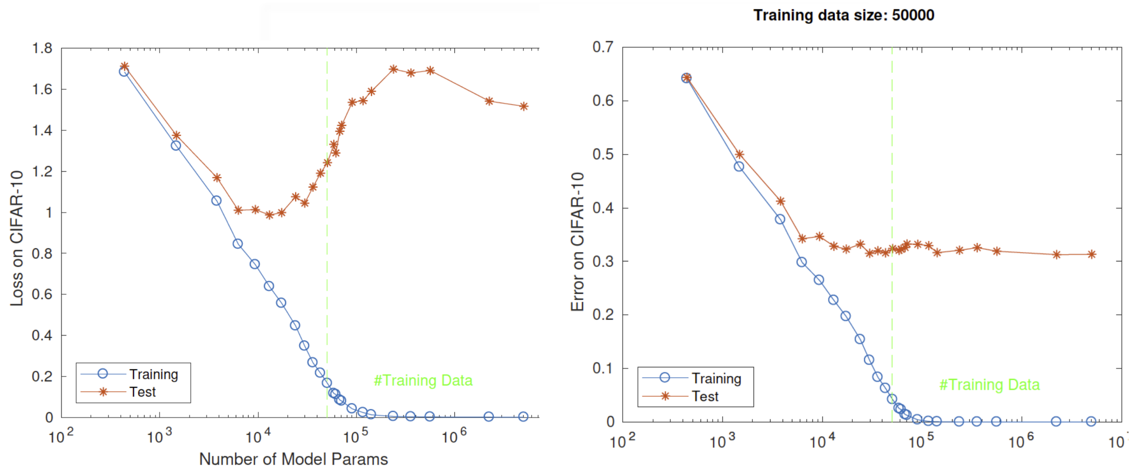

Figure 3: When *training error* becomes zero, *test error* (misclassification) does not increase (resistance to overfitting) but *test loss* (cross-entropy/hinge) increases showing overfitting as model complexity grows.

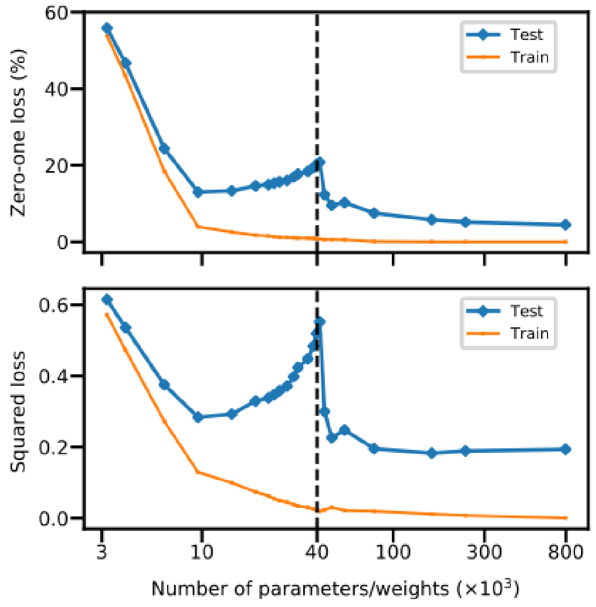

Fig. 3. Double-descent risk curve for a fully connected neural network on MNIST. Shown are training and test risks of a network with a single layer of H hidden units, learned on a subset of MNIST ($n = 4 \cdot 10^3$, $d = 784$, $K = 10$ classes). The number of parameters is $(d+1) \cdot H + (H+1) \cdot K$. The interpolation threshold (black dashed line) is observed at $n \cdot K$.

Figure 4: Double descent curves.

Mikhail Belkin, Daniel Hsu, Siyuan Ma, Soumik Mandal. Reconciling modern machine learning practice and the bias-variance trade-off. PNAS , 2019, 116 (32).

Can you reproduce it and extend your study with Cifar10 dataset?

4. Many works on generalization bounds learned in class (e.g. Peter Bartlett, Sasha Rakhlin, etc.) try to explain the mystery of good generalization performance of overparameterized models based on

$$\text{test error} \leq \text{training error} + \frac{\text{complexity}}{\sqrt{n}}$$

where n is the sample size and "complexity" does not depend on the number of parameters of a network. Recently, Nagarajan and Kolter argued (`https://arxiv.org/abs/1902.04742`) that all these bounds are insufficient to explain the phenomenon. Can you give your own analysis or perspective?

5 Challenge from Project 1

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.

5.1 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`

5.2 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`

5.3 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`

5.4 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 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`