UCSD COGS 181 Fall 2017

Search this site

Instructor: Zhuowen Tu
Syllabus
Calendar and Class Notes
Reading List
Homework Assignments
Contact Us
Useful Links

Homework Assignments >

Final Project

posted Nov 2, 2017, 10:34 PM by Zhuowen Tu **[updated Dec 10, 2017, 11:47 AM]**

Due date: 11:59Pm, 12/16

Late Policy: 5% of the total points will be deducted for the first day past due and every 10% of the total points will be deducted for every extra day afterwards.

Report format: Write a report with >1,500 words including main sections: a) abstract, b) introduction, c) method, d) experiment, e) conclusion, and f) references. You can follow the paper format as e.g leading machine learning journals such as Journal of Machine Learning Research (http://www.jmlr.org/) or IEEE Trans. on Pattern Analysis and Machine Intelligence (http://www.computer.org/web/tpami), or leading conferences like NIPS (https://papers.nips.cc/) and ICML (http://icml.cc/2016/?page_id=151). Please submit your report and attach your code as supplementary materials to ted (you can alternatively provide a link to your github page in your report too).

Bonus points: If you feel that your work deserves bonus points due to reasons such as: a) novel ideas and applications, b) large efforts in your own data collection/preparation, c) state-of-the-art results on your applications, or d) new algorithms or neural network architecture, **please create a "Bonus Points" section to specifically describe why you deserve bonus points.** In general, we evaluation your justifications based on the review guidelines based on e.g. CVPR/NIPS/ICCV/ICLR.

In addition, there will be optional presentation for the final project to receive bonus points. You can either submit a 3-5 minutes of short video clip to ted as supplementary materials to your report, or to have a physical presence for you presentation on Friday 12/15, 11:30a-2:29pm in CSB 001 (sing up here).

Note that requirement for the word count (>1,500) only applies to a single-student project. For team-based projects, each team only needs to write one final report but the role of each team member needs to be clearly defined ans specified. The final project report is also supposed to be much longer thatn 1,500 words, depending upon how many (maximum 3) members there are in your team.

Word count:
One-person team: >1,500
Two-persons team: > 2,200
Three-persons team: > 2,900

See below a link about writing a scientific paper: http://abacus.bates.edu/~ganderso/biology/resources/writing/HTWtoc.html
The format of your references can be of any kind that is adopted in the above journals or conferences.

Grading: The merit and grading of your project can be judged from aspects described below that are common when reviewing a paper:
1. Interestingness of the problem you are studying. (10 points).

2. How challenging and large is the dataset you are studying? (10 points)

3. Any aspects that are new in terms of algorithm development, uniqueness of the data, or new applications? (20 points)

Note that we encourage you to think something beyond just downloading existing code and train on a standard benchmarks. Basically, you are expected to complete a report that is worth reading (even not publishable to some extent). If you have done a good job of relative thorough investigation of e.g. different architectures and different parameters, it is considered to be somewhat "new". The experiences you have are worth reading for the others who have not tried them before. When someone is reading your report, he/she will feel like something worthy is there including your own attempts for algorithms, a non-standard dataset or application, general conclusion about your parameter tuning, what the neural network structure might be a better choice etc.

In a nutshell, this definition of "new" is somewhat different from the aspect of "being novel" when reviewing a paper that is submitted to e.g. NIPS. Will add this part to the project description.

4. Is your experimental design comprehensive? Have you done thoroughly experiments in tuning hyper parameters? (30 points)

Tuning hyper-parameters in your final project will need to be more comprehensive that what was done in HW4.
For example, if you are performing CNN classification on the Tiny ImageNet dataset, some options to consider include
a. Comparing two different architectures chosen from e.g. LeNet, AlexNet, VGG, GoogleNet or ResNet
b. Trying to vary the number of layers.
c. Trying to adopt different optimization methods, for example Adam vs. stocahstic gradient descent
d. Trying different pooling functions, average pooling, max pooling, stochastic pooling
e. Trying to use different activation functions such as ReLu, Sigmoid etc.

See e.g. how the significance was justified in the ResNet pape (you don't have to follow this paper though): https://arxiv.org/pdf/1512.03385.pdf

5. Is your report written in a professional way with sections including abstract, introduction, method/architecture description, experiments (data and problem description, hyper-parameters, training process etc.), conclusion, and references? (30 points)

6. Bonus points will be assigned to projects that have adopted new methods, worked on novel applications, and/or have done a thorough comparison against the existing methods and possible choices.

There will be three options for the final project (if you have your own idea, please come to talk to me):

Option (1): Convolutional neural networks Train a convolutional neural networks method on Tiny ImageNet dataset (http://pages.ucsd.edu/~ztu/courses/tiny-imagenet-200.zip) You can choose any deep learning platforms including MatConvNet (http://www.vlfeat.org/matconvnet/), Theano (http://deeplearning.net/software/theano/), Torch (http://torch.ch/), Caffe (http://caffe.berkeleyvision.org/), MxNet (https://github.com/dmlc/mxnet), and TensorFlow. See in the link other possible platforms you can use: http://deeplearning.net/software_links/ You can train a model by building your own network structure or by adopting/following standard networks like AlexNet, GoogLeNet, VGG, etc. You are also welcome to use datasets other than ImageNet, e.g. CIFAR-10, and Kaggle datasets (https://www.kaggle.com/) such as the deep sea image competition: https://www.kaggle.com/c/datasciencebowl

Option (2): (Individual only, no team wok) Deep belief networks from G. Hinton (http://www.cs.toronto.edu/~hinton/csc2515/projects/deep1.html) His course webpage: http://www.cs.toronto.edu/~hinton/csc2515/lectures.html Details about the deep belief networks (DBN) can be found in their Science paper: Hinton, G. E., Osindero, S., Welling, M. and Teh, Y. Unsupervised Discovery of Non-linear Structure using Contrastive Backpropagation. http://onlinelibrary.wiley.com/doi/10.1207/s15516709cog0000_76/epdf Links to some other packages that you might consider using: http://deeplearning.net/software_links/ Try to use the MNIST dataset.

Code are available at https://github.com/albertbup/deep-belief-network or https://github.com/blackecho/Deep-Learning-TensorFlow (You may also use any other DBN implementation) MNIST dataset is available at http://www.cs.toronto.edu/~hinton/MatlabForSciencePaper.html Task: Try to use varying number of training samples vs. testing samples to see the behavior of the DBN algorithm. Also apply SVM (decision tree, boosting, random forests etc.) on the same setting and compare the performances of DBN with SVM. Wen using DBN, you will need to determine how many hidden layers the DBN should have, how many units each layer has, and how long the pre-training should be. Due to the big overhead in training deep learning, you can try to reduce your training data by trying e.g. 1,000 and 3,000 samples. Other variations you can consider by fixing training size to e.g. 1,000 samples: a. Number of hidden layers: 1, 2, and 3 b. Number of units in the first layer: 100, 500 c. Number of units in the second layer: 200, 500 d. Learning rate: 0.05, 0.1, 0.5, 1.0.

Option (3): (Individual only, no team wok) Char RNN. You can read more about “The Unreasonable Effectiveness of Recurrent Neural Networks” at the link http://karpathy.github.io/2015/05/21/rnn-effectiveness/

Code are available at https://github.com/karpathy/char-rnn or https://github.com/jcjohnson/torch-rnn or https://github.com/sherjilozair/char-rnn-tensorflow (You may also use any other char-rnn implementation). Tiny shakespeare dataset is available at https://github.com/karpathy/char-rnn/blob/master/data/tinyshakespeare/input.txt and complete Sherlock Holmes is available at https://sherlock-holm.es/stories/plain-text/cnus.txt You can also try out other interesting applications as described in the post, such as Wikipedia, Algebraic Geometry (Latex) or Linux Source Code, but you need to collect the dataset by yourself. You need to work on at least one dataset/application and try to produce meaningful results by using the char rnn model.

Option (4): (please discuss it with the instructor when planning). A topic of your own about visual, acoustic, language and other data modeling using modern deep learning techniques including convolutional neural networks, recurrent neural networks, auto-encoder etc. You can look for interesting topics on recent NIPS, CVPR, ICLR, ICCV, ACL, AAAI, etc.

Some interesting project you can consider working on:

Language modeling:
https://github.com/andrewt3000/DL4NLP/blob/master/README.md

Recurrent Neural Networks:
char-rnn (http://karpathy.github.io/2015/05/21/rnn-effectiveness/)

Face recognition:
OpenFace (https://cmusatyalab.github.io/openface/)
Neural Artistic Style Transferring:
https://github.com/andersbll/neural_artistic_style

Audio:
Google's recent release of a massive audio dataset: https://research.google.com/audioset/

Image Super-resolution:
https://github.com/david-gpu/srez
https://github.com/alexjc/neural-enhance

Image Captioning:
https://github.com/karpathy/neuraltalk2
Show-and-Tell (https://github.com/tensorflow/models/tree/master/im2txt)

Object detection:
Faster-RCNN (https://github.com/rbgirshick/py-faster-rcnn)
YOLO (https://pjreddie.com/darknet/yolo/)

Project reports from the Standford cn231n class:
2016: http://cs231n.stanford.edu/reports2016
2015: http://cs231n.stanford.edu/reports.html

Comments

You do not have permission to add comments.