Course: DD2422 - Final Project

Ideas & tips for your project

It might be daunting, given the amount of material on the web, to try and narrow down the scope and subject matter of your project. In this document I will help you achieve this by suggesting some interesting projects that you could complete and where labelled datasets exist.

1 General guidelines

What recognition, synthesis, manipulation or translation (or some other) task interests you? Is there a network architecture or training algorithm that you would love to investigate in further detail? Is there a way you can explore this interest in a meaningful yet computational feasible way? If you think the answer is yes, then you may have found the topic for your project. You should study the literature and the plethora of tutorials and blog posts on the web regarding your project idea. Find out the most common and successful deep learning solution to your project topic. If it's an architecture or training algorithm you're investigating then find out what types of problems this network is used to solve and what datasets exist that you can use for your experiments.

Remember you do not need to do novel work. What you need to do is define some relevant questions on the different aspects of the deep learning approach you chose. Conduct meaningful experiments regarding those questions and make and discuss the conclusions that can be made from these experiments. The set of questions can include trying the various techniques taught during the lectures. But you not limited by the content of the lectures.

2 Computer vision

2.1 Idea 1: Image classification

One of the classic tasks of computer vision is image classification. That is, given an image, predict whether it contains an object from a set of predefined classes. In fact, image classification was the first big success story in the resurgence of convolutional networks. You can train a deep ConvNet to perform image classification using ImageNet. ImageNet [**?**] contains 1000 classes and contains $\tilde{1}.3$ million images. To make things computational feasible given your resources, you will use a 200 classes subset of those images

which are also down-sampled from the original size (`tiny_imagenet.zip`). These are the sets used by *Stanford's cs231 course* and are kindly shared by the instructors of that course. That means, you can compare the results of your methods with those of Stanford cs231 students. Of course, you are free to use the original ImageNet training and test set for your experiments if you have access to computational resources (modern GPUs).

2.2 Idea 2: Prediction of Attributes

The Multi-Task Facial Landmark (MTFL) dataset contains images of faces and are annotated with **1**) 5 facial landmarks **2**) attributes of gender (male/female), smiling (yes/no), wearing glasses (yes/no) and head pose (left profile/left/frontal/right/right profile). You could train a deep ConvNet to predict one of these attributes or all of them. One option would be to take a pre-trained deep ConvNet (VGG, ResNet, GoogLeNet,...), replace its output layer with the outputs you wish to predict and then fine-tune the network to adapt to this particular problem.

2.3 Idea 3: More experiments with CIFAR-10

During the assignments you played around with the `CIFAR-10` dataset. You could investigate now ,using *GPU powered software* different issues related to training and pushing classification performance of convolutional networks (and fully connected networks).

Push performance What is the best classification performance you can achieve using either a fully-connected network and/or a convolutional neural network using the modern tricks (related in the lectures and those in your reading) been invented/rediscovered in the past 5 years? Does augmenting your training data, using batch normalization, etc. help performance and by how much? Can you replicate the top performing results? This webpage Classification datasets results has a list of the top performing networks on CIFAR-10.

Investigate the importance of network depth The course has deep learning in its title. Maybe you want to convince yourself experimentally that a network's depth is actually a key component in its final accuracy. Here you could design a set of experiments to try and highlight the importance of the number of layers in the final network and its final performance. Here you would need to think about creating networks with the same number of parameters but with different depths and how they are distributed throughout the network. You could also investigate the use of the composition of small spatial convolutional filters versus large spatial filters.

2.4 Your own specification

Are you enthusiastic and have enough time to spend on the project to gain extra experience? You are absolutely free to choose any task you are excited about and a corresponding dataset. You can downsample the dataset to make it fit the computational resources available to you. Be aware that training a "non-classification" network can be sometimes more challenging. We will take that into account when evaluating the quality of the project though. For a set of datasets related to the different visual recognition tasks please refer to the extra resources section below.

2.5 Publicly available computer vision datasets

Here are some classic visual recognition and classification tasks with a corresponding dataset:

- Face recognition/identification/verification: To uncover/match the identity of a face. Dataset: LFW
- Scene classification: To classify a given image into different scene labels. Dataset: Places
- Semantic segmentation: To classify every pixel of an image into a class. Dataset: MS COCO
- Human detection: To detect the location of all the humans in an image (if any). Datasets: Caltech, Daimler
- Pose estimation: To estimate the location of the joints of a given person. Dataset: MPII
- Video classification: Classify a video into a set of pre-defined classes. Dataset: Youtube Sports
- Depth estimation: Estimate the depth at each pixel from the RGB data. Dataset: NYU
- Gaze estimation: Estimate the eyes gaze of a person based on his image. Dataset: MPII Gaze

The webpages CV Datasets on the web and Yet Another Computer Vision Index To Datasets (YACVID) have a more comprehensive list of available datatsets.

3 Natural language processing

NLP is a domain where variations of RNNs come into their own. You could potentially use some variation of RNNs to perform some form of text classification, generation and/or translation. The excellent site spro/practical-pytorch has several links to several tutorials, such as Practical PyTorch: Translation with a Sequence to Sequence Network and Attention or Practical PyTorch: Classifying Names with a Character-Level RNN exercise suggestions. These tutorials and exercise suggestions could definitely form the basis for interesting projects.

4 Speech/Sound

Unfortunately, my hands-on experience in speech recognition and processing is very limited. But I'm very open to finding out more by reading your projects in the area. Here are two Speech Recognition datasets that may be of practical help:

- CMU Robust Speech Recognition Group: Census Database
- LibriSpeech ASR corpus.

Or here is a link to code, a dataset and paper description of how to use a LSTM to *compose* folk music Folk music style modelling using LSTMs. It is unclear to me how long training would take in this case.

5 Popular software packages for deep learning

These frameworks are currently the most popular among academic researchers. (Most take advantage of Nvidia's deep learing libraries so their computational timings do not differ that much.)

- MatConvNet: by Andrea Vedaldi et al. from Oxford University based on `MATLAB`.
- Caffe: created by Yangqing Jia and maintained by BVLC at Berkeley and open-source community. In `C++` and `Cuda` with limited python and MATLAB wrappers.
- Torch7: Maintained collaboratively by people at NYU, Facebook, and Google DeepMind. Written in `LuaJIT`

- PyTorch: Maintained collaboratively by people at facebook, NYU, ParisTech, Nvidia, ... Written in `python`.
- TensorFlow: The open-source Google deep learning framework. It has both a `C++` and `python` API.
- Theano: Mainly developed by the academics at Universite de Montreal. Has a `python` API.
- Deeplearning4j: by SkyMind a San Francisco-based business intelligence and enterprise software firm. Written in `Java` and `Scala`.

If I've missed a popular package, please let me know and I can add it. You are free to choose whichever package you like and feel most comfortable with.

6 Top-tier conferences for inspiration

Here are some top-tier conferences where you can find relevant papers from the fields of computer vision and deep learning:

- Main focus **Computer vision**
  i) ICCV ii) ECCV iii) CVPR iv) BMVC
- Main focus: **Spoken Language Processing**
  i) InterSpeech ii) EuroSpeech
- Main focus: **Natural Language Processing**
  i) InterSpeech ii) EMNLP
- Main focus: **Learning Representations**
  i) ICLR
- Main focus: **Machine Learning**
  i) NIPS ii) ICML