Project introduction

Project name: **Toolbox for managing the training neural networks**

This document is based on the combination of:

- The original project introduction PDF by Pyry Takala
- Material Pyry gave us via email
- Notes we have made during Skype discussions

Introduction

Deep learning is one of the newest trends in machine learning. Instead of specifying features that a machine should learn, neural networks can learn these features from data. Recent breakthroughs of deep learning include state-of-the-art image classification algorithms link1, computers playing Atari-games above human-level link2 and flexible tools for analyzing natural language link3. Training of neural networks is often challenging, with many practical difficulties:

- how do we know what is happening inside a neural network?
- Where is the network making errors?
- How is the training of the network proceeding?
- How do we manage a queue of different experiments?
- Why is training the network too slow?

Project goals

A good toolbox for deep learning would let a researcher easily specify experiments, manage a queue of experiments, and automatically monitor networks during training. During train-time, a diagnostics toolbox can perform various analyses on the training log and network parameters to detect possible problems early on. Notifications can be sent to a researcher early on so that expensive computing time is not wasted on an experiment that is unlikely to give good results. Naturally, the tool should not create a huge computational overhead, and usability should be good. Majority of the work involves creating a tool that helps a user define network inputs, manage an experiment queue, and visualize intermediary and final values of the network. The final product will be a tool for training neural networks that should be agnostic of the deep learning framework (e.g. frameworks *Torch* or *Theano* could both be used), and can benefit any neural networks researchers. Requirement gathering could be done from various deep learning users at the university, and potentially students could talk to some researchers abroad as well.

Tools and technology

The toolbox could be written e.g. as an interactive web tool (e.g. with JavaScript) that could be always run on a server that manages the experiments.

Requirements for the students

The toolbox could be implemented for instance as a hybrid of JavaScript and Python. Some other language can be also considered if it appears more suitable for the task. Students participating in this project will need to be willing to read a little bit about neural networks, but a detailed view is not required. The interface of the tool should be created in English.

Legal Issues

Potentially, the resulting code could be released under an open-source license after the project. Signing the non-disclosure agreement (NDA) included in the Aalto's contract template is not required.

Client

Aalto University's Deep learning and Bayesian modeling group conducts research in the field of neural networks. Recent projects include for instance *DeepBeat* a neural network that generates rap link4, an image-processing framework network *Ladder* that learns to recognize images from very small training datasets link5, and an ongoing project of financial predictions. The toolbox would ideally be used by all researchers at Aalto and also researchers at other universities and companies. The student will have a good opportunity to get to know the field of machine learning and deep learning during the project.

Client representative

- Organization: Aalto University
- Role: Researcher
- Name: Pyry Takala
- E-mail: pyry.takala@aalto.fi

Notes about deep learning

- Its a reborn topic.

- Less than 10 researchers do neural networks research at Aalto. Namely Pyry Takala, Antti Rasmus, Mathias Berglund, Jelena Lukatina.
- Some top companies like Google, Microsoft, Siri, Amazon (Echo) have started to research into the field. Notable are also Montreal LISA lab, Google DeepMind.
- The project could be useful to many deep learning labs and people in other fields running time consuming experiments (e.g. physics simulations).

Materials

- Specify experiments
- rnn_experiments.xlsx

Manage queue

- rnn_experiments.xlsx
- experiment1.slurm
- http://deeplearning.net/software/jobman/intro.html

Monitor experiments

- notification (e.g. email) if X
- saving & loading requirements

Analyze (visualise) running experiments

- [error analysis][http://www.doc.ic.ac.uk/~sgc/teaching/pre2012/v231/errorplot.gif]
- time analysis
- weight norm analysis (e.g. per layer)
- Analyze (visualise) ready experiments
- [http://karpathy.github.io/2015/05/21/rnn-effectiveness/]
- error analysis per input feature

Training log examples

- log.txt
- [Torch][http://torch5.sourceforge.net/manual/newbieTutorial.html]
- [Lasagne][http://lasagne.readthedocs.org/en/latest/user/tutorial.html]
- [Pylearn2][http://daemonmaker.blogspot.ca/2014/12/monitoring-experiments-in-pylearn2.html]

Muutamia lisäresursseja

- Neuroverkkoblogi jossa aika paljon demoja ja visualisointeja: [Colah][http://colah.github.io/]
- Kattava [lista resursseja][https://github.com/ChristosChristofidis/awesome-deep-learning]

Muita vaatimuksia

- Tärkeää myös voida seurata tietokoneiden resursseja, mm. gpun muisti, ram, disk-space. Ideaalisti tietää jo ennen ajoa että esim. diskspace riittää
- Profilointi (python, theano, etc.)