The course: Deep Learning on Hardware

Course Professor: Prof. Avi Mandelzon
Course Assistant: Prof. Alex Bronstein
Tutor/Practical: Chaim Baskin

Course Name: "Deep Learning on Hardware"

Course Language: English

Prerequisites:

- Logical Design,
- Mathematics, Probability.

Course Syllabus in Hebrew:

Algorithms like Deep Learning have been used today in many different fields, from low-energy systems with low computation costs to "hyper" systems that base on powerful computers.

In order to allow rapid development of applications in this field, several new software environments have emerged, which allow efficient execution of the algorithms. In this course, we will focus on the relationship between the efficiency of the algorithms, their quality, and hardware/software solutions.

We will learn software methods that allow the use of hardware accelerators, such as GPGPU, and expand the discussion to the latest research in the field.

Study topics:

- Background on DNN algorithms such as CNN, RNN, and others with examples of use.
- Efficient use of software packages such as PyTorch.
- Advanced algorithms such as Deep Reinforcement Learning, Variational Autoencoders, and others.
- CUDA – NVIDIA accelerators.
- Programming Massively Parallel Processors.
- Foundations of parallel programming – CUDA.
- API applications of computer vision and natural language processing using deep learning.
- PyTorch - http://pytorch.org/
- Wen-mei W. Hwu, “Programming Massively Parallel Processors”, Morgan Kaufmann

Graded topics (1-5):

1. Understanding the principles of solving problems using DNN algorithms.
2. Understanding how to write programs using CUDA.
3. Understanding how to use PyTorch software for constructing DNNs.
4. Learning different optimization methods that allow efficient implementation of learning problems on advanced systems.
5. Doing technical work (small) in the field.
Deep Learning on Computation Accelerators

The course will be taught in English

**English syllabus:**

Deep learning is widely used in many market segments ranging from mobile devices to supercomputers. Recently different SW packages as well as special HW accelerators were developed to support deep learning. The course will focus on algorithms, programming languages and new SW/HW interfaces that aim to allow execution of deep learning algorithms in a productive and efficient way.

**Learning Outcomes:**

At the end of the course, the student will

1. Understand and be able to apply notions in deep learning
2. Know how to program GPUs using CUDA
3. Know how to effectively use PyTorch SW packages
4. Know how to optimize SW and HW performance in deep neural network applications
5. Perform a small research project using the studied notions and techniques

**Grad:**

30% Drills, 30% final presentation and 40% final project

Detailed syllabys is below:
<table>
<thead>
<tr>
<th>WW</th>
<th>Lecture</th>
<th>Tutorial</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Introduction</td>
<td>Setup environment, administration and Intro</td>
</tr>
<tr>
<td>2</td>
<td>Elements of machine learning: data-driven approach, k-nearest neighbor</td>
<td>Numerical computation in Python: numpy (Tensor computing)</td>
</tr>
<tr>
<td></td>
<td>classifier, linear classification,</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Linear regression</td>
<td>Signal processing in Python: (Tensor computing)</td>
</tr>
<tr>
<td></td>
<td>Loss functions, optimization, descent methods, stochastic gradient</td>
<td></td>
</tr>
<tr>
<td></td>
<td>descent</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Neural networks: multi-layered perceptrons, backpropagation</td>
<td>Intro to PyTorch</td>
</tr>
<tr>
<td>5</td>
<td>Convolutional neural networks</td>
<td>Training convolution neural networks in PyTorch</td>
</tr>
<tr>
<td>6</td>
<td>Training neural networks: art and science</td>
<td>Training convolution neural networks in Python: AlexNet, VGG, GoogLeNet,</td>
</tr>
<tr>
<td></td>
<td>Activation functions, initialization, dropout, batch normalization,</td>
<td>ResNet, etc.</td>
</tr>
<tr>
<td></td>
<td>update rules, data augmentation and domain transfer learning</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Data decomposition vs. functional decomposition</td>
<td>Final project definition</td>
</tr>
<tr>
<td>8</td>
<td>Hardware for DNN training</td>
<td>Introduction to CUDA</td>
</tr>
<tr>
<td>9</td>
<td>Elements of parallel computing</td>
<td>Efficient implementation of matrix operators and convolution on CUDA</td>
</tr>
<tr>
<td>10</td>
<td>GPU as an accelerator</td>
<td>GPU in PyTorch, performance considerations</td>
</tr>
<tr>
<td>11</td>
<td>Recurrent neural networks: RNN, LSTM and applications in NLP, CV</td>
<td>RNNs, LSTM training</td>
</tr>
<tr>
<td>12</td>
<td>Variational Autoencoders (VAEs), Deep Q-Learning</td>
<td>VAEs, Deep Q-Learning</td>
</tr>
<tr>
<td>13</td>
<td>Summary and material completion</td>
<td></td>
</tr>
</tbody>
</table>