

Deep Learning Software Packages

TensorFlow

<https://www.tensorflow.org/>

Ohad Shitrit
BGU

Outline

- Motivation
- Python
- Deep Learning software packages
- TensorFlow - basics
- Example - Mnist
- API
- Parallel computation
- Train your own model
- Playground

Motivation

Why do we need software package for ML / DL ?

- You don't want to spend your time debugging your SGD algo.
- You do want to spend your time on Data Science and Model planning
- Let the experts do their work, you are a researcher
- Open Source - community
- Optimization - GPU
- Complexity

Credit

- Technical information and Examples from TensorFlow website:
<https://www.tensorflow.org>
- White paper : <https://arxiv.org/abs/1603.04467>

Python

Most of the EE's graduates knows Matlab well

- You can do a lot of nice things with Matlab, even Deep Learning
- Matlab is not open source -> python
- Free
- Object Oriented
- Numpy, Scipy, sklearn
- DL software packages developed in Python, Lua, C++.
- Fun
- Ipython

■ Python Example

Deep Learning Packages

C++

Caffe - simple,
fast, ConvNets,
first?

Python

Theano - Graph,
python, long
compilation
Tensorflow -
Graph, python,
google

“Like” Python

Torch - Popular,
easy to create
layers, Lua?

Other

MatConvNet -
Matlab,
optimized, easy
to understand

Wiki

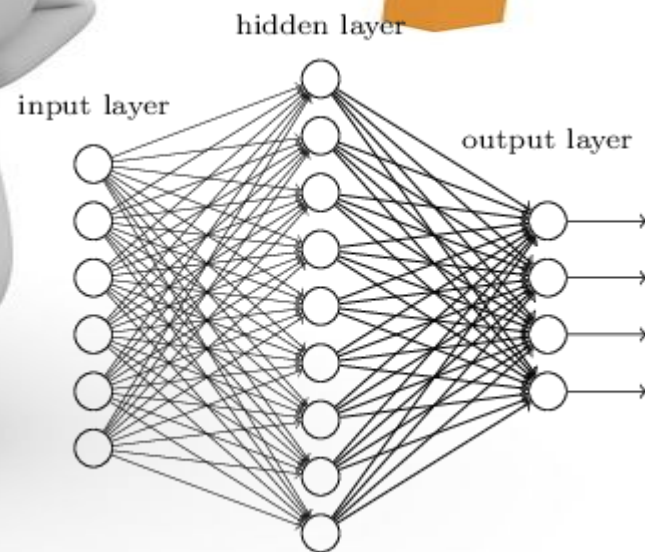
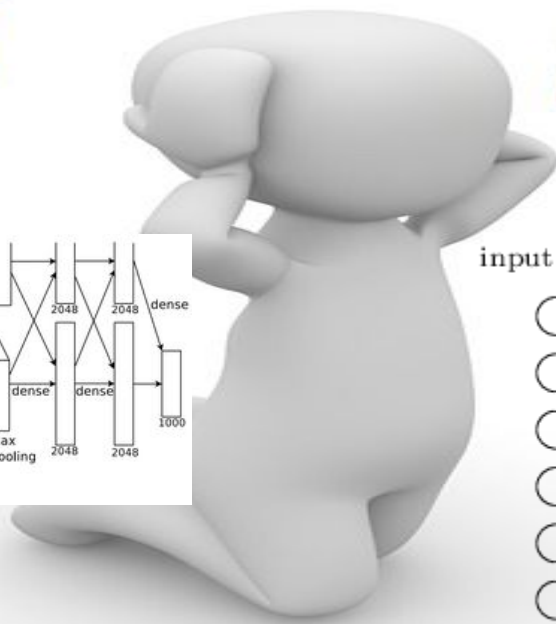
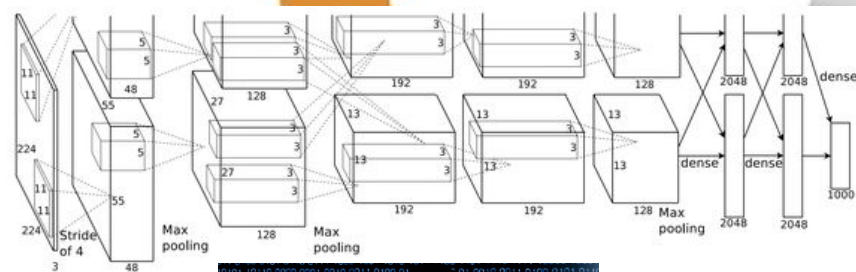
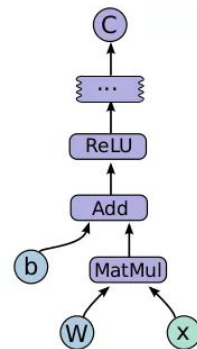
https://en.wikipedia.org/wiki/Comparison_of_deep_learning_software

TensorFlow

TensorFlow

From the website:

TensorFlow™ is an open source software library for numerical computation using data flow graphs. Nodes in the graph represent mathematical operations, while the graph edges represent the multidimensional data arrays (tensors) communicated between them. The flexible architecture allows you to deploy computation to one or more CPUs or GPUs in a desktop, server, or mobile device with a single API. TensorFlow was originally developed by researchers and engineers working on the Google Brain Team within Google's Machine Intelligence research organization for the purposes of conducting machine learning and deep neural networks research, but the system is general enough to be applicable in a wide variety of other domains as well.



TensorFlow - Basics

Tensor - Multidimensional array, shape = [N, H, W, C]

N - number of examples (batch)

H - height

W - width

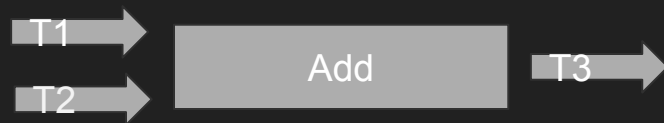
C - Channels (RGB)

```
tf.Variable(tf.zeros([1,10,10,3]))
```

TensorFlow - Basics

Represent computations as graphs

- Node=ops (operations)
- Build a graph - symbolic
- Define inputs (placeholders) and outputs
- Launch the graph in a session
- Session - place graph onto devices (cpu, gpu), provides methods to execute them (SGD)
- Return Tensor/Numpy arrays



■ Graph Example

TensorFlow - Loss

- Define the “Loss” function:

```
loss = tf.reduce_mean(tf.square(y - y_data))  
optimizer = tf.train.GradientDescentOptimizer(0.5)  
train = optimizer.minimize(loss)
```

- It's all symbolic, not need to provides any analytical derivative (for backprop)

TensorFlow - Training

Once the graph is ready, run it with a session:

```
# Before starting, initialize the variables. We will 'run' this first.  
init = tf.initialize_all_variables()
```

```
# Launch the graph.  
sess = tf.Session()  
sess.run(init)
```

```
# Fit the model.  
for step in range(201):  
    sess.run(train)  
    if step % 20 == 0:  
        print(step, sess.run(W), sess.run(b))
```

TensorFlow - Example

- Mnist data-set
- Create function for each stage
- Run from main script
- Save checkpoints
- Feed input using dictionary + next_batch()
- TensorBoard
- Lenet-5 (Conv)
 - **In order to run tensorflow: Unsetenv LD_LIBRARY_PATH**

TensorFlow - API

- Nothing more to say...

https://www.tensorflow.org/versions/r0.10/api_docs/python/index.html

TensorFlow - Parallel computation

- Parallel data source (Replica)
- Model parallelism - conv ?
- Multicore and multithreaded parallelism
- Distributed TensorFlow

https://www.tensorflow.org/versions/r0.10/how_tos/distributed/index.html

TensorFlow - Train your own model

- 90% of the work is **DATA**
- Define the method for feeding your model
 - TensorFlow
 - Other
- Build a model (you can use one of the common models)
 - Use the functionality of TF (Compact code)
- Create an evaluation function (TPR, TNR, Accuracy...)

TensorFlow - Train your own model

- Test your model on a small size of data (Overfit)
- Run hyperparameters “grid search”
- Validate the “checkpoint” saver is working well :-<
- Do sanity with the first checkpoint (Save & Load)
- Train the model and start praying .. :)
- TENSORBOARD :)

TensorFlow - let's play

- Playground -

<http://playground.tensorflow.org/>

<http://cs.stanford.edu/people/karpathy/convnetjs/demo/cifar10.html>

<https://github.com/jtoy/awesome-tensorflow>

Questions?

Good Luck
Ohad Shitrit

