Practical Deep Learning: an introduction to Keras and Tensorflow

Winter School in Quantitative Systems Biology: Learning and Artificial Intelligence

Eis Annavini

THE BUILDING BLOCKS OF DL

So far we have used simple operations as the building boxes for our models

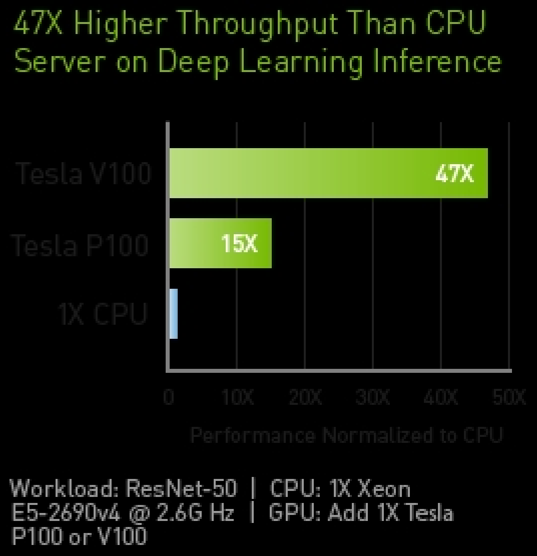
- Matrix multiplication
- Elementwise addition/subtraction
- Convolutions
- Basic mathematical functions

Why should we need specialized libraries?

THE BIRTH OF CUDA

- In 2007 NVIDIA released the first version of CUDA
- Resurgence of interest in AI around 2009
- Dramatic speedup in training thanks to GPU
- Growth of the computing ecosystem (CPU, GPU, TPU, TrueNorth)

We need to abstract the structure of the network from the details of the implementation


47X Higher Throughput Than CPU Server on Deep Learning Inference

Workload: ResNet-50 | CPU: 1X Xeon E5-2690v4 @ 2.6G Hz | GPU: Add 1X Tesla P100 or V100

TENSORFLOW

- Released by Google in 2015
- Tensor manipulation framework specifically designed for ANN
- Handles allocation and management of hardware resources
- Graph based computation

- Most searched term on Google…
- .. But still, one among many (CNTK, PyTorch, Caffe2)

KERAS

- Abstraction layer applied on top of other frameworks (TensorFlow, CNTK, Theano)
- Network description is kept simple and clean
- Fast and simple prototyping and experimentation
- Friendly modular and extensible

Being able to go from idea to result with the least possible delay is key to doing good research.

KERAS

- Models
  - Sequential
  - Functional programming API
- Layers
  - Dense
  - Convolutional (1D, 2D, 3D)
  - Pooling
  - …
- Callbacks
  - Early stopping
  - Tensorboard
  - …

A SIMPLE NETWORK WRITTEN IN KERAS

```
img = layers.Input(shape=(28, 28, 1,), name='images’)
x = layers.Flatten()(img)
x = layers.Dense(100, activation=‘sigmoid’, name='hidden')(x)
out = layers.Dense(10, activation='softmax', name='output')(x)
model = models.Model(img, out)
model.compile(optimizer=‘sgd’, loss=‘categorical_crossentropy’,
metrics=['accuracy’])
model.fit(data, labels, epochs=n_epochs)
```

- www.tensorflow.org

- www.keras.io