Intel® Xeon Phi™ Workshop
Convolutional Neural Network

Alexander Kataev
leading software engineer
Singularis Lab, LLC
ANN

input layer

hidden layer 1  hidden layer 2

output layer
Convolutional Neural Networks
Convolution

\[ o(x, y) = \sum_{k,l} w(k, l) i(x - k, y - l) \]

\[ o(x, y, f) = \sum_{k,l,d} w(k, l, d) i(x - k, y - l, d) \]
for ( od : out_depth )
    for ( id : in_depth )
        for ( y : out_height )
            for ( x : out_width )
                s[od,y,x] = 0
        for ( wy : conv_height )
            for ( wx : conv_width )
                s[od,y,x] += w[od,wy,wx] * in[id, y, x]
Subsampling (pooling)

Input: 4x4

Pool (2,2)

Output: 2x2
Fully Connected (Inner product)

input: 3  fc (6)  out: 6
CNN Frameworks

Theano
Torch
Caffe
Pylearn2
Tensorflow
MXNet
Lasagne
Keras
Chainer
DeepLearnToolbox
Cuda-Convnet
RNNLM
... (over 9000)
CIFAR 10 Dataset

- Airplane
- Automobile
- Bird
- Cat
- Deer
- Dog
- Frog
- Horse
- Ship
- Truck

32x32 color images
10 classes
60000 images
50000 training,
10000 testing

*Learning Multiple Layers of Features from Tiny Images, Alex Krizhevsky, 2009*
Network architecture

Data: input: 32x32x3

Feature 1: 16x16x32
Operation: conv 5x5x32
pool 2x2

Feature 2: 8x8x32
Operation: conv 5x5x32
pool 2x2

Feature 3: inner: 4x4x64
Operation: conv 5x5x64
pool 2x2
fc 64

Out: 64
Softmax: 10
Tiny-cnn

A header only, dependency-free deep learning framework in C++11

• tbb, openmp
• simple code
• SSE/AVX with C++ templates
• CMake

https://github.com/nyanp/tiny-cnn
Changes to repo code

• xeon_toolchain.cmake and phi_toolchain.cmake
• removed dependency on OpenCV
• fixed errors with C++11
• in config.h change float_t to float
phi-toolchain.cmake

SET(CMAKE_SYSTEM_NAME Linux)
SET(CMAKE_SYSTEM_PROCESSOR k1om)
SET(CMAKE_SYSTEM_VERSION 1)

# where is the target environment
SET(CMAKE_FIND_ROOT_PATH /opt/software/intel/lib/mic/)

# search for programs in the build host directories
set(CMAKE_FIND_ROOT_PATH_MODE_PROGRAM NEVER)
# for libraries and headers in the target directories
set(CMAKE_FIND_ROOT_PATH_MODE_LIBRARY ONLY)
set(CMAKE_FIND_ROOT_PATH_MODE_INCLUDE ONLY)

# specify the cross compiler
SET(CMAKE_C_COMPILER icc)
SET(CMAKE_CXX_COMPILER icpc)
SET(CMAKE_C_FLAGS "-mmic " CACHE STRING "" FORCE)
SET(CMAKE_CXX_FLAGS "-mmic " CACHE STRING "" FORCE)
$ cd tiny-cnn
$ mkdir build-xeon
$ cd build-xeon
$ cmake -DUSE_AVX=ON -DCMAKE_TOOLCHAIN_FILE=../xeon_toolchain.cmake ..
$ make
$ ssh <xeon-node>
$ ./example_cifar_train ../../data/ 0.01 ⇐ Do not try on login!!!
learning rate:0.01
load models...
start learning

0%  10  20  30  40  50  60  70  80  90  100%
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
********************************************************************33.8093s elapsed.
392/1000
$ cd tiny-cnn
$ mkdir build-phi
$ cd build-phi
$ cmake -DCMAKE_TOOLCHAIN_FILE=../phi_toolchain.cmake ..
$ make
$ ssh <phi-node>
$ cmake/tiny-cnn/build-phi/example_cifar_train cnn/data/ 0.01 learning rate: 0.01
load models...
start learning

0% 10 20 30 40 50 60 70 80 90 100%
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
**************************************************244.4s elapsed.
393/1000
First run

<table>
<thead>
<tr>
<th>xeon</th>
<th>phi</th>
</tr>
</thead>
<tbody>
<tr>
<td>14 cores, up to 56 threads + AVX</td>
<td>60 cores, up to 240 threads</td>
</tr>
<tr>
<td>33.8s</td>
<td>244.4</td>
</tr>
</tbody>
</table>
Disable automatic vectorization

<table>
<thead>
<tr>
<th>xeon</th>
<th>phi</th>
</tr>
</thead>
<tbody>
<tr>
<td>14 cores, up to 56 threads + AVX</td>
<td>60 cores, up to 240 threads, -no-vec</td>
</tr>
<tr>
<td>--</td>
<td>145.2s</td>
</tr>
</tbody>
</table>

Modify phi-toolchain.cmake:

```cmake
SET(CMAKE_C_FLAGS "-mmic -no-vec" CACHE STRING "" FORCE)
SET(CMAKE_CXX_FLAGS "-mmic -no-vec" CACHE STRING "" FORCE)
```
with KNC (AVX512)

<table>
<thead>
<tr>
<th>xeon</th>
<th>phi</th>
</tr>
</thead>
<tbody>
<tr>
<td>14 cores, up to 56 threads + AVX</td>
<td>60 cores, up to 240 threads, -no-vec</td>
</tr>
<tr>
<td>--</td>
<td>123.1s</td>
</tr>
</tbody>
</table>

Add `-DUSE_AVX=ON` to cmake.
Look into modified product.h for details.
Thanks for attention!

Alexander Kataev
Leading software engineer
Singularis Lab, LLC
alexander.kataev@singularis-lab.com

Dmitry Kryzhanovsky
CEO at Singularis Lab, LLC
dmitry.kryzhanovsky@singularis-lab.com