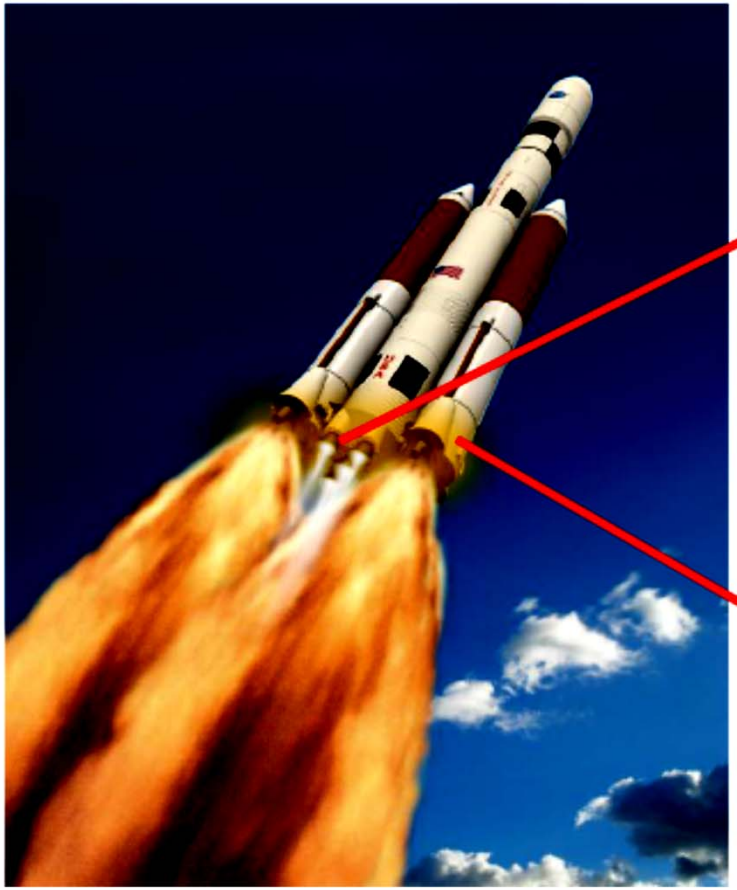
یادگیری عمیق در عمل
Deep Learning in Action

عادل قاضی خانی
هیئت علمی گروه کامپیوتر دانشگاه امام رضا (ع)

یادگیری عمیق

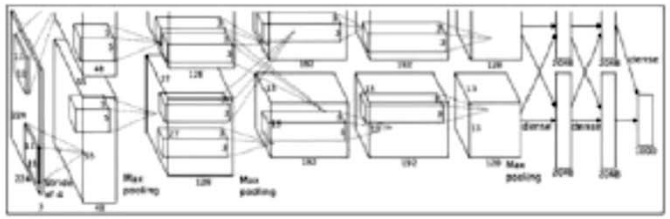
Deep Learning Era



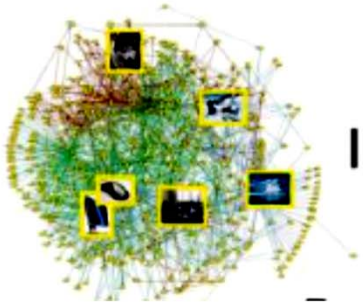
Engine

NVIDIA et al.

"AlexNet"

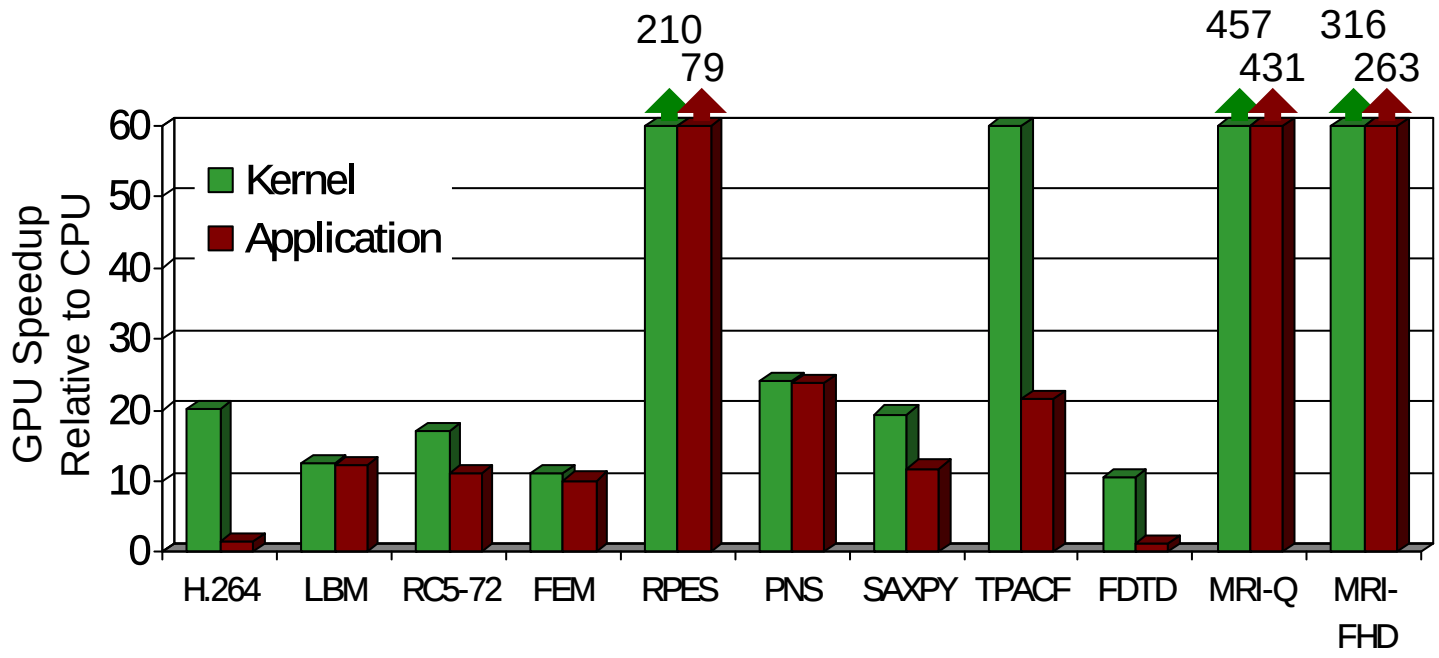


Fuel



IMAGENET

افزایش سرعت در GPU نسبت به CPU



- GeForce 8800 GTX vs. 2.2GHz Opteron 248
- 10× speedup in a kernel is typical, as long as the kernel can occupy enough parallel threads
- 25× to 400× speedup if the function's data requirements and control flow suit the GPU and the application is optimized

کتابخانه های یادگیری عمیق

Caffe2

Caffe

TensorFlow

PYTORCH

dmlc
mxnet

Keras

Chainer

Microsoft
CNTK

نصب کتابخانه ها

- سیستم عامل windows یا ubuntu
- نصب پایتون
- معمولا با دستور pip در پایتون انجام می شود

- مثلا

- pip install http://download.pytorch.org/whl/cpu/torch-0.4.1-cp27-cp27mu-linux_x86_64.whl
- pip install torchvision

نصب کتابخانه ها

❖برای استفاده از GPU برای یادگیری عمیق:

- باید کتابخانه های مورد نظر(CUDA) با توجه به GPU و سیستم عامل دانلود گردد (nvidia.com,...) و طی مراحلی نصب گردد
- معمولا در Ubuntu

نمونه ای از شبکه کانولوشن در کتابخانه Keras

- طبقه بندی داده های MNIST که تصاویر حروف دست نویس هستند

- این شبکه شامل دو بخش:
- ۱- استخراج ویژگی (لایه های کانوولوشن و pooling)
- ۲- طبقه بندی

نمونه ای از شبکه کانولوشن در کتابخانه Keras

- ۱- استخراج ویژگی (لایه های کانوولوشن و pooling)

کتابخانه های پایتون

```
from keras import layers
from keras import models
```

تعداد فیلترها

اندازه فیلترها

لایه های استخراج ویژگی

```
model = models.Sequential()
model.add(layers.Conv2D(32, (3, 3), activation='relu',
input_shape=(28, 28, 1)))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(64, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(64, (3, 3), activation='relu'))
```

نمونه ای از شبکه کانولوشن در کتابخانه Keras

- ۱- استخراج ویژگی (لایه های کانوولوشن و pooling)

```
>>> model.summary()

Layer (type)                     Output Shape          Param #
=================================================================
conv2d_1 (Conv2D)                (None, 26, 26, 32)    320
maxpooling2d_1 (MaxPooling2D)    (None, 13, 13, 32)    0
conv2d_2 (Conv2D)                (None, 11, 11, 64)    18496
maxpooling2d_2 (MaxPooling2D)    (None, 5, 5, 64)      0
conv2d_3 (Conv2D)                (None, 3, 3, 64)      36928
=================================================================
==
Total params: 55,744
```

نمونه ای از شبکه کانولوشن در کتابخانه Keras

- ۲- طبقه بندی

لایه های تماما متصل

```
model.add(layers.Flatten())
model.add(layers.Dense(64, activation='relu'))
model.add(layers.Dense(10, activation='softmax'))
```

تعداد کلاسها

نمونه ای از شبکه کانولوشن در کتابخانه Keras

- ۲- طبقه بندی

```
>>> model.summary()
Layer (type)                     Output Shape          Param #
=================================================================
conv2d_1 (Conv2D)                (None, 26, 26, 32)    320
maxpooling2d_1 (MaxPooling2D)    (None, 13, 13, 32)    0
conv2d_2 (Conv2D)                (None, 11, 11, 64)    18496
maxpooling2d_2 (MaxPooling2D)    (None, 5, 5, 64)      0
conv2d_3 (Conv2D)                (None, 3, 3, 64)      36928
flatten_1 (Flatten)              (None, 576)           0
dense_1 (Dense)                  (None, 64)            36928
dense_2 (Dense)                  (None, 10)            650
=================================================================
Total params: 93,322
```

نمونه ای از شبکه کانولوشن در کتابخانه Keras

- **آموزش شبکه**

کتابخانه های پایتون

```
from keras.datasets import mnist
from keras.utils import to_categorical
```

داده های آموزش و آزمایش

```
(train_images, train_labels), (test_images, test_labels) = mnist.load_data()
train_images = train_images.reshape((60000, 28, 28, 1))
train_images = train_images.astype('float32') / 255

test_images = test_images.reshape((10000, 28, 28, 1))
test_images = test_images.astype('float32') / 255
train_labels = to_categorical(train_labels)
test_labels = to_categorical(test_labels)
```

آموزش شبکه

```
model.compile(optimizer='rmsprop',loss='categorical_crossentropy',
metrics=['accuracy'])
model.fit(train_images, train_labels, epochs=5, batch_size=64)
```