Caffe Workshop

Install and run caffe: (Completed)

1. install Oracle VM VirtualBOX
2. download Ubuntu 14.04 and put it in the virtualbox. After that, tune up the default memory set to at least 4 GB for the later use. (I experienced being informed of lack of memory when do the compilation.)
3. set prerequisites for caffe installation.(dependencies) -----type all the listed necessary command in the terminal. (http://caffe.berkeleyvision.org/install_apt.html
4. download caffe package from github
5. compile all
6. set the caffe path to environment

To train model with caffe:

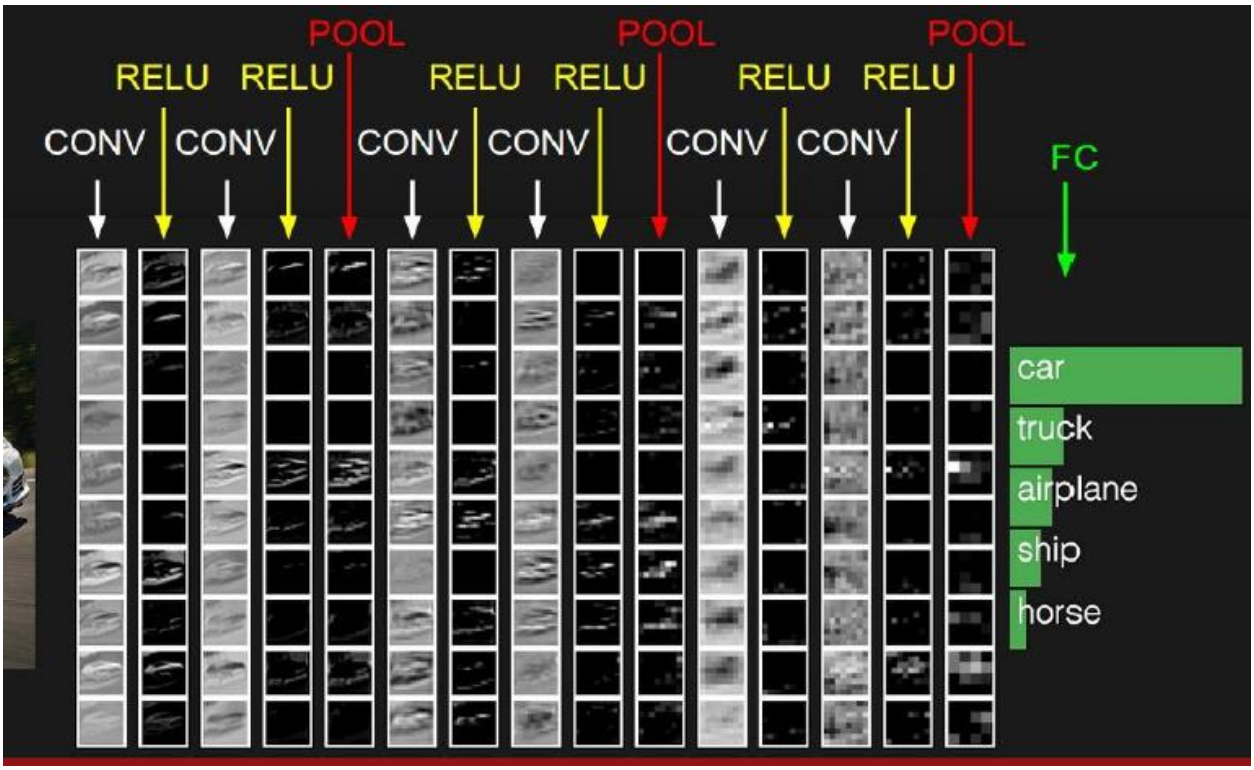
(sample: http://caffe.berkeleyvision.org/gathered/examples/mnist.html)

1. prepare Datasets. (create lmdb)-----two commands (Completed)

2. **write the network definition protobuf and solver protobuf files.(the most complicated and obscure part)**

3. train the model with several commands

How to configure the Conv layers:(3 layers: conv; POOL; FC)

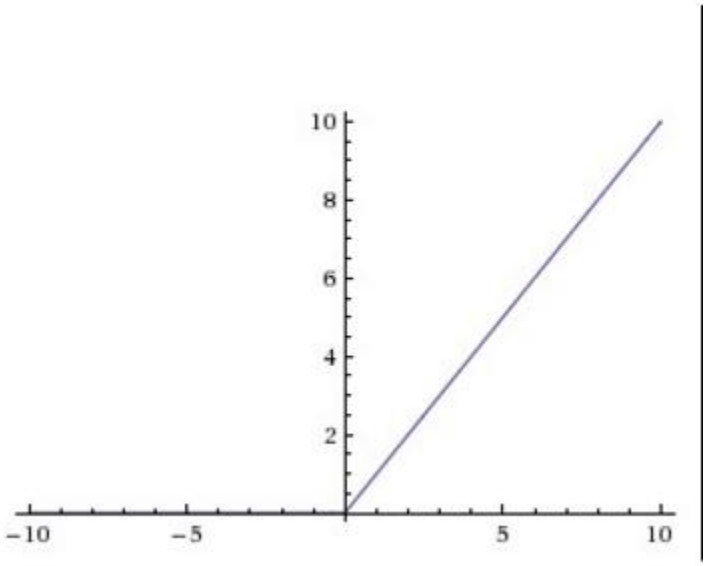


1. convolution layer:

bottom: where the input data goes
top: where the output data comes
lr_mult: learning rate multipliers
decay_mult: decay multipliers
convolution parameters include:

num_output: the number of filters
kernel_size: size of each filter
stride: number of pixels to step between each filter application

- RELU: Rectified Linear Unit activation function



***** In biologically inspired neural networks, the **activation function** is usually an abstraction representing the rate of action potential firing in the cell. In its simplest form, this **function** is binary—that is, either the neuron is firing or not.

2. pooling layer:
to make the representations smaller and more manageable
operates over each activation map independently

the most common type: max pooling:
three Parameters:
their spatial extent F(always set 2 or 3)
the stride S (always set 2)
zero-padding

- Accepts a volume of size $W_1 \times H_1 \times D_1$
- Requires three hyperparameters:
  - their spatial extent F,
  - the stride S,
- Produces a volume of size $W_2 \times H_2 \times D_2$ where:
  - $W_2 = (W_1 - F)/S + 1$
  - $H_2 = (H_1 - F)/S + 1$
  - $D_2 = D_1$
- Introduces zero parameters since it computes a fixed function of the input
- Note that it is not common to use zero-padding for Pooling layers

3. Fully connected Layer (FC layer): contains neurons that connect to the entire input volume, as in ordinary Neural Networks. In other words, neurons between two adjacent layers are fully pairwise connected, but neurons within a single layer share no connections

The architecture of Memnet model is:

Conv-RELU-POOL-LRN(Normalization Layer)-Conv-RELU-POOL-LRN(Normalization Layer) - Conv-RELU-Conv-RELU-Conv-RELU-POOL-InnerProduct-RELU-Dropout-InnerProduct-RELU-Dropout-InnerProduct

(Dropout:

*dropout units in the neural network, along with their incoming and outgoing connections

*prevent neural network from overfitting and provides a way of approximately combining many different neural network architectures efficiently.)

Trends of Architecture of ConvNets:

1. smaller filters and deeper architectures
2. getting rid of POOL and FC layers