Appendix: Building high-level features using large scale unsupervised learning

Quoc V. Le QUOCLE@CS.STANFORD.EDU
Marc'Aurelio Ranzato RANZATO@GOOGLE.COM
Rajat Monga RAJATMONGA@GOOGLE.COM
Matthieu Devin MDEVIN@GOOGLE.COM
Kai Chen KAICHEN@GOOGLE.COM
Greg S. Corrado GCORRADO@GOOGLE.COM
Jeff Dean JEFF@GOOGLE.COM
Andrew Y. Ng ANG@CS.STANFORD.EDU

Abstract

In this appendix, we discuss more details regarding the algorithm, its implementation, test set for 3D-transformed faces, experimental results for parameter sensitivity. We also present further visualizations for the learned neurons.

A. Training and test images

A subset of training images is shown in Figure 1. As can be seen, the positions, scales, orientations of faces in the dataset are diverse. A subset of test images for identifying the face neuron is shown in Figure 2.

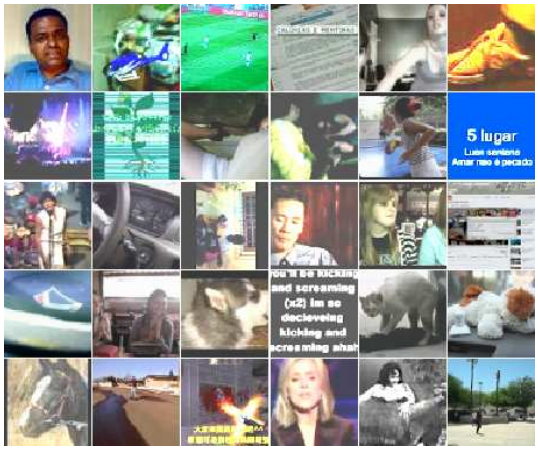

Figure 1. Thirty randomly-selected training images (shown before the whitening step).

Figure 2. Some example test set images (shown before the whitening step).

B. Models

Central to our approach in this paper is the use of locally-connected networks. In these networks, neurons only connect to a local region of the layer below.

In Figure 3, we show the connectivity patterns of the neural network architecture described in the paper. The actual images in the experiments are 2D, but for simplicity, our images in the visualization are in 1D.

C. Model Parallelism

We use model parallelism to distribute the storage of parameters and gradient computations to different machines. In Figure 4, we show how the weights are divided and stored in different "partitions," or more

Appearing in *Proceedings of the 29th International Conference on Machine Learning*, Edinburgh, Scotland, UK, 2012.

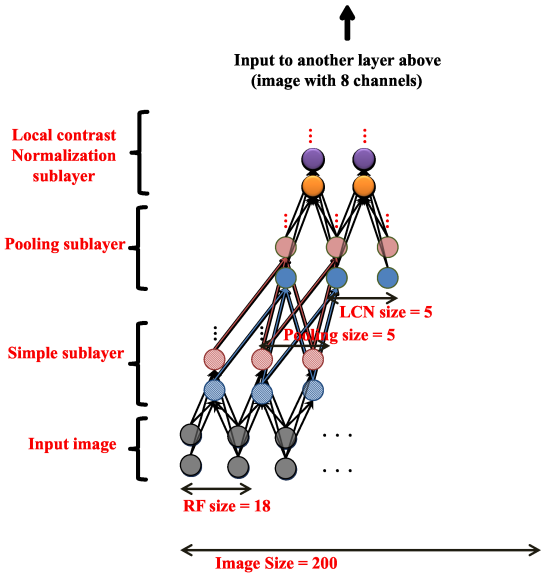

Figure 3. Diagram of the network we used with more detailed connectivity patterns. Color arrows mean that weights only connect to only one map. Dark arrows mean that weights connect to all maps. Pooling neurons only connect to one map whereas simple neurons and LCN neurons connect to all maps.

simply, machines (see also (Krizhevsky, 2009)).

D. Further multicore parallelism

Machines in our cluster have many cores which allow further parallelism. Hence, we split these cores to perform different tasks. In our implementation, the cores are divided into three groups: reading data, sending (or writing) data, and performing arithmetic computations. At every time instance, these groups work in parallel to load data, compute numerical results and send to network or write data to disks.

E. Parameter sensitivity

The hyper-parameters of the network are chosen to fit computational constraints and optimize the training time of our algorithm. These parameters can be changed at the expense of longer training time or more computational resources. For instance, one could increase the size of the receptive fields at an expense of using more memory, more computation, and more network bandwidth per machine; or one could increase the number of maps at an expense of using more machines and memories.

These hyper-parameters also could affect the performance of the features. We performed control experiments to understand the effects of the two hyper-parameters: the size of the receptive fields and the number of maps. By varying each of these parameters and observing the test set accuracies, we can gain an understanding of how much they affect the performance on the face recognition task. Results, shown in Figure 5, confirm that the results are only slightly sensitive to changes in these control parameters.

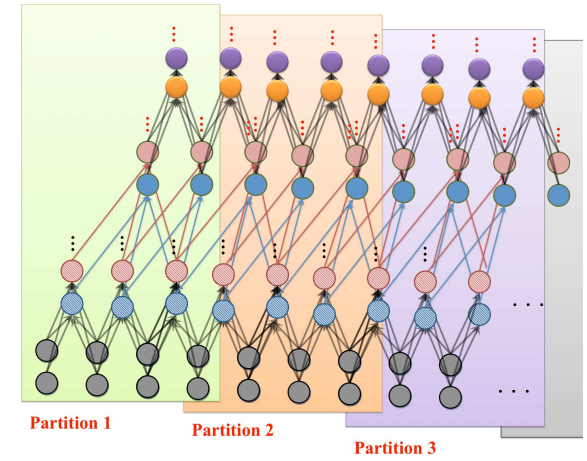

Figure 4. Model parallelism with the network architecture in use. Here, it can be seen that the weights are divided according to the locality of the image and stored on different machines. Concretely, the weights that connect to the left side of the image are stored in machine 1 ("partition 1"). The weights that connect to the central part of the image are stored in machine 2 ("partition 2"). The weights that connect to the right side of the image are stored in machine 3 ("partition 3").

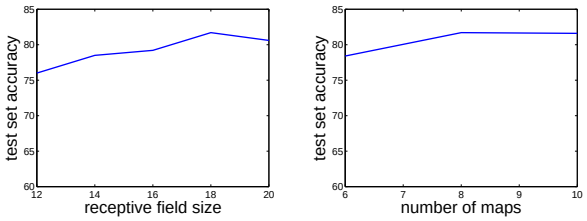

Figure 5. Left: effects of receptive field sizes on the test set accuracy. Right: effects of number of maps on the test set accuracy.

F. Example out-of-plane rotated face sequence

In Figure 6, we show an example sequence of 3D (out-of-plane) rotated faces. Note that the faces are black and white but treated as a color picture in the test. More details are available at the

webpage for The Sheffield Face Database dataset – http://www.sheffield.ac.uk/eee/research/iel/research/face

Figure 6. A sequence of 3D (out-of-plane) rotated face of one individual. The dataset consists of 10 sequences.

G. Best linear filters

In the paper, we performed control experiments to compare our features against "best linear filters."

This baseline works as follows. The first step is to sample 100,000 random patches (or filters) from the training set (each patch has the size of a test set image). Then for each patch, we compute its cosine distances between itself and the test set images. The cosine distances are treated as the feature values. Using these feature values, we then search among 20 thresholds to find the best accuracy of a patch in classifying faces against distractors. Each patch gives one accuracy for our test set.

The reported accuracy is the best accuracy among 100,000 patches randomly-selected from the training set.

H. Histograms on the entire test set

Here, we also show the detailed histograms for the neurons on the entire test sets.

The fact that the histograms are distinctive for positive and negative images suggests that the network has learned the concept detectors.

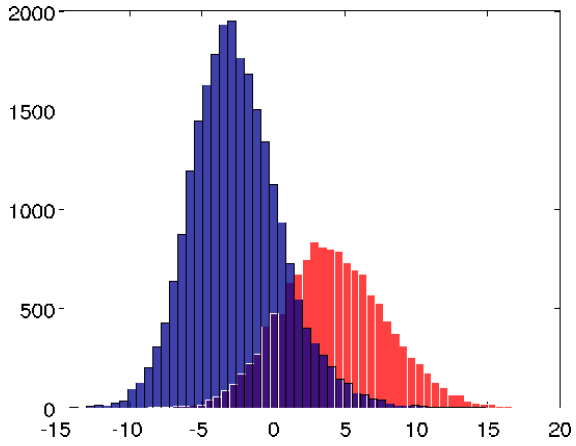

Figure 7. Histograms of neuron's activation values for the best face neuron on the test set. Red: the histogram for face images. Blue: the histogram for random distractors.

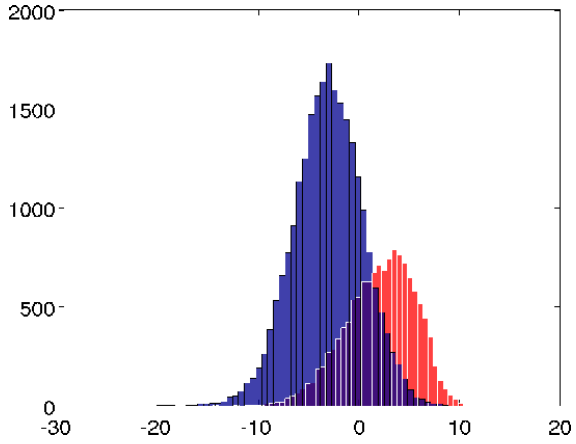

Figure 8. Histograms for the best human body neuron on the test set. Red: the histogram for human body images. Blue: the histogram for random distractors.

I. Most responsive stimuli for cats and human bodies

In Figure 10, we show the most responsive stimuli for cat and human body neurons on the test sets. Note that, the top stimuli for the human body neuron are black and white images because the test set images are black and white (Keller et al., 2009).

References

Keller, C., Enzweiler, M., and Gavrila, D. M. A new benchmark for stereo-based pedestrian detection. In *Proc. of the IEEE Intelligent Vehicles Symposium*, 2009.

Krizhevsky, A. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

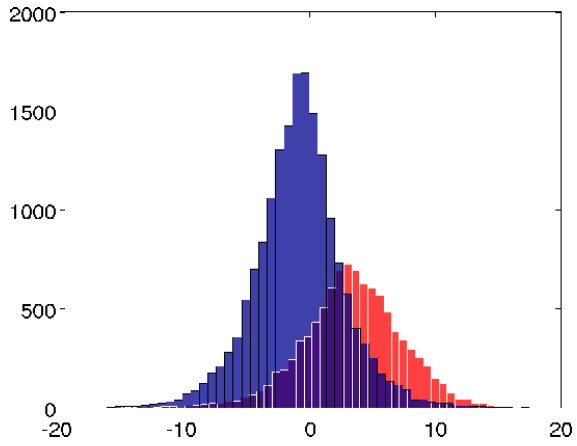

Figure 9. Histograms for the best cat neuron on the test set. Red: the histogram for cat images. Blue: the histogram for random distractors.

Figure 10. Top: most responsive stimuli on the test set for the cat neuron. Right: Most responsive human body stimuli on the test set for the human body neuron.