Supervised Learning of Image Restoration with Convolutional Networks Supplementary Material: Specific Methods

Viren Jain[1], Joseph F. Murray[1,2], Fabian Roth[1,2], Srinivas Turaga[1], Valentin Zhigulin[1,2], Kevin L. Briggman[3], Moritz N. Helmstaedter[3], Winfried Denk[3], and H. Sebastian Seung[1,2]

[1]Brain & Cognitive Sciences, Massachusetts Institute of Technology, Cambridge, MA, USA

[2]Howard Hughes Medical Institute, Cambridge, MA, USA

[3]Biomedical Optics, Max Planck Institute for Medical Research, Heidelberg, Germany

1. Evaluation Methodology

Training set voxel-wise restoration accuracy was found by measuring the percentage of voxels that agreed with a binary human classification. For the convolutional networks, we used the restoration provided by the training epoch at which convergence was observed (i.e., little to no fluctuation in training error for several successive epochs). For methods whose outputs are real-valued, such as the convolutional networks or diffusion, a threshold was applied to produce a binary classification. The threshold value was chosen to maximize accuracy on the training set.

Testing set performance was measured by preserving all parameters, threshold values, etc., from training set optimization and then measuring voxel agreement on a separate part of the image that had beeen manually classified by the same human as for the training set.

Individual samples in voxel-wise accuracy measurement cannot be considered to truly satisfy independent and identically distributed (IID) assumptions due to correlations resulting from a contiguous image space. Therefore, the training and test areas were divided into ten non-overlapping regions, and accuracy was then measured within each region. The standard error of measurement was then calculated from the variance in accuracy among these regions. Although these image regions are not truly statistically independent, we expect the correlations in the accuracy measurements of these regions to be relatively weak, so that the standard error of measurement is a reasonable estimate of the error bar.

2. Procedures

2.1. Convolutional networks

The filters w and thresholds θ are free parameters of the algorithm that are chosen by gradient learning, using a version of the backpropogation algorithm adapted to multi-layer convolutional networks [7]. We used the cross-entropy cost function as our optimization criterion.

A stochastic, online gradient learning procedure was found to be more efficient than batch training (see [3] for an interesting study of the virtues of online learning methods). Each training epoch consisted of a random sequence of $6 \times 6 \times 6$ cubes chosen from the training set. The network parameters were then updated after being evaluated on each cube in the sequence. This learning procedure was highly reliable; larger cube sizes decreased training speed but did not improve training quality. The learning rate of each free parameter was controlled by a stochastic approximation to the diagonal of the hessian matrix [8].

Before learning, a weight in filter w^k_{ab} was initialized to a random value chosen from a normal distribution with zero mean and a variance inversely proportional to the square root of the number of voxels in the filter (e.g., $\frac{1}{\sqrt{125}}$for a $5 \times 5 \times 5$ filter). Although such details were not critical to the success of the learning procedure, in practice they were found to speed convergence rates.

2.2. MRF/CRF Models

2.2.1 Inference

Inference with our MRF and CRF models was done by maximizing the posterior distribution $p(x|y)$ with respect to x, which is known as the maximum a posteriori (MAP) estimate.

For finding the MAP with min-cut [4, 5] we constructed a graph with the nonnegative adjacency weights given by w and source/sink connections given by $\max\{0, +\tilde{b}_i\}$, $\max\{0, -\tilde{b}_i\}$ respectively, where

$$\tilde{b_i} = b + \frac{\xi_1 - \xi_{-1}}{2\sigma^2}\left(y_i - \frac{\xi_1 + \xi_{-1}}{2}\right) \quad (1)$$

for the MRF; and

$$\tilde{b_i} = y_i + b \quad (2)$$

for the CRF. To compute the cut we employed the code provided with [4].

We also tried simulated annealing with a fast annealing schedule for finding an approximation to the MAP solution [9, 6]. At each time step, the new value for a randomly chosen variable x_i was sampled from the conditional distribution

$$P(x_i|x_{-i}, y_i) \propto e^{x_i(\{w\otimes x\}_i+\tilde{b_i})/T} \tag{3}$$

where x_{-i} denotes all variables except for x_i, and T is the temperature. The initial value for the temperature was $T = 10$. This value was decreased by 1/20 each timestep for 80 steps. The MAP estimated by simulated annealing was usually very similar to the MAP computed with min-cut.

2.2.2 MRF Learning

The updates for maximum likelihood estimation of the parameters w and b for the prior are given by Boltzmann machine learning updates [1],

$$\Delta w_j \propto \left\langle \sum_i x_{i+j}x_i \right\rangle_0 - \left\langle \sum_i x_{i+j}x_i \right\rangle_\infty \tag{4}$$

and

$$\Delta b \propto \left\langle \sum_i x_i \right\rangle_0 - \left\langle \sum_i x_i \right\rangle_\infty \tag{5}$$

where $\langle\cdot\rangle_0$ is the average with respect to the empirical distribution given by the data and $\langle\cdot\rangle_\infty$ is the average with respect to the model distribution given by the parameters w and b. Because computing the average with respect to the model distribution requires sampling the entire configuration space it is intractable for the dimensions of our data. We therefore applied pseudolikelihood learning [2] which estimates the model distribution average by the mean field generated by the data,

$$\left\langle \sum_i x_{i+j}x_i \right\rangle_\infty \approx \left\langle \frac{1}{2}\sum_i (x_{i+j}\sigma_i + \sigma_{i+j}x_i) \right\rangle_0 \tag{6}$$

and

$$\left\langle \sum_i x_i \right\rangle_\infty \approx \left\langle \sum_i \sigma_i \right\rangle_0 \tag{7}$$

where $\sigma_i = \tanh\left((w \otimes x)_i + b\right)$ is the conditional $P(x_i|x_{-i}, y_i)$.

We ran the update 400 times with a learning rate of 0.1.

2.2.3 CRF Learning

The Boltzmann updates for CRF learning are the same as for the MRF. However, in CRF learning we replaced the model distribution average by the zero temperature limit computed by the min-cut algorithm. We used 400 updates and a learning rate of 0.01.

2.3. Anisotropic diffusion

We applied a 3 dimensional verison of Perona-Malik anisotropic diffusion to the image, that iterates the following discretized update to image x at each position i [10] :

$$X_i^{t+1} = X_i^t + \frac{\lambda}{|\eta_i|}\sum_{j\in\eta_i} g(\nabla X_{i,j}^t)\nabla X_{i,j}^t \tag{8}$$

where η_i denotes a neighborhood for pixel at position i, t indexes discrete time steps, λ is a diffusion rate, $\nabla X_{i,j}^t = X_i^t - X_j^t$ and g is the edge-stopping function:

$$g(x) = \frac{1}{1+\frac{x^2}{K^2}}. \tag{9}$$

Parameters λ, K, the number of iterations, and the threshold value were optimized by grid search to maximize classification accuracy on the training set. We also tried cohence-enhancing diffusion [11], which on this dataset yielded very similar results to Perona-Malik.

2.4. Segmentation

Segmentations were produced from binary restorations using a simple 3d connected-components algorithm: the restoration was converted into an undirected graph, in which each voxel of the image is a node in the graph and edges are inserted between two nodes when their corresponding voxels are (1) direct neighbors along any of the principle axes in the 3d image space (i.e., a 6-connected neighborhood) and (2) share the same binary restoration value. Segmented domains are then produced by searching for connected components of this graph.

References

[1] D. Ackley, G. Hinton, and T. Sejnowski. A learning algorithm for Boltzmann machines. *Cognitive Science*, 1985. 2

[2] J. Besag. Efficiency of pseudolikelihood estimation for simple Gaussian fields. *Biometrika*, 1977. 2

[3] L. Bottou and Y. LeCun. Large scale online learning. *NIPS*, 2003. 1

[4] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *PAMI*, 2004. 1, 2

[5] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *PAMI*, 2001. 1

[6] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 1983. 2

[7] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1:541–551, 1989. 1

[8] Y. LeCun, L. Bottou, G. Orr, and K. Muller. Efficient backprop. *Neural Networks: Tricks of the Trade*, pages 9–50, 1998. 1

[9] S. Li. *Markov random field modeling in image analysis*. Springer-Verlag, New York, 2001. 2

[10] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *PAMI*, 12(7):629–639, 1990. 2

[11] J. Weickert. Coherence-enhancing diffusion of colour images. *Image and Vision Computing*, 17(3):201–212, 1999. 2