Neural Network Image Deconvolution

John E. Tansley, Martin J. Oldfield and David J.C. MacKay*
Cavendish Laboratory
Cambridge, CB3 0HE. United Kingdom.

ABSTRACT. We examine the problem of deconvolving blurred text. This is a task in which there is strong prior knowledge (*e.g.*, font characteristics) that is hard to express computationally. These priors are implicit, however, in mock data for which the true image is known. When trained on such mock data, a neural network is able to learn a solution to the image deconvolution problem which takes advantage of this implicit prior knowledge. Prior knowledge of image positivity can be hard–wired into the functional architecture of the network, but we leave it to the network to learn most of the parameters of the task from the data. We do not need to tell the network about the point spread function, the intrinsic correlation function, or the noise process.

Neural networks have been compared with the optimal linear filter, and with the Bayesian algorithm MemSys, on a variety of problems. The networks, once trained, were faster image reconstructors than MemSys, and had similar performance.

1 Traditional image reconstruction methods

Optimal linear filters

In many imaging problems, the data measurements $\{d_m\}$ are linearly related to the underlying image $\mathbf{f}$:

$$d_m = \sum_j R_{mj} f_j + \nu_m . \tag{1}$$

The vector ν denotes the inevitable noise which corrupts real data. In the case of a camera which produces a blurred picture, the vector $\mathbf{f}$ denotes the true image, $\mathbf{d}$ denotes the blurred and noisy picture, and the linear operator $\mathbf{R}$ is a convolution defined by the point spread function of the camera. In this special case, the true image and the data vector reside in the same space; but it is important to maintain a distinction between them. We will use the subscript $m = 1 \ldots N$ to run over data measurements, and the subscripts $i, j = 1 \ldots k$ to run over image pixels.

One might speculate that since the blur was created by a linear operation, then perhaps it might be deblurred by another linear operation. We derive the *optimal linear filter* in two ways.

Bayesian derivation

We assume that the linear operator $\mathbf{R}$ is known, and that the noise ν is Gaussian and independent, with a known standard deviation σ_ν.

$$P(\mathbf{d}|\mathbf{f}, \sigma_\nu, \mathcal{H}) = \frac{1}{(2\pi\sigma_\nu^2)^{N/2}} \exp\left(- \sum_m \left(d_m - \sum_j R_{mj} f_j \right)^2 \Big/ \left(2\sigma_\nu^2\right) \right) \tag{2}$$

*Corresponding author. Email: mackay@mrao.cam.ac.uk.

We assume that the prior probability of the image is also Gaussian, with a standard deviation σ_f.

$$P(\mathbf{f}|\sigma_f,\mathcal{H}) = \frac{\det^{-\frac{1}{2}}\mathbf{C}}{(2\pi\sigma_f^2)^{k/2}}\exp\left(-\sum_{i,j} f_i C_{ij} f_j / (2\sigma_f^2)\right) \tag{3}$$

If we assume no correlations among the pixels then the symmetric, full rank matrix $\mathbf{C}$ is equal to the identity matrix $\mathbf{I}$. The more sophisticated 'intrinsic correlation function' model uses $\mathbf{C} = [\mathbf{G}\mathbf{G}^{\mathrm{T}}]^{-1}$, where $\mathbf{G}$ is a convolution that takes us from an imaginary 'hidden' image, which is uncorrelated, to the real correlated image. The intrinsic correlation function should not be confused with the point spread function $\mathbf{R}$ which defines the image to data mapping. A zero-mean Gaussian prior is clearly a poor assumption if it is known that all elements of the image $\mathbf{f}$ are positive but let us proceed. We are now able to infer the posterior probability of an image $\mathbf{f}$ given the data $\mathbf{d}$.

$$P(\mathbf{f}|\mathbf{d},\sigma_\nu,\sigma_f,\mathcal{H}) = \frac{P(\mathbf{d}|\mathbf{f},\sigma_\nu,\mathcal{H})P(\mathbf{f}|\sigma_f,\mathcal{H}))}{P(\mathbf{d}|\sigma_\nu,\sigma_f,\mathcal{H})} \tag{4}$$

In words,

$$\text{Posterior} = \frac{\text{Likelihood} \;\times\; \text{Prior}}{\text{Evidence}}. \tag{5}$$

The 'evidence' $P(\mathbf{d}|\sigma_\nu,\sigma_f,\mathcal{H})$ is the normalizing constant for this posterior distribution. Here it is unimportant, but it is used in a more sophisticated analysis to compare, for example, different values of σ_ν and σ_f, or different point spread functions $\mathbf{R}$.

Since the posterior distribution is the product of two Gaussian functions of $\mathbf{f}$, it is also a Gaussian, and can therefore be summarized by its mean, which is also the *most probable image*, $\mathbf{f}_{\mathrm{MP}}$, and its covariance matrix:

$$\Sigma_{\mathbf{f}|\mathbf{d}} \equiv [-\nabla\nabla \log P(\mathbf{f}|\mathbf{d},\sigma_\nu,\sigma_f,\mathcal{H})]^{-1}, \tag{6}$$

which defines the joint error bars on $\mathbf{f}$. In this equation, the symbol ∇ denotes differentiation with respect to the parameters $\mathbf{f}$. We can find $\mathbf{f}_{\mathrm{MP}}$ by differentiating the log of the posterior, and solving for the derivative being zero. We obtain:

$$\mathbf{f}_{\mathrm{MP}} = \left[\mathbf{R}^{\mathrm{T}}\mathbf{R} + \frac{\sigma_\nu^2}{\sigma_f^2}\mathbf{C}\right]^{-1}\mathbf{R}^{\mathrm{T}}\mathbf{d}. \tag{7}$$

The operator $\left[\mathbf{R}^{\mathrm{T}}\mathbf{R} + \frac{\sigma_\nu^2}{\sigma_f^2}\mathbf{C}\right]^{-1}\mathbf{R}^{\mathrm{T}}$ is called the optimal linear filter. When the term $\frac{\sigma_\nu^2}{\sigma_f^2}\mathbf{C}$ can be neglected, the optimal linear filter is the pseudoinverse "$\mathbf{R}^{-1}$" $= [\mathbf{R}^{\mathrm{T}}\mathbf{R}]^{-1}\mathbf{R}^{\mathrm{T}}$. The term $\frac{\sigma_\nu^2}{\sigma_f^2}\mathbf{C}$ 'regularizes' this ill-conditioned inverse.

The optimal linear filter can also be manipulated into the form:

$$\text{Optimal linear filter} = \mathbf{C}^{-1}\mathbf{R}^{\mathrm{T}}\left[\mathbf{R}\mathbf{C}^{-1}\mathbf{R}^{\mathrm{T}} + \frac{\sigma_\nu^2}{\sigma_f^2}\mathbf{I}\right]^{-1}. \tag{8}$$

Minimum square error derivation

The orthodox derivation of the optimal linear filter starts by assuming that we will 'estimate' the true image $\mathbf{f}$ by a linear function of the data:

$$\hat{\mathbf{f}} = \mathbf{W}\mathbf{d}. \tag{9}$$

The linear operator $\mathbf{W}$ is then 'optimized' by minimizing the expected sum-squared error between $\hat{\mathbf{f}}$ and the unknown true image . (Interestingly, any quadratic metric using a symmetric positive definite matrix gives the same optimal linear filter.) In the following equations, summations over repeated indices i, j, m are implicit. The expectation $\langle\cdot\rangle$ is over both the statistics of the random variables $\{\nu_m\}$, and the ensemble of images $\mathbf{f}$ which we expect to bump into. We assume that the noise is zero mean and uncorrelated to second order with itself and everything else, with $\langle \nu_m \nu_{m'} \rangle = \sigma_\nu^2 \delta_{mm'}$.

$$\begin{aligned} \langle E \rangle &= \frac{1}{2} \left\langle (W_{im} d_m - f_i)^2 \right\rangle & (10) \\ &= \frac{1}{2} \left\langle (W_{im} R_{mj} f_j - f_i)^2 \right\rangle + \frac{1}{2} W_{im} W_{im} \sigma_\nu^2. & (11) \end{aligned}$$

Differentiating, and introducing $\mathbf{F} \equiv \langle f_{j'} f_j \rangle$ (cf $\sigma_f^2 \mathbf{C}^{-1}$ in the Bayesian derivation above), we find that the optimal linear filter is:

$$\mathbf{W}_{\text{opt}} = \mathbf{F}\mathbf{R}^{\mathrm{T}} \left[\mathbf{R}\mathbf{F}\mathbf{R}^{\mathrm{T}} + \sigma_\nu^2 \mathbf{I} \right]^{-1}. \tag{12}$$

If we identify $\mathbf{F} = \sigma_f^2 \mathbf{C}^{-1}$, we obtain the optimal linear filter (8) of the Bayesian derivation. The ad hoc assumptions made in this derivation were the choice of a quadratic error measure, and the decision to use a linear estimator. It is interesting that without explicit assumptions of Gaussian distributions, this derivation has reproduced the same estimator as the Bayesian posterior mode, $\mathbf{f}_{\text{MP}}$.

Other image models

The better matched our model of images $P(\mathbf{f}|\mathcal{H})$ is to the real world, the better our image reconstructions will be, and the less data we will need to answer any given question. The Gaussian models which lead to the optimal linear filter fail to specify that all images are positive. This leads to the most pronounced problems where the image under observation has high contrast. Optimal linear filters applied to radio astronomical data give reconstructions with negative areas in them, corresponding to patches of sky that suck energy out of radio telescopes. The 'Maximum Entropy' model for image deconvolution [2] was a great success principally because this model forced the reconstructed image to be positive. The spurious negative areas and complementary spurious positive areas are eliminated, and the dynamic range of the reconstruction is greatly enhanced.

The 'Classic maximum entropy' model assigns an entropic prior $P(\mathbf{f}|\alpha, \mathbf{m}, \mathcal{H}_{\text{Classic}}) = \exp(\alpha S(\mathbf{f}, \mathbf{m}))/Z$, where $S(\mathbf{f}, \mathbf{m}) = \sum_i (f_i \log(m_i/f_i) + f_i - m_i)$ [6]. This model enforces positivity; the parameter α defines a characteristic dynamic range by which the pixel values are expected to differ from the default image $\mathbf{m}$.

The 'ICF maximum entropy' model [1] introduces an expectation of spatial correlations into the prior on $\mathbf{f}$ by writing $\mathbf{f} = \mathbf{G}\mathbf{h}$, where $\mathbf{G}$ is a convolution with an intrinsic correlation function, and putting a classic maxent prior on $\mathbf{h}$.

The 'Fermi-Dirac' model generalizes the entropy function so as to enforce an upper bound on intensity as well as the lower bound of positivity. This model is appropriate where the underlying image is bounded between two grey levels, as in the case of printed text.

All these models are implemented in the MemSys package.

2 Supervised neural networks for image deconvolution

'Neural network' researchers often exploit the following strategy. Given a problem currently solved with a standard data modelling algorithm: interpret the computations performed by the algorithm as a parameterized mapping from an input to an output, and call this mapping a neural network; then adapt the parameters to examples of the desired mapping so as to produce another mapping that solves the task better. By construction, the neural network can reproduce the standard algorithm, so this data-driven adaptation can (one expects) only make the performance better.

There are several reasons why standard algorithms can be bettered in this way. (1) Algorithms are often not designed to minimize the real objective function. For example, in speech recognition, a hidden Markov model is designed to model the speech signal, whereas the real objective is to discriminate between different words. If an inadequate model is being used, the neural-net-style training of the model will focus the resources of the model on the aspects relevant to the discrimination task. Discriminative training of hidden Markov models for speech recognition does improve their performance. (2) The neural network can be more flexible than the standard model; some of the adaptive parameters might have been viewed as fixed features by the original designers. (3) The net can find properties in the data that were not included in the original model.

In this paper we apply this neural network attitude to a toy image reconstruction problem. The task is to reconstruct an image of a piece of text from blurred data. This is not viewed as a character recognition task: we perform the reconstruction on a pixel by pixel basis; the neural network is expected to learn general characteristics of the font, but not to memorize the alphabet. We start from the optimal linear filter. If the point spread function is a convolution, then the filter of equation (9) should also be a convolution. Such a filter can be viewed as the very simplest neural network — a single linear neuron that computes:

$$\hat{f}_{(x,y)} = \sum_{(u,v)} w_{(u,v)} d_{(x+u,y+v)}. \tag{13}$$

where (x, y) label the coordinates of points in the image. The neuron has a two-dimensional input which might be about twice the size of the point spread function, and a single output corresponding to a single pixel in the image. The network receives a patch from a data image $\mathbf{d}$ as input, and its single output would be trained to produce the pixel value at the centre of that patch of data in the true image $\mathbf{f}$. As the trained network is scanned across a blurred image, its output produces a deconvolved image, pixel by pixel. The minimum square error derivation of the optimal linear filter in the previous section corresponds to training this neuron on an ensemble of examples $\{\mathbf{d}, \mathbf{f}\}$ where the original images $\mathbf{f}$ have correlations defined by the matrix $\mathbf{F}$.

The first advantage of training such a neuron on real data is that the neuron can

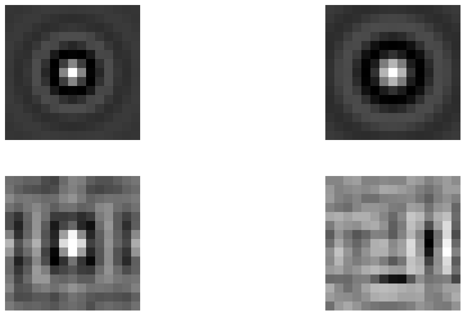

Figure 1: **Optimal linear filters and neural networks**

implicitly learn the correlations $\mathbf{F}$ from the data. One need not explicitly know the point spread function $\mathbf{R}$, the noise statistics σ_ν^2 or the correlation statistics $\mathbf{F}$; the optimization process implicitly learns all these for itself. This network could also learn the appropriate filter if the noise in the data were spatially correlated. Further advantages of a neural network approach arise when we imagine using a more sophisticated network than a linear one. By changing the function performed by the output unit, we can hard-wire prior knowledge into the net. For example, if we know that the true image is everywhere positive, then we can use a non-linear output function which only assumes positive values. In the toy problem studied here, we know that the true image has only two possible intensity levels (black and white, or $t = 0$ and $t = 1$), so we can make the network into a classifier which discriminates between these different possibilities. We define the output of the network to be:

$$P(t = 1|\mathbf{d}, \mathbf{w}) = \frac{1}{1 + e^{-(\mathbf{d}\cdot\mathbf{w} + w_0)}}. \tag{14}$$

By introducing additional non-linear processing between the input and the output, one might allow the network to select from a richer space of non-linear filters. Such a network could implicitly learn a more complicated prior probability distribution for images, learn a more complicated noise model, and learn about non-linear detector responses. We do not go that far in this paper. Here we report the performance achievable using just a single neuron.

Training with limited amounts of data

If our training set $\{\mathbf{d}, \mathbf{f}\}$ is small in size, a network trained to minimize the error on training data will 'overfit' the data. We cope with this by putting a standard Gaussian prior on the network parameters. We find parameters $\mathbf{w}$ that maximize the posterior probability, *i.e.*, the product of the likelihood (factors of the form (14)) and the prior. We optimize the variance of the prior (the 'weight decay constant') using approximate Bayesian methods [5, 4]. (Amusingly, these Bayesian regularization methods are descended from those developed in the Bayesian Maximum entropy method.)

Figure 2: From left to right: original image; blurred and noisy data; reconstruction by MemSys; reconstruction by trained network.

Example

We created data sets from an image of text, using various degrees of blurring and adding various amounts of noise. The blur was spatially Gaussian, and noise was additively Gaussian. For each data set, an optimal linear filter was created, a MemSys deconvolution using the Fermi-Dirac prior was performed, and a neural network was trained on a small patch of the image. The network was trained on three hundred examples (*i.e.*, just a six letter word in the image).

We first contrast the properties of the network with the optimal linear filter. In all cases the trained network outperforms the optimal linear filter in terms of sum-squared error, the difference being greatest for the most difficult problems. In figure 1 we display examples of the parameters of the optimal linear filter and the network. In the case of a small blur radius (a standard deviation of one pixel), the network looks similar to the optimal linear filter except for a slight squareness produced by the font statistics. At a larger blur radius (two pixels standard deviation), the neural net's weights are completely unlike the optimal linear filter, and are also satisfyingly hard for a human to explain — a good sign that the network is doing something useful!

In table 1 we summarize the relative performance of MemSys and a neural network with a 13×13 input. In the cases with blurring and noise, the neural net's performance is slightly inferior to MemSys's. Where there is noise only, the net significantly outperforms MemSys. It is conjectured that the network would have done better had it been trained on more examples (the training set consisted of only 6 characters of text). Figure 2 shows patches of reconstructions given by the two methods for the case of the small blurring radius.

The comparison between these methods favours the network most strongly when we turn to the computational requirement. Once trained, a network can process an image in seconds (about 8 seconds for a well-programmed network with a 13×13 input on a 256×256 image). Whereas MemSys takes 10–15 minutes to process the same image.

3 Discussion

The neural network approach has proved a viable image reconstruction strategy in a problem where there are strong implicit priors in the data. Bayesian Maxent image reconstruction with MemSys depends on knowledge of the point spread function, and assumptions about the noise process and the prior on images. In contrast, the network approach requires examples of data for which the true image is known, but does not require explicit knowledge

Blur radius	Noise level	'Difficulty'	MemSys error	Net error
1	medium	15.7	7.4	8.8
2	medium	26.3	16.3	18.2
0	high	66.9	22.0	13.9

Table 1: **Performance of network relative to MemSys**
The 'difficulty' of a task is the sum-squared error between the data and the true image. The performance measure for reconstruction is the sum-squared error between the reconstruction and the true image. Both are in the same arbitrary units.

of the point spread function, noise level, or image statistics; these are 'learnt' implicitly from the data, so that our reconstruction ability is not limited by our inability to express a good prior over images. Once trained, a neural network is a much faster image reconstruction device.

It will be interesting to attempt more realistic problems, and investigate networks using more complex non-linear computations. A more sophisticated form of prior knowledge that could be incorporated is the spatial smoothness of the point spread function, which leads us to expect spatial smoothness in the deconvolving filter also. This prior expectation can be incorporated by changing the regularizer from $\alpha \sum W_m W_m/2$ to $\alpha \sum C_{mm'} W_m W_{m'}/2$, with appropriate cross terms between the parameters. Equivalently, one can retain the former regularizer, and blur the input data before feeding it to the network. This may sound surprising, but blurring the data even more can indeed enhance the performance of such networks [3].

References

[1] S.F. Gull. Developments in maximum entropy data analysis. In J. Skilling, editor, *Maximum Entropy and Bayesian Methods, Cambridge 1988*, pages 53–71, Dordrecht, 1989. Kluwer.

[2] S.F. Gull and G.J. Daniell. Image reconstruction from incomplete and noisy data. *Nature*, 272:686–690, 1978.

[3] I. Guyon, V.N. Vapnik, B.E. Boser, L.Y. Bottou, and S.A. Solla. Structural risk minimization for character recognition. In J.E. Moody, S.J. Hanson, and R.P. Lippmann, editors, *Advances in Neural Information Processing Systems 4*, pages 471–479, San Mateo, California, 1992. Morgan Kaufmann.

[4] D.J.C. MacKay. The evidence framework applied to classification networks. *Neural Computation*, 4(5):698–714, 1992.

[5] D.J.C. MacKay. A practical Bayesian framework for backpropagation networks. *Neural Computation*, 4(3):448–472, 1992.

[6] J. Skilling. Classic maximum entropy. In J. Skilling, editor, *Maximum Entropy and Bayesian Methods, Cambridge 1988*, Dordrecht, 1989. Kluwer.