Adaptive Calibration of Imaging Array Detectors

Marco Budinich and Renato Frison¹ Dipartimento di Fisica & INFN, Via Valerio 2, 34127 Trieste, Italy (Submitted to: Neural Computation - June 13th 1998)

Abstract: In this paper we present two methods for non-uniformity correction of imaging array detectors based on neural networks, both of them exploit image properties to supply lack of calibrations and maximize the entropy of the output. The first method uses a self-organizing net that produces a linear correction of the raw data with coefficients that adapt continuously. The second method employs a kind of contrast equalization curve to match pixel distributions. Our work originates from Silicon detectors but the treatment is general enough to be applicable to many kinds of array detectors like those used in Infrared imaging or in high energy physics.

1. Introduction

One substantial problem of image detectors is that of non uniform response i.e. the same flux of photons do not produce the same output on different pixels. To restore uniformity it is necessary to correct individually each pixel and appropriate calibration procedures can determine the needed parameters.

In a certain detectors like Infrared [Scribner 1991 & 1993, Bolduc 1996] and Silicon detectors [Arfelli 1996], intrinsic instabilities require frequent re-calibrations. We propose here two new neural networks for continuous detector self adjustment eliminating the need of specific, repeated, calibrations. Our idea traces back to the observation that biological photoreceptors do not need calibrations and still can cope with substantially different pixel responses.

Purpose of this paper is to show that it's possible to calculate pixel parameters without explicit calibrations substituting the lacking information with some hypotheses on incoming data. A simple example is this: if one knows that pixel A always receives the same flux of pixel B, then, after a few images, one can get the relative calibration of A and B: the knowledge that A and B must see the same partly replaces information that would come from a calibration.

We will present two different methods, based on two different sets of hypotheses. Our first method make the hypothesis that images arriving on the detector follow a Gibbs

¹mbh@trieste.infn.it, http://www.ts.infn.it/~mbh/MBHgeneral.html, renato@mediastudio.it

distribution [Rangarajan 1995]; from here we derive a learning rule for a self-organizing network of the type analyzed by Yuille [Yuille 1995] that can correct pixel inequalities.

Our second method makes two hypotheses: that all pixels have same distribution and that the mutual information of the data channel is maximal (see e.g. [Laughlin 1987] and [Atick 1992]). Also these hypotheses suffice to design an effective self-organizing network.

In section 2 we define mathematically the problem; following sections explain the underlying theory of our networks while last section is dedicated to numerical results obtained both on synthetic and on real world images.

2. Standard non-uniformity correction

For each pixel $\,i\,$ the detected signal $\,x_i\,$ depends on the photon flux $\,\varphi_i$; let's assume

(1)
$$\mathbf{x}_{\mathbf{i}} = \phi_{\mathbf{i}} \, \mathbf{g}_{\mathbf{i}} + \mathbf{o}_{\mathbf{i}}$$

where g_i and o_i represent gain and offset of the *i*-th ensemble detector-electronics and these coefficients embody all the details of the energy conversion process.

When a constant flux of photons hits the detector (i.e. all ϕ_i are equal, a "flat field"), one usually gets a noisy image because gain and offset are different for each pixel². The general problem is to find pixel coefficients α_i and β_i restoring uniformity in the corrected image y_i

(2)
$$y_i = \alpha_i x_i + \beta_i;$$

the ideal case is $\alpha_i = \frac{1}{g_i}$ and $\beta_i = -\frac{o_i}{g_i}$ giving $y_i = \phi_i$.

The first two images of figure 6 show examples after and before this correction. This image is a test digital radiography recorded by the SYRMEP project (SYnchrotron Radiation for Medical Physics, see [Arfelli 1996]) that efficiently detects X rays by means of a Silicon chip subdivided in 48 pixels (diodes). Like in scanners, a bi-dimensional image is obtained moving the detector relatively to the specimen and the complete image is built incrementally. This explains why different gains and offsets in each pixel produce horizontal lines in the rough image.

The traditional 'two points' calibration method requires two images with known, uniform, photon fluxes $\vec{\phi}^1$ and $\vec{\phi}^2$ (arrows indicate vectors and upper index images), that, for a detector with N pixels, give rise to a system of 2 N equations (2)

 $^{^{2}}$ We ignore another source of non uniformity due to the stochastic nature of the photon conversion process, observable for example, reading several times the same pixel at a constant flux. This kind of noise is negligible when there are sufficiently many photons.

 $\alpha_i x_i^j + \beta_i = \phi_i^j \ (i = 1, \dots, N \quad j = 1, 2) \ \text{ in the 2 N unknowns } \alpha_i \ \text{ and } \ \beta_i \ \text{ that in block matrix form are:}$

(3)
$$\mathbf{X}\mathbf{\vec{c}} = \mathbf{\vec{\phi}}_{\mathrm{T}}$$

where $\vec{c}' = (\alpha_1, \alpha_2, ..., \alpha_N, \beta_1, \beta_2, ..., \beta_N)$ is the 2N vector of the unknowns ('stands for transpose), $\vec{\phi}_T' = (\phi_1^1, \phi_2^1, ..., \phi_N^1, \phi_1^2, \phi_2^2, ..., \phi_N^2)$ the 2N vector of the fluxes and $\mathbf{X} = \begin{pmatrix} \mathbf{X}_D^1 & \mathbf{1} \\ \mathbf{X}_D^2 & \mathbf{1} \end{pmatrix}$ a block square matrix, of dimension 2N, containing raw

detector data in diagonal matrices of size N $\mathbf{X}_{D}^{j} = \begin{pmatrix} x_{1}^{j} & 0 & \dots & 0 \\ 0 & x_{2}^{j} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & x_{D}^{j} \end{pmatrix}$.

In non pathological cases (i.e. full rank ${\bf X}$), the exact solution $\vec{c} = {\bf X}^{-1} \vec{\varphi}_T$ is:

(4)
$$\vec{c}' = \left(\frac{1}{g_1}, \dots, \frac{1}{g_N}, -\frac{o_1}{g_1}, \dots, -\frac{o_N}{g_N}\right).$$

When detector coefficients g_i and o_i vary with time this calibration procedure has to be repeated frequently.

In most real world problems one ignores the values of the fluxes $\vec{\phi}^1$ and $\vec{\phi}^2$ and has only a linear transformation of the fluxes, $a \quad \vec{\phi}^j + b$, with unknown coefficients a and b. In this case the general solution of (3) is a whole 2 dimensional 'solutions space' that can be written as a linear combination of two linearly independent vectors

(5)
$$\vec{c}' = a \left(\frac{1}{g_1}, \dots, \frac{1}{g_N}, -\frac{o_1}{g_1}, \dots, -\frac{o_N}{g_N} \right) + b \left(0, \dots, 0, 1, \dots, 1 \right)$$

the first of which is the exact solution (4) while we call the second 'non informative solution' because, even if it satisfies exactly our system, it produces a perfectly flat image completely uncorrelated to the incoming flux. Any vector (5) is a valid solution and different vectors produce equally good images differing only by overall multiplicative and additive constants.

We examine now adaptive correction techniques that search a correction vector \vec{c} during image acquisition without a separate calibration procedure; our goal is, in a sense, to find a solution of (3) without knowing $\vec{\phi}_T$. Advantages are obvious: it's not necessary to perform a costly calibration procedure and \vec{c} can 'follow' the detector if its properties change in time. In the next two sections we give two different methods of adaptive calibration both presented in the neural network paradigm.

3. Adaptive correction by means of flux estimation

The heart of this method is the calculation of an estimate \vec{f} to replace the real fluxes in (3) based only on detector information and properties of natural images. We begin, along a track opened by Scribner [Scribner 1991, Scribner 1993], presenting a simpler method to finish with a better procedure relying on a more quantitative model of natural images distribution that will solve the problem.

Natural images have amplitude spectra inversely proportional to the frequency in every direction of the Fourier plane [Field 1994], this means that low spatial frequencies are the most common ones and consequently neighboring pixels tend to see the same. This argument, together with arguments based on the connections of biological photoreceptors and their ability to correct adaptively, suggest to estimate the real flux ϕ_i by the average of neighboring pixels

(6)
$$f_i = \frac{1}{n_i} \sum_{k \in V_i} y_k$$

where V_i represents the set of the n_i pixels neighboring i. Substituting these estimates to the real fluxes ϕ_i in (3) we obtain a new system of equations

(7)
$$\alpha_i x_i^j + \beta_i = \frac{1}{n_i} \sum_{k \in V_i} \alpha_k x_k^j + \beta_k \quad i = 1...N \quad j = 1...P$$

where $P \ge 2$ is the total number of images. To simplify the notation we introduce the adjacency matrix A: for a detector with N pixels it's a square matrix of size N that lists the adjacency relations of each pixel with the weights given by (7).

For example, in our particular case of a one dimensional detector and a set of neighbors of two pixels, ${\bf A}$ is a tridiagonal matrix

(8)
$$\mathbf{A} = \begin{pmatrix} 1 & -1 & 0 & 0 & \dots & 0 \\ -\frac{1}{2} & 1 & -\frac{1}{2} & 0 & \dots & 0 \\ 0 & -\frac{1}{2} & 1 & -\frac{1}{2} & \dots & 0 \\ 0 & \dots & \dots & \dots & \dots & 0 \\ 0 & \dots & 0 & -\frac{1}{2} & 1 & -\frac{1}{2} \\ 0 & \dots & 0 & 0 & -1 & 1 \end{pmatrix}$$

and system (3) can be written succinctly using the block matrix \mathbf{D} of size $N P \ge 2 N$

(9)
$$\mathbf{D}\vec{\mathbf{c}} = \vec{\mathbf{0}}$$
 where $\mathbf{D} = \begin{pmatrix} \mathbf{A}\mathbf{X}_{\mathbf{D}}^{1} & \mathbf{A} \\ \mathbf{A}\mathbf{X}_{\mathbf{D}}^{2} & \mathbf{A} \\ \cdots & \cdots \\ \mathbf{A}\mathbf{X}_{\mathbf{D}}^{\mathbf{P}} & \mathbf{A} \end{pmatrix}$

where X_D^j is the data matrix of (3) and \vec{c} the vector of the unknowns α_i and β_i .

Usually there are P >> 2 images and the system $\mathbf{D}\vec{c} = \vec{0}$ is overdetermined having more equations than unknowns. In general these systems do not admit an exact solution and one takes as solution the vector \vec{c} that minimizes the semipositive definite quantity (the sum of the squared 'reminders'):

(10)
$$\vec{\mathbf{r}} \cdot \vec{\mathbf{r}} = \vec{\mathbf{c}} \cdot \mathbf{D} \cdot \mathbf{D} \vec{\mathbf{c}}$$

to which we add a constraint to keep $~\vec{c}~$ normalized at 1 to escape the trivial solution $\vec{c}~=~\vec{0}$. The quantity to minimize becomes

(11)
$$\vec{c} \cdot \vec{D} \cdot \vec{D} \vec{c} + \lambda(\vec{c} \cdot \vec{c} - 1)$$

and deriving it with respect to \vec{c} one finds that the solution is given by the eigenvector of D'D corresponding to the minimal eigenvalue. Usually this 2 N square matrix is too complex to be diagonalized analytically but one can gain some information from the structure of the adjacency matrix A.

In our particular case the adjacency matrix (8) has rank N-1³ from which it derives that **D'D** has rank 2 N-1 and thus its minimum eigenvalue is 0. It's easy to check that the corresponding eigenvector is just the non informative solution of (5). This situation is not peculiar of our detector but generalizes to most array-like detectors where the non informative solution always satisfies the system $\mathbf{D}\vec{c} = \vec{0}$ (this derives analytically from the boundary conditions of the adjacency matrix).

We leave aside this problem for a moment to investigate numerical methods of solution and to show that they map easily to a neural network implementation. With the method of the penalty function, (11) can be solved numerically minimizing

(12)
$$\vec{c} \cdot \vec{D} \cdot \vec{D} \vec{c} + q(\vec{c} \cdot \vec{c} - 1)^2$$

 ${\bf q}$ being a parameter. This form is semipositive definite and it's safe to adopt gradient descent and the iterative rule conducing to the minimum is

(13)
$$\vec{\mathbf{c}}_{t+1} = \vec{\mathbf{c}}_t - \eta \mathbf{D}' \mathbf{D} \vec{\mathbf{c}}_t - 2\eta q (\vec{\mathbf{c}}' \vec{\mathbf{c}} - 1) \vec{\mathbf{c}}_t$$

 $\eta\;$ being the usual parameter giving the step size.

This minimization process needs all P images already in D and is thus a batch process. This rule is equivalent to a more handy on-line process [Hertz 1992, Ljung 1977] in which every image is processed separately giving rise to the update rule

³See e.g. [Milotti 1995] or [Frison 1997] for more extensive discussions on eigenvalues and eigenvectors of this kind of matrices.

(14)
$$\vec{c}_{t+1} = \vec{c}_t - \eta \tilde{\mathbf{X}}' \tilde{\mathbf{A}} \tilde{\mathbf{X}} \vec{c}_t - 2\eta q (\vec{c}' \vec{c} - 1) \vec{c}_t \text{ where}$$

$$\tilde{\mathbf{X}} = \begin{pmatrix} \mathbf{X}_{\mathrm{D}}^{\mathrm{t}} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \end{pmatrix}$$
 and $\tilde{\mathbf{A}} = \begin{pmatrix} \mathbf{A}' & \mathbf{A} & \mathbf{A}' & \mathbf{A} \\ \mathbf{A}' & \mathbf{A} & \mathbf{A}' & \mathbf{A} \end{pmatrix}$

 \mathbf{X}_D^t being the diagonal data matrix containing data arrived at time t. In the limit $q \to \infty$ and $\eta \to 0$, \vec{c}_t converges to the solution of the constrained problem (12), i.e. to the unitary eigenvector corresponding to the minimal eigenvalue of D'D. One can also prove that this is the only stable solution of the system [Frison 1997].

The neural network of figure 1 implements rule (14) in a parallel fashion:



Fig. 1: Uniformity correction network: inputs come from the detector and outputs carry data corrected with (2). Each neuron receives its neighbors' outputs and uses them to update its weights with algorithm (14) or (19).

raw data arriving from the detector feeds the input of a layer of neurons, one for each detector pixel and each neuron implements equation (2) transforming raw data x_i into corrected data y_i . In this view the coefficients α_i and β_i represent the weight and the threshold of each linear neuron. Beyond these connections each neuron receives also the outputs of its immediate neighbors (there are no constraints on the neighborhood structure) using them to implement learning rule (14) that modifies coefficients α_i and β_i . We note that the network is not truly local because the second term of (14), normalizing the solution, needs data from all the neurons.

This network behaves as expected converging quickly to the eigenvector of the minimal eigenvalue of $\mathbf{D'D}$. Unfortunately, as already noted, this solution corresponds to the non informative solution and thus, when tested on real data, after a few epochs the output fades reducing to a constant output independent on what's on the input. In order to avoid

convergence to the non informative solution one could add ad hoc constraints to (12) but, despite many attempts, we haven't been able to find a satisfactory one⁴.

Now we add some further information formulating more quantitative hypotheses on the distribution of detector images $P(\vec{\phi})$ and we show that this allows to escape this problem.

Markov random field models [Geman 1984, Besag 1974, Rangarajan 1995] assume that the distribution of natural images $P(\vec{\phi})$ has a local neighborhood structure following a Gibbs distribution

(15)
$$P = \frac{e^{-\beta} U}{Z}$$

where Z is the partition function normalizing the distribution, β the inverse of a temperature and U an energy function of the state of the system defined as a sum of terms each one referring to appropriate neighborhoods (cliques) of the pixels. A very plausible form of energy to adopt for our problem is (10):

(16)
$$\mathbf{U} = \vec{\mathbf{c}}' \, \mathbf{D}' \, \mathbf{D} \vec{\mathbf{c}} \; .$$

We accept this hypothesis and to put at work this information in our net we take the approach of the self-organizing nets of Yuille et al. [Yuille 1995] that replace the energy-like function usually minimized in neural networks by the Kullback Leibler distance between the distribution of the images actually produced and the theoretical distribution of natural images. Figure 2 presents a scheme of the process:



Fig. 2 : Working principles of the self-organizing net of [Yuille 1995].

the image $\vec{\phi}$ of distribution $P(\vec{\phi})$ arrives on the detector producing, via the conversion process f, the image \vec{x} of distribution $P_D(\vec{x})$. This image is then corrected by function g, depending on parameters \vec{c} , to produce the final image \vec{y} of distribution $P_{DD}(\vec{y};\vec{c})$. If the correction process perfectly compensates the distortions introduced by

⁴Substituting x_i to y_i in the flux estimator (6), as made in [Scribner 1991], the non informative solution isn't anymore the stable solution of the system that converges to a "reasonable" solution. Unfortunately, as proved in [Frison 1997], this solution does not coincide with the exact solution (4) neither in the very simple case of completely uniform images.

the detector (i.e. if $g = f^{-1}$) then the corrected image will be equal to the real one and their distributions will coincide.

Yuille and collaborators propose to choose the parameters \vec{c} in such a way that the distributions $P_{DD}(\vec{y};\vec{c})$ and $P(\vec{\varphi})$ are as similar as possible. The criterion adopted to measure the similarity of the distributions is their Kullback Leibler distance (namely the entropy of $P_{DD}(\vec{y};\vec{c})$ relative to $P(\vec{\varphi})$):

(17)
$$KL(\vec{c}) = \int P_{DD}(\vec{y};\vec{c}) \log \frac{P_{DD}(\vec{y};\vec{c})}{P(\vec{y})} d\vec{y} \ge 0$$

that vanishes when the two distributions are equal. So, in a nutshell, the basic idea is: let's vary \vec{c} , thus varying the correction function g, to make the distributions as similar as possible. We will implement this by gradient descent along $KL(\vec{c})$.

In our case of a one dimensional detector $U = \sum_{i} \left(y_i - \frac{1}{2} y_{i-1} - \frac{1}{2} y_{i+1} \right)^2$, $y_i = \alpha_i x_i + \beta_i$ while $P_{DD}(\vec{y}; \vec{c})$ can be calculated from the detector data by the relation $P_{DD}(\vec{y}; \vec{c}) = \frac{P_D(\vec{x})}{\left| \frac{\partial \vec{y}}{\partial \vec{x}} \right|}$. We estimate the Kullback Leibler distance (17) with a discrete

approximation

(18)
$$\sum_{\vec{y}} \log \frac{P_{DD}(\vec{y}; \vec{c})}{P(\vec{y})}$$

where the sum extends to all images. Substituting all ingredients ([Frison 1997] contains the details) one finds the following update rule to descend along the gradient of the Kullback Leibler distance

(19)
$$\vec{c}_{t+1} = \vec{c}_t - \eta \beta \tilde{\mathbf{X}}' \tilde{\mathbf{A}} \tilde{\mathbf{X}} \vec{c}_t + \eta \begin{pmatrix} 1/\vec{\alpha} \\ \vec{0} \end{pmatrix}$$

where the last vector has the first N $\,$ components equal to $\, \frac{1}{\alpha_i} \,$ the last $N \,$ being zero.

The two terms originating from sum (18) combine their effect to determine the two terms of rule (19): the first is equal to that of rule (14) and minimizes the energy of the output (16). This equality is not surprising since both approaches actually minimize (10). The second term of the update rule is peculiar of this approach and derives from a maximum entropy principle encouraging output variability. In practice the second term forbids the system to converge to the non informative solution where all the first N coefficients are zero.

We observe also that here we don't need a normalization term as that present in (14) since rule (19) forbids the trivial solution $\vec{c} = \vec{0}$. This has the notable advantage of making

rule (19) completely "local" and, when implemented on a neural network like that of figure 1, neurons need information only from their neighbors.

4. Adaptive correction via cumulative distribution

The strategy of our second method is that of equalizing, for each pixel, its distribution instead of the single values. With the notation introduced in figure 2 let the distributions of the i-th pixel be $P_D(x_i)$ and $P_{DD}(y_i, \vec{c})$ respectively. In the reasonable hypothesis that the distribution of true images $P(\phi_i)$ is identical for all the pixels we will look for a transformation of the raw data x_i such that the distributions of the corrected data $y_i = g(x_i, \vec{c})$ are all identical. This request is not enough to avoid the non informative solution that give all identical, delta-like, y_i distributions and we need a further request.

Information theory [Bell 1995] provide the second hypothesis: consider every pixel as a deterministic information channel in which the raw data x_i constitutes the input and the corrected data y_i the output. It's well known that the mutual information between input and output is maximal when output distribution is uniform.

We now have sufficient elements to determine uniquely the needed transfer functions $\vec{y} = g(\vec{x}, \vec{c})$: it is sufficient to ask that they give rise to identical output distributions while maximizing the mutual input output information. We remark that with this method the images distributions is not specified; the only request is their equality.

Our two requirements are satisfied if and only if the transfer function is the cumulative input distribution: the simple proof takes the standard relation $P_{DD}(y_i, \vec{c})dy_i = P_D(x_i)dx_i$ and adds the mutual information request $P_{DD}(y_i, \vec{c}) = \text{const.}$ to get:

(20)
$$y_i = \frac{1}{\text{const.}} \int_{-\infty}^{x} P_D(x_i) dx_i .$$

As in the preceding paragraph we search an iterative method to find explicitly the correction functions. Let us start by selecting all transfer functions with output in the same range, say [0, 1]. In this case the maximization of the input -output mutual information is equivalent to the request that the entropy of each output y_i is maximized separately. With these ingredients maximizing the entropy of the outputs is equivalent to take transfer functions as in (20) but with the advantage that the process can be done iteratively. The output entropy is

(21)
$$H(y_i) = -\int_{-\infty}^{+\infty} P_{DD}(y_i, \vec{c}) \log P_{DD}(y_i, \vec{c}) dy_i \stackrel{\text{def}}{=} -\left\langle \log P_{DD}(y_i, \vec{c}) \right\rangle$$

where the <> brackets indicate the expectation value; changing variables we get

(22)
$$H(y_i) = \left\langle \log \left| \frac{dy_i}{dx_i} \right| \right\rangle - \left\langle \log P_D(x_i) \right\rangle$$

where the dependence from the parameters \vec{c} is contained in the term with the derivative. One can maximize this entropy ascending its gradient namely:

(23)
$$\Delta c_k \propto \frac{\partial H}{\partial c_k} = \left(\frac{\partial y_i}{\partial x_i}\right)^{-1} \frac{\partial}{\partial c_k} \left(\frac{\partial y_i}{\partial x_i}\right).$$

To complete the picture and obtain a working algorithm, we have to choose an appropriate family of functions, with values in [0, 1], that can approximate the cumulative distributions (20) with arbitrary precision (here the linear functions (2) are not useful). Once chosen this family of functions we will update their parameters with rule (23) until they approximate closely their objectives (20).

The universal approximation properties of feed-forward neural networks [Hornik 1991] allow us to use a neural network for this task. For example the network of figure 3 can approximate, given sufficient hidden neurons, any continuous function with arbitrary precision. The output of this network is given by

(24)
$$y_i = \sum_{j=1}^{H} w_j h_j = \sum_{j=1}^{H} \frac{w_j}{1 + e^{-\alpha_j x_i - \beta_j}}$$

where $\,H\,$ is the number of hidden neurons, $\,h_{j}\,$ their transfer functions (the usual sigmoidal) and $\,w_{i}\,$ their output weights.



Fig. 3: Basic block of the feed-forward neural network used for the method of the cumulative distribution. Each input is fed to a net of this kind, that, given sufficient hidden neurons H, can approximate with arbitrary precision any continuous function y = g(x).

The global neural network used by this method has the same general structure of that of figure 1 with the difference that here the neurons are replaced by networks like those of figure 3 and the training rules are obtained from (23) when using (24) as correcting functions. Working out all the derivatives one can obtain the 3 H learning rules that define

an unsupervised on-line algorithm that approximates the cumulative distributions of each pixel (20). We don't report exactly this result here but instead the numerically more stable set of rules

$$\Delta \alpha_{j} = \eta \frac{w_{j}^{2}h_{j}(1 - h_{j}) + w_{j}^{2}\alpha_{j}x_{i}h_{j}(1 - h_{j}) - 2w_{j}^{2}\alpha_{j}x_{i}h_{j}^{2}(1 - h_{j})}{\sum_{j=1}^{H}w_{j}^{2}\alpha_{j}h_{j}(1 - h_{j})}$$

$$\Delta \beta_{j} = \eta \frac{w_{j}^{2}\alpha_{j}h_{j}(1 - h_{j}) - 2w_{j}^{2}\alpha_{j}h_{j}^{2}(1 - h_{j})}{\sum_{j=1}^{H}w_{j}^{2}\alpha_{j}h_{j}(1 - h_{j})}$$

$$\Delta w_{j} = \eta \frac{2w_{j}\alpha_{j}h_{j}(1 - h_{j})}{\sum_{i=1}^{H}w_{j}^{2}\alpha_{j}h_{j}(1 - h_{j})} + 2\eta w_{j}q \left(\sum_{j=1}^{H}w_{j}^{2} - 1\right)$$

obtained replacing w_j^2 to w_j in (24) and adding a constraint term $\sum\limits_{j=1}^H w_j^2 = 1$ that has the effects of maintaining w_j positive and of maintaining more easily the outputs y_i in the assigned range [0,1].

Our numerical tests indicate that 2 or 3 hidden neurons are sufficient to build good enough approximations of several of the standard distributions. In our particular case one hidden neuron is already sufficient.

5. Numerical results

To verify numerically these algorithms we introduce two measures of the quality of the corrected images. For synthetic images, beyond the detected image \vec{x} , one also has the true image $\vec{\phi}$ and the value of the true detector coefficients (1), thus one has also the true solution and can calculate the angle ψ between the solution found by our methods and the true solution space (5). This angle is an excellent measure of the quality of the solution.

When considering real images one does not have any more the solutions space (5) and must resort to some other method to measure the quality of the found solution. Since a successful correction vector must reproduce uniformly flat regions of the image, we estimate the uniformity of regions of the corrected image that correspond to regions known to be uniform. To do so, following [Scribner 1993], we introduce an estimate of the signal to noise ratio

(26)
$$SNR = \frac{y_{ave}^1 - y_{ave}^2}{\sqrt{\sigma_{y^1}^2 + \sigma_{y^2}^2}}$$

where y_{ave}^i are the average values of flat zones of the image and $\sigma_{v^i}^2$ their variances.

Let us examine first the case of a computer generated image chosen to be as similar as possible to a real one produced in SYRMEP: it is 250 times 48 pixels wide with 1024 gray

levels, the coefficients g_i and o_i were generated randomly with uniform distribution in the intervals [0.75, 1.25] and [0.15, 0.25] respectively and the image had values in [0, 1] to which a Poissonian noise was added.

The first learning algorithm (19) had $\eta = 5 \ 10^{-7}$, $\beta = 20,000$, and was run for 5,000 epochs. Figure 4 shows its convergence property towards solution space.



Fig. 4: Convergence of the solution vector, on the abscissa an epoch corresponds to a number of elementary learning steps (19) equivalent to the total number of pixels of the image. ψ is the angle between the solution found by this method and the space of exact solutions (5). The lower curve shows poorer convergence of the algorithm of footnote 4.

The second learning algorithm (25) had $\eta = 0.01$, one hidden neuron, and was run for 100 epochs. In figure 5 there are in sequence from the top: the "true" image ϕ , the unretouched detected image and the corrected images: the first after algorithm (19) has been applied, the second one corrected with the cumulative algorithm (25). Near the images there are the cross sections at pixel 138, indicated in the true image. The values of SNR, calculated using (26), are also reported.



Fig. 5: Sequence of synthetic images, from top: "true" image ϕ , as detected x and corrected (y) with first (19) and second (25) adaptive methods. Histograms at the right are the cross sections at pixel 138 whose position is marked on the first image; vertical scale is arbitrary.

The algorithms were also run on a real SYRMEP image and the results are in figure 6 that has the same structure of figure 5. We remark that in this case the "true" image isn't available and we replaced it with an image corrected with the, traditional, two points method.

In this case the parameters of the two algorithms were $\eta = 3.3 \ 10^{-7}$, $\beta = 30,000$, run for 5,000 epochs and $\eta = 0.03$, one hidden neuron, run for 300 epochs respectively.



Fig. 6: Sequence of SYRMEP images, from top: image corrected with the traditional two points method, detected image x and corrected (y) with first (19) and second adaptive methods (25). The histograms at the right are the cross sections at pixel 83.

From these data we notice that both methods are able to equalize the images without any prior calibration and that the quality obtained is high and surely comparable to that of standard approaches. In the near future we plan to investigate more carefully the numerical properties of these methods and to compare their relative merits in real world problems. In our particular application the method of the cumulative distribution is not favored because it introduces a non-linear correction function.

In conclusion we presented two adaptive algorithms dedicated to equalization of pixel response based on neural networks. Both methods share a sound theoretical foundation and provide good numerical results. While the first one produces a linear correction to the raw data estimating the expected output, the second produces a non-linear correction similar to contrast equalization. Both methods, one way or another, employ entropy maximization of the outputs.

References:

[Arfelli 1996]	F. Arfelli, G. Barbiellini, V. Bonvicini, G. Cantatore, E. Castelli, M. Di Michiel, R. Longo, A. Olivo, P. Poropat, M. Prest, F. Tomasini, G. Tromba, A. Vacchi and N. Zampa, <i>New Developments in the Field of</i> <i>Silicon Detectors for Digital Radiology</i> , Nuclear Instruments and Methods in Physics Research A 377 , 1996, 508-513;
[Atick 1992]	J.J. Atick, <i>Could Information Theory Provide an Ecological Theory of Sensory Processing ?</i> , Network 3 (1992) pp. 213-251;
[Bell 1995]	A.J. Bell and T.J. Sejnowski, <i>An Information-Maximization Approach to Blind Separation and Blind Deconvolution</i> , Neural Computation 7 , 1995, 1129-1159;
[Besag 1974]	J. Besag, <i>Spatial Interaction and the Statistical Analysis of Lattice Systems</i> , Journal of the Royal Statistical. Society, series B 36 , 1974, 192-326;
[Bolduc 1996]	P. Bolduc, P. Chevrette, J. Fortin and A. Zaccarin, <i>Enhancement of point-source targets in an IR staring FPA sensor</i> , presented at SPIE 96 conference on Infrared Imaging Systems;
[Field 1994]	D.J. Field, <i>What is the Goal of Sensory Coding</i> ? , Neural Computation 6 , pp. 559-601, 1994;
[Frison 1997]	R. Frison, <i>Reti Neurali per la Correzione Adattativa di Nonuniformità nei Rivelatori a Matrice</i> , degree thesis in Physics, University of Trieste (Italy), 1997;
[Geman 1984]	S. Geman and D. Geman, <i>Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images</i> , IEEE Transactions on PAMI 6 , 1984, 721-741;
[Hertz 1992]	J. Hertz, A. Krogh, R.G. Palmer, <i>Introduction to the Theory of Neural Computation,</i> Addison-Wesley Publishing Company, 1992;
[Hornik 1991]	K. Hornik, <i>Approximation Capabilities of Multilayer Feedforward Networks,</i> Neural Networks 4 (1991) pp. 251-257;
[Laughlin 1987]	S.B. Laughlin, <i>Form and Function in Retinal Processing</i> , TINS, Vol. 10 , No. 11 (1987) pp. 478-483;
[Ljung 1977]	Ljung L., <i>Analysis of Recursive Stochastic Algorithms</i> , IEEE Transactions on Automatic Control, Vol. AC-22 , No. 4, 1977;
[Milotti 1995]	E. Milotti, <i>Linear Processes that Produce 1/f or Flicker Noise</i> , Physical Review E 51 (4), 1995, pp.3087-3103;
[Rangarajan 1995]	A. Rangarajan, R. Chellappa, <i>Markov Random Field Models in Image Processing</i> , The Handbook of Brain Theory and Neural Networks, The MIT Press, pp. 564-567, 1995;
[Scribner 1991]	D.A. Scribner, J.T. Caulfield, K.A. Sarkady, and M.R. Kruer, <i>Adaptive Non uniformity Correction for IR Focal Plane Arrays Using Neural Networks</i> , SPIE Vol. 1541, July 1991;
[Scribner 1993]	D.A. Scribner, K.A. Sarkady, M.R. Kruer, Caulfield M.R., Hunt J.D., Colbert M. and Descour M., <i>Adaptive Retina-Like Preprocessing for</i> <i>Imaging Detector Arrays</i> , Proceedings of the 1993 IEEE International Conference on Neural Networks, San Francisco (CA), March 28 - April 1, 3 Volumes, IEEE Service Center, Piscataway NJ, U.S.A., vol. III p. 1955;
[Yuille 1995]	A.L. Yuille, S.M. Smirnakis and L. Xu, <i>Bayesian Self-Organization Driven by Prior Probability Distributions</i> , Neural Computation 7 (1995) pp. 580-593.