Color Mapping Using Neural Networks

D. Adkins
Data Card Corporation, Minneapolis, Minnesota

V. S. Cherkassky and E. S. Olson
University of Minnesota, Minneapolis, Minnesota

Abstract

Color devices such as scanners, printers and CRTs have device specific coordinate systems. It is desirable to be able to produce mappings between the coordinate system of a particular device and a device independent coordinate system that closely approximates human perception. In this way, color coordinates can easily be specified, transformed, or transported between various input and output devices. Mappings between color coordinate spaces can be achieved by function restoration, when a number of input-output samples of the mapping are available. Feed-forward multi-layer neural network have been shown to be able to perform non-linear non-parametric functional restoration, as is the case with color coordinate mapping. This type of network was used to map the Lab coordinate space onto the RGB coordinate space of an actual and a computer modeled dye sublimation printer. A neuron activation function, is introduced herein, which has advantages that would be useful to function restoration problems such as color mapping. The effectiveness of this model is tested by observing the error between the model's prediction and the ideal correct output on a number of known samples.

I. Introduction

Accurate reproduction of color is often necessary in many applications. For example, companies who use colored logos wish to have them reproduced consistently, so that they become recognized for that color. Incorrect reproduction in such a case is easily noticeable and may cause the viewer to associate a lack of quality with that company. There are many other cases where accurate color restoration is desirable.

If we are able to specify a color in a uniform coordinate system that closely represents human perception, such as the Lab coordinate system, it would useful to have a transformation to convert these coordinates into the CMY or RGB coordinates. The CMY coordinates produce should be the correct amount of dye to recreate the color specified. We want to determine a set of functions, as presented in equations (1) to (3) that will perform this task. Linear equations are often used, but they do not offer the accuracy required. The task of producing a function that will meet these criteria can be considered as a function restoration problem.

$$c = g_c(L,a,b) \tag{1}$$

$$m = g_m(L,a,b) \tag{2}$$

$$y = g_{y(}L,a,b) \tag{3}$$

This task could be performed using a parametric method, by attempting to discover the true functional form of the transformation. Because the physics of each individual device is different, many different models would need to be created. For each device, the specific parameters of the model would need to be tuned. This method is not viable due to the extreme complexity of color devices. Color devices such as the dye sublimation color printer, which we will be considering, are extremely non-linear. This type of printer, for instance, is prone to dye bleedback, and nonlinearities in the print head. A non-parametric approach would be preferable.

A neural network approach could be used to determine these functions non-parametrically. The input to the network would be the device independent color coordinates specified in the Lab system. The output of the network are the device dependent coordinates of the printer. The output would specify the amount of dye that the network believes would most closely reproduce the color specified by the Lab coordinates.

II. Problem Development

If you are given an unknown function $g(\mathbf{x})$, which you are required to approximate, and you are able to obtain input-output samples to this function for a variety of different $\mathbf{x}$ values, how can you use this information to determine the functions value between the known examples? This is the task that we have in equation (4).

$$h = g(\mathbf{x}); (h_i, \mathbf{x}_i = 1,...,I \tag{4}$$

Determining the value of $g(\mathbf{x})$ between samples can be approximated by interpolation. If $\mathbf{x}$ is one dimensional, linear interpolation can be used to approximate the function in between the samples. In the case of color restoration, $\mathbf{x}$ is three dimensional. Linear interpolation in higher dimension is not a straightforward approach for the following reason: in a higher dimension the number of nearest samples would over-determine a linear surface. For example, if $\mathbf{x}$ is three dimensional and the samples are equally distributed in a cubic fashion, there would be eight nearest samples. Only four points would be required to determine a linear surface. A piecewise linear interpolation could be used but this would utilize

less information than is contained in all eight nearest samples. As well, there would be several choices of samples to use in the interpolation, each giving a different result. A non-linear interpolation can be created by solving Laplace's equation, where the boundary conditions are set to correspond to the values of the eight nearest samples. This interpolation function requires that the samples be equally distributed. Laplace's equations is shown for three dimensions in equation (5).

$$d_{xx}V + d_{yy}V + d_{zz}V = 0 \quad (5)$$

The boundary conditions in the color mapping problem would be of the form shown in equations (6) through (13).

$$V(0,0,0) = h_1 \quad (6)$$
$$V(0,0,1) = h_2 \quad (7)$$
$$V(0,1,0) = h_3 \quad (8)$$
$$V(0,1,1) = h_4 \quad (9)$$
$$V(1,0,0) = h_5 \quad (10)$$
$$V(1,0,1) = h_6 \quad (11)$$
$$V(1,1,0) = h_7 \quad (12)$$
$$V(1,1,1) = h_8 \quad (13)$$

The h's are the sample outputs as stated in equation (4). A general solution that can satisfy the boundary conditions of equations (6) through (13) is given in equation (14).

$$V(x,y,z) = C_1xyz + C_2xy + C_3yz + C_4xz - C_5x - C_6y - C_7z + C_8 \quad (14)$$

Solving for the particular solution of equation (14) yields equation (15).

$$V(x,y,z;h_1...h_8) = C_1xyz + C_2xy + C_3yz + C_4xz - C_5x - C_6y - C_7z + C_8 \quad (15)$$

The constants of equation (15) based on the boundary conditions are given in equations (16) though (23).

$$C_8 = h_1 \quad (16)$$
$$C_7 = h_2 - h_1 \quad (17)$$
$$C_6 = h_3 - h_1 \quad (18)$$
$$C_5 = h_5 - h_1 \quad (19)$$
$$C_4 = h_6 - h_5 - h_2 + h_1 \quad (20)$$
$$C_3 = h_4 - h_3 - h_2 + h_1 \quad (21)$$
$$C_2 = h_7 - h_5 - h_3 + h_1 \quad (22)$$
$$C_1 = h_8 - h_7 - h_6 + h_5 - h_4 + h_3 + h_2 - h_1 \quad (23)$$

I will refer to equation (15) as the Laplace interpolation function. The use of the Laplace interpolation function is only valid within the boundaries determined by the nearest samples. In this derivation the bounds would be for values of **x** that are within the unit cube. Laplace's equation has certain properties which motivate us to use it as a interpolation function. The value of V at a particular point is the average of the points that surround it. The function V has no local maxima or minima. The surface area of V is minimized while still being constrained to the boundary conditions. This interpolation is analogous to stretching a rubber sheet between a set of points. [1]

There is a further advantage to using the Laplace interpolation function. This advantage is that a basis kernel function can be constructed from the Laplace interpolation function for use in a neural network. The three dimensional Laplace interpolation function can be decomposed into eight functions, one for each of the samples. This can be seen by observing equation (24).

$$h_1V(x,y,z;1,0,...,0) + h_2V(x,y,z;0,1,...,0) + ... + h_8V(x,y,z;0,0,...,1) = V(x,y,z;h_1...h_8) \quad (24)$$

The h's of equation (24) which are the sample outputs can be regarded as weights to a neural network. Proceeding further one can construct an activation function for the neurons of a neural network that utilizes this decomposition. The activation function is a form of radial basis function. Of ten times Gaussian radial basis functions are used, but they tend to interpolate with a great deal of oscillation.[2] The output of the ith neuron in the hidden layer of a multi-layer network would given by equation (25).

if($x_{i+1} > x > x_{i-1}$ and $y_{i+1} > y > y_{i-1}$ and $z_{i+1} > z > z_{i-1}$)

$$o_i = V\left[\frac{|x - x_i|}{c}, \frac{|y - y_i|}{c}, \frac{|z - z_i|}{c}\right]$$

else

$$o_i = 0 \quad (25)$$

The value of $\mathbf{x_i}$ in equation (25) is the center of the basis function for the ith neuron. It is also the location of the ith sample. Because the Laplace interpolation function has been derived for samples that are distributed with a distance of one, the value of c is used to scale the basis func-tion width for other distributions. The scaling constant c of equation (25) must be set so that the value of o_i goes to zero when any of the resulting coordinates of $\mathbf{x} - \mathbf{x_i}$ is such that is greater than or equal to the distance between basis centers. The value of **x** is the value for which we want to determine the functions response. By summing the activities of the all the neurons according to the equation (26),

$$O_{net} = \sum_{i=0}^{1} h_i o_i \quad (26)$$

the approximated value of equation (4) can be determined. Only the neurons, whose basis centers correspond to the nearest samples will contribute. All others will have zero value. A further benefit of using a network of basis functions of this type, is that only the neurons of the nearest neighbors to the input value need be calculated. This fact means that the processing time for calculating the approximation, only increases linearly as a function of the number of neurons per dimension. When the network is implemented on a non-parallel computer, this method can save computational costs.

This method can be viewed as a look-up table with polynomial interpolation between entries.

There are other difficulties in the color mapping problem that must be dealt with. This method would not be able to predict dye concentration for colors outside the printers gamut. To deal with this situation, Lab coordinates should be tested to determine if they are within the gamut of the printer. If not, these Lab coordinates should be mapped onto the closest Lab coordinate that are within the printer's gamut. The closest Lab values can be used for calculation of dye concentrations. The gamut can be determined from samples that are on the boundary of the printer coordinate space. Because dye concentration are specified in the range 0 to 1, the required samples would be those corresponding to coordinates on the surface of the unit cube. The corresponding Lab coordinates to these samples would determine a volume in the Lab coordinate space. The gamut of the printer specified in Lab space would no longer by cubic, but would be a highly contorted shape. Decision boundaries can be formed by interpolating between the samples that are known to be on the boundary of the gamut. The decision boundaries could be learned by neural network classifier.

III. Neural Network Implementation

The method previously described can be interpreted as a neural network. The architecture that represents this method is shown in figure 1. The intended inputs to the network are the Lab coordinates, and the intended outputs are the CMY or RGB coordinates. The inputs and outputs are shown on the figure 1, which also indicates the unidirectional flow of information through the network. The circles represent neurons, and the lines represent connections. The neurons in each layer operate differently. The input neurons do not provide any processing, but simply pass the value with high fan-out. The middle or hidden layer neurons incorporate a transfer functions that implements the Laplace interpolation function as described previously in equation (25). The output neurons calculate a weighted sum of the inputs they receive. These weights are associated with the connections, and are determined from the available samples. If the samples are distributed to correspond to the centers of the basis function for each neuron, the weights can be set immediately to the values of the sample outputs. If not, a set of weights can be determined iteratively. The connections of the input to middle layer have weights that are permanently set to one. The network would compute as in equation (27), with the h's of equation (26) being replaced with w's to denote that the weights may be determined iteratively from the samples.

$$O_{net} = \sum_{i=0}^{I} w_i o_i \qquad (27)$$

If only unequally distributed samples can be obtained, a network of this type can be trained with an iterative gradient descent method, to obtain the weights. The method, which was used is shown in equation (28).

$$w_i(k+1) = w_i(k) + \beta(k)[h_k - O_k(\mathbf{x})]K(\mathbf{x}, \mathbf{x}_i) \text{ for all } i \qquad (28)$$

It utilizes a factor $\beta(k)$ that is decreasing function that regulates the learning rate and causes that final set of weights to "settle down" toward the end of the training process. The term $[h_k - O_k(\mathbf{x})]$, adjusts the weights according to the amount of error observed. The term h_k denotes the sample output, and $O_k(\mathbf{x})$ denotes the current network prediction. The kernel function $K(\mathbf{x}, \mathbf{x_k})$ alters the weights in inverse proportion to the distance from the sample position $\mathbf{x}$ to $\mathbf{x}_i$, the center position associated with each weight or neuron. This kernel term we used causes only the weights for the nearest eight neurons to be altered.

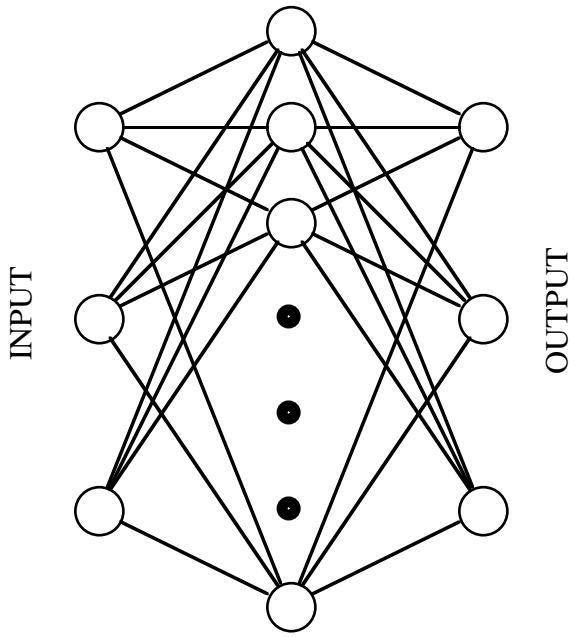

Figure 1. Feedforward Neural Network

Its value is one for weight-example distances of zero, and decreases to zero for increasing distance. The same functional form as equation (25) above, was used for the kernel function. Many different kernel functions could be used instead of this one. Indeed other neighborhood functions may be preferable.

VI. Results

This network was trained to approximate the mapping from the three Lab coordinates to the three RGB coordinates. The network contained six neurons per dimension. 192 samples were used. The examples were gathered by printing the samples equally distributed in the printer coordinate space. Their spectral response was measured with a spectrophotometer and the Lab values were then calculated . As stated before it would have been preferable to have examples equally distributed in the input space of the Lab coordinate system; however, this would require an *a priori* knowledge of the mapping.

The network was tested by producing 96 randomly generated colors in the printer coordinate space. The Lab coordinates for this test set were also determined. The network was given these Lab values for prediction of RGB values. The rms error for the network was found to be .035, .045, and .055 for each of the RGB coordinates respectively after 24000 iterations of learning. It should be noted that because the test sample set was taken from the printer coordinate space, the resulting Lab coordinates were guaranteed to be with in the gamut of the color printer during testing.

The network was also trained with computer generated examples, rather than real measured examples. The color printing process was simulated by the approximate equations shown as equation (29).[3]

$$C(\lambda) = W(\lambda)P(\lambda)Cyan_o(\lambda)^c Magenta_o(\lambda)^m Yellow_o(\lambda)^y \quad (29)$$

$W(\lambda)$ stands for the spectra of the white light. $P(\lambda)$ stands for the filter function of the paper. Likewise, $Cyan_o(\lambda)^c$ stands for the filter function of the Cyan dye as a function of wavelength and concentration. The concentration is specified by the lower case letter in the exponent. The exponents range form zero for no dye to one for maximum concentration. $Cyan_o(\lambda)$ represents the transmission at maximum concentration for the cyan dye. The functions are represented the same way for the Magenta and Yellow dyes. Much lower error rates were achieved when the number of neurons per dimension was increased. In this mapping, network sizes up to 40 neurons per dimension were used. The samples were continuously randomly generated. It is impractical to use larger networks for actual mappings of real samples due to the fact that increased quantities of samples would be necessary and data collection of samples is a time consuming process. The error achieved corresponded to an average of about .004 for all dyes. This error is close to being indistinguishable to humans.

V. Conclusions and Future Work

It is our intention to further this research by also producing mappings between the color scanner and the Lab coordinate system. This should provide a complete system which can be tested by observing the difference between scanned pictures and the corresponding printed pictures predicted by the chain of networks. Adding the scanner to the system, will allow a determination of how well colors that are out of the printer gamut can be reproduced. A method of determining the decision boundaries for the printer gamut will have to be chosen and implemented. It is also our intention to use other methods, (iterative and otherwise) to learn the network weights, to determine which will provide the most accurate set of weights.

Comparisons should be made between this method and others available methods of color restoration. These comparisons should include computation costs and the level of accuracy achieved by each method. Comparisons should be made with linear methods, other neural network methods, and conventional statistical methods.

Acknowledgments

The authors would also like to acknowledge DataCard Corporation whose grant has made this research possible.

References

1. Griffiths, D. J., *Introduction to Electrodynamics*, Prentice Hall, Englewood Cliffs, New Jersey, 1989, pp. 112-115.
2. Y. F. Wong, How Radial Basis Functions Work, in *Intl. Joint Conf. Neural Networks*, Seattle, July 1992, **2**: pp. 133-138.
3. H. J. Trussell, DSP Solutions Run the Gamut for Color Systems, *IEEE Signal Processing Magazine*, April, pp. 8-23 (1993).