Artificial Neural Networks and Colour Recipe Prediction

Stephen Westland
Colour & Imaging Institute, Kingsway House, Derby University, DE22 3HL

INTRODUCTION

Artificial Neural Networks (ANNs) represent a style of computing that is motivated by an understanding of human neural information processes. The relatively small size of most practical ANNs and the simplicity of their design, however, prohibit their use as plausible models of the human neural system in all but a few specific situations (see [1] for a review). The human brain consists of the order of 10^{12} neurons. In addition, the architecture of the neural structure is complex and there are believed to be many different types of neuron, each having its own characteristic behaviour [1,2].

Most ANNs are based on a few hundred units at most, the behaviour of the units is simple, and the units are usually arranged in several layers. Despite the apparent simplicity of ANNs they have been shown to be a powerful tool for problem solving [3-5]. An ANN is not explicitly programmed to complete a given task; rather, it adapts and acquires knowledge over time in order to complete it.

Previous work has shown that ANNs in general, and multi-layer perceptrons in particular, may be able to solve computer colorant formulation (also known as recipe prediction) problems [6-9]. Colorant formulation is an important procedure in manufacturing industry and concerns the prediction of a product formulation to meet a desired colour specification. The formulation is normally expressed in terms of which colorants (specific dyes or pigments) to use and at what concentrations.

Traditional approaches to computer colorant formulation have aimed to develop a parametric model of the relationship between the chemical formulation of a product and its colour specification [10,11]. Specifically, a traditional colorant formulation system relies upon being able to predict the reflectance R at a given wavelength λ for a mixture of known colorants. The Kubelka-Munk (K-M) theory [11] and variants of it have been used almost exclusively for this purpose. The K-M theory characterises colorants in terms of radiation absorption and scattering coefficients derived from a small number of calibration samples and has been very successful in providing effective solutions for industry. There are a sizeable number of situations, however, where applications of the K-M theory do not work well at all and it is suggested that there are three principle reasons for failure:

I. The K-M model may not adequately describe the interaction of a colorant with light, either individually or in mixture with other colorants.

II. The K-M model requires the preparation of special samples for use as a calibration database. For many coloration processes these samples are prepared in a laboratory and there are often systematic and random differences between laboratory and bulk preparations.

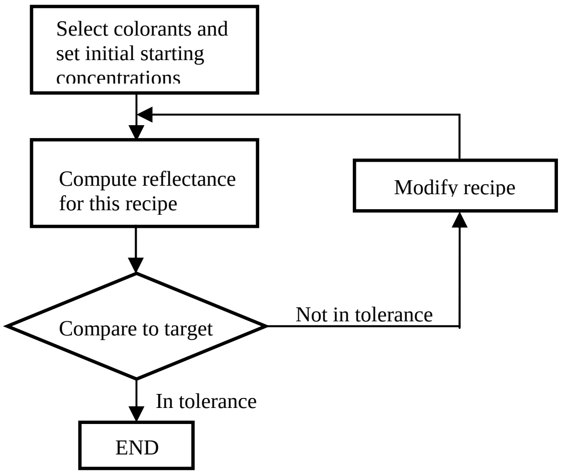
III. The coloration system may be poorly controlled.

Type I errors are fundamental problems with the K-M theory whereas type II errors are problems of application of the theory. Type III errors are caused by problems of inconsistency in the coloration and/or measurement process. Only type I errors indicate a break-down of the K-M theory but nevertheless it is suggested that the types II and III account for a large proportion of application failures. Colorant formulation systems using ANNs to replace the K-M equations promise to alleviate type I and II errors but cannot be expected to be of assistance if the coloration system is not reproducible (type III).

COLORANT PREDICTION USING ANNs

It is important to note that the K-M theory predicts reflectance from colorant concentrations and yet in practice it is exactly the inverse relationship that is required. That is, it is necessary to predict the concentrations of colorants for a specified target reflectance spectrum. Computer colorant formulation systems are therefore typically constructed according to the design illustrated by Figure 1.

Figure 1: Simple algorithm for colorant prediction to illustrate the iterative use of a colour prediction module.


Select colorants and set initial starting concentrations
Compute reflectance for this recipe
Modify recipe
Compare to target
Not in tolerance
In tolerance
END

It is important to distinguish between the two processes that we term *colour prediction* and *colorant prediction*. Colour prediction is the prediction of spectral reflectance (and ultimately colour) from a given recipe of colorants. Colorant prediction (or recipe prediction) is the inverse process; given a target reflectance spectrum a recipe of colorants is predicted. It is not easy to solve the inverse problem in the general case. Since the K-M theory is a colour prediction system rather than a colorant prediction system it is necessary to use an iterative procedure (as illustrated by Fig. 1) in order to achieve the process of colorant prediction. In general, ANNs are also suitable only for colour prediction. However, some previous studies [6,7] have used ANNs to predict a recipe formulation from reflectance of tristimulus values. This approach is appealing since the trained network will produce a recipe directly without the need for iteration. Such networks will only ever produce a single formulation for a given target vector since the network is entirely deterministic in its feed-forward operation. This may not be serious if it could be argued that the network could produce the optimum formulation. However, in general, if the training set contains many (or even two) distinctly different output vectors (recipes) for the same input vector then it is unlikely that the trained network would generate either of these output vectors when presented with the input vector. The problem is that the training set contains one ⇒ many mappings.

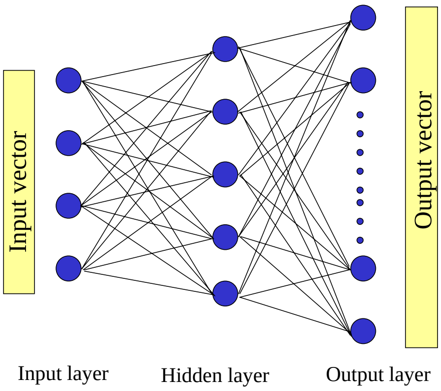
For a simple three- or four-colorant system the reflectance ⇒ formulation mapping may be the most efficient way to proceed since one ⇒ many mappings might not occur. Even for larger systems this approach may have some merit if the training data are sufficiently constrained, but in general the use of formulation ⇒ reflectance transformations (colour prediction rather than colorant prediction) offers a more robust solution to the problem.

NEURAL NETWORK DESIGN

There are many different types of ANN but this work is based upon multi-layer perceptrons (MLP) [3,4]. MLPs consist of layers of simple processing units that process information from the input layer to the output layer. The units in each layer send information to the units in the next layer via weighted connections. Units compute a weighted sum on their received input compute and output based on some transfer function on their input. Such networks can effect mappings from input vectors (presented to the first layer of the network to output vectors (the output of the final or output layer of units). For colour prediction, each unit in the input layer represents the concentration of a particular colorant in a formulation vector. Clearly if the number of available colorants in the system is large, but most formulations contain only three colorants, then the majority of the input units will have zero for most formulation vectors. The number of output units is usually 16 or 31, depending on the sampling interval of the available reflectance data, so that each unit in the output layer represents the reflectance at a particular wavelength. There are two reasons why it is probably better to predict reflectance, rather than tristimulus values, at the output. Firstly, it is not necessary to predict XYZ directly since XYZ can easily be calculated from reflectance. If we predict XYZ we are requiring the network to accomplish more work than is necessary. Secondly, a tristimulus specification is illuminant specific, whereas if the reflectance is known it is possible to compute the tristimulus specification for any illuminant.

The neural network also contains a hidden layer of units. It has been shown that an MLP with a single hidden layer of units is able to compute any computable transformation to any degree of accuracy [5]. Although more than one hidden layers of units is sometimes employed, for practical reasons the work reported here used only a single hidden layer. This general scheme is represented by Figure 2.

Figure 2: Schematic diagram to show MLP structure. The input vector provides the input to units (shown as filled circles) in the input layer. Information is processed from left to right via weighted connections (shown as solid lines). The output of the units in the output layer is the output vector. Hidden and output units also receive weighted input from a bias unit (not illustrated) of fixed output 1. In the diagram only four output units are shown for clarity but if the desired output vector is a 31-dimensional or 16-dimensional reflectance spectrum then either 16 or 31 units are required respectively.



EXPERIMENTAL

A set of six basic lithographic printing inks were obtained (Coates) comprising: clear, white, black, cyan, magenta and yellow. Mixtures of these inks were prepared and printed onto Leneta opacity charts (Form N2A) using a K-bar automatic coating machine. The spectral reflectance factors, at 10nm intervals, of each printed sample were measured using a sphere-based reflectance spectrophotometer. The number of samples prepared was 163 and these were randomly separated into a training set (123 samples) and a validation set (40 samples).

In addition, for comparison purposes, a set of conventional calibration samples were prepared for use with the K-M theory. The details of the specific implementation of the theory used are not presented in this paper but the implementation was an absolute two-constant approach that is described elsewhere [12].

The 123 samples in the training set were used to train an MLP of the type illustrated in Fig. 2. The MPL had six input units, each corresponding to the concentrations of one of the six printing inks used in the mixtures. The output layer contained 31 units corresponding to reflectance at 400nm, 410nm, 430nm … etc. The network was trained using the back-propagation algorithm with momentum. The algorithm takes two parameters: the learning rate and the momentum term. A modified version of the algorithm was used whereby the learning rate linearly reduced during training from 0.1 to 0.001. The momentum term was fixed at 0.1. Networks were trained using 6, 7 and 8 units in the hidden layer and for a total of 10,000 epochs. One epoch is a term used to describe the case where each of the samples in the training set are presented to the network, the errors at the output layer computed, and small changes made to the weights in the network according to the back-propagation algorithm.

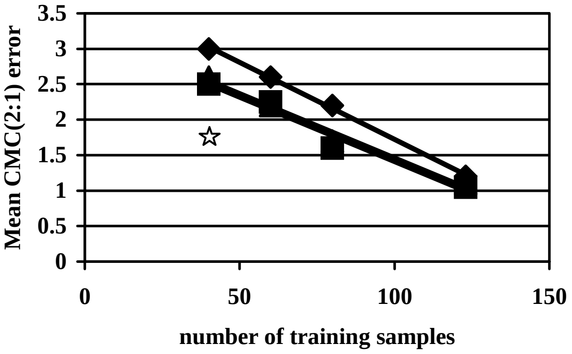
Networks were trained using the full 123 samples in the training set. However, other networks were trained using 80, 60 and 40 training samples by sub-sampling the training set. The purpose of this was to investigate the effect of number of samples in the training set on network performance.

RESULTS

The trained network was tested by computing colour differences between predicted reflectance spectra and actual reflectance for the 40 samples in the validation set. It is important to note that these samples were not used in the training of the network. Performance on these samples therefore represents a reasonable estimate of the generalization performance of the network. For each sample in the validation set the CMC(2:1) colour difference was computed and the mean colour difference across all samples was then computed. The performance of the neural networks was compared with that of the K-M prediction system in colour prediction mode for the same validation samples. Figure 3 summarises the results.

Figure 3 shows that there was little effect of the number of units in the hidden layer on network performance. The results also show that there is a steady improvement in the performance of the networks as the number of samples in the training set is increased. When the number of training examples was 123 the network performance gave a mean ∆E of about 1 unit whereas the K-M model gave a performance of about 1.5 mean ∆E.

Figure 3: Mean CMC colour difference errors for the neural networks and for the K-M model. The neural network data are shown for 6 (diamonds), 7 (squares) and 8 (triangles) hidden units. The solid lines show linear regression fits for each of the three sets of data. The performance of the K-M model is shown by a single star.



DISCUSSION

It has been shown that the performance of the trained network increases with increasing training. This is an expected result given that ANNs are closely related to statistical techniques. It has also been shown that, for this case study, the ANNs outperformed the performance of the K-M model. There are several caveats however. Firstly, it is clear that in order to outperform the K-M model the ANNs required more training samples. The K-M model required about 40 calibration samples. Secondly, there are many implementations of the K-M model and the one reported here is merely representative of a greater set of models.

There are a number of key points that make the ANN technique interesting and worthy of further study.

1. Although the ANN required more training samples than the K-M model, it is not clear that if the K-M model had access to further samples then further improvement would result. The K-M equations do not easily lend themselves to using additional samples whereas the ANN is highly suited to dealing with large numbers of training samples.

2. It is clear from Figure 3 that, in this study, had we had access to more than 123 training samples then further improvement (below 1.0 ΔE) would have been likely.

3. The x-axis on Figure 3 is not synonymous with work. The K-M model requires specially created calibration samples whereas the ANN can use (indeed would best use) historical production samples.

4. The use of production samples to train the ANN greatly reduces the chance of Type II errors (see earlier).

It has been shown that ANNs can effectively perform colour prediction. Study of Figure 1 shows that this will allow them to form the basis of a recipe prediction system since the K-M models in this paper and the ANNs both perform functionally identical colour prediction. In a modular recipe prediction system the K-M module for colour prediction can be replaced by a neural module.

REFERENCES

1. AJ Schofield, *PhD thesis*, Keele University (1993).

2. DH Hubel, *Eye, brain and vision*, Scientific American Library (1988).

3. I Aleksander and H Morton, *An introduction to neural computing*, Chapman & Hall (1992).

4. E Davalo and P Naïm, *Neural Networks* (1991).

5. K Funahashi, *Neural Networks*, **2** (3) (1989).

6. JM Bishop, MJ Bushnell and S Westland, *Color Research and Application*, **16** (1), 3-9 (1991).

7. T Tokunaga and Y Honda, *Kako Gijutsu* (Dyeing & Finishing Technology), **26** (8), 553-557 (1991).

8. S Westland, JM Bishop, MJ Bushnell and AL Usher, *J.S.D.C*, **107**, 235-237 (1991).

9. C de M Bezerra and CJ Hawkyard, *J.S.D.C*, **116** (5/6), 163 (2000).

10. RH Park and EI Stearns, *J. Opt. Soc. Am.*, **34** (2), 112 (1944).

11. JH Nobbs, *Rev. Pro. Coloration*, **15**, 66-75 (1986).

12. S Westland, *PhD Thesis*, Leeds University (1988).