Coactive neuro-fuzzy modelling for colour recipe prediction

Mizutani, Eiji
Informations Systems Dept., Kansai Paint Co., Ltd.

Jyh-Shing R. Jang
Department of Computer Science, National Tsing Hua University

Nishio, Kenichi
Video Technology Center, Sony Corp.

高木，英行
Department of Acoustic Design, Kyushu Institute of Design

他

https://hdl.handle.net/2324/4485874


バージョン:

権利関係:
Coactive Neuro-Fuzzy Modeling for Color Recipe Prediction

Eiji Mizutani 1, Jyh-Shing R. Jang 2, Kenichi Nishio 3
eiji@joho.kansai.co.jp, jang@cs.nthu.edu.tw, nishio@cv.sony.co.jp
Hideyuki Takagi 4, and David M. Auslander 5
takagi@kyushu-id.ac.jp, dma@euler.berkeley.edu

1) Information Systems Dept., Kansai Paint Co., Ltd, 4-3-6 Fushimi Chuo, Osaka 541 JAPAN
2) Dept. of Computer Science, National Tsing Hua University, Hsinchu, Taiwan
3) Video Technology Center, Sony Corp., 5-9-12 Kitashinagawa Shinagawa, Tokyo 141, JAPAN
4) Acoustic Design Dept., Kyushu Institute of Design, 4-9-1 Shiobaru, Minami-ku, Fukuoka 815, JAPAN
5) Dept. of Mechanical Engineering, University of California at Berkeley, CA 94720, USA

ABSTRACT

We explore neuro-fuzzy approaches to computerized color recipe prediction, which relates surface spectral reflectance of a target color to several colorant proportions. The approaches are expressed within the framework of CANDIS (CoActive Neuro-Fuzzy Inference System) where both Neural Networks (NNs) and Fuzzy Systems (FSs) play active roles together in pursuit of a given task.

To find an ideal adaptive model for this problem, we have investigated a variety of structures; they feature knowledge-embedded architectures and an adaptive FS, which serves to determine color selection. They have enormous potential for augmenting prediction capability.

1. Introduction

Colors enliven our daily lives. We often need a specific color exactly matched with our favorite color. Yet our color perception is very sensitive. Hence, in the color industry, it is important to predict the proportions of several colorants (or color pigments) needed to produce the same color as a given target color. The color matching must be done very well to satisfy severe requirements. In a practical situation, it is necessary to examine the color match in daylight as well as in artificial light. It is actually an arduous task. The prediction process involves choice of necessary colorants, and requires high precision of their proportions.

Simple NN approaches to color recipe prediction have been discussed well [1, 10, 8]. Under practical environments, we sometimes encounter severe standards, which may be hard to meet by employing an NN alone. In the past few years, Neuro-Fuzzy models have been much studied; the literature promises far wider industrial applications. We contend that another approach such as fuzzy modeling must complement simple NNs to enhance their overall performance.

2. CANFIS Concept

CANFIS has expanded the concept of a single-output system, ANFIS [5] (Adaptive Network-based Fuzzy Inference System), which realizes fuzzy inferencing in the network-layered representation. Figure 1 illustrates a comparison between the TSK fuzzy inference system [11] and a corresponding ANFIS to perform the following fuzzy rules (for further details, refer to [5, 6]):

Rule 1: If X is A1 and Y is B1, then C1 = p1X + q1Y + r1.
Rule 2: If X is A2 and Y is B2, then C2 = p2X + q2Y + r2.

CANFIS, a generalized ANFIS, can produce
multiple outputs and take advantage of non-linear rule formations [9]. Figure 2 describes a simple CANFIS model to produce multiple outputs. This model shares antecedent parts unlike MANFIS (Multiple ANFISs) model wherein several ANFISs are just juxtaposed.

2.1. Rule Formation

Suppose we have a sigmoidal function as a neuron function in the consequent layer such as Functions 3, 4, 5, and 6 in Figure 2. Then we have a nonlinear consequent, $C_{\text{non}}$:

$$C_{\text{non}} = \frac{1}{1 + \exp[-(p_1 X + q_1 Y + r_1)]}. \tag{1}$$

In this case, we have a sigmoidal rule.

Furthermore, when we introduce an NN in a rule construction, we have a rather complicated structure as illustrated in Figure 3 (above). Moreover, when such neural consequents are entwined; that is, when two neural consequents, "Neural Rule1" and "Neural Rule2" are combined to form one neural rule (Local Expert $NN_1$), and "Neural Rule3" and "Neural Rule4" are fused into another neural rule (Local Expert $NN_2$), we have a construction similar to a typical modular network as illustrated in Figure 3 (below) where the outputs of two local expert NNs are mediated by an integrating unit (typically a gating network) [4, 3, 7]. In another perspective, a task is split up among several neural rules or NNs. Thus, the entire model may be able to overcome the individual limitation of any one NN; so the average load of each NN may be reduced. Here the idea is task decomposition. (CANFIS with neural rules can be regarded as an introduction of fuzzy concept into a modular network.)

We shall show how CANFIS can be generalized for application to color recipe prediction in the next section.

3. CANFIS modeling for Color Recipe Prediction

The training data consist of 1,446 Munsell color chips and the checking data of 302 standard paint color chips of the Japan Paint Manufacturers Association. Both data sets were sampled from surface spectral reflectance at 16 points, ranging from...
400 nm to 700 nm in wavelength (20-nm intervals).
We have such 16 spectral inputs and 10 colorant
outputs. In our color recipe prediction, we must
select several color pigments out of the ten colorant
candidates, and determine their proportions.

3.1. Neuron Functions

In our data sets, the average number of color
pigments to produce any color was about four. This
means six of the ten final outputs should be zero.
In addition, we sometimes need to predict precise
proportions with enough precision to specify levels
such as 0.01%. It is an important concern in color
recipe prediction, we must select several color pigments out of the ten colorant candidates, and determine their proportions.

3.1. Neuron Functions

In our data sets, the average number of color
pigments to produce any color was about four. This
means six of the ten final outputs should be zero.
In addition, we sometimes need to predict precise
proportions with enough precision to specify levels
such as 0.01%. It is an important concern in color
recipe prediction, we must select several color pigments out of the ten colorant candidates, and determine their proportions.

3.1. Neuron Functions

In our data sets, the average number of color
pigments to produce any color was about four. This
means six of the ten final outputs should be zero.
In addition, we sometimes need to predict precise
proportions with enough precision to specify levels
such as 0.01%. It is an important concern in color
recipe prediction, we must select several color pigments out of the ten colorant candidates, and determine their proportions.

3.1. Neuron Functions

In our data sets, the average number of color
pigments to produce any color was about four. This
means six of the ten final outputs should be zero.
In addition, we sometimes need to predict precise
proportions with enough precision to specify levels
such as 0.01%. It is an important concern in color
recipe prediction, we must select several color pigments out of the ten colorant candidates, and determine their proportions.

3.1. Neuron Functions

In our data sets, the average number of color
pigments to produce any color was about four. This
means six of the ten final outputs should be zero.
In addition, we sometimes need to predict precise
proportions with enough precision to specify levels
such as 0.01%. It is an important concern in color
recipe prediction, we must select several color pigments out of the ten colorant candidates, and determine their proportions.

3.1. Neuron Functions

In our data sets, the average number of color
pigments to produce any color was about four. This
means six of the ten final outputs should be zero.
In addition, we sometimes need to predict precise
proportions with enough precision to specify levels
such as 0.01%. It is an important concern in color
recipe prediction, we must select several color pigments out of the ten colorant candidates, and determine their proportions.

3.1. Neuron Functions

In our data sets, the average number of color
pigments to produce any color was about four. This
means six of the ten final outputs should be zero.
In addition, we sometimes need to predict precise
proportions with enough precision to specify levels
such as 0.01%. It is an important concern in color
recipe prediction, we must select several color pigments out of the ten colorant candidates, and determine their proportions.

3.1. Neuron Functions

In our data sets, the average number of color
pigments to produce any color was about four. This
means six of the ten final outputs should be zero.
In addition, we sometimes need to predict precise
proportions with enough precision to specify levels
such as 0.01%. It is an important concern in color
recipe prediction, we must select several color pigments out of the ten colorant candidates, and determine their proportions.

3.1. Neuron Functions

In our data sets, the average number of color
pigments to produce any color was about four. This
means six of the ten final outputs should be zero.
In addition, we sometimes need to predict precise
proportions with enough precision to specify levels
such as 0.01%. It is an important concern in color
recipe prediction, we must select several color pigments out of the ten colorant candidates, and determine their proportions.

3.1. Neuron Functions

In our data sets, the average number of color
pigments to produce any color was about four. This
means six of the ten final outputs should be zero.
In addition, we sometimes need to predict precise
proportions with enough precision to specify levels
such as 0.01%. It is an important concern in color
recipe prediction, we must select several color pigments out of the ten colorant candidates, and determine their proportions.
3.2. Fuzzy Partitionings

In fuzzy modeling, it is of importance to determine a reasonable number of MFs to maintain appropriate linguistic meanings. In all simulation examples of ANFIS [5], MFs were set up for all inputs using grid partitions, but here is a question: is it necessary to set up MFs for all 16 spectral inputs? In such a case, we must have the following 16 fuzzy rules:

- Rule 1: If $X_1$ (at 400nm) is $A_1$, then $C_1$,
- Rule 2: If $X_2$ (at 420nm) is $A_2$, then $C_2$,
- ... etc,
- Rule 16: If $X_{16}$ (at 700nm) is $A_{16}$, then $C_{16}$.

where $A_i$ denotes fuzzy linguistic labels. (Note that the visible range of color spectrum is 400 nm - 700 nm.) These rules may not make sense since we do not have such explicit knowledge per wavelength. Without explicit domain-knowledge, adaptive learning mechanisms enable ANFIS / CANFIS to cause a buildup of fuzzy rules automatically. But if the initial setup of MFs has no meanings, it must be futile to extract fuzzy rules from a fuzzy logic point of view. Blindly applying fuzzy MFs to all scalar inputs may turn out to be meaningless.

To keep meaningful fuzzy rules, the number of MFs should be carefully determined so that those fuzzy rules can be specified. Fortunately, there is a formula to transform surface spectra of color to perceptual attributes, ‘lightness,’ ‘hue,’ and ‘chroma’ [12]. These three values must be more suitable for treating color in a linguistically meaningful way than the 16 spectral values. Hence, we use them as three inputs for MFs but not 16 inputs of surface spectral reflectance. When we invert the 3-D partitionings in the color-attribute space to the 16-D of spectral inputs, certain complicated partitions must be constructed in the 16-D space.

Notice that an NN with 16 spectral inputs did a better job than an NN with three inputs of ‘lightness,’ ‘hue,’ and ‘chroma.’ Thus, as illustrated in Figures 6 and 7, five color neural rules have 16 inputs (color spectra) whereas MFs have those three inputs (color attributes).

3.3. Knowledge-embedded structures

Adaptive fuzzy membership functions (MFs) specify the degree of membership of five color regions (red, yellow, green, blue, violet) according to perceptual attributes of color, ‘lightness,’ ‘hue,’ and ‘chroma’ [12]. They determine what weight should be assigned to each rule’s output in order to produce a final output. We have applied the colorist’s knowledge to the CANFIS architecture so that several connections can be pruned. For example, a green rule (weighted by a green MF) has no effect on the proportions of red pigments because of the green-red complementary color relationship. After the pruning process, each color rule has fewer output units than the ten final outputs; Figure 6 pictures this idea, which can be viewed as a variation of a modular network in Figure 3 (below). Again, in this experiment, six of the ten final outputs should be zero because the desired number of color pigments to produce any color is about four. Reducing the number of zero outputs...
Table 1: Detail descriptions of the four representative CANFIS models. The bell-shaped MFs and the modified bell MFs are defined in Equation (4) and Equation (5), respectively.

(a) CANFIS with 5 sigmoidal rules (see Equation (1)) with pruned connections
5 bell-shaped MFs are set up for hue angle alone (i.e., 5 rules are for five color regions)
(b) CANFIS with 45 linear rules as in Figure 7 with NO pruned connections
3 bell-shaped MFs are set up for lightness
3 bell-shaped MFs are set up for chroma
5 bell-shaped MFs are set up for hue angle
(c) CANFIS with 5 neural rules as in Figure 5(below) with NO pruned connections
5 modified bell MFs are set up for hue angle alone
5 neural rules have the same model size (i.e., Each neural rule has 22 hidden units)
(d) CANFIS with 5 neural rules as shown in Figure 6 with pruned connections
5 modified bell MFs are set up for hue angle alone
Each neural rule is heuristically optimized; i.e., their model sizes are specified in Table 2

Can possibly have a positive impact on the input-output mappings.

3.4. Fuzzy Membership Functions

Fuzzy membership functions play a role of an integrating unit; we consider the following bell-shaped MFs:

$$\mu(x) = \frac{1}{1 + \left| \frac{x-c}{a} \right|^{2b}}, \quad (4)$$

or

$$\mu_{\text{mod}}(x) = \max\left\{ \frac{2}{1 + \left| \frac{x-c}{a} \right|^{2b}} - 1, 0 \right\}, \quad (5)$$

where \(\{a,b,c\}\) is an adjustable parameter set. The latter definition is a modified bell MF, which is just upper half part of the original bell-shaped MF.

In structural terms of CANFIS with neural rules, it has many adjustable parameters compared with single NN models (such as \(NN_{\text{normal}}\) and \(NN_{\text{mod}}\)). To accelerate training, we can employ the modified bell MFs defined in Equation (5) to control the number of firing rules; the functions are instrumental for this color recipe prediction because it may not be necessary to use more than two color rules at the same time. More specifically, when the target color is in a region between green and yellow; that is, when both the “yellow rule” and the “green rule” are fired, the neighboring “red rule” and “blue rule,” are not necessarily fired because of the yellow-blue and green-red complementary color relationships. Several weight-updating procedures for unnecessary or inactive rules can then be skipped when iterative training procedures are employed.

Table 2: An optimal model structure of five color NNs and their initial number of training / test data. Note that each neural rule has fewer outputs than the ten final outputs.

<table>
<thead>
<tr>
<th>Five Color Neural Rule</th>
<th>Model Size</th>
<th>Training Data</th>
<th>Checking Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>RuleRed</td>
<td>16 x 16 x 16 x 8</td>
<td>650</td>
<td>38</td>
</tr>
<tr>
<td>RuleYellow</td>
<td>16 x 16 x 17 x 8</td>
<td>707</td>
<td>200</td>
</tr>
<tr>
<td>RuleGreen</td>
<td>16 x 21 x 7</td>
<td>521</td>
<td>105</td>
</tr>
<tr>
<td>RuleBlue</td>
<td>16 x 15 x 8</td>
<td>363</td>
<td>65</td>
</tr>
<tr>
<td>RuleRed</td>
<td>16 x 17 x 6</td>
<td>409</td>
<td>48</td>
</tr>
</tbody>
</table>

4. Simulation

We have tested various CANFIS models; because of the space limitation, we show the several representative CANFIS models as shown in Table 1. Table 3 shows performance comparison of those CANFIS models as well as single NN models: \(NN_{\text{normal}}\) and \(NN_{\text{mod}}\), \(NN_{\text{mod}}\) is an improved NN model with the modified sigmoidal function.

There must be many possible optimal rule formations; Table 2 shows one of them, which we found through trial and error implementation, using a model depicted in Figure 6. Each color neural rule (or local color NN) has a different model size because different number of data goes into each neural rule; those initial number of data is also shown in Table 2. Notice that each neural rule has fewer outputs than the ten final outputs because of the pruning procedure discussed in Section 3.3. By contrast, we also tested a CANFIS in which all five neural rules had the same model size.

Through the preliminary CANFIS modeling, we found that CANFIS with the truncation filter functions defined in Equation (2) at the fuzzy association layer worked better than CANFIS with the identity functions. We did not present any results about this, but the results were more or less the same as those of \(NN_{\text{normal}}\) and \(NN_{\text{mod}}\); \(NN_{\text{mod}}\) is an improved NN model with the modified sigmoidal function.

We used a priori knowledge to prune some connections as discussed in Section 3.3. When we employed such pruned structures, we did obtain at least a few percent improvement in terms of pigment errors. (Note that we did not present any comparisons about this finding in Table 3.)

5. Discussion

We have applied a variety of CANFIS structures as a touchstone to the color recipe prediction for discovering what CANFIS model belongs to the truly
Table: 3. Performance comparison between single NN models: $NN_{\text{normal}}$ and $NN_{\text{mod}}$, and four representative CANFIS models. Table 1 details the four CANFIS models.

<table>
<thead>
<tr>
<th>Model</th>
<th># of membership functions</th>
<th>Rule</th>
<th>Checking error ($10^{-2}$)</th>
<th>Specified # of pigments</th>
<th>Parameter number</th>
</tr>
</thead>
<tbody>
<tr>
<td>CANFIS (a)</td>
<td>hue lightness chroma</td>
<td>sigmoidal</td>
<td>7.99</td>
<td>3.46</td>
<td>852</td>
</tr>
<tr>
<td>CANFIS (b)</td>
<td>5 0 3</td>
<td>linear</td>
<td>2.59</td>
<td>3.76</td>
<td>7,683</td>
</tr>
<tr>
<td>CANFIS (c)</td>
<td>5 0 5</td>
<td>neural</td>
<td>1.90</td>
<td>3.85</td>
<td>3,035</td>
</tr>
<tr>
<td>CANFIS (d)</td>
<td>5 0 5</td>
<td>neural</td>
<td>1.41</td>
<td>4.00</td>
<td>2,691</td>
</tr>
<tr>
<td>$NN_{\text{normal}}$</td>
<td>- - -</td>
<td>1 neural</td>
<td>2.62</td>
<td>6.66</td>
<td>925</td>
</tr>
<tr>
<td>$NN_{\text{mod}}$</td>
<td>- - -</td>
<td>1 neural</td>
<td>2.03</td>
<td>3.90</td>
<td>925</td>
</tr>
</tbody>
</table>

excellent class; at this stage, the CANFIS model (d) presented the best performance as in Table 3.

CANFIS with 45 linear rules has much more modifiable parameters than other CANFIS models (see Table 3). Accordingly, it required much more computation time than other CANFIS models; so we did not successfully optimize parameter setups for CANFIS with 45 linear rules in heuristic ways. Because of its huge construction, it seems difficult to introduce neural rules (or local color expert NNs) to CANFIS with 11 MFs (or 45 rules); therefore, we did not test CANFIS with 45 neural rules. From a color science standpoint, however, it is a better idea to consider the three color attributes: ‘lightness,’ ‘hue,’ and ‘chroma,’ than to consider the ‘hue’ aspect alone. We thus must make an effort to upgrade such a CANFIS model with 11 MFs. At the same time, we should consider whether there is a more effective way to represent human visual sensitivity to color in the perceptual attribute space than the use of bell-shaped MFs. We may need to contrive a more sophisticated MF.

It was observed that modified bell MFs defined in Equation (5) were useful in accelerating learning because we used the iterative gradient decent (GD) backpropagation method. In the future work, we should apply to CANFIS models the previously-proposed hybrid learning algorithm, which is based on a combination of the GD method and the least-squares estimation method [5, 6].

6. Conclusion

We have demonstrated the strength of knowledge-embedded CANFIS models in the computer color recipe prediction.

By constructing MFs in the color-attribute space, the CANFIS models have realized meaningful and concise representations of colorists’ knowledge. These encouraging results confirm that CANFIS offers superior performance and better learning capabilities than single NN models. But the results are not conclusive in determining most appropriate CANFIS architectures; we need to investigate more experimentations.

To explore the discussed concerns must be our next step, which may endow a breakthrough in understanding neuro-fuzzy modeling. We believe such efforts could pave the way for a new generation of CANFIS.

References