Learning diagnostic features: The delta rule does Bubbles

Thomas Hannagan

Jonathan Grainger

It has been shown (Murray & Gold, 2004a) that the Bubbles paradigm for studying human perceptual identification can be formally analyzed and compared to reverse correlation methods when the underlying identification model is conceived as a linear amplifier (LAM). However the usefulness of a LAM for characterizing human perceptual identification mechanisms has subsequently been questioned (Gosselin & Schyns, 2004). In this article we show that a simple linear model that is formally analogous to the LAM—a linear perceptron trained with the delta rule—can make sense of several Bubbles experiments in the context of letter identification. Specifically, an analysis of input-output connection weights after training revealed that the most positive weights clustered around letter parts in a way that mimicked the diagnostic parts of letters revealed by the Bubbles technique (Fiset et al., 2008). Our results suggest that linear observer models are indeed unreasonably effective, at least as first approximations to human letter identification mechanisms.

Introduction

Common sense and empirical evidence suggest that certain features are more important than others when identifying objects such as single letters. The Bubbles technique (Gosselin & Schyns, 2001) was developed expressly in order to empirically determine the diagnostic features of visual stimuli and has been successfully applied to natural scenes (McCotter, Gosselin, Sowden, & Schyns, 2005), faces (Vinette et al., 2004), facial expressions of emotions (Smith, Cottrell, Gosselin, & Schyns, 2005), letters (Fiset et al., 2008; Fiset et al., 2009), and words (Blais et al., 2009). Accompanying such versatile applications, an informal attempt was made to couch human visual categorization processes within a general framework, the so-called RAP framework stating that “potent” information is a product of “represented” information (R) and “available” information (A) (Gosselin & Schyns, 2002). This effort was later carried out further and the Bubbles method criticized by Murray and Gold (2004a). Under certain assumptions, Murray and Gold were able to provide a formal description of how information is used in a Bubbles experiment that relates Bubbles not only to the RAP framework but also to previous categorization techniques (the “reverse correlation” method) and spells out possible biases and limits of Bubbles.

However this progress in formalizing the product of a Bubbles experiment and comparing it to reverse correlations was achieved at the cost of assuming that a simple linear model (hereafter LAM, for linear amplifying model) underlies the human ability for visual categorization. It was assumed, specifically, that subjects use a linear function to compare the representation of the input to pre-existing representations of the different alternatives allowed in the task. In its simplest form, a subject would essentially calculate the dot product between, for instance, a representation corresponding to the visual letter input, and all 26 “letter template” vectors previously formed by the subjects for uppercase letters in the alphabet. Such an assumption has been deemed unrealistic (Gosselin & Schyns, 2004), which would marginalize the relevance to bubbles of Murray and Gold’s analysis and criticism (Murray & Gold, 2004a, 2004b).

More specifically, Gosselin and Schyns (2004) claimed that a LAM is not sufficiently general in scope to impose any prescriptive standards on the conduct of research in visual categorization. In what follows we will show that on the contrary, a LAM has relevance for understanding data obtained using the Bubbles method on letter identification. The two observations on which we build our computational study are the linearity of a perceptron network and the diagnostic properties of the most widespread learning algorithm for perceptrons, the delta rule (Widrow & Hoff, 1960).
The LAM, perceptrons, and the delta rule

Using the notation introduced in Murray and Gold (2004a), a LAM (sometimes also referred to as template matcher, Murray, 2012) can be defined as a system making a decision on the nature of an input $I$, by linearly comparing it to existing templates $T_i$ under noisy conditions $Z$. In the case of a forced choice between $n$ alternatives, the decision function $R$ might be for instance:

$$R = \text{argmax}_{i=1..n}(I \otimes T_i + Z).$$

Here, $I$, $T_i$, and $Z$ are real matrices of equal dimensions, and $\otimes$ is any linear operator with a scalar output, for instance the cross-correlation operator as in Murray and Gold (2004a) or the convolution operator that multiplies two matrices element-wise and returns the sum of the elements.

When the convolution operator is used, this system is analogous to a standard perceptron network, the simplest neural network that can perform a direct mapping from input to output. In the network’s input layer, each unit stands for an element of the input matrix $I$, whereas in the output layer each unit stands for a possible decision. The two layers are linked with full feedforward connections (i.e., each unit in the input layer sends a connection to every unit in the output layer). In the case of 26 decisions $A, B, ..., Z$, each set of connections arriving at output unit $A, B, ..., Z$ is a matrix of same dimension as the input, respectively written as $T_A$, $T_B$, ..., $T_Z$.

The activation of output unit $i$ is given by a function of its net input $\text{net}_i = 1 \otimes T_i + s_i$, where $s_i$ is the “bias” weight. We will use a “saturated” linear activation function throughout this article: a linear function bounded below by zero and above by one. The weight matrices $T_A$, $T_B$, ..., $T_Z$ are initially set to zero. However under the influence of a learning algorithm and the repeated presentation of inputs $I_A$, $I_B$, ..., $I_Z$, the weights progressively come to embody the templates for $A$, $B$, ..., $Z$.

Possibly the simplest supervised associative learning algorithm and the one we will use in this article is the “delta rule,” also known as the Widrow-Hoff rule (Widrow & Hoff, 1960), an algorithm that has formal connections with conditioning models (Wagner & Rescorla, 1972) and that we will describe more formally in the next section. The delta rule has been applied to the computational modeling of several cognitive phenomena and processes, including for instance conditioning (Gluck & Bower, 1988), memory (McClelland & Rumelhart, 1985), language acquisition and processing (Baayen, Milin, Filipović Durdević, Hendrix, & Marelli, 2011), category learning, and associative learning (Kruschke & Bradley, 1995), but its use extends well outside of cognitive science whenever decision making is involved, notably in the medical field where it can be used to operate diagnostics (Brause et al., 2001).

The notion that the delta rule could help explain the diagnostic information used by human subjects in a Bubbles experiment is therefore well founded. In addition, the so-called perceptron convergence theorem (Nilsson, 1965) ensures that if the patterns on which the delta rule is applied are linearly separable, then in the case of a binary classification task the perceptron will converge to a solution. To our knowledge, however, a simple perceptron trained with the delta rule has not yet been applied to diagnostic visual feature extraction in general and certainly not to the specific case of letter identification.

The LAM and the perceptron are trivially analogous systems, when templates in the former are seen as connection sets in the latter. Then the linear convolution operator between inputs and templates in LAM exactly corresponds to the linear propagation of inputs along multiplicative weights in the perceptron (i.e., the net input computation). We will now show how the perceptron trained under the delta rule can account for behavioral data recently gathered using the Bubbles method on letter identification tasks.

Results

The Bubbles experiment of Fiset et al. (2008), which we simulate in this paper, was based on a letter identification task. It consisted in the short presentation (for 200 ms) of letter stimuli that were occluded by a uniformly gray mask except for small apertures at randomly chosen locations (bubbles) that could be of one of five different sizes. Each subject performed 100 blocks of 260 trials for each case. A strong appeal of both the Bubbles and reverse correlation paradigms is that they can produce, in a compelling visual output, “classification images” that arguably reveal the diagnostic features that subjects use for target identification (Ahumada, 1996). The signal detection analysis underlying these images is similar in each case and involves contrasting the average noise fields—or sums of bubble windows—presented to the subjects across different stimulus/response trials. Specifically, and to paraphrase Murray (2012), a reverse correlation classification image obtains by regressing all observers’ responses against additive Gaussian stimulus noise, whereas a Bubbles classification image is found by regressing correct observers’ responses against sparse multiplicative noise.

A key point is that when the underlying model is a linear amplifier, it can be shown that these classification images are either directly scaled versions of the
Figure 1. Diagnostic features for letter identification in humans (blue regions, data from Fiset et al., 2008) and in the model (red regions), for uppercase (upper panel) and lowercase letters (lower panel). The human diagnostic regions correspond to bubbles classification images, whereas the model diagnostic regions correspond to letter weights obtained after training with the delta rule.
templates, if one uses reverse correlation methods, or blurred versions of the template convolved with a particular difference of input signals, in the case of the Bubbles method (Murray & Gold, 2004a). Therefore in what follows we will start by directly comparing the templates obtained by training a perceptron network on a letter identification task to the classification images revealed in a bubbles experiment.

Simulation procedure

Two perceptron networks were trained to recognize either uppercase or lowercase letters. All computations were done in floating point and implemented in the Python language. Letter inputs were 26 black Arial letters centered on a white background, with dimensions 188 × 188 pixels for lowercase letters and 128 × 128 pixels for uppercase letters. The connection weights were initially set to zero and evolved as the networks were repeatedly and uniformly exposed to letters. One training epoch consisted of the presentation of all 26 letters exactly once. Each network was trained until a criterion for response accuracy was reached or 500 epochs had elapsed, and on each trial within an epoch, weight modification proceeded according to the delta learning rule described in the Appendix. Correct identification was granted whenever the maximally active unit in the output layer corresponded to the target letter. In both the uppercase and lowercase conditions, the network had reached 100% accuracy at the end of training. We note that for each condition, the total number of learning trials undergone by the network (500 × 26) was of the same magnitude as the number of trials completed by the human subjects (100 × 260).

Perceptron templates and classification images

Figure 1 compares the Bubbles classification images reported for uppercase and lowercase letters by Fiset et al. (2008) to the letter templates embodied in the connection weights of the perceptron network after training in both conditions. Bubbles images were z-scored classification images whose colored pixels exceeded the significance threshold of a statistical test (the so-called “Pixel Test,” Chauvin, Worsley, Schyns, Arguin, & Gosselin, 2005). As in Fiset et al. (2008), a global Bubbles classification image was then obtained for each letter in each case by summing over the five “partial” classification images corresponding to different bubble sizes. Finally, each letter weight matrix and each classification image was normalized and thresholded. The threshold was obtained in the same way for all weight matrices and all classification images and set to one eighth of the range of values in the matrix considered. This was done so as to emphasize the diagnostic regions falling inside letter bodies as opposed to outside. The resulting diagnostic regions were then coded by a color gradient of increasingly dark red for the perceptron and increasingly dark blue for the bubbles classification images before being superimposed to the letter inputs.

Visual inspection of Figure 1 already suggests that the agreement between model and bubbles data is quite strong. With the possible exception of uppercase letters M and W, the letter regions found useful by human subjects for a detection task are closely matched by the strongest connection weights in the model. This is nowhere more obvious than for what one might call “subset/superset letter pairs” like O and Q, I and F, F and E, F and P, or P and R, pairs of letters that differ only by the presence of a single localized feature in one and not in the other (for instance a diagonal stroke on the bottom right of superset letters Q and R, respectively not present in O and P). In these cases the delta rule emphasizes the connection weights that come from the critical diagnostic region in the superset letter.

Table 1. Cross-correlation values and Frobenius distances between human classification images and trained perceptron weights for uppercase and lowercase letters.

<table>
<thead>
<tr>
<th>Letter</th>
<th>Uppercase</th>
<th>Lowercase</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.823</td>
<td>0.809</td>
</tr>
<tr>
<td>B</td>
<td>0.819</td>
<td>0.798</td>
</tr>
<tr>
<td>C</td>
<td>0.795</td>
<td>0.822</td>
</tr>
<tr>
<td>D</td>
<td>0.799</td>
<td>0.862</td>
</tr>
<tr>
<td>E</td>
<td>0.822</td>
<td>0.837</td>
</tr>
<tr>
<td>F</td>
<td>0.757</td>
<td>0.837</td>
</tr>
<tr>
<td>G</td>
<td>0.806</td>
<td>0.846</td>
</tr>
<tr>
<td>H</td>
<td>0.782</td>
<td>0.854</td>
</tr>
<tr>
<td>I</td>
<td>0.825</td>
<td>0.854</td>
</tr>
<tr>
<td>J</td>
<td>0.782</td>
<td>0.772</td>
</tr>
<tr>
<td>K</td>
<td>0.804</td>
<td>0.814</td>
</tr>
<tr>
<td>L</td>
<td>0.803</td>
<td>0.936</td>
</tr>
<tr>
<td>M</td>
<td>0.773</td>
<td>0.802</td>
</tr>
<tr>
<td>N</td>
<td>0.825</td>
<td>0.787</td>
</tr>
<tr>
<td>O</td>
<td>0.787</td>
<td>0.888</td>
</tr>
<tr>
<td>P</td>
<td>0.799</td>
<td>0.855</td>
</tr>
<tr>
<td>Q</td>
<td>0.805</td>
<td>0.863</td>
</tr>
<tr>
<td>R</td>
<td>0.794</td>
<td>0.811</td>
</tr>
<tr>
<td>S</td>
<td>0.794</td>
<td>0.846</td>
</tr>
<tr>
<td>T</td>
<td>0.820</td>
<td>0.850</td>
</tr>
<tr>
<td>U</td>
<td>0.861</td>
<td>0.787</td>
</tr>
<tr>
<td>V</td>
<td>0.821</td>
<td>0.800</td>
</tr>
<tr>
<td>W</td>
<td>0.798</td>
<td>0.776</td>
</tr>
<tr>
<td>X</td>
<td>0.788</td>
<td>0.812</td>
</tr>
<tr>
<td>Y</td>
<td>0.818</td>
<td>0.821</td>
</tr>
<tr>
<td>Z</td>
<td>0.810</td>
<td>0.807</td>
</tr>
</tbody>
</table>

Table 1: Cross-correlation values and Frobenius distances between human classification images and trained perceptron weights for uppercase and lowercase letters.
Table 1 quantifies the similarity between perceptron connection weights and human classification images in two different ways, using cross-correlations or the Frobenius norm, for uppercase and lowercase letters. The Frobenius norm takes values between zero (identical signals) and one (opposite signals) and is sensitive to location shifts. The cross-correlation measure also takes values between zero (opposite signals) and one (identical signals) and is invariant to location shifts in the 2-D signals being compared. We used the same clipped matrices as presented in Figures 1 and 2, implying all elements were positive, and for our cross-correlations we further normalized all matrices to the unit norm. It can be seen that the cross-correlation is very high in both cases, with an average value of 0.80 for uppercase letters and 0.83 for lowercase letters, whereas the Frobenius norm is more stringent with average values of 0.63 for uppercase letters and 0.58 for lowercase letters. In both cases, cross-correlation and Frobenius norms were very strongly anticorrelated, showing that for all comparison purposes location shifts could be safely neglected. The high similarities we obtain are proof that the delta rule selects diagnostic regions that are remarkably similar to those used by human subjects.

It is possible to further investigate the extent of this agreement by codifying the diagnostic regions used by humans or by the model into certain types of features. Following Fiset et al. (2008), who considered 10 complementary types of letter features that frequently appear in the letter identification literature, we determined the proportion of diagnostic network weights that fell into each of these feature types and compared our results to the bubbles experimental data. For any given letter, a feature was considered to be active at some location if more than 10 letter weights coming from the feature region, as defined by Fiset et al. (2008), were positive.

Figure 2 shows that the order of importance assigned to feature types is on the whole similar for humans and for the model (correlation coefficient \( r = 0.65 \)). Indeed, the top five most useful feature types according to humans and to the model had four features in common (terminations, horizontals, slants tilted right, and slants tilted left). Most notably, and although the delta rule
incorrectly gives more importance to verticals than to horizontals, it assigns the highest importance to terminations. This pattern of results compares favorably with those obtained by the ideal observer model studied in Fiset et al. (2008), which correlated only weakly with the same data ($r = 0.16$) and disagreed on the most useful feature types.

**Weight strengths and number of bubbles**

We now turn to another aspect of the phenomenology of letter identification revealed by the Bubbles method. Figure 3 compares the average number of bubbles required for identification, as reported for uppercase and lowercase letters in Fiset et al. (2008), to
the Frobenius norm of the connection weight matrices obtained in the uppercase and lowercase training conditions. Because the experimentally observed number of bubbles was independent from the thresholds and statistical tests later applied to obtain classification images, we carried out this comparison with raw letter weights rather than with the reprocessed weights from Figure 1. Hence Figure 1, because it only takes positive weights into account, is here only of limited help and sometimes misleading to form an intuition of what the norm of a letter is.

The high correlations obtained for the lowercase ($r = 0.69$, $p < 0.001$) and uppercase letters ($r = 0.60$, $p = 0.001$) demonstrate that in addition to the type and spatial location of diagnostic features, the magnitudes of letter weights also reflect the number of bubbles necessary for identification. In other words, the larger the norm of the weight matrix in the model, the more bubbles are needed to identify the corresponding letter experimentally. This result might appear counterintuitive at first but reflects an interesting trade-off in the model between the letter surface area covered by the weights and their strength. Specifically, the delta rule drives the perceptron from its initial condition of zero weights, through an early condition of large areas of weak and noncontiguous weights to its mature condition of small contiguous areas of strong weights. Because of the quadratic character of the Frobenius norm, a strong increase in a limited number of weights has a higher impact than a small increase in a large number of weights. According to the purely pixel-based comparisons made by the model, letters like W, M, and A (bottom left in Figure 3, upper panel) that can be easily distinguished in the alphabet will need less bubbles to be identified. Being less confusable, these letters will also have their weights less modified by the delta rule: These will stay closer to the early state of spread out weak weight regions, and their magnitudes will remain small. On the contrary, for letters like O and Q (top right on Figure 3, upper panel) that can easily be confused because they are part of subset/superset pairs, the network will be driven to put relatively strong weights on small diagnostic input regions, which makes for large weight magnitudes. But since the diagnostic regions will be smaller and contiguous, and because bubbles are uniformly distributed, it will take many useless bubbles before the diagnostic regions eventually get hit and useful information can be let through.

**Discussion**

It has been claimed that the human letter identification system cannot be construed as, or even informed by, a linear amplifier model. We have observed that a perceptron network trained by the delta rule is analogous to a linear amplifier, and we have demonstrated that it can indeed capture several important aspects of human letter identification as revealed by the Bubbles method. Consequently a first contribution of this article is to show that the LAM-based analysis initially proposed by Murray and Gold (2004a) has much more explanatory power and bearings on what the Bubbles method achieves than has been previously acknowledged (Gosselin & Schyns, 2004). This parallels the recent findings reported in Murray (2012), that standard Bubbles images are very similar to the theoretical Bubbles images obtained from a linear model.

It is worth recalling here that the perceptron model we have used can only operate a direct mapping from raw input pixels to output units. It has no ability to rotate, scale, or reframe the input, no notion of symmetries, no spatial frequency filters with which to decompose and analyze the input, no simple or complex units that would detect oriented bars or edges, and no hidden layers that could perform any other kind of sophisticated computations. These are severe limitations, and indeed it has been known for decades that such networks without a hidden layer can only solve linearly separable discrimination tasks. The finding that this outrageously simple model can nevertheless accommodate so much of the Bubbles data on letter identification is therefore unsettling. At the very least, this demonstrates that the task of letter identification as defined in the experiments of Fiset et al. (2008) would be linearly separable if the target letters were presented in clear conditions. This fact is likely to impact on the type of strategies and features used by subjects: Experiments with stimuli of randomly varying sizes, locations, or shapes within a session may require other processing strategies and diagnostic features from the subject that would presumably not be well captured by a linear perceptron model with a convolution operator and raw pixel inputs. As it happens, Watson and Ahumada (2008) recently introduced a set of template matching models to predict visual acuity from aberated retinal images of letter stimuli and under conditions of noise and location shifts. The models all used realistic preprocessing steps such as optical and neural transfer functions and differed only in their template matching procedure. They found that although an ideal observer model best captured human performance, all models performed at a high level of accuracy, including a linear template matcher using a cross-correlation matching operator. This establishes that linear template matchers that use more sophisticated operators and preprocessing steps can emulate human behavior in realistic conditions.
We have also shown how the perceptron with delta rule can outperform an ideal observer model on accounting for the types of features used by humans during letter identification. A case in point is that only the perceptron model places the most emphasis on “termination” features, like humans do. It is still unclear why this should be so, considering the above mentioned limitations of the network, in particular the lack of any edge feature detectors or of any preprocessing of the input (we note that these specific limitations are also shared with the ideal observer). Part of the reason for the model’s behavior has to do with the exact placing of letter stimuli during training and the fact that the delta learning rule will give more importance to these features that are unique to a letter. Clearly, because not all letters are of same width, if they are centered on the input layer then terminations of wide letters like A, M, or W will be unique and selected by the delta rule. Although this argument does not go all the way to explaining the prevalence of terminations, some of which are emphasized despite overlapping significantly across letters (for instance terminations in I, J, K, and L), it can actually explain the importance of vertical features over horizontal ones for the model. Indeed, horizontal features will have more overlap in the training set because they tend to fall always on the top, middle, or bottom of Arial letters and because letter inputs are adjusted for height. Hence and contrarily to the experimental data, these features will be less diagnostic for the model than vertical features, which exhibit much more variability in locations.

It should be noted that both the perceptron and the ideal observer model used by Fiset et al. (2008) point to a very local type of explanation for the diagnostic features used by human subjects in bubbles experiments. In the ideal observer model, the nature and relative importance of letter features are entirely determined by the experimental stimuli and the bubbles distribution during test trials. In the perceptron model, these same features are determined by letter templates discovered by the delta rule from the experimental stimuli themselves. In neither case is the knowledge acquired on other letter exemplars or with other viewing conditions being taken into account, which again greatly restricts the type of explanations that can be provided by these models. In fact the success of the perceptron model at mimicking human behavior actually suggests that much, though clearly not all (e.g., horizontal vs. vertical features), of what is being discovered through a Bubbles experiment is independent from the details of the subjects’ histories and from their previous expertise with visual letters. Indeed in our simulations this rich previous history is not taken into account, and learning is simplified as a process of repeated exposure to standardized letter targets, which are exactly those used in the bubbles experiments.

We would argue that none of the limitations we have discussed for the LAM and the perceptron, including their linearity, are insurmountable. The LAM itself essentially only makes a statement about the existence of letter templates and of a linear mechanism involved in comparing them to inputs, but it says nothing about the sophistication of letter templates, which could be arbitrarily complex or high-dimensional. In our perceptron implementation of the LAM, the low sophistication of letter templates directly reflects the simplistic nature of the input code, but there are a number of other codes that could advantageously replace these crude pixel inputs, from pyramidal kernels (Bosch, Zisserman, & Munoz, 2007) to SIFT features (scale invariant feature transforms; Lowe, 2004) or shape contexts (Mori, Belongie, & Malik, 2005) to name a few.

Some of these codes attempt to emulate the properties of primary visual areas (Pinto, Barhomi, Cox, & DiCarlo, 2011), while other codes attempt to integrate natural image statistics. In fact taking such a step would recast letter identification into the more mainstream research effort that is generic object recognition. We have argued that some limitations of the model arise because it uses a unique supervised process, and indeed for most computer vision scientist and many neuroscientists of vision, it is useful to distinguish between two stages when modeling visual processes. The first stage essentially performs an analysis of the visual input into universal features; it is often unsupervised and can use deep networks or any of the above mentioned codes (e.g., Serre, Oliva, & Poggio, 2007), whereas the later stage is one of feature selection specific to the task, which uses supervised classifiers that are most commonly of the generalized linear type known as support vector machines (see for instance Pinto, Barhomi, Cox, & DiCarlo, 2008).

Following on the procedure outlined in recent instantiations of state-of-the-art visual object recognition models like HMAX (hierarchical model and X; Serre et al., 2007), the LAM/perceptron’s input code could be upgraded to the product of an unsupervised learning process that turns images into a high-dimensional vector of frequently occurring feature combinations detectors, as determined by the statistics of a training base of natural images. A supervised linear classifier operating on such a code could possibly explain why for instance horizontal features are preferred by human subjects over vertical ones. Last but not least, turning a set of highly entangled patterns into a high-dimensional feature space through a nonlinear function is a demonstrated way to obtain linearly separable patterns (Cover, 1965; see also DiCarlo & Cox, 2007 for a discussion in the context of human vision).
Conclusion

We have observed that a LAM and a linear perceptron are formally analogous systems, and that the delta rule achieves diagnostic feature selection. Building on these observations we have demonstrated that a LAM, when implemented as a perceptron network trained by the delta rule, can reproduce many different aspects of the data gathered on human letter identification using the bubbles experimental method. Not only does this demonstrate that the letter targets used in these experiments are linearly separable based on their pixels alone, but it also brings new information to the existing debate on how to conceive of the bubbles method, supporting the usefulness of Murray and Gold’s (2004a) analysis that had assumed an underlying linear model for visual categorization. Our study also establishes that a trained neural network, although inferior to an ideal model in terms of optimal decision taking, actually provides a superior account of human data on letter identification. Finally we have outlined possible extensions to the LAM that would circumvent its current limitations.

Keywords: letter recognition, delta rule, diagnostic features, bubbles, classification images

Acknowledgments

We thank Daniel Fiset and Frédéric Gosselin for providing the visual stimuli and classification images from Fiset et al. (2008), Jasmin Léveillé and Arnaud Rey for stimulating discussions, and Myriam Chanceaux for her valuable help in an earlier version of this article. This research was conducted under the ERC research grant 230313.

Commercial relationships: none.
Corresponding author: Thomas Hannagan.
Email: thom.hannagan@gmail.com.
Address: Laboratoire de Psychologie Cognitive, CNRS, Aix-Marseille University, Marseille, France.

Footnotes

1 The “bias” unit from which these bias connection weights originate is always clamped to one in order to ensure that the separating hyperplane defined by the connection weights is affine.

2 Alternatively, a proportional count of the number of weights per feature types could be used.

References


### Appendix

#### Description of the model and training procedure

The model is a simple perceptron network with one input layer consisting of $X \times Y + 1$ input units, where $X$ and $Y$ were the dimensions in pixels of the input images and varied across conditions, fully connected to one output layer of 26 output units. The network was trained to recognize the 26 letters of the alphabet, either in uppercase (upper condition) or in lowercase (lower condition). All letters used the Arial font and were black printed on a white background, for an image of dimensions $188 \times 188$ pixels in the lowercase condition and $128 \times 128$ pixels in the uppercase condition. Before being presented to the network, each letter image was converted to a bitmap and rasterized into a large vector of $X \times Y$ elements. Presenting the input to the network means clamping each input unit to the value of its corresponding element in the input. The value is then propagated forward along multiplicative weights, and the net input thereby received by each output unit is used to calculate its output activity. In our model, the output activity of a unit is a linear function of its net input, bounded by [0, 1]. We emulate competition in the output layer by setting the most activated output unit to one and the others to zero. One training epoch consisted of the presentation of all letters exactly once. Connection weights were initialized at zero, and modified at each trial according to the delta rule.
The delta rule states that connection $W_{ij}$ between input unit $i$ and output unit $j$ should be increased (respectively decreased) in proportion to the product of the activation of $i$ and the delta error. The delta error is simply defined as the difference (the “delta”) between the expected target activation and the actual output activation, $\Delta_j = T_j - A_j$:

\[
W_{ij}^{t+1} = W_{ij}^t + \lambda \Delta_j f'(\text{net}_j) A_i, \\
\forall i \in \{1, \ldots, XY\}, \forall j \in \{1, \ldots, 26\}
\]

(A1)

where the learning rate $\lambda$ was set to 0.0001 throughout the simulations. $Ax$ is the activation of unit $x$, $T$ is the target vector associated to the input, e.g., $(1, 0, \ldots, 0)$ for $A$ and $(0, 0, \ldots, 1)$ for $Z$, and where $\text{net}_x$ is the net input to unit $x$. Given that in our case the activation function $f$ is linear, its derivative can be absorbed in the learning rate, reducing the formula to:

\[
W_{ij}^{t+1} = W_{ij}^t + \lambda \Delta_j A_i, \forall i \in \{1, \ldots, XY\}, \forall j \in \{1, \ldots, 26\}
\]

(A2)

The model was trained until the average delta error reached below 0.01, which required 323 epochs of training for lowercase letters and 242 for uppercase letters. In both conditions, the model achieved 100% recognition by the end of training. For each letter unit in the output layer, all the trained weights afferent to it were resized as a 2-D image (of sizes $188 \times 188$ or $128 \times 128$, depending on the lowercase or uppercase training condition). These were normalized and thresholded in order to be compared to the global human classification image for the same letter. The global classification image was the sum of the classification images over the five different bubble sizes tested in Fiset et al. (2008). All programs were written in Python and the Neurolab package was used to implement the networks. The Neurolab package can be found at https://pypi.python.org/pypi/neurolab and the code for the model is available upon request to the first author.