# Counting Objects with Biologically Inspired Regulatory-Feedback Networks

Tsvi Achler, Dervis Can Vural, Eyal Amir

*Abstract*— **Neural networks are relatively successful in recognizing individual patterns. However, when images consist of combination of patterns, a preprocessing step of segmentation is required to avoid combinatorial explosion of the training phase. In practical applications, segmentation is a context dependent task which itself requires recognition. In this paper we propose and develop a biologically inspired neural architecture that can recognize and count an arbitrary collection of objects even if trained with individual objects, without making use of additional segmentation algorithms.**

**The two essential features that govern the neurons in this algorithm are 1. dynamical feedback and 2. competition for activation. We show analytically that while the equations governing the output neurons are highly nonlinear in individual feature amplitudes, they are linear in groups of feature amplitudes. We further demonstrate through simulations, that our architecture can precisely count and recognize scenes in which three and four non-overlapping patterns are presented simultaneously. The ability to generalize numerosity outside the training distribution with a simple learning scheme, lack of connection weights and segmentation algorithms prove regulatory feedback networks not only beneficial for machine learning tasks but also for biological modeling of animal vision.**

## I. Introduction

In this paper we address the problem of recognizing multiple objects that appear together in arbitrary combinations, with repetitions allowed. Since the total number of such possible combinations increases exponentially as a function of the number of objects, the amount of training required by conventional parameter optimization techniques quickly becomes unfeasible. This problem is known in the connectionist literature as the Superposition Catastrophe [1][2]. A common solution that overcomes this difficulty is to segment the image before testing. Unfortunately segmentation is a strongly domain dependent task; it requires prior knowledge of what is to be segmented.

The purpose of this paper is to raise the question whether the recognition of combination of objects can be handled without using the conventional approach of parameter optimization followed by segmentation. Instead, we suggest an architecture motivated by a feedback-based "pre-synaptic inhibition' neural configuration found overwhelmingly throughout the brain. We analytically show how this kind of network is able to recognize collections of objects after being trained with single objects (except for the pathological cases specified in section III.B). We demonstrate this capability experimentally for various combinations of letters as well as randomly generated patterns.

Counting requires an ability to contend with novel combinations of previously learned patterns, thus nicely embodies the above mentioned problem. Though several computational models have been proposed to count [3]-[5], counting and recognition without segmentation remains difficult. More specialized counting methods focus on patterns with simple features such as blood cells or pollen [6]-[8] or isolate regions by utilizing template pattern matching algorithms confined to local regions [9].

An ability to inherently analyze scenes and assess count during recognition appears innate in humans and animals [10][11]. The ability to perform without counting one-by-one (segmenting) may be essential for survival, such as, when determining an escape path from a predator pack. The fact that our algorithm simply uses neurons and connections without additional external algorithms suggests that it may be a viable candidate for biological modeling.

Our method can be qualitatively described as an implementation of Self-Regulatory Feedback, where each input is regulated by its own output nodes [12]-[16]. Though connections are determined by supervised learning, they are not trained in a conventional sense (i.e. through parameter optimization) since there are no connection weights to optimize. A more general version of the regulatory feedback networks that includes optimized connection weights has been previously proposed [17]. However optimized weights may be one of the causes that hinders the algorithms' performance given novel combinations.

In our algorithm, patterns are presented as a scene, where a common 'bag-of-features' feature extractor decomposes patterns equally across the whole scene into simple features to be recognized. The number of instances of these basic features are added across the scene. In the first set of experiments 26 letter patterns are trained using a single presentation of each. During testing, up to four simultaneous letters are placed in the scene and using only the uniform extractor information, the algorithm determines 1) which letters are present 2) how many times the same letter is present. In the second set of experiments the procedure is repeated using randomly generated patterns instead of letters.

Manuscript Received January 5, 2009. This work was supported in part by U.S. Geospatial Agency Grant HM158-06–BAA-0001

T. Achler is with the Computer Science Department, University of Illinois at Urbana Champaign, Urbana IL, 61801 USA (217-244-7118; fax: 217-265-6591; e-mail: achler@uiuc.edu)

Dervis Can Vural is with the Physics Department, University of Illinois at Urbana Champaign (e-mail: vural@uiuc.edu)

Eyal Amir is with the Computer Science Department, University of Illinois at Urbana Champaign, Urbana IL, 61801 USA (e-mail: eyal@cs.uiuc.edu)

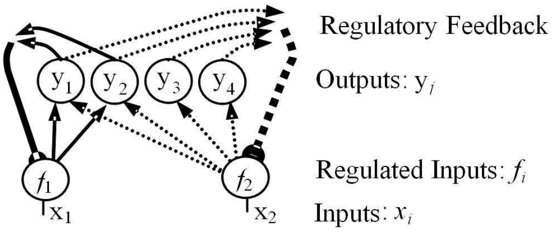

Fig. 1. **Self Regulation**. If $x_1$ affects $y_1$ and $y_2$, then pre-synaptic feedback $f_1$ from $y_1$ and $y_2$ inhibits $x_1$. Similarly if $x_2$ affects $y_1$, $y_2$, $y_3$, and $y_4$ then $f_2$ from $y_1$, $y_2$, $y_3$ and $y_4$ regulates $x_2$

### A. Network Structure

The tight association between inputs (via pre-synaptic cells) and outputs (via post-synaptic cells) required by self-regulatory feedback is depicted in fig(1). Each neuron is regulated by the post-synaptic use of its information. The activation of output $i$ is governed by the nonlinear difference equation

$$y_i(t+1) = \frac{y_i}{N_i} \sum_{j \in R_i} f_j \tag{1}$$

with pre-synaptic inhibition term

$$f_j = \frac{x_j}{Y_j}$$

and feedback term

$$Y_j = \sum_{k \in S_j} y_k(t)$$

For any post-synaptic cell $i$, $R_i$ denotes the set of all pre-synaptic cell connections and $N_i$ denotes the number of connections in $R_i$. For any input $x_j$, $S_j$ denotes the set of all post-synaptic cells connected to it. The total amount of feedback from post-synaptic cells to input $x_j$ is $Y_j$. $f_j$ is the input value after negative feedback. Self-Regulatory networks do not rely on weight parameters. Binary connections are sufficient, simplifying connectivity and training [13][14].

The information $x_j$ can be fully expressed to the output layer only if $Y_j = 1$ (which occurs when inputs and outputs are matched). If several post-synaptic cells are overly active, no post-synaptic cell will receive the full activity of $x_j$ because $Y_j > 1$ thus $f_j < x_j$. Conversely, if $x_j$ is not appropriately represented by the network $Y_j < 1$ and the input is boosted $f_j > x_j$. This negative feedback regulation occurs for every input-output interaction.

The output state of the network at time $t+1$ is determined by input and output states of the network at time $t$. The value of each output node is computed using the inputs connected to it, and the other outputs that share each of these inputs. We iterate the network until all outputs converge to a fixed point.

The iterative nature allows robust inference during the recognition phase. The outputs $\{y\}$ are bounded between zero and a value determined by the inputs $\{x\}$ and this class of equations settle to a steady state [18].

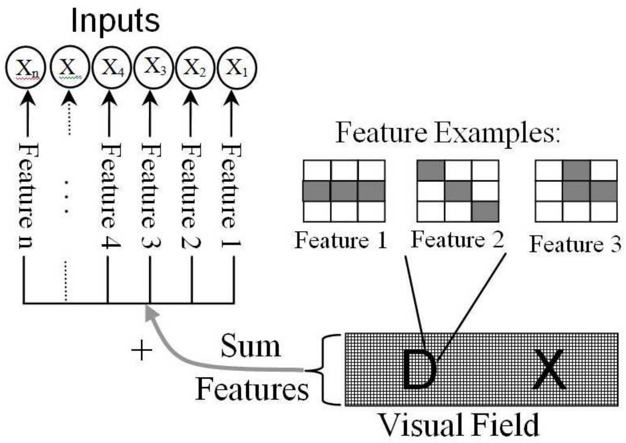

Fig. 2. **Feature Extractor**. If a feature pattern is present anywhere in the visual field, its feature node is incremented. The feature nodes serve as an input vector to the classifiers.

This network implements a powerful multi-class classifier that can process simultaneous patterns [12]. It maintains a simple input-output connectivity where each unit only connects to its own inputs, yet makes complex recognition decisions based on distributed processing [16].

### B. Data Representation

We run two sets of experiments. In the first, we prepare the 26 letters of the alphabet, each placed on a 5x5 pixel grid. Any given pixel can be either black or white. These letters are placed next to one other on a large grid arbitrarily, such that the spacing is no less than 3 pixels.

In the second set of experiments we prepare the visual field with combinations of 30 randomly generated 5x5 patterns. The patterns are formed such that each pixel has a 1/2 probability of being on or off. We chose 4 of these 30 randomly generated patterns each time and place them next to one other such that the spacing is no less than 3 pixels. Then we test all possible occurrences of 4 simultaneous random patterns.

### C. Feature Extraction

The extractor is designed to be similar in spirit to the feature extraction found in the primary visual cortex, and is commonly used in cognitive models [19]. Each feature is defined as a 3x3 black and white pattern(see fig-2). If the pattern $i$ is present anywhere in the visual field then $x_i$ is incremented by one. Since a window of 5x5 can contain 9 different 3x3 windows, there are a total of $2^9$ possible feature vectors.

### D. Training

To recognize patterns, one output (post-synaptic) cell is designated for each letter. When the pattern of a letter is encountered during the training phase, this activates a set of feature cells. In other words, during training, if a feature $x_i$ is present in a class $j$, the input node $i$ is connected to the output node that corresponds to class $j$. After this is done for each class the connections are kept the same during the testing phase. Note that no weights distinguish between the features; each connection is either on or off.

## II. Analytical Results and Simulation

Our results are in the form of derivations and simulations. We first show that scaling all the inputs by $c$ will lead to a scaling of all outputs by $c$. Since the inputs correspond to the number of features, and the outputs correspond to the number of individual objects, it is reasonable to expect such a scaling behavior. We then show that if we scale all features corresponding only to particular object $A$ by $c$ (yet keep the rest unchanged), then only the value of the output that corresponds to $A$ will scale (whereas the rest of the outputs will remain unchanged).

Finally, we report the outcomes of our numeric simulations, and demonstrate that our algorithm can count and recognize combinations of letters and random patterns.

### *A. Superposition Ability of Regulatory Feedback Networks*

It was shown earlier[18] that this class of networks converge in the limit $t \to \infty$. The condition for convergence is

$$y(t+1) = y(t).$$

Let's denote the final value of each output by the subscript 0. The output vector $\vec{y}$ converges to $\vec{y}_0 = (y_{01}, y_{01}, \ldots)$, that satisfies the equation

$$\begin{aligned} N_i &= \sum_j D_{ij}^{(0)}, \qquad (2) \\ D_{ij}^{(0)} &= \frac{x_j}{\sum_k y_{0k}}. \end{aligned}$$

If all inputs are scaled by a constant

$$x_i' = c x_i$$

then eqn(2) remains invariant when the outputs are scaled by the same amount,

$$y_{0i}' = c y_{0i}.$$

In general, the same argument holds for the entire trajectory of an output: If eqn(1) is satisfied by $y(t)$, then

$$\begin{aligned} \tilde{y}_i(t+1) &= \frac{\tilde{y}_i(t)}{N_i} \sum_{j=1}^{N_i} \tilde{D}_{ij} \\ \tilde{D}_{ij} &= \frac{c x_j}{\sum_k \tilde{y}_k} \end{aligned}$$

is satisfied by $\tilde{y}(t) = c y(t)$. This means that if the number of features of A is scaled by c, so must the output corresponding to A.

Let us now demonstrate how the network reacts to separate objects appearing in multitude. If the network is trained to recognize two objects A and B individually, then the outputs $y_A$ and $y_B$, corresponding to the number of A and B satisfy the equations

$$N_A = \sum_{j \in A \cap B} \frac{x_j}{y_A + y_B} + \sum_{j \in A/B} \frac{x_j}{y_A} \qquad (3)$$

$$N_B = \sum_{j \in A \cap B} \frac{x_j}{y_A + y_B} + \sum_{j \in B/A} \frac{x_j}{y_B} \qquad (4)$$

in steady state. Here $A \cap B$ denotes the set of input nodes that are associated with the features common to A and B. $A/B$ and $B/A$ denotes the input nodes that are associated with the features of exclusively A and exclusively B, respectively.

If one A and one B appears together, we have the features

$$x_j = \begin{cases} 2 & \text{if} \quad j \in A \cap B \\ 1 & \text{if} \quad j \in A/B \text{ or } B/A \end{cases}$$

present and we observe that eqn(3) and (4) are satisfied for

$$\begin{aligned} y_A &= 1 \\ y_B &= 1. \end{aligned}$$

Now, if we scale the number of features of A by $\alpha$, and that of B by $\beta$, the common features will be scaled by $\alpha + \beta$ and the above equations will converge to a new equilibrium point $y_A'$ and $y_B'$

$$N_A = \sum_{j \in A \cap B} \frac{(\alpha+\beta) x_j}{y_A' + y_B'} + \sum_{j \in A/B} \frac{\alpha x_j}{y_A'} \qquad (5)$$

$$N_B = \sum_{j \in A \cap B} \frac{(\alpha+\beta) x_j}{y_A' + y_B'} + \sum_{j \in B/A} \frac{\beta x_j}{y_B'} \qquad (6)$$

Thus, if $y_A = 1$ and $y_B = 1$ satisfies eqn(3) and (4), then

$$\begin{aligned} y_A' &= \alpha \\ y_B' &= \beta \end{aligned}$$

must satisfy eqn(5) and (6).

We proved that even though our output cells have a nonlinear dependence to individual input activations, they have a linear dependence to particular groups of input activations. While the former property enables recognition, the latter enables counting. The mentioned linearity is primarily due to our use of binary connections and pre-synaptic inhibition instead of connection weights.

Note that we only showed that the $\vec{y}(t)$ that corresponds to "counting' is a solution of our governing equations, but have not shown that this solution is unique. In fact it is not, and we will discuss such cases in more detail in section III.B.

### *B. Simulations*

Since the governing equations may diverge when certain outputs reach the value zero, if at any time a node takes the value 0, it is replaced by a small number ($10^{-7}$). We take the initial conditions of each node to be a random number, uniformly distributed between $10^{-7}$ and 1.

Two types of "scenes' are simulated: One composed of three letters simultaneously and the other composed of four letters simultaneously. In each case the letters are separated further apart than three squares, the width of the feature

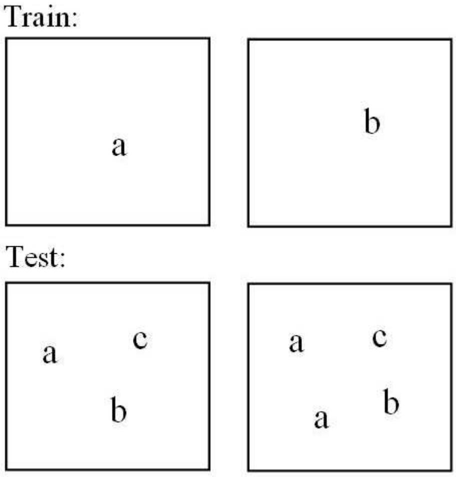

Fig. 3. **Examples of Multiple Pattern Scenes**. Training with single letters (top). Testing with three letters (bottom left) and four letters (bottom right)

extractor window. Hence the feature extractor never gets overlapped features. After the nodes converge to a fixed point we report the activity values of the labeled letter nodes.

In scenes of separate letters i.e. 'a b c' each corresponding node converges to a value of 1, and all others to zero. With one repeat, i.e. 'a a b' the corresponding repeated node converges to a value of 2, the non-repeated node converges to a value of 1, and all others to zero. When three letters are the same, i.e. 'a a a' the node corresponding to the repeated letter converges to a value of 3, capturing all the activity. All other nodes converges to zero. Regardless of scenario, the summed activity of all nodes is 3. All 17,576 possible combinations of 3 letters fall into these three categories capturing both recognition and counting.

We repeat this procedure with four letters and get the same outcome. In scenes of separate letters i.e. 'a b c d' each corresponding node converges to a value of 1. With one repeat i.e.. 'a a c d' the corresponding repeated node converges to a value of 2, while single letter nodes converges to a value of 1. With double repeats i.e. 'a a b b' corresponding repeated nodes both converges to a value of 2. With triple repeats i.e. 'a a a b' the corresponding repeated node converges to a value of 3. With quadruple repeats i.e. 'a a a a' the repeated node converges to a value of 4. In all possible 456,976 four letter combinations, the activity across all nodes sum to 4.

We then present the network with all 26 letters occurring together. All output nodes converge to 1.

In order to show that our algorithm is not only limited to letters, we generate 30 random patterns and test our algorithm with all possible (repeating and non-repeating) combinations of 4. As is with the case of letters, in each case the corresponding node converges to the correct number of patterns with 100% success rate.

## III. Discussion

We demonstrate the ability of regulatory feedback networks to recognize and count multiple letters in a scene without resorting to segmentation algorithms. This outcome is possible because our method overcomes the superposition catastrophe problem by avoiding both parameter optimization and segmentation [12][1]. After learning to recognize only single letter patterns, regulatory feedback networks are able to correctly recognize and count simultaneous combinations of those letters (tested on all combinations of 3 and 4 simultaneous letters). Subsequently this is optimal for situations where simultaneous patterns may emerge.

### A. Limits and Future Work

The main limitation to this model's applicability is that learning is not as generalizable as conventional algorithms. In part this is due to the newness of this method and its inherent nonlinearity leading to difficulty deriving general learning rules. It also requires a developing a new set of learning rules that address inherent limits of the model itself. Its limits can be summarized as A. Generality of features (two defining features appearing together must not constitute a separate defining feature) and B. Flexibility of training (currently our method does not allow for variations within a class).

### B. Generality of Features

Let $x_1$ and $x_2$ be the defining features of classes $A$ and $B$ respectively. In problems where the appearance of $x_A$ and $x_B$ constitutes the defining feature of a third class $C$, our network may or may not yield the desired outcome. We call such $x_1$ and $x_2$ pairs "dependent features". We can demonstrate what happens in these cases by looking at the steady state equations,

$$\begin{aligned} N_A &= \frac{x_1}{y_A + y_C} \\ N_B &= \frac{x_2}{y_B + y_C} \\ N_C &= \frac{x_1}{y_A + y_C} + \frac{x_2}{y_B + y_C}. \end{aligned}$$

When $x_1$ and $x_2$ are both 1, the third equation is simply a linear combination of the first two. Therefore we are left with two equations and three unknowns,

$$\begin{aligned} 1 &= \frac{1}{y_A + y_C} \\ 1 &= \frac{1}{y_B + y_C} \end{aligned}$$

Even though the desired outcome $\{y_A, y_B, y_C\} = \{0, 0, 1\}$ satisfies these equations, in general one could end up with any vector of the form $\{1-u, 1-u, u\}$, for $0 < u < 1$, depending on the initial conditions. In order to resolve these cases, the learning algorithm must be able to either redefine features, or increase the dimensionality of the problem by adding new defining features that are "independent". A trivial way of doing this is to use a feature extraction windows of varying sizes and shapes. Ideally speaking, the number of dimensions of $\vec{x}$ should be greater than or equal to the number of dimensions of $\vec{y}$.

### C. Flexibility of Training

Even though this architecture can cope with noisy data [12] we do not have a uniform way to learn variants of characteristic patterns (such as different fonts). This is because our simple training scheme consists of forming binary connections according to a single sample. One way of extending the training algorithm to data with more variation is by introducing extra layers to the network; whenever the network comes across a drastically different rendition of a letter, say $\mathbb{B}$ instead of $\mathcal{B}$, it could form a new node and dedicate this to the different rendition, $y_{\mathbb{B}}$ and then connect this to the node that corresponds to the usual rendition $y_{\mathcal{B}}$ in a hierarchical fashion. This way activity in either $y_{\mathcal{B}}$ or $y_{\mathbb{B}}$ will stimulate a third node that corresponds to the letter B. Another way of extending training is through clustering methods which can take several data points and produce an averaged prototype.

### D. Relevance Beyond Counting

We showed that if extracted features are summed across the visual field, then the self regulatory feedback networks can sum (count) complex objects. A similar relation holds for any function. If a different function combines extracted features $\vec{x}$ the same function will apply to $\vec{y}$. For example Webers Law states that the minimum amount of noticeable sensory intensity scales linearly with the total amount of sensory intensity. The law applies for light, sound, motor senses, and has been reported to occur within cortical regions of the brain [20]. By analogy if a function that follows Webers law determines the input layer combinations of $\vec{x}$ then that function will appear in the output $\vec{y}$, in accordance the biological findings [20]. Further quantitative analysis is the subject of a future study.

## References

[1] Rachkovskij, D. A. and E. M. Kussul "Binding and normalization of binary sparse distributed representations by context-dependent thinning" *Neural Computation*, vol. 13 no. 2: pp. 411-452, 2001.

[2] F. Rosenblatt, *"Principles of neurodynamics; perceptrons and the theory of brain mechanisms"*, Washington, Spartan Books, 1962

[3] Drazen D "A neural model of quantity discrimination" *NeuroReport*, vol. 15 pp. 2077-2081, 2004

[4] Matthew C. Casey, Khurshid Ahmad "A competitive neural model of small number detection" *Neural Networks*, vol. 19 no. 10 2006

[5] Grossberg, S. and Repin, D. V. "A neural model of how the brain represents and compares multi-digit numbers: spatial and categorical processes" *Neural Networks*, vol. 16 pp. 1107-1140, 2003

[6] Branda, J. A. and A. Kratz, "Effects of yeast on automated cell counting." *Am J Clin Pathol*, vol. 1: pp. 15-26, 2006

[7] D. Kunz, "Possibilities and limitations of automated platelet counting procedures in the thrombocytopenic range." *Semin Thromb Hemost*, vol. 27(3): pp. 229-235, 2001

[8] Swolin, B., P. Simonsson, et al. "Differential counting of blood leukocytes using automated microscopy and a decision support system based on artificial neural networks–evaluation of DiffMaster Octavia." *Clin Lab Haematol*, vol. 25, no. 3, pp, 139-147, 2003.

[9] A. A. Abu-Tarif, V. Khiznichenko, et al. "Object segmentation, quantification, counting, and tracking" *Proc. SPIE*, vol. 5324, pp. 8-16, 2004.

[10] Feigenson, L., Dehaene, S., and Spelke, E. "Core systems of number" *Trends in Cognitive Sciences*, vol. 8 pp. 307-314, 2004

[11] Hubbard, E. M., Piazza, M., Pinel, P., and Dehaene, S. "interactions between number and space in parietal cortex" *Nature Reviews Neuroscience*, vol. 6 pp. 435-448, 2005

[12] T. Achler, C. Omar, et al., "Shedding weights: More with less." *Proceedings of the 2008 IEEE International Joint Conference on Neural Networks*, IJCNN pp. 3020-3027, 2008

[13] T. Achler "Input shunt networks" *Neurocomputing*, vol. 44-46: pp. 249-255, 2002

[14] T. Achler, "Object classification with recurrent feedback neural networks" *Proc. SPIE Evolutionary and Bio-inspired Computation: Theory and Applications*, vol. 6563, pp. 65630K-1-12, 2007.

[15] J. A. Reggia, C. L. D'Autrechy, Sutton G. G., Weinrich M, "Competitive distribution theory of neocortical dynamics" *Proc. SPIE*, vol. 4, no. 3, 1992.

[16] T. Achler and E. Amir, "Input feedback networks: Classification and inference based on network structure." *Artificial General Intelligence*, vol 1: pp. 15-26, 2008

[17] J. A. Reggia, "Virtual lateral inhibition in parallel activation models of associative memory" *Proc. IJCAI, Los Angeles*, vol. 1, pp. 244-248, 1986.

[18] F. E. McFadden "Convergence of competitive activation models based on virtual lateral inhibition" *Neural Networks*, vol. 8, no. 6, pp. 865-875, 1995.

[19] A Treisman, S Gormican "Feature analysis in early vision: Evidence from search asymmetries" *Psychological Review*, vol. 95, no. 1, pp. 15, 1988.

[20] L. Pessoa and R. Desimone "From humble neural beginnings comes knowledge of numbers" *Neuron*, vol. 37 no. 1: pp. 4-6, 2003