Shedding Weights: More With Less

Tsvi Achler, Cyrus Omar, and Eyal Amir

Abstract—**Traditional connectionist models place an emphasis on learned weights. Based on neurobiological evidence, a new approach is developed and experimentally shown to be more robust for disambiguating novel combinations of stimuli. It does not require variable weights and avoids many training related issues. This approach is compared with traditional weight-learning methods. The network is better able to function in different scenarios and can recognize multiple stimuli even if it is only trained on single stimuli.**

I. Introduction

QUALITATIVELY intelligence can be described ability to apply learned information to a new scenario. Classification is an important aspect of this intelligence. We review two connectionist classifiers that can be considered biologically plausible and evaluate their ability to generalize. These classifiers are 1) *Neural Networks* (NNs) and 2) a new type of network we call *Regulatory Feedback Networks* (RFNs). We argue that the most 'intelligent' connectionist networks require the least amount of computational resources, variables and connections, while performing successfully in multiple unforeseen scenarios.

Our biologically motivated network is inspired by the massive number of feedback connections found in the brain [1-3]. Due to its feedback structure it is called *Regulatory Feedback Networks*. Compared to traditional NNs, RFNs perform simpler processing during the training phase, but more extensive processing during testing phase. Instead of determining connection weights based on the training set like NNs, only positive binary associations are determined (e.g. *feature1* implies *X*). Negative associations (e.g. *feature1* implies not-*Y*) are not determined, but inferred during the test phase through feedback connections. This allows the network to be more robust and function outside of its training distribution.

We outline the biological evidence for RFNs and compare them with traditional NNs. The comparisons explore 1) the type of problems each addresses 2) the training requirements 3) resources needed and 4) the number of variables required. A similar analysis can be extended to other common methods such as *Support Vector Machines* (SVMs) or nonparametric computational methods. However, the focus of this paper is limited to biologically plausible connectionist models.

Our stimuli are designed to bring the key issues into the forefront that may be lost in large implementations. The scenarios include: 1) ideal stimuli degraded by random noise 2) multiple stimuli simultaneously, such as may occur with a poorly segmented scene. RFNs show particular promise in detecting novel combinations of multiple stimuli, even though RFNs require: less training, less variables and avoid many training related questions.

Our results are significant because they illustrate the problems associated with connection weights and how a regulatory feedback approach can resolve some of them.

Manuscript received December 15, 2007. This work was supported in part by the U.S. National Geospatial Agency Grant HM1582-06--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).

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

II. Background

A. Connection weights

The idea of adjusting connection weights has been a cornerstone throughout the development of connectionist models [4-6]. The advantage of this method is that in combination with learning algorithms it can have a good degree of autonomy.

However, it requires high degrees of freedom where numerous sets of weights can be chosen for virtually any problem. Weights are adjusted per task for specific applications. Without appropriately selected training sets weight learning can suffer from difficulties such as over-fitting, local minima and catastrophic interference; 'forgetting' of previously learned tasks [7,8]. Furthermore, the function of each unit in relation to its structure can be unclear.

These learning algorithms are difficult to describe in terms of biologically viable neural mechanisms. Most learning theories, such as backpropagation, employ feedback connections only during learning [6]. The feedback connections are not used during classification.

Lastly, recent studies of neuron processing, challenge the idea that synaptic properties can be described by trained connection weights. For example, studies of homeostatic plasticity indicate that synapse strengths appear to maintain a fixed amount of total network activity [9,10].

B. Limits of Learning and Test Distribution Assumptions

Connectionist networks such as NNs are trained with the assumption that the training distribution is similar to the testing distribution [11]. This limitation allows the correlation between input features to outcomes to be determined a-priori through a training set. Unfortunately, this limitation is commonly violated in the natural environment [12] such as in a scene with many stimuli. Suppose a network is trained on stimulus *A* and *B* presented by themselves. If they appear side-by-side their

simultaneous appearance is outside the training distribution. This distribution limit, may limit the network's 'intelligence'.

With efforts to carefully train and avoid of over-training, NNs can maintain some ability to generalize [13]. However, this does not resolve the problems. For example, NNs can be trained to recognize A & B simultaneously. However, if C is separately trained, then simultaneous appearance of both A & C, or B & C still fall outside of the training distribution. Training NNs for every pair of possible stimuli (or triplets, quadruplets, etc.) may require a huge amount of training to cover unforeseen combinations. This is combinatorialy impractical.

To overcome these limits, a scene is often segmented into smaller individual stimuli. But, determining segmentation boundaries is not trivial [14]. Even if feedback or attention processes are employed to guide segmentation [15, 16], segmentation is not flawless. Furthermore, humans can resolve certain stimuli without segmentation [17]. Thus, combined over-reliance on segmentation and distribution limits may result in suboptimal performance.

C. *Importance of Feedback in the Brain*

The ability to modify information in earlier pathways appears to be an important component of recognition processing. Feedback connections can be found in virtually all areas of brain processing. The thalamic system (including thalamic nuclei the Lateral Geniculate Nucleus-LGN, and Medial Geniculate Nucleus-MGN) projects to the cerebral cortex and receives extensive cortical projections back. Relay cells feed forward to the cortex and pyramidal cells feed back to the cortex. These connections have been described as forming a ubiquitous 'triad' structure of regulatory feedback [18,19]. Furthermore, the thalamus may receive as many connections from the cortex as it projects to the cortex [19].

The *Olfactory Bulb* (OB), which processes odors, is analogous to the thalamus. The OB is a separate, modular structure which can easily be studied. Compared to visual or auditory signals, odorous signals are generally reduced in spatial fidelity [20]. Thus odorant processing involves primarily recognition processing as opposed to visual or auditory processing which can encompass both recognition and localization.

Regulatory feedback modulation of early processing appears not one, but two levels of processing within the OB: the local and global circuits. The local circuit, found within the OB glomeruli sub-structure, receives inputs from the olfactory nerve, and connects to mitral/tufted output cells. The output cells simultaneously activate juxtoglomerular cells (Q cell equivalent) which pre-synaptically inhibit the olfactory nerve axons. The global circuit, found between the OB and olfactory cortex, receives inputs through the mitral/tufted cells, and projects information back to granule cells within the OB. These granule cells (another Q cell equivalent) inhibit the mitral/tufted cells.

This feedback establishes nonlinear mechanisms to alter cell activation, based on its previous activity. Such mechanisms can be found within dendritic-dendritic connections involved in coincidence detection (via intercellular NMDA channels). These channels are found in both granule cells [21] and juxtoglomerular cells [22]. Together GABA channels (inhibitory connections), calcium & NMDA channels (multiplicative dynamics) and feedback form an regulatory feedback system of inhibition [23,24]. This forms the basis of RFNs.

Thus, 'intelligent' classifiers that can generalize outside the training distribution can be more biologically plausible, require simpler training and tolerate segmentation errors.

III. Regulatory Feedback Networks

In light of this evidence, RFNs use top-down regulatory feedback to modify input activation. The modified input activity is then re-distributed to the network. This is repeated iteratively to dynamically determine stimuli relevance. In this manner top-down regulatory feedback determines the relevance of inputs to an output node.

This model does not assume calculations are based on predetermined connection weights. All input features are connected equally to associated output nodes. Thus each representation is equally connected to all of its parts. Since connection weights are not relevant, only qualitative relations between feature membership to classes need to be determined during setup; e.g. $y_1 \epsilon \{x_1, x_3 \ldots\}$, $y_2 \epsilon \{x_2, x_3 \ldots\}$. Only positive associations between inputs and outputs are encoded. A potentially large number of negative associations are not encoded. Though RFNs are nonlinear, they are easy to simulate. They are well behaved: stable and maximally bounded by the values of the inputs.

A. *Model Function*

RFN can be qualitatively said to inhibit *ambiguous* inputs in order to amplify the discriminatory ability of the classifier. An input can be said to be *ambiguous during classification* if the current candidate representations of the input tend to share it. This distinction is crucial – traditional NN notions of a feature being *uninformative* are associated statically between representation and its input feature(s). Thus NN feature extraction methods will assign reduced weights to features that are generally uninformative as determined by the training set. Using our approach features weights are not assigned. Subsequently, a feature's information content can be dynamically evaluated under different contexts. This is robust because it does not require the distribution of noise to be known in advance or assumed in the training set. Thus, RFNs are more flexible towards unpredicted combinations of priors since no weighted relations between features or nodes are defined a-priori.

B. *Model Schematic*

RFN is unique due to the tight association between input features and outputs representations. This is implemented by a triad of interconnections between an input, the output it supports and feedback from that output (Figure 1). Every input has a corresponding feedback 'Q', which samples the output processes that the input cell activates. The feedback modulates the input.

Every output must project to the feedback Q processes that correspond to its inputs. For example if an output process receives inputs from I_1 and I_2 it must project to Q_1 and Q_2. If it receives inputs from I_1, it only needs to project to Q_1.

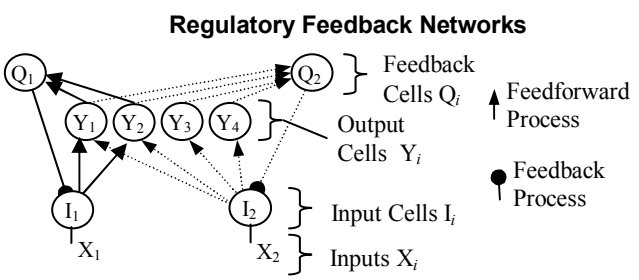

Figure 1: Regulatory Feedback Schematic. Every feed-forward connection has an associated feedback connection. If I_1 projects to Y_1 & Y_2, then Q_1 must receive projections from Y_1 & Y_2 and feedback to the input cell I_1. Similarly if I_2 projects to Y_1, Y_2, Y_3, & Y_4, then Q_2 receives projections from Y_1, Y_2, Y_3, & Y_4 and projects to I_2.

This creates a situation where an output cell can only receive full activation from an input if that input's Q is low. The Q is low if the sum of activation of the outputs that use that input is low. Thus, if representations that share the input are very active, no cell will receive full activation from that input. If outputs share inputs, they inhibit each other at their common inputs, forcing the outcome of competition to rely on other non-overlapping inputs. The more cells have overlapping inputs, the more competition exists between them. The less overlap between two output cells, the less competition, more independent from each other the output cells can be.

The networks dynamically test recognition of representations using 1) regulatory feedback to the individual inputs of representations 2) modifying the next input state based on the input's use 3) re-evaluating representations based on new activity. Steps 1-3 are continuously cycled through. This requires a tight association between inputs and outputs and feedback processes.

RFN is flexible because it doesn't a-priori define which input is ambiguous. Which input is ambiguous depends on which representation(s) are active which in turn depends on which stimuli and task are being evaluated.

C. Equations

This section introduces general nonlinear equations governing RFN. Borrowing nomenclature from engineering control theory, this type of inhibitory feedback is negative feedback, in other words stabilizing or regulatory feedback.

For any output cell Y denoted by index a, let N_a denote the input connections to cell Y_a. For any input cell I denoted by index b, let M_b denote the feedback connections to input cell I_b. The feedback to input I_b is defined as Q_b. Q_b, is a function of the sum of activity from all cells Y_j that receive activation from that input:

$$Q_b = \sum_{j \in M_b} Y_j(t) \qquad (1)$$

Input I_b is regulated based on Q_b, which is determined by the activity of all the cells that project to the input, and driven by X_b which is the raw input value.

$$I_b = \frac{X_b}{Q_b} \qquad (2)$$

The activity of Y_a is a product of its previous activity and the input cells that project to it. This property can arise biologically from NMDA channels that are found within neuron membranes. These channels are mediated by previous neuron activity. If a self-multiplicative (delay) term is not included in *equation 3*, the network can immediately change values and the feedback will not be a function of previous activity. Thus the equations are designed so the output cells are proportional to their input activity, inversely proportional to their Q feedback and also depend on their previous activity [1-3]. A simple division was chosen for *equation 2*, because a subtraction does not guarantee that the stabilizing feedback will be proportional in size to a perturbation. n_a represents the number of processes in set N_a.

$$Y_a(t+\Delta t) = \frac{Y_a(t)}{n_a} \sum_{i \in N_a} I_i \qquad (3)$$
$$= \frac{Y_a(t)}{n_a} \sum_{i \in N_a} \frac{X_i}{Q_i} = \frac{Y_a(t)}{n_a} \sum_{i \in N_a} \left(\frac{X_i}{\sum_{j \in M_i} Y_j(t)} \right)$$

D. Stability

Stability of these equations and other variants have been previously analyzed [25]. In RFN all variables are limited to positive values. Thus the values of Y can not become negative and have a lower bound of *0*. Furthermore the upper values of Y are bounded as well. By definition every input (in N_a) to which Y_a projects, will have Y_a in the feedback branches (M_b) of input I_b. To achieve the largest possible value, all other cells should go to zero. The equation then reduces to:

$$Y_a(t+\Delta t) \le \frac{1}{n_a} \sum_{i \in N_a} \left(\frac{Y_a(t) \cdot X_{\max}}{Y_a(t)} \right) = \frac{1}{n_a} \sum_{i \in N_a} X_{\max} \le \frac{X_{\max} \cdot n_a}{n_a} = X_{\max}$$

If X values are bounded by $X_{max}=1$, then each Y_a is bounded by *1*. Thus the values of Y are bounded between by the amplitude of the inputs and *0*. The picard existence and uniqueness theorem states that if a nonlinear differential equation is bounded and is well behaved locally then it will have a unique solution, i.e.[26].

E. Learning Requirements & Combinatorial Plausibility

For RFN setup, input features active in a labeled training example are connected in a positive binary fashion to their representative output node. More sophisticated methods can include clustering and pruning [27], but are beyond the scope of this paper. Learning is simpler in RFNs because no connection weights are involved. The number of connections required per cell is a function of the number of inputs the cell uses. Addition of a new cell to the network requires only that it forms symmetrical connections to its inputs and not directly connect with the other output cells.

Consequently, the number of connections of a cell representation is independent of the size or composition of the classification network, allowing large and complex feedback networks to be combinatorially and biologically practical [3].

F. Homeostatic Plasticity

Lastly, RFNs to some extent incorporate Homeostatic Plasticity. Homeostatic Plasticity describes the phenomena where biological neurons have been shown to auto-regulate baseline activity by normalizing synapse strength across all synapses through multiple cellular mechanisms [9,10]. In the regulatory feedback network each classifier cell strives for a total activation of 1. If a cell has N inputs each connection contributes 1/N activity. This guarantees that the cell will be as active as its most active inputs. However network function is not strictly dependent on this homeostatic criterion.

With these seemingly simple connections the RFN model can behave informatively in complex scenarios.

IV. Experiment and Network Set-up

This section describes the setup of the networks and methods of the experiments. RFNs and NNs are evaluated using letter examples. The stimuli are 26 letters of the alphabet each composed of 5x5 pixels. Experiments test the networks on noisy and simultaneous presentations of the letters. A feature extractor is used to allow several letters to be represented at once in the visual field.

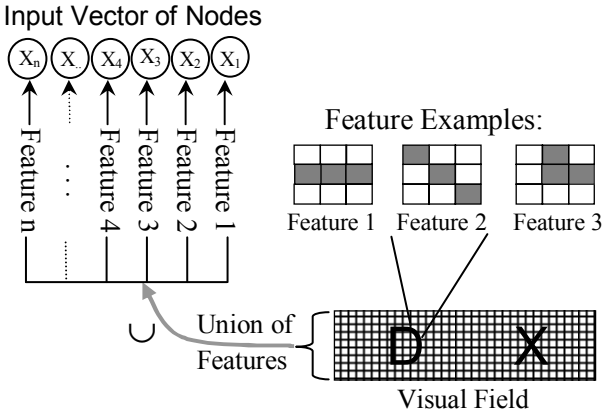

Figure 2: Feature Extractor. If a feature pattern is present anywhere in the visual field, its feature node is activated. The feature nodes serve as an input vector to the classifiers.

The extractor's features are embodied by a 3x3 pixel grid where each possible pattern is considered a feature. Thus there are 512 possible features. The 3x3 pixel grid is applied to the whole visual field. The binary union of features found at every possible location makes the extractor's features location-invariant. Each node represents the maximal value of its feature in the visual field. For example if feature 1 is present once or twenty times in the visual field X_1 will have the value *1*. The values of the *512* possible features form an input vector for the classifiers.

Noisy stimuli are generated using real-valued exponential noise with mean amplitude μ, which is added to the prototypical stimulus. The resulting noisy stimulus vector is given to the classifiers and their result is compared to its original label. All letters are tested and the results are combined to determine the percent of letters correctly identified for that noise level.

Multiple stimuli are generated by placing the prototypical stimuli in the visual field with a distance of greater than three pixels between the individual stimuli. This allows a simple union of prototypical input vectors. All possible N-letter combinations are tested. For example, if *N=2* there are *325* possible two letter combinations. The top *n* active nodes of each network are selected and compared to the original stimuli. The appropriate *n/n* histogram bin is incremented. For example, if *N=4* and the four most active letters are correctly matched, then the *4/4* histogram bin is incremented. If only *3* letters are matched, the *3/4* histogram bin is incremented.

The networks are setup or trained using the features extracted from the 26 prototypical letters. 26 output nodes are created: one for each letter.

A. Neural Network Setup

Both backpropagation and optimized learning methods are used to illustrate training issues.

'Naïve' NNs are trained via backpropagation with momentum. 100,000 training examples are randomly chosen with replacement from the prototypes. The number of hidden units used is increased until was sufficient to learn the input-output mapping to a component-wise tolerance of 0.05. The learning rate is .1, the momentum is 1 and there is a bias unit projecting to both hidden and output layers with value of 1. To improve 'Naïve' performance the NNs are retrained with noisy stimuli and multiple letter combinations. This is further described in the results section.

Optimized learning is performed using the most recent version of the *Waikato Environment for Knowledge Analysis* (WEKA) package currently available [28].

B. Regulatory Feedback Network Setup

To setup RFN, 26 output cells are defined. Each output cell is connected to input features that are active when its prototypical stimulus is displayed.

The RFN test phase involves an iterative algorithm. The more it is allowed to iterate, the more refined the outputs values can become. RFN is iterated until the maximum component-wise change between iterations is below a tolerance of 0.01. The starting activation value of each output is 0.01 for all trials.

RFN iterations can be optimized by eliminating the evaluation of 0-valued inputs and feedback units using sparse matrix techniques. Since the feature vector is often sparse, this results in speedups. Additional optimizations can be performed by removing output units which are zeroed (beyond the scope of this paper).

V. Results

The experiments test the networks on noisy and simultaneous presentations of the letters. This section evaluates the performance of the classifiers and methods to increase performance. Issues regarding setup of the networks and experiments are discussed in section *IV*.

In section *A*, the tolerance of the networks to random noise is tested. RFN has a good tolerance. However, with appropriate training and optimization NNs can outperform RFN. In sections *B, C* and *D* we test the networks for their ability to recognize simultaneous stimuli. With appropriate training and optimization NNs can perform almost as well as RFN on two simultaneous letters. However training and optimization is prohibitive in the four letter case. These limitations become even clearer when NNs must operate in multiple scenarios (as may be necessary in a brain). Thus, in section *E* we show that optimizing for noise affects multiple letter recognition and visa versa. This is significant because retraining a NN for every situation may not be as practical as an inherently more robust network (section *F*).

We begin by remarking that after setup and training described in section *IV*, both NNs and RFNs are able to accurately recognize all 26 letters individually. The resulting NNs had on average 12 hidden units.

A. Letters with Random noise

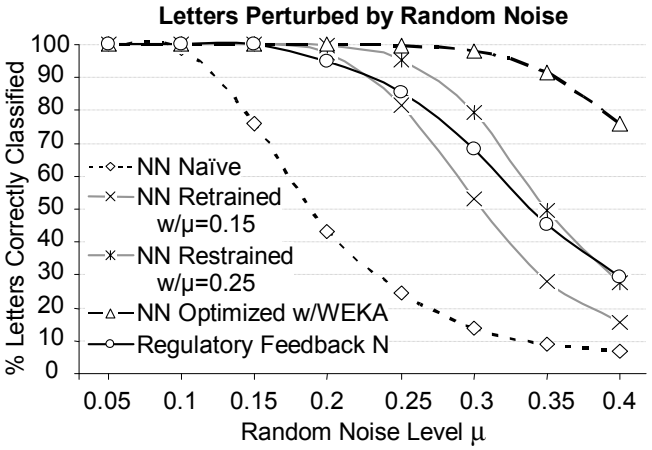

Figure 3: Tolerance of Random Noise. NN implemented w/WEKA is most tolerant. The Naïve NN is least tolerant. RFN is in-between. Tolerance of NN of highly depends on the choice of training set.

Given random noise, naive NNs (trained only on the 26 letters via backpropagation) perform sub-optimally compared to RFN. Yet, the RFNs set-up is even simpler.

Naive NNs achieve less than 70% correct responses when the mean exponential noise level μ is greater than *0.15*. RFNs achieve less than 70% correct responses only when μ is greater than *0.25*. See '*NN Naïve*' and '*Regulatory Feedback N*' in *figure 3*.

Retraining naïve NNs with noisy stimuli greatly improves their performance. Retraining requires 50,000 samples and the degree of improvement depends on the choice of training examples. For example retraining using stimuli with μ=0.15 does not increase performance as much as retraining with μ=0.25. Thus re-training stimuli must be carefully chosen. After such training NNs performance is similar to RFN.

NNs optimized using WEKA perform even better. This method is most optimal for random noise.

Since our goal is to evaluate RFNs in several scenarios as a general classifier, no attempt is made to optimize RFNs.

In summary, though RFN is not an ideal method for distinguishing stimuli in random noise, it performs this task well without optimization.

B. Two Letters Presented Simultaneously

RFN is better suited to disambiguate multiple stimuli. In this task naive NNs perform sub-optimally compared with RFN. Retraining and WEKA optimization improves NN performance almost to the level of RFN.

Naïve NNs correctly characterize 48 ± 3% of two letter combinations, and at least one letter in 96 ± 1% of the combinations. These statistics are determined with 9 NNs and calculated using *student's t* at the confidence level of *p=.05*. Though naïve NNs did not perform as well as RFNs they performed much better than chance: 0.4% two letters, and 17% for one letter out of two.

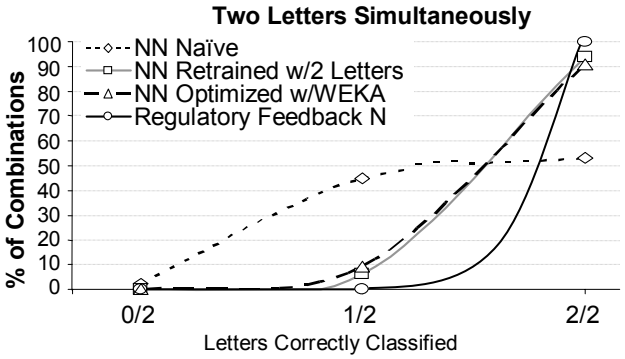

Figure 4: Two Letters Presented Simultaneously. RFN correctly determined all 2 letter combinations. NNs can be retrained and optimized achieve similar results.

NN retraining requires 100,000 samples randomly drawn from the set of all possible pairs of letters. Retraining significantly improves the NN's two-letter performance to 94% of 2/2 combinations. WEKA-optimized NNs also performed well, though slightly less at 91% of 2/2 combinations. See *figure 4*. The accuracy of NNs can possibly be increased further but this requires more training and resources.

In summary, RFN is the most ideal method tested here for distinguishing two simultaneous stimuli. This is bolstered by considering the simplicity involved in RFN setup.

C. Four Letters Presented Simultaneously

The networks are tested on all combinations of four letters simultaneously. The four most active nodes are evaluated for whether they match the four simultaneously presented stimuli. RFNs are able recognize correctly all 4 of the letters in 91% of the combinations. Naïve NNs perform poorly in this task and correctly identify only 3% of the combinations.

Retraining requires 100,000 samples drawn from the set of 14,950 possible four-letter combinations. Due to the number of combinations this is a significantly larger amount of training. The number training operations required increases in a combinatorial fashion with the number of letters presented simultaneously. Retrained NNs correctly identify 32% of 4 letter combinations. This is a marked improvement over naïve NNs and slightly better than the WEKA optimized NNs, which correctly identifies 28% of combinations. See *figure 5*.

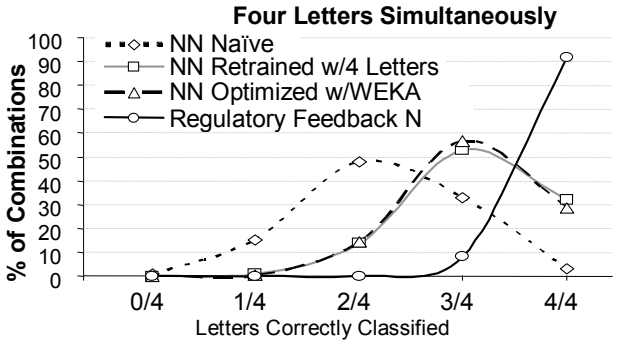

Figure 5: Four Letters Presented Simultaneously. RFN remains the most optimal method even when compared to retrained and optimized NNs. While retraining increased 4/4 performance, RFN achieved by far the most matches.

Thus, the RFN is the most ideal method tested here for distinguishing four simultaneous stimuli. Yet it is the simplest to set-up.

D. Five Letters

RFN are able to recognize correctly 79% of 5/5 letters from the 65780 possible combinations. This is astonishing given the amount of feature overlap in the inputs due to the multiple stimuli. Retraining and evaluation of NNs on five letters is not attempted because of the prohibitive number of examples required.

E. Retraining Can Reduce Robustness

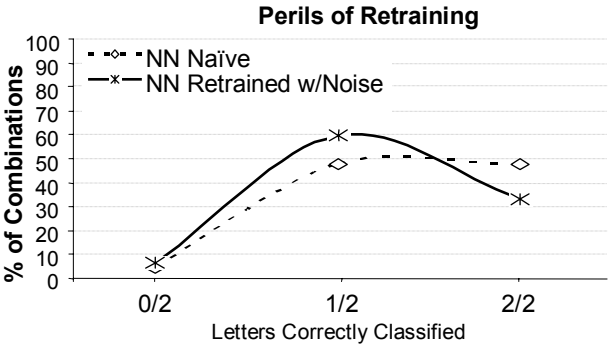

Figure 6: Retraining for Noise Can Decrease Multi-Letter Performance. After retraining naïve NNs for noise, the two letter performance is reduced. 2/2 matches decrease by 14% while 1/2 matches equivalently increase.

In many cases with retraining NN performance can be made comparable to RFN performance. However, when networks are trained for a particular task their performance in other tasks may decrease. Retraining a NN for the noise task can decrease its performance on the two-letter task. This is demonstrated in *figure 6*. After retraining naïve NNs on noisy stimuli, the composite of 9 re-trained networks show on average 47 less correct identifications of 2 letter combinations. This represents a 14% decrease in correct classifications ($p<<0.001$; $n=9$).

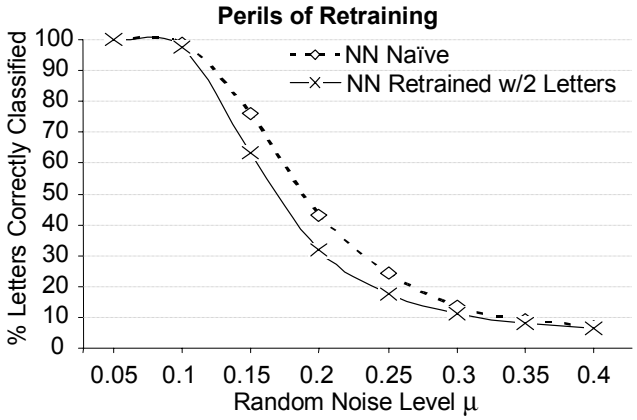

Figure 7: Retraining for Multiple Letters Can Decrease Tolerance to Noise. After retraining naïve NNs for two letters, noise performance is reduced. A smaller number of noisy letters are correctly classified at each noise level.

Similarly, retraining naïve NNs for the two letter task can decrease its performance in the noise task. *Figure 9* shows the decreased performance in a NN retrained for the two-letter task and tested for the noise task. To overcome this cross-interference, these scenarios would have to be carefully trained together which is not a simple problem (discussed further in section F). The innate robustness of RFN is a powerful way to avoid this issue.

F. Resource comparison: NN vs. RFN

In this section we compare the resources required for each network's set-up, connections, training phase, retraining phase and test phase.

RFN structure requires a sparse connectivity matrix for each output. The 'connection strengths' are determined by the number of feature nodes that are connected as inputs. Thus none of the 'connection strengths' are variable. The NN requires 12 hidden nodes and a bias node: subsequently 6494 variable weights.

The RFN was connected based on a single prototype for each letter. This requires only a single presentation. In contrast the naïve NN required up to 100,000 training cycles.

During the test phase, it takes NN's the equivalence of about 10,702 arithmetic operations to pass activation through its feed forward network regardless of stimulus difficulty. For single letters it takes RFN an average of 25,000 operations or 6 iterations to achieve less than a .01 change between iterations. For four simultaneous letters it takes an average of 90,000 operations or 23 iterations before reaching this criterion. RFNs may be interrupted at any point and in many cases the basic trends are observable within the first three iterations.

To achieve performance comparable to RFNs, NNs require various amounts of retraining based on the task. For two letters, 100,000 cycles were required for each of the 325 two letter combinations. For four letters, 100,000 cycles were required for each of the 14950 combinations. For noisy stimuli 50,000 cycles were required.

To be robust in all these tasks simultaneously, a naïve NN would have to be trained for all of them. The training sets would have to be carefully chosen and interwoven to avoid catastrophic interference [7,8]. Otherwise, NNs function can degrade as shown in figures 6 and 7.

Thus, inherent robustness may be more optimal than a training algorithm approach.

VI. Conclusion

In this section we summarize our results and discuss the significance of our findings in relation to natural scenarios.

Using a very simple set-up process, RFN is robust in multiple scenarios. RFNs are also computationally economical because they do not require many variables or a combinatorialy-implausible connectivity matrix [3].

In some scenarios such as stimulus perturbation by random noise, NNs can be trained to exceed RFN performance. However, random noise is not necessarily a very common classification scenario. For example, random noise may naturally occur during a fog or sandstorm.

On the other hand, scenes composed of multiple stimuli commonly occur in natural scenarios. Within these scenarios RFNs are most practical. Such scenarios can be described as occurring outside of the training distribution.

Organisms must maintain flexibility and cope with these scenarios. For example, an animal which has only seen single predators may encounter two predators at the same time. That animal will not have time for retraining.

Not only can NN training requirements limit the network's ability to generalize, the NN training strategy requires carefully planned training protocols. Since retraining on one scenario can degrade previous training on another, NN retraining for multiple simultaneous scenarios are even more likely to become complex and combinatorially intractable. As the number of classes and features or the complexity of the environment grows, the number of training operations grows nonlinearly. Thus plausible NN training requires a combinatorially intractable amount of retraining.

Furthermore, backprop-like learning signals are yet to be found within biological networks. Thus, from a biological and computational perspective RFN may be a more plausible model of the brain [2].

NN training requirements may vary with optimized training algorithms such as: quickprop, resilient backprop and conjugate gradient (i.e. [29]). However, such algorithms require other constraints and do not address the exploding complexity required by the training set.

Yet NNs are important. RFN are not suited for all applications. For example, since RFNs function outside of the training distribution, RFNs are not universal approximators. Thus they may not be able to imitate any arbitrary function like NNs can.

In conclusion, regulatory feedback networks represent an important approach to classification. RFNs require fewer resources and are more robust. Even a structure that is only slightly better able to generalize without retraining can conserve many computational resources. Thus, studies to gain insight about inherent robustness are very important.

References

[1] Achler T, (2002) Input Shunt Networks, Neurocomputing, 44-46c: 249-255.

[2] Achler T, (2007) Object classification with recurrent feedback neural networks, Proc. SPIE, Evolutionary and Bio-inspired Computation: Theory and Applications, Vol. 6563.

[3] Achler, T., Amir, E. (2008) Input Feedback Networks: Classification and Inference Based on Network Structure, Artificial General Intelligence Proceedings V1: 15-26

[4] Hinton G. E, Sejnowski T. J. (1986) Learning and Relearning in Boltzmann Machines. In Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Vl 1 MIT

[5] Rosenblatt, F. (1958) The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain, Psychological Review, V65.

[6] Rumelhart, D. E., & McClelland, J. L. (1986). On learning the past tenses of English verbs. In J. L. McClelland & D. E. Rumelhart (Eds.), Parallel distributed processing (V2, p216-271). MIT Press.

[7] McCloskey, M. & Cohen, N.J. (1989) Catastrophic interference in connectionist networks: The sequential learning problem. In The Psychology of Learning and Motivation, V24. NY Acad, p109–165.

[8] Sharkey, N.E. & Sharkey, A.J.C. (1995) An analysis of catastrophic interference. Connection Science, 7, 301–329.

[9] Turrigiano G G, Nelson S B. (2004) Homeostatic Plasticity In The Developing Nervous System, Nature Reviews Neuroscience 5, 97-107

[10] Marder E, Goaillard JM. (2006) Variability, compensation and homeostasis in neuron and network function. Nat Rev Neurosci, 7(7):563-74.

[11] M. Sugiyama, Active (2006) Learning in Approximately Linear Regression Based on Conditional Expectation of Generalization Error, Journal of Machine Learning Research, 141-166:7.

[12] Marcus, G.F. (1998) Rethinking eliminative connectionism Cognit. Psychol. 37, 243—282.

[13] Elman, J.L. (1998). Generalization, simple recurrent networks, and the emergence of structure. In M.A. Gernsbacher and S.J. Derry (Eds.) Proceedings of the Twentieth Annual Conference of the Cognitive Science Society.

[14] Charles, J. J., L. I. Kuncheva, B. Wells, and I. S. Lim. 2006. An evaluation measure of image segmentation based on object centres. Image Analysis and Recognition, Pt 1 4141:283-294.

[15] van der Velde, F., de Kamps M., (2001) From Knowing What to Knowing Where: Modeling Object-Based Attention with Feedback Disinhibition of Activation, Journal of Cognitive Neuroscience, 13(4), 479-491

[16] Herd, S.A. & O'Reilly, R.C. (2005). Serial visual search from a parallel model. Vision Research, 45, 2987-2992.

[17] Duncan, J. and G. W. Humphreys (1989). "Visual-Search and Stimulus Similarity." Psychological Review 96(3): 433-458.

[18] Reggia, J. A., Dautrechy C. L., (1992). "A Competitive Distribution-Theory of Neocortical Dynamics." Neural Computation 4(3): 287-317.

[19] LaBerge, D. (1997) "Attention, Awareness, and the Triangular Circuit". Consciousness and Cognition, 6, 149-181

[20] Atema J, (1998) Distribution of Chemical Stimuli, in Sensory Biology Of Aquatic Animals, Atema J, Fay RR, Proper AN & Tavolga eds, p 29-56. Springer-Verlag, NY

[21] Chen WR, Xiong W, & Shepherd GM, (2000) Analysis of relations between NMDA receptors and GABA release at olfactory bulb reciprocal synapses, Neuron, 25:625–633.

[22] Aroniadou-Anderjaska V, Zhou F-M, Priest CA, Ennis M, & Shipley MT, (2000) Tonic and synaptically evoked presynaptic inhibition of sensory input to the rat olfactory bulb via GABAB heteroreceptors. J Neurophysiol, 84: 1194–1203.

[23] Isaacson J S. and Strowbridge B W (1998) Olfactory Reciprocal Synapses: Dendritic Signaling in the CNS, Neuron, Vol. 20, 749–761.
[24] Lowe, G. (2002) Inhibition of backpropagating action potentials in mitral cell secondary dendrites. J Neurophysiol 88: 64–85.
[25] McFadden F E. (1995) Convergence of Competitive Activation Models Based on Virtual Lateral Inhibition, Nrl Ntwrks V8#6
[26] Nagle, Saff, Snider (2000) Fundamentals of Differential Equations and Boundary Value Problems, 3rd ed, Ch13.
[27] J. S. Sanchez, L.I. Kuncheva (2007) Data reduction using classifier ensembles, Proc. 11th European Symposium on Artificial Neural Networks
[28] Ian H. Witten and Eibe Frank (2005) "Data Mining: Practical machine learning tools and techniques", 2nd Edition, Morgan Kaufmann, San Francisco, 2005. Software available at http://www.cs.waikato.ac.nz/ml/weka/ (version 3.5.6)
[29] Fahlman, S.E. (1989), "Faster-Learning Variations on Back-Propagation: An Empirical Study", in Touretzky, D., Hinton, G, and Sejnowski, T., eds., Proceedings of the 1988 Connectionist Models Summer School, Morgan Kaufmann, 38-51.