# Shedding Weights: More With Less

Tsvi Achler, Cyrus Omar, and Eyal Amir

*Abstract*—**Traditional connectionist classification models place an emphasis on learned synaptic weights. Based on neurobiological evidence, a new approach is developed and experimentally shown to be more robust for disambiguating novel combinations of stimuli. It requires less training, variables and avoids many training related questions. Instead of determining all connection weights a-priori based on the training set, only positive binary associations are encoded (i.e. *X has Y*). Negative associations (i.e. *X does not have Z*) are not encoded, but inferred during the test phase through feedback connections. This allows the network to function outside its training distribution. For example, the network is able to recognize multiple stimuli even if it is only trained on single stimuli. We compare the accuracy and generalization of this network with traditional weight learning networks.**

## I. Introduction

Qualitatively intelligence is the ability to apply the learned information to a new scenario. Classification is an important aspect of this intelligence. Connectionist networks such as Neural Networks (NNs) are trained with the assumption that the training distribution is similar to the testing distribution [1]. This limitation allows the correlation between input features to outcomes to be determined a-priori through a training set. However, this limitation is commonly violated in the natural environment [2] such as in a scene with many stimuli. For example, if a network is trained on stimulus A and B separately, if they appear side-by-side their simultaneous appearance is outside the training distribution. Thus the distribution limit, may limit the network's 'intelligence'.

By careful training and avoidance of over-training, NNs can maintain some ability to generalize [3]. 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.

Another approach treats multiple stimuli as a scene and segments or isolates each individual stimulus. However, determining segmentation boundaries is not trivial [4].

Even if feedback or attention processes are employed to guide segmentation [5, 6], it is not flawless. However, humans can resolve certain stimuli without segmentation [5-7]. Combined over-reliance on segmentation and distribution limits may result in suboptimal performance. Thus, 'intelligent' classifiers that can generalize outside the training distribution can require simpler training and tolerate segmentation errors.

We propose a new type of biologically motivated network [8-10] which is inspired by the massive number of feedback connections found in the brain. Due to this structure this form of connectionist neural network are named *Input Feedback Network* (IFN). Compared to traditional NNs, IFNs performs simpler processing during the training phase, but more extensive processing during testing phase. It encodes positive binary associations during training and infers negative associations during testing. NN processing during testing can be described as proceeding via a simple feed-forward process. IFN processing during testing can be described as a feed-forward/feedback loop which evaluates the relevance of each input feature dynamically [8-10].

We outline the biological evidence for IFNs 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. Similar issues can be explored with other methods such as support vector machines (SVM) or nonparametric 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 larger implementations. The scenarios include: 1) ideal stimuli degraded by exponential noise with mean $\mu$ 2) multiple stimuli simultaneously, such as may occur with a poorly segmented scene. IFNs show particular promise in detecting novel combinations of multiple stimuli, even though IFNs require less training, variables and avoid many training related questions.

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

## II. Background

### A. Connection weights

The idea of adjusting connection weights has been a cornerstone throughout the development of connectionist models including Perceptrons, Parallel Distributed Networks, and Boltzmann Machines [11-13]. Networks based on weight adjustment are powerful, flexible and in combination with learning algorithms can have a good degree of autonomy. These methods allow high degrees of freedom where numerous sets of weights can be chosen for virtually any problem. Weights are adjusted

per task for specific applications.
However without appropriately selected training sets, they can suffer from difficulties such as over-fitting, local minima and catastrophic interference; 'forgetting' of previously learned tasks [14,15]. 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 [13]: 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 [16,17].

### B. Importance of Feedback

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 input 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.
Input 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 input feedback system of inhibition [23,24]. This forms the basis of IFNs.

## III. Input Feedback Networks

In light of this evidence, IFNs use top-down input feedback to modify input activation. The new input activity is then re-distributed to the network. This is repeated iteratively to dynamically determine stimuli relevance. In this manner top-down input 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. Each representation is equally connected to all of its parts. Since connection weights are not needed, the only free parameters represent qualitative relations of feature membership in classes i.e. $y_1 \epsilon \{x_1, x_3 \ldots\}$, $y_2 \epsilon \{x_2, x_3 \ldots\}$. Thus, only positive associations between inputs and outputs are encoded, and a potentially large number of negative associations are not explicitly encoded. Though IFNs are nonlinear, they are easy to simulate. They are well behaved: stable and maximally bounded by the values of the inputs.

### A. Qualitative Functionality

IFN 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, IFNs are more flexible towards unpredicted combinations of priors since no weighted relations between features or nodes are defined a-priori.

### B. Model Schematic

IFN 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.

This can also be described from the perspective of the output processes. 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 $Q_1$.

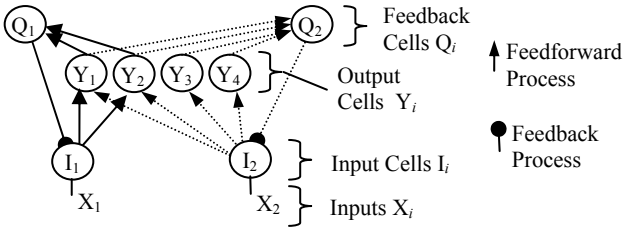



Fig. 1. Input Feedback Schematic. Every feedforward connection has an associated feedback connection. If $I_1$ projects to $Y_1$ & $Y_2$, then $Q_1$ must receive projections from $Y_1$ & $Y_2$ and provide feedback to the input cell $I_1$. Similarly if $I_2$ projects to $Y_1$, $Y_2$, $Y_3$, & $Y_4$, then $Q_2$ 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) input 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.

IFN 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 IFN. Borrowing nomenclature from engineering control theory, this type of inhibitory feedback is negative feedback, in other words stabilizing 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 can arise biologically from NMDA channels, which depend on previous activity. If such multiplicative (delay) term is not included, 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 [4]-[5]. A simple division was chosen because for example a subtraction does not guarantee that the stabilizing (negative feedback) feedback will be proportional to the perturbation. $n_a$ is 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 variations have been previously analyzed [25]. In our model, 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 the purpose of this work, input features active in a labeled training example were connected in a positive binary fashion to a representative cell. More sophisticated methods can include clustering and pruning [27], but are beyond the scope of this paper. Learning is simpler in IFNs because no connection weights to be learned. 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 [10].

### F. Homeostatic Plasticity

Lastly, IFNs 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 [16,17]. In the input 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 IFN model can behave informatively in complex scenarios.

## IV. Set-up for comparisons

IFNs and NNs were evaluated using letter examples. The stimuli are composed of the 26 letters of the alphabet, each 5x5 pixels. These are placed in a visual field that allows several letters to be present at once. A simple bag-of-features type feature extractor is applied to all stimuli in the visual field.

The arbitrary extractor is implemented via 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 and the binary union of resultant features serves as inputs to the classifiers, making the features location-invariant. Noisy and simultaneous presentations of these ideal stimuli were tested. For multiple simultaneous stimuli, the 512 features extracted for each letter are combined together into one feature vector, which is presented to the network. Each feature in the combined input represents the maximal value of that feature across all stimuli. For noisy stimuli, real-valued exponential noise was added to the ideal stimuli.

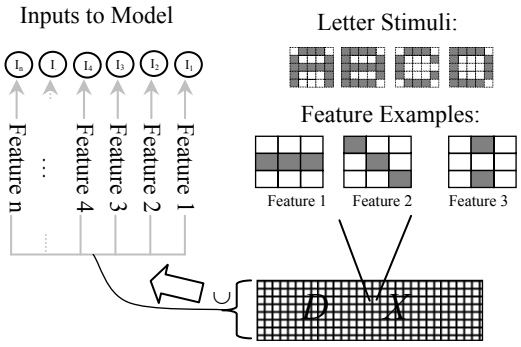



Fig 2. Simple feature extractor that presents collective pixel patterns presented to the networks.

The networks were trained on the features extracted from the 26 prototypical letters. 26 output nodes were created one for each letter.

### A. Neural Networks

We use both old and new methods of NN training to illustrate the potential pitfalls of training. 'Naïve' NNs were trained via backpropagation with momentum. 100,000 training examples were randomly chosen with replacement from the prototypes. The number of hidden units used was increased until was sufficient to learn the input-output mapping to a component-wise tolerance of 0.05. The learning rate was .1, the momentum was 1 and there was a bias unit projecting to both hidden and output layers with value of 1. To improve 'Naïve' NN performance it was retrained with noisy stimuli and multiple letter combinations.

'WEKA Optimized' NNs were implemented using WEKA using the most recent version available [28].

### B. Multiple Stimuli Presentation

All possible N-letter combinations are tested. For example with two letter presentations there are 325 possible combinations.

The top *n* active nodes of each network were selected and compared to the original stimuli. If there is a complete match then we increment the count in the *n/n* histogram bin. Incomplete matches were counted in their bins respectively.

### C. IFN

For IFN, 26 output cells were defined and each output cell was connected to the active input cells. IFN involves an iterative algorithm where the longer it is allowed to run the more refined the outputs values become. The IFN was evaluated until the maximum component-wise change between iterations was below a tolerance of 0.001. The starting value of the IFN was 0.01 for all outputs in all trials.

Evaluation could be optimized by eliminating the evaluation of 0-valued inputs and feedback units using sparse matrix techniques. Since the feature vector was often sparse, this resulted in speedups. Additional optimizations could be performed (beyond scope of paper) by removing output units which are zeroed because of these removed inputs.

## V. Results

The resulting NN had 12 hidden units. Both NN and IFN were able to accurately recognize all 26 letters individually.

The performance of NN and IFN were evaluated on letter classification 1) in a noisy environment and 2) within multiple simultaneous letters. We tested NN and IFN on all possible combinations of 2, 4 and 5 letters for their ability to generalize to multiple simultaneous stimuli.

*A. Random noise*

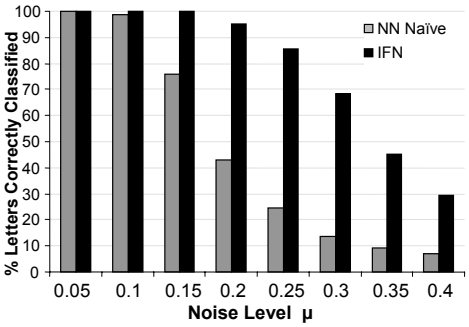



Fig 3: Naïve NN vs IFN Resistance to Noise

Naive NNs were sub-optimal compared with IFN, achieving less than 70% correct with $\mu>0.15$. IFN achieved less than 70% correct with $\mu>0.25$.

*B. NNs Retrained for Random noise*

However retraining with noisy stimuli greatly improved the performance of the NN's. The re-training was performed with 50000 samples. The degree of improvement depended on the choice noisy training examples. For example re-training using stimuli with μ=0.15 did not increase performance as much as training with μ=0.25. Thus re-training stimuli must be carefully chosen. However, NNs trained using WEKA performed better than IFNs. Since our goal is to evaluate IFNs as a general classifier, no attempt was made to optimize IFNs.

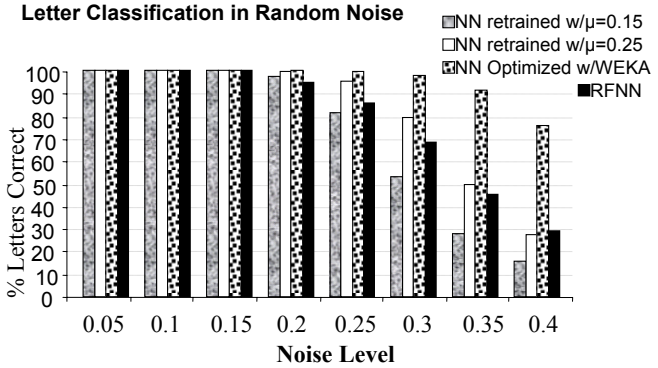



Figure 4: Classification of letters altered by random noise after NN retraining or optimization. Different retraining strategies give different results. NN can be retrained to be more optimal than IFN.

*C. Two Letters*

The two letter case showed similar trends as noise. Naive NNs were sub-optimal compared with IFN, and retraining improved the NNs. However in this case, IFN performed 100%.

Naïve NNs correctly characterized 48±3% (p=.05; n=9) of two letter combinations, and at least one letter in 96±1%(p=.05; n=9) of the combinations (using student t). Though NNs did not perform as well as IFNs they performed much better than chance (0.4% & 17%) respectively.

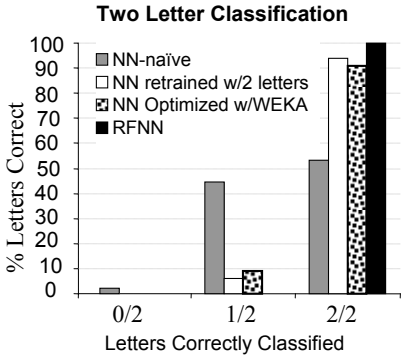



Figure 5: Two Letter Classification improved with training. However, IFN is most optimal.

NN retraining was done using 100,000 samples randomly drawn from the set of all possible pairs of letters. Retraining significantly improved 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. The accuracy of NNs can likely increase further given more training.

*D. Four letters*

The networks were tested on all combinations of four letters simultaneously. The four most active nodes were evaluated for whether they match the four simultaneously presented stimuli. IFNs were still able recognize correctly all 4 of the letters 91% of the time. Naïve NNs were not as successful. Retraining was accomplished using 100,000 samples drawn from the set of 14,950 possible four-letter combinations. However this time a significantly larger amount of training would be required to match the performance of the IFN.

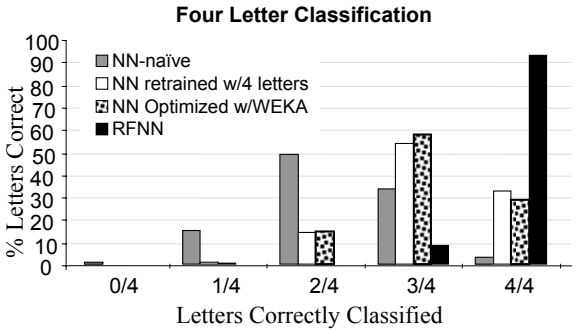



Figure 6: Four Letter Classification improved with training. However, this requires more resources that two letters and IFN is still most optimal.

*E. Five Letters*

IFN were still able to recognize correctly 79% of 5/5 letters from the 65780 possible combinations (Figure 7.) This is astonishing given the amount of feature overlap in the inputs due to multiple stimuli at once.

At 5 letters, naïve NNs still performed well above chance. Based on chance less than one 5/5 classification should have occurred while in the characteristic NN presented there were 470. Retraining the neural networks on five letters was not attempted because of the prohibitive number of examples required.

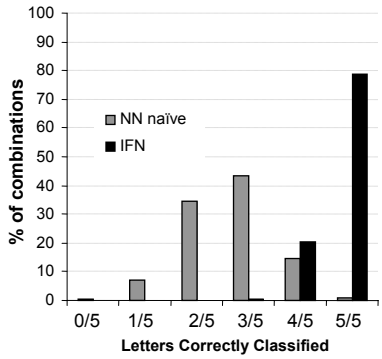
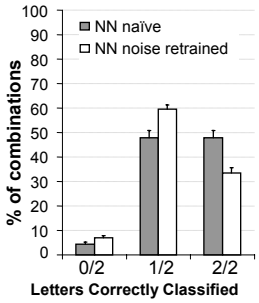


Figure 7: Five Letter Classification

### F. Retraining Can Reduce Robustness

Although neural network performance could be made comparable to IFN performance in many cases, when networks were trained for a particular task, their performance during other tasks decreased. Training for the noise task decreased the performance of the networks on the two-letter task (figure 8). The composite of 9 retrained networks showed on average 47 less 2/2 combinations were recognized. This represents a 14% decrease in correct classifications ($p<<0.001$; $n=9$).



**Retraining Interference 1**

Fig 8: Retraining for noise decreases two letter task performance

Similarly, figure 9 shows the decreased performance of a neural network 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 later sections). In IFN the robustness is innate.

**Retraining Interference 2**

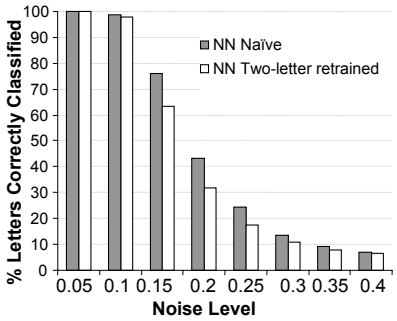


Fig 9: Retraining for two letters decreases noise performance

### G. Resource comparison: NN vs. IFN

IFN structure requires a sparse connectivity matrix for each output. The overall connection strengths correspond to input size. Thus none of the 'connection strengths' are variable. The NN required 12 hidden nodes and a bias node: subsequently 6494 variable weights.

The IFN was trained on a single prototype of each letter, and required only a single view. The naïve NN required up to 100,000 training cycles.

After training both networks were able to classify all letters. It took NN's the equivalence of 10,702 arithmetic operations to pass activation through its feed forward network regardless of stimulus difficulty. For single letters it took IFN an average of 25,000 operations or 6 iterations to achieve a .01 maximum change between iterations. For four simultaneous letters it took an average of 90,000 operations or 23 iterations before reaching this criterion. IFNs may be interrupted at any point and in many cases the final trends were observable within the first three iterations.

To achieve performance comparable to IFNs, retraining of NNs was required: 100,000 cycles using 14950 combinations for four letters, 100,000 cycles using 325 two letter combinations, 50,000 cycles with noisy stimuli $\mu=.25$. These parameters were shown for separate NNs. To be as robust as IFN one NN would have to be trained on all of these. These training sets would have to be carefully chosen and interwoven to avoid catastrophic interference [14,15]. As the number of classes and features or the complexity of the environment grows, the number of training operations grows nonlinearly. Thus in certain scenarios NN retraining may require a combinatorially intractable requirement of training.

The number of required training cycles may change 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.

Thus a structure that is better able to inherently generalize without training is be more suitable for recognition processing, especially given multiple stimuli.

## VI. Conclusion

Our analysis shows that there is no simple solution to all classification problems. Neural networks can be trained to exceed IFN especially if the test stimuli are perturbed by high levels of random noise. However this may require carefully planned training protocols.

If the test stimuli are composed of multiple stimuli, IFNs are more practical. Since retraining on one scenario can degrade previous training on another, NN retraining for multiple simultaneous scenarios can become complex and combinatorially intractable. IFNs are also more economical because they do not require as many variables and do not require complete connectivity.

Yet IFN may not be suited for all tasks. For example, since IFN function outside of the training distribution, IFNs are not universal approximators. Thus they may not be able to imitate any arbitrary function.

However from a biological survival/evolution perspective 1) NN training can impose a limitation on the network's ability to generalize 2) stimuli outside of the training distribution often occur in the natural environment in the form of multiple stimuli 3) maintaining similar distributions during training and testing requires computational costs in terms of training and weight parameters 4) the quality of feedback signal requiring learning within distribution (i.e. backprop) have not been found biologically.

In conclusion, input feedback networks are an important approach. Organisms likely take the IFN approach requiring fewer resources (along with other methods) to help solve recognition problems important for their survival.

## REFERENCES


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

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

[3] 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.

[4] 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.

[5] 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

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

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

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

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

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

[11] 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

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

[13] 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.

[14] 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.

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

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

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

[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.