Input Feedback Networks: Classification and Inference Based on Network Structure

Tsvi Achler and Eyal Amir
Department of Computer Science, University of Illinois at Urbana-Champaign

We present a mathematical model of interacting neuron- like units that we call Input Feedback Networks (IFN). Our model is motivated by a new approach to biological neural networks, which contrasts with current approaches (e.g. Layered Neural Networks, Perceptron, etc.). Classification and reasoning in IFN are accomplished by an iterative algorithm, and learning changes only structure. Feature relevance is determined during classification. Thus it emphasizes network structure over edge weights. IFNs are more flexible than previous approaches. In particular, integration of a new node can affect the outcome of existing nodes without modifying their prior structure. IFN can produce informative responses to partial inputs or when the networks are extended to other tasks. It also enables recognition of complex entities (e.g. images) from parts. This new model is promising for future contributions to integrated human-level intelligent applications due to its flexibility, dynamics and structural similarity to natural neuronal networks.

Introduction

Regulation through feedback is a common theme found in biology including gene expression and physiological homeostasis. Furthermore, feedback connections can be found ubiquitously throughout neuronal networks of the brain. Yet the role of feedback in neuronal networks is often under-appreciated. Most connectionist models determine connection weights during a learning phase and during the testing phase employ a simple feedforward structure.

The contribution of this paper is to demonstrate that feedback employed during the test phase can perform powerful distributed processing. In our feedback model every output node only inhibits its own inputs. Thus the model is named *Input Feedback Networks* (IFN). This paper reinforces preliminary results of the IFN [1, 2] and explores its ability to perform recognition and intelligent inference.

Our analysis is divided into two subsections: combinatorial considerations and functional analysis. The combinatorial section discusses the implications of adding a new representation in various networks and the plausibility of large networks. The functional analysis section explores through exemplars how IFN behaves in complex scenarios.

The scenarios are: 1) when an output node with a small input vector completely overlaps with an output node composed of a larger input vector. Subsequently, the smaller vector is innately inhibited given the inputs of the larger vector. 2) Multiple overlapping representations can cooperate and compete under differing circumstances. 3) Network 'logic' can be adjusted by simply biasing activation of an output. 4)

Depending on inputs states, inference can be conducted through distributed processing over an infinite number of chains. 5) Overlap of representations determines processing difficulty and the role initial conditions may have on inference.

Some of these results are preliminary. However, it is clear that based on a simple premise of feedback, IFN offers a flexible structure and dynamic approach to classification of stimuli and Artificial Intelligence.

1. Background

Traditional connectionist classifier models can be broken down into two broad strategies: 1) *Neural Network* (NN) type algorithms that primarily rely on connection weight adjustments, implemented by learning algorithms. 2) *Lateral Competition* (LC) algorithms that involve competition between 'similar' representations. LC and NN methods can overlap. For example, weight adjustments occur between LC connections. We argue that both connection weights and direct output inhibition are specific to tasks. Furthermore they can be combinatorically implausible and can limit processing.

1.1. Connection Weight Adjustments

The idea of adjusting connection weights has been a corner stone throughout the development of connectionist models including Perceptrons, Parallel Distributed Networks, Markov Models, Bayesian Networks, Boltzmann Machines, and even Support Vector Machines [3-7]. 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, this approach suffers from difficulties such as over-fitting, local minima and catastrophic interference ('forgetting' of previously learned tasks, e.g. [8, 9]). The role of Neural Network (NN) structure in relation to function or image parts is unclear. Learning algorithms are difficult to describe in terms of biologically viable neural mechanisms. Lastly, recent studies of neuron processing, challenge the idea that a synapse can be estimated by a connection weight [10, 11].

1.2. Lateral Competition

In LC models cells inhibit their neighbors or their neighbors' inputs. Such models include: Hopfield, Winner-take-all (WTA) and Lateral Inhibitory networks i.e. [12-15]. In WTA networks are engineered so that a cell inhibits all other possible representations. Thus every cell must be connected to a common inhibitory network. This puts biologically implausible combinatorial demands on connectivity and limits parallel processing. WTA does not address how recognition of parts interacts with overall classification. Carpenter and Grossberg recognized this problem and proposed a mechanism which evaluates and ranks the number of inputs used in each part [12]. The output(s) encompassing the most inputs is chosen as the best solution. But this mechanism evaluates one cell at a time, and does not address when partial representations should compete.

Lateral Inhibition and Hopfield networks are relaxed versions of WTA where the amount of competition between cells is engineered with LC connection weights. Lateral inhibition was originally proposed to describe neuronal activity within the retina based on neighboring cells with simple spatial receptive fields. The closest neighbors compete most and competition decreases as cells are farther apart spatially (i.e. [15]). However lateral inhibitory and Hopfield connections become somewhat intractable as object representations become more complex because the number of possible competitors becomes huge. Let's take the pattern '1' for example, '1' can compete with the representation of letter 'I', vertical bars, other letters, other numbers or anything vertical.

2. Input Feedback Network Structure and Function

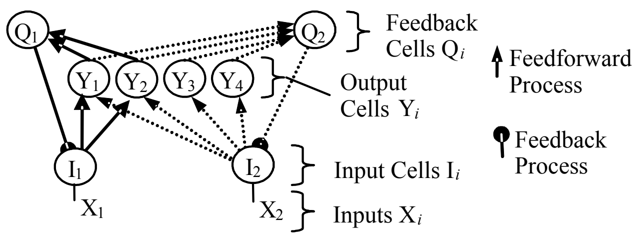


Figure 1. Input Feedback Schematic. Every feedforward connection has an associated feedback connection. If I_1(*e.g. white*) projects to Y_1(*e.g. pingpong*) & Y_2(*lychee*), then Q_1 must receive projections from Y_1 & Y_2 and provide feedback to the input cell I_1. Similarly if I_2 (*e.g. round*) projects to Y_1, Y_2, Y_3(*orange*), & Y_4(*planet*), then Q_2 receives projections from Y_1, Y_2, Y_3, & Y_4 and projects to I_2.

IFN is composed of simple binary connections between input features and output vectors. Yet, this becomes surprisingly powerful when combined with regulatory feedback to inputs during the testing phase [2]. This structure maintains its simplicity in large networks but can still make complex recognition decisions based on distributed processing.

The IFN structure is shown in Figure 1. The network delineates a simple rule of connectivity based on a triad of interconnections between an input, the output it supports, and feedback from that output. Every input has a corresponding feedback 'Q', which samples the output processes that the input cell activates and modulates the input amplitude. These triads are intermeshed independently of what is happening with other such triads of other inputs. The underlying triad holds true from both the perspective of the input processes and output processes. Every output must project to the feedback Q processes that correspond to the inputs the output receives. 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.

The networks are designed to dynamically re-evaluate each cell's activation based on 1) input feedback onto the individual inputs in order to 2) modify the input state based on the input's use. The input's use is inversely proportional to the output cells that the input activates. Lastly 3) re-evaluating each cell's activity based on its state and the re-evaluated input. Steps 1-3 are cycled through continuously as the network converges to its solution. Each cell inhibits only its inputs based on input feedback. Feedback provides a continuous measure of the use of each input, which determines competition [1, 2, 16, 17].

2.1. Dynamic Evaluation of Ambiguity

Inputs that project to multiple simultaneously active output nodes are ambiguous. They are ambiguous because many representations use them. Such inputs hold little value and become inhibited. In contrast, inputs that project to one non-active representation are boosted.

The only way an output cell process can receive full activation from the input is if it is the only cell active using that input, reducing Q to 1. If two competing cells share similar inputs, they inhibit each other at the common inputs, forcing the outcome of competition to rely on other non-overlapping inputs. The more overlapping inputs two cells have, the more competition exists between them. The less overlap between two cells, the less competition, more 'parallel' or independent from each other the cells can be. This is a trend supported by human search efficiency data that compares similarity between search objects and reaction times [18].

2.2. Simple Connectivity

Feedback networks do not require a vast number of connections; the number of connections required for competition 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 about its inputs and not directly connect with the other output cells. Thus the number of connections of a specific cell in feedback competition is independent of the size or composition of the classification network, allowing large and complex feedback networks to be combinatorially and biologically practical.

2.3. Flexibility

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.

2.4. Neurophysiology & Neuroanatomy Evidence

The ability to modify information in earlier pathways appears to be an important component of recognition processing. Input feedback connections can be found in virtually all higher areas of brain processing. The thalamic system (including thalamic nuclei the Lateral Geniculate Nucleus-LGN, and Medial Geniculate Nucleus-MGN) projects to all of the cerebral cortex and receives extensive cortical innervations back to the same processing areas. Relay cells feed forward to the cortex and to pyramidal cells which feed back from the cortex. Together they form the ubiquitous triad structure of input feedback. The thalamus may even receive more feedback connections from the cortex than 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 receives inputs through the mitral/tufted cells, within the next structure, the olfactory cortex and projects information back to granule cells within the OB (another Q cell equivalent) which inhibit the mitral/tufted cells.

Nonlinear mechanisms which alter cell activation based on previous activity are found in dendritic-dentritic 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) form an input feedback system of inhibition [23, 24].

2.5. Mathematical Formulation and Investigation

This section introduces general nonlinear equations governing IFN. For any cell/vector Y denoted by index a, let N_a denote the input connections to cell Y_a. For any input edge I denoted by index b, let M_b denote the feedback connections to input I_b. The amount of shunting inhibition at a single input I_b is defined as Q_b for that input. Q_b, is a function of the sum of activity from all cells Y_j that receive activation from that input:

$$Q_b = \sum_{j\in M_b} Y_j(t) \qquad \textit{Eq 1} \qquad\qquad I_b = \frac{X_b}{Q_b} \qquad \textit{Eq 2}$$

Input I_b is regulated based on the 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. The activity of Y_a is dependent on its previous activity and the input cells that project to it. The input activity that is transferred to the output cells is inversely proportional to the Q feedback.

$$Y_a(t+\Delta t) = \frac{Y_a(t)}{n_a}\sum_{i\in N_a} I_i \qquad \textit{Eq 3}$$

$$= \frac{Y_a(t)}{n_a}\sum_{i\in N_a}\frac{X_i}{Q_i} = \frac{Y_a(t)}{n_a}\sum_{i\in N_a}\left(\frac{X_i}{\sum\limits_{j\in M_i} Y_j(t)}\right) \qquad \textit{Eq 4}$$

2.6. Stability

Stability of these equations presented here and variations including negative values have been previously analyzed [25, 26]. The subset used here are limited to positive values of all variables, thus these equations will always be positive given positive values of the components. Thus the values of Y can not become negative and have a lower bound of *0*.

Furthermore the values have an upper bound. By definition since N_a is the set of input connections to cell Y_a, then M_b will contain cell Y_a the within the set of input feedback connections to input I_b. To achieve the largest possible value, all other cells should go to zero. In that case, the equation then reduces to

$$Y_a(t+\Delta t) \le \frac{1}{n_a}\sum_{i\in N_a}\left(\frac{Y_a(t)\cdot X_i}{Y_a(t)}\right) = \frac{1}{n_a}\sum_{i\in N_a} X_i \le \frac{X\max\cdot n_a}{n_a} = X\max \qquad \textit{Eq 5}$$

where n_a is the number of processes in set N_a. If X values are bounded by an *Xmax* then the values of Y are bounded by positive numbers between zero and *Xmax*.

The Picard existence and uniqueness theorem states that if a differential equation is bounded and is well behaved locally then will have a unique solution i.e. [27].

2.7. Learning

The distinguishing feature of this approach is the emphasis on the test phase as opposed to the conventional emphasis on learning. Thus, we purposefully deemphasize learning. Since IFN uses simple positive binary connections, learning is simpler. This means that only features present during label presentation are encoded. Negative associations such as *'Y_1' is unlikely given feature 'X_1'*, are not encoded. Instead, they are estimated using the recursive feedback processes. Thus learning can involve encoding simple (Hebbian-like) correlations between input features and output vector labels. More sophisticated learning methods can include clustering and pruning ie. [28].

3. Combinatorial Analysis

In this section we outline combinatorial problems of adding a new node to large networks and describe how IFN structure avoids these problems.

3.1. The Ping-Pong, Peeled Lychee Example

Assume three huge brains composed of each type of network (NN, LC, IFN) and recognizes among other things a ping-pong ball with a 'ping-pong ball' cell [29]. Now a new stimulus appears for the first time: a peeled lychee. It has many similar input features to a ping-pong ball (ie. color and size). Each model should 1) incorporate a lychee node 2) assure the 'ping pong' node does not predominate given lychee input features.

In all of the brains the input features representative of peeled lychee must connect to the node representative of lychee. In NNs the whole network may potentially be adjusted. Given lychee the connection weights of the features that support lychee are boosted and those that support ping-pong are reduced. Similarly in LC given lychee, weighted lateral connections must be created where the representation of lychee inhibits the representation of ping-pong (and any other similar representation) and visa versa. No matter how obscure or unlikely the relation of lychees and ping-pong balls are in most life situations NNs and LC networks must re-arrange their structure for it. This can potentially degrade previously learned associations [8, 9]. As the networks become larger, the cost of evaluating each node against the others becomes impractical. In IFN the node's input-output relations are sufficient. Thus no further modification of the structure is required.

In summary, in LC the number of connections can be implausible: potentially every node may be required to connect to every other node (WTA). In weight adjustment paradigms (NNs), connection weights are allowed a high degree of freedom and need to be adjusted for all tasks a-priori. IFNs rely on input feedback instead of direct connections or weights and do not require modification. Thus are more combinatorially practical.

4. Functional Analysis

We analyze interactions between different compositions of node representations and degenerate cases to assess their informative value. We analyze 1) how nodes with smaller representations that are overlapped by larger representations behave 2) how multiple partially overlapping representations can simulate a binding scenario, where certain parts cooperate and others compete, 3) control of overlapping behavior 4) the behavior of infinite number of partial overlapping representations linked in chains, and 5) winner-less competition.

4.1. Composition by Overlap of Nodes

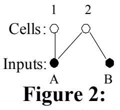


Figure 2: Example 1

In the most general example, example 1, two cells are connected such that the inputs of Y_1 (input A) completely overlaps with a larger Y_2. But Y_2 also receives an independent input, B [2]. In IFN, Y_1 has one input connection, thus by definition its input connection weight is one. It is 'Homeostatic': its maximal activity is *1* when all of its inputs (input A) are *1*. Y_2 has two input connections so for it to be 'Homeostatic' its input connection weights are one-half. When both inputs (A & B) are *1* the activity of Y_2 sums to *1*. Note these weights are predetermined by the network. Connections are permanent and never adjusted. Input *A* projects to both Y_1 & Y_2, thus receives inhibitory feedback from both Y_1 & Y_2. Input *B* projects only to Y_2 so it receives inhibitory feedback from Y_2. The most encompassing representation will predominate without any special mechanism to adjust the weighting scheme. Thus, if inputs A and B are active Y_2 wins. This occurs because when both inputs are active, Y_1 must compete for all of its inputs with Y_2, however Y_2 only needs to compete for half of its inputs (the input shared with Y_1) and it gets the other half 'free'. This allows Y_2 to build up more activity and in doing so inhibit Y_1.

The solutions are presented as (*input values*)→(*output vectors*) in the pattern $(X_A,X_B,)\rightarrow(Y_1,Y_2)$. The steady state solution for example 1 is $(X_A, X_B)\rightarrow(X_A - X_B, X_B)$. Substituting our input values we get (1,1)→(0,1), (1,0)→(1,0). Given only input *A* the smaller cell wins the competition for representation. Given both inputs the larger cell wins the competition for representation. Though we used binary inputs, the solution is defined for any positive real *X* input values [2]. The mathematical equations and their derivation can be found in the Appendix.

Thus smaller representation completely encompassed by a larger representation become is inhibited when the inputs of the larger one are present. The smaller representation is unlikely given features specific only to the large representation. It demonstrates that IFN determines negative associations ('Y_1' *is unlikely given feature* 'X_B') even though they are not directly encoded. This is a general example and data sets may have many forms of overlap.

In order to encode such negative associations using conventional methods, they would have to be 'hard-wired' into the network. With NN, each possible set of stimuli combinations would have to be trained. With LC each possible negative association would have to be explicitly connected.

4.2. Multiple Parts in Binding Scenario

IFN can simultaneously evaluate the criteria of three different representations. In e.g. 2, expanded from e.g. 1, three cells partially overlap. As in e.g. 1, Y_1 competes for its single input with Y_2. However, now Y_2 competes for its other input with Y_3 and Y_3 competes for only one of its inputs.

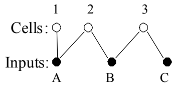


Figure 3: Example 2

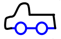
The steady state solution is $(X_A, X_B, X_C) \rightarrow (X_A – X_B + X_C, X_B – X_C, X_C)$. If $X_B \le X_C$ then $Y_2 = 0$ and the equations become $(X_A\ , 0, \frac{X_B + X_C}{2})$. If $X_C = 0$ the solution becomes that of e.g. 1: $(X_A, X_B, 0) \rightarrow (X_A - X_B, X_B, 0)$. The results are: (1, 0, 0) → (1, 0, 0);

(1, 1, 0) → (0, 1,0); (1, 1, 1) → (1, 0, 1). Derivations can be found in the appendix.

Thus if input A is active, Y_1 wins. If inputs A and B are active and Y_2 wins for the same reasons this occurs in e.g 1. However, if inputs A, B and C are active then Y_1 and Y_3 win. The network as a whole chooses the cell or cells that best represent the input pattern with the least amount of competitive overlap.

In e.g. 2, Y_2 must compete with all of its inputs: A with Y_1, B with Y_3. Y_3 only competes for half of its inputs (input B) getting input C 'free'. Since Y_2 is not getting its other input (input A) 'free' it is at a competitive disadvantage to Y_3. Together Y_1 and Y_3, mutually benefit from each other and force Y_2 out of competition. Competitive information travels indirectly 'through' the representations. Given active inputs A and B, the activity state of Y_1 is determined by input C through Y_3. If input C is 0 then Y_1 becomes inactive. If input C is 1, Y_1 becomes active. However, Y_3 does not even share input A with Y_1.

4.2.1. Binding

Choosing Y_2 given inputs A, B, and C is equivalent to choosing the irrelevant features for binding. Below we attempt to put this issue in a 'binding' context. Lets assign inputs A, B, and C to represent spatially invariant features of an image of a pickup truck  where feature A represents circles, C represents the truck body (without chassis and wheels), and feature B represents a horizontal bar. Y_1 is assigned to represent wheels and thus when it is active, feature A is interpreted as wheels. Y_2 represents a barbell composed of a bar adjacent to two round weights (features A and B). Note: even though Y_2 includes circles (feature A), they do not represent wheels (Y_1), they represent barbell weights. Thus if Y_2 is active feature A is interpreted as part of the barbell. Y_3 represents a pickup truck body without wheels (features B and C), where feature B is interpreted as part of the truck chassis. Now given an image of a pickup, all features simultaneously (A, B and C), choosing the barbell (Y_2) even though technically a correct representation, is equivalent to a binding error within the wrong context in light of all of the inputs. In that case the complete picture is not analyzed in terms of the best fit given all of the information present. Similar to case 1, the most encompassing representations mutually predominate without any special mechanism to adjust the weighting scheme.

Thus the networks are able to evaluate and bind representations in a sensible manner for these triple cell combinations. To emulate this in traditional network each possible combination will have to be trained for each input combination.

4.3. Search

The previous section demonstrated that these networks display useful distributed properties. In this section we describe how the networks can be modified to perform tasks.

Suppose we want to use our network to 'search' for specific stimuli. Object-specific neurons in the temporal lobe show biasing (a slight but constant increase in baseline activity of that cell) when the animal is trained to actively look for that shape. During recognition, the biased cell rapidly gains activity at the expense of others [30]. Our network can be biased in a similar fashion. We repeat example 2 but want to ask the question: can a barbell shape be present? We introduce a small bias to Y_2 (representing barbell) according to the equation:

$$Y_2(t+dt)=\frac{Y_2(t)}{2}\left(\frac{X_A}{Y_1(t)+Y_2(t)}+\frac{X_B}{Y_2(t)+Y_3(t)}\right)+b$$

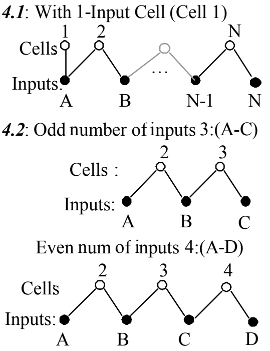
Choose a bias b of 0.2 and activating all inputs (1, 1, 1) → (*0.02, 0.98, 0.71*). The network now overrides its inherent properties and indicates that the inputs of Y_2 are present.

Thus network function can be radically adjusted simply by biasing activation. Most importantly, this did not require adjustment of any connection weights. Biasing involved only the desired representation and can be easily turned on and off. In traditional methods such as NNs weights would need to be re-learned and redistributed throughout the network for this new task.

4.4. Composition by Infinite Chains

Cells can be further linked at infinitum and the cell representations interact indirectly by transferring their dynamic activation through the chain.

Figure 4:
Chains of 2-Input Cells



4.4.1: With 1-Input Cell (Cell 1)

4.4.2: Odd number of inputs 3:(A-C)

Even num of inputs 4:(A-D)

4.4.1. Including a 1-Input cell

Consider the for example case where there are N cells, and N inputs, and all inputs are 1. If N is an *odd* number then at steady state the *odd* numbered cells will be 1 and even ones 0. If N is *even*, the *even* cells will be 1 and *odd* ones zero. The general solutions (Y_1, Y_2..., Y_N) where i & j represent cell numbers are:

$$\left(\sum_{1\le j\le N} X_{odd}-X_{even},\ \sum_{2\le j\le N} X_{even}-X_{odd},\ \sum_{i\le j\le N} X_{even}-X_{odd},X_N\right)$$

For example with N=4: (1,1,1,1) → (0,1,1) and N=5: (1,1,1,1,1) → (1,0,1,1). Thus these input configurations can be represented by binary values.

4.4.2. Without a 1-Input cell

If cell 1 (the only cell with one input) is not present, then the network does not have a favorable set of cells to resolve an odd number of inputs. Two-input cells can not match in a binary manner the inputs of an odd numbered chain.

Thus, in these cases the solution becomes more complicated. In the case of three inputs (covered by two cells) the mathematical solutions are:

$$Y_1 = \frac{X_A(X_A + X_B + X_C)}{2(X_A + X_C)} \qquad Y_2 = \frac{X_C(X_A + X_B + X_C)}{2(X_A + X_C)}.$$

When inputs are (1,1,1) then output cells become (¾, ¾). Furthermore if only the middle input is active (0,1,0) then the forces on both cells are symmetrical, the equation collapses to $2(Y_1(t)+Y_2(t))=X_B$ and the solution depends on initial conditions (also see section *5.5*).

In case of 4 inputs distributed over 3 cells the solution becomes:

$$(\frac{X_A(\Sigma X)}{2(X_A + X_C)}, \frac{-(\Sigma X)(X_A X_D - X_C X_B)}{2(X_A + X_C)(X_B + X_D)}, \frac{X_D(\Sigma X)}{2(X_B + X_D)}) \quad \text{Where } \Sigma X = X_A + X_B + X_C + X_D.$$

When all inputs are *1*, the cells settle on a binary solution (1,0,1). Cases with more inputs become progressively more complicated. Thus the structure can greatly affect the ability to efficiently represent inputs.

4.4.3. Subchains

If any input in the chain is zero, this will break the chain into independent components composed of the right and left parts of the chain from the zero input. These can function as smaller chains. For example if input D=*0*, the chains involving inputs A-C and E-N become independent. Thus the outcome of network, is determined by distributed cell dynamics involving input values and cell representations. Further analysis is remains for future research.

4.5. Analysis of Node Overlap

Lastly overlap is a key determinant of processing independence [2]. If two representations completely overlap they may also be dependent on initial conditions of the cells. Suppose there exists only two cells Y_1 & Y_2 with n_{1indep} and n_{2indep} representing each cells' independent inputs. Furthermore K_1 & K_2 represent the average input values to these independent inputs. The steady state solution is of the form:

$$k_{initial} \frac{Y_1^{\,n_1}}{Y_2^{\,n_2}} = e^{(n_{2\,Indep} - n_{1\,Indep} + \frac{n_{1\,Indep}K_1}{Y_1} - \frac{n_{2\,Indep}K_2}{Y_2})t} = e^{\lambda t} \ ,$$

where $k_{initial}$ represents initial conditions of the network. See derivations in Appendix.

If $\lambda > 0$ then $Y_1(t=\infty) \rightarrow 0$, if $\lambda < 0$ then $Y_2(t=\infty) \rightarrow 0$. In these cases either Y_1 or Y_2 predominates. However, if $\lambda = 0$ then the solution is not sufficiently independent and is a function with the form $Y_2 = k\ Y_1$. The initial value of *k* will affect the final solution. Further analysis is the topic of future research.

5. Conclusion

With simple combinatorially-plausible binary relations input feedback offers a flexible and dynamic approach to intelligent applications. It is well suited for classification of stimuli composed of parts because its distributed interactions can resolve overlap of multiple representations. These interactions give preference to the broadest representations and can determine when representations should cooperate or compete. But, if representations are not sufficiently independent they may depend on initial conditions. However, the network can also be manipulated to perform search tasks by biasing output representations. These properties demonstrate that such networks can be an integral part of intelligent inference and provide a new direction for future research.

Appendix

Section 5.1, example 1 IFN equations are:

$$Y_1(t+dt)=\frac{Y_1(t)X_A}{Y_1(t)+Y_2(t)} \quad , \quad Y_2(t+dt)=\frac{Y_2(t)}{2}\left(\frac{X_A}{Y_1(t)+Y_2(t)}+\frac{X_B}{Y_2(t)}\right).$$

The network solution at steady state is derived by setting $Y_1(t+dt)=Y_1(t)$ and $Y_2(t+dt)=Y_2(t)$ and solving these equations. The solutions are $Y_1 = X_A - X_B$ and $Y_2 = X_B$. If $X_A \le X_B$ then $Y_1 = 0$ and the equation for Y_2 becomes: $Y_2 = \frac{X_A + X_B}{2}$.

Section 5.2, example 2 equations: Y_1(t+dt) remains the same as example 1. $Y_2(t+dt)=\frac{Y_2(t)}{2}\left(\frac{X_A}{Y_1(t)+Y_2(t)}+\frac{X_B}{Y_2(t)+Y_3(t)}\right)$, $Y_3(t+dt)=\frac{Y_3(t)}{2}\left(\frac{X_B}{Y_2(t)+Y_3(t)}+\frac{X_C}{Y_3(t)}\right)$

Solving for steady state by setting $Y_1(t+dt)=Y_1(t)$, $Y_2(t+dt)=Y_2(t)$, and $Y_3(t+dt)=Y_3(t)$, we get $Y_1=X_A-X_B+X_C$, $Y_2=X_B-X_C$, $Y_3= X_C$. If $X_C=0$ the solution becomes that of e.g. 1: $Y_1=X_A-X_B$ and $Y_2=X_B$. If $X_B \le X_C$ then $Y_2=0$ and the equations become $Y_1 = X_A$ and $Y_3 = \frac{X_B + X_C}{2}$.

Section 5.5: The overlap region is defined by the number of inputs that overlap between two cells N_{over}, and the number of inputs that are independent of overlap N_{indep}. If Y_1 & Y_2 overlap then N_{over} of Y_1 = N_{over} of Y_2. Thus $n_1=n_{1Indep}+n_{over}$ and $n_2=n_{2Indep}+n_{over}$. Thus K_{over} and N_{over} of cell 1 = K_{over} and N_{over} of Y_2 by definition of overlap:

$$Y_1(t+dt)=\frac{Y_1(t)}{n_1}\left(\frac{n_{1Indep}K_1}{Y_1(t)}+\frac{n_{over}K_{over}}{Y_2(t)+Y_1(t)}\right) \text{ and } Y_2(t+dt)=\frac{Y_2(t)}{n_2}\left(\frac{n_{2Indep}K_2}{Y_2(t)}+\frac{n_{over}K_{over}}{Y_1(t)+Y_2(t)}\right).$$

Substituting $Y(t+dt)=Y+Y'$ and subtracting out the overlap region the equations reduce to: $n_1\frac{Y_1'}{Y_1}-n_2\frac{Y_2'}{Y_2}=n_2-n_1+\frac{n_{1Indep}K_1}{Y_1}-\frac{n_{2Indep}K_2}{Y_2}$. Integrating and raising exponents of both sides we get: $k_{initial}\frac{Y_1^{n_1}}{Y_2^{n_2}}=e^{(n_{2Indep}-n_{1Indep}+\frac{n_{1Indep}K_1}{Y_1}-\frac{n_{2Indep}K_2}{Y_2})t}=e^{\lambda t}$.

Acknowledgements

We would like to thank Burak M. Erdogan, Frances R. Wang and anonymous reviewers for helpful suggestions. This work was supported by the U.S. National Geospatial Agency Grant HM1582-06--BAA-0001.

References

[1] Achler, T., *Input shunt networks.* Neurocomputing, 2002. **44**: p. 249-255.

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

[3] Hinton, G.E. and T.J. Sejnowski, *Unsupervised learning : foundations of neural computation*. Computational neuroscience. 1999, Cambridge, Mass.: MIT Press. xvi, 398 p.

[4] Minsky, M.L. and S. Papert, *Perceptrons; an introduction to computational geometry*. 1969, Cambridge, Mass.,: MIT Press. 258 p.

[5] Rosenblatt, F., *The perceptron: a probabilistic model for information storage and organization in the brain.* Psychol Rev, 1958. **65**(6): p. 386-408.

[6] Rumelhart, D.E., J.L. McClelland, and University of California San Diego. PDP Research Group., *Parallel distributed processing : explorations in the microstructure of cognition*. Computational models of cognition and perception. 1986, Cambridge, Mass.: MIT Press.

[7] Vapnik, V.N., *The nature of statistical learning theory*. 1995, New York: Springer. xv, 188 p.

[8] Lewandowsky, S., *Catastrophic Interference and Generalization in Neural Networks.* International Journal of Psychology, 1992. **27**(3-4): p. 653-653.

[9] Rueckl, J.C., *Reducing Catastrophic Interference through Weight Modulation.* Bulletin of the Psychonomic Society, 1992. **30**(6): p. 457-457.

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

[11] Turrigiano, G.G. and S.B. Nelson, *Homeostatic plasticity in the developing nervous system.* Nat Rev Neurosci, 2004. **5**(2): p. 97-107.

[12] Carpenter, G.A. and S. Grossberg, *A Massively Parallel Architecture for a Self-Organizing Neural Pattern-Recognition Machine.* Computer Vision Graphics and Image Processing, 1987. **37**(1): p. 54-115.

[13] Douglas, R.J. and K.A. Martin, *Neuronal circuits of the neocortex.* Annu Rev Neurosci, 2004. **27**: p. 419-51.

[14] Hopfield, J.J., *Neural Networks and Physical Systems with Emergent Collective Computational Abilities.* Proceedings of the National Academy of Sciences of the United States of America-Biological Sciences, 1982. **79**(8): p. 2554-2558.

[15] Kuffler, S.W., *Discharge patterns and functional organization of mammalian retina.* J Neurophysiol, 1953. **16**(1): p. 37-68.

[16] Nigrin, A., *Neural networks for pattern recognition*. 1993, Cambridge, Mass.: MIT Press. xvii, 413 p.

[17] Reggia, J.A., et al., *A Competitive Distribution-Theory of Neocortical Dynamics.* Neural Computation, 1992. **4**(3): p. 287-317.

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

[19] LaBerge, D., *Attention, awareness, and the triangular circuit.* Conscious Cogn, 1997. **6**(2-3): p. 149-81.

[20] Atema, J., *Distribution of Chemical Stimuli*, in *Sensory biology of aquatic animals*, J. Atema, Editor. 1988, Springer-Verlag: New York. p. p 29-56 xxxvi, 936 p.

[21] Chen, W.R., W. Xiong, and G.M. Shepherd, *Analysis of relations between NMDA receptors and GABA release at olfactory bulb reciprocal synapses.* Neuron, 2000. **25**(3): p. 625-33.

[22] Aroniadou-Anderjaska, V., et al., *Tonic and synaptically evoked presynaptic inhibition of sensory input to the rat olfactory bulb via GABA(B) heteroreceptors.* J Neurophysiol, 2000. **84**(3): p. 1194-203.

[23] Isaacson, J.S. and B.W. Strowbridge, *Olfactory reciprocal synapses: Dendritic signaling in the CNS.* Neuron, 1998. **20**(4): p. 749-761.

[24] Lowe, G., *Inhibition of backpropagating action potentials in mitral cell secondary dendrites.* J Neurophysiol, 2002. **88**(1): p. 64-85.

[25] Mcfadden, F.E., *Convergence of Competitive Activation Models Based on Virtual Lateral Inhibition.* Neural Networks, 1995. **8**(6): p. 865-875.

[26] Mcfadden, F.E., Y. Peng, and J.A. Reggia, *Local Conditions for Phase-Transitions in Neural Networks with Variable Connection Strengths.* Neural Networks, 1993. **6**(5): p. 667-676.

[27] Nagle, R.K., E.B. Saff, and A.D. Snider, *Fundamentals of differential equations and boundary value problems*. 5th ed. 2008, Boston: Pearson Addison-Wesley.

[28] J. S. Sanchez, L.I.K. *Data reduction using classifier ensembles*. in *Proc. 11th European Symposium on Artificial Neural Networks*. 2007.

[29] Quiroga, R.Q., et al., *Invariant visual representation by single neurons in the human brain.* Nature, 2005. **435**(7045): p. 1102-7.

[30] Chelazzi, L., et al., *Responses of neurons in inferior temporal cortex during memory-guided visual search.* J Neurophysiol, 1998. **80**(6): p. 2918-40.