2007

An analysis of connectivity

Tayeb Al Karim

An Analysis of Connectivity

Master of Science Thesis

Tayeb Al Karim
Rochester Institute of Technology
Department of Computer Science
102 Lomb Memorial Drive
Rochester, New York 14623
USA

August 6, 2007

Signatures

I, Tayeb Al Karim, submit this thesis in partial fulfillment of the requirements for the degree of Master of Science in Computer Science. It is approved by the committee members below.

Tayeb Al Karim

Dr. Roger S. Gaborski
Committee Chair

Dr. Zack Butler
Reader

Dr. Leon Reznik
Observer

Acknowledgments

First of all, thanks to everyone who I bugged while working on this Thesis. Being able to bounce ideas off of people, rant about how things weren't working, and getting people to respond and tell me how things wouldn't work made it possible to finally get something that actually does work. Thanks to Dr. Zack Butler, Dr. Roger Gaborski, TJ, the Johns, and RJ for reading through my thesis and offering advice as to how things could be explained and written better. A big thanks to Chris, who did both the above, but also had to put up with being the sieve through which I filtered all my ideas before telling them to anyone else.

Abstract

Recent evidence in biology indicates crossmodal, which is to say information sharing between the different senses, influences in the brain. This helps to explain such phenomenon as the McGurk effect, where even though a person knows that he is seeing the lip movement "GA" and is hearing the sound "BA", the person usually can't help but think that they are hearing the sound "DA". The McGurk effect is an example of where the visual sense influences the perception of the audio sense. These discoveries transition old feedforward models of the brain to ones that rely on feedback connections and, more recently, crossmodal connections. Although we have many software systems that rely on some form of intelligence, i.e. person recognition software, speech to text software, etc, very few take advantage of crossmodal influences.

This thesis provides an analysis of the importance of connections between explicit modalities in a recurrent neural network model. Each modality is represented as an individual recurrent neural network. The connections between the modalities and the modalities themselves are trained by applying a genetic algorithm to generate a population of the full model to solve certain types of classification problems.

The main contribution of this work is to experimentally show the relative importance of feedback and crossmodal connections. From this it can be argued that the utilization of crossmodal information at an earlier stage of decision making can boost the accuracy and reliability of intelligent systems.

Contents

List of Figures

Chapter 1

Introduction

1.1 Multimodal Integration

1.1.1 Motivation and Prior Work

The emergence of cheap, powerful, and unobtrusive sensors has enabled the creation of context aware systems. In being context aware, these systems can react to the environment and behave in the correct manner for a given situation (A. Pnevmatikakis and Polymenakos, 2006a). In order to correctly judge the context of a scene (by deciding if a certain person is in the scene or not, for example), data from multiple senses can be utilized. The use of multiple sources of data allow for robustness in cases where a certain sensor's data becomes noisy, sensors fail, certain sensors have better information than others. The challenge in using multiple data sources is in finding a way to handle what becomes a massive amount of data.

The identification problem

The term "identification". in biometrics, can mean one of two things. Either we wish to verify a person's identity or we wish to classify a person with her identity. The verification problem is a binary one; either the person is who she claims she is or she is not. The classification problem is more complex (Fox and Reilly, 2004; Veeramachaneni et al., 2003).

There are two types of classification problems, closed set classification and open set classification. In the closed set classification problem, the person presented can be any of a certain number of people (R) that the classifier knows about. In the open set classification problem, the person presented may or may not be in the set of people about which the classifier

knows. In the open set case, we treat the classifier as having to choose between $R + 1$ classifications. The $+1$ comes from the idea that the result "not in the set" should be a valid one. The classification problem can be seen as an extension of the verification problem in that we can treat a classification among R individuals as R separate verification tasks, the most positive result of which we can consider our correct class (Veeramachaneni et al., 2003).

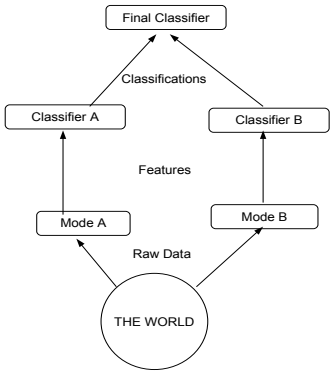


Figure 1.1: Late Integration

Classification using Multimodal Integration

Multimodal integration, or fusion, can be accomplished at an early stage of classification (so-called feature integration) or at a later stage (semantic or score integration) (Wu et al., 1999; Fox and Reilly, 2004; E. et al., 1998). In late integration, each modality performs the classification step on its own and these classifications are all taken into account when performing the final classification (See figure 1.1). We can use the unimodal analysis methods that have been previously developed and tested, thereby cutting down on research and implementation time. One thing to note would be that late integration does not take into account any correlations between the signals taken from the different modalities (Wu et al., 1999). When combining the classifications, each modality is weighted differently according to how much trust is put into the individual modality. These weights, as will be seen

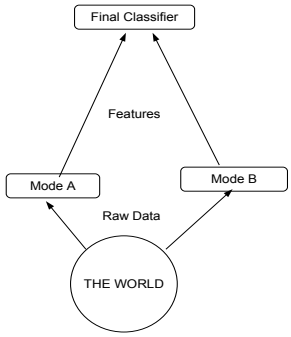


Figure 1.2: Early Integration with fused features

later, can be static and empirically determined or can be found via a search using particle swarms, genetic algorithms (more on this in section 1.1.3), or statistically.

Early integration can be thought of in two ways. The first is when the feature vectors produced by each modality are simply concatenated together and pushed through a classifier (See figure 1.2). This method, although potentially taking advantage of the correlation between different modalities, suffers from what has been labeled the "curse of dimensionality" (Wu et al., 1999; Fox and Reilly, 2004). This refers to the fact that while the feature space grows linearly the run time for the classification algorithms generally increase exponentially, which may cause the problem to become intractable. Another issue faced by this method of integration is that the frame rates for each modality are not necessarily in sync. This method also does not take into account the reliability of each modality, bad signals may potentially skew classification results. Finally, it may be the case that one modality may have more features than the others and therefore be given more weight implicitly.

Another way to think of early integration is that the modalities may influence the feature extraction of other modalities (See figure 1.3). An extension of this idea is that, if a modality were to classify its own input (as in late integration), it would take the other modalities' raw information

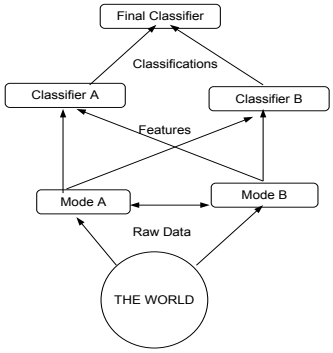


Figure 1.3: Early Integration with influenced features

into account. This method has the advantage of potentially using the correlations between modalities to reduce noise in the signals (which is most likely uncorrelated (Michalowski and Simmons, 2006)) and to make better features/judgments without falling into the trap of searching through an extremely high dimensional space. Using other modalities in extracting one modality's features may also give a way to make that modality more trustworthy (by virtue of cleaning up the features, for example).

Biological Inspiration

What does biology have to say? We have, until recently, been more attuned to the idea that our brains work by taking our basic sense data and moving it up through our brains' pathways into more complex ideas in a unimodal sense. It was thought that only after the sense data had become general enough that the different modalities would begin to interact. Recently, evidence has been uncovered that different senses interact with each other much earlier in the pathway. (Meredith, 2002) gives a good overview of multimodal convergence. Of note would be that the convergence of different modalities into the same region does not show true multimodal convergence, as unless the neurons can be shown to influence each other, the signals can not be said to have converged. As evidence for neuron-level convergence is hard to find, little is known about multimodal convergence. A lot of what is known

about multimodal convergence is known through the study of the superior colliculus. The upper three layers obtain information from the retina and primary visual cortex. The lower three layers also obtain information from the non primary visual areas, maps of the body surface, and the auditory space. Because of the multimodal nature of this area, many neurons exhibit influences from multiple modalities. Findings have also shown that there is evidence that eye position modulates auditory responses as early as in A1 (Meredith, 2004).

Evidence of early multimodal integration is given by (Schroeder and Foxe, 2005). Recent findings throw into question the idea that multisensory convergence is a high level process that occurs after sensory data from each modality is processed separately. The authors report somatosensory and visual inputs in regions posterior to A1. One possible functional result of this are that the somatosensory and visual information may allow for auditory localization. Another is that the information primes the auditory sense, resetting it such that it can examine its data for a particular sound. For example, we expect to hear the strike of a nail when we see a hammer coming down.

(Joassina et al., 2004) studies interference effects, when one sense interferes with the perception of another sense such as described in the McGurk effect previously. The authors suggest that the auditory information could influence the processing of visual information in the fusiform gyrus around 90 - 130 milliseconds, the visual information would effect the auditory information in the associative auditory area around 140 - 200 milliseconds, and finally, from 260 - 280 milliseconds, the unimodal and crossmodal areas could interact with the semantic areas. One thing to take into account would be that normal reaction time, time it takes for an observer to react to a simple stimulus, is around 200 - 300 milliseconds. This suggests that multimodal influences are in effect before an item is recognized. Another suggestion put forth by the authors is that the senses would send projections to the superior colliculli, where the modalities are integrated and connections are sent back to modulate the primary sensory areas.

A similar view is taken by (Macaluso and Driver, 2005), where the authors suggest areas of feedforward multimodal convergence (late integration) in the parietal cortex, temporal cortex, frontal cortex, superior colliculus, the basal ganglia, and the putamen. The idea is then put forward that the feedforward convergence is not the whole story, as there is evidence of multimodal cells in areas previously thought of as unimodal, such as the visual cortex and somatosensory cortex.

In (Gardner, 1974; Gruber and Vaneche, 1977; Piaget, 1952), Piaget

argues that the first stage of learning is that of discovering the relationship between what a person does and what happens to his senses after doing - the development of the sensorimotor system. In this way, a person learns how to move and control his limbs, certain habits, coordination between different senses, and finally logic (discovery of means). (Piaget, 1952) states that motors should be treated as senses instead of something separate.

Implementation of Multimodal Systems

A framework for simulating a sensorimotor system is developed and explored in (Coen, 2006). Such a framework allows for the development of the first stage of learning - a relationship between the found classes in the sensory domains given to the system and the motor outputs of the system. The learning stages in (Coen, 2006) closely resemble Piaget's proposed stages. The organization of the different motor outputs and the senses are discovered on their own, the relationships between the different senses and motor outputs are then discovered, and finally, because the motor outputs are thought of as senses, by fixing the motor output to a certain value, the system can imagine what the senses would be. This effectively figures out how the system should change its surroundings to get a specific sensory reading, which draws a parallel to the discovery of means. (Coen, 2006)'s system also emphasizes the influence of one sense onto another early in the perceptual system, rather than in a post processing stage.

(A. Pnevmatikakis and Polymenakos, 2006b) discusses the merits of multimodal systems in context awareness for smart rooms, rooms that can react to a person's presence and commands. Adaptive background, shadow deletion, and gate algorithms are used in order to segment images of people in scenes. Direction of arrival and time delay estimation algorithms are used to figure out the direction and location of audio signals separately. (A. Pnevmatikakis and Polymenakos, 2006b) then defines contextual states as triggered by reaching target probabilities of all the perceptions for that specific context.

(Choudhury et al., 1998a) segments out faces and then classifies them using an eigenvalue based approach. Audio speaker identification is accomplished by use of Hidden Markov Models. Fusion is done by assigning confidence scores for audio and visual signals for each person to be identified and creating a Bayes net to represent this information.

(Erdogan et al., 2005b) uses gaussian mixture models to recognize the speaker in audio signals and the face in visual signals. Audio signals are modeled using mel frequency cepstra coefficients (MFCCs). Face data is

modeled by a PCA method. The authors further integrate driver signals to identify the person driving a car. Fusion is accomplished by applying different weights to each modality when attempting classification. To find the best weights, an exhaustive search is conducted over the training data.

(Coen, 2006; Vogt, 1998; Braun and Gero, 2006) all recognize the importance of self-training in learning systems. The ability to self-train allows the system to be more dynamic in that it is not constrained to data with which it may initially be trained with. (Coen, 2006; Vogt, 1998) also utilize perceptual grounding in their systems, which means that the systems discover meaning intrinsic to the data collected itself. By using perceptual grounding, the systems do not need prior information to understand the data fed in.

1.1.2 Classifier Types

The main classifiers used in the literature are Hidden Markov Models (HMMs), Gaussian Mixture Models (GMMs), Support Vector Machines (SVMs), and Vector Quantization (VQ). Each of these classifiers takes in as input a fixed length feature vector and outputs a classification for that feature vector (with an optional confidence level associated with that classification).

Hidden Markov Models allow us to probabilistically decide what the state of some sequence of feature vectors lie in. For example, given and audio signal with the spoken word "seven", we can determine if the word spoken was "seven" by checking if, after passing the audio signal through the HMM as a sequence of syllables, we reach a state where the maximum likely state is that of the spoken word "seven". In terms of classification, the person whose HMM outputs the largest value for the state "seven" is most likely to be the one that spoke the word "seven". Gaussian Mixture Models allow us to find the probability that the feature passed in is part of a specific (gaussian) model of a certain class.

The use of Hidden Markov Models and Gaussian Mixture Models share a few common traits. In general, one HMM or GMM is trained per individual per modality that the system is supposed to classify. A background HMM or GMM is also usually created. This background model is trained on all training data whereas the models used for each individual is trained with that individuals data. The background data is used during run-time to normalize the outputs of the individual HMMs or GMMs.

Support Vector Machines apply a structural risk minimization algorithm to the data set. This is a fancy way of saying that SVMs find the boundary least prone to given error for a given data set (yielding maximum separation

of classes). One thing to note about SVMs is that they output a binary result (Gutschoven and Verlinde, 2000) as it separates two classes of data.

Finally, the vector quantization algorithm generates a set of clusters from the data given to it. The centroids of the clusters are collectively labeled the codebook of the data set. When deciding what class to which a feature vector belongs, the data is matched against the centroids.

1.1.3 Late Integration Methods

In late integration, each modality has already made a guess as to the classification of the person presented to the system. As no modality is made equal, a weighted decision must be made as to the final classification of the individual. While these weights may be arrived at experimentally, it may be advantageous to have a system that automatically finds these weights. To this effect, some of the papers surveyed discuss how to automate the process of finding the weights. Their methods will be discussed presently.

Genetic Algorithms

Genetic Algorithms enable us to explore the weight-space in a stochastic hill climbing manner. If we consider an 'individual' to be a point in the weight space, we can define the fitness of that individual as how well the classification system performs using that individual's set of weights. Given a population of such individuals, the classical genetic algorithm operators (mutation and crossover) can be used to explore the weight space and, over time, yield increasingly better solutions.

Particle Swarm

Another method in which to find the weight vector is via the particle swarm algorithm. In this algorithm, much like in genetic algorithms, an individual is a point in the weight-space. The difference between particle swarms and genetic algorithms is that instead of applying crossover and mutation to explore the weight-space, an individual in the particle swarm tends to move toward its own personal best location found (which can be stored in a history of where the particular individual has been) and the global best location found (from the history of all individuals). This algorithm was used by (Veeramachaneni et al., 2003) for both feature extraction and fusion.

Statistical Analysis

The most common way of figuring out the weight vector is by doing a statistical analysis on the available data (Fox and Reilly, 2004; Wu et al., 1999). This can be done by methods such as analyzing the variance (Erdogan et al., 2005a) or the average distance to the mean (Falavigna and Brunelli, 1994).

General Methodology

In the literature, the most general methodology for late integration models is as follows:

→ read in data for each modality

→ construct (normalized) features for each modality

→ have each modality classify the person in question

→ find the appropriate weight for each modality

→ use modality weights and classifications to make final classification

When training and testing their classifiers, most papers generally added some noise into their data (Iwano et al., ; Veeramachaneni et al., 2003; Choudhury et al., 1998b). One of the papers trained on the clean data and tested on noisy data (Fox and Reilly, 2004). In general, for every 3 elements of training data,there was one element of test data.

In checking the value of multimodal systems, the papers reviewed compared them against the unimodal systems. Most papers, if they allowed more than two modalities, included classifications according to different combinations of the modalities. (Yang and Hauptmann, 2004) also compared the full multimodal classification results against random classification (literally choosing the class randomly) results.

1.1.4 Early Integration Methods

In early integration, the feature vectors from each modality are used to influence the other modalities' feature extraction method or classification. Early integration can also signify feeding the feature vectors in toto to one classification method, providing both visual and auditory feature vectors as the input into a standard feed forward neural network, for example.

Fuzzy Neural Network

The problem of classification can be redefined as a problem of finding the membership of an individual to the fuzzy sets "in the set" and "not in the set" (E. et al., 1998). A fuzzy neural network is a type of feed forward neural network that represent fuzzy rules. This network can develop a representation of these membership sets through a standard supervised training algorithm.

Self Organizing Maps

The idea of automatic feature extraction for one modality being influenced by features for another modality is recent in the field. As such, there is no literature on the topic in relation to person identification as of yet. However, work has started on creating algorithms and data types that represent this idea. (Coen, 2006) describes a structure, Slice, that classifies data given to it based on a self organizing algorithm that takes into account co-occurrences of activity in other modalities.

1.1.5 In Summary

The idea of multimodal systems and their various implementations have been explored. An explanation was made of the classification problem and how it relates to the person identification problem. The two different schemes of multimodal integration have been described, that of late integration and early integration. A discussion was made of the recent biological evidence towards multimodal integration. There has been a substantial amount of research done in searching the weight space for late integration. Research into early integration, in contrast, has not been explored as much. Also unexplored, prior to this thesis, is the usefulness of crossmodal influences in software systems, as compared to feedforward and feedback influences.

1.2 Neural Networks and Evolutionary Algorithms

1.2.1 Motivation and Prior Work

Evolutionary algorithms have been used to cover the weight space that needs to be searched in neural network training. The generation of intelligence for games is one environment in which Neural Networks trained with Evolutionary Algorithms has become well used. In one instance, players were evolved to test the balance of different rules in an ancient predecessor of

chess, Hnefatafl, to find which rule set was most probably used (Hingston, 2007). Among other games, Neural Networks and Evolutionary Algorithms have also been used to create master-level Othello players (Chong et al., 2005) and Tic Tac Toe players (Yau et al., 2007).

Consider a game like checkers. A standard way of programming an artificial player for the game would be to guide the search through the possible set of moves through the use of a heuristic function. These heuristic functions can quickly grow complex, making it hard to document and program. The functions may also require a high cost in tuning parameters such as the weights given to individual pieces and positions. In (Chellapilla and Fogel, 1999), the heuristic function in a standard minimax search was replaced by a feed forward neural network whose weights were found through an evolutionary algorithm. The evolutionary algorithm, without using any expert knowledge about how to play the game, was able to generate networks that were able to play at a master level. They argue that most heuristics provide a linear stimulus-response mapping, which is a limiting factor for most games. Neural Networks, however, can be used to find nonlinear mappings. The addition of using an evolutionary algorithm to train the neural networks provides a way to find solutions to games where no expert can be found and no data is supplied for training (the networks play against each other). The individuals found in such a fashion are distinctive, each with different styles of play and the ability to make mistakes - which end up being more fun for the end user. The authors quote Hofstadter on randomness;

> .. to a program that exploits randomness, all pathways are open, even if most have a very low probability; conversely, to a program whose choices are always made by consulting a fixed deterministic strategy, many pathways are *a priori* completely closed off. This means that many creative ideas will simply never get discovered...

(Thompson et al., 2007) used a sandbox game, EvoTanks to explore the dynamics of evolving AI players. In the game, one tank must destroy the other tank in a given time limit and with an unlimited amount of ammo. If neither tank gets destroyed, the game is considered a draw. Although the AI players found novel and interesting ways at defeating an opponent, the AI would be good only against that opponent. The intelligence for the game, like checkers, is heavily dependent on what the opponent will do. In EvoTanks, however, the set of possible states is exponentially larger and less predictable. Each tank is controlled by a 3 layer feed forward Neural Network, with the weights being found via a genetic algorithm. The authors

found that although training against scripted opponents produced better AI agents , co-evolved agents (agents train against other agents), when tested against the scripted opponents performed at reasonably strong levels. The authors argued that the co-evolution is the better choice when training for games as the method would allow a more complete search of the possible behaviors.

(Parker and Parker, 2007) conducted an similar experiment using the game Xpilot. In their case, feed forward neural networks were evolved to learn not only the weights to use, but also what inputs to use.

In (Germa L. Osella Massa, 2006), we see that different neural networks modules, each specializing in a different task, may be combined together to obtain some desired result. They extend on a framework called NEAT (NeuroEvolution of Augmenting Topologies) in which the history and structure of the networks are considered in the crossover operations. The networks in NEAT are speciated and individual networks share their species' fitness. NEAT increments the complexity of its training incrementally, so that the simpler solutions have a higher probability of being searched first.

The extension to NEAT adds in the concept of modular recurrent networks. The modules are first trained individually and then put into a larger decision making network. The output of each module goes to a final recurrent output network. The output network also has access to the input layer, the connections being created and mutated as the networks evolve. To test their extension, the authors trained a network whose task was to avoid obstacles and reach a light source.

As another example of evolutionary algorithms training recurrent neural networks, (Dhahri and Alimi, 2006) used an evolutionary strategy to create radial basis function feed forward neural networks to be used for chaotic time series predictions. (Chen and Miikkulainen,) evolved recurrent neural networks to create melodies in the style of Bartok. Though the resultant melodies were simplistic, the authors were able to evolve networks that generated melodies according to the rules of music theory and specific style of Bartok that sounded good locally (measure to measure).

Evolutionary Neural Networks have also been used in the classification domain. In (Castellani, 2006), a feed forward neural network and its input feature vector is evolved and ran against several classification problems. Its results were comparable to standard back propagation models and generally better than PCA models. (Wallace and Bluff, 2000) trained evolutionary feed forward neural networks to classify irises with a 92% accuracy rate. (Davila, 2003) evolved neural networks according to relative strengths of the layers, not individual nodes, and was able to classify musical keys with

better accuracy than could be done with other recurrent strategies.

1.2.2 General Strategies

In the simplest case, the network is represented as a feed forward neural network and the evolutionary algorithm is used to add and remove nodes from the hidden layer and change the weights of connections between layers. More complex algorithms allow the genotype to define the connection strategy between layers or allow the networks to run a back propagation algorithm between generations to allow for further convergence (in this case, the evolutionary algorithm can be thought of as being used to explore the search space).

At the more complex side of things, a genetic algorithm can be used to find the weights of a fully connected network and its input connections. A single network can be thought of as a specialized module and weights can then be evolved between modules to form a more complex type of network.

Chapter 2

Neural Network - EA System

The following sections describe the structures used in this thesis. The neural network is used to analyze time variant data from the different modalities passed in and reach a conclusion about what it thinks the data represents. It is designed to allow the creation of both feedforward and feedback links between nodes. Crossmodal connections are modeled in the connections created from one modality to another. Feedforward connections are modeled by connections coming from the input nodes and go to the output nodes. Feedback connections are modeled by the connections that go from the output nodes back to the input nodes.

A genetic algorithm is applied to evolve the weights and connections in the neural networks towards the task of identification. The initial population of networks is done at random. Mutation operators include simple modifications of the weights (negation, adding some delta), changing threshold and bias values, adding and removing edges and nodes, and changing the connections between subnetworks. Crossover operators include swapping subnetworks and connection schemes.

This system was used in order to allow the network to be able to handle naturally recurrent data, as in audio and video signals. The structure, to be described in the coming sections, was made to segment out the different modalities and to allow connections between the different modalities to be easily generated. The choice of using a genetic algorithm instead of an algorithmic training scheme to allow the generation of different connections, instead of simply finding weights over a fully connected graph. One reason to use the neural network structure proposed is that, in execution, it has the capacity of saving time in that it does not consider every node in the network.

2.1 System Description

2.1.1 Neural Network Design

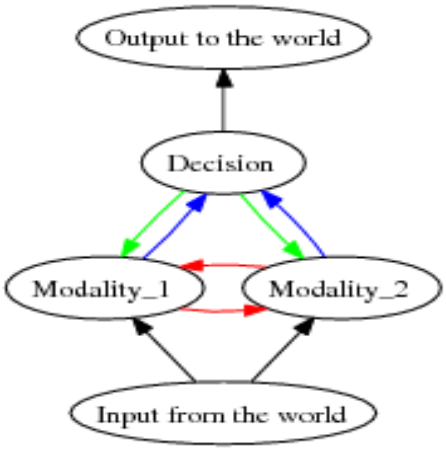


Figure 2.1: An example of a full network with two input modalities and a decision modality. Red arrows represent the crossmodal connections, green arrows represent the feedback connections, and the blue arrows represent the feedforward connections.

Full Network Structure The full network (see figure 2.1) consists of one subnetwork for each modality, an integrator subnetwork, input and output connections to the world, and directed connections between the subnetworks.

Subnetwork Structure A subnetwork is a directed graph with no disjoint nodes. Each edge of the graph has a weight associated with it. There are an arbitrary number of nodes in the graph with edges that satisfy the first constraint.

Node Structure Each node has a current sum value, a sigmoidal squashing function $f(x)$, a threshold, a bias, and a set of afferent and efferent connections.

Connections from Input Modalities Each input modality is assigned a unique subnetwork. For each element in the feature vector for the input, an weighted edge is created from that element to some node in the subnetwork.

Connections between Subnetworks A connection consists of a weighted mapping from nodes in one subnetwork to nodes in another subnetwork.

Output One subnetwork in the Full Network is designated as the output subnetwork. A set of neurons in this subnetwork are randomly chosen as output nodes.

Network Evaluation As the network described isn't the standard feed-foward model, the following method for evaluating the network at each point in time is proposed. This method tries to ensure that all the nodes get evaluated fairly by evaluating all the nodes in random order. At each time step, all the nodes that take input from the world are added to an event queue. Thereafter, half of the nodes are randomly chosen and added to the event queue. Finally, the nodes used as output to the world are added to the event queue. As nodes come off the event queue, we mark that they have been visited. If the node coming off the event queue has already been visited, we simply skip it. If $f(sum + bias) \geq threshold$ for a node, the node goes to all its efferent nodes and add $w * f(sum + bias)$, where w is the weight associated with the efferent connection, to the efferent node. If, when adding to the efferent node's sum the efferent node is be able to fire, that node is moved to the front of the queue.

2.1.2 Genetic Algorithm Design

Genotype The genotype for each subnetwork is the graph for the subnetwork itself. The genotype for a subnetwork to subnetwork connection is a definition as to what nodes are connected from one subnetwork to the other and the weights associated with those connections. The full network genotype consists of a collection of the subnetwork and subnetwork connection genotypes. There are be 3 subnetworks in total, a subnetwork to handle visual data, another to handle audio data, and a final to integrate the two and output the final answer.

Genotype to Phenotype Conversion The genotypes in this case map directly to the phenotype.

Fitness Function The fitness is a measure of how often the system manages to get the output correct over the training set. More specifically, the fitness function is $\frac{numbercorrect}{totalattempts}$.

Parent Selection The parent selection is a random selection weighted towards individuals with a higher fitness.

Survival Selection The best individuals from the combined population of parents and children is kept as the top two thirds of the population. The bottom third is selected via a tournament selection.

Mutation Scheme The weights in the connections (both in the subnetworks and in the subnetwork connections) is randomly negated and transformed from a range of $[-1, 1]$. Connections are randomly added and destroyed. Other mutations include which nodes are designated input and output nodes and changing bias and threshold values for the nodes.

Crossover Scheme Two parents contribute to creating a child. The subnetworks are chosen from the parents in a random fashion. The connection between subnetworks from the same parent is a copy of the same connection from the parent. If the subnetworks are from different parents, a new connection is generated. The world input and output connections for a subnetwork are copied from the parent the subnetwork is taken from.

Chapter 3

Experiments

The classification problem is a good example of a problem that can be helped by multimodal fusion. For this thesis, a system that utilizes both visual and auditory information to identify a person has been implemented. Therefore, there are three modalities in the system; the visual, the auditory, and the decision network. Each modality has been represented as a separate recurrent neural network. The full system has been modeled as the set of modalities and weighted connections between each modality. The first stage of learning, learning how to use the senses, is represented in finding the weights the recurrent neural networks for each modality and for the connections between each modality. The connections between the auditory and visual modalities represent early fusion and cross modal influence. The connections going from the auditory and visual modalities to the decision modalities represent the feedforward connections. Finally, the connections starting from the decision modality to the auditory and the visual modalities represent the feedback connections. A more specific description of the features used and expected results can be found in 3.3. To test the proposed system, two sandbox experiments (see 3.1 and 3.2) have been performed and a preliminary analysis has been made.

3.1 Sandbox Experiment - One Output Node

3.1.1 Experimental setup

In order to test the network structure, a sandbox experiment was conducted in which the networks were trained to differentiate between pairs of sine waves. Each sine wave is defined by a sample increment (s), an amplitude

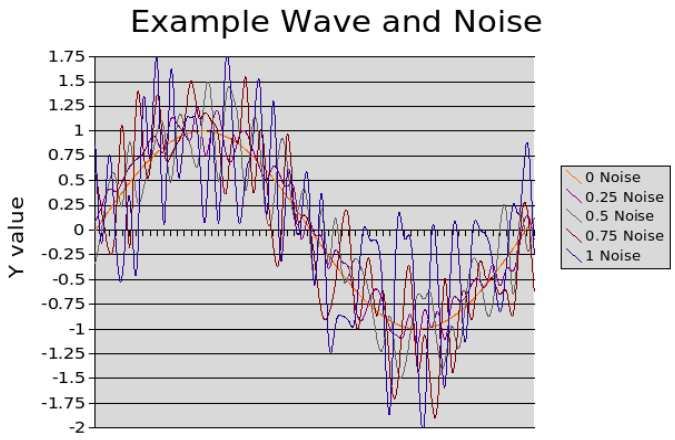


Figure 3.1: Example wave form and noise. The parameters used are: sample rate = .1, amplitude = 1, offset = 0.

(a), and an initial offset (o) such that given the equation:

$$y = a * sin(o + s * t)$$

where y is the vertical position of the sine wave and t is a counter that is initialized at 0 and incremented by one at each time step (see figure 3.1).

50 sets of training data were created per generation, randomly choosing whether the training set would be for the pair A, B or the pair C, D. For each training set, 50 samples of feature vectors were created. A network's decision for each set was based on its output in the last 25 outputs for that set. If an output node was fired more than half the time, then the output node is considered 'on'. Otherwise, that node is considered 'off'.

Given these four sine waves, A,B,C, and D, the networks were trained such that its output node would be on when five time steps of wave A were fed into the first input network and five time steps of wave B were fed into the second input network. When C and D are shown, the output node should be off. Sine waves A and C are members of the first input modality. Sine waves B and D are members of the second input modality.

For the genetic algorithm, 75% of the children were created via the crossover and mutation method and the final 25% were created through mutation. There were 100 individuals in the population for each generation.

3.1.2 Procedure

The parameters for the sine waves A,B,C, and D, were generated randomly such that the sample rate was between .01 and 1, the amplitude was between .5 and 3, and the initial offset was between -2 and 2. Five sets of networks were then trained each with increasing amounts of noise. The noise values used were 0, .25, .5, .75, and 1. Noise is defined as a random amount between the range of 0 and the upper bound passed in (the noise value) added to the value at each time step. The set of networks with a noise level of 0 were used as the control group and the other sets were used to analyse the relationships between the noise level and the usage of connections between the networks. This procedure was repeated for a total of 40 times.

3.1.3 Hypothesis

As the noise in the input signals increase, the strength of the crossmodal connections and the feedback connections should increase as well. If crossmodal connections are indeed strong, we should expect to see that the ratio of crossmodal strength over feedforward strength should be close to or greater than one. Similarly, if the feedback connections are strong, we should expect to see that the feedback over feedforward ratio is close to or greater than one.

3.1.4 Results

As can be seen in figure 3.3, both the crossmodal importance and the feedback importance tend to be bimodal, either around zero importance or equal importance. Figure 3.4 shows that the average importance tends to increase as noise increases. Figure 3.2 gives an overall view as to how the importance values look across all runs and noise. Note that only the top graph for each run was used in analysis. Figure A.1 shows the error rates and the raw feedforward, feedback, and crossmodal influences. Note that the error rates are generally less than 15%.

3.2 Sandbox Experiment - Two Output Nodes

3.2.1 Experimental setup

The setup is the same as the one output node experiment except that the networks now have two outputs. If the input pair was A, B, the first output should be larger than the second output. Otherwise, if the input pair was C, D, the second output should be larger than the first.

	CM/FF	FB/FF
Count (> 0):	34	44
% (>0):	75.56	97.78
AVG (>0):	0.97	1.01
STDDEV(>0):	0.06	0.08

Figure 3.2: Shows the number of importances > 0, the percent of importances > 0, the average of all importances > 0, and the standard deviation of all importances > 0 for experiment one. As the first run was thrown out in this dataset due to a lack of convergence, the total number of importances is 45.

3.2.2 Procedure

The procedure is the same as in the one output node experiment.

3.2.3 Hypothesis

The results should be the same as the first set of experiments. The difference here is that there should be more crossmodal importance as there is a more complex output.

3.2.4 Results

The results are similar to that of experiment one. As can be seen in figure 3.6, both the crossmodal importance and the feedback importance tend to be bimodal, either around zero importance or equal importance. Figure 3.7 shows that the average importance tends to increase as noise increases. The difference between this experiment and experiment one is that there is a larger standard deviation in the average importance per noise level in this experiment. Figure 3.5 gives an overall view as to how the importance values look across all runs and noise. Note that only the top graph for each run was used in analysis. Figure A.1 shows the error rates and the raw feedforward, feedback, and crossmodal influences. Note that the error rates are generally less than 25%.

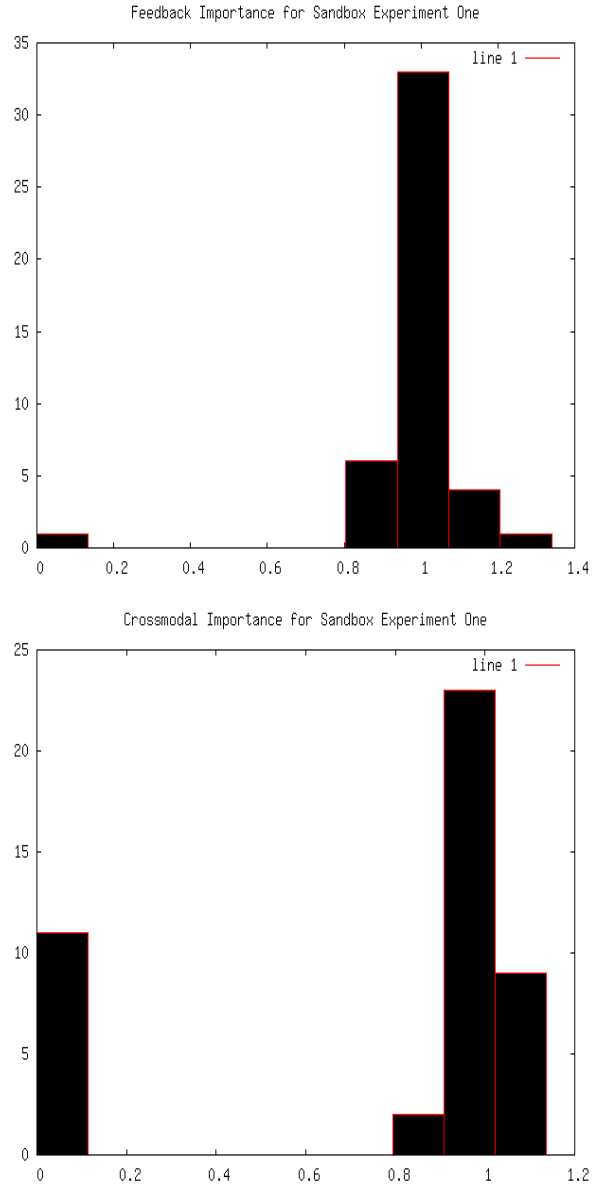


Figure 3.3: Frequency histograms for feedback (FB) and crossmodal (CM) importances in experiment one.

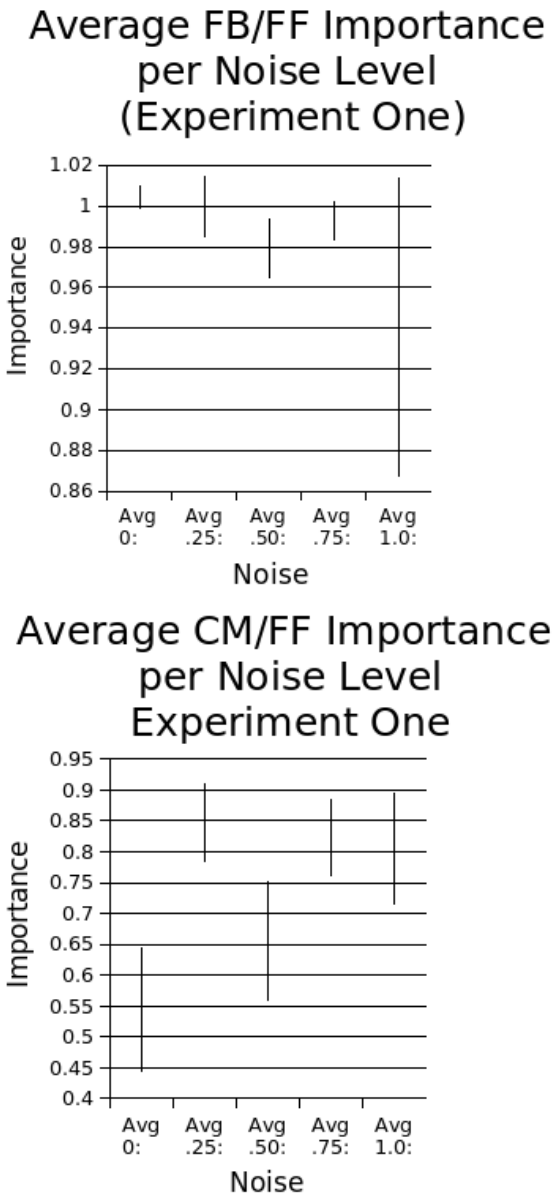


Figure 3.4: Average Importance at each noise level for feedback (FB) and crossmodal (CM) connections for experiment one. The top of the bar represents the maximum of one standard deviation, the bottom the minimum

	CM/FF	FB/FF
Count (> 0):	41	50
% (>0):	82	100
AVG (>0):	1.02	1.01
STDDEV(>0):	0.11	0.08

Figure 3.5: Shows the number of importances > 0, the percent of importances > 0, the average of all importances > 0, and the standard deviation of all importances > 0 for experiment two. There were 10 runs, with 5 noise levels each in this dataset, resulting in 50 crossmodal importance and 50 feedback importance values.

3.3 Person Identification Experiment

3.3.1 Experimental Setup

For the genetic algorithm, 50% of the children were created via the crossover and mutation method, 25% were created through mutation, and the final 25% were created as new individuals. There were 100 individuals in the population for each generation.

At each frame, if the previous 20 frames were from the same speaker, the previous 10 frames were taken into account to decide the output of the network. If the output node fired more than 50% of the time, then the output node was deemed to be on.

For the neural network, there were two outputs. Each output represented one of the individuals speaking. If no one was speaking in the signal given, then both outputs should be off. If the person speaking was the first person, then the first output should be on. If the second person was speaking, the second output should be on. If both outputs are on, then the decision as to who is speaking is decided by whichever output was greater.

3.3.2 Dataset

The dataset used was the freely available CUAVE dataset, provided by Clemson University. It provides a large audio-visual speaker database of digit utterences, including both single-speaker and two-speaker videos.

Michael Richard Siracusa provides a feature-extracted version of the group section of the CUAVE database in which the audio frames are synced with the video frames, 13 MFCCs are generated for each frame, both facial

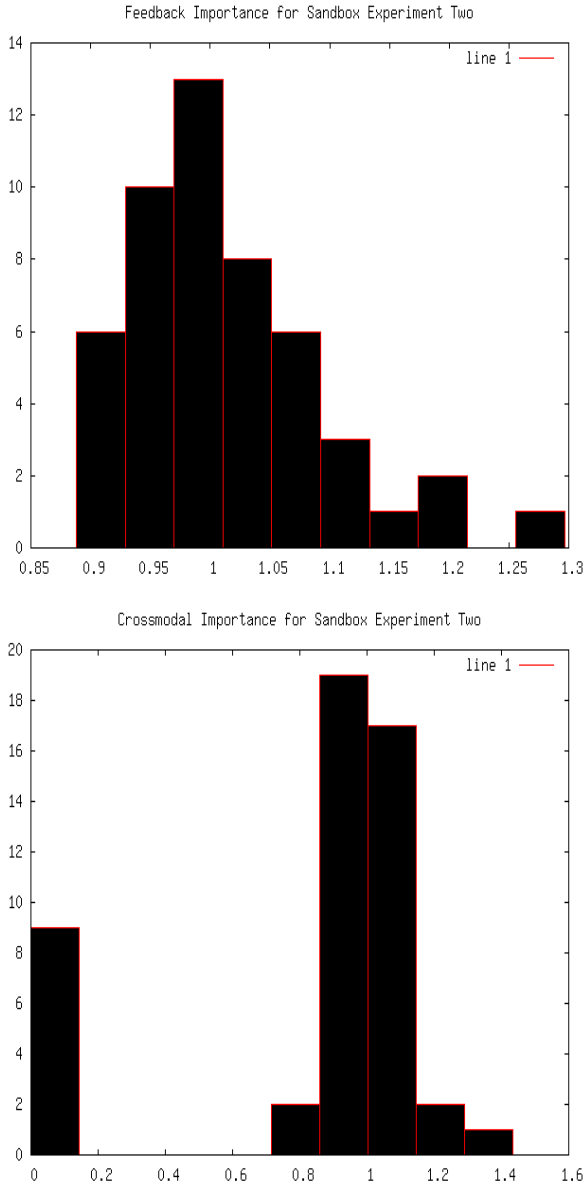


Figure 3.6: Frequency histograms for feedback (FB) and crossmodal (CM) importances in experiment two.

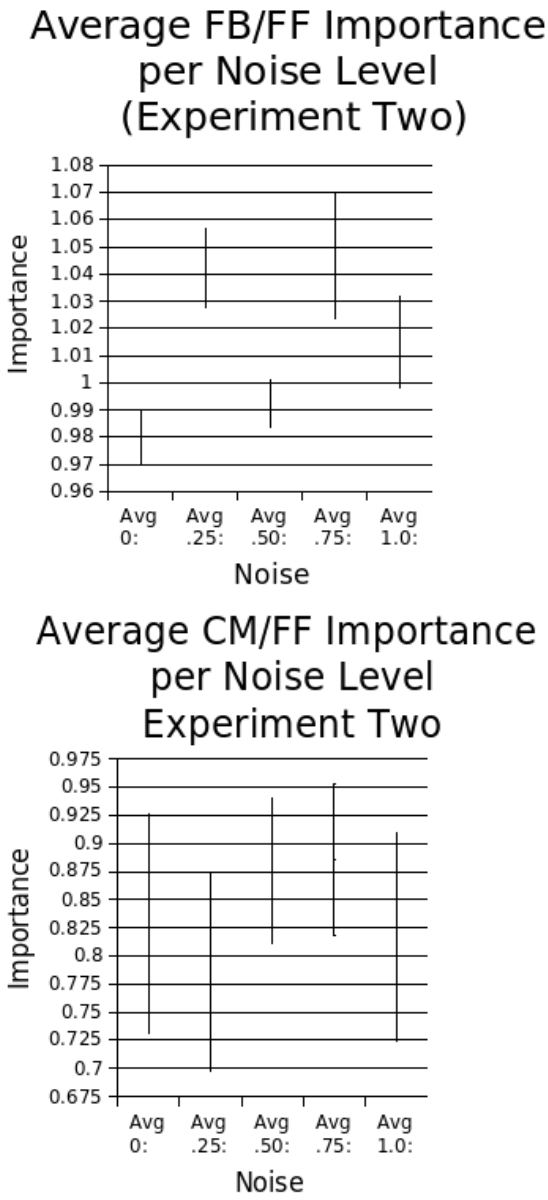


Figure 3.7: Average Importance at each noise level for feedback (FB) and crossmodal (CM) connections for experiment one. The top of the bar represents the maximum of one standard deviation, the bottom the minimum

images are segmented out into seperate grayscale image, and the ground truth (who is speaking at what time) is given for every frame. The ground truth is provided by (Besson et al., 2006).

3.3.3 Features used

Visual Signals

All the extracted grayscale face images were normalized such that the values were all between 0 and 1. Each image was also scaled down into a 10x10 image. The 100 values in this image make up the visual feature vector. Example images are shown in figure 3.8 and figure 3.9

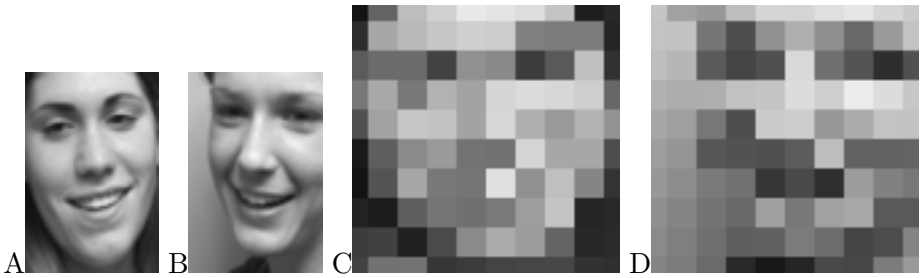


Figure 3.8: A. original image of the first person in group 11. B. 10x10 image fed into the network for the first person in group 11. C. original image of the second person in group 11. D. 10x10 image fed into the network for the second person in group 11. These images were taken from frame 100.

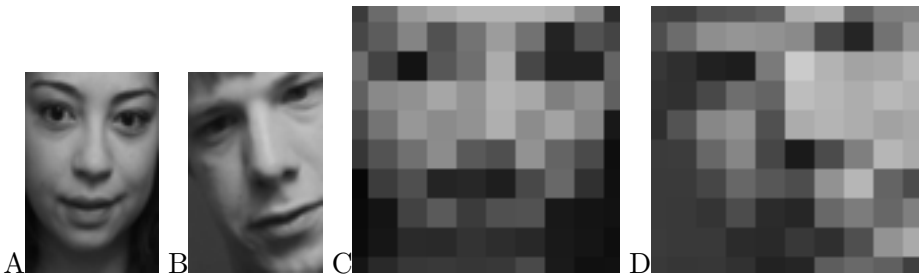


Figure 3.9: A. original image of the first person in group 18. B. 10x10 image fed into the network for the first person in group 18. C. original image of the second person in group 18. D. 10x10 image fed into the network for the second person in group 18. These images were taken from frame 100.

Auditory Signals

Mel Frequency Cepstra Coefficients (MFCCs) were used to model audio data. This is because the Mel scale (a specific type of logarithmic scale) closely models how humans hear different frequencies in sound data. The power spectrum of the audio in the Mel scale is called the Mel Frequency Cepstral. The coeffient vectors calculated from the cepstral has the property that the euclidean distance between two vectors represent a good measure of comparison for the audio data (Falavigna and Brunelli, 1994).

Test Data

Each group video was considered as a seperate two-person identification task. The first half of the video was used for training the networks and the full video was used for validating the results. There were 22 group videos in all. Each group had data for no audio (for which the inputs to the video was zeroes), the first person speaking, and for the second person speaking. For the person that was speaking, the corresponding visual signal was provided to the network to make its decision.

3.3.4 Procedure

The genetic algorithm - neural network method described was used to find a neural network capable of discriminating between the two speakers for each group video.

3.3.5 Hypothesis

It is expected that for every run, the crossmodal importance and the feedback importance should both have a value of approximately 1. This would show that these types of connections are at least as important as feedforward connections in computation, which should follow as a result of the video signal being inherently noisy (due to the downsampling) and of the input signals having a relevant time dimension.

3.3.6 Results

Figure 3.11 shows a histogram of the crossmodal importance values found from this experiment. Figure 3.11 also shows a histogram of the feedback importance values found from this experiment. Note how most of the values for importance are greather than 0.9. Also to note would be that the networks achieved verification error rates of, on average, 31% and a standard

deviation of 9%. Note that only the top graph for each run was used in analysis. Example outputs are shown in figure 3.10. Figure A.1 shows the error rates and the raw feedforward, feedback, and crossmodal influences.

3.4 Analysis Method

3.4.1 Connectivity

Before running the analysis, the network is first pruned such that equivalent connections are merged together (connections using the same source and destination nodes), nodes that can not be reached from the input nodes are removed, and nodes that can not reach the output nodes are removed.

The existence of a path from one node to another implies that the first node influences the second. Furthermore, the existence of a path from one node to a node that another node connects to implies that both nodes interact and that the final result of one node is influenced by the other. In this analysis, we look at the connectivity between certain nodes. The notation used is $C(a,b)_{length,distance}$, where a represents the source node and b represents the destination node. If a path exists, $C(a,b)_{length,distance} = 1$, otherwise 0.

Distance, in this case, refers to the maximum number of steps used to find candidate destination nodes (see figure 3.12). The destination set consists of all nodes that are efferent to the given destination node within the given distance. For example, a distance of zero would only include the destination node. A distance of one would include the destination node and all nodes that are one step away from the destination node. The maximum path length refers to the maximum number of steps allowed from a given source node to a destination node.

The total influence is defined over the lengths in the range of $[0, m]$ and the distances over the range of $[0, n]$. Because the influence degrades as the length and distance increase, we can redefine the nodal influence as a function of the minimum length and distance needed to connect nodes a and b.

$$I(a,b) = 1 - \frac{\sqrt{length_{min}^2 + distance_{min}^2}}{\sqrt{m^2 + n^2}}$$

The influence table for this experiment (maximum length and distance were both set to 10) is displayed in figure 3.13.

The feedforward influence is defined as the connectivity from the input nodes to the output nodes. Using the above example and a set of output

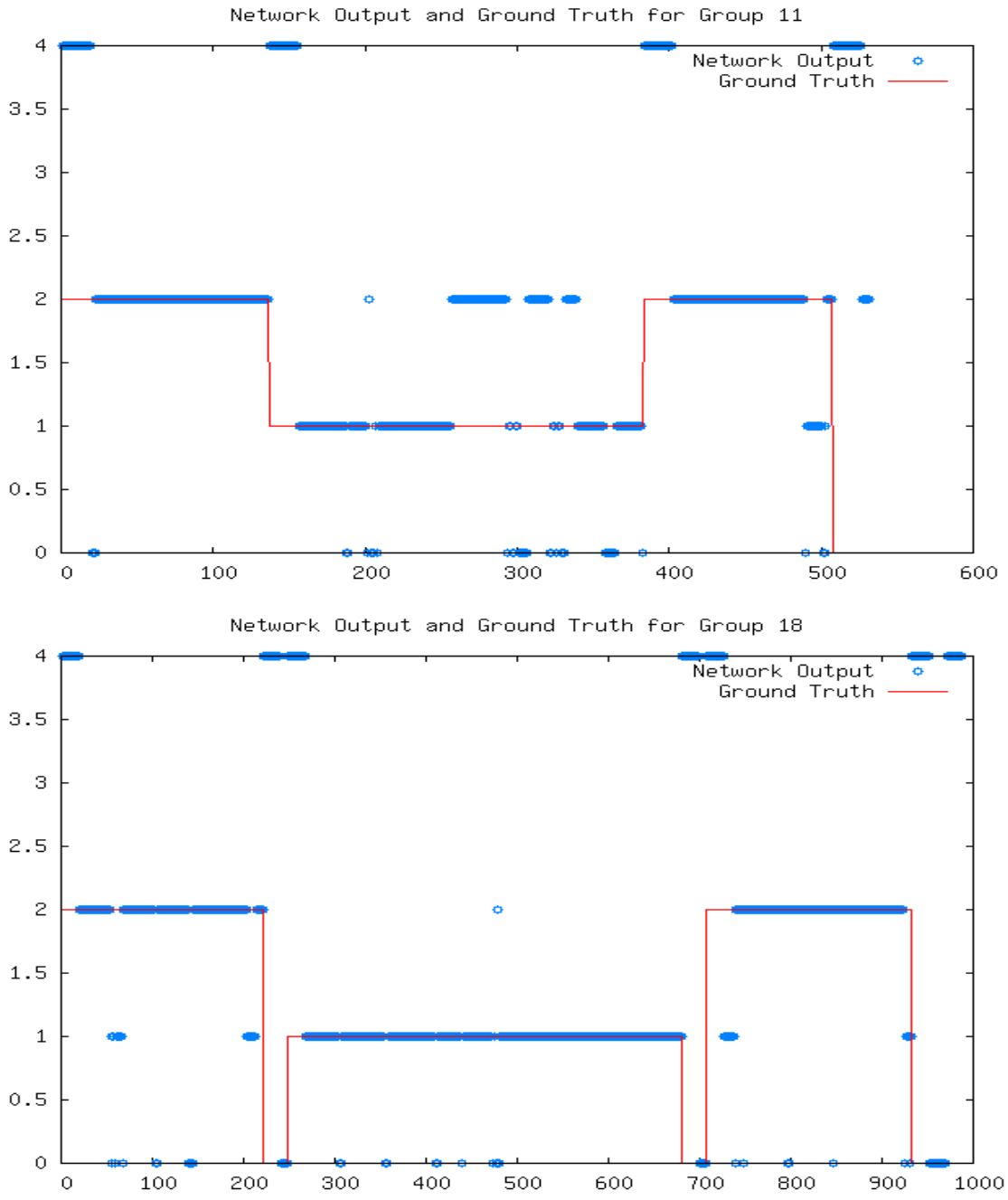


Figure 3.10: Output values for two different groups. An output of 0 signifies no speaker. An output of 1 signifies the first speaker. An output of 2 signifies the second speaker. An output of 4 signifies that the output was not taken into consideration as the speaker had changed at within the last 20 frames. The x-axis represents the frame number. The resultant graphs for groups 11 and 18 are shown in A.2.

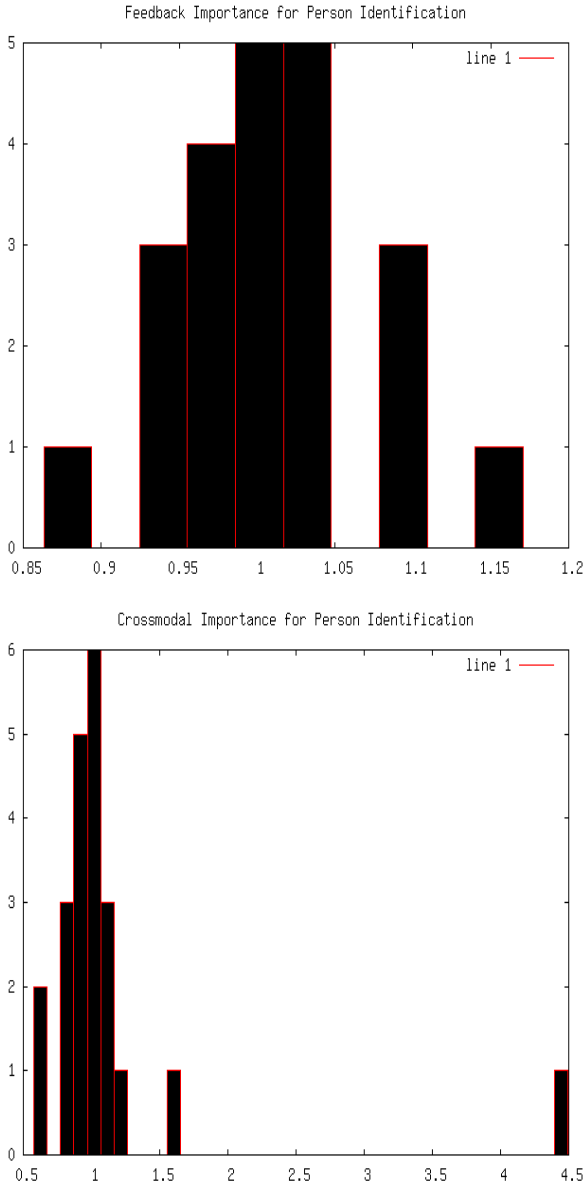


Figure 3.11: Frequency histograms for feedback (FB) and crossmodal (CM) importances in experiment three.

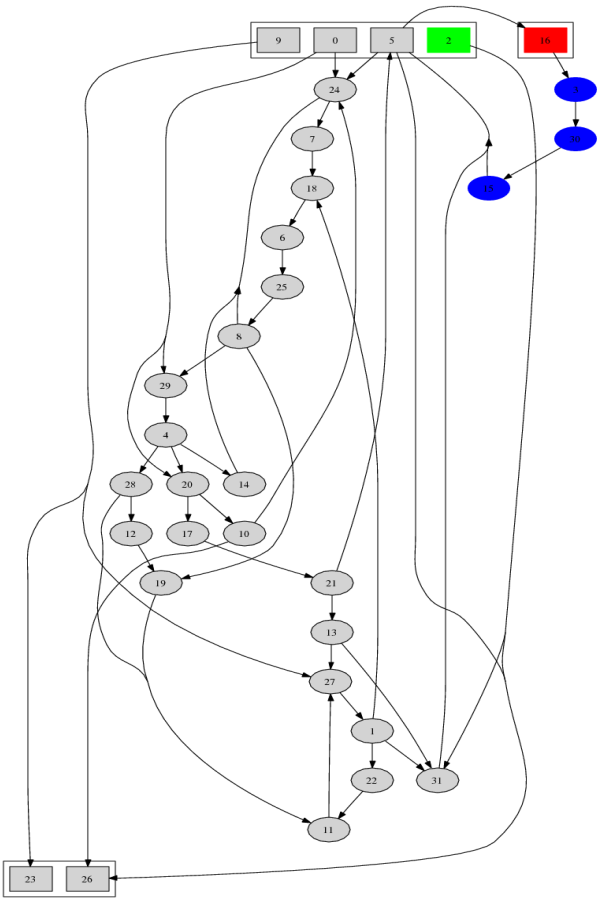


Figure 3.12: One of the solution networks found during experimentation. The green node represents the start node. The red node represents the destination node. The blue nodes represent all the nodes that will be considered when searching for connectivity from the start node to the destination node with a given distance of 3. If the starting node can reach any of the blue nodes or the destination node within the desired maximum length, the connectivity from the start node to the end node is 1, otherwise it is 0.

Influence Table for maximum length = 10, maximum distance = 10

maximum length \ distance	0	1	2	3	4	5	6	7	8	9
0	1	0.93	0.86	0.79	0.72	0.65	0.58	0.51	0.43	0.36
1	0.93	0.9	0.84	0.78	0.71	0.64	0.57	0.5	0.43	0.36
2	0.86	0.84	0.8	0.75	0.68	0.62	0.55	0.49	0.42	0.35
3	0.79	0.78	0.75	0.7	0.65	0.59	0.53	0.46	0.4	0.33
4	0.72	0.71	0.68	0.65	0.6	0.55	0.49	0.43	0.37	0.3
5	0.65	0.64	0.62	0.59	0.55	0.5	0.45	0.39	0.33	0.27
6	0.58	0.57	0.55	0.53	0.49	0.45	0.4	0.35	0.29	0.24
7	0.51	0.5	0.49	0.46	0.43	0.39	0.35	0.3	0.25	0.19
8	0.43	0.43	0.42	0.4	0.37	0.33	0.29	0.25	0.2	0.15

Figure 3.13: The influence table for maximum length = 10 and maximum distance = 10

nodes, $Out = \{o_1, o_2, o_3, ..., o_n\}$, we can define the feedforward influence as

$$I_{feedforward} = \frac{\sum_{i \in A_{inputs} \cup B_{inputs}} \sum_{j \in Out} C(i,j)}{\|A_{inputs}\| * \|B_{inputs}\| * \|Out\|}$$

The crossmodal influence, $I_{crossmodal}$, is defined as the connectivity between input nodes across two modalities. For example, if modality A has inputs $A_{inputs} = \{a_0, a_1, a_2, ..., a_n\}$ and modality B has inputs $B_{inputs} = \{b_0, b_1, b_2, ..., b_n\}$, the crossmodal influence would be defined as

$$I_{crossmodal} = \frac{\sum_{i \in A_{inputs}} \sum_{j \in B_{inputs}} C(i,j) + C(j,i)}{2 * \|A_{inputs}\| * \|B_{inputs}\|}$$

The feedback influence is defined in a similar fashion, from the output nodes back into the input nodes. Feedback, in this case, does not imply a loop from a node back to itself. It is defined as the influence that the decision nodes, a high level signal, have on the input nodes, the low level signals. Using the above examples,

$$I_{feedback} = \frac{\sum_{i \in Out} \sum_{j \in A_{inputs} \cup B_{inputs}} C(i,j)}{\|A_{inputs}\| * \|B_{inputs}\| * \|Out\|}$$

We can now define crossmodal importance and feedback importance through their relations to the feedforward influence.

$$Crossmodal\_Importance = I_{crossmodal} / I_{feedforward}$$

$$Feedback\_Importance = I_{feedback}/I_{feedforward}$$

Chapter 4

Discussion

When analyzing the graphs generated from the experiments, it was found that the resulting networks did not contain modality-specific subgraphs with easily traceable connections between subgraphs as was expected. Instead the genetic algorithm found recurrent solutions, where the path of computation for a modality traversed through all the subgraphs by way of the subgraph-subgraph connections.

As such, instead of analyzing the explicit subgraph connections directly, the method described in section 3.4 was used to evaluate the resulting graphs. Weights are not taken into account in this metric because we are more interested in whether or not one part of the graph influences the result (a binary decision) than in the strength of the influence itself.

From the first sandbox experiment, it can be seen that the feedback influence was, in almost all cases, at least as or more important than the feedforward influence. This was expected because the "modalities" in this case were sine waves, a recurrent signal. The crossmodal importance was more varied, as a significant percentage of the values were zero or close to zero. It is interesting to note, however, that the results were bimodal. Either the crossmodal importance was zero or the crossmodal influence was at least as important as the feedforward influence. This may be because, especially in cases where there is little noise, the decision can be made by looking at one modality. However, when crossmodal influences are needed (possibly because of increased noise), the information for computation comes about evenly from all modalities. The results for the second sandbox experiment were similar. The average crossmodal importance for the second sandbox experiments was higher than those of the first sandbox experiment, though not as high as expected. The standard deviation for the second experiment's

crossmodal importance was larger than that of the first experiment, which can be explained by the fact that the output function was more complex.

The results from the person identification experiment (experiment three) were in line with the sandbox experiments. To note would be that there are no zero-crossmodal or feedback importance values. This may be because the visual and auditory data relate to the same event and therefore more in correspondence with each other than in the previous two experiments. As the two modalities relate to each other more readily, we see an increased tendency towards a higher crossmodal importance value.

The error rate in the person identification experiment was somewhat high, but as this thesis is about the connectivity between modalities as long as the error was less than random the analysis used should hold. In regards to error, it would be worth looking into the results taken in a study of human person recognition done by (Bigun et al., 2001). 4000 humans were asked to classify a given image of a face to 10 possible candidate images. In their study, they found that humans were, on the average, 55% to 75% successful in classifying the image correctly. Keep in mind that this classification task was was in one modality.

In (Mihaila, 2005), there was a 100% recognition reported under no noise conditions using the CUAVE dataset. Under noisy face conditions, they achieved 100% recognition under the best set of fusion weights and a minimum of 79.31%. (Dean et al., 2005) reported a 2% error in identification using the CUAVE database, weighting the visual and audio signals equally.

The results from this thesis suggests that there is a high rate of information sharing between modalities in the best solutions found for problems that take into account multiple sources of corresponding noisy data. In looking at the resultant graphs, it is seen that this information sharing occurs at an early stage in processing data, which is much earlier than in the standard feedforward model used in many applications written today. This idea of early integration is in line with recent findings in biology where information from one modality affects the *sensing* of another one (for example, the McGurk affect) (Meredith, 2002).

Chapter 5

Future Work

This thesis opens up avenues for further research into the usage of crossmodal and feedback information in computational tasks. Various suggestions are made as to possible future work in this chapter.

One idea would be to run the experiments described over a larger set of data. This would allow a more reliable set of statistically significant data from which to draw conclusions. The roadblock to this is the prohibitive computation time from running the NN-GA system to generate the data set. It took three weeks to generate the graphs for all the experiments, even though the computation was spread over 50+ computers.

It would be interesting to see the resultant graphs for a more complex problem. Another interesting experiment to run would be one with more than two input modalities and to analyze the influences of pairs of modalities with respect to other pairs.

Another idea would be a more complex analysis function. The analysis presented does not take into account the weights between nodes and the thresholds and biases associated with each node. One path to pursue would be to run the networks through a large dataset and count how often edges are used.

Appendix A

Appendix

A.1 Result Charts

On the following pages are influence and importance values for all the runs conducted in this thesis.

Experiment One Influence and Importance Values

Run #	Noise	Error	FF	CM	FB	CM/FF	FB/FF
Run_1	0	0.2	0.87	0.82	0.86	0.94	0.99
Run_1	0.25	0.22	0.78	0.8	0.73	1.03	0.94
Run_1	0.5	0.18	0.93	0.93	0.93	1	1
Run_1	0.75	0.42	0.82	0.77	0.79	0.93	0.96
Run_1	1	0.46	0	0	0	#DIV/0!	#DIV/0!
Run_2	0	0	0.91	0.89	0.91	0.98	1
Run_2	0.25	0	0.79	0.76	0.69	0.96	0.88
Run_2	0.5	0	0.72	0.72	0.58	1	0.81
Run_2	0.75	0	0.86	0.83	0.86	0.96	1
Run_2	1	0	0.89	0.83	0.81	0.93	0.91
Run_3	0	0	0.93	0	0.93	0	1
Run_3	0.25	0	0.8	0.79	0.82	0.99	1.03
Run_3	0.5	0	0.83	0.78	0.83	0.94	1
Run_3	0.75	0	0.92	0.84	0.92	0.91	1
Run_3	1	0	0.81	0.85	0.79	1.04	0.98
Run_4	0	0	0.8	0.73	0.81	0.91	1.01
Run_4	0.25	0	0.93	0	0.93	0	1
Run_4	0.5	0	0.93	0	0.93	0	1
Run_4	0.75	0	0.84	0.68	0.81	0.82	0.97
Run_4	1	0	0.69	0	0.93	0	1.34
Run_5	0	0	0.93	0	0.93	0	1
Run_5	0.25	0.02	0.82	0.76	0.82	0.92	1
Run_5	0.5	0	0.81	0.84	0.84	1.04	1.03
Run_5	0.75	0	0.77	0.8	0.8	1.04	1.03
Run_5	1	0	0.8	0.84		1.04	0
Run_6	0	0.02	0.84	0.83	0.85	0.99	1.02
Run_6	0.25	0.1	0.84	0.8	0.77	0.95	0.91
Run_6	0.5	0.04	0.81	0.8	0.86	0.99	1.06
Run_6	0.75	0	0.75	0.7	0.78	0.94	1.04
Run_6	1	0.12	0.81	0.85	0.82	1.06	1.01
Run_7	0	0	0.84	0	0.84	0	1
Run_7	0.25	0	0.85	0.79	0.81	0.93	0.95
Run_7	0.5	0	0.87	0.8	0.8	0.93	0.93
Run_7	0.75	0	0.81	0.74	0.73	0.91	0.9
Run_7	1	0.08	0.65	0.73	0.73	1.14	1.12
Run_8	0	0	0.79	0.83	0.83	1.05	1.06
Run_8	0.25	0	0.79	0.69	0.82	0.88	1.05
Run_8	0.5	0	0.79	0.79	0.77	1	0.98
Run_8	0.75	0.12	0.79	0.72	0.83	0.92	1.06
Run_8	1	0.1	0.82	0	0.9	0	1.1
Run_9	0	0	0.83	0.8	0.79	0.97	0.95
Run_9	0.25	0	0.72	0.75	0.79	1.05	1.1
Run_9	0.5	0	0.93	0	0.93	0	1
Run_9	0.75	0	0.78	0.72	0.73	0.92	0.94
Run_9	1	0	0.78	0.8	0.78	1.03	1
Run_10	0	0	0.93	0	0.93	0	1
Run_10	0.25	0	0.75	0.71	0.81	0.95	1.08
Run_10	0.5	0	0.93	0	0.93	0	1
Run_10	0.75	0	0.93	0	0.93	0	1
Run_10	1	0	0.85	0.85	0.85	1	1

Experiment Two Influence and Importance Values

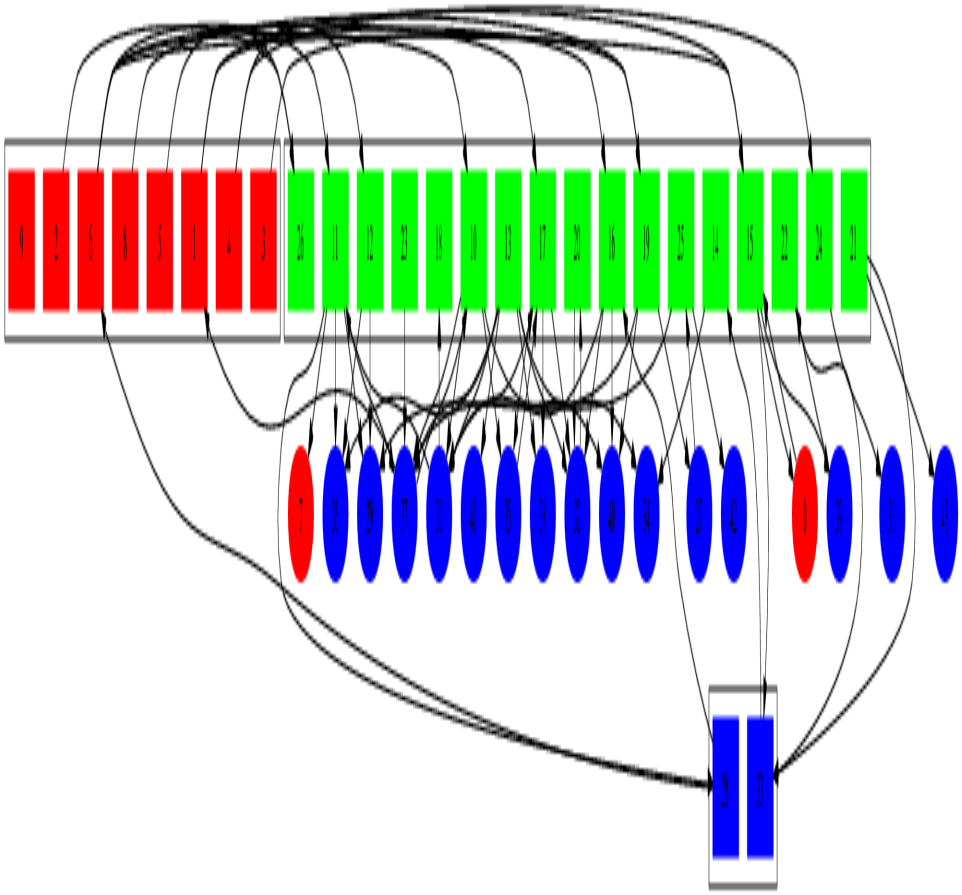
Run #	Noise	Error	FF	CM	FB	CM/FF	FB/FF
Run_1	0	0	0.37	0	0.37	0	1
Run_1	0.25	0	0.74	0.77	0.65	1.05	0.89
Run_1	0.5	0.02	0.71	0.76	0.77	1.07	1.08
Run_1	0.75	0.02	0.82	0.71	0.82	0.87	1
Run_1	1	0.04	0.71	0.68	0.73	0.97	1.04
Run_2	0	0	0.46	0	0.46	0	1
Run_2	0.25	0	0.46	0	0.45	0	0.98
Run_2	0.5	0.08	0.67	0.72	0.67	1.08	1
Run_2	0.75	0	0.68	0.71	0.72	1.04	1.06
Run_2	1	0.02	0.62	0.8	0.73	1.29	1.18
Run_3	0	0	0.47	0.68	0.47	1.43	1
Run_3	0.25	0	0.8	0.77	0.74	0.96	0.92
Run_3	0.5	0.02	0.75	0.68	0.8	0.91	1.07
Run_3	0.75	0	0.68	0.76	0.64	1.12	0.94
Run_3	1	0.02	0.8	0.83	0.83	1.03	1.03
Run_4	0	0	0.62	0.75	0.61	1.21	0.99
Run_4	0.25	0.06	0.46	0	0.46	0	1
Run_4	0.5	0.04	0.77	0.76	0.74	0.98	0.96
Run_4	0.75	0.04	0.72	0.61	0.73	0.85	1.02
Run_4	1	0.14	0.74	0.77	0.7	1.05	0.94
Run_5	0	0.06	0.79	0.78	0.82	1	1.04
Run_5	0.25	0.04	0.69	0.77	0.75	1.12	1.09
Run_5	0.5	0.08	0.83	0.81	0.79	0.98	0.95
Run_5	0.75	0.14	0.74	0.75	0.76	1.02	1.03
Run_5	1	0.2	0.73	0.68	0.68	0.93	0.93
Run_6	0	0	0.46	0	0.46	0	1
Run_6	0.25	0	0.67	0.74	0.75	1.11	1.12
Run_6	0.5	0.02	0.71	0	0.68	0	0.96
Run_6	0.75	0	0.77	0.74	0.72	0.95	0.93
Run_6	1	0	0.74	0	0.74	0	1
Run_7	0	0.1	0.78	0.74	0.72	0.95	0.92
Run_7	0.25	0.12	0.8	0.72	0.79	0.9	0.98
Run_7	0.5	0.02	0.72	0.69	0.68	0.96	0.96
Run_7	0.75	0.04	0.79	0	0.76	0	0.96
Run_7	1	0.06	0.73	0	0.72	0	0.98
Run_8	0	0	0.71	0.71	0.75	0.99	1.05
Run_8	0.25	0.02	0.71	0.73	0.82	1.03	1.15
Run_8	0.5	0	0.72	0.75	0.76	1.03	1.05
Run_8	0.75	0.02	0.72	0.77	0.84	1.07	1.18
Run_8	1	0	0.62	0.68	0.69	1.1	1.12
Run_9	0	0.1	0.77	0.73	0.71	0.94	0.91
Run_9	0.25	0	0.67	0.68	0.73	1.01	1.08
Run_9	0.5	0.06	0.75	0.72	0.72	0.96	0.97
Run_9	0.75	0.08	0.61	0.67	0.79	1.09	1.3
Run_9	1	0.26	0.77	0.79	0.78	1.03	1.02
Run_10	0	0	0.83	0.78	0.76	0.94	0.91
Run_10	0.25	0.12	0.79	0.75	0.84	0.94	1.06
Run_10	0.5	0.28	0.81	0.79	0.83	0.97	1.02
Run_10	0.75	0.22	0.83	0.69	0.83	0.83	1
Run_10	1	0.28	0.84	0.78	0.78	0.92	0.92

Experiment Three Influence and Importance Values

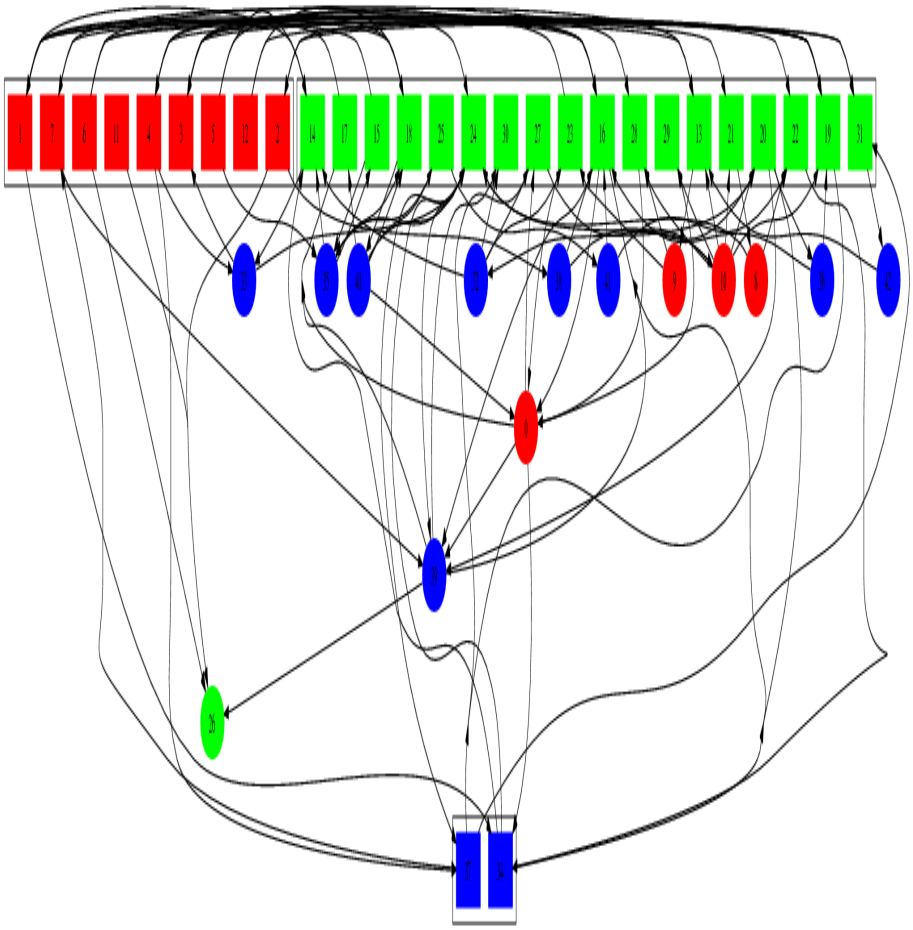
Run#	Error	FF	CM	FB	Points Tested	Points Trained	Total Points	Validation Error	CM/FF	FB/FF
Run_1	0.17	0.73	0.76	0.72	193	627	798	0.31	1.04	0.98
Run_2	0.25	0.44	0.42	0.44	162	654	825	0.25	0.95	0.99
Run_3	0.14	0.7	0.64	0.73	249	844	996	0.3	0.91	1.04
Run_4	0.08	0.68	0.76	0.79	244	922	1093	0.26	1.11	1.17
Run_5	0.28	0.82	0.84	0.83	215	754	944	0.29	1.03	1.02
Run_6	0.09	0.68	0.54	0.69	166	534	687	0.31	0.79	1.02
Run_7	0.11	0.65	0.77	0.7	167	334	410	0.5	1.19	1.08
Run_8	0.15	0.78	0.75	0.73	387	931	1083	0.42	0.96	0.94
Run_9	0.04	0.42	0.48	0.4	101	674	826	0.15	1.15	0.97
Run_10	0.12	0.09	0.41	0.09	112	352	485	0.32	4.49	1
Run_11	0.1	0.66	0.42	0.62	132	455	531	0.29	0.64	0.93
Run_12	0.13	0.74	0.64	0.76	200	613	743	0.33	0.86	1.03
Run_13	0.24	0.75	0.76	0.76	235	530	663	0.44	1.02	1.02
Run_14	0.17	0.14	0.08	0.14	213	622	802	0.34	0.58	1
Run_15	0.29	0.65	0.71	0.71	166	549	682	0.3	1.1	1.11
Run_16	0.23	0.43	0.68	0.37	231	578	745	0.4	1.56	0.86
Run_17	0.15	0.78	0.78	0.77	98	492	587	0.2	1.01	0.99
Run_18	0.17	0.8	0.75	0.75	123	856	988	0.14	0.94	0.94
Run_19	0.18	0.7	0.58	0.69	142	524	686	0.27	0.82	0.98
Run_20	0.19	0.72	0.66	0.79	161	831	1002	0.19	0.92	1.1
Run_21	0.31	0.74	0.74	0.77	232	601	772	0.39	1	1.03
Run_22	0.22	0.65	0.65	0.63	233	560	693	0.42	0.99	0.96

A.2 Graphs

On the following pages are the best graphs from runs 11 and 18 of the person identification experiments. Red is the audio modality. Green is the visual modality. Blue is the decision modality. The red squares are the nodes to which the audio input signals go to. The green squares are the nodes to which the visual input signals go to. The blue squares are the output nodes.



Group 11



Group 18

Bibliography

A. Pnevmatikakis, F. Talantzis, J. S. and Polymenakos, L. (2006a). Robust multimodal audio-visual processing for advanced context awareness in smart spaces. *Artificial Intelligence Applications and Innovations (AIAI06)*, pages 290–301.

A. Pnevmatikakis, F. Talantzis, J. S. and Polymenakos, L. (2006b). Robust multimodal audio-visual processing for advanced context awareness in smart spaces. In *Artificial Intelligence Applications and Innovations*.

Besson, P., Monaci, G., Vandergheynst, P., and Kunt, M. (2006). Experimental evaluation framework for speaker detection on the CUAVE database. Technical report. ITS.

Bigun, J., wai Choy, K., and Olsson, H. (2001). Evidence on skill differences of women and men concerning face recognition. *Lecture Notes in Computer Science*, 2091:44–??

Braun, S. and Gero, J. (2006). A self-training system that learns through experimentation. In *International Design Conference*.

Castellani, M. (2006). Anne - a new algorithm for evolution of artificial neural network classifier systems. In *IEEE Congress on Evolutionary Computation*.

Chellapilla, K. and Fogel, D. (1999). Evolution, neural networks, games, and intelligence. volume 87, pages 1471–1496.

Chen, C.-C. J. and Miikkulainen, R. Creating melodies with evolving recurrent neural networks.

Chong, S. Y., Tan, M. K., and White, J. D. (2005). Observing the evolution of neural networks learning to play the game of othello. *IEEE Trans. Evolutionary Computation*, 9(3):240–251.

Choudhury, T., Clarkson, B., Jebara, T., and Pentland, A. (1998a). Multimodal person recognition using unconstrained audio and video.

Choudhury, T., Clarkson, B., Jebara, T., and Pentland, A. (1998b). Multimodal person recognition using unconstrained audio and video.

Coen, M. H. (2006). *Multimodal dynamics: Self-Supervised Learning in Perceptual and Motor Systems*. PhD thesis, MIT.

Davila, J. J. (2003). Evolution of hierarchical neural networks for time-dependent cognitive processes: Key recognition for musical compositions. In Sarker, R., Reynolds, R., Abbass, H., Tan, K. C., McKay, B., Essam, D., and Gedeon, T., editors, *Proceedings of the 2003 Congress on Evolutionary Computation CEC2003*, pages 716–722, Canberra. IEEE Press.

Dean, D. B., Lucey, P. J., and Sridharan, S. (2005). Audio-visual speaker identification using the cuave database.

Dhahri, H. and Alimi, A. M. (2006). The modified differential evolution and the rbf (mde-rbf) neural network for time series prediction. In *IEEE Joint Conference on Neural Networks*.

E., P., N., K., and van den Herik J. (1998). Enhancing recognition systems through an integrated processing ofvisual and audio information. *Systems, Man, and Cybernetics*, 2:1591 – 1596.

Erdogan, H., Ercil, A., Ekenel, H., Bilgin, S., Eden, I., Kirisci, M., and Abut, H. (2005a). Multi-modal person recognition for vehicular applications. *Lecture Notes In Computer Science 3541*, pages 366–375.

Erdogan, H., Eril, A., Ekenel, H. K., Bilgin, S. Y., Eden, I., Kirisi, M., and Abut, H. (2005b). Multi-modal person recognition for vehicular applications. In *6th International Workshop Multiple Classifier Systems*.

Falavigna, D. and Brunelli, R. (1994). Person recognition using acoustic and visual cues.

Fox, N. A. and Reilly, R. B. (2004). Robust multi-modal person identification with tolerance of facial expression. In *SMC (1)*, pages 580–585.

Gardner, H. E. (1974). *The Quest For Mind*. University of Chicago Press.

Germa L. Osella Massa, Hernan Vinuesa, L. L. (2006). Modular creation of neuronal networks for autonomous robot control. In *IEEE Robotics Symposium*.

Gruber, H. and Vaneche, J. (1977). *The Essential Piaget*. Basic Books.

Gutschoven, B. and Verlinde, P. (2000). Multi-modal identity verification using support vector machines (svm.

Hingston, P. (2007). Evolving players for an ancient game: Hnefatafl. In *IEEE 2007 Symposium on Computational Intelligence and Games (CIG '07)*.

Iwano, K., Hirosem, T., Kamibayashi, E., and Furui, S. Audio-visual person authentication using speech and ear images.

Joassina, F., Mauragea, P., Bruyera, R., Crommelinckb, M., and Campanellaa, S. (2004). When audition alters vision: an event-related potential study of the cross-modal interactions between faces and voices. *Neuroscience Letters*.

Macaluso, E. and Driver, J. (2005). Multisensory spatial interactions: a window onto functional integration in the human brain. *Trends in Neuroscience*.

Meredith, M. A. (2002). On the neuronal basis for multisensory convergence: a brief overview. *Brain research. Cognitive brain research.*, 14(1):31–40.

Meredith, M. A. (2004). Cortico-cortical connectivity and the architecture of crossmodal circuits. *Handbook of Multisensory Processes*.

Michalowski, M. P. and Simmons, R. (2006). Multimodal person tracking and attention classification. In *HRI '06: Proceeding of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*, pages 349–350, New York, NY, USA. ACM Press.

Mihaila, A. (2005). *Person Identification from Video using Facial and Speaker Features*. PhD thesis, University of Joensuu.

Parker, M. and Parker, G. (2007). The evolution of multi-layer neural networks for the control of xpilot agents. In *IEEE 2007 Symposium on Computational Intelligence and Games (CIG '07)*.

Piaget, J. (1952). *Origins of intelligence in children*. International University Press.

Schroeder, C. E. and Foxe, J. (2005). Multimodal contributions to low-level unisensory processing. *Current Opinion in Neurobiology*.

Thompson, T., Levine, J., and Hayes, G. (2007). Evotanks: Co-evolutionary development of game-playing agents. In *IEEE 2007 Symposium on Computational Intelligence and Games (CIG '07)*.

Veeramachaneni, K., Osadciw, L., and Varshney, P. (2003). Adaptive multimodal biometric fusion algorithm using particle swarm.

Vogt, P. (1998). Perceptual grounding in robots. In *EWLR-6: Proceedings of the 6th European Workshop on Learning Robots*, pages 126–141, London, UK. Springer-Verlag.

Wallace, J. G. and Bluff, K. (2000). Neuro-architecture-motivated anns and cortical parcellation. In *IJCNN '00: Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks (IJCNN'00)-Volume 5*, page 5647, Washington, DC, USA. IEEE Computer Society.

Wu, L., Oviatt, S. L., and Cohen, P. R. (1999). Multimodal integration - a statistical view. *IEEE Transactions on Multimedia*, 1(4):334–341.

Yang, J. and Hauptmann, A. G. (2004). Multi-modal analysis for person type classification in news video. In Lienhart, R. W., Babaguchi, N., and Chang, E. Y., editors, *Storage and Retrieval Methods and Applications for Multimedia 2005. Edited by Lienhart, Rainer W.; Babaguchi, Noboru; Chang, Edward Y. Proceedings of the SPIE, Volume 5682, pp. 165-172 (2004).*, pages 165–172.

Yau, Y. J., Teo, J., and Anthony, P. (2007). Pareto evolution and co-evolution in cognitive neural agents synthesis for tic-tac-toe. In *IEEE 2007 Symposium on Computational Intelligence and Games (CIG '07)*.