Modelling Human Perception and Attention Using Recurrent Neural Networks

Thesis

Submitted in partial fulfillment of the requirements of
BITS F421T Thesis

By

Tanay Agrawal
ID No. 2015B4A80567P

Under the supervision of:

Dr. Sridharan Devarajan
&
Dr. B.K. Sharma

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE PILANI, PILANI CAMPUS
November 2019

Declaration of Authorship

I, Tanay Agrawal, declare that this Thesis titled, 'Modelling Human Perception and Attention Using Recurrent Neural Networks' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: ______________________________

Date: ______________________________

Certificate

This is to certify that the thesis entitled, "*Modelling Human Perception and Attention Using Recurrent Neural Networks*" and submitted by Tanay Agrawal ID No. 2015B4A80567P in partial fulfillment of the requirements of BITS F421T Thesis embodies the work done by him under my supervision.

Supervisor	*Co-Supervisor*
Dr. Sridharan Devarajan	Dr. B.K. Sharma
Assistant Professor,	Associate Professor,
IISC Bangalore	BITS Pilani, Pilani Campus
Date:	Date:

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE PILANI, PILANI CAMPUS

Abstract

Master Of Science (Hons.)

Modelling Human Perception and Attention Using Recurrent Neural Networks

by Tanay AGRAWAL

Neuroscience and Artificial Intelligence are two fields that drive each other forward. The purpose of this project is to train an Artificial Neural network to do a task that is common in Neuroscience studies for studying attention in the brain. The ANN is treated as a simulation of the real brain which helps in gaining insight of its working. Most of the similar work involves architectures that are difficult to analyse because of their complexity. So, this project aims to use a simple architecture that correctly represents the brain as well is easy to analyse.

Acknowledgements

I thank Dr. Sridharan Devarajan and Dr. B.K. Sharma for their guidance, and also all the members of Cognition Lab, Centre for Neuroscience, IISc Bangalore for their support.

Contents

Chapter 1

Introduction

1.1 General Statement of the Problem

It is known that we pay visual attention to different points in space and to different features. But, the mechanism of this attention in the brain is still a mystery.

Artificial neural networks as the name suggests, have been designed to mimic the functioning of the brain. Fundamentally, they are function approximators, and after learning the relation between the given input and output, they serve as a black box - the learnt function is unknown to us. Studying the working of this black box will allow us to study the relationship between our variables.

There are multiple psycho-physics tasks designed to study visual attention. training a neural network to perform this task will give us the mathematically optimal approach to that task. This project uses architectures similar to the human brain to study such tasks and includes analysis of these networks to gain insight into their working.

1.2 Significance of the Thesis

The findings of this project will redound to the benefit of society considering the benefits of learning about the functioning of the brain. Directed attention can help reduce the resource requirements of processes greatly - studying visual attention will allow us to minimise requirements in computer vision problems. Also, gaining insight into the working of the human brain will contribute towards curing diseases of the brain.

1.3 Research Question

This project aims to study visual attention towards particular features in an input video stream - contrast, arrangement of the pixels spatially. For this, a neural network is trained to learn to pay attention to one of two tasks in a dual psycho-physics task, given an input cue directing it to one of the two tasks. Then the working of the network is analysed. The aim is to simulate working of part of the brain during the chosen task.

1.4 Limitations and Delimitations

Analysing the working of neural networks is hard as the number of parameters is too high. One can approximate its working, but not fully understand the working. To minimise the abstraction of its working, simplistic architectures are used.

The neurons of the brain are different from those of neural networks. To make the study plausible, the architecture designed is similar to that of the brain.

Due to scarcity of human data required for the training of the network, synthetic data with 100% accuracy is used for training and testing. But, since the task at hand is to study attention only and not other behavioural aspects, this should not affect the outcome.

Chapter 2

Design of Proposed Model and Task

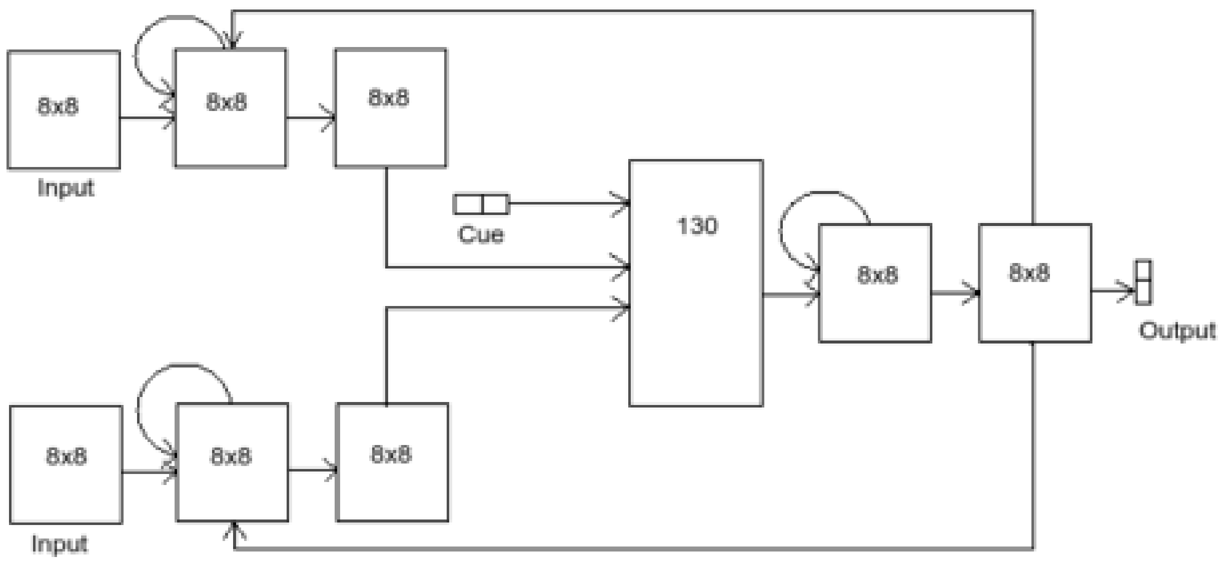


Figure 1. Architecture of the model

Figure 1 shows the architecture of the model. The network is divided into three parts, the first two are for change detection - contrast change detection and orientation change detection (explained later). Every part has three layers, the input layer, a recurrent hidden layer and a fully connected (FC) hidden layer. The third network has an output layer of size 2, which gives ”1” for change and ”0” for no change. The third network’s input is the concatenated vector of the 2 FC hidden layers from the previous networks and the cue.

The input video has grayscale frames contains gratings of four possible orientations - horizontal, vertical and two diagonal directions - and four possible contrast states. At any frame, the contrast or the orientation can change to any of the four possible states. The cue, which is a vector - [10, 0] or [0. 10] - corresponding to orientation and contrast respectively, tells the third network which change to detect.

There is a feedback from the third network to the first two for changing the sensory encoding, which is what the activations of the first two networks symbolise. Readout change can be incorporated by adding a linear layer to the input to the third network, but the weights of this layer in the trained model did not deviate much from 1, to which they were initialised. Since, the result was not significant, to reduce the complexity of the model that layer was removed.

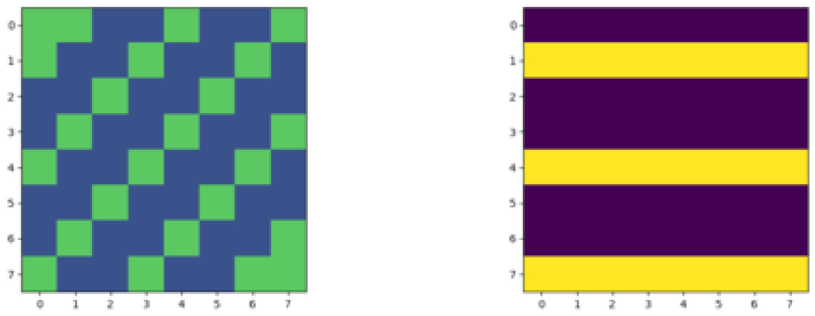


Figure 2. Sample inputs

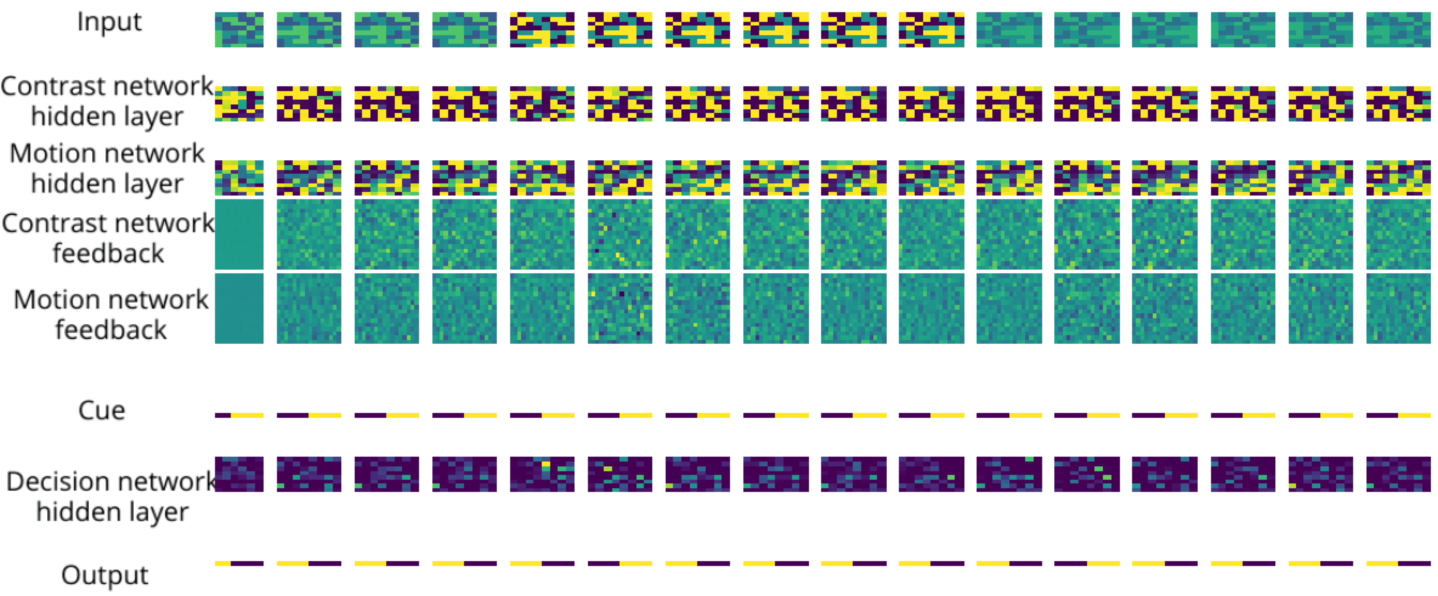


Figure 3. Activation of various layers

Mathematical model:

Contrast and orientation change detection networks (these two networks are identical):

Weight matrices of size $\mathrm{W}_1 : [64, 64], W_2 : [64, 64], W_3 : [64, 64], W_4 : [64, 64]$
$Hidden\ layers\ of\ size H_1 : [16, 64], H_2 : [16, 64], H_3 : [16, 64]$

I = (I - mean(I)) / range(I)

For each row of the input (I),
$\mathrm{H}_{1n} = tanh(I_n * W_1 + H_{1n-1}$ * $\mathrm{W}_2 + H_{3n-1}$ * $\mathrm{W}_3); n > 1$
$H_{11} = tanh(I_1 * W_1)$

$\mathrm{H}_{2n} = tanh(H_{1n} * W_4)$

Third network:

Weight matrices of size $\mathrm{W}_5 : [130, 64], W_6 : [64, 64], W_7 : [64, 64], W_8 : [64, 2]$
$Hidden\ layers\ of\ size\ H_4 : [16, 64], H_5 : [16, 64]$
$Output\ layer\ of\ size\ O[16, 2]$

For each row of the input (I'),
$\mathrm{H}_{4n} = tanh(I'_n * W_5 + H_{4n-1}$ * $\mathrm{W}_6); n > 1$
$H_{41} = tanh(I'_1 * W_5)$

$\mathrm{H}_{3n} = H_{4n} * W_7$
$H_{5n} = tanh(H_{3n})$

$\mathrm{O}_n = H_{5n} * W_8 For\ each\ element\ in\ O_n$,
$O_{ni} = exp(O_{ni}) / \sum_{i=1}^{2} exp(O_{ni})$

Chapter 3

Methodology

3.1 Input Generation and Network Training

The input is generated on the fly during training and testing, randomising the input pattern or orientation of the grating and contrast (depending on the task). It is stored in numpy arrays. The tasks considered in chronological order are as follows:

1) An input video of resolution 360x360, with the first frame as a random pattern. Consecutive frames have the same pattern. At some random frame, part of the pattern will change and this change will be seen in all the following frames. The network has to find the frame when this change occurs first.

Network design:
The hidden layer has 10 extracted features from the input using gaussian filters. The gaussian filters are of size 2x2 to 29x29 with an increment of 3 in width and height. These are connected to a PFC Layer with random and sparse connections. This layer has recurrent connections too. The PFC layer is connected back to the hidden layer, again with sparse and random connections. The hidden layer outputs of the activations of these 10 layers. The output denotes the change location, if it is present. Rectified Linear Unit (ReLU) was used as the activation function for all connections.

Accuracy:
When there is one change in the video and test set is of size 5,000: 100%
When there are 5,000 changes in a video with 80,000 frames: 98.8%

2) An input video with resolution 8x8, with the first frame as a random pattern. Consecutive frames have the the pattern moving in one of eight directions - four cardinal and four diagonal. At any random frame, the pattern can change the direction of motion. The contrast can also

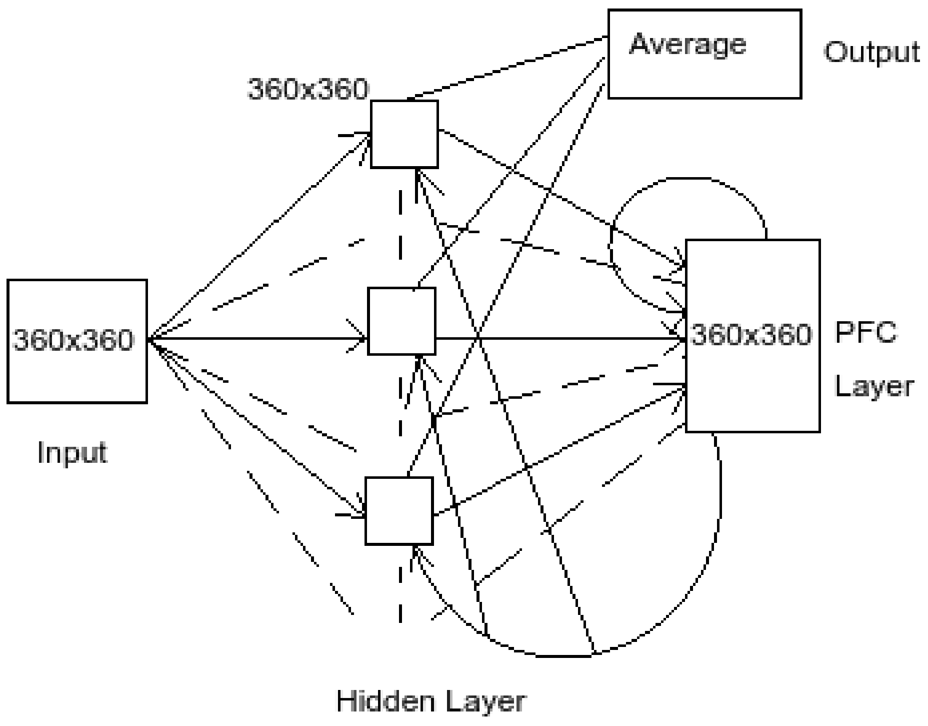


Figure 4. Architecture of model 1

change at any of the frames. The network is given a cue telling it to find the frame with either motion direction change or contrast change. The task was modified later which is discussed in the challenges subsection.

Network design:
The network design is similar to Figure 1. It did not have the feedback connections from the third network to the first two. Rest everything was the same. The network design was modified later which is discussed in the subsequent challenges section.

Challenges:
The task and the network evolved underwent multiple changes. These are discussed in this section.

The contrast change detection network was trained first with ReLU non-linearity but it was giving chance accuracy. Then, input was scaled to the range (-1, 1) and the non-linearity changed to tanh which improved the training plot shape slightly but there was not much change in accuracy. Then, I started training the motion direction change detection network. This also gave chance accuracy initially.

The network was slightly modified after this. The final linear layer output length was changed to 8 outputs and the tasks changed to discrimination tasks. Now the input frames could be moving in any of the 8 directions and have one of the 8 levels of contrast. The motion direction discrimination network wasn't learning the task still and various strategies were tried to train it. Initially, random initialisation of weights and Adam optimiser was used with learning rate (lr) = .001. This failed. Then the learning rates were changed to .0005 till 80 epochs, .0003 till 160 and .0001 after that till 1000 epochs. This also did not change much. The problem for optimisation was mainly that there were too many local minima and the network was getting stuck in 1 every time. 2 possible solutions were tried:
1) Changing the initialisations
2) Transfer learning from simpler tasks
The former did not work but the latter eventually gave 95.2% accuracy for the task and is detailed ahead.

First the input had frames moving in one constant direction with constant contrast. Then, one change frame was introduced for motion direction. Then, one change frame was introduced for contrast change. Then 2 change frames for both the tasks. Finally, random changes were allowed for both.

After training the model for motion direction change detection, contrast network was trained. For this, Adam optimiser did not work well and Adagrad was used instead. A similar transfer learning approach was applied but the accuracy achieved was chance. So, the task was slightly modified where the network had to learn a step of motion in a particular direction in a set of otherwise static frames at any random frame. The motion direction discrimination network was fine trained for this task.

The contrast discrimination model was still not learning owing to the fact that it doesn't connections between different frames' hidden layer. So, the initialisation of the hidden to hidden layer weights of the rnn was changed to 0. For the remaining weights, the input to hidden weights were initialised to 1. Xavier initialisation with normal distribution was used for the linear layers. Finally, 96% accuracy was achieved for the model.

The motion change discrimination network learnt to disregard the contrast change by looking at the constant value in the image (127). Its contingency table was as expected. The diagonal values were maximum and the network didn't confuse between directions of motion rather either gave the correct direction or missed the change. The contrast network's contingency table was maximum along the diagonals and decreased on moving away from the diagonals row-wise(in the direction of actual output). This is expected as the network is learning to detect the difference between values and may get confused if the difference values are closer.

After this, the third network was trained. The architecture was modified - as the third network did not require a RNN, linear layers were used: 6 layers, 1 for input, 1 for output and 4 hidden layers. ReLU non-linearity was used after all hidden layers. The network was provided the correct output (contrast/motion state) with the relevant cue for 5 consecutive states alternatively. The graph for training loss with error looked like a step function, oscillating between the two different values for each task. It required a very low learning rate for training, and thus a lot of epochs. This was remedied by using a high learning rate initially and when the span of oscillations shifted to a lower set of of values and became constant, the learning rate was divided by 100. This gave an accuracy of 86% for both the tasks.

3) The third task is what led to the final model discussed in chapter 2. Like the second task, it went through multiple changes which are discussed below. The initial task for the first two networks was state discrimination and for the third network, it was change detection. Training details are as follows:
The input can change at any frame, unconstrained. Cue was changed from [0,1]/[1,0] to [0,10]/[10,0] as when given as input to the third network, the cue information was lost in the recurrent layer. Training set size was 100,000, validation set size was 25,000 and test set size was 5000. Batch size for training and validation was 200. Thus, there were 500 iterations in each epoch. The training was done for 1000 epochs. The learning rate (lr) was varied as follows:
Epoch 1-70: lr=0.0005
Epoch 71-140: lr=0.0003
Epoch 141-210: lr=0.0001
Epoch 211-1000: lr = 0.00006
The motion and contrast cued inputs were alternated after every six epochs during training. L2 regularisation was used with lambda = 0.008 for motion cued and lambda = 0.006 for contrast cued input. Cross entropy loss function was used. The contrast and motion discrimination networks were trained first and fixed having accuracy of 97.8% and 98.6% respectively. When feedback is zero, the network accuracy is 95.3% for contrast and 96.1% for motion. The accuracy goes to 100% when feedback is allowed. Steps for implementing:
Retraining the contrast and motion discrimination networks with ReLU non-linearity to avoid vanishing gradient problem. Since the network wasn't learning with random initialisation, an initialisation which is the average of the the weights of the third network trained only for motion and then only for contrast was taken. What this aimed to achieve was to reduce the number of local minima by reducing random firing (as the neurons which need to fire less for both motion and contrast are initialised to low values). The feedback weights were kept same for both the networks. Since the network was still not learning, different weights for both feedbacks were preferrede over having same weights but with additional layers in the feedback. This worked and the accuracies are reported as above.

After training this network, we realised that the feedback's work is to improve the accuracy of the previous networks, but since the accuracy of the previous network is already so high, this was not possible. So, noise was added to the input. Multiple types of noises were tried. The aim was to reduce the accuracy of both the networks, while not providing either network any information to assist them in the tasks. The motion task was changed to orientation discrimination (giving a grating as input like the first task), as given the same architecture, the difficulty for the network to perform the tasks was disparate. This network was trained with 82% and 89% accuracies for contrast and motion networks, and 100% for the complete network. The noises tried were:
1) Every pixel will switch its state with another pixel with a chosen probability. (It was set at 45% after multiple values were tried.) 2) When the pixel changes state, a uniform or gaussian noise noise is added to it. 3) To every frame, all possible orientations with randomised contrast state would be added after multiplying with a randomised constant in the range [0.01, 0.1]

After this, we realised the output should be checked at a frame after the change in the input as the feedback is delayed by a time step as shown in the equations in chapter 2 (H3 is the feedback layer). The third network and feedback were retrained with this change and we achieved the same accuracy, 100%.

3.2 Analysis of the networks

Note: The results are compiled in the next chapter. This section contains the procedure of the analysis.

3.2.1 For the second task and model discussed above (after the final changes)

Analysis I: Firing analysis

The activations of various layers were visualised. The following variations were plotted:
1) Temporal average of the activations
2) Difference in firing of the averages given different cue conditions
3) Difference in of the averages given different cue conditions with every element divided by the sum of the two terms which are subtracted

Analysis II

Aim:
1) To see how the third network is performing the function AB + CD (A is contrast cue, C is motion cue, B is contrast data, D is motion data).
2) To make the activation matrices sparse to make them more coherent.

3) Use the weight matrices for analysis as analysing activations for random inputs skews the results due to the non-homogeneity of the inputs.

Procedure:
1) The weight matrix is arranged such that rows correspond to output neurons and columns correspond to input neurons. For all columns, for positive weights, values lying in (Maximum - 1 standard deviation) were kept and for negative values, (Minimum + 1 standard deviation) were kept. This was done for all weight matrices. This was done for making the matrices sparse.
2) Then the non-zero values were set to one, signifying that these connections allow the passage of more information compared to others.
3) After this, starting from the last weight matrix with the output as the final eight neurons the sum of the matrix was first summed column wise resulting in a 64 vector. Now if the nth value in this sum is zero, then the nth row of the previous weight matrix was set to zero. This was done for all adjacent pairs of weight matrices. The significance of this step is that the activations of the neurons which have a lot of information coming in, but are not passing that information have not been considered.
4) Now, different combinations of input (the 130 sized layer in the architecture) were tried: Setting A, B, C, D (described in the aim) to 1 in different permutations
5) A new activation function was used in place of ReLU for all layers, the lowest .8/.6 standard deviation values were set to zero and the remaining set to one. This was done to keep the outputs homogenous and increase understandability.
Note: During this, relu didn't have a role to play as all weights are either zero or one and inputs are one. The activations value simply indicate the amount of information received from the previous layer.

Mathematical representation of the procedure:
1) In W5, W6, W7, W8, for each row this operation was applied elementwise:
X = 0, if X > 0 and X does not belong to belongs to [Maximum(row) - standard deviation(row), Maximum]
= 0, if X < 0 and X does not belong to belongs to [Minimum, Minimum(row) + standard deviation(row)]
= 1, otherwise
2) Now, for all weight matrices this operation was applied element wise:
X = 1, if X is not 0
= 0, otherwise
3) Now, W8 is summed row wise resulting in a vector of length 64. If Xi is zero, where i denotes the ith element in the vector, then in W7's ith column is set to zero.
This step is done for W7 and W8 and so on...
4) -Same as above-

5)The activation function was changed from ReLU to:
X = 0, x belong to [0, (.6 or .8)*standard deviation(row)]
= 1, otherwise

Analysis III

A particular input was taken and 10,000 noise samples having a gaussian distribution (mean = 0, std dev = .04/0.05) were applied to it. Principle component analysis (PCA) was applied to the activations of the hidden layers to reduce the dimensions from 64 to 2. These first 2 components were plotted. This was done with different cue conditions.

The above was repeated with PCA applied only to activations without noise, separately for each plot.

3.2.2 For the third task and network discussed in the previous section

Decoding Analysis

Taking the contrast and motion networks, state and change was decoded from the hidden state that go as input into the third network. This was done for multiple scenarios, like feedback present or absent, value of the cue, etc. The accuracies are shown in a table in the next chapter.

Chapter 4

Findings

4.1 Analysis Results

The results of all the analyses done are compiled in this section.

4.1.1 Analysis I

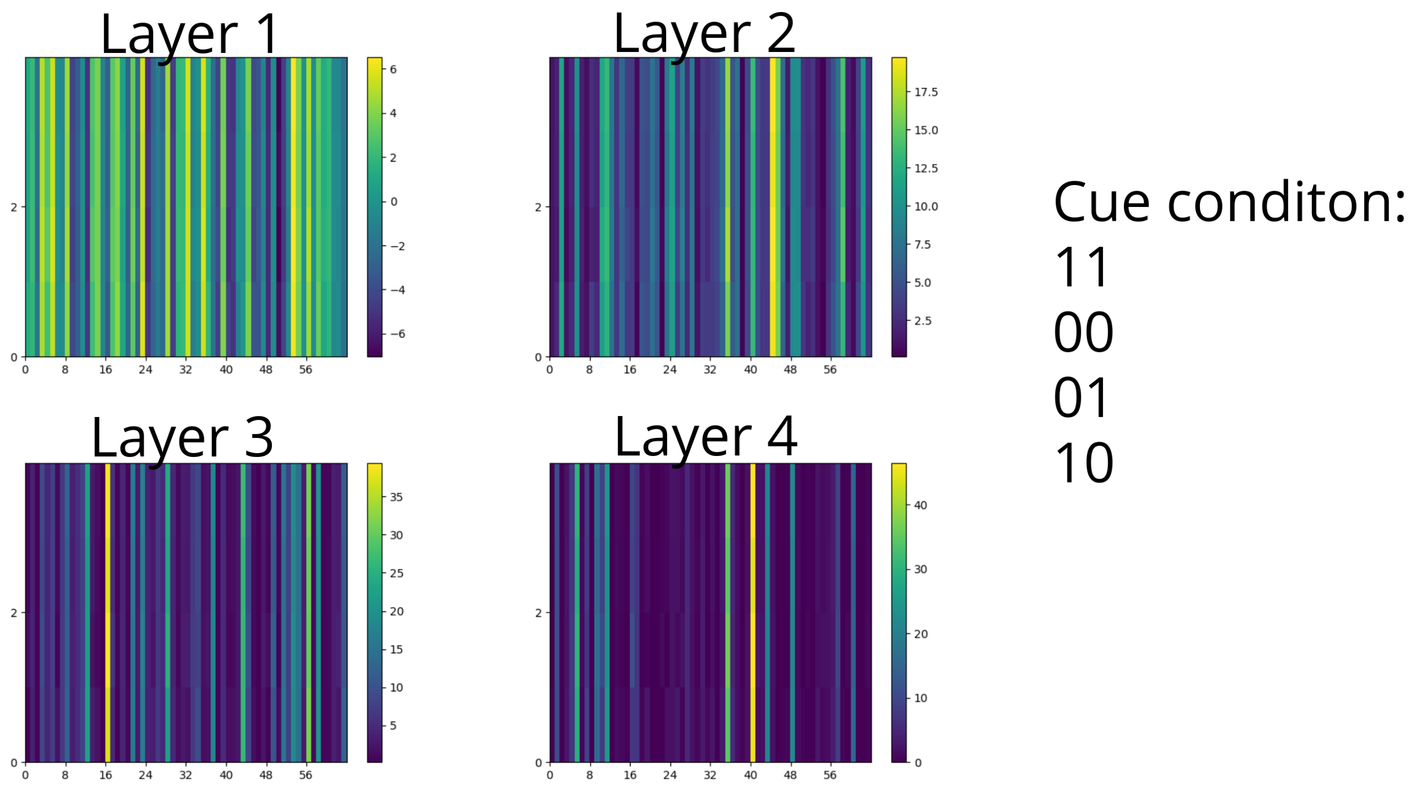


Figure 5. Average firing of individual layers

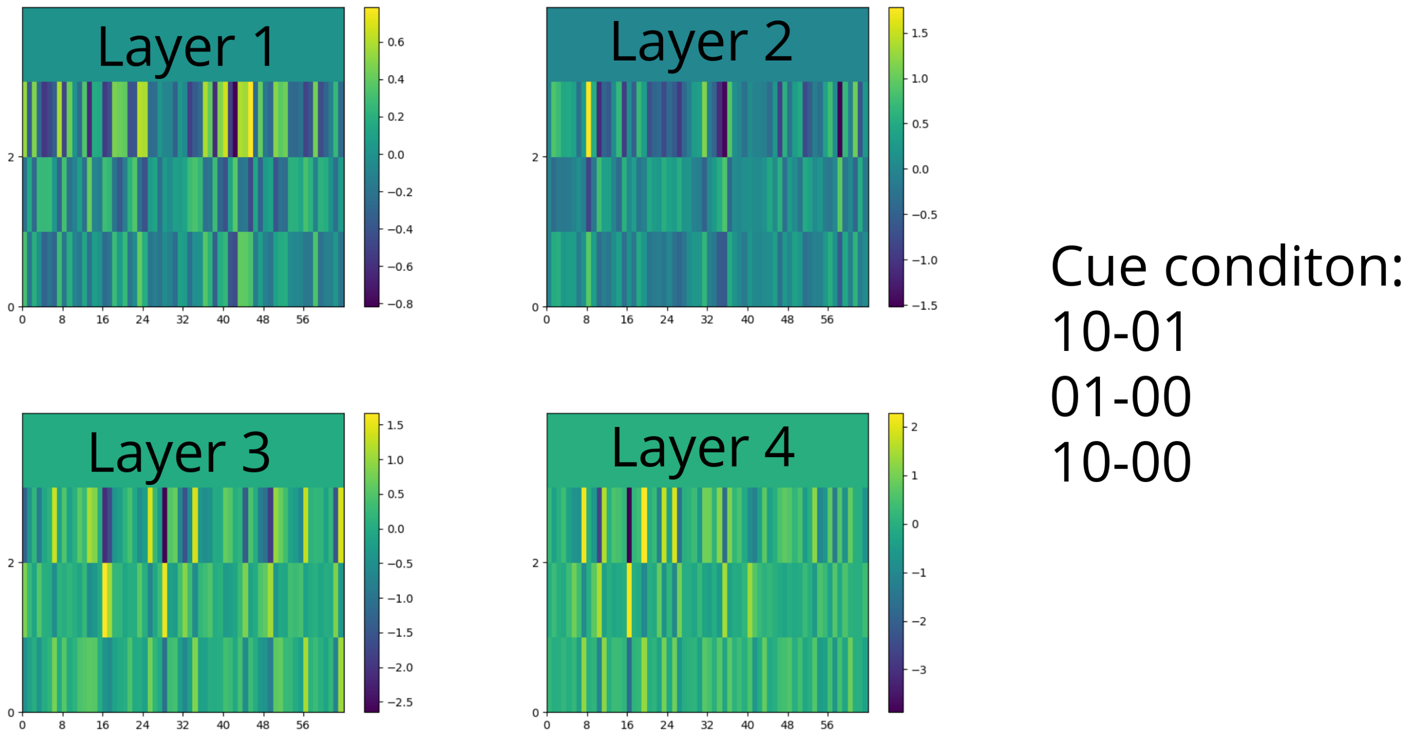


Figure 6. Difference in firing rates of individual layers given different conditions

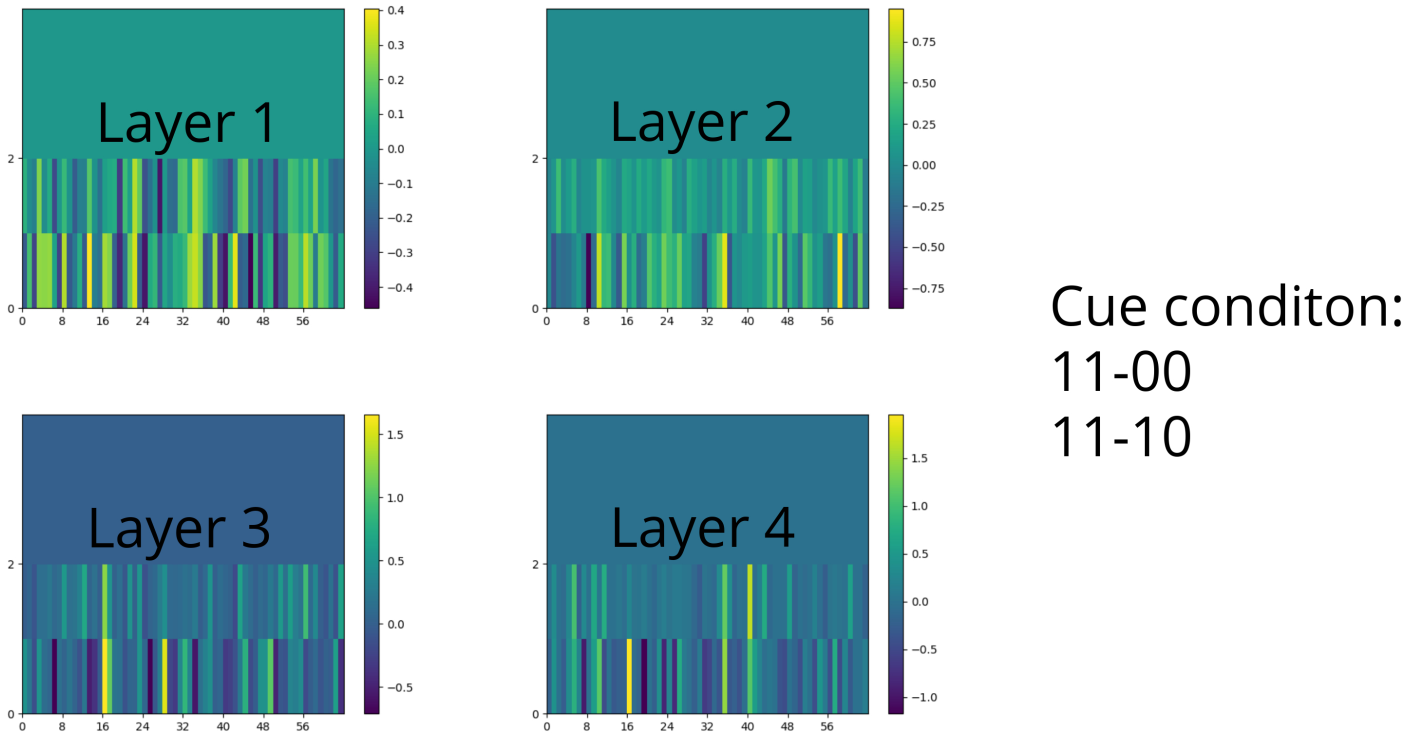


Figure 7. Difference in firing rates of individual layers given different conditions

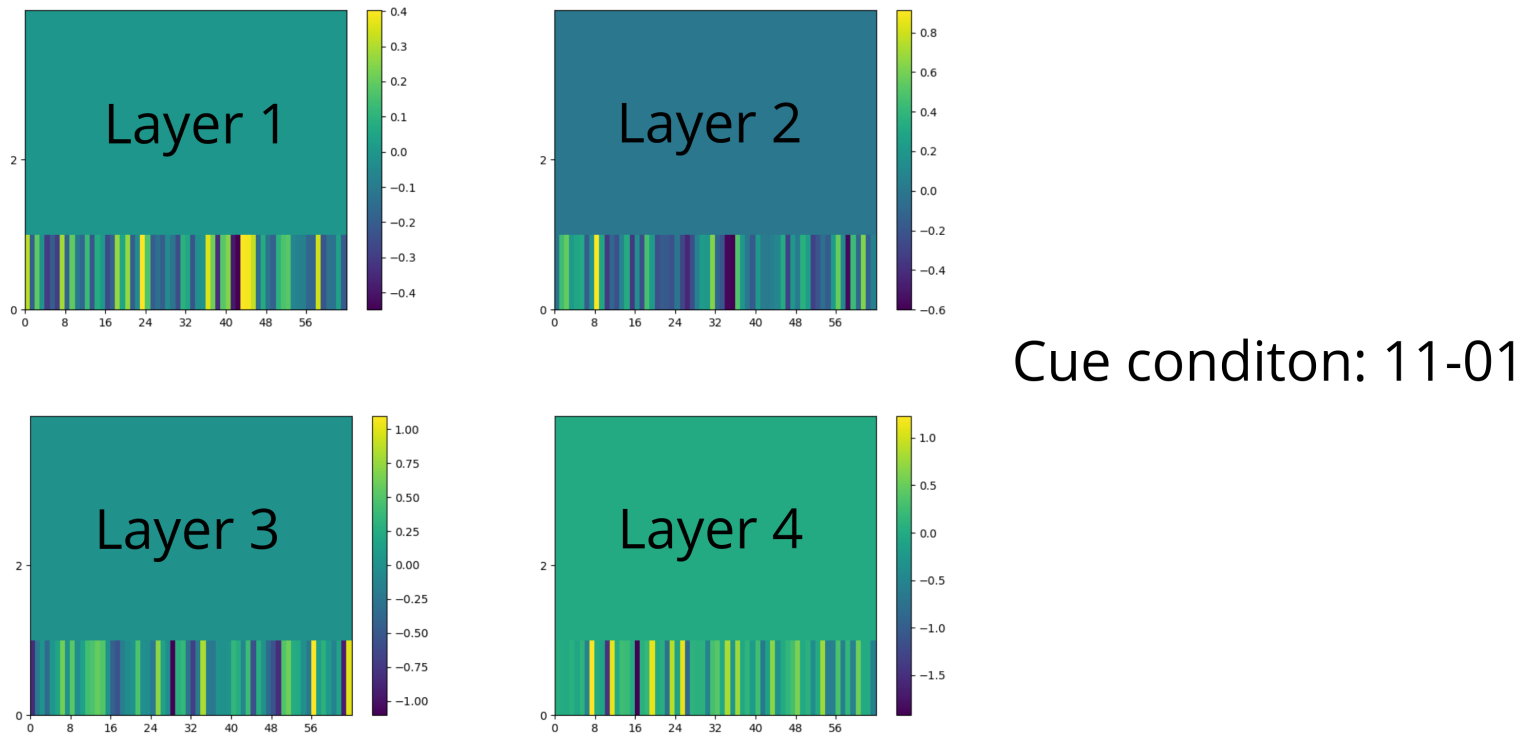


Figure 8. Difference in firing rates of individual layers given different conditions

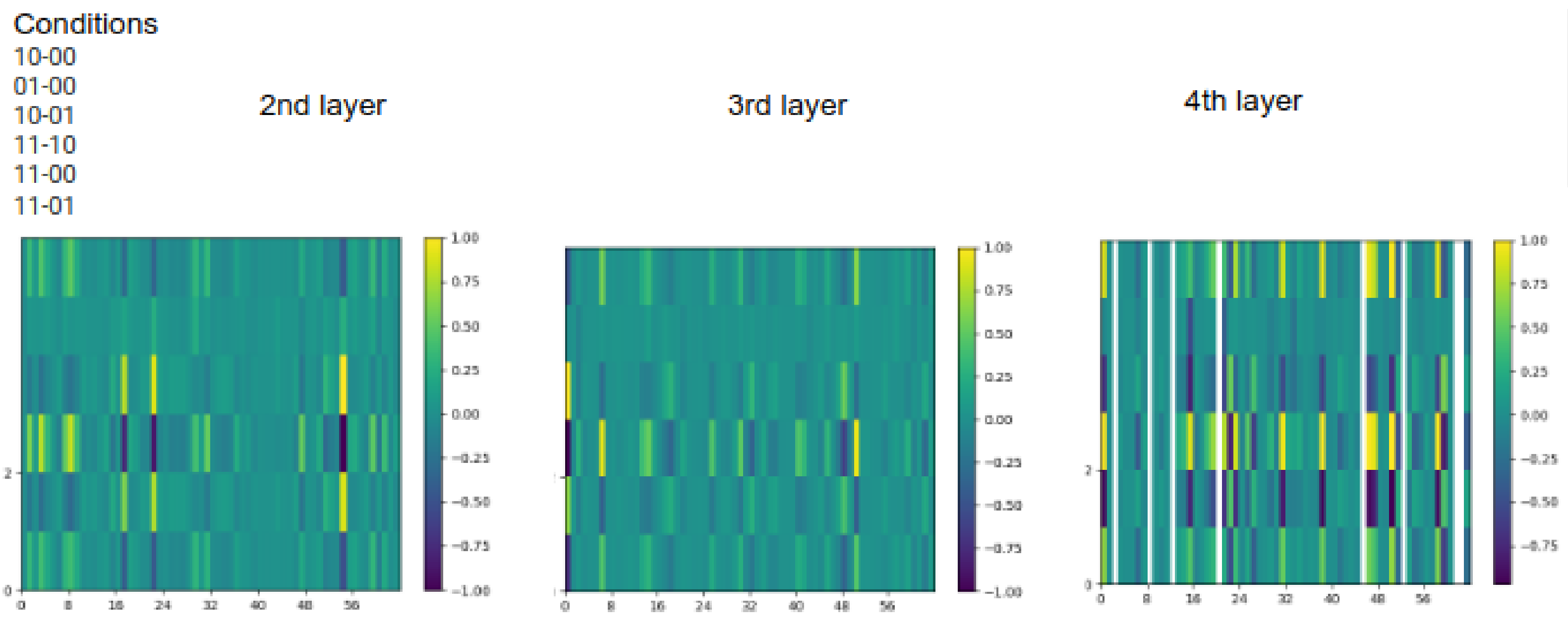


Figure 9. Difference in firing rates of individual layers given different conditions (every element divided by sum of the terms in difference)

4.1.2 Analysis II

Note: The observations for a subset of conditions are attached here as the trends are similar for all. Attaching all the figures is not feasible.

In the following figures, purple denotes 0 and yellow denotes 1.

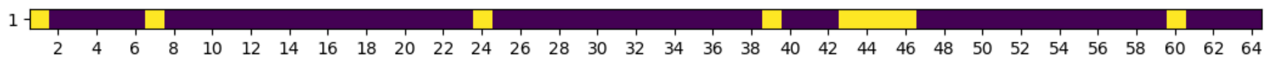


Figure 10. Processed activation of the first hidden layer with input as contrast cue

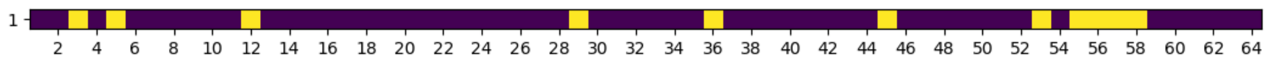


Figure 11. Processed activation of the second hidden layer with input as contrast cue

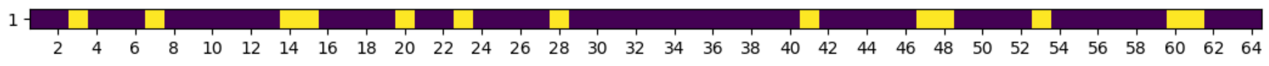


Figure 12. Processed activation of the third hidden layer with input as contrast cue

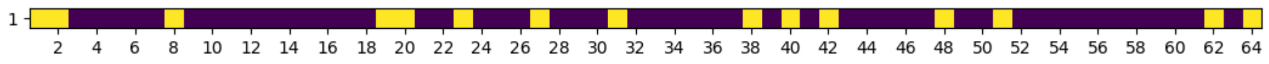


Figure 13. Processed activation of the fourth hidden layer with input as contrast cue

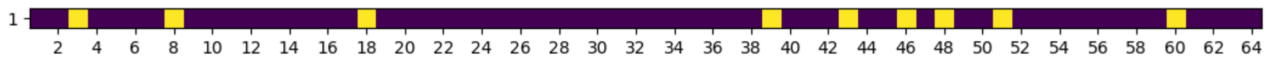


Figure 14. Processed activation of the first hidden layer with input as motion cue

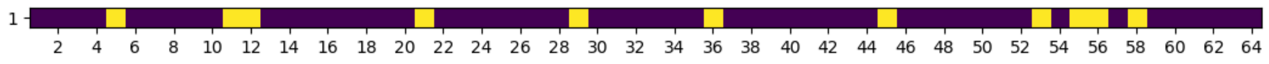


Figure 15. Processed activation of the second hidden layer with input as motion cue

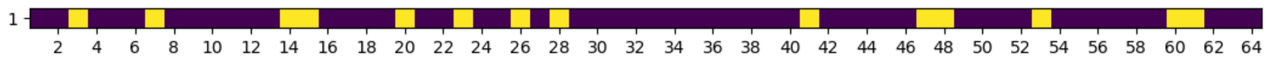


Figure 16. Processed activation of the third hidden layer with input as motion cue

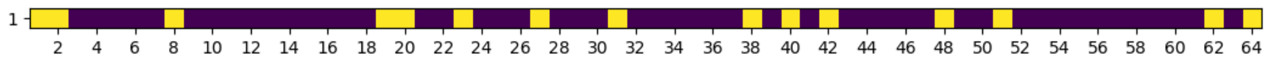


Figure 17. Processed activation of the fourth hidden layer with input as motion cue

In the following figures, purple denotes none of the networks contribituting to the firing of the neurons, blue signifies one of the network contributing, and yellow signifies both the networks contributing.

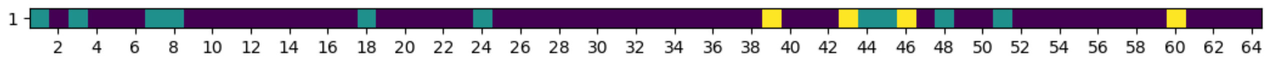


Figure 18. Processed activation of the first hidden layer with input as motion and contrast cue

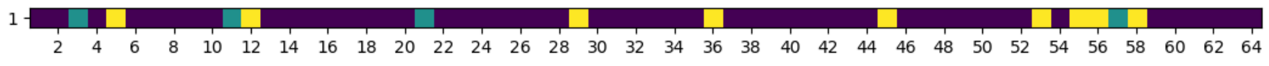


Figure 19. Processed activation of the second hidden layer with input as motion and contrast cue

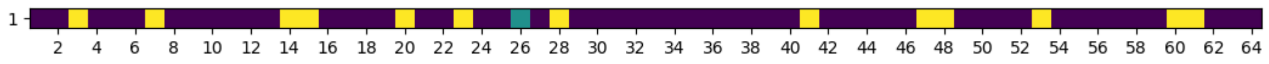


Figure 20. Processed activation of the third hidden layer with input as motion and contrast cue

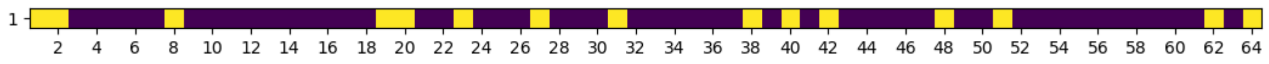


Figure 21. Processed activation of the fourth hidden layer with input as motion and contrast cue

In the initial layers, the number of neurons firing for an input are less than those compared to later layers. For two inputs set to 1, (out of A,B,C,D; refer to previous chapter for notation), the common neurons are more in the later layers as compared to the initial layers, i.e. blue colour reduces as we move to the higher layers.

Owing to the above the observations, it can be concluded that in the initial layers, the network stores the information for all inputs in different neurons, whereas the later layers have the same set of neurons firing for different outputs. Also, traditional ANNs are good at computing "AND" and "OR" functions, but not "XOR". The observations are similar to how XOR is achieved from these universal operations and our objective function is close to XOR (as B =A' in training) [See Appendix A]. The cue does not increase the passage of information for the corresponding data. This is seen from the fact that there is not much difference in activations when input is contrast cue + contrast data and contrast cue + motion data. Rather, the network stores these independently and learns to perform 2 different AND functions separately similar to the gate diagram of XOR [See Appendix A].

4.1.3 Analysis III

For the following graphs, columns denote contrast uncued, contrast cued, motion uncued, motion cued from left to right. The rows denote hidden layers 2 to 4 from top to bottom. Red denotes activations for original image and blue denotes activations for changed image. The aim for the analysis was to find the decoding axis. Since, the 2 clusters are very close to each other and not circular, it is difficult to do that. This analysis confirms the observations in Analysis II - the shape of the data in the plots tells us the contribution of the dimensions. Since, initial layers are closer to a circle, it shows that more dimensions i.e. more neurons contribute to the activation. So, the fact that activations of higher layers is sparse, is reinforced.

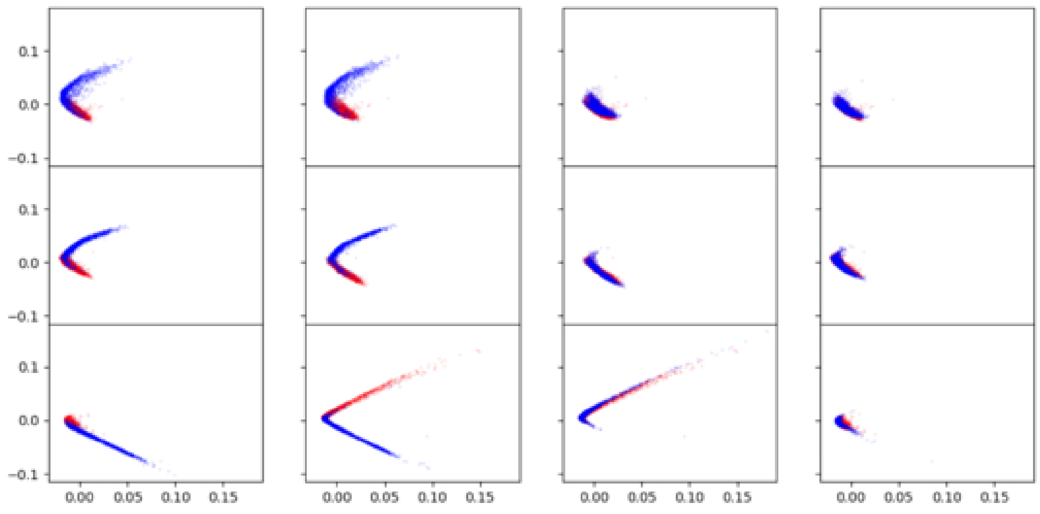


Figure 22. PCA applied on all samples for a hidden layers (across columns) Noise with std dev = 0.04

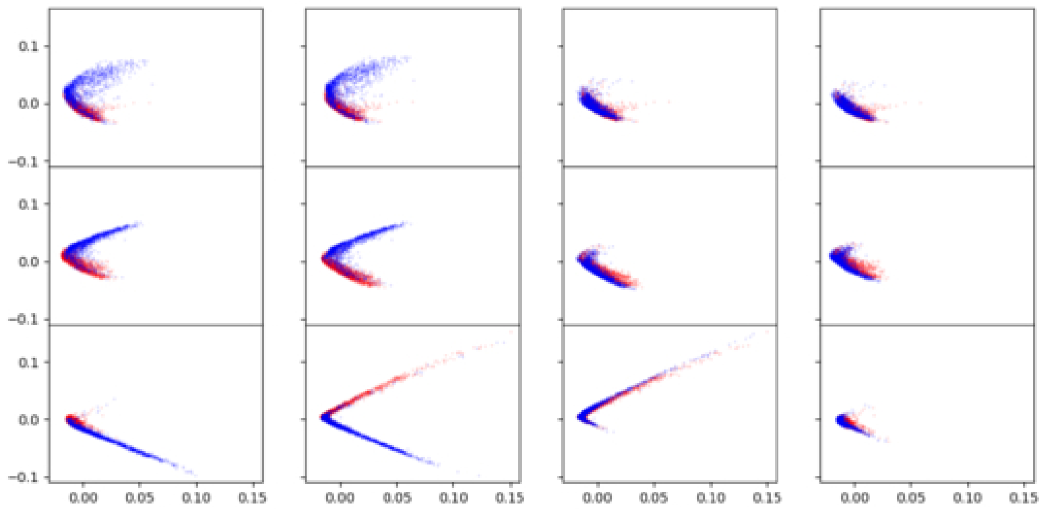


Figure 23. PCA applied on all samples for a hidden layers (across columns) Noise with std dev = 0.05

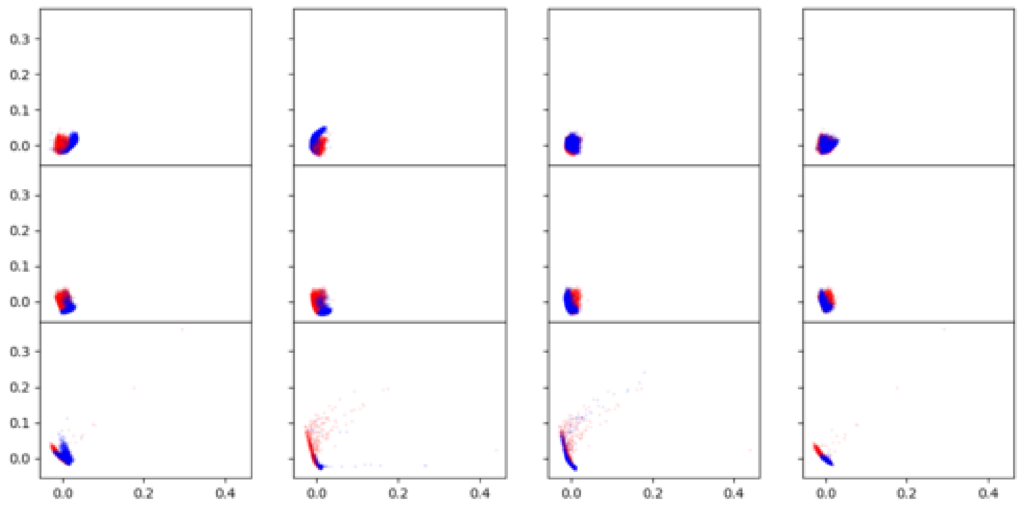


Figure 24. PCA applied on original image for each plot. Noise with std dev = 0.04

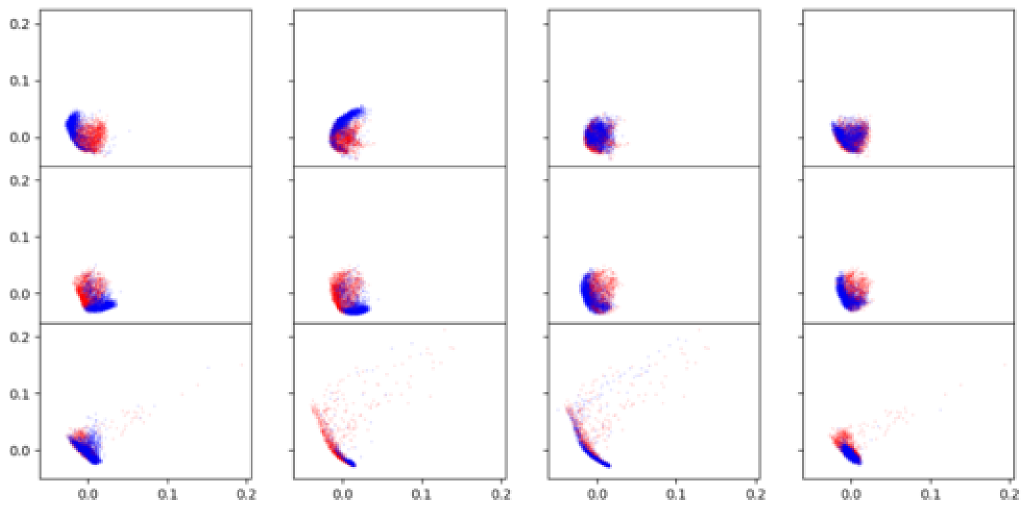


Figure 25. PCA applied on original image for each plot. Noise with std dev = 0.05

4.1.4 Decoding Analysis

0 signifies contrast and 1 signifies motion in all contexts.
CNF is the decoder trained with contrast data, contrast cue, on hidden layer of contrast network; CF is the decoder with the same conditions, but with feedback
MNF is the decoder trained with motion data, motion cue, on hidden layer of motion network, without feedback; MF is the decoder with the same conditions, but with feedback

In table 1, for decoders trained without feedback, the network is unable to discriminate between contrast and motion changes. The accuracy is around 86.66% for majority of the time, since the networks are predicting all the changes irrespective of the nature. The decoder trained on the motion network performs worse due to the encoding in the motion hidden layer being sparse as the motion discrimination task is easier as compared to contrast for the chosen architecture. Looking at the accuracies of the decoders with feedback, it is seen that the feedback works to decrease the accuracies of the decoders with a mismatch of the task and network used for decoding. It can also be seen that the contrast network is more affected by the feedback, which can also be noted from the fact that it had lower accuracy initially and in the final output (after the third input), the accuracy is same as motion.

In table 2, it can be seen that the feedback changes the task that the networks perform. The networks were trained for state discrimination, but the accuracies decrease with feedback. Since the network does not have information about the cue without feedback, the accuracies for decoders without feedback do not depend on the cue. The fact that contrast network is affected more by feedback is reinforced by this table as the accuracy drops more than motion (37.25% for contrast from 82%, and 64.58% for motion from 89%).

Decoder	Data	Cue	Network	Accuracy
CNF	0	0	0	86.66%
CNF	0	0	1	86.66%
CNF	0	1	0	86.66%
CNF	0	1	1	86.66%
CNF	1	0	0	86.66%
CNF	1	0	1	86.66%
CNF	1	1	0	86.66%
CNF	1	1	1	86.66%
MNF	0	0	0	80.47%
MNF	0	0	1	86.66%
MNF	0	1	0	80.37%
MNF	0	1	1	86.66%
MNF	1	0	0	80.98%
MNF	1	0	1	86.66%
MNF	1	1	0	80.67%
MNF	1	1	1	86.66%
CF	0	0	0	88.16%
CF	0	0	1	15.9%
CF	0	1	0	86.57%
CF	0	1	1	19.43%
CF	1	0	0	84.07%
CF	1	0	1	16.65%
CF	1	1	0	86.53%
CF	1	1	1	19.05%
MF	0	0	0	59.56%
MF	0	0	1	86.66%
MF	0	1	0	76.15%
MF	0	1	1	85.40%
MF	1	0	0	61.36%
MF	1	0	1	86.66%
MF	1	1	0	76.67%
MF	1	1	1	87.89%

Table 1. Accuracies for decoders for the change detection task

Decoder	Data	Cue	Network	Accuracy
CNF	0	0	0	81.2%
CNF	0	0	1	15.11%
CNF	0	1	0	81.2%
CNF	0	1	1	15.11%
CNF	1	0	0	12.43%
CNF	1	0	1	14.60%
CNF	1	1	0	12.43%
CNF	1	1	1	14.60%
MNF	0	0	0	16.87%
MNF	0	0	1	13.05%
MNF	0	1	0	16.87%
MNF	0	1	1	13.05%
MNF	1	0	0	12.39%
MNF	1	0	1	89.54%
MNF	1	1	0	12.39%
MNF	1	1	1	89.54%
CF	0	0	0	37.25%
CF	0	0	1	12.09%
CF	0	1	0	18.15%
CF	0	1	1	12.52%
CF	1	0	0	12.63%
CF	1	0	1	12.55%
CF	1	1	0	9.38%
CF	1	1	1	8.21%
MF	0	0	0	7.8%
MF	0	0	1	8.31%
MF	0	1	0	12.12%
MF	0	1	1	12.26%
MF	1	0	0	11.64%
MF	1	0	1	17.49%
MF	1	1	0	12.56%
MF	1	1	1	64.58%

Table 2. Accuracies for decoders for the state discrimination task

Chapter 5

Conclusion and Scope

The project aimed to train models which incorporate attention. Initiallt, two different types of attention were taken in the project - spatial attention and feature attention. Feature attention was analysed finally due to simpler architecture needed for it. Multiple models have been trained and analysed. The current models have to be analysed further and the architectures improved to incorporate more aspects of our visual attention.

It was seen that even though initial networks may have low accuracy, the choice network with feedback to the initial networks achieved a very high accuracy. The mechanism of working was analysed for the network without feedback. In the future, it can be studied that how feedback reduces accuracy for uncued tasks.

Studying attention is important for two main reasons. First, it will help in reducing the time and computational requirements for automation tasks as one network can learn to do multiple tasks as compared to specialised network for each. Second, it will help understand the functioning of the brain which will help in curing mental diseases like alzhimer's.

The approach to use neural networks for this study is effective as the brain is known to optimise functions, so if we model the structural connectivity of the brain using a network, then the learnt mathematically optical weights will give insight about the functional connectivity.

The final model trained models the lower areas of the visual processing area of the brain by the contrast and motion/orientation networks. It tries to simulates the pre-frontal cortex by the third network which chooses which task to perform depending on the cue. The feedback connections are also biologically plausible as different regions of the brain are jnown to be connected.

Concluding, the project assumed the physical connectivity of the brain and found the mathematically optimal functional connectivity for that task. We tried to analyse the learnt weights to analyse this learnt connectivity.

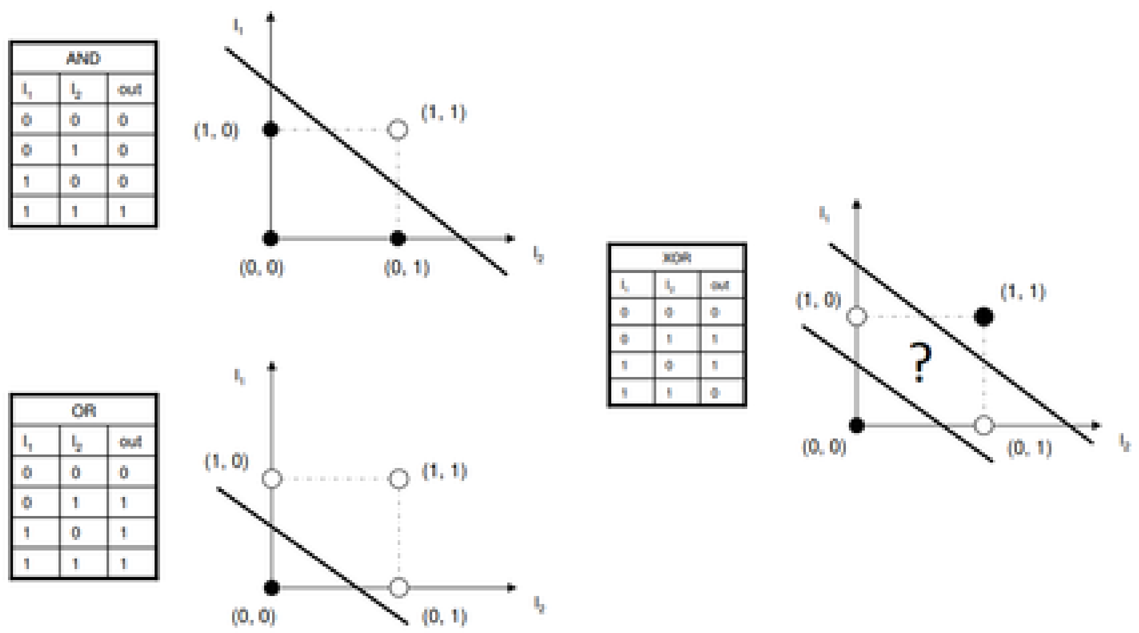
Appendix A

Neural Network Training and Logic Gates

The idea behind neural networks is simple. You have input and output pairs. You define a set of matrices that are multiplied to the input to give the output. The size of the matrices, the exact equations used define the architecture of the network. The matrices are initialised randomly initially. Then, you define a cost function that tells you the difference between your output with random matrices and the desired output. You update your matrices using the gradient of this cost function w.r.t. the matrix elements. Iterating through this process you receive the required weight matrices.

Now, AND and OR functions are easier for a network to learn because they are linearly separable when you plot them. XOR is not linearly separable and requires two lines to separate the points. So, a single layer cannot learn an XOR function. Figure 26 shows this.

Mathematically to achieve XOR from AND and OR is the logic gate architecture. Comparing our model to this, it is observed that the number of layers required can be justified by this and all the analysis can be tied together to explain the possible working of the network.



AND		
I_1	I_2	out
0	0	0
0	1	0
1	0	0
1	1	1

OR		
I_1	I_2	out
0	0	0
0	1	1
1	0	1
1	1	1

XOR		
I_1	I_2	out
0	0	0
0	1	1
1	0	1
1	1	0

Figure 26. Graphs of OR, AND and XOR functions with two two inputs; ***figure taken from:https://medium.com/@lucaspereira0612/solving-xor-with-a-single-perceptron-34539f395182***

References:

[1] Warasinee Chaisangmongkon, Sruthi K. Swaminathan, David J. Freedman, and Xiao-Jing Wang. Computing by Robust Transience: How the Fronto-Parietal Network Performs Sequential, Category-Based Decisions. Neuron 93, 1504–1517

[2] Francesca Mastrogiuseppe and Srdjan Ostojic. Linking Connectivity, Dynamics, and Computations in Low-Rank Recurrent Neural Networks. Neuron 99, 609–623

[3] Douglas A. Ruff and Marlene R. Cohen. Simultaneous multi-area recordings suggest that attention improves performance by reshaping stimulus representations. Nature Neuroscience, 22, pages1669–1676(2019)

[4] Daniel Birman and Justin L. Gardner. A flexible readout mechanism of human sensory representations. Nature Communications (2019) 10:3500

[5] Valerio Mante, David Sussillo, Krishna V. Shenoy and William T. Newsome. Context-dependent computation by recurrent dynamics in prefrontal cortex. Nature, 503, pages78–84(2013)

[6] Guangyu Robert Yang, Madhura R. Joglekar, H. Francis Song, William T. Newsome and Xiao-Jing Wang. Task representations in neural networks trained to perform many cognitive tasks. Nature Neuroscience, 22, pages297–306(2019)

[7] A. Emin Orhan and Wei Ji Ma. A diverse range of factors affect the nature of neural representations underlying short-term memory. Nature Neuroscience - Vol 22, 2019 -pages 275–283

[8] Nicolas Y. Masse, Guangyu R. Yang, H. Francis Song, Xiao-Jing Wang and David J. Freedman. Circuit mechanisms for the maintenance and manipulation of information in working memory. Nature Neuroscience, 22, pages1159–1167(2019)

[9] Jogendra Nath Kundu, Srinivas K., R. Venkatesh Babu, Devarajan Sridharan. A biologically-inspired sparse, topographic recurrent neural network model for robust change detection. NIPS 2017 Workshop on Cognitively Informed Artificial Intelligence