Imperial College London

Department of Computing

A Neural Model of Psychotherapy Motivated by Attachment and CBT

by

Zheng Lin

Submitted in partial fulfillment of
the requirements for the MSc Degree in
Computing Science / Artificial intelligence
of Imperial College London

September 6th, 2013

Abstract

We develop an emotionally influenced goal-directed decision making model based on Levine's architecture for cognitive-emotional decision making in the human animal. Employing the notion of strong attractor in Hopfield network, and integrating with the above model, we model attachment types and how they undergo change in psychotherapy. We also study how the degrees of strong patterns influence its stability in the present of other random patterns. Finally, we introduce a feedback control theory to mimic the control from orbitofrontal cortex to the amygdala, of which the outputs are influenced by heuristic signals when the cortex control is absent.

Acknowledgements

In doing this project, I have been fortunate to be supervised by Professor Abbas Edalat, without whose help this work would never happen.

I am also grateful to my friends, Kaiyan Xiao, Bingqin Qu, Siyi Jiang, Shixiong Da, Shimei Xiao, without whom my college life would lose its joy.

Table of Contents

Chapter 1 1

Introduction 1

1.1 Introduction and Motivation 1

1.2 Report Structure 2

Chapter 2 3

Hopfield Network 3

2.1 Background Review 3

2.2 Explore the Model 10

Chapter 3 14

Restricted Boltzmann Machine 14

3.1 Background Review 14

3.2 Training RBMs 17

Chapter 4 20

Decision Making Network 20

4.1 The Network of Needs 20

• Modelling the Network of Needs with Hopfield Network 22

4.2 The Network of Decision Rules 24

4.3 Connection of Needs network and Decision network 30

Chapter 5 34

Reinforcement Learning 34

5.1 Psychotherapy 34

5.2 Q-learning 35

5.3 Design of Q-learning 36

5.4 Experiments 38

Chapter 6 42

Bring it All Together 42

Chapter 7 45

Evaluation and Conclusion 45

7.1 Evaluation 45

7.2 Future Work 46

7.3 Conclusion 47

Chapter 1

Introduction

1.1 Introduction and Motivation

The World Health Organization (WHO) reports that there are over a third of people in most countries suffering from mental disorders, which has become the major issue of our society. A number of types of psychotherapy have been developed to deal with emotional problems and mental health conditions and have been proven to be effective, such as psychodynamic psychotherapy (Shedler, 2010) and cognitive behavioral therapy (Rothbaum, 2000). Furthermore, some researchers suggested that it is the time to model the behavior of the psychotherapy (Wedemann, 2009) mathematically.

The Hopfield network (Hopfield, 1982), based on Hebbian's rule for learning (Hebb, 1949), emerges to model a simple conceptual brain model. Edalat (2013a) and his students, via mathematically approaches and computer simulations, proved that the strong attractors are more stable than simple attractor in the Hopfield model and studied the relation between different strong attractors to show how a strong attractor could be replaced by other strong attractors. He proposed that strong attractors could be used to model attachment types in psychotherapy, and change between strong attractors could explain how attachment types undergo change. Such change can be achieved by reinforcement learning reinforcing the positive attachment types or behaviors because the brain is plastic and its structure can change during development. The best-known learning rule is spike time dependent plasticity (STDP), which can be thought of reinforcement learning in conjunction with a reward signal。

The RBMs, a simplified form of Boltzmann machines (Hinton, 1986), is famous of its high efficiency and high effectiveness. It has successfully modelled different types of data, such as unlabelled images (Hinton, 2006), user rating of movies (Salakhutdinov, 2007), temporal signals (Mohamed, 2010). We propose here that RBMs are able to category some metaphorical forms of emotional signals in one's brain.

The motivation of this project is, using computer technology, to build a brain model that mimics the change between super attractors under reinforcement learning, and the influence of the dominant attractors on subcortical decision making. We aim to use artificial neural networks (Hopfield model and RBM) to construct an artificial brain based on the brain model proposed by Levine (2009), which successfully describes an emotionally influenced, goal-directed decision making network in the human animal.

1.2 Report Structure

This report consists of an introduction chapter that has shown above and four main chapters:

- Chapter2 constitutes a summary of background knowledge about content-addressable memory and Hopfield network, and an experiment on the study of stability of strong attractors in Hopfield model.
- This report consists of an introduction chapter that has shown above and four main chapters:
- Chapter3 constitutes a summary of background knowledge about restricted Boltzmann machine and descriptions of a RBM we trained for this project.
- Chapter4 give a full description of Levine's emotionally influenced goal-directed decision making model, and full description of how my model copes with Levine's.
- Chapter5 constitutes a summary of background knowledge about psychotherapy and Q-learning algorithm. We also put a full description of Q-learning algorithm designed to mimic the behavior of psychotherapy. We also studied the difference between the model with and without the control from OFC.
- Chapter6 describes the entire system containing all components we designed in previous chapters, and it also describes how these components communicate with each other.
- In Chapter7 we look our project back in a critical way, and suggest some points that can be improved in the future.

Chapter 2

Hopfield Network

2.1 Background Review

- **The associative memory**

The Hopfield network can be thought of associative memory, where the network stores a set of p patterns ξ_i^μ, and recalls the most appropriate pattern when a stimulus, ζ_i, to the network is presented. The patterns are labeled by $\mu = 1, 2, \dots, p$, and the neurons in the network are label by $i = 1, 2, \dots, N$. The most appropriate pattern recalled by the network is the pattern that is closet to ζ_i. These stored patterns are called attractors in the space of all possible states of the network (see Fig 2.1). Different attractors have their own basins of attraction in the whole configuration space. The configuration states of the network will eventually converge into one of the attractors, as shown by the trajectories sketched. In the Hopfield network, the stored patterns are not the only attractor, and there are others such as spurious states, which will be discussed in the following section.

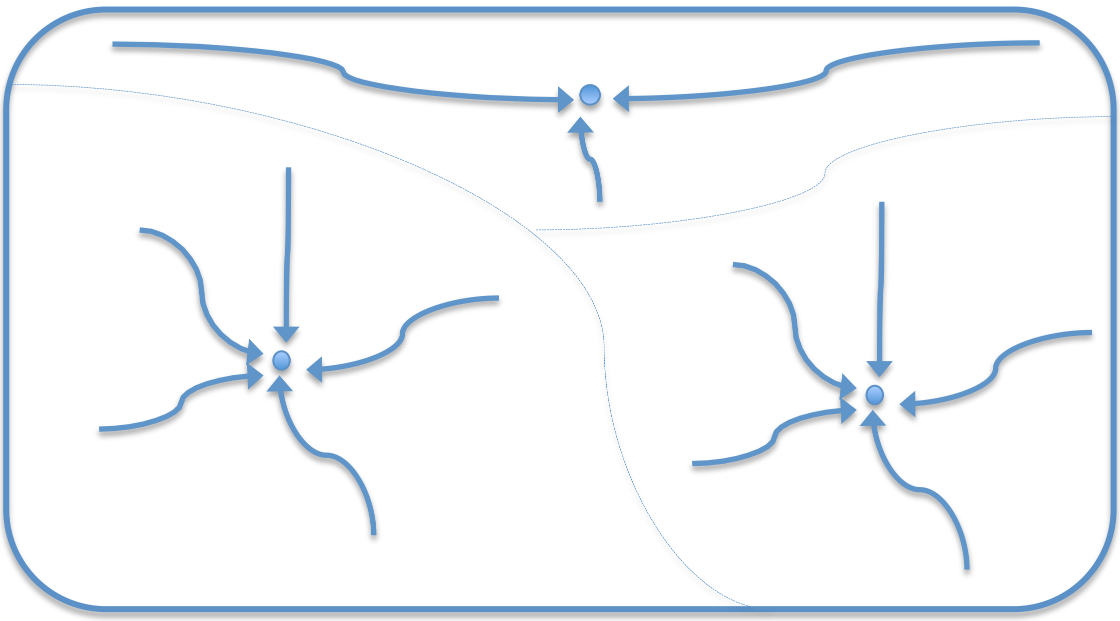


FIGURE 2.1 Schematic configuration space of a model with three attractors.

- **Hopfield model**

A basic Hopfield network is a form of single-layered recurrent network of artificial neurons, G = ⟨V, E⟩ , which comprises a set V of nodes (or McCulloch-Pitts neurons) and a set $E \subseteq V \times V$ of edges (or connections). Each node of the network is connected symmetrically to any other node in V except for the node itself, that is, the network forbids self-connections. A connectivity matrix W can be generalized to account for different connection strength, which can be +1 (firing), -1 (not firing), or 0, accounts for disconnected pairs. The model neuron receives weighted outputs from other neurons, and yields a +1 or -1 signal according to whether the sum of the received signals is above or below a certain threshold, leading to the dynamics of the network to read

$$S_i \coloneqq sgn\left(\sum_j \omega_{ij} S_j - \theta_i\right) \tag{2.1}$$

where the sign function $sgn(x)$ is taken to be

$$sgn(x) = \begin{cases} 1 & if\ x \geq 0; \\ -1 & if\ x < 0; \end{cases} \tag{2.2}$$

and the threshold θ_i is dropped, taking $\theta_i = 0$, because it makes no sense if the random patterns are considered to be stored in the network. Thus Equation (2.1) becomes

$$S_i \coloneqq sgn\left(\sum_j \omega_{ij} S_j\right) \tag{2.3}$$

The neurons of the network can be updated either synchronously or asynchronously. The synchronous updating rule updates all units simultaneously at each time step, while the asynchronous updating rule updates them one at a time. For this project, asynchronous updating rule is chosen because it is more natural for both brains and artificial networks. It will proceed in the way:

At each time step, a unit will be selected randomly, and rule (2.3) is applied to the unit.

- **"Hebb rule" learning phase**

During a learning phase, the network memorizes desired patterns, ξ^μ, by changing the connectivity matrix according to

$$\omega_{ij} = \frac{1}{N} \sum_{\mu=1}^{p} \xi_i^\mu \xi_j^\mu \tag{2.4}$$

where p is the total number of stored patterns labeled by μ.

As the hypothesis made by Hebb that "units that fire together, wire together" has similarity with (2.4), this equation is usually called the "Hebb rule". The "Hebb rule" Equation (2.4) accounts directly for the correlation between the firing of the pre/post neurons. However, Equation (2.4) is probably not physiologically reasonable because neither of the units is firing ($\xi_i^\mu = \xi_j^\mu = -1$) can change the weights positively. In this project, we use the unchanged form.

- **Stability**

The analysis of the stability of all stored patterns in the network is important because the configuration of the system is supposed to converge to some stable attractor that is near to the initial configuration. A particular pattern ξ_i^v is said to be stable if it satisfies the stability condition

$$sgn(h_i^v) = \xi_i^v \text{ (for all } i) \tag{2.5}$$

where the net input to unit i in pattern v, h_i^v, is the sum of weighted output from other neurons

$$h_i^v \equiv \sum_j \omega_{ij}\xi_j^v \tag{2.6}$$

and the Hebb rule allows to substitute ω_{ij} with $\frac{1}{N}\sum_{\mu=1}^{p}\xi_i^\mu\xi_j^\mu$, so that

$$h_i^v = \frac{1}{N}\sum_j\sum_\mu \xi_i^\mu\xi_j^v\xi_j^v \tag{2.7}$$

If we extract the special term $\mu = v$ from the sum on μ, the net input becomes

$$h_i^v = \xi_i^v + \frac{1}{N}\sum_j\sum_{\mu\neq v}\xi_i^\mu\xi_j^\mu\xi_j^v \tag{2.8}$$

If the second term, which is call the crosstalk term, were small enough or zero, pattern v could be concluded to be stable immediately according to the stability condition described above, because the sign of h_i^v will not change if the magnitude of the crosstalk term is small enough. Therefore, the number of stored patterns p is supposed to be small enough (compared to the size of the system, N) to ensure that for all i and v the crosstalk term can be small enough, that is, all stored patterns are stable. Determining the maximum number of stored patterns refers to the analysis of storage capacity.

- **Storage capacity**

The storage capacity p_{max} of the network is the fundamental analysis of the Hopfield network. It refers to the maximum number of patterns that can be stored with acceptable errors, in terms of P_{error}, which is the probability of error per bit. To understand the term P_{error}, let us multiply the crosstalk term by $-\xi_i^v$, which is desired, obtaining the quantity

$$C_i^v = -\xi_i^v \frac{1}{N}\sum_j \sum_{\mu\neq v} \xi_i^\mu \xi_j^\mu \xi_j^v \qquad (2.9)$$

If unit i of pattern v is unstable, C_i^v is positive and larger than 1, which leads to the change of the sign of h_i^v. For each unit of stored patterns that are generated randomly, the probability P_{error} that this unit is unstable:

$$P_{error} = Prob(C_i^v > 1) \qquad (2.10)$$

With the assumption that both N and p are large compared to 1, the distribution of values for C_i^v is subject to a binomial distribution with mean zero and variance $\sigma^2 = p/N$. But this distribution can be thought of a Gaussian distribution with the same mean and variance (see Fig 2.2). Therefore, the probability P_{error} is defined as

$$P_{error} = \frac{1}{\sqrt{2\pi}\sigma}\int_1^\infty e^{-x^2/2\sigma^2} dx \qquad (2.11)$$

and then it can be expressed in terms of N and p

$$P_{error} = \frac{1}{2}[1 - \mathrm{erf}\left(\sqrt{\frac{N}{2p}}\right)] \qquad (2.12)$$

where the term $\mathrm{erf}(x)$ is called error function and it is defined by

$$\mathrm{erf}\,(x) \equiv \frac{2}{\sqrt{\pi}}\int_0^x \exp(-u^2)\,du \qquad (2.13)$$

Once P_{error} is defined in terms of p and N, it is necessary to choose a criterion for acceptable performance so that we can find the maximum number of patterns that can be stored in the system with little bits slip compared to the desired pattern. For instance, if we accept that 0.37% of the bits are unstable initially, the maximum number of patterns can be $p_{max} = 0.138N$. The whole procedure above is based on the assumption that all stored patterns are generated randomly. However, realistic patterns will be somewhat correlated, so that the capacity of the system will decrease dramatically.

- **The Energy function**

Based on the two assumptions that symmetric connections and no self-connections, Hopfield introduced the idea of an energy function into the Hopfield network. The energy function is a Lyapunov function that decreases during the system evolves until its states reach some local minimum (stored pattern), and it is expressed as following:

$$H = -\tfrac{1}{2}\textstyle\sum_{ij} \omega_{ij} S_i S_j. \qquad (2.14)$$

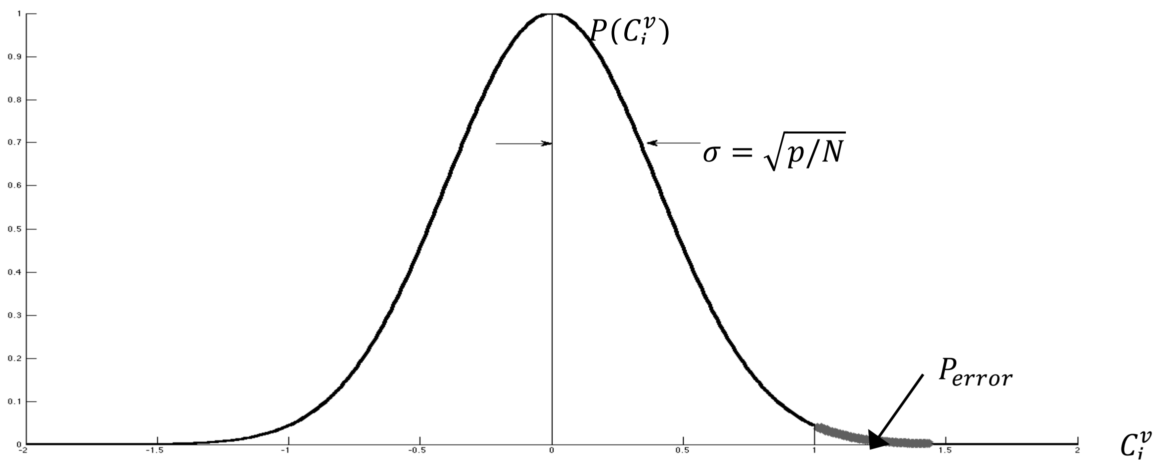


FIGURE 2.2 The distribution of values for C_i^v given by (2.13). For p random patterns and N units this is a Gaussian function with mean zero and variance $\sigma^2 = p/N$. The brushed area is P_{error}, the probability of error per bit.

And for symmetric connections Equation (2.14) can be separated into a term accounting for self-connections (ii terms) and a term accounting for the distinct pairs ij

$$H = C - \sum_{(ij)} \omega_{ij} S_i S_j. \qquad (2.15)$$

The new value S_i' of S_i is calculated using equation (2) for a particular i; the new value, H', of the network after updating is as follows:

$$H'(S) = -\frac{1}{2} \sum_{i \neq j} \omega_{ij} S_i' S_j. \qquad (2.16)$$

If the state of the unit is unchanged, $H'(S) - H(S) = 0$, otherwise, $S_i' = -S_i$, and the following is applied:

$$H'(S) - H(S) = -\frac{1}{2} \sum_{i \neq j} \omega_{ij} S_i' S_j + \frac{1}{2} \sum_{i \neq j} \omega_{ij} S_i S_j$$

$$= S_i \sum_{i \neq j} \omega_{ij} S_j$$

$$= S_i \sum_j \omega_{ij} S_j - \omega_{ii} \qquad (2.17)$$

Because S_i' and the term $\sum_j \omega_{ij} S_j$ have the same sign, the left term must be negative. The right term is also negative because the Hebb rule defines ω_{ii} as the ratio of the number of stored patterns to the size of the network. Therefore, the energy of the network will always decrease until the network reaches a steady state (i.e., attractor) with its dynamical updating rule.

For simplicity, we assign $\omega_{ii} = 0$. It makes no significant difference to the stability of the patterns when the network size is very large. Additionally, it is proven (Kanter and Sompolinsky, 1987) that assuming $\omega_{ii} = 0$ prevents producing stable spurious states near a desired attractor.

- **Spurious states**

The Hebb rule shows that there are local minima of the energy function, i.e., attractors, at the desired pattern, ξ^{u}. However, linear combinations of an odd number of patterns results in a stable mixture state, ξ_i^{mix}, leading to poor performance of the system.

Consider the simplest case of these mixture states, the symmetric combinations of three stored patterns as follows:

$$\xi_i^{mix} = sgn(\pm\xi_i^{\mu 1} \pm \xi_i^{\mu 2} \pm \xi_i^{\mu 3}). \qquad (2.18)$$

It means that the ξ_i^{mix} has the same sign as $\xi_i^{\mu 1}$ if and only if $\xi_i^{\mu 2}$ and $\xi_i^{\mu 3}$ both are opposite to $\xi_i^{\mu 1}$. The Hamming distance from ξ_i^{mix} to $\xi_i^{\mu 1}$ is therefore N/4, where N is the size of the network.

- **The "temperature"**

The analogue of temperature can be introduced to network theory to use stochastic units. This is an idea borrowed from some simple models of magnetic materials in statistical physics. For deterministic units we have its dynamics that:

$$S_i := \begin{cases} +1 \text{ if } h_i \geq 0 \\ -1 \text{ otherwise} \end{cases}, \qquad (2.19)$$

where h_i is the input to unit i, while the stochastic rule:

$$S_i := \begin{cases} +1 \quad \text{with probability } g(h_i); \\ -1 \text{ with probability } 1 - g(h_i). \end{cases} \qquad (2.20)$$

The pseudo temperature, T, characterises the function $g(h)$ and the sigmoid-shaped function is adopted to represent $g(h)$

$$g(h) = f_\beta(h) \equiv \frac{1}{1+\exp(-2\beta h)} \qquad (2.21)$$

where β is related to T by

$$\beta = \frac{1}{T}. \qquad (2.22)$$

The sigmoid function with different values of β is illustrated in Fig 2.3. As

$$1 - f_\beta(h) = f_\beta(-h) \qquad (2.23)$$

the dynamic rule (2.19) can be of the symmetrical form

$$\mathrm{Prob}(S_i = \pm 1) = f_\beta(\pm h_i) = \frac{1}{1+\exp(\mp 2\beta h_i)}. \qquad (2.24)$$

As shown in Fig 2.3, we can find that when the value of the pseudo temperature $T \to 0$, the system dynamics become deterministic.

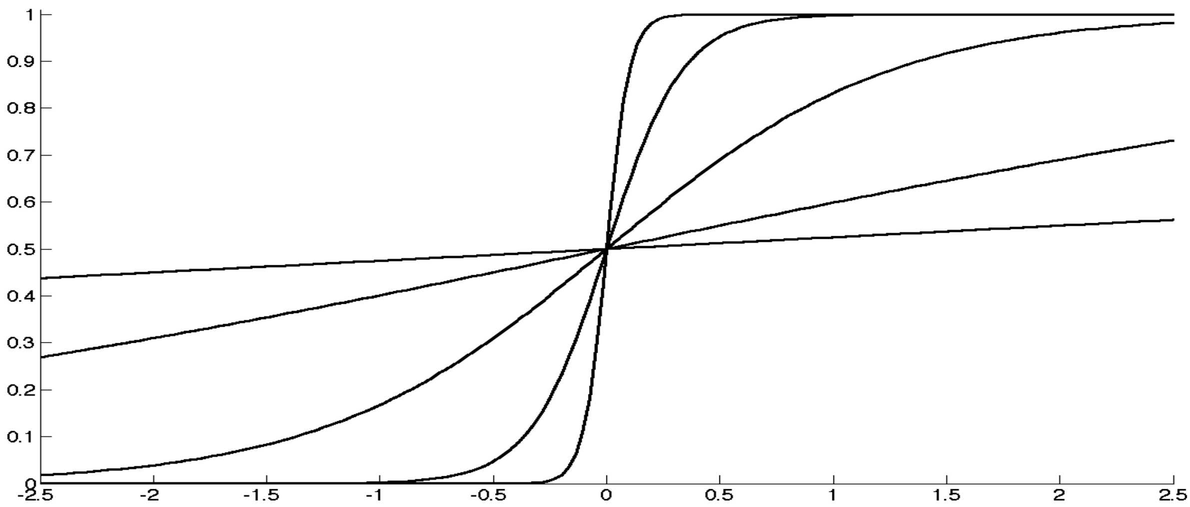


FIGURE 2.3 The sigmoid function $f_\beta(h)$ for several values of β.

- **Strong attractors of Hopfield network**

A strong attractor of Hopfield neural network is a pattern that has been stored d times in the network. We denote that the degree of this pattern μ is d_μ, which means that pattern ξ_μ occurs d_μ times in the system that stores p patterns in total. Pattern ξ_μ is thought of a strong pattern if $d_\mu > 1$; otherwise the stored pattern is called sample pattern and its corresponding attractor is canned simple attractor. The properties, such as stability and the basin in the energy space, of the strong pattern have been examined by using fundamental mathematical techniques and various simulations (Edalat and Mancinelli, 2013a).

The analysis of stability of strong patterns is similar to the analysis of stability of strong patterns: a normal distribution is introduced to obtain P_{error}, the probability of error per bit. For a single strong pattern with degree d_1, as $N \to \infty$, P_{error} is given by

$$P_{error} = \frac{1}{2}\left[1 - \mathrm{erf}\left(d_1\sqrt{\frac{N}{2(p-1)}}\right)\right], \qquad (2.25)$$

where the error function $\mathrm{erf}(x)$ has been explain above.

From Equation (2.25), it is clear that even though the total number of stored pattern p is much greater than the size of the system N, where the classical storage capacity is greatly exceeded, the probability that the strong pattern ξ^1 was stable is still very high if d_1 is large enough (the capacity becomes $p = 0.138 \times N \times d_1^2$, which has been proven by Edalat (2013b)). And it is also confirmed that the higher the degree the bigger the basin size of the strong pattern. Therefore, if there exists multiple strong patterns, the system configuration will eventually converge to the higher degree attractor with higher probability.

- **Modelling attachment types using strong attractors**

The main objective of this section is to model attachment types by using strong attractors in Hopfield network. In my project, different attachment types are interpreted as six basic emotions: anger, disgust, fear, happiness, sadness and surprise. In training phase, a Hopfield network stores these basic emotions as binary patterns. In addition, the changes that they can experience are modelled.

Before we build the model, some properties of the system is examined. An experiment is set to analyse the relation between the degree of strong patterns and the maximum number of stored patterns, in terms of the similarity of the input pattern and the recalled pattern, with respect to hamming distance. Also, we study the performance of the Hopfield network when two strong patterns have the same degree.

2.2 Explore the Model

Six simple smiley images corresponding to the six basic emotions are treated as the stored patterns in the Hopfield network. These smiley images are useful since they somewhat represent the abstract concept of different attachment types.

In this experiment the six emotion images are stored as the strong patterns with p_r simple patterns that are greeted randomly, and I have chosen the anger image as the strongest pattern in the initial Hopfield network. This initial Hopfield network represents the memory that is inherited from the early relationship with primary caregiver in infancy.

The six smiley images are that of an anger, a disgust, a fear, a happiness, a sadness and a surprise. These are binary images are of dimensions 25×25 pixels, as shown below:

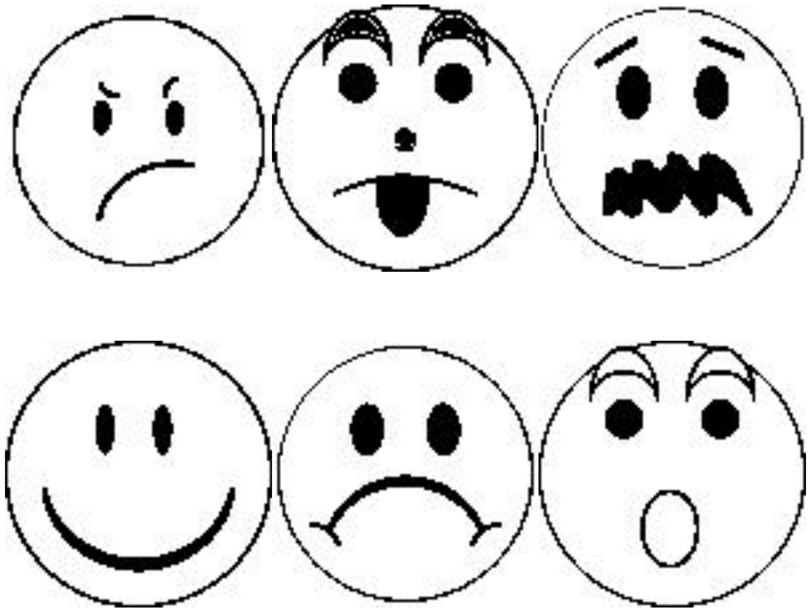


FIGURE 2.4 Anger, disgust, fear, happiness, sadness and surprise smiley faces.

- **The Network**

The Hopfield network is composed of 625 McCulloch-Pitts neurons (one neuron corresponding to one pixel of the image) and edges connecting each pair of them. The updating rule is both deterministic (i.e. with pseudo temperature $T = 0$), and asynchronous (updating them neurons one at a time). This experiment will be carried out in two case: storing identical patterns and storing similar patterns. In the case that storing identical patterns, this network is storing happiness face for three times, sadness for two times, and each of disgust, fear and surprise once. In addition, as the strongest pattern, the degree of anger face k will be greater than four. And the number p_r of stored patterns that are generated randomly is chosen carefully so that the storage capacity does not exceed. For the case that storing similar patterns, we will firstly generate a cluster of each pattern by adding random noise on the original one, and store the random perturbation of the strong attractors instead of the original pattern.

In the training phase, the changes of the weights ω_{ij} are subject to the standard Hebb rule, and the assumptions that $\omega_{ij} = \omega_{ji}$ and $\omega_{ii} = 0$ are applied.

- **Storing Identical Patterns**

In this experiment, the original smiley images are stored in the Hopfield network, and the degree of each pattern is:

- Happiness: 3
- Sadness: 2
- Disgust: 1

- Fear: 1
- Surprise: 1

The degrees of patterns above are constant, while the degrees of anger pattern and random pattern, notated as p_k and p_r respectively, are varying so that we can find the relation between the stability of strong patterns and its degree.

The test image will be a random image, with equal probability for being firing and not firing for each unit.

For each combination of p_k and p_r, the experiment is repeated for 20 times. And the number of update iteration will be up to 100. We determine the stability of the strong pattern by measuring the Hamming distance between the recalled pattern and the strongest pattern in the network. The results are displayed in Fig 2.5.

- **Comments**

As expected, the stability of a strong pattern decreases as the degree of other random patterns increases. The strong pattern can always stabilized if its degree is large enough, even in present of very high degree of random patterns. According to this result, we can propose that the strong attractors in a Hopfield network can be used to model the attachment type in developmental psychotherapy.

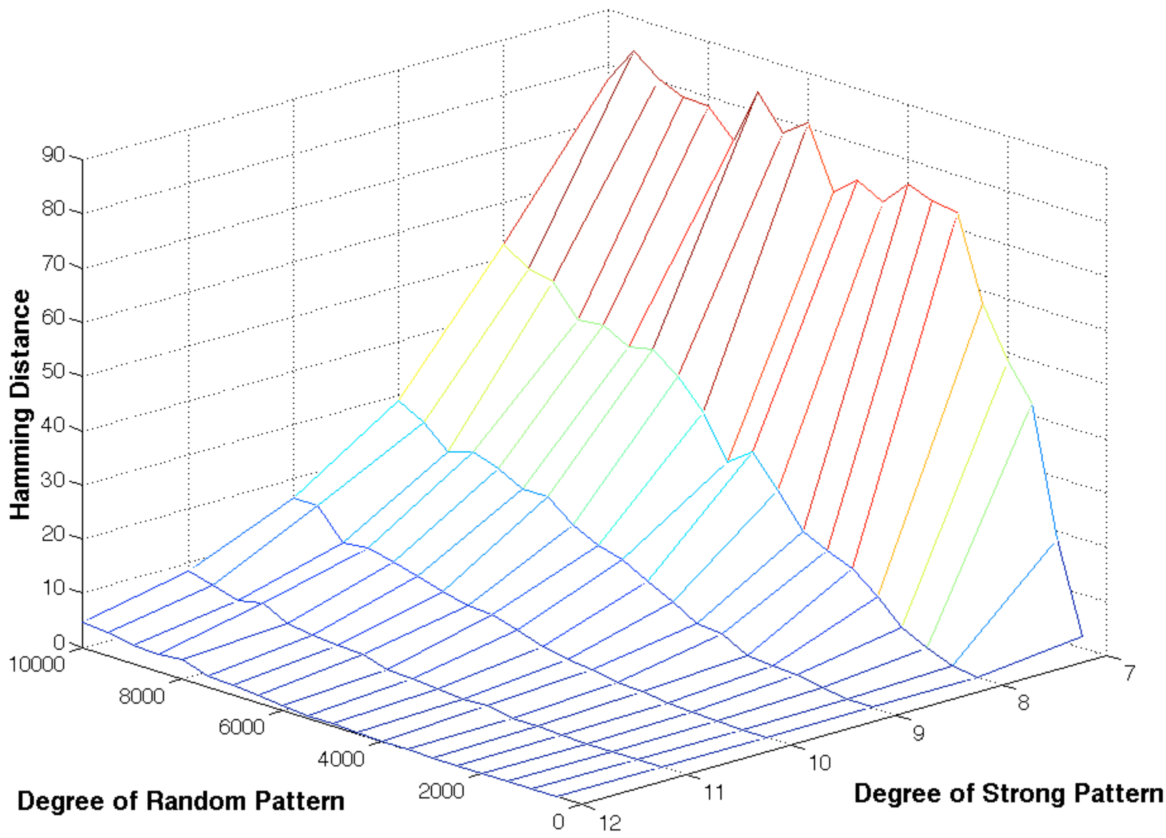


FIGURE 2.5 The relation between the strong pattern stability and the degrees of strong pattern and degrees of random patterns. The stability is measured using average Hamming distance from the recalled patterns and the stored strongest pattern.

Chapter 3

Restricted Boltzmann Machine

3.1 Background Review

- **Boltzmann machine**

The Boltzmann machine is a stochastic network with symmetric connections, $\omega_{ij} = \omega_{ji}$, trained by a general learning rule introduced by Hinton and Sejnowski [Hinton and Sejnowski, 1983, 1986; Ackley, Hinton, and Sejnowski, 1985]. The probability of states of the Boltzmann machine is based on the Boltzmann distribution of statistical mechanics. It can be seen as a Hopfield network with a hidden layer (Figure 3.1), and the hidden layer has no connection to the outside world. The problem is thus to find the appropriate connection between every two units with the learning rule. It is unsupervised learning because the representations the hidden units use from the training patterns is not specified.

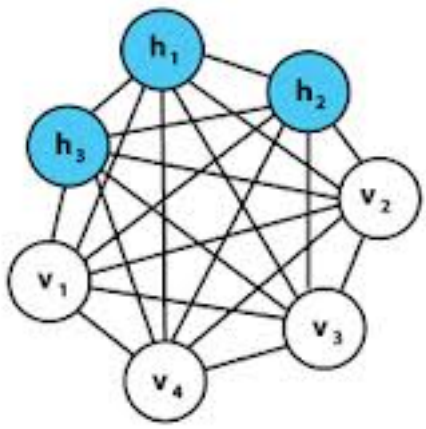


FIGURE 3.1 A Boltzmann machine has visible units, v_i, and hidden units, h_i. This system is fully connected.

- **Restricted Boltzmann Machine**

Restricted Boltzmann machines (RBMs) are Boltzmann machines with no connections between units in the hidden layer or between units in the visible layer (Figure 3.2). Compared to Boltzmann machines, RBMs are less weighted and thus perform much faster. They have been applied to represent various types of data, such as user ratings of movies (Salakhutdinov et al., 2007), labelled or unlabelled images (Hinton et al., 2006), and bags of words that represent documents (Salakhutdinov and Hinton, 2009).

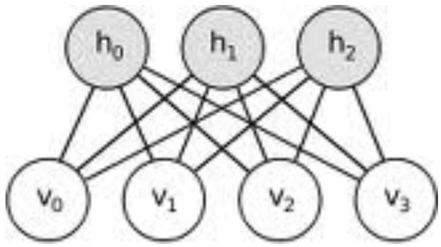


FIGURE 3.2 A Restricted Boltzmann machine has visible units, v_i, and hidden units, h_i. No visible unit is connected to any other visible unit, and no hidden unit is connected to any other hidden unit.

An RBM has symmetric weight connections; thus its configuration states, ***(v, h)***, including both the visible units and hidden units, form an energy function, similar to the Hopfield networks, as follows:

$$E\{(\boldsymbol{v},\boldsymbol{h})\} = -\sum_{i\in visible} a_i v_i - \sum_{j\in hidden} b_j h_j - \sum_{i,j} v_i h_j \omega_{ij} \tag{3.1}$$

where v_i and h_j represent the binary states of the visible units *i* and the hidden units *j*, with their biases a_i and b_j, respectively, and ω_{ij} is the weight between them.

This network uses the Boltzmann-Gibbs distribution from statistical mechanics to assign a probability to each pair of visible and hidden units as follows:

$$P\{(\boldsymbol{v},\boldsymbol{h})\} = \frac{1}{Z} e^{-E\{(\boldsymbol{v},\boldsymbol{h})\}} \tag{3.2}$$

where the partition function, *Z*, is the appropriate normalisation factor as follows:

$$Z = \sum_{S(v_i,h_j)} e^{-E\{(v_i,h_j)\}} \tag{3.3}$$

The probability of a visible unit, v_i, irrespective to any hidden units, $\boldsymbol{h}$, is given as follows:

$$P\{(v_i)\} = \frac{1}{Z}\sum_{\boldsymbol{h}} e^{-E\{(\boldsymbol{v},\boldsymbol{h})\}} \tag{3.4}$$

The network is trained by adjusting the weights and biases to minimise the energy of the desired patterns and to raise the energy of others because Equation (3.3) shows that the pattern with the lowest energy contributes the most to the partition function. The derivative of the log probability of a visible vector with respect to a weight is as follows:

$$\frac{\partial \log p(v)}{\partial \omega_{ij}} = \langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{model} \tag{3.5}$$

where <> denotes expectations over data or models according to their distribution. Application of Equation (3.5) results in a very simple learning rule with the training data as follows:

$$\Delta\omega_{ij} = \epsilon\big(\langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{model}\big), \tag{3.6}$$

where ϵ is the learning rate.

An RBM has no direct connections between visible units or between hidden units; thus, given a visible vector, the binary state of each hidden unit, h_j, is set to 1 with a probability as follows:

$$p(h_i = 1|\boldsymbol{v}) = \sigma(b_j + \textstyle\sum_i v_i \omega_{ij}), \tag{3.7}$$

where $\sigma(x)$ is the logistic sigmoid. Given a hidden vector, an unbiased sample of the binary state of each hidden unit, v_i, is set to 1 with a probability as follows:

$$p\big(v_j = 1\big|\boldsymbol{h}\big) = \sigma(a_i + \textstyle\sum_j h_j \omega_{ij}). \tag{3.8}$$

Using Equation (3.7), an unbiased sample, $v_i h_i$, of $\langle v_i h_i \rangle_{data}$ can be obtained. However, it is more difficult to obtain an unbiased sample of $\langle v_i h_i \rangle_{model}$. It must be done by setting the state of all visible units randomly, and then performing alternating Gibbs sampling for a long time. One round of alternating Gibbs sampling involves updating all of the hidden units using Equation (3.7), and then updating all of the visible units using Equation (3.8).

Hinton proposed a much simpler learning procedure in 2002, which is presented in Algorithm 1. This learning procedure has achieved success in many applications.

ALGORITHM 1: **A learning procedure proposed by Hinton**

1) Initialise all weights
2) **for** each epoch
3) **for** each training example
4) set visible units to a training example
5) update hidden units using Equation (16)
6) calculate $\langle v_i h_j \rangle_{data}$
7) "reconstruct" visible units using Equation (17)
8) do step 4
9) calculate $\langle v_i h_j \rangle_{recon}$
10) $\Delta\omega_{ij} = \epsilon\left(\langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{recon}\right)$
11) update weights
12) **Until** all training examples used
13) **Until** maximum epoch reached

3.2 Training RBMs

A restricted Boltzmann machine is trained to accurately categorize the six emotional patterns and one metaphorical pattern, named "thoughtful" pattern (see image below) when the input is some image with noise.

- **Data**

The data used for this experiment are clusters of the seven images mentioned above. Clusters are generated by a simple probabilistic method, starting with a root image and flipping each bit with some probability p. Examples of each image with noise are displayed in Fig 3.3.

Starting from one of the seven iamges, 50 patterns are generated for each root image flipping with probability $p = 0.1$.

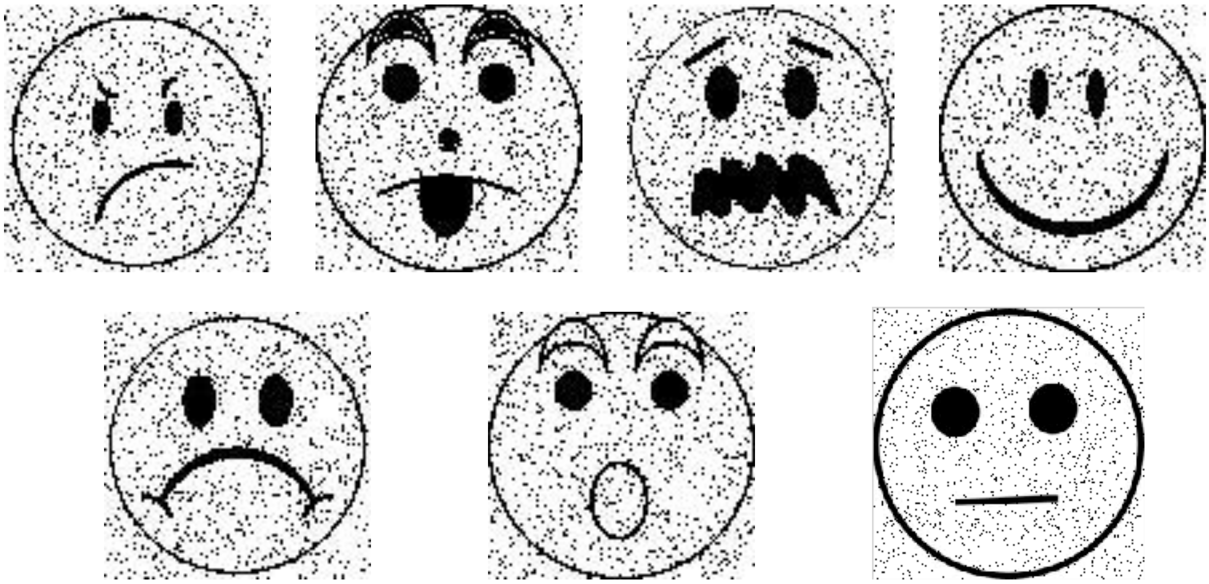


FIGURE 3.3 Anger, disgust, fear, happiness, sadness and surprise smiley faces with noise.

- **Implementation**

The learning procedure is implemented based on Algorithm 1. In addition, some modifications are applied to improve the performance of the system.

- **Mini-batch Training**

To increase training efficiency, the system divides all training examples into 10 mini-batches and then updates the network batch-by-batch, instead of updating the network example-by-example (see line 3 of Algorithm 1). Therefore, the training examples are first ordered randomly then placed into mini-batches of 10 examples each.

- **Initialisation**

Large initial random values for the weights may result in a poor final network, even though the initial learning occurs faster. In this experiment, the weights are initially set to a small value, chosen from a Gaussian distribution with zero mean and a standard deviation of 0.01. All biases are initially set to 0.

- **Size of hidden layer**

The way to determine the number of hidden units in the hidden layer is different from the intuition derived from discriminative machine learning, where the number of output units is equal to the number of bits that is used to specify labels. In the generative learning, however, the model of high-dimension data uses more units to specify the latent properties of a training case than number of bits used to specify a

label.
In this experiment, we varied the number of hidden units from 10 to 30, and observed that if the number were low, the learner would output an identical vector to specify different categories, namely, the learner is not able to distinguish some patterns mentioned above. On the other hand, if the number is too high, the learner will introduce some redundancy, using two or more vectors to describe one pattern.

- **Learning rate**

Keeping the learning rate reasonably low is a good choice since too large learning rate may make weights explode. The drawback of low parameter is it results in a slow learning procedure.

- **Results**

After many attempt, we finally determined the number of hidden units $n_h = 18$, and the learning rate $\lambda = 0.25$. Using this combination of parameters, we observed the trained RBM has the capability of identifying all of the 50 examples (described in the data section) belonging to any cluster by a distinguished vector.

The binary vectors from the trained RBM corresponding to each pattern are shown below:

- Anger → [1 1 1 1 1 1 1 0 1 1 0 0 0 1 1 1 0 1]
- Disgust → [1 0 1 1 1 1 1 0 1 1 0 0 0 1 1 1 1 0]
- Fear → [1 1 1 1 1 1 1 0 1 1 0 0 0 1 0 1 1 0]
- Happiness → [1 0 1 1 1 1 1 1 1 1 0 0 1 0 0 1 1 0]
- Sadness → [1 0 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 0]
- Surprise → [1 0 1 1 1 1 1 0 1 1 0 0 1 1 1 1 1 0]
- Thoughtful → [1 0 1 1 1 1 1 0 1 1 0 0 1 0 0 1 1 0]

These binary vectors are thought of binary values that are decoded to values in decimal place. These decimal values are used for control and comparison purpose in the following section.

- **Remark**

In this experiment, we only cared about the ability of the trained RBM to categorize the seven clusters. We did not pay too much effort on the efficiency of the algorithm, such as tuning the learning rate carefully, considering the momentum, and studying the effect of different types of neurons. Furthermore, it is easy to obtain remarkable performance (100% accuracy) because the dimension of input images is small. More efforts have to be paid if a real face image is used.

Chapter 4

Decision Making Network

In human's brain, multiple decision rules coexist and compete with each other. These rules are either irrationally heuristic or deliberative. The decision rules at different levels of sophistication activate the different regions of the brain.

Levine [2009] presented that it is proved that the heuristic decisions are inherited from emotionally influenced decisions made by other mammals, and the deliberative decisions are involving logic calculation and working memory manipulation. He illustrated a decision-making system in one's brain encompassing a network of needs, a network of decision rules, and the communication between these two subsystems. The network of needs involves physiological as well as psychological needs in different levels. These needs compete with each other, and the state of the needs network moves within these needs due to discontent. The network of decision rules consists four connected area: the amygdala, the orbitofrontal cortex (OFC), anterior cingulate (ACC) and dorsolateral prefrontal cortex (DLPFC), which account for various decision rules on specific tasks. These four regions comprise a three-layer network, in which the vigilance determines the status of activation of each layer. The state of the needs network influences the vigilance so that the winning needs can be dominant to implement the corresponding decision rules.

4.1 The Network of Needs

- **Levine's Network of Needs**

The network of needs exists in the regions including hypothalamus, midbrain and brain stem. Theses regions comprise a competitive-cooperative network is assumed to be an attractor network with an energy function that decreases as the system evolving. These attractors stored in the system compete with each other, and the some more "optimal" attractors, having lower energy, have more chance to win, so that the state of the needs network will converge the them.

The scheme of the needs network developed by Levine (see Fig 4.1) illustrates biologically how the state of the network moves from attractor with higher energy to the one with low energy. That is, as shown in Fig 4.1, states with the smallest values of the energy function V are the most desirable. In the scheme, the world modeler, akin to working memory areas in DLPFC, accounts for calculating the energy function for each possible state of the need subsystem and memorizing the one with the lowest energy. The "creative discontent" module, which is analogous to some part of the amygdala, receives excitation V signal from the needs subsystem and inhibition V signal from DLPFC, and it is activated when the energy of current state of needs module is greater than the energy of the optimal state stored in DLPFC. Such activation will lead amygdala to send random noise to needs module so that the state can be changed to the global optimization with some probability, which is influenced by the amount of the generated noise.

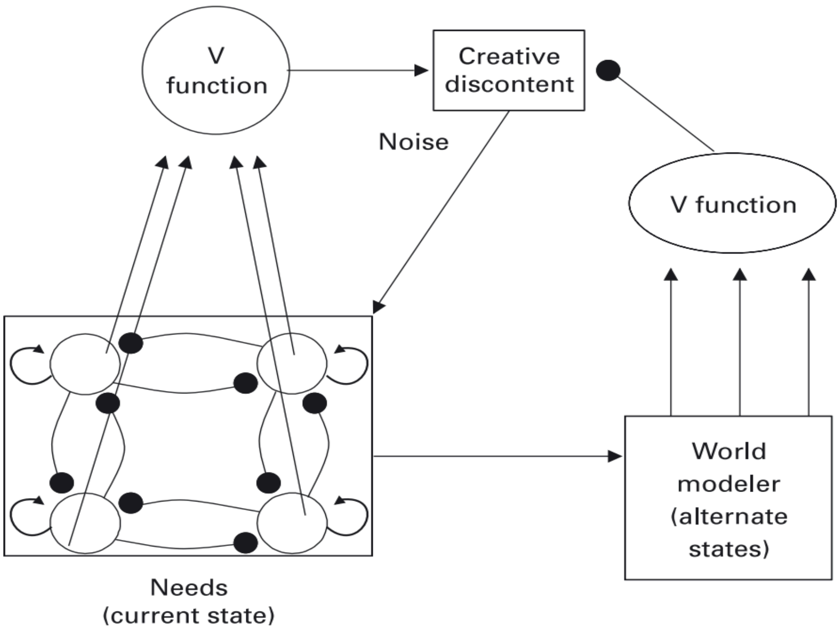


FIGURE 4.1 The schema of a competitive attractor network. This diagram is borrowed from Levine's work (2009). Refer to his paper for detailed description.

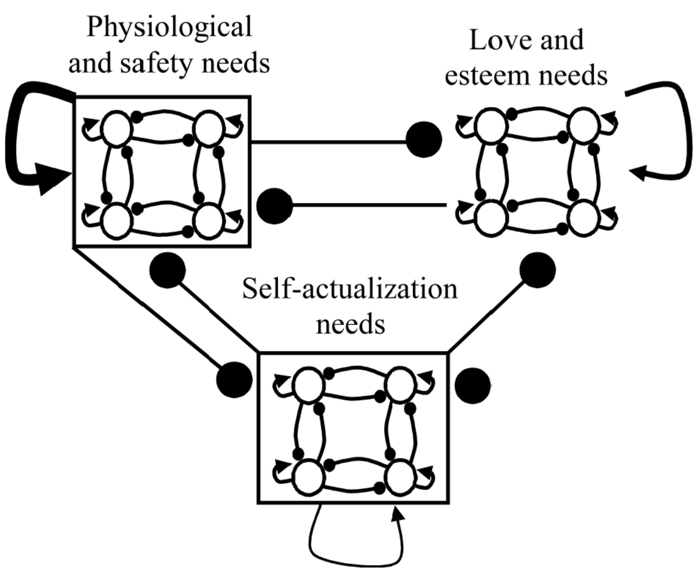


FIGURE 4.2 A neural network rendition of Maslow's hierarchy of needs. This diagram is borrowed from Levine's work (2009). Refer to his paper for detailed description.

- **The Needs Module**

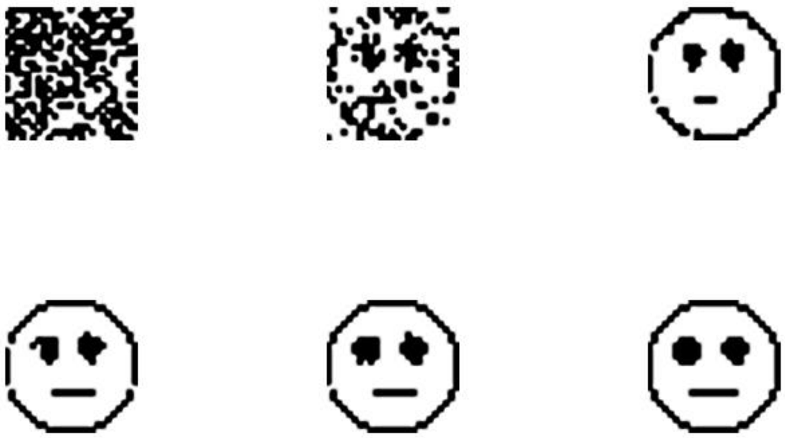
Levine developed the needs module as the interpretation of Maslow's hierarchy of needs (1968). Maslow added the idea of psychological needs, such as love, esteem, and self-fulfillment, to the purely physiological needs. These needs in the needs module inhibit others while send excitatory signal to themselves, and normally the physiological needs have the strongest self-excitation (see Fig 4.2). The degrees of strong attractors in this network depend on the number of needs that these attractors have satisfied.

- **Modelling the Network of Needs with Hopfield Network**

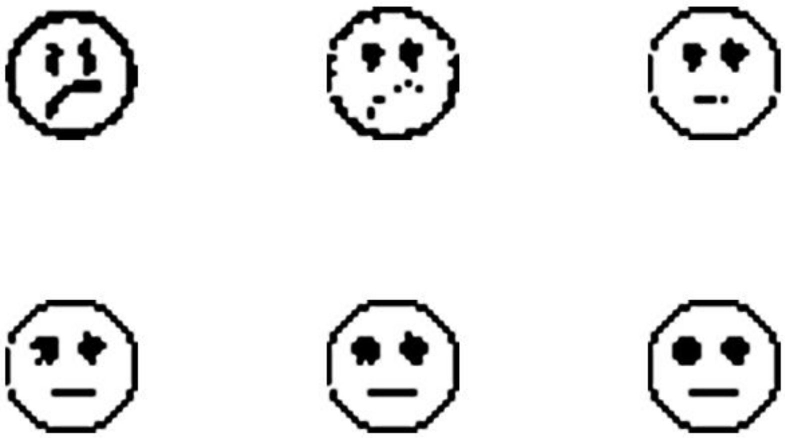
The rendition of Maslow's hierarchy of needs in Fig 4.2 can be modified to a two-layer hierarchy containing need for cognition (Cacioppo & Petty, 1982) and need for cognitive closure (Webster &Kruglanski, 1994). Need for cognition is thought of the motivation to analyze arguments deeply, and individuals with high need for cognition are more likely to make deliberative decisions. Need for cognitive closure is thought of the motivation to make decisions without thinking about the relevant issues, and individuals with high need for cognitive closure are more likely to make heuristic decisions.

Here we employ a standard Hopfield network designed in Section 2 to model Levine's needs diagram. Six basic emotions and one "thoughtful" pattern are stored as strong patterns in the system with various degrees. Let us assume that the emotional patterns are strongly correlated to the need for cognition and the thoughtful pattern is strongly correlated to the need for cognitive closure. That is, increasing the degree of one of the emotional patterns leads to enhancing the satisfaction of need for cognitive closure, and increasing the degree of the "thoughtful" pattern leads to enhancing the satisfaction of need for cognition. As shown in Fig 4.3, the two needs compete with each other in the needs module, and the six emotional patterns compete with others within the need for cognitive closure. In the Hopfield network, the initial configuration of the system will be updated during iterations, and it will eventually converge to the strongest pattern with the biggest probability. Such process that travelling from a state with high energy to a state with the lowest energy is analogous to the process presented in Fig 4.1.

For example, if the "thoughtful" pattern is the dominant attractor stored in the Hopfield network, that is, the "thoughtful" pattern has higher degree than any other strong pattern in the system, the brain model containing this Hopfield network is thought of having high need for cognition. Therefore, as in Levine's model, the energy of "thoughtful" pattern will be stored in the working memory, and then any state of the network with higher energy will be disturbed by random noise so that the needs module can move to the "thoughtful" attractor. Similar to this process, given any initial configuration, the dynamics of the Hopfield network will eventually converge to its strongest attractor. Experiment 4.1 presents one example of the dynamics of the Hopfield network when the input pattern to the system is generated randomly. And Experiment 4.2 shows the dynamics of the system when the input pattern is another strong attractor stored in the network. On the other hand, if one of the six emotional patterns is the dominant attractor, the brain model is thought of having high need for cognitive closure. In this project, the initial configuration of the Hopfield network is designed to be an "anger" type (i.e. the anger emotion being dominant).



EXPERIMENT 4.1 Dynamics of the Hopfield network when the input pattern is random (top left corner). The system states eventually converge to strongest attractor (button right corner).



EXPERIMENT 4.2 Dynamics of the Hopfield network when the input pattern is another strong attractor (top left corner). The system states eventually converge to strongest attractor (button right corner).

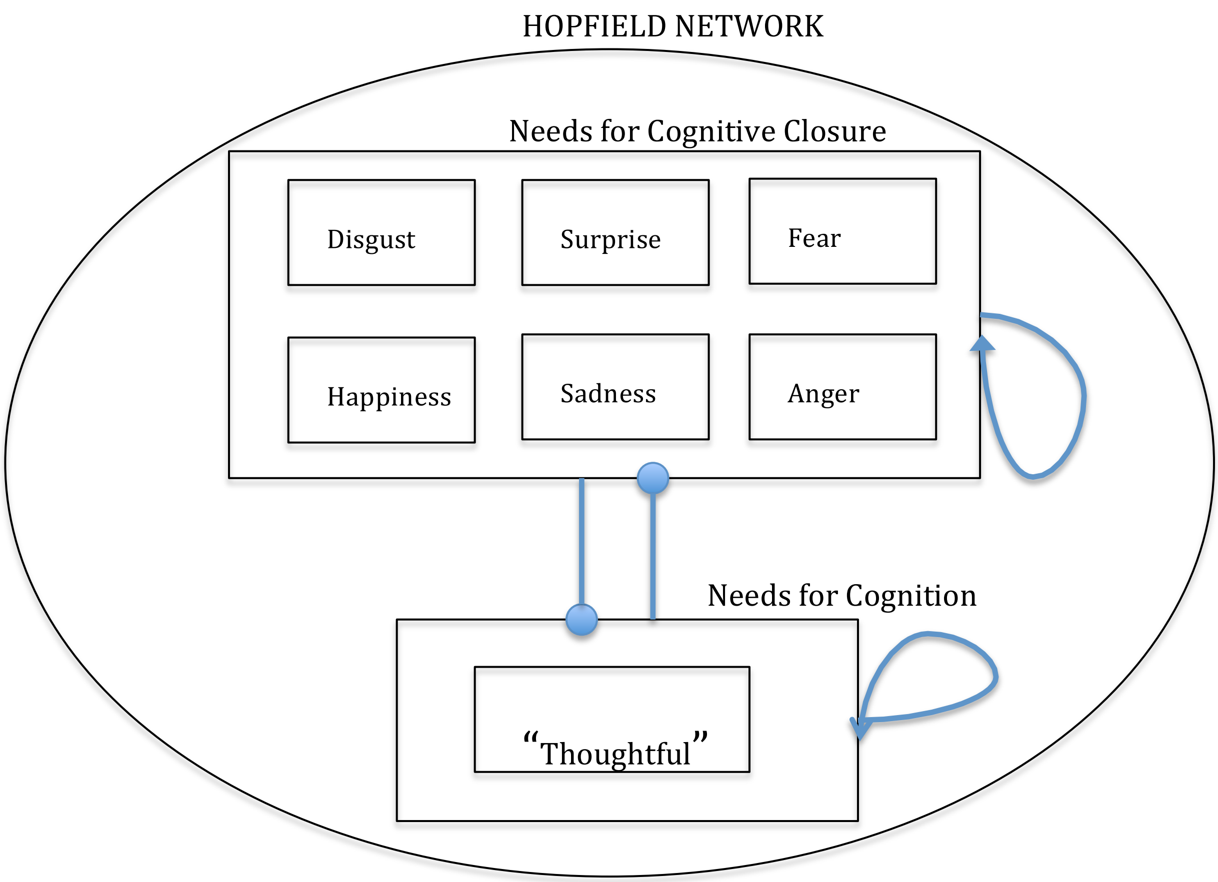


FIGURE 4.3 The Hopfield network containing the needs for cognitive closure and needs for cognition. The needs for cognitive closure are correlated to the six basic emotional patterns, and the needs for cognition are correlated to the "thoughtful" pattern. Similar to the relationship of the needs in Maslow's hierarchy, each of these two kinds of needs receives inhibition signals from another and excitation signals from itself.

4.2 The Network of Decision Rules

- **Levine's Network of Decision Rules**

The network of decision rules accounts for encoding various decision rules. These rules are generated in different regions of the brain. As fMRI measurement showed, the orbitofrontal cortex (OFC) and anterior cingulate (ACC) are activated more strongly than the amygdala when individuals violate the heuristic rules, while the amygdala is activated more strongly when individuals follow the heuristic rules. Furthermore, previous works (DeNeys et al., 2008) showed that there was greater activation of the dorsolateral prefrontal cortex (DLPFC), another region of prefrontal cortex, in individuals who are dealing with sophisticated tasks that need calculation or probabilistic reasoning.

Both of OFC and amygdala are involved in emotional process: OFC integrating sensory and affective information and amygdala dealing with primary emotional experience, but OFC is at higher level than the amygdala. Therefore, the amygdala and OFC are analogous to primary and secondary sensory cortex respectively.

As presented in Levine's paper (2009), Carpenter and Grossberg (1987) have successfully modelled the relationships between primary and secondary sensory cortex by adaptive resonance theory (ART). Fig 4.3 shows a simple form of ART, containing a layer of primary sensory nodes F_1, identified with amygdala, and a layer of secondary nodes F_2, identified with OFC. The letter r in the box is called vigilance. If the similarity of the output patterns from F_1 and F_2 is greater than r, the response of amygdala and OFC is said to be matching. If a matching occurs, enough nodes in F_1 will be excited; otherwise, F_1 activity will be shut off by F_2. The bidirectional connections indicate that these two layers have effect on each other, that is, the OFC collects affective information from both secondary sensory cortex and amygdala, and the emotional reaction of the amygdala is controlled by OFC. It has been shown that the OFC is responsible to adjust the internal emotional signals so that appropriate response outputs is made to adapt to particular environments (Schore, 1998). This OFC-amygdala loops accounts for making both heuristic and deliberative rules. Next, we discuss the DLPFC-OFC loops, which are only required for deliberative rule.

The DLPFC-OFC loops have similar structure to the OFC-amygdala loops. The feedback relationships between OFC and amygdala can be used to describe the relationships between DLPFC and OFC, that is, the DLPFC is at higher level of abstraction than OFC, and has ability to control the response of OFC (Levine, 2009, presented the work of Dias, Robbins, &Roberts, 1996, as evidence). fMRI measurements (Van Veen & Carter, 2006) also showed that the ACC can play the role of the orienting subsystem (the box with r) in Fig 4.3, because ACC accounts for conflict or potential error detection.

Now, these regions, amygdala, OFC, ACC, and DLPFC can be integrated to build a three-layer hierarchical network that encodes both heuristic and deliberative rules (see Fig 4.4): OFC-amygdala loops generating simple heuristics and DLPFC-OFC loops dealing with complex tasks. The vigilance with high value makes individuals be aware of the conflict between heuristics and logical truth more easily.

- **Modeling the Network of Decision Rules with Restricted Boltzmann Machine**

Three identical restricted Boltzmann machines are introduced to simulate the behaviours of amygdala, OFC and DLPFC. This restricted Boltzmann machine, designed and trained in Section 3, is able to categorize seven different patterns (including six emotional patterns and one "thoughtful" pattern) and represent them using seven distinct 17-unit vectors. These three layers respond to input signals in different ways (discussed when combined with the network of needs in next subsection). And the outputs from them will be compared with respect to value r in ACC. We say that mismatch occurs if the hamming distance from amygdala output to OFC output (or from OFC output to DLPFC output) is smaller than r. For individuals with "thoughtful" pattern as the strongest attractor (high need for cognition), the vigilance (value r) is always very high, while for individuals with one of the emotional patterns as the strongest attractor (high need for cognitive closure), the vigilance is low. In other words mismatch is easier to happen in "thoughtful" person, while a person with low vigilance is always following the heuristic signals.

As far as the control from OFC to amygdala (or from DLPFC to OFC) is concerned, we designed two versions of this brain system in this project: one regarding the OFC as a controller to adjust the outputs from amygdala, and another one disregarding the control activity of OFC. However, before we can design such controller to simulate the control activity of OFC, we have to firstly know the dynamics of the controlled system – the 17-neuron population in amygdala.

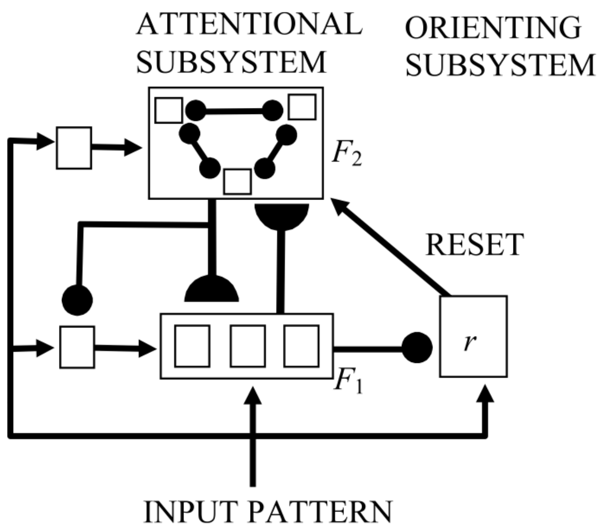


FIGURE 4.3 Architecture of adaptive resonance theory. This diagram is borrowed from Levine's work (2009). Refer to his paper for detailed description.

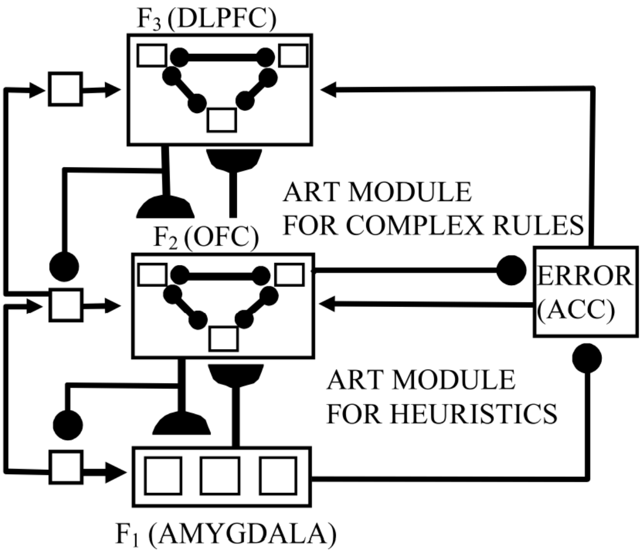


FIGURE 4.4 Network that encodes both heuristic and deliberative rules. This diagram is borrowed from Levine's work (2009). Refer to his paper for detailed description.

- **Modeling and Feedback Controlling**

To model dynamics of the 17-neuron population in amygdala, we take these neurons as neural system existing in a neurobiological system, although the neurons used in this project are binary units. Dynamics of real neural population are very complex and highly nonlinear, but it is still a good approximation that considering the population system as a linear system, because nonlinear modeling cannot improve the estimate of behavior of real neural system (Rieke et al., 1997), and the techniques for nonlinear modeling related to neurobiological mechanisms are not well recognized. As mentioned in Eliasmith's book, Neural Engineering (Page 112), such linear approximation is still biologically plausible because it is relevant to the production of postsynaptic currents (PSCs) in the postsynaptic cell. For simplicity, a PSC model can be written as

$$h_{psc}(t) = e^{-\frac{t}{\tau_{psc}}}, \tag{4.1}$$

where τ_{psc} is the synaptic time constant. And we can apply this model to the population system, so that the transfer function of the 17-neuron population is

$$h(t) = \frac{1}{\tau} e^{-t/\tau}, \tag{4.2}$$

where τ is the synaptic time constant. For convenience of control, we can rewrite Equation (4.2) in frequency domain using Laplace transformation:

$$\mathcal{L}(f(t)) = \int_0^\infty e^{-st} f(t)dt$$

$$= f(s).$$

So, the Laplace transform, $h(s)$, of $h(t)$ as given by (4.2) is

$$h(s) = \frac{1}{1+s\tau}. \qquad (4.3)$$

Experiment 4.3 presents different system response to a unit input $u(t) = 1$, with various time constant τ. The system output $y(t)$ in terms of time t is the convolution of the system dynamics $h(t)$ and the input signal $u(t)$:

$$y(t) = h(t) * u(t)$$

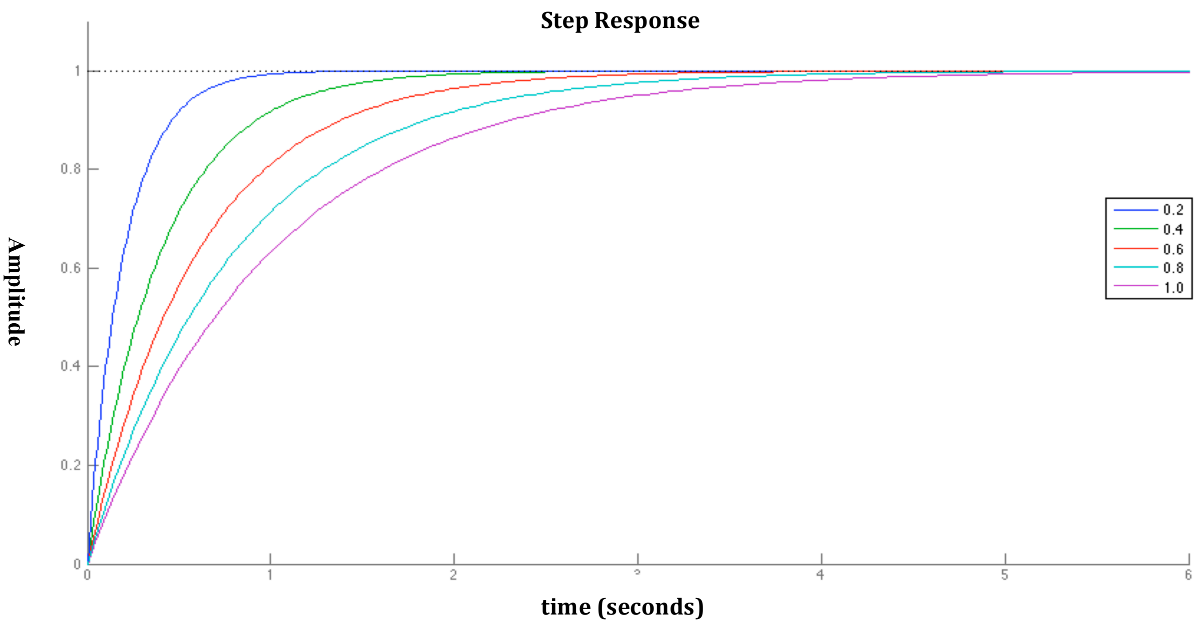
$$= \int_{-\infty}^{+\infty} u(\tau)h(t-\tau)d\tau. \qquad (4.4)$$

The Laplace form of $y(t)$ is

$$y(s) = h(s)u(s), \qquad (4.5)$$

where $u(s) = \frac{1}{s}$ is the Laplace form of $u(t)$.

In Experiment 4.3, all of the output signals converge to 1, and the system with smaller time constant



EXPERIMENT 4.3 Dynamics of $h(t)$ in response to a unit input for several values of time constant.

Once a model of the system is identified, we can design an engineering feedback controller to control the system to output appropriate response. Here, we used a proportional gain controller to perform the control from the OFC to amygdala.

A proportional gain controller (P controller) is the simplest feedback controller used in industrial control system. Fig 4.5 shows a schema of a feedback control loop with P controller. The controller $c(t)$ outputs a control signal $u(t)$ to the controlled system $g(t)$ according to the difference (error $e(t)$) between the measured system output $y(t)$ and the desired reference $r(t)$. This controller attempts to minimize the error and increase the speed of system response.

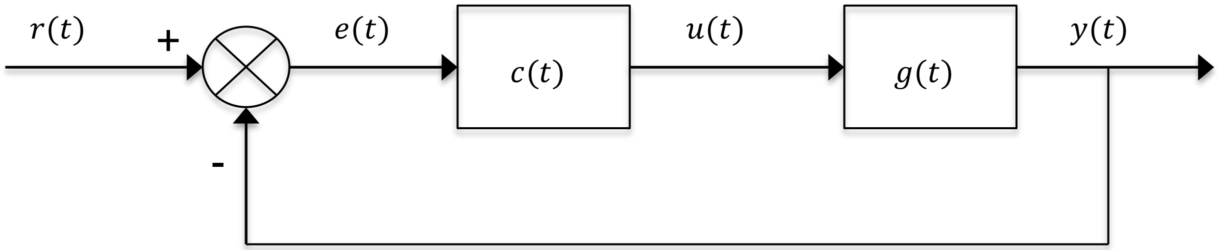


FIGURE 4.5 A block diagram of a P controller in a feedback loop. The dynamics of $c(t)$ and $y(t)$ are analogue to the control behavior of the OFC and the amygdala respectively.

The P controller has a mathematical expression, referring to Fig 4.5:

$$c(t) = K_p \tag{4.6}$$

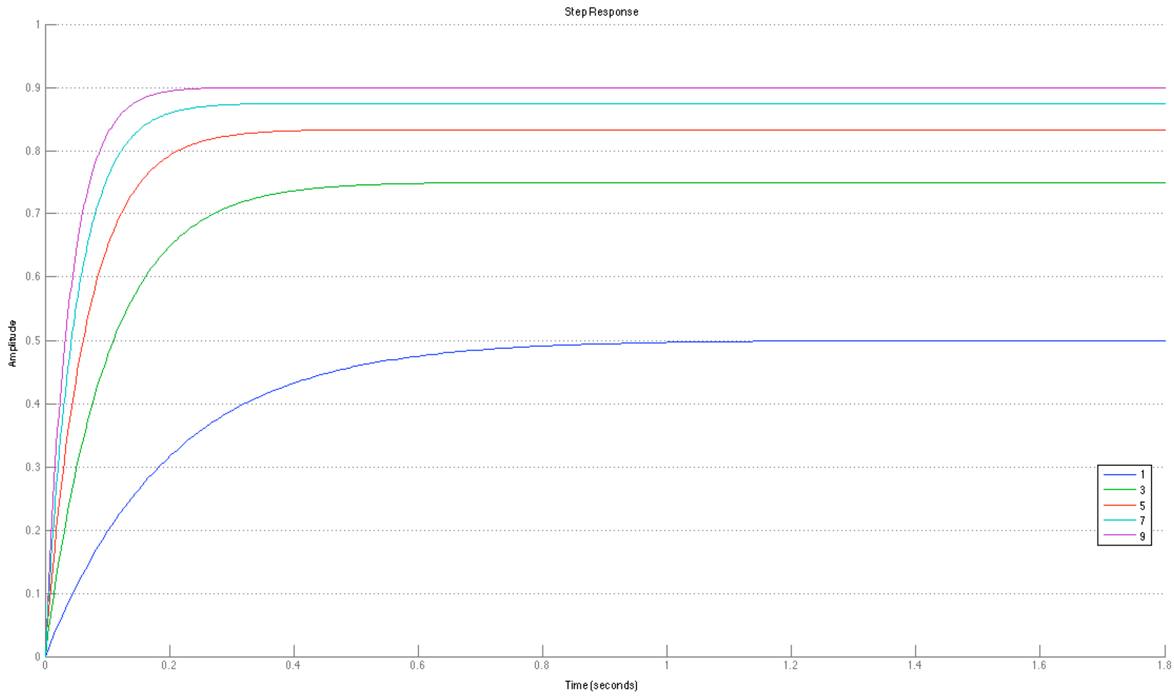
where K_p is called proportional gain, and the dynamics of the controller output can be expressed as

$$u(t) = K_p e(t), \tag{4.7}$$

and

$$e(t) = r(t) - y(t). \tag{4.8}$$

Experiment 4.4 shows how the parameter K_p has effect on the dynamics of the feedback system. Giving the reference $r(t) = 1$ and $\tau = 0.4$, we tune the value of K_p from 1 to 9 and observe the response of $g(t)$ in terms of time t. As we increase the value of K_p, the system responds faster, and the values that the system converges to are closer to the reference. However, a large K_p may make the system unstable because a large change in control signals to the system will occur if the error signal is large.



EXPERIMENT 4.4 Dynamics of a feedback control system in response to a unit reference for several values of K_p.

4.3 Connection of Needs network and Decision network

Levine (2009) claimed that different needs in the module of Fig 4.2 could results in different vigilance in the network of Fig 4.4. Here, we assigned a low value of vigilance to the need for cognitive closure, and a high value to the need for cognition. Such that a model in which the thoughtful pattern is dominant has more ability to detect the mismatch between the OFC and amygdala (or the DLPFC and OFC) in Fig 4.4. In addition, if the vigilance is low, the OFC-amygdala loop is more activated than the DLPFC-OFC loop, and then the person will generate heuristic decisions; otherwise, the DLPFC-OFC loop is dominant in the network of decision rule, and the person will generate deliberative rules.

In my DECIDER model, the input pattern will firstly stimulate the Hopfield network. The dynamics of the network will eventually bring its state to an attractor with the lowest energy.

If the recalled pattern is "thoughtful" pattern, then the Hopfield network will send high-level vigilance (a constant r) to the error detector (ACC), and the DLPFC-OFC loop is chosen to generate complex decision rules. The RBM in the DLPFC receives a "thoughtful" signal from the Hopfield network, and categorizes it into a 17-unit vector. The RBM in the OFC accounts for categorizing the input pattern into another 17-unit vector. If the similarity (measured by Hamming distance) of these two generated vectors is smaller than the vigilance, then we say that a mismatch occurs and the DECIDER is supposed to generate a deliberative rule. Because of the high vigilance, the DLPFC-OFC loop has no chance to make heuristic rules. It is noticeable that there is no control from DLPFC to OPC in my model.

On the other hand, if the recalled pattern is one of the six emotional patterns, the Hopfield network will send low-level vigilance to ACC, and the OFC-amygdala loop is chosen to account for making decisions. Similar to the loop above, the RBM categorizes an emotional pattern from the Hopfield network, and OFC categorizes the input pattern. The Hamming distance between the two output vectors are compared to the low-level vigilance. A heuristic decision is generated if the similarity is higher than the vigilance; otherwise, a deliberative decision is made. Furthermore, the control from the OFC to amygdala has effect on the output of the RBM in amygdala. A well-trained OFC (i.e. the proportional gain K_p is tuned to be appropriate) can adjust the amygdala output to some value close to the output of the OFC.

- **Experiments**

These experiments were carried out to study how the decisions were made in my model in two different situations: in the first experiment the dominant attractor in Hopfield network was "thoughtful" pattern, which satisfied the needs for cognition; in the second experiment the dominant attractor was a emotional pattern, which satisfied the needs for cognitive closure. We observed the performance of the three identical RMBs and how their outputs were compared so that decisions could be made.

1. A "thoughtful" pattern recalled

The Hopfield network was trained with (pattern name: degree):

- "Thoughtful": 6
- Anger: 4
- Happiness: 3
- Sadness: 2
- Disgust: 1
- Fear: 1
- Surprise: 1.

The "anger" smiley image is chosen to input to the system. And then the network recalls the "thoughtful" pattern. Therefore, the value of vigilance is chosen to be 16 (a high-level vigilance), and in the DLPFC-OFC loop, the RBM in the DLPFC categorises the "thoughtful" pattern into a 17-unit vector:

1	0	1	1	1	1	1	1	0	0	1	0	1	0	1	1	1

and the OFC categorises the anger pattern into

1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	1

The Hamming distance between these two vector is 5, so the similarity is $17 - 5 = 12$, which is smaller than the vigilance. Therefore, the DECIDER is supposed to make deliberative decisions.

2. An emotional pattern recalled

The Hopfield network was trained with (pattern name: degree):

- "Thoughtful": 4
- Anger: 3
- Happiness: 6
- Sadness: 2
- Disgust: 1
- Fear: 1
- Surprise: 1.

The "anger" smiley image is chosen to input to the system. And then the network recalls the happiness pattern. Therefore, the value of vigilance is chosen to be 2 (a low-level vigilance), and in the OFC-amygdala loop, the RBM in the amygdala categorises the happiness pattern into a 17-unit vector:

1	0	1	1	1	1	1	1	0	0	1	0	1	0	1	1	1

and the OFC categorises the anger pattern into

1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	1

First, let us disregard the effect of the control from the OFC to amygdala. The Hamming distance between these two vector is 5, so the similarity is $17 - 5 = 12$, which is higher than the vigilance. Therefore, the DECIDER is supposed to make heuristic decisions. Even though this two patterns are different.

Then, let us take account of the effect of the control. The dynamics of the amygdala is designed as

$$h(s) = \frac{1}{1+0.6\tau},$$

and the OFC is thought of an well-trained P controller, therefore

$$K_p = 9.$$

The 17-unit vector is thought of being in a form of Gray code, which is a binary numerical system where two successive values differ in only one bit. So, these two vectors are transferred to decimal place according to the transformation mechanism of Gray code.

In the control loop, the decimal value of the OFC vector is treated as the reference signal. As shown in Experiment 4.4, the output of amygdala eventually converges to the output of the OFC. After controlling, the output of the amygdala in decimal place is encoded back to Gray code vector. By measuring the Hamming distance, we find that the similarity of these two outputs is increased.

Chapter 5

Reinforcement Learning

5.1 Psychotherapy

Psychotherapy is a form of therapeutic interaction where a trained therapist is involved to talk to a patient, client, patient, family, couple, or group who are suffering from emotional problems. Particularly Psychotherapy is used to treat depression and anxiety disorders, which are influenced by early attachment patterns. As Schore (2003, p. 69) stated, the developmentally based, affectively focused psychotherapy can alter early attachment patterns, especially from negative patterns to positive ones. Such altering process can be well understood referring to previous findings that the orbitofrontal system accounts for "emotion-relative learning" (Rolls, Hornak, Wade, & McGrath, 1994) and that it can be modified in the later periods of life (Barbas, 1995). Furthermore, the study by Hariri, Bookheimer, and Mazziotta (2000) proved that the most modern psychotherapeutic methods support the modulating process from high-level regions of the right prefrontal cortex, such as orbitofrontal cortex, to the most basic level in the brain.

Cognitive behavioral therapy (CBT) is a form of psychotherapy that helps patients manage their problems by changing the way they think and behave, for example, breaking problems down into smaller parts, stop negative cycles due to the interconnections of feelings, thoughts, physical sensation and actions. CBT is different from other talking treatments, such as psychotherapy, because of its focus on current problems rather that past issues.

There are several assumptions made by traditional CBT about patients (Young, 2003, p. 3-5). Firstly, it assumes that patients are willing to follow the treatment protocol rather than resisting it due to complicated therapy. Secondly, it assumes that patients can make use of their cognitions and emotions and report them to the therapist. It is difficult because patients in early stage of the therapy lose the ability to do so because of their engagement in cognitive and affective avoidance. Thirdly, patients are presumed to change their negative cognitions and behaviours after a series of practices. However, some patients are just thought of being hopeless to change because they are lack of psychological flexibility. Fourthly, it assumes that there is a collaborative relationship between the patient and the therapist. Finally, the patient is supposed to have easy-to-address targets. As a conclusion, the traditional CBT has trouble to succeed due to Early Maladaptive Schemas (Young, 2003, p. 43-44), which make it difficult to meet the assumptions above.

5.2 Q-learning

Q-learning algorithm (Watkins, 1989) is a well-known model-free reinforcement learning technique. For any given finite Markov decision process (MDP), the algorithm has the capability of helping an agent find the optimal action sequence, without building maps of the domains.

The objective of using Q-learning algorithm is to find Q value for each combination of actions and states. The Q value is an estimated of the expected reward the agent will receive if it chose some action at some state. This estimation will converge to the true Q function if every state-action pair can be visited infinitely. To perform Q-learning algorithm, the optimal policy given a state is defined firstly: exploration strategy, immediate rewards or penalty associated with different chosen actions, learning rate, discount factor and initial conditions.

- **Exploration Strategy**

The exploration strategy refers to how the agent chooses an action for next state. A well-designed exploration strategy can prevent the solution getting stuck in the local minimum.

- **Immediate rewards or penalty**

The immediate rewards influence the re-evaluation of the quality of each state-action combination. For each state, this is the key to change the dominant action, for example, if the initial Q value for the action "smile", associated with some state, is very low, due to emotional disorder, we can increase the Q value by giving rewards to "smile" and giving penalty to "be anger".

- **Learning rate**

The learning rate determines how much old information will be neglected if new information is received. The learning rate varies from 0 to 1, and "1" means the learner considers the new information only and ignores all past information, while "0" means the learner does not learn anything.

- **Discount factor**

The discount factor determines to what extent the learner will consider the future rewards. The learner is thought of being "myopic" if the discount factor is "0", meaning it only considers current rewards, or it considers long-term rewards if the factor is close to "1". The Q function will diverge if the discount factor is greater than 1.

- **Initial conditions**

The initial conditions of Q function have to be defined before the learning starts otherwise such iterative algorithm does not work. The initial value can be designed in various ways (Shteingart, 2013 & Optimistic Initial Values).

These defined policies are applied to update the Q values according to the conventional update rule:

$Q(s, a_p) \leftarrow Q(s, a_p) + \lambda[R_{a_p}(s, s') + \gamma]$

where

$Q(s, a_p)$: Q value of s-a_p combination
$R_{a_p}(s, s')$: reward observed after perform a_p in state s
s: current state
a_p: chosen action in state s
s': observed state following a_p
λ: learning rate ($0 < \lambda \leq 1$)

and

$$\gamma = \delta \max_{a'_p} Q(s', a'_p) - Q(s, a_p)$$

where δ ($0 \leq \delta \leq 1$) is the discount factor.

5.3 Design of Q-learning

- **Aims and Objectives**

The main purpose of this work is to apply the Q-learning algorithm to mimic the psychotherapy process. Two experiments are carried out to study that how the discount factor is related to the therapeutic effect, and how the initial configuration of the attractor-competitive network influences the therapeutic effect.

- **Outline of Implementation**

As mentioned in previous section, the CBT is capable of changing attachment types, and as a kind of psychotherapy, it is also able to enhance to control from the orbitofrontal cortex to amygdala. In this project, such attachment types are modeled on a metaphorical level using the six emotional images and the "thoughtful" image. These images are stored in the Hopfield network as strong patterns with different degree k. The control behavior from OFC to amygdala is achieved implementing feedback controller with P controller. The parameter k_p is analogous to the control strength. Finally, in this project, a Q-learning algorithm is designed to demonstrate that, during the period of receiving psychotherapy, how the modulating process from orbitofrontal cortex to amygdala is enhanced and how the patterns in the attractor-competitive network are altered between each other. That is, the degrees of patterns in Hopfield network, and the parameter k_p are changing during the operation of Q-learning algorithm.

- **Hopfield network**

The Hopfield network is trained with eight kinds of patterns (anger, happiness, sadness, disgust, fear, surprise, “thoughtful” and random pattern). The starting degrees of these patterns are denoted by a, b, c, d, e, f, g, h respectively. Four patterns (anger, happiness, sadness and “thoughtful”) are considered to change degree during the Q-learning process.

- **Feedback control loop**

The initial value of k_p is set to 0.5. This value will increase after successful treatment.

- **Q-learning**

Based on the Q-learning algorithm designed by Cittern & Edalat(2013), we used and ordinal vector with four elements ($[t\ u\ v\ w]$) to represent the states. Each element of such ordinal vector allocated to one of the four chosen patterns (see table below) and the value of each element is the rank of the degree of the corresponding pattern (see Table 5.1 for an example). The state therefore represents the relative values of pattern degrees: number 1 is assigned to the pattern with the highest degree and the number 4 is assigned to the pattern with the lowest degree. There are 4! states for the permutations of all strict ordinal vector (i.e. for the case where $t \neq u \neq v \neq w$), and 4!/2! states for the non-strict ordinal vector with two equal elements ($t \neq u \neq v = w$), and 4!/3! states for the non-strict ordinal vector with three equal elements ($t \neq u = v = w$), and 4!/4! states for the case where all the elements are equal ($t = u = v = w$), giving a state space S size of 41.

Pattern name	Anger	Happiness	Sadness	Thoughtful
Degree	10	14	3	5
Ordinal vector	2	1	4	3

TABLE 5.1 An example showing how the pattern degrees and the ordinal vector are related. In this case the degree of happiness pattern is the highest among these four patterns, so its value in the ordinal vector is 1.

For each state $s \in S$, there is a corresponding finite set of valid actions A to choose. For simplicity it is assumed that all states share the same action set A = {BeAngery, BeHappy, BeSad, BeThoughtful}, resulting in 41*4 state-action combinations.

A probabilistic ‘Boltzmann’ action selection rule [Kaelbling et al., 1996] is introduced to prevent the Q values getting stuck in a local minimum. This rule allows state-action combination with low Q value to be chosen as much of the state space as possible. According to the rule, the probability that action a_{pi} is selected is:

$$P(a_{pi}|s) = \frac{k^{Q(s,a_{pi})}}{\sum_i k^{Q(s,a_{pi})}}$$

where k is an exploration parameter, increasing which results in reducing the probability of selecting those actions with low Q value. Therefore, the action selection rule with high exploration parameter can be thought of deterministic selection rule (i.e. always choosing the action associated with the highest Q value).

The discount factor varies from 0.1 to 0.9, and for simplicity the learning rate is set to 1. The purpose of CBT is to learn to deal with current problems in an appropriate way, so choosing the action "BeThoughtful" is supposed to receive the highest immediate rewards. As a result, we define the reward for each action as following:

Pattern name	Anger	Happiness	Sadness	Thoughtful
Reward	0	0.5	0.2	1

The initial value of each state-action combination is equal to the value of the corresponding element in the state vector. For example, if some state s' is [1, 4, 2, 3], then the Q value of s'-BeAngery is 1, and of s'-BeHappy is 4.

In conclusion, the Q-learning algorithm is working as shown in Algorithm 5.1. It is notable that the algorithm is terminated once the "thoughtful" pattern becomes the dominant pattern in the Hopfield. Otherwise, it will continue running until meet the maximum iteration.

5.4 Experiments

- **Experiment 1**

In this experiment we studied the how the discount factor is related to the numbers of iterations required to change from one of the emotional pattern to "thoughtful" pattern. Also, the relation between the exploration rate k in the probabilistic 'Boltzmann' action selection rule and the required rounds of iteration is interested.

The experiment goes as follows:

- The dominant pattern is initially set to anger pattern
- The exploration rate k varies from 1: 4
- Discount factor varies from 0.1 to 0.9

- An individual experiment is running up to 10,000 rounds
- The number of rounds required to meet "thoughtful" pattern is recorded
- Each individual experiment is repeated for 20 times

Initially the degrees of the eight patterns are set as follows

- Anger: 8
- Happiness: 5
- Sadness: 3
- Disgust: 1
- Fear: 1
- Surprise: 1
- "Thoughtful": 2
- Random: 500

Eventually, we obtain the values from 200 times of each individual experiment. And we take the mean values so that we can obtain a more accurate estimate for the Q-learning behavior. The results are displayed in Fig 5.1

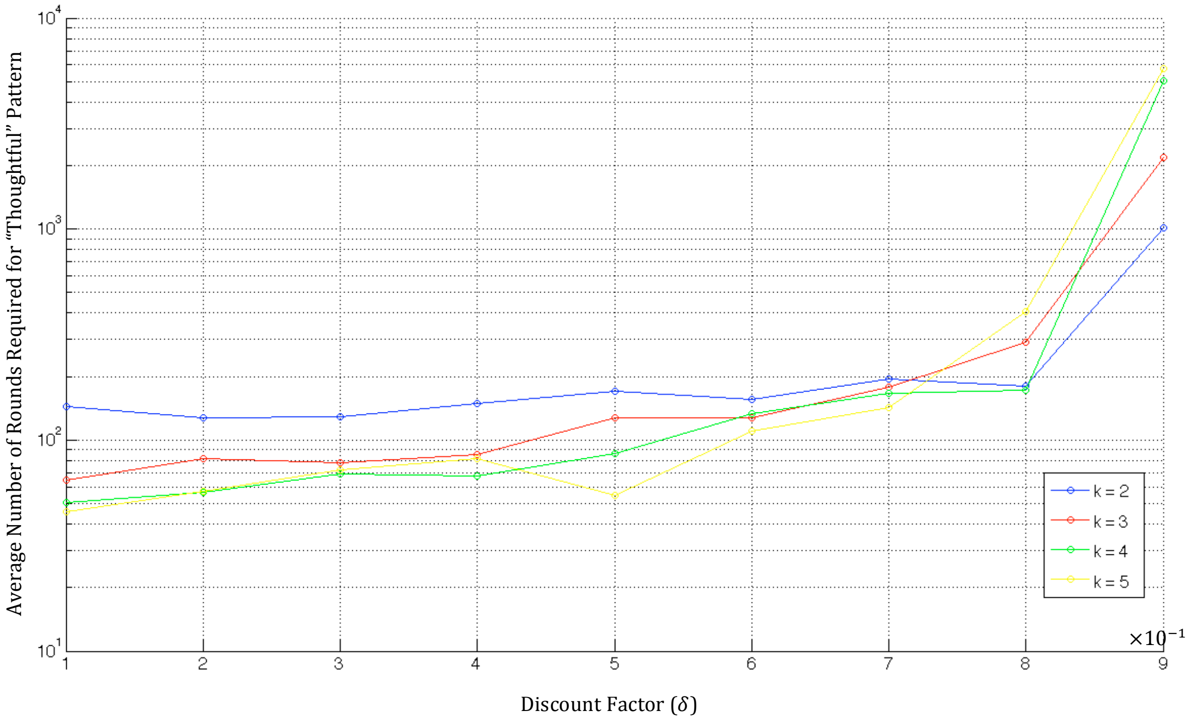


FIGURE 5.1 The average number of rounds required for a "thoughtful" pattern when a patient received CBT treatment for different discount factor (δ). Each line in the plot corresponds to a particular exploration rate (k) in the action selection rule. The y-axis is presented in log scale because the values for $\delta = 0.9$ are unexpectedly high.

Comments

As the exploration rate k is increased, the average number of rounds required for obtaining a successful CBT treatment (i.e. "thoughtful" pattern becomes dominant in the Hopfield network) decreases. The patient with high value of k is thought of a patient who is willing to comply with treatment protocol, which is a major assumption of traditional CBT. It is also observed that, in general, the patient with higher discount factor (i.e. the patient strives for long-term rewards) achieves a successful treatment slower than those who prefer immediate rewards.

We also observed that the number of rounds required for "thoughtful" pattern being dominant is unexpected high when discount factor $\delta > 0.9$ (i.e. the patient who prefer future rewards). The patient will receive rewards every time a reinforced action is chosen, even though there is no state transition occurring, that is, $R_{a_p}(s, s') > 0$ for $s = s'$. Therefor, Q values for $Q(s, a)$ are updated even if no state transition has occurred. For the situation that a patient has high discount factor $\delta > 0.9$, these non-reinforced actions will receive large Q values when the patient takes some action a in state s that do not lead to any transitions. This results in more state-action exploration.

- Experiment 2

In this experiment we studied the how the initial degrees of the stored patterns in the Hopfield network are related to the numbers of iterations required to change from one of the emotional pattern to "thoughtful" pattern.

The experiment goes as follows:

- The dominant pattern is initially set to anger pattern
- The initial degree of the anger pattern varies from 10 to 20
- The initial degrees of other patterns are as constant
- Discount factor $\delta = 0.2$
- Exploration rate $k = 2$
- An individual experiment is running up to 10,000 rounds
- The number of rounds required to meet "thoughtful" pattern is recorded
- Each individual experiment is repeated for 20 times

Initially the degrees of the eight patterns are set as follows

- Anger: 10 to 20
- Happiness: 5

- Sadness: 3
- Disgust: 1
- Fear: 1
- Surprise: 1
- "Thoughtful": 2
- Random: 500

Eventually, we obtain the values from 200 times of each individual experiment. And we take the mean values so that we can obtain a more accurate estimate for the Q-learning behavior. The results are displayed in Fig 5.2

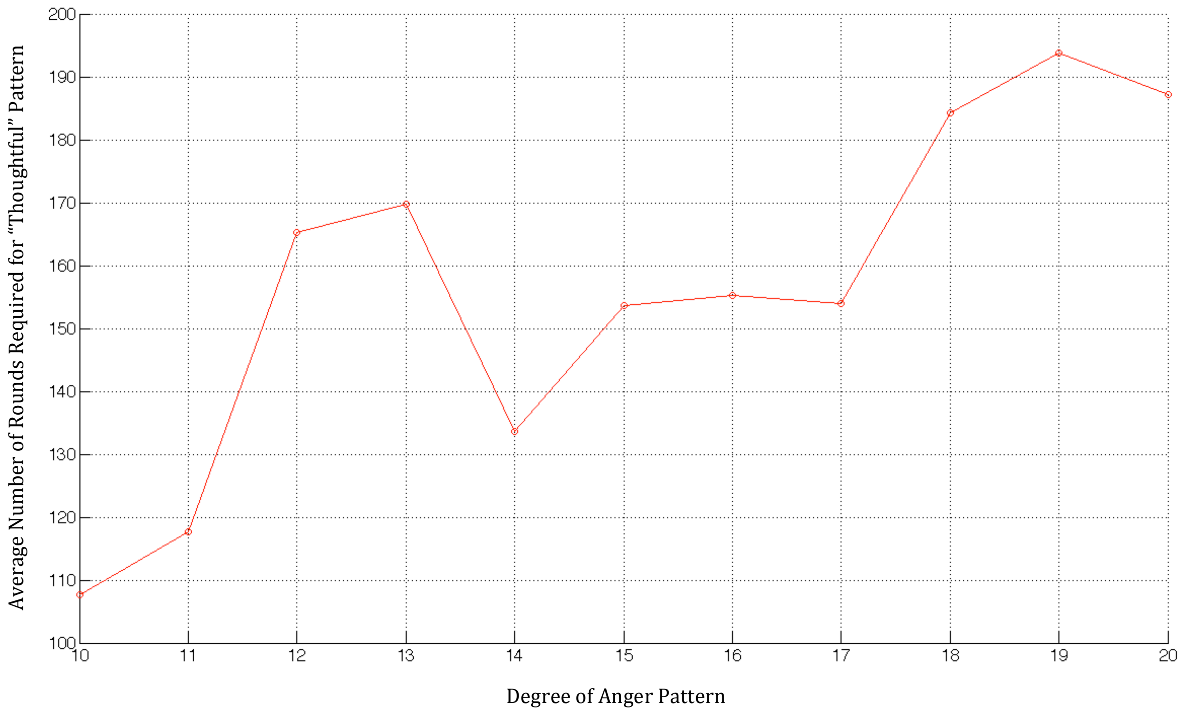


FIGURE 5.2 The average number of rounds required for a "thoughtful" pattern when a patient received CBT treatment for initial degree of anger pattern.

Comments

Expect for the anger pattern, the degrees of any other patterns are viewed as constants. In general, as the degree of anger pattern is increased, the average number of rounds required for obtaining a successful CBT treatment (i.e. "thoughtful" pattern becomes dominant in the Hopfield network) increases. The patient with high degree of anger pattern is thought of a patient who has difficulty to comply with treatment protocol, which is a major assumption of traditional CBT.

Chapter 6

Bring it All Together

Integrating the components discussed above, we can now obtain a super-model containing a super-network DECIDER, which simulates how emotional connections influences decision rules, and reinforcement learning, which mimics the effect of CBT treatment. Fig 6.2 illustrates the structure of DECIDER and how subsystems inside communicate with each other. Fig 6.1 illustrates how DECIDERE copes with the input pattern from out side world and decisions as outputs.

The Hopfield network contains two types of needs: needs for cognition and needs for cognitive closure. Each type of needs inhibits another one and excites itself. Needs for cognitive closure are associated with six basic emotional patterns: anger, happiness, disgust, sadness, surprise and fear. Needs for cognition are associated with a metaphorical pattern: “thoughtful”. These patterns are stored in the Hopfield network as strong attractors, recalled once the network receives stimulus. If the recalled attractor is one of the six emotional patterns, it will be treated as an input signal to amygdala; otherwise, it will be viewed as an input signal to DLPFC.

Three identical well-trained restricted Boltzmann machines (RBMs) represent three regions accounting for various decision rules in one’s brain: dorsolateral prefrontal cortex (DLPFC), orbitofrontal cortex (OFC) and amygdala. This trained RBM has the capability of categorizing seven patterns (six emotional patterns plus “thoughtful” pattern). It receives binary images with 25*25 pixels as input, and reduces the dimension of the input to 17-unit vectors. As mentioned above, DLPFC receives the “thoughtful” pattern as input and amygdala receives the emotional patterns as inputs in addition, OFC categorizes the input patterns from outside world directly. The DLPFC-OFC loop works if the recalled pattern of Hopfield network is “Thoughtful”, and it accounts for generating deliberative rules. On the other hand, the OFC-amygdala works if the recalled pattern is one of the emotional patterns, and it deals with heuristic rules and some deliberative rules.

The anterior cingulate (ACC) has a function to detect the potential conflict between the outputs of DLPFC and OFC (or OFC and amygdala). It is viewed as a selective constant vigilance: the vigilance is very high if the recalled pattern is the “thoughtful” pattern, or it is low if the recalled pattern is an emotional pattern. The decision-making loop outputs heuristic rules if the similarity between the two RBM outputs is higher than the vigilance; otherwise it makes deliberative decisions. Since the vigilance for the DLPFC-OFC loop is very high, it is impossible that the DLPFC-OFC loop makes heuristic decisions. For OFC-amygdala loop, if mismatch occurs (i.e. similarity smaller than the vigilance), the decisions are made based on the emotion output from amygdala; otherwise, it makes sophisticated rules.

The first task of the reinforcement learning is to reinforce the positive patterns stored in the Hopfield network. It works with a variety of adjustable parameters, such as exploration rate, discount factor, and learning rate. Different combinations of the values of these parameters represent the patient in various situations. Secondly, the reinforcement learning aims to reinforce the control from the OFC to amygdala by simply increasing the parameter of the controller in the feedback control system. Such feedback system consists of the OFC as controller, and the amygdala as the plant to be controlled. It aims to adjust the output of amygdala to get close to the output of OFC, which is treated as a reference signal in the control loop.

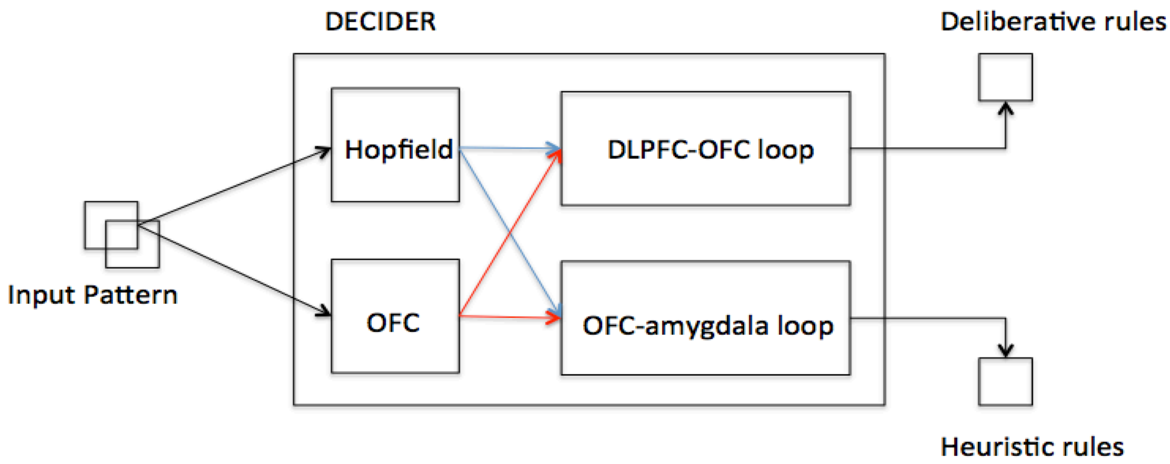


FIGURE 6.1 An illustration of DECIDER coping with the outside world.

ALGORITHM 5.1: **A Q-learning algorithm**

1) For each state-action pair (s, a), initialize the table entry $Q(s,a)$ to the corresponding value in the state vector
2) Observe state s
3) **do**
4) select and do action according to Boltzmann selection rule
5) receive immediate rewards r
6) observe the new state s'
7) update the table entry for $Q(s,a)$ as following:

$$Q(s,a_p) \leftarrow Q(s,a_p) + \lambda[R_{a_p}(s,s') + \gamma]$$

8) $s \leftarrow s'$
9) **until** termination conditions met

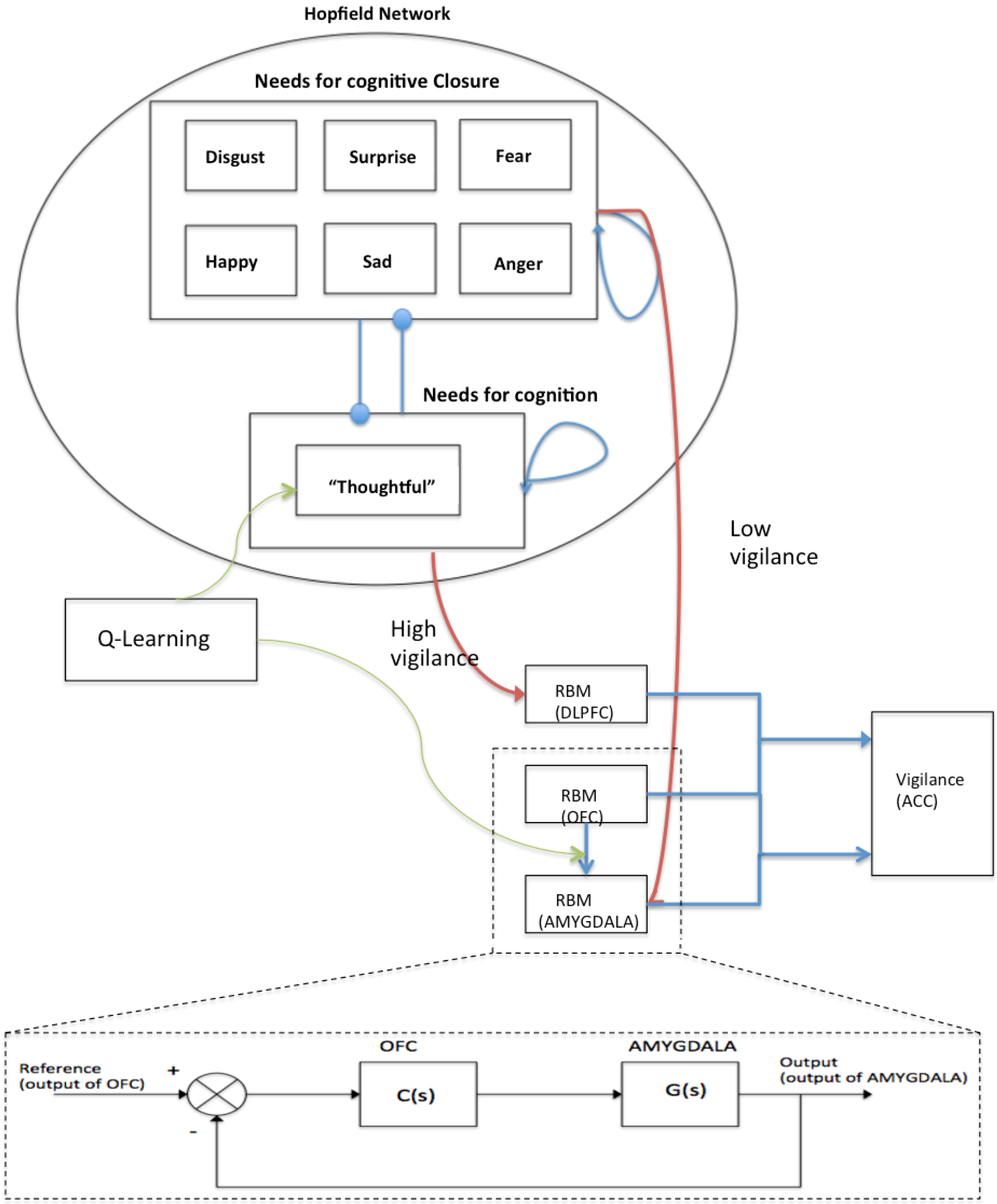


FIGURE 6.2 DECIDER model combining Hopfield network as needs network, and RBMs as decision network. Q-learning influences the attractors in the Hopfield network and the control from OFC to amygdala. Needs for cognitive closure outputs low-level vigilance signals to ACC. Needs for cognition outputs high-level vigilance signals to ACC. The control from OFC to amygdala is designed as a feedback back control loop.

Chapter 7

Evaluation and Conclusion

7.1 Evaluation

- **Hopfield**

The experiments implemented in Chapter 2 have shown that increasing the degree of a strong attractor makes the attractor more stable, as well as that increasing the number of other simple attractors destabilizes the strong attractor. It proves the results of previous study on the stability of strong attractors (Edalat, 2003).

Limitation

As discussed Chapter2, the Hopfield network was trained using the standard Hebb's rule, which is lack of the capability to deal with correlation issues. Correlation issues introduce some spurious states into the network. It may be a problem if there are two or more strong attractors having the same degree. The recalled pattern will be the linear combination of these attractors instead of any of them, which is meaningless and cannot be identified by the trained RBM. The correlation issue can be solved by some techniques such as pseudo-inverse methods.

- **Restricted Boltzmann Machine**

The RBM trained in Chapter 3 can precisely categorises the seven clusters, which are generated via adding random noise on seven emotion images.

Limitation

The training efficiency of this learner may not be good enough since we only cared about its classification performance. Furthermore, we used some emotion images as training examples rather than real emotional sequences, which made the classification task much simpler.

- **DECIDER and Reinforcement Learning**

The supermodel DECIDER consists of a needs network, which is thought of a competitive attractor network, and a decision making network, in which two loops work for decisions in different level.

DECIDER simulates the procedure for decision-making, which is emotionally influenced and goal-directed. In the DECIDER, the Hopfield network works as a competitive attractor network that is analogous to hypothalamic/midbrain/brain stem network in one's brain. Emotional signals influencing the decision making is modelled using strong attractors stored in the Hopfield network. And then it outputs emotional signals or "thoughtful" signals to the decision making network, which is made of three identical trained RBMs. This network generates decisions dependent on the signals from the Hopfield network, and the similarity of the outputs of the three RBMs. We also simulated the control from the orbitofrontal cortex to amygdala using a feedback control loop. Reinforcement learning accounts for reinforcing the positive attractors in the Hopfield network and reinforcing the OFC control by simply increasing the value of the controller parameter.

Limitation

In the needs network, instead of the hierarchy of Maslow's needs, we assumed that the needs included in it are needs for cognition, which is defined as the tendency to think about issues deeply, and needs for cognitive closure, which is defined as the tendency to make a decision quickly.

When implementing the feedback control loop for simulating the control from the OFC to amygdala, we assumed that the neurons are neurobiological rather than binary units so that we can model the dynamics of the neuron population to be controlled.

The OFC in the feedback control loop was assumed to be a proportional controller, which is the simplest controller in the control-engineering world. And the parameter is not refined during the procedure of the reinforcement learning. For simplicity, it is changed after the Q-learning finished.

There is no control activity from the DLPFC to OFC.

7.2 Future Work

Here I concluded that works could be done for future research:

- Improving the performance of the Hopfield network so that it is robust to correlation patterns.
- Training the restricted Boltzmann machine to be able to categorise emotions on real human face, so that the system is able to cope with the real world.
- Using artificial neural network to represent the control from the OFC to amygdala, so that the model is more biologically plausible.
- Modeling the connections between the DLPFC to OFC.
- Designing a reinforcement learning that reinforce the control in terms of time (or iteration in discrete domain)

- Redesigning the super model using spiking neurons.

7.3 Conclusion

The model built here can be thought of the pioneer to model attachment types in psychotherapy. There are still some assumptions we have made and drawbacks we have not overcome due to the time restrictions. The topic that modelling the attachment types and the changes between them in a neural network is interesting, and I will carry on in this topic following the results I achieved in this project.

Bibliography

Ackley, D.H., G.E. Hinton, and T.J. Sejnowski (1985). A Learning Algorithm for Boltzmann Machines. *Cognitive Science* **9**, 147-169.

Barbas, H. (1995). Anatomic basis of cognitive-emotional interactions in the primate prefrontal cortex. Neuroscience & Biobehavioral Reviews, 19(3), 499-510.

Cacioppo, J. T., & Petty, R. E. (1982). The need for cognition. Journal of personality and social psychology, 42(1), 116.

Carpenter, G. A., & Grossberg, S. (1987). A massively parallel architecture for a self-organizing neural pattern recognition machine. Computer vision, graphics, and image processing, 37(1), 54-115.

Cittern, D., and Edalat, A (2013). An Adaptive Model of Child-Parent Relationships. *Unpublished.*

De Neys, W., Vartanian, O., & Goel, V. (2008). Smarter Than We Think When Our Brains Detect That We Are Biased. Psychological Science, 19(5), 483-489.

Dias, R., Robbins, T. W., & Roberts, A. C. (1996). Dissociation in prefrontal cortex of affective and attentional shifts. Nature, 380(6569), 69-72.

Edalat, A., & Mancinelli, F. (2013a). Strong Attractors of Hopfield Neural Networks to Model Attachment Types and Behavioural Patterns. In Proceedings of International Joint Conference on Neural Networks (IJCNN).

Edalat, A. (2013b). Capacity of Strong Attractor patterns to model behavioural and cognitive prototypes. In the Proceedings of The *Neural Information Processing Systems (*NIPS).

Eliasmith, C., & Anderson, C. C. H. (2004). *Neural engineering: Computation, representation, and dynamics in neurobiological systems* (pp. 112). MIT Press.

Hariri, A. R., Bookheimer, S. Y., & Mazziotta, J. C. (2000). Modulating emotional responses: effects of a neocortical network on the limbic system.Neuroreport, 11(1), 43-48.

Hebb, D. O. (1949). *The organization of behavior: A neuropsychological approach*. John Wiley & Sons.

Herz, J., Krogh, A., & Palmer, R. G. (1991). Introduction to the theory of neural computation. *Lecture Notes Volume in the Santa Fe Institute Studies in the Sciences of Complexity. Cambridge, MA, Perseus Books*.

Hinton, G. E. (2002). Training products of experts by minimizing contrastive divergence. *Neural computation*, *14*(8), 1771-1800.

Hinton, G. E., Osindero, S., & Teh, Y. W. (2006). A fast learning algorithm for deep belief nets. *Neural computation*, *18*(7), 1527-1554.

Hinton, G. E., & Salakhutdinov, R. (2009). Replicated softmax: an undirected topic model. In *Advances in neural information processing systems* (pp. 1607-1614).

Hinton, G.E. and T.J. Sejnowski (1983). Optimal Perceptual Inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Washington 1983), 448-453. New York: IEEE.

Hinton, G.E. and T.J. Sejnowski (1986). Learning and Relearning in Bolzmann Machines. In *Parallel Distributed Processing*, vol. 1, chap. 7.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, *79*(8), 2554-2558.

Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement learning: A survey. *arXiv preprint cs/9605103*.

Levine, D. S. (2009). Brain pathways for cognitive-emotional decision making in the human animal. Neural Networks, 22(3), 286-293.

Maslow, A. H., & Lowry, R. (1968). Toward a psychology of being.

Mohamed, A. R., & Hinton, G. (2010, March). Phone recognition using restricted boltzmann machines. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on* (pp. 4354-4357). IEEE.

Optimistic Initial Values, http://webdocs.cs.ualberta.ca/~sutton/book/ebook/node21.html

Rolls, E. T., Hornak, J., Wade, D., & McGrath, J. (1994). Emotion-related learning in patients with social and emotional changes associated with frontal lobe damage. Journal of Neurology, Neurosurgery & Psychiatry, 57(12), 1518-1524.

Rothbaum, B. O., Meadows, E. A., Resick, P., & Foy, D. W. (2000). Cognitive-behavioral therapy.

Salakhutdinov, R., Mnih, A., & Hinton, G. (2007, June). Restricted Boltzmann machines for collaborative filtering. In *Proceedings of the 24th international conference on Machine learning* (pp. 791-798). ACM.

Schore, Allan N.(1998). "The experience-dependent maturation of an evaluative system in the cortex." Brain and values: Is a biological science of values possible: 337-358.

Schore, Allan N. (2003). Affect dysregulation & disorders of the self (pp. 60 & pp.69). New York: WW Norton.

Shedler, J. (2010). The efficacy of psychodynamic psychotherapy. *American Psychologist*, *65*(2), 98.

Shteingart, H., Neiman, T., & Loewenstein, Y. (2013). The role of first impression in operant learning. Journal of Experimental Psychology: General,142(2), 476.

van Veen, V., & Carter, C. S. (2006). Conflict and cognitive control in the brain.Current Directions in

Psychological Science, 15(5), 237-240.

Watkins, C. J., & Dayan, P. (1992). Q-learning. Machine learning, 8(3-4), 279-292.

Watkins, C. J. (1989), Learning from delayed rewards. PhD thesis, University of Cambridge.

Webster, D. M., & Kruglanski, A. W. (1994). Individual differences in need for cognitive closure. Journal of personality and social psychology, 67(6), 1049.

Wedemann, R. S., Donangelo, R., & de Carvalho, L. A. (2009). Generalized memory associativity in a network model for the neuroses. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, *19*(1), 015116-015116.

Young, J. E., Klosko, J. S., & Weishaar, M. E. (2003). Schema therapy: A practitioner's guide(pp.3-5 & pp. 43-44). Guilford Press.