university of groningen
faculty of mathematics and natural sciences

Simulation of deliberation with a simple neural network

Bachelor's thesis

July 2012

Student: Martien Scheepens

Primary supervisor: prof.dr. Michael Biehl

Secondary supervisor: prof. dr. Alexandru Telea

Contents

Chapter 1

Introduction

Stephan Hartmann and Soroush Rafiee-Rad analyzed the difference between voting and deliberation in various groups. They showed that deliberation and voting perform different in finding the truth. In general voting is the better method to find the truth, but for small groups with group members who differ in quality, deliberation has a greater chance to arrive at truth. In this thesis it is investigated, if it is possible to simulate deliberation with perceptrons as group members (a perceptron is a simple neural network that can be trained to make decisions for linear problems) and if the results are comparable with the results of Hartmann and Rafiee-Rad. Both questions were answered positively. Furthermore a relation was found which makes it possible to compare heterogeneous groups and to predict the result based on results of other groups.

Chapter 2

Theory

2.1 Perceptron

A perceptron is a layered feed-forward network created by Frank Rosenblatt in 1957 [1] [2]. A perceptron has an input layer, an output layer and zero or more hidden layers. The weight of each connection can be adjusted. The weights of every connection are stored in weight vectors $\mathbf{w}_i$. The output of a perceptron is often determined with a threshold function, for example a sign function. Before using a perceptron to classify input, it must be trained by a teacher. To train the perceptron we need a set of examples (input) ξ and the expected output $E^{\mu(t)} = \mathbf{w}(t) \cdot xi^{\mu(t)} S^{\mu(t)}$, where $S = \pm 1$ are labels.
The learning rule for a perceptron network (Rosenblatt algorithm) is

$$\mathbf{w}(t+1) = \begin{cases} \mathbf{w}(t) + \frac{1}{N}\xi^{\mu(t)} S^{\mu(t)} & if\ E^{\mu(t)} \leq 0 \\ \mathbf{w}(t) & else \end{cases} \tag{2.1}$$

The perceptron is not updated if the signum of the output was correct, otherwise a correction term is added to the weight vector $\mathbf{w}_i$

2.2 Hebbian learning

Hebbian learning was introduced by Donald Hebb in 1949 [1] [3] and originates from neuroscience. It describes a possible way for learning in the human brain and is applied in neural networks, too.
Hebbian learning is local. The output $S^{\mu(t)}$ of a simple system is matrix multiplication between the input vectors ξ and the weight vectors $\mathbf{w}$.

$$S^{\mu(t)} = \xi^T \mathbf{w} \tag{2.2}$$

The update of a weight vector $\mathbf{w}_i$ is a multiplication of the output $S^{\mu(t)}$ of the system with the input vector ξ_i. η is a factor to control the change of the learning rate.

$$\Delta \mathbf{w}_i = \eta S^{\mu(t)} \xi_i \tag{2.3}$$

2.3 Deliberation versus Voting

Stephan Hartmann and Soroush Rafiee-Rad compared the results for voting and deliberation in various groups with different sizes [4].
The results for voting are calculated with Condorcet's jury theorem. Condorcet's jury theorem describes the chance of a group arriving at the truth. The voters are independent and have the same probability p to make the correct decision (homogeneous group). If the probability $p > \frac{1}{2}$, the group will make the correct decision. Furthermore, increasing the group size will result in a better decision of the group. If the probability $p > \frac{1}{2}$ and $p_1 > p_0$, the result for p_1 is better than for p_0[5].
Deliberation means that the members of the group try to convince other members, aim is to reach a consensus. The results for deliberation are calculated with a Bayesian Model and two variables on the reliability of each member. It is shown by Hartmann and Rafiee-Rad that deliberation is truth-conductive.
Conclusion of Hartmann and Rafiee-Rad was that voting has a higher chance of arriving at truth for homogeneous and almost homogeneous groups. For heterogeneous groups deliberation performs better than voting with a maximum for the group size of 50 members.

Chapter 3

Building a neural network

I want to show that a simple neural network can be used to simulate the deliberation phase. The steps to modify perceptrons for this task are shown in the next sections.

3.1 Creating an ensemble

A requirement for the ensemble is that the students have a predefined distance to the teacher at the start of the simulation. It is possible to make an ensemble with regular perceptron training and a stop condition in the algorithm, but constructing the students mathematically is quicker and models the fact that perceptron training is not completely reliable. Also comparison of simulation runs is easier since the results of these runs can be compared based on the start value.
To construct the perceptrons, the perceptrons must fulfill $|\mathbf{w}_i| = 1$ and $\mathbf{w}_i \cdot \mathbf{B} = R$ with $\mathbf{B}$ as teacher and the preset overlap R, R is defined as

$$R \equiv \mathbf{w} \cdot \mathbf{B} \tag{3.1}$$

The first step is to create a normalized vector $\tilde{\mathbf{w}}_i$ with random components, the initial state is $|\tilde{\mathbf{w}}_i| = 1$. Then we have to calculate a and b for equation 3.2 from properties 3.3 and 3.4.

$$\mathbf{w}_i = a \cdot \tilde{\mathbf{w}}_i + b \cdot \mathbf{B} \tag{3.2}$$

$$\mathbf{w}_i \cdot \mathbf{w}_i \Leftrightarrow a^2 + 2ab \cdot \tilde{\mathbf{w}}_i \cdot \mathbf{B} + b^2 = 1 \tag{3.3}$$

$$\mathbf{w}_i \cdot \mathbf{B} \Leftrightarrow a \cdot \tilde{\mathbf{w}}_i \cdot \mathbf{B} + b = R \tag{3.4}$$

The solutions for a and b are

$$a = \sqrt{\frac{1 - R^2}{1 - \tilde{\mathbf{w}}_i \cdot \mathbf{B}}} \tag{3.5}$$

$$b = R - a\tilde{\mathbf{w}}_i \cdot \mathbf{B} \tag{3.6}$$

With calculating a and b and substituting 3.5 and 3.6 in 3.2, we get normalized vectors (students) which are placed on a multidimensional cone with the teacher vector on the central axis. The mean of the students is in a ideal situation identical to the position of the teacher vector.

3.2 Communication in the ensemble

We want to simulate a phase of deliberation within the ensemble. There are several possibilities for implementing the discussion:

- Random: a student acts as teacher for a randomly chosen student.
  The student and the teacher are chosen randomly. The roles are shuffled every iteration.
- Consulting: all students act as teacher for one student.
  One student is appointed randomly the role of student. All other students will act as teacher for that student. The sequence of teachers is randomized.
- Broadcasting: one student acts as teacher for the entire ensemble.
  One student is appointed randomly the role of teacher. The student will fulfill this role during one round and will communicate with every other student once.

To determine which method is the best the methods have to be compared. The methods can be compared on how well the ensemble grows together and what the difference with the teacher is.
The overlap of the group with the teacher (see formula 3.1) will be used to determine the best method. Aim is to find the method with the highest overlap R:

$$R \equiv \frac{\overline{w_i} \cdot B}{|\overline{w_i}|} \tag{3.7}$$

The mean of all possible overlap combinations between all students. This can be used to measure the distances of the members of the group.

$$S \equiv \overline{\frac{w_i \cdot w_j}{|w_i||w_j|}} \tag{3.8}$$

The mean of the overlap of every student with the teacher:

$$T \equiv \overline{\frac{w_i \cdot \mathbf{B}}{|w_i|}} \tag{3.9}$$

The values in the table show that all types of communication have the same starting position. The results for choosing randomly a student generates the best results for the overlap with the teacher (R_{final}). Broadcasting performs worst, but all students in the ensemble merge towards one opinion (S_{final}). The performance of broadcasting is probably bad because of the importance of one student. He can tell all the other students his opinion and all other students will adjust their opinion and move towards this student. Why consulting is not working was not investigated. The approach with random roles obviously performs best.

	random	consulting	broadcasting
$T_{initial}$	0.7	0.7	0.7
$S_{initial}$	0.5	0.5	0.5
$R_{initial}$	0.98	0.98	0.98
T_{final}	0.89	0.69	0.53
S_{final}	0.90	0.73	0.99
R_{final}	0.95	0.81	0.51

Table 3.1: Results for different types of communication

3.3 Deliberation in the ensemble

The deliberation phase in the discussion is modeled by the learning step of the ensemble. As shown earlier the students are placed on a cone with the teacher as its center. The ideal method should contract the circle of the cone where the students are placed uniformly. The method with random pairing approaches will be used as communication method.

3.3.1 Rosenblatt

In a normal perceptron architecture the Rosenblatt algorithm is used, equation 2.1. If the condition $E^{\mu(t)} \leq 0$ is true, the perceptron is updated. The condition is based on the angle of the teacher to the students and the aim is to minimize this angle.
A possible output of the simulation is shown in figure 3.1. The line in the plot is $R(t)$ (see formula 3.7), the overlap of the group with the teacher. The quality of the output of the ensemble decreases with every iteration.

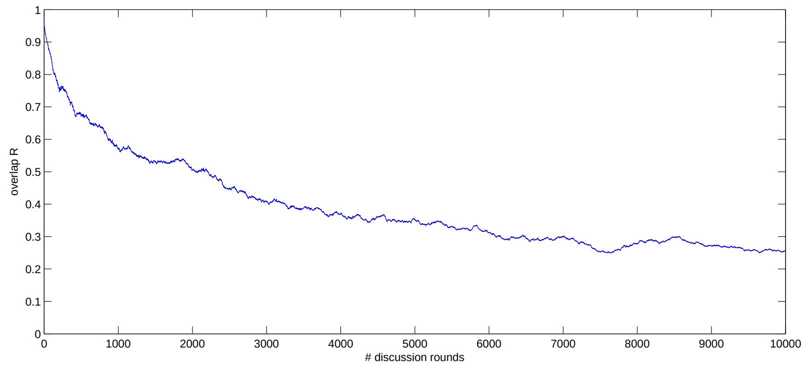


Figure 3.1: Simulation with Rosenblatt learning step

In a situation where the omniscient teacher is absent and replaced by a (randomly chosen) student acting as teacher, unwanted behavior occurs. If the ensemble members have a similar opinion on one example, no communication takes place and there is no learning effort. If the ensemble members differ in opinion, the member in the role of student is forced to recalculate his position and is shifted towards his teacher. The effect of these two options is that the difference between the mean of the ensemble and the truth increases. After moving away from the truth, the ensemble will not be able to find the truth anymore. Furthermore the ensemble cannot merge to one common output, since they only recalculate positions if the difference is great enough.

3.3.2 Hebbian Learning

The solution for the problem is to manipulate the condition $E^{\mu(t)} \leq 0$. It is possible to drop the condition, which means that we use Hebbian Learning for the deliberation phase [6] [7]. Another possibility is to use the opposite condition $E^{\mu(t)} > 0$. This means that agreements between ensemble members are enhanced, while disagreements are ignored. After changing the condition, the output of the ensemble is stable. The results for both options are similar after a great number of steps, the vectors of the students merge towards their mean. The results are shown in figure 3.2.
The option with the inversed condition was chosen for the next tests since the results are slightly better.

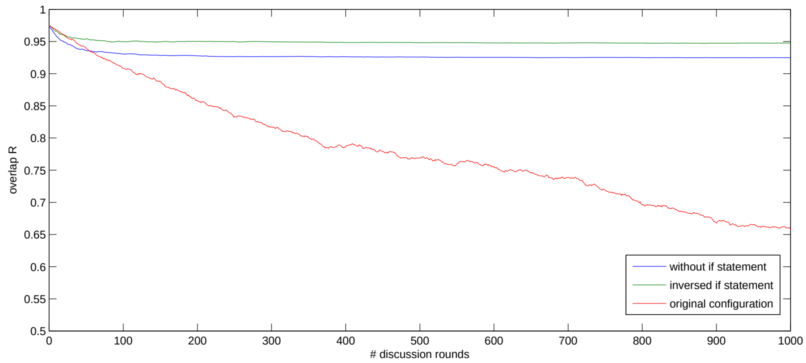


Figure 3.2: Comparison of Rosenblatt and Hebb

Chapter 4

Simulation and Results

For all simulations and graphics the following parameters were used unless indicated otherwise: $R = 0.7$, $n = 20$, $ensemblesize = 20$, $durationdiscussion = 1000$. The discussion length of 1000 rounds (with each having $ensemblesize$ interactions) was arbitrarily chosen, the output of the ensemble is usually stable after about 500-600 rounds. The ensemble size was chosen for a good performance/quality ratio.

4.1 Effect of the initial overlap

The initial overlap $R_{initial}$ is a parameter in the procedure to create an ensemble 3.6. The effect of manipulating $R_{initial}$ is shown in figure 4.1. The prediction of the ensemble (R_{final}) gets better with an increased $R_{initial}$.

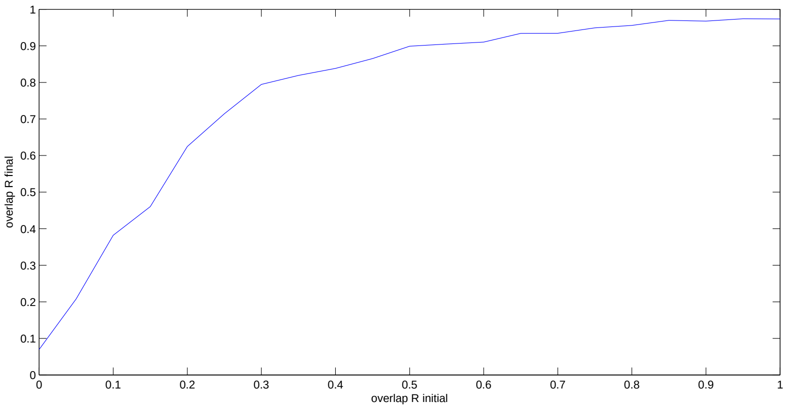


Figure 4.1: Effect of $R_{initial}$ on the final overlap R_{final}

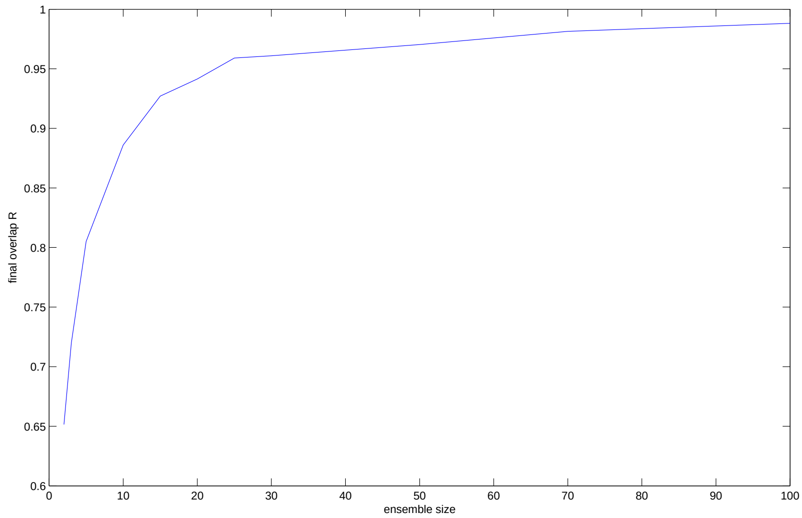


Figure 4.2: Effect of the ensemble size on the final overlap R_{final}

4.2 Effect of the ensemble size

The ensemble performs better with more members, see figure 4.2. Since the members of the ensemble are placed on a circle with the teacher as center, a sufficient number of members in the ensemble is important. With a few students the chance is low that these students have equal distances to each other, with a high number it is more likely that the distribution is uniform. A second aspect is that unsupervised learning needs data with redundancy, otherwise the network cannot separate noise from patterns.

4.3 Effect of size of the input pattern

The dimensionality of the input pattern has very little or no effect on the final overlap R_{final} (see table below).

dimension	5	10	20	30	50	100
final overlap R_{final}	0.955	0.943	0.954	0.935	0.942	0.942

4.4 Effect of using a limited set of examples

All results in the preceding tests are made with an ensemble that has access to an unlimited set of examples (examples are generated randomly when needed). Usually the amount of available data is limited and is divided in a training set and a test set. To simulate a test set, the patterns are generated before starting the algorithm, a random function picks every round an example. As shown in figure 4.3 more examples will give a better result. The growth seems to be of a logarithmic nature.

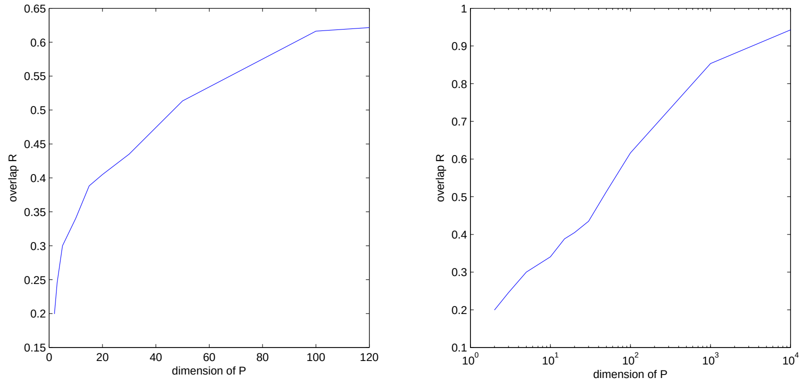


Figure 4.3: Effect of size of the set with examples on the final overlap R_{final}. On the left with linear scale, on the right with logarithmic scale

4.5 Heterogeneous ensembles

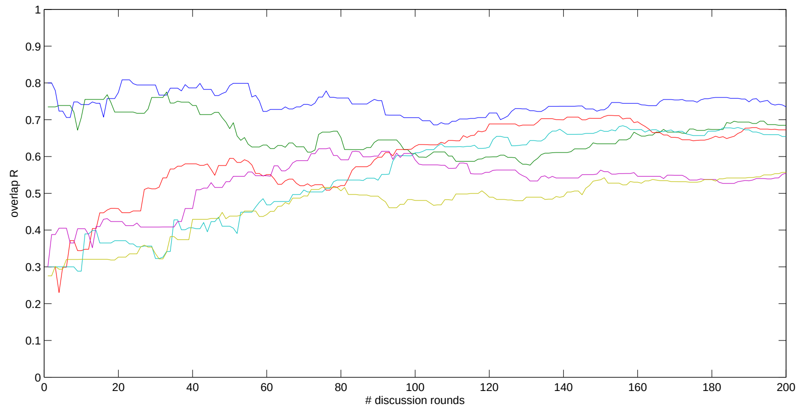


Figure 4.4: Evolution of student overlap $R_i(t)$ in a small simulation with a heterogeneous ensemble, 4 members with $R_{initial} = 0.3$ and 2 members with $R_{initial} = 0.8$

Heterogeneous ensembles are ensembles where a part of the group has an other initial overlap $R_{initial}$ than the other part. The question is, if un-

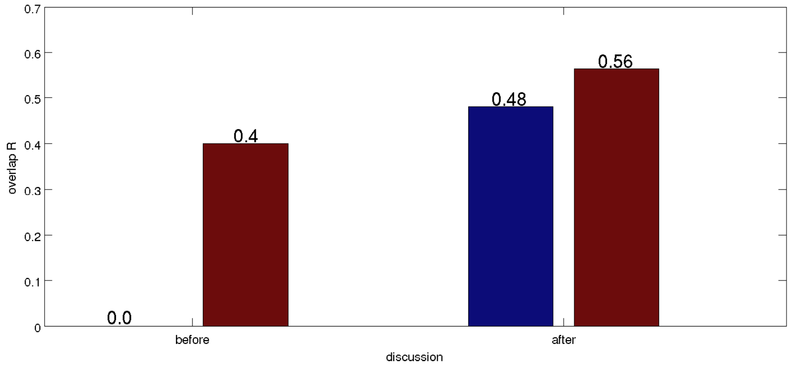


Figure 4.5: Overlap of two equal sized groups before and after the discussion

expected behavior occurs or if the heterogeneous ensemble behaves like a homogeneous ensemble.
A simple simulation with six students (four students have $R_{initial} = 0.3$, two students have $R_{initial} = 0.8$) shows that the outputs of the members merge (figure 4.4). The students with the low R improve a lot, while the students the R value of the other students decreases slightly (see also figure 4.5).
The simulation was repeated with different values for R_1 and R_2 and varying ratios $R_1 : R_2$. To compare the overlap R_{final} from the different simulations, the mean of the overlap at the start was calculated:

$$R_{heterogenous} = \sum_{i=0}^{n} \frac{R_i}{n}. \tag{4.1}$$

The result is plotted in figure 4.6.

The plot shows just one line, it appears that a relationship exists between the mean of all R_i in a group and R_final. If one knows all R_i in an ensemble, it is possible to make a prediction for the final overlap. A heterogeneous group can be simulated with a simpler homogeneous group of the same size.

4.6 Reliability

A student will learn from any other student who acts as teacher (unless the if statement discussed in 3.2 is not true). In real world some people believe or disbelieve other people.
The model of Hartmann and Rafiee-Rad uses two reliabilities for the group members. The first reliability is the chance of making the correct decision. The other value is the chance of a group member to see what the first reliability of an other group member is. A low value for the second reliability

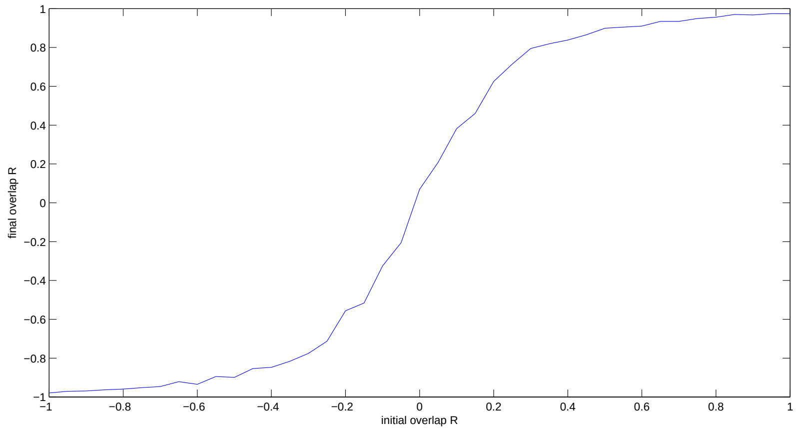


Figure 4.6: Final overlap of the students as function of the mean of the initial overlap

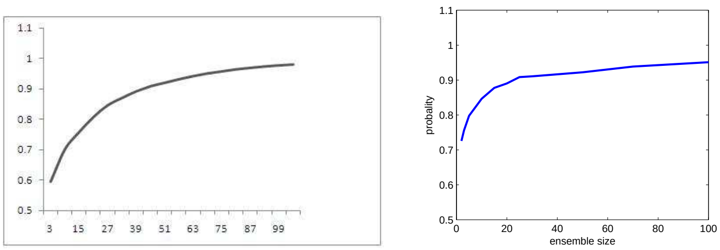
means that a member will not learn from other members since he cannot track their opinion.
To simulate reliabilities in my model, each student has a static vector with reliability values for the other students. Using the reliability vectors for a homogeneous group is meaningless, the reliability value behaves as a factor which just slows the process down.
A simulation with two (or more) independent groups with different R_i provides no interesting results. If the members of each group are distributed uniformly on the cone, each group should arrive near the truth. The effect is comparable with reducing the ensemble size. The use of reliability vectors for a heterogeneous group is for the same reason not interesting.

Chapter 5

Conclusion and Outlook

5.1 Comparison with Hartmann & Rafiee-Rad

The simulation of the deliberation process in a group was possible with regular perceptrons. The result of the deliberation process with perceptrons cannot be compared easily with the results of Hartmann & Rafiee-Rad. The figures below look similar, but by adjusting the perceptron parameters the right figure could look the same as the left figure. This might look more impressive but the mathematical model of Hartmann & Rafiee-Rad is more sophisticated than simple perceptron interactions. Therefore I can only conclude that in both models there is a relationship between the group size and the probability that the group chooses the correct answer. If the group size increases, the probability increases with the value $p_{max} = 1$ as asymptote.



(a) Hartmann and Rafiee-Rad (b) section 4.2

Figure 5.1: Results of Hartmann & Rafiee-Rad and my work for the group size (homogeneous group), figures have similar scale.

5.2 Outlook

It was shown that perceptrons are suitable to simulate deliberation in a group, with the best result if the order of students is chosen randomly. It might be interesting to check whether other learning algorithms will complete the task and the results are comparable. I looked briefly into two other types of communication and considered broadcasting and consulting as not suitable for the task. It should be possible to find other types of interaction for the ensemble. The interaction could be based on the distance of the two students.
Another improvement is to create a dynamic reliability vector. A reliability value could be increased if the two students influenced each other earlier and decrease if they repel each other. Also the reliability vector could be based on the number neighbors. In other words, if many other students are near to the chosen student of that moment, the reliability increases.

Bibliography

[1] J. Hertz, A. Krogh & R.G. Palmer (1991). *Introduction to the theory of neural computation.* Boulder: Westview Press.

[2] F. Rosenblatt (1957). The Perceptron–a perceiving and recognizing automaton. Report 85-460-1, Cornell Aeronautical Laboratory.

[3] D.O. Hebb (1949). *The Organization of Behavior.* New York: Wiley and Sons.

[4] S. Hartmann & S. Rafiee-Rad (2010). *Voting, Deliberation and Truth.* `http://stephanhartmann.org/HartmannRafieeRad_VDT.pdf`. Retrieved on 01-12-2011.

[5] P.J. Boland (1989). Majority systems and the Condorcet Jury Theorem. *The Statistician*, **38**, 181-189.

[6] D. Bolle & G.M. Shim (1995). Nonlinear Hebbian training of the perceptron. *Network: Computation in Neural Systems*, **6**, 619-633.

[7] M. Biehl & A. Mietzner (1993). Statistical Mechanics of Unsupervised Learning. *Europhysics Letters*, **24**, 421-426.

Matlab code

The source code is hosted at `http://martien.home.fmf.nl/scriptie/`.

Acknowledgement

I would like to thank Soroush Rafiee-Rad for providing the figures of his research.