

## Using habituation in machine learning

Stephen Marsland

Computer Science Group, School of Engineering and Advanced Technology, Massey University, Private Bag 11-222, Palmerston North 4442, New Zealand

### ARTICLE INFO

#### Article history:

Received 1 March 2008

Revised 26 May 2008

Accepted 26 May 2008

Available online 21 July 2008

#### Keywords:

Habituation

Machine learning

Neural network

Novelty detection

### ABSTRACT

Habituation, a decrement in response to a stimulus that is presented repeatedly without ill effect, can be identified in almost all animals. It can also be used in machine learning to provide a variety of different applications, such as novelty detection, recency encoding, and temporal signal pre-processing. This paper examines how habituation can be mathematically modelled, and discusses how well these models fit the revised characteristics of habituation. It then demonstrates how the models can be combined with neural networks in order to realise the various applications. Finally, some simple experimental results are presented that demonstrate the effectiveness of the methods.

© 2008 Elsevier Inc. All rights reserved.

### 1. Introduction

Other papers in this issue discuss the effects of habituation, and the mechanisms that cause it, in real learning systems. The importance of these investigations is clear, especially when the comment of Zeaman (1976, chap. 9) is taken into consideration:

“Habituation is like rats and cosmic rays. If you are a psychologist, it is hard to keep them out of your laboratory”

This paper differs from the others in this issue in that it does not examine habituation in an animal. Rather, it considers numerical models of habituation, and how and why they should be used in artificial intelligence and machine learning. The ubiquity of habituation in animals suggests that it is of great importance, and this is sufficient reason to study whether or not it can have application in artificial as well as natural learning systems. The success of neural networks as an algorithmic model came about because of the abstraction of a mathematical model of the neuron as an integrate-and-fire device (McCulloch & Pitts, 1943). Connected with a very simple model of synaptic plasticity (usually Hebbian learning; see e.g., Rumelhart, McClelland, and the PDP Research Group, 1986) they enable computer algorithms to learn to identify structures in data.

Habituation can be studied in this light as a very simple model of neuronal learning, simply that synaptic efficacy decreases as stimuli are perceived repeatedly without either positive or negative effects.

There are two types of model that can be developed. The first is to model the *effects* of habituation, whilst the second is to model the biological *mechanisms* by which habituation occurs. Whilst this second can be a useful check of such mechanisms when they are postulated in order to check that they do provide the expected behaviour, they are not the focus here. Rather, in this paper, simple models of habituation are used to demonstrate that habituation can provide a set of useful behaviours for learning, here machine learning. Instead of studying habituation in order to better understand it, we seek to exploit mechanisms that demonstrate it for practical use—rather than asking what we can do for habituation, we ask what habituation can do for us.

In addition to the models of habituation that are described in the paper, several researchers have proposed more complete models of learning that include habituation. Several of these models are aimed at modelling infant learning, including (Sirois & Mareschal, 2002, 2004; Thomas & Gilmore, 2004), and whilst not directly related to the habituation modelling described here, (Thomas & Gilmore, 2004) is of interest for its examination of novelty detection in infants.

In this paper, we will consider three different problems: novelty detection (and attention focus), recency detection, and temporal learning. These are chosen as three simple demonstrations of the utility of habituation in machine learning. We will couple standard neural networks with simple models that demonstrate some of the characteristics of habituation (as discussed in Section 2.1) and provide experimental evidence that the resulting system does provide the expected behaviour. Other artificial learning systems that use habituation can certainly be created, but the three problems chosen demonstrate the different aspects of habituation clearly and produce simple but effective learning systems.

E-mail address: [s.r.marsland@massey.ac.nz](mailto:s.r.marsland@massey.ac.nz)

## 2. Methods

### 2.1. Characteristics and models of habituation

Elsewhere in this issue, (Rankin et al. (2008)) the revised characteristics of habituation are presented, based on the original list of Thompson and Spencer (1966). In this section, we discuss which of these are of particular interest for the machine learning applications that were identified above.

The primary effect of habituation is that repeated application of the stimulus results in progressive decrease in the response, and this is clearly an essential for any model that claims to be demonstrating habituation. In general, the decrement is considered to be exponential decay to some (possibly non-zero) asymptotic level, and this is simple to model mathematically, leading to the most commonly found model of habituation, which consists of a single differential equation:

$$\tau \frac{dy(t)}{dt} = \alpha(y_0 - y(t)) - S(t). \quad (1)$$

Within this model,  $y(t)$  is the strength of the signal, which starts at value  $y_0$  and decays exponentially. There are two parameters of the model,  $\alpha$  and  $\tau$ . The former controls the rate of recovery, whilst the second is the time constant. It is possible to fit these to experimental data, which is the approach taken by Alonso, Moreno, Vázquez, and Santacreu (2005), but here typical values are taken, namely  $y_0 = 1, \tau = 33, \alpha = 1.05$  and  $1.2$  in Fig. 1 and  $y_0 = 1, \tau = 100, \alpha = 1.05$  and  $1.2$  in Fig. 2. In the figures, a stimulus is presented for 250 presentations, and then not presented for a time equivalent to 100 before being re-presented.

$S(t)$  is a variable that indicates whether or not a stimulus is applied at the current time step (the length of which is unspecified by the model).  $S$  is often taken to have value  $S(t) = 1$  if the stimulus is perceived at the current timestep, and  $S(t) = 0$  otherwise. Given these two values it is possible to solve the differential equation explicitly using an integrating factor, which leads to two different behaviours depending on the value of  $S$ . If  $S$  has a non-zero value then:

$$y = y_0 - \frac{S}{\alpha} (1 - \exp(-\alpha t / \tau)), \quad (2)$$

whilst if  $S = 0$  then the model shows exponential recovery towards the original value  $y_0$  (where  $y_1$  is the value it reaches at the end of the habituation training):

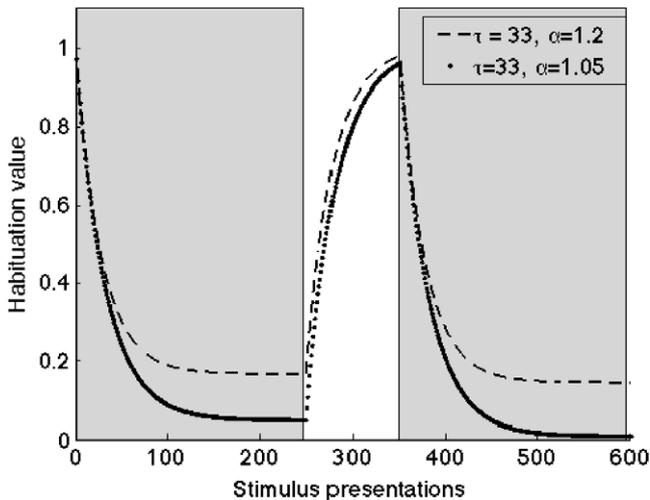


Fig. 1. Habituation curve based on Eq. 1 with a relatively small time constant  $\tau$ . The shaded regions show when a stimulus was presented.

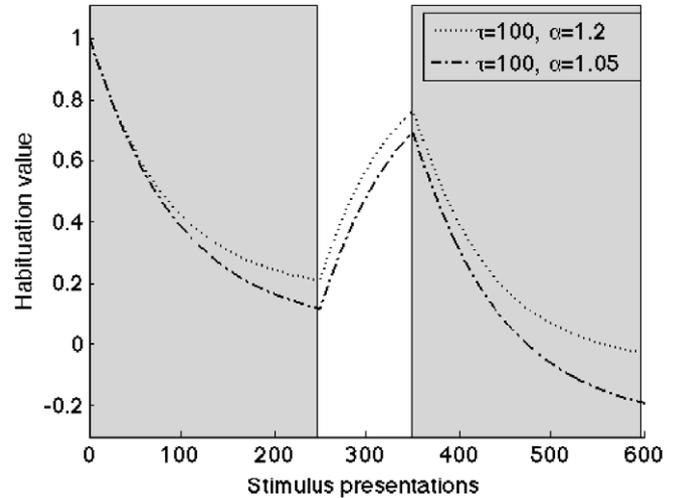


Fig. 2. Habituation curve based on Eq. 1 with a large time constant  $\tau$ . The shaded regions show when a stimulus was presented.

$$y = y_0 - (y_0 - y_1) \exp(\alpha t / \tau). \quad (3)$$

This model is often credited to Stanley (1976), but in his paper he refers to his model as a simplification of that of Groves and Thompson (1970). One element of this simplification is the lack of the second process of the dual-process theory. As well as the exponential reduction in response the model also demonstrates spontaneous recovery when the stimulus is withheld. This is caused by presenting an input of 0 whenever the stimulus is not perceived. This behaviour can be avoided by not presenting any input whenever a stimulus is not perceived. The result is that the recovery of Eq. 3 is not seen.

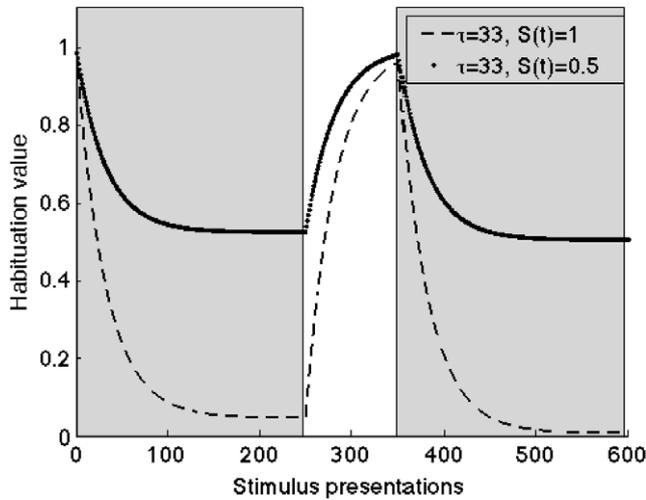
On its own this model does not provide any other of the recognised characteristics of habituation. In particular, there is no stimulus specificity since the stimulus is purely represented by its strength; the use of a machine learning algorithm with the habituation model deals with this, as is discussed in Section 2.2. (In fact, there is evidence in the psychological literature (Lewkowicz (1994, chap. 8) and references therein) that during early development, response and habituation is linked to stimulus strength, not modality.) More seriously for the verisimilitude of the model, it does not demonstrate dishabituation—where presentation of another stimulus results in an increase of the original habituated response—nor show the characteristic that less intense stimulation (given by lower values of  $S(t)$ ) leads to faster habituation, as can be seen in Fig. 3, nor does it demonstrate what is referred to as long-term habituation, which is the fact that repeated training sessions of habituation lead to faster habituation.

This last point was dealt with in a development of the model by Wang and Arbib (1992); see also Wang (1994). They added a coupling to Eq. 1 with a slowly evolving secondary equation that causes slow habituation initially and progressively faster habituation as training progresses. The model is described by two equations:

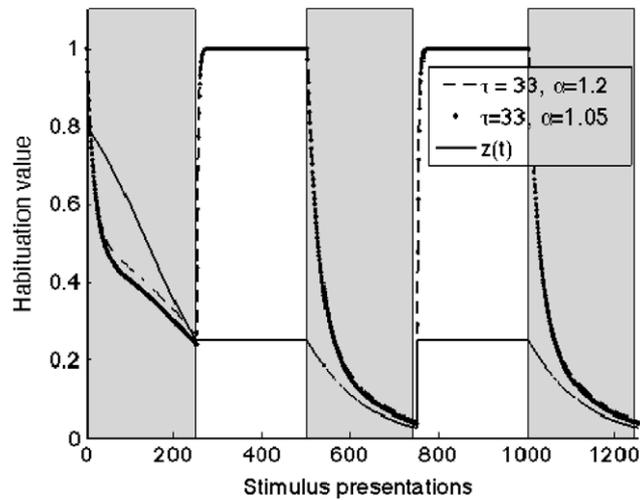
$$\tau \frac{dy(t)}{dt} = \alpha z(t)(y_0 - y(t)) - \beta y(t)S(t), \quad (4)$$

$$\frac{dz(t)}{dt} = \gamma z(t)(z(t) - 1)S(t). \quad (5)$$

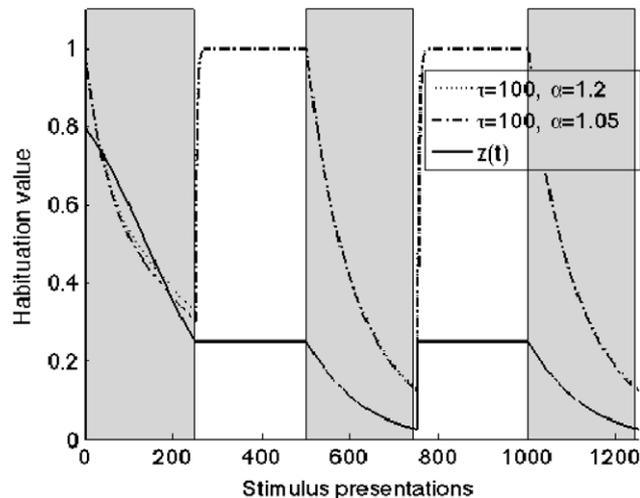
There are two extra parameters in this model:  $\beta$ , which takes account of the contribution of the stimulus to the habituation, and  $\gamma$ , which controls the speed at which  $z(t)$  evolves.  $\beta$  was set to value 1, and  $\gamma$  to 0.001 in the graphs shown in Figs. 4 and 5, where the stimulus is presented for 250 timesteps, then removed for the same number before being reapplied. The speed-up in the



**Fig. 3.** The model of habituation in Eq. 1 does not accurately represent the fact that lower stimulus strengths cause faster habituation. The shaded regions show when a stimulus was presented.



**Fig. 4.** Habituation curves based on Eqs. 4 and 5, with a time constant of  $\tau = 33$  and varying values of recovery parameter  $\alpha$ . The evolution of the relatively slowly evolving function  $z(t)$  is also shown. The shaded regions show when a stimulus was presented.



**Fig. 5.** Habituation curves based on Eqs. 4 and 5, with a time constant of  $\tau = 100$  and varying values of recovery parameter  $\alpha$ . The evolution of the relatively slowly evolving function  $z(t)$  is also shown. The shaded regions show when a stimulus was presented.

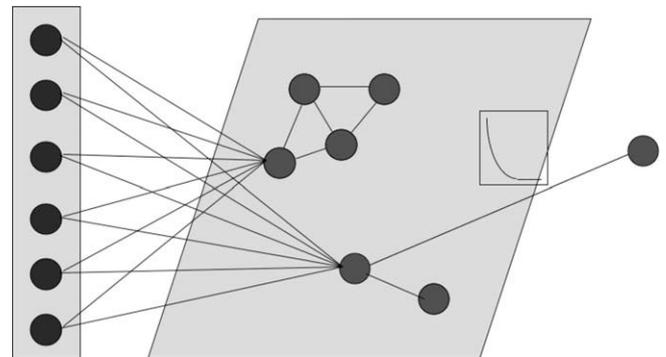
habituation when the stimulus is presented for the second time can be clearly seen, but for these values of the parameters the third habituation process is no faster than the second as  $z(t)$  has already reached the bottom of its dynamics.

This model still does not demonstrate dishabituation, nor several of the other characteristics, such as different results for different frequencies of stimulation. However, Wang and Arbib (1992) use this model and combine it with a functional model of the medial pallium in the toad *bufo bufo* in order to demonstrate the *habituation hierarchy*. This is an interesting effect whereby an animal that has habituated to a stimulus A may also demonstrate habituation to a second stimulus B in some cases, but not others. The complexity of the habituation hierarchy is that for an animal that has habituated to stimulus A, and therefore to stimulus B, the presentation of a dishabituation stimulus may cause the animal to dishabituate to A, but not to B. Thus, an animal is habituated to a stimulus it has never seen, but not to the one that it was trained on.

It is clearly possible to produce more complex models that would reproduce more of the functional characteristics of habituation. Some of them can be done very simply, such as the habituation to specific stimuli that is discussed next. Even dishabituation can be included in this way, either by including a separate and competing sensitisation process (as in Groves & Thompson, 1970) or using a simple check on the strength of the signal. However, for the purposes of this paper, whether or not this is useful depends upon the applications that habituation is put to. Given the focus on making machine learning algorithms that perform useful tasks, rather than understanding the mechanisms of habituation, the creation of more complex models is not considered here. Were applications to be identified where these are useful behaviours (as they are for animals) then they can be added to a model.

### 2.2. Habituating to specific stimuli

It seems clear that the models of habituation described above cannot be expected to provide stimulus specificity given the lack of any form of stimulus description in the model. The habituation process that is described by the models can be applied as a form of learning to any system that can learn to recognise individual stimuli. In the context of machine learning this is generally a neural network or similar algorithm (for an introduction to neural networks see Haykin (1999) or many similar references). Fig. 6 shows a schematic of this. Each element of the neural network recognises different stimuli through its weighted connections with the input vector, and each element has its own habituation counter that is controlled by one of the models described above.



**Fig. 6.** Schematic of combining a model of habituation with a neural network. The output of the neural network is modified by the amount of habituation that has already occurred at that node. When a node is firing often the signal to that node is rapidly habituated, whereas for a node that fires only infrequently the output from the node is strong.

In this way, stimuli that are perceived frequently will be habituated to, whereas those that are perceived only infrequently will not. The types of machine learning algorithm that are suited to this recognition of unlabelled stimuli are typically clustering algorithms. The most famous of these is the Self-Organising Map of Kohonen (1982), although there are others that may be more suited to this task, such as the Growing Neural Gas (GNG) by Fritzke (1997) and the Grow When Required (GWR) network by Marsland, Shapiro, and Nehmzow (2002).

The reason why these latter algorithms are more suited to the task is that the Self-Organising Map is not designed for *on-line* usage. It is intended that all of the data to be presented to the network is collected together and then the algorithm is trained repeatedly on it until the network stabilizes. In this way the algorithm is guaranteed to produce a topologically correct ordering. The other algorithms mentioned above are designed for *on-line* usage, where they learn as data are presented to them and respond immediately to changes in the inputs.

All of these networks work in a similar way. Input vectors are fed into the neural network by having values inserted into the nodes on the left of the figure. These are multiplied by the weighted connections to each of nodes in the map field of the network, producing an activation of that node. The higher the activation the more that node recognises the input. The winning node is selected, and it and its neighbours (those that are connected to it by an edge) are trained by having their weights moved closer to the input.

In this paper, the model of Marsland et al. (2002) is used. It is a self-organising neural network that adapts to recognise inputs that are presented to the network and that has an habituation counter that is used in the learning rule in order to modify the weights more strongly if the node has not been trained much. The algorithm is described qualitatively next. Marsland et al. (2002) describes the algorithm in more detail and presents results of using it for novelty detection and attention focus using experiments based on a robotic platform.

The network has two important components—the nodes, with their associated weight vectors, and the edges that link the nodes to form neighbourhoods of nodes that represent similar perceptions. Both nodes and edges can be created and destroyed during the learning process.

The technique used for creating and destroying network edges is known as competitive Hebbian learning (Martinetz & Schulten, 1991). Inputs are presented to the network and the activation of all nodes is computed as the inner product of the input vector and the weight vector for that node. The node with the highest activation is selected as the winner, and an edge is generated between it and the second-best matching unit. Edge connections have an associated ‘age’, which is originally set to zero, and is incremented at each time step for each edge that is connected to the winning node. The only exception is the edge that links the best-matching and second-best units, whose age is reset to zero. Edges whose age exceeds some maximum age are removed. Any node that has no neighbours, i.e., that has no edge connections, is removed.

New nodes can be added at any time. A new node is added when the activity of the best-matching node is not sufficiently high (defined by a parameter known as the *insertion threshold*). The activity of nodes is calculated using the Euclidean distance between the weights for the node and the input. To allow for the fact that recently created nodes may not yet have been trained to match their intended output correctly, which would mean that the node should be trained more rather than a new node created, each node is equipped with a way of measuring how often the node has fired. This is the role of habituation in the algorithm.

Habituation is also used to ensure that nodes that have fired frequently are trained less, since the learning rate is multiplied by the habituation counter for that unit. This removes a problem that networks that learn continuously often suffer from—the weights of well-trained nodes continue to move slightly, so that the network does not converge. As with most self-organising networks, the setting of the learning rates is usually based on prior experimentation. Finally, it means that the GWR network can be used as a novelty filter without any modification, if the node that fires has not fired before, or fired very infrequently, then the input is novel.

So, when an input is presented to the network, the activity of each node in the map space is calculated and a winner picked. If this best-matching node represents the input well then the activity of that node will be close to one. In that case the best-matching node is trained a little, as are its neighbours. However, if the activity of the network is below the insertion threshold then either the node has only recently been added to the map and is still being trained, or there is a mismatch between the node and the input. If the node is a new one then the node has not habituated to its input, and so the node is trained a little and habituates.

Otherwise, a new node is needed to represent the input better. This node is added between the (badly matched) winning node, which caused the problem, and the input, with the weights of the new node being initialised to be the mean average of the weights for the best-matching node and the input. This method of node generation, and in particular the insertion threshold, can be thought of as tunable generalisation; the amount to which the network generalises between similar perceptions is controlled by the amount of discrepancy between perceptions that triggers a new node.

The insertion threshold is an important parameter. If the value is set very close to 1 then more nodes are produced and the input is represented very well. For lower values fewer nodes are added. Typically values of between 0.8 and 0.95 are used. For the results demonstrated in this paper a value of 0.95 was used.

This combination of habituation model and neural network can be turned to a variety of different applications in machine learning, and these are described next.

### 2.3. Novelty detection

Novelty detection is the recognition that a stimulus differs markedly from those that have been perceived before. It can be a useful survival trait for animals, since the unexpected novel perception could signify a potential predator (or possible victim). By detecting novel features the animal’s attention is focussed first to the most potentially dangerous features of its environment. In this way, novelty detection can be a method of reducing the large amounts of extraneous information that the organism is receiving, so that the animal can concentrate on the most important stimuli. For a discussion of these uses of novelty, see e.g., O’Keefe and Nadel (1978) and Knight (1996). For artificial learning systems this is not generally such a strong motivation for novelty detection. However, it has received much interest as a method of learning about classes that are under-represented in a dataset. For example, consider the problem of identifying anomalies in medical data. It is typical for there to be substantially more negative results than positive, and yet it is the positive ones that are important.

A neural network that is trained on this data will typically learn to identify virtually everything as a negative result, since in general this will lead to high percentage accuracy. One solution to this problem is to train the neural network only on negative examples, but ask it to identify any inputs that do not match its acquired model of negative examples. In this way, any outliers or potential positive results are highlighted, which is novelty detection (Tarassenko, Hayton, Cerneaz, & Brady, 1995). The earliest discussion

of novelty detection in neural networks seems to be the recognition by Kohonen and Oja (1976) that the standard autoencoder neural network can be modified to learn an orthogonal projection of the principal components of the data, which they term the novelty. This has been used in an investigation of habituation in infants by Sirois and Mareschal (2002). For an overview of novelty detection in machine learning, see Marsland (2003).

#### 2.4. Recency detection

Brown and Xiang (1998) suggested that judgement of prior occurrence is an important part of learning, and one that occurred in the perirhinal cortex. They suggested that there are several types of neuron that are instrumental in this task, which they called *recency neurons* that fire strongly for perceptions that have been perceived recently, *familiarity neurons* that give information about the relative familiarity of a stimulus, and *novelty neurons* that respond strongly to presentations of novel stimuli.

The extension of the habituating neural network to this problem of recency detection is a very simple process. A second habituation counter is attached to each node of the neural network. Both of the counters implement one of the two models of habituation described in Section 2.1, but only one of them gets the  $S(t) = 0$  stimulus when that node does not fire (labelled as  $C_1$  in Table 1), the other receives no input at all (labelled as  $C_2$ ) and so its habituation value does not change (it does not spontaneously recover). The first of these two counters also habituates faster, so that only a few presentations of a stimulus are needed for habituation. Table 1 then shows the possible outputs of the two counters together with interpretations of their meanings in the context of Brown and Xiang (1998).

This mechanism enables stimuli that have been seen only recently, and stimuli that have been seen before, but not recently, to be identified, which might have useful applications where the time when presentations are seen is important.

#### 2.5. Temporal learning

The last application of habituation that is presented in this paper is one of temporal learning. The idea here is to consider how learning should be modified when the temporal context of inputs is important. This could be because a signal is spread across many stimuli (such as a bird song or rustling sound in the bushes). These are known as time series data in statistics and machine learning, and problems can be to recognise future trends or to identify patterns that run over several time steps.

Most neural networks are not capable of performing this type of learning without changes, because their current state does not contain information about the previous time step, only about the current weights. The simplest thing to do is to modify the inputs so that as well as the current time step, there are additional inputs that provide the inputs from the previous time step, and possible several other historical ones. This is known as a time-delay neural network and, whilst it can be successful, it adds significantly to the

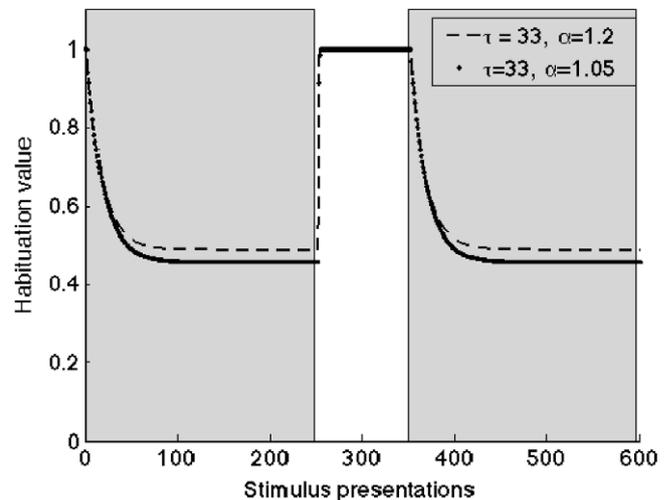


Fig. 7. Habituation curve based on Eq. 6. Note the faster recovery time and higher asymptotic level. The shaded regions show when a stimulus was presented.

complexity of the network and to the amount of data that is required for learning. Another problem with it is that there is no way to differentiate between the input for the current time step and that of the previous one, and the number of historical steps to use has to be decided a priori. An alternative is to use leaky neurons that retain some of their activation from the previous time step, with a known decay function. For a review of these methods, see e.g., Haykin (1999).

It was suggested by Robin, Abbas, and Hun (1990) that habituation can be used to provide a solution to this problem, and Stiles and Ghosh (1997) produced a method of pre-processing the inputs for a neural network based on precisely this idea. They use a difference equation that is derived from Eq. 4 by setting  $z(t) = 1$  for all  $t$ :

$$y(t+1) = y(t) + \frac{1}{\tau}(\alpha(y_0 - y(t)) - y(t)I(t)), \quad (6)$$

where  $I(t)$  is the original input value before any habituation has taken place. This is similar to Eq. 1 except that the input stimulus is multiplied by current habituated value. This changes the dynamics a little, as can be seen in Fig. 7. Stiles and Ghosh (1997) demonstrate that the method allows for more efficient learning of temporal signals than using the time-delay neural network for the task of classifying minke whale song. We have not tested this method further, but report it as an application of habituation in machine learning.

### 3. Results

The first experiment that is presented here is a simple demonstration of using a neural network with habituation as a novelty filter. A training set of data is created that contains Gaussianly distributed random numbers taken from three clusters. The data consists of the two coordinate values of the datapoints shown in Fig. 8, where the ellipses drawn are the 95% confidence intervals on the data.

The neural network of Marsland et al. (2002) was used to learn about the dataset shown in Fig. 8, which resulted in a neural network that has 42 nodes and 48 edges. The positions of the nodes and their neighbourhood connections can be seen in Fig. 9. The network was then tested on data also drawn at random from the same three clusters, but with a fourth added in that differed from the others. In Fig. 10 the points that the network finds to be novel are highlighted with a square. It can be seen that 100% of the points from the new set are found to be novel, as are a set of others that

Table 1

By using two habituation counters,  $C_1$  that spontaneously recovers and  $C_2$  that does not, it is possible to distinguish stimuli that have been seen, but not recently, or those that have only been seen recently, as shown in this table.

State of habituation counter		Interpretation
$C_1$ (recovering)	$C_2$ (no recovery)	
Not habituated	Not habituated	Novel
Not habituated	Habituated	Familiar
Habituated	Not habituated	Recent
Habituated	Habituated	Normal

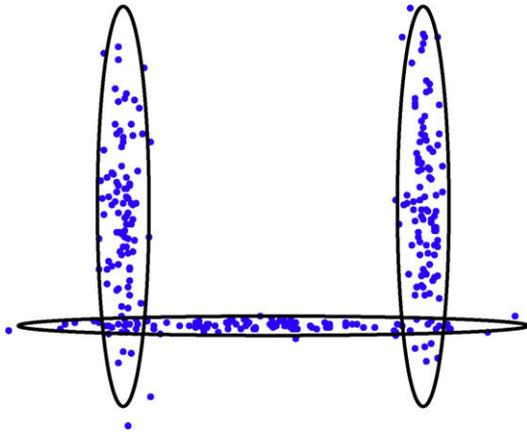


Fig. 8. A set of training data points taken from three Gaussian clusters (with 95% confidence interval drawn as ellipses)

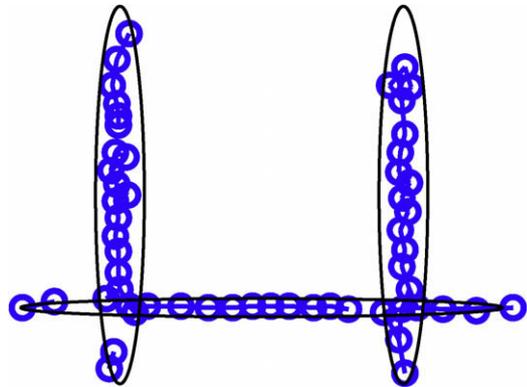


Fig. 9. The neural network trained on the data in Fig. 8. It has 42 nodes and 58 edges.

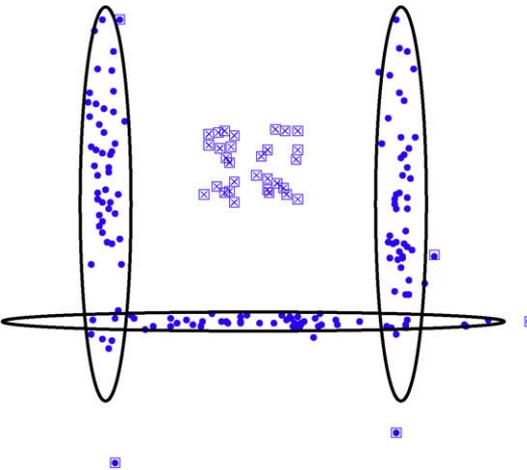


Fig. 10. The test set, which has an extra class added. Points marked with a square were considered to be novel.

fall outside the 95% confidence intervals of the Gaussian clusters. No unexpected points were found to be novel.

The second experiment looks at a more realistic problem in novelty detection. The *biomed* dataset from <http://lib.stat.cmu.edu/datasets> (Cox, Johnson, & Kafadar, 1982) was used. This contains 209 datapoints from a set of four medical tests. Of the 194 complete datapoints 127 are of normals and the remaining 67

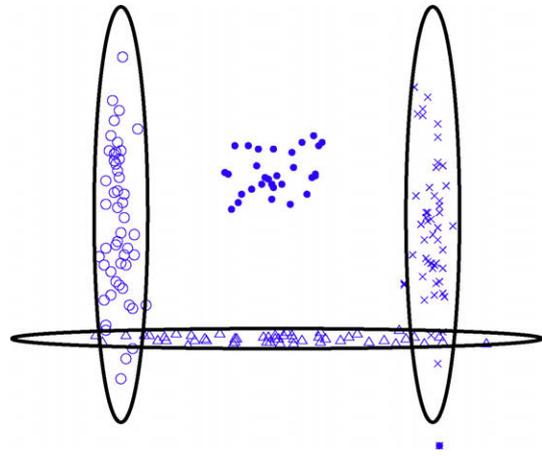


Fig. 11. A recency experiment using the same dataset as in Fig. 8. The left-hand ellipse was trained first, and then not presented again, so that the  $C_1$  habituation counter recovered from this training. Data from the right-hand ellipse was presented throughout training, whilst data from the bottom ellipse was only presented at the end of training. Data from the centre was not seen at all during training, and should therefore be novel. The empty circles marks familiar data, the triangles data seen only recently, the crosses data seen frequently, which the filled circles identify novel data.

demonstrate 1 or more abnormalities, signifying the presence of a genetic disease.

100 of the normal datapoints were used as a training set, and the other 27, together with the abnormal class, were used for testing. The habituation-based novelty detector labelled 56 of the 67 novel inputs correctly, and only 2 of the normal data were misclassified. This is comparable with the best-known methods that have been optimised for this problem.

We also tested the method of recency detection proposed above. This was done using a similar framework to the first experiment, except that now the four classes shown as ellipses were split so that one was novel, one was trained first and therefore largely forgotten by the  $C_1$  counter, and of the other two, one was trained a lot and the other one only a little so that they split into the ‘perceived recently but not learnt’ and the ‘familiar’ group. Fig. 11 shows the results of this experiment, and it can be seen that the results are as expected. However, getting this result required some tweaking of the time constants for the two habituation counters, making the procedure rather difficult to use in practice without some more experiments.

#### 4. Discussion

This paper has demonstrated the application of habituation to machine learning. By taking a simple model of habituation as an exponentially decaying function (with dishabituation being the reverse process) it is possible to use standard neural networks as novelty detectors, an application that has many uses in machine learning.

The aim of the model needs to be considered. This purpose here has been to demonstrate that habituation provides a useful mechanism for learning, not to imply that the methods here have any psychological or biological relevance. It is common for machine learning to take models that are inspired by biology and apply them to real world problems. However, the model is not intended to imply any underlying biological reality, nor does it model all (or even most) of the expected characteristics of habituation.

So to what extent is this actually a model of habituation? This is a question that can be debated from different viewpoints. In this

paper, we have considered it a model of habituation if it demonstrates the primary effect, i.e., progressive decrease in response to a repeatedly presented stimulus. However, this is very simplistic and whilst it can provide useful application benefits, it does not contribute to an understanding of what habituation is, nor why it is so ubiquitous. If there is any benefit to the psychological literature it is the demonstration that the phenomenon, however implemented, does show the expected gross behaviours.

Two different simple models of habituation from the literature are considered here. Whilst the second is considerably more complex than the first, it still does not provide a complete model in that it does not demonstrate even the majority of the characteristics of habituation described elsewhere in this issue. The models are, however, sufficient for their primary purpose of providing a computational description of the primary effects of habituation, namely the decrement in response to a stimulus that is perceived repeatedly without ill effect.

An essential part of the applications of habituation that have been demonstrated in this paper is a neural network. This is an associative learner, which is used to recognise the particular stimulus being presented. For machine learning, the habituation process alone is not sufficient for useful behaviour, it is only when it is coupled with the associative (Hebbian) learner that applications can be found. This seems at odds with the description of habituation as purely non-associative learning.

The paper has discussed three different applications to which habituation can be turned in the context of machine learning. In the primary one, that of novelty detection, the combination of habituation with a standard unsupervised neural network leads to an algorithm that performs as well as any other on standard datasets that are used to demonstrate the application. This is a promising result, and one that has led to it being used for a variety of other tasks in the area. It demonstrates that habituation is a useful way to perform novelty detection, as may well be the case in animals as well.

## Acknowledgments

The original motivation for some of this work was the author's Ph.D. Thesis at the University of Manchester under the supervision of Jonathan Shapiro and Ulrich Nehmzow. Their support and encouragement is gratefully acknowledged. Thanks also to the referees, who made some very useful suggestions for improvement.

## References

Alonso, L., Moreno, R., Vázquez, M., & Santacreu, J. (2005). Simulation of the filtering role of habituation to stimuli. *The Spanish Journal of Psychology*, 8(2), 134–141.

- Brown, M., & Xiang, J.-Z. (1998). Recognition memory: Neuronal substrates of the judgement of prior occurrence. *Progress in Neurobiology*, 55, 149–189.
- Cox, L., Johnson, M., & Kafadar, K. (1982). Exposition of statistical graphics technology. *ASA proceedings of the statistical computation section*, 3, 55–56.
- Fritzke, B. (1997). A self-organizing network that can follow non-stationary distributions. In *Proceedings of the international conference on artificial neural networks (ICANN'97)* (pp. 613–618). Berlin: Springer.
- Groves, P., & Thompson, R. (1970). Habituation: A dual-process theory. *Psychological Review*, 77, 419–450.
- Haykin, S. (1999). *Neural networks: A comprehensive foundation* (2nd ed.). New Jersey, USA: Prentice-Hall.
- Knight, R. (1996). Contribution of human hippocampal region to novelty detection. *Nature*, 383, 256–259.
- Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43, 59–69.
- Kohonen, T., & Oja, E. (1976). Fast adaptive formation of orthogonalizing filters and associative memory in recurrent networks on neuron-like elements. *Biological Cybernetics*, 25, 85–95.
- Lewkowicz, D. (1994). *Development of intersensory perception in human infants*. Lawrence Erlbaum, pp. 165–204.
- Marsland, S. (2003). Novelty detection in learning systems. *Neural Computing Surveys*, 3, 157–195.
- Marsland, S., Shapiro, J., & Nehmzow, U. (2002). A self-organising network that grows when required. *Neural Networks*, 15(8–9), 1041–1058.
- Martinetz, T. M., & Schulten, K. J. (1991). A “neural-gas” network learns topologies. In T. Kohonen, K. Mäkisara, O. Simula, & J. Kangas (Eds.), *Artificial neural networks* (pp. 397–402). Amsterdam: Elsevier.
- McCulloch, W., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 7, 115.
- O'Keefe, J., & Nadel, L. (1978). *The Hippocampus as a cognitive map*. Oxford: Oxford University Press.
- Rankin, et al. (2008). Habituation revisited: An updated and revised description of the behavioral characteristics of habituation. *This volume*.
- Robin, D., Abbas, P., & Hun, L. (1990). Neural responses to auditory patterns. *Journal of the Acoustic Society of America*, 87(4), 1673–1682.
- Rumelhart, D., McClelland, J., and the PDP Research Group (1986). *Parallel distributed processing: Explorations in the microstructure of cognition*, Vol. 1: Foundations. Cambridge, MA: MIT Press.
- Sirois, S., & Mareschal, D. (2002). Models of habituation in infancy. *Trends in Cognitive Sciences*, 6(7), 293.
- Sirois, S., & Mareschal, D. (2004). An interacting systems model of infant habituation. *Journal of Cognitive Neuroscience*, 16(8), 1352–1362.
- Stanley, J. (1976). Computer simulation of a model of habituation. *Nature*, 261, 146–148.
- Stiles, B., & Ghosh, J. (1997). A habituation based neural network for spatio-temporal classification. *Neurocomputing*, 15(3), 273–307.
- Tarassenko, L., Hayton, P., Cerneaz, N., Brady, M. (1995). Novelty detection for the identification of masses in mammograms. In *Proceedings of the fourth IEE international conference on artificial neural networks (ICANN'95)*, pp. 442–447.
- Thomas, H., & Gilmore, R. O. (2004). Habituation assessment in infancy. *Psychological Methods*, 9(1), 70–92.
- Thompson, R., & Spencer, W. (1966). Habituation: A model phenomenon for the study of neuronal substrates of behaviour. *Psychological Review*, 73, 16–43.
- Wang, D. (1994). A neural model of synaptic plasticity underlying short-term and long-term habituation. *Adaptive Behavior*, 2, 111–129.
- Wang, D., & Arbib, M. (1992). Modelling the dishabituation hierarchy. The role of the primordial hippocampus. *Biological Cybernetics*, 67, 535–544.
- Zeaman, D. (1976). The unicity of novelty–familiarity (habituation?) effects. In T. Tighe, & R. Leaton (Eds.), *Habituation: Perspectives from child development, animal behavior, and neurophysiology*. Hillsdale, New Jersey: Lawrence Erlbaum Associates.