The Integration of Diffusely-Distributed Salience Signals Into A Neural Network

Leendert Remmelzwaal, Jonathan Tapson and George F. R. Ellis

University of Cape Town, South Africa

+27833016991

+27216503465

jtapson@gmail.com

Article note:

Abstract:

We describe an artificial neural network which is designed to model the interaction between the human cortex's local interconnections and the ascending pathways from the limbic system. The model consists of a conventional multilayer perceptron (MLP), modelling the cortex, to which has been added a network carrying a diffusely-distributed ascending salience signal. We show that the local network's learning rate is responsive to salience inputs, so that it learns faster, and shows one-time learning, when input events have high associated salience. We show also that the network learns to infer high salience for inputs similar to others which have previously been presented with high salience. This suggests that a simple mechanism may underly the interaction between the brain's diffuse value systems, and the local processing in the cortex.

Keywords:

Abbreviations:

Introduction

The human brain has at least two broad types of connection within the cortex, besides the motor and sensory systems (Nolte, 2008). The first type consists of localized inter-neuronal connections forming layered neural networks. The second class of connections relating to the cortex consists of diffuse projections from the limbic system to the cortex (Nelson & Leffman, 1963) and are known as monoamine systems (Andén *et al.*, 1966), or ascending systems (Edelman & Tononi, 2000). From their nuclei of origin they send axons up and down the nervous system in a diffused spreading pattern. The effect of the monoamine systems projecting profusely is that each associated neurotransmitter (for example norepinephrine and dopamine) affects large populations of neurons, allowing

non-local interactions to occur in the brain. The release of the neurotransmitter modulates the probability that neurons in the neighbourhood of value-system axons will fire after receiving glutamatergic input (Panksepp, 2004), thus they are an important mechanism effecting neural plasticity. These systems are thought to bias neuronal responses affecting both learning and memory by guiding neuronal group selection, and for this reason that they are sometimes termed value systems (Edelman & Tononi, 2000).

In this paper we link the ascending systems with the concept of salience. The salience of an entity refers to its state or quality of standing out, or receiving increased attention, relative to entities that are related either spatially (i.e. through similar visual objects) or temporally (i.e. in terms of related concepts). For example, a salient memory would be one that significantly stands out among others. Beebe and Lachmann (1994) investigated infant development, and classified three core principles of salience that describe the interaction structures of the cortex, namely the principles of *ongoing regulations*, *disruption and repair*, and *heightened affective moments*. We focus on ongoing regulations and heightened affective moments. Ongoing regulation describes a characteristic pattern of repeated interactions, such as a child interacting with their mother or father, resulting in increased salience towards certain associations and memories. A heightened affective moment, on the other hand, describes a dramatic moment standing out among other events, often leading to one-time learning, and an instantaneous attachment of salience to certain associations.

We have attempted to address the question: by what mechanism do the ascending systems modulate the local processes in the cortex, in response to salient events?

We have used an artificial neural network (ANN) as a simplified model of localized inter-neuronal connections in the cortex. These connections allow complex processes such as pattern recognition (Bishop, 1999) and naming to occur (Hawkins & Blakeslee, 2005). However, simplified ANN models fall short in representing non-local interactions, including diffuse projections from the limbic system to the cortex. Although standard ANNs allow associational learning with multiple-trial (or what Beebe and Lachmann would term ongoing regulations: Beebe & Lachmann, 1994) they do not provide the capacity for *one-time learning* (Rock & Heimer, 1959), which has been the subject of discussion in the psychology literature for over 50 years. While the details of how single-

trial learning works in the cortex are in dispute, the fact that single-trial learning sometimes occurs is well established by researchers such as Armstrong (Armstrong *et al*., 2006) and Lee-Teng (Lee-Teng & Sherman, 1966). Waelti and Berridge (Waelti *et al*., 2001; Berridge & Robinson, 1998), among others, have reported that single-trial learning is often associated with the neurotransmitter dopamine. Dopamine is one of the neurotransmitters that is released throughout the cortex by the ascending systems, and is associated both with the neural coding of basic rewards, and in reinforcement signals defining an agent's goals (Montague *et al*., 2004). It therefore plays an important role in brain functioning (Downara *et al*., 2003).

It is clear then that non-local flows of information represented by the ascending systems (i.e. the salience effects of neurotransmitters such as dopamine) are a significant feature of the structure of the cortex. In this paper, we model this feature by modifying the structure and training of a standard type of ANN. Most ANNs are built on the fundamental assumption that all neurons are identical and behave similarly; neural processing is achieved only by varying the weights of the connections between neurons. However, the release of a neurotransmitter affects the probability that an individual neuron will fire, and therefore to model the effect of non-local flows of information represented by the ascending systems, we propose that neurons cannot be treated identically. To capture this effect, the standard model of a neuron in an ANN will be modified. Altering the probability that neurons in the neighbourhood of value-system axons will fire requires modifications to the neuronal firing threshold, rather than to the weighting of the signals received by a neuron. We postulate that adjusting the threshold of a node in an ANN is a simplified mathematical model of what happens when neurotransmitters bias neuronal responses. This underlies the design of what we have termed a *salience-affected artificial neural network* (SANN).

In the following sections of this paper we describe the following:

1. A standard ANN is modified by adding a diffuse network which provides a salience signal of equal magnitude to all neurons. A similar diffuse network is used to measure the evoked output salience of the ANN, by suitably summing the evoked output signal from all neurons.
2. The neurons are modified by the salience signal so that their likelihood of firing is increased, if they are already active at the time of reception of a high salience

signal. The neuron is unaffected by salience if it is inactive at that time. Hence the pattern of neural activation at the instant when the salience signal arrives is recorded through this process.

3. The neural network is then trained and tested on a conventional task (pattern recognition), with some patterns being defined to have high salience and therefore to be accompanied by a strong diffuse input salience signal.

4. The effect of input salience on the speed and accuracy of the ANN's learning of the pattern recognition task is measured, and the evoked salience output in response to patterns associated with high salience in initial presentation is also measured.

SANN Implementation

Forward Propagation of the Signals

In carrying out this study we chose to modify a generic fully-connected multilayer perceptron (MLP) ANN which used a standard back-propagation algorithm during training[1], as it is a well-characterized model from which many other neural networks are derived. The ANN was designed with a single hidden layer for simplicity, as we only intended to demonstrate the concept of salience in this research. The neural network was modified in a way that each node had a threshold variable that could be externally modified by an *input salience signal* (S). We use the same notation as Bishop (Bishop, 1999) for clarity. For a conventional three-layer multilayer perceptron (MLP) such as forms the basis of the SANN, with d input neurons x_i, M hidden layer neurons with outputs z_j, and n output layer neurons with outputs y_k, the outputs of the network can be expressed as:

$$y_k = g\left(\sum_{j=1}^{M} w_{kj}^{(2)} g\left(\sum_{j=1}^{d} w_{ji}^{(1)} x_i \right) \right)$$

Equation 1

> **Comment [T1]:** On right: i=1

where the superscripts refer to the layer; the w terms are the interconnecting weights indexed by the subscripts i, j and k; and g is the nonlinear output function for the

---

[1] The ANN section of the source code designed in this research was originally written in Python and placed in the public domain by Neil Schemenauer (nas@arctrix.com). For this research, the source code was adapted from Python into Matlab code.

neuron. The addition of the salience network has the effect of adding a unique offset threshold term *T* in the nonlinear function for each neuron so that:

$$y_k = g\left(\sum_{j=1}^{M} w_{kj}^{(2)} g\left(\sum_{j=1}^{d} w_{ji}^{(1)} x_i + T_j\right) + T_k\right)$$

Equation 2

In the SANN we approximate the sigmoidal function with the the *tanh* function:

$$g(a, T) = \tanh(a + T)$$

Equation 3

where *a* represents the sum of weighted inputs $\sum wx$ for the neuron.

Backpropagation of Error During Learning

What happens when we back-propagate the output error during training? We use the conventional notation where the network's total output error is denoted E^n, the sum of weights at any neuron is a_j or a_k depending on layer, and the individual neuron errors are δ_j or δ_k. Individual errors may be defined in different ways, but we use the standard definition:

$$\delta_j \equiv \frac{\delta E^n}{\delta a_j}$$

Equation 4

If we use the standard sum-of-squares expression for the total output error:

$$E^n = \frac{1}{2}\sum_{k=1}^{n}(y_k - t_k)^2$$

Equation 5

where t_k are the target values for the output, then the output neuron errors are:

$$\delta_k = y_k - t_k$$

Equation 6

The hidden layer errors are defined by the backpropagation formula:

$$\delta_j = g'(a_j) w_{kj} \delta_k$$

Equation 7

which for the SANN with a *tanh* activation is:

$$\delta_j = \text{sech}^2(a_j + T_j) w_{kj}(y_k - t_k)$$

Equation 8

5

The backpropagation algorithm will include a specified learning rate η, so that the weight updates for the different layers will be:

$$\Delta w_{ji} = -\eta \delta_j x_i \qquad \text{\textbf{Equation 9}}$$

and

$$\Delta w_{kj} = -\eta \delta_k z_j \qquad \text{\textbf{Equation 10}}$$

We can see from this that the salience-modulated variable T_j has no effect on the updating of the weights w_{kj}, although it may have contributed to the error in forward propagation. The variables T_j do affect the updating of weights w_{ji} through δ_j as seen in **Equation 9** and **Equation 10**. If we plot Δw_{ji} as a function of the neuron's current input and threshold ($a_j + T_j$), for arbitrary values of input signal and output error, as shown in the figure below, we can see that the learning rate is highest when the activation is close to the threshold.
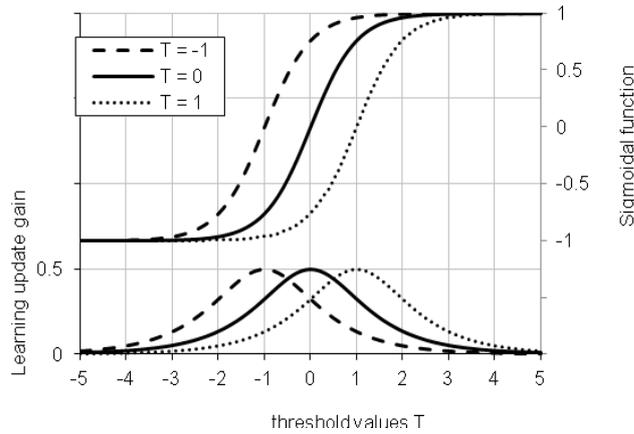

Fig. 1 The neuron's sigmoidal activation functions (top) and learning update rates (bottom) for three values of threshold T. It can be seen that a salience-driven change in T modifies the operating point of the function; however, the point of maximum learning tracks this operating point. The input salience signal applied to each node has the effect of pushing active neuron's thresholds closer to their activation signal for positive salience, or moving them further away from this signal for negative salience. The learning update gain (in arbitrary units) is highest when the

neuron's activation level is closest to the neutral threshold $g(a + T) = 0$, as is the case in conventional backpropagation

This comes as no surprise, as the backpropagation algorithm is designed to provide learning in such a way as to push the network towards a decision, i.e. away from this neutral position. We can conclude from this analysis that the backpropagation algorithm operates well in spite of the non-zero threshold, because the use (in the algorithm) of the derivative of the activation function centers the varying learning rate at the appropriate operating point.

Backpropagation of Salience through the Network

After conventional backpropagation, the threshold is updated as a function of the input salience according to the rule:

$$\Delta T_j = S\beta z_j \qquad \textbf{Equation 11}$$

$$\Delta T_k = S\beta y_k \qquad \textbf{Equation 12}$$

where S is the input salience value and β is the salience learning rate. This update occurs after the backpropagation of error. The effect is to increase the activation of the neuron (for positive β). This is not strictly backpropagation, in the sense that the salience or error or activation from the output layer is not backpropagated to the hidden layer; the threshold is updated based only on individual neurons' activations and the global salience. Hence it records the pattern of neural activation at the instant when the salience signal is received. A threshold limit (T_{lim}) was introduced to limit the range of variation permitted for the neuron's threshold. The new threshold of a node was determined by:

$$T_{new} = \min(T_{\lim}, \max(-T_{\lim}, T_{old} + \Delta T)) \qquad \textbf{Equation 13}$$

where T_{old} is the previous threshold value. Each neuron in the SANN had the option of being directly affected by the external input salience signal (S) during training. During testing the threshold variables of each node were collected, creating what we defined as

a *reverse salience signal* (S'). The reverse salience signal (S') was defined as the sum of the reverse salience signal *(S')* produced by each node, as determined by:

$$S'_i = \sum_{k=1}^{d} T_k y_k{}_i + \sum_{j=1}^{M} T_j z_j \qquad \textbf{Equation 14}$$

The reverse salience signal *(S')* for each node was calculated as the product of the present neuron activation level (y_k or z_j) and the present threshold value (T_k or T_j). The change in threshold affects the sigmoidal functions as shown in **Error! Reference source not found.**

Comment [T4]: correct

We can reformulate the salience-altered threshold as an extra input term, here θ added into each neuron with a specific weight, in the manner of a fixed bias term. The output for the network would then be:

$$y_k = g\left(\sum_{j=1}^{M} w_{kj}{}^{(2)} g\left(\sum_{j=1}^{d} w_{ji}{}^{(1)} x_i + w_{js}\theta \right) + w_{ks}\theta \right) \qquad \textbf{Equation 15}$$

where w_{xs} is the weight connecting the salience input with neuron x. This would give update rules:

$$\Delta w_{j\theta} = -S\beta z_j \qquad \textbf{Equation 16}$$

$$\Delta w_{k\theta} = -S\beta y_k \qquad \textbf{Equation 17}$$

Note that whilst these are superficially similar to **Equation 9** and **Equation 10** the error is not propagated between layers, and the nonlinear activation function plays no part. The "error" in this case is the distance from the neutral threshold.

This representation of the salience input, as a mechanism which creates a bias term, offers an explanation of why the method works. Bias terms can be considered as defining the position of a neutral hyperplane in the input space so that all points on this plane give a neuron output of zero. The hyperplane corresponds to:

8

$$\sum w_{id} x_i = -T_d \qquad \textbf{Equation 17}$$

for neuron d with inputs from x_i.

If we consider each neuron to be projecting points in this i-dimensional input space on to a single output dimension, and the bias terms to be creating a neutral or separating hyperplane in the input space (defined by equation 17), then a salience-induced change in threshold is effectively a shift in the position of that hyperplane, in the direction which increases the distance of the salient point from the hyperplane. This increases the activation resulting from inputs with that salience, as is desired.

A schematic overview of the *input salience signals* and *reverse salience signals* interconnections in the SANN model are illustrated in **Fig.**
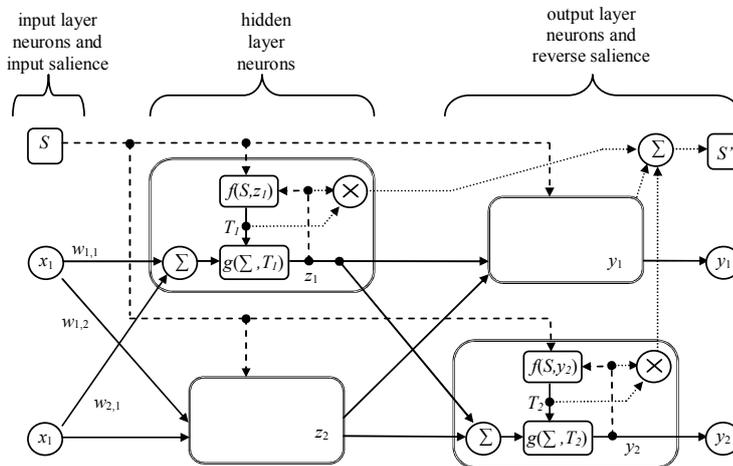
9
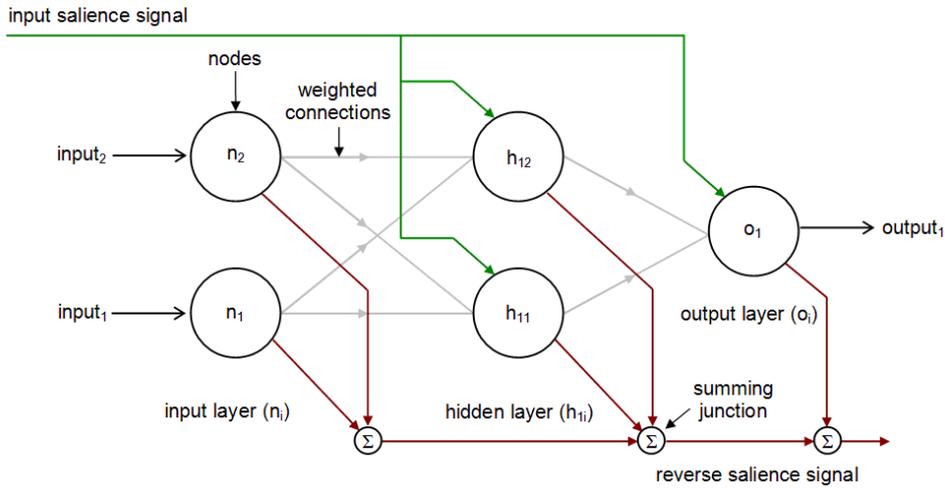
OR

Fig. 2 An SANN with an input layer, one hidden layer, and an output layer. Conventional MLP neural pathways are shown with solid arrows, salience pathways with dashed arrows, and reverse salience with dotted arrows. It can be seen that the salience input S is the same for all neurons, and is combined with the local activation (z or y) to alter the local threshold T. The T values and the activations are combined to give the reverse salience output S'. Function $g()$ is the activation function as per equation (*2*), and $f()$ is the threshold update function as per equations (*12*) and (*13*). The conventional MLP connection weights are given by $w_{i,j}$ and are labeled only in the first layer, to avoid visual clutter. Similarly, only one neuron in each layer has been shown in full, and the reverse salience signal from z_2 has been omitted

SANN Training and Testing

To train and test the SANN we required a simple experiment on a predictably distributed, semantically neutral data set to demonstrate the functionality of the SANN. We generated an 8-bit Boolean sequence, mapped to a normalized output (Table 1). The SANN was trained with an exhaustive input data set of 2^8 (256) elements, mapped to an output described by:

$$sum = (x_{i1} + x_{i3} + x_{i5} + x_{i7}) - (x_{i2} + x_{i4} + x_{i6} + x_{i8}) \qquad \textbf{Equation 17}$$

where x_i was the input data element, and x_{ij} refers to the jth bit of the data element. The sum was then mapped into an output layer of 8 nodes. These 8 bits reflected the sum, as described by:

Sum	Output Layer							
	x_{i1}	x_{i2}	x_{i3}	x_{i4}	x_{i5}	x_{i6}	x_{i7}	x_{i8}
-4	1	0	0	0	0	0	0	0
-3	0	1	0	0	0	0	0	0
-2	0	0	1	0	0	0	0	0
-1	0	0	0	1	0	0	0	0
0	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0
2	0	0	0	0	0	0	1	0
3	0	0	0	0	0	0	0	1

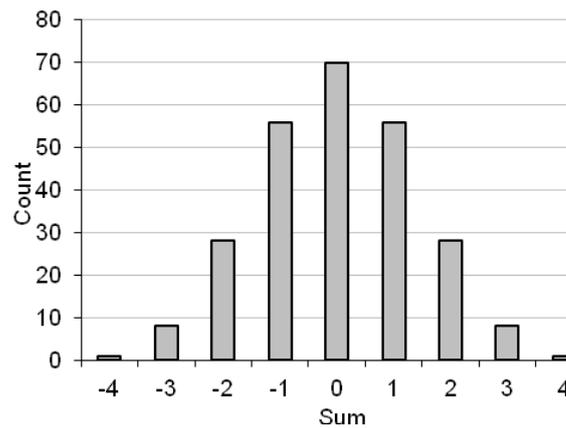**Table 1** Mapping of the output layer elements based on the input data elements

Fig. 3 Normal distribution of the sum values of the 8-bit data set

11

We designed an SANN with 8 input bits, a hidden layer of 6 nodes, and an output layer of 8 nodes. The SANN underwent two tests, namely a multiple-trial and single trial learning algorithm. In both tests, the momentum factor was set to 0.1, the learning rate set to 0.15 and the number of iterations was 100.

Multiple-trial learning algorithm

The first test was a multiple-trial learning simulation, where the performance of the SANN was evaluated under salience-affected and salience-free simulations. The SANN was first trained with an input salience signal of value 0 applied to all input combinations, and then a second SANN was trained with an input salience signal of value 1 attached only to a single data element. This data element was arbitrarily chosen to be {0,1,0,1,0,0,1,0}. We will refer to this as the *salience-affected element*.

The results indicated that after salience-free training of the SANN, the reverse salience signal was also salience-free from all input data elements (**Fig. **). During salience-affected training, the SANN associated salience values similar to the salience-affected element to the data elements most similar to the salience-affected element, as determined by their Hamming distance from the salience-affected element. This confirms that our hypothesis was correct, and that training with a few images will result in images with similar features returning similar reverse salience signal values. This result also supports the principle of ongoing regulation suggested by Beebe and Lachmann (1994).
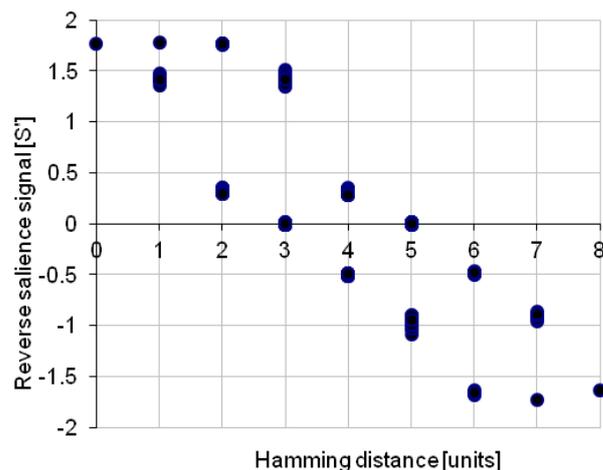
Fig. 4 Reverse salience signal vs. Hamming distance from the salience-affected data element, for multiple-trial learning.

To successfully simulate the value system proposed by Edelman and Beebe, it was desired that the threshold variation has minimal impact on the performance and output of the neural network in terms of pattern recognition. The testing demonstrated that although the SANN had a final cumulative error slightly higher than an SANN trained salience-free, it is evident from **Fig.** that errors fluctuate during training, and therefore we can deduce that the difference in error of a salience-affected SANN performs similarly to a salience-free SANN.
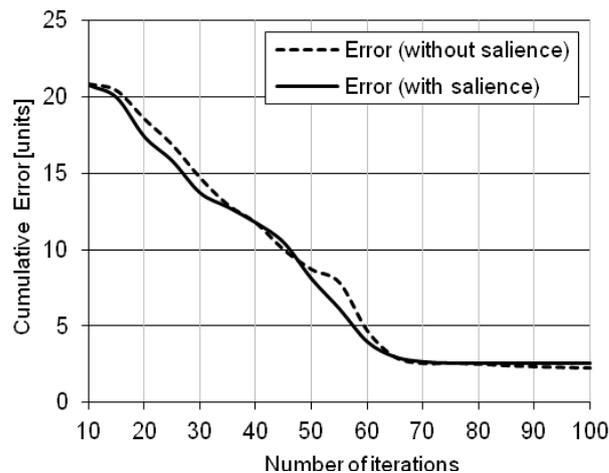
Fig. 5 Typical performance of an SANN trained with and without salience, over the period of 10 – 100 iterations

This result suggests that the input salience signal has a negligible effect on the performance of a SANN during training. This can be extended to describe the input salience signal of an SANN as independent of, or conceptually orthogonal to the SANN output signal (i.e. with negligible variations).

Single-trial learning algorithm

The second test was designed to apply a single learning iteration with salience, after the neural network had been trained salience-free. The SANN was first trained with an input salience signal of $S = 0$ applied to all input combinations for 99 iterations, and then the

13

SANN was trained with a single iteration whereby an input salience signal of $S = 1$ was attached to the salience-affected element (chosen above).

The results indicated that after single-trail learning with the SANN results in a salience profile similar to the multiple-trial salience profile (**Fig.**). When compared the profiles produced a correlation coefficient of $R^2 = 0.9901$, indicating that the profiles are remarkably similar. This result supports the principle of one-time learning suggested by Beebe and Lachmann (1994). The fundamental difference between the salience profiles is that the single-trial learning produced a salience magnitude of 9% of the salience of the multiple-trial example. This magnitude could be increased by introducing a salience-amplification factor, or repeated training iterations with salience.
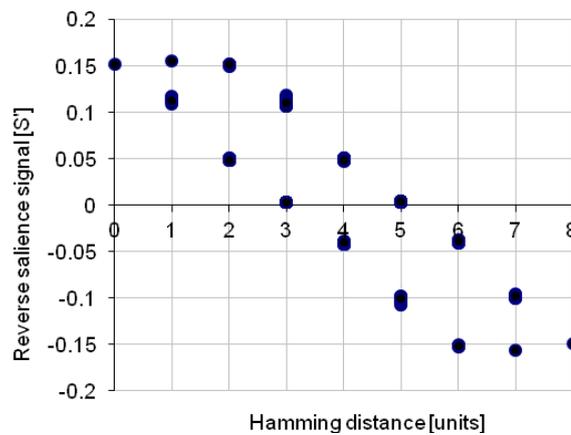
Fig. 6 Reverse salience signal vs. Hamming distance from the salience-affected data element, for single-trial learning

Face recognition learning algorithm

To demonstrate the performance of an SANN in a real-world application, we designed a facial recognition application which incorporated the use of salience signals. The application trained and tested an SANN on 200 images of faces[2], and allowed the user to adjust an input salience signal for each image during the training phase, and calculated

---

[2] Image data was provided by the CBCL face image database (http://cbcl.mit.edu/cbcl/).

the reverse salience signal during the testing phase[3]. The facial recognition application used a non-negative matrix factorization (NMF) (Lee & Seung, 1999 and 2001) parts-based feature-extraction algorithm to break the images down into 49 independent inputs for the SANN. There is psychological and physiological evidence for parts-based representations occurring in the brain. Parts-based representations emerge by virtue of two properties; firstly the firing rates of neurons are never negative, and secondly the synaptic strengths do not change sign (Lee & Seung, 1999). The SANN was designed with 49 inputs, a single hidden layer of 10 units and a single output layer. The output of the neural network during training was chosen as the average pixel grayscale value for each image. The output was not chosen to be random values because verifying the results required the output to be somehow related to the faces in the dataset. The neural network was trained with 200 iterations in all tests. Five similar faces were identified to observe the effect of salience[4]. These faces were all similar as they were photos of the same person. Of the five faces, three were trained with an input salience signal of value 1, while the others were trained with an input salience signal of value 0.

The SANN was subjected to a multiple-trial learning simulation, where the performance of the SANN was evaluated under salience-affected and salience-free simulations. The SANN was first trained with an input salience signal of value 0 applied to all input combinations, and then a second SANN was trained with an input salience signal of value 1 attached only to images 9, 10 and 11. From the results of the experiment (Fig 7) we observed that images 2 and 3 returned a significantly high reverse salience signal value, despite being trained with an input salience signal of value 0. This confirms that our hypothesis was correct, and that training with a few images will result in images with similar features returning similar reverse salience signal values.

---

[3] The NMF section of the source code was adapted from an existing NMF algorithm implementation: Hoyer, P.O. August (2006). *NMF Pack*. Helsinki, Finland. (http://www.cs.helsinki.fi/)
[4] From the dataset images 9, 10 and 11 were trained with an input salience signal of value 1, while images 2 and 3 were trained with an input salience signal of value 0.
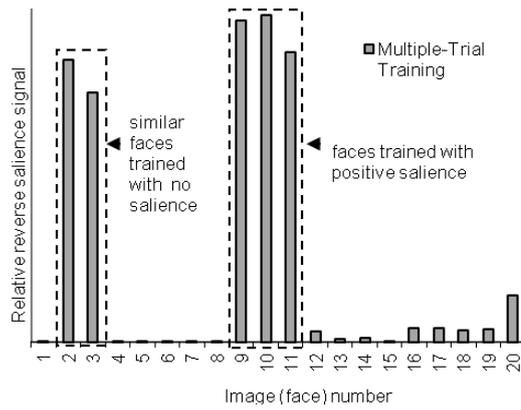
Fig. 7 Reverse salience signals after training the SANN with salience. Images 9, 10 and 11 were trained with an input salience signal of value 1, while the remaining images were trained with an input salience signal of value 0. Noticeably, images 2 and 3 returned a relatively high reverse salience signal value, despite being trained with an input salience signal of value 0

Implications

The research described in this paper has implications at the computational, biological and psychological levels. At a computation level, the modeling of ongoing regulations in neural networks will allow us to extract more information (in the form of salience) from a standard ANN, without significantly adding to the complexity of the ANN structure. Furthermore, exploring the effect of heightened affective moments on a neural network will enable the future training of neural networks with a single training iteration.

At the biological level, at the local level, an SANN enables modeling of the effects of the peculiar nature of synaptic connections as opposed to gap-junction connections. In synaptic connections the electric inter-neural signal is converted to a chemical signal that crosses the synaptic gap, and then gets converted back to an electrical signal. With the simplified model of gap-junction connections, the signal propagates as the direct transmission of an electric signal. The key point is that a synaptic connection allows non-local modulation of local synaptic processes via diffuse projection (Edelman, 2004) of neurotransmitters to the synapse region, while the simplified model of gap-junction connections does not allow such effects. It is this non-local modulation of local synaptic processes that can be modeled by an SANN.

16

At a global level, the model predicts that in addition to the salience signals transmitted by the ascending systems, there should be reverse salience signals from the cortex to the limbic system. In other words, the ascending systems are also descending systems. This is in accord with the principle of reciprocal connection that Edelman emphasizes (Edelman and Tononi 2000). It is also in accord with the well known psychological phenomenon that cortical activation, such as face recognition, can activate emotional reponses (Damasion 1995). Presumably this descending effect can be verified by physiological investigation.

At a behavioural level, an SANN allows the modeling of the effect of affective states on brain activity and on memory (for example, adding an emotional tag to significant memories), because the ascending systems originate in the limbic system which is the seat of affective states. Thus a SANN potentially represents the effects of emotions on cortical activity, which are known to be significant (Damásio, 1995). Thus a SANN models the two-way interaction between emotional centres and the cortex.

Discussion

The SANN designed in this paper has simulated the fundamental features and behaviour that we associate with salience and learning. When presented with input data with a high associated salience, the combined system not only learns rapidly to produce a strong output salience response, but also produces this response for similar data elements. Despite slight fluctuations in performance during training, this salience processing effectively operates independently to the pattern recognition processing performed by the locally-connected network of a standard ANN.

When trained with single-trials, an SANN will produce similar reverse salience signal profiles to the multiple-trial training response. This indicates that an SANN can be trained with a single-trial training iteration, with similar effects to multiple-trial training, which is not a feature of standard ANNs. The salience profiles attained in single-trial training could closer resemble the salience profiles from multiple trial training, given that either a salience-amplification factor is introduced or if multiple training iterations with salience are permitted. We associate this result with the expectation that single events of very high salience must produce a learned response; to the best of our

17

knowledge, this is the first demonstration of this phenomenon in an artificial neural network. There are both ascending and descending effects which should me mirrored in neural structure in the brain.

References

Armstrong, C. M., DeVito, L. M., & Clelan, T. A. (2006). One-Trial Associative Odor Learning in Neonatal Mice. *Chem. Senses* , 31:343-349.

Beebe, B., & Lachmann, F. M. (1994). Representation and Internalization in Infancy: Three Principles of Salience. *Psychoanalytic Psychology* , 11:127-165.

Berridge, K. C., & Robinson, T. E. (1998). What is the role of dopamine in reward: hedonic impact, reward learning, or incentive salience? *Brain Research Reviews* , 28(3):309-369.

Bishop, C. M. (1999). *Neural Networks for Pattern Recognition.* Oxford: Oxford University Press.

Damásio, A. (1995). *Descartes' error: Emotion, reason, and the human brain.* New York: Avon Books.

Downara, J., Mikulisa, D. J., & Davisa, K. D. (2003). Neural correlates of the pro-longed salience of painful stimulation. *NeuroImage* , 20(3):1540-1551.

Edelman, G. M. (2004). *Wider than the sky: A revolutionary view of consciousness.* New York: Yale University Press.

Edelman, G. M., & Tononi, G. (2000). *Consciousness: How matter becomes imagination.* New York: Allen Lane.

Hawkins, J., & Blakeslee, S. (2005). *On Intelligence.* New York: Henry Holt.

Lee, D. D., & Seung, H. S. (2001). Algorithms for Non-negative Matrix Factorization. *Adv. Neural Inf. Proc. Syt.* , 13:556-562.

Lee, D. D., & Seung, H. S. (1999). Learning the parts of objects by non-negative factorization. *Nature* , 6755:788-791.

Lee-Teng, E., & Sherman, M. (1966). Memory consolidation of one-trial learning in chicks. *Proc. Natl. Acad. Sci.* , 56(3):926-931.

Montague, P. R., Hyman, S. E., & Cohen, J. D. (2004). Computational roles for dopamine in behavioural control. *Nature* , 431:760-767.

Nolte, J. (2008). *The Human Brain: An Introduction to Its Functional Anatomy.* Philadelphia: Mosby.

Panksepp, J. (2004). *Affective neuroscience: The foundations of human and animal emotions.* Oxford: Oxford University Press.

Rock, I., & Heimer, W. (1959). Further Evidence of One-Trial Associative Learning. *Am. J. Psychology* , 72(1):1-16.

Waelti, P., Dickinson, A., & Schultz, W. (2001). Dopamine responses comply with basic assumptions of formal learning theory. *Nature* , 412:43-48.