Implementing Bayes' Rule with Neural Fields

Raymond H. Cuijpers[1,⋆] and Wolfram Erlhagen[2]

Nijmegen, P.O. Box 9104, The Netherlands
Tel.: +31 24 3612608; Fax: +31 24 3616066
r.cuijpers@nici.ru.nl
[2] Department of Mathematics for Science and Technology, University of Minho,
4800-058 Guimares, Portugal
wolfram.erlhagen@mct.uminho.pt

Abstract. Bayesian statistics is has been very successful in describing behavioural data on decision making and cue integration under noisy circumstances. However, it is still an open question how the human brain actually incorporates this functionality. Here we compare three ways in which Bayes rule can be implemented using neural fields. The result is a truly dynamic framework that can easily be extended by non-Bayesian mechanisms such as learning and memory.

Keywords: Bayesian statistics, Neural fields, Decision making, Population coding.

1 Introduction

Bayesian statistics has become a popular framework for describing various kinds of human behaviour under circumstances of uncertainty [1,2,3]. Generally, it is assumed that populations of neurons could encode probability distributions and indeed this can be used to predict overt behaviour in monkey [4]. One way to implement Bayes' rule is to let each neuron represent the likelihood of an entire probability distribution [5,6]. However, in these studies Bayes' rule is implicit in the decoding mechanism. Another way is to represent probability distributions in the log domain [7], so that Bayes' rule can be implemented by simply adding neuronal activities.

The neural field architecture has been quite successful in explaining behavioural data [8,9,10] and as a control architecture in cognitive robots [11]. Neural fields can support bi-stable activity patterns [12,13], which makes them suitable for memory and decision mechanisms [14,9]. However, these neural field properties are non-Bayesian. For one, the reported neural fields cannot support sustained multimodal activity patterns in a way that is necessary for representing arbitrary probability distributions. On the other hand, Bayesian statistics does not incorporate any temporal dynamics. We wondered whether it is possible to combine the best of both worlds. In particular, we wanted to know whether it is possible to implement Bayes' rule with neural fields.

[⋆] Corresponding author.

V. Kůrková et al. (Eds.): ICANN 2008, Part II, LNCS 5164, pp. 228–237, 2008.

2 Neural Fields of Leaky Integrate-and-Fire Neurons

A common model of a neuron in computational neuroscience is the so-called sigma node representing the average behaviour of leaky integrate-and-fire neurons [15]. Each sigma node has an internal state $u_i(t)$, which is analogous to a biological neuron's membrane potential, and an output activity $r_i(t)$, which is analogous to a biological neuron's firing rate [15]. For ease of reference we will use the names of the biological analogies. The sigma node reflects the average behaviour of one or more biological neurons. The firing rate is related to the membrane potential by:

$$r_i(t) = f(u_i(t))$$

where f is a thresholding function such as the Heaviside step function or the sigmoid function. The function f is often called the activation function.

The membrane potential changes dynamically depending on the input from other neurons and the leaking current.

$$\tau \frac{du_i(t)}{dt} = h - u_i(t) + \sum w_{ij} f(u_j(t)) \,, \tag{1}$$

where τ is a time constant, h is the equivalent of the resting potential, and w_{ij} are the connection weights between neuron i and neuron j. The first term on the right hand side results in an exponential decay of the membrane potential with time constant τ and the second term on the right hand side reflects the weighted sum of inputs from other sigma node neurons (Fig. 1A).

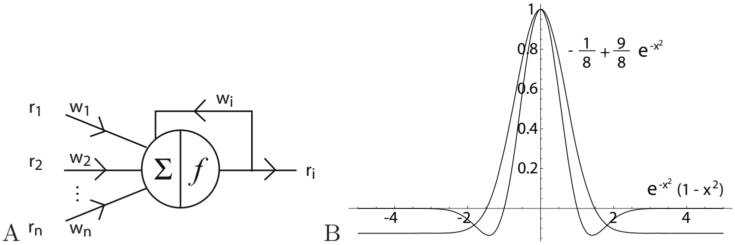

Fig. 1. a) Schematic diagram of a sigma node neuron i. b) Graphs of the lowered Gaussian and Mexican hat function with long and short inhibitory tails, respectively.

The structure of a neural network is determined by the connection weights w_{ij} in (1). It has been observed that in biological neural tissues neurons excite and inhibit nearby neurons in a centre-surround fashion (see e.g. [4]). In the continuous limit we obtain a neural field, whose dynamics are governed by Amari's equation for a neural field [16]:

$$\tau \frac{\partial u(x,t)}{\partial t} = h - u(x,t) + (w * f(u))(x,t) + S(x,t) \,. \tag{2}$$

Here $u(x,t)$ denotes the membrane potential of a neuron at location x, and w is the kernel imposing the centre-surround structure within the neural field and $*$ denotes spatial convolution, which is defined as $f*g = \int f(x-y)g(y)dy$. The term $S(x,t)$ denotes the external input of the neural field. The constant h in Amari's equation can easily be absorbed in the definitions of $u(x,t)$ and f, so we will omit it hereafter. The centre-surround excitation-inhibition is commonly modelled by a lowered Gaussian (Fig. 1B):

$$w(x-y) = A\exp\left(-\frac{(x-y)^2}{2\sigma^2}\right) - w_{\text{inhib}}\,, \tag{3}$$

where A and σ denote the amplitude and width of the lowered Gaussian, respectively. The constant $\mathrm{w}_{\text{inhib}}$ determines the amount by which the Gaussian is lowered. The Gaussian shaped kernel effectively smoothes the neural field activity locally because it averages the activity of nearby neurons. Other commonly used kernels, such as the Mexican hat function (Fig. 1B), have similar properties. With a suitable choice of parameters A and w_{inhib} neural fields governed by (2)(3) can sustain a self-stabilised local bump of activity [12,14,13]. This is clearly useful for memory-like functions especially because the self-stabilising capability makes the neural field very robust against noise. The global inhibition w_{inhib} prevents the co-existence of two or more self-stabilising bumps. The Mexican hat function (Fig. 1B) does admit multiple bumps [14], but their amplitudes cannot vary independently. This rules out the possibility to represent multi-modal probability distributions in this way.

3 Amari's Equation and Robustness against Noise

3.1 Amari's Equation with Linearised Recurrent Connection Kernel

Robustness against noise is a key concept that leads to Amari's equation (2). In order to obtain robustness against noise, the field activity must be smoothed spatially as time progresses. A simple way to achieve this is by replacing at every time step Δt a fraction of the field activity by the smoothed field activity. In the presence of sustained external input, another fraction is replaced by the external input. The latter is necessary to prevent exponential growth of the total field activity. If $u(x,t)$ denotes the activity of neuron x at time t and $S(x,t)$ represents external input, we can formalise this in the following way:

$$u(x,t+\Delta t) = (1-\varepsilon)u(x,t) + \alpha\varepsilon\,(k*u)(x,t) + (1-\alpha)\varepsilon S(x,t)\,, \tag{4}$$

where $u(x,t+\Delta t)$ is the updated neural field activity, $k(x)$is the smoothing kernel and α,ε are constants. The constant α controls the balance between smoothing and external input, and ε controls the updating rate. We can rewrite this as a partial differential equation by having $\varepsilon = \frac{\Delta t}{\tau}$ and taking the limit $\Delta t \to 0$ to obtain:

$$\tau\frac{\partial u(x,t)}{\partial t} = -u(x,t) + \alpha\,(k*u)(x,t) + (1-\alpha)S(x,t)\,. \tag{5}$$

where τ is some time constant.

When the external input $S(x,t) = s(x)$ is constant over time, the neural field activity $u(x,t)$ decays to a stationary solution. Let $u_\infty(x)$ denote the stationary solution, then it is easy to show that the external input must have the following form:

$$s(x) = (k_{ext} * u_\infty)(x) \equiv \frac{1}{1-\alpha}((\delta - \alpha k) * u_\infty)(x)\,, \tag{6}$$

where $\delta(x)$ is Dirac's delta function. The kernel defined in (6) has an interesting property. Suppose that a second neural field receives external input from the first using connection weights given by $k_{ext}(x) = (\delta(x) - \alpha k(x))/(1-\alpha)$, then the stationary solution of the second field equals the firing rate of the first field. In other words, the neural field activity $u(x,t)$ decays exponentially to $s(x)$. To make this statement more precise, we can show that the total activity decays exponentially over time. If we define the total activity of a neural field $z(x,t)$ as:

$$\bar{z}(t) = \int z(x,t)dx\,, \tag{7}$$

then we can rewrite 5 using 7 to:

$$\tau \frac{d\bar{u}(t)}{dt} = -\bar{u}(t) + \alpha K \bar{u}(t) + (1-\alpha)\bar{S}(t)\,. \tag{8}$$

Here we have used the shift invariance of the kernel and defined $K = \int k(x-y)dx$. If the total external input $\bar{S}(t) = \bar{s}$ is constant, the solution of (8) is simply:

$$\bar{u}(t) = A \exp(-(1-\alpha K)\frac{t}{\tau}) + \frac{1-\alpha}{1-\alpha K}\bar{s}\,. \tag{9}$$

This shows that the total activity $\bar{u}(t)$ decays exponentially with time constant $(1-\alpha K)/\tau$ to a constant times $\bar{s}$. This constant equals 1 independently of the value of α when $K = 1$.

3.2 Non-linear Dynamics of Amari's Equation

Equation (5) resembles Amari's original equation (2): the constants α and $1-\alpha$ can easily be absorbed in the definitions of the kernel and the external input, but the activation function f introduces a non-linearity that cannot be removed so easily. However, when the field activity is in the linear range of f, the dynamics of (5) will approximate that of (2). Amari's original equation can thus be obtained by replacing the convolution term $k * u \to k * f(u)$. This gives:

$$\tau \frac{\partial u(x,t)}{\partial t} = -u(x,t) + \alpha(k * f(u))(x,t) + (1-\alpha)S(x,t)\,. \tag{10}$$

We obtain the stationary solution of the external input for the non-linear Amari's equation in a similar way as in (6):

$$s(x) = \frac{1}{1-\alpha}\left(u_\infty(x) - \alpha(k * r_\infty)(x)\right) \tag{11}$$

$$\approx (k_{\text{ext}} * r_\infty)(x)\,, \tag{12}$$

where $r_\infty(x) = f(u_\infty(x))$. Clearly, the latter approximation is valid only when the activation function f is approximately linear.

4 A Probabilistic Interpretation for Neural Fields

In Bayesian decision making the likelihood of an event y for a given hypothesis x is transformed into a belief in that hypothesis. This inference is governed by Bayes' rule, which is defined as:

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)}\,, \tag{13}$$

where $p(x)$ is the prior probability, $p(x|y)$ the posterior probability, and $p(y|x)$ the likelihood of x. Typically, x is variable and y is constant so that the probability $p(y)$ is an arbitrary normalisation constant. In many decision problems, making a decision amounts to finding the value of x that maximises the posterior probability - the so-called *maximum-a-posteriori.*

In order to implement Bayes' rule with neural fields we need to represent the probabilities in such a way that Bayes' rule can be computed using simple addition of neural field activities. This can be achieved by representing probability distributions in the log domain [7]. However, if the dynamics of the neural fields are governed by Amari's equation (2) the activation function introduces a non-linearity in the system. In order to obtain similar dynamics as in the linearised version of Amari's equation (5) we need to scale the log-probabilities to the linear range of the activation function. In particular, if we use the following sigmoid function:

$$f(x) = \frac{1}{1+\exp\left(-4(x-\frac{1}{2})\right)}\,, \tag{14}$$

then $f(x)$ is approximately linear within the range (0,1) with slope $f'(\frac{1}{2}) = 1$ at $x = \frac{1}{2}$. Thus, we wish the function $g(x)$ to linearly remap the log-probabilities to the range $(0, 1)$. However, the log-probabilities lie in the range $(-\infty, 0)$, so that such a linear transformation does not exist. Therefore, we define a minimal probability value, say $p_{min} = 10^{-16}$, below which we ignore approximation errors. Then, we can define the function g as:

$$g(x) = 1 - \frac{x}{\ln p_{min}}\,, \tag{15}$$

which maps the range $(\ln p_{min}, 0)$ to $(0, 1)$. We can now make the following identification:

$$u(x,t) = g(\ln(p_t(x)))\,, \tag{16}$$

where $u(x, t)$ is the neural field activity of neuron representing x at time t, and $p_t(x)$ is some probability distribution of x at time t. With this identification we can incorporate Bayes' rule in the following way. Suppose neural field A is encoding the likelihood $p_t(y|x)$ of x at time t and neural field B is encoding the prior probability $p_t(x)$. We need to construct a third neural field C receiving inputs from neural fields A and B as external input, so that it encodes the posterior probability $p_t(x|y)$. Let

$$\begin{aligned} u_A(x,t) &= g(\ln\, p_t(y|x))\,, \\ u_B(x,t) &= g(\ln\, p_t(x))\,, \\ h_C &= -g(\ln\, p(y))\,, \end{aligned} \tag{17}$$

where $u_A(x,t)$ and $u_B(x,t)$ are the activities of neural fields A and B and h_C is a constant reflecting homogeneous global inhibition of neural field C. In case of the linearised Amari's equation , the stationary activity of neural field C can be shown to encode the posterior probability $p_t(x|y)$ by setting the external input to:

$$S_{C,\text{linear}}(x,t) = (k_{\text{ext}} * u_A)(x,t) + (k_{\text{ext}} * u_B)(x,t) + \frac{1-\alpha K}{1-\alpha} h_C\,, \tag{18}$$

where k_{ext} is the kernel defined in (6). Here we have used the shift-invariance of the kernel k (see (8)) to simplify the constant term involving h_C. By construction the external input of neural field C equals $k_{ext} * g(\ln\, p_t(x|y))$ and it follows from (6) that the steady state activity encodes the posterior probability as desired.

In a similar way we obtained the expression for the non-linear Amari's equation without approximating the kernel (11) and with approximated kernel (12):

$$\begin{aligned} S_{C,\text{non-linear}}(x,t) = &\frac{1}{1-\alpha}(u_A(x,t) - \alpha(k * f(u_A))(x,t) \\ &+ u_B(x,t) - \alpha(k * f(u_B))(x,t) \\ &+ h_C - \alpha K f(h_C))\,, \end{aligned} \tag{19}$$

$$S_{C,\text{approximate}}(x,t) = (k_{\text{ext}} * r_A)(x,t) + (k_{\text{ext}} * r_B)(x,t) + \frac{1-\alpha K}{1-\alpha} f(h_C) \tag{20}$$

In the remaining part of this paper we will refer to neural fields with external inputs given by (18), (19) and (20) by linear, non-linear and approximate, respectively.

5 Neural Field Simulations

For the simulations we used a scenario where the field position x represents the angular position subdivided in 100 intervals. The simulations were done using discrete versions of (5),(2) using Euler discretisation of both space and time. We used $n = 100$ neurons for encoding field position ($x_i = i\Delta x$ with $i = 0, \ldots, n-1$ and $\Delta x = 1$). Time was discretised as shown in (4) with $\Delta t = 1$. The parameters τ and α were set to $\tau = 10$ and $\alpha = \frac{1}{2}$. To prevent edge effects we used circular neural fields so that neuron $i = 99$ neighbours neuron $i = 0$. For the recurrent connections kernel we used a Von Mises distribution - the circular equivalent of the Gaussian normal distribution - so that $K = 1$. The Von Mises distribution is given by:

$$\text{vonmises}(x,\kappa) = \frac{1}{2\pi I_0(\kappa)} \exp \kappa \cos(\frac{2\pi x}{n\Delta x})\,, \tag{21}$$

where κ is a width parameter and $I_m(\kappa)$ is the modified Bessel function of the first kind with $m = 0$. The width parameter was set to $\kappa = 3^{-2}$ which approximately corresponds to a Gaussian width of $\sigma = 3$. For the likelihood $p(y|x)$ and the prior $p(x)$ we used Von Mises distributions centred on $x = 60$ and $x = 30$, respectively. The corresponding widths were set to $\kappa = 2^{-2}$ and $\kappa = 3^{-2}$. In order to simulate Bayes' rule the external inputs of the neural fields representing the posterior distribution are given by (18)-(20). White noise was added to the external input directly with an amplitude of ± 0.05.

How the neural field activities change over time is shown in the top row of Fig. 2 for linear (left), non-linear (middle), and approximate (right) neural fields. In the bottom row the corresponding firing rates are shown. Both the linear and non-linear neural fields converge to the posterior field activity, but the approximate neural field shows noticeable differences: the neural field activities are restricted to the range (0,1) due to the sigmoid activation function in (14). Consequently, the neural field activity due to the prior and the associated firing rates are not fully suppressed as is evident by the bumps at $x = 30$.

Fig. 3 shows the decoded probability distributions of the linear (left), non-linear (middle) and approximate neural fields (right). All neural fields build up activities at approximately the same location on similar time scales, but their amplitudes differ considerably. The amplitude of the approximate neural field (right) is much less than the other two. This is also evident if we plot the cross-section at different times in comparison to the true posterior probability distribution (bottom). It can

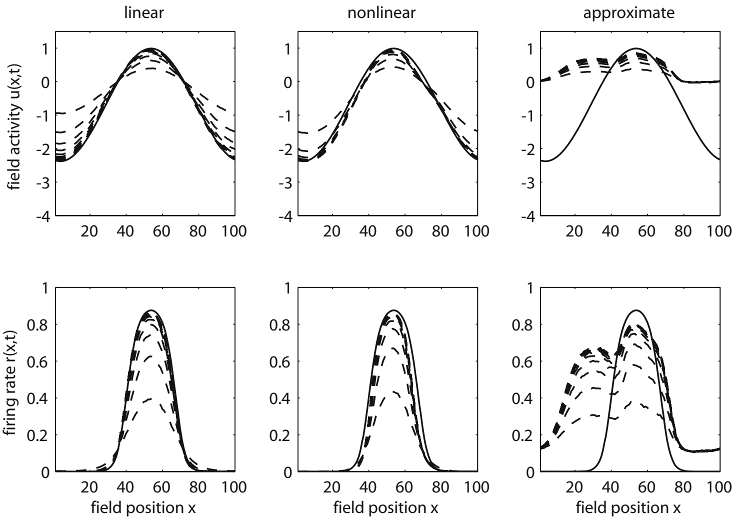

Fig. 2. Neural field activities $u(x, t)$ (top) and firing rates $r(x, t)$ (bottom) for linear (left), non-linear (middle) and approximate kernel. Neural field activities and firing rates are shown for every 10^{th} of 100 iterations (dashed curves). The solid curves indicate the stationary state corresponding to the true posterior.

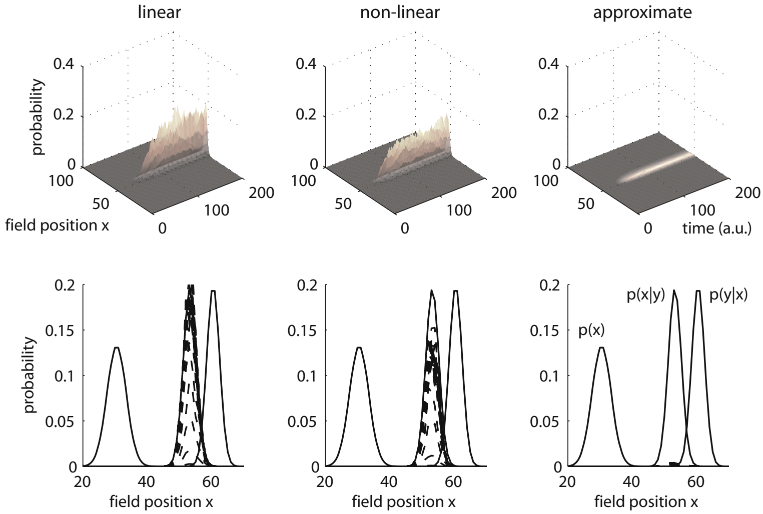

Fig. 3. Decoded probability distributions as a function of time (top) and for every 10th time step (bottom; dashed curves)

be seen that the approximations of the non-linear and approximate neural fields mainly affect the amplitudes of the decoded probability distributions.

The error in amplitude is easily amended by normalising the decoded probability distributions. To address this we compared the location and width of the normalised decoded probability distribution with those of the true posterior. In Fig. 4 the average error in peak location (left) and peak width (right) is shown as a function of time. For the peak location we used the expected value of the posterior distribution. For the peak width we used the standard deviation. The averages were taken from 200 randomly generated Von Mises priors and likelihoods. The location of the Von Mises distributions of both prior and likelihood were randomly drawn from a uniform distribution covering the entire neural field ($x \in [0, 99]$). The width of the distributions was uniformly drawn from $\sigma \in [1, 25]$ with $\kappa = 1/\sigma^2$. From Fig. 4 it is clear that the average error in peak location lies within 1 neuronal unit ($\frac{2\pi}{100} = 3.6°$) irrespective of which type of neural field was used. The average error in peak width is initially largely overestimated after which it decays to a constant value. Both the linear and non-linear neural fields approximately converge to the true posterior width within about 20 iterations. The rate of convergence is somewhat faster for the non-linear field. The approximate neural field does not converge to zero but stabilises on an overestimate of 3 neuronal units (10.8°). The angular error can be reduced by increasing the number of neurons. This is shown by the thin dashed line in Fig. 4. The error in decoded peak location for n=1000 is the same as for n=100 showing that the angular error is decimated (left). The error in decoded peak width is about ten

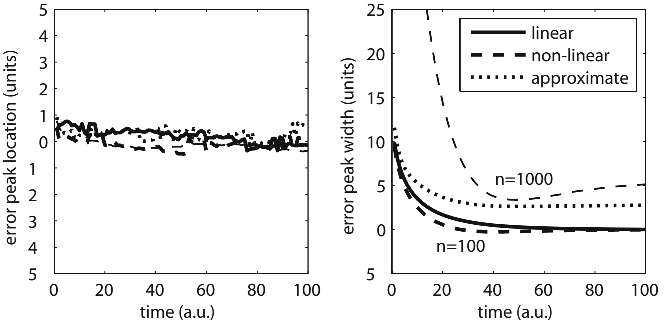

Fig. 4. The mean error in peak location (left) and width (right) of the decoded peak. The lines indicate the errors obtained for linear (solid line), non-linear (dashed line) and approximate (dotted lines) neural field. The time constant was set to $\tau = 10$.

times larger (initial value 100.2) reflecting the fact that the probability distributions are ten times wider when expressed in neuronal units.

6 Discussion and Conclusions

We have developed a way in which Bayes' rule can be represented using neural fields. To do so we superimposed the neural field activities representing the log-likelihood and the log-prior distribution. Due to the non-linear activation function in Amari's equation the superposition of field activities leads to an approximation error that mainly affects the decoded amplitude of the posterior distribution. The approximation error can be fixed by normalisation, but the approximate neural field overestimates the width of the decoded posterior distribution. The approximation error depends on how well the neural field activities fall in the linear range of the activation function. If necessary, the linear approximation could be reduced by adjusting this mapping, but the signal-to-noise ratio will deteriorate.

The proposed implementation of Bayes' rule can be used to build a dynamic version of Bayesian decision making as was used in [17]. Whether the dynamics is in agreement with experimental findings is still an open issue. An additional benefit is that Bayesian inference can now be smoothly coupled with non-Bayesian mechanisms such as Hebbian learning, memory and decision mechanisms.

Acknowledgements

The present study was supported by the EU-Project "Joint Action Science and Technology" (IST-FP6-003747).

References

1. Gold, J.I., Shadlen, M.N.: The neural basis of decision making. Annu. Rev. Neurosci. 30, 535–574 (2007)
2. Hillis, J.M., Watt, S.J., Landy, M.S., Banks, M.S.: Slant from texture and disparity cues: optimal cue combination. J Vis 4, 967–992 (2004)
3. Kuss, M., Jäkel, F., Wichmann, F.A.: Bayesian inference for psychometric functions. J. Vis. 5, 478–492 (2005)
4. Chen, Y., Geisler, W.S., Seidemann, E.: Optimal decoding of correlated neural population responses in the primate visual cortex. Nat. Neurosci. 9, 1412–1420 (2006)
5. Ma, W.J., Beck, J.M., Latham, P.E., Pouget, A.: Bayesian inference with probabilistic population codes. Nat. Neurosci. 9, 1432–1438 (2006)
6. Zemel, R.S., Dayan, P., Pouget, A.: Probabilistic interpretation of population codes. Neural Comput. 10, 403–430 (1998)
7. Rao, R.P.N.: Bayesian computation in recurrent neural circuits. Neural Comput. 16, 1–38 (2004)
8. Cisek, P.: Integrated neural processes for defining potential actions and deciding between them: a computational model. J. Neurosci. 26, 9761–9770 (2006)
9. Wilimzig, C., Schneider, S., Schner, G.: The time course of saccadic decision making: dynamic field theory. Neural Netw. 19, 1059–1074 (2006)
10. Erlhagen, W., Schöner, G.: Dynamic field theory of movement preparation. Psychol. Rev. 109, 545–572 (2002)
11. Erlhagen, W., Mukovskiy, A., Bicho, E.: A dynamic model for action understanding and goal-directed imitation. Brain Res. 1083, 174–188 (2006)
12. Guo, Y., Chow, C.C.: Existence and stability of standing pulses in neural networks: I. existence. SIAM J. Appl. Dyn. Sys. 4, 217–248 (2005)
13. Taylor, J.G.: Neural bubble dynamics in two dimensions: foundations. Bioligal Cybernetics 80, 393–409 (1999)
14. Erlhagen, W., Bicho, E.: The dynamic neural field approach to cognitive robotics. J. Neural. Eng. 3, R36–R54 (2006)
15. Trappenberg, T.P.: Fundamentals of computational neuroscience. Oxford University Press, New York (2002)
16. Amari, S.: Dynamics of pattern formation in lateral-inhibition type neural fields. Biol. Cybern. 27, 77–87 (1977)
17. Cuijpers, R.H., van Schie, H.T., Koppen, M., Erlhagen, W., Bekkering, H.: Goals and means in action observation: a computational approach. Neural Netw. 19, 311–322 (2006)