Latency-based probabilistic information processing in a learning feedback hierarchy

Alexander Gepperth

Abstract— In this article, we study a three-layer neural hierarchy composed of bi-directionally connected recurrent layers which is trained to perform a synthetic object recognition task. The main feature of this network is its ability to represent, transmit and fuse probabilistic information, and thus to take near-optimal decisions when inputs are contradictory, noisy or missing. This is achieved by a neural space-latency code which is a natural consequence of the simple recurrent dynamics in each layer. Furthermore, the network possesses a feedback mechanism that is compatible with the space-latency code by making use of the attractor properties of neural layers. We show that this feedback mechanism can resolve/correct ambiguities at lower levels. As the fusion of feedback information in each layer is achieved in a probabilistically coherent fashion, feedback only has an effect if low-level inputs are ambiguous.

I. INTRODUCTION

With the advent of Bayesian inference accounts of biological information processing [2], not only the question of neural coding received renewed interest, but also the issue how neural populations can implement the required mathematical operations for inference, most notably population-level multiplication which seems to be a requirement for a full treatment of probability distributions. In this article we explore a neural space/latency code to offer a new perspective on inference operations. While a previous publication[3] has demonstrated that such a neural code can implement Bayes-optimal decision making as a consequence of recurrent neural population dynamics, this article is dedicated to the study of hierarchical information processing, especially learning and feedback and their role in the probabilistic representation of information. Evaluations are carried out in simulated scenario inspired by works on robotic object recognition[1].

A. Approach overview and article structure

The architecture we describe and study here (see Fig. 1) contains a number of recurrent neural layers (see Sec. II-B) connected by feed-forward and feed-back connections that are subject to learning, and whose precise workings are described in Sec. II-C. In Sec. II-A and Fig. 2, we will describe the stimuli that serve as input to the lowest network layer

In this section, we explain salient points of the presented work along with a number of "shortcuts" introduced to keep things as conceptually simple as possible.

Alexander Gepperth is with ENSTA ParisTech, 828 Blvd des Marechaux, 91762 Palaiseau, France (email: alexander.gepperth@ensta-paristech.fr) and with INRIA FLOWERS, 100 avenue de la Liberation, 44801 Talence, France.
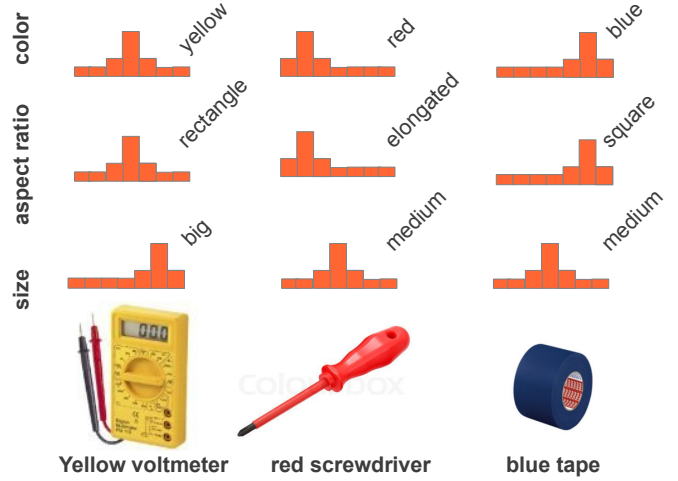
Fig. 2. Simplified synthetic object recognition task: objects and their visual properties (expressed as feature histograms) in the three modalities color, aspect ratio and size. Please note the overlap in the "size" modality between screwdriver and tape, leading to potential ambiguities in feedforward processing.

a) Input stimuli: We simulate a robotic object recognition task modeled after [1] and containing the objects "red screwdriver", "yellow voltmeter" and "blue tape" (see Fig. 2). The simplification consists of choosing a small number of objects, and an equally small number of discrete values possible in each of the three visual modalities analyzed. This task is therefore not a real object recognition benchmark but rather a tool to illustrate properties of our model.

b) Role of lateral connections: Although common theories of neural coding generally work on the single-neuron level without reference to other neurons in the same population[4], [5], [6], [7], [8], it is a fact that biological processing makes heavy use of **lateral connections**: which motivated us to investigate how this can be reconciled with a probabilistic interpretation of neural activities. In addition to feedforward and feedback connections, each layer in our model also contains strong lateral connections. We posit that these contain a *data model* expressing the probability $P(\vec{M}|\vec{S})$ of an underlying "true" stimulus $\vec{M}$ given the noisy/mixed/corrupted stimulus $\vec{S}$. Although these connections should be learned from data for real problems, the simple synthetic classification problem allows us to use space-independent interaction kernels with local excitation/global inhibition encoding the data model of a single peak at an arbitrary position.
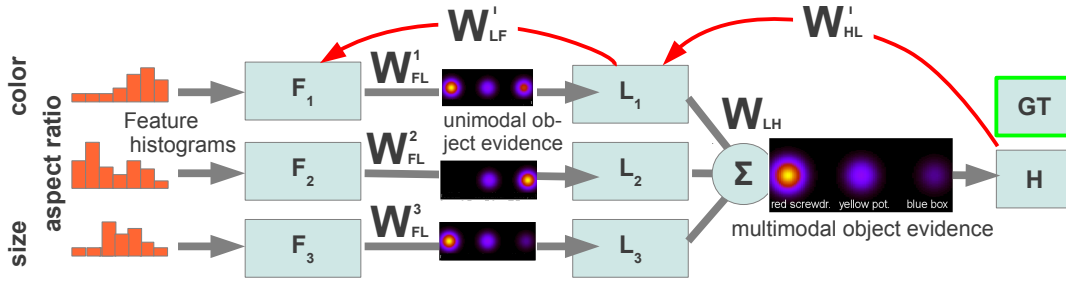
Fig. 1. Neural hierarchy which is the basis for all investigations in this article. Recurrent neural layers F_i, L_i and H are indicated by named blue boxes, feed-forward/feed-back transmission by gray/red arrows. Modeled after a robotic object recognition task[1], the lowest layers F_i receive feature histograms computed from the visual modalities color, aspect ratio and size, while layers L_i and H process unimodal/multimodal object recognition results, respectively. The feedforward weights $\vec{W}_{\mathrm{FL}}^i$ and $\vec{W}_{\mathrm{LH}}$ are adapted with the help of an external supervision signal GT that contains object identity information, whereas the learning of feedback connections $\vec{W}_{\mathrm{FL}}^i$ and $\vec{W}_{\mathrm{LH}}$ is self-supervised by the respective target representations, L_i and F_i.

c) Role of recurrent dynamics: Given such a data model, **recurrent dynamics** defined by the lateral connections will converge to a "decision" state maximally compatible with the input and the data model. For the neuron model we use, this can be shown through Lyapunov analysis[9]. Decision latency **encodes** confidence, i.e., match of input to data model, resulting in localized attractor states of different latency, which is a well-documented effect[1], [10]. In [3], we proposed a probabilistic interpretation of this combined space/latency code, where the position of the attractor state represents the most likely "true" stimulus $\vec{M}^* = \arg\max P(\vec{M}|\vec{S})$ whereas the latency expresses its confidence $P(\vec{M}^*|\vec{S})$. Thus, neural populations do not represent full distributions since sub-leading interpretations of the input $\vec{S}$ are suppressed.

d) Role of adaptive connections: As in other models, **feed-forward connections** define neural selectivities by implementing a "tuning function". These selectivities are shaped by online logistic regression (with an object label provided as supervision signal), just as are the weights of **feedback connections** where logistic regression attempts to predict lower-level activity as a function of higher-level activity (no external supervision required). These predictions that are carried by feedback connections are fused with lower-level layer activities for **disambiguation** or **error correction** purposes. Indeed, this article makes a point of showing how online learning can smoothly and stably interact with recurrent information processing.

e) "Gating" of feedback through attractor dynamics: Non-dynamic hierarchical models would need to distinguish a feedforward and a feedback phase of processing. In our case this is not required, and each layer sends feedforward and feedback information freely whenever sufficient activity develops. Since only the lowest layers of the network are directly connected to the input, they will form stable attractor states before next-higher layers, whose feedback will consequently be ignored, thus avoiding uncontrolled cyclic loops. In order to permit feedback to take effect, it is necessary to explicitly suppress neural activity, starting at the layers directly below the highest one. After suppression, neural activity will re-converge taking into account feedback

input, and eventually send feedback to lower layers which are then in turn suppressed (see Fig. 4). This process, which is reminiscent of cortical oscillations[11]), is triggered by the presence of a decision (i.e., activity) in the highest hierarchy layer and is stopped only upon reaching the lowest layer.

B. Related work

There exists a large body of literature[5], [4], [8], [12], [4], [6], [7] on probabilistic aspects of neural coding. Most authors explicitly assume that neural population activity is related to probability distributions[5], [4], [12], [6], [7]. A very influential idea posits that single-neuron activity is related to log-probability[6], [7], [8], which would allow to perform multiplications by summation. Other authors have questioned the practicability of this scheme[5] as it would require re-encoding at each hierarchy level which is deemed unfeasible. An alternative approach[5] is to consider single neuron's firing rates as the realizations of Poisson-like random variables whose mean is determined by the match of neural preferences to afferent input. Under certain conditions, sums of two such variables can be proven to come from a distribution whose mean corresponds to the product of individual means, thus realizing a multiplication by summation.

The effect of input ambiguity/conflict on the latency of attractor formation in the dynamic neural field model has been documented in [1], [10]. However no functional role has been proposed for this effect up to this date. The effect of response latency is ubiquitous in neurobiology, and there is converging evidence from both physiological[13], [14], [15], [16] and behavioral [17], [18] investigations that it plays a role in the neural encoding of information. Observed neural response latency is linked to different causes, some of which are overlapping: neurons in the striate cortex, for example, encode stimulus contrast into response latency[14]. On the behavioral side, it has been found that decision making processes typically take longer depending on the number of conflicting alternatives[17], conceivably reflecting increased response latency on the neural level. Similar effects have been observed in language processing[18], where the ambiguity, i.e., the number of different interpretations,

gives rise to delayed responses. In contract to works which train recurrent networks directly for function approximation purposes[19], [20], the recurrent layers we describe here are not, for the moment, trained (although this is an obvious next step) but rather aim to perform signal completion and correction in the sense of Hopfield models, a side effect of which is the variable time-to-convergence which we propose to exploit.

Since biological neural networks are strongly hierarchical, the existence of response latency automatically implies the existence of input latency at higher hierarchy levels. In [15], it is speculated how input latency could be decoded in downstream neural populations; the neural space/latency code which we describe here is just such a mechanism, which is computationally simple and biologically plausible.

C. Key research questions

In contrast to the investigations of [3] where the space/latency code was shown to approximate Bayesian inference and optimal decision, this article focuses on effects and issues arising from learning and the construction of deep hierarchies. In particular, we investigate the following topics for the architecture shown in 1:

Plausible decisions in the face of noise and ambiguity We show that each hierarchy layer can decode the space/latency code encoded by preceding layers[3], and demonstrate that this allows to take plausible classification decisions at the top-level. In contrast to [3], we will not show Bayes-optimality explicitly but rely on plausibility arguments, demonstrating that ambiguous modalities have reduced influence and that this information is preserved through all hierarchy levels. Here, ambiguity can arise from noise or due to inherently ambiguous object models, see Fig. 2.

Concurrent learning and feedback We show that feedback can be active during learning and is beneficial in situations where learning would normally be impossible due to ambiguous object definitions.

Disambiguation and error correction due to multi-layer feedback mechanism We show that information coming from higher hierarchy layers has the potential to correct ambiguous or erroneous inputs at all lower hierarchy stages. In line with the probabilistic interpretation of the space/latency code[3], we show that such an effect is only possible for layers whose inputs exhibit a sufficient degree of ambiguity.

II. METHODS

We base our investigation on the dynamic neural field model [21]. which was originally proposed to describe pattern formation in the visual cortex. Essentially, dynamic neural fields are a class of recurrent neural network models that have been extensively used for modeling cognitive phenomena like decision making [22], motor planning [10], spatial cognition [23], eye movement preparation [24], [25] and object recognition [1], [26]. Basic elements are simple dynamic-state neurons, a fixed lateral connectivity, and a (usually sigmoid) non-linearity.
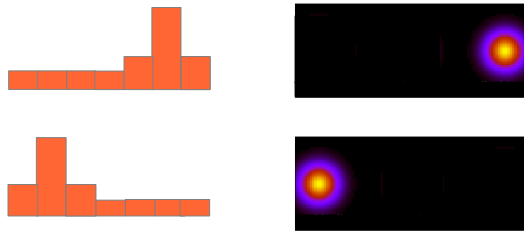

Fig. 3. Examples of the one-dimensional encoding of feature histograms (left row) into population codes $H_i(\vec{x}, t)$ (right row) suitable as input to neural layers.

A. Input stimuli

We simulate a robotic object recognition task containing the objects "red screwdriver", "yellow voltmeter" and "blue tape" (see Fig. 2) which are represented by synthetically generated "measurements" in the modalities of color, aspect ratio and size. As in [1], measurements are represented by histograms over a single dimension in each of these modalities. Input histograms are assumed to be unimodal and Gaussian, with only three possible locations for these Gaussians of peak strength 1. Actual representation of input stimuli is effected by population-encoding feature histograms along one axis of a two-dimensional image of dimensions $(60, 10)^T$ which can directly serve as input to a recurrent neural layer. This (simplified) population encoding step is schematically depicted in Fig. 3.

B. Single-layer model equations

We use a slightly modified version of the rate-coded dynamics proposed in [21]:

$$\tau \dot{u} = -u + \alpha f_I[S] + \beta \left(w * f[u] \right) + \gamma \sigma + h \qquad (1)$$

Here, the quantity $u(\vec{x}, t)$ represents the membrane potential of the field at time t and position $\vec{x}$, $S(\vec{x}, t)$ the afferent input, $w(\vec{x} - \vec{x}\prime)$ the fixed lateral *interaction kernel*, $f[u]$ the point-wise applied *transfer function* given by $f[u] = \left(1 + \exp(\frac{-2(u-\theta)}{\nu}) \right)^{-1}$, and $\sigma(\vec{x}, t)$ normally distributed white noise. τ determines the time scale of field evolution, and h is the *resting potential*, i.e., the equilibrium potential in case of no input. In addition to the original model, we include a point-wise applied *input transfer function* $f_I[S]$ which is a tool to bring the sum of inputs into a value range where it can excite the field effectively. f_I will be replaced by a homeostatic self-adaptation process in the future, such as proposed in [27]. Here, the goal is to amplify weak inputs somewhat but to limit input values to $[0, 1]$. We therefore choose

$$f_I[S] = \begin{cases} k_I S & \text{if } k_I S \leq 1 \\ 1 & \text{else} \end{cases} \qquad (2)$$

with a suitably chosen constant k_I that depends on average input strength. The coefficients α, β and γ respectively determine the contribution of afferent input, lateral recurrent interactions and noise. The interaction kernel $w(\vec{x} - \vec{x}\prime)$ is usually chosen to be symmetric: $w(\vec{x} - \vec{x}\prime) = a_0 G_{\mu=0, \sigma_{\text{on}}}(\vec{x} -$
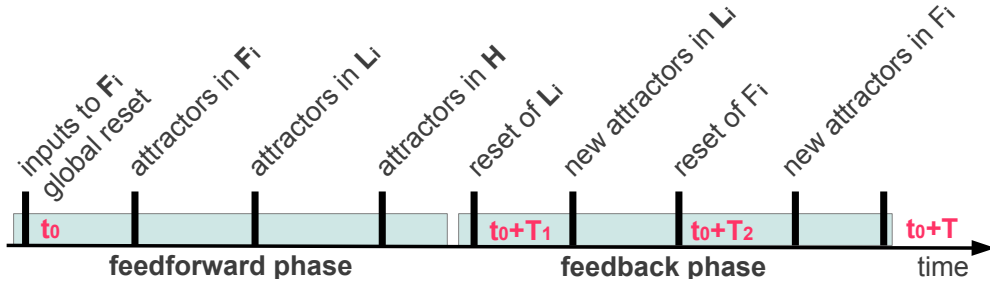
Fig. 4. Time course of a single input presentation to the network starting at simulation time t_0. Please note that the division into a feedforward and feedback phase is made here for instruction. In reality, feedback is generated and transmitted at all times, even in the "feedforward phase", although it has no effect there due to the attractor properties of recurrent layers. The resetting operations in the feedback phase thus do not change network dynamics, they just allow feedback to take effect by suppressing attractor solutions and forcing a re-convergence, this time taking feedback influence into account.

$\vec{x}\prime) - b_0 G_{\mu=0,\sigma_{\text{off}}}(\vec{x} - \vec{x}\prime) - c_0$, where $G_{\mu=0,\sigma}(\vec{x})$ denotes a Gaussian with mean μ and standard deviation σ, and $\sigma_{\text{on}} < \sigma_{\text{off}}$. The constants a_0, b_0, c_0 are chosen suitably to achieve the desired level of local excitation/inhibition(a_0, b_0) as well as global inhibition (c_0). To ensure numerical stability, we clip the neural field potentials $u(\vec{x}, t)$ whenever they exceed the range defined by $[u_{\min}, u_{\max}]$.

C. Learning and transmission in feed-forward and feedback connections

The total input to an arbitrary layer X is defined by the sum of its feedforward and feedback components. For the lowest hierarchy layer, feed-forward inputs are the population-encoded feature histograms $H_i(\vec{x}, t)$, see Figs. 2 and 3. The symbol $u^X(\vec{x}, t)$ denotes that layer's dynamic membrane potential governed by Eqn.(1), whereas $\sigma(x) = \frac{1}{1+e^{-x}}$ denotes the logistic function used by logistic regression learning. As illustrated in Fig. 1, W_{FL}^i denote weights from layers F_i to L_i, W_{LH}^i denotes weights from layers L_i to H and similarly for feedback connections.

$$S^{F_i}(\vec{x}, t) = \vec{A}_i(\vec{x}, t) + \sigma\left(W_{\text{LF}}^i \cdot u^{F_i}\right)$$
$$S^{L_i}(\vec{x}, t) = \sigma\left(W_{\text{FL}}^i \cdot u^{F_i}\right) + \sigma\left(W_{\text{HL}}^i \cdot u^H\right)$$
$$S^H(\vec{x}, t) = \sigma\left(\sum_i W_{\text{LH}}^i \cdot u^{L_i}\right) \quad (3)$$

Logistic regression learning assumes the existence of input and target representations denoted I, T and adapts the weights W_{IT} by minimizing the quadratic error

$$\sigma\left(W_{\text{IT}} \cdot I - T\right)^2 \to \min \quad (4)$$

by online gradient descent using a step size λ_{LR}. Note that a representation, denoted D, that will receive the signals computed from these weights according to Eqn.(3), need not be identical to T. As shown in Fig. 1, feed-forward connections connect either layers $F_i \to L_i$ so that we have $I = F_i, D = L_i$, or $L_i \to H$ which gives $I = L_i, D = H$. In both cases however the learning target is the externally given object identity: $T = \text{GT}$. For feedback connections

$H \to L_i$ and $L_i \to F_i$, no external supervision is needed and we always have $T = D$.

D. Temporal organization of a single input presentation

Feature histograms $\vec{H}_i$ are presented to the lowest hierarchy levels F_i of the network at time $t_0 + 1$ and maintained for a total of T simulation steps. Directly before this happens, at time t_0, all field potentials are reset to the resting potential h, see Sec. II-B. Subsequently, field potentials and weights evolve freely according to the dynamic model of Eqn. (1) and the weight adaptation rule of Eqn. (4), leading to eventual weight adaptation and attractor formation. At $t_0 + T_1 < t_0 + T$, potentials in the fields L_i are reset to the resting potential, whereas fields F_i are reset at a later time $t_0 + T_1 < t_0 + T_2 < t_0 + T$. These two resetting operations have the goal of permitting feedback influence during the re-convergence of the fields, which is not possible before the reset due to the stability of attractor solutions which ignores inputs. For feedback to be present, an attractor state must have formed in H at $t_0 + T_1$, and likewise a new attractor state must have formed in L_i after the reset at $t_0 + T_1$. These conditions can be easily met by evaluating average convergence time, whose variation due to the space/latency code is normally bounded. An overview over the temporal order of events during a single input presentation is given in Fig. 4.

III. Experiments

For all experiments, the layers F_i, L_i and H are connected as indicated in Fig. 1. The length of a single input presentation is set to $T = 400$, the first reset is performed at $T_1 = 200$ and the second at $T_2 = 300$. The learning rate for all feed-forward and feedback connections is $\lambda_{\text{LR}} = \frac{0.05}{60 \cdot 100}$. New inputs, as shown in Fig.2 and encoded as described in Sec. II-A are provided every T iterations. The three different simulated objects are presented one after the other as described in Sec. II-D, and repeat every $3T$ iterations. In the beginning there is a learning phase of 24000 iterations corresponding to 20 presentations per object, after which learning is disabled for performing experiments. Recurrent
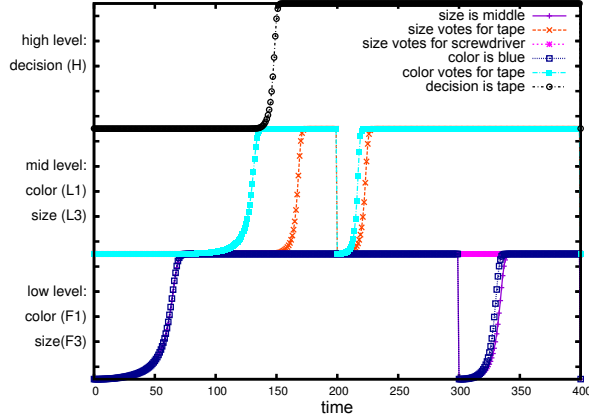
Fig. 5. Disambiguation by feedback influence: time course of activity in selected layers when presenting the "blue tape" object. Lowest row: unambiguous feature fields F_1 (color) and F_3 (size) have exactly the same latency. Middle row: unimodal object field L_1 (color) and L_3 (size), the latter being unable to build activity due to ambiguity. The peak is achieved only when there is a decision in the top-level layer which is transmitted back to L_3, thus achieving a disambiguation of what would otherwise be an impossible-to-resolve deadlock. Note that the latency in L_3 after the first reset is almost the same as latency in L_1 since feedback input strongly reduces input ambiguity.

Fig. 6. Illustrating the space-latency code: Time course of neural activity in selected layers of the hierarchy when presenting the "yellow voltmeter" object and corrupting histogram input for the color modality. Lowest row: activity in layers F_1 (color) and F_2 (aspect ratio): Middle row: activity in unimodal object layers L_1 (color) and L_2 (aspect ratio). Top row: final decision of the network, expressed by activity in layer H. When comparing the responses of low-level color field F_1 and aspect ratio field F_2, a clear latency difference may be perceived. A similar difference is consequently observed for L_1 and L_2. After the first reset, mid-level color layer is corrected by feedback, and after the second reset the low-level color field F_1 is corrected as well. The size modality is not shown as it behaves identically to aspect ratio here (i.e., has no noise).

layers evolve according to the dynamics defined in Sec. II-B, and feedforward and feedback connections are learned as stated in Sec. II-C. We choose a uniform parametrization of neural field layers of size 60x10 (see Sec. II-B): $\tau = 15$, $\theta = 0$, $\nu = 2.5$, $\alpha = 1$, $\beta = 4$, $\gamma = 0.11$, $\sigma_{on}=3$, $\sigma_{off} = 6$, $a_0 = b_0 = 1$, $c_0 = 0.55$, $h = -1$. Zero-padding boundary conditions are used for all lateral interactions. The input transfer function constant k_I is set to 1 for the fields F_i, to 1.8 for the fields L_i and to 1.3 for the field H. We always identify response latency with the number of elapsed simulation "ticks" until an activity ≥ 0.9 is first observed in a certain field.

A. Reactivation and disambiguation by feedback

Using the experimental protocol described at the beginning of this section, we investigate the reaction of the proposed system to ambiguity. This ambiguity stems from the fact that there is an overlap between the objects "red screwdriver" and "blue tape" in feature space since they have both the size "medium". Therefore, the size "medium" votes for both of these objects in the mid-level of the hierarchy (i.e., the field L_3 for the modality "size"). Due to the properties of logistic regression, this fact will lead to diminished input strength to L_3: instead of one peak of amplitude 1.0, there will now be two peaks, reflecting the aforesaid ambiguous vote, of amplitude $\tilde{0}.5$. Activity will still appear in L_3 since the input transfer function of that field sees to it that these inputs are boosted to a sufficient strength. However, as a consequence of competition between the two possible objects, the response will be delayed, reflecting its lower confidence
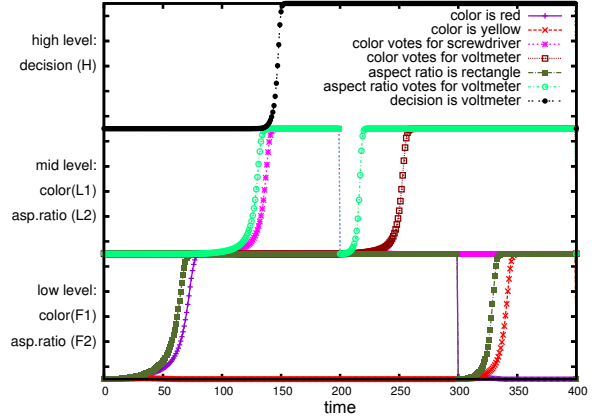
(i.e., probability under the data model). This in turn will delay activity buildup in the highest layer H, expressing that the top-level decision is not as certain as it could be as it is partially based on very ambiguous data. Indeed, the vote of L_3 is not really taken into account in defining the response of H as it comes too late, demonstrating the basic principle of probabilistic information processing in this architecture: later-coming inputs have less influence in attractor formation in recurrent layers. The time course of neural activities in this experiment is given in Fig. 5.

B. Space-latency code and its influence on decision making

We investigate what happens when the feature histogram in a single modality is ambiguous and also incorrect. We use the experimental setup and parameters described at the beginning of this section, repeatedly replaying the three objects of Fig 2 and their visual properties after a learning phase. Only one aspect differs: for the presentations of the "yellow voltmeter" object, we put a Gaussian of strength 1.0 at the (incorrect) position "red", as well as a Gaussian of strength 0.8 at the (correct) position "yellow" in the color modality (field F_1), simulating a measurement ambiguity leading to a locally wrong conclusion. This wrong conclusion is propagated forward to L_1 where it activates the "screwdriver" population. As the other modalities, that is to say the fields $L_i, i \neq 1$, vote for the correct object (yellow voltmeter), a correct high-level decision will be taken in H all the same. Feedback from H to L_1 will not have an effect because the attractor state in L_1 ignores all input. After the first reset
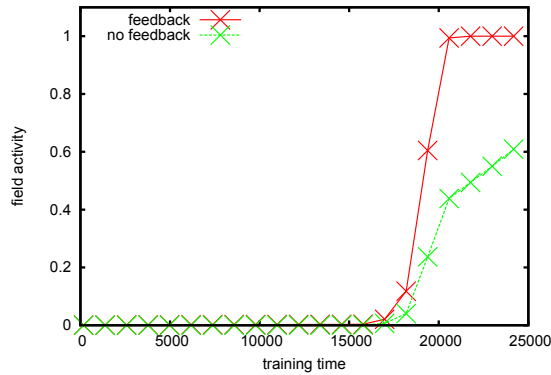
Fig. 7. Feedback improves learning efficiency. For the unimodal object representation field of the ambiguous size modality L_3, we show the maximal activity at the site "red screwdriver" in response to the regularly presented "red screwdriver" object over the course of learning. As can be seen, this amplitude increases much more rapidly if feedback is activated due to the disambiguation capability of feedback from the top-level field H.

at $T_1 = 200$, this attractor state is suppressed, and therefore the "voltmeter" population can now be activated in L_1. After the second reset at $T_2 = 300$, a similar thing happens in the representation F_1, i.e., a re-convergence to the correct stable state "yellow" caused by feedback. The precise time course of neural activities in this experiment is shown in fig. 6.

C. Influence of feedback on learning

As stated before, feedback is active at all times, especially during the learning process in the first 24000 iterations. As the previous two experiments clearly demonstrate, this does not in any way corrupt the learning process since feedforward and feedback connections, which are after all the result of learning, behave in the expected way. It it however legitimate to ask about whether feedback has a (positive or negative) impact on learning efficiency.

When training feedforward connections, the goal is to map a certain representation I (any of the F_i or L_i) to GT, see Fig. 1 and Sec. II-C, which is only possible if there is activity in I which is not always the case. For example, in the size modality a value of "medium" may indicate the presence of both the "red screwdriver" or "blue tape" objects. The input from F_3 to L_3 will therefore be ambiguous and activity in L_3 will appear, if at all, with strong delay which impairs learning[1].

Here, we observe the development of activity at the "middle" site in L_3 for presentations of the "red screwdriver" object. Without feedback, we expect that this quantity will increase more slowly than with feedback, as feedback provides additional input to that site coming from the correct high-level decision in H. Activity in H is formed by inputs

---

[1]One may be tempted to think that it will not appear at all as both votes are equally strong. However as the feedforward connections are also adapted to GT during a single object presentation, the correct vote will prevail after some time.

from the color and aspect ratio modalities which are unambiguous. The feedback disambiguation thus achieved in L_3 will make this f¡ield converge faster, improving the efficiency of learning as more time steps are available for it. The results are shown in Fig. 7 and confirm our reasoning, showing that activity rises much more quickly when feedback is present.

IV. DISCUSSION AND OUTLOOK

A. Summary and general assessment

This article is based on a novel probabilistic interpretation of neural activities making use of biologically plausible neural dynamics. Based on a simple object recognition "toy task", we have shown prominent properties of the approach when dealing with ambiguity or conflict in the system. Namely, it was shown that these effects are translated into a delay of neural responses, which will reduce or eliminate their influence on the next level of neural layers. This space/latency code was thus shown to be *bf encoded* and *decoded* implicitly by recurrent neural layers involving a data model encoded into lateral connections. We have furthermore proposed and validated a stable multi-stage feedback mechanism that removes a principal shortcoming of the approach, i.e., the destruction of sub-leading interpretations, and we have demonstrated that this feedback mechanism can run in parallel to online learning processes which are a necessary pre requisite when working with real-world data. Most importantly, this study has documented the beneficial use of feedback information regarding error correction and disambiguation, both of which will be of high value in noisy environments. As a last point, we wish to underscore that our approach, although relying on a rate-coded model, will very likely function just as well with a spiking neuron model (as the underlying recurrent dynamics is essentially that of an integrate-and-fire model), and that this approach combines the temporal properties hitherto attributed only to spiking models and the simplicity and efficiency of rate-coded approaches.

B. Shortcomings of this study

There are several principal shortcomings in the investigation presented here: first of all, learning is conducted in a slightly unrealistic fashion, using object identity information (encoded in the representation GT) for training feed-forward connections at all layers. A better way would be to provide this information only to the highest-level layer and propagate it downwards via feedback connections, which seems compatible with the space-latency code approach but would require some adaptations to the structure of the hierarchy. Secondly, the lateral interaction kernels we use are non-adaptive and encode a very restricted data model: a single peak of fixed size. If real data, which will be much more noisy and ambiguous than those used here, should be connected to the architecture, adaptive lateral interaction kernels will be necessary. Such learned kernels would implement a true data model and could thus realize far more interesting and useful operations than shown here. And lastly, the ad

hoc way of choosing input transfer functions for boosting input strength to the region where they will actually create activity in subsequent layers needs evidently to be replaced by an automatic adaptation process of a slow, homeostatic nature as, e.g., outlined in[28] and implemented in[27].

C. Outlook

More in the long term, another interesting avenue of research could be to investigate *sampling*[29] with the presented architecture. Instead of excitatory feedback that strengthens lower-level patterns, inhibitory feedback could also remove them, thus eliminating the most probable interpretation under the data model and allow others to be represented. In this way, it is conceivable to traverse a cascade of interpretations at each hierarchy level in descending order from very probable to improbable, realizing in a single structure a representation of all relevant input interpretations and thus coming very close to a truly Bayesian treatment of information by neural models.

REFERENCES

[1] C Faubel and G Schöner. Learning to recognize objects on the fly: a neurally based dynamic field approach. *Neural Networks*, 21(4):562–576, May 2008.

[2] AL Yuille and HH Bulthoff. Bayesian decision theory and psychophysics. In *In Perception as Bayesian Inference*, pages 123–161. University Press, 1996.

[3] A Gepperth. Processing and transmission of confidence in recurrent neural hierarchies. *Neural Processing Letters*, 2013.

[4] RS Zemel, P Dayan, and A Pouget. Probabilistic interpretation of population codes. *Neural Comput*, 10(2):403–430, Feb 1998.

[5] WJ Ma, J Beck, P Latham, and A Pouget. Bayesian inference with probabilistic population codes. *Nature Neuroscience*, 9(11), 2006.

[6] RH Cuijpers and W Erlhagen. Implementing Bayes' rule with neural fields. In *International Conference On Artificial Neural Networks (ICANN)*. Springer-Verlag, 2008.

[7] R Rao. Bayesian computation in recurrent neural circuits. *Neural Comput*, 16(1):1–38, Jan 2004.

[8] J Gold and M Shadlen. Neural computations that underlie decisions about sensory stimuli. *Trends Cogn Sci*, 5(1):10–16, Jan 2001.

[9] S Kubotan and K Aihara. Anayzing global dynamics of a neural field model. *Neural Processing Letters*, 21, 2005.

[10] W Erlhagen and G Schöner. Dynamic field theory of movement preparation. *Psychological review*, 109(3):545, 2002.

[11] D Osipova, A Takashima, R Oostenveld, G Fernandez, E Maris, and O Jensen. Theta and gamma oscillations predict encoding and retrieval of declarative memory. *Journal of neuroscience*, 2006.

[12] DC Knill and A Pouget. The Bayesian brain: the role of uncertainty in neural coding and computation. *Trends Neurosci*, 27(12):712–719, Dec 2004.

[13] MW Oram, D Xiao, B Dritschel, and KR Payne. The temporal resolution of neural codes: does response latency have a unique role? *Philos Trans R Soc Lond B Biol Sci*, 357(1424):987–1001, Aug 2002.

[14] DS Reich, F Mechler, and JD Victor. Temporal coding of contrast in primary visual cortex: when, what, and why. *J Neurophysiol*, 85(3):1039–1050, Mar 2001.

[15] R Kiani, H Esteky, and K Tanaka. Differences in onset latency of macaque inferotemporal neural responses to primate and non-primate faces. *J Neurophysiol*, 94(2):1587–1596, Aug 2005.

[16] T Michelet, GH Duncan, and P Cisek. Response competition in the primary motor cortex: Corticospinal excitability reflects response replacement during simple decisions. *Journal of Neurophysiology*, 104(1), 2010.

[17] E Hazeltine, RA Poldrack, and JDE Gabrieli. Neural activation during response competition. *Journal of Cognitive Neuroscience, Supplement 2*, pages 118–129, 2000.

[18] R Borowsky and MEJ Masson. Semantic ambiguity effects in word identification. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 22(1):63, 1996.

[19] S Timotheou. The random neural network: A survey. *The Computer Journal*, 53(3):251–267, 2010.

[20] H Jaeger. Adaptive nonlinear system identification with echo state networks. In *Advances in Neural Information Processing Systems*, 2003.

[21] S-I Amari. Mathematical foundations of neurocomputing. *Proceedings of the IEEE*, 78(9):1441–1463, 1990.

[22] P Cisek. Integrated neural processes for defining potential actions and deciding between them: a computational model. *J Neurosci*, 26(38):9761–9770, Sep 2006.

[23] JS Johnson, JP Spencer, and G Schöner. Moving to higher ground: The dynamic field theory and the dynamics of visual cognition. *New Ideas Psychol*, 26Kamps(2):227–251, Aug 2008.

[24] C Wilimzig, S Schneider, and G Schöner. The time course of saccadic decision making: dynamic field theory. *Neural Netw*, 19(8):1059–1074, Oct 2006.

[25] NP Rougier and J Vitay. Emergence of attention within a neural population. *Neural Netw*, 19(5):573–581, Jun 2006.

[26] Gustavo Deco and Edmund T Rolls. A neurodynamical cortical model of visual attention and invariant object recognition. *Vision Res*, 44(6):621–642, Mar 2004.

[27] M Garcia Ortiz and A Gepperth. Neural self-adaptation for large-scale system building. In *First International Conference on Cognitive Neurodynamics*, 2009.

[28] GG Turrigiano and SB Nelson. Homeostatic plasticity in the developing nervous system. *Nat Rev Neurosci*, 5(2):97–107, Feb 2004.

[29] J Fiser, P Berkes, G Orbàn, and M Lengyel. Statistically optimal perception and learning: from behavior to neural representations. *Trends Cogn Sci.*, 2010.