Neural network trained with supervision represents uncertainty by nonlinear moments

Li Wenliang, Maneesh Sahani

Summary Making optimal inferences about the state of the world from noisy sensory information requires that accurate and flexible representations of the concomitant uncertainty be learnt. How might this happen? It is obvious that supervised learning from noisy inputs must retain some information about uncertainty to perform optimally (e.g. [3]), but it is unclear whether such a representation will be flexible or generic enough to underpin general probabilistic computation. Here, we analyse the representation of uncertainty that arises through supervised learning in a network tuned to propagate probabilistic messages using the recently proposed distributed distributional coding (DDC) scheme. Following previous work [4, 6], the DDC assumes that neurons represent uncertainty through the expectations of pre-specified basis functions under the encoded distribution. DDCs can be used to generate state-of-the-art performance in unsupervised learning of intractable models using a biologically plausible representation of uncertainty. We trained recurrent neural networks (RNNs) to estimate the posterior mean of a non-linear dynamical system without explicitly enforcing a DDC-like representation. Nonetheless, the RNN in which propagation was consistent with DDC message passing performed better than other networks, and its hidden units preserved more information about the posterior variance. Indeed, we found that activities in the hidden layer of this RNN could be interpreted as posterior expectations of functions over the latent variables; these functions did well not only in predicting the hidden activities, but also in reconstructing the posterior distributions. Thus, we conclude that flexible DDC-like codes for uncertainty are learnt naturally within networks of the suitable architecture.

Additional detail We generated data from a state-space model (Fig. 1A) with zero-mean linear-Gaussian latent variables $\mathbf{y}_{1:T}$ and generalised linear Poisson observations $\mathbf{x}_{1:T}$, and trained neural networks to predict the posterior mean $\mathbb{E}(\mathbf{y}_t|\mathbf{x}_{1:t})$. Although this supervised training objective is not explicitly probabilistic, optimal computation requires that prediction of $\mathbf{y}_t$ based on past observations $\mathbf{x}_{1:t-1}$ be combined with information from the current observation $\mathbf{x}_t$ in a manner that respects the uncertainties in each. Thus, an optimal network will need to have learnt *implicit* representations of uncertainty. Our approach differs from that used with probabilistic population codes [2] or the neural engineering framework [1] which provides neural implementations of closed-form equations to *explicitly* carry out uncertainty estimation.

All networks have three levels of neurons shown in Fig. 1(B-E). The observations $\mathbf{x}$ are first transformed into a *feature* representation $\boldsymbol{\phi}_t = \tanh(\mathbf{W} \cdot \mathbf{x}_t + \boldsymbol{b})$. Each $\boldsymbol{\phi}_t$ is combined with the previous *hidden* representation $\mathbf{h}_{t-1}$, which contains the message from all observations up to time $t-1$, to form a new representation $\mathbf{h}_t$ (see below for different combination rules). The latent prediction $\hat{\mathbf{y}}_t$ is read out linearly from $\mathbf{h}_t$. The loss function is the mean squared error (MSE) between the prediction and the true latent values $\frac{1}{T}\sum_{t=1}^{T} \|\hat{\mathbf{y}}_t - \mathbf{y}_t\|^2$ which drives the RNN to learn to predict the posterior mean of the latent variables.

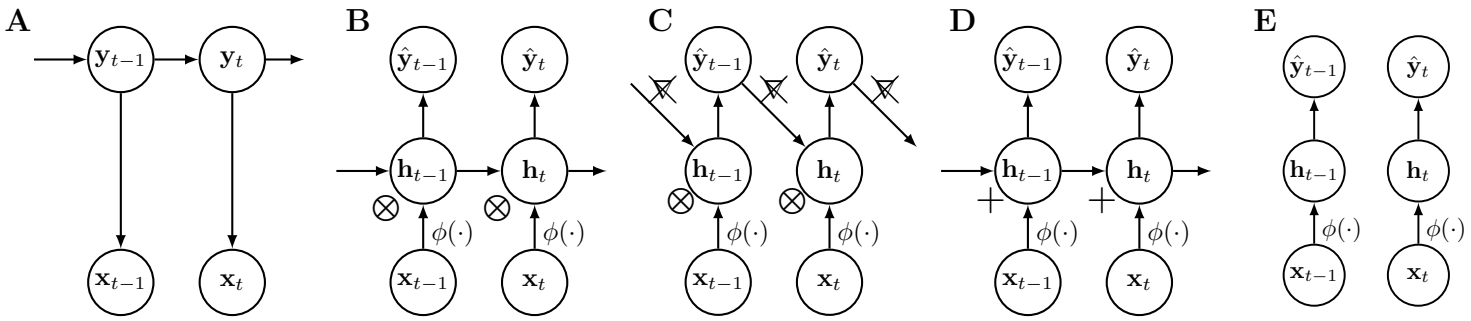

Figure 1: (**A**) Generative model of the state-space model, arrows represent conditional distributions. (**B-E**) Networks, arrows represent functions. (**B**) TensorRNN, $\otimes$ indicates tensor product between $\mathbf{h}_{t-1}$ and $\boldsymbol{\phi}(\mathbf{x})$. (**C**) MeanRNN, $\not\otimes$ indicates no backward gradients while forward connection is intact. (**D**) AddRNN, $+$ indicates addition between $\mathbf{h}_{t-1}$ and $\boldsymbol{\phi}(\mathbf{x})$. (**E**) StaticNN, no temporal structure.

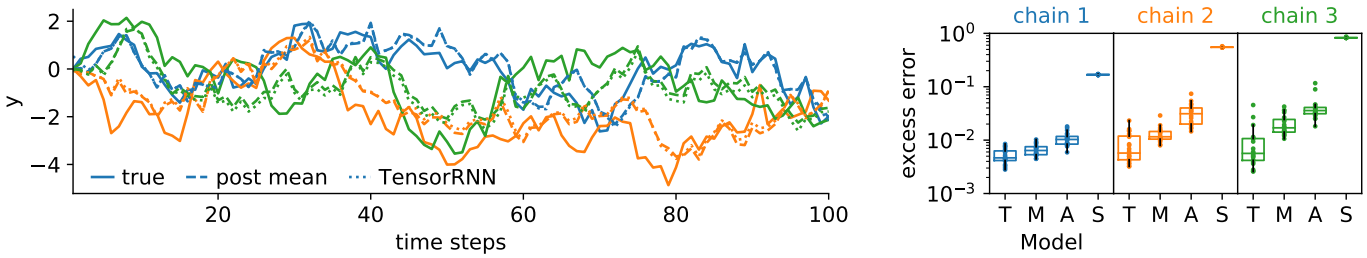

Figure 2: Performance of the models. Left, examples of latent chains (true), posterior means by particle filter and TensorRNN. Right, test MSE in excess of the particle filter for different models and latent chains. Differences between models for each chain were all significant ($p < 0.01$).

The networks differ in how messages are propagated. DDC message passing is borrowed from kernel belief propagation [5]: $h_{t,i} = \sum_{j,k} V_{i,j,k} h_{t-1,j} \phi_{t,k} + c_i$ [1], giving *TensorRNN* (Fig. 1B). The *MeanRNN* (Fig. 1C) propagates message from $\hat{\mathbf{y}}_{t-1}$ instead of $\mathbf{h}_{t-1}$. During training, the gradients from future $\mathbf{h}_{t:T}$ are stopped from reaching the previous prediction $\hat{\mathbf{y}}_{t-1}$ to prevent additional information from forming in $\hat{\mathbf{y}}$. *AddRNN* (Fig. 1D) is the classical RNN where the feature representation $\boldsymbol{\phi}_t$ is added to $\mathbf{h}_t$. The last network, *StaticNN* (Fig. 1E), completely ignores the history and uses only the current observation to predict the latent variable.

In our experiment, the observations $\mathbf{x}$ were most informative about chain 1 and least informative about chain 3. The hidden representations $\mathbf{h}$ in all networks had 5 dimensions. After training, we tested the networks on 1000 new random draws from the state-space model. A near-optimal posterior mean was found by particle filtering with 3000 particles. Examples of the true latent chains and posterior mean estimates from the particle filter and TensorRNN are shown in Fig. 2 left. The MSE of TensorRNN was significantly lower than the other models ($p < 0.01$, Welch's t-test, Fig. 2 right).

The results so far suggest that the hidden representations in TensorRNN may contain more uncertainty information than the others for message passing. A naive way to verify this is to linearly decode the posterior variance from the hidden units, for which TensorRNN outperformed the others (Table 1). Following DDC, we assumed that the hidden units represent the posterior distribution by its expectations of some functions over the latent variables $h_{t,i} = \mathbb{E}[f_i(\mathbf{y}_t)|\mathbf{x}_{1:t}]$ and estimated these functions $f_i(\cdot)$ by kernel function approximation. Examples of such functions are shown in Fig. 3. Test performance of these functions in predicting the latent representations reached $R^2 > 0.95$ in all models except StaticNN ($R^2 < 0.8$). Furthermore, we quantified the effectiveness of these functions by measuring how well the posterior mean embeddings can be reconstructed using these functions. Indeed, the functions learned by TensorRNN had the lowest reconstruction error compared to the other networks.

Network	chain 1	chain 2	chain 3
TensorRNN	0.932	0.885	0.834
MeanRNN	0.852	0.753	0.586
PlusRNN	0.879	0.733	0.594
StaticNN	0.801	0.427	0.189

Table 1: R^2 of posterior variance prediction from hidden representation $\mathbf{h}$.

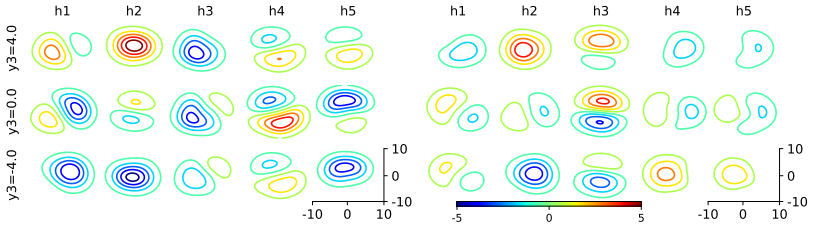

Figure 3: Functions found in TensorRNN (left) and MeanRNN (right). Contours are slices along the 3rd dimension of $\mathbf{y}$.

[1] C. Eliasmith and C. H. Anderson. MIT Press, Cambridge, MA, USA, 2002.
[2] W. J. Ma, J. M. Beck, P. E. Latham, and A. Pouget. *Nat Neurosci*, 9(11):1432–1438, Nov. 2006.
[3] A. E. Orhan and W. J. Ma. *Nature Communications*, 8(1):138, dec 2017.
[4] M. Sahani and P. Dayan. *Neural Comput.*, 15(10):2255–2279, Oct. 2003.
[5] L. Song, A. Gretton, D. Bickson, Y. Low, and C. Guestrin. In *AISTATS*, pages 707–715, 2011.
[6] R. S. Zemel, P. Dayan, and A. Pouget. *Neural Comput.*, 10(2):403–430, 1998.

[1] Equivalent to taking the outer product of $\mathbf{h}_t$ and $\boldsymbol{\phi}_t$, then contracted with $\mathbf{V}$ along the last two dimensions, add bias.