

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/313844935>

Estimating Nonlinear Dynamics with the ConvNet Smoother

Article · February 2017

CITATIONS

2

READS

42

4 authors, including:



Umut Güçlü

Radboud University

29 PUBLICATIONS 356 CITATIONS

[SEE PROFILE](#)



Eric Maris

Radboud University

96 PUBLICATIONS 7,564 CITATIONS

[SEE PROFILE](#)



Marcel van Gerven

Radboud University

142 PUBLICATIONS 1,854 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Latent response models [View project](#)



Bayesian Connectomics [View project](#)

Estimating Nonlinear Dynamics with the ConvNet Smoother

Luca Ambrogioni*, Umut Güçlü, Eric Maris and Marcel A. J. van Gerven

Radboud University, Donders Institute for Brain, Cognition and Behaviour, Nijmegen, The Netherlands

* l.ambrogioni@donders.ru.nl

Abstract

Estimating the state of a dynamical system from a series of noise-corrupted observations is fundamental in many areas of science and engineering. The most well-known method, the Kalman smoother (and the related Kalman filter), relies on assumptions of linearity and Gaussianity that are rarely met in practice. In this paper, we introduced a new dynamical smoothing method that exploits the remarkable capabilities of convolutional neural networks to approximate complex nonlinear functions. The main idea is to generate a training set composed of both latent states and observations from an ensemble of simulators and to train the deep network to recover the former from the latter. Importantly, this method only requires the availability of the simulators and can therefore be applied in situations in which either the latent dynamical model or the observation model cannot be easily expressed in closed form. In our simulation studies, we show that the resulting ConvNet smoother has almost optimal performance in the Gaussian case even when the parameters are unknown. Furthermore, the method can be successfully applied to extremely nonlinear and non-Gaussian systems. Finally, we empirically validate our approach via the analysis of measured brain signals.

1 Introduction

Estimating the state of a dynamical system from a finite set of indirect and noisy measurements is a key objective in many fields of science and engineering [1]. For example, meteorological forecasting requires the estimation of the dynamical state of a series of atmospheric variables from a sparse set of noisy measurements [2]. Another example, essential for several modern technologies, is the localization of physical objects from indirect measurements such as radar, sonar or optical recordings [1]. Even the human brain consistently deals with this problem as it has to integrate a heterogeneous series of indirect sensory inputs in its internal representation of the external world [3]. In the rest of the paper we will refer to this class of problems as *dynamical smoothing*, considering the related problem of *dynamical filtering* as a special case (i.e. 0-lag smoothing).

In this paper, we introduce a new nonlinear smoothing approach that only requires the ability to sample from measurements and dynamical models by leveraging the exceptional flexibility and representational capabilities of deep convolutional neural networks (ConvNets) [4]. The key idea is to generate synthetic samples of signal and noise from an ensemble of generators in order to train a deep neural network to recover the latent dynamical state from the observations. This allows the use of very realistic models, where the signal and noise structure can be tailored to the specific application without any concern about their analytic tractability. Furthermore, the procedure completely circumvents the problem of model selection and parameter estimation since the training set can be constructed by hierarchically sampling the model and its parameters from an appropriate ensemble. Importantly, since we can generate an arbitrarily large number of training data, we can train arbitrarily complex deep networks without over-fitting.

1.1 Related works

Conventional solutions to the dynamical smoothing problem usually rely on a series of mathematical assumptions on the nature of the signal and the measurement process. For example, when the state dynamic is linear, the measurement model is Gaussian and all the parameters are known, the optimal solution is given by the Kalman smoother (also known as the Rauch–Tung–Striebel smoother) [5]. For nonlinear state dynamics and/or a non-Gaussian measurement model, the dynamical smoothing problem can no longer be solved exactly. In these cases a common approximation is the extended Kalman smoother (EKS), which works by linearizing the state dynamics and the measurement model at each time point [6]. A more modern approach is the unscented Kalman smoother (UKS) that approximates the dynamical smoothing distribution by passing a set of selected points (sigma points) through the exact nonlinear functions of the state dynamics and the measurement model [7]. Unfortunately, these methods may introduce systematic biases in the estimated state and require the availability of both a prior and a likelihood function in closed form. In theory these shortcomings can be overcome by using sampling methods such as particle smoothers [1]. However, these methods require a large number of samples in order to be reliable and are affected by particle degeneration.

In recent years, several authors pioneered the use of deep neural networks and stochastic optimization on dynamical filtering and smoothing problems. Haarnoja and collaborators introduced the backprop KF, a deterministic filtering method based on a low-dimensional differentiable dynamical model whose input is obtained from high-dimensional measurements through a deep neural network [8]. Importantly, all the parameters of this model can be optimized using stochastic gradient descent. Improvements in stochastic variational inference led to several applications to dynamical filtering and smoothing problems. Using variational methods, Krishnan and collaborators introduced the deep Kalman filters [9]. In this work, the authors assume that the distribution of the latent states is Gaussian with mean and covariance matrix determined from the previous state through a parameterized deep neural network. The parameters of the network are trained using stochastic backpropagation [10]. Similarly, Archer and collaborators introduced the use of stochastic black-box variational inference as a tool for approximating the posterior distribution of nonlinear smoothing problems [11]. Despite the impressive performance of modern variational inference techniques, these methods share with the more conventional methods such as EKF and UKF the problem of introducing a systematic bias due to the constraints in the model and in the variational distribution. Conversely, the bias of our method can be arbitrarily reduced since neural networks are universal function approximators and we can use a limitless number of training samples.

2 The model

A state space model is defined as a dynamic equation that determines how the latent state evolves through time, and a measurement model, that determines the outcome of measurements performed on the latent state. If we assume that the measurement process does not affect the latent state, a general stochastic state model can be expressed as follows

$$x(t) = F_\theta[x(0), \dots, x(t-1); \xi_t] \quad (1)$$

$$y(t) = G_\phi[x(0), \dots, x(t); \zeta_t] \quad (2)$$

where $x(t)$ is the latent state at time t and $y(t)$ is the measurement at time t . The functions F_θ and G_ϕ determine the dynamic evolution of the latent state and the measurement process respectively. Since the system is causal and is not disturbed by the measurements, the dynamic function F_θ takes as input the past values of x and a random variable ξ_t that introduces stochasticity in the dynamics. Analogously, the measurement function G_ϕ takes as inputs the past and present values of x and a random variable ζ_t that accounts for the randomness in the measurements.

In a dynamical smoothing problem, we aim to recover the latent state x from the set of measurements $\{y(0), \dots, y(T)\}$, where T is the final time point. Usually, the functional form of F_θ

and G_ϕ is assumed to be known in advance. Nevertheless, these functions typically depend on a set of parameters (θ and ϕ respectively) that need to be inferred from the data.

2.1 Deterministic smoothing as a regression problem

We will focus on deterministic smoothing, where the output of the analysis is a point estimate of x instead of a full probability distribution. If we have a training set where both the dynamical state x and the measurements y are observable, then the problem reduces to a simple regression problem where we construct a deterministic mapping $f(y; w) = \hat{x}$ by minimizing a suitable cost function of the form

$$C(w) = \sum_j D[x(t_j), f(y(t_j); w)] , \quad (3)$$

over the space of some parameters w . In this expression, the function $D[x, \hat{x}]$ is some sensible measure of deviation between the real dynamical state x and our estimate \hat{x} .

In most situations the state x is not directly observable. However, in the usual dynamical smoothing setting, we assume to have access to the dynamical model that generated the data and the measurements. In this case, the model can simply be used for generating a synthetic training set. If the dynamical model is not exactly known, we can still construct a training set by sampling from a large parametrized family of plausible models. Clearly, this requires the use of a sophisticated regression model that is able to learn very complicated functional dependencies. To this end, we make use of convolutional neural networks, a regression (and classification) method, which has been shown to achieve state-of-the-art performance in many machine learning tasks [12, 13, 14].

2.2 Convolutional neural networks

In this subsection, we briefly describe the details of the convolutional neural network which was used in our experiments. The network comprised a number of dilated convolution layers [15] with 60 one-dimensional kernels of length three and rectified linear units, followed by a fully-connected layer with n one-dimensional kernels of length n where n is the signal length. In most experiments, n was 200.

Dilated convolution layers are similar to regular convolution layers with the exception that successive kernel elements have holes between each other, whose size is determined by a dilation factor. As a result, they ensure that the feature map length remains the same as the receptive field length increases. Note that regular convolution layers can be considered dilated convolution layers with a dilation factor of one.

The dilation factor of the first two layers were one, which doubled after every subsequent layer. The number of convolution layers was chosen to be the largest possible value such that the receptive field length of the last convolution layer was less than the signal length. That is,

$$m = \arg \max_x \left(3 + 2 \sum_{i=0}^{x-2} 2^i \right) < n \quad (4)$$

where m is the number of convolution layers. In all experiments, m equalled seven.

We initialized the bias terms to zero and the weights to samples drawn from a scaled Gaussian distribution [16]. We used Adam [17] with initial $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$ and a mini-batch size of 1500 to train the network for 20,000 epochs by iteratively minimizing a smooth approximation of the Huber loss function [18], called the pseudo-Huber loss function [19]:

$$C(w) = \sum_j \sqrt{1 + (x(t_j) - f(y(t_j); w))^2} - 1. \quad (5)$$

2.3 The ConvNet smoother

The idea behind the ConvNet smoother is simple. We train a convolution neural network to recover the sequence of simulated dynamical states $\{x(0), \dots, x(T)\}$ from the set of simulated measurements $\{y(0), \dots, y(T)\}$. The network is trained on simulated data that are sampled using the state evolution function F_θ and the measurement function G_ϕ . If we do not know the parameters θ and ϕ in advance, the training set can still be constructed by randomly sampling θ and ϕ prior to each sample of $\{x(0), \dots, x(T)\}$ and $\{y(0), \dots, y(T)\}$. In this way, we leave to the network the burden of adapting to the specific parameter setting every time a new series of observations is presented as input. Clearly, if the parameter space is large, we would need a more complex network in order to properly learn the mapping. Fortunately, since we can generate an arbitrarily large number of data points, we can potentially train any complex network without overfitting on the training set.

3 Experiments

In the following, we validate the ConvNet smoother in two simulation studies and in an analysis of real brain data acquired using magnetoencephalography (MEG).

3.1 Analysis of Gaussian dynamics

When the latent dynamical state is a Gaussian process (GP) with known covariance function, the optimal smoother (in a least-squares sense) is given by the expected value of a GP regression [20]. Here, we compare the performance of the ConvNet smoother with the expectation of both an exact GP regression (with known covariance and noise parameters) and an optimized GP regression (where the parameters are obtained by maximizing the marginal likelihood of the model given the data). The ConvNet was trained with samples drawn from a GP equipped with a squared exponential covariance function [20]. For each sample, the length scale and the amplitude parameters were sampled from a log-normal distribution (length scale: $\mu = -1.9$, $\sigma = 0.8$; amplitude: $\mu = 0$, $\sigma = 0.5$). Hence, the ConvNet smoother is effectively trained on a large family of GPs. These samples were then corrupted with Gaussian white noise whose standard deviation was itself sampled from a log-normal distribution ($\mu = -0.9$, $\sigma = 0.5$). The network was trained to recover the ground truth function from the noisy data.

Figure 1 shows the results on a test set comprised of 200 trials. Panel A shows an example trial. The estimate obtained using the ConvNet smoother is less smooth than the GP estimates but it does a good job at tracking the ground truth signal. Panel B shows the absolute deviations of the ConvNet, exact GP and optimized GP models. The performance of the ConvNet is only slightly worse than the (optimal) exact GP. Furthermore, the ConvNet significantly outperforms the GP optimized by maximal likelihood.

3.2 Analysis of nonlinear dynamics

In this section, we show that the ConvNet smoother can be used in complex situations where neither the exact likelihood or the exact prior can be easily expressed in closed form. This freedom allows to train the network on a very general noise model that is likely to approximate the real noise structure of the measured data as a special case.

As dynamical model, we used the following stochastic anharmonic oscillator equation:

$$\frac{d^2x}{dt^2} = -\omega_0^2 x - \beta \frac{dx}{dt} + k_2 x^2 + k_3 x^3 + \xi(t), \quad (6)$$

where ω_0 is the asymptotic undamped angular frequency for small oscillations, β is the damping coefficient and both k_2 and k_3 regulate the deviation from harmonicity. The term $\xi(t)$ is additive Gaussian white noise and introduces stochasticity to the dynamics. We discretized the stochastic

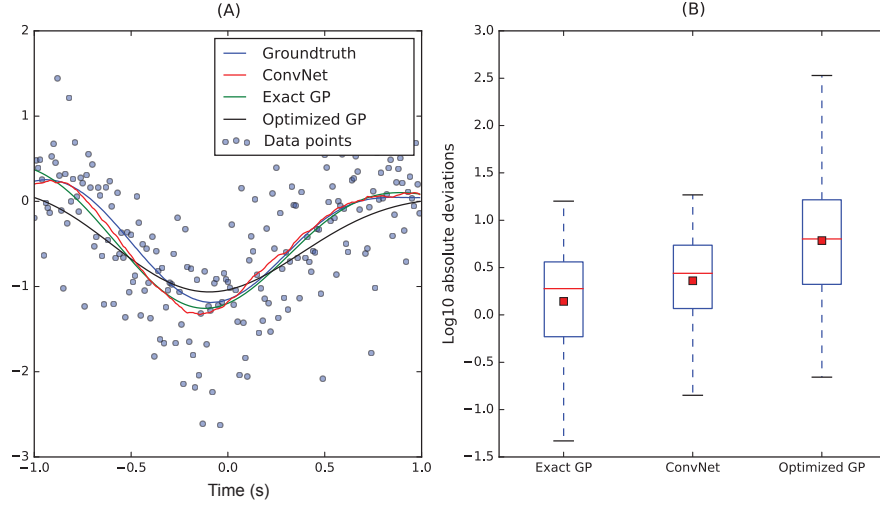


Figure 1. Analysis of Gaussian signals. A) Analysis of an example signal. B) Performance of ConvNet, exact GP and optimized GP. The boxplot shows the absolute deviation between reconstructed signal. The red lines show the median while the red squares show the mean.

dynamical process using the Euler-Maruyama scheme with integration step equal to 0.01 seconds. The parameters of the dynamical model were kept fixed ($\omega_0 = 5$, $\beta = 0.2$, $k_2 = 15$, $k_3 = -0.5$). This procedure gave a total of $N = 200$ time points for each simulated trial. The experiment is divided in two parts. In the first, we used a Gaussian observation model and we compared our method with existing dynamical smoothing techniques. In the second, we used a more complex parameterized observation model in order to demonstrate the flexibility of the ConvNet method.

3.2.1 Gaussian observation model

In this first part of the experiment, we use an observation model where the measurements are corrupted by Gaussian white noise. The standard deviation of the measurement noise was equal to 20. We generated a total of 49900 training pairs ($\{x(t_1), \dots, x(t_N)\}, \{y(t_1), \dots, y(t_N)\}$) and 1000 test pairs. We compared the ConvNet smoother with EKS and UKS.

Figure 2, Panel A shows the latent state of an example trial recovered using the ConvNet smoother, EKS and UKS. The absolute deviations between the recovered latent state and the ground truth signal are shown in Panel B. From the boxplots, it is clear that in this experiment the ConvNet greatly outperforms the other methods.

3.2.2 Conditionally dependent observation model

In this experiment, we use a complex observation model where the measurements are not statistically independent given the latent state. In these situations, EKS and UKS cannot be directly applied. The observations $y(t)$ were obtained as follows:

$$y(t) = x(t) + \eta t + \gamma(t) + \zeta(t), \quad (7)$$

where ηt is a linear trend with slope η sampled at random from a normal distribution with zero mean and standard deviation equal to 10; $\gamma(t)$ is a pure jump process with exponential inter-jump interval (mean equal to 0.5 s) and scaled Cauchy jump size (scale equal to 1.5); $\zeta(t)$ is Student-t white noise with scale sampled from a gamma distribution (with scale equal to 0.3, shape equal to 2) and degrees of freedom sampled from a uniform distribution over the integers from 2 to 21. All the

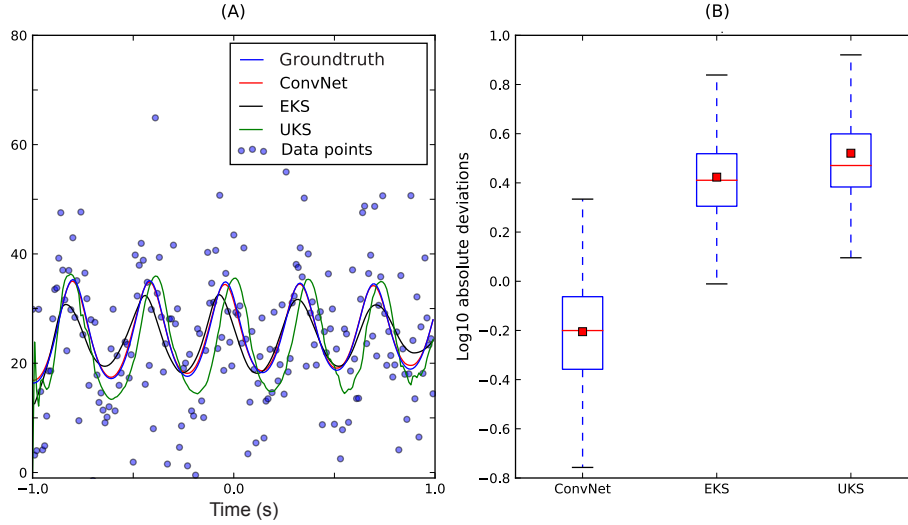


Figure 2. Analysis of nonlinear signals corrupted by conditionally independent noise. A) Analysis of an example signal. B) Performance of ConvNet, EKF and UKF. The boxplot shows the log10 of the absolute deviation between reconstructed signal. The red lines show the median while the red squares show the mean.

parameters of the noise model were sampled every time a new trial was generated. We generated a total of 99900 training pairs $(\{x(t_1), \dots, x(t_N)\}, \{y(t_1), \dots, y(t_N)\})$ and 1000 test pairs.

Figure 3, Panels A–D shows the results in the test set for four example trials. We can see that these trials are characterized by highly heterogeneous waveforms of the latent dynamical process and very variable noise structure. Visually, the method seems to maintain high performance for a wide range of signal and noise characteristics. For example, Panel B shows a trial with a large discontinuous jump while Panel C shows a trial with very pronounced outliers. The median log10 absolute deviation of the model output from the ground truth dynamical signal was 0.32 while the upper and lower quantiles were -0.51 and 1.38 respectively.

3.3 Analysis of brain oscillations

As a final example, we used the ConvNet smoother for recovering 10 Hz brain oscillations (the so-called alpha rhythm; see [21]) from MEG measurements. While several neural field equations have been proposed [22], so far there is not a universally accepted dynamical model of cortical electromagnetic activity. Therefore, for this application, we generated the dynamical latent state using an empirical procedure that is meant to capture the qualitative features of alpha oscillations without resorting to an explicit equation of motion. The idea is to sample from a sufficiently large family of signal and noise models, in order to capture the observed data as special case. After training, the ConvNet should be able to recognize the appropriate signal and noise characteristics directly from the input time series.

In our example, the oscillatory waveform was generated as follows:

$$x(t) = \mathcal{A}(t)f\left(\cos(\omega(t)t + \phi_0)\right), \quad (8)$$

where the envelope $\mathcal{A}(t)$ and the angular frequency parameter $\omega(t)$ are sampled from a GP and the initial phase ϕ_0 is sampled from a uniform distribution. Furthermore, the nonlinear function $f(a)$ has the following form:

$$f(a) = w_1a + w_2a^2 + w_3a^3 + w_4a^4 + w_5a^5, \quad (9)$$

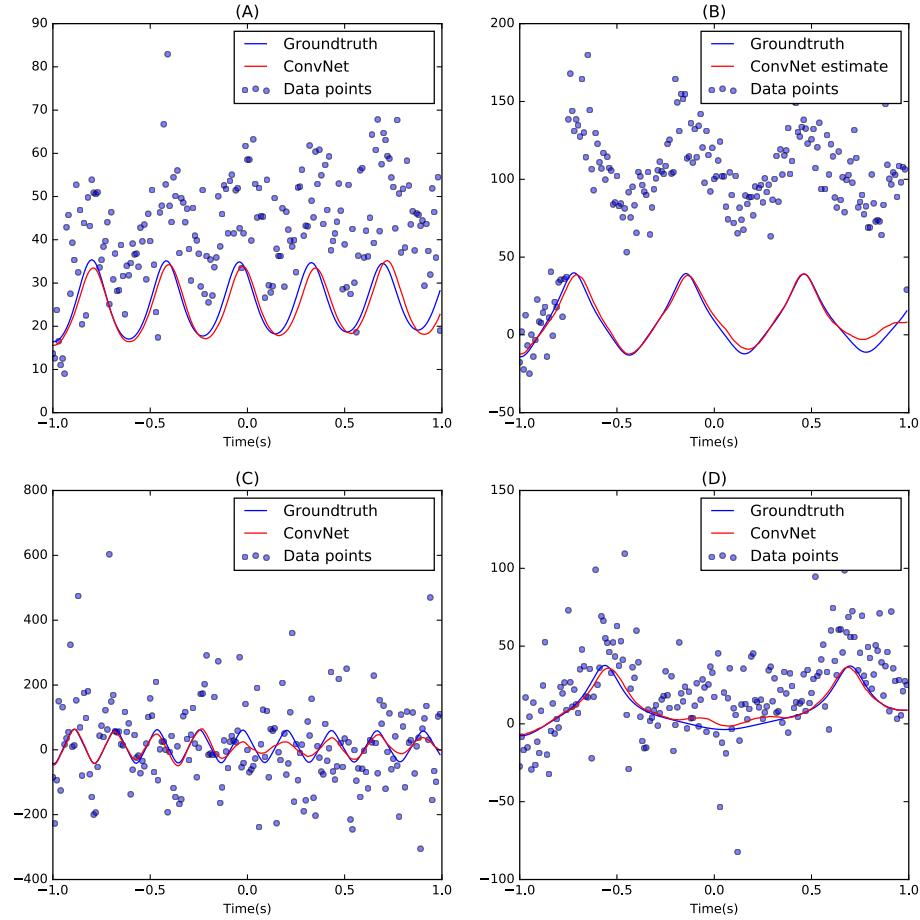


Figure 3. Analysis of nonlinear signals corrupted by conditionally dependent noise. Panels A to D show the input data points (blue dots), ground truth signal (blue line) and ConvNet estimates (red line) for four example trials.

where the Taylor coefficients w_1, w_3 and w_5 were sampled from truncated t distributions ($df = 3$, from 0 to ∞) and the coefficients w_2 and w_4 were sampled from t distributions ($df = 3$). This allows to generate synthetic alpha oscillations with variable waveform, amplitude and peak frequency.

The network was trained on the synthetic data set and then applied to a resting-state MEG data set. The experimental procedure is described in [23]. We compared the resulting estimate with the estimate obtained by applying a band-pass Butterworth (two-pass, 4th order, from 8 Hz to 12 Hz) filter to the MEG data. Figure 4 shows the result in two example trials. In Panel A, the oscillatory alpha activity is very prominent across the whole trial. Note that the ConvNet smoother is able to recover the highly anharmonic waveform without introducing a substantial amount of noise. In Panel B, we can see that the oscillatory activity is absent in the first half of the trial and becomes prominent in the second half. Importantly, the ConvNet smoother almost completely suppresses the oscillatory response in the first part while the linear filter exhibits a low amplitude oscillation.

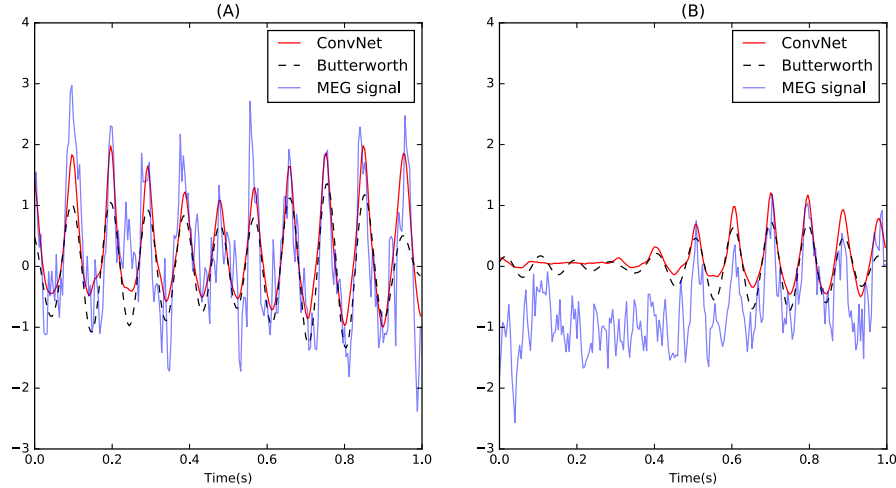


Figure 4. Analysis of brain oscillations. Panels A-B show the raw MEG signal (blue line) and the estimate of the alpha oscillations time series obtained using Butterworth filter (dashed line) and ConvNet smoother (red line) for two example trials.

4 Conclusions

In this paper, we introduced the use of deep convolution neural networks trained on synthetic data for nonlinear smoothing. The ConvNet smoother requires no analytic work by the practitioner besides the design of an appropriate ensemble of signal and noise simulators. Importantly, imperfect prior knowledge about the signal and the noise model is compensated by the remarkable capacity of deep convolutional neural networks to recognize patterns in the data.

Several improvements are possible. First, the model can easily be used for forecasting by training the network with an initial segment of noisy time series as input and the full noise-free time series as output. This kind of application could have a major impact given the importance of ensemble forecasting methods in fields such as meteorology [2]. Second, the method can be adapted to online filtering by using an autoregressive convolutional architecture where the filter kernels only have access to previous time points [24]. Third, the uncertainty over the latent dynamical signal can be estimated either by drop-out [25] or by using a conditional density estimation neural network for estimating the full conditional distribution of the dynamical state given the data [26]. This latter approach may be considered as an application of the recently introduced framework for Bayesian conditional density estimation [27].

References

1. S. Särkkä, *Bayesian Filtering and Smoothing*. Cambridge University Press, 2013.
2. T. N. Krishnamurti, V. Kumar, A. Simon, A. Bhardwaj, T. Ghosh, and R. Ross, “A review of multimodel superensemble forecasting for weather, seasonal climate, and hurricanes,” *Reviews of Geophysics*, vol. 54, no. 2, pp. 336–377, 2016.
3. C. D. Mathys, E. I. Lomakina, J. Daunizeau, S. Iglesias, K. H. Brodersen, K. J. Friston, and K. E. Stephan, “Uncertainty in perception and the hierarchical gaussian filter,” *Frontiers in Human Neuroscience*, vol. 8, p. 825, 2014.

4. J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.
5. M. Briers, A. Doucet, and S. Maskell, "Smoothing algorithms for state-space models," *Annals of the Institute of Statistical Mathematics*, vol. 62, no. 1, pp. 61–89, 2010.
6. H. W. Sorenson, *Kalman filtering: Theory and Application*. IEEE, 1985.
7. E. A. Wan and R. van der Merwe, "The unscented Kalman filter for nonlinear estimation," *Adaptive Systems for Signal Processing, Communications, and Control Symposium*, pp. 153–158, 2000.
8. T. Haarnoja, A. Ajay, S. Levine, and P. Abbeel, "Backprop KF: Learning discriminative deterministic state estimators," *Advances in Neural Information Processing Systems*, pp. 4376–4384, 2016.
9. R. G. Krishnan, U. Shalit, and D. Sontag, "Deep kalman filters," *arXiv preprint arXiv:1511.05121*, 2015.
10. D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," *arXiv preprint arXiv:1401.4082*, 2014.
11. E. Archer, I. M. Park, L. Buesing, J. Cunningham, and L. Paninski, "Black box variational inference for state space models," *arXiv preprint arXiv:1511.07367*, 2015.
12. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, pp. 1097–1105, 2012.
13. K. E. X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.
14. A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.
15. F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *arXiv:1511.07122*, 2015.
16. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.
17. D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv:1412.6980*, 2014.
18. P. J. Huber, "Robust estimation of a location parameter," *The Annals of Mathematical Statistics*, vol. 35, no. 1, pp. 73–101, 1964.
19. P. Charbonnier, L. Blanc-Feraud, G. Aubert, and M. Barlaud, "Deterministic edge-preserving regularization in computed imaging," *IEEE Transactions on Image Processing*, vol. 6, no. 2, pp. 298–311, 1997.
20. C. E. Rasmussen, *Gaussian Processes for Machine Learning*. The MIT press, 2006.
21. J. L. Cantero, M. Atienza, and R. M. Salas, "Human alpha oscillations in wakefulness, drowsiness period, and rem sleep: Different electroencephalographic phenomena within the alpha band," *Clinical Neurophysiology*, vol. 32, no. 1, pp. 54–71, 2002.
22. G. Deco, V. K. Jirsa, P. A. Robinson, M. Breakspear, and K. Friston, "The dynamic brain: From spiking neurons to neural masses and cortical fields," *PLoS Computational Biology*, vol. 4, no. 8, 2008.

23. L. Ambrogioni and E. Maris, “Complex-valued Gaussian process regression for time series analysis,” *arXiv preprint arXiv:1611.10073*, 2016.
24. B. Uria, M. Côté, K. Gregor, I. Murray, and H. Larochelle, “Neural autoregressive distribution estimation,” *Journal of Machine Learning Research*, vol. 17, no. 205, pp. 1–37, 2016.
25. Y. Gal and Z. Ghahramani, “Dropout as a Bayesian approximation: Representing model uncertainty in deep learning,” *arXiv preprint arXiv:1506.02142*, vol. 2, 2015.
26. P. M. Williams, “Using neural networks to model conditional multivariate densities,” *Neural Computation*, vol. 8, no. 4, pp. 843–854, 1996.
27. G. Papamakarios and I. Murray, “Fast ε -free inference of simulation models with Bayesian conditional density estimation,” *Advances in Neural Information Processing Systems*, pp. 1028–1036, 2016.