Neural Network Online Training with Sensitivity to Multiscale Temporal Structure

Matt Jones, David Mayo, Tyler R. Scott, Mengye Ren, Gamaleldin ElSayed, Katherine Hermann, Michael C. Mozer
mcjones@google.com

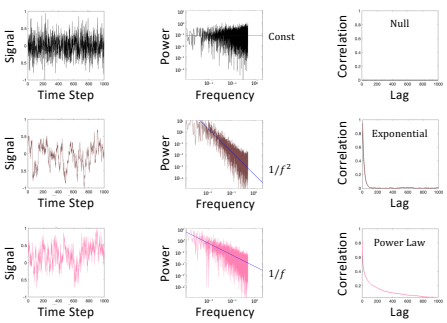
MemARI
NeurIPS 2022 | December 2nd
New Orleans, USA

Contributions

- Realistic nonstationarity as $1/f$ noise in data-generating parameters
- Kalman filter for Bayesian inference over $1/f$ noise
- Extended Kalman filter for NN optimization
- Efficient variational approximation, amounts to subweights with different learning and decay rates
- Variational multiscale EKF learns well in nonstationary tasks

Temporal structure in realistic environments

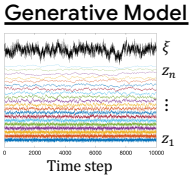
- Real environments have temporal structure (not iid)
- Have long-range autocorrelations (not diffusion or random walk)



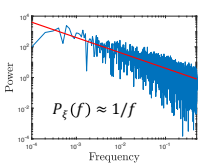
Bayesian inference over $1/f$ noise

- Generative model: sum of diffusion processes at different timescales
- Inferential model: Kalman filter over all processes jointly [KTS07]

Generative Model



$$\xi(t) = \sum_{i=1}^{n} z_i(t)$$



Ornstein-Uhlenbeck process: $\mathrm{d}z_i = -\tau_i^{-1} z_i \mathrm{d}t + \rho \tau_i^{-1/2} \mathrm{d}W(t)$

Timescales $\tau_i = \nu^i \ (\nu > 1)$

Inferential Model

Loss gradient: $\mathcal{L} = \frac{1}{2}(\hat{\xi} - \xi)^2$

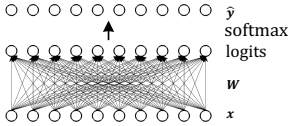
$$\boldsymbol{Z}(t)|\boldsymbol{\xi}_{\le t} \sim \mathcal{N}\big(\boldsymbol{\omega}(t), \boldsymbol{S}(t)\big)$$

$$\boldsymbol{\omega}(t+1) = \boldsymbol{D}\left(\boldsymbol{\omega}(t) - \frac{\boldsymbol{S}(t)}{\mathbf{1}^\top \boldsymbol{S}(t) \mathbf{1}} \mathbf{1}\left(\hat{\xi}(t) - \xi(t)\right)\right)$$

Decay: $\boldsymbol{D} = \mathrm{diag}\big(e^{-1/\boldsymbol{\tau}}\big)$ — Preconditioner

Evaluation in synthetic nonstationary tasks

- Linear regression
  - Data sampling: $y = \boldsymbol{x}^\top \boldsymbol{\beta}, \ \boldsymbol{\beta} \sim 1/f$ noise
  - Predictive model: $y = \boldsymbol{x}^\top \boldsymbol{\theta}$
- 10-way classification
  - Data sampling: $y \sim \mathrm{softmax}(\boldsymbol{l}), \boldsymbol{x}|y \sim \mathcal{N}(\boldsymbol{\mu}_y, \boldsymbol{I})$
    $\{\boldsymbol{l}, (\boldsymbol{\mu}_y)_y\} \sim 1/f$ noise
  - Predictive model:
    $\hat{\boldsymbol{y}} = \mathrm{softmax}(\boldsymbol{x}^\top \boldsymbol{W})$



Extended Kalman filter and variational inference

- Assume $1/f$ noise in optimal network parameters
- Extended Kalman filter uses Jacobian to linearize network
- Variational approximation tracks variance of each component

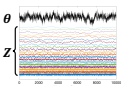
Predictive Model

$\boldsymbol{y} = h(\boldsymbol{x}, \boldsymbol{\theta})$

NN with weights $\boldsymbol{\theta} \in \mathbb{R}^m$

Generative Model

$\boldsymbol{\theta} = \sum \boldsymbol{z}_i, \ \boldsymbol{z}_i(t) \in \mathbb{R}^m$



Inferential Model

$$\boldsymbol{Z} = (\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n)$$

$$\boldsymbol{Z}|\boldsymbol{x}_{<t}, \boldsymbol{y}_{<t} \sim \mathcal{N}\big(\boldsymbol{\omega}(t), \boldsymbol{S}(t)\big)$$

Extended Kalman filter (EKF) [SW89,PF03]

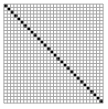
$$h(\boldsymbol{x}(t), \boldsymbol{\theta}) \approx \hat{\boldsymbol{y}} + \boldsymbol{J}_h\big(\boldsymbol{\omega}(t)\big)\big(\boldsymbol{\theta} - \boldsymbol{\omega}(t)\big)$$

$$\boldsymbol{y}|\hat{\boldsymbol{y}} \sim \mathcal{N}\big(\hat{\boldsymbol{y}}, \boldsymbol{R}_{\hat{\boldsymbol{y}}}\big), \quad \boldsymbol{R}_{\hat{\boldsymbol{y}}} = \mathrm{Var}(\boldsymbol{y}|\hat{\boldsymbol{y}})$$

Variational approximation

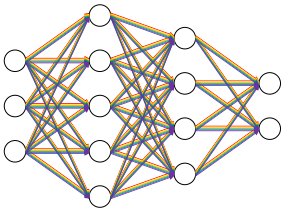
$$\boldsymbol{S}(t) \approx \mathrm{diag}\big(\boldsymbol{s}^2(t)\big)$$

Linear in NN size



Variational EKF optimizer

- Each weight decomposed as sum of subweights
- Subweight ω_{ij} is mean estimate of θ_j at timescale i (z_{ij})
- Decay rates come from $1/f$ generative model
- Learning rates come from EKF posterior variance



$$w_j = \sum_{i=1}^{n} \omega_{ij}$$

Timescale i
Weight j

Mean update

Adaptive learning rate

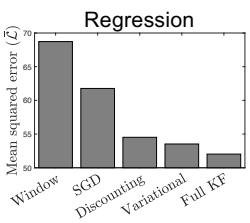
$$\omega_{ij}(t+1) = \underbrace{e^{-1/\tau_i}}_{\text{Decay}} \Big(\omega_{ij}(t) - \tilde{\boldsymbol{s}}_{ij}^2(t) \underbrace{\partial_{w_j} \tilde{\mathcal{L}}(t)}_{\text{Loss gradient}}\Big)$$

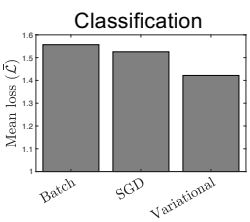
Variance update

$$\tilde{s}_{ij}^{-2}(t) = \underbrace{\big[\boldsymbol{J}^\top \boldsymbol{R}^{-1} \boldsymbol{J}\big]_{ij,ij}}_{\text{Fisher information}} + \underbrace{e^{-2/\tau_i}}_{\text{Decay}} \underbrace{\tilde{s}_{ij}^{-2}(t-1)}_{\text{Previous posterior}} + \underbrace{2\rho^2\big(1 - e^{-2/\tau_i}\big)}_{\text{New noise}}$$

Posterior precision — Prior precision

- Training methods
  - **Window:** use past H observations
  - **SGD:** update after each observation
  - **Discounting:** power-law weighting
  - **Variational EKF**
  - **Full EKF**





[KTS07] Konrad P. Kording, Joshua B. Tenenbaum, and Reza Shadmehr. The dynamics of memory as a consequence of optimal adaptation to a changing body. *Nature Neuroscience*, 10(6):779–786, 2007.

[PF03] Gintaras V Puskorius and Lee A Feldkamp. Parameter-based kalman filter training: Theory and implementation. In Simon Haykin, editor, *Kalman Filtering and Neural Networks*, pages 23–67. John Wiley & Sons, Inc., 2003.

[SW89] Sharad Singhal and Lance Wu. Training multilayer perceptrons with the extended Kalman algorithm. volume 1, 1989.