Deep Kalman Filters

Chizhov Nikita   Ekaterina Ivanova   Vadim Liventsev

Skolkovo Institute of Science and Technology

October 26, 2018

Kalman Filter

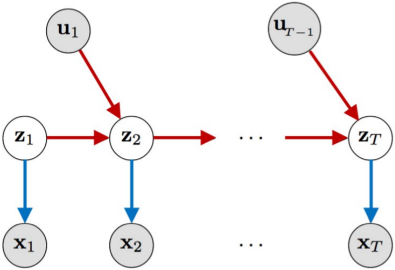


Figure: Model scheme

Model

Goal. Fit generative model to a sequence of observations and actions.

- Observations $\vec{x} = (x_1, \dots, x_T)$, $x_t \in \mathbb{R}^d$, noisy, non-linear function of latent space
- Actions $\vec{u} = (u_1, \dots, u_{T-1})$, $x_t \in \mathbb{R}^c$
- Latent space $\vec{z} = (z_1, \dots, z_T)$, $x_t \in \mathbb{R}^s$

$$z_1 \sim \mathcal{N}(\mu_0; \Sigma_0)$$
$$z_t \sim \mathcal{N}(G_\alpha(z_{t1}, u_{t1}, \delta_t), S_\beta(z_{t1}, u_{t1}, \delta_t))$$
$$x_t \sim \Pi(F_\kappa(z_t)).$$

$G_\alpha, S_\beta, F_\kappa$ are nonlinear functions of the previous state, parametrized by deep neural networks.
Distribution of the observations x_t are Multinomial.
Parameters of the generative model $\theta = \alpha, \beta, \kappa$.

Maximizing ELBO

We would like to fit generative model w.r.t. parameters θ.

$$\max_{\theta} log_{\theta} p(x_1, \dots, x_T | u_1, \dots, u_{T-1})$$

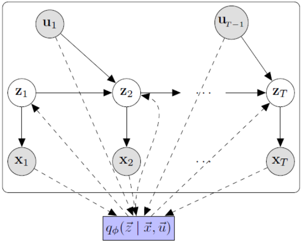
Using variational inference we reduce the problem to optimization of lower bound.

$$q_{\phi}((\vec{z})|\vec{x}, \vec{u}) = \prod_{t=1}^{T} q(z_t | z_{t-1}, x_1, \dots, x_T, \vec{u}) \sim \text{prior}$$

ELBO of conditional log likelihood:

$$\log p_{\theta}(\vec{x}|\vec{u}) \geq \mathcal{L}(x; (\theta, \phi)) =$$
$$\sum_{t=1}^{T} \mathop{\mathbb{E}}_{q_{\phi}(z_t|\vec{x},\vec{u})} [\log p_{\theta}(x_t|z_t)] - \mathrm{KL}(q_{\phi}(z_1|\vec{x}, \vec{u})||p_0(z_1))$$
$$- \sum_{t=2}^{T} \mathop{\mathbb{E}}_{q_{\phi}(z_{t-1}|\vec{x},\vec{u})} [\mathrm{KL}(q_{\phi}(z_t|z_{t-1}, \vec{x}, \vec{u})||p_0(z_t|z_{t-1}, u_{t-1}))] .$$

Algorithm



(a) Deep Kalman Filter

Algorithm 1 Learning Deep Kalman Filters

while $notConverged()$ **do**
- $\vec{x} \leftarrow sampleMiniBatch()$
- Perform inference and estimate likelihood:
- 1. $\hat{z} \sim q_\phi(\vec{z}|\vec{x}, \vec{u})$
- 2. $\hat{x} \sim p_\theta(\vec{x}|\hat{z})$
- 3. Compute $\nabla_\theta \mathcal{L}$ and $\nabla_\phi \mathcal{L}$ (Differentiating
- 4. Update θ, ϕ using ADAM

end while

(b) Algorithm for learning DKF

Generative networks

State transitions $z_t \rightarrow z_{t+1}$ are parametrized by a 2-layer stacked LSTM
Mean of normal distribution $p(\vec{x}|\vec{z})$ is parametrized by a multilayer perceptron
Mean of normal distribution $q(\vec{z}|\vec{x})$ is one of the following

- *q-INDEP*: $q(z_t|x_t, u_t)$ parameterized by an MLP
- *q-LR*: $q(z_t|x_{t1}, x_t, x_{t+1}, u_{t1}, u_t, u_{t+1]})$ parameterized by an MLP
- *q-RNN*: $q(z_t|x_1, \ldots, x_t, u_1, \ldots u_t)$ parameterized by a RNN
- *q-BRNN*: $q(z_t|x_1, \ldots, x_T, u_1, \ldots, u_T)$ parameterized by a bi-directional RNN

Dataset

For our project we used **Healing MNIST**: a synthetic dataset is obtained by producing sequence of rotated images.

- Actions $\vec{u}$ are the rotations
- Observations $\vec{x}$ are the rotated images.

We have *60000 of training sequences*, each sequence has *5 rotated images*.

To one sequence of three consecutive squares is superimposed with the top-left corner of the images.

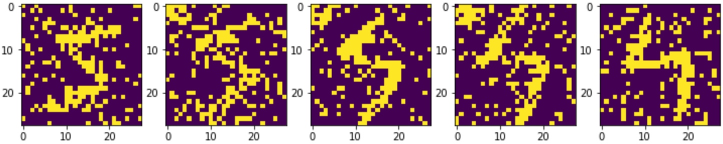


Figure: Sequence example.

Rotation encoding

In the reproduced paper authors don't specify how they represent a rotation with $\vec{u}$ [1]

We used an encoding scheme inspired by *Attention is all you need*

$$s_i = \sin(\frac{\pi}{2}\frac{\theta}{\theta_{max}} i), \;\; i \in [1, 25] \tag{1}$$

$$c_i = \cos(\frac{\pi}{2}\frac{\theta}{\theta_{max}} i), \;\; i \in [1, 25] \tag{2}$$

$$\vec{u} = (\vec{s}, \vec{c}) \tag{3}$$

[1] Maybe that's the secret sauce that makes their model work?

Results

We implemented

1. Healing MNIST
2. Kalman filter architecture proposed by the original authors
3. Stacked LSTM for the transitional model and 2 options (*q-INDEP* and *q-RNN*) for the inference model
4. Simple Variational Autoencoder
5. Simple Autoencoder

The original results *did not reproduce*. Simple autoencoder works best on Healing MNIST

Reconstructions

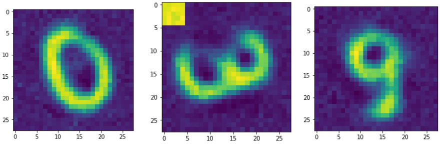


Figure: Simple autoencoder

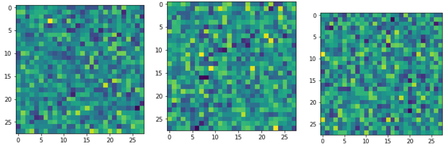


Figure: Deep Kalman filters, variational autoencoder

References

Krishnan, Rahul G., Uri Shalit, and David Sontag. "Deep kalman filters." *arXiv preprint arXiv:1511.05121 (2015)*

Krishnan, Rahul G., Uri Shalit, and David Sontag. "Structured Inference Networks for Nonlinear State Space Models." *AAAI. 2017.*