<탄생배경> Reviewer: 김정학

'Latent ODEs for Irregularly-Sampled Time Series'의 논문과 마찬가지로 이산적 특징을 가지고 있는 시계열데이터를 해결하기 위해서 나오게 됐다.

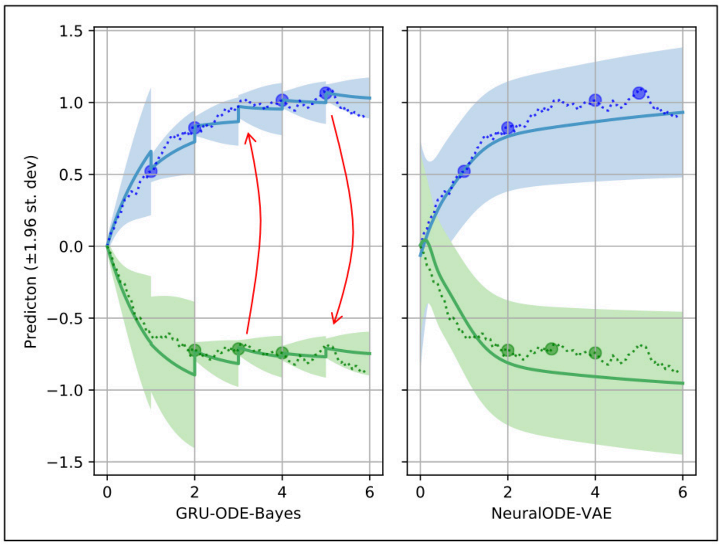
이 논문에서는 두 가지 method를 제시한다.

① Continuous time version of GRU

→ 기존 GRU는 discrete했는데 수식을 통해서 GRU의 ode func를 도출하였다.

② Baysian update network

→ 데이터가 없는 부분은 ODE로 적분하고 데이터가 있으면 GRU로 hidden layer를 업데이트한다.



이를 기반으로 구현한 모델은 두 가지 장점이 있다.

① 빠르게 θ를 추정한다.

② 변수간의 상관관계까지 배운다. 그림에서 한 변수에서 관측치가 발견되면 다른 변수에 영향을 주는 것을 확인할 수 있다 (red arrow)

학습하려는 데이터는 SDE를 따른다고 가정한다(일반적이라고 한다.)

$$dY(t) = \mu(Y(t))dt + \sigma(Y(t))dW(t)$$ 이고

논문의 모델이 $\mu(Y(t))$와 $\sigma(Y(t))$를 학습하는 것이 목표이다.

<모델 구현 및 학습>

1 GRU-ODE

위에서 언급했듯이 GRU를 continuous하게 바꿀것이다.

우리가 알고있는 GRU는 아래다.

reset gate $r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r)$

update gate $z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z)$

update vector $g_t = \tanh(W_h x_t + U_h(r_t \odot h_{t-1}) + b_h)$

GRU의 hidden state $h_t = z_t \odot h_{t-1} + (1 - z_t) \odot g_t$ 이고

$\Delta h_t = h_t - h_{t-1} = z_t \odot h_{t-1} + (1 - z_t) \odot g_t - h_{t-1}$

$= (1 - z_t) \odot (g_t - h_{t-1})$ 이므로

미분 방정식은 $\frac{dh(t)}{dt} = (1 - z(t)) \odot (g(t) - h(t))$ 이다.

이렇게 구하면 ODE Func로 사용할 수 있다. !

추가로 만약 관측치의 변수 중에 연속적 데이터가 있다면 그냥 위 GRU-ODE에 $x(t)$로 넣어주기만 하면 된다. 없으면 안 넣으면 그만.

이런 GRU-ODE는 특징이 있다.

① Boundness

hidden state가 $[-1, 1]$ 범위내에서만 변화한다. ($[-1, 1]$ 밖에서 시작한다고 해도)

② Continuity

$h(t)$가 연속적이다.

③ General numerical Integration

적분시 step-size를 adaptive하게 바꾸는 적분기를 사용할 수 있다.

2 GRU-Bayes

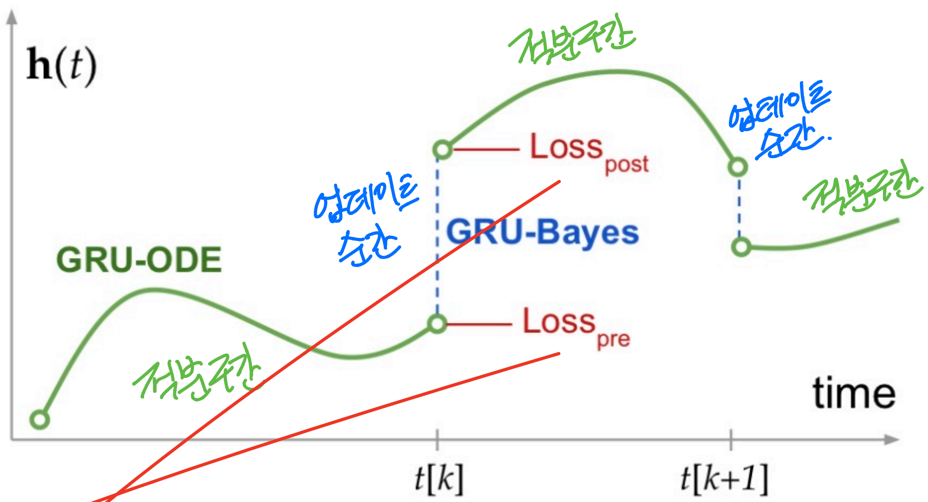
II은 ODE로 적분하는 부분이고 이건 업데이트하는 부분이다.

관측치가 있는 시간까지 도달했을 때, 업데이트를 한다. 식은 다음과 같다.

$$h(t_+) = GRU(h(t_-), f(y[k], m[k], h(t_-)))$$

GRU에 들어가는 인풋은 업데이트 전의 히든 벡터 $h(t_-)$ 와 그 시간에서의 관측치 $y[k]$ 와 mask vector $m[k]$ 와 $h(t_-)$를 preprocessing 하는 함수 f의 결과값이다.

종합했을 때 모델의 forward 그래프



3 Loss 값

$Loss_{pre}$: 업데이트하기 전에 관측값 y를 잘 나타내는지

negative log likelihood 사용

$$\text{Loss}_{\text{pre}} = -\sum_{j=1}^{D} m_j \log p(y_j|\theta = f_{\text{obs}}(\mathbf{h}_-)_j),$$

$Loss_{post}$: 업데이트한 후의 확률분포와

$P_{Bayes} \propto P_{pre} \cdot P_{obs}$ 분포의 차이를 loss로 사용

→ Baysian update를 모델이 흉내낼 수 있게 학습

$$\mathrm{Loss}_{\mathrm{post}} = \sum_{j=1}^{D} m_j D_{KL}(p_{\mathrm{Bayes},j} || p_{\mathrm{post},j}).$$

◦ 학습과정

Algorithm 1 GRU-ODE-Bayes

Input: Initial state $\mathbf{h}_0$,
observations $\mathbf{y}$, mask $\mathbf{m}$,
observation times $\mathbf{t}$, final time T.
Initialize $\mathrm{time} = 0, \mathrm{loss} = 0, \mathbf{h} = \mathbf{h}_0$.
for $k = 1$ **to** K **do**
$\{$ODE evolution to $\mathbf{t}[k]\}$
$\mathbf{h} = \mathbf{GRU\text{-}ODE}(\mathbf{h}, \mathrm{time}, \mathbf{t}[k])$
$\mathrm{time} = t[k]$
$\{$Pre-jump loss$\}$
$\mathrm{loss} \mathrel{+}= \mathrm{Loss}_{\mathrm{pre}}(\mathbf{y}[k], \mathbf{m}[k], \mathbf{h})$
$\{$Update$\}$
$\mathbf{h} = \mathbf{GRU\text{-}Bayes}(\mathbf{y}[k], \mathbf{m}[k], \mathbf{h})$
$\{$Post-jump loss$\}$
$\mathrm{loss} \mathrel{+}= \lambda . \mathrm{Loss}_{\mathrm{post}}(\mathbf{y}[k], \mathbf{m}[k], \mathbf{h})$
end for
$\{$ODE evolution to $T\}$
$\mathbf{h} = \mathbf{GRU\text{-}ODE}(\mathbf{h}, \mathbf{t}[N_K], T)$
return $(\mathbf{h}, \mathrm{loss})$