The Sequential Monte Carlo Transformer: a stochastic self-attention model for sequence prediction

Alice Martin, Sylvain Le Corff, Olivier Pietquin, Charles Ollion, Florian Strub

June 2nd 2020

Outline

Towards Generative Models for Neural Networks

Today's Neural Networks are excellent predictors: they give with great accuracy a single-point estimate of a given target for complex ML problems.

Yet, one open research question is the **design of neural generative models able to output a full predictive distribution** of observations given input data, to:

- Model **the inherent variability of the observations** (*aleatoric uncertainty*).
- Estimate **the level of confidence of the neural network** in its predictions (*epistemic uncertainty*).

Why outputting a prediction distribution (instead of a single-point estimate) matters?

Uncertainty quantification is key when designing AI systems for critical applications

Such as Medical Diagnosis, Autonomous Driving.
Such AI systems needs to be safe, and to provide a red flag when they are uncertain, so that human intervention can be used instead.

For some ML problems, there are no universal ground truth.

In Language Modelling, they are various ways to express an idea using a language.
Creating Language Models outputting a distribution of possible text sequences could:

- Improve Language Diversity.
- Solve some of the well-known problems in today's Neural Language Models.

The Transformer in a nutshell (1)

- A new sequence transduction model without recurrence or convolution. An alternative to recurrent neural networks for sequence modeling.
- Relies entirely on **the self-attention mechanism** to model global dependencies regardless of their distance in the input sequence.

Self-attention mechanism in the Transformer

$(X_s)_{s \geqslant 1}$ is a sequence of observations indexed by $\mathbb{N}$.
Each input data X_s is associated **with a query** q_s **and a set of key-value** (k_s, v_s) computed from linear transformations of the input.
From the set of keys and queries, **a softmax score function** is computed, which determines how much focus to place on each input in X_{-s} as X_s is processed.

The Transformer in a nutshell (2)

scaled-dot product attention

Transformer models use a *scaled dot product attention* to compute the attention score π:

$$\pi_s = \mathrm{softmax}(Q_s K_s^T / \sqrt{r})$$

final attention vector

This softmax score is used to compute a weighted sum of the values vectors:

$$\mathrm{Attention}(Q_s, K_s, V_s) = \mathrm{softmax}\left(Q_s K_s^T / \sqrt{r}\right) V_s$$

$\rightarrow$ This keeps the values of the elements of the sequence the prediction is focused on, and drops-out irrelevant elements.

The Transformer Architecture

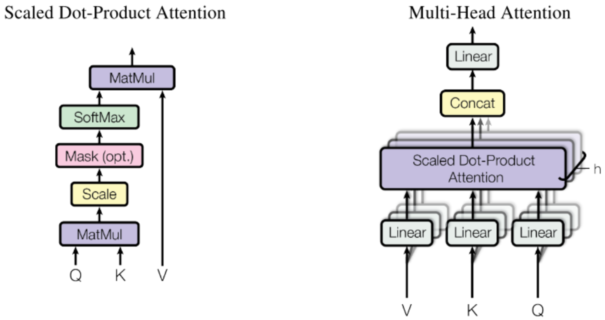


Figure 2: (left) Scaled Dot-Product Attention. (right) Multi-Head Attention consists of several attention layers running in parallel.

[Attention is all you Need, Vaswani al, 2017]

Introducing The Sequential Monte-Carlo (SMC) Transformer

1. Idea: Injecting noise in the self-attention model of *a Recurrent Transformer* to **model the inherent variability of observations** for sequential data.

2. In doing so, the self-attention parameters (latent states) become unobserved states of the dynamic model.
   $\rightarrow$ Intractability of the log-likelihood !

3. These unobserved states are estimated as a set of M particles using Sequential Monte Carlo Methods.

4. This gives **a generative model outputting a predictive distribution** in a supervised learning setting.

The SMC Transformer: stochastic self-attention model (I)

stochastic self-attention model when processing X_s given a window Δ of past observations

For all $0 \leqslant s \leqslant T$

$$q^h(s) = W^{h,q} X_s + \Sigma_{h,q}^{1/2} \varepsilon_q^h(s) \; ,$$

$$\kappa^h(s) = W^{h,\kappa} X_s + \Sigma_{h,\kappa}^{1/2} \varepsilon_\kappa^h(s)$$

$$v^h(s) = W^{h,v} X_s + \Sigma_{h,v}^{1/2} \varepsilon_v^h(s) \; ,$$

$W^{h,q}$, $W^{h,\kappa}$ and $W^{h,v}$ are unknown matrices
$(\Sigma_{h,q}, \Sigma_{h,\kappa}, \Sigma_{h,v})$are unknown semi definite-positive matrices
$(\varepsilon_q^h, \varepsilon_\kappa^h, \varepsilon_v^h)$are independent standard Gaussian random vectors in $\mathbb{R}^{\mathrm{r}}$.

The SMC Transformer: stochastic self-attention model (II)

Attention score at time t:

$$\mathrm{score}^h(t) = q^h(t)^T K^h(t) \quad \text{and}$$
$$\pi^h(t) = \mathrm{softmax}\left(\mathrm{score}^h(t)/\sqrt{r}\right).$$

$K^h(t) = [\kappa^h(s)]_{s=t-\Delta}^{s=t}$ is the sequence of past queries until $s = t$.

Attention vector at time t:

$$z^h(t) = \sum_{s=0}^{\Delta} \pi_s^h(t) v^h(t-s) + \Sigma_{h,z}^{1/2} \varepsilon_z^h(t)$$

$\pi_s^h(t)$ denotes the s-th component of $\pi^h(t)$.
$(\Sigma_{h,z})$ are unknown semi definite-positive matrices.
(ε_z^h) are independent standard Gaussian random vectors in $\mathbb{R}^{\mathrm{r}}$.

The SMC Transformer: Observation Model

In a classification setting, the observation model provides a probability vector $G_{\eta_{obs}}(r_t)$ on the finite observation space X based on the self-attention vectors.

In a regression framework, the observation model is given by

$$X_t = G_{\eta_{obs}}(r_t) + \varepsilon_t \ ,$$

where $G_{\eta_{obs}}$ is a FFNN with linear ouput layer and ε_t is a centered noise, e.g a centered Gaussian random vector with unknown variance Σ_{obs}.

The Training algorithm: Estimating the log-likelihood (1)

When injecting noise in the self-attention model of the Transformer, the latent attention parameters are unobserved random variables.
This leads to **an intractable likelihood function.**

The Fisher Trick

Maximum Likelihood Estimation may still be defined **using Fisher's identity**:

$$\nabla_\theta \log p_\theta(X_{1:T}) = \mathbb{E}_\theta[\nabla_\theta \log p_\theta(\zeta_{1:T}, X_{1:T}) | X_{1:T}] \qquad (1)$$

$\zeta_{1:T} = \{z_{1:T}, q_{1:T}, \kappa_{1:T}, v_{1:T}\}$ are the unobserved latent states of the stochastic dynamical system.

The Expectation of right term of equation (1) is estimated using **Sequential Monte Carlo Methods.**

The Training algorithm: Estimating the log-likelihood (2)

$$\nabla_\theta \log p_\theta(X_{1:T}) = \mathbb{E}_\theta[\nabla_\theta \log p_\theta(\zeta_{1:T}, X_{1:T}) | X_{1:T}]$$

Decomposition of the complete data likelihood:

$$p_\theta(X_{1:T}, \zeta_{1:T}) = \prod_{t=1}^{T} p_\theta(\zeta_t | \zeta_{t-\Delta:t-1}, X_{t-\Delta:t-1}) p_\theta(X_t | \zeta_{t-\Delta:t}, X_{t-\Delta:t-1})$$

$$p_\theta(X_t | \zeta_{t-\Delta:t}, X_{t-\Delta:t-1}) = G_{\eta_{obs}}(r(t))_{X_t}$$

in the classification setting, and

$$p_\theta(X_t | \zeta_{t-\Delta:t}, X_{t-\Delta:t-1}) = \varphi_{G_{\eta_{obs}}(z(t)), \Sigma_{\mathrm{obs}}}(X_t)$$

in the regression setting.

Sequential Monte Carlo (SMC) Methods in a nutshell.

$\zeta_{1:T} = \{z_{1:T}, q_{1:T}, \kappa_{1:T}, v_{1:T}\}$ are estimated as **a set of particles trajectories** $\xi^m_{1:T}$ sampled from the posterior distribution of $\zeta_{1:T}$ given $X_{1:T}$.

This set of particles are associated with importance resampling weights $(\omega^m_n)_{1 \leqslant m \leqslant M}$ such that $\sum_{m=1}^{M} \omega^m_T = 1$.

Final score function using SMC methods:

$$S_{\theta,T} = \nabla_\theta \log p_\theta(X_{1:T}) = \mathbb{E}_\theta[\nabla_\theta \log p_\theta(\zeta_{1:T}, X_{1:T}) | X_{1:T}]$$

$$S^M_{\theta,T} = \sum_{m=1}^{M} \omega^m_n \nabla_\theta \log p_\theta(\xi^m_{1:T}, X_{1:T}) \ ,$$

The SMC Transformer: The SMC algorithm

For all $t \geqslant 1$, once the observation X_t is available, the weighted particle sample $\{(\omega_t^m, \xi_{1:t}^m)\}_{m=1}^M$ is transformed into a new weighted particle sample **in 2 steps.**

Particle selection

1. Sample I_{t+1}^m in $\{1, \dots, M\}$ with probabilities proportional to $\{\omega_t^j\}_{1 \leqslant j \leqslant M}$.
2. Sample ξ_{t+1}^m using the observation model on the resampled trajectories $\xi_{1:t}^m$ until time t.

The SMC Transformer: The SMC algorithm

For all $t \geqslant 1$, once the observation X_t is available, the weighted particle sample $\{(\omega_t^m, \xi_{1:t}^m)\}_{m=1}^M$ is transformed into a new weighted particle sample **in 2 steps**.

Particle mutation

For any $m \in \{1, \dots, M\}$, the ancestral line $\xi_{1:t+1}^{\ell}$ is updated as follows:

$$\xi_{1:t+1}^m = (\xi_{1:t}^{I_{t+1}^m}, \xi_{t+1}^m)$$

and is associated with the importance weight defined by:

$$\omega_{t+1}^m \propto p_\theta(X_{t+1} | \xi_{t+1-\Delta:t+1}^m, X_{t+1-\Delta:t}) \,.$$

In the classification setting: $\omega_{t+1}^m \propto [G_{\eta_{obs}}(r_{t+1}^m)]_{X_{t+1}}$

In the regression setting:
$\omega_{t+1}^m \propto \exp\{-\|X_{t+1} - G_{\eta_{obs}}(r_{t+1}^m)\|_{\Sigma_{\mathrm{obs}}}^2/2\}$.

Particle Selection and Mutation in a nutshell.

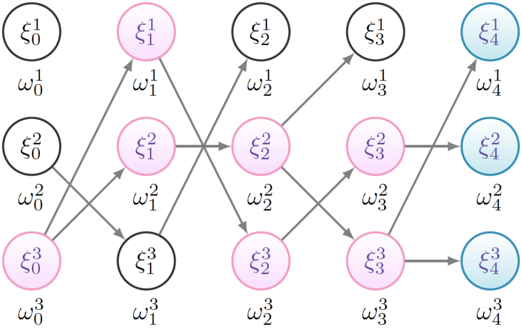
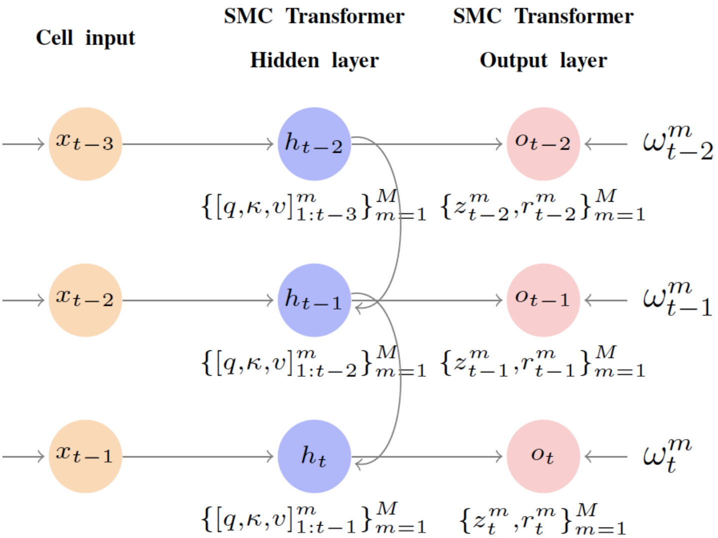


Figure 2: Auxiliary particle filter: $M = 3, n = 4$.

The SMC Transformer Architecture



Experiments: Outline

- On a synthetic dataset: Is the SMC Transformer able to learn the true variability of the observations for a known observation model ?

- On a real-word dataset: Application to uncertainty quantification on a time-series forecasting problem.

Comparison with MC-Dropout [Gal and Ghahramani, 2015].

Experiments: Synthetic Datasets (I)

Evaluation on synthetic auto-regressive time-series and a sequence length of 24 observations.

- **model I - unimodal gaussian noise**:

  $$X_0 \sim \mathcal{N}(0, 1)\ ,\ X_{t+1} = \alpha X_t + \sigma \varepsilon_{t+1}\ ,$$

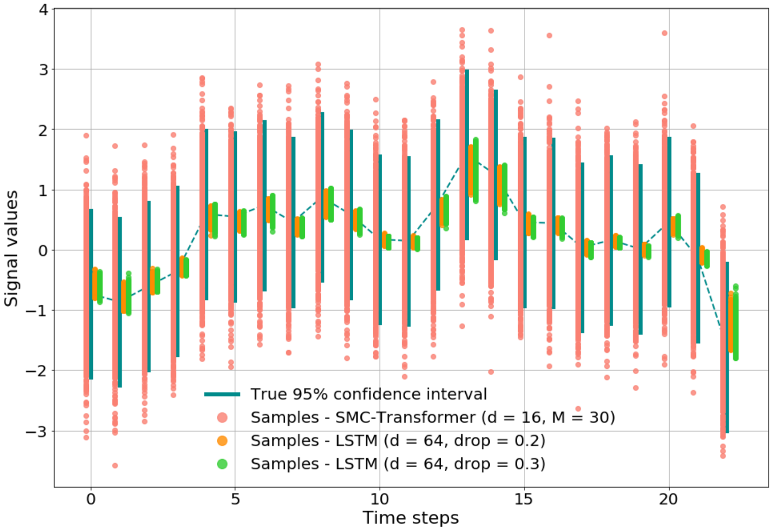
  $(\varepsilon_t)_{1 \leqslant t \leqslant 24}$ are i.i.d standard Gaussian variables independent of X_0.

- **model II - multimodal Gaussian Noise**:

  $$X_0 \sim \mathcal{N}(0, 1)\ ,\ X_{t+1} = \alpha U_{t+1} X_t + \beta(1 - U_{t+1}) X_t + \sigma \varepsilon_{t+1}\ ,$$

  $(\varepsilon_t)_{1 \leqslant t \leqslant 24}$ are i.i.d standard Gaussian variables independent of X_0.
  $(U_t)_{1 \leqslant t \leqslant 24}$ are i.i.d Bernoulli random variables with parameter p.

Synthetic Datasets: Predictive Distributions



Real-Word Dataset: Covid Data

Evaluation of the performance of the stochastic Transformer on the Covid-19 dataset[1] gathering daily deaths from the Covid-19 disease in 3261 US cities.

Estimation of the Observation's Variability

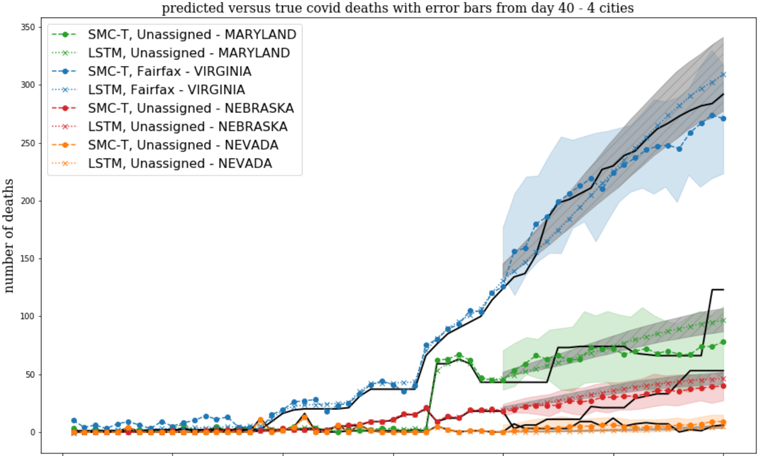
The variability of the observations is different for every sample and estimated with a two-steps procedure:

- First estimation of the global variance of the observations at training time
- Fine-tuning of the estimated noise per test sample at inference time using using 30 iterations of an Expectation Maximization algorithm on the first 40 days.

[1] https://github.com/CSSEGISandData/COVID-19

Real-Word Dataset: Inference Results

Multi-step predictions of covid daily deaths (mean predictions with confidence intervals) for a LSTM with Dropout and a SMC Transformer models.



Conclusion: Pros and Limits of the SMC Transformer

pros

- One of the few Generative Models based on Recurrent Architecture for sequence prediction
- Predicts with great accuracy a known model of observations compared to the most popular predictive distribution algorithm, *MC-Dropout*
- The algorithm gives a flexible framework for uncertainty quantification at inference.
- The SMC Transformer layer can be used as a "plug-and-play" tool for uncertainty quantification on deeper neural networks encoding sequential data.

Limits

- Complexity of the Model and the Architecture
- Computational Cost