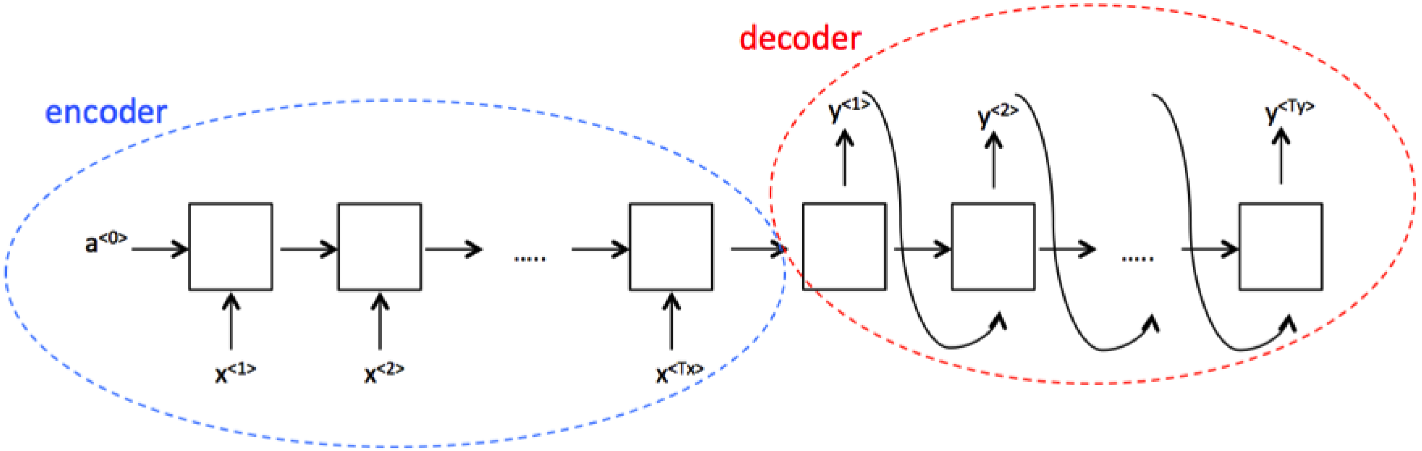
UNIVERSITY OF TORONTO | Engineering

Probabilistic Sequence-to-Sequence Learning

Table of Contents

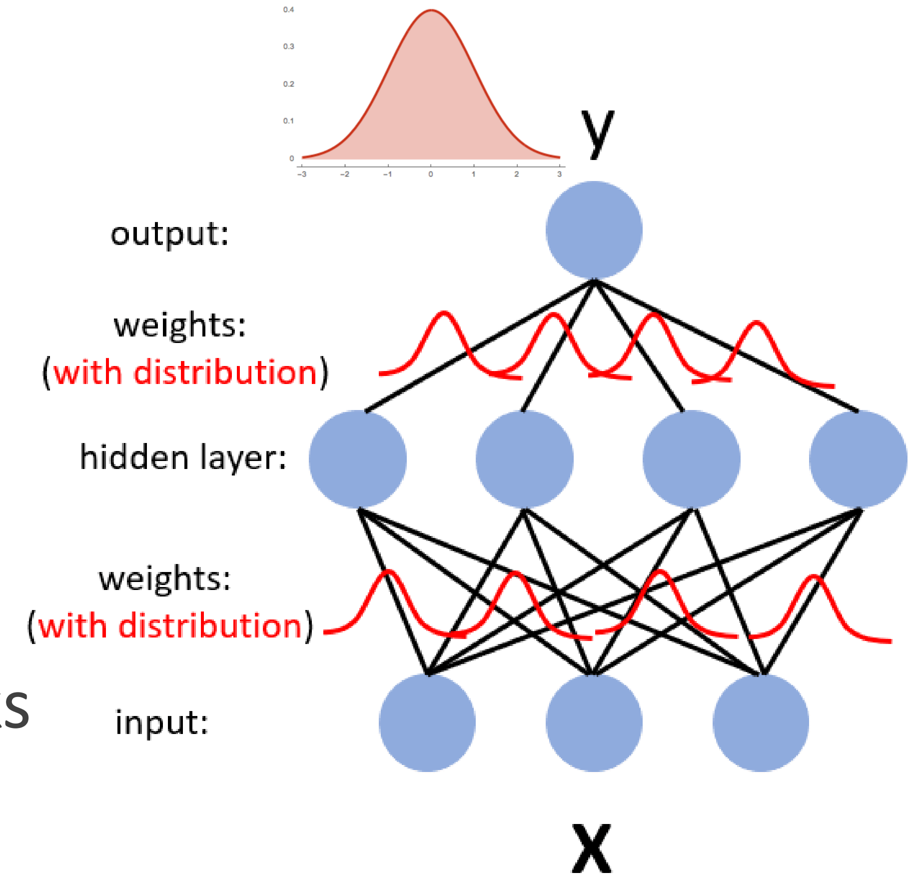
Sequence-to-Sequence Learning

- Sequential models are widely used in natural language processing, computer vision, robotics control, finance and etc.
- Sequence-to-sequence learning models encode the input sequence into a hidden state via a deep neural network (DNN) and decode the output sequence via another DNN from the hidden state
- X can be an image, and Y can be a sentence to describe that image
- X can be control input (acceleration and direction) to an autonomous vehicle, and Y can be the state (position and speed) of the vehicle



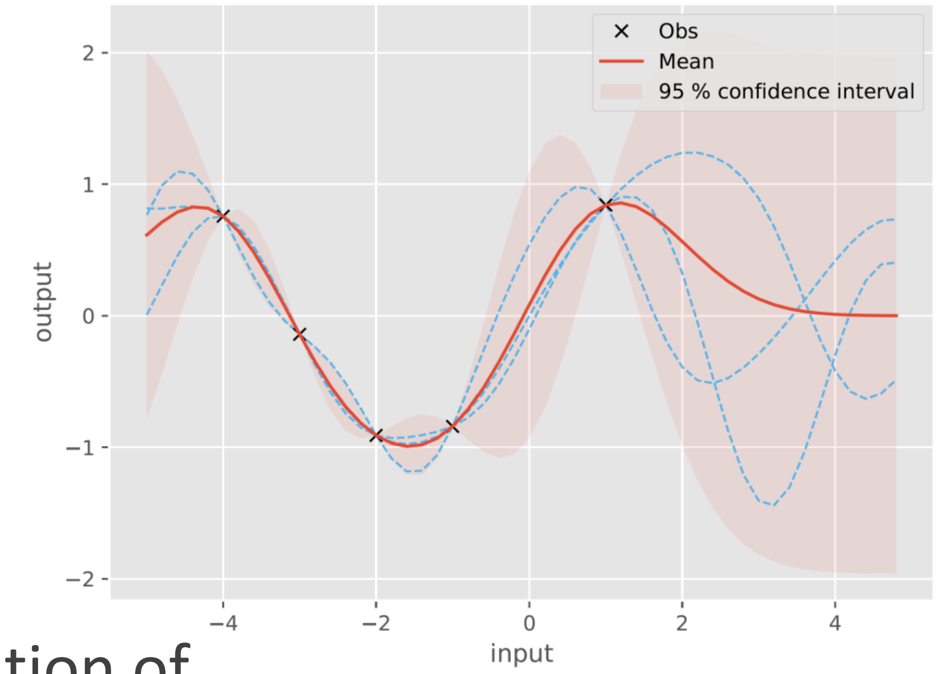
Probabilistic Inference

- Stand DNNs are non-probabilistic
- It is import to quantify the uncertainty in output sequences
- Bayesian neural networks can be used to reason about model uncertainty (Gal & Ghahramani 2016)
- However, training Bayesian neural networks requires computation-intense Variational Inference or Markov Chain Monte Carlo



Deep Kernel Learning

- The extent of prior knowledge and uncertainty expressed in a parametric model (e.g. DNN) is relatively limited
- Gaussian processes (GP) provide richer representation of uncertainty (Rasmussen & Williams 2006)



- Deep kernel learning models based on combination of CNN, RNN and GP lead to superior results in probabilistic prediction (Al et al. 2017, Wilson et al. 2017)
- We introduce a GP layer into encoder-decoders for probabilistic sequence-to-sequence learning

Highlights

- A new encoder-decoder architecture is developed for sequence-to-sequence learning with predictive distribution
- The new method, termed Attentive-GP, is based on the combination of the Transformer architecture and GP regression
- We also develop a block-wise training algorithm to train the proposed method effectively

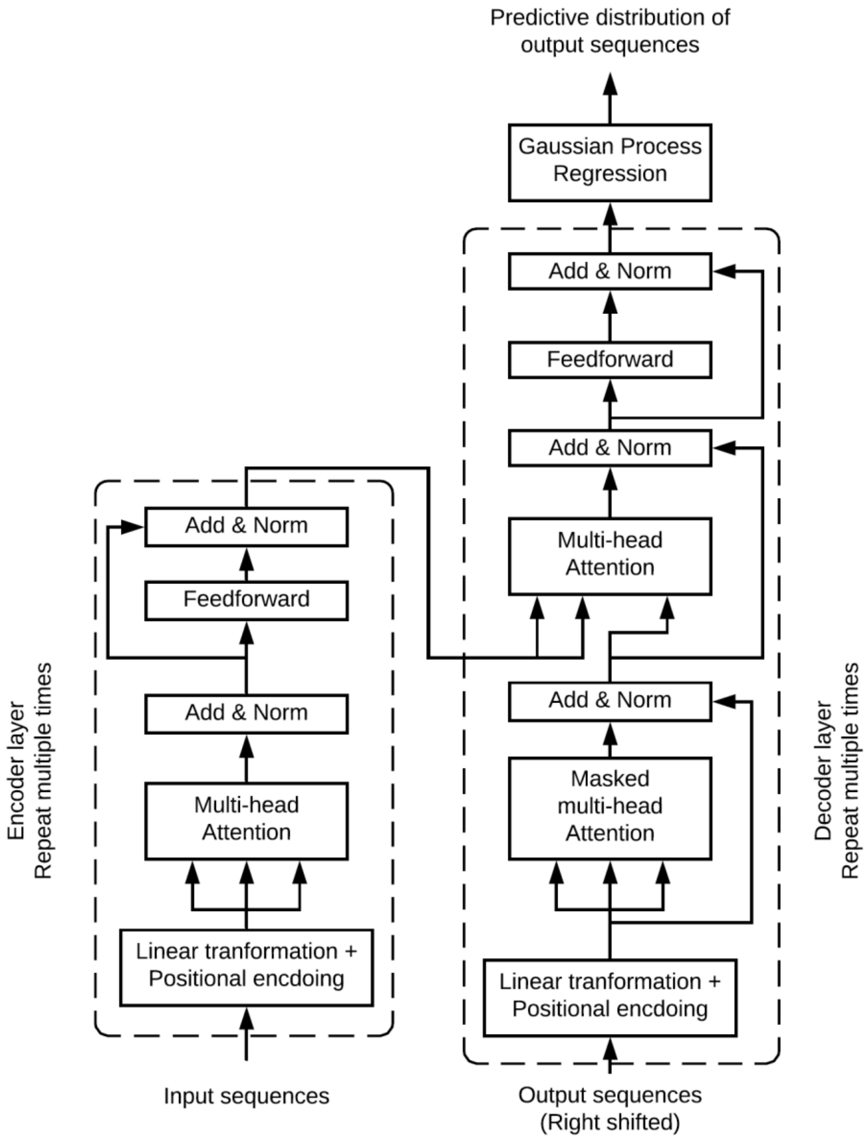
Attentive-GP

- The input sequence $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ and past output sequence $\{\mathbf{y}_1, \dots \mathbf{y}_{i-1}\}$ are propagated through Transformer to get the most relevant feature for future output

$$\bar{\mathbf{x}}_i = \phi(\{\mathbf{x}_1, \dots, \mathbf{x}_N\}, \{\mathbf{y}_1, \dots \mathbf{y}_{i-1}\})$$

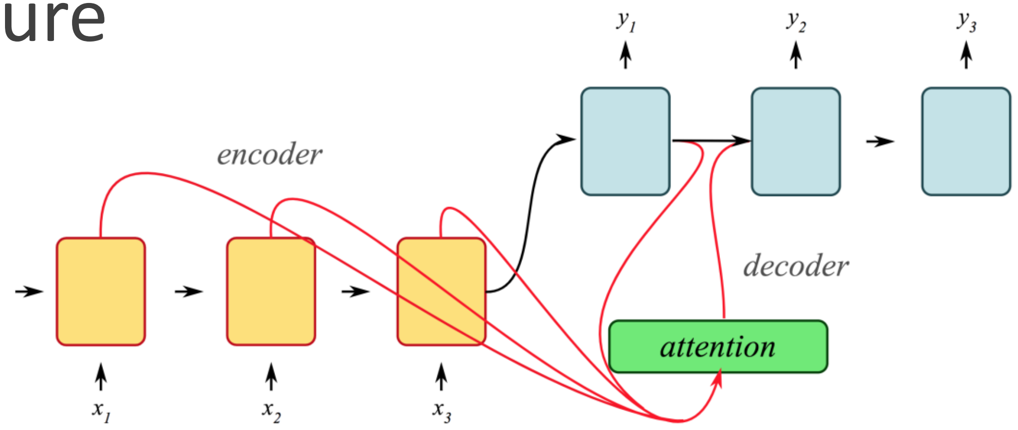
- The feature is mapped to future output via GP regression layer

$$\mathbf{y}_i = f(\bar{\mathbf{x}}_i) + \epsilon_i$$



Encoder-Decoder Architecture

- Transformer – an attention based encoder-decoder is the current state of the art for sequence-to-sequence learning
- The attention mechanism searches the most relevant information between input and output sequences
- Encoder-decoders with attention are capable of learning very long sequences with complicated structure



Gaussian Process Regression Layer

- GP is a non-parametric Bayesian model
- The posterior predictive distribution of test data can be derived from the joint distribution of training and test data

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu} \\ \boldsymbol{\mu}_* \end{bmatrix}, \begin{bmatrix} K_{X,X} + \sigma^2 I & K_{X,X_*} \\ K_{X_*,X} & K_{X_*,X_*} \end{bmatrix} \right)$$

- GP offers rich representation of uncertainty via kernelized covariance function (e.g. Squared Exponential kernel function)

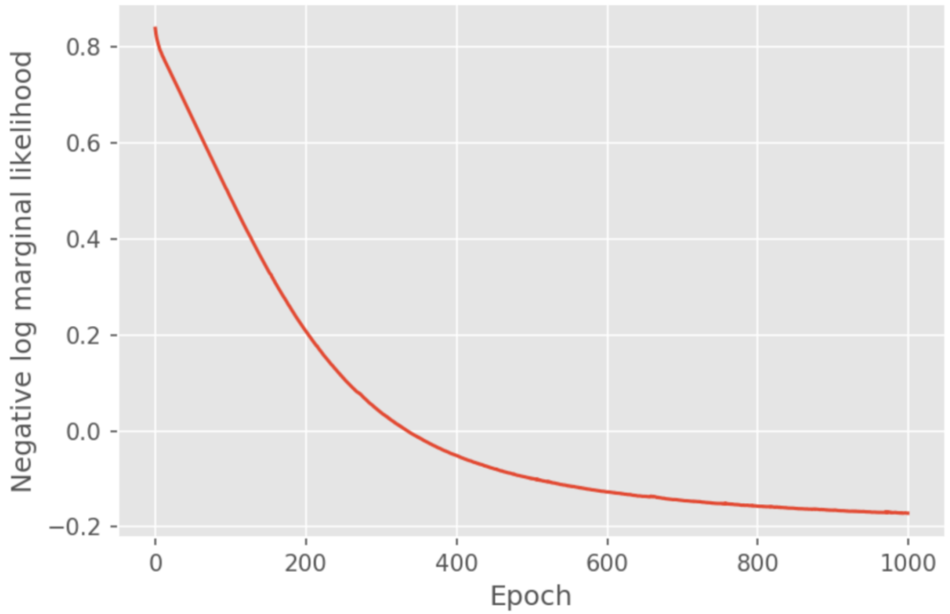
$$\kappa\left(\bar{\mathbf{x}}_i, \bar{\mathbf{x}}_j\right) = \exp\left(-\frac{1}{2} \left(\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j\right)^\top \Theta^{-2} \left(\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j\right) \right)$$

Training

- Let $\mathbf{W}$ be parameters in Transformer
- Let $\boldsymbol{\theta}$ be the parameters in GP layer
- The model is trained by minimizing the negative log marginal likelihood w.r.t. to $\mathbf{W}$ and $\boldsymbol{\theta}$ alternatingly
- The proposed training algorithm converges to a stationary point in theory and practice

Algorithm 1

Initialize parameters $\theta_0^0, \mathbf{W}_0^0$
for $k = 1, 2, 3, \ldots$ **do**
  for $\tau = 1, 2, \ldots T$ **do**
    Update $\mathbf{W}_{k-1}^{\tau-1}$ based on mini-batch data
  end for
  $\mathbf{W}_k^0 \leftarrow \mathbf{W}_{k-1}^T$
  for $\tau = 1, 2, \ldots T$ **do**
    Update $\boldsymbol{\theta}_{k-1}^{\tau-1}$ based on full-batch training data
  end for
  $\boldsymbol{\theta}_k^0 \leftarrow \boldsymbol{\theta}_{k-1}^T$
end for
return $\theta, \mathbf{W}$



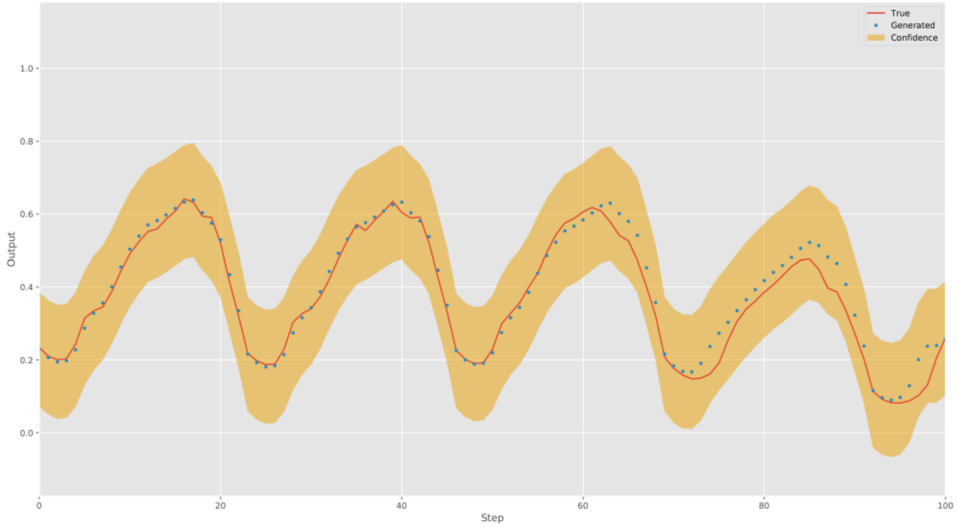
Numerical Results

- **Robot Arm**: Input sequence – 21 connected robot arm joint position, speed and acceleration; Output sequence – robot arm joint torque
- **Suspension System**: Input sequence – external disturbance to the vehicle; Output sequence – relative position move
- **Smart Grid**: Input sequence – temperature in 11 cities; Output sequence - total electrical load on the grid

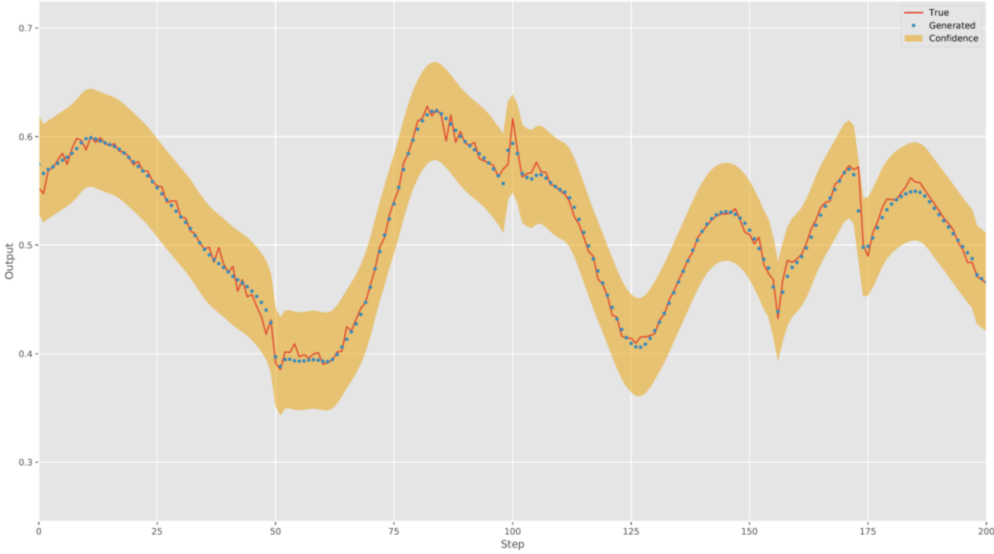
Root Mean Squared Error	Robot Arm	Suspension System	Smart Grid
RNN	4.924e-2	4.474e-2	9.714e-2
LSTM	4.704e-2	4.284e-2	8.624e-2
GRU	4.799e-2	4.176e-2	8.672e-2
RNN Enc-Dec	1.353e-2	1.903e-2	3.564e-2
Transformer	7.068e-3	1.495e-2	3.325e-2
Attentive-GP	6.573e-3	1.304e-2	3.242e-2

Numerical Results

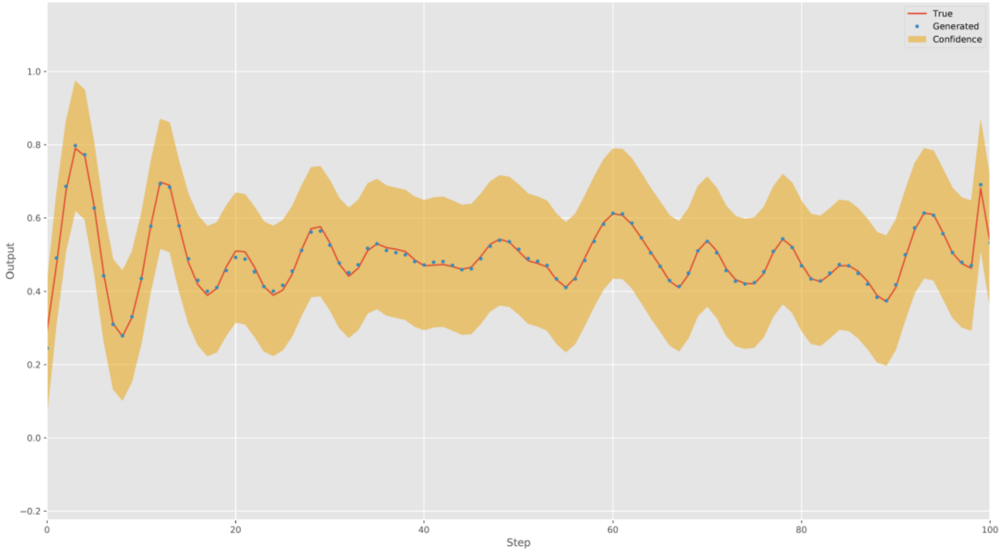
Generated Output Sequence of Smart Grid



Generated Output Sequence of Robot Arm



Generated Output Sequence of Suspension System



On-going Research

- Reinforcement Learning
  - Value function approximation using deep kernel learning approaches
- Digital Twin
  - Virtual environment upon probabilistic sequence-to-sequence learning for RL agents