Dual Stage Attention Based Recurrent Neural Network for Time Series (Qin et. al.) Mathematical Model

Sonam Ghosh

October 2018

First Stage - Input attention mechanism to adaptively extract relevant input features at each time step by referring to previous encoder hidden State.
Second Stage - Temporal Attention mechanism to select relevant encoder hidden states across all time steps.
Given n driving series (input features):

$$\mathbf{X} = (\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^n)^\intercal = (\mathbf{x}_1, \mathbf{x}_2, \ldots, x_T) \in \mathbb{R}^{n \times T}$$

Where T is the length of window size and

$$\mathbf{x}_t = (x_t^1, x_t^2, \ldots, x_t^n)^\intercal \in \mathbb{R}^n$$

Is the vector of n exogenous input series at time t. Previous values of the target series is given by

$$(y_1, y_2, \ldots, y_{T-1})$$

where $y_t \in \mathbb{R}$.
The model learns a **nonlinear** mapping ($F(\cdot)$ nonlinear func) to the current value of the target series y_T:

$$\hat{y}_T = F(y_1, \ldots, y_{T-1}, \mathbf{x}_1, \ldots, \mathbf{x}_T) \tag{1}$$

Encoder - RNN that encodes input sequences into feature representation for machine trnanslation.
For an input sequence $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T), \mathbf{x}_T \in \mathbb{R}^n$, the encoder is appled to learn a mapping from $\mathbf{x}_t \to \mathbf{h}_t$:

$$\mathbf{h}_t = f_1(\mathbf{h}_{t-1}, \mathbf{x}_t) \tag{2}$$

Where $\mathbf{h}_t \in \mathbb{R}^m$ is the hidden state of the encoder at time t, m is the size of the hidden state, and f_1 is a non-linear activation function where a LSTM is used.

The LSTM unit has a memory cell with state $\mathbf{s}_t$ at time t that is controlled by three sigmoid gates—the forget gate $\mathbf{f}_t$, input gate $\mathbf{i}_t$, and output gate $\mathbf{o}_t$ formulated as follows:

$$\mathbf{f}_t = \sigma(\mathbf{W}_f[\mathbf{h}_{t-1}; \mathbf{x}_t] + \mathbf{b}_f) \tag{3}$$

$$\mathbf{i}_t = \sigma(\mathbf{W}_i[\mathbf{h}_{t-1}; \mathbf{x}_t] + \mathbf{b}_i) \tag{4}$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_o[\mathbf{h}_{t-1}; \mathbf{x}_t] + \mathbf{b}_o) \tag{5}$$

$$\mathbf{s}_t = \mathbf{f}_t \odot \mathbf{s}_{t-1} + \mathbf{i}_t \odot \tanh(\mathbf{W}_s[\mathbf{h}_{t-1}; \mathbf{x}_t] + \mathbf{b}_s) \tag{6}$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{s}_t) \tag{7}$$

Where $[\mathbf{h}_{t-1}; \mathbf{x}_t] \in \mathbb{R}^{m+n}$ is a concatenation of the previous hidden state $\mathbf{h}_{t-1}$ and current input $\mathbf{x}_t$. $\mathbf{W}_f, \mathbf{W}_i, \mathbf{W}_o, \mathbf{W}_s \in \mathbb{R}^{m\times(m+n)}$ and $\mathbf{b}_f, \mathbf{b}_i, \mathbf{b}_o, \mathbf{b}_s \in \mathbb{R}^m$ are learning parameters. σ and $\odot$ are the logistic sigmoid function and element-wise multiplication respectively.
Input attention based encoder - adaptively select relevant driving series. Given an k-th input driving series $\mathbf{x}^k = (x_1^k, x_2^k, \dots, x_T^k)^\intercal \in \mathbb{R}^T$. The input attention mechanism is given by a multi-layer perceptron, referring to previous hidden state $\mathbf{h}_{t-1}$ and unit cell $\mathbf{s}_{t-1}$ in the encoder LSTM unit given by:

$$e_t^k = \mathbf{v}_e^\intercal \tanh(\mathbf{W}_e[\mathbf{h}_{t-1}; \mathbf{s}_{t-1}] + \mathbf{U}_e\mathbf{x}^k) \tag{8}$$

$$\alpha_t^k = \frac{\exp(e_t^k)}{\sum_{i=1}^n \exp(e_t^i)} \tag{9}$$

where $\mathbf{v}_e \in \mathbb{R}^T, \mathbb{W}_e \in \mathbb{R}^{T\times 2m}, \mathbf{U}_e \in \mathbb{R}^{T\times T}$ are learning parameters. α_t^k is the attention weight that measures importance of k-th input feature at time t.
Adaptively extract driving series with

$$\tilde{\mathbf{x}}_t = (\alpha_t^1 x_t^1, \alpha_t^2 x_t^2, \dots, \alpha_t^n x_t^n)^\intercal \tag{10}$$

Hidden state at time t is updated to

$$\mathbf{h}_t = f_1(\mathbf{h}_{t-1}, \tilde{\mathbf{x}}_t) \tag{11}$$

Decoder - utilization of another LSTM unit to decode the encoded input information to predict $\hat{y}_T$.

Temporal attention is used to adaptively select relevant encoder hidden states across all time. Attention weight comes from previous decoder hidden state $\mathbf{d}_{t-1} \in \mathbb{R}^p$ and cell state of LSTM unit $\mathbf{s'}_{t-1} \in \mathbb{R}^p$:

$$l_t^i = \mathbf{v}_d^\intercal \tanh(\mathbf{W}_d[\mathbf{d}_{t-1}; \mathbf{s'}_{t-1}] + \mathbf{U}_d \mathbf{h}_i), \qquad 1 \leq i \leq T \tag{12}$$

$$\beta_t^i = \frac{\exp(l_t^i)}{\sum_{j=1}^T \exp(l_t^j)} \tag{13}$$

Where $[\mathbf{d}_{t-1}; \mathbf{s'}_{t-1}] \in \mathbb{R}^{2p}$ is a concatenation of previous hidden state and cell state of LSTM unit. $\mathbf{v}_d \in \mathbb{R}^m, \mathbf{W}_d \in \mathbb{R}^{m \times 2p}, \mathbf{U}_d \in \mathbb{R}^{m \times m}$ are learning parameters. Attention weight β_t^i represents importance of i-th encoder hidden state for prediction.

After mapping of encoder hidden state to temporal component of input, the attention mechanism provides the context vector as a weighted sum of all encoder hidden states:

$$\mathbf{c}_t = \sum_{i=1}^T \beta_t^i \mathbf{h}_i \tag{14}$$

Combine with given target series $(y_1, y_2, \ldots y_{T-1})$:

$$\tilde{y}_{t-1} = \tilde{\mathbf{w}}^\intercal [y_{t-1}; \mathbf{c}_{t-1}] + \tilde{b} \tag{15}$$

where $[y_{t-1}; \mathbf{c}_{t-1}] \in \mathbb{R}^{m+1}$ is concatenation of the decoder input y_{t-1} and context vector $\mathbf{c}_{t-1}$. $\tilde{\mathbf{w}} \in \mathbb{R}^{m+1}, \tilde{b} \in \mathbb{R}$ map concatenation to size of decoder input. Update of decoder hidden state is given by:

$$\mathbf{d}_t = f_2(\mathbf{d}_{t-1}, \tilde{y}_{t-1}) \tag{16}$$

Where f_2 the nonlinear function is another LSTM unit for long-term dependency modeling. The update functions are given by

$$\mathbf{f}'_t = \sigma(\mathbf{W}'_f[\mathbf{d}_{t-1}; \tilde{y}_{t-1}] + \mathbf{b}'_f) \tag{17}$$

$$\mathbf{i}'_t = \sigma(\mathbf{W}'_i[\mathbf{d}_{t-1}; \tilde{y}_{t-1}] + \mathbf{b}'_i) \tag{18}$$

$$\mathbf{o}'_t = \sigma(\mathbf{W}'_o[\mathbf{d}_{t-1}; \tilde{y}_{t-1}] + \mathbf{b}'_o) \tag{19}$$

$$\mathbf{s}'_t = \mathbf{f}'_t \odot \mathbf{s}'_{t-1} + \mathbf{i}'_t \odot \tanh(\mathbf{W}'_s[\mathbf{d}_{t-1}; \tilde{y}_{t-1}] + \mathbf{b}'_s) \tag{20}$$

$$\mathbf{d}_t = \mathbf{o}'_t \odot \tanh(\mathbf{s}'_t) \tag{21}$$

Where $[\mathbf{d}_{t-1}; \tilde{y}_{t=1}] \in \mathbb{R}^{p+1}$ is the concatenation of previous hidden state $\mathbf{d}_{t-1}$ and decoder input $\tilde{y}_{t-1}$. $\mathbf{W}'_f, \mathbf{W}'_i, \mathbf{W}'_o, \mathbf{W}'_s \in \mathbb{R}^{p \times (p+1)}, \mathbf{b}'_f, \mathbf{b}'_i, \mathbf{b}'_o, \mathbf{b}'_s \in \mathbb{R}^p$ are learning parameters.

Model is used to approximate the function F, the nonlinear mapping function to obtain an estimate of current output $\hat{y}_T$ with observation of all inputs as well as previous outputs.

$$\begin{aligned} \hat{y}_T &= F(y_1, \ldots, y_{T-1}, \mathbf{x}_1, \ldots, \mathbf{x}_T) \\ &= \mathbf{v}_y^\intercal(\mathbf{W}_y[\mathbf{d}_T; \mathbf{c}_T] + \mathbf{b}_w) + b_v \end{aligned} \tag{22}$$

where $[\mathbf{d}_T; \mathbf{c}_T] \in \mathbb{R}^{p+m}$ is a concatenation of the decoder hidden state and the context vector. $\mathbf{W}_y \in \mathbb{R}^{p\times(p+m)}, \mathbf{b}_w \in \mathbb{R}^p$ map the concatenation to size of decoder hidden states. Linear function with weights $\mathbf{v}_y \in \mathbb{R}^p, b_v \in \mathbb{R}$ provides the final prediction result.

Minibatch Stochastic Gradient descent with Adam Optimizer is used to **train** the model. Mini batch size—128, learning rate—0.001 and reduced by 10% after 10000 iterations. DARNN is smooth and differentiable. Objective function is given by

$$\mathcal{O}(y_T, \hat{y}_T) = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_T^i - y_T^i)^2 \tag{23}$$

Where N is number of training samples.