0.1 UNN maths

Our data is generated in the following way:

$$y_{j,t,k}(X) = RBF(X_j)_t + e_{j,k} \tag{1}$$

$e_{j,0}, e_{j,1} \sim N(0,2)$ independently.
$k \in [0,1]$, j is observation index.
$X_j = \{x_{j,1}, x_{j,2}, .., x_{j,N}\}$ with $x_{j,i} \sim U[5,15]$ and N being the sequence length.

For simplicity we want to ignore k so that we can talk about the usual 1-D case. Therefore, we create a mapping such that $y_{j,t,k} -- > y^k_{j,t}$. And so, for the purposes of log-likelihood, we can proceed as-if k is not there and we have a joint distribution indexed by t.

Our goal is to predict $P(y_{j,t+1}|y_{j,1}, y_{j,2}, \ldots y_{j,t})$.
Since our data is unordered, our dataset might look like:
$$\begin{bmatrix} y_{2,4} & y_{2,3} & y_{2,1} & y_{2,2} \\ y_{4,1} & y_{4,4} & y_{4,2} & y_{4,3} \\ y_{1,1} & y_{1,2} & y_{1,4} & y_{1,3} \end{bmatrix}$$

For maximizing the log-likelihood, we index $y_{j,i}$ by their respective rows (r) and columns (c).
So $y_{2,4} -> y_{1,1}$. Also, given the parameters of the network θ the rows of the matrix are conditionally independent. For row r, we have (notice that we start from the second column of the matrix our log-likelihood since beforehand we have no history)

$$\begin{aligned} logP(y_{r,2:N};\theta, x_{r,1:N}) &= log(P(y_{r,2}|y_{r,1};\theta, x_{r,1}, x_{r,2}) \ldots P(y_{r,N}|y_{r,1:(N-1)};\theta, x_{r,1:N})) \\ &= \frac{1}{N}\sum_{i=2}^{N} -\frac{1}{2} * \frac{(y_{r,i} - \mu(y_{r,<i};\theta, x_{r,\leq i}))^2}{\sigma(y_{r,<i};\theta, x_{r,\leq i}))^2} - log(\sigma(y_{r,<i};\theta, x_{r,\leq i})) \end{aligned} \tag{2}$$

And for the whole dataset this would be:

$$
\begin{aligned}
\sum_{j=1}^{M} logP(y_{j,r,2:N};\theta,x_{r,1:N}) &= \sum_{j=1}^{M} log(P(y_{j,r,2}|y_{j,r,1};\theta,x_{j,r,1},x_{j,r,2})(y_{j,r,N}|y_{j,r,1:(N-1)};\theta,x_{j,r,1:N})) \\
&= \frac{1}{M}\sum_{j=1}^{M}\frac{1}{N}\sum_{i=2}^{N} -\frac{1}{2} * \frac{(y_{j,r,i}-\mu(y_{j,r,<i};\theta,x_{j,r,\leq i}))^2}{\sigma(y_{j,r,<i};\theta,x_{j,r,\leq i}))^2} - log(\sigma(y_{j,r,<i};\theta,x_{j,r,\leq i})
\end{aligned}
\tag{3}
$$

Where we assume here-under that $y_{j,t}|y_{j,<t};x_{j,\leq t} \sim N(\mu(y_{j,<t};x_{j,\leq t},\theta),\sigma(y_{j,<t};x_{j,\leq t},\theta))$

And $y_{r,<i} = [y_{r,1},y_{r,2},\dots,y_{r,i-1}]$ and $y_{r,2:N} = [y_{r,1},y_{r,2},\dots,y_{r,N}]$

0.2 Model

Predicting target variable y_{t+1} is calculated by:

First, we embed the x-values, y-values, and k-values (which sequence 0/1).

$$x_{t,j} = embedding(x_t)_j \tag{4}$$

$$y_{t,j} = embedding(y_t)_j \tag{5}$$

$$k_{t,j} = embedding(k_t)_j \tag{6}$$

embedding $\in R^e$

Secondly, we create a dot product attention:

$$Q_{t,j} = \sum_{h=1}^{e} x_{t,h} * wq_{h,j} \tag{7}$$

$$K_{t,j} = \sum_{h=1}^{e} x_{t,h} * wk_{h,j} \tag{8}$$

$$V_{t,j} = \sum_{h=1}^{e} y_{t,h} * wv_{h,j} \tag{9}$$

Q_t (Query), K_t (Key), V_t (Value) ($\in R^e$) and ($wq, wk, wv \in R^{exe}$)
The next layer ("MatMul") is defined as:

$$(dot)_{t,m} = \sum_{h=1}^{e} Q_{th} * K^T_{hm} \tag{10}$$

Where $dot \in R^{(t+1)x(t+1)}$, t is the row index embedding of target variable y_t in Q, and m is the row index embedding of target variable y_m in K.

The "score" layer is used to normalise the dot layer and is calculated as:

$$score_{j,k} = \frac{exp^{(dot)_{j,k}}}{\sum_{h=1}^{j-1} exp^{(dot)_{j,h}}} \tag{11}$$

Where j=2:t+1. Notice that score will be a lower triangular matrix (upper = 0). $score \in R^{txt}$ and t is the index representing the row of similarities between the embedded y_t and the embedded y_k, with $k \leq t$.
The last layer indicates how much attention we should pay to each of the previous embedded (V) variables in the sequence. The output from this layer we call "*att*".

$$att_{t,j} = \sum_{h=1}^{t} score_{t,k} * V_{k,j} \tag{12}$$

$$out_{t,j} = \sum_{h=1}^{\ell} att_{t,h} * A1_{h,j} + \sum_{h=1}^{\ell} x_{t+1,h} * A2_{h,j} + \sum_{h=1}^{\ell} k_{t+1,h} * A3_{h,j} \tag{13}$$

$$out_{t,j} = max(0.01 * out_{t,j}, out_{t,j}) \tag{14}$$

$$out_{t,j} = \sum_{h=1}^{\ell} out_{t,h} * A_4 h, j \tag{15}$$

A1, A2 and A3 $\in R^{ex\ell)}$, A4 $\in R^{\ell x2}$