Fundamentals of RNN and LSTM Network

Kim Choeun

August 5, 2024

Seoul National University

The Roots of RNN

Delay Differential Equation

$$\frac{d\vec{s}(t)}{dt} = A\vec{s}(t) + B\vec{r}(t-\tau_0) + C\vec{x}(t) + \vec{\phi}$$
$$\vec{r}(t-\tau_0) = G\left(\vec{s}(t-\tau_0)\right)$$

- $A, B, C, \vec{\phi}$ are the parameters
- t : time in continuous domain

  $\vec{s}(t)$: state of the system at time t

  $\vec{x}(t)$: external input to the system

  $\vec{r}(t)$: readout signal. obtained from state signal via $G(\cdot)$ in NN

  τ_0 : time delay term. represents memory aspect of the system.

  $G(\cdot)$: warping non-linearity

Deriving RNN Formulation

- Discretize time steps using Backward Euler Method

Denote the duration of the sampling time step as ΔT,

$$
\begin{aligned}
t &= n\Delta T \\
\frac{d\vec{s}(t)}{dt} &\approx \frac{\vec{s}(n\Delta T + \Delta T) - \vec{s}(n\Delta T)}{\Delta T} \\
\frac{\vec{s}(n\Delta T + \Delta T) - \vec{s}(n\Delta T)}{\Delta T} &\approx A\vec{s}(n\Delta T + \Delta T) + B\vec{r}(n\Delta T + \Delta T - \tau_0) \\
&\quad + C\vec{x}(n\Delta T + \Delta T) + \vec{\phi}
\end{aligned}
$$

Set the delay, $\tau_0 = \Delta T$

$$
\begin{aligned}
\vec{s}(n\Delta T + \Delta T) - \vec{s}(n\Delta T) = \Delta T \,(& A\vec{s}((n+1)\,\Delta T) + B\vec{r}(n\Delta T) \\
& + C\vec{x}((n+1)\,\Delta T) + \vec{\phi}\,)
\end{aligned}
$$

Deriving RNN Formulation

ΔT can be dropped from the arguments, which leaves the time axis dimensionless.

$$(I - (\Delta T)\, A)\, \vec{s}\,[n+1] = \vec{s}\,[n] + ((\Delta T)\, B)\, \vec{r}\,[n] + ((\Delta T)\, C)\, \vec{x}\,[n+1] + (\Delta T)\, \vec{\phi}$$

Define $W_s = (I - (\Delta T)\, A)^{-1}$,

$$\vec{s}\,[n+1] = W_s \vec{s}\,[n] + ((\Delta T)\, W_s B)\, \vec{r}\,[n] + ((\Delta T)\, W_s C)\, \vec{x}\,[n+1] + \left((\Delta T)\, W_s \vec{\phi}\right)$$

Shift the index, n, forward by 1 step,

$$\vec{s}\,[n] = W_s \vec{s}\,[n-1] + ((\Delta T)\, W_s B)\, \vec{r}\,[n-1] + ((\Delta T)\, W_s C)\, \vec{x}\,[n] + \left((\Delta T)\, W_s \vec{\phi}\right)$$

Canonical RNN Formulation

Redefine the weight matrices and the bias vector as

$$W_r = (\Delta T)\, W_s B$$
$$W_x = (\Delta T)\, W_s C$$
$$\vec{\theta_s} = (\Delta T)\, W_s \vec{\phi}$$

Then, we get the canonical RNN Formulation:

$$\vec{s}[n] = W_s \vec{s}[n-1] + W_r \vec{r}[n-1] + W_x \vec{x}[n] + \vec{\theta_s}$$
$$\vec{r}[n] = G\left(\vec{s}[n]\right)$$

Standard RNN Formulation

From canonical RNN form,

$$\vec{s}[n] = W_s\vec{s}[n-1] + W_r\vec{r}[n-1] + W_x\vec{x}[n] + \vec{\theta_s}$$
$$\vec{r}[n] = G(\vec{s}[n])$$

let A be a diagonal matrix with large negative entries ($a_{ii} \ll 0$) for the stability and $\Delta T = 1$.

Then, $W_s = (I - A)^{-1} \approx diag\left(\frac{1}{|a_{ii}|}\right)$.
Since $\frac{1}{|a_{ii}|} \approx 0$, the effect of $\vec{s}[n-1]$ on the system's trajectory will be negligible.

Standard RNN Formulation

Then, we get the standard RNN Formulation:

$$\vec{s}[n] = W_r \vec{r}[n-1] + W_x \vec{x}[n] + \vec{\theta_s}$$
$$\vec{r}[n] = G\left(\vec{s}[n]\right)$$

Again, for the stability of equations, the eigenvalues of $\widehat{W} = W_s + W_r$ should lie within the complex-valued unit circle.
Consider the best case scenario, where B is a diagonal matrix. $(B = \Lambda_B)$

$$W_s \approx 0$$
$$\widehat{W} \approx W_r \approx -A^{-1}B$$
$$\widetilde{W} = -A^{-1}\Lambda_B = diag\left(\mu_i\right) \text{ with } \mu_i = \frac{\lambda_i}{|a_{ii}|}$$

A necessary and sufficient condition for stability is that $0 < \mu_i < 1$.
If any μ_i fails to satisfy this condition, the system will be unstable, causing the elements of $\vec{r}[n]$ to enter the flat regions of the warping nonlinearity at some value of index, n.

RNN Unrolling

RNN Unrolling

- $\vec{v}\,[n]$: the sequence of the 'ground truth' output values

  N : the length of the sequence, $\vec{v}\,[n]$
- Assume that $\vec{v}\,[n]\rfloor_{0\leq n\leq N-1}$ is subdivided into M non-overlapping varying-length segments with $K_m\,(\leq N)$ samples per segement.

$$\vec{v}\,[n]\rfloor_{0\leq n\leq N-1} = \sum_{m=0}^{M-1} \vec{v}_m\,[n]$$

$$\vec{v}_m\,[n] = \vec{w}_m[n] \odot \vec{v}\,[n]\rfloor_{0\leq n\leq N-1}$$

$$= \begin{cases} \vec{v}[n], & j(m) \leq n \leq j(m) + K_m - 1 \\ \vec{0}, & \textit{otherwise} \end{cases}$$

$$j(m) = \begin{cases} \sum_{i=0}^{m-1} K_i, & 1 \leq m \leq M-1 \\ 0, & m = 0 \end{cases}$$

$$\Theta \equiv \left\{ W_r, W_x, \vec{\theta}_s \right\}$$

RNN Unrolling

Proposition 1.

Given the standard RNN system parametrized by Θ, assume that there exists a value of Θ, at which the objective function is close to an optimum as measured by some acceptable bound. Further, assume that there exist non-zero finite constants, M and K_m, such that $K_m \leq N$, where $0 \leq m \leq M-1$, and that the ground truth output sequence, $\vec{v}[n]\rfloor_{0 \leq n \leq N-1}$, can be partitioned into mutually independent segment-level ground truth output subsequences. Then a single, reusable RNN cell, unrolled for an adjustable number of steps, K_m, is computationally sufficient for seeking Θ that optimizes the objective function over the training set and for inferring outputs from unseen inputs.

The mutual independence assumption between segments leads to initializing the state signal of each segment to a random vector or to zero.

RNN Unrolling

- Truncated unrolled RNN system

$$\vec{\tilde{s}}\,[n=-1]=\vec{0}$$

$$\vec{\tilde{s}}\,[n]=\begin{cases} W_r\vec{\tilde{r}}\,[n-1]+W_x\vec{\tilde{x}}_m[n]+\vec{\theta_s}, & 0\leq n\leq K_m-1 \\ \vec{0}, & \textit{otherwise} \end{cases}$$

$$\vec{\tilde{r}}\,[n]=\begin{cases} G\left(\vec{\tilde{s}}\,[n]\right), & 0\leq n\leq K_m-1 \\ \vec{0}, & \textit{otherwise} \end{cases}$$

$$\vec{\tilde{x}}_m\,[n]=\begin{cases} \vec{x}\,[n+j(m)], & 0\leq n\leq K_m-1 \\ \vec{0}, & \textit{otherwise} \end{cases}$$

$$0\leq m\leq M-1$$

- According to Proposition 1, the RNN unrolling technique is justified by partitioning a single output sequence into multiple independent subsequences and placing restrictions on the initialization of the state between subsequences. However, adhering to these conditions may be problematic in terms of modeling sequences in practical applications.

RNN Training Difficulties

BPTT

- Truncated unrolled RNN systems are commonly trained using 'Back Propagation Through Time' (BPTT).
- The objective function, E, depends on the readout signal, $\vec{\tilde{r}}[n]$ and takes on the same form for all segments. (now omit tilde for convenience)

$$\vec{\chi}[n] \equiv \vec{\nabla}_{\vec{r}[n]} E = \frac{\partial E}{\partial \vec{r}[n]}$$

$$\vec{\psi}[n] \equiv \vec{\nabla}_{\vec{s}[n]} E = \frac{\partial E}{\partial \vec{s}[n]}$$

$$E = \sum_{n=0}^{K_m - 1} E\left(\vec{r}[n]\right)$$

BPTT

From

$$\begin{aligned}\vec{s}[n+1] &= W_r \vec{r}[n] + W_x \vec{x}_m[n+1] + \vec{\theta}_s \\ \vec{r}[n] &= G\left(\vec{s}[n]\right) \\ \vec{r}[n+1] &= G\left(\vec{s}[n+1]\right),\end{aligned}$$

the total partial derivative of the objective function w.r.t $\vec{r}[n]$ and $\vec{s}[n]$:

$$\begin{aligned}\vec{\chi}[n] &= \frac{\partial E\left(\vec{r}[n]\right)}{\partial \vec{r}[n]} + W_r \vec{\psi}[n+1] \\ \vec{\psi}[n] &= \vec{\chi}[n] \odot \frac{dG\left(\vec{z}\right)}{d\vec{z}} \rfloor_{z=\vec{s}[n]} \\ &= \left(\frac{\partial E\left(\vec{r}[n]\right)}{\partial \vec{r}[n]} + W_r \vec{\psi}[n+1]\right) \odot \frac{dG\left(\vec{z}\right)}{d\vec{z}} \rfloor_{z=\vec{s}[n]}\end{aligned}$$

Gradient Vanishing/Exploding

$$\frac{\partial \vec{\psi}[n]}{\partial \vec{\psi}[l]} = \prod_{k=n+1}^{l} W_r \odot \frac{dG(\vec{z})}{d\vec{z}} \rfloor_{z=\vec{s}[k]}$$

$$\| \frac{\partial \vec{\psi}[n]}{\partial \vec{\psi}[l]} \| \sim \left(\| W_r \| \cdot \| \frac{dG(\vec{z})}{d\vec{z}} \| \right)^{l-n}$$

$$\sim \| W_r \|^{l-n} \cdot \| \frac{dG(\vec{z})}{d\vec{z}} \|^{l-n}$$

- If all eigenvalues of W_r satisfy the requirement for stability, i.e., $0 < \mu_i < 1$, then $\| W_r \| < 1 \Rightarrow$ Gradient Vanishing
- If at least one eigenvalue of W_r violates the requirement for stability, the term $\| W_r \|^{l-n}$ will grow exponentially.
  (a) $\vec{r}[n]$ eventually saturate at the rails (the flat regions) of the warping function $\Rightarrow$ Gradient Vanishing
  (b) $\vec{s}[n]$ is initially biased in the quasi-linear region of the warping function and $\vec{x}_m[n]$ guides the system to stay in this mode for a large number of steps $\Rightarrow$ Gradient Exploding

From RNN to Vanilla LSTM Network

Canonical RNN Formulation

The LSTM network was invented with the goal of addressing the vanishiing gradients problem.
The network cell can be rationalized from the canonical RNN cell.

$$\vec{s}[n] = W_s\vec{s}[n-1] + W_r\vec{r}[n-1] + W_x\vec{x}[n] + \vec{\theta_s}$$
$$\vec{r}[n] = G\left(\vec{s}[n]\right)$$

Several modifications to the cell's design

$$\vec{s}[n] = \vec{\mathcal{F}}_s\left(\vec{s}[n-1]\right) + \vec{\mathcal{F}}_u\left(\vec{r}[n-1], \vec{x}[n]\right)$$
$$\vec{r}[n] = G_d\left(\vec{s}[n]\right)$$
$$\vec{\mathcal{F}}_s\left(\vec{s}[n-1]\right) = W_s\vec{s}[n-1]$$
$$\vec{\mathcal{F}}_u\left(\vec{r}[n-1], \vec{x}[n]\right) = W_r\vec{r}[n-1] + W_x\vec{x}[n] + \vec{\theta_s}$$

$G_d\left(\cdot\right)$ is the hyperbolic tangent.

Gate Signals

According to the previous equations, the state signal blends both sources of information in equal proportions at every step. These proportions can be made adjustable by multiplying the two quantities by the special "gate" signals.

$$\vec{s}[n] = \vec{g}_{cs}[n] \odot \vec{\mathcal{F}}_s\left(\vec{s}[n-1]\right) + \vec{g}_{cu}[n] \odot \vec{\mathcal{F}}_u\left(\vec{r}[n-1], \vec{x}[n]\right)$$
$$\vec{0} \leq \vec{g}_{cs}[n], \vec{g}_{cu}[n] \leq \vec{1}$$

- Further Modifications
  (a) Since $\vec{\mathcal{F}}_s\left(\vec{s}[n-1]\right) = W_s\vec{s}[n-1]$ and W_s is a diagonal matrix, reparametrize $\vec{g}_{cs}[n]$ so that the first term be $\vec{g}_{cs}[n] \odot \vec{s}[n-1]$.
  (b) For the second term, $\vec{s}[n-1]$ and $\vec{s}[n]$ are connected with W_r for every step. This can lead to vanishing/exploding gradients problem.
  $\Rightarrow$ Introduce another gate $\vec{g}_{cr}[n]$ and define $\vec{v}[n] = \vec{g}_{cr}[n] \odot \vec{r}[n]$. Use $\vec{v}[n-1]$ instead of $\vec{r}[n-1]$ in $\vec{\mathcal{F}}_u$. $(\vec{0} \leq \vec{g}_{cr}[n] \leq \vec{1})$

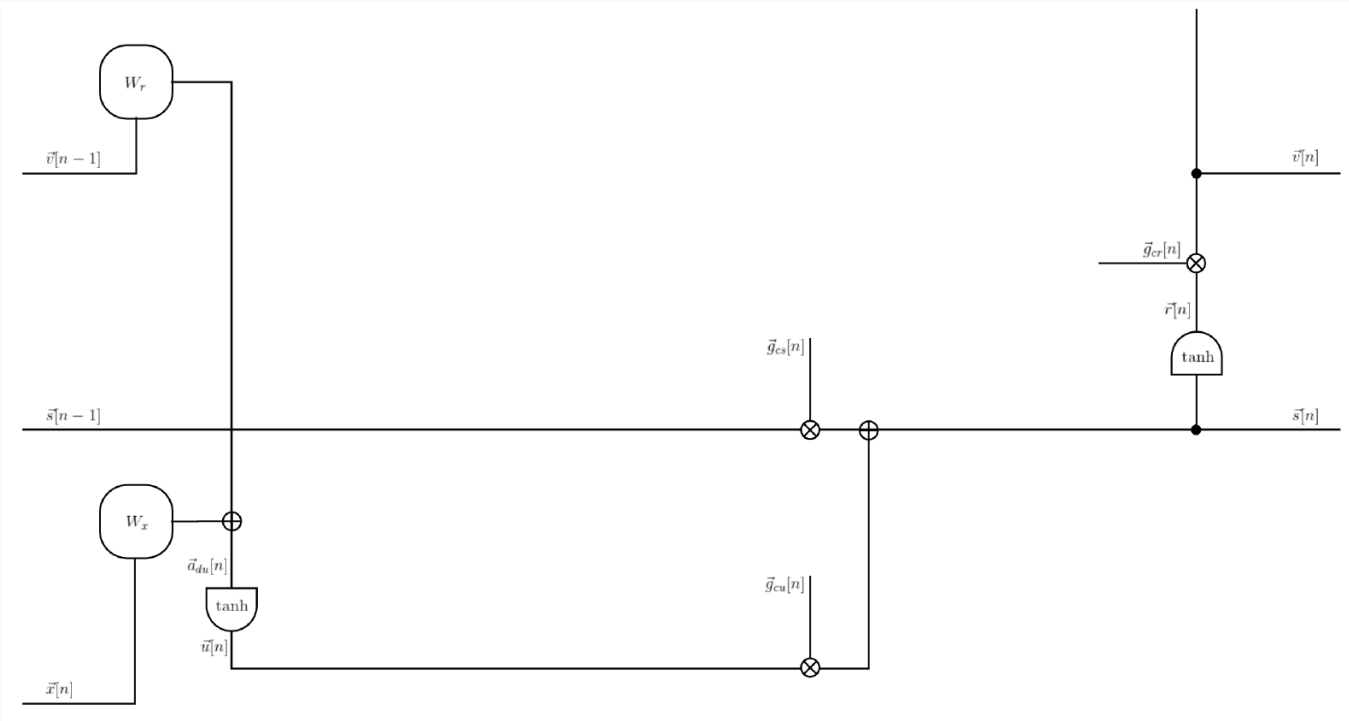
Gate Signals

(c) The external input signal, $\vec{x}[n]$ is multiplied by a 'control gate', $\vec{g}_{cx}[n]$ for the system flexibility.

(d) To maintain the same dynamic range of first two terms in $\vec{\mathcal{F}}_u$, it is tempered by the saturating warping nonlinearity, $G_d(z)$, so as to produce the update candidate signal, $\vec{u}[n]$.

$$
\begin{aligned}
\vec{s}[n] &= \vec{g}_{cs}[n] \odot \vec{s}[n-1] + \vec{g}_{cu}[n] \odot \vec{u}[n] \\
\vec{v}[n] &= \vec{g}_{cr}[n] \odot G_d\left(\vec{s}[n]\right) \\
\vec{\mathcal{F}}_u\left(\vec{v}[n-1], \vec{x}[n]\right) &= W_r \vec{v}[n-1] + \vec{g}_{cx}[n] \odot W_x \vec{x}[n] + \vec{\theta}_s \\
\vec{u}[n] &= G_d\left(\vec{\mathcal{F}}_u\left(\vec{v}[n-1], \vec{x}[n]\right)\right)
\end{aligned}
$$

Gate Signals



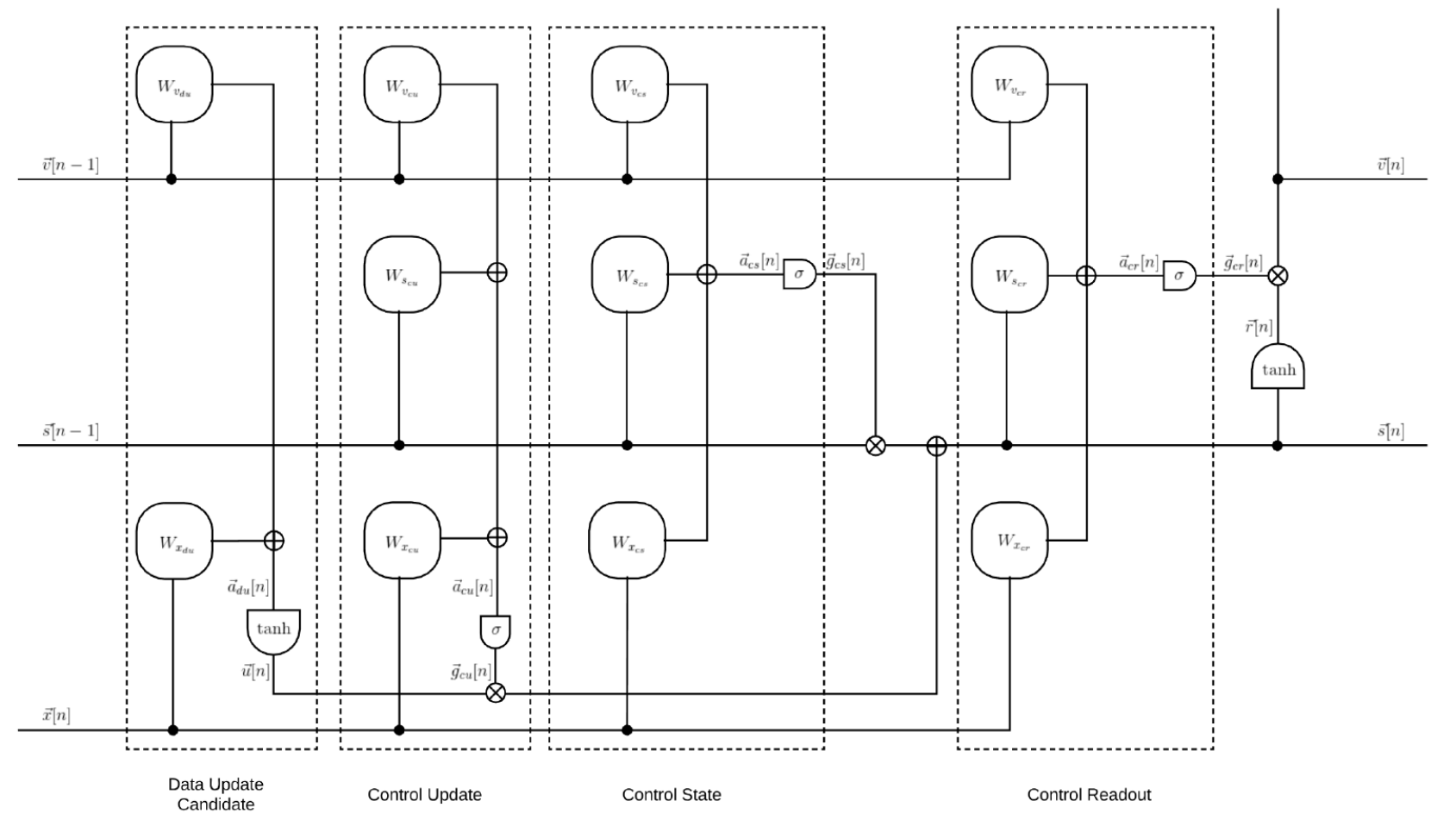
$$G_c(z) \equiv \sigma(z) \equiv \frac{1}{1+e^{-z}}$$

$$\vec{g}_{cs}[n] = G_c\left(W_{x_{cs}}\vec{x}[n] + W_{s_{cs}}\vec{s}[n-1] + W_{v_{cs}}\vec{v}[n-1] + \vec{\theta}_{cs}\right)$$

$$\vec{g}_{cu}[n] = G_c\left(W_{x_{cu}}\vec{x}[n] + W_{s_{cu}}\vec{s}[n-1] + W_{v_{cu}}\vec{v}[n-1] + \vec{\theta}_{cu}\right)$$

$$\vec{g}_{cr}[n] = G_c\left(W_{x_{cr}}\vec{x}[n] + W_{s_{cr}}\vec{s}[n] + W_{v_{cr}}\vec{v}[n-1] + \vec{\theta}_{cr}\right)$$

Vanilla LSTM



The Vanilla LSTM Network Mechanism in Detail

Notations

- n : index of a step in the segment (or subsequence); $n = 0, \ldots, K-1$
- K : number of steps in the unrolled segment (or subsequence)
- G_c : monotonic, bipolarly-saturating warping function for control/throttling purposes (acts as a "gate")
- G_d : monotonic, negative-symmetric, bipolarly-saturating warping function for data bounding purposes
- d_x : dimensionality of the input signal to the cell
- d_s : dimensionality of the state signal of the cell
- $\vec{x} \in \mathbb{R}^{d_x}$: the input signal to the cell
- $\vec{s} \in \mathbb{R}^{d_s}$: the state signal to the cell
- $\vec{v} \in \mathbb{R}^{d_s}$: an accumulation node of the cell
- $\vec{u} \in \mathbb{R}^{d_s}$: the update candidate signal for the state signal of the cell
- $\vec{r} \in \mathbb{R}^{d_s}$: the readout candidate signal of the cell
- $g \in \mathbb{R}^{d_s}$: a gate output signal of the cell for control/throttling purposes
- $E \in \mathbb{R}$: objective function to be minimized as part of the model training procedure

LSTM details

- Data Set Standardization

$$\vec{\mu} = \frac{1}{N}\sum_{n=0}^{N-1}\vec{x}_0[n]$$

$$\mathcal{V} = \frac{1}{N-1}\sum_{n=0}^{N-1}\left(\vec{x}_0[n]-\vec{\mu}\right)\left(\vec{x}_0[n]-\vec{\mu}\right)^T$$

$$\vec{x}\,[n] = \left[diag\left(\sqrt{\mathcal{V}_{ii}}\right)\right]^{-1}\left(\vec{x}_0[n]-\vec{\mu}\right)$$

- Warping (Activation) Functions

$$G_c(z) \equiv \sigma(z) \equiv \frac{1}{1+e^{-z}}$$

$$G_d(z) \equiv tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$$

Vanilla LSTM system Derivatives

$$\frac{dG_c(z)}{dz} = G_c(z)\left(1 - G_c(z)\right)$$
$$\frac{dG_d(z)}{dz} = 1 - \left(G_d(z)\right)^2$$

$$\vec{\chi}[n] \equiv \vec{\nabla}_{\vec{v}[n]} E = \frac{\partial E}{\partial \vec{v}[n]}$$
$$\vec{\rho}[n] \equiv \vec{\nabla}_{\vec{r}[n]} E = \frac{\partial E}{\partial \vec{r}[n]}$$
$$\vec{\gamma}[n] \equiv \vec{\nabla}_{\vec{g}[n]} E = \frac{\partial E}{\partial \vec{g}[n]}$$
$$\vec{\alpha}[n] \equiv \vec{\nabla}_{\vec{a}[n]} E = \frac{\partial E}{\partial \vec{a}[n]}$$
$$\vec{\psi}[n] \equiv \vec{\nabla}_{\vec{s}[n]} E = \frac{\partial E}{\partial \vec{s}[n]}$$

Vanilla LSTM system Derivatives

$$\vec{\chi}[n] = \left(\frac{\partial \vec{y}[n]}{\partial \vec{v}[n]}\right)^T \left(\frac{\partial E}{\partial \vec{y}[n]}\right) + \vec{f}_{\chi}[n+1]$$

$$\vec{\rho}[n] = \left(\frac{\partial \vec{v}[n]}{\partial \vec{r}[n]}\right)^T \left(\frac{\partial E}{\partial \vec{v}[n]}\right) = (\nabla_{\vec{v}[n]} E) \odot \vec{g}_{cr}[n] = \vec{\chi}[n] \odot \vec{g}_{cr}[n]$$

$$\vec{\gamma}_{cr}[n] = \frac{\partial E}{\partial \vec{v}[n]} \frac{\partial \vec{v}[n]}{\partial \vec{g}_{cr}[n]} = (\nabla_{\vec{v}[n]} E) \odot \vec{r}[n] = \vec{\chi}[n] \odot \vec{r}[n]$$

$$\vec{\alpha}_{cr}[n] = \vec{\gamma}_{cr}[n] \odot \frac{\partial \vec{g}_{cr}[n]}{\partial \vec{a}_{cr}[n]} = \vec{\gamma}_{cr}[n] \odot \frac{dG_c(z)}{dz} \rfloor_{z=\vec{a}_{cr}[n]} = \vec{\chi}[n] \odot \vec{r}[n] \odot \frac{dG_c(z)}{dz} \rfloor_{z=\vec{a}_{cr}[n]}$$

where $\vec{f}_{\chi}[n+1] = W_{v_{cu}}^T \vec{\alpha}_{cu}[n+1] + W_{v_{cs}}^T \vec{\alpha}_{cs}[n+1] + W_{v_{cr}}^T \vec{\alpha}_{cr}[n+1] + W_{v_{du}}^T \vec{\alpha}_{du}[n+1]$

Vanilla LSTM system Derivatives

$$\vec{\psi}[n] = \vec{\rho}[n] \odot \frac{\partial \vec{r}[n]}{\partial \vec{s}[n]} + \frac{\partial \vec{a}_{cr}[n]}{\partial \vec{s}[n]} \vec{\alpha}_{cr}[n] + \vec{f}_{\psi}[n+1]$$

$$= \vec{\rho}[n] \odot \frac{dG_d(z)}{dz} \rfloor_{z=\vec{s}[n]} + W_{s_{cr}} \vec{\alpha}_{cr}[n] + \vec{f}_{\psi}[n+1]$$

$$= \vec{\chi}[n] \odot \vec{g}_{cr}[n] \odot \frac{dG_d(\vec{z})}{d\vec{z}} \rfloor_{z=\vec{s}[n]} + W_{s_{cr}} \vec{\alpha}_{cr}[n] + \vec{f}_{\psi}[n+1]$$

$$\vec{\alpha}_{cs}[n] = \vec{\psi}[n] \odot \frac{\partial \vec{s}[n]}{\partial \vec{g}_{cs}[n]} \odot \frac{\partial \vec{g}_{cs}[n]}{\partial \vec{a}_{cs}[n]} = \vec{\psi}[n] \odot \vec{s}[n-1] \odot \frac{dG_c(\vec{z})}{d\vec{z}} \rfloor_{z=\vec{a}_{cs}[n]}$$

$$\vec{\alpha}_{cu}[n] = \vec{\psi}[n] \odot \frac{\partial \vec{s}[n]}{\partial \vec{g}_{cu}[n]} \odot \frac{\partial \vec{g}_{cu}[n]}{\partial \vec{a}_{cu}[n]} = \vec{\psi}[n] \odot \vec{u}[n] \odot \frac{dG_c(\vec{z})}{d\vec{z}} \rfloor_{z=\vec{a}_{cu}[n]}$$

$$\vec{\alpha}_{du}[n] = \vec{\psi}[n] \odot \frac{\partial \vec{s}[n]}{\partial \vec{u}[n]} \odot \frac{dG_d(\vec{z})}{d\vec{z}} \rfloor_{z=\vec{a}_{du}[n]} = \vec{\psi}[n] \odot \vec{g}_{cu}[n] \odot \frac{dG_d(\vec{z})}{d\vec{z}} \rfloor_{z=\vec{a}_{du}[n]}$$

where $\vec{f}_{\psi}[n+1] = W_{s_{cu}}^T \vec{\alpha}_{cu}[n+1] + W_{s_{cs}}^T \vec{\alpha}_{cs}[n+1] + \vec{g}_{cs}[n+1] \odot \vec{\psi}[n+1]$

Error Gradient Sequences in Vanilla LSTM System

$$\vec{\psi}[n] = \vec{\chi}[n] \odot \vec{g}_{cr}[n] \odot \frac{dG_d(\vec{z})}{d\vec{z}}\rfloor_{z=\vec{s}[n]} + W_{s_{cr}}\vec{\chi}[n] \odot \vec{r}[n] \odot \frac{dG_c(z)}{dz}\rfloor_{z=\vec{a}_{cr}[n]} + \vec{f}_{\psi}[n+1]$$

$$= \left(\left(\frac{\partial \vec{y}[n]}{\partial \vec{v}[n]}\right)^T \left(\frac{\partial E}{\partial \vec{y}[n]}\right) + \vec{f}_{\chi}[n+1]\right) \odot \vec{g}_{cr}[n] \odot \frac{dG_d(\vec{z})}{d\vec{z}}\rfloor_{z=\vec{s}[n]}$$

$$+ W_{s_{cr}}\left(\left(\frac{\partial \vec{y}[n]}{\partial \vec{v}[n]}\right)^T \left(\frac{\partial E}{\partial \vec{y}[n]}\right) + \vec{f}_{\chi}[n+1]\right) \odot \vec{r}[n] \odot \frac{dG_c(z)}{dz}\rfloor_{z=\vec{a}_{cr}[n]} + \vec{f}_{\psi}[n+1]$$

$$\frac{\partial \vec{\psi}[k-1]}{\partial \vec{\psi}[k]} = \left(\frac{\partial \vec{\psi}[k-1]}{\partial \vec{f}_{\chi}[k]}\right)\left(\frac{\partial \vec{f}_{\chi}[k]}{\partial \vec{\psi}[k]}\right) + \left(\frac{\partial \vec{\psi}[k-1]}{\partial \vec{f}_{\psi}[k]}\right)\left(\frac{\partial \vec{f}_{\psi}[k]}{\partial \vec{\psi}[k]}\right)$$

$$= \left(\frac{\partial \vec{\psi}[k-1]}{\partial \vec{f}_{\chi}[k]}\right)\left\{\left(\frac{\partial \vec{f}_{\chi}[k]}{\partial \vec{\alpha}_{cu}[k]}\right)\left(\frac{\partial \vec{\alpha}_{cu}[k}{\partial \vec{\psi}[k]}\right) + \left(\frac{\partial \vec{f}_{\chi}[k]}{\partial \vec{\alpha}_{cs}[k]}\right)\left(\frac{\partial \vec{\alpha}_{cs}[k}{\partial \vec{\psi}[k]}\right)\right.$$

$$\left. + \left(\frac{\partial \vec{f}_{\chi}[k]}{\partial \vec{\alpha}_{du}[k]}\right)\left(\frac{\partial \vec{\alpha}_{du}[k}{\partial \vec{\psi}[k]}\right)\right\} + \left(\frac{\partial \vec{\psi}[k-1]}{\partial \vec{f}_{\psi}[k]}\right)\left\{\left(\frac{\partial \vec{f}_{\psi}[k]}{\partial \vec{\alpha}_{cu}[k]}\right)\left(\frac{\partial \vec{\alpha}_{cu}[k}{\partial \vec{\psi}[k]}\right)\right.$$

$$\left. + \left(\frac{\partial \vec{f}_{\psi}[k]}{\partial \vec{\alpha}_{cs}[k]}\right)\left(\frac{\partial \vec{\alpha}_{cs}[k}{\partial \vec{\psi}[k]}\right) + diag\left[\vec{g}_{cs}[k]\right]\right\}$$

Error Gradient Sequences in Vanilla LSTM System

$$
\begin{aligned}
&= \left(diag\left[\vec{g}_{cr}[k-1] \odot \frac{dG_d(\vec{z})}{d\vec{z}} \rfloor_{z=\vec{s}[k-1]} \right] + W_{s_{cr}}\, diag\left[\vec{r}[k-1] \odot \frac{dG_c(\vec{z})}{d\vec{z}} \rfloor_{z=\vec{a}_{cr}[k-1]} \right] \right) \\
&\times \left\{ W_{v_{cu}}\, diag\left[\vec{u}[k] \odot \frac{dG_c(\vec{z})}{d\vec{z}} \rfloor_{z=\vec{a}_{cu}[k]} \right] + W_{v_{cs}}\, diag\left[\vec{s}[k-1] \odot \frac{dG_c(\vec{z})}{d\vec{z}} \rfloor_{z=\vec{a}_{cs}[k]} \right] \right. \\
&\left. + W_{v_{du}}\, diag\left[\vec{g}_{cu}[k] \odot \frac{dG_d(\vec{z})}{d\vec{z}} \rfloor_{z=\vec{a}_{du}[k]} \right] \right\} + \left(W_{s_{cu}}\, diag\left[\vec{u}[k] \odot \frac{dG_c(\vec{z})}{d\vec{z}} \rfloor_{z=\vec{a}_{cu}[k]} \right] \right. \\
&\left. + W_{s_{cs}}\, diag\left[\vec{s}[k-1] \odot \frac{dG_c(\vec{z})}{d\vec{z}} \rfloor_{z=\vec{a}_{cs}[k]} \right] + diag\left[\vec{g}_{cs}[k] \right] \right) \\
&= diag\left[\vec{u}[k] \odot \frac{dG_c(\vec{z})}{d\vec{z}} \rfloor_{z=\vec{a}_{cu}[k]} \right] \\
&\times \left\{ W_{v_{cu}} \left(diag\left[\vec{g}_{cr}[k-1] \odot \frac{dG_d(\vec{z})}{d\vec{z}} \rfloor_{z=\vec{s}[k-1]} \right] + W_{s_{cr}}\, diag\left[\vec{r}[k-1] \odot \frac{dG_c(\vec{z})}{d\vec{z}} \rfloor_{z=\vec{a}_{cr}[k-1]} \right] \right) + W_{s_{cu}} \right\} \\
&+ diag\left[\vec{s}[k-1] \odot \frac{dG_c(\vec{z})}{d\vec{z}} \rfloor_{z=\vec{a}_{cs}[k]} \right] \\
&\times \left\{ W_{v_{cs}} \left(diag\left[\vec{g}_{cr}[k-1] \odot \frac{dG_d(\vec{z})}{d\vec{z}} \rfloor_{z=\vec{s}[k-1]} \right] + W_{s_{cr}}\, diag\left[\vec{r}[k-1] \odot \frac{dG_c(\vec{z})}{d\vec{z}} \rfloor_{z=\vec{a}_{cr}[k-1]} \right] \right) + W_{s_{cs}} \right\} \\
&+ diag\left[\vec{g}_{cu}[k] \odot \frac{dG_d(\vec{z})}{d\vec{z}} \rfloor_{z=\vec{a}_{du}[k]} \right] \\
&\times \left\{ W_{v_{du}} \left(diag\left[\vec{g}_{cr}[k-1] \odot \frac{dG_d(\vec{z})}{d\vec{z}} \rfloor_{z=\vec{s}[k-1]} \right] + W_{s_{cr}}\, diag\left[\vec{r}[k-1] \odot \frac{dG_c(\vec{z})}{d\vec{z}} \rfloor_{z=\vec{a}_{cr}[k-1]} \right] \right) \right\} \\
&+ diag\left[\vec{g}_{cs}[k] \right] \\
&= Q\left(k-1,\, k; \tilde{\Theta} \right) + diag\left[\vec{g}_{cs}[k] \right]
\end{aligned}
$$

Vanishing Gradient in Vanilla LSTM System

$$\frac{\partial \vec{\psi}[n]}{\partial \vec{\psi}[l]} = \prod_{k=n+1}^{l} \frac{\partial \vec{\psi}[k-1]}{\partial \vec{\psi}[k]}$$

- A sufficient condition for driving the residual $\| \frac{\partial \vec{\psi}[n]}{\partial \vec{\psi}[l]} \|$ to zero is maintaining $\| \frac{\partial \vec{\psi}[k-1]}{\partial \vec{\psi}[k]} \| < 1$ at each step with the index, k.
  - (a) $Q\left(k-1, k; \tilde{\Theta}\right) = [0]$ and $\vec{g}_{cs}[k] = \vec{0}$ for all values of the step index, k. This is the case of the network being perpetually 'at rest', which is not interesting from the practical standpoint.
  - (b) $\vec{g}_{cs}[k] \approx \vec{1}$ and $Q\left(k-1, k; \tilde{\Theta}\right) = -diag\left[\vec{g}_{cs}[k]\right]$ for some value of the step index, k. However, satisfying this condition would require a very careful orchestration of all signals.
  - (c) The spectral radius of $\left[Q\left(k-1, k; \tilde{\Theta}\right) + diag\left[\vec{g}_{cs}[k]\right]\right]$ is less than unity for all values of the step index, k. This behavior would not be due to a degenerate mode of the system, but as a consequence of the particular patterns, occurring in the training data. In other words, some dependencies are naturally short-range.

Constant Error Carousel

$$\| \frac{\partial \vec{\psi}[k-1]}{\partial \vec{\psi}[k]} \| \leq \| Q\left(k-1, k; \tilde{\Theta}\right) \| + \| diag\left[\vec{g}_{cs}[k]\right] \|$$

- The most emblematic regime of the LSTM network arises when $\| Q\left(k-1, k; \tilde{\Theta}\right) \| < 1$. The following alternatives create favorable circumstances for this condition to hold :
  - (a) $\| W_{s_{cu}} \| < \frac{1}{2}$, $\| W_{v_{cu}} \| < \frac{1}{2}$, $\| W_{s_{cs}} \| < \frac{1}{2}$, $\| W_{v_{cs}} \| < \frac{1}{2}$, $\| W_{v_{du}} \| < 1$
  - (b) the state signal saturates the readout data warping function, $\| W_{s_{cr}} \| < \frac{1}{2}$, $\| W_{s_{cu}} \| < \frac{1}{2}$, $\| W_{s_{cs}} \| < \frac{1}{2}$
  - (c) the state signal saturates the readout data warping function, the accumulation signal for the control readout gate saturates its control warping function, $\| W_{s_{cu}} \| < \frac{1}{2}$, $\| W_{s_{cs}} \| < \frac{1}{2}$
  - (d) the control readout gate is turned off, $\| W_{s_{cu}} \| < \frac{1}{2}$, $\| W_{s_{cs}} \| < \frac{1}{2}$
  - (e) the accumulation signals for the control update gate and the control state gate saturate their respective control warping functions, the update candidate accumulation signal saturates the update candidate data warping function
  - (f) the control update gate is turned off, the control state gate is turned off

Constant Error Carousel

If $\| Q\left(k-1, k; \tilde{\Theta}\right) \| < 1$, then

$$\| \frac{\partial \vec{\psi}[n]}{\partial \vec{\psi}[l]} \| \sim \prod_{k=n+1}^{l} \| diag\left[\vec{g}_{cs}[k]\right] \| \leq 1 \ .$$

As long as the elements of $\vec{g}_{cs}[n]$ are fractions, the error gradient will naturally decay. However, if the model is trained to saturate $\vec{g}_{cs}[n]$ at $\vec{1}$, then the error gradient is recirculated through Constant Error Carousel.