02/11/23 Deep Learning

- Recap
- GRU
- LSTM

- Recap: BPTT for RNN

$$\theta^{(r)} = \theta^{(r-1)} - \eta \nabla_{\theta^{(r-1)}} R(\theta) \quad (3)$$

$$R(\theta) = \frac{1}{T} \sum_{t=1}^{T} d(y^{(t)}, \hat{y}^{(t)}) \quad (4)$$

Recap:

$$z_m^{(t)} = \sigma(\alpha_m^T \cdot x^{(t)} + \gamma_m^T \cdot z^{(t-1)}) \quad (1)$$

$$\hat{y}_k^{(t)} = g_k(\beta_k^T \cdot z^{(t)}) \quad (2)$$

- $\frac{\partial R(\theta)}{\partial \gamma_{mp}}$ expression ; γ_{mp} is the parameter connecting the p^{th} hidden node at time $t-1$ to the m^{th} hidden node at time t.

$$\frac{\partial R(\theta)}{\partial \gamma_{mp}} = \frac{\partial R(\theta)}{\partial y} \cdot \frac{\partial y}{\partial z_m} \cdot \frac{\partial z_m^{(t)}}{\partial \gamma_{mp}} \quad (5)$$

$$= \frac{\partial z_m^{(t)}}{\partial \gamma_{mp}}$$ is now the expensive computation (6)

- BPTT (full) is expensive computationally
- BPTT also leads to vanishing/exploding gradients
- Truncated BPTT to overcome these issues.

- Gated Recurrent Unit (GRU):

$$z_m^{(t)} = s_m^{(t)} z_m^{(t-1)} + (1 - s_m^{(t)}) \tilde{z}_m^{(t)} \quad (7)$$

$s_m^{(t)}$ ↳ update gate ; $\tilde{z}_m^{(t)}$ ↳ candidate update

$$\tilde{z}_m^{(t)} = \sigma(\alpha_m^T x^{(t)} + r_m^{(t)} \cdot \gamma_m^T \cdot z^{(t-1)}) \quad (8)$$

$$\tilde{z}_m^{(t)} = \tanh\left(\underline{\alpha}_m^T \cdot \underline{x}^{(t)} + \left(r_m^{(t)} \underline{\gamma}_m^T \cdot \underline{z}^{(t-1)}\right)\right) \quad -(9)$$

$r_m^{(t)}$ ↳ reset gate

$$\hat{y}_k^{(t)} = g_k\left(\underline{\beta}_k^T \cdot \underline{z}^{(t)}\right) \quad -(10)$$

RNN:
$$z_m^{(t)} = \sigma\left(\underline{\alpha}_m^T \cdot \underline{x}^{(t)} + \underline{\gamma}_m^T \cdot z_m^{(t)}\right)$$

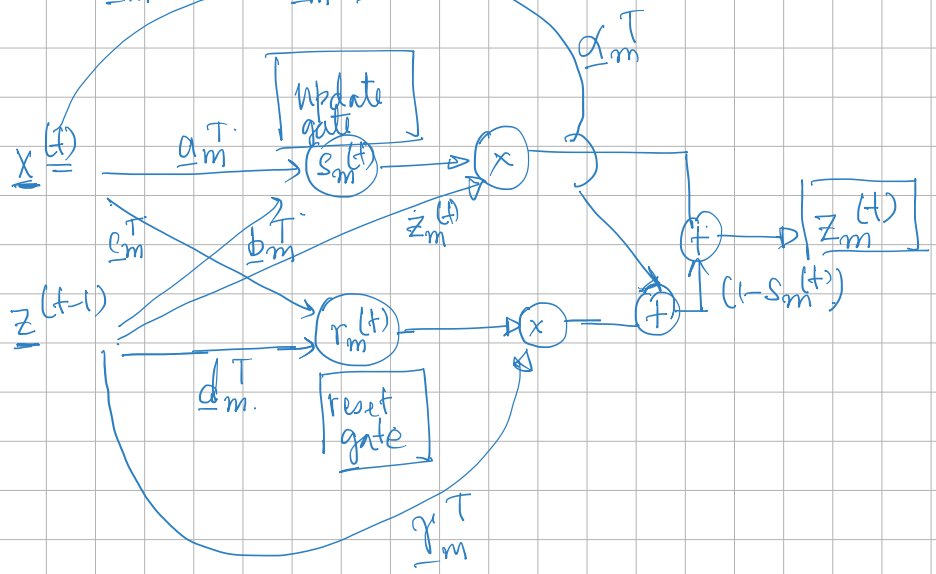
- GRU vs RNN
  - $s_m^{(t)}$ – allows us to weight or give importance to the past state $z_m^{(t-1)}$
    - this in turn allows for paying attention in the long range.
  - $r_m^{(t)}$ – allows us to reset the past hidden state $z_m^{(t-1)}$ or pay lesser importance to it. This allows the model to pay attention to the short term or near term input

∴ We now have both longterm and short term control knobs.

- The added parameters are the update gate $s_m^{(t)}$ & the reset gate $r_m^{(t)}$.
- Both $s_m^{(t)}$ and $r_m^{(t)}$ take values in $[0,1)$

$$s_m^{(t)} = \sigma\left(\underline{a}_m^T \cdot \underline{x}^{(t)} + \underline{b}_m^T \underline{z}^{(t-1)}\right) \quad -(11)$$

$$r_m^{(t)} = \sigma\left(\underline{c}_m^T \cdot \underline{x}^{(t)} + \underline{d}_m^T \cdot \underline{z}^{(t-1)}\right) \quad -(12)$$



- $\theta = \left\{\underline{\alpha}^{(1..M)}, \underline{\beta}^{(1..K)}, \underline{\gamma}^{(1..M)}, \underline{a}^{(1..M)}, \underline{b}^{(1..M)}, \underline{c}^{(1..M)}, d^{(1..M)}\right\}$
- Bias is a part of these parameter vectors $\alpha_{m0}, \beta_{k0}, \gamma_{m0}, a_{m0} \cdots d_{m0}$

- Long-Short Term Memory (LSTM)

- Basic memory unit