Training RNNs

In RNN: Back Propagation Through Time (BPTT).

\[ \hat{s}_t = \Phi(\hat{x}_t \cdot W_x + \hat{s}_{t-1} \cdot W_s) \]

e.g., \[ \hat{y}_t = \tanh(\hat{x}_t \cdot W_x + \hat{s}_{t-1} \cdot W_s) \]

\[ \bar{y}_t = \hat{s}_t \cdot W_y \]

\[ E_t = (d_t - \bar{y}_t)^2 \quad \text{error function is MSE} \]

How does BPTT work?

\[ t = 3 \]

\[ E_3 = (\bar{d}_3 - \bar{y}_3)^2 \]

↓ Calculated output

desired
output

but we also need info from t=2 and t=1.
\[ E_3 = (d_3 - \bar{y}_3)^2 \]

\[ \frac{\partial E_3}{\partial w_y} = \frac{\partial E_3}{\partial \bar{y}_3} \cdot \frac{\partial \bar{y}_3}{\partial w_y} \]

\[ \rightarrow \bar{s}_1 \rightarrow \bar{s}_2 \rightarrow \bar{s}_3 \]

\[ \frac{\partial \bar{s}_1}{\partial w_s} \]

\[ \frac{\partial \bar{s}_2}{\partial w_s} \]

\[ \frac{\partial \bar{s}_3}{\partial w_s} \]

\[ W_s \rightarrow \bar{s}_1 \leftarrow W_s \rightarrow \bar{s}_2 \leftarrow W_s \rightarrow \bar{s}_3 \]

\[ \frac{\partial E_3}{\partial w_s} = \frac{\partial E_3}{\partial \bar{y}_3} \cdot \frac{\bar{y}_3}{\partial w_s} \cdot \frac{\partial \bar{s}_3}{\partial w_s} \]

\[ + \frac{\partial E_3}{\partial \bar{y}_3} \cdot \frac{\partial \bar{y}_3}{\partial w_s} \cdot \frac{\partial \bar{s}_2}{\partial w_s} \cdot \frac{\partial \bar{s}_3}{\partial w_s} \]

\[ + \frac{\partial E_2}{\partial \bar{y}_3} \cdot \frac{\partial \bar{y}_3}{\partial w_s} \cdot \frac{\partial \bar{s}_2}{\partial w_s} \cdot \frac{\partial \bar{s}_1}{\partial w_s} \]
More generally: \[
\frac{\partial E_N}{\partial W_S} = \sum_{i=1}^{N} \frac{\partial E_N}{\partial y_N} \cdot \frac{\partial y_N}{\partial s_i} \cdot \frac{\partial s_i}{\partial W_S}
\]

\[E_3 = (d_3 - y_3)^2\]

\[
\frac{\partial E_3}{\partial W_x} = \frac{\partial E_3}{\partial y_3} \cdot \frac{\partial y_3}{\partial s_3} \cdot \frac{\partial s_3}{\partial W_x} + \frac{\partial E_3}{\partial y_3} \cdot \frac{\partial y_3}{\partial s_2} \cdot \frac{\partial s_2}{\partial W_x} + \frac{\partial E_3}{\partial y_3} \cdot \frac{\partial y_3}{\partial s_1} \cdot \frac{\partial s_1}{\partial W_x}
\]
More generally: \[ \frac{\partial E_N}{\partial W_x} = \sum_{i=1}^{N} \frac{\partial E_N}{\partial y_N} \cdot \frac{\partial y_N}{\partial s_i} \cdot \frac{\partial s_i}{\partial W_x} \]

- **for training**: updating weights every \( N \) steps
  
  mini-batch

- **What happens if we have too many steps?**
  up to 8-10 steps, this BPTT works, but beyond that, the vanishing gradient problem happens. \( \rightarrow \) LSTM is born!
Gradient clipping: avoiding Exploding Gradient Problem.

At each timestep $t$:

$$S = \frac{\partial y}{\partial W_{ij}} > \text{threshold}$$

If so, normalize the gradients.