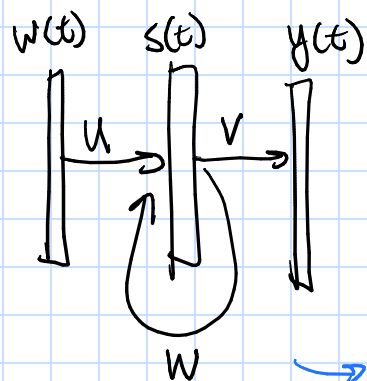


RNNs



$$U \in \mathbb{R}^{D \times H}$$

$$W \in \mathbb{R}^{H \times H}$$

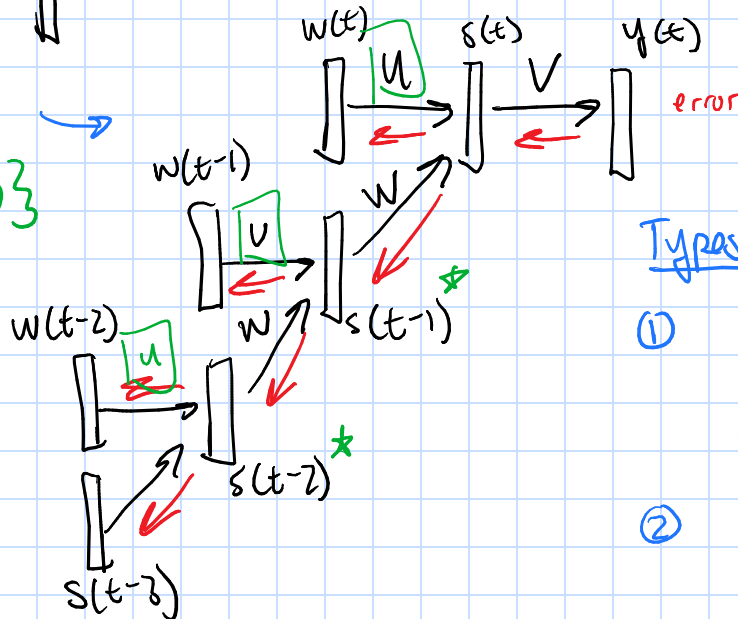
$$V \in \mathbb{R}^{H \times O}$$

$D \rightarrow$ Input dim
 $H \rightarrow$ Hidden dim
 $O \rightarrow$ Output dim

(★ Extra office hour today @ 5-6 PM)

- Weights are found using backprop through time

$$Pr\{w(t+1) | w(t), s(t-1)\}$$



Types of tasks

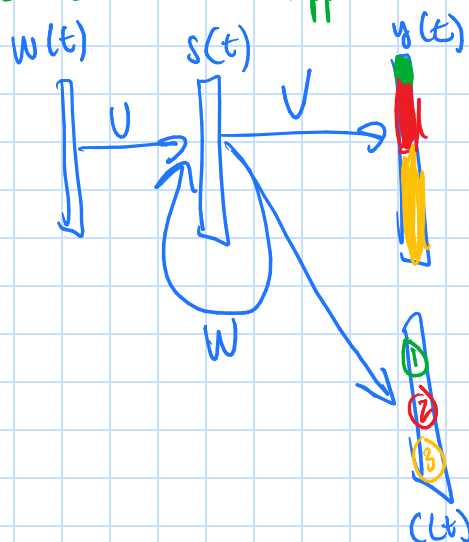
- ① $(x_1, x_2, x_3, \dots, x_t)$
 \downarrow
 $(y_1, y_2, y_3, \dots, y_t)$
- ② $(x_1^{(i)}, x_2^{(i)}, x_3^{(i)}, \dots, x_t^{(i)})$

y_j

Training Advice

- Don't propagate errors back through all time
- Use regularization (L_1, L_2 , Dropout)
- Gradients can explode \therefore Clip gradients $(-5, 5)$
- Complexity for large models

- Same words appear more frequently than others (Zipf's law)



- Only update the weights of the words inside of the class

$$Pr\{w(t+1) | s(t)\} = Pr\{c_i | s(t)\}$$

$$\cdot Pr\{w_i | c_i, s(t)\}$$

Evaluating a Language Models

Perplexity

$$P = 2^{\left\{ \frac{1}{N} \sum_{i=1}^N \log_2 q(w_i) \right\}}$$

$w_i \rightarrow \text{correct}$ $q(w_j) > q(w_i)$
 $w_j \rightarrow \text{choose}$

Generative Adversarial Network

Nets: G, D

