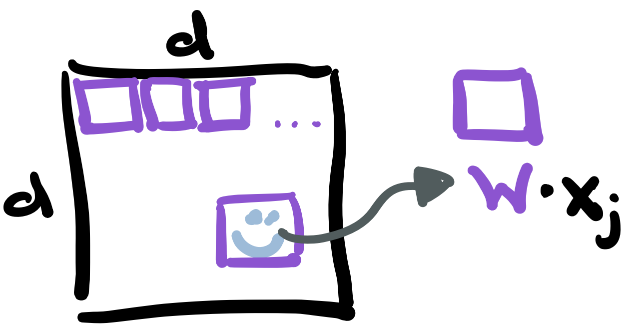
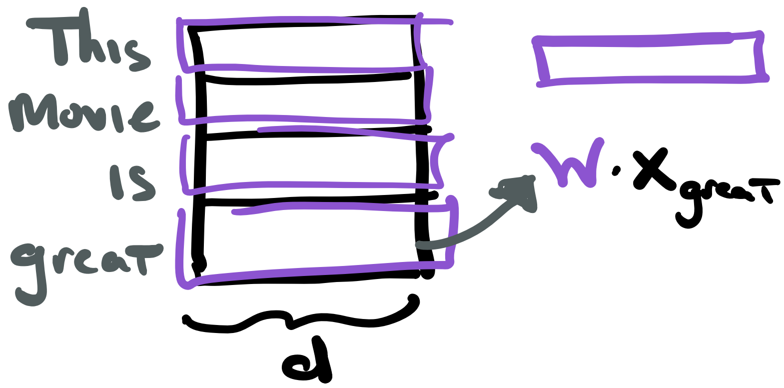
DS 4440 Recurrent Neural Networks

Last few lectures... ConvNets





These filters are position invariant.

But data often is sequentially structured — e.g.

- Language
- Genetic code
- Physiological data
- Stock prices

$$x_t \sim P(x_t \mid x_{t-1} \dots x_1)$$

$$y_t = f(x_t \dots x_1)$$

Key Issue

How can we model an arbitrary length Sequence with a finite set of parameters?

Markov Assumption

$$P(x_\tau \mid x_{\tau-1} \ldots x_1) = P(x_\tau \mid x_{\tau-1} \ldots x_{t-k})$$

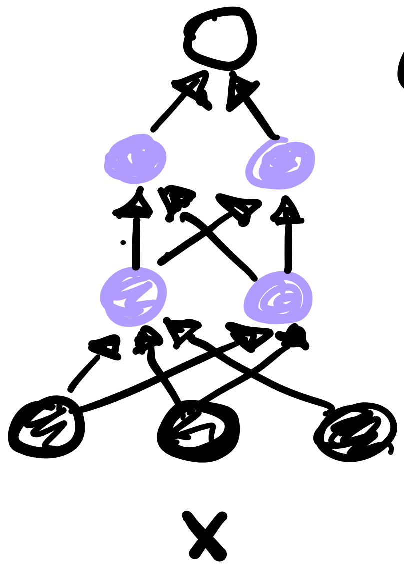
RNNs learn to induce a fixed length representation of $x_1 \ldots x_T$.

Example Learning to count in Variable length Sequences

$$\underbrace{\begin{bmatrix} [a\ b\ b] \\ [b\ b\ a\ a\ a\ b] \\ [a\ a\ b\ a\ b\ b\ b\ b] \end{bmatrix}}_{X} \quad \underbrace{\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}}_{Y} \begin{cases} 1 & \text{if Count(b) is even} \\ 0 & \text{otherwise} \end{cases}$$

Unclear how we could model this with MLPs or ConvNets. We need **State**.

MLPs have **hidden units**

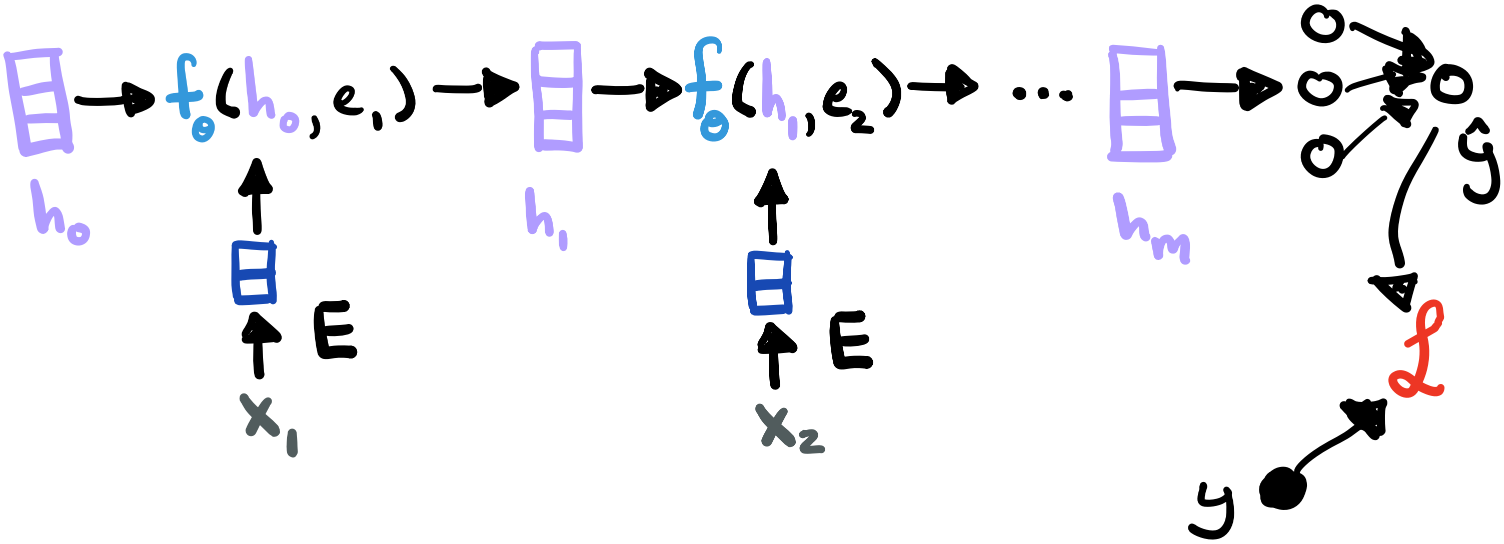


Q. Why doesn't this suffice?

A. Need to process X in order, update h at each step.

RNNs

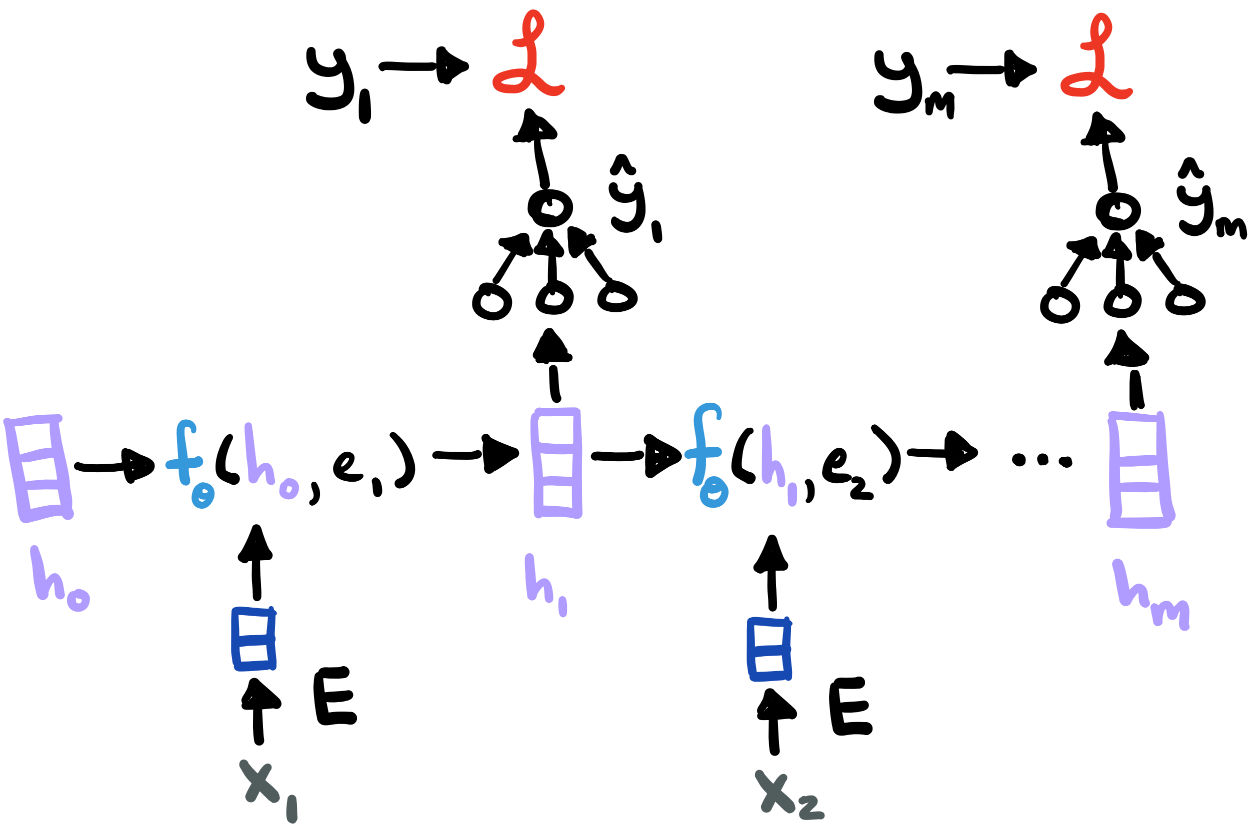
$X = \{x_1, x_2, \ldots x_m\}, \quad y \in \mathbb{R}$



f_θ is <u>shared</u>.

Sequence Tagging: $\hat{y}$ at each τ.

$x = \{x_1, x_2, \dots x_m\}$, $y = \{y_1, y_2 \cdots y_m\}$



Formally (batched)

$$x_\tau \in \mathbb{R}^{\overset{\text{batch size}}{b} \times \overset{\text{input dims}}{d}} \qquad h_\tau \in \mathbb{R}^{b \times \overset{\text{hidden size}}{h}}$$

$$\underset{(b\times h)}{h_{\tau+1}} \leftarrow \sigma(\underset{(b\times d)(d\times h)}{x_\tau W_x} + \underset{(b\times h)(h\times h)}{h_\tau W_h} + \underset{(1\times h)}{b_h})$$

(See Colab Notebook)

Backprop through time

Assume Simple Model

$$\boxed{\hat{y} = W^o \cdot h_3}$$

$$\begin{tikzcd}
& h_0 \arrow[d] \\
x_1 \arrow[r] & \boxed{W^h} \arrow[r] & h_1 \arrow[dl] \\
x_2 \arrow[r] & \boxed{W^h} \arrow[r] & h_2 \arrow[dl] \\
x_3 \arrow[r] & \boxed{W^h} \arrow[r] & h_3 \arrow[r] & \boxed{W^o} \arrow[r] & \hat{y} \arrow[r] & \mathcal{L} & y \arrow[l]
\end{tikzcd}$$

$$\nabla_{W^o} \mathcal{L} = \frac{\partial \mathcal{L}}{\partial \hat{y}} \cdot \nabla_{W^o} \hat{y}$$

What about ∇_{W^h}?

$$\nabla_{W^h} = \sum_{\tau=1}^{T} \nabla_{W^h} h_\tau \cdot \underbrace{\nabla_{h_\tau} \mathcal{L}}$$

$$\nabla_{h_3} \mathcal{L} = \frac{\partial \mathcal{L}}{\partial \hat{y}} \cdot \underline{\nabla_{h_3} \hat{y}} = W^o \in \mathbb{R}^d$$

But ∇_{h_2} More Complicated

$$\nabla_{h_2}\mathcal{L} = \underbrace{(\nabla_{h_2} h_3)}_{\downarrow}\cdot(\nabla_{h_3}\mathcal{L})$$

$$\begin{array}{cc} d & \begin{bmatrix} \frac{\partial h_{3,1}}{\partial h_{2,1}} & \cdots & \frac{\partial h_{3,1}}{\partial h_{2d}} \\ & \ddots & \\ \frac{\partial h_{3d}}{\partial h_{21}} & & \frac{\partial h_{3d}}{\partial h_{2d}} \end{bmatrix} \\ & d \end{array}$$

Jacobian

Problem long sequences require repeat multiplies. Consider

$$\nabla_{h_1}\mathcal{L} = (\nabla_{h_1} h_2)\cdot(\nabla_{h_2} h_3)\cdot(\nabla_{h_3}\mathcal{L})$$

... And this is a toy example.

Gradients ∇ tend to $\boxed{\text{Explode}}$ or VANISH.

Can address Explosions via gradient clipping:

If $\text{Norm}(\nabla_\theta) \geq \tau$

$$\nabla_\theta \leftarrow \nabla_\theta / z$$