Setting Up

$X^{(i)<t>}$ ith training sample

$T_x^{(i)}$ Length of ith training sample $0 \le t \le T_x^{(i)}$

$y^{(i)<t>}$

$T_y^{(i)}$

Vocabulary	One-hot Encoding
Word 1	0 0 1 0 0
Word 2	0 0 0 1 0
⋮	
Word 10,000	0 0 0 0 1

<UNK>

Top 10,000 Most Common Words.

Why not NN?

Standard NN Problems

1) Inputs, Outputs Diff lengths
2) Not Share features learned across diff positions of text

Long-term Dependency

Activation Functions

$a^{<0>} = \vec{0}$ (A Fake Vector As Start)

$$a^{<t>} = g(W_{aa} a^{<t-1>} + W_{ax} X^{<t>} + b_a)$$

$$\hat{y}^{<t>} = g(W_{ya} a^{<t>} + b_y)$$

Simplified Vector Form

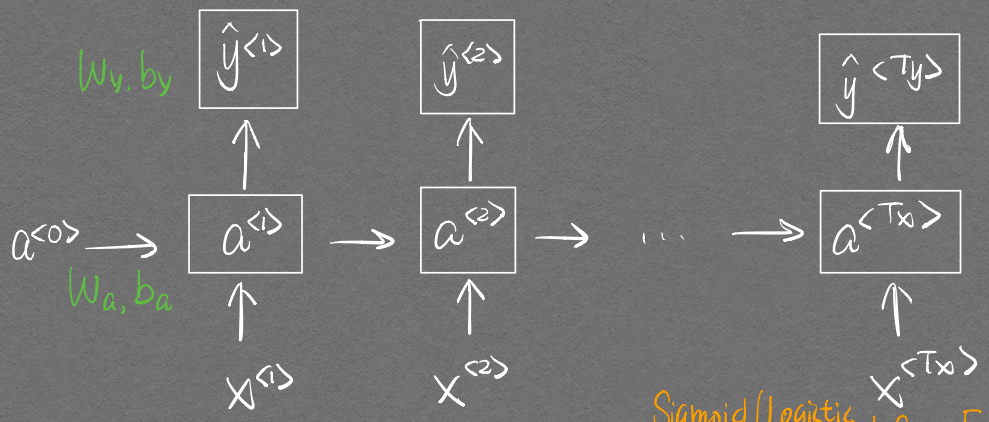
$$a^{<t>} = g(W_a [a^{<t-1>}, X^{<t>}] + b_a)$$

$$\begin{bmatrix} W_{aa} & W_{ax} \end{bmatrix} \begin{bmatrix} a^{<t-1>} \\ \hline X^{<t>} \end{bmatrix} = W_{aa} a^{<t-1>} + W_{ax} X^{<t>}$$

$$\hat{y}^{<t>} = g(W_y a^{<t>} + b_y)$$

Forward Propogation & Backward Propagation

$$\mathcal{L} = \sum \left(\mathcal{L}^{\langle 1 \rangle}, \mathcal{L}^{\langle 2 \rangle}, \dots, \mathcal{L}^{\langle T_y \rangle} \right)$$



Sigmoid / Logistic | Cross Entropy

$$\mathcal{L}^{\langle t \rangle}(\hat{y}^{\langle t \rangle}, y^{\langle t \rangle}) = -y^{\langle t \rangle} \log \hat{y}^{\langle t \rangle} - (1 - y^{\langle t \rangle}) \log (1 - \hat{y}^{\langle t \rangle})$$

$$\mathcal{L}(\hat{y}, y) = \sum_{i=1}^{T_y} \mathcal{L}^{\langle t \rangle}(\hat{y}^{\langle t \rangle}, y^{\langle t \rangle})$$

Diff Types of RNNs

- Speech Recognition
- → Music Generation
- Sentiment Classification
- DNA Sequence
- Machine Translation
- Video Activity Recognition
- Name Entity Recognition

$(T_x = T_y = 1)$ (One To One)

$T_x = T_y$ Many To Many

$T_x \neq T_y$ Encoder → Decoder

$T_x > T_y$

$T_y = 1$ Many To One

$T_x < T_y$

$T_x = 1$ One To Many.

Language Model & Sequence Generation

P("The Apple & Pair Salad") < P("The Apple & Pear Salad")

P(Sentence) = ?

Language Model $P(y^{<1>}, y^{<2>}, \cdots, y^{<T_y>})$

Training Set: Large Corpus of English Text

Tokenize

One-hot Encoding

(Add <EOS> Tag)

<UNK> Tag / <OOV> Tag

Sampling Novel Sequences

P(word 1)

P(word 2)

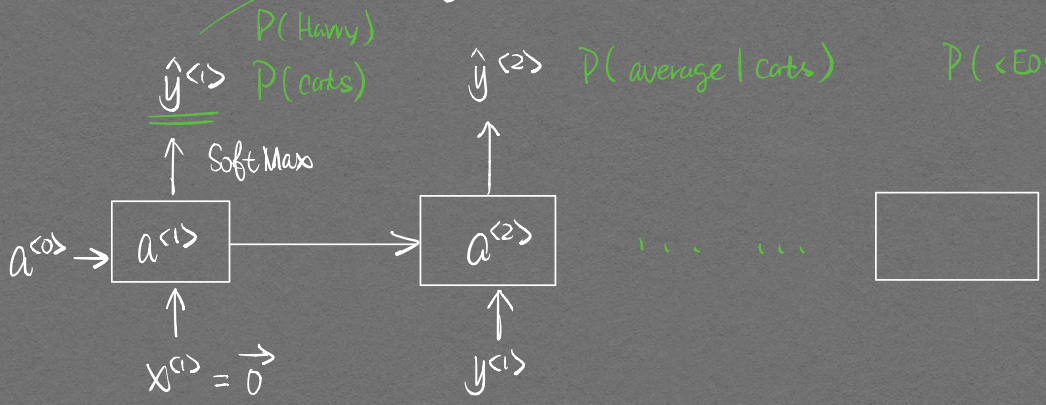
⋮

P(<UNK>)

A Distribution Over Vocabulary

np.random.choice

While <UNK>:
Regenerate Words



$$\mathcal{L}(\hat{y}^{<t>}, y^{<t>}) = -\sum_i y_i^{<t>} \log \hat{y}_i^{<t>}$$ Soft Max Loss

$$\mathcal{L} = \sum_t \mathcal{L}^{<t>}(\hat{y}^{<t>}, y^{<t>})$$

$$P(y_1, y_2, y_3) = P(y_1) \cdot P(y_2 | y_1) \cdot P(y_3 | y_1, y_2)$$

Language Model predicts Probabilities

Character Language Model

A B ... Z

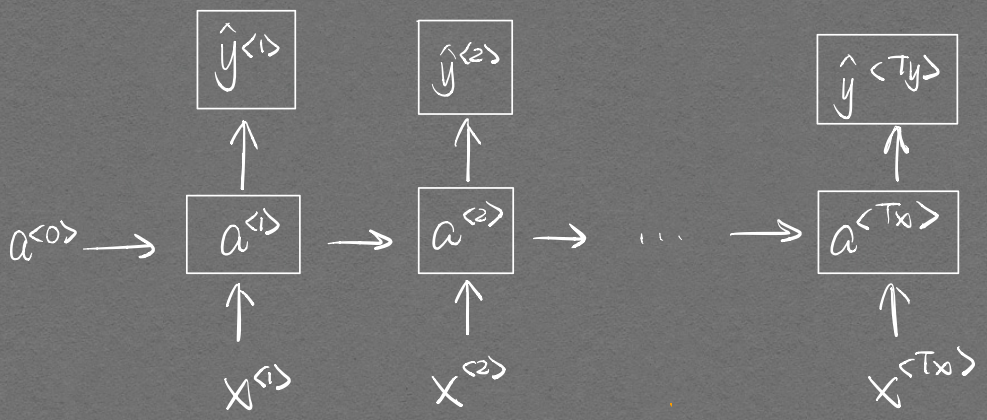
a b ... z

No <UNK> Word.

Computationally Expensive.

Specialized Applications

Vanishing Gradients With RNN



Back Propagation Doesn't work well on Earlier Inputs

Long-term Dependence → Long-term Memory

Requires

"Gradient Clipping" ← Exploding Gradients

Gradients

Increase Exponentially — Blow-up Params → NAN's

$g/\|g\|$ Normalize Too Large Gradients — Numerical Overflow

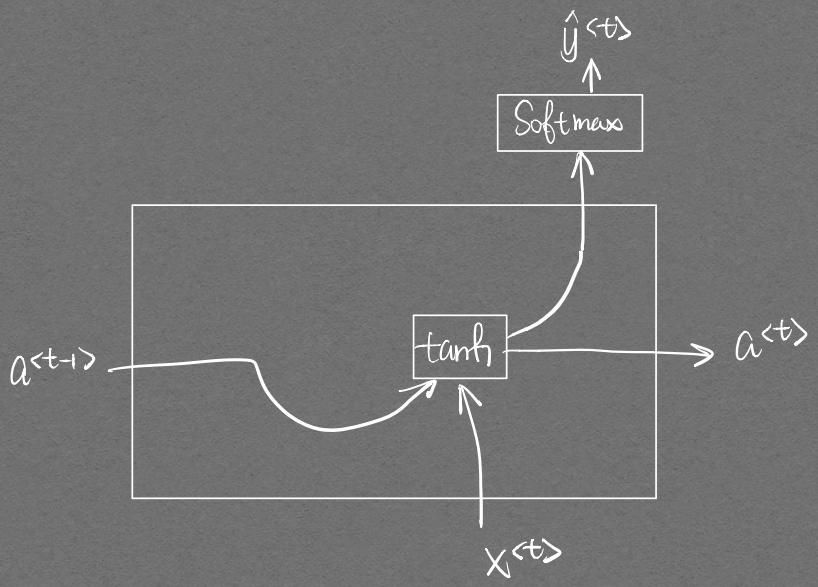
Decrease Exponentially

→ GRU

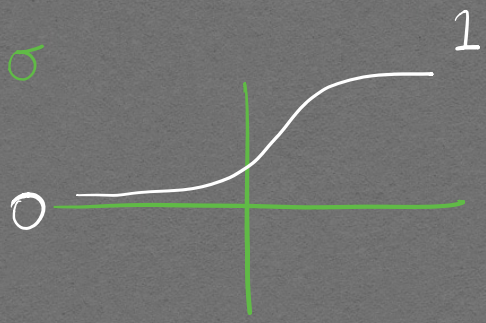
→ LSTM ← Vanishing Gradients

RNN Unit $a^t = g(W_a [a^{<t-1>}, x^{<t>}] + b_a)$

$\hat{y}^{<t>} = g(W_y a^{<t>} + b_y)$



Simplified GRU Unit



c = memory cell

$c^{<t>} = a^{<t>}$

$\tilde{c}^{<t>} = \tanh(W_c [c^{<t-1>}, x^{<t>}] + b_c)$

Candidate to Replace c

$\Gamma_u = \sigma(W_u [c^{<t-1>}, x^{<t>}] + b_u)$ Easy To Set to 0

$= $ Sigmoid 0/1

(Γ_u = Gamma u) "Update Gate" Update or not

$c^{<t>} = \Gamma_u \cdot \hat{c}^{<t>} +$

$(1 - \Gamma_u) \cdot \hat{c}^{<t-1>}$

$$= \begin{cases} \tilde{c}^{\langle t \rangle}, & \Gamma_u = 1 \quad \text{Update} \\ \hat{c}^{\langle t-1 \rangle}, & \Gamma_u = 0 \quad \text{Don't Update} \end{cases}$$

Maintaining Memory

or close to 0

very very small 1E-6

GRU

Full GRU Unit

c = memory cell

$c^{\langle t \rangle} = a^{\langle t \rangle}$

$$\tilde{c}^{\langle t \rangle} = \tanh(W_c[\Gamma_r \cdot c^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_c)$$

Candidate to Replace c

Γ_r = relevance "how relevant $c^{\langle t-1 \rangle}$ is to compute $c^{\langle t \rangle}$"

$$\Gamma_u = \sigma(W_u[c^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_u)$$

(Γ_u = Gamma u) "Update Gate" Update or not A vector of dimension equal to the number of hidden units

$$c^{\langle t \rangle} = \Gamma_u \cdot \hat{c}^{\langle t \rangle} + (1 - \Gamma_u) \cdot \hat{c}^{\langle t-1 \rangle}$$

$$= \begin{cases} \hat{c}^{\langle t \rangle}, & \Gamma_u = 1 \quad \text{Update} \\ \hat{c}^{\langle t-1 \rangle}, & \Gamma_u = 0 \quad \text{Don't Update} \end{cases}$$

$$\Gamma_r = \sigma(W_r[c^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_r)$$

GRU

$$\tilde{c}^{\langle t \rangle} = \tanh(W_c[\Gamma_r \cdot c^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_c)$$

$$\Gamma_u = \sigma(W_u[c^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_u)$$

$$\Gamma_r = \sigma(W_r[c^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_r)$$

LSTM

$$\tilde{c}^{\langle t \rangle} = \tanh(W_c[a^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_c)$$

$$\Gamma_u = \sigma(W_u[a^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_u)$$

Update

$$\Gamma_f = \sigma(W_f[a^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_f)$$

$$c^{\langle t \rangle} = \Gamma_u \cdot \tilde{c}^{\langle t \rangle} + (1 - \Gamma_u) \cdot \tilde{c}^{\langle t-1 \rangle}$$

$$a^{\langle t \rangle} = c^{\langle t \rangle}$$

Two Gates

Cheap

Scalable

Forget

$$\Gamma_o = \sigma(W_o [a^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_o)$$

Output

$$c^{\langle t \rangle} = \Gamma_u \cdot \tilde{c}^{\langle t \rangle} + \Gamma_f \cdot c^{\langle t-1 \rangle}$$

$$a^{\langle t \rangle} = \Gamma_o \cdot \tanh(c^{\langle t \rangle})$$

Three Gates

Expensive

Flexible

Powerful

LSTM

$$\tilde{c}^{\langle t \rangle} = \tanh(W_c [a^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_c)$$

$$\Gamma_u = \sigma(W_u [a^{\langle t-1 \rangle}, x^{\langle t \rangle}, c^{\langle t-1 \rangle}] + b_u)$$

Update

"Peephole Connection"

$$\Gamma_f = \sigma(W_f [a^{\langle t-1 \rangle}, x^{\langle t \rangle}, c^{\langle t-1 \rangle}] + b_f)$$

Forget

$$\Gamma_o = \sigma(W_o [a^{\langle t-1 \rangle}, x^{\langle t \rangle}, c^{\langle t-1 \rangle}] + b_o)$$

Output

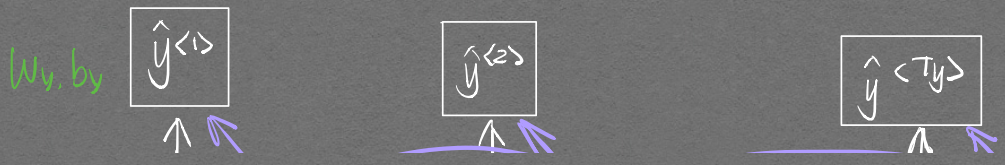
$$c^{\langle t \rangle} = \Gamma_u \cdot \tilde{c}^{\langle t \rangle} + \Gamma_f \cdot c^{\langle t-1 \rangle}$$

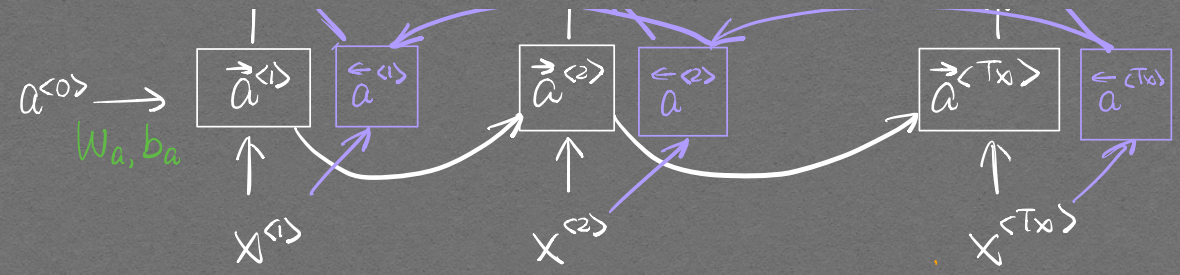
$$a^{\langle t \rangle} = \Gamma_o \cdot c^{\langle t \rangle}$$

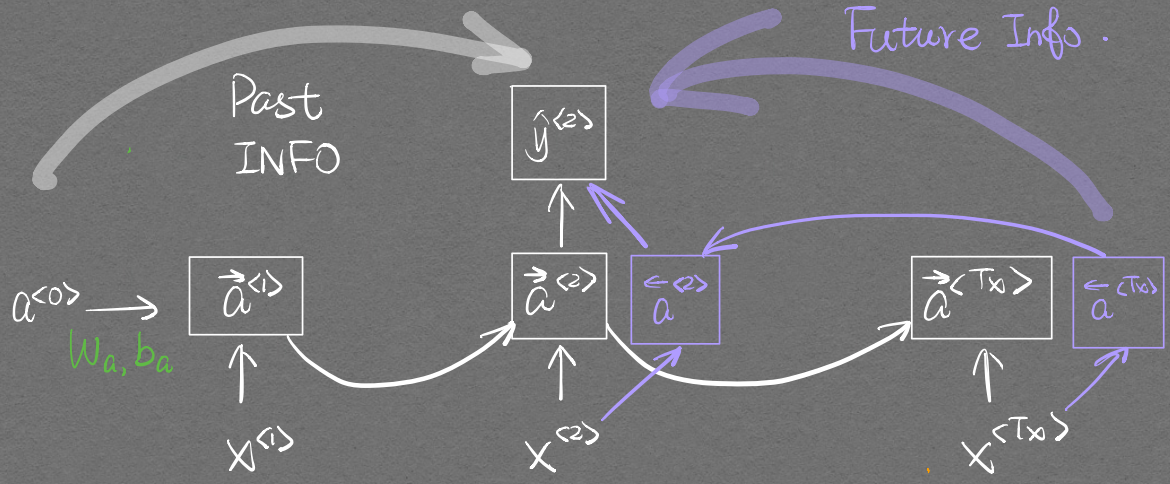
Bi-LSTM

Getting Information From Future

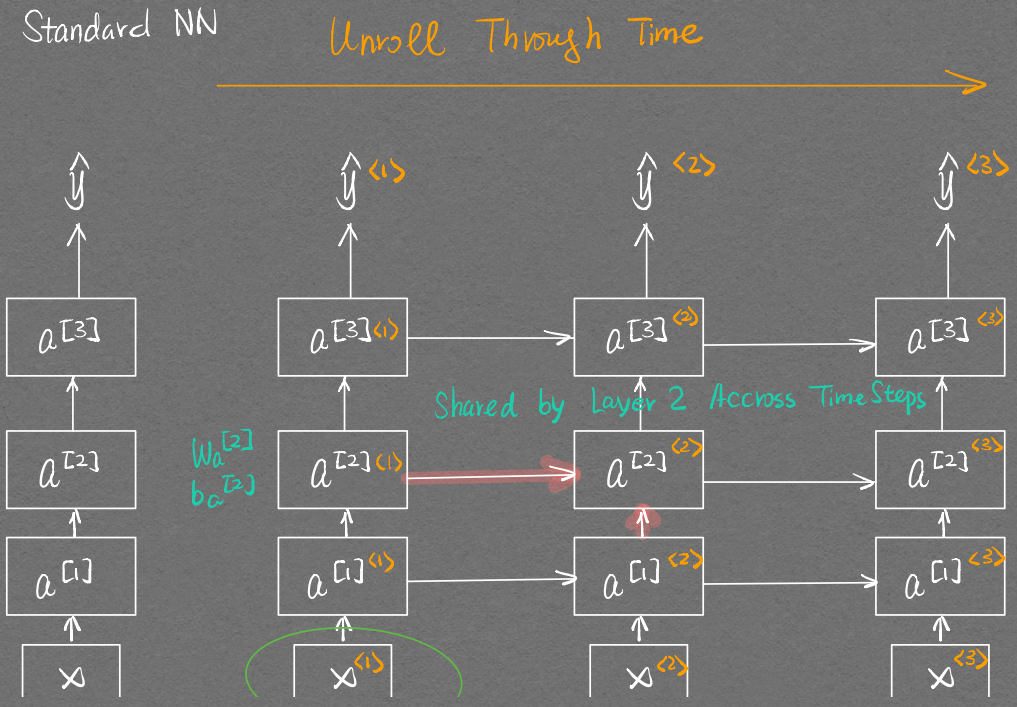
Acyclic Graph

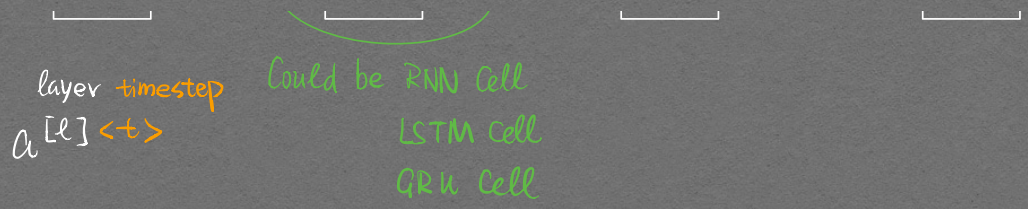



$a^{<0>}$
W_a, b_a
$\vec{a}^{<1>}$
$\overleftarrow{a}^{<1>}$
$\vec{a}^{<2>}$
$\overleftarrow{a}^{<2>}$
$\vec{a}^{<T_x>}$
$\overleftarrow{a}^{<T_x>}$
$x^{<1>}$
$x^{<2>}$
$x^{<T_x>}$


Past INFO
Future Info.
$\hat{y}^{<2>}$
$a^{<0>}$
W_a, b_a
$\vec{a}^{<1>}$
$\vec{a}^{<2>}$
$\overleftarrow{a}^{<2>}$
$\vec{a}^{<T_x>}$
$\overleftarrow{a}^{<T_x>}$
$x^{<1>}$
$x^{<2>}$
$x^{<T_x>}$

Deep-RNN


Standard NN
Unroll Through Time
$\hat{y}$
$\hat{y}^{<1>}$
$\hat{y}^{<2>}$
$\hat{y}^{<3>}$
$a^{[3]}$
$a^{[3]<1>}$
$a^{[3]<2>}$
$a^{[3]<3>}$
Shared by Layer 2 Accross Time Steps
$W_a^{[2]}$
$b_a^{[2]}$
$a^{[2]}$
$a^{[2]<1>}$
$a^{[2]<2>}$
$a^{[2]<3>}$
$a^{[1]}$
$a^{[1]<1>}$
$a^{[1]<2>}$
$a^{[1]<3>}$
x
$x^{<1>}$
$x^{<2>}$
$x^{<3>}$



$$a^{[2]\langle 2\rangle} = g\left(W_a^{[2]}\left[a^{[2]\langle 1\rangle}, a^{[1]\langle 2\rangle}\right] - b_a^{[2]}\right)$$

hidden State a:

The activation $a^{\langle t\rangle}$ that is passed to the RNN from one time step to another is called a "hidden state".

Loop Structure:

1. forward pass
2. cost computation
3. backward pass
4. parameter update