

Whether you need to predict a next word or a label - LSTMs are here to help!

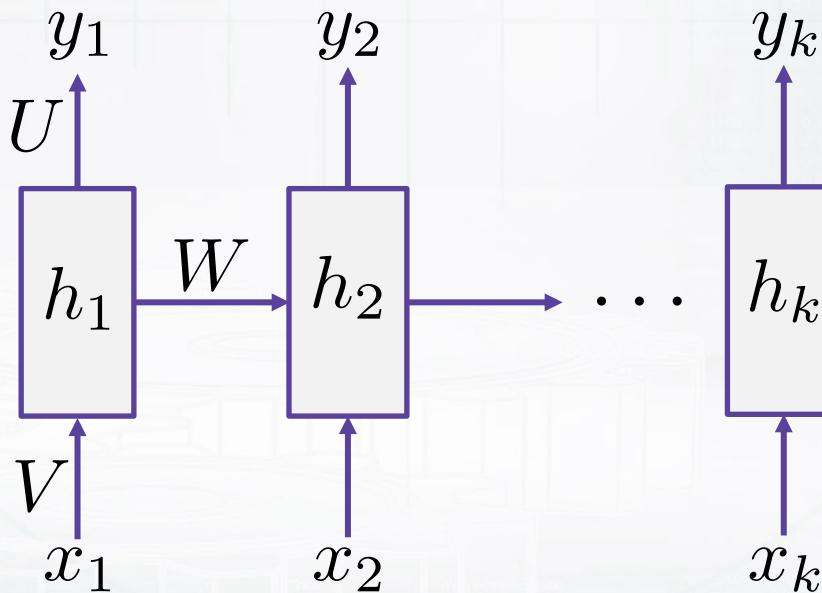


Recap: Recurrent Neural Networks

- Extremely popular architecture for any sequential data:

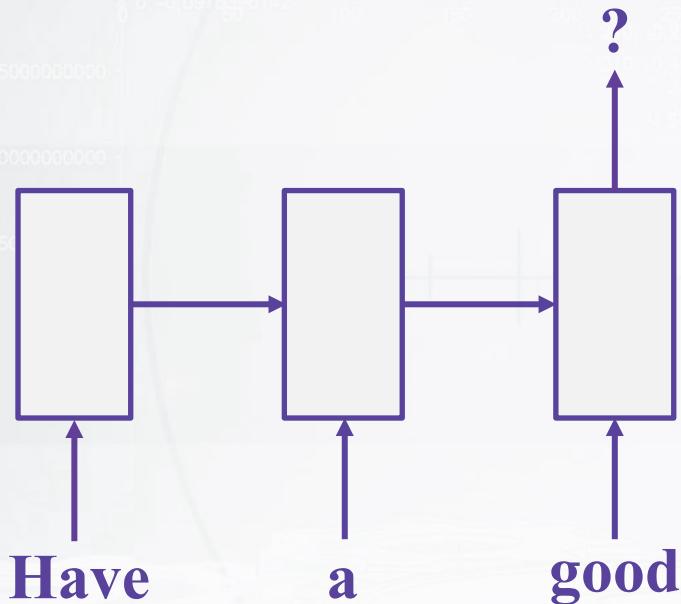
$$h_i = f(Wh_{i-1} + Vx_i + b)$$

$$y_i = Uh_i + \tilde{b}$$



RNN Language Model

- Predicts a next word based on a previous context



Architecture:

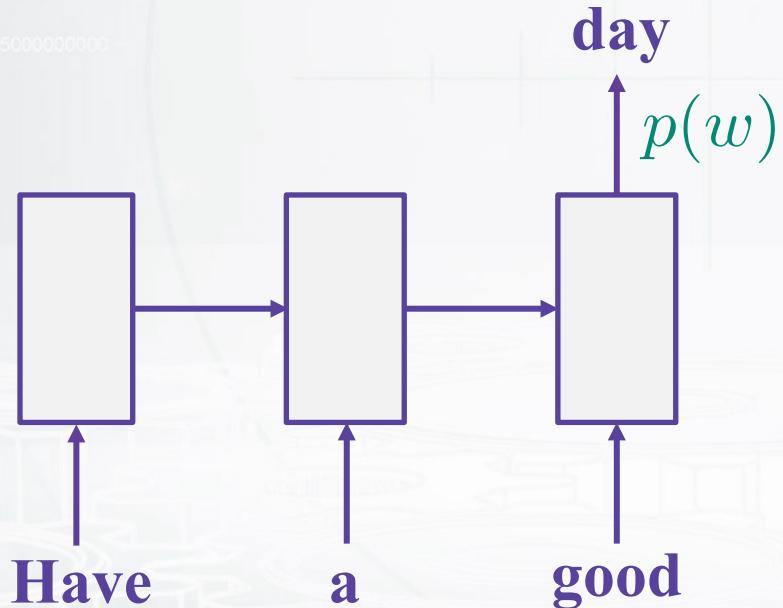
- Use the current state output
- Apply a linear layer on top
- Do *softmax* to get probabilities

Mikolov, Karafiat, Burget, Cernocky, and Khudanpur. Recurrent neural network based language model. INTERSPEECH 2010.

How do we train it?

Cross-entropy loss (for one position):

$$-\log p(w_i) = - \sum_{w \in V} [w = w_i] \log p(w)$$



Only one non-zero

- Target: word w_i
- Output: probabilities $p(w)$

How do we use it to generate language?

500

1.0000000000

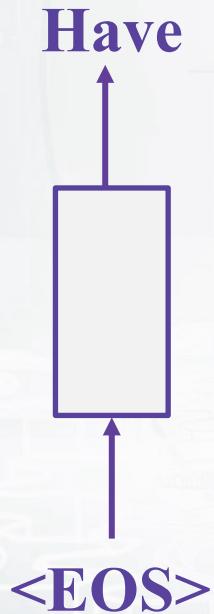
Idea:

- Feed the previous output as the next input
- Take *argmax* at each step (greedily) or use *beam search*

How do we use it to generate language?

Idea:

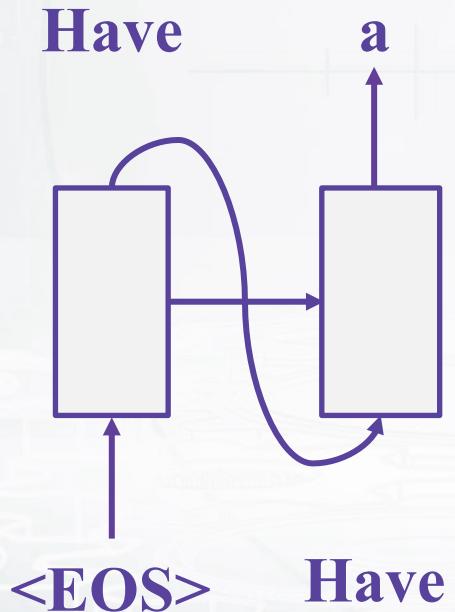
- Feed the previous output as the next input
- Take *argmax* at each step (greedily) or use *beam search*



How do we use it to generate language?

Idea:

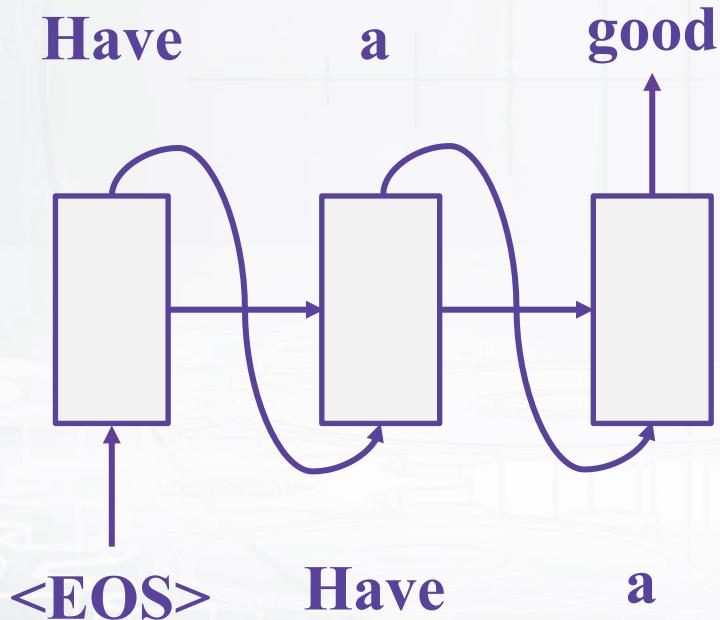
- Feed the previous output as the next input
- Take *argmax* at each step (greedily) or use *beam search*



How do we use it to generate language?

Idea:

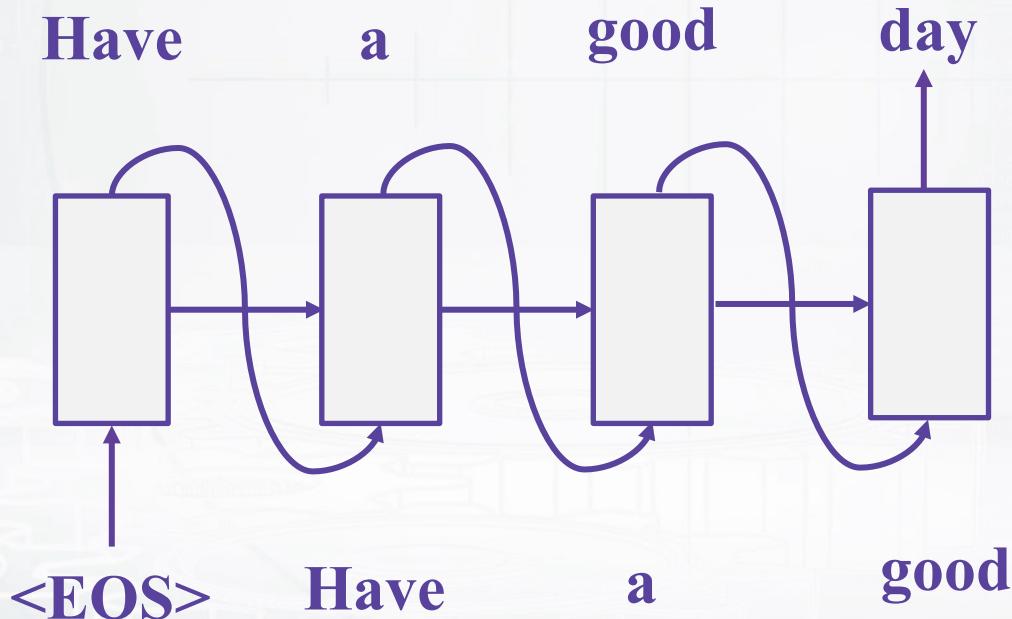
- Feed the previous output as the next input
- Take *argmax* at each step (greedily) or use *beam search*



How do we use it to generate language?

Idea:

- Feed the previous output as the next input
- Take *argmax* at each step (greedily) or use *beam search*



RNN Language Model

- RNN-LM has lower *perplexity* and *word error rate* than 5-gram model with Kneser-Ney smoothing.
- The experiment is held on Wall Street Journal corpus:

Model	# words	PPL	WER
KN5 LM	200K	336	16.4
KN5 LM + RNN 90/2	200K	271	15.4
KN5 LM	1M	287	15.1
KN5 LM + RNN 90/2	1M	225	14.0
KN5 LM	6.4M	221	13.5
KN5 LM + RNN 250/5	6.4M	156	11.7

- Later experiments: char-level RNNs can be very effective!

Mikolov, Karafiat, Burget, Cernocky, and Khudanpur. Recurrent neural network based language model. INTERSPEECH 2010.

Character-level RNN: Shakespeare example

PANDARUS:

Alas, I think he shall be come approached and the day
When little strain would be attain'd into being never fed,
And who is but a chain and subjects of his death,
I should not sleep.

Second Senator:

They are away this miseries, produced upon my soul,
Breaking and strongly should be buried, when I perish
The earth and thoughts of many states.

DUKE VINCENTIO:

Well, your wit is in the care of side and that.

Second Lord:

They would be ruled after this chamber, and
my fair nues begun out of the fact, to be conveyed,
Whose noble souls I'll have the heart of the wars.

Cook your own Language Model

- Use LSTMs or GRUs, and gradient clipping

<https://colah.github.io/posts/2015-08-Understanding-LSTMs/>

- Start with one layer, then stack 3-4, use skip connections
- Use dropout for regularization:

Zaremba, Sutskever, Vinyals. Recurrent Neural Network Regularization, 2014.

- Have a look into TF tutorial for a working model:
<https://www.tensorflow.org/tutorials/recurrent>
- Tune learning rate schedule in SGD or use Adam
- Explore state-of-the-art improvements:
 - **July 2017:** On the State of the Art of Evaluation in Neural Language Models.
 - **August 2017:** Regularizing and Optimizing LSTM Language Models.

Sequence tagging tasks

- Part-of-Speech tagging
- Named Entity Recognition
- Semantic Role Labelling

- ...

BIO-notation:

- B – beginning, I – inside, O – outside

Book a table for 3 in Domino's pizza

Sequence tagging tasks

- Part-of-Speech tagging
- Named Entity Recognition
- Semantic Role Labelling

- ...

BIO-notation:

- B – beginning, I – inside, O – outside

B_ACT	I_ACT	I_ACT	O	B_NUM_PER	O	B_LOC	I_LOC	
Book	a	table	for	3		in	Domino's	pizza

Bi-directional LSTM

- Universal approach for sequence tagging
- You can stack several layers + add linear layers on top
- Trained by cross-entropy loss coming from each position

