CSCE 771: Computer Processing of Natural Language
Lecture 11: Language Models, Word Representations

PROF. BIPLAV SRIVASTAVA, AI INSTITUTE

22ND SEPTEMBER, 2022

Carolinian Creed: "I will practice personal and academic integrity."

Acknowledgement: Used materials by Jurafsky & Martin,

Organization of Lecture 11

- Opening Segment
  - Project – List and In-Class Updates
  - Announcements
- Main Lecture

Main Section
- Language models and prob. parsing connections
- RNN
- AutoEncoders
- Exercise

- Concluding Segment
  - About Next Lecture – Lecture 12

10	Sep 20 (Tu)	Review: Machine Learning for NLP, Evaluation – Metrics
11	Sep 22 (Th)	Prelim. for Language Model – Vector embeddings, CNN/ RNN
12	Sep 27 (Tu)	Guest Lecture – Dr. Amitava Das: Glove, Word2Vec, Transformer
13	Sep 29 (Th)	Representation: Ontology, Knowledge Graph, QUIZ
14	Oct 4 (Tu)	Representation: Embeddings, Language Models
15	Oct 6 (Th)	Entity extraction
16	Oct 11 (Tu)	Guest Lecture – Dr. Amitava Das: Using lang models to solve NLP tasks

Review of Lecture 10

- ML– Supervised
- ML - Unsupervised
- Neural Networks: general familiarity

Project Name
Water - South Carolina
Evolving Firearm Regulations
Target aspect based sentiment analysis for urban neighborhoods
Extracting synthesis procedure from solar cell perovskite based scientific publications.
Entity Recognition : Water Data Regulations
TOS: Banks' Terms of Services summary
Water Regulation Summarization
Predicting the 2022 gubernatorial election of South Carolina using sentiment analysis of Twitter.
Scientific Artical Summarization
New FastText [with Election data]
Chatbot to answer quesries regarding WHO Water Regulations
Verifying various foods connection to improve diabetes using NLP techniques
Summarization of Terms and conditions
Chatbot for Elections FAQ - State of Mississippi
Image Captioning using Transformer Models
Specialist Doctor Recommendation System
Application of Artificial Neural Networks (ANN) to Automatic Speech Recognition (ASR) on a Novel Dataset created using YouTube
Detecting and rating severity of urgency in short, one-time crisis events vs. ongoing ones
Water Regulations - Arizona
Damaged doc. prediction (10%)
Visual Question Answering

Project List

Project Update Presentations

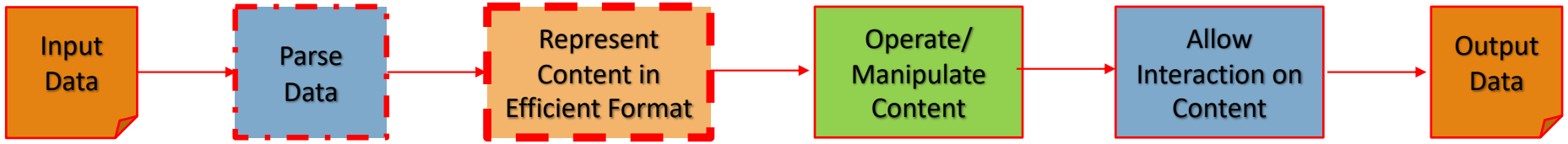
- W1 - Sep 26
- W2 – Oct 3
  - **Review presentation for class: 3 min each – Oct 4, 2022**
- W3 – Oct 10
- W4 – Oct 17
- W5 – Oct 24
  - **Review presentation for class: 3 min each – Oct 27, 2022**
- W6 – Oct 31
- W7 – Nov 7
- W8 – Nov 14
- W9 – Nov 21

Milestones

- Penalty: **not** ready by Sep 15, 2022 **[-20%]**
- Project report **not** ready by Nov 10, 2022 **[-20%]**
- Project presentations **not** ready by Nov 15, 2022 **[-10%]**

Main Lecture

Representation



Connection: PCFG and Language Model

The probability of an ambiguous sentence S is the sum of the probabilities of all the parse trees for the sentence:

$$P(S) = \sum_{T\,s.t.\,S=\text{yield}(T)} P(T,S)$$

$$= \sum_{T\,s.t.\,S=\text{yield}(T)} P(T)$$

But a PCFG also assigns a probability to the substrings of a sentence

From Jurafsky & Martin

Language Model

Problem:
Given a sentence fragment, predict what word(s) come next

Applications:
- Spelling correction
- speech recognition
- machine translation,
- …

Language Model:
estimate probability of substrings of a sentence

$$P(w_i|w_1, w_2, \ldots, w_{i-1}) = \frac{P(w_1, w_2, \ldots, w_{i-1}, w_i)}{P(w_1, w_2, \ldots, w_{i-1})}$$

Bigram approximation

$$P(w_i|w_1, w_2, \ldots, w_{i-1}) \approx \frac{P(w_{i-1}, w_i)}{P(w_{i-1})}$$

From Jurafsky & Martin

Language Model

Markovify library
https://github.com/jsvine/markovify

Language Model:
estimate probability of substrings of a sentence

$$P(w_i|w_1, w_2, \ldots, w_{i-1}) = \frac{P(w_1, w_2, \ldots, w_{i-1}, w_i)}{P(w_1, w_2, \ldots, w_{i-1})}$$

See code samples with Markovify library on Github

- *Prepare data – two datasets shown*
- *Try generator:*
  - *https://github.com/biplav-s/course-nl/blob/master/l7-language/code/TryMarkovifyLangModel.ipynb*

Preq-requisites for Understanding Advanced Language Models

- Advanced language models need pre-requisites to understand
  - BERT, Transformers, GPT-2 and GPT-3

- Understand word representation
- Understand context representation
- Understand machine learning/ neural methods

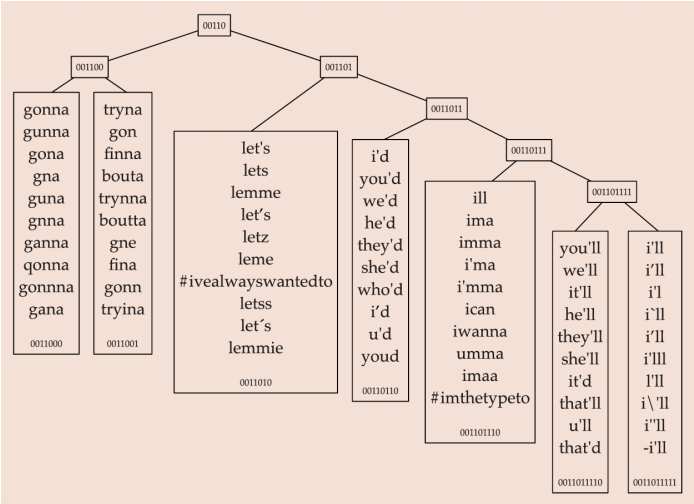
Commentary: http://jalammar.github.io/illustrated-gpt2/

Contextual Word Embeddings

- Words as discrete
- Words with distributional assumptions:
  - Context: given a word, its nearby words or sequences of words
  - Words used in similar ways are likely to have related meanings; i.e., words used in the same (similar) context have related meanings
    - No claim about meaning except relative similarity v/s dis-similarity of words

Contextual Representation by Clustering

- Cluster words by context
- Compare with words in a manually-created taxonomy, e.g., Wordnet



The 10 most frequent words in clusters in the section of the hierarchy with prefix bit string 00110.

Owoputi, O., O'Connor, B., Dyer, C., Gimpel, K., Schneider, N., and Smith, N.A. Improved part-ofspeech tagging for online conversational text with word clusters. In Proceedings of 2013 NAACL.

Credit:
Contextual Word Representations: Putting Words into Computers", by Noah Smith, CACM June 2020

Contextual Representation by Dimensionality Reduction

- Creating word vectors in which each dimension corresponds to the frequency the word type occurred in some context.
- Strategy 1: select contexts
  - Examples
    - Custom methods
    - TF-IDF
  - Approach
    - Use words
      - Words in the neighborhood
      - Words of specific types
    - Build vectors
    - Use vector operations to derive meaning

Credit:
Contextual Word Representations: Putting Words into Computers", by Noah Smith, CACM June 2020

context words	v(astronomers)	v(bodies)	v(objects)
't			1
,		2	1
.	1		1
1			1
And			1
Belt			1
But	1		
Given			1
Kuiper			1
So	1		
and		1	
are		2	1
between			1
beyond		1	
can			1
contains		1	
from	1		
hypothetical			1
ice		1	
including		1	
is	1		
larger		1	
now	1		
of	1		

$$\text{cosine\_similarity}(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \cdot \|\mathbf{v}\|}$$

	astronomers	bodies	objects
astronomers	$\frac{14}{\sqrt{14} \cdot \sqrt{14}} = 1$	$\frac{0}{\sqrt{24} \cdot \sqrt{14}} = 0$	$\frac{1+1}{\sqrt{14} \cdot \sqrt{16}} \approx 0.134$
bodies		$\frac{24}{\sqrt{24} \cdot \sqrt{24}} = 1$	$\frac{2+2+2}{\sqrt{24} \cdot \sqrt{16}} \approx 0.306$
objects			$\frac{16}{\sqrt{16} \cdot \sqrt{16}} = 1$

Bodies and **objects** are **most similar** (0.306) than
- **Bodies** and **astronomers** (0)
- **Objects** and **astronomers** (0.134)

Contextual Representation by Dimensionality Reduction - 1

- Strategy 2: learn contexts from documents. Vector size is given as input

- Train a neural network to learn vector representation
  - value placed in each dimension of each word type's vector is a parameter that will be optimized
  - Selection of parameter values is done using iterative algorithms / gradient descent
  - **Hope** is that ***different senses*** in which a word is used will be captured through the learning procedure as long as the dataset is large enough to represent all senses. Paper quotes: 30 meanings of **get**

- **Optionally**: Sometime task specific inputs are given during pre-processing, processing or post-processing

Disadvantage: individual dimensions are no longer interpretable

Contextual Representation by Dimensionality Reduction -2

- Strategy 2: learn contexts from documents. Vector size is given as input

Sometime task specific inputs are given during pre-processing, processing or post-processing

- Pre-processing
  - Vector initialization by pre-training. Called **finetuning**
- Processing
  - **Knowledge-infusion** (emerging area)
- Post-processing
  - Adjust output vectors so that word types that are related in reference taxonomy (like WordNet) are closer to each other in vector space. Called **retrofitting**.

Credit:
Contextual Word Representations: Putting Words into Computers", by Noah Smith, CACM June 2020

Recap: Language Model

Problem:
Given a sentence fragment, predict what word(s) come next

Language Model:
estimate probability of substrings of a sentence

$$P(w_i|w_1, w_2, \ldots, w_{i-1}) = \frac{P(w_1, w_2, \ldots, w_{i-1}, w_i)}{P(w_1, w_2, \ldots, w_{i-1})}$$

Bigram approximation

$$P(w_i|w_1, w_2, \ldots, w_{i-1}) \approx \frac{P(w_{i-1}, w_i)}{P(w_{i-1})}$$

Applications:
- Spelling correction
- speech recognition
- machine translation,
- …

From Jurafsky & Martin

Recap: Logistic Regression in a Slide

Function estimate (linear)
W: weight, b: bias

$$f(X_j) = X_j W + b$$

Update Weight

$$W^* = W - \eta \frac{dL}{dW}$$

Error Term (mean squared error)

$$MSE = \frac{1}{n} \sum_{j=1}^{n} \left[f(X_{j\cdot}) - y_j \right]^2$$

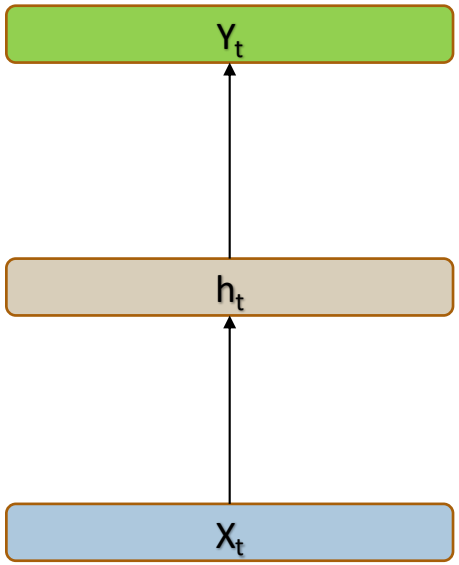
Common Code Pattern

```
y = tf.matmul(x, W) + b
loss = tf.reduce_mean(tf.square(y - y_label))
```

(Feed forward) NN

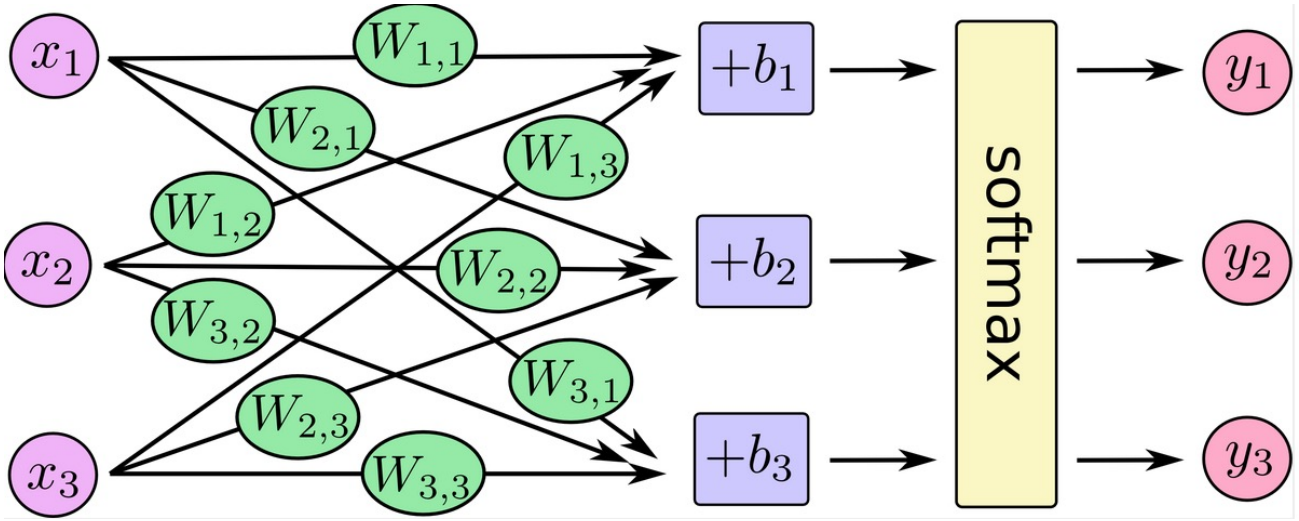
Propoagation

$$f(X_j) = X_j W + b$$



Intuitive Description: https://jalammar.github.io/visual-interactive-guide-basics-neural-networks/, https://jalammar.github.io/feedforward-neural-networks-visual-interactive

Using (Feed forward) NN



Softmax

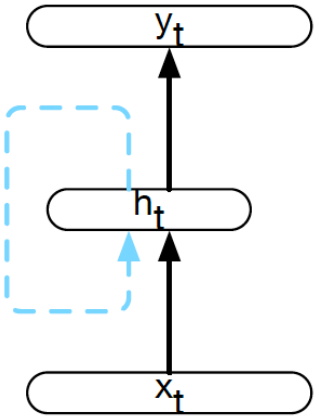
$$f(x)= \frac{1}{1+e^{-x}}$$

Source; see also : https://jalammar.github.io/visual-interactive-guide-basics-neural-networks/, https://jalammar.github.io/feedforward-neural-networks-visual-interactive

RNN - Recurrent Neural Networks

- **Recurrence**: A *recurrence* relation is an equation that defines a sequence based on a rule that gives the next term as a *function* of the previous term(s). [https://mathinsight.org/definition/recurrence_relation]

- Simple Recurrent NN or Elman Network

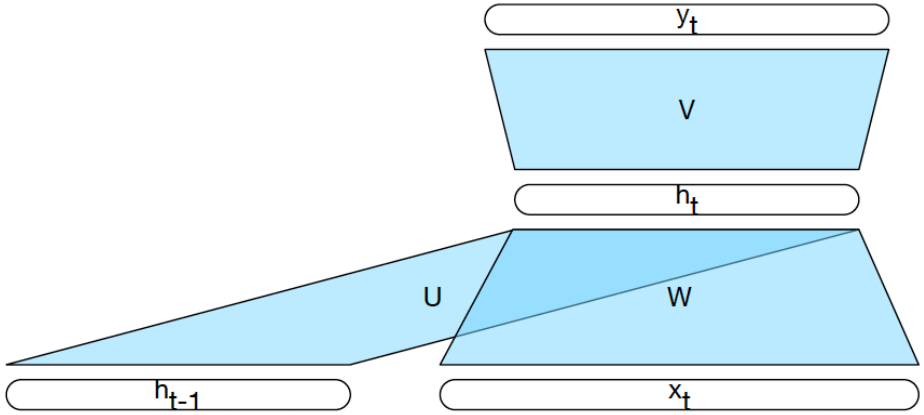


Source: Jurafsky and Martin

RNN

Recurrence unrolled

U, W, V are Weights to be learned



$$h_t = g(Uh_{t-1} + Wx_t)$$
$$y_t = f(Vh_t)$$

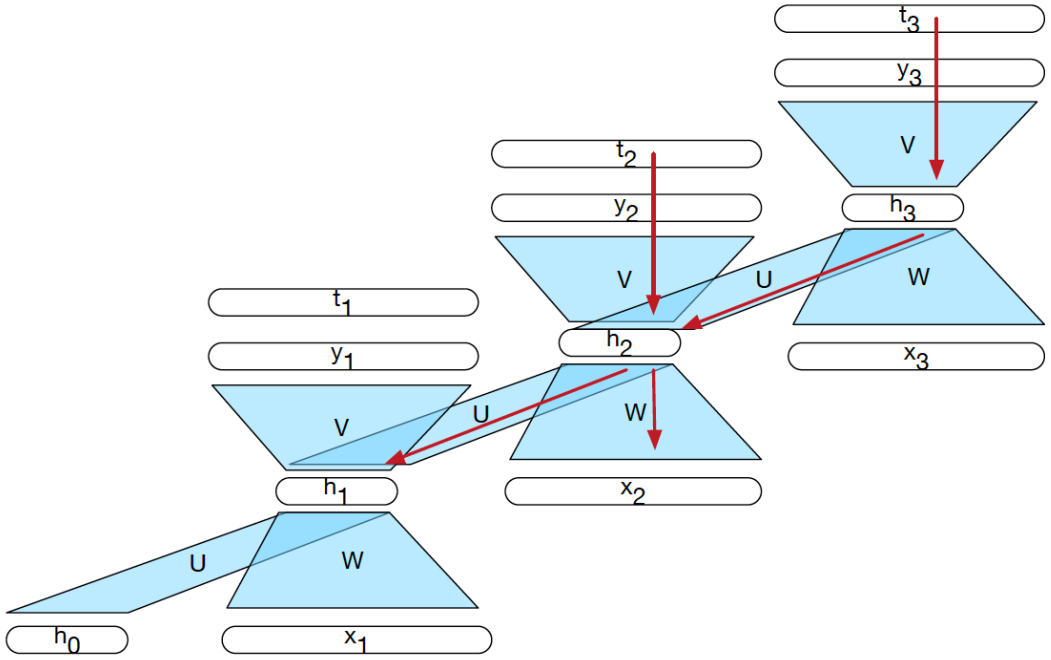
$$y_t = softmax(Vh_t)$$

Source: Jurafsky and Martin

RNN Backpropagation of Errors

Recurrence unrolled

U, W, V are Weights to be learned



Source: Jurafsky and Martin

RNN-based Language Model

- Based on characters or words
- At each step (i.e., character or word)
  - the network retrieves a word embedding for the current word as input
  - combines it with the hidden layer from the previous step to
    - compute a new hidden layer
    - generate an output layer which is passed through a softmax layer to generate a probability distribution over the entire vocabulary.

$$P(w_n|w_1^{n-1}) = y_n$$
$$= softmax(Vh_n)$$

Prob. of a word

$$P(w_1^n) = \prod_{k=1}^{n} P(w_k|w_1^{k-1})$$
$$= \prod_{k=1}^{n} y_k$$

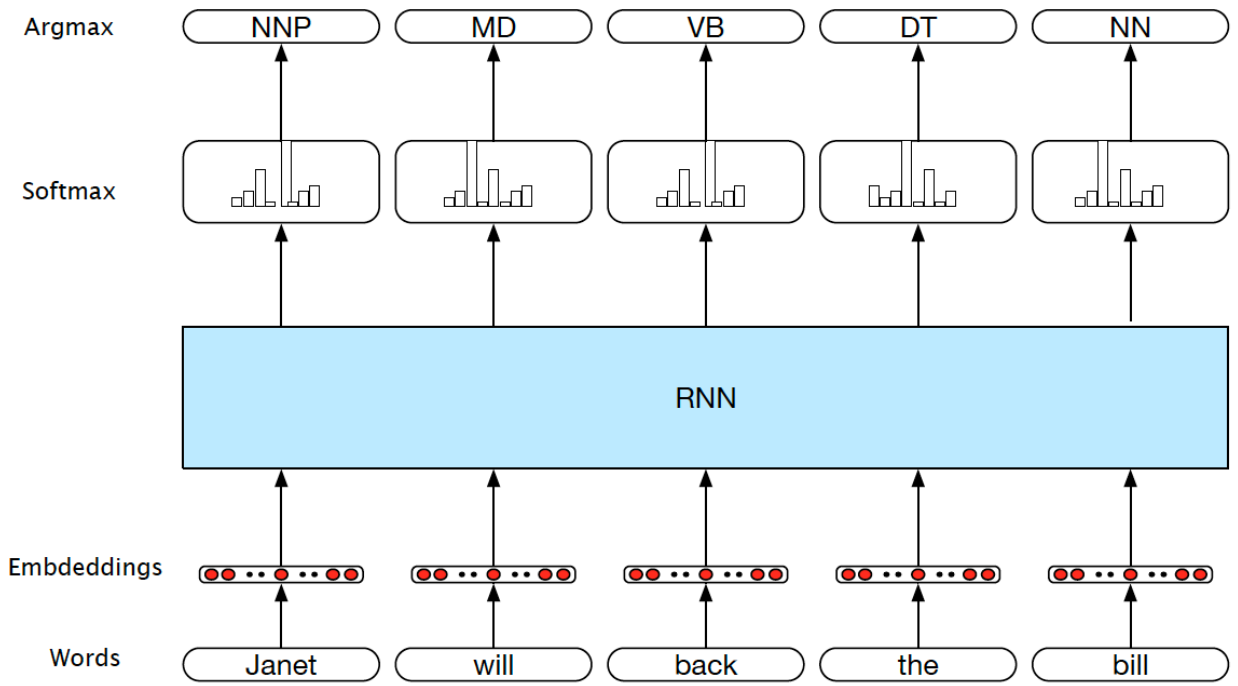
Prob. of a sequence

Source: Jurafsky and Martin

RNN Discussion

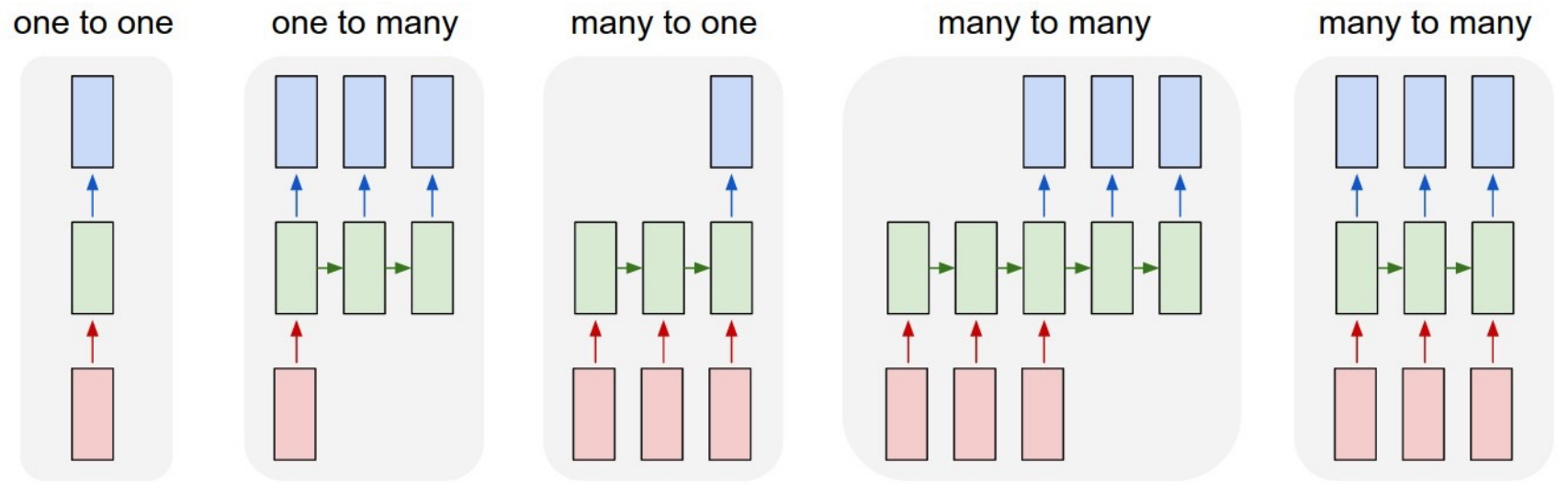
- Language model
  - Not dependent on N-gram boundaries
  - Whole sequence is the context
- Program generation
  - Complexity is Turing-complete
  - In practical terms: On the Practical Computational Power of Finite Precision RNNs for Language Recognition, Gail Weiss, Yoav Goldberg, Eran Yahav, ACL 2018, https://www.aclweb.org/anthology/P18-2117/

RNN Usage Example: Sentence Labeling



Source: Jurafsky and Martin

RNN - Many Applications



Language model

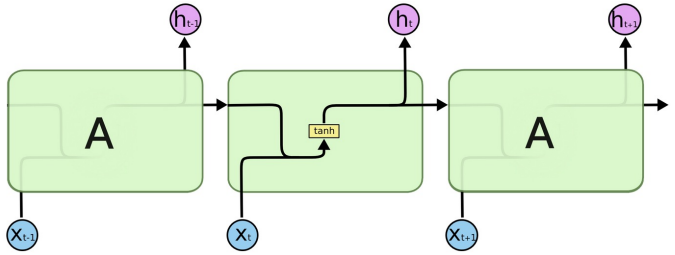
Caption generation

Sentiment detection

Machine translations

Source: http://karpathy.github.io/2015/05/21/rnn-effectiveness/

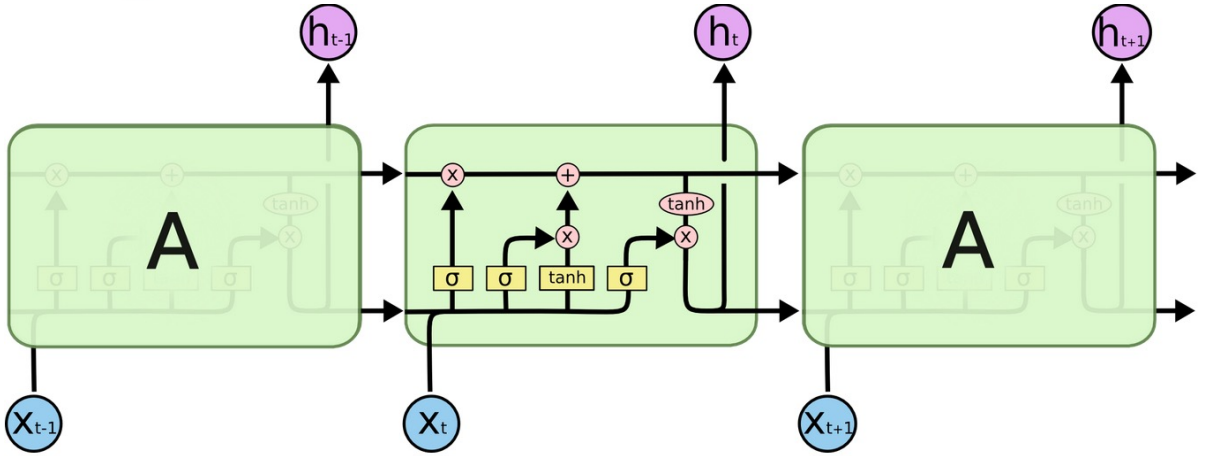
RNN and LSTM - Long Short Term Memory



RNN

LSTM

To learn long term relationships; has 4 NNs



Source and details: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Auto-Encoder

- A model which can produce its inputs

- Usages
  - Input compression
  - Sequence generation
  - Produce recommendation
  - Denoising

Auto-Encoder

- Example 1: numeric array
- Example 2 - exercise: character array
- Code sample:
  - https://github.com/biplav-s/course-nl-f22/blob/main/sample-code/l11-nn-dl/AutoEncoder%20Sequence%20using%20LSTM.ipynb

Contextual Word Embeddings

	Name	Description	URL, References
1.	Elmo (embeddings from language models)	Contextual, deep, character-based	https://allennlp.org/elmo; Deep contextualized word representations, Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, Luke Zettlemoyer. NAACL 2018.
2	Word2Vec	Word-based, prediction focus	*Mikolov, Tomas; et al. (2013). "Efficient Estimation of Word Representations in Vector Space". arXiv:1301.3781 [cs.CL]. Mikolov, Tomas (2013). "Distributed representations of words and phrases and their compositionality". Advances in Neural Information Processing Systems. arXiv:1310.4546.*
3	Glove	Word-based, count	https://nlp.stanford.edu/projects/glove/, Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. [pdf] [bib]
4	Fasttext	Variation of word2vec, works with N-gram, words not in voacbulary	

Commentaries:
https://jalammar.github.io/illustrated-bert/ , https://cai.tools.sap/blog/glove-and-fasttext-two-popular-word-vector-models-in-nlp/

Demonstration: Word2Vec Using Gensim

See sample code on GitHub:
https://github.com/biplav-s/course-nl/blob/master/l7-language/code/Word%20embedding%20with%20Gensim.ipynb

Lecture 11: Concluding Comments

- We reviewed connections between parsing and language model
- We discussed language models and are focusing on pre-requisites needed to understand them
- We discussed contextual word representations as a stepping stone

Concluding Segment

About Next Lecture – Lecture 12

Lecture 12

- LLMs: Glove, Word2Vec, Transformer
- Insights into creating them