Natural Language Processing (NLP)

Pretraining:

DATA/COMPSCI 182 Deep Learning
Lecture 16 10/31/2024

Why some NLP, in this Deep Learning course ?

- Natural text
- Self-supervised !
- Let us take on a real problem !
  - Talk about tokens, sequences, embeddings, decoding, encoding, downstream tasks ….. all in this real setting
- And natural language (processing), is THE problem
  - Large *Language* ;) Models
  - Everything else - image, video, audio, biology … (models) inspired by language models

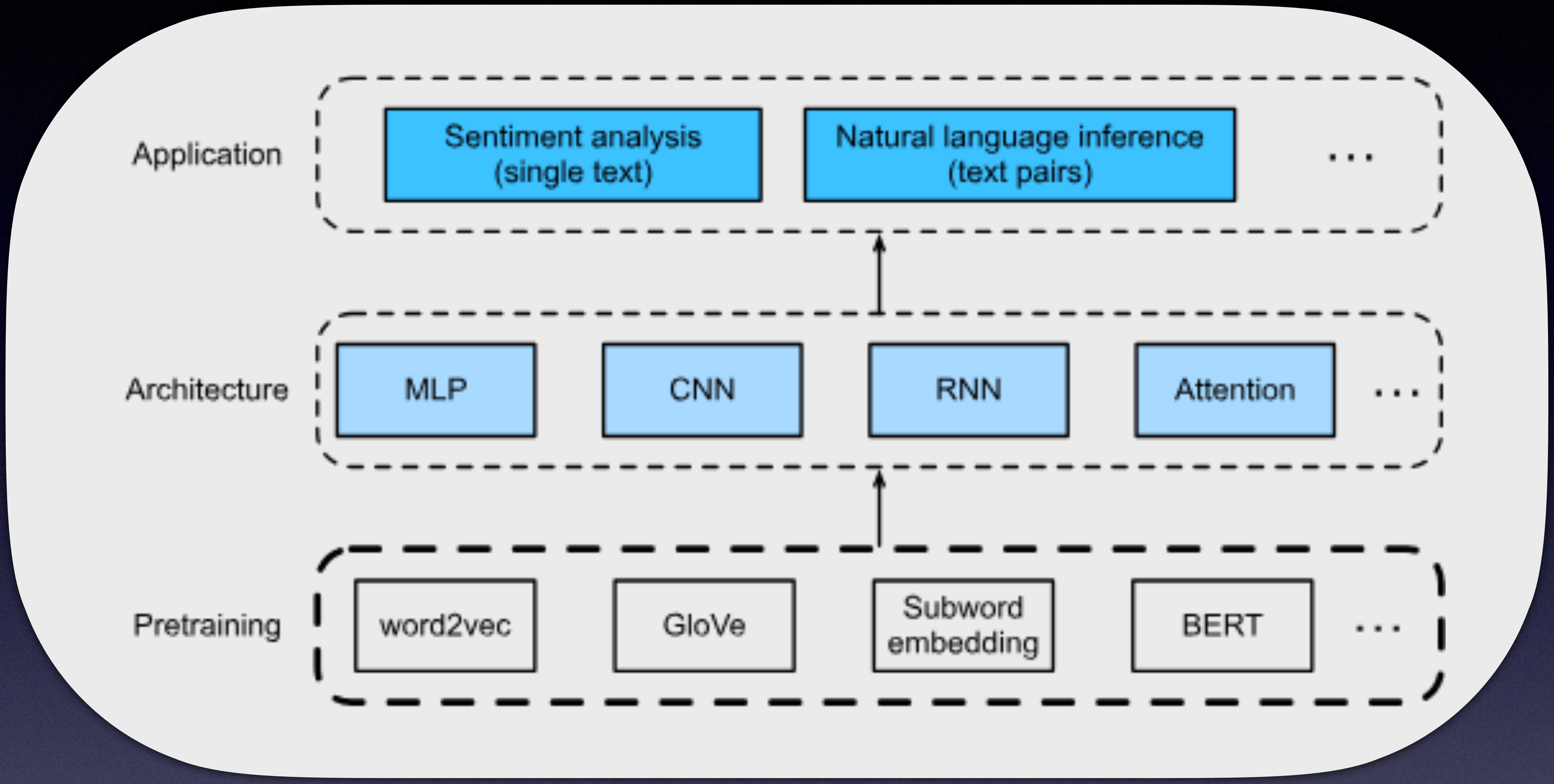
NLP Pretraining

- Embeddings
- Word (token) Embeddings
  - We will learn what they are
- Create them, word2vec
  - By building (in PyTorch) DNNs, and training them over some meaningful data
- Increasing sophistication
  - The “iconic” transformers: ELMo, GPT, and BERT
    - The key aspects of each, and the key differences accross each

I will add ....

- NLP, Language Models are entire courses in themselves !
- The material is drawn from Ch 15 (NLP: Pretraining) from Dive into Deep Learning
    - Select subset

Word Embeddings (word2vec) [d2lai 15.1.1-3]

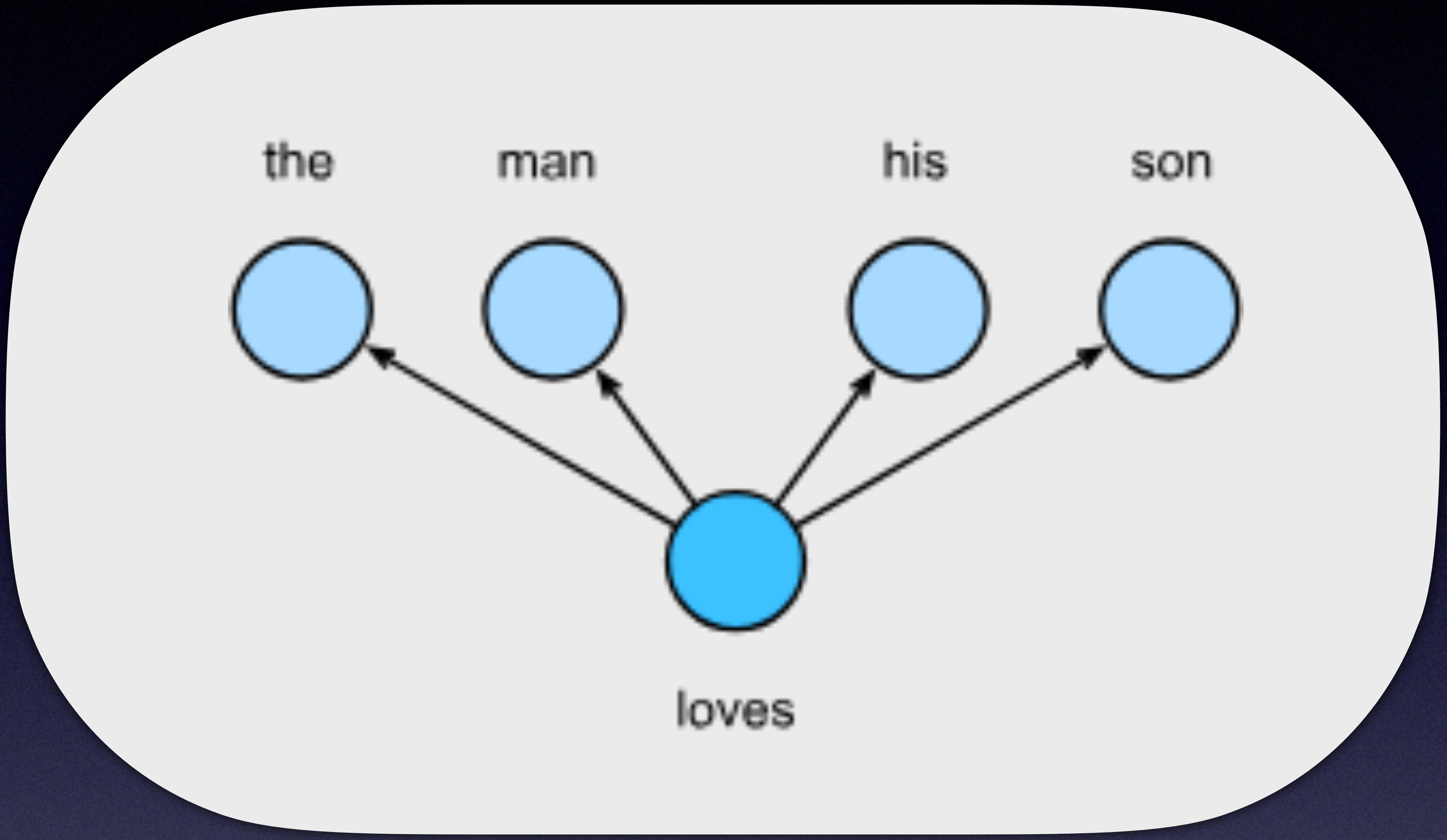


- Pretrained text representations —> DNNs —> different **downstream** NLP applications
- Today we focus on **upstream** representation training

word2vec: Skip-gram Model

$$\frac{\mathbf{x}^\top \mathbf{y}}{\|\mathbf{x}\|\|\mathbf{y}\|} \in [-1, 1].$$

- Encodings: one-hot is not a very good idea



$$P(\text{"the"}, \text{"man"}, \text{"his"}, \text{"son"} \mid \text{"loves"}).$$

$$P(\text{"the"} \mid \text{"loves"}) \cdot P(\text{"man"} \mid \text{"loves"}) \cdot P(\text{"his"} \mid \text{"loves"}) \cdot P(\text{"son"} \mid \text{"loves"}).$$

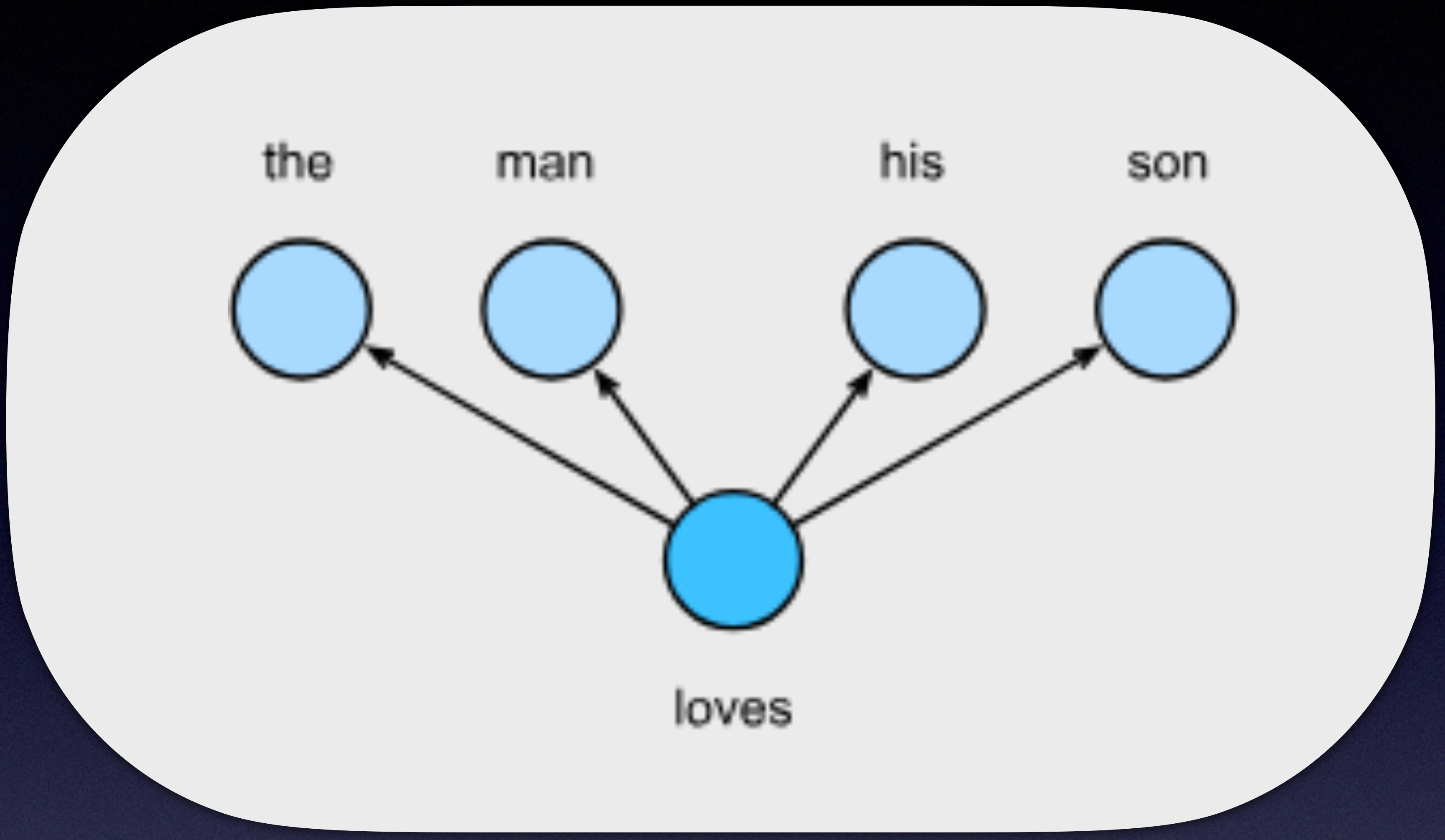
- word2vec: combines skip-gram and CBOW (continuous-bag-of-words)
- Probability of generating context words, given center word

$$P(w_o \mid w_c) = \frac{\exp(\mathbf{u}_o^\top \mathbf{v}_c)}{\sum_{i \in \mathcal{V}} \exp(\mathbf{u}_i^\top \mathbf{v}_c)}$$

- Independence assumption !
- Likelihood

$$\prod_{t=1}^{T} \prod_{-m \le j \le m,\ j \ne 0} P(w^{(t+j)} \mid w^{(t)}),$$

word2vec: Skip-gram Model



$$P(\text{"the"}, \text{"man"}, \text{"his"}, \text{"son"} \mid \text{"loves"}).$$

$$P(\text{"the"} \mid \text{"loves"}) \cdot P(\text{"man"} \mid \text{"loves"}) \cdot P(\text{"his"} \mid \text{"loves"}) \cdot P(\text{"son"} \mid \text{"loves"}).$$

- Any word with index i in the dictionary
  - v_i, and u_i are d-dimensional vectors, center and context respectively
- Conditional probability of generating a context word w_o given the center word w_c
  - Softmax on vector dot products

$$P(w_o \mid w_c) = \frac{\exp(\mathbf{u}_o^\top \mathbf{v}_c)}{\sum_{i \in \mathcal{V}} \exp(\mathbf{u}_i^\top \mathbf{v}_c)}$$

- Likelihood function of skip-gram model
  - Context window: m, Sequence length: T

$$\prod_{t=1}^{T} \prod_{-m \leq j \leq m,\ j \neq 0} P(w^{(t+j)} \mid w^{(t)}),$$

skip-gram model: training

- Skip-gram model parameters: the center and context word vector for *each* word in the vocabulary

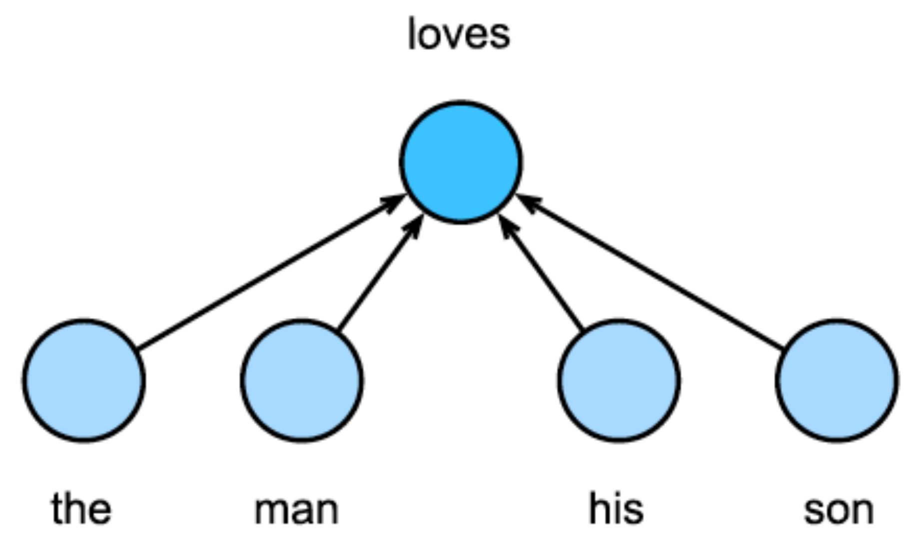
$$\prod_{t=1}^{T} \prod_{-m \leq j \leq m,\ j \neq 0} P(w^{(t+j)} \mid w^{(t)}),$$

- Minimizing this loss function

$$-\sum_{t=1}^{T} \sum_{-m \leq j \leq m,\ j \neq 0} \log P(w^{(t+j)} \mid w^{(t)}).$$

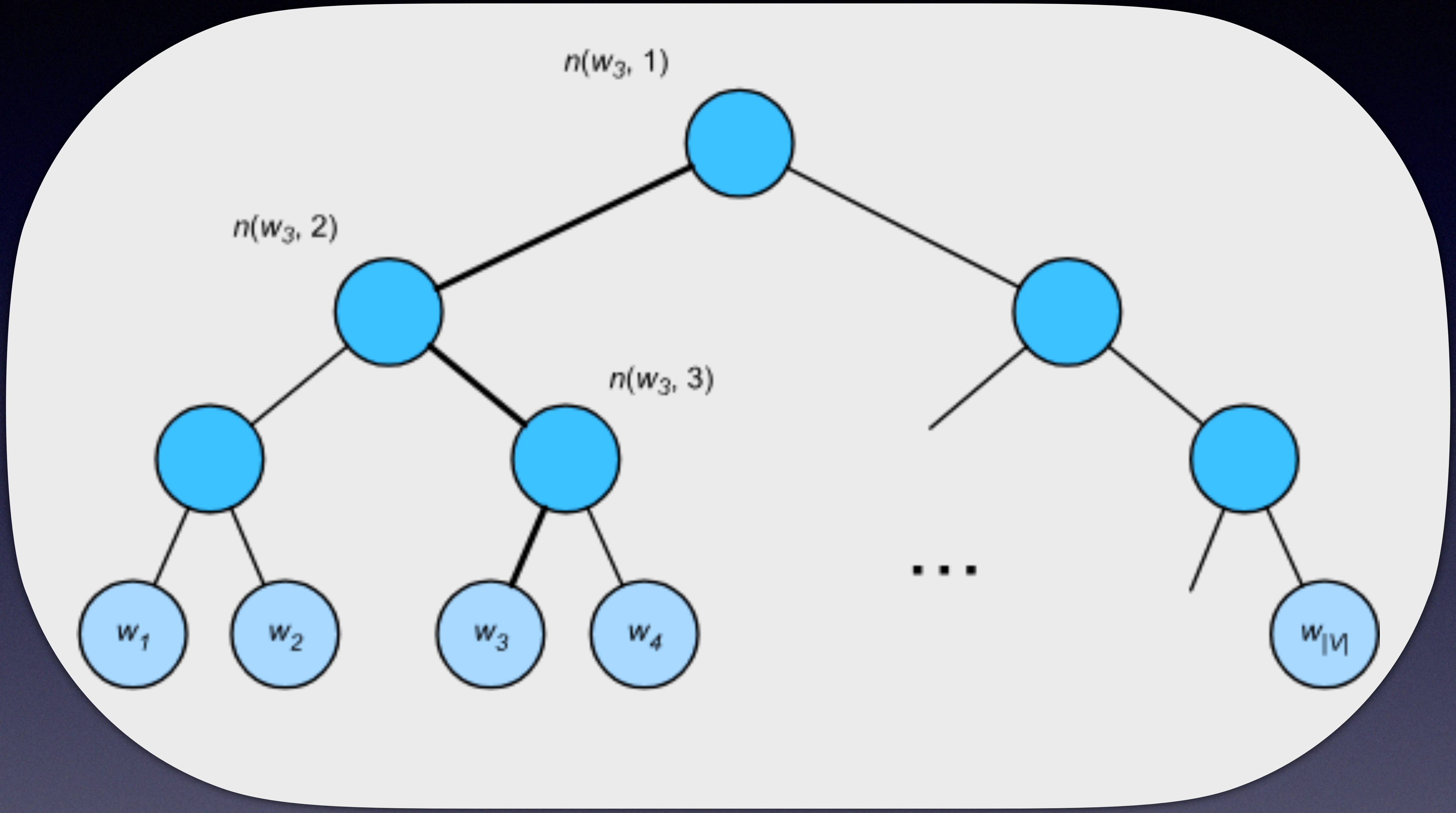
- Stochastic Gradient Descent
- Optimization ?
  - Sample shorter sequences
- Continuous-Bag-Of-Words: CBOW
  - Assumes *center* word is generated, given context words

$$P(\text{"loves"} \mid \text{"the"}, \text{"man"}, \text{"his"}, \text{"son"})$$



Approximate Training [d2lai 15.2.2]

- Negative Sampling
  - Random negative words
- Hierarchical Softmax
  - Binary tree where each leaf is a word
  - Path: product of probabilities to leaf
  - Efficient !
  - Parameters: weight vectors associated with each node in the binary tree
- Skip [d2lai 15.2.2]
  - Set of words for training
  - High frequency words are not very useful
    - Filter out
- Subsampling



Pretraining word2vec [d2lai 15.4]

Notebook: Pretraining word2vec.ipynb

Subword Embedding

- **Byte Pair Encoding**

b a n a n a   b a n d a n a
b an a n a   b an d an a
b ana na   b ana d ana
bana na   bana d ana
bana na   bana dana

Emeddings: Further evolution

from Context-Independent to Context-Sensitive

- **ELMo** (Embeddings from Language Models)
  - Deep Contextualized Word Representations
  - arXiv:1802.05365v2 (2018)
- *f(x) —> f(x,c(x))*
- ELMo: combines all the *intermediate layer representations* from pretrained bidirectional LSTM as the output representation
- BUT ….. significant additional work for each (NLP) application

Deep contextualized word representations

Matthew E. Peters†, **Mark Neumann**†, **Mohit Iyyer**†, **Matt Gardner**†,
{matthewp,markn,mohiti,mattg}@allenai.org

Christopher Clark*, **Kenton Lee***, **Luke Zettlemoyer**†*
{csquared,kentonl,lsz}@cs.washington.edu

†Allen Institute for Artificial Intelligence
*Paul G. Allen School of Computer Science & Engineering, University of Washington

Abstract

We introduce a new type of *deep contextualized* word representation that models both (1) complex characteristics of word use (e.g., syntax and semantics), and (2) how these uses vary across linguistic contexts (i.e., to model polysemy). Our word vectors are learned func- … guage model (LM) objective on a large text corpus. For this reason, we call them ELMo (Embeddings from Language Models) representations. Unlike previous approaches for learning contextualized word vectors (Peters et al., 2017; McCann et al., 2017), ELMo representations are deep, in the sense that they are a function of all of the in-

from Task-Specific to Task-Agnostic

- **GPT**
  - https://openai.com/index/language-unsupervised/ (June 2018)
- Addtional linear layer: for tasks

Improving Language Understanding by Generative Pre-Training

Alec Radford OpenAI alec@openai.com

Karthik Narasimhan OpenAI karthikn@openai.com

Tim Salimans OpenAI tim@openai.com

Ilya Sutskever OpenAI ilyasu@openai.com

Best of both: **BERT**

- BERT:Bidirectional Encoder Representations from Transformers (May 2019)
- arXiv:1810.04805
  - Pretrained transformer encoder
  - Encodes context bidirectionally
  - Minimal architecture change for wide variety of NLP tasks

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

Jacob Devlin Ming-Wei Chang Kenton Lee Kristina Toutanova
Google AI Language
{jacobdevlin,mingweichang,kentonl,kristout}@google.com

Comparison: ELMo, GPT, BERT