Deep Language Models

Nicholas Dronen [1]

[1]HERE, North America

March 13, 2017

What is a language model?

A language model estimates the probability of a word w_i given preceding words $w_{i-(n-1)}$, $w_{i-(n-2)}$, ..., w_{i-1}.

For a bigram model (i.e., when $n = 2$), the probability of a length-k sequence $w_1 \ldots w_k$, denoted w_1^k, is:

$$P(w_1^k) \approx \prod_{j=1}^{k} P(w_j | w_{j-1})$$

Applications of language models

- As a generative model: given some initial state (random or sampled from a data set), generate a statistically likely sequence of words.

Applications of language models

- As a generative model: given some initial state (random or sampled from a data set), generate a statistically likely sequence of words.
- As a discriminative model: given a document, provide a point estimate of the probability of the document. (Generalizes to multiclass classification.)

Fundamental limitation of language models

- The space of linguistic expression is infinite.

Fundamental limitation of language models

- The space of linguistic expression is infinite.
- Data sets are finite.

Fundamental limitation of language models

- The space of linguistic expression is infinite.
- Data sets are finite.
- As n increases, the probability of encountering a sequence (of in-vocabulary words) that did not occur in the training set increases.

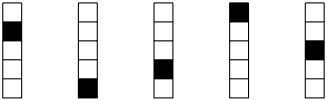
Fundamental limitation of language models

- The space of linguistic expression is infinite.
- Data sets are finite.
- As n increases, the probability of encountering a sequence (of in-vocabulary words) that did not occur in the training set increases.
- How do (non-deep) language models address this?

Fundamental limitation of language models

Denote a word w as a vector v of length $|V|$ with 1 at v_{i_w} and 0 elsewhere, where V is the set of words in the vocabulary and i is a vector of indices.

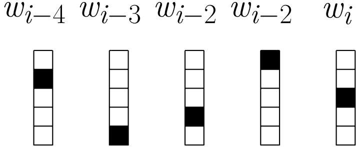
w_{i-4} w_{i-3} w_{i-2} w_{i-2} w_i



What is the cosine similarity of any pair of words?

Fundamental limitation of language models

Denote a word w as a vector v of length $|V|$ with 1 at v_{i_w} and 0 elsewhere, where V is the set of words in the vocabulary and i is a vector of indices.

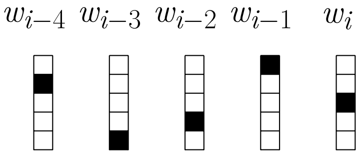


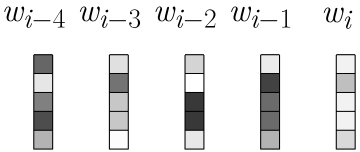
What is the cosine similarity of any pair of words?

What behavior would the distributional hypothesis lead you to expect of word representations?

Representation matters

Deep language models use learned, continuous representations, which behave in concordance with the distributional hypothesis.





Continuous representations and generalization

DT	NN	VBZ	VBG	IN	DT	NN
The	cat	is	walking	in	the	bedroom
A	dog	was	running	in	a	room
The	cat	is	running	in	a	room
A	dog	is	walking	in	a	bedroom
The	dog	was	walking	in	the	room

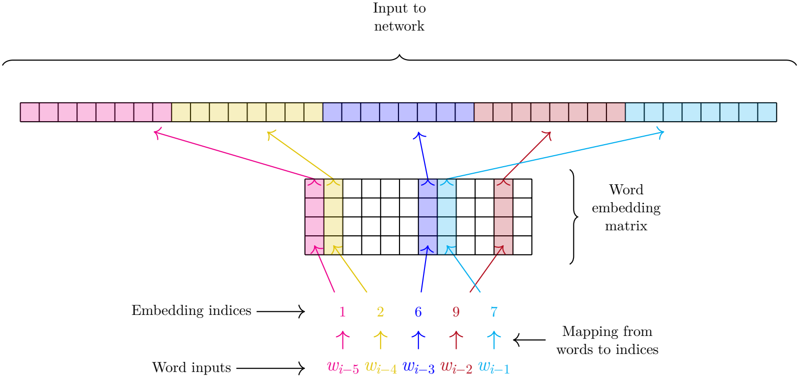
Papers for today

- “A Neural Probabilistic Language Model”, Bengio et al, 2003
- “On the difficulty of training Recurrent Neural Networks”, Pascanu et al, 2013
- “Recurrent neural network based language model”, Mikolov et al, 2010

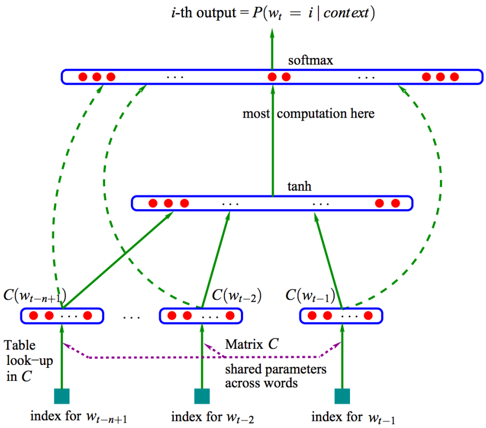
Functional view of models

$$f(w_{i-n}, w_{i-n+1}, \ldots, w_{i-1}) \rightarrow w_i \text{ (Bengio et al, 2003)}$$
$$f(w_{i-1}) \rightarrow w_i \text{ (Mikolov et al, 2010)}$$

Word embeddings



A Neural Probabilistic Language Model



What is the most expensive operation in this network?
Why the skip connections?

The curse of the normalization term

$$x = (C_{w_{t-1}}, C_{w_{t-2}}, \ldots, C_{w_{t-n+1}})$$

$$y = b + Wx + U\, tanh(d + Hx)$$

$$\hat{P}(w_t | w_{t-1}, \ldots, w_{t-n+1}) = \frac{e^{y_{w_t}}}{\sum_i e^{y_i}}$$

The time complexity of a forward pass through the network is O(|V|(nm + h)), where

- V is the set of words in the vocabulary,
- n is the n-gram order,
- m is the dimensions of the word embeddings,
- and h is the number of hidden units.

Attacking the normalization term bottleneck

- Data-parallel approach

Attacking the normalization term bottleneck

- Data-parallel approach
  - One host, shared memory (“SMP”)

Attacking the normalization term bottleneck

- Data-parallel approach
  - One host, shared memory ("SMP")
  - Each processor computes
  - Suffers from lock contention

Attacking the normalization term bottleneck

- Data-parallel approach
  - One host, shared memory ("SMP")
  - Each processor computes
  - Suffers from lock contention
  - Asynchronous version: lock-free parameter updates (cf. Hogwild)

Attacking the normalization term bottleneck

- Data-parallel approach
  - One host, shared memory (“SMP”)
  - Each processor computes
  - Suffers from lock contention
  - Asynchronous version: lock-free parameter updates (cf. Hogwild)
- Parameter-parallel approach

Attacking the normalization term bottleneck

- Data-parallel approach
  - One host, shared memory ("SMP")
  - Each processor computes
  - Suffers from lock contention
  - Asynchronous version: lock-free parameter updates (cf. Hogwild)
- Parameter-parallel approach
  - Multiple hosts, distributed memory

Attacking the normalization term bottleneck

- Data-parallel approach
  - One host, shared memory (“SMP”)
  - Each processor computes
  - Suffers from lock contention
  - Asynchronous version: lock-free parameter updates (cf. Hogwild)
- Parameter-parallel approach
  - Multiple hosts, distributed memory
  - Each host computes all network operations up to, and excluding, the softmax.

Attacking the normalization term bottleneck

- Data-parallel approach
  - One host, shared memory ("SMP")
  - Each processor computes
  - Suffers from lock contention
  - Asynchronous version: lock-free parameter updates (cf. Hogwild)
- Parameter-parallel approach
  - Multiple hosts, distributed memory
  - Each host computes all network operations up to, and excluding, the softmax.
  - The *unnormalized* outputs are shared across hosts

Attacking the normalization term bottleneck

- Data-parallel approach
  - One host, shared memory ("SMP")
  - Each processor computes
  - Suffers from lock contention
  - Asynchronous version: lock-free parameter updates (cf. Hogwild)
- Parameter-parallel approach
  - Multiple hosts, distributed memory
  - Each host computes all network operations up to, and excluding, the softmax.
  - The *unnormalized* outputs are shared across hosts
  - The normalization term is computed centrally (via MPI).

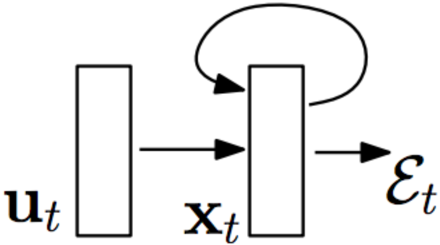
Discussion of results (Brown corpus)

	n	c	h	m	direct	mix	train.	valid.	test.
MLP1	5		50	60	yes	no	182	284	268
MLP2	5		50	60	yes	yes		275	257
MLP3	5		0	60	yes	no	201	327	310
MLP4	5		0	60	yes	yes		286	272
MLP5	5		50	30	yes	no	209	296	279
MLP6	5		50	30	yes	yes		273	259
MLP7	3		50	30	yes	no	210	309	293
MLP8	3		50	30	yes	yes		284	270
MLP9	5		100	30	no	no	175	280	276
MLP10	5		100	30	no	yes		265	**252**
Del. Int.	3						31	352	336
Kneser-Ney back-off	3							334	323
Kneser-Ney back-off	4							332	321
Kneser-Ney back-off	5							332	321
class-based back-off	3	150						348	334
class-based back-off	3	200						354	340
class-based back-off	3	500						326	**312**
class-based back-off	3	1000						335	319
class-based back-off	3	2000						343	326
class-based back-off	4	500						327	312
class-based back-off	5	500						327	312

Discussion of results (AP News corpus)

	n	h	m	direct	mix	train.	valid.	test.
MLP10	6	60	100	yes	yes		104	**109**
Del. Int.	3						126	132
Back-off KN	3						121	127
Back-off KN	4						113	119
Back-off KN	5						112	**117**

Recurrent neural networks



$$x_t = \sigma(\mathbf{W}_{rec}x_{t-1} + \mathbf{W}_{in}u_t + b)$$

Vanishing and exploding gradients

- Deeper networks (e.g. long-range BPTT RNNs) exacerbate this problem.

Vanishing and exploding gradients

- Deeper networks (e.g. long-range BPTT RNNs) exacerbate this problem.
- Sufficient condition for vanishing gradients: largest eigenvalue of $\mathbf{W}_{rec}$ is < 1.

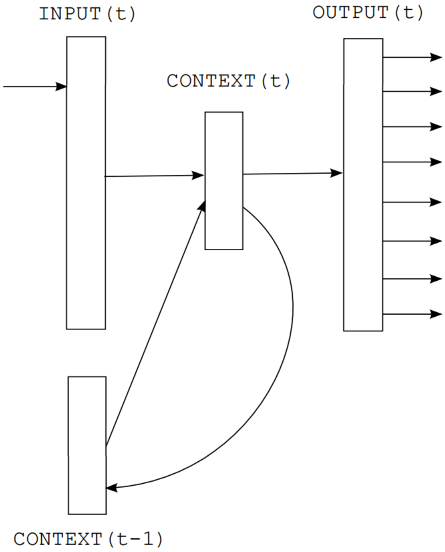
Vanishing and exploding gradients

- Deeper networks (e.g. long-range BPTT RNNs) exacerbate this problem.
- Sufficient condition for vanishing gradients: largest eigenvalue of $\mathbf{W}_{rec}$ is < 1.
- Necessary condition for exploding gradients: largest eigenvalue is > 1.

Vanishing and exploding gradients

- Deeper networks (e.g. long-range BPTT RNNs) exacerbate this problem.
- Sufficient condition for vanishing gradients: largest eigenvalue of $\mathbf{W}_{rec}$ is < 1.
- Necessary condition for exploding gradients: largest eigenvalue is > 1.
- Orthogonal initialization is common solution; "Exact solutions to the nonlinear dynamics of learning in deep linear neural networks", Saxe et al, https://arxiv.org/abs/1312.6120

Recurrent neural network based language model



Discussion of results

Table 1: *Performance of models on WSJ DEV set when increasing size of training data.*

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

Discussion of results

Table 2: *Comparison of various configurations of RNN LMs and combinations with backoff models while using 6.4M words in training data (WSJ DEV).*

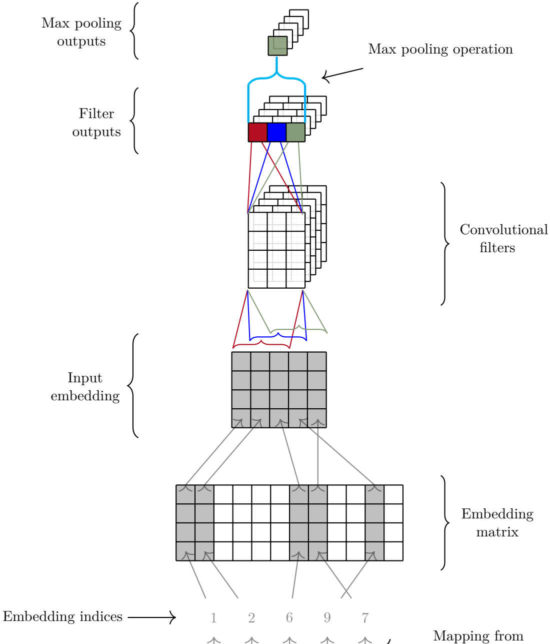
	PPL		WER	
Model	RNN	RNN+KN	RNN	RNN+KN
KN5 - baseline	-	221	-	13.5
RNN 60/20	229	186	13.2	12.6
RNN 90/10	202	173	12.8	12.2
RNN 250/5	173	155	12.3	11.7
RNN 250/2	176	156	12.0	11.9
RNN 400/10	171	152	12.5	12.1
3xRNN static	151	143	11.6	11.3
3xRNN dynamic	128	121	11.3	11.1

Discussion of results

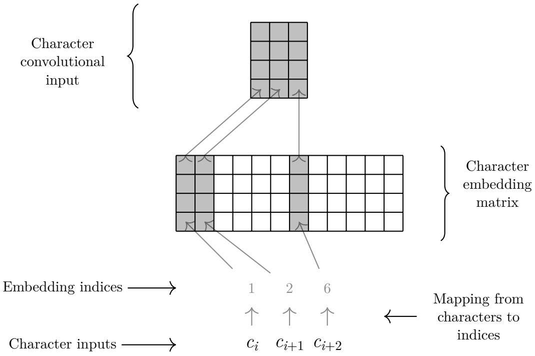
Table 3: *Comparison of WSJ results obtained with various models. Note that RNN models are trained just on 6.4M words.*

Model	DEV WER	EVAL WER
Lattice 1 best	12.9	18.4
Baseline - KN5 (37M)	12.2	17.2
Discriminative LM [8] (37M)	11.5	16.9
Joint LM [9] (70M)	-	16.7
Static 3xRNN + KN5 (37M)	11.0	15.5
Dynamic 3xRNN + KN5 (37M)	10.7	16.3[4]

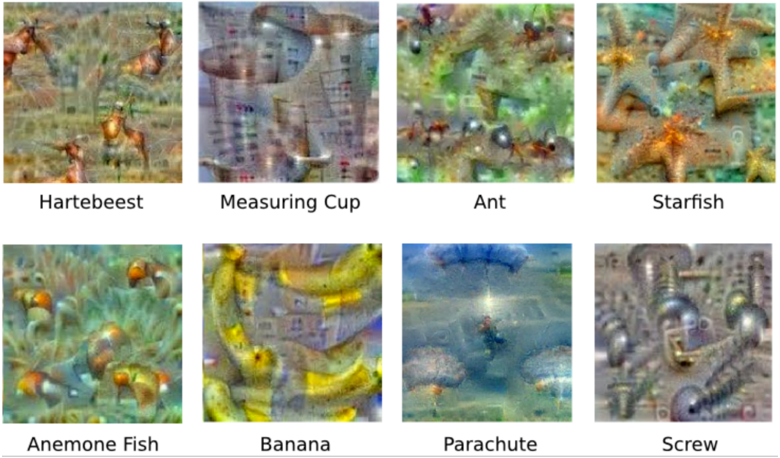
Convolutional Language Models



Character Convolutional Language Models



Generative neural networks are improving quickly



Deep language models are improving quickly

Varying the code of sentiment	Varying the code of tense
this movie was awful and boring .	this was one of the outstanding thrillers of the last decade
this movie was funny and touching .	this is one of the outstanding thrillers of the all time
	this will be one of the great thrillers of the all time
jackson is n't very good with documentary	
jackson is superb as a documentary productions	i thought the movie was too bland and too much
	i guess the movie is too bland and too much
you will regret it	i guess the film will have been too bland
you will enjoy it	

Table 3. Samples by varying one attribute code while fixing the others. Left column: each pair of sentences is generated by varying the sentiment code while fixing the tense code and $\boldsymbol{z}$. Right column: each triple of sentences is generated by varying the tense code while fixing the sentiment code and $\boldsymbol{z}$.

Controllable text generation, Hu et al arXiv:1703.00955

Questions?