IN5550: Neural Methods in Natural Language Processing
Sub-lecture 4.4
Deep learning and language models

Andrey Kutuzov

University of Oslo

14 February 2023

Contents

New way: neural language modeling

- Neural LM model proposed in [Bengio et al., 2003]:

New way: neural language modeling

- Neural LM model proposed in [Bengio et al., 2003]:
- concatenate learned embeddings of the previous k words;

New way: neural language modeling

- Neural LM model proposed in [Bengio et al., 2003]:
- concatenate learned embeddings of the previous k words;
- this concatenation is fed into a feed-forward neural network...

New way: neural language modeling

- Neural LM model proposed in [Bengio et al., 2003]:
- concatenate learned embeddings of the previous k words;
- this concatenation is fed into a feed-forward neural network...
- ...with hidden layers and non-linearities;

New way: neural language modeling

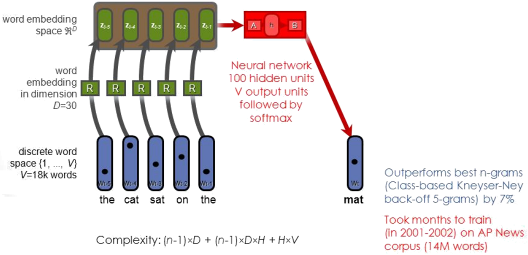
- Neural LM model proposed in [Bengio et al., 2003]:
- concatenate learned embeddings of the previous k words;
- this concatenation is fed into a feed-forward neural network...
- ...with hidden layers and non-linearities;
- cross-entropy loss, the next words as the gold predictions.

New way: neural language modeling

- Neural LM model proposed in [Bengio et al., 2003]:
- concatenate learned embeddings of the previous k words;
- this concatenation is fed into a feed-forward neural network...
- ...with hidden layers and non-linearities;
- cross-entropy loss, the next words as the gold predictions.
- Output probability distribution over possible next words across the vocabulary V (using softmax and the second embedding matrix).

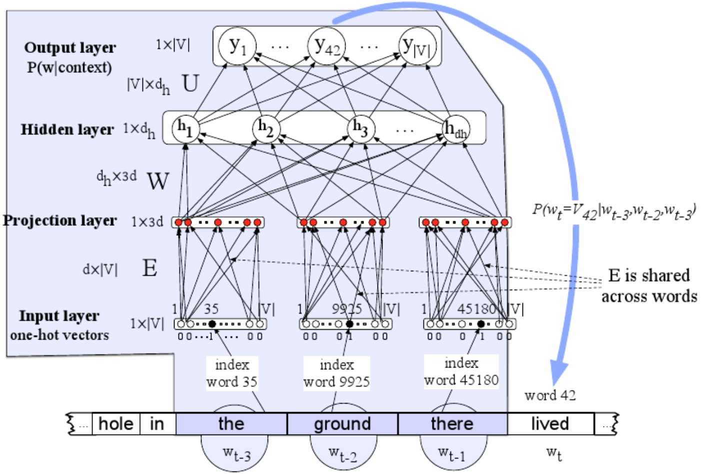
New way: neural language modeling

- Neural LM model proposed in [Bengio et al., 2003]:
- concatenate learned embeddings of the previous k words;
- this concatenation is fed into a feed-forward neural network...
- ...with hidden layers and non-linearities;
- cross-entropy loss, the next words as the gold predictions.
- Output probability distribution over possible next words across the vocabulary V (using softmax and the second embedding matrix).
- Input and output vocabularies can be different.



[Bengio et al, 2001, 2003; Schwenk et al, "Connectionist language modelling for large vocabulary continuous speech recognition", *ICASSP* 2002]

New way: neural language modeling



Feedforward neural LM moving through the text of 'The Hobbit'

(from Jurafsky and Martin, 2019)

New way: neural language modeling

The world is changing fast

- Modern state-of-the-art neural language models are mostly based on recurrent or transformer architectures.

New way: neural language modeling

The world is changing fast

- Modern state-of-the-art neural language models are mostly based on recurrent or transformer architectures.
- This online demo uses transformer-based GPT-2 [Radford et al., 2019] for language generation:
  - https://talktotransformer.com/

New way: neural language modeling

The world is changing fast

- Modern state-of-the-art neural language models are mostly based on recurrent or transformer architectures.
- This online demo uses transformer-based GPT-2 [Radford et al., 2019] for language generation:
  - `https://talktotransformer.com/`
- (and of course you are aware of ChatGPT)
- More on that in the next lectures.

New way: neural language modeling

Benefits

- Outperform non-neural LMs as measured by perplexity.

New way: neural language modeling

Benefits

- Outperform non-neural LMs as measured by perplexity.
- Scale well: higher k leads to linear increase in the parameters number...

New way: neural language modeling

Benefits

- Outperform non-neural LMs as measured by perplexity.
- Scale well: higher k leads to linear increase in the parameters number...
- ...in traditional LMs it was exponential.

New way: neural language modeling

Benefits

- Outperform non-neural LMs as measured by perplexity.
- Scale well: higher k leads to linear increase in the parameters number...
- ...in traditional LMs it was exponential.
- Words in different positions share statistical strength.

New way: neural language modeling

Benefits

- Outperform non-neural LMs as measured by perplexity.
- Scale well: higher k leads to linear increase in the parameters number...
- ...in traditional LMs it was exponential.
- Words in different positions share statistical strength.
- Generalizations to unseen data: similar words get similar representations:
  - '*fox eats*': seen 1000 times; '*dog eats*': seen 1000 times; '*wolf eats*': seen 0 times; $\hat{P}([wolf, eats]) \gg 0$, because '*wolf*' is similar to '*fox*' and '*dog*'.

New way: neural language modeling

Benefits

- Outperform non-neural LMs as measured by perplexity.
- Scale well: higher k leads to linear increase in the parameters number...
- ...in traditional LMs it was exponential.
- Words in different positions share statistical strength.
- Generalizations to unseen data: similar words get similar representations:
  - '*fox eats*': seen 1000 times; '*dog eats*': seen 1000 times; '*wolf eats*': seen 0 times; $\hat{P}([wolf, eats]) \gg 0$, because '*wolf*' is similar to '*fox*' and '*dog*'.
- Can easily add more hidden layers.

New way: neural language modeling

Benefits

- Outperform non-neural LMs as measured by perplexity.
- Scale well: higher k leads to linear increase in the parameters number...
- ...in traditional LMs it was exponential.
- Words in different positions share statistical strength.
- Generalizations to unseen data: similar words get similar representations:
  - '*fox eats*': seen 1000 times; '*dog eats*': seen 1000 times; '*wolf eats*': seen 0 times; $\hat{P}([wolf, eats]) \gg 0$, because '*wolf*' is similar to '*fox*' and '*dog*'.
- Can easily add more hidden layers.

Shortcomings

- Expensive softmax over V in the output layer.

New way: neural language modeling

Benefits

- Outperform non-neural LMs as measured by perplexity.
- Scale well: higher k leads to linear increase in the parameters number...
- ...in traditional LMs it was exponential.
- Words in different positions share statistical strength.
- Generalizations to unseen data: similar words get similar representations:
  - '*fox eats*': seen 1000 times; '*dog eats*': seen 1000 times; '*wolf eats*': seen 0 times; $\hat{P}([wolf, eats]) \gg 0$, because '*wolf*' is similar to '*fox*' and '*dog*'.
- Can easily add more hidden layers.

Shortcomings

- Expensive softmax over V in the output layer.
- Increasing the output $|V|$ can significantly slow down the network.

New way: neural language modeling

Benefits

- Outperform non-neural LMs as measured by perplexity.
- Scale well: higher k leads to linear increase in the parameters number...
- ...in traditional LMs it was exponential.
- Words in different positions share statistical strength.
- Generalizations to unseen data: similar words get similar representations:
  - '*fox eats*': seen 1000 times; '*dog eats*': seen 1000 times; '*wolf eats*': seen 0 times; $\hat{P}([wolf, eats]) \gg 0$, because '*wolf*' is similar to '*fox*' and '*dog*'.
- Can easily add more hidden layers.

Shortcomings

- Expensive softmax over V in the output layer.
- Increasing the output $|V|$ can significantly slow down the network.
- There are ways to deal with this (more next week).

Contents

What about word embeddings? Let's recall:

- '*Generalizations: similar words get similar representations in the embedding layer*'

Neural LM and word embeddings

What about word embeddings? Let's recall:

- '*Generalizations: similar words get similar representations in the embedding layer*'
- Yes: the neural LM learns representations for words as a byproduct of the training process.

What about word embeddings? Let's recall:

- '*Generalizations: similar words get similar representations in the embedding layer*'
- Yes: the neural LM learns representations for words as a byproduct of the training process.
- These representations are similar for semantically similar words.

What about word embeddings? Let's recall:

- '*Generalizations: similar words get similar representations in the embedding layer*'
- Yes: the neural LM learns representations for words as a byproduct of the training process.
- These representations are similar for semantically similar words.
- But this is exactly what we need: good word embeddings from an auxiliary unsupervised (or semi-supervised) task.

Neural LM and word embeddings

What about word embeddings? Let's recall:

- '*Generalizations: similar words get similar representations in the embedding layer*'
- Yes: the neural LM learns representations for words as a byproduct of the training process.
- These representations are similar for semantically similar words.
- But this is exactly what we need: good word embeddings from an auxiliary unsupervised (or semi-supervised) task.
- Language models are trained on raw texts, no manual annotation needed.

What about word embeddings? Let's recall:

- '*Generalizations: similar words get similar representations in the embedding layer*'
- Yes: the neural LM learns representations for words as a byproduct of the training process.
- These representations are similar for semantically similar words.
- But this is exactly what we need: good word embeddings from an auxiliary unsupervised (or semi-supervised) task.
- Language models are trained on raw texts, no manual annotation needed.
- And we have lots of raw texts.
- Language modeling is a tool to provide good embeddings for other tasks

Neural LM and word embeddings

What about word embeddings? Let's recall:

- '*Generalizations: similar words get similar representations in the embedding layer*'
- Yes: the neural LM learns representations for words as a byproduct of the training process.
- These representations are similar for semantically similar words.
- But this is exactly what we need: good word embeddings from an auxiliary unsupervised (or semi-supervised) task.
- Language models are trained on raw texts, no manual annotation needed.
- And we have lots of raw texts.
- Language modeling is a tool to provide good embeddings for other tasks

How come that we can get good word embeddings without any manual supervision? Will see next week!

Contents

Next group session: February 15

- Working with word embeddings

Contents

Next week lecture trailer

- Obligatory 1 results

Next week lecture trailer

- Obligatory 1 results

Distributional hypothesis and distributed word embeddings

- Distributional hypothesis: '*Meaning is context*'
- Word2vec revolution.
- Training word embeddings on large text corpora.

References I

Bengio, Y., Ducharme, R., and Vincent, P. (2003).
A neural probabilistic language model.
Journal of Machine Learning Research, 3:1137–1155.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019).
Language models are unsupervised multitask learners.
Technical report, OpenAI Blog.