Combining Neural Language Models with n-grams

Sargur N. Srihari
srihari@cedar.buffalo.edu

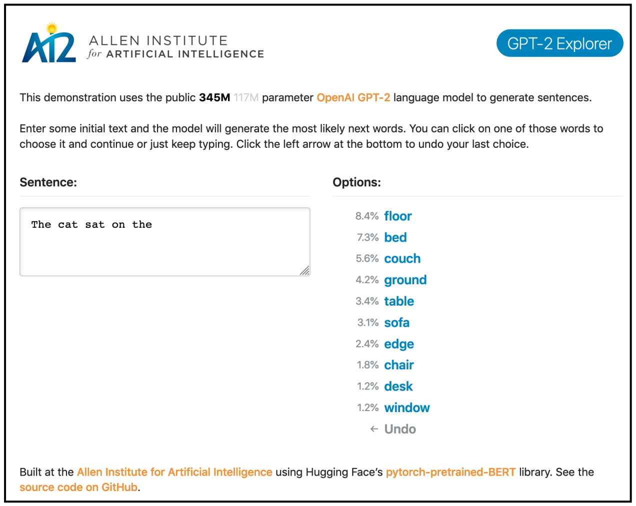
This is part of lecture slides on Deep Learning:
http://www.cedar.buffalo.edu/~srihari/CSE676

Topics

1. N-gram Models
2. Neural Language Models
3. High-Dimensional Outputs
4. Combining Neural Language Models with n-grams
5. Neural Machine Translation
6. Other Applications

Goal of Language Modeling

- Goal: Estimate probabilities of word sequences
  – Equivalent to estimating conditional probability of a word given preceding words, by chain rule
  – It is key to NLP, with applications to
    - Type-ahead systems
    - Machine Translation
    - Automatic Speech Recognition



- Basic Model
  – Goal is to estimate the probability $P(w_t|\mathbf{c})$ of a next word w_t given its context sequence $\mathbf{c}=(w_{t-1},\ldots,w_1)$
    - the context being empty if $t=1$

N-gram models

- $P(w_t|\mathbf{c})$: Probability of word w_t given $\mathbf{c} = (w_{t-1},\ldots,w_1)$
  - Relies on Markov assumption:
    - Next word depends only on $N{-}1$ previous words:

$$P(w_t|\mathbf{c}) = P(w_t|\mathbf{c}_{N-1}) \text{ where } \mathbf{c}_{N-1} = (w_{t-1},\ldots,w_{t-N+1})$$

Bigram Counts

	i	want	to	eat	chinese	food	lunch	spend
i	5	827	0	9	0	0	0	2
want	2	0	608	1	6	6	5	1
to	2	0	4	686	2	0	6	211
eat	0	0	2	0	16	2	42	0
chinese	1	0	0	0	0	82	1	0
food	15	0	15	0	1	4	0	0
lunch	2	0	0	0	0	1	0	0
spend	1	0	1	0	0	0	0	0

Bigram Probabilities

	i	want	to	eat	chinese	food	lunch	spend
i	0.002	0.33	0	0.0036	0	0	0	0.00079
want	0.0022	0	0.66	0.0011	0.0065	0.0065	0.0054	0.0011
to	0.00083	0	0.0017	0.28	0.00083	0	0.0025	0.087
eat	0	0	0.0027	0	0.021	0.0027	0.056	0
chinese	0.0063	0	0	0	0	0.52	0.0063	0
food	0.014	0	0.014	0	0.00092	0.0037	0	0
lunch	0.0059	0	0	0	0	0.0029	0	0
spend	0.0036	0	0.0036	0	0	0	0	0

  - Maximum likelihood estimate:

$$P(w_t|\mathbf{c}_{N-1}) = \mathrm{C}(w_t, \mathbf{c}_{N-1}) / \mathrm{C}(\mathbf{c}_{N-1})$$

    - where $\mathrm{C}(\bullet)$ is no. of occurrences of sequence • in the training corpus

N-gram with Backoff

- For high order models, e.g, $N = 5$, only a small fraction of N-grams appear in training corpus
  - a problem of data sparsity
    - with 0 probability for almost all sentences
- This is overcome by n-gram (NG) with back-off:

$$\mathbf{P}_N^{NG}(w_t) = \begin{cases} p_{w_t,\mathbf{c}_{N-1}} & \text{if not zero} \\ \mathbf{P}_{N-1}^{NG}(w_t)\alpha_{\mathbf{c}_{N-1}} & \text{otherwise} \end{cases}$$

  - where α are back-off coefficients and p are discounted probabilities

Source: https://arxiv.org/pdf/1804.07705.pdf

Neural Language Model

- Another way to reduce sparsity: encode context $\mathbf{c}$ as a fixed length dense vector h_t
  - Each word w is mapped to embedding vector $v(w)$
  - The sequence of vectors $v(w_1),\ldots,v(w_t)$ is then fed to a neural network f to produce h_t
  - A linear classifier a is then applied to h_t to estimate the probability distribution over the next word:

$$\mathbf{P}^{NN}(\bullet|\mathbf{c}) = a\left(f\left(v(w_1),\ldots,v(w_{t-1})\right)\right)$$

  - Different networks can be used as encoder:
    - Fully connected, convolutional, recurrent. LSTM

Compare N-grams to Neural model

- Neural models need less memory and generalize better
  - But increased computation at training and test time
  - Increased computation with increased no of parameters
- N-gram models achieve high model capacity
- See next

Capacity of N-gram models

- N-gram models achieve high capacity
  – By storing frequencies of very many tuples

	i	want	to	eat	chinese	food	lunch	spend
i	5	827	0	9	0	0	0	2
want	2	0	608	1	6	6	5	1
to	2	0	4	686	2	0	6	211
eat	0	0	2	0	16	2	42	0
chinese	1	0	0	0	0	82	1	0
food	15	0	15	0	1	4	0	0
lunch	2	0	0	0	0	1	0	0
spend	1	0	1	0	0	0	0	0

  – Require little computation to process an example
    - By looking up a few tuples that match the current context
  – With hash tables or trees to access counts, computation is independent of capacity

Capacity of neural network

- Doubling no. of parameters of a neural network doubles computation time
- Layers based on matrix multiplication use amount of computation proportional to no. of parameters

Ensemble of neural net and n-gram

- Can add capacity by combining two models:
  1. Neural language model
  2. N-gram language model
- It can reduce test error if ensemble members make independent mistakes
- Best performing language model is often an ensemble of a neural language model with n-grams

Pairing neural net with maximum entropy model

- Viewed as training a neural net with extra inputs connected to output, not to any other part
  - Extra inputs are indicators for presence of particular N-grams in the input context, so these variables are very high-dimensional and very sparse
- Increase in model capacity is huge
  - New portion of the architecture contains upto $|sV|^N$ parameters
  - But added computation needed to process an input is minimal because the extra inputs are sparse