Sequence Model

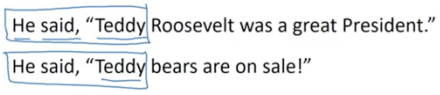
Sequence models such RNN, GRU, LSTM and few other algorithms will be discussed.

RNN (Recurrent neural network) and BRNN (bidirectional RNN)

First why normal NN doesn't work for word sequence:

- Input and output have different length in examples
- Doesn't share feature learned across different position

BRNN, for detecting the name in the sentence, normal RNN consider the info from left to right, and does not consider info after the word. In BRNN, it considers words before and after to identify the names in the sentences. Use case:



Here we do backprop through the time. RNN has vanishing gradient problem that can be taken care of *using gradient clipping*.

GRU (Gated recurrent unit)

It's a modified RNN that captures long range connections and help a lot with the vanishing gradient problem. As we read the sentence from left to right, it has a *memory cell*. For example, in the senses (the cat which was on patio was full) remembers whether in the sentence there is a cat or cats.(c(t)=1, if cat is singular and 0 if it's plural, and update the value when new concept shows up in sentence)

LSTM (long term, short term memory)

LSTM allows you to learn very long range of connections better in compare to GRU.

LSTM vs GRU:

- GRUs train faster and perform better than LSTMs on less training data if you are doing language modeling (not sure about other tasks).
- GRUs are simpler and thus easier to modify, for example adding new gates in case of additional input to the network. It's just less code in general.
- LSTMs should in theory remember longer sequences than GRUs and outperform them in tasks requiring modeling long-distance relations.
- The GRU unit controls the flow of information like the LSTM unit, but without having to use a memory unit. It just exposes the full hidden content without any control.
- GRU is better than LSTM as it is easy to modify and doesn't need memory units, therefore, faster to train than LSTM and give as per performance.
- LSTM is more powerful, flexible and generalized version of GRU
- The key difference between a GRU and an LSTM is that a GRU has two gates (reset and update gates) whereas an LSTM has three gates (namely input, output and forget gates).

Bidirectional RNN (BRNN)

It takes information from before and after the sequence. For this method, you need a full sequence before you can make a prediction.

Word Embedding

One-hot-key vector is a vector of all zero except one. Instead of using one-hot-key, it's better to use the word embedding feature vector. In table below, for each work we assign a probability for each type (dense embedding). So, it can distinguish that orange and apple are fruit, while man and woman are in another group.

Featurized representation: word embedding

	Man (5391)	Woman (9853)	King (4914)	Queen (7157)	Apple (456)	Orange (6257)
Gender	-1	1	-0.95	0.97	0.00	0.01
Royal	0.01	0.02	0.93	0.95	-0.01	0.00
Age	0.03	0.02	0.7	0.69	0.03	-0.02
Food	0.04	0.01	0.02	0.01	0.95	0.97
size						
cost						
alive						
verb						

300

e_{5391} e_{9853}

I want a glass of orange juice.

I want a glass of apple_____.

Andrew Ng

Transfer learning and word embeddings

1. Learn word embeddings from large text corpus. (1-100B words)

   (Or download pre-trained embedding online.)

2. Transfer embedding to new task with smaller training set. (say, 100k words)

Analogies

	Man (5391)	Woman (9853)	King (4914)	Queen (7157)	Apple (456)	Orange (6257)
Gender	−1	1	-0.95	0.97	0.00	0.01
Royal	0.01	0.02	0.93	0.95	-0.01	0.00
Age	0.03	0.02	0.70	0.69	0.03	-0.02
Food	0.09	0.01	0.02	0.01	0.95	0.97

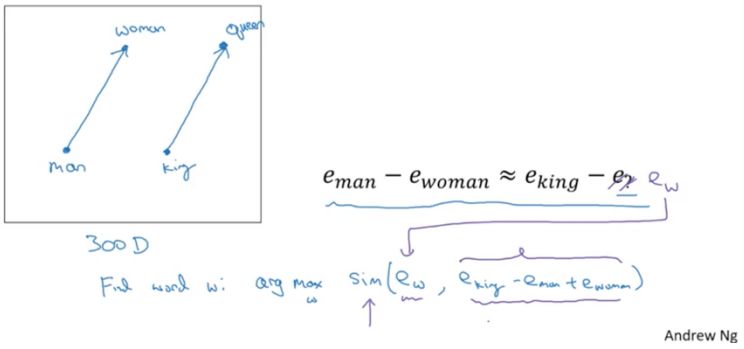
e_{5391} e_{man} e_{woman}

Man → Woman as King → ?

$e_{man} - e_{woman} \approx \begin{bmatrix} -2 \\ 0 \\ 0 \\ 0 \end{bmatrix}$

$e_{King} - e_{queen} \approx \begin{bmatrix} -2 \\ 0 \\ 0 \\ 0 \end{bmatrix}$

Analogies using word vectors



The most common similarity function is Cosine similarity.
Using this similarity and word embedding, the algorithm can find similarity between boy and man vs girl and woman.

$$\text{sim}(u, v) = \frac{u^T v}{\|u\|_2 \|v\|}$$

Word2Vec: Skip-gram model
Problems: computational speed, because we need to carry out the huge sum over.
Solution: use hierarchal SoftMax.

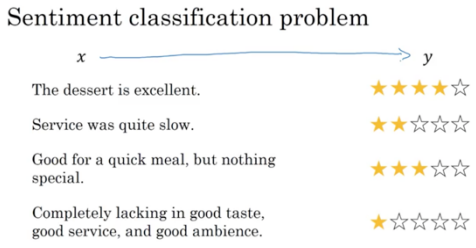
Negative sampling: link
This takes care of computational cost of skip gram model. We sample the target and context, target is 1 if it's positive, else (if it's negative) it's 0.

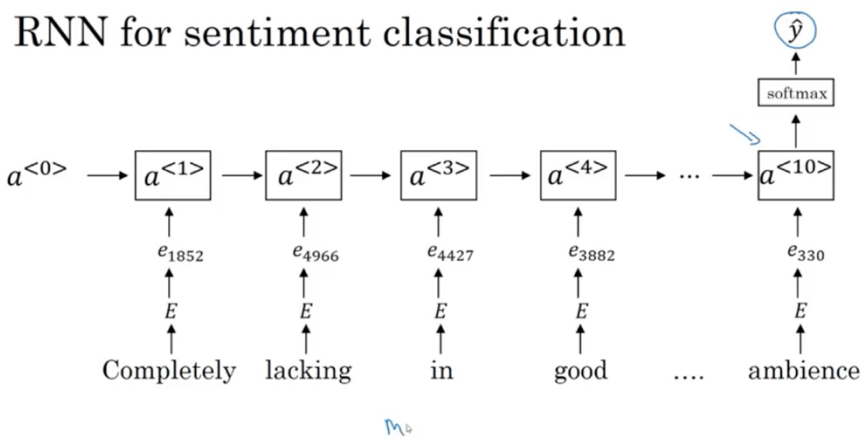
I want a glass of orange juice to go along with my cereal.

Context	word	target?
orange	juice	1
orange	king	0
orange	book	0

GloVe word vector: link
It is a simple algorithm. It is stand for global vectors for word representation.

Applications using word embedding
Sentiment Classification
As an example, shown here, we would like to find the problems that causes the bad starts in review.
You can use RNN for the classification, instead of summing over word embeddings.
In RNN, find E (one hot vector) and multiply by embedding matrix.
(many to one RNN)





Debiasing word embeddings

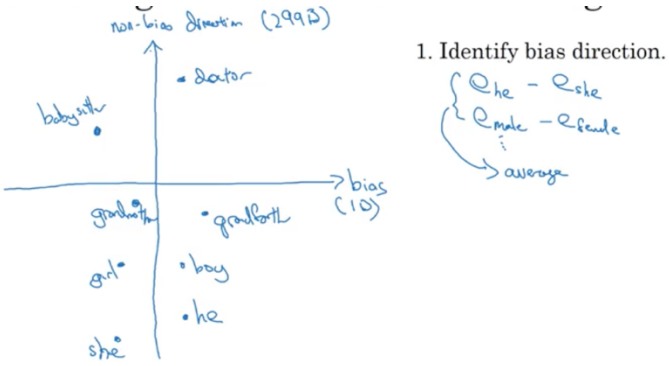
ML can learn analogies such as man as woman is similar to king as queen. But we don't like the stereotype relations such as:

Man: computer programmer, woman: homemaker

Father: doctor, mother: nurse

Word embedding can reflect gender, ethnicity, age, sexual orientation, and other biases of the text used to train the model. We can take following actions to reduce the bias:

1- Identify the bias direction



2- Neutralize: get rid of the bias

3- Equalize the pairs

Various sequence to sequence architecture

AlexNet: Application image captioning

Conditional language model: can be used for translation of one language to another

Greedy search: not a good algorithm to use for translation (pick the best first word)

Approximate search algorithm: it finds the best sentence, which is better than greedy search.

Beam search

We use fragment network. It considers multiple better output (beam width B=3, it considers 3 best ones). Decodes the network and keep in memory just few of them with highest probabilities.

Bleu Score (bilingual evaluation understudy)

There might be several good answers to the problem. Bleu score helps in this regard. As an example:

French: Le chat est sur le tapis.

Reference 1: The cat is on the mat.

Reference 2: There is a cat on the mat.

It will compare the outputs and see if it appears in references and pick that one.

Attention model: only allows to pay attention to part of model.