Deep Models for Text and Sequence

Irdi Balla

22/10/2017

1 Problems

- Words with high frequency give not much information about the document while those with very low frequency could give a lot more information. Its hard to create a dataset with many infrequent words.
- We use different words to denote similar things. Its better for the model to be able to share parameters for them.

2 Unsupervised Learning

- Training without labels
- Similar words occur in similar context

3 Embeddings

Words are mapped to vectors called embeddings. Embeddings are close when to each-other for similar words and far apart otherwise.

4 Word2Vec

- a way to learn embeddings
- Map every word of a sentence to an embedding. Then we use that embedding to predict the context of the word. In this setting the context are the words nearby. Pick a random word near and that is your target. Then train the model as if it were a supervised learning problem.

5 Recurrent Networks

- share weights across time

- when making a decision you want to take the past into account. The way to do that is to use a model to remember the past.
- To compute the parameter updates od a RNN we need to back-prop in time.
- All the derivatives are applied to the same parameters which means a lot of correlated updates at the same time. This is bad for SGD as it causes exploding and vanishing gradients.
- To fix exploding gradients we simply shrink the step when the norm grows too big.

$$\delta\omega < -\delta\omega \frac{\delta_{max}}{max(|\delta\omega|, \delta_{max})} \tag{1}$$

- To fix vanishing gradient we use LSTM
- L2-regularization works
- Dropout works if you do it on the input or output