Convolutional Networks for NLP

Shubham Gupta

February 27, 2020

1 RNN to CNN

- RNN cannot capture phrases without context
- Capture too much of last words in final vector
- **CNN Idea**: Compute vectors for every possble word subsequence of certain length.
- Group these representation

2 What is a convolution?

- Used in vision applications
- 2D Discrete convolution
  - Patch of numbers i.e matrix of numbers.
  - Slide this patch over the image
  - Multiply numbers in patch with numbers in image
  - Sum up the result

2.1 1D convolution for text

- Dense word vectors
- Apply filter(or kernel) of size 3 i.e of taking 3 words at a time
- Dimensions of the word vectors are referred to as **channels**. I think this stems from the fact that CNN are generally used for images which can are RGB encoded(3 channels)
- Apply dot product. Do this recursively. Dot product will give a single number
- Add zero padding at both ends so that length after convolution is the same length as the input.
- Because we get a single number after applying convolution, we have reduced the channel count from 3 to 1. THIS IS BAD because we lose information

- *Solution*: Use multiple filters. Use 3 filters instead of 1 to get 3 channel output. Called *wide convolution*
- Summarize the output of convolution. For 1D, This is called *max pooling*. Just select the max value per channel
- Could do something like average pooling as well. But max pooling works better in practice.
- Moving down one element at a time is called **stride**. One element would be stride of 1. Most common.
- **Local Pooling**: Max pool set of rows. Set is selected according to the stride. This will also reduce computation
- **K-max polling**: Keep the top K max values IN THE ORDER THEY OCCUR.
- **Dilated Convolution**: Skip some rows in between. dilation = 2 will take alternate rows

3 Single Layer CNN for sentence classification

- Represent all words as a single vector
- Add padding only to the right
- Max pooling
- Multi channel input idea: Use 2 sets of pre-train word vectors. Backprop into one set, keep other set "frozen". Both added before max pooling.
- Final softmax layer for classification

3.1 Regularization

- Use dropout
- At test, no dropout. Scale weight matrix to get vectors of same scale

4 Model Comparison

- **Bag of vectors**: Baseline for simple classification problems. Add ReLU layers for moar performance increase
- **Window Model**: Good for single word classification. Low context. Eg: POS, NER
- **CNN**: Good for classification. Zero padding reqd. Parallelize well on GPU
- **RNN**: Not best for classification. Slower than cNN. Good for sequence tagging. Great for LM. AMAZING with attention

5 Gated units used vertically

- Apply gates vertically for CNN
- Residual block. $F(x) + x$. Also called ResNet
- Need to use padding for convnet
- **Highway Block**: Similar to resnet. It has forget and input gate. $F(x)T(x) + x.C(x)$

6 Batch Normalization

- Used in CNN
- Transform conv output of a batch by scaling activation to gaussian i.e 0 mean and unit variance. Similar to Z-score in statistics

7 1x1 convolutions

- Kernel size = 1
- Fully connected linear layer across channels
- Lesser params than fully connected layers

8 Translation

- CNN for encoding and RNN for decoding
- Pretty good performance
- Paper: Kalchnrenner and Blunsom. "Recurrent continous translation models"

9 Convolution over character

- Used convolution to generate word embeddings
- Fixed window of embeddings for POS

10 Quasi RNN

- RNN are slow.
- Combine RNN and CNN to get best of both
- Stick relation between t and $t-1$ through conv op.
- Gives pseudo recurrence
- Deeper NN will be better
- Better and faster than LSTM