ECG 1786 Lecture #3

Last Day: How Word Embeddings are Trained

Work-in-flight: Assignment 2 Classification

Today: Classification using word embeddings

- Comments on Assignment 1?

So, now we know/see how embeddings represent meaning
- two words with the same meaning would have similar embeddings
- one word with multiple meanings: still a problem → later

A key general application in machine learning is classification. You have previously seen picture classification.
- now we’d like to classify sentences, paragraphs, documents → into what?

1. Sentiment - positive or negative -> large -1 → 1

2. Named Entity Recognition - identify something that can describe in many ways
- e.g., specific reason for guilty smiling
  makes me calm → relates me

3. Style of telling "charge talk" in behavior change

4. Politics - left vs. right

5. Depression / Anxiety in speech

6. Suicidality: Facebook detects (Google to see)

7. Lawyers look for specific facts/rep

- now that we have text → embeddings, we can use a NN to work with it/classify it.
In assignment 2, you be training two types of networks to detect if a sentence is objective (a statement of fact) or subjective (an opinion).

- You'll also look inside the network to see what it is learning (and use a software tool to gain insight on particular predictions).

- Dataset used for training: Pan & Lee (Cornell) created the 'subjective' subsetting sentences: More reviews from Rotten Tomatoes
  - assumed to be subjective (not always true)

Objective sentences: plot summaries from IMDB
  - assumed to be objective (stated as fact)
  - about what happened in movie (not always true)

- Do no special tokenization: always have "unknown" token for words not in embedding matrix
  - will use GloVe embeddings: again, but larger dimension 300.

- So the input to the models in A2 is the sequence of word embeddings.

The movie made me feel sad.

0
1
2
3
4
5
6
7
8
9
sentences are of different lengths, while neural nets (except RNNs) are mostly set up for fixed-size inputs. We must "pad" inputs with zeroes if necessary, or "zero-pad" to make a batch all equal length.

A baseline model is used first: 1. 
- Compute the average across all word embeddings — a little in every input.
- What does this accomplish? "Compressing meaning".
- Get a fixed-length vector. Perhaps 100 dimensions is exact to encode the meaning.
- Feed result into an MLP (multi-layer perceptron) of just 1 neuron (just 1 layer)

Works surprisingly well.

Average is often used to represent sentences. 
- Why? They are same size as embedding.
- Can explore what it means.

Method 2: A Convolutional Neural Net (CNN)
- Consider the input again.

embed size: 100

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
- would like to look for: 1. Single word 3-4.
  - Core
- train kernels of size \( K \times 100 \)
  where \( K = \) width in terms of *words.*

- Recall: Kernels "scan" (sweep) across picture
  the field of a picture. // Error: what is learned
  in picture cues
- In this case a \( 2 \times 100 \) kernel would
  just sweep across the sentence once.

- It would be trained to look for
  a 2-word "pattern" of meaning
  that would contribute to leaming subjective
  adjective

- Is one enough? - No, suspect
  we would need several
- Is one size enough? - Maybe not

\[
\Rightarrow A_2 \text{ suggests have } A_1 \text{ kernels of size } K_1 \times 100
\]
\[
\Rightarrow A_2 \text{ of size } K_2 \times 100 \text{ (note)}
\]
- Each one randomly initialized.

- What is the output size that you get
  from an input sentence of \( N \) words and
  a kernel of size \( 2 \times 100 \)? (where 100 is embedding
dimension)

\[
\Rightarrow \text{get } N-1 \text{ values or just } N_1 \text{ sets}
\]
- Similar for \( K_2 \), get \( N-K_2+1 \) values.

\[
\Rightarrow \text{vow in } A_2 \text{ suggests picking the maximum}
\]
axial across all \( N-K+1 \) values 

\[
\Rightarrow \text{feed all values from all kernels}
\]
into an MLP.
- does it make sense what is happening.

- it should look for patterns of 1-6 words or so that give sense of objective or subjective.

- how could you know what the kernels are trained to look for? closest words: discuss.

- I asked to do this in A2; these should be textual features just like vision can has visual features.

  Recurrent Neural Network

- Aside RNNs have traditionally been used for text -> have been replaced by Transformers.
  - were problematic all along -> Convergence tricky; GPU unclear
  - create bottleneck for information flow -> hidden vector
— another way to look at what the neural net is looking for is to ask which word(s) were significant in the decision using the integrated gradient method.

— talk through details of assignment

— dataset/tokenization/iterators are given

— gradco — how/what
- In introduction on embeddings.
  - Words are now numbers.
  - Just like pictures become numbers.
    - E.g., each pixel is R, G, B numbers.
    - Each pixel has 3-dimensional Red, Green, Blue.
    - Each word has 2-0 in AIBS, 8D in A154.
    - GloVe has 50-100-300.

=> Far more information in a word's embedding than a pixel — e.g., 100's numbers vs 3.

⇒ Those 100 numbers might contain all meaning?

- What about "1 picture is worth 1000 words."
  - Picture has many pixels.

- CNN Review

  - CNN has kernels that are convolved with pixels.
  - Kernel is learned through backprop.
  - Have multiple.
    - E.g., 10, 15, 20, etc.