ECE 1786 Lecture #3

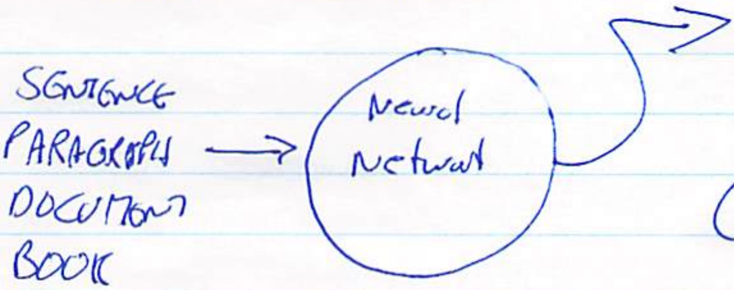
Last Day: How Word Embeddings are Trained.

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
→ range $-1 \longrightarrow +1$

② Named Entity Recognition
- identify something that can describe in many ways.
- e.g. specific reason for quitting smoking
"makes me calm" ≡ "relaxes me"

③ Style of talking
"change talk" } in behavior change
"sustain talk"

④ Politics - left v. right.

⑤ Depression / Anxiety in speech.

⑥ Suicidality → FaceBook suicide detector. (Google to see)

⑦ Lawyers → look for specific facts/refs

- 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). ⇒ DEMO w. gradio

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

- Dataset used for training: Pang & Lee (Cornell) created the 'subjective'

  SUBJECTIVE SENTENCES: movie reviews from Rotten Tomatoes
  - assumed to be subjective (not always true)

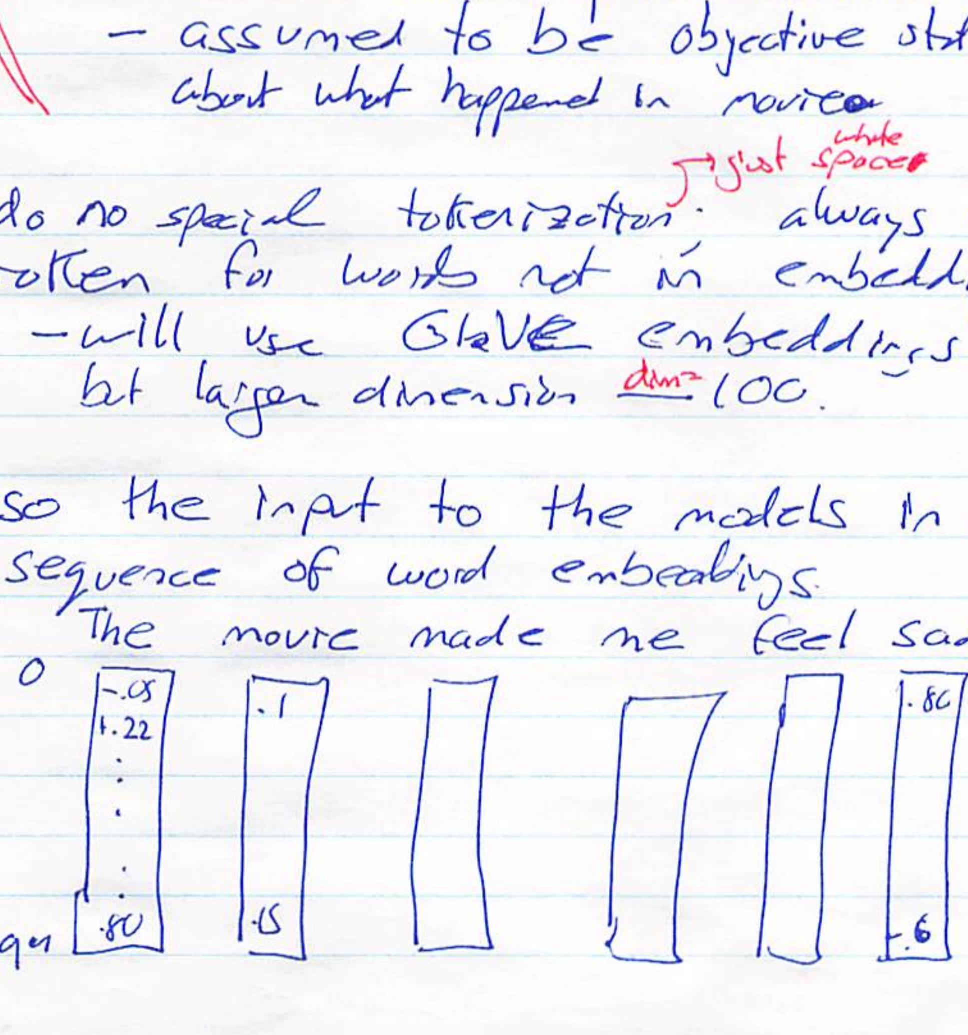
  OBJECTIVE SENTENCES: plot summaries from ~~IMDB~~ IMDB
  - assumed to be objective statements of fact about what happened in movie (not always true)

  source is important

- do no special tokenization: → just white space. always have "unknown" token for words not in embedding matrix
  - will use GloVe embeddings again, but larger dimension dim=100.

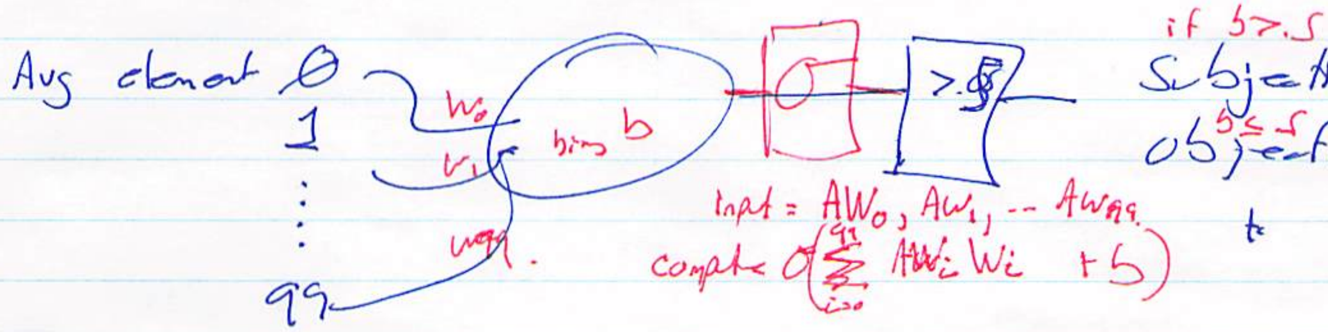
- so the input to the models in A2 is ~~the~~ → a sentence of words that is converted into a sequence of word embeddings.

The movie made me feel sad.



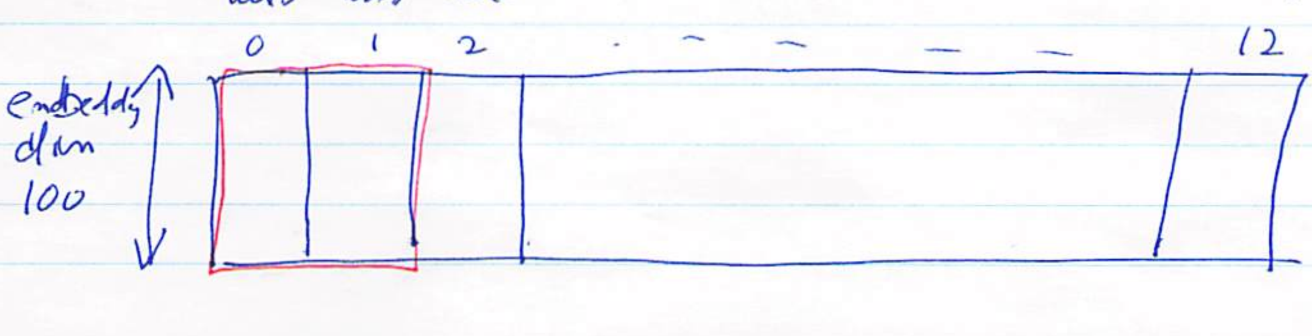
- sentences are of different lengths, while neural nets (except RNNs) are mostly set up for fixed-size ~~inputs~~ inputs.
  - → ~~can~~ must "pad" inputs with zeroes ~~if necessary~~ ~~(will be fun error)~~ to make a batch all equal length

- A baseline model is ~~done~~ used first i. A2 (→ discuss.)
  - Compute the average across all word embeddings — a little in assign 1 - recall $\frac{\text{one} + \text{ten}}{2} \approx \text{five}$
  - → what does this accomplish → "compressing meanings" $\frac{\text{happy} + \text{sad}}{2}$.
  - → get a fixed-length vector.
    - perhaps 100 dimensions is enough to encode 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.
- → $W_0 - W_{99}$ same size as embedding → can explore what it means. → Review CNNs

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



- would like to look for ① single words that indicate subj. obj. 3–4.
CNN ② pairs + 3, 4, 5....

- 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; // review what is learned in picture CNNs
- in this case ~~the~~ a ~~K~~ 2 $\times 100$ kernel would just sweep across the sentence once.

vs
vision CNN

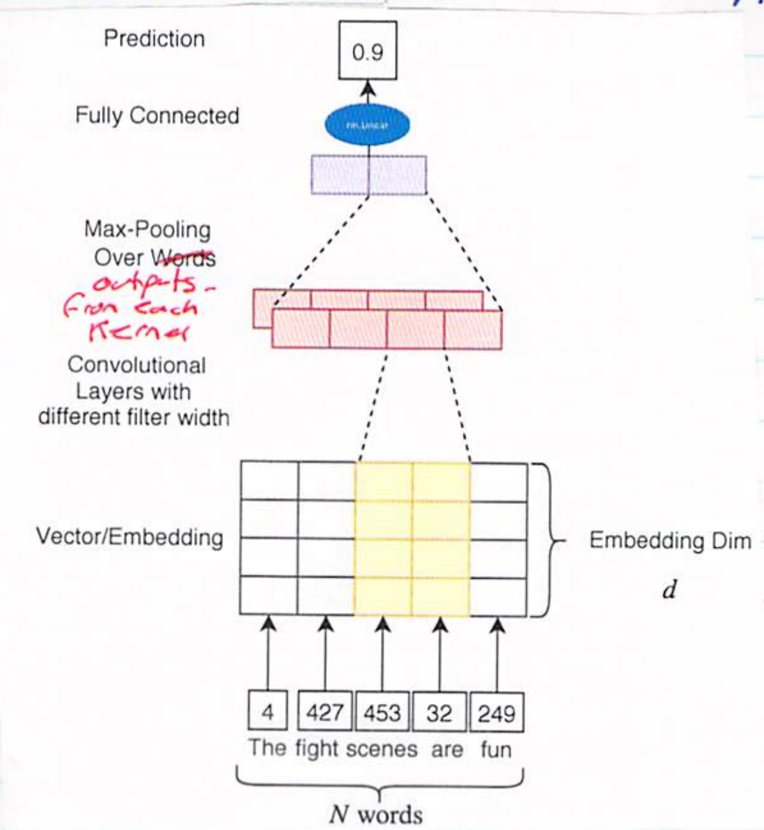
- it would be trained to look for a 2-word "pattern" of meaning that would contribute to learning subjective/objective

- is one enough? – no, suspect would need several
- is one size enough → maybe not

⇒ A2 suggests have n_1 kernels of size $K_1 \times 100$
  ⋮ n_2 " " " $K_2 \times 100$
- each one (kernel) randomly initialized.

- what is the output size that you get from an input sentence of N words and a kernel of size 2×100? (where 100 is embedding dim)
  - assume that stride = 1 (recall?)
⟶ get $N-1$ values out. – got n_1 sets
  – similar for K_2, get $N-K_2+1$ values.
⇒ Yoon [4] in A2 suggests picking the maximum across all $N-K+1$ values ⇒ i.e. max pool
⇒ 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 ~~impact~~ sense of objective or subjective.

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

  ↳ ie what do they learn?

→ asked to do this in A2: these should be textual features just like ~~vision~~ vision CNN has visual features.

Recurrent Neural Networks

→ [Aside RNNs have traditionally been used for text
- have been replaced by Transformers.
→ were problematic all along → convergence tricky; LSTM GRU unclear
→ create bottleneck for information flow → hidden vector.
→ will skip in favour of:

- 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

- gradio - show/motivate

Lecture 3 Addenda.

- in introduction on embeddings.
  - words are now numbers.
  - just like pictures became numbers
    → e.g. each pixel is R, G, B numbers.
  - each pixel has 3-dimensions Red Green, Blue.
  - each word has 2-D in A1B3 8D in A1S4.
    GloVe has 50 → 100 → 300 D.

⟹ far more information in 1 word's embedding than 1 pixel - e.g. 100's numbers vs 3.

→ those 100 numbers might contain all meaning?

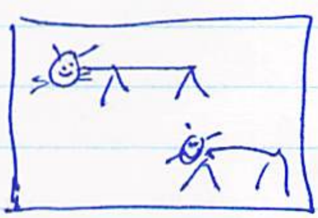
- What about "1 picture is worth 1000 words"
  - picture has many pixels

- baseline works surprisingly well - if keep training will get better

---

- CNN Review

picture



- CNN has kernels that are convolved with picture.
- Kernel is learned through back prop.
- have multiple

e.g.

etc