Video 31

Sunday, November 29, 2020 3:49 PM

Sequence modelling

Goal: model the distribution over a language's sentences.
(e.g., over English sentences)

Obtain rule of conditional probabilities:

$$p(w_1, \dots, w_T) = \prod_{t=1}^{T} p(w_t \mid w_1, \dots, w_{t-1})$$

$w_1, \dots, w_T$: that word in a sentence

T: is the # words in the sentence

distribution over each word depends on all previous words.

Approach w/ a RNN:

ex: the quick brown fox

Training time:

using each word in the sentence as both the input & target of the model.

(i) Each word appears as a target before it appears as an input.

→ no information flow between the word used as an input and the word used as a target.

→ RNN cannot simply "copy" the sentence.

Inference time:

Sample each of the words in sequence from the predictions of the model.

For each time step:

→ compute the output units
(one output unit per word in our language dictionary)

output units are passed through a softmax to produce a distribution over the words in our dictionary.

→ sample the word from the word distribution predicted by the model

for ex: pick the word w/ the highest output.

→ feed the word back to the model as the input to the next time step.

⚠ when the dictionary is large, predicting can be computationally difficult

↳ use a hierarchical softmax

↳ model language at a character level.

↳ the model will learn & predict one character at a time
(rather than one word at a time)

↳ we only have to model a fixed "smaller" number of characters
(i.e., letters and punctuations)

Example: trained on Wikipedia
character-level RNN

politics:
He was elected President during the Revolutionary War and forgave Opus Paul at Rome. The regime of his crew of England, is now Arab women's icons in and the demons that use something between the characters' sisters in lower coloration were always operated on the line of the ephemerable street, respectively, the graphic or other facility for deformation of a given proportion of large segments at RTL(S). The B every chord was a "strongly cold internal palette pour even the white blade."

transportation

not a word

(Geoff Hinton's course from 2011)

→ text coherent

→ local coherency of the text.

Neural Machine Translation

goal: take a sentence in one language (e.g., French) & output its translation in another language (e.g., English)

why is this challenging?

↳ sentences may not have the same length.

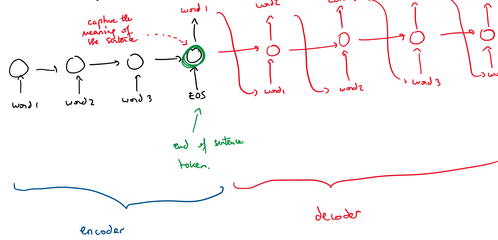
↳ words won't necessarily be aligned across languages

Instead: have the RNN

(1) read the sentence in one language

(2) memorize its meaning

(3) generate a sentence in the other language



⚠ encoder & the decoder do not share weights.

Powerful approach: encoders & decoders can be used for ≠ language pairs.

multiple languages & mapping them to a common abstract semantic space

=> we can translate despite not having aligned datasets for a pair of languages