Sequence-to-sequence models: Part II

Seq2Seq / encoder-decoder models

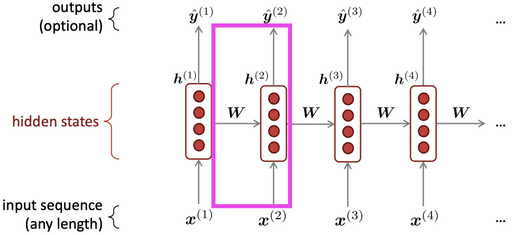
Ryan Chan

3 May 2023

Outline

Recap: Recurrent Neural Networks

- RNNs [Hopfield, 1982; Rumelhart et al., 1985] are capable of conditioning the model on *all* previous tokens (in theory)



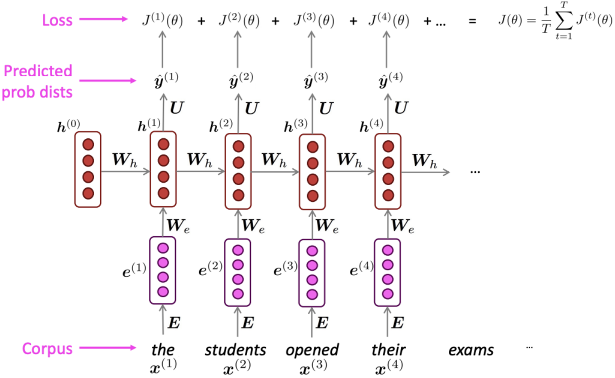
[1]

- Core idea: the hidden layer from previous timestep provides a form of memory or context that informs decisions to be made later in the sequence
- The same weights are applied at every timestep

[1] Manning et al. [2017]

Recap: Training RNN language models

- We use the cross-entropy loss between the output distribution $\hat{\boldsymbol{y}}^{(t)}$ and $\boldsymbol{y}^{(t)}$ (and average over the entire batch/corpus)
- We use teacher-forcing: input the previous words/tokens to predict the next

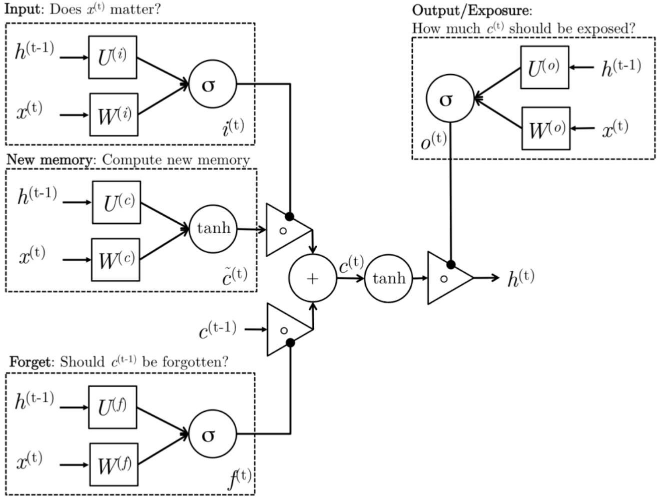


[2] Manning et al. [2017]

Recap: LSTMs

- "Vanilla" RNNs can be difficult to train
  - Vanishing/exploding gradients mean its hard to preserve memory over many timesteps
  - The hidden layers/states and weights are asked to do a *lot* of work
- Long short-term memory (LSTM) networks [Hochreiter and Schmidhuber, 1997; Gers et al., 2000] are the most commonly used extension to RNNs
  - In practice, could preserve information to about 100 timesteps rather than 7 in "vanilla" RNNs
  - See "Understanding LSTM Networks" [Olah, 2015] for a bit more of a deeper explanation and walkthrough

Recap: LSTM architecture

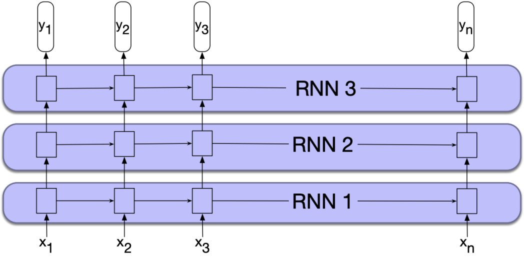


3

[3] Manning et al. [2017]

Stacked / Deep RNNs

- We can use the entire sequence of outputs from one RNN as the input of another RNN to create a stack of RNNs



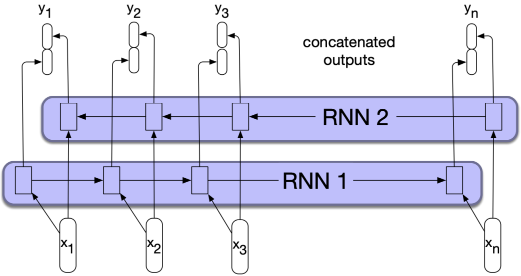
[4]

  - Outputs of one layer serves as input to a subsequent layer
  - Typically outperforms single-layer networks but increases training cost

[4] Jurafsky and Martin [2019, Chapter 9]

Bidirectional RNNs

- Run two separate RNNs: left-to-right (we've seen above) and right-to-left
- Then concatentate the hidden states from both RNNs
  - Alternatives: element-wise mean or sum

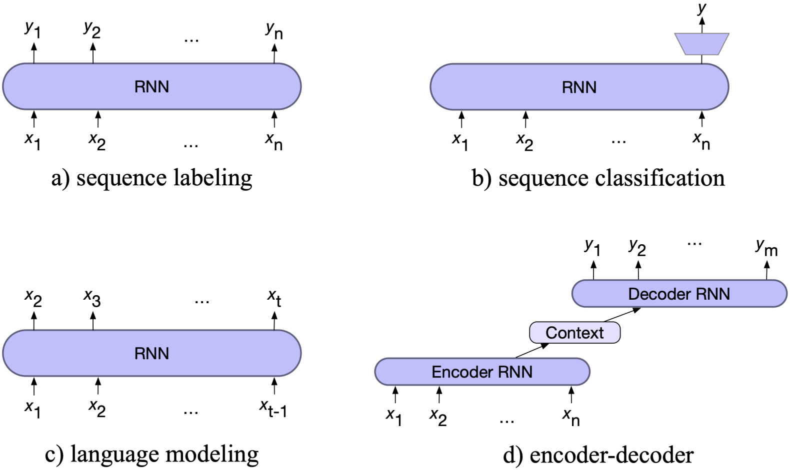


[5]

  - Effective if you have access to a full input sentence (to encode a sentence)
  - "BERT": Bidirectional Encoder Representations from Transformers

[5] Jurafsky and Martin [2019, Chapter 9]

RNN tasks

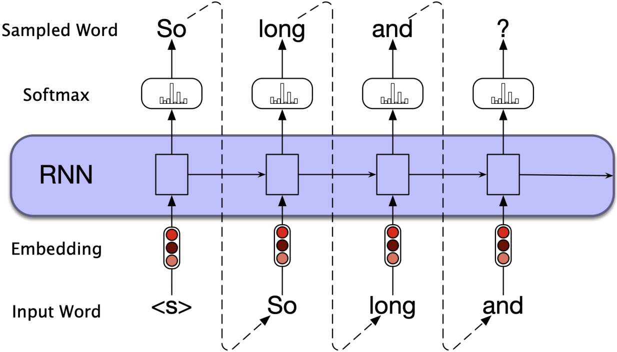


[6]

[6] Jurafsky and Martin [2019, Chapter 9]

Language modelling

- Causal language modelling / autogregressive generation: incremental generation of words by repeated sampling of the next word conditional on previous choices
  - <s>: beginning of sentence marker
  - </s>: end of sentence marker

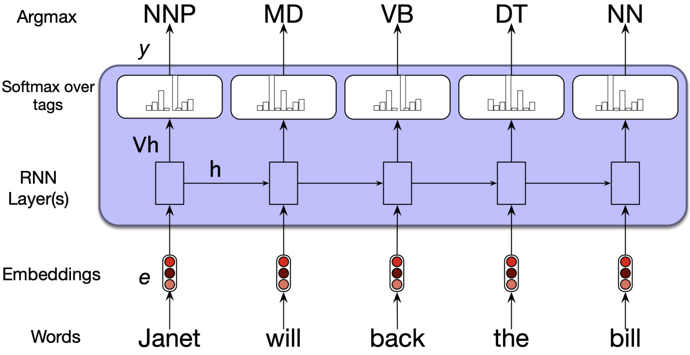


7

[7] Jurafsky and Martin [2019, Chapter 9]

Sequence labelling

- In sequence labelling, we aim to assign a label to each element in the sequence:
  - e.g. part-of-speech tagging, named entity recognition

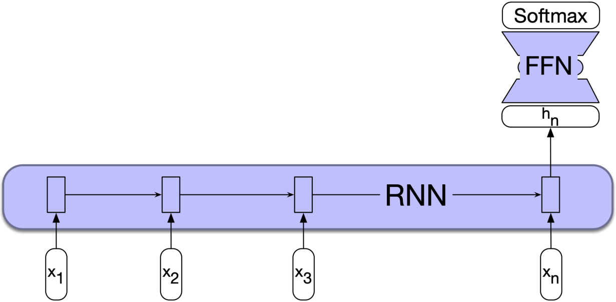


[8]

[8] Jurafsky and Martin [2019, Chapter 9]

Sequence classification

- In sequence classification, we use the RNN to classify the entire sequence (rather than the individual elements/tokens):
  - e.g. sentiment analysis, spam detection, document-level topic classification
- The loss comes from the final task, and we backpropagate through the entire network, i.e end-to-end training

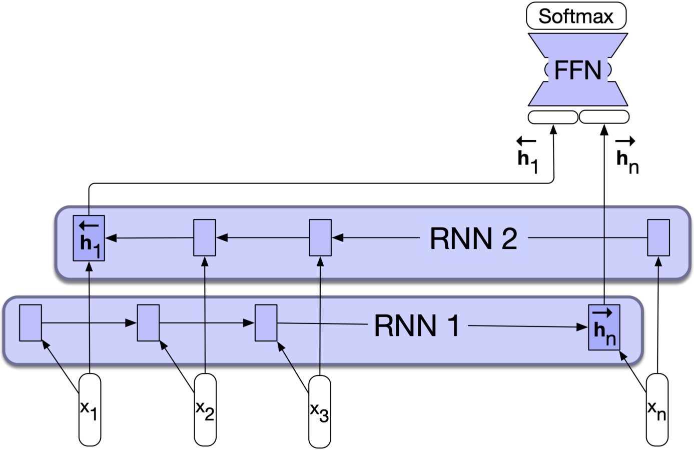


9

[9] Jurafsky and Martin [2019, Chapter 9]

Sequence classification (with bidirectional RNN)

- If we have access to the full sequence, a bidirectional RNN can provide a more effective encoding of the sequence

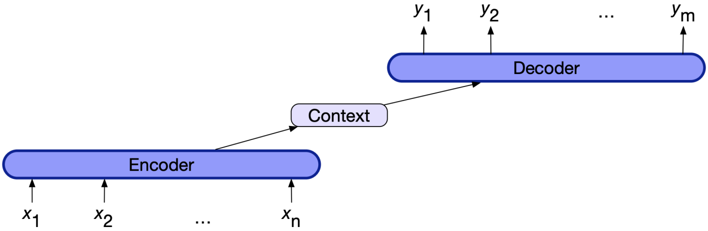


10

[10] Jurafsky and Martin [2019, Chapter 9]

Seq2Seq / Encoder-decoder RNN

- An architecture which takes in an input sequence and outputs a sequence that is of different length
  - Sequence labelling (part-of-speech tagging / named entity recognition) have two sequences that have the same length
- Some example use cases:
  - Machine translation, summarisation, question answering



[11]

[11] Jurafsky and Martin [2019, Chapter 9]

Seq2Seq architecture

- Able to generate context appropriate, arbitrary length output sequences *given* an input sequence
  - Encoder network: takes in input sequence to produce a contextualised representation of it
  - Decoder network: takes in the context produced to generate a text-specific output sequence
- For example, an RNN Seq2Seq model:
  - The *encoder RNN* produces a context

$$\boldsymbol{c} = f(\boldsymbol{h}^e_{1:n}) \tag{1}$$

    where $\boldsymbol{h}^e_{1:n} = \{\boldsymbol{h}^e_1, \ldots, \boldsymbol{h}^e_n\}$ are the hidden states of the encoder RNN
  - The *decoder RNN* accepts $\boldsymbol{c}$ and generates a sequence of hidden states $\boldsymbol{h}^d_{1:m}$ to form corresponding sequence of output states $\boldsymbol{y}_{1:m}$ (where possibly $m \neq n$)

Machine translation

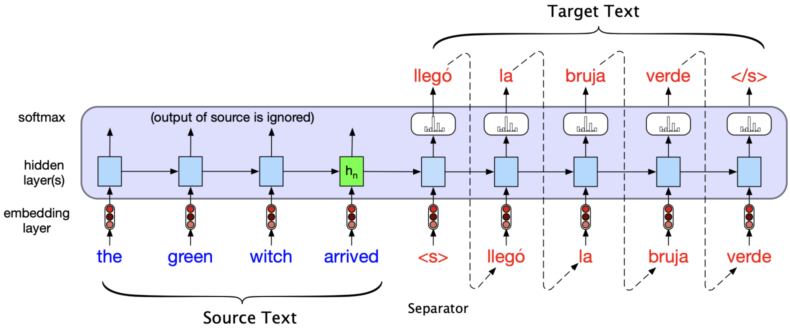
- Machine translation (MT) is the task of translating a sentence x in one language (source language) to a sentence y (target language)
- Early 1950s: MT research started but relied heavily on simple rule-based systems using word substitution
- 1990s-2010s: Statistical machine translation (SMT)
  - Want to find the "best" target sentence y given the source sentence x:

$$\text{argmax}_y p(y|x) = \text{argmax}_y p(x|y)p(y) \tag{2}$$

  - $p(x|y)$: translation model whose goal is to model how phrases are translated (trained with sentence pairs)
  - $p(y)$: language model whose goal is to write sensible sentences
- Systems were extremely complex which required lots of feature engineering and human effort to maintain
- Repeated effort for different language pairs

Seq2Seq for machine translation

- Neural machine translation (NMT): use a single end-to-end neural network
- Each training example is a pair of strings: `source, target`
  - Concatenate with a separator token, e.g. `<s>`, `<EOS>`, etc.



[12] Jurafsky and Martin [2019, Chapter 9]

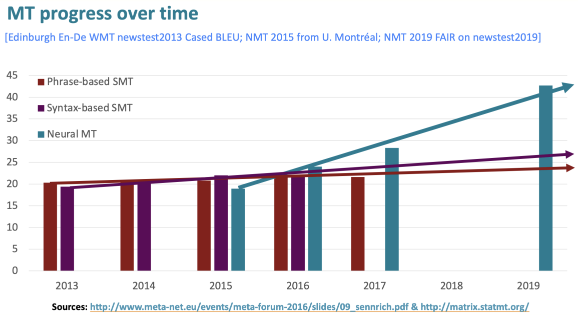
Seq2Seq for machine translation

- The sequence-to-sequence model for NMT is an example of a conditional language model
  - Language model: the decoder is predicting the next word in the target sentence y
  - Conditional: the predictions are also *conditioned* on some source sentence x
- NMT *directly* estimates $p(y|x)$:

$$p(y|x) = p(y_1|x) \cdot p(y_2|y_1, x) \cdot p(y_3|y_{1:2}, x) \cdots p(y_T|y_{1:T-1}, x) \tag{3}$$

Seq2Seq successes in machine translation

- BLEU: Bilingual Evaluation Understudy
  - Compares a machine-written translation to one or several human-written translation(s) and computes a similarity score based on *n*-gram precision
  - Adds a penalty for translations which are too short
- BLEU is useful but not perfect: there are many valid ways to translate a sentence

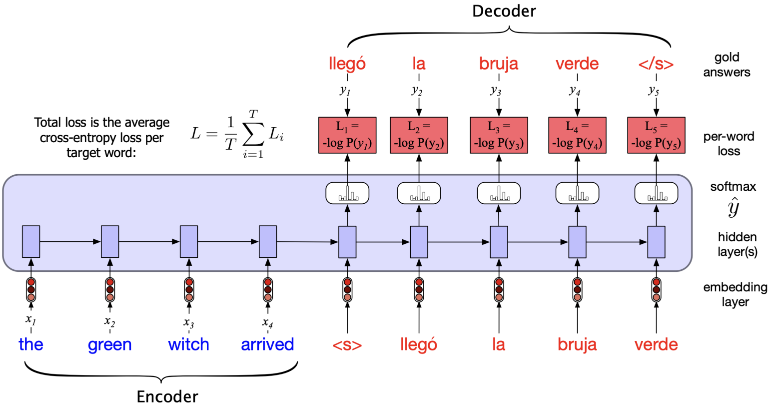


13

[13] Manning et al. [2017, Lecture 6]

Training Seq2Seq

- Use teacher-forcing and compute the average loss over the predicted sequence
- Encoder-decoder architectures are trained end-to-end



14

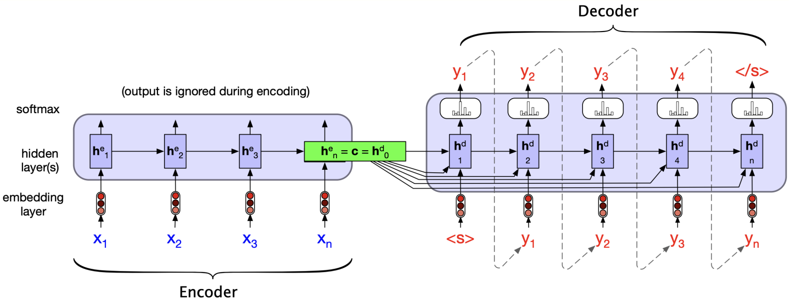
[14] Jurafsky and Martin [2019, Chapter 9]

Propagating the context

- Problem: the influence of the context, $\boldsymbol{c}$, decreases as the output sentence is generated
- Solution: Make $\boldsymbol{c}$ available at each step of the decoding process:

$$\boldsymbol{h}_t^d = g(\hat{\boldsymbol{y}}_{t-1}, \boldsymbol{h}_{t-1}^d, \boldsymbol{c}), \quad \text{for } t = 1, \ldots, m \tag{4}$$

where $\boldsymbol{c} = f(\boldsymbol{h}_{1:n}^e)$ and $\boldsymbol{h}_0^d = \boldsymbol{c}$

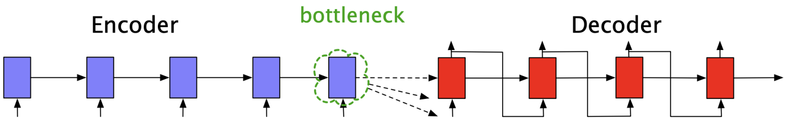


[15]

[15] Jurafsky and Martin [2019, Chapter 9]

Weaknesses of Seq2Seq

- Problem: the context $\boldsymbol{c}$ from the encoder must represent everything about the meaning of the source text
  - Information at different parts of the source text may not be equally represented in the context vector
- The attention mechanism is a solution: allows the decoder to access all hidden states in the encoder, not just $\boldsymbol{c}$



[16]

[16] Jurafsky and Martin [2019, Chapter 9]

References

Gers, F. A., Schmidhuber, J., and Cummins, F. (2000). Learning to forget: Continual prediction with LSTM. *Neural Computation*, 12(10):2451–2471.

Hochreiter, S. and Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8):2554–2558.

Jurafsky, D. and Martin, J. H. (2019). Speech and language processing (3rd (draft) ed.).

Manning, C., Socher, R., Fang, G. G., and Mundra, R. (2017). CS224n: Natural Language Processing with Deep Learning.

Olah, C. (2015). Understanding LSTM Networks.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1985). Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science.