COMP3220 — Document Processing and the Semantic Web

Week 06 L1: Advanced Topics in Deep Learning

Diego Mollá

Department of Computer Science
Macquarie University

COMP3220 2020H1

Programme

Reading

- Deep Learning book, section 8.1.

Programme

Generating Text Sequences

- One of the advances of deep learning versus shallower approaches to machine learning is its ability to process complex contexts.
- This has allowed significant advances in image and text processing.
- We have seen how to process text sequences for text classification.

Text generation as a particular case of text classification

- Given a piece of text ...
- Predict the next character.

Text Generation as Character Prediction

- Our training data is a set of samples of the form:

  Input Text fragment.

  Label Next character to predict.

- We do not need to manually annotate the training data: the data are self-annotated.
- This means that we can easily gather training data for text generation.
- This is the idea for training language models (next slide).

Language Models

- Given a collection of text, we can train a language model that can be used to generate text in the same style.

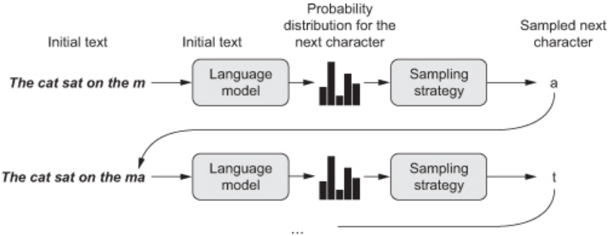


Figure 8.1 of Chollet (2018).

Implementing Character-level LSTM Text Generation

- The architecture of the model is of the kinds we have seen for text classification.
  - The input is a sequence of characters.
  - The "class" to predict is the next character to generate.
- If we add an embedding layer after the input, This layer will learn character embeddings.

```
model = tf.keras.models.Sequential()
model.add(layers.Embedding(len(chars), 20, input_len=maxlen))
model.add(layers.LSTM(128))
model.add(layers.Dense(len(chars), activation='softmax))
```

Generating Text

- Remember that the output of a prediction is a probability distribution.
- To generate the next character, we can sample from the probability distribution.
- We can determine how deterministic the sampling is:
  - We can always return the character with highest probability ...
  - Or we can select a character randomly ...
  - Or we can do something in between, according to a "temperature" parameter.

```
import numpy as np
def reweight_distribution(original_distribution, temperature=0.5):
    distribution = np.log(original_distribution) / temperature
    distribution = np.exp(distribution)
    return distribution / np.sum(distribution)
```

Figure: Different Reweightings

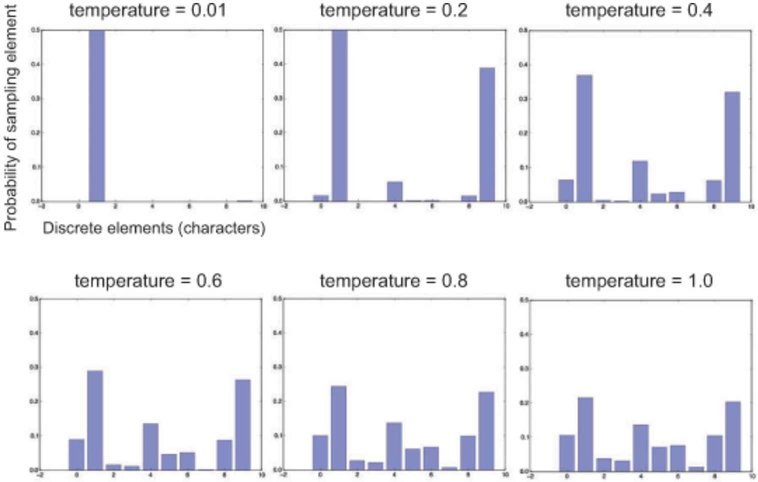


Figure 8.2 of Chollet (2018)

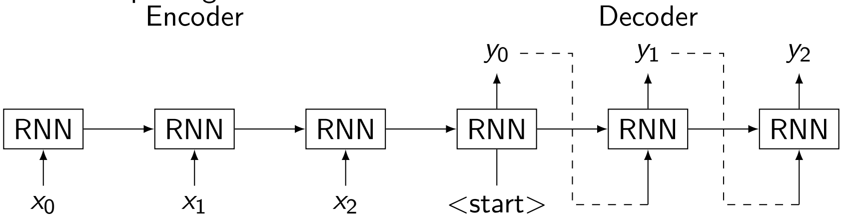
Example

See notebook . . .

Programme

The Encoder-Decoder Architecture

- Composed of an encoder and a decoder.
  - The encoder can be an RNN chain that takes the input.
  - The decoder can be an RNN that takes the output of the previous RNN as input.
- Revolutionised machine translation and many other text processing applications.
- The encoder stage can be something non-textual, e.g. images for caption generation.

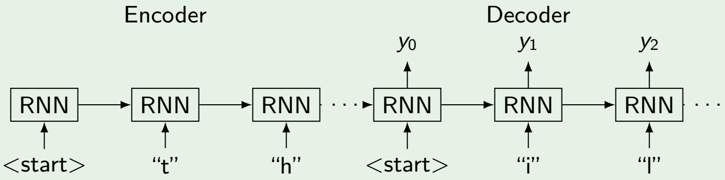


Training the Encoder-Decoder Architecture

A common approach to train the encoder-decoder architecture is to apply teacher forcing:

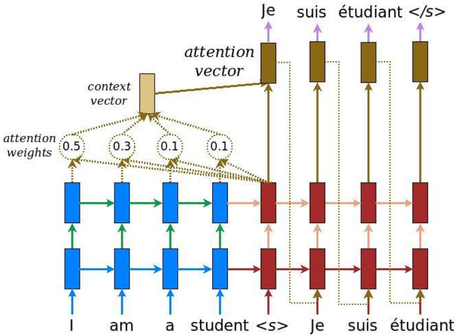
- Use the target sequence to guide the training of the decoder.
- For example, in an English to French machine translation system, we feed the target French translation to the decoder.

"The weather is fine" $\longrightarrow$ "Il fait bon"



Attention: An Improvement to the Encoder-Decoder Architecture

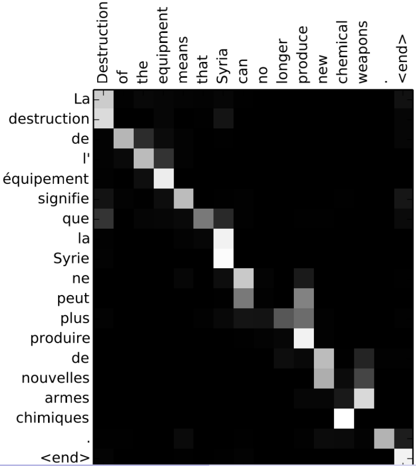
Attention is an enhancement in the seq2seq architecture that allows to focus on parts of the input during the generation stage by the decoder.



https://github.com/tensorflow/tensorflow/blob/r1.13/tensorflow/contrib/eager/python/examples/nmt_with_attention/nmt_with_attention.ipynb

Attention for MT

Very useful to start understanding the decision processes of the model.



Attention in Caption Generation

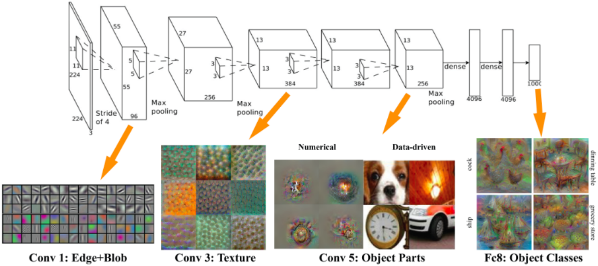
A woman is throwing a frisbee in a park.

Xu et al. (2015) arXiv:1502.03044

Programme

Interpretability

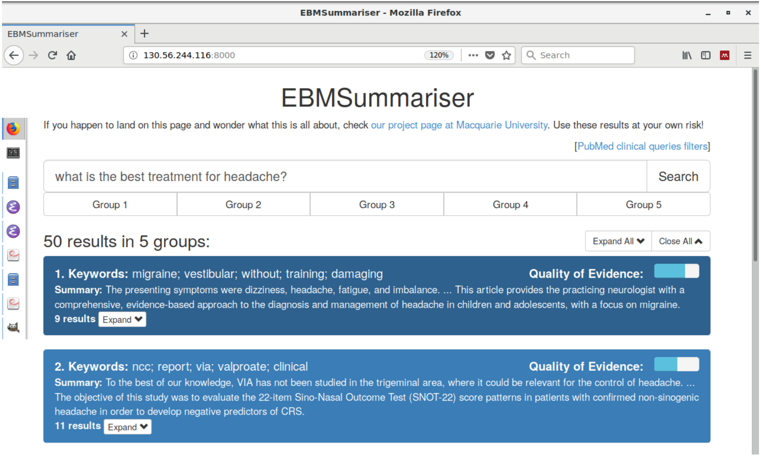
- It is very difficult to interpret most weights in a neural model.
- Approaches like attention help to visualise some of the processes but much more is needed.
- Current research in image processing can visualise interpretations of middle layers. How to do the same with text?



http://vision03.csail.mit.edu/cnn_art/index.html

Justifiability

How can someone justify a decision made by a neural model?



Small Training Data

- Deep learning excels when there are large volumes of training data.
- But obtaining labelled training data is expensive ...
  - ⇒ We can add unsupervised and semi-supervised tasks, e.g. pre-train word embeddings.
- ... and some domains and languages have very little data ...
  - ⇒ Transfer learning: Pre-train on one domain and adapt the learnt model to another domain.

Small Training Data

- Deep learning excels when there are large volumes of training data.
- But obtaining labelled training data is expensive ...
  - ⇒ We can add unsupervised and semi-supervised tasks, e.g. pre-train word embeddings.
- ... and some domains and languages have very little data ...
  - ⇒ Transfer learning: Pre-train on one domain and adapt the learnt model to another domain.

Small Training Data

- Deep learning excels when there are large volumes of training data.
- But obtaining labelled training data is expensive ...
  - $\Rightarrow$ We can add unsupervised and semi-supervised tasks, e.g. pre-train word embeddings.
- ... and some domains and languages have very little data ...
  - $\Rightarrow$ Transfer learning: Pre-train on one domain and adapt the learnt model to another domain.

Small Training Data

- Deep learning excels when there are large volumes of training data.
- But obtaining labelled training data is expensive . . .
  - ⇒ We can add unsupervised and semi-supervised tasks, e.g. pre-train word embeddings.
- . . . and some domains and languages have very little data . . .
  - ⇒ Transfer learning: Pre-train on one domain and adapt the learnt model to another domain.

Small Training Data

- Deep learning excels when there are large volumes of training data.
- But obtaining labelled training data is expensive . . .
  - ⇒ We can add unsupervised and semi-supervised tasks, e.g. pre-train word embeddings.
- . . . and some domains and languages have very little data . . .
  - ⇒ Transfer learning: Pre-train on one domain and adapt the learnt model to another domain.

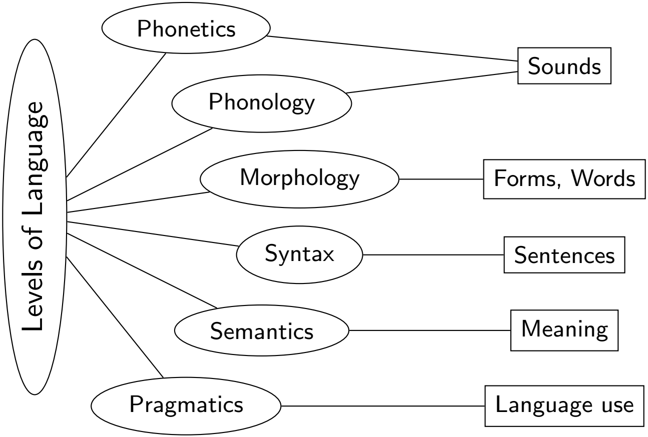
Incorporating Knowledge

- Early natural language systems could easily incorporate knowledge.
  - Ontologies, databases, information given by the user, etc.
- Deep learning approaches find this more difficult.
- Question answering and dialogue systems often do not remember what has been said before.

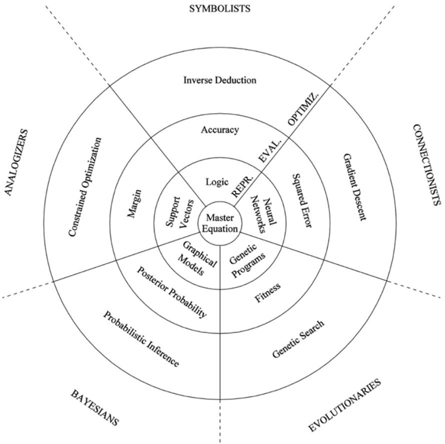
message	Where do you live now?
response	I live in Los Angeles.
message	In which city do you live now?
response	I live in Madrid.
message	In which country do you live now?
response	England, you?

Vinyals & Le (2015) https://arxiv.org/abs/1506.05869

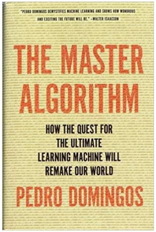
Incorporating Linguistic Knowledge



Deep Learning is Not Everything



Deep Learning is with the "Connectionists" tribe.



Take-home Messages

1. Text generation as a task of character (or word) prediction.
2. We may want to control the level of randomness when generating text based on a "temperature" parameter.
3. Describe the encoder-decoder architecture. What is this architecture good for?
4. What is teacher forced training and what is it good for?
5. Comment on current open challenges in deep learning.

What's Next

Weeks 7-12

- Semantic Web (Rolf Schwitter).
- Assignment 2 submission deadline on Friday 1 May 2020.