An Introduction to Neural Machine Translation

Prof. John D. Kelleher
@johndkelleher

ADAPT Centre for Digital Content Technology
Dublin Institute of Technology, Ireland

June 25, 2018

Outline

Image from https://blogs.msdn.microsoft.com/translation/

Higher quality neural translations for a bunch more languages

Barak Turovsky
Product Lead (and proud Russian speaker!), Google Translate

Published Mar 6, 2017

Last November, people from Brazil to Turkey to Japan discovered that Google Translate for their language was suddenly more accurate and easier to understand. That's because we introduced neural machine translation—using deep neural networks to translate entire sentences, rather than just phrases—for eight languages overall. Over the next couple of weeks, these improvements are coming to Google Translate in many more languages, starting right now with Hindi, Russian and Vietnamese.

Neural translation is a lot better than our previous technology, because we translate whole sentences at a time, instead of pieces of a sentence. (Of course

Image from https://www.blog.google/products/translate

Image from
https://techcrunch.com/2017/08/03/facebook-finishes-its-move-to-neural-machine-translation/

Linguee's Founder Launches DeepL in Attempt to Challenge Google Translate

*by **Florian Faes** on August 30, 2017*

Barely two years after bursting into the translation tech scene, neural machine translation (NMT) is everything the MT community is talking about. Microsoft, Google, Facebook, and other large technology companies have all transitioned to NMT, as did the European Patent Office and the World Intellectual Property Organization. Even end-buyers are starting to build their own systems based on open-source models.

Image from `https://slator.com/technology/linguees-founder-launches-deepl-attempt-challenge-google-translate/`

Neural Networks 101

What is a function?

A function maps a set of inputs (numbers) to an output (number)[1]

$$sum(2, 5, 4) \rightarrow 11$$

[1] This introduction to neural network and machine translation is based on: Kelleher (2016)

What is a WEIGHTEDSUM function?

$$\text{WEIGHTEDSUM}(\underbrace{[x_1, x_2, \ldots, x_m]}_{\text{Input Numbers}}, \underbrace{[w_1, w_2, \ldots, w_m]}_{\textit{Weights}})$$
$$= (x_1 \times w_1) + (x_2 \times w_2) + \cdots + (x_m \times w_m)$$

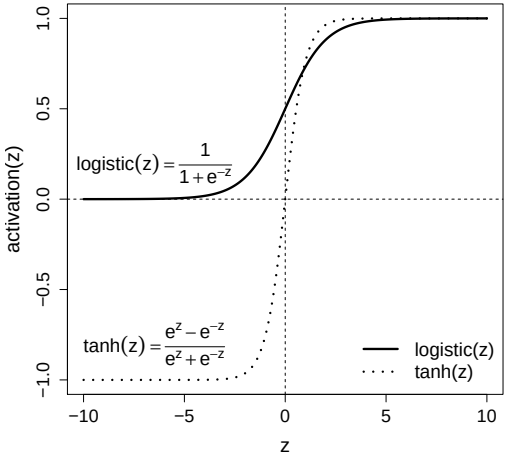
$$\begin{aligned}\text{WEIGHTEDSUM}([3, 9], [-3, 1]) &= (3 \times -3) + (9 \times 1) \\ &= -9 + 9 \\ &= 0\end{aligned}$$

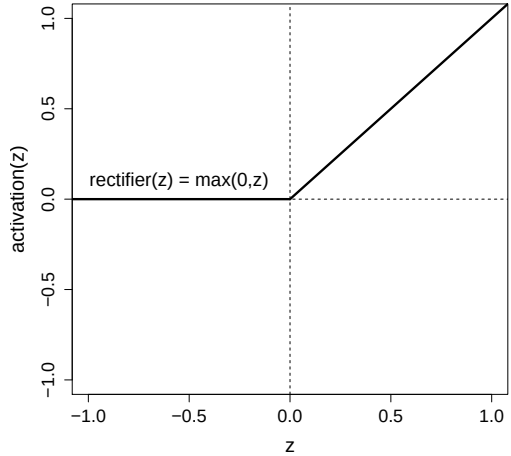
NDR

What is an ACTIVATION function?

An ACTIVATION function takes the output of our WEIGHTEDSUM function and applies another mapping to it.

What is an ACTIVATION function?





NDR

What is an ACTIVATION function?

$$\text{ACTIVATION} =$$
$$\text{LOGISTIC}(\text{WEIGHTEDSUM}((\underbrace{[x_1, x_2, \ldots, x_m]}_{\text{Input Numbers}}, \underbrace{[w_1, w_2, \ldots, w_m]}_{\textit{Weights}}))$$

$$\begin{aligned}
&\text{LOGISTIC}(\text{WEIGHTEDSUM}([3, 9], [-3, 1])) \\
&\quad = \text{LOGISTIC}((3 \times -3) + (9 \times 1)) \\
&\quad = \text{LOGISTIC}(-9 + 9) \\
&\quad = \text{LOGISTIC}(0) \\
&\quad = 0.5
\end{aligned}$$

NDR

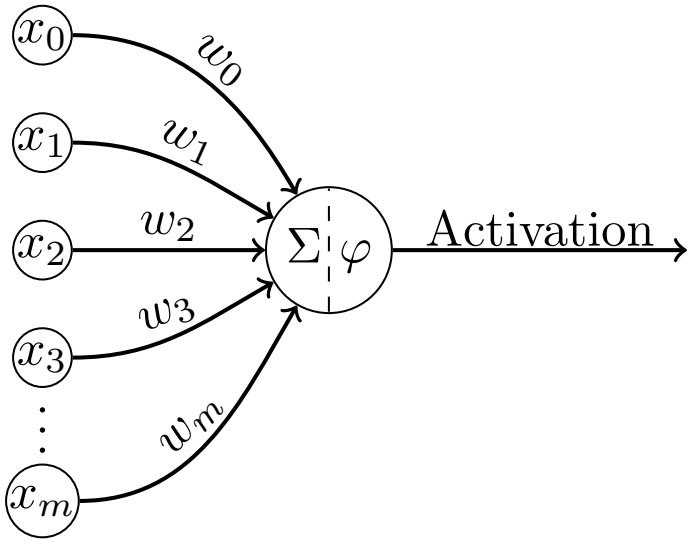
What is a NEURON?

The simple list of operations that we have just described defines the fundamental building block of a neural network: the NEURON.

$$\text{NEURON} = \text{ACTIVATION}(\text{WEIGHTEDSUM}((\underbrace{[x_1, x_2, \ldots, x_m]}_{\text{Input Numbers}}, \underbrace{[w_1, w_2, \ldots, w_m]}_{\textit{Weights}}))$$

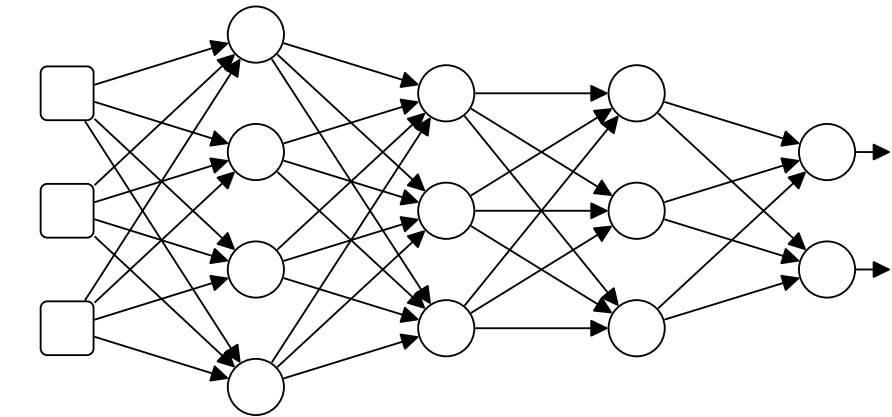
NDR

What is a NEURON?



NDR

What is a Neural Network?



NDR

Training a Neural Network

- We train a neural network by iteratively updating the weights
- We start by randomly assigning weights to each edge
- We then show the network examples of inputs and expected outputs and update the weights using Backpropogation so that the network outputs match the expected outputs
- We keep updating the weights until the network is working the way we want

NDR

Word Embeddings

Word Embeddings

- Language is sequential and has lots of words.

“a word is characteriezed by the company it keeps”

— Firth, 1957

Word Embeddings

1. Train a network to predict the word that is missing from the middle of an n-gram (or predict the n-gram from the word)
2. Use the trained network weights to represent the word in vector space.

NDR

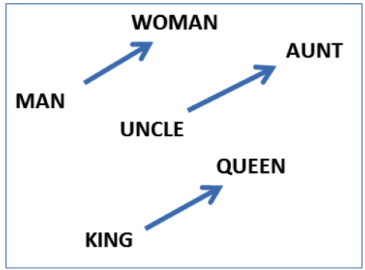
Word Embeddings

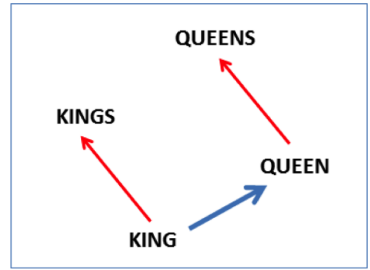
Each word is represented by a vector of numbers that positions the word in a multi-dimensional space, e.g.:

$$
\begin{aligned}
king &= < 55, -10, 176, 27 > \\
man &= < 10, 79, 150, 83 > \\
woman &= < 15, 74, 159, 106 > \\
queen &= < 60, -15, 185, 50 >
\end{aligned}
$$

NDR

Word Embeddings





$$vec(King) - vec(Man) + vec(Woman) \approx vec(Queen)^2$$

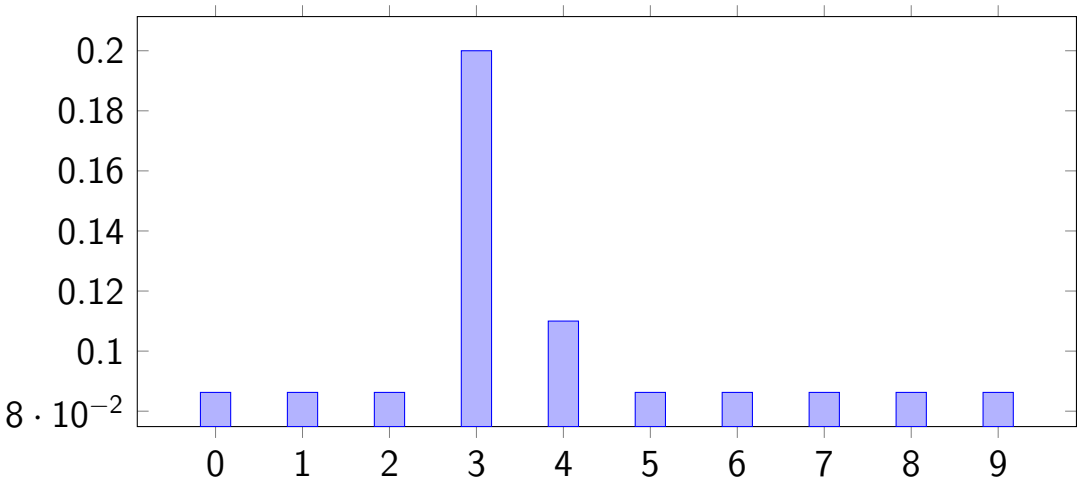
[2] Linguistic Regularities in Continuous Space Word Representations Mikolov et al. (2013)

Language Models

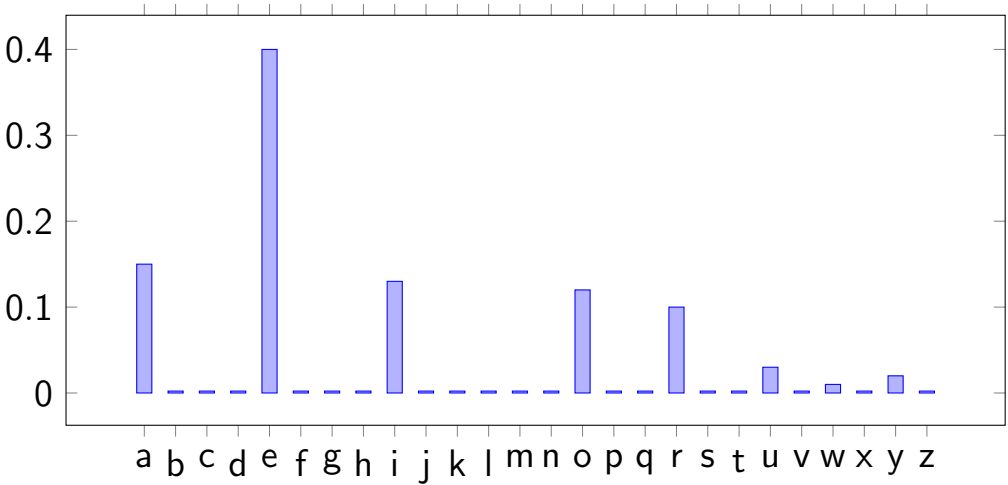
Language Models

- Language is sequential and has lots of words.

1,2,?


0.2
0.18
0.16
0.14
0.12
0.1
$8 \cdot 10^{-2}$
0
1
2
3
4
5
6
7
8
9

Th?


0.4
0.3
0.2
0.1
0
a b c d e f g h i j k l m n o p q r s t u v w x y z

- A language model can compute:
  1. the probability of an upcoming symbol:

$$P(w_n|w_1, \ldots, w_{n-1})$$

  2. the probability for a sequence of symbols[3]

$$P(w_1, \ldots, w_n)$$

[3] We can go from 1. to 2. using the Chain Rule of Probability $P(w_1, w_2, w3) = P(w_1)P(w_2|w_1)P(w_3|w_1, w_2)$

- Language models are useful for machine translation because they help with:
  1. word ordering

$$P(\textit{Yes I can help you}) > P(\textit{Help you I can yes})$$[4]

  2. word choice

$$P(\textit{Feel the Force}) > P(\textit{Eat the Force})$$

[4] Unless its Yoda that speaking

Neural Language Models

Recurrent Neural Networks

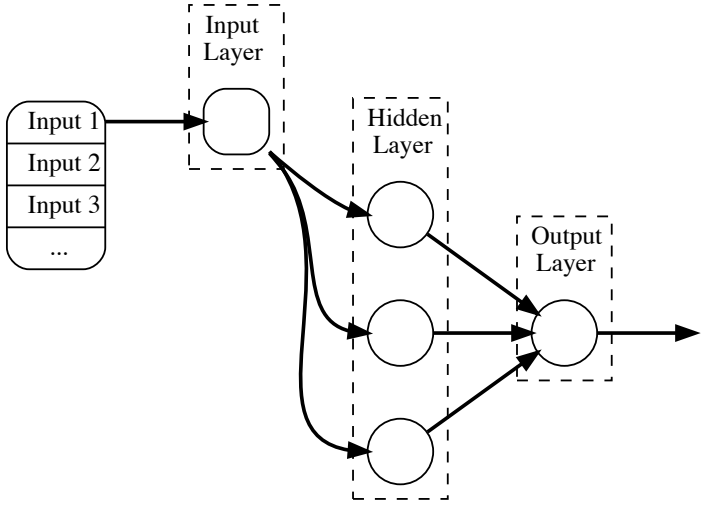
A particular type of neural network that is useful for processing sequential data (such as, language) is a Recurrent Neural Network.

Recurrent Neural Networks

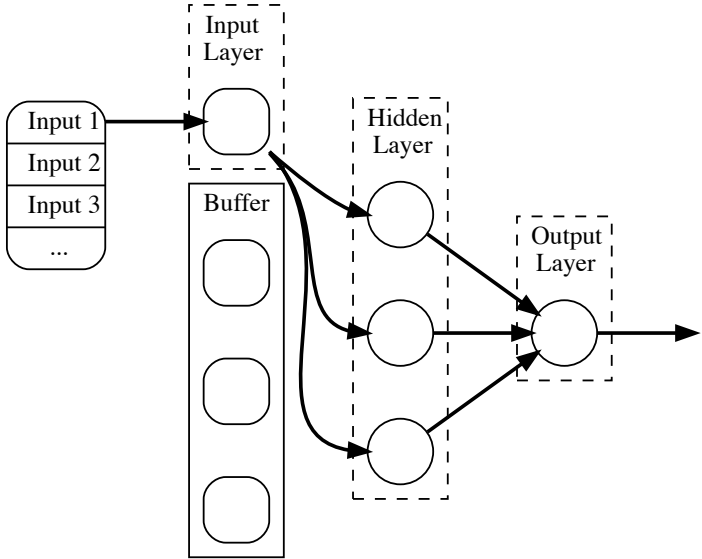
Using an RNN we process our sequential data one input at a time.

In an RNN the outputs of some of the neurons for one input are feed back into the network as part the next input.

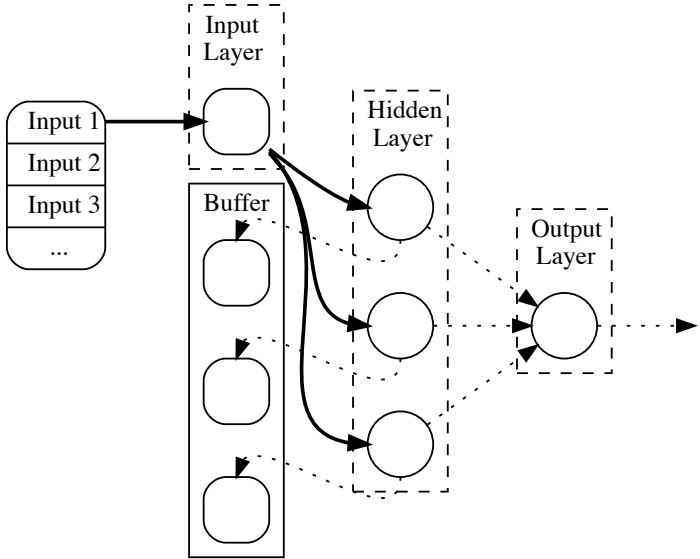
Simple Feed-Forward Network



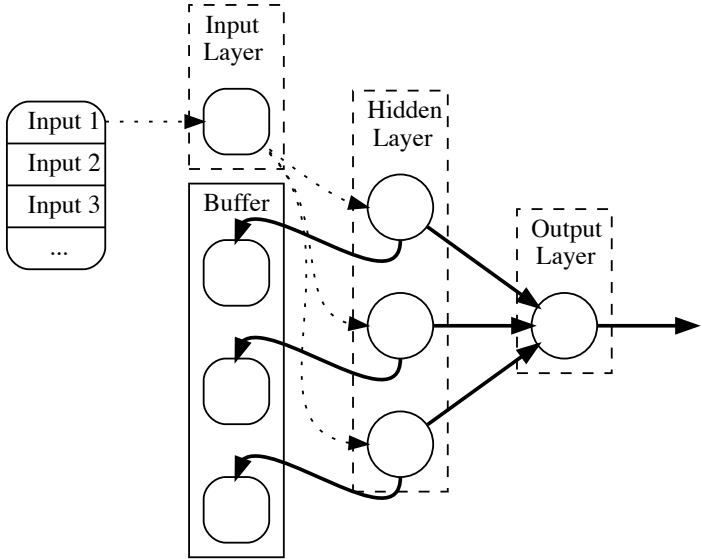
Recurrent Neural Networks



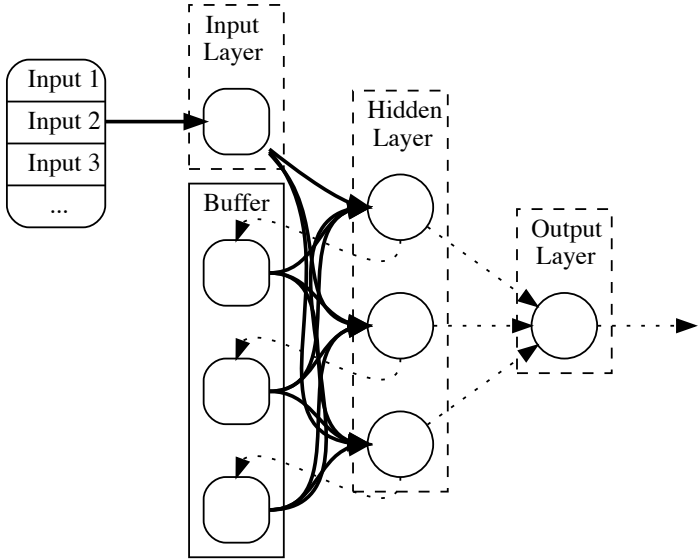
Recurrent Neural Networks



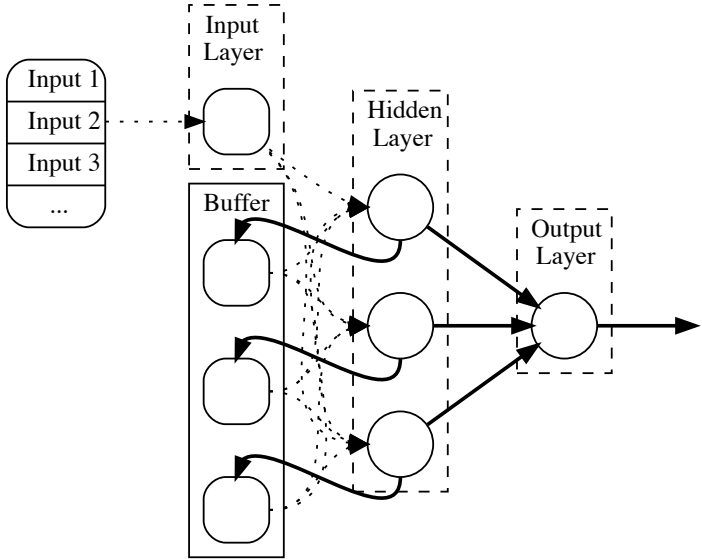
Recurrent Neural Networks



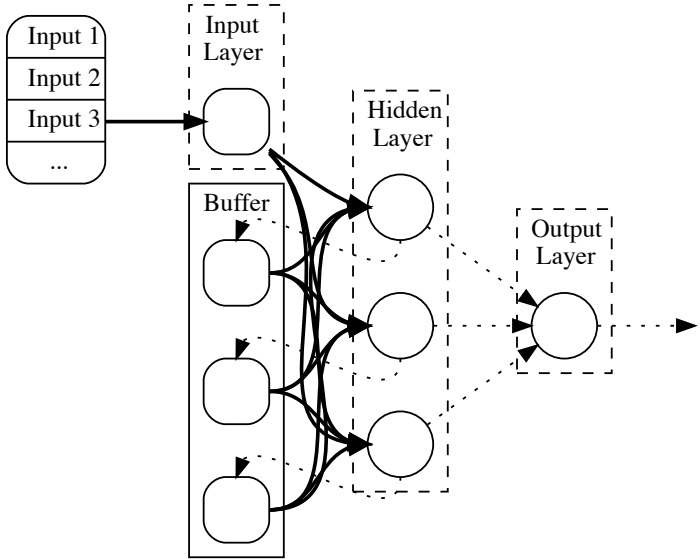
Recurrent Neural Networks



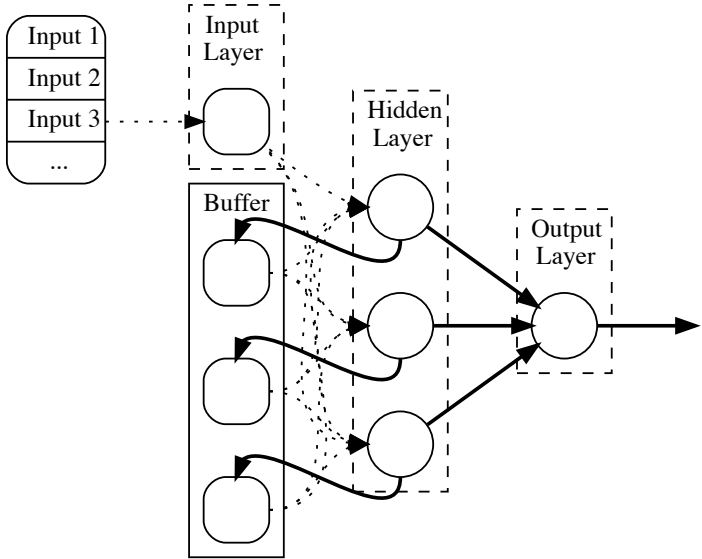
Recurrent Neural Networks



Recurrent Neural Networks



Recurrent Neural Networks



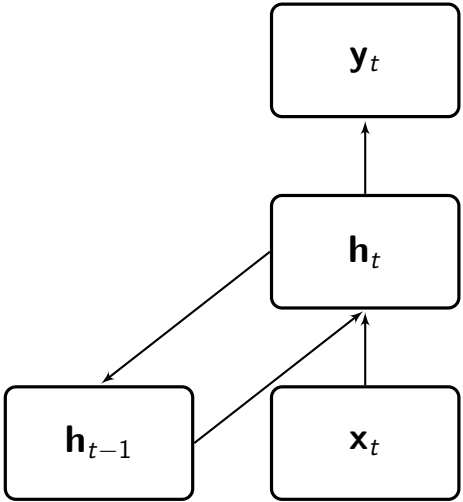


Figure: Recurrent Neural Network

$$\mathbf{h}_t = \phi((\mathbf{W}_{hh} \cdot \mathbf{h}_{t-1}) + (\mathbf{W}_{xh} \cdot \mathbf{x}_t))$$

$$\mathbf{y}_t = \phi(\mathbf{W}_{hy} \cdot \mathbf{h}_t)$$

NDR

Recurrent Neural Networks

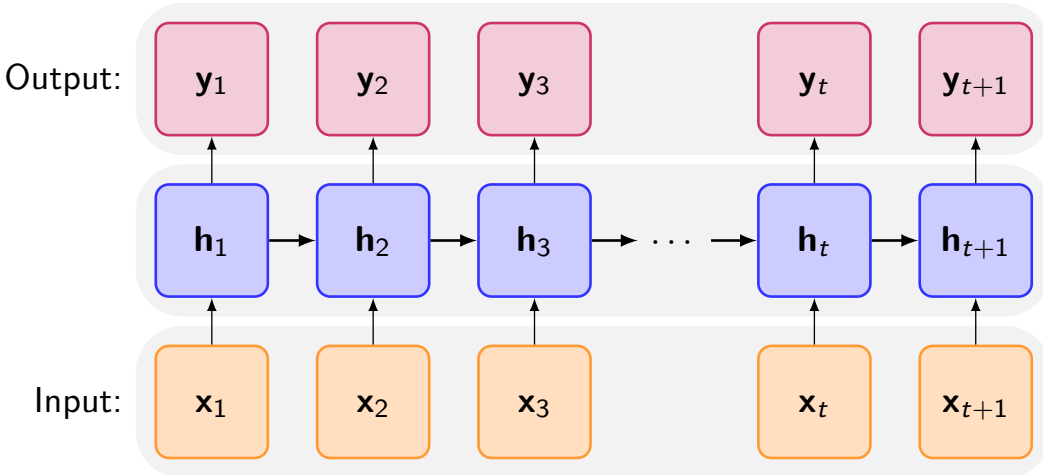
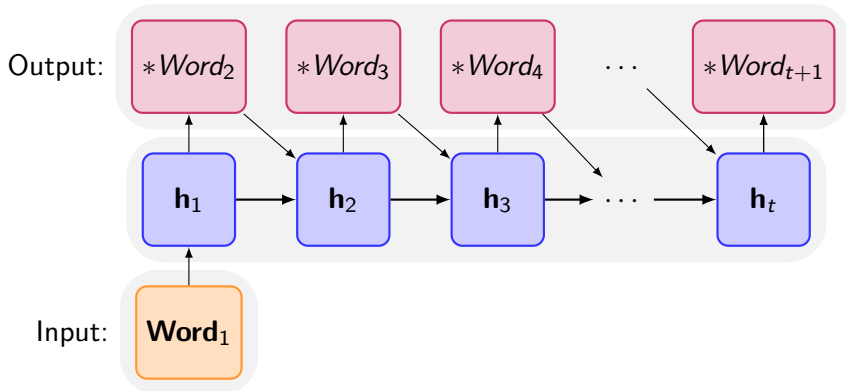


Figure: RNN Unrolled Through Time

Hallucinating Text



Hallucinating Shakespeare

PANDARUS: Alas, I think he shall be come approached and the day When little srain would be attain'd into being never fed, And who is but a chain and subjects of his death, I should not sleep.

Second Senator: They are away this miseries, produced upon my soul, Breaking and strongly should be buried, when I perish The earth and thoughts of many states.

DUKE VINCENTIO: Well, your wit is in the care of side and that.

From: http://karpathy.github.io/2015/05/21/rnn-effectiveness/

NDR

Neural Machine Translation

Neural Machine Translation

1. RNN Encoders
2. RNN Language Models

Encoders

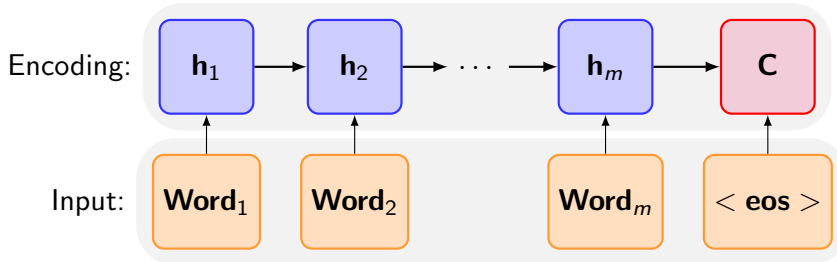


Figure: Using an RNN to Generate an Encoding of a Word Sequence

NDR

Language Models

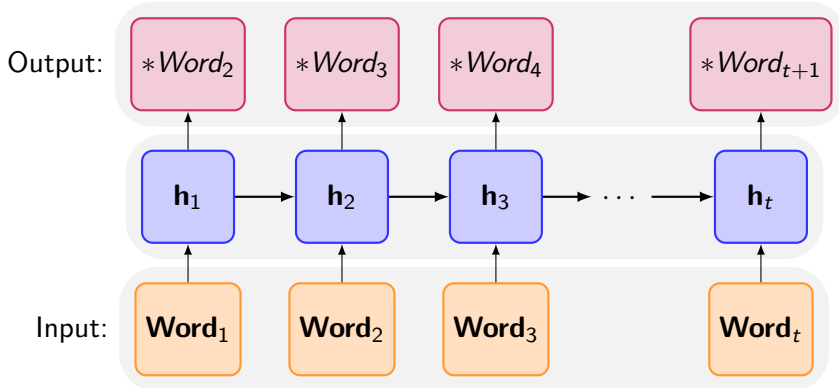


Figure: RNN Language Model Unrolled Through Time

NDR

Decoder

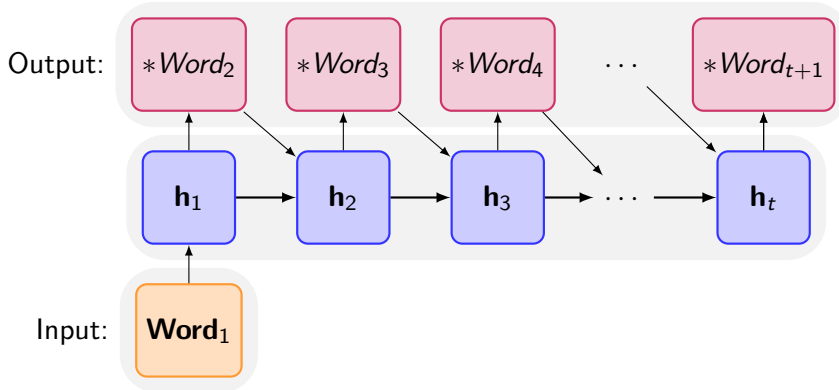


Figure: Using an RNN Language Model to Generate (Hallucinate) a Word Sequence

Encoder-Decoder Architecture

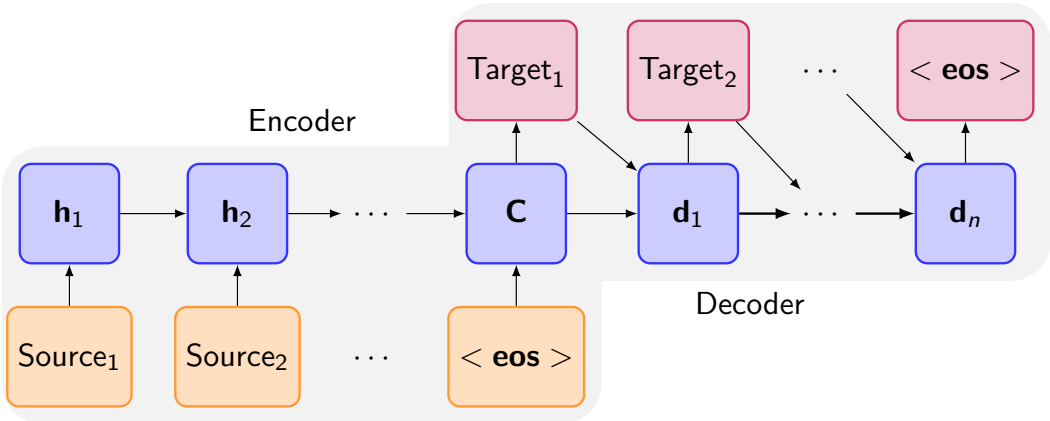


Figure: Sequence to Sequence Translation using an Encoder-Decoder Architecture

NDR

Neural Machine Translation

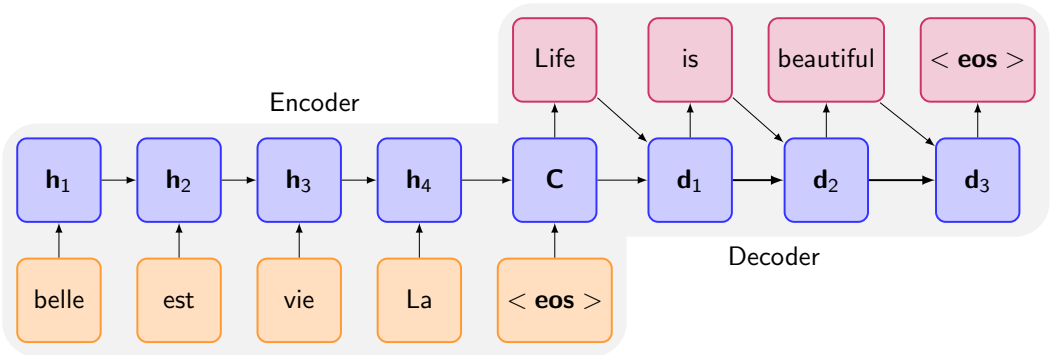


Figure: Example Translation using an Encoder-Decoder Architecture

NDR

Beyond NMT: Image Annotation

Image Annotation

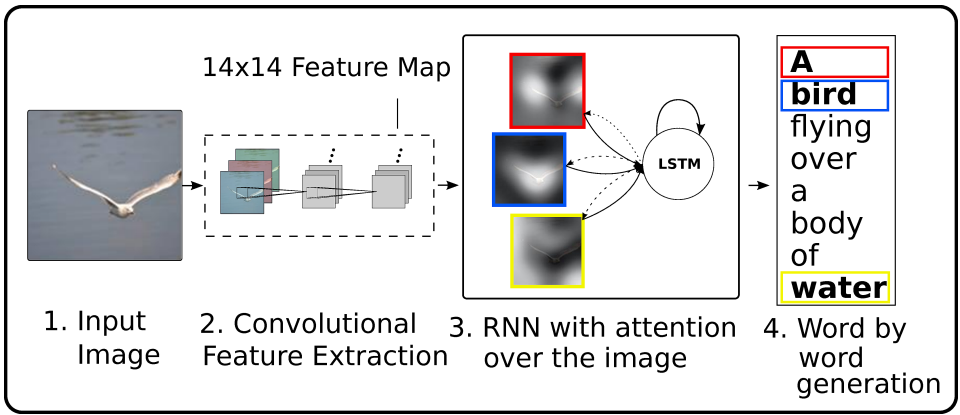
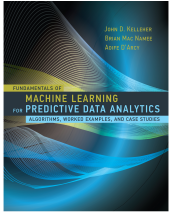


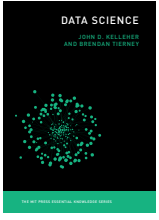
Image from Image from Show, Attend and Tell: Neural Image Caption Generation with Visual Attention Xu et al. (2015)

NDR

Thank you for your attention

john.d.kelleher@dit.ie
@johndkelleher
www.machinelearningbook.com
https://ie.linkedin.com/in/johndkelleher





Acknowledgements: The ADAPT Centre is funded under the SFI Research Centres Programme (Grant 13/RC/2106) and is co-funded under the European Regional Development Fund.

Books: Kelleher et al. (2015) Kelleher and Tierney (2018)

References I

Kelleher, J. (2016). Fundamentals of machine learning for neural machine translation. In *Translating Europe Forum 2016: Focusing on Translation Technologies*. The European Commission Directorate-General for Translation, European Commission doi: 10.21427/D78012.

Kelleher, J. D., Mac Namee, B., and D'Arcy, A. (2015). *Fundamentals of Machine Learning for Predictive Analytics: Algorithms, Worked Examples and Case Studies*. MIT Press, Cambridge, MA.

Kelleher, J. D. and Tierney, B. (2018). *Data Science*. MIT Press.

NDR

References II

Mikolov, T., Yih, W.-t., and Zweig, G. (2013). Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751.

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., and Bengio, Y. (2015). Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning*, pages 2048–2057.

NDR