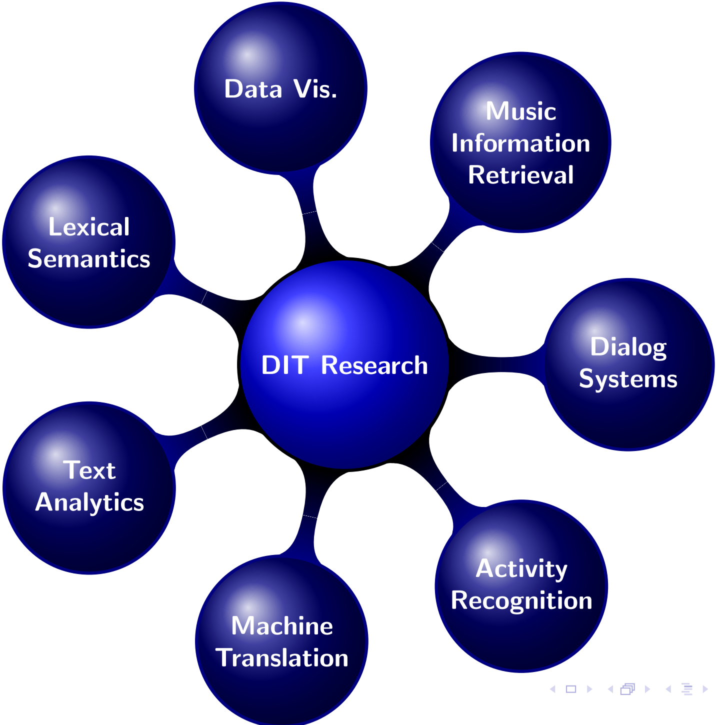
Attention Models and Spatial Representations in Encoder-Decoder Deep Learning Architectures

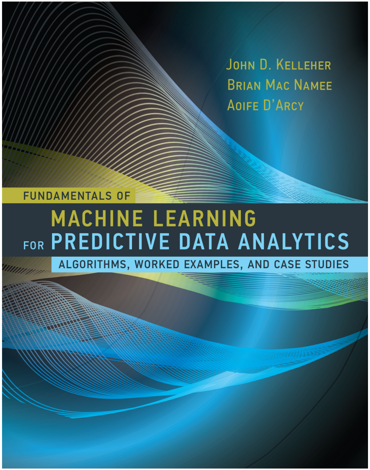
John D. Kelleher
The ADAPT Centre for Digital Content Technology
&
School of Computing, Dublin Institute of Technology

Centre for Linguistic Theory and Studies in Probability (CLASP)
University of Gothenburg
11^{th} March 2016

Outline


Data Vis.
Music Information Retrieval
Lexical Semantics
DIT Research
Dialog Systems
Text Analytics
Activity Recognition
Machine Translation

Fundamentals of Machine Learning for Predictive Data Analytics. Kelleher, Mac Namee, and D'Arcy. MIT Press



www.machinelearningbook.com

Robert Ross

Giancarlo Salton

- Three get online resources to learn more about deep-learning:
  1. Andrej Karpathy's blog available at:

     karpathy.github.io
  2. Christopher Olah blog (aka colah's blog) available at:

     http://colah.github.io
  3. Michael Nielson's online book "Neural Networks and Deep Learning" available at:

     neuralnetworksanddeeplearning.com/

What is Deep Learning?

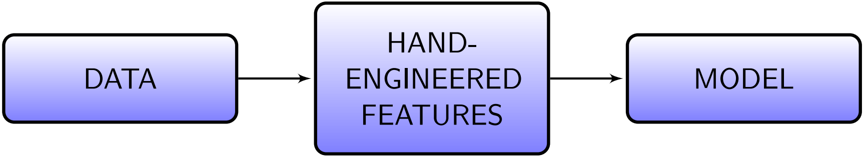
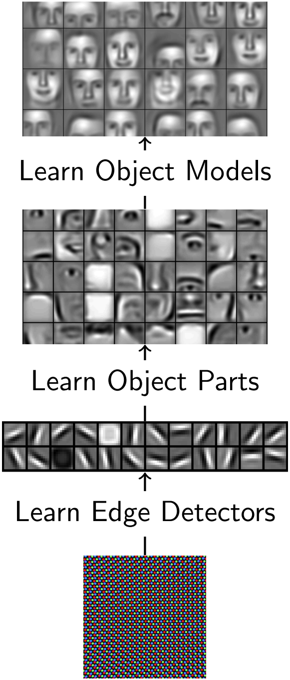


Figure: Standard ML

Figure: Deep Learning

1

[1] Inspired by slide from Kevin Duh (Deep Learning Tutorial)



Convolutional Deep Belief Networks
for Scalable Unsupervised Learning
of Hierarchical Representations,
Lee et al. In ICML 2009.

Deep Learning and Language

- Language is sequential and has lots of words.

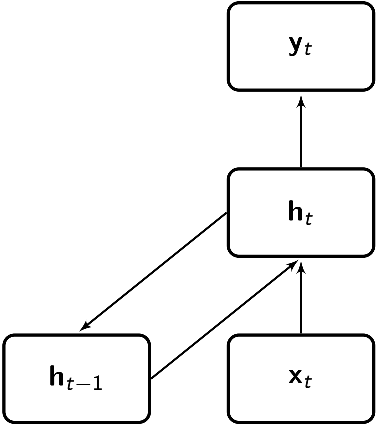


Figure: Recurrent Neural Network

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

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

- An RNN is as deep as your sentence is long.

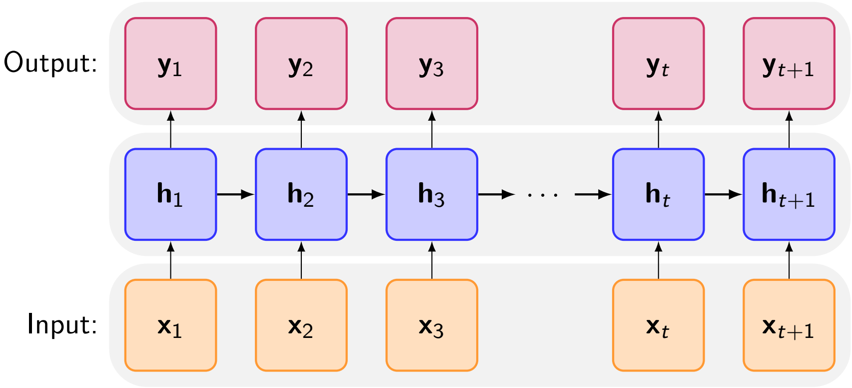


Figure: RNN unrolled through time.

- If you are at time t and you try to backpropogate to time k you will find that you derivatives become zero (vanishing gradient)[2]
- This is because you will have to do t-k multiplications[3]
- The implication of this is that the input at k will not influence the output at t
- If you need a long memory to learn you task a standard RNN won't work!

[2] or explode (exploding gradient)

[3] When we calculate the derivative of the error with respect to the transition parameters $\mathbf{W}_{hh}$ we need to apply the Chain Rule $\left(\frac{d}{dx} f\left(g\left(x\right)\right) = \frac{d}{d\ g(x)} f\left(g\left(x\right)\right) \times \frac{d}{dx} g\left(x\right)\right)$ to go back through the network k steps because h_t is dependent on h_{t-1} this results in W_{hh} being multiplied by itself many times

- In order for an RNN to have a long memory each cell in the network needs to learn:
    1. when to forget
    2. when to write something new to memory
    3. when to write something out
- LSTM cells do this by using a gating mechanism based on component wise multiplication

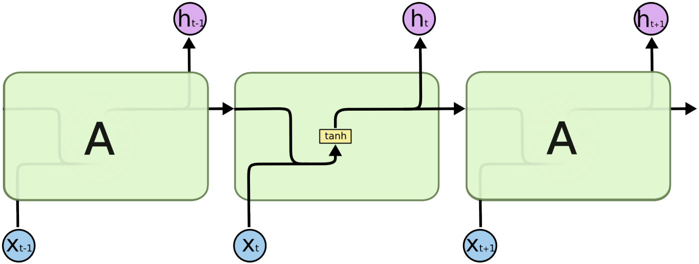


Figure: The repeating module in a standard RNN contains a single layer.

4

4 http://colah.github.io/posts/2015-08-Understanding-LSTMs/

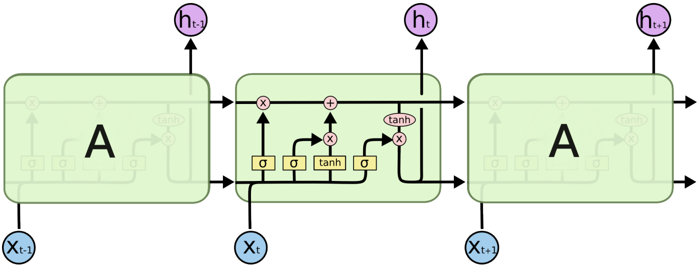


Figure: The repeating module in an LSTM contains four interacting layers.

5

[5] http://colah.github.io/posts/2015-08-Understanding-LSTMs/

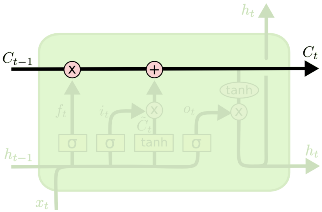
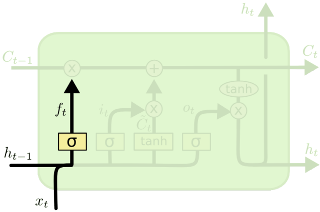


Figure: The cell state is kind of like a conveyor belt.

6

6 http://colah.github.io/posts/2015-08-Understanding-LSTMs/

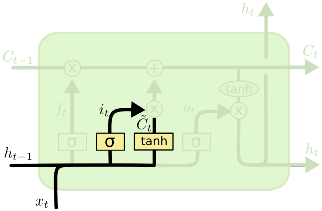


$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \ + \ b_f\right)$$

Figure: What information will we throw away from the cell state: the forget gate.

7

[7] http://colah.github.io/posts/2015-08-Understanding-LSTMs/

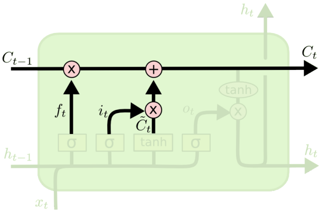


$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \ + \ b_C)$$

Figure: What information will we add to the cell state: the input gate and calculating a new vector C

8

8 http://colah.github.io/posts/2015-08-Understanding-LSTMs/

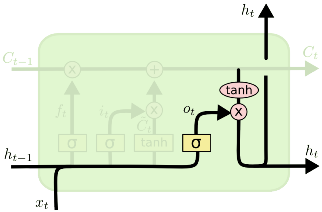


$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Figure: Update the cell state: applying our forget and input decisions

9

9 http://colah.github.io/posts/2015-08-Understanding-LSTMs/



$$o_t = \sigma\left(W_o\ [h_{t-1}, x_t]\ +\ b_o\right)$$

$$h_t = o_t * \tanh\left(C_t\right)$$

Figure: What should we output: a filtered version of the cell state

10

10 http://colah.github.io/posts/2015-08-Understanding-LSTMs/

- Language is sequential and has lots of words.

- One-hot (1-of-k)

$$cat = [0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0]$$

$$dog = [0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]$$

- One-hot (1-of-k)

$$cat = [0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0]$$

$$dog = [0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]$$

- Dimensionality is the size of the vocabulary
- Representation does not 'naturally' encode the semantic relationship between words

- Fortunately we use neural networks to learn low-dimensional word vectors (embeddings) directly from a corpus.

[11] See *inter alia.*: A Neural Probabilistic Language Model (Bengio et al., 2003); Natural Language Processing (Almost) from Scrath (Collobert et al, 2011); Efficient Estimation of Word Representations in Vector Space (Mikolov et al,. 2013), aka. word2vec (skip-gram and cbow); Glove: Global Vectors for Word Representation (Pennington et al., 2014)

- Fortunately we use neural networks to learn low-dimensional word vectors (embeddings) directly from a corpus.
- How?

[11] See *inter alia.*: A Neural Probabilistic Language Model (Bengio et al., 2003); Natural Language Processing (Almost) from Scrath (Collobert et al, 2011); Efficient Estimation of Word Representations in Vector Space (Mikolov et al,. 2013), aka. word2vec (skip-gram and cbow); Glove: Global Vectors for Word Representation (Pennington et al., 2014)

- Fortunately we use neural networks to learn low-dimensional word vectors (embeddings) directly from a corpus.
- How?
- Train the network to predict the word that is missing from the middle of an n-gram (or predict the n-gram from the word) and use the trained network weights to represent the word in vector space.[11]

[11] See *inter alia.*: A Neural Probabilistic Language Model (Bengio et al., 2003); Natural Language Processing (Almost) from Scrath (Collobert et al, 2011); Efficient Estimation of Word Representations in Vector Space (Mikolov et al,. 2013), aka. word2vec (skip-gram and cbow); Glove: Global Vectors for Word Representation (Pennington et al., 2014)

“a word is characteriezed by the company it keeps”

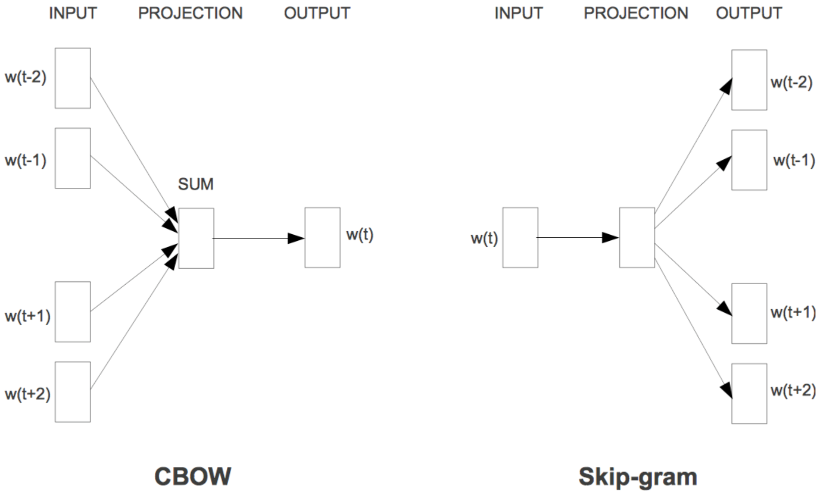
— Firth, 1957

“words which are similar in meaning occur in similar contexts’

— Rubenstein & Goodenough, 1965

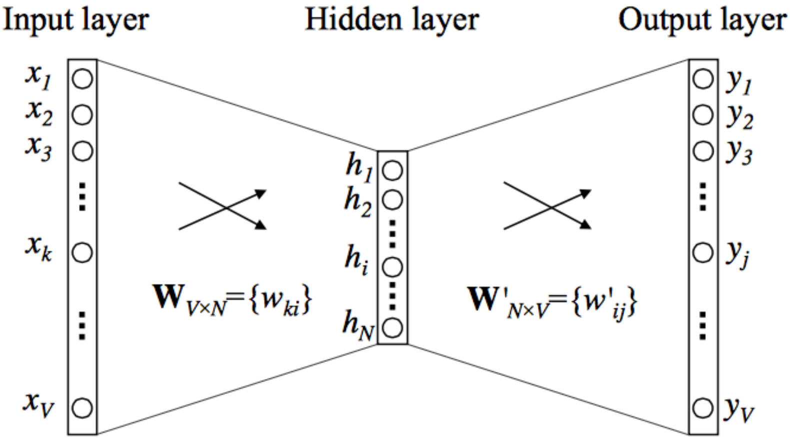
“a representation that captures much of how words are used in natural context will capture much of what we mean by meaning’

— Landauer & Dumais, 1997



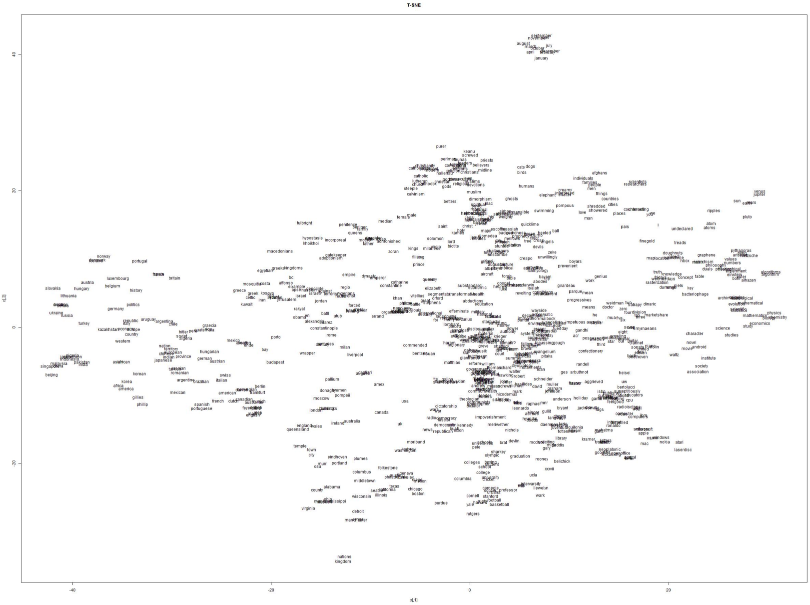
[12]

[12] Efficient Estimation of Word Representations in Vector Space (Mikolov et al., 2013)



[13]

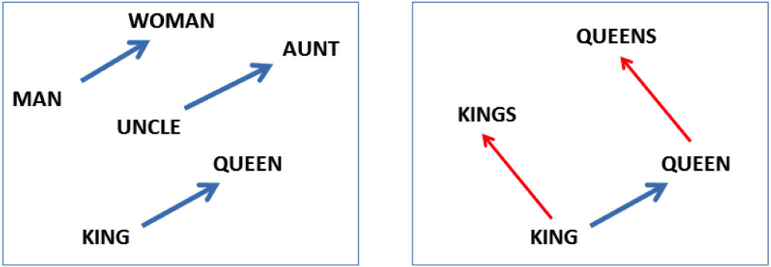
[13] http://www.folgertkarsdorp.nl/word2vec-an-introduction/



14

14 http://www.codeproject.com/Tips/788739/Visualization-of-High-Dimensional-Data-using-t-SNE

catholic
lutheran
steeple
calvinism
female
male
median
fulbright
penitence
family
queens
hypostasis
khoikhoi
incorporeal
father
admonished
kings
zoran
macedonians
adoptionism
prince
norway
portugal
egyptian
kingdoms
empire
emperor
spain
britain
austria
luxembourg
bc
affonso
catharine
constantine
belgium
mosquitia
costa
hungary
history
greece
regio
greek
celtic
iran
kosovo
jerusalem
israel
jordan
vitellius
khan
battle
politics
kuwait
germany
forced
napoleon
louis
rakyat
batll
utub
errand
obama
turkey
republic
italy
uruguay
argentina
chile
alexandria
suarez
graecia
kazakhstan
europe
country
hitler
peru
constantinople
algeria
rome
porto
western
mexico
centuries
commended
nation
territory
european
chinese
indian
province
hungarian
bay
milan
wrapper
liverpool
pakistan
india
asia
african
japanese
german
austrian
budapest
russian
romanian
korean
cagliari
swiss
pallium
argentine
brazilian
italian
korea
africa
america
berlin
amex
frankfurt
mexican
american
bremen
pompeii
moscow
gillies
canadian
usa
phillip
french
dutch
spanish
portuguese
london
canada
english
australia
ireland
uk
england
wales
queensland
moribund
temple
town
washington
city
eindhoven



$$vec(King) - vec(Man) + vec(Woman) \approx vec(Queen)$$ [15]

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

Language Models

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

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

  2. the probability for a sequence of words[16]

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

[16] 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

$$P(\textit{Yes I can help you}) > P(\textit{Help you I can yes})^{17}$$

  2. word choice

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

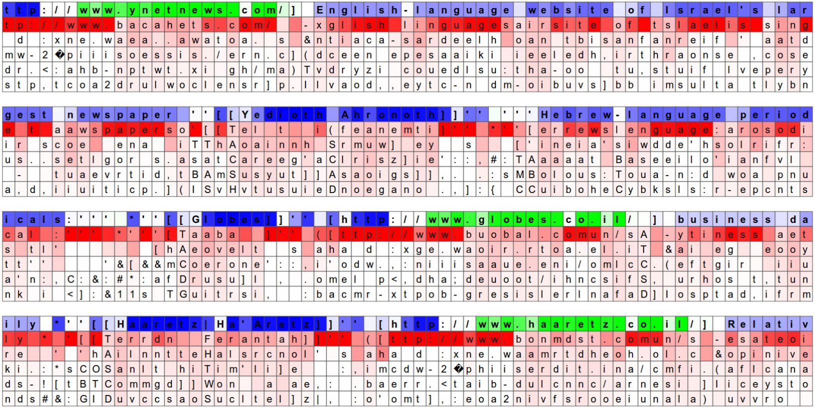
[17] Unless its Yoda that speaking

- How can RNN be trained for language modelling?[18]
  1. Step t_0:
     1.1 Initialise h_0
  2. Step t_1:
     2.1 Input first word w_1
     2.2 Calculate y_1, the probability distribution over the vocabulary for the next word w_2 given the first word w_1 and the context vector h_0[19]
     2.3 Error vector is computed using cross entropy between y_1 and a vector using 1-of-k encoding for the desired w_2[20]
     2.4 Weights updated with standard backprop.
  3. Step t_2:
     3.1 Input second word w_2
     3.2 ...

[18] For a more detailed explanation of training RNNs for language modelling see: Recurrent neural network based language model, Mikolov et al. 2010.

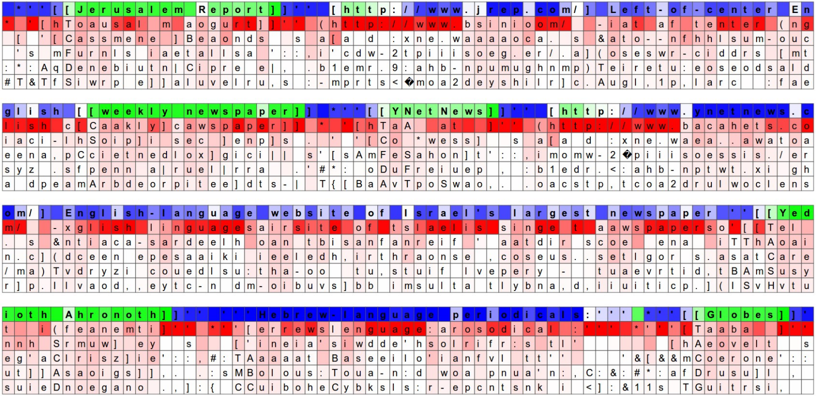
[19] Typically we use a Softmax to ensure that y_t is a valid probability distribution

[20] $H(p, q) = -\sum_x p(x) \; log \; q(x)$. See https://jamesmccaffrey.wordpress.com/2013/11/05/why-you-should-use-cross-entropy-error-instead-of-classification-error-or-mean-squared-error-for-neural for a nice discussion on why to use cross entropy



[21]

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

 22

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

Cell sensitive to position in line:

The sole importance of the crossing of the Berezina lies in the fact that it plainly and indubitably proved the fallacy of all the plans for cutting off the enemy's retreat and the soundness of the only possible line of action--the one Kutuzov and the general mass of the army demanded--namely, simply to follow the enemy up. The French crowd fled at a continually increasing speed and all its energy was directed to reaching its goal. It fled like a wounded animal and it was impossible to block its path. This was shown not so much by the arrangements it made for crossing as by what took place at the bridges. When the bridges broke down, unarmed soldiers, people from Moscow and women with children who were with the French transport, all--carried on by vis inertiae--pressed forward into boats and into the ice-covered water and did not, surrender.

Cell that turns on inside quotes:

"You mean to imply that I have nothing to eat out of.... On the contrary, I can supply you with everything even if you want to give dinner parties," warmly replied Chichagov, who tried by every word he spoke to prove his own rectitude and therefore imagined Kutuzov to be animated by the same desire.

Kutuzov, shrugging his shoulders, replied with his subtle penetrating smile: "I meant merely to say what I said."

Cell that robustly activates inside if statements:

```
static int __dequeue_signal(struct sigpending *pending, sigset_t *mask,
    siginfo_t *info)
{
  int sig = next_signal(pending, mask);
  if (sig) {
   if (current->notifier) {
    if (sigismember(current->notifier_mask, sig)) {
     if (!(current->notifier)(current->notifier_data)) {
      clear_thread_flag(TIF_SIGPENDING);
      return 0;
     }
    }
   }
   collect_signal(sig, pending, info);
  }
  return sig;
}
```

A large portion of cells are not easily interpretable. Here is a typical example:

```
/* Unpack a filter field's string representation from user-space
 * buffer. */
char *audit_unpack_string(void **bufp, size_t *remain, size_t len)
{
 char *str;
 if (!*bufp || (len == 0) || (len > *remain))
  return ERR_PTR(-EINVAL);
 /* Of the currently implemented string fields, PATH_MAX
  * defines the longest valid length.
  */
```

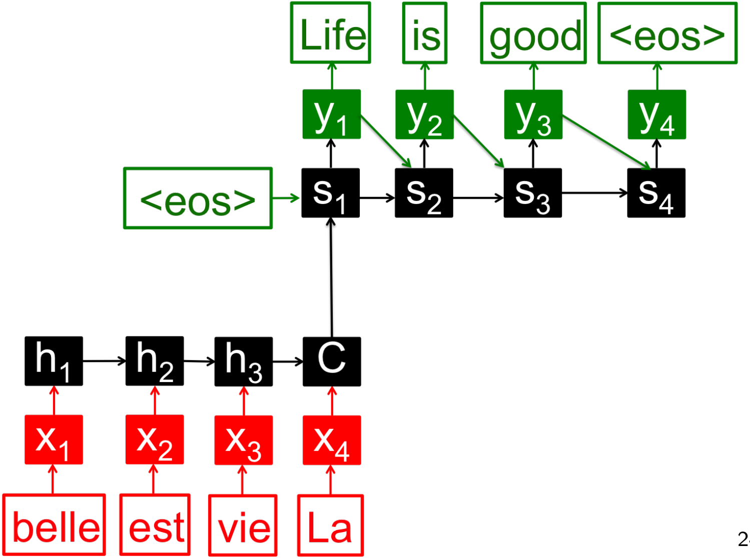
[23]

[23] http://karpathy.github.io/2015/05/21/rnn-effectiveness/

Machine Translation

- When we are translating a word in a source sentence the decision of what word to choose for the translation may be dependent on:
  1. the words that become before the word in the source sentence
  2. the words that we have already output in the target sentence
  3. and and the words that come after the word in the source sentence.

- When we are translating a word in a source sentence the decision of what word to choose for the translation may be dependent on:
  1. the words that become before the word in the source sentence
  2. the words that we have already output in the target sentence
  3. and and the words that come after the word in the source sentence.
- So, it makes sense to process the full source sentence before we start translating (that allows us to look ahead in the source during translation).



[24][25]

Figure: Encoder-Decoder Architecture

[24] For details see Sequence to Sequence Learning with Neural Networks (Sutskever et al. 2014)

[25] Note: the decoder in this architecture is a language model

- We want to minimise $\mathcal{J}_t$

$$\mathcal{J}_t = \sum_{(x,y)\in\mathcal{D}} -log\ p(y|x)$$

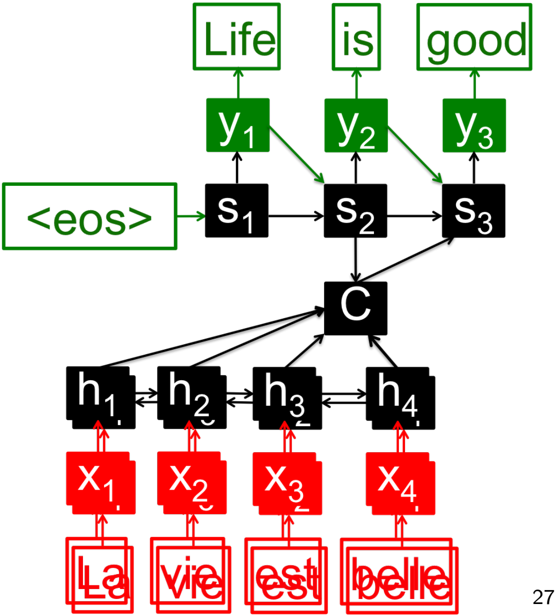
- where $\mathcal{D}$ is a parallel training corpus and the log probability of each sentence generated is calculated using: [26]

$$log\ p(y|x) = \sum_{j=1}^{m} log\ p(y_j|y_{<j}, x)$$

[26] For details see Effective Approaches to Attention-based Neural Machine Translation (Luong et al, 2015)

Global attention model

- add a neural network to the architecture that learns the weights for each word in the encoder at each time step in the decoder
- this network uses S_{t-1} as input and the output is used in the calculation of S_t

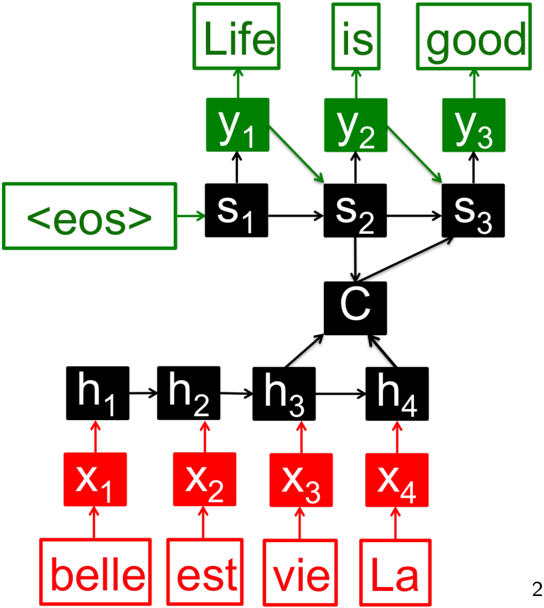


[27]

[27] For details see Neural machine translation by jointly learning to align and translate (Bahadanau et al. 2014). Note this architecture uses a global attention model, Gated Recurrent Units and bidirectional input.

Local Attention Model

- Idea: apply a normal distribution over global attention weights
- Define a window size (e.g., 10 words either side of a word) and let $sd = \frac{|window|}{2}$
- At each time step in the decoder
  1. calculate a global attention distribution
  2. a NN predicts pos. of the word in the input to center the window on, inputs include s_t and the length of the input sentence.
  3. Let
  $$x = \frac{(word\ offset)^2}{2 \times (sd)^2}$$
  4. Attention weights for words inside the window $=$ $e^{-x} \times \textit{global attention weight}$
  5. Attention weights for words outside the window $= 0$



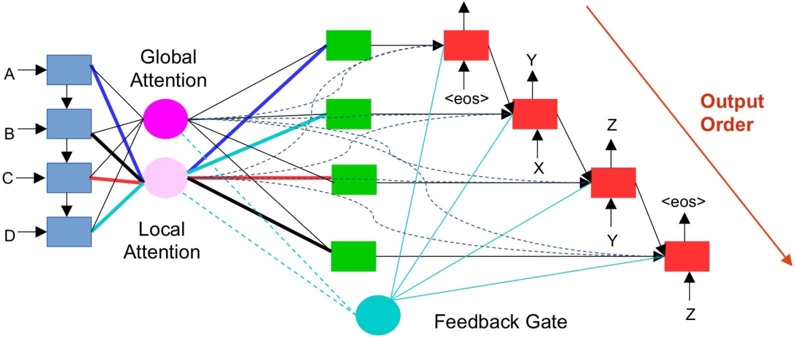
[28]

[28] For details see Effective Approaches to Attention-based Neural Machine Translation (Luong et al. 2015). Note this architecture uses a local attention model, LSTMs and reversed input.

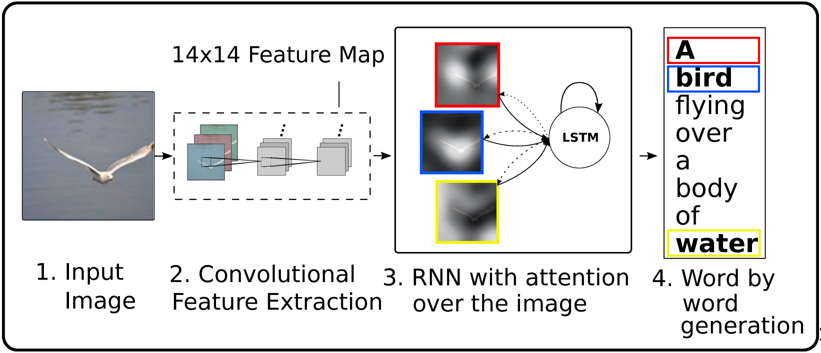
Handling Idioms

- Use both global and local attention
- Switch between the attentions when idiom is detected
- Intuition is that perplexity inside an idiom is low

Handling Idioms



Beyond MT: Image Annotation



[29]

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

- The standard system architecture in image captioning systems is to combine:
  1. a Convolutional Neural Network (used for image processing)
  2. with a Recurrent Neural Network (implementing a language model and used to generate the caption)

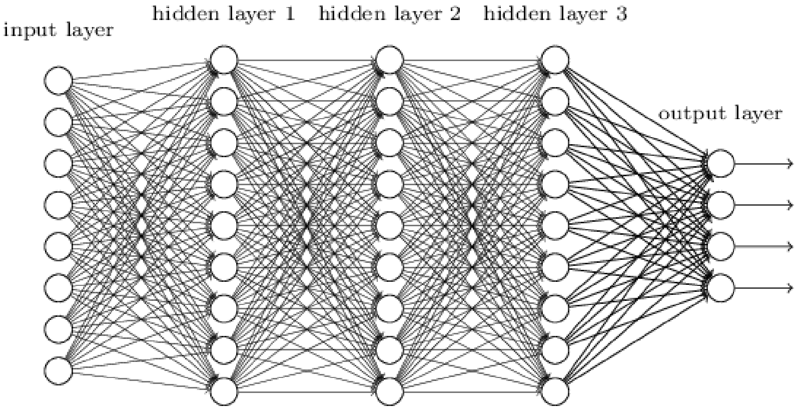


Figure: A fully connected feed forward neural network

30

30 Image taken from Neural Networks and Deep Learning by Michael Nielson available at neuralnetworksanddeeplearning.com/

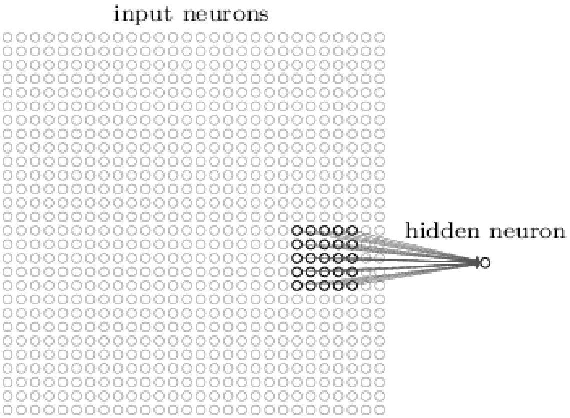


Figure: Illustration of a local receptive field

[31]

---

[31] Image taken from Neural Networks and Deep Learning by Michael Nielson available at `neuralnetworksanddeeplearning.com/`

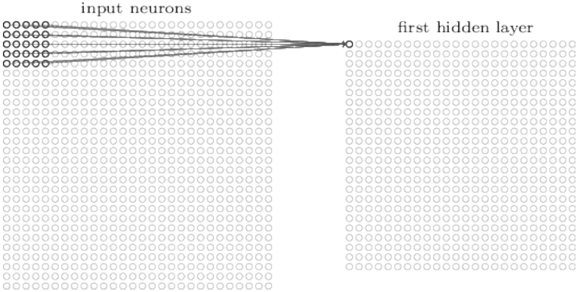


Figure: Local receptive field at Position 1

32

[32] Image taken from Neural Networks and Deep Learning by Michael Nielson available at neuralnetworksanddeeplearning.com/

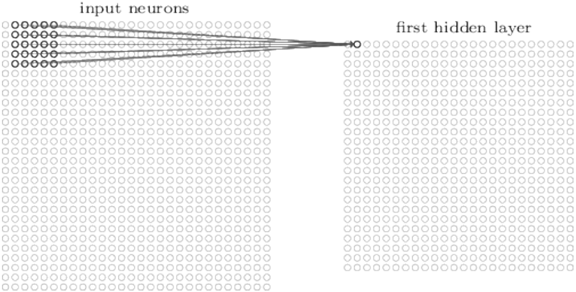


Figure: Local receptive field at Position 2

33

[33] Image taken from Neural Networks and Deep Learning by Michael Nielson available at neuralnetworksanddeeplearning.com/

- The neurons in the first hidden layer all share the same weights and bias
- In other words they all learn to react to a same feature (or pattern) in the input just at different locations in the image (i.e., each neuron monitors its own local receptive field for the feature).
- We use the term feature map to describe the map from the input layer to the hidden layer.

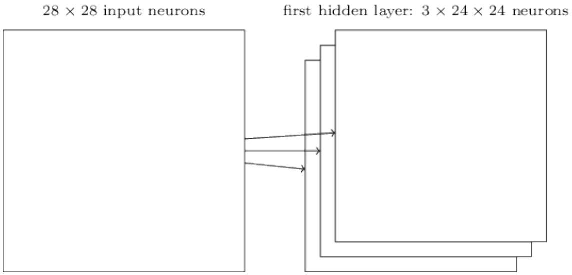


Figure: One Network multiple Feature Maps

34

[34] Image taken from Neural Networks and Deep Learning by Michael Nielson available at neuralnetworksanddeeplearning.com/

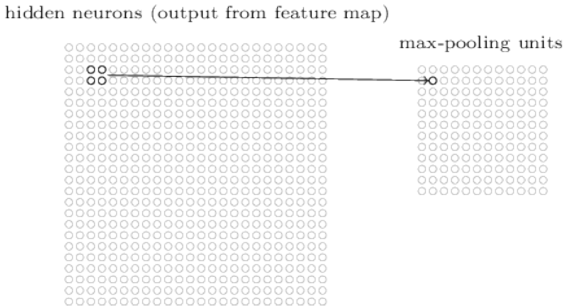


Figure: Pooling: discards exact positional information

35

[35] Image taken from Neural Networks and Deep Learning by Michael Nielson available at neuralnetworksanddeeplearning.com/

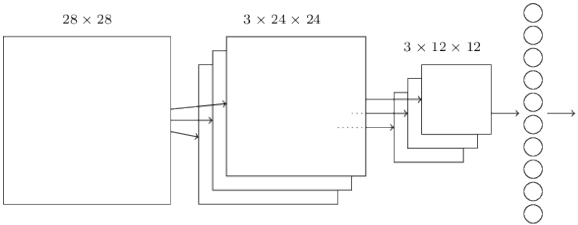
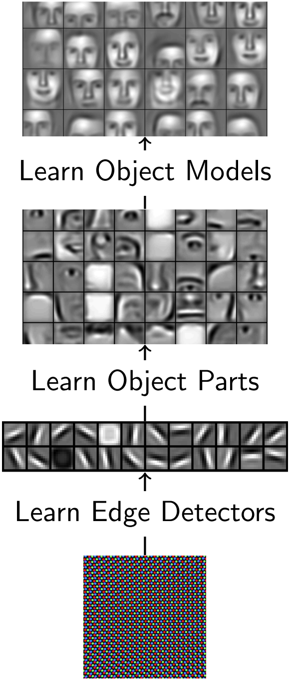


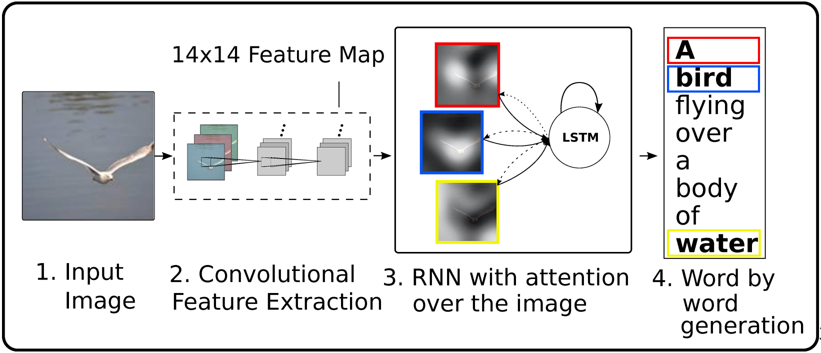
Figure: Complete Network with a Final Fully Connected Output Layer

36

36 Image taken from Neural Networks and Deep Learning by Michael Nielson available at neuralnetworksanddeeplearning.com/

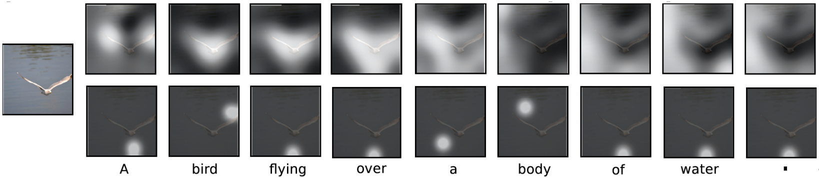


Convolutional Deep Belief Networks
for Scalable Unsupervised Learning
of Hierarchical Representations,
Lee et al. In ICML 2009.



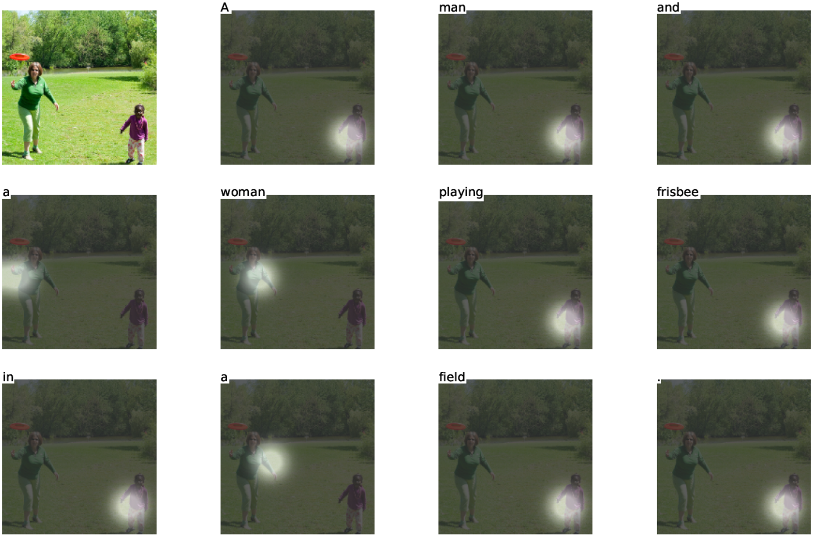
[37]

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



[38]

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



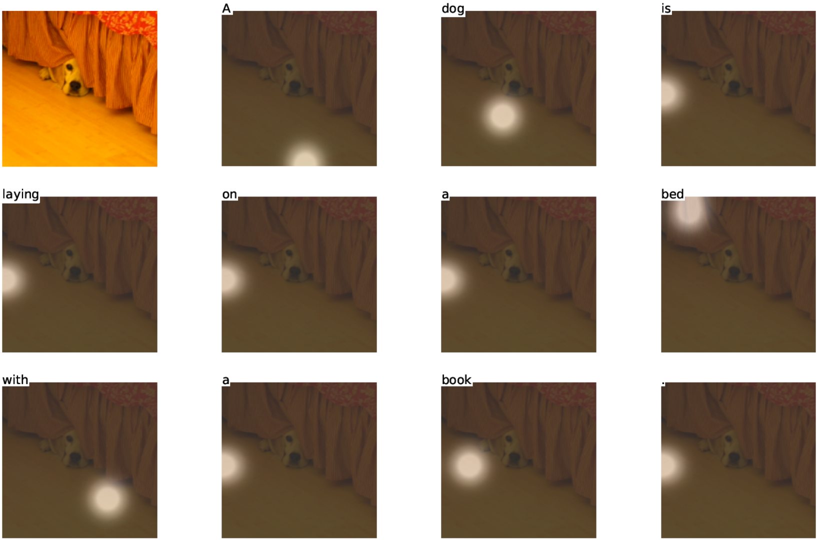
(a) A man and a woman playing frisbee in a field. 39

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

(b) A woman is throwing a frisbee in a park.

40

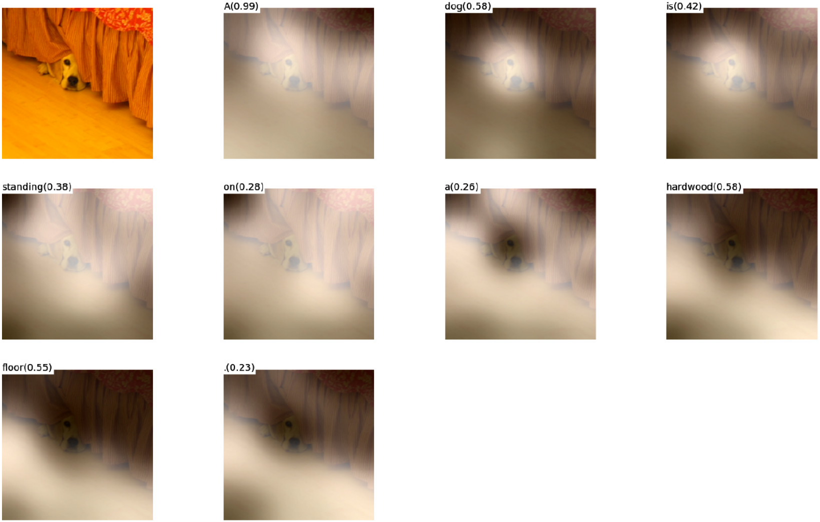
[40] Image from Show, Attend and Tell: Neural Image Caption Generation with Visual Attention (Xu et al. 2015).



(a) A dog is laying on a bed with a book.

41

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



(b) A dog is standing on a hardwood floor. [42]

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

Conclusions

- It may be possible for a CNN network to learn a vague spatial relationship between the outputs of different objects models (neurons in higher layers firing) but if this is what is happening then I believe the model is learning something like *man+left+women* and I don't believe the model will be able to generalise from this:

$$object1+left+object2 \neq object2+left+object1$$

- Although these systems generate spatial descriptions it is my contention that they do not have an explicit spatial representation and instead they are simply using the language model to predict what spatial term to use given the landmark and target object

- Although these systems generate spatial descriptions it is my contention that they do not have an explicit spatial representation and instead they are simply using the language model to predict what spatial term to use given the landmark and target object

Possible Implications

- Perspective 1: although DL seems to be making great strides in processing and integrating multimodal data at the moment DL architectures still struggle with spatial language
- Perspective 2: these systems seem to do fine on a lot of examples without any spatial representations, so when are representations necessary (maybe functional relationships are reflected in linguistic co-ocurrence patterns ...)

- One of the things I really like about deep learning is that it provides a natural way to learn multimodal representations.

- One of the things I really like about deep learning is that it provides a natural way to learn multimodal representations.
- However, we seemed to have moved from designed features to fitting hyper-parameters!
  - learning rate, mini-batch size, number of layers, number of units per layer, regularization constant, non-linearity, initialisation parameters, number of training epochs.

- One of the things I really like about deep learning is that it provides a natural way to learn multimodal representations.
- However, we seemed to have moved from designed features to fitting hyper-parameters!
  - learning rate, mini-batch size, number of layers, number of units per layer, regularization constant, non-linearity, initialisation parameters, number of training epochs.
- So, how to do deep learning in an eco-friendly way is the real challenge.

Thank you for your attention

`john.d.kelleher@dit.ie`

`@johndkelleher`

`www.comp.dit.ie/jkelleher`

`www.machinelearningbook.com`