RNN INTRO

+ TEXT SPECIFICALLY

martin @ reddragon.ai
sam @ reddragon.ai

16 October 2017

WiFi : SG-Guest

Problems with Installation? **ASK!**

RED DRAGON AI

PLAN OF ACTION

TODAY

- Start with CNN features wrap-up

- Text & Embeddings

- RNN Intro + Text

- Char-RNN

- RNN Project

RED DRAGON AI

PLAN OF ACTION

WEDNESDAY

- Public holiday

  - == Work on projects at home

- Actual session deferred to 'after' :

  - Expected : Monday 20-Nov-2017

RED DRAGON AI

PLAN OF ACTION
THURSDAY (OPTIONAL)

- PyTorch Deep Learning MeetUp Group

  - Hosted by Facebook

- Typical Contents :

  - ~~Talk for people starting out~~

  - Things from the bleeding-edge

  - Other talks (including 'lightning')

- MeetUp.com / PyTorch-and-Deep-Learning-Singapore

RED DRAGON AI

PLAN OF ACTION

MONDAY (!)

- RNN project wrap-up

- 'Advanced CNNs'

RED DRAGON AI

OUTLINE

- Recurrent Neural Networks
    - Basic idea & problems...
    - GRUs and LSTMs

- Natural Language Processing
    - Tokenization, etc
    - Word Embeddings

- Application : UPPER-CASE NER

PROCESSING SEQUENCES

- Methods so far have fixed input size

- But that is untrue in many domains :

  - Text data (sequences of words)

  - Text data (sequences of characters)

  - Dialog (eg: chatbots)

  - Audio speech data

  - Video clips / movies

- Need a technique that can be applied iteratively

RED DRAGON AI

PROCESSING SEQUENCES

VARIABLE-LENGTH INPUT DOESN'T "FIT"

- Run network for each timestep

  - ... with the same parameters

- But 'pass along' internal state

- This state is 'hidden depth'

  - ... and should learn features that are useful

RED DRAGON AI

RECURRENT NEURAL NETWORKS

- Apply one network at each step of the input

- Including an internal state that carries forward step-wise
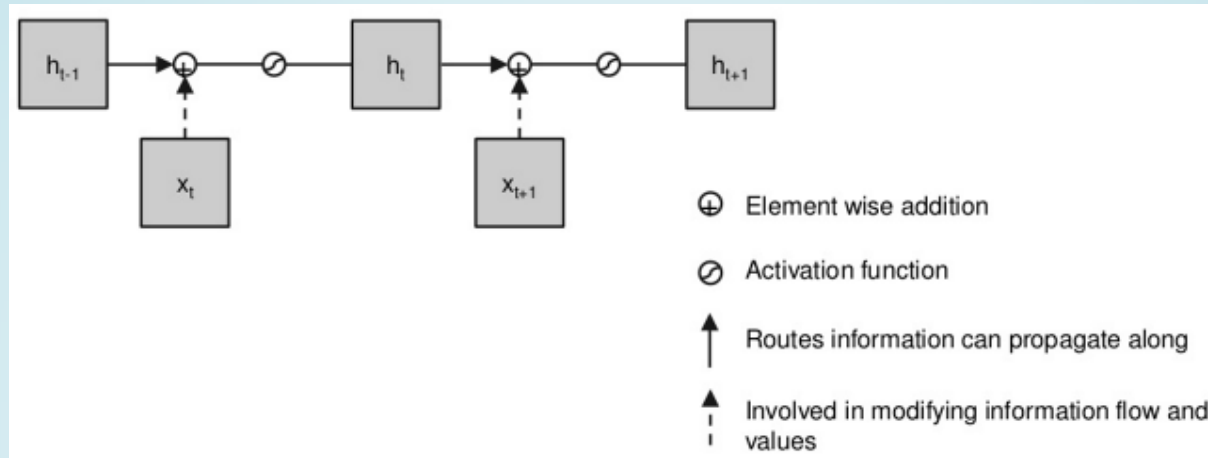
- Everything is still *differentiable*

RED DRAGON AI

BASIC RNN
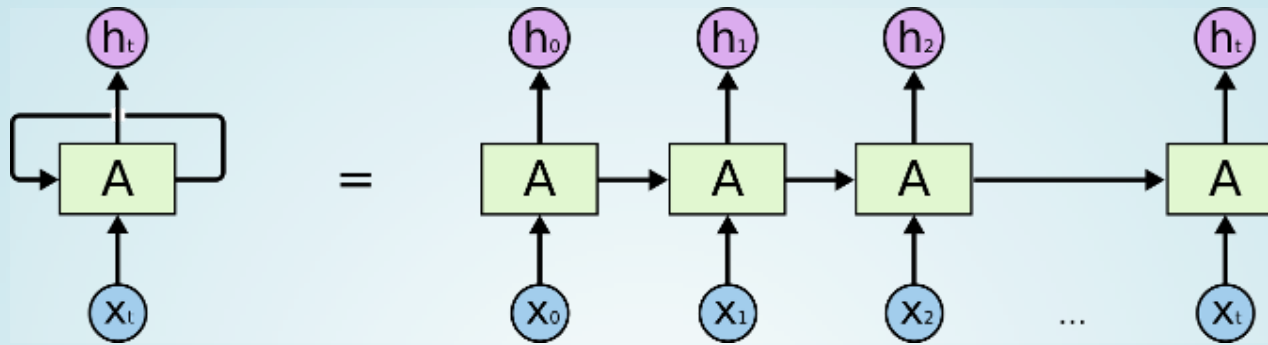
RNN chain

CHAINING STATE

- Each node 'knows' history

- ... all weights are 'tied'

- Network depth is time-wise

RED DRAGON AI

PLAIN RNN

Simplest RNN (but has gradient problem)

GRADIENT PROBLEM

- In a long sequence, early inputs are 'deep'

- Propagating errors must go through many layers

- Each layer multiplies by an additional factor

- So gradients can explode or vanish
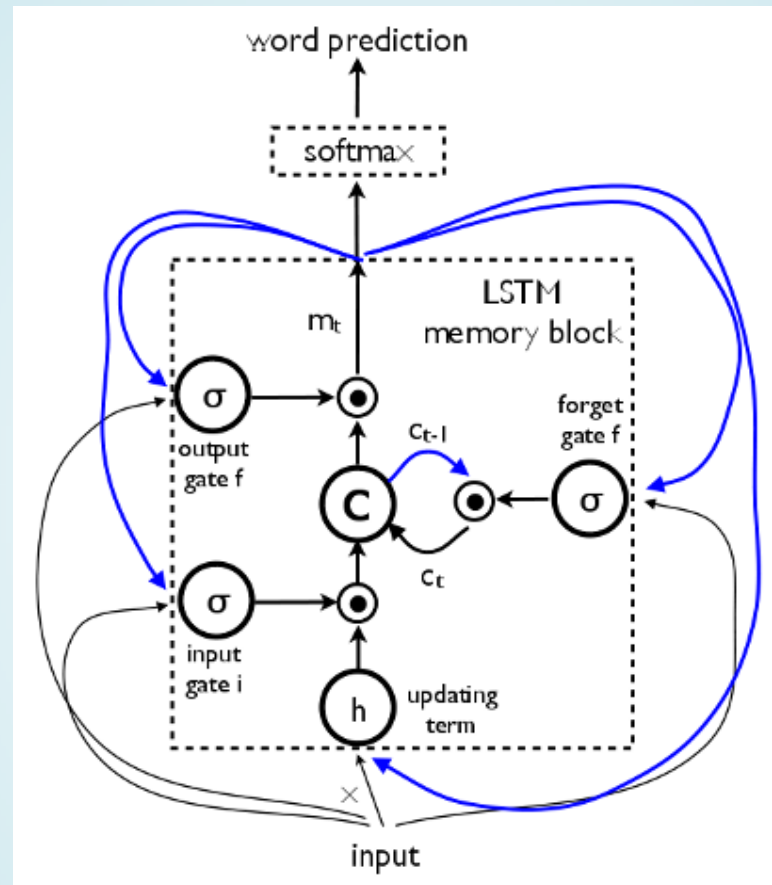
RED DRAGON AI

SOLUTION TO GRADIENT PROBLEM

- Instead of always multiplying by weights …

  - … have a straight-through path

  - … so there's a 'gradient 1.0' route

- Idea : Switch the routing on and off

RED DRAGON AI

GATED RECURRENT UNITS
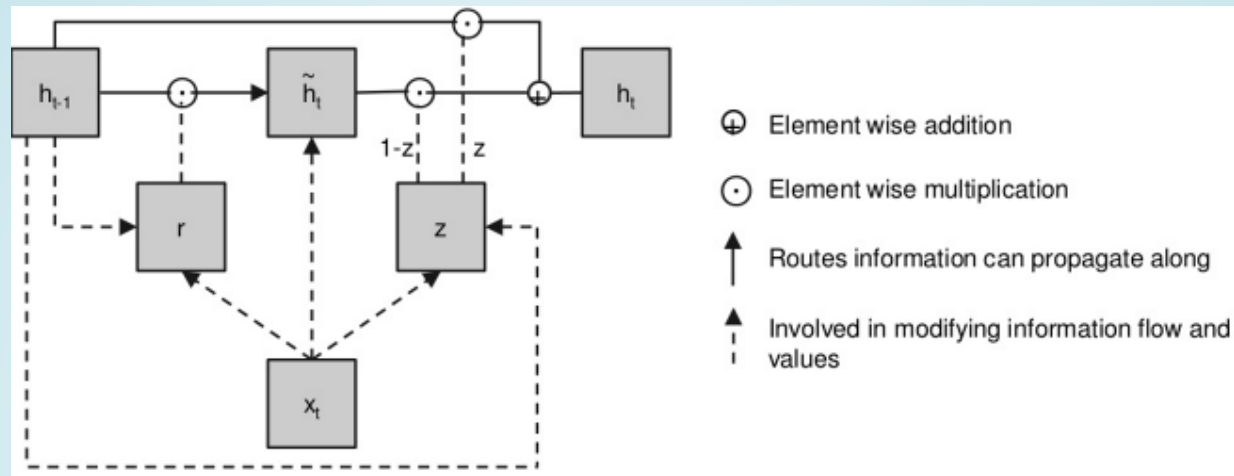
A GRU

LSTM UNITS

A Long Short-Term Memory (LSTM) Unit

RED DRAGON AI

PILED HIGHER AND DEEPER

- Can also pile up layers

- … and run forwards and backwards

RED DRAGON AI

KEY POINT

- Everything is still *differentiable*

- So : RNNs can be trained

- So : RNNs might be able to learn to do NLP

TEXT

- Documents

- Paragraphs

- Sentences

- Words

- Characters

PRE-PROCESSING

- Encoding

- Sentence-splitting

- Tokenization

- Vocabulary

- ... exceptions

ENCODING

- Can you open the file ?

- Are you 'unicode clean' ?

- Is 'e' the same as 'é' ?

- Is '»' the same as '"' or '"' ?

- Is '·' the same as '•' ?

- How about 'fi' vs 'fi' ?

RED DRAGON AI

SENTENCE SPLITTING

Mr. Sam Smith earned $2012.00
in the U.S.A. in 2012.

RED DRAGON AI

TOKENISATION

- Good idea to have a single standard

- Popular approach : Penn-TreeBank (PTB)

- Need to think again for :

  - Chinese (no word spaces)

  - Japanese (unicode punctuation)

  - ... (research is English-centric)

- `This is n't so easy .`

VOCABULARY BUILDING

- Suppose we have converted sentences to tokens

- Build a dictionary and convert tokens to indices

- Simple frequency analysis :
  - Stop-words (very frequent)
  - Common / normal words
  - Very rare words
  - Typos and Junk
  - <UNK>

RED DRAGON AI

UNDERSTANDING TEXT

- English ~100k+ words

- Bag-of-words

- Word Embeddings

RED DRAGON AI

BAG-OF-WORDS

- Just convert a sentence into set of words

    - (Throw away ordering)

- Simple statistical analysis (TF-IDF) :

    - Often very effective

- No idea that "jumps" ~ "jump"

- No idea that "jump" ~ "spring"

- No idea that "spring" ~ "summer"

WORD EMBEDDINGS

- Major advances : word2vec & GloVe

- Words that are close in the text should have close representations

- Assign a vector (~300d) to each word

  - Slide a 'window' over the text

  - Vectors for words in window are nudged together

- Keep iterating until 'good enough'

- (corpus size : 1 billion words +)

RED DRAGON AI

WORD EMBEDDING

- A map from `"token"` → `Float[100]`

- Train over corpus on windows of words

- The vector-space of words self-organizes...

FRANCE	JESUS	XBOX	REDDISH	SCRATCHED	MEGABITS
454	1973	6909	11724	29869	87025
AUSTRIA	GOD	AMIGA	GREENISH	NAILED	OCTETS
BELGIUM	SATI	PLAYSTATION	BLUISH	SMASHED	MB/S
GERMANY	CHRIST	MSX	PINKISH	PUNCHED	BIT/S
ITALY	SATAN	IPOD	PURPLISH	POPPED	BAUD
GREECE	KALI	SEGA	BROWNISH	CRIMPED	CARATS
SWEDEN	INDRA	PSNUMBER	GREYISH	SCRAPED	KBIT/S
NORWAY	VISHNU	HD	GRAYISH	SCREWED	MEGAHERTZ
EUROPE	ANANDA	DREAMCAST	WHITISH	SECTIONED	MEGAPIXELS
HUNGARY	PARVATI	GEFORCE	SILVERY	SLASHED	GBIT/S
SWITZERLAND	GRACE	CAPCOM	YELLOWISH	RIPPED	AMPERES

(eg: `word2vec` or `GloVe`)

RED DRAGON AI

EMBEDDING VISUALISATION
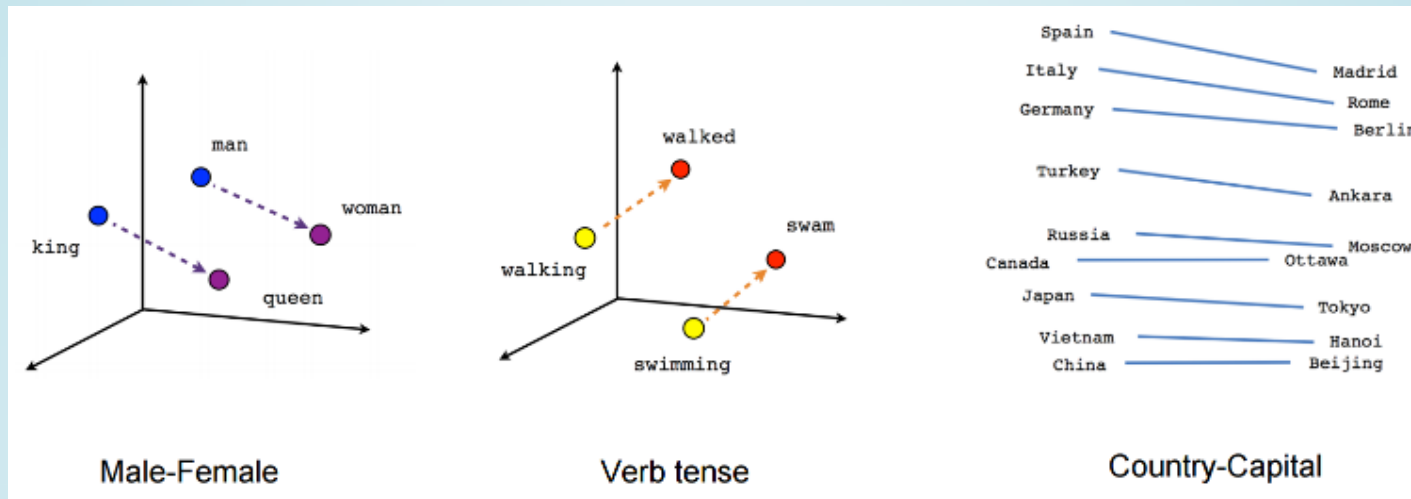
Highlighting Analogies

RED DRAGON AI

TEXT DEMO TIME

github.com/mdda/deep-learning-workshop/
notebooks/5-RNN/
5-Text-Corpus-and-Embeddings.ipynb

APPLICATION : RNNS FOR TEXT

- Building a quality NLP system

- Essential component : Named Entity Recognition (NER)

- Has to be flexible / trainable

- Has to understand regional quirks

RED DRAGON AI

NER : QUICK EXAMPLE

- Transform :

    - Soon after his graduation , Jim Soon became Managing Director of Lam Soon .

- Into :

    - Soon after his graduation , Jim_Soon^{PER} became Managing Director of Lam_Soon^{ORG} .

LEARNING NAMED ENTITY RECOGNITION

- Can we train an RNN to do 'NER'?

- Steps :

  - Create / select a word embedding

  - Get a NER annotated dataset

  - Train an RNN on the dataset
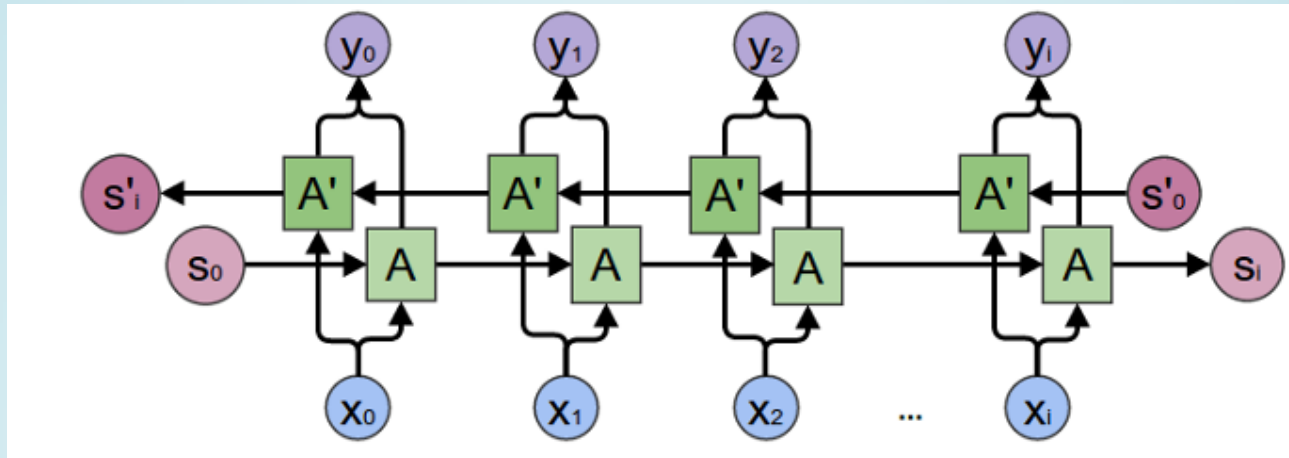
RED DRAGON AI

DEMO DATASET

- Human annotated Corpora are difficult to distribute :

    - Use NLTK to annotate Wikipedia

    - Train RNN on machine annotations

    - Look at performance vs NLTK

RED DRAGON AI

LET'S MAKE IT "INTERESTING"

- **Twist : Restrict RNN input to single case text**

RED DRAGON AI

NETWORK PICTURE

Bidirectional GRU RNN

RNN-TAGGER-KERAS

github.com/mdda/deep-learning-workshop/
notebooks/5-RNN/
6-RNN-Tagger-keras.ipynb

WRAP-UP

- Text processing is messy

- Word Embeddings are magic

- RNNs can be applied to lots of things

- Having a GPU is VERY helpful

- QUESTIONS -

MARTIN.ANDREWS @ REDDRAGON.AI

My blog : http://mdda.net/

GitHub : mdda

RED DRAGON AI