Movie Reviews Sentiment Analysis

COMP-599: Final Project Presentation
Dec 12, 2016

~Nicolas Angelard-Gontier - 260532513

Outline

- Dataset introduction
- Related work
- Models
  - Random Forest (Bag-of-Words / Word-to-Vec)
  - LSTM Recurrent Neural Network
- Experiments & Results
- Discussion

Movie Reviews Dataset

- Work inspired by two kaggle competitions:

IMDB	**Rotten Tomatoes**
25k (+50k unlabeled) training examples [12.5k | 12.5k]	156,060 training examples [7,072 | 27,272 | 79,556 | 32,927 | 9,206]
25,000 test examples	66,292 test examples
Binary classification (0, 1)	Multi-class classification (0, 1, 2, 3, 4)

Movie Reviews Dataset

- Work inspired by two kaggle competitions:

IMDB	**Rotten Tomatoes**
25k (+50k unlabeled) training examples [12.5k | 12.5k]	156,060 training examples [7,072 | 27,272 | 79,556 | 32,927 | 9,206]
25,000 test examples	66,292 test examples
Binary classification (0, 1)	Multi-class classification (0, 1, 2, 3, 4)

31,874 training examples [15,771 | 16,102]	181,060 training examples [19,572 | 27,272 | 79,556 | 32,927 | 21,706]

Movie Reviews Dataset

- Work inspired by two kaggle competitions:

IMDB	**Rotten Tomatoes**
25k (+50k unlabeled) training examples [12.5k | 12.5k]	156,060 training examples [7,072 | 27,272 | 79,556 | 32,927 | 9,206]
25,000 test examples	66,292 test examples
Binary classification (0, 1)	Multi-class classification (0, 1, 2, 3, 4)

31,874 training examples [15,771 | 16,102]	181,060 training examples [19,572 | 27,272 | 79,556 | 32,927 | 21,706]

Ex:

"The first time you see The Second Renaissance it may look boring. Look at it at least twice and definitely watch part 2. It will change your view of the matrix. Are the human people the ones who started the war ? Is AI a bad thing ?" (1)	good for the goose (3) -- good (3) -- for the goose (2)
“I would love to have that two hours of my life back. It seemed to be several clips from Steve's Animal Planet series that was spliced into a loosely constructed script. Don't Go, If you must see it, wait for the video ...” (0)	A series of escapades demonstrating the adage that what is good for the goose is also good for the gander , some of which occasionally amuses but none of which amounts to much of a story . (1)

Related work

- As we saw during the last few class lecture, a lot of algorithms are proposed in the literature for NLP problems.
- Interesting to investigate because: better understanding of natural langage, better movie recommendations
- This is just an overview of 2 methods:
  - (pretty good) baseline algorithm: Random Forest
  - More complex classification algorithm: LSTM RNN
- Unfortunate spoiler alert:
  - Combining both datasets didn't yield surprising results.
  - No ground breaking super efficient algorithm for this task will be presented.

Models & Features

Bag-of-Words

- Learns a vocabulary from all reviews.
  Models each review by counting the number of times each word appears.

- Example:
  R1 = “this movie was good”
  R2 = “this one was bad bad bad”

  Vocab: {“this”, “movie”, “was”, “good”, “one”, “bad”}

  R1 = [1, 1, 1, 1, 0, 0]
  R2 = [1, 0, 1, 0, 1, 3]

  => Ri[k] = count(word_k in Ri) <=

- Limit Vocab size to 5k, 10k top words (no stop words).

Word2Vec

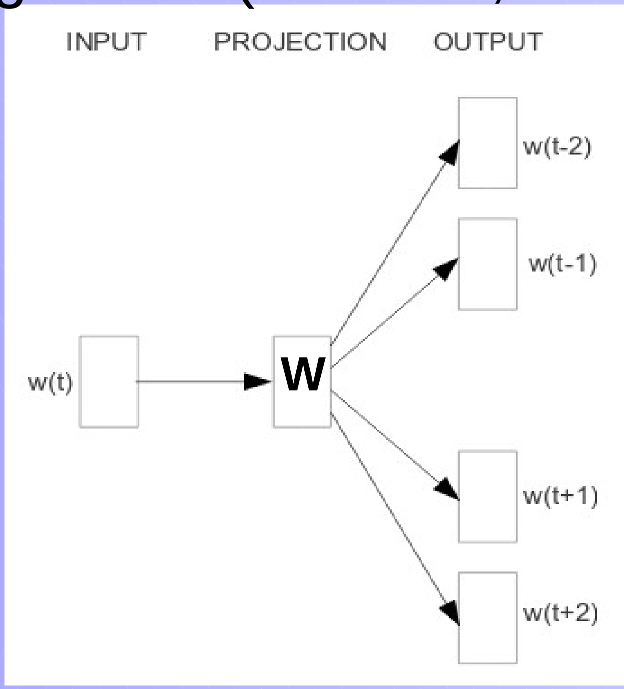
- Learns distributed representations for words.
  Can use unlabeled data (+50k reviews from IMDB).

  A word embedding is a parametrized function (W) mapping words to high-dimensional vectors (n=300 or 500 dimensions)
  ex: “cat” -> [0.3, -0.67, ..., 0.19] $W : words \rightarrow \mathbb{R}^n$

- Skip-Gram architechture: predicts surrounding words (5 before, 5 after in our case) given the current word.
  The learned parameter IS the embedding.

- Transfer Learning: learn a representation (W) on task A (skip-gram), and use it on task B (sentiment analysis, ...)

- From word to reviews?
  - Average word embeddings



Mikolov et. al., 2013

Random Forest

Large collection of decision trees (m=100, 500) -> ensemble method.

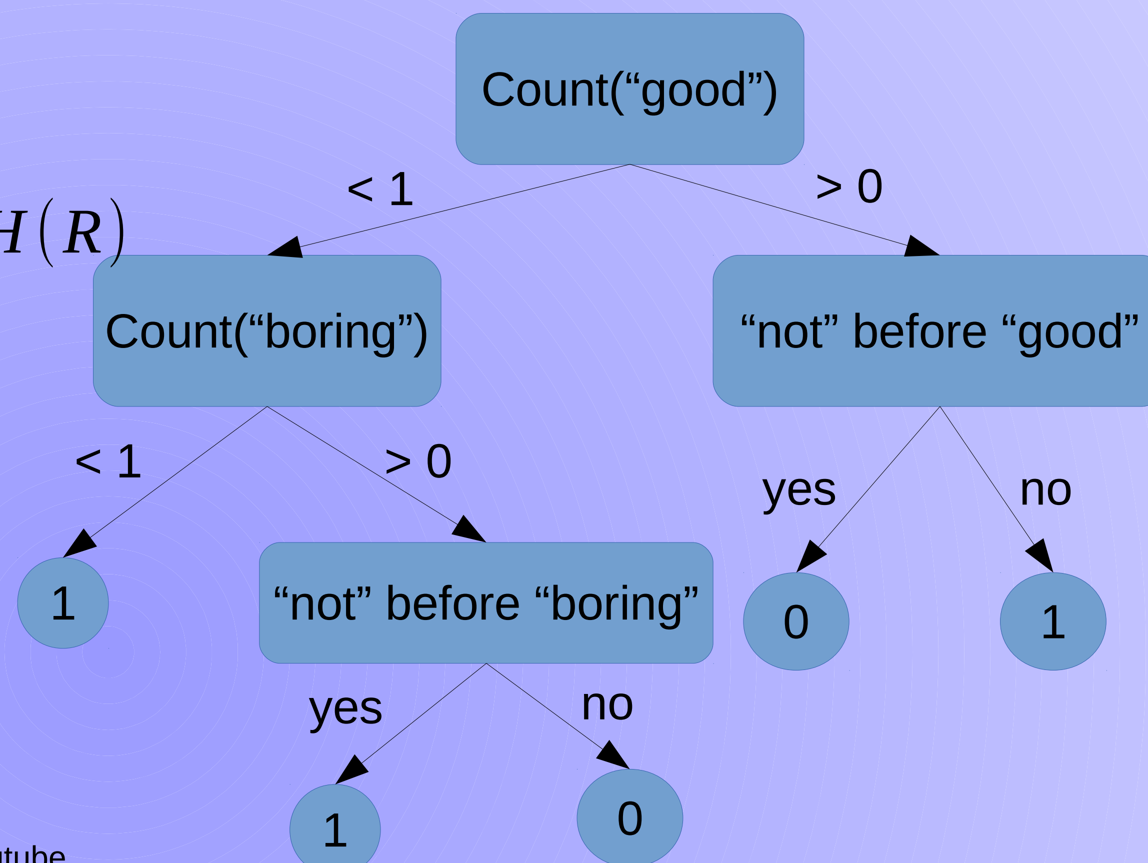
Each decision tree is trained on multiple **random** subsets of reviews.

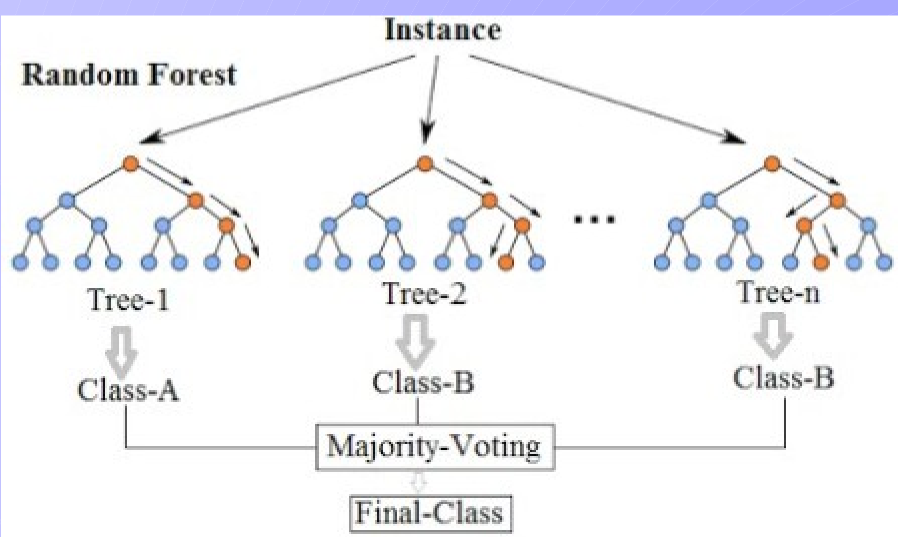
D = {R1, R2, ..., Rn} => {S1, S2, ..., Sm} where each S = {Ri, ..., Rj}
Repeat the above multiple times.

Tree features selected according to “information gained”:
IG(T1) = H(D) – H(D | T1)

$$H(D) = -\sum_{i \in D} p_i \log p_i$$

$$H(D|T_1) = W_L * H(L) + W_R * H(R)$$



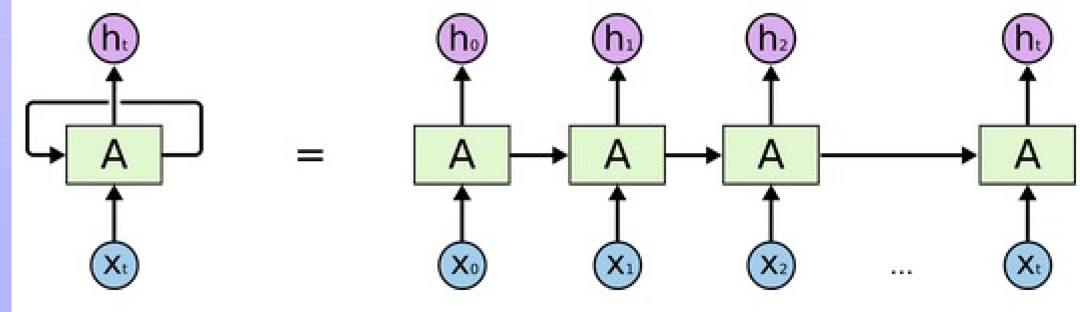


Random Forest based Classification – Be Expert In Minutes@Youtube

LSTM Recurent Neural Network

- RNN idea: use previous events to inform later ones. Allowing information to persist.

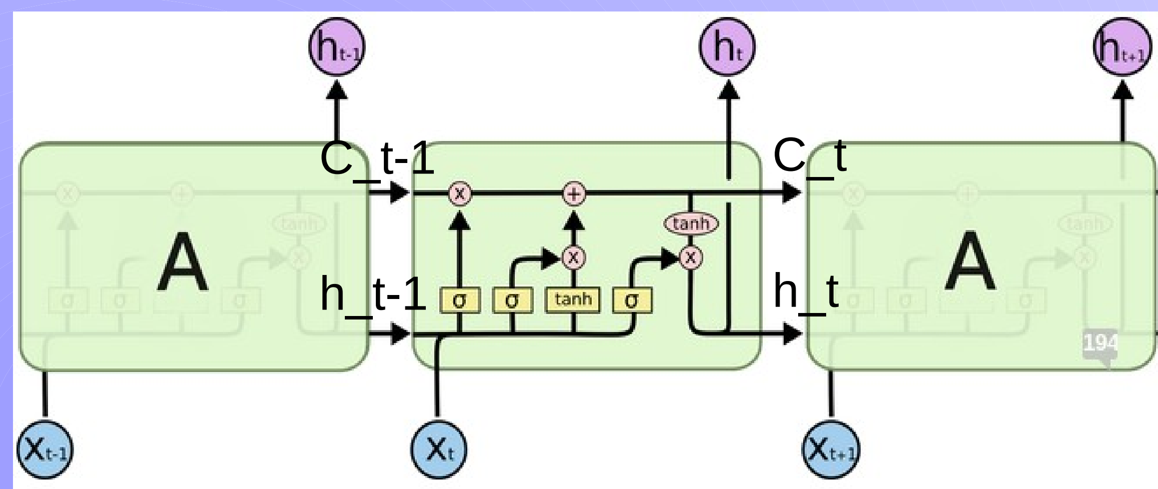
- Issue: long-term dep. are hard to capture in original RNN.
  h(t) = tanh{ h(t-1), x(t) }



An unrolled recurrent neural network.

Colah's blog: http://colah.github.io/posts/2015-08-Understanding-LSTMs/

- Solution: use LSTM units. h(t) = LSTM{ h(t-1), x(t) }
  - forget gate: what to forget
    $$f_t = \sigma(W_f \cdot x_t + U_f \cdot h_{t-1} + b_f)$$
  - input gate: what to update
    $$i_t = \sigma(W_i \cdot x_t + U_i \cdot h_{t-1} + b_i)$$
  - cell state: what to write
    $$C_t = \tanh(W_c \cdot x_t + U_c \cdot h_{t-1} + b_c)$$
  - output gate: what to output
    $$o_t = \sigma(W_o \cdot x_t + U_o \cdot h_{t-1} + b_o)$$
    $$h_t = o_t * \tanh(C_t)$$

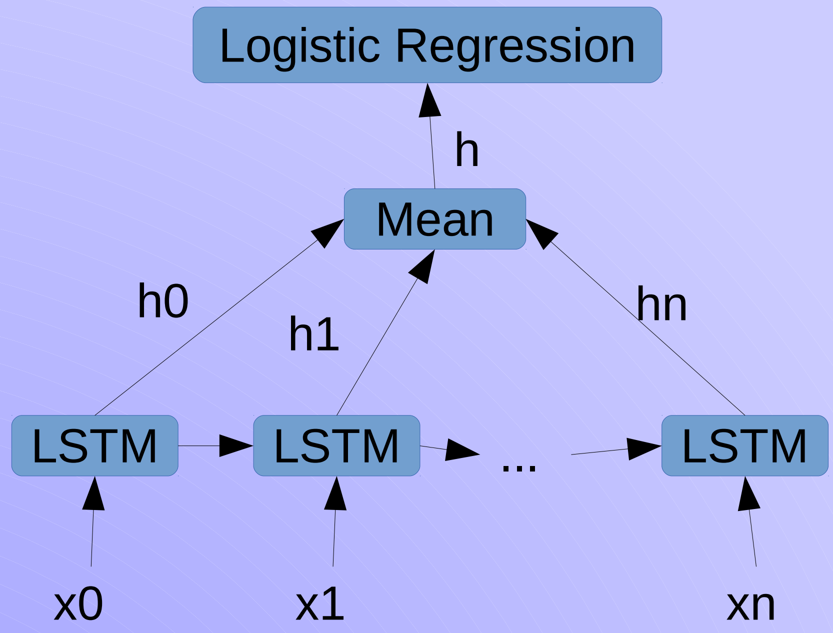


The repeating module in an LSTM contains four interacting layers.

Colah's blog: http://colah.github.io/posts/2015-08-Understanding-LSTMs/

LSTM Recurent Neural Network

- The model:
  - x's are word embeddings being learned by the model (size 300)
  - h_i's are new representations of the review at each time step
  - h is the final review representation
  - Prediction proba = softmax (U.h + b)
  - Predicted class = argmax(probas)



- Loss function to minimize:
  - cost = average negative log probability of the target class

$$-\frac{1}{N}\sum_{i=1}^{N}\log\left(p_{target\,class}+\epsilon\right)$$

- Trained with ADAM optimizer: combines advantages of AdaGrad & RMSProp -> both better versions of SGD.
(Diederik Kingma, Jimmy Ba, 2014, https://arxiv.org/abs/1412.6980)

Experiments & Results

IMDB submissions

Random Forest (BoW)			
Forest Size	Number of Features	Extended Data	Test set score
500	bow=10k	original	0.85904
		extended	**0.86088**
	bow=5k	original	0.85684
		extended	0.85556
100	bow=10k	original	0.84892
		extended	0.85068
	bow=5k	original	0.84316
		extended	0.84728
Random Forest (W2V)			
500	w2v=500	original	0.83492
		extended	**0.83704**
	w2v=300	original	0.83492
		extended	0.83308
100	w2v=500	original	0.83280
		extended	0.83332
	w2v=300	original	0.83100
		extended	0.82936

LSTM RNN – embedding size = 300	
original	**0.87379**
extended	0.87376
LSTM RNN – embedding size = 500	
original	0.83504
extended	0.86336

Best submission:
0.99259 accuracy

Rotten Tomatoes submissions

Random Forest (BoW)			
Forest Size	Number of Features	Extended Data	Test set score
500	bow=10k	original	0.58497
		extended	**0.59054**
	bow=5k	original	0.58411
		extended	0.58120
100	bow=10k	original	0.58903
		extended	0.58390
	bow=5k	original	0.58371
		extended	0.58090
Random Forest (W2V)			
500	w2v=500	original	0.57091
		extended	**0.59509**
	w2v=300	original	0.56974
		extended	0.59437
100	w2v=500	original	0.56678
		extended	0.59280
	w2v=300	original	0.56607
		extended	0.59018

LSTM RNN – embedding size = 300	
original	**0.61193**
extended	0.59117
LSTM RNN – embedding size = 500	
original	0.61066
extended	0.59858

Best submission:
0.76527 accuracy

Discussion

- IMDB task much easier:
  - Binary vs Multi-class classification
  - Full long reviews vs many small parts (no previous info available)
- Expected a bigger improvement with LSTM RNN. Extensions:
  - Stacking multiple LSTM layers?
  - Tree structured LSTM?
- Extending one data with the other is only slightly better for Random Forest.

Thank you.

Questions?