Recurrent Neural Networks - Long Short Term Memory Networks

Group Number • 12
Varsha Venkatesh • 110310657 • vavenkatesh@cs.stonybrook.edu
Deepika Peringanji • 110614748• dperinganji@cs.stonybrook.edu

Overview

Motivation

RNNs

LSTMs

RNN LSTM application

Citation

Deep Sentence Embedding Using Long Short-Term Memory Networks - Hamid Palangi, Li Deng, Yelong Shen, Jianfeng Gao, Xiaodong He, Jianshu Chen, Xinying Song, Rabab Ward

References

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

https://en.wikipedia.org/wiki/Recurrent_neural_network

https://en.wikipedia.org/wiki/Long_short-term_memory

Motivation

➢ In traditional neural networks we assume that all inputs (and outputs) are independent of each other

➢ Case: Predicting the next word given a sentence - Do neural networks work here?

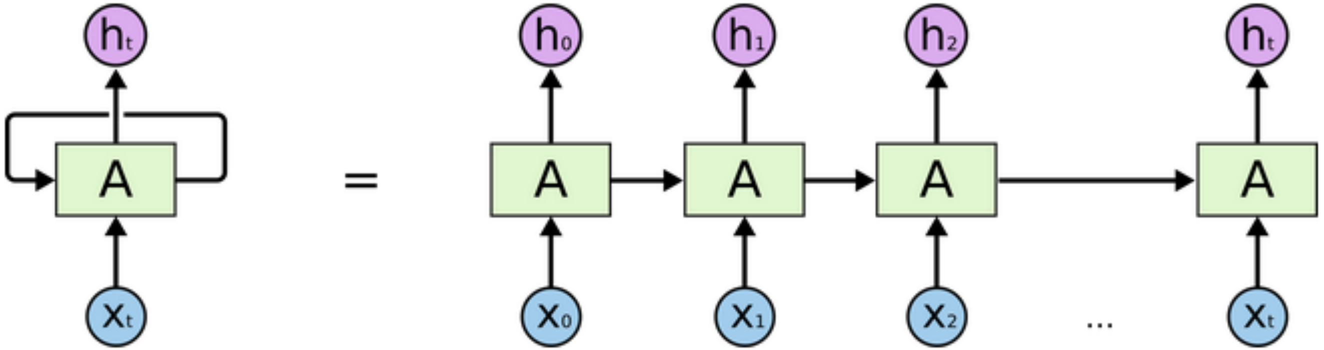
➢ Answer: No, Because they don’t have the knowledge of previous inputs

➢ So there is a need to develop models which would remember previous inputs also

What are RNNs?

- RNNs - Recurrent Neural Networks
- The idea behind RNNs is to make use of sequential information
- RNNs are called *recurrent* because they perform the same task for every element of a sequence, with the output being dependent on the previous computations
- Another way to think about RNNs is that they have a “memory” which captures information about what has been calculated so far

RNNs



An unrolled recurrent neural network.

Recurrent neural networks address this issue. They are networks with loops in them, allowing information to persist

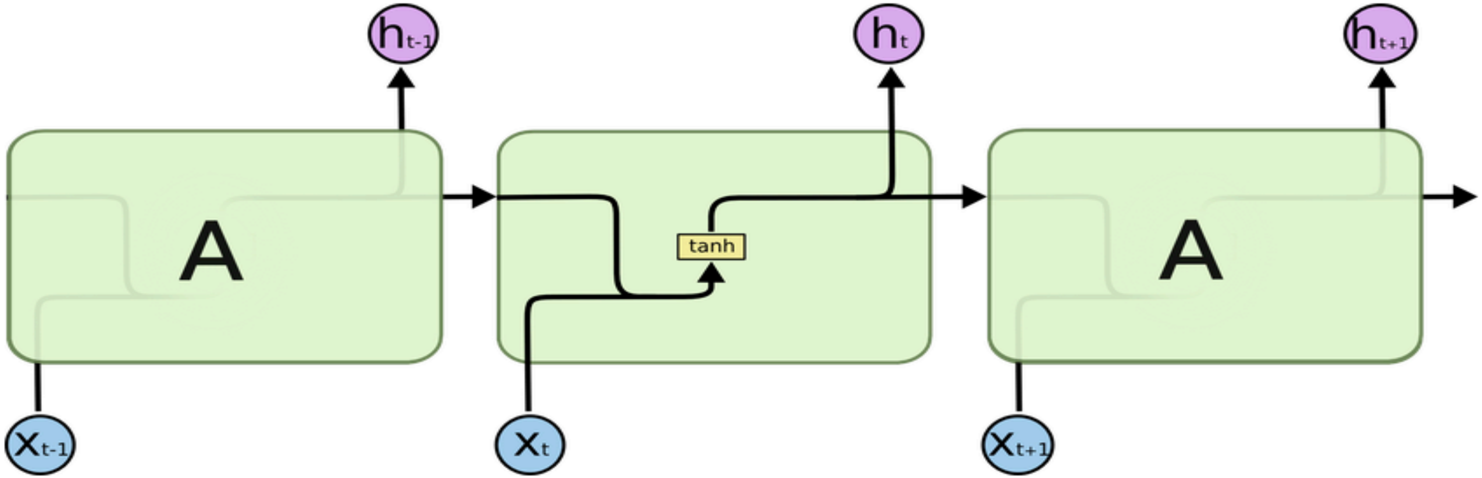
A → Chunk of neural network

x_i → i^{th} word in the sentence

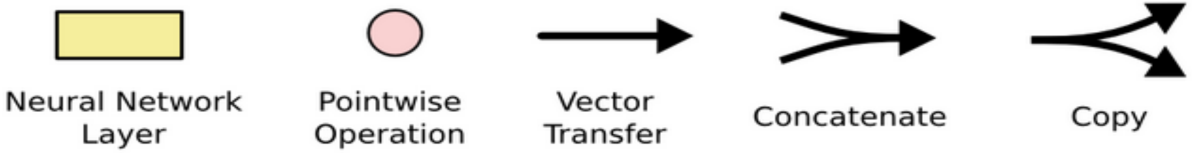
h_i → output at step i

Example: Stars are in the SKY

RNN details



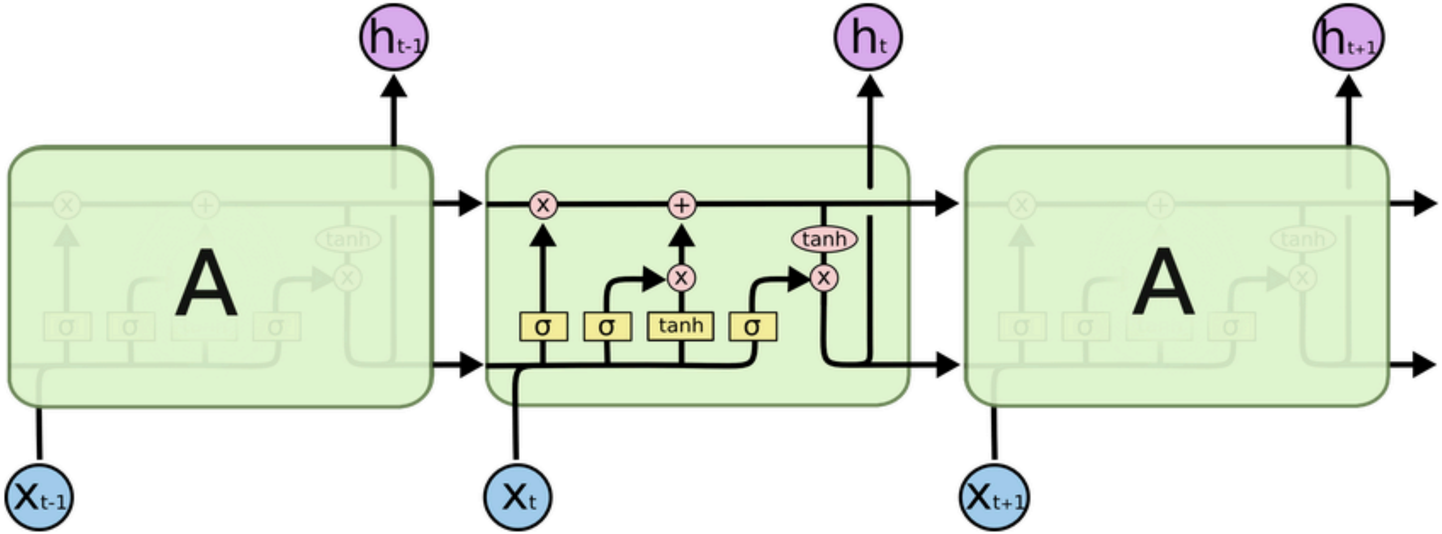
The repeating module in a standard RNN contains a single layer.



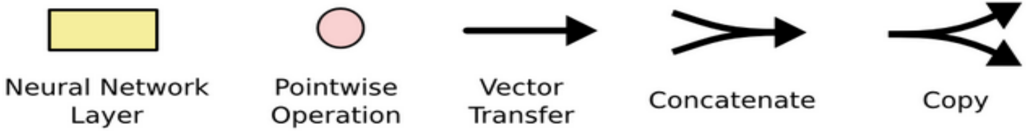
Problem with RNNs

- Example: I grew up in France...I speak fluent FRENCH
- Recent information suggests that the next word is probably the name of a language, but to narrow down to language, we need the context of France, from further back
- Have tendency to store previous information but don't have long term memory/knowledge
- As gap grows RNNs cannot learn to connect the information in the sentences
- Solution: LSTMs - Long Short Term Memory cells
- LSTMs are explicitly designed to avoid the long-term dependency problem

How LSTMs work?

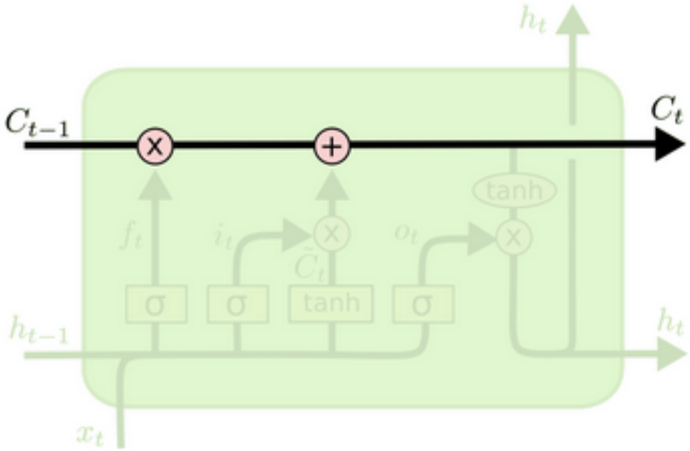


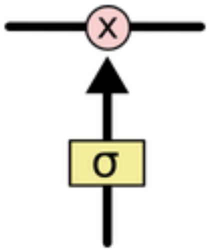
The repeating module in an LSTM contains four interacting layers.



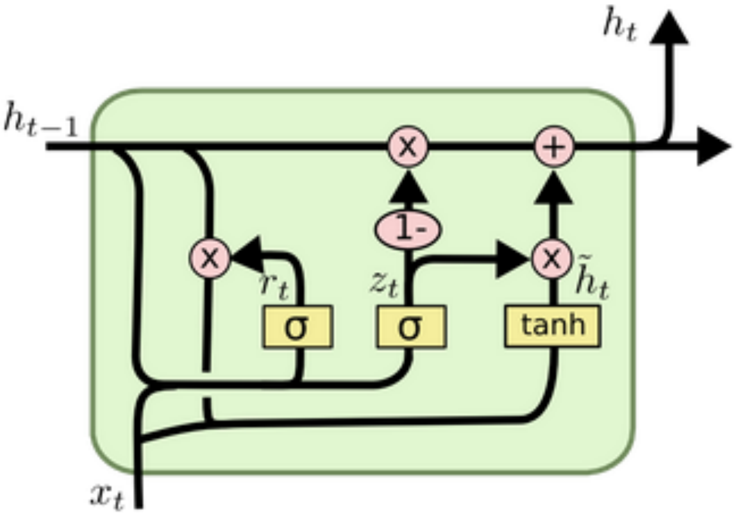
Core Idea Behind LSTMs

- The key to LSTMs is the cell state, the horizontal line running through the top of the diagram
- The LSTM does have the ability to remove or add information to the cell state, carefully regulated by structures called gates
- Gates are composed of a sigmoid neural net layer and a pointwise multiplication operation
- The sigmoid layer outputs numbers between zero and one
- The sigmoid layer controls the information flow into the cell
- An LSTM has three of these gates, to protect and control the cell state





A Variant of LSTM cell



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

Forward and Backward Pass

Forward pass - The input gate learns to decide when to let activation pass into the memory cell and the output gate learns when to let activation pass out of the memory cell.

Backwards pass - The output gate is learning when to let error flow into the memory cell, and the input gates are learning when to let it flow out of the memory cell and to through the rest of the network.

Example Application of RNN LSTMs

Question Answering System

NLP techniques have proved extensively useful in Question-answering systems.

Our main task was to develop a system which can solve the questions asked by a 4th grader with respect to physical processes.

The solution to the problem of Question Answering system is to match the question with the right answer which is done by comparing the phrases extracted from the questions and answers in four categories such as

UNDERGOER

ENABLER,

TRIGGER

Why Phrase Similarity?

As we are dealing with physical processes, it is intuitively better approach to find the similarity between these individual phrases and then combine them to form total score.

Hence Phrase Similarity is the basic problem to be addressed.

Phrase Embedding

- Phrase Embedding is a hot topic of today’s NLP
- Converting a given input sentence into vector is called sentence embedding which gives the semantic meaning of the sentence
- The problem of learning the semantic meaning of a sentence have many applications apart from question answering system, for examples in
    - document retrieval
    - Handwriting Recognition

How is it done?

Inspired from word embedding, phrase embeddings can be used to find similarity between two phrases.

After mapping phrases into a unified vector space,finding distance between vectors gives the most matched phrases using Cosine Similarity,

$$\frac{word2vec(s1)^{T}.\ word2vec(s2)}{||word2vec(s1)||\ ||word2vec(s2)||}$$

Example of two similar phrases is as follows

Using LSTM RNN

LSTM RNN’s main advantage is its ability to capture long term memory

LSTM RNN takes each word of the sentence and embeds them into a semantic vector

This semantic vector represents the semantic meaning of a sentence

LSTM RNN allow for automatic detection of keywords in a given sentence

Using LSTM RNN

The LSTM RNN model automatically discards the unimportant words and detects the main keywords of the sentence.
These keywords activate different cells of the LSTM RNN.
It is observed that words of similar topic activate the same cell

Example :

Sentence vector is formed from the input English sentence and the vector is used to generate an equivalent French sentence using LSTM RNN.

Prediction Stratergy

Each question answer has 4 pair of sentences or 4 features - QUNDERGOER, AUNDERGOER), (QENABLER, AENABLER), (QTRIGGER, ATRIGGER), (QRESULT, ARESULT)

These features can be found by the existing system provided by using existing NLP Techniques.

Once our neural model is trained, for each answer choice:

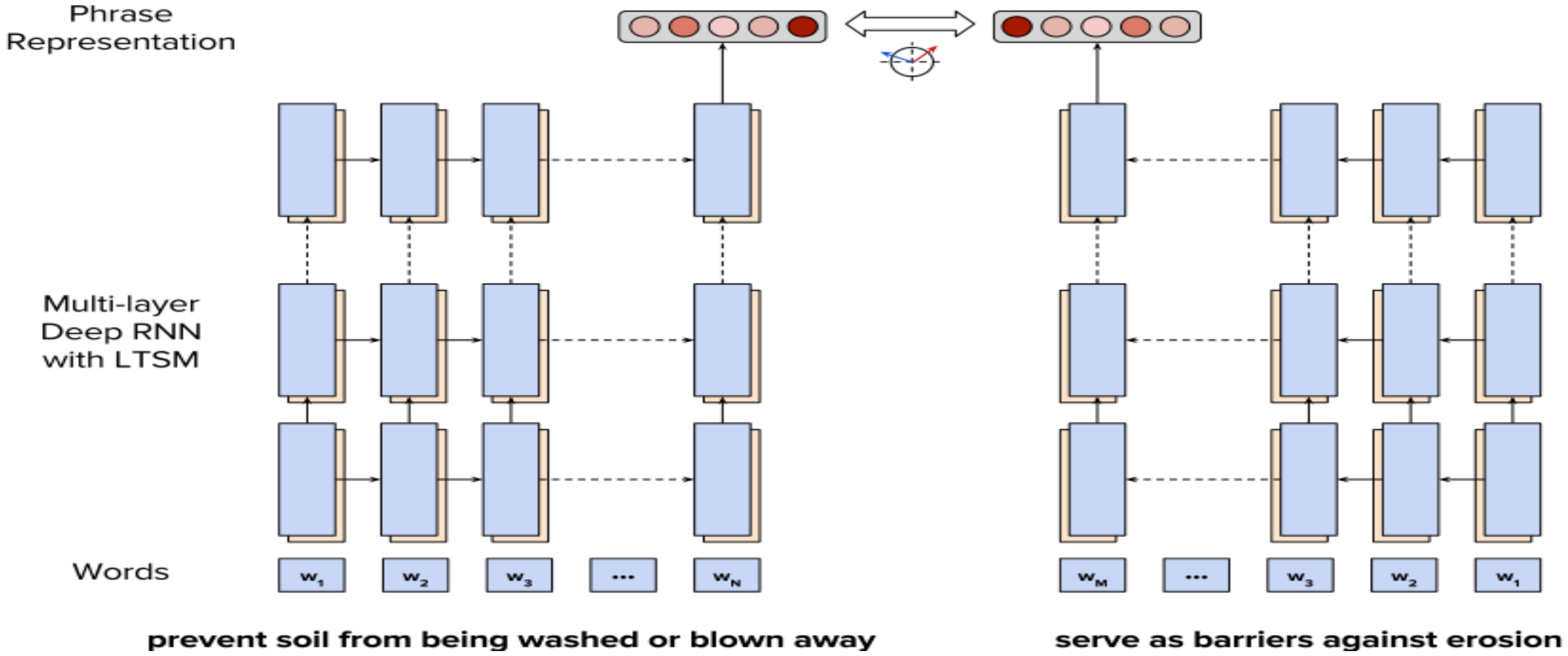
We find out similarity score for pair of sentences in each feature.

We average out the similarity score found for the 4 features of a given answer choice.

This score is then considered to be likelihood score of this answer being the correct choice for this question.

We simply repeat this process for all answer choices of a given question, and report the answer choice with the highest likelihood score.

Phrase Representation using RNN



Approach

We have two sentences as input between which we are finding similarity so two RNNs are used, one for each input sentence.

Error is generated using cost function ***C(I) = (CS(I) − ET (I))*** backpropagate the error and learn parameters for both RNNs.

I corresponds to an input instance. An input instance consists of two sentences, each of which is initially passed to corresponding RNN to get their o(n) value at first epoch.

CS(I) is the cosine similarity score of dot product of o(n) vector of two sentences(n1, n2)

CS(I) = o(n1).o(n2)

ET (I) is the entailment score/ground truth for this input instance

Questions?

Thank You!