Navigation

Start Here Blog Books About Contact

Search...

A Gentle Introduction to Long Short-Term Memory Networks by the Experts

by **Jason Brownlee** on May 24, 2017 in **Long Short-Term Memory Networks**

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network capable of learning order dependence in sequence prediction problems.

This is a behavior required in complex problem domains like machine translation, speech recognition, and more.

LSTMs are a complex area of deep learning. It can be hard to get your hands around what LSTMs are, and how terms like bidirectional and sequence-to-sequence relate to the field.

In this post, you will get insight into LSTMs using the words of research scientists that developed the
problems.

There are few that are better at clearly and precisely articulating both the promise of LSTMs and how [redacted]

We will explore key questions in the field of LSTMs using quotes from the experts, and if you're interested, you will be able to dive into the original papers from which the quotes were taken.

A Gentle Introduction to Long Short-Term Memory Networks by the Experts
Photo by Oran Viriyincy, some rights reserved.

The Promise of Recurrent Neural Networks

Recurrent neural networks are different from traditional feed-forward neural networks.

This difference in the addition of complexity comes with the promise of new behaviors that the traditi

[redacted]

> *Recurrent networks … have an internal state that can represent context information. … [they] keep information about past inputs for an amount of time that is not fixed a priori, but rather depends on its weights and on the input data.*
>
> …
>
> *A recurrent network whose inputs are not fixed but rather constitute an input sequence can be used to transform an input sequence into an output sequence while taking into account contextual information in a flexible way.*

— Yoshua Bengio, et al., Learning Long-Term Dependencies with Gradient Descent is Difficult, 1994.

The paper defines 3 basic requirements of a recurrent neural network:

- That the system be able to store information for an arbitrary duration.
- That the system be resistant to noise (i.e. fluctuations of the inputs that are random or irrelevant
- That the system parameters be trainable (in reasonable time).

The paper also describes the "minimal task" for demonstrating recurrent neural networks.

Context is key.

Recurrent neural networks must use context when making predictions, but to this extent, the context

> *… recurrent neural networks contain cycles that feed the network activations from a previous predictions at the current time step. These activations are stored in the internal states of the temporal contextual information. This mechanism allows RNNs to exploit a dynamically chan history*

— Hassim Sak, et al., Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling, 2014

Need help with LSTMs for Sequence Pre

LSTMs Deliver on the Promise

The success of LSTMs is in their claim to be one of the first implements to overcome the technical problems and deliver on the promise of recurrent neural networks.

> *Hence standard RNNs fail to learn in the presence of time lags greater than 5 – 10 discrete t target signals. The vanishing error problem casts doubt on whether standard RNNs can inde time window-based feedforward networks. A recent model, "Long Short-Term Memory" (LST learn to bridge minimal time lags in excess of 1000 discrete time steps by enforcing constant (CECs) within special units, called cells*

— Felix A. Gers, et al., Learning to Forget: Continual Prediction with LSTM, 2000

The two technical problems overcome by LSTMs are vanishing gradients and exploding gradients, b

> *Unfortunately, the range of contextual information that standard RNNs can access is in practice quite limited. The problem is that the influence of a given input on the hidden layer, and therefore on the network output, either decays or blows up exponentially as it cycles around the network's recurrent connections. This shortcoming … referred to in the literature as the vanishing gradient problem … Long Short-Term Memory (LSTM) is an RNN architecture specifically designed to address the vanishing gradient problem.*

— Alex Graves, et al., A Novel Connectionist System for Unconstrained Handwriting Recognition, 2009

The key to the LSTM solution to the technical problems was the specific internal structure of the unit

> *… governed by its ability to deal with vanishing and exploding gradients, the most common challenge in designing and training RNNs. To address this challenge, a particular form of recurrent nets, called LSTM, was introduced and applied with great success to translation and sequence generation.*

— Alex Graves, et al., Framewise Phoneme Classification with Bidirectional LSTM and Other Neural Network Architectures, 2005.

How do LSTMs Work?

Rather than go into the equations that govern how LSTMs are fit, analogy is a useful tool to quickly get a handle on how they work.

> *We use networks with one input layer, one hidden layer, and one output layer… The (fully) se*
> *and corresponding gate units…*
>
> …
>
> *Each memory cell's internal architecture guarantees constant error ow within its constant err*
> *bridging very long time lags. Two gate units learn to open and close access to error ow withir* *t*
> *gate affords protection of the CEC from perturbation by irrelevant inputs. Likewise, the multip*
> *perturbation by currently irrelevant memory contents.*

— Sepp Hochreiter and Jurgen Schmidhuber, Long Short-Term Memory, 1997.

Multiple analogies can help to give purchase on what differentiates LSTMs from traditional neural ne

> *The Long Short Term Memory architecture was motivated by an analysis of error flow in existing RNNs which found that long time lags were inaccessible to existing architectures, because backpropagated error either blows up or decays exponentially.*
>
> *An LSTM layer consists of a set of recurrently connected blocks, known as memory blocks. These blocks can be thought of as a differentiable version of the memory chips in a digital computer. Each one contains one or more recurrently connected memory cells and three multiplicative units – the input, output and forget gates – that provide continuous analogues of write, read and reset operations for the cells. … The net can only interact with the cells via the gates.*

— Alex Graves, et al., Framewise Phoneme Classification with Bidirectional LSTM and Other Neural Network Architectures, 2005.

It is interesting to note, that even after more than 20 years, the simple (or vanilla) LSTM may still be the best place to start when applying the technique.

> *The most commonly used LSTM architecture (vanilla LSTM) performs reasonably well on various datasets…*
>
> *Learning rate and network size are the most crucial tunable LSTM hyperparameters …*
>
> *… This implies that the hyperparameters can be tuned independently. In particular, the learning rate can be calibrated first using a fairly small network, thus saving a lot of experimentation time.*

— Klaus Greff, et al., LSTM: A Search Space Odyssey, 2015

What are LSTM Applications?

It is important to get a handle on exactly what type of sequence learning problems that LSTMs are s

> *Long Short-Term Memory (LSTM) can solve numerous tasks not solvable by previous learnir*
> *(RNNs).*
>
> *…*
>
> *… LSTM holds promise for any sequential processing task in which we suspect that a hierar*
> *advance what this decomposition is.*

— Felix A. Gers, et al., Learning to Forget: Continual Prediction with LSTM, 2000

> *The Recurrent Neural Network (RNN) is neural sequence model that achieves state of the art performance on important tasks that include language modeling, speech recognition, and machine translation.*

— Wojciech Zaremba, Recurrent Neural Network Regularization, 2014.

> *Since LSTMs are effective at capturing long-term temporal dependencies without suffering from the optimization hurdles that plague simple recurrent networks (SRNs), they have been used to advance the state of the art for many difficult problems. This includes handwriting recognition and generation, language modeling and translation, acoustic modeling of speech, speech synthesis, protein secondary structure prediction, analysis of audio, and video data among others.*

— Klaus Greff, et al., LSTM: A Search Space Odyssey, 2015

What are Bidirectional LSTMs?

A commonly mentioned improvement upon LSTMs are bidirectional LSTMs.

> *The basic idea of bidirectional recurrent neural nets is to present each training sequence forv nets, both of which are connected to the same output layer. … This means that for every poi sequential information about all points before and after it. Also, because the net is free to use there is no need to find a (task-dependent) time-window or target delay size.*
>
> *… for temporal problems like speech recognition, relying on knowledge of the future seems a base our understanding of what we've heard on something that hasn't been said yet? Howev words, and even whole sentences that at first mean nothing are found to make sense in the l*

— Alex Graves, et al., Framewise Phoneme Classification with Bidirectional LSTM and Other Neural

> *One shortcoming of conventional RNNs is that they are only able to make use of previous co processing the data in both directions with two separate hidden layers, which are then fed forwards to the same output layer. … Combining BRNNs with LSTM gives bidirectional LSTM, which can access long-range context in both input directions*

— Alex Graves, et al., Speech recognition with deep recurrent neural networks, 2013

> *Unlike conventional RNNs, bidirectional RNNs utilize both the previous and future context, by processing the data from two directions with two separate hidden layers. One layer processes the input sequence in the forward direction, wh*

direction. The output of current time step is then generated by combining both layers' hidden [redacted]

— Di Wang and Eric Nyberg, A Long Short-Term Memory Model for Answer Sentence Selection in Question Answering, 2015

What are seq2seq LSTMs or RNN Encoder-Decoders?

The sequence-to-sequence LSTM, also called encoder-decoder LSTMs, are an application of LSTMs that are receiving a lot of attention given their impressive capability.

> *… a straightforward application of the Long Short-Term Memory (LSTM) architecture can solve general sequence to sequence problems.*
>
> *…*
>
> *The idea is to use one LSTM to read the input sequence, one timestep at a time, to obtain la*
> *then to use another LSTM to extract the output sequence from that vector. The second LSTM*
> *language model except that it is conditioned on the input sequence.*
>
> *The LSTM's ability to successfully learn on data with long range temporal dependencies mak*
> *the considerable time lag between the inputs and their corresponding outputs.*
>
> *We were able to do well on long sentences because we reversed the order of words in the so*
> *training and test set. By doing so, we introduced many short term dependencies that made th*
> *simple trick of reversing the words in the source sentence is one of the key technical contribu*

— Ilya Sutskever, et al., Sequence to Sequence Learning with Neural Networks, 2014

> *An "encoder" RNN reads the source sentence and transforms it into a rich fixed-length vector representation, which in turn in used as the initial hidden state of a "decoder" RNN that generates the target sentence. Here, we propose to follow this elegant recipe, replacing the encoder RNN by a deep convolution neural network (CNN). … it is natural to use a CNN as an image "encoder", by first pre-training it for an image classification task and using the last hidden layer as an input to the RNN decoder that genera*

— Oriol Vinyals, et al., Show and Tell: A Neural Image Caption Generator, 2014

> *… an RNN Encoder–Decoder, consists of two recurrent neural networks (RNN) that act as an encoder and a decoder pair. The encoder maps a variable-length source sequence to a fixed-length vector, and the decoder maps the vector representation back to a variable-length target sequence.*

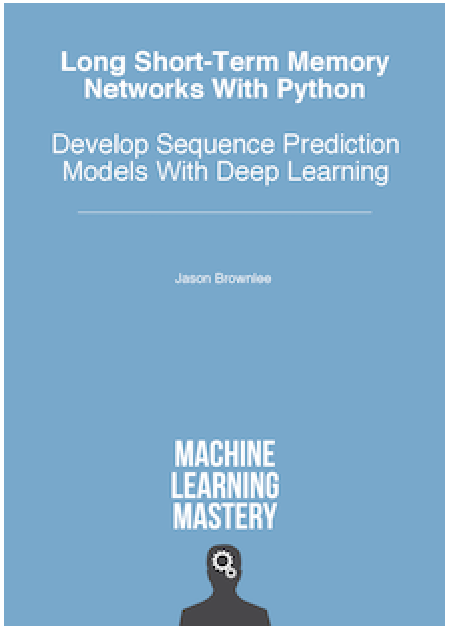
— Kyunghyun Cho, et al., Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation, 2014

Summary

In this post, you received a gentle introduction to LSTMs in the words of the research scientists that developed and applied the techniques.

This provides you both with a clear and precise idea of what LSTMs are and how they work, as well … in the field of recurrent neural networks.

Did any of the quotes help your understanding or inspire you?
Let me know in the comments below.



Skip the Academics. Jus[redacted]

Click to learn more.

About Jason Brownlee

Dr. Jason Brownlee is a husband, proud father, academic researcher, author, professional devel
to helping developers get started and get good at applied machine learning. Learn more.
View all posts by Jason Brownlee →

< The Promise of Recurrent Neural Networks for Time Series Forecasting

On the Suitability of Long Short-Term Memory Networks for Time Series Forecasting >

Get Your Start in Machine Learning

8 Responses to *A Gentle Introduction to Long Short-Term Memory Networks by the Experts*

Mehrdad May 26, 2017 at 5:36 am # REPLY

I am not expert but I think it's better to use time steps instead of time lags, As most papers use it.
I also confused about definition of time lags in another article here 🙂

Jason Brownlee June 2, 2017 at 11:49 am # REPLY

Yes, it is better tot use past observations as time steps when inputting to the model.

Dhineshkumar July 8, 2017 at 12:06 am #

Hi Jason,
Can you please tell me how LSTMs are different from Autoregressive neural networks?

Jason Brownlee July 9, 2017 at 10:47 am #

Yes, no fixed length input or output sequences.

Claudio July 11, 2017 at 8:33 am # REPLY

Hello, good explanation and intoduction.
Can you please help me with something? The input layers of a LSTM net.

For exemple, if I have this:

model.add(LSTM(4))

model.add(Dense(1))

How many neurons I have on my input layers? I think the first line of code refer to the hidden layers, how things get in?

Jason Brownlee July 11, 2017 at 10:39 am # REPLY

These are not input layers, but are instead hidden layers.

You must specify the size of the expected input as an argument “input_shape=(xx,xx)” on the first hid

The input_shape specifies a tuple that specifies the number of time steps and features. A feature is t

See this post for more:
http://machinelearningmastery.com/5-step-life-cycle-long-short-term-memory-models-keras/

Does that help?

abc September 30, 2017 at 1:21 am #

waste of my time.

Jason Brownlee September 30, 2017 at 7:43 am # REPLY

Sorry to hear that.

Leave a Reply

Name (required)

Email (will not be published) (required)

Website

SUBMIT COMMENT

Welcome to Machine Learning Mastery

Hi, I'm Dr. Jason Brownlee.
My goal is to make practitioners like YOU awesome at applied machine learning.

Read More

Deep Learning for Sequence Prediction

Cut through the math and research papers.
Discover 4 Models, 6 Architectures, and 14 Tutorials.

POPULAR


Time Series Prediction with LSTM Recurrent Neural Networks in Python with Keras
JULY 21, 2016


Your First Machine Learning Project in Python Step-By-Step
JUNE 10, 2016


Develop Your First Neural Network in Python With Keras Step-By-Step
MAY 24, 2016

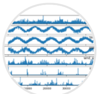

Sequence Classification with LSTM Recurrent Neural Networks in Python with Keras
JULY 26, 2016

How to Setup a Python Environment for Machine Learning and Deep Learning with Anaconda
MARCH 13, 2017

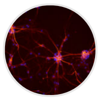
Time Series Forecasting with the Long Short-Term Memory Network in Python
APRIL 7, 2017

Multi-Class Classification Tutorial with the Keras Deep Learning Library
JUNE 2, 2016

Regression Tutorial with the Keras Deep Learning Library in Python
JUNE 9, 2016



Multivariate Time Series Forecasting with LSTMs in Keras
AUGUST 14, 2017



How to Implement the Backpropagation Algorithm From Scratch In Python
NOVEMBER 7, 2016

Privacy | Contact | About

Get Your Start in Machine Learning