Title: Long short-term memory

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

PageID: 10711453

Categories: Category:1997 in artificial intelligence, Category:Deep learning, Category:Neural network architectures



---

Supervised learning

Unsupervised learning

Semi-supervised learning

Self-supervised learning

Reinforcement learning

Meta-learning

Online learning

Batch learning

Curriculum learning

Rule-based learning

Neuro-symbolic AI

Neuromorphic engineering

Quantum machine learning

Classification

Generative modeling

Regression

Clustering

Dimensionality reduction

Density estimation

Anomaly detection

Data cleaning

AutoML

Association rules

Semantic analysis

Structured prediction

Feature engineering

Feature learning

Learning to rank

Grammar induction

Ontology learning

Multimodal learning

Apprenticeship learning

Decision trees

Ensembles Bagging Boosting Random forest

Bagging

Boosting

Random forest

k -NN

Linear regression

Naive Bayes

Artificial neural networks

Logistic regression

Perceptron

Relevance vector machine (RVM)

Support vector machine (SVM)

BIRCH

CURE

Hierarchical

k -means

Fuzzy

Expectation–maximization (EM)

DBSCAN

OPTICS

Mean shift

Factor analysis

CCA

ICA

LDA

NMF

PCA

PGD

t-SNE

SDL

Graphical models Bayes net Conditional random field Hidden Markov

Bayes net

Conditional random field

Hidden Markov

RANSAC

k -NN

Local outlier factor

Isolation forest

Autoencoder

Deep learning

Feedforward neural network

Recurrent neural network LSTM GRU ESN reservoir computing

LSTM

GRU

ESN

reservoir computing

Boltzmann machine Restricted

Restricted

GAN

Diffusion model

SOM

Convolutional neural network U-Net LeNet AlexNet DeepDream

U-Net

LeNet

AlexNet

DeepDream

Neural field Neural radiance field Physics-informed neural networks

Neural radiance field

Physics-informed neural networks

Transformer Vision

Vision

Mamba

Spiking neural network

Memtransistor

Electrochemical RAM (ECRAM)

Q-learning

Policy gradient

SARSA

Temporal difference (TD)

Multi-agent Self-play

Self-play

Active learning

Crowdsourcing

Human-in-the-loop

Mechanistic interpretability

RLHF

Coefficient of determination

Confusion matrix

Learning curve

ROC curve

Kernel machines

Bias–variance tradeoff

Computational learning theory

Empirical risk minimization

Occam learning

PAC learning

Statistical learning

VC theory

Topological deep learning

AAAI

ECML PKDD

NeurIPS

ICML

ICLR

IJCAI

ML

JMLR

Glossary of artificial intelligence

List of datasets for machine-learning research List of datasets in computer vision and image processing

List of datasets in computer vision and image processing

Outline of machine learning

v

t

e

Long short-term memory ( LSTM ) [ 1 ] is a type of recurrent neural network (RNN) aimed at mitigating the vanishing gradient problem [ 2 ] commonly encountered by traditional RNNs. Its relative insensitivity to gap length is its advantage over other RNNs, hidden Markov models , and other sequence learning methods. It aims to provide a short-term memory for RNN that can last thousands of timesteps (thus " long short-term memory"). [ 1 ] The name is made in analogy with long-term memory and short-term memory and their relationship, studied by cognitive psychologists since the early 20th century.

An LSTM unit is typically composed of a cell and three gates : an input gate, an output gate, [ 3 ] and a forget gate. [ 4 ] The cell remembers values over arbitrary time intervals, and the gates regulate the flow of information into and out of the cell. Forget gates decide what information to

discard from the previous state, by mapping the previous state and the current input to a value between 0 and 1. A (rounded) value of 1 signifies retention of the information, and a value of 0 represents discarding. Input gates decide which pieces of new information to store in the current cell state, using the same system as forget gates. Output gates control which pieces of information in the current cell state to output, by assigning a value from 0 to 1 to the information, considering the previous and current states. Selectively outputting relevant information from the current state allows the LSTM network to maintain useful, long-term dependencies to make predictions, both in current and future time-steps.

LSTM has wide applications in classification , [ 5 ] [ 6 ] data processing , time series analysis tasks, [ 7 ] speech recognition , [ 8 ] [ 9 ] machine translation , [ 10 ] [ 11 ] speech activity detection, [ 12 ] robot control , [ 13 ] [ 14 ] video games , [ 15 ] [ 16 ] healthcare . [ 17 ]

Motivation

In theory, classic RNNs can keep track of arbitrary long-term dependencies in the input sequences. The problem with classic RNNs is computational (or practical) in nature: when training a classic RNN using back-propagation , the long-term gradients which are back-propagated can "vanish" , meaning they can tend to zero due to very small numbers creeping into the computations, causing the model to effectively stop learning. RNNs using LSTM units partially solve the vanishing gradient problem , because LSTM units allow gradients to also flow with little to no attenuation. However, LSTM networks can still suffer from the exploding gradient problem. [ 18 ]

The intuition behind the LSTM architecture is to create an additional module in a neural network that learns when to remember and when to forget pertinent information. [ 4 ] In other words, the network effectively learns which information might be needed later on in a sequence and when that information is no longer needed. For instance, in the context of natural language processing , the network can learn grammatical dependencies. [ 19 ] An LSTM might process the sentence " Dave , as a result of his controversial claims, is now a pariah" by remembering the (statistically likely) grammatical gender and number of the subject Dave , note that this information is pertinent for the pronoun his and note that this information is no longer important after the verb is .

Variants

In the equations below, the lowercase variables represent vectors. Matrices W q {\displaystyle W_{q}} and U q {\displaystyle U_{q}} contain, respectively, the weights of the input and recurrent connections, where the subscript q {\displaystyle _{q}} can either be the input gate i {\displaystyle i} , output gate o {\displaystyle o} , the forget gate f {\displaystyle f} or the memory cell c {\displaystyle c} , depending on the activation being calculated. In this section, we are thus using a "vector notation". So, for example, c t ∈ R h {\displaystyle c_{t}\in \mathbb {R} ^{h}} is not just one unit of one LSTM cell, but contains h {\displaystyle h} LSTM cell's units.

See [ 20 ] for an empirical study of 8 architectural variants of LSTM.

LSTM with a forget gate

The compact forms of the equations for the forward pass of an LSTM cell with a forget gate are: [ 1 ] [ 4 ]

where the initial values are c 0 = 0 {\displaystyle c_{0}=0} and h 0 = 0 {\displaystyle h_{0}=0} and the operator ■ {\displaystyle \odot } denotes the Hadamard product (element-wise product). The subscript t {\displaystyle t} indexes the time step.

Variables

Letting the superscripts d {\displaystyle d} and h {\displaystyle h} refer to the number of input features and number of hidden units, respectively:

x t ∈ R d {\displaystyle x_{t}\in \mathbb {R} ^{d}} : input vector to the LSTM unit

f t ∈ ( 0 , 1 ) h {\displaystyle f_{t}\in {(0,1)}^{h}} : forget gate's activation vector

i t ∈ ( 0 , 1 ) h {\displaystyle i_{t}\in {(0,1)}^{h}} : input/update gate's activation vector

o t ∈ ( 0 , 1 ) h {\displaystyle o_{t}\in {(0,1)}^{h}} : output gate's activation vector

h t ∈ ( − 1 , 1 ) h {\displaystyle h_{t}\in {(-1,1)}^{h}} : hidden state vector also known as output vector of the LSTM unit

c ~ t ∈ ( − 1 , 1 ) h {\displaystyle {\tilde {c}}_{t}\in {(-1,1)}^{h}} : cell input activation vector

c t ∈ R h {\displaystyle c_{t}\in \mathbb {R} ^{h}} : cell state vector

W ∈ R h × d {\displaystyle W\in \mathbb {R} ^{h\times d}} , U ∈ R h × h {\displaystyle U\in \mathbb {R} ^{h\times h}} and b ∈ R h {\displaystyle b\in \mathbb {R} ^{h}} : weight matrices and bias vector parameters which need to be learned during training

Activation functions

σ g {\displaystyle \sigma _{g}} : sigmoid function .

σ c {\displaystyle \sigma _{c}} : hyperbolic tangent function.

σ h {\displaystyle \sigma _{h}} : hyperbolic tangent function or, as the peephole LSTM paper [ 21 ] [ 22 ] suggests, σ h ( x ) = x {\displaystyle \sigma _{h}(x)=x} .

Peephole LSTM

The figure on the right is a graphical representation of an LSTM unit with peephole connections (i.e. a peephole LSTM). [ 21 ] [ 22 ] Peephole connections allow the gates to access the constant error carousel (CEC), whose activation is the cell state. [ 21 ] h t − 1 {\displaystyle h_{t-1}} is not used, c t − 1 {\displaystyle c_{t-1}} is used instead in most places.

Each of the gates can be thought as a "standard" neuron in a feed-forward (or multi-layer) neural network: that is, they compute an activation (using an activation function) of a weighted sum. i t , o t {\displaystyle i_{t},o_{t}} and f t {\displaystyle f_{t}} represent the activations of respectively the input, output and forget gates, at time step t {\displaystyle t} .

The 3 exit arrows from the memory cell c {\displaystyle c} to the 3 gates i , o {\displaystyle i,o} and f {\displaystyle f} represent the peephole connections. These peephole connections actually denote the contributions of the activation of the memory cell c {\displaystyle c} at time step t − 1 {\displaystyle t-1} , i.e. the contribution of c t − 1 {\displaystyle c_{t-1}} (and not c t {\displaystyle c_{t}} , as the picture may suggest). In other words, the gates i , o {\displaystyle i,o} and f {\displaystyle f} calculate their activations at time step t {\displaystyle t} (i.e., respectively, i t , o t {\displaystyle i_{t},o_{t}} and f t {\displaystyle f_{t}} ) also considering the activation of the memory cell c {\displaystyle c} at time step t − 1 {\displaystyle t-1} , i.e. c t − 1 {\displaystyle c_{t-1}} .

The single left-to-right arrow exiting the memory cell is not a peephole connection and denotes c t {\displaystyle c_{t}} .

The little circles containing a × {\displaystyle \times } symbol represent an element-wise multiplication between its inputs. The big circles containing an S -like curve represent the application of a differentiable function (like the sigmoid function) to a weighted sum.

Peephole convolutional LSTM

Peephole convolutional LSTM. [ 23 ] The ∗ {\displaystyle *} denotes the convolution operator.

Training

An RNN using LSTM units can be trained in a supervised fashion on a set of training sequences, using an optimization algorithm like gradient descent combined with backpropagation through time to compute the gradients needed during the optimization process, in order to change each weight of the LSTM network in proportion to the derivative of the error (at the output layer of the LSTM network) with respect to corresponding weight.

A problem with using gradient descent for standard RNNs is that error gradients vanish exponentially quickly with the size of the time lag between important events. This is due to lim n → ∞ W n = 0 {\displaystyle \lim _{n\to \infty }W^{n}=0} if the spectral radius of W {\displaystyle W} is smaller than 1. [ 2 ] [ 24 ]

However, with LSTM units, when error values are back-propagated from the output layer, the error remains in the LSTM unit's cell. This "error carousel" continuously feeds error back to each of the LSTM unit's gates, until they learn to cut off the value.

CTC score function

Many applications use stacks of LSTM RNNs [ 25 ] and train them by connectionist temporal classification (CTC) [ 5 ] to find an RNN weight matrix that maximizes the probability of the label sequences in a training set, given the corresponding input sequences. CTC achieves both alignment and recognition.

Alternatives

Sometimes, it can be advantageous to train (parts of) an LSTM by neuroevolution [ 7 ] or by policy gradient methods, especially when there is no "teacher" (that is, training labels).

Applications

Applications of LSTM include:

Robot control [ 13 ]

Time series prediction [ 7 ]

Speech recognition [ 26 ] [ 27 ] [ 28 ]

Rhythm learning [ 22 ]

Hydrological rainfall–runoff modeling [ 29 ]

Music composition [ 30 ]

Grammar learning [ 31 ] [ 21 ] [ 32 ]

Handwriting recognition [ 33 ] [ 34 ]

Human action recognition [ 35 ]

Sign language translation [ 36 ]

Protein homology detection [ 37 ]

Predicting subcellular localization of proteins [ 38 ]

Time series anomaly detection [ 39 ]

Several prediction tasks in the area of business process management [ 40 ]

Prediction in medical care pathways [ 41 ]

Semantic parsing [ 42 ]

Object co-segmentation [ 43 ] [ 44 ]

Airport passenger management [ 45 ]

Short-term traffic forecast [ 46 ]

Drug design [ 47 ]

Financial forecasting [ 48 ]

Activity classification in video [ 49 ]

2015: Google started using an LSTM trained by CTC for speech recognition on Google Voice . [ 50 ] [ 51 ] According to the official blog post, the new model cut transcription errors by 49%. [ 52 ]

2016: Google started using an LSTM to suggest messages in the Allo conversation app. [ 53 ] In the same year, Google released the Google Neural Machine Translation system for Google Translate which used LSTMs to reduce translation errors by 60%. [ 10 ] [ 54 ] [ 55 ]

Apple announced in its Worldwide Developers Conference that it would start using the LSTM for quicktype [ 56 ] [ 57 ] [ 58 ] in the iPhone and for Siri. [ 59 ] [ 60 ]

Amazon released Polly , which generates the voices behind Alexa, using a bidirectional LSTM for the text-to-speech technology. [ 61 ]

2017: Facebook performed some 4.5 billion automatic translations every day using long short-term memory networks. [ 11 ]

Microsoft reported reaching 94.9% recognition accuracy on the Switchboard corpus , incorporating a vocabulary of 165,000 words. The approach used "dialog session-based long-short-term memory". [ 62 ]

2018: OpenAI used LSTM trained by policy gradients to beat humans in the complex video game of Dota 2, [ 15 ] and to control a human-like robot hand that manipulates physical objects with unprecedented dexterity. [ 14 ] [ 63 ]

2019: DeepMind used LSTM trained by policy gradients to excel at the complex video game of Starcraft II . [ 16 ] [ 63 ]

History

Development

Aspects of LSTM were anticipated by "focused back-propagation" (Mozer, 1989), [ 64 ] cited by the LSTM paper. [ 1 ]

Sepp Hochreiter's 1991 German diploma thesis analyzed the vanishing gradient problem and developed principles of the method. [ 2 ] His supervisor, Jürgen Schmidhuber , considered the thesis highly significant. [ 65 ]

An early version of LSTM was published in 1995 in a technical report by Sepp Hochreiter and Jürgen Schmidhuber , [ 66 ] then published in the NIPS 1996 conference. [ 3 ]

The most commonly used reference point for LSTM was published in 1997 in the journal Neural Computation . [ 1 ] By introducing Constant Error Carousel (CEC) units, LSTM deals with the vanishing gradient problem . The initial version of LSTM block included cells, input and output gates. [ 20 ]

( Felix Gers , Jürgen Schmidhuber, and Fred Cummins, 1999) [ 67 ] introduced the forget gate (also called "keep gate") into the LSTM architecture in 1999, enabling the LSTM to reset its own state. [ 20 ] This is the most commonly used version of LSTM nowadays.

(Gers, Schmidhuber, and Cummins, 2000) added peephole connections. [ 21 ] [ 22 ] Additionally, the output activation function was omitted. [ 20 ]

Development of variants

(Graves, Fernandez, Gomez, and Schmidhuber, 2006) [ 5 ] introduce a new error function for LSTM: Connectionist Temporal Classification (CTC) for simultaneous alignment and recognition of sequences.

(Graves, Schmidhuber, 2005) [ 26 ] published LSTM with full backpropagation through time and bidirectional LSTM.

(Kyunghyun Cho et al., 2014) [ 68 ] published a simplified variant of the forget gate LSTM [ 67 ] called Gated recurrent unit (GRU).

(Rupesh Kumar Srivastava, Klaus Greff, and Schmidhuber, 2015) used LSTM principles [ 67 ] to create the Highway network , a feedforward neural network with hundreds of layers, much deeper than previous networks. [ 69 ] [ 70 ] [ 71 ] Concurrently, the ResNet architecture was developed. It is equivalent to an open-gated or gateless highway network. [ 72 ]

A modern upgrade of LSTM called xLSTM is published by a team led by Sepp Hochreiter (Maximilian et al, 2024). [ 73 ] [ 74 ] One of the 2 blocks (mLSTM) of the architecture are parallelizable like the Transformer architecture, the other ones (sLSTM) allow state tracking.

Applications

2001: Gers and Schmidhuber trained LSTM to learn languages unlearnable by traditional models such as Hidden Markov Models. [ 21 ] [ 63 ]

Hochreiter et al. used LSTM for meta-learning (i.e. learning a learning algorithm). [ 75 ]

2004: First successful application of LSTM to speech Alex Graves et al. [ 76 ] [ 63 ]

2005: Daan Wierstra, Faustino Gomez, and Schmidhuber trained LSTM by neuroevolution without a teacher. [ 7 ]

Mayer et al. trained LSTM to control robots . [ 13 ]

2007: Wierstra, Foerster, Peters, and Schmidhuber trained LSTM by policy gradients for reinforcement learning without a teacher. [ 77 ]

Hochreiter, Heuesel, and Obermayr applied LSTM to protein homology detection the field of biology . [ 37 ]

2009: Justin Bayer et al. introduced neural architecture search for LSTM. [ 78 ] [ 63 ]

2009: An LSTM trained by CTC won the ICDAR connected handwriting recognition competition. Three such models were submitted by a team led by Alex Graves . [ 79 ] One was the most accurate model in the competition and another was the fastest. [ 80 ] This was the first time an RNN won international competitions. [ 63 ]

2013: Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton used LSTM networks as a major component of a network that achieved a record 17.7% phoneme error rate on the classic TIMIT natural speech dataset. [ 28 ]

2017: Researchers from Michigan State University , IBM Research , and Cornell University published a study in the Knowledge Discovery and Data Mining (KDD) conference. [ 81 ] Their time-aware LSTM (T-LSTM) performs better on certain data sets than standard LSTM.

See also

Attention (machine learning)

Deep learning

Differentiable neural computer

Gated recurrent unit

Highway network

Long-term potentiation

Prefrontal cortex basal ganglia working memory

Recurrent neural network

Seq2seq

Transformer (machine learning model)

Time series

References

Further reading

Monner, Derek D.; Reggia, James A. (2010). "A generalized LSTM-like training algorithm for second-order recurrent neural networks" (PDF) . Neural Networks . 25 (1): 70– 83. doi : 10.1016/j.neunet.2011.07.003 . PMC 3217173 . PMID 21803542 . High-performing extension of LSTM that has been simplified to a single node type and can train arbitrary architectures

Gers, Felix A.; Schraudolph, Nicol N.; Schmidhuber, Jürgen (Aug 2002). "Learning precise timing with LSTM recurrent networks" (PDF) . Journal of Machine Learning Research . 3 : 115– 143.

Gers, Felix (2001). "Long Short-Term Memory in Recurrent Neural Networks" (PDF) . PhD thesis .

Abidogun, Olusola Adeniyi (2005). Data Mining, Fraud Detection and Mobile Telecommunications: Call Pattern Analysis with Unsupervised Neural Networks . Master's Thesis (Thesis). University of the Western Cape. hdl : 11394/249 . Archived (PDF) from the original on May 22, 2012. original with two chapters devoted to explaining recurrent neural networks, especially LSTM.

original with two chapters devoted to explaining recurrent neural networks, especially LSTM.

External links

Recurrent Neural Networks with over 30 LSTM papers by Jürgen Schmidhuber 's group at IDSIA

Zhang, Aston; Lipton, Zachary; Li, Mu; Smola, Alexander J. (2024). "10.1. Long Short-Term Memory (LSTM)" . Dive into deep learning . Cambridge New York Port Melbourne New Delhi Singapore: Cambridge University Press. ISBN 978-1-009-38943-3 .

v

t

e

History timeline

timeline

Companies

Projects

Parameter Hyperparameter

Hyperparameter

Loss functions

Regression Bias–variance tradeoff Double descent Overfitting

Bias–variance tradeoff

Double descent

Overfitting

Clustering

Gradient descent SGD Quasi-Newton method Conjugate gradient method

SGD

Quasi-Newton method

Conjugate gradient method

Backpropagation

Attention

Convolution

Normalization Batchnorm

Batchnorm

Activation Softmax Sigmoid Rectifier

Softmax

Sigmoid

Rectifier

Gating

Weight initialization

Regularization

Datasets Augmentation

Augmentation

Prompt engineering

Reinforcement learning Q-learning SARSA Imitation Policy gradient

Q-learning

SARSA

Imitation

Policy gradient

Diffusion

Latent diffusion model

Autoregression

Adversary

RAG

Uncanny valley

RLHF

Self-supervised learning

Reflection

Recursive self-improvement

Hallucination

Word embedding

Vibe coding

Machine learning In-context learning

In-context learning

Artificial neural network Deep learning

Deep learning

Language model Large language model NMT

Large language model

NMT

Reasoning language model

Model Context Protocol

Intelligent agent

Artificial human companion

Humanity's Last Exam

Artificial general intelligence (AGI)

AlexNet

WaveNet

Human image synthesis

HWR

OCR

Computer vision

Speech synthesis 15.ai ElevenLabs

15.ai

ElevenLabs

Speech recognition Whisper

Whisper

Facial recognition

AlphaFold

Text-to-image models Aurora DALL-E Firefly Flux Ideogram Imagen Midjourney Recraft Stable Diffusion

Aurora

DALL-E

Firefly

Flux

Ideogram

Imagen

Midjourney

Recraft

Stable Diffusion

Text-to-video models Dream Machine Runway Gen Hailuo AI Kling Sora Veo

Dream Machine

Runway Gen

Hailuo AI

Kling

Sora

Veo

Music generation Riffusion Suno AI Udio

Riffusion

Suno AI

Udio

Word2vec

Seq2seq

GloVe

BERT

T5

Llama

Chinchilla AI

PaLM

GPT 1 2 3 J ChatGPT 4 4o o1 o3 4.5 4.1 o4-mini 5

1

2

3

J

ChatGPT

4

4o

o1

o3

4.5

4.1

o4-mini

5

Claude

Gemini Gemini (language model) Gemma

Gemini (language model)

Gemma

Grok

LaMDA

BLOOM

DBRX

Project Debater

IBM Watson

IBM Watsonx

Granite

PanGu-Σ

DeepSeek

Qwen

AlphaGo

AlphaZero

OpenAI Five

Self-driving car

MuZero

Action selection AutoGPT

AutoGPT
Robot control
Alan Turing
Warren Sturgis McCulloch
Walter Pitts
John von Neumann
Claude Shannon
Shun'ichi Amari
Kunihiko Fukushima
Takeo Kanade
Marvin Minsky
John McCarthy
Nathaniel Rochester
Allen Newell
Cliff Shaw
Herbert A. Simon
Oliver Selfridge
Frank Rosenblatt
Bernard Widrow
Joseph Weizenbaum
Seymour Papert
Seppo Linnainmaa
Paul Werbos
Geoffrey Hinton
John Hopfield
Jürgen Schmidhuber
Yann LeCun
Yoshua Bengio
Lotfi A. Zadeh
Stephen Grossberg
Alex Graves
James Goodnight
Andrew Ng
Fei-Fei Li
Alex Krizhevsky
Ilya Sutskever
Oriol Vinyals
Quoc V. Le

Ian Goodfellow

Demis Hassabis

David Silver

Andrej Karpathy

Ashish Vaswani

Noam Shazeer

Aidan Gomez

John Schulman

Mustafa Suleyman

Jan Leike

Daniel Kokotajlo

François Chollet

Neural Turing machine

Differentiable neural computer

Transformer Vision transformer (ViT)

Vision transformer (ViT)

Recurrent neural network (RNN)

Long short-term memory (LSTM)

Gated recurrent unit (GRU)

Echo state network

Multilayer perceptron (MLP)

Convolutional neural network (CNN)

Residual neural network (RNN)

Highway network

Mamba

Autoencoder

Variational autoencoder (VAE)

Generative adversarial network (GAN)

Graph neural network (GNN)

Category