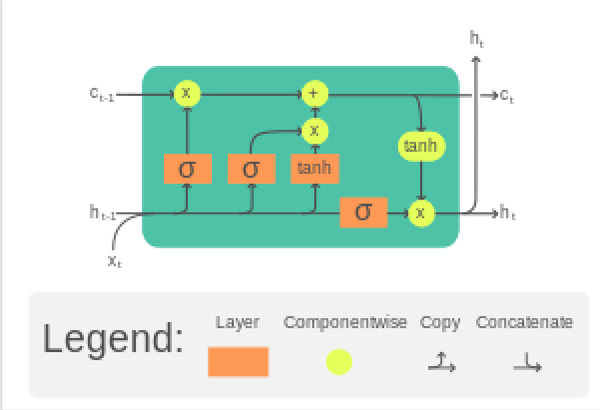
WIKIPEDIA
The Free Encyclopedia

Long short-term memory

Long short-term memory (**LSTM**)[1] network is a recurrent neural network (RNN), aimed at dealing with the vanishing gradient problem[2] present in 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] It is applicable to classification, processing and predicting data based on time series, such as in handwriting,[3] speech recognition,[4][5] machine translation,[6][7] speech activity detection,[8] robot control,[9][10] video games,[11][12] and healthcare.[13]



The Long Short-Term Memory (LSTM) cell can process data sequentially and keep its hidden state through time.

A common LSTM unit is composed of a **cell**, an **input gate**, an **output gate**[14] and a **forget gate**.[15] The cell remembers values over arbitrary time intervals and the three *gates* regulate the flow of information into and out of the cell. Forget gates decide what information to discard from a previous state by assigning a previous state, compared to a current input, a value between 0 and 1. A (rounded) value of 1 means to keep the information, and a value of 0 means to discard it. Input gates decide which pieces of new information to store in the current state, using the same system as forget gates. Output gates control which pieces of information in the current 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.

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" (that is, they can tend to zero) or "explode" (that is, they can tend to infinity),[2] because of the computations involved in the process. RNNs using LSTM units partially solve the vanishing gradient problem, because LSTM units allow gradients to also flow *unchanged*. However, LSTM networks can still suffer from the exploding gradient problem.[16]

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.[15] 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.[17] 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 and U_q contain, respectively, the weights of the input and recurrent connections, where the subscript $_q$ can either be the input gate i, output gate o, the forget gate f or the memory cell c, depending on the activation being calculated. In this section, we are thus using a "vector notation". So, for example, $c_t \in \mathbb{R}^h$ is not just one unit of one LSTM cell, but contains h LSTM cell's units.

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][15]

$$\begin{aligned}
f_t &= \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \\
i_t &= \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \\
o_t &= \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \\
\tilde{c}_t &= \sigma_c(W_c x_t + U_c h_{t-1} + b_c) \\
c_t &= f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \\
h_t &= o_t \odot \sigma_h(c_t)
\end{aligned}$$

where the initial values are $c_0 = 0$ and $h_0 = 0$ and the operator $\odot$ denotes the Hadamard product (element-wise product). The subscript t indexes the time step.

Variables

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

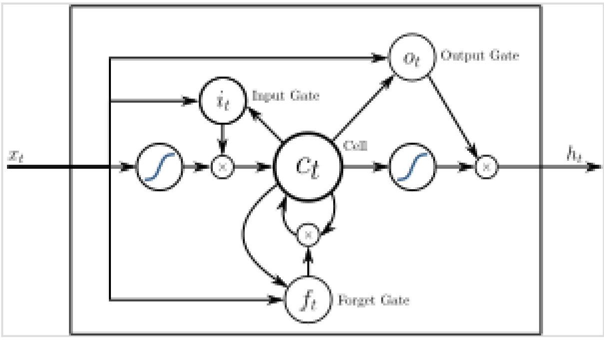
- $x_t \in \mathbb{R}^d$: input vector to the LSTM unit
- $f_t \in (0,1)^h$: forget gate's activation vector
- $i_t \in (0,1)^h$: input/update gate's activation vector
- $o_t \in (0,1)^h$: output gate's activation vector
- $h_t \in (-1,1)^h$: hidden state vector also known as output vector of the LSTM unit
- $\tilde{c}_t \in (-1,1)^h$: cell input activation vector
- $c_t \in \mathbb{R}^h$: cell state vector
- $W \in \mathbb{R}^{h \times d}$, $U \in \mathbb{R}^{h \times h}$ and $b \in \mathbb{R}^h$: weight matrices and bias vector parameters which need to be learned during training

Activation functions

- σ_g: sigmoid function.
- σ_c: hyperbolic tangent function.

- σ_h: hyperbolic tangent function or, as the peephole LSTM paper[18][19] suggests, $\sigma_h(x) = x$.

Peephole LSTM



A peephole LSTM unit with input (i.e. i), output (i.e. o), and forget (i.e. f) gates

The figure on the right is a graphical representation of an LSTM unit with peephole connections (i.e. a peephole LSTM).[18][19] Peephole connections allow the gates to access the constant error carousel (CEC), whose activation is the cell state.[18] h_{t-1} is not used, c_{t-1} is used instead in most places.

$$
\begin{aligned}
f_t &= \sigma_g(W_f x_t + U_f c_{t-1} + b_f) \\
i_t &= \sigma_g(W_i x_t + U_i c_{t-1} + b_i) \\
o_t &= \sigma_g(W_o x_t + U_o c_{t-1} + b_o) \\
c_t &= f_t \odot c_{t-1} + i_t \odot \sigma_c(W_c x_t + b_c) \\
h_t &= o_t \odot \sigma_h(c_t)
\end{aligned}
$$

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 and f_t represent the activations of respectively the input, output and forget gates, at time step t.

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

The single left-to-right arrow exiting the memory cell is *not* a peephole connection and denotes c_t.

The little circles containing a $\times$ 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.[20] The $*$ denotes the convolution operator.

$$
\begin{aligned}
f_t &= \sigma_g(W_f * x_t + U_f * h_{t-1} + V_f \odot c_{t-1} + b_f) \\
i_t &= \sigma_g(W_i * x_t + U_i * h_{t-1} + V_i \odot c_{t-1} + b_i) \\
c_t &= f_t \odot c_{t-1} + i_t \odot \sigma_c(W_c * x_t + U_c * h_{t-1} + b_c) \\
o_t &= \sigma_g(W_o * x_t + U_o * h_{t-1} + V_o \odot c_t + b_o) \\
h_t &= o_t \odot \sigma_h(c_t)
\end{aligned}
$$

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 $\lim_{n\to\infty} W^n = 0$ if the spectral radius of W is smaller than 1.[2][21]

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[22] and train them by connectionist temporal classification (CTC)[23] 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[24] or by policy gradient methods, especially when there is no "teacher" (that is, training labels).

Success

There have been several successful stories of training, in a non-supervised fashion, RNNs with LSTM units.

In 2018, Bill Gates called it a "huge milestone in advancing artificial intelligence" when bots developed by OpenAI were able to beat humans in the game of Dota 2.[11] OpenAI Five consists of five independent but coordinated neural networks. Each network is trained by a policy gradient method without supervising teacher and contains a single-layer, 1024-unit Long-Short-Term-Memory that sees the current game state and emits actions through several possible action heads.[11]

In 2018, OpenAI also trained a similar LSTM by policy gradients to control a human-like robot hand that manipulates physical objects with unprecedented dexterity.[10]

In 2019, DeepMind's program AlphaStar used a deep LSTM core to excel at the complex video game Starcraft II.[12] This was viewed as significant progress towards Artificial General Intelligence.[12]

Applications

Applications of LSTM include:

- Robot control[9]
- Time series prediction[24]

- Speech recognition[25][26][27]
- Rhythm learning[19]
- Hydrological rainfall–runoff modeling[28]
- Music composition[29]
- Grammar learning[30][18][31]
- Handwriting recognition[32][33]
- Human action recognition[34]
- Sign language translation[35]
- Protein homology detection[36]
- Predicting subcellular localization of proteins[37]
- Time series anomaly detection[38]
- Several prediction tasks in the area of business process management[39]
- Prediction in medical care pathways[40]
- Semantic parsing[41]
- Object co-segmentation[42][43]
- Airport passenger management[44]
- Short-term traffic forecast[45]
- Drug design[46]
- Market Prediction[47]
- Activity Classification in Video[48]

Timeline of development

1989: Mike Mozer's work on focused back-propagation[49] will later be cited by the main LSTM paper.[1] Mozer's equation (3.1) anticipates aspects of LSTM cells: c_i(t+1) = d_i c_i(t) + f(x(t)), where c_i(t) is the activation of the i-th self-connected "context unit" at time step t, x(t) is the current input, f is a non-linear function, and d_i is a real-valued "decay weight" that can be learned. The residual connection in the "constant error carousel" of an LSTM cell simplifies this by setting d_i = 1.0: c_i(t+1) = c_i(t) + f(x(t)). The LSTM paper[1] calls this "LSTM's central feature," and states: "Note the similarity to Mozer's fixed time constant system (1992) -- a time constant of 1.0 is appropriate for potentially infinite time lags."

1991: Sepp Hochreiter analyzed the vanishing gradient problem and developed principles of the method in his German diploma thesis,[2] which was called "one of the most important documents in the history of machine learning" by his supervisor Juergen Schmidhuber.[50]

1995: "Long Short-Term Memory (LSTM)" is published in a technical report by Sepp Hochreiter and Jürgen Schmidhuber.[51]

1996: LSTM is published at NIPS'1996, a peer-reviewed conference.[14]

1997: The main LSTM paper is published 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.[52]

1999: Felix Gers, Jürgen Schmidhuber, and Fred Cummins introduced the forget gate (also called "keep gate") into the LSTM architecture,[53] enabling the LSTM to reset its own state.[52]

2000: Gers, Schmidhuber, and Cummins added peephole connections (connections from the cell to the gates) into the architecture.[18][19] Additionally, the output activation function was omitted.[52]

2001: Gers and Schmidhuber trained LSTM to learn languages unlearnable by traditional models such as Hidden Markov Models.[18][54]

Hochreiter et al. used LSTM for meta-learning (i.e. learning a learning algorithm).[55]

2004: First successful application of LSTM to speech Alex Graves et al.[56][54]

2005: First publication (Graves and Schmidhuber) of LSTM with full backpropagation through time and of bi-directional LSTM.[25][54]

2005: Daan Wierstra, Faustino Gomez, and Schmidhuber trained LSTM by neuroevolution without a teacher.[24]

2006: Graves, Fernandez, Gomez, and Schmidhuber introduce a new error function for LSTM: Connectionist Temporal Classification (CTC) for simultaneous alignment and recognition of sequences.[23] CTC-trained LSTM led to breakthroughs in speech recognition.[26][57][58][59]

Mayer et al. trained LSTM to control robots.[9]

2007: Wierstra, Foerster, Peters, and Schmidhuber trained LSTM by policy gradients for reinforcement learning without a teacher.[60]

Hochreiter, Heuesel, and Obermayr applied LSTM to protein homology detection the field of biology.[36]

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.[3] One was the most accurate model in the competition and another was the fastest.[61] This was the first time an RNN won international competitions.[54]

2009: Justin Bayer et al. introduced neural architecture search for LSTM.[62][54]

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.[27]

2014: Kyunghyun Cho et al. put forward a simplified variant of the forget gate LSTM[53] called Gated recurrent unit (GRU).[63]

2015: Google started using an LSTM trained by CTC for speech recognition on Google Voice.[57][58] According to the official blog post, the new model cut transcription errors by 49%.[64]

2015: Rupesh Kumar Srivastava, Klaus Greff, and Schmidhuber used LSTM principles[53] to create the Highway network, a feedforward neural network with hundreds of layers, much deeper than previous networks.[65][66][67] 7 months later, Kaiming He, Xiangyu Zhang; Shaoqing Ren, and Jian Sun won the

ImageNet 2015 competition with an open-gated or gateless Highway network variant called Residual neural network.[68] This has become the most cited neural network of the 21st century.[67]

2016: Google started using an LSTM to suggest messages in the Allo conversation app.[69] In the same year, Google released the Google Neural Machine Translation system for Google Translate which used LSTMs to reduce translation errors by 60%.[6][70][71]

Apple announced in its Worldwide Developers Conference that it would start using the LSTM for quicktype[72][73][74] in the iPhone and for Siri.[75][76]

Amazon released Polly, which generates the voices behind Alexa, using a bidirectional LSTM for the text-to-speech technology.[77]

2017: Facebook performed some 4.5 billion automatic translations every day using long short-term memory networks.[7]

Researchers from Michigan State University, IBM Research, and Cornell University published a study in the Knowledge Discovery and Data Mining (KDD) conference.[78][79][80] Their Time-Aware LSTM (T-LSTM) performs better on certain data sets than standard LSTM.

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".[59]

2018: OpenAI used LSTM trained by policy gradients to beat humans in the complex video game of Dota 2,[11] and to control a human-like robot hand that manipulates physical objects with unprecedented dexterity.[10][54]

2019: DeepMind used LSTM trained by policy gradients to excel at the complex video game of Starcraft II.[12][54]

2021: According to Google Scholar, in 2021, LSTM was cited over 16,000 times within a single year. This reflects applications of LSTM in many different fields including healthcare.[13]

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
- Time aware long short-term memory
- Transformer (machine learning model)
- Time series

References

1. Sepp Hochreiter; Jürgen Schmidhuber (1997). "Long short-term memory" (https://www.researchgate.net/publication/13853244). *Neural Computation*. **9** (8): 1735–1780. doi:10.1162/neco.1997.9.8.1735 (https://doi.org/10.1162%2Fneco.1997.9.8.1735).

PMID 9377276 (https://pubmed.ncbi.nlm.nih.gov/9377276). S2CID 1915014 (https://api.semanticscholar.org/CorpusID:1915014).

2. Hochreiter, Sepp (1991). *Untersuchungen zu dynamischen neuronalen Netzen* (http://www.bioinf.jku.at/publications/older/3804.pdf) (PDF) (diploma thesis). Technical University Munich, Institute of Computer Science.
3. Graves, A.; Liwicki, M.; Fernández, S.; Bertolami, R.; Bunke, H.; Schmidhuber, J. (May 2009). "A Novel Connectionist System for Unconstrained Handwriting Recognition". *IEEE Transactions on Pattern Analysis and Machine Intelligence*. **31** (5): 855–868. CiteSeerX 10.1.1.139.4502 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.139.4502). doi:10.1109/tpami.2008.137 (https://doi.org/10.1109%2Ftpami.2008.137). ISSN 0162-8828 (https://www.worldcat.org/issn/0162-8828). PMID 19299860 (https://pubmed.ncbi.nlm.nih.gov/19299860). S2CID 14635907 (https://api.semanticscholar.org/CorpusID:14635907).
4. Sak, Hasim; Senior, Andrew; Beaufays, Francoise (2014). "Long Short-Term Memory recurrent neural network architectures for large scale acoustic modeling" (https://web.archive.org/web/20180424203806/https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/43905.pdf) (PDF). Archived from the original (https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/43905.pdf) (PDF) on 2018-04-24.
5. Li, Xiangang; Wu, Xihong (2014-10-15). "Constructing Long Short-Term Memory based Deep Recurrent Neural Networks for Large Vocabulary Speech Recognition". arXiv:1410.4281 (https://arxiv.org/abs/1410.4281) [cs.CL (https://arxiv.org/archive/cs.CL)].
6. Wu, Yonghui; Schuster, Mike; Chen, Zhifeng; Le, Quoc V.; Norouzi, Mohammad; Macherey, Wolfgang; Krikun, Maxim; Cao, Yuan; Gao, Qin (2016-09-26). "Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation". arXiv:1609.08144 (https://arxiv.org/abs/1609.08144) [cs.CL (https://arxiv.org/archive/cs.CL)].
7. Ong, Thuy (4 August 2017). "Facebook's translations are now powered completely by AI" (https://www.theverge.com/2017/8/4/16093872/facebook-ai-translations-artificial-intelligence). *www.allthingsdistributed.com*. Retrieved 2019-02-15.
8. Sahidullah, Md; Patino, Jose; Cornell, Samuele; Yin, Ruiking; Sivasankaran, Sunit; Bredin, Herve; Korshunov, Pavel; Brutti, Alessio; Serizel, Romain; Vincent, Emmanuel; Evans, Nicholas; Marcel, Sebastien; Squartini, Stefano; Barras, Claude (2019-11-06). "The Speed Submission to DIHARD II: Contributions & Lessons Learned". arXiv:1911.02388 (https://arxiv.org/abs/1911.02388) [eess.AS (https://arxiv.org/archive/eess.AS)].
9. Mayer, H.; Gomez, F.; Wierstra, D.; Nagy, I.; Knoll, A.; Schmidhuber, J. (October 2006). "A System for Robotic Heart Surgery that Learns to Tie Knots Using Recurrent Neural Networks". *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. pp. 543–548. CiteSeerX 10.1.1.218.3399 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.218.3399). doi:10.1109/IROS.2006.282190 (https://doi.org/10.1109%2FIROS.2006.282190). ISBN 978-1-4244-0258-8. S2CID 12284900 (https://api.semanticscholar.org/CorpusID:12284900).
10. "Learning Dexterity" (https://openai.com/research/learning-dexterity/). *OpenAI*. July 30, 2018. Retrieved 2023-06-28.
11. Rodriguez, Jesus (July 2, 2018). "The Science Behind OpenAI Five that just Produced One of the Greatest Breakthrough in the History of AI" (https://web.archive.org/web/20191226222000/https://towardsdatascience.com/the-science-behind-openai-five-that-just-produced-one-of-the-greatest-breakthrough-in-the-history-b045bcdc2b69?gi=24b20ef8ca3f). *Towards Data Science*. Archived from the original (https://towardsdatascience.com/the-science-behind-openai-five-that-just-produced-one-of-the-greatest-breakthrough-in-the-history-b045bcdc2b69) on 2019-12-26. Retrieved 2019-01-15.

12. Stanford, Stacy (January 25, 2019). "DeepMind's AI, AlphaStar Showcases Significant Progress Towards AGI" (https://medium.com/mlmemoirs/deepminds-ai-alphastar-showcases-significant-progress-towards-agi-93810c94fbe9). *Medium ML Memoirs*. Retrieved 2019-01-15.
13. Schmidhuber, Jürgen (2021). "The 2010s: Our Decade of Deep Learning / Outlook on the 2020s" (https://people.idsia.ch/~juergen/2010s-our-decade-of-deep-learning.html). *AI Blog*. IDSIA, Switzerland. Retrieved 2022-04-30.
14. Hochreiter, Sepp; Schmidhuber, Juergen (1996). *LSTM can solve hard long time lag problems* (https://dl.acm.org/doi/10.5555/2998981.2999048). Advances in Neural Information Processing Systems (https://neurips.cc).
15. Felix A. Gers; Jürgen Schmidhuber; Fred Cummins (2000). "Learning to Forget: Continual Prediction with LSTM". *Neural Computation*. **12** (10): 2451–2471. CiteSeerX 10.1.1.55.5709 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.55.5709). doi:10.1162/089976600300015015 (https://doi.org/10.1162%2F089976600300015015). PMID 11032042 (https://pubmed.ncbi.nlm.nih.gov/11032042). S2CID 11598600 (https://api.semanticscholar.org/CorpusID:11598600).
16. Calin, Ovidiu (14 February 2020). *Deep Learning Architectures*. Cham, Switzerland: Springer Nature. p. 555. ISBN 978-3-030-36720-6.
17. Lakretz, Yair; Kruszewski, German; Desbordes, Theo; Hupkes, Dieuwke; Dehaene, Stanislas; Baroni, Marco (2019), "The emergence of number and syntax units in" (https://aclanthology.org/N19-1002/), *The emergence of number and syntax units* (https://pure.uva.nl/ws/files/49723040/N19_1002.pdf) (PDF), Association for Computational Linguistics, pp. 11–20, doi:10.18653/v1/N19-1002 (https://doi.org/10.18653%2Fv1%2FN19-1002), hdl:11245.1/16cb6800-e10d-4166-8e0b-fed61ca6ebb4 (https://hdl.handle.net/11245.1%2F16cb6800-e10d-4166-8e0b-fed61ca6ebb4), S2CID 81978369 (https://api.semanticscholar.org/CorpusID:81978369)
18. Gers, F. A.; Schmidhuber, J. (2001). "LSTM Recurrent Networks Learn Simple Context Free and Context Sensitive Languages" (ftp://ftp.idsia.ch/pub/juergen/L-IEEE.pdf) (PDF). *IEEE Transactions on Neural Networks*. **12** (6): 1333–1340. doi:10.1109/72.963769 (https://doi.org/10.1109%2F72.963769). PMID 18249962 (https://pubmed.ncbi.nlm.nih.gov/18249962). S2CID 10192330 (https://api.semanticscholar.org/CorpusID:10192330).
19. Gers, F.; Schraudolph, N.; Schmidhuber, J. (2002). "Learning precise timing with LSTM recurrent networks" (http://www.jmlr.org/papers/volume3/gers02a/gers02a.pdf) (PDF). *Journal of Machine Learning Research*. **3**: 115–143.
20. Xingjian Shi; Zhourong Chen; Hao Wang; Dit-Yan Yeung; Wai-kin Wong; Wang-chun Woo (2015). "Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting". *Proceedings of the 28th International Conference on Neural Information Processing Systems*: 802–810. arXiv:1506.04214 (https://arxiv.org/abs/1506.04214). Bibcode:2015arXiv150604214S (https://ui.adsabs.harvard.edu/abs/2015arXiv150604214S).
21. Hochreiter, S.; Bengio, Y.; Frasconi, P.; Schmidhuber, J. (2001). "Gradient Flow in Recurrent Nets: the Difficulty of Learning Long-Term Dependencies (PDF Download Available)" (https://www.researchgate.net/publication/2839938). In Kremer and, S. C.; Kolen, J. F. (eds.). *A Field Guide to Dynamical Recurrent Neural Networks*. IEEE Press.
22. Fernández, Santiago; Graves, Alex; Schmidhuber, Jürgen (2007). "Sequence labelling in structured domains with hierarchical recurrent neural networks". *Proc. 20th Int. Joint Conf. On Artificial Intelligence, Ijcai 2007*: 774–779. CiteSeerX 10.1.1.79.1887 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.79.1887).

23. Graves, Alex; Fernández, Santiago; Gomez, Faustino; Schmidhuber, Jürgen (2006). "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks". *In Proceedings of the International Conference on Machine Learning, ICML 2006*: 369–376. CiteSeerX 10.1.1.75.6306 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.75.6306).
24. Wierstra, Daan; Schmidhuber, J.; Gomez, F. J. (2005). "Evolino: Hybrid Neuroevolution/Optimal Linear Search for Sequence Learning" (https://www.academia.edu/5830256). *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI), Edinburgh*: 853–858.
25. Graves, A.; Schmidhuber, J. (2005). "Framewise phoneme classification with bidirectional LSTM and other neural network architectures". *Neural Networks*. **18** (5–6): 602–610. CiteSeerX 10.1.1.331.5800 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.331.5800). doi:10.1016/j.neunet.2005.06.042 (https://doi.org/10.1016%2Fj.neunet.2005.06.042). PMID 16112549 (https://pubmed.ncbi.nlm.nih.gov/16112549). S2CID 1856462 (https://api.semanticscholar.org/CorpusID:1856462).
26. Fernández, S.; Graves, A.; Schmidhuber, J. (9 September 2007). "An Application of Recurrent Neural Networks to Discriminative Keyword Spotting" (http://dl.acm.org/citation.cfm?id=1778066.1778092). *Proceedings of the 17th International Conference on Artificial Neural Networks*. ICANN'07. Berlin, Heidelberg: Springer-Verlag: 220–229. ISBN 978-3540746935. Retrieved 28 December 2023.
27. Graves, Alex; Mohamed, Abdel-rahman; Hinton, Geoffrey (2013). "Speech recognition with deep recurrent neural networks". *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. pp. 6645–6649. arXiv:1303.5778 (https://arxiv.org/abs/1303.5778). doi:10.1109/ICASSP.2013.6638947 (https://doi.org/10.1109%2FICASSP.2013.6638947). ISBN 978-1-4799-0356-6. S2CID 206741496 (https://api.semanticscholar.org/CorpusID:206741496).
28. Kratzert, Frederik; Klotz, Daniel; Shalev, Guy; Klambauer, Günter; Hochreiter, Sepp; Nearing, Grey (2019-12-17). "Towards learning universal, regional, and local hydrological behaviors via machine learning applied to large-sample datasets" (https://hess.copernicus.org/articles/23/5089/2019/). *Hydrology and Earth System Sciences*. **23** (12): 5089–5110. arXiv:1907.08456 (https://arxiv.org/abs/1907.08456). Bibcode:2019HESS...23.5089K (https://ui.adsabs.harvard.edu/abs/2019HESS...23.5089K). doi:10.5194/hess-23-5089-2019 (https://doi.org/10.5194%2Fhess-23-5089-2019). ISSN 1027-5606 (https://www.worldcat.org/issn/1027-5606).
29. Eck, Douglas; Schmidhuber, Jürgen (2002-08-28). "Learning the Long-Term Structure of the Blues". *Artificial Neural Networks — ICANN 2002*. Lecture Notes in Computer Science. Vol. 2415. Springer, Berlin, Heidelberg. pp. 284–289. CiteSeerX 10.1.1.116.3620 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.116.3620). doi:10.1007/3-540-46084-5_47 (https://doi.org/10.1007%2F3-540-46084-5_47). ISBN 978-3540460848.
30. Schmidhuber, J.; Gers, F.; Eck, D.; Schmidhuber, J.; Gers, F. (2002). "Learning nonregular languages: A comparison of simple recurrent networks and LSTM". *Neural Computation*. **14** (9): 2039–2041. CiteSeerX 10.1.1.11.7369 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.11.7369). doi:10.1162/089976602320263980 (https://doi.org/10.1162%2F089976602320263980). PMID 12184841 (https://pubmed.ncbi.nlm.nih.gov/12184841). S2CID 30459046 (https://api.semanticscholar.org/CorpusID:30459046).
31. Perez-Ortiz, J. A.; Gers, F. A.; Eck, D.; Schmidhuber, J. (2003). "Kalman filters improve LSTM network performance in problems unsolvable by traditional recurrent nets". *Neural Networks*. **16** (2): 241–250. CiteSeerX 10.1.1.381.1992 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.381.1992). doi:10.1016/s0893-6080(02)00219-8 (https://doi.org/10.1016%2Fs0893-6080%2802%2900219-8). PMID 12628609 (https://pubmed.ncbi.nlm.nih.gov/12628609).

32. A. Graves, J. Schmidhuber. Offline Handwriting Recognition with Multidimensional Recurrent Neural Networks. Advances in Neural Information Processing Systems 22, NIPS'22, pp 545–552, Vancouver, MIT Press, 2009.
33. Graves, A.; Fernández, S.; Liwicki, M.; Bunke, H.; Schmidhuber, J. (3 December 2007). "Unconstrained Online Handwriting Recognition with Recurrent Neural Networks" (http://dl.acm.org/citation.cfm?id=2981562.2981635). *Proceedings of the 20th International Conference on Neural Information Processing Systems*. NIPS'07. USA: Curran Associates Inc.: 577–584. ISBN 9781605603520. Retrieved 28 December 2023.
34. Baccouche, M.; Mamalet, F.; Wolf, C.; Garcia, C.; Baskurt, A. (2011). "Sequential Deep Learning for Human Action Recognition". In Salah, A. A.; Lepri, B. (eds.). *2nd International Workshop on Human Behavior Understanding (HBU)*. Lecture Notes in Computer Science. Vol. 7065. Amsterdam, Netherlands: Springer. pp. 29–39. doi:10.1007/978-3-642-25446-8_4 (https://doi.org/10.1007%2F978-3-642-25446-8_4). ISBN 978-3-642-25445-1.
35. Huang, Jie; Zhou, Wengang; Zhang, Qilin; Li, Houqiang; Li, Weiping (2018-01-30). "Video-based Sign Language Recognition without Temporal Segmentation". arXiv:1801.10111 (https://arxiv.org/abs/1801.10111) [cs.CV (https://arxiv.org/archive/cs.CV)].
36. Hochreiter, S.; Heusel, M.; Obermayer, K. (2007). "Fast model-based protein homology detection without alignment" (https://doi.org/10.1093%2Fbioinformatics%2Fbtm247). *Bioinformatics*. **23** (14): 1728–1736. doi:10.1093/bioinformatics/btm247 (https://doi.org/10.1093%2Fbioinformatics%2Fbtm247). PMID 17488755 (https://pubmed.ncbi.nlm.nih.gov/17488755).
37. Thireou, T.; Reczko, M. (2007). "Bidirectional Long Short-Term Memory Networks for predicting the subcellular localization of eukaryotic proteins". *IEEE/ACM Transactions on Computational Biology and Bioinformatics*. **4** (3): 441–446. doi:10.1109/tcbb.2007.1015 (https://doi.org/10.1109%2Ftcbb.2007.1015). PMID 17666763 (https://pubmed.ncbi.nlm.nih.gov/17666763). S2CID 11787259 (https://api.semanticscholar.org/CorpusID:11787259).
38. Malhotra, Pankaj; Vig, Lovekesh; Shroff, Gautam; Agarwal, Puneet (April 2015). "Long Short Term Memory Networks for Anomaly Detection in Time Series" (https://web.archive.org/web/20201030224634/https://www.elen.ucl.ac.be/Proceedings/esann/esannpdf/es2015-56.pdf) (PDF). *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning — ESANN 2015*. Archived from the original (https://www.elen.ucl.ac.be/Proceedings/esann/esannpdf/es2015-56.pdf) (PDF) on 2020-10-30. Retrieved 2018-02-21.
39. Tax, N.; Verenich, I.; La Rosa, M.; Dumas, M. (2017). "Predictive Business Process Monitoring with LSTM Neural Networks". *Advanced Information Systems Engineering*. Lecture Notes in Computer Science. Vol. 10253. pp. 477–492. arXiv:1612.02130 (https://arxiv.org/abs/1612.02130). doi:10.1007/978-3-319-59536-8_30 (https://doi.org/10.1007%2F978-3-319-59536-8_30). ISBN 978-3-319-59535-1. S2CID 2192354 (https://api.semanticscholar.org/CorpusID:2192354).
40. Choi, E.; Bahadori, M.T.; Schuetz, E.; Stewart, W.; Sun, J. (2016). "Doctor AI: Predicting Clinical Events via Recurrent Neural Networks" (http://proceedings.mlr.press/v56/Choi16.html). *JMLR Workshop and Conference Proceedings*. **56**: 301–318. arXiv:1511.05942 (https://arxiv.org/abs/1511.05942). Bibcode:2015arXiv151105942C (https://ui.adsabs.harvard.edu/abs/2015arXiv151105942C). PMC 5341604 (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5341604). PMID 28286600 (https://pubmed.ncbi.nlm.nih.gov/28286600).
41. Jia, Robin; Liang, Percy (2016). "Data Recombination for Neural Semantic Parsing". arXiv:1606.03622 (https://arxiv.org/abs/1606.03622) [cs.CL (https://arxiv.org/archive/cs.CL)].

42. Wang, Le; Duan, Xuhuan; Zhang, Qilin; Niu, Zhenxing; Hua, Gang; Zheng, Nanning (2018-05-22). "Segment-Tube: Spatio-Temporal Action Localization in Untrimmed Videos with Per-Frame Segmentation" (https://qilin-zhang.github.io/_pages/pdfs/Segment-Tube_Spatio-Temporal_Action_Localization_in_Untrimmed_Videos_with_Per-Frame_Segmentation.pdf) (PDF). *Sensors*. **18** (5): 1657. Bibcode:2018Senso..18.1657W (https://ui.adsabs.harvard.edu/abs/2018Senso..18.1657W). doi:10.3390/s18051657 (https://doi.org/10.3390%2Fs18051657). ISSN 1424-8220 (https://www.worldcat.org/issn/1424-8220). PMC 5982167 (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5982167). PMID 29789447 (https://pubmed.ncbi.nlm.nih.gov/29789447).
43. Duan, Xuhuan; Wang, Le; Zhai, Changbo; Zheng, Nanning; Zhang, Qilin; Niu, Zhenxing; Hua, Gang (2018). "Joint Spatio-Temporal Action Localization in Untrimmed Videos with Per-Frame Segmentation". *2018 25th IEEE International Conference on Image Processing (ICIP)*. 25th IEEE International Conference on Image Processing (ICIP). pp. 918–922. doi:10.1109/icip.2018.8451692 (https://doi.org/10.1109%2Ficip.2018.8451692). ISBN 978-1-4799-7061-2.
44. Orsini, F.; Gastaldi, M.; Mantecchini, L.; Rossi, R. (2019). *Neural networks trained with WiFi traces to predict airport passenger behavior*. 6th International Conference on Models and Technologies for Intelligent Transportation Systems. Krakow: IEEE. arXiv:1910.14026 (https://arxiv.org/abs/1910.14026). doi:10.1109/MTITS.2019.8883365 (https://doi.org/10.1109%2FMTITS.2019.8883365). 8883365.
45. Zhao, Z.; Chen, W.; Wu, X.; Chen, P.C.Y.; Liu, J. (2017). "LSTM network: A deep learning approach for Short-term traffic forecast". *IET Intelligent Transport Systems*. **11** (2): 68–75. doi:10.1049/iet-its.2016.0208 (https://doi.org/10.1049%2Fiet-its.2016.0208). S2CID 114567527 (https://api.semanticscholar.org/CorpusID:114567527).
46. Gupta A, Müller AT, Huisman BJH, Fuchs JA, Schneider P, Schneider G (2018). "Generative Recurrent Networks for De Novo Drug Design" (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5836943). *Mol Inform*. **37** (1–2). doi:10.1002/minf.201700111 (https://doi.org/10.1002%2Fminf.201700111). PMC 5836943 (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5836943). PMID 29095571 (https://pubmed.ncbi.nlm.nih.gov/29095571).
47. Saiful Islam, Md.; Hossain, Emam (2020-10-26). "Foreign Exchange Currency Rate Prediction using a GRU-LSTM Hybrid Network" (https://doi.org/10.1016%2Fj.socl.2020.100009). *Soft Computing Letters*. **3**: 100009. doi:10.1016/j.socl.2020.100009 (https://doi.org/10.1016%2Fj.socl.2020.100009). ISSN 2666-2221 (https://www.worldcat.org/issn/2666-2221).
48. {{Cite Abbey Martin, Andrew J. Hill, Konstantin M. Seiler & Mehala Balamurali (2023) Automatic excavator action recognition and localisation for untrimmed video using hybrid LSTM-Transformer networks, International Journal of Mining, Reclamation and Environment, DOI: 10.1080/17480930.2023.2290364}}
49. Mozer, Mike (1989). "A Focused Backpropagation Algorithm for Temporal Pattern Recognition". *Complex Systems*.
50. Schmidhuber, Juergen (2022). "Annotated History of Modern AI and Deep Learning". arXiv:2212.11279 (https://arxiv.org/abs/2212.11279) [cs.NE (https://arxiv.org/archive/cs.NE)].
51. Sepp Hochreiter; Jürgen Schmidhuber (21 August 1995), *Long Short Term Memory* (ftp://ftp.idsia.ch/pub/juergen/fki-207-95.ps.gz), Wikidata Q98967430
52. Klaus Greff; Rupesh Kumar Srivastava; Jan Koutník; Bas R. Steunebrink; Jürgen Schmidhuber (2015). "LSTM: A Search Space Odyssey". *IEEE Transactions on Neural Networks and Learning Systems*. **28** (10): 2222–2232. arXiv:1503.04069 (https://arxiv.org/abs/1503.04069). Bibcode:2015arXiv150304069G (https://ui.adsabs.harvard.edu/abs/2015arXiv150304069G). doi:10.1109/TNNLS.2016.2582924 (https://doi.org/10.1109%2FTNNLS.2016.2582924). PMID 27411231 (https://pubmed.ncbi.nlm.nih.gov/27411231). S2CID 3356463 (https://api.semanticscholar.org/CorpusID:3356463).

53. Gers, Felix; Schmidhuber, Jürgen; Cummins, Fred (1999). "Learning to forget: Continual prediction with LSTM". *9th International Conference on Artificial Neural Networks: ICANN '99*. Vol. 1999. pp. 850–855. doi:10.1049/cp:19991218 (https://doi.org/10.1049%2Fcp%3A19991218). ISBN 0-85296-721-7.
54. Schmidhuber, Juergen (10 May 2021). "Deep Learning: Our Miraculous Year 1990-1991". arXiv:2005.05744 (https://arxiv.org/abs/2005.05744) [cs.NE (https://arxiv.org/archive/cs.NE)].
55. Hochreiter, S.; Younger, A. S.; Conwell, P. R. (2001). "Learning to Learn Using Gradient Descent". *Artificial Neural Networks — ICANN 2001* (http://www.bioinf.jku.at/publications/older/1504.pdf) (PDF). Lecture Notes in Computer Science. Vol. 2130. pp. 87–94. CiteSeerX 10.1.1.5.323 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.5.323). doi:10.1007/3-540-44668-0_13 (https://doi.org/10.1007%2F3-540-44668-0_13). ISBN 978-3-540-42486-4. ISSN 0302-9743 (https://www.worldcat.org/issn/0302-9743). S2CID 52872549 (https://api.semanticscholar.org/CorpusID:52872549).
56. Graves, Alex; Beringer, Nicole; Eck, Douglas; Schmidhuber, Juergen (2004). *Biologically Plausible Speech Recognition with LSTM Neural Nets*. Workshop on Biologically Inspired Approaches to Advanced Information Technology, Bio-ADIT 2004, Lausanne, Switzerland. pp. 175–184.
57. Beaufays, Françoise (August 11, 2015). "The neural networks behind Google Voice transcription" (http://googleresearch.blogspot.co.at/2015/08/the-neural-networks-behind-google-voice.html). *Research Blog*. Retrieved 2017-06-27.
58. Sak, Haşim; Senior, Andrew; Rao, Kanishka; Beaufays, Françoise; Schalkwyk, Johan (September 24, 2015). "Google voice search: faster and more accurate" (http://googleresearch.blogspot.co.uk/2015/09/google-voice-search-faster-and-more.html). *Research Blog*. Retrieved 2017-06-27.
59. Haridy, Rich (August 21, 2017). "Microsoft's speech recognition system is now as good as a human" (http://newatlas.com/microsoft-speech-recognition-equals-humans/50999). *newatlas.com*. Retrieved 2017-08-27.
60. Wierstra, Daan; Foerster, Alexander; Peters, Jan; Schmidhuber, Juergen (2005). "Solving Deep Memory POMDPs with Recurrent Policy Gradients" (https://people.idsia.ch/~juergen/lstm-policy-gradient-2010.html). *International Conference on Artificial Neural Networks ICANN'07*.
61. Märgner, Volker; Abed, Haikal El (July 2009). "ICDAR 2009 Arabic Handwriting Recognition Competition". *2009 10th International Conference on Document Analysis and Recognition*. pp. 1383–1387. doi:10.1109/ICDAR.2009.256 (https://doi.org/10.1109%2FICDAR.2009.256). ISBN 978-1-4244-4500-4. S2CID 52851337 (https://api.semanticscholar.org/CorpusID:52851337).
62. Bayer, Justin; Wierstra, Daan; Togelius, Julian; Schmidhuber, Juergen (2009). "Evolving memory cell structures for sequence learning". *International Conference on Artificial Neural Networks ICANN'09, Cyprus*.
63. Cho, Kyunghyun; van Merrienboer, Bart; Gulcehre, Caglar; Bahdanau, Dzmitry; Bougares, Fethi; Schwenk, Holger; Bengio, Yoshua (2014). "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation". arXiv:1406.1078 (https://arxiv.org/abs/1406.1078) [cs.CL (https://arxiv.org/archive/cs.CL)].
64. "Neon prescription... or rather, New transcription for Google Voice" (https://googleblog.blogspot.com/2015/07/neon-prescription-or-rather-new.html). *Official Google Blog*. 23 July 2015. Retrieved 2020-04-25.
65. Srivastava, Rupesh Kumar; Greff, Klaus; Schmidhuber, Jürgen (2 May 2015). "Highway Networks". arXiv:1505.00387 (https://arxiv.org/abs/1505.00387) [cs.LG (https://arxiv.org/archive/cs.LG)].

66. Srivastava, Rupesh K; Greff, Klaus; Schmidhuber, Juergen (2015). "Training Very Deep Networks" (http://papers.nips.cc/paper/5850-training-very-deep-networks). *Advances in Neural Information Processing Systems*. **28**. Curran Associates, Inc.: 2377–2385.
67. Schmidhuber, Jürgen (2021). "The most cited neural networks all build on work done in my labs" (https://people.idsia.ch/~juergen/most-cited-neural-nets.html). *AI Blog*. IDSIA, Switzerland. Retrieved 2022-04-30.
68. He, Kaiming; Zhang, Xiangyu; Ren, Shaoqing; Sun, Jian (2016). *Deep Residual Learning for Image Recognition* (https://ieeexplore.ieee.org/document/7780459). *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, NV, USA: IEEE. pp. 770–778. arXiv:1512.03385 (https://arxiv.org/abs/1512.03385). doi:10.1109/CVPR.2016.90 (https://doi.org/10.1109%2FCVPR.2016.90). ISBN 978-1-4673-8851-1.
69. Khaitan, Pranav (May 18, 2016). "Chat Smarter with Allo" (http://googleresearch.blogspot.co.at/2016/05/chat-smarter-with-allo.html). *Research Blog*. Retrieved 2017-06-27.
70. Metz, Cade (September 27, 2016). "An Infusion of AI Makes Google Translate More Powerful Than Ever | WIRED" (https://www.wired.com/2016/09/google-claims-ai-breakthrough-machine-translation/). *Wired*. Retrieved 2017-06-27.
71. "A Neural Network for Machine Translation, at Production Scale" (http://ai.googleblog.com/2016/09/a-neural-network-for-machine.html). *Google AI Blog*. 27 September 2016. Retrieved 2020-04-25.
72. Efrati, Amir (June 13, 2016). "Apple's Machines Can Learn Too" (https://www.theinformation.com/apples-machines-can-learn-too). *The Information*. Retrieved 2017-06-27.
73. Ranger, Steve (June 14, 2016). "iPhone, AI and big data: Here's how Apple plans to protect your privacy | ZDNet" (https://www.zdnet.com/article/ai-big-data-and-the-iphone-heres-how-apple-plans-to-protect-your-privacy). *ZDNet*. Retrieved 2017-06-27.
74. "Can Global Semantic Context Improve Neural Language Models? – Apple" (https://machinelearning.apple.com/2018/09/27/can-global-semantic-context-improve-neural-language-models.html). *Apple Machine Learning Journal*. Retrieved 2020-04-30.
75. Smith, Chris (2016-06-13). "iOS 10: Siri now works in third-party apps, comes with extra AI features" (http://bgr.com/2016/06/13/ios-10-siri-third-party-apps/). *BGR*. Retrieved 2017-06-27.
76. Capes, Tim; Coles, Paul; Conkie, Alistair; Golipour, Ladan; Hadjitarkhani, Abie; Hu, Qiong; Huddleston, Nancy; Hunt, Melvyn; Li, Jiangchuan; Neeracher, Matthias; Prahallad, Kishore (2017-08-20). "Siri On-Device Deep Learning-Guided Unit Selection Text-to-Speech System" (http://www.isca-speech.org/archive/Interspeech_2017/abstracts/1798.html). *Interspeech 2017*. ISCA: 4011–4015. doi:10.21437/Interspeech.2017-1798 (https://doi.org/10.21437%2FInterspeech.2017-1798).
77. Vogels, Werner (30 November 2016). "Bringing the Magic of Amazon AI and Alexa to Apps on AWS. – All Things Distributed" (http://www.allthingsdistributed.com/2016/11/amazon-ai-and-alexa-for-all-aws-apps.html). *www.allthingsdistributed.com*. Retrieved 2017-06-27.
78. "Patient Subtyping via Time-Aware LSTM Networks" (http://biometrics.cse.msu.edu/Publications/MachineLearning/Baytasetal_PatientSubtypingViaTimeAwareLSTMNetworks.pdf) (PDF). *msu.edu*. Retrieved 21 Nov 2018.
79. "Patient Subtyping via Time-Aware LSTM Networks" (http://www.kdd.org/kdd2017/papers/view/patient-subtyping-via-time-aware-lstm-networks). *Kdd.org*. Retrieved 24 May 2018.
80. "SIGKDD" (http://www.kdd.org). *Kdd.org*. Retrieved 24 May 2018.

[1]

Further reading

- Monner, Derek D.; Reggia, James A. (2010). "A generalized LSTM-like training algorithm for second-order recurrent neural networks" (http://www.cs.umd.edu/~dmonner/papers/nn2012.pdf) (PDF). *Neural Networks*. **25** (1): 70–83. doi:10.1016/j.neunet.2011.07.003 (https://doi.org/10.1016%2Fj.neunet.2011.07.003). PMC 3217173 (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3217173). PMID 21803542 (https://pubmed.ncbi.nlm.nih.gov/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" (http://www.jmlr.org/papers/volume3/gers02a/gers02a.pdf) (PDF). *Journal of Machine Learning Research*. **3**: 115–143.
- Gers, Felix (2001). "Long Short-Term Memory in Recurrent Neural Networks" (http://www.felixgers.de/papers/phd.pdf) (PDF). *PhD thesis*.
- Abidogun, Olusola Adeniyi (2005). *Data Mining, Fraud Detection and Mobile Telecommunications: Call Pattern Analysis with Unsupervised Neural Networks* (http://etd.uwc.ac.za/xmlui/handle/11394/249). *Master's Thesis* (Thesis). University of the Western Cape. hdl:11394/249 (https://hdl.handle.net/11394%2F249). Archived (https://web.archive.org/web/20120522234026/http://etd.uwc.ac.za/usrfiles/modules/etd/docs/etd_init_3937_1174040706.pdf) (PDF) from the original on May 22, 2012.
  - original (http://etd.uwc.ac.za/bitstream/handle/11394/249/Abidogun_MSC_2005.pdf) with two chapters devoted to explaining recurrent neural networks, especially LSTM.

External links

- Recurrent Neural Networks (http://www.idsia.ch/~juergen/rnn.html) with over 30 LSTM papers by Jürgen Schmidhuber's group at IDSIA
- Dolphin, R (12 November 2021). "LSTM Networks – A Detailed Explanation" (https://towardsdatascience.com/lstm-networks-a-detailed-explanation-8fae6aefc7f9). *Article*.
- Herta, Christian. "How to implement LSTM in Python with Theano" (http://christianherta.de/lehre/dataScience/machineLearning/neuralNetworks/LSTM.html). *Tutorial*.

1. Abbey Martin, Andrew J. Hill, Konstantin M. Seiler & Mehala Balamurali (2023) Automatic excavator action recognition and localisation for untrimmed video using hybrid LSTM-Transformer networks, International Journal of Mining, Reclamation and Environment, DOI: 10.1080/17480930.2023.2290364