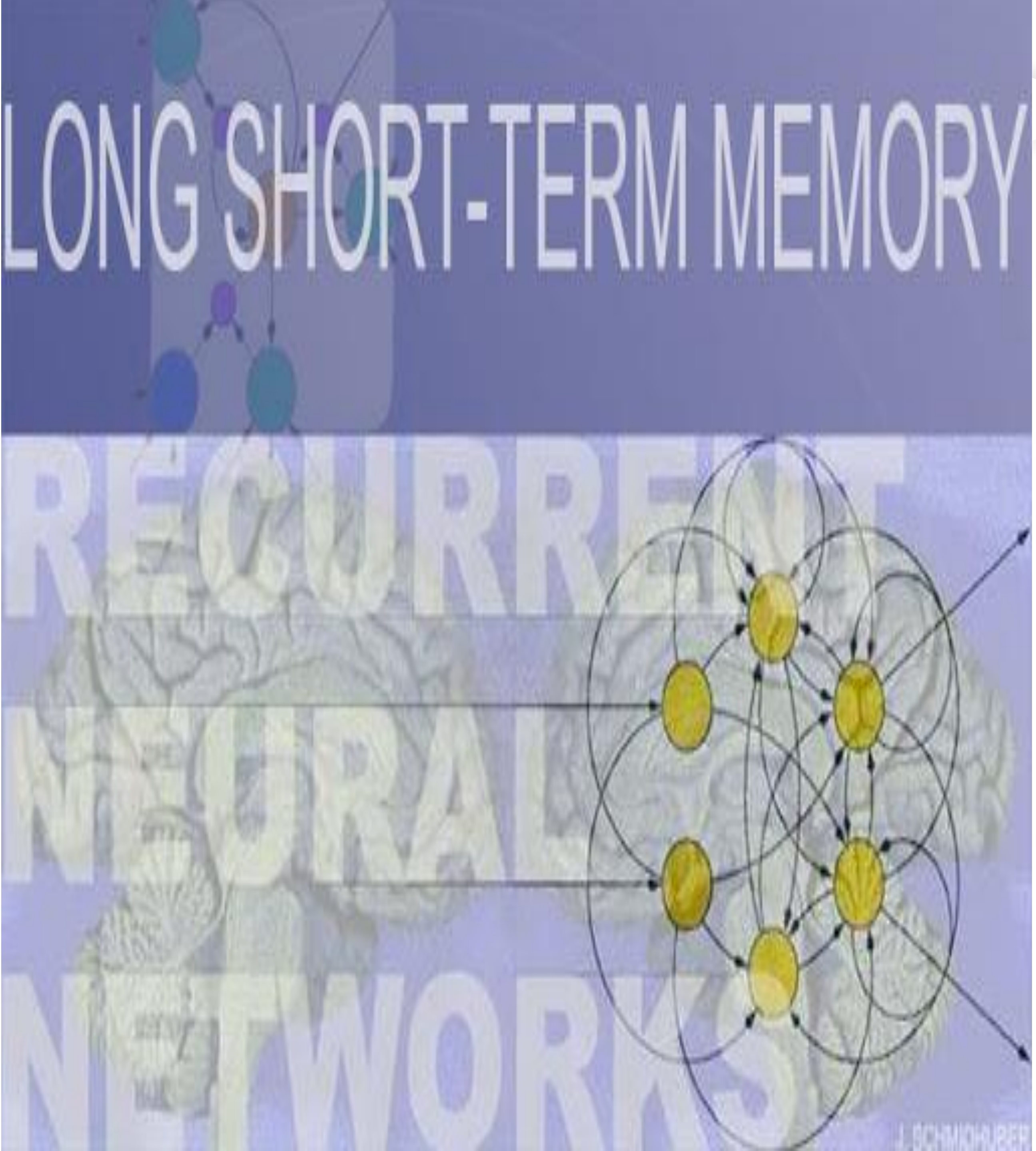


Recurrent Neural Network and LSTMs

Neural Network Design Seminar Presentation

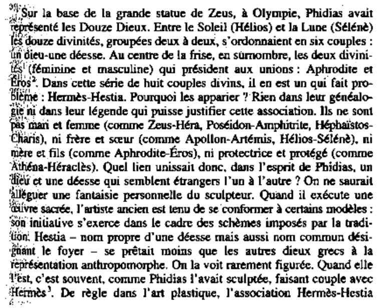
Supervisor:
Dr. Montazer

Student:

Rasoul Norouzi

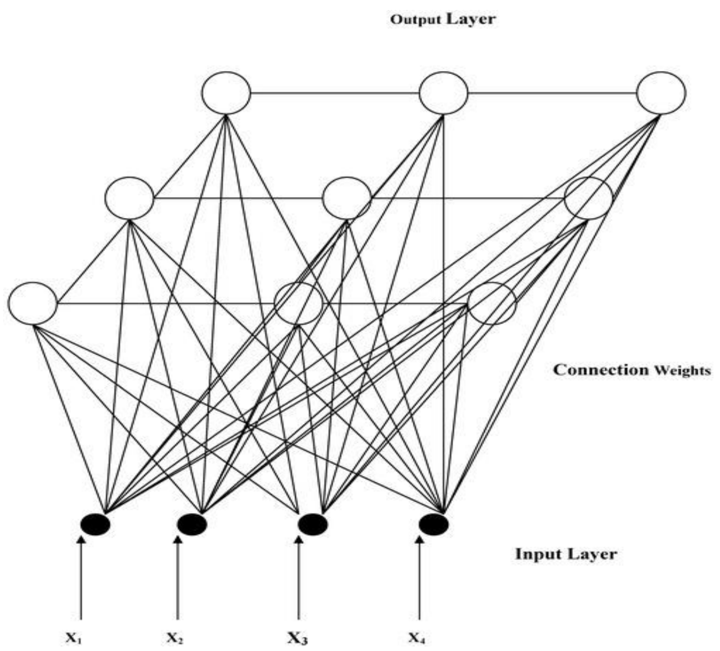
Why Recurrent Neural Network?

- In feed-forward networks, data is assumed to be independent.
- Many kind of data in the real world are not independent of each other and are like an interconnected chain.
- Feed-forward networks are not suitable for sequential data where each data's current value depends on previous data.
- Data such as stock indexes, the letters of the alphabet in words, and the words themselves in the larger set that make up sentences, audio data, frames of a video, and so on.


Sur la base de la grande statue de Zeus, à Olympie, Phidias avait représenté les Douze Dieux. Entre le Soleil (Hélios) et la Lune (Sélénè) les douze divinités, groupées deux à deux, s'ordonnaient en six couples : un dieu-une déesse. Au centre de la frise, en surnombre, les deux divinités (féminine et masculine) qui président aux unions : Aphrodite et Éros[2]. Dans cette série de huit couples divins, il en est un qui fait problème : Hermès-Hestia. Pourquoi les apparier ? Rien dans leur généalogie ni dans leur légende qui puisse justifier cette association. Ils ne sont pas mari et femme (comme Zeus-Héra, Poséidon-Amphitrite, Héphaïstos-Charis), ni frère et sœur (comme Apollon-Artémis, Hélios-Sélénè), ni mère et fils (comme Aphrodite-Éros), ni protectrice et protégé (comme Athéna-Héraclès). Quel lien unissait donc, dans l'esprit de Phidias, un dieu et une déesse qui semblent étrangers l'un à l'autre ? On ne saurait alléguer une fantaisie personnelle du sculpteur. Quand il exécute une œuvre sacrée, l'artiste ancien est tenu de se conformer à certains modèles : son initiative s'exerce dans le cadre des schèmes imposés par la tradition. Hestia – nom propre d'une déesse mais aussi nom commun désignant le foyer – se prêtait moins que les autres dieux grecs à la représentation anthropomorphe. On la voit rarement figurée. Quand elle l'est, c'est souvent, comme Phidias l'avait sculptée, faisant couple avec Hermès[3]. De règle dans l'art plastique, l'association Hermès-Hestia

History of Recurrent Neural Networks

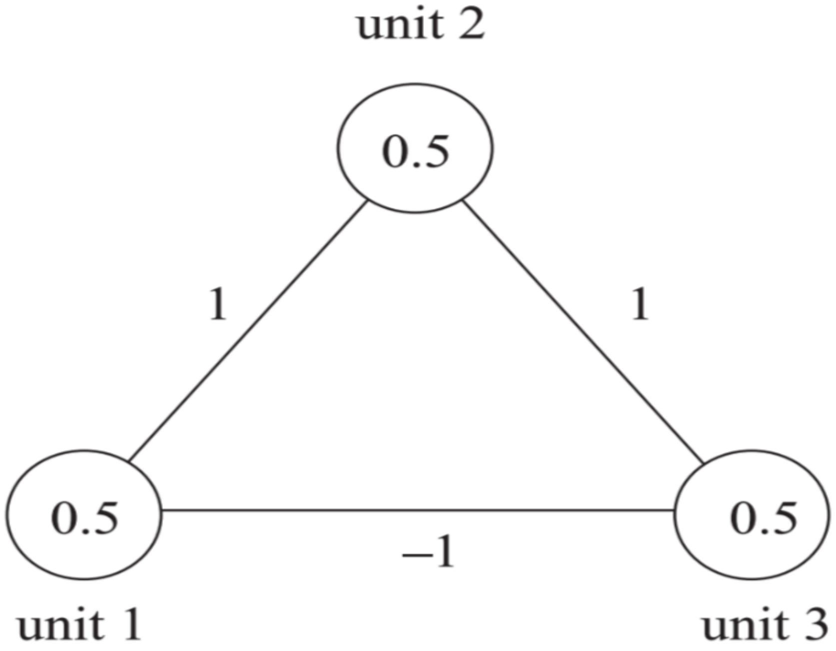
+ The first recurrent neural network was introduced in 1977 as the fully recurrent neural network by Anderson and Cohen.



(Yuan et al., 2009)

History of Recurrent Neural Networks

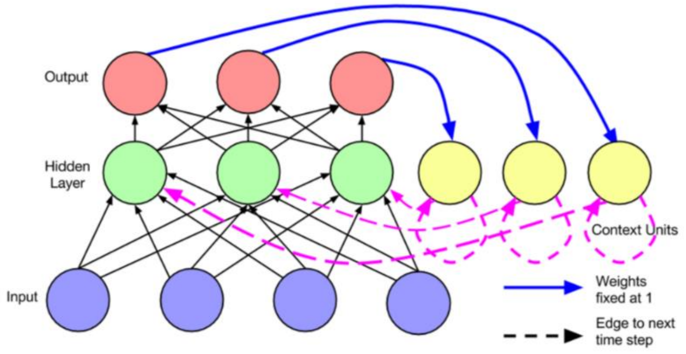
- In 1982, the Hopfield network was introduced by John Hopfield.



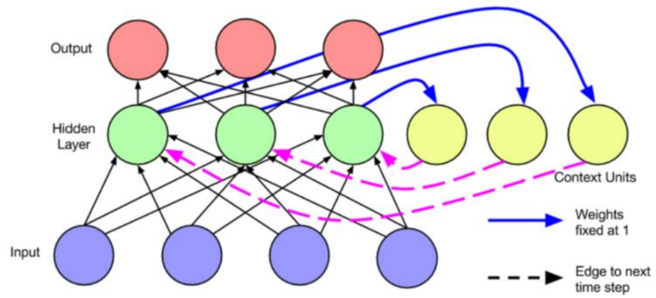
(Rojas, 1996)

History of Recurrent Neural Networks

- Michael Jordan invented the Jordan Network in 1986. Like other networks, it was a feed network that had a hidden layer and a special layer called the state unit.
- The architecture proposed by Elman in 1990 was simpler than Jordan's original architecture. Corresponding to each neuron in the hidden layer there is a context unit.

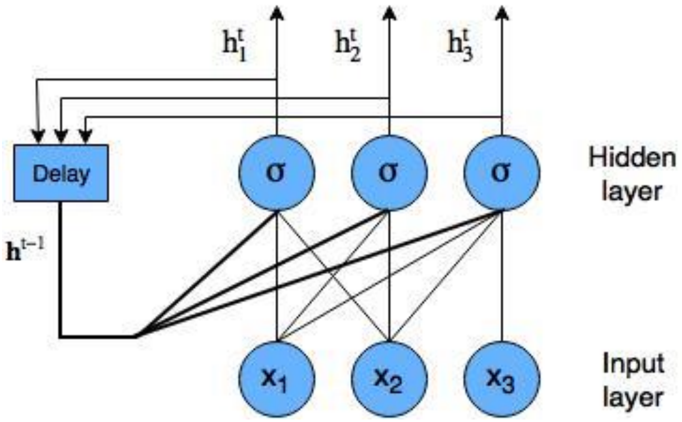


Jordan's network, with recurrent edges of output neurons

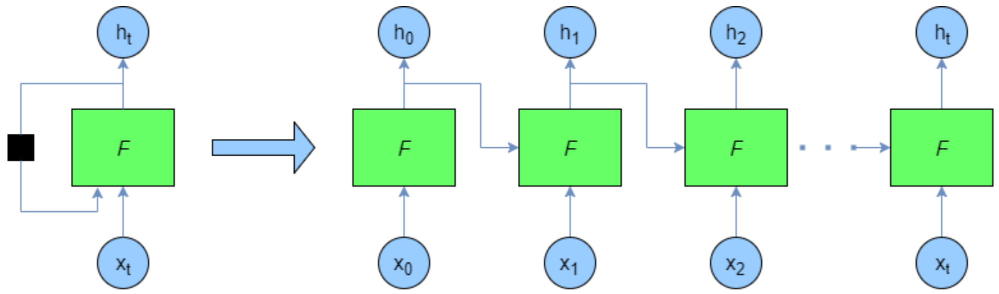


Elman Network , with recurrent edges of context layer's neurons

Vanilla Recurrent Neural Network Mechanism and Architecture



Vanilla recurrent neural network with three input neurons



An unrolled recurrent neural network

$$z_t = w^{xh} x_{t-1} + w^{hh} h_{t-1} + b_h$$

Hidden layer output :$h_t = tahnh(z_t)$

$$y_t = w^{hy} h_t + b_y$$

Expected output :$p_t = softmax(yi)$

$$\hat{y}_k = \frac{e^{a_k}}{\sum_{k'=1}^{K} e^{a_{k'}}} \text{ for } k = 1 \text{ to } k = K.$$

Backpropagation Through the Time (BPT)

loss function (cross entropy)

$$C = -\sum_{p}^{n}\sum_{k}^{o} d_{pk}\ln y_{pk} + (1-d_{pk})\ln(1-y_{pk})$$

Cost function

$$\mathbf{J_t} = \text{crossentropy}(\mathbf{p_t}, \mathbf{labels_t})$$

Output error respect to current time step

$$\delta_o = -\frac{\partial J_t}{\partial_{(p_t)}}\frac{\partial_{(p_t)}}{\partial_{(y_t)}}$$

hidden state error in the current time step

$$\delta_h = -\sum \frac{\partial J_t}{\partial p_t}\frac{\partial p_t}{\partial y_t}\frac{\partial y_t}{\partial h_t}\frac{\partial h_t}{\partial z_t}$$

calculate hidden state error in one backward step

$$\delta_h(\mathrm{t}-1) = -\sum \delta_h(t) * w_t^{hh} * \acute{f}(h(t-1))$$

Calculate the amount of error with respect to the weight of the hidden layer to the output

$$\frac{\boldsymbol{\partial J_t}}{\boldsymbol{\partial w^{\mathbf{hy}}}} = \boldsymbol{\Sigma}\frac{\boldsymbol{\partial J_t}}{\boldsymbol{\partial w^{\mathbf{hy}}}} = \sum \frac{\boldsymbol{\partial J_t}}{\boldsymbol{\partial P_t}}\frac{\boldsymbol{\partial P_t}}{\boldsymbol{\partial y_t}}\frac{\boldsymbol{\partial y_t}}{\boldsymbol{W^{\mathbf{hy}}}}$$

calculate the amount of error with respect to the weight of the hidden layer to the hidden layer

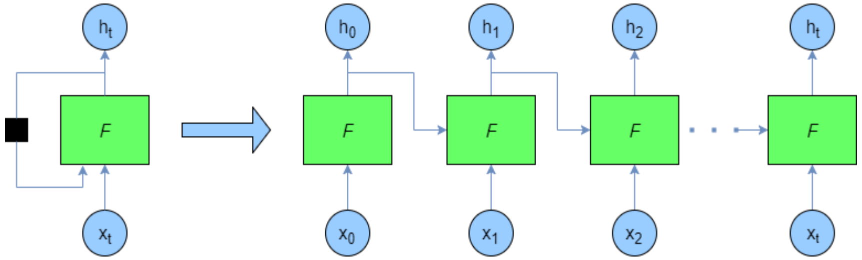
$$\frac{\partial \mathbf{J_t}}{\partial \mathbf{W^{hh}}} = \sum_{k=0}^{t}\frac{\partial \mathbf{J_t}}{\partial \mathbf{h_t}}\frac{\partial \mathbf{h_t}}{\partial \mathbf{h_k}}\frac{\partial \mathbf{h_k}}{\partial \mathbf{z_k}}\frac{\partial \mathbf{z_k}}{\partial \mathbf{W^{hh}}}$$

Calculate the amount of error with respect to the weight of the input neuron to the hidden layer neuron

$$\frac{\partial \mathbf{J_t}}{\partial \mathbf{W^{xh}}} = \sum_{k=0}^{t}\frac{\partial \mathbf{J_t}}{\partial \mathbf{h_t}}\frac{\partial \mathbf{h_t}}{\partial \mathbf{h_k}}\frac{\partial \mathbf{h_k}}{\partial \mathbf{z_k}}\frac{\partial \mathbf{z_k}}{\partial \mathbf{W^{xh}}}$$

Example of calculating the error changes in the third time step relative to the model parameters

$$\frac{\partial \mathbf{J_3}}{\partial \mathbf{W^{hy}}} = \frac{\partial \mathbf{J_3}}{\partial \mathbf{p_3}} \frac{\partial \mathbf{p_3}}{\partial \mathbf{y_3}} \frac{\partial \mathbf{y_3}}{\partial \mathbf{W^{hy}}}$$



$$\begin{aligned}\frac{\partial \mathbf{J_3}}{\partial \mathbf{W^{hh}}} &= \frac{\partial \mathbf{J_3}}{\partial \mathbf{p_3}} \frac{\partial \mathbf{p_3}}{\partial \mathbf{y_3}} \frac{\partial \mathbf{y_3}}{\partial \mathbf{h_3}} \frac{\partial \mathbf{h_3}}{\partial \mathbf{z_3}} \frac{\partial \mathbf{z_3}}{\partial \mathbf{W^{hh}}} \\ &+ \frac{\partial \mathbf{J_3}}{\partial \mathbf{p_3}} \frac{\partial \mathbf{p_3}}{\partial \mathbf{y_3}} \frac{\partial \mathbf{y_3}}{\partial \mathbf{h_3}} \frac{\partial \mathbf{h_3}}{\partial \mathbf{z_3}} \frac{\partial \mathbf{z_3}}{\partial \mathbf{h_2}} \frac{\partial \mathbf{h_2}}{\partial \mathbf{z_2}} \frac{\partial \mathbf{z_2}}{\partial \mathbf{W^{hh}}} \\ &+ \frac{\partial \mathbf{J_3}}{\partial \mathbf{p_3}} \frac{\partial \mathbf{p_3}}{\partial \mathbf{y_3}} \frac{\partial \mathbf{y_3}}{\partial \mathbf{h_3}} \frac{\partial \mathbf{h_3}}{\partial \mathbf{z_3}} \frac{\partial \mathbf{z_3}}{\partial \mathbf{h_2}} \frac{\partial \mathbf{h_2}}{\partial \mathbf{z_2}} \frac{\partial \mathbf{z_2}}{\partial \mathbf{h_1}} \frac{\partial \mathbf{h_1}}{\partial \mathbf{z_1}} \frac{\partial \mathbf{z_1}}{\partial \mathbf{W^{hh}}}\end{aligned}$$

$$\begin{aligned}\frac{\partial \mathbf{J_2}}{\partial \mathbf{W^{hh}}} &= \frac{\partial \mathbf{J_2}}{\partial \mathbf{p_2}} \frac{\partial \mathbf{p_2}}{\partial \mathbf{y_2}} \frac{\partial \mathbf{y_2}}{\partial \mathbf{h_2}} \frac{\partial \mathbf{h_2}}{\partial \mathbf{z_2}} \frac{\partial \mathbf{z_2}}{\partial \mathbf{W^{hh}}} \\ &+ \frac{\partial \mathbf{J_2}}{\partial \mathbf{p_2}} \frac{\partial \mathbf{p_2}}{\partial \mathbf{y_2}} \frac{\partial \mathbf{y_2}}{\partial \mathbf{h_2}} \frac{\partial \mathbf{h_2}}{\partial \mathbf{z_2}} \frac{\partial \mathbf{z_2}}{\partial \mathbf{h_1}} \frac{\partial \mathbf{h_1}}{\partial \mathbf{z_1}} \frac{\partial \mathbf{z_1}}{\partial \mathbf{W^{hh}}}\end{aligned}$$

$$\frac{\partial \mathbf{J_1}}{\partial \mathbf{W^{hh}}} = \frac{\partial \mathbf{J_1}}{\partial \mathbf{p_1}} \frac{\partial \mathbf{p_1}}{\partial \mathbf{y_1}} \frac{\partial \mathbf{y_1}}{\partial \mathbf{h_1}} \frac{\partial \mathbf{h_1}}{\partial \mathbf{z_1}} \frac{\partial \mathbf{z_1}}{\partial \mathbf{W^{hh}}}$$

$$\begin{aligned}\frac{\partial \mathbf{J_3}}{\partial \mathbf{W^{xh}}} &= \frac{\partial \mathbf{J_3}}{\partial \mathbf{p_3}} \frac{\partial \mathbf{p_3}}{\partial \mathbf{y_3}} \frac{\partial \mathbf{y_3}}{\partial \mathbf{h_3}} \frac{\partial \mathbf{h_3}}{\partial \mathbf{z_3}} \frac{\partial \mathbf{z_3}}{\partial \mathbf{W^{xh}}} \\ &+ \frac{\partial \mathbf{J_3}}{\partial \mathbf{p_3}} \frac{\partial \mathbf{p_3}}{\partial \mathbf{y_3}} \frac{\partial \mathbf{y_3}}{\partial \mathbf{h_3}} \frac{\partial \mathbf{h_3}}{\partial \mathbf{z_3}} \frac{\partial \mathbf{z_3}}{\partial \mathbf{h_2}} \frac{\partial \mathbf{h_2}}{\partial \mathbf{z_2}} \frac{\partial \mathbf{z_2}}{\partial \mathbf{W^{xh}}} \\ &+ \frac{\partial \mathbf{J_3}}{\partial \mathbf{p_3}} \frac{\partial \mathbf{p_3}}{\partial \mathbf{y_3}} \frac{\partial \mathbf{y_3}}{\partial \mathbf{h_3}} \frac{\partial \mathbf{h_3}}{\partial \mathbf{z_3}} \frac{\partial \mathbf{z_3}}{\partial \mathbf{h_2}} \frac{\partial \mathbf{h_2}}{\partial \mathbf{z_2}} \frac{\partial \mathbf{z_2}}{\partial \mathbf{h_1}} \frac{\partial \mathbf{h_1}}{\partial \mathbf{z_1}} \frac{\partial \mathbf{z_1}}{\partial \mathbf{W^{xh}}}\end{aligned}$$

$$\begin{aligned}\frac{\partial \mathbf{J_2}}{\partial \mathbf{W^{xh}}} &= \frac{\partial \mathbf{J_2}}{\partial \mathbf{p_2}} \frac{\partial \mathbf{p_2}}{\partial \mathbf{y_2}} \frac{\partial \mathbf{y_2}}{\partial \mathbf{h_2}} \frac{\partial \mathbf{h_2}}{\partial \mathbf{z_2}} \frac{\partial \mathbf{z_2}}{\partial \mathbf{W^{xh}}} \\ &+ \frac{\partial \mathbf{J_2}}{\partial \mathbf{p_2}} \frac{\partial \mathbf{p_2}}{\partial \mathbf{y_2}} \frac{\partial \mathbf{y_2}}{\partial \mathbf{h_2}} \frac{\partial \mathbf{h_2}}{\partial \mathbf{z_2}} \frac{\partial \mathbf{z_2}}{\partial \mathbf{h_1}} \frac{\partial \mathbf{h_1}}{\partial \mathbf{z_1}} \frac{\partial \mathbf{z_1}}{\partial \mathbf{W^{xh}}}\end{aligned}$$

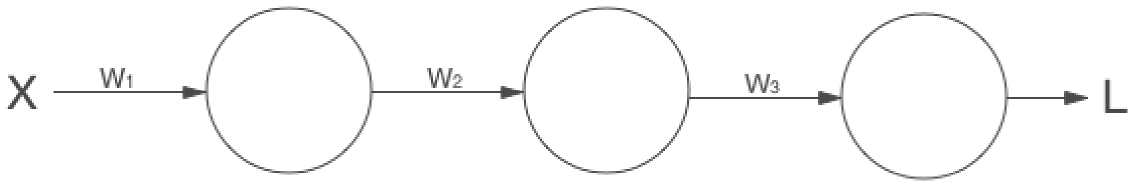
$$\frac{\partial \mathbf{J_1}}{\partial \mathbf{W^{xh}}} = \frac{\partial \mathbf{J_1}}{\partial \mathbf{p_1}} \frac{\partial \mathbf{p_1}}{\partial \mathbf{y_1}} \frac{\partial \mathbf{y_1}}{\partial \mathbf{h_1}} \frac{\partial \mathbf{h_1}}{\partial \mathbf{z_1}} \frac{\partial \mathbf{z_1}}{\partial \mathbf{W^{xh}}}$$

Steps of vanilla recurrent neural network training

1. For each training sequence X = (x1, x2,…, xk) where x1 and xk are the sequences' symbol for the start and end, respectively. Do the second step to the last step.
2. Set the value of the latent neuron activators to 0.5 and the value of the latent state of the previous step for the input of the first time step (x1) to zero (h0 = 0)
3. Follow step 4 to the last step to reach the end of the sequence
4. Apply the vector xi to the network and present the vector xi + 1 as the target response to the output units.
5. Calculate the activation value of latent and output neurons
6. Calculate the cost function in terms of network output and actual output.
7. Calculate the BPT
8. Update network weights
9. Check the stop condition (reaching the desired error threshold or ending Epochs)

(Werbos, 1990, Pascanu et al., 2013)

The problem of gradient vanishing and gradient explosion



$$\frac{\partial Loss}{\partial W_3} = \frac{\partial Loss}{\partial f(z_3)} \cdot \frac{\partial f(z_3)}{\partial W_3} = \frac{\partial Loss}{\partial f(z_3)} \cdot f'(z_3) \cdot W_3$$

$$\frac{\partial Loss}{\partial W_1} = \frac{\partial Loss}{\partial f(z_3)} \cdot \frac{\partial f(z_3)}{\partial f(z_2)} \cdot \frac{\partial f(z_2)}{\partial f(z1)} \cdot \frac{\partial f(z_1)}{\partial W_1}$$

$$= \frac{\partial Loss}{\partial f(z_3)} \cdot f'(z_3) \cdot W_3 \cdot f'(z_2) \cdot W_2 \cdot f'(z_1) \cdot W_1$$

If we take the weight between zero and one, then consecutive multiplying will make it insignificant, and this will reduce the effects of the primary layers of the network and make network training difficult. The same can be said from the opposite point of view; if we chose weights with values greater than one, consecutive multiplying would greatly increase the cost function.

What are solutions???

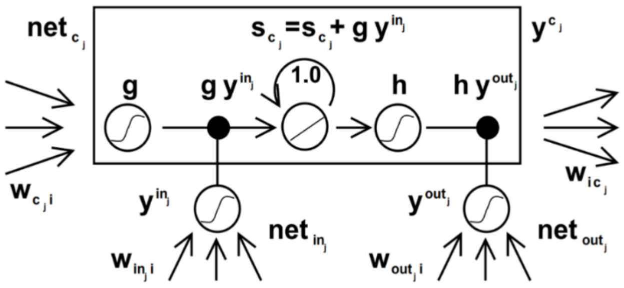
Gradian Vanishing

- Using RELU activator
- Using metaheuristic methods
- LSTMs

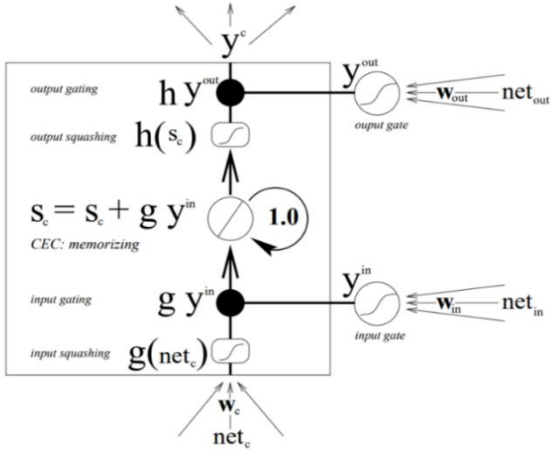
Gradian Explosion

- Using RELU activator
- Cutting the gradian
- Liming the amount of sequences
- LSTMs

Long Short Term Memory (LSTM)

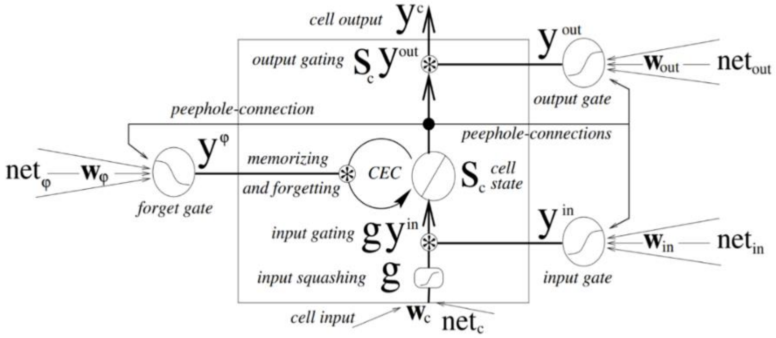


The first generation of LSTM networks (Hochreiter and Schmidhuber, 1997)



The second generation of LSTM networks (Gers et al., 1999)

- First Generation LSTM Networks (1995-1997): These networks had two input and output gates, as well as a recurrent edge, activation function, and an identity function.
- Second Generation LSTM Introduce forget gate (1999)
- Third Generation: Introducing Peephole Connection (2000)
- It has been going on since 2000 ...



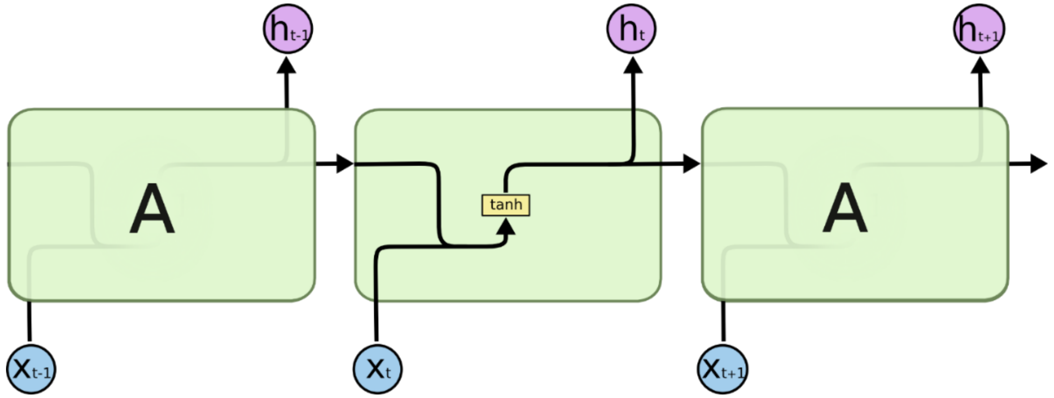
Third Generation LSTM Networks (Gers and Schmidhuber, 2000)

Comparison of LSTM network performance with vanilla recurrent networks

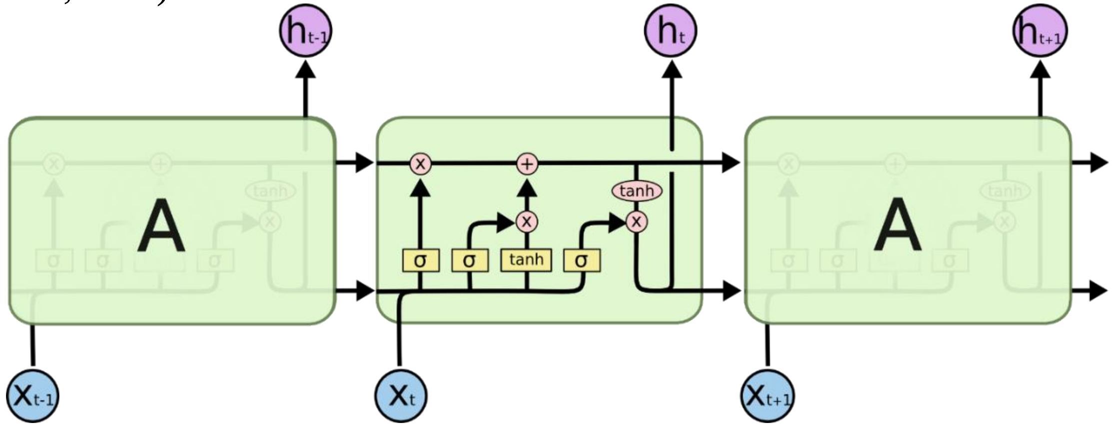
Method	Delay p	Learning rate	# weights	% Successful trials	Success after
RTRL	4	1.0	36	78	1,043,000
RTRL	4	4.0	36	56	892,000
RTRL	4	10.0	36	22	254,000
RTRL	10	1.0-10.0	144	0	> 5,000,000
RTRL	100	1.0-10.0	10404	0	> 5,000,000
BPTT	100	1.0-10.0	10404	0	> 5,000,000
CH	100	1.0	10506	33	32,400
LSTM	100	1.0	10504	100	5,040

(Hochreiter and Schmidhuber, 1997)

LSTM Networks’ Mechanism



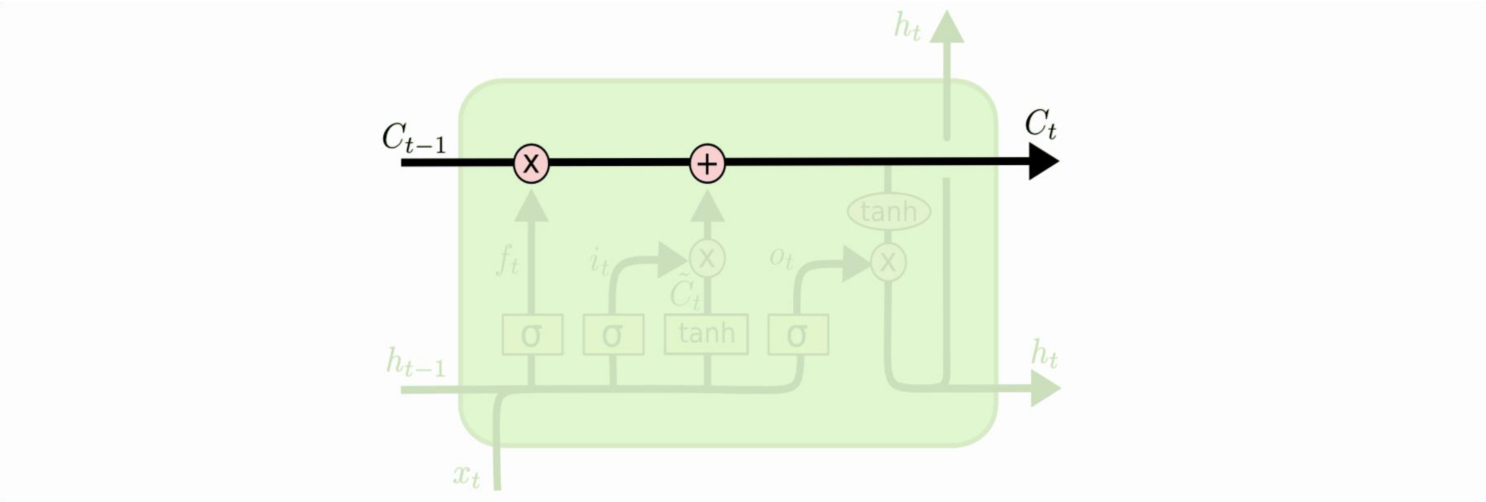
Repetitive modules in vanilla recursive neural networks have only one layer (colah, 2015).



“The repeating module in an LSTM contains four interacting layers” (colah, 2015)

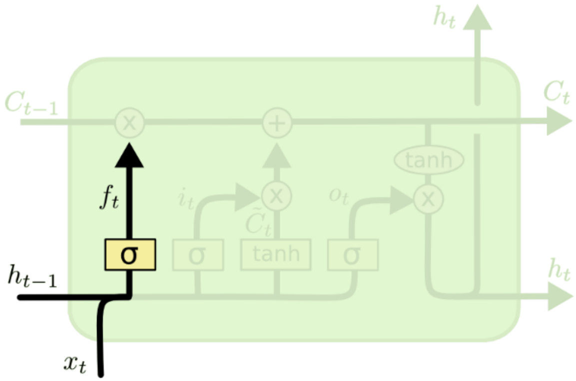
LSTM Networks’ Mechanism

The main element of LSTMs is the state cell, which is actually a horizontal line at the top of the shape. The state cell can be thought of as a conveyor belt that moves from the beginning to the end of a sequence or chain with partial linear interactions (its structure is very simple and little change occurs in it).



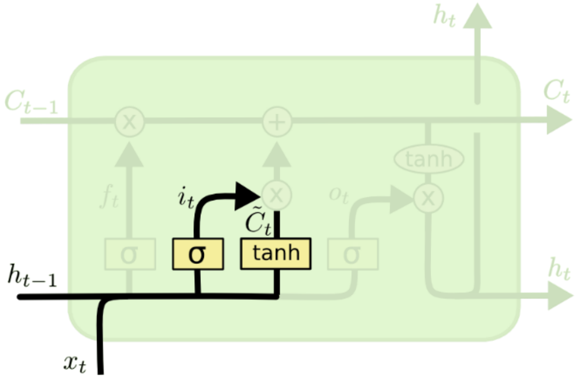
LSTM Networks' Mechanism

The first step in LSTM is deciding on the information we want to clear from the state cell. This decision is made by a sigmoid layer called the forget gate.



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \ + \ b_f\right)$$

The next step is to decide what new information we want to store in the state cell.

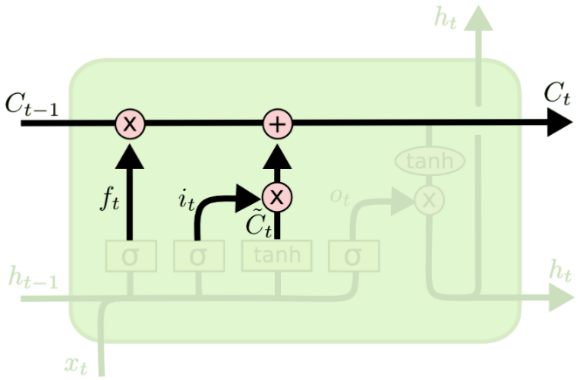


$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \ + \ b_C)$$

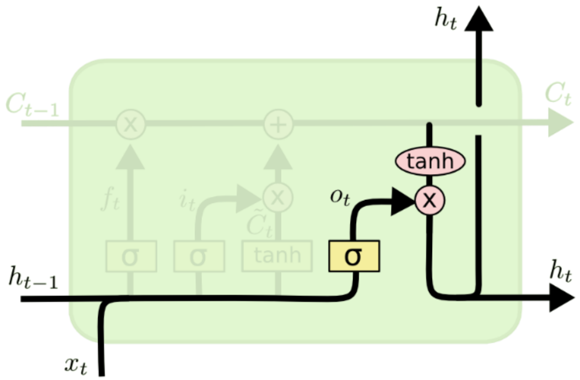
LSTM Networks' Mechanism

Now it's time to update the old state cell, Ct-1, to the new state cell, Ct.



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

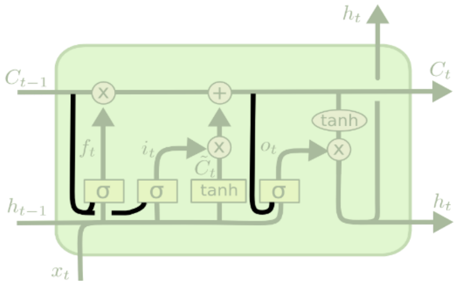
Finally, we have to decide what information we are going to output. This output will be in respect to the state cell's value but will pass through a certain filter.



$$o_t = \sigma\left(W_o \left[h_{t-1}, x_t\right] + b_o\right)$$

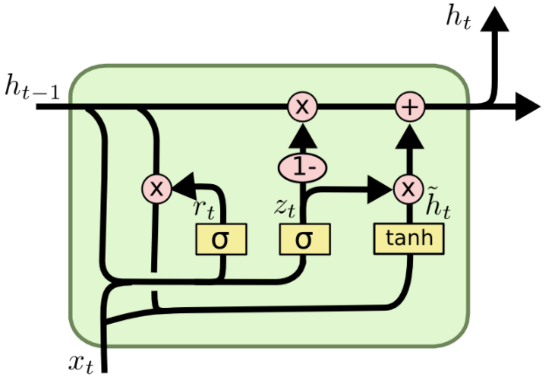
$$h_t = o_t * \tanh\left(C_t\right)$$

Other LSTMs

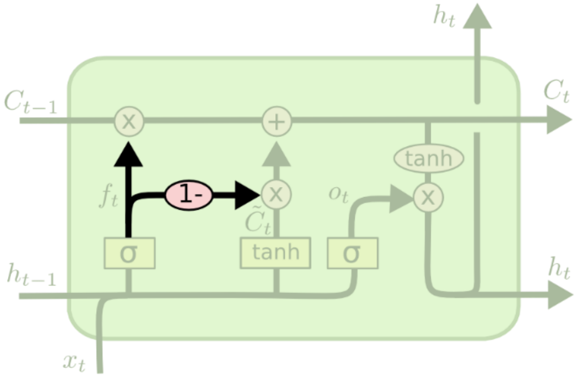


$$f_t = \sigma\left(W_f \cdot [\boldsymbol{C_{t-1}}, h_{t-1}, x_t] \;+\; b_f\right)$$
$$i_t = \sigma\left(W_i \cdot [\boldsymbol{C_{t-1}}, h_{t-1}, x_t] \;+\; b_i\right)$$
$$o_t = \sigma\left(W_o \cdot [\boldsymbol{C_t}, h_{t-1}, x_t] \;+\; b_o\right)$$

Peephole



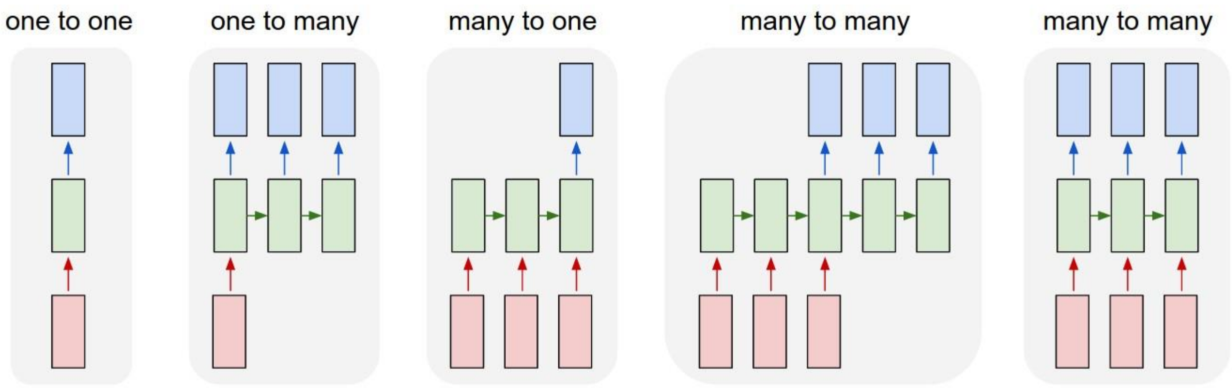
$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$
$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$
$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

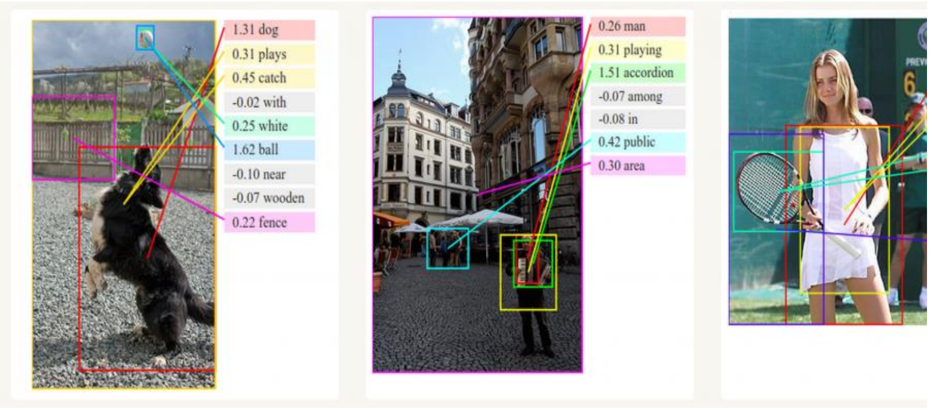


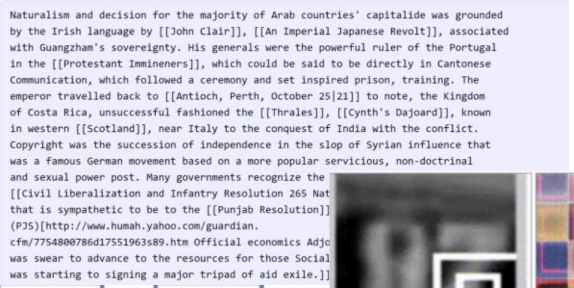
$$C_t = f_t * C_{t-1} + \boldsymbol{(1 - f_t)} * \tilde{C}_t$$

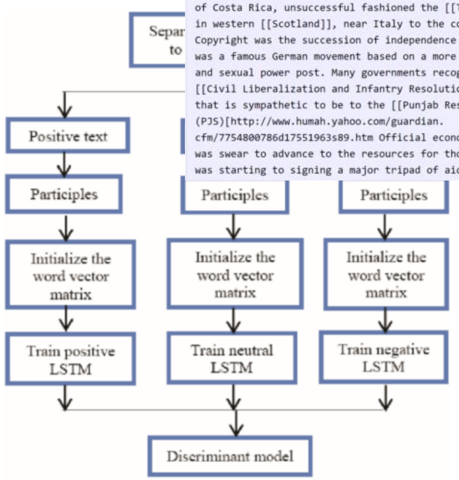
GRU-(Cho et al., 2014)

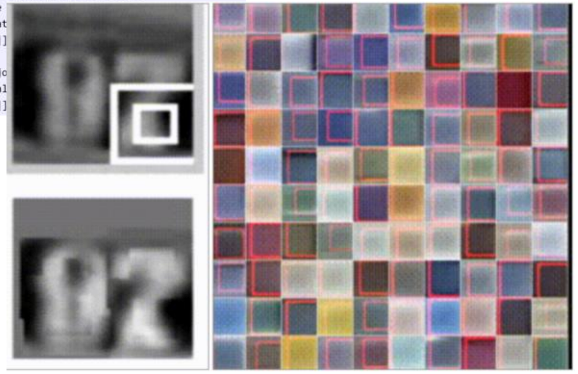
Architectures and Applications of Standard Recurrent Neural Network and LSTM






Naturalism and decision for the majority of Arab countries' capitalide was grounded by the Irish language by [[John Clair]], [[An Imperial Japanese Revolt]], associated with Guangzham's sovereignty. His generals were the powerful ruler of the Portugal in the [[Protestant Immineners]], which could be said to be directly in Cantonese Communication, which followed a ceremony and set inspired prison, training. The emperor travelled back to [[Antioch, Perth, October 25|21]] to note, the Kingdom of Costa Rica, unsuccessful fashioned the [[Thrales]], [[Cynth's Dajoard]], known in western [[Scotland]], near Italy to the conquest of India with the conflict. Copyright was the succession of independence in the slop of Syrian influence that was a famous German movement based on a more popular servicious, non-doctrinal and sexual power post. Many governments recognize the
[[Civil Liberalization and Infantry Resolution 265 Nat
that is sympathetic to be to the [[Punjab Resolution]]
(PJS)[http://www.humah.yahoo.com/guardian.
cfm/7754800786d17551963s89.htm Official economics Adjo
was swear to advance to the resources for those Social
was starting to signing a major tripad of aid exile.]]




Left: RNN learns to read house numbers. Right: RNN learns to paint house numbers.

- Sequence generation like text or music
- Predict the trend of stock indices
- image captioning
- Translation machines
- Photo and video classification
- Convert speech to text
- Labeling and classification
- Genetic sequencing

(karpathy,2015.)

Some Points About LSTM

- Compared to standard recurrent neural networks, these networks were highly capable of handling long sequences.
- One of the drawbacks of LSTM networks is the need for a lot of data and the high amount of computation that makes the learning process difficult. To solve these problems, scientists have made improvements to LSTM networks over time;
- There is no definite indication of the modified LSTM networks' superiority over the classical LSTM described above. Although LSTM networks have a slower learning process than modified LSTM networks, this is a drawback. The predictive power of LSTM networks will be higher if they have the right amount of data and provide the right time to the network.
- It is the most widely used network in people's daily lives. Virtual keyboards on smartphones that predict words and sentences for us, voice assistants Alexa, Cortana, Bixby, Siri, etc., classification of movies and content on social networks, etc., all of which are powered by recurrent neural networks, especially LSTMs.

More Intuition

1. Andy. “Recurrent Neural Networks and LSTM Tutorial in Python and TensorFlow.” Adventures in Machine Learning, 3 Apr. 2018, adventuresinmachinelearning.com/recurrent-neural-networks-lstm-tutorial-tensorflow/.
2. CHO, K., VAN MERRIËNBOER, B., GULCEHRE, C., BAHDANAU, D., BOUGARES, F., SCHWENK, H. & BENGIO, Y. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.
3. GERS, F. A. & SCHMIDHUBER ,J. Recurrent nets that time and count. Neural Networks, 2000. IJCNN 2000, Proceedings of the IEEE-INNS-ENNS International Joint Conference on, 2000. IEEE, 189-194.
4. GERS, F. A., SCHMIDHUBER, J. & CUMMINS, F. 1999. Learning to forget: Continual prediction with LSTM.
5. GRAVES, A. 2013. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*.
6. GREFF, K., SRIVASTAVA, R. K., KOUTNÍK, J., STEUNEBRINK, B. R. & SCHMIDHUBER, J. 2017. LSTM: A search space odyssey. *IEEE transactions on neural networks and learning systems,* 28**,** 2222-2232.
7. HOCHREITER, S. & SCHMIDHUBER, J. 1997. Long short-term memory. *Neural computation,* 9, 1735-1780.
8. LECUN, Y., BENGIO, Y. & HINTON, G. 2015. Deep learning. nature, 521, 436.
9. LI, D. & QIAN, J. Text sentiment analysis based on long short-term memory. Computer Communication and the Internet (ICCCI), 2016 IEEE International Conference on, 2016. IEEE, 471-475.
10. LIPTON, Z. C., BERKOWITZ, J. & ELKAN, C. 2015. A critical review of recurrent neural networks for sequence learning. arXiv preprint arXiv:1506.00019.
11. NIELSEN, M. A. 2015. Neural networks and deep learning, Determination Press.
12. OLAH, C. 2015. Understanding lstm networks, 2015. URL https://colah. github. io/posts/2015-08-Understanding-LSTMs.
13. PASCANU, R., MIKOLOV, T. & BENGIO, Y. On the difficulty of training recurrent neural networks. International Conference on Machine Learning, 2013. 1310-1318.
14. ROJAS, R. 1996. The backpropagation algorithm. Neural networks. Springer.
15. SEPTIANDRI, A. A. 2017. Predicting the Gender of Indonesian Names. arXiv preprint arXiv:1707.07129.
16. WERBOS, P. J. 1990. Backpropagation through time: what it does and how to do it. Proceedings of the IEEE, 78, 1550-1560.
17. YUAN, H., VAN DER WIELE, C. & KHORRAM, S. 2009. An Automated Artificial Neural Network System for Land Use/Land Cover Classification from Landsat TM Imagery. Remote Sensing, 1, 243.

Happy :)