Recurrent Neuronal Nets and Applications

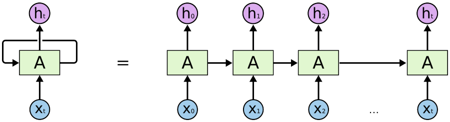
Li CHENG

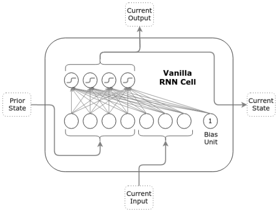
Bioinformatics Institute, Singapore

Disclaimer: Many of the contents are from the Internet.

RNNs

Recurrent Neuronal Nets or RNNs



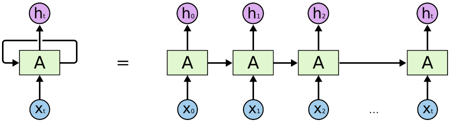


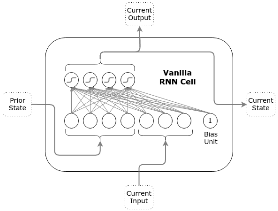
Work principle

$$h_t = \phi\left(Wx_t + Uh_{t-1} + b\right)$$

RNNs

Recurrent Neuronal Nets or RNNs





Work principle

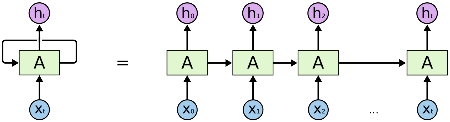
$$h_t = \phi\left(Wx_t + Uh_{t-1} + b\right)$$

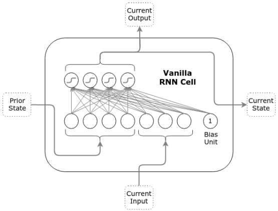
How to train

Back-propagation Through Time

RNNs

Recurrent Neuronal Nets or RNNs





Work principle

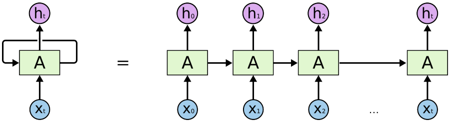
$$h_t = \phi\left(Wx_t + Uh_{t-1} + b\right)$$

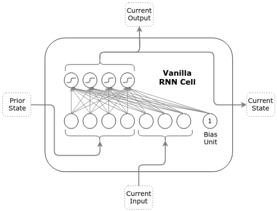
How to train Back-propagation Through Time

Main issue Vanishing (or Exploding) Gradients over long term

RNNs

Recurrent Neuronal Nets or RNNs





Work principle

$$h_t = \phi\left(Wx_t + Uh_{t-1} + b\right)$$

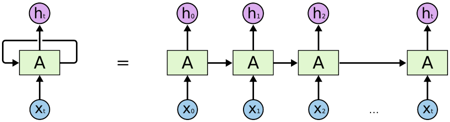
How to train Back-propagation Through Time

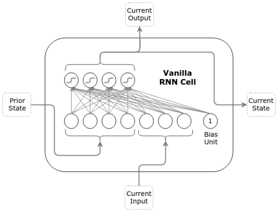
Main issue Vanishing (or Exploding) Gradients over long term

Main solution Long Short-Term Memory nets (LSTMs) and friends

RNNs

Recurrent Neuronal Nets or RNNs





Work principle

$$h_t = \phi\left(Wx_t + Uh_{t-1} + b\right)$$

How to train Back-propagation Through Time

Main issue Vanishing (or Exploding) Gradients over long term

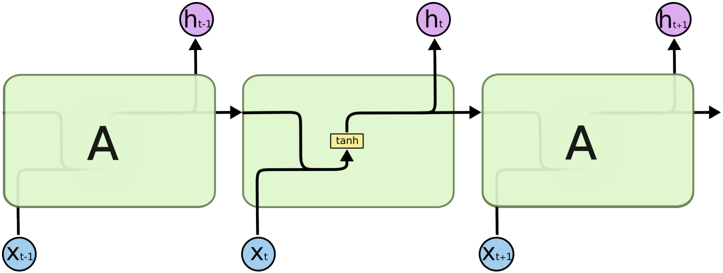
Main solution Long Short-Term Memory nets (LSTMs) and friends

Will focus on LSTMs in what follows

(NC97): S. Hochreiter and J. Schmidhuber, Long Short-Term Memory, Neural Computation, 1997.

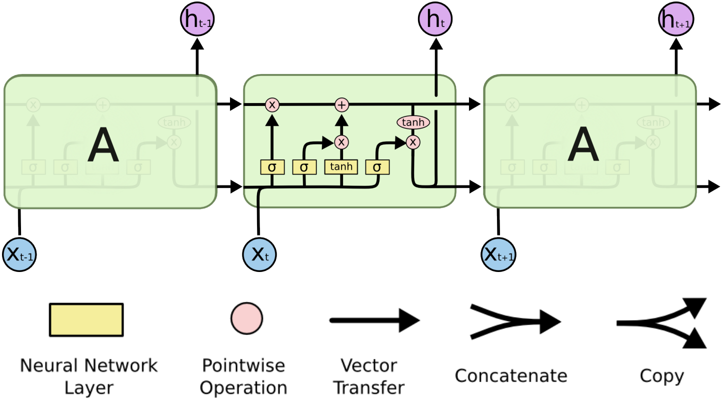
LSTMs

A standard RNN is



LSTMs

Meanwhile, a standard LSTM is

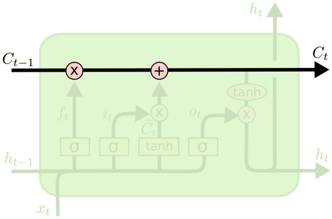


LSTMs

Let us look at each component of LSTM:

LSTMs

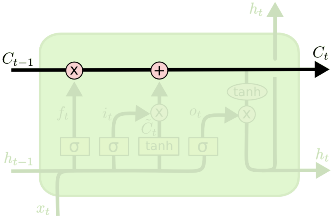
Let us look at each component of LSTM:



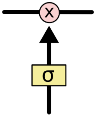
cell state

LSTMs

Let us look at each component of LSTM:



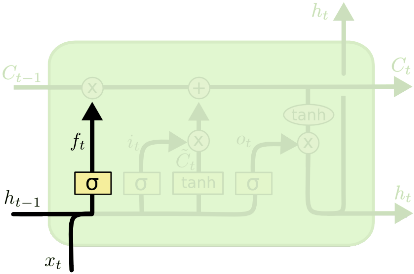
cell state



gate

LSTMs

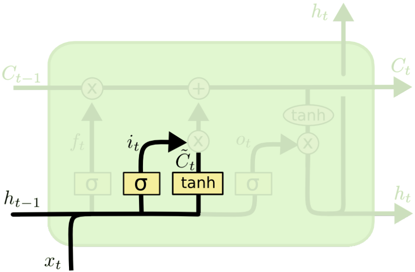
Forget gate:



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

LSTMs

Information gate:

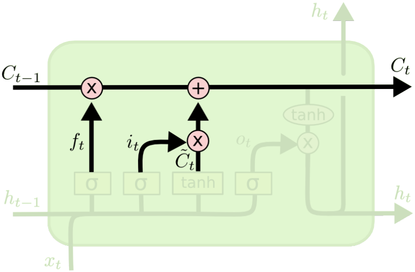


$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$

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

LSTMs

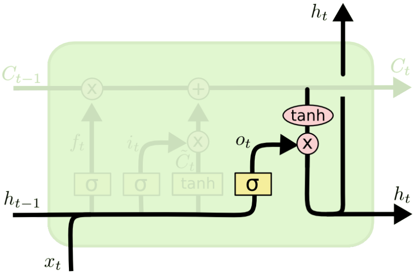
Update of the cell state:



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

LSTMs

Output:

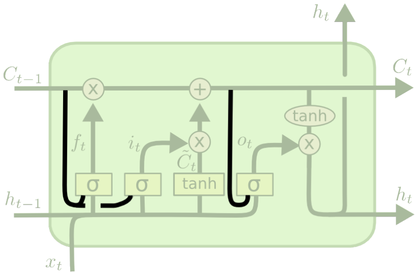


$$o_t = \sigma\left(W_o\ [h_{t-1}, x_t]\ +\ b_o\right)$$

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

Variants of LSTMs

Peephole connection:

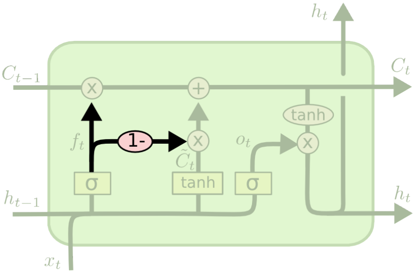


$$f_t = \sigma\left(W_f \cdot [\boldsymbol{C_{t-1}}, h_{t-1}, x_t] \; + \; b_f\right)$$
$$i_t = \sigma\left(W_i \cdot [\boldsymbol{C_{t-1}}, h_{t-1}, x_t] \; + \; b_i\right)$$
$$o_t = \sigma\left(W_o \cdot [\boldsymbol{C_t}, h_{t-1}, x_t] \; + \; b_o\right)$$

(IJCNN00): F. Gers and J. Schmidhuber, Recurrent Nets that Time and Count, IJCNN, 2000.

Variants of LSTMs

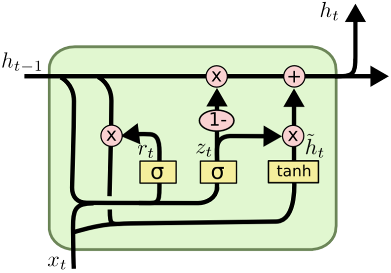
Couple forget and input gates:



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

Variants of LSTMs

Gated Recurrent Unit (GRU):



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

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

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

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

(EMNLP14): K. Cho et al., Learning Phrase Representations using RNN EncoderDecoder for Statistical Machine Translation, EMNLP, 2014.

Variants of LSTMs

Strongly-Typed LSTMs or GRUs: an attempt to ensure overall consistency among RNNs' complicated quasi-linear operations

Strongly-Typed Quasi-Linear Algebra

Quasi-linear algebra is linear algebra supplemented with nonlinear functions that act coordinatewise.

Definition 1. *Dot-products are denoted by $\langle \mathbf{w}, \mathbf{x} \rangle$ or $\mathbf{w}^\intercal \mathbf{x}$. A **type** $\mathcal{T} = \left(V, \langle \bullet, \bullet \rangle, \{\mathbf{t}_i\}_{i=1}^d\right)$ is a d-dimensional vector space equipped with an inner product and an orthogonal basis such that $\langle \mathbf{t}_i, \mathbf{t}_j \rangle = \mathbb{1}_{[i=j]}$.*

From a type perspective, apply an SVD to $\mathbf{V} = \mathbf{P}\mathbf{D}\mathbf{Q}^\intercal$ and observe that $\mathbf{V}^2 = \mathbf{P}\mathbf{D}\mathbf{Q}^\intercal\mathbf{P}\mathbf{D}\mathbf{Q}^\intercal$. Each multiplication by $\mathbf{P}$ or $\mathbf{Q}^\intercal$ transforms the input to a new type, obtaining

$$\boxed{\underbrace{\mathcal{T}_\mathbf{h} \xrightarrow{\mathbf{D}\mathbf{Q}^\intercal} \mathcal{T}_{\text{lat}_1} \xrightarrow{\mathbf{P}} \mathcal{T}_{\text{lat}_2}}_{\mathbf{V}} \underbrace{\xrightarrow{\mathbf{D}\mathbf{Q}^\intercal} \mathcal{T}_{\text{lat}_3} \xrightarrow{\mathbf{P}} \mathcal{T}_{\text{lat}_4}}_{\mathbf{V}}}.$$

Thus $\mathbf{V}$ sends $\mathbf{z} \mapsto \mathcal{T}_{\text{lat}_2}$ whereas $\mathbf{V}^2$ sends $\mathbf{z} \mapsto \mathcal{T}_{\text{lat}_4}$. Adding terms involving $\mathbf{V}$ and $\mathbf{V}^2$, as in Eq. (2), entails adding vectors expressed in different orthogonal bases – which is analogous to adding joules to volts. The same problem applies to LSTMs and GRUs.

(ICML16): D. Balduzzi, M. Ghifary, Strongly-typed recurrent neural networks, ICML, 2016.

Variants of LSTMs

Also many many other variants, such as recurrent residual/highway nets, grid LSTMs...

Variants of LSTMs

Also many many other variants, such as recurrent residual/highway nets, grid LSTMs...

Then

which variant is better than others? Arxiv15, ICML15

Variants of LSTMs

Also many many other variants, such as recurrent residual/highway nets, grid LSTMs...

Then

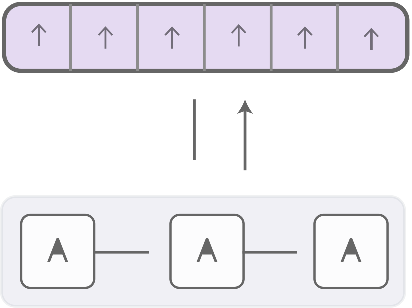
which variant is better than others? Arxiv15, ICML15

to visualize and understand RNNs ICLR16

(Arxiv15): K. Greff, R Srivastava, J Koutnk, B. Steunebrink, J Schmidhuber, LSTM: A Search Space Odyssey, Arxiv, 2015.
(ICML15): Rafal Jozefowicz, Wojciech Zaremba and Ilya Sutskever, An Empirical Exploration of Recurrent Network Architectures, ICML, 2015.
(ICLR16): Andrej Karpathy, Justin Johnson, Li Fei-Fei, Visualizing and Understanding Recurrent Networkss, ICLR, 2016.

Extensions of LSTMs

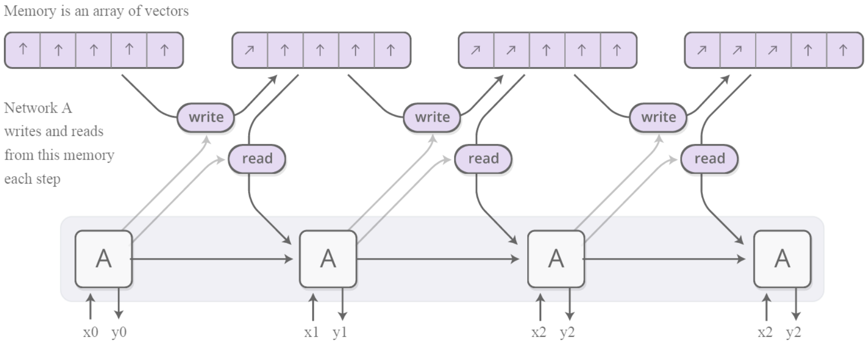
Example 1: Neural Turing Machines:



(Arxiv14): K. Cho et al., Neural Turing machine, Arxiv, 2014.

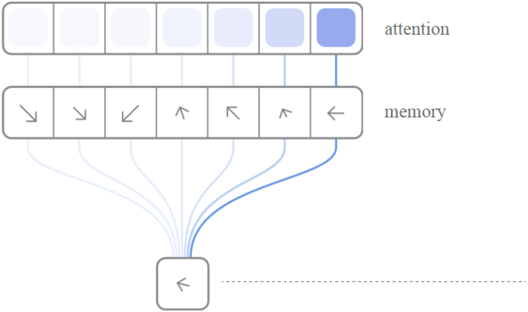
Extensions of LSTMs: Neural Turing Machines

Access to external memory to read and write



Extensions of LSTMs: Neural Turing Machines

Read by attention distribution



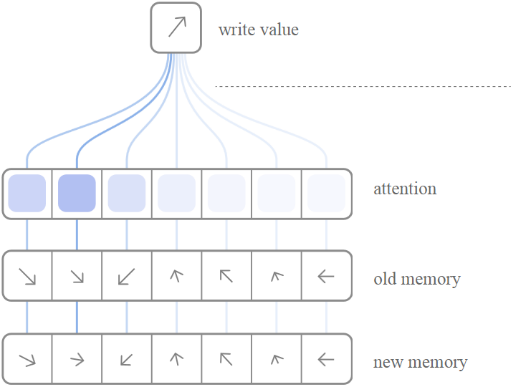
The RNN gives an attention distribution which describe how we spread out the amount we care about different memory positions

The read result is a weighted sum.

$$r \leftarrow \sum_i a_i M_i$$

Extensions of LSTMs: Neural Turing Machines

Write by another attention distribution



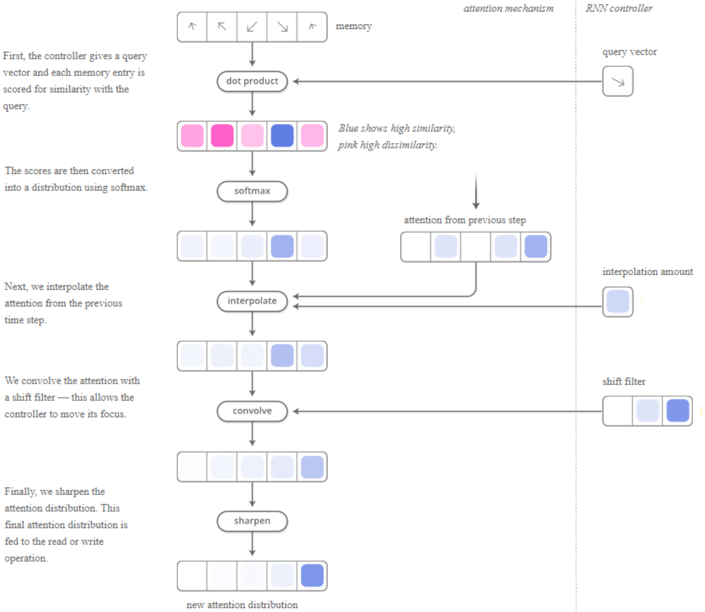
Instead of writing to one location, we write everywhere, just do different extents.

The RNN gives an attention distribution, describing how much we should change each memory position towards the write value.

$$M_i \leftarrow a_i w + (1 - a_i) M_i$$

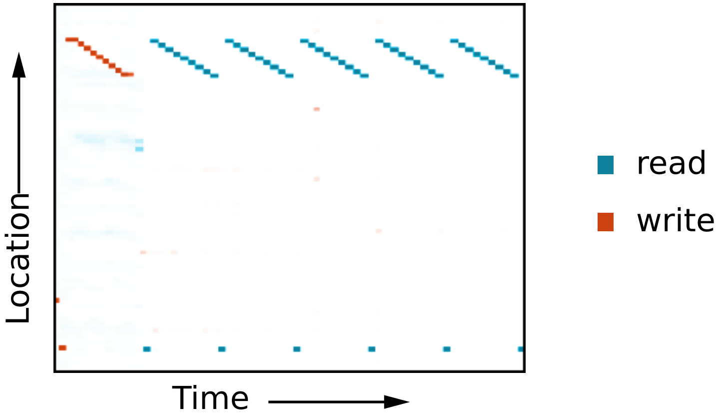
Extensions of LSTMs: Neural Turing Machines

Attention mechanism: content-based & location-based



Extensions of LSTMs: Neural Turing Machines

What it can do: e.g. a copy task



Extensions/friends of Neural Turing Machines (NTMs)

Neural GPU Can add and multiply numbers. Arxiv15a

Reinforcement Learning NTM RL instead. Arxiv15b

Neural Random Access Machines Use pointer. Arxiv15c

Using stacks or queues NIPA15a & NIPA15b

Memory networks similar idea. Arxiv14 & Arxiv15d

(Arxiv15a): L. Kaiser, I. Sutskever, Neural GPUs Learn Algorithms, Arxiv, 2015.
(Arxiv15b): W. Zaremba, I. Sutskever, Reinforcement Learning Neural Turing Machines, Arxiv, 2015.
(Arxiv15c): K. Kurach, M. Andrychowicz, I. Sutskever, Neural Random Access Machines, Arxiv, 2015.
(NIPA15a): E. Grefenstette, K.M. Hermann, M. Suleyman, P. Blunsom, Learning to Transduce with Unbounded Memory, NIPS, 2015.
(NIPA15b): A. Joulin, T. Mikolov, Inferring Algorithmic Patterns with Stack-Augmented Recurrent Nets, NIPS, 2015.
(Arxiv15): J. Weston, S. Chopra, A. Bordes, Memory Networks, Arxiv, 2014.
(Arxiv15d): A. Kumar, O. Irsoy, J. Su, J. Bradbury, R. English, B. Pierce, P. Ondruska, I. Gulrajani, R. Socher, Ask Me Anything: Dynamic Memory Networks for Natural Language Processing, Arxiv, 2015.

Extensions/friends of Neural Turing Machines (NTMs)

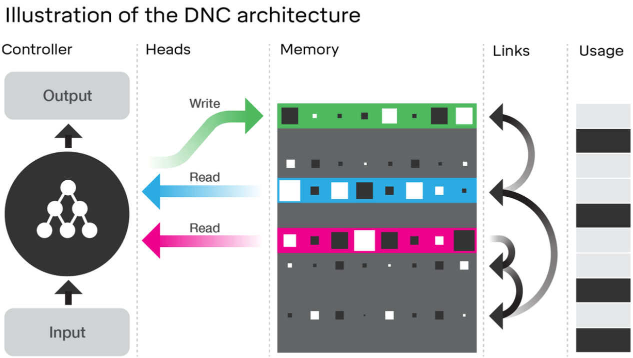
It has in fact been extensively studied long time back in the forms of associative memories etc.

(BookOxfordU93): Edited by M. H. Hassoun, Associative neural memories: theory and implementation, Oxford Univ. Press, 1993.
(ACCSS92): S. Das, C. Giles, and G. Z. Sun, Learning context-free grammars: capabilities and limitations of a recurrent neural networks with an external stack memory, ACCSS, 1992.
(ACCSS96): P. Grunwald, recurrent network that performs a context-sensitive prediction task, ACCSS, 1996.
(NIPS93): . C. Mozer and S. Das, A connectionist symbol manipulator that discovers the structure of context-free languages, NIPS, 1993.
(ConnSci99): P. Rodriguez, J. Wiles, and J. L. Elman, A recurrent eural network that learns to count, Connection science, 1999.

Extensions of Neural Turing Machines (NTMs)

One notable extension:

Differentiable neural computers Nature16

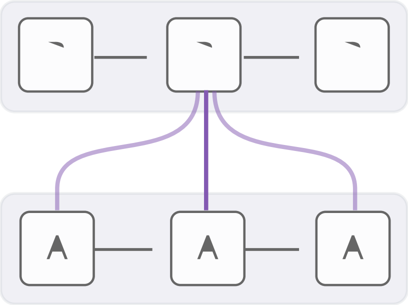


And many others ...

(nature16): Alex Graves, et al., Hybrid computing using a neural network with dynamic external memory, Nature, 2016.

Extensions of LSTMs

Example 2: Attentional interface (focus on a part of the input):

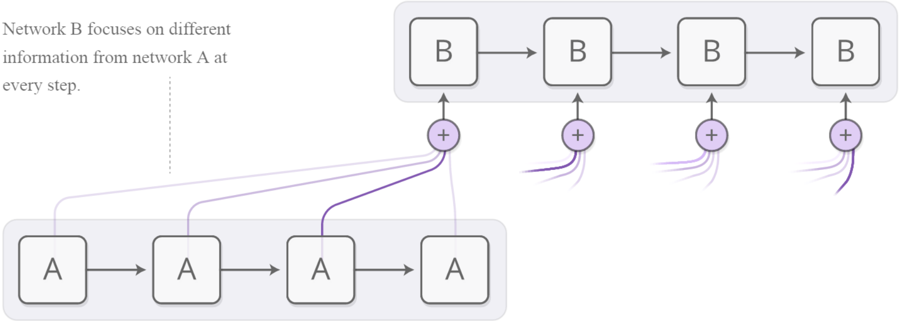


Many research efforts, e.g. Arxiv14

(Arxiv14): D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, Arxiv, 2014.

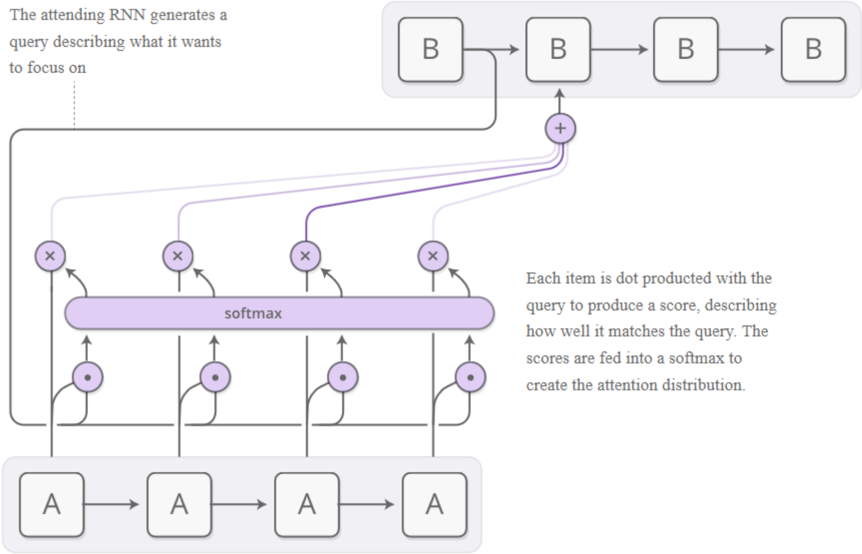
Attentional interface

Focus on a part of the input:



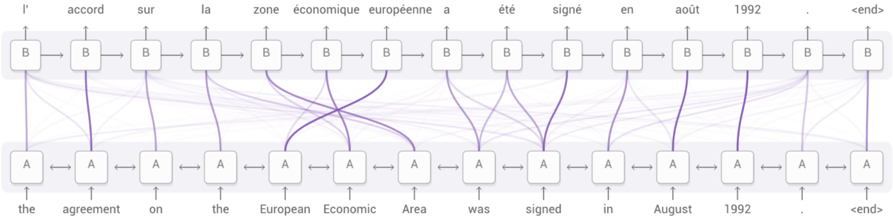
Attentional interface

Usually generated with content-based attention:



Attentional interface

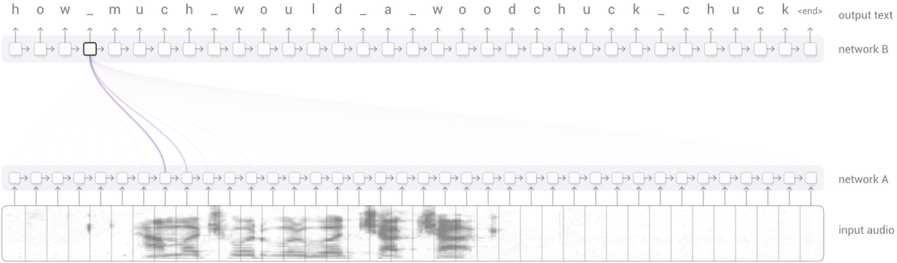
An exemplar usage in machine translation:



(Arxiv14): D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, Arxiv, 2014.

Attentional interface

Another example in voice recognition (Arxiv15):



(Arxiv15): W. Chan, N. Jaitly, Q.V. Le, O. Vinyals, Listen, Attend and Spell, Arxiv, 2015.

Attentional interface

Yet another example in image captioning (e.g. ICML15):

A woman is throwing a frisbee in a park.

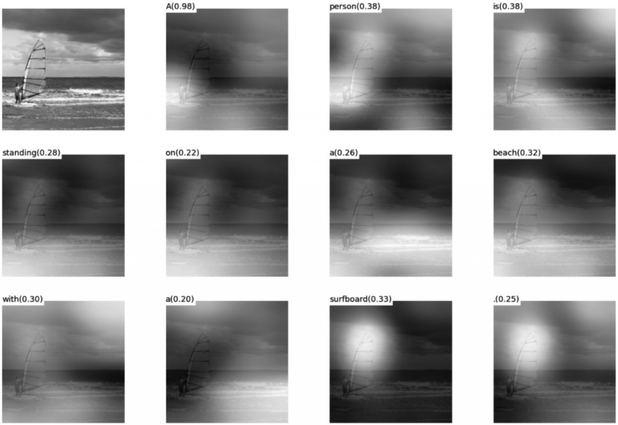
A dog is standing on a hardwood floor.

A stop sign is on a road with a mountain in the background.

(ICML15): K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R.S. Zemel, Y. Bengio, Show, attend and tell: Neural image caption generation with visual attention, ICML, 2015.

Attentional interface

Yet another example in image captioning (e.g. ICML15):



(b) A person is standing on a beach with a surfboard.

(ICML15): K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R.S. Zemel, Y. Bengio, Show, attend and tell: Neural image caption generation with visual attention, ICML, 2015.

Applications

Visual Attentions: Learns to draw numbers and things:

(NIPS14): Volodymyr Mnih, Nicolas Heess, Alex Graves, Koray Kavukcuoglu, Recurrent Models of Visual Attention, NIPS, 2014.

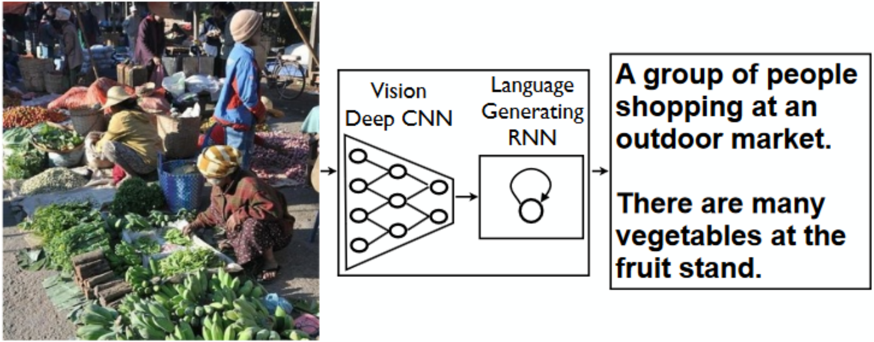
Applications

Visual Attentions: Learns to draw house numbers:

(ICML15): K Gregor, I Danihelka, A Graves, D Rezende, D Wierstra, DRAW: A Recurrent Neural Network For Image Generation, ICML, 2015.

Applications

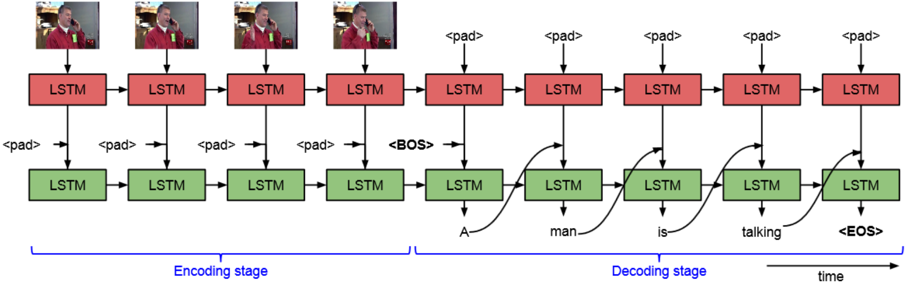
Image captioning, e.g.



(CVPR15): Oriol Vinyals, Alexander Toshev, Samy Bengio, Dumitru Erhan, Show and Tell: A Neural Image Caption Generator, CVPR, 2015.

Applications

Video captioning, e.g.



(ICML15): S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, K. Saenko, S2VT: Sequence to Sequence Video to Text, ICML, 2015.

Applications

visual question answering, e.g.

DAQUAR 1553
What is there in front of the sofa?
Ground truth: table
IMG+BOW: table (0.74)
2-VIS+BLSTM: table (0.88)
LSTM: chair (0.47)

COCOQA 5078
How many leftover donuts is the red bicycle holding?
Ground truth: three
IMG+BOW: two (0.51)
2-VIS+BLSTM: three (0.27)
BOW: one (0.29)

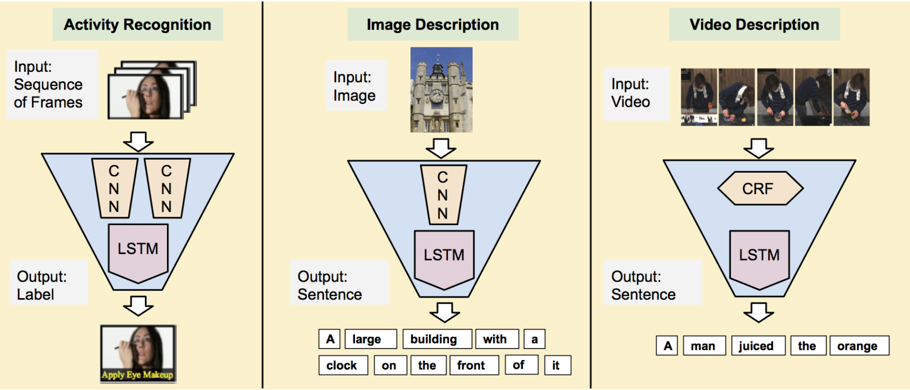
COCOQA 1238
What is the color of the tee-shirt?
Ground truth: blue
IMG+BOW: blue (0.31)
2-VIS+BLSTM: orange (0.43)
BOW: green (0.38)

COCOQA 26088
Where is the gray cat sitting?
Ground truth: window
IMG+BOW: window (0.78)
2-VIS+BLSTM: window (0.68)
BOW: suitcase (0.31)

(NIPS15): Mengye Ren, Ryan Kiros, Richard Zemel, Exploring Models and Data for Image Question Answering, NIPS, 2015.

Applications

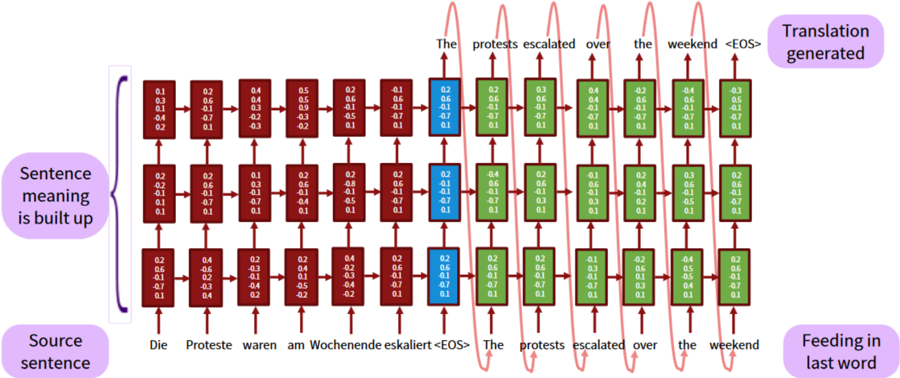
Action recognition, e.g.



(ICML15): J. Donahue, L Hendricks, M. Rohrbach, S. Venugopalan, S. Guadarrama, K. Saenko, T. Darrell, Long-term Recurrent Convolutional Networks for Visual Recognition and Description, CVPR, 2015.

Applications

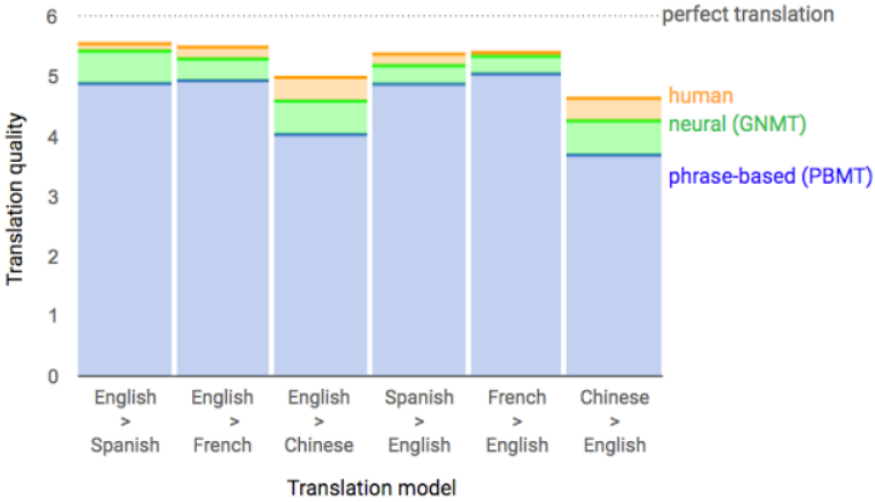
Neural Machine Translation:



(Arxiv16): Y Wu, et al., Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation, Arxiv, 2016.

Applications

Neural Machine Translation



(Arxiv16): Y Wu, et al., Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation, Arxiv, 2016.

Applications

and a lot more ...

Possible Activation Functions

The activation function f defines the neuron output. It could be a

- step function
- piecewise linear function
- tanh function
- logistic regression (sigmoid) function
- ...

Note: f is often bounded, non-constant, monotonically-increasing, and continuous.