Deconstructing Recurrent Neural Networks

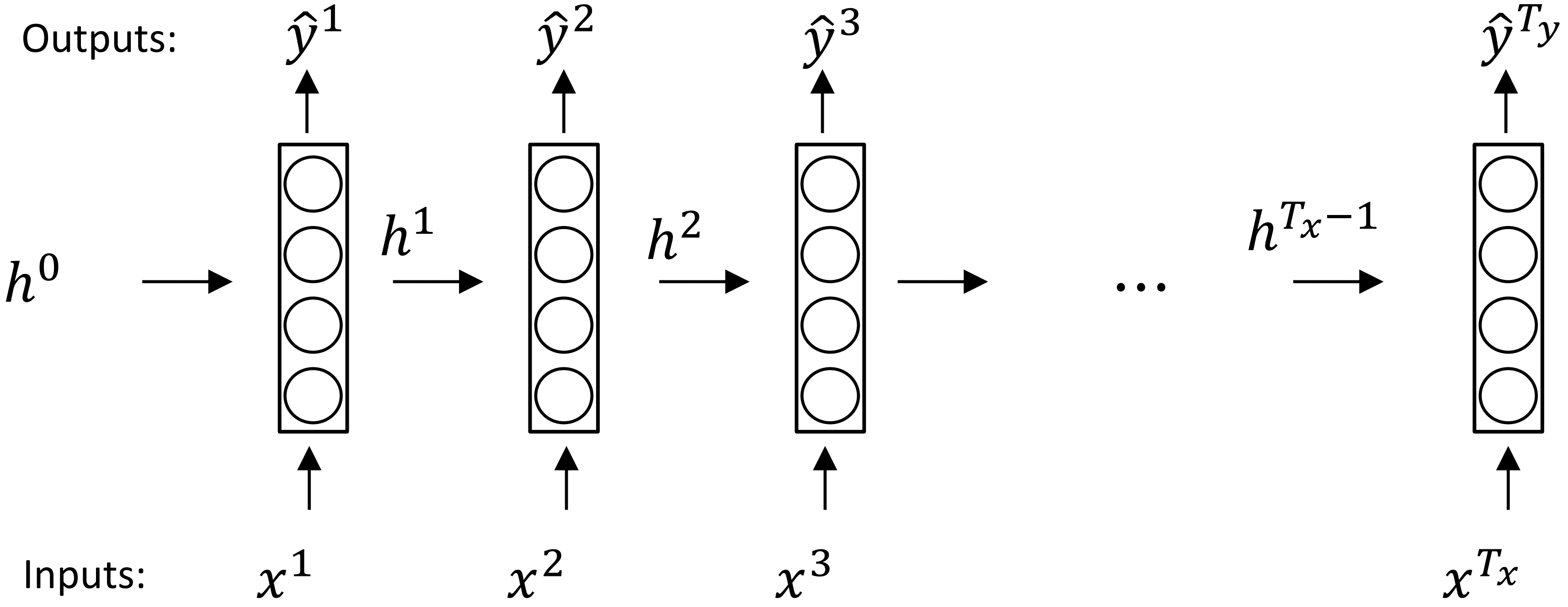
30 April, 2020

Alexey Zaytsev, head of Laboratory LARSS, PhD

Foundations of Data Science

Skoltech
Skolkovo Institute of Science and Technology

Forward propagation through Recurrent Neural Network

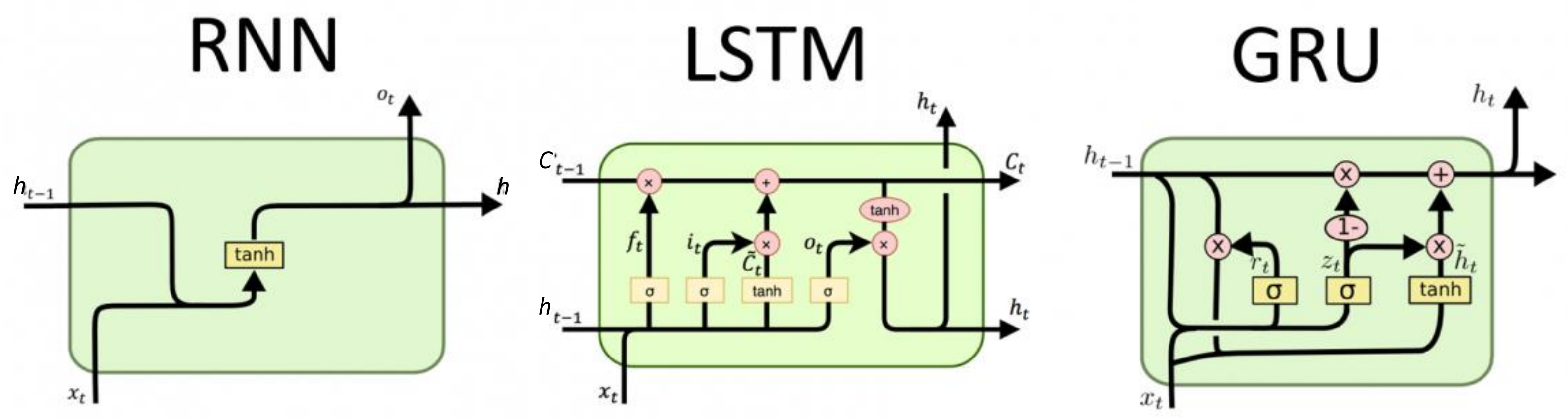


Input For NLP problems | It | is | truth | universally …

Adopted from Andrew Ng

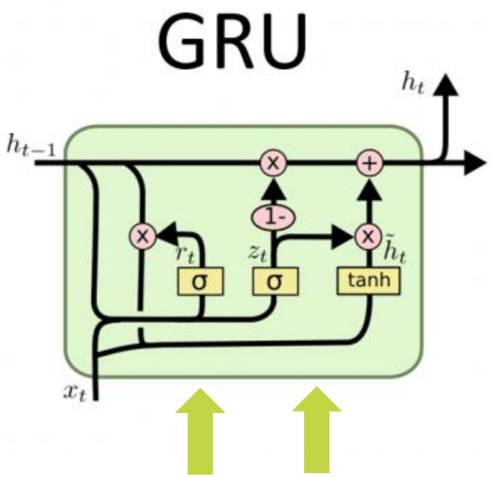
Good RNN units are LSTM and GRU. Vanilla RNN is not good.

- LSTM: Long Short Term Memory
- GRU: Gated Recurrent Unit



GRU – Gated Recurrent Unit

- Update gate – what to pay attention to
- Reset gate – what to forget

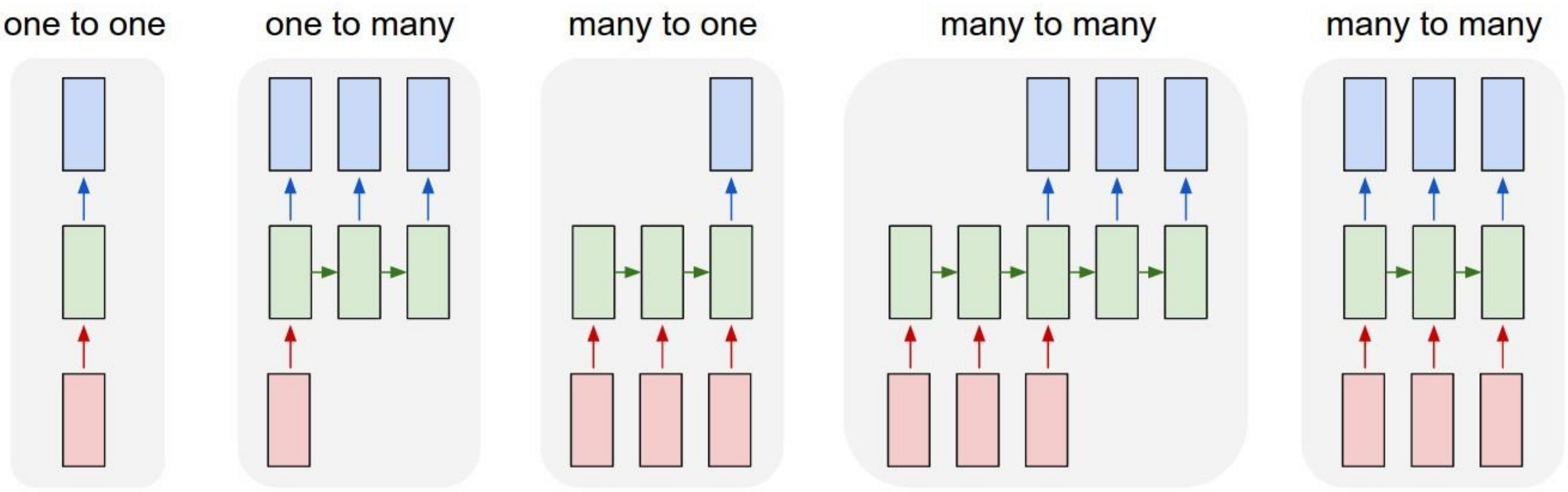


$$\boldsymbol{r}^t = \sigma(W_{xr}\boldsymbol{x}^t + W_{hr}\boldsymbol{h}^{t-1} + b_r)$$
$$\boldsymbol{z}^t = \sigma(W_{xz}\boldsymbol{x}^t + W_{hz}\boldsymbol{h}^{t-1} + b_z)$$

$$\widetilde{\boldsymbol{h}}^t = \tanh(W_{xh}\boldsymbol{x}^t + W_{hr}(\boldsymbol{r}^t \odot \boldsymbol{h}^{t-1}) + b_h)$$
$$\boldsymbol{h}^t = \boldsymbol{z}^t \odot \boldsymbol{h}^{t-1} + (1 - \boldsymbol{z}^t) \odot \widetilde{\boldsymbol{h}}^t$$

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

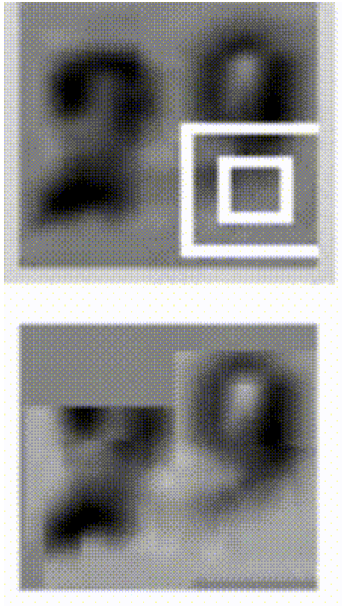
The Unreasonable Effectiveness of Recurrent Neural Networks



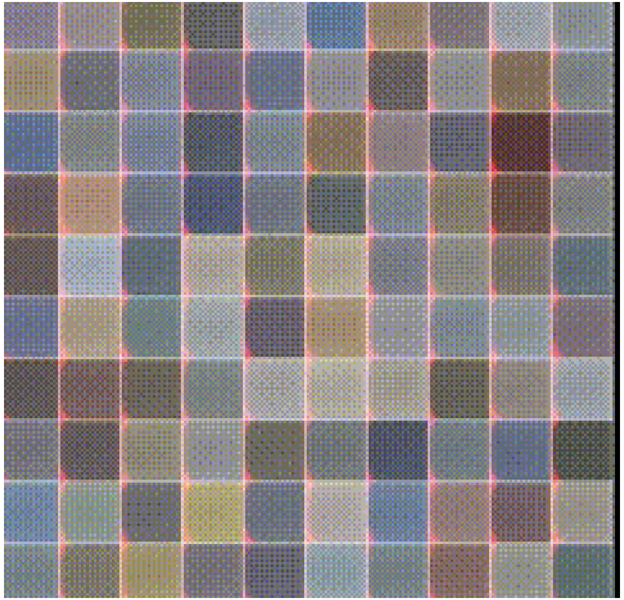
Example: Image captioning | Sentiment analysis | Machine Translation | Video classification

http://karpathy.github.io/2015/05/21/rnn-effectiveness/

Sequential processing in absence of sequences



RNN learns to
read house numbers



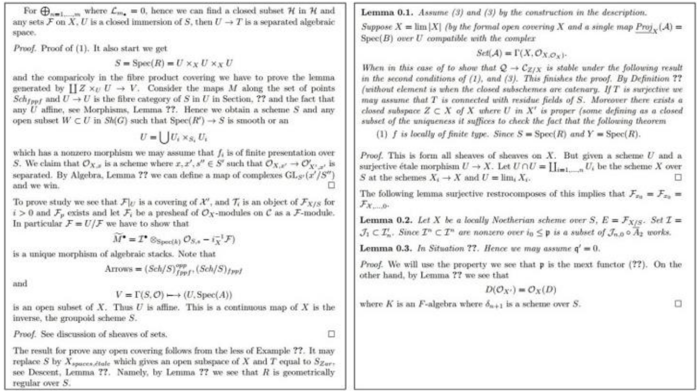
RNN learns to
paint house numbers

http://karpathy.github.io/2015/05/21/rnn-effectiveness/
Ba, J., Mnih, V., Kavukcuoglu, K. Multiple object recognition with visual attention. arXiv preprint arXiv:1412.7755.
Gregor, K. et al. 2015. Draw: A recurrent neural network for image generation. arXiv preprint arXiv:1502.04623.

Skoltech
Skolkovo Institute of Science and Technology

Text generation (now there are better alternatives)

- Paul Graham generator
- Shakespeare
- Wikipedia
- Algebraic Geometry (Latex)
- Linux Source Code
- Generating Baby Names



For $\bigoplus_{n=1,\ldots,m}$ where $\mathcal{L}_{m_\bullet} = 0$, hence we can find a closed subset $\mathcal{H}$ in $\mathcal{H}$ and any sets $\mathcal{F}$ on X, U is a closed immersion of S, then $U \to T$ is a separated algebraic space.

Proof. Proof of (1). It also start we get

$$S = \mathrm{Spec}(R) = U \times_X U \times_X U$$

and the comparicoly in the fibre product covering we have to prove the lemma generated by $\coprod Z \times_U U \to V$. Consider the maps M along the set of points Sch_{fppf} and $U \to U$ is the fibre category of S in U in Section, ?? and the fact that any U affine, see Morphisms, Lemma ??. Hence we obtain a scheme S and any open subset $W \subset U$ in $Sh(G)$ such that $\mathrm{Spec}(R') \to S$ is smooth or an

$$U = \bigcup U_i \times_{S_i} U_i$$

which has a nonzero morphism we may assume that f_i is of finite presentation over S. We claim that $\mathcal{O}_{X,x}$ is a scheme where $x, x', s'' \in S'$ such that $\mathcal{O}_{X,x'} \to \mathcal{O}'_{X',x'}$ is separated. By Algebra, Lemma ?? we can define a map of complexes $\mathrm{GL}_{S'}(x'/S'')$ and we win. □

To prove study we see that $\mathcal{F}|_U$ is a covering of $\mathcal{X}'$, and $\mathcal{T}_i$ is an object of $\mathcal{F}_{X/S}$ for $i > 0$ and $\mathcal{F}_p$ exists and let $\mathcal{F}_i$ be a presheaf of $\mathcal{O}_X$-modules on $\mathcal{C}$ as a $\mathcal{F}$-module. In particular $\mathcal{F} = U/\mathcal{F}$ we have to show that

$$\widetilde{M}^\bullet = \mathcal{I}^\bullet \otimes_{\mathrm{Spec}(k)} \mathcal{O}_{S,s} - i_X^{-1}\mathcal{F})$$

is a unique morphism of algebraic stacks. Note that

$$\mathrm{Arrows} = (Sch/S)^{opp}_{fppf}, (Sch/S)_{fppf}$$

and

$$V = \Gamma(S, \mathcal{O}) \longmapsto (U, \mathrm{Spec}(A))$$

is an open subset of X. Thus U is affine. This is a continuous map of X is the inverse, the groupoid scheme S.

Proof. See discussion of sheaves of sets. □

The result for prove any open covering follows from the less of Example ??. It may replace S by $X_{spaces,étale}$ which gives an open subspace of X and T equal to S_{Zar}, see Descent, Lemma ??. Namely, by Lemma ?? we see that R is geometrically regular over S.

Lemma 0.1. *Assume (3) and (3) by the construction in the description.*

Suppose $X = \lim |X|$ (by the formal open covering X and a single map $\underline{Proj}_X(\mathcal{A}) = \mathrm{Spec}(B)$ over U compatible with the complex

$$Set(\mathcal{A}) = \Gamma(X, \mathcal{O}_{X,\mathcal{O}_X}).$$

When in this case of to show that $\mathcal{Q} \to \mathcal{C}_{Z/X}$ is stable under the following result in the second conditions of (1), and (3). This finishes the proof. By Definition ?? (without element is when the closed subschemes are catenary. If T is surjective we may assume that T is connected with residue fields of S. Moreover there exists a closed subspace $Z \subset X$ of X where U in X' is proper (some defining as a closed subset of the uniqueness it suffices to check the fact that the following theorem

(1) f is locally of finite type. Since $S = \mathrm{Spec}(R)$ and $Y = \mathrm{Spec}(R)$.

Proof. This is form all sheaves of sheaves on X. But given a scheme U and a surjective étale morphism $U \to X$. Let $U \cap U = \coprod_{i=1,\ldots,n} U_i$ be the scheme X over S at the schemes $X_i \to X$ and $U = \lim_i X_i$. □

The following lemma surjective restrocomposes of this implies that $\mathcal{F}_{x_0} = \mathcal{F}_{x_0} = \mathcal{F}_{X,\ldots,0}$.

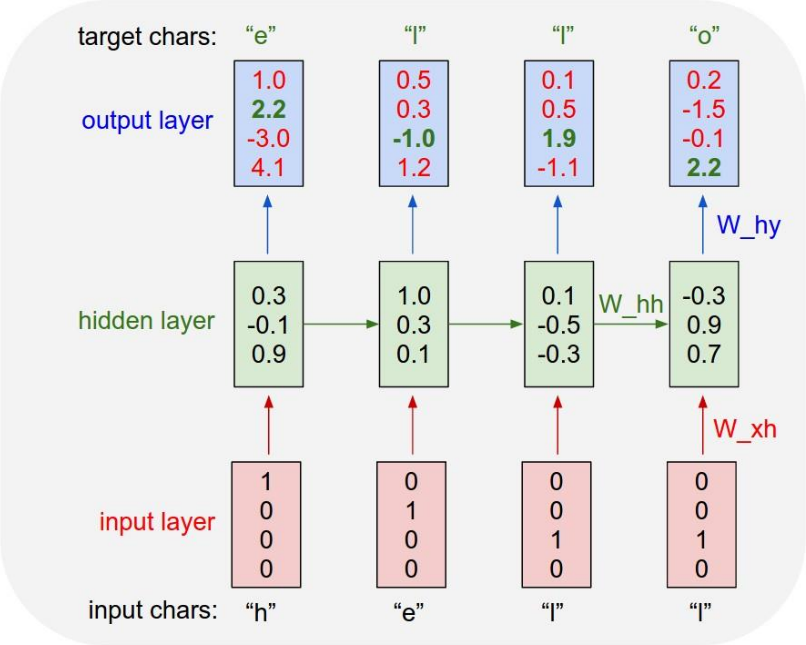
Lemma 0.2. *Let X be a locally Noetherian scheme over S, $E = \mathcal{F}_{X/S}$. Set $\mathcal{I} = \mathcal{J}_1 \subset \mathcal{I}'_n$. Since $\mathcal{I}^n \subset \mathcal{I}^n$ are nonzero over $i_0 \leq \mathfrak{p}$ is a subset of $\mathcal{J}_{n,0} \circ \overline{A}_2$ works.*

Lemma 0.3. *In Situation ??. Hence we may assume $\mathfrak{q}' = 0$.*

Proof. We will use the property we see that $\mathfrak{p}$ is the mext functor (??). On the other hand, by Lemma ?? we see that

$$D(\mathcal{O}_{X'}) = \mathcal{O}_X(D)$$

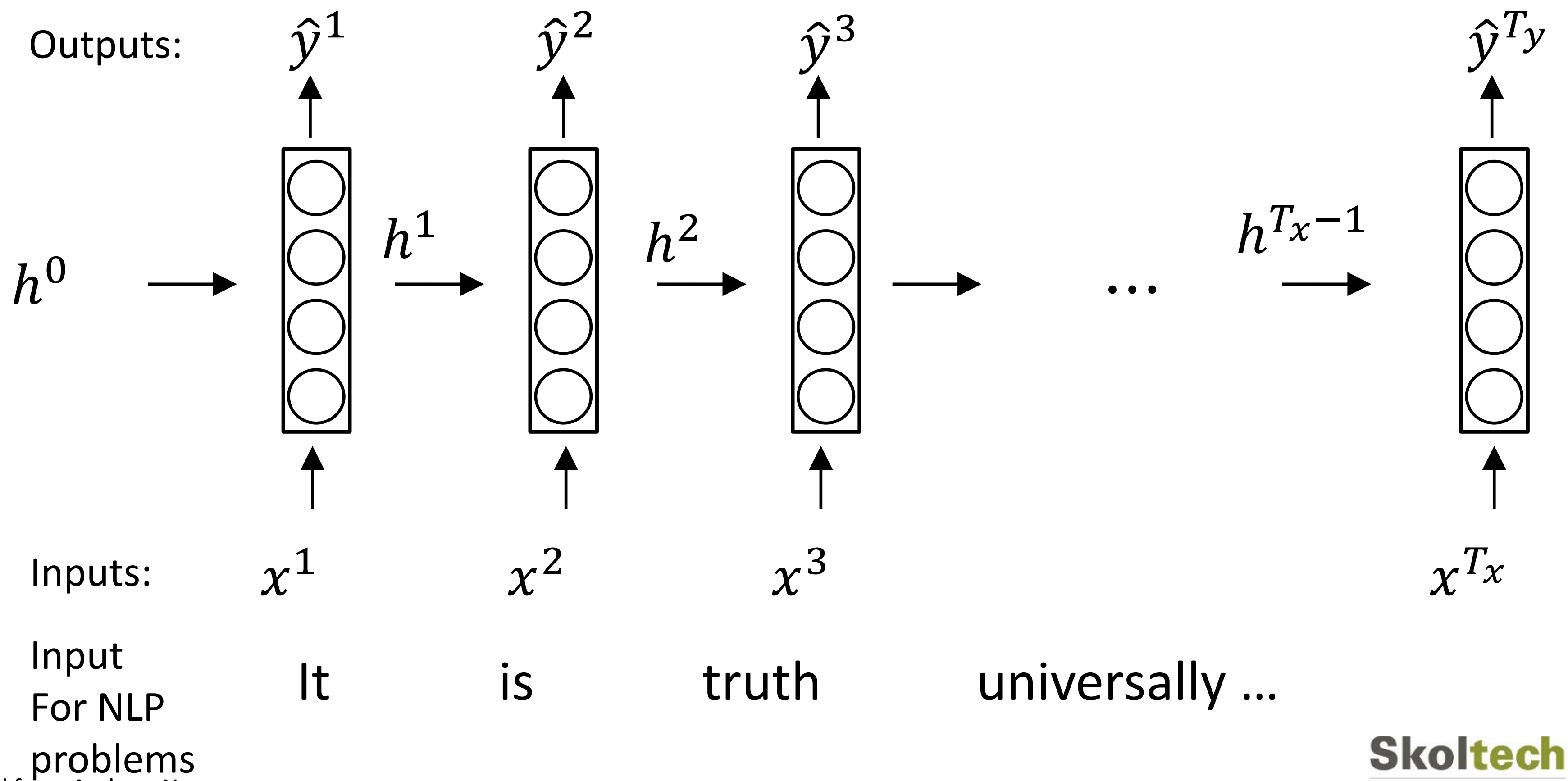
where K is an F-algebra where δ_{n+1} is a scheme over S. □



http://karpathy.github.io/2015/05/21/rnn-effectiveness/

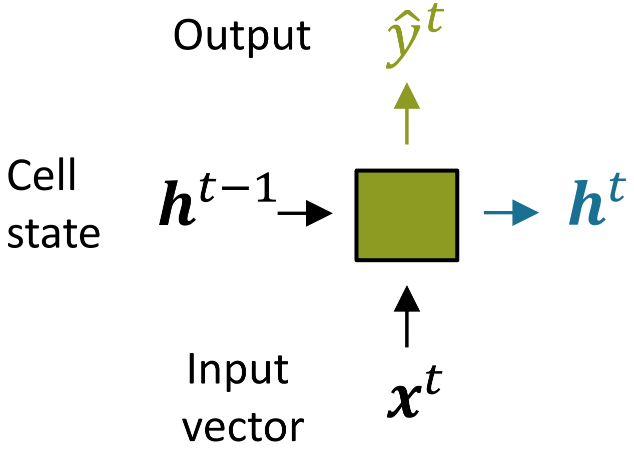
Skoltech
Skolkovo Institute of Science and Technology

Forward propagation through RNN



Adopted from Andrew Ng

Simple RNN model



$$\boldsymbol{h}^t = f_h(\boldsymbol{x}^t, \boldsymbol{h}^{t-1})$$

$$\boldsymbol{h}^t = \tanh(V\boldsymbol{x}^t + W\boldsymbol{h}^{t-1} + b_h)$$

$$\hat{y}^t = f_y(\boldsymbol{h}^t)$$

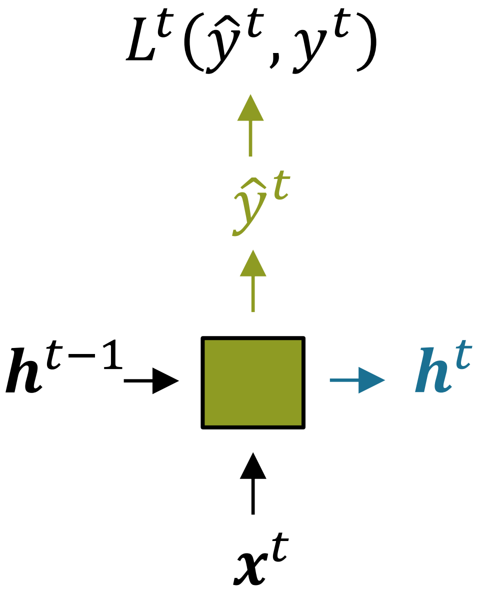
$$\hat{y}^t = \mathrm{softmax}(U\boldsymbol{h}^t + b_y)$$

Backpropagation w.r.t. U

$$L = \sum_{i=1}^{T_y} L^i(\hat{y}^i, y^i)$$

$$\frac{\partial L}{\partial U} = \sum_{i=1}^{T_y} \frac{\partial L_i}{\partial U} = \sum_{i=1}^{T_y} \frac{\partial L_i}{\partial \hat{y}^i} \frac{\partial \hat{y}^i}{\partial U}$$

$$\hat{y}^t = \mathrm{softmax}(U\boldsymbol{h}^t + b_y)$$



Backpropagation w.r.t. W

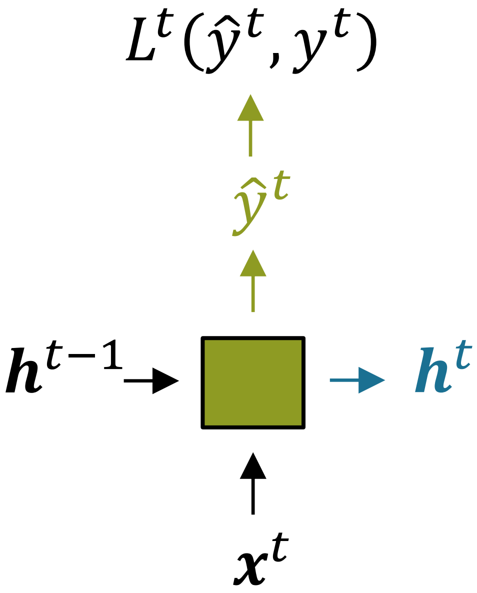
$$L = \sum_{i=1}^{T_y} L^i(\hat{y}^i, y^i)$$

$$\boldsymbol{h}^t = \tanh(V\boldsymbol{x}^t + W\boldsymbol{h}^{t-1} + b_h)$$

$$\frac{\partial L_i}{\partial \hat{y}^i}\frac{\partial \hat{y}^i}{\partial W} = \frac{\partial L_i}{\partial \hat{y}^i}\frac{\partial \hat{y}^i}{\partial h_t}\left(\frac{\partial h_t}{\partial W} + \frac{\partial h_t}{\partial h_{t-1}}\frac{\partial h_{t-1}}{\partial W} + \cdots\right)$$

$$\frac{\partial L_i}{\partial \hat{y}^i}\frac{\partial \hat{y}^i}{\partial W} = \frac{\partial L_i}{\partial \hat{y}^i}\frac{\partial \hat{y}^i}{\partial h_t}\sum_{i=0}^{T_y}\left(\prod_{j=i+1}^{T_y}\frac{\partial h_j}{\partial h_{j-1}}\right)\frac{\partial h_i}{\partial W}$$

$$\hat{y}^t = \mathrm{softmax}(U\boldsymbol{h}^t + b_y)$$



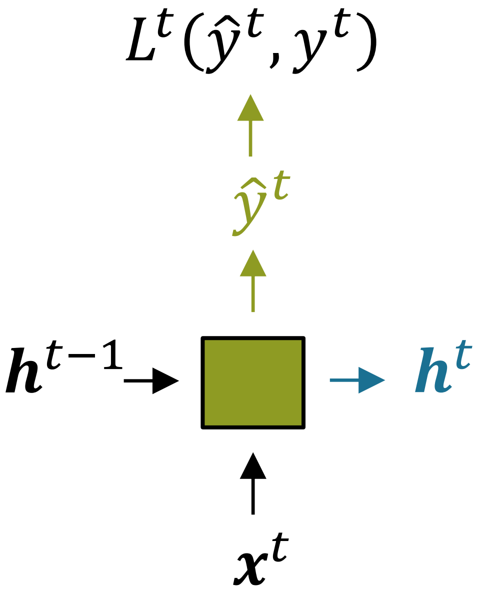
Jacobian matrix is the cause of all problems

$$\boldsymbol{h}^t = \tanh(V\boldsymbol{x}^t + W\boldsymbol{h}^{t-1} + b_h)$$

$$\hat{y}^t = \mathrm{softmax}(U\boldsymbol{h}^t + b_y)$$

$$\frac{\partial L_i}{\partial \hat{y}^i}\frac{\partial \hat{y}^i}{\partial W} = \frac{\partial L_i}{\partial \hat{y}^i}\frac{\partial \hat{y}^i}{\partial h_t}\left(\frac{\partial h_t}{\partial W} + \frac{\partial h_t}{\partial h_{t-1}}\frac{\partial h_{t-1}}{\partial W} + \cdots\right)$$

$$\frac{\partial L_i}{\partial \hat{y}^i}\frac{\partial \hat{y}^i}{\partial W} = \frac{\partial L_i}{\partial \hat{y}^i}\frac{\partial \hat{y}^i}{\partial h_t}\sum_{i=0}^{T_y}\left(\prod_{j=i+1}^{T_y}\frac{\partial h_j}{\partial h_{j-1}}\right)\frac{\partial h_i}{\partial W}$$



$$\left\|\frac{\partial h_j}{\partial h_{j-1}}\right\|_2 > 1$$ gradient exploration

$$\left\|\frac{\partial h_j}{\partial h_{j-1}}\right\|_2 < 1$$ gradient vanishing

http://cs224d.stanford.edu/notebooks/vanishing_grad_example.html

Skoltech
Skolkovo Institute of Science and Technology

Gradient exploding and vanishing

$$\boldsymbol{h}^j = f\left(V\boldsymbol{x}^j + W\boldsymbol{h}^{j-1} + b_h\right) = f(\boldsymbol{z}_t)$$

$$\frac{\partial \boldsymbol{h}_j}{\partial \boldsymbol{h}_{j-1}} = \frac{\partial \boldsymbol{h}_j}{\partial \boldsymbol{z}_j} \frac{\partial \boldsymbol{z}_j}{\partial \boldsymbol{h}_{j-1}} = \mathrm{diag}(f'(\boldsymbol{z}_t))W$$

$$\left\|\frac{\partial \boldsymbol{h}_j}{\partial \boldsymbol{h}_{j-1}}\right\| \leq \left\|\mathrm{diag}\left(f'(\boldsymbol{z}_t)\right)\right\| \cdot \|W\|$$

Skoltech
Skolkovo Institute of Science and Technology

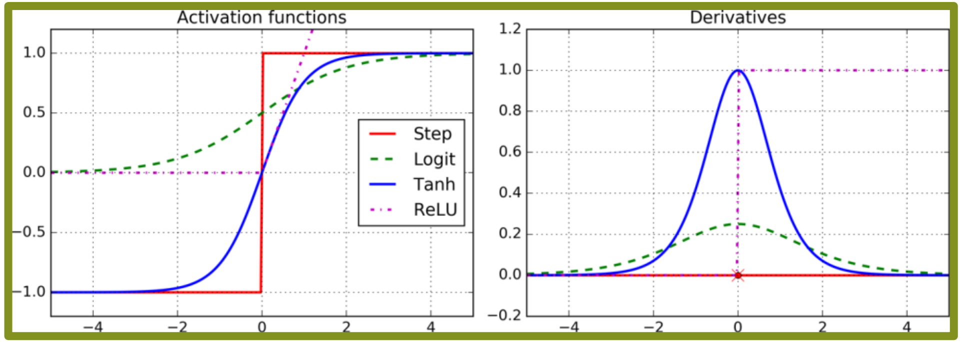
Gradient exploding and vanishing

$$\boldsymbol{h}^j = f\left(V\boldsymbol{x}^j + W\boldsymbol{h}^{j-1} + b_h\right) = f(\boldsymbol{z}_t)$$

$$\frac{\partial \boldsymbol{h}_j}{\partial \boldsymbol{h}_{j-1}} = \frac{\partial \boldsymbol{h}_j}{\partial \boldsymbol{z}_j}\frac{\partial \boldsymbol{z}_j}{\partial \boldsymbol{h}_{j-1}} = \mathrm{diag}(f'(\boldsymbol{z}_t))W$$

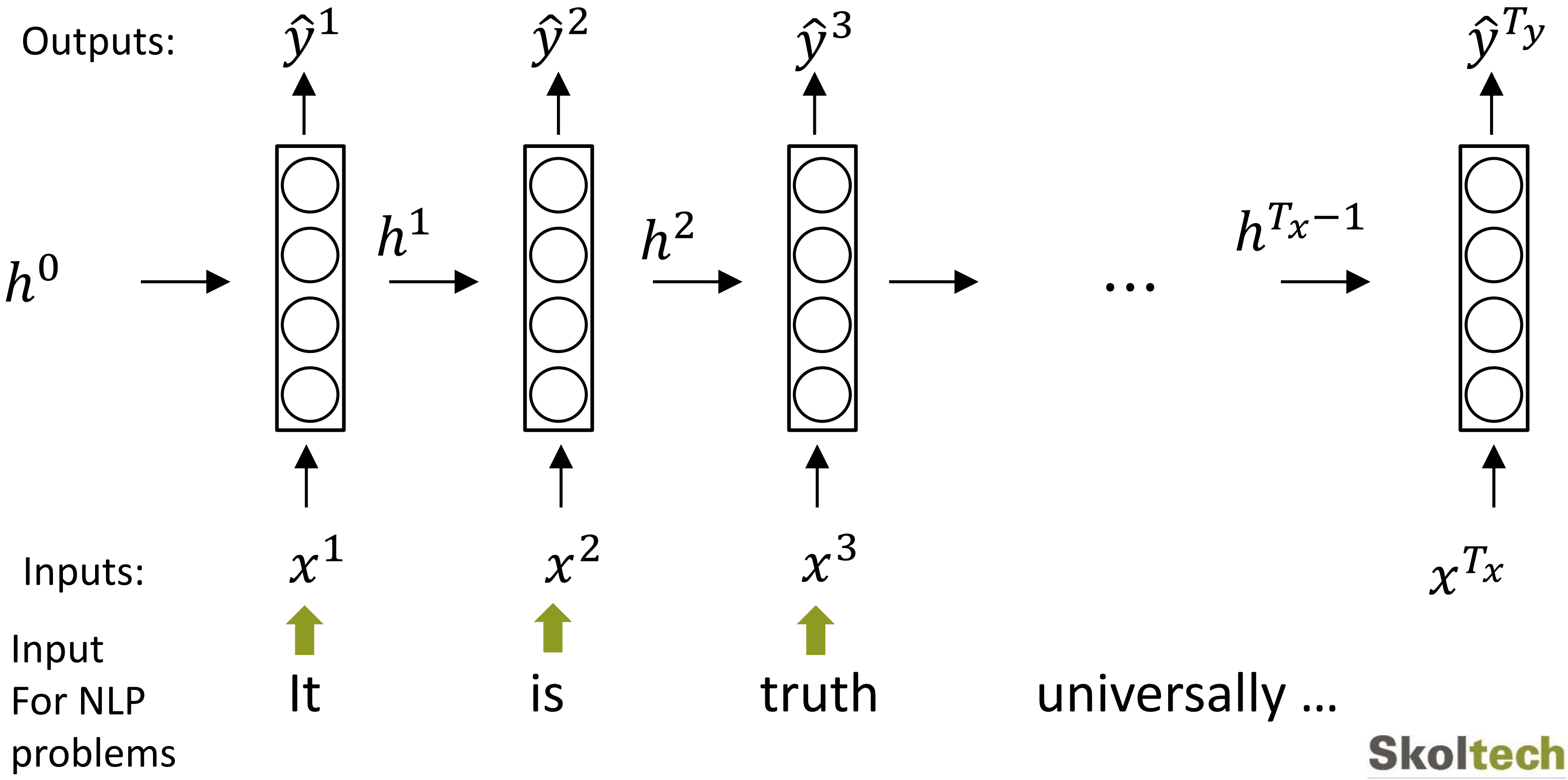
activation function
– ReLU is the best option

- The initialization matters
  For orthogonal matrix $Q^T = Q^{-1}$
  all eigenvalues equals one
- Regularization



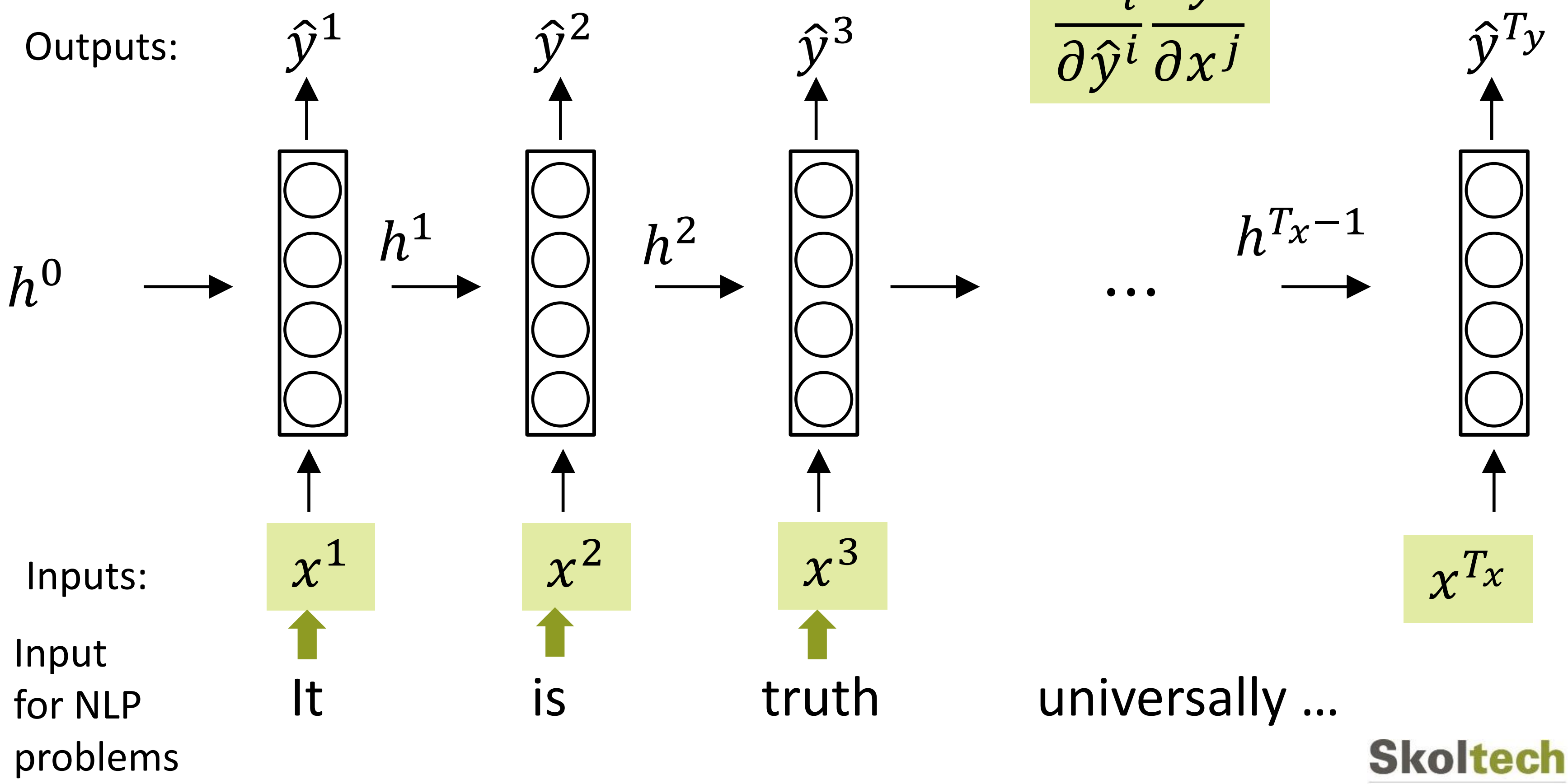
Skoltech
Skolkovo Institute of Science and Technology

Embedding layer for RNNs



Backpropagation for embedding layer for RNNs

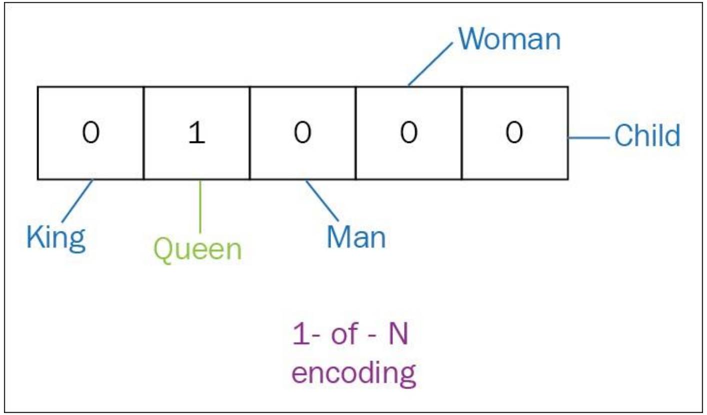
$$\frac{\partial L_i}{\partial \hat{y}^i}\frac{\partial \hat{y}^i}{\partial x^j}$$



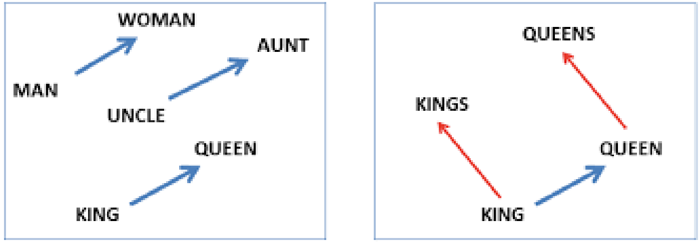
Learning of embeddings

End2end learning of embeddings for:

- Words
- Codes of transactions (gas station)
- Prescription drugs
- Symbols or groups of symbols



One-hot encoding is also an embeddings



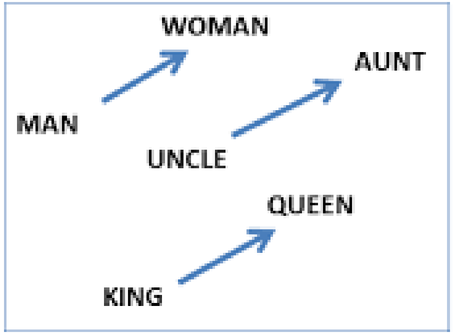
But we want something smarter

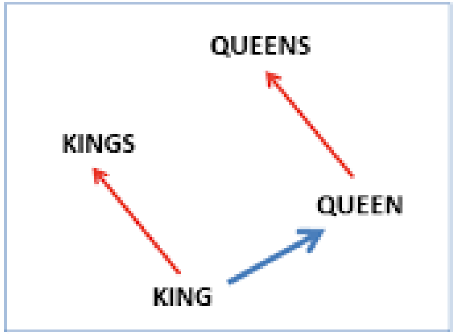
Why we need embeddings?

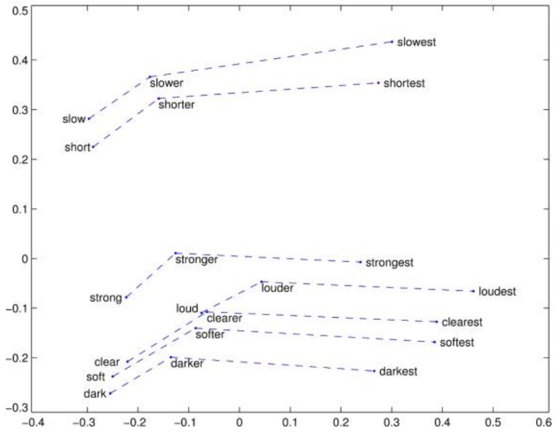
Paris – France
Moscow – Russia
Vienna – Austria
Antananarivo – Madagascar

In NLP it is hard to measure quality of a solution, for example in Machine translation

- Check that we learn something adequate by comparing feature vectors for objects with similar relationships



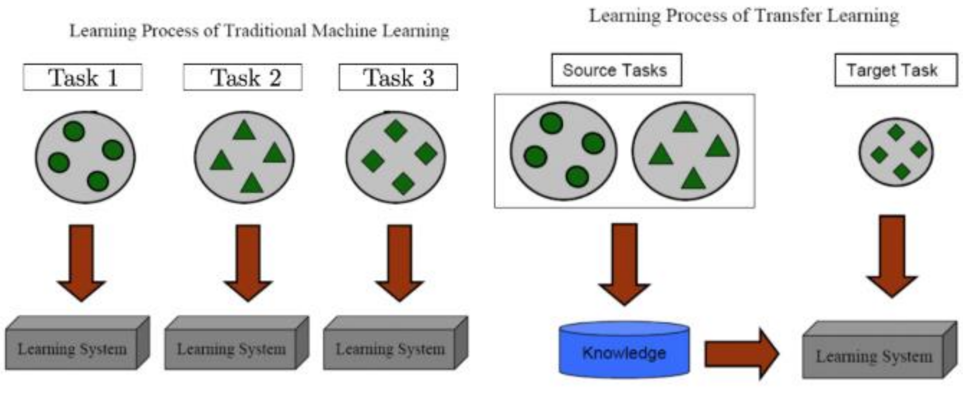




Why we need embeddings?

We finally have representation learning –
but now for sequences

- Use embeddings for transfer learning



(a) Traditional Machine Learning (b) Transfer Learning

Why we need embeddings?

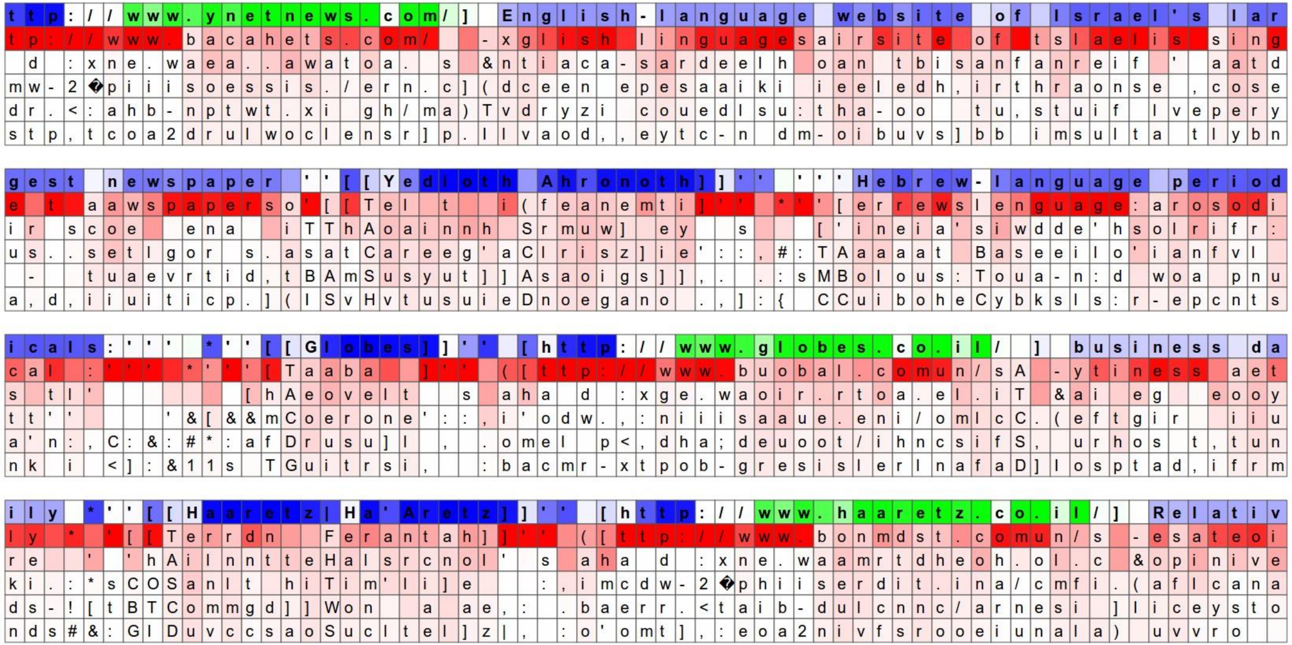
Natural way to look at tokens

- Use embeddings to find their similarity

	animal	fluffiness	dangerous	spooky
aardvark	0.97	0.03	0.15	0.04
black	0.07	0.01	0.20	0.95
cat	0.98	0.98	0.45	0.35
duvet	0.01	0.84	0.12	0.02
zombie	0.74	0.05	0.98	0.93

Visualizing embeddings

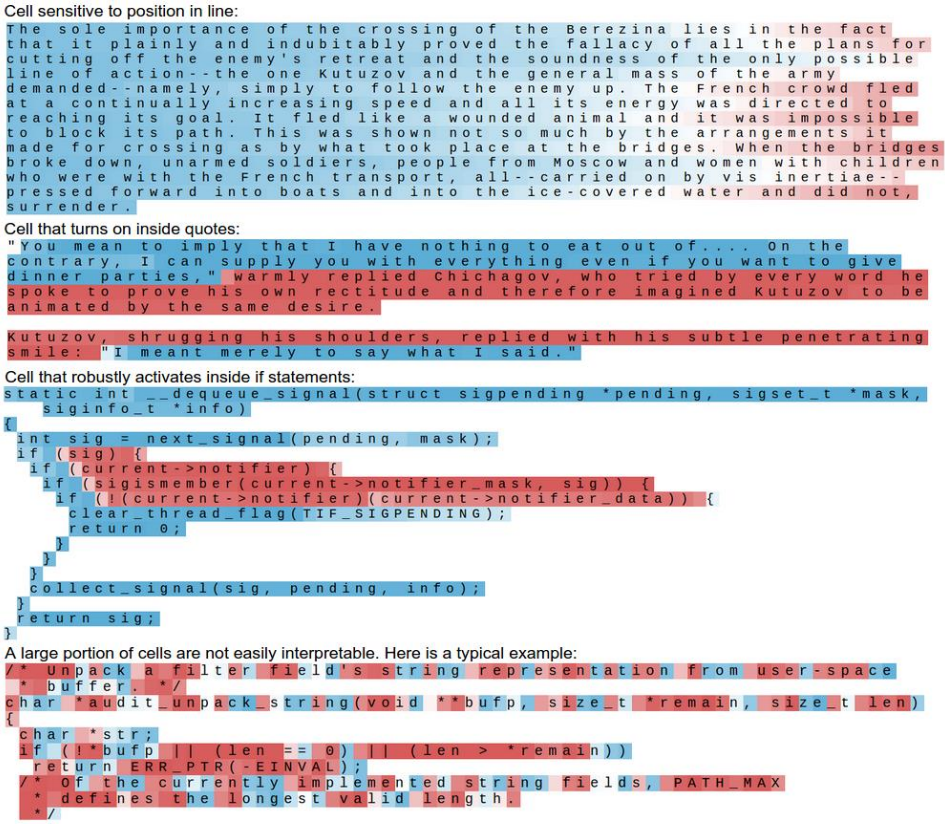
Character-level LSTM, look at some dimensions



The neuron highlighted in this image seems to get very excited about URLs and turns off outside of the URLs. The LSTM is likely using this neuron to remember if it is inside a URL or not.

Visualizing embeddings

Character-level LSTM, look at some dimensions

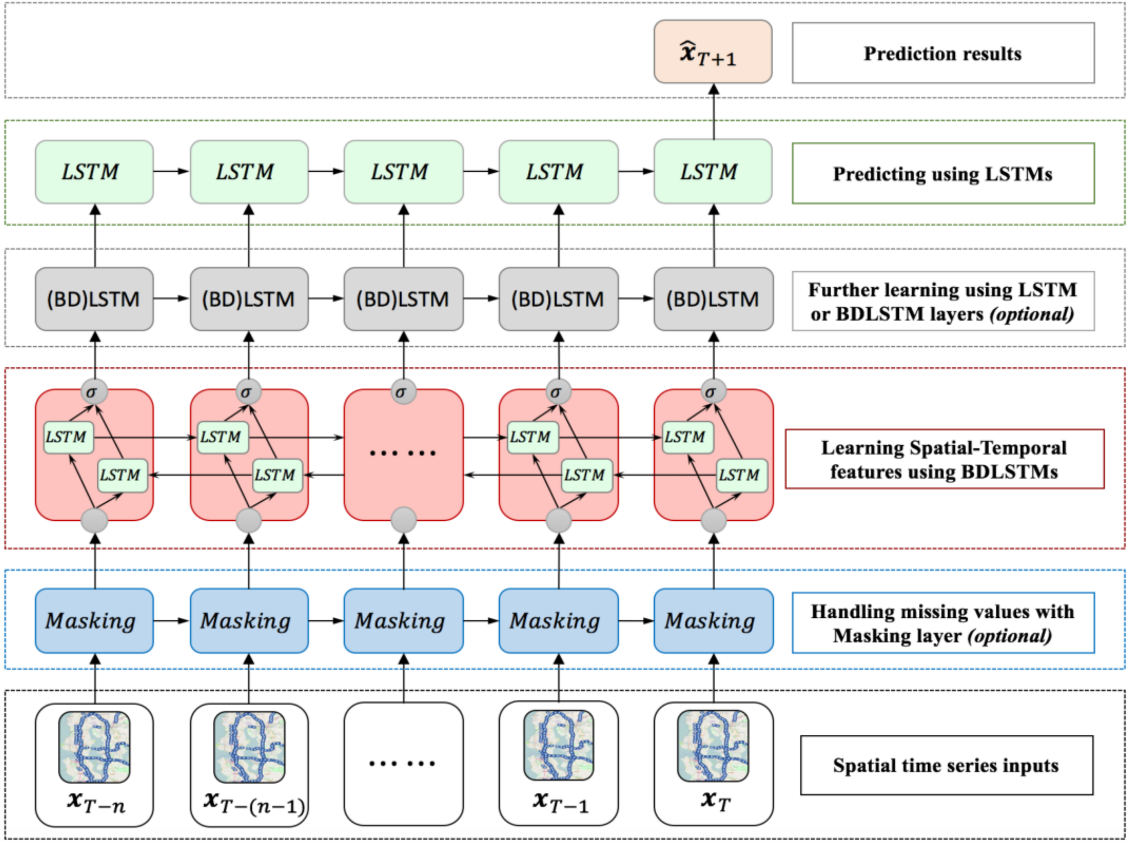


Look at this video
for more examples
for LSTMViz

http://lstm.seas.harvard.edu/

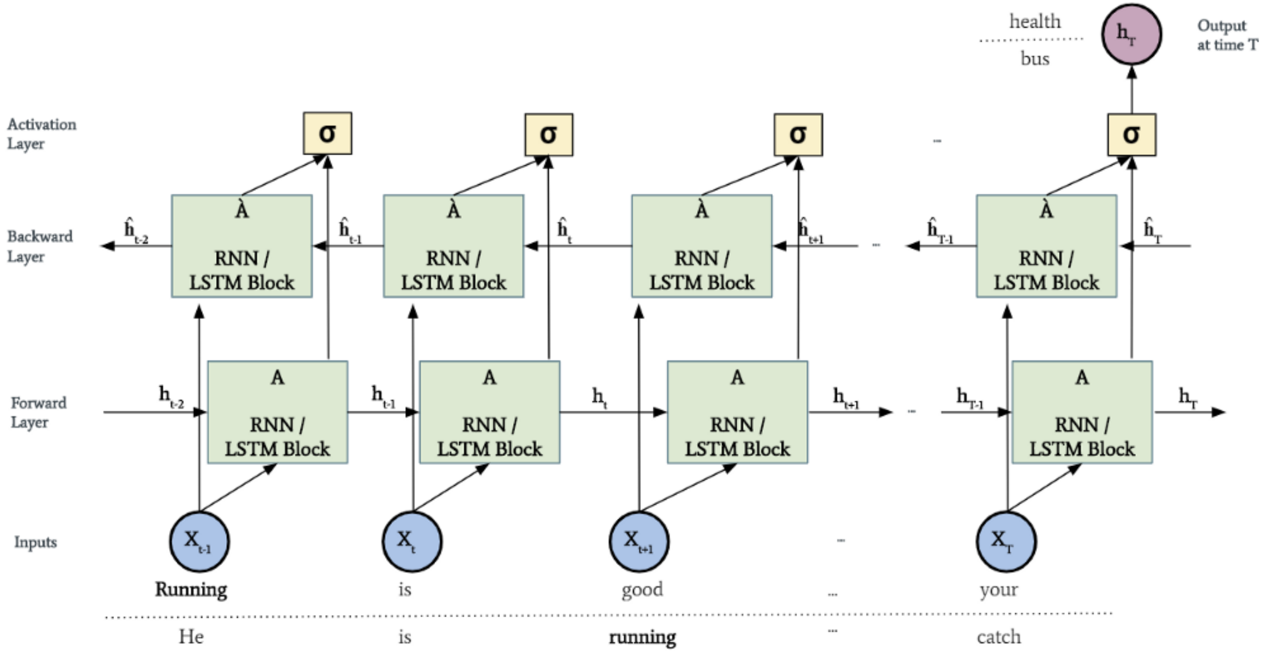
Skoltech
Skolkovo Institute of Science and Technology

Various architectures: Going deeper



Stacked Bidirectional and Unidirectional LSTM Recurrent Neural Network for Network-wide Traffic Speed Prediction Zhiyong Cui, Ruimin Ke, Ziyuan Pu, Yinhai Wang

Various architectures: Bidirectional LSTM

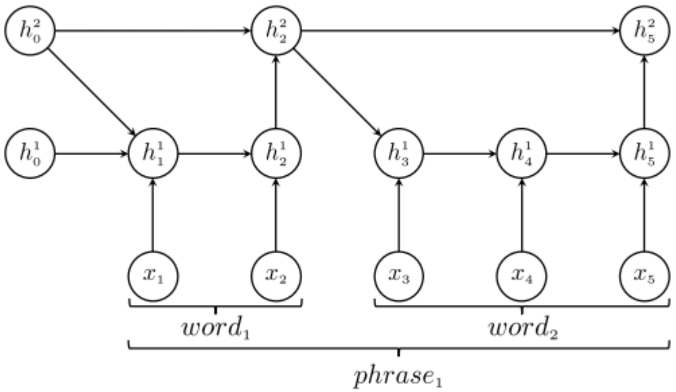


https://mc.ai/review-generation-using-bidirectional-long-short-term-memorylstm/

Mike Schuster and Kuldip K. Paliwal, Bidirectional Recurrent Neural Networks, Trans. on Signal Processing 1997

Various architectures: Hierarchical RNN

Our goal is to use hierarchical structure in data

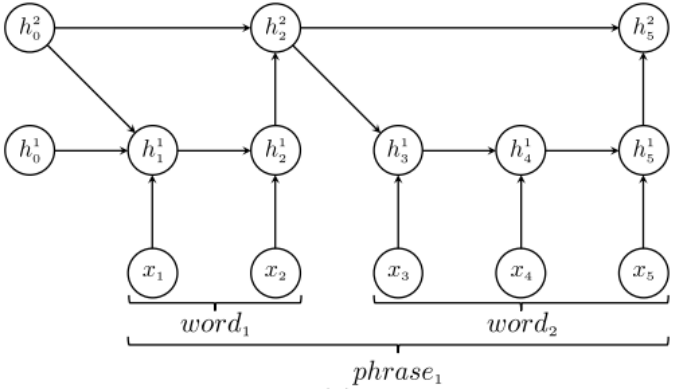


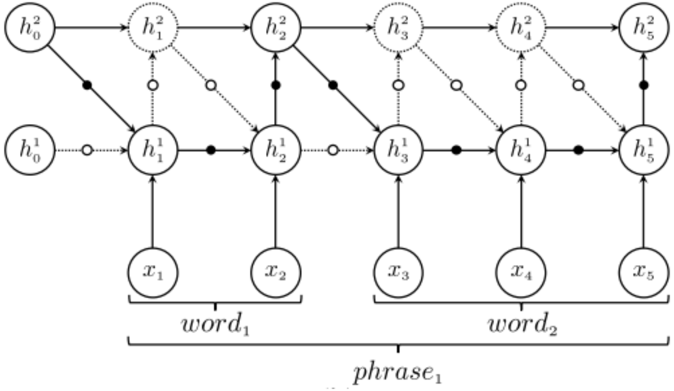
Hierarchical structure can be provided
and used with the following architecture

Ling, W. et al. 2015. Character-based neural machine translation. arXiv preprint
Sordoni A. et al. 2015. A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. ACM CIKM. 2015.

Various architectures: Hierarchical RNN

Our goal is to discover hierarchical structure in data





Hierarchical structure can be provided and used

Hierarchical structure can be learned

The boundary detector is the key

Chung, J., Ahn, S., & Bengio, Y. 2016. Hierarchical multiscale recurrent neural networks. arXiv preprint arXiv:1609.01704. ICLR 2017.

Various architectures: Hierarchical RNN

Finally some formulas!

$$\mathbf{h}_t^\ell, \mathbf{c}_t^\ell, z_t^\ell = f_{\text{HM-LSTM}}^\ell(\mathbf{c}_{t-1}^\ell, \mathbf{h}_{t-1}^\ell, \mathbf{h}_t^{\ell-1}, \mathbf{h}_{t-1}^{\ell+1}, z_{t-1}^\ell, z_t^{\ell-1}). \tag{1}$$

Here, $\mathbf{h}$ and $\mathbf{c}$ denote the hidden and cell states, respectively. The function $f_{\text{HM-LSTM}}^\ell$ is implemented as follows. First, using the two boundary states z_{t-1}^ℓ and $z_t^{\ell-1}$, the cell state is updated by:

$$\mathbf{c}_t^\ell = \begin{cases} \mathbf{f}_t^\ell \odot \mathbf{c}_{t-1}^\ell + \mathbf{i}_t^\ell \odot \mathbf{g}_t^\ell & \text{if } z_{t-1}^\ell = 0 \text{ and } z_t^{\ell-1} = 1 \text{ (UPDATE)} \\ \mathbf{c}_{t-1}^\ell & \text{if } z_{t-1}^\ell = 0 \text{ and } z_t^{\ell-1} = 0 \text{ (COPY)} \\ \mathbf{i}_t^\ell \odot \mathbf{g}_t^\ell & \text{if } z_{t-1}^\ell = 1 \text{ (FLUSH)}, \end{cases} \tag{2}$$

and then the hidden state is obtained by:

$$\mathbf{h}_t^\ell = \begin{cases} \mathbf{h}_{t-1}^\ell & \text{if COPY}, \\ \mathbf{o}_t^\ell \odot \mathtt{tanh}(\mathbf{c}_t^\ell) & \text{otherwise}. \end{cases} \tag{3}$$

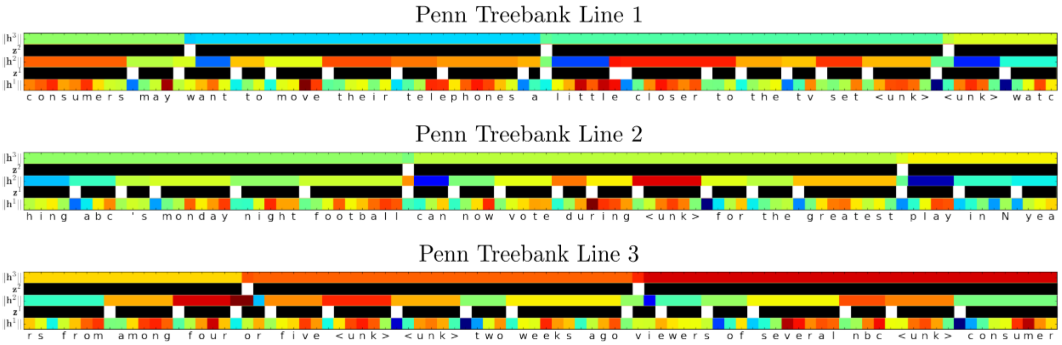
But boundary states are discrete. How can we optimize them?

Chung, J., Ahn, S., & Bengio, Y. 2016. Hierarchical multiscale recurrent neural networks. arXiv preprint arXiv:1609.01704. ICLR 2017.

Various architectures: Hierarchical RNN

- Straight-through estimator
- REINFORCE

← Simpler but biased approached used by the authors



Learned hierarchical structure

Also: the slope annealing trick

Increase slope of a sigmoid during training to turn it into a step function

Chung, J., Ahn, S., & Bengio, Y. 2016. Hierarchical multiscale recurrent neural networks. arXiv preprint arXiv:1609.01704. ICLR 2017.

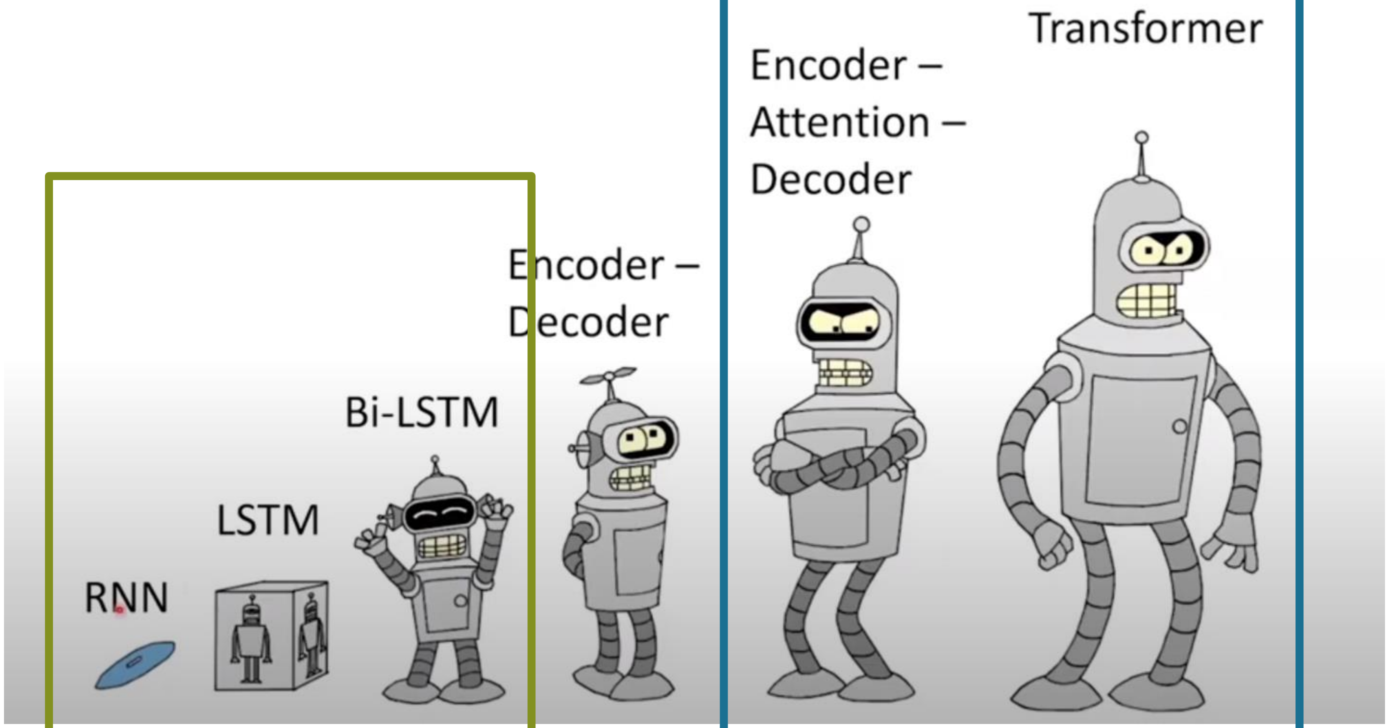
Other issues with RNNs

- Technical details e.g. Batch normalization or Dropout
- Memory mechanism
- Many applied problems solved with RNNs
- Theoretical view on RNNs
    - They are Turing complete

Siegelmann and Sontag. 1992. On the computational power of neural nets

Take-home messages

- Problems with gradients are intrinsic to RNN architecture, but can be fixed with LSTM or GRU
- Embedding is a fundamental concept: it is representation learning for RNNs
- We still can solve various complex problems using various architectures



This week

Next week

Sources

- https://github.com/kjw0612/awesome-rnn#blogs
- Recurrent Neural Networks | in Deep learning course by MIT 6.S191
- Coursera course on Sequence models
  https://www.coursera.org/learn/nlp-sequence-models
- http://colah.github.io/posts/2015-08-Understanding-LSTMs/