Recurrent Neural Networks

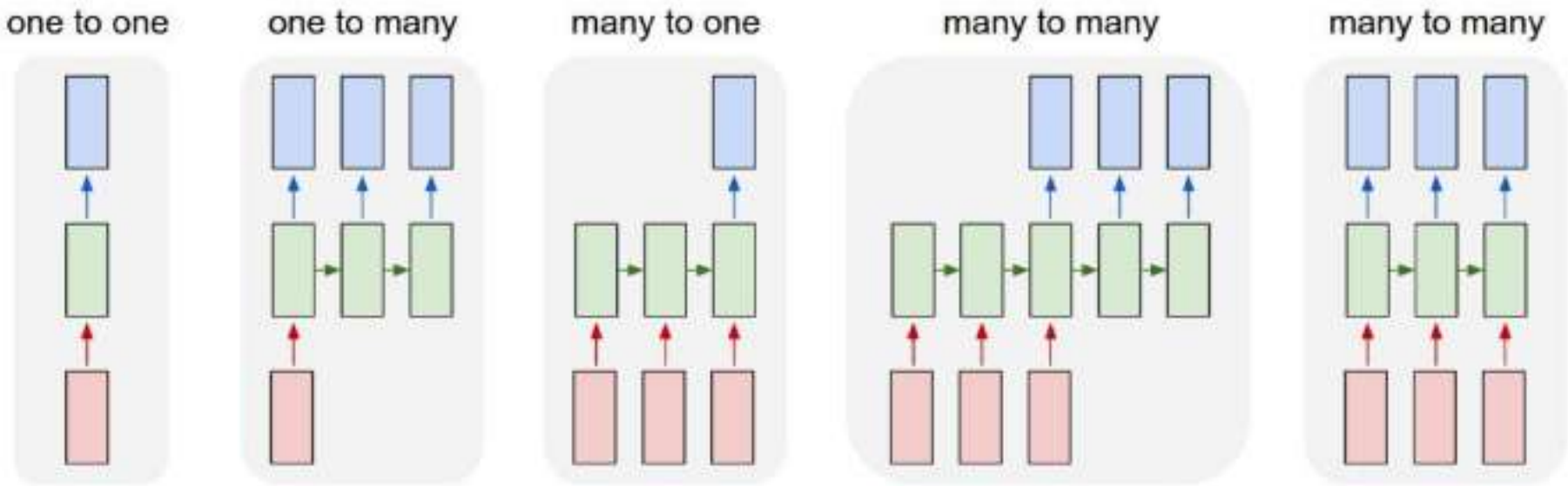
E. Scornet

Fall 2020

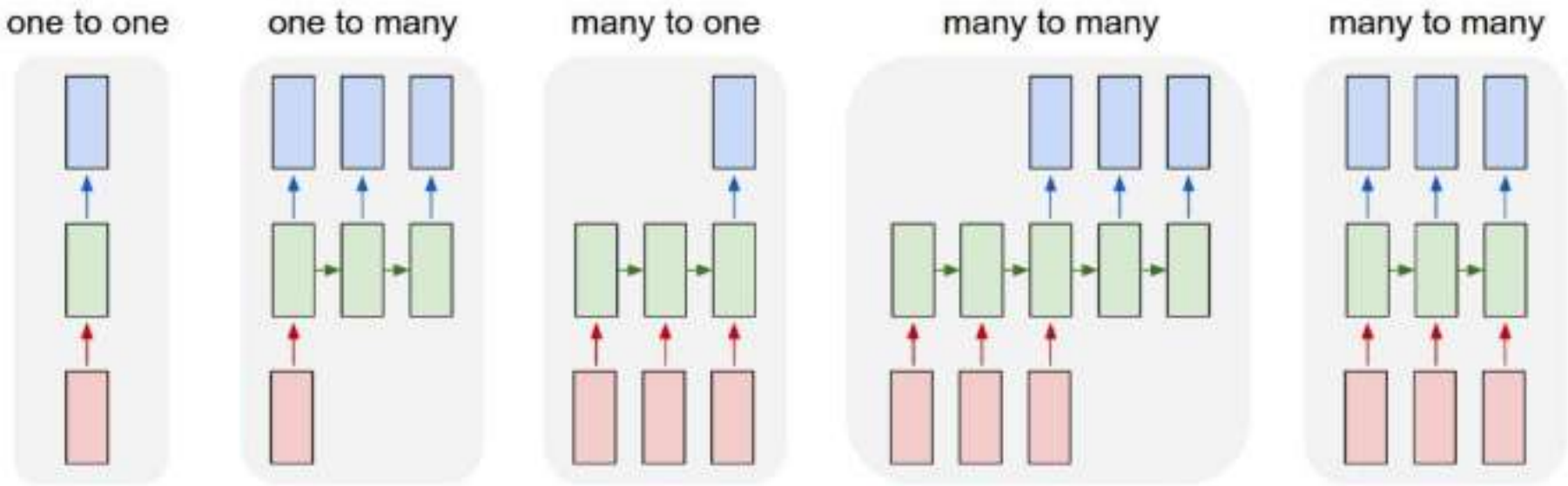
Outline

1. Introduction
2. RNN architectures
3. Vanishing/Exploding gradient: problem and solutions
   - Vanishing/Exploding gradient
   - Penalization - Weight initialization
   - GRU and LSTM
4. Miscellaneous
   - Truncated backpropagation
   - Regularization
5. Applications
   - Scene labeling: image/image
   - Image Captioning: image/sequence of words
   - Sentiment classification: sequence of words/sentiment
   - Speech synthesis/recognition
   - Video classification on frame level: sequence of image/sequence of label
   - Generating text/music

RNNs offer a lot of variability



RNNs offer a lot of variability



- Vanilla Neural Networks
- Image Captioning: image/sequence of words
- Sentiment classification: sequence of words/sentiment
- Translation: sequence of words/sequence of words
- Video classification on frame level: sequence of images/sequence of labels

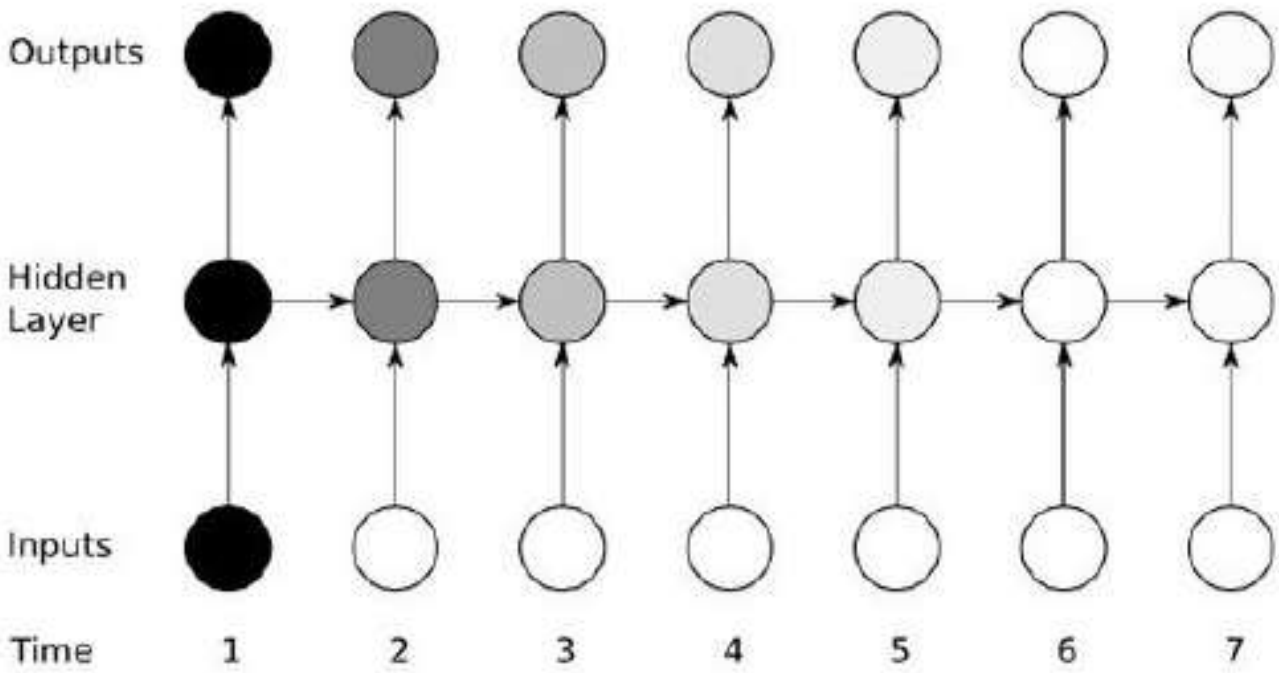
Limitations of Neural Networks

- ANNs can't deal with sequential or "temporal" data
- ANNs lack memory
- ANNs have a fixed architecture: fixed input size and a fixed output size
- RNNs are more "biologically realistic" because of recurrent connectivities found in the visual cortex of the brain

Outline

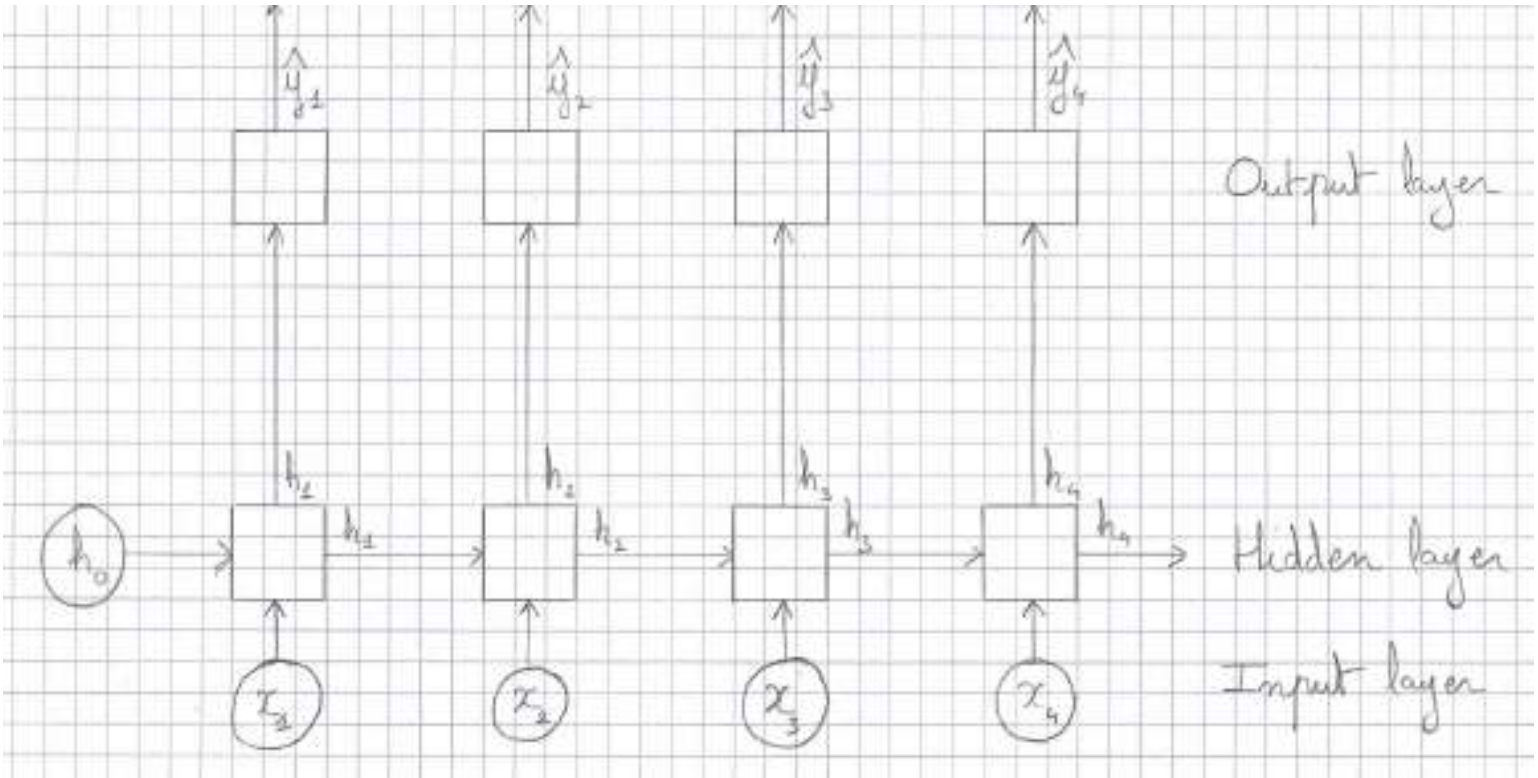
1. Introduction
2. RNN architectures
3. Vanishing/Exploding gradient: problem and solutions
   - Vanishing/Exploding gradient
   - Penalization - Weight initialization
   - GRU and LSTM
4. Miscellaneous
   - Truncated backpropagation
   - Regularization
5. Applications
   - Scene labeling: image/image
   - Image Captioning: image/sequence of words
   - Sentiment classification: sequence of words/sentiment
   - Speech synthesis/recognition
   - Video classification on frame level: sequence of image/sequence of label
   - Generating text/music

RNN Structure

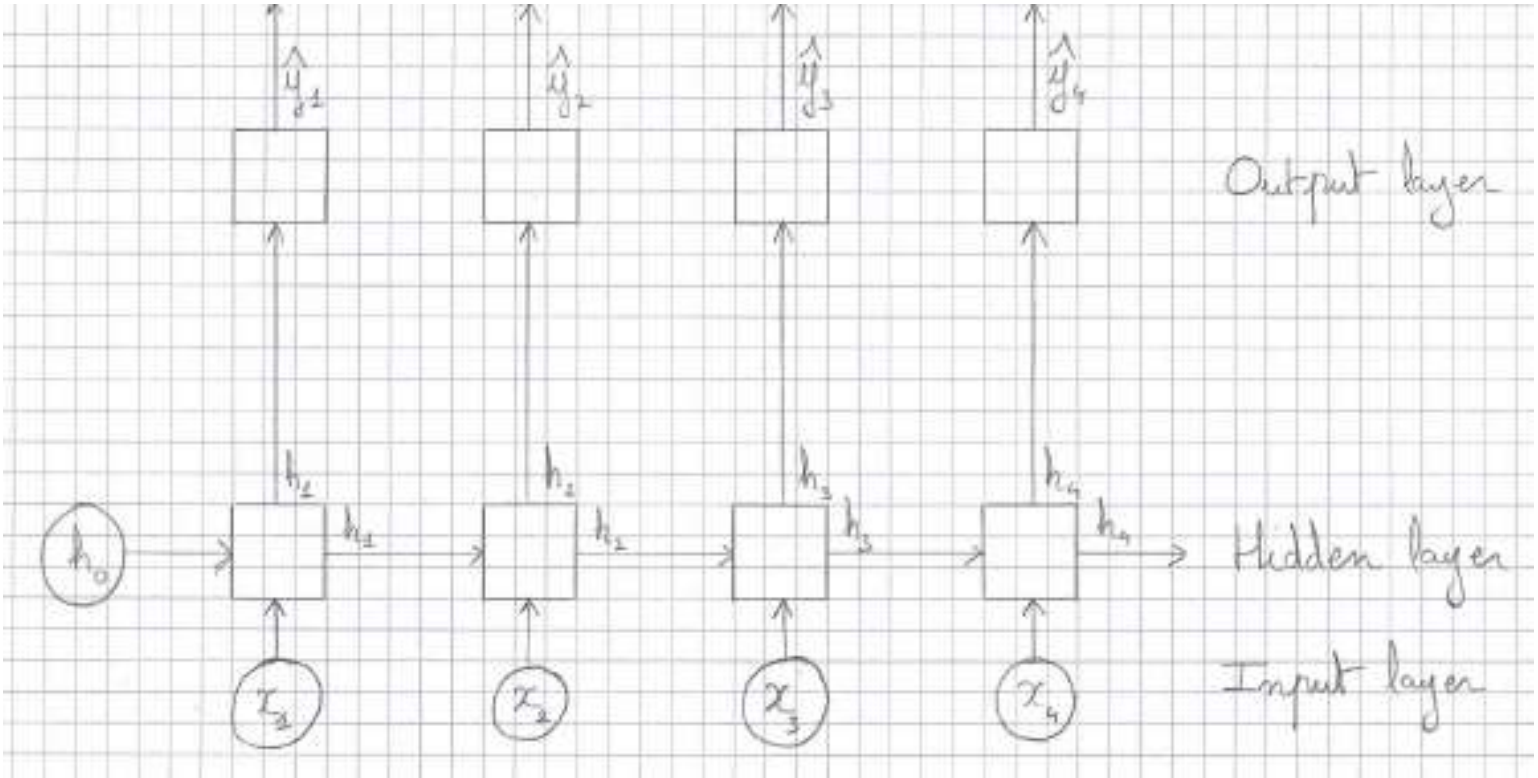


Definition of RNN

- Input layer - Data comes sequentially: $\mathbf{x}_1, \mathbf{x}_2, \ldots$
- Hidden Layer - Hidden state of the network at time t: $\mathbf{h}_t$
- Output layer - For the input x_t, the prediction is given by $\hat{y}_t$



Definition of RNN



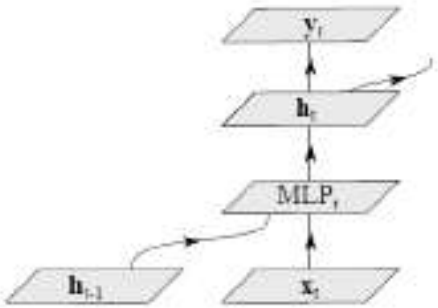
Hidden neuron:

$$\mathbf{h}_t = \tanh(W_{HH}\mathbf{h}_{t-1} + W_{IH}\mathbf{x}_t + \mathbf{b}_h)$$

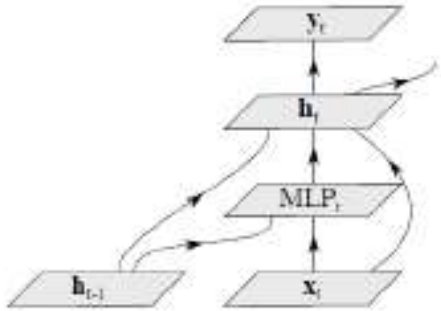
Output neuron:

$$\hat{y}_t = \mathrm{softmax}(W_{HO}\mathbf{h}_t + \mathbf{b}_{out})$$

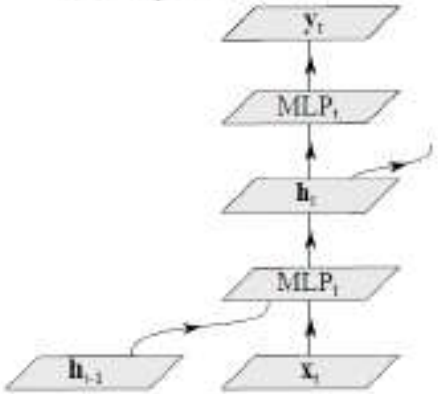
Deep RNN



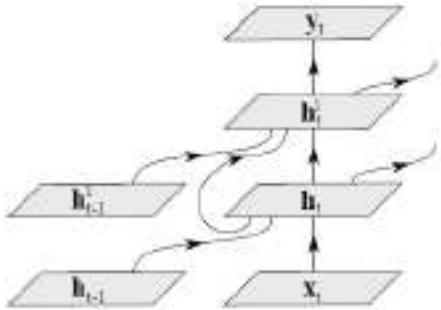
(a) Input to hidden.



(b) Input to hidden with short-cut.



(c) Hidden to hidden and output.



(d) Stack of hidden states.

Bi-directional RNN

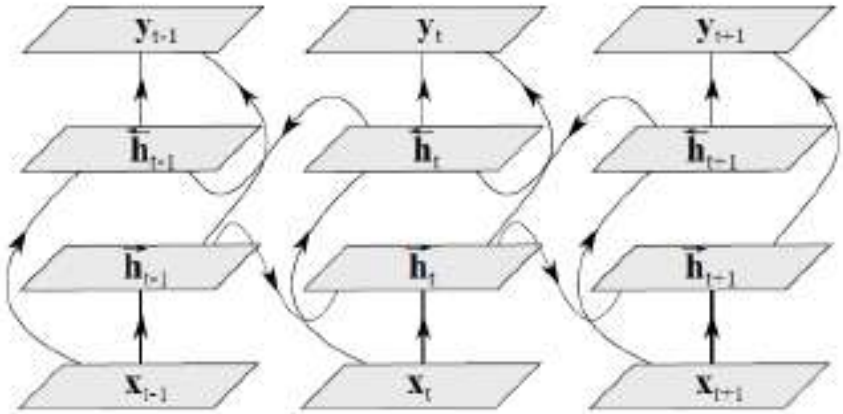


Figure: bi-directional recurrent neural network (BRNN)

$$\mathbf{y}_t = W_{\overrightarrow{HO}}\overrightarrow{\mathbf{h}_t} + W_{\overleftarrow{HO}}\overleftarrow{\mathbf{h}_t} + \mathbf{b}_o$$

Other types of RNN

- Multi-dimensional Recurrent Neural Networks

  ["Multi-Dimensional Recurrent Neural Networks", Graves, Fernández, and Jürgen Schmidhuber 2007]

- Recurrent Convolutional neural networks

  ["Recurrent convolutional neural network for object recognition", Liang and Hu 2015]

- Differential recurrent neural networks

  ["Differential recurrent neural networks for action recognition", Veeriah et al. 2015]

- Structurally Constrained Recurrent Neural Networks.

  ["Learning longer memory in recurrent neural networks", Tomas Mikolov, Joulin, et al. 2014]

Review on RNN: ["Recent Advances in Recurrent Neural Networks", Salehinejad et al. 2017]

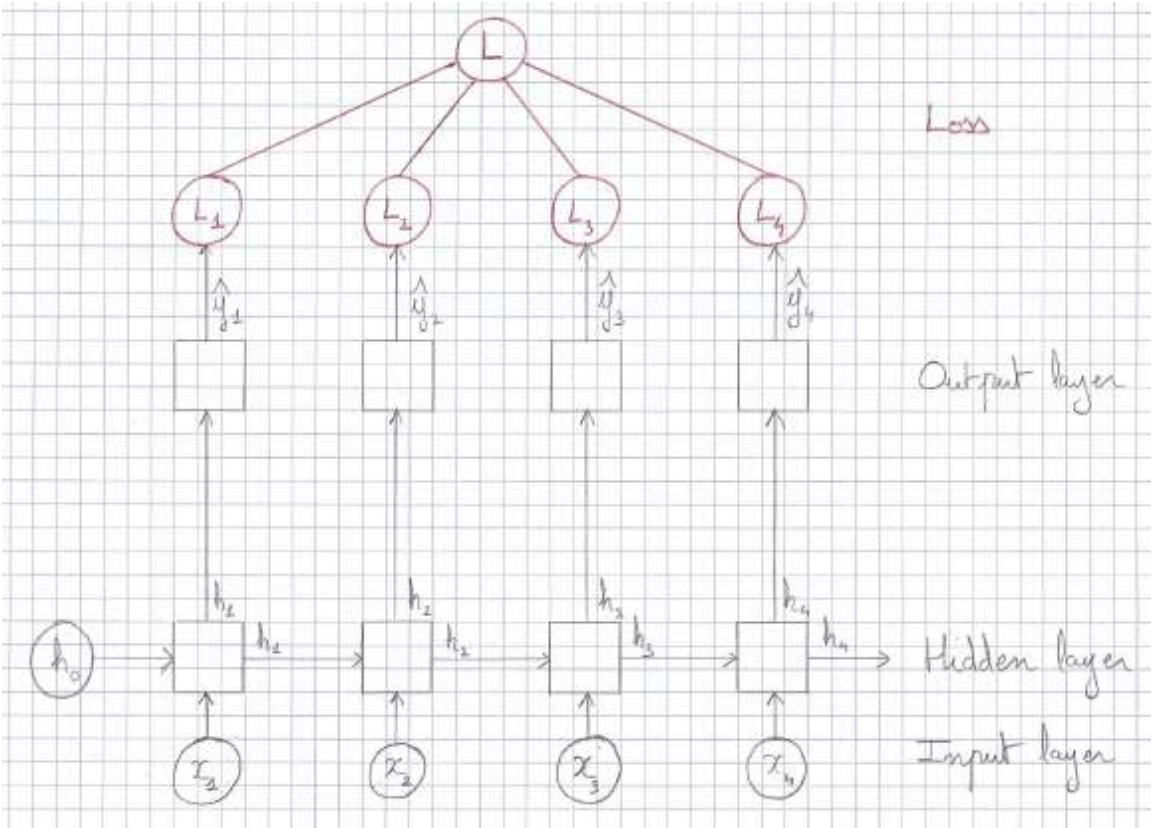
Outline

1. Introduction
2. RNN architectures
3. Vanishing/Exploding gradient: problem and solutions
   - Vanishing/Exploding gradient
   - Penalization - Weight initialization
   - GRU and LSTM
4. Miscellaneous
   - Truncated backpropagation
   - Regularization
5. Applications
   - Scene labeling: image/image
   - Image Captioning: image/sequence of words
   - Sentiment classification: sequence of words/sentiment
   - Speech synthesis/recognition
   - Video classification on frame level: sequence of image/sequence of label
   - Generating text/music

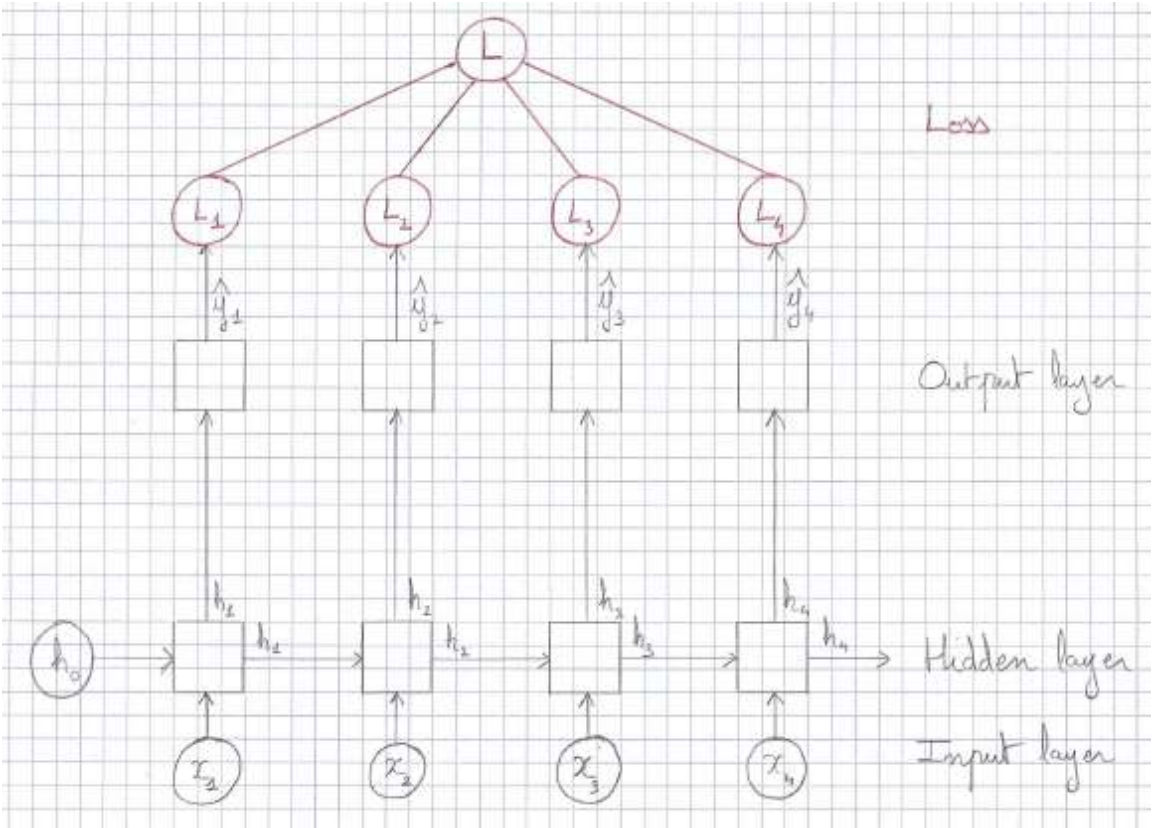
Outline

1. Introduction
2. RNN architectures
3. Vanishing/Exploding gradient: problem and solutions
   - Vanishing/Exploding gradient
   - Penalization - Weight initialization
   - GRU and LSTM
4. Miscellaneous
   - Truncated backpropagation
   - Regularization
5. Applications
   - Scene labeling: image/image
   - Image Captioning: image/sequence of words
   - Sentiment classification: sequence of words/sentiment
   - Speech synthesis/recognition
   - Video classification on frame level: sequence of image/sequence of label
   - Generating text/music

Loss



Loss



The backpropagation equation is given by

$$\frac{\partial L_T}{\partial W_{HH}} = \frac{\partial L_T}{\partial y_T} \sum_{k=1}^{T} \frac{\partial y_T}{\partial \mathbf{h}_T} \left(\prod_{m=k+1}^{T} \frac{\partial \mathbf{h}_m}{\partial \mathbf{h}_{m-1}} \right) \frac{\partial \mathbf{h}_k}{\partial W_{HH}}$$

Gradient

Backpropagation equation

$$\frac{\partial L_T}{\partial W_{HH}} = \frac{\partial L_T}{\partial y_T} \sum_{k=1}^{T} \frac{\partial y_T}{\partial \mathbf{h}_T} \left(\prod_{m=k+1}^{T} \frac{\partial \mathbf{h}_m}{\partial \mathbf{h}_{m-1}} \right) \frac{\partial \mathbf{h}_k}{\partial W_{HH}}$$

where

$$\frac{\partial \mathbf{h}_m}{\partial \mathbf{h}_{m-1}} = W_{HH}^T \mathrm{diag}(\tanh'(W_{HH}\mathbf{h}_{m-1} + W_x \mathbf{x}_m)).$$

Any problems?

Exercise - "Vanishing gradient"

Recall that the 2-norm of a matrix A is given by

$$\begin{aligned}\|A\|_2 &= \sup_{\mathbf{x}\neq 0} \frac{\|A\mathbf{x}\|_2}{\|\mathbf{x}\|_2} \\ &= \sup\{\|A\mathbf{x}\|_2, \|\mathbf{x}\|_2 = 1\} \\ &= \sqrt{\lambda_{max}(A^T A)}.\end{aligned}$$

Assuming that $\tanh'(u) \leq \gamma$, and that the largest eigenvalue of W_h^T is bounded above strictly by $1/\gamma$.

$$\left\| \frac{\partial \mathbf{h}_m}{\partial \mathbf{h}_{m-1}} \right\| \leq \left\| W_{HH}^T \right\| \left\| \text{diag}(\tanh'(W_h \mathbf{h}_{m-1} + W_x \mathbf{x}_m)) \right\| < 1.$$

Thus, there exists $0 < \eta < 1$ such that

$$\left\| \prod_{m=k+1}^{T} \frac{\partial \mathbf{h}_m}{\partial \mathbf{h}_{m-1}} \right\| \leq \eta^{T-k}.$$

As $T - k$ gets larger, the contribution of the kth term to the gradient decreases exponentially fast.

Outline

1. Introduction
2. RNN architectures
3. Vanishing/Exploding gradient: problem and solutions
   - Vanishing/Exploding gradient
   - Penalization - Weight initialization
   - GRU and LSTM
4. Miscellaneous
   - Truncated backpropagation
   - Regularization
5. Applications
   - Scene labeling: image/image
   - Image Captioning: image/sequence of words
   - Sentiment classification: sequence of words/sentiment
   - Speech synthesis/recognition
   - Video classification on frame level: sequence of image/sequence of label
   - Generating text/music

Penalization

- Exploding gradient: clipping the gradient

  If the gradient is too large, threshold the gradient. Threshold can be chosen by looking at statistics of the gradient over several updates.

  ["Empirical evaluation and combination of advanced language modeling techniques", Tomáš Mikolov et al. 2011]

- Vanishing gradient: add a contraint
  Enforce parameter updates associated with small gradient variation. Penalization:

$$\Omega = \sum_k \left(\frac{\left\| \frac{\partial L}{\partial \mathbf{h}_{k+1}} \frac{\partial \mathbf{h}_{k+1}}{\partial \mathbf{h}_k} \right\|}{\left\| \frac{\partial L}{\partial \mathbf{h}_{k+1}} \right\|} - 1 \right)^2 .$$

["On the difficulty of training recurrent neural networks", Pascanu et al. 2013]

Clever weight initialization

- The weight matrix W_h is initialized as the identity, biases are set to zero, with the ReLU activation function.

  ["A simple way to initialize recurrent networks of rectified linear units", Le et al. 2015]

- Learn a weight matrix that is a mixture of the Identity matrix and another matrix (mix of long-term and small-term dependencies).

  ["Learning longer memory in recurrent neural networks", Tomas Mikolov, Joulin, et al. 2014]

- Initialize W randomly among definite positive matrix (real and positive eigenvalues) with one eigenvalue of 1 and the other less (or equal) than 1.

  ["Improving performance of recurrent neural network with relu nonlinearity", Talathi and Vartak 2015]

Outline

1. Introduction
2. RNN architectures
3. Vanishing/Exploding gradient: problem and solutions
   - Vanishing/Exploding gradient
   - Penalization - Weight initialization
   - GRU and LSTM
4. Miscellaneous
   - Truncated backpropagation
   - Regularization
5. Applications
   - Scene labeling: image/image
   - Image Captioning: image/sequence of words
   - Sentiment classification: sequence of words/sentiment
   - Speech synthesis/recognition
   - Video classification on frame level: sequence of image/sequence of label
   - Generating text/music

Improving hidden units in RNN

Output gate (for reading)

$$\mathbf{o}_t = \sigma(W_{o,h}\mathbf{h}_{t-1} + W_{o,x}\mathbf{x}_t + \mathbf{b}_o)$$

Input gate (for writing)

$$\mathbf{i}_t = \sigma(W_{i,h}\mathbf{h}_{t-1} + W_{i,x}\mathbf{x}_t + \mathbf{b}_i)$$

Forget gate (for remembering)

$$\mathbf{f}_t = \sigma(W_{f,h}\mathbf{h}_{t-1} + W_{f,x}\mathbf{x}_t + \mathbf{b}_f)$$

Candidate hidden state.

$$\tilde{\mathbf{h}}_t = \tanh(W_h(\mathbf{o}_t \odot \mathbf{h}_{t-1}) + W_x\mathbf{x}_t + \mathbf{b})$$

The final state $\mathbf{h}_t$ is given by

$$\mathbf{h}_t = \mathbf{f}_t \odot \mathbf{h}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{h}}_t.$$

Warning: the forget gate is used for forgetting, but it actually operates as a remember gate: 1 in a forget gate means remembering everything not forgetting everything.

Improving hidden units in RNN: failure

The previous hidden units described by

$$\mathbf{h}_t = \mathbf{f}_t \odot \mathbf{h}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{h}}_t$$

fail.

Two problems:

- The forget gate and the input gate are not synchronized at the beginning of the training, which can cause the hidden states to become large and unstable.
- Since the hidden state is not bounded, the gates can be saturated, which implies difficulties to train the network.

Empirical evidence:

["LSTM: A search space odyssey", Greff et al. 2017]

Gated Recurrent Unit

One way to circumvent this issue is to specify explicitly the dependence structure between the forget gate and the writing gate.

For example, we can set the forget gate to 1 minus the writing gate:

$$\mathbf{h}_t = (1 - \mathbf{i}_t) \odot \mathbf{h}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{h}}_t.$$

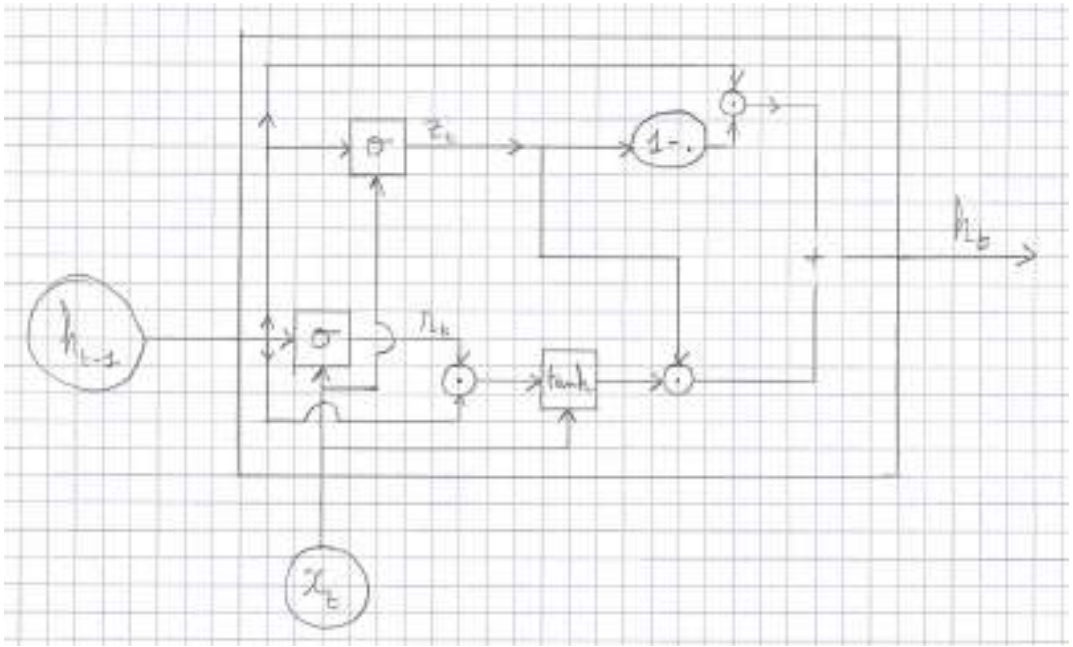
In that case, the new hidden state $\mathbf{h}_t$ is a weighted average of the previous hidden state $\mathbf{h}_{t-1}$ and the newly created candidate $\tilde{\mathbf{h}}_t$.

Consequently, $\mathbf{h}_t$ is bounded if $\mathbf{h}_{t-1}$ and $\tilde{\mathbf{h}}_t$ are, which is the case using bounded activation functions.

This is exactly the Gated Recurrent Unit.

Gated Recurrent Unit

["Empirical evaluation of gated recurrent neural networks on sequence modeling", Chung et al. 2014]



Reset gate (read gate)

$$\mathbf{r}_t = \sigma(W_{r,h}\mathbf{h}_{t-1} + W_{r,x}\mathbf{x}_t + \mathbf{b}_r)$$

Update gate (forget gate)

$$\mathbf{z}_t = \sigma(W_{z,h}\mathbf{h}_{t-1} + W_{z,x}\mathbf{x}_t + \mathbf{b}_z)$$

Candidate hidden state

$$\tilde{\mathbf{h}}_t = \tanh(W_h(\mathbf{r}_t \odot \mathbf{h}_{t-1}) + W_x\mathbf{x}_t + \mathbf{b})$$

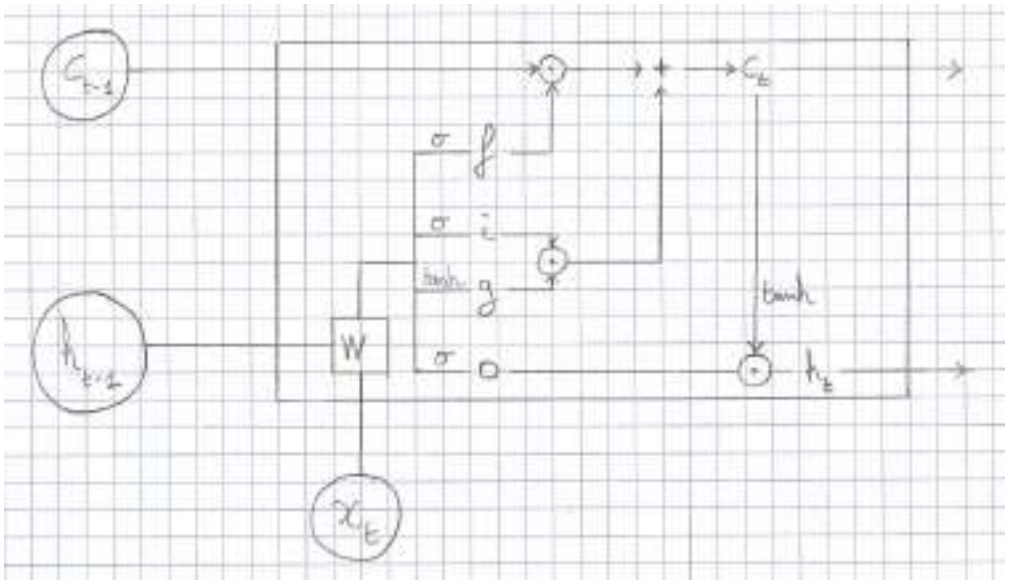
Hidden state

$$\mathbf{h}_t = \mathbf{z}_t \odot \mathbf{h}_{t-1} + (1 - \mathbf{z}_t) \odot \tilde{\mathbf{h}}_t$$

Long Short Term Memory (LSTM)

LSTM is another way to circumvent the issue of unboundedness of internal states.

["Long short-term memory", Hochreiter and Jürgen Schmidhuber 1997]



LSTM equations:

$$\mathbf{i}_t = \sigma(W_{i,h}\mathbf{h}_{t-1} + W_{i,x}\mathbf{x}_t + b_i)$$
$$\mathbf{o}_t = \sigma(W_{o,h}\mathbf{h}_{t-1} + W_{o,x}\mathbf{x}_t + b_o)$$
$$\mathbf{f}_t = \sigma(W_{f,h}\mathbf{h}_{t-1} + W_{f,x}\mathbf{x}_t + b_f)$$
$$\mathbf{g}_t = \tanh(W_{g,h}\mathbf{h}_{t-1} + W_{g,x}\mathbf{x}_t + b_g)$$

Cell state

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \mathbf{g}_t$$

Hidden state

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t)$$

The prediction of the network at time t only depends on $\mathbf{h}_t$ and not on $\mathbf{c}_t$.

Comparison of LSTM and GRU

- The traditional recurrent unit always replaces the activation, or the content of a unit with a new value computed from the current input and the previous hidden state. On the other hand, both LSTM unit and GRU keep the existing content and add the new content on top of it.
  - it is easy for each unit to remember the existence of a specific feature in the input stream for a long series of steps. Any important feature, decided by either the forget gate of the LSTM unit or the update gate of the GRU, will not be overwritten but be maintained as it is.
  - Creates shortcut paths that bypass multiple temporal steps. These shortcuts allow the error to be back-propagated more easily with less vanishing effect.

Empirical results.

- GRU perform comparably to LSTM and are better than standard RNN (particularly on music and speech modelling)
- No clear consensus between GRU and LSTM.

For LSTM:

- the squashing function $\tanh(\mathbf{c}_t)$ is important
- forget gate is important

["LSTM: A search space odyssey", Greff et al. 2017]

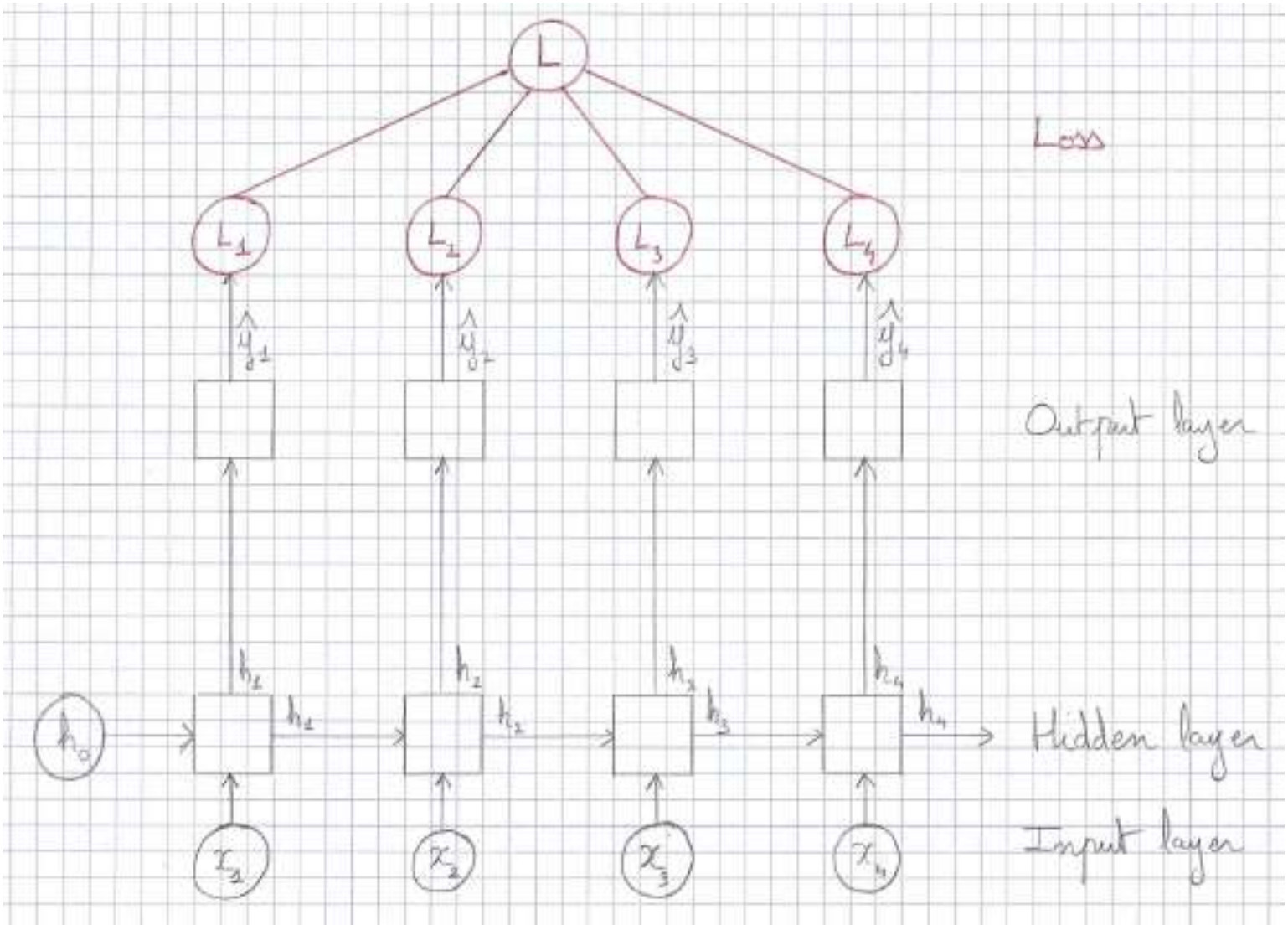
Outline

1. Introduction
2. RNN architectures
3. Vanishing/Exploding gradient: problem and solutions
   - Vanishing/Exploding gradient
   - Penalization - Weight initialization
   - GRU and LSTM
4. Miscellaneous
   - Truncated backpropagation
   - Regularization
5. Applications
   - Scene labeling: image/image
   - Image Captioning: image/sequence of words
   - Sentiment classification: sequence of words/sentiment
   - Speech synthesis/recognition
   - Video classification on frame level: sequence of image/sequence of label
   - Generating text/music

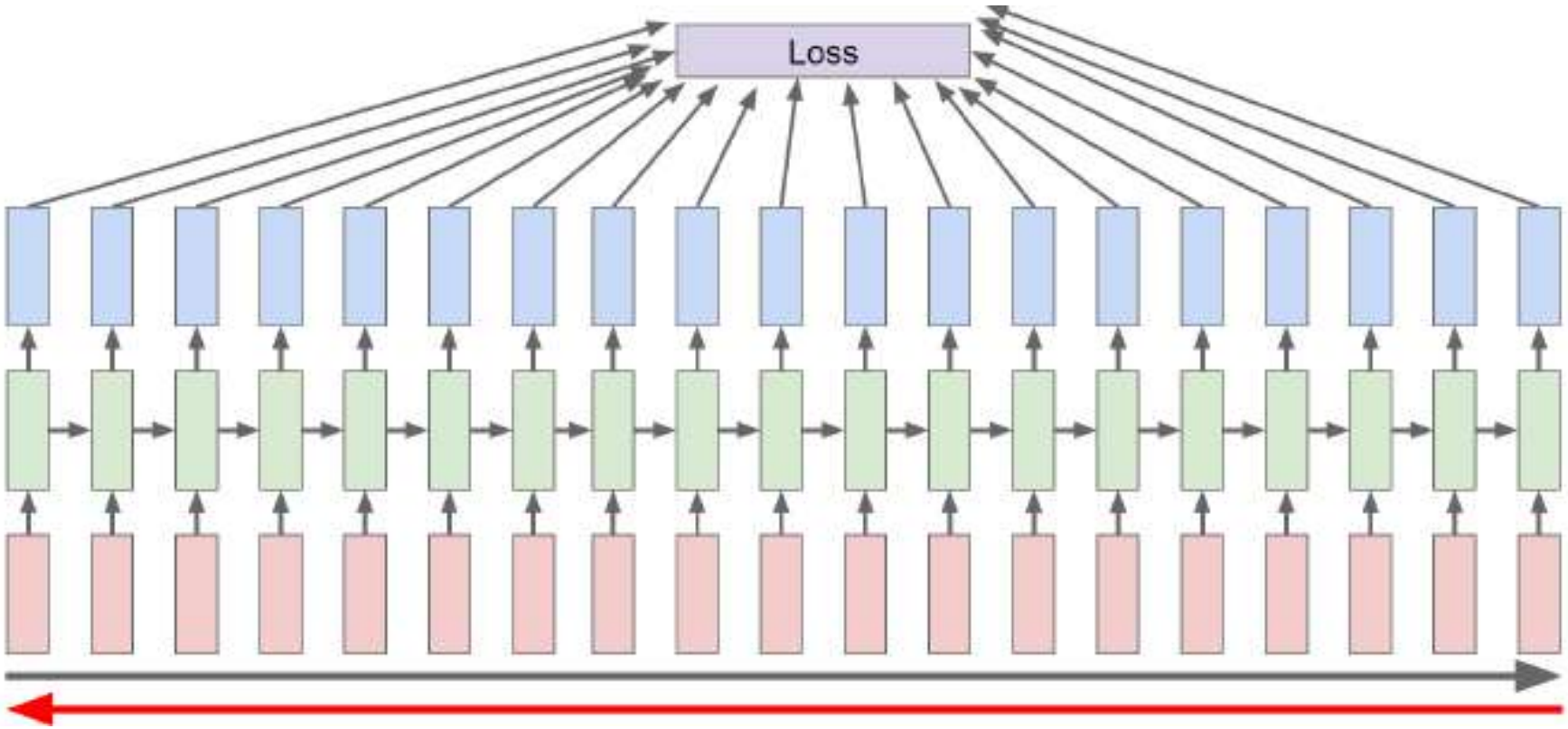
Outline

1. Introduction
2. RNN architectures
3. Vanishing/Exploding gradient: problem and solutions
   - Vanishing/Exploding gradient
   - Penalization - Weight initialization
   - GRU and LSTM
4. Miscellaneous
   - Truncated backpropagation
   - Regularization
5. Applications
   - Scene labeling: image/image
   - Image Captioning: image/sequence of words
   - Sentiment classification: sequence of words/sentiment
   - Speech synthesis/recognition
   - Video classification on frame level: sequence of image/sequence of label
   - Generating text/music

Loss

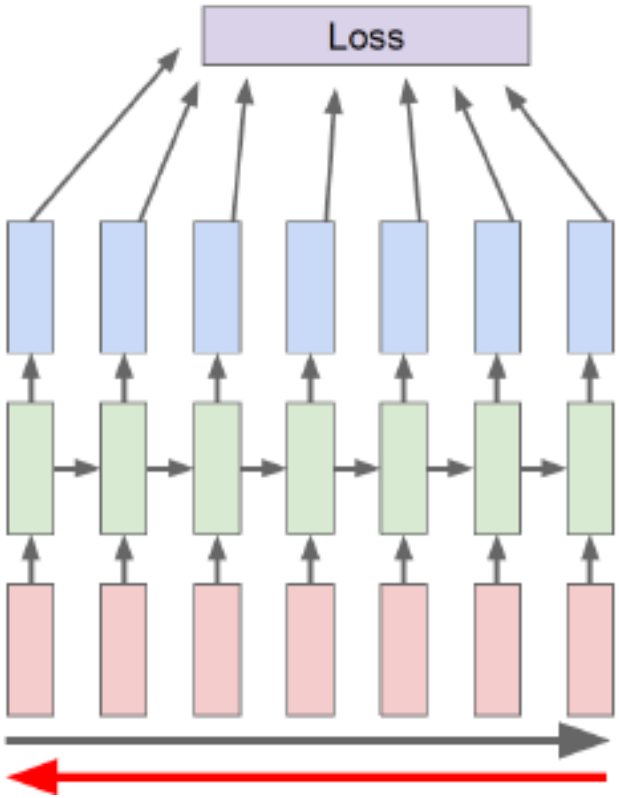


Backpropagation



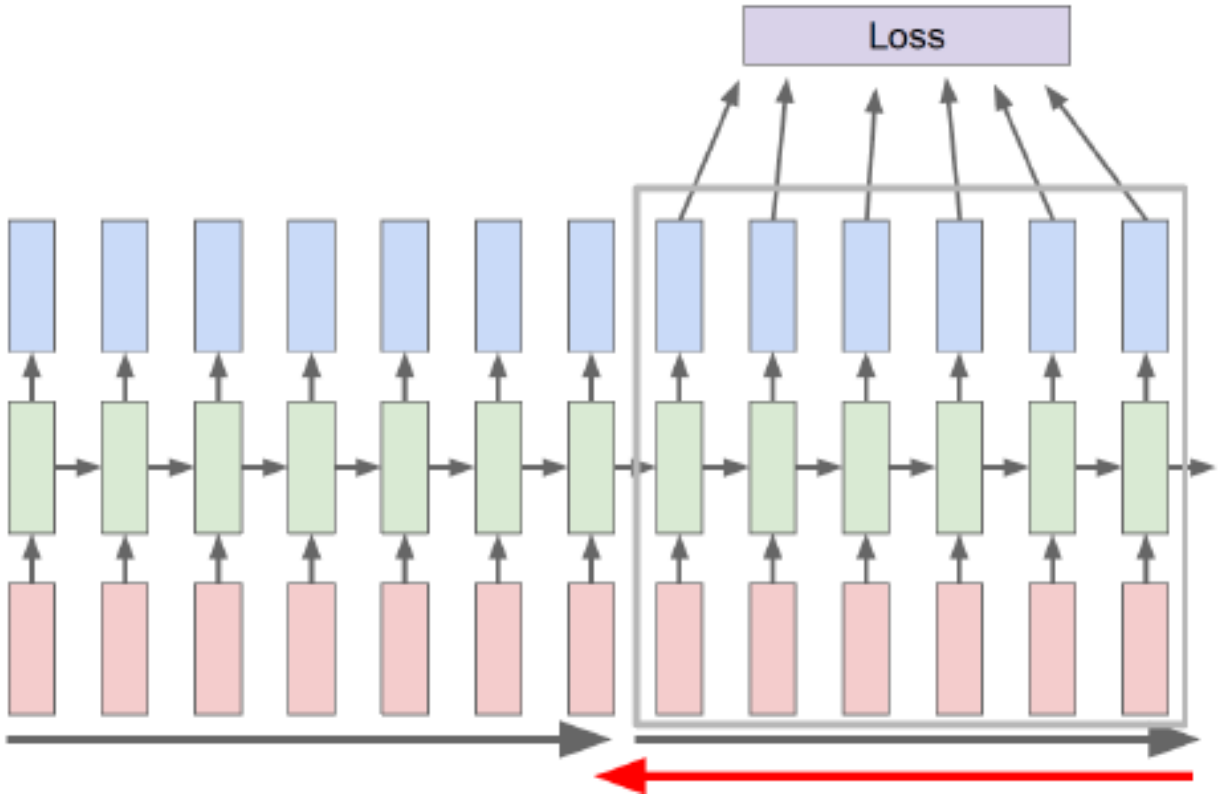
Problem: one gradient step is too costly. It requires a pass through the entire data set.

Truncated backpropagation



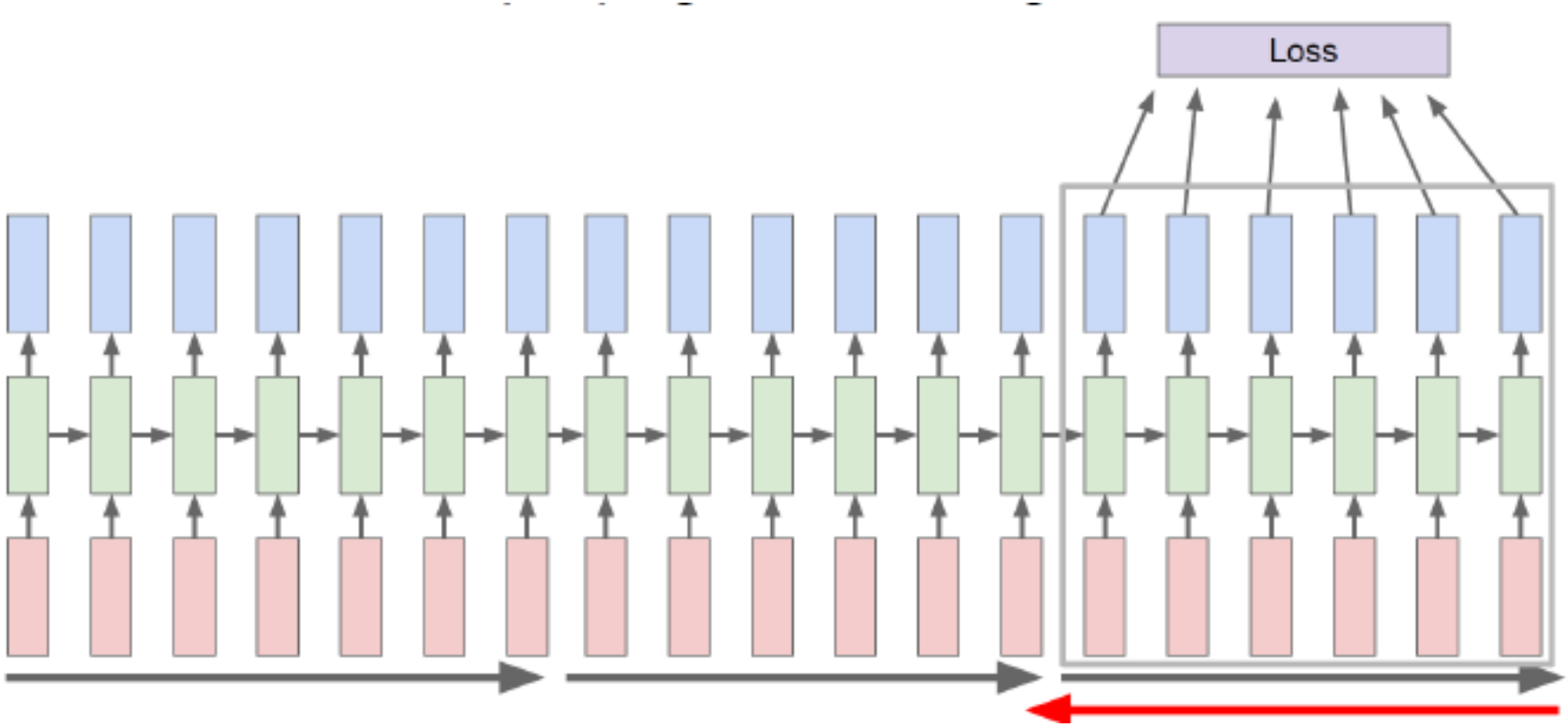
Choose a small number of steps (usually 100) and back-propagate only onto these data.

Truncated backpropagation



Propagate the weights and use backpropagation on the second batch of data.

Truncated backpropagation



Pursue...

Outline

1. Introduction
2. RNN architectures
3. Vanishing/Exploding gradient: problem and solutions
   - Vanishing/Exploding gradient
   - Penalization - Weight initialization
   - GRU and LSTM
4. Miscellaneous
   - Truncated backpropagation
   - Regularization
5. Applications
   - Scene labeling: image/image
   - Image Captioning: image/sequence of words
   - Sentiment classification: sequence of words/sentiment
   - Speech synthesis/recognition
   - Video classification on frame level: sequence of image/sequence of label
   - Generating text/music

Regularization

- L_1 or L_2 penalization
$$\mathcal{L}_{regularized}(\mathcal{D}_n, \theta) = \mathcal{L}(\mathcal{D}_n, \theta) + \lambda \|\theta\|_p^p,$$
for $p = 1, 2$.
- Activation Stabilization
$$\mathcal{L}_{stabilized}(\mathcal{D}_n, \theta) = \mathcal{L}(\mathcal{D}_n, \theta) + \lambda \frac{1}{T} \sum_{t=1}^{T} (\|\mathbf{h}_t\|_2 - \|\mathbf{h}_{t-1}\|_2)^2.$$
Experiments on language modelling and phoneme recognition show state-of-the-art performances for this approach.
["Regularizing rnns by stabilizing activations", Krueger and Memisevic 2015]
- Dropout (hidden state, forward connections...)
["Rnndrop: A novel dropout for rnns in asr", Moon et al. 2015]
["A theoretically grounded application of dropout in recurrent neural networks", Gal and Ghahramani 2016]
["Recurrent dropout without memory loss", Semeniuta et al. 2016]
- Hidden activation preservation Forcing some hidden units to keep their activation from the previous timestep ($\mathbf{h}_t = \mathbf{h}_{t-1}$):
$$\mathbf{h}_t = \mathbf{k} \odot \mathbf{h}_t + (1 - \mathbf{k}) \odot \mathbf{h}_{t-1},$$
where $\mathbf{k}$ is a Bernoulli mask. ["Zoneout: Regularizing rnns by randomly preserving hidden activations", Krueger, Maharaj, et al. 2016]

Outline

1. Introduction
2. RNN architectures
3. Vanishing/Exploding gradient: problem and solutions
   - Vanishing/Exploding gradient
   - Penalization - Weight initialization
   - GRU and LSTM
4. Miscellaneous
   - Truncated backpropagation
   - Regularization
5. Applications
   - Scene labeling: image/image
   - Image Captioning: image/sequence of words
   - Sentiment classification: sequence of words/sentiment
   - Speech synthesis/recognition
   - Video classification on frame level: sequence of image/sequence of label
   - Generating text/music

Outline

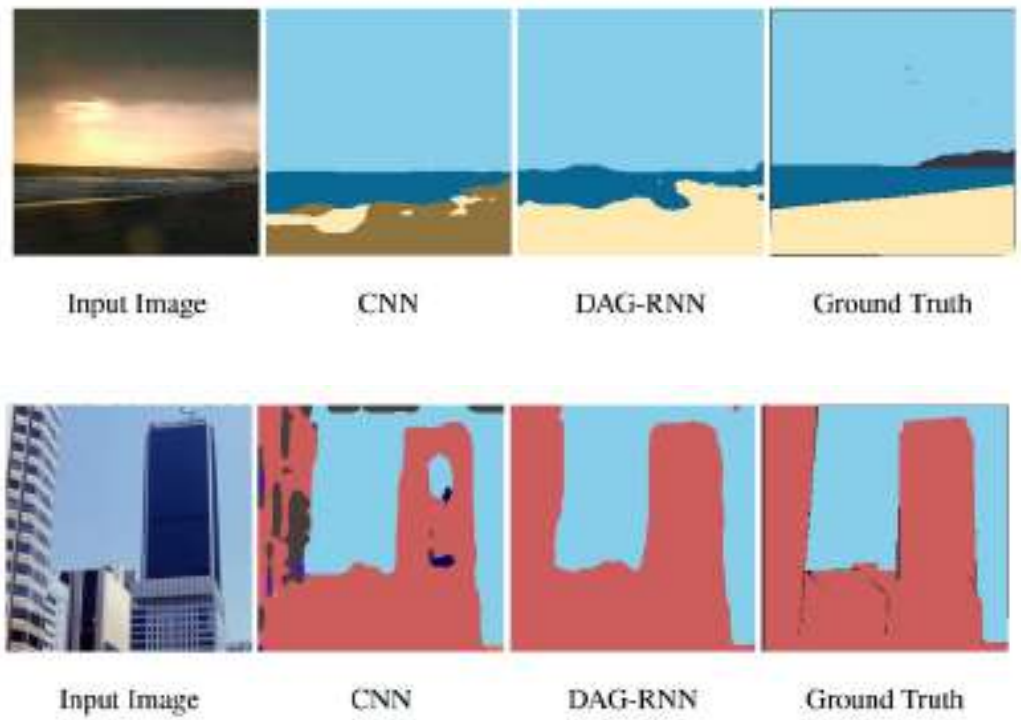
1. Introduction
2. RNN architectures
3. Vanishing/Exploding gradient: problem and solutions
   - Vanishing/Exploding gradient
   - Penalization - Weight initialization
   - GRU and LSTM
4. Miscellaneous
   - Truncated backpropagation
   - Regularization
5. Applications
   - Scene labeling: image/image
   - Image Captioning: image/sequence of words
   - Sentiment classification: sequence of words/sentiment
   - Speech synthesis/recognition
   - Video classification on frame level: sequence of image/sequence of label
   - Generating text/music

Scene labeling: DAG-RNN

["Dag-recurrent neural networks for scene labeling", Shuai et al. 2016]

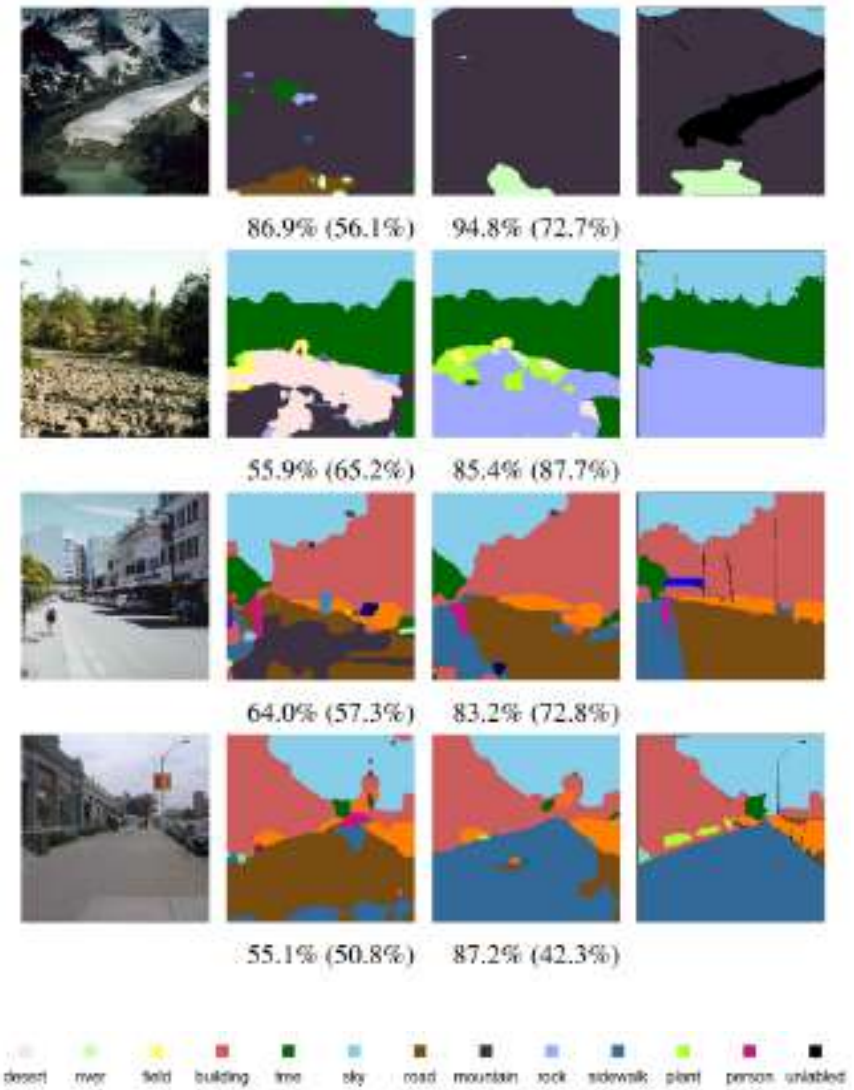
DAG-RNN is able to significantly boost the discriminative power of local representations by modeling their contextual dependencies. As a result, it can produce smoother and more semantically meaningful labeling map.

Scene labeling: DAG-RNN



With the local representations extracted from CNN, the 'sand' pixels (in the first image) are likely to be misclassified as 'road', and the 'building' pixels (in the second image) are easy to get confused with 'streetlight'.

Scene labeling: DAG-RNN



From left to right:

1. input images,
2. local prediction maps (CNN),
3. contextual labeling maps (DAG-RNN)
4. and their ground truth.

The numbers outside and inside the parentheses are global and class accuracy respectively.

Outline

1. Introduction
2. RNN architectures
3. Vanishing/Exploding gradient: problem and solutions
   - Vanishing/Exploding gradient
   - Penalization - Weight initialization
   - GRU and LSTM
4. Miscellaneous
   - Truncated backpropagation
   - Regularization
5. Applications
   - Scene labeling: image/image
   - Image Captioning: image/sequence of words
   - Sentiment classification: sequence of words/sentiment
   - Speech synthesis/recognition
   - Video classification on frame level: sequence of image/sequence of label
   - Generating text/music

Image Captioning: Neural Image Caption

["Show and tell: A neural image caption generator", Vinyals et al. 2015]

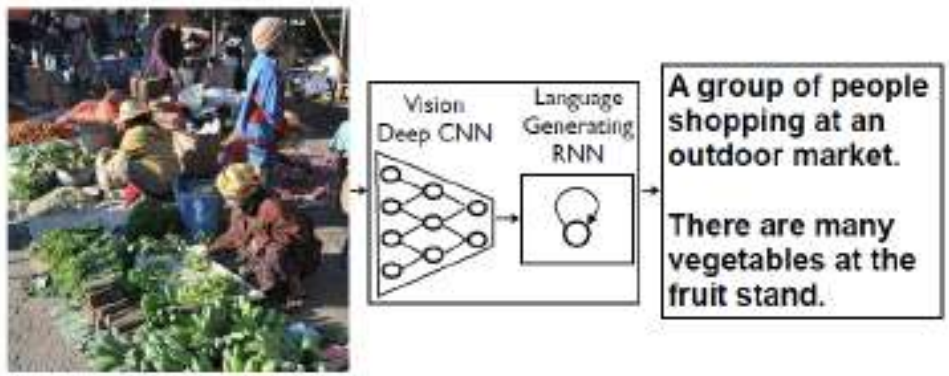
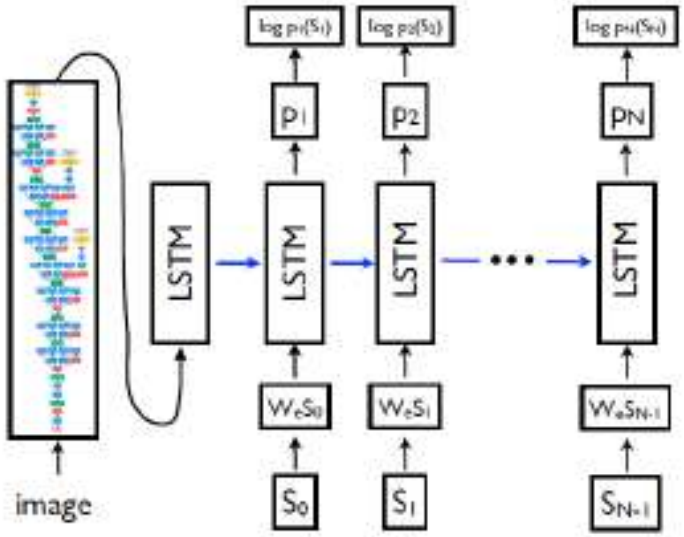


Image Captioning: Neural Image Caption



Aim:

$$\theta^\star \in \underset{\theta}{\text{argmax}} \sum_{(I,S)} \log(p(S|I))$$

where I is the input image and S the sentence describing the image. Since the sentence length can be arbitrary long, the log probability is rewritten as

$$\log(p(S|I)) = \sum_{t=0}^{N} p(S_t|I, S_0, \ldots, S_{t-1}).$$

Image Captioning: Neural Image Caption

Inference time. Two approaches:

- Sampling: sample the first word using p_1 then use this word as input to sample the second word according to p_2. Repeat the process until the network produces a stop word.
- BeamSearch: Choose the k best sentences of length t then use this set to generate the k best sentences of length $t + 1$.

How to compare two sentences?
Example:

- Candidate: the the the the the the the
- Reference 1: the cat is on the mat
- Reference 2: There is a cat on the mat

Metric:

- Precision : 7/7
- BLEU (bilingual evaluation understudy): 2/7 (maximum number of times a word is encountered in any reference sentence)

["BLEU: a method for automatic evaluation of machine translation", Papineni et al. 2002]

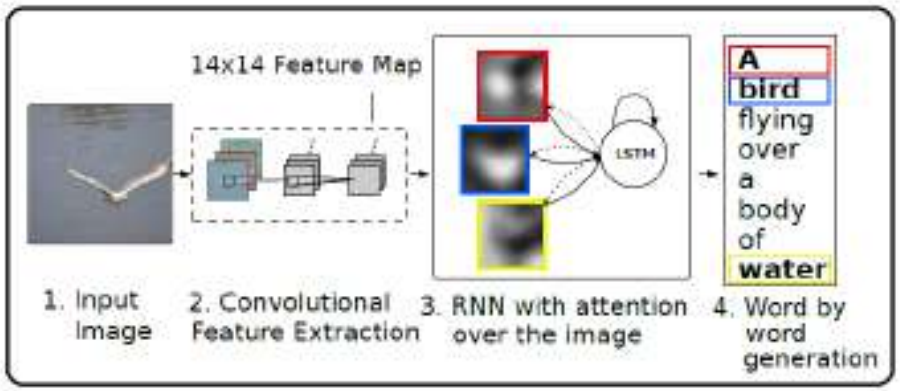
`http://nic.droppages.com/`

Image Captioning: Neural Image Caption

Figure 5. A selection of evaluation results, grouped by human rating.

Image Captioning with attention mechanism

["Show, attend and tell: Neural image caption generation with visual attention", Xu et al. 2015]



Predicted sequence of words: $\{y_1, \ldots, y_C\}$, $y_i \in \mathbb{R}^K$, where K is the size of the dictionary.

Image features: $\{a_1, \ldots, a_L\}$, where $a_i \in \mathbb{R}^D$ is a feature corresponding to a small precise area in the image (extraction from a early layer of a CNN).

$$\begin{pmatrix} \mathbf{i}_t \\ \mathbf{f}_t \\ \mathbf{o}_t \\ \mathbf{g}_t \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} T_{D+m+n,n} \begin{pmatrix} \mathbf{E}\mathbf{y}_{t-1} \\ \mathbf{h}_{t-1} \\ \hat{\mathbf{z}}_t \end{pmatrix}$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \mathbf{g}_t$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t).$$

Image Captioning with attention mechanism

$$\hat{\mathbf{z}}_t = \sum_{i=1}^{L} s_{t,i}\mathbf{a}_i,$$

where $s_{t,i} = 1$ if position i in the image should be selected at time t.

$$\mathbb{P}[s_{t,i} = 1 | s_{j<t}, \mathbf{a}] = \alpha_{t,i},$$

$$e_{ti} = f_{att}(\mathbf{a}_i, \mathbf{h}_{t-1}),$$

$$\alpha_{t,i} = \frac{\exp(e_{ti})}{\sum_j \exp(e_{tj})}.$$

Figure 2. Attention over time. As the model generates each word, its attention changes to reflect the relevant parts of the image. "soft" (top row) vs "hard" (bottom row) attention. (Note that both models generated the same captions in this example.)

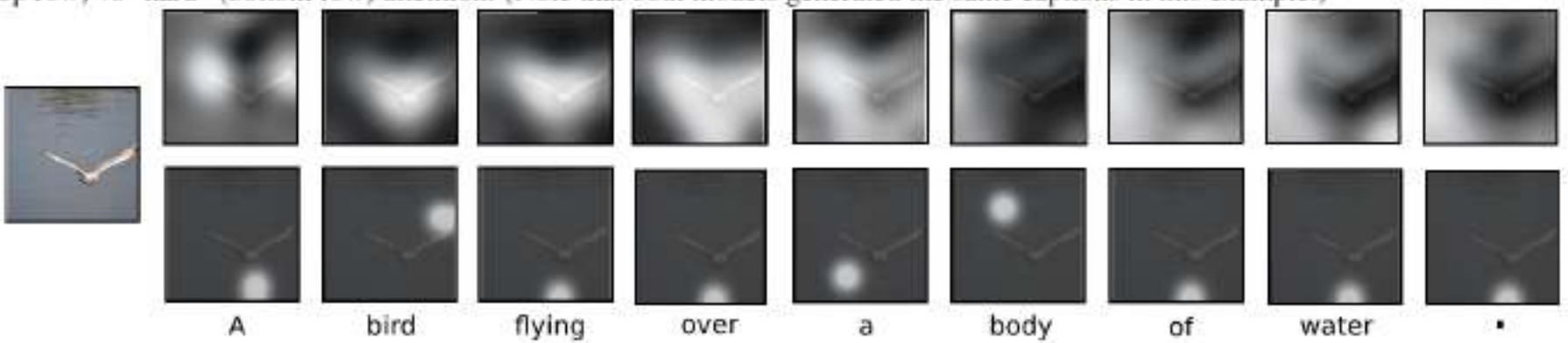


Image Captioning with attention mechanism

Figure 3. Examples of attending to the correct object (*white* indicates the attended regions, *underlines* indicated the corresponding word)

Image Captioning with attention mechanism

Figure 5. Examples of mistakes where we can use attention to gain intuition into what the model saw.

A large white bird standing in a forest.

A woman holding a clock in her hand.

A man wearing a hat and a hat on a skateboard.

A person is standing on a beach with a surfboard.

A woman is sitting at a table with a large pizza.

A man is talking on his cell phone while another man watches.

DenseCap

["Densecap: Fully convolutional localization networks for dense captioning", Johnson et al. 2016]

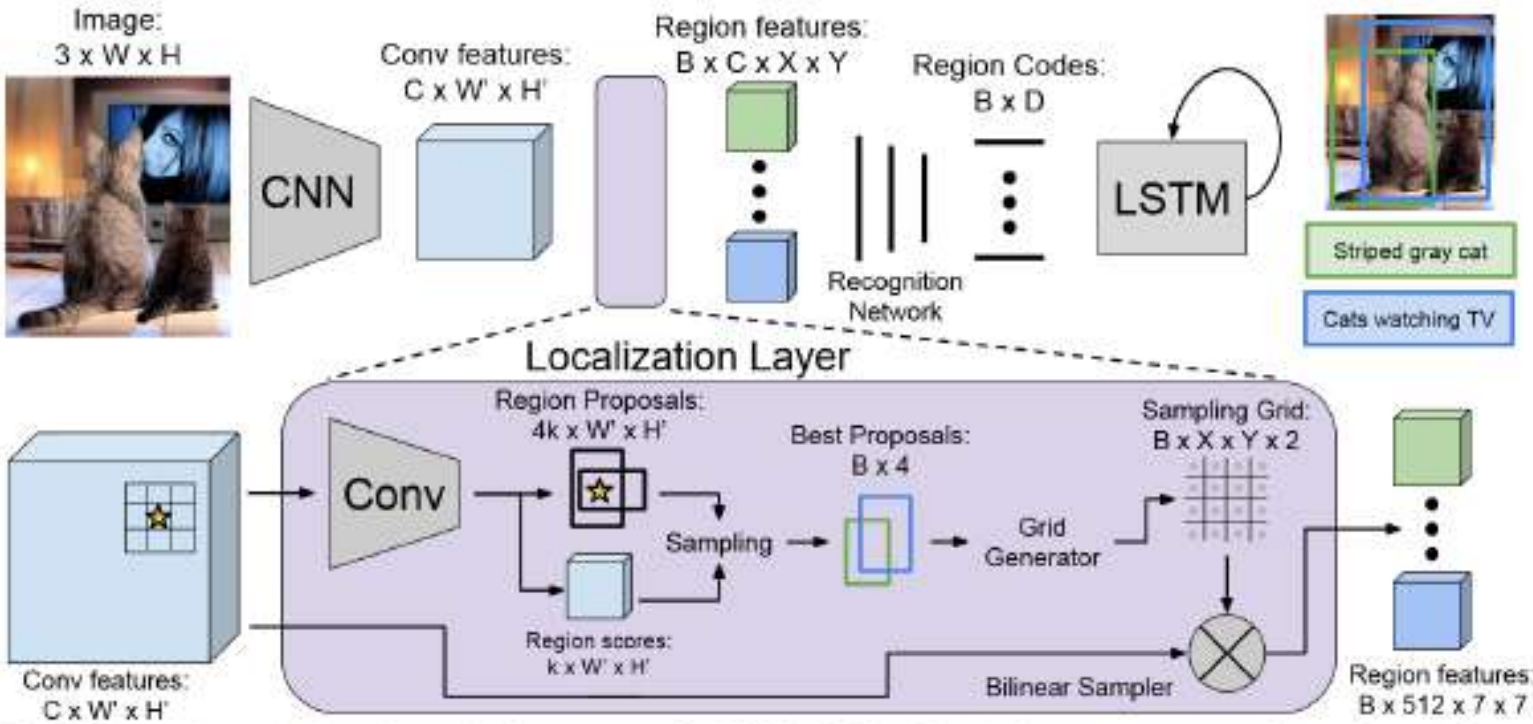
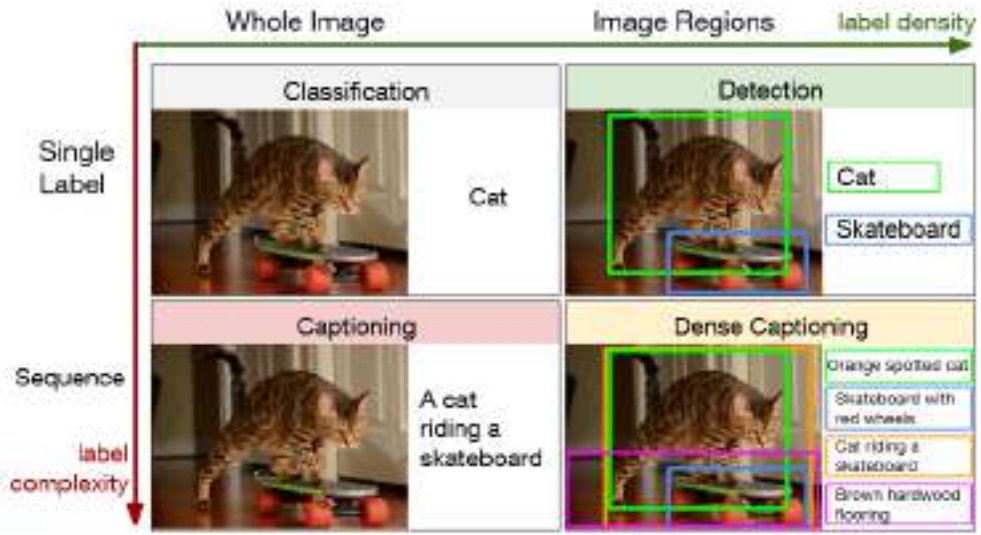


Figure 2. Model overview. An input image is first processed a CNN. The Localization Layer proposes regions and smoothly extracts a batch of corresponding activations using bilinear interpolation. These regions are processed with a fully-connected recognition network and described with an RNN language model. The model is trained end-to-end with gradient descent.

DenseCap



https://cs.stanford.edu/people/karpathy/densecap/

Try it!

https://deepai.org/machine-learning-model/densecap

Outline

1. Introduction
2. RNN architectures
3. Vanishing/Exploding gradient: problem and solutions
   - Vanishing/Exploding gradient
   - Penalization - Weight initialization
   - GRU and LSTM
4. Miscellaneous
   - Truncated backpropagation
   - Regularization
5. Applications
   - Scene labeling: image/image
   - Image Captioning: image/sequence of words
   - Sentiment classification: sequence of words/sentiment
   - Speech synthesis/recognition
   - Video classification on frame level: sequence of image/sequence of label
   - Generating text/music

Text/Sentiment classification

["A convolutional neural network for modelling sentences", Kalchbrenner et al. 2014]

They propose a shallow CNN architecture leveraging on k-max pooling which returns the top k activations in the original order in the input sequence.

["Multichannel variable-size convolution for sentence classification", Yin and Schütze 2016]

They use hierarchical convolution architecture and further exploration of multichannel and variable size feature detectors. The pooling operation can help the network deal with variable sentence lengths.

["Recurrent Convolutional Neural Networks for Text Classification.", Lai et al. 2015]

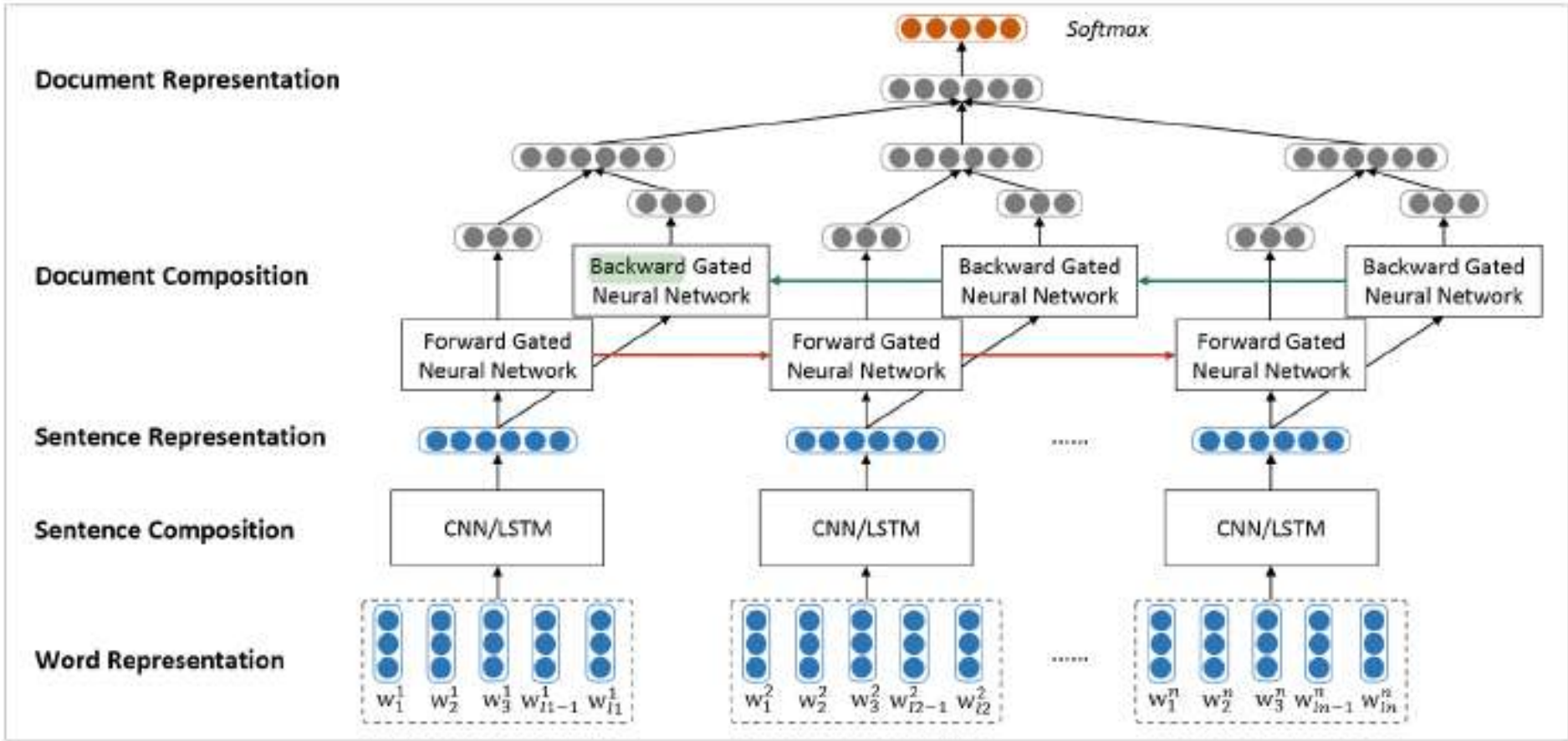
RCNNs are used for text classification on several datasets.

["Document modeling with gated recurrent neural network for sentiment classification", Tang et al. 2015]

A variety of document classification tasks is proposed in the literature using RNNs. A GRU is adapted to perform document level sentiment analysis.

RNN for document embedding

["Document modeling with gated recurrent neural network for sentiment classification", Tang et al. 2015]



RNN for document embedding

Word embedding: represent each word as an element of $\mathbb{R}^d$.
Two different manners of creating word vectors:

- Skip-gram: predict surrounding words of a given word.
- Continuous Bag Of Words (CBOW): Predict a word given surrounding words.

Different types of algorithms:

- Glove

  ["Glove: Global vectors for word representation", Pennington et al. 2014]

- Word2vec

  ["Distributed representations of words and phrases and their compositionality", Tomas Mikolov, Sutskever, et al. 2013]

- FastText

  ["Bag of tricks for efficient text classification", Joulin et al. 2016]

RNN for document embedding

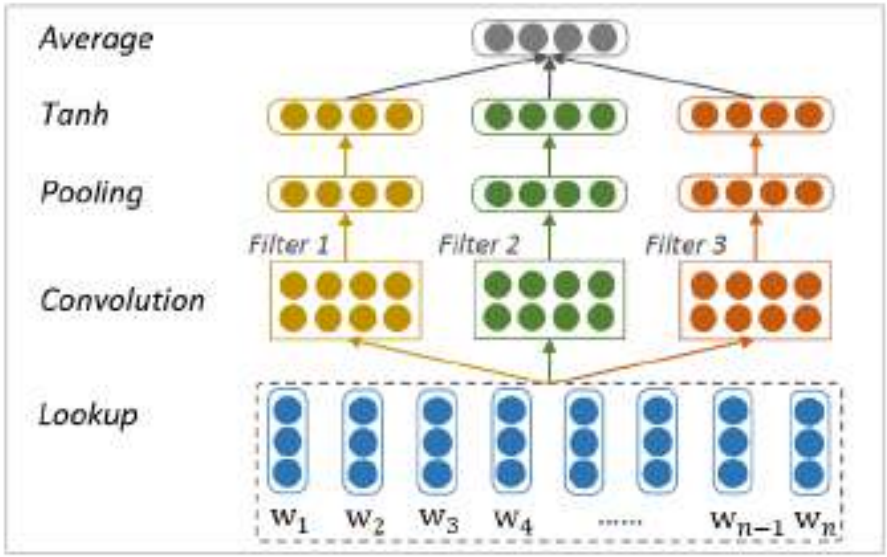
Each word w_i is mapped to its embedding representation $e_i \in \mathbb{R}^d$.

Output of convolutional layer:

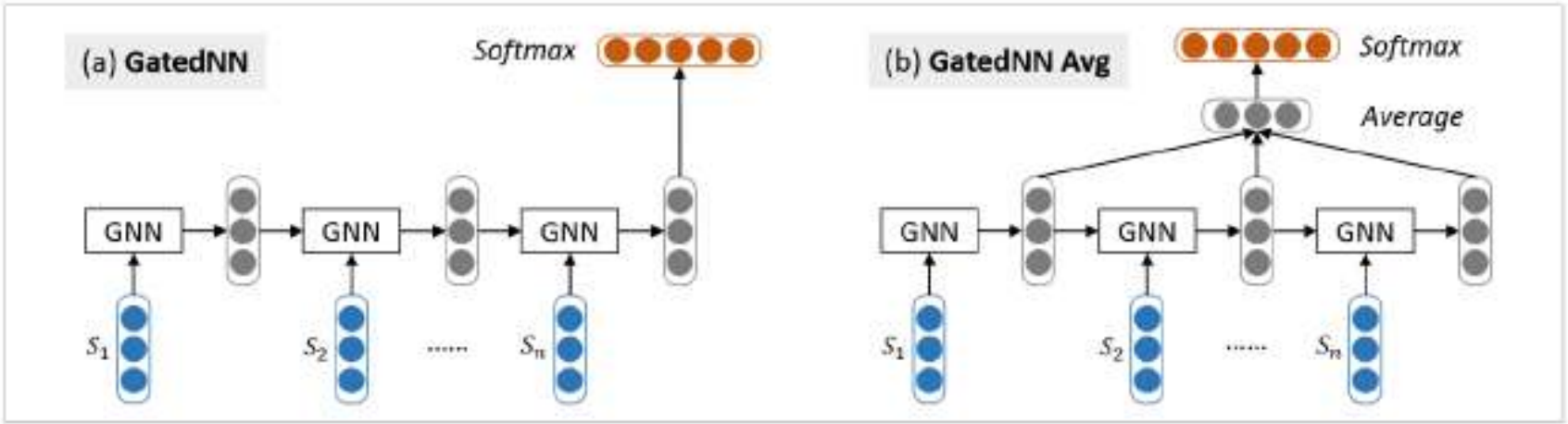
$$O_c = W_c I_c + b_c,$$

where

- $W_c \in \mathbb{R}^{l_{oc} \times d.l_c}$,
- $b_c \in \mathbb{R}^{l_o c}$,
- l_{oc} is the length of the output layer,
- l_c the size of the window,
- $I_c = [e_i, \ldots, e_{i+l_c-1}]$.



RNN for document embedding



Speech emotion recognition

["Abandoning emotion classes-towards continuous emotion recognition with modelling of long-range dependencies", Wöllmer et al. 2008]

A LSTM network is shown to have better performance than support vector machines (SVMs) and conditional random fields (CRFs), possibly due to a better modelling of long-term dependencies.

["High-level feature representation using recurrent neural network for speech emotion recognition", Lee and Tashev 2015]

They introduce a BLSTM for speech emotion recognition.

["Adieu features? end-to-end speech emotion recognition using a deep convolutional recurrent network", Trigeorgis et al. 2016]

They design a deep convolutional LSTM. This model gives state-of-the-art performance when tested on the RECOLA dataset, as the convolutional layers learns to remove background noise and outline important features in the speech, while the LSTM models the temporal structure of the speech sequence.

Outline

1. Introduction
2. RNN architectures
3. Vanishing/Exploding gradient: problem and solutions
   - Vanishing/Exploding gradient
   - Penalization - Weight initialization
   - GRU and LSTM
4. Miscellaneous
   - Truncated backpropagation
   - Regularization
5. Applications
   - Scene labeling: image/image
   - Image Captioning: image/sequence of words
   - Sentiment classification: sequence of words/sentiment
   - Speech synthesis/recognition
   - Video classification on frame level: sequence of image/sequence of label
   - Generating text/music

Speech synthesis

The goal of speech synthesis is to generate speech sounds directly from a text. It has been known for many years that the speech sounds generated by shallow structured HMM networks are often muffled compared with natural speech.

RNNs were first used for speech synthesis to leverage these sequential dependencies ["Text-to-speech conversion with neural networks: A recurrent TDNN approach", Karaali et al. 1998] ["Speech synthesis using artificial neural networks trained on cepstral coefficients", Tuerk and Robinson 1993] and were then replaced with LSTM models to better learn long term sequential dependencies ["Unidirectional long short-term memory recurrent neural network with recurrent output layer for low-latency speech synthesis", Zen and Sak 2015].

The BLSTM has been shown to perform very well in speech synthesis due to the ability to integrate the relationship with neighbouring frames in both future and past time steps ["TTS synthesis with bidirectional LSTM based recurrent neural networks", Fan et al. 2014] ["Prosody contour prediction with long short-term memory, bi-directional, deep recurrent neural networks.", Fernandez et al. 2014]

WaveNet

["WaveNet: A generative model for raw audio.", Van Den Oord et al. 2016].

WaveNet is a CNN capable of generating speech, using dilated convolutions. WaveNet has shown better performance than LSTMs and HMMs.

Through the use of dilated causal convolutions, WaveNet can model long-range temporal dependencies by increasing it's receptive field of input.

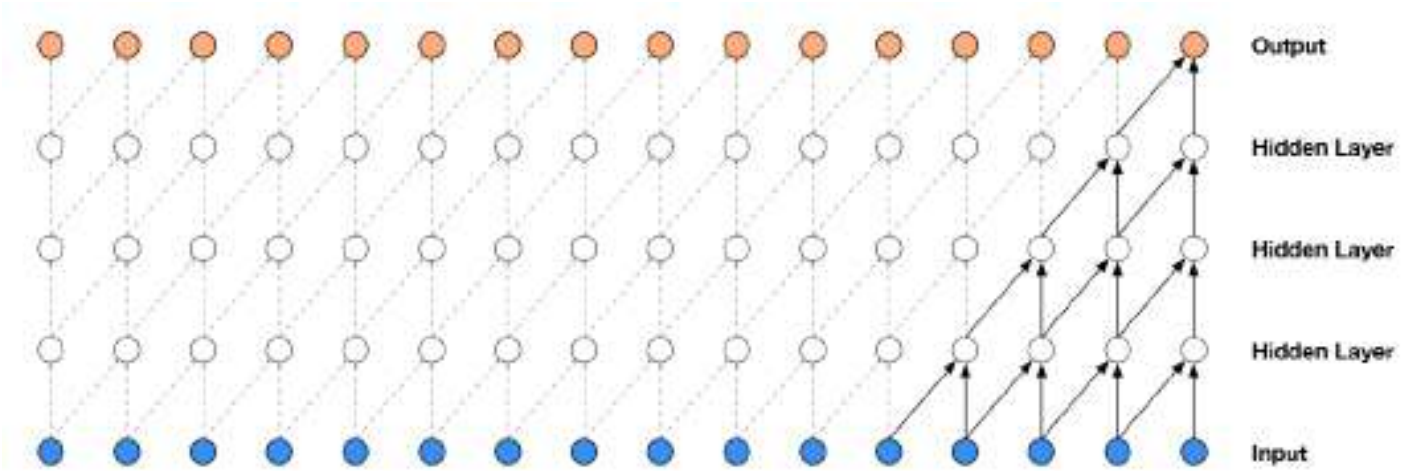


Figure: Causal convolutions

WaveNet

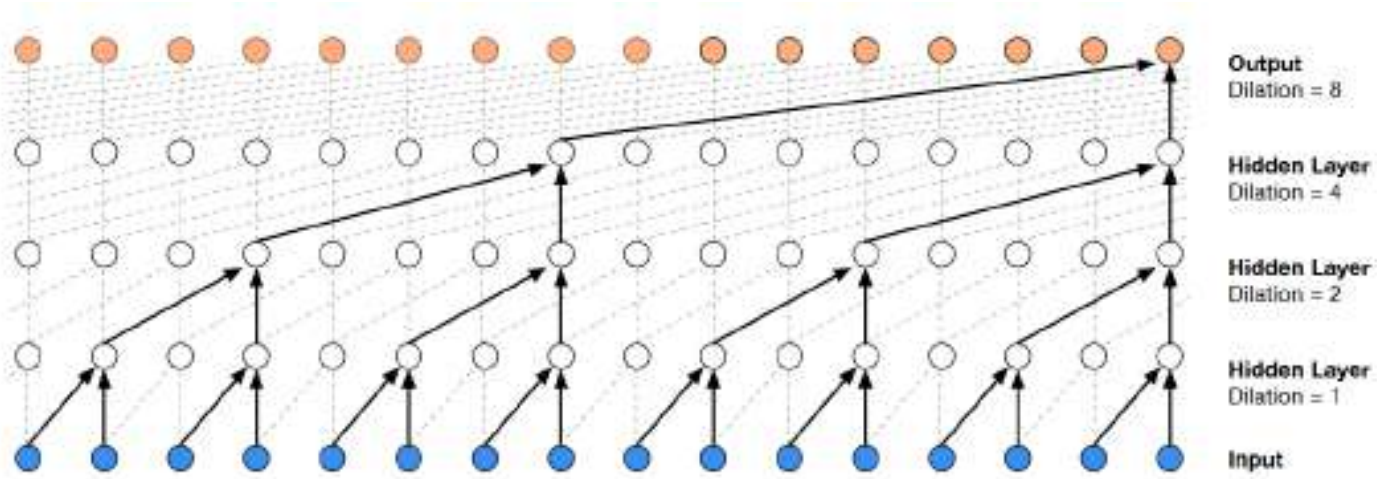


Figure: Dilated convolutions

Output signals are transformed via

$$f(x_t) = \text{sign}(x_t)\frac{\ln 1 + \mu|x_t|}{\ln(1+\mu)},$$

where $x_t \in (-1, 1)$ and $\mu = 255$, and then quantized.

`https://deepmind.com/blog/wavenet-generative-model-raw-audio/`

Speech recognition

["Survey on speech emotion recognition: Features, classification schemes, and databases", El Ayadi et al. 2011]

Automatic Speech Recognition (ASR) is the technology that converts human speech into spoken words . Before applying CNN to ASR, this domain has long been dominated by the Hidden Markov Model and Gaussian Mixture Model (GMM-HMM) methods which usually require extracting hand-craft features on speech signals

["Convolutional neural networks for speech recognition", Abdel-Hamid et al. 2014]

CNNs have shown better performance over GMM-HMMs and general DNNs , since they are well suited to exploit the correlations in both time and frequency domains through the local connectivity and are capable of capturing frequency shift in human speech signals.

["Advances in very deep convolutional neural networks for lvcsr", Sercu and Goel 2016]

Very deep CNNs have shown impressive performance in ASR .

Outline

1. Introduction
2. RNN architectures
3. Vanishing/Exploding gradient: problem and solutions
   - Vanishing/Exploding gradient
   - Penalization - Weight initialization
   - GRU and LSTM
4. Miscellaneous
   - Truncated backpropagation
   - Regularization
5. Applications
   - Scene labeling: image/image
   - Image Captioning: image/sequence of words
   - Sentiment classification: sequence of words/sentiment
   - Speech synthesis/recognition
   - Video classification on frame level: sequence of image/sequence of label
   - Generating text/music

Video

While different tasks have been performed on videos using RNNs, they are most prevalent in video description generation. This application involves components of both image processing and natural language processing.

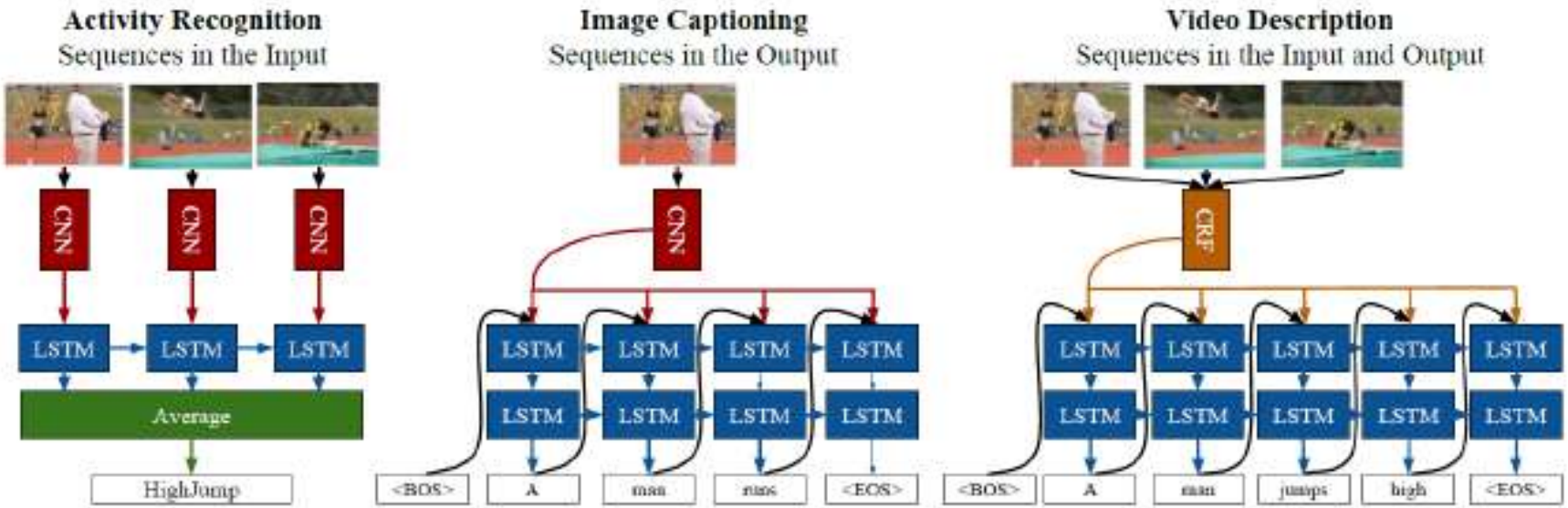
["Translating videos to natural language using deep recurrent neural networks", Venugopalan et al. 2014]
They introduce a LSTM model, which directly connects to a deep CNN. This is the first end-to-end solution for video annotations.

["Video description generation incorporating spatio-temporal features and a soft-attention mechanism", Yao et al. 2015]
They introduce a 3-dimensional convolutional architecture for feature extraction. These features were then fed to an LSTM model based on a soft-attention mechanism to dynamically control the flow of information from multiple video frames.

["Long-term recurrent convolutional networks for visual recognition and description", Donahue et al. 2015]
Another approach to model the dynamics of videos differently from spatial variations, is to feed the CNN based features of individual frames to a sequence learning module e.g., a recurrent neural network.

Video Description

["Long-term recurrent convolutional networks for visual recognition and description", Donahue et al. 2015]



Outline

1. Introduction
2. RNN architectures
3. Vanishing/Exploding gradient: problem and solutions
   - Vanishing/Exploding gradient
   - Penalization - Weight initialization
   - GRU and LSTM
4. Miscellaneous
   - Truncated backpropagation
   - Regularization
5. Applications
   - Scene labeling: image/image
   - Image Captioning: image/sequence of words
   - Sentiment classification: sequence of words/sentiment
   - Speech synthesis/recognition
   - Video classification on frame level: sequence of image/sequence of label
   - Generating text/music

Generating text/music

LSTMs have improved RNN models for language modeling due to their due to their ability to learn long-term dependencies in a sequence better than a simple hidden state

["LSTM neural networks for language modeling", Sundermeyer et al. 2012]

LSTMs are also used to generate complex text and online handwriting sequences with long-range structure, simply by predicting one data point at a time.

["Generating sequences with recurrent neural networks", Graves 2013]

RNNs are also used to capture poetic style in works of literature and generate lyrics, for example Rap lyric generation

["Chinese poetry generation with recurrent neural networks", Zhang and Lapata 2014]["GhostWriter: Using an LSTM for automatic rap lyric generation", Potash et al. 2015], ["Generating topical poetry", Ghazvininejad et al. 2016]

["A first look at music composition using lstm recurrent neural networks", Eck and Juergen Schmidhuber 2002a] ["Finding temporal structure in music: Blues improvisation with LSTM recurrent networks", Eck and Juergen Schmidhuber 2002b]

It has been shown that RNN models struggle to keep track of distant events that indicate the temporal structure of music. LSTM models have since been adapted in music generation to better learn the long-term temporal structure of certain genres of music

Speech and Audio

With the introduction of the connectionist temporal classification (CTC) function, RNNs are capable of leveraging sequence learning on unsegmented speech data

["Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks", Graves, Fernández, Gomez, et al. 2006]

Since then, the popularity of RNNs in speech recognition has exploded. Developments in speech recognition then used the CTC function alongside newer recurrent network architectures, which were more robust to vanishing gradients to improve performance and perform recognition on larger vocabularies

["Towards end-to-end speech recognition with recurrent neural networks", Graves and Jaitly 2014] ["Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition", Sak et al. 2014] ["End-to-end attention-based large vocabulary speech recognition", Bahdanau et al. 2016]

Iterations of the CTC model, such as the sequence transducer and neural transducer have incorporated a second RNN to act as a language model to tackle tasks such as online speech recognition. These augmentations allows the model to make predictions based on not only the linguistic features, but also on the previous transcriptions made.

["A neural transducer", Jaitly et al. 2015]

["Generating sequences with recurrent neural networks", Graves 2013]

Ossama Abdel-Hamid et al. “Convolutional neural networks for speech recognition”. In: *IEEE/ACM Transactions on audio, speech, and language processing* 22.10 (2014), pp. 1533–1545.

Dzmitry Bahdanau et al. “End-to-end attention-based large vocabulary speech recognition”. In: *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE. 2016, pp. 4945–4949.

Junyoung Chung et al. “Empirical evaluation of gated recurrent neural networks on sequence modeling”. In: *arXiv preprint arXiv:1412.3555* (2014).

Jeffrey Donahue et al. “Long-term recurrent convolutional networks for visual recognition and description”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 2625–2634.

Moataz El Ayadi, Mohamed S Kamel, and Fakhri Karray. “Survey on speech emotion recognition: Features, classification schemes, and databases”. In: *Pattern Recognition* 44.3 (2011), pp. 572–587.

Douglas Eck and Juergen Schmidhuber. “A first look at music composition using lstm recurrent neural networks”. In: *Istituto Dalle Molle Di Studi Sull Intelligenza Artificiale* 103 (2002).

Douglas Eck and Juergen Schmidhuber. "Finding temporal structure in music: Blues improvisation with LSTM recurrent networks". In: *Neural Networks for Signal Processing, 2002. Proceedings of the 2002 12th IEEE Workshop on*. IEEE. 2002, pp. 747–756.

Yuchen Fan et al. "TTS synthesis with bidirectional LSTM based recurrent neural networks". In: *Fifteenth Annual Conference of the International Speech Communication Association*. 2014.

Raul Fernandez et al. "Prosody contour prediction with long short-term memory, bi-directional, deep recurrent neural networks." In: *Interspeech*. 2014, pp. 2268–2272.

Alex Graves, Santiago Fernández, and Jürgen Schmidhuber. "Multi-Dimensional Recurrent Neural Networks". In: *CoRR* abs/0705.2011 (2007). arXiv: 0705.2011. URL: http://arxiv.org/abs/0705.2011.

Yarin Gal and Zoubin Ghahramani. "A theoretically grounded application of dropout in recurrent neural networks". In: *Advances in neural information processing systems*. 2016, pp. 1019–1027.

Marjan Ghazvininejad et al. "Generating topical poetry". In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. 2016, pp. 1183–1191.

Alex Graves and Navdeep Jaitly. “Towards end-to-end speech recognition with recurrent neural networks”. In: *International Conference on Machine Learning*. 2014, pp. 1764–1772.

Alex Graves, Santiago Fernández, Faustino Gomez, et al. “Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks”. In: *Proceedings of the 23rd international conference on Machine learning*. ACM. 2006, pp. 369–376.

Alex Graves. “Generating sequences with recurrent neural networks”. In: *arXiv preprint arXiv:1308.0850* (2013).

Klaus Greff et al. “LSTM: A search space odyssey”. In: *IEEE transactions on neural networks and learning systems* 28.10 (2017), pp. 2222–2232.

Sepp Hochreiter and Jürgen Schmidhuber. “Long short-term memory”. In: *Neural computation* 9.8 (1997), pp. 1735–1780.

Navdeep Jaitly et al. “A neural transducer”. In: *arXiv preprint arXiv:1511.04868* (2015).

Justin Johnson, Andrej Karpathy, and Li Fei-Fei. “Densecap: Fully convolutional localization networks for dense captioning”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 4565–4574.

Armand Joulin et al. “Bag of tricks for efficient text classification”. In: *arXiv preprint arXiv:1607.01759* (2016).

Orhan Karaali et al. “Text-to-speech conversion with neural networks: A recurrent TDNN approach”. In: *arXiv preprint cs/9811032* (1998).

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. “A convolutional neural network for modelling sentences”. In: *arXiv preprint arXiv:1404.2188* (2014).

David Krueger and Roland Memisevic. “Regularizing rnns by stabilizing activations”. In: *arXiv preprint arXiv:1511.08400* (2015).

David Krueger, Tegan Maharaj, et al. “Zoneout: Regularizing rnns by randomly preserving hidden activations”. In: *arXiv preprint arXiv:1606.01305* (2016).

Siwei Lai et al. “Recurrent Convolutional Neural Networks for Text Classification.” In: *AAAI*. Vol. 333. 2015, pp. 2267–2273.

Ming Liang and Xiaolin Hu. “Recurrent convolutional neural network for object recognition”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 3367–3375.

Quoc V Le, Navdeep Jaitly, and Geoffrey E Hinton. "A simple way to initialize recurrent networks of rectified linear units". In: *arXiv preprint arXiv:1504.00941* (2015).

Jinkyu Lee and Ivan Tashev. "High-level feature representation using recurrent neural network for speech emotion recognition". In: (2015).

Tomáš Mikolov et al. "Empirical evaluation and combination of advanced language modeling techniques". In: *Twelfth Annual Conference of the International Speech Communication Association*. 2011.

Tomas Mikolov, Ilya Sutskever, et al. "Distributed representations of words and phrases and their compositionality". In: *Advances in neural information processing systems*. 2013, pp. 3111–3119.

Tomas Mikolov, Armand Joulin, et al. "Learning longer memory in recurrent neural networks". In: *arXiv preprint arXiv:1412.7753* (2014).

Taesup Moon et al. "Rnndrop: A novel dropout for rnns in asr". In: *Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop on*. IEEE. 2015, pp. 65–70.

Kishore Papineni et al. “BLEU: a method for automatic evaluation of machine translation”. In: *Proceedings of the 40th annual meeting on association for computational linguistics*. Association for Computational Linguistics. 2002, pp. 311–318.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. “On the difficulty of training recurrent neural networks”. In: *International Conference on Machine Learning*. 2013, pp. 1310–1318.

Peter Potash, Alexey Romanov, and Anna Rumshisky. “GhostWriter: Using an LSTM for automatic rap lyric generation”. In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. 2015, pp. 1919–1924.

Jeffrey Pennington, Richard Socher, and Christopher Manning. “Glove: Global vectors for word representation”. In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014, pp. 1532–1543.

Hojjat Salehinejad et al. “Recent Advances in Recurrent Neural Networks”. In: *arXiv preprint arXiv:1801.01078* (2017).

Tom Sercu and Vaibhava Goel. “Advances in very deep convolutional neural networks for lvcsr”. In: *arXiv preprint arXiv:1604.01792* (2016).

Bing Shuai et al. “Dag-recurrent neural networks for scene labeling”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 3620–3629.

Haşim Sak, Andrew Senior, and Françoise Beaufays. “Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition”. In: *arXiv preprint arXiv:1402.1128* (2014).

Stanislau Semeniuta, Aliaksei Severyn, and Erhardt Barth. “Recurrent dropout without memory loss”. In: *arXiv preprint arXiv:1603.05118* (2016).

Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. “LSTM neural networks for language modeling”. In: *Thirteenth annual conference of the international speech communication association*. 2012.

Duyu Tang, Bing Qin, and Ting Liu. “Document modeling with gated recurrent neural network for sentiment classification”. In: *Proceedings of the 2015 conference on empirical methods in natural language processing*. 2015, pp. 1422–1432.

Christine Tuerk and Tony Robinson. “Speech synthesis using artificial neural networks trained on cepstral coefficients”. In: *Third European Conference on Speech Communication and Technology*. 1993.

George Trigeorgis et al. “Adieu features? end-to-end speech emotion recognition using a deep convolutional recurrent network”. In: *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE. 2016, pp. 5200–5204.

Sachin S Talathi and Aniket Vartak. “Improving performance of recurrent neural network with relu nonlinearity”. In: *arXiv preprint arXiv:1511.03771* (2015).

Aäron Van Den Oord et al. “WaveNet: A generative model for raw audio.” In: *SSW*. 2016, p. 125.

Subhashini Venugopalan et al. “Translating videos to natural language using deep recurrent neural networks”. In: *arXiv preprint arXiv:1412.4729* (2014).

Oriol Vinyals et al. “Show and tell: A neural image caption generator”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 3156–3164.

Vivek Veeriah, Naifan Zhuang, and Guo-Jun Qi. “Differential recurrent neural networks for action recognition”. In: *2015 IEEE International Conference on Computer Vision (ICCV)*. IEEE. 2015, pp. 4041–4049.

Martin Wöllmer et al. “Abandoning emotion classes-towards continuous emotion recognition with modelling of long-range dependencies”. In: *Proc. 9th Interspeech 2008 incorp. 12th Australasian Int. Conf. on Speech Science and Technology SST 2008, Brisbane, Australia*. 2008, pp. 597–600.

Kelvin Xu et al. “Show, attend and tell: Neural image caption generation with visual attention”. In: *International conference on machine learning*. 2015, pp. 2048–2057.

Li Yao et al. “Video description generation incorporating spatio-temporal features and a soft-attention mechanism”. In: *arXiv preprint arXiv:1502.08029* (2015).

Wenpeng Yin and Hinrich Schütze. “Multichannel variable-size convolution for sentence classification”. In: *arXiv preprint arXiv:1603.04513* (2016).

Xingxing Zhang and Mirella Lapata. “Chinese poetry generation with recurrent neural networks”. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2014, pp. 670–680.

Heiga Zen and Haşim Sak. “Unidirectional long short-term memory recurrent neural network with recurrent output layer for low-latency speech synthesis”. In: *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE. 2015, pp. 4470–4474.