Echo State Networks and Image Captioning

Deep Learning Lecture 7

Samuel Cheng

School of ECE
University of Oklahoma

Spring, 2017
(Slides credit to Stanford CS231n and Hinton et al.)

Table of Contents

1. Echo-state networks

2. Image captioning

3. Conclusions

Logistics

- HW 2 due today
  - 5% penalty per day starting tomorrow
- Soubhi is the winner of HW2
  - He will present the solution next week

Project proposal

- Submit your team info, project title, and abstract by the class of the week after spring break ($\sim$ 3/24)
  - 5% of the total course
- Will talk more about this after we cover the course materials

Class rescheduling

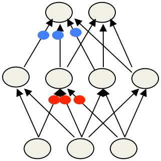
- We will need to reschedule the classes on 3/24 (OUHSC Biomedical Engineering Symposium) and 3/31 (Basketball game)
- Note that your presentations will be scheduled accordingly also

Review and Overview

- We talked about RNNs last time. Vanilla RNNs can be difficult to train because of gradient vanishing problems. There are several potential solutions (according to Hinton)
  - Gated modification (unanimous): LSTM, GRU
  - Better optimizers: Hessian-free methods (conjugate gradient)
  - (Much) better initialization: echo-state networks
- We will look into echo-state networks and also an RNN application—image captioning

The key idea of echo state networks (perceptrons again?)

- A very simple way to learn a feedforward network is to make the early layers random and fixed.
- Then we just learn the last layer which is a linear model that uses the transformed inputs to predict the target outputs.
  - A big random expansion of the input vector can help.



- The equivalent idea for RNNs is to fix the input→hidden connections and the hidden→hidden connections at random values and only learn the hidden→output connections.
  - The learning is then very simple (assuming linear output units).
  - Its important to set the random connections very carefully so the RNN does not explode or die.

Reservoir computing

- ESNs now are categorized into a group of techniques called **reservoir computing**
  - The untrained RNN part of an ESN is called a dynamical **reservoir**
  - And its states are termed **echoes** of its input history
- Liquid state machines (LSMs) were invented independently around the same time (early 2000's) and were based on a similar idea
  - LSMs use spiking neural networks to increase level of realism in a neural simulation
- See `http://www.nature.com/articles/srep22381`, `http://reservoir-computing.org/`, and a practical guide to applying echo states networks

How to set random connections in echo state networks

- Set the hidden→hidden weights so that the length of the activity vector stays about the same after each iteration
  - This allows the input to echo around the network for a long time
- Use sparse connectivity (i.e. set most of the weights to zero)
  - This creates lots of loosely coupled oscillators (suggested by original paper)
- Choose the scale of the input→hidden connections very carefully
  - They need to drive the loosely coupled oscillators without wiping out the information from the past that they already contain
- The learning is so fast that we can try many different scales for the weights and sparsenesses
  - This is often necessary

How to set random connections in echo state networks

- Set the hidden→hidden weights so that the length of the activity vector stays about the same after each iteration
  - This allows the input to echo around the network for a long time
- Use sparse connectivity (i.e. set most of the weights to zero)
  - This creates lots of loosely coupled oscillators (suggested by original paper)

- Choose the scale of the input→hidden connections very carefully
  - They need to drive the loosely coupled oscillators without wiping out the information from the past that they already contain
- The learning is so fast that we can try many different scales for the weights and sparsenesses
  - This is often necessary

How to set random connections in echo state networks

- Set the hidden→hidden weights so that the length of the activity vector stays about the same after each iteration
    - This allows the input to echo around the network for a long time
- Use sparse connectivity (i.e. set most of the weights to zero)
    - This creates lots of loosely coupled oscillators (suggested by original paper)
- Choose the scale of the input→hidden connections very carefully
    - They need to drive the loosely coupled oscillators without wiping out the information from the past that they already contain
- The learning is so fast that we can try many different scales for the weights and sparsenesses
    - This is often necessary

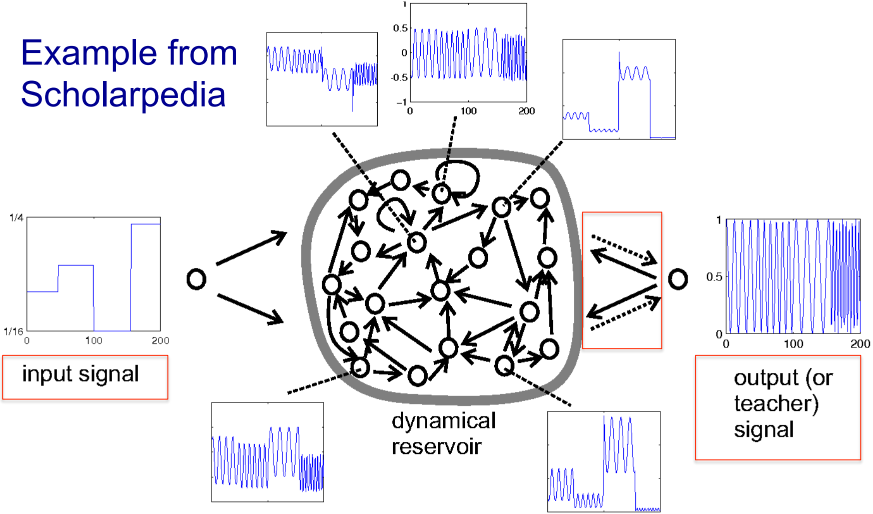
A simple example of an echo state network

INPUT SEQUENCE A real-valued time-varying value that specifies the frequency of a sine wave

TARGET OUTPUT SEQUENCE A sine wave with the currently specified frequency

LEARNING METHOD Fit a linear model that takes the states of the hidden units as input and produces a single scalar output

Example from Scholarpedia

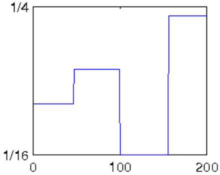


input signal

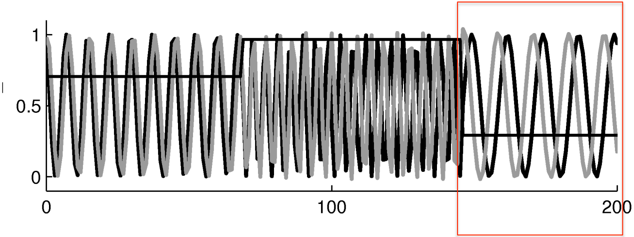
dynamical reservoir

output (or teacher) signal

The target and predicted outputs after learning



input signal



Beyond echo state networks

- Good aspects of ESNs Echo state networks can be trained very fast because they just fit a linear model
- They demonstrate that it is very important to initialize weights sensibly
- They can do impressive modeling of one-dimensional time-series
  - but they cannot compete seriously for high-dimensional data like pre-processed speech

- Bad aspects of ESNs They need many more hidden units for a given task than an RNN that learns the hidden→hidden weights
- Ilya Sutskever (2012) has shown that if the weights are initialized using the ESN methods, RNNs can be trained very effectively
  - He uses rmsprop with momentum

Beyond echo state networks

- Good aspects of ESNs Echo state networks can be trained very fast because they just fit a linear model
- They demonstrate that it is very important to initialize weights sensibly
- They can do impressive modeling of one-dimensional time-series
    - but they cannot compete seriously for high-dimensional data like pre-processed speech

- Bad aspects of ESNs They need many more hidden units for a given task than an RNN that learns the hidden→hidden weights
- Ilya Sutskever (2012) has shown that if the weights are initialized using the ESN methods, RNNs can be trained very effectively
    - He uses rmsprop with momentum

Beyond echo state networks

- Good aspects of ESNs Echo state networks can be trained very fast because they just fit a linear model
- They demonstrate that it is very important to initialize weights sensibly
- They can do impressive modeling of one-dimensional time-series
  - but they cannot compete seriously for high-dimensional data like pre-processed speech

- Bad aspects of ESNs They need many more hidden units for a given task than an RNN that learns the hidden→hidden weights
- Ilya Sutskever (2012) has shown that if the weights are initialized using the ESN methods, RNNs can be trained very effectively
  - He uses rmsprop with momentum

Demo

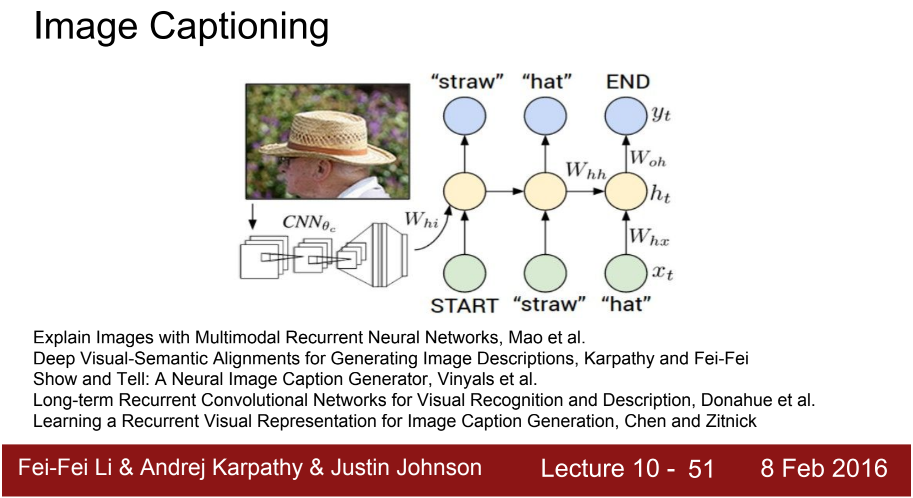
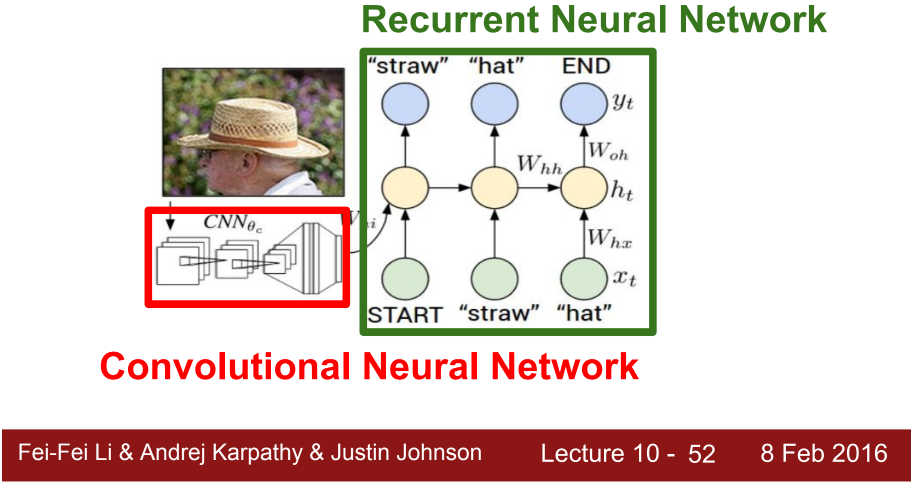


Image Captioning

Explain Images with Multimodal Recurrent Neural Networks, Mao et al.
Deep Visual-Semantic Alignments for Generating Image Descriptions, Karpathy and Fei-Fei
Show and Tell: A Neural Image Caption Generator, Vinyals et al.
Long-term Recurrent Convolutional Networks for Visual Recognition and Description, Donahue et al.
Learning a Recurrent Visual Representation for Image Caption Generation, Chen and Zitnick

Fei-Fei Li & Andrej Karpathy & Justin Johnson Lecture 10 - 51 8 Feb 2016



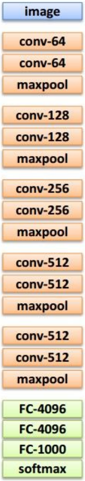
Recurrent Neural Network

Convolutional Neural Network

Fei-Fei Li & Andrej Karpathy & Justin Johnson Lecture 10 - 52 8 Feb 2016

test image

Image captioning



image

conv-64
conv-64
maxpool

conv-128
conv-128
maxpool

conv-256
conv-256
maxpool

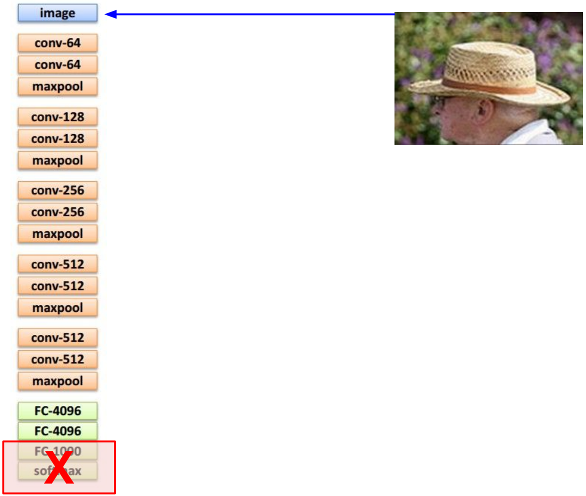
conv-512
conv-512
maxpool

conv-512
conv-512
maxpool

FC-4096
FC-4096
FC-1000
softmax

test image

Image captioning



image

conv-64

conv-64

maxpool

conv-128

conv-128

maxpool

conv-256

conv-256

maxpool

conv-512

conv-512

maxpool

conv-512

conv-512

maxpool

FC-4096

FC-4096

~~FC-1000~~

~~softmax~~

test image

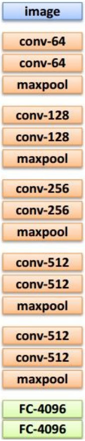
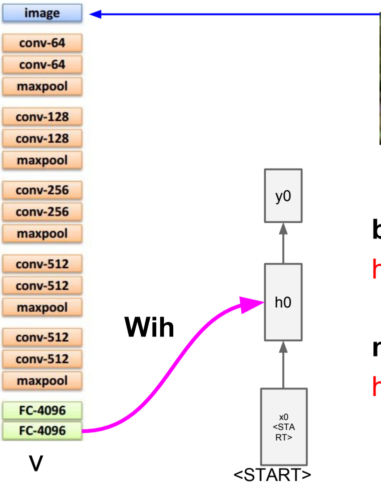


image
conv-64
conv-64
maxpool
conv-128
conv-128
maxpool
conv-256
conv-256
maxpool
conv-512
conv-512
maxpool
conv-512
conv-512
maxpool
FC-4096
FC-4096

test image

x0
<STA
RT>

<START>



test image

before:

h = tanh(Wxh * x + Whh * h)

now:

h = tanh(Wxh * x + Whh * h + **Wih * v**)

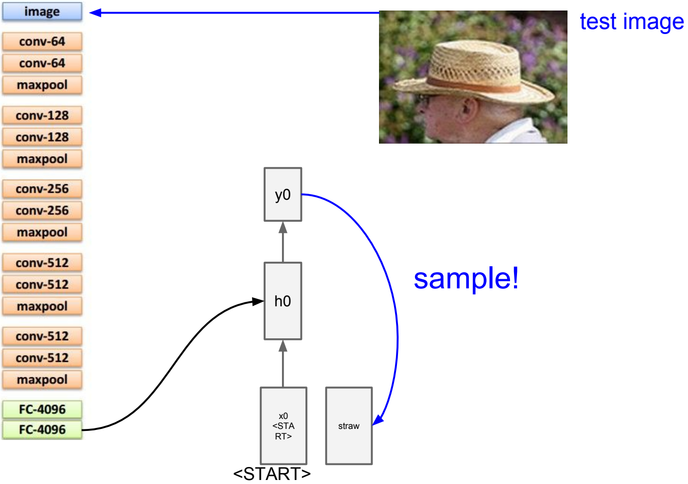


image
conv-64
conv-64
maxpool
conv-128
conv-128
maxpool
conv-256
conv-256
maxpool
conv-512
conv-512
maxpool
conv-512
conv-512
maxpool
FC-4096
FC-4096
test image
y0
h0
sample!
x0 <STA RT>
straw
<START>

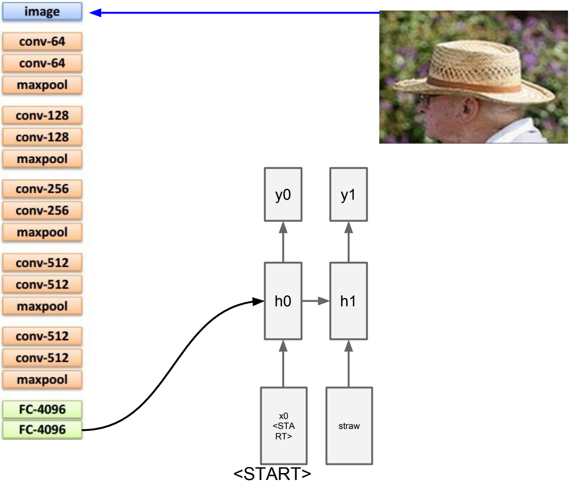


image
conv-64
conv-64
maxpool
conv-128
conv-128
maxpool
conv-256
conv-256
maxpool
conv-512
conv-512
maxpool
conv-512
conv-512
maxpool
FC-4096
FC-4096
y0
y1
h0
h1
x0 <START>
straw
<START>

test image

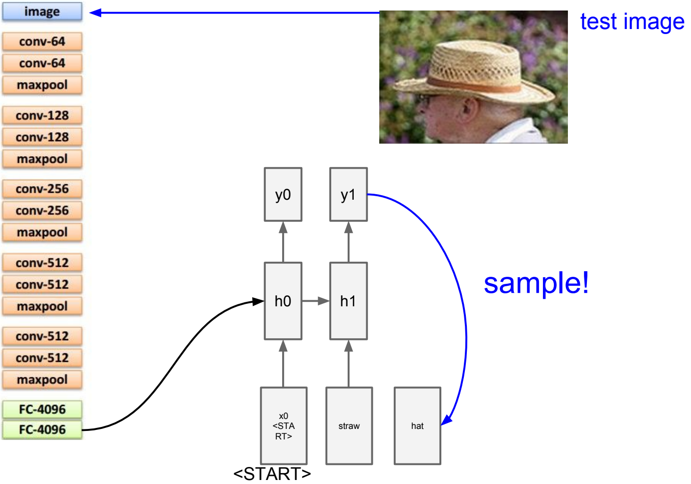


image
conv-64
conv-64
maxpool
conv-128
conv-128
maxpool
conv-256
conv-256
maxpool
conv-512
conv-512
maxpool
conv-512
conv-512
maxpool
FC-4096
FC-4096
test image
y0
y1
h0
h1
sample!
x0
<STA
RT>
straw
hat
<START>

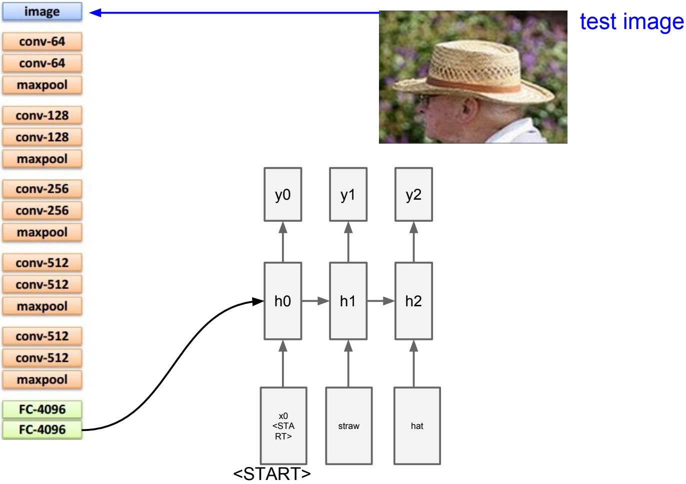


image
conv-64
conv-64
maxpool
conv-128
conv-128
maxpool
conv-256
conv-256
maxpool
conv-512
conv-512
maxpool
conv-512
conv-512
maxpool
FC-4096
FC-4096
test image
y0
y1
y2
h0
h1
h2
x0 <START>
straw
hat
<START>

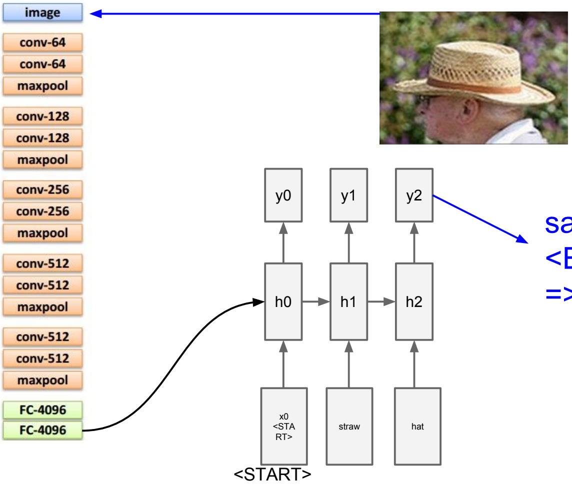


image
conv-64
conv-64
maxpool
conv-128
conv-128
maxpool
conv-256
conv-256
maxpool
conv-512
conv-512
maxpool
conv-512
conv-512
maxpool
FC-4096
FC-4096

y0
y1
y2
h0
h1
h2
x0 <START>
straw
hat
<START>

test image

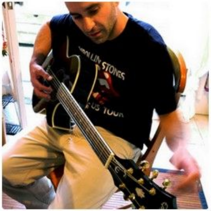
sample <END> token => finish.

Image Sentence Datasets

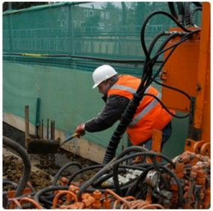
a man riding a bike on a dirt path through a forest.
bicyclist raises his fist as he rides on desert dirt trail.
this dirt bike rider is smiling and raising his fist in triumph.
a man riding a bicycle while pumping his fist in the air.
a mountain biker pumps his fist in celebration.

Microsoft COCO
[Tsung-Yi Lin et al. 2014]
mscoco.org

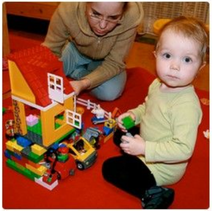
currently:
~120K images
~5 sentences each



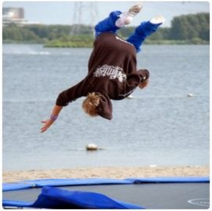
"man in black shirt is playing guitar."



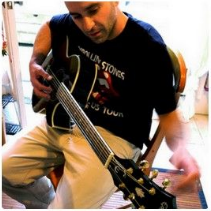
"construction worker in orange safety vest is working on road."



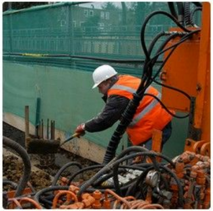
"two young girls are playing with lego toy."



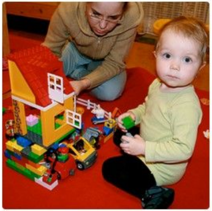
"boy is doing backflip on wakeboard."



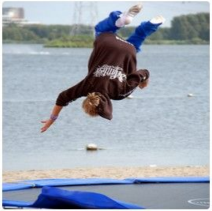
"man in black shirt is playing guitar."



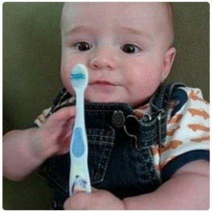
"construction worker in orange safety vest is working on road."



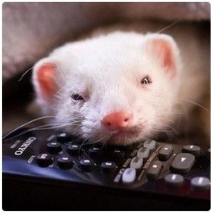
"two young girls are playing with lego toy."



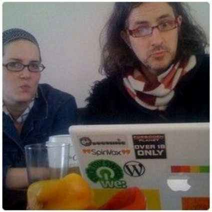
"boy is doing backflip on wakeboard."



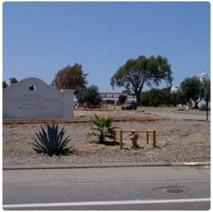
"a young boy is holding a baseball bat."



"a cat is sitting on a couch with a remote control."



"a woman holding a teddy bear in front of a mirror."

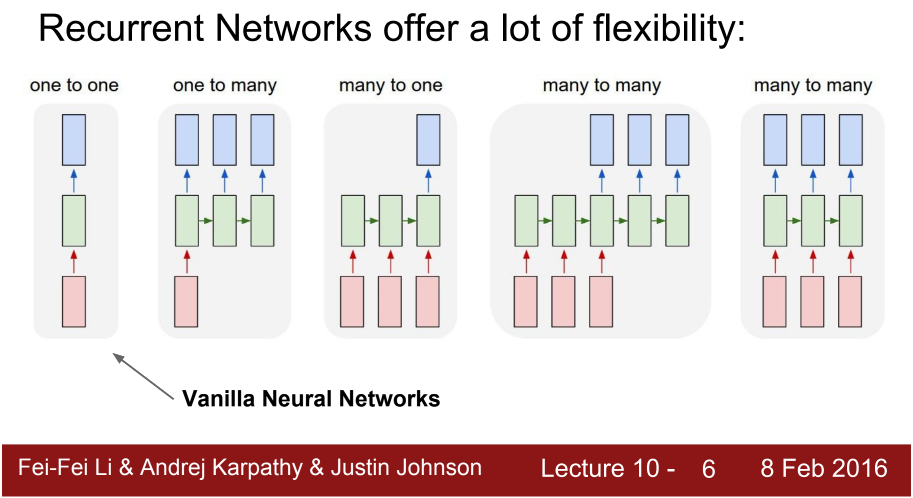


"a horse is standing in the middle of a road."

More examples

Conclusions

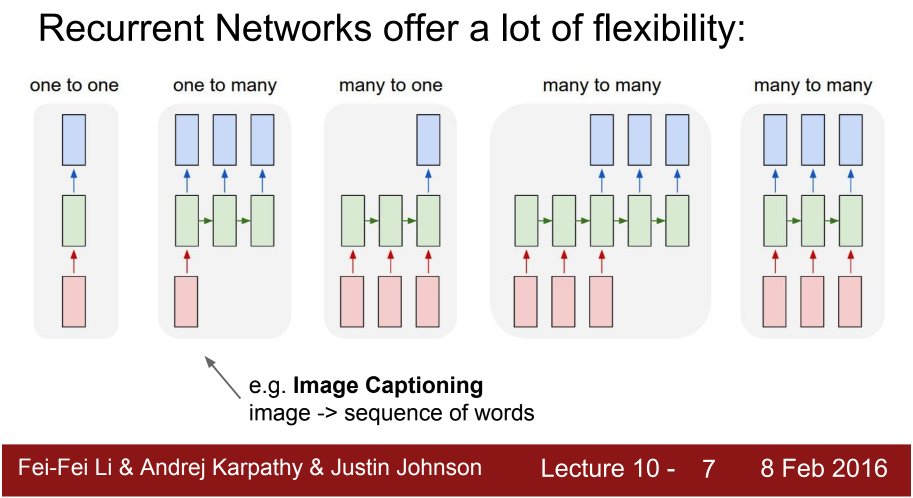
- RNNs allow a lot of flexibility in architecture design and have many applications



Recurrent Networks offer a lot of flexibility:

Vanilla Neural Networks

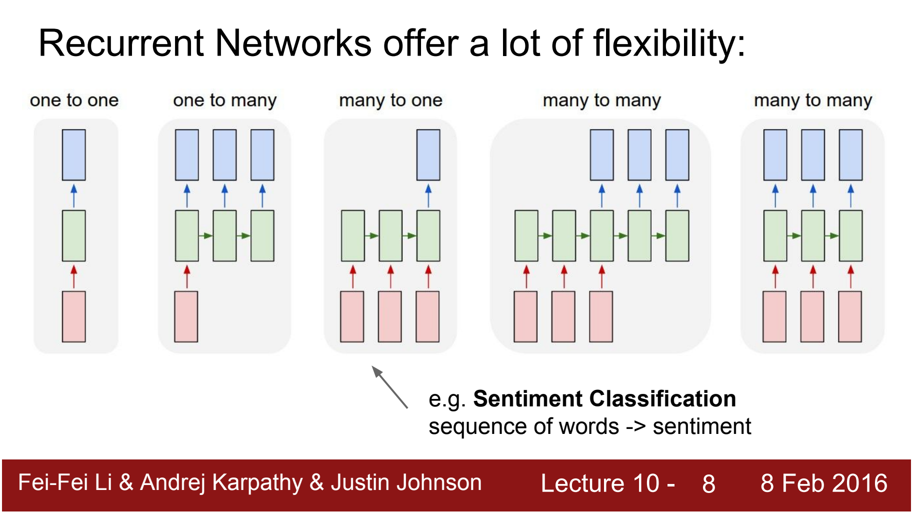
Fei-Fei Li & Andrej Karpathy & Justin Johnson Lecture 10 - 6 8 Feb 2016



Recurrent Networks offer a lot of flexibility:

e.g. **Image Captioning**
image -> sequence of words

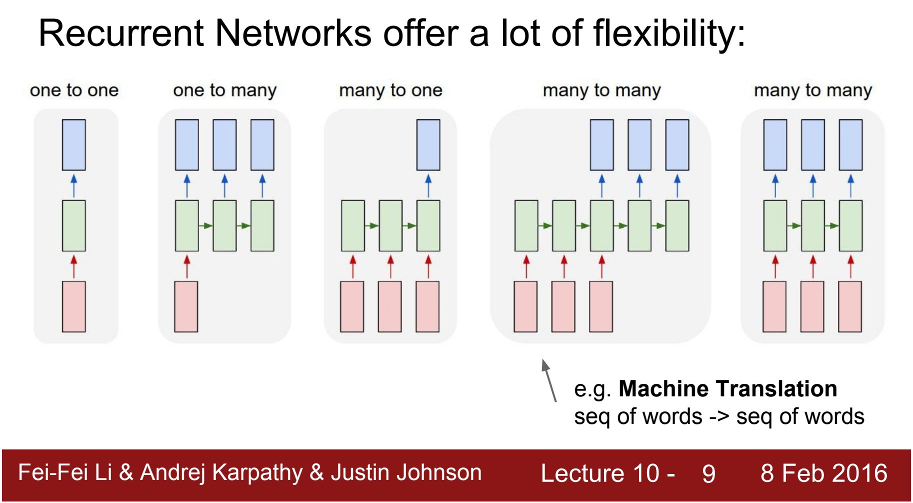
Fei-Fei Li & Andrej Karpathy & Justin Johnson Lecture 10 - 7 8 Feb 2016



Recurrent Networks offer a lot of flexibility:

e.g. **Sentiment Classification**
sequence of words -> sentiment

Fei-Fei Li & Andrej Karpathy & Justin Johnson Lecture 10 - 8 8 Feb 2016



Recurrent Networks offer a lot of flexibility:

one to one

one to many

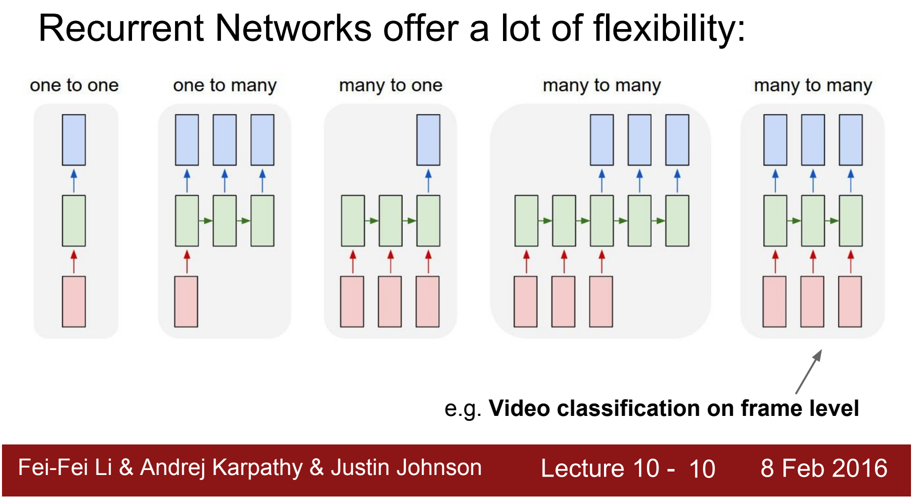
many to one

many to many

many to many

e.g. **Machine Translation**
seq of words -> seq of words

Fei-Fei Li & Andrej Karpathy & Justin Johnson Lecture 10 - 9 8 Feb 2016



Recurrent Networks offer a lot of flexibility:

e.g. **Video classification on frame level**

Fei-Fei Li & Andrej Karpathy & Justin Johnson Lecture 10 - 10 8 Feb 2016

Conclusions

- RNNs allow a lot of flexibility in architecture design and have many applications
- Vanilla RNNs are simple but don't work very well
- Common to use LSTM or GRU: their additive interactions improve gradient flow
- Backward flow of gradients in RNN can explode or vanish. Exploding is controlled with gradient clipping. Vanishing problem could be mitigated by gating (LSTM)
- Better/simpler architectures are a hot topic of current research
- Better optimization techniques such as Hessian-free methods could be used to avoid gating structures like LSTM
- Echo state networks are another possibility but may not work very well for high dimensional inputs

Conclusions

- RNNs allow a lot of flexibility in architecture design and have many applications
- Vanilla RNNs are simple but don't work very well
- Common to use LSTM or GRU: their additive interactions improve gradient flow
- Backward flow of gradients in RNN can explode or vanish. Exploding is controlled with gradient clipping. Vanishing problem could be mitigated by gating (LSTM)
- Better/simpler architectures are a hot topic of current research
- Better optimization techniques such as Hessian-free methods could be used to avoid gating structures like LSTM
- Echo state networks are another possibility but may not work very well for high dimensional inputs

Conclusions

- RNNs allow a lot of flexibility in architecture design and have many applications
- Vanilla RNNs are simple but don't work very well
- Common to use LSTM or GRU: their additive interactions improve gradient flow
- Backward flow of gradients in RNN can explode or vanish. Exploding is controlled with gradient clipping. Vanishing problem could be mitigated by gating (LSTM)
- Better/simpler architectures are a hot topic of current research
- Better optimization techniques such as Hessian-free methods could be used to avoid gating structures like LSTM
- Echo state networks are another possibility but may not work very well for high dimensional inputs

Conclusions

- RNNs allow a lot of flexibility in architecture design and have many applications
- Vanilla RNNs are simple but don’t work very well
- Common to use LSTM or GRU: their additive interactions improve gradient flow
- Backward flow of gradients in RNN can explode or vanish. Exploding is controlled with gradient clipping. Vanishing problem could be mitigated by gating (LSTM)
- Better/simpler architectures are a hot topic of current research
- Better optimization techniques such as Hessian-free methods could be used to avoid gating structures like LSTM
- Echo state networks are another possibility but may not work very well for high dimensional inputs

Conclusions

- RNNs allow a lot of flexibility in architecture design and have many applications
- Vanilla RNNs are simple but don't work very well
- Common to use LSTM or GRU: their additive interactions improve gradient flow
- Backward flow of gradients in RNN can explode or vanish. Exploding is controlled with gradient clipping. Vanishing problem could be mitigated by gating (LSTM)
- Better/simpler architectures are a hot topic of current research
- Better optimization techniques such as Hessian-free methods could be used to avoid gating structures like LSTM
- Echo state networks are another possibility but may not work very well for high dimensional inputs

Conclusions

- RNNs allow a lot of flexibility in architecture design and have many applications
- Vanilla RNNs are simple but don’t work very well
- Common to use LSTM or GRU: their additive interactions improve gradient flow
- Backward flow of gradients in RNN can explode or vanish. Exploding is controlled with gradient clipping. Vanishing problem could be mitigated by gating (LSTM)
- Better/simpler architectures are a hot topic of current research
- Better optimization techniques such as Hessian-free methods could be used to avoid gating structures like LSTM
- Echo state networks are another possibility but may not work very well for high dimensional inputs

Presentation continuing next week!

Date	Student	Package
3/3	Aakash	Tensorflow
	Soubhi	Tensorflow
3/10	**Ahmad A**	Theano
	Tamer	Theano
3/24	Ahmad M	Keras
	Obada	Keras
3/31	Muhanad	Caffe
	Siraj	Caffe
4/7	Dong	Torch
	Varun	Lasagne
4/14	Naim	MatConvNet