Lecture 17
Temporal Processing and RNNs
kindergarten classroom
“What color is the chair?”
“What color is the chair?”
red
“What will the girl do next?”
Event prediction

What can happen here?
Event prediction

What can happen here?

Video database

Event prediction

What can happen here?

Video database

Liu, Yuen, Torralba. CVPR 2009; Yuen, Torralba. ECCV 2010
What can happen here?

What can happen here?

Prediction

What can happen here?  

Nearest neighbor  

Prediction  

What can happen here?

What can happen here?

Liu, Yuen, Torralba. CVPR 2009; Yuen, Torralba. ECCV 2010

Prediction

What to predict?

**Pixels?**
- Walker, Gupta, Hebert (2014)
- Ranzato, Szlam, Bruna, Mathieu, Collobert, Chopra (2014)
- Srivastava, Mansimov, Salakhutdinov (2015)

**Motions and Trajectories?**
- Yuen, Torralba (2010)
- Pintea, van Gemert, Smeulders (2014)
- Walker, Gupta, Hebert (2015)
- Fragkiadaki, Agrawal, Levine, Malik (2016)
What to predict?

**Pixels?**
- Walker, Gupta, Hebert (2014)
- Ranzato, Szlam, Bruna, Mathieu, Collobert, Chopra (2014)
- Srivastava, Mansimov, Salakhutdinov (2015)

**Motions and Trajectories?**
- Yuen, Torralba (2010)
- Pintea, van Gemert, Smeulders (2014)
- Walker, Gupta, Hebert (2015)
- Fragkiadaki, Agrawal, Levine, Malik (2016)

**Labels?**
- Pei, Jia, Zhu (2011)
- Kitani, Ziebart, Bagnell, Hebert (2012)
- Koppula, Anand, Joachims, Saxena (2013)
- Fouhey, Zitnick (2014)
- Hoai, De la Torre (2014)
- Lan, Chen, Savarese (2014)
hug

Time

Vondrick, Pirsiavash, Torralba. CVPR 2016
Visual Representations

Deep Convolutional Network for Scene Recognition
What to predict?

Pixels?
Walker, Gupta, Hebert (2014)
Ranzato, Szlam, Bruna, Mathieu, Collobert, Chopra (2014)
Srivastava, Mansimov, Salakhutdinov (2015)

Motions and Trajectories?
Yuen, Torralba (2010)
Pintea, van Gemert, Smeulders (2014)
Walker, Gupta, Hebert (2015)
Fragkiadaki, Agrawal, Levine, Malik (2016)

Visual Representations
Generic high-level knowledge without annotations

Labels?
Pei, Jia, Zhu (2011)
Kitani, Ziebart, Bagnell, Hebert (2012)
Koppula, Anand, Joachims, Saxena (2013)
Fouhey, Zitnick (2014)
Hoai, De la Torre (2014)
Lan, Chen, Savarese (2014)
Fragkiadaki, Levine, Felsen, Malik (2015)
Learning to Anticipate

Video

Time

Prediction ConvNet

Fixed Representation
(eg AlexNet)

$\ell_2$ loss

Vondrick, Pirsiavash, Torralba. CVPR 2016
Forecasting Categories

Train linear classifier with a little labeled data

Past Image -> Predictor -> SVM -> Future label

Vondrick, Pirsiavash, Torralba. CVPR 2016
Input and Prediction

Hand Shake

Hug

Future (not seen)

Vondrick, Pirsiavash, Torralba. CVPR 2016
Sequences

“An”, “evening”, “stroll”, “through”, “a”, “city”, “square”
How do we model sequences?

- **One to One**
  - Input: No sequence
  - Output: No sequence
  - Example: “standard” classification / regression problems

- **One to Many**
  - Input: No sequence
  - Output: Sequence
  - Example: Im2Caption

- **Many to One**
  - Input: Sequence
  - Output: No sequence
  - Example: sentence classification, multiple-choice question answering

- **Many to Many**
  - Input: Sequence
  - Output: Sequence
  - Example: machine translation, video captioning, open-ended question answering, video question answering

[http://karpathy.github.io/2015/05/21/rnn-effectiveness/](http://karpathy.github.io/2015/05/21/rnn-effectiveness/)

Credit: Dhruv Batra, Andrej Karpathy
Convolutions in time
[fig from FeatureNet: Machining feature recognition based on 3D Convolution Neural Network]
It bothered him that the dog at three fourteen (seen from the side) should have the same name as the dog at three fifteen (seen from the front).
— “Funes the Memorius”, Borges 1962

“The Persistence of Memory”, Dali 1931
Douglas
Recurrent Neural Networks (RNNs)

Outputs

Hidden

Inputs

\[ W \]
Recurrent Neural Networks (RNNs)

Outputs $\hat{y}$

Hidden $h$

Inputs $x$

\[ \text{time} \]
Recurrent Neural Networks (RNNs)

\[ h^{(t)} = f(h^{(t-1)}, x^{(t)}) \]
\[ y^{(t)} = g(h^{(t)}) \]
Recurrent Neural Networks (RNNs)

\[ h^{(t)} = f(h^{(t-1)}, x^{(t)}) \]
\[ y^{(t)} = g(h^{(t)}) \]
Recurrent Neural Networks (RNNs)

\[ a^{(t)} = Wh^{(t-1)} + Ux^{(t)} + b \]

\[ h^{(t)} = \tanh(a^{(t)}) \]

\[ o^{(t)} = Vh^{(t)} + c \]

\[ \hat{y}^{(t)} = \text{softmax}(o^{(t)}) \]
Deep Recurrent Neural Networks (RNNs)

Outputs $\hat{y}$

Hidden

Inputs $x$

$\mathbf{W}_L$

$\mathbf{U}_L$

$\mathbf{W}_1$

$\mathbf{U}_1$

$\mathbf{W}_2$

$\mathbf{U}_2$

$\mathbf{V}$

Time
Backprop through time

\[
\frac{\partial \hat{y}^{(t)}}{\partial x^{(0)}} = \frac{\partial \hat{y}^{(t)}}{\partial h^{(t)}} \frac{\partial h^{(t)}}{\partial h^{(t-1)}} \cdots \frac{\partial h^{(1)}}{\partial h^{(0)}} \frac{\partial h^{(0)}}{\partial x^{(0)}}
\]
\[
\frac{\partial J}{\partial W} = \sum_{t=0}^{T} \frac{\partial \mathcal{L}(\hat{y}^{(t)}, y^{(t)})}{\partial W}
\]
Reccurrent linear layer

\[ a^{(t)} = W h^{(t-1)} + U x^{(t)} + b \]

\[ \frac{\partial L}{\partial a^{(t)}} = \frac{\partial L}{\partial h^{(t-1)}} W \]

\[ \frac{\partial L}{\partial x^{(t)}} = \frac{\partial L}{\partial a^{(t)}} U \]

\[ \frac{\partial J}{\partial W} = \sum_{t=0}^{T} \frac{\partial L(\hat{y}^{(t)}, y^{(t)})}{\partial W} \]
The problem of long-range dependences

- Capturing long-range dependences requires propagating information through a long chain of dependences.
- Old observations are forgotten
- Stochastic gradients become high variance (noisy), and gradients may vanish or explode
Memory unit

Rufus

Rufus!

W

W

time
The problem of long-range dependences

Why not remember everything?

- Memory size grows with $t$

- This kind of memory is **nonparametric**: there is no finite set of parameters we can use to model it

- RNNs make a Markov assumption — the future hidden state only depends on the immediately preceding hidden state

- By putting the right info in to the hidden state, RNNs can model dependences that are arbitrarily far apart
The problem of long-range dependences

Other methods exist that do directly link old “memories” (observations or hidden states) to future predictions:

- Temporal convolutions

- Attention (see https://arxiv.org/abs/1706.03762)

- Memory networks (see https://arxiv.org/abs/1410.3916)
LSTMs
Long Short Term Memory

A special kind of RNN designed to avoid forgetting.

Related to resnets: inductive bias is that state transition is an identity function.

This way the default behavior is not to forget an old state. Instead of forgetting by default, the network has to learn to forget.
[Slide derived from Chris Olah: http://colah.github.io/posts/2015-08-Understanding-LSTMs/]
[Slide derived from Chris Olah: http://colah.github.io/posts/2015-08-Understanding-LSTMs/]
\[ C_t = \text{Cell state} \]

[Slide derived from Chris Olah: http://colah.github.io/posts/2015-08-Understanding-LSTMs/]
Decide what information to throw away from the cell state.

Each element of cell state is multiplied by \( \sim 1 \) (remember) or \( \sim 0 \) (forget).

\[
  f_t = \sigma \left( W_f \cdot [h_{t-1}, x_t] + b_f \right)
\]
Decide what new information to add to the cell state.

which indices to write to

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

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

what to write to those indices

[Slide derived from Chris Olah: http://colah.github.io/posts/2015-08-Understanding-LSTMs/]
Forget selected old information, write selected new information.

\[ C_t = f_t \times C_{t-1} + i_t \times \tilde{C}_t \]
After having updated the cell state's information, decide what to output.

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

\[ h_t = o_t \times \tanh (C_t) \]

[Slide derived from Chris Olah: http://colah.github.io/posts/2015-08-Understanding-LSTMs/]
Examples
Texture synthesis by non-parametric sampling

Synthesizing a pixel

non-parametric sampling

Input image

Models \( P(p|N(p)) \)

[Efros & Leung 1999]
Texture synthesis with a deep net

Input partial image

Predicted color of next pixel

“white”

[PixelRNN, PixelCNN, van der Oord et al. 2016]
Input partial image

Predicted color of next pixel

“white”

[PixelRNN, PixelCNN, van der Oord et al. 2016]
Recall from lecture 12: we can represent colors as discrete classes

\[ y \in \mathbb{R}^{H \times W \times K} \]

\[ \mathcal{L}(y, f_\theta(x)) = H(y, \text{softmax}(f_\theta(x))) \]
And we can interpret the learner as modeling $P(\text{next pixel} | \text{previous pixels})$:

**Softmax regression** (a.k.a. multinomial logistic regression)

\[ \hat{y} \equiv [P_\theta(Y = 1|X = x), \ldots, P_\theta(Y = K|X = x)] \quad \text{← predicted probability of each class given input } x \]

\[ H(y, \hat{y}) = -\sum_{k=1}^{K} y_k \log \hat{y}_k \quad \text{← picks out the -log likelihood of the ground truth class } y \]

\[ f^* = \arg\min_{f \in \mathcal{F}} \sum_{i=1}^{N} H(y_i, \hat{y}_i) \quad \text{← max likelihood learner!} \]
Network output

... →

P(next pixel | previous pixels)

\[ P(p_i | p_1, \ldots, p_{i-1}) \]
Network output

\[ p_i \sim P(p_i|p_1, \cdots, p_{i-1}) \]
Network output

\[ p_i \sim P(p_i|p_1, \ldots, p_{i-1}) \]
$p_i \sim P(p_i | p_1, \cdots, p_{i-1})$
Network output

\[ p_i \sim P(p_i | p_1, \cdots, p_{i-1}) \]
\( p_1 \sim P(p_1) \)
\( p_2 \sim P(p_2|p_1) \)
\( p_3 \sim P(p_3|p_1, p_2) \)
\( p_4 \sim P(p_4|p_1, p_2, p_3) \)

\[ \{p_1, p_2, p_3, p_4\} \sim P(p_4|p_1, p_2, p_3)P(p_3|p_1, p_2)P(p_2|p_1)P(p_1) \]

\( p_i \sim P(p_i|p_1, \ldots, p_{i-1}) \)
Autoregressive probability model

\[ p \sim \prod_{i=1}^{N} P(p_i|p_1, \ldots, p_{i-1}) \]

\[ P(p) = \prod_{i=1}^{N} P(p_i|p_1, \ldots, p_{i-1}) \] \hspace{1cm} \text{General product rule}

The sampling procedure we defined above takes exact samples from the learned probability distribution (pmf).

Multiplying all conditionals evaluates the probability of a full joint configuration of pixels.
Autoregressive probability model

\[ p \sim P(p) \]

Models that allow us to sample, i.e. \textit{generate}, images from scratch are called \textbf{generative models}.

We will see more examples in a future lecture.
Samples from PixelRNN
Image completions (conditional samples) from PixelRNN

[PixelRNN, van der Oord et al. 2016]
Abundant Unlabeled Video

• Lacks semantic labels, but:
• Abundantly available
• Contextual relations “for free”
Why is video generation hard?

Uncertainty in future causes blurry predictions

Predictions from Ranzato et al 2015
How many ways can this image change?

64x64 image

Even tiny images can change in many ways

Vondrick, Pirsiavash, Torralba. NIPS 2016 (to appear)
Learning a Model of Scene Dynamics

- General purpose model of scene dynamics
  - Generation: future prediction
  - Recognition: action classification
- Learn model of plausible scene dynamics
  - Hard because no negative data

Vondrick, Pirsiavash, Torralba. NIPS 2016
Video Generator

Vondrick, Pirsiavash, Torralba. NIPS 2016
Generated Videos (not real)

Beach

Golf Course

Train Station

Hospital

Vondrick, Pirsiavash, Torralba. NIPS 2016
Foreground/Background

Background + Foreground = Generation

Vondrick, Pirsiavash, Torralba. NIPS 2016 (to appear)
Predicting Dynamics

Vondrick, Pirsiavash, Torralba. NIPS 2016
Object Emergence

Find what input activates a “neuron” the most
Object Emergence

“person” emergent hidden unit
• Task: learn to predict plausible sounds for silent videos

• To do well: must learn about material, physical interaction
Audio Signal Processing

signal

44.1 kHz

[Sethares, 2007]
Audio Signal Processing

signal

windows

windowed segments

spectrum of segment

FFT

analysis

44.1 kHz

[Sethares, 2007]
Audio Signal Processing

[Sethares, 2007]
Gestalt principles

- Not grouped
- Proximity
- Similarity
- Similarity
- Common Fate
- Common Region
AUDITORY SCENE ANALYSIS
The Perceptual Organization of Sound

Albert S. Bregman

Albert S. Bregman
Grouping by harmonicity

The tone that is not related to the shared fundamental, i.e., is mistuned, it will be heard as a separate sound. The rest are all grouped into a single sound.
Grouping by common fate

Two subsets undergo different patterns of change: a faster and a slower one. This is played twice. We hear two distinct tones, each formed from one of the subsets. The slow-moving one sounds pure and the fast-moving one, rich.
Perceptual continuation
Perceptual continuation

![Diagram showing frequency over time](image)
The Greatest Hits dataset
The Greatest Hits dataset

- 978 videos of people probing scenes with a drumstick
- 46,620 hits and scratches
- Material, action, and reaction labels (used for analysis)
Sound and materials

Waveform

cochlear filtering

envelope & compressive nonlinearity

Mean cushion cochleagram

Frequency subband

Time → Impact time

Source: McDermott & Simoncelli 2011
Sound and materials

Concrete

Grass

Cushion
Predicting sound features

Cochleagram

Regression

Video

Time
Predicting sound features

- Two-stream CNN: color + spacetime images
Generating a waveform

Audio

Parametric synthesis

Cochleagram

[Slaney95]
Generating a waveform

Example-based synthesis

Audio

Cochleagram

Example-based synthesis
Recap: full model

Video

Cochleagram

Waveform

Sound matching

CNN

LSTM

Time →
Other actions?
Other actions?
Some Experiments on the Recognition of Speech, with One and with Two Ears*

E. COLIN CHERRY
Imperial College, University of London, England, and Research Laboratory of Electronics, Massachusetts Institute of Technology, Cambridge, Massachusetts

The cocktail party problem
Unsupervised noisy object discovery
Unsupervised noisy object discovery
Unsupervised sound separation
Related work

- Andrew Owens, Alexei A. Efros. *Audio-Visual Scene Analysis with Self-Supervised Multisensory Features*
- Arda Senocak, Tae-Hyun Oh, Junsik Kim, Ming-Hsuan Yang, In So Kweon. *Learning to Localize Sound Source in Visual Scenes*
- Ariel Ephrat, Inbar Mosseri, Oran Lang, Tali Dekel, Kevin Wilson, Avinatan Hassidim, William T. Freeman, Michael Rubinstein. *Looking to Listen at the Cocktail Party: A Speaker-Independent Audio-Visual Model for Speech Separation*
- Relja Arandjelovic, Andrew Zisserman. *Objects that Sound*
- Ruohan Gao, Rogerio Feris, Kristen Grauman. *Learning to Separate Object Sounds by Watching Unlabeled Video*
Sound Separation: Segmentation of Spectrograms

5.5kHz

6s

Time

Frequency

Violin

Tuba
Sound Separation: Segmentation of Spectrograms

- Spectrogram of Mixture
  - 5.5kHz
  - 6s
- Spectrogram Mask
- Output Spectrogram
  - 5.5kHz
  - 6s

Predict Violin
This is what we want at the end:

Input video (multiple instruments) → Video Analysis Network → K visual feature maps → Audio Analysis Network → U-Net → K audio feature maps → Audio Synthesizer Network → Predicted masks \( \hat{M} \) → iSTFT → Sound of a Pixel

Advanced components include:
- STFT
- Dilated ResNet
- U-Net
- K visual feature maps
- K audio feature maps
- iSTFT

Input sound (multiple instruments)
Sound object separation

K object classes

Video Analysis Network

Estimated sound 1 $\tilde{S}_1$

Sound object separation

K audio Channels

video 1 sound $S_1$

$S_1 + S_2$

video sound 2
Training Time

Video Analysis Network

- Dilated ResNet
- Max pooling
- K visual feature maps

Audio Synthesizer Network

- Predicted masks $\hat{M}$
- iSTFT

Audio Analysis Network

- STFT
- Sound spectrogram
- U-Net
- K audio feature maps

Audio Synthesizer Network

- iSTFT
- Sound of target video

Visual features

$\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_K$
Test Time: using Pixel Feature instead

Audio Synthesizer Network

Video Analysis Network

Audio Analysis Network

STFT

Predicted masks

Video Analysis Network

U-Net

Dilated ResNet

K visual feature maps

K audio feature maps

Sound spectrogram

K visual features

Sound of a Pixel

iSTFT
http://sound-of-pixels.csail.mit.edu/
http://sound-of-pixels.csail.mit.edu/
6. Assembly Line

http://sound-of-pixels.csail.mit.edu/