Language Technologies Institute

Carnegie Mellon University

Multimodal Machine Learning

Lecture 5.1: Multimodal alignment

Louis-Philippe Morency

** Original course co-developed with Tadas Baltrusaitis.*
Spring 2021 edition taught by Yonatan Bisk

Administrative Stuff

Second Project Assignment (Due Sunday 10/10)

Main goals:

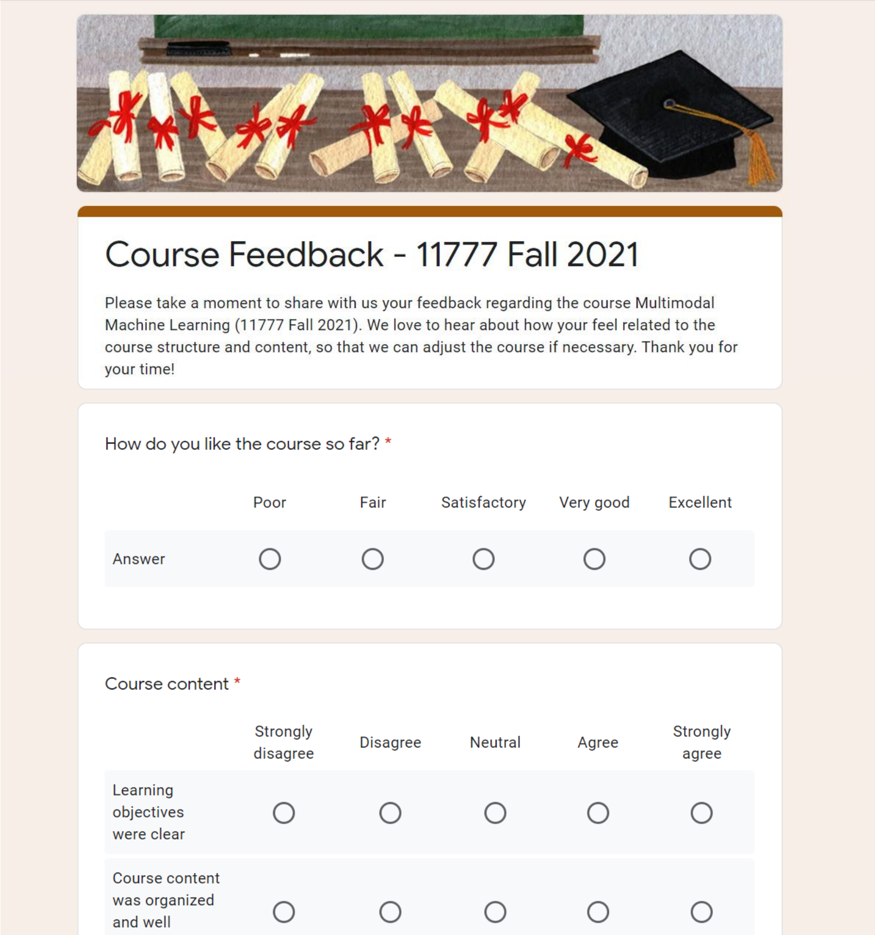
- Get familiar with unimodal representations
  - Learn about tools based on CNNs, word2vec, BERT, …
- Understand the structure in your unimodal data
  - Perform some visualization of the unimodal data
- Explore qualitatively the unimodal data
  - How does it relate to your labels? Look at specific examples

Examples of unimodal analyses:

- What are the different verbs used in the VQA questions?
- What objects do not get detected? Are they important?
- Visualize face embeddings with respect of emotion labels

Share Your Thoughts!

https://forms.gle/ZUBcMZVf4Ttv2uQ66



Course Feedback - 11777 Fall 2021

Please take a moment to share with us your feedback regarding the course Multimodal Machine Learning (11777 Fall 2021). We love to hear about how your feel related to the course structure and content, so that we can adjust the course if necessary. Thank you for your time!

How do you like the course so far? *

	Poor	Fair	Satisfactory	Very good	Excellent
Answer	○	○	○	○	○

Course content *

	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
Learning objectives were clear	○	○	○	○	○
Course content was organized and well	○	○	○	○	○

Deadline
Please submit your feedback about this course before this Sunday 10/3

Optional,
but greatly appreciated! ☺

Anonymous, by default.

- You can optionally share your email address if you want us to follow-up with you directly.

Language Technologies Institute

Carnegie Mellon University

Multimodal Machine Learning

Lecture 5.1: Multimodal alignment

Louis-Philippe Morency

** Original course co-developed with Tadas Baltrusaitis.*
Spring 2021 edition taught by Yonatan Bisk

Lecture objectives

- Multimodal alignment
- Explicit signal alignment
  - Dynamic Time Warping
    - Canonical Time Warping
  - Multi-view video alignment
  - Speech alignment
    - Connectionist Temporal Classification
- Implicit alignment
  - Hard attention
  - Spatial Transformer Networks

Multimodal alignment

Multimodal-alignment

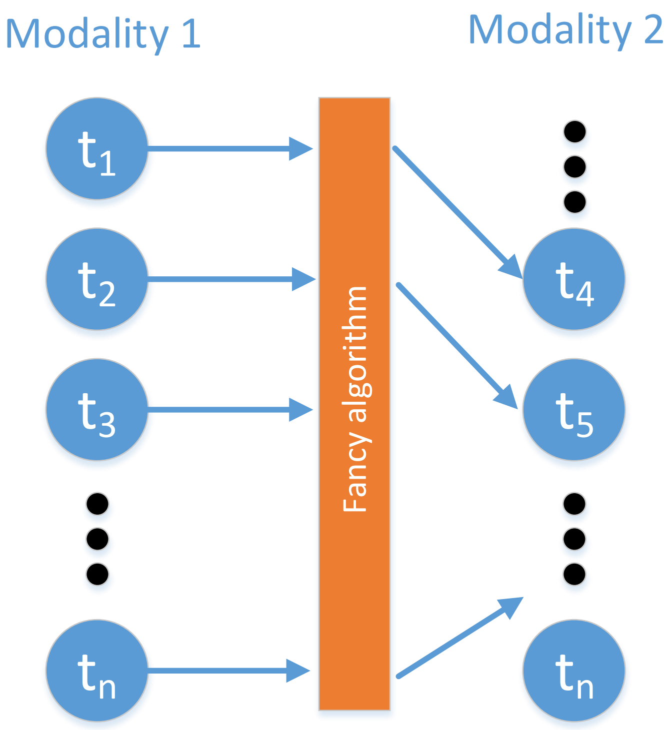
Multimodal alignment – finding relationships and correspondences between two or more modalities

Two types

- **Explicit** – alignment is the task in itself
- **Implicit / Latent** – alignment helps when solving a different task (for example using “Attention” module)

Examples ?

- Images with captions
- Recipe steps with a how-to video
- Phrases/words of translated sentences

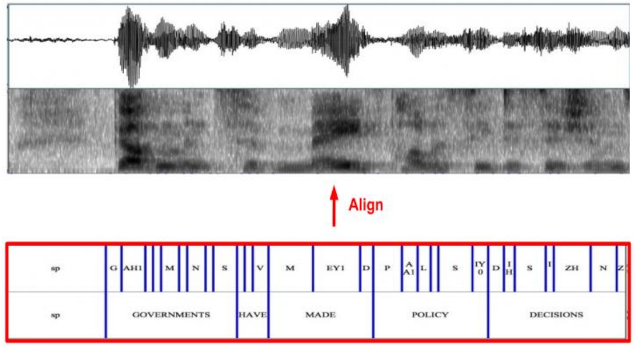


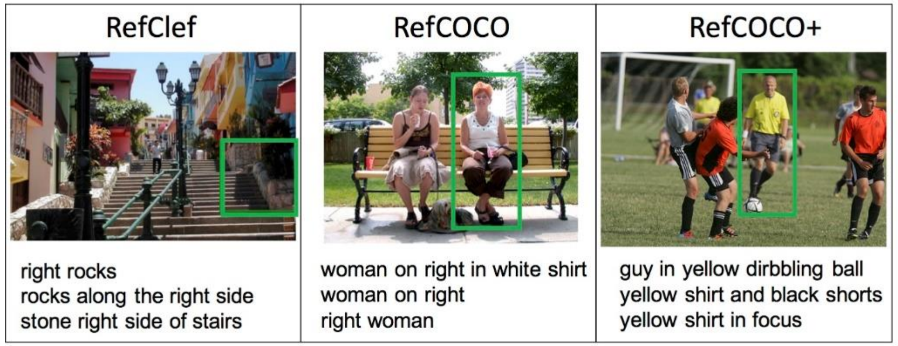
Explicit multimodal-alignment

Explicit alignment - goal is to find correspondences between modalities

In other words: the alignment is part of the loss function

- Aligning speech signal to a transcript
- Aligning two out-of sync sequences
- Language grounding

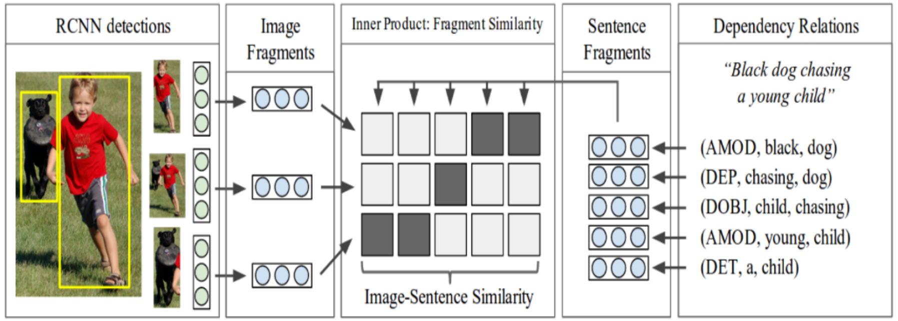


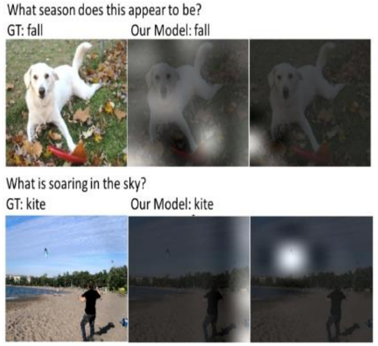


Implicit multimodal-alignment

Implicit alignment - uses internal latent alignment of modalities in order to better solve various problems

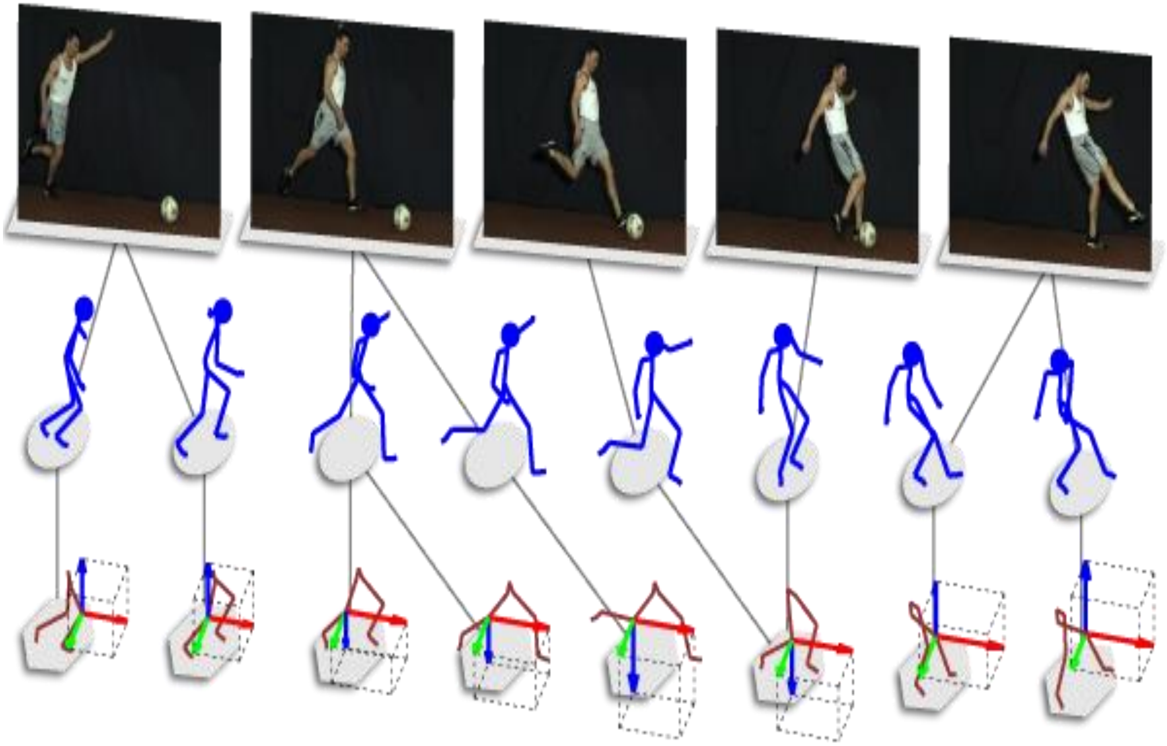
- Machine Translation
- Cross-modal retrieval
- Image & Video Captioning
- Visual Question Answering





Explicit alignment

Temporal sequence alignment



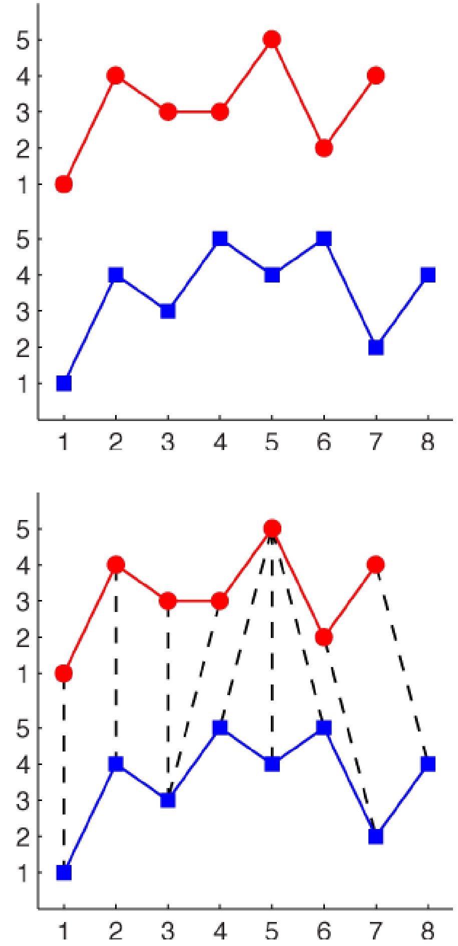
Applications:
- Re-aligning asynchronous data
- Finding similar data across modalities (we can estimate the aligned cost)
- Event reconstruction from multiple sources

Let's start unimodal – Dynamic Time Warping

- We have two unaligned temporal unimodal signals
  - $\mathbf{X} = \left[\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{n_x}\right] \in \mathbb{R}^{d \times n_x}$
  - $\mathbf{Y} = \left[\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_{n_y}\right] \in \mathbb{R}^{d \times n_y}$
- Find set of indices to minimize the alignment difference:

$$L(\boldsymbol{p}^x, \boldsymbol{p}^y) = \sum_{t=1}^{l} \left\| \boldsymbol{x}_{\boldsymbol{p}_t^x} - \boldsymbol{y}_{\boldsymbol{p}_t^y} \right\|_2^2$$

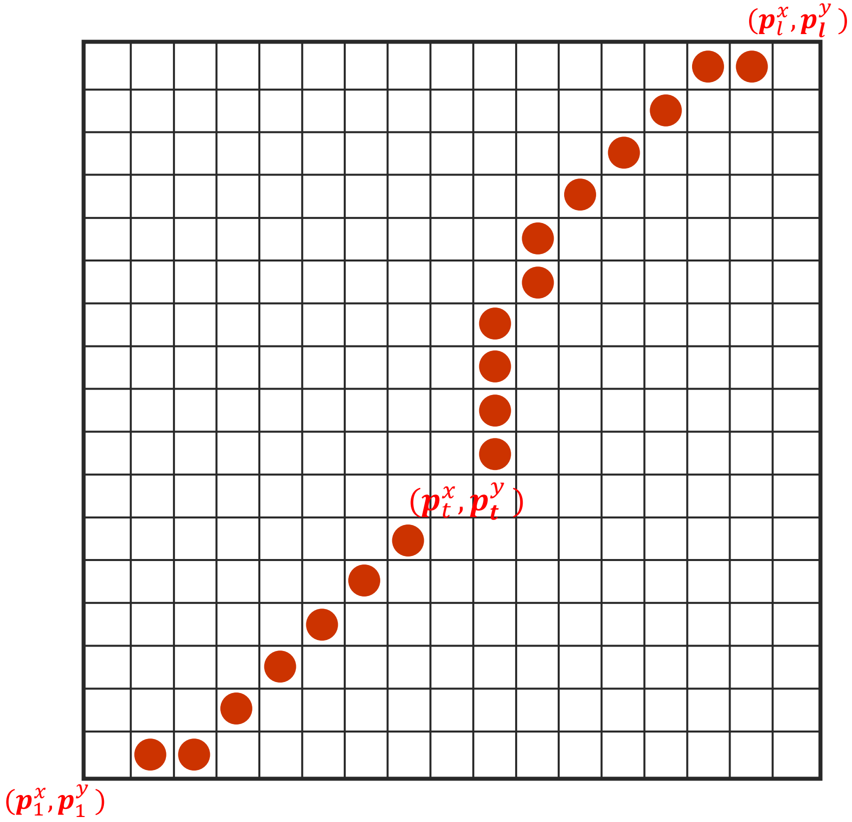
- Where $\boldsymbol{p}^x$ and $\boldsymbol{p}^y$ are index vectors of same length
- Dynamic Time Warping is designed to find these index vectors



Dynamic Time Warping continued

Lowest cost path in a cost matrix

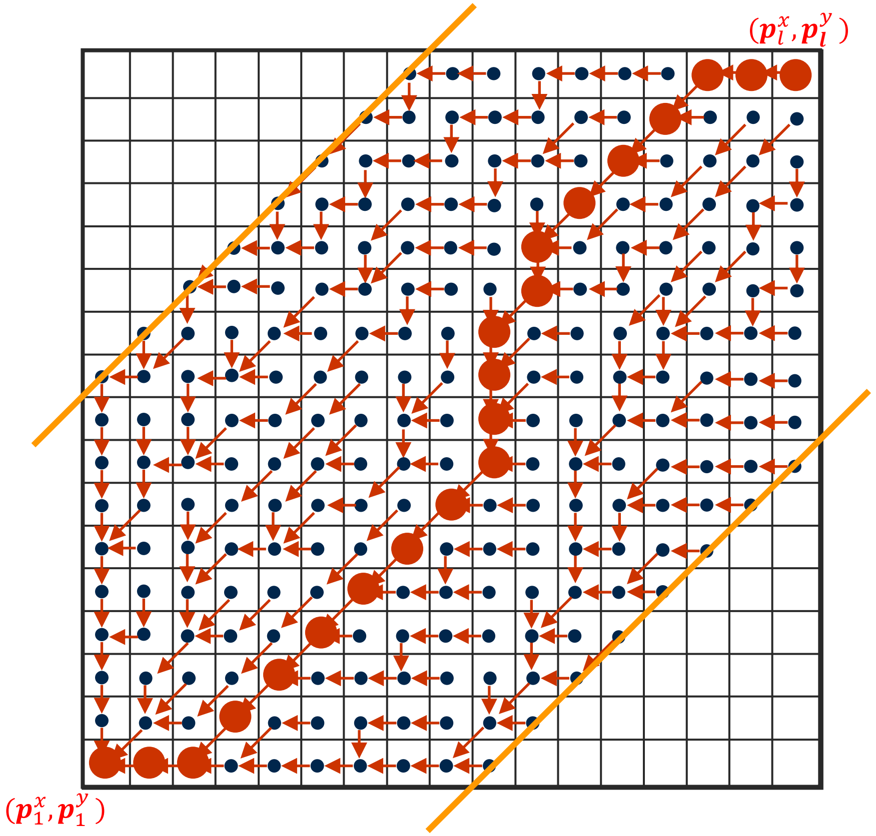
- Restrictions?
  - Monotonicity – no going back in time
  - Continuity - no gaps
  - Boundary conditions - start and end at the same points
  - Warping window - don’t get too far from diagonal
  - Slope constraint – do not insert or skip too much



Dynamic Time Warping continued

Lowest cost path in a cost matrix

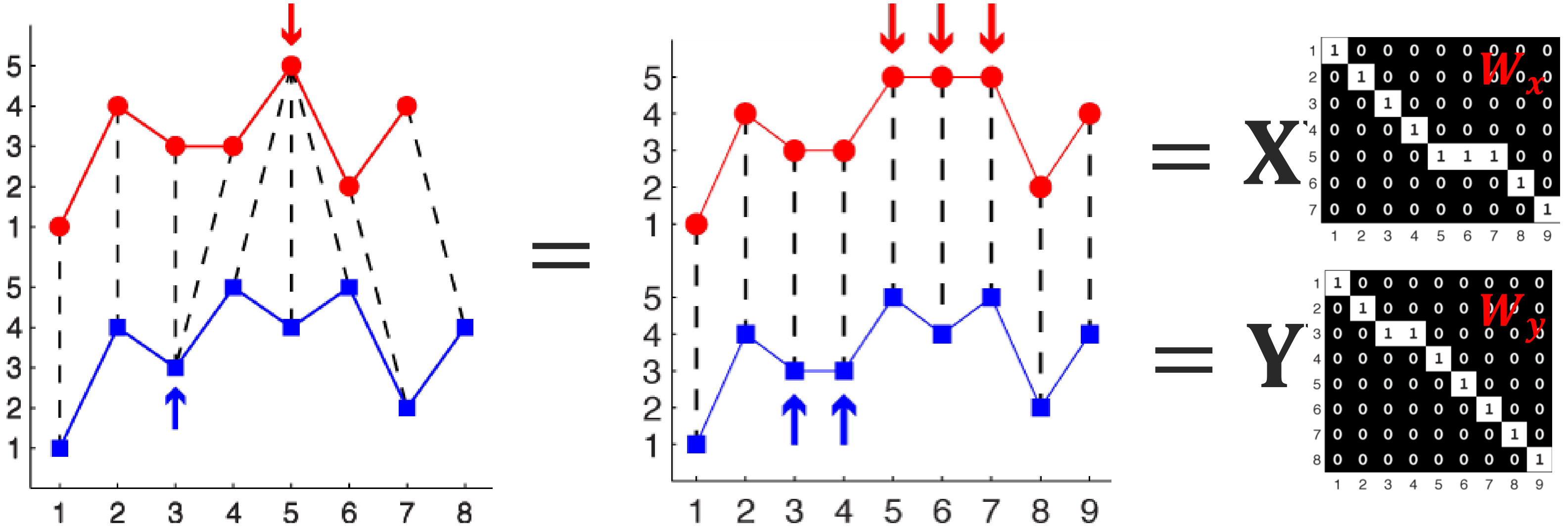
- Solved using dynamic programming while respecting the restrictions



DTW alternative formulation

$$L(\boldsymbol{p}^x, \boldsymbol{p}^y) = \sum_{t=1}^{l} \left\| \boldsymbol{x}_{p_t^x} - \boldsymbol{y}_{p_t^y} \right\|_2^2$$

Replication doesn't change the objective!



Alternative objective:

$$L(\boldsymbol{W}_x, \boldsymbol{W}_y) = \left\| \boldsymbol{X}\boldsymbol{W}_x - \boldsymbol{Y}\boldsymbol{W}_y \right\|_F^2$$

Frobenius norm $\|\boldsymbol{A}\|_F^2 = \sum_i \sum_j |a_{i,j}|^2$

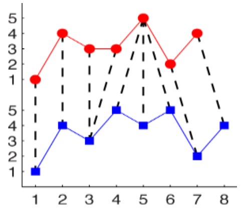
$\boldsymbol{X}, \boldsymbol{Y}$ – original signals (same #rows, possibly different #columns)

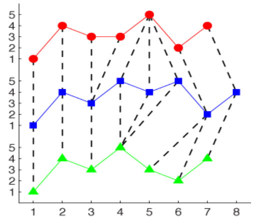
$\boldsymbol{W}_x, \boldsymbol{W}_y$ - alignment matrices

A differentiable version of DTW also exists…
https://arxiv.org/pdf/1703.01541.pdf

DTW – Some Limitations

- Computationally complex





m sequences

$$O\left(\prod_{i=1}^{m} n_i\right)$$

- Sensitive to outliers

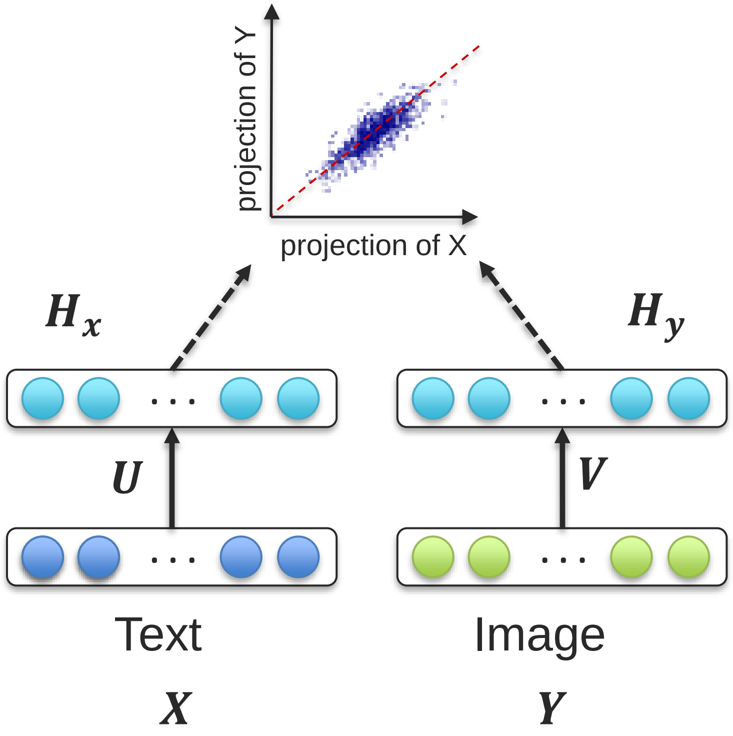
- Unimodal!

Canonical Correlation Analysis reminder

maximize: $tr(\boldsymbol{U}^{\boldsymbol{T}}\boldsymbol{\Sigma}_{\boldsymbol{XY}}\boldsymbol{V})$

subject to: $\boldsymbol{U}^{\boldsymbol{T}}\boldsymbol{\Sigma}_{\boldsymbol{YY}}\boldsymbol{U} = \boldsymbol{V}^{\boldsymbol{T}}\boldsymbol{\Sigma}_{\boldsymbol{YY}}\boldsymbol{V} = \boldsymbol{I}\ ,\ \boldsymbol{u}_{(j)}^{\boldsymbol{T}}\boldsymbol{\Sigma}_{\boldsymbol{XY}}\boldsymbol{v}_{(i)} = \boldsymbol{0}$ for $i \neq j$

1. Linear projections maximizing correlation
2. Orthogonal projections
3. Unit variance of the projection vectors



Canonical Correlation Analysis reminder

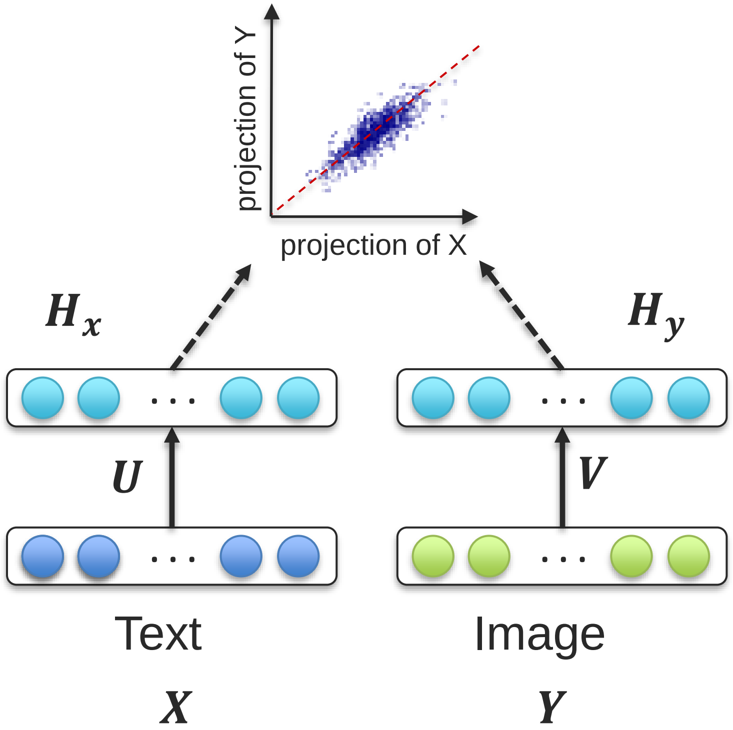
When data is normalized it is actually equivalent to smallest RMSE reconstruction

CCA loss can also be re-written as:

$$L(\boldsymbol{U},\boldsymbol{V}) = ||\mathbf{U}^T\mathbf{X} - \mathbf{V}^T\mathbf{Y}||_F^2$$

subject to:

$$\boldsymbol{U}^T\boldsymbol{\Sigma}_{YY}\boldsymbol{U} = \boldsymbol{V}^T\boldsymbol{\Sigma}_{YY}\boldsymbol{V} = \boldsymbol{I},\ \boldsymbol{u}_{(j)}^T\boldsymbol{\Sigma}_{XY}\boldsymbol{v}_{(i)} = \mathbf{0}$$



Canonical Time Warping

Dynamic Time Warping + Canonical Correlation Analysis = Canonical Time Warping

$$L(\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W}_x, \boldsymbol{W}_y) = \left\| \mathbf{U}^T \mathbf{X} \mathbf{W}_\mathbf{x} - \mathbf{V}^T \mathbf{Y} \mathbf{W}_\mathbf{y} \right\|_F^2$$

- Allows to align multi-modal or multi-view (same modality but from a different point of view)
- $\boldsymbol{W}_x, \boldsymbol{W}_y$ – temporal alignment
- $\boldsymbol{U}, \boldsymbol{V}$ – cross-modal (spatial) alignment

[Canonical Time Warping for Alignment of Human Behavior, Zhou and De la Tore, 2009]

Canonical Time Warping

$$L(\boldsymbol{U},\boldsymbol{V},\boldsymbol{W}_x,\boldsymbol{W}_y) = \left\|\mathbf{U}^T\mathbf{X}\mathbf{W}_\mathbf{x} - \mathbf{V}^T\mathbf{Y}\mathbf{W}_\mathbf{y}\right\|_F^2$$

Optimized by Coordinate-descent – fix one set of parameters, optimize another

Generalized Eigen-decomposition

$\boldsymbol{W}_x, \boldsymbol{W}_y$ $\quad$ $\boldsymbol{U}, \boldsymbol{V}$

Gauss-Newton

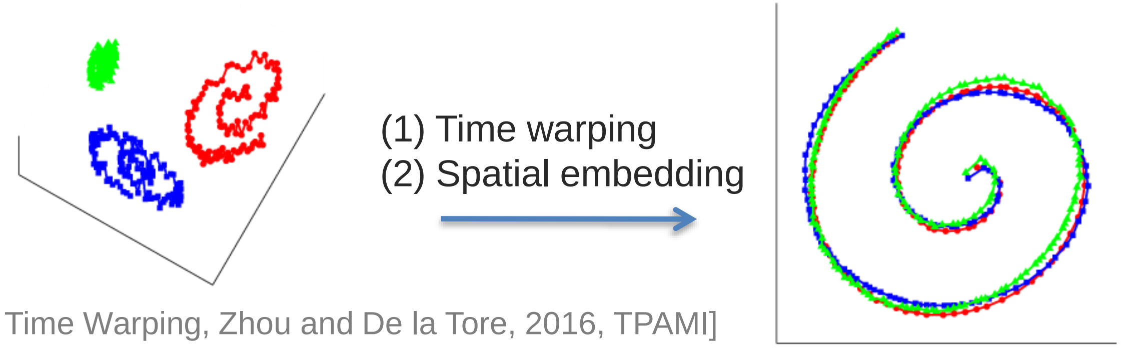
[Canonical Time Warping for Alignment of Human Behavior, Zhou and De la Tore, 2009, NIPS]

Generalized Time warping

Generalize to multiple sequences all of different modality

$$L(\boldsymbol{U_i}, \boldsymbol{W_i}) = \sum_{i=1} \sum_{j=1} \left\| \mathbf{U}_i^T \mathbf{X}_\mathrm{i} \mathbf{W}_\mathrm{i} - \mathbf{U}_j^T \mathbf{X}_\mathrm{j} \mathbf{W}_j \right\|_F^2$$

- $\boldsymbol{W_i}$ – set of temporal alignments
- $\boldsymbol{U_i}$ – set of cross-modal (spatial) alignments



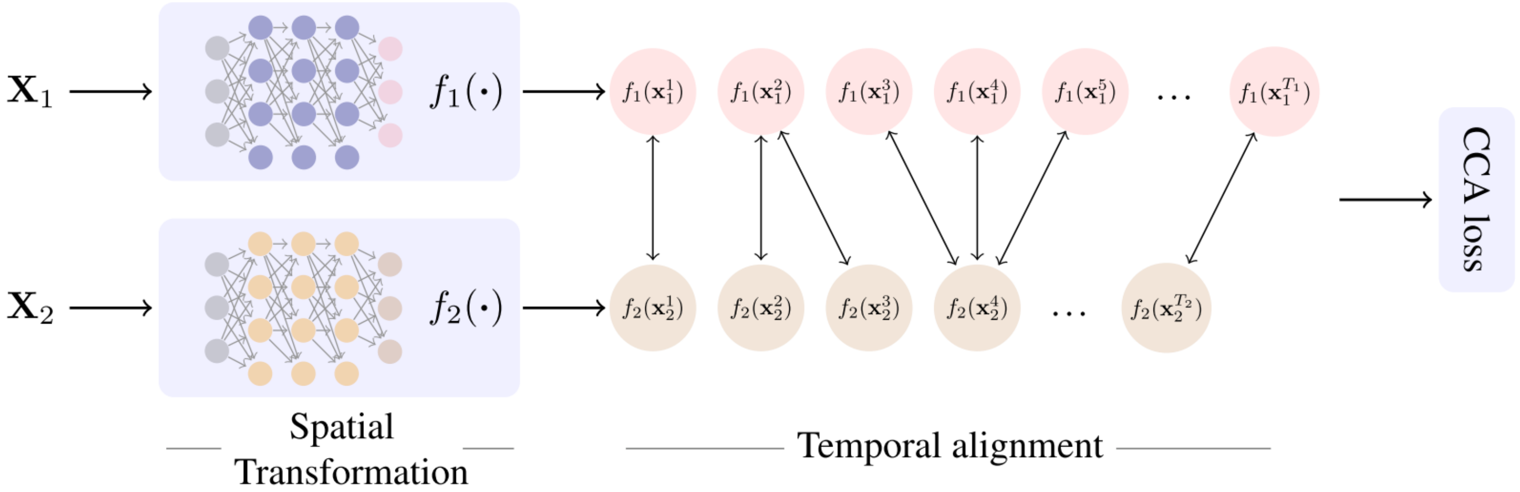
[Generalized Canonical Time Warping, Zhou and De la Tore, 2016, TPAMI]

Alignment examples (multimodal)

Deep Canonical Time Warping

$$L(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \boldsymbol{W}_x, \boldsymbol{W}_y) = \left\| f_{\boldsymbol{\theta}_1}(\mathbf{X})\mathbf{W_x} - f_{\boldsymbol{\theta}_2}(\mathbf{Y})\mathbf{W_y} \right\|_F^2$$

Could be seen as generalization of DCCA and GTW



[Deep Canonical Time Warping, Trigeorgis et al., 2016, CVPR]

Deep Canonical Time Warping

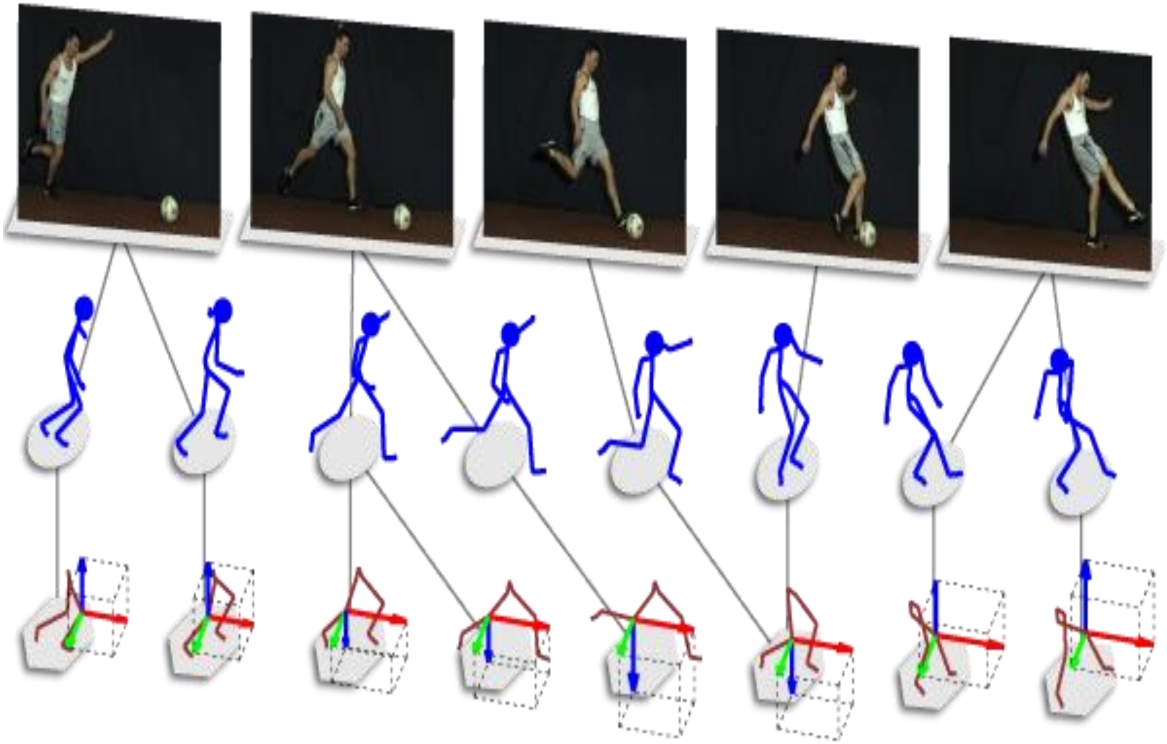
$$L(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \boldsymbol{W}_x, \boldsymbol{W}_y) = \left\| f_{\boldsymbol{\theta}_1}(\mathbf{X})\mathbf{W_x} - f_{\boldsymbol{\theta}_1}(\mathbf{Y})\mathbf{W_y} \right\|_F^2$$

- The projections are orthogonal (like in DCCA)
- Optimization is again iterative:
  - Solve for alignment ($\boldsymbol{W}_x, \boldsymbol{W}_y$) with fixed projections ($\boldsymbol{\theta}_1, \boldsymbol{\theta}_2$)
    - Eigen decomposition
  - Solve for projections ($\boldsymbol{\theta}_1, \boldsymbol{\theta}_2$) with fixed alignment ($\boldsymbol{W}_x, \boldsymbol{W}_y$)
    - Gradient descent
  - Repeat till convergence

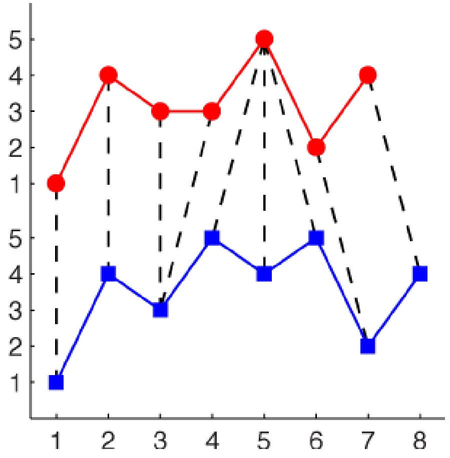
[Deep Canonical Time Warping, Trigeorgis et al., 2016, CVPR]

Multi-View
Video Alignment and
Representation Learning

Temporal sequence alignment

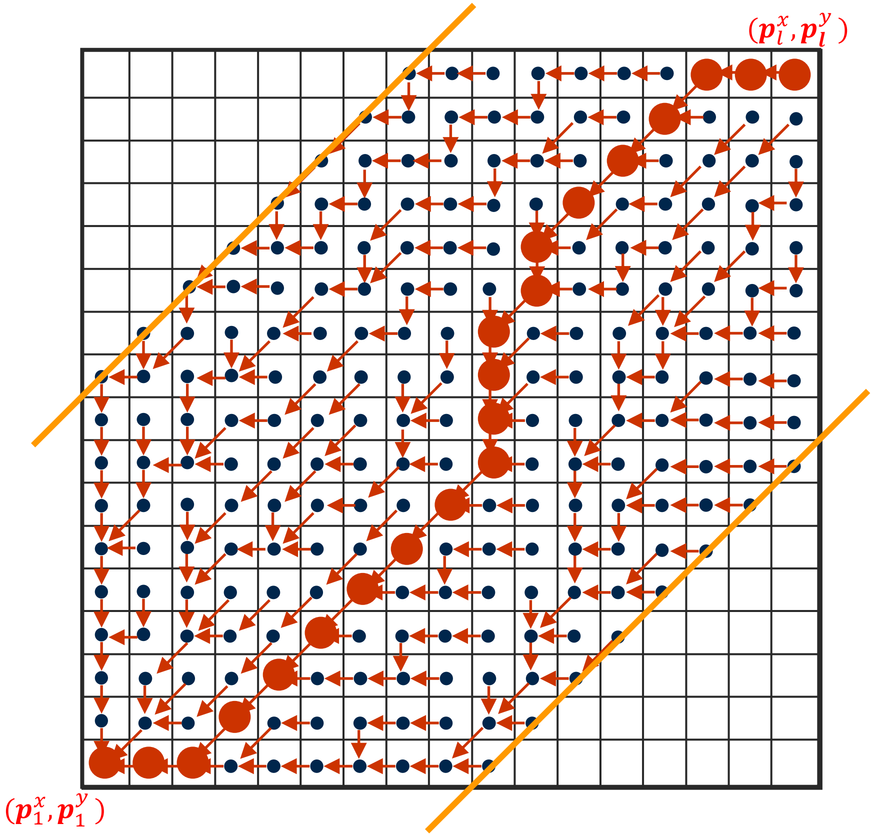


Reminder: Dynamic Time Warping for Sequence Alignment



$$L(\boldsymbol{p}_t^x, \boldsymbol{p}_t^y) = \sum_{t=1}^{l} \left\| \boldsymbol{x}_{\boldsymbol{p}_t^x} - \boldsymbol{y}_{\boldsymbol{p}_t^y} \right\|_2^2$$

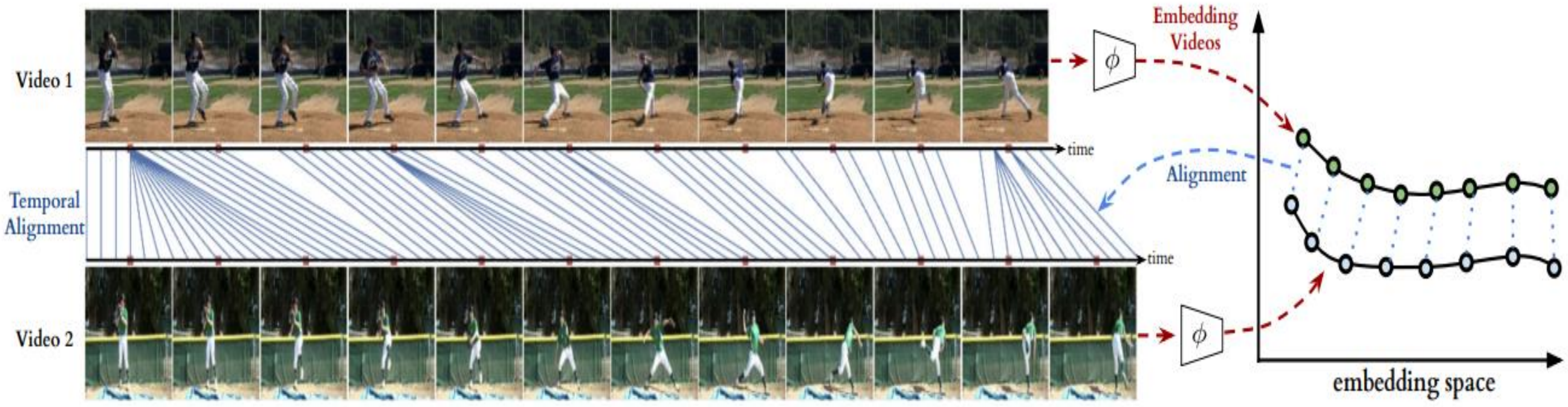
Solved with dynamic programming…



But how to do alignment and representation learning at the same time?

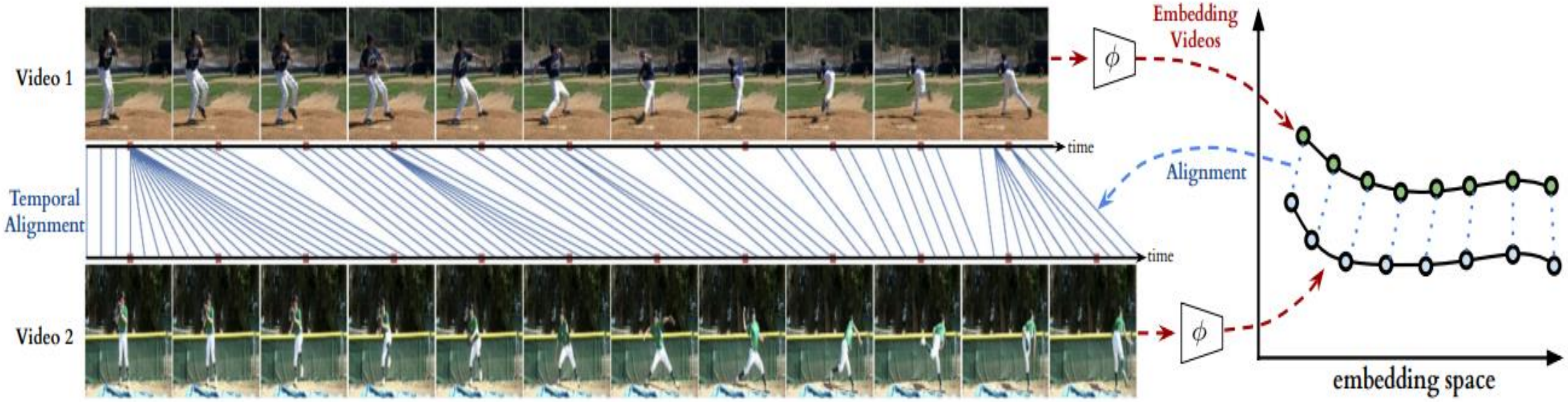
Temporal Alignment and Neural Representation Learning

Premise: we have paired video sequences that can be be temporally aligned



How can we define a loss function to enforce the alignment between sequences while at the same time learning good representations?

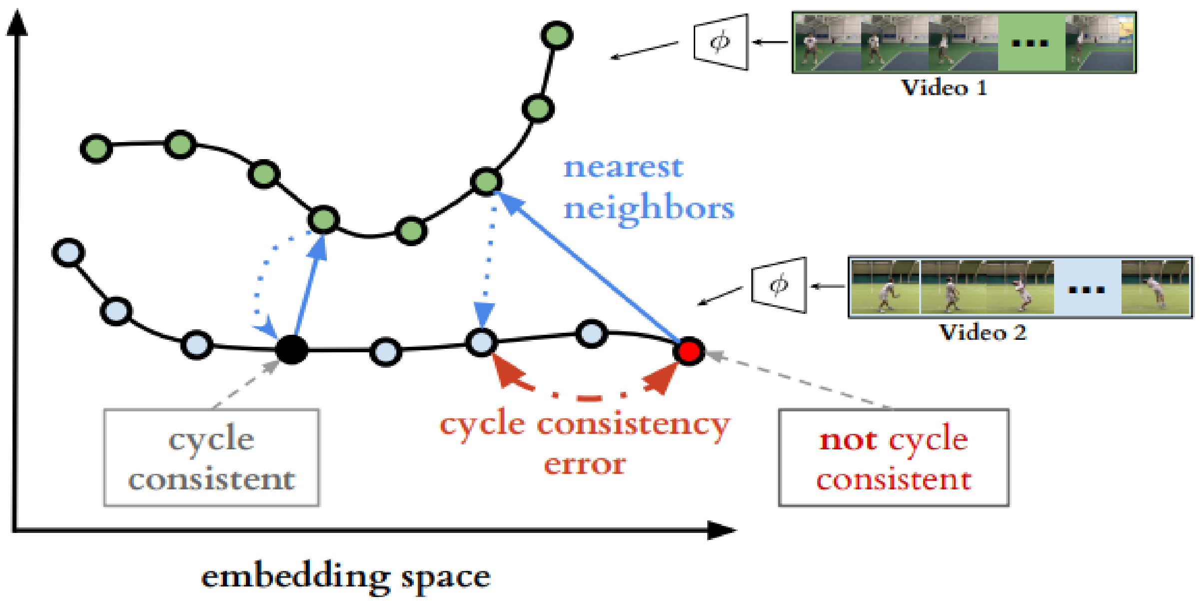
Temporal Cycle-Consistency Learning



Self-supervised approach to learn an embedding space where two similar video sequences can be aligned temporally

Temporal Cycle-Consistency Learning

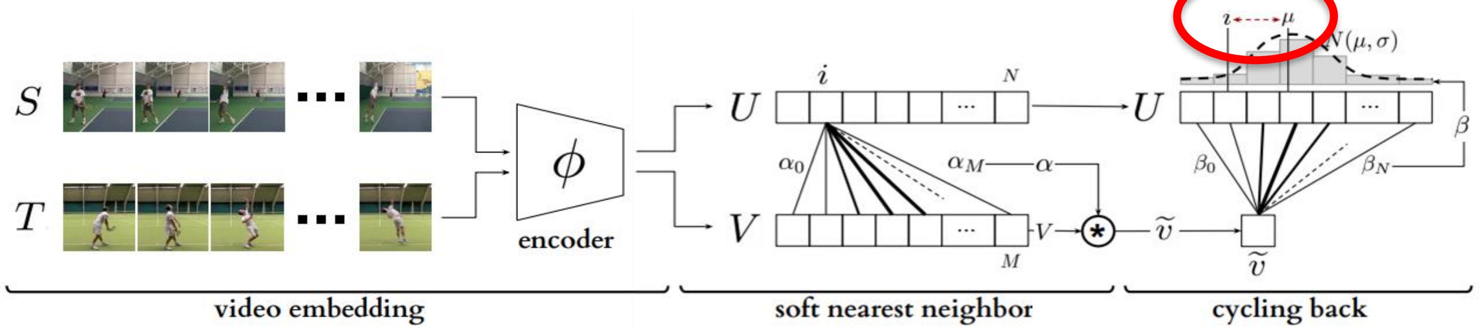
Solution: Representation learning by enforcing **Cycle consistency**



Main idea: My closest neighbor also views me as their closest neighbor

Temporal Cycle-Consistency Learning

penalty!



Compute "soft" / "weighted" nearest neighbour:

distances: $$\alpha_j = \frac{e^{-||u_i - v_j||^2}}{\sum_k^M e^{-||u_i - v_k||^2}}$$

Soft nearest neighbor: $$\widetilde{v} = \sum_j^M \alpha_j v_j,$$

Find the nearest neighbor the other way and then penalize the distance:

$$\beta_k = \frac{e^{-||\widetilde{v} - u_k||^2}}{\sum_j^N e^{-||\widetilde{v} - u_j||^2}}$$

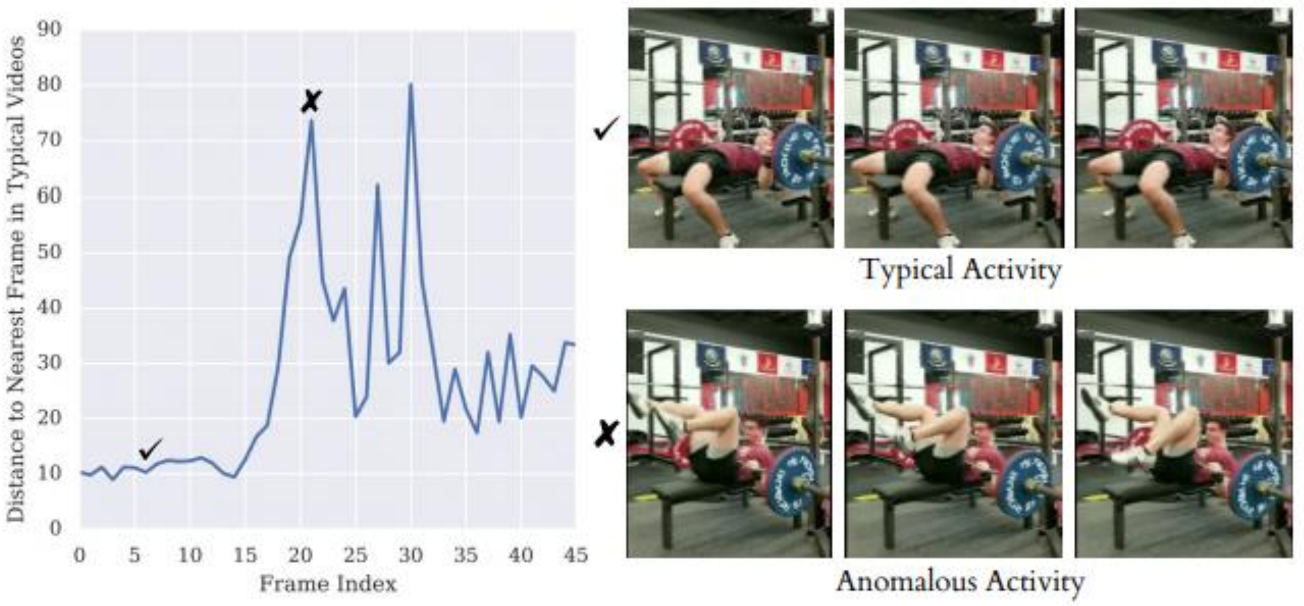
$$L_{cbr} = \frac{|i - \mu|^2}{\sigma^2} + \lambda \log(\sigma)$$

Temporal Cycle-Consistency Learning

Nearest Neighbour Retrieval

Temporal Cycle-Consistency Learning

Anomaly Detection



How could you extend this idea to multimodal?
Course project idea? ☺

Alignment for Speech Recognition

Architecture of Speech Recognition

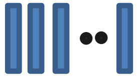
$$\widehat{\boldsymbol{W}} = \underset{\boldsymbol{W}}{\operatorname{argmax}} P(\boldsymbol{W}|\boldsymbol{O})$$

$$= \underset{\boldsymbol{W}}{\operatorname{argmax}} P(\boldsymbol{A}|\boldsymbol{O})P(\boldsymbol{O}|\boldsymbol{Q})P(\boldsymbol{Q}|\boldsymbol{L})P(\boldsymbol{L}|\boldsymbol{W})P(\boldsymbol{W})$$

$P(\boldsymbol{A}|\boldsymbol{O})$ Feature Extraction → $P(\boldsymbol{O}|\boldsymbol{Q})$ Frame Classification → $P(\boldsymbol{Q}|\boldsymbol{L})$ Sequence Model → $P(\boldsymbol{L}|\boldsymbol{W})$ Lexicon Model → $P(\boldsymbol{W})$ Language Model →

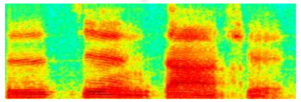
$\boldsymbol{A}$

Speech Audio

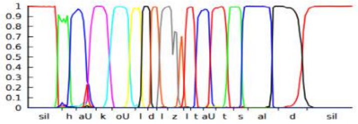


$\boldsymbol{O}$

Feature Frames



Spectogram



$\boldsymbol{Q}$

Sequence States

t ah m aa t ow

$\boldsymbol{L}$

Phonemes

tomato

$\boldsymbol{W}$

Words

$\boldsymbol{W}$

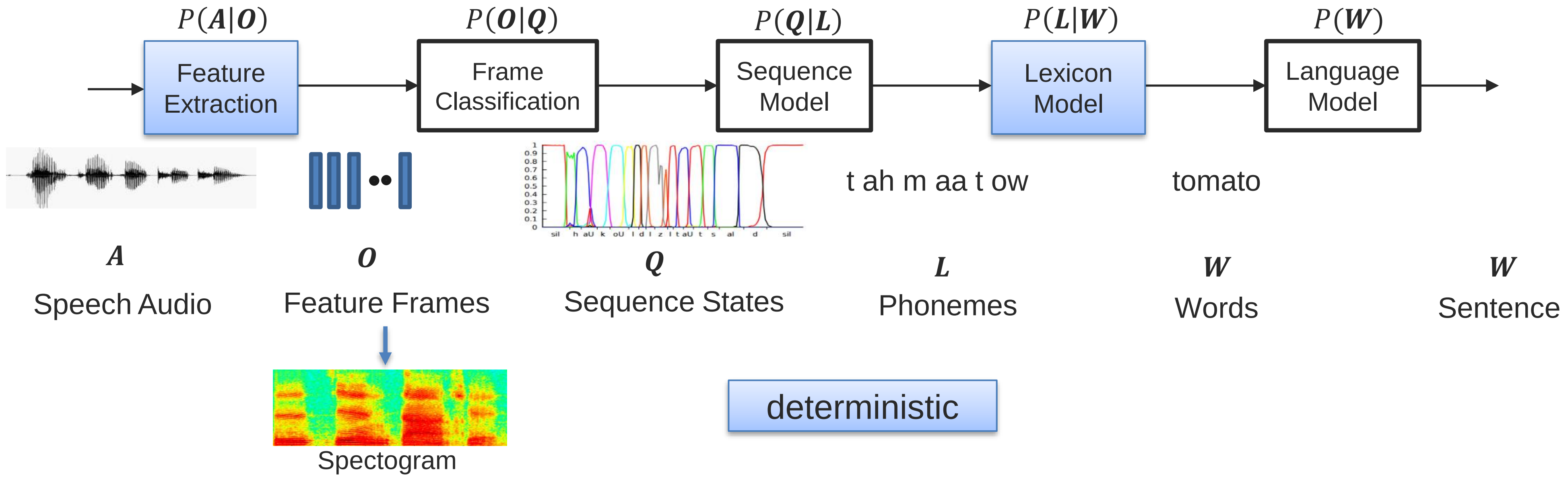
Sentence

http://slazebni.cs.illinois.edu/spring17/lec26_audio.pdf

Architecture of Speech Recognition

$$\widehat{\boldsymbol{W}} = \underset{\boldsymbol{W}}{\operatorname{argmax}}\, P(\boldsymbol{W}|\boldsymbol{O})$$

$$= \underset{\boldsymbol{W}}{\operatorname{argmax}}\, P(\boldsymbol{A}|\boldsymbol{O})P(\boldsymbol{O}|\boldsymbol{Q})P(\boldsymbol{Q}|\boldsymbol{L})P(\boldsymbol{L}|\boldsymbol{W})P(\boldsymbol{W})$$

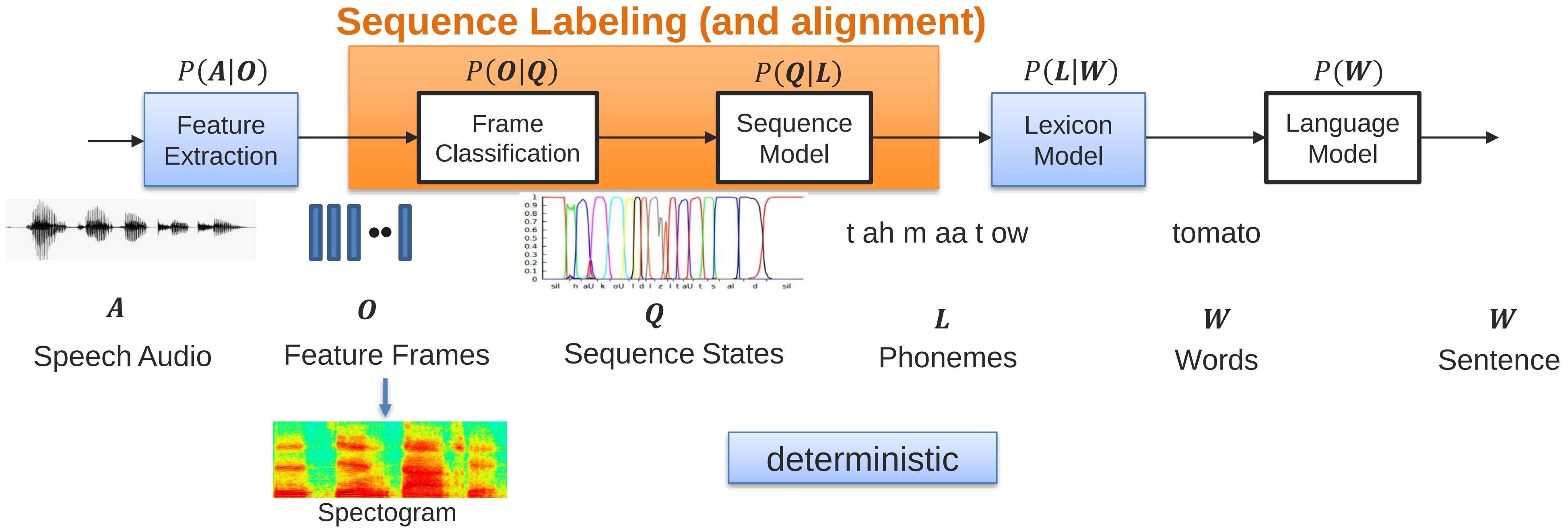


http://slazebni.cs.illinois.edu/spring17/lec26_audio.pdf

Architecture of Speech Recognition

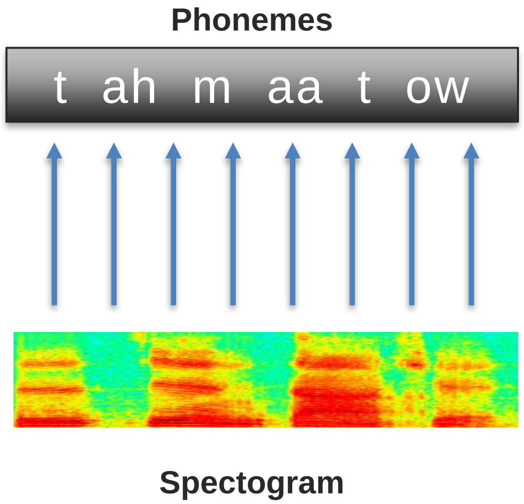
$$\hat{\boldsymbol{W}} = \underset{\boldsymbol{W}}{\operatorname{argmax}}\, P(\boldsymbol{W}|\boldsymbol{O})$$

$$= \underset{\boldsymbol{W}}{\operatorname{argmax}}\, P(\boldsymbol{A}|\boldsymbol{O})P(\boldsymbol{O}|\boldsymbol{Q})P(\boldsymbol{Q}|\boldsymbol{L})P(\boldsymbol{L}|\boldsymbol{W})P(\boldsymbol{W})$$



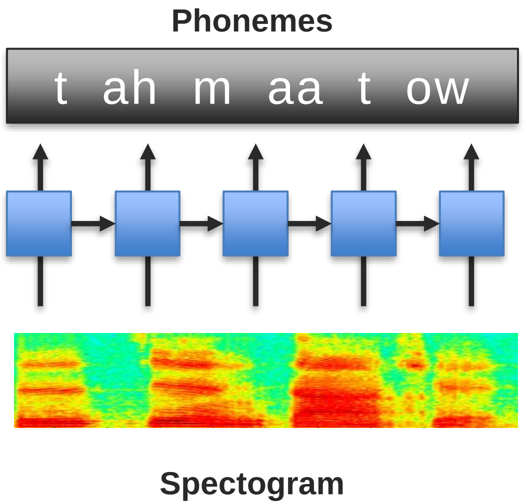
http://slazebni.cs.illinois.edu/spring17/lec26_audio.pdf

Sequence Labeling and Alignment

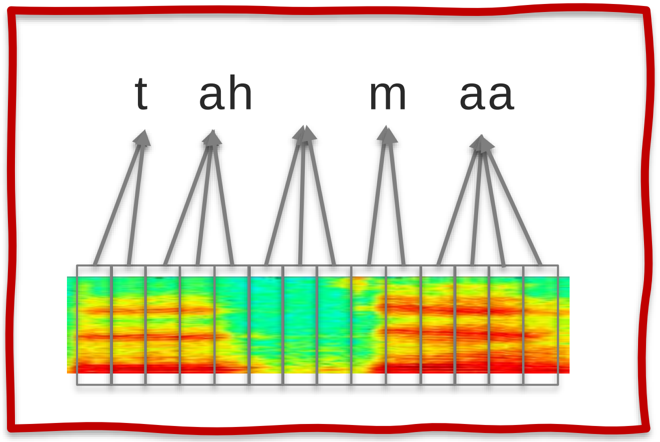


How can we predict the sequence of phoneme labels from the sequence of audio frames?

Potential Solution: Sequence Labeling with RNN



Challenge: many-to-1 alignment



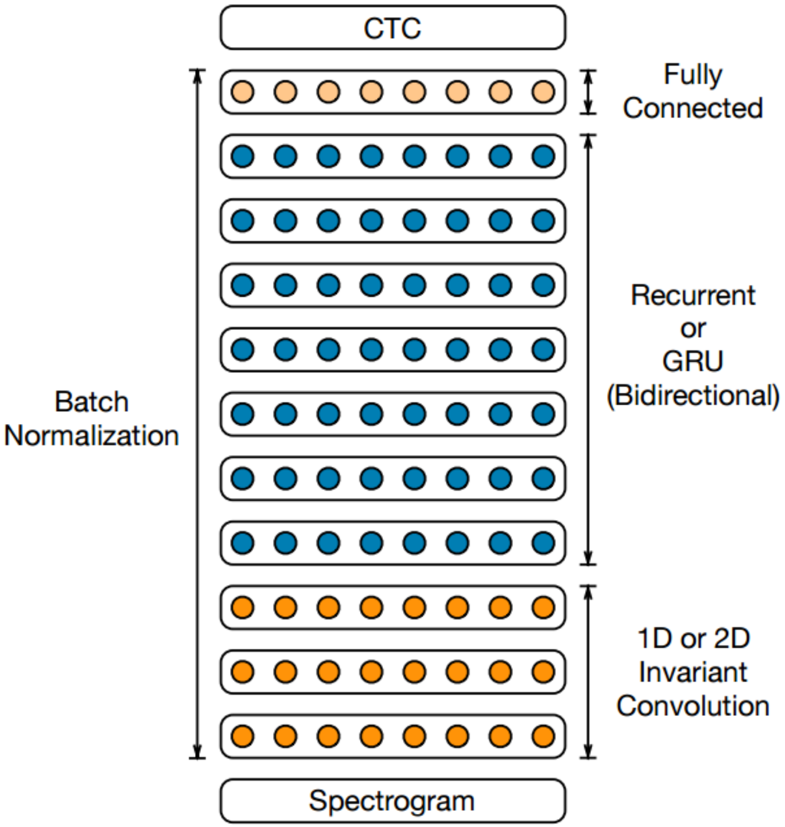
What should be the loss function?

Connectionist Temporal Classification (CTC)

CTC is used in speech recognition systems that were almost in par with human performances.

Test set	Deep speech 2	Human
WSJ eval'92	3.60	5.03
WSJ eval'93	4.98	8.08
LibriSpeech test-clean	5.33	5.83
LibriSpeech test-other	13.25	12.69

Deep Speech 2



Amodei, Dario, et al. "Deep speech 2: End-to-end speech recognition in english and mandarin." (2015)

Connectionist Temporal Classification (CTC)

Training examples $S = \{(\boldsymbol{x_1}, \boldsymbol{z_1}), \dots (\boldsymbol{x_N}, \boldsymbol{z_N})\} \in \mathcal{D}_{\mathcal{X}\times\mathcal{Z}}$

$\boldsymbol{x} \in \mathcal{X}$ are spectrogram frames

$$\boldsymbol{x} = (x_1, x_2, \dots, x_T)$$

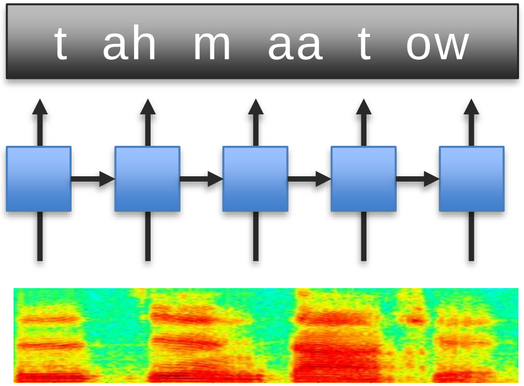
$\boldsymbol{z} \in \mathcal{Z}$ are phoneme transcripts

$$\boldsymbol{z} = (z_1, z_2, \dots, z_U)$$

Not the same length
$U \leq T$

defined over the space of labels L

Phonemes (z)



Spectogram (x)

Goal: train temporal classifier $h : \mathcal{X} \rightarrow \mathcal{Z}$

Loss: Negative log likelihood

$$L(S;\theta) = -\sum_{(\boldsymbol{x},\boldsymbol{z})\in S} \ln\big(p_\theta(\boldsymbol{z}|\boldsymbol{x})\big)$$

Connectionist Temporal Classification (CTC)

Rule-based alignment:

1) Remove all blanks
2) Remove repeated labels

$l = \{a\}$	$l = \{bee\}$
_aaa____	bbbeee_ee
___aaaa_	_bb_ee__e
_aaaaaaa	__bbbe_e_

Temporal alignment

③ Predicted labels $\boldsymbol{l}$

$$P(\boldsymbol{l}|\boldsymbol{x}) = \sum_{\boldsymbol{\pi}} \underbrace{P(\boldsymbol{l}|\boldsymbol{\pi})}_{\text{Temporal alignment}} P(\boldsymbol{\pi}|\boldsymbol{x})$$

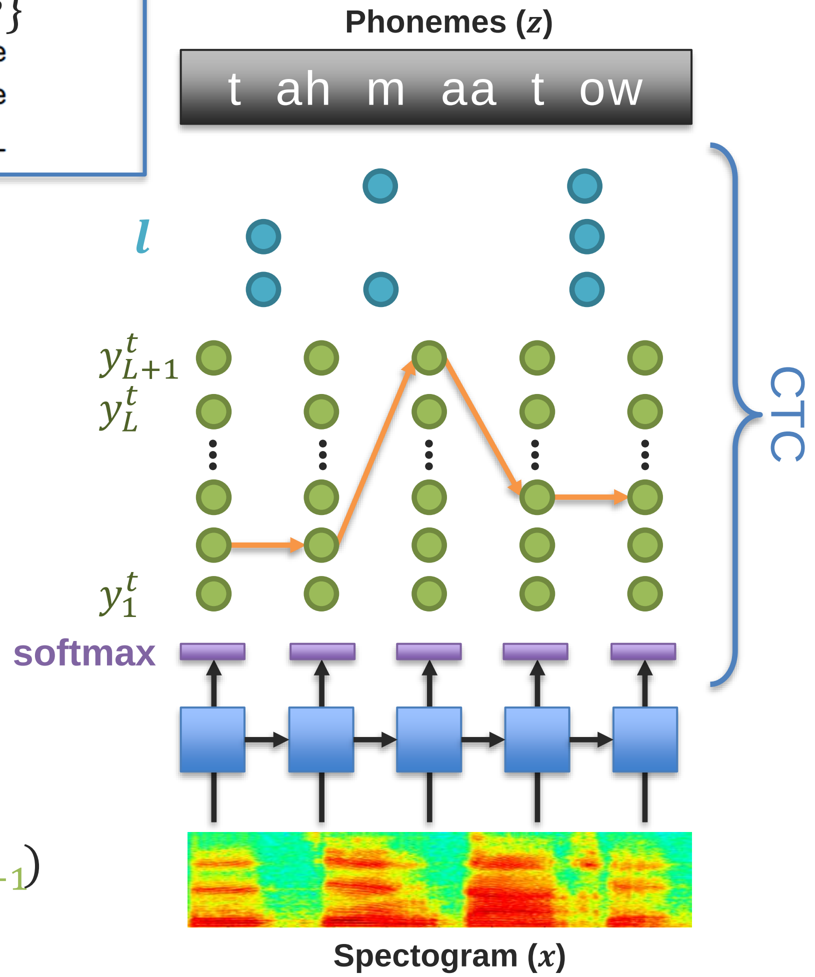
② Path $\boldsymbol{\pi}$ over the activations:

$$\mathrm{P}(\boldsymbol{\pi}|\boldsymbol{x}) = \prod_{t=1}^{T} y^t_{\boldsymbol{\pi}_t}, \forall \boldsymbol{\pi} \in L'^T$$

① Output activations (distribution):

$$\boldsymbol{y} = f_\theta(\boldsymbol{x}), \text{ where } \boldsymbol{y}^t = (y^t_1, y^t_2, \dots, y^t_L, y^t_{L+1})$$

for 'blank' or no label



Connectionist Temporal Classification (CTC)

4 Most probable sequence labels

$$\hat{z} = h(x) = \arg\max_{l \in L^T} P(\boldsymbol{l}|\boldsymbol{x})$$

3 Predicted labels $\boldsymbol{l}$

$$P(\boldsymbol{l}|\boldsymbol{x}) = \sum_{\boldsymbol{\pi}} P(\boldsymbol{l}|\boldsymbol{\pi})P(\boldsymbol{\pi}|\boldsymbol{x})$$

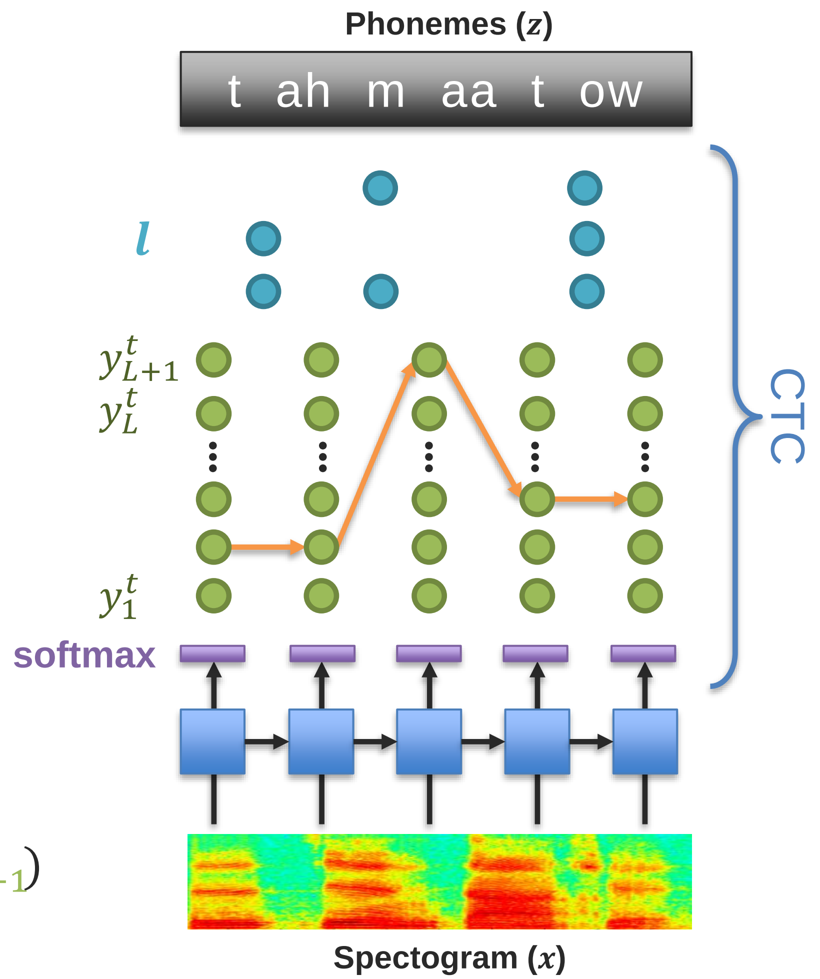
2 Path $\boldsymbol{\pi}$ over the activations:

$$\mathrm{P}(\boldsymbol{\pi}|\boldsymbol{x}) = \prod_{t=1}^{T} y_{\boldsymbol{\pi}_t}^{t}, \forall \boldsymbol{\pi} \in L'^T$$

1 Output activations (distribution):

$$\boldsymbol{y} = f_\theta(\boldsymbol{x}), \text{ where } \boldsymbol{y}^t = (y_1^t, y_2^t, \dots, y_L^t, y_{L+1}^t)$$

for 'blank' or no label



CTC Optimization

4 Most probable sequence labels

$$z^* = h(x) = \arg\max_{l \in L^T} P(\boldsymbol{l}|\boldsymbol{x})$$

Option 1: Select most probable path $\boldsymbol{\pi}$

$$\pi^* = \arg\max_{\pi} P(\boldsymbol{\pi}|\boldsymbol{x})$$

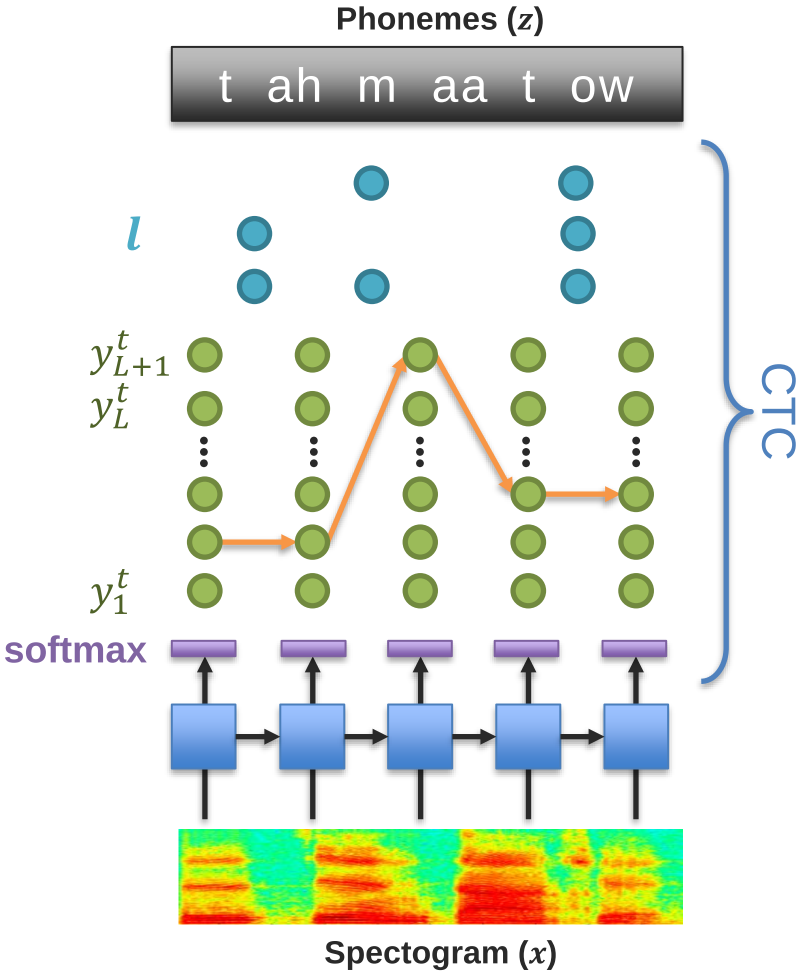
Get most probable labels z^* directly from π^*

Option 2: Solve using dynamic programming

Forward-backward algorithm

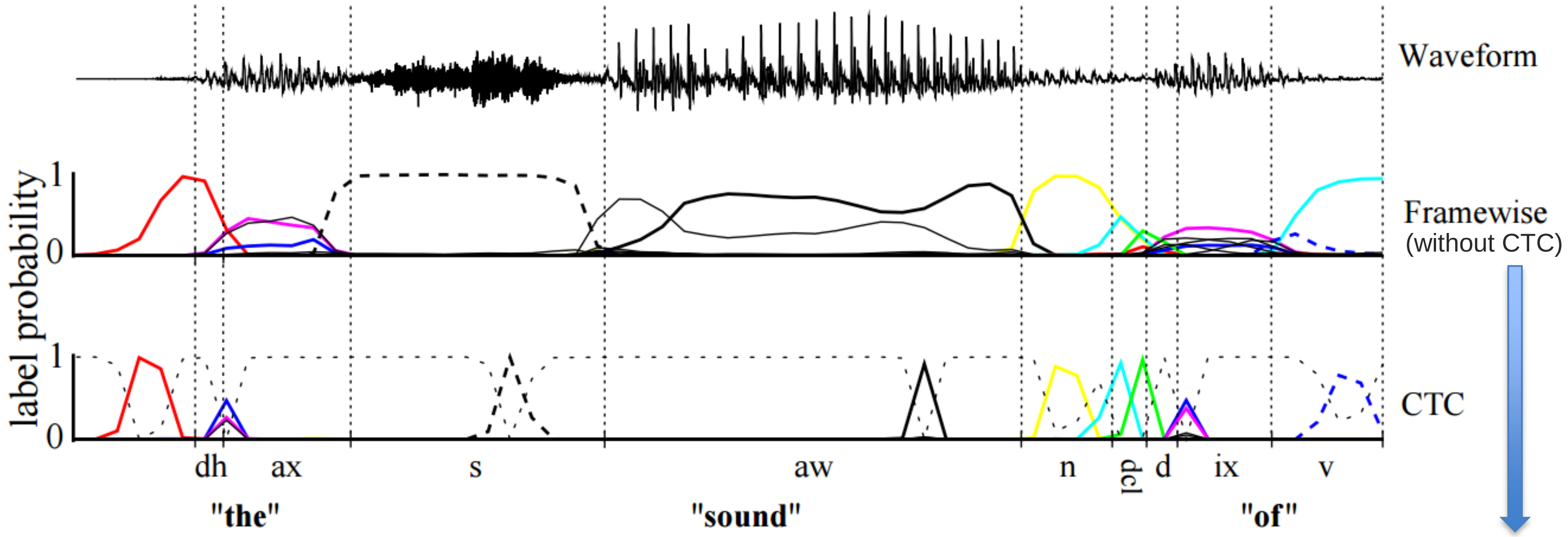
- Forward variables α
- Backward variables β

$$P(l|x) = \sum_{t=1}^{T} \sum_{s=1}^{|l|} \frac{\alpha_t(s)\beta_t(s)}{y^t_{l_s}}$$



Visualizing CTC Predictions

“Framewise” modeling: Learned using phoneme segmentation (vertical lines)



It gets penalized for mistakes around the boundaries

Why are CTC predictions so “peaky”?

CTC focuses on the phoneme transitions

Implicit alignment

Implicit alignment

We looked how to explicitly align temporal data

Could use that as an internal (hidden) step in our models?

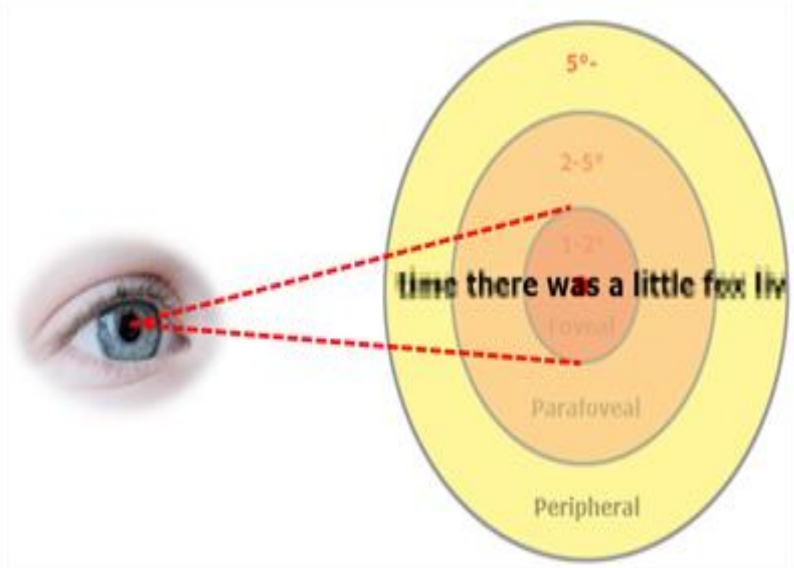
Can we instead encourage the model to align data when solving a different problem?

Yes!

Potential Solution and Inspiration: Human Attention

Foveal vision – we only see in “high resolution” in 2 degrees of vision

- We focus our attention selectively to certain words (for example our names)
- We attend to relevant speech in a noisy room



Implicit and "Uni-Directional" Alignment

Modality A
(query)

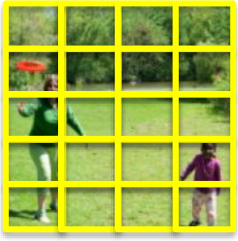
A woman is throwing a frisbee

Modality B
(key)

1 Hard attention

2 Warping

3 Soft attention
(discussed on Thursday)



Glimpse Network (Hard Attention)

Hard attention

Soft attention requires computing a representation for the whole image or sentence

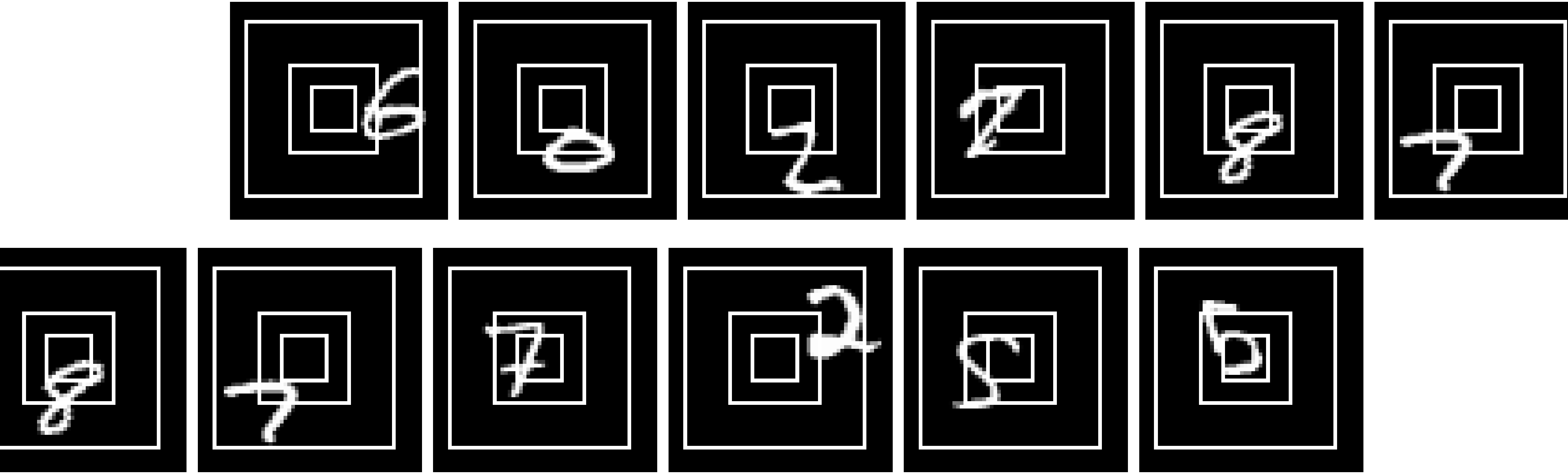
(more details during next lecture)

Hard attention on the other hand forces looking only at one part

- Main motivation was reduced computational cost rather than improved accuracy (although that happens a bit as well)
- **Saccade followed by a glimpse – how human visual system works**

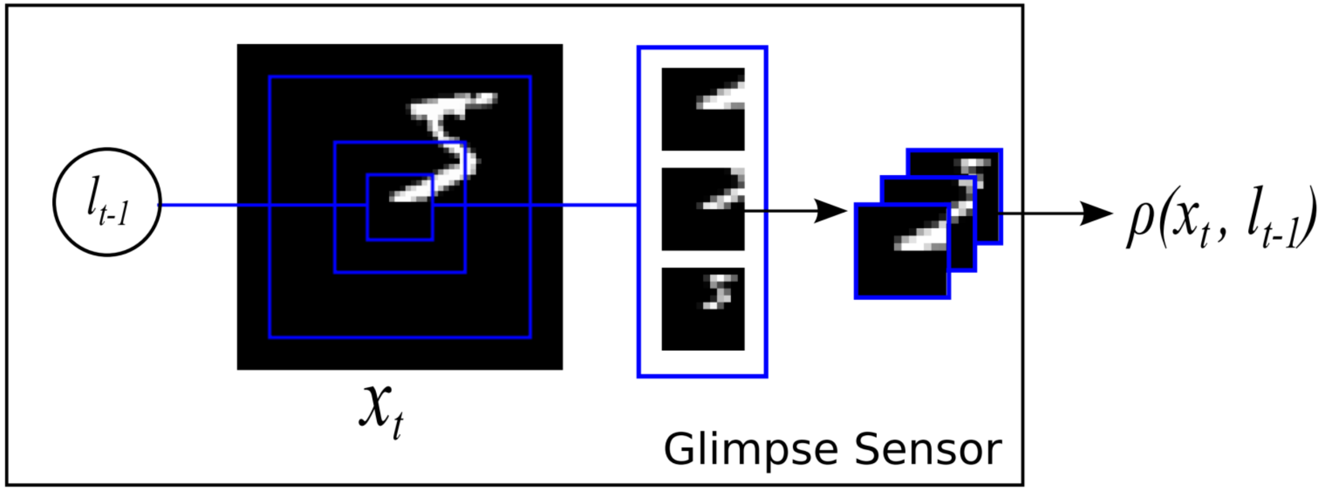
[Recurrent Models of Visual Attention, Mnih, 2014]
[Multiple Object Recognition with Visual Attention, Ba, 2015]

Hard attention examples



Glimpse Sensor

Looking at a part of an image at different scales

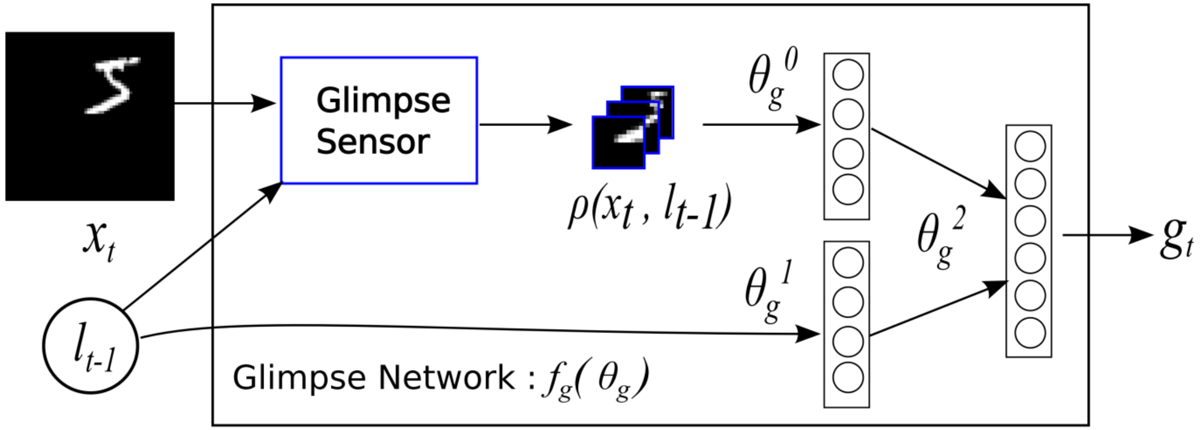


- At a number of different scales combined to a single multichannel image (human retina like representation)
- Given a location l_t output an image summary at that location

[Recurrent Models of Visual Attention, Mnih, 2014]

Glimpse network

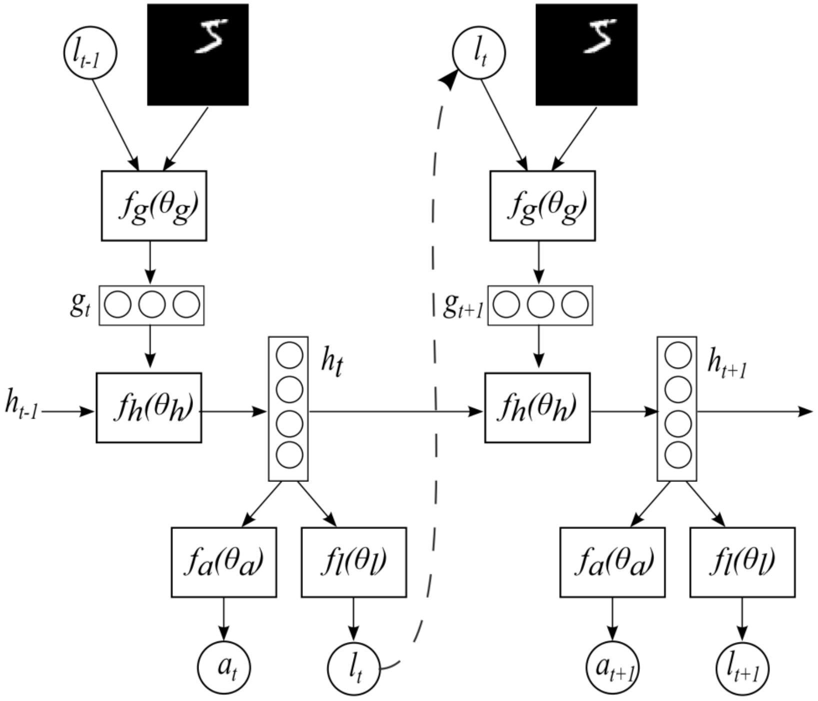
- Combining the Glimpse and the location of the glimpse into a joint network



- The glimpse is followed by a feedforward network (CNN or a DNN)
- The exact formulation of how the location and appearance are combined varies, the important thing is combining **what** and **where**
- Differentiable with respect to glimpse parameters but not the location

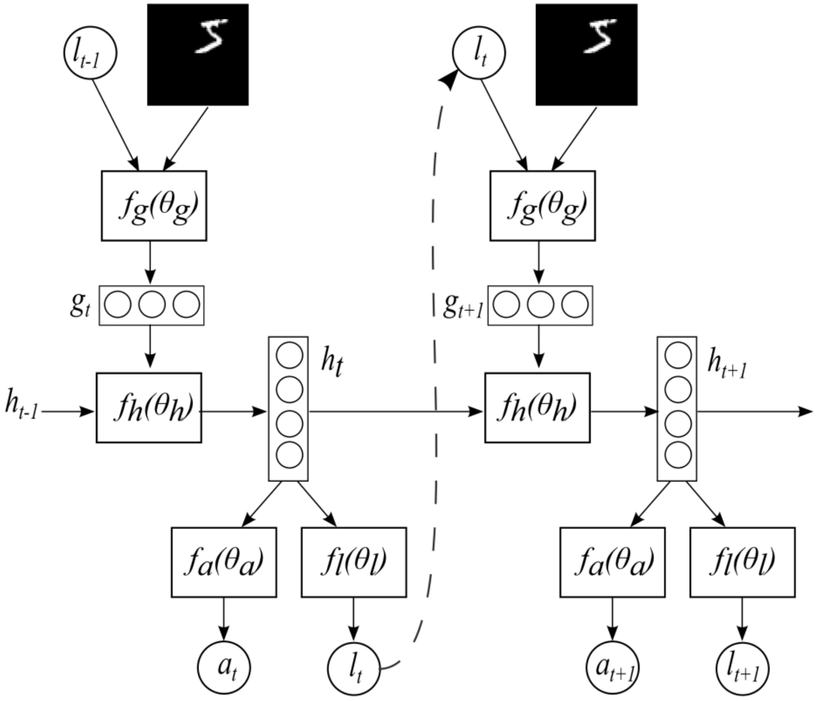
Overall Architecture - Emission network

- Given an image a glimpse location l_t, and optionally an action a_t
- Action can be:
  - Some action in a dynamic system – press a button etc.
  - Classification of an object
  - Word output
- This is an RNN with two output gates and a slightly more complex input gate!



Recurrent model of Visual Attention (RAM)

- Sample locations of glimpses leading to updates in the network
- Use gradient descent to update the weights (the glimpse network weights are differentiable)
- The emission network is an RNN
- Not as simple as backprop but doable
- Turns out this is very similar and in some cases equivalent to reinforcement learning using the REINFORCE learning rule [Williams, 1992]

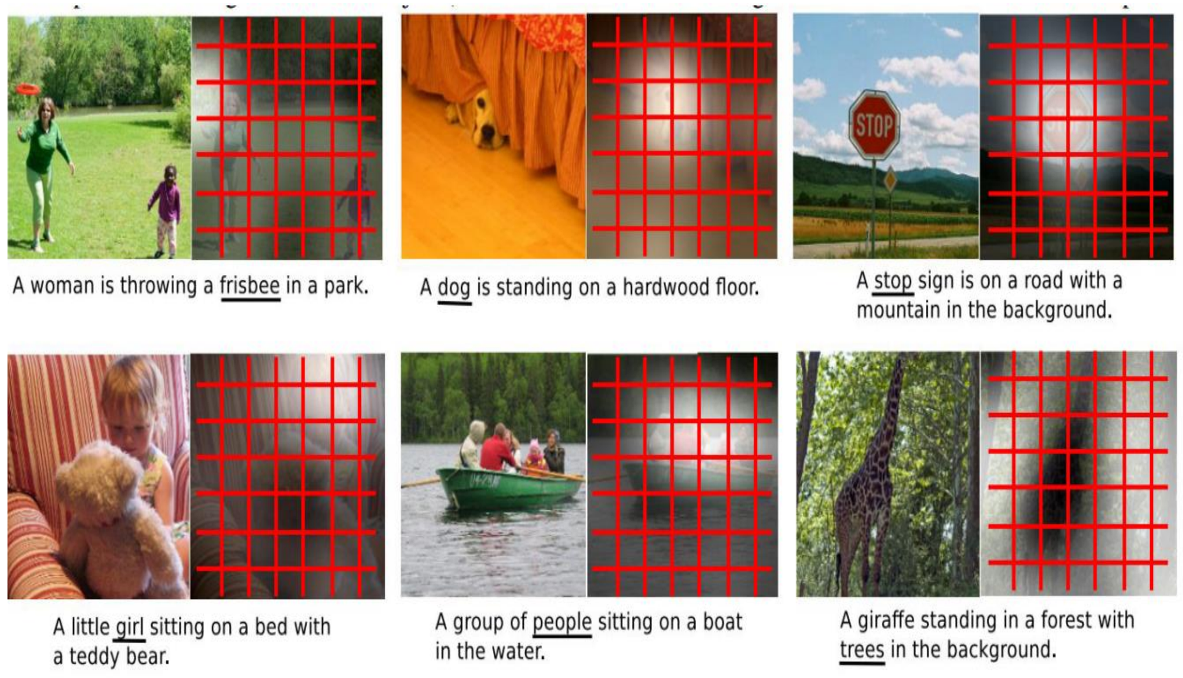


But not the “real” transformer!

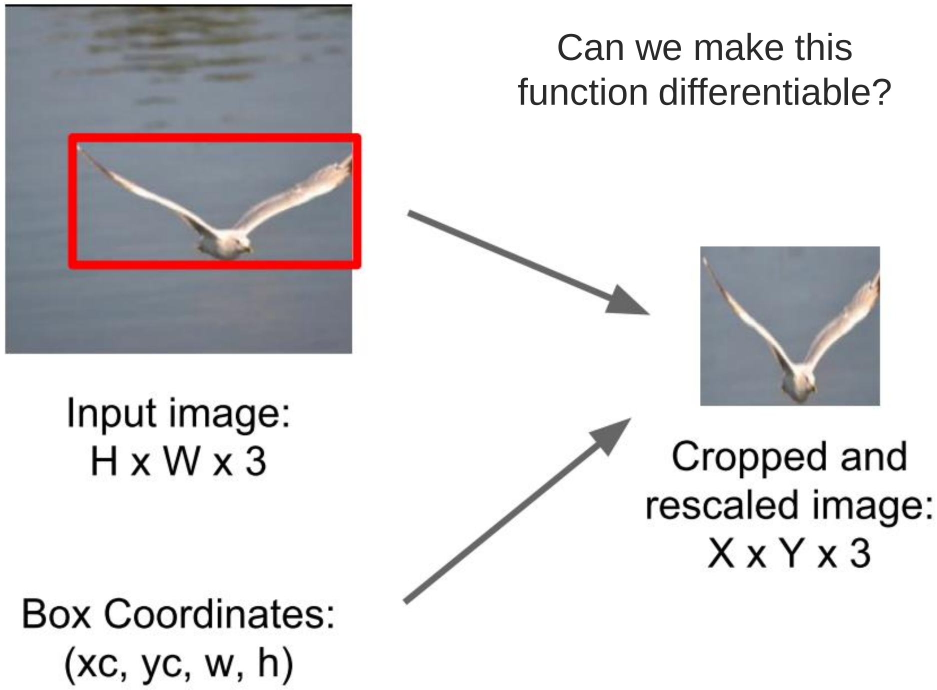
Spatial Transformer networks (Warping)

Some limitations of grid-based attention

Can we fixate on small parts of image but still have easy end-to-end training?



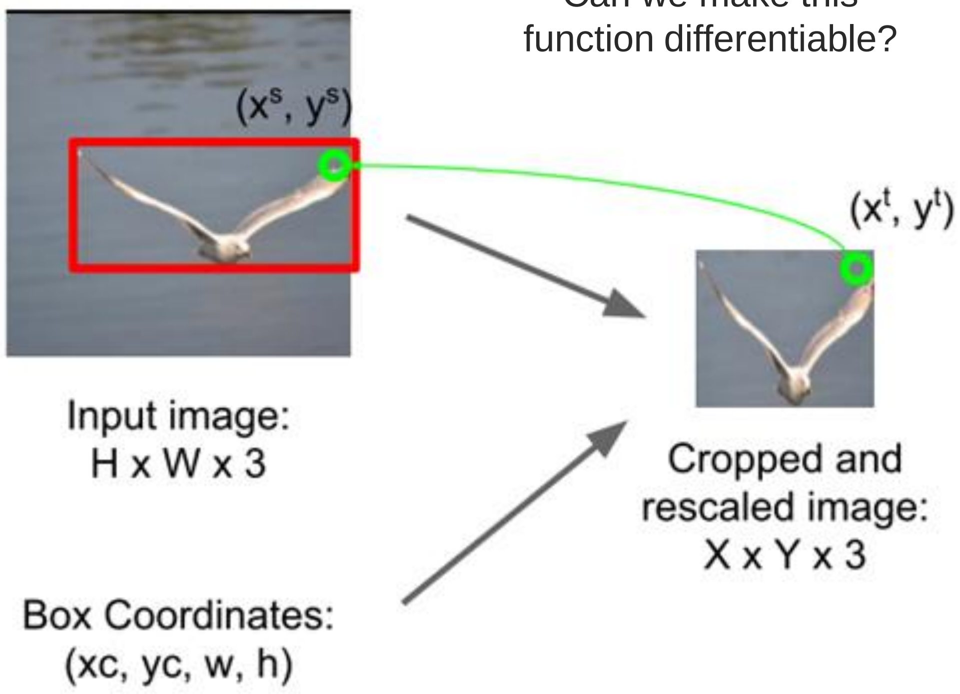
Spatial Transformer Networks



Spatial Transformer Networks

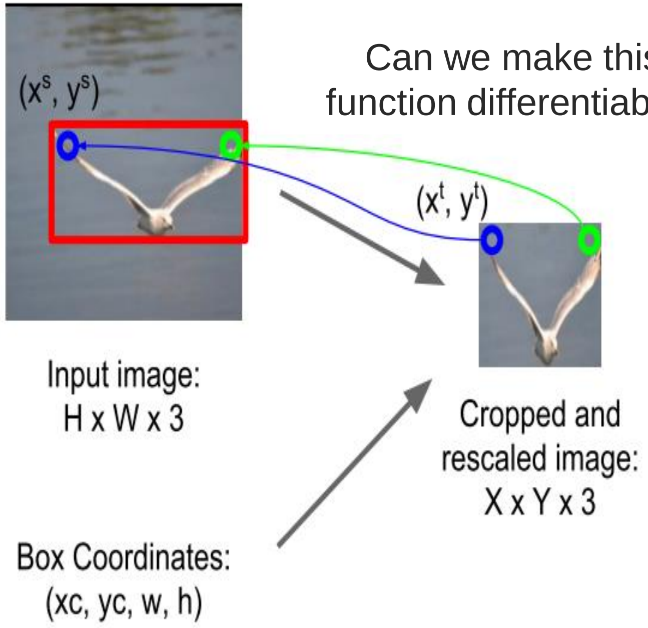
Idea: Function mapping pixel coordinates (x^t, y^t) of output to pixel coordinates (x^s, y^s) of input

Can we make this function differentiable?



$$\begin{pmatrix} x_i^s \\ y_i^s \end{pmatrix} = \begin{bmatrix} \theta_{1,1} & \theta_{1,2} & \theta_{1,3} \\ \theta_{2,1} & \theta_{2,2} & \theta_{2,3} \end{bmatrix} \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix}$$

Spatial Transformer Networks

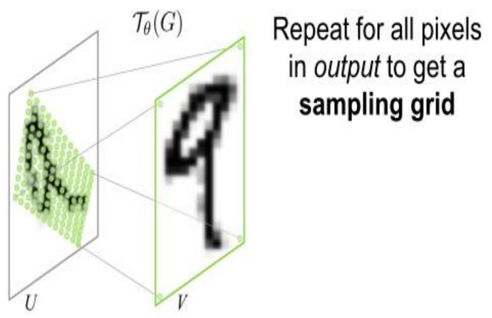


Can we make this function differentiable?

Idea: Function mapping pixel coordinates (x^t, y^t) of output to pixel coordinates (x^s, y^s) of input

$$\begin{pmatrix} x_i^s \\ y_i^s \end{pmatrix} = \begin{bmatrix} \theta_{1,1} & \theta_{1,2} & \theta_{1,3} \\ \theta_{2,1} & \theta_{2,2} & \theta_{2,3} \end{bmatrix} \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix}$$

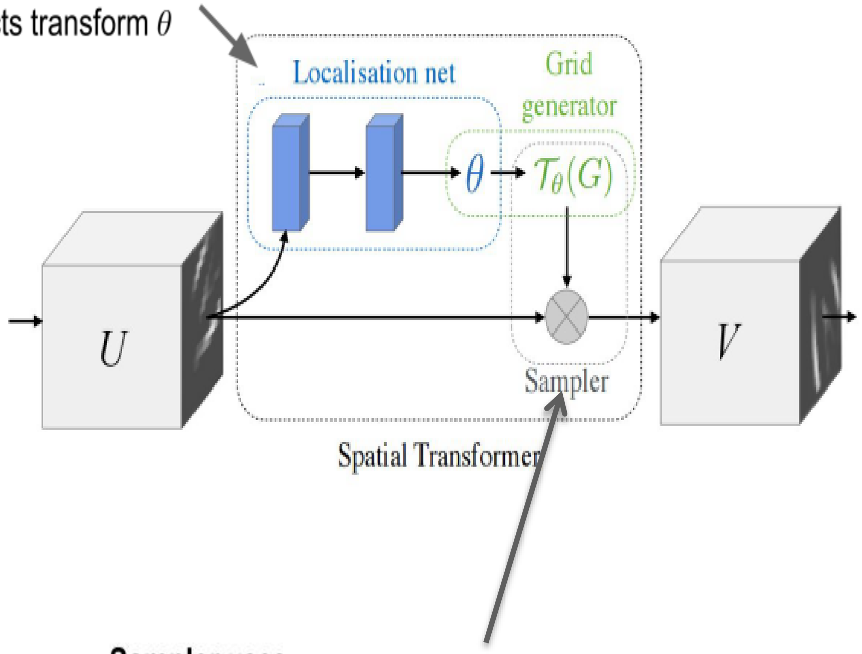
Network “attends” to input by predicting θ



Spatial Transformer Networks

A small **Localization network** predicts transform θ

Input: Full image



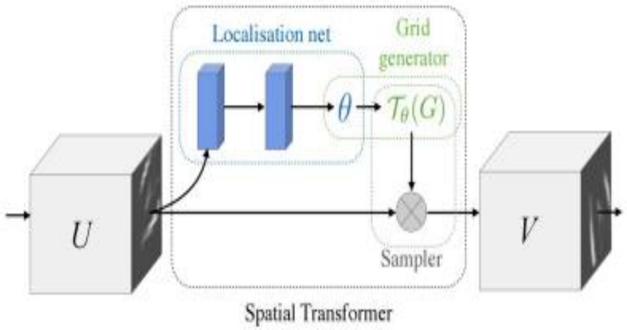
Output: Region of interest from input

Sampler uses bilinear interpolation to produce output

$$V_i^c = \sum_n^H \sum_m^W U_{nm}^c \max(0, 1 - |x_i^s - m|) \max(0, 1 - |y_i^s - n|)$$

Spatial Transformer Networks

Differentiable “attention / transformation” module



Insert spatial transformers into a classification network and it learns to attend and transform the input