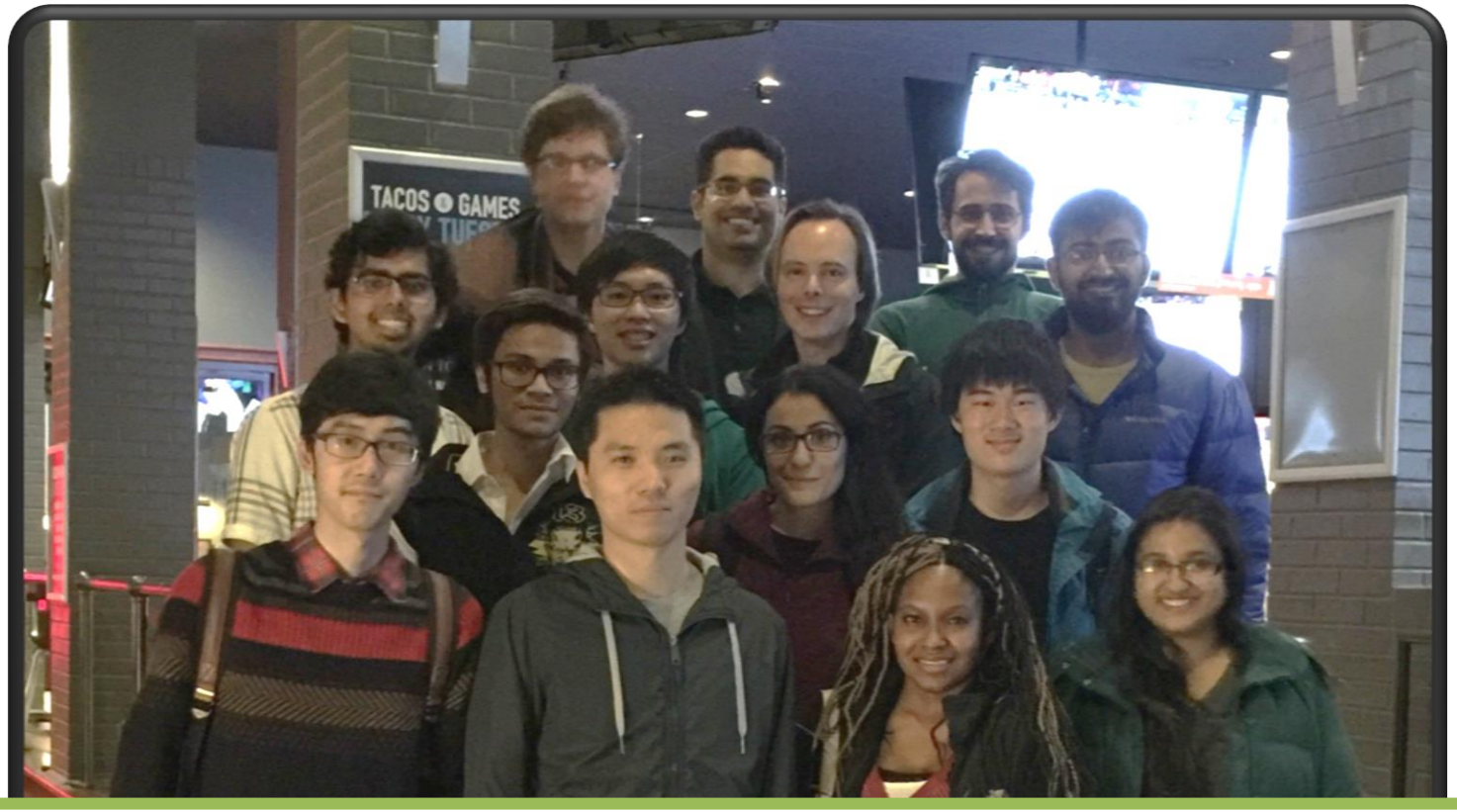
Language Technologies Institute

Carnegie Mellon University

Tutorial on Multimodal Machine Learning

Louis-Philippe Morency
Tadas Baltrusaitis

MultiComp Lab @ CMU



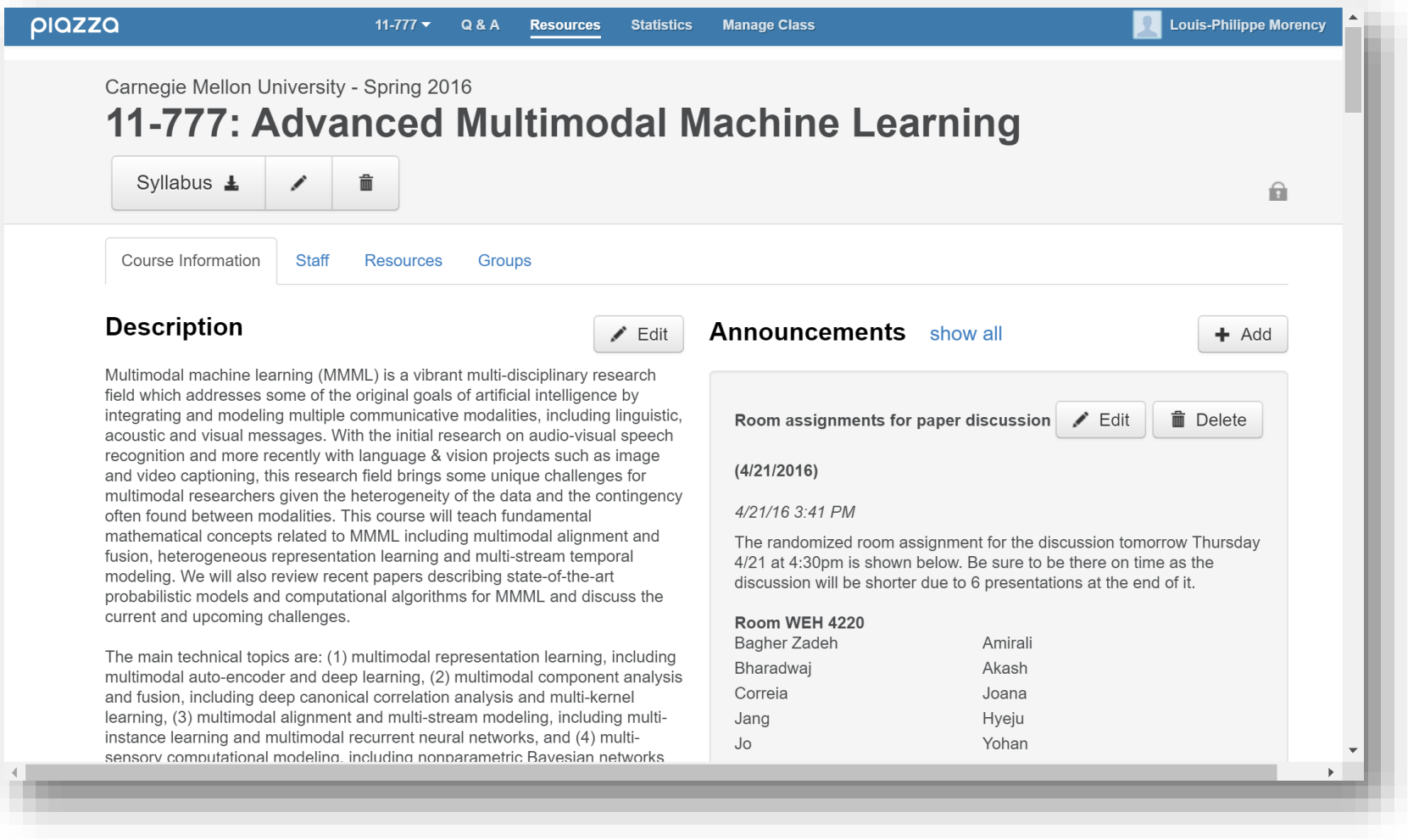
Algorithms to analyze, recognize and predict human subtle communicative behaviors in social context.

Your Instructors

Louis-Philippe Morency
morency@cs.cmu.edu

Tadas Baltrusaitis
tbaltrus@cs.cmu.edu

CMU Course 11-777: Multimodal Machine Learning



Tutorial Schedule

- **Introduction**
  - What is Multimodal?
    - Historical view, multimodal vs multimedia
  - Why multimodal
    - Multimodal applications: image captioning, video description, AVSR,…
  - Core technical challenges
    - Representation learning, translation, alignment, fusion and co-learning
- **Basic concepts – Part 1**
  - Linear models
    - Score and loss functions, regularization
  - Neural networks
    - Activation functions, multi-layer perceptron
  - Optimization
    - Stochastic gradient descent, backpropagation
- **Micro-break [5-10 minutes]**

Tutorial Schedule

- **Unimodal representations**
  - Language representations
    - Distributional hypothesis and word embedding
  - Visual representations
    - Convolutional neural networks
  - Acoustic representations
    - Spectrograms, autoencoders
- **Multimodal representations**
  - Joint representations
    - Visual semantic spaces, multimodal autoencoder
  - Orthogonal joint representations
    - Component analysis
  - Parallel multimodal representations
    - Similarity metrics, canonical correlation analysi
- **Coffee break [20 minutes]**

Tutorial Schedule

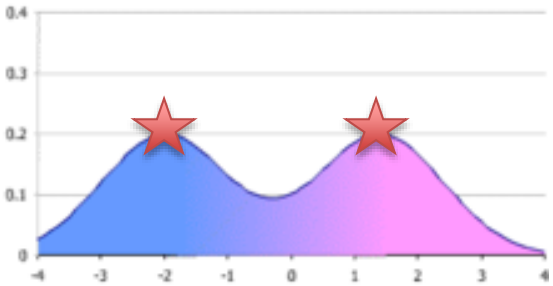
- **Basic concepts – Part 2**
  - Language models
    - Unigrams, bigrams, skip-grams, skip-thought
  - Unimodal sequence modeling
    - Recurrent neural networks, LSTMs
  - Optimization
    - Backpropagation through time
- **Multimodal translation and mapping**
  - Encoder-decoder models
    - Machine translation, image captioning
  - Generative vs retrieval approaches
    - Viseme generation, visual puppetry
- **Micro-break [5-10 minutes]**

Tutorial Schedule

- **Modality alignment**
  - Latent alignment approaches
    - Attention models, multi instance learning
  - Explicit alignment
    - Dynamic time warping
- **Multimodal fusion and co-learning**
  - Model free approaches
    - Early and late fusion, hybrid models
  - Kernel-based fusion
    - Multiple kernel learning
  - Multimodal graphical models
    - Factorial HMM, Multi-view Hidden CRF
- **Future direction discussion and concluding remarks**

What is Multimodal?

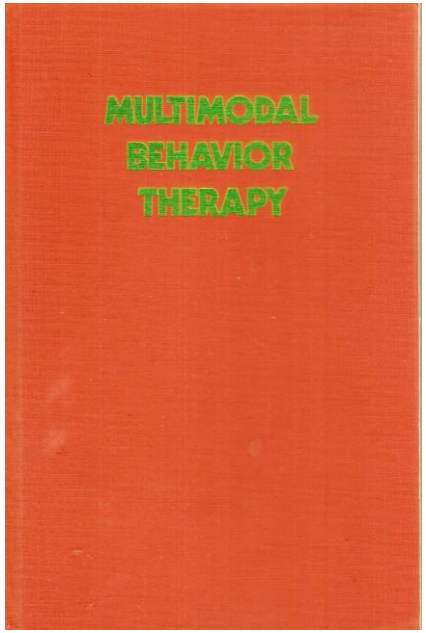
What is Multimodal?



Multimodal distribution

- Multiple modes, i.e., distinct “peaks” (local maxima) in the probability density function

What is Multimodal?



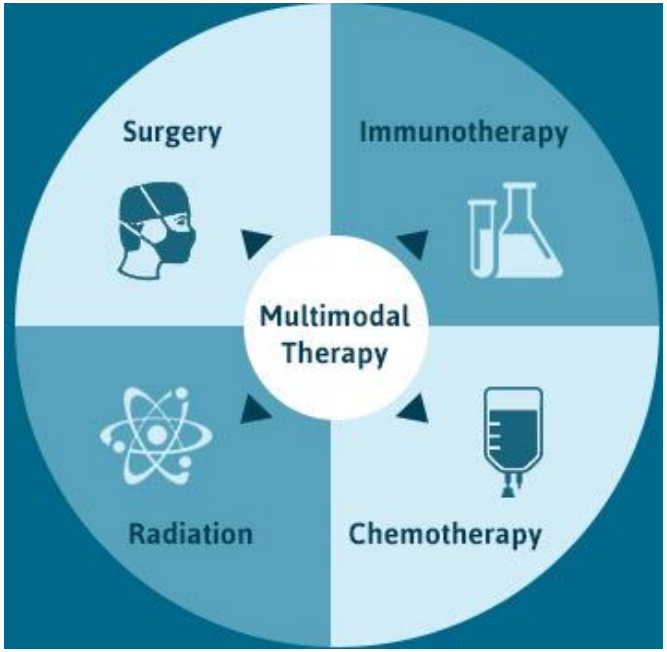
by Arnold Lazarus
[1973]

7 dimensions of personality (or *modalities*):

- Behavior,
- Affect,
- Sensation,
- Imagery,
- Cognition,
- Interpersonal relationships
- Drugs/biology

BASIC I.D.

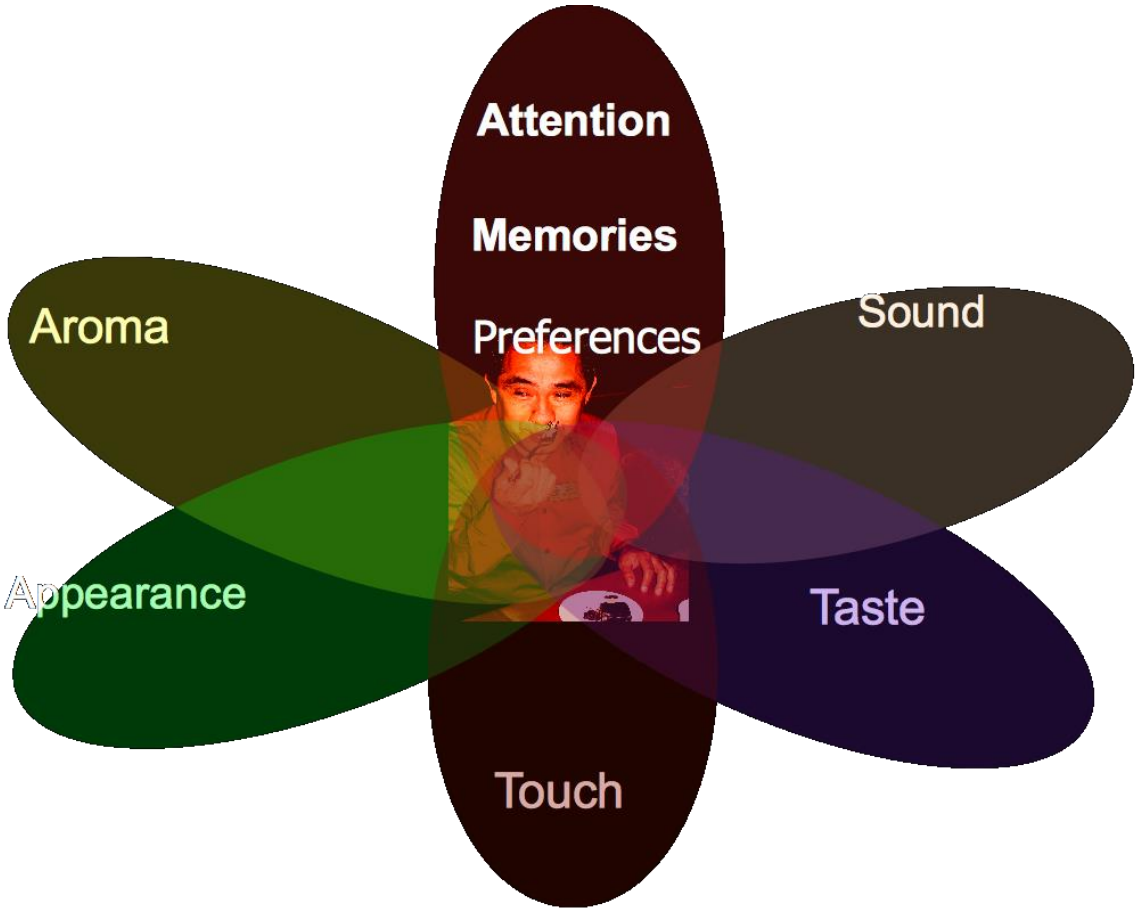
What is Multimodal?



Multimodal Treatment

"Therapy that combines more than one method of treatment. Also called combination therapy and multimodality therapy"

What is Multimodal?



Sensory Modalities

What is Multimodal?

Modality

The way in which something happens or is experienced.

- *Modality* refers to a certain type of information and/or the representation format in which information is stored.
- *Sensory modality:* one of the primary forms of sensation, as vision or touch; channel of communication.

Medium ("middle")

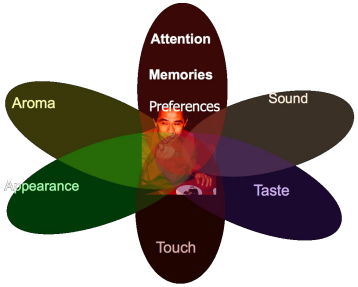
A means or instrumentality for storing or communicating information; system of communication/transmission.

- *Medium* is the means whereby this information is delivered to the senses of the interpreter.

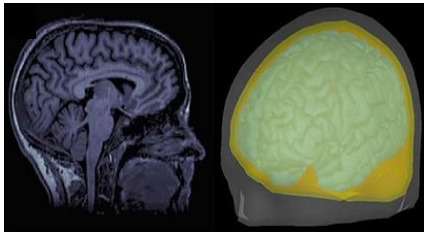
Examples of Modalities

- Natural language (both spoken or written)
- Visual (from images or videos)
- Auditory (including voice, sounds and music)
- Haptics / touch
- Smell, taste and self-motion
- Physiological signals
  - Electrocardiogram (ECG), skin conductance
- Other modalities
  - Infrared images, depth images, fMRI

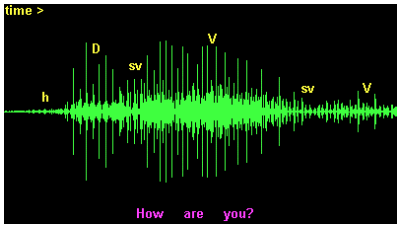
Multiple Communities and Modalities



Psychology

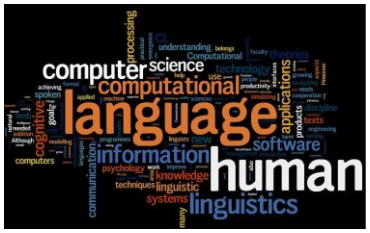


Medical



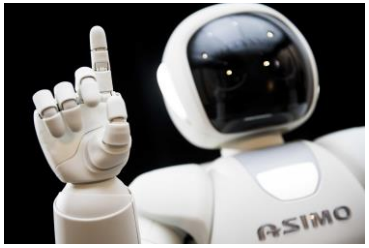
Speech

Vision

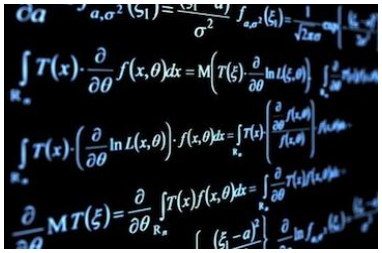


Language

Multimedia



Robotics



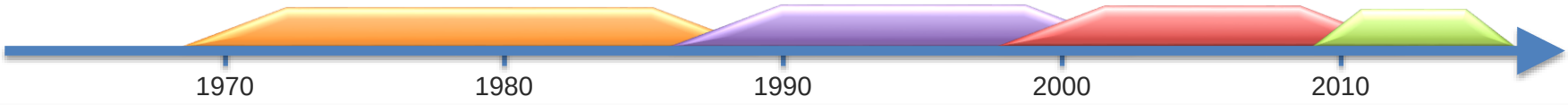
Learning

A Historical View

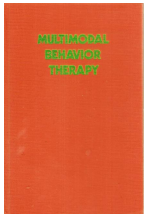
Prior Research on "Multimodal"

Four eras of multimodal research

- The "behavioral" era (1970s until late 1980s)
- The "computational" era (late 1980s until 2000)
- The "interaction" era (2000 - 2010)
- The "deep learning" era (2010s until …)
  - Main focus of this tutorial



The "Behavioral" Era (1970s until late 1980s)



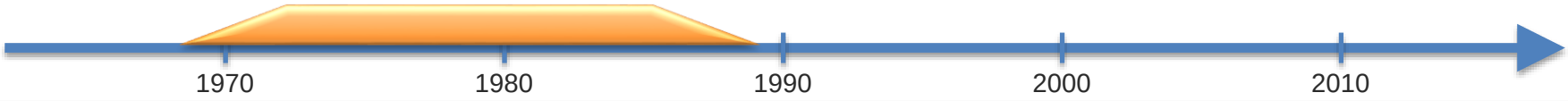
Multimodal Behavior Therapy by Arnold Lazarus [1973]

- 7 dimensions of personality (or *modalities*)

Multi-sensory integration (in psychology):

- Multimodal signal detection: Independent decisions vs. integration [1980]
- Infants' perception of substance and temporal synchrony in multimodal events [1983]
- A multimodal assessment of behavioral and cognitive deficits in abused and neglected preschoolers [1984]

❑ TRIVIA: Geoffrey Hinton received his B.A. in Psychology ☺

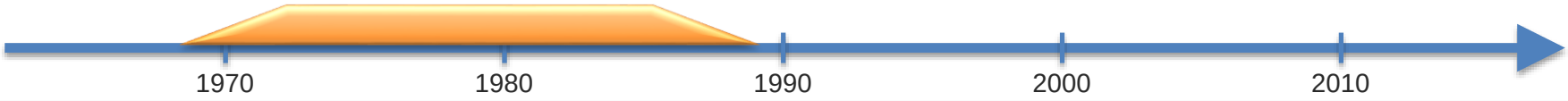


Language and Gestures

David McNeill
University of Chicago
Center for Gesture and Speech Research

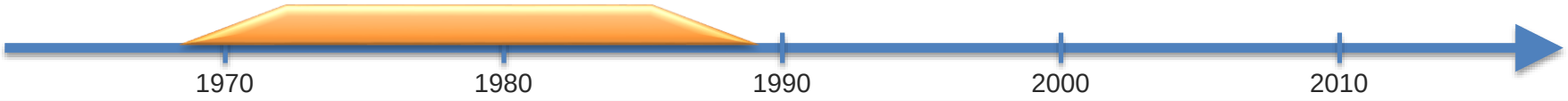
"For McNeill, gestures are in effect the speaker's thought in action, and integral components of speech, not merely accompaniments or additions."

- TRIVIA: Justine Cassell was a student of David McNeill



The McGurk Effect (1976)

Hearing lips and seeing voices – Nature



➢ The "Computational" Era(Late 1980s until 2000)

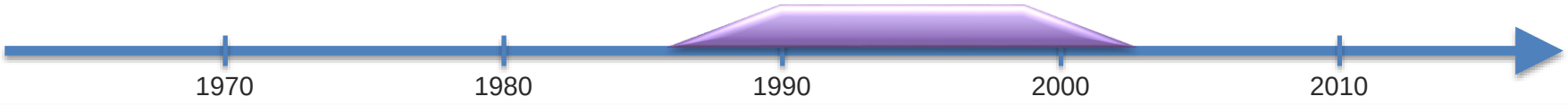
1) Audio-Visual Speech Recognition (AVSR)

- Motivated by the McGurk effect
- First AVSR System in 1986

  "Automatic lipreading to enhance speech recognition"
- Good survey paper [2002]

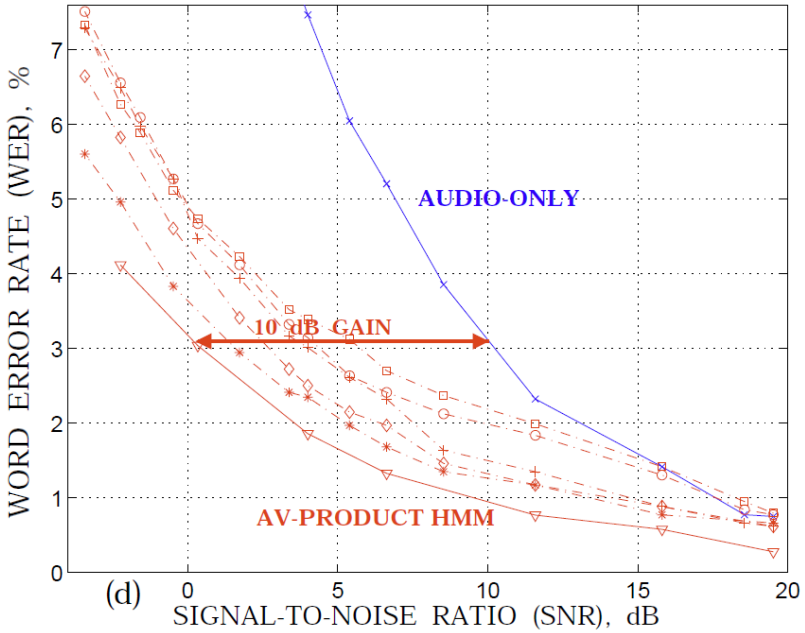
  "Recent Advances in the Automatic Recognition of Audio-Visual Speech"

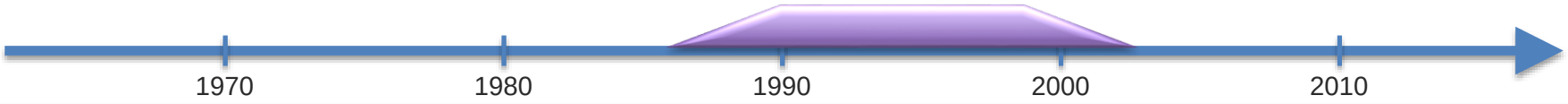
❑ TRIVIA: The first multimodal deep learning paper was about audio-visual speech recognition [ICML 2011]



➢ The "Computational" Era(Late 1980s until 2000)

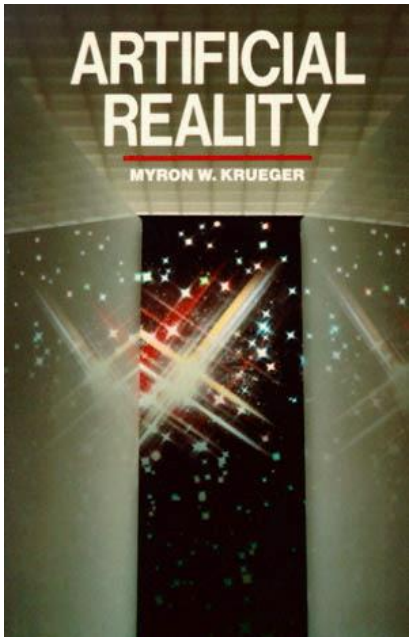
1) Audio-Visual Speech Recognition (AVSR)





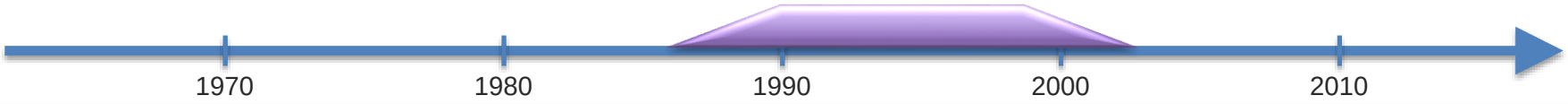
➢ The "Computational" Era (Late 1980s until 2000)

2) Multimodal/multisensory interfaces



Artificial reality describes Myron Krueger's interactive immersive environments, based on video recognition techniques, that put a user in full, unencumbered contact with the digital world.

- Started his work in 1960s
- Book published in 1983

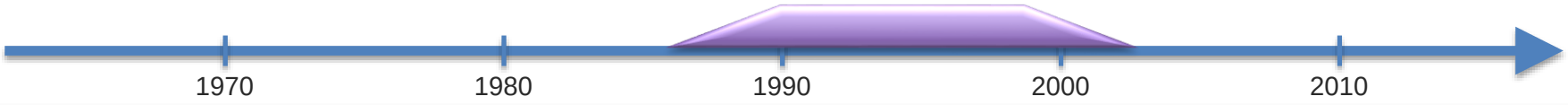


➢ The "Computational" Era (Late 1980s until 2000)

2) Multimodal/multisensory interfaces

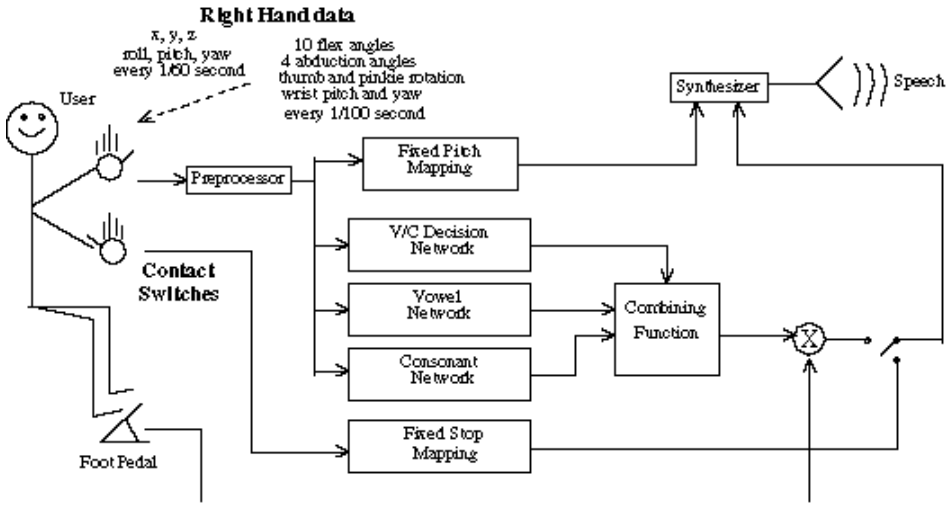
- Multimodal Human-Computer Interaction (HCI)

"Study of how to design and evaluate new computer systems where human interact through multiple modalities, including both input and output modalities."

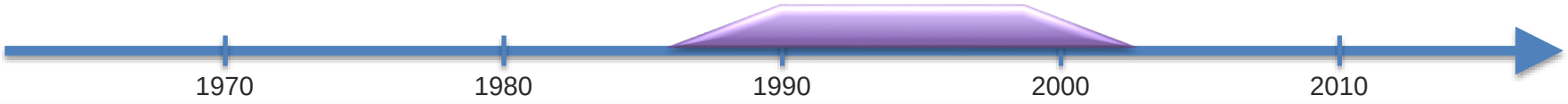


➢ The "Computational" Era (Late 1980s until 2000)

2) Multimodal/multisensory interfaces



[Glove-talk: A neural network interface between a data-glove and a speech synthesizer](#) *By Sidney Fels & Geoffrey Hinton [CHI'95]*



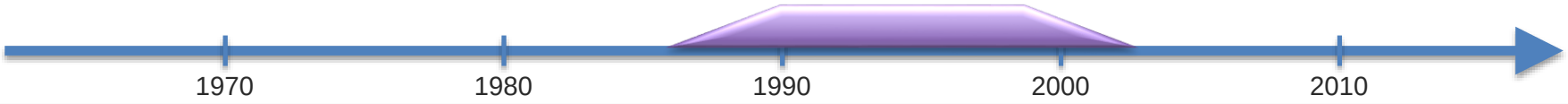
➢ The "Computational" Era (Late 1980s until 2000)

2) Multimodal/multisensory interfaces

pFinder: Real-time Tracking of human body

by C. Wren, A. Azarbayejani, T. Darrell and A. Pentland [1995]

- TRIVIA: Most cited paper by Trevor Darrell



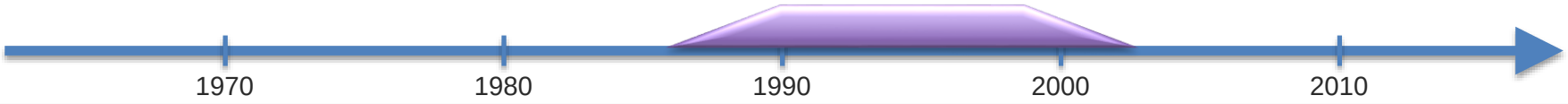
➢ The "Computational" Era (Late 1980s until 2000)

2) Multimodal/multisensory interfaces

Rosalind Picard

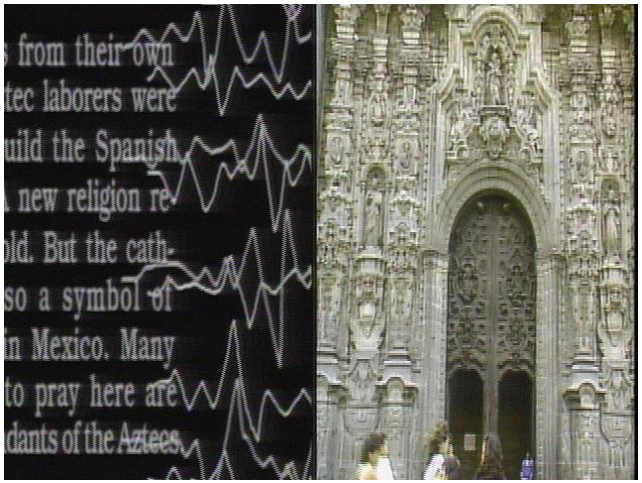
Affective Computing *is computing that relates to, arises from, or deliberately influences emotion or other affective phenomena.*

- TRIVIA: Rosalind Picard came from the same group (MIT, Sandy Pentland)



➢ The "Computational" Era (Late 1980s until 2000)

3) Multimedia Computing

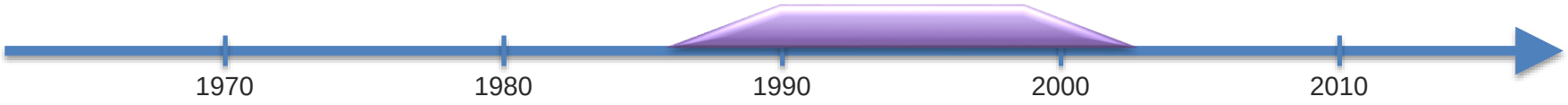


Carnegie Mellon University

informedia
digital video understanding

[1994-2010]

"The Informedia Digital Video Library Project automatically combines speech, image and natural language understanding to create a full-content searchable digital video library."

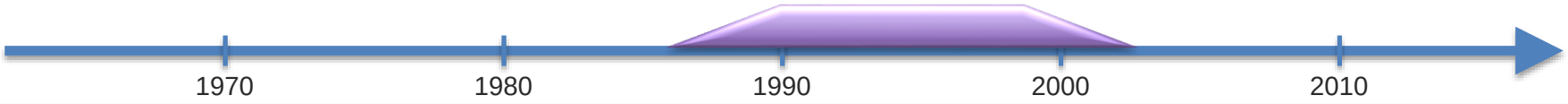


➢ The "Computational" Era (Late 1980s until 2000)

3) Multimedia Computing

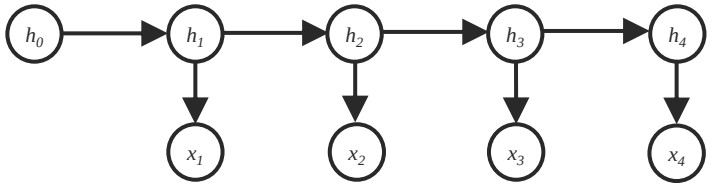
Multimedia content analysis

- **Shot-boundary detection (1991 -)**
  - Parsing a video into continuous camera shots
- **Still and dynamic video abstracts (1992 -)**
  - Making video browsable via representative frames (keyframes)
  - Generating short clips carrying the essence of the video content
- **High-level parsing (1997 -)**
  - Parsing a video into semantically meaningful segments
- **Automatic annotation (indexing) (1999 -)**
  - Detecting prespecified events/scenes/objects in video

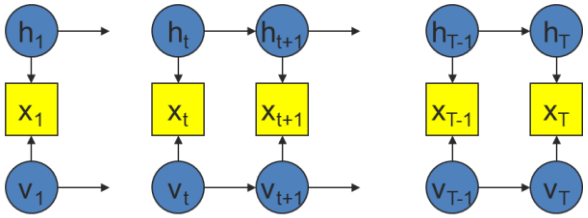


Multimodal Computation Models

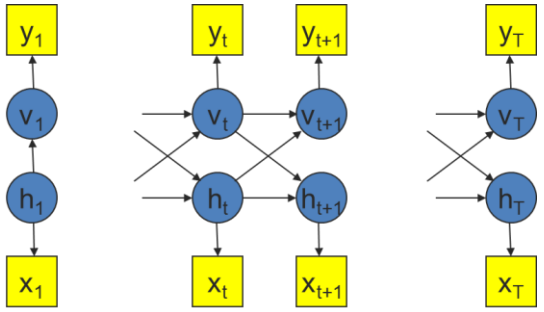
- Hidden Markov Models [1960s]

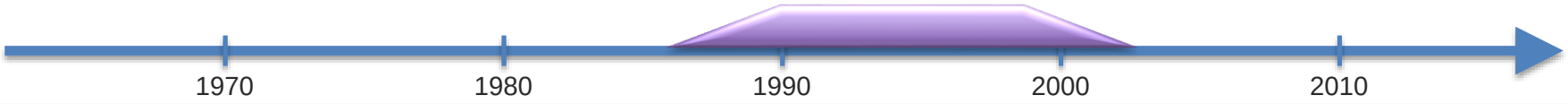


- ❑ Factorial Hidden Markov Models [1996]



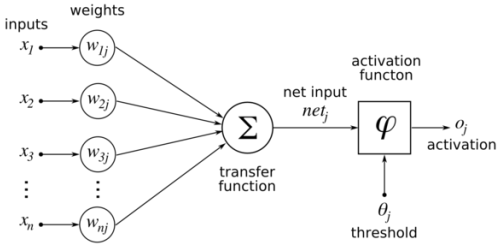
- ❑ Coupled Hidden Markov Models [1997]



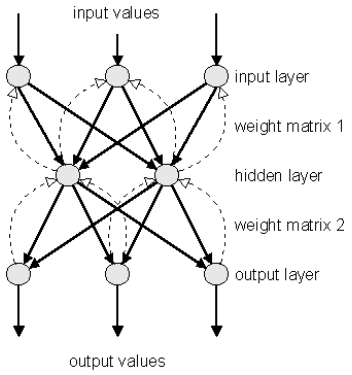


Multimodal Computation Models

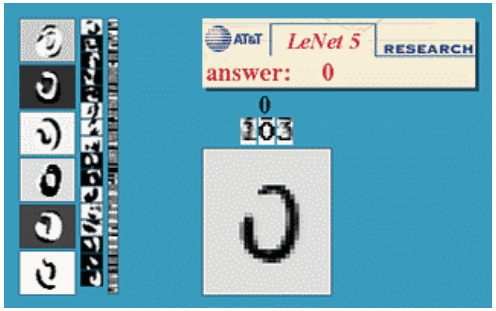
- Artificial Neural Networks [1940s]

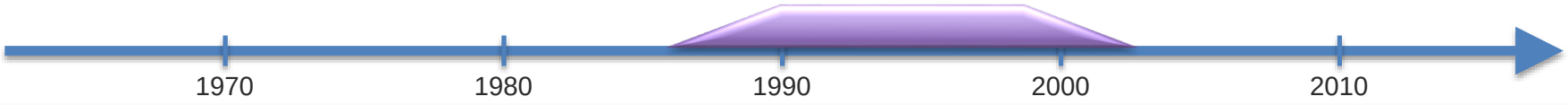


- Backpropagation [1975]



- Convolutional neural networks [1980s]



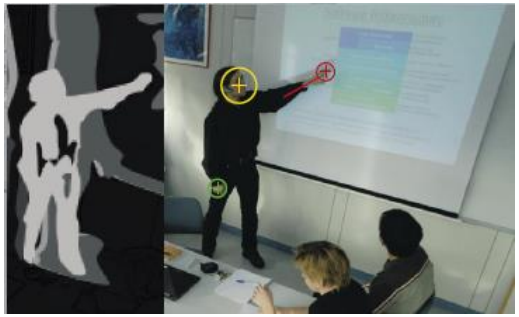


➢ The "Interaction" Era (2000s)

1) Modeling Human Multimodal Interaction

AMI Project [2001-2006, IDIAP]

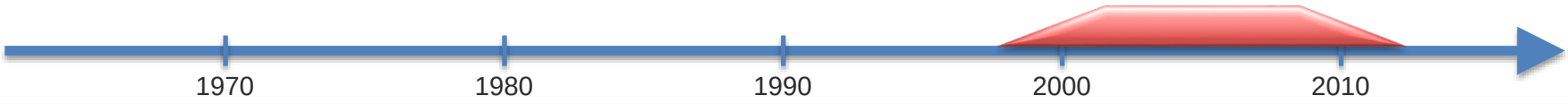
- 100+ hours of meeting recordings
- Fully synchronized audio-video
- Transcribed and annotated



CHIL Project [Alex Waibel]

- Computers in the Human Interaction Loop
- Multi-sensor multimodal processing
- Face-to-face interactions

❑ TRIVIA: Samy Bengio started at IDIAP working on AMI project



➢ The "Interaction" Era (2000s)

1) Modeling Human Multimodal Interaction

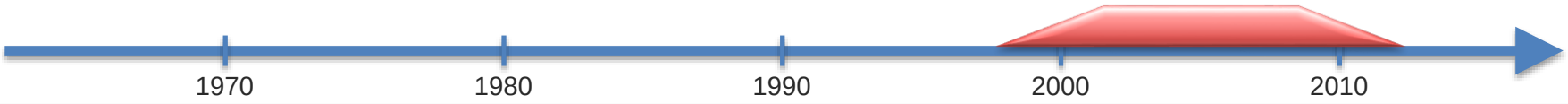
CALO Project [2003-2008, SRI]

- Cognitive Assistant that Learns and Organizes
- Personalized Assistant that Learns (PAL)
- Siri was a spinoff from this project

SSP Project [2008-2011, IDIAP]

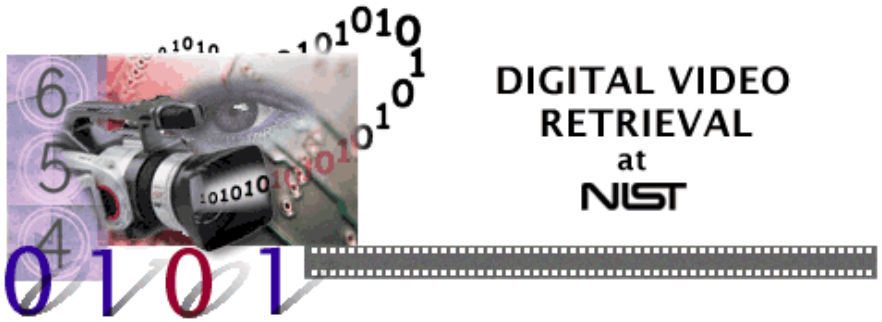
- Social Signal Processing
- First coined by Sandy Pentland in 2007
- Great dataset repository: http://sspnet.eu/

❑ TRIVIA: LP's PhD research was partially funded by CALO ☺



➢ The "Interaction" Era (2000s)

2) Multimedia Information Retrieval

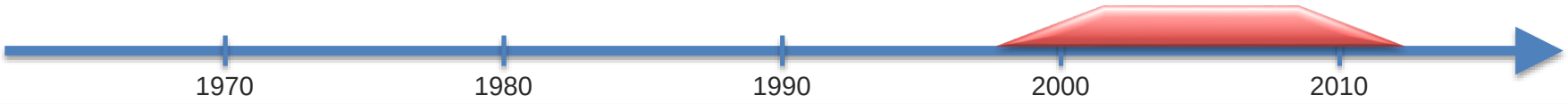


"Yearly competition to promote progress in content-based retrieval from digital video via open, metrics-based evaluation"

[Hosted by NIST, 2001-2016]

Research tasks and challenges:

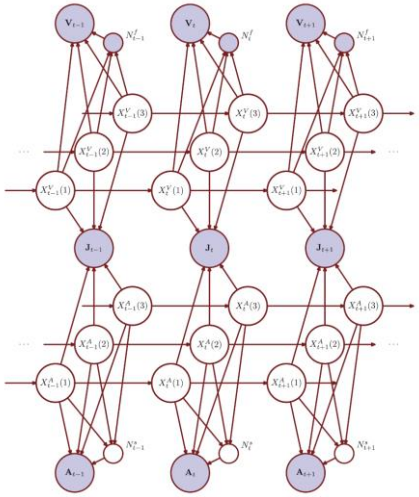
- Shot boundary, story segmentation, search
- "High-level feature extraction": semantic event detection
- Introduced in 2008: copy detection and surveillance events
- Introduced in 2010: Multimedia event detection (MED)

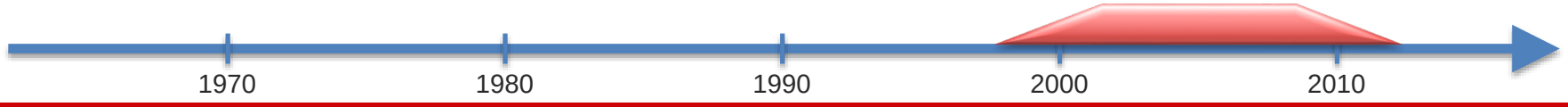


Multimodal Computational Models

- Dynamic Bayesian Networks
  - Kevin Murphy's PhD thesis and Matlab toolbox
  - Asynchronous HMM for multimodal [Samy Bengio, 2007]

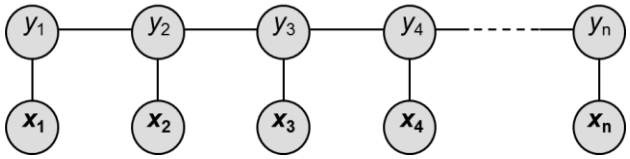
Audio-visual speech segmentation



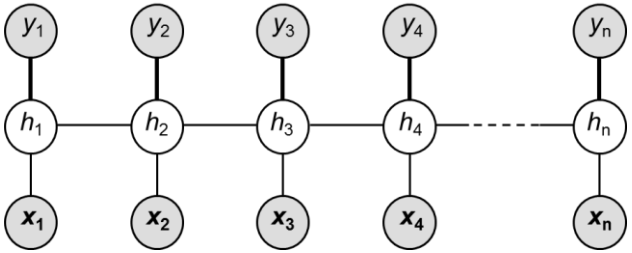


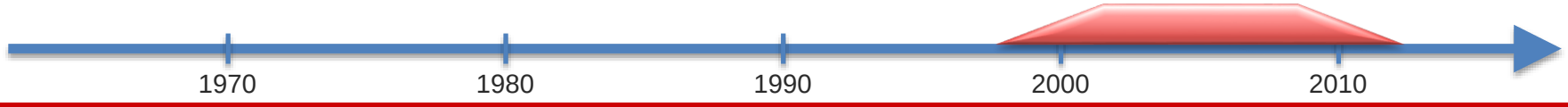
Multimodal Computational Models

- Discriminative sequential models
  - Conditional random fields [Lafferty et al., 2001]



  - Latent-dynamic CRF [Morency et al., 2007]





➢ The "deep learning" era (2010s until …)

Representation learning (a.k.a. deep learning)

- Multimodal deep learning [ICML 2011]
- Multimodal Learning with Deep Boltzmann Machines [NIPS 2012]
- Visual attention: Show, Attend and Tell: Neural Image Caption Generation with Visual Attention [ICML 2015]

Key enablers for multimodal research:

- New large-scale multimodal datasets
- Faster computer and GPUS
- High-level visual features
- "Dimensional" linguistic features

Our tutorial focuses on this era!

➢ The "deep learning" era (2010s until …)

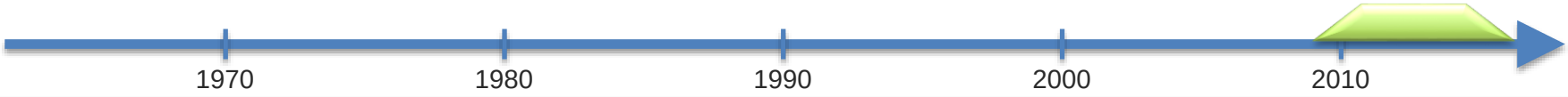
Many new challenges and multimodal corpora !!

Audio-Visual Emotion Challenge (AVEC, 2011-)

- Emotional dimension estimation
- Standardized training and test sets
- Based on the SEMAINE dataset

Emotion Recognition in the Wild Challenge (EmotiW 2013-)

- Emotional dimension estimation
- Standardized training and test sets
- Based on the SEMAINE dataset



➢ The "deep learning" era (2010s until …)

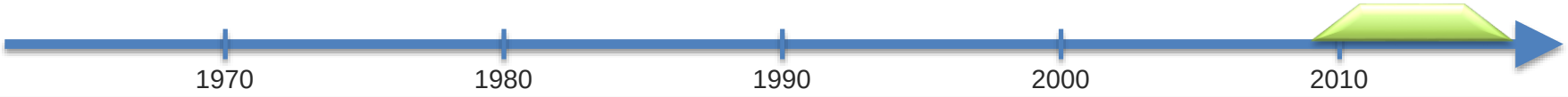
Renew of multimedia content analysis !

- Image captioning

The man at bat readies to swing at the pitch while the umpire looks on.

A large bus sitting next to a very tall building.

- Video description
- Visual Question-Answer



Real world tasks tackled by MMML

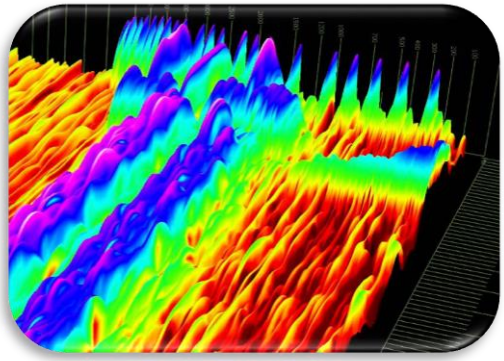
- Affect recognition
  - Emotion
  - Persuasion
  - Personality traits
- Media description
  - Image captioning
  - Video captioning
  - Visual Question Answering
- Event recognition
  - Action recognition
  - Segmentation
- Multimedia information retrieval
  - Content based/Cross-media

Core Technical Challenges

Multimodal Machine Learning

Verbal

Vocal



Visual

Core Technical Challenges:

Representation Translation Alignment

Fusion Co-Learning

These challenges are non-exclusive.

Core Challenge 1 - Representation

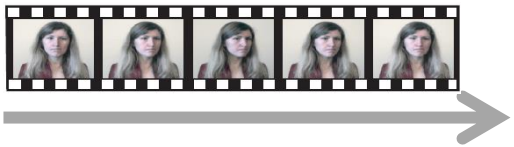
Heterogonous data:

- **Verbal modality**

We saw the yellow dog

- **Vocal modality**

- **Visual modality**



Representation:

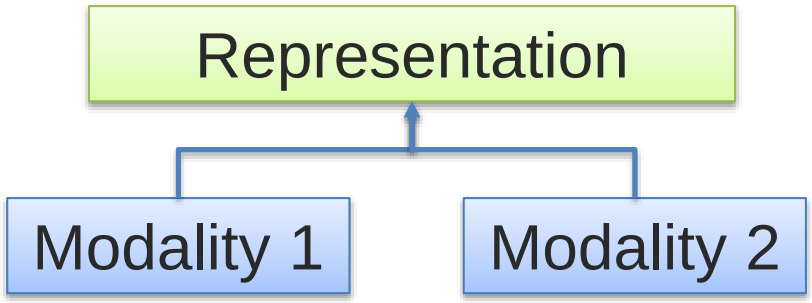
"Computer interpretable description of the multimodal data (e.g., vector, tensor)"

Challenges:

I. Symbols and signals
II. Different granularities
III. Static and sequential
IV. Different noise distribution
V. Unbalanced proportions

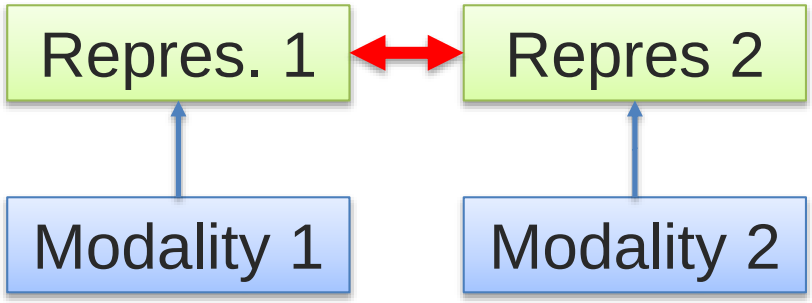
Core Challenge 1 - Representation

A Joint representations:



- Simplest version: modality concatenation (early fusion)
- Can be learned supervised or unsupervised
- Multimodal factor analysis

B Coordinated representations:



- Similarity-based methods (e.g., cosine distance)
- Orthogonality constraints (e.g., canonical correlation)

Core Challenge 2 – Translation / Mapping

Visual gestures
(both speaker and listener gestures)

Transcriptions
+
Audio streams

Marsella et al., Virtual character performance from speech, SIGGRAPH/Eurographics Symposium on Computer Animation, 2013

Core Challenge 2 – Translation / Mapping

- **Visual animations**

- **Image captioning**

- **Speech synthesis**

Translation / mapping:

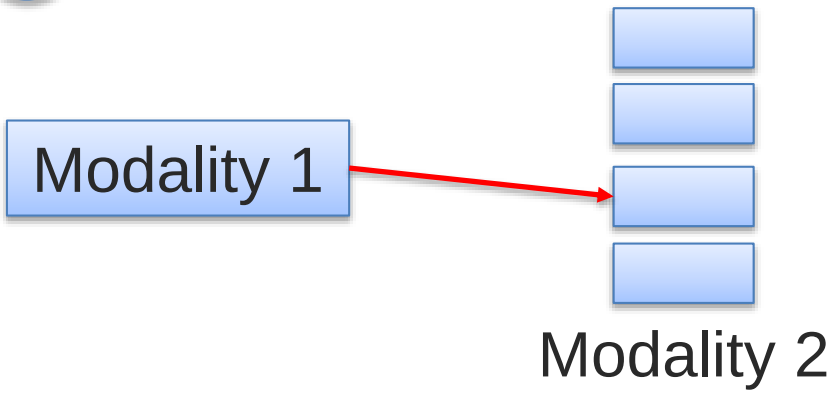
"Process of changing data from one modality to another"

Challenges:

I. Different representations
II. Multiple source modalities
III. Open ended translations
IV. Subjective evaluation
V. Repetitive processes

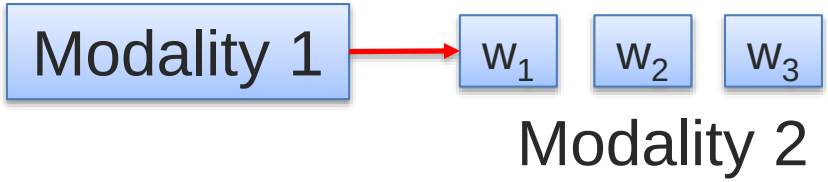
Core Challenge 2 – Translation / Mapping

A Bounded translations:



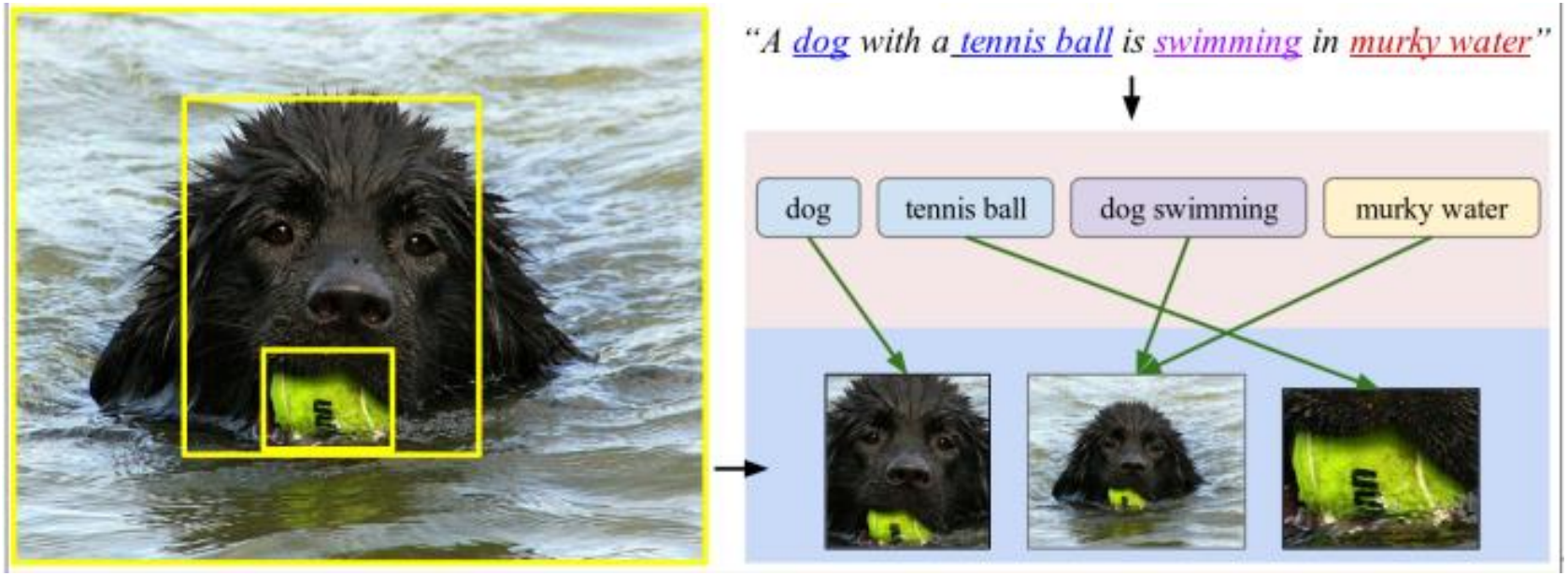
- E.g., multi-class retrieval
- Best translation may still require subjective evaluation (e.g., expressed emotion)

B Open-ended translations:



- E.g., sentence generation
- Almost always require subjective evaluations

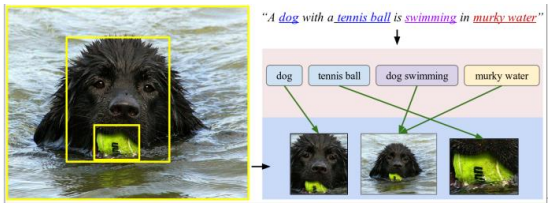
Core Challenge 3 – Alignment



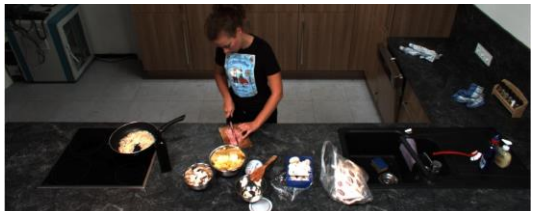
Karpathy et al., Deep Fragment Embeddings for Bidirectional Image Sentence Mapping, https://arxiv.org/pdf/1406.5679.pdf

Core Challenge 3 – Alignment

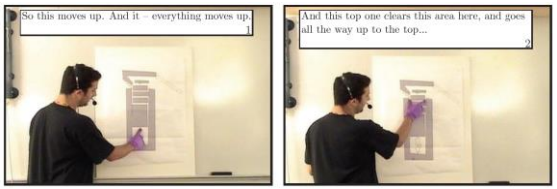
- **Image caption alignment**



- **Video description alignment**
  (e.g., cooking instruction videos)



- **Language-gesture co-reference**



Alignment:

"Identifying relations between elements from two or more different modalities"

Challenges:

I. Long range dependencies
II. Ambiguous segmentation
III. Limited annotated datasets with explicit alignments
IV. One-to-many relationships

Core Challenge 4 – Fusion

- **Audio-visual speech recognition**

"This tutorial is amazing"

- **Multimodal emotion recognition**

- **Multimedia event detection**

(a) answer-phone (a) get-out-car (a) fight-person

Fusion:

"Process of joining information from two or more modalities to perform a prediction"

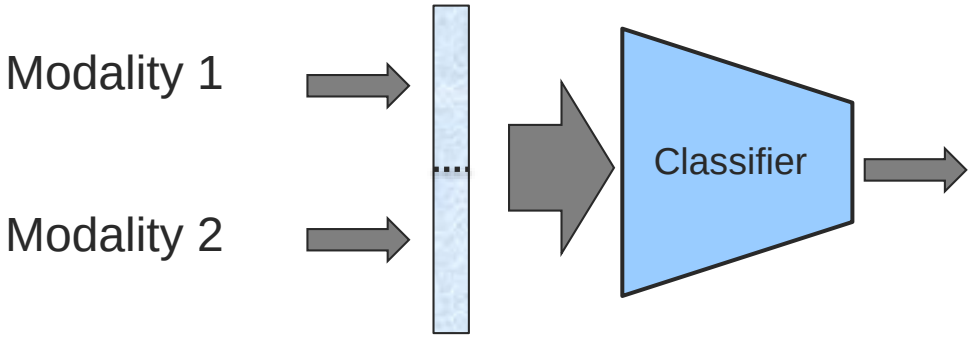
Challenges:

I. Different similarity metrics
II. Temporal contingency
III. Varying predictive power
IV. Different noise topology
V. Ambiguous correspondence

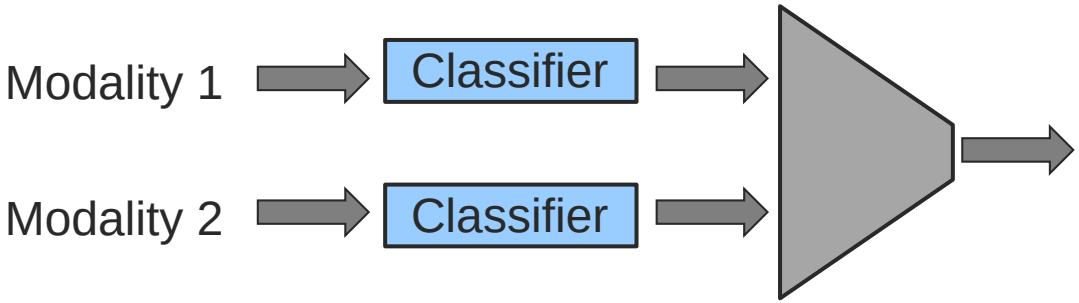
Core Challenge 4 – Fusion

A Model free approaches:

1) Early fusion



2) Late fusion



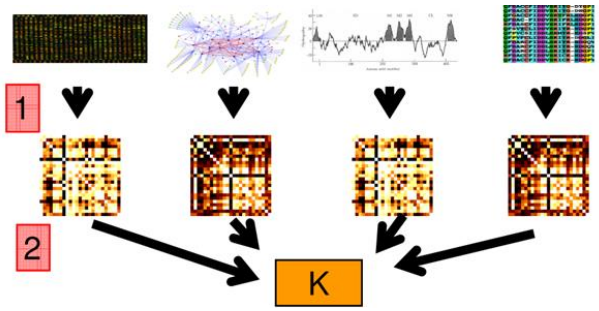
Core Challenge 4 – Fusion

B Model-based (intermediate) fusion:

1) Deep neural networks

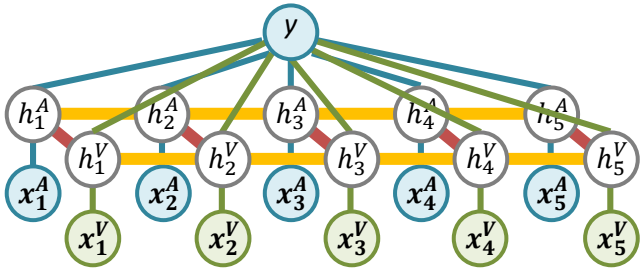
- Related to multimodal representation learning

2) Kernel-based methods



Multiple kernel learning

3) Graphical models



Multi-View Hidden CRF

Core Challenge 5 – Co-Learning

Co-learning:

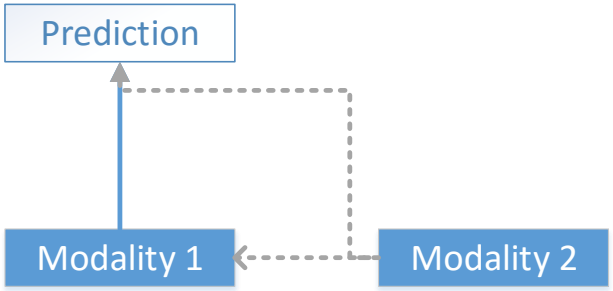
"Transferring knowledge between modalities and their representations"

Challenges:

I. Partially-observable views
II. Pivot identification
III. Collaborative overfitting
IV. Imperfect predictions
V. Potential divergence

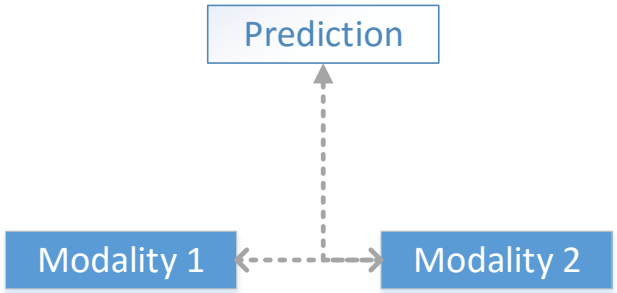
Core Challenge 5 – Co-Learning

A Unidirectional co-learning:



- One modality may have more resources
- Related to bootstrapping and domain adaptation

B Bidirectional co-learning:



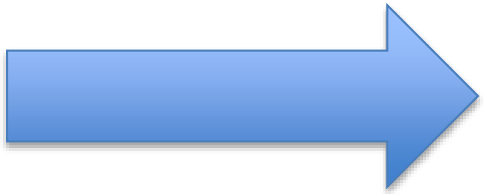
- Famous example: co-training [Blum & Mitchell]
- Strong correspondence requirement to be succesful

Basic Concepts: Score and Loss Functions

Basic Components of a Classifier

Image

(Size: 32*32*3)



?

1. Define a prediction score function
2. Define the loss function
3. Optimization technique

1) Score Function

Image

(Size: 32*32*3)

Duck ?
Cat ?
Dog ?
Pig ?
Bird ?

What should be the prediction score for each label class?

For linear classifier:

Input observation (*i*th *element of the dataset*) [3072x1]

$$f(x_i; W, b) = Wx_i + b$$

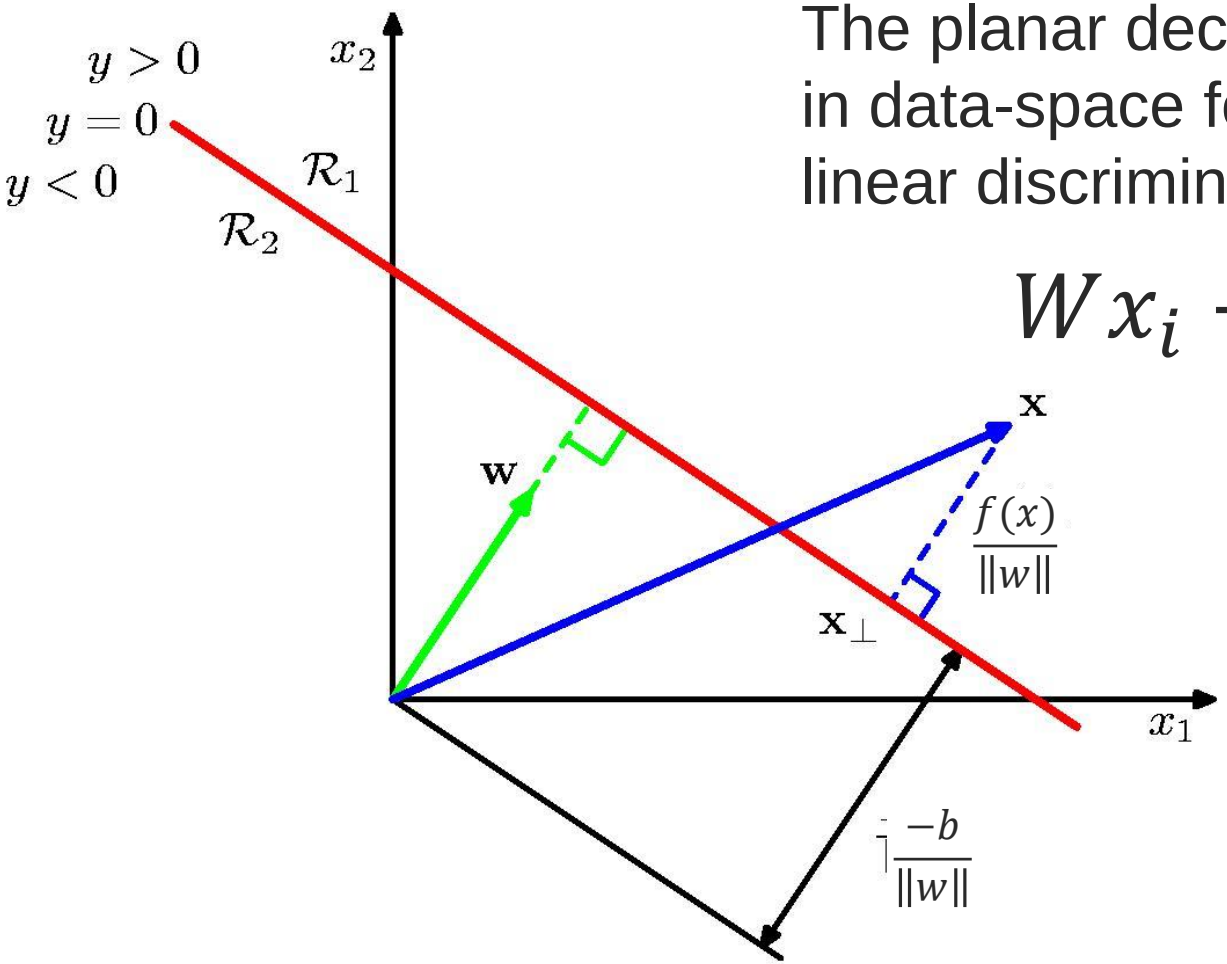
Class score [10x1]

Weights [10x3072]

Bias vector [10x1]

Parameters [10x3073]

Interpreting a Linear Classifier



The planar decision surface in data-space for the simple linear discriminant function:

$$Wx_i + b > 0$$

Some Notation Tricks

General formulation of linear classifier: $f(x_i; W, b)$

"dog" linear classifier:

$f(x_i; W_{dog}, b_{dog})$ or

$f(x_i; W, b)_{dog}$ or f_{dog}

Linear classifier for label *j*:

$f(x_i; W_j, b_j)$ or

$f(x_i; W, b)_j$ or f_j

Some Notation Tricks

$$f(x_i; W, b) = Wx_i + b \quad \rightarrow \quad f(x_i; W) = Wx_i$$

Weights x Input + Bias

[10x3072] [3072x1] [10x1]

Weights x Input

[10x3073] [3073x1]

The bias vector will be the last column of the weight matrix

Add a “1” at the end of the input observation vector

2) Loss Function

(or cost function or objective)

Scores $f(x_i; W)$ | **Label** $y_i = 2\ (dog)$ → **Loss** $L_i = ?$

Image x_i

(Size: 32*32*3)

0 (duck) ?	-12.3
1 (cat) ?	45.6
2 (dog) ?	98.7
3 (pig) ?	12.2
4 (bird) ?	-45.3

How to assign only one number representing how "unhappy" we are about these scores?

Multi-class problem

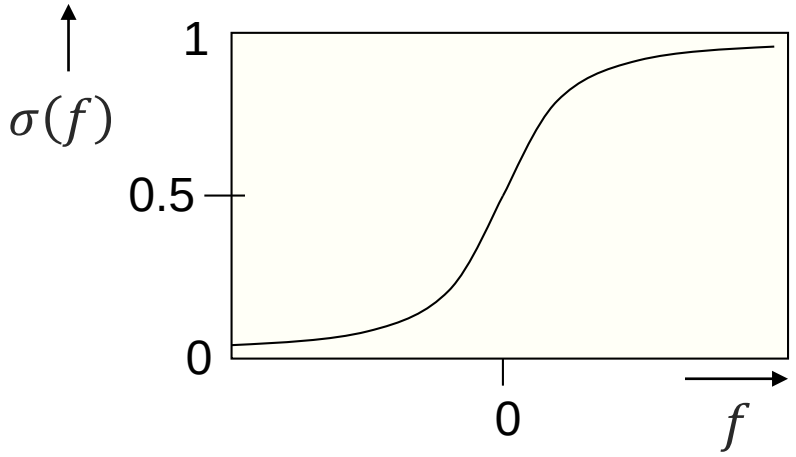
The loss function quantifies the amount by which the prediction scores deviate from the actual values.

First Loss Function: Cross-Entropy Loss

(or logistic loss)

Logistic function:

$$\sigma(f) = \frac{1}{1 + e^{-f}}$$



First Loss Function: Cross-Entropy Loss

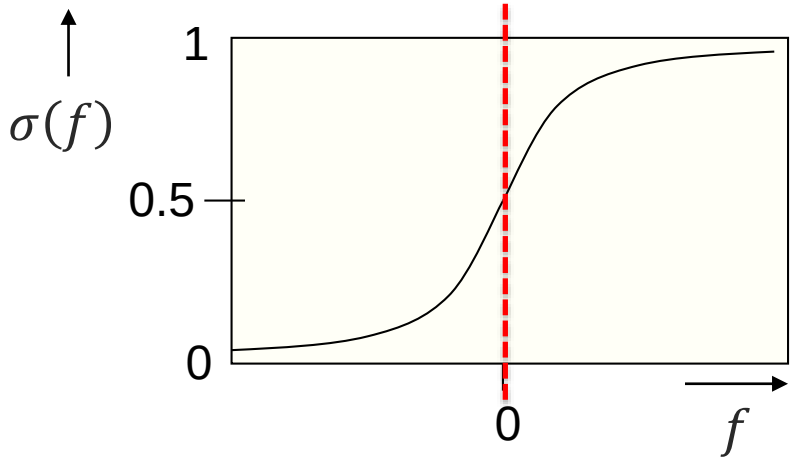
(or logistic loss)

Logistic function:

$$\sigma(f) = \frac{1}{1 + e^{-f}}$$

Logistic regression:
(two classes)

$$p(y_i = \text{"}dog\text{"} | x_i; w) = \sigma\left(w^T x_i\right)$$



First Loss Function: Cross-Entropy Loss

(or logistic loss)

Logistic function:

$$\sigma(f) = \frac{1}{1 + e^{-f}}$$

Logistic regression:
(two classes)

$$p(y_i = "dog" | x_i; w) = \sigma\left(w^T x_i\right)$$

Softmax function:
(multiple classes)

$$p(y_i | x_i; W) = \frac{e^{f_{y_i}}}{\sum_j e^{f_j}}$$

First Loss Function: Cross-Entropy Loss

(or logistic loss)

Cross-entropy loss:

$$L_i = -\log\left(\frac{e^{f_{y_i}}}{\sum_j e^{f_j}}\right)$$

Softmax function

Minimizing the negative log likelihood.

Second Loss Function: Hinge Loss

(or Multi-class SVM loss)

$$L_i = \sum_{j \neq y_i} \max(0, f(x_i, W)_j - f(x_i, W)_{y_i} + \Delta)$$

loss due to example i

sum over all incorrect labels

difference between the correct class score and incorrect class score

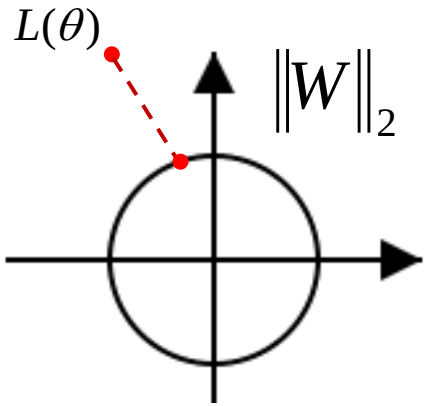
Regularization

Regularization factor

$$L_i = -\log\left(\frac{e^{f_{y_i}(x_i;W)}}{\sum_j e^{f_j(x_i;W)}}\right) + \lambda R(W)$$

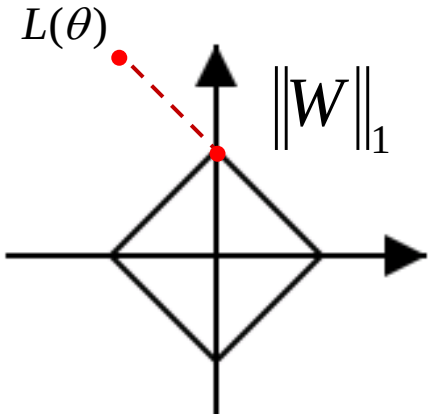
L-2 Norm (Gaussian prior):

$$R(W) = \|W\|_2$$



L-1 Norm (Laplacian prior):

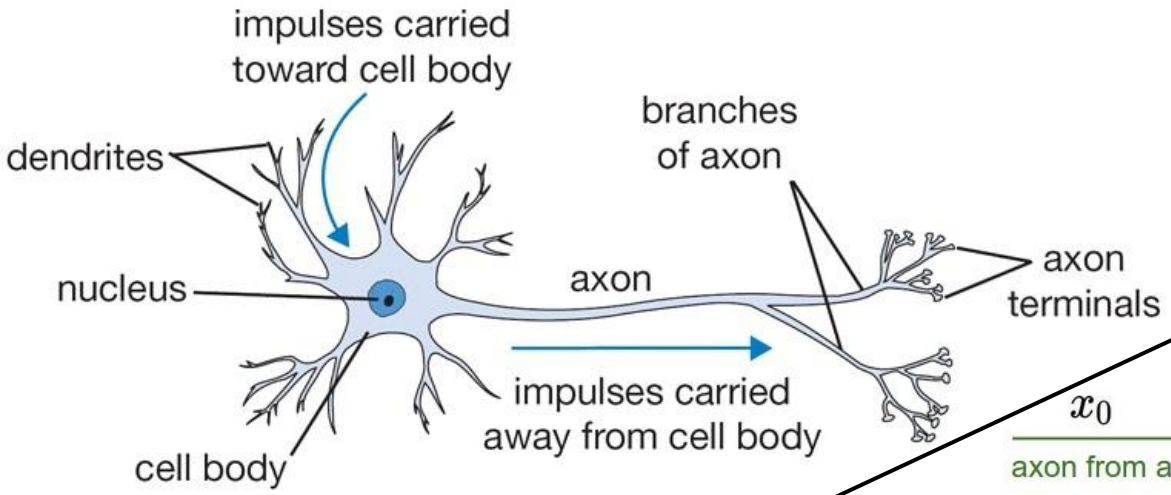
$$R(W) = \|W\|_1$$

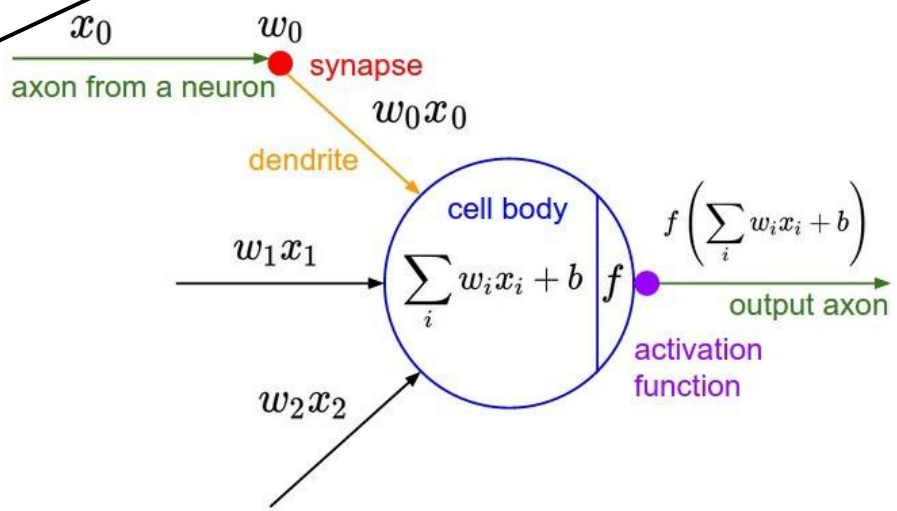


Basic Concepts: Neural Networks

Neural Networks – inspiration

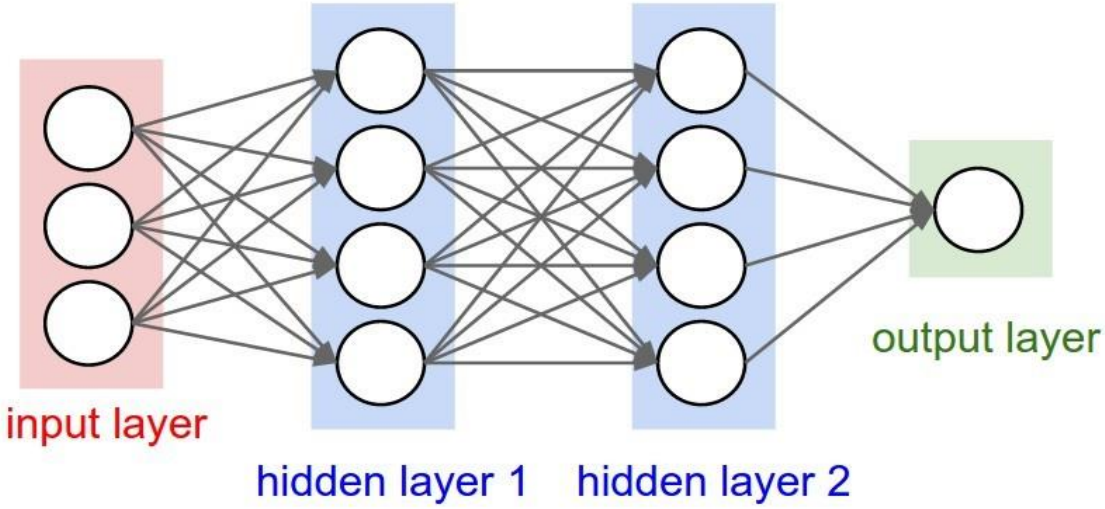
- Made up of artificial neurons





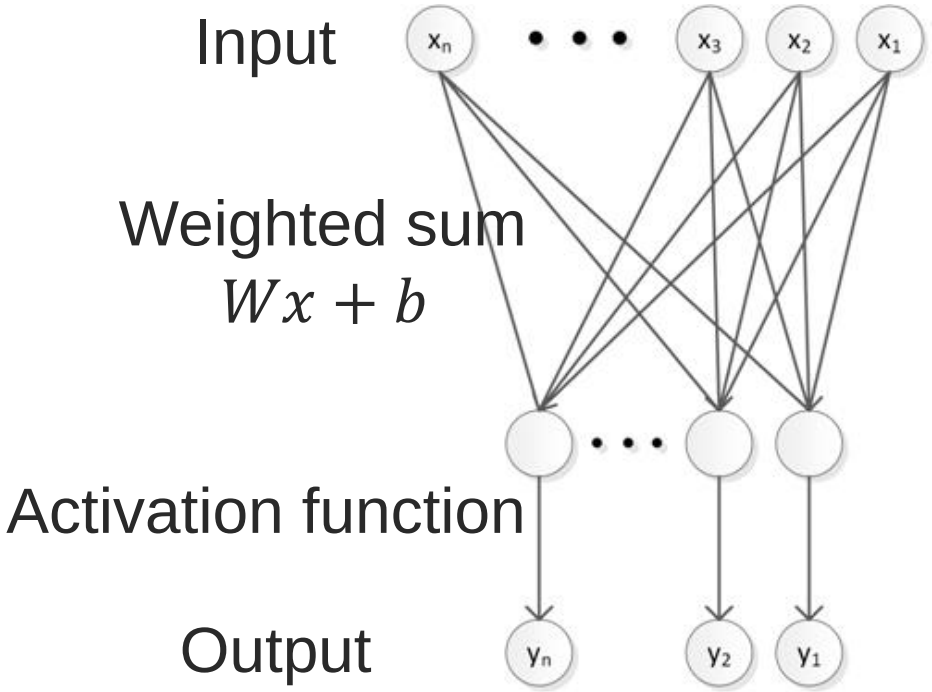
Neural Networks – score function

- Made up of artificial neurons
  - Linear function (dot product) followed by a nonlinear activation function
- Example a Multi Layer Perceptron



Basic NN building block

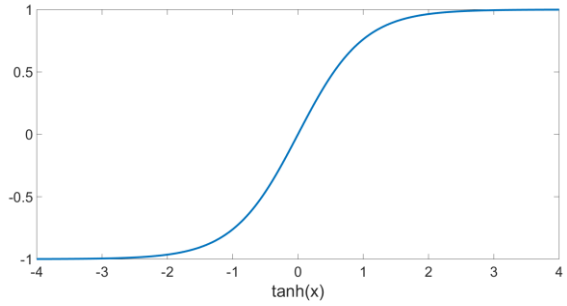
- Weighted sum followed by an activation function

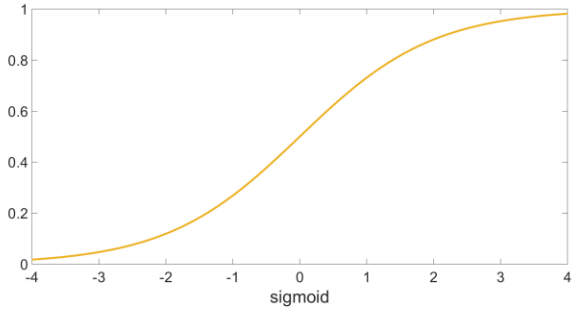


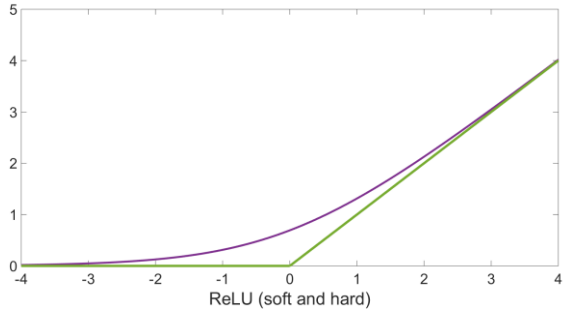
$$y = f(Wx + b)$$

Neural Networks – activation function

- $f(x) = \tanh(x)$
- Sigmoid - $f(x) = (1 + e^{-x})^{-1}$
- Linear – $f(x) = ax + b$
- ReLU $f(x) = \max(0, x) \sim \log(1 + \exp(x))$
  - Rectifier Linear Units
  - Faster training - no gradient vanishing
  - Induces sparsity







Neural Networks – loss function

- Already discussed it – cross-entropy, Euclidean loss, cosine similarity, etc.
- Combine it with the score function to have an end-to-end training objective
- As example use Euclidean loss for data-point *I*

$$L_i = (f(x_i) - y_i)^2 = (f_{3;W_3}(f_{2;W_2}(f_{1;W_1}(x_i))))^2$$

- Full loss is computed across all training samples

$$L = \sum_i (f(x_i) - y_i)^2$$

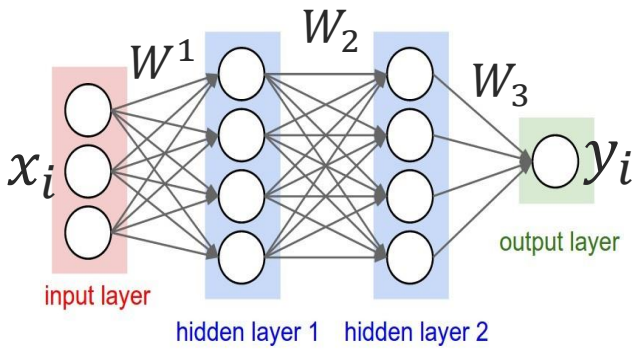
Multi-Layer Feedforward Network

Activation functions (individual layers)

$$f_{1;W_1}(x) = \sigma(W_1 x + b_1)$$

$$f_{2;W_2}(x) = \sigma(W_2 x + b_2)$$

$$f_{3;W_3}(x) = \sigma(W_3 x + b_3)$$



Score function

$$y_i = f(x_i) = f_{3;W_3}(f_{2;W_2}(f_{1;W_1}(x_i)))$$

Loss function (e.g., Euclidean loss)

$$L_i = (f(x_i) - y_i)^2 = (f_{3;W_3}(f_{2;W_2}(f_{1;W_1}(x_i))))^2$$

Basic Concepts: Optimization

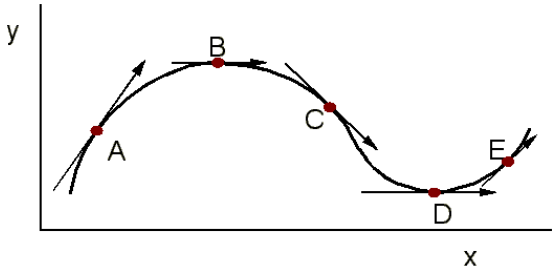
Optimizing a generic function

- We want to find a minimum (or maximum) of a generic function
- How do we do that?
  - Searching everywhere (global optimum) is computationally infeasible
  - We could search randomly from our starting point (mostly picked at random) – impractical and not accurate
  - Instead we can follow the gradient

What is a gradient?

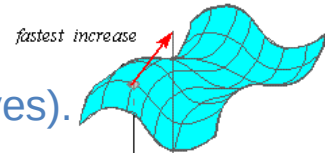
- Geometrically
  - Points in the direction of the greatest rate of increase of the function and its magnitude is the slope of the graph in that direction
- More formally in 1D

$$\frac{df(x)}{dx} = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$



- In higher dimensions

$$\frac{\partial f}{\partial x_i}(a_1, \ldots, a_n) = \lim_{h \to 0} \frac{f(a_1, \ldots, a_i + h, \ldots, a_n) - f(a_1, \ldots, a_i, \ldots, a_n)}{h}.$$

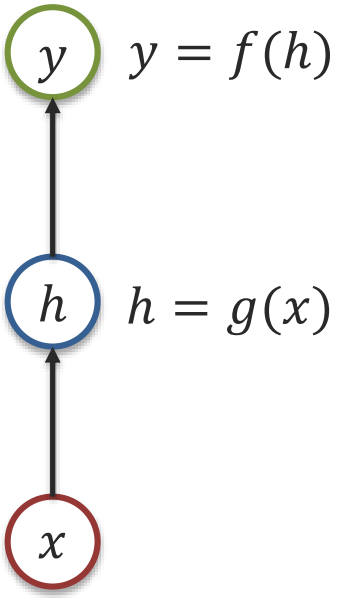


➢ In multiple dimension, the **gradient** is the vector of (partial derivatives).

Gradient Computation

Chain rule:

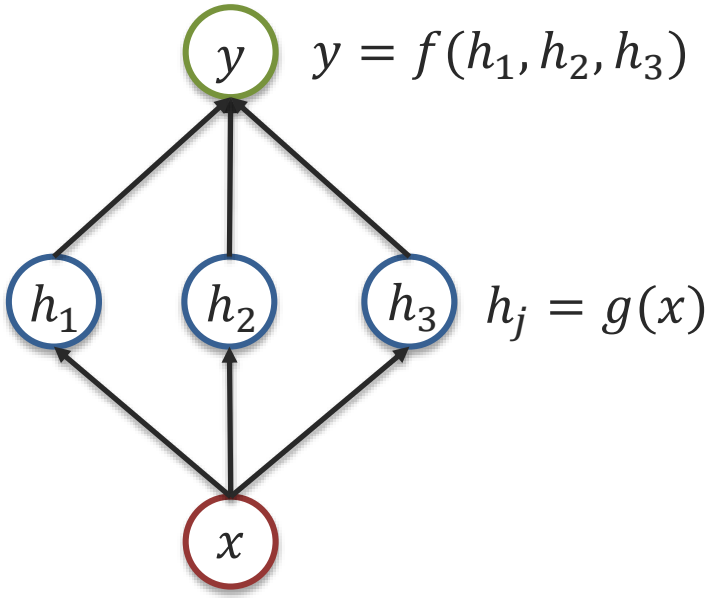
$$\frac{\partial y}{\partial x} = \frac{\partial y}{\partial h}\frac{\partial h}{\partial x}$$



Optimization: Gradient Computation

Multiple-path chain rule:

$$\frac{\partial y}{\partial x} = \sum_j \frac{\partial y}{\partial h_j} \frac{\partial h_j}{\partial x}$$



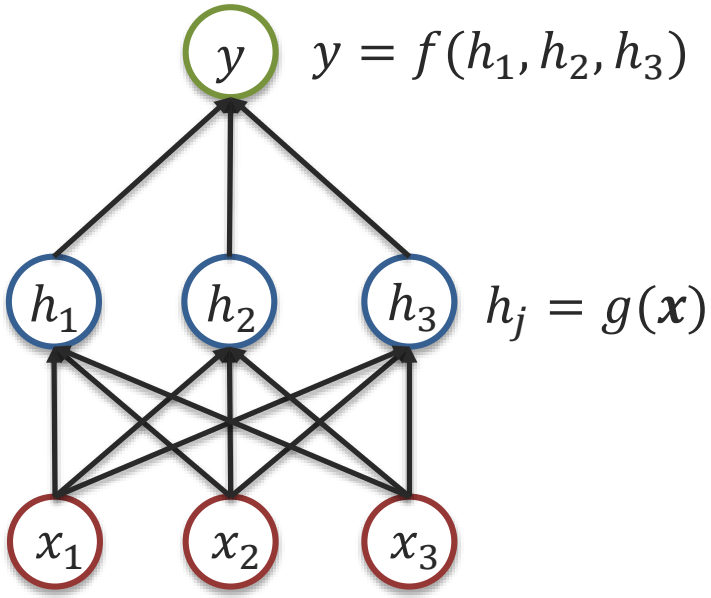
Optimization: Gradient Computation

Multiple-path chain rule:

$$\frac{\partial y}{\partial x_1} = \sum_j \frac{\partial y}{\partial h_j} \frac{\partial h_j}{\partial x_1}$$

$$\frac{\partial y}{\partial x_2} = \sum_j \frac{\partial y}{\partial h_j} \frac{\partial h_j}{\partial x_1}$$

$$\frac{\partial y}{\partial x_3} = \sum_j \frac{\partial y}{\partial h_j} \frac{\partial h_j}{\partial x_1}$$



Optimization: Gradient Computation

Vector representation:

$$\nabla_x y = \left[\frac{\partial y}{\partial x_1}, \frac{\partial y}{\partial x_2}, \frac{\partial y}{\partial x_3}\right]$$

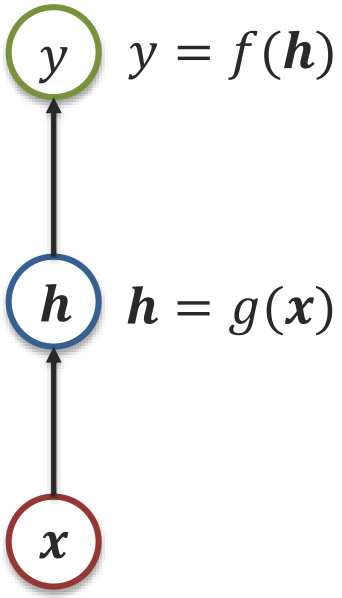
Gradient

$$\nabla_x y = \left(\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}}\right)^T \nabla_{\boldsymbol{h}} y$$

"local" Jacobian

(matrix of size $|h| \times |x|$ computed using partial derivatives)

"backprop" Gradient



Backpropagation Algorithm

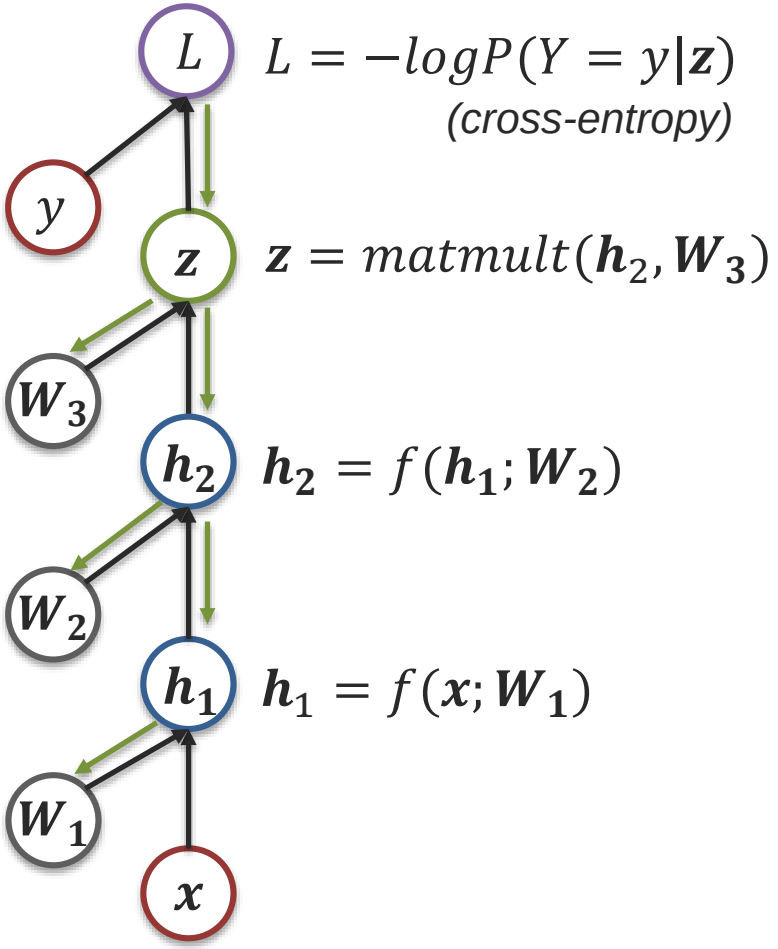
Forward pass

- Following the graph topology, compute value of each unit

Backpropagation pass

- Initialize output gradient = 1
- Compute "local" Jacobian matrix using values from forward pass
- Use the chain rule:

Gradient = "local" Jacobian x "backprop" gradient



Computational Graph: Multi-layer Feedforward Network

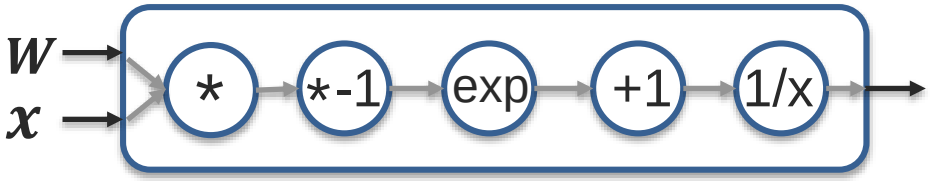
Computational unit:

$\boldsymbol{h} = f(\boldsymbol{x}; \boldsymbol{W})$

- Multiple input
- One output
- Vector/tensor

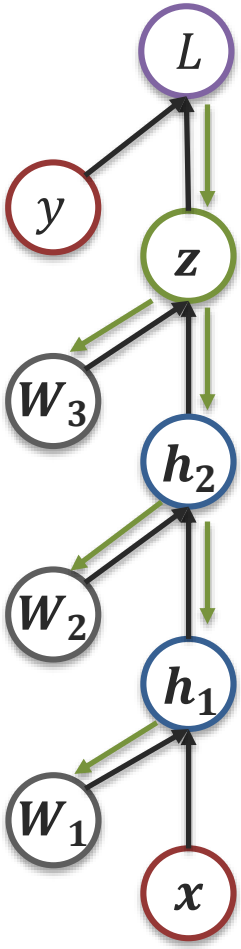
- Sigmoid unit:

$h_j = (1 + e^{-W_j \boldsymbol{x}})^{-1}$



Differentiable "unit" function!

(or close approximation to compute "local Jacobian)



$L = -logP(Y = y|\boldsymbol{z})$

(cross-entropy)

$\boldsymbol{z} = matmult(\boldsymbol{h_2}, \boldsymbol{W_3})$

$\boldsymbol{h_2} = f(\boldsymbol{h_1}; \boldsymbol{W_2})$

$\boldsymbol{h_1} = f(\boldsymbol{x}; \boldsymbol{W_1})$

Optimization - how to follow the gradient

- Many methods for optimization
  - **Gradient Descent (actually the "simplest" one)**
  - Newton methods (use Hessian)
  - Quasi-Newton (use approximate Hessian)
    - BFGS
    - LBFGS
    - Don't require learning rates
    - But, do not work with stochastic and batch methods
- All of them look at the gradient

Gradient Descent Example 1

Gradient Descent Example 1

Gradient Descent Example 1

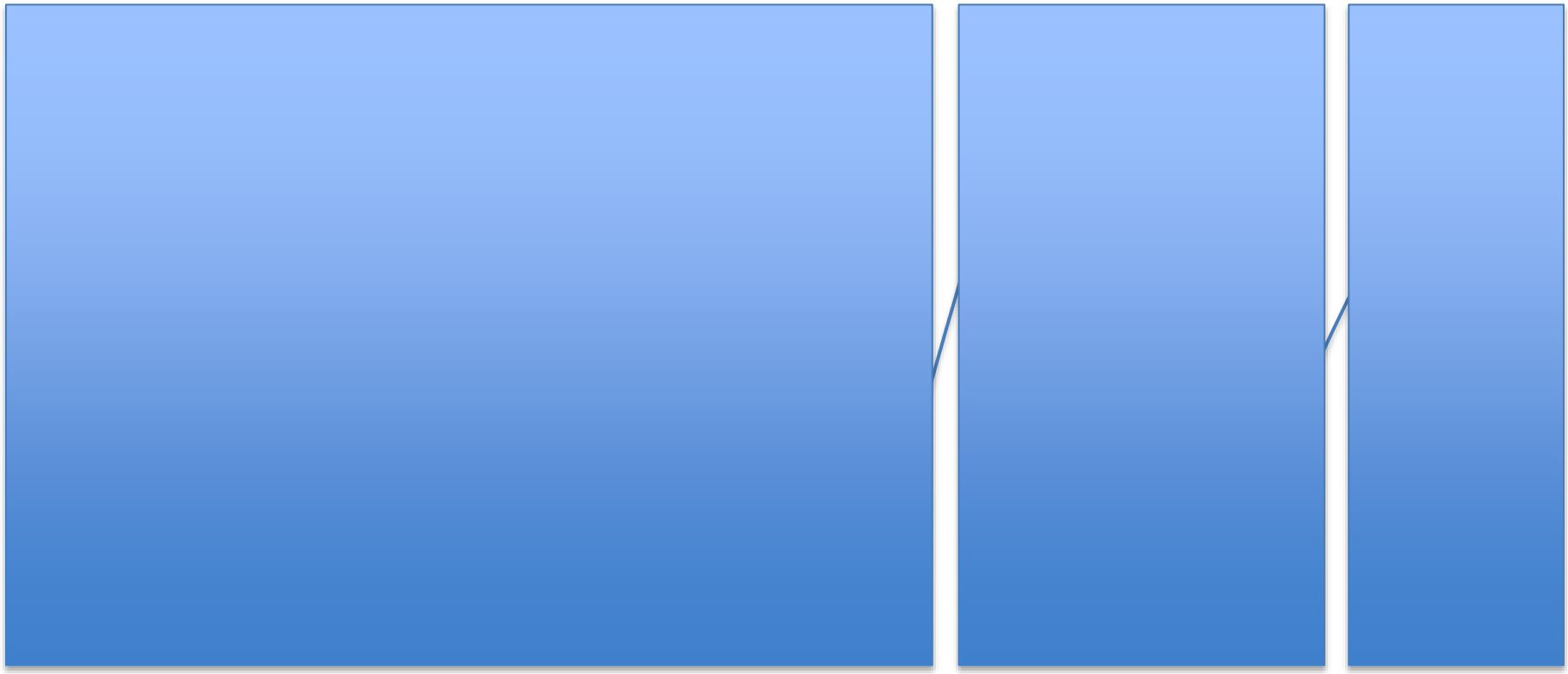
Gradient Descent Example 1

Gradient Descent Example 1

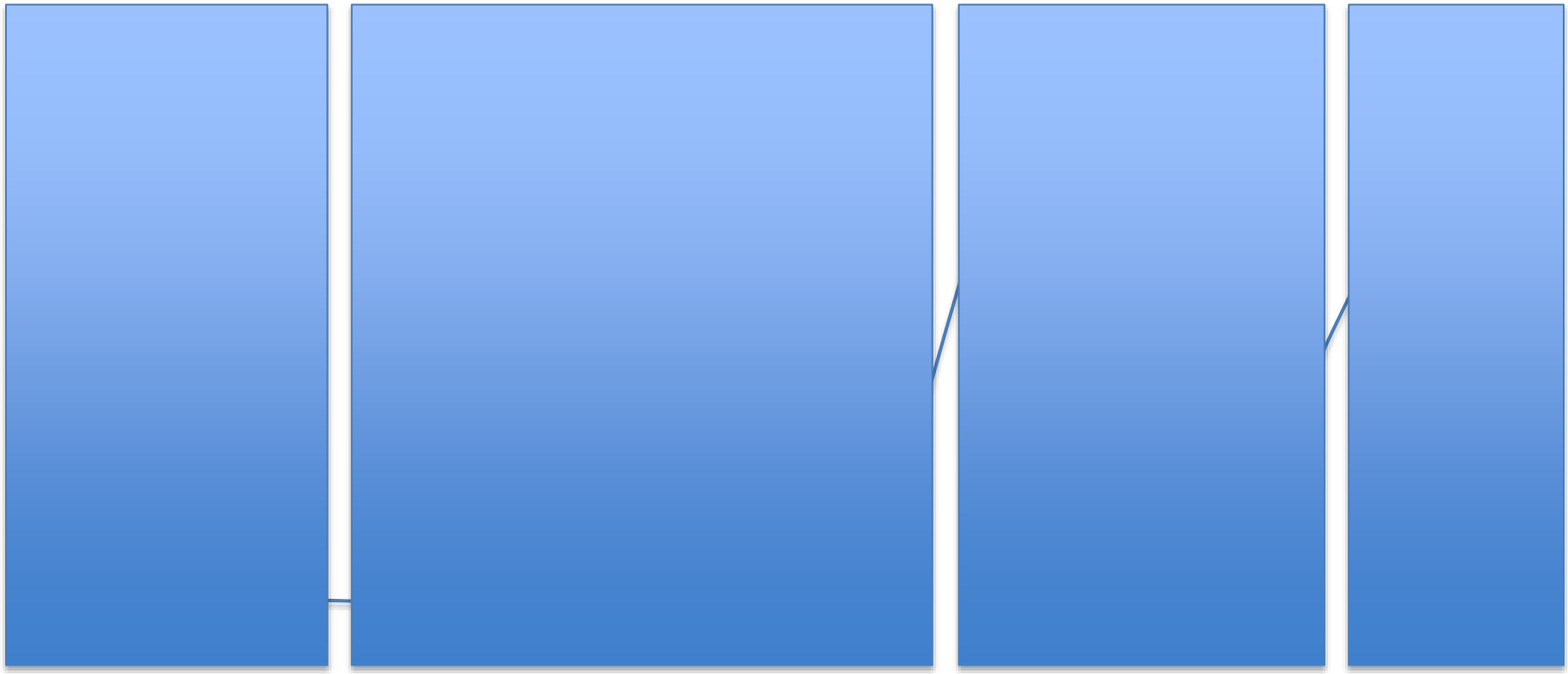
- Converged

Gradient Descent Example 2 – Large Learning Rate

Gradient Descent Example 2 – Large Learning Rate

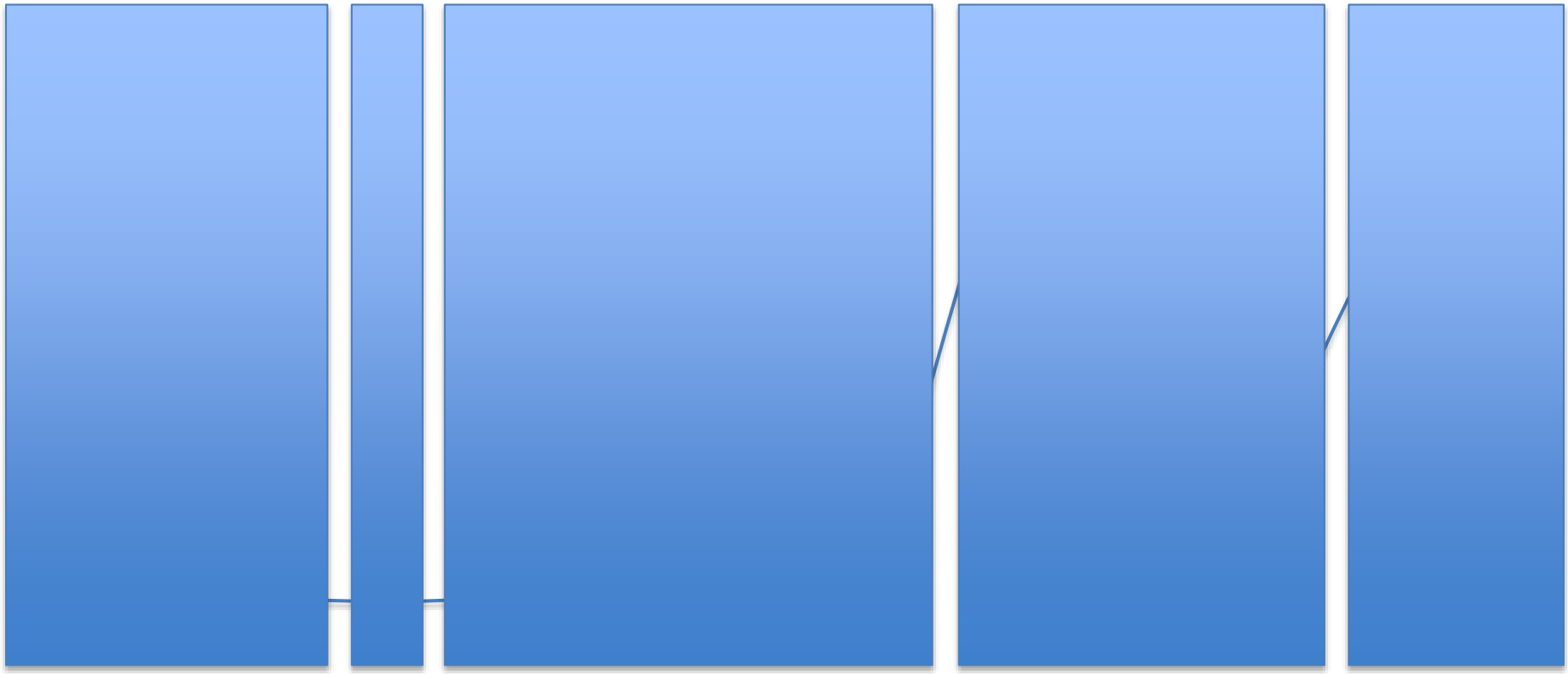


Gradient Descent Example 2 – Large Learning Rate



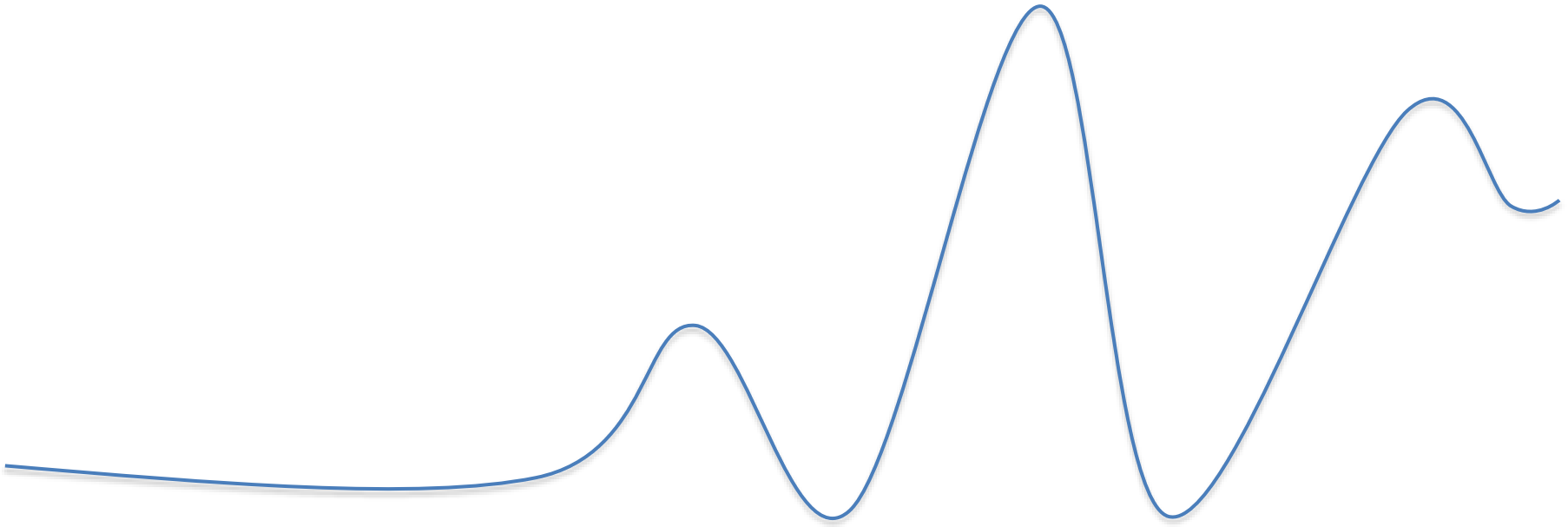
Gradient Descent Example 2 – Large Learning Rate

- Convergence reached

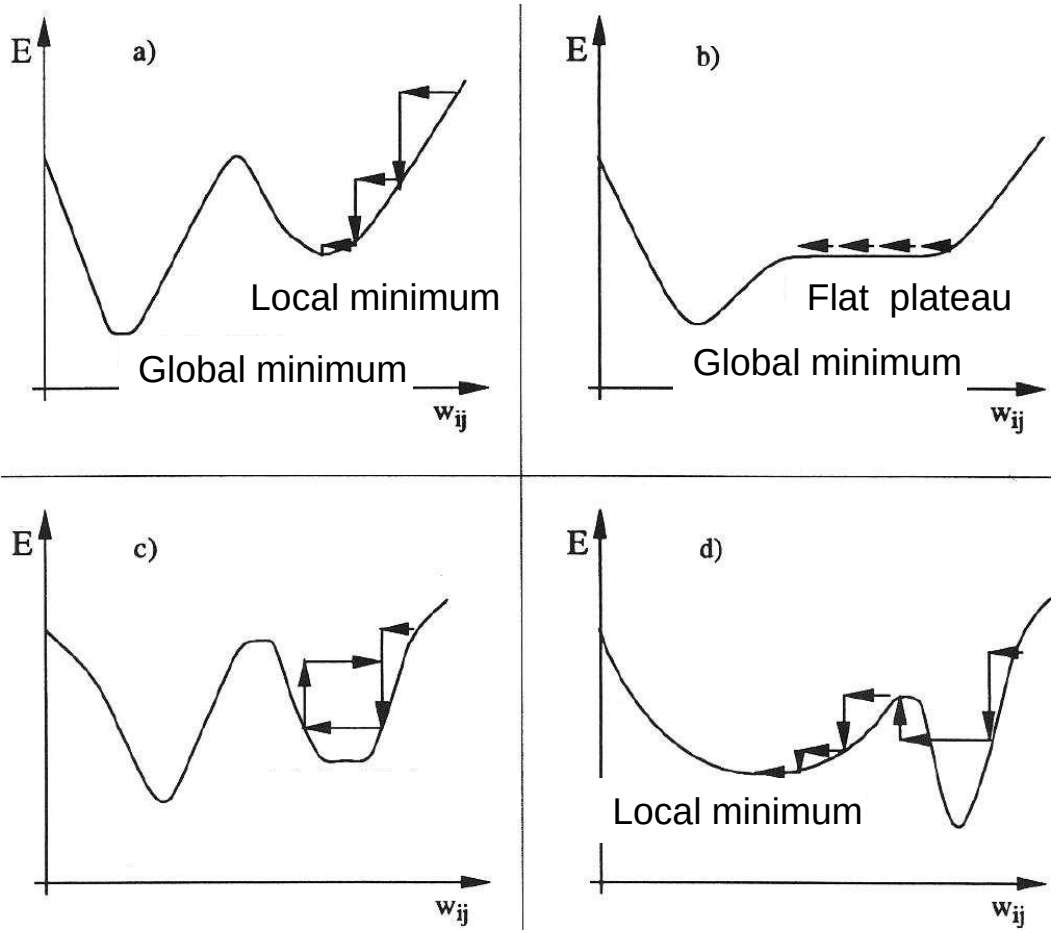


Gradient Descent

- We are looking at a potentially very complex surface through a pinhole and hope that we reach a **good enough** (not optimal) value



Potential issues



- Problems that can occur?
  - Getting stuck in local minima (global minimum is never found) (a)
  - Getting stuck on flat plateaus of the error-plane (b)
  - Oscillations in error rates (c)
  - Learning rate is critical (d)

Parameter Update Strategies

Stochastic ("batch") gradient descent:

$$\theta^{(t+1)} = \theta^{t} - \epsilon_k \nabla_\theta L$$

- $\theta^{(t+1)}$: New model parameters
- θ^{t}: Previous parameters
- ϵ_k: **Learning rate** at iteration *k*
- $\nabla_\theta L$: Gradient of our loss function

$$\epsilon_k = (1 - \alpha)\epsilon_0 + \alpha\epsilon_\tau$$

- ϵ_k: **Learning rate** at iteration *k*
- $(1 - \alpha)$: **Decay**
- ϵ_0: Initial learning rate
- $\alpha\epsilon_\tau$: Decay learning rate linearly until iteration τ

Extensions:

- with momentum
- AdaGrad
- RMSProp

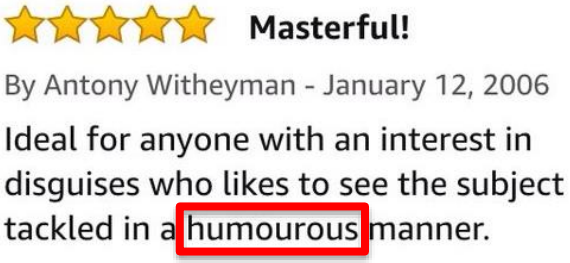
Summary – Backpropagation and Deep Learning

- Originally inspired by human brain neurons
- Today, researchers in representation learning and deep learning are extending this definition
- Based on “computational units” which can be differentiable (for backpropagation)
- Can also include other machine learning “units” such Conditional Random Fields
- Second-order methods can also be used to optimize deep networks (by computing Hessian)

Unimodal representations: Language Modality

Unimodal Classification – Language Modality

Written language



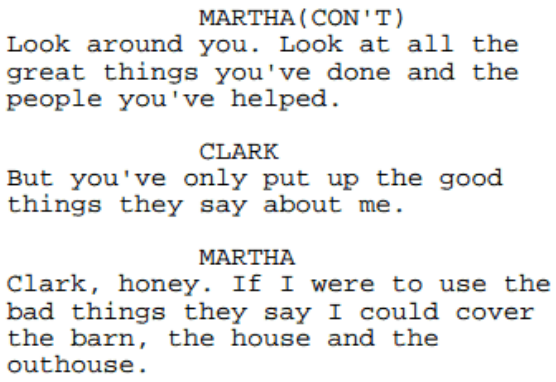
★★★★★ **Masterful!**

By Antony Witheyman - January 12, 2006

Ideal for anyone with an interest in disguises who likes to see the subject tackled in a humourous manner.

0 of 4 people found this review helpful

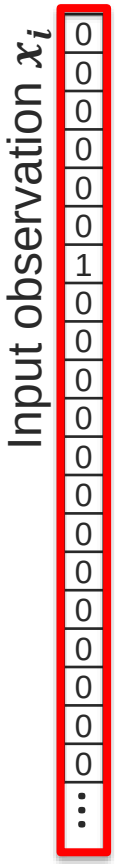
Spoken language



MARTHA(CON'T)
Look around you. Look at all the great things you've done and the people you've helped.

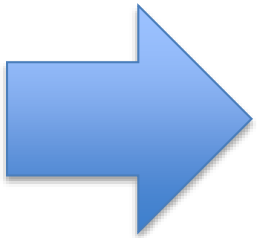
CLARK
But you've only put up the good things they say about me.

MARTHA
Clark, honey. If I were to use the bad things they say I could cover the barn, the house and the outhouse.



"one-hot" vector

$|x_i|$ = number of words in dictionary



Word-level classification

Part-of-speech ?
(noun, verb,…)

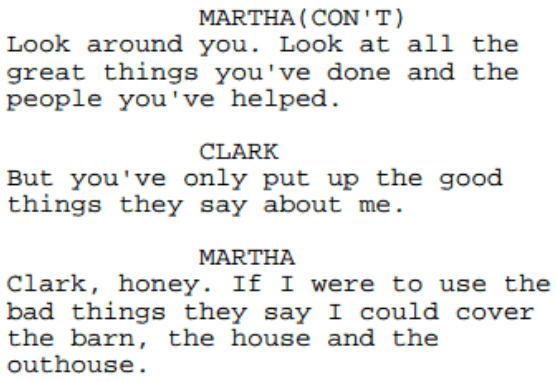
Sentiment ?
(positive or negative)

Named entity ?
(names of person,…)

Unimodal Classification – Language Modality

Written language

Spoken language

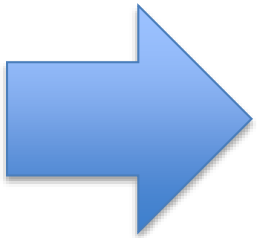


Input observation x_i

“bag-of-word” vector

$|x_i|$ = number of words in dictionary

Document-level classification

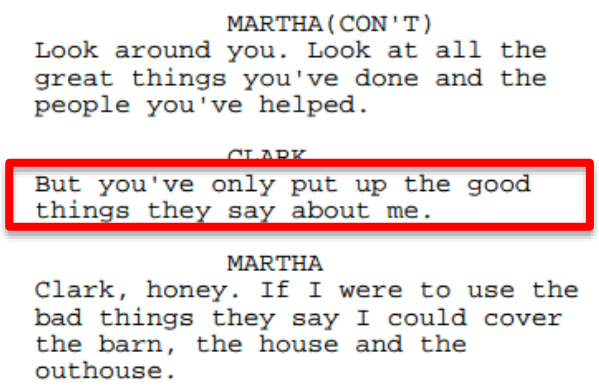


Sentiment ?
(positive or negative)

Unimodal Classification – Language Modality

Written language

Spoken language

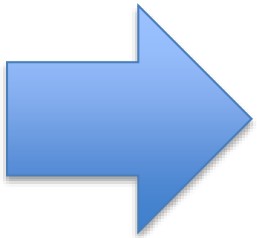


Input observation x_i

“bag-of-word” vector

$|x_i|$ = number of words in dictionary

Utterance-level classification



Sentiment ?
(positive or negative)

How to Learn (Better) Language Representations?

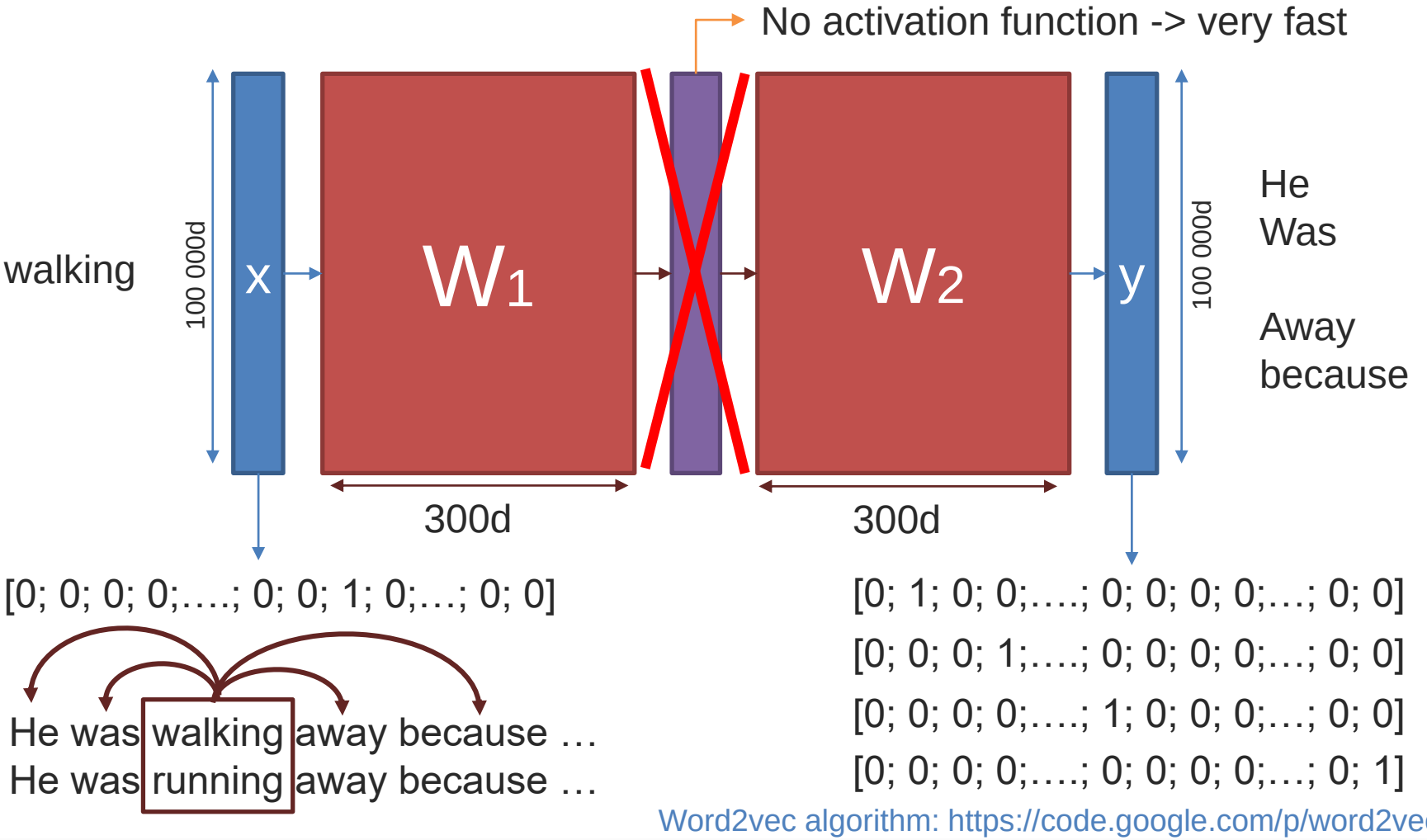
Distribution hypothesis: Approximate the word meaning by its surrounding words

Words used in a similar context will lie close together

Instead of capturing co-occurrence counts directly, predict surrounding words of every word

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0}\log p(w_{t+j}|w_t)$$

How to Learn (Better) Language Representations?



Word2vec algorithm: https://code.google.com/p/word2vec/

How to use these word representations

If we would have a vocabulary of 100 000 words:

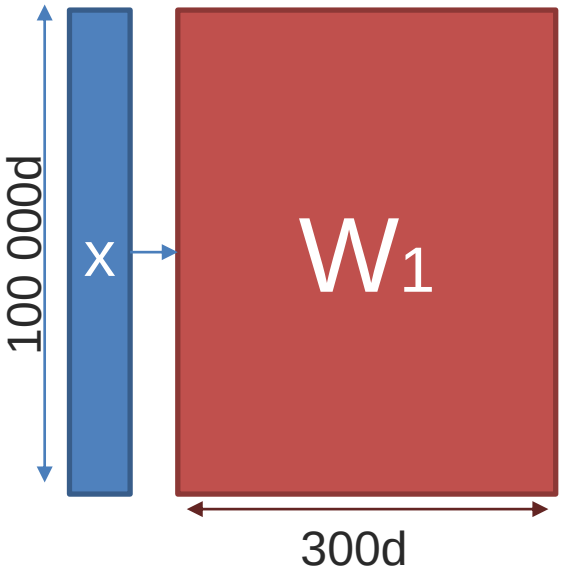
Classic NLP: 100 000 dimensional vector

Walking: [0; 0; 0; 0;….; 0; 0; 1; 0;…; 0; 0]

Running: [0; 0; 0; 0;….; 0; 0; 0; 0;…; 1; 0]

Similarity = 0.0

Transform: x'=x*W



Goal: 300 dimensional vector

Walking: [0,1; 0,0003; 0;….; 0,02; 0.08; 0,05]

Running: [0,1; 0,0004; 0;….; 0,01; 0.09; 0,05]

Similarity = 0.9

Word representations: examples

Word similarity test:

Trained on 400 million tweets having 5 billion words

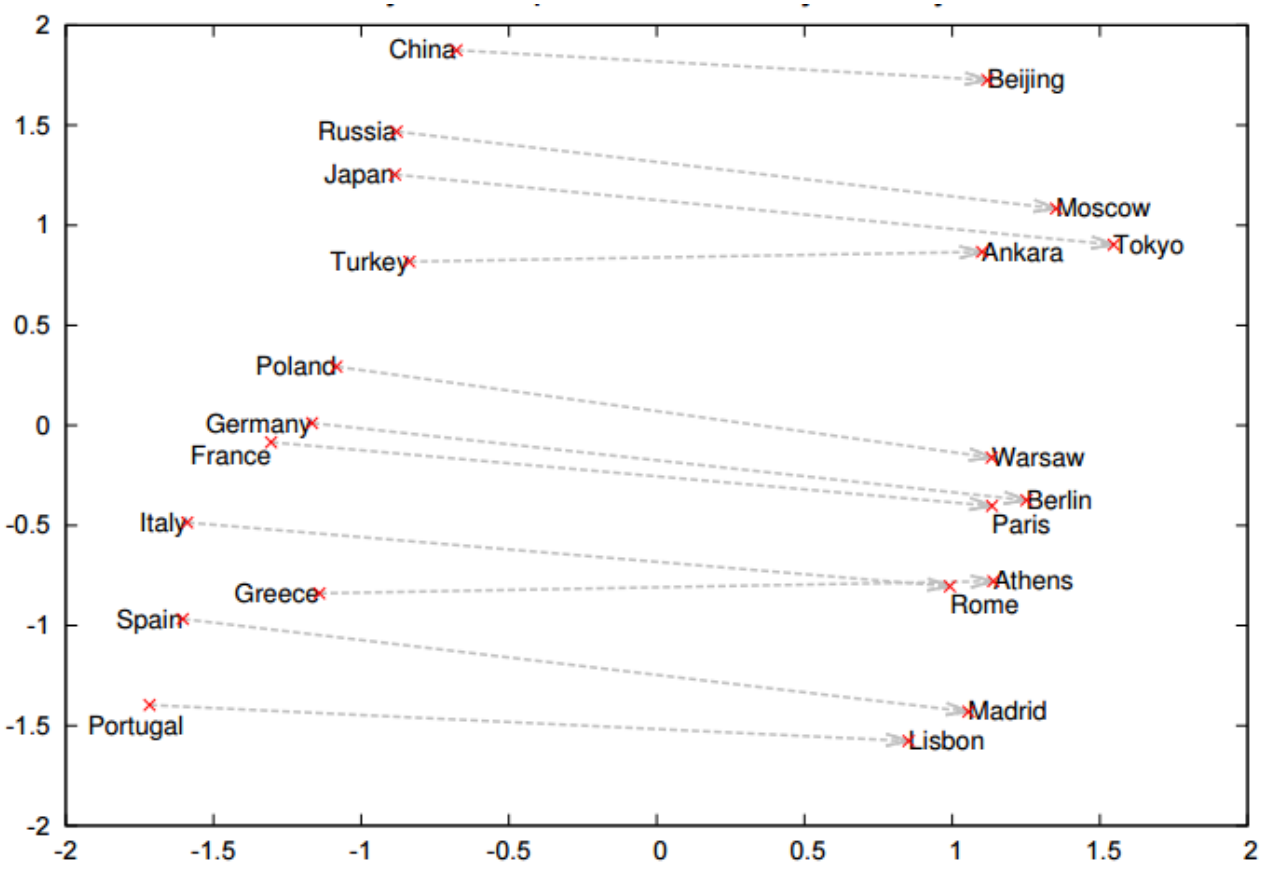
Input: running	Cosine similarity	Input: :)	Cosine similarity
runnin	0.758099	:))	0.885355
runing	0.702119	=)	0.836011
Running	0.69014	:D	0.818340
runnning	0.669039	;))	0.814380
sprinting	0.587385	(:	0.809806
runnung	0.578426	:)))	0.808298
run	0.576671	:-)	0.798115
walking/running	0.563114	:))))	0.777765
runin	0.556682	;)	0.772422
walking	0.542137	:-))	0.758584

Vector space models of words

- While learning these word representations, we are actually building a vector space in which all words reside with certain relationships between them
- Encodes both syntactic and semantic relationships
- This vector space allows for algebraic operations:

Vec(king) – vec(man) + vec(woman) ≈ vec(queen)

Vector space models of words: semantic relationships



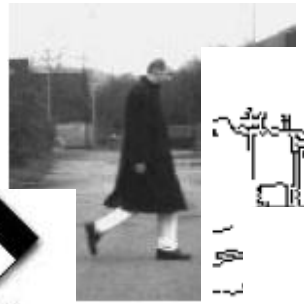
Trained on the Google news corpus with over 300 billion words

Mikolov et al., “Distributed Representations of Words and Phrases and their Compositionality”, NIPS 2013

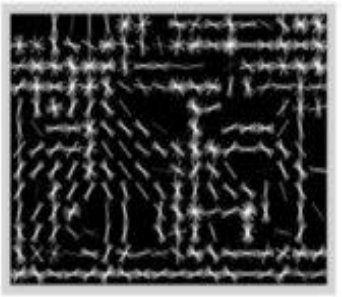
Unimodal representations: Visual Modality

Visual Descriptors

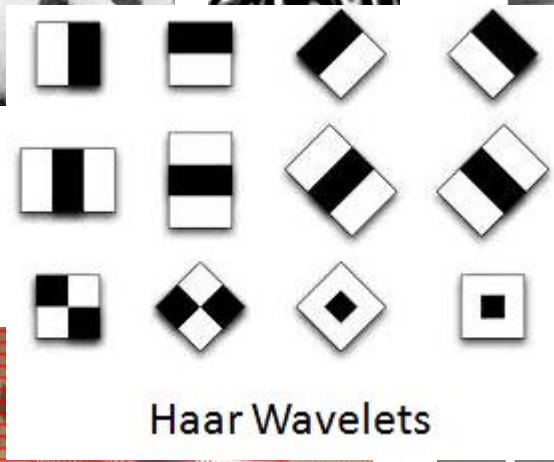
I

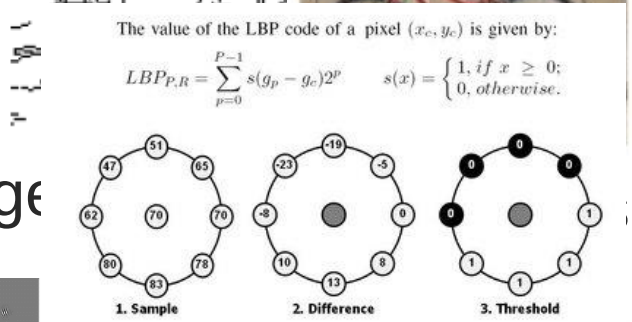


Edge

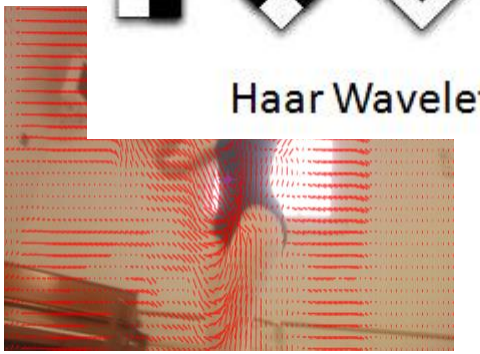


of Oriented G

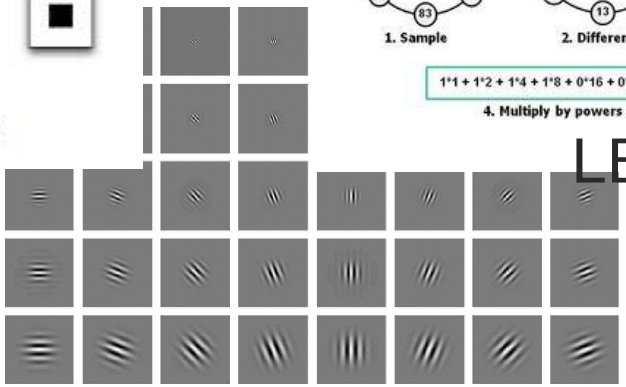




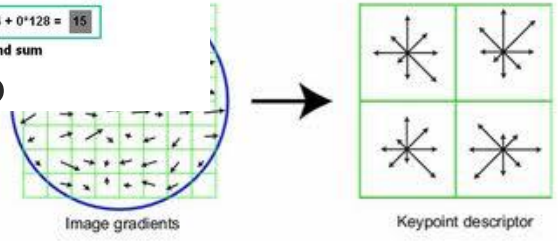
LBP



Optical Flow



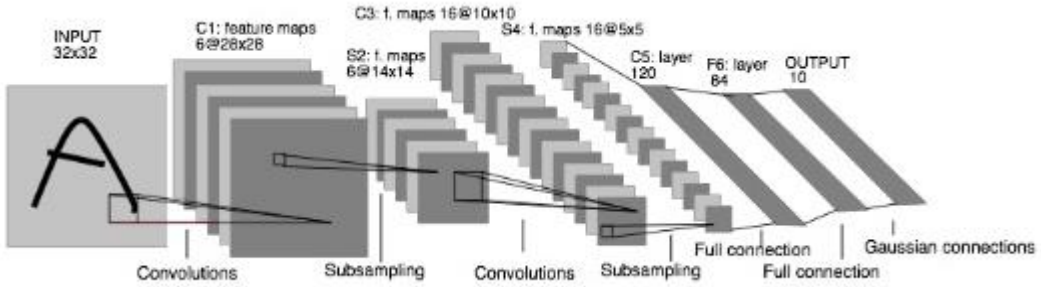
Gabor Jets

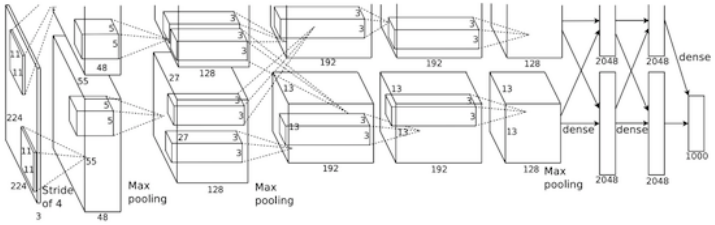


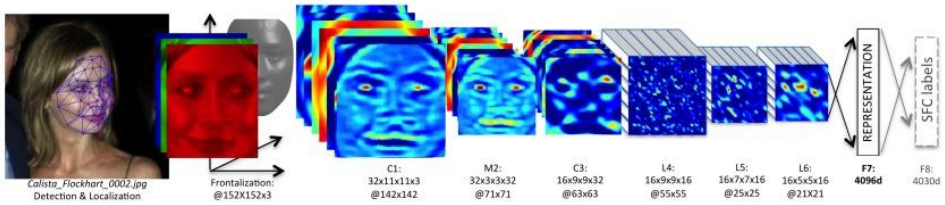
SIFT descriptors

Convolutional Neural Networks

- Extremely popular in Computer Vision
- State of the art results – object recognition, face recognition, segmentation, OCR, visual emotion recognition

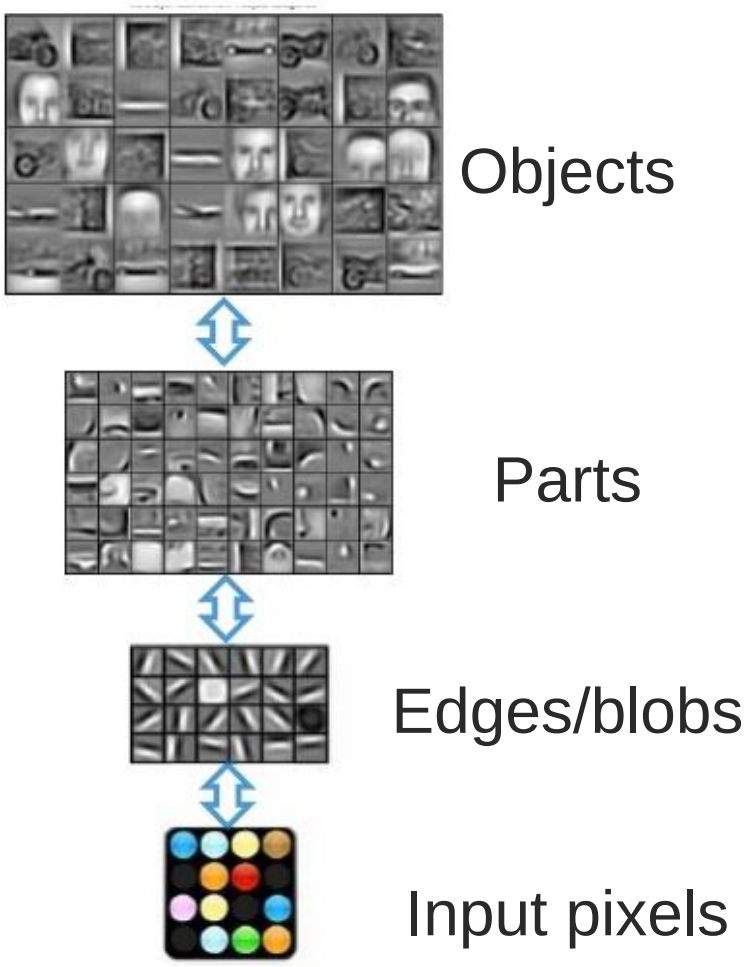






Why Convolutional Neural Networks

- Using Multi Layer perceptrons does not work well for images
- Too many parameters to learn
- Want to exploit image structure
  - Restrict what the model can learn
- Intention to build more abstract representation as we go up every layer
- Addition of:
  - Convolution layer
  - Pooling layer



Convolutional Neural Networks

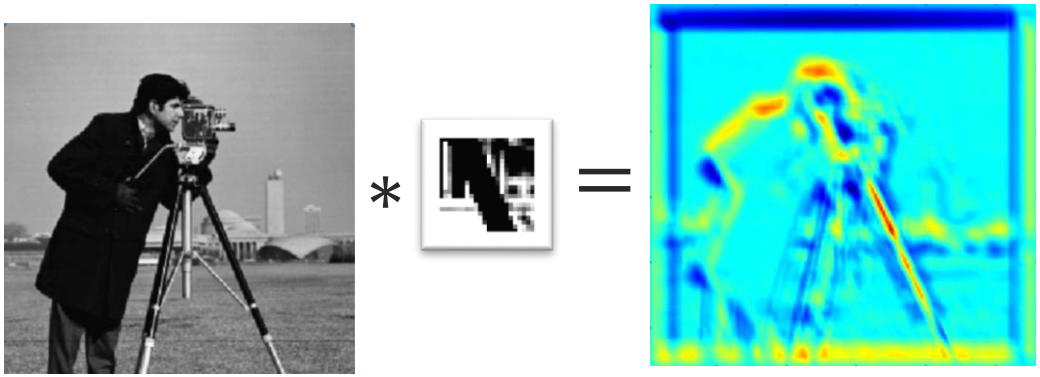
- They are called convolutional as they imitate convolution operation
- A basic mathematical operation (that given two functions returns a function)

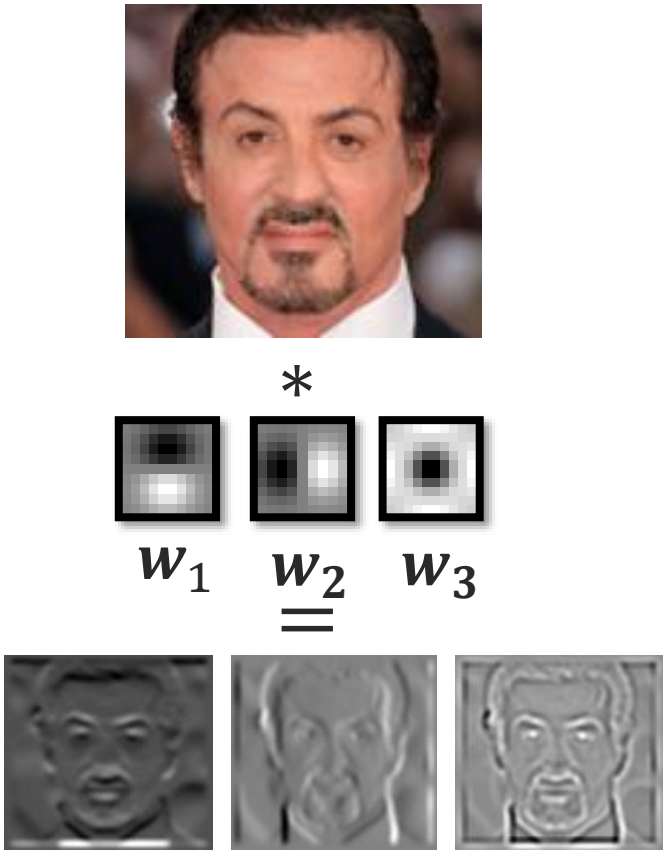
$$(f * g)[n] \stackrel{\text{def}}{=} \sum_{m=-\infty}^{\infty} f[m]\, g[n-m]$$

- Can use tricks to make it really efficient
  - Fourier transform

Convolution in 2D

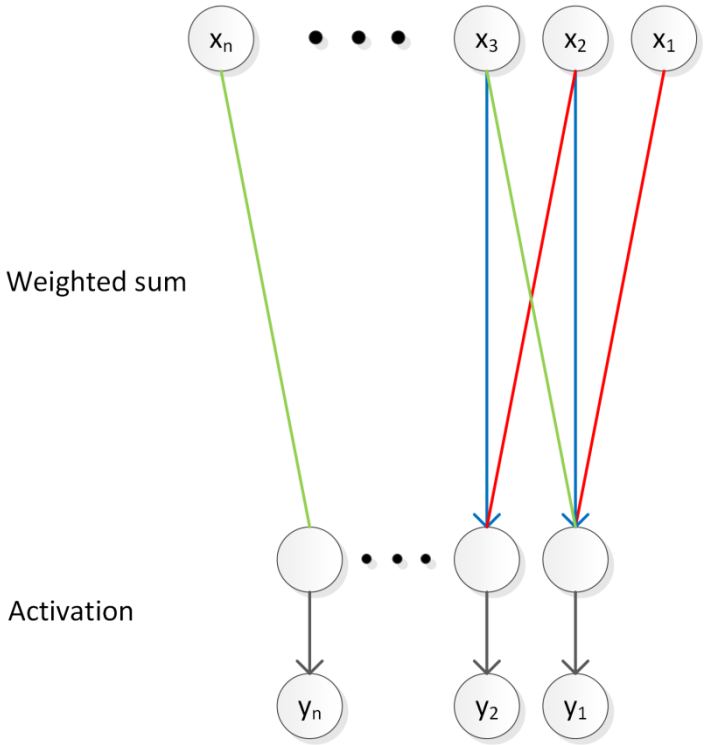
- Intuition
  - Correlation between signals
- Can also be done in multichannel images with multichannel kernels





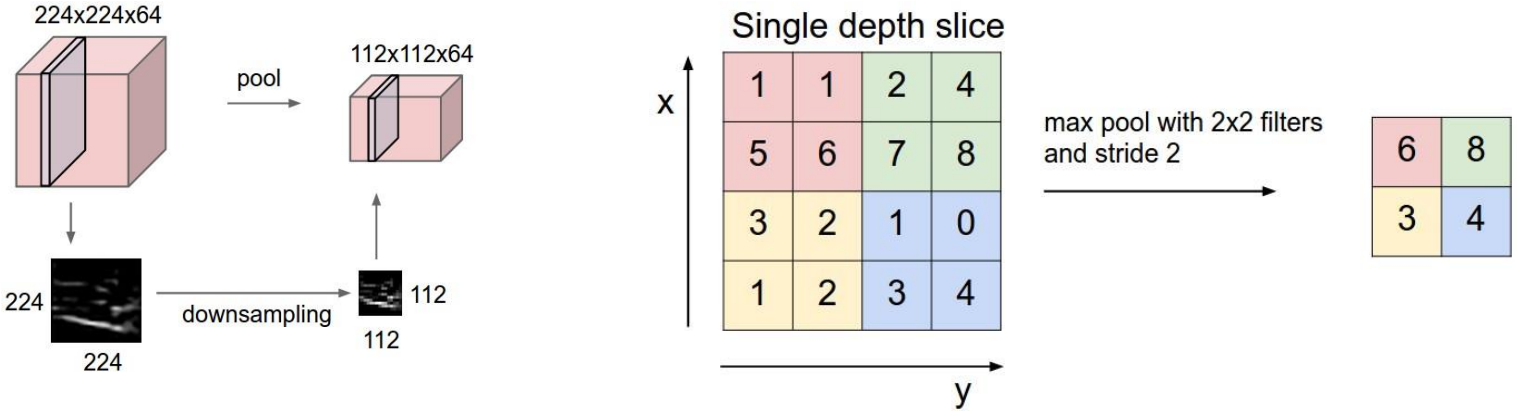
Convolution Layer

- Not a fully connected layer
- Shared weights
- Same colour indicates same (shared) weight
- Instead of learning a full matrix W mapping from every input to output, we learn a set of kernels instead (say 11x11)



Pooling layer

- Used for sub-sampling

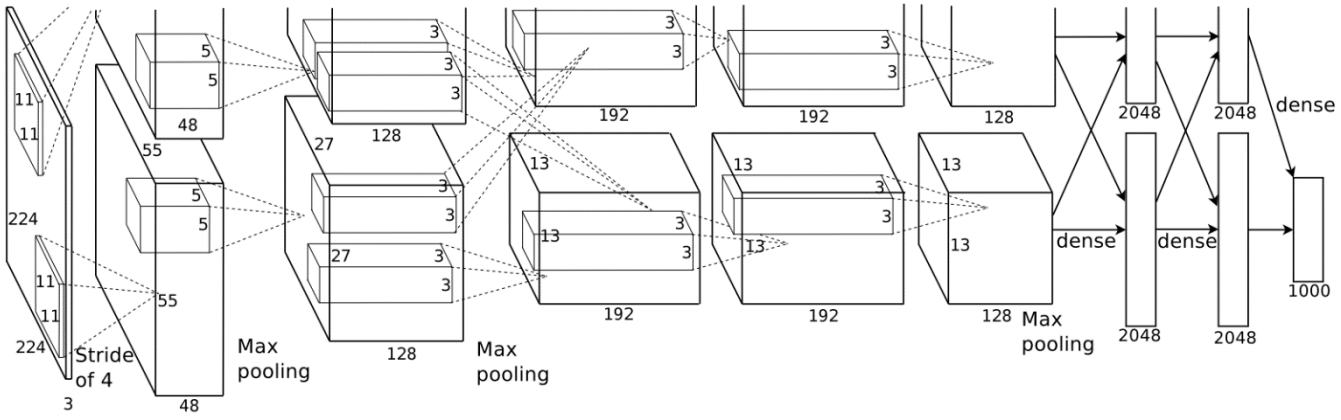


Pick the maximum value from input using a smooth and differentiable approximation

$$y = \frac{\sum_{i=1}^{n} x_i e^{\alpha x_i}}{\sum_{i=1}^{n} e^{\alpha x_i}}$$

Example: AlexNet Model

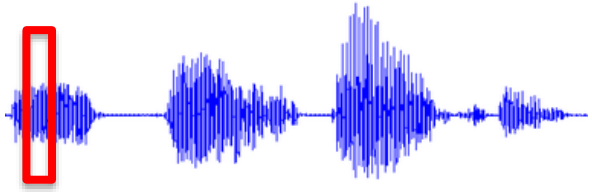
- Used for object classification task
  - 1000 way classification task – pick one
- Architecture inspired partially by how much parameters can fit on a single GPU
- In total over 61M parameters



Unimodal representations: Acoustic Modality

Unimodal Classification – Acoustic Modality

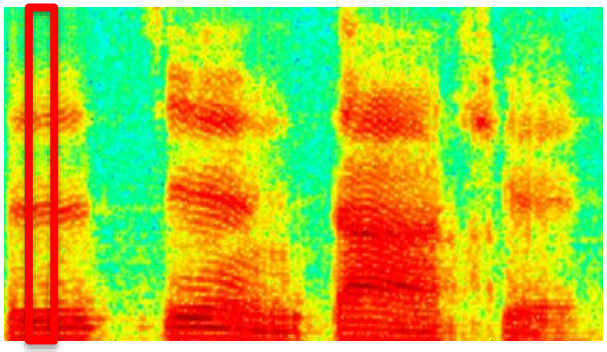
Digitalized acoustic signal



- Sampling rates: 8~96kHz
- Bit depth: 8, 16 or 24 bits
- Time window size: 20ms
  - Offset: 10ms

Input observation x_i

0.21
0.14
0.56
0.45
0.9
0.98
0.75
0.34
0.24
0.11
0.02

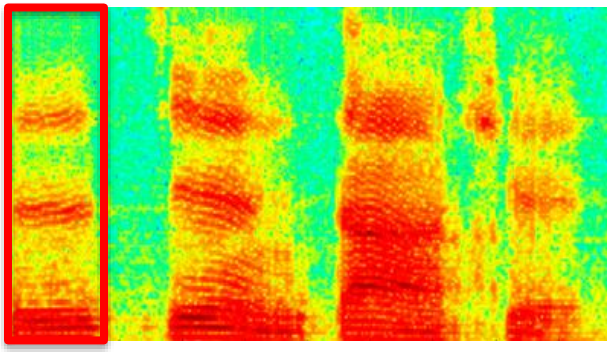


Spectogram

Unimodal Classification – Acoustic Modality

Digitalized acoustic signal

- Sampling rates: 8~96kHz
- Bit depth: 8, 16 or 24 bits
- Time window size: 20ms
  - Offset: 10ms



Spectogram

Input observation x_i

0.21
0.14
0.56
0.45
0.9
0.98
0.75
0.34
0.24
0.11
0.02
0.24
0.26
0.58
0.9
0.99
0.79
0.45
0.34
0.24
⋮

Emotion ?

Spoken word ?

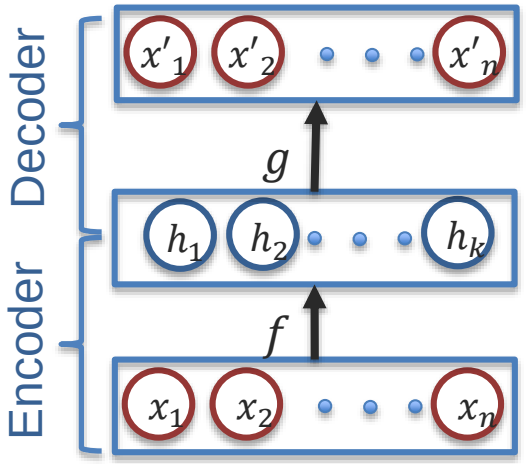
Voice quality ?

Audio representation for speech recognition

- Speech recognition systems historically much more complex than vision systems – language models, vocabularies etc.
- Large breakthrough of using representation learning instead of hand-crafted features
  - [Hinton et al., Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups, 2012]
- The field of ASR was largely static up to then
- A huge boost in performance (up to 30% on some datasets)

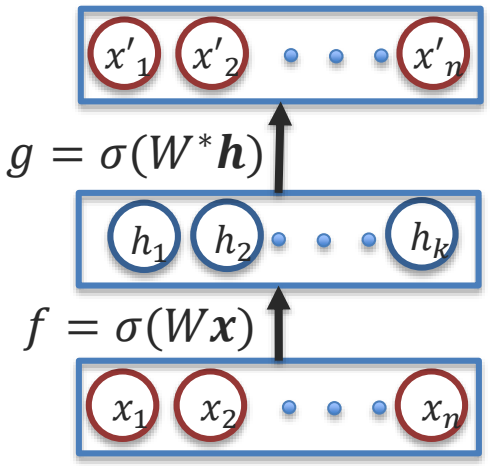
Autoencoders

- What does auto mean?
  - Greek for self – self encoding
- Feed forward network intended to reproduce the input
- Two parts encoder/decoder
  - $x' = f(g(x))$ – score function
  - g - encoder
  - f - decoder



Autoencoders

- Mostly follows Neural Network structure
  - A matrix multiplication followed by a sigmoid
- Activation will depend on type of $\boldsymbol{x}$
  - Sigmoid for binary
  - Linear for real valued
- Often we use *tied weights* to force the sharing of weights in encoder/decoder
  - $W^* = W^T$
- word2vec is actually a bit similar to autoencoder (except for the auto part)

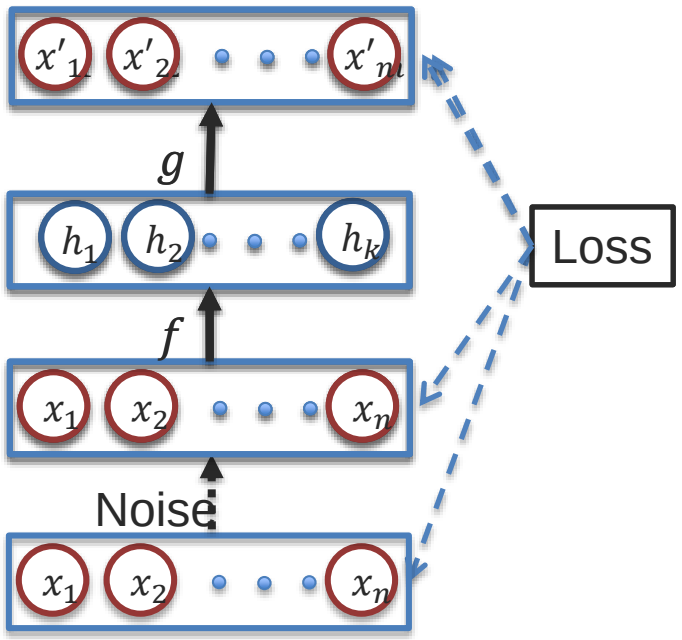


Hidden layer dimensionality

- Smaller that input - Undercomplete
  - Will compress the data, reconstruction of data far from training distribution will be difficult
  - Linear-linear encoder-decoder with Euclidean loss is actually equivalent to PCA (under certain data normalization)
- Larger than input - Overcomplete
  - No compression needed
  - Can trivially learn to just copy, so no structure is extracted
  - Not encourage to lean meaningful features, **a problem**

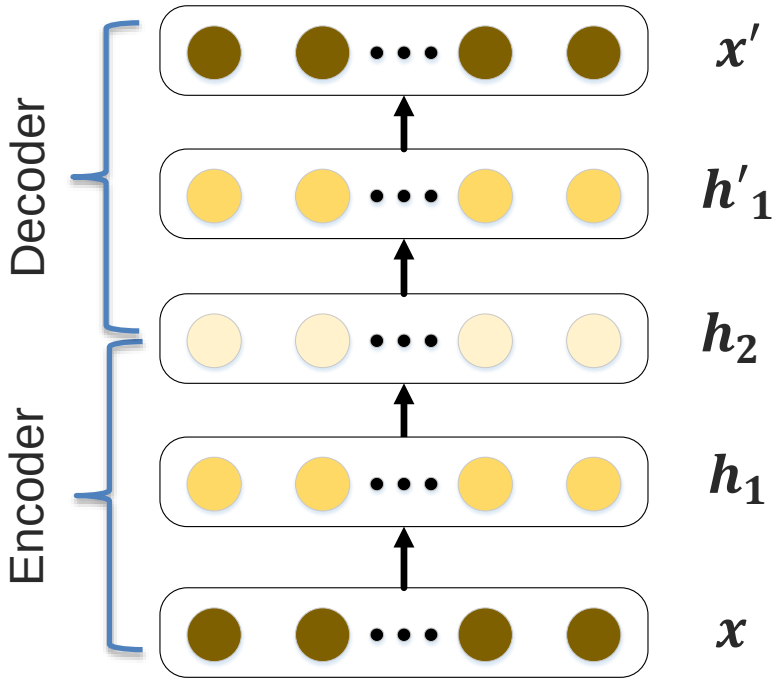
Denoising autoencoder

- Simple idea
  - Add noise to input $\boldsymbol{x}$ but learn to reconstruct original
- Leads to a more robust representation and prevents copying
- Learns what the relationship is to represent a certain $\boldsymbol{x}$
- Different noise added during each epoch



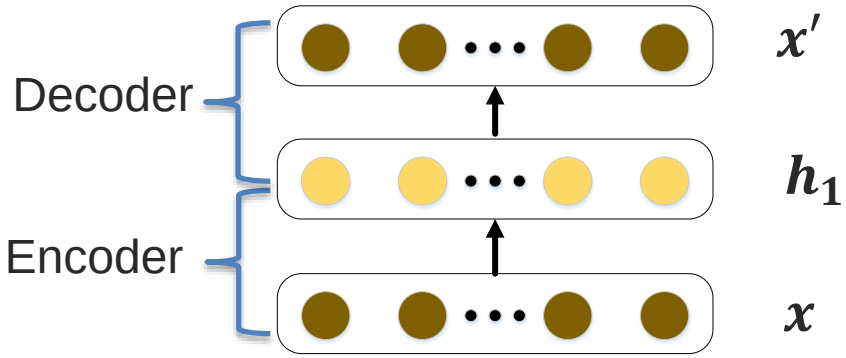
Stacked autoencoders

- Can stack autoencoders as well
- Each encoding unit has a corresponding decoder
- Inference as before is feed forward structure, but now with more hidden layers



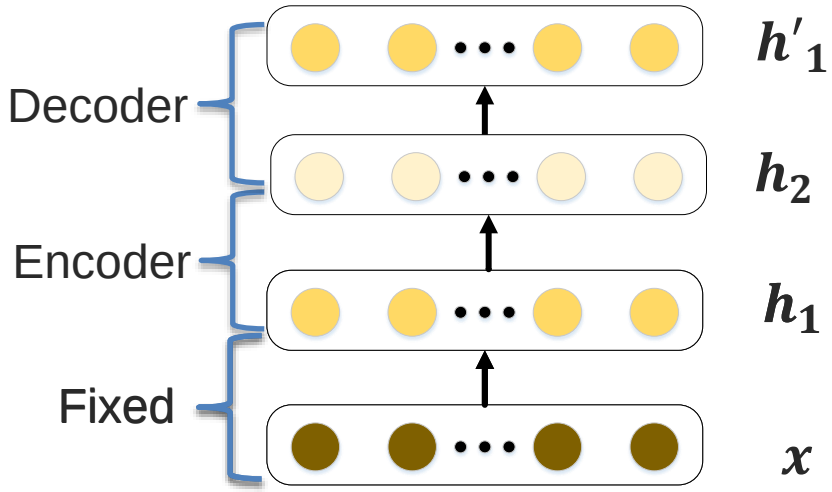
Stacked autoencoders

- Greedy layer-wise training
- Start with training first layer
  - Learn to encode $\boldsymbol{x}$ to $\boldsymbol{h_1}$ and to decode $\boldsymbol{x}$ from $\boldsymbol{h_1}$
  - Use backpropagation



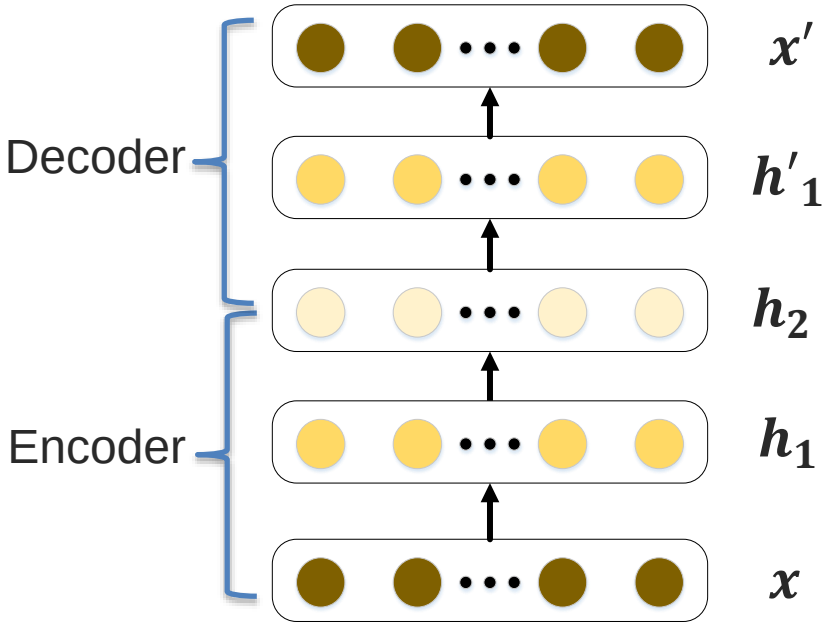
Stacked autoencoders

- Map from all $\boldsymbol{x}$'s to $\boldsymbol{h_1}$'s
  - Discard decoder for now
- Train the second layer
  - Learn to encode $\boldsymbol{h_1}$ to $\boldsymbol{h_2}$ and to decode $\boldsymbol{h_2}$ from $\boldsymbol{h_1}$
  - Repeat for as many layers



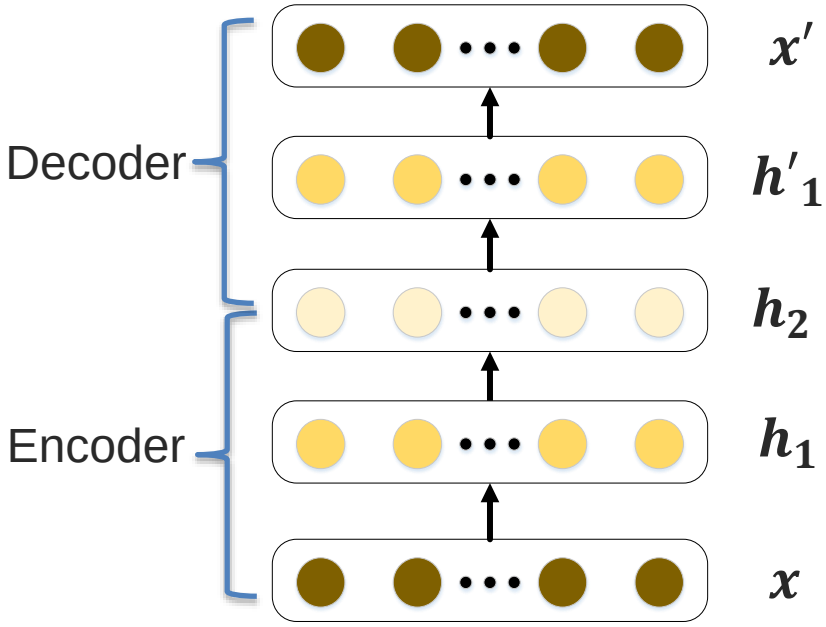
Stacked autoencoders

- Reconstruct using previously learned decoders mappings
- Fine-tune the full network end-to-end



Stacked denoising autoencoders

- Can extend this to a denoising model
- Add noise when training each of the layers
  - Often with increasing amount of noise per layer
  - 0.1 for first, 0.2 for second, 0.3 for third



Multimodal Representations

Multimodal Representations

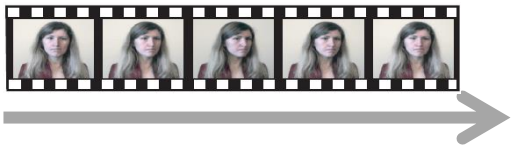
Heterogonous data:

- **Verbal modality**

We saw the yellow dog

- **Vocal modality**

- **Visual modality**



Representation:

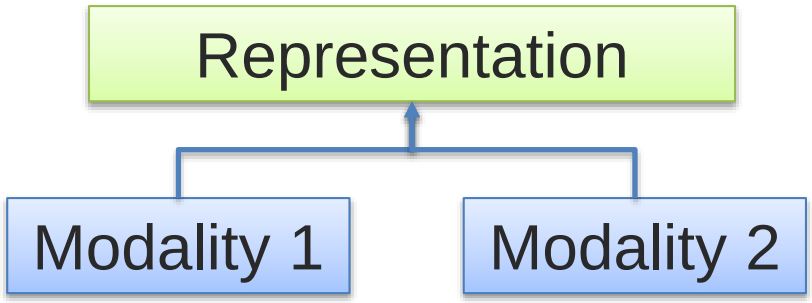
"Computer interpretable description of the multimodal data (e.g., vector, tensor)"

Challenges:

I. Symbols and signals
II. Different granularities
III. Static and sequential
IV. Different noise distribution
V. Unbalanced proportions

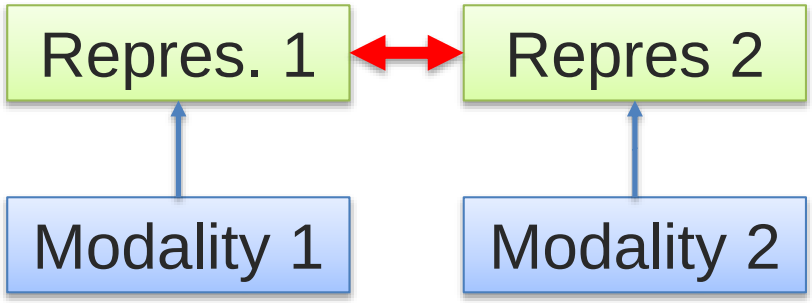
Multimodal Representations

A Joint representations:



- Simplest version: modality concatenation (early fusion)
- Can be learned supervised or unsupervised
- Multimodal factor analysis

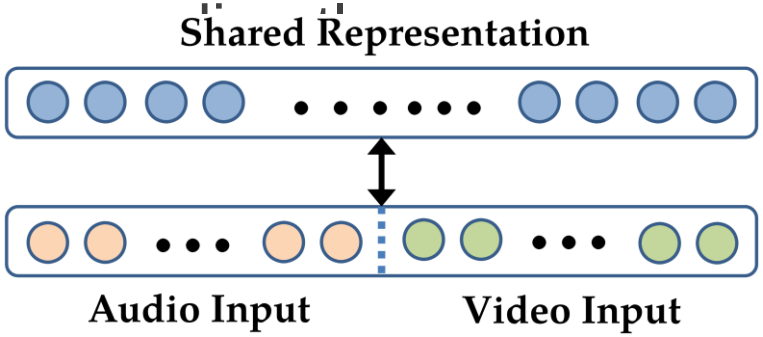
B Coordinated representations:



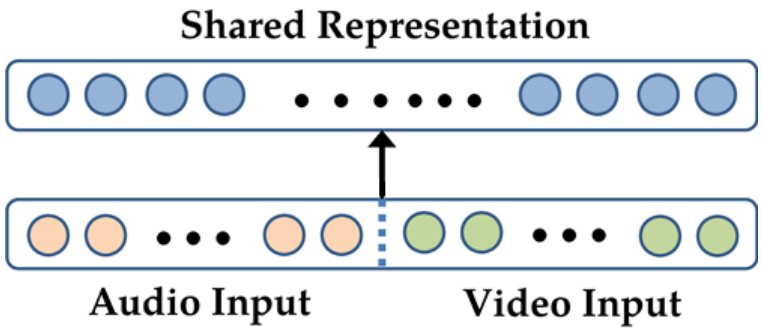
- Similarity-based methods (e.g., cosine distance)
- Orthogonality constraints (e.g., canonical correlation)

Shallow multimodal representations

- Want deep multimodal representations
  - Shallow representations do not capture complex relationships
  - Often shared layer only maps to the shared section



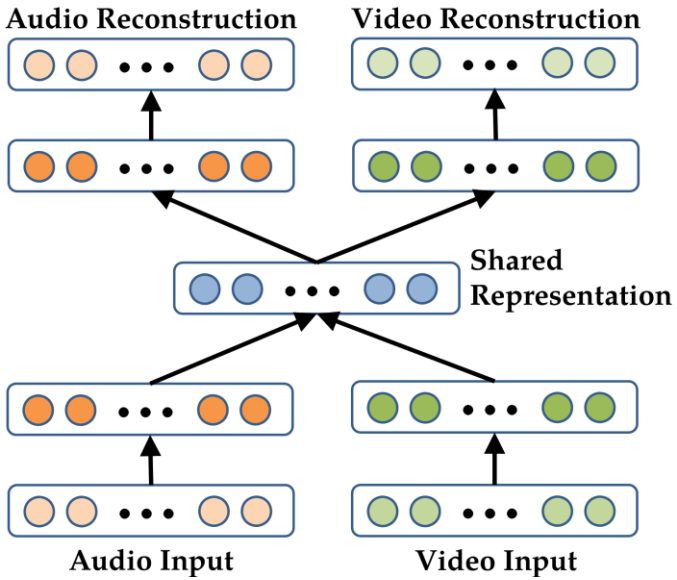
Shallow RBM



Shallow Autoencoder

Deep Multimodal autoencoders

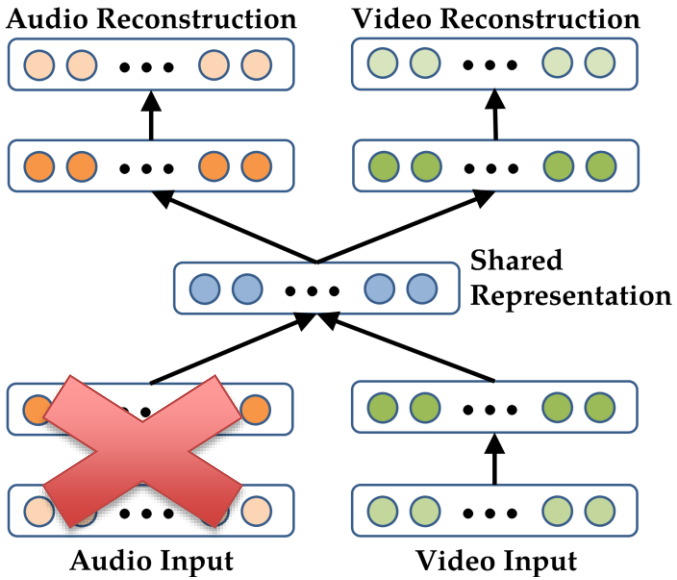
- A deep representation learning approach
- A bimodal auto-encoder
  - Used for Audio-visual speech recognition



[Ngiam et al., Multimodal Deep Learning, 2011]

Deep Multimodal autoencoders - training

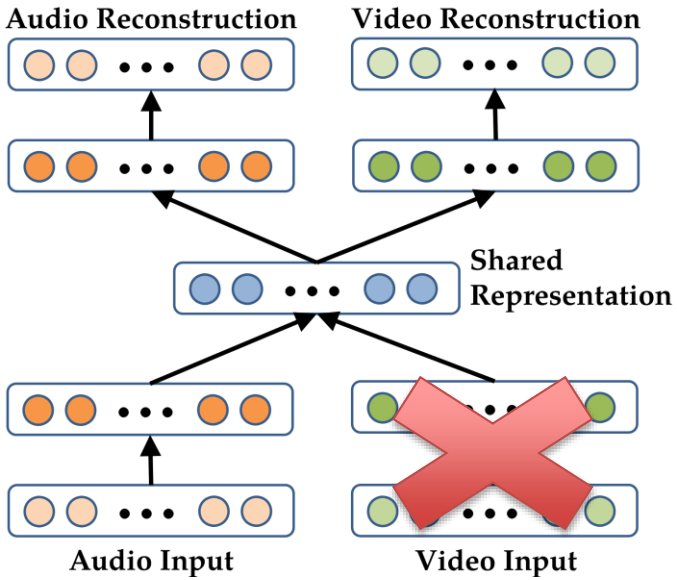
- Individual modalities can be pretrained
  - RBMs
  - Denoising Autoencoders
- To train the model to reconstruct the other modality
  - Use both
  - Remove audio



[Ngiam et al., Multimodal Deep Learning, 2011]

Deep Multimodal autoencoders - training

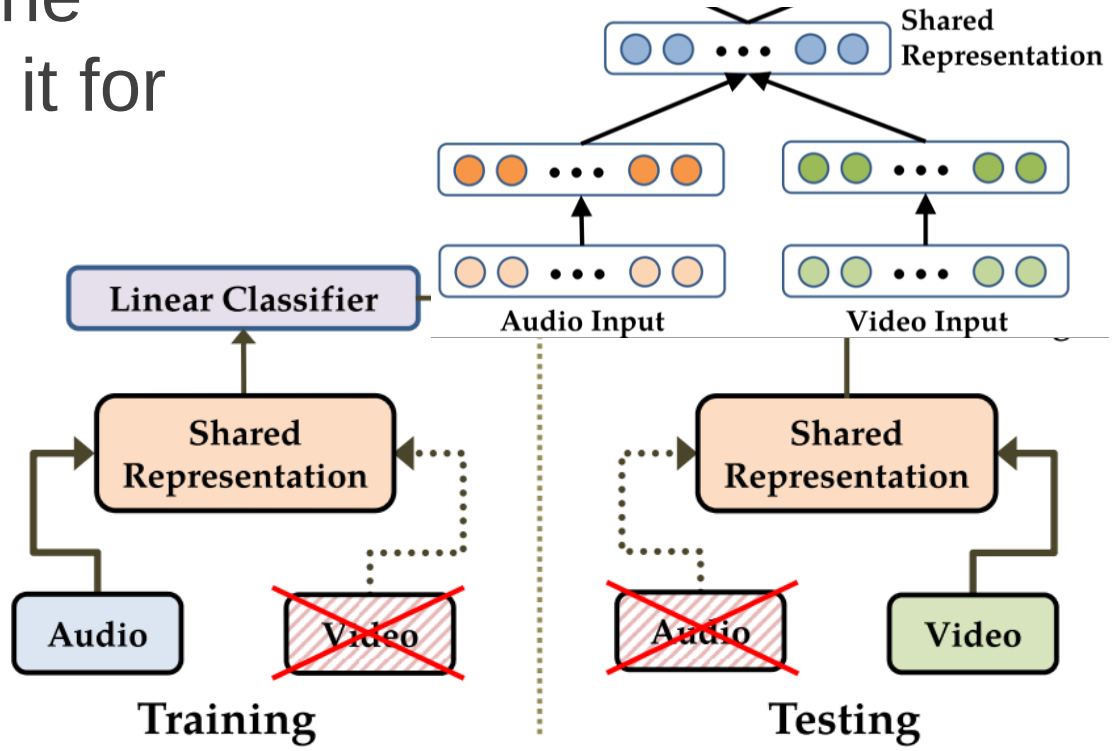
- Individual modalities can be pretrained
  - RBMs
  - Denoising Autoencoders
- To train the model to reconstruct the other modality
  - Use both
  - Remove audio
  - Remove video



[Ngiam et al., Multimodal Deep Learning, 2011]

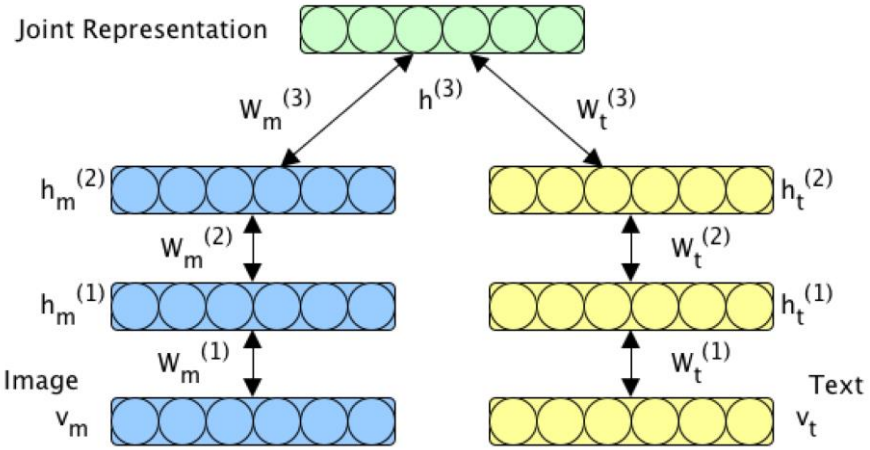
Deep Multimodal autoencoders

- Can now discard the decoder and used it for AVSR task
- Interesting experir
  - "Hearing to see"



Deep Multimodal Boltzmann machines

- Generative model
- Individual modalities trained like a DBN
- Multimodal representation trained using Variational approaches
- Used for image tagging and cross-media retrieval
- Reconstruction of one modality from another is a bit more “natural” than in autoencoder representation
- Can actually sample text and images

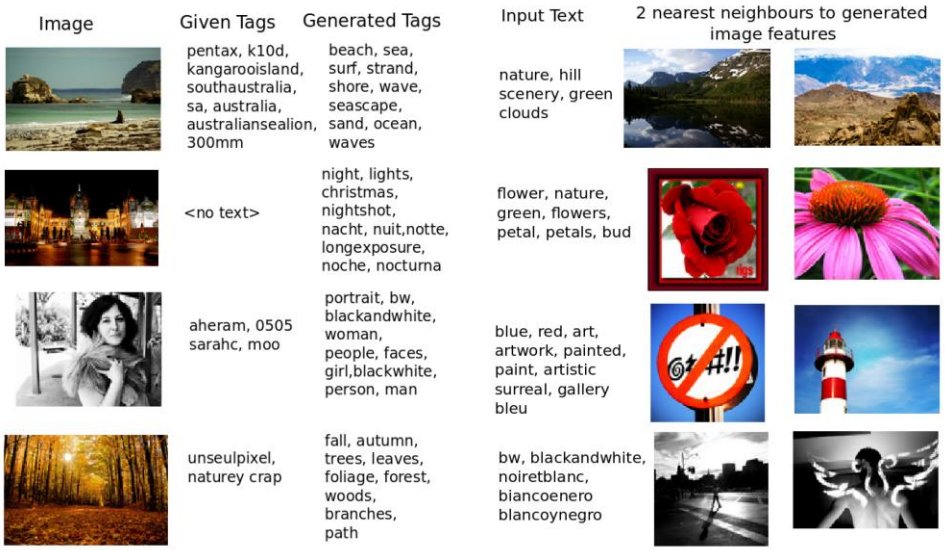


[Srivastava and Salakhutdinov, Multimodal Learning with Deep Boltzmann Machines, 2012, 2014]

Deep Multimodal Boltzmann machines

- Pretraining on unlabeled data helps
- Can use generative models

Model	MAP	Prec@50
Random	0.124	0.124
SVM (Huiskes et al., 2010)	0.475	0.758
LDA (Huiskes et al., 2010)	0.492	0.754
DBM	0.526 ± 0.007	0.791 ± 0.008
DBM (using unlabelled data)	**0.585** ± 0.004	**0.836** ± 0.004



- Code is available
  - http://www.cs.toronto.edu/~nitish/multimodal/

Srivastava and Salakhutdinov, “Multimodal Learning with Deep Boltzmann Machines”, NIPS 2012

Comparing deep multimodal representations

- Difference between them and the RBMs and the autoencoders
- Overall very similar behaviour

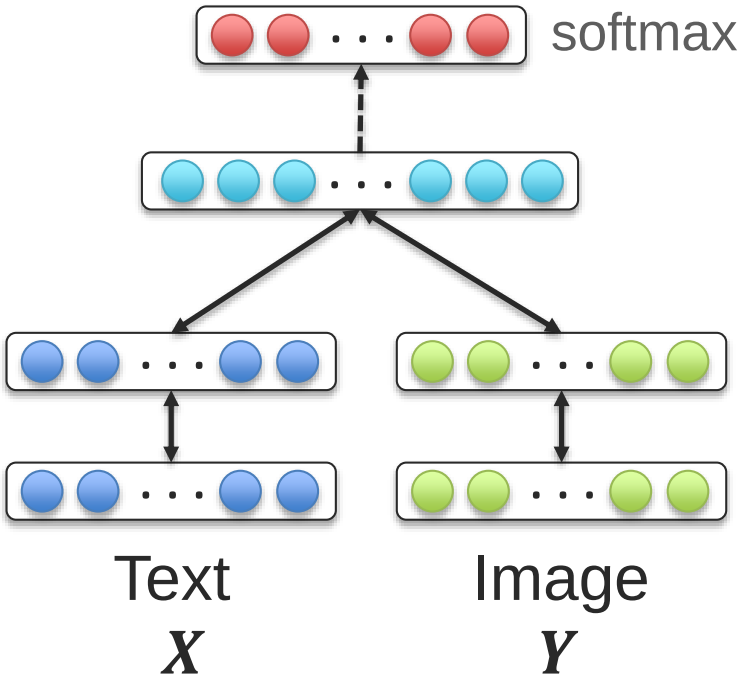
Model	DBN	DAE	DBM
Logistic regression on joint layer features	0.599 ± 0.004	0.600 ± 0.004	0.609 ± 0.004
Sparsity + Logistic regression on joint layer features	0.626 ± 0.003	0.628 ± 0.004	0.631 ± 0.004
Sparsity + discriminative fine-tuning	0.630 ± 0.004	0.630 ± 0.003	0.634 ± 0.004
Sparsity + discriminative fine-tuning + dropout	0.638 ± 0.004	0.638 ± 0.004	$\mathbf{0.641 \pm 0.004}$

Srivastava and Salakhutdinov, “Multimodal Learning with Deep Boltzmann Machines”, NIPS 2012

Recap - Multimodal Representation Learning

Learn (unsupervised) a joint representation between multiple modalities where similar unimodal concepts are closely projected.

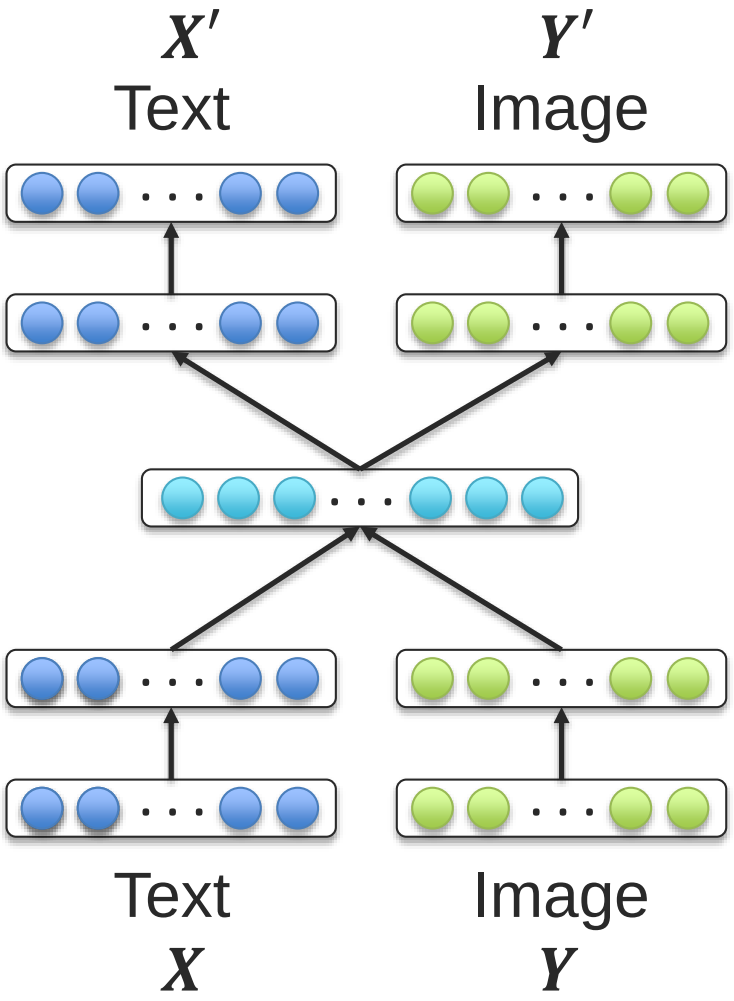
- Deep Multimodal Boltzmann machines



Recap - Multimodal Representation Learning

Learn (unsupervised) a joint representation between multiple modalities where similar unimodal concepts are closely projected.

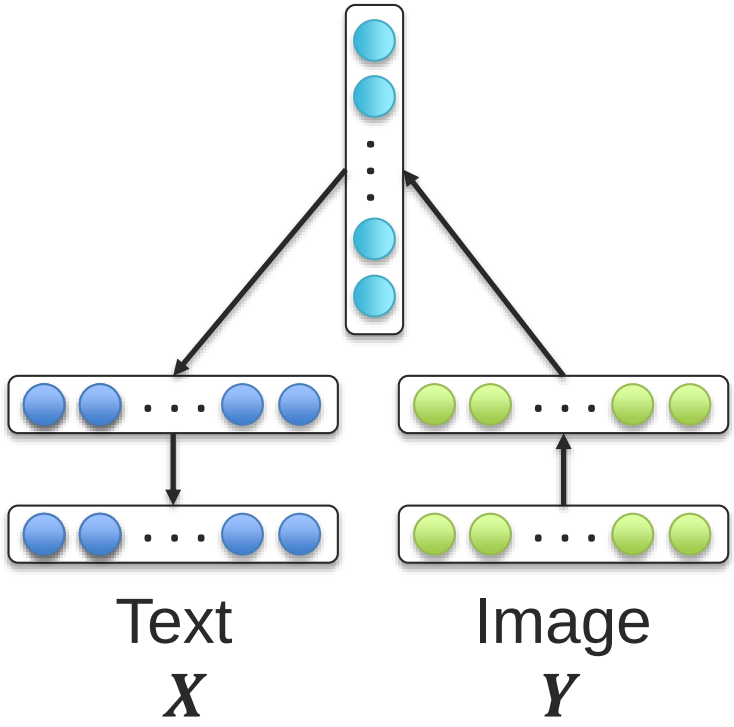
- Deep Multimodal Boltzmann machines
- Stacked Autoencoder



Recap - Multimodal Representation Learning

Learn (unsupervised) a joint representation between multiple modalities where similar unimodal concepts are closely projected.

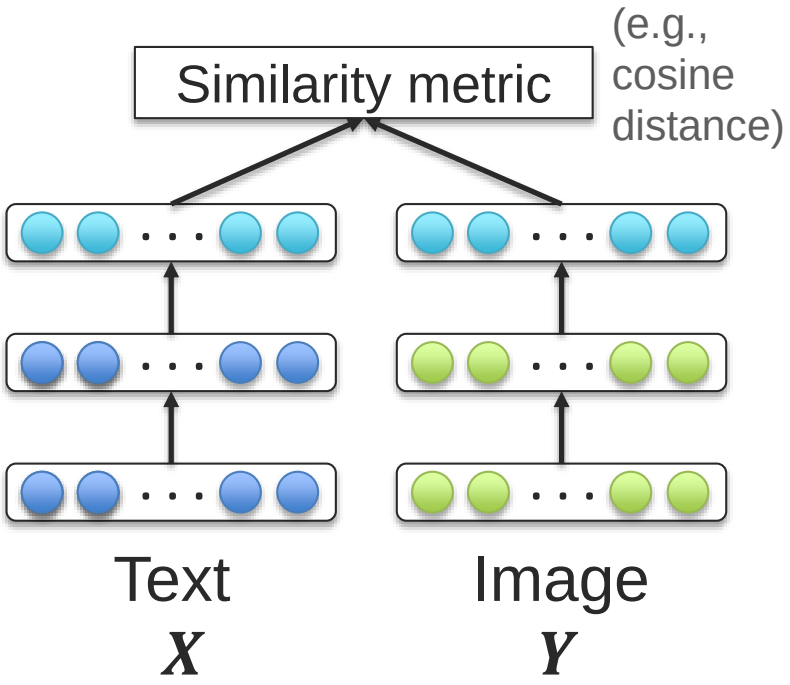
- Deep Multimodal Boltzmann machines
- Stacked Autoencoder
- Encoder-Decoder
  (to be discussed in details during Part 2 of the tutorial)



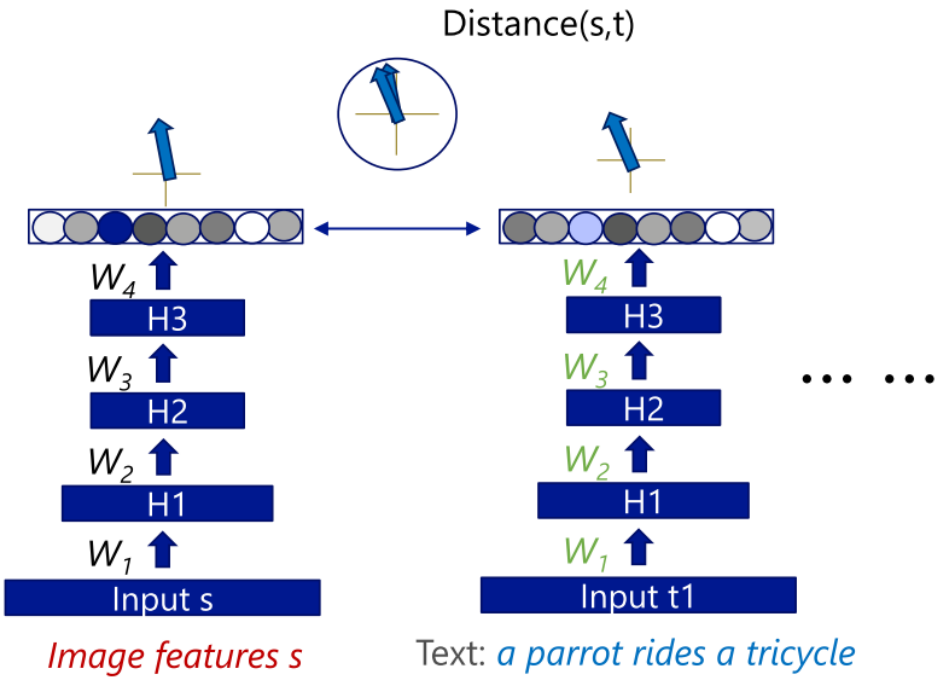
Coordinated Multimodal Representations

Coordinated Multimodal Representations

Learn (unsupervised) two or more coordinated representations from multiple modalities. A loss function is defined to bring closer these multiple representations.

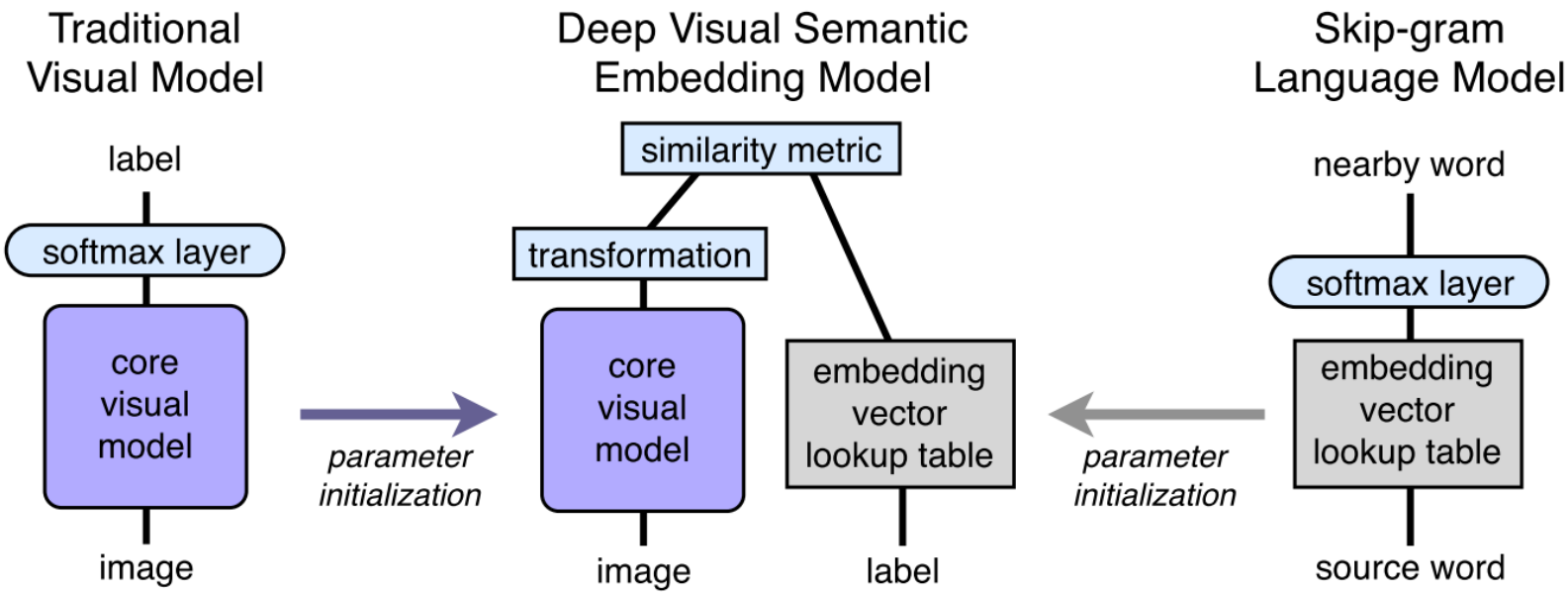


Coordinated Multimodal Embeddings



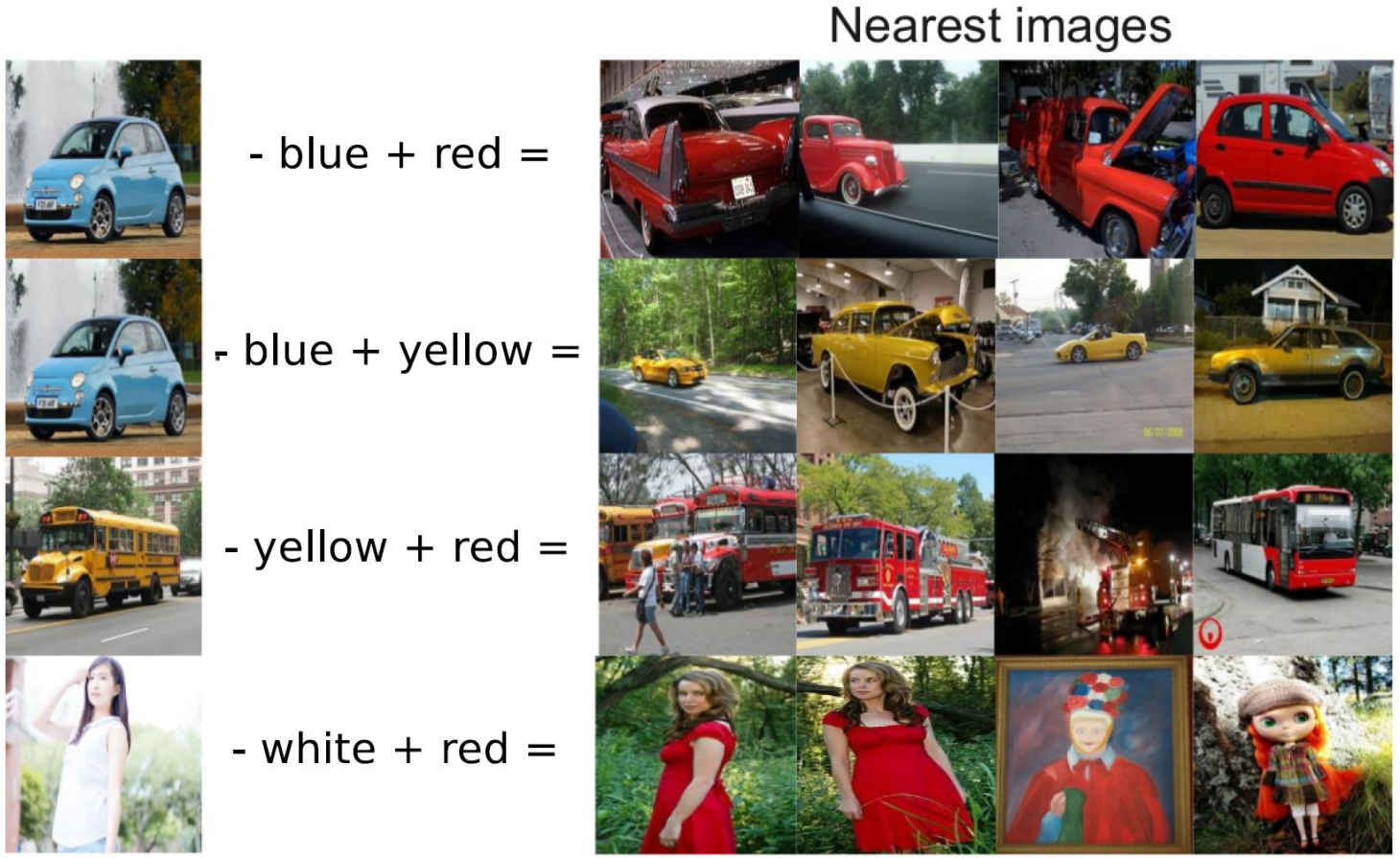
[Huang et al., Learning Deep Structured Semantic Models for Web Search using Clickthrough Data, 2013]

Coordinated Multimodal embeddings



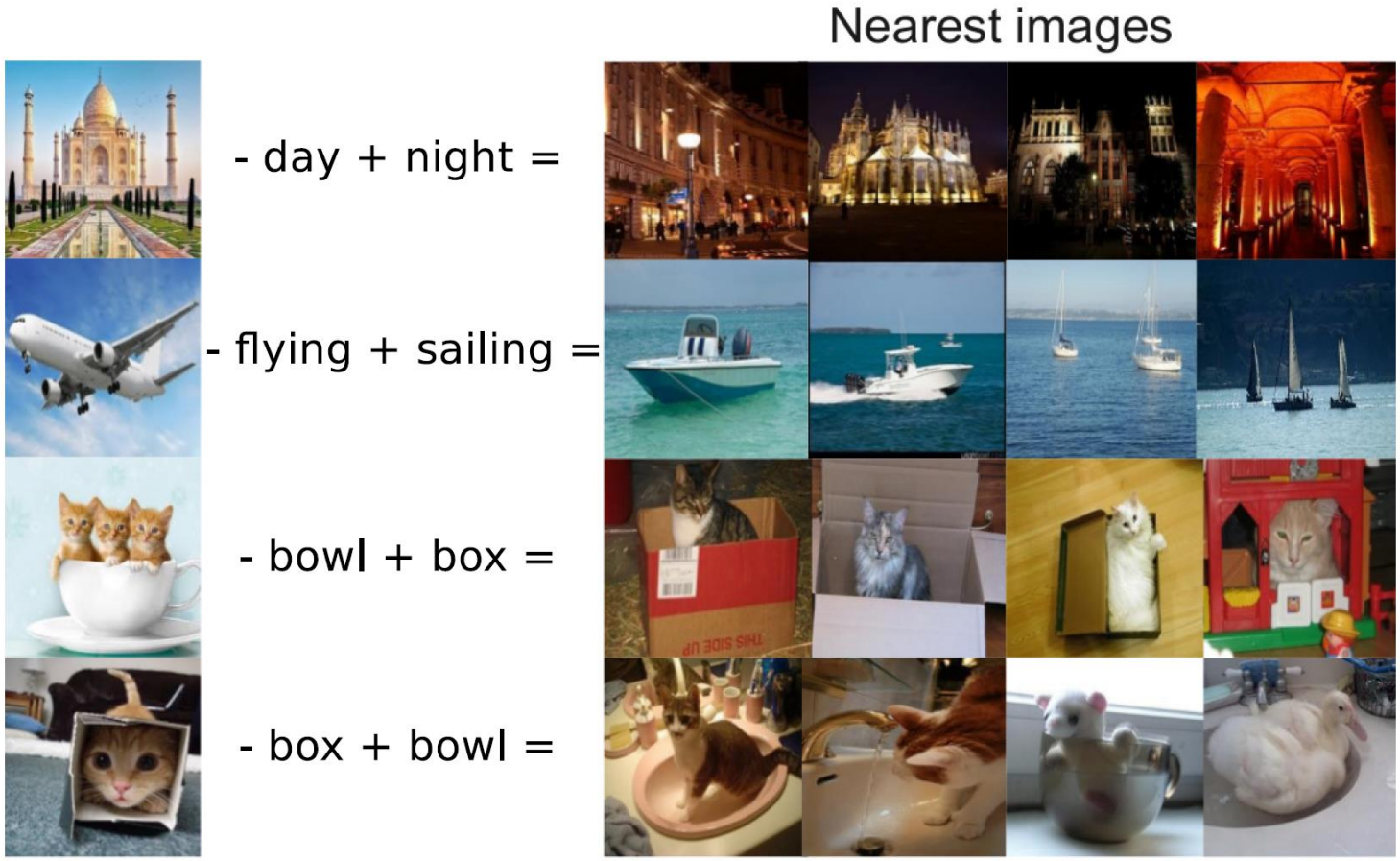
[Frome et al., DeViSE: A Deep Visual-Semantic Embedding Model, 2013]

Multimodal Vector Space Arithmetic



[Kiros et al., Unifying Visual-Semantic Embeddings with Multimodal Neural Language Models, 2014]

Multimodal Vector Space Arithmetic



[Kiros et al., Unifying Visual-Semantic Embeddings with Multimodal Neural Language Models, 2014]

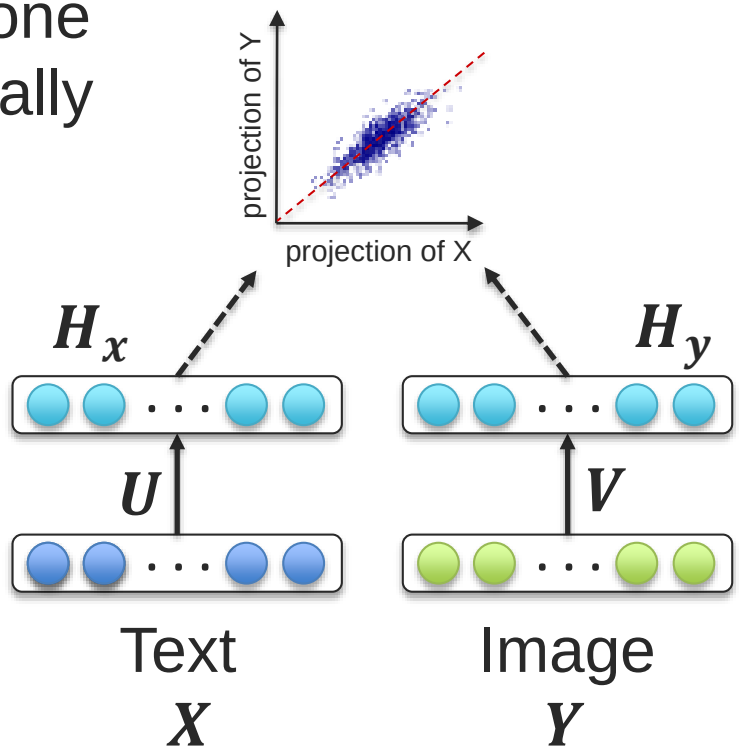
Canonical Correlation Analysis

"canonical": reduced to the simplest or clearest schema possible

1. Learn two linear projections, one for each view, that are maximally correlated:

$$(\boldsymbol{u}^*, \boldsymbol{v}^*) = \underset{\boldsymbol{u},\boldsymbol{v}}{\operatorname{argmax}}\, corr(\boldsymbol{H_x}, \boldsymbol{H_y})$$

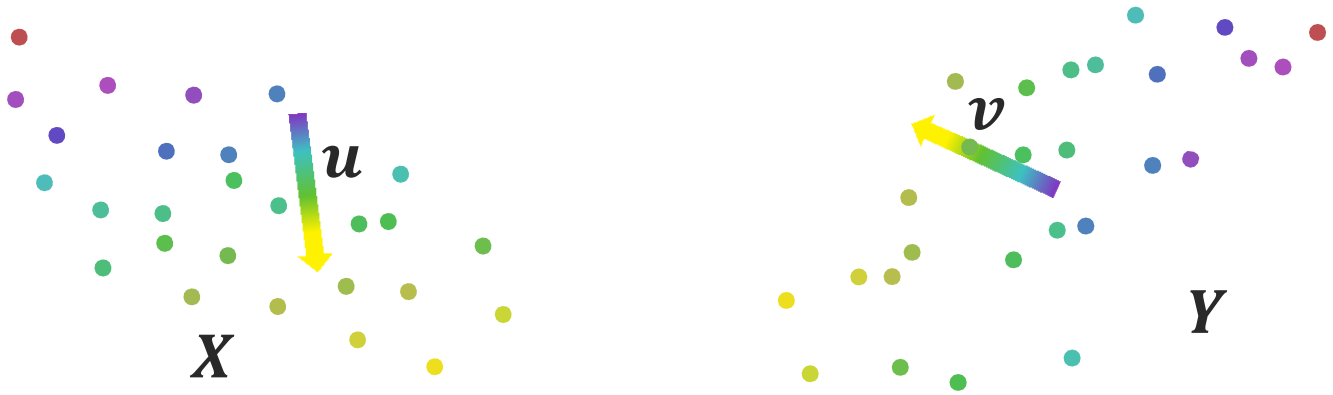
$$= \underset{\boldsymbol{u},\boldsymbol{v}}{\operatorname{argmax}}\, corr(\boldsymbol{u^T X}, \boldsymbol{v^T Y})$$



Correlated Projection

1. Learn two linear projections, one for each view, that are maximally correlated:

$$(\boldsymbol{u}^*, \boldsymbol{v}^*) = \underset{\boldsymbol{u},\boldsymbol{v}}{\operatorname{argmax}}\, corr\left(\boldsymbol{u}^T\boldsymbol{X}, \boldsymbol{v}^T\boldsymbol{Y}\right)$$



Two views $\boldsymbol{X}, \boldsymbol{Y}$ where same instances have the same color

Canonical Correlation Analysis

We want to learn multiple projection pairs $\left(\boldsymbol{u}_{(i)}\boldsymbol{X}, \boldsymbol{v}_{(i)}\boldsymbol{Y}\right)$:

$$\left(\boldsymbol{u}^*_{(i)}, \boldsymbol{v}^*_{(i)}\right) = \underset{\boldsymbol{u}_{(i)}, \boldsymbol{v}_{(i)}}{\operatorname{argmax}} corr\left(\boldsymbol{u}^{\boldsymbol{T}}_{(i)}\boldsymbol{X}, \boldsymbol{v}^{\boldsymbol{T}}_{(i)}\boldsymbol{Y}\right) \quad \approx \boldsymbol{u}^{\boldsymbol{T}}_{(i)}\boldsymbol{\Sigma}_{\boldsymbol{XY}}\boldsymbol{v}_{(i)}$$

2 We want these multiple projection pairs to be orthogonal ("canonical") to each other:

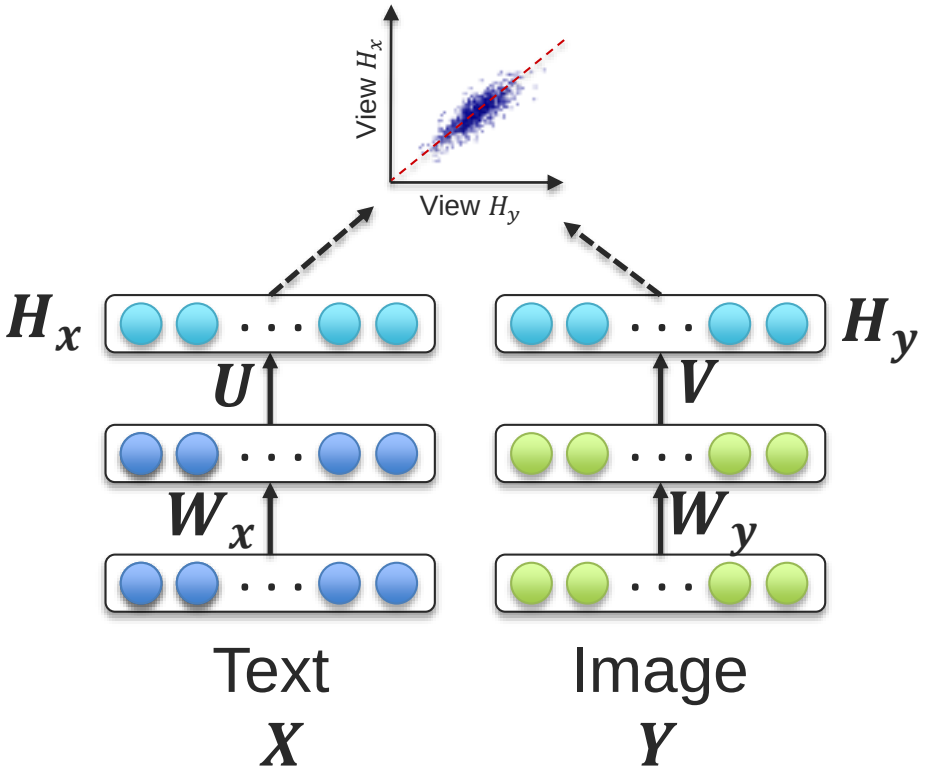
$$\boldsymbol{u}^{\boldsymbol{T}}_{(i)}\boldsymbol{\Sigma}_{\boldsymbol{XY}}\boldsymbol{v}_{(j)} = \boldsymbol{u}^{\boldsymbol{T}}_{(j)}\boldsymbol{\Sigma}_{\boldsymbol{XY}}\boldsymbol{v}_{(i)} = \boldsymbol{0} \qquad \text{for } i \neq j$$

Deep Canonical Correlation Analysis

Same objective function as CCA:

$$\underset{\boldsymbol{V},\boldsymbol{U},\boldsymbol{W_x},\boldsymbol{W_y}}{\text{argmax}}\ corr(\boldsymbol{H_x},\boldsymbol{H_y})$$

1. Linear projections maximizing correlation
2. Orthogonal projections

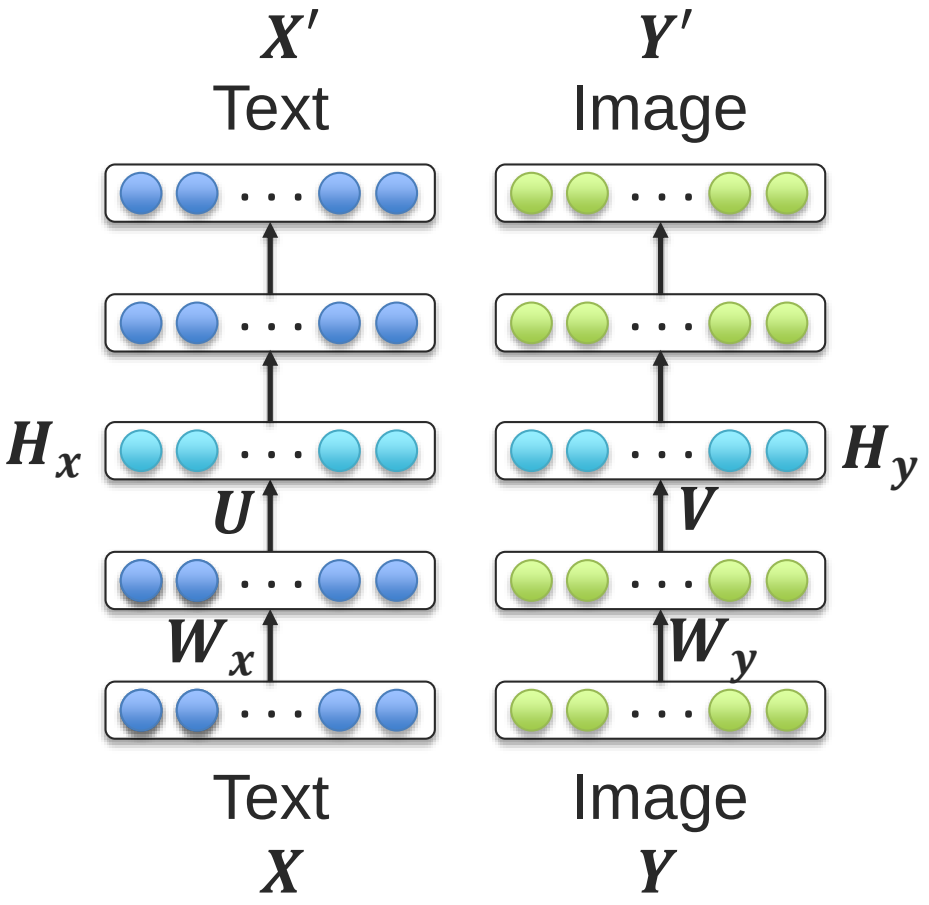


Andrew et al., ICML 2013

Deep Canonical Correlation Analysis

Training procedure:

1. Pre-train the models parameters using denoising autoencoders

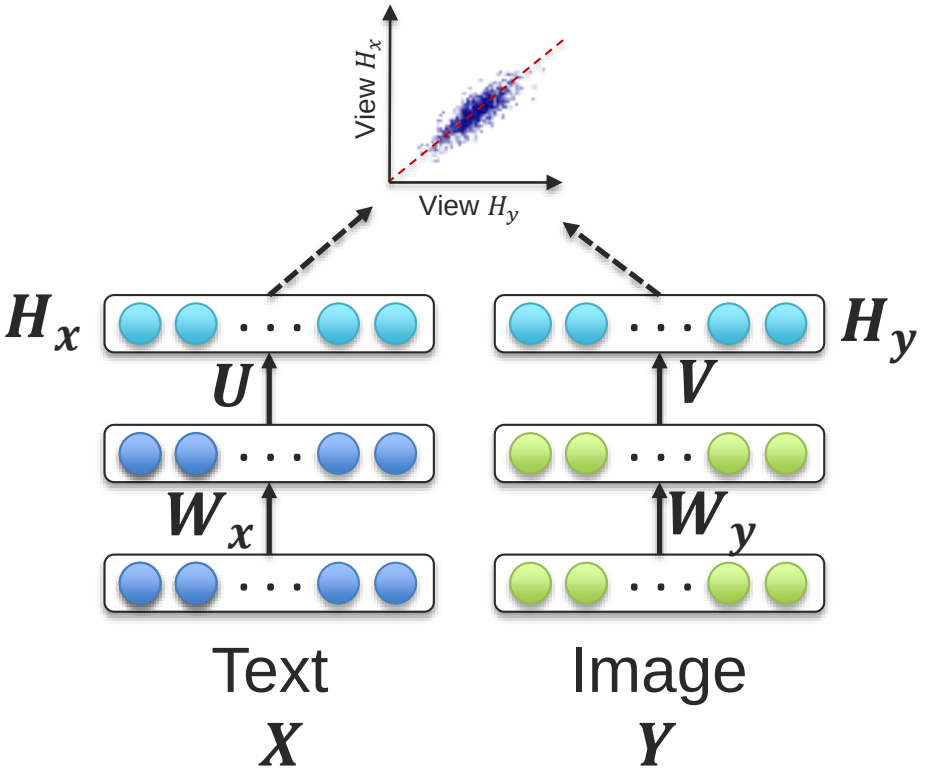


Andrew et al., ICML 2013

Deep Canonical Correlation Analysis

Training procedure:

1. Pre-train the models parameters using denoising autoencoders
2. Optimize the CCA objective functions using large mini-batches or full-batch (L-BFGS)

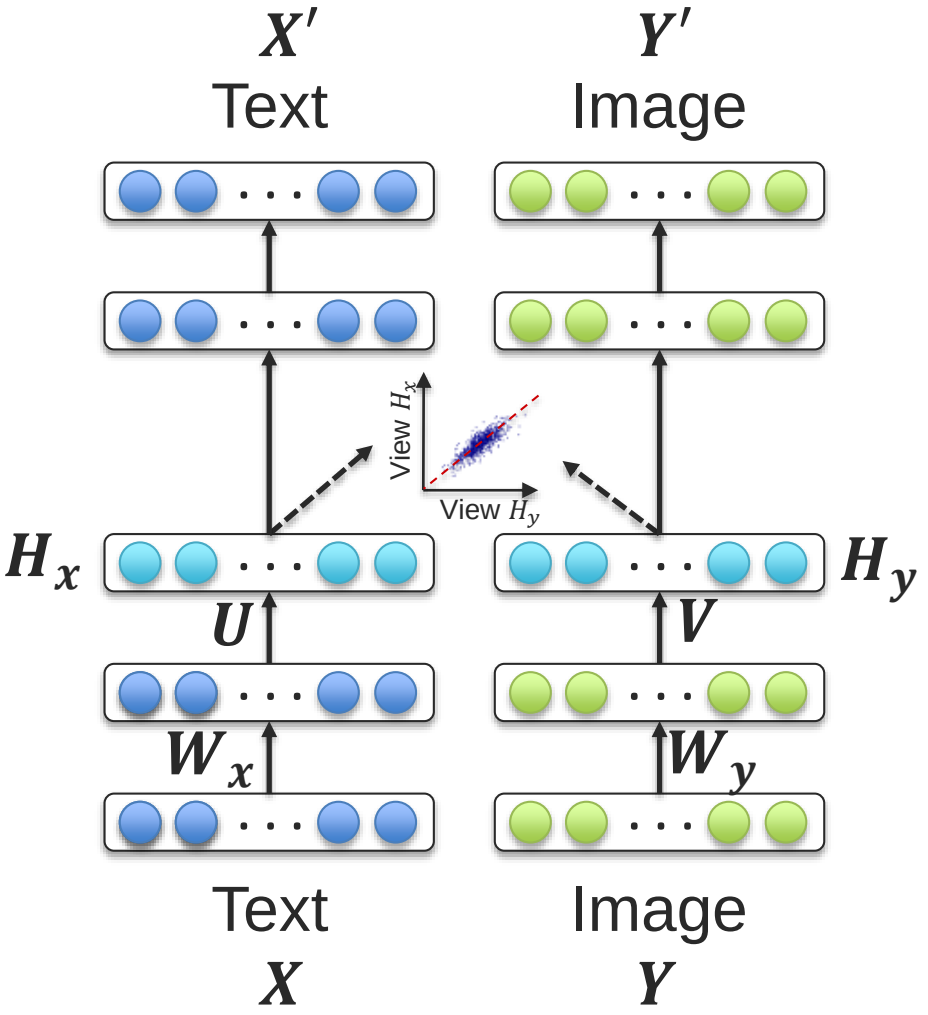


Andrew et al., ICML 2013

Deep Canonically Correlated Autoencoders (DCCAE)

Jointly optimize for DCCA and autoencoders loss functions

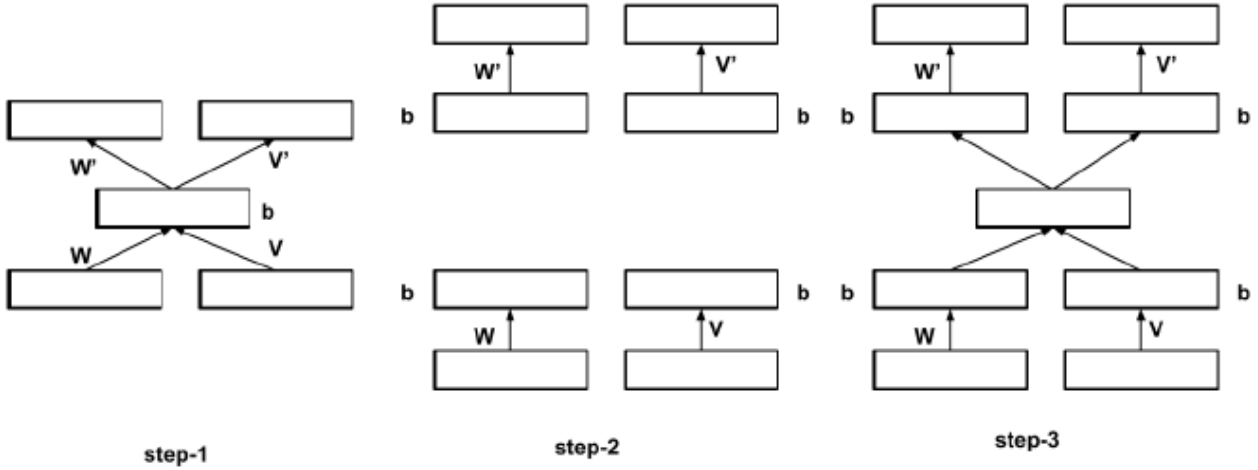
- A trade-off between multi-view correlation and reconstruction error from individual views



Wang et al., ICML 2015

Deep Correlational Neural Network

1. Learn a shallow CCA autoencoder (similar to 1 layer DCCAE model)
2. Use the learned weights for initializing the autoencoder layer
3. Repeat procedure



Chandar et al., Neural Computation, 2015