# deep learning and applications in non-cognitive domains

# part 3

**truyen tran**
deakin university
truyen.tran@deakin.edu.au
prada-research.net/~truyen

**AI'16, Hobart, Dec 5th 2016**


http://www.tocci.com/wp-content/uploads/2015/06/150623_logotrends_feature.jpg

# REVIEW OF PART I: MOSTLY SUPERVISED LEARNING

**Neural net** as function approximation & feature detector

**Three architectures**: FFN ⟶ RNN ⟶ CNN

**Bag of tricks**: dropout ⟶ piece-wise linear units ⟶ skip-connections ⟶ adaptive stochastic gradient ⟶ data augmentation

# PART III: ADVANCED TOPICS

Unsupervised learning

Complex domain structures: Relations (explicit & implicit), graphs & tensors

Memory, attention & execution

Learning to learn

How to position ourselves


# UNSUPERVISED LEARNING

# WHY NEURAL UNSUPERVISED LEARNING?

**Motivation**: Humans mainly learn by exploring without clear instructions and labelling

**Representational richness**:

- FFN are functional approximator
- RNN are program approximator, can estimate a program behaviour and generate a string
- CNN are for translation invariance

**Compactness**: Representations are (sparse and) distributed.

- Essential to perception, compact storage and reasoning

**Accounting for uncertainty**: Neural nets can be stochastic to model distributions

**Symbolic representation**: realisation through sparse activations and gating mechanisms

# APPROACHES TO UNSUPERVISED LEARNING

**Try to explain the data** e.g., learning disentangled representations

**Generative models** – generate authentic samples

Optimizing some objective functions (may be more than one, may not be likelihood)

Preserve some quantities (volumes, variances, flow, local probabilities etc)

Manifold assumption: intrinsic dimensions are smaller and locally linear/smooth

….

Exploiting the structure of the world, e.g., smoothness, predictiveness, locality.

# OBJECTIVE FUNCTIONS FOR UNSUPERVISED LEARNING

Data likelihood – classic (RBM, VAE)

Prediction-like:

- Auto-encoding: predicting the data itself
- Pseudo-likelihood: One variable (subset) given the rest. With and without variable ordering.
- Predict whether the input comes from the data generating distribution or some other distribution (as a probabilistic classifier) (Noise-Constrastive Estimation)

Others

- Learn an invertible function such that the transformed distribution is as factorial as possible (NICE, and when considering approximately invertible functions, the variational autoencoders)
- Learn a stochastic transformation so that if we were to apply it many times we would converge to something close to the data generating distribution (Generative Stochastic Networks, generative denoising autoencoders, diffusion inversion = nonequilibrium thermodynamics)
- Learn to generate samples that cannot be distinguished by a classifier from the training samples (GAN = generative adversarial networks)

https://www.quora.com/Yoshua-Bengio-Apart-from-trying-to-reconstruct-the-input-as-in-Autoencoder-what-other-tasks-could-prove-useful-for-unsupervised-learning-of-deep-networks

# PREDICTING NEIGHBOURS AND THEIR POSITIONS

Word embedding with skip-grams is a kind of pseudo-likelihood within a sliding window (Mikolov et al, 2013)

Language models – predicting the next word using RNN/LSTM (Mikolov, 2012)

Pixel RNN (van den Oord et al, ICML'16): predicting next pixel

NADE (Larochelle et al, AISTATS'11, JMLR'16): predicting next variable

Multi-prediction training of DBM (Goodfellow et al, NIPS'13)

Pixel video networks (Kalchbrenner, 2016): predicting the next frame.

# UNSUPERVISED METHODS

Word embedding

Language model

Pixel RNN

RBM → DBN → DBM + {recurrent, convolution}

DAE → DDAE → Generative Stochastic Nets

Deconvolutional nets

Helmholtz machine → Variational AE

Generative Adversarial Nets (GAN)

NADE → MADE

Skip-thought

Variational RNN

Deep topic models

Sum-product networks

Deep CCA

# WE WILL BRIEFLY COVER

Word embedding

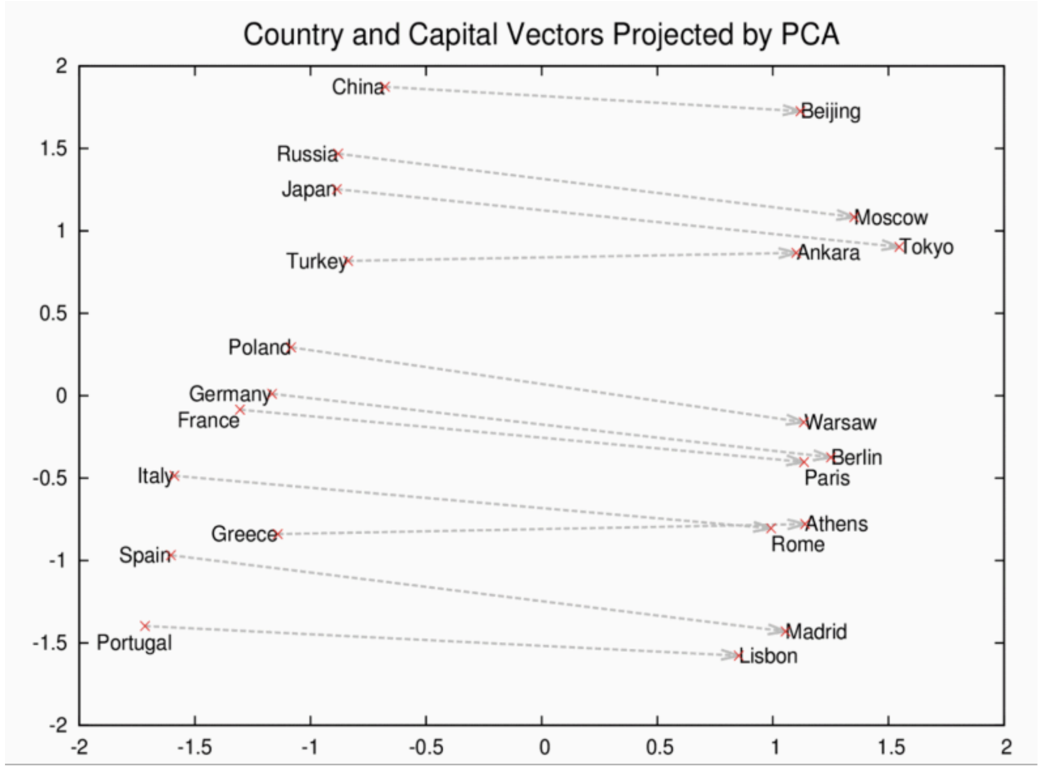
Deep autoencoder

RBM → DBN → DBM

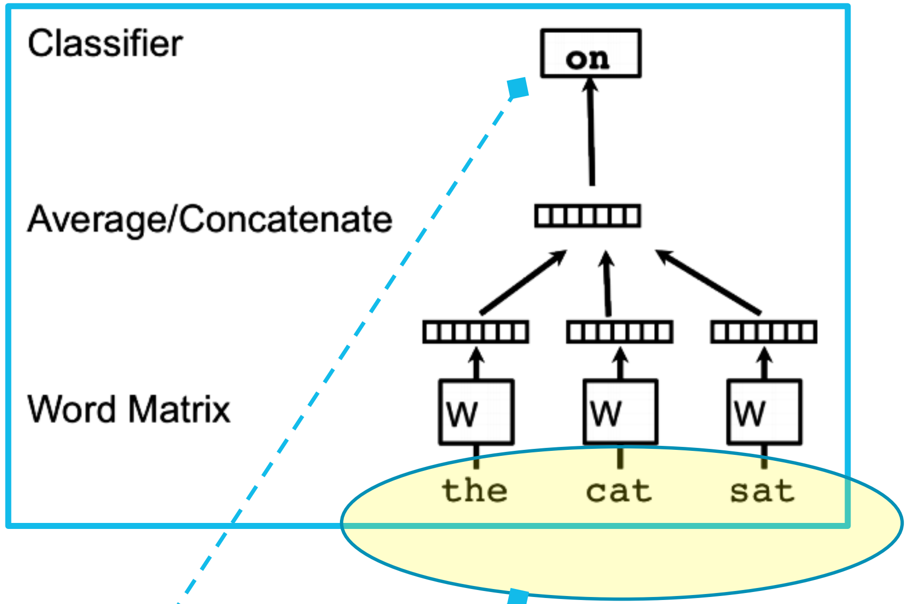
Variational AutoEncoder (VAE)

Generative Adversarial Net (GAN)

# WORD EMBEDDING



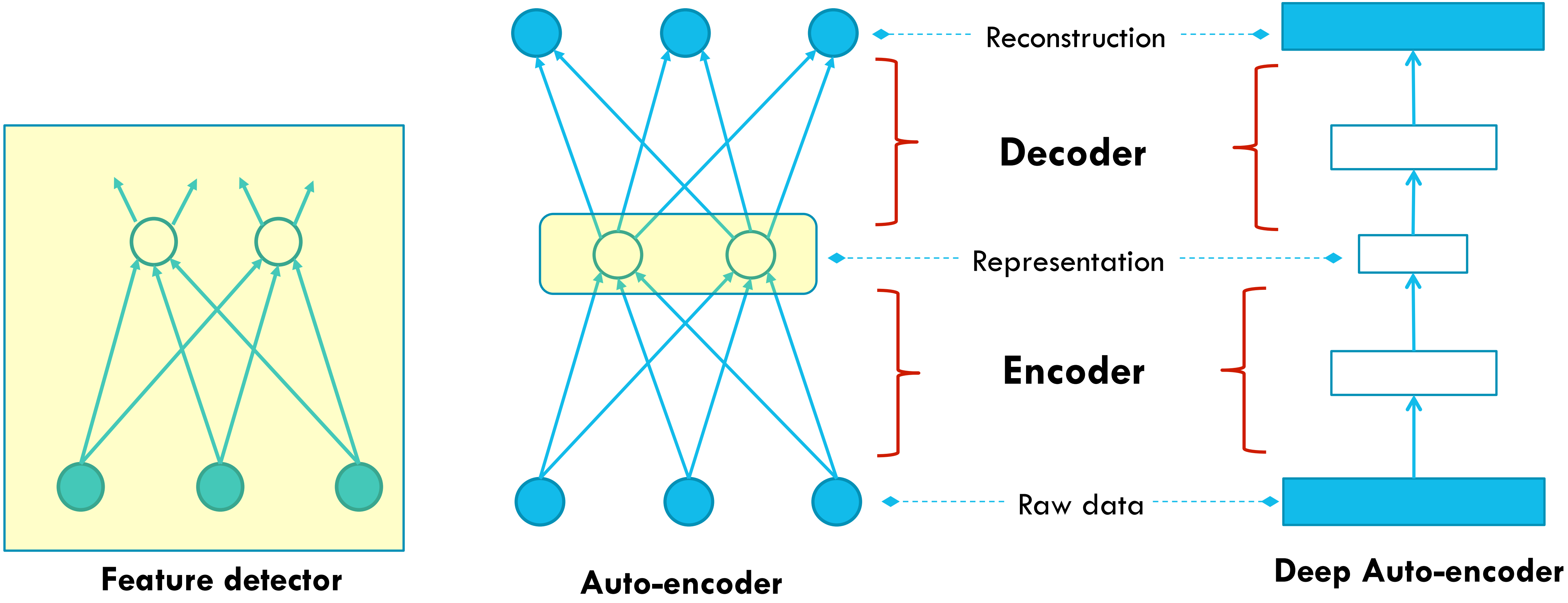

(Mikolov et al, 2013)




$$P\left(w_t \mid C_t\right) = \frac{e^{V_{w_t}^{\top} f(C_t)}}{\sum_{w \in Vocab} e^{V_w^{\top} f(C_t)}}$$

$$f(C_t) = \frac{1}{|C_t|} \sum_{w \in C_t} W_w$$

# DEEP AUTOENCODER – SELF RECONSTRUCTION OF DATA



# GENERATIVE MODELS

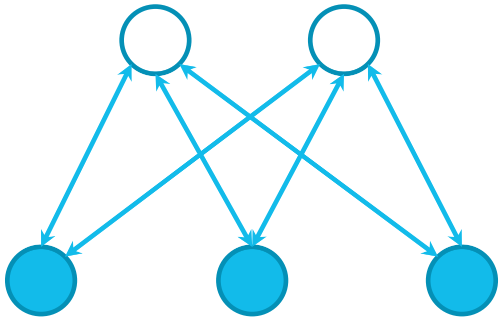
**Many applications:**

- Text to speech
- Simulate data that are hard to obtain/ share in real life (e.g., healthcare)
- Generate meaningful sentences conditioned on some input (foreign language, image, video)
- Semi-supervised learning
- Planning

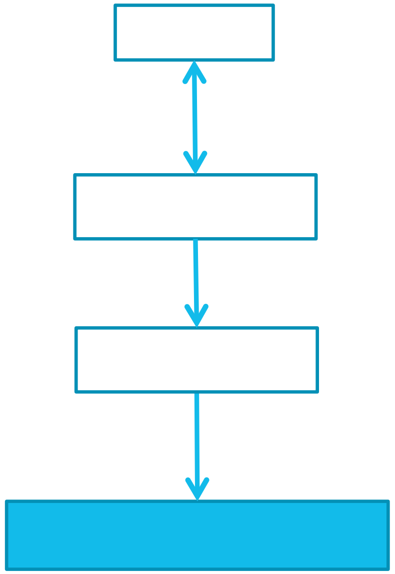
$$\mathbf{v} \sim P_{model}(\mathbf{v})$$
$$P_{model}(\mathbf{v}) \approx P_{data}(\mathbf{v})$$

# A FAMILY: RBM → DBN → DBM

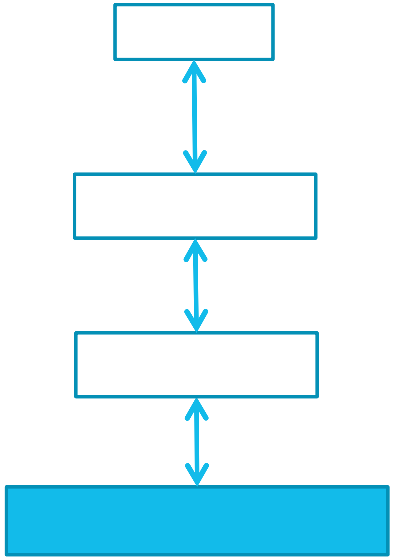
$$p(\mathbf{v},\mathbf{h};\psi) \propto \exp\left[\underbrace{-E(\mathbf{v},\mathbf{h};\psi)}_{\textit{energy}}\right]$$



**Restricted Boltzmann Machine**
**(~1994, 2001)**

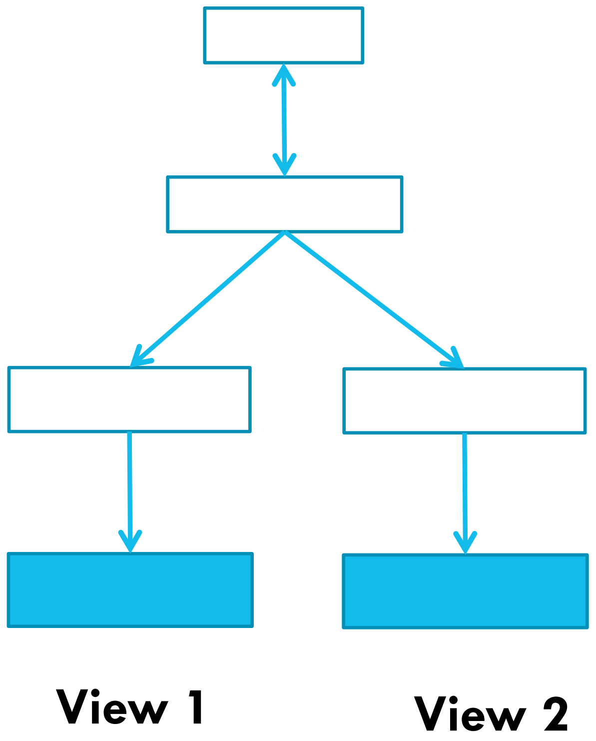


**Deep Belief Net**
**(2006)**

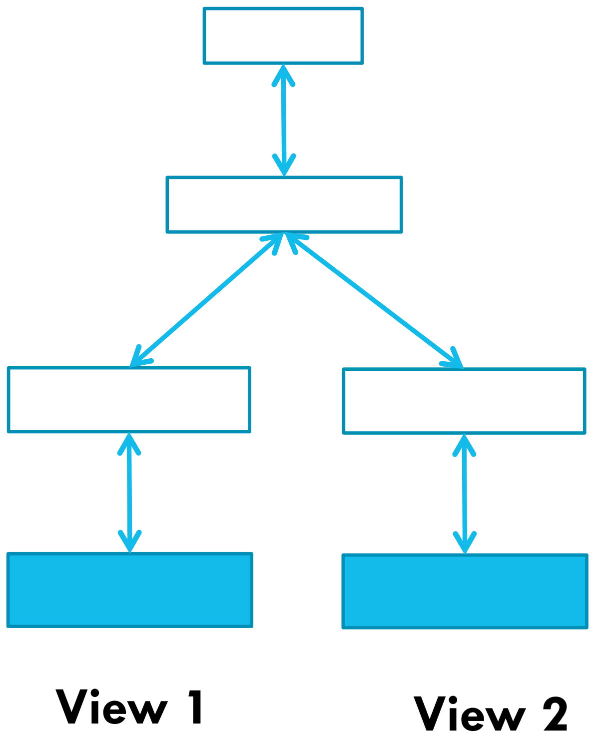


**Deep Boltzmann Machine**
**(2009)**

# APPLICATION: MULTI-MODAL/VIEW/TYPE/ PART MODELS



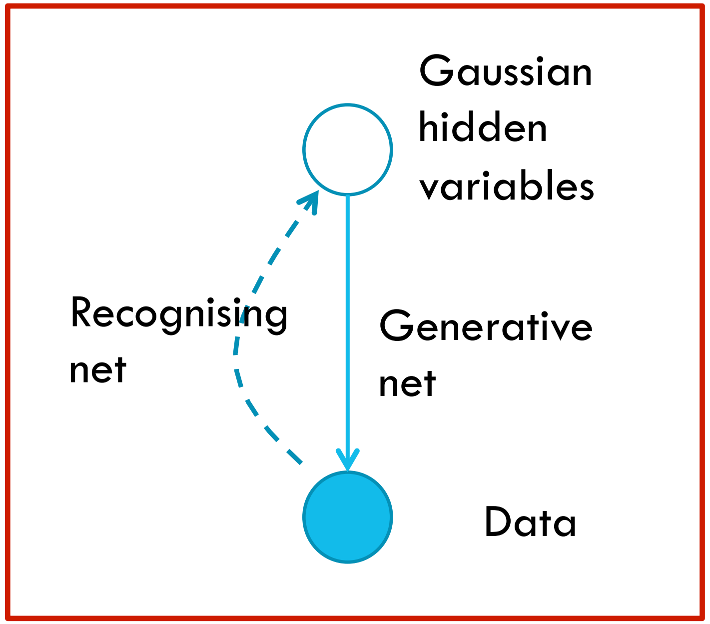

Multimodal DBN

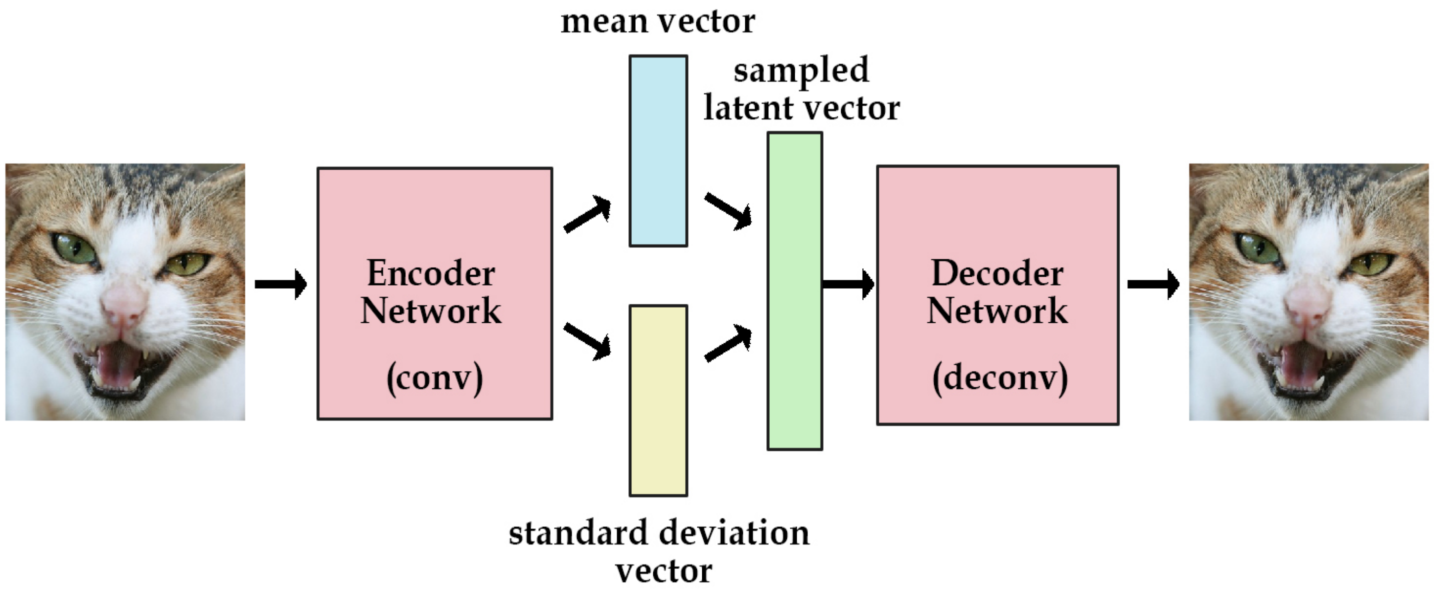



Multimodal DBM

# VARIATIONAL AUTOENCODER
## (KINGMA & WELLING, 2014)

Two separate processes: generative (hidden → visible) versus recognition (visible → hidden)




http://kvfrans.com/variational-autoencoders-explained/

# GAN: GENERATIVE ADVERSARIAL NETS
## (GOODFELLOW ET AL, 2014)

Yann LeCun: *GAN is one of best idea in past 10 years!*

*Instead of modeling the entire distribution of data, learns to map ANY random distribution into the region of data, so that **there is no discriminator that can distinguish sampled data from real data.***

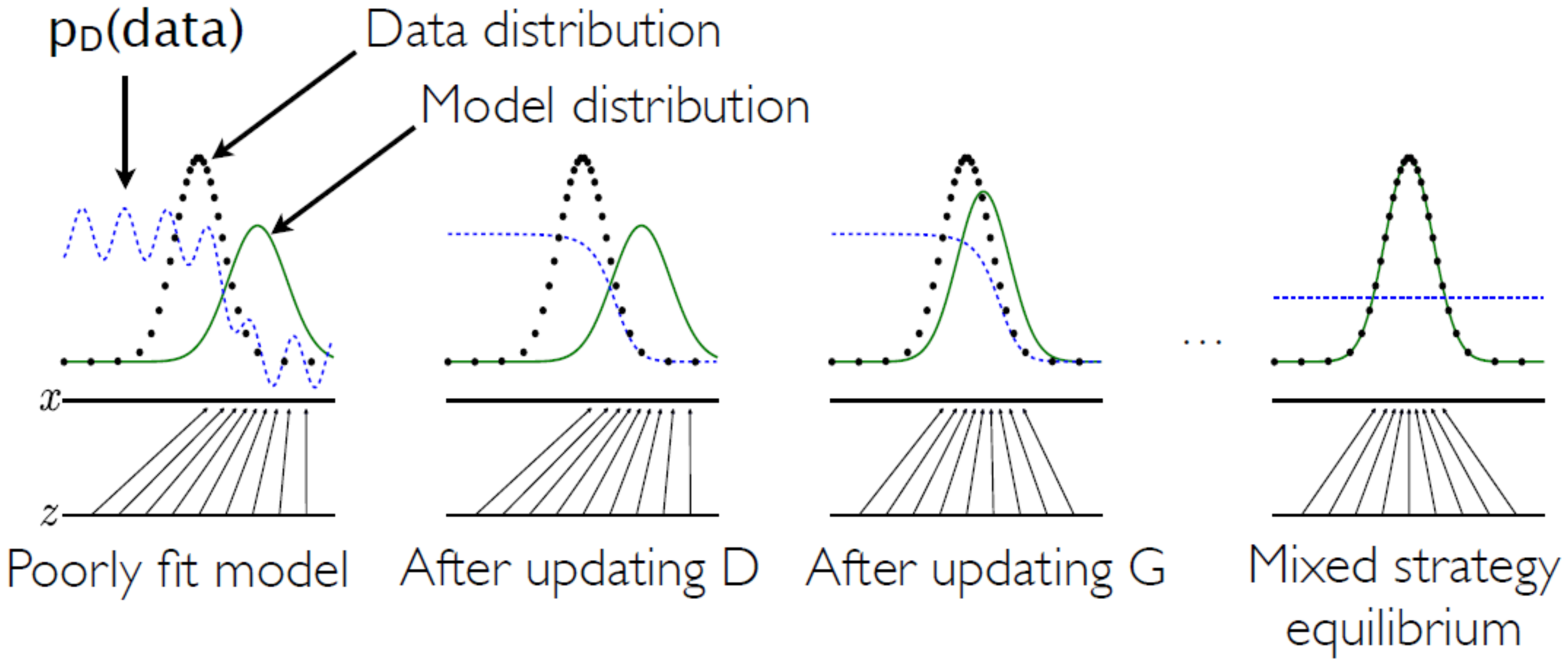
$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

Binary discriminator, usually a neural classifier

Any random distribution in any space

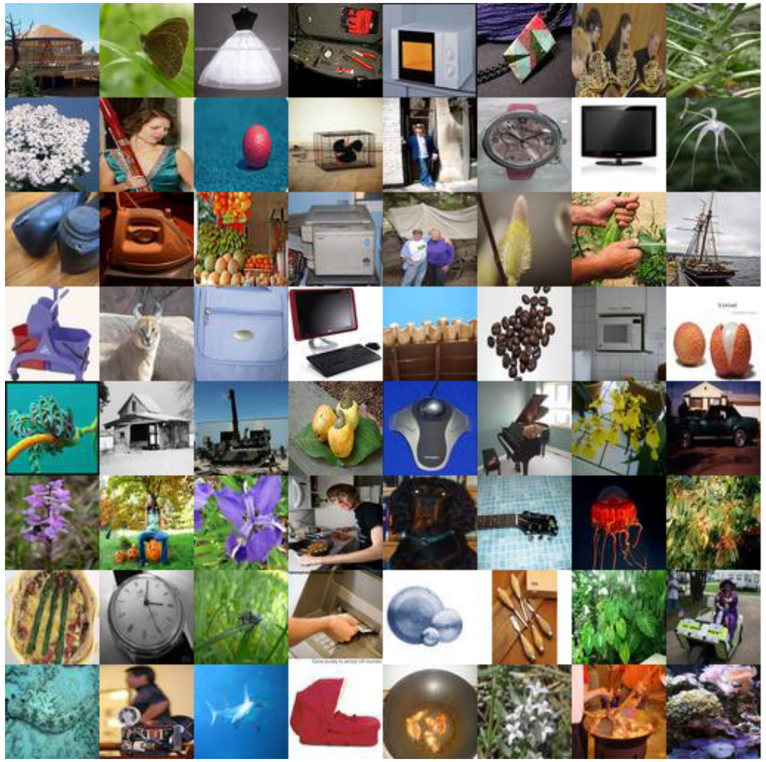
Neural net that maps z → x

# GAN: LEARNING DYNAMICS
## (ADAPTED FROM GOODFELLOW'S, NIPS 2014)

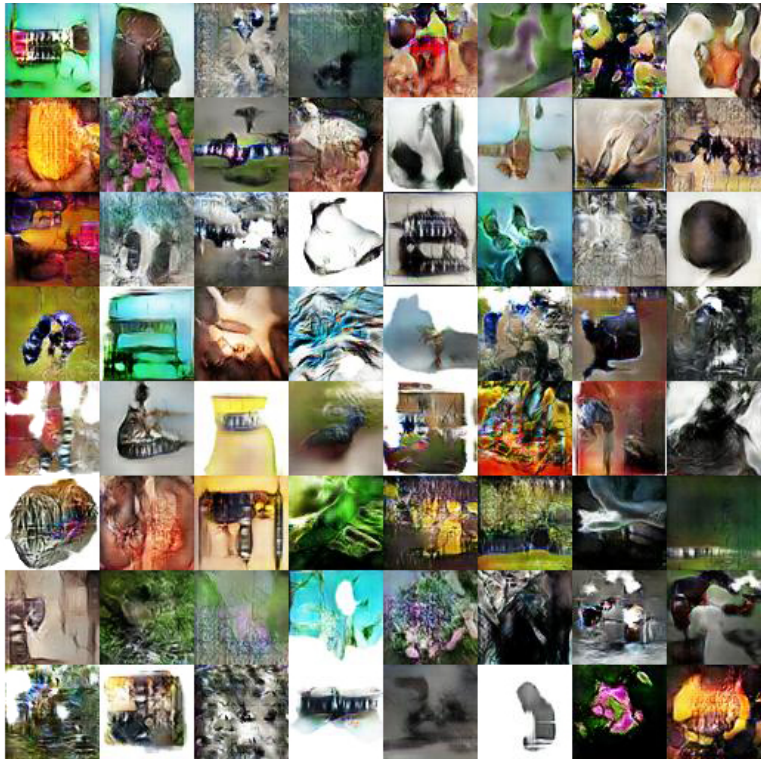


# GAN: GENERATED SAMPLES

The best quality pictures generated thus far!



Real



Generated



http://kvfrans.com/generative-adversial-networks-explained/

# PART III: ADVANCED TOPICS

Unsupervised learning & Generative models

Complex domain structures: Relations (explicit & implicit), graphs & tensors

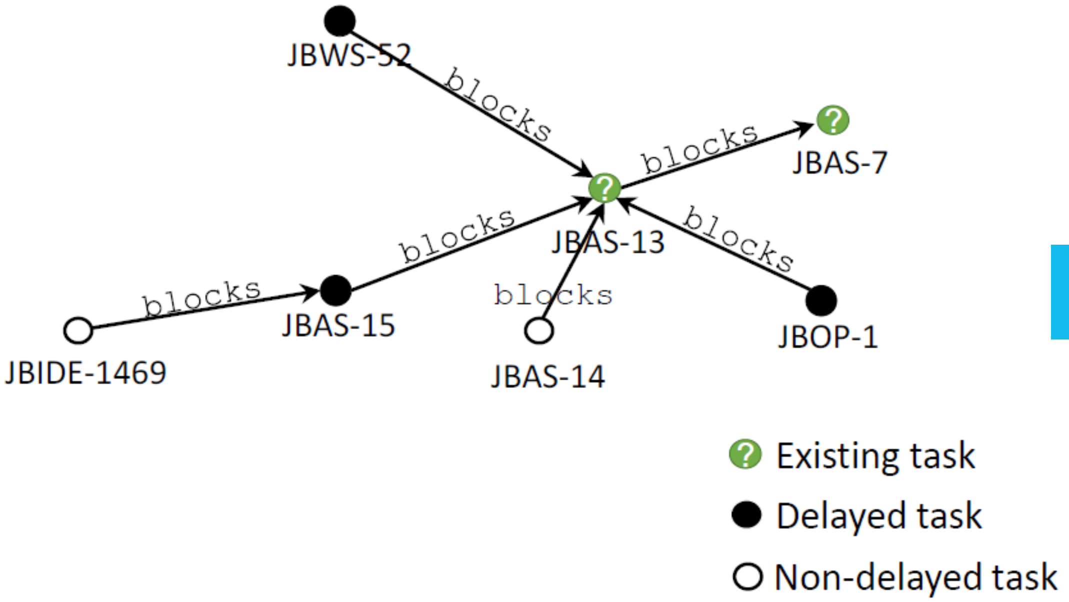
Memory, attention & execution

Learning to learn

How to position ourselves

# EXPLICIT RELATIONS

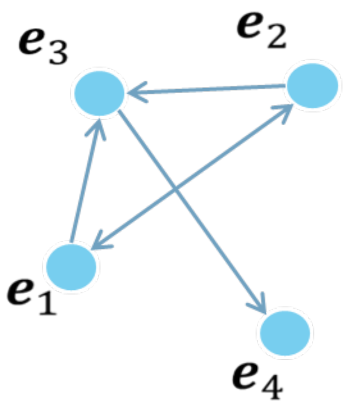
Canonical problem: **collective classification**, a.k.a. structured outputs, networked classifiers

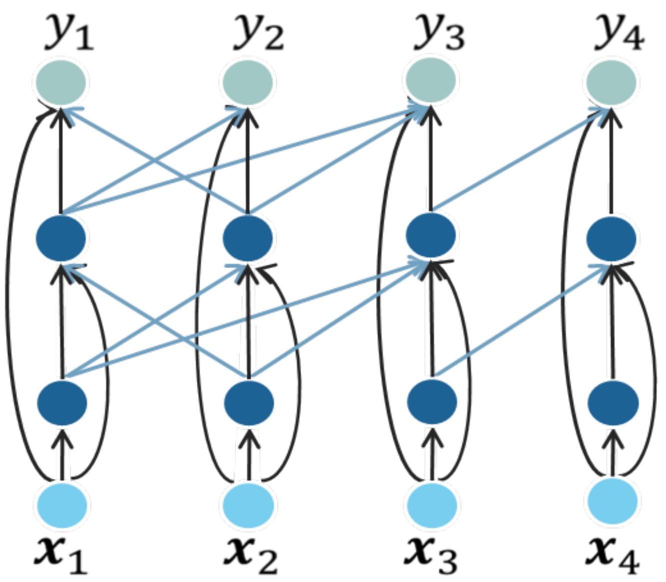



- Stacked inference
- Neural conditional random fields
- Column networks

**Each node has its own attributes**

# STACKED INFERENCE




**Relation graph**

**Stacked inference**

Depth is achieved by stacking several classifiers.

Lower classifiers are frozen.

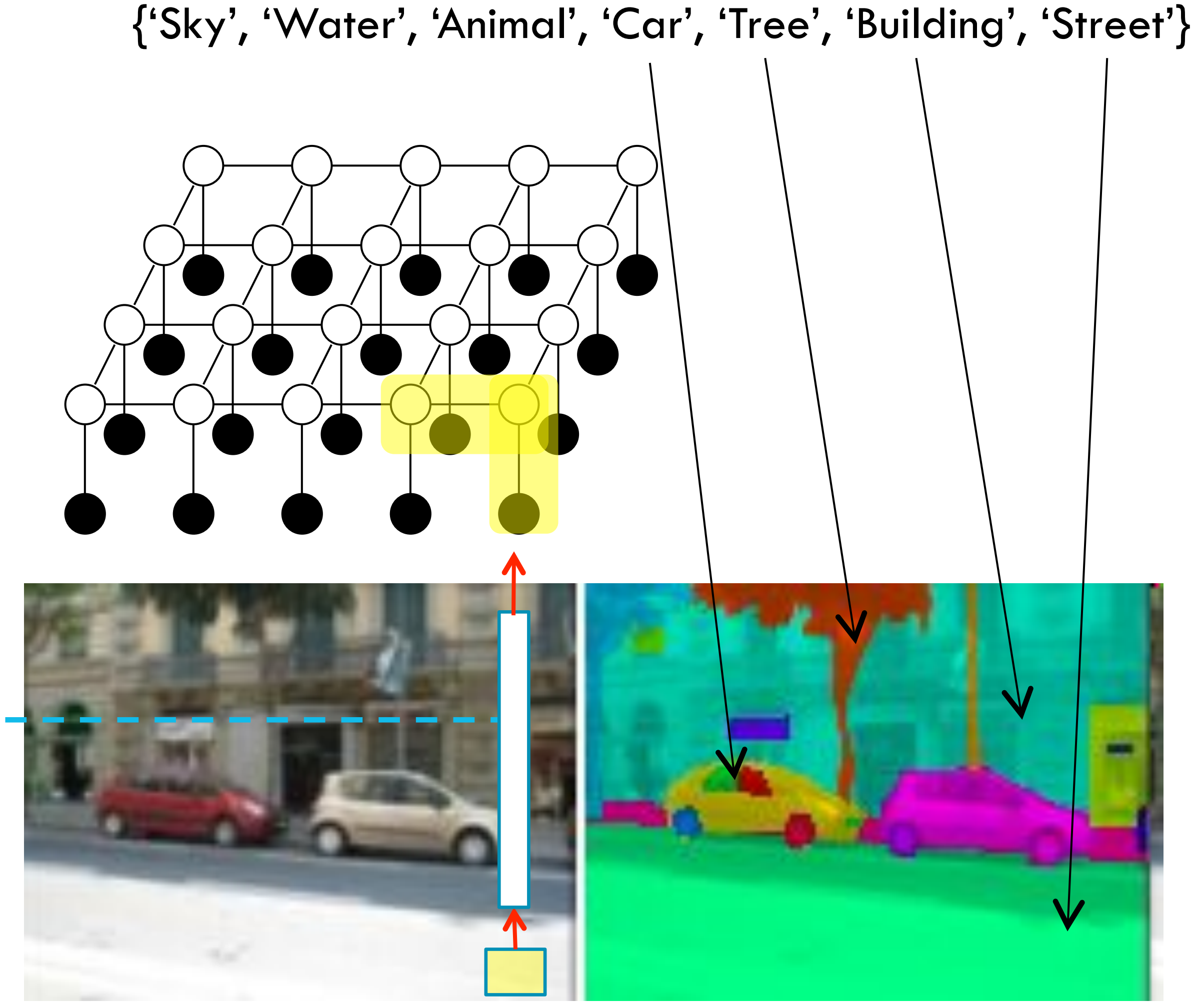
# NEURAL CONDITIONAL RANDOM FIELDS

**Background**: probabilistic graphical models, a semi-formal way to encode (probabilistic) relations:

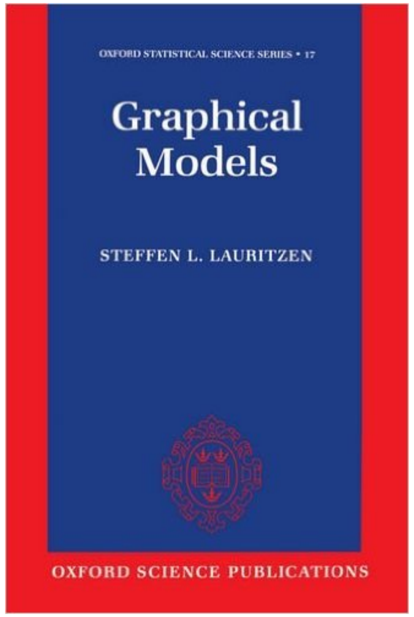
- Conditional dependence between local variables (Bayesian networks)
- Local potential functions (Markov random fields)

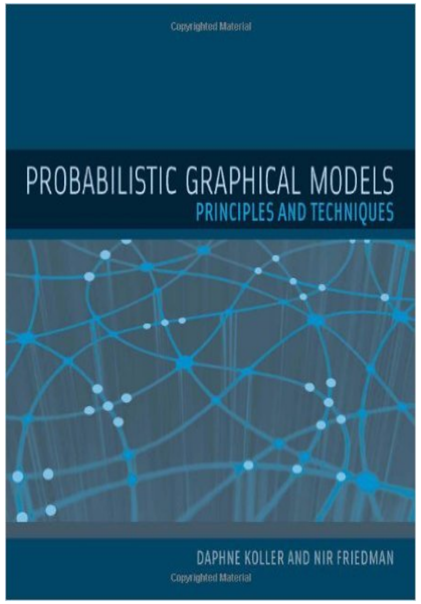
<u>A CRF is a Markov random field conditioned on input variable</u>

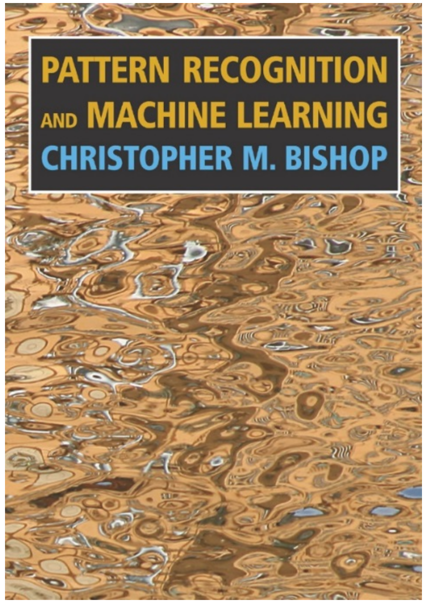
- **Deep nets are for feature extraction**
- Collective inference is principled but difficult
- Mean-field approximation can be seen as a RNN

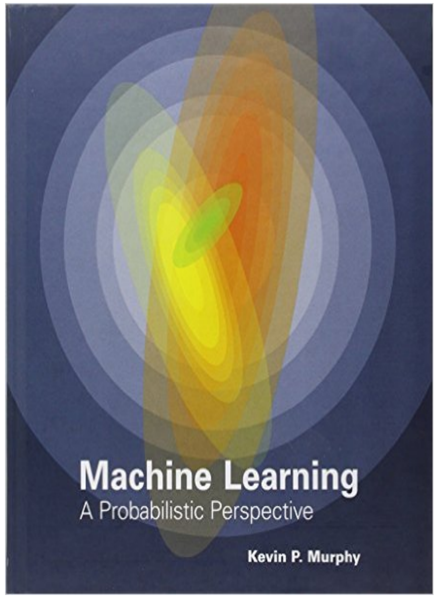


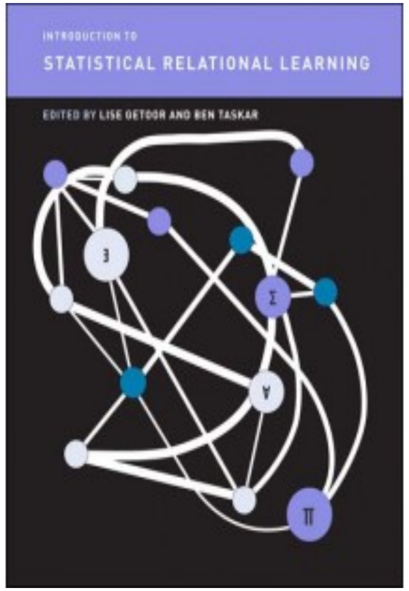
# MORE BACKGROUND ON GRAPHICAL MODELS & STATISTICAL RELATIONAL LEARNING



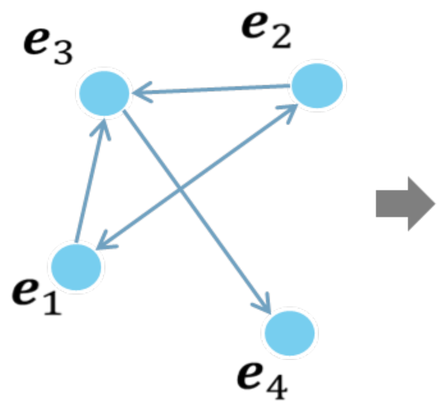



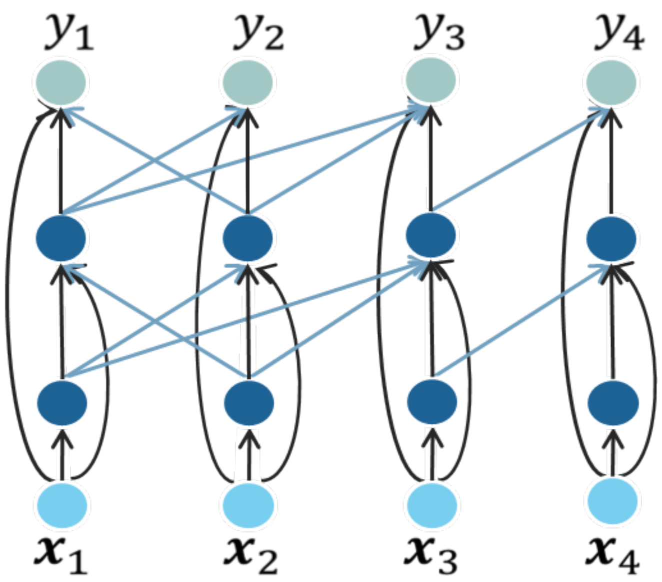




**Coursera course by D. Koller**

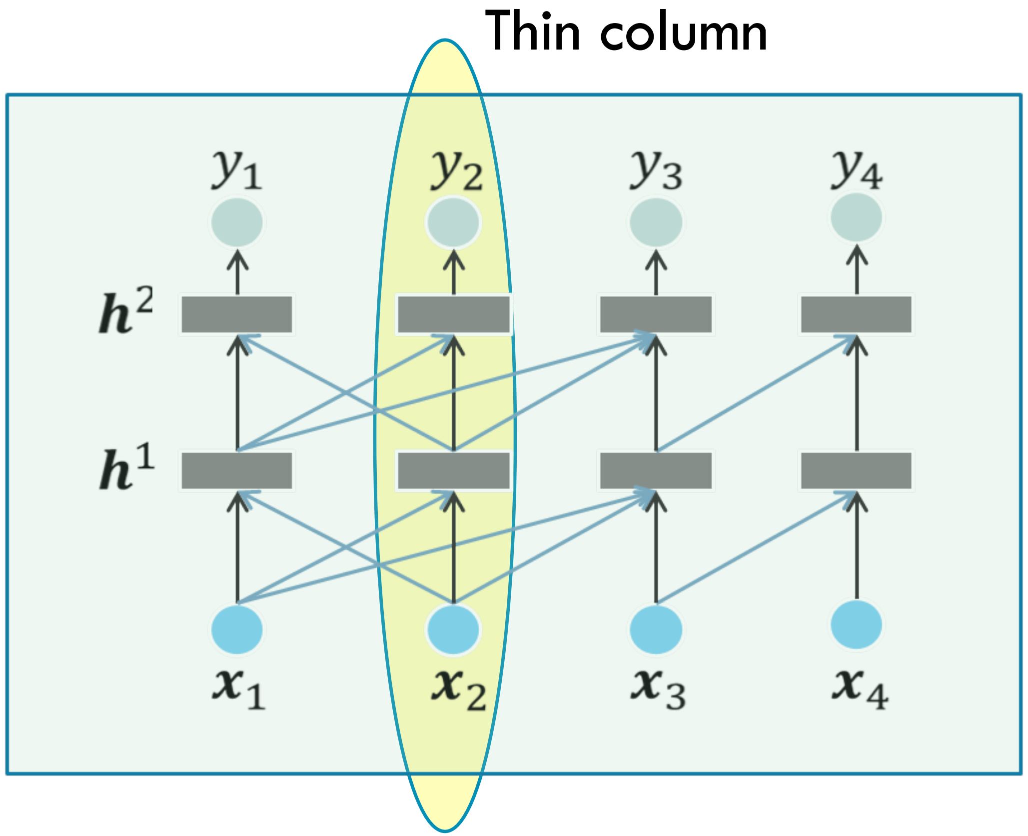
# COLUMN NETWORKS
## (PHAM ET AL, @ AAAI'16)




Relation graph




Stacked learning

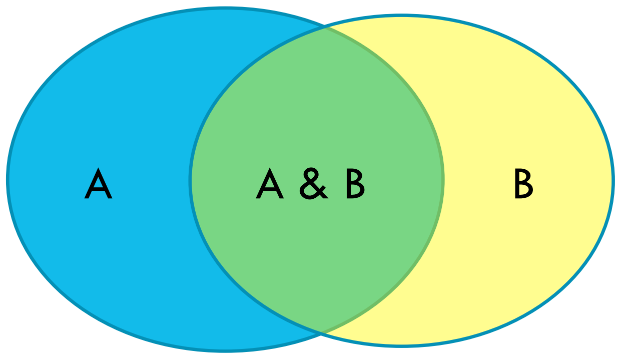



Column nets

# IMPLICIT RELATIONS IN CO-OCCURRENCE OF MULTI-[X] WITHIN A CONTEXT

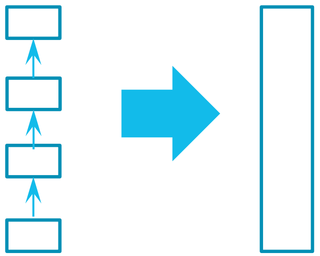
X can be:
- Labels
- Tasks
- Views/parts
- Instances
- Sources

**Much of recent machine learning!**

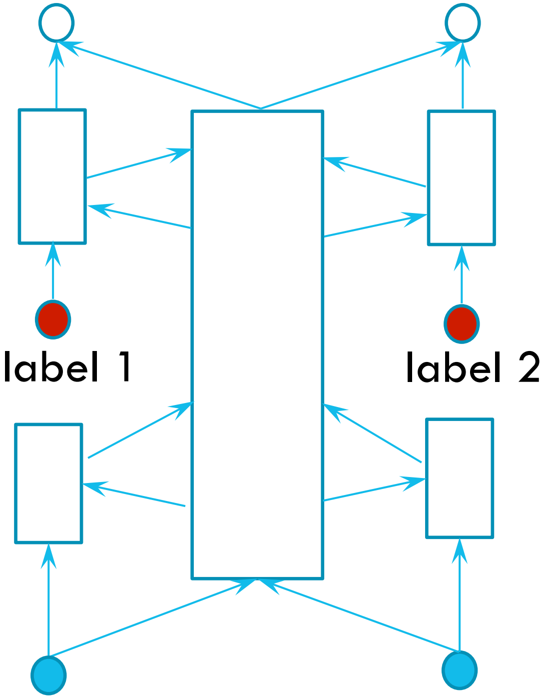



The common principle is to exploit the shared statistical strength

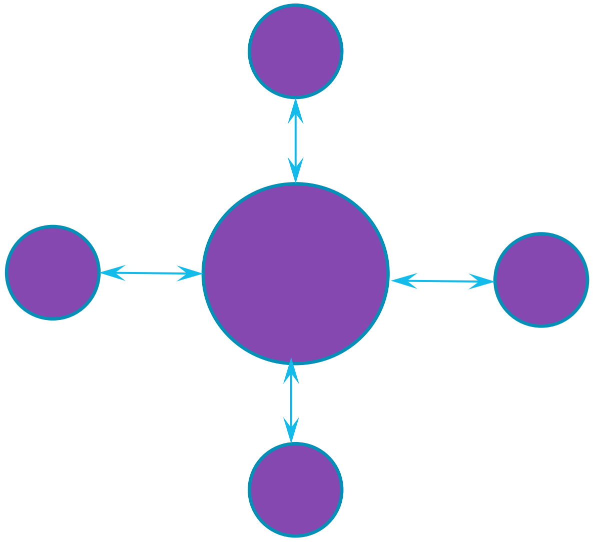
# COLUMN BUNDLE FOR N-TO-M MAPPING
## (PHAM ET AL, WORK IN PROGRESS)



Column




N-to-m setting



Cross-sectional star topology

# GRAPHS AS DATA

**Goal**: representing a graph as a vector

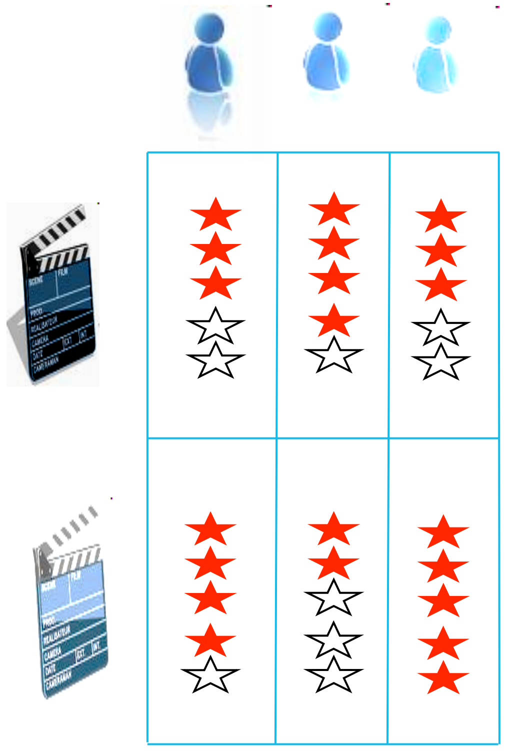
Many applications

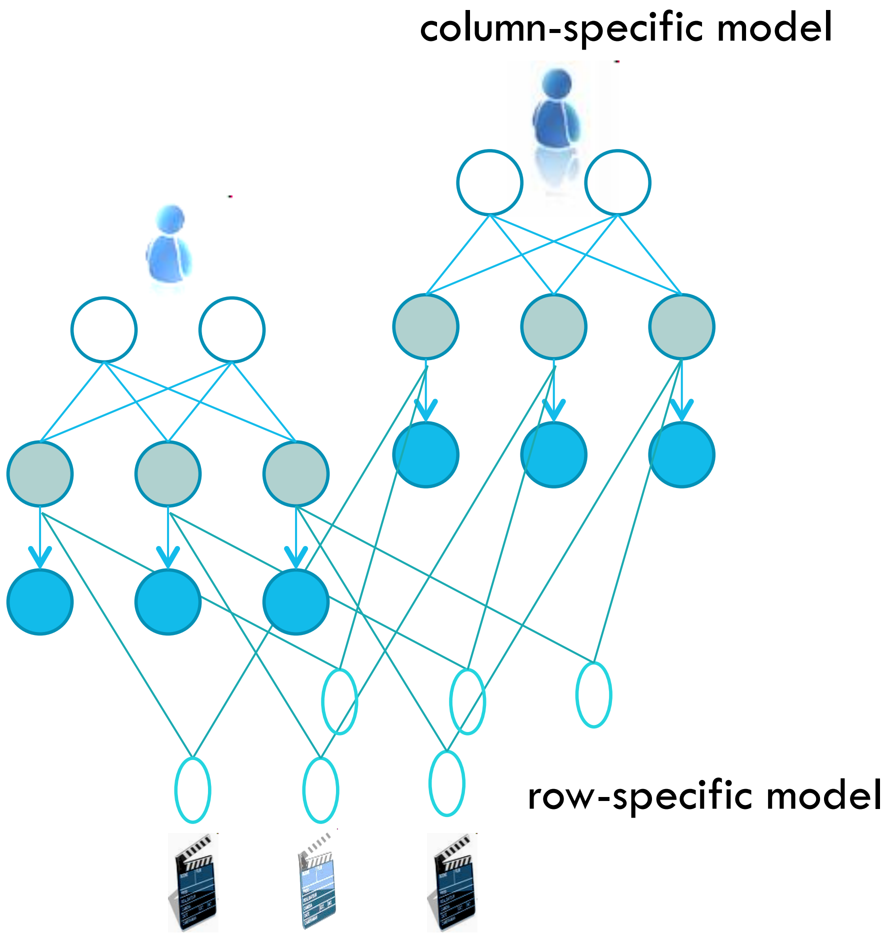
- Drug molecules
- Object sub-graph in an image
- Dependency graph in software deliverable

Recent works:

- Graph recurrent nets, similar to column nets (Pham et al, 2017).
- Graph variational autoencoder (Kipf & Welling, 2016)
- Convolutions for graph (LeCun, Welling and many others)

# RBM FOR MATRIX DATA (TRAN ET AL, 2009, 2012)

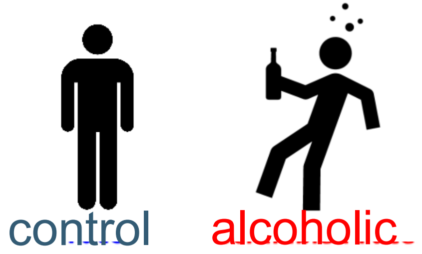




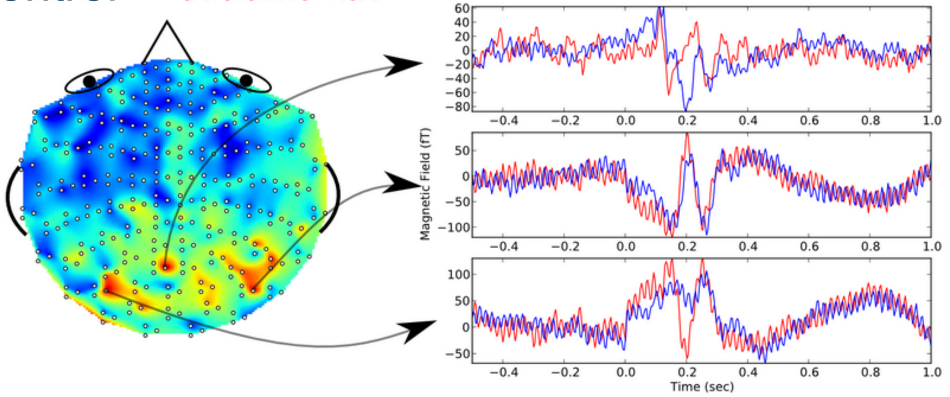
# TENSOR EXAMPLE: EEG-BASED ALCOHOLIC DIAGNOSIS

EEG dataset collected by Zhang *et al.* [2]

- 122 subjects
- 64 electrodes placed on the scalp
- Data small, big supervised models won't work!
- Solution: Unsupervised learning + nearest neighbor



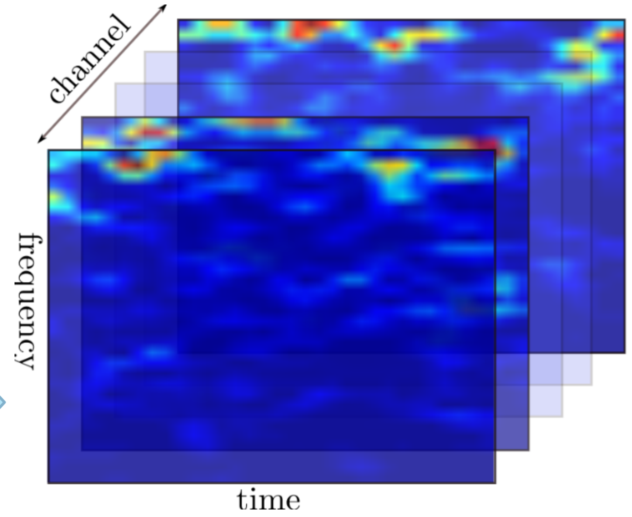

256Hz




*transform*

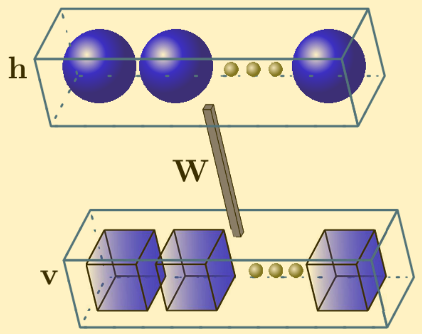
STFT (64, 54)

64x64x64

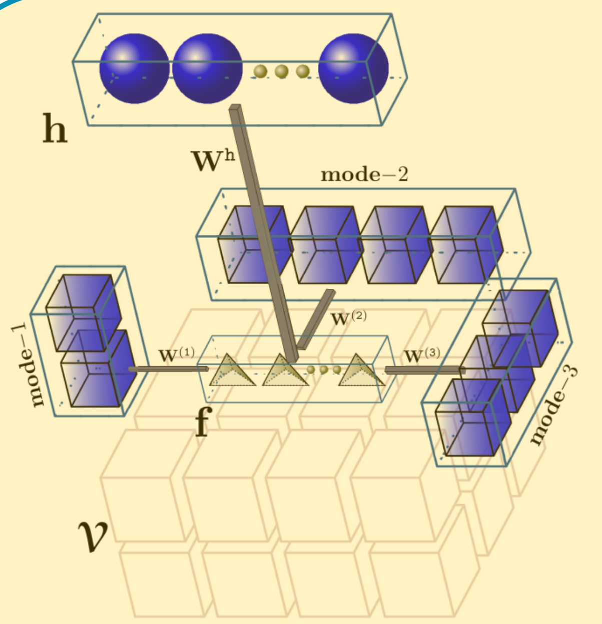



3D Spectrogram

# TENSOR RESTRICTED BOLTZMANN MACHINE (TV.RBM, NGUYEN ET AL, AAAI'15)



RBM



Tensor-variate RBM (TvRBM)

$$p(\mathbf{v}, \mathbf{h}; \psi) \propto \exp\left[\underbrace{-E(\mathbf{v}, \mathbf{h}; \psi)}_{\textit{energy}}\right]$$

$$-\left[\mathcal{F}(\mathbf{v}) + \mathbf{a}^\top \mathbf{v} + \mathbf{b}^\top \mathbf{h} + \mathbf{v}^\top \mathbf{W} \mathbf{h}\right] \qquad -\left[\mathcal{F}(\mathscr{V}) + \langle \mathscr{A}, \mathscr{V} \rangle + \mathbf{b}^\top \mathbf{h} + \langle \mathscr{V}, \mathscr{W} \bar{\times}_{N+1} \mathbf{h} \rangle\right]$$

$$\mathrm{w}_{d_1 d_2 \ldots d_N k} = \sum_{\mathrm{f}=1}^{\mathrm{F}} \sum_{d_1 d_2 \ldots d_N k} \mathrm{w}^{(1)}_{d_1 \mathrm{f}} \cdots \mathrm{w}^{(N)}_{d_N \mathrm{f}} \mathrm{w}^{\mathbf{h}}_{k\mathrm{f}}$$

*Parameter space* $\mathcal{O}(N^N)$ $\mathcal{O}(N^2)$

**RBM = Stochastic Autoencoder**

# EEG-BASED ALCOHOLIC DIAGNOSIS WITH UNSEEN SUBJECTS

36 subjects for testing

Vary the rest for training

| | Classification error (%) | | | | |
|---|---|---|---|---|---|
| Method | 5% | 10% | 25% | 50% | 100% |
| Pixel | 52.78 | 41.67 | 38.89 | 37.24 | 36.11 |
| Tucker | 52.78 | 44.44 | 44.44 | 38.89 | 33.33 |
| PARAFAC | 58.33 | 52.78 | 52.78 | 48.67 | 44.44 |
| RBM | – | – | – | – | – |
| TvRBM | **47.22** | **36.11** | **27.78** | **25.00** | **19.44** |

# PART III: ADVANCED TOPICS

Unsupervised learning & Generative models

Complex domain structures: Relations (explicit & implicit), graphs & tensors

**Memory, attention & execution**

Learning to learn

How to position ourselves

# WHY MEMORY & ATTENTION?

Long-term dependency

- E.g., outcome depends on the far past
- Memory is needed (e.g., as in LSTM)

Complex program requires multiple computational steps

- Each step can be selective (attentive) to certain memory cell

Operations: Encoding | Decoding | Retrieval

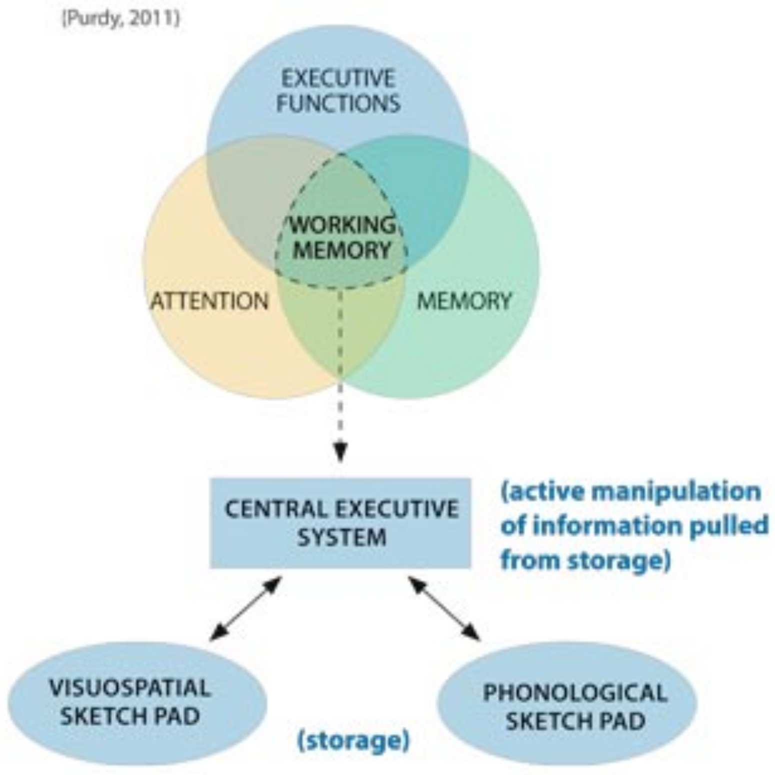
# MEMORY TYPES

Short-term/working (temporary storage)

Episodic (events happened at specific time)

Long-term/semantic (facts, objects, relations)

Procedural (sequence of actions)




http://www.rainbowrehab.com/executive-functioning/

# ATTENTION MECHANISM

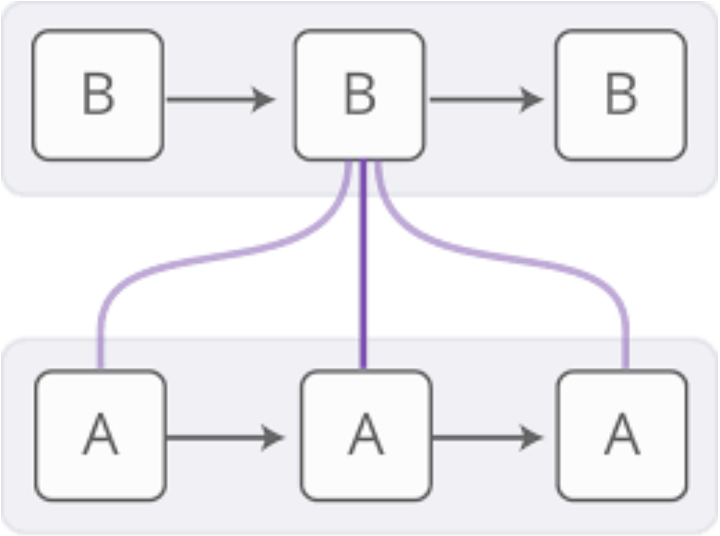
Need attention model to select or ignore certain inputs

Human exercises great attention capability – the ability to filter out unimportant noises

- Foveating & saccadic eye movement

In life, events are not linear but interleaving.

Pooling (as in CNN) is also a kind of attention




http://distill.pub/2016/augmented-rnns/

# APPLICATIONS

Machine reading & question answering

- Attention to specific events/words/sentences at the reasoning stage

Machine translation

- Word alignment – attend to a few source words
- Started as early as IBM Models (1-5) in early 1990s

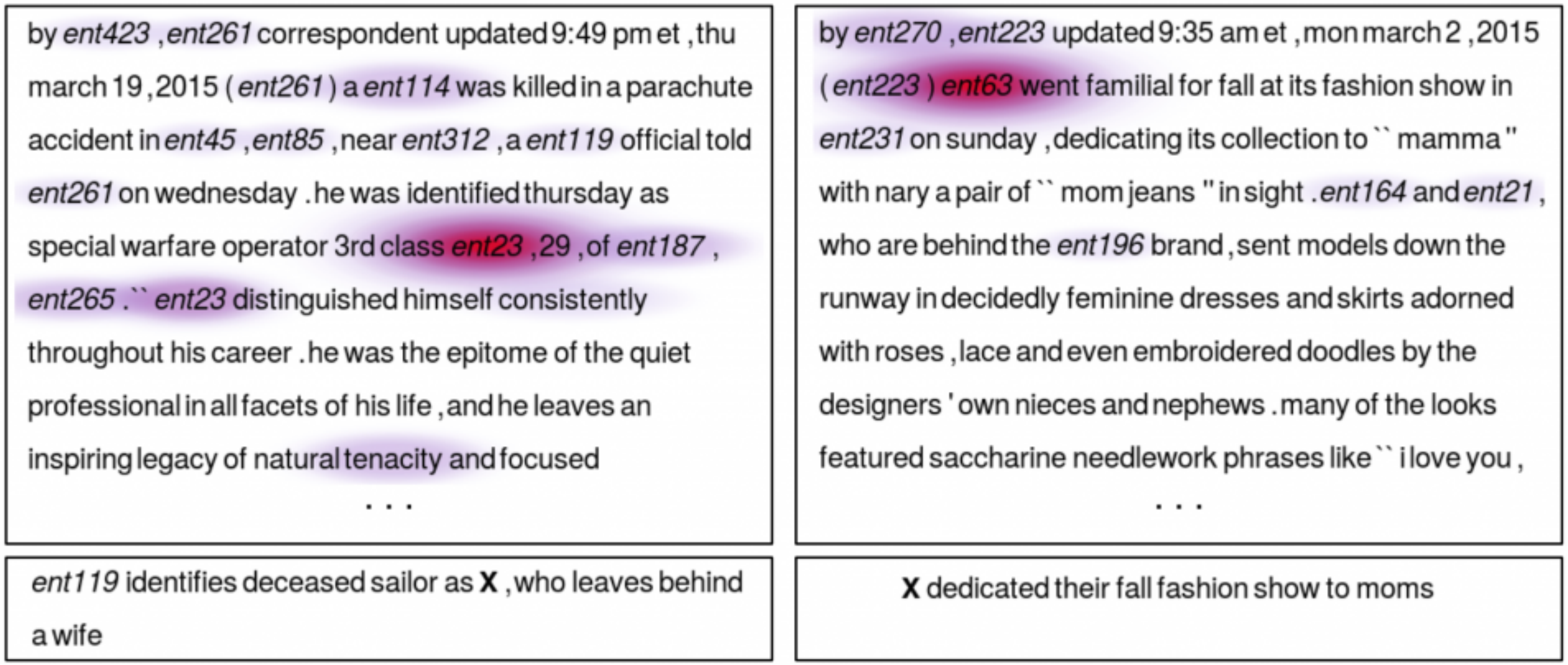
Speech recognition

- A word must be aligned to a segment of soundwave

Healthcare

- Diseases can be triggered by early events and take time to progress
- Illness has memory – negative impact to the body and mind

# EXAMPLE: MACHINE READING
## (HERMANN ET AL, 2015)



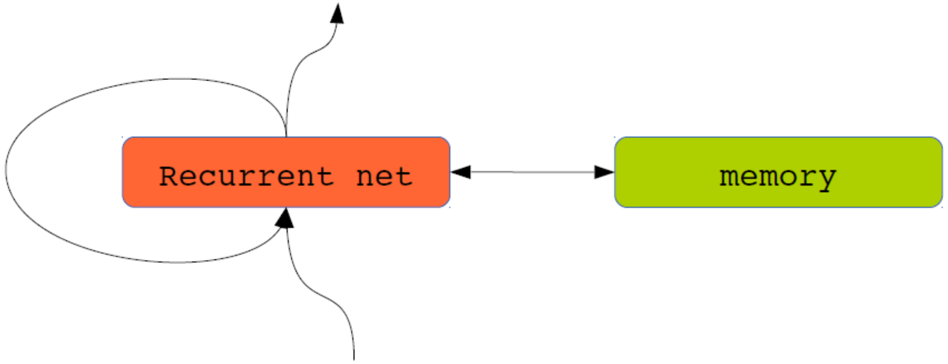
# EXECUTION (RNN) + MEMORY + ATTENTION

Memory networks of Facebook: (Weston et al, Facebook, 2015); (Sukhbaatar et al, 2015) – associative memory

Dynamic memory networks of MetaMind: (Kumar et al, 2015) – episodic memory

Neural Turing machine of DeepMind (Graves et al. 2014) -- tape

Stacked-augmented RNN for learning algorithmic sequences (Joulin & Mikolov, 2015) -- stack




(LeCun, 2015)

# END-TO-END MEMORY NETWORKS
## (SUKHBAATAR ET AL, 2015)

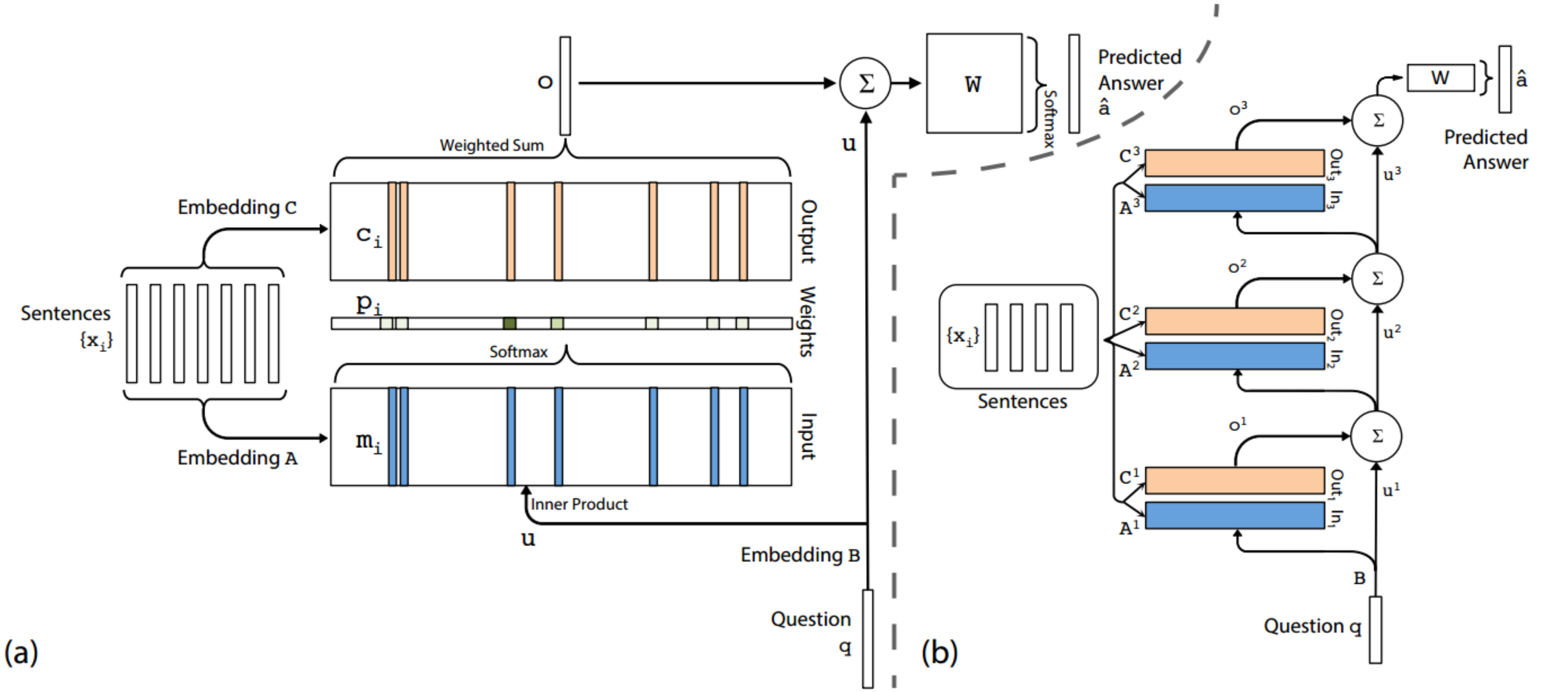
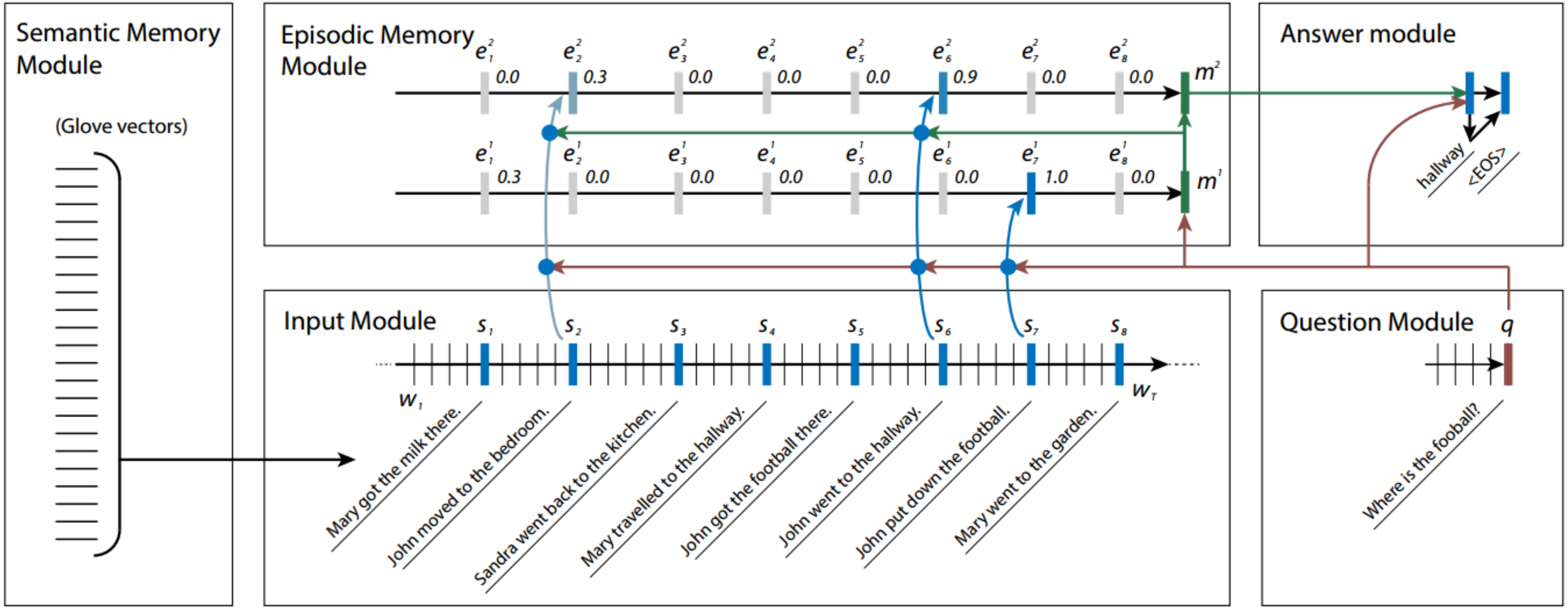



Figure 1: (a): A single layer version of our model. (b): A three layer version of our model. In practice, we can constrain several of the embedding matrices to be the same (see Section 2.2).

# DYNAMIC MEMORY NETWORKS
## (KUMAR ET AL, 2015)



# NEURAL TURING MACHINE (DEEPMIND, GRAVES ET AL, 2014)

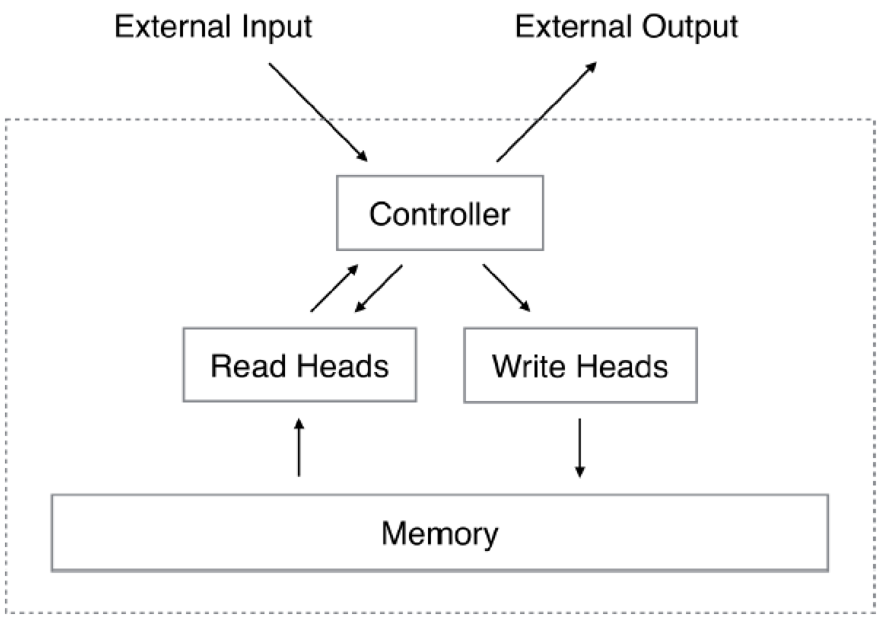



**Figure 1: Neural Turing Machine Architecture.** During each update cycle, the controller network receives inputs from an external environment and emits outputs in response. It also reads to and writes from a memory matrix via a set of parallel read and write heads. The dashed line indicates the division between the NTM circuit and the outside world.

# NTM: DIFFERENTIABLE COMPUTER

Learn to program.

*All operations are differentiable*.

Back to the basic of computer primitives:

- Arithmetic
- Data movements
- Control jumps

Computer architectures:

- CPU with very-limited memory (registers).
- RAM to hold rapidly-created variables.
- Hard-disks to hold large-scale static data (missing in NTM, present in Memory Nets).

# PART III: ADVANCED TOPICS

Unsupervised learning & Generative models

Complex domain structures: Relations (explicit & implicit), graphs & tensors

Memory, attention & execution

Learning to learn

How to position ourselves

# SMARTER LEARNING

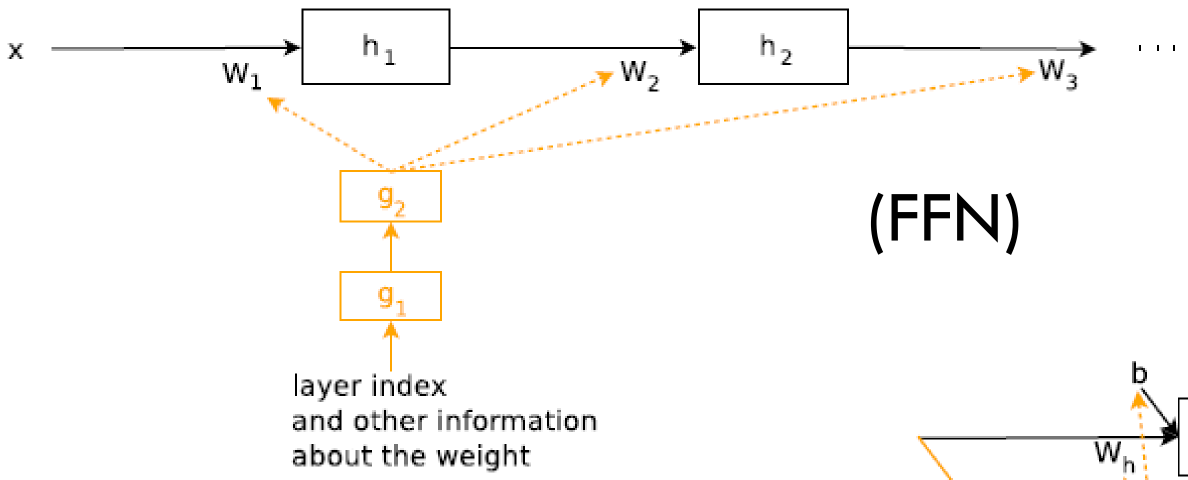
Learn more than one thing at a time

Leverage what is known

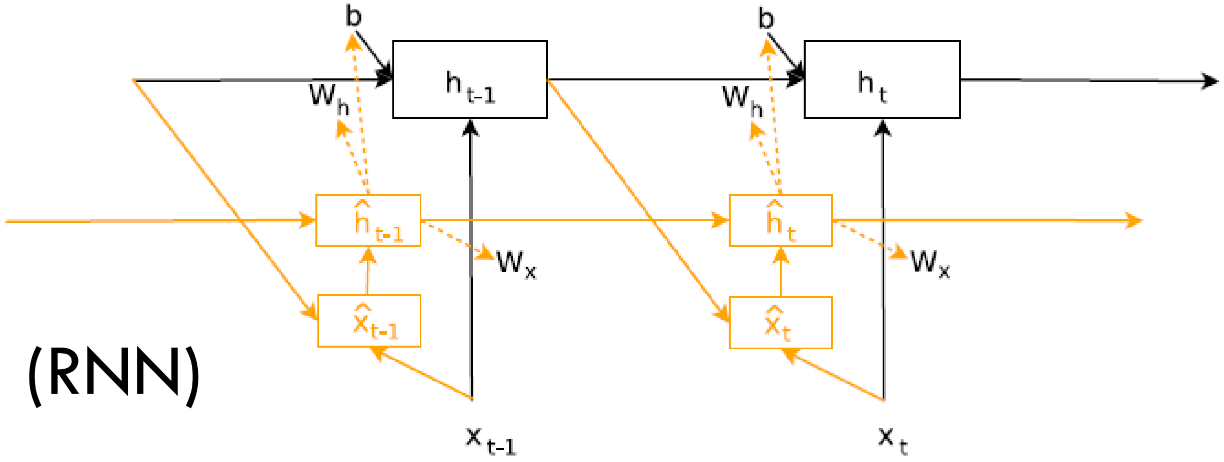
Lifelong, interleaved learning

Learn to program to program

# HYPERNETWORKS: NETWORK TO GENERATE NETWORKS (HA ET AL, 2016)

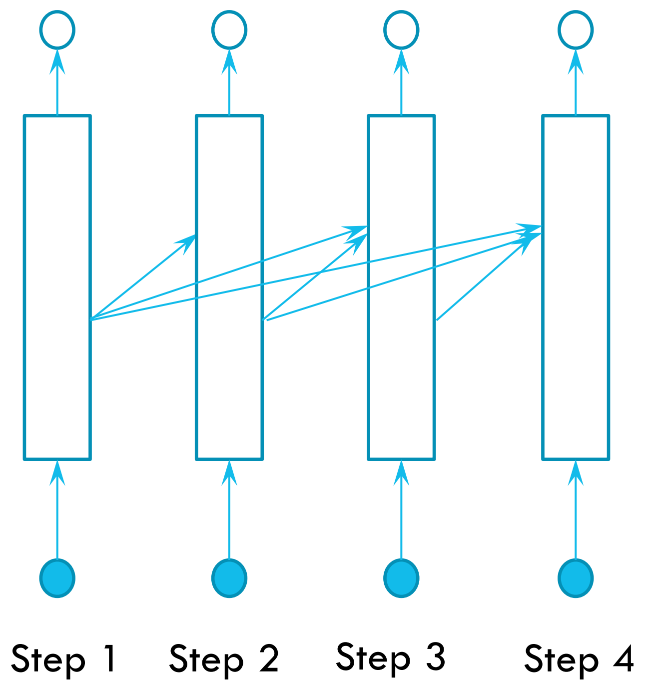



(FFN)




(RNN)

# SEQUENTIAL, LIFELONG LEARNING (DO ET AL, WORK IN PROGRESS)




Boosting

Transfer learning

Curriculum learning

Domain adaptation

Syllabus learning

Interleaved learning

# PART III: ADVANCED TOPICS

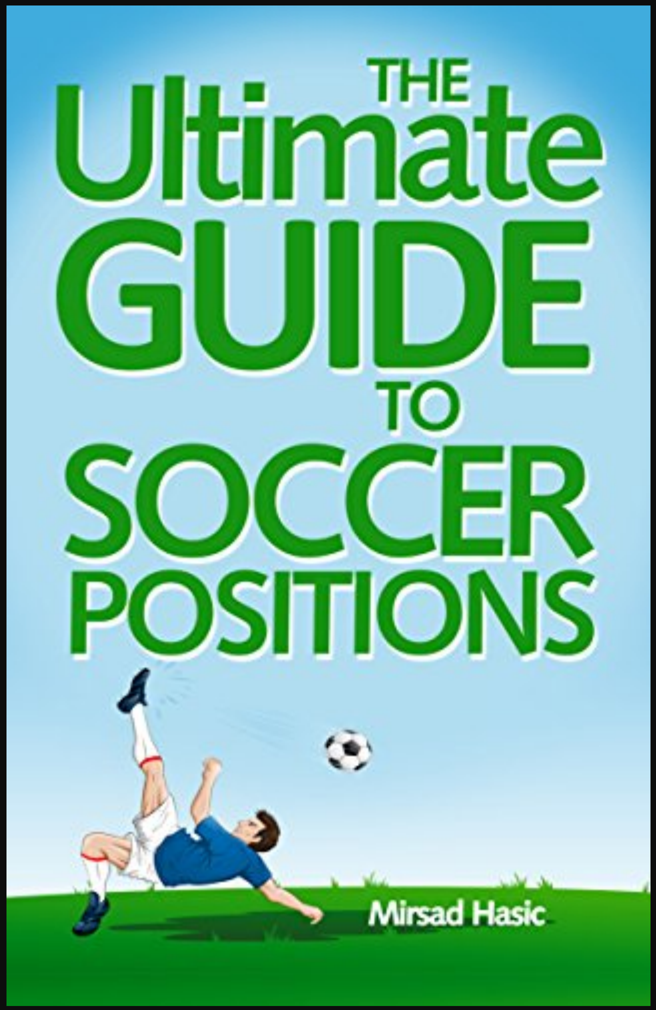
Unsupervised learning & Generative models

Complex domain structures: Relations (explicit & implicit), graphs & tensors

Memory, attention & execution

Learning to learn

How to position ourselves

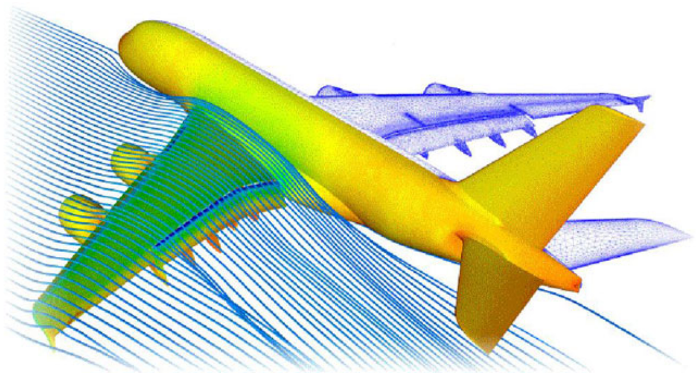


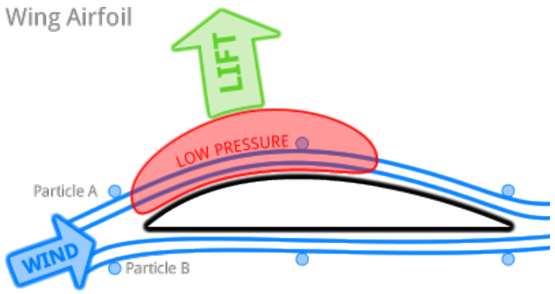
# IN CASE YOU'RE WORRIED ABOUT WHAT IS LEFT

Current deep learning is pre-Newtonian mechanics

Equivalent to demonstrating that *heavier-than-air flying* possible, without figuring out **aerodynamics**

We need to find **law of physics** (intelligence), not building flapping wings (simulating neurons)






Sources:
http://aero.konelek.com/aerodynamics/aerodynamic-analysis-and-design
http://www.foolishsailor.com/Sail-Trim-For-Cruisers-work-in-progress/Sail-Aerodynamics.html

# POSITION YOURSELF

"[…] the dynamics of the game will evolve. In the long run, the right way of playing football is to position yourself intelligently and to wait for the ball to come to you. You'll need to run up and down a bit, either to respond to how the play is evolving or to get out of the way of the scrum when it looks like it might flatten you." (*Neil Lawrence, 7/2015, now with Amazon*)

http://inverseprobability.com/2015/07/12/Thoughts-on-ICML-2015/

# THE ROOM IS WIDE OPEN

**Architecture engineering**

**Non-cognitive apps**

Going Bayesian

**Unsupervised learning**

Graphs

Reinforcement learning

**Modelling of invariance**

Learning while preserving privacy

Integrating with cognitive neuroscience

Better data efficiency

Learning under adversarial stress

Mixing learning and reasoning

Multimodality

Better optimization

Non-gradient learning

Symmetry, group theory and all that

From distributed to symbolic representation

http://smerity.com/articles/2016/architectures_are_the_new_feature_engineering.html

# DO SOMETHING HARDER

#Ref: http://www.inference.vc/deep-learning-is-easy/

Advances that make it easy:

- Effective adaptive SGDs like Adagrad, Adam, RMSProp – less worries about convergence speed and learning scheduling.
- Automatic differentiation – no worries about getting the gradient right.
- Packages like Keras, Lasagne make things supper easy
- Trained models for vision and NLPs are powerful – off-the-shelf feature extractor works well.

Building a complicated network is like building a Lego structure

“There is also a feeling in the field that low-hanging for deep learning is disappearing.”

“A NEW IDEA IS JUST RE-PACKAGING OF OLD IDEAS”

# OPEN QUESTIONS

Is this just yet-another-toolbox or a way of thinking?

Is this a right approach to AI?

Q|A

# Thank you!

http://ahsanqawl.com/2015/10/qa/