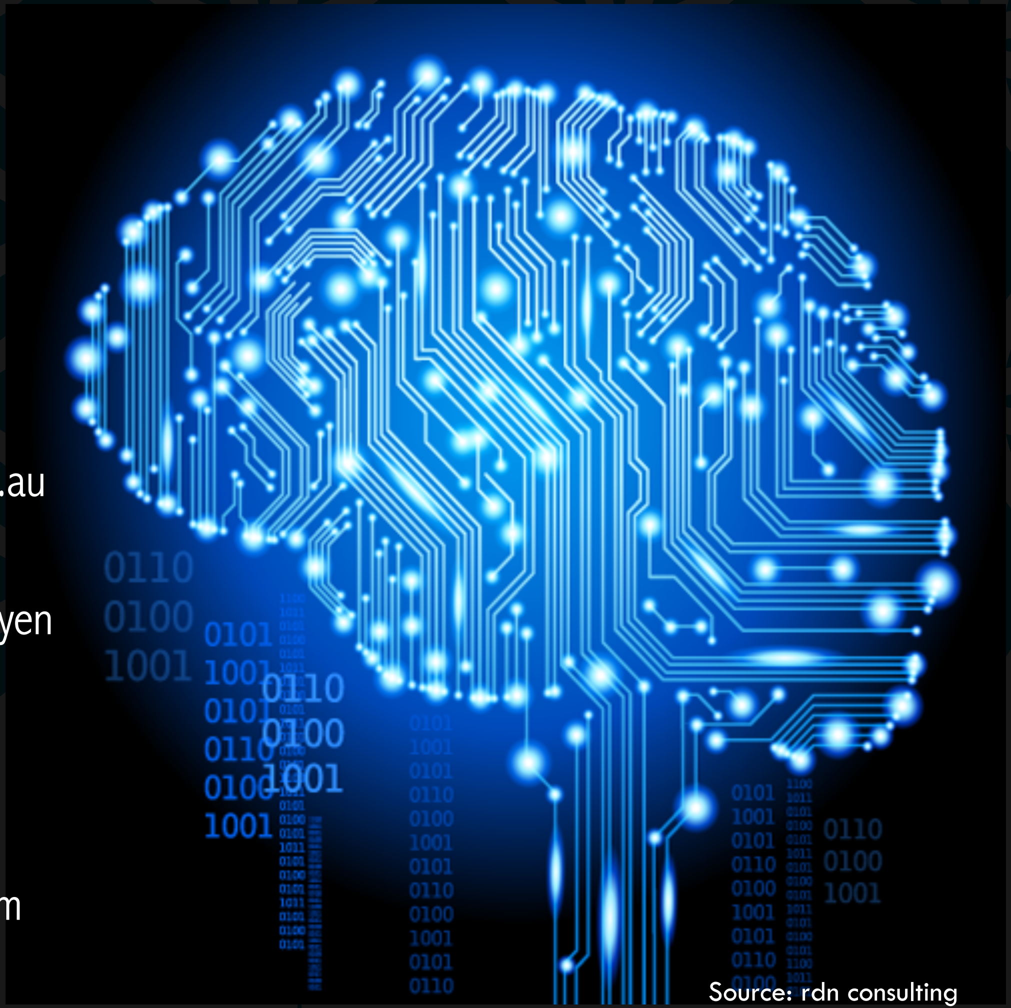
DEEP ARCHITECTURE ENGINEERING

Truyen Tran
Deakin University

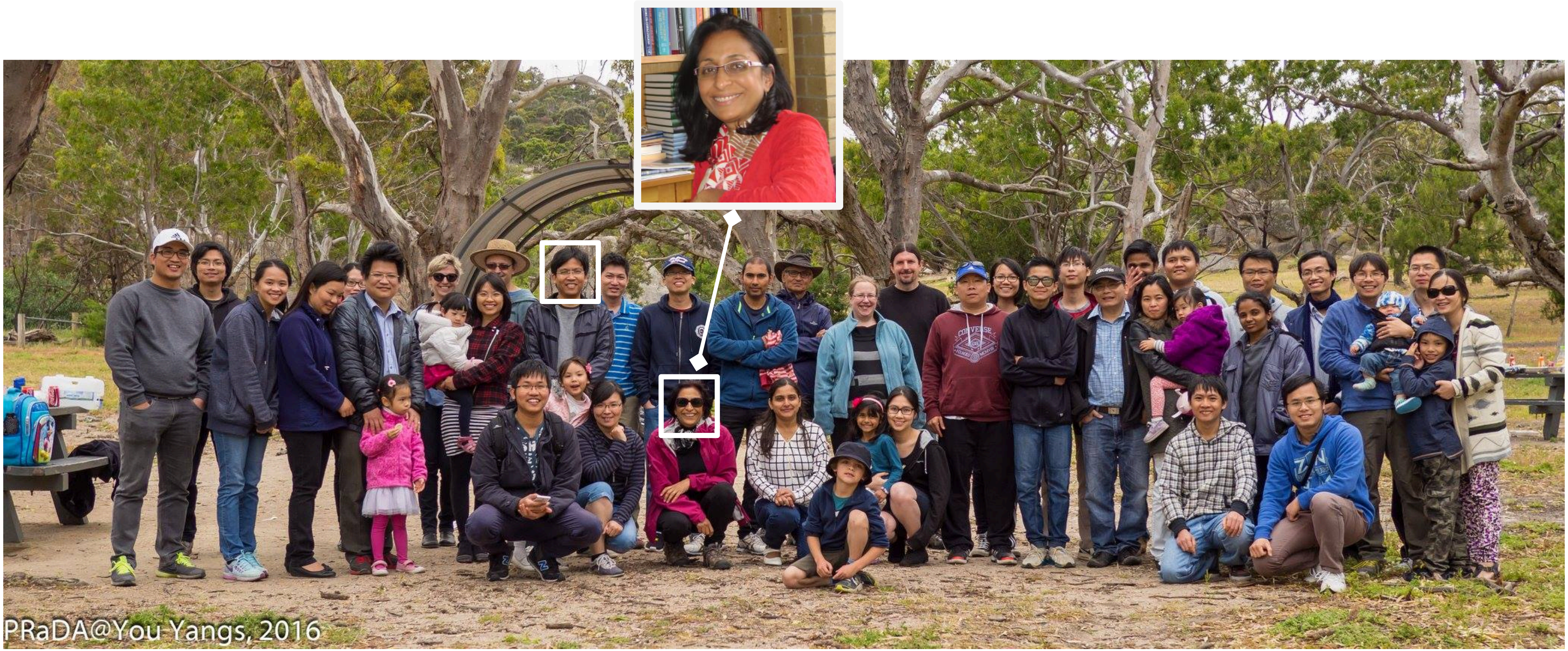
truyen.tran@deakin.edu.au
prada-research.net/~truyen
@truyenoz
letdataspeak.blogspot.com
goo.gl/3jJ1O0

Hanoi, Jan 10th 2017



Source: rdn consulting

PRADA @ DEAKIN, GEELONG CAMPUS



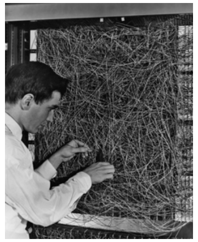
AGENDA

Part I: Introduction to (mostly supervised) deep learning

Part II: Architecture engineering

Yann LeCun

1988

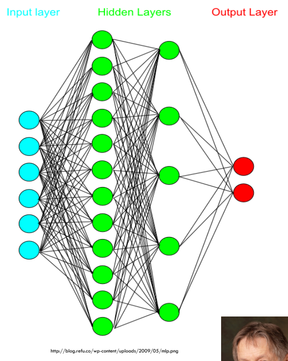


Rosenblatt's perceptron

1958

Geoff Hinton

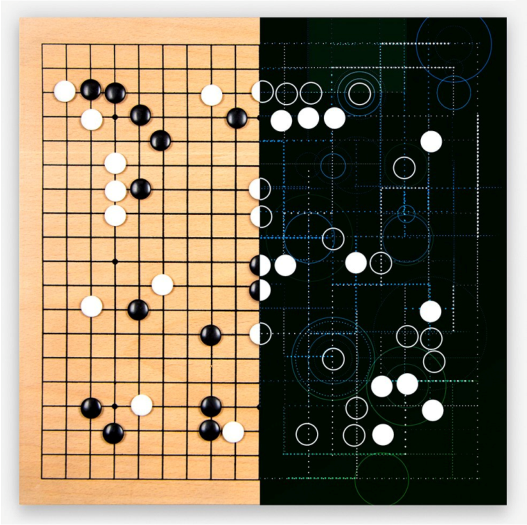
2006



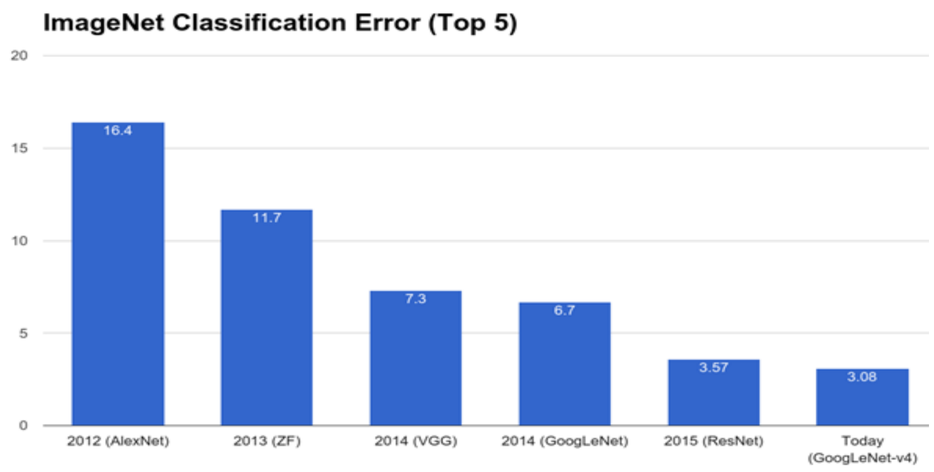
1986

2012

AlphaGo

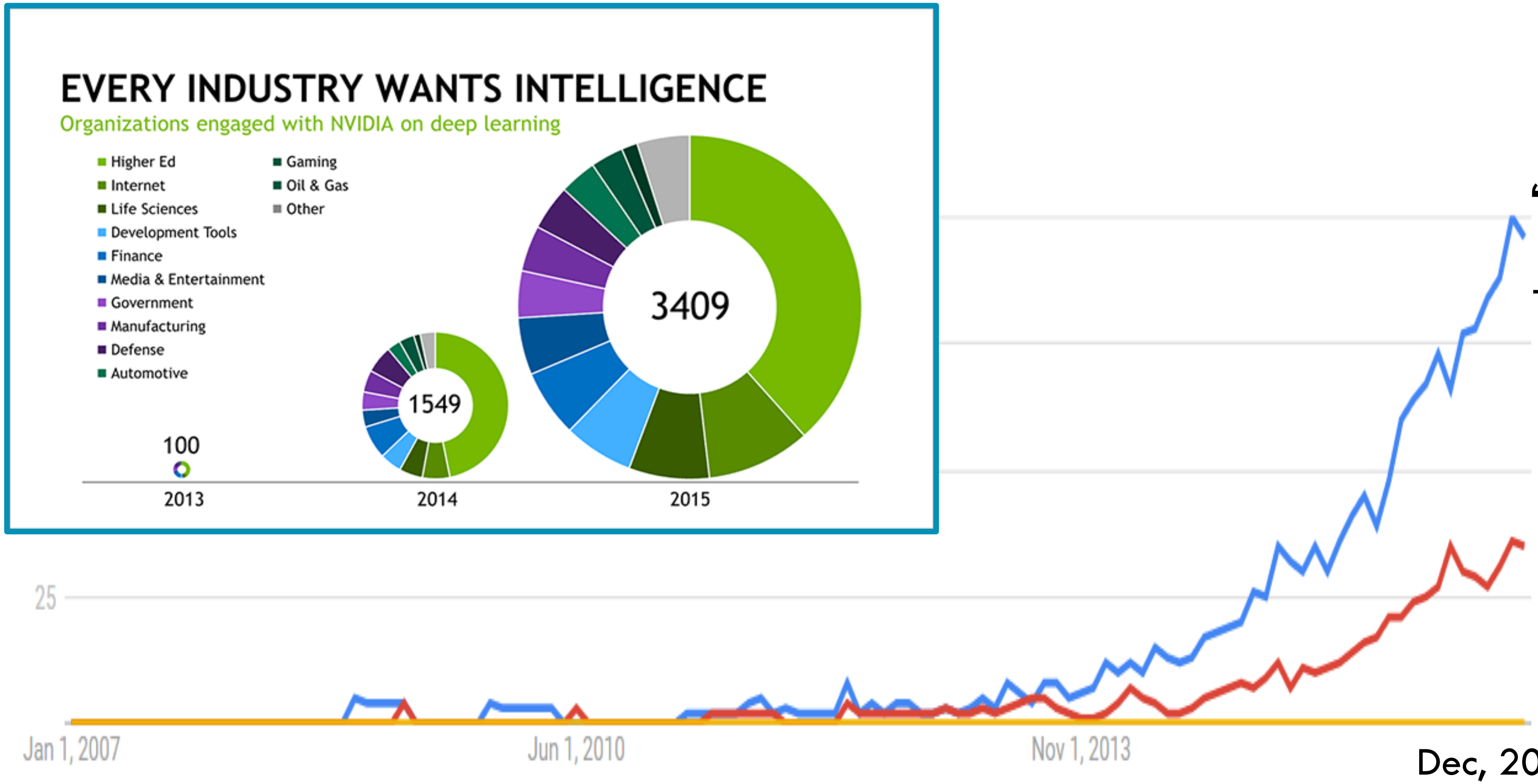


2016-2017



http://redcatlabs.com/2016-07-30_FifthElephant-DeepLearning-Workshop/#/

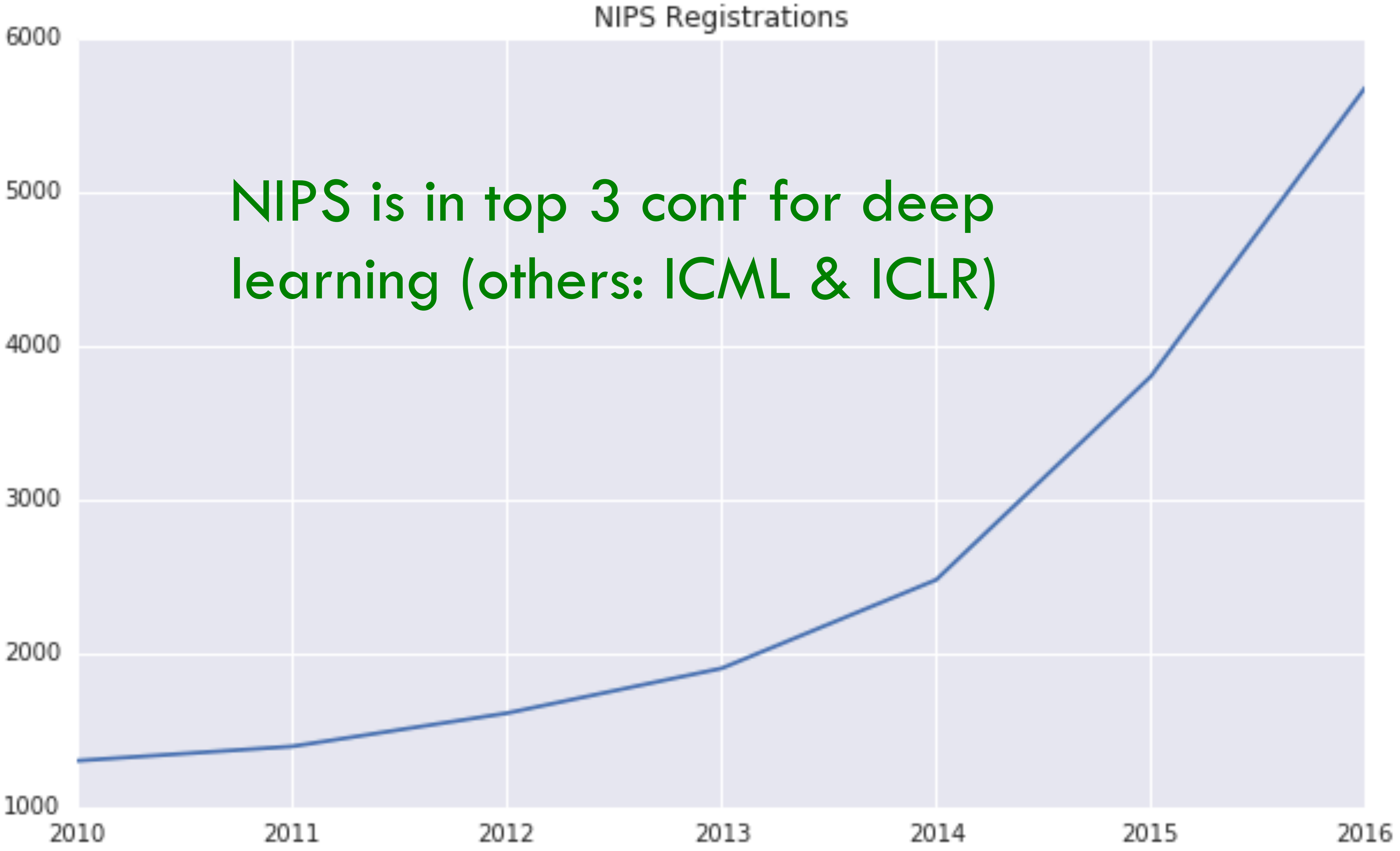
DEEP LEARNING IS SUPER HOT



"deep learning" + **data**

"deep learning" + **intelligence**

Dec, 2016



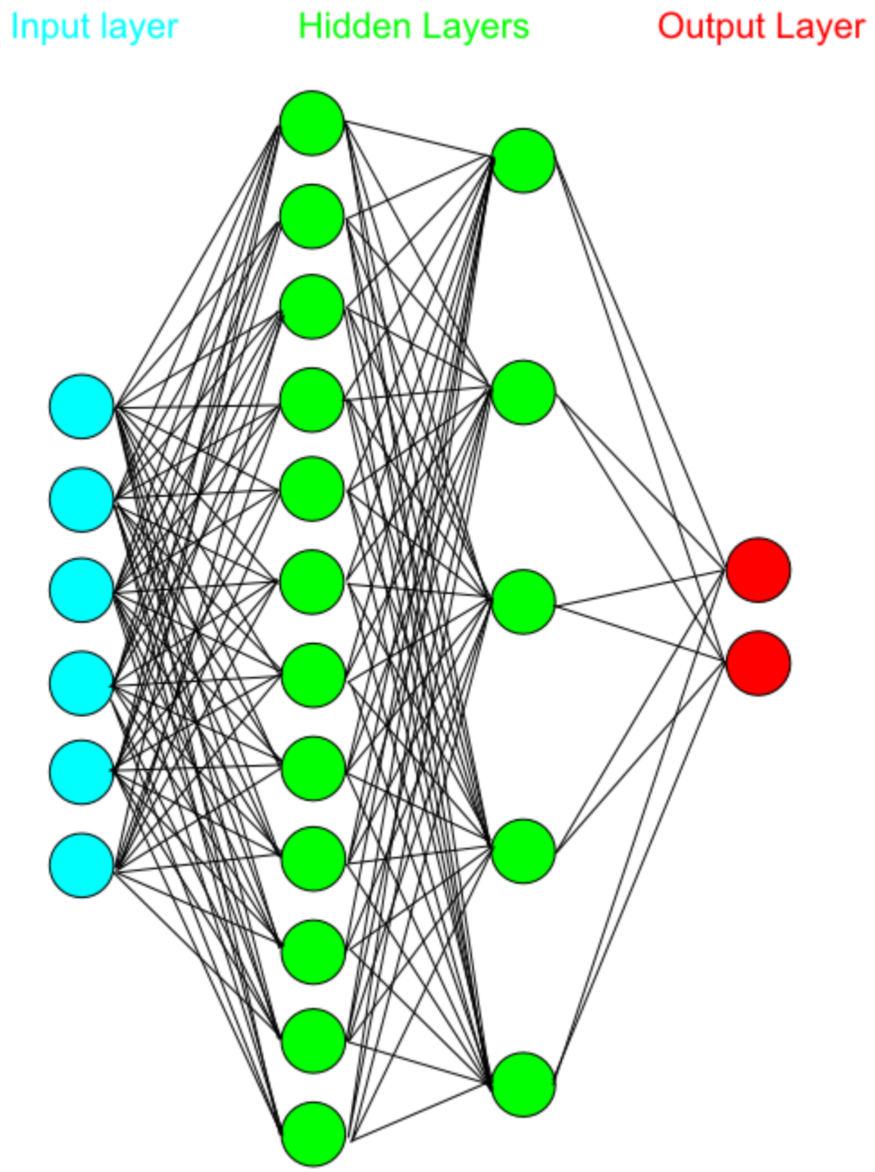
NIPS is in top 3 conf for deep learning (others: ICML & ICLR)

WHAT IS DEEP LEARNING?

Fast answer: multilayer perceptrons (aka deep neural networks) of the 1980s rebranded in 2006.

- But has a lot more hidden layers (10-100X).

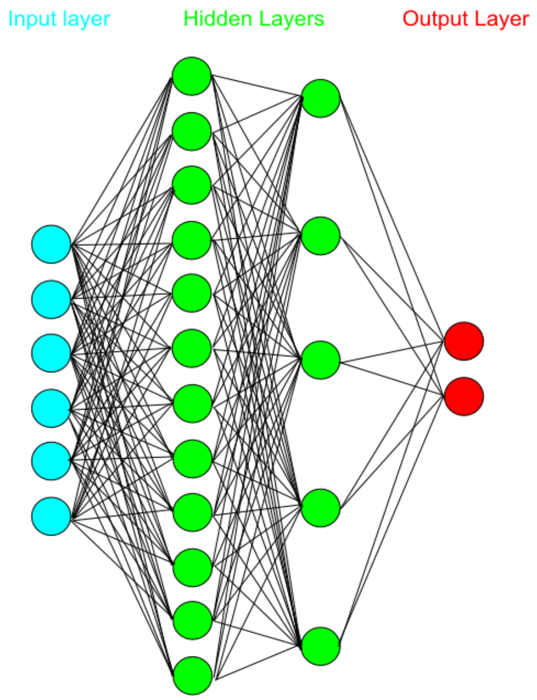
Slow answer: multilayer abstraction, recursive function, multiple steps of computation, iterative estimation, compositionality of the world, better priors, advances in **compute**, **data** & **optimization**, **neural architectures**, etc.



http://blog.refu.co/wp-content/uploads/2009/05/mlp.png

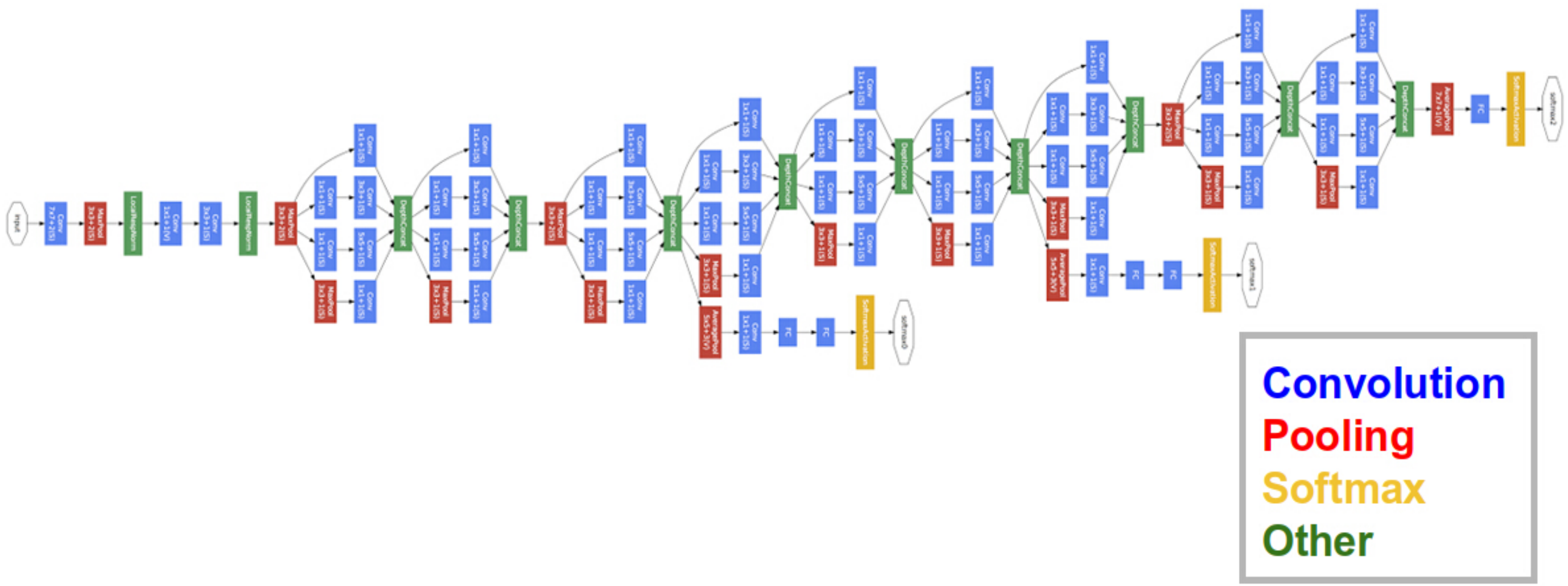
MUCH HAS CHANGED

1986



http://blog.refu.co/wp-content/uploads/2009/05/mlp.png

2016



THE LEARNING IS ALSO CHANGING

Supervised learning

(mostly machine)

A → B

Will be quickly solved for "easy" problems (Andrew Ng)

Unsupervised learning

…an)

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

$$\ldots(\mathbf{v}) \approx P_{data}(\mathbf{v})$$

Anywhere in between:
semi-supervised learning,
reinforcement learning,
lifelong learning.

STARTING POINT: FEATURE LEARNING

In typical machine learning projects, 80-90% effort is on feature engineering

- A right feature representation doesn't need much work. Simple linear methods often work well.

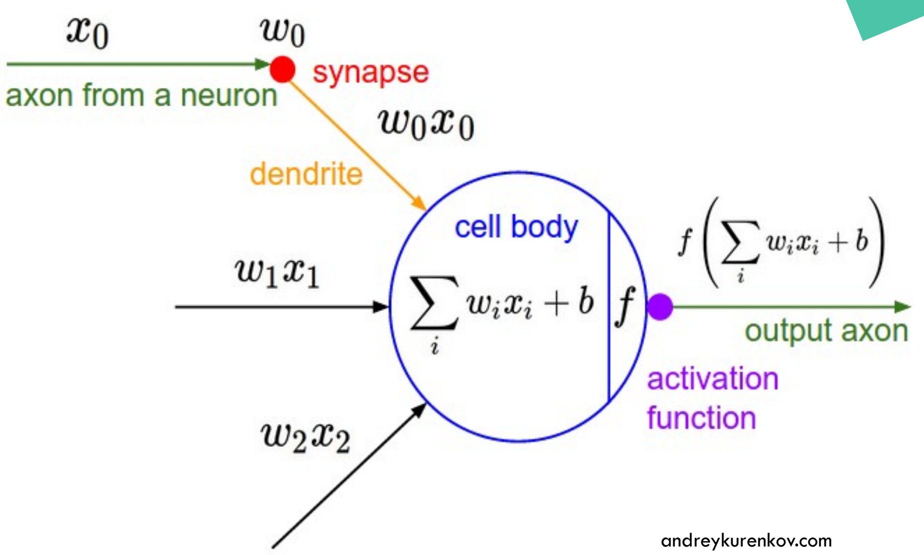
Text: BOW, n-gram, POS, topics, stemming, tf-idf, etc.

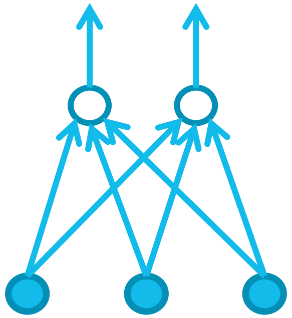
Software: token, LOC, API calls, #loops, developer reputation, team complexity, report readability, discussion length, etc.

Try yourself on Kaggle.com!

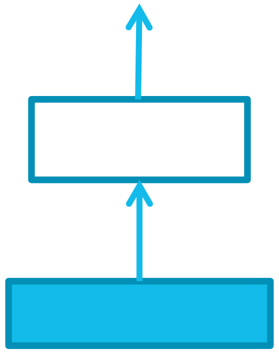
DEEP LEARNING AS FEATURE LEARNING

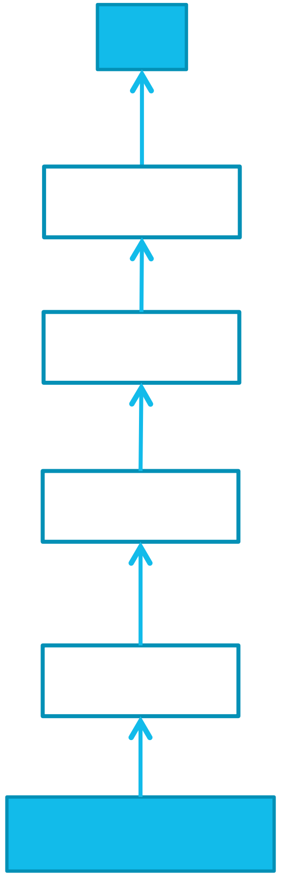
Integrate-and-fire neuron





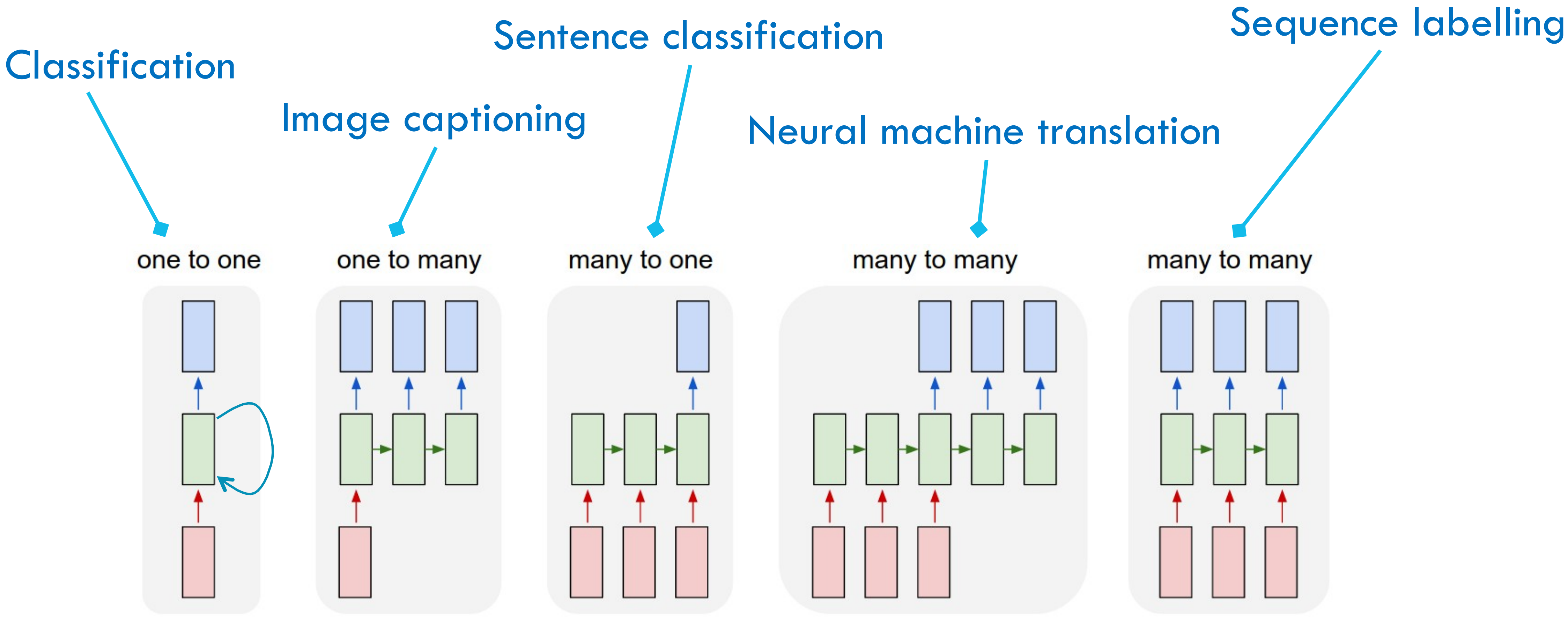
Feature detector





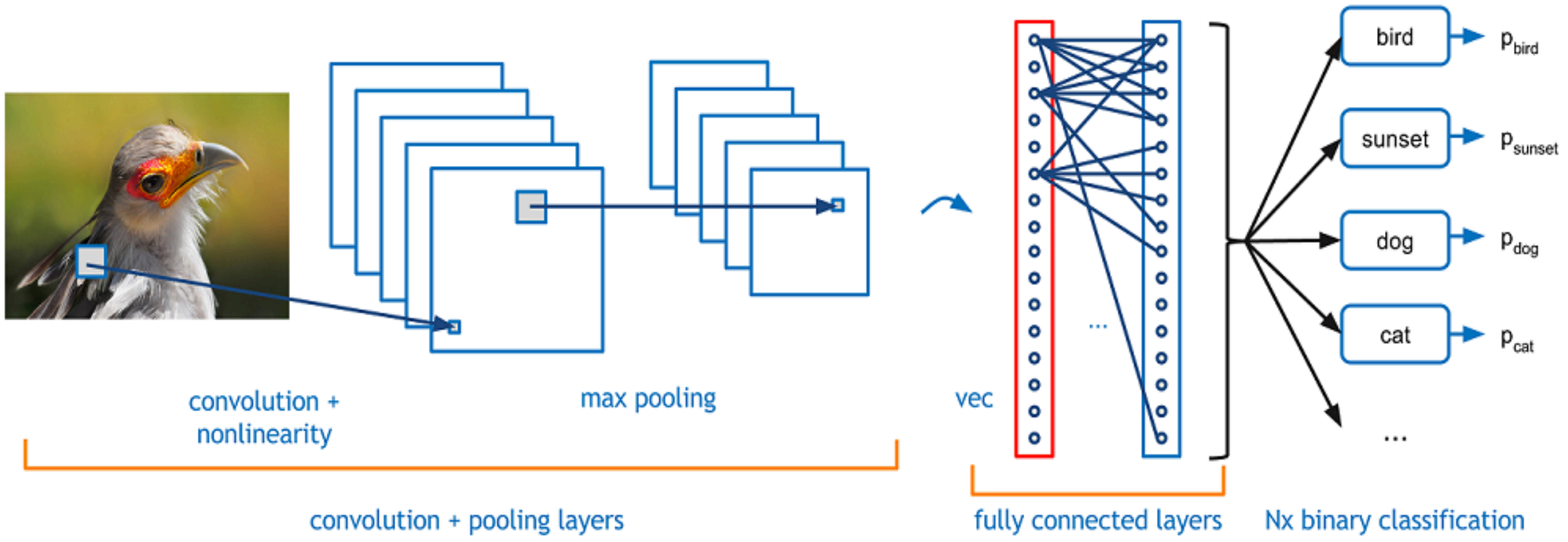
Block representation

RECURRENT NEURAL NETWORKS



Source: http://karpathy.github.io/assets/rnn/diags.jpeg

CONVOLUTIONAL NETS

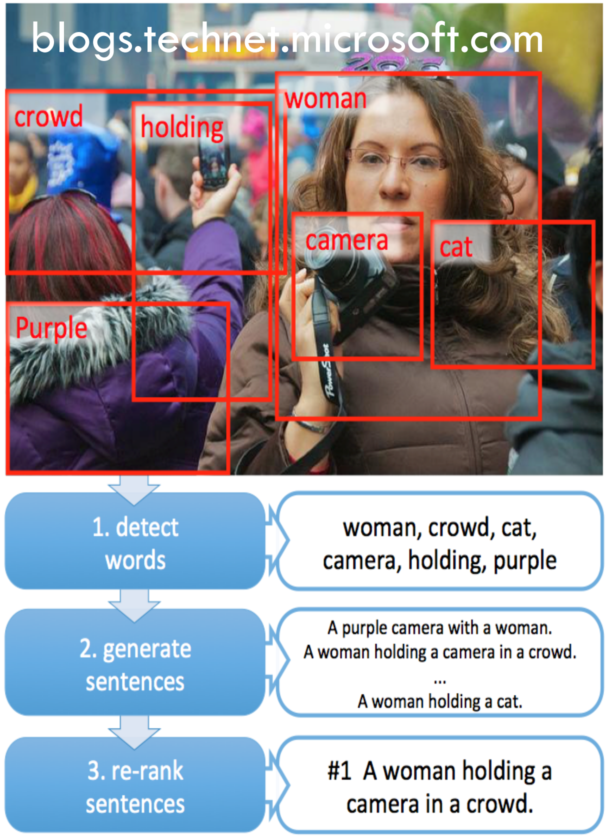


adeshpande3.github.io

DEEP LEARNING IN COGNITIVE DOMAINS

Google Translate

Where human can recognise, act or answer accurately within seconds



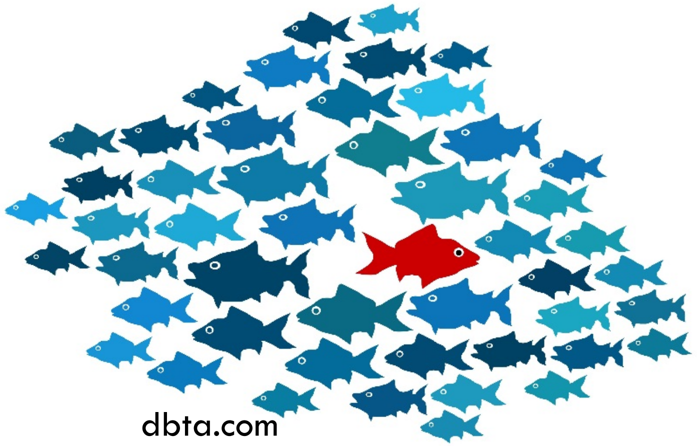
DEEP LEARNING IN NON-COGNITIVE DOMAINS

- **Where humans need extensive training to do well**
- Domains that demand transparency & interpretability.

… healthcare

… security

… genetics, foods, water …



WHY IT WORKS: PRINCIPLES

Expressiveness

- Can represent the complexity of the world → Feedforward nets are universal function approximator
- Can compute anything computable → Recurrent nets are Turing-complete

Learnability

- Have mechanism to learn from the training signals → Neural nets are highly trainable

Generalizability

- Work on unseen data → Deep nets systems work in the wild (Self-driving cars, Google Translate/Voice, AlphaGo)

WHY IT WORKS: PRACTICE

Strong/flexible priors (80-90% gain):

- Have good features → Feature engineering (Feature learning)
- Respect data structure → HMM, CRF, MRF, Bayesian nets (FFN, RNN, CNN)
- Theoretically motivated model structures, regularisation & sparsity → SVM, compressed sensing (Architecture engineering + hidden norm)
- Respect the manifold assumption, class/region separation → Metric + semi-supervised learning (Sesame net)
- Disentangle factors of variation → PCA, ICA, FA (RBM, DBN, DBM, DDAE, VAE, GAN, multiplicative neuron)

Uncertainty quantification (1-5% gain):

- Leverage Bayesian, ensemble → RF, GBM (Dropout, batch-norm, Bayesian neural nets)

Sharing statistical strength (1-10% gain):

- Encourage model reuse → transfer learning, domain adaption, multitask learning, lifelong learning (Column Bundle, Deep CCA, HyperNet, fast weight)

END OF PART I

WHAT IS ARCHITECTURE ENGINEERING?

The art and science of designing neural nets to better fit problem/task/data/performance structures

Examples:

SUPERVISED: FFN, CNN, RNN, Mem Net, Neural Turing Machine, Dynamic Mem Net, DeepCare, Deepr, Highway Net, LSTM, ResNet, HyperNet, DeepMat, Column Net, Column Bundle, TensorNet, etc.

UNSUPERVISED: RBM, DBN, DBM, DAE, DDAE, NADE, MADE, GAN, VAE, Moment Match Net, Ladder Net, etc.

TWO ISSUES IN LEARNING

1. Slow learning and local traps
   - Partly solved using Adaptive Stochastic Gradient Descents.
   - Better solved with Architecture engineering.
2. Data/model uncertainty and overfitting
   - Many models possible
   - Models are currently very big with hundreds of millions parameters
   - Deeper is more powerful, but more parameters.
   - **The best way to reduce model uncertainty: Architecture engineering**

POPULAR ARCHITECTURES

Image classification: CNN + FFN

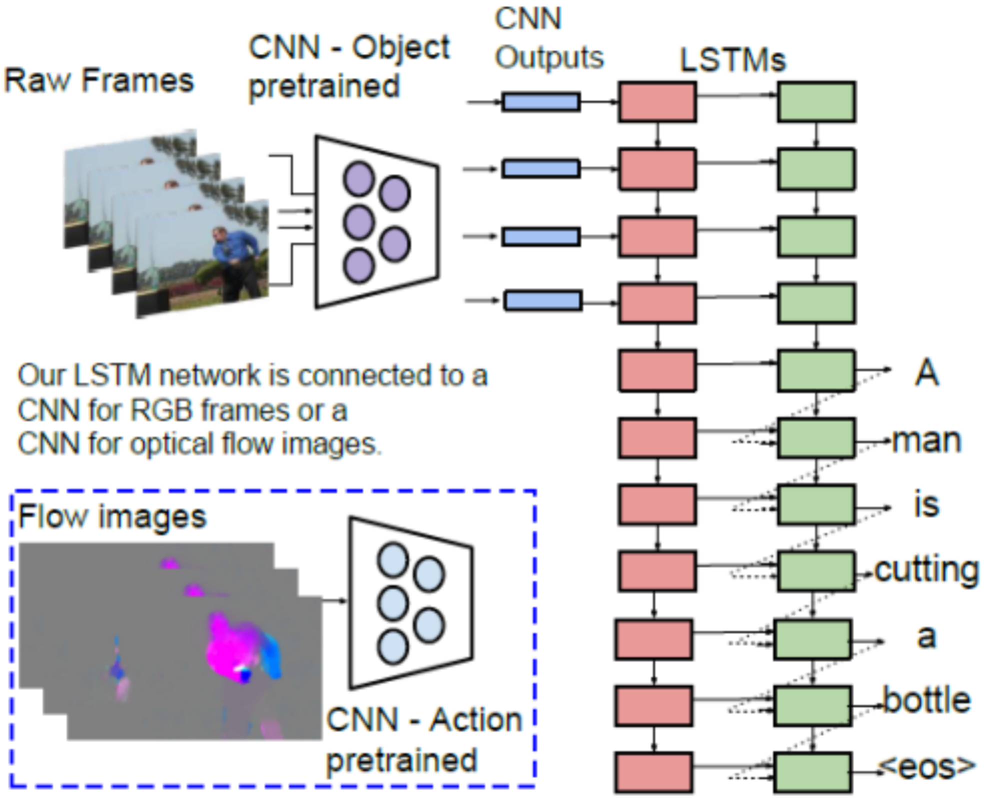
Video modelling: CNN + RNN

Image caption generation: CNN + RNN

Sentence classification: CNN + FFN

Sentence classification: RNN + FFN

Regular shapes (chain, tree, grid): CNN | RNN



https://vincentweisen.wordpress.com/2016/05/30/ammai-lecture-14-deep-learning-methods-for-image-captioning/

REGARDLESS OF PROBLEM TYPES, THERE ARE JUST ~~FOUR~~FIVE STEPS

Step 0: Collect LOTS of high-quality data

- Corollary: Spend LOTS of time, $$ and compute power

Step 1: Specify the **computational graph** Y = F(X; W)

Step 2: Specify the loss L(W; D) for data D = {(X1,Y1), (X2,Y2), … }

Step 3: Differentiate the loss w.r.t. W (now mostly automated)

Step 4: Optimize the loss (a lot of tools available)

SPECIFY COMPUTATIONAL GRAPHS

Everything is a <u>computational graph</u> from end-to-end.

Each block has an input and an output, and some tensor operators.

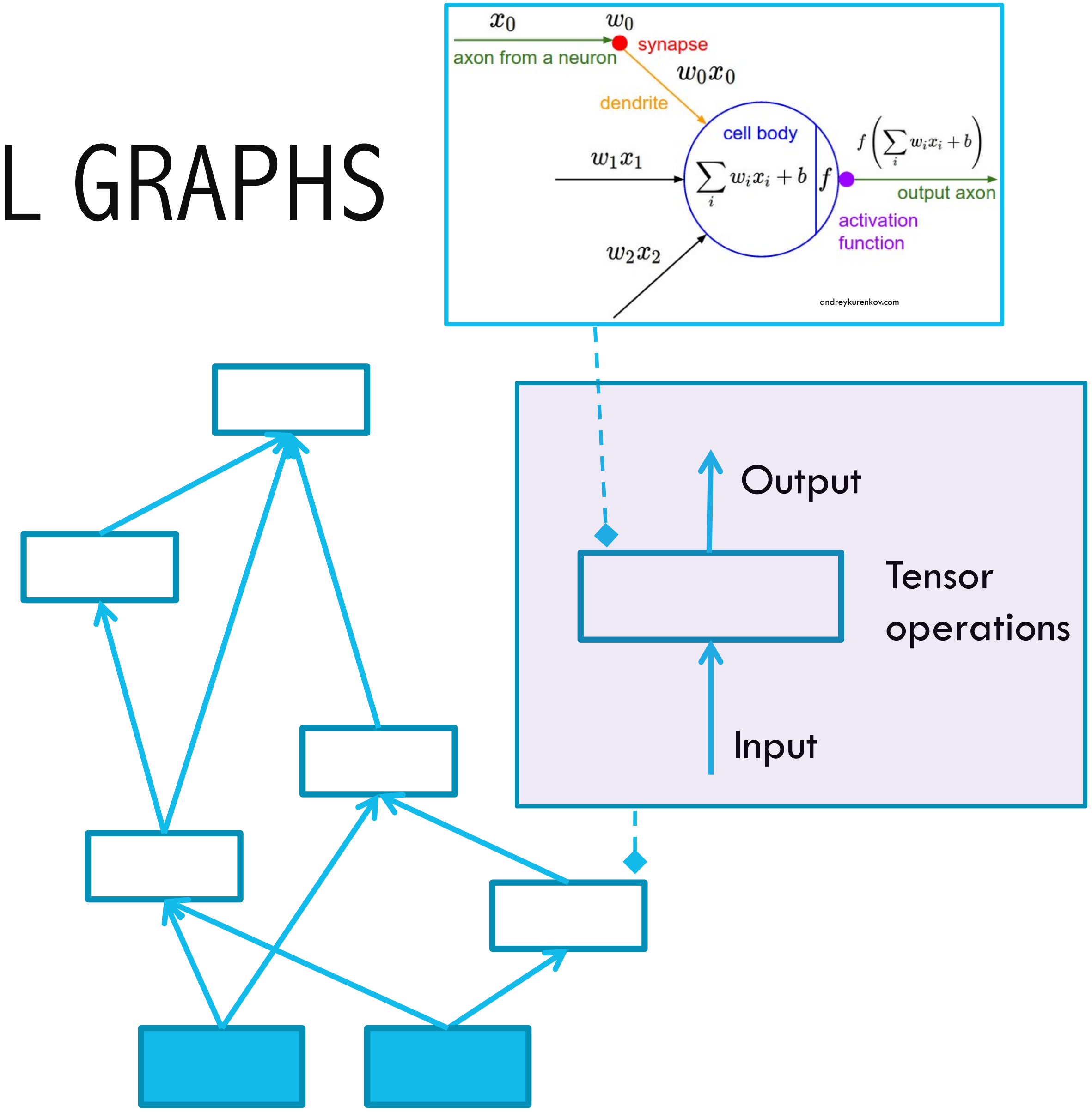
- Hence the name <u>Tensor</u>Flow.

Vectors (1D), matrices (2D) and tensors N-D) operations

Element-wise transforms

Automatic differentiation naturally supported

... It is Lego building exercise!



DEEP LEARNING AS NEW ELECTRONICS

DEEP LEARNING AS NEW ELECTRONICS

Analogies:

- Neuron as feature detector → SENSOR, FILTER
- Multiplicative gates → AND gate, Transistor, Resistor
- Attention mechanism → SWITCH gate
- Memory + forgetting → Capacitor + leakage
- Skip-connection → Short circuit
- Computational graph → Circuit
- Compositionality → Modular design

Relationships

- **Now**: Electronics redesigned to support tensors in deep learning
- Prediction: Deep learning helps to design faster electronics

#ARCHITECTURE-ENGINEERING @PRADA

Flexible gates (p-norm)

Sequences (Long-deep highway)

Events/episodes + intervention (DeepCare)

Predictive motifs (Deepr)

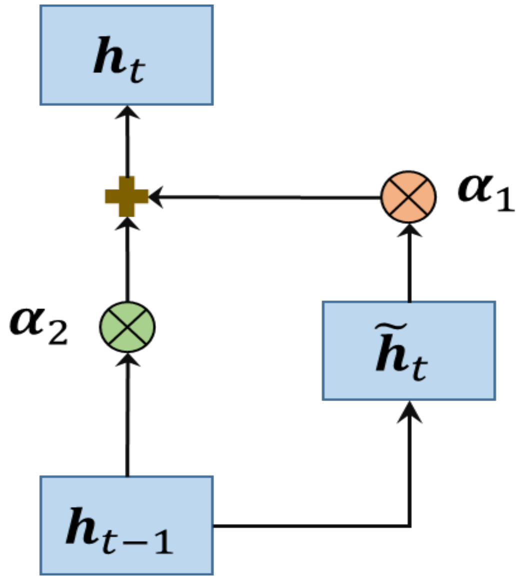
Matrices (DeepMat)

Graphs & relations (Column Nets)

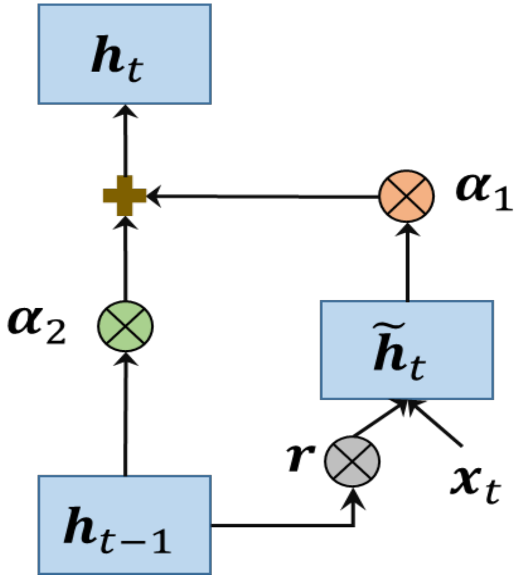
Permutation (Neural Choice)

Multi-X (Column Bundle)

Highway networks and Gated Recurrent Units



$$\begin{aligned} h_t &= \alpha_1 * \tilde{h}_t + \alpha_2 * h_{t-1} \\ 1 &= \alpha_1 + \alpha_2 \end{aligned}$$



$$\tilde{h}_t = g\left(W h_{t-1} + b\right)$$

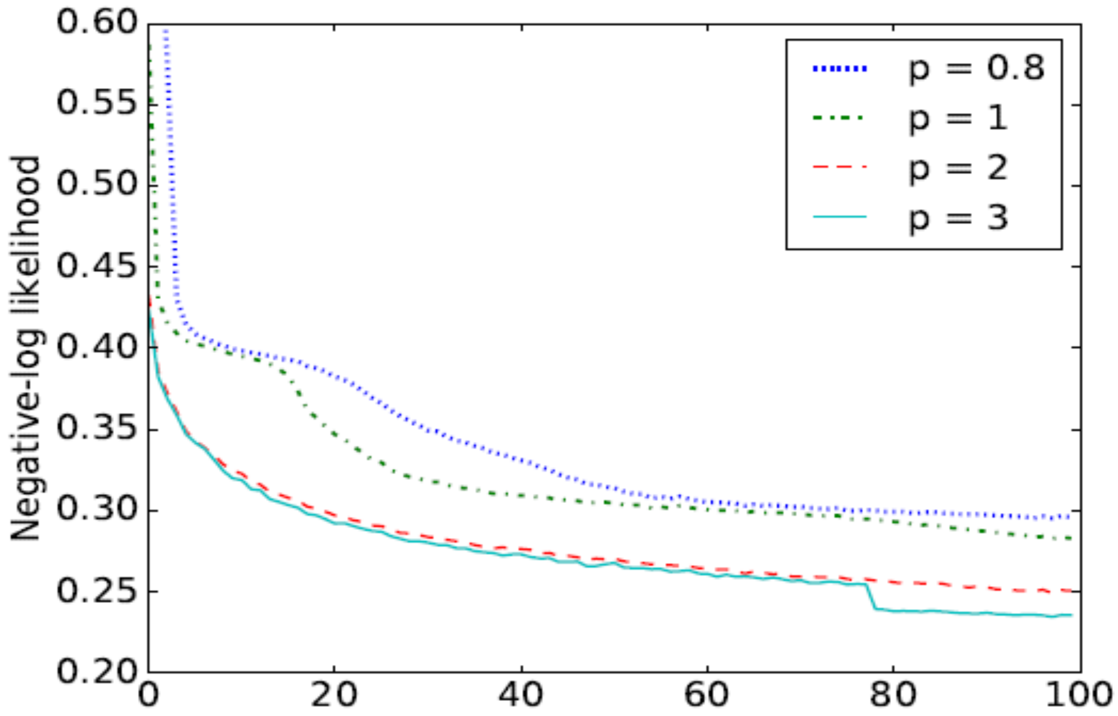
$$\begin{aligned} r_t &= \sigma\left(W_r x_t + U_r h_{t-1} + b_r\right) \\ \tilde{h}_t &= \tanh\left(W_h x_t + U_h\left(r_t * h_{t-1}\right) + b_h\right) \end{aligned}$$

A VERY SIMPLE SOLUTION: P-NORM

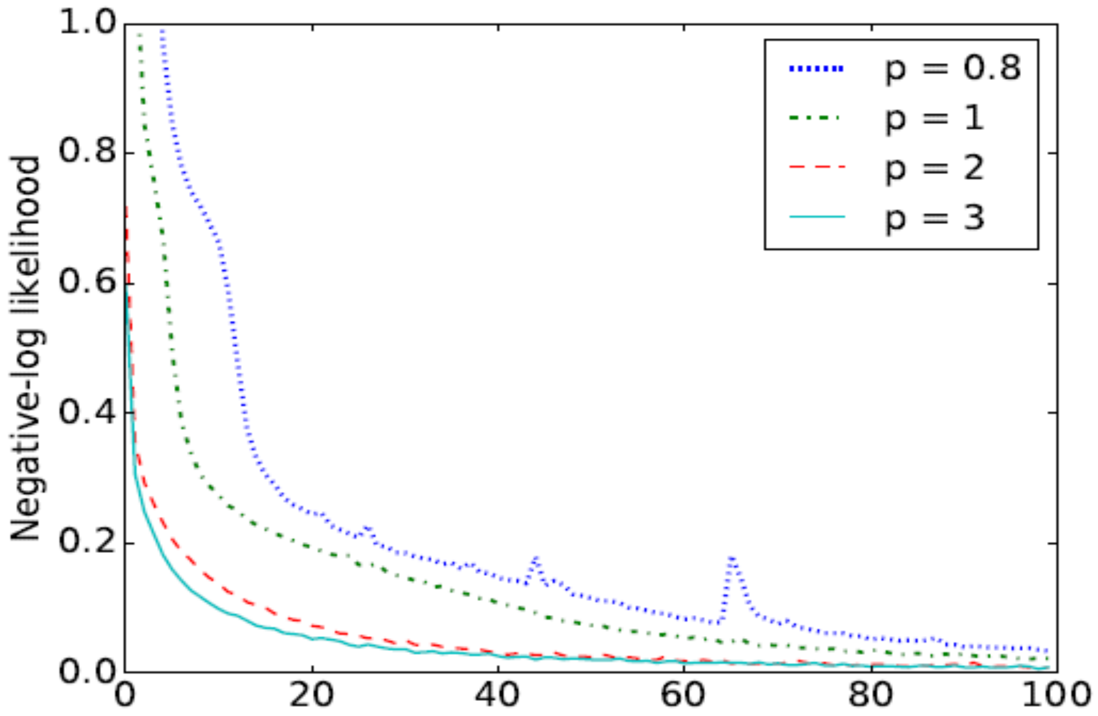
$$(\boldsymbol{\alpha}_1^p + \boldsymbol{\alpha}_2^p)^{\frac{1}{p}} = \mathbf{1}, \quad \text{equivalently:} \quad \boldsymbol{\alpha}_2 = (\mathbf{1} - \boldsymbol{\alpha}_1^p)^{\frac{1}{p}}$$

$$p = 5 \quad \boldsymbol{\alpha}_1 = \mathbf{0.9} \qquad \boldsymbol{\alpha}_2 = \mathbf{0.865}$$

p-norm + highway network for vector data



(a) MiniBoo



(b) Sensorless

p-norm + highway network for vector data

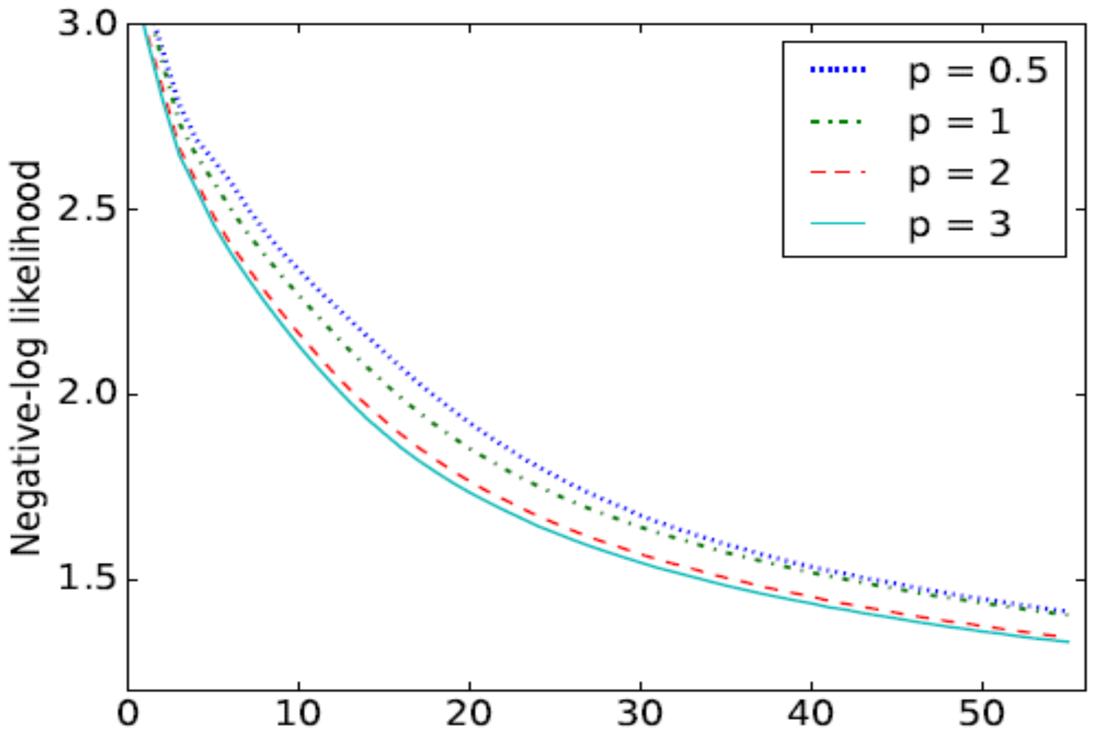
(a) MiniBoo dataset

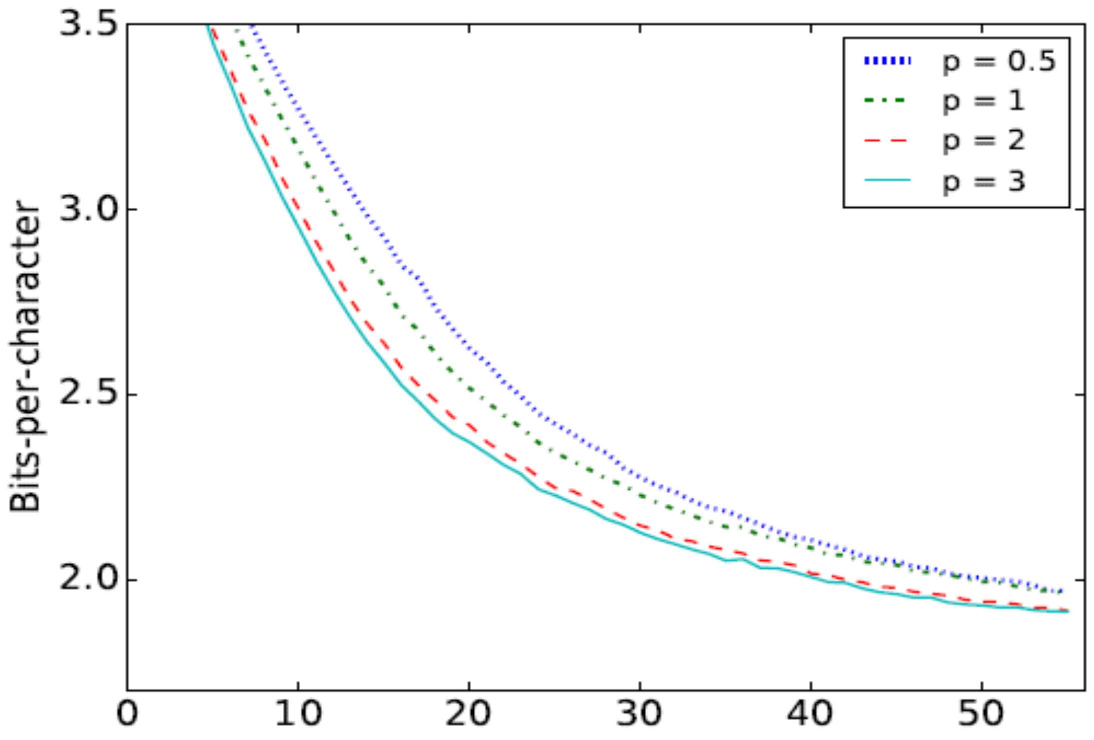
p	epochs to 89%	F1-score (%)
0.8	N/A	88.5
1	94	89.1
2	**33**	90.2
3	**33**	**90.4**

(b) Sensorless dataset

p	epochs to 99%	macro F1-score (%)
0.8	92	99.1
1	77	99.4
2	41	99.7
3	**35**	**99.7**

p-norm + GRU for sequential data

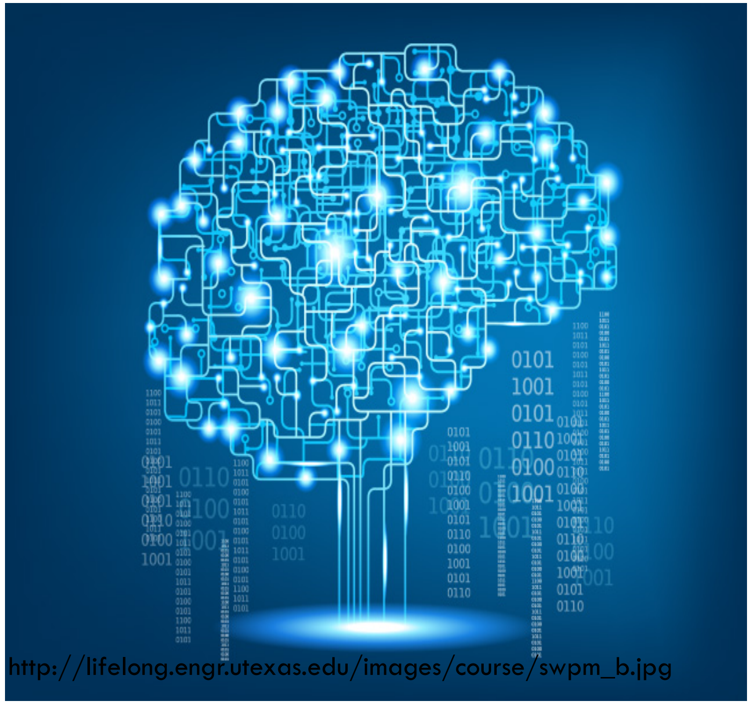




PART II: ARCHITECTURE ENGINEERING

- Flexible gates (p-norm)
- Sequences (LSTM + Long-deep)
- Episodes + intervention (DeepCare)
- Predictive motifs (Deepr + DeepURL)
- Matrices (DeepMat)
- Graphs & relations (Column Nets)
- Permutation (Neural Choice)
- Multi-X (Column Bundle)

TOWARDS INTELLIGENT ASSISTANTS IN SOFTWARE ENGINEERING



Motivations: Software is eating the world. Open source codebases are very rich and large.

Goal: To model code, text, team, user, execution, project & enabled business process → answer any queries by developers, managers, users and business

For now:

- LSTM for code language model
- LD-RNN for report representation
- Stacked/deep inference (later)

A DEEP LANGUAGE MODEL FOR SOFTWARE CODE (DAM ET AL, FSE'16 SE+NL)

A good language model for source code would capture the long-term dependencies

The model can be used for various prediction tasks, e.g. defect prediction, code duplication, bug localization, etc.

The model can be extended to model software and its development process.

UNIVERSITY OF WOLLONGONG AUSTRALIA

DEAKIN UNIVERSITY AUSTRALIA

CHARACTERISTICS OF SOFTWARE CODE

Repetitiveness

- E.g. for (int i = 0; i < n; i++)

Localness

- E.g. *for (int size* may appear more often that *for (int i* in some source files.

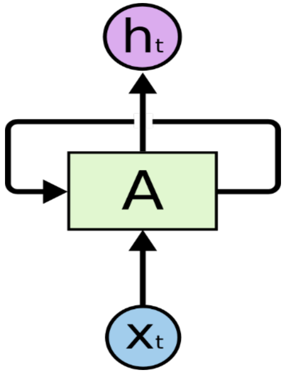
Rich and explicit structural information

- E.g. nested loops, inheritance hierarchies

Long-term dependencies

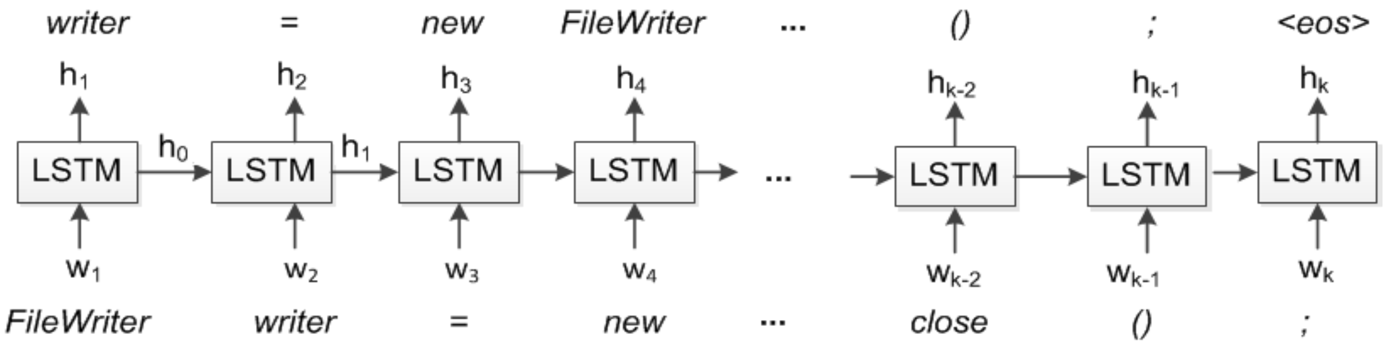
- *try* and *catch* (in Java) or file *open* and *close* are not immediately followed each other.

CODE LANGUAGE MODEL




```
FileWriter writer = new FileWriter(file);
writer.write(``This is an example'');
int count = 0;
System.out.prinltln(``Long gap'');
    ......

writer.flush();
writer.close();
```



Previous work has applied RNNs to model software code *(White et al, MSR 2015)*

RNNs however do not capture the long-term dependencies in code

EXPERIMENTS

Built dataset of 10 Java projects: Ant, Batik, Cassandra, Eclipse-E4, Log4J, Lucene, Maven2, Maven3, Xalan-J, and Xerces.

Comments and blank lines removed. Each source code file is tokenized to produce a sequence of code tokens.

- Integers, real numbers, exponential notation, hexadecimal numbers replaced with <num> token, and constant strings replaced with <str> token.
- Replaced less "popular" tokens with <unk>

Code corpus of 6,103,191 code tokens, with a vocabulary of 81,213 unique tokens.

EXPERIMENTS (CONT.)

sent-len	embed-dim	RNN	LSTM	improv %
10	50	13.49	12.86	**4.7**
20		10.38	9.66	**6.9**
50		7.93	6.81	**14.1**
100		7.20	6.40	**11.1**
200		6.64	5.60	**15.7**
500		6.48	4.72	**27.2**
100	20	7.96	7.11	**10.7**
	50	7.20	6.40	**11.1**
	100	7.23	5.72	**20.9**
	200	9.14	5.68	**37.9**

Table 1: Perplexity on test data (the smaller the better).

Both RNN and LSTM improve with more training data (whose size grows with sequence length).

LSTM consistently performs better than RNN: 4.7% improvement to 27.2% (varying sequence length), 10.7% to 37.9% (varying embedding size).

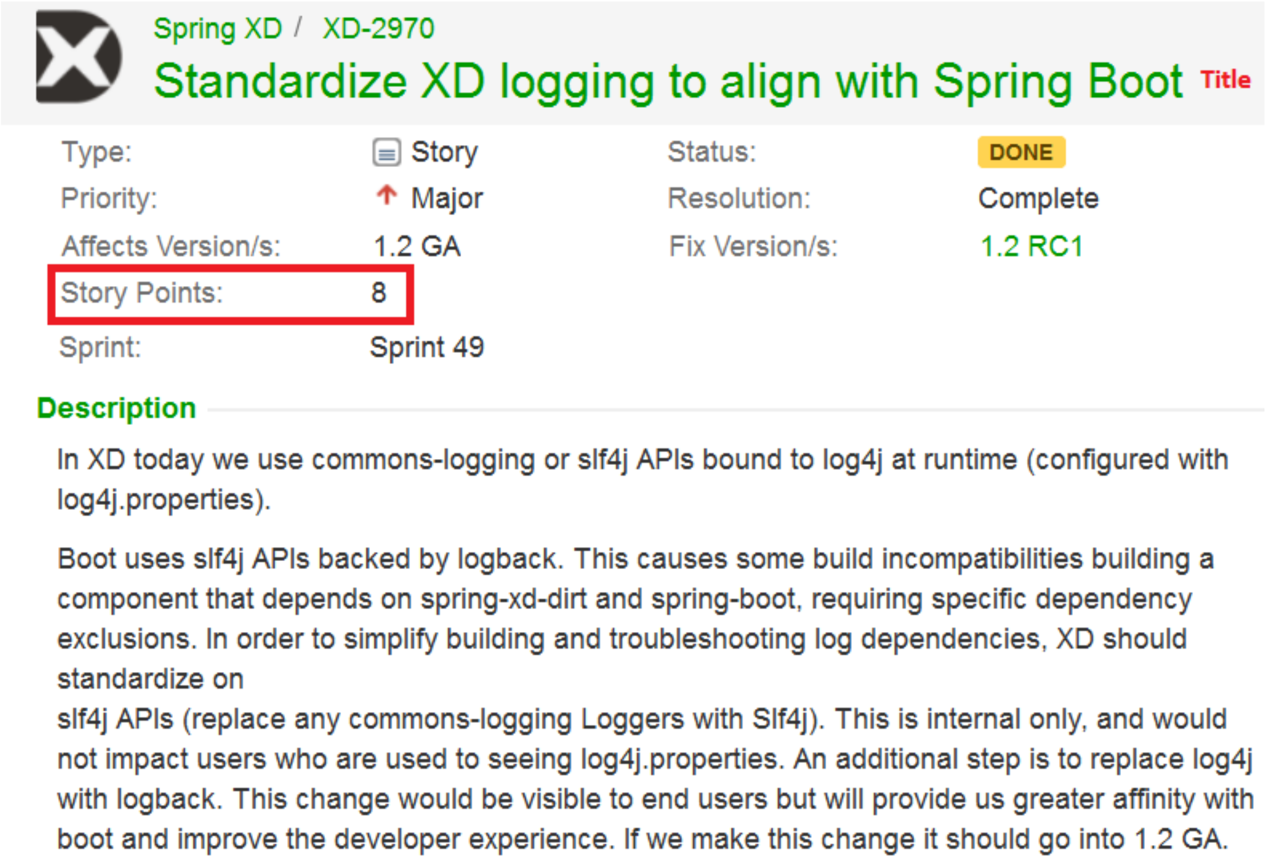
STORY POINT ESTIMATION

Traditional estimation methods require experts, LOC or function points.

- Not applicable early
- Expensive

Feature engineering is not easy!

Needs a cheap way to start from just a documentation.



Spring XD / XD-2970

Standardize XD logging to align with Spring Boot Title

Type:	Story	Status:	DONE
Priority:	Major	Resolution:	Complete
Affects Version/s:	1.2 GA	Fix Version/s:	1.2 RC1
Story Points:	8		
Sprint:	Sprint 49		

Description

In XD today we use commons-logging or slf4j APIs bound to log4j at runtime (configured with log4j.properties).

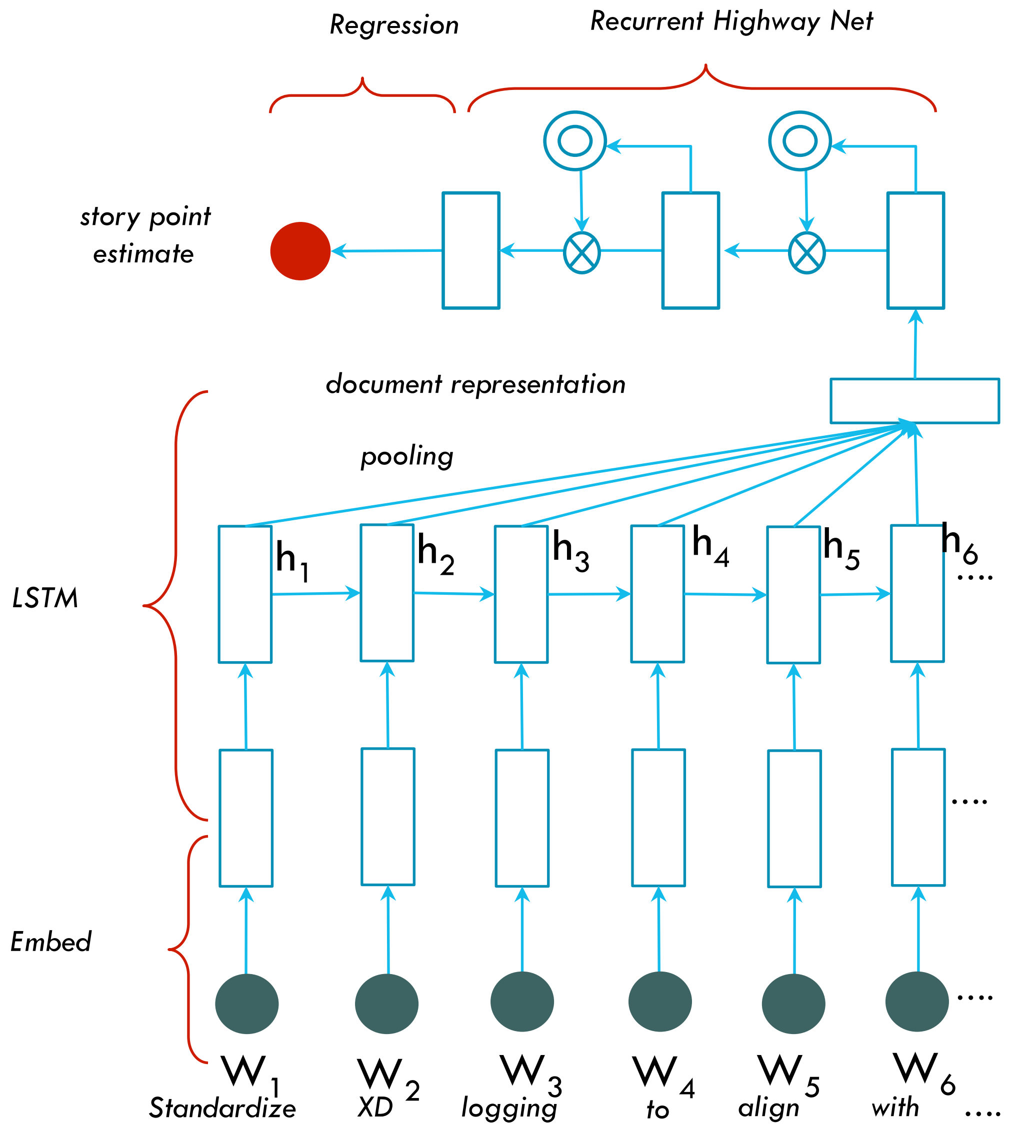
Boot uses slf4j APIs backed by logback. This causes some build incompatibilities building a component that depends on spring-xd-dirt and spring-boot, requiring specific dependency exclusions. In order to simplify building and troubleshooting log dependencies, XD should standardize on
slf4j APIs (replace any commons-logging Loggers with Slf4j). This is internal only, and would not impact users who are used to seeing log4j.properties. An additional step is to replace log4j with logback. This change would be visible to end users but will provide us greater affinity with boot and improve the developer experience. If we make this change it should go into 1.2 GA.

LD-RNN FOR REPORT REPRESENTATION
(CHOETKIERTIKUL ET AL, WORK IN PROGRESS)

LD = Long Deep

LSTM for document representation

Highway-net with tied parameters for story point estimation



RESULTS

MAE = Mean Absolute Error

$$SA = \left(1 - \frac{MAE}{MAE_{rguess}}\right) \times 100$$

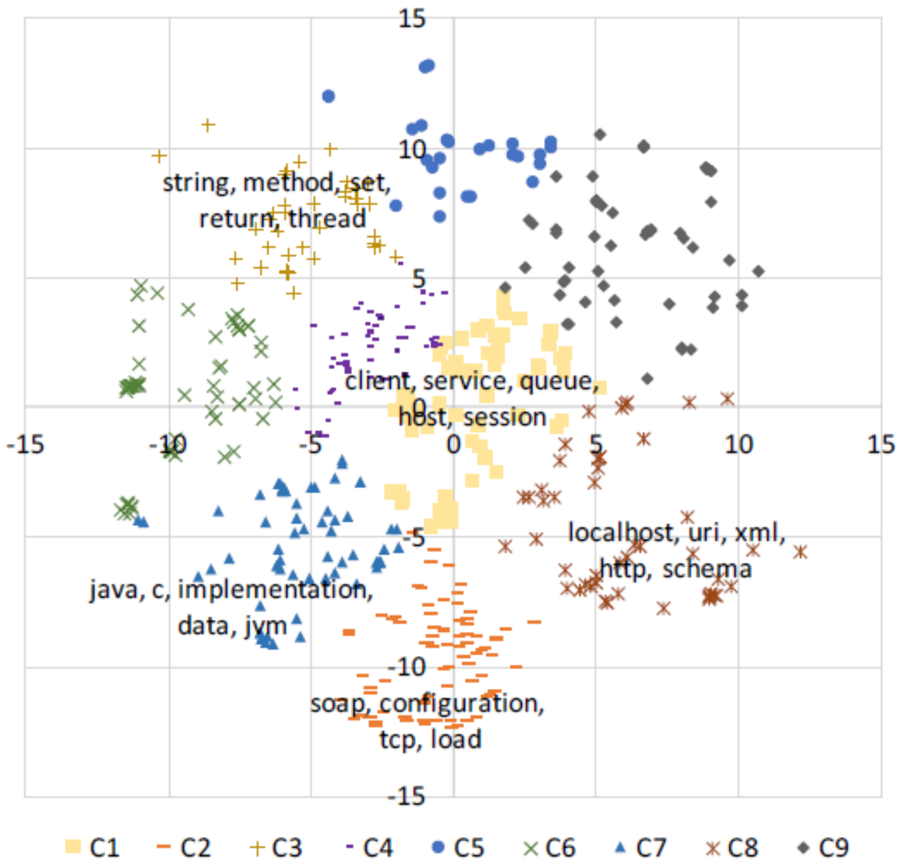


Fig. 4. Top-500 word clusters used in the Apache's issue reports

Proj	Technique	MAE	SA	Proj	Technique	MAE	SA
ME	LD-RNN	**1.02**	**59.03**	JI	LD-RNN	**1.38**	**59.52**
	LSTM+RF	1.08	57.57		LSTM+RF	1.71	49.71
	BoW+RF	1.31	48.66		BoW+RF	2.10	38.34
	Mean	1.64	35.61		Mean	2.48	27.06
	Median	1.73	32.01		Median	2.93	13.88
UG	LD-RNN	**1.03**	**52.66**	MD	LD-RNN	**5.97**	**50.29**
	LSTM+RF	1.07	50.70		LSTM+RF	9.86	17.86
	BoW+RF	1.19	45.24		BoW+RF	10.20	15.07
	Mean	1.48	32.13		Mean	10.90	9.16
	Median	1.60	26.29		Median	7.18	40.16
AS	LD-RNN	**1.36**	**60.26**	DM	LD-RNN	**3.77**	**47.87**
	LSTM+RF	1.62	52.38		LSTM+RF	4.51	37.71
	BoW+RF	1.83	46.34		BoW+RF	4.78	33.84
	Mean	2.08	39.02		Mean	5.29	26.85
	Median	1.84	46.17		Median	4.82	33.38
AP	LD-RNN	**2.71**	**42.58**	MU	LD-RNN	**2.18**	**40.09**
	LSTM+RF	2.97	37.09		LSTM+RF	2.23	38.73
	BoW+RF	2.96	37.34		BoW+RF	2.31	36.64
	Mean	3.15	33.30		Mean	2.59	28.82
	Median	3.71	21.54		Median	2.69	26.07

PART II: ARCHITECTURE ENGINEERING

Flexible gates (p-norm)

Sequences (Deep-long highway)

Episodes + intervention (DeepCare)

Predictive motifs (Deepr + Deepic + DeepURL)

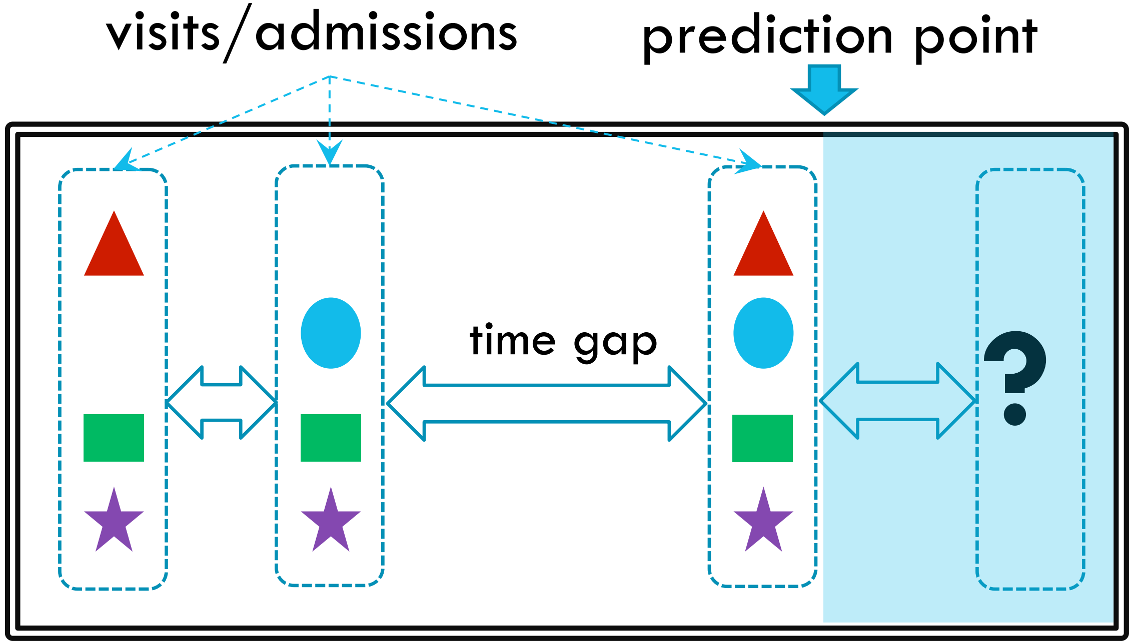
Matrices (DeepMat)

Graphs & relations (Column Nets)

Permutation (Neural Choice)

Multi-X (Column Bundle)

PREDICTIVE HEALTH USING ELECTRONIC MEDICAL RECORDS (EMR)



- Time-stamped
- Coded data: diagnosis, procedure & medication
- Text not considered, but in principle can be mapped in to vector using LSTM

EMR CHALLENGES

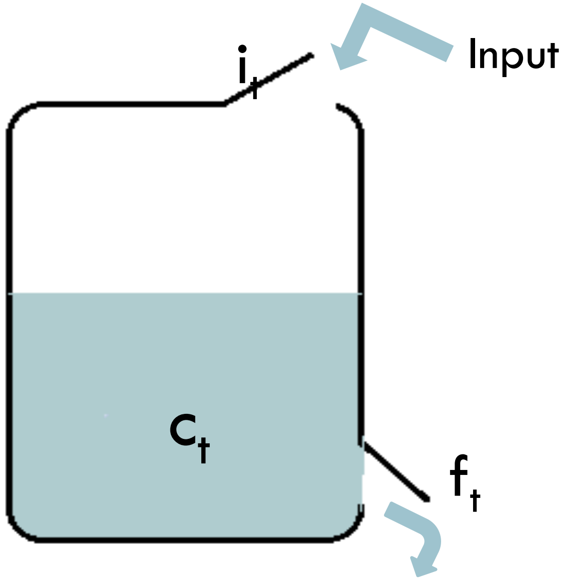
Long-term dependencies

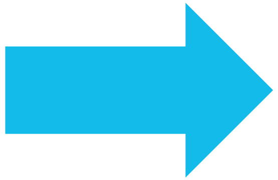
Episodic, event-based with time-stamps

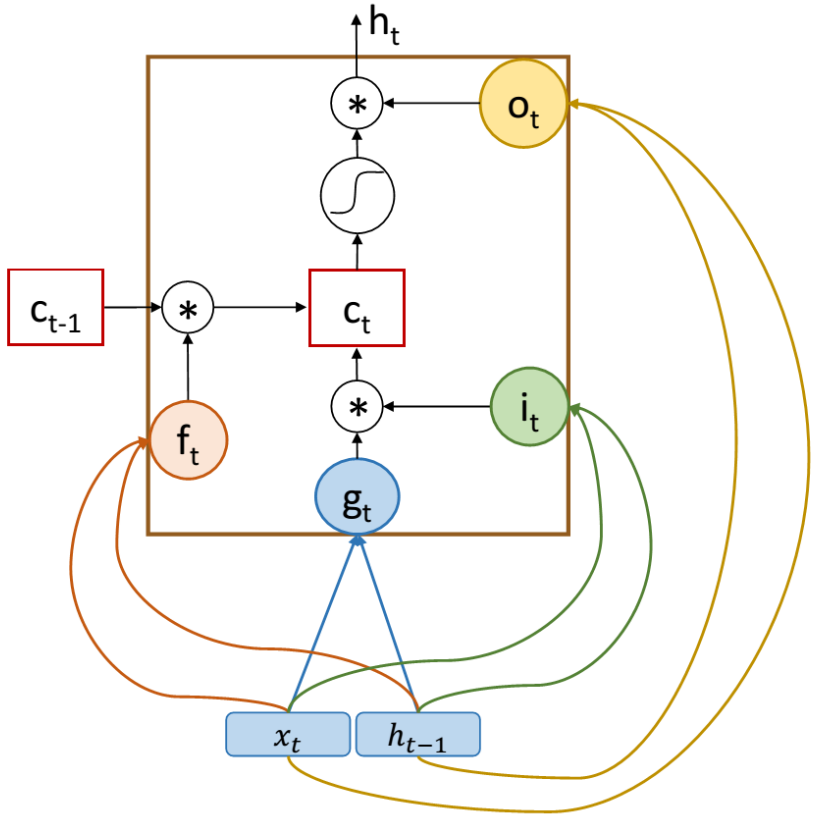
Interventions change the natural dynamics of diseases

Each EMR is a sequence of sets of multiple types

LONG SHORT-TERM MEMORY (LSTM)

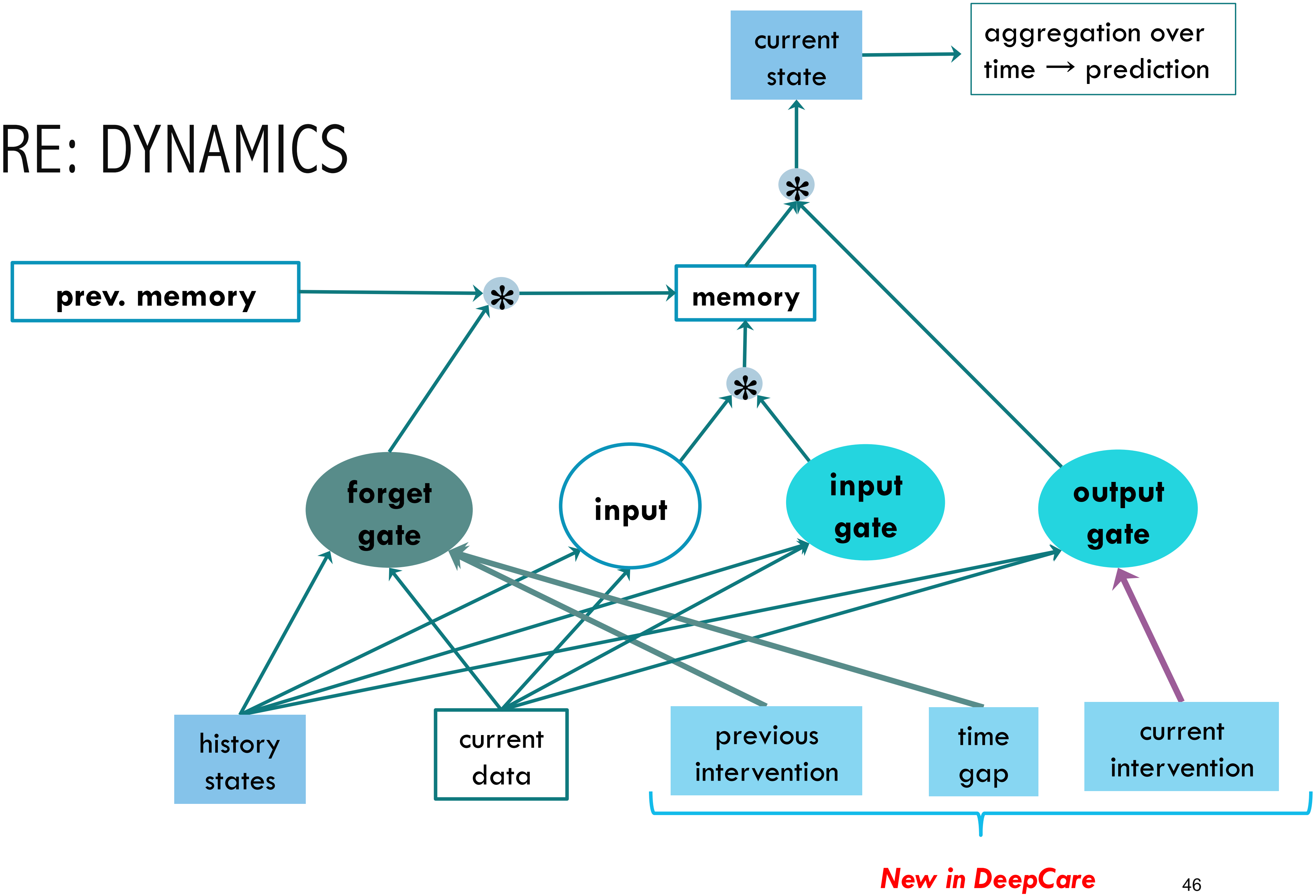




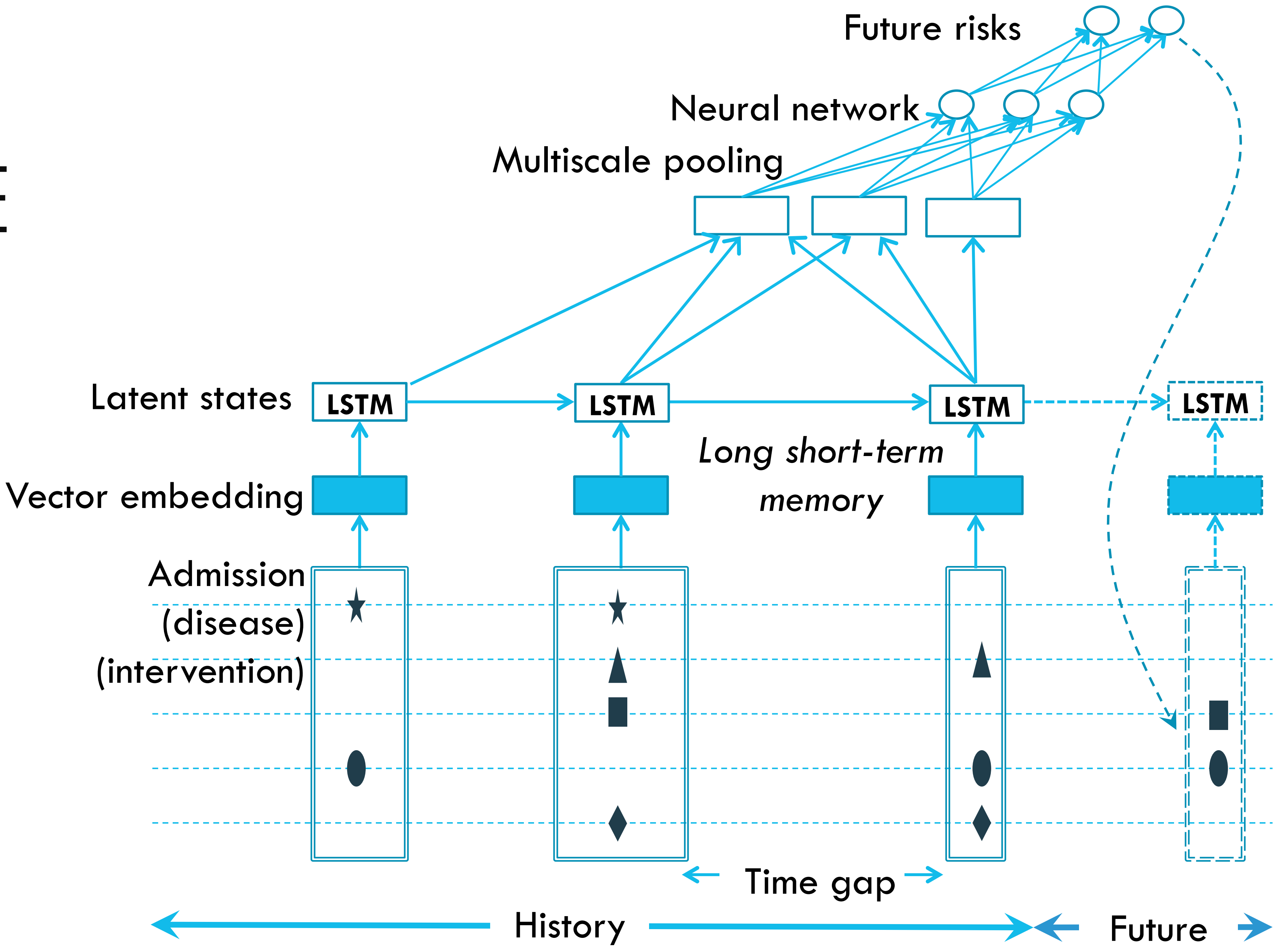


$$\boldsymbol{i}_t = \sigma\left(W_i \boldsymbol{x}_t + U_i \boldsymbol{h}_{t-1} + \boldsymbol{b}_i\right)$$
$$\boldsymbol{f}_t = \sigma\left(W_f \boldsymbol{x}_t + U_f \boldsymbol{h}_{t-1} + \boldsymbol{b}_f\right)$$
$$\boldsymbol{o}_t = \sigma\left(W_o \boldsymbol{x}_t + U_o \boldsymbol{h}_{t-1} + \boldsymbol{b}_o\right)$$
$$\boldsymbol{c}_t = \boldsymbol{f}_t * \boldsymbol{c}_{t-1} + \boldsymbol{i}_t * \tanh\left(W_c \boldsymbol{x}_t + U_c \boldsymbol{h}_{t-1} + \boldsymbol{b}_c\right)$$
$$\boldsymbol{h}_t = \boldsymbol{o}_t * \tanh(\boldsymbol{c}_t)$$

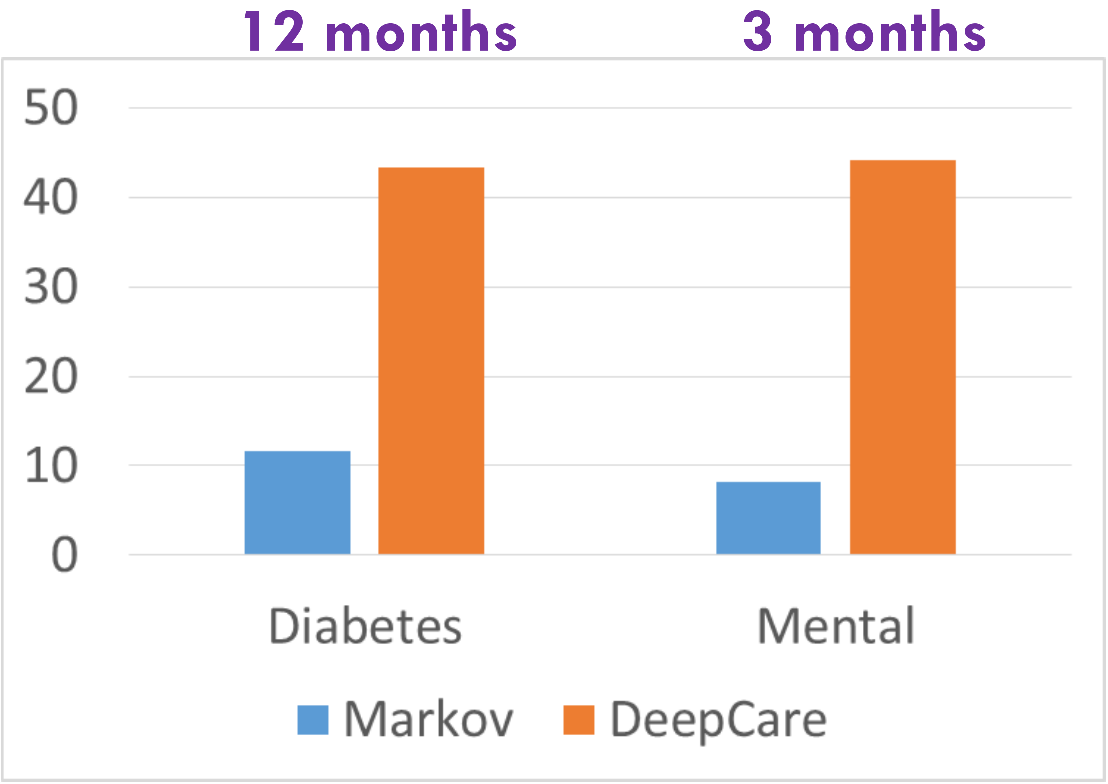
DEEPCARE: DYNAMICS



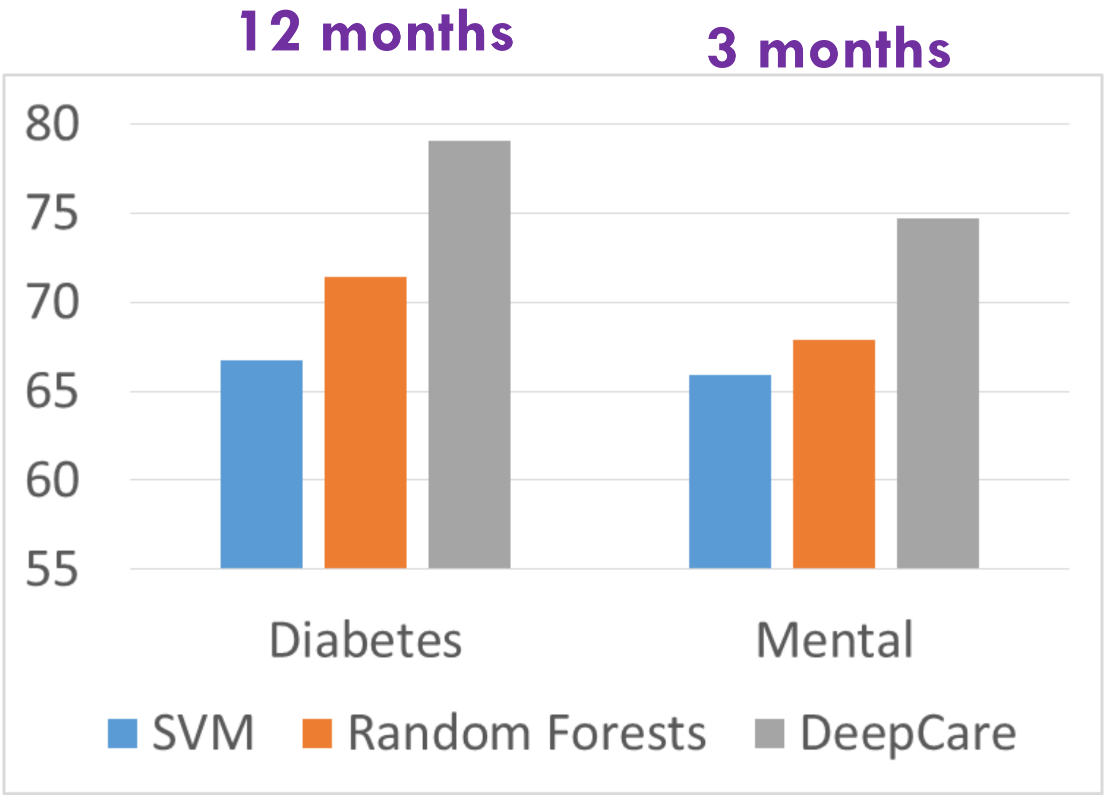
DEEPCARE: STRUCTURE



DEEPCARE: PREDICTION RESULTS



Intervention recommendation (precision@3)



Unplanned readmission prediction (F-score)

PART II: ARCHITECTURE ENGINEERING

Flexible gates (p-norm)

Sequences (Deep-long highway)

Episodes + intervention (DeepCare)

Predictive motifs (Deepr + Deepic + DeepURL)

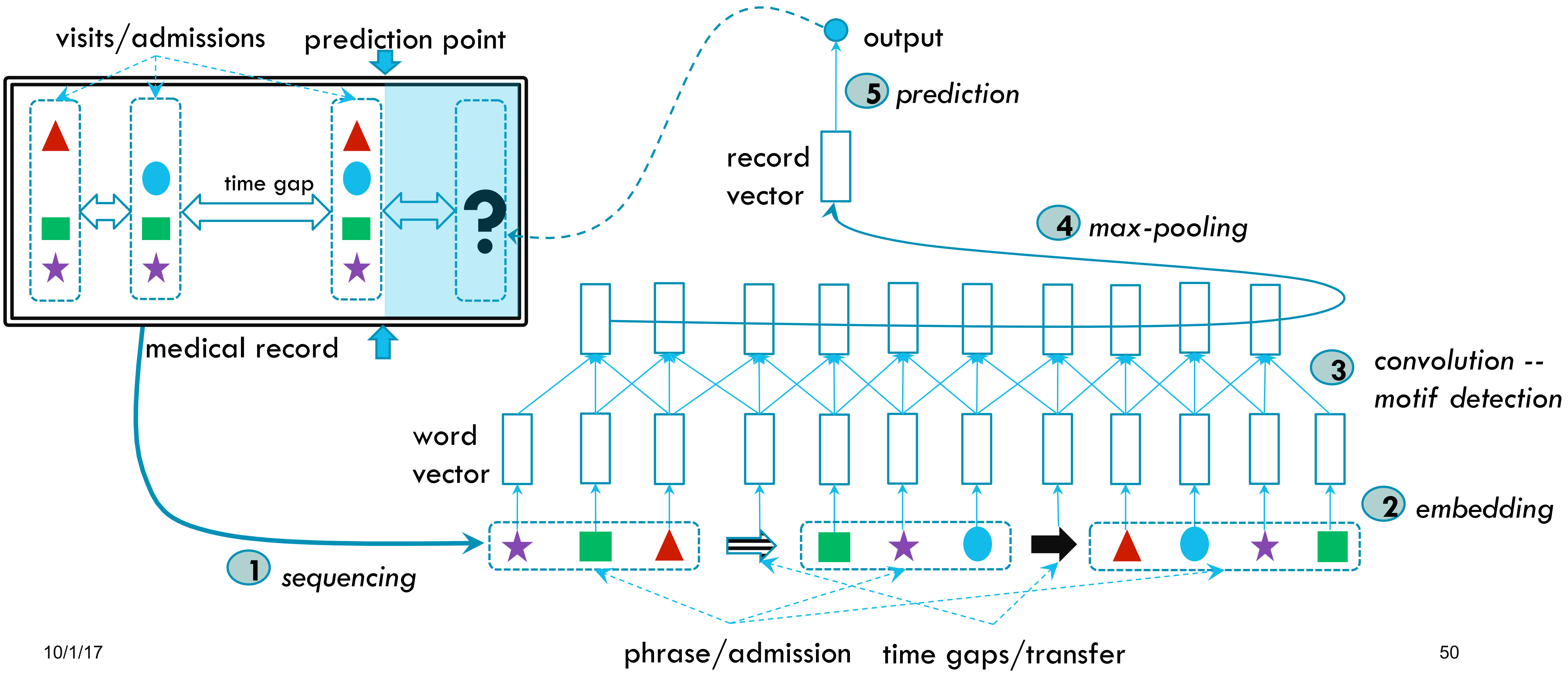
Matrices (DeepMat)

Graphs & relations (Column Nets)

Permutation (Neural Choice)

Multi-X (Column Bundle)

DEEPR: CNN FOR REPEATED MOTIFS AND SHORT SEQUENCES (NGUYEN ET AL, J-BHI, 2016)



DISEASE EMBEDDING & MOTIFS DETECTION

E11 + I48 + I50

Type 2 diabetes mellitus

Atrial fibrillation and flutter

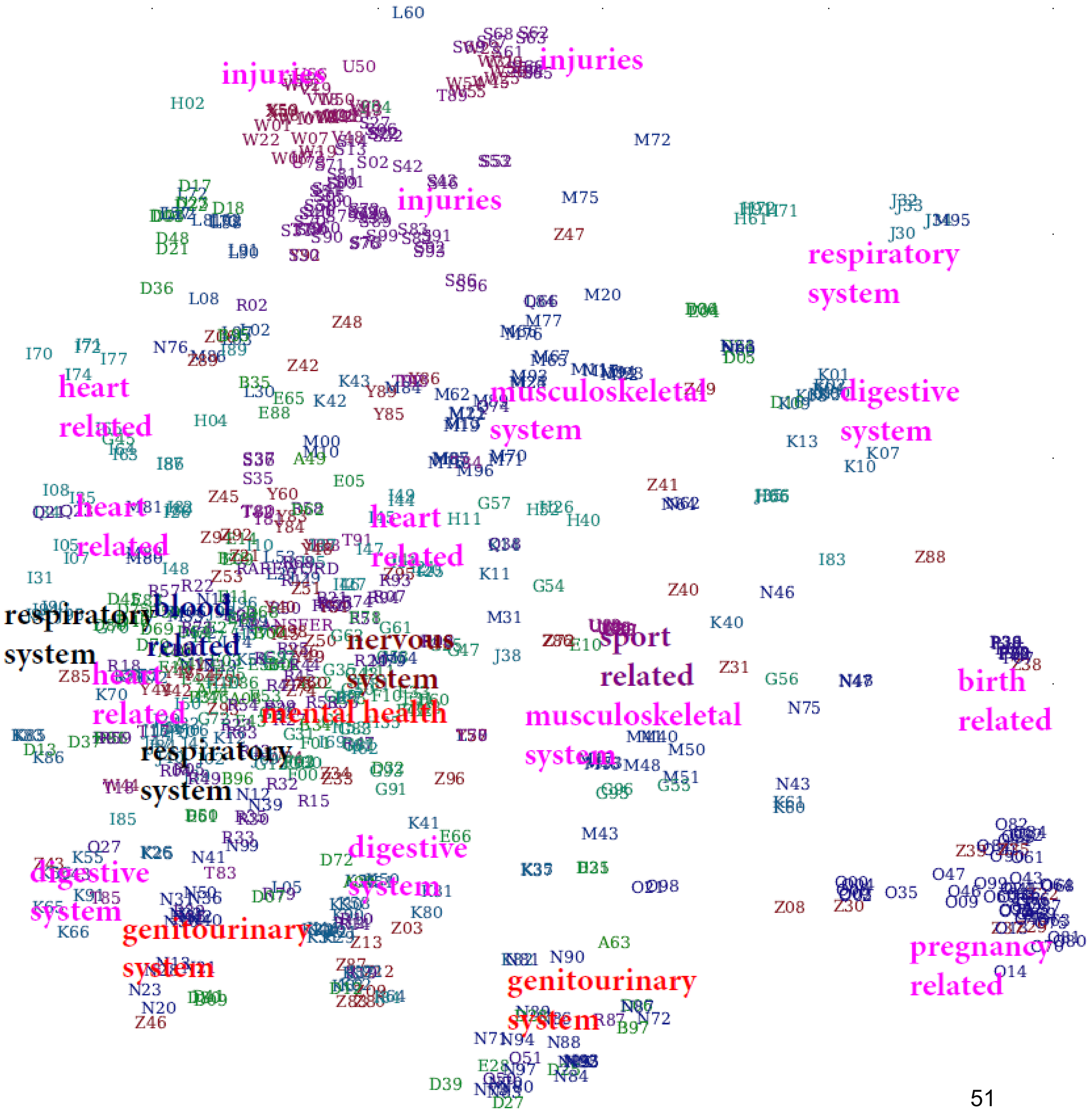
Heart failure

E11 + I50 + N17

Type 2 diabetes mellitus

Heart failure

Acute kidney failure



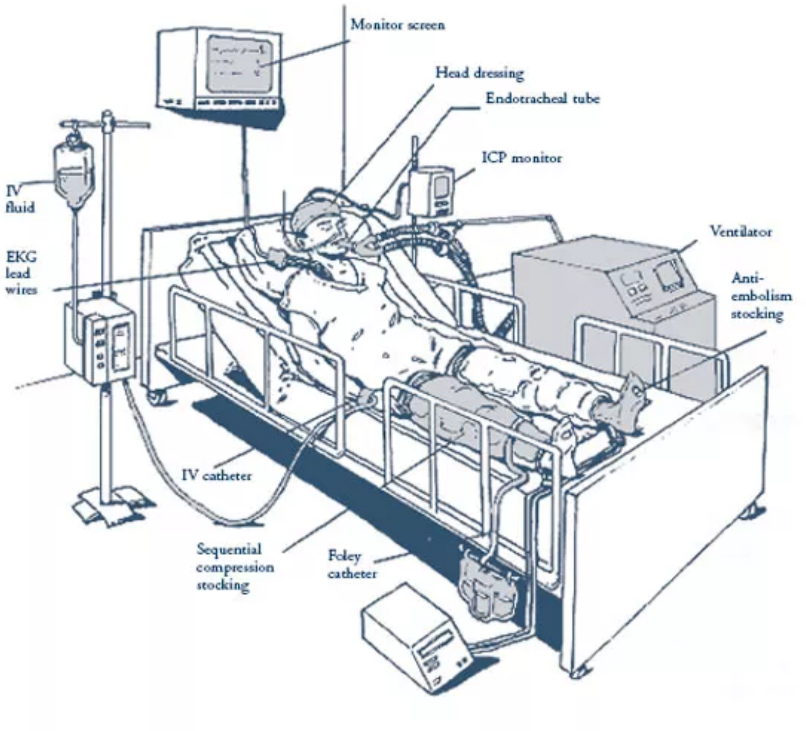
DEEPIC: MORTALITY PREDICTION IN INTENSIVE CARE UNITS (WORK IN PROGRESS)

Existing methods: LSTM with missingness and time-gap as input.

New method: **Deepic**

Steps:

- Measurement quantization
- Time gap quantization
- Sequencing words into sentence
- CNN

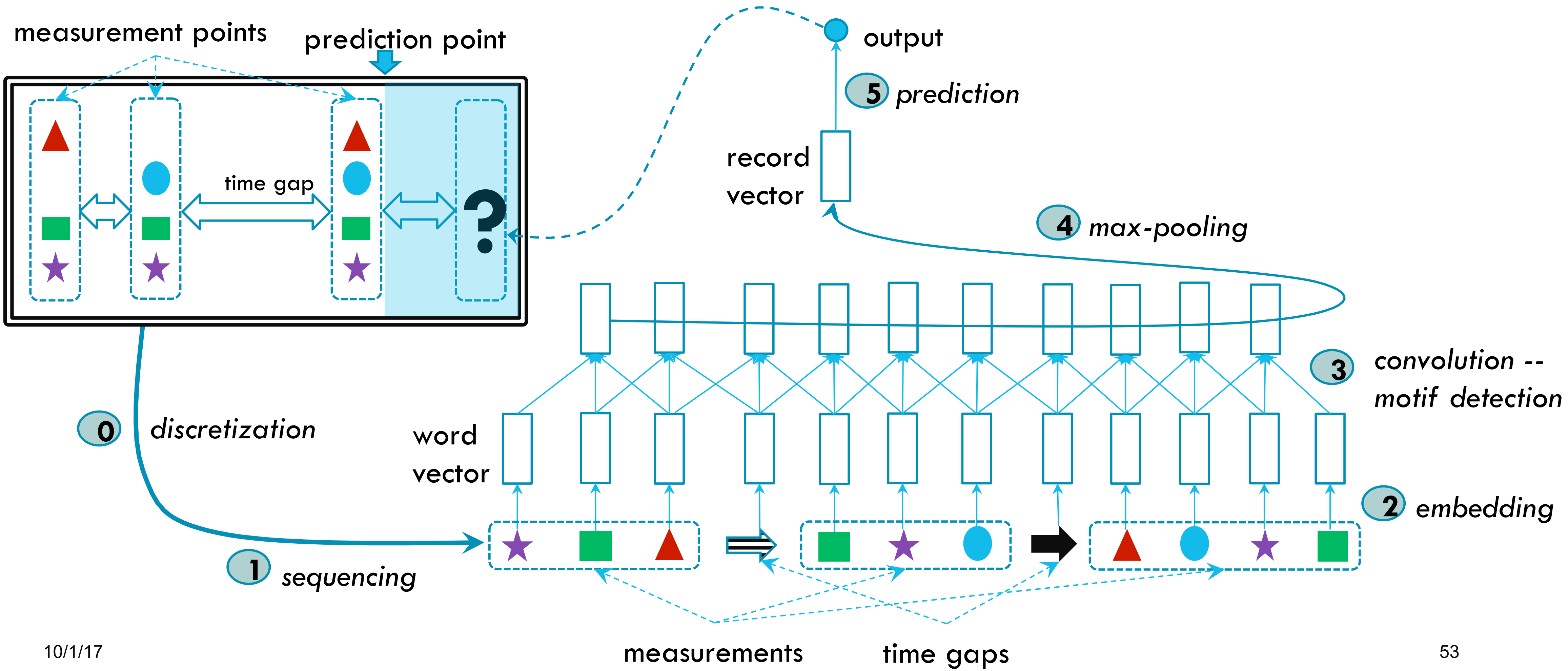


http://www.healthpages.org/brain-injury/brain-injury-intensive-care-unit-icu/

Time,Parameter,Value
00:00,RecordID,132539
00:00,Age,54
00:00,Gender,0
00:00,Height,-1
00:00,ICUType,4
00:00,Weight,-1
00:07,GCS,15
00:07,HR,73
00:07,NIDiasABP,65
00:07,NIMAP,92.33
00:07,NISysABP,147
00:07,RespRate,19
00:07,Temp,35.1
00:07,Urine,900
00:37,HR,77
00:37,NIDiasABP,58
00:37,NIMAP,91
00:37,NISysABP,157
00:37,RespRate,19
00:37,Temp,35.6
00:37,Urine,60

Data: **Physionet 2012**

DEEPIC: **SYMBOLIC** & TIME GAP REPRESENTATION OF DATA



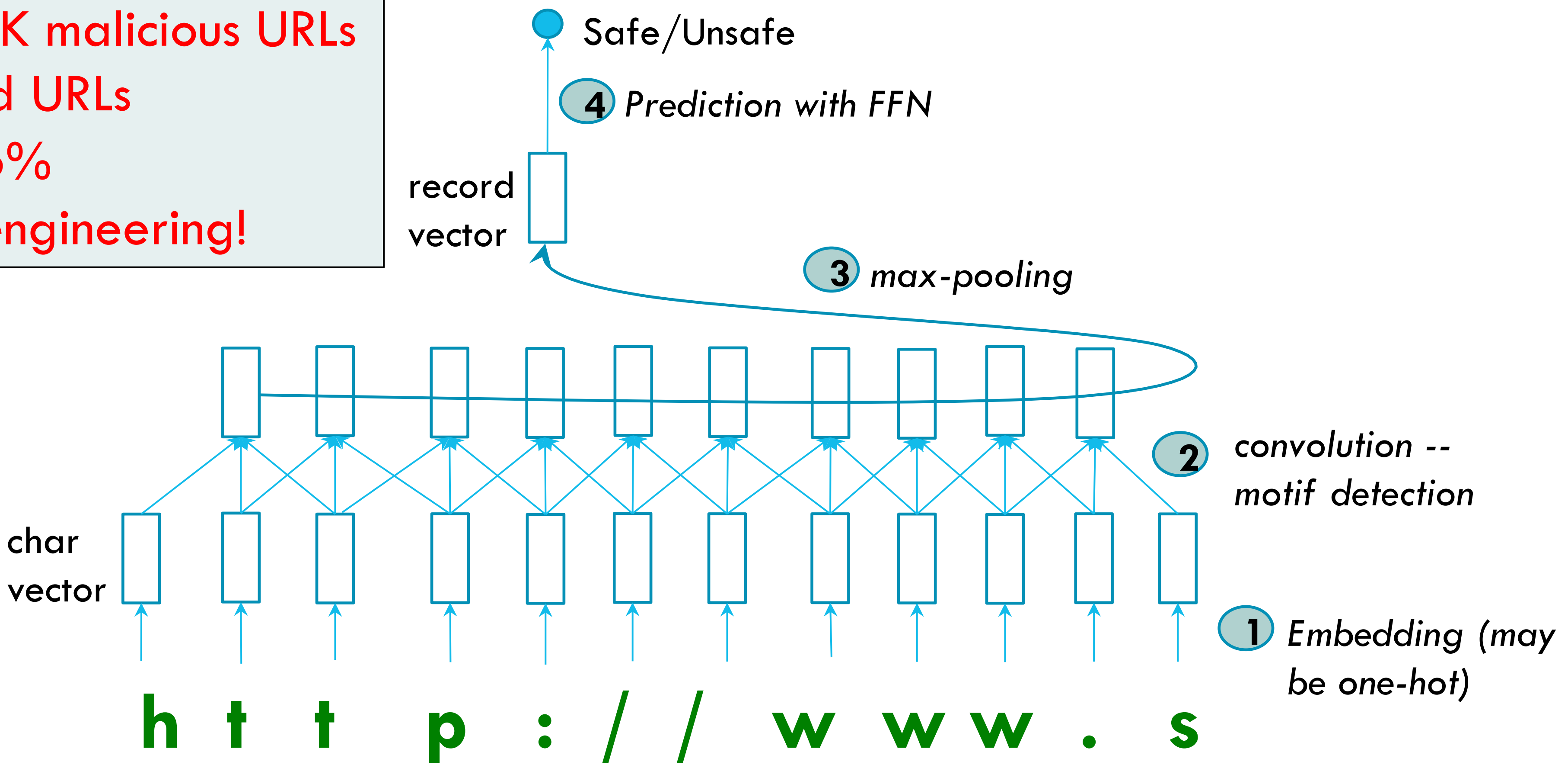
LAVASOFT

LAVASOFT

MALICIOUS URL CLASSIFICATION

MODEL OF MALICIOUS URLS

Train on 900K malicious URLs
1,000K good URLs
Accuracy: 96%
No feature engineering!



PART II: ARCHITECTURE ENGINEERING

Flexible gates (p-norm)

Sequences (Deep-long highway)

Episodes + intervention (DeepCare)

Predictive motifs (Deepr)

Matrices (DeepMat)

Graphs & relations (Column Nets)

Permutation (Neural Choice)

Multi-X (Column Bundle)

DEEPMAT: MATRICES GENERALIZE VECTORS (KIEN DO ET AL. 2017)

ECG/EEG: row (channel), column (time steps)

Healthcare: row (measures), column (time interval)

Face of multiple views

Image with multiple distortions

Image of multiple over-lapping batches/parts

Documents of multiple parts (e.g., title, abstract, etc).

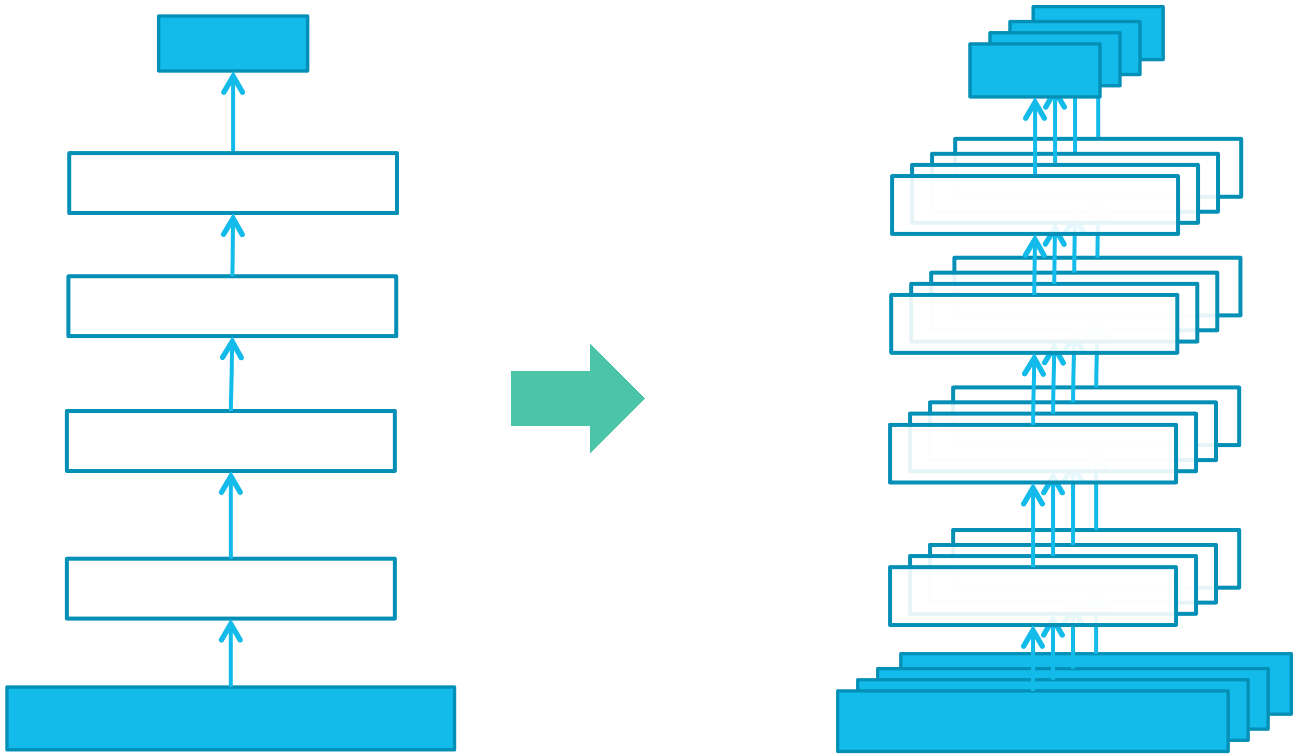
Multiple outcomes: time-horizons

Video as a sequence of 2D images

Video as a sequence of 3D short clips

Correlation/interaction matrix over time: neoronal net, email, friendship

VEC2VEC → MAT2MAT



$$Y = W H_T \Lambda$$

$$H_t = \sigma\left(A_t H_{t-1} C_t + B_t\right)$$

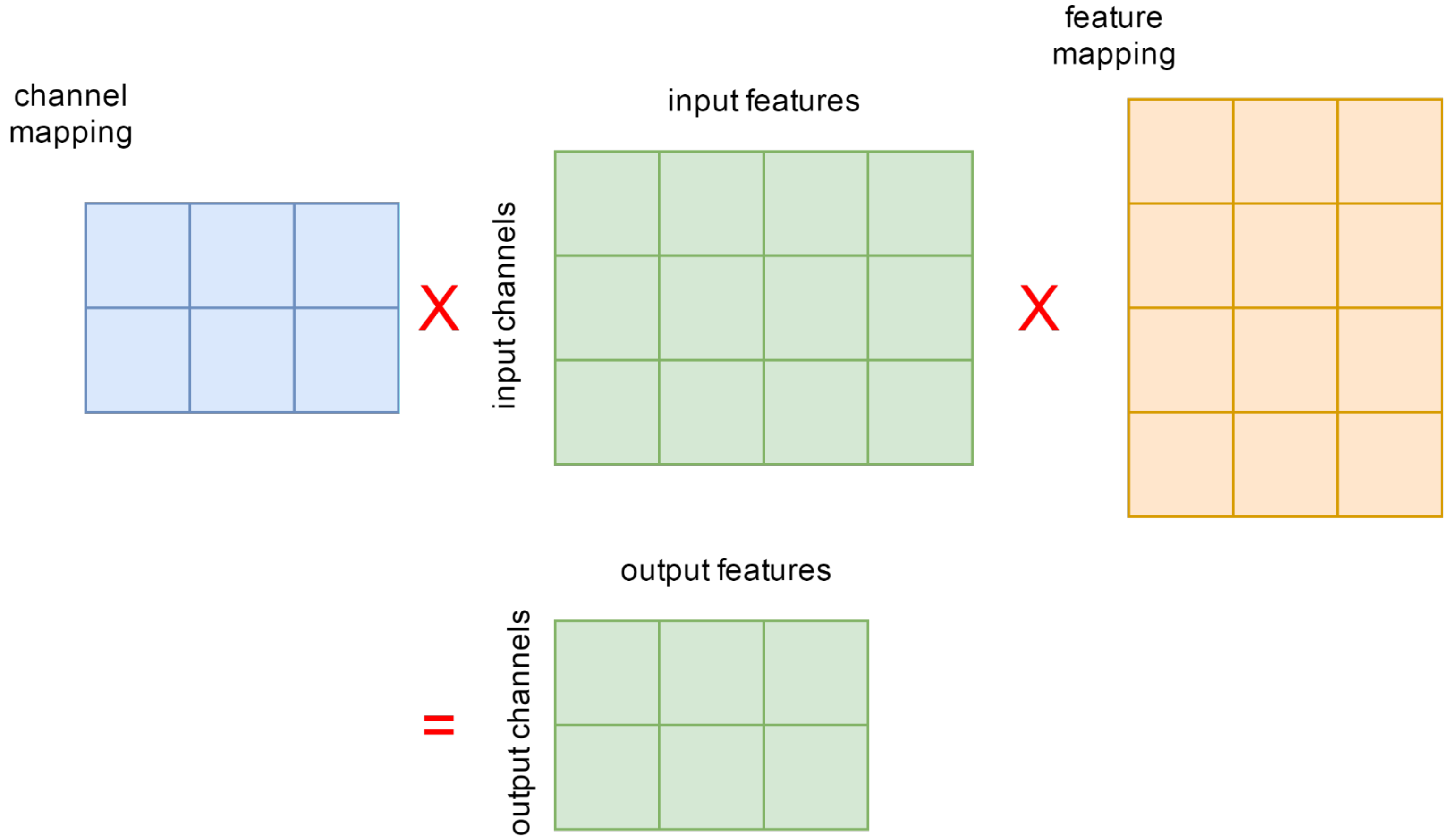
$$H_1 = \sigma\left(V X U + B_1\right)$$

MATRIX RNN

The forward pass at a layer:

$$H_t = \sigma(U_x^\intercal X_t V_x + U_h^\intercal H_{t-1} V_h + B)$$

MATRIX-MATRIX MAPPING



MATRIX-NN VS VECTOR-NN

matrix-NN	vector-NN	Error	# Parameters
H1: (20, 20) H2: (10, 10)	H1: 400 H2: 100	matrix-NN: 2.45% vector-NN: 1.46%	matrix-NN: 3,030 vector-NN: 355,110
H1: (50, 50) H2: (20, 20)	H1: 2500 H2: 400	matrix-NN: 1.73% vector-NN: **1.40%**	matrix-NN: 11,710 vector-NN: **2,966,910**
H1: (100, 100) H2: (50, 50)	H1: 10000 H2: 2500	matrix-NN: **1.38%** vector-NN: >1.40%	matrix-NN: **53,110** vector-NN: 32,877,510

Table 1: Comparison between Matrix Nets and Vector Nets over MNIST

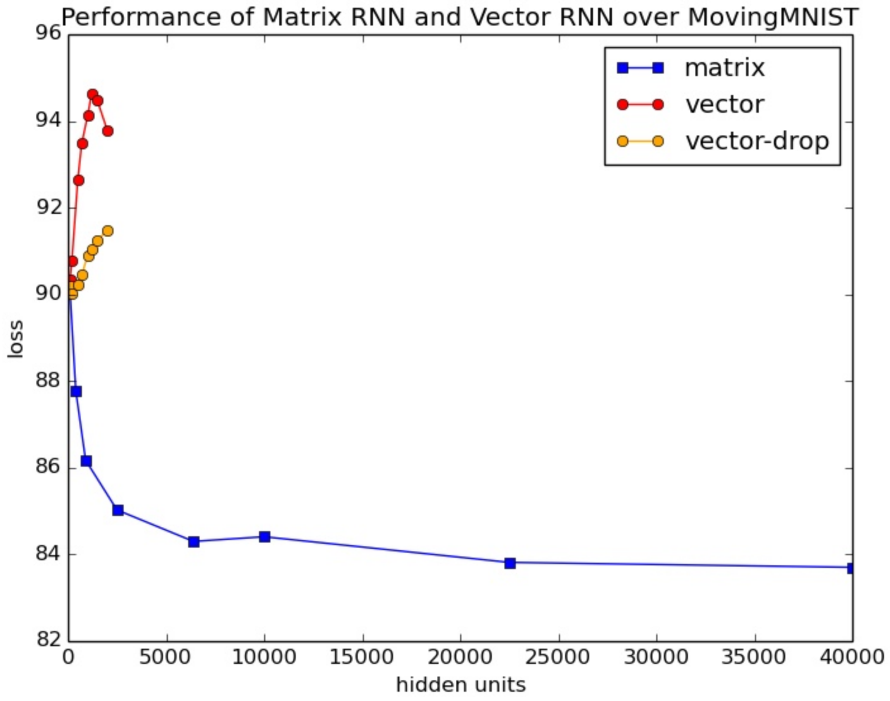
matrix-NN	vector-NN	Error	Parameters
H1: (20, 20) H2: (10, 10)	H1: 400 H2: 100	matrix-NN: 4.26% vector-NN: **1.86%**	matrix-NN: 6,538 vector-NN: **850,738**
H1: (50, 50) H2: (20, 20)	H1: 2500 H2: 400	matrix-NN: 2.15% vector-NN: 2.41%	matrix-NN: 24,638 vector-NN: 2,966,910
H1: (100, 100) H2: (50, 50)	H1: 10000 H2: 2500	matrix-NN: **1.76%** vector-NN: >2.41%	matrix-NN: **126,538** vector-NN: 45,267,538

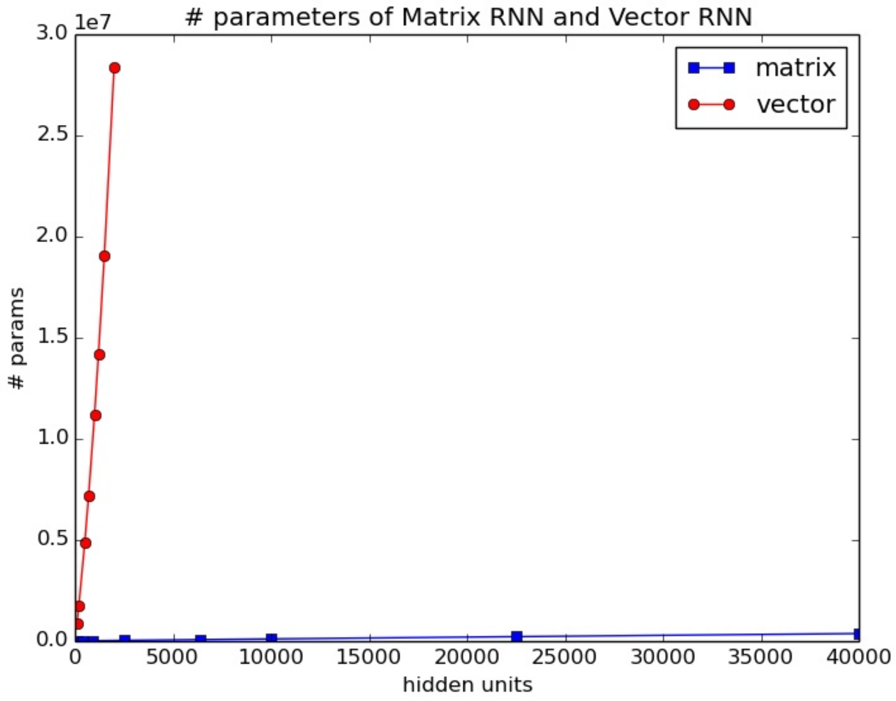
Table 2: Comparison between Matrix Nets and Vector Nets over Extended Yale Face B

MATRIX RNN VS VECTOR RNN

vector-RNN # hidden units: [100, 200, 500, 700, 1000, 1200, 1500, 2000]

matrix-RNN # hidden units: [(10, 10), (20, 20), (30, 30), (50, 50), (80, 80), (100, 100), (150, 150), (200, 200)]





PART II: ARCHITECTURE ENGINEERING

Flexible gates (p-norm)

Sequences (Deep-long highway)

Episodes + intervention (DeepCare)

Predictive motifs (Deepr)

Matrices (DeepMat)

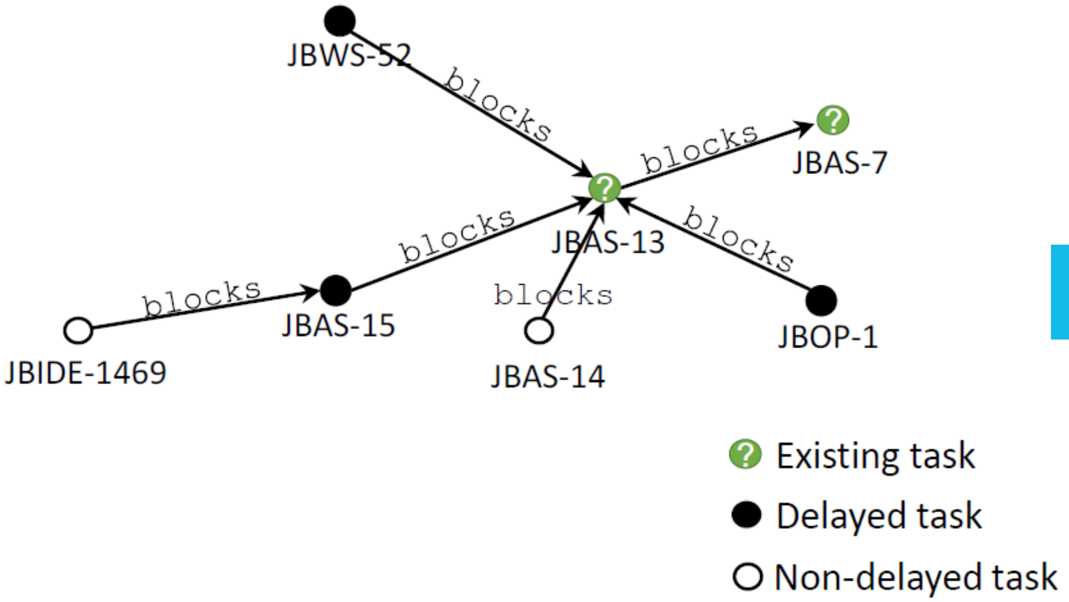
Graphs & relations (Column Nets)

Permutation (Neural Choice)

Multi-X (Column Bundle)

EXPLICIT RELATIONS

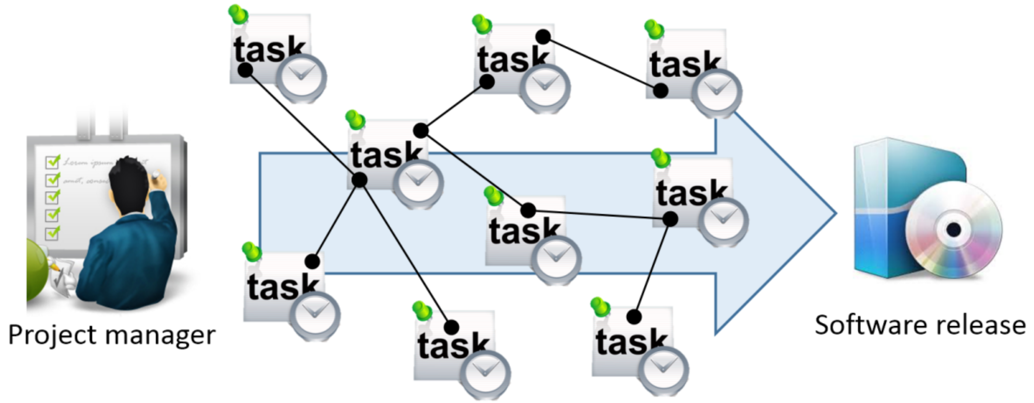
Canonical problem: **collective classification**, a.k.a. structured outputs, networked classifiers

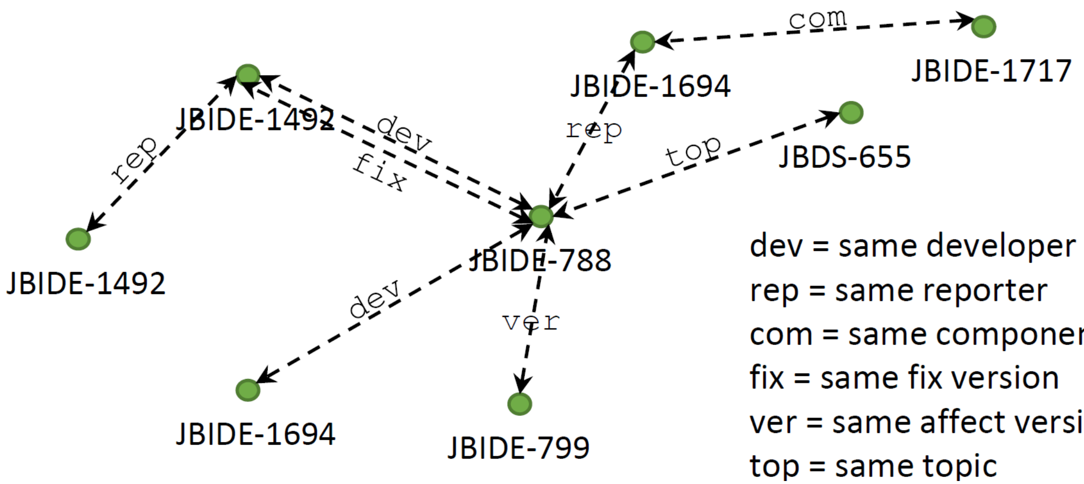


Each node has its own attributes

- Stacked inference
- (Neural) conditional random fields
- Column networks

TASK DEPENDENCY IN SOFTWARE PROJECT
(CHOETKIERTIKUL ET AL, WORK IN PROGRESS)



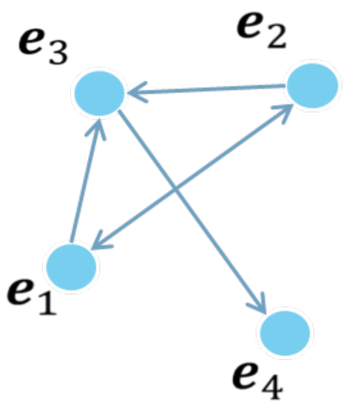


dev = same developer
rep = same reporter
com = same component
fix = same fix version
ver = same affect version
top = same topic

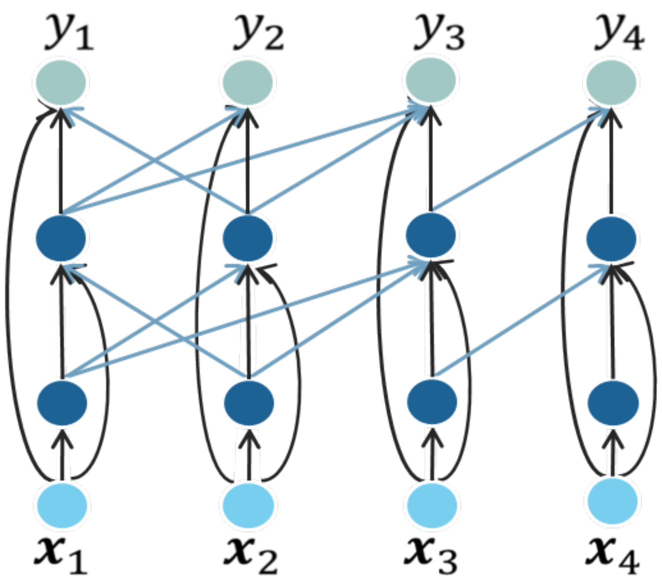
Approximately, one-third of IT projects went over the scheduled time

82% software projects missed schedules

STACKED INFERENCE



Relation graph

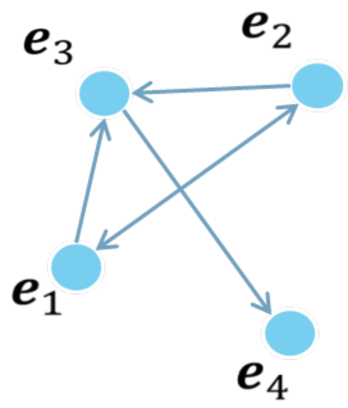


Stacked inference

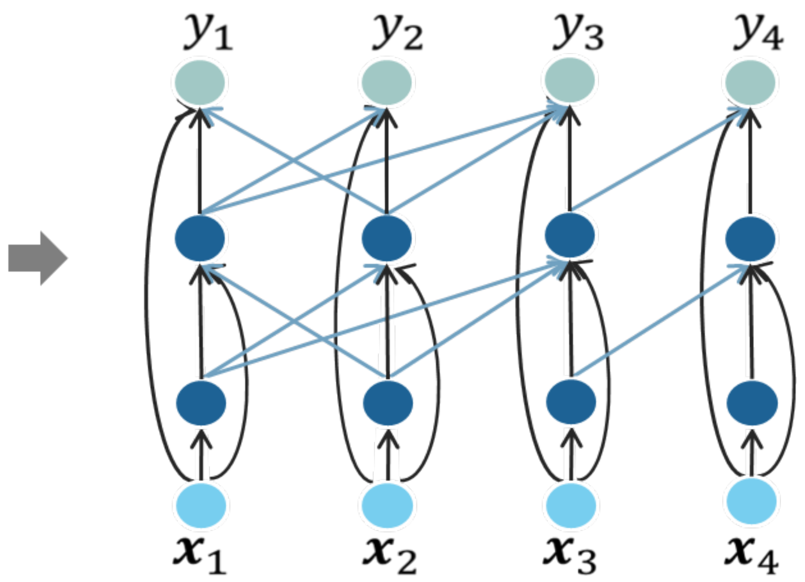
Depth is achieved by stacking several classifiers.

Lower classifiers are frozen.

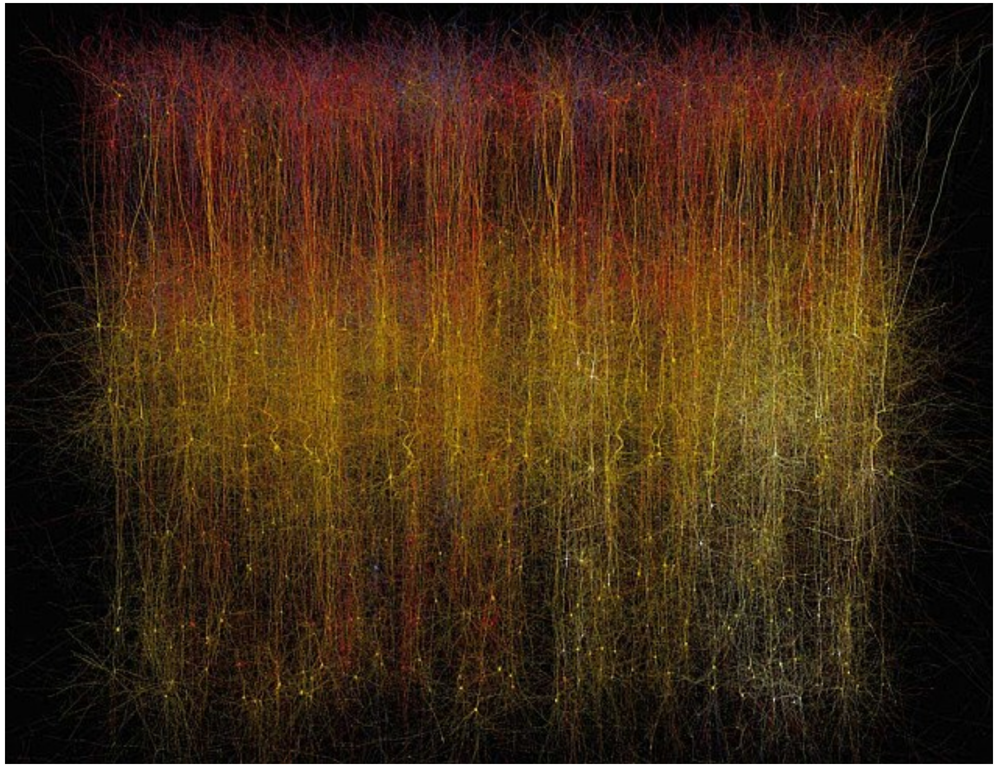
COLUMN NETWORKS: INSPIRATION



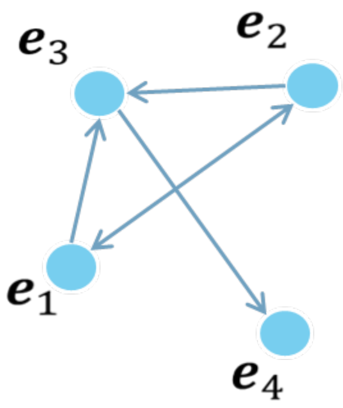
Relation graph



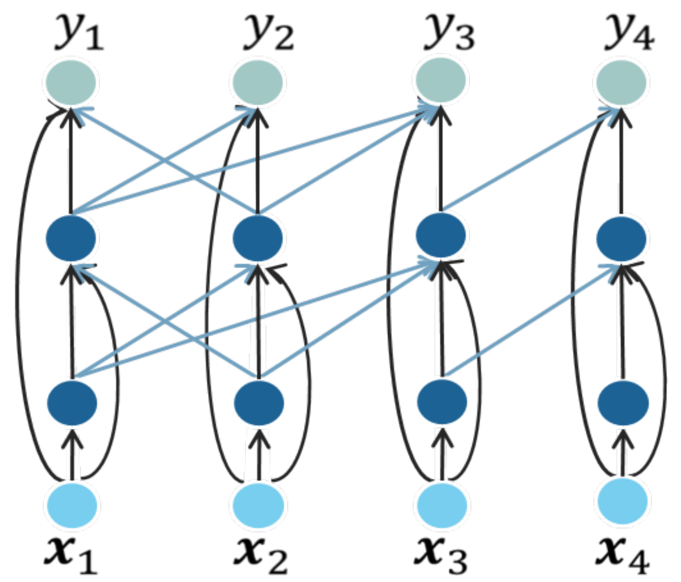
Stacked inference



COLUMN NETWORKS: DESIGN
(TRANG PHAM ET AL, @ AAAI'16)

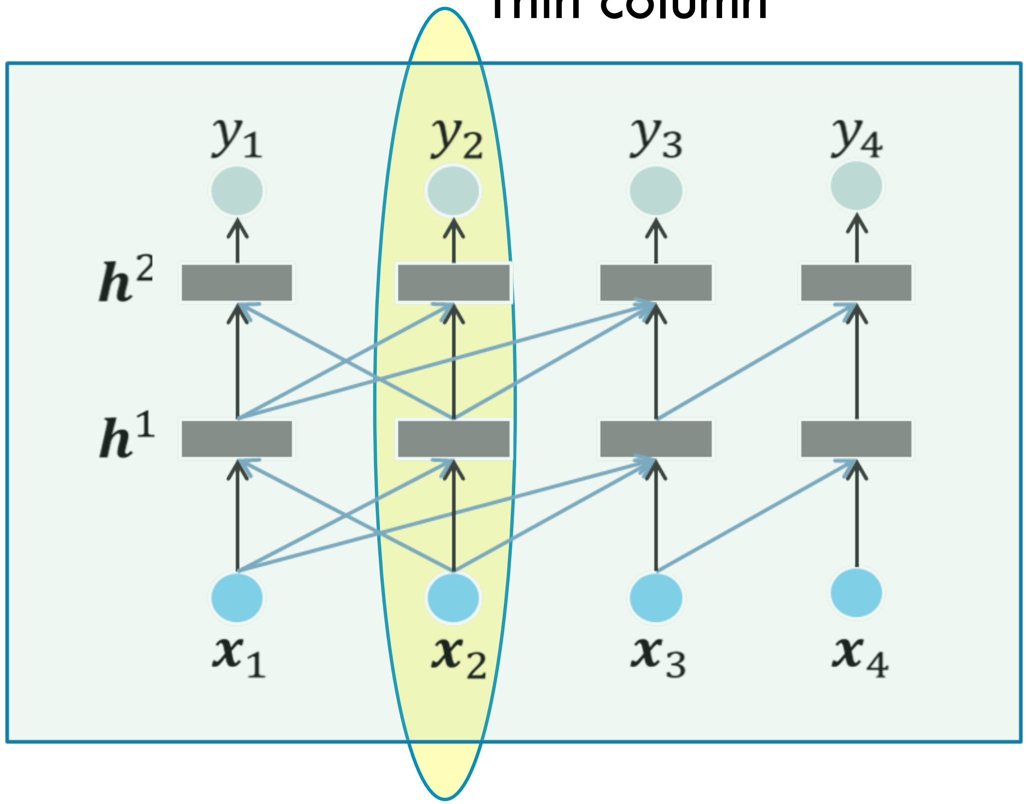


Relation graph



Stacked learning

Thin column



Column nets

PART II: ARCHITECTURE ENGINEERING

Flexible gates (p-norm)

Sequences (Deep-long highway)

Episodes + intervention (DeepCare)

Predictive motifs (Deepr)

Matrices (DeepMat)

Graphs & relations (Column Nets)

Permutation (Neural Choice)

Multi-X (Column Bundle)

KEY PROBLEM: RANKING

- Raking web documents in search engines
- Movie recommendation
- Advertisement placement
- Tag recommendation
- Expert finding in a community network
- Friend ranking in a social network
- ???

LEARNING-TO-RANK

- Web documents in search engines
  - query: *keywords*
- Movie recommendation
  - query: *an user*
- Advertisement placement
  - query: *a Web page*
- Tag recommendation
  - query: *a web object*
- Friend ranking in a social network
  - query: *an user*

Learn to rank responses to a query

A ML approach to Information Retrieval

- Instead of hand-engineering similarity measures, learn it

Two key elements

- Choice model → rank loss (how right/wrong is a ranked list?)
- Scoring function → mapping features into score (how good is the choice?)

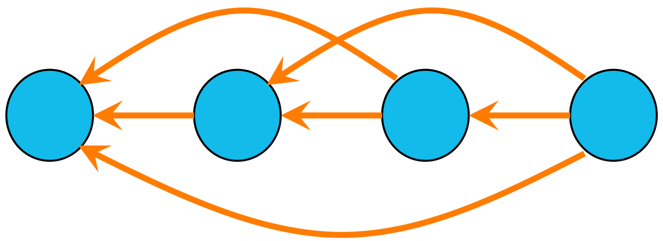
CHOICE-BY-ELIMINATION

Forward selection does not fit competitive situations

- Sport tournament, grant selection

Choice-by-elimination:

- Given a set of items with associated utility
- For each step, identify the worst item and remove it
- Repeat until one item is left
- Rank the items by the reverse order of removal



$$P(\boldsymbol{\pi}) = Q(\pi_N) \prod_{i=1}^{N-1} Q(\pi_i \mid \boldsymbol{\pi}_{i+1:N})$$

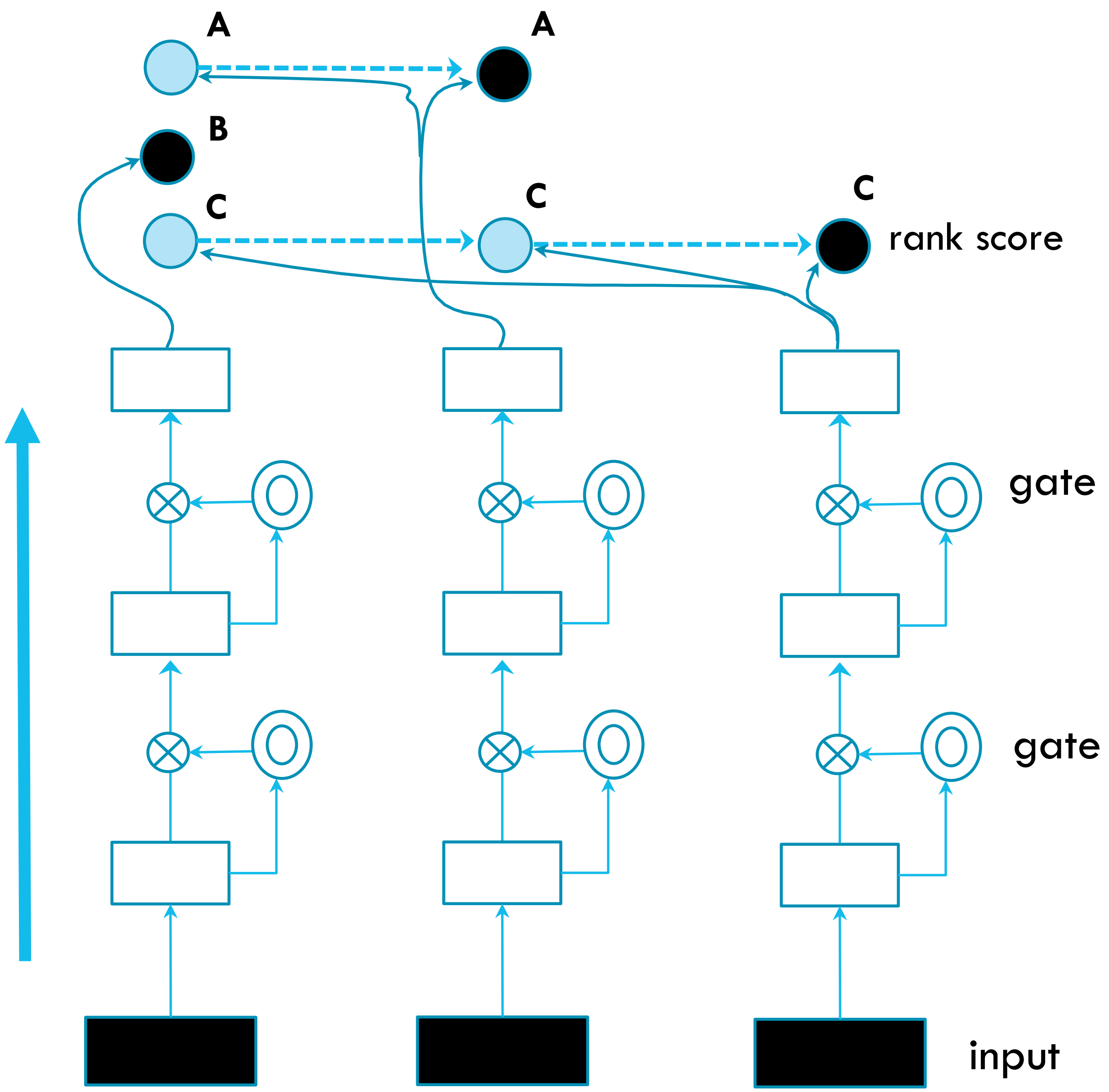
$$Q(\pi_i | \boldsymbol{\pi}_{i+1:N}) = \frac{\exp\left(-f(x_{\pi_i})\right)}{\sum_{j=1}^{i} \exp\left(-f(x_{\pi_j})\right)}$$

HIGHWAY NETS FOR RANKING

The networks represent the scoring function

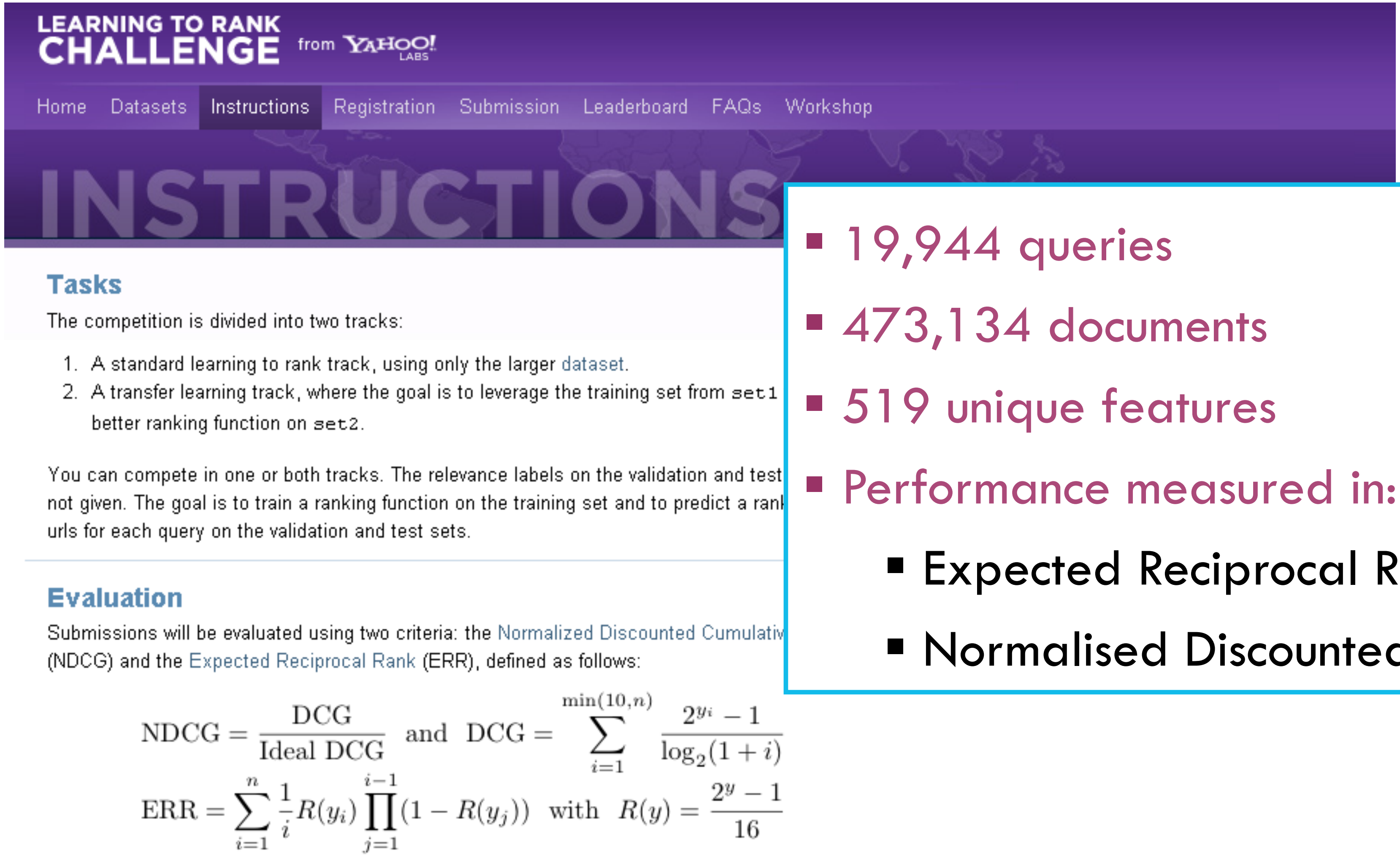
All networks are linked through the rank loss – neural choice by elimination

It is a structured output problem (permutation)



Parameter-tying highway networks

YAHOO! L2R CHALLENGE (2010)


LEARNING TO RANK CHALLENGE from YAHOO! LABS

Home Datasets Instructions Registration Submission Leaderboard FAQs Workshop

INSTRUCTIONS

Tasks

The competition is divided into two tracks:

1. A standard learning to rank track, using only the larger dataset.
2. A transfer learning track, where the goal is to leverage the training set from `set1` better ranking function on `set2`.

You can compete in one or both tracks. The relevance labels on the validation and test not given. The goal is to train a ranking function on the training set and to predict a rank urls for each query on the validation and test sets.

Evaluation

Submissions will be evaluated using two criteria: the Normalized Discounted Cumulativ (NDCG) and the Expected Reciprocal Rank (ERR), defined as follows:

$$\mathrm{NDCG} = \frac{\mathrm{DCG}}{\mathrm{Ideal\ DCG}} \quad \text{and} \quad \mathrm{DCG} = \sum_{i=1}^{\min(10,n)} \frac{2^{y_i}-1}{\log_2(1+i)}$$

$$\mathrm{ERR} = \sum_{i=1}^{n} \frac{1}{i} R(y_i) \prod_{j=1}^{i-1} (1 - R(y_j)) \quad \text{with} \quad R(y) = \frac{2^y - 1}{16}$$

- 19,944 queries
- 473,134 documents
- 519 unique features
- Performance measured in:
  - Expected Reciprocal Rank (ERR)
  - Normalised Discounted Cumulative Gain (NDCG)

RESULTS

As of 2011 – Forward selection + quadratic rank function

	ERR	NDCG@1	NDCG@5
Rank Regress	0.4882	0.683	0.6672
RankNet	0.4919	0.6903	0.6698
Ranking SVM	0.4868	0.6797	0.6662
ListMLE	0.4955	0.6993	0.6705
PairTies-D	0.4941	0.6944	0.6725
PairTies-RK	0.4946	0.6970	0.6716
PMOP-FD	0.5038	0.7137	0.6762
PMOP-Gibbs	0.5037	0.7105	**0.6792**
PMOP-MH	**0.5045**	**0.7139**	0.6790

Rank 41 out of 1500

As of 2016 – Backward elimination + deep nets

	Placket-Luce			Choice by elimination		
Rank function	ERR	NDCG@1	NDCG@5	ERR	NDCG@1	NDCG@5
SGTB	0.497	0.697	0.673	0.506	0.705	0.681
Neural nets	**0.501**	**0.705**	**0.688**	0.509	0.719	0.697

Rank?

PART II: ARCHITECTURE ENGINEERING

Flexible gates (p-norm)

Sequences (Deep-long highway)

Episodes + intervention (DeepCare)

Predictive motifs (Deepr)

Matrices (DeepMat)

Graphs & relations (Column Nets)

Permutation (Neural Choice)

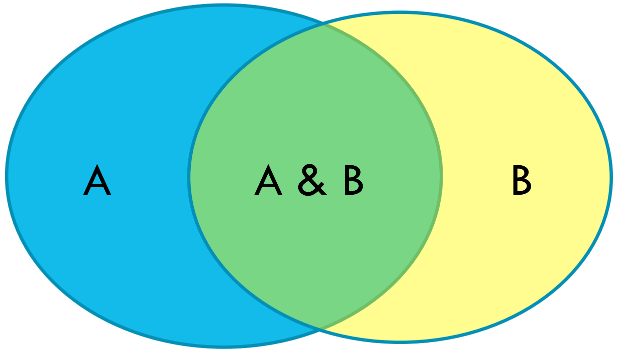
Multi-X (Column Bundle)

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

X can be:

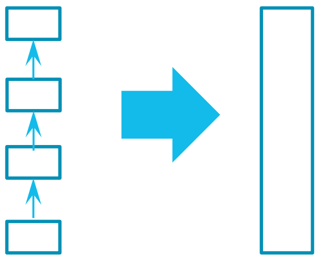
- Labels
- Tasks
- Views/parts
- Instances
- Sources

Much of recent machine learning!

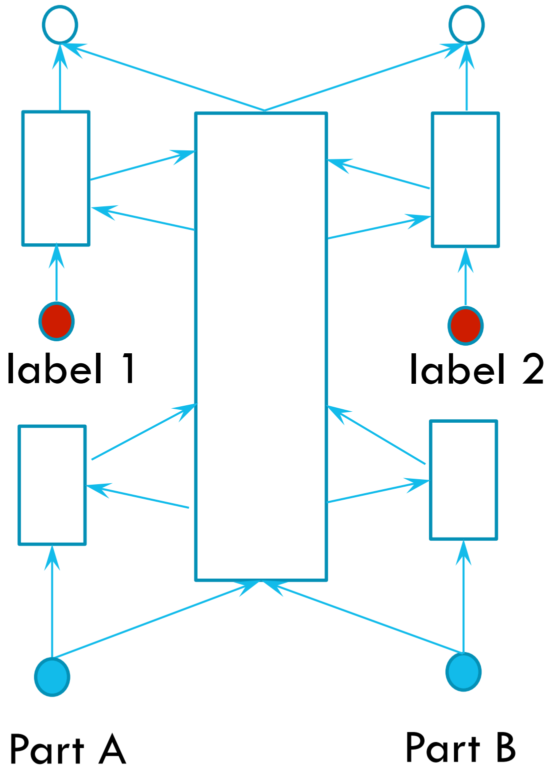


The common principle is to exploit the shared statistical strength

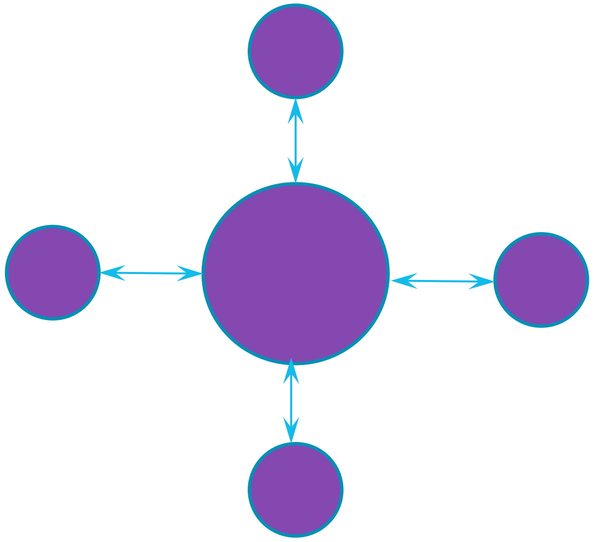
COLUMN BUNDLE FOR N-TO-M MAPPING
(PHAM ET AL, WORK IN PROGRESS)



Column



N-to-m setting



Cross-sectional star topology

RESULT: MULTILABEL LEARNING

Method	Movielens		tmc2007		MediaMill	
	MicroF1	H_Loss	MicroF1	H_Loss	MicroF1	H_Loss
PCC	**55.6**	0.229	73.2	0.058	***56.0***	0.035
BPNN	53.8	0.196	66.9	0.067	55.4	0.039
LLSF	51.8	0.208	64.9	0.064	54.0	**0.031**
HWN	53.0	**0.190**	***76.0***	***0.053***	22.4	0.035
CLB	***54.3***	***0.191***	**76.5**	**0.049**	**56.7**	***0.032***

Table 1

H_Loss: Hamming Loss

RESULT: MULTIVIEW LEARNING

Method	Youtube		NUS-WIDE	
	MicroF1	H_Loss	MicroF1	H_Loss
HW	97.3	0.027	53.1	0.022
2views-MRBM-HW	95.2	0.048	50.0	0.023
2views-CLB	97.9	0.021	56.9	**0.019**
CLB	**98.0**	**0.020**	**57.7**	**0.019**

Table 2

RESULT: MULTI-INSTANCE LEARNING

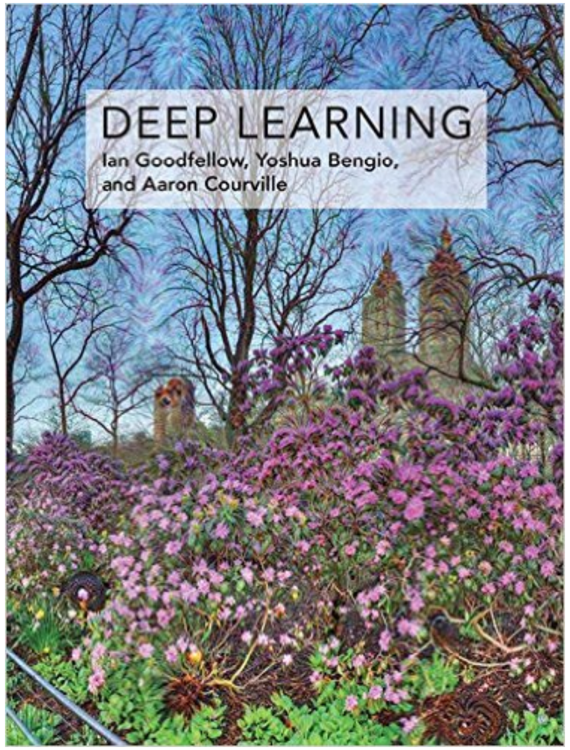
Method	IMDB	
	MicroF1	H_Loss
HW	83.9	0.163
CLB	**85.4**	**0.150**

RESOURCES

mxnet

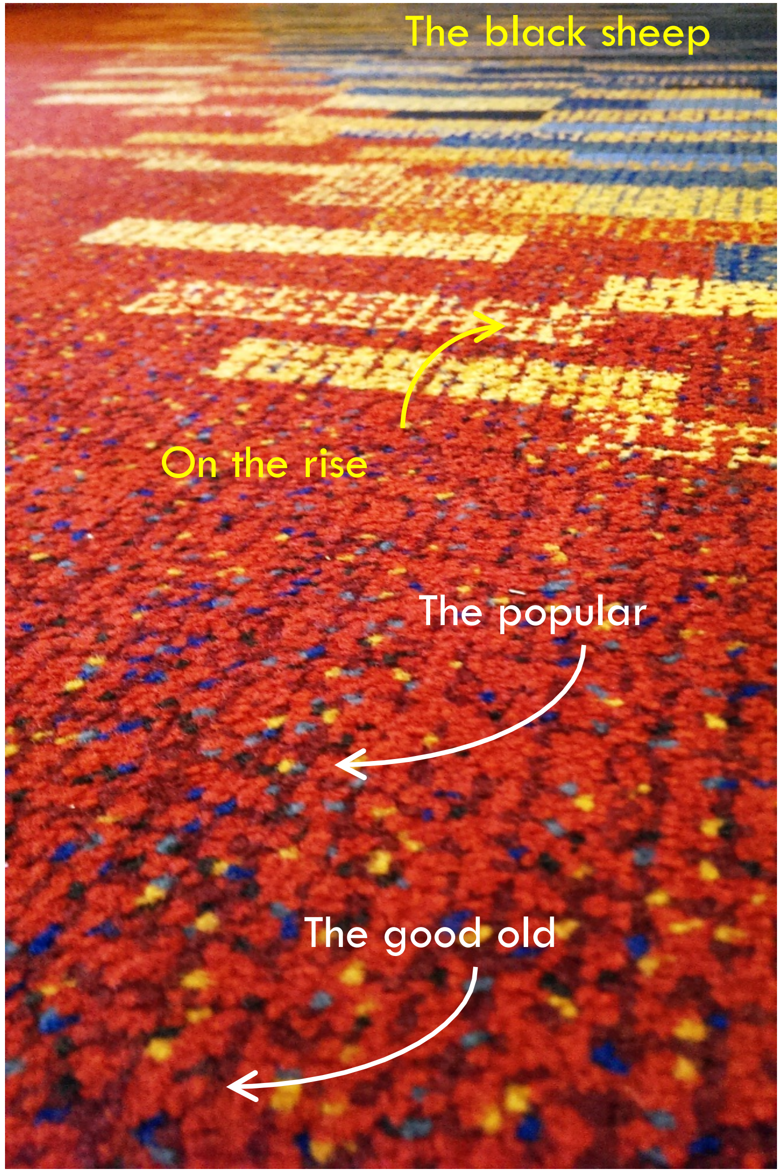
K + TensorFlow

theano



Thank you!

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



- Group theory (Lie algebra, renormalisation group, spin-class)

- Differential Turing machines
- Memory, attention & reasoning
- Reinforcement learning & planning
- Lifelong learning

- Dropouts & batch-norm
- Rectifier linear transforms & skip-connections
- Highway nets, LSTM & CNN

- Representation learning (RBM, DBN, DBM, DDAE)
- Ensemble
- Back-propagation
- Adaptive stochastic gradient

TWO MAJOR VIEWS OF "DEPTH" IN DEEP LEARNING

- [2006-2012] **Learning layered representations, from raw data to abstracted goal** (DBN, DBM, SDAE, GSN).
  - Typically 2-3 layers.
  - High hope for unsupervised learning. A conference set up for this: **ICLR**, starting in 2013.
  - **We will return in Part III.**
- [1991-1997] & [2012-2016] **Learning using multiple steps, from data to goal** (LSTM/GRU, NTM/DNC, N2N Mem, HWN, CLN).
  - Reach hundreds if not thousands layers.
  - Learning as credit-assignment.
  - Supervised learning won.
  - Unsupervised learning took a detour (VAE, GAN, NADE/MADE).

WHEN DOES DEEP LEARNING WORK?

Lots of data (e.g., millions)

Strong, clean training signals (e.g., when human can provide correct labels – cognitive domains).

- Andrew Ng of Baidu: When humans do well within sub-second.

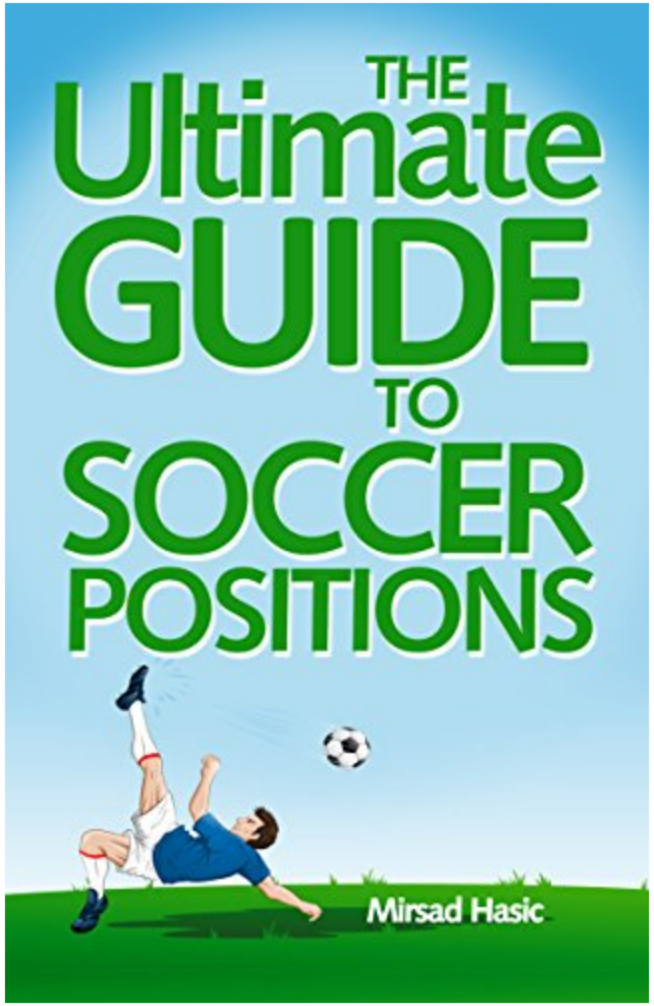
Data structures are well-defined (e.g., image, speech, NLP, video)

Data is compositional (luckily, most data are like this)

The more primitive (raw) the data, the more benefit of using deep learning.

BONUS: HOW TO POSITION

"[...] 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

Unsupervised learning

Graphs

Learning while preserving privacy

Modelling of domain invariance

Better data efficiency

Multimodality

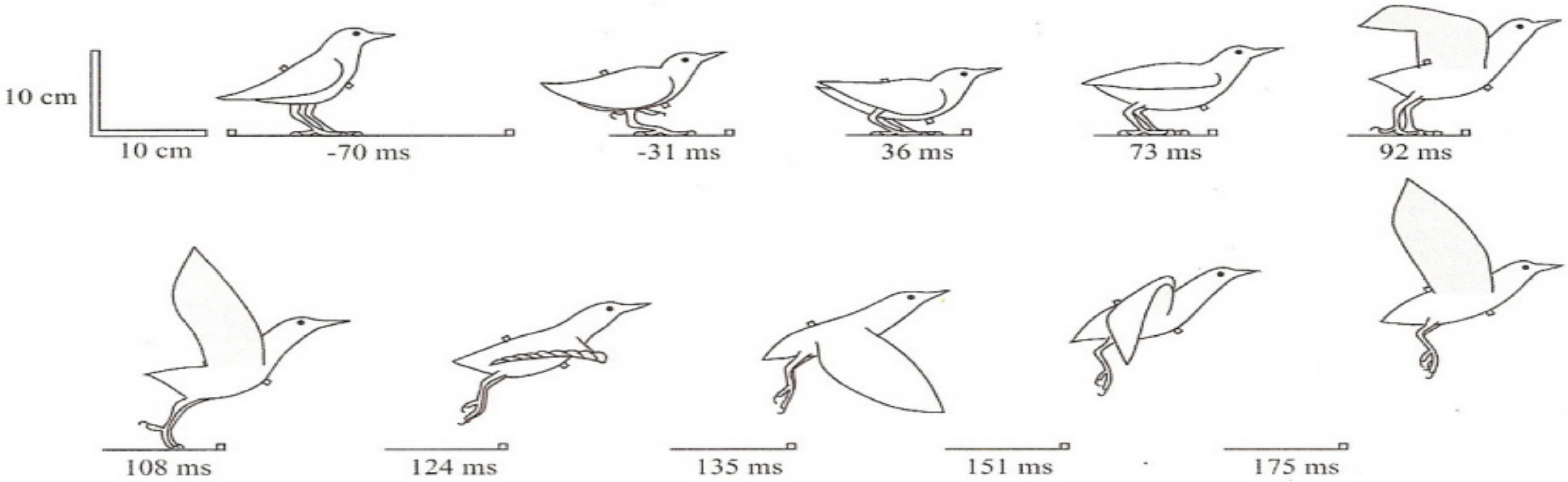
Learning under adversarial stress

Better optimization

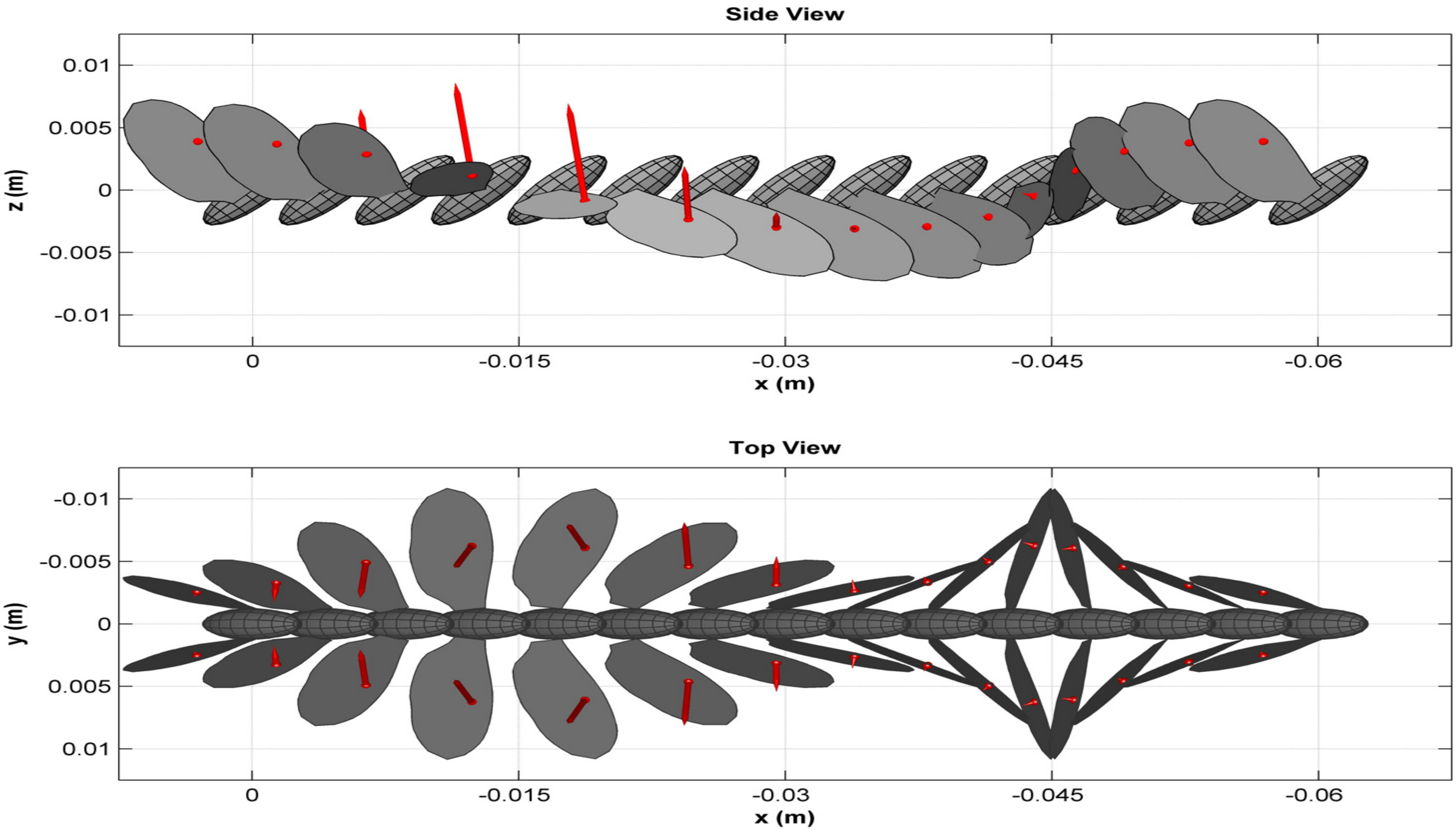
Going Bayesian

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

Early approach to heavier-than-air flight

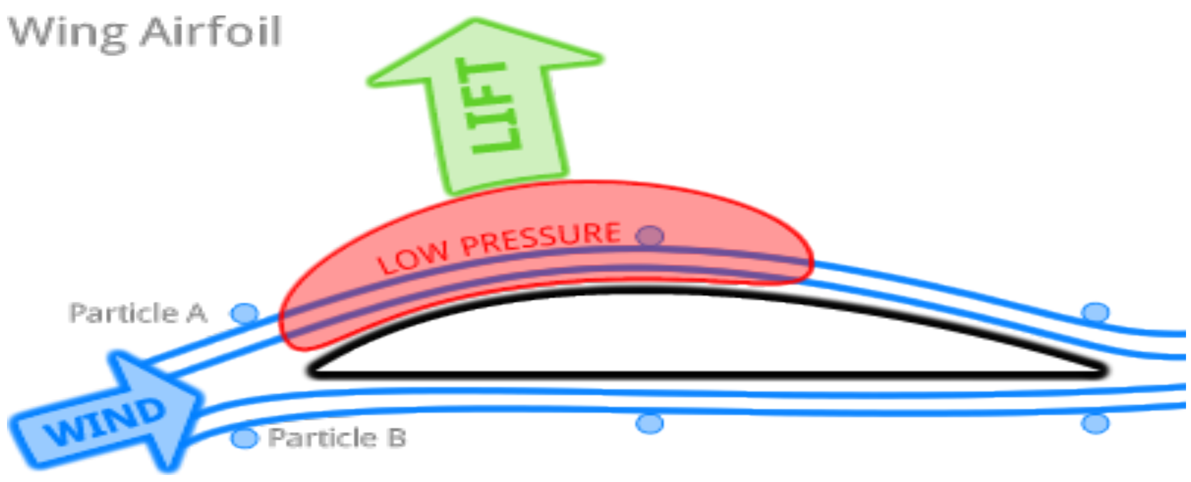


http://people.eku.edu/ritchisong/554notes2.html



Side View
z (m)
0.01
0.005
0
-0.005
-0.01
0
-0.015
-0.03
-0.045
-0.06
x (m)
Top View
y (m)
-0.01
-0.005
0
0.005
0.01
0
-0.015
-0.03
-0.045
-0.06
x (m)

A FASTER WAY



Enabling factors

- ✓ Aerodynamics
- ✓ Powerful engines
- ✓ Light materials
- ✓ Advances in control
- ✓ Established safety practices

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