



# NIPS 2016 Highlights

Norman Hendrich

University of Hamburg, Informatics Department  
Vogt-Koelln-Str. 30, D-22527 Hamburg, Germany

17 Januar 2017



## NIPS 2016

- Review Process Analysed

## Plenary Speech and Tutorials

- Predictive Learning

- Nuts and Bolts of Deep Learning

- Deep RL Tutorial

- Generative Adversarial Networks

## Workshops

- Efficient Methods for DNNs

- Continual Learning Workshop

- Neurobotics Workshop



# Neural Information Processing Systems 2016

# NIPS 2016

BARCELONA · SPAIN · DECEMBER 5-10 | <http://nips.cc>

TUTORIALS	INVITED SPEAKERS	SYMPOSIUM	ORGANIZING COMMITTEE																																																													
<b>Deep Reinforcement Learning Through Policy Optimization</b> Peter Abbeel (OpenAI, UC Berkeley) and John Schulman (OpenAI)	<b>Reproducible Research: the Case of the Human Microbiome</b> Susan Holmes (Stanford University)	<b>Recurrent Neural Networks and other Machines that Learn Algorithms</b> Alex Graves (Google DeepMind)	<b>General Chairs:</b> Daniel Lee (University of Pennsylvania) Massimo Sordani (The University of Tokyo)																																																													
<b>Large-scale Optimization: Beyond Stochastic Gradient Descent and Conjugate</b> Francis Bach (INRIA, ENS) and Suvrit Sra (MIT)	<b>Dynamic Legged Robots</b> Marc Raibert (Boston Dynamics)	<b>Jeonggun Schuster (GSIA)</b> <b>Rupesh Srivastava (IDBIA)</b> <b>Sepp Hochreiter (Johannes Kepler University)</b>	<b>Program Chairs:</b> Erikne von Luxburg (University of Tübingen) Isabelle Guyon (CopSint)																																																													
<b>Variational Inference: Foundations and Modern Methods</b> David Blei (Columbia), Shariq Mohamed (Google DeepMind) and Rajesh Ranganath (Princeton)	<b>Intelligent Biogeophers</b> Drew Purves (Google DeepMind)	<b>Deep Learning</b> Navdeep Jaitly (Google) Roger Grosse (University of Toronto)	<b>Tutorials Chair:</b> Joele Pinesau (McGill University) Hanna Wallach (Microsoft)																																																													
<b>Natural Language Processing for Computational Social Science</b> Cristian Danescu-Niculescu-Mizil (Cornell) and Lillian Lee (Cornell)	<b>Predictive Learning</b> Yann LeCun (Facebook and New York University)	<b>Machine Learning and the Law</b> Adrian Weller (Cambridge, Alan Turing Inst.) Conrad McDonnell (Gray's Inn Tax Chambers)	<b>Workshop Chairs:</b> Raia Hadsni (Google DeepMind)																																																													
<b>Generative Adversarial Networks</b> Ian Goodfellow (OpenAI)	<b>Machine Learning and Likelihood-Free Inference in Particle Physics</b> Kyle Cranmer (New York University)	<b>Jatinder Singh (University of Cambridge)</b> <b>Thomas Grant (University of Cambridge)</b>	<b>Demonstration Chair:</b> Russel Hornet (Washington University)																																																													
<b>Theory and Algorithms for Forecasting Non-stationary Time Series</b> Vitaly Kuznetsov (Google) and Mehryar Mohri (Courant Institute, Google Research)	<b>Learning About the Brain: Neuroimaging and Beyond</b> Inria RIB (IBM T.J. Watson Research Center)	<b>Engineering Principles From Stable And Developing Brains</b> Saket Navkhaha (The Salk Institute for Biological Studies)	<b>Publications Chair &amp; Electronic Proceedings Chair:</b> Romain Garnett (Washington University)																																																													
<b>Deep Learning for Building AI Systems</b> Andrew Ng (Baidu, Stanford University)	<b>ML Foundations and Methods for Precision Medicine and Healthcare</b> Saeed Sarda (Johns Hopkins) and Peter Schütten (Johns Hopkins)	<b>Program Managers:</b> Kourosh Mousavi (McGill University and MPI) Rohit Saxena, Behzad Tabibian (MPI for Intelligent Systems)																																																														
<b>Crowdsourcing: Beyond Label Generation</b> Jenn Wortman Vaughan (Microsoft Research)	<p><b>PROGRAM COMMITTEE</b></p> <table border="1"> <tbody> <tr> <td>Abhishek Bosevic (Brown)</td> <td>David Rosenberg (Amazon)</td> <td>William Scheraga (Cornell)</td> </tr> <tr> <td>Aravind Cheung (Google)</td> <td>David Rosenberg (Amazon)</td> <td>William Scheraga (Cornell)</td> </tr> <tr> <td>Aravind Cheung (Google)</td> <td>David Rosenberg (Amazon)</td> <td>William Scheraga (Cornell)</td> </tr> <tr> <td>Aravind Cheung (Google)</td> <td>David Rosenberg (Amazon)</td> <td>William Scheraga (Cornell)</td> </tr> <tr> <td>Aravind Cheung (Google)</td> <td>David Rosenberg (Amazon)</td> <td>William Scheraga (Cornell)</td> </tr> </tbody> </table>	Abhishek Bosevic (Brown)		David Rosenberg (Amazon)	William Scheraga (Cornell)	Aravind Cheung (Google)	David Rosenberg (Amazon)	William Scheraga (Cornell)	Aravind Cheung (Google)	David Rosenberg (Amazon)	William Scheraga (Cornell)	Aravind Cheung (Google)	David Rosenberg (Amazon)	William Scheraga (Cornell)	Aravind Cheung (Google)	David Rosenberg (Amazon)	William Scheraga (Cornell)	<table border="1"> <tbody> <tr> <td>Tommi Jaakkola (MIT)</td> <td>Mikhail Sirovica (Google)</td> <td>Gregory Wornat (Facebook)</td> </tr> <tr> <td>Tommi Jaakkola (MIT)</td> <td>Mikhail Sirovica (Google)</td> <td>Gregory Wornat (Facebook)</td> </tr> <tr> <td>Tommi Jaakkola (MIT)</td> <td>Mikhail Sirovica (Google)</td> <td>Gregory Wornat (Facebook)</td> </tr> <tr> <td>Tommi Jaakkola (MIT)</td> <td>Mikhail Sirovica (Google)</td> <td>Gregory Wornat (Facebook)</td> </tr> <tr> <td>Tommi Jaakkola (MIT)</td> <td>Mikhail Sirovica (Google)</td> <td>Gregory Wornat (Facebook)</td> </tr> </tbody> </table>	Tommi Jaakkola (MIT)	Mikhail Sirovica (Google)	Gregory Wornat (Facebook)	Tommi Jaakkola (MIT)	Mikhail Sirovica (Google)	Gregory Wornat (Facebook)	Tommi Jaakkola (MIT)	Mikhail Sirovica (Google)	Gregory Wornat (Facebook)	Tommi Jaakkola (MIT)	Mikhail Sirovica (Google)	Gregory Wornat (Facebook)	Tommi Jaakkola (MIT)	Mikhail Sirovica (Google)	Gregory Wornat (Facebook)	<table border="1"> <tbody> <tr> <td>Yann Lecun (Facebook)</td> <td>Dougal Mastrorosso (Google)</td> <td>Adrian Weller (Cambridge)</td> </tr> <tr> <td>Yann Lecun (Facebook)</td> <td>Dougal Mastrorosso (Google)</td> <td>Adrian Weller (Cambridge)</td> </tr> <tr> <td>Yann Lecun (Facebook)</td> <td>Dougal Mastrorosso (Google)</td> <td>Adrian Weller (Cambridge)</td> </tr> <tr> <td>Yann Lecun (Facebook)</td> <td>Dougal Mastrorosso (Google)</td> <td>Adrian Weller (Cambridge)</td> </tr> <tr> <td>Yann Lecun (Facebook)</td> <td>Dougal Mastrorosso (Google)</td> <td>Adrian Weller (Cambridge)</td> </tr> </tbody> </table>	Yann Lecun (Facebook)	Dougal Mastrorosso (Google)	Adrian Weller (Cambridge)	Yann Lecun (Facebook)	Dougal Mastrorosso (Google)	Adrian Weller (Cambridge)	Yann Lecun (Facebook)	Dougal Mastrorosso (Google)	Adrian Weller (Cambridge)	Yann Lecun (Facebook)	Dougal Mastrorosso (Google)	Adrian Weller (Cambridge)	Yann Lecun (Facebook)	Dougal Mastrorosso (Google)	Adrian Weller (Cambridge)	<table border="1"> <tbody> <tr> <td>Michael Sordani (University of Tokyo)</td> <td>Yann Lecun (Facebook)</td> <td>David Rosenberg (Amazon)</td> </tr> <tr> <td>Michael Sordani (University of Tokyo)</td> <td>Yann Lecun (Facebook)</td> <td>David Rosenberg (Amazon)</td> </tr> <tr> <td>Michael Sordani (University of Tokyo)</td> <td>Yann Lecun (Facebook)</td> <td>David Rosenberg (Amazon)</td> </tr> <tr> <td>Michael Sordani (University of Tokyo)</td> <td>Yann Lecun (Facebook)</td> <td>David Rosenberg (Amazon)</td> </tr> <tr> <td>Michael Sordani (University of Tokyo)</td> <td>Yann Lecun (Facebook)</td> <td>David Rosenberg (Amazon)</td> </tr> </tbody> </table>	Michael Sordani (University of Tokyo)	Yann Lecun (Facebook)	David Rosenberg (Amazon)	Michael Sordani (University of Tokyo)	Yann Lecun (Facebook)	David Rosenberg (Amazon)	Michael Sordani (University of Tokyo)	Yann Lecun (Facebook)	David Rosenberg (Amazon)	Michael Sordani (University of Tokyo)	Yann Lecun (Facebook)	David Rosenberg (Amazon)	Michael Sordani (University of Tokyo)	Yann Lecun (Facebook)
Abhishek Bosevic (Brown)	David Rosenberg (Amazon)	William Scheraga (Cornell)																																																														
Aravind Cheung (Google)	David Rosenberg (Amazon)	William Scheraga (Cornell)																																																														
Aravind Cheung (Google)	David Rosenberg (Amazon)	William Scheraga (Cornell)																																																														
Aravind Cheung (Google)	David Rosenberg (Amazon)	William Scheraga (Cornell)																																																														
Aravind Cheung (Google)	David Rosenberg (Amazon)	William Scheraga (Cornell)																																																														
Tommi Jaakkola (MIT)	Mikhail Sirovica (Google)	Gregory Wornat (Facebook)																																																														
Tommi Jaakkola (MIT)	Mikhail Sirovica (Google)	Gregory Wornat (Facebook)																																																														
Tommi Jaakkola (MIT)	Mikhail Sirovica (Google)	Gregory Wornat (Facebook)																																																														
Tommi Jaakkola (MIT)	Mikhail Sirovica (Google)	Gregory Wornat (Facebook)																																																														
Tommi Jaakkola (MIT)	Mikhail Sirovica (Google)	Gregory Wornat (Facebook)																																																														
Yann Lecun (Facebook)	Dougal Mastrorosso (Google)	Adrian Weller (Cambridge)																																																														
Yann Lecun (Facebook)	Dougal Mastrorosso (Google)	Adrian Weller (Cambridge)																																																														
Yann Lecun (Facebook)	Dougal Mastrorosso (Google)	Adrian Weller (Cambridge)																																																														
Yann Lecun (Facebook)	Dougal Mastrorosso (Google)	Adrian Weller (Cambridge)																																																														
Yann Lecun (Facebook)	Dougal Mastrorosso (Google)	Adrian Weller (Cambridge)																																																														
Michael Sordani (University of Tokyo)	Yann Lecun (Facebook)	David Rosenberg (Amazon)																																																														
Michael Sordani (University of Tokyo)	Yann Lecun (Facebook)	David Rosenberg (Amazon)																																																														
Michael Sordani (University of Tokyo)	Yann Lecun (Facebook)	David Rosenberg (Amazon)																																																														
Michael Sordani (University of Tokyo)	Yann Lecun (Facebook)	David Rosenberg (Amazon)																																																														
Michael Sordani (University of Tokyo)	Yann Lecun (Facebook)	David Rosenberg (Amazon)																																																														



# Barcelona

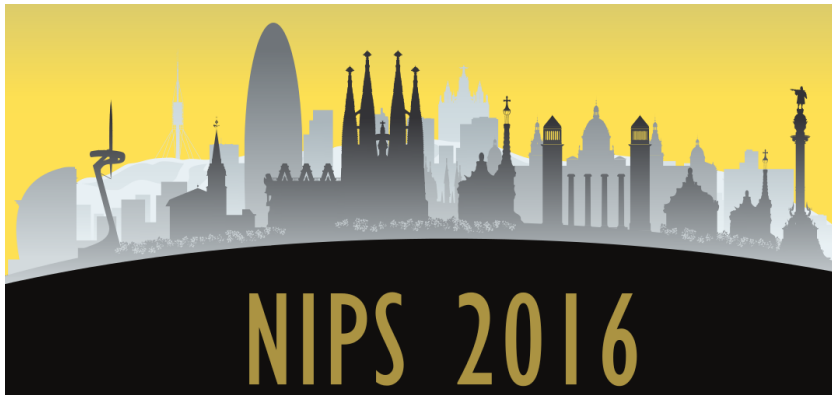
- ▶ capital city of Catalonia region
- ▶ 2nd largest city in Spain
- ▶ 1.6 M people (4.7 M urban area)
  
- ▶ gothic, art-nouveau/modernist and modern architecture icons
- ▶ hills, harbour, and beaches
- ▶ party all year round
- ▶ UNESCO world heritage
- ▶ Olympic games 1992
- ▶ FCB, Camp Nou, Lionel Messi
  
- ▶ NIPS-16 at CCIB conference center
- ▶ conveniently, 70 m from the beach





# Las Ramblas, Parc Güell, Beach



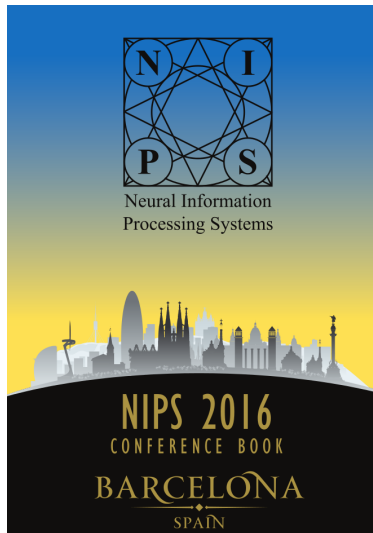




- ▶ 05-10 Dec 2016, Centre Convencions Internacional Barcelona
- ▶ organizers:
  - ▶ Daniel D. Lee, Masashi Sugiyama, General Chairs
  - ▶ Ulrike von Luxburg, Isabelle Guyon, Program Chairs
- ▶ 2500 submissions, 568 accepted (22%)
- ▶ 45 oral presentations (2%), 523 posters (20%)
- ▶ 13674 reviews
  
- ▶ two track conference, four poster sessions
- ▶ 1 plenary lecture: Yann LeCun „Predictive Learning“
- ▶ 9 tutorials, 3 symposia, 50 workshops
- ▶ ca. 40 exhibitors
- ▶ ca. 6000 participants
  
- ▶ open access: digest and all papers online



- ▶ deep learning everywhere
  - ▶ image recognition
  - ▶ MNIST, CIFAR, ...
  - ▶ recursive architectures
  - ▶ deep RL and A3C
- ▶ lots of Bayesian approaches
- ▶ few papers on „classical“ algorithms
- ▶ no SVMs at all







# Opening Ceremony: Terry's Law





# Opening Ceremony: Program Chairs





# Submissions and Reviews

## NIPS 2016 in numbers

- 2500+ submissions
- 568 accepted papers (45 orals, 523 posters)
- 100 area chairs
- 3242 active reviewers, 13,674 reviews in total

I



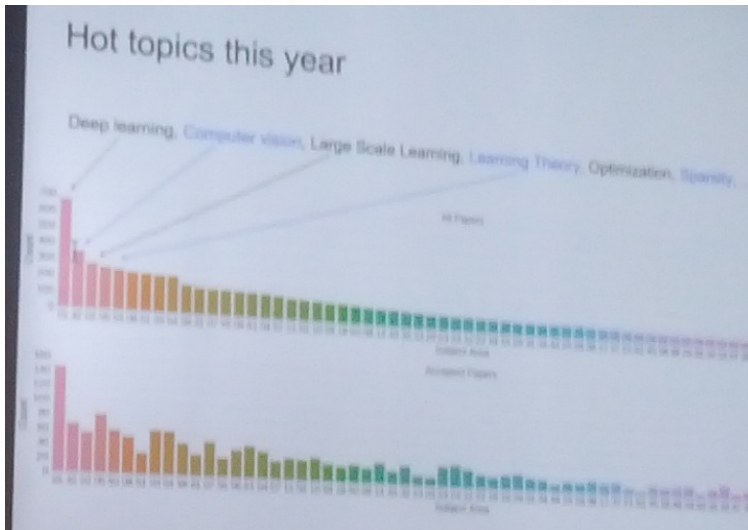
Registration numbers 2002-2016





# Hot Topics

Deep Learning, CV, Large-scale, Theory, plus a large tail





## Grow while decreasing bias and variance?

- The Ground Truth about "best papers" is unknown.
- All we can do is "vote" among many reviewers and measure variance.
- NIPS 2014 experiment: double the review process in part to measure variance.

**We aimed at doubling the number of reviewers for EVERYBODY (less variance) and increase reviewer diversity (less bias):**

- Reviewer "cloning"
- Volunteer reviewers



# Reviews: Quality of Submissions

## Quality of submissions

Distribution of Review Scores (in percent of submissions)

	1	2 (sub-standard)	3 (poster level: top 30%)	4 (oral level: top 3%)	5 (award level: top 0.1%)
Impact	6.5	36.1	45.7	10.5	1.1
Quality	6.7	38	44.7	9.5	1.1
Novelty	6.4	34.8	48.1	9.7	1.1
Clarity	7.1	28	48.6	14.6	1.8

More than expected!

Reviewers found: over 50% of the submissions of NIPS poster quality or better.



# Reviews: Automatic Area Chairs?

## The Automatic AC?

Could we automate the decision process? Idea: After discussions, ACs give weights to reviews. Then we compute a weighted global score and threshold it.

Survey: Opinions on automatic AC (by authors, area chairs, reviewers)





## NIPS 2016

Review Process Analysed

## Plenary Speech and Tutorials

Predictive Learning

Nuts and Bolts of Deep Learning

Deep RL Tutorial

Generative Adversarial Networks

## Workshops

Efficient Methods for DNNs

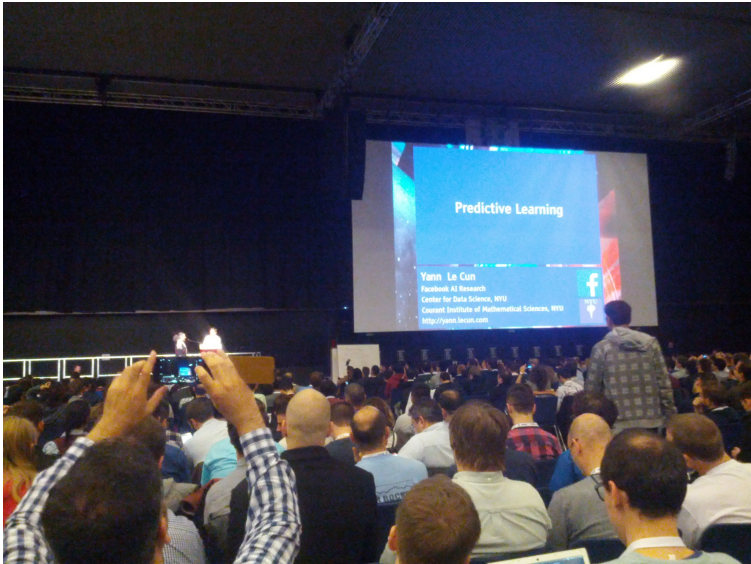
Continual Learning Workshop

Neurobotics Workshop






# Yann LeCun Plenary Talk: Predictive Learning



Predictive Learning

Yann LeCun  
Facebook AI Research  
Center for Data Science, NYU  
Courant Institute of Mathematical Sciences, NYU  
<http://yann.lecun.com>



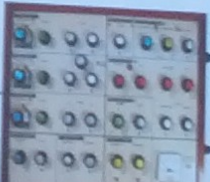


# Supervised Learning

## Supervised Learning

Y Lecun

- We can train a machine on lots of examples of tables, chairs, dog, cars, and people
- But will it recognize table, chairs, dogs, cars, and people it has never seen before?





# Very Deep Convolutional Nets


**Very Deep ConvNet Architectures** Y LeCun

Small kernels, not much subsampling (fractional subsampling).

**VGG**

**GoogLeNet**

**ResNet**





## DeepMask: ConvNet Locates and Recognizes Objects

[Pinheiro, Collobert, Dollar ICCV 2015]

ConvNet produces object masks and categories

Y LeCun

x: 3x224x224

VGG

512x14x14

1x1 conv

512x14x14

5x5 conv

5x5x5

2x2 pool

512x7x7

1074x2x1

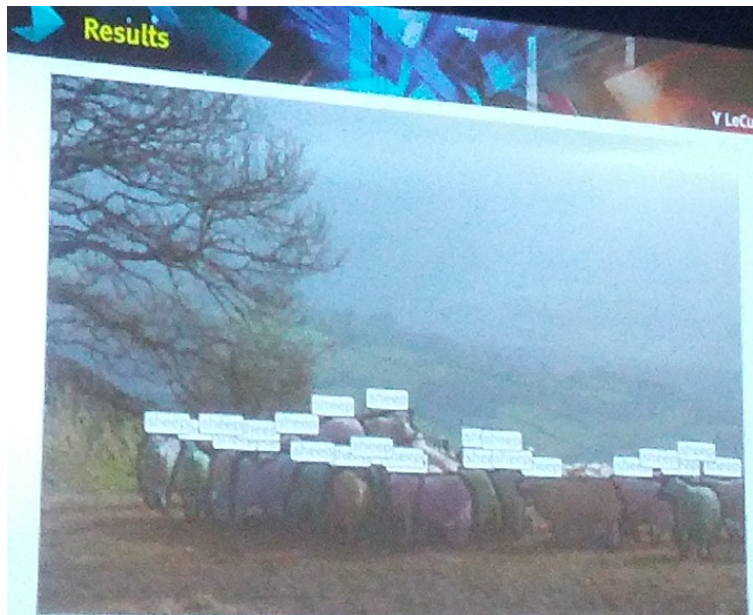
2048x1x1

$f_{\text{mask}}(x): 224x224$

$f_{\text{cat}}(x): 1x1$



# DeepMask: Sheep





# Obstacles to Progress in AI

f

## Obstacles to Progress in AI

Y LeCun

- Machines need to learn/understand how the world works
  - ▶ Physical world, digital world, people, ....
  - ▶ They need to acquire some level of common sense
- They need to learn a very large amount of background knowledge
  - ▶ Through observation and action
- Machines need to **perceive** the state of the world
  - ▶ So as to make accurate predictions and planning
- Machines need to **update** and remember estimates of the state of the world
  - ▶ Paying attention to important events. Remember relevant events
- Machines need to **reason and plan**
  - ▶ Predict which sequence of actions will lead to a desired state of the world
- Intelligence & Common Sense =

**Perception** + **Predictive Model** + **Memory** + **Reasoning & Planning**



# Learning Efficiency: the Cake Analogy

## How Much Information Does the Machine Need to Predict?

Y LeCun

### ■ "Pure" Reinforcement Learning (cherry)

- ▶ The machine predicts a scalar reward given once in a while.
- ▶ **A few bits for some samples**

### ■ Supervised Learning (icing)

- ▶ The machine predicts a category or a few numbers for each input
- ▶ Predicting human-supplied data
- ▶ **10→10,000 bits per sample**

### ■ Unsupervised/Predictive Learning (cake)

- ▶ The machine predicts any part of its input for any observed part.
- ▶ Predicts future frames in videos
- ▶ **Millions of bits per sample**



■ (Yes, I know, this picture is slightly offensive to RL folks. But I'll make it up)

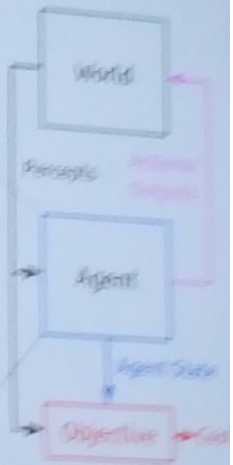
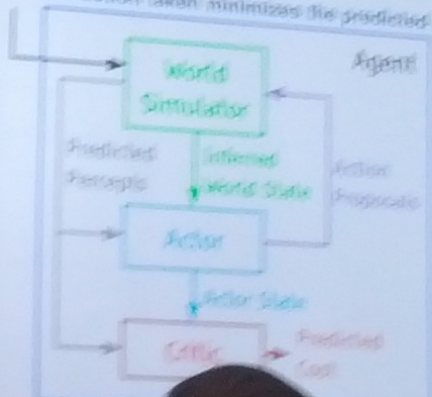


# Predicting + Planning = Reasoning



## AI System: Predicting + Planning = Reasoning

- The essence of intelligence is the ability to predict!
- To plan ahead, we simulate the world!
- The action taken minimizes the predicted cost!





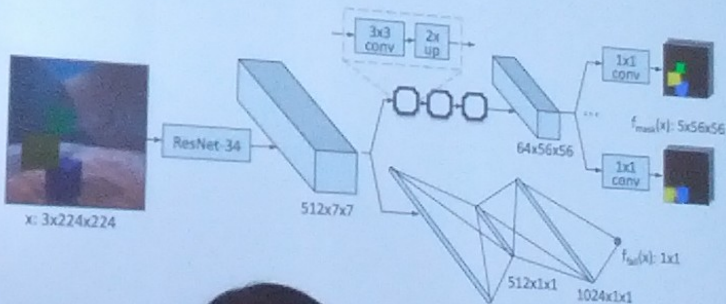


# PhysNet: Learning Physics

## Learning Physics (PhysNet)

- [Lerer, Gross, Fergus arxiv:1603.01312]
- ▶ ConvNet produces object masks that predict the trajectories of falling blocks
- ▶ Uses the Unreal game engine.

Y LeCun





# Memory/Stack-Augmented Recurrent Nets

## Memory/Stack-Augmented Recurrent Nets

- [Joulin & Mikolov, ArXiv:1503.01007]
  - ▶ Stack-augmented RNN
- [Sukhbaatar, Sziem, Weston, Fergus NIPS 2015]
  - ▶ ArXiv:1503.08895
- Weakly-supervised MemNN:
  - ▶ discovers which memory location to use.

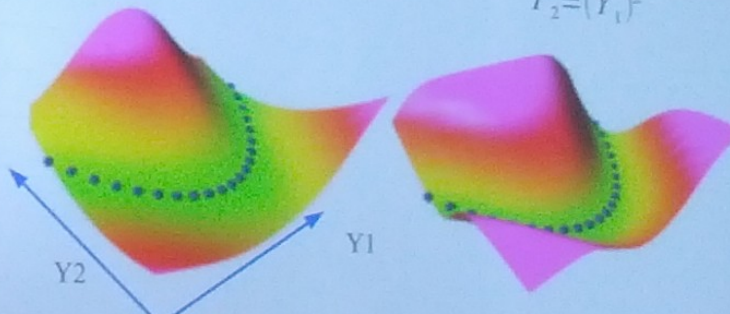


# Capturing Dependencies with an Energy Function

## Capturing Dependencies Between Variables with an Energy Function

- The energy surface is a “contrast function” that takes low values on the manifold, and higher values everywhere else
- ▶ Special case: energy = negative log density
- ▶ Example: the samples live in the manifold

$$Y_2 = (Y_1)^2$$



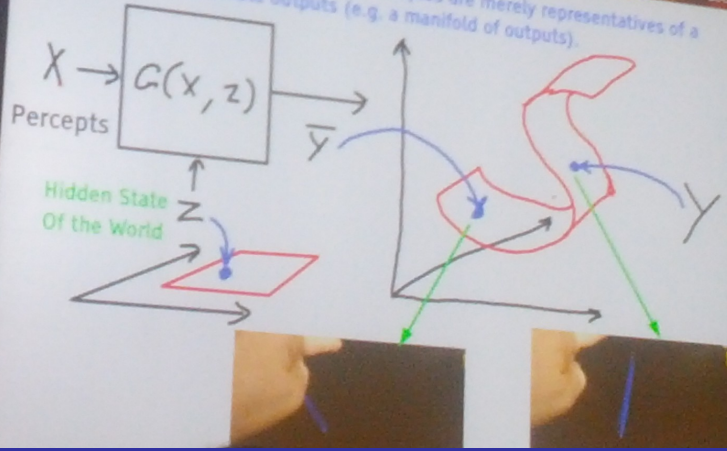


# Train GAN to Predict World Behaviour

## The Hard Part: Prediction Under Uncertainty

Y Lecun

■ Invariant prediction: The training samples are merely representatives of a whole set of possible outputs (e.g. a manifold of outputs).





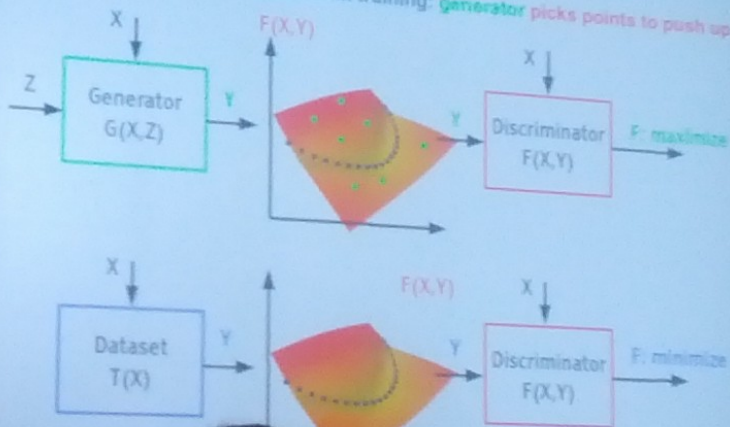
# Adversarial Training

f

## Adversarial Training: A Trainable Objective Function

Y Lec 6

- Adversarial Training [Goodfellow et al. NIPS 2014]
- Energy-based view of adversarial training: **generator** picks points to push up





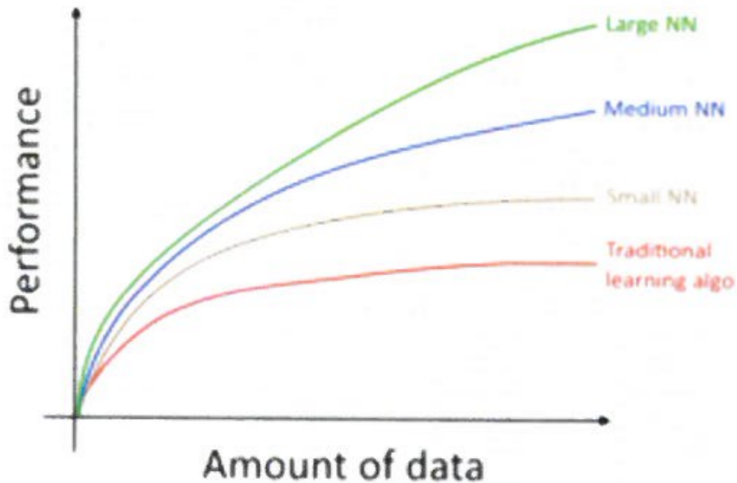


# Giveaways: New Tux Mutations





# Andrew Ng: Nuts and Bolts of DL



Andrew Ng, handout slides





# The Rise of End-to-End Learning

Learning with integer or real-valued outputs:

Problem	X	Y
Spam classification	Email	Spam/Not spam (0/1)
Image recognition	Image	Integer label
Housing price prediction	Features of house	Price in dollars
Product recommendation	Product & user features	Chance of purchase

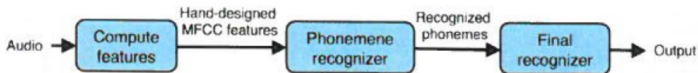
Learning with complex (e.g., string valued) outputs:

Problem	X	Y	Example
Image captioning	Image	Text	Mao et al., 2014
Machine translation	English text	French text	Suskever et al., 2014
Question answering	(Text, Question) pair	Answer text	Bordes et al., 2015
Speech recognition	Audio	Transcription	Hannun et al., 2015
TTS	Text features	Audio	van der Oord et al., 2016

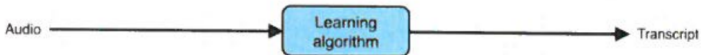


# End-to-End Learning: Speech Recognition

## Traditional model



## End-to-end learning



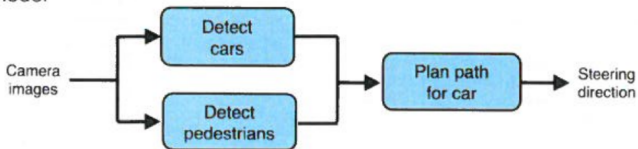
This works well given enough labeled (audio, transcript) data.

- ▶ the era of hand-coded features is over...

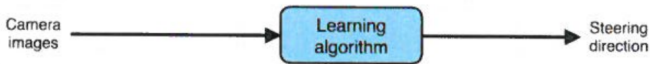


# End-to-End Learning: Autonomous Driving

## Traditional model



## End-to-end learning

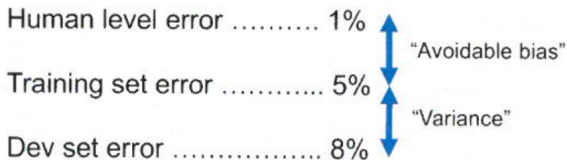


Given the safety-critical requirement of autonomous driving and thus the need for extremely high levels of accuracy, a pure end-to-end approach is still challenging to get to work. End-to-end works only when you have enough (x,y) data to learn function of needed level of complexity.



# Traditional Training Data Split

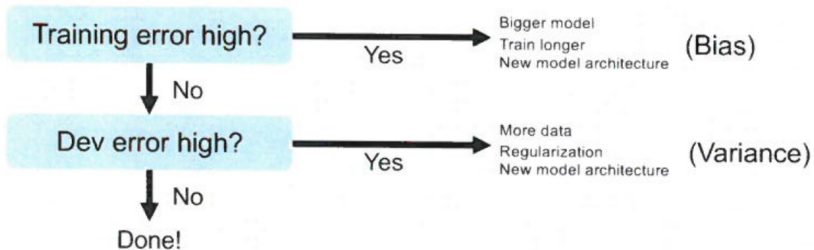
Say you want to build a human-level speech recognition system. You split your data into train/dev/test:



Compared to earlier eras, we still talk about bias and variance, but somewhat less about the "tradeoff" between them.



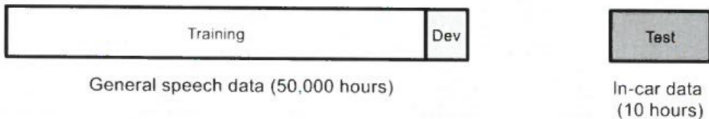
# Basic Machine Learning Recipe





# Check your Training Data Distributions

Say you want to build a speech recognition system for a new in-car rearview mirror product. You have 50,000 hours of general speech data, and 10 hours of in-car data. How do you split your data? This is a **bad** way to do it:

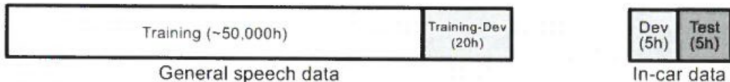


Having mismatched dev and test distributions is not a good idea. Your team may spend months optimizing for dev set performance only to find it doesn't work well on the test set.



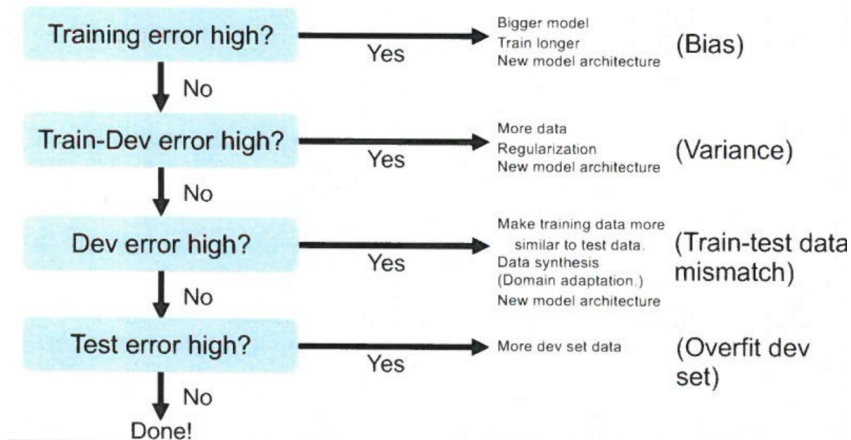
# Better Training Data Split

Better way: Make the dev and test sets come from the same distribution.





# Improved Machine Learning Recipe







# General Human/Bias/Variance Analysis

	General speech data (50,000 hours)	In-car speech data (10 hours)
Performance of humans	Human-level error	(Carry out human evaluation to measure.)
Performance on examples you've trained on	Training error	(Insert some in-car data into training set to measure.)
Performance on examples you haven't trained on	Training-Dev error	Dev/Test error

↑ "Avoidable bias"  
↓ "Variance"/degree of overfitting

← Data mismatch →



# Human Level Performance

You'll often see the fastest performance improvements on a task while the ML is performing worse than humans.

- Human-level performance is a proxy for Bayes optimal error, which we can never surpass.
- Can rely on human intuition: (i) Have humans provide labeled data.  
(ii) Error analysis to understand how humans got examples right.  
(iii) Estimate bias/variance. E.g., On an image recognition task, training error = 8%, dev error = 10%. What do you do? Two cases:





Suppose that on an image labeling task:

Typical human ..... 3% error

Typical doctor ..... 1% error

Experienced doctor ..... 0.7% error

Team of experienced doctors .... 0.5% error

What is “human-level error”?

Answer: For purpose of driving ML progress, 0.5% is best answer since it's closest to Bayes error.

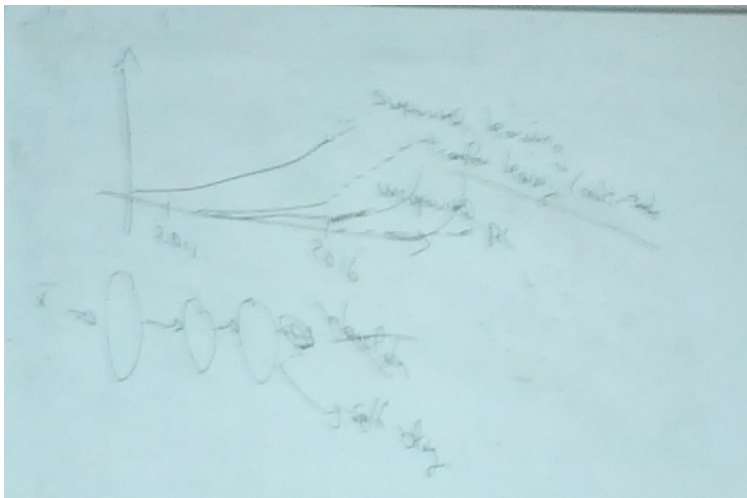


proven Stanford Univ. approach to PhD research:

- ▶ read (lots of) papers
- ▶ try to reproduce their results
- ▶ understand what works and what doesn't
  
- ▶ get your hands dirty (real data, real systems)



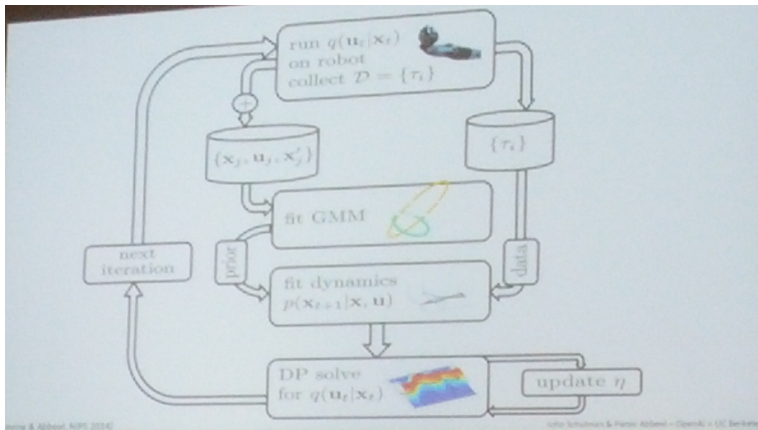
# Deep-Learning Outlook...



- ▶ low-hanging fruit: supervised, larger datasets, transfer learning
- ▶ mid-term: unsupervised learning
- ▶ far in the future: real robust RL applications

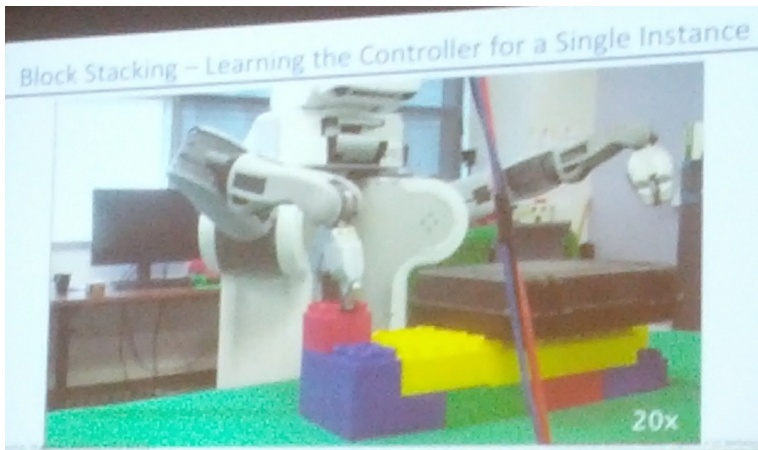


# Abbeel & Shulman: Deep RL Tutorial



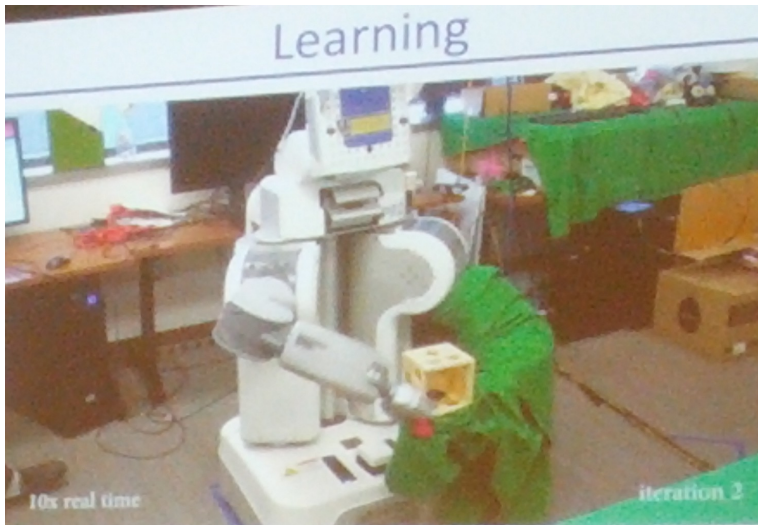


# PR2 Manipulation Learning: Duplo

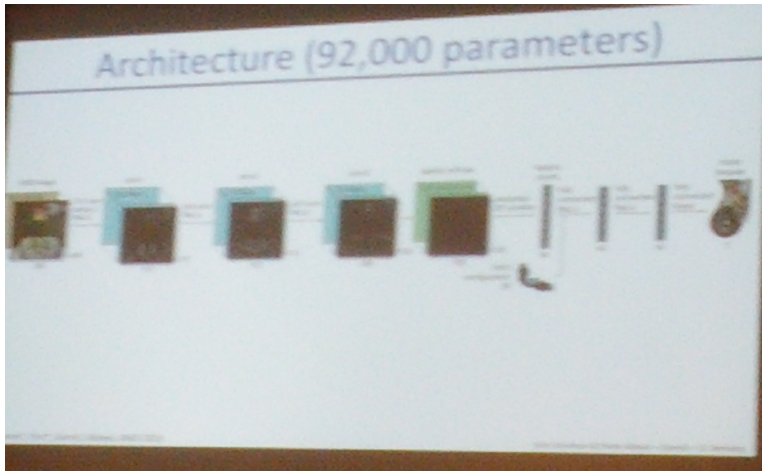




# PR2 Manipulation Learning: Shape-sorting Box

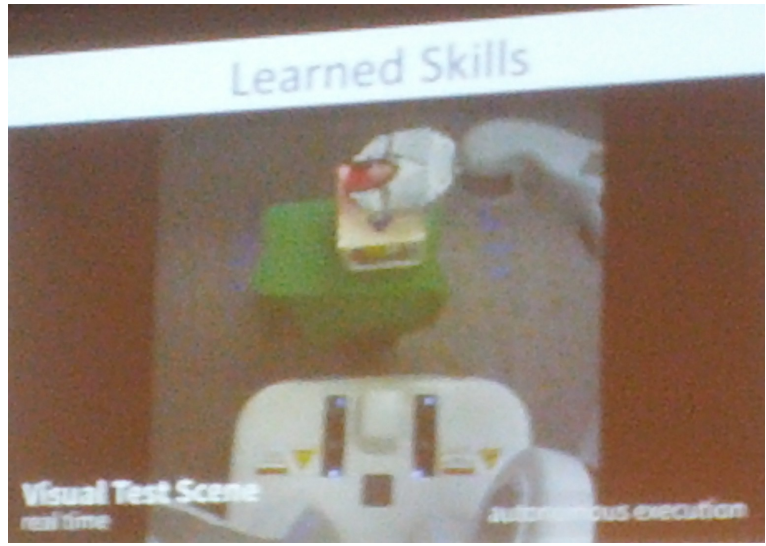








# PR2 Manipulation learning: Autonomous Execution





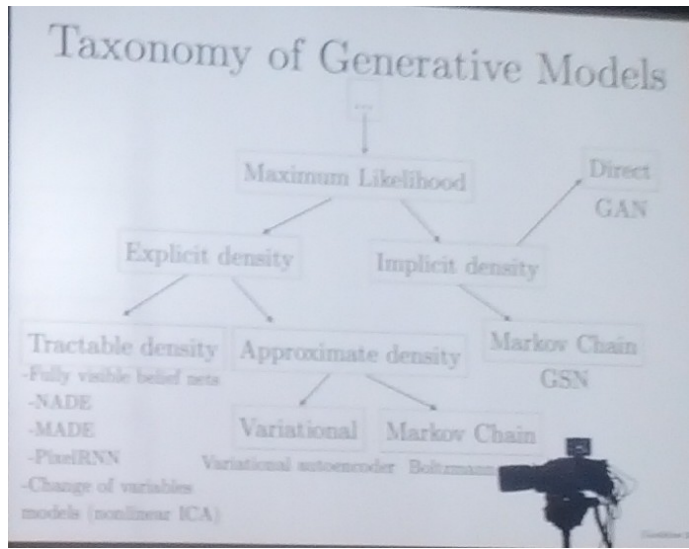
- ▶ off-policy policy gradients / Off-policy actor-critic
- ▶ exploration
- ▶ auxiliary objectives
- ▶ multi-task and transfer (incl. sim2real)
- ▶ meta-RL
- ▶ 24/7 data collection
- ▶ safety
- ▶ architectures
- ▶ inverse RL, model-based RL, hierarchical RL

## How to get started?

- ▶ deep RL courses on the web
- ▶ deep RL code bases
- ▶ environments, e.g. OpenAI Universe

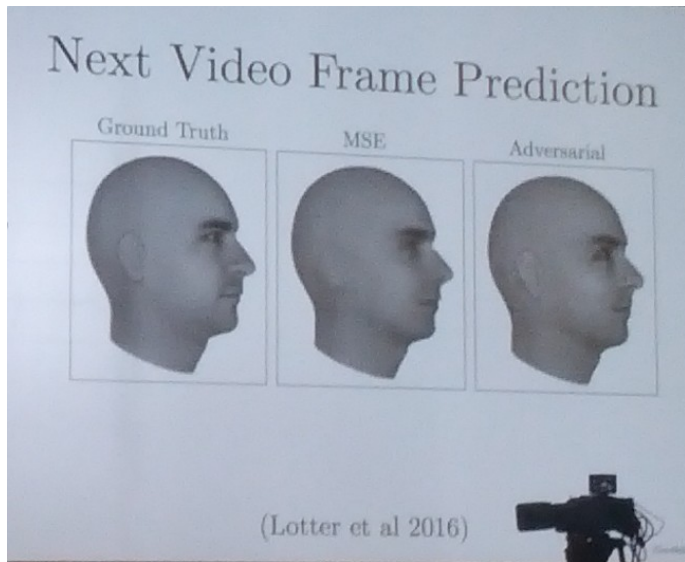


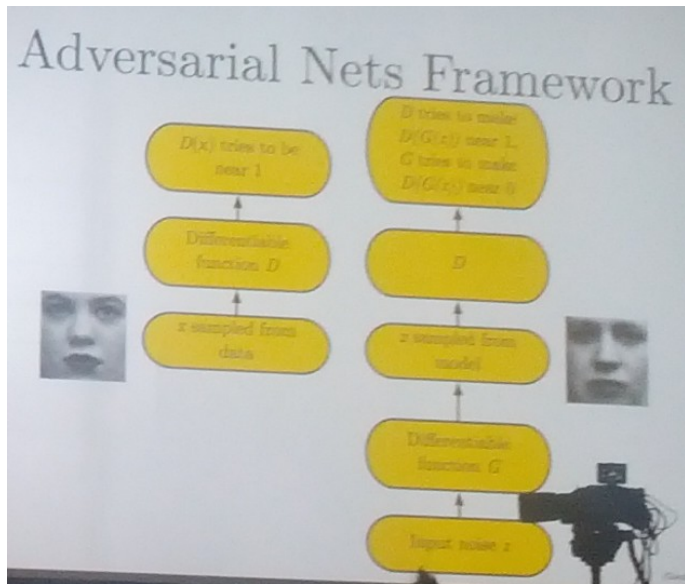
<https://universe.openai.com>





# GAN: Video Frame Prediction







## Minimax Game

$$J^{(D)} = -\frac{1}{2} \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}} \log D(\mathbf{x}) - \frac{1}{2} \mathbb{E}_{\mathbf{z}} \log (1 - D(G(\mathbf{z})))$$

$$J^{(G)} = -J^{(D)}$$

- Equilibrium is a saddle point of the discriminator loss
- Resembles Jensen-Shannon divergence
- Generator minimizes the log-probability of the discriminator being correct





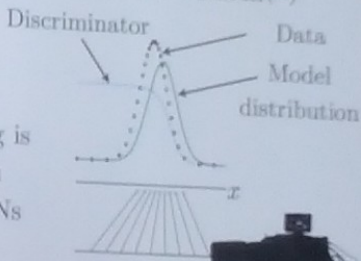


## Discriminator Strategy

Optimal  $D(x)$  for any  $p_{\text{data}}(x)$  and  $p_{\text{model}}(x)$  is always

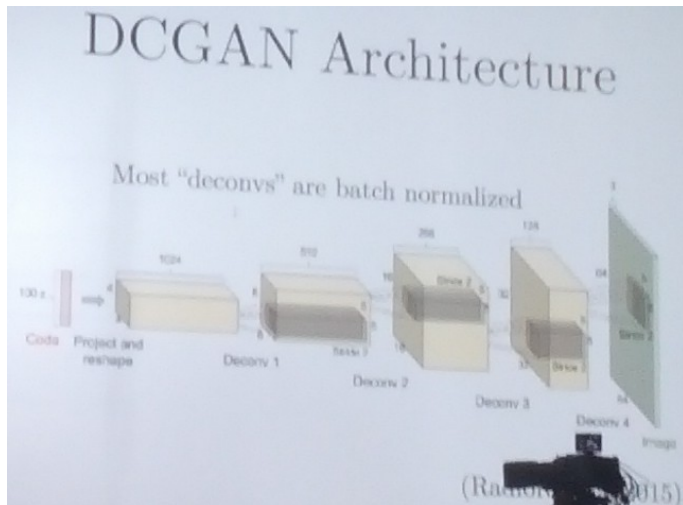
$$D(x) = \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_{\text{model}}(x)}$$

Estimating this ratio using supervised learning is the key approximation mechanism used by GANs



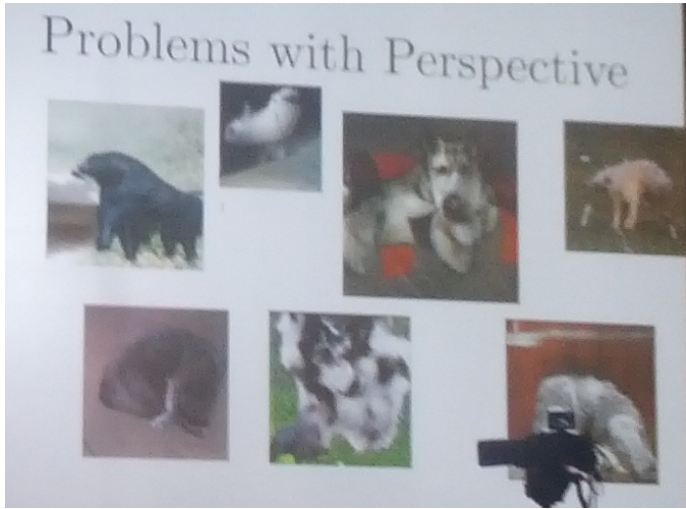


# GAN: DCGAN Architecture





# GAN: Problems with Perspective





## NIPS 2016

Review Process Analysed

## Plenary Speech and Tutorials

Predictive Learning

Nuts and Bolts of Deep Learning

Deep RL Tutorial

Generative Adversarial Networks

## Workshops

Efficient Methods for DNNs

Continual Learning Workshop

Neurobotics Workshop



# Workshop on Efficient Methods for DNNs

M. Rastegari & M. Courbariaux, <http://allenai.org/plato/emdnn/>

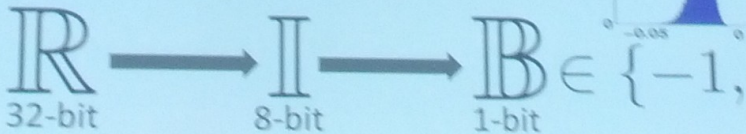
- ▶ nVidia
- ▶ Intel „Nervana“ (hardware+compiler+cloud)
- ▶ binarized networks, real-time on iPhone/Raspberry
- ▶ FPGAs and ASICs
- ▶ good poster session
  
- ▶ fun discussions
  - ▶ Intel: we have 10x more multipliers
  - ▶ nVidia: what you have is vaporware



## Lower Precision



### Reducing Precision

- Saving Memory
- Saving Computation





# Float vs. Binary

 * 	Operations	Memory	Computation
$\mathbb{R}$ * $\mathbb{R}$	+ - ×	1x	1x
$\mathbb{R}$ * $\mathbb{B}$	+ -	~32x	~2x
$\mathbb{B}$ * $\mathbb{B}$	XNOR Bit-count	~32x	~58x

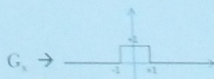
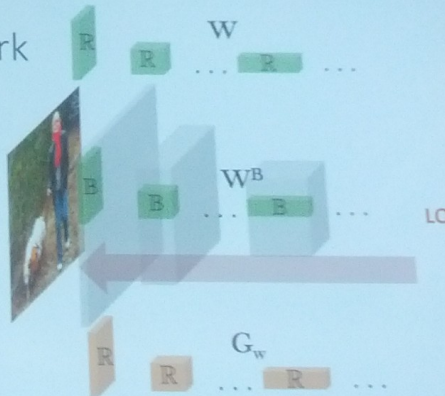


# Deep Binary Weight Network

## Binary Weight Network

*Train for binary weights:*

1. Randomly initialize  $W$
2. For  $iter = 1$  to  $N$
3. Load a random input image  $X$
4.  $W^B = \text{sign}(W)$
5.  $\alpha = \frac{\|W\|_1}{n}$
6. Forward pass with  $\alpha, W^B$
7. Compute loss function  $C$
8.  $\frac{\partial C}{\partial W} = \text{Backward pass with } \alpha, W^B$
9. Update  $W$  ( $W = W - \frac{\partial C}{\partial W}$ )

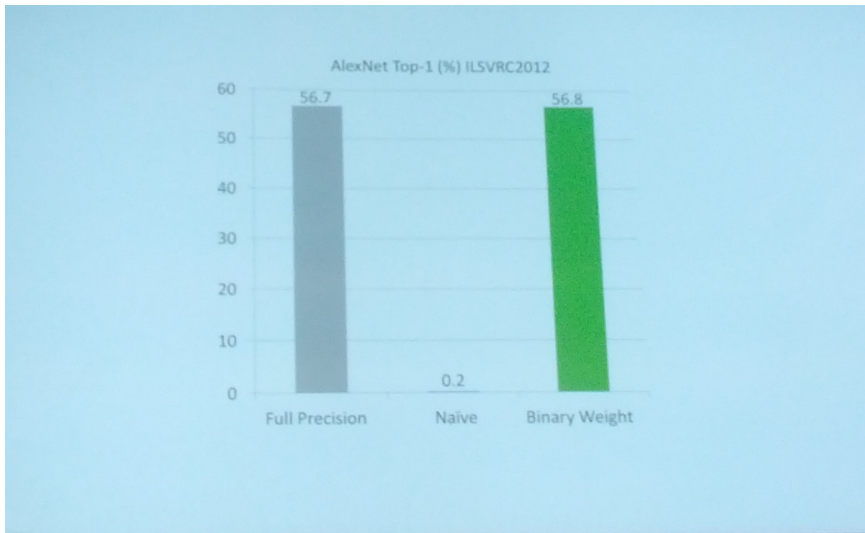


[Hinton et al. 2012]





# AlexNet vs. Binary AlexNet





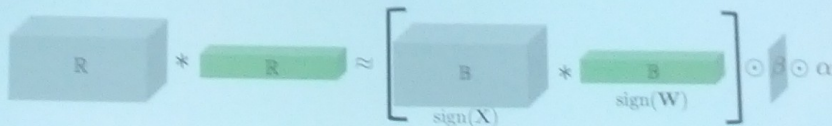
## Binary Input and Binary Weight (XNOR-Net)

$$\underbrace{\underbrace{R}_{X} \odot \underbrace{R}_{W}}_Y \approx \underbrace{\beta \alpha}_{\gamma} \underbrace{\underbrace{B}_{X^B} \odot \underbrace{B}_{W^B}}_{Y^B}$$

$$Y \approx \gamma Y^B$$

$$Y^{B^*}, \gamma^* = \arg \min_{Y^B, \gamma} \|Y - \gamma Y^B\|^2$$

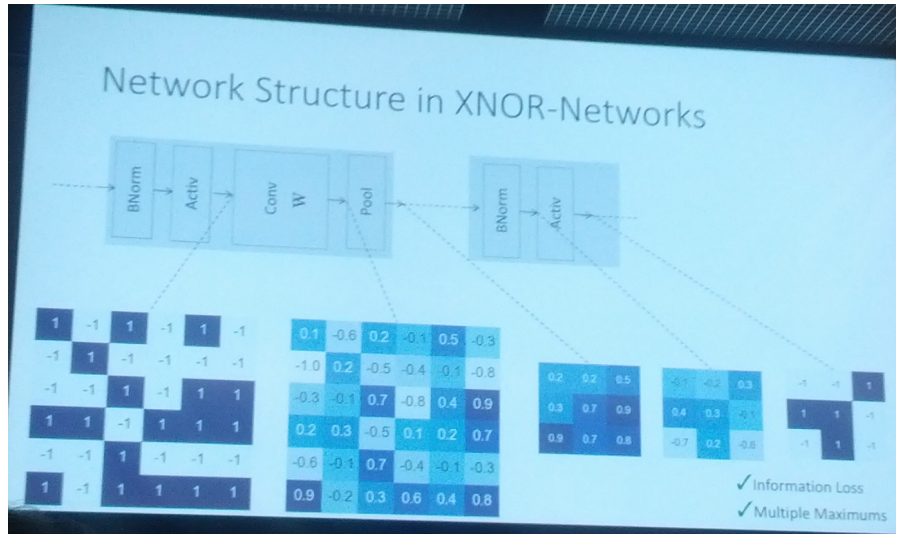
$$Y^{B^*} = \text{sign}(Y) \quad \gamma^* = \frac{1}{n} \|Y\|_1$$



1. Randomly initialize  $W$
2. For  $iter = 1$  to  $N$
3. Load a random input image  $X$
4.  $W^B = \text{sign}(W)$
5.  $\alpha = \frac{\|W\|_1}{n}$
6. Forward pass with  $\alpha, W^B$
7. Compute loss function  $C$
8.  $\frac{\partial C}{\partial W} =$  Backward pass with  $\alpha, W^B$
9. Update  $W$  ( $W = W - \frac{\partial C}{\partial W}$ )

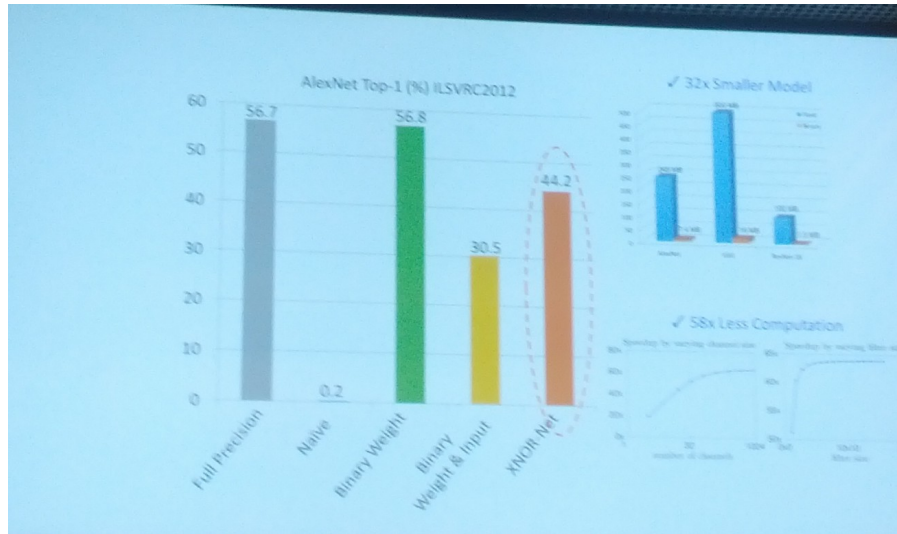


# Network Structure in XNOR-Nets





# XNOR-Net Performance



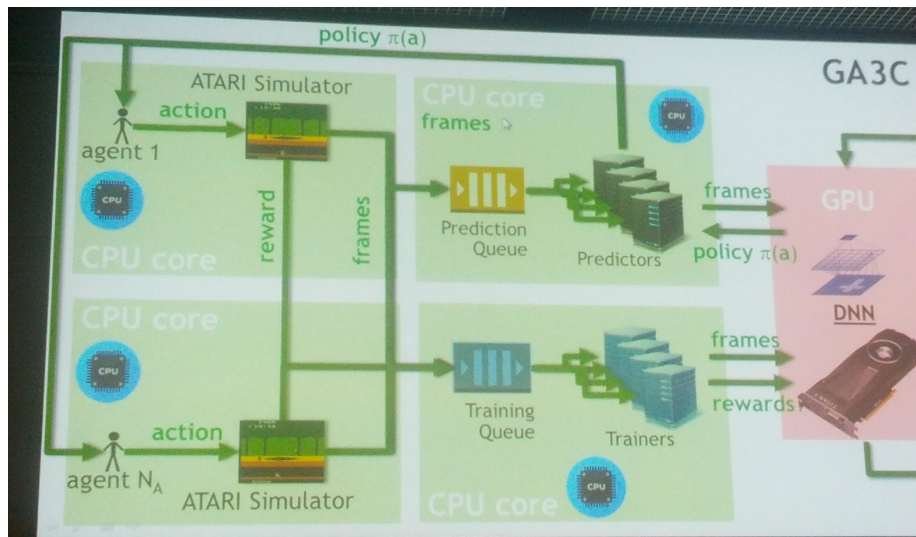


# Live Demo: AlexNet on iPhone and Raspberry Zero





# nVidia: A3C on GPU

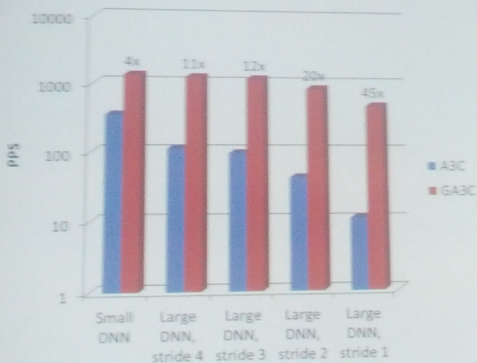




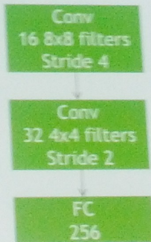
# nVidia: GA3C vs A3C

## GA3C vs. A3C\*

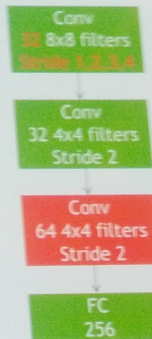
(\* our TensorFlow implementation)



Small DNN (A3C)



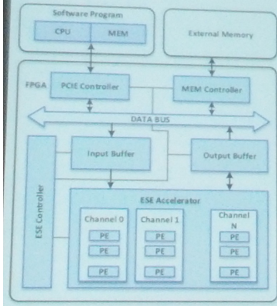
Large DNN Stride 1,2,3,4



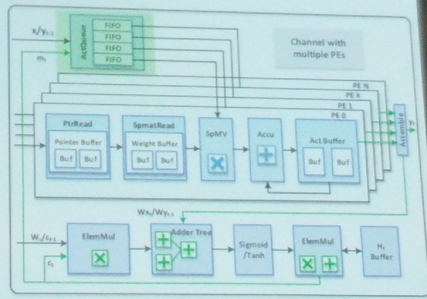




## Hardware Architecture



(a)



(b)

Expression

Scheduling

Acceleration

Stanford University



# DNN FPGA Speedup and Efficiency

## Speedup and Energy Efficiency

Plat.	Matrix	Matrix Size	Sparsity (%) <sup>1</sup>	Compress. Matrix (Bytes) <sup>2</sup>	ESE on FPGA (ours)					CPU		GPU		
					Theoret. Comput. Time (μs)	Real Comput. Time (μs)	Total Operat. (GOP)	Real Perform. (GOP/s)	Equ. Operat. (GOP)	Equ. Perform. (GOP/s)	Dense	Sparse	Dense	Sparse
	$W_{ix}$	1024 × 153	11.7	18304	2.9	5.36	0.0012	218.6	0.010	1870.7				
	$W_{fx}$	1024 × 153	11.7	18272	2.9	5.36	0.0012	218.2	0.010	1870.7				
	$W_{ox}$	1024 × 153	11.8	18560	2.9	5.36	0.0012	221.6	0.010	1870.7	1518.4 <sup>3</sup>	670.4	34.2	58.0
	$W_{oz}$	1024 × 153	11.5	17984	2.8	5.36	0.0012	214.7	0.010	1870.7				
	$W_{ix}$	1024 × 512	11.3	59360	9.3	10.31	0.0038	368.5	0.034	3254.6				
	$W_{fx}$	1024 × 512	11.5	60416	9.4	10.01	0.0039	386.3	0.034	3352.1				
	$W_{ox}$	1024 × 512	11.2	58880	9.2	9.89	0.0038	381.2	0.034	3394.5	3225.0 <sup>4</sup>	2288.0	81.3	166.0
	$W_{oz}$	1024 × 512	11.5	60128	9.4	10.04	0.0038	383.5	0.034	3343.7				
	$W_{ym}$	512 × 1024	10.0	52416	8.2	18.66	0.0034	214.2	0.034	2142.7	1273.9	611.5	124.8	63.4
<b>Total</b>	<b>3248128</b>	<b>11.2</b>	<b>364320</b>	<b>57.0</b>	<b>82.7</b>	<b>0.0233</b>	<b>282.2</b>	<b>0.208</b>	<b>2515.7</b>	<b>6017.3</b>	<b>3569.9</b>	<b>240.3</b>	<b>287.4</b>	

	ESE	CPU		GPU	
		Dense	Sparse	Dense	Sparse
Latency	82.7us	6017us	3569us	240us	287us
Power	41W	111W	38W	202W	136W
Performance	2.9x	0.039	0.067	1x	0.84
Energy Efficiency	14.3x	0.071	0.355	1x	1.25
Compression Ratio	20x	1	10	1x	10

Scheduling

Acceleration

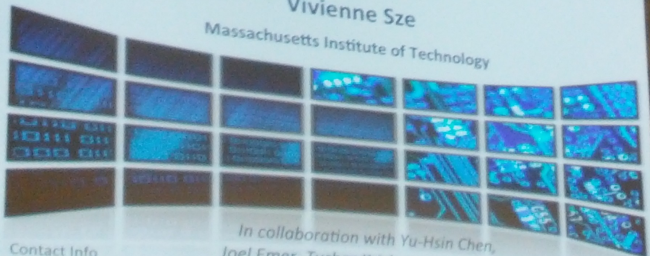
Stanford University



# Vivienne Sze: EyeRiss Energy-Efficient DNNs

Joint Design of Algorithms and Hardware for Energy-Efficient DNNs

Vivienne Sze  
Massachusetts Institute of Technology

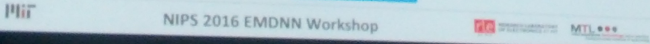


*In collaboration with Yu-Hsin Chen,  
Joel Emer, Tushar Krishna, Tien-Ju Yang*

Contact Info  
email: [sze@mit.edu](mailto:sze@mit.edu)  
website: [www.rle.mit.edu/eems](http://www.rle.mit.edu/eems)

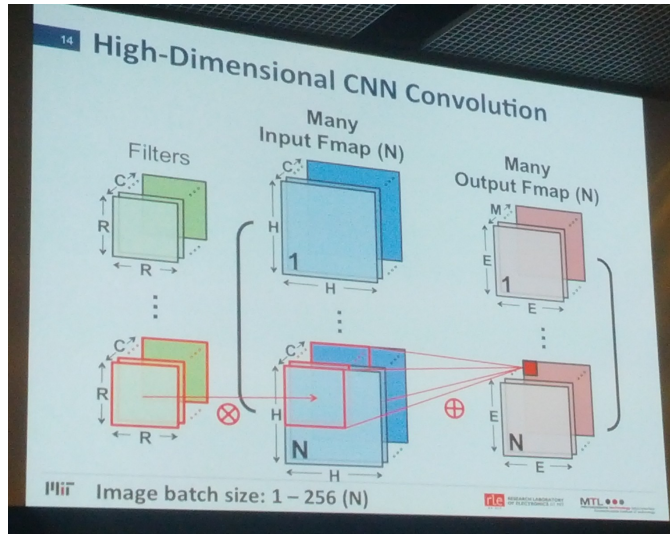
[Follow @eems\\_mit](#)

MIT NIPS 2016 EMDNN Workshop





# High-Dimensional CNN Convolution



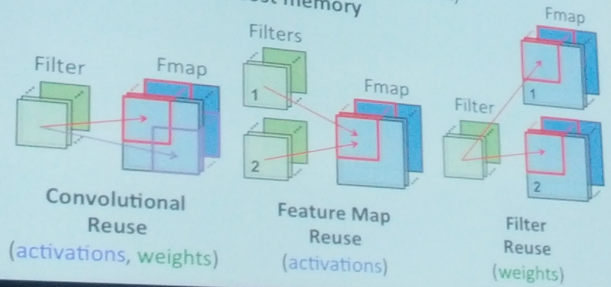


# Properties we can leverage

19

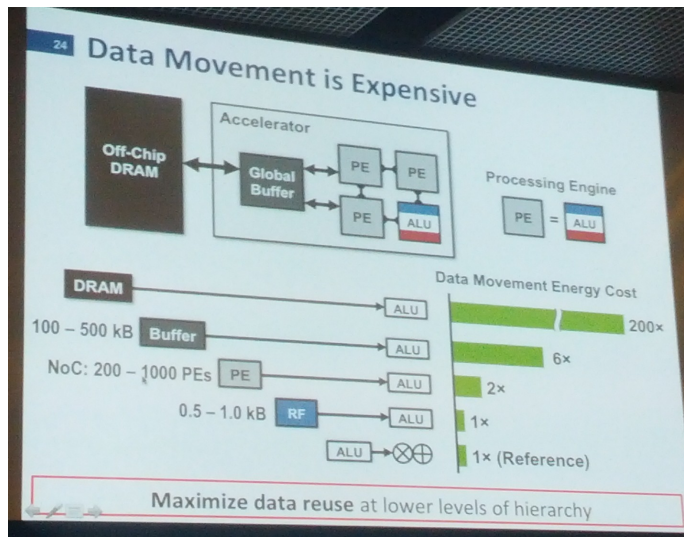
## Properties We Can Leverage

- Operations exhibit **high parallelism**  
→ high throughput possible
- **Input data reuse** opportunities (up to 500x)  
→ exploit **low-cost memory**





# Data Movement is Expensive





# CNN Convolution Options

41 CNN Convolution – The Full Picture

Multiple **fmaps**:  
 Filter 1 \* Fmap 1 & 2 = Psum 1 & 2

Multiple **filters**:  
 Filter 1 & 2 \* Fmap 1 = Psum 1 & 2

Multiple **channels**:  
 Filter 1 \* Fmap 1 = Psum

Map rows from **multiple fmaps, filters and channels** to same PE to exploit other forms of reuse and local accumulation



## 56 Chip Spec & Measurement Results<sup>1</sup>

Technology	TSMC 65nm LP 1P9M
On-Chip Buffer	108 KB
# of PEs	168
Scratch Pad / PE	0.5 KB
Core Frequency	100 – 250 MHz
Peak Performance	33.6 – 84.0 GOPS
Word Bit-width	16-bit Fixed-Point
Natively Supported CNN Shapes	Filter Width: 1 – 32
	Filter Height: 1 – 12
	Num. Filters: 1 – 1024
	Num. Channels: 1 – 1024
	Horz. Stride: 1–12
	Vert. Stride: 1, 2, 4

4000 µm

4000 µm

Global Buffer

Spatial Array (168 PEs)

AlexNet: For 2.66 GMACs [8 billion 16-bit inputs (16GB) and 2.7 billion outputs (5.4GB)], only requires 208.5MB (buffer) and 15.4MB (DRAM)

MIT

MIT

MIT





# Comparison with (mobile) GPU

56

## Comparison with GPU

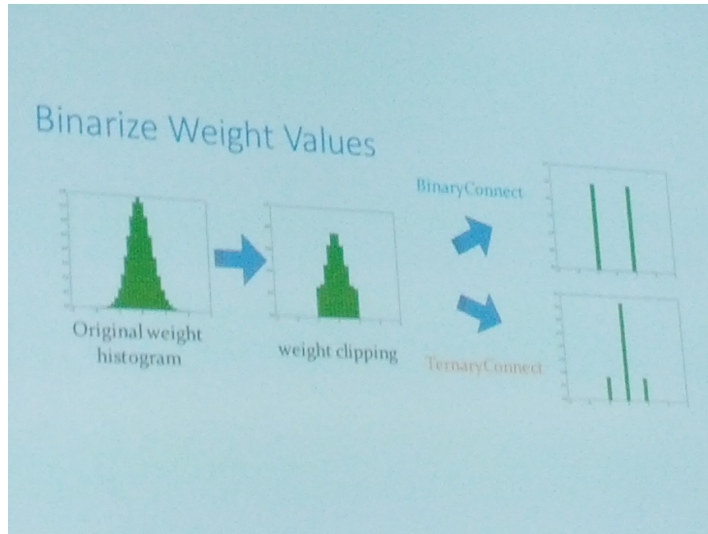
	<i>This Work</i>	NVIDIA TK1 (Jetson Kit)
Technology	65nm	28nm
Clock Rate	200MHz	852MHz
# Multipliers	168	192
On-Chip Storage	Buffer: 108KB Spad: 75.3KB	Shared Mem: 64KB Reg File: 256KB
Word Bit-Width	16b Fixed	32b Float
Throughput <sup>1</sup>	34.7 fps	68 fps
Measured Power	278 mW	Idle/Active <sup>2</sup> : 3.7W/10.2W
DRAM Bandwidth	127 MB/s	1120 MB/s <sup>3</sup>

1. AlexNet Convolutional Layers Only
2. Board Power
3. Modeled from [Tan, SC11]





# Yoshua Bengio: Low Precision Neural Nets





# Binarized Neural Nets

## NIPS'2016: Binarized Neural Networks

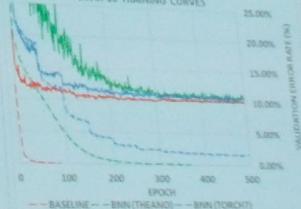
Hubara, Courbariaux, Soudry, El-Yaniv & Bengio

- Binarize not just weights but also ACTIVATIONS
- Gets rid of both MULTIPLICATIONS and ADDITIONS
  - replaced by POPCOUNT (counts #1's in binary vector, 1 op)
  - major speed-up, even on GPU, meant for specialized hardware
- Approaches SOTA: MNIST, CIFAR-10 & SVHN, slower convergence

GPU KERNELS' EXECUTION TIMES



CIFAR-10 TRAINING CURVES





# Continual Learning Workshop

Mark Ring et.al., <https://sites.google.com/site/cldlnips2016/>

<i>Time</i>	<i>Title</i>	<i>Speaker</i>
8:30 AM	Welcoming remarks and introduction	Mark Ring (Cogital)
8:50 AM	Invited talk	Rich Sutton (U of Alberta)
9:20 AM	Spotlight 1	B. Lüders, M. Schläger and S. Risi
9:30 AM	Spotlight 2	T. Miconi
9:40 AM	Invited talk	Claudia Clopath (Imperial College)
10:10 AM	<i>Coffee break (+ poster set-up)</i>	
10:40 AM	Invited talk	Satinder Singh Bajeva (U of Michigan/Cogital)
11:10 AM	Spotlight 3	E. de Jong
11:20 AM	Poster session	(see <a href="#">submission</a> page for full list)
12:20 PM	<i>Lunch</i>	
2:00 PM	Invited talk	Honglak Lee (U of Michigan)
2:30 PM	Spotlight 4	M. Pickett, R. Al-Rfou, L. Shao and C. Tar
2:40 PM	Spotlight 5	H. Kasaei
2:50 PM	Invited talk	Eric Eaton (U of Pennsylvania)
3:20 PM	Systems that are alert to context and surprise	Hava Siegelmann (DARPA)
3:30 PM	<i>Coffee break</i>	
4:00 PM	Invited talk	Raia Hasdell (DeepMind)
4:30 PM	Invited talk	Doina Precup (McGill University)
5:00 PM	Panel discussion	(all invited speakers)
6:00 PM	<i>Workshop ends</i>	



# Satinder Singh: Simple Continual Learning Demonstration

## Some elements of Continual Learning

- Learn new Skills (Options)
- Learn new Knowledge (Predictions/Models)
- Reuse / Incorporate learned Skills and Knowledge to learn more complex Skills and Knowledge
- Intrinsic Motivation to drive experience in the absence of (or more accurately too long a delay in) Extrinsic Rewards
- More experienced agents (humans) as a particularly salient target of Intrinsic Motivation
- Likely needs a Cognitive Architecture that supports control of internal processing
- Increasingly competent agent over time (not just in terms of Knowledge and Skills it has but also in terms at how well it does at accumulating Extrinsic Rewards)



# Neurobotics Workshop

<https://rueckert.lima-city.de/NIPSW2016/WebContent>

Day One, Neurobotics WS, Fri Dec 9th 2016		
	14.20-14.30	Introduction by <b>Elmar Rueckert</b> and <b>Martin Riedmiller</b>
Session One: Reinforcement Learning, Imitation, and Active Learning		
1	14.30-15.00	<b>Juergen Schmidhuber</b> (Scientific Director of the Swiss AI Lab IDSIA)
	15.00-15.30	Posters and Coffee
2	15.30-16.00	<b>Sergey Levine</b> (University of California, Berkeley)
3	16.00-16.30	<b>Pieter Abbeel</b> (University of California, Berkeley)
4	16.30-17.00	<b>Johanni Brea</b> (École polytechnique fédérale de Lausanne, EPFL)
	17.00-17.20	Posters and Coffee
5	17.20-17.45	<b>Paul Schrater</b> (University of Minnesota)
6	17.45-18.10	<b>Frank Hutter</b> (University Freiburg)
7	18.10-18.35	<b>Raia Hadsell</b> (Google DeepMind)
	18.35-19.00	Panel Discussion, Session One

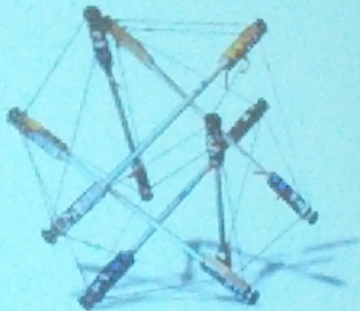
## day two

Day Two, Neurobotics WS, Sat Dec 10th 2016		
Session One: Reinforcement Learning, Imitation, and Active Learning		
	08.30-08.35	Introduction by <b>Elmar Rueckert</b> and <b>Martin Riedmiller</b>
8	08.35-09.05	<b>Robert Legenstein</b> (Graz University of Technology)
9	09.05-09.35	<b>Sylvain Calinon</b> (Idiap Research Institute, EPFL Lausanne)
10	09.35-10.05	<b>Chelsea Finn</b> (University of California, Berkeley)
11	10.05-10.35	<b>Peter Stone</b> (University of Texas at Austin)
	10.35-11.00	Posters and Coffee
12	11.00-11.30	<b>Paul Verschure</b> (Catalan Institute of Advanced Research)
Session Two: Model Representations and Features		
13	11.30-12.00	<b>Tobi Delbrück</b> (University of Zurich and ETH Zurich)
14	12.00-12.30	<b>Moritz Grosse-Wentrup</b> (Max Planck Institute Tuebingen)
15	12.30-13.00	<b>Kristian Kersting</b> (Technische Universität Dortmund)
	13.00-14.00	Lunch break
Session Three: Feedback and Control		
16	14.00-14.30	<b>Emo Todorov</b> (University of Washington)
17	14.30-15.00	<b>Richard Sutton</b> (University of Alberta)
	15.00-15.30	Posters and Coffee
18	15.30-16.00	<b>Bert Kappen</b> (Radboud University)
19	16.00-16.30	<b>Jean-Pascal Pfister</b> (University of Zurich and ETH Zurich)
	16.30-17.00	Posters and Coffee
20	17.00-17.30	<b>Jan Babic</b> (Josef Stefan Institute Ljubljana)
21	17.30-18.00	<b>Martin Giese</b> (University Clinic Tübingen)
	18.00-18.30	Panel Discussion, Session Two and Session Three



## Tensegrity Robotics: NASA SuperBall

- Rigid rods connected by elastic cables
- Controlled by motors that extend / contract cables
- Properties:
  - Lightweight
  - Low cost
  - Capable of withstanding significant impact
- NASA investigates them for space exploration



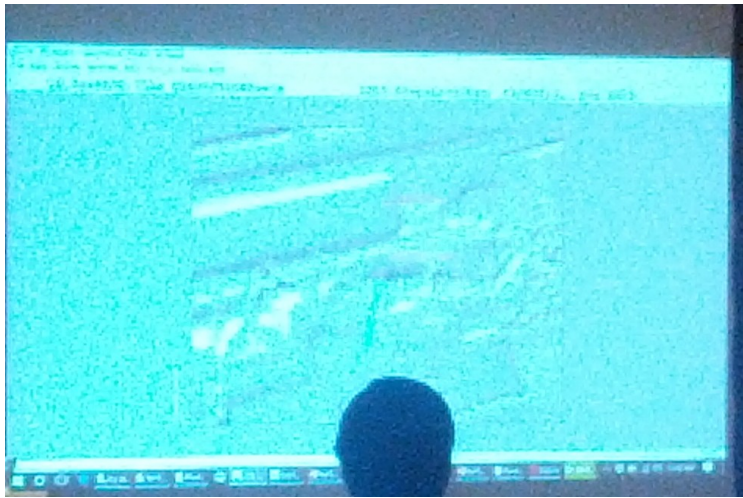


- ▶ custom event-based image sensor, CMOS prototype
- ▶ per pixel brightness ( $\log I$ ) change detection
- ▶ only active pixels are read out (image gradients)
  
- ▶ static scenes: no image readout at all
- ▶ requires custom software to reconstruct full image
- ▶ fast tracking of moving objects
- ▶ ego-motion scene-reconstruction
- ▶ rock-paper-scissors live demo
  
- ▶ insight: spiking neurons a means to save energy



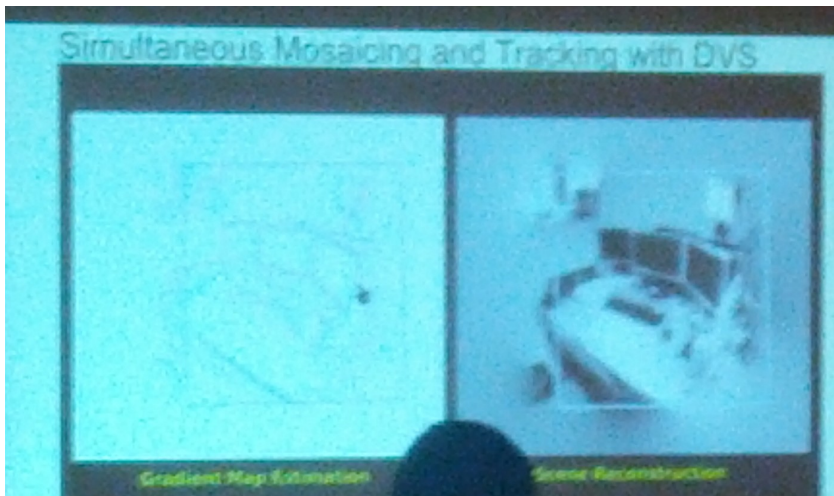


# DVS Camera: Live Demo





# DVS Camera: Tracking





## Simultaneous Optical Flow and Intensity Estimation from an Event Camera

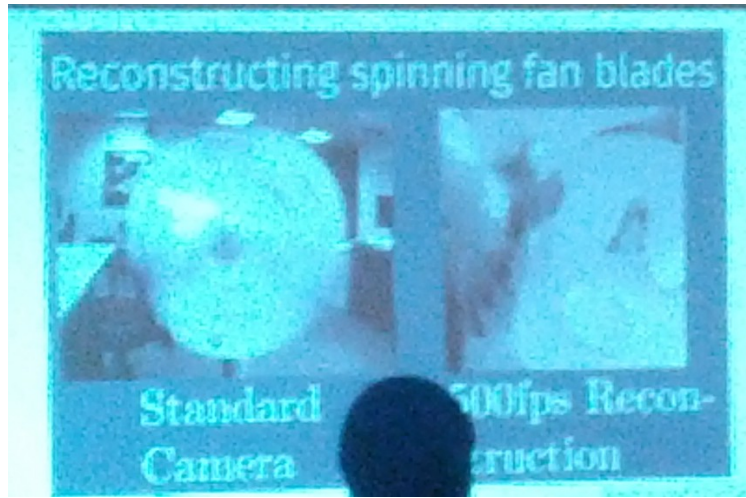
$$\min_{u,v,I} \int_{\Omega} \int_{T_1}^{T_2} (\lambda_1 \|u_x\|^2 + \lambda_2 \|u_y\|^2 + \lambda_3 \|I_x\|^2) \\ + (\lambda_4 \|I_x - \delta(I)\| + \lambda_5 \|I_y - \delta(I)\|) dt dx \\ + \int_{\Omega} \sum_{t=1}^{P(x)} (\lambda_6 |I_t - I_{t-1}| - b p_t) dx_t$$

- Smoothness term
- Optical flow term
- No-event term
- Event term

Patrick Barron, Andrew Owens and Sifan Tang  
 Microsoft Research, Imperial College London  
 2016 Best paper award

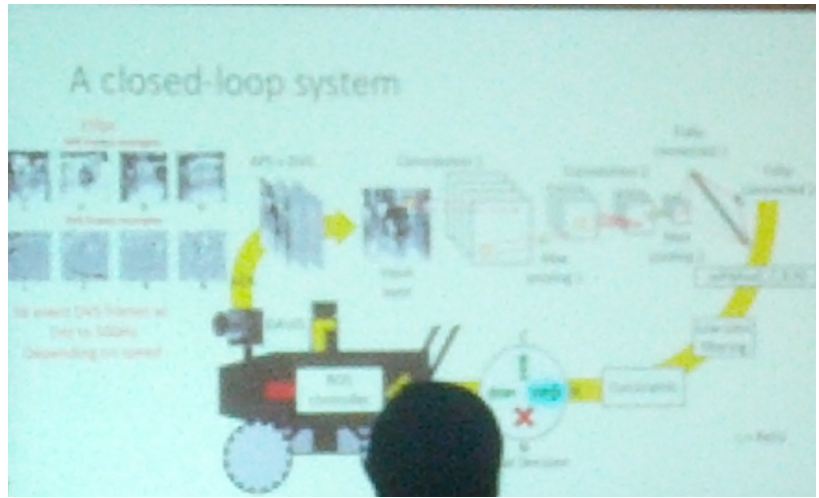


# DVS Camera: 500fps Spinning Fan Blades



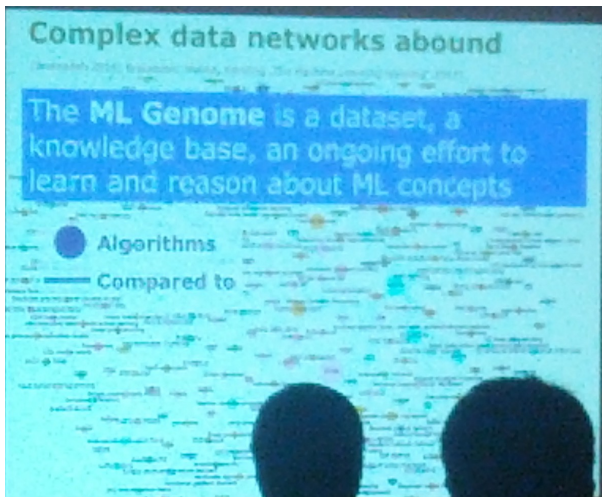


# DVS Camera: Autonomous Vehicle with DNN



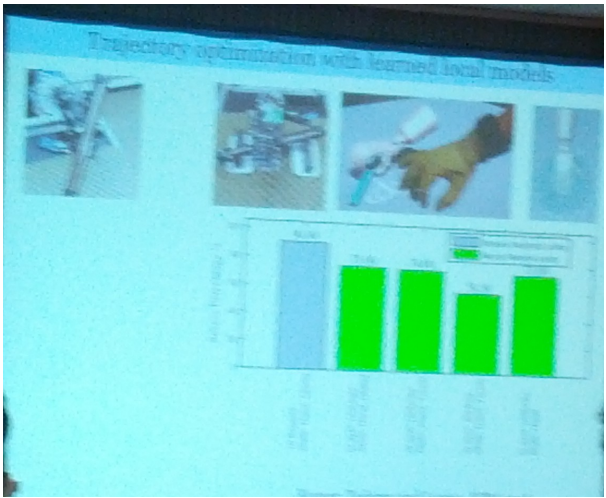


# Kristian Kersting: ML Genome: Learning about Learning Algorithms



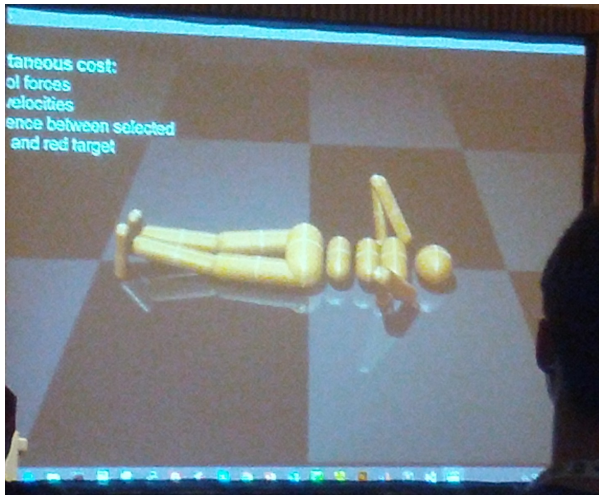


# Emo Todorov: Optimal Control





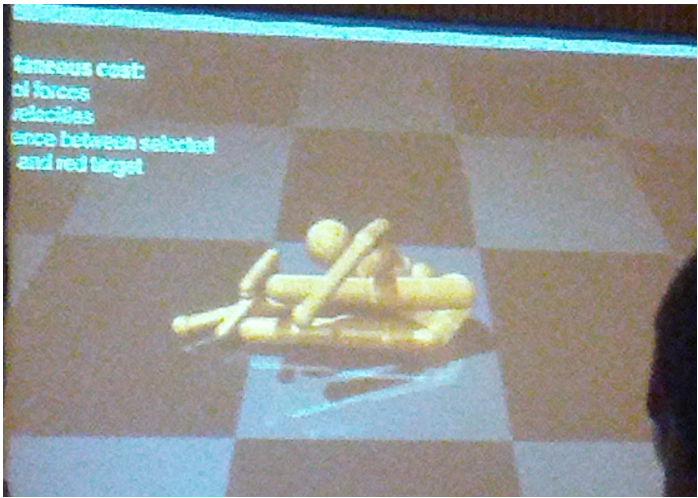
# Goal Directed Dynamics







# Keeping your head up while falling





(disclaimer: from memory, not accurate transcript...)

**Q:** you are using dynamics, are you also going to try DNNs?

**A:** Well, we know that NNs are universal function approximators. So, there exists this network where I write any title and it outputs a complete paper that is then accepted at Nature. Am I going to chase this network? No!



## An Outline of an AI Architecture for Continual Learning & Neurorobotics

Rich Sutton

Reinforcement Learning & Artificial Intelligence Lab  
University of Alberta, Canada  
with thanks to David Borrajo, Sander Singh, and Mark Roy

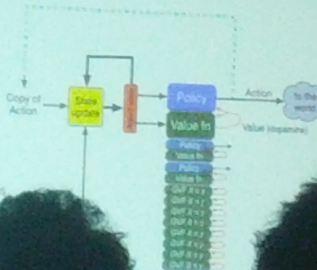




# RL with Auxiliary Tasks

Now let's add some auxiliary tasks, as in UNREAL (Jaderberg et al. 2016)

- These can be handled as additional pseudo-rewards and pseudo-terminations (subgoals).
- each with its learned pseudo-policy and value fn, as in Horde (AAMAS 2011)
- Prediction tasks too (even more numerous).
- each with pseudo-reward, pseudo-termination, & policy
- and learning its own pseudo-value function
- There is a whole world of auxiliary tasks – goals and predictions beyond reward and value – all learned in the usual way
- We can work on a gazillion auxiliary tasks!
- And we can do it in parallel!
- Giving a use for all that computation from Moore's law







## Combined Black-Box and Analytical Optimization

Englert & Toussaint: *Combined Optimization and Reinforcement Learning for Manipulation Skills*, R:SS'16

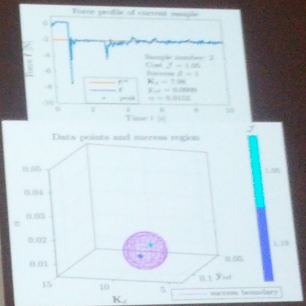

- CORL (Combined Optimization and RL):
  - Policy parameters  $w$
  - **analytically known cost function**  $J(w) = \mathbb{E}\{\sum_{t=0}^T c_t(x_t, u_t) | w\}$
  - **projection**, implicitly given by a constraint  $h(w, \theta) = 0$
  - **unknown black-box return function**  $R(\theta) \in \mathbb{R}$
  - **unknown black-box success constraint**  $S(\theta) \in \{0, 1\}$
  - Problem:

$$\min_{w, \theta} J(w) - R(\theta) \quad \text{s.t.} \quad h(w, \theta) = 0, S(\theta) = 1$$

- Alternate path optimization  $\min_w J(w) \quad \text{s.t.} \quad h(w, \theta) = 0$   
with Bayesian Optimization  $\max_{\theta} R(\theta) \quad \text{s.t.} \quad S(\theta) = 1$

3/20

Exploring the success region  
Exploiting inside



The top plot, titled "True profile of current range", shows a signal over time. The y-axis is labeled "True (A)" and ranges from -10 to 0. The x-axis is labeled "Time (s)" and ranges from 0 to 10. The plot shows a blue line with a sharp initial spike and then a noisy signal fluctuating around a mean value. A legend indicates:  $\mu$  (orange line),  $\sigma$  (blue line), and  $\mu$  (red line). The plot also includes the following statistics: Sample number: 2, Class:  $\mathcal{C}^1 = 1.00$ , Success:  $\mathcal{S} = 1$ ,  $K_{\text{est}} = 1.00$ ,  $\text{Est} = 0.0000$ , and  $\sigma = 0.0000$ .

The bottom plot, titled "Data points and success region", is a 3D scatter plot with axes labeled  $K_{\text{est}}$  (x-axis, 0 to 15),  $\text{Est}$  (y-axis, 0.00 to 0.05), and  $\sigma$  (z-axis, 0.00 to 0.05). A purple shaded sphere represents the success region. A legend at the bottom right indicates "success boundary". A color bar on the right side of the plot ranges from 0.00 (blue) to 1.00 (red).

00:46 01:38

