Cargese 2018-08-27

Deep Learning: Past, Present and Future

Yann LeCun
Facebook AI Research
New York University
http://yann.lecun.com

facebook
Artificial Intelligence Research

Deep Learning Today

History and State of the Art

facebook
Artificial Intelligence Research

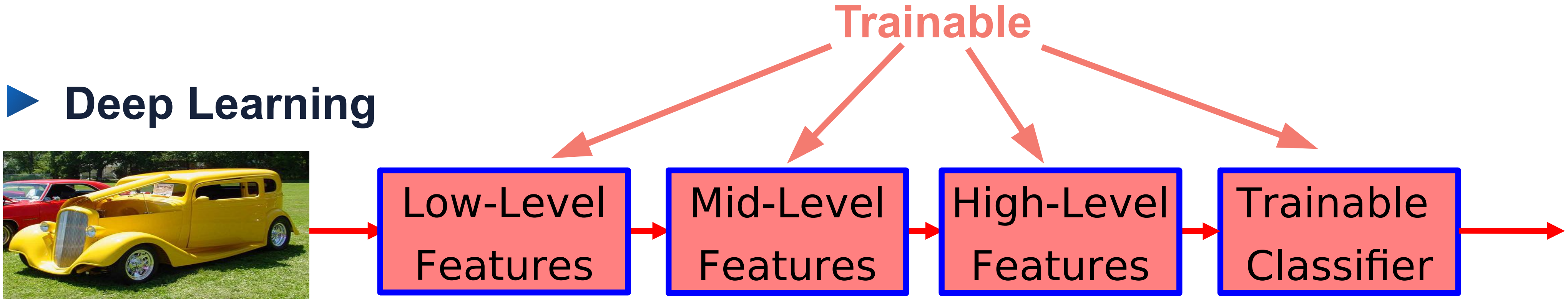
Supervised learning

- **Training a machine by showing examples instead of programming it**
- **When the output is wrong, tweak the parameters of the machine**
- **Works well for:**
  - Speech→words
  - Image→categories
  - Portrait→ name
  - Photo→caption
  - Text→topic
  - ….

Deep Learning

- **Traditional Machine Learning**

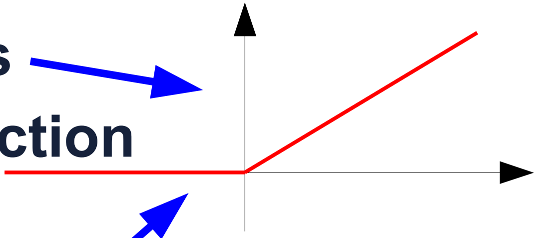
- **Deep Learning**

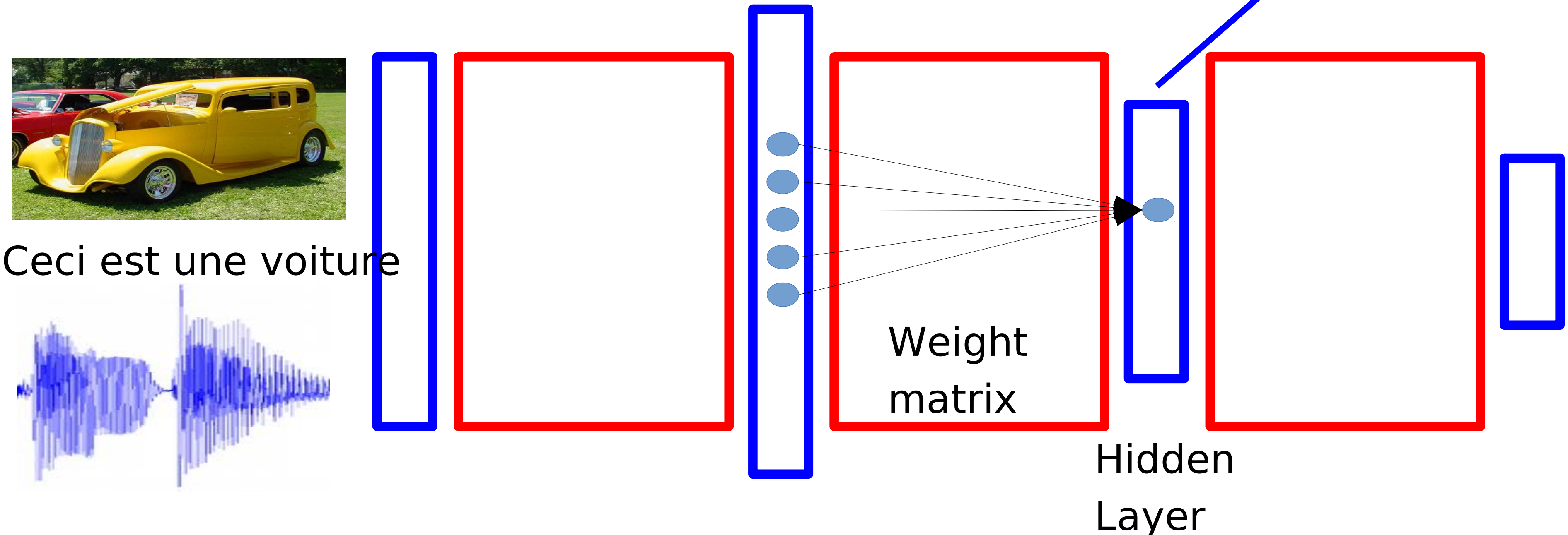


Multi-Layer Neural Nets

- **Multiple Layers of simple units**
- **Each units computes a weighted sum of its inputs**
- **Weighted sum is passed through a non-linear function**
- **The learning algorithm changes the weights**

$$ReLU(x) = max(x, 0)$$



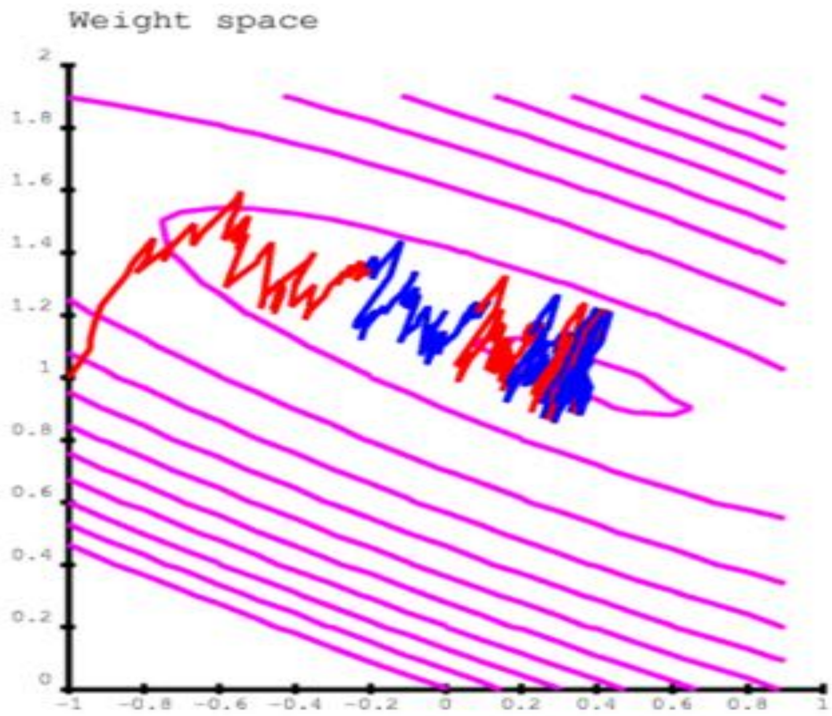


Supervised Machine Learning = Function Optimization

Function with adjustable parameters

Objective Function

traffic light: -1

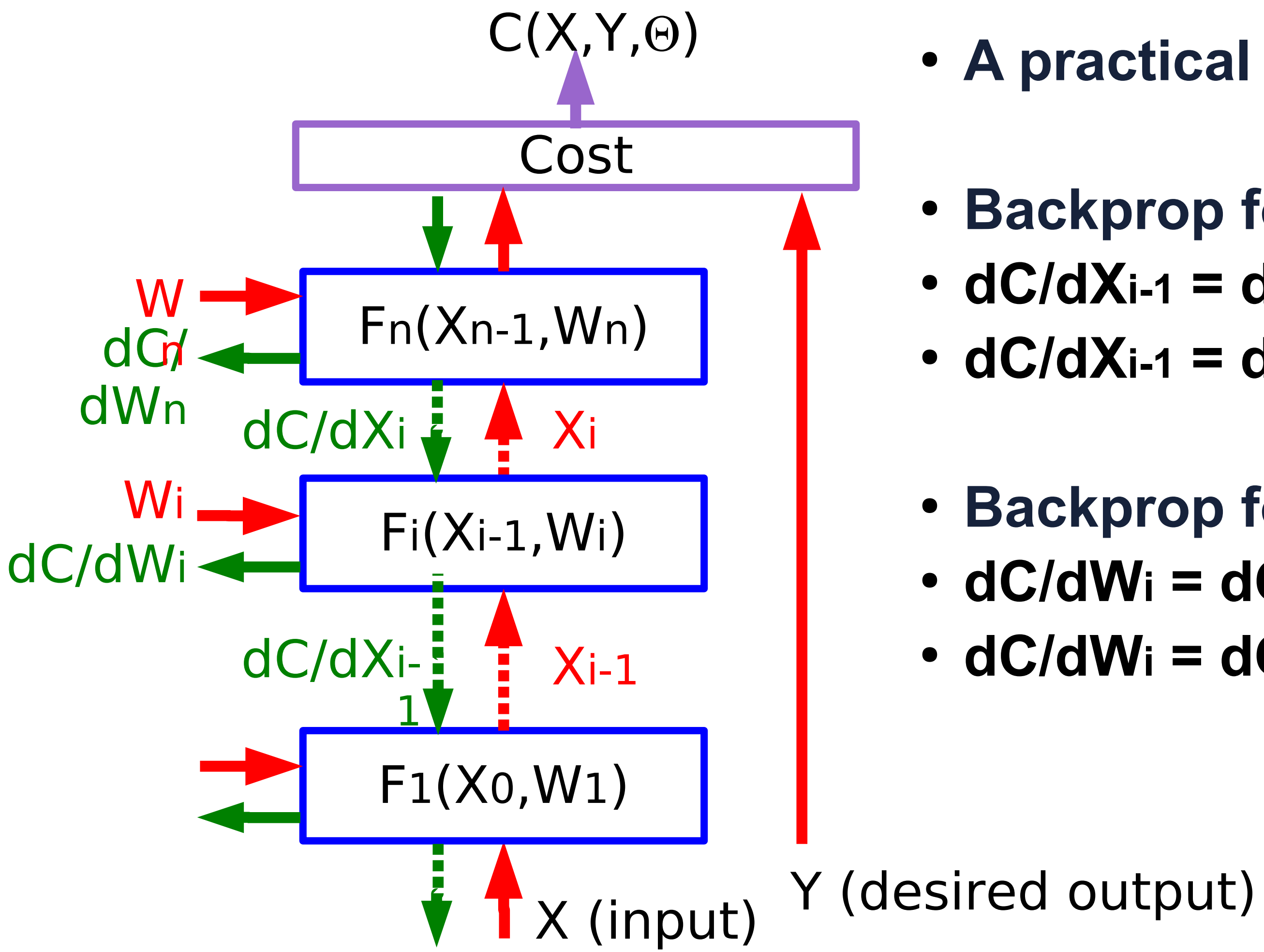


- **It's like walking in the mountains in a fog and following the direction of steepest descent to reach the village in the valley**
- **But each sample gives us a noisy estimate of the direction. So our path is a bit random.**

$$W_i \leftarrow W_i - \eta \frac{\partial L(W,X)}{\partial W_i}$$

- **Stochastic Gradient Descent (SGD)**

Computing Gradients by Back-Propagation



- **A practical Application of Chain Rule**

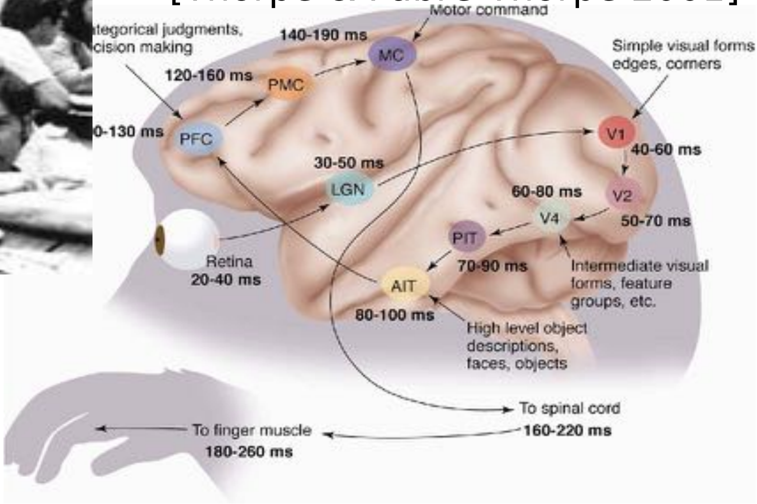
- **Backprop for the state gradients:**
- **$dC/dX_{i-1} = dC/dX_i \,.\, dX_i/dX_{i-1}$**
- **$dC/dX_{i-1} = dC/dX_i \,.\, dF_i(X_{i-1},W_i)/dX_{i-1}$**

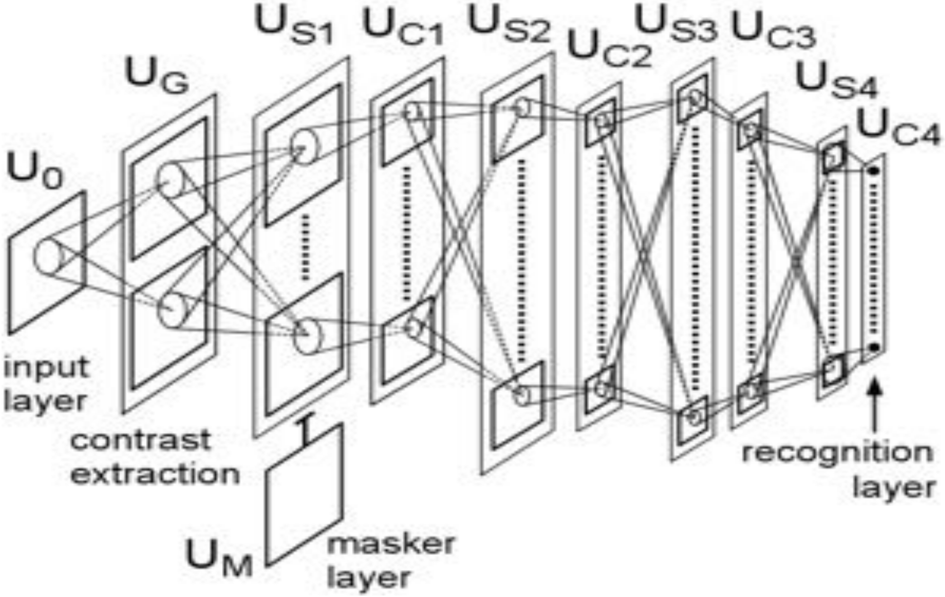
- **Backprop for the weight gradients:**
- **$dC/dW_i = dC/dX_i \,.\, dX_i/dW_i$**
- **$dC/dW_i = dC/dX_i \,.\, dF_i(X_{i-1},W_i)/dW_i$**

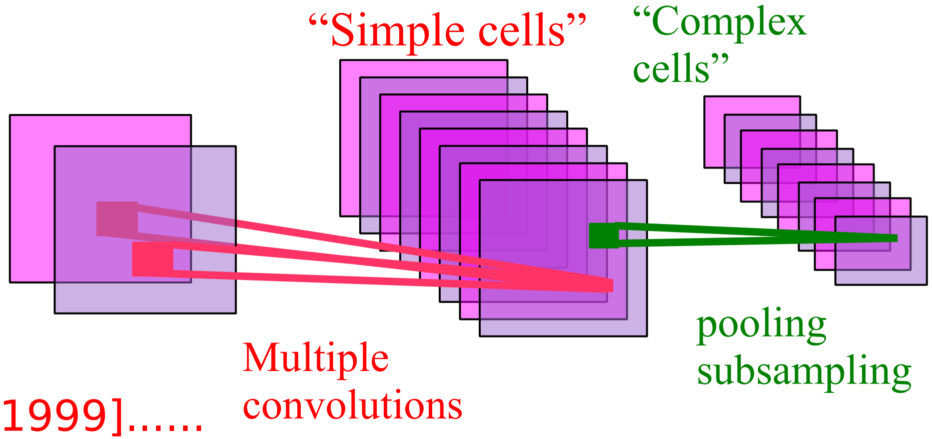
Hubel & Wiesel's Model of the Architecture of the Visual Cortex

- **[Hubel & Wiesel 1962]:**
  - simple cells detect local features
  - complex cells "pool" the outputs of simple cells within a

[Thorpe & Fabre-Thorpe 2001]

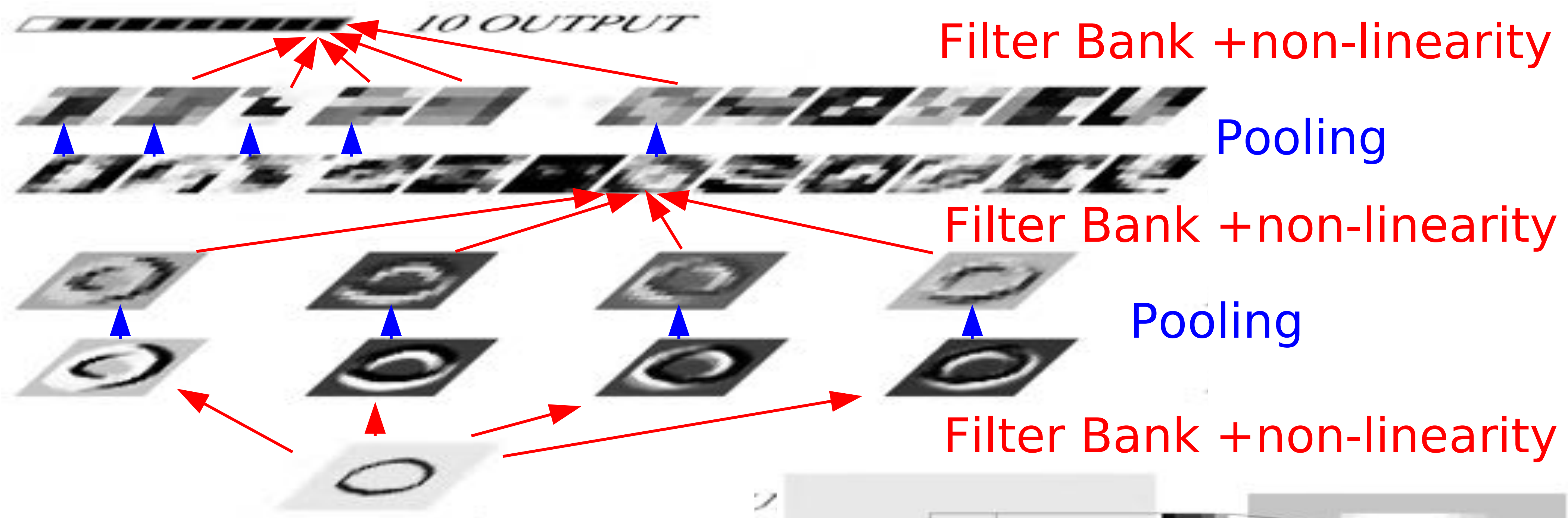




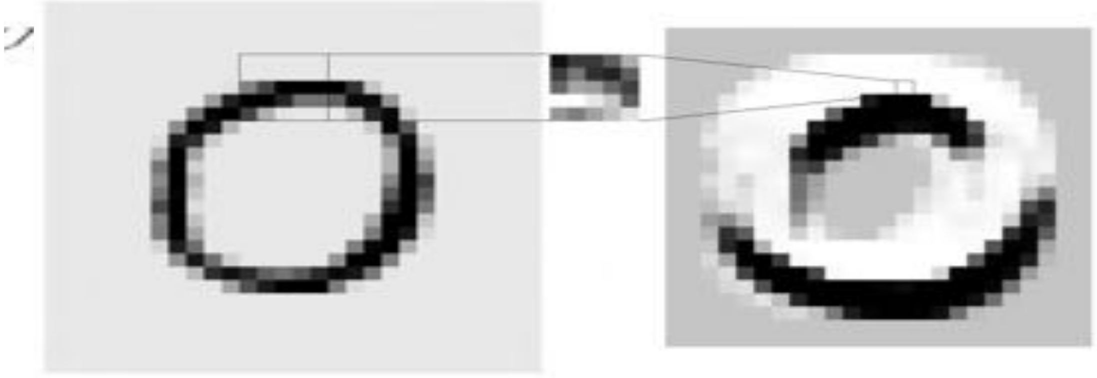


[Fukushima 1982][LeCun 1989, 1998],[Riesenhuber 1999]......

Convolutional Network Architecture [LeCun et al. NIPS 1989]



- **Inspired by [Hubel & Wiesel 1962] & [Fukushima 1982] (Neocognitron):**
  - simple cells detect local features
  - complex cells "pool" the outputs of simple cells within a retinotopic neighborhood.

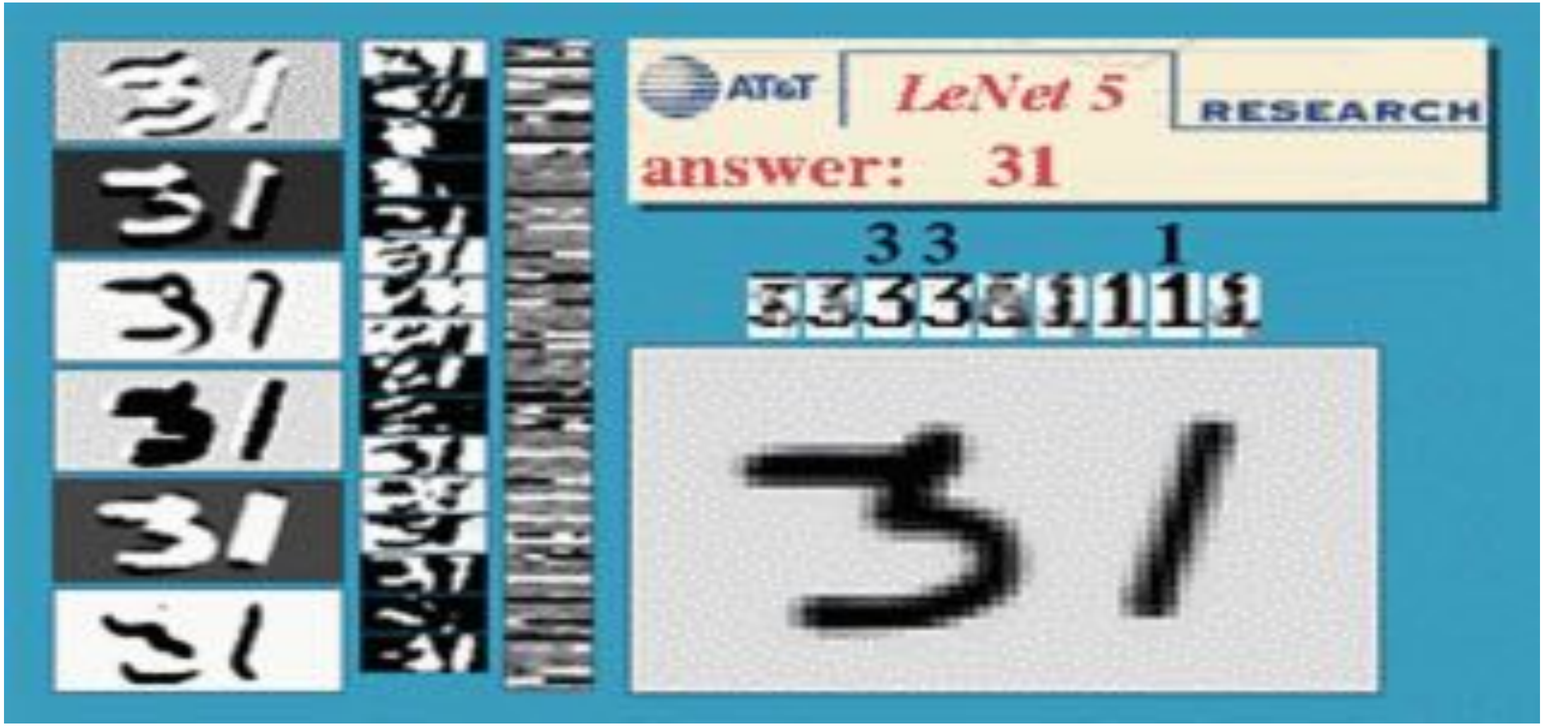


Convolutional Network (LeNet5, vintage 1990)

- Filters-tanh → pooling → filters-tanh → pooling → filters-tanh

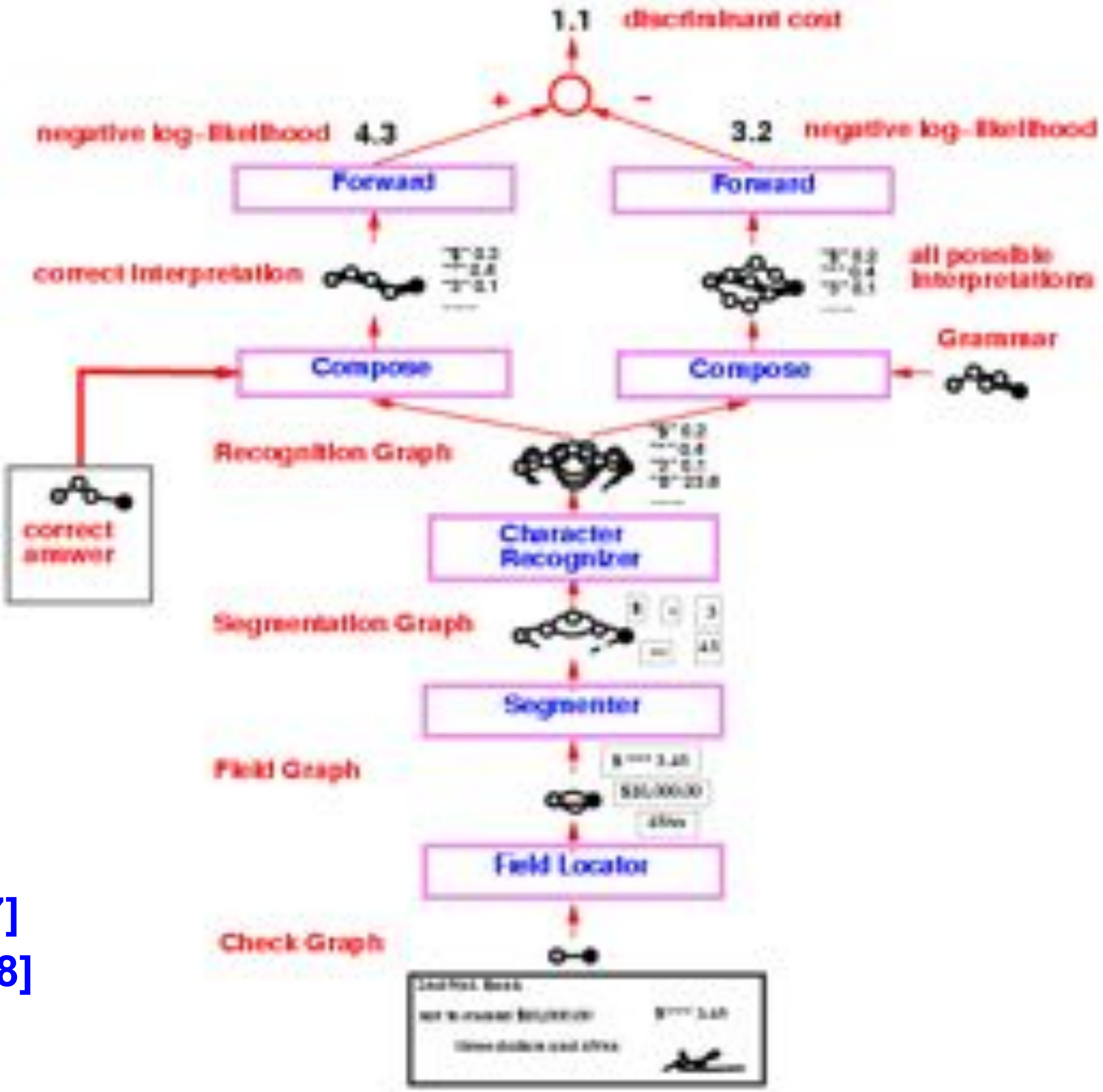
ConvNets can recognize multiple objects

- All layers are convolutional
- Networks performs simultaneous segmentation and recognition



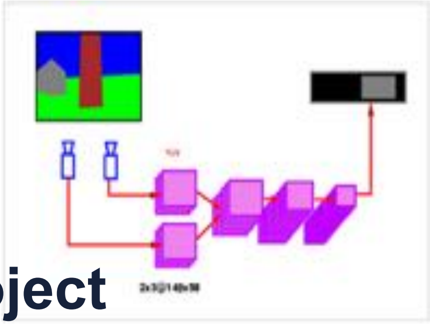
Check Reader (AT&T 1995)

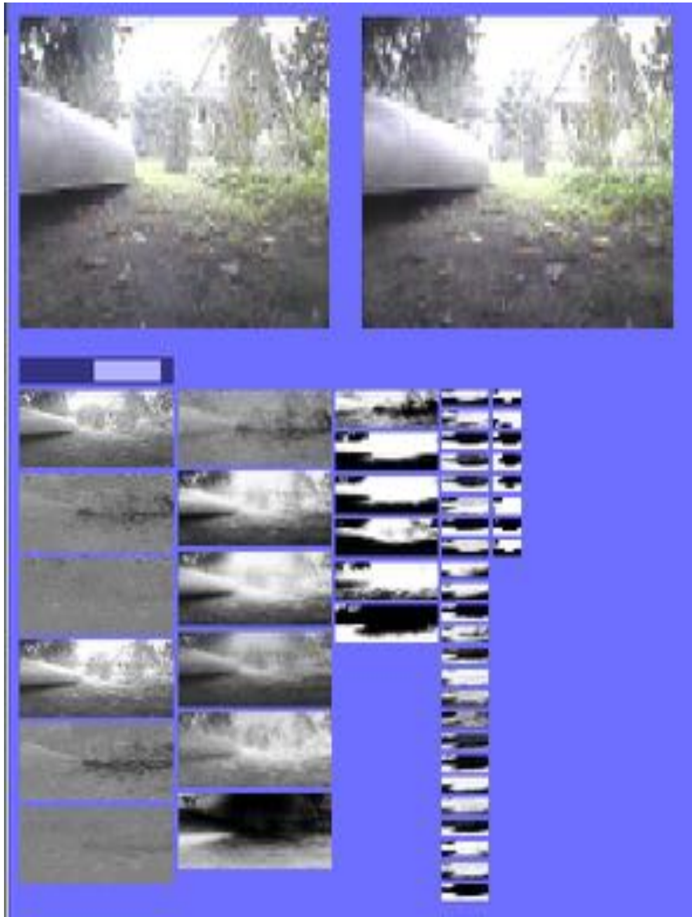
- Graph transformer network trained to read **check amounts.**
- Trained globally with Negative-Log-Likelihood loss (MMI).
- 50% percent correct, 49% reject, 1% error (detectable later in the process).
- **Fielded in 1996**, used in many banks in the US and Europe.
- Processed an estimated **10% to 20% of all the checks written in the US in the early 2000s.**
- [LeCun, Bottou, Bengio ICASSP1997] [LeCun, Bottou, Bengio, Haffner 1998]



DAVE: obstacle avoidance through imitation learning

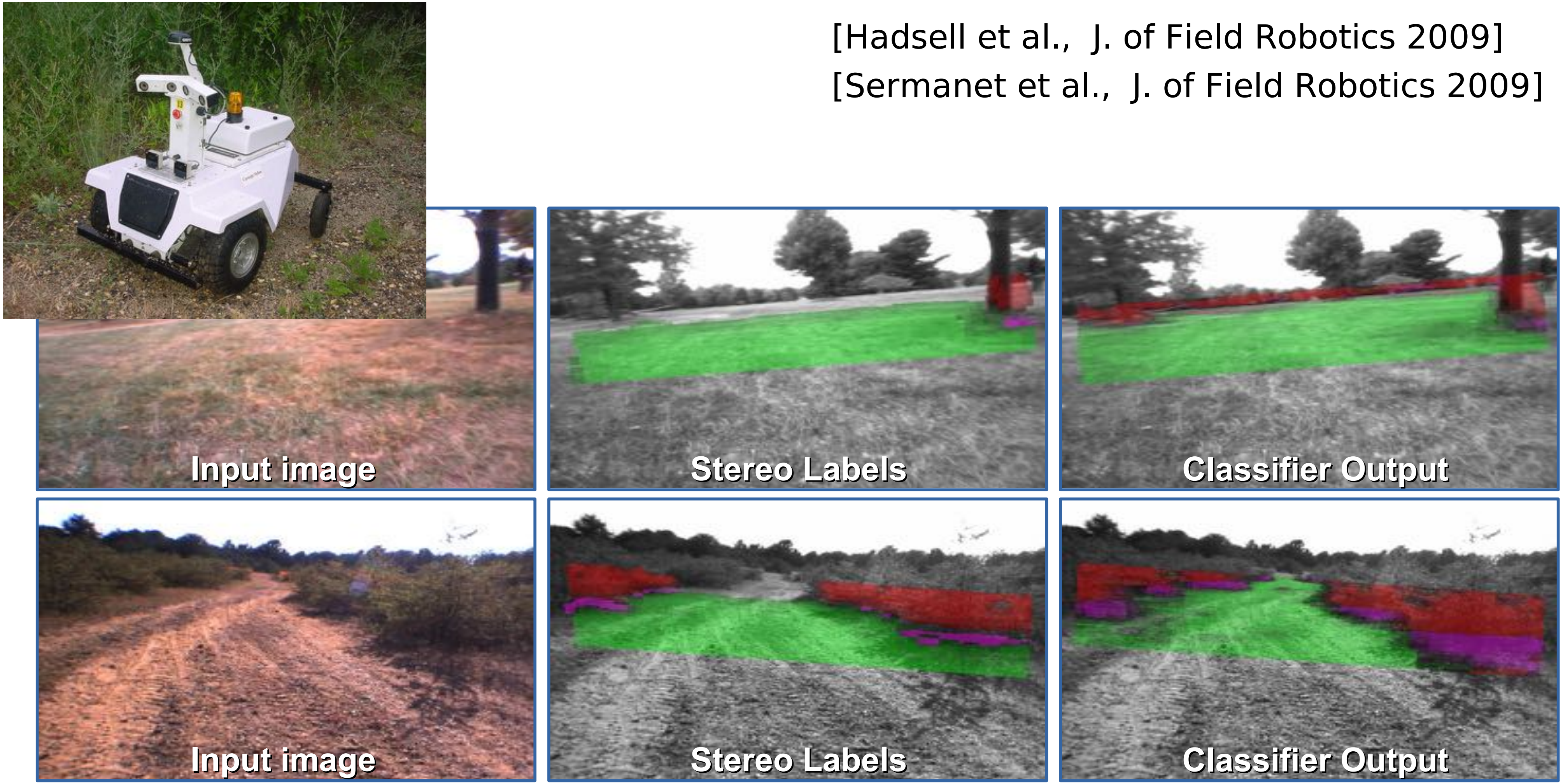
- **Fall 2003 project at Net-Scale Technologies (Urs Muller)**
  - [LeCun et al. NIPS 2005] (rejected from RSS 2005).
- **Human driver data**
- **Image →[convnet]→steering**
- **20 minutes of training data**
- **Motivated the DARPA LAGR project**



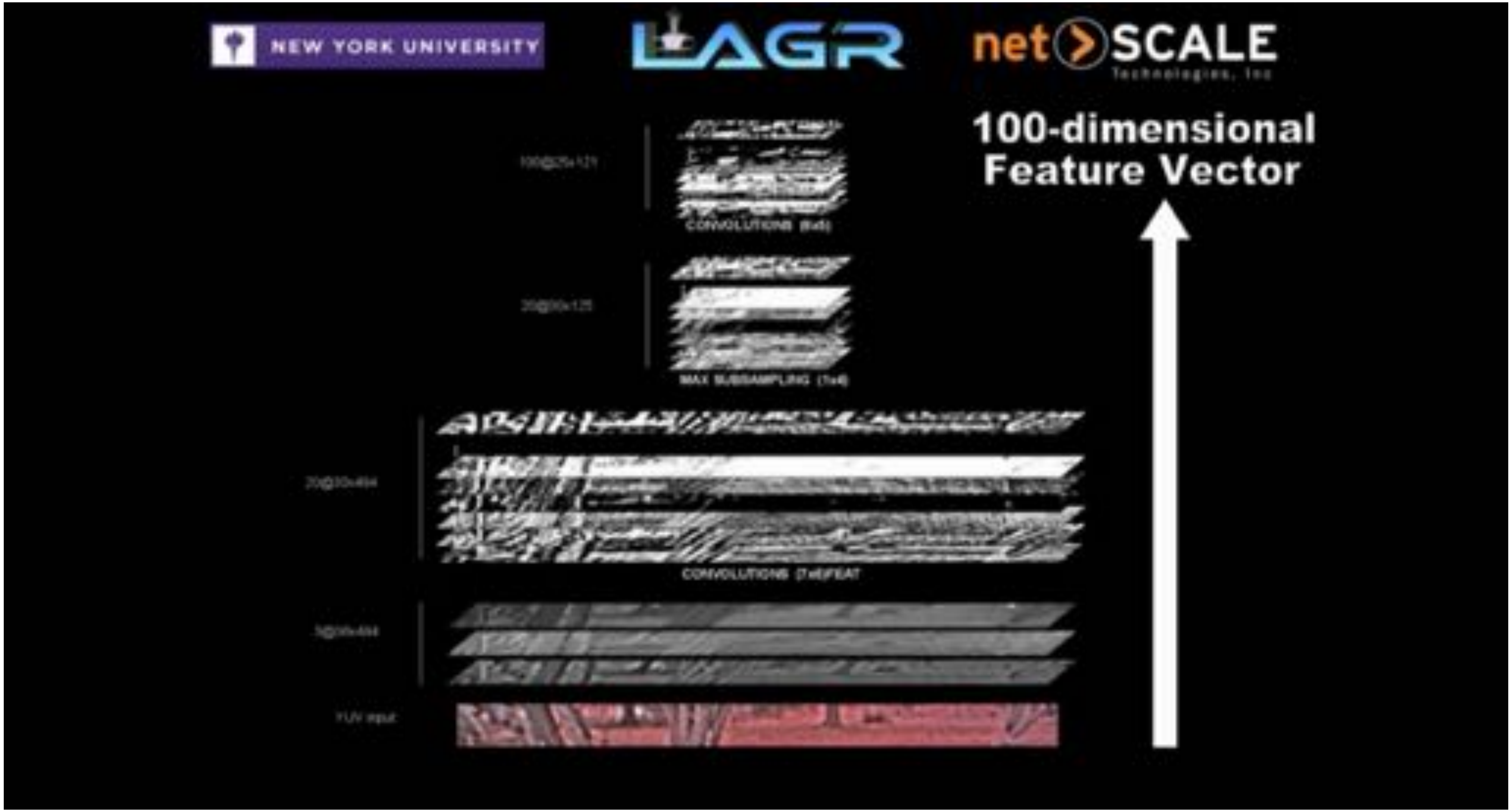


Semantic Segmentation with ConvNet for off-Road Driving

[Hadsell et al., J. of Field Robotics 2009]
[Sermanet et al., J. of Field Robotics 2009]



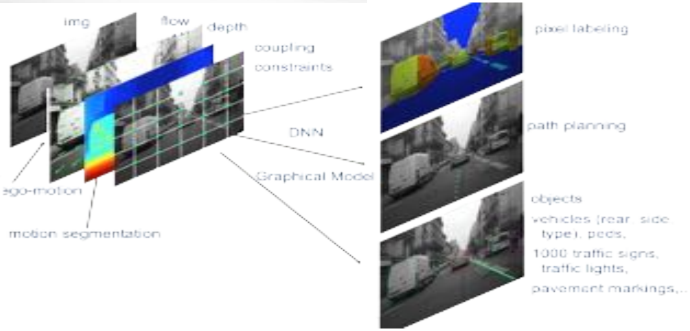
LAGR Video


NEW YORK UNIVERSITY
LAGR
net SCALE
100-dimensional
Feature Vector
CONVOLUTIONS
MAX SUBSAMPLING
CONVOLUTIONS

Semantic Segmentation with ConvNets (33 categories)

Driving Cars with Convolutional Nets

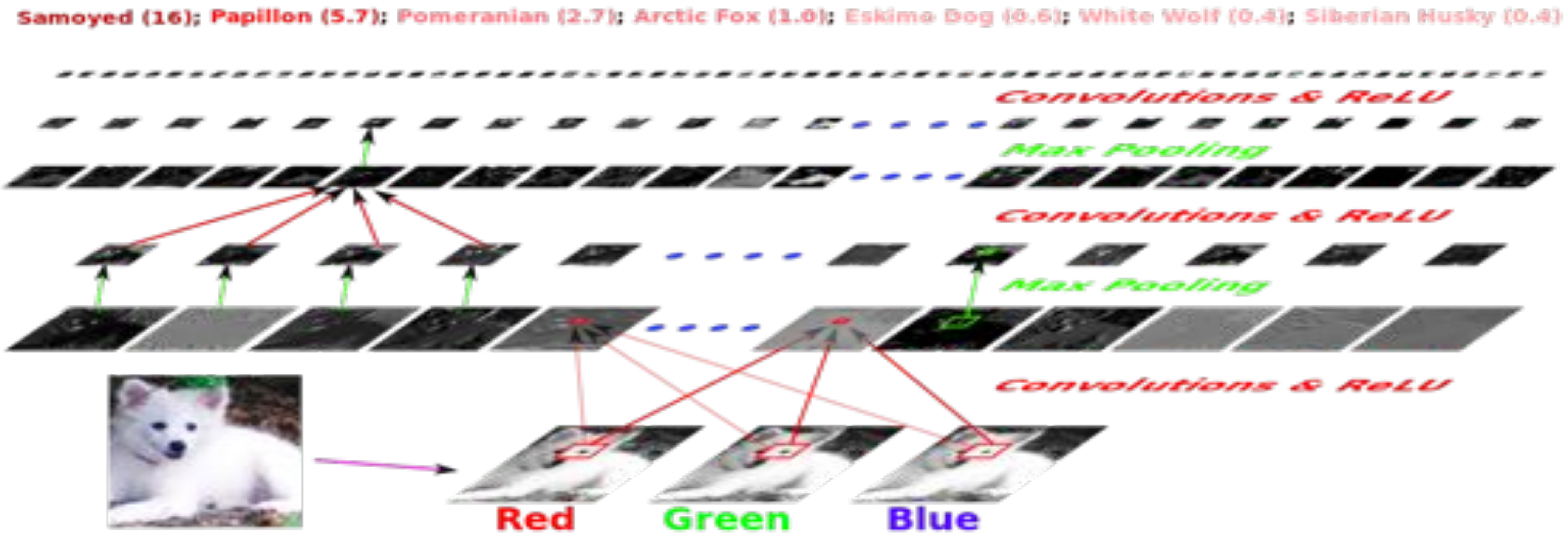
- **MobilEye**



- **NVIDIA**

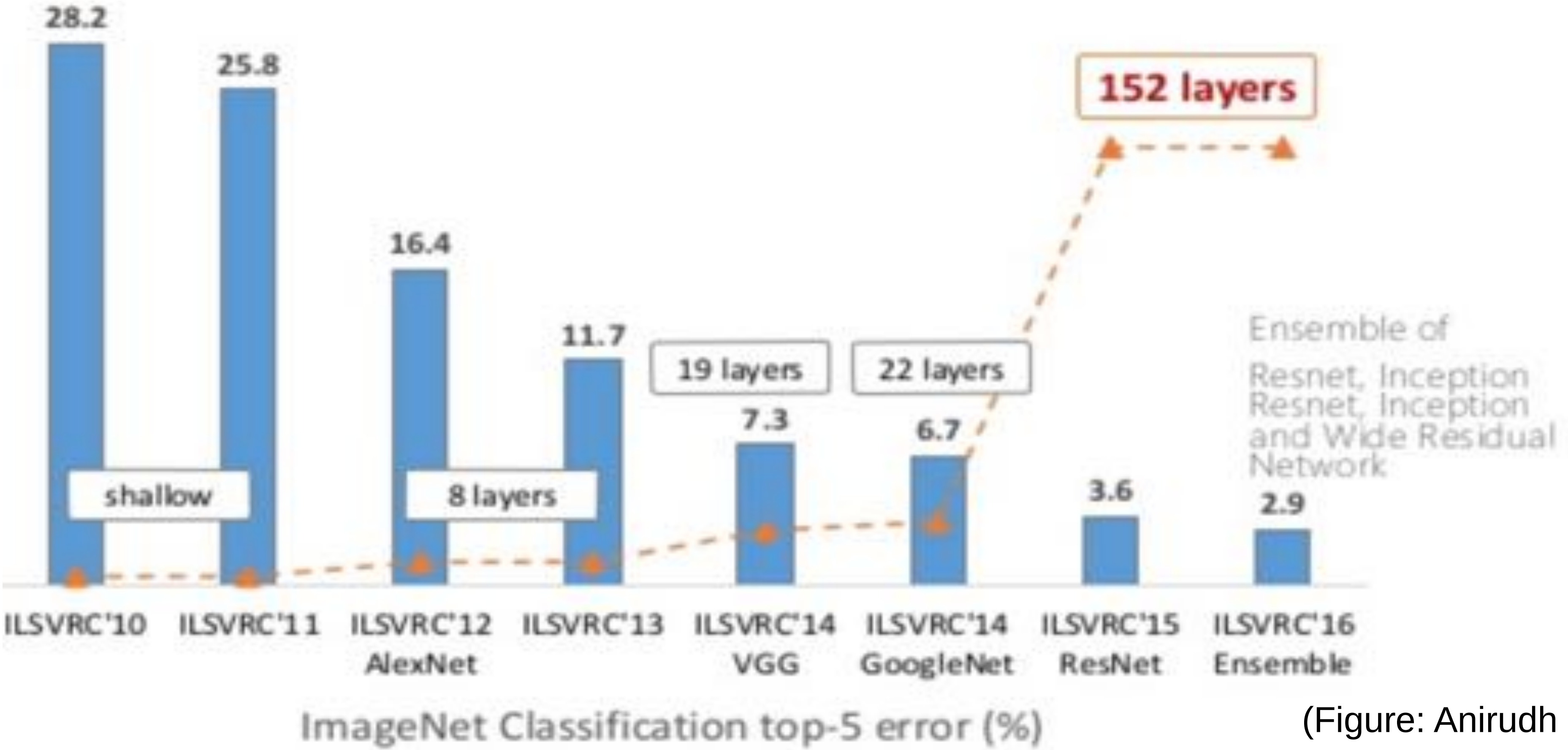
Deep Convolutional Nets for Object Recognition

- AlexNet [Krizhevsky et al. NIPS 2012], OverFeat [Sermanet et al. 2013]
- 1 to 10 billion connections, 10 million to 1 billion parameters, 8 to 20 layers.



Error Rate on ImageNet

- **Depth inflation**

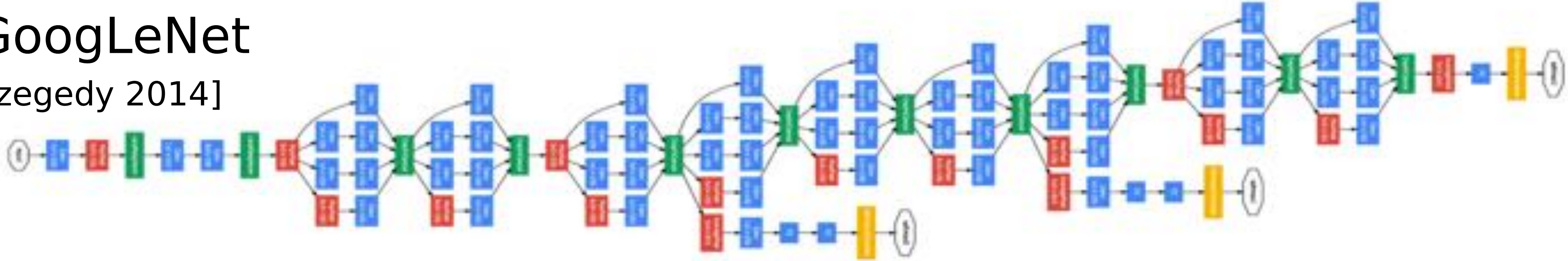


(Figure: Anirudh Koul)

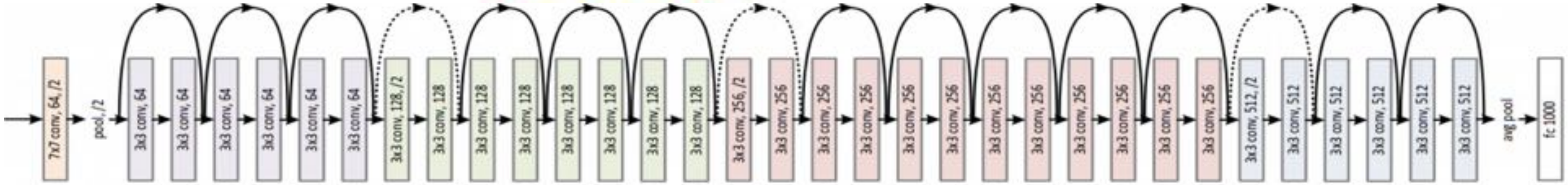
Deep ConvNets (depth inflation)

VGG
[Simonyan 2013]

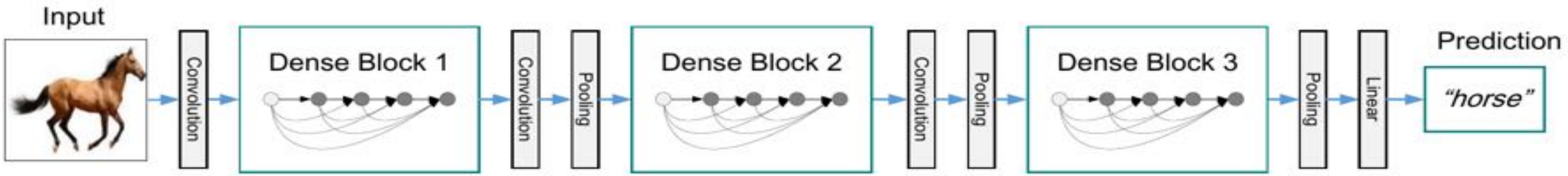
GoogLeNet
Szegedy 2014]



ResNet
[He et al. 2015]

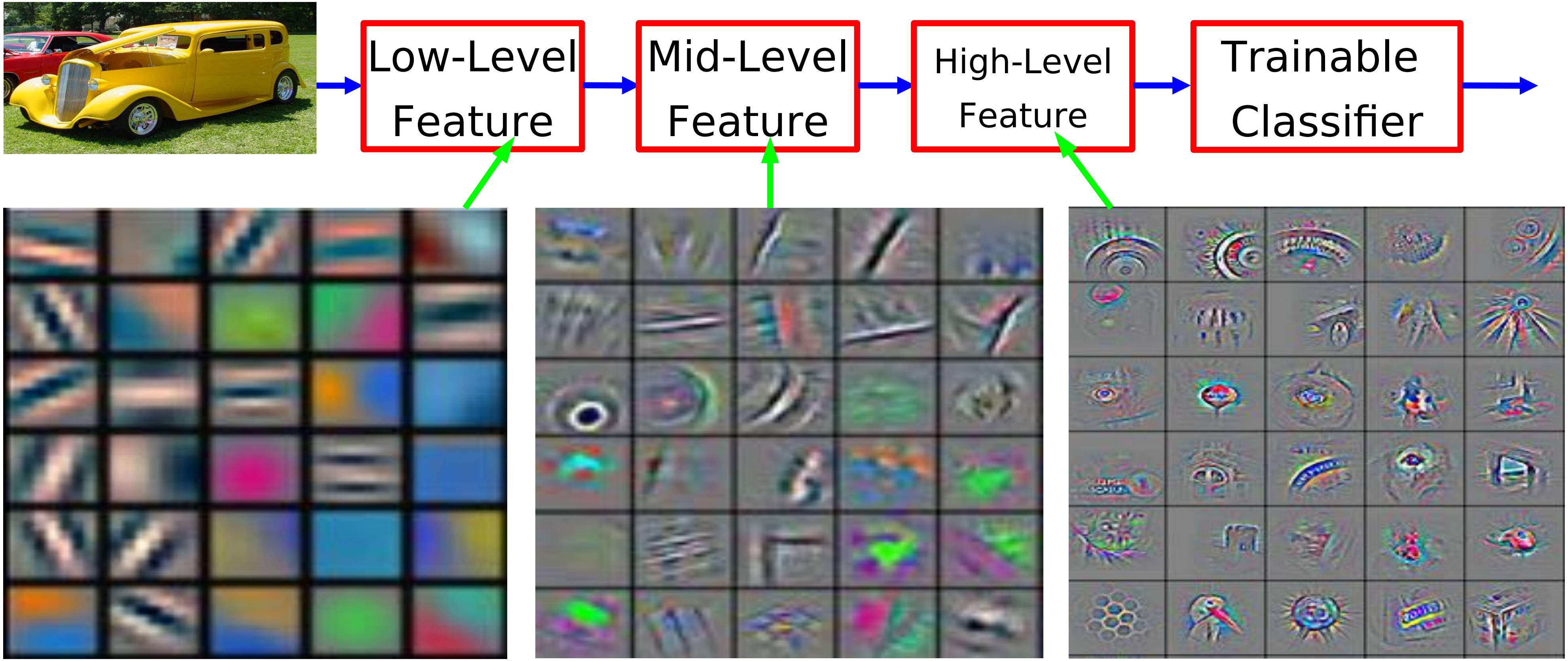


DenseNet
[Huang et al 2017]



Multilayer Architectures == Compositional Structure of Data

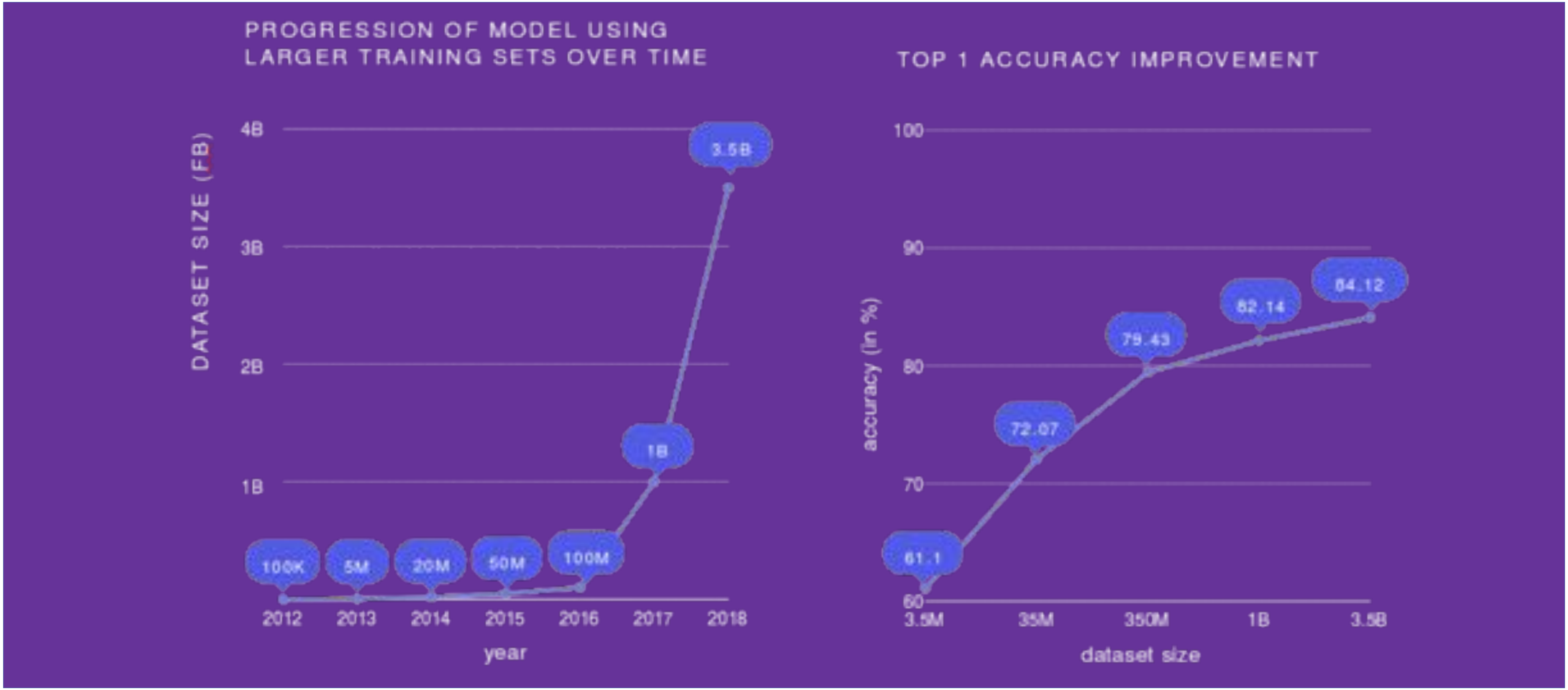
- **Natural is data is compositional => it is efficiently representable hierarchically**



Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

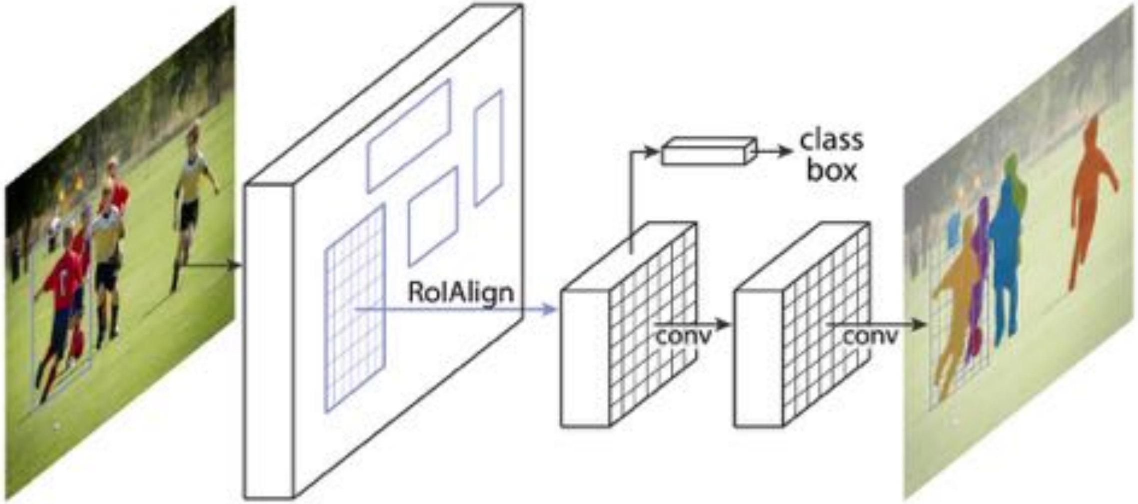
Learning from hash tags on 3.5 billion images

- **Pretraining on 3.5b instagram images with hashtags. Training/test on ImageNet**



Mask R-CNN: instance segmentation

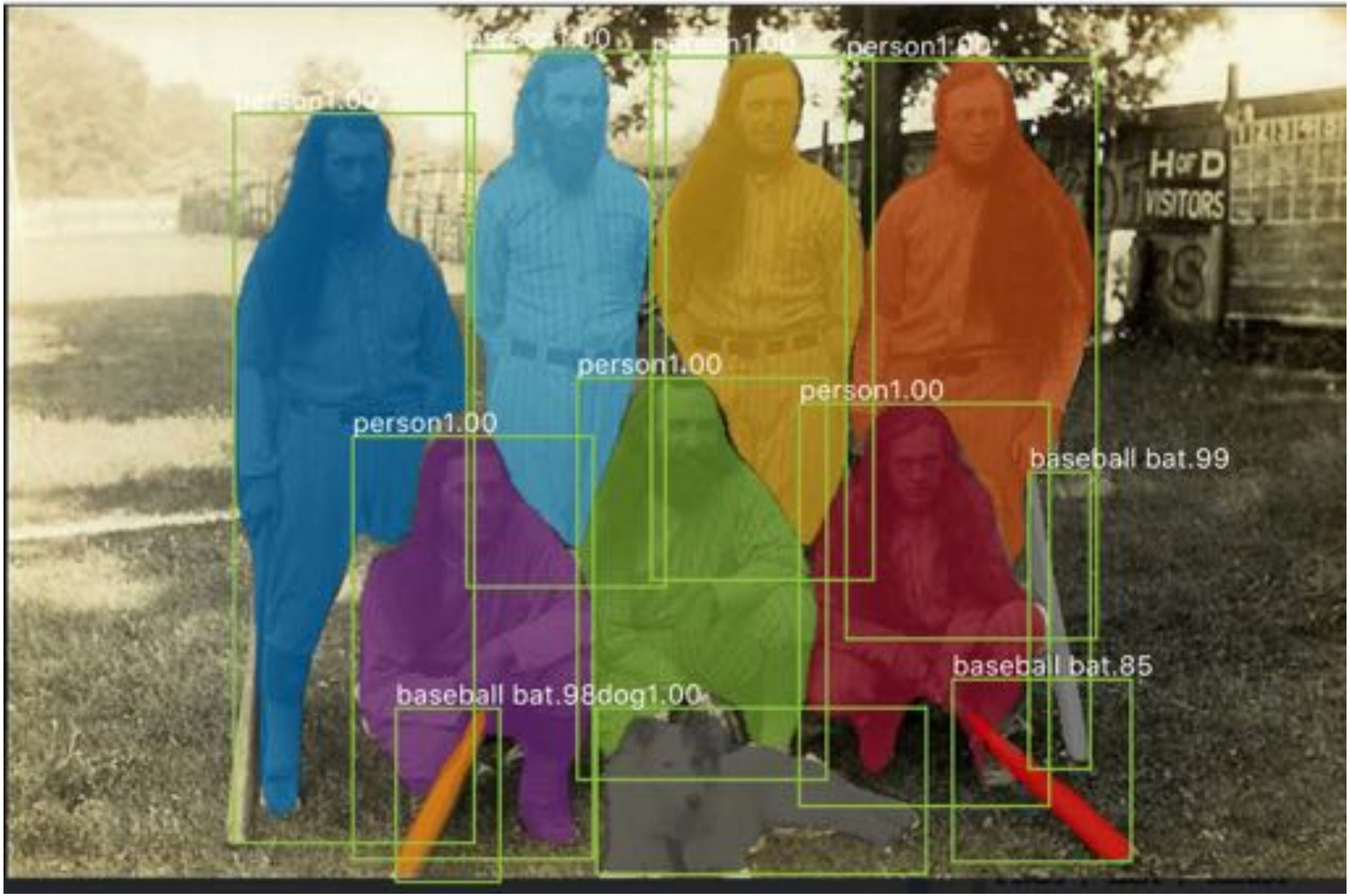
- [He, Gkioxari, Dollar, Girshick arXiv:1703.06870]
- ConvNet produces an object mask for each region of interest
- Combined ventral and dorsal pathways



	backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
MNC [7]	ResNet-101-C4	24.6	44.3	24.8	4.7	25.9	43.6
FCIS [20] +OHEM	ResNet-101-C5-dilated	29.2	49.5	-	7.1	31.3	50.0
FCIS+++ [20] +OHEM	ResNet-101-C5-dilated	33.6	54.5	-	-	-	-
Mask R-CNN	ResNet-101-C4	33.1	54.9	34.8	12.1	35.6	51.1
Mask R-CNN	ResNet-101-FPN	35.7	58.0	37.8	15.5	38.1	52.4
Mask R-CNN	ResNeXt-101-FPN	**37.1**	**60.0**	**39.4**	**16.9**	**39.9**	**53.5**

Mask-RCNN Results on COCO dataset

- **Individual objects are segmented.**



Mask-RCNN Results on COCO dataset

- **Individual objects are segmented.**

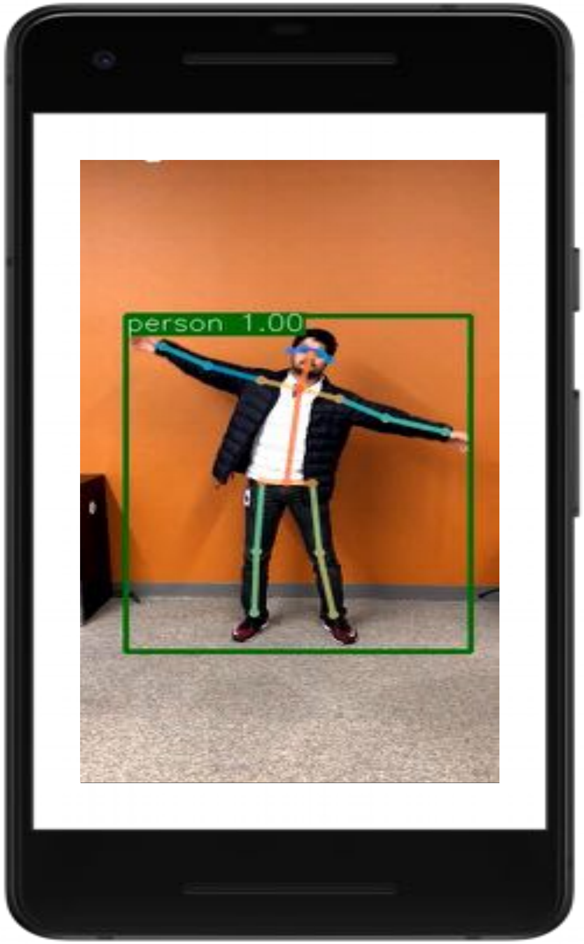
Mask R-CNN Results on COCO test set

Mask R-CNN Results on COCO test set

Figure 4. More results of **Mask R-CNN** on COCO test images, using ResNet-101-FPN and running at 5 fps, with 35.7 mask AP (Table 1).

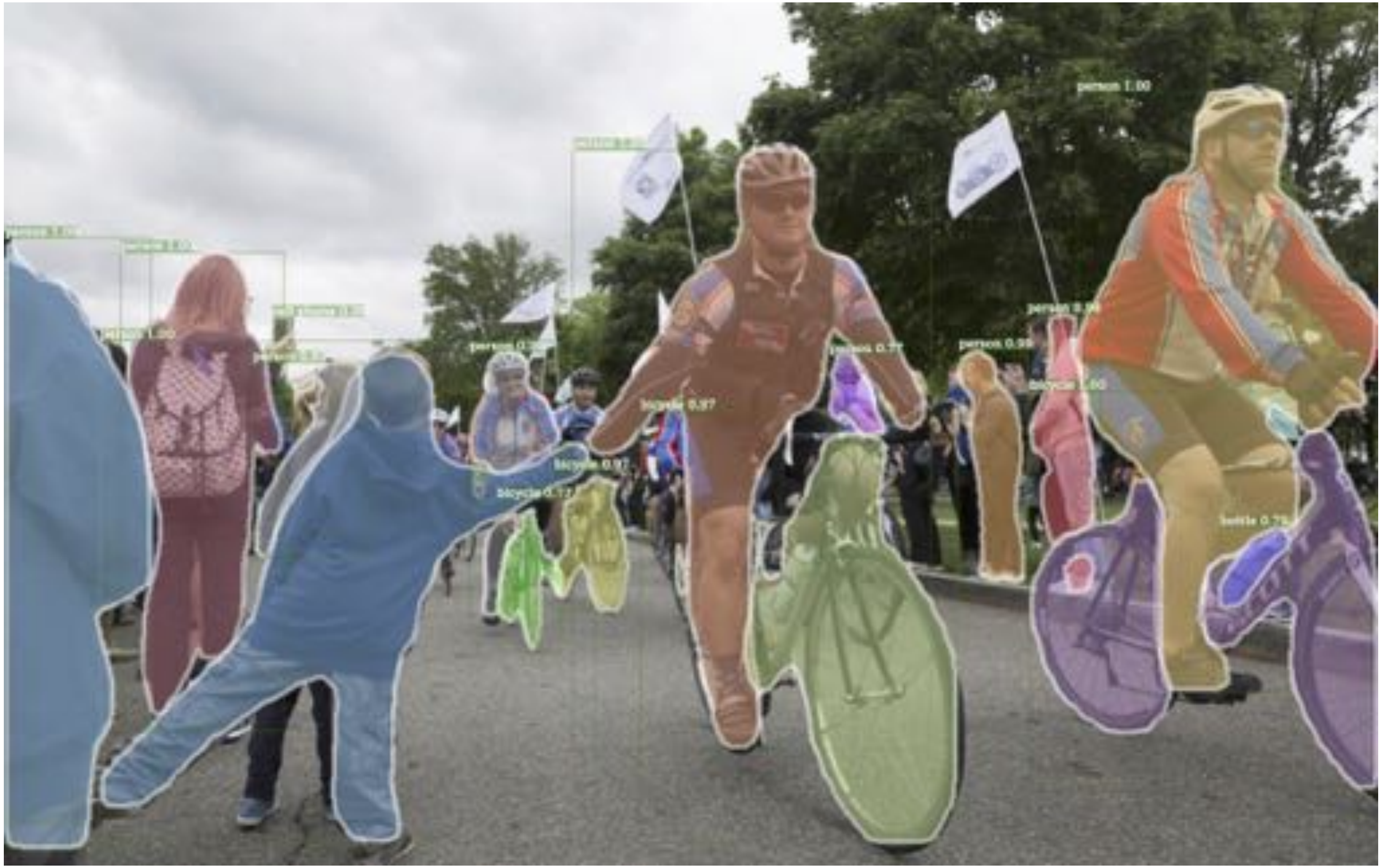
Real-Time Pose Estimation on Mobile Devices

- **Maks R-CNN running on Caffe2Go**



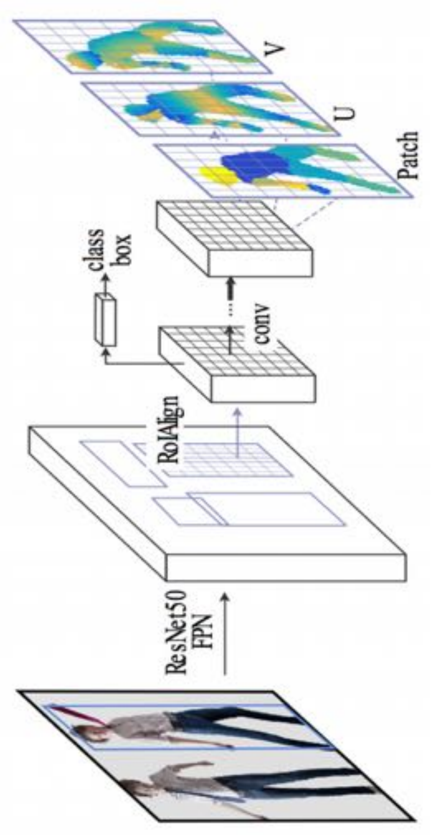
Detectron: open source vision

https://github.com/facebookresearch/Detectron



DensePose: real-time body pose estimation

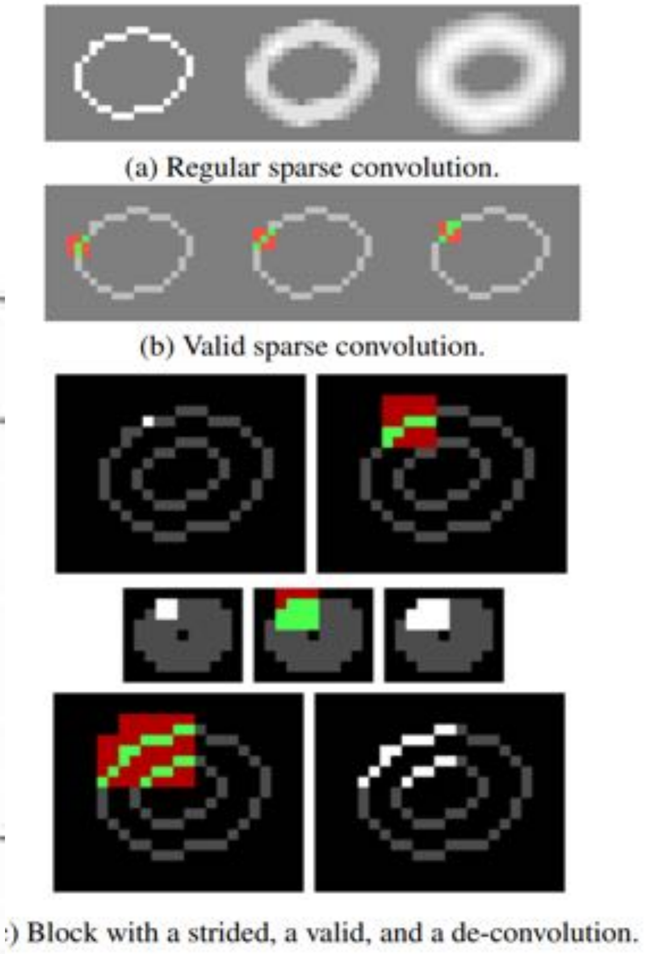
- [Guler, Neverova, Kokkinos CVPR 2018] http://densepose.org
  - 20 fps on a single GPU



3D Semantic Segmentation with Sparse ConvNets

- **ShapeNet competition results ArXiv:1710.06104]**
- **Winner: Submanifold Sparse ConvNet**
  - [Graham & van der Maaten arXiv 1706.01307]
  - PyTorch: https://github.com/facebookresearch/SparseConvNet

method	mean
SSCN	**86.00**
PdNet	85.49
DCPN	84.32
PCNN	82.29
PtAdLoss	77.96
KDTNet	65.80
DeepPool	42.79
NN	77.57
[19]	84.74



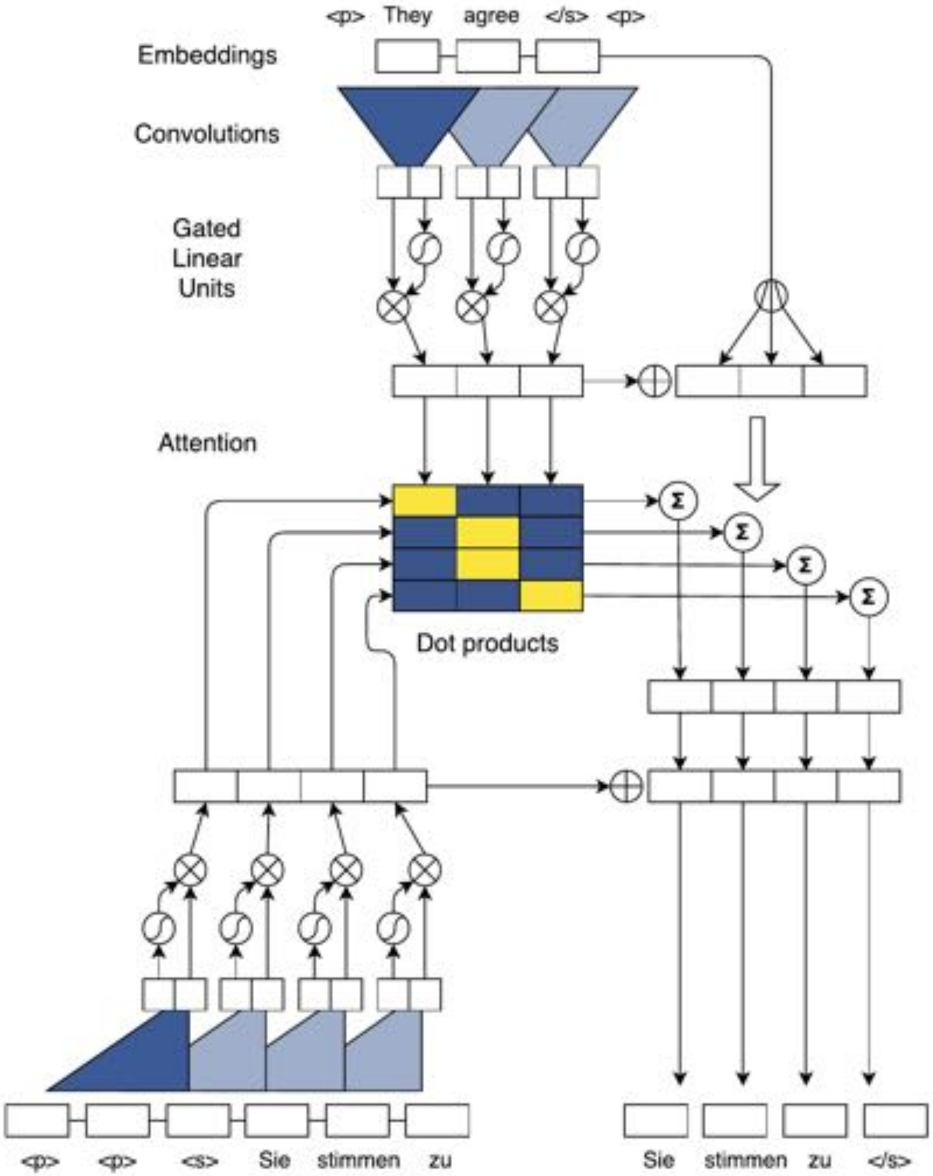
FairSeq for Translation

- **[Gehring et al. ArXiv:1705.03122]**

WMT'16 English-Romanian	BLEU
Sennrich et al. (2016b) GRU (BPE 90K)	28.1
ConvS2S (Word 80K)	29.45
ConvS2S (BPE 40K)	29.88

WMT'14 English-German	BLEU
Luong et al. (2015) LSTM (Word 50K)	20.9
Kalchbrenner et al. (2016) ByteNet (Char)	23.75
Wu et al. (2016) GNMT (Word 80K)	23.12
Wu et al. (2016) GNMT (Word pieces)	24.61
ConvS2S (BPE 40K)	25.16

WMT'14 English-French	BLEU
Wu et al. (2016) GNMT (Word 80K)	37.90
Wu et al. (2016) GNMT (Word pieces)	38.95
Wu et al. (2016) GNMT (Word pieces) + RL	39.92
ConvS2S (BPE 40K)	40.46



Applications of ConvNets

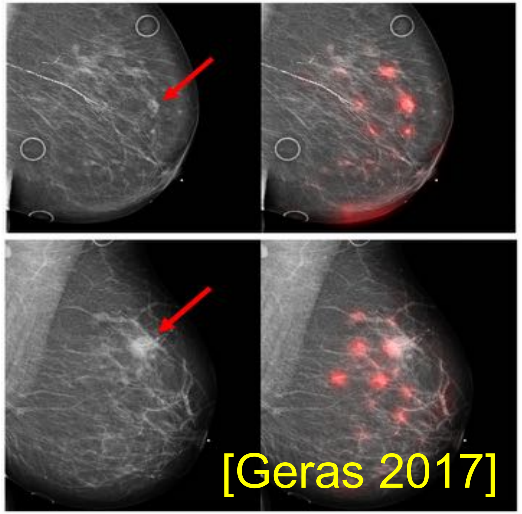
- **Self-driving cars, visual perception**
- **Medical signal and image analysis**
  - Radiology, dermatology, EEG/seizure prediction….
- **Bioinformatics/genomics**
- **Speech recognition**
- **Language translation**
- **Image restoration/manipulation/style transfer**
- **Robotics, manipulation**
- **Physics**
  - High-energy physics, astrophysics
- **New applications appear every day**
  - E.g. environmental protection,….

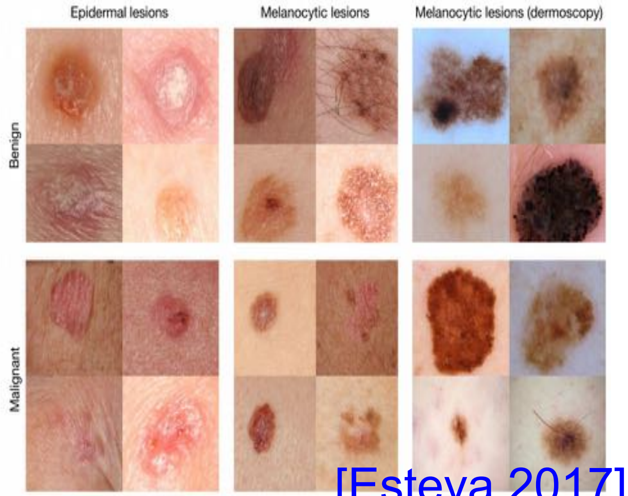
Applications of Deep Learning

- **Medical image analysis**
- **Self-driving cars**
- **Accessibility**
- **Face recognition**
- **Language translation**
- **Virtual assistants***
- **Content Understanding for:**
  - Filtering
  - Selection/ranking
  - Search
- **Games**
- **Security, anomaly detection**
- **Diagnosis, prediction**
- **Science!**


[Mnih 2015]

[MobilEye]

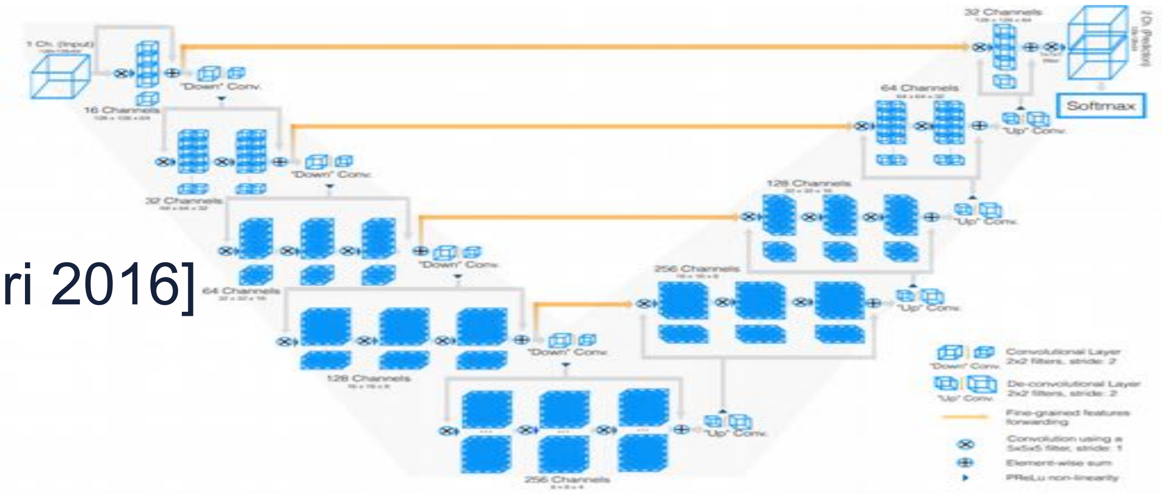

[Geras 2017]



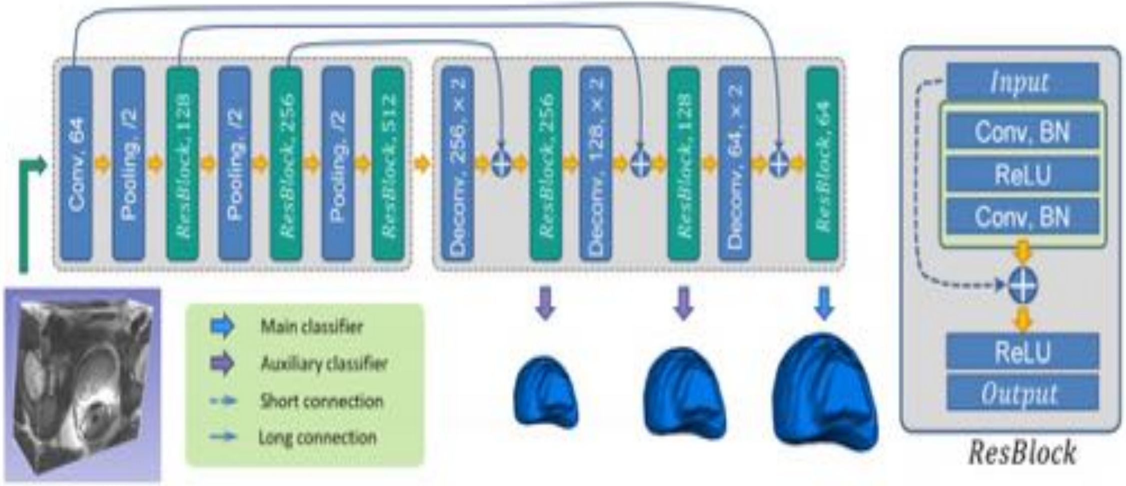
[Esteva 2017]

3D ConvNets for Prostate Segmentation in MRI

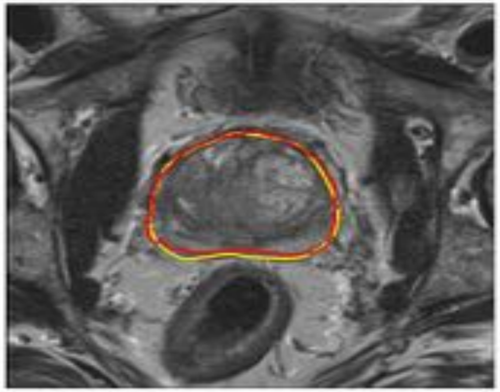
- **V-Net**
  - [Milletari 2016]

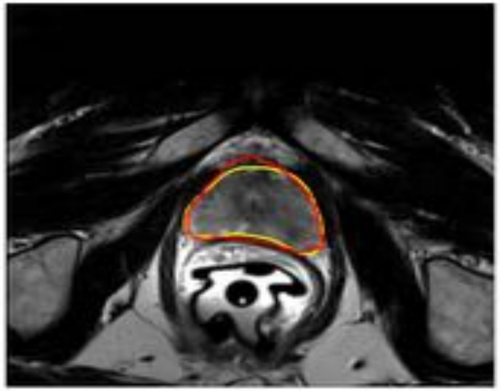


- **CUMED**
  - [Yu 2017]



- **PROMISE12 dataset**



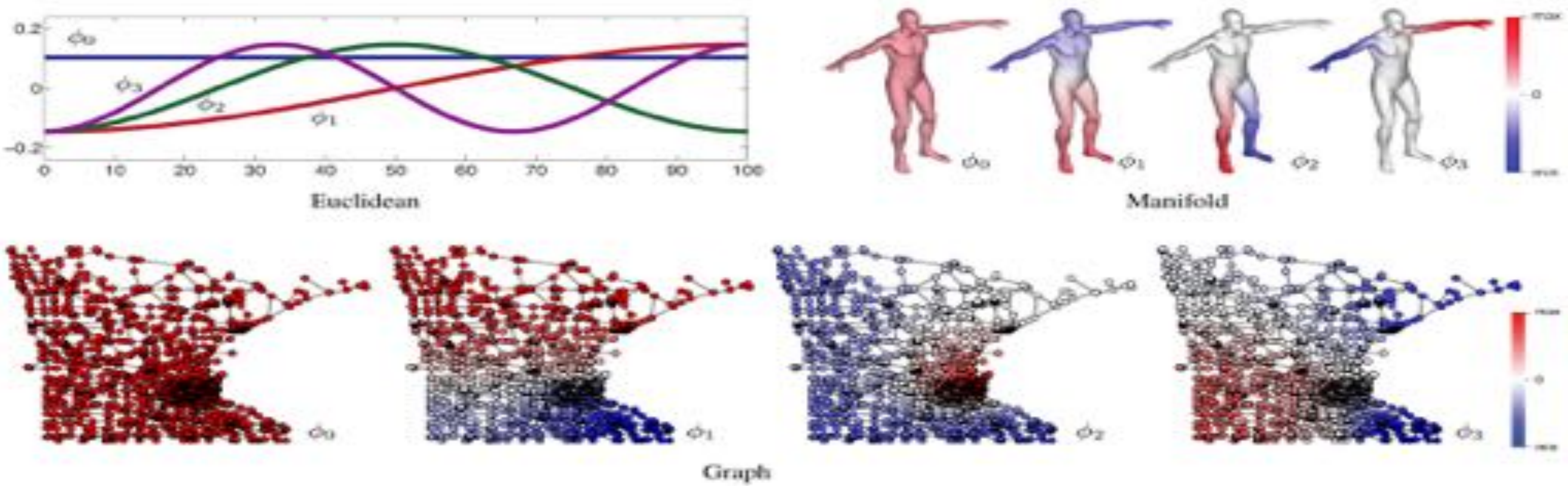


NVIDIA Autonomous Driving Demo

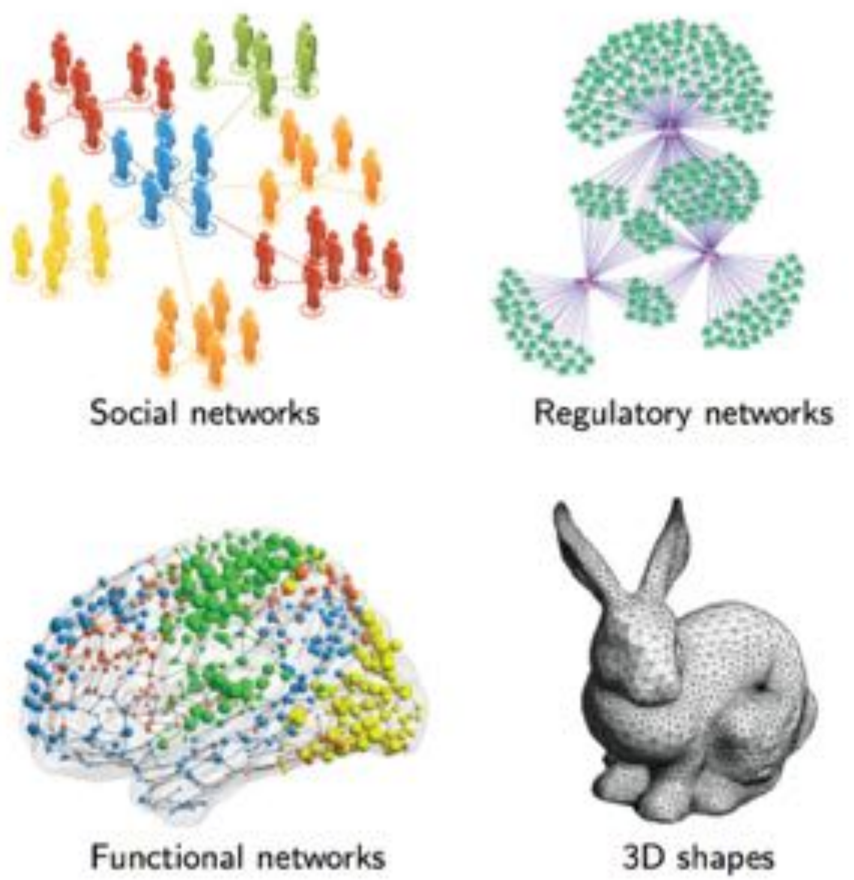
- In bucolic New Jersey

Spectral Networks: Convolutional Nets on Irregular Graphs

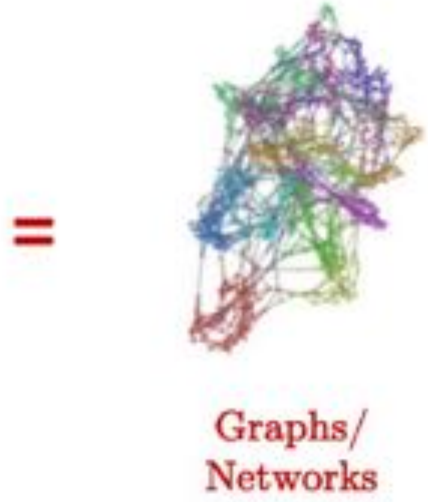
- Convolutions are diagonal operators in Fourier space
- The Fourier space is the eigenspace of the Laplacian
- We can compute graph Laplacians
- Review paper: [Bronstein et al. 2016, ArXiv:1611.08097]



ConvNets on Graphs (fixed and data-dependent)



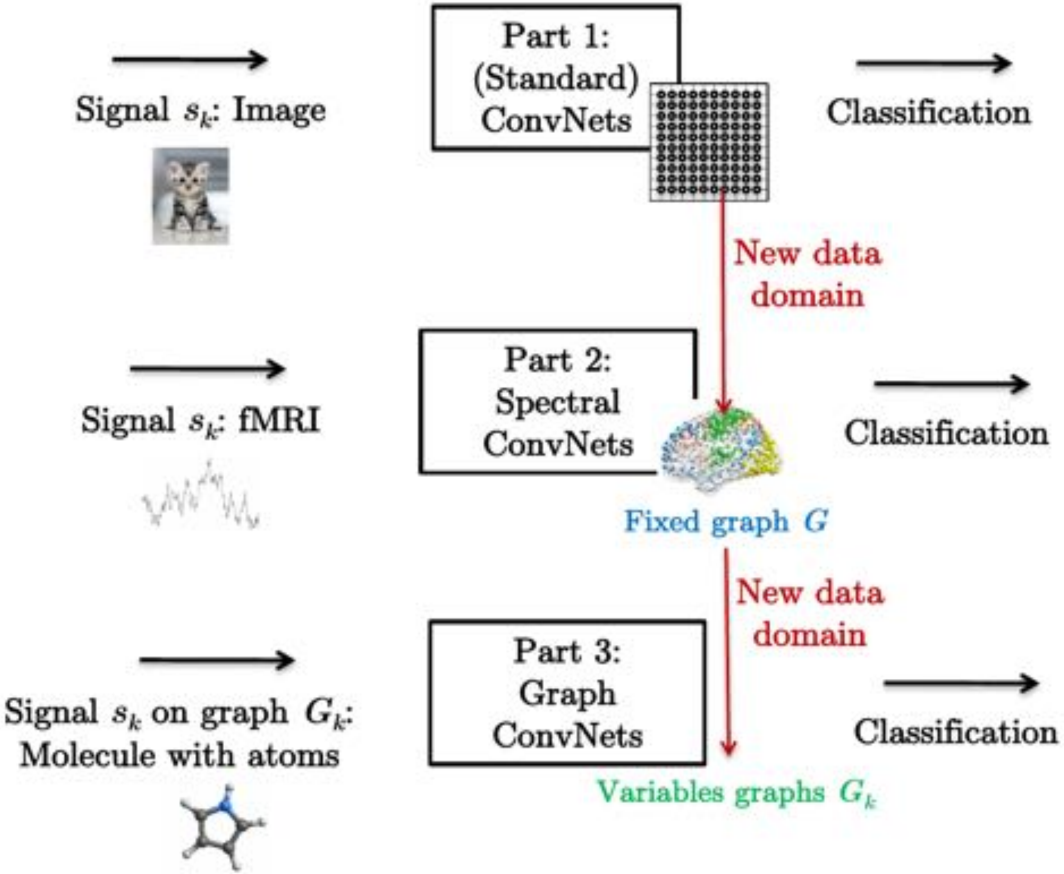
- **Graphs can represent: Natural language, social networks, chemistry, physics, communication networks...**



- **[Bresson 2018]**

Spectral ConvNets / Graph ConvNets

- **Regular grid graph**
  - Standard ConvNet
- **Fixed irregular graph**
  - Spectral ConvNet
- **Dynamic irregular graph**
  - Graph ConvNet



[Bresson 2018]
IPAM workshop:
http://www.ipam.ucla.edu/programs/workshops/new-deep-learning-techniques/

What About (Deep) Reinforcement Learning?

It works great …

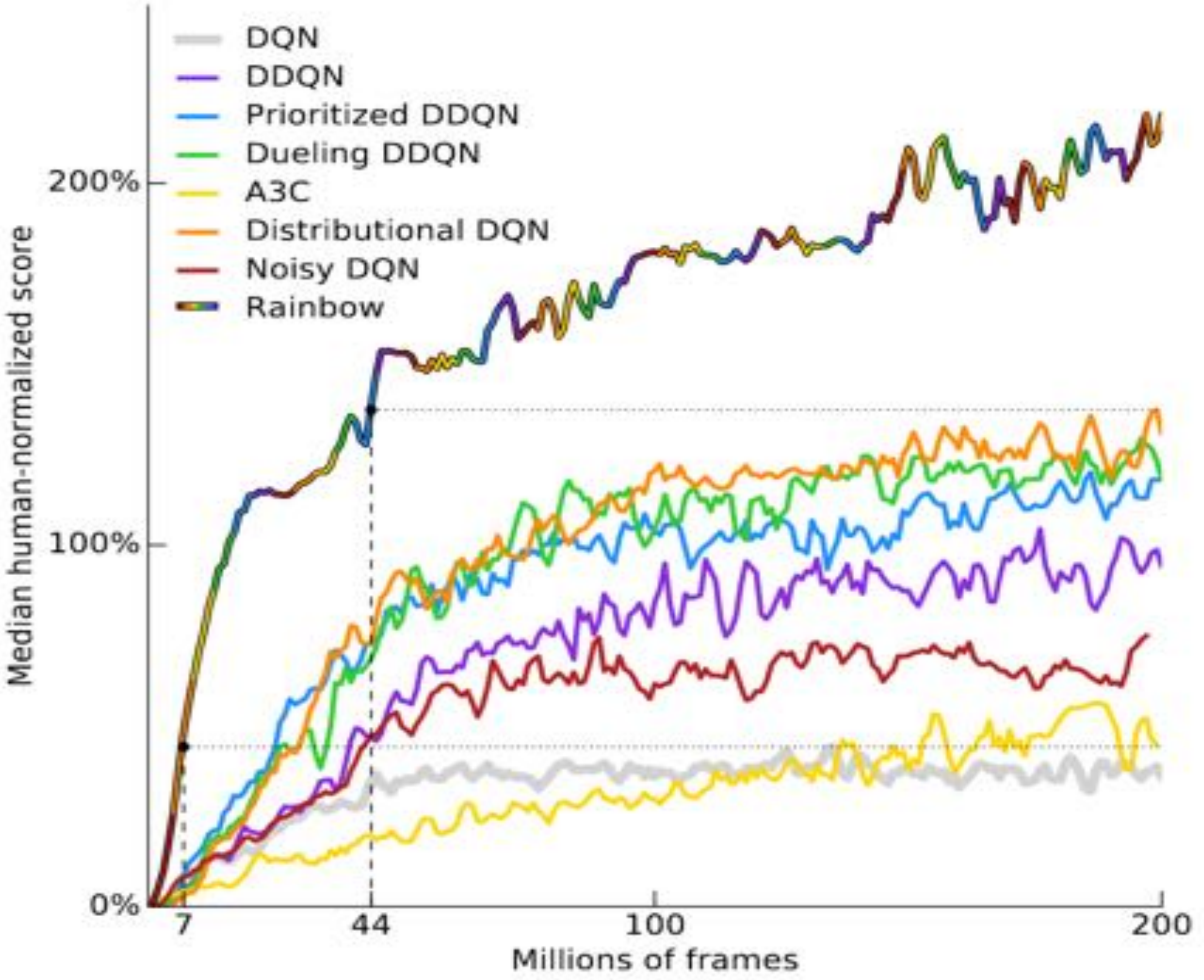
…for games and virtual environments

Reinforcement Learning works fine for games

- **RL works well for games**
  - Playing Atari games [Mnih 2013], Go [Silver 2016, Tian 2018], Doom [Tian 2017], StarCraft (work in progress at FAIR, DeepMind….)
  - RL requires too many trials.
  - RL often doesn’t really work in the real world

Pure RL requires many, many trials to learn a task

- [Hessel ArXiv:1710.02298]
- Median performance on 57 Atari games relative to human performance (100%=human)
- Most methods require over 50 million frames to match human performance (230 hours of play)
- The best method (combination) takes 18 million frames (83 hours).



Pure RL is hard to use in the real world

- **Pure RL requires too many trials to learn anything**
  - it's OK in a game
  - it's not OK in the real world
- **RL works in simple virtual world that you can run faster than real-time on many machines in parallel.**

- **Anything you do in the real world can kill you**
- **You can't run the real world faster than real time**

Open Source Projects from FAIR

- **PyTorch: deep learning framework http://pytorch.org**
  - Many examples and tutorials. Used by many research groups.
- **FAISS: fast similarity search (C++/CUDA)**
- **ParlAI: training environment for dialog systems (Python)**
- **ELF: distributed reinforcement learning framework**

- **ELF OpenGo: super-human go-playing engine**
- **FastText: text classification, representation, embedding (C++)**
- **FairSeq: neural machine translation with ConvNets, RNN...**
- **Detectron / Mask-R-CNN: complete vision system**
- **DensePose: real-time body pose tracking system**
- **https://github.com/facebookresearch**

What are we missing?

To get to “real” AI

What current deep learning methods enables

- **What we can have**
  - Safer cars, autonomous cars
  - Better medical image analysis
  - Personalized medicine
  - Adequate language translation
  - Useful but stupid chatbots
  - Information search, retrieval, filtering
  - Numerous applications in energy, finance, manufacturing, environmental protection, commerce, law, artistic creation, games,…..
- **What we cannot have (yet)**
  - Machines with common sense
  - Intelligent personal assistants
  - “Smart” chatbots”
  - Household robots
  - Agile and dexterous robots
  - Artificial General Intelligence (AGI)

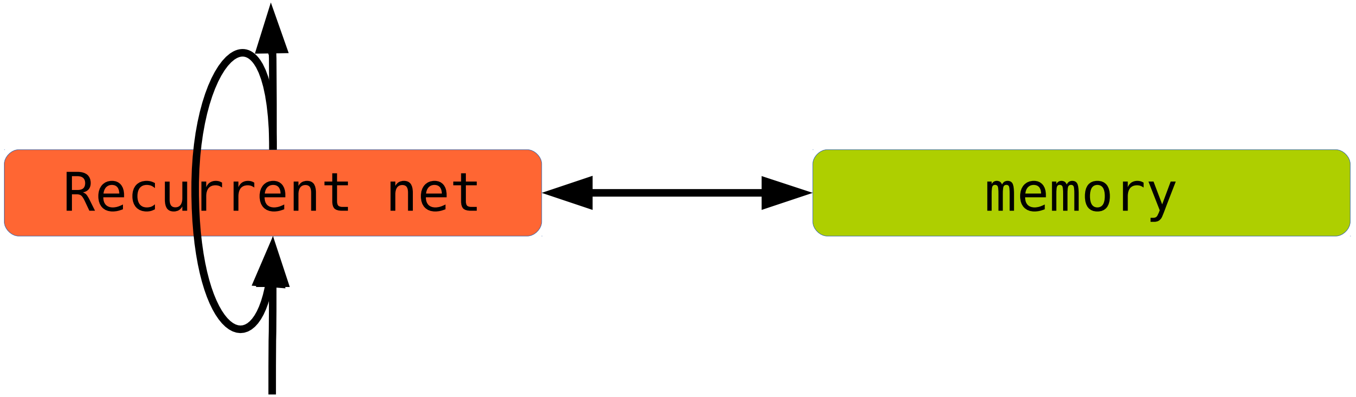
Differentiable Programming: Marrying Deep Learning With Reasoning

Neural nets with dynamic, data-dependent structure,
A program whose gradient is generated automatically.

facebook
Artificial Intelligence Research

Augmenting Neural Nets with a Memory Module

- **Recurrent networks cannot remember things for very long**
  - The cortex only remember things for 20 seconds
- **We need a "hippocampus" (a separate memory module)**
  - LSTM [Hochreiter 1997], registers
  - **Memory networks** [Weston et 2014] (FAIR), associative memory
  - **Stacked-Augmented Recurrent Neural Net** [Joulin & Mikolov 2014] (FAIR)
  - **Neural Turing Machine** [Graves 2014],
  - **Differentiable Neural Computer** [Graves 2016]



Dialog through Prediction [Weston et al. 2016]

Mary went to the hallway.

John moved to the bathroom.

Mary travelled to the kitchen.

Where is Mary? A:playground

No, that's incorrect.

Where is John? A:bathroom

Yes, that's right!

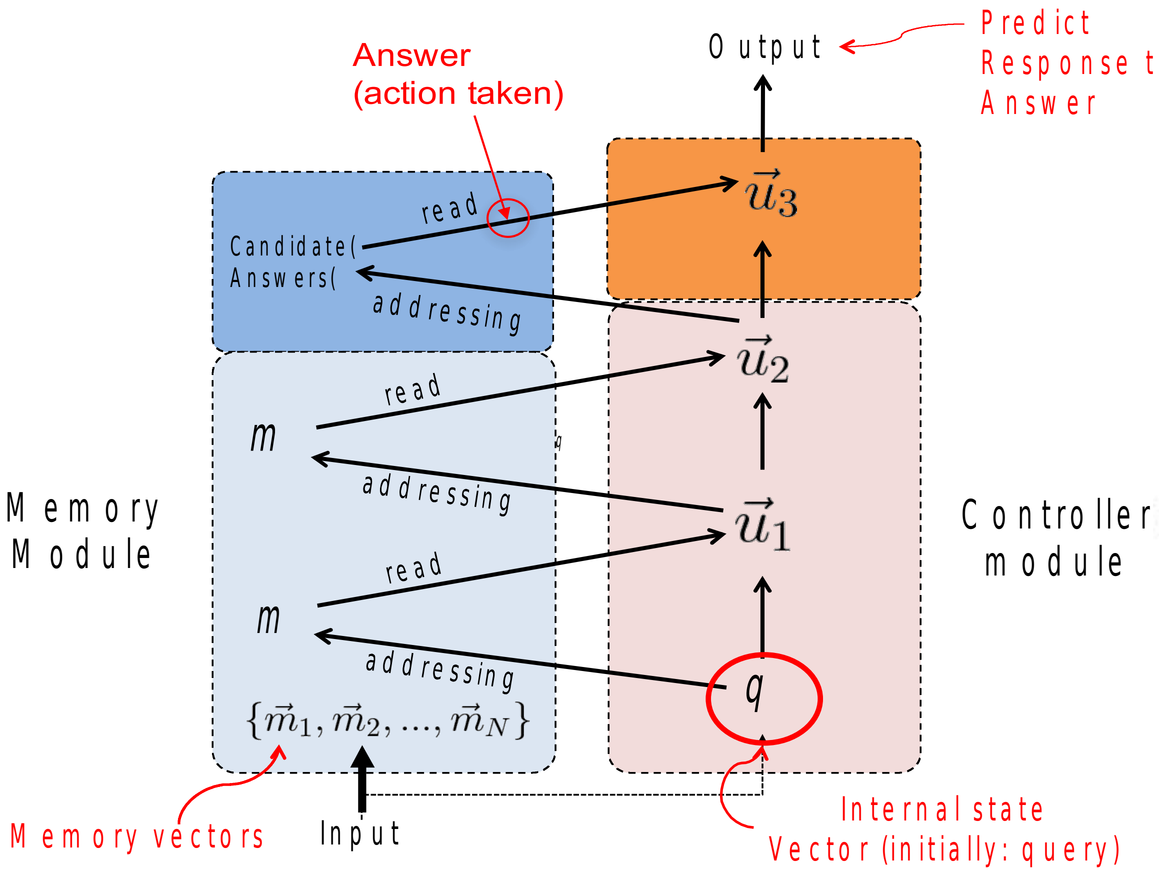
If you can predict this, you are most of the way to knowing how to answer correctly.

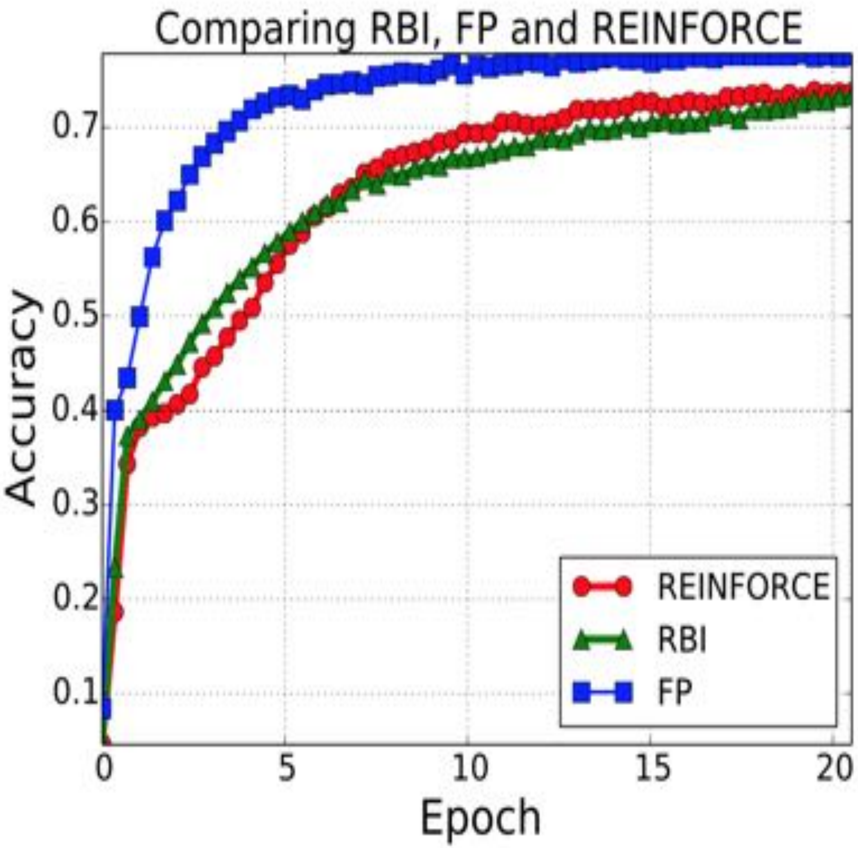
Dialog through Prediction [Weston et al. 2016]

Figure 2: **Human Dialogue from Mechanical Turk (based on WikiMovies)** The human teacher's dialogue is in black and the bot is in red. We show examples where the bot answers correctly (left) and incorrectly (right). Real humans provide more variability of language in both questions and textual feedback than in the simulator setup (cf. Figure 1).

Sample dialogues with correct answers from the bot:	
Who wrote the Linguini Incident ?	richard shepard
Richard Shepard is one of the right answers here.	
What year did The World Before Her premiere?	2012
Yep! That's when it came out.	
Which are the movie genres of Mystery of the 13th Guest?	crime
Right, it can also be categorized as a mystery.	
Sample dialogues with incorrect answers from the bot:	
What are some movies about a supermarket ?	supermarket
There were many options and this one was not among them.	
Which are the genres of the film Juwanna Mann ?	kevin pollak
That is incorrect. Remember the question asked for a genre not name.	
Who wrote the story of movie Coraline ?	fantasy
That's a movie genre and not the name of the writer. A better answer would of been Henry Selick or Neil Gaiman.	

Dialog through Prediction [Weston et al. 2016]



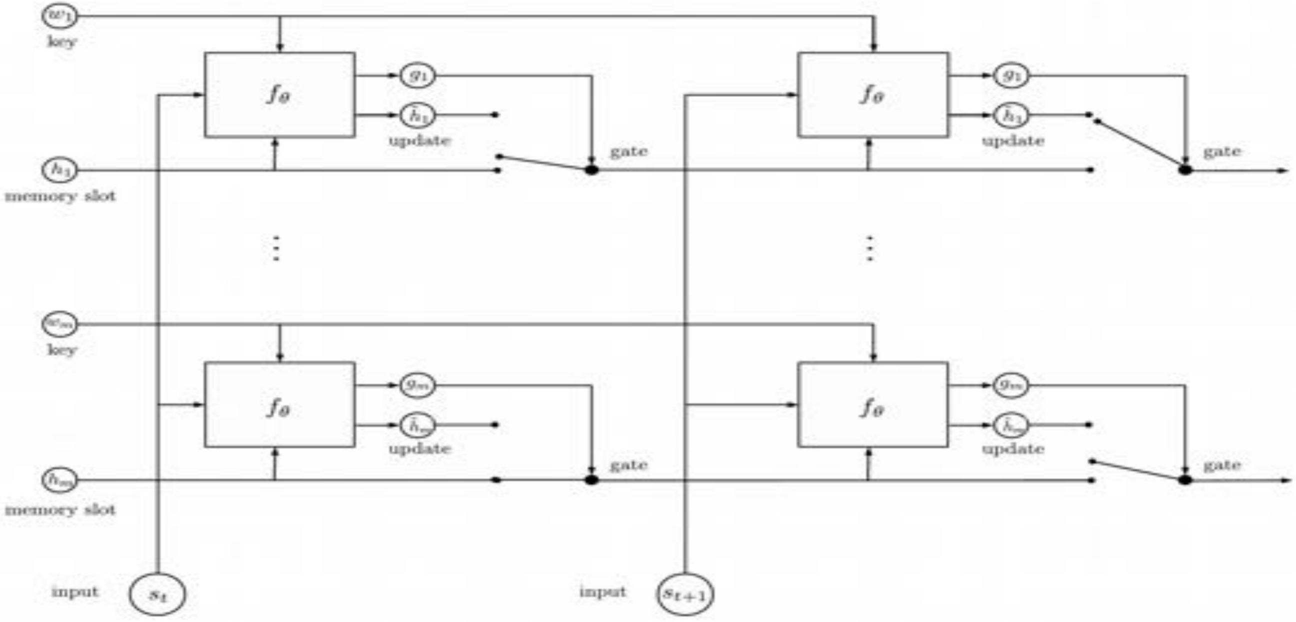


Tested on WikiMovies.

Forward Prediction MemNN (FP) with textual rewards perform better than numerical rewards!

EntNet: Entity Recurrent Neural Net

- **Maintains a current estimate of the state of the world.**
- **Each module is a recurrent net with a "memory"**
- **Each input event causes some of the memory cells to get updated**
- **"Tracking the World State with Recurrent Entity Networks", [Henaff, Weston, Szlam, Bordes, LeCun, ICLR 2017]**



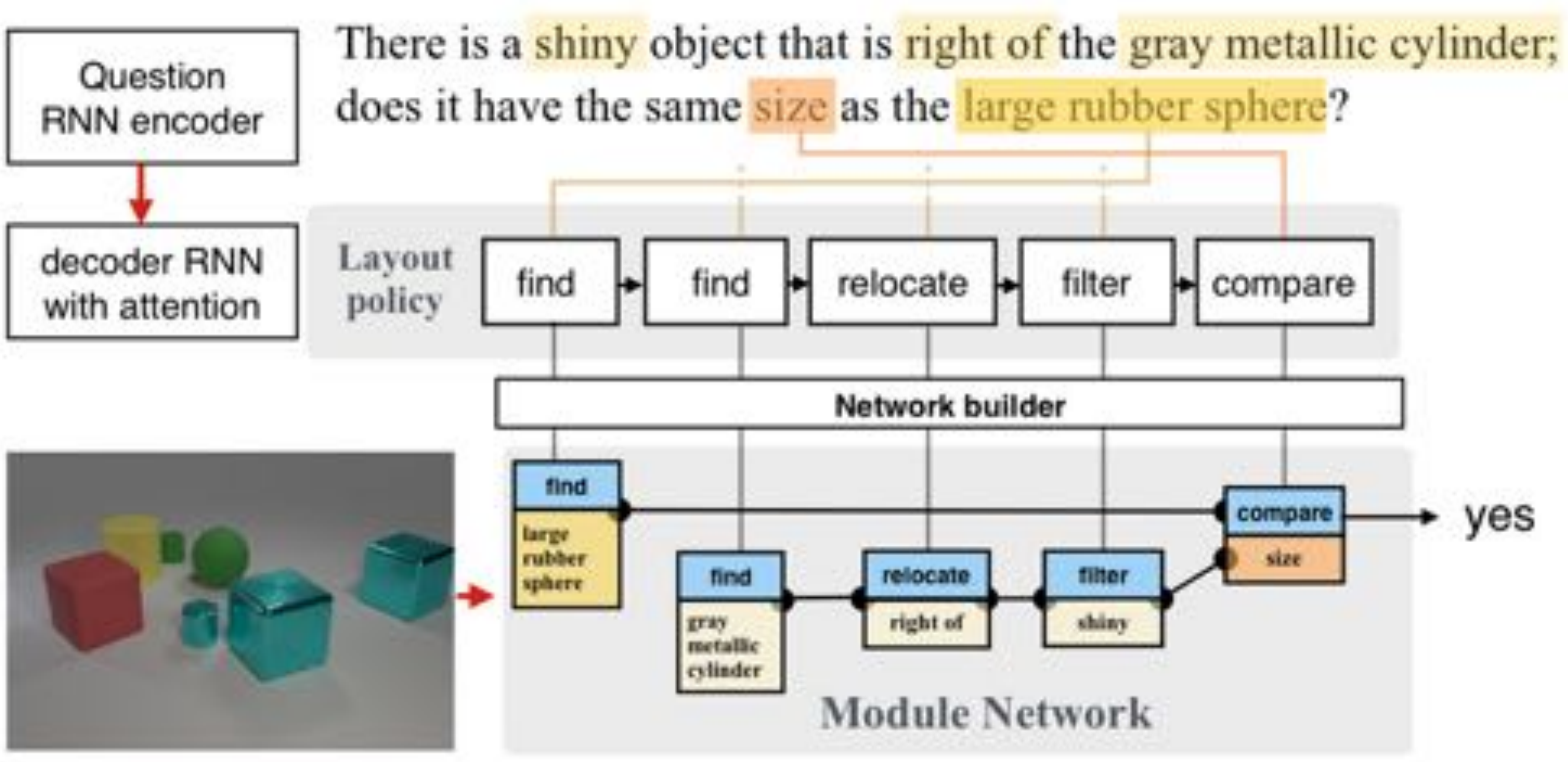
EntNet is the first model to solve all 20 bAbI tasks

Task	D-NTM	MemN2N	DNC	DMN+	EntNet
1: 1 supporting fact	4.4	0	0	0	0
2: 2 supporting facts	27.5	0.3	0.4	0.3	0.1
3: 3 supporting facts	71.3	2.1	1.8	1.1	4.1
4: 2 argument relations	0	0	0	0	0
5: 3 argument relations	1.7	0.8	0.8	0.5	0.3
6: yes/no questions	1.5	0.1	0	0	0.2
7: counting	6.0	2.0	0.6	2.4	0
8: lists/sets	1.7	0.9	0.3	0.0	0.5
9: simple negation	0.6	0.3	0.2	0.0	0.1
10: indefinite knowledge	19.8	0	0.2	0	0.6
11: basic coreference	0	0.0	0	0.0	0.3
12: conjunction	6.2	0	0	0.2	0
13: compound coreference	7.5	0	0	0	1.3
14: time reasoning	17.5	0.2	0.4	0.2	0
15: basic deduction	0	0	0	0	0
16: basic induction	49.6	51.8	55.1	45.3	0.2
17: positional reasoning	1.2	18.6	12.0	4.2	0.5
18: size reasoning	0.2	5.3	0.8	2.1	0.3
19: path finding	39.5	2.3	3.9	0.0	2.3
20: agent's motivation	0	0	0	0	0
Failed Tasks (> 5% error):	9	3	2	1	**0**
Mean Error:	12.8	4.2	3.8	2.8	**0.5**

- **Posted on arXiv in Nov 2016**
- **Presented at ICLR in May 2017**
- **Since then two other groups have used similar ideas and improved the results**
  - DeepMind
  - Umass Amherst
- **Posting on arXiv accelerates the rate of progress of science**

Inferring and executing programs for visual reasoning

https://research.fb.com/visual-reasoning-and-dialog-towards-natural-language-conversations-about-visual-data/



PyTorch: differentiable programming

- **Software 2.0:**
  - The operations in a program are only partially specified
  - They are trainable parameterized modules.
  - The precise operations are learned from data, only the general structure of the program is designed.

How do Humans and Animal Learn?

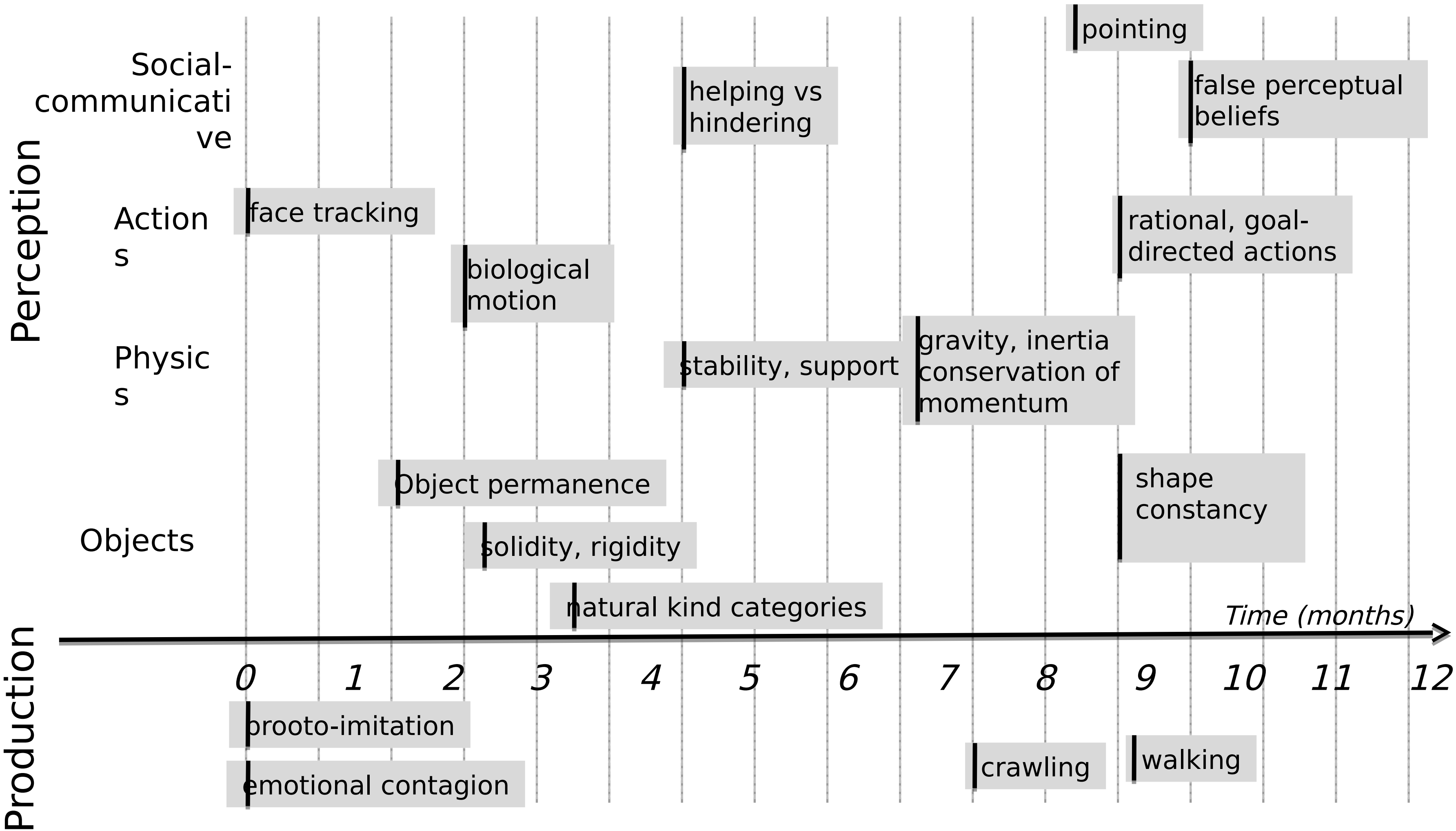
So quickly

Babies learn how the world works by observation

- **Largely by observation, with remarkably little interaction.**

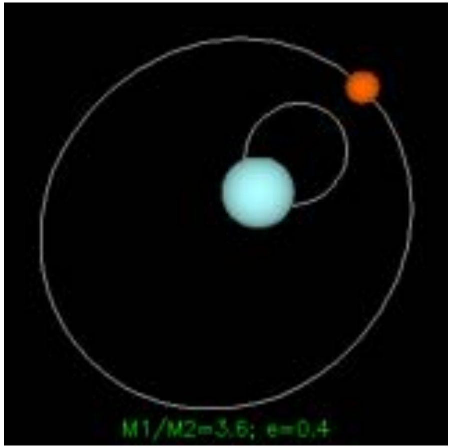
Photos courtesy of Emmanuel Dupoux

Early Conceptual Acquisition in Infants [from Emmanuel Dupoux]



Prediction is the essence of Intelligence

- **We learn models of the world by predicting**



Three Types of Learning

- **Reinforcement Learning**
  - The machine predicts a scalar reward given once in a while.
  - **weak feedback**
- **Supervised Learning**
  - The machine predicts a category or a few numbers for each input
  - **medium feedback**
- **Self-supervised Predictive Learning**
  - The machine predicts any part of its input for any observed part.
  - Predicts future frames in videos
  - **A lot of feedback**

How Much Information is the Machine Given during Learning?

- **"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**
- **Self-Supervised Learning (cake génoise)**
  - The machine predicts any part of its input for any observed part.
  - Predicts future frames in videos
  - **Millions of bits per sample**

Two Big Questions on the way to “Real AI”

- **How can machines learn as efficiently as humans and animals?**
  - By observation
  - without supervision
  - with very little interactions with the world
- **How can we train machines to plan and act (not just perceive)?**
  - Where inference involves a complex iterative process

- **Learning predictive forward models of the world under uncertainty**
  - Learning hierarchical representations of the world unsupervised
  - Enabling long-term planning using the model
  - Enabling learning in the real world with few interactions

With thanks
To
Alyosha Efros

Common Sense is the ability to fill in the blanks

- Infer the state of the world from partial information
- Infer the future from the past and present
- Infer past events from the present state

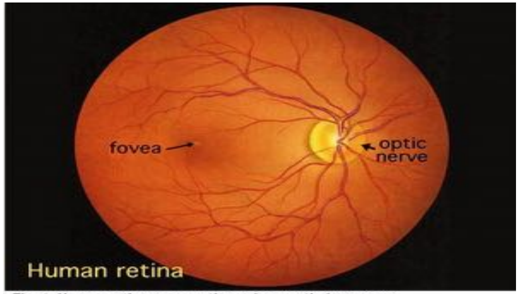


Fig. 1. Human retina as seen through an opthalmoscope.

- Filling in the visual field at the retinal blind spot
- Filling in occluded images, missing segments in speech
- Predicting the state of the world from partial (textual) descriptions
- Predicting the consequences of our actions
- Predicting the sequence of actions leading to a result

- Predicting any part of the past, present or future percepts from whatever information is available.

- That's what self-supervised predictive learning is
- But really, that's what many people mean by unsupervised learning

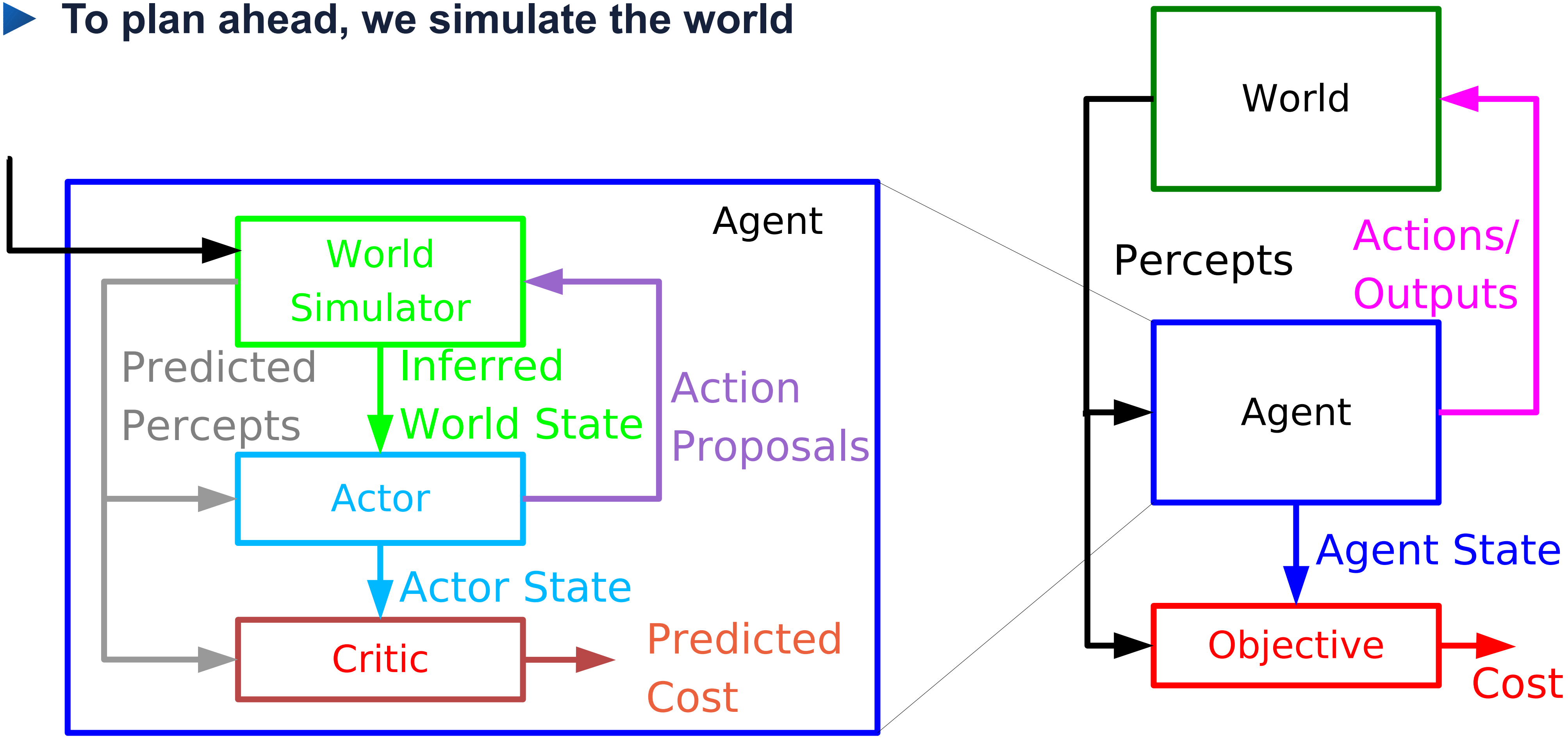
Learning Predictive Models of the World

Learning to predict, reason, and plan, Learning Common Sense.

facebook
Artificial Intelligence Research

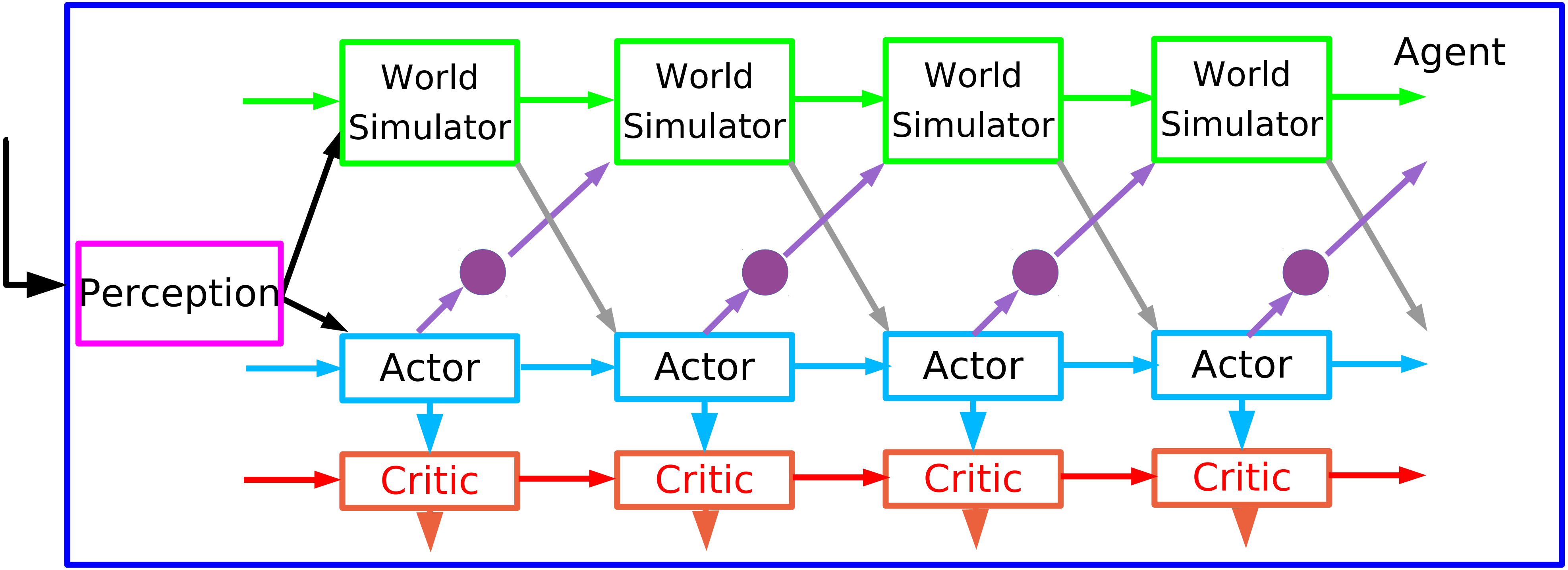
Planning Requires Prediction

- **To plan ahead, we simulate the world**



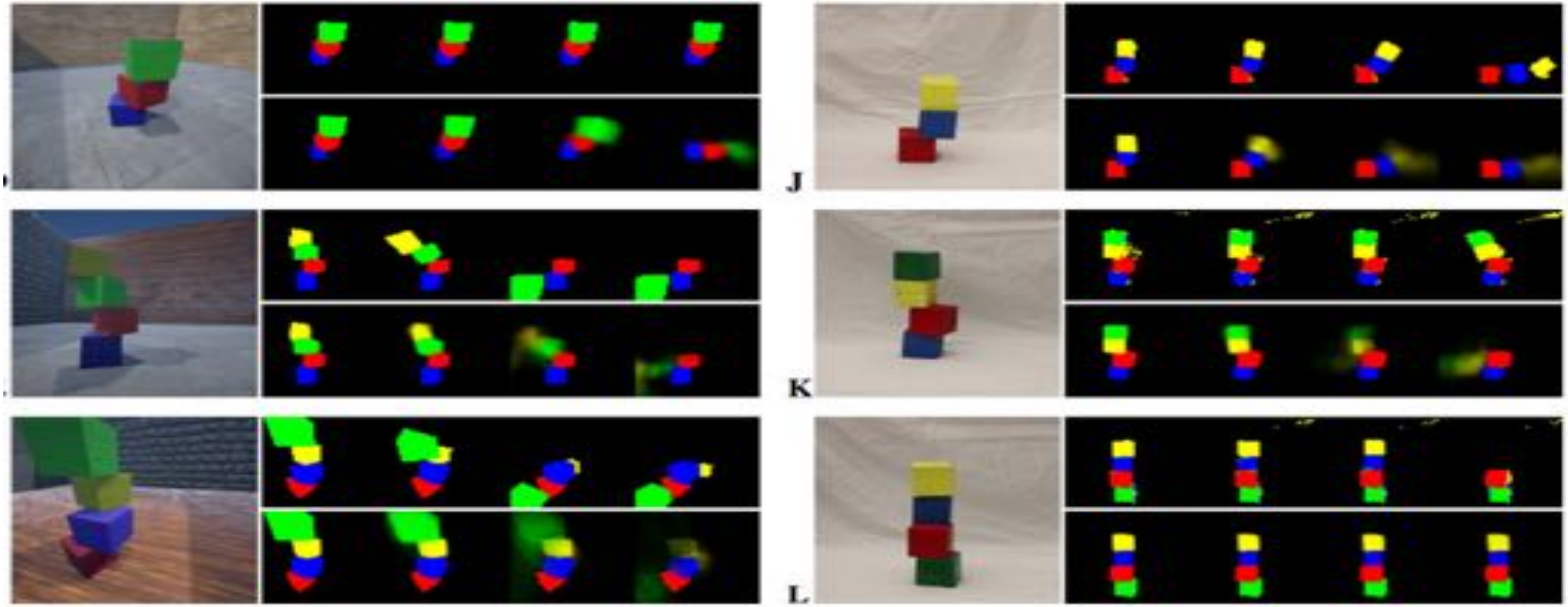
Training the Actor with Optimized Action Sequences

- **1. Find action sequence through optimization**
- **2. Use sequence as target to train the actor**
  - Over time we get a compact policy that requires no run-time optimization



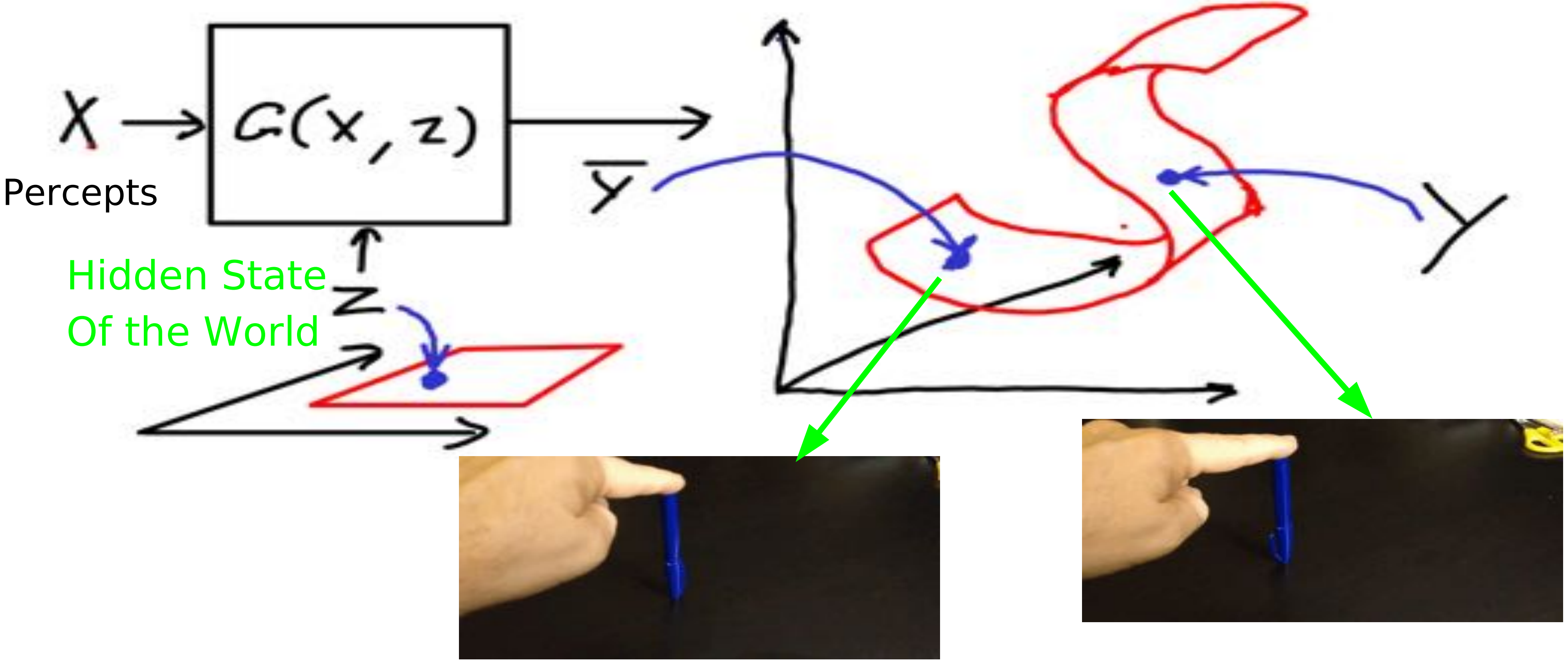
Learning Physics (PhysNet)

- **[Lerer, Gross, Fergus ICML 2016, arxiv:1603.01312]**
  - ConvNet produces object masks that predict the trajectories of falling blocks. Blurry predictions when uncertain



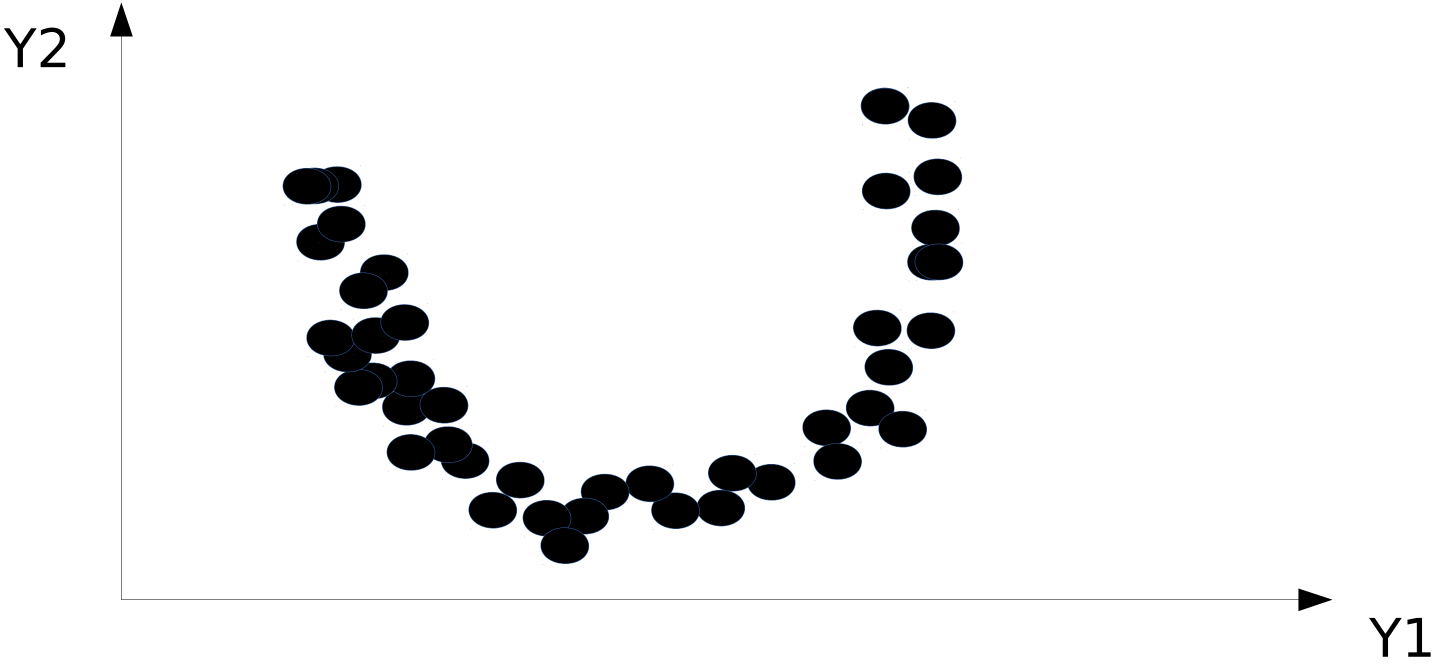
The Hard Part: Prediction Under Uncertainty

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



Learning the "Data Manifold": Energy-Based Approach

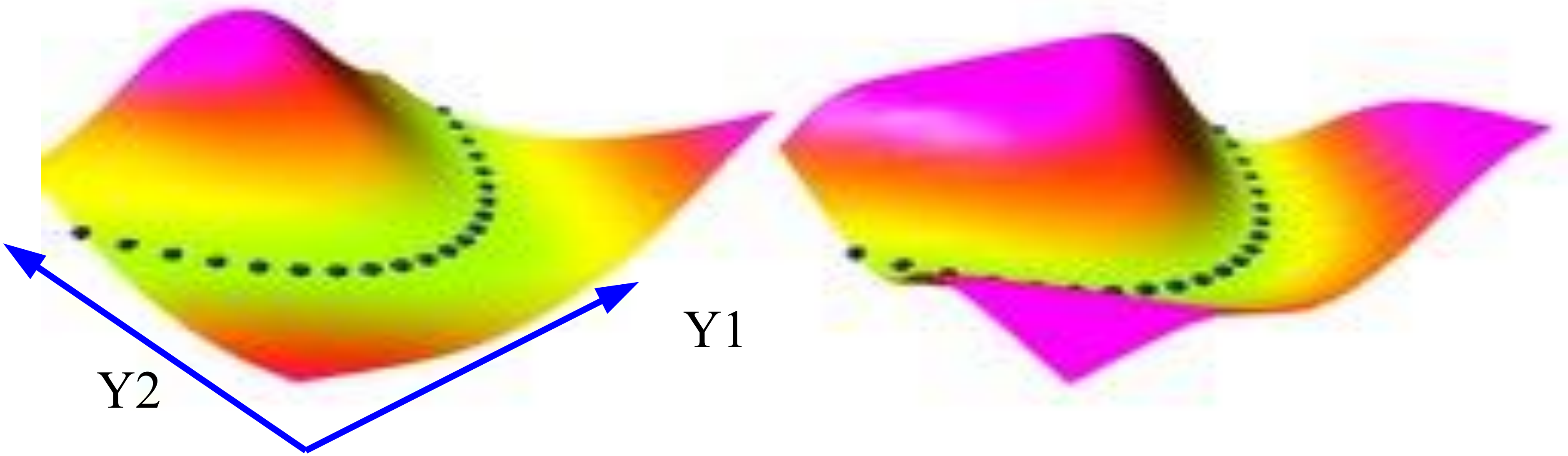
- **Learning an energy function (or contrast function) that takes**
  - Low values on the data manifold
  - Higher values everywhere else



Capturing Dependencies Between Variables with an Energy Function

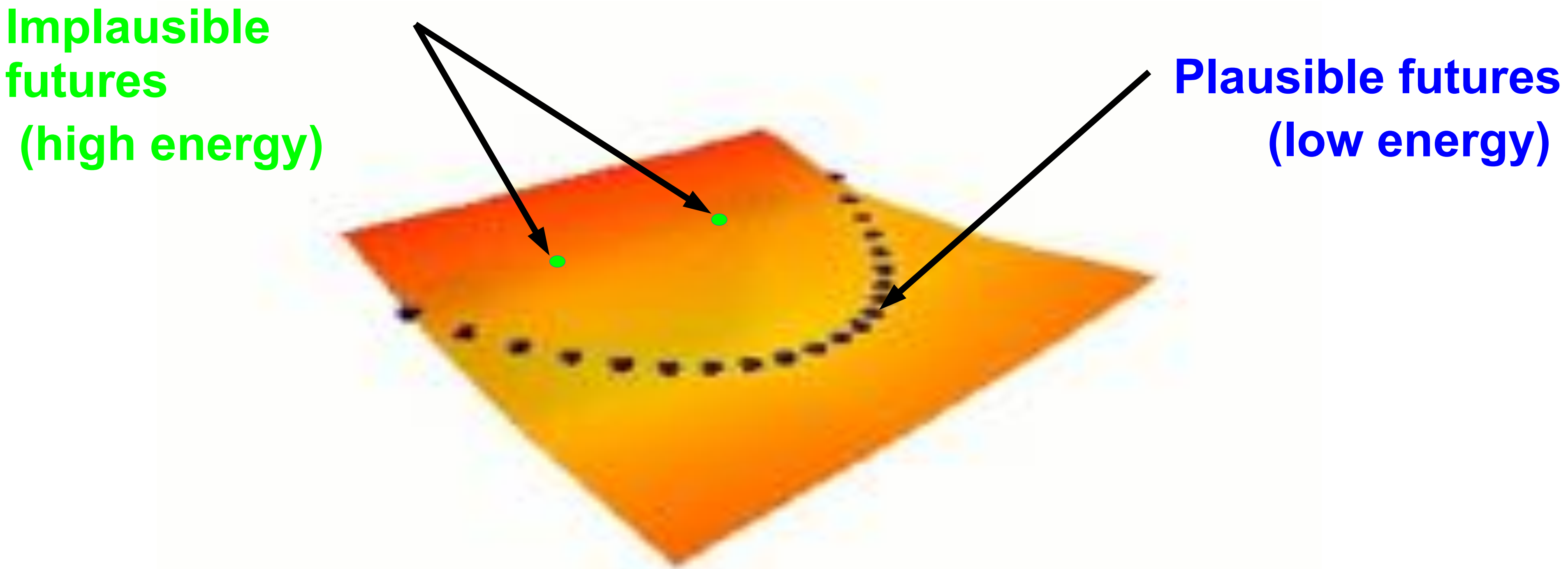
- **The energy surface is a "contrast function" that takes low values on the data manifold, and higher values everywhere else**
  - Special case: energy = negative log density
  - Example: the samples live in the manifold

$$Y_2 = (Y_1)^2$$



Energy Function for Data Manifold

- **Energy Function:** Takes low value on data manifold, higher values everywhere else
- **Push down on the energy of desired outputs. Push up on everything else.**
- **But how do we choose where to push up?**

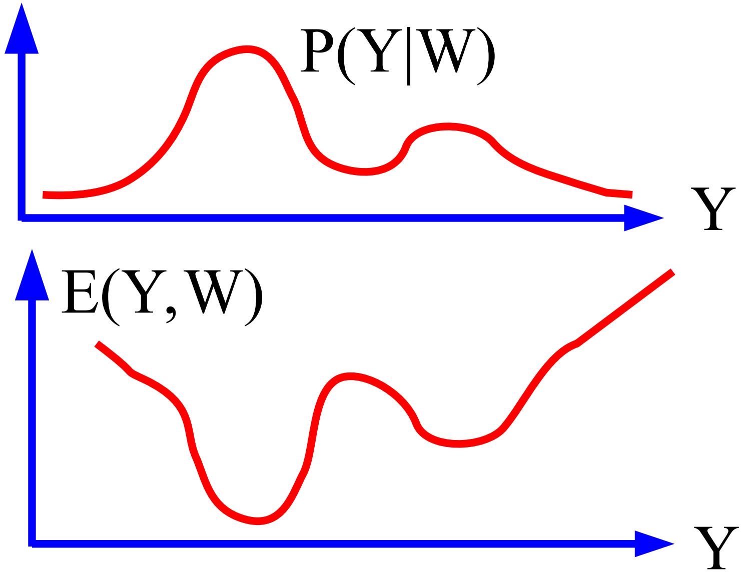


Transforming Energies into Probabilities (if necessary)

- **The energy can be interpreted as an unnormalized negative log density**
- **Gibbs distribution: Probability proportional to exp(-energy)**
  - Beta parameter is akin to an inverse temperature
- **Don't compute probabilities unless you absolutely have to**
  - Because the denominator is often intractable

$$P(Y|W) = \frac{e^{-\beta E(Y,W)}}{\int_y e^{-\beta E(y,W)}}$$

$$E(Y,W) \propto -\log P(Y|W)$$



Learning the Energy Function

- **parameterized energy function E(Y,W)**
  - Make the energy low on the samples
  - Make the energy higher everywhere else
  - Making the energy low on the samples is easy
  - But how do we make it higher everywhere else?

Seven Strategies to Shape the Energy Function

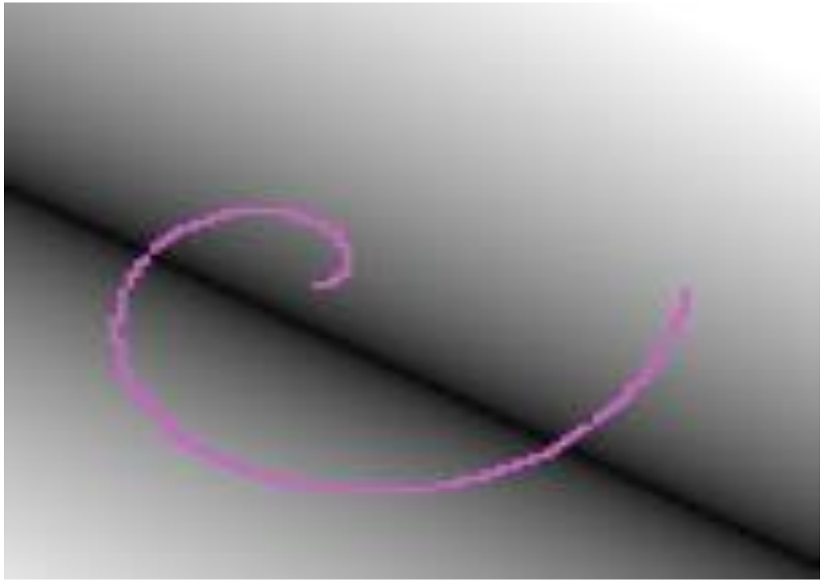
- **1. build the machine so that the volume of low energy stuff is constant**
  - PCA, K-means, GMM, square ICA
- **2. push down of the energy of data points, push up everywhere else**
  - Max likelihood (needs tractable partition function or variational approximation)
- **3. push down of the energy of data points, push up on chosen locations**
  - Contrastive divergence, Ratio Matching, Noise Contrastive Estimation, Min Probability Flow
- **4. minimize the gradient and maximize the curvature around data points**
  - score matching
- **5. train a dynamical system so that the dynamics goes to the manifold**
  - denoising auto-encoder
- **6. use a regularizer that limits the volume of space that has low energy**
  - Sparse coding, sparse auto-encoder, PSD
- **7. if E(Y) = ||Y - G(Y)||^2, make G(Y) as "constant" as possible.**
  - Contracting auto-encoder, saturating auto-encoder

#1: constant volume of low energy
Energy surface for PCA and K-means

- **1. build the machine so that the volume of low energy stuff is constant**
  - PCA, K-means, GMM, square ICA...

PCA

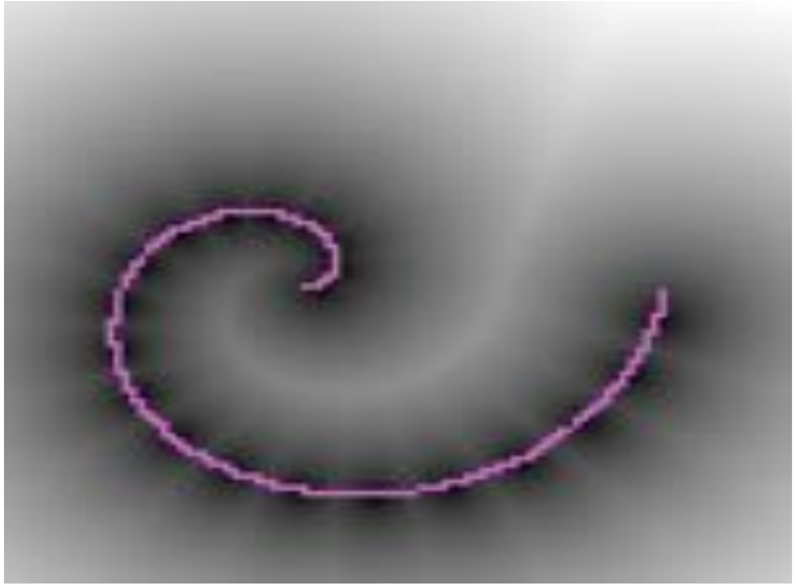
$$E(Y) = ||W^T W Y - Y||^2$$



K-Means,

Z constrained to 1-of-K code

$$E(Y) = min_z \sum_i ||Y - W_i Z_i||^2$$



•#2: push down of the energy of data points, push up everywhere else

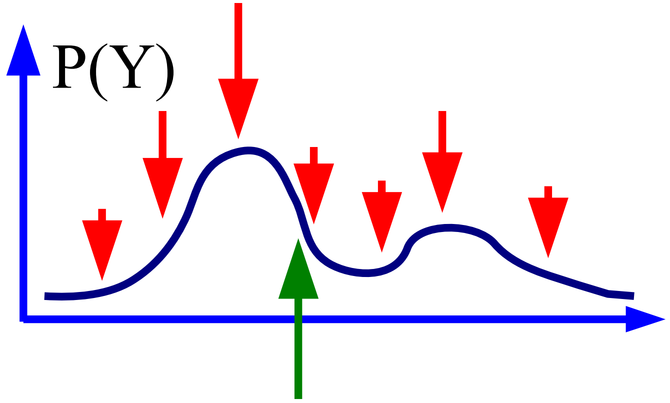
- **Max likelihood (requires a tractable partition function)**

Maximizing P(Y|W) on training samples

$$P(Y|W) = \frac{e^{-\beta E(Y,W)}}{\int_y e^{-\beta E(y,W)}}$$

make this big

make this small

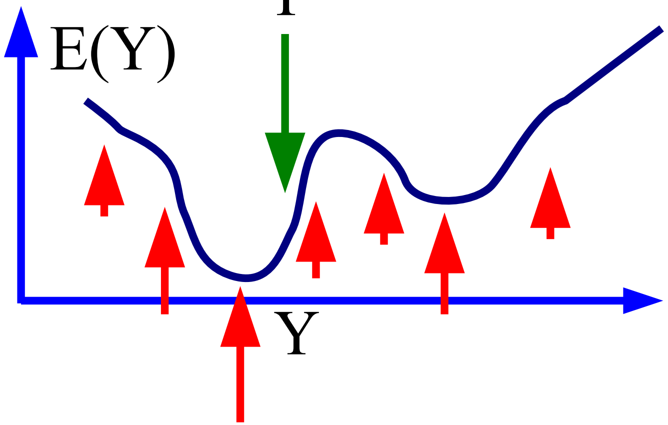


Minimizing -log P(Y,W) on training samples

$$L(Y,W) = E(Y,W) + \frac{1}{\beta} \log \int_y e^{-\beta E(y,W)}$$

make this small

make this big



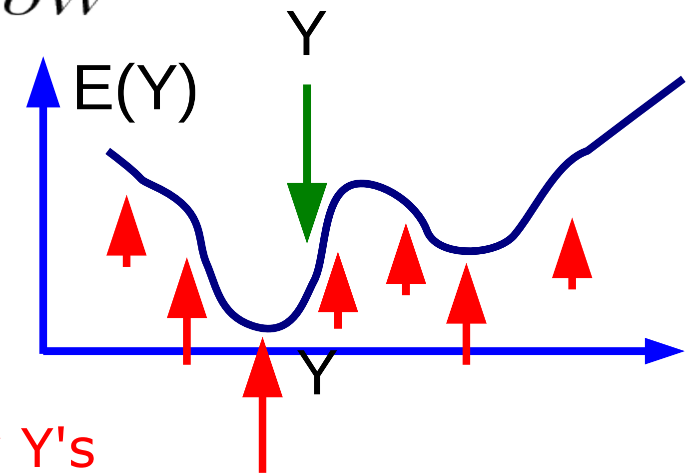
•#2: push down of the energy of data points, push up everywhere else

Gradient of the negative log-likelihood loss for one sample Y:

$$\frac{\partial L(Y,W)}{\partial W} = \frac{\partial E(Y,W)}{\partial W} - \int_y P(y|W)\frac{\partial E(y,W)}{\partial W}$$

Gradient descent:

$$W \leftarrow W - \eta \frac{\partial L(Y,W)}{\partial W}$$



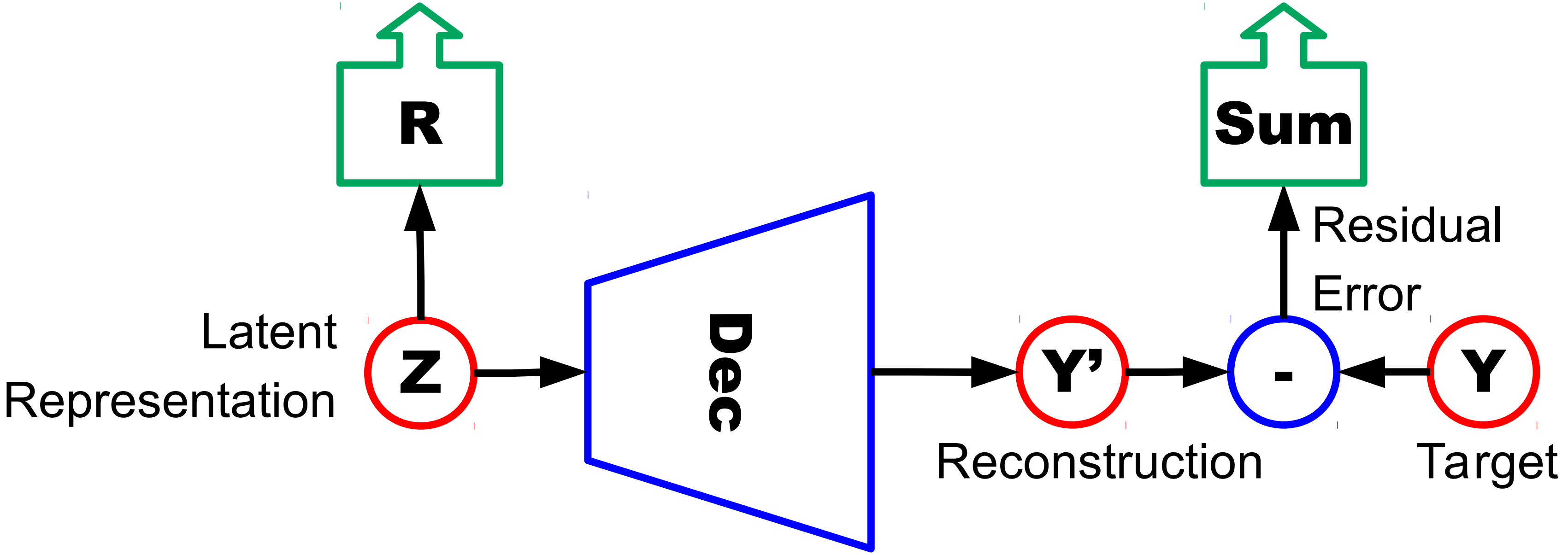
Pushes down on the energy of the samples

Pulls up on the energy of low-energy Y's

$$W \leftarrow W - \eta \frac{\partial E(Y,W)}{\partial W} + \eta \int_y P(y|W)\frac{\partial E(y,W)}{\partial W}$$

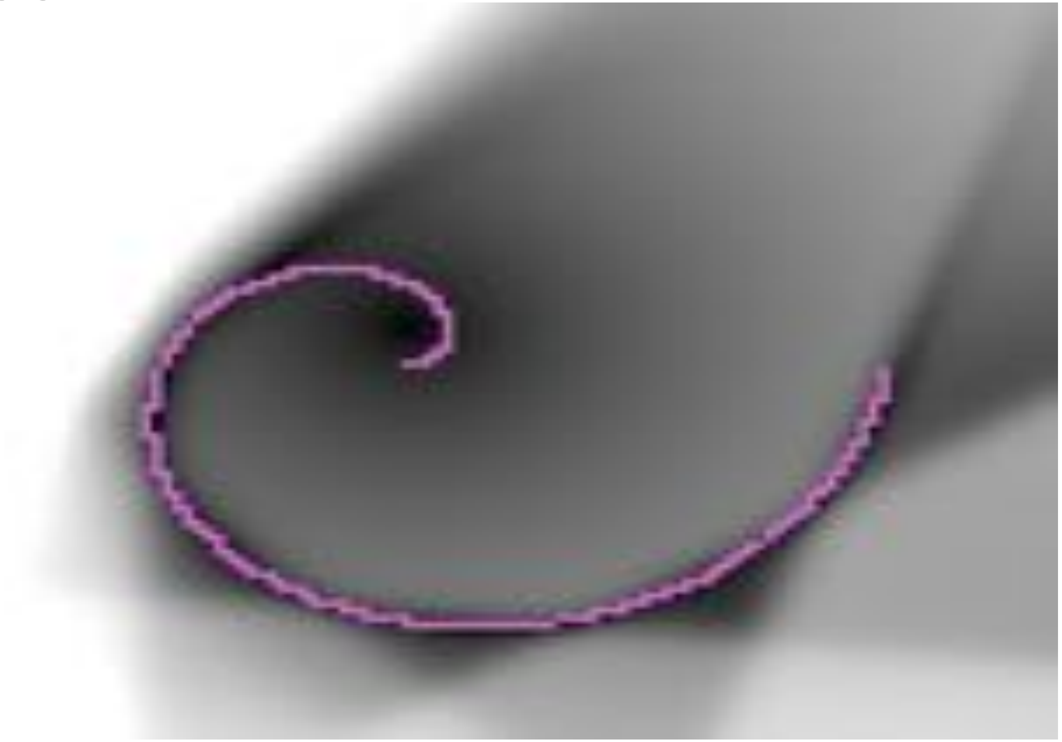
The "Decoder with Restricted Latent Variable" Model

- Y' = Dec(Z)   Z* = argmin || Y – Dec(Z) || + R(Z)
- Linear decoder: K-Means, basis pursuit, K-SVD, sparse coding,….
- Multilayer/non-linear decoder: GLO [Bojanowski et al. 2017]



#6. use a regularizer that limits the volume of space that has low energy

- **Sparse coding, sparse auto-encoder, Predictive Sparse Decomposition**

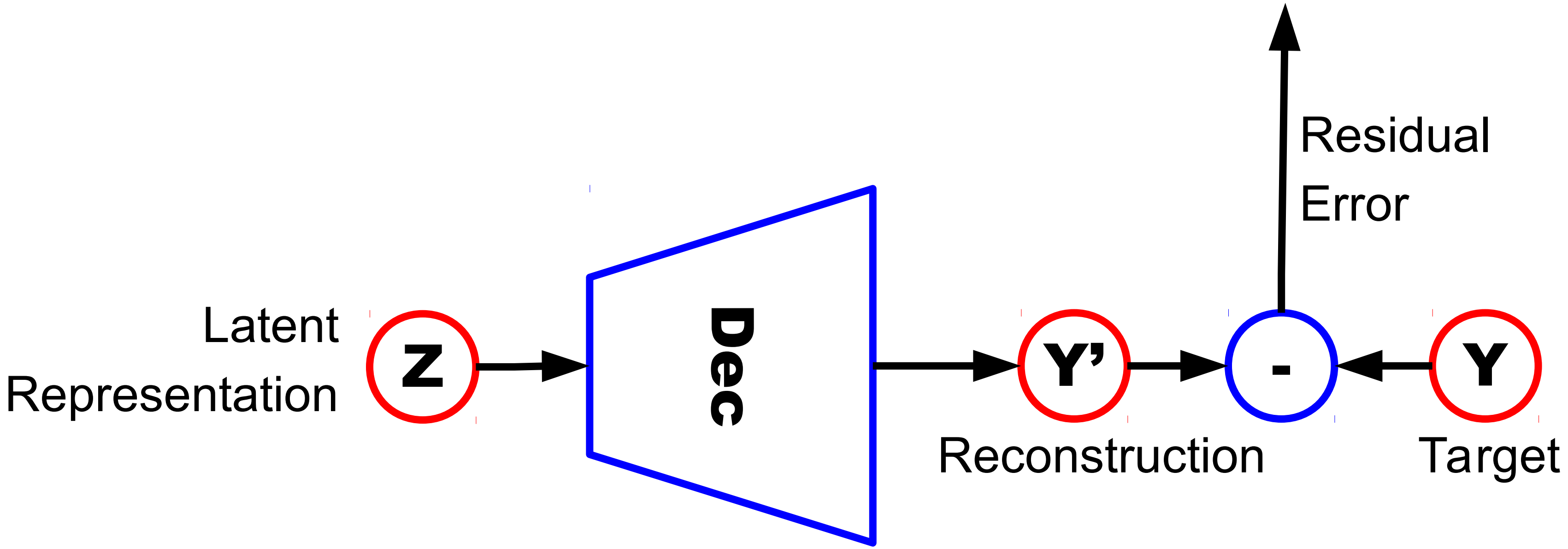


Learning Generative (Forward) Models With Latent Variables

Generation through Latent Optimization

[Bojanowski, Joulin, Lopez-Paz, Szlam arxiv:1707.05776]

- **Y' = Dec(Z)**   **Z* = argmin || Y – Dec(Z) ||**



Generation through Latent Optimization

[Bojanowski, Joulin, Lopez-Paz, Szlam arxiv:1707.05776]

- **Original**
- **Reconstr. D=100**
- **Reconstr D=512**

Generation through Latent Optimization

[Bojanowski, Joulin, Lopez-Paz, Szlam arxiv:1707.05776]

- **Interpolation in Z space**

- **Interpolation in pixels**

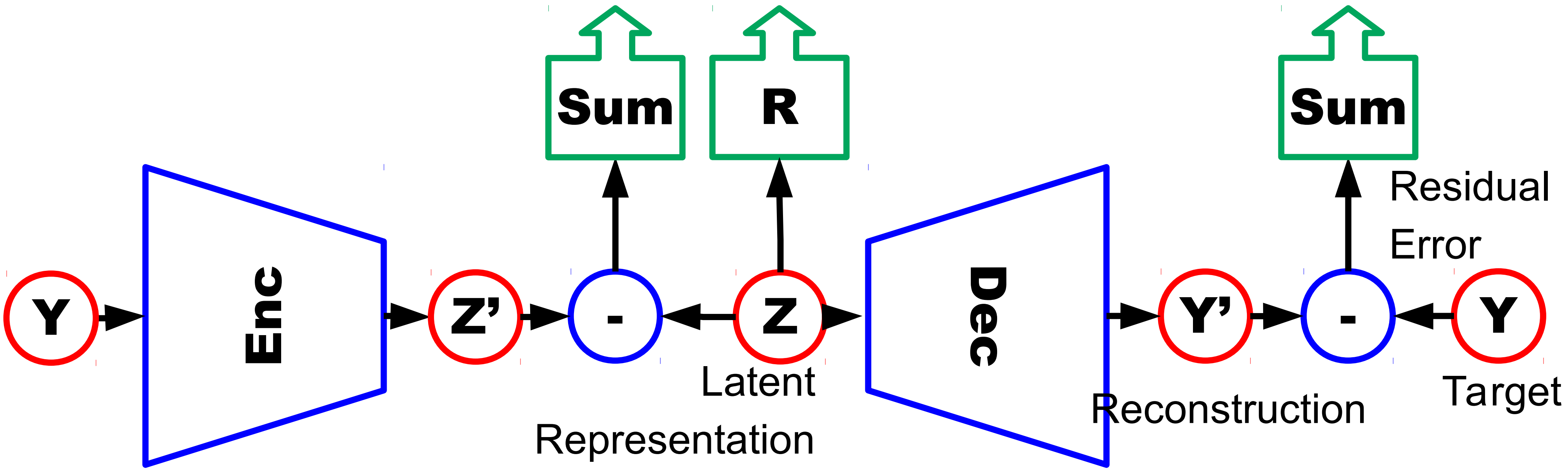
Convolutional Sparse Auto-Encoders

[Kavukcuoglu NIPS 2010]

"Learning convolutional feature hierarchies for visual recognition"

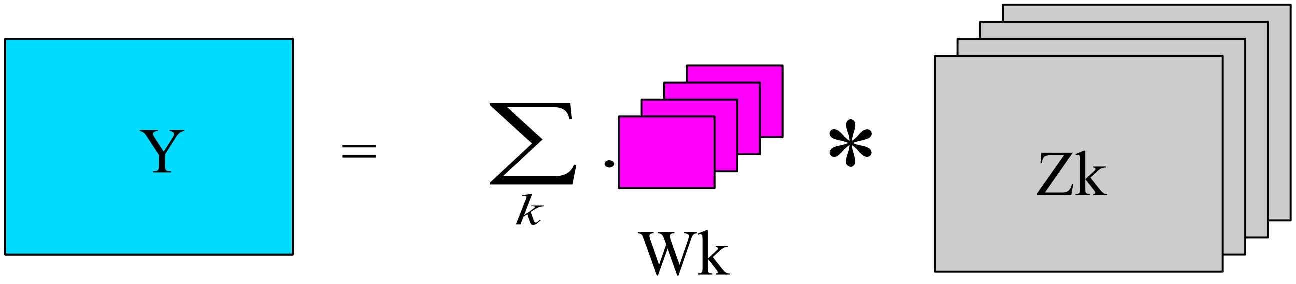
The “Encoder-Decoder with latent vars” Model

- Z* = argmin || Y – Dec(Z) || + R(Z) + || Z – Enc(Y) ||
- Linear decoder: Predictive Sparse Decomposition [Kavukcuoglu 2009]
- Convolutional decoder [[Kavukcuoglu 2010]



Convolutional Sparse Coding

- **Replace the dot products with dictionary element by convolutions.**
  - Input Y is a full image
  - Each code component Zk is a feature map (an image)
  - Each dictionary element is a convolution kernel
- **Regular sparse coding** $E(Y,Z) = ||Y - \sum_k W_k Z_k||^2 + \alpha \sum_k |Z_k|$
- **Convolutional S.C.** $E(Y,Z) = ||Y - \sum_k W_k * Z_k||^2 + \alpha \sum_k |Z_k|$

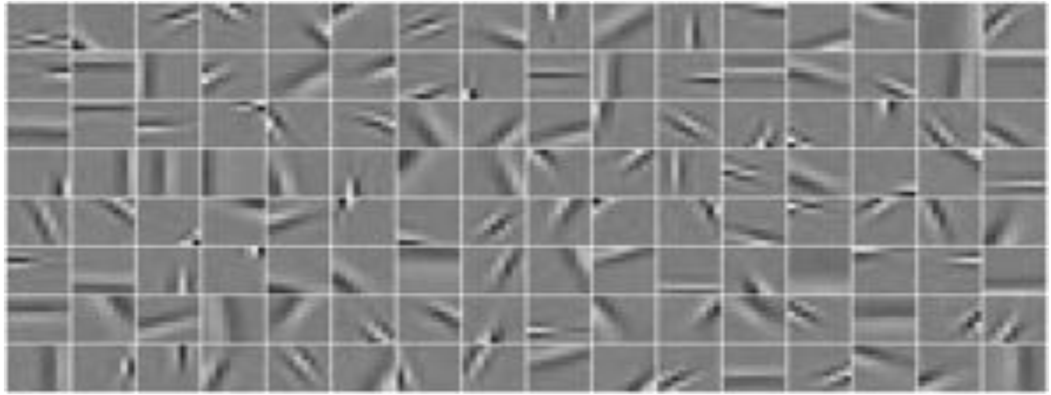


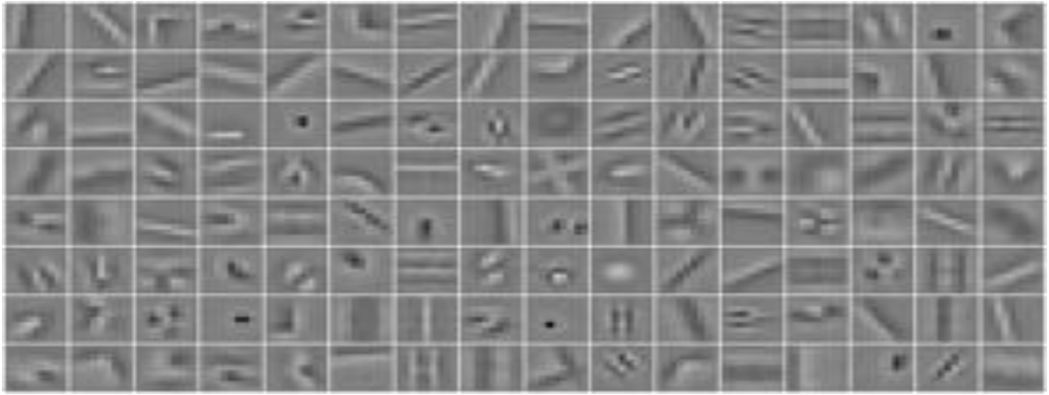
"deconvolutional networks" [Zeiler, Taylor, Fergus CVPR 2010]

Convolutional PSD: Encoder with a soft sh() Function

- **Convolutional Formulation**
  - Extend sparse coding from ***PATCH*** to ***IMAGE***

$$\mathcal{L}(x,z,\mathcal{D}) = \frac{1}{2}||x - \sum_{k=1}^{K} \mathcal{D}_k * z_k||_2^2 + \sum_{k=1}^{K} ||z_k - f(W^k * x)||_2^2 + |z|_1$$

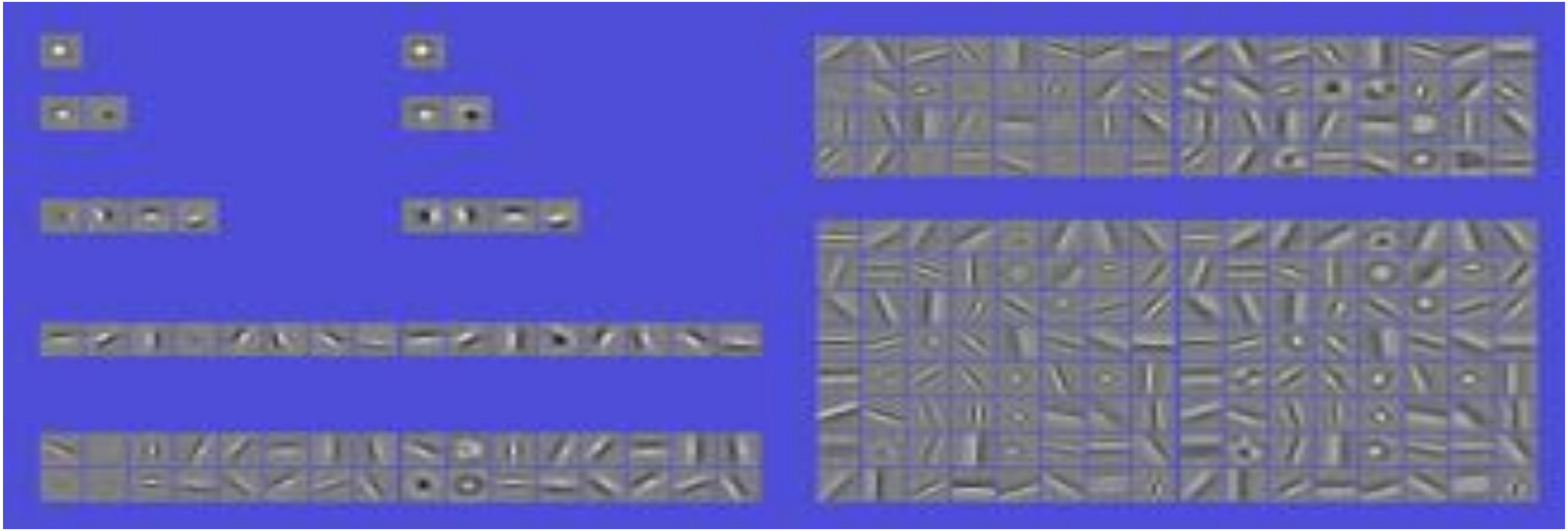




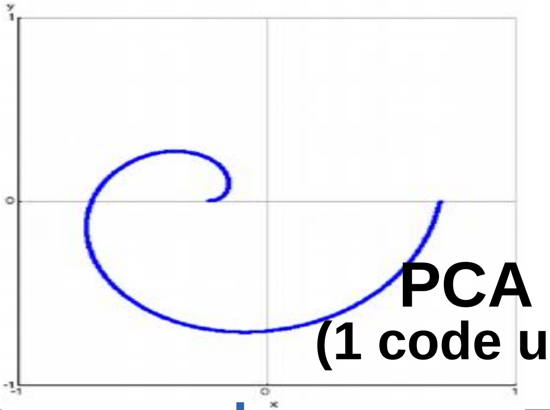
- ***PATCH*** based learning
- ***CONVOLUTIONAL*** learning

Convolutional Sparse Auto-Encoder on Natural Images

- Filters and Basis Functions obtained with 1, 2, 4, 8, 16, 32, and 64 filters.

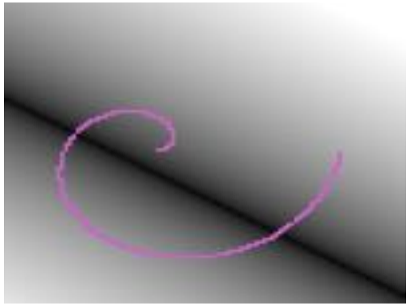


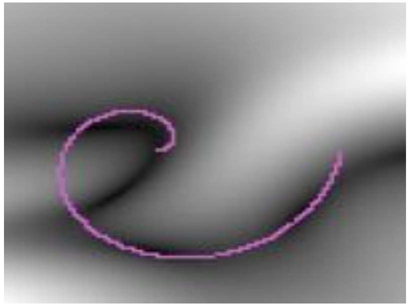
Energy Functions of Various Methods

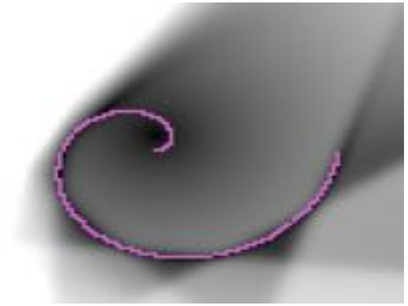


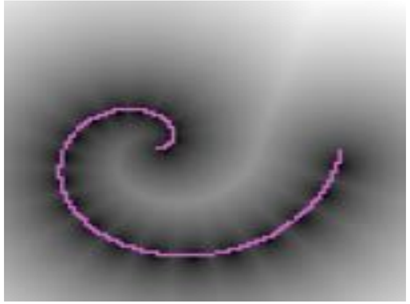
- **2 dimensional toy dataset: points on a spiral**
- **Visualizing the energy surface**
  - black = low energy, white = high energy

	PCA (1 code unit)	**autoencoder** (1 code unit)	**sparse coding** (20 code units)	**K-Means** (20 code units)
encoder	$W'Y$	$\sigma(W_e Y)$	$\sigma(W_e Z)$	$-$
decoder	WZ	$W_d Z$	$W_d Z$	WZ
energy	$\|\|Y-WZ\|\|^2$	$\|\|Y-WZ\|\|^2$	$\|\|Y-WZ\|\|^2$	$\|\|Y-WZ\|\|^2$
loss	$F(Y)$	$F(Y)$	$F(Y)$	$F(Y)$
pull-up	dimension	dimension	sparsity	1-of-N code









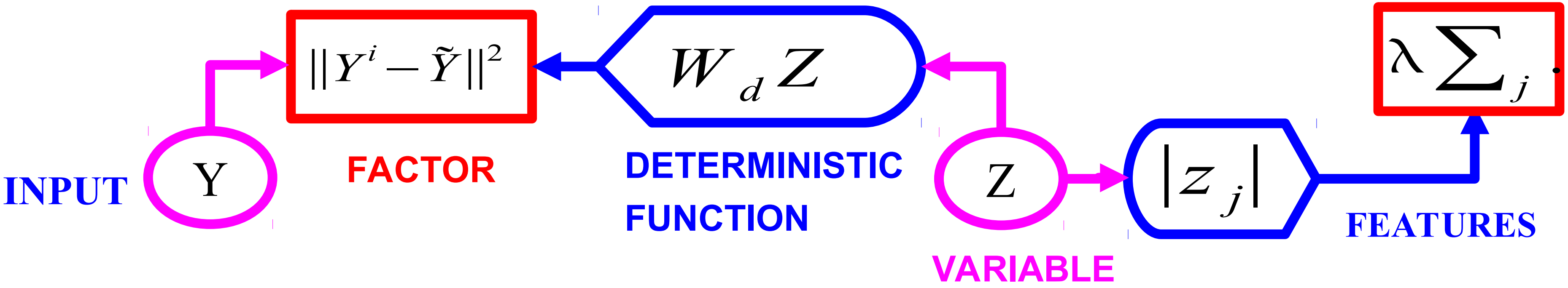
Learning to Perform Approximate Inference LISTA

Sparse Modeling: Sparse Coding + Dictionary Learning

[Olshausen & Field 1997]

- **Sparse linear reconstruction**
- **Energy = reconstruction_error + code_prediction_error + code_sparsity**

$$E(Y^i, Z) = \|Y^i - W_d Z\|^2 + \lambda \sum_j |z_j|$$

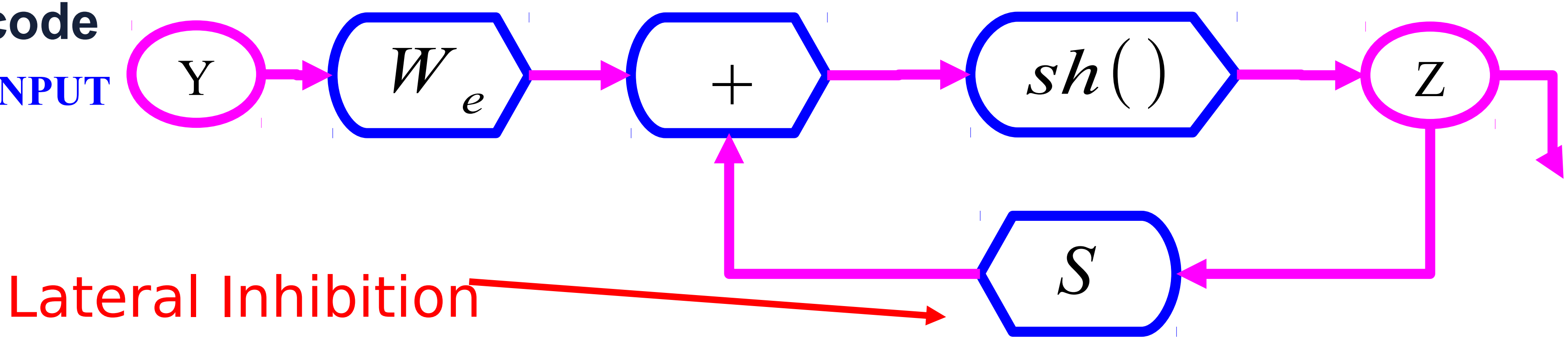


- **Inference is expensive: ISTA/FISTA, CGIHT, coordinate descent....**

$$Y \rightarrow \hat{Z} = argmin_Z E(Y, Z)$$

Better Idea: Give the “right” structure to the encoder

- **ISTA/FISTA: iterative algorithm that converges to optimal sparse code**



$$Z(t+1) = \mathrm{Shrinkage}_{\lambda/L}\left[Z(t) - \frac{1}{L} W_d^T (W_d Z(t) - Y) \right]$$

- **ISTA/FISTA reparameterized:**

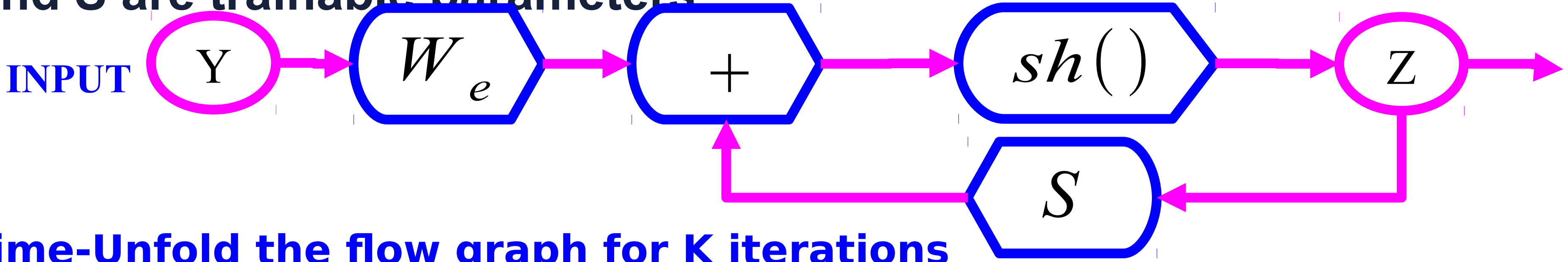
$$Z(t+1) = \mathrm{Shrinkage}_{\lambda/L}\left[W_e^T Y + S Z(t) \right]; \quad W_e = \frac{1}{L} W_d; \quad S = I - \frac{1}{L} W_d^T W_d$$

- **LISTA (Learned ISTA): learn the We and S matrices to get fast solutions**

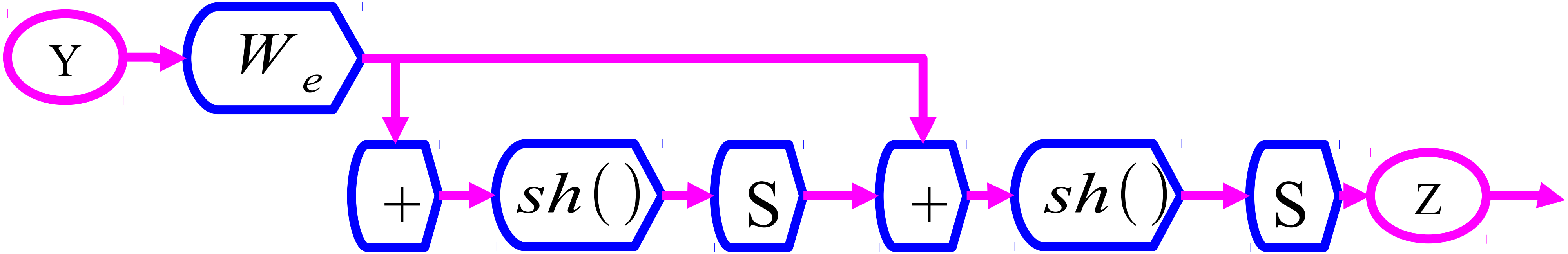
[Gregor & LeCun, ICML 2010], [A. Bronstein et al. ICML 2012], [Rolfe & LeCun ICLR 2013]

LISTA: Train We and S matrices to give a good approximation quickly

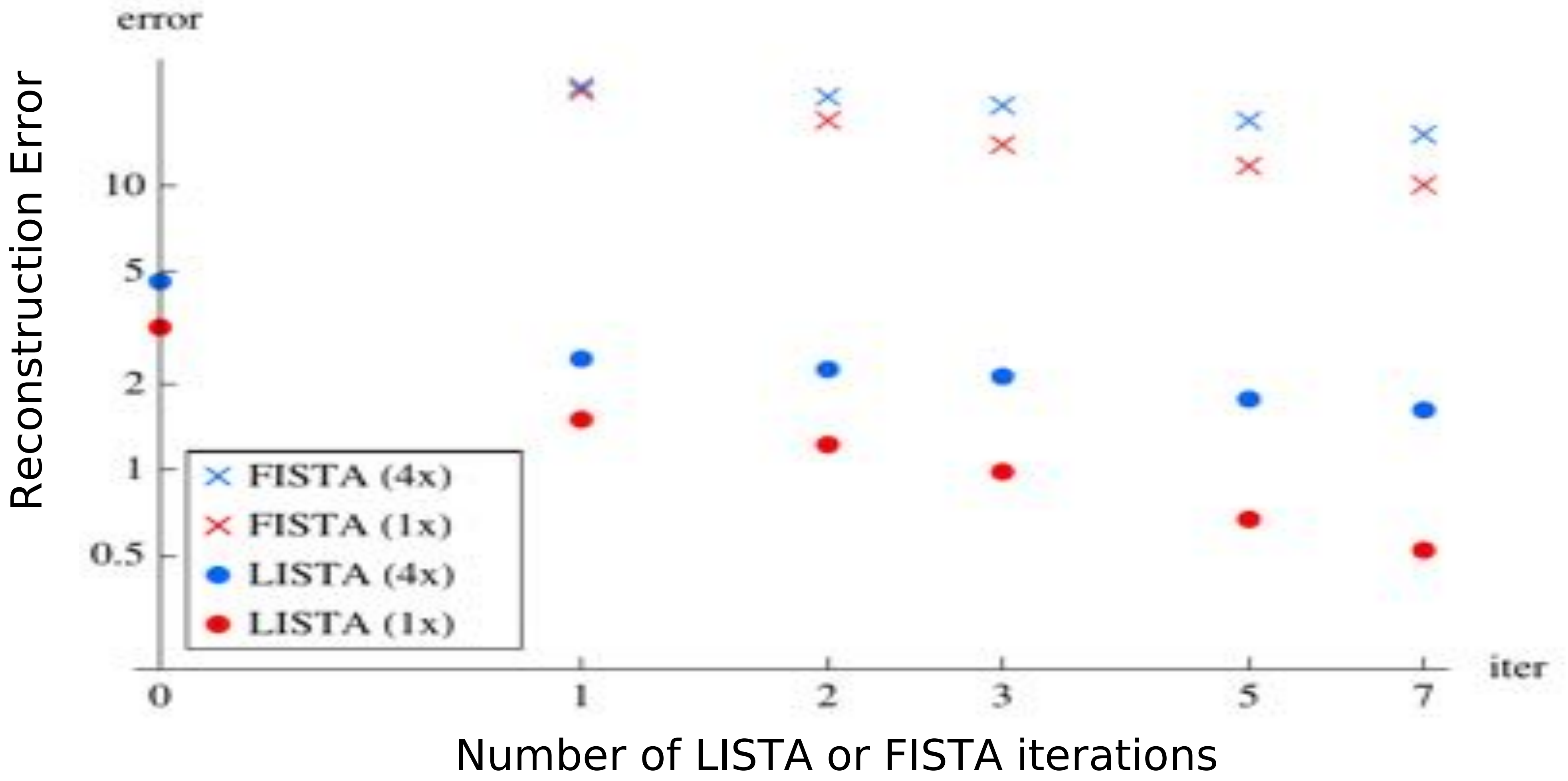
- **Think of the FISTA flow graph as a recurrent neural net where We and S are trainable parameters**



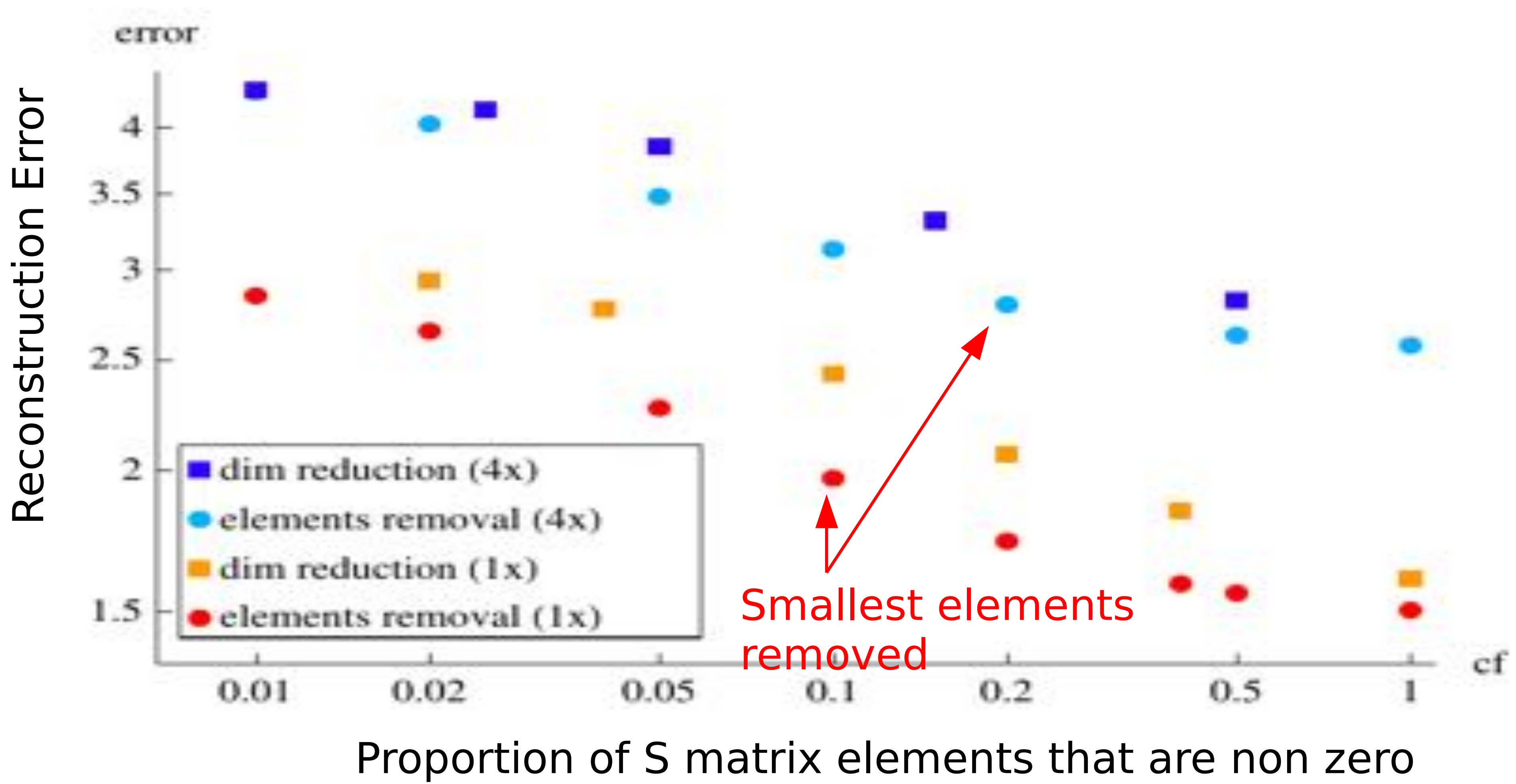
- **Time-Unfold the flow graph for K iterations**
- **Learn the We and S matrices with “backprop-through-time”**
- **Get the best approximate solution within K iterations**



Learning ISTA (LISTA) vs ISTA/FISTA

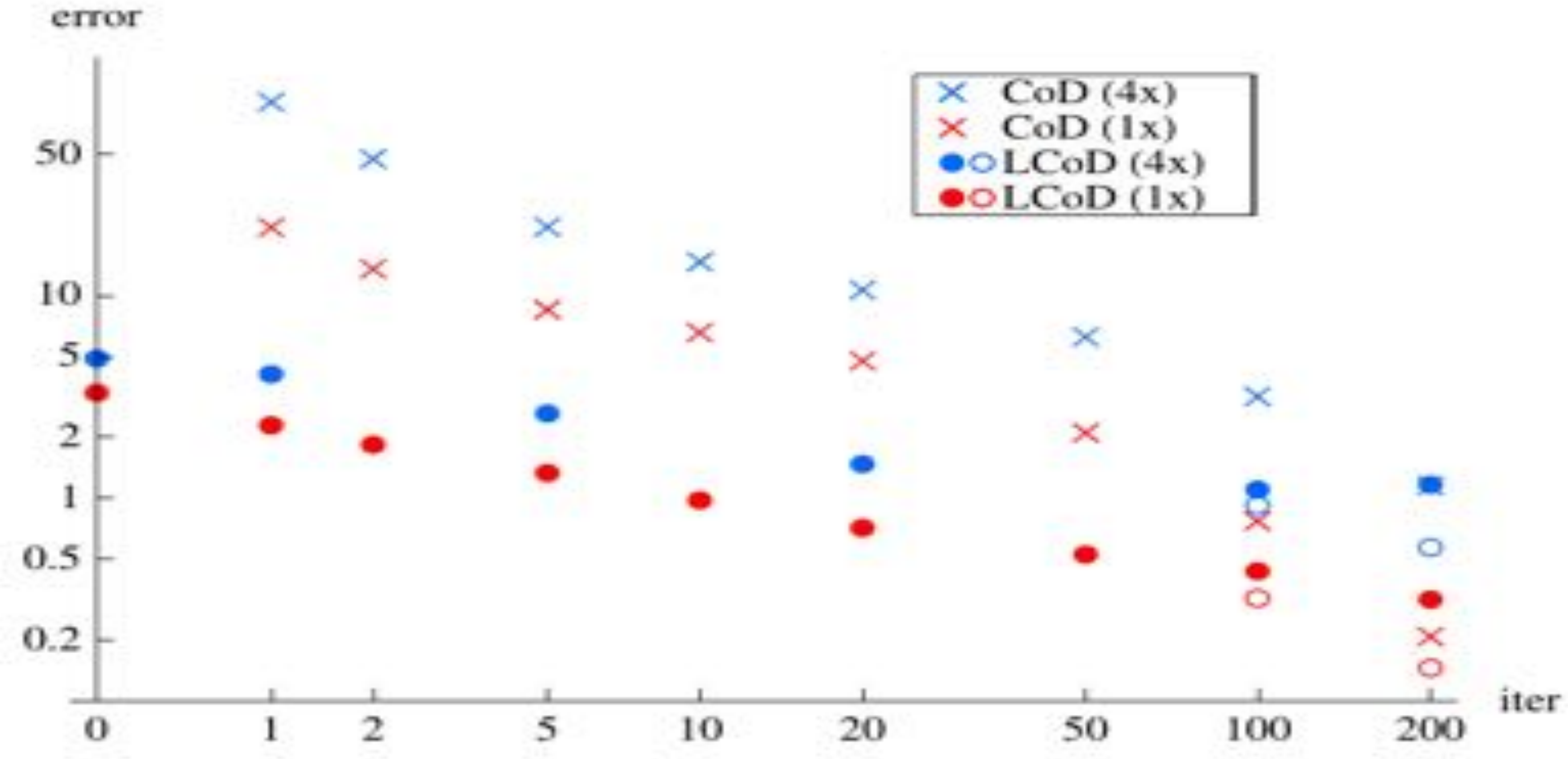


LISTA with partial mutual inhibition matrix



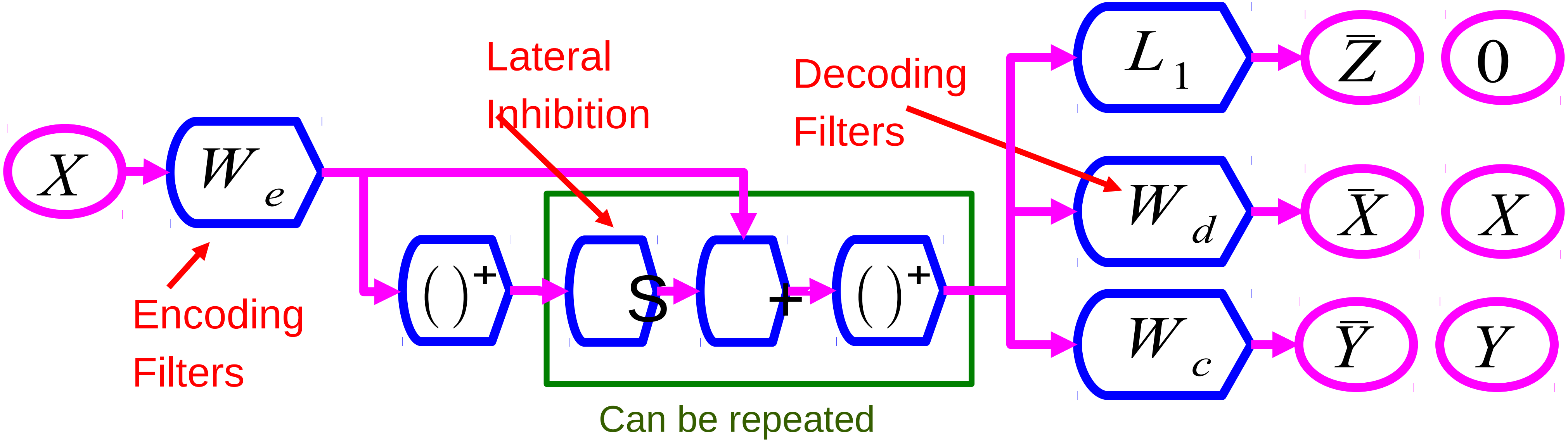
Learning Coordinate Descent (LcoD): faster than LISTA

Reconstruction Error



Number of LISTA or FISTA iterations

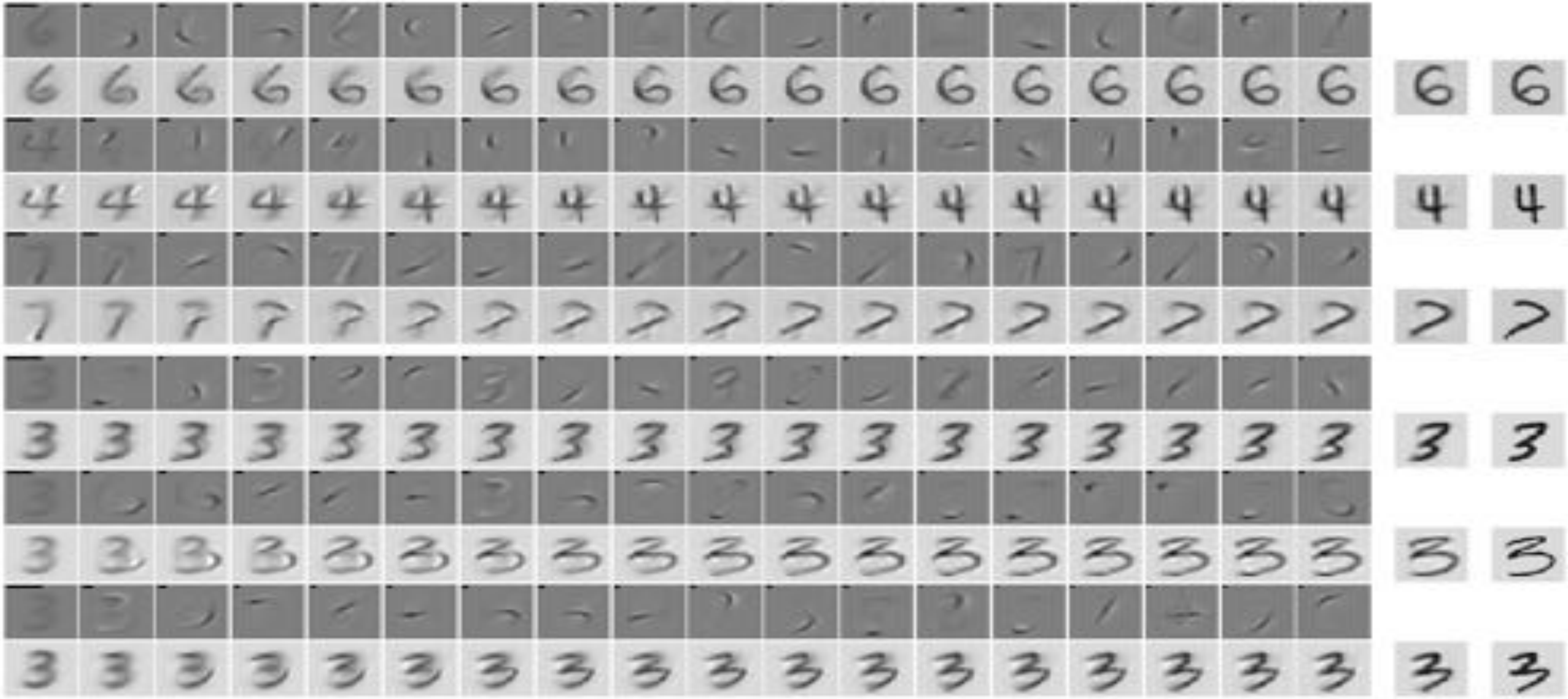
Discriminative Recurrent Sparse Auto-Encoder (DrSAE)



[Rolfe & LeCun ICLR 2013]

DrSAE Discovers manifold structure of handwritten digits

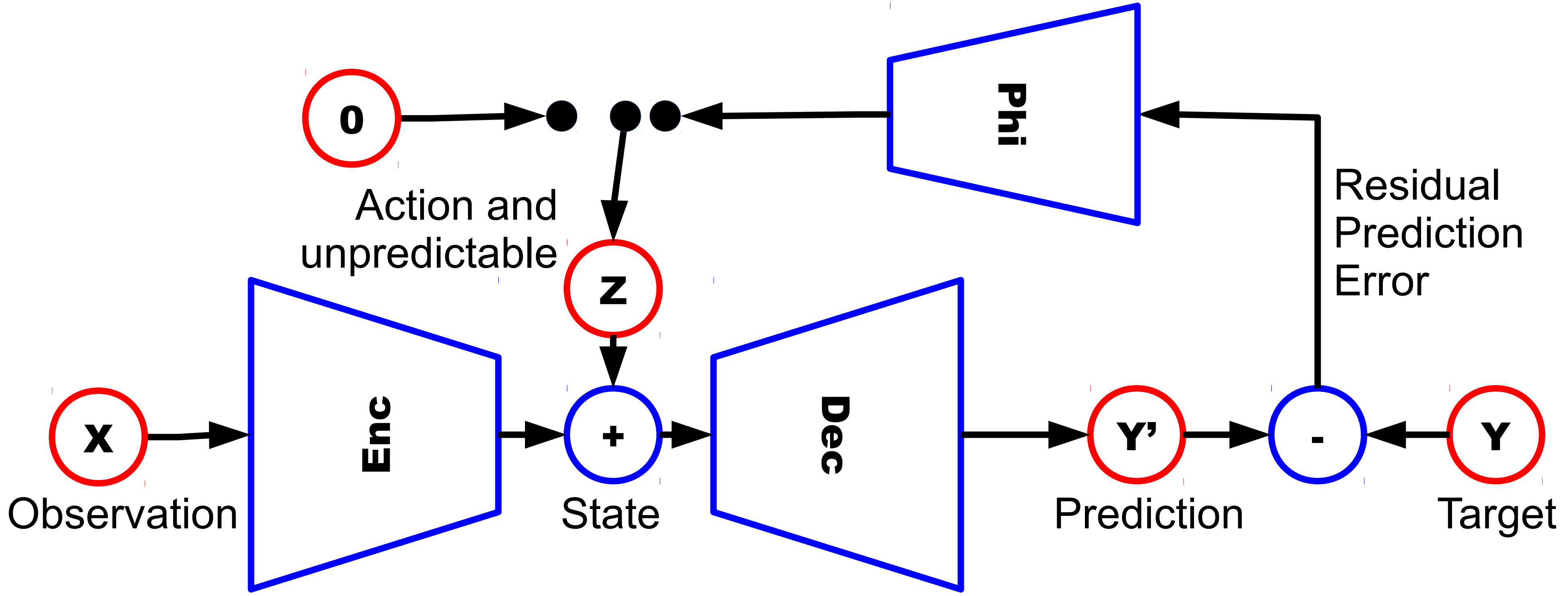
- Image = prototype + sparse sum of “parts” (to move around the



Error Encoding Networks

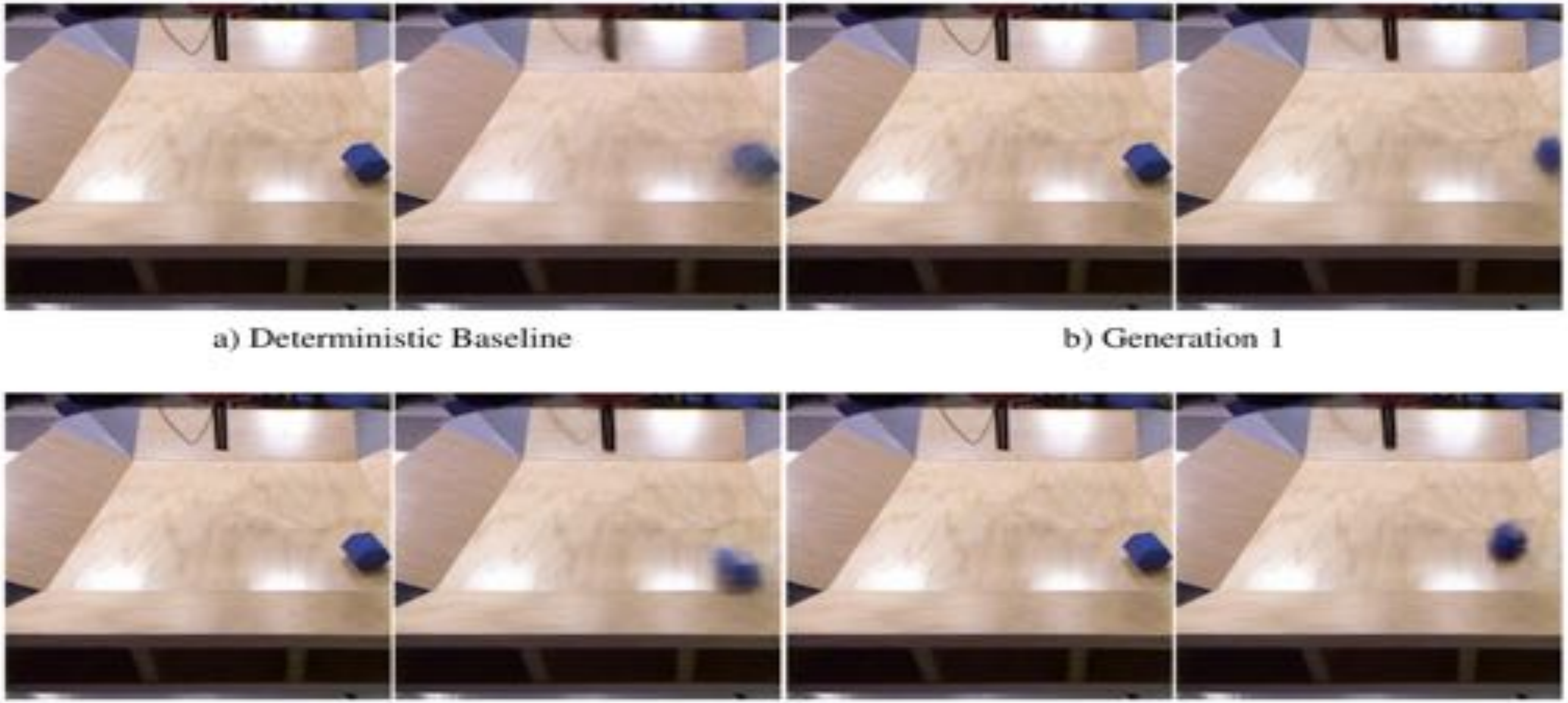
Error Encoding Network:
Forward model that infers actions & unpredictable latent variables

- [Henaff, Zhao, LeCun ArXiv:1711.04994]
- Y' = Dec(Enc(X) + Z)   with Z=0 or   Z = Phi(Y-Y')



Forward model that infers the action

- Trained to predict the position of an object after being poked by a robot arm [Agrawal et al.NIPS 2016]
- Latent variable contains result of arm movement



a) Deterministic Baseline b) Generation 1

c) Generation 2 d) Generation 3

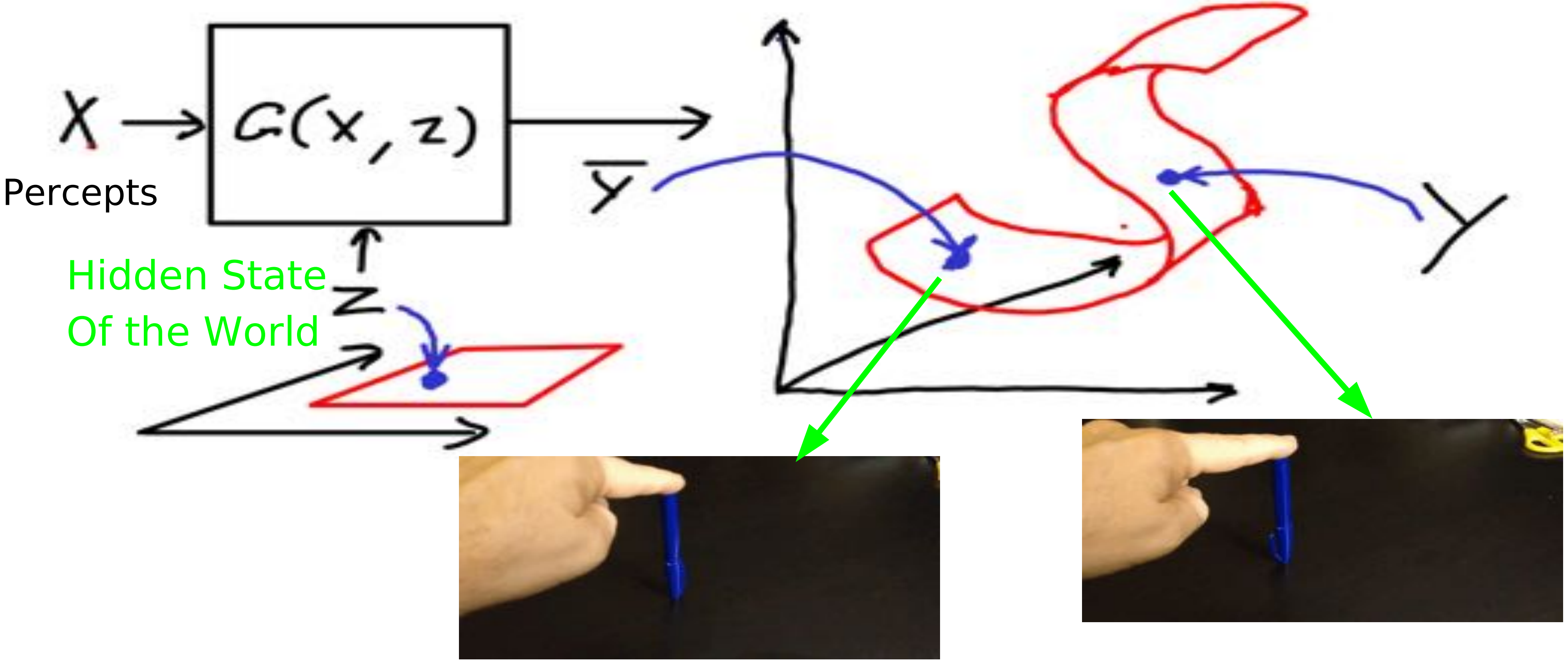
Forward model that infers the action

- Video: predictions as Z varies

Adversarial Training

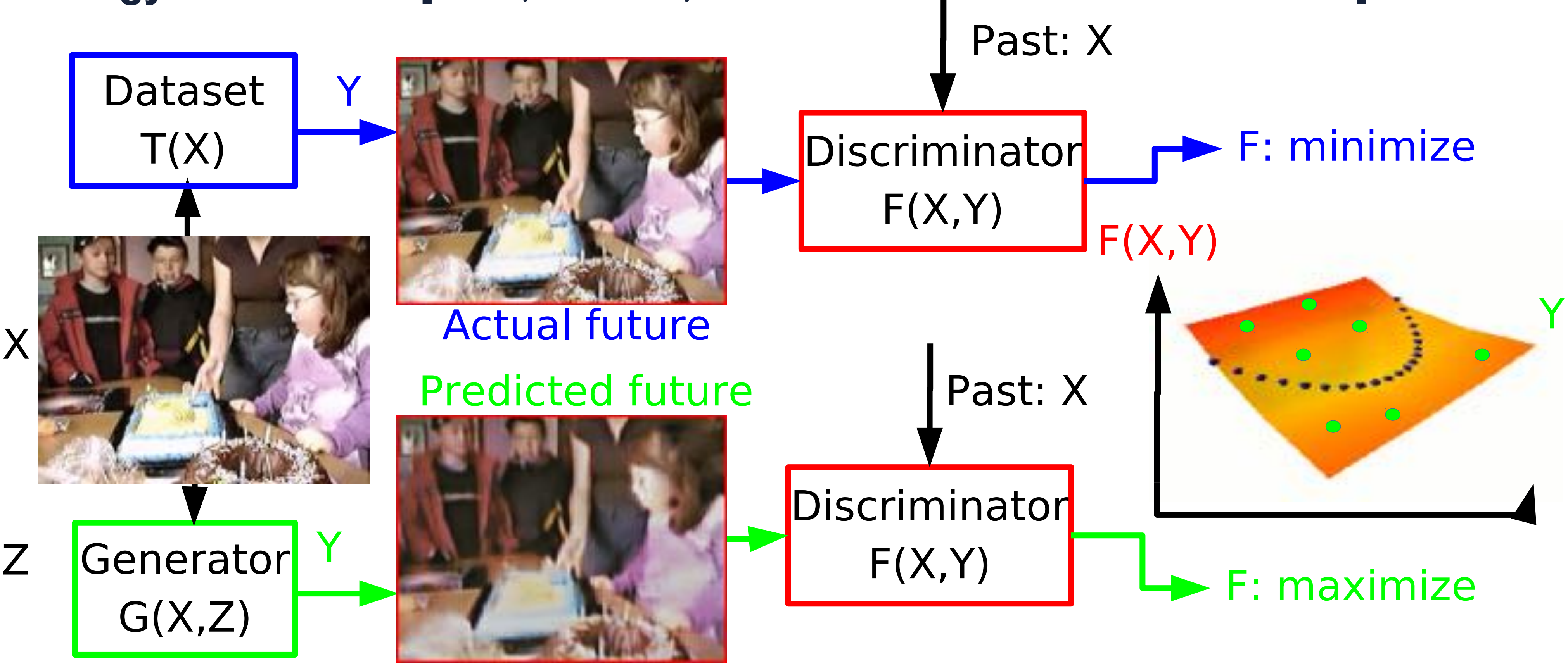
Predicting under Uncertainty: Adversarial Training

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



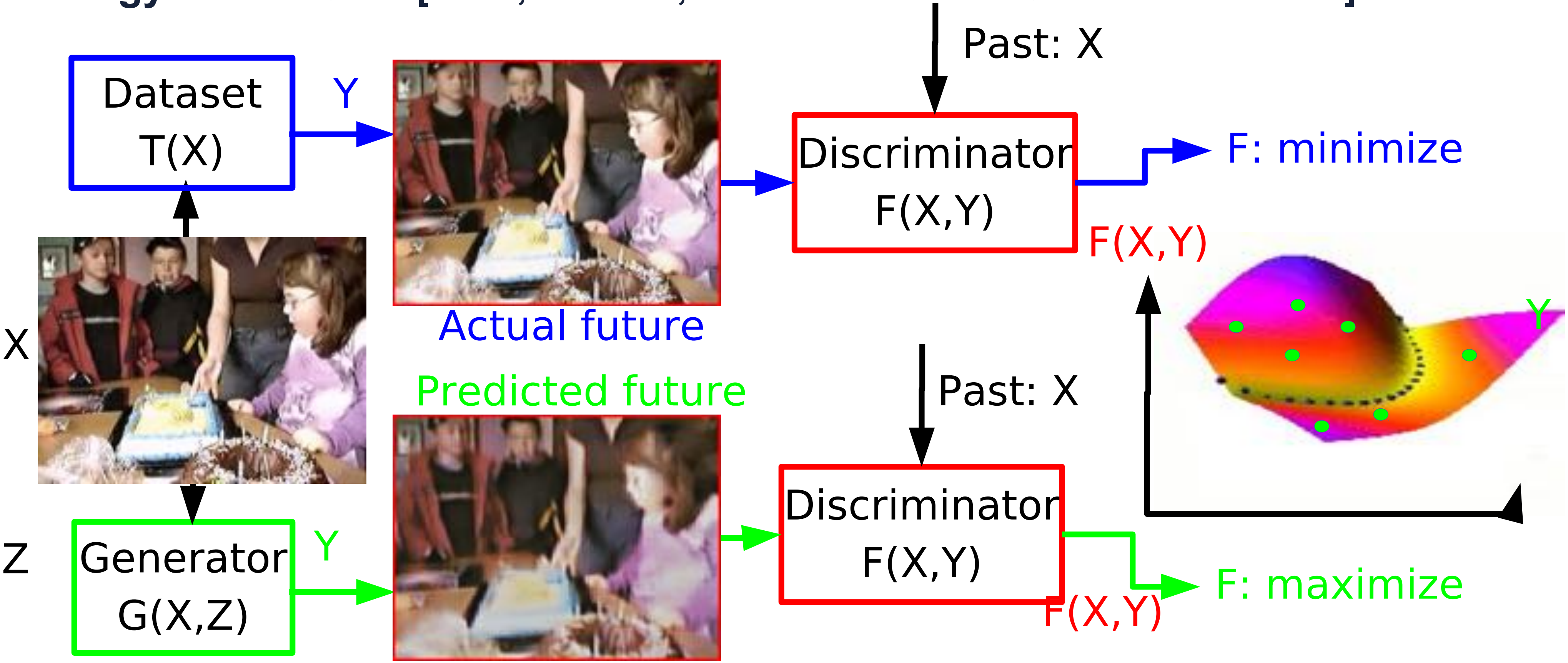
Adversarial Training: the key to prediction under uncertainty?

- **Generative Adversarial Networks (GAN) [Goodfellow et al. NIPS 2014],**
- **Energy-Based GAN [Zhao, Mathieu, LeCun ICLR 2017 & arXiv:1609.03126]**



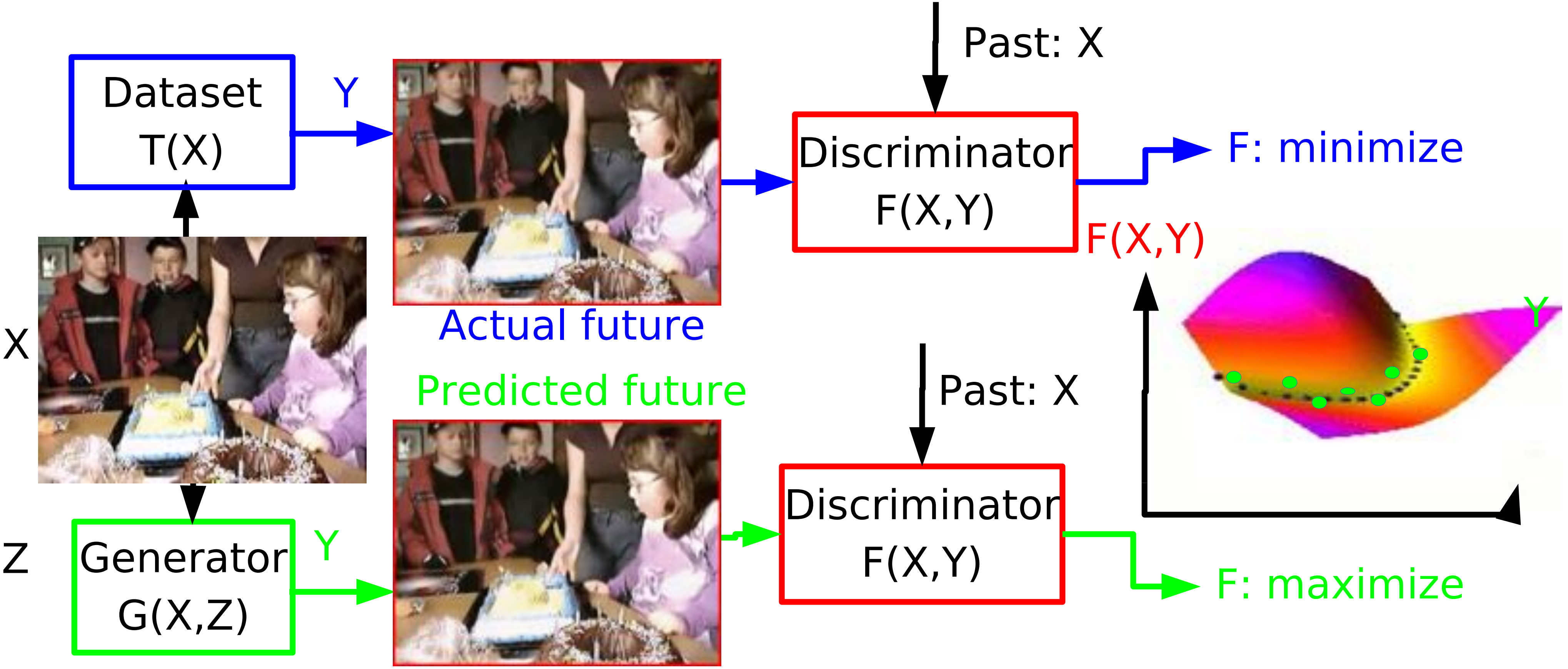
Adversarial Training: the key to prediction under uncertainty?

- **Generative Adversarial Networks (GAN)** [Goodfellow et al. NIPS 2014],
- **Energy-Based GAN** [Zhao, Mathieu, LeCun ICLR 2017 & arXiv:1609.03126]



Adversarial Training: the key to prediction under uncertainty?

- **Generative Adversarial Networks (GAN) [Goodfellow et al. NIPS 2014],**
- **Energy-Based GAN [Zhao, Mathieu, LeCun ICLR 2017 & arXiv:1609.03126]**



DCGAN: “reverse” ConvNet maps random vectors to images

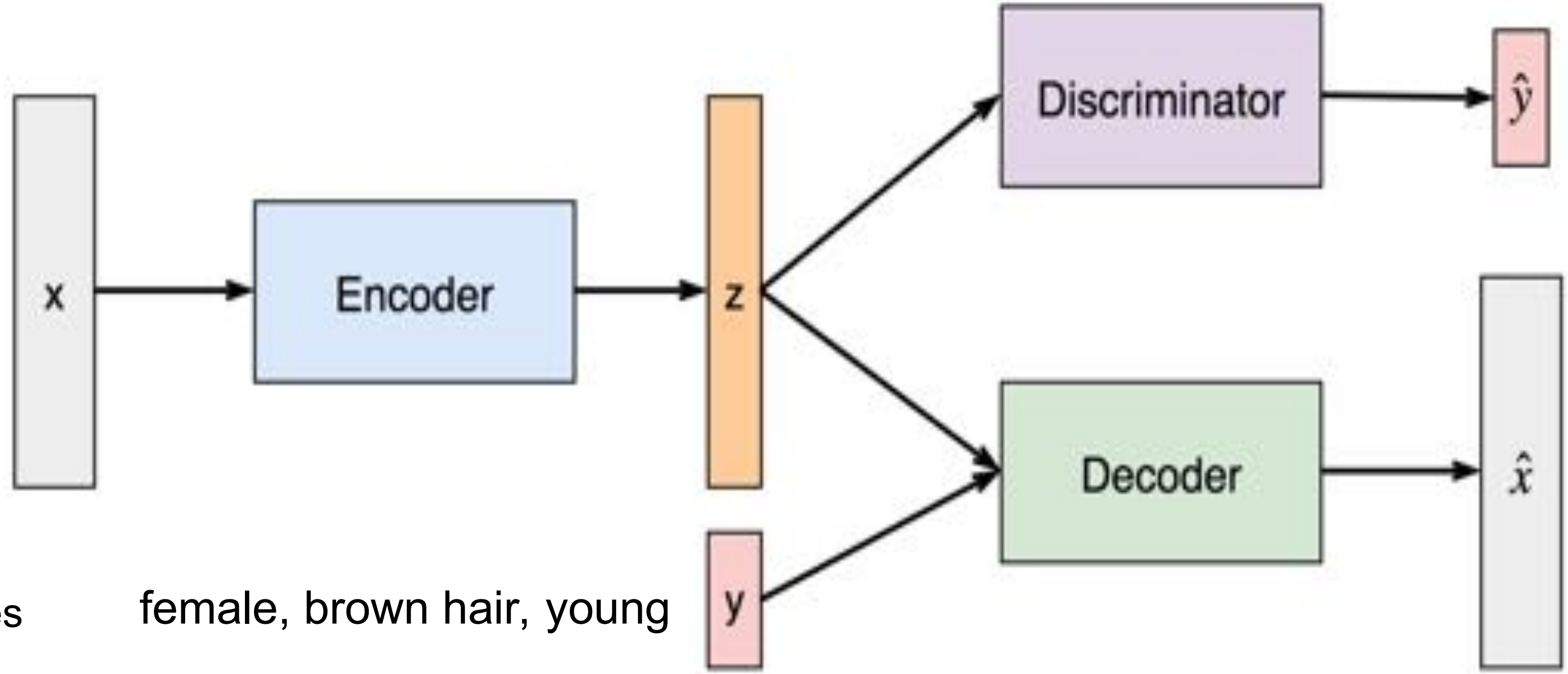
- **DCGAN: adversarial training to generate images.**
- **[Radford, Metz, Chintala 2015]**
  - Input: random numbers; output: bedrooms.

Faces “invented” by a neural net (from NVIDIA)

- **From random numbers [Karras et al. ICLR 2018]**

Fader Network: Auto-Encoder with two-part code

- **[Lample, Zeghidour, Usunier, Bordes, Denoyer, Ranzato arXiv:1706.00409]**
- **Discriminator trains Encoder to remove attribute information Y from code Z**
  - Discriminator trained (supervised) to predict attributes.
  - Encoder trained to prevent discriminator from predicting attributes



Training data:
images with attributes

female, brown hair, young

Varying Attributes

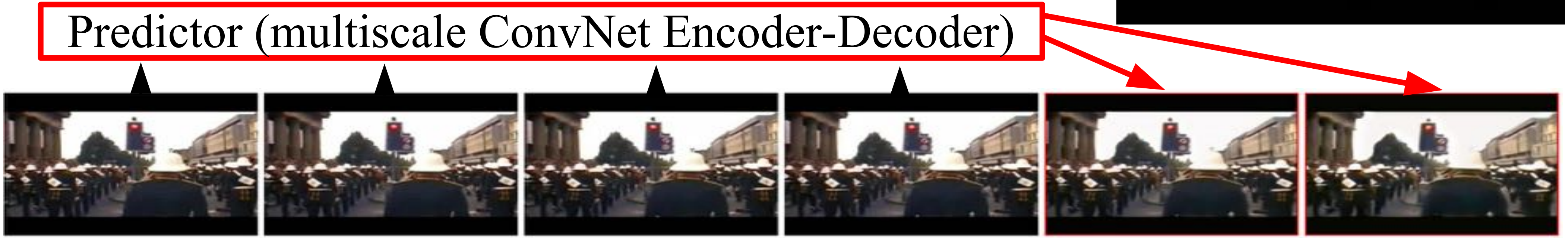
- **Young to old and back, male to female and back**

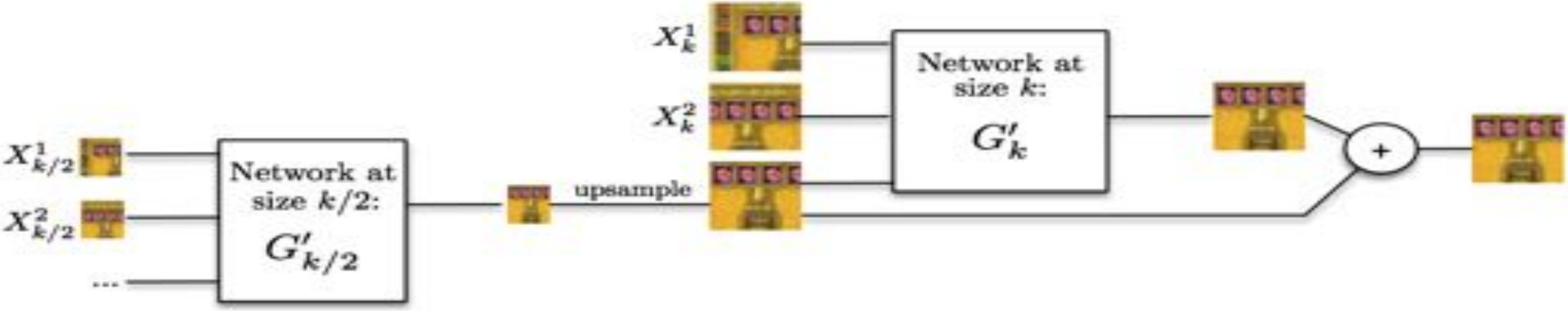
Video Prediction with Adversarial Training

[Mathieu, Couprie, LeCun ICLR 2016]
arXiv:1511:05440

Multi-Scale ConvNet for Video Prediction

- **4 to 8 frames input → ConvNet → 1 to 8 frames out**
- **Multi-scale ConvNet, without pooling**
- **If trained with least square: blurry output**





Predictive Unsupervised Learning

- **Our brains are “prediction machines”**
- **Can we train machines to predict the future?**
- **Some success with “adversarial training”**
  - [Mathieu, Couprie, LeCun arXiv:1511:05440]
- **But we are far from a complete solution.**







Video Prediction: predicting 5 frames

Video Prediction in Semantic Segmentation Space

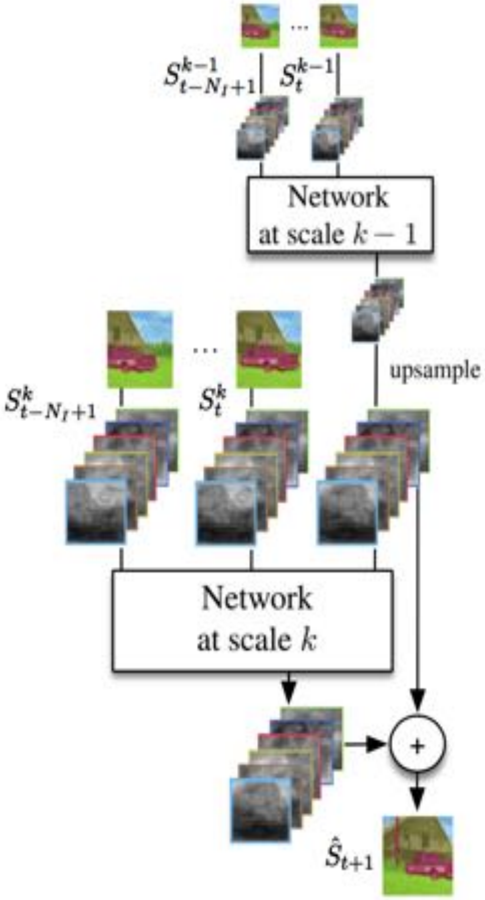
[Luc, Neverova, Couprie, Verbeek,
& LeCun ICCV 2017]

facebook
Artificial Intelligence Research

Temporal Predictions of Semantic Segmentations

- **Predictions a single future frame**
  - CityScape dataset [Cordt et al. CVPR 2016]

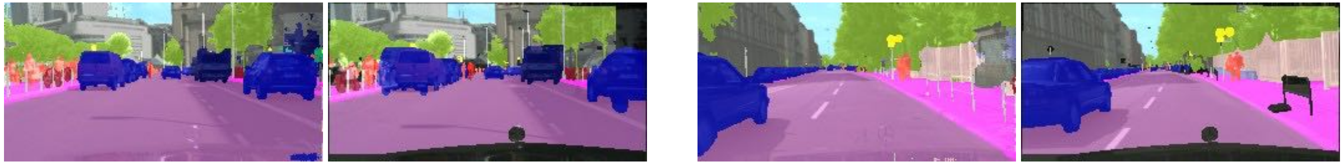
Method	PSNR	SSIM	IoU GT	IoU SEG	IoU-MO GT	IoU-MO SEG
Copy last input	20.6	0.65	49.4	54.6	43.4	48.2
Warp last input	20.9	0.67	50.4	55.5	44.9	49.8
Model X2X	24.0	**0.77**	23.0	22.3	12.8	11.4
Model S2S	—	—	**58.3**	64.9	53.8	59.8
Model S2S-adv.	—	—	**58.3**	**65.0**	**53.9**	**60.2**
Model XS2X	**24.2**	**0.77**	22.4	22.5	10.8	10.0
Model XS2S	—	—	58.2	64.6	53.7	59.9
Model XS2XS	24.0	0.76	55.5	61.1	50.7	55.8



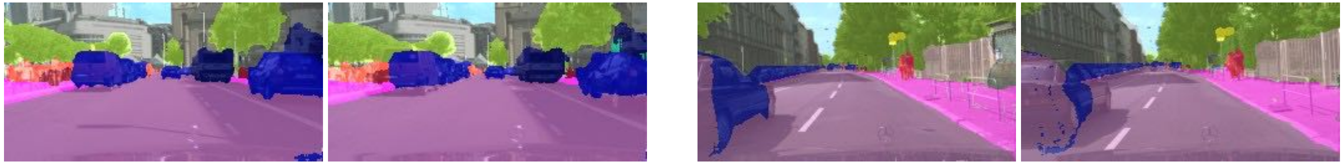
2: Multi-scale architecture of the S2S model th

Temporal Predictions of Semantic Segmentations

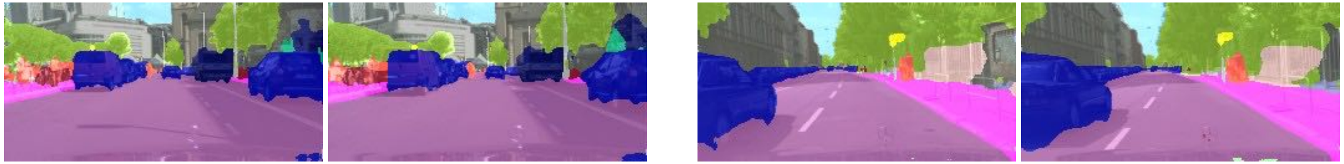
- **Prediction 9 frames ahead (0.5 seconds)**
- **Auto-regressive model**



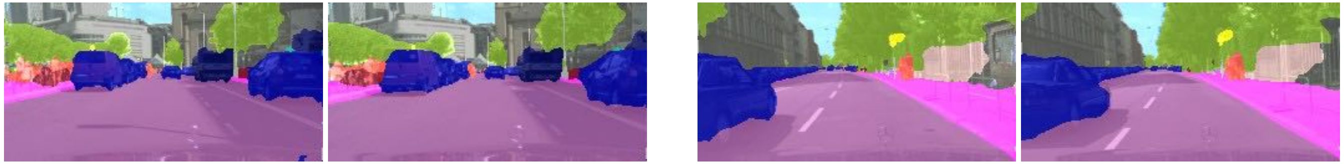
X_t, S_t — X_{t+9}, GT — X_t, S_t — X_{t+9}, GT



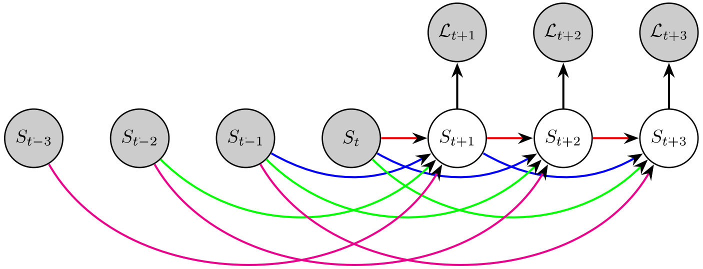
Batch predictions at $t+3$ — at $t+9$ — Optical flow at $t+3$ — at $t+9$



Autoregressive pred. at $t+3$ — at $t+9$ — Autor. adv. pred. at $t+3$ — at $t+9$



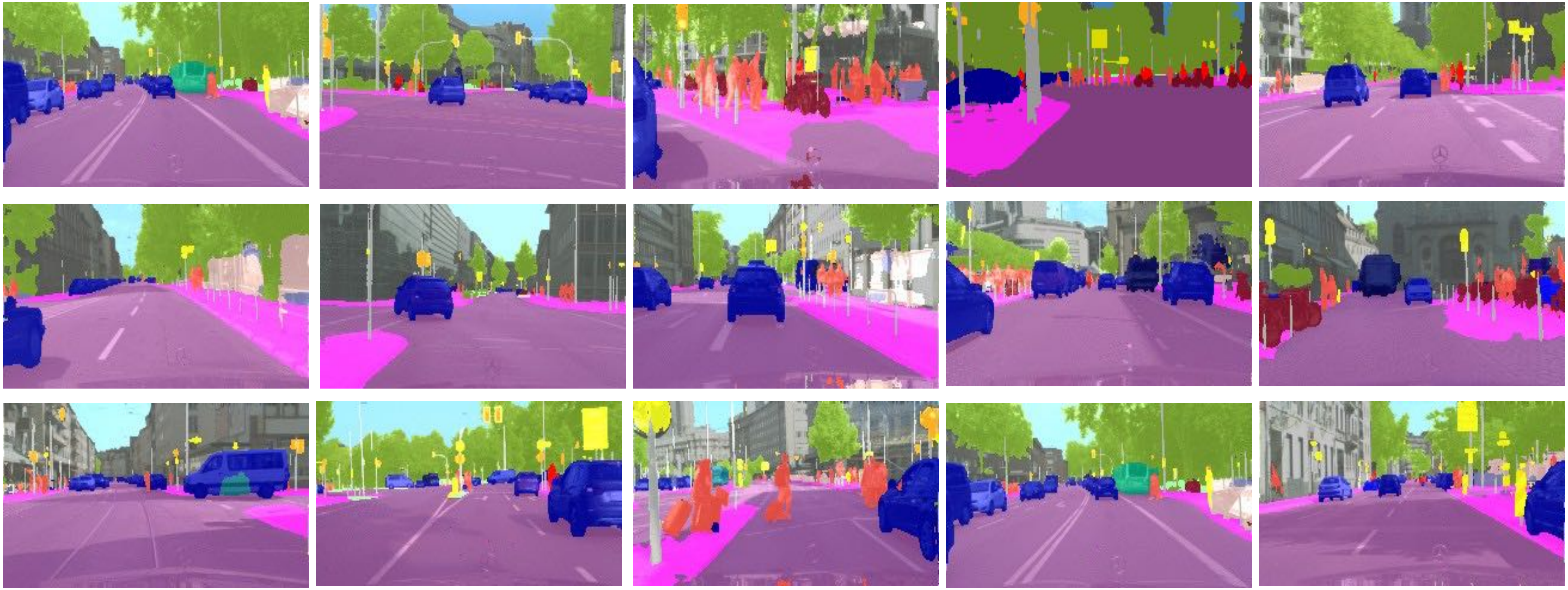
AR fine-tune pred. at $t+3$ — at $t+9$ — AR fine-tune pred. at $t+3$ — at $t+9$



Model	IoU GT	IoU SEG	IoU-MO GT
Copy last input	36.9	39.2	26.8
Warp last input	37.5	39.5	27.9
S2S, AR	45.3	47.2	36.4
S2S-adv, AR	45.1	47.2	37.3
S2S, AR, fine-tune	**46.7**	**49.7**	**39.3**
XS2XS, AR	39.3	40.8	27.4
S2S, batch	42.1	44.2	32.8
XS2S, batch	42.3	44.6	33.1
XS2XS, batch	41.2	43.5	31.4

Temporal Predictions of Semantic Segmentations

- **Prediction 9 frames ahead (0.5 seconds)**
- **Auto-regressive model**



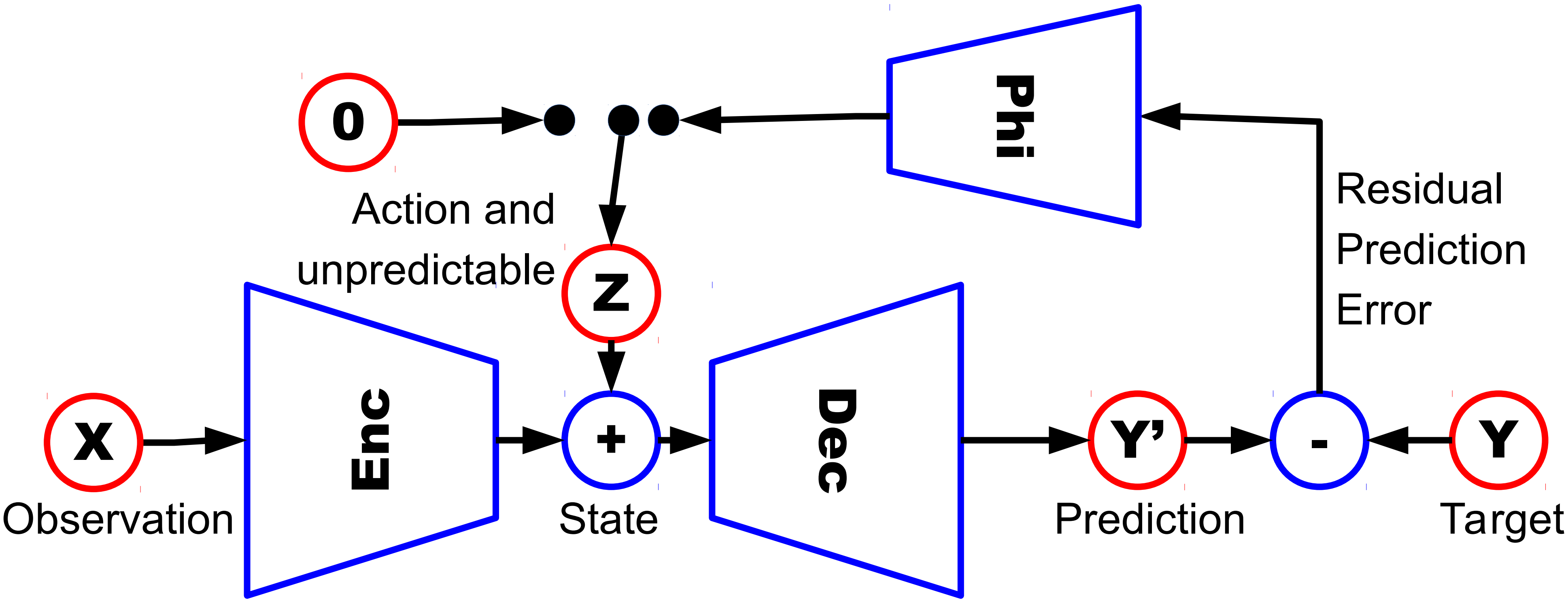
Trained Forward Models for Planning and Learning Skills

[Henaff, Zhao, LeCun ArXiv:1711.04994]
[Henaff, Whitney, LeCun Arxiv:1705.07177]

facebook
Artificial Intelligence Research

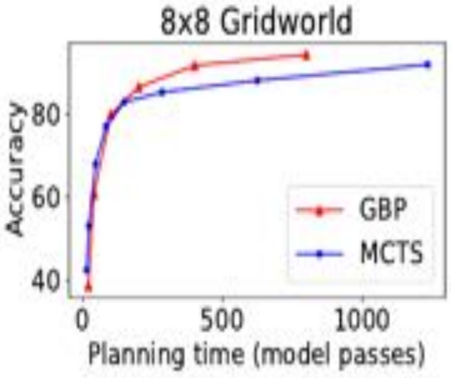
Error Encoding Network:
Forward model that infers actions & unpredictable latent variables.

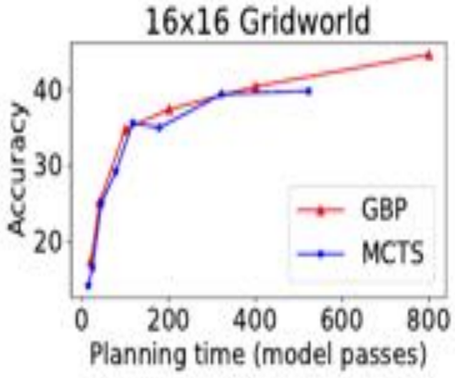
- [Henaff, Zhao, LeCun ArXiv:1711.04994]
- Y’ = Dec(Enc(X) + Z)   with Z=0 or   Z = Phi(Y-Y’)

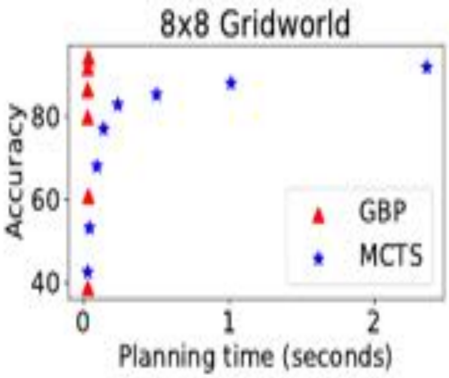


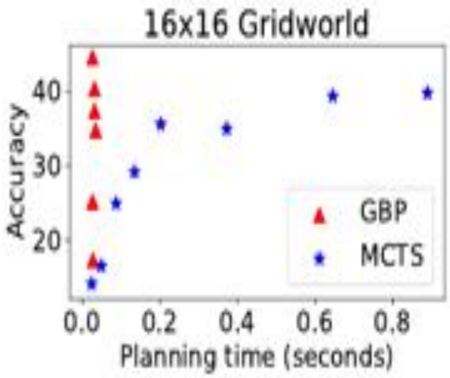
Grid world with forward model

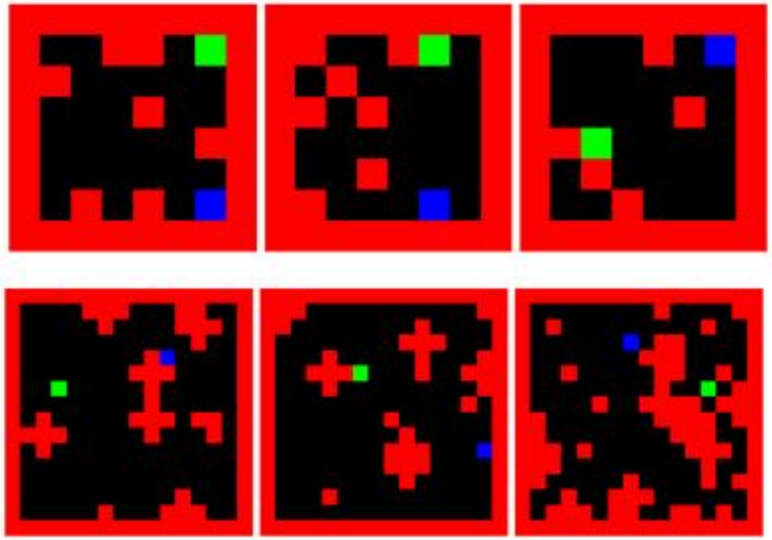
- **Grid world**







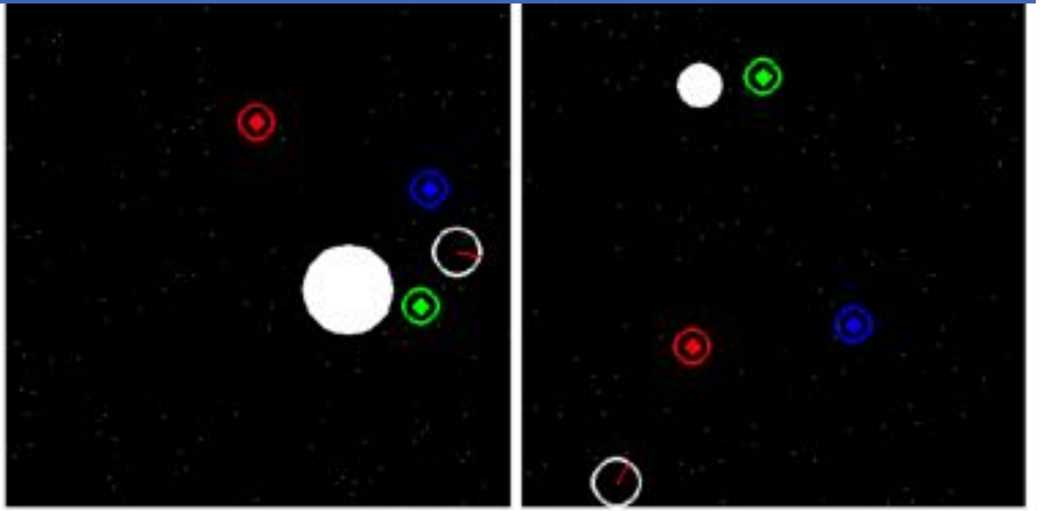




MAP	METHOD	ACC.	TIME (S)	ENV. STEPS
8×8	TRPO*	86.9	< 0.001	-
	TRPO (OURS)	82.4	< 0.001	22M
	MCTS (R=2000)	91.8	2.36	54K
	GBP (R=40)	94.0	0.03	54K
	DISTGBP	91.4	< 0.001	54K
	GBP (NO NOISE)	25.6	0.03	54K
16×16	TRPO*	33.1	< 0.001	3M
	MCTS (R=2000)	39.8	0.90	110K
	GBP (R=2000)	66.4	0.51	110K
	DISTGBP	52.6	< 0.001	110K
	GBP (NO NOISE)	07.8	0.51	110K

Spaceship control

- Planet with gravity, targets,
- Ship with orientable thruster

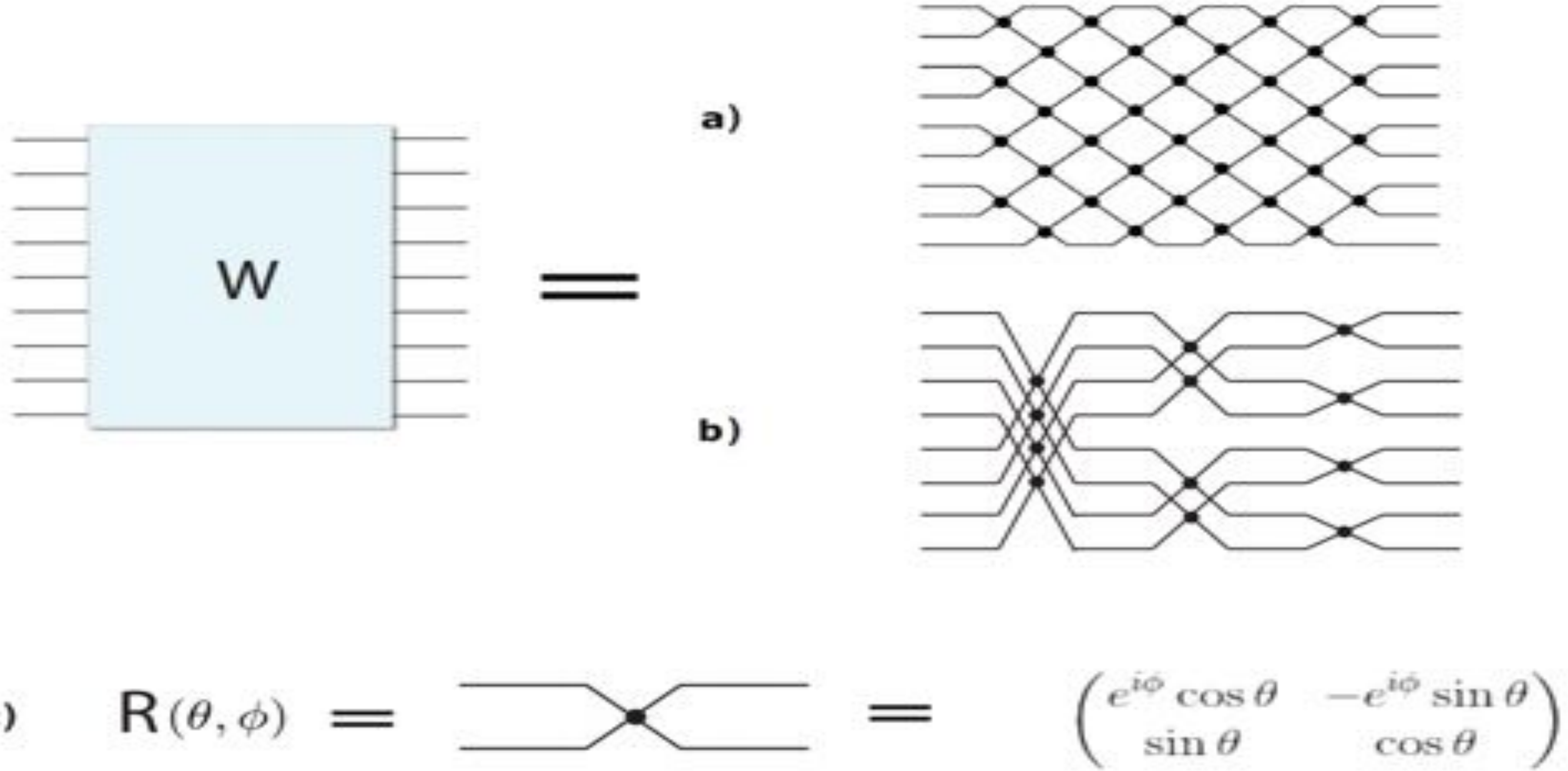


METHOD	AVERAGE REWARD	TIME (S)	ENV. STEPS
RANDOM	-62.7	-	0
A2C	-19.2	0.01	3.8M
GBP	11.1	0.19	800K
DISTGBP	12.2	0.01	800K

Reversible Recurrent Nets

Approximating a unitary matrix with a product of Givens transforms

- **"Linearithmic" approximations of unitary matrices**



Linearithmic Hessian Matrix Learning (with SGD)

- **Least square with x a random vector, and y the "real" product Hx**

$$H \approx Q_1 Q_2 \ldots Q_{\lg(n)} D Q_{\lg(n)}^T \ldots Q_2^T Q_1^T$$

$$L(\omega, x^{(j)}, y^{(j)}) = ||Q_\omega D_\omega Q_\omega^T x^{(j)} - y^{(j)}||_2^2$$

Algorithm 1 Hessian matrix learning

Input: set $(x^{(j)}, y^{(j)})$ for $j = 1..m$
while not converged **do**
  Randomly draw $j \in 1..m$
  Compute gradient $g_{\tilde{\omega},j} = \frac{\partial L(\tilde{\omega}, x^{(j)}, y^{(j)})}{\partial \tilde{\omega}}$
  Update $\tilde{\omega} \leftarrow \tilde{\omega} - \alpha g_{\tilde{\omega},j}$
  Normalize all the *Givens* to project $Q_{\tilde{\omega}}$ on Q
end while

Linearithmic Hessian Matrix Learning (with SGD)

- **Decompose the diagonalized Hessian H=QDQ'**
- **Decompose each Q into n.log n/2 elementary 2D rotations**
  - 2D rotations are organized on an FFT-like graph
  - Prod … joint pairs of varia

$$H \approx Q_1 Q_2 \dots Q_{\lg(n)} D Q_{\lg(n)}^T \dots Q_2^T Q_1^T$$

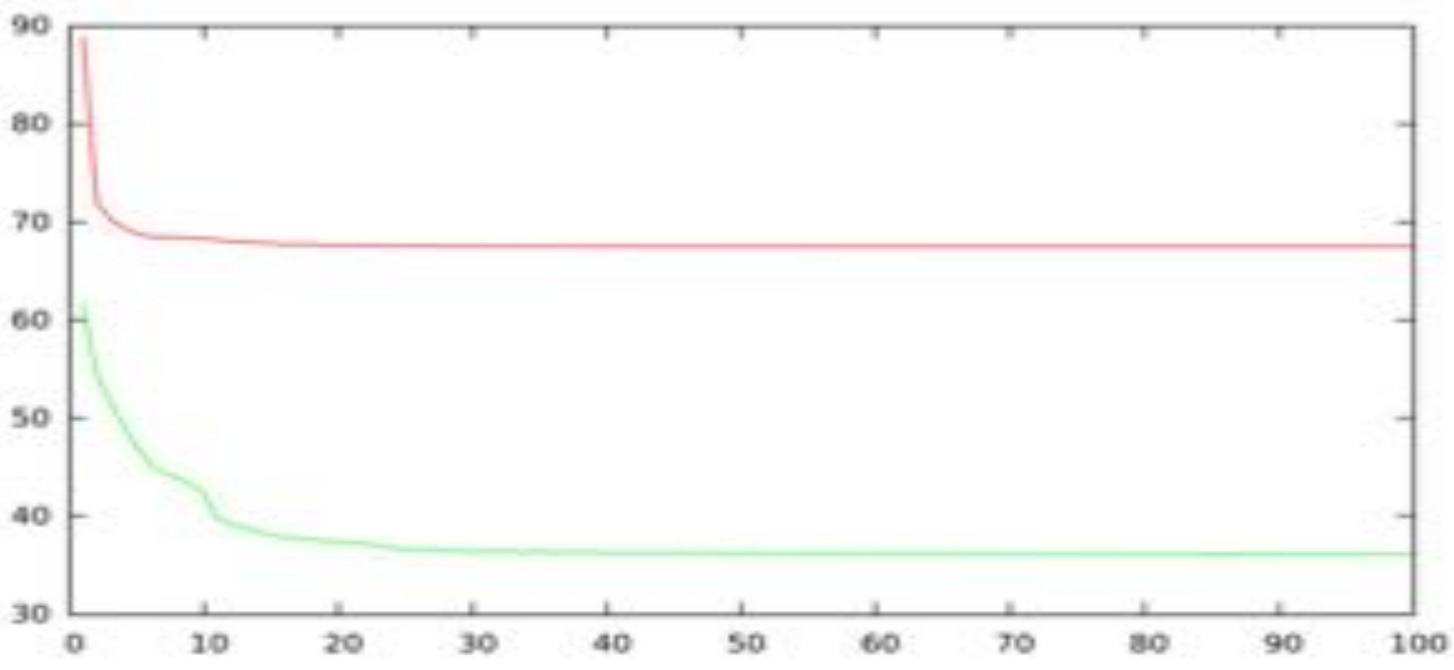
- **View the product Hx as a linear multilayer net.**
- **Minimize a least square error between the products of a set vectors by the real hessian y=Hx, and the product of the same vector by the approximate hessian QDQ'x**

$$L(\omega, x^{(j)}, y^{(j)}) = ||Q_\omega D_\omega Q_\omega^T x^{(j)} - y^{(j)}||_2^2$$

- **Train that with SGD, using backprop to compute the gradient through QDQ'**

Linearithmic Hessian Matrix Learning (with SGD)

- **Learning a random covariance matrix, dimension 64.**
  - Average angle between random vectors multiplied by the real matrix and multiplied by the approximation: 35 degrees.



Linearithmic Hessian

- **Learning with linearithmic hessian**
  - Lest square to solve: $H_{u_t}(u_t - u_{t-1}) \approx \nabla_{u_t}\ell - \nabla_{u_{t-1}}\ell$
  - u is parameter
  - Grad l is gradient

Algorithm 2 Optimization with linearithmic Hessian

Parameters: Learning rates α and β
Initialize $\widetilde{\omega}$ such that $D_{\widetilde{\omega}} = I$ and $Q_{\widetilde{\omega}} \in \mathcal{Q}$.
Initialize random u_0
Set $t = 0$
while not converged **do**
  Compute $\nabla_{u_t}\ell$
  if $t \neq 0$ **then**
    Set $\delta u = u_t - u_{t-1}$
    Set $\delta g = \nabla_t \ell - \nabla_{t-1}\ell$
    Update $\widetilde{\omega} \leftarrow \widetilde{\omega} - \alpha \frac{\partial L(\widetilde{\omega}, \delta u, \delta g)}{\partial \widetilde{\omega}}$
    Project $Q_{\widetilde{\omega}}$ on $\mathcal{Q}$ as in Equation 6
  end if
  Set $u_{t+1} = u_t - \beta \widehat{H}_{\widetilde{\omega}} \nabla_u \ell$
  Update $t \leftarrow t + 1$
end while

RNN parameterized with linearithmic unitary transforms

- **[Jing, Shen, Dubček, Peurifoy, Skirlo, LeCun, Tegmark, Soljačić ICML 2017 arXiv:1612.05231]**

Model	Time complexity of one online gradient step	number of parameters in the hidden matrix	Transition matrix search space
URNN	$\mathcal{O}(TN\log N)$	$\mathcal{O}(N)$	subspace of $\mathbf{U}(N)$
PURNN	$\mathcal{O}(TN^2+N^3)$	$\mathcal{O}(N^2)$	full space of $\mathbf{U}(N)$
EURNN (tunable style)	$\mathcal{O}(TNL)$	$\mathcal{O}(NL)$	tunable space of $\mathbf{U}(N)$
EURNN (FFT style)	$\mathcal{O}(TN\log N)$	$\mathcal{O}(N\log N)$	subspace of $\mathbf{U}(N)$

- **Predicting the next pixel on MNIST**

Model	hidden size (capacity)	number of parameters	validation accuracy	test accuracy
LSTM	80	16k	0.908	0.902
URNN	512	16k	**0.942**	0.933
PURNN	116	16k	0.922	0.921
EURNN (tunable style)	1024 (2)	13.3k	0.940	**0.937**
EURNN (FFT style)	512 (FFT)	9.0k	0.928	0.925

The Future Impact of AI

facebook
Artificial Intelligence Research

Technology drives & motivates Science (and vice versa)

- **Science drives technology, but technology also drives science**
- **Sciences are born from the study of technological artifacts**
  - Telescope → optics
  - Steam engine → thermodynamics
  - Airplane → aerodynamics
  - Calculators → computer science
  - Telecommunication → information theory
- **What is the equivalent of thermodynamics for intelligence?**
  - Are there underlying principles behind artificial and natural intelligence?
  - Are there simple principles behind learning?
  - Or is the brain a large collection of “hacks” produced by evolution?

Thank you

facebook
Artificial Intelligence Research