IJCAI-16

Microsoft Research Cambridge | Carnegie Mellon University | Stanford | FBK Fondazione Bruno Kessler

Deep Neural Decision Forests

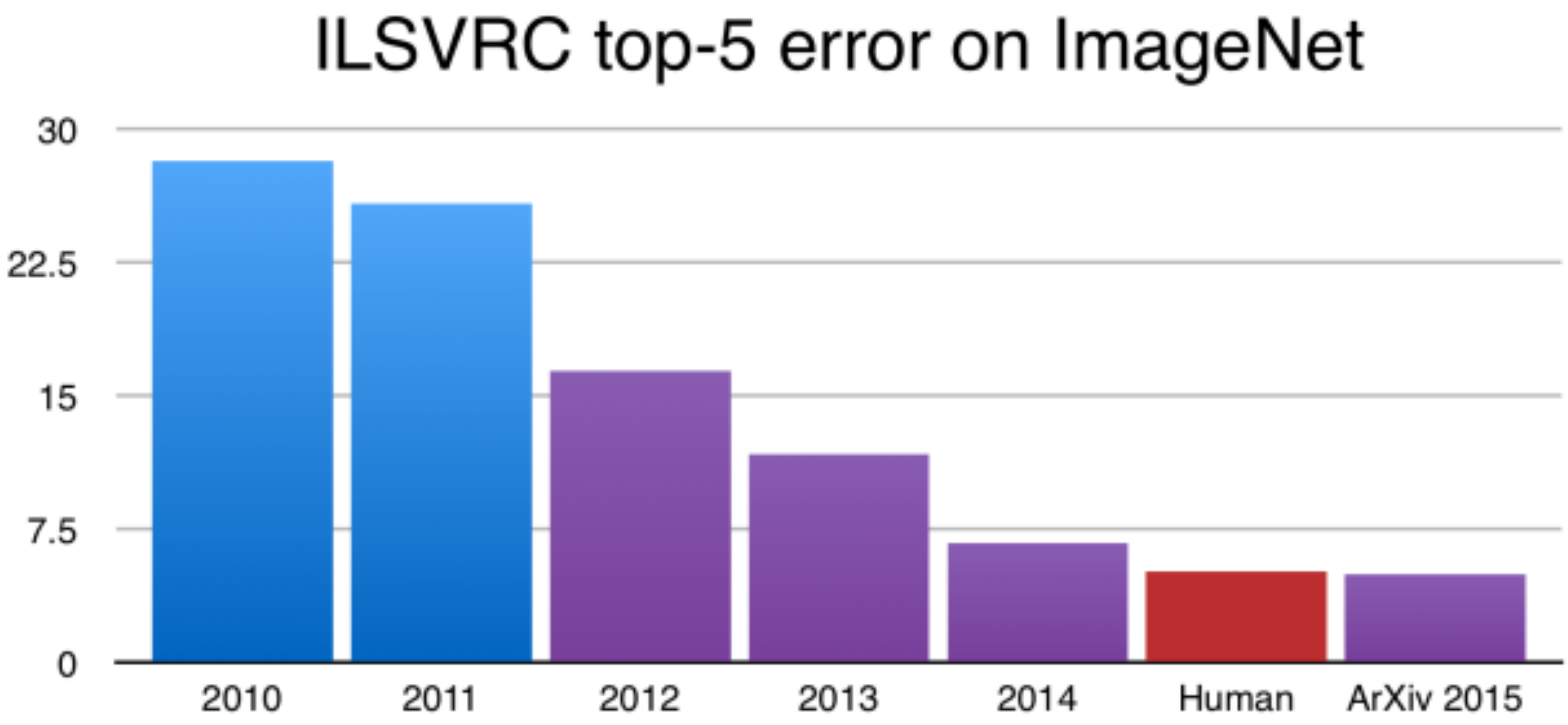
Peter Kontschieder, Microsoft Research
Madalina Fiterau, CMU/Stanford (presenter)
Antonio Criminisi, Microsoft Research
Samuel Rota-Bulò, Fondazione Bruno Kessler

ICCV15 International Conference on Computer Vision

Marr Prize Recipient

Deep Learning Performance

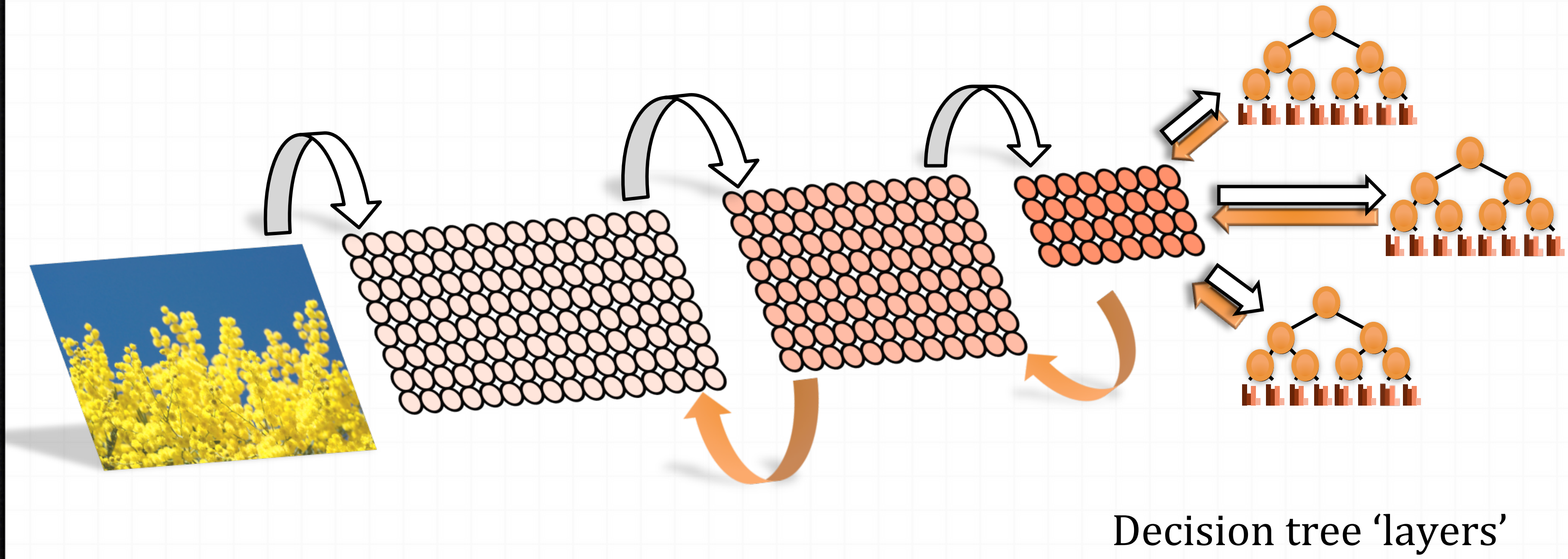
- Data complexity captured by representations
- Representations gradually become more sophisticated



Source: Chiyuan Zhang 2015

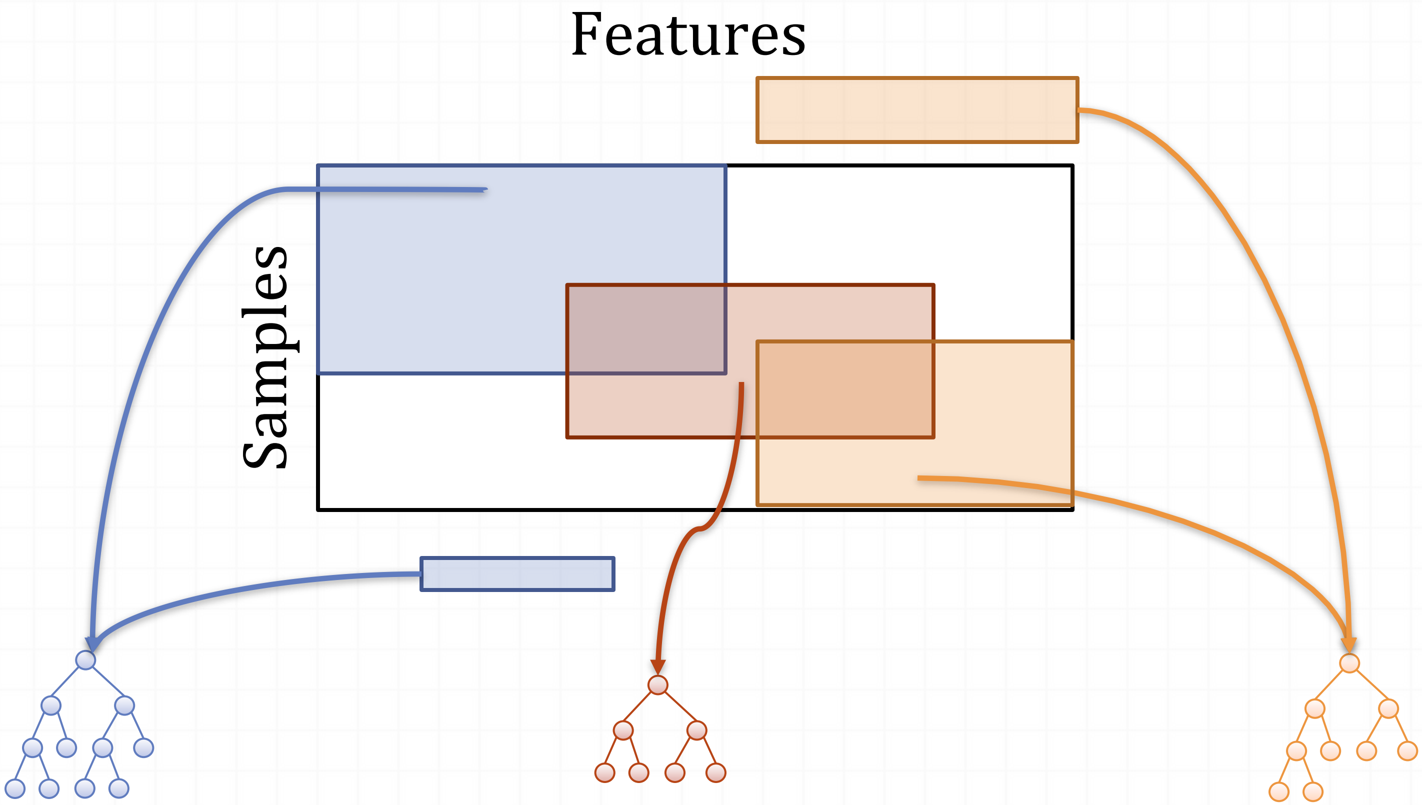
Deep Learning + Accurate Classifier

- Leverage representation learning via stacked conv. layers
- End-to-end deep learning architecture



Back-propagation Trees

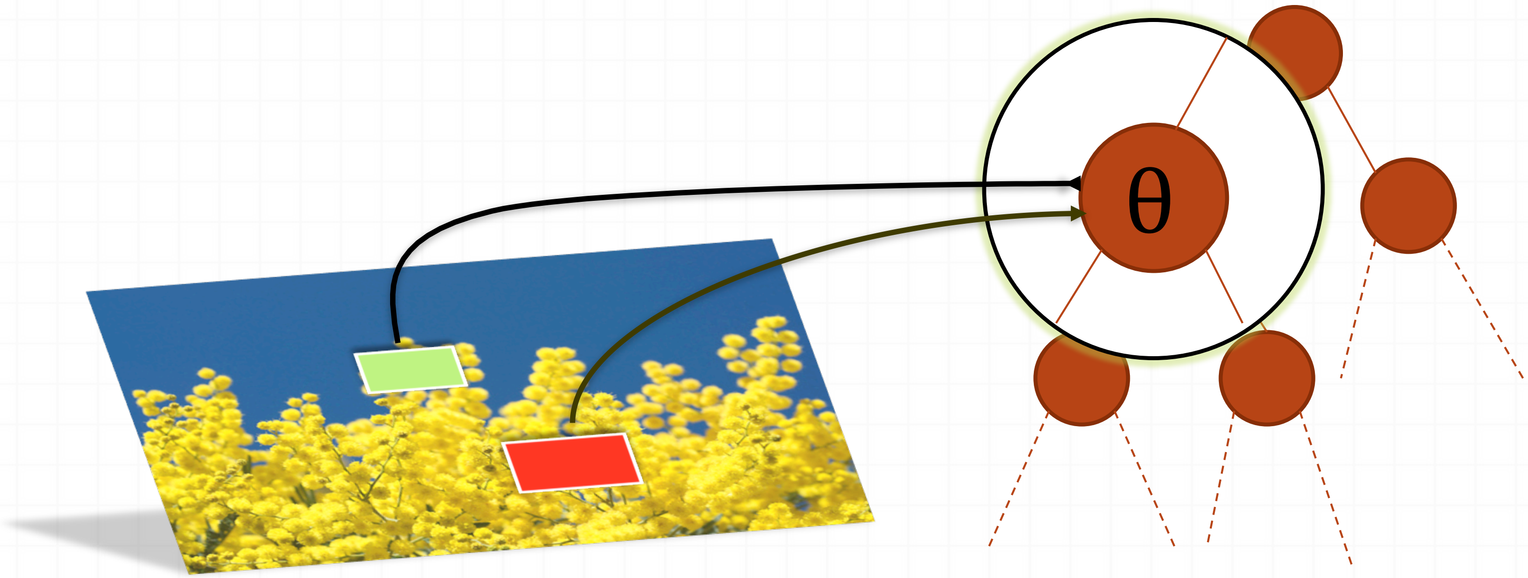
- Trees updated on-the-fly to allow data shift across submodels



Key: differentiable global loss

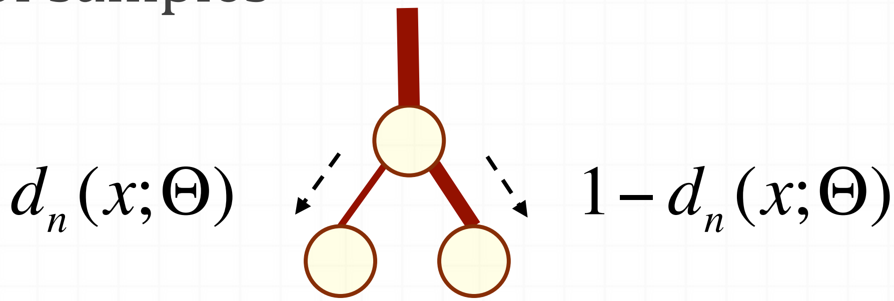
Back-propagation Trees

- Structure adapted to allow back propagation

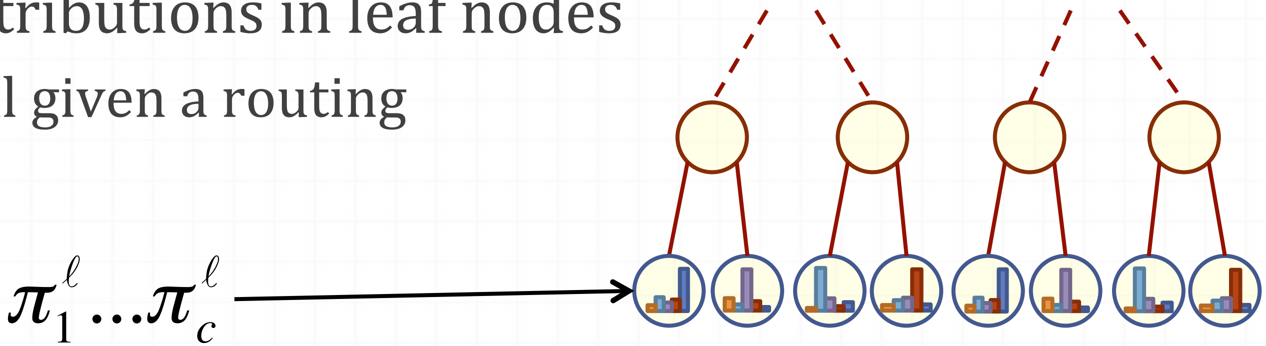


Back-propagation Trees

- Soft routing of samples



- Class distributions in leaf nodes
  - optimal given a routing



- Log loss Objective

Modeling Node Splits

- Hierarchical routing along path Φ_ℓ to leaf ℓ

$$\mu_\ell(x;\Theta) = \prod_{n \in \phi_\ell} d_n(x;\Theta)^{1_{\ell \leftarrow n}} (1 - d_n(x;\Theta))^{1_{n \to \ell}}$$

Sigmoid function

1 if ℓ belongs to left subtree of n

1 if ℓ belongs to right subtree of n

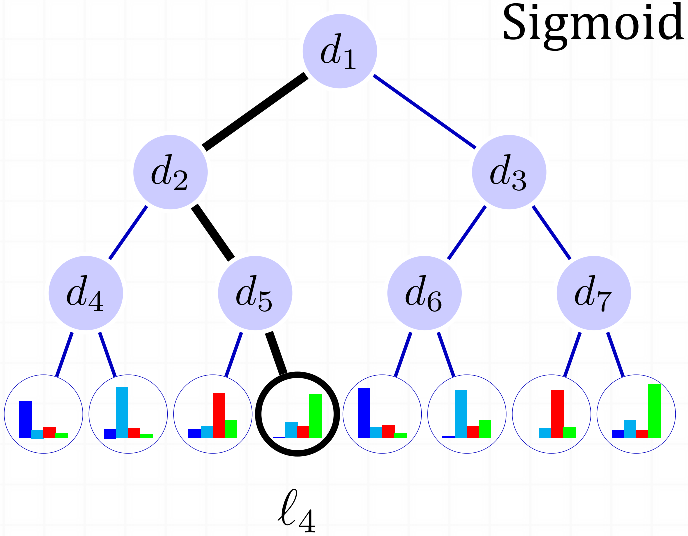


Image by Samuel Rota-Bulò

$$\Phi_{\ell_4} = \{n_1, n_2, n_5\}$$

$$\mu_{\ell_4}(x;\Theta) = \sigma(\theta_1^T x)(1 - \sigma(\theta_2^T x))(1 - \sigma(\theta_5^T x))$$

Objective function

- Per sample likelihood term

$$\mathbb{P}[y|\boldsymbol{x}, \boldsymbol{\pi}, \Theta] = \sum_{\ell \in \mathcal{L}} \mu_\ell(\boldsymbol{x}; \Theta)\pi_y^\ell$$

  - Weighted sum over set of all leaves L

- Overall objective function

$$Q(\mathcal{T}; \Theta, \Pi) = -\frac{1}{|\mathcal{T}|} \sum_{(\boldsymbol{x},y)\in\mathcal{T}} \log\left(\sum_{\ell\in\mathcal{L}} \mu_\ell(\boldsymbol{x}; \Theta)\pi_y^\ell\right)$$

Gradient for split parameter

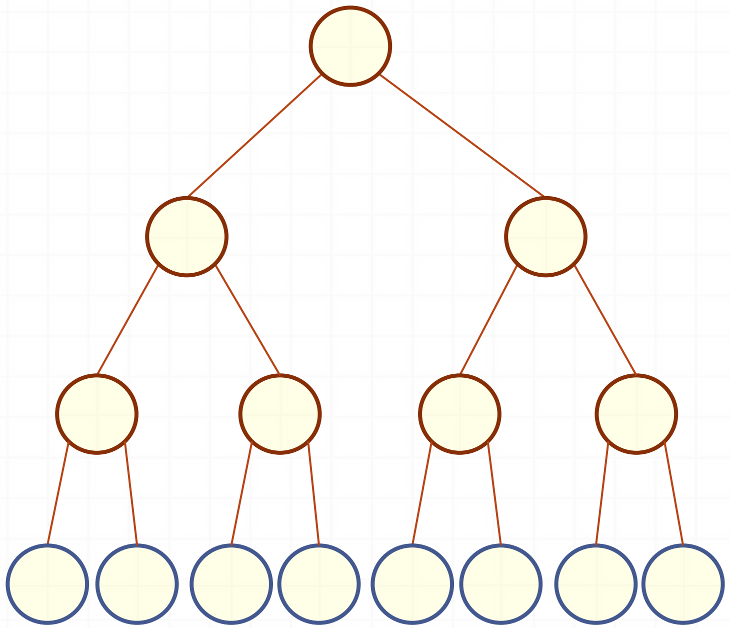
$$\frac{\partial}{\partial \boldsymbol{\theta}_m} Q(\mathcal{T};\Theta,\Pi) = -\frac{1}{|\mathcal{T}|} \sum_{(\boldsymbol{x},y)\in\mathcal{T}} \boldsymbol{x} \frac{\sum_{\ell\in\mathcal{L}_m} \left[\mathbb{1}_{\ell\swarrow m} - \sigma(\boldsymbol{\theta}_m^\top \boldsymbol{x}) \right] \mu_\ell(\boldsymbol{x};\Theta)\pi_y^\ell}{\sum_{\ell'\in\mathcal{L}} \mu_{\ell'}(\boldsymbol{x};\Theta)\pi_y^{\ell'}}$$

$$= -\frac{1}{|\mathcal{T}|} \sum_{(\boldsymbol{x},y)\in\mathcal{T}} \boldsymbol{x} \frac{\left(1-\sigma(\boldsymbol{\theta}_m^\top \boldsymbol{x})\right) \left[\sum_{\ell\in\mathcal{L}_m^{\swarrow}} \mu_\ell(\boldsymbol{x};\Theta)\pi_y^\ell \right] - \sigma(\boldsymbol{\theta}_m^\top \boldsymbol{x}) \left[\sum_{\ell\in\mathcal{L}_m^{\searrow}} \mu_\ell(\boldsymbol{x};\Theta)\pi_y^\ell \right]}{\sum_{\ell'\in\mathcal{L}} \mu_{\ell'}(\boldsymbol{x};\Theta)\pi_y^{\ell'}}$$

Result after forward pass in leaves

Bottom-up sweep, collecting left- and right-subtree contributions

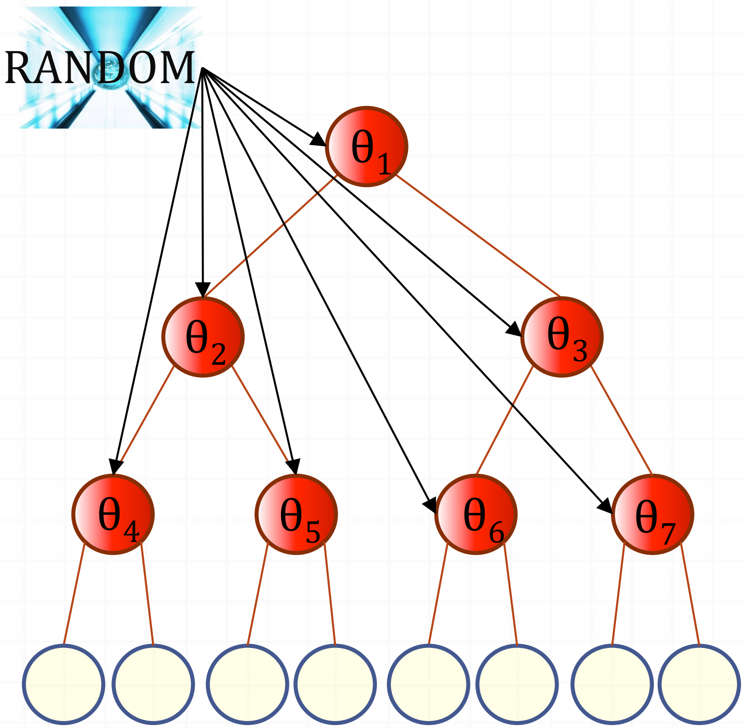
Training Procedure



Consider Fixed Structure

Training Procedure

Θ = RandomUniform([-0.7,0.7])

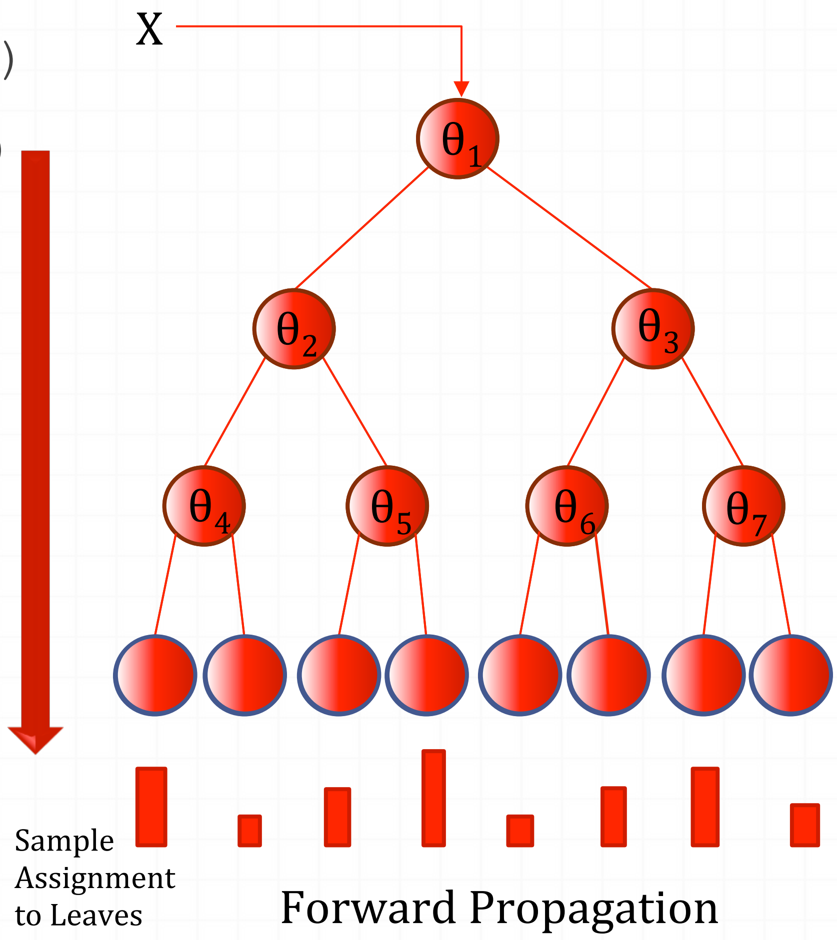


Initialize splits randomly

Training Procedure

Θ = RandomUniform([-0.7,0.7])

M = ForwardPropagation(Θ, X)

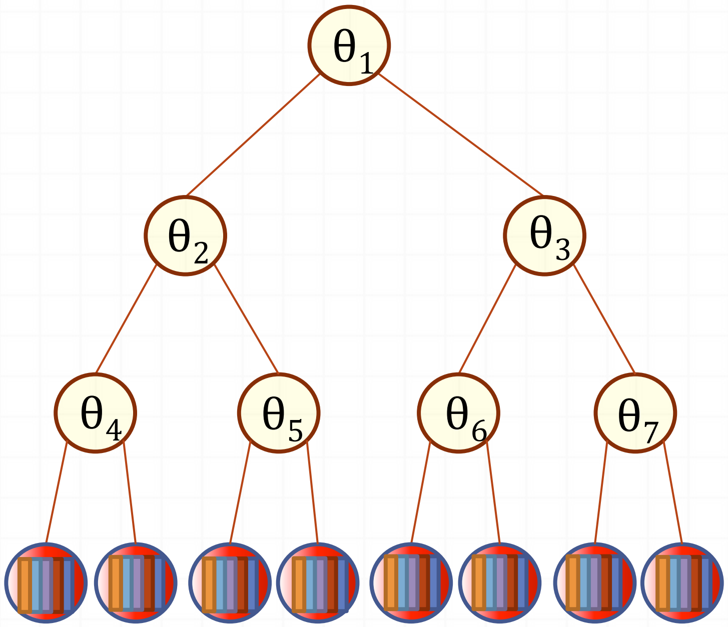


Training Procedure

Θ = RandomUniform([-0.7,0.7])

M = ForwardPropagation(Θ, X)

Π = Uniform(Labels)



Initialize leaf distributions uniformly

Training Procedure

Θ = RandomUniform([-0.7,0.7])

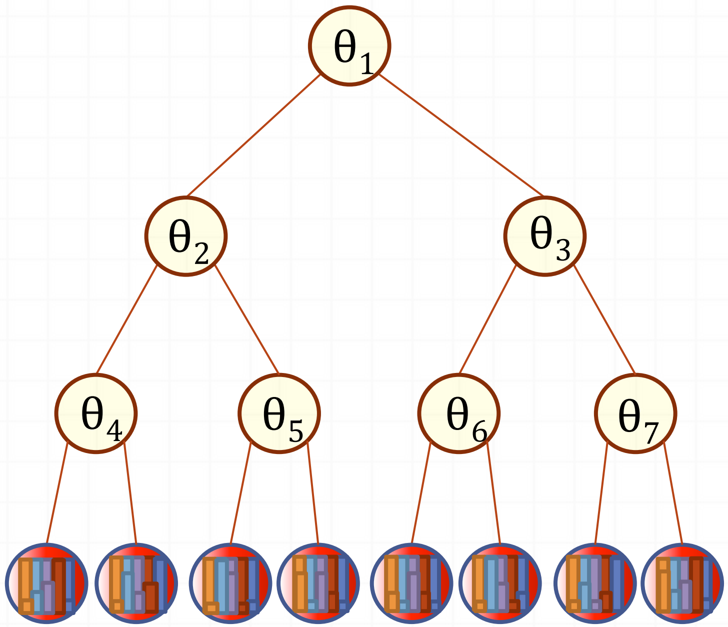
M = ForwardPropagation(Θ, X)

Π = Uniform(Labels)

Π = UpdatePosterior(Θ, Π, M)

$$\pi_c^\ell \leftarrow \frac{1}{Z^\ell} \sum_{i=1}^{|\mathcal{T}|} \frac{\mathbb{1}_{y_i=c}\, \mu_\ell(\boldsymbol{x}_i;\Theta)\, \pi_c^\ell}{\sum_{\ell' \in \mathcal{L}} \mu_{\ell'}(\boldsymbol{x}_i;\Theta) \pi_y^{\ell'}}$$

Adapted from [Rota Bulò & Kontschieder, CVPR'14]



Leaf Distribution Optimization

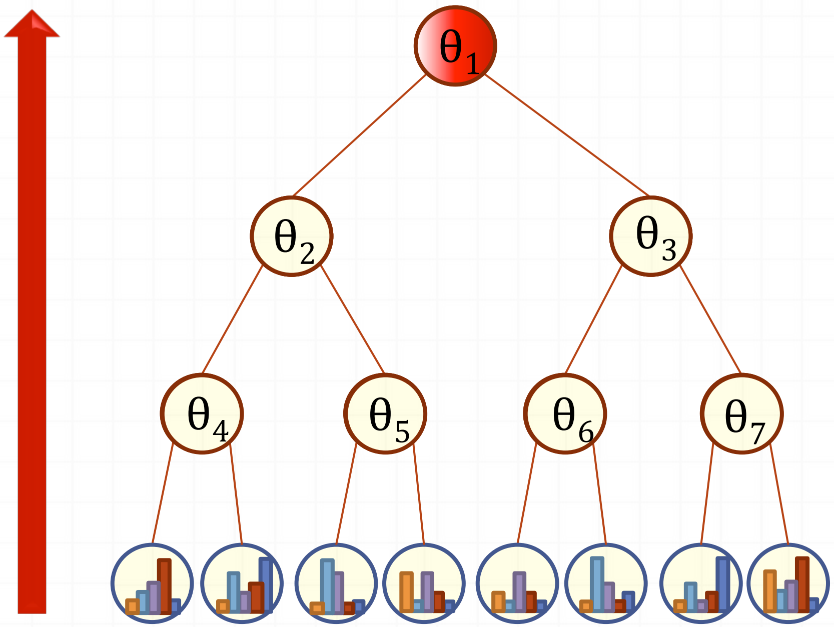
Training Procedure

Θ = RandomUniform([-0.7,0.7])

M = ForwardPropagation(Θ, X)

Π = Uniform(Labels)

Π = UpdatePosterior(Θ, Π, M)



Θ = UpdateSplittingWeights(Θ,Π,X)

Back-propagation

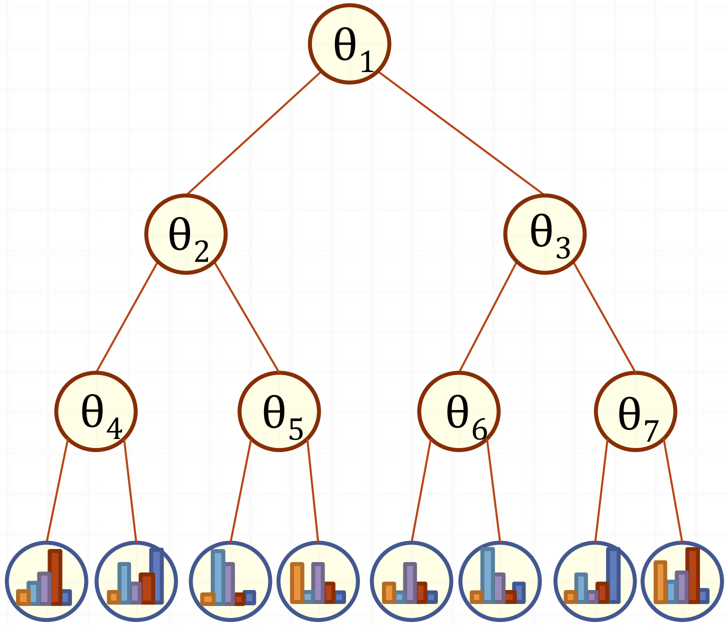
Training Procedure

Θ = RandomUniform([-0.7,0.7])

M = ForwardPropagation(Θ, X)

Π = Uniform(Labels)

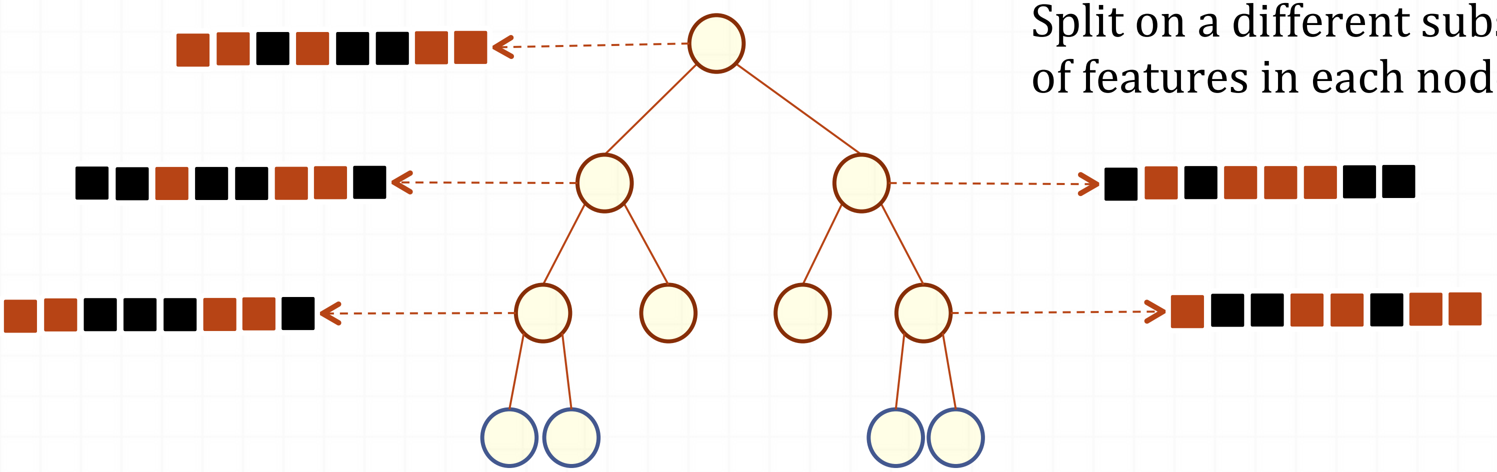
Π = UpdatePosterior(Θ, Π, M)



Θ = UpdateSplittingWeights(Θ,Π,X) Repeat for a number of Epochs

Sparse splits

- Subspace selection per split node
- Allows for high-dimensional input space

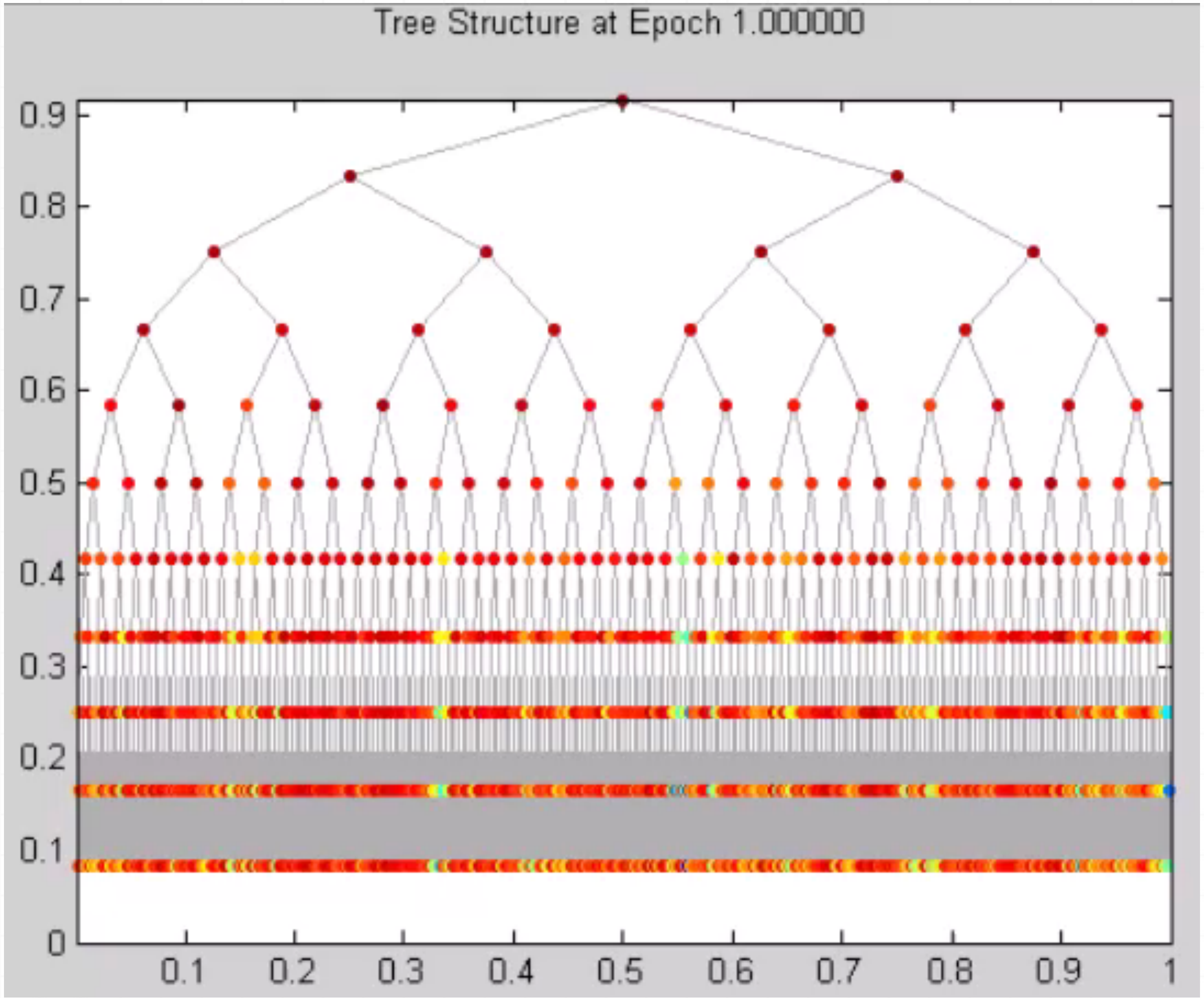


Split on a different subset of features in each node.

- Obtain uncorrelated trees

Flexible Tree Structure

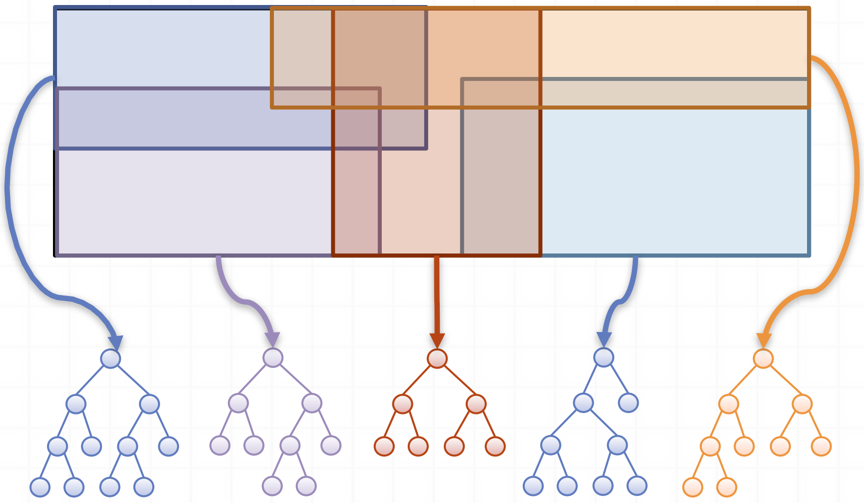
- Split nodes with highest entropy at each epoch



Data Assignment to Trees

- Trees are given different parts of the input space
- Helps keep trees uncorrelated

- For image data, we train trees on different patches



Results Summary

	RF % Error	ADF % Error	BPF % Error	# Train	# Test	Classes	Features	Tree input features	Depth	# Trees
G50C	18.91 ± 1.33	18.71 ± 1.27	**17.4 ± 1.52**	50	500	2	50	10 (random)	5	50
Letter	4.75 ± 0.10	3.52 ± 0.12	**2.92 ± 0.17**	16000	4000	26	16	8 (random)	10	70
USPS	5.96 ± 0.21	5.59 ± 0.16	**5.01 ± 0.24**	7291	2007	10	256	10x10 patches	10	100
MNIST	3.21 ± 0.07	2.71 ± 0.1	2.8 ± 0.12	60000	10000	10	784	15x15 patches	10	80
Char7k	17.76 ± 0.13	16.67 ± 0.21	**16.04 ± 0.2**	66707	7400	62	62	10 (random)	12	200

Deep Neural Decision Forests

- Network outputs become features used by the BPF
- End-to-end deep learning architecture

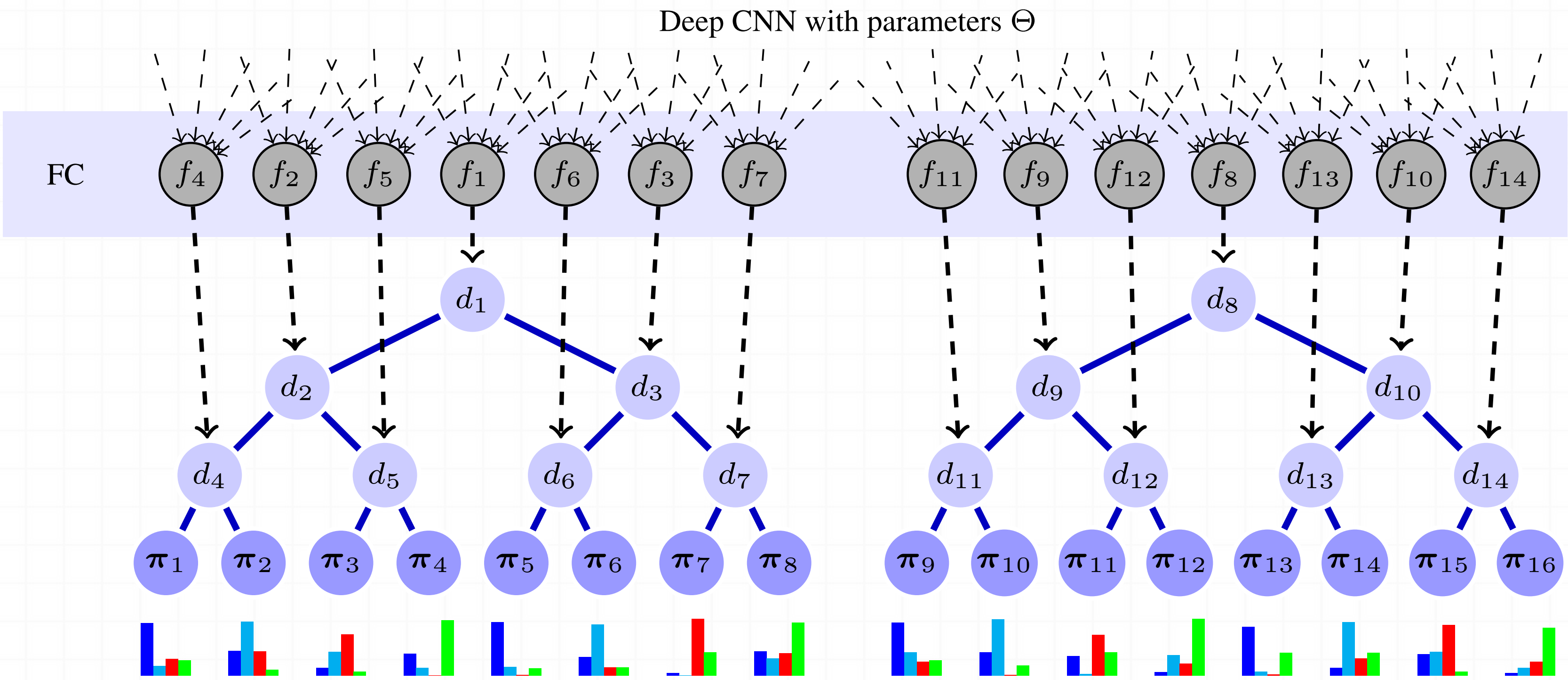


Image by Samuel Rota-Bulò

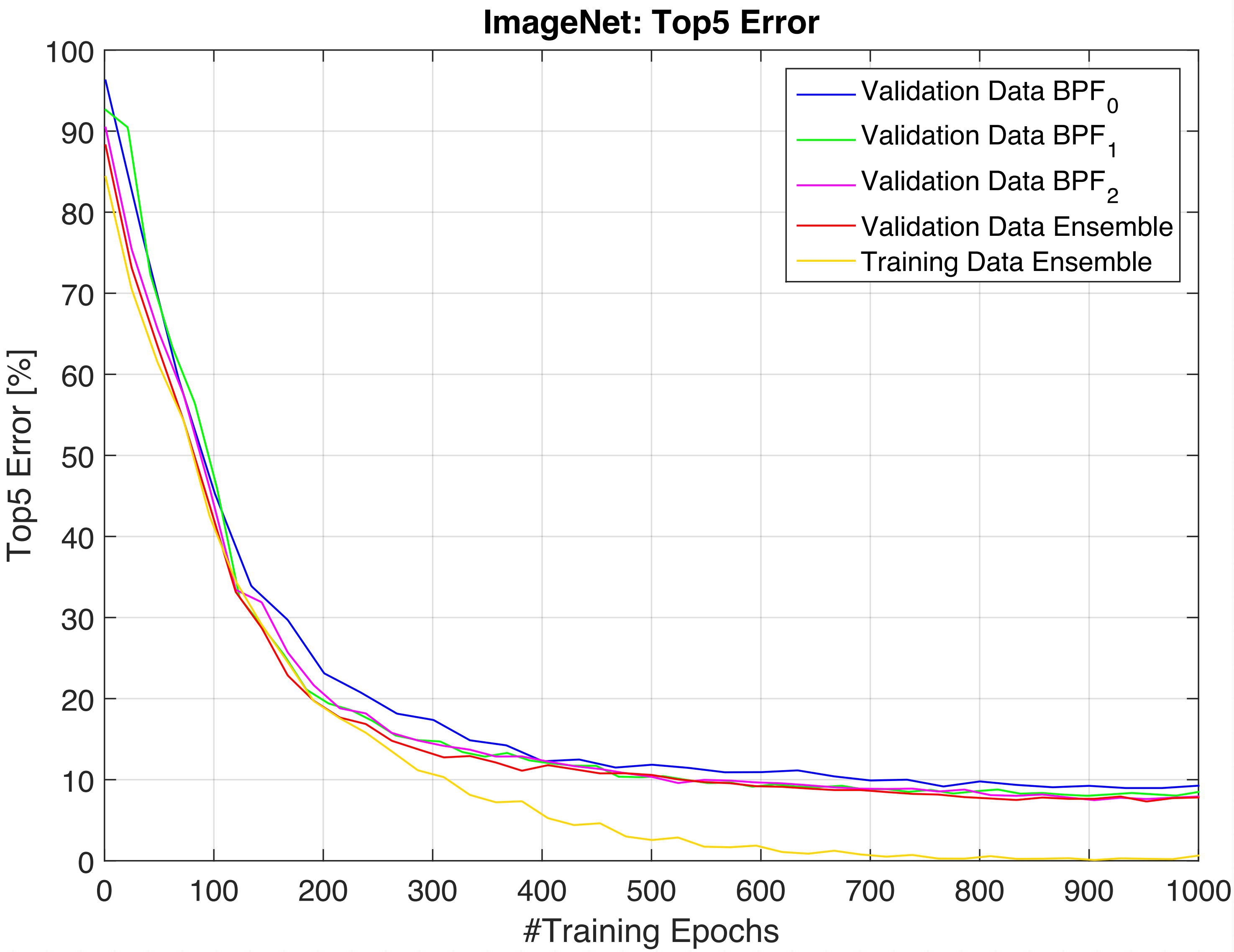
ImageNet Experiment

- 100 million samples,
- 100,000 synsets (classes)
- Modified GoogLeNet[1], replaced Softmax with BPF
- Softmax layers replaced with BPF layers
- Top-5 error reduced from 10.07% to 7.84%.

Description	Top 1 Error	Top 5 Error
1 model, 1 crop	27.8147%	7.84%
1 model, 10 crops	26.9058%	7.08%
7 models, 1 crop	24.1270%	6.38%

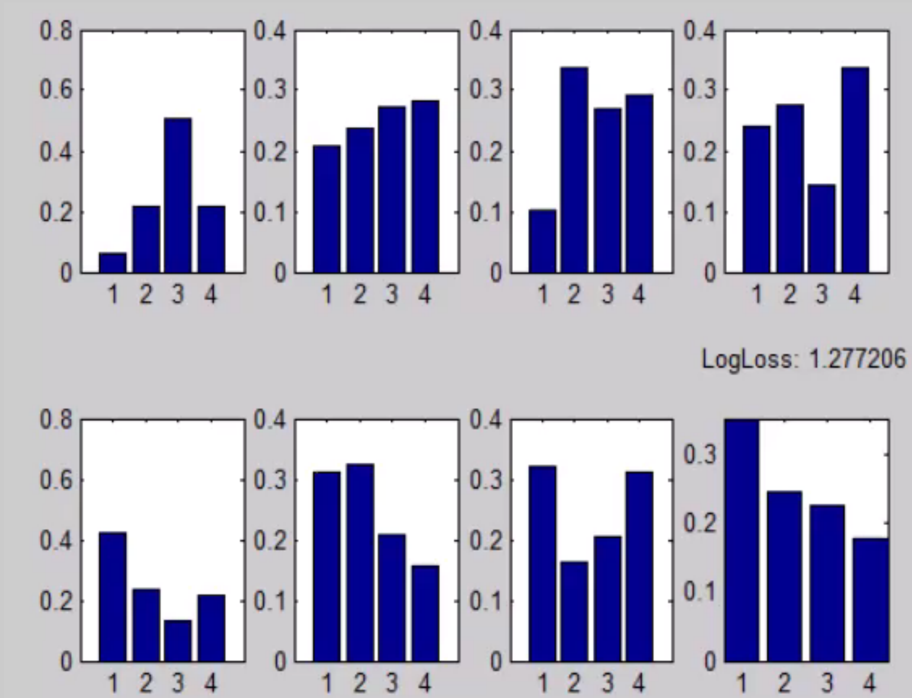
[1] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. *Going deeper with convolutions.* CoRR, abs/1409.4842, 2014.

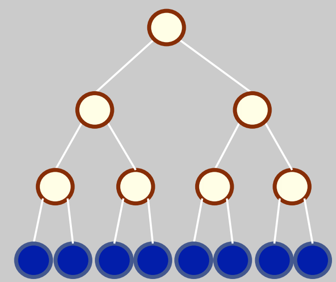
Observation: Learning Curve



Observation: Learning Leaf Predictors

Visualization of learning process



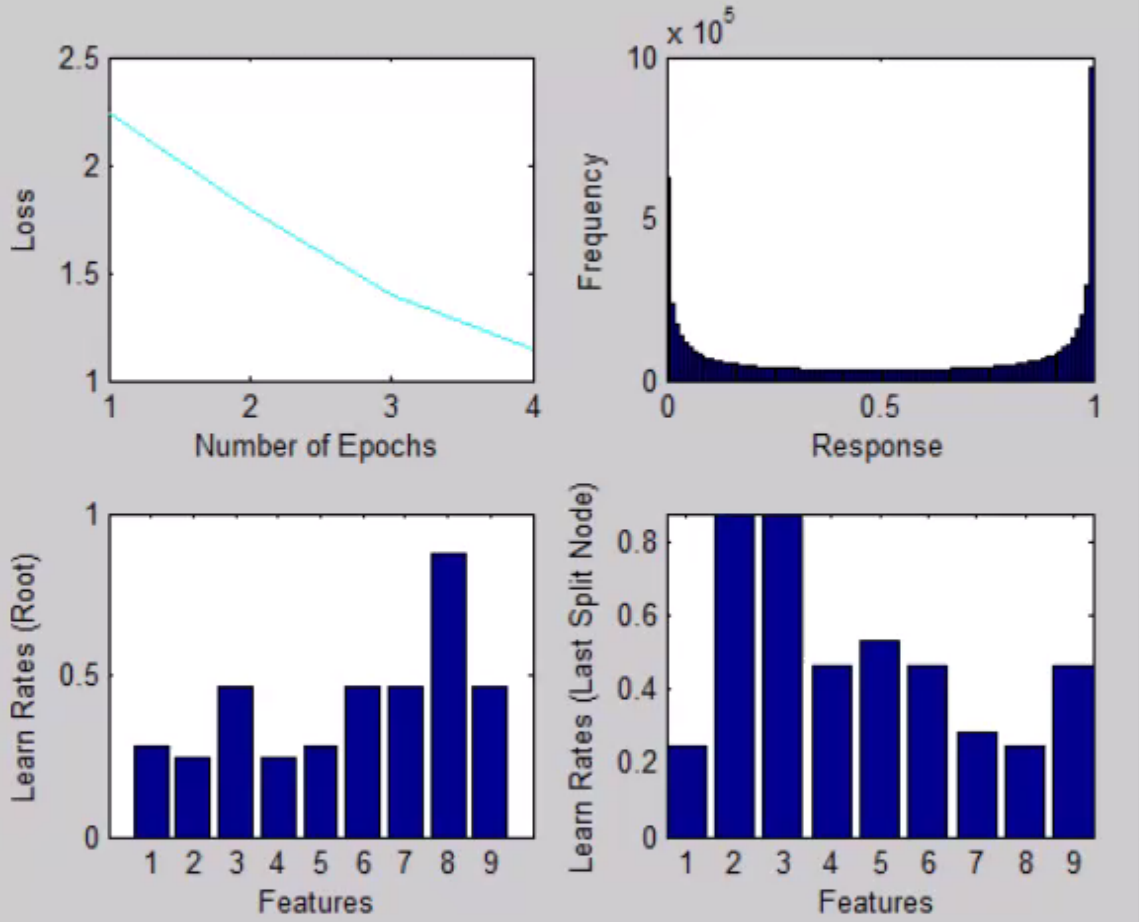


4-class toy example, 8 leaves

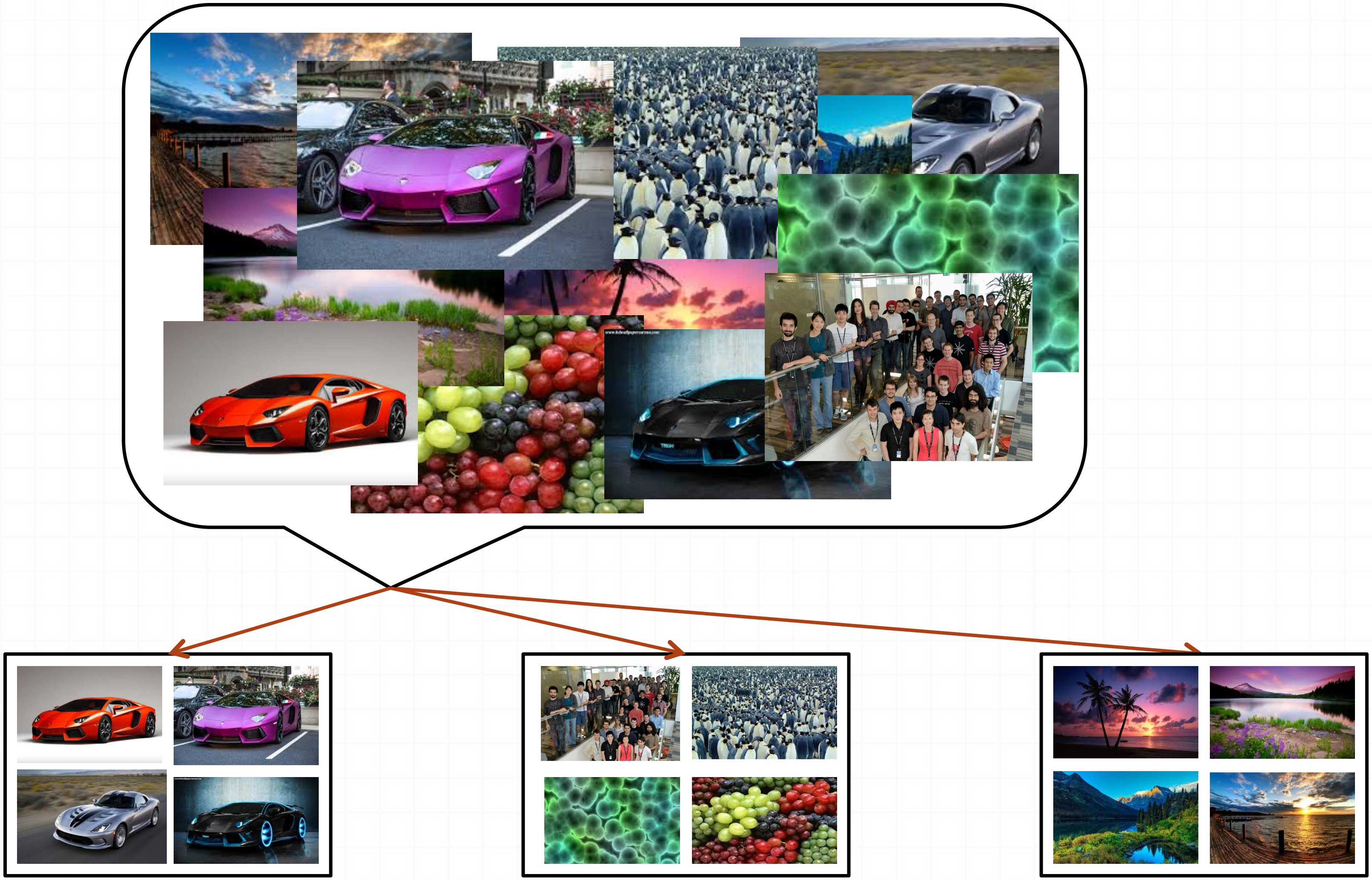
$$\pi_c^\ell \leftarrow \frac{1}{Z^\ell} \sum_{i=1}^{|\mathcal{T}|} \frac{\mathbb{1}_{y_i=c}\, \mu_\ell(\boldsymbol{x}_i;\Theta)\, \pi_c^\ell}{\sum_{\ell' \in \mathcal{L}} \mu_{\ell'}(\boldsymbol{x}_i;\Theta) \pi_y^{\ell'}}$$

Adapted from [Rota Bulò & Kontschieder, CVPR'14]

Observations: Sigmoid outputs



Observation: similarity routing



Madalina Fiterau

mfiterau@cs.stanford.edu

Thanks!

Collaborators:

Peter Kontschieder, Microsoft Research

Antonio Criminisi, Microsoft Research

Samuel Rota-Bulò, Fondazione Bruno Kessler