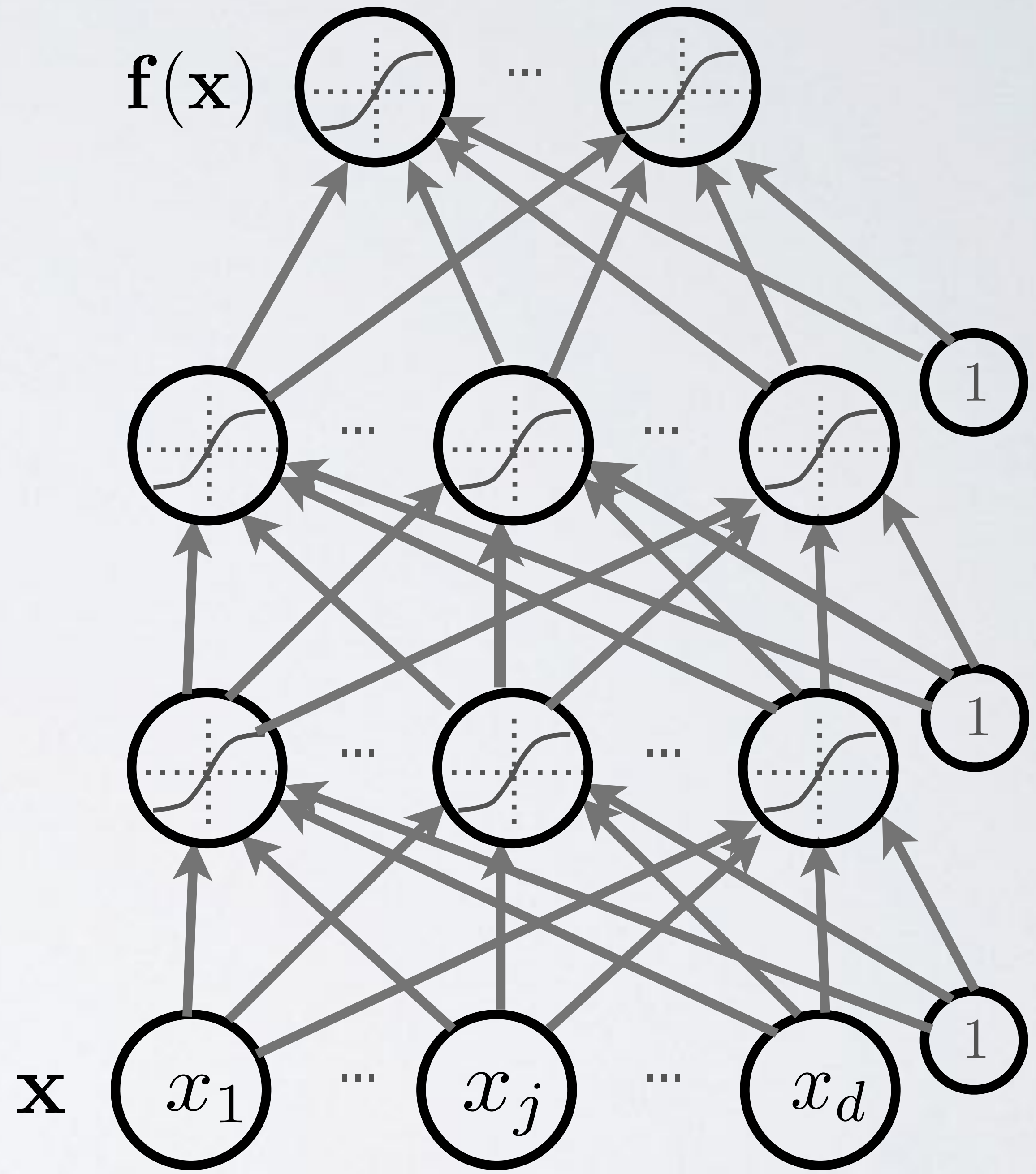
Neural Networks

Hugo Larochelle (@hugo_larochelle)
Google Brain

NEURAL NETWORKS

- What we'll cover
  - types of learning problems
    - definitions of popular learning problems
    - how to define an architecture for a learning problem
  - unintuitive properties of neural networks
    - adversarial examples
    - optimization landscape of neural networks



Neural Networks

Types of learning problems

SUPERVISED LEARNING

Topics: supervised learning

- Training time
  - data :

$$\{\mathbf{x}^{(t)}, y^{(t)}\}$$

  - setting :

$$\mathbf{x}^{(t)}, y^{(t)} \sim p(\mathbf{x}, y)$$

- Test time
  - data :

$$\{\mathbf{x}^{(t)}, y^{(t)}\}$$

  - setting :

$$\mathbf{x}^{(t)}, y^{(t)} \sim p(\mathbf{x}, y)$$

- Example
  - classification
  - regression

UNSUPERVISED LEARNING

Topics: unsupervised learning

- Training time
  - data :

    $$\{\mathbf{x}^{(t)}\}$$

  - setting :

    $$\mathbf{x}^{(t)} \sim p(\mathbf{x})$$

- Test time
  - data :

    $$\{\mathbf{x}^{(t)}\}$$

  - setting :

    $$\mathbf{x}^{(t)} \sim p(\mathbf{x})$$

- Example
  - distribution estimation
  - dimensionality reduction

SEMI-SUPERVISED LEARNING

Topics: semi-supervised learning

- Training time
  - data :

$$\{\mathbf{x}^{(t)}, y^{(t)}\}$$

$$\{\mathbf{x}^{(t)}\}$$

  - setting :

$$\mathbf{x}^{(t)}, y^{(t)} \sim p(\mathbf{x}, y)$$

$$\mathbf{x}^{(t)} \sim p(\mathbf{x})$$

- Test time
  - data :

$$\{\mathbf{x}^{(t)}, y^{(t)}\}$$

  - setting :

$$\mathbf{x}^{(t)}, y^{(t)} \sim p(\mathbf{x}, y)$$

MULTITASK LEARNING

Topics: multitask learning

- Training time
  - data :

$$\{\mathbf{x}^{(t)}, y_1^{(t)}, \ldots, y_M^{(t)}\}$$

  - setting :

$$\mathbf{x}^{(t)}, y_1^{(t)}, \ldots, y_M^{(t)} \sim p(\mathbf{x}, y_1, \ldots, y_M)$$

- Test time
  - data :

$$\{\mathbf{x}^{(t)}, y_1^{(t)}, \ldots, y_M^{(t)}\}$$

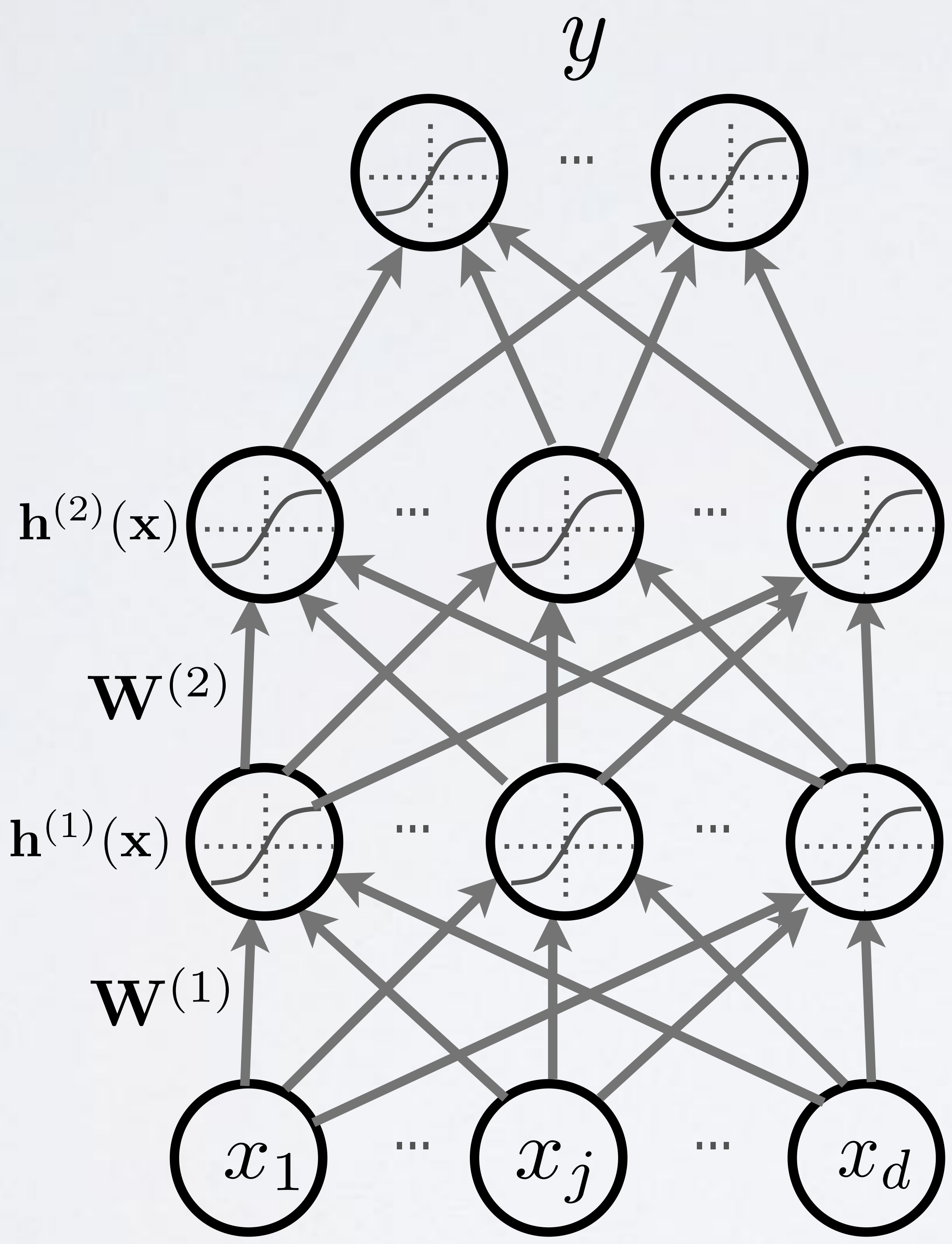
  - setting :

$$\mathbf{x}^{(t)}, y_1^{(t)}, \ldots, y_M^{(t)} \sim p(\mathbf{x}, y_1, \ldots, y_M)$$

- Example
  - object recognition in images with multiple objects

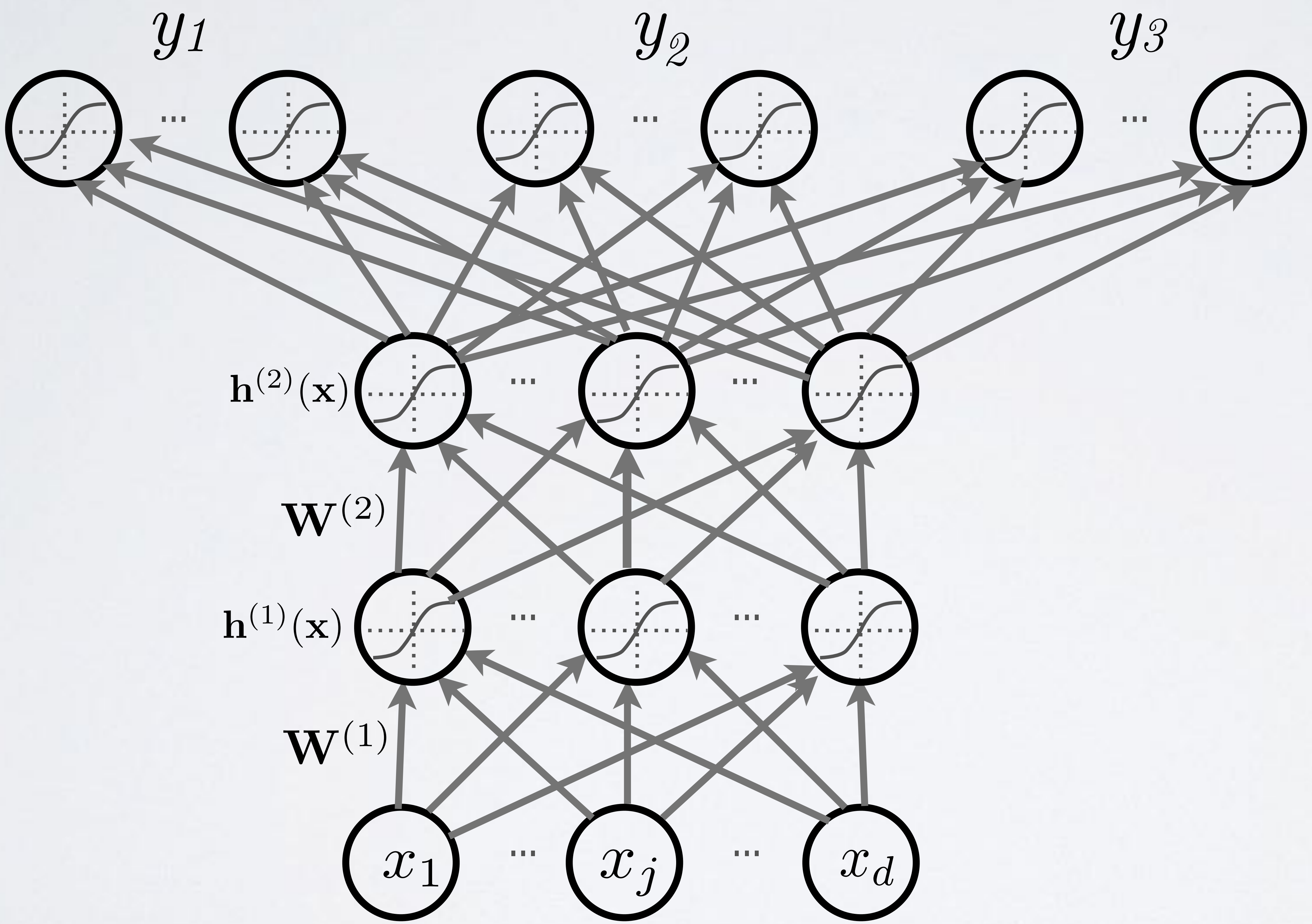
MULTITASK LEARNING

Topics: multitask learning



MULTITASK LEARNING

Topics: multitask learning



TRANSFER LEARNING

Topics: transfer learning

- Training time
  - data :

$$\{\mathbf{x}^{(t)}, y_1^{(t)}, \ldots, y_M^{(t)}\}$$

  - setting :

$$\mathbf{x}^{(t)}, y_1^{(t)}, \ldots, y_M^{(t)} \sim p(\mathbf{x}, y_1, \ldots, y_M)$$

- Test time
  - data :

$$\{\mathbf{x}^{(t)}, y_1^{(t)}\}$$

  - setting :

$$\mathbf{x}^{(t)}, y_1^{(t)} \sim p(\mathbf{x}, y_1)$$

STRUCTURED OUTPUT PREDICTION

Topics: structured output prediction

- Training time
  - data :

    $$\{\mathbf{x}^{(t)}, \mathbf{y}^{(t)}\}$$

    of arbitrary structure (vector, sequence, graph)
  - setting :

    $$\mathbf{x}^{(t)}, \mathbf{y}^{(t)} \sim p(\mathbf{x}, \mathbf{y})$$

- Test time
  - data :

    $$\{\mathbf{x}^{(t)}, \mathbf{y}^{(t)}\}$$

  - setting :

    $$\mathbf{x}^{(t)}, \mathbf{y}^{(t)} \sim p(\mathbf{x}, \mathbf{y})$$

- Example
  - image caption generation
  - machine translation

DOMAIN ADAPTATION

Topics: domain adaptation, covariate shift

- Training time
  - data :

$$\{\mathbf{x}^{(t)}, y^{(t)}\}$$
$$\{\bar{\mathbf{x}}^{(t')}\}$$

  - setting :

$$\mathbf{x}^{(t)} \sim p(\mathbf{x})$$
$$y^{(t)} \sim p(y|\mathbf{x}^{(t)})$$
$$\bar{\mathbf{x}}^{(t)} \sim q(\mathbf{x}) \approx p(\mathbf{x})$$

- Test time
  - data :

$$\{\bar{\mathbf{x}}^{(t)}, y^{(t)}\}$$

  - setting :

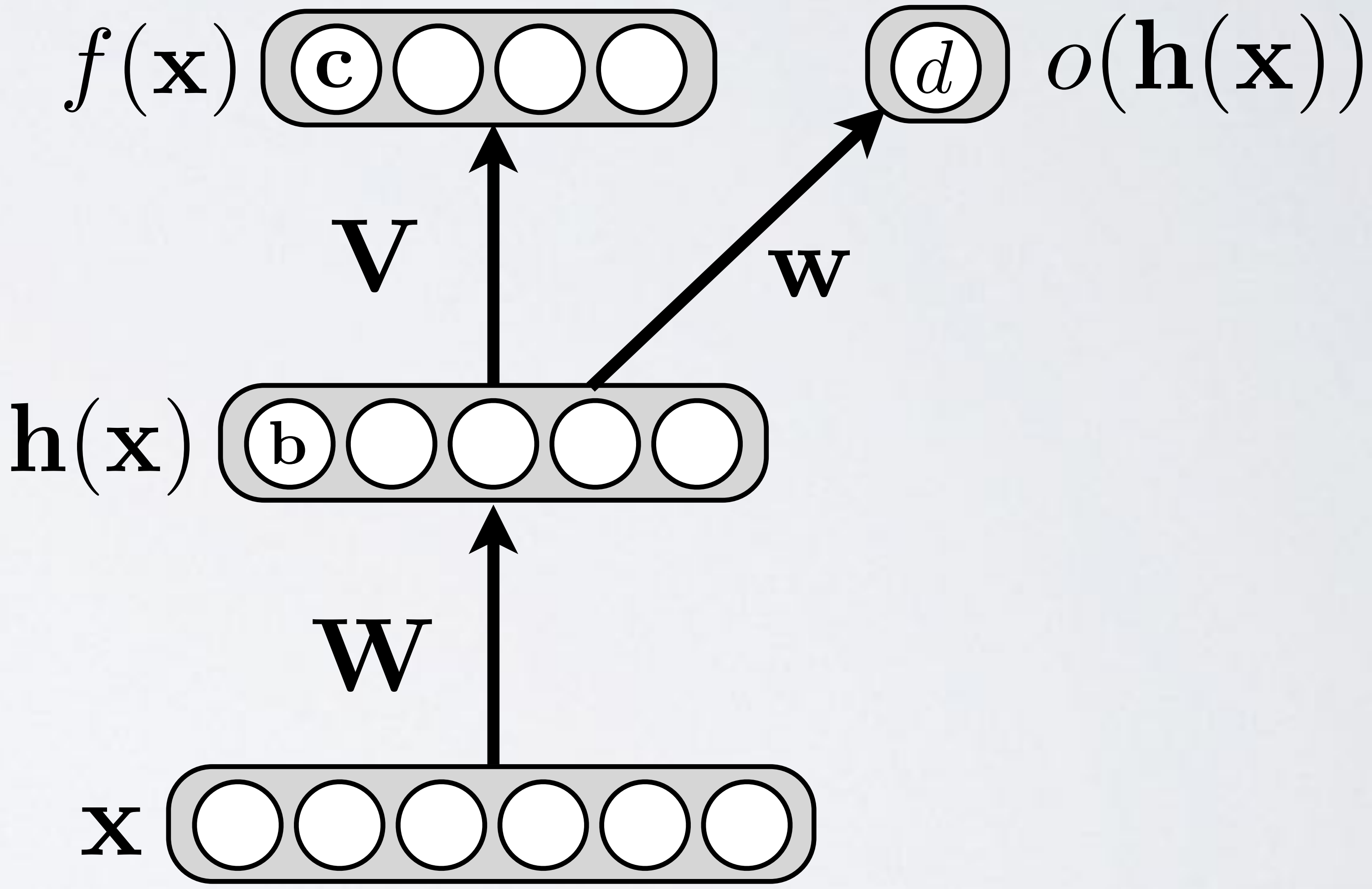
$$\bar{\mathbf{x}}^{(t)} \sim q(\mathbf{x})$$
$$y^{(t)} \sim p(y|\bar{\mathbf{x}}^{(t)})$$

- Example
  - classify sentiment in reviews of different products
  - training on synthetic data but testing on real data (sim2real)

DOMAIN ADAPTATION

Topics: domain adaptation, covariate shift

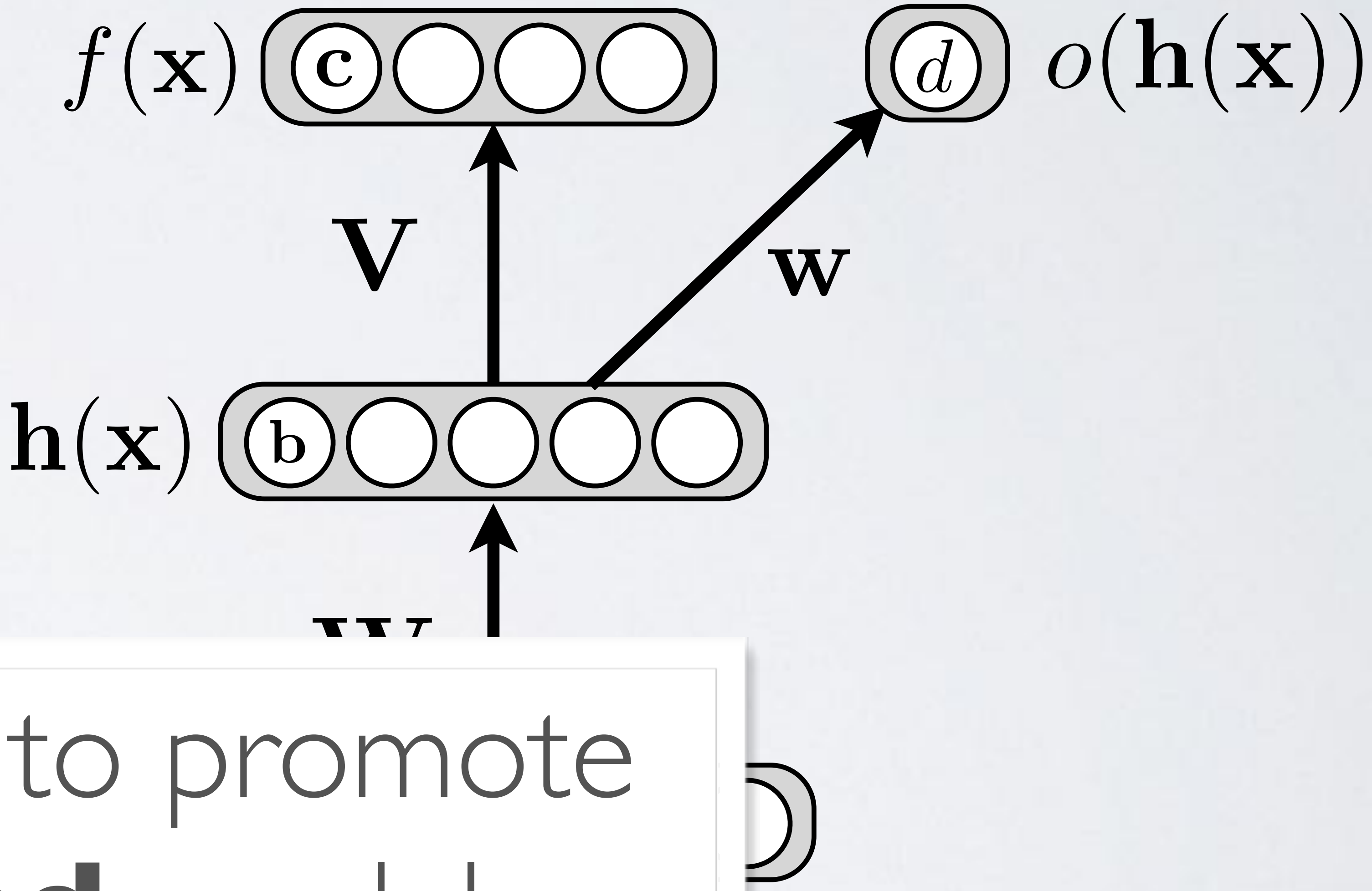
- Domain-adversarial networks (Ganin et al. 2015) train hidden layer representation to be
  1. **predictive** of the target class
  2. **indiscriminate** of the domain
- Trained by stochastic gradient descent
  - for each random pair $\mathbf{x}^{(t)}, \bar{\mathbf{x}}^{(t')}$
    1. update $\mathbf{W}, \mathbf{V}, \mathbf{b}, \mathbf{c}$ in opposite direction of gradient
    2. update $\mathbf{w}, d$ in direction of gradient



DOMAIN ADAPTATION

Topics: domain adaptation, covariate shift

- Domain-adversarial networks (Ganin et al. 2015) train hidden layer representation to be
  1. **predictive** of the target class
  2. **indiscriminate** of the domain



- Trained by stochastic gradient descent
  - for each random pair $\mathbf{x}^{(t)}, \bar{\mathbf{x}}^{(t')}$
    1. update $\mathbf{W}, \mathbf{V}, \mathbf{b}, \mathbf{c}$ i
    2. update $\mathbf{w}, d$ in dire

May also be used to promote **fair** and **unbiased** models ...

ONE-SHOT LEARNING

Topics: one-shot learning

- Training time
  - data :

$$\{\mathbf{x}^{(t)}, y^{(t)}\}$$

  - setting :

$$\mathbf{x}^{(t)}, y^{(t)} \sim p(\mathbf{x}, y)$$

subject to $y^{(t)} \in \{1, \ldots, C\}$

- Test time
  - data :

$$\{\mathbf{x}^{(t)}, y^{(t)}\}$$

  - setting :

$$\mathbf{x}^{(t)}, y^{(t)} \sim p(\mathbf{x}, y)$$

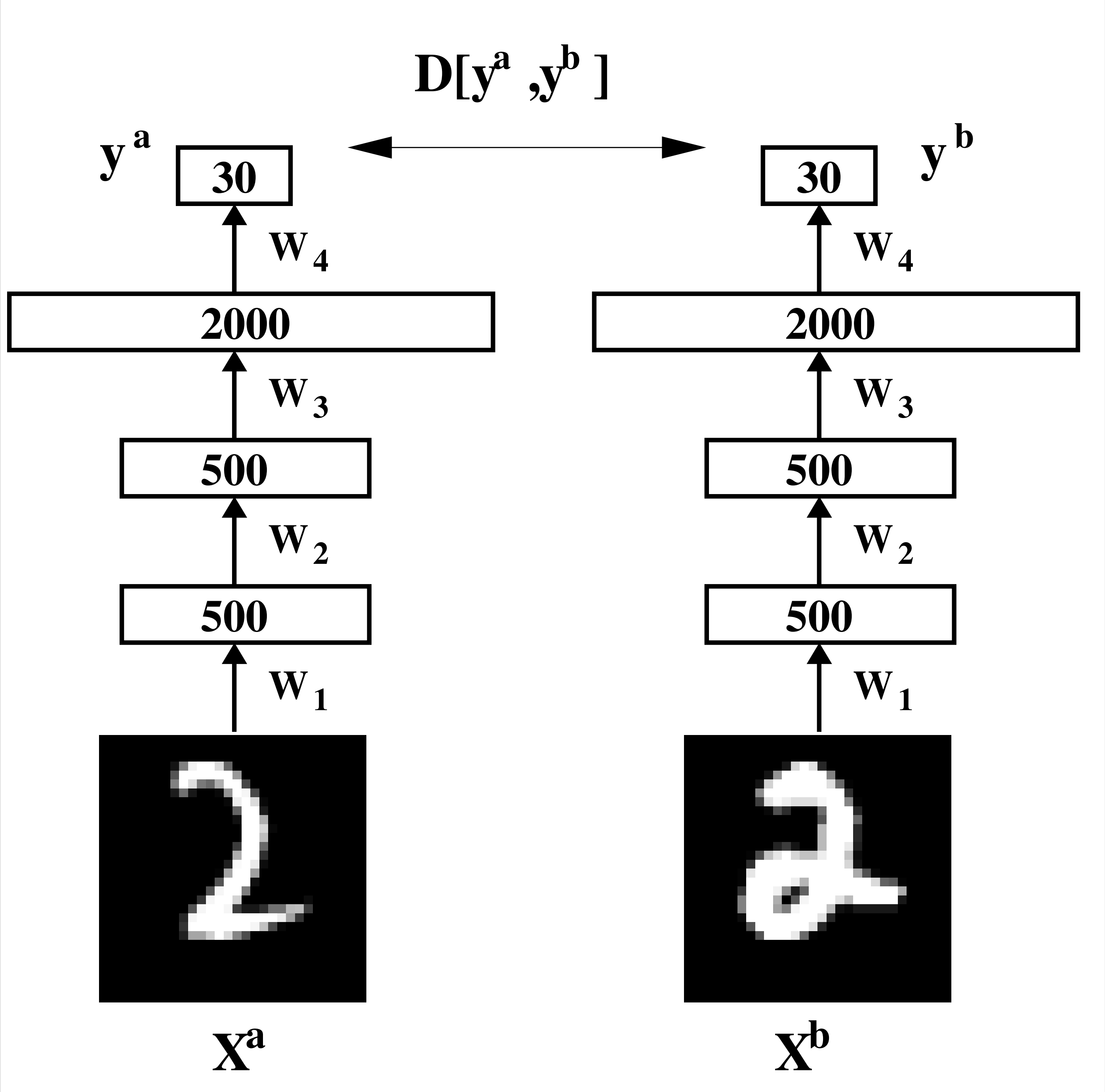
subject to $y^{(t)} \in \{C+1, \ldots, C+M\}$

  - side information :
    - a single labeled example from each of the M new classes

- Example
  - recognizing a person based on a single picture of him/her

ONE-SHOT LEARNING

Topics: one-shot learning



Siamese architecture
(figure taken from Salakhutdinov and Hinton, 2007)

ZERO-SHOT LEARNING

Topics: zero-shot learning, zero-data learning

- Training time
  - data :

$$\{\mathbf{x}^{(t)}, y^{(t)}\}$$

  - setting :

$$\mathbf{x}^{(t)}, y^{(t)} \sim p(\mathbf{x}, y)$$

  subject to $y^{(t)} \in \{1, \ldots, C\}$

  - side information :
    - description vector $\mathbf{z}_c$ of each of the C classes

- Test time
  - data :

$$\{\mathbf{x}^{(t)}, y^{(t)}\}$$

  - setting :

$$\mathbf{x}^{(t)}, y^{(t)} \sim p(\mathbf{x}, y)$$

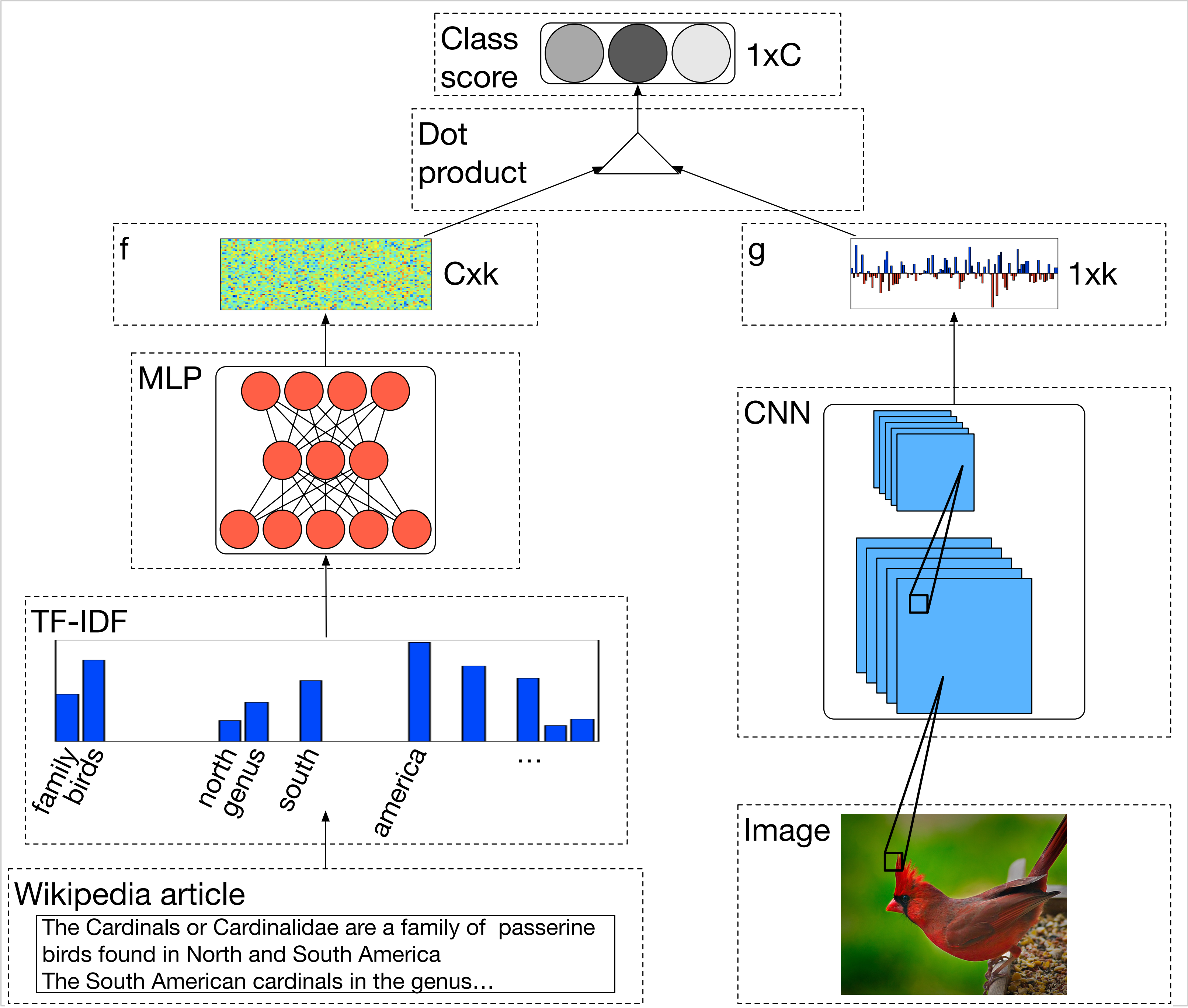
  subject to $y^{(t)} \in \{C+1, \ldots, C+M\}$

  - side information :
    - description vector $\mathbf{z}_c$ of each of the new M classes

- Example
  - recognizing an object based on a worded description of it

ZERO-SHOT LEARNING

Topics: zero-shot learning, zero-data learning



Ba, Swersky, Fidler, Salakhutdinov
arxiv 2015

DESIGNING NEW ARCHITECTURES

Topics: designing new architectures

- Tackling a new learning problem often requires designing an adapted neural architecture

- Approach 1: use our intuition for how a human would reason about the problem

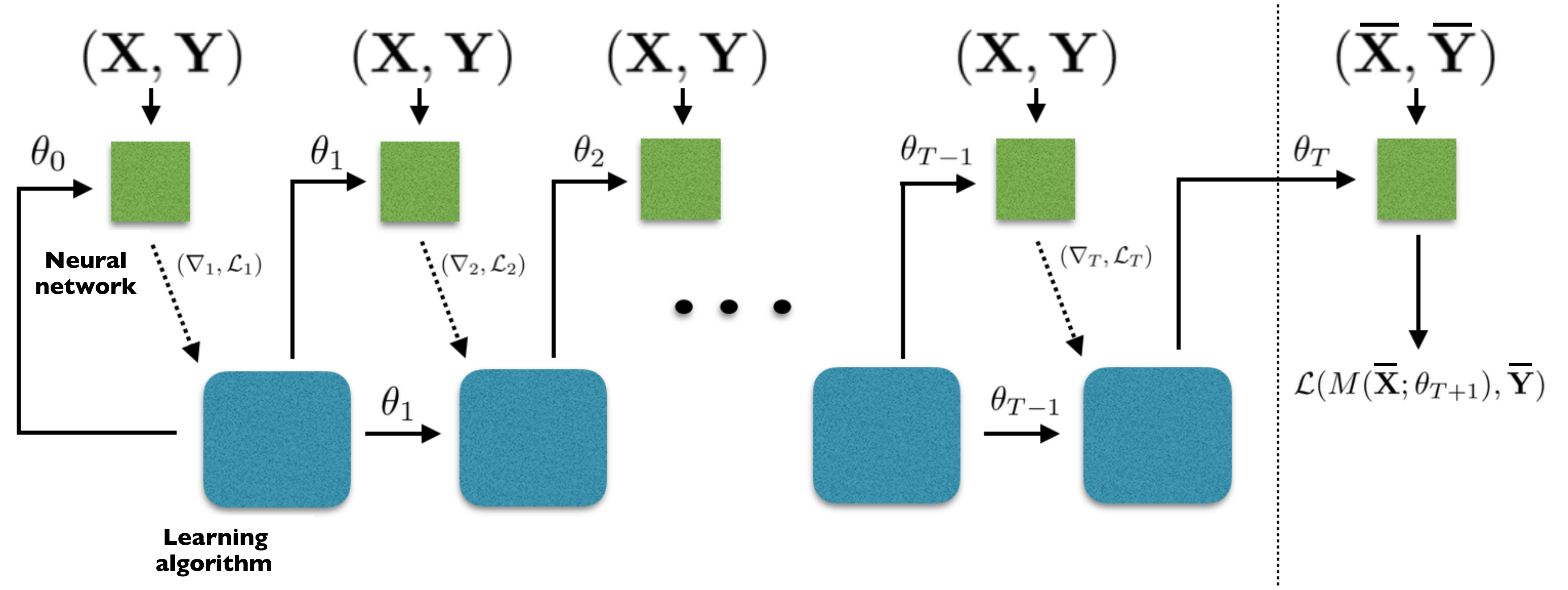
- Approach 2: take an existing algorithm/procedure and turn it into a neural network

DESIGNING NEW ARCHITECTURES

Topics: designing new architectures

- Many other examples
  - structured prediction by unrolling probabilistic inference in an MRF
  - planning by unrolling the value iteration algorithm (Tamar et al., NIPS 2016)
  - few-shot learning by unrolling gradient descent on small training set

Ravi and Larochelle, ICLR 2017



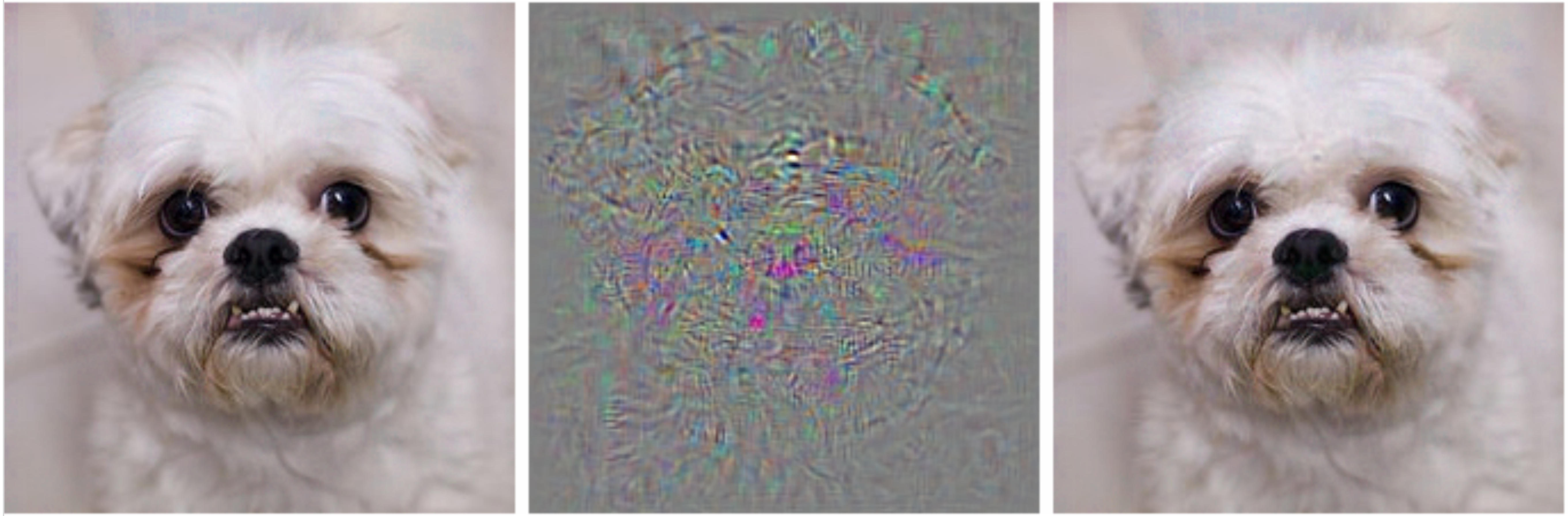
Neural networks

Unintuitive properties of neural networks

THEY CAN MAKE DUMB ERRORS

Topics: adversarial examples

- *Intriguing Properties of Neural Networks*
  Szegedy, Zaremba, Sutskever, Bruna, Erhan, Goodfellow, Fergus, ICLR 2014



Correctly classified | Difference | Badly classified

THEY CAN MAKE DUMB ERRORS

Topics: adversarial examples

- Humans have adversarial examples too

- However they don't match those of neural networks

THEY CAN MAKE DUMB ERRORS

Topics: adversarial examples

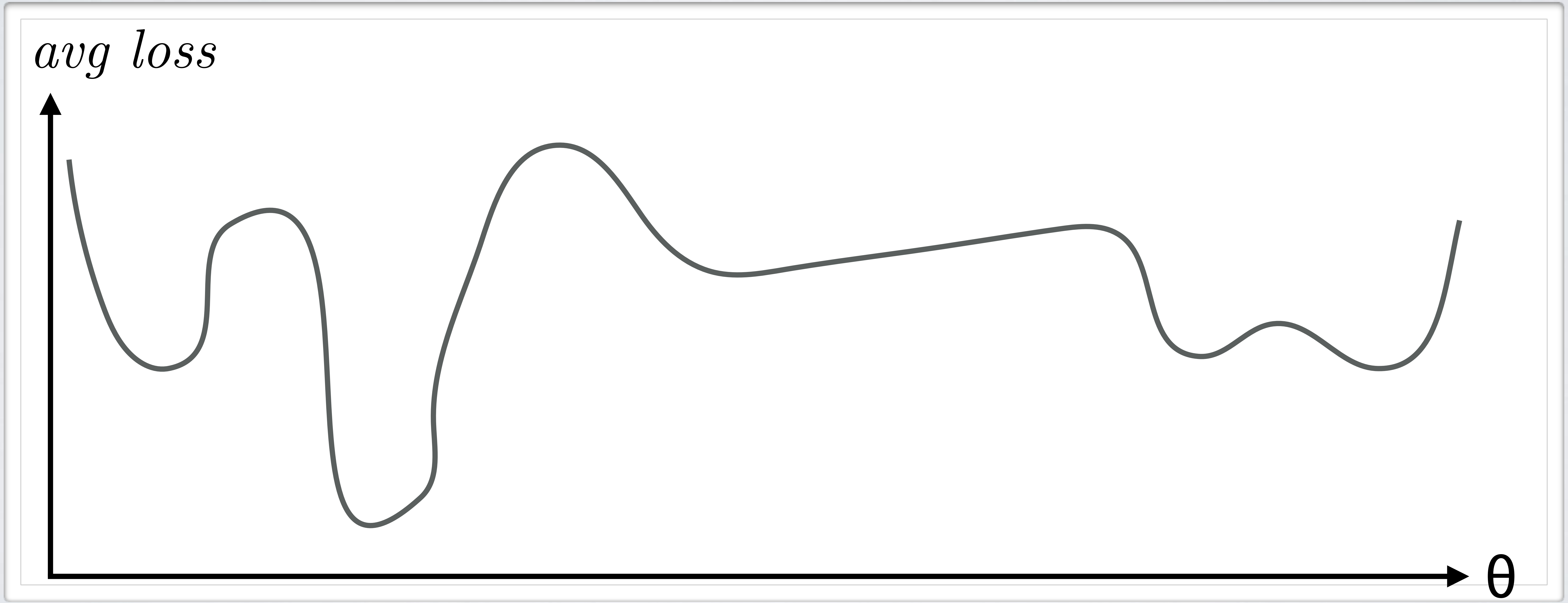
- Humans have adversarial examples too

- However they don't match those of neural networks

THEY ARE STRANGELY NON-CONVEX

Topics: non-convexity, saddle points

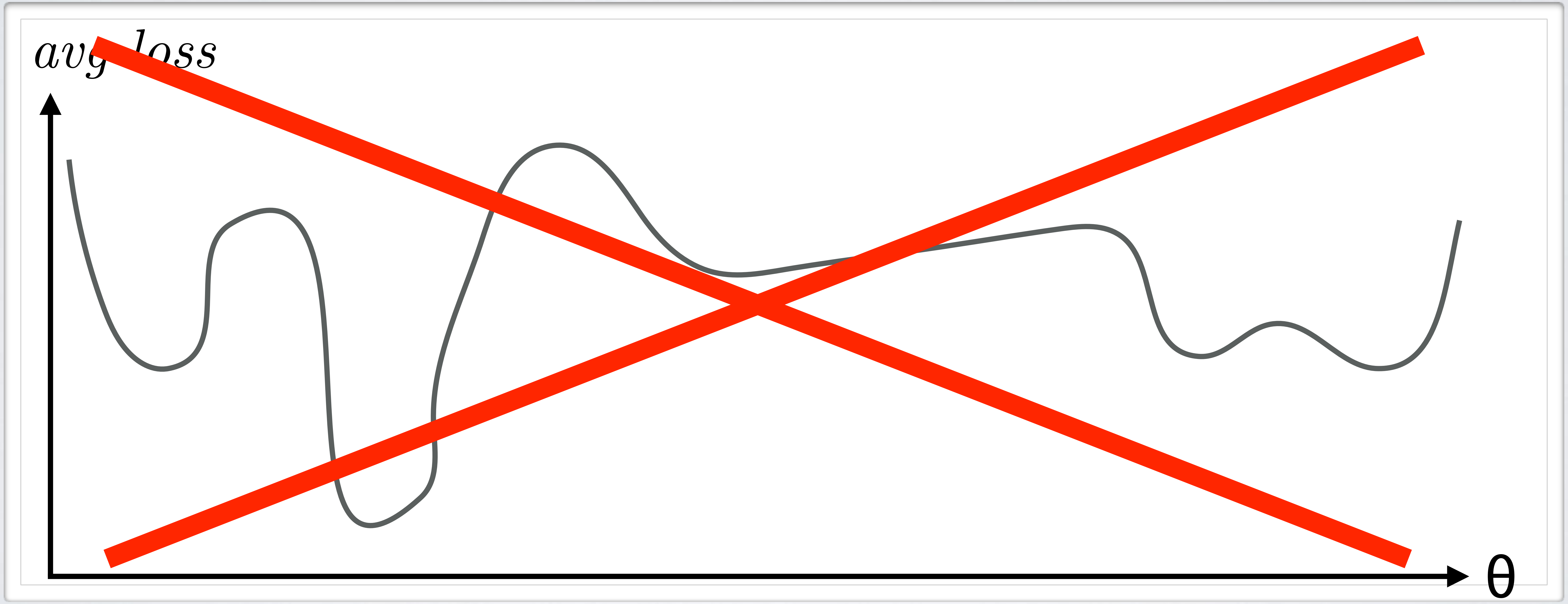
- *Identifying and attacking the saddle point problem in high-dimensional non-convex optimization* Dauphin, Pascanu, Gulcehre, Cho, Ganguli, Bengio, NIPS 2014



THEY ARE STRANGELY NON-CONVEX

Topics: non-convexity, saddle points

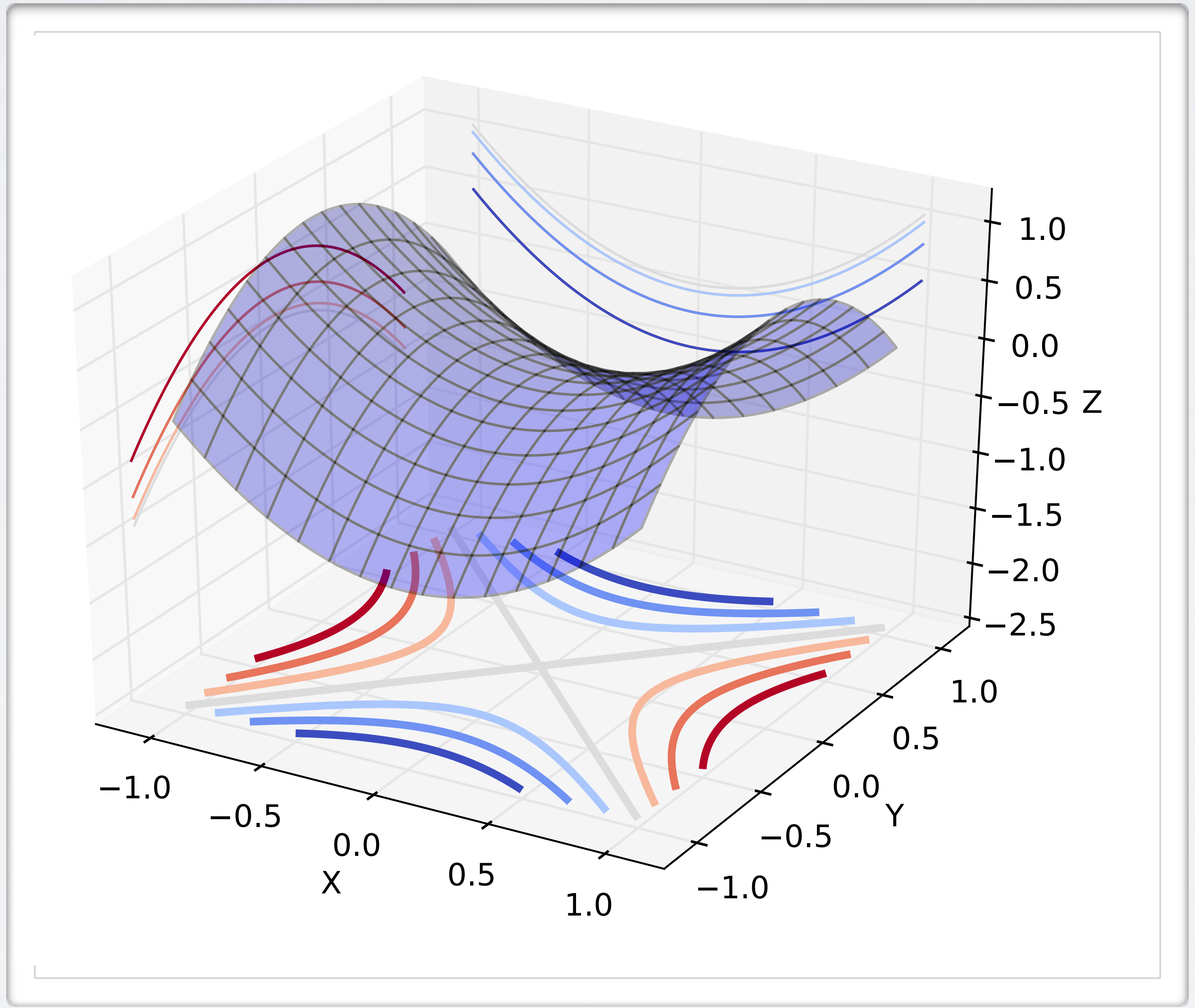
- *Identifying and attacking the saddle point problem in high-dimensional non-convex optimization* Dauphin, Pascanu, Gulcehre, Cho, Ganguli, Bengio, NIPS 2014



THEY ARE STRANGELY NON-CONVEX

Topics: non-convexity, saddle points

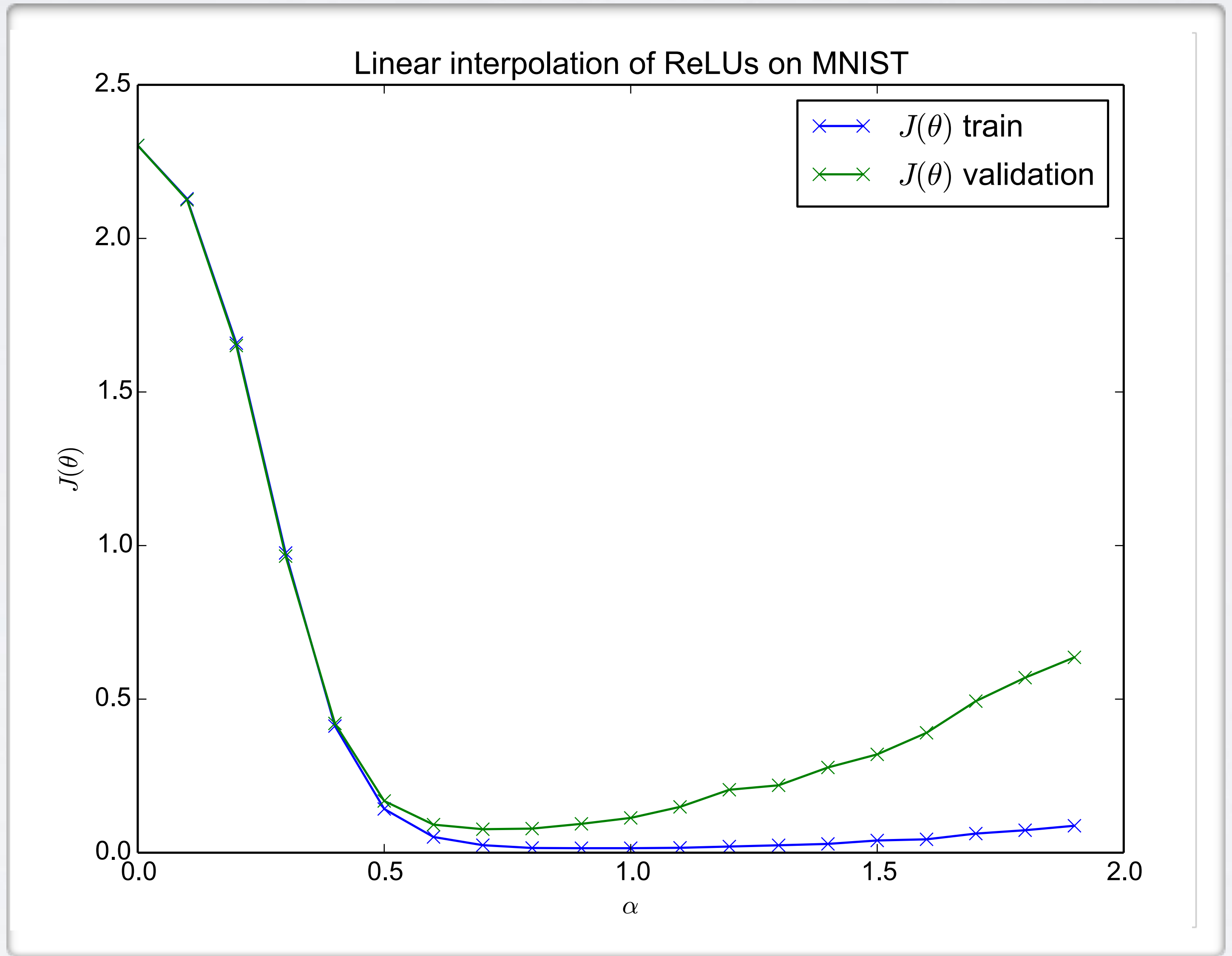
- *Identifying and attacking the saddle point problem in high-dimensional non-convex optimization* Dauphin, Pascanu, Gulcehre, Cho, Ganguli, Bengio, NIPS 2014



THEY ARE STRANGELY NON-CONVEX

Topics: non-convexity, saddle points

- *Qualitatively Characterizing Neural Network Optimization Problems* Goodfellow, Vinyals, Saxe, ICLR 2015



THEY ARE STRANGELY NON-CONVEX

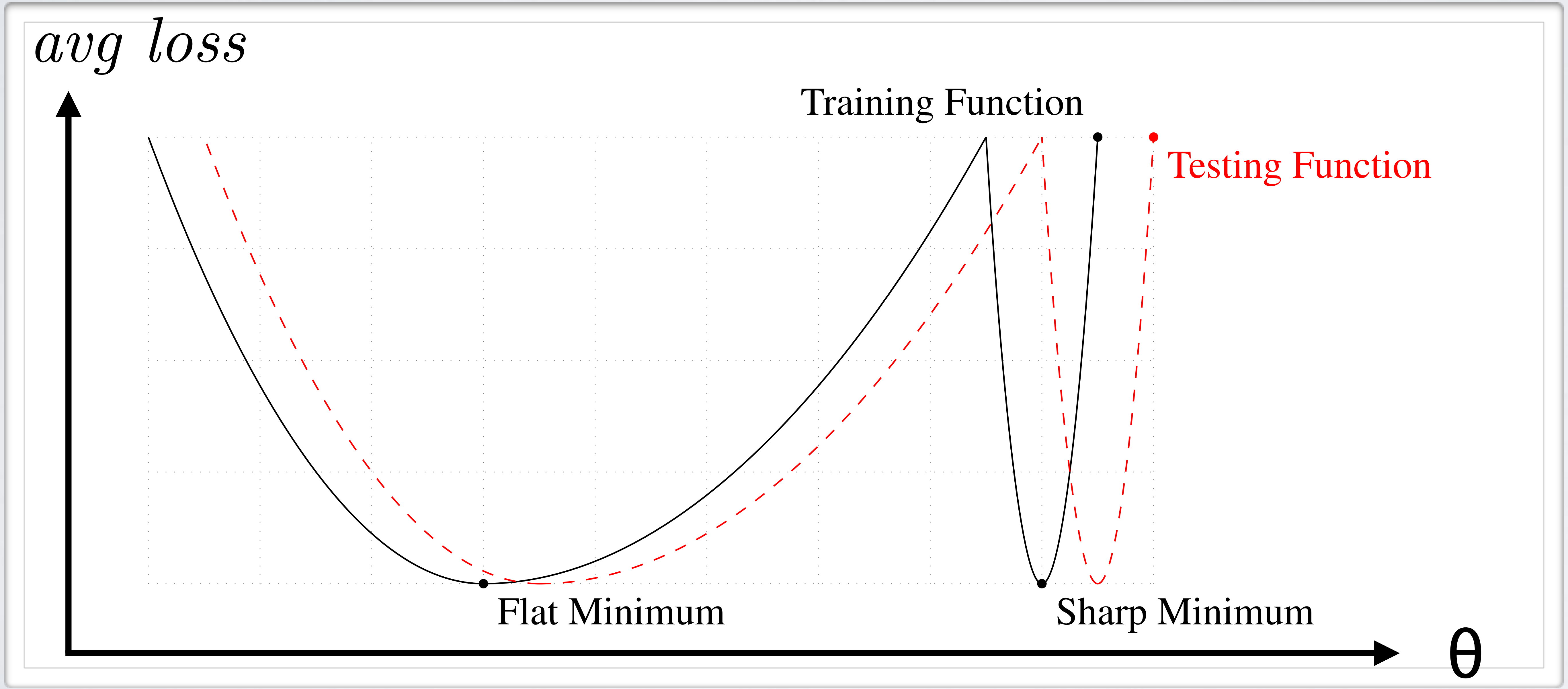
Topics: non-convexity, saddle points

- If dataset is created by labeling points using a *N*-hidden units neural network
  - training another *N*-hidden units network is likely to fail
  - but training a larger neural network is more likely to work!
    (saddle points seem to be a blessing)

THEY WORK BEST WHEN BADLY TRAINED

Topics: sharp vs. flat miniman

- *Flat Minima*
  Hochreiter, Schmidhuber, Neural Computation 1997



THEY WORK BEST WHEN BADLY TRAINED

Topics: sharp vs. flat miniman

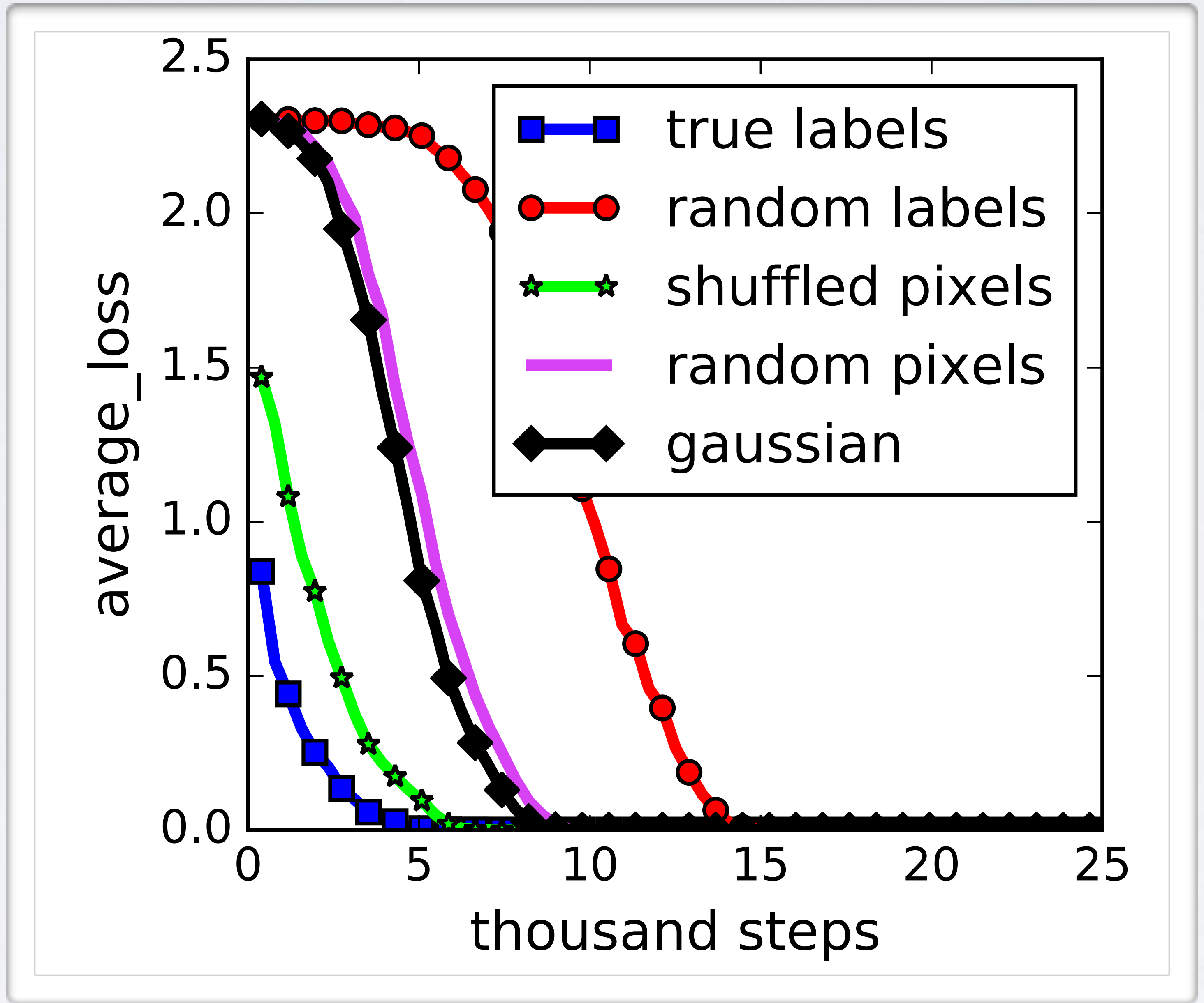
- *On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima* Keskar, Mudigere, Nocedal, Smelyanskiy, Tang, ICLR 2017
  - found that using large batch sizes tends to find sharper minima and generalize worse

- This means that we can't talk about generalization without taking the training algorithm into account

THEY CAN EASILY MEMORIZE

Topics: model capacity vs. training algorithm

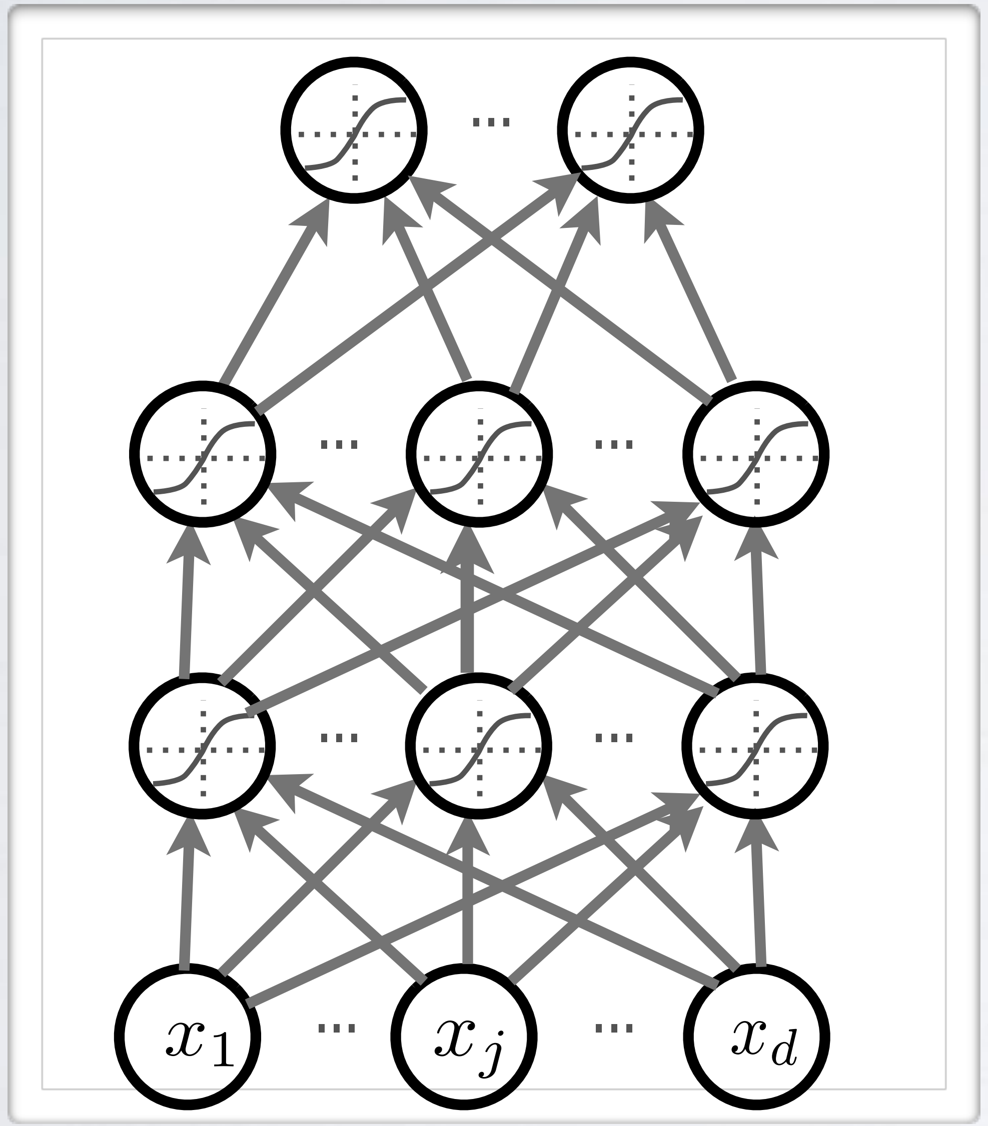
- *Understanding Deep Learning Requires Rethinking Generalization*
  Zhang, Bengio, Hardt, Recth, Vinyals, ICLR 2017



THEY CAN BE COMPRESSED

Topics: knowledge distillation

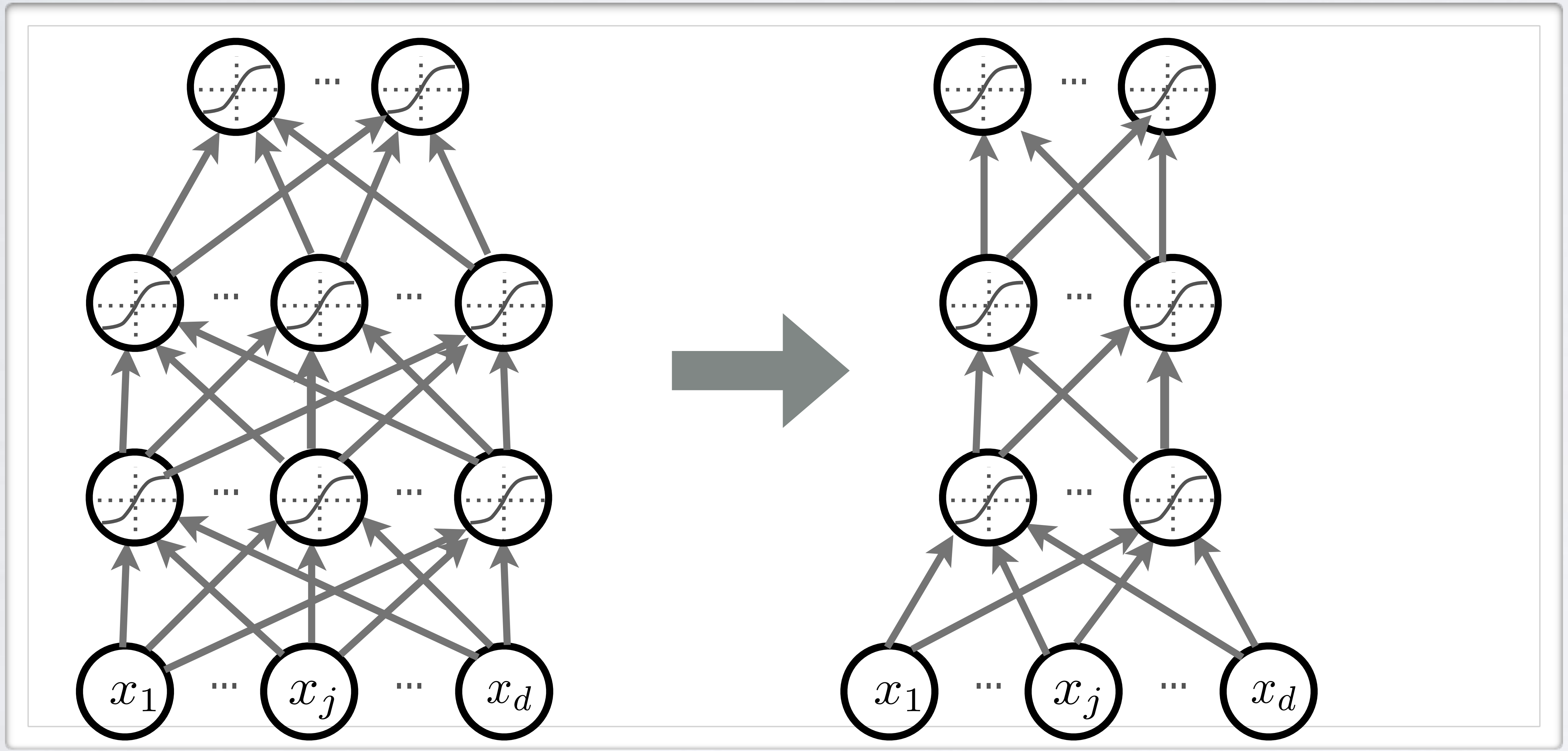
- *Distilling the Knowledge in a Neural Network*
  Hinton, Vinyals, Dean, arXiv 2015



THEY CAN BE COMPRESSED

Topics: knowledge distillation

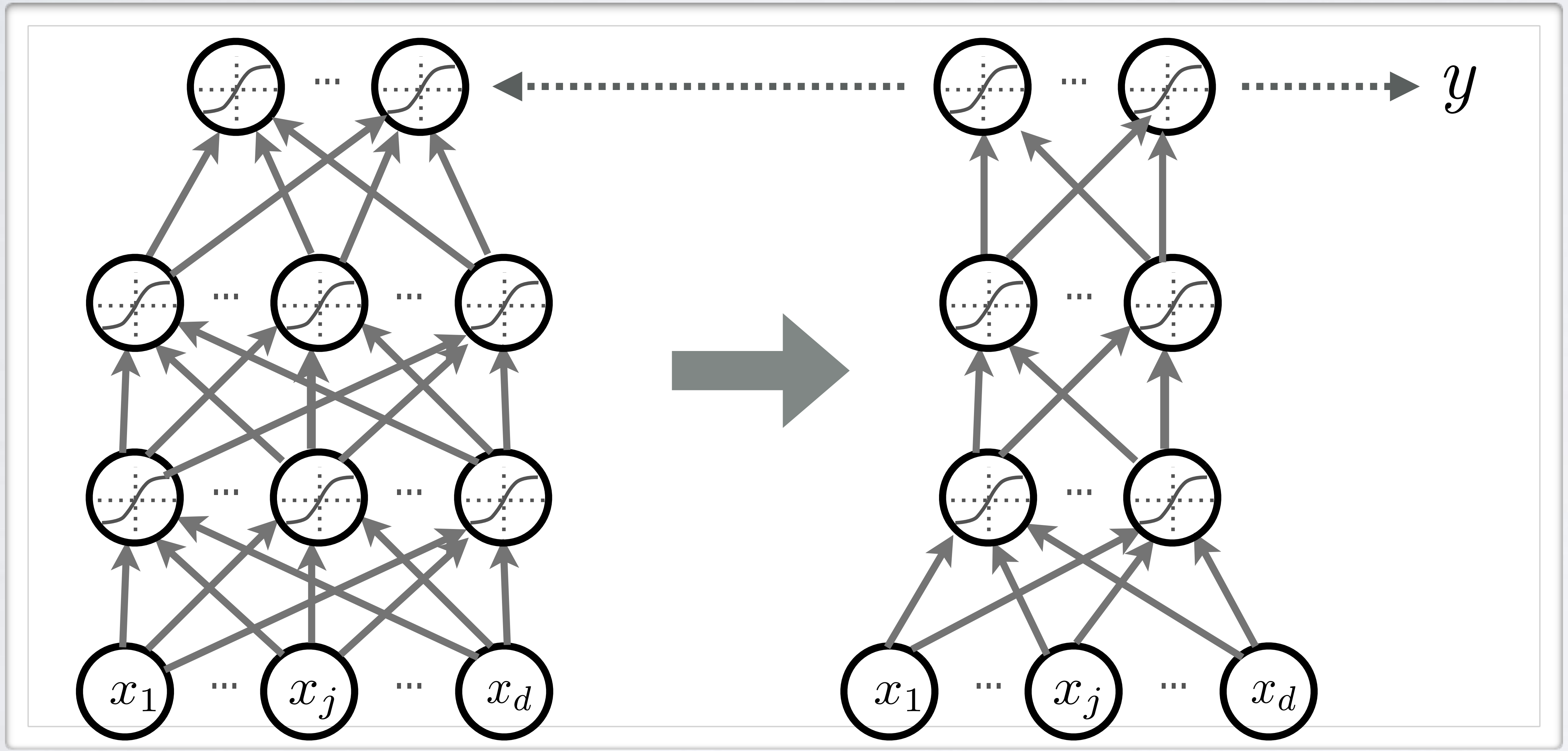
- *Distilling the Knowledge in a Neural Network*
  Hinton, Vinyals, Dean, arXiv 2015



THEY CAN BE COMPRESSED

Topics: knowledge distillation

- *Distilling the Knowledge in a Neural Network*
  Hinton, Vinyals, Dean, arXiv 2015



THEY CAN BE COMPRESSED

Topics: knowledge distillation

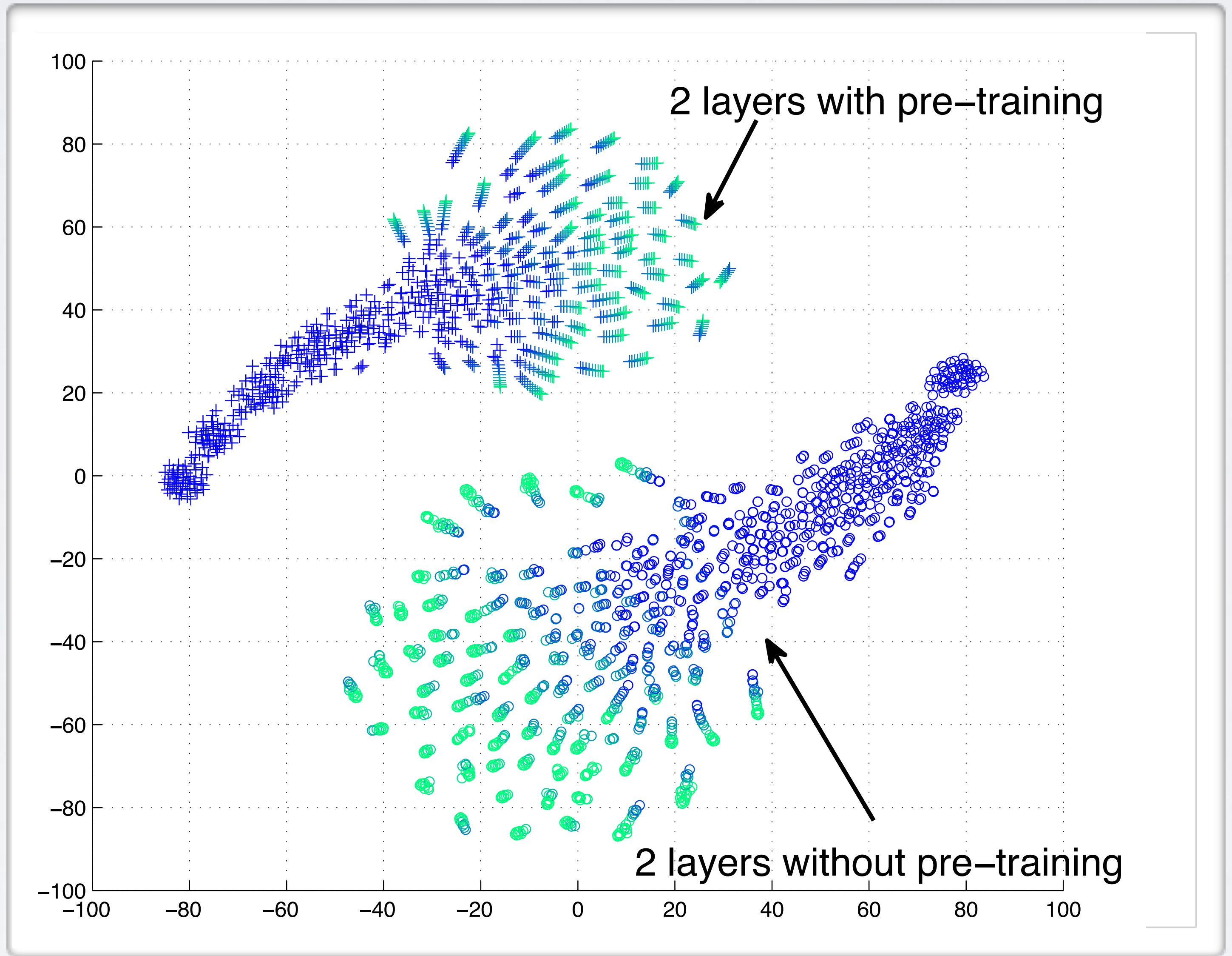
- Can successfully distill
  - a large neural network
  - an ensemble of neural network

- Works better than training it from scratch!
  - *Do Deep Nets Really Need to be Deep?*
    Jimmy Ba, Rich Caruana, NIPS 2014

THEY ARE INFLUENCED BY INITIALIZATION

Topics: impact of initialization

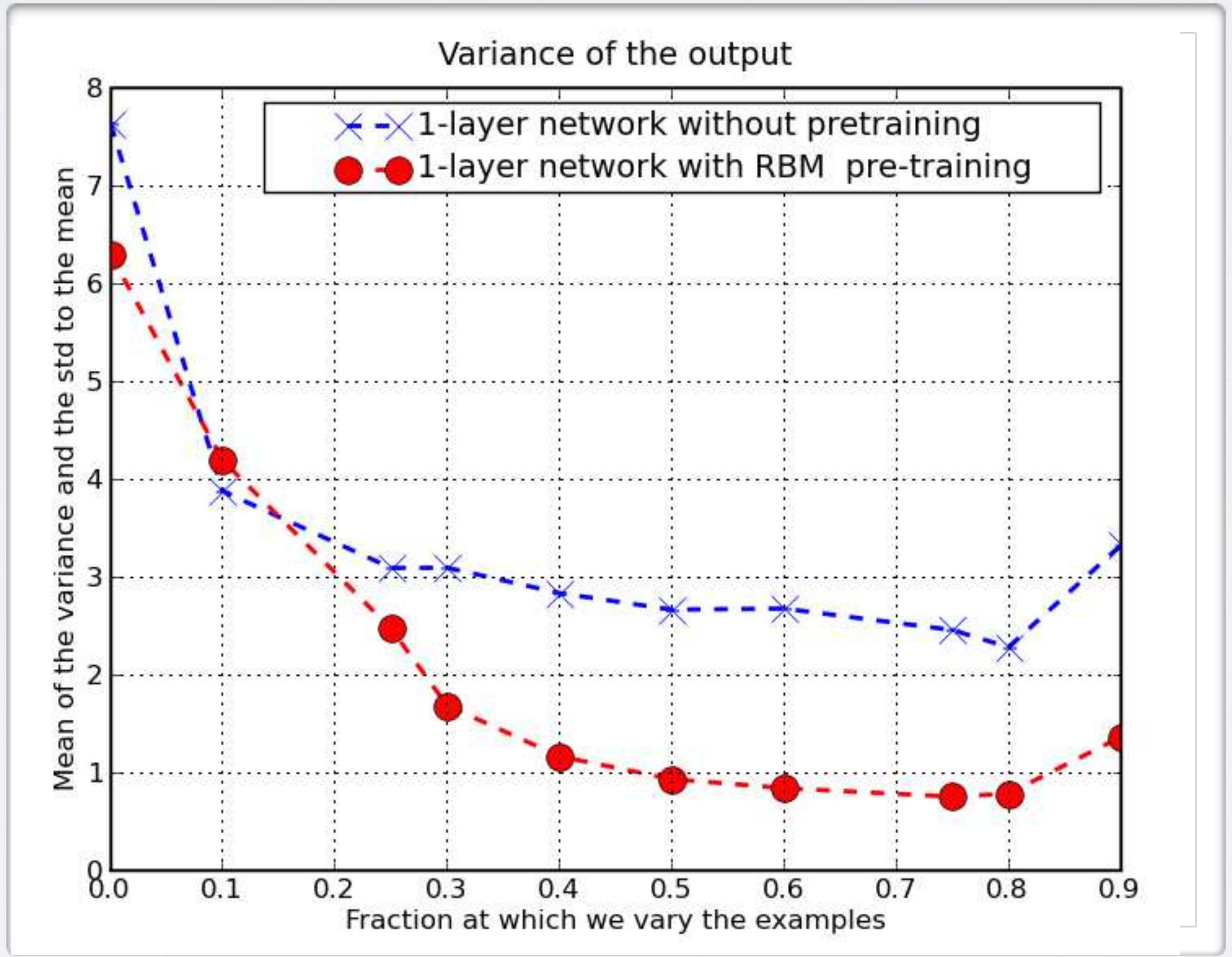
- *Why Does Unsupervised Pre-Training Help Deep Learning*
  Erhan, Bengio, Courville, Manzagol, Vincent, JMLR 2010



THEY ARE INFLUENCED BY FIRST EXAMPLES

Topics: impact of early examples

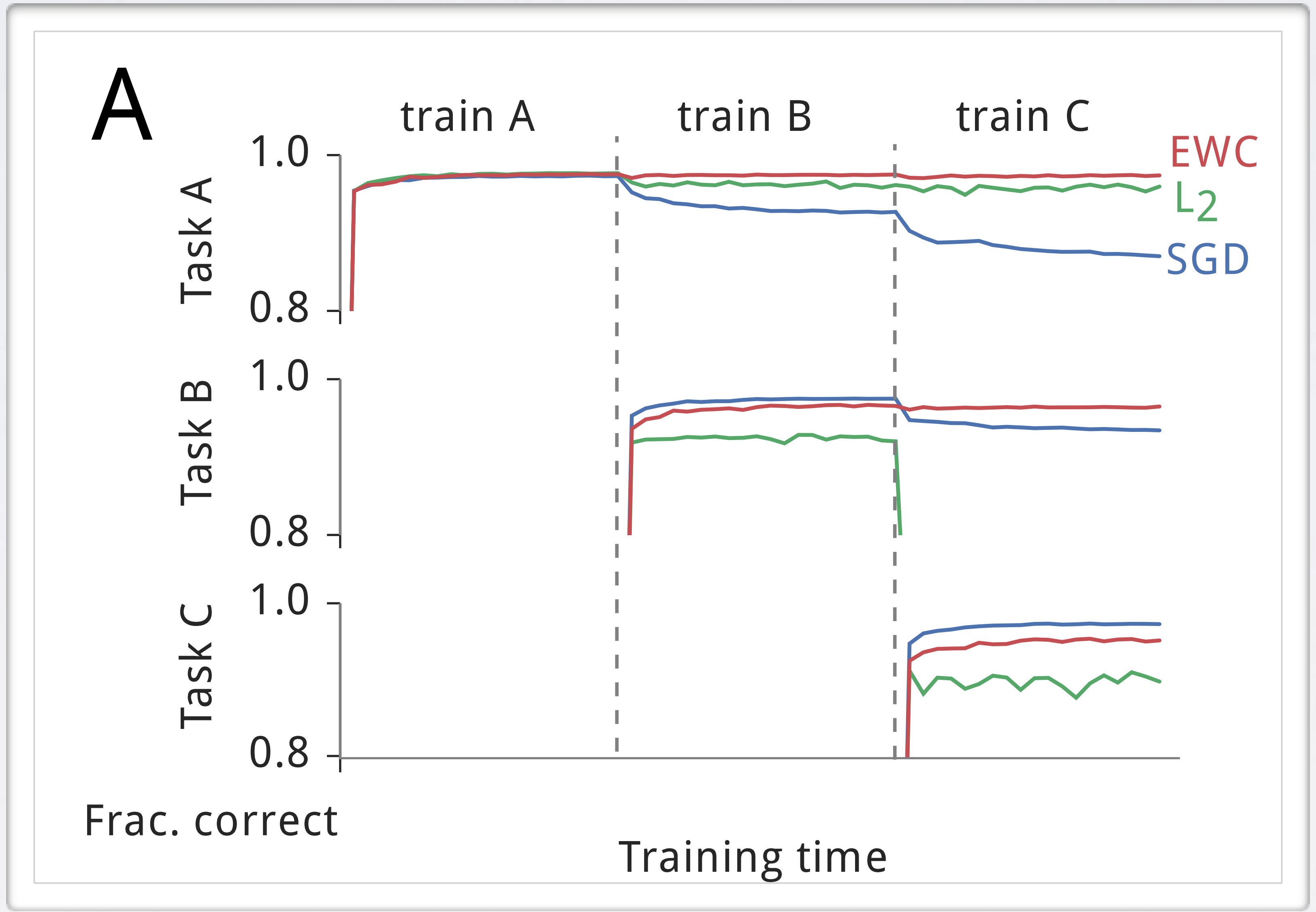
- *Why Does Unsupervised Pre-Training Help Deep Learning*
  Erhan, Bengio, Courville, Manzagol, Vincent, JMLR 2010



YET THEY FORGET WHAT THEY LEARNED

Topics: lifelong learning, continual learning

- *Overcoming Catastrophic Forgetting in Neural Networks*
  Kirkpatrick et al. PNAS 2017



SO THERE IS A LOT MORE TO UNDERSTAND!!

MERCI!