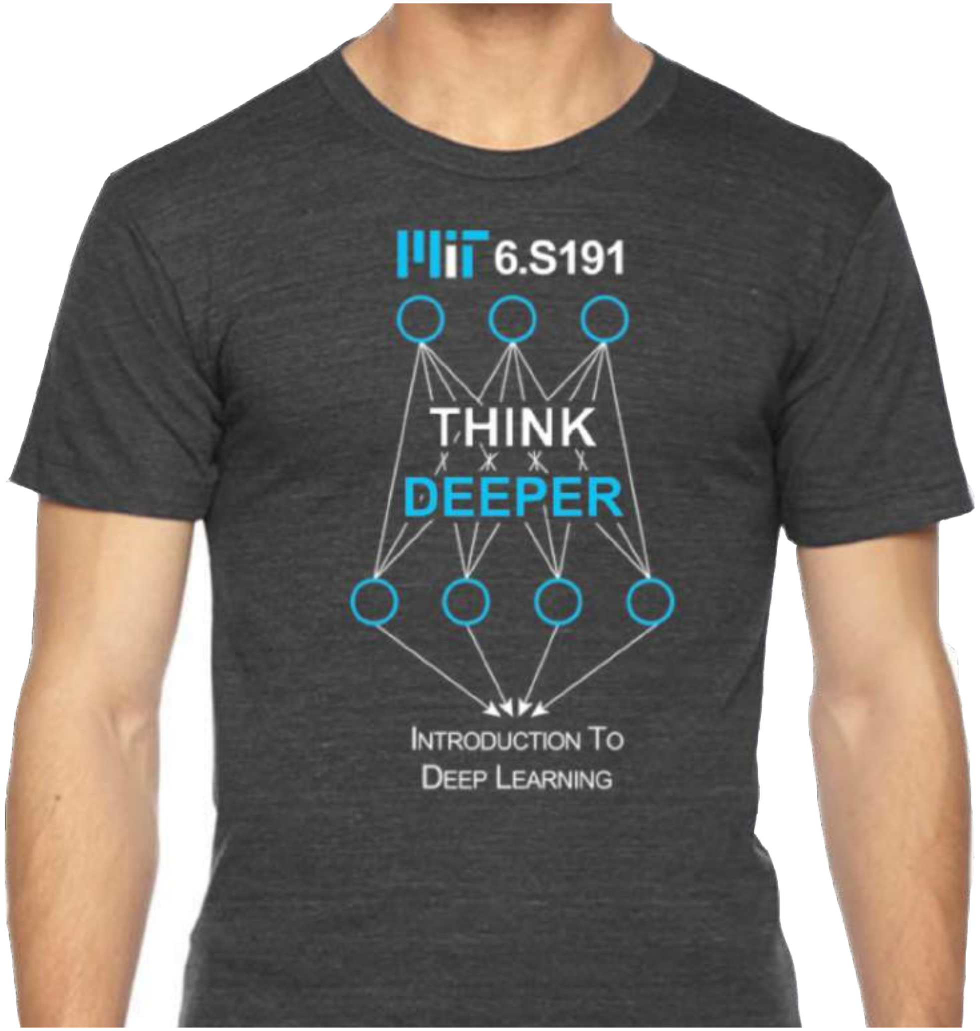
Deep Learning: Limitations and New Frontiers

MIT 6.S191

Ava Soleimany

January 30, 2019

T-shirts! Today!



Course Schedule

Session	Part 1	Part 2	Lab
1	 Introduction to Deep Learning [Slides] [Video] *coming soon!*	 Deep Sequence Modeling [Slides] [Video] *coming soon!*	 Intro to TensorFlow, Music Generation with RNNs [Code] *coming soon!*
2	 Deep Computer Vision [Slides] [Video] *coming soon!*	 Deep Generative Models [Slides] [Video] *coming soon!*	 De-biasing Facial Recognition Systems [Code] *coming soon!*
3	 Deep Reinforcement Learning [Slides] [Video] *coming soon!*	 Limitations and New Frontiers [Slides] [Video] *coming soon!*	 Model-Free Reinforcement Learning [Code] *coming soon!*
4	 Data Visualization for Machine Learning [Info][Slides] [Video] *coming soon!*	 Biologically Inspired Learning [Info][Slides] [Video] *coming soon!*	 Work time for paper reviews/project proposals
5	 Learing and Perception [Info][Slides] [Video] *coming soon!*	 Final Project Presentations	 Judging and Awards Ceremony

Final Class Project

Option 1: Proposal Presentation

- Present a novel deep learning research idea or application
- Groups of 1 welcome
- Listeners welcome
- Groups of 2 to 4 to be eligible for prizes, incl. 1 for-credit student
- 3 minutes
- Proposal instructions:

  goo.gl/JGJ5E7

- Judged by a panel of industry judges
- Top winners are awarded:

3x NVIDIA RTX 2080 Ti
MSRP: $4000

4x Google Home
MSRP: $400

Final Class Project

Option 1: Proposal Presentation

- Present a novel deep learning research idea or application
- Groups of 1 welcome
- Listeners welcome
- Groups of 2 to 4 to be eligible for prizes, incl. 1 for-credit student
- 3 minutes
- Proposal instructions:
  goo.gl/JGJ5E7

Proposal Logistics

- >= 1 for-credit student to be eligible for prizes
- Prepare slides on Google Slides
- **Group submit by today 10pm:**
  goo.gl/rV6rLK
- In class project work: **Thu, Jan 31**
- **Slide submit by Thu 11:59 pm:**
  goo.gl/7smL8w
- Presentations on **Friday, Feb 1**

Final Class Project

Option 1: Proposal Presentation

- Present a novel deep learning research idea or application
- Groups of 1 welcome
- Listeners welcome
- Groups of 2 to 4 to be eligible for prizes, incl. 1 for-credit student
- 3 minutes
- Proposal instructions:

  goo.gl/JGJ5E7

Option 2: Write a 1-page review of a deep learning paper

- Grade is based on clarity of writing and technical communication of main ideas
- Due **Friday 1:00pm** (before lecture)

Thursday: Visualization in ML + Biologically Inspired Learning

Fernanda Viegas,
Co-Director Google PAIR
Data Visualization for Machine Learning

Google

Dmitry Krotov,
MIT-IBM Watson AI Lab
Biologically Inspired Deep Learning

IBM Research

Final project work

Ask us questions!

Open office hours!

Work with group members!

Friday: Learning and Perception + Project Proposals + Awards + Pizza

Jan Kautz,
VP of Research
Learning and Perception

Project Proposals!

Judging and Awards!

Pizza Celebration!

So far in 6.S191…

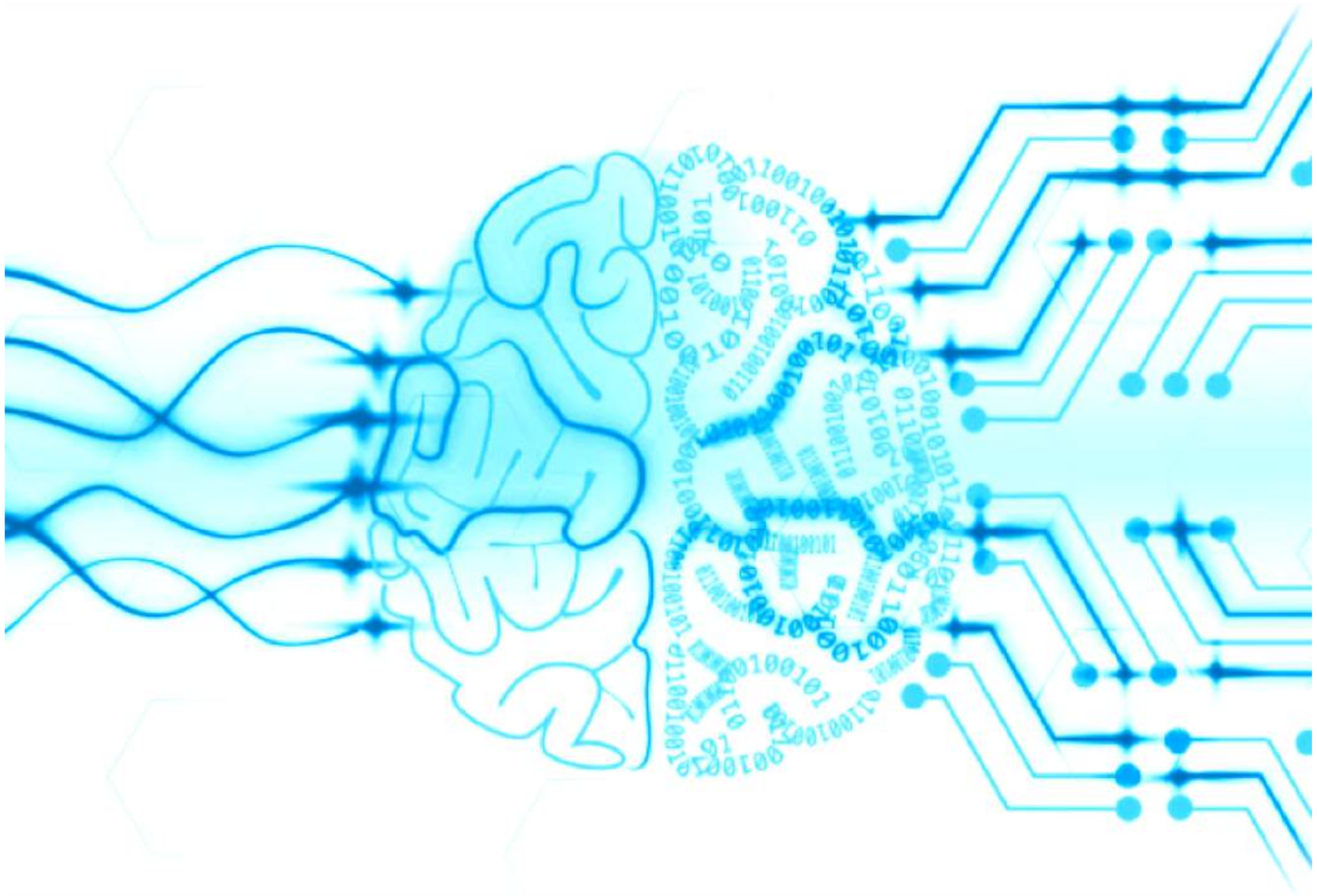
The Rise of Deep Learning

So far in 6.S191…

Data

- Signals
- Images
- Sensors

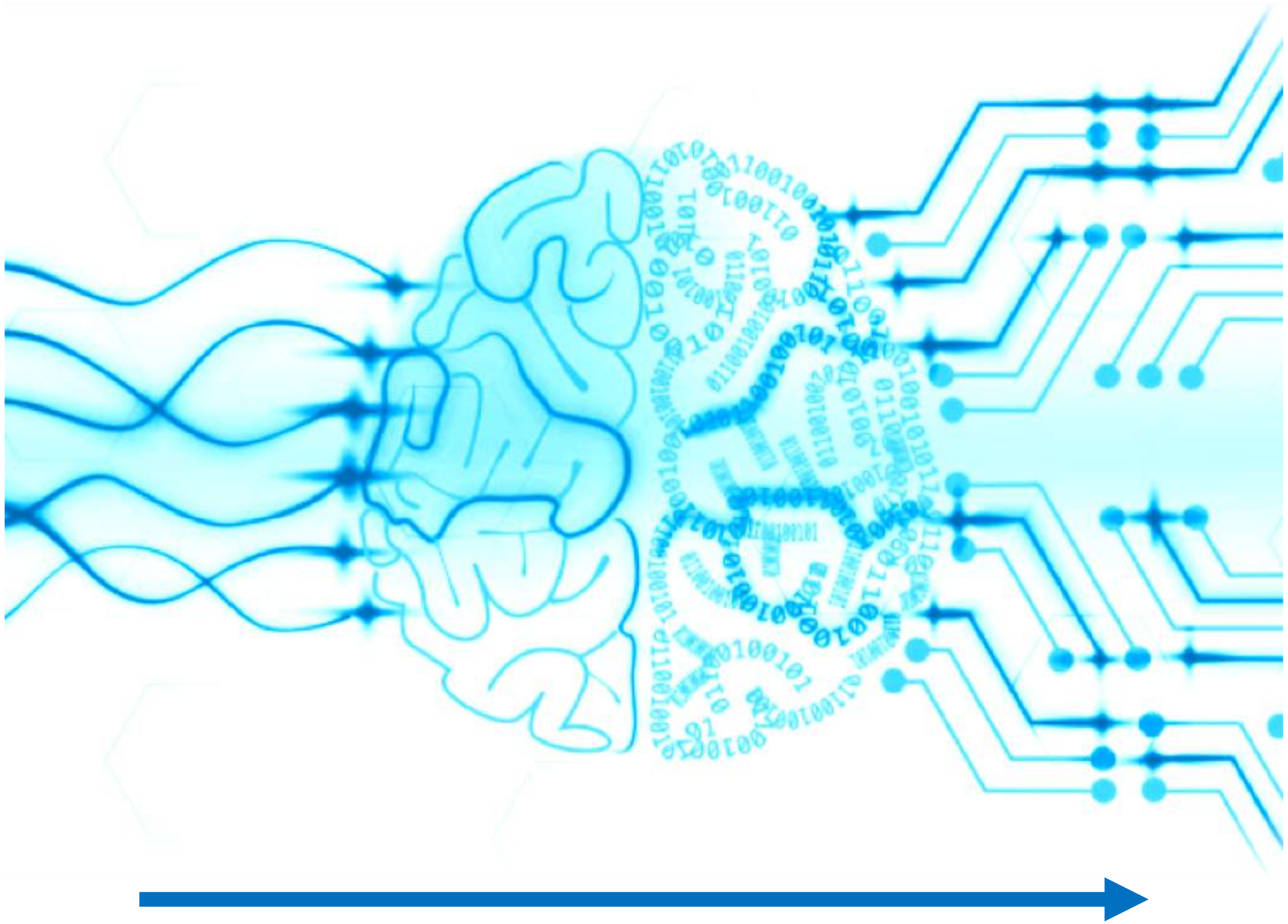
…



So far in 6.S191…

Data
- Signals
- Images
- Sensors

…



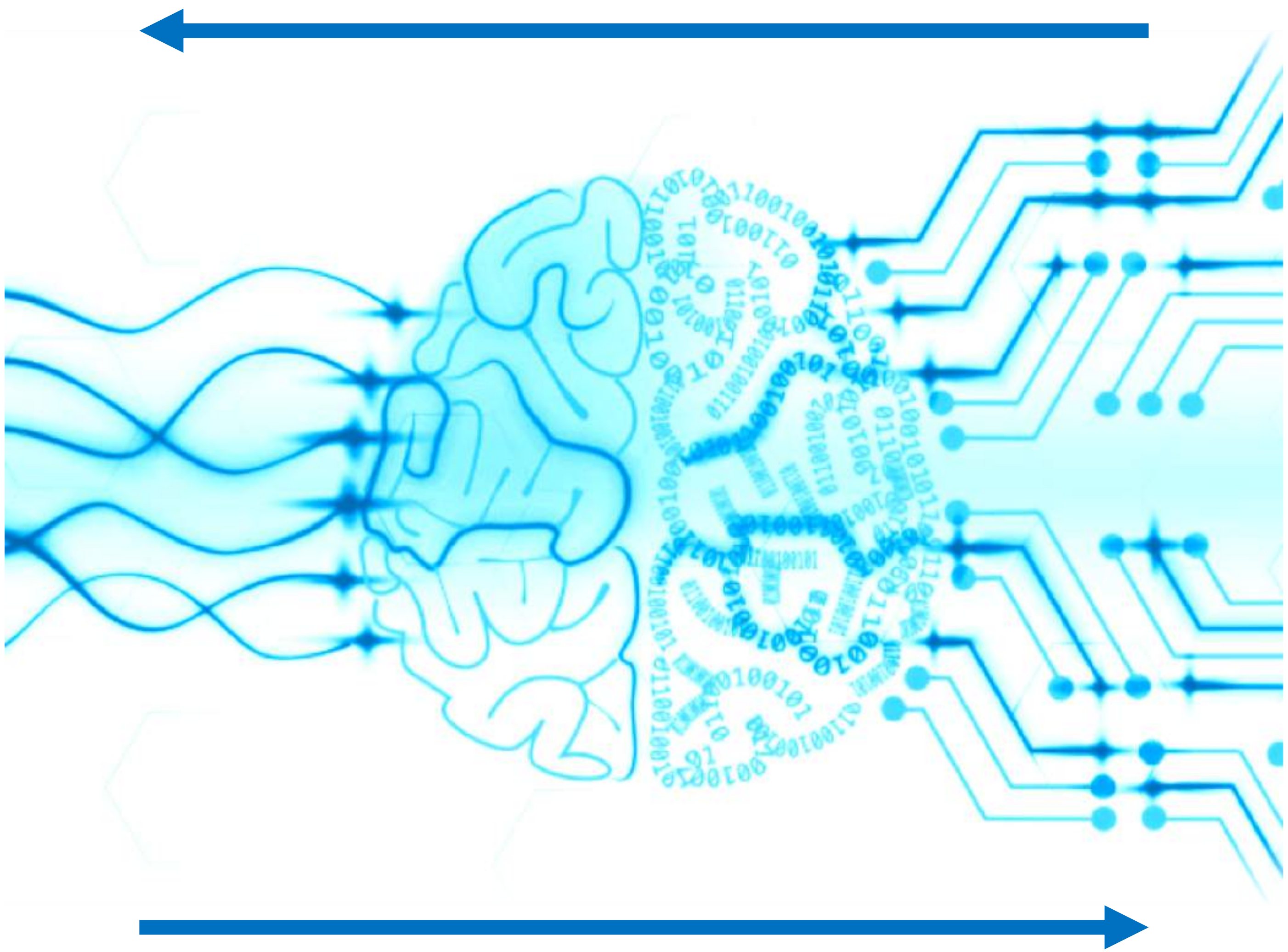
Decision
- Prediction
- Detection
- Action

…

So far in 6.S191…

Data
- Signals
- Images
- Sensors

…



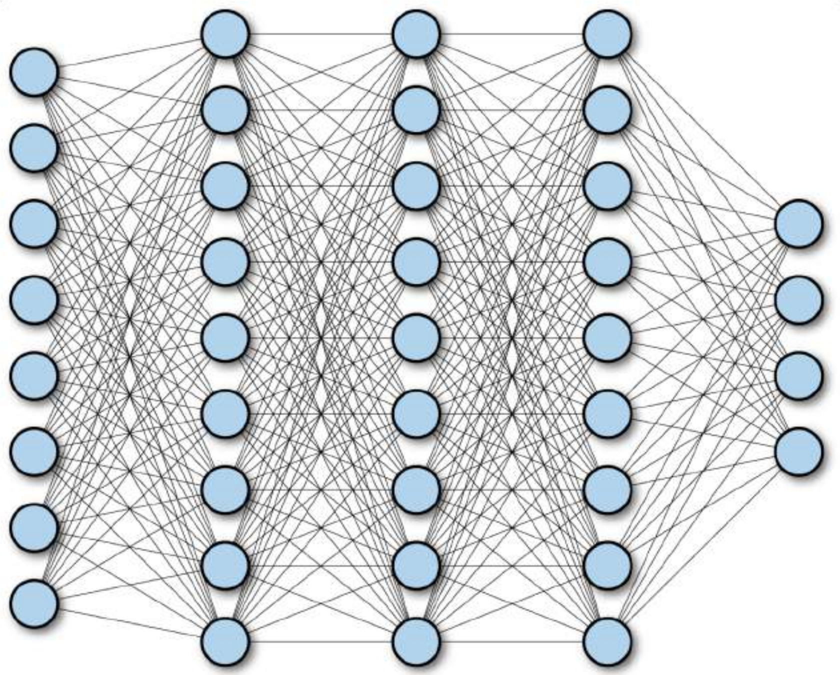
Decision
- Prediction
- Detection
- Action

…

Power of Neural Nets

Universal Approximation Theorem

A feedforward network with a single layer is sufficient to approximate, to an arbitrary precision, any continuous function.



Hornik et al. *Neural Networks.* (1989)

Power of Neural Nets

Universal Approximation Theorem

A feedforward network with a single layer is sufficient to approximate, to an arbitrary precision, any continuous function.

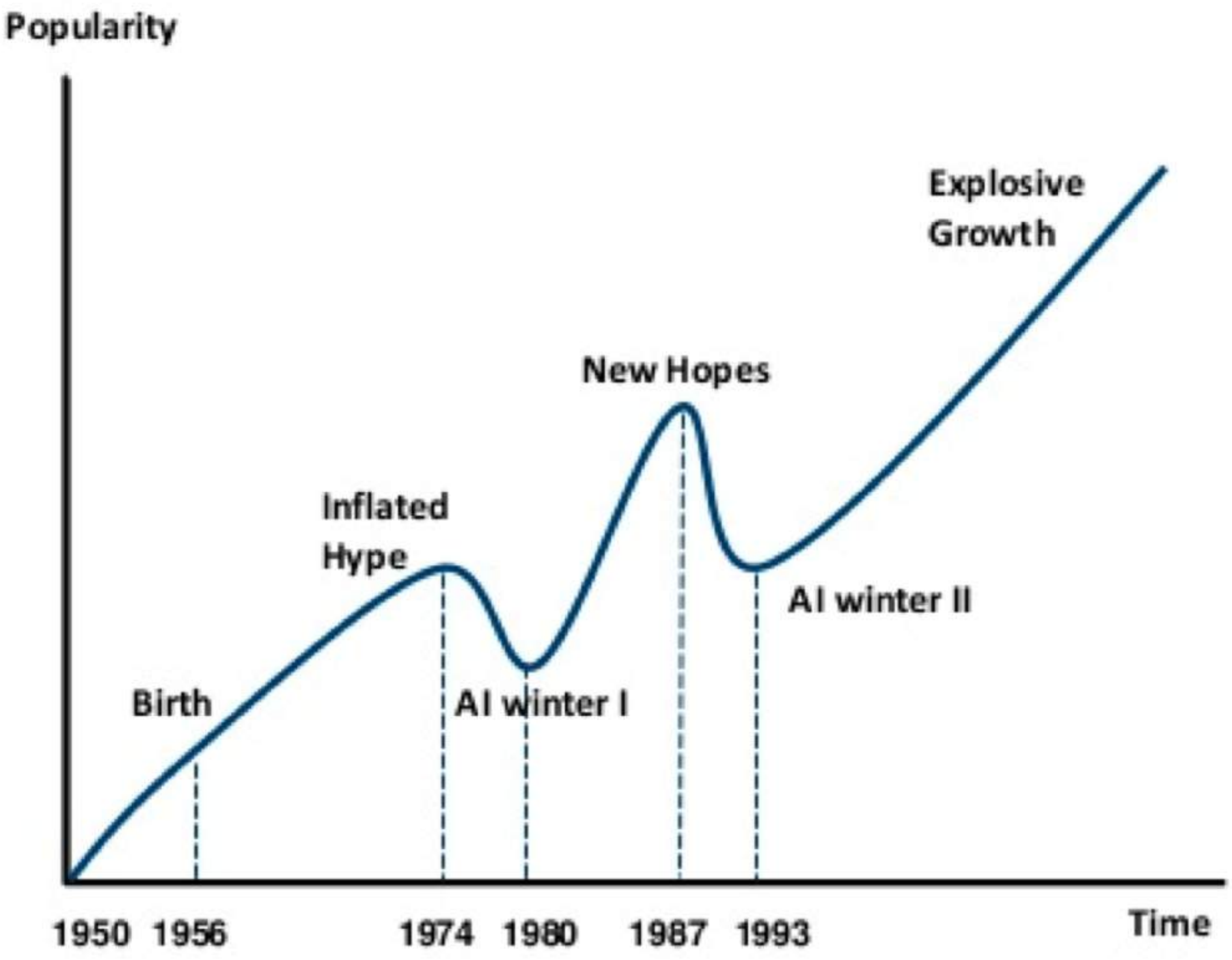
Caveats:

The number of hidden units may be infeasibly large

The resulting model may not generalize

Hornik et al. *Neural Networks.* (1989)

Artificial Intelligence "Hype": Historical Perspective



Limitations

Rethinking Generalization

"Understanding Deep Neural Networks Requires Rethinking Generalization"

dog

banana

dog

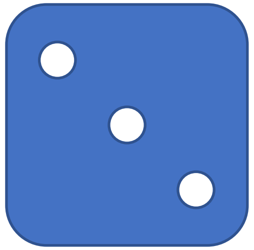
tree

Zhang et al. *ICLR*. (2017)

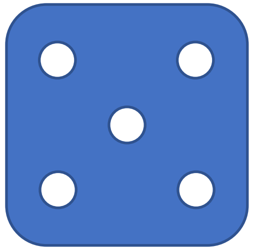
Rethinking Generalization

"Understanding Deep Neural Networks Requires Rethinking Generalization"

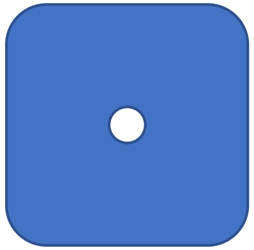
dog



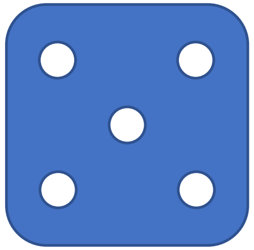
banana



dog



tree



Zhang et al. *ICLR.* (2017)

Rethinking Generalization

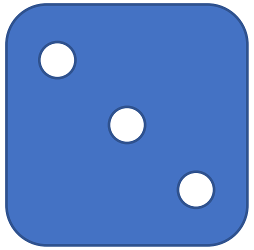
"Understanding Deep Neural Networks Requires Rethinking Generalization"

dog

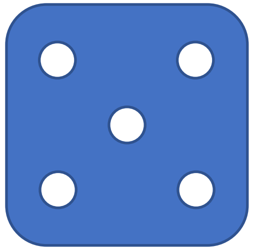
banana

dog

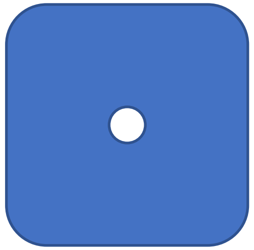
tree



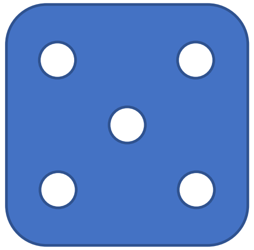
banana



dog



tree



dog

Zhang et al. *ICLR.* (2017)

Rethinking Generalization

"Understanding Deep Neural Networks Requires Rethinking Generalization"

~~dog~~

banana

~~banana~~

dog

~~dog~~

tree

~~tree~~

dog

Zhang et al. *ICLR.* (2017)

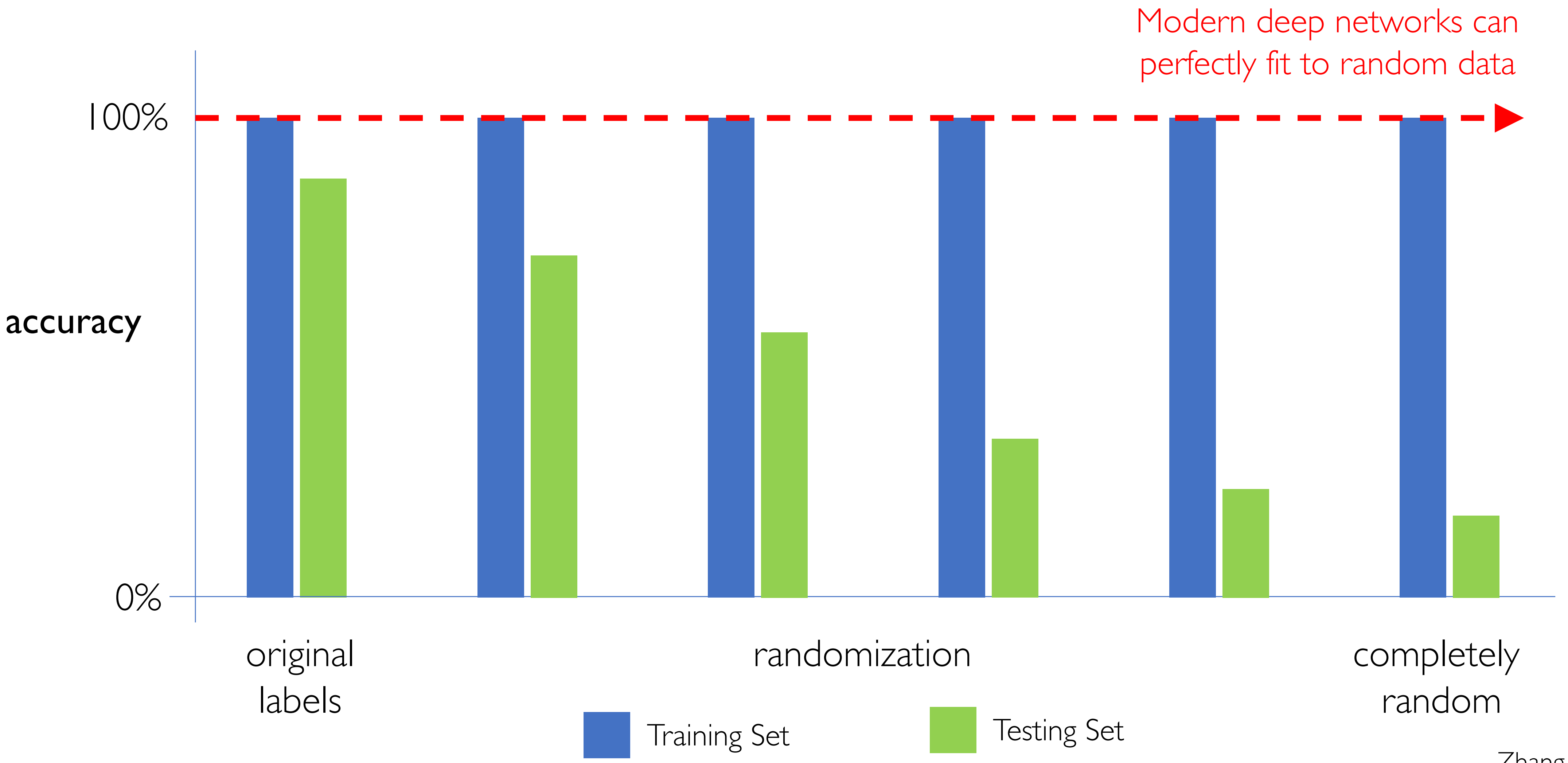
Capacity of Deep Neural Networks

Zhang et al. *ICLR.* (2017)

Capacity of Deep Neural Networks

Zhang et al. *ICLR.* (2017)

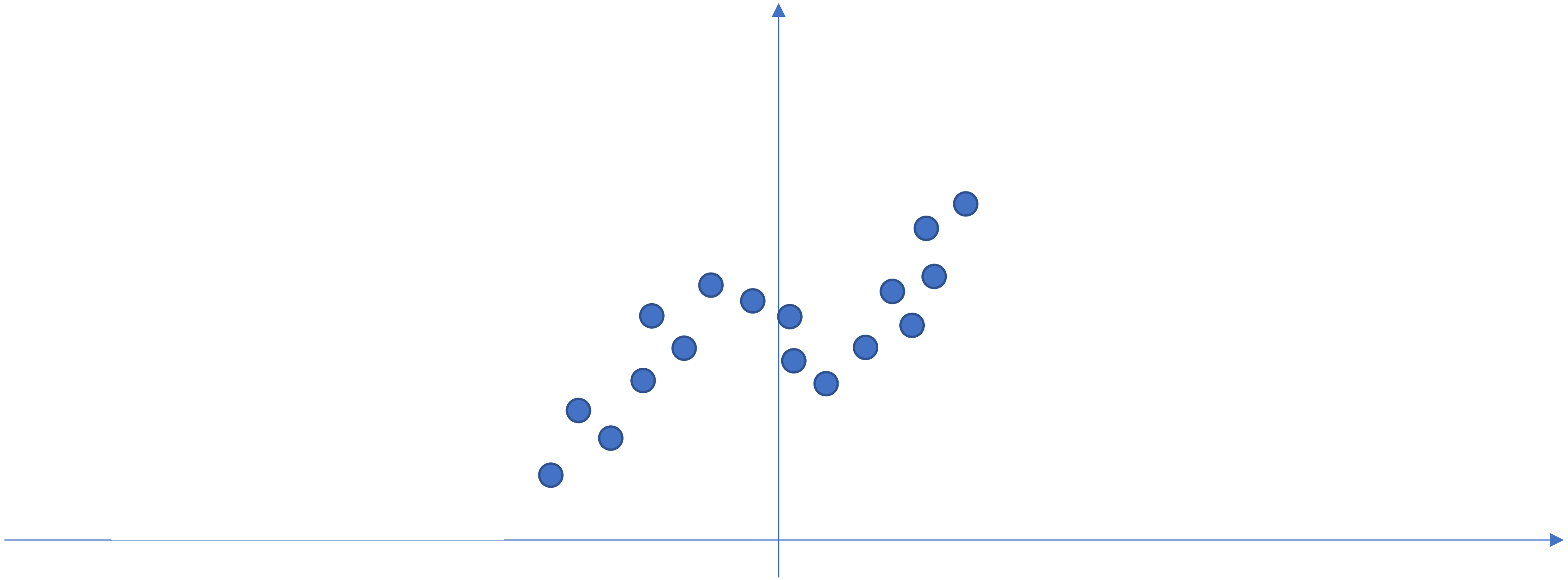
Capacity of Deep Neural Networks



Zhang et al. *ICLR.* (2017)

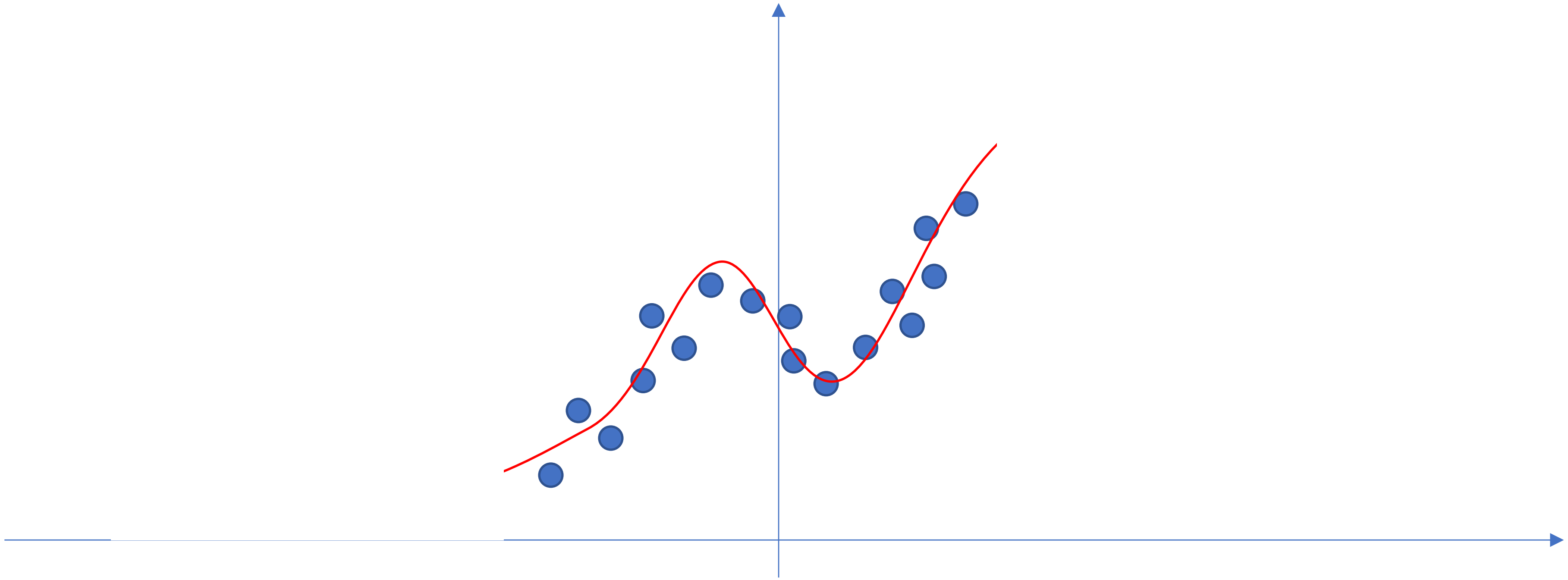
Neural Networks as Function Approximators

Neural networks are **excellent** function approximators



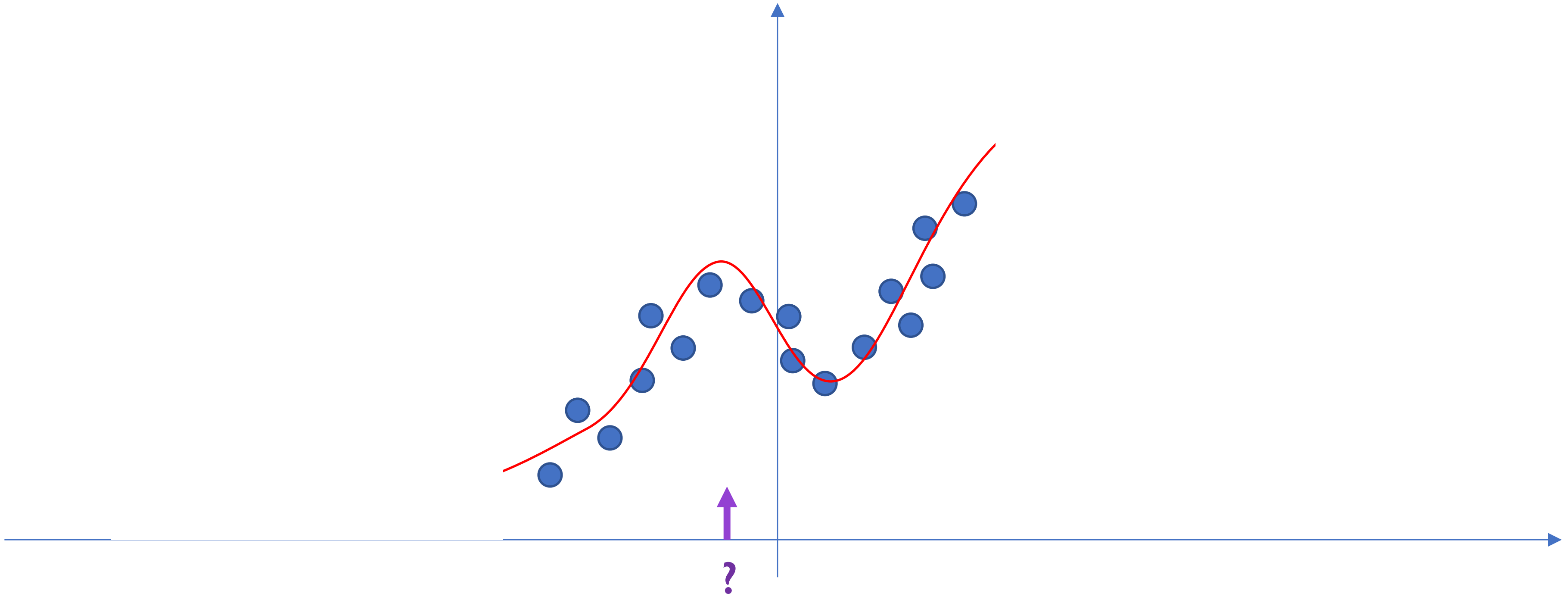
Neural Networks as Function Approximators

Neural networks are **excellent** function approximators



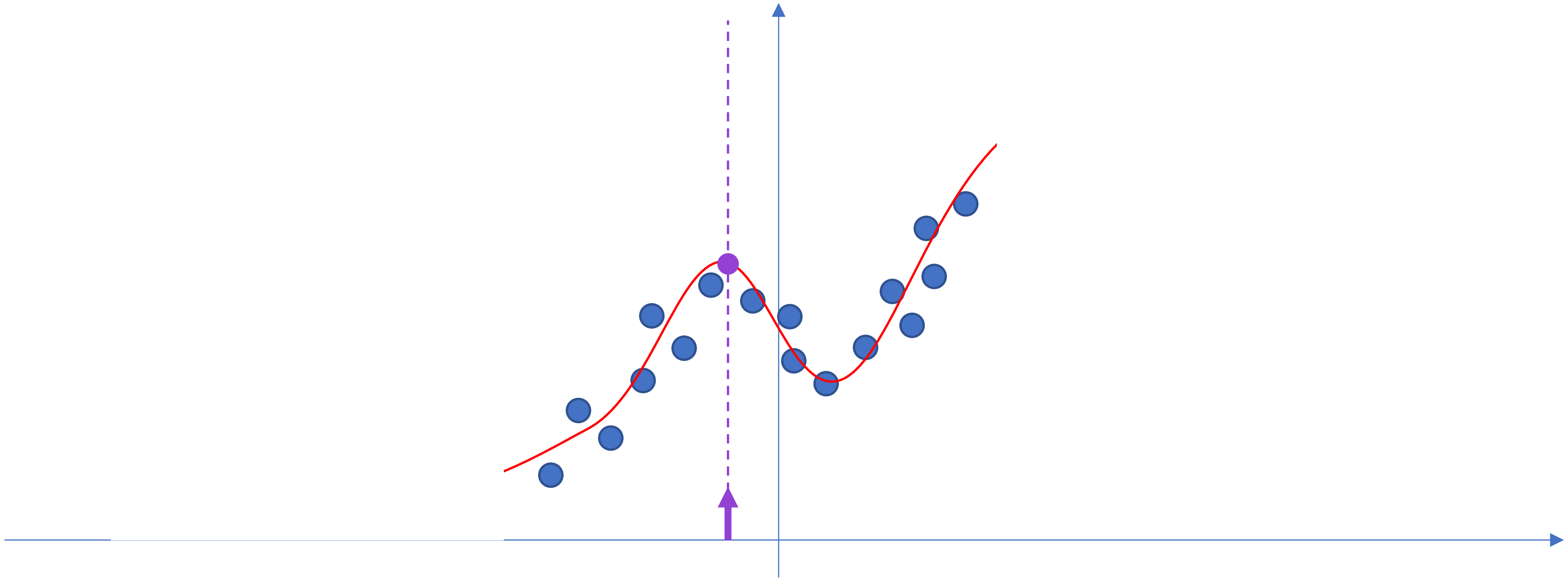
Neural Networks as Function Approximators

Neural networks are **excellent** function approximators



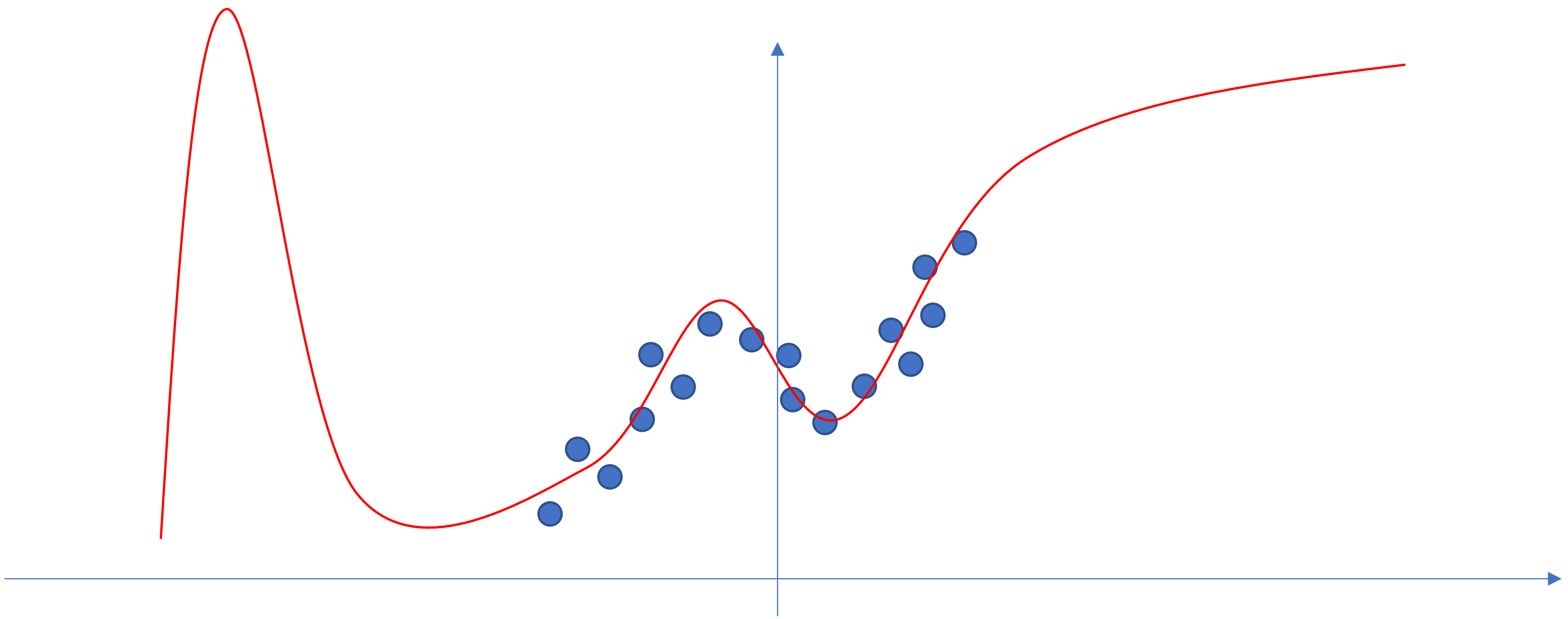
Neural Networks as Function Approximators

Neural networks are **excellent** function approximators



Neural Networks as Function Approximators

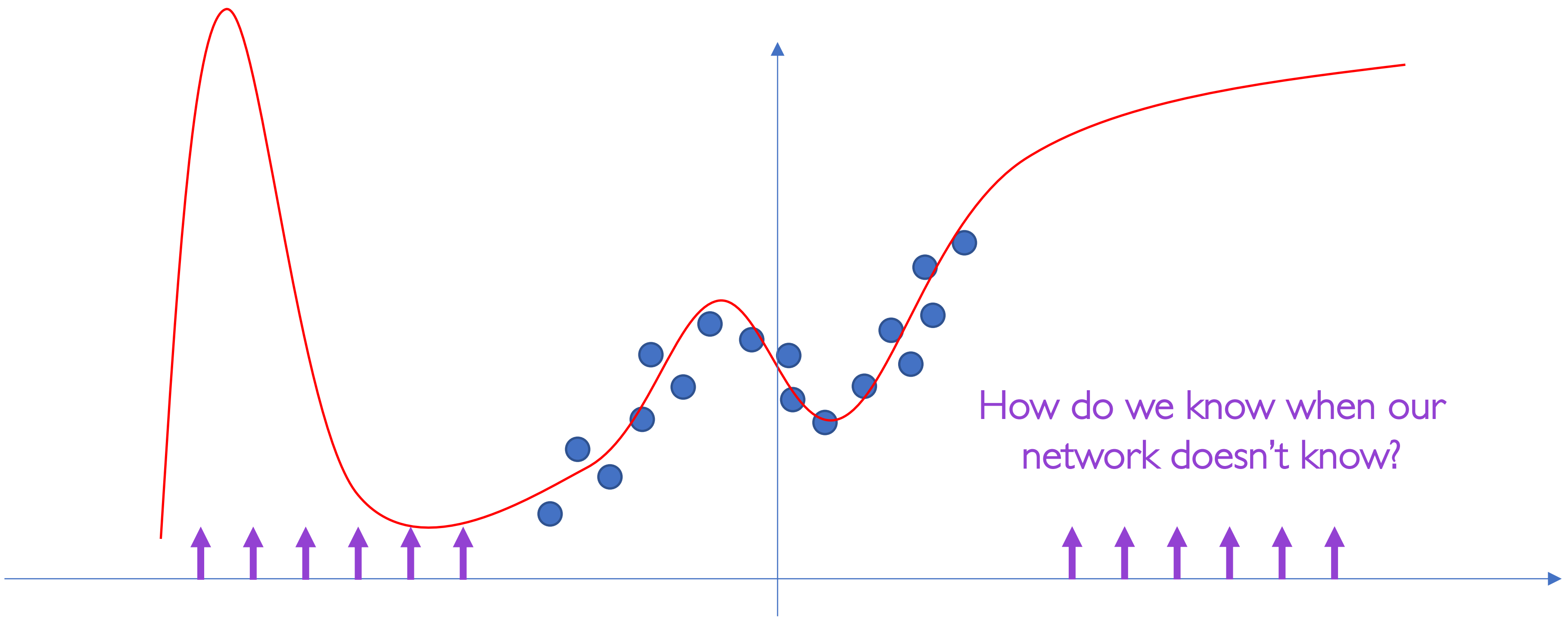
Neural networks are **excellent** function approximators



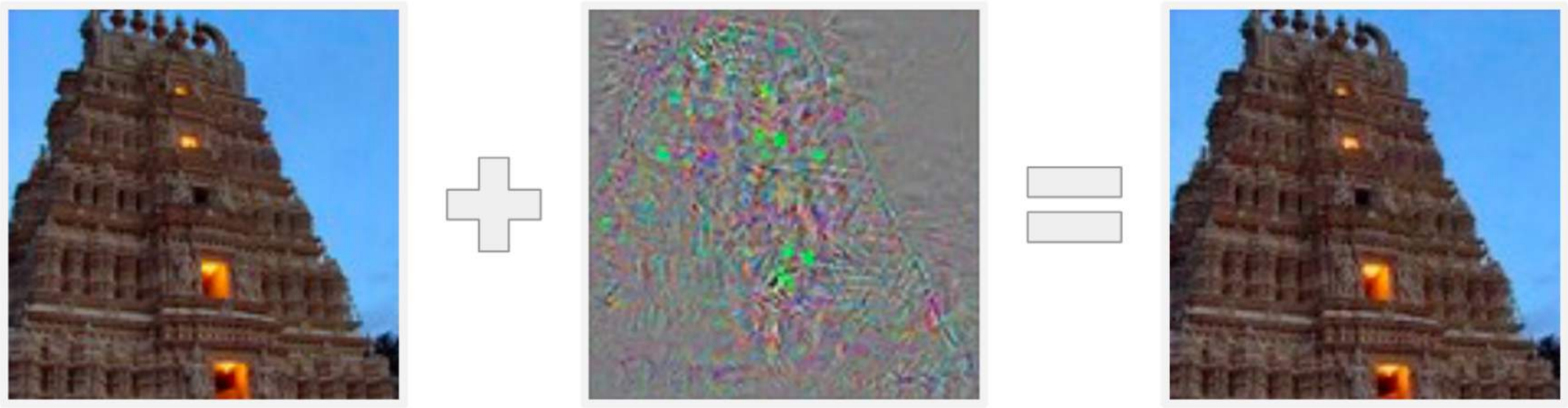
Neural Networks as Function Approximators

Neural networks are **excellent** function approximators

...when they have training data



Adversarial Attacks on Neural Networks



Original image

Temple (97%)

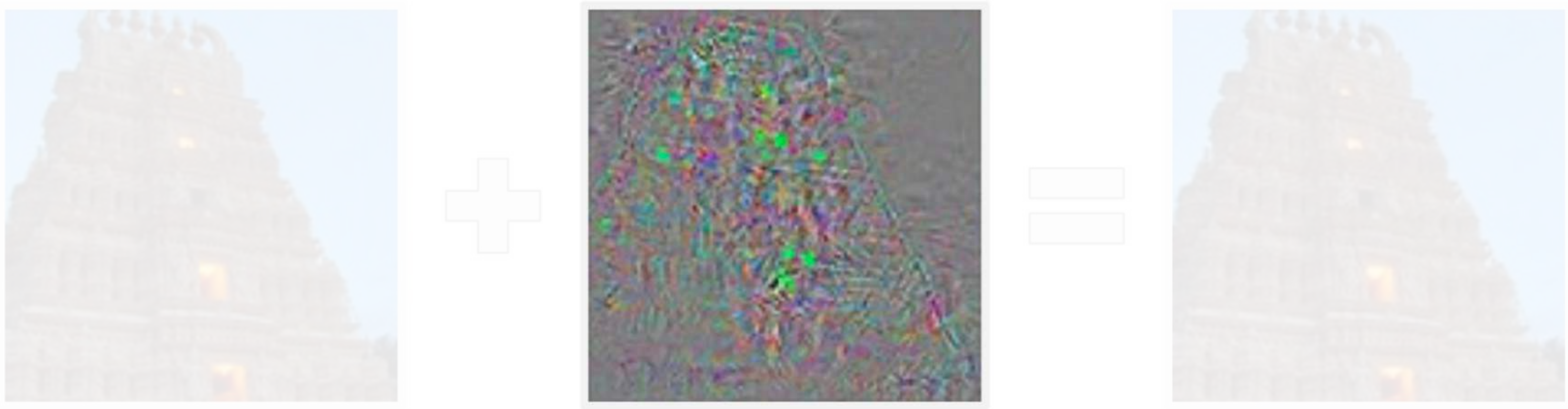
Perturbations

Adversarial example

Ostrich (98%)

Despois. "Adversarial examples and their implications" (2017).

Adversarial Attacks on Neural Networks



Original image
Temple (97%)

Perturbations

Adversarial example
Ostrich (98%)

Adversarial Attacks on Neural Networks

Remember:

We train our networks with gradient descent

$$\theta \leftarrow \theta - \eta \frac{\partial J(\theta, x, y)}{\partial \theta}$$

"How does a small change in weights decrease our loss"

Adversarial Attacks on Neural Networks

Remember:

We train our networks with gradient descent

$$\theta \leftarrow \theta - \eta \frac{\partial J(\theta, x, y)}{\partial \theta}$$

"How does a small change in weights decrease our loss"

Adversarial Attacks on Neural Networks

Remember:

We train our networks with gradient descent

$$\theta \leftarrow \theta - \eta \frac{\partial J(\theta, x, y)}{\partial \theta}$$

Fix your image x,
and true label y

"How does a small change in weights decrease our loss"

Adversarial Attacks on Neural Networks

Adversarial Image:

Modify image to increase error

$$x \leftarrow x + \eta \frac{\partial J(\theta, x, y)}{\partial x}$$

"How does a small change in the input increase our loss"

Goodfellow et al. *NIPS* (2014)

Adversarial Attacks on Neural Networks

Adversarial Image:

Modify image to increase error

$$x \leftarrow x + \eta \frac{\partial J(\theta, x, y)}{\partial x}$$

"How does a small change in the input increase our loss"

Adversarial Attacks on Neural Networks

Adversarial Image:

Modify image to increase error

$$x \leftarrow x + \eta \frac{\partial J(\theta, x, y)}{\partial x}$$

Fix your weights θ, and true label y

"How does a small change in the input increase our loss"

Synthesizing Robust Adversarial Examples

Athalye et al. *ICML*. (2018)

Neural Network Limitations…

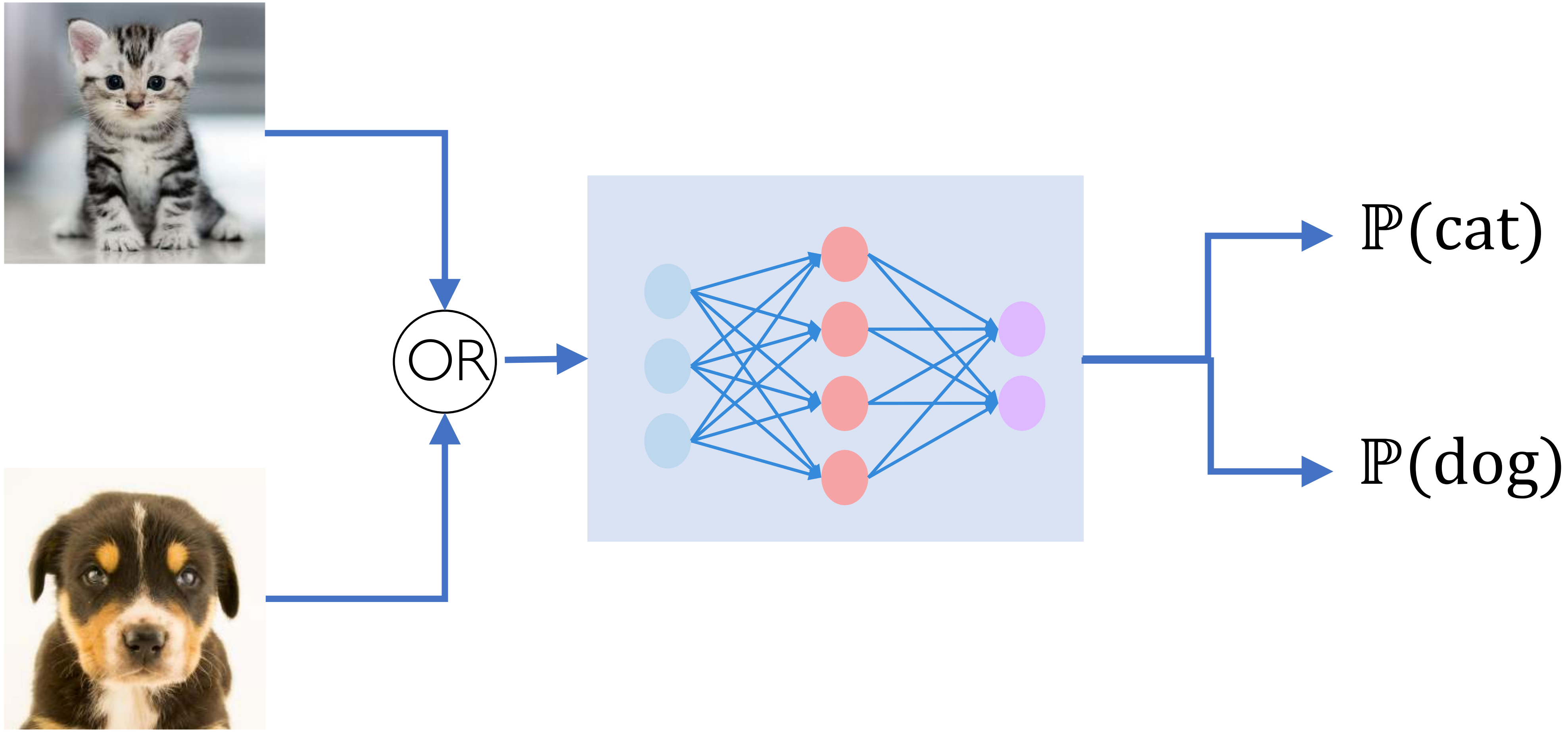
- Very **data hungry** (eg. often millions of examples)
- **Computationally intensive** to train and deploy (tractably requires GPUs)
- Easily fooled by **adversarial examples**
- Can be subject to **algorithmic bias**
- Poor at **representing uncertainty** (how do you know what the model knows?)
- Uninterpretable **black boxes**, difficult to trust
- **Finicky to optimize**: non-convex, choice of architecture, learning parameters
- Often require **expert knowledge** to design, fine tune architectures

Neural Network Limitations…

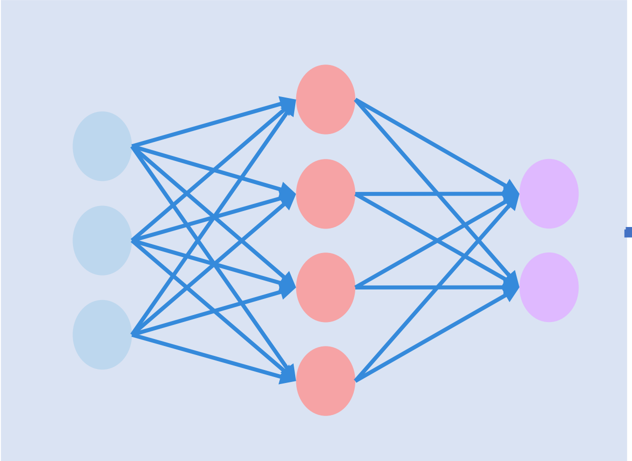
- Very **data hungry** (eg. often millions of examples)
- **Computationally intensive** to train and deploy (tractably requires GPUs)
- Easily fooled by **adversarial examples**
- Can be subject to **algorithmic bias**
- Poor at **representing uncertainty** (how do you know what the model knows?)
- Uninterpretable **black boxes**, difficult to trust
- **Finicky to optimize**: non-convex, choice of architecture, learning parameters
- Often require **expert knowledge** to design, fine tune architectures

New Frontiers 1:
Bayesian Deep Learning

Why Care About Uncertainty?



Why Care About Uncertainty?



$\mathbb{P}(\text{cat}) = 0.2$

$\mathbb{P}(\text{dog}) = 0.8$

Remember: $\mathbb{P}(\text{cat}) + \mathbb{P}(\text{dog}) = 1$

Bayesian Deep Learning for Uncertainty

Network tries to learn output, $\boldsymbol{Y}$, directly from raw data, $\boldsymbol{X}$

Find mapping, f, parameterized by weights $\boldsymbol{\theta}$ such that

$$\min \mathcal{L}(\boldsymbol{Y}, f(X; \boldsymbol{\theta}))$$

Bayesian neural networks aim to learn a posterior over weights, $\mathbb{P}(\boldsymbol{\theta}|\boldsymbol{X}, \boldsymbol{Y})$:

$$\mathbb{P}(\boldsymbol{\theta}|\boldsymbol{X}, \boldsymbol{Y}) = \frac{\mathbb{P}(\boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{\theta})\mathbb{P}(\boldsymbol{\theta})}{\mathbb{P}(\boldsymbol{Y}|\boldsymbol{X})}$$

Bayesian Deep Learning for Uncertainty

Network tries to learn output, $\boldsymbol{Y}$, directly from raw data, $\boldsymbol{X}$

Find mapping, f, parameterized by weights $\boldsymbol{\theta}$ such that

$$\min \mathcal{L}(\boldsymbol{Y}, f(X; \boldsymbol{\theta}))$$

Bayesian neural networks aim to learn a posterior over weights, $\mathbb{P}(\boldsymbol{\theta}|\boldsymbol{X}, \boldsymbol{Y})$:

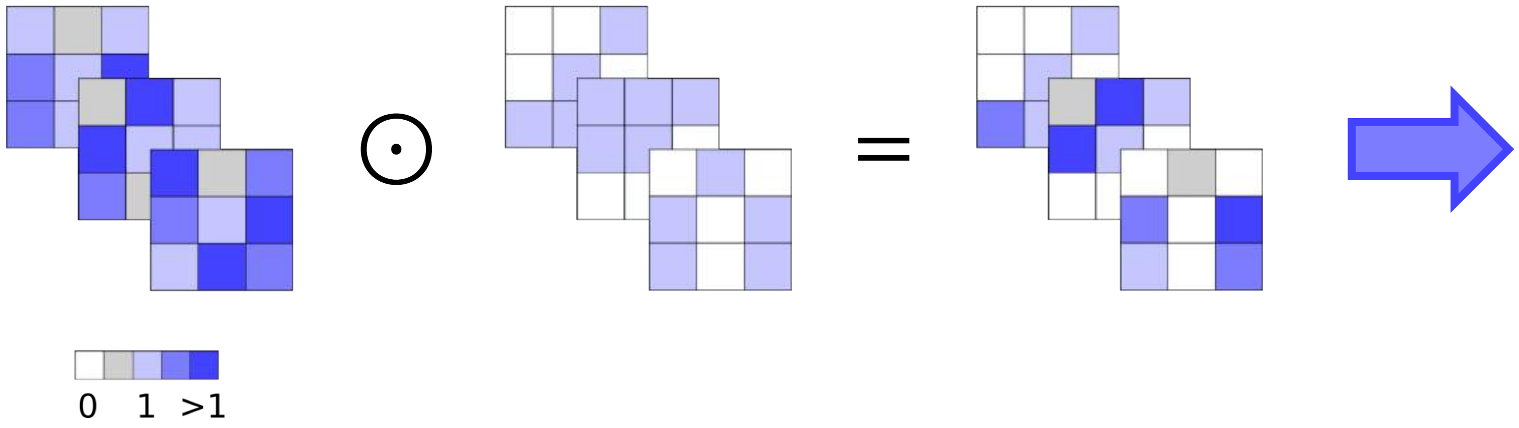
Intractable! $$\mathbb{P}(\boldsymbol{\theta}|\boldsymbol{X}, \boldsymbol{Y}) = \frac{\mathbb{P}(\boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{\theta})\mathbb{P}(\boldsymbol{\theta})}{\mathbb{P}(\boldsymbol{Y}|\boldsymbol{X})}$$

Elementwise Dropout for Uncertainty

Evaluate T stochastic forward passes through the network $\{\boldsymbol{\theta}_t\}_{t=1}^{T}$

Dropout as a form of stochastic sampling $\quad z_{w,t} \sim \mathit{Bernoulli}(p) \quad \forall\, w \in \boldsymbol{\theta}$

Unregularized Kernel $\boldsymbol{\theta}$ | Bernoulli Dropout $z_{\boldsymbol{\theta},t}$ | Stochastic Sampled $\boldsymbol{\theta}_t$



$$\mathbb{E}(\widehat{\boldsymbol{Y}}|\boldsymbol{X}) = \frac{1}{T}\sum_{t=1}^{T} f(\boldsymbol{X}|\boldsymbol{\theta}_t)$$

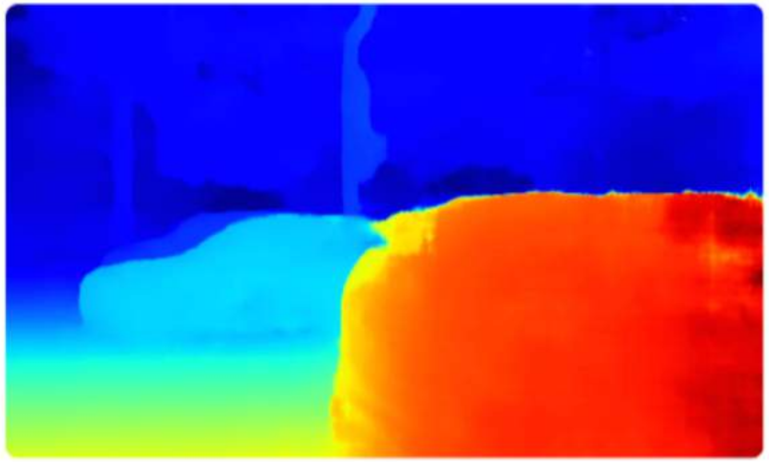
$$Var(\widehat{\boldsymbol{Y}}|\boldsymbol{X}) = \frac{1}{T}\sum_{t=1}^{T} f(\boldsymbol{X})^2 - \mathbb{E}(\widehat{\boldsymbol{Y}}|\boldsymbol{X})^2$$

Gal and Ghahramani, *ICML*, 2016.

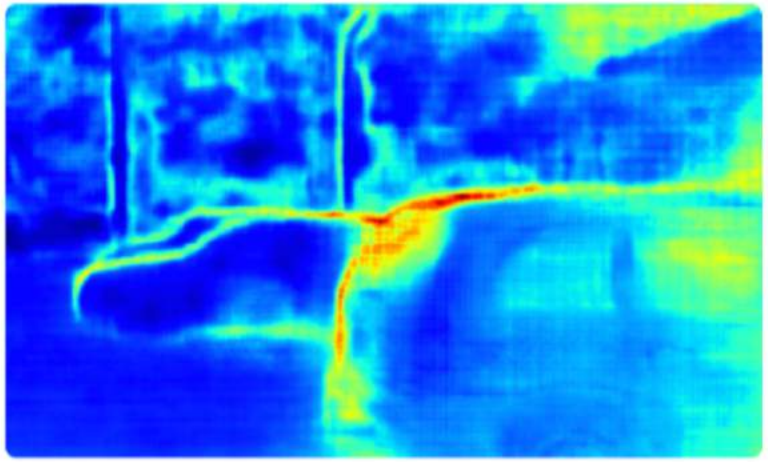
Amini, Soleimany, et al., *NIPS Workshop on Bayesian Deep Learning*, 2017.

Model Uncertainty Application

Input image



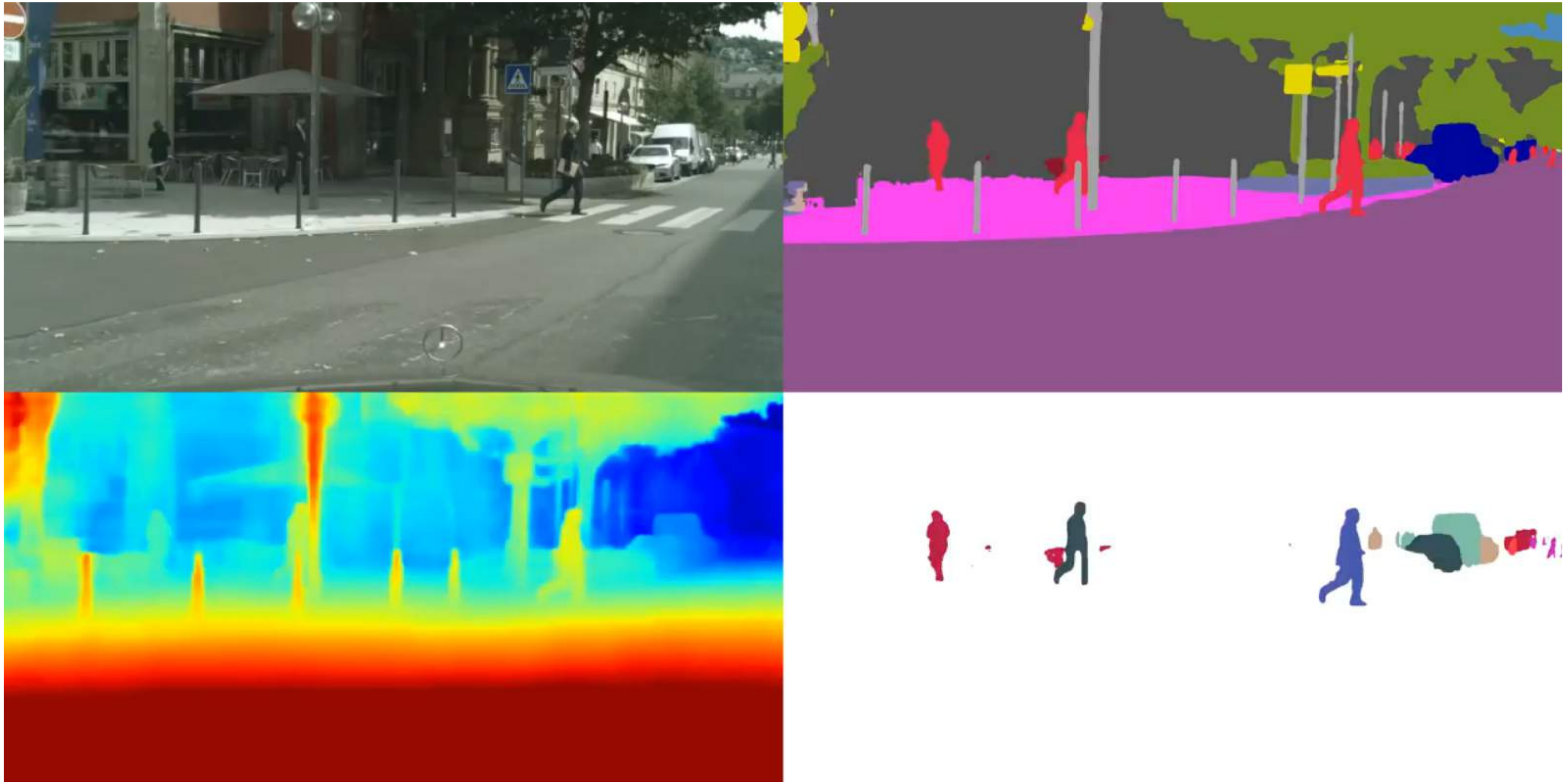
Predicted Depth



Model Uncertainty

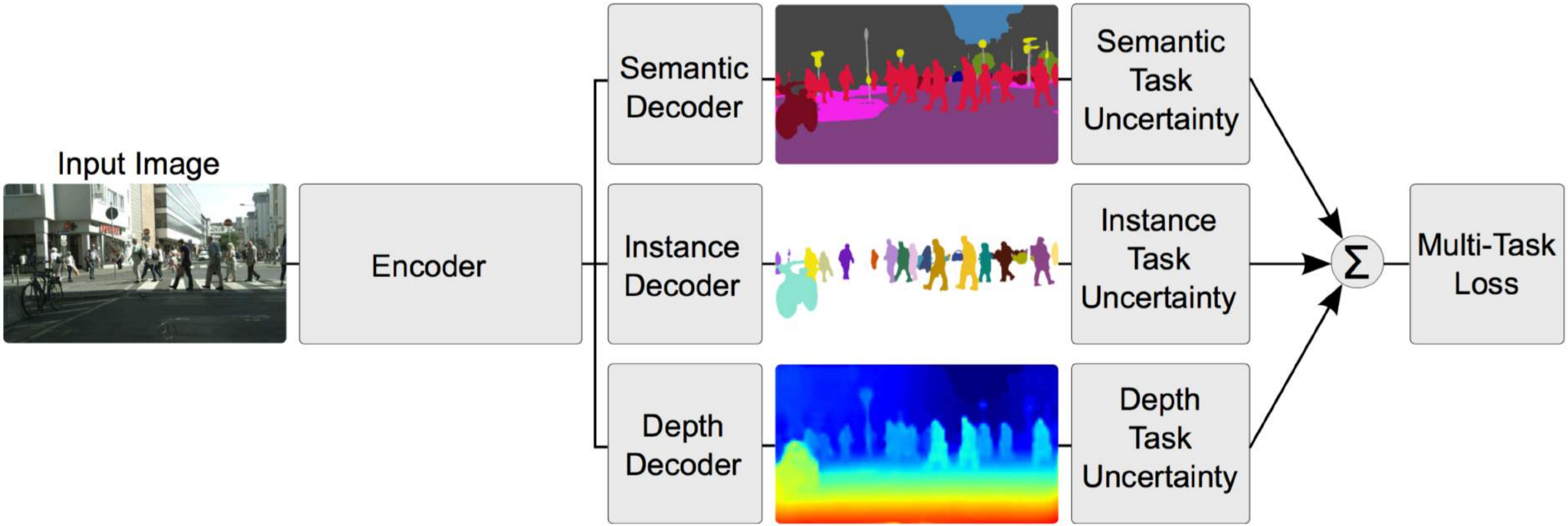
Kendall, Gal, *NIPS*, 2017.

Multi-Task Learning Using Uncertainty



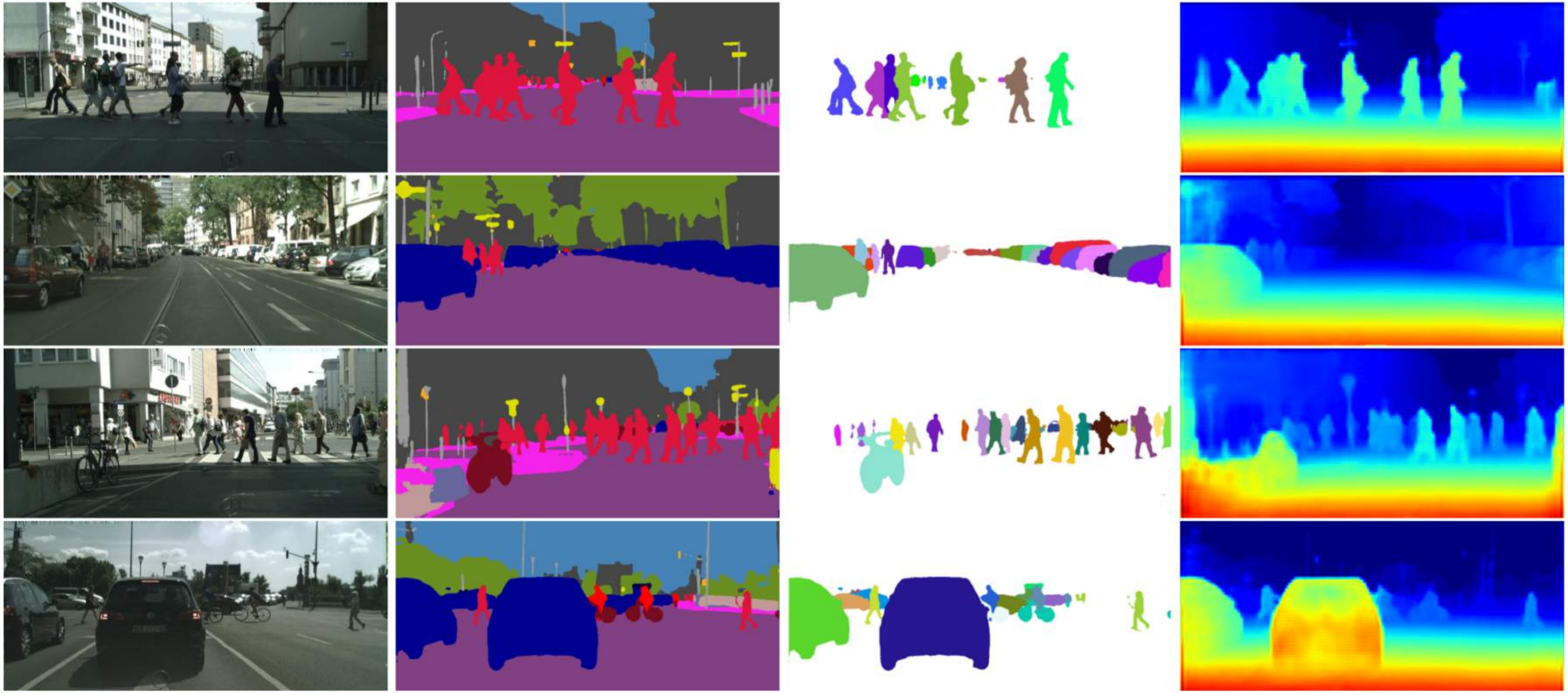
Kendall, et al., *CVPR*, 2018.

Multi-Task Learning Using Uncertainty



Kendall, et al., *CVPR*, 2018.

Multi-Task Learning Using Uncertainty

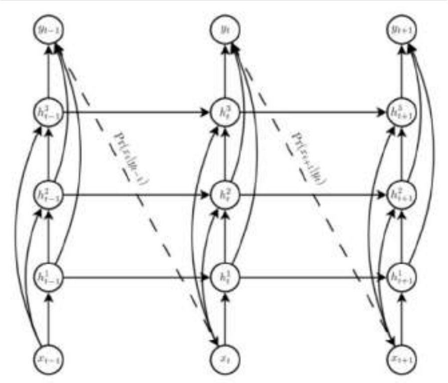


Kendall, et al., *CVPR*, 2018.

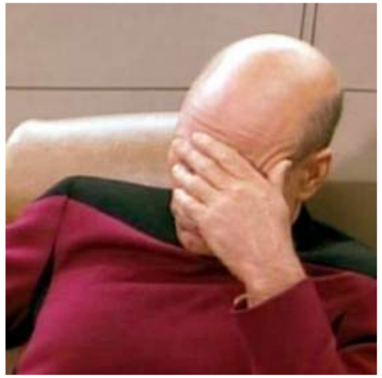
New Frontiers II:
Learning to Learn

Motivation: Learning to Learn

Standard deep neural networks are optimized for **a single task**



Complexity of models increases

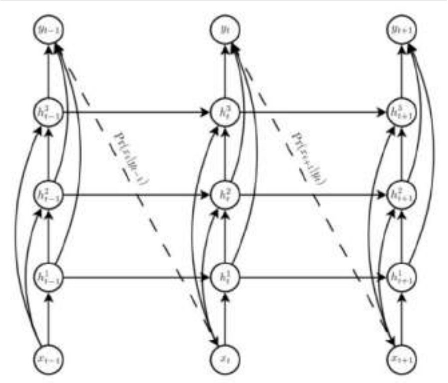


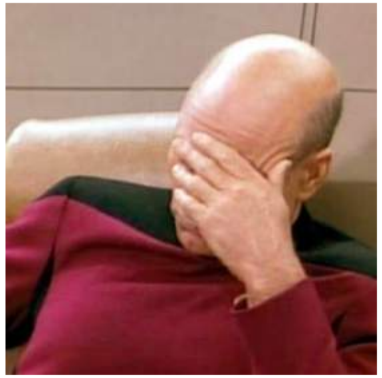
Greater need for specialized engineers

Often require **expert knowledge** to build an architecture for a given task

Motivation: Learning to Learn

Standard deep neural networks are optimized for **a single task**





Complexity of models increases

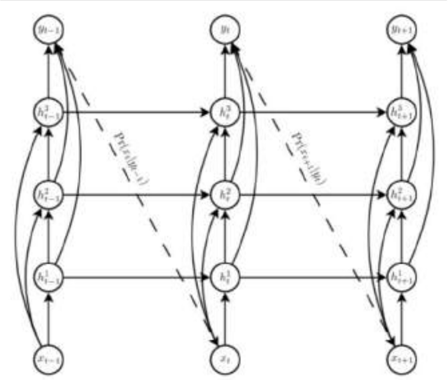
Greater need for specialized engineers

Often require **expert knowledge** to build an architecture for a given task

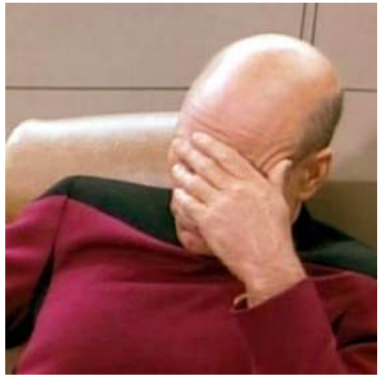
Build a learning algorithm that **learns which model** to use to solve a given problem

Motivation: Learning to Learn

Standard deep neural networks are optimized for **a single task**



Complexity of models increases



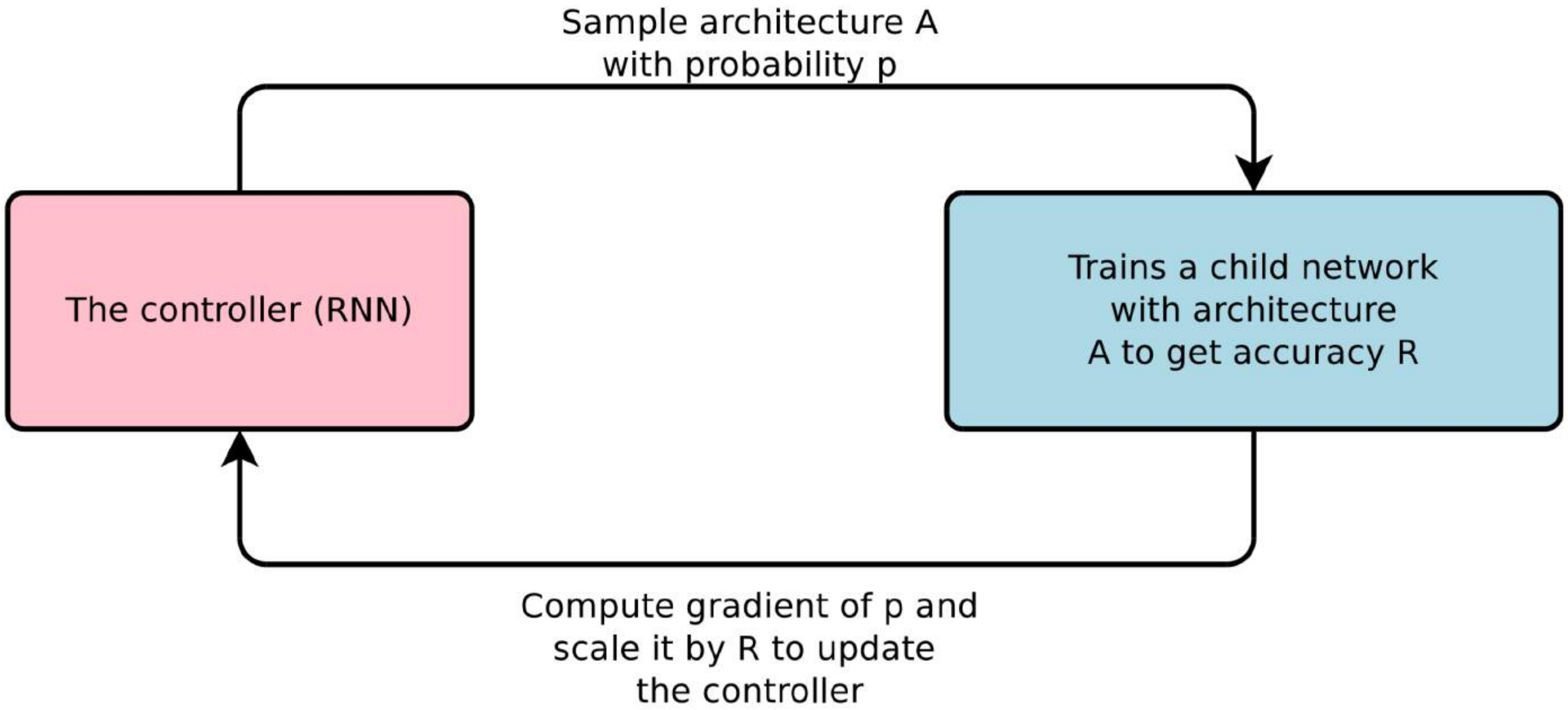
Greater need for specialized engineers

Often require **expert knowledge** to build an architecture for a given task

Build a learning algorithm that **learns which model** to use to solve a given problem

AutoML

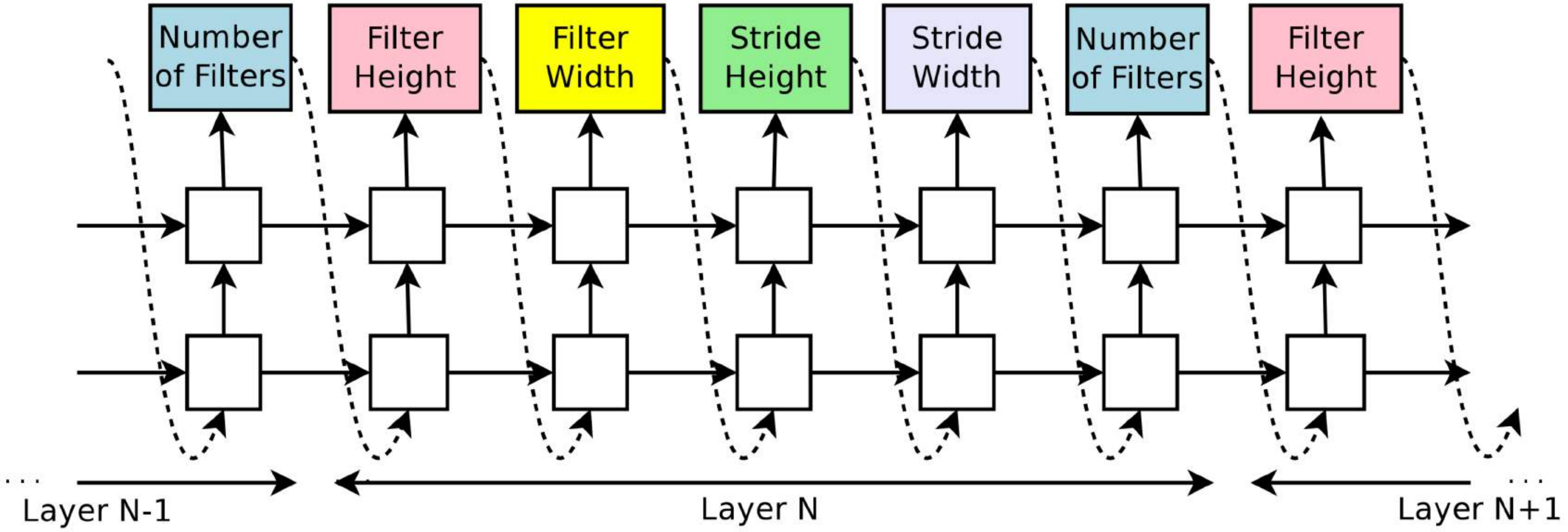
AutoML: Learning to Learn



Zoph and Le, *ICLR* 2017.

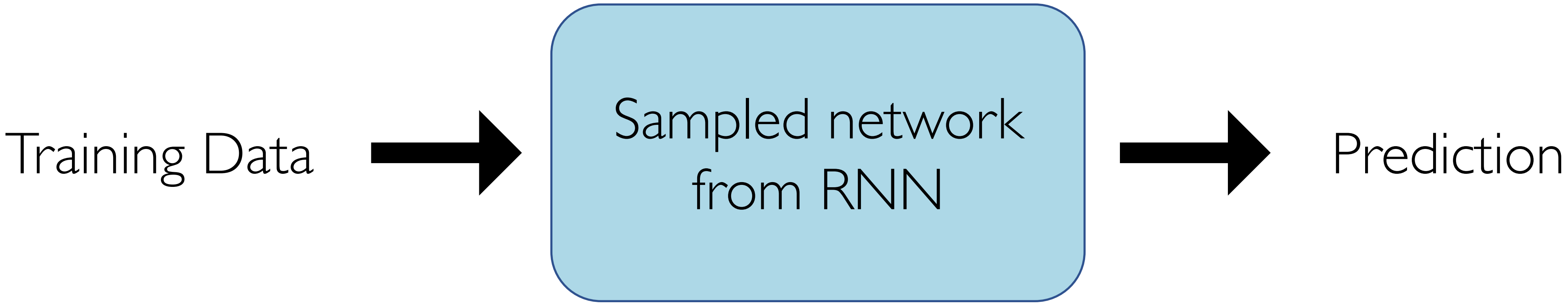
AutoML: Model Controller

At each step, the model samples a brand new network



Zoph and Le, *ICLR* 2017.

AutoML: The Child Network



Compute final accuracy on this dataset.
Update RNN controller based on the accuracy of the child network after training.

Zoph and Le, *ICLR* 2017.

AutoML on the Cloud

AutoML Vision BETA

Start with as little as a few dozen photographic samples, and Cloud AutoML will do the rest.

AutoML Natural Language BETA

Automatically predict text categories through either single or multi-label classification.

AutoML Translation BETA

Upload translated language pairs to train your own custom model.

Google Cloud.

AutoML Spawns a Powerful Idea

- Design an AI algorithm that can build new models capable of solving a task
- Reduces the need for experienced engineers to design the networks
- Makes deep learning more accessible to the public

Connection to
Artificial General Intelligence:
the ability to intelligently reason about how we learn

Questions?