Deep Learning
Limitations and New Frontiers

Final Class Project

Option 1:

Choose one of the topic dealt with in the course,
study deeply the state of the art, compare solutions in the literature, possibly executing comparative tests

Option 2:

Present a novel deep learning
research idea or application ideally concerning your research field

The Rise of Deep Learning

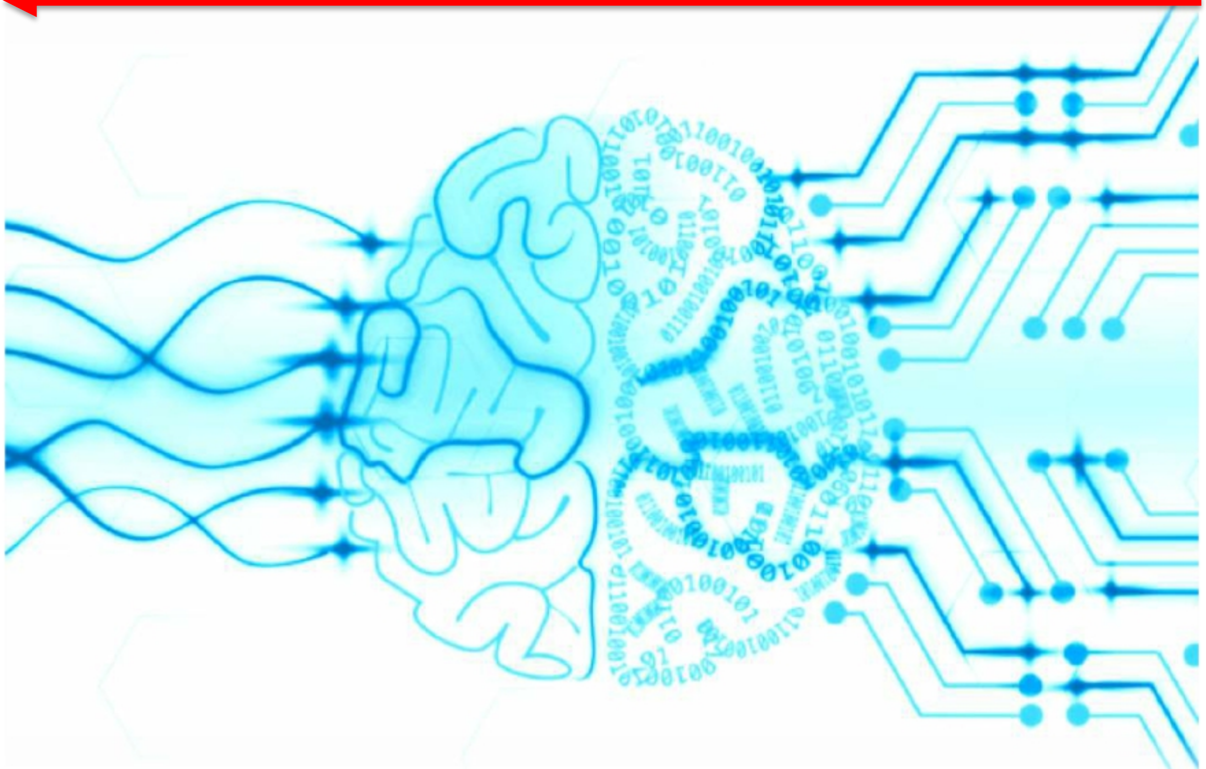
So far..

Generative Models

Data

- Signals
- Images
- Sensors

...



Decision

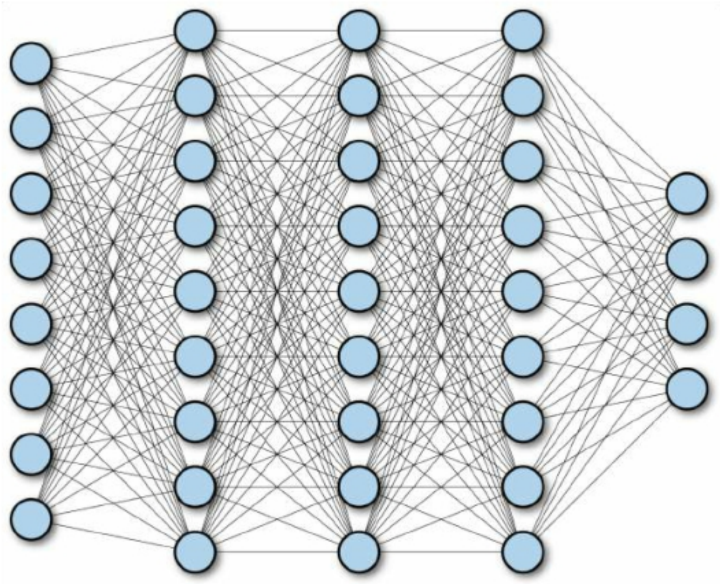
- Prediction
- Detection
- Action

...

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.

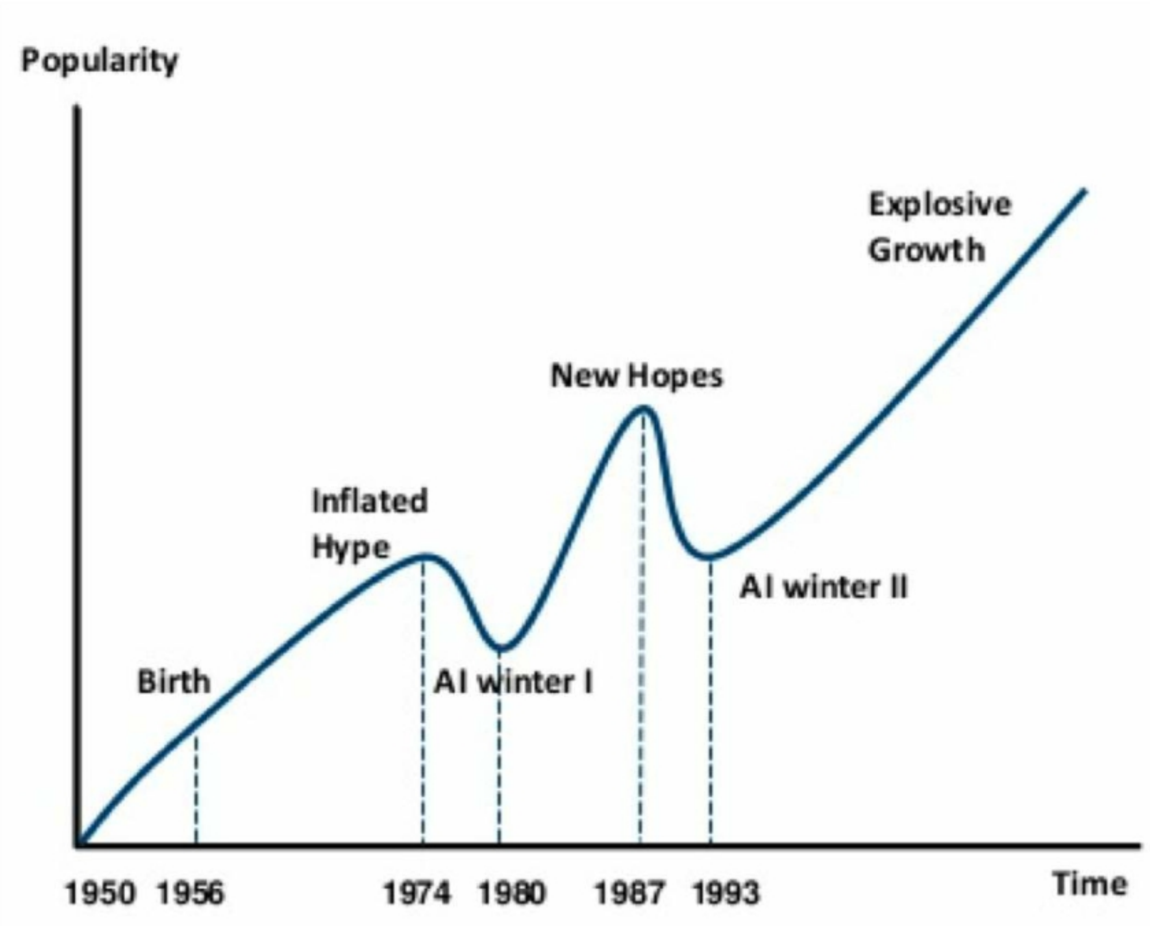
Caveats:

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

~~banana~~

dog

~~dog~~

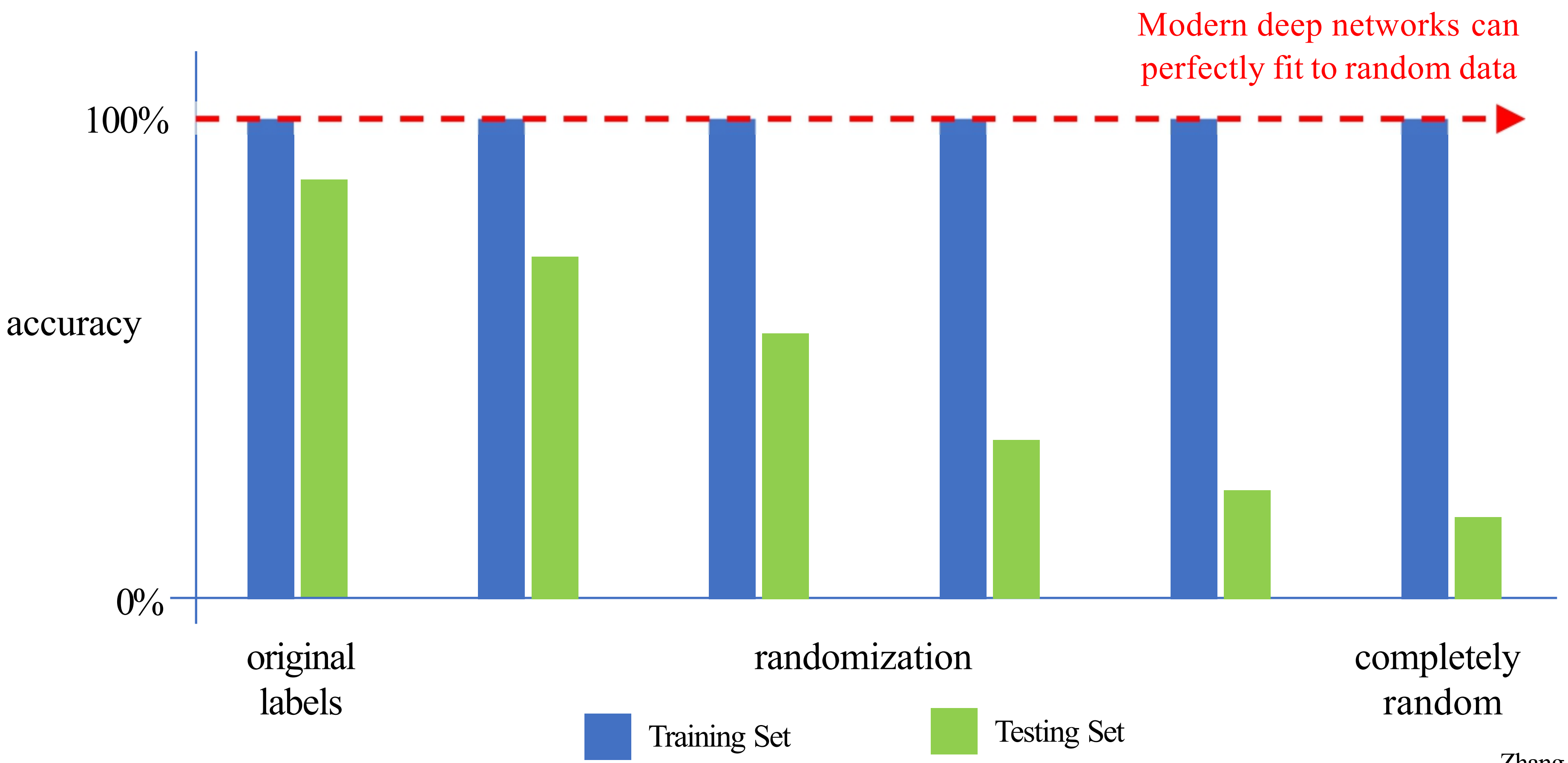
tree

~~tree~~

dog

Zhang et al. *ICLR*. (2017)

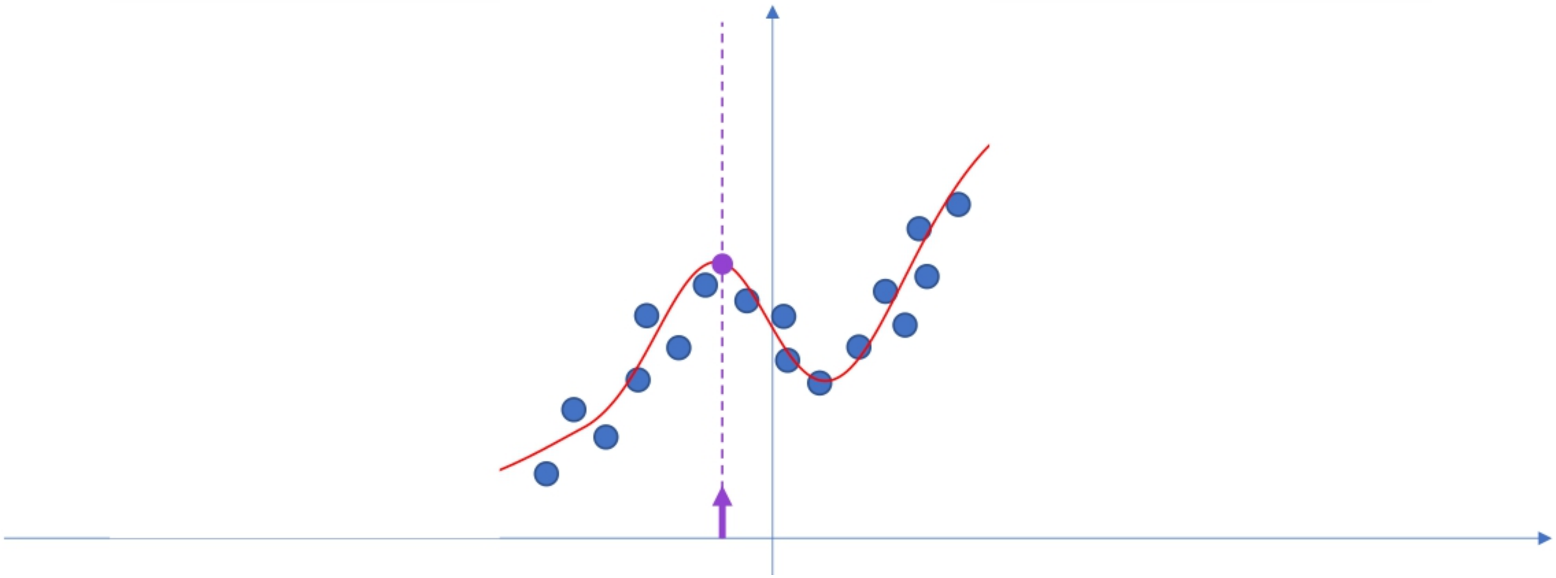
Capacity of Deep Neural Networks



Zhang et al. *ICLR*. (2017)

Neural Networks as Function Approximators

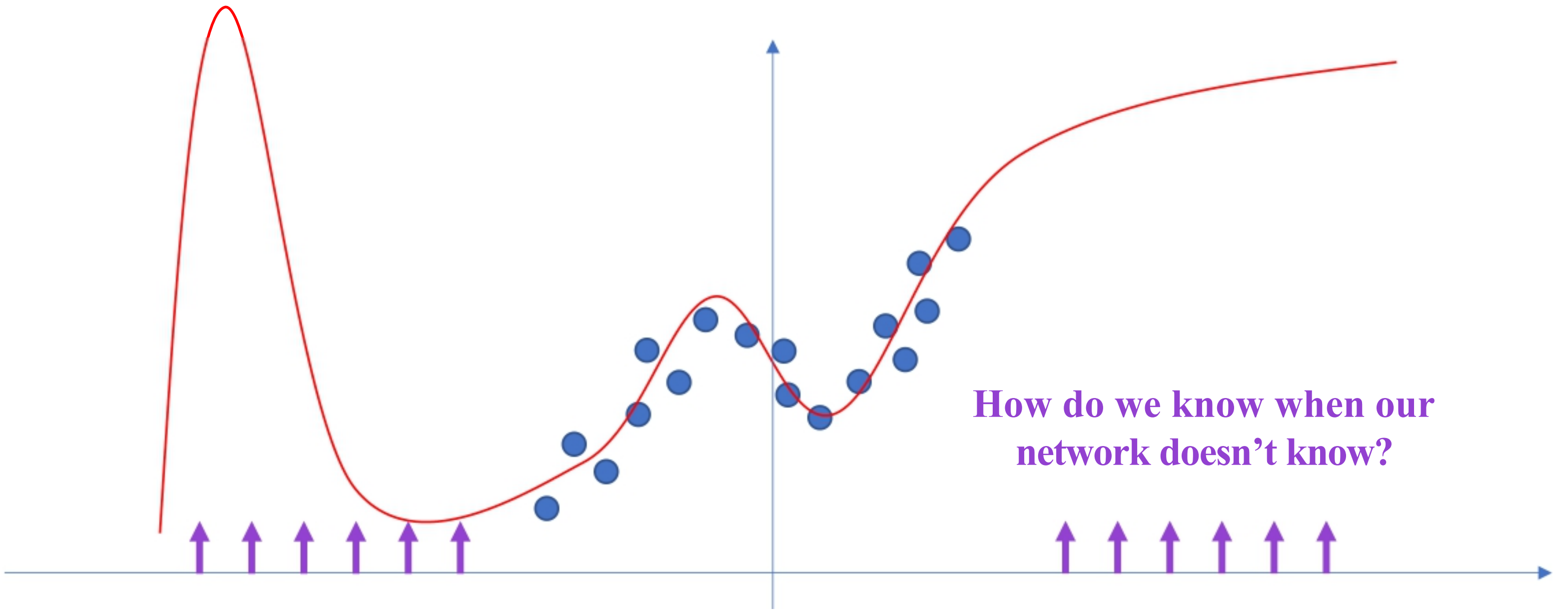
Neural networks are **excellent** function approximators



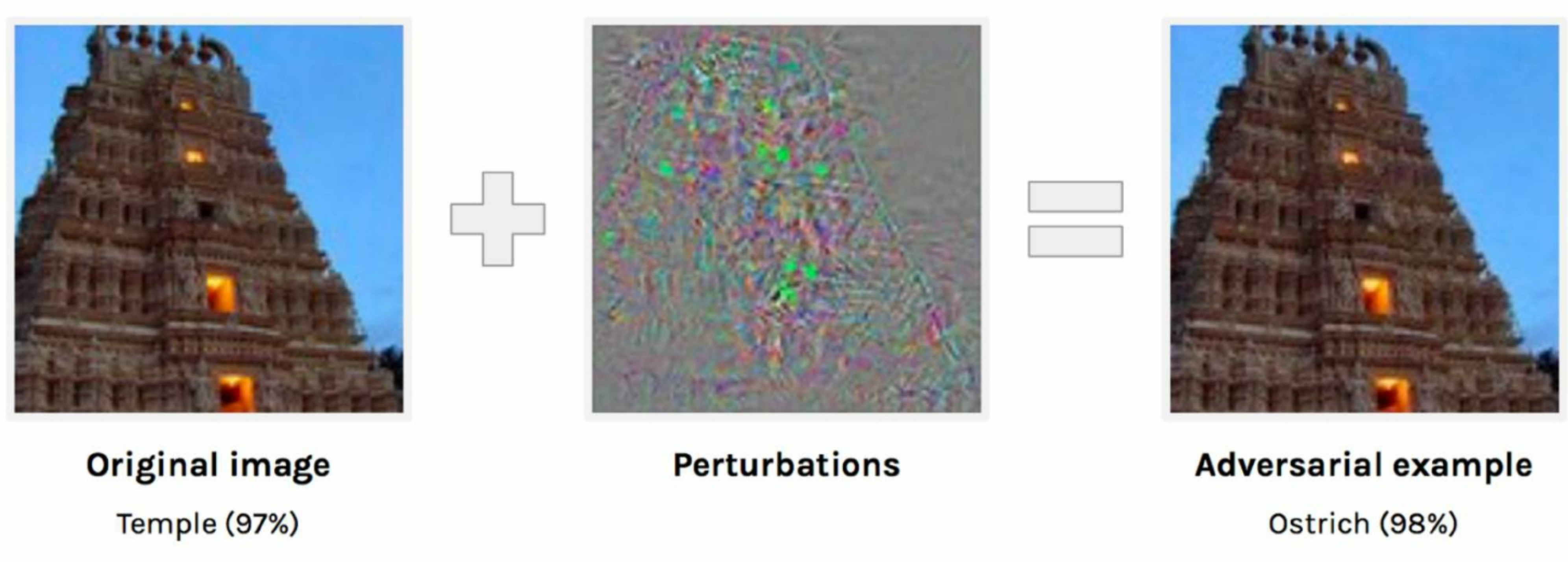
Neural Networks as Function Approximators

Neural networks are **excellent** function approximators

…when they have training data



Adversarial Attacks on Neural Networks



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

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

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

Fix your weights $\boldsymbol{\theta}$, and true label $\boldsymbol{y}$

"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

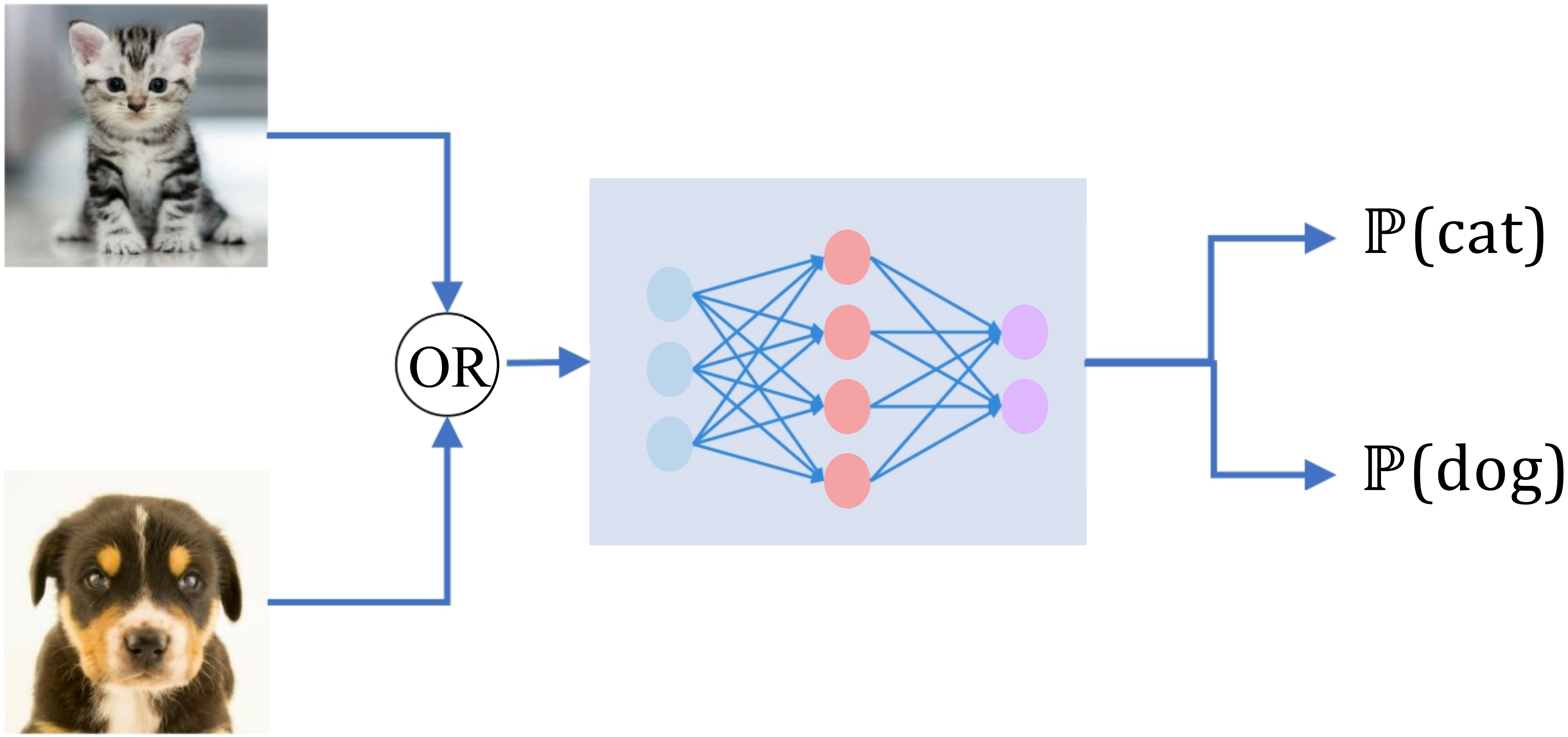
Neural Network Limitations…

- 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

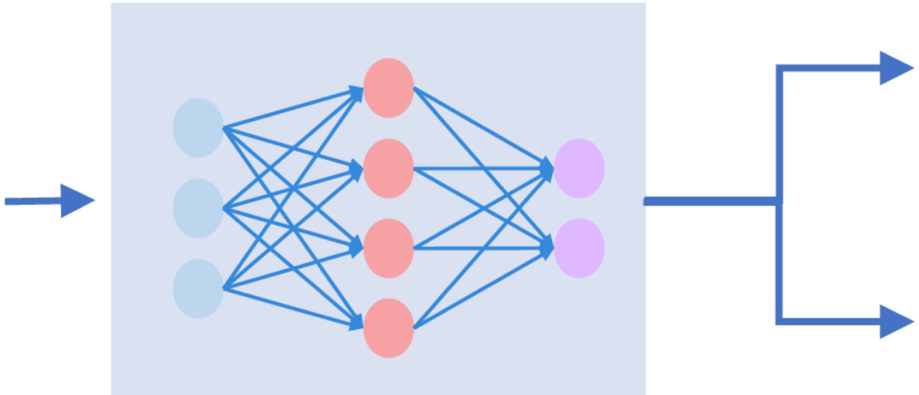
New Frontiers 1:

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

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

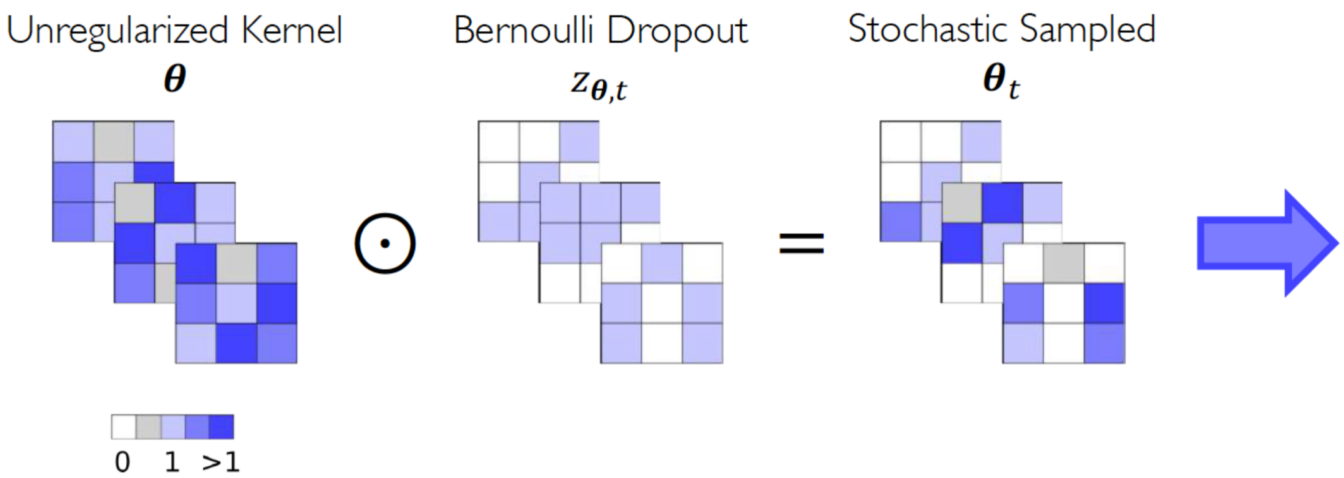
Bayesian neural networks aim to learn a posterior over weights, $\mathbb{P}(\boldsymbol{\theta}|\boldsymbol{X}, \boldsymbol{Y})$:

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 $z_{w,t} \sim Bernoulli(p) \quad \forall\, w \in \boldsymbol{\theta}$



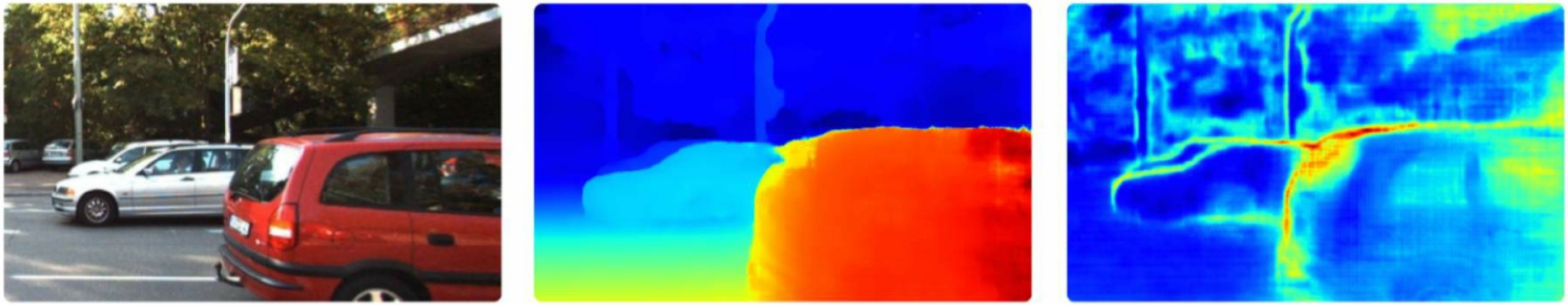
$$\mathbb{E}(\widehat{\boldsymbol{Y}}|\boldsymbol{X}) = \frac{1}{T}\sum_{t=1}^{T} f(\boldsymbol{X}|\boldsymbol{\theta}_t)$$

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

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

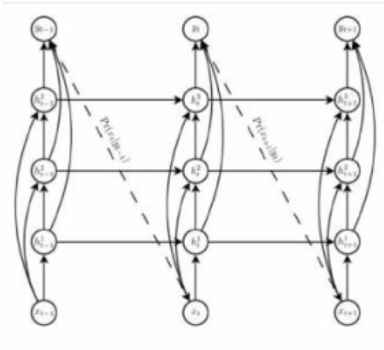
Kendall, Gal, *NIPS,* 2017.

New Frontiers II:

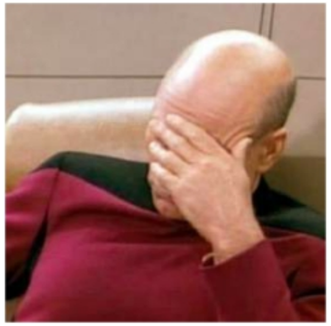
Learning to Learn

Motivation: Learning to Learn

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



Complexity of models increases



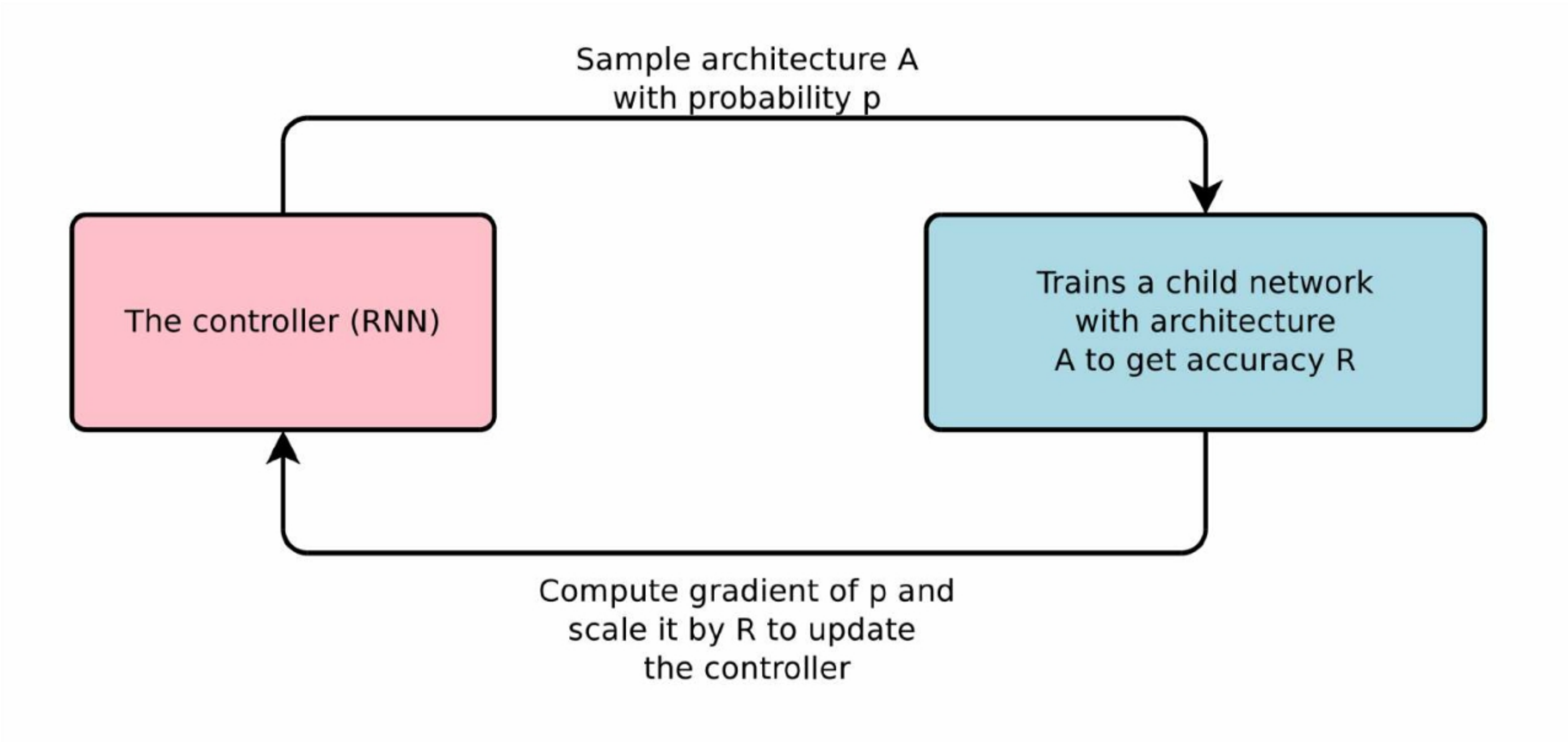
Greater need for specialized engineers

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

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

AutoML

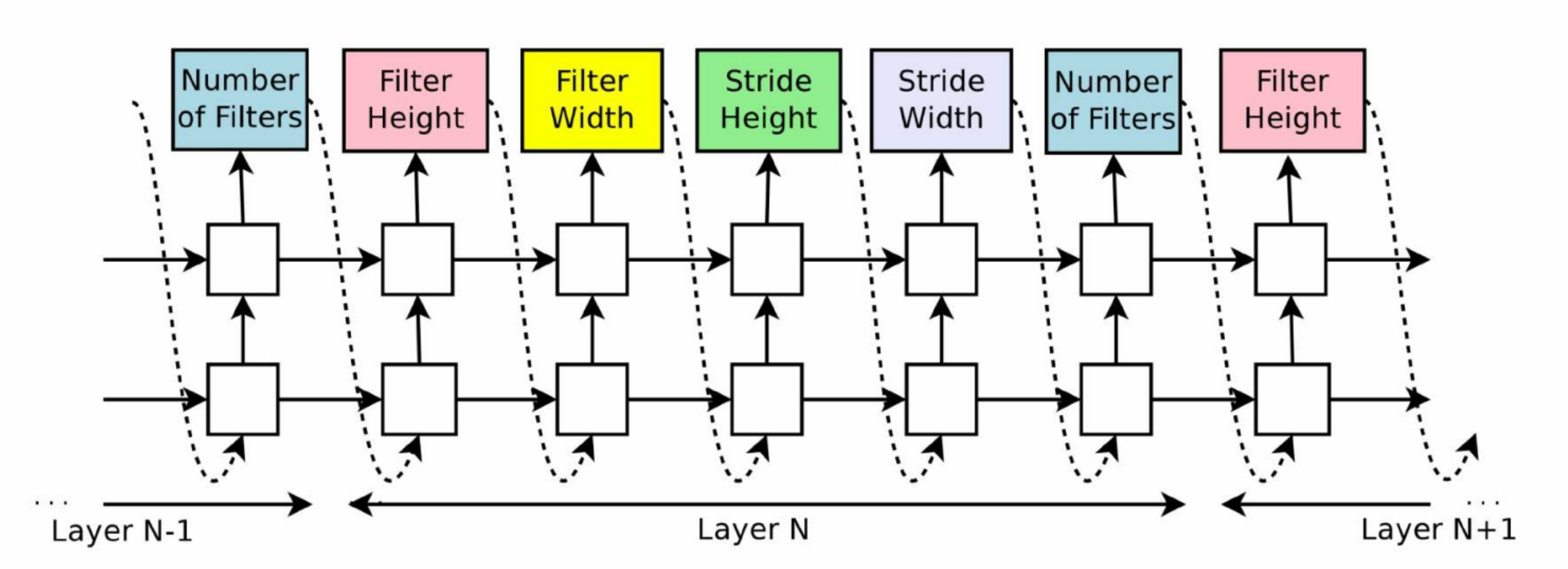
AutoML: Learning to Learn



Zoph and Le, *ICLR* 2017.

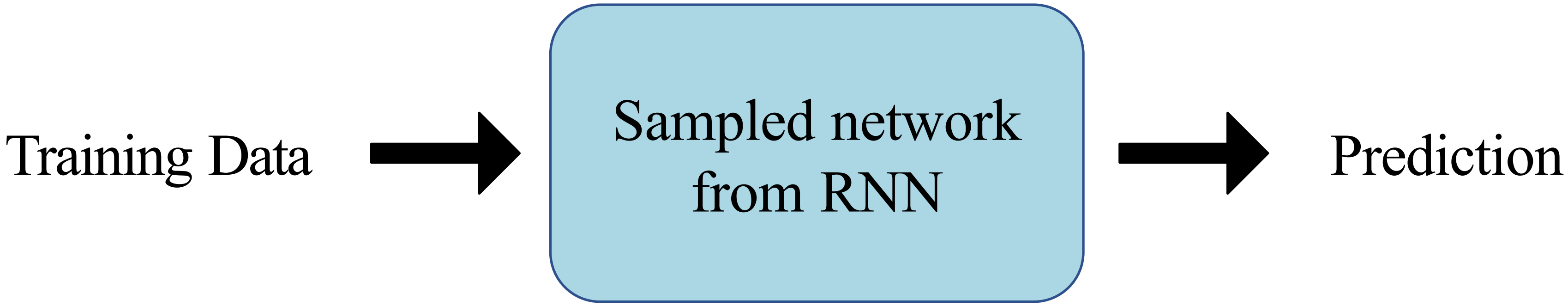
AutoML: Model Controller

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.