

# On the Expressive Power of Deep Neural Networks

Maithra Raj<sup>1,2</sup> Ben Poole<sup>3</sup> Jon Kleinberg<sup>1</sup> Surya Ganguli<sup>3</sup>  
Jascha Sohl Dickstein<sup>2</sup>

<sup>1</sup>Cornell University

<sup>2</sup>Google Brain

<sup>3</sup>Stanford University

ICLR, 2017

Presenter: Ritambhara Singh

## 1 Introduction

- Motivation
- State-of-the-art
- Contributions

## 2 Measures of Expressivity

- Neuron Transitions and Activity Patterns
- Trajectory length

## 3 Insights

- Expressivity and Network Stability
- Trajectory Regularization

## 1 Introduction

- Motivation
- State-of-the-art
- Contributions

## 2 Measures of Expressivity

- Neuron Transitions and Activity Patterns
- Trajectory length

## 3 Insights

- Expressivity and Network Stability
- Trajectory Regularization

# Motivation

- Understanding of how and why neural networks achieve empirical success is lacking.

# Motivation

- Understanding of how and why neural networks achieve empirical success is lacking.
- *Neural Network Expressivity*: To characterize how structural properties of neural network affect functions it is able to compute.

# Motivation

- Understanding of how and why neural networks achieve empirical success is lacking.
- *Neural Network Expressivity*: To characterize how structural properties of neural network affect functions it is able to compute.
- Neural Network (NN) Architecture:  $A$  (certain depth, width, layer type)
- All parameters of the network:  $W$
- Input:  $x$
- Associated Function:  $F_A(x; W)$

# Motivation

- Understanding of how and why neural networks achieve empirical success is lacking.
- *Neural Network Expressivity*: To characterize how structural properties of neural network affect functions it is able to compute.
- Neural Network (NN) Architecture:  $A$  (certain depth, width, layer type)
- All parameters of the network:  $W$
- Input:  $x$
- Associated Function:  $F_A(x; W)$
- **Goal**: To understand how behavior of  $F_A(x; W)$  changes when  $A$  changes.

## 1 Introduction

- Motivation
- State-of-the-art
- Contributions

## 2 Measures of Expressivity

- Neuron Transitions and Activity Patterns
- Trajectory length

## 3 Insights

- Expressivity and Network Stability
- Trajectory Regularization



- Studying expressivity using highly theoretical approaches like, comparison to boolean circuits etc.
- **Drawback:** Results shown on shallow networks that are different from deep networks used today.
- Understanding benefits of depth for neural networks, showing separations between deep and shallow networks.
- **Drawback:** Results on very specific choice of weights (hand-coded) and focus on only lower bounds.

## 1 Introduction

- Motivation
- State-of-the-art
- Contributions

## 2 Measures of Expressivity

- Neuron Transitions and Activity Patterns
- Trajectory length

## 3 Insights

- Expressivity and Network Stability
- Trajectory Regularization

# Contributions

- Propose easily computable measures of NN expressivity.
- Study input transformation by the network by measuring *trajectory length*, find exponential depth dependence of these measures.
- Show that all weights are not equal and optimizing weights of lower layers matter more.
- Propose new method of *Trajectory Regularization*, which is as good as batch normalization but more computationally efficient.

## 1 Introduction

- Motivation
- State-of-the-art
- Contributions

## 2 Measures of Expressivity

- Neuron Transitions and Activity Patterns
- Trajectory length

## 3 Insights

- Expressivity and Network Stability
- Trajectory Regularization

## Definition

Given two points,  $x_0, x_1 \in R^m$ ,  $x(t)$  is a *trajectory* (between  $x_0$  and  $x_1$ ) if  $x(t)$  is a curve parameterized by a scalar  $t \in [0, 1]$ , with  $x(0) = x_0$  and  $x(1) = x_1$  .

## Definition

For fixed  $W$ , a neuron with piecewise linear region *transitions* between inputs  $x, x + \delta$  if its activation function switches linear regions between  $x$  and  $x + \delta$ .

# Activation Pattern

## Definition

*Activation pattern*,  $AP(F_A(x(t)); W)$ , is a string of form  $\{0, 1\}^N$  (for ReLUs) and  $\{-1, 0, 1\}^N$  (for hard tanh) of the network encoding the linear region of activation function of **every** neuron, for an input  $x$  and weights  $W$ .

# (Tight) Upper Bound for Number of Activation Patterns

## Theorem

Let  $A_{(n,k)}$  denote a fully connected network with  $n$  hidden layers of width  $k$ , and inputs in  $R^m$ . Then the number of activation patterns  $A(F_{A_{(n,k)}}(R^m; W))$  is upper bounded by  $O(k^{mn})$  for ReLU activation, and  $O((2k)^{mn})$  for hard tanh.



# Regions in Input Space

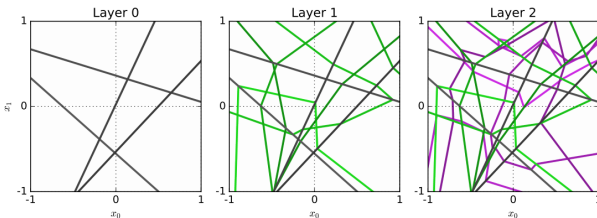
## Theorem

*Given the corresponding function of a neural network  $F_A(R^m; W)$  with ReLU or hard tanh activations, the input space is partitioned into convex polytopes, with  $F_A(R^m; W)$  corresponding to a different linear function on each region.*

# Regions in Input Space

## Theorem

Given the corresponding function of a neural network  $F_A(R^m; W)$  with ReLU or hard tanh activations, the input space is partitioned into convex polytopes, with  $F_A(R^m; W)$  corresponding to a different linear function on each region.



## 1 Introduction

- Motivation
- State-of-the-art
- Contributions

## 2 Measures of Expressivity

- Neuron Transitions and Activity Patterns
- Trajectory length

## 3 Insights

- Expressivity and Network Stability
- Trajectory Regularization

# Trajectory Length

## Definition

Given a trajectory,  $x(t)$ , its length  $l(x(t))$ , is the standard arc length:

$$l(x(t)) = \int_t \left\| \frac{dx(t)}{dt} \right\| dt \quad (1)$$

# Bound on Growth of Trajectory Length

- $A_{(n,k)}$  is fully connected network with  $n$  hidden layers of width  $k$  each.
- Initialize weights  $\sim \mathcal{N}(0, \sigma_w^2/k)$  and biases  $\sim \mathcal{N}(0, \sigma_b^2)$ .

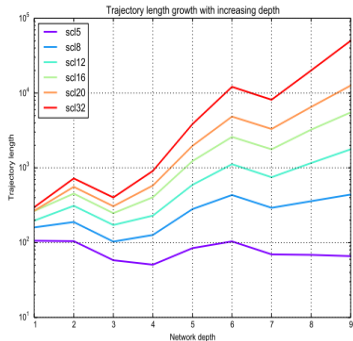
## Theorem

Let  $F_A(x', W)$  be a ReLU or hard tanh random neural network and  $x(t)$  a one dimensional trajectory with  $x(t + \delta)$  having a non-trivial perpendicular component to  $x(t)$  for all  $t$  and  $\delta$  (i.e, not a line). Then defining  $z^{(d)}(x(t)) = z^{(d)}(t)$  to be the image of the trajectory in layer  $d$  of the network:

$$E[I(z^{(d)}(t))] \geq O\left(\frac{\sigma_w \sqrt{k}}{\sqrt{k+1}}\right)^d I(x(t))[\text{ReLU}] \quad (2)$$

$$E[I(z^{(d)}(t))] \geq O\left(\frac{\sigma_w \sqrt{k}}{\sigma_w^2 + \sigma_b^2 + k\sqrt{\sigma_w^2 + \sigma_b^2}}\right)^d I(x(t))[\text{hardtanh}] \quad (3)$$

# Bound on Growth of Trajectory Length



# Transitions proportional to trajectory length

## Theorem

Let  $F_{A(n,k)}$  be a hard tanh network with  $n$  hidden layers each of width  $k$ .  
And let

$$g(k, \sigma_w, \sigma_b, n) = O\left(\frac{\sqrt{k}}{\sqrt{1 + \frac{\sigma_w^2}{\sigma_b^2}}}\right)^n \quad (4)$$

Then  $T(F_{A(n,k)}(x(t); W)) = O(g(k, \sigma_w, \sigma_b, n))$  for  $W$  initialized with weight and bias scales  $\sigma_w, \sigma_b$ .

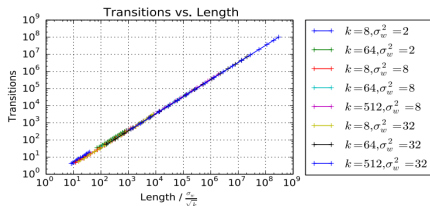
# Transitions proportional to trajectory length

## Theorem

Let  $F_{A(n,k)}$  be a hard tanh network with  $n$  hidden layers each of width  $k$ . And let

$$g(k, \sigma_w, \sigma_b, n) = O\left(\frac{\sqrt{k}}{\sqrt{1 + \frac{\sigma_w^2}{\sigma_b^2}}}\right)^n \quad (4)$$

Then  $T(F_{A(n,k)}(x(t); W)) = O(g(k, \sigma_w, \sigma_b, n))$  for  $W$  initialized with weight and bias scales  $\sigma_w, \sigma_b$ .





## 1 Introduction

- Motivation
- State-of-the-art
- Contributions

## 2 Measures of Expressivity

- Neuron Transitions and Activity Patterns
- Trajectory length

## 3 Insights

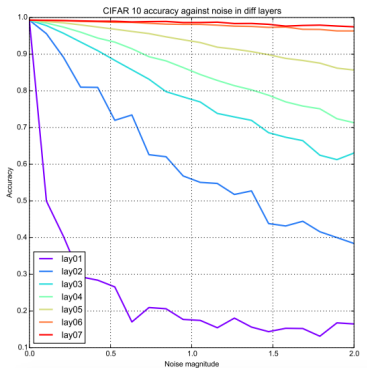
- Expressivity and Network Stability
- Trajectory Regularization

# Expressivity and Network Stability

- A perturbation at a layer grows exponentially in the remaining depth after that layer

# Expressivity and Network Stability

- A perturbation at a layer grows exponentially in the remaining depth after that layer



# Outline

## 1 Introduction

- Motivation
- State-of-the-art
- Contributions

## 2 Measures of Expressivity

- Neuron Transitions and Activity Patterns
- Trajectory length

## 3 Insights

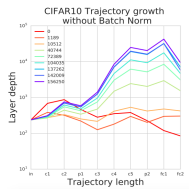
- Expressivity and Network Stability
- Trajectory Regularization

# Trajectory Regularization

- Initial growth of trajectory length enables greater functional expressivity, however, large trajectory growth in the learnt representation results in unstable representation.

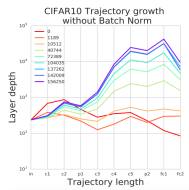
# Trajectory Regularization

- Initial growth of trajectory length enables greater functional expressivity, however, large trajectory growth in the learnt representation results in unstable representation.



# Trajectory Regularization

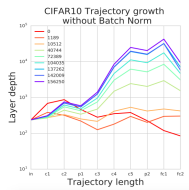
- Initial growth of trajectory length enables greater functional expressivity, however, large trajectory growth in the learnt representation results in unstable representation.



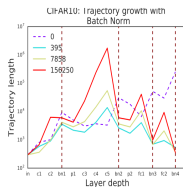
- Batch normalization layers reduce trajectory length, helping stability

# Trajectory Regularization

- Initial growth of trajectory length enables greater functional expressivity, however, large trajectory growth in the learnt representation results in unstable representation.



- Batch normalization layers reduce trajectory length, helping stability





# Summary

- Presented interrelated **measures of expressivity** of NN.
- Analysis of **trajectories** gives insight for performance of trained NNs.
- Developed new regularization method, **trajectory regularization**.
- Future work
  - Linking measures of expressivity to other properties of NN performance.
  - Natural connection between adversarial samples and trajectory length.