Gradients in Neural Networks: Interpretation, and Applications in Image Understanding

Ghassan AlRegib, PhD
Professor

Mohit Prabhushankar, PhD
Postdoctoral Fellow

Omni Lab for Intelligent Visual Engineering and Science (OLIVES)
School of Electrical and Computer Engineering
Georgia Institute of Technology
{alregib, mohit.p}@gatech.edu

Oct 08, 2023 – Kuala Lumpur, Malaysia

Gradients in Neural Networks: Interpretation, and Applications in Image Understanding

To cite this Tutorial:

Ghassan AlRegib, and Mohit Prabhushankar. Tutorial on ‘A Multifaceted View of Gradients in Neural Networks: Extraction, Interpretation, and Applications in Image Understanding’. IEEE International Conference on Image Processing (ICIP 2023), Kuala Lumpur, Malaysia, Oct 8, 2023.

Omni Lab for Intelligent Visual Engineering and Science (OLIVES)
School of Electrical and Computer Engineering
Georgia Institute of Technology
{alregib, mohit.p}@gatech.edu

Oct 08, 2023 – Kuala Lumpur, Malaysia

Tutorial Materials

Accessible Online

SCAN ME

https://alregib.ece.gatech.edu/ieee-icip-2023-tutorial/

{alregib, mohit.p}@gatech.edu

IEEE ICIP 2023 Tutorial

Title: A Multi-Faceted View of Gradients in Neural Networks: Extraction, Interpretation and Applications in Image Understanding

Type / Duration: Half-Day Tutorial (3h)

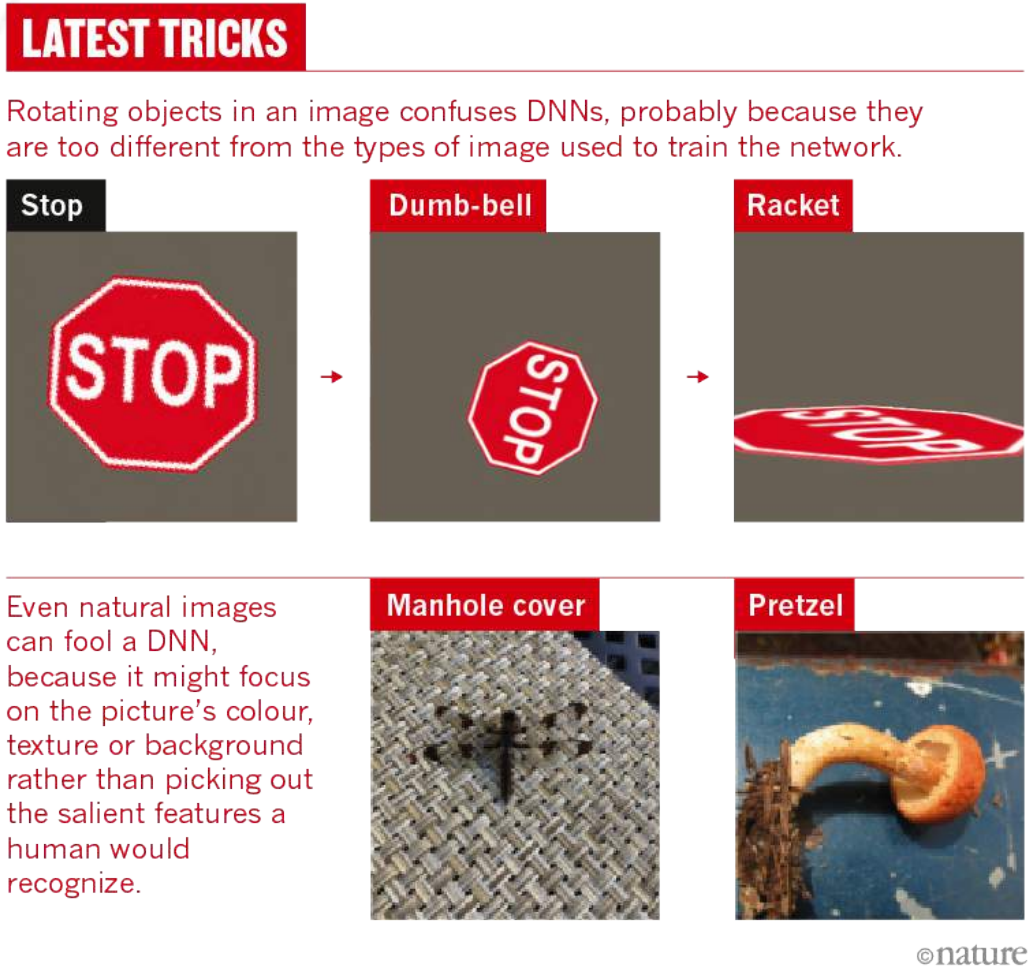
Deep Learning

Expectation vs Reality

People’s expectation of AI and Deep Learning

Deep Learning

Expectation vs Reality



Deep Learning

Expectation vs Reality

"The best-laid plans of sensors and networks often go awry"

- Engineers, probably

Requirements: Deep Learning-enabled systems must predict correctly on novel data

Novel data sources:

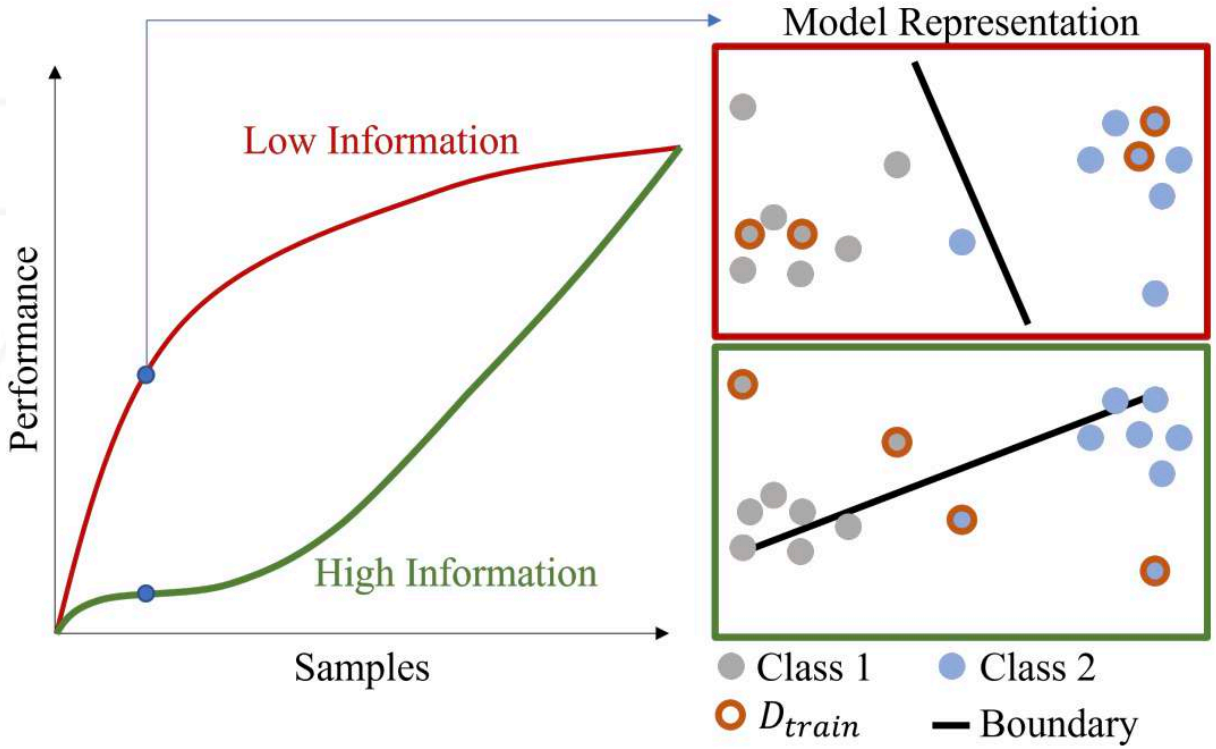
- Test distributions
- Anomalous data
- Out-Of-Distribution data
- Adversarial data
- Corrupted data
- Noisy data
- New classes
- …

Temel, Dogancan, et al. "Cure-tsd: Challenging unreal and real environments for traffic sign detection." *IEEE Transactions on Intelligent Transportation Systems* (2017).

Deep Learning at Training

Overcoming Challenges at Training: Part 1

The most novel/aberrant samples should not be used in early training



- The first instance of training must occur with less informative samples
- Ex: For autonomous vehicles, less informative means
  - Highway scenarios
  - Parking
  - No accidents
  - No aberrant events

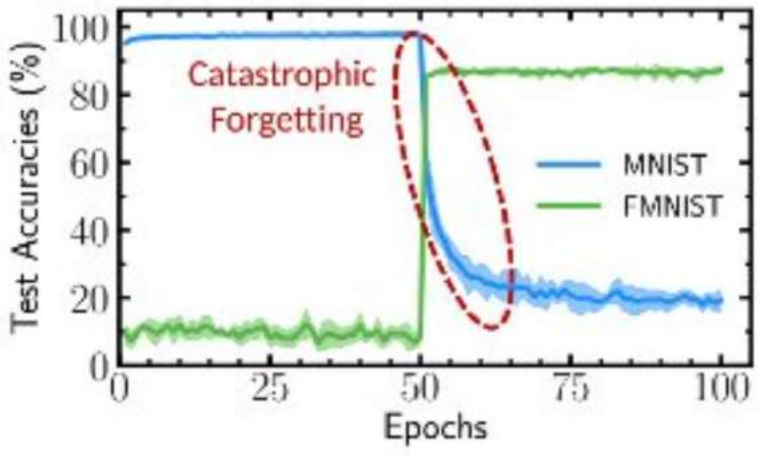
Novel samples = Most Informative

Benkert, R., Prabushankar, M., AlRegib, G., Pacharmi, A., & Corona, E. (2023). Gaussian Switch Sampling: A Second Order Approach to Active Learning. *IEEE Transactions on Artificial Intelligence*.

Deep Learning at Training

Overcoming Challenges at Training: Part 2

Subsequent training must not focus only on novel data



Catastrophic Forgetting

- The model performs well on the new scenarios, while forgetting the old scenarios
- A number of techniques exist to overcome this trend
- However, they affect the overall performance in large-scale settings
- It is not always clear **if and when** to incorporate novel scenarios in training

Where to handle novel data?

Laborieux, Axel, et al. "Synaptic metaplasticity in binarized neural networks." *Nature communications* 12.1 (2021): 2549.

We handle novel data at Inference!!

Novel data sources:

- Test distributions
- Anomalous data
- Out-Of-Distribution data
- Adversarial data
- Corrupted data
- Noisy data
- New classes
- …

Model Train

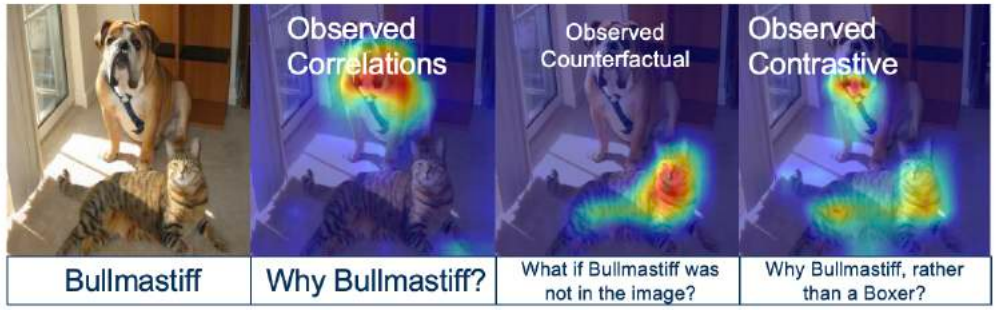
At Inference

Objective

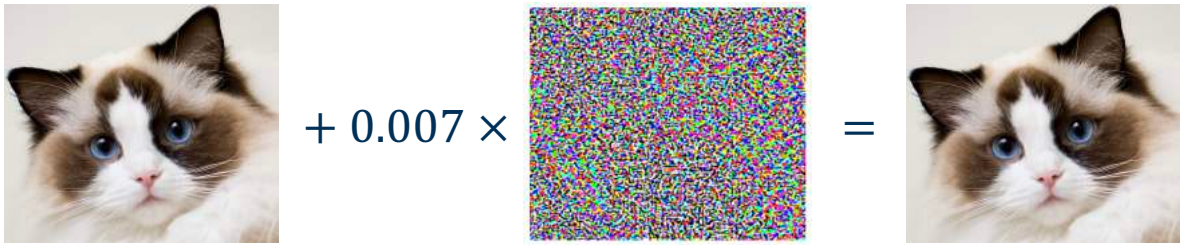
Objective of the Tutorial

To present methodologies to handle novel data at inference using gradients of neural networks

At the end of the tutorial you will be able to

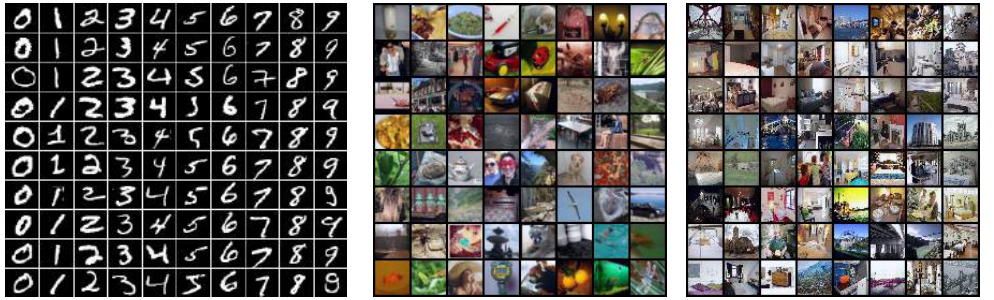


Obtain fine-grained explanations



Engineer (and detect) adversarial examples

Construct XAI techniques for Image Quality Assessment



Training Dataset Testing Dataset

Perform Out-Of-Distribution and Anomaly Detection

Objective

Objective of the Tutorial

To present methodologies to handle novel data at inference using gradients of neural networks

- Part 1: Gradients in Neural Networks
  - Neural network basics, gradient descent, and properties of gradients
- Part 2: Gradients as Information
  - Visual explanations, robust recognition
- Part 3: Gradients as Uncertainty
  - Anomaly, Out-Of-Distribution, corruption, and adversarial detection
- Part 4: Gradients as Expectancy-Mismatch
  - Image Quality Assessment, human visual saliency
- Part 5: Conclusion and Future Directions

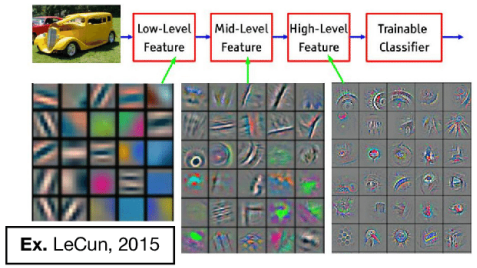
Interpretation, and Applications of Gradients

Part I: Gradients in Neural Networks

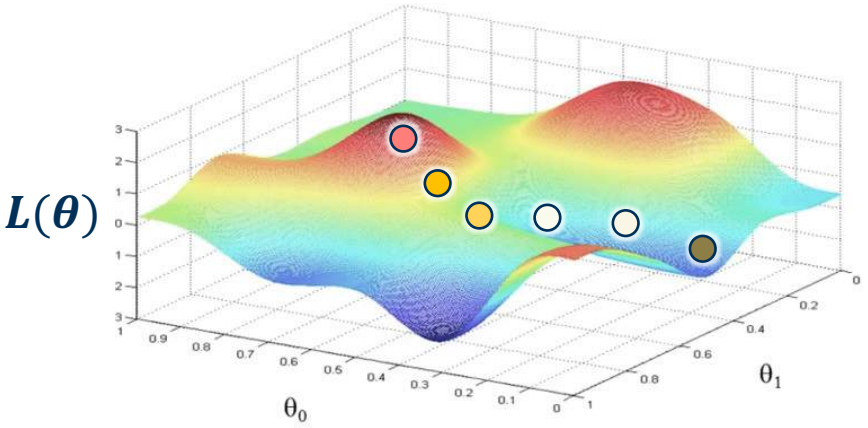
Objectives

Objectives in Part 1

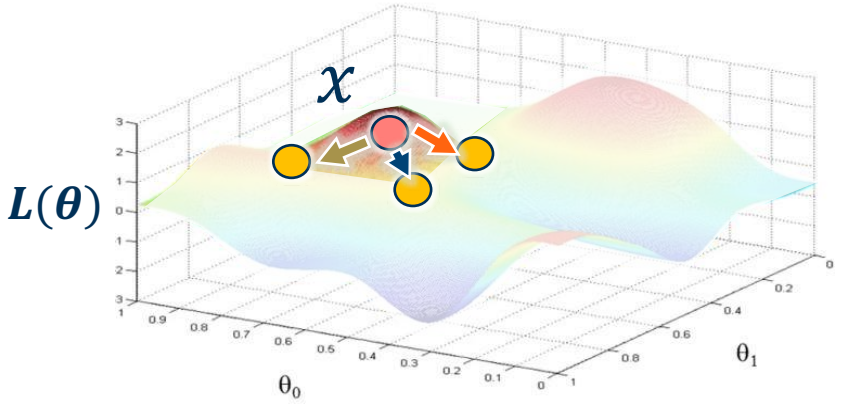
At the end of Part 1 you will be able to



1. Describe the basics of neural networks



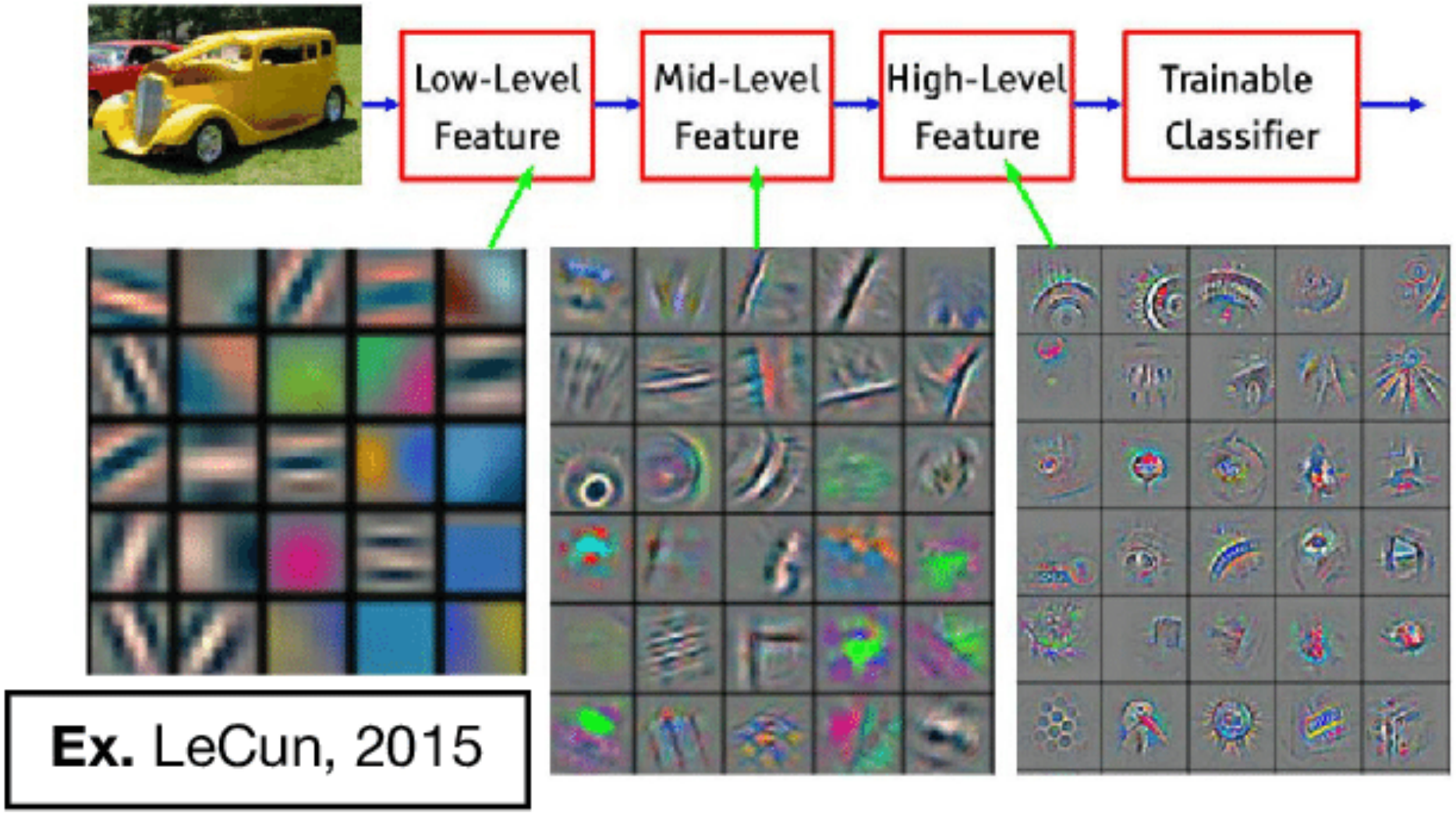
2. Discuss the role of gradients in optimization



3. Discuss relevant properties of gradients

Deep Learning

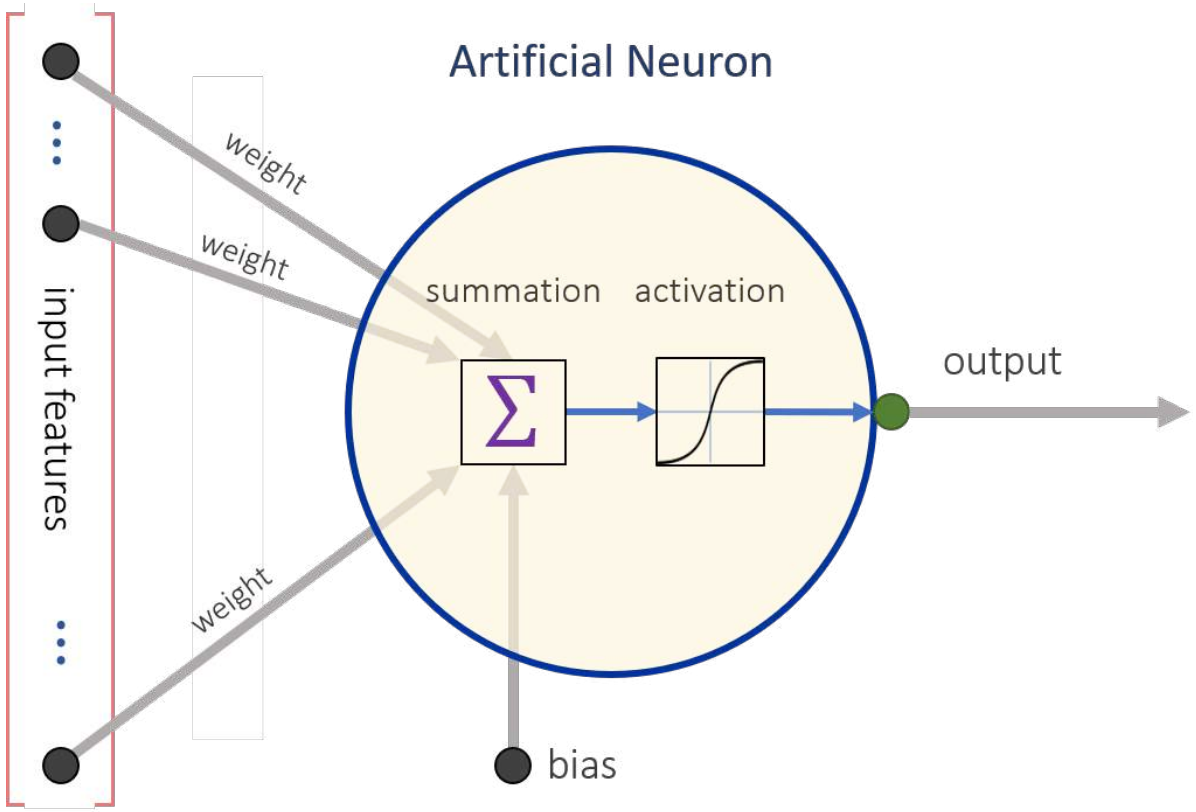
Overview



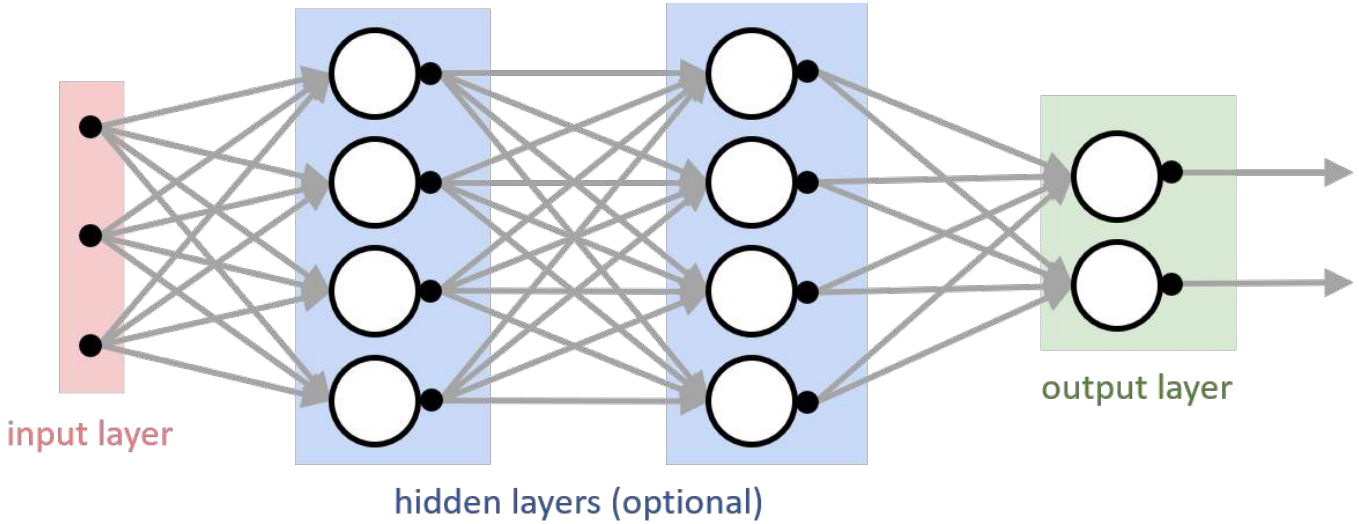
The underlying computation unit is the Neuron

Artificial neurons consist of:

- A single output
- Multiple inputs
- Input weights
- A bias input
- An activation function



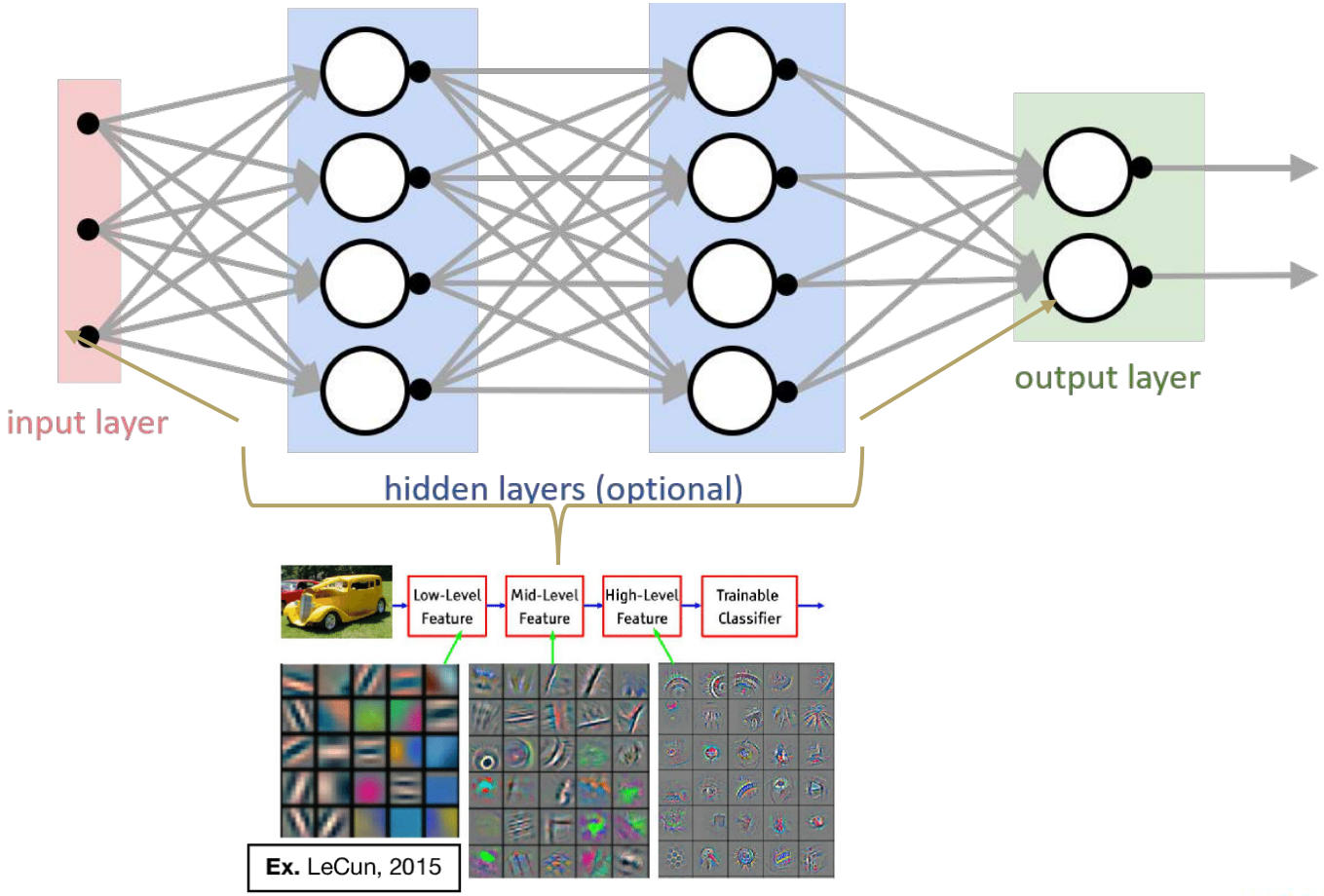
Neurons are stacked and densely connected to construct ANNs



Typically, a neuron is part of a network organized in layers:

- An input layer (Layer 0)
- An output layer (Layer K)
- Zero or more hidden (middle) layers (Layers $1 \ldots K-1$)

Stationary property of images allow for a small number of convolution kernels

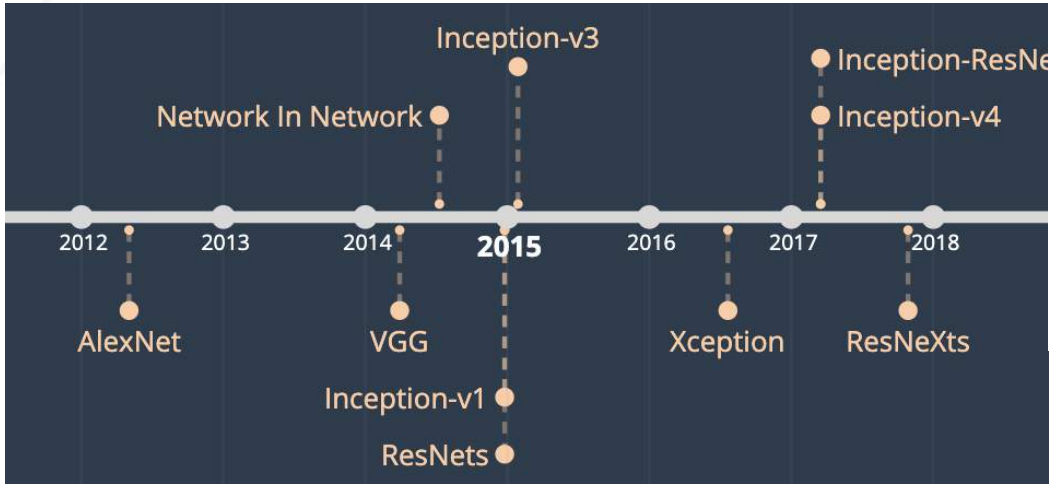


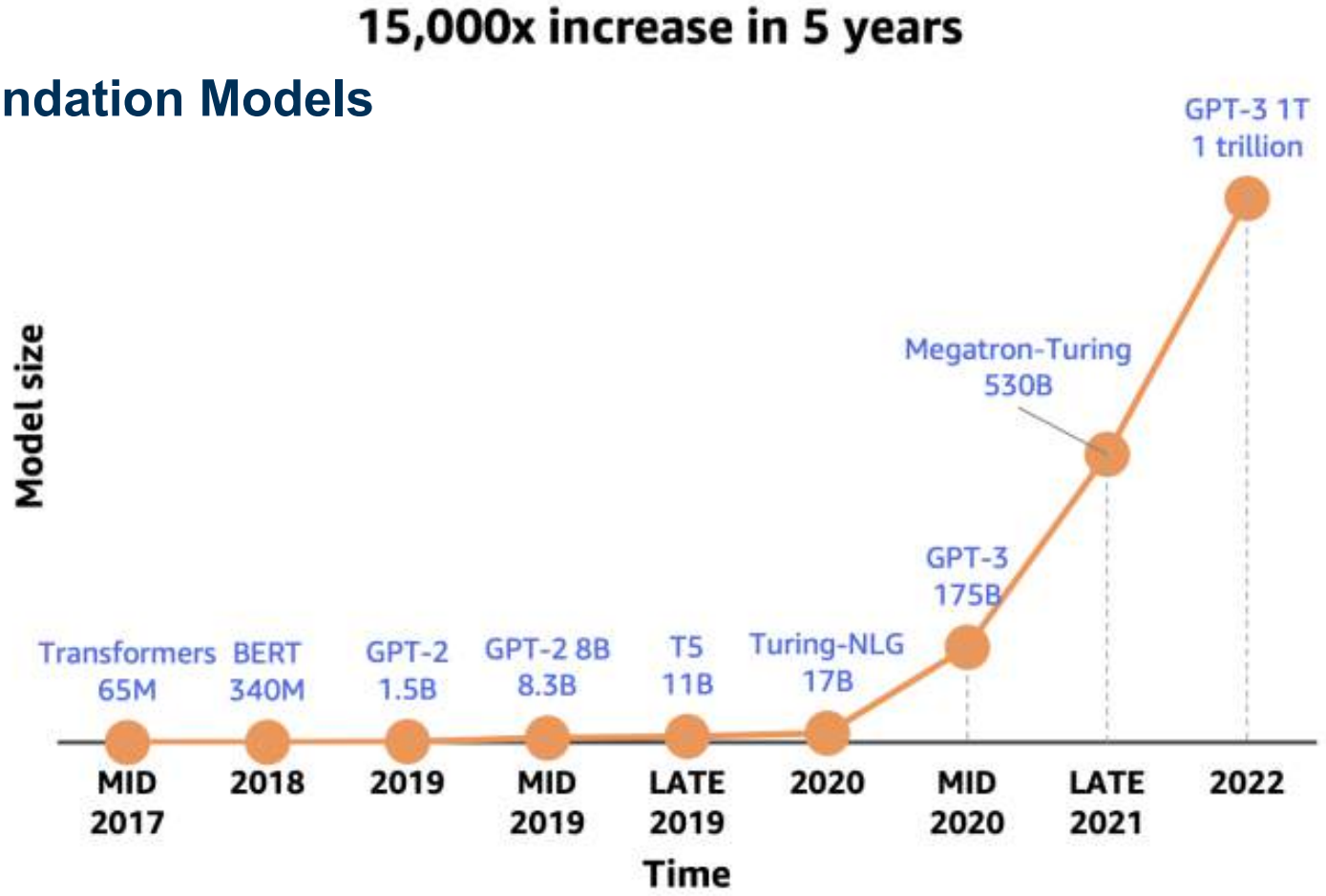
Deep Deep Deep … Deep Deep Learning

Recent Advancements

Transformers, Large Language and Foundation Models

The number of parameters in models has increased exponentially





Training Neural Networks

Stochastically and via Gradient updates

Iteratively reduce a loss function $L(\theta)$ to find the optimal parameters θ

- θ is a combination of weights and biases
- Compute the gradients of a loss function iteratively and update the weights according to the update rule:

$$\theta(t+1) = \theta(t) - \alpha \frac{\partial L(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$$

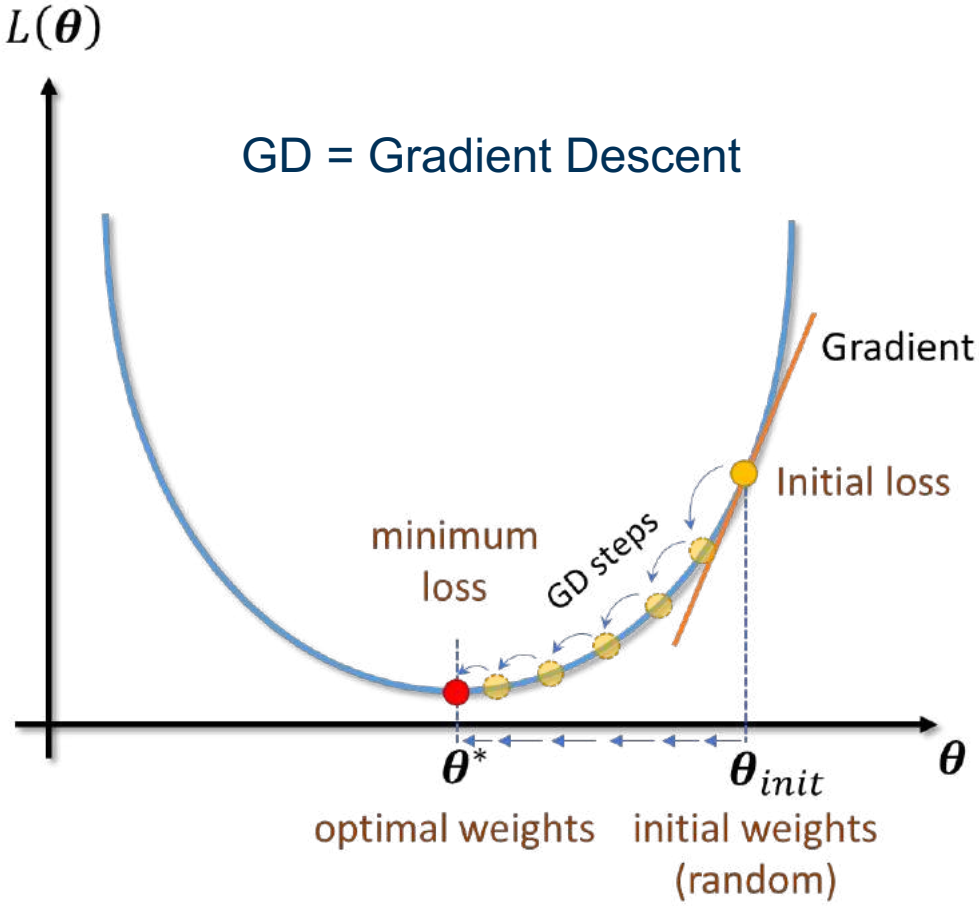
$\theta =$ Weights, biases

$t =$ Iteration step

$\alpha =$ Step Length

$L(\theta) =$ Loss function between prediction and ground truth

$\frac{\partial L(\theta)}{\partial \theta} =$ Gradient w.r.t weights and biases

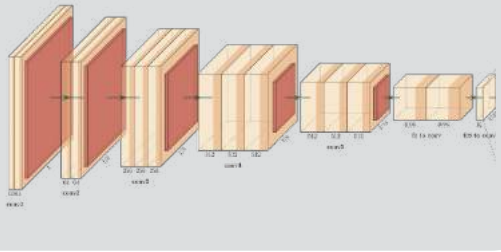


Training Neural Networks

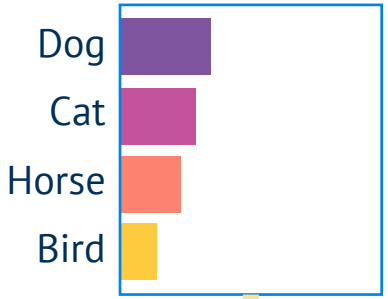
Gradient Descent in Action

Gradients construct the manifold

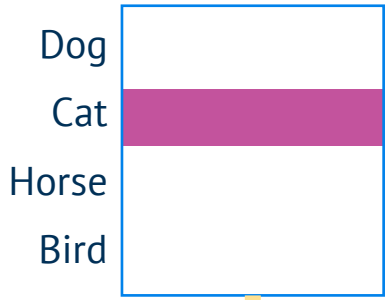
Network $f(\theta)$



Predicted Class Probability



Ground-Truth Label

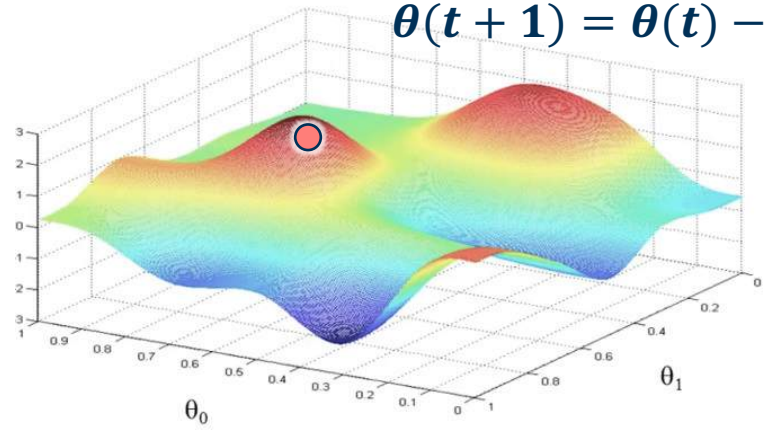


Loss $L(\theta)$

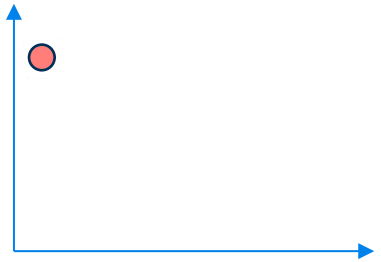
Backprop $L(\theta)$ to generate **gradients** $\nabla_\theta L(\theta)$

$$\theta(t+1) = \theta(t) - \alpha \nabla_\theta L(\theta)$$

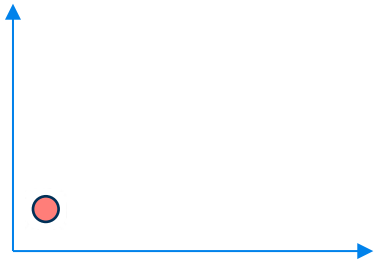
$L(\theta)$



Loss



Accuracy

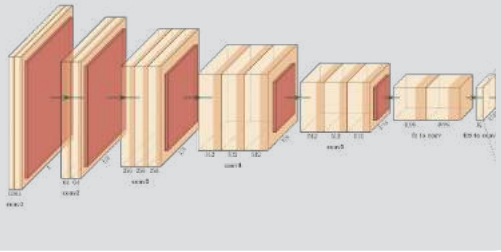


Training Neural Networks

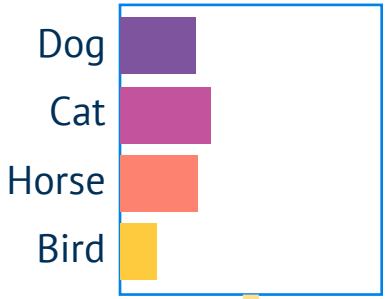
Gradient Descent in Action

Gradients construct the manifold

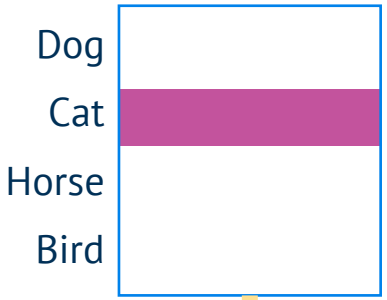
Network $f(\theta)$



Predicted Class Probability



Ground-Truth Label

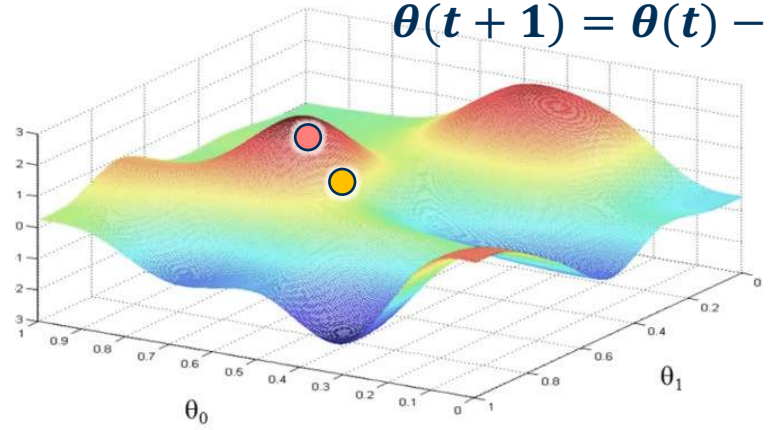


Loss $L(\theta)$

Backprop $L(\theta)$ to generate **gradients** $\nabla_\theta L(\theta)$

$$\theta(t+1) = \theta(t) - \alpha \nabla_\theta L(\theta)$$

$L(\theta)$



Loss

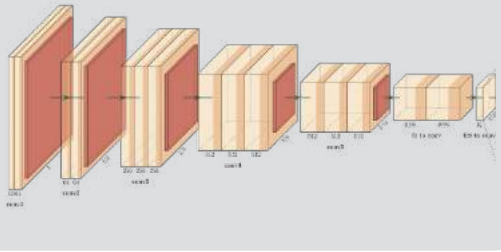
Accuracy

Training Neural Networks

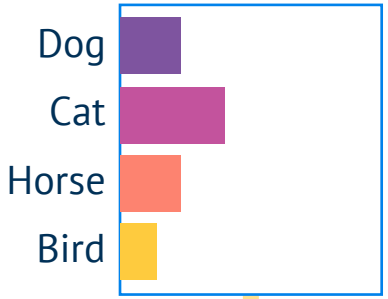
Gradient Descent in Action

Gradients construct the manifold

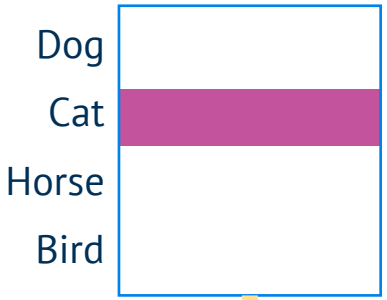
Network $f(\theta)$



Predicted Class Probability



Ground-Truth Label

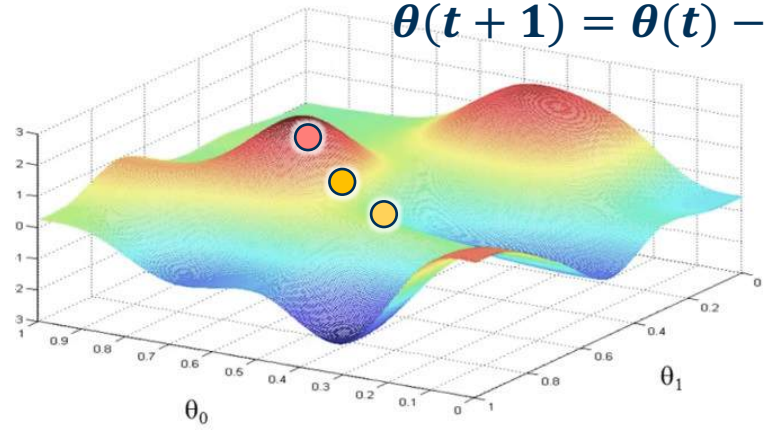


Loss $L(\theta)$

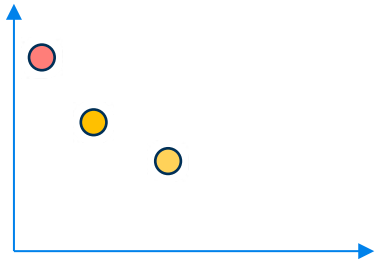
Backprop $L(\theta)$ to generate **gradients** $\nabla_\theta L(\theta)$

$$\theta(t+1) = \theta(t) - \alpha \nabla_\theta L(\theta)$$

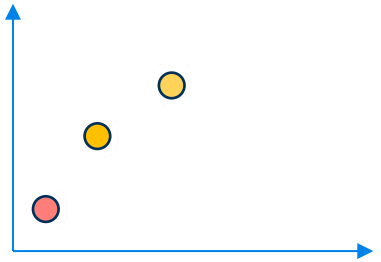
$L(\theta)$



Loss



Accuracy

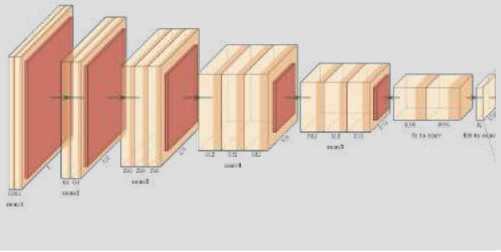


Training Neural Networks

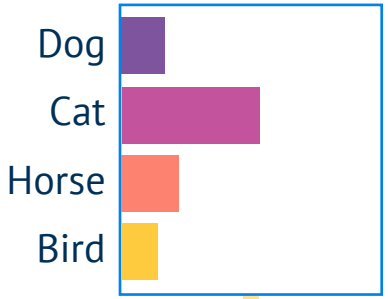
Gradient Descent in Action

Gradients construct the manifold

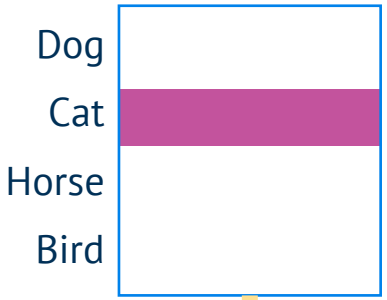
Network $f(\theta)$



Predicted Class Probability



Ground-Truth Label

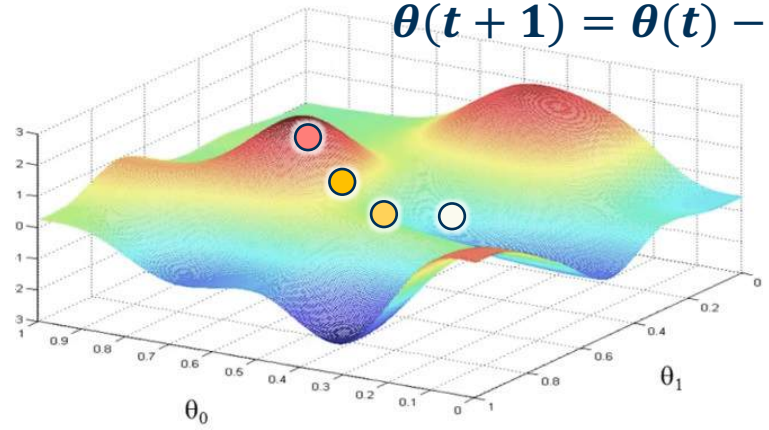


Loss $L(\theta)$

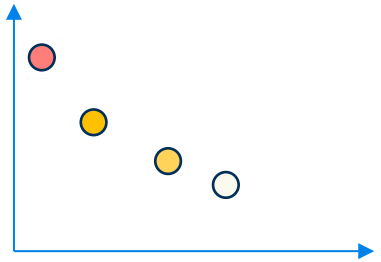
Backprop $L(\theta)$ to generate **gradients** $\nabla_\theta L(\theta)$

$$\theta(t+1) = \theta(t) - \alpha \nabla_\theta L(\theta)$$

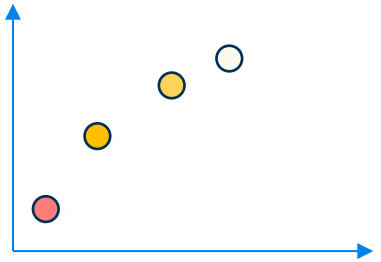
$L(\theta)$



Loss



Accuracy

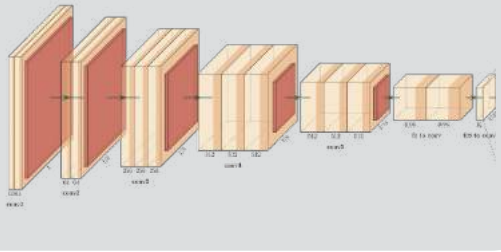


Training Neural Networks

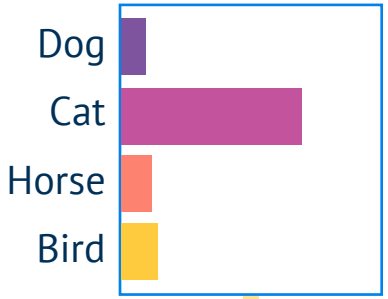
Gradient Descent in Action

Gradients construct the manifold

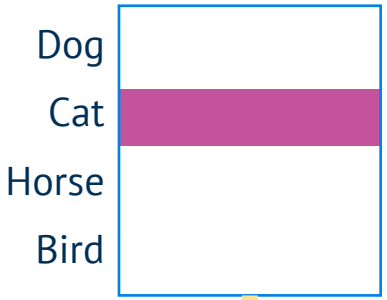
Network $f(\theta)$



Predicted Class Probability



Ground-Truth Label

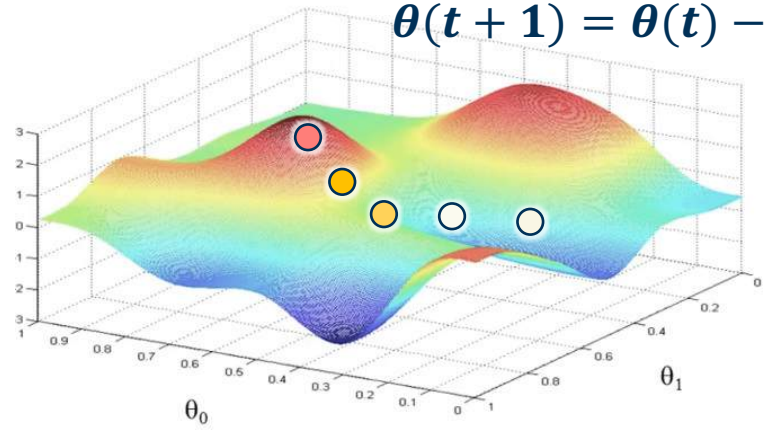


Loss $L(\theta)$

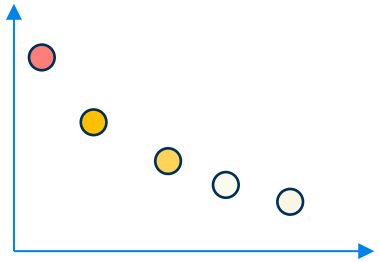
Backprop $L(\theta)$ to generate **gradients** $\nabla_\theta L(\theta)$

$$\theta(t+1) = \theta(t) - \alpha \nabla_\theta L(\theta)$$

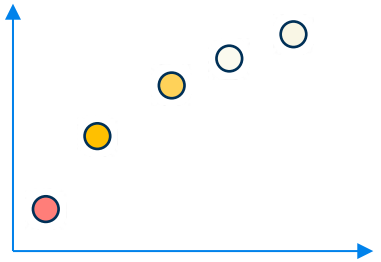
$L(\theta)$



Loss



Accuracy

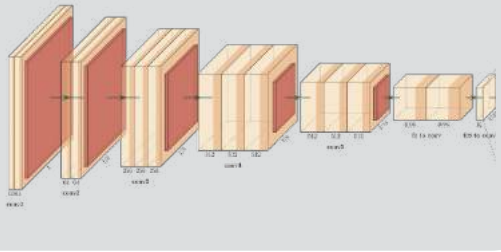


Training Neural Networks

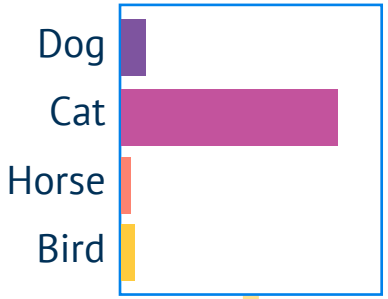
Gradient Descent in Action

Gradients construct the manifold

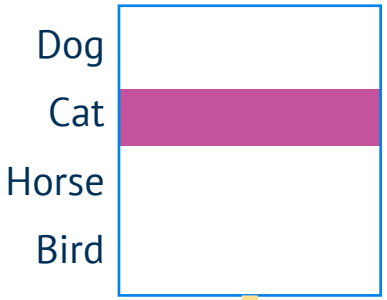
Network $f(\theta)$



Predicted Class Probability



Ground-Truth Label

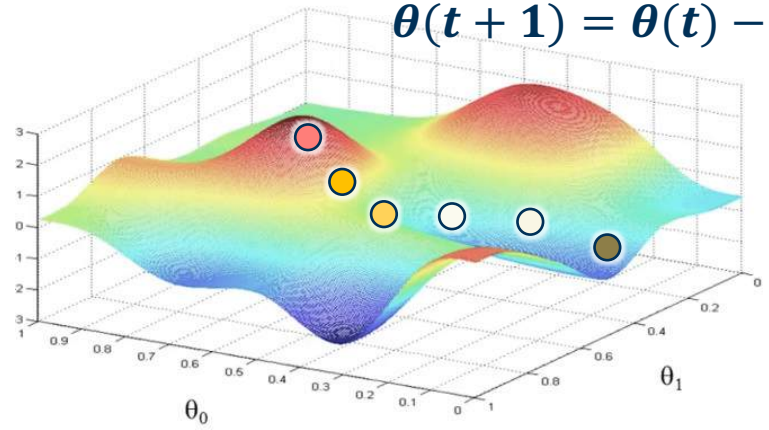


Loss $L(\theta)$

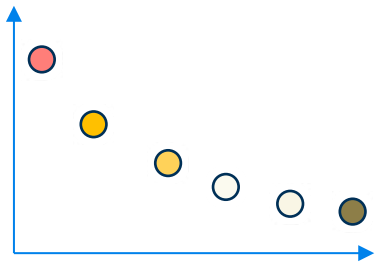
Backprop $L(\theta)$ to generate **gradients** $\nabla_\theta L(\theta)$

$$\theta(t+1) = \theta(t) - \alpha \nabla_\theta L(\theta)$$

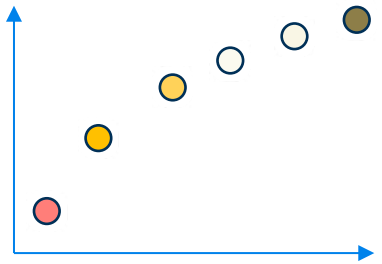
$L(\theta)$



Loss



Accuracy

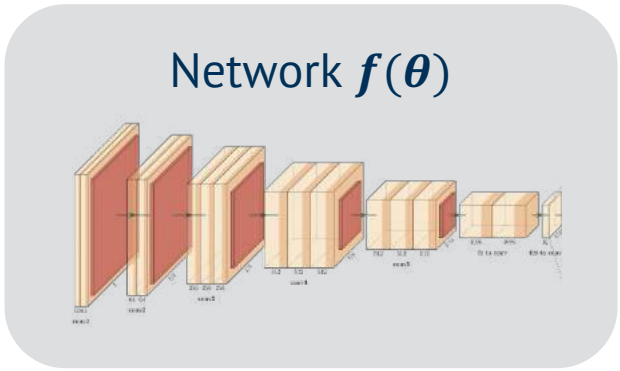


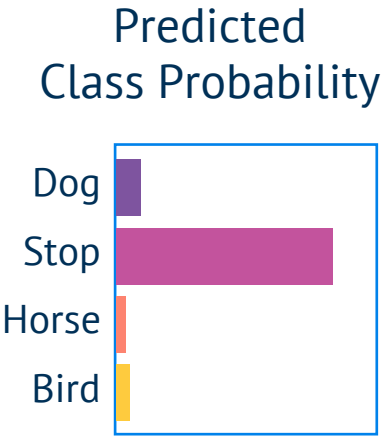
Our Goal

To Characterize Data at Inference

Goal: Given the novel data point, the network, and its prediction, *characterize* the data as a function of the learned knowledge

Given





Goal

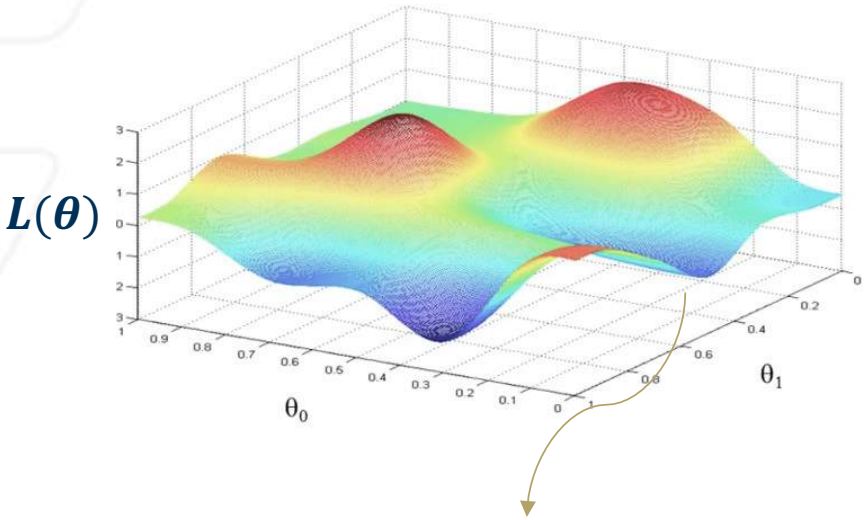
Represent the novel green traffic sign as a function of the learned red traffic sign

Our Claim: Gradients provide the methodology!

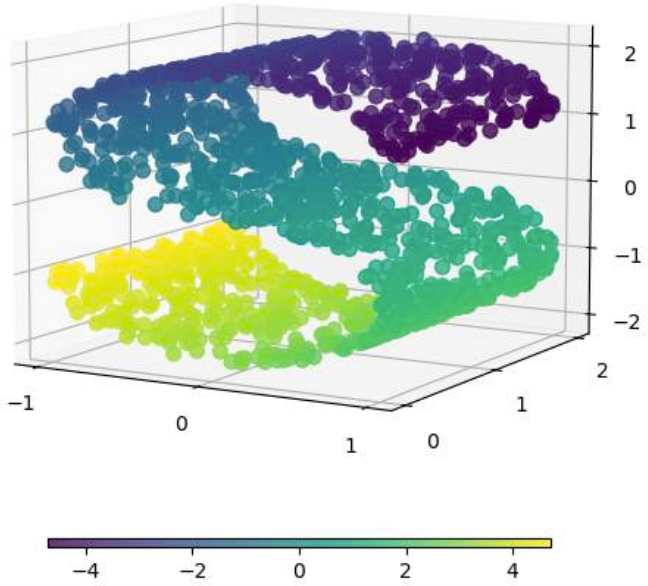
Challenges at Inference

A Quick note on Manifolds..

Manifolds are compact topological spaces that allow exact mathematical functions



Toy visualizations generated using functions (and thousands of generated data points)



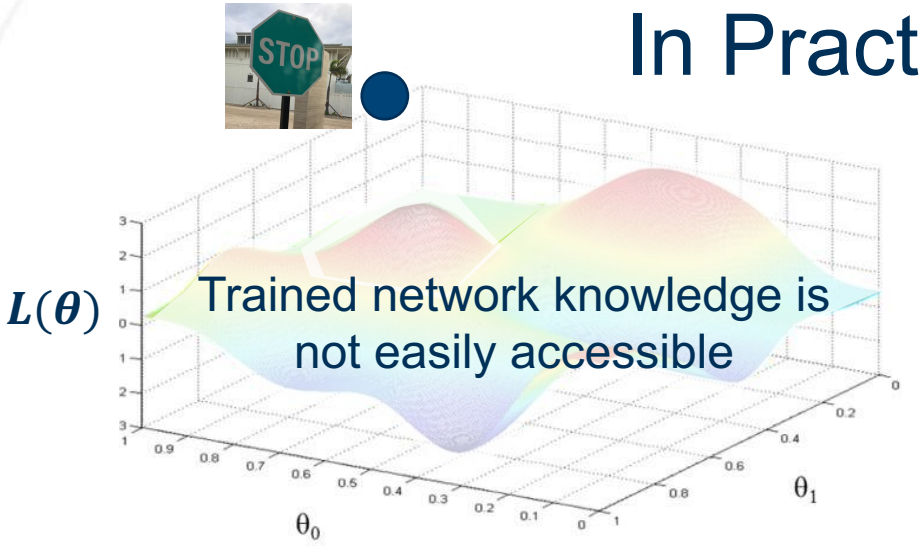
Real data visualizations generated using dimensionality reduction algorithms (Isomap)

Challenges at Inference

Manifolds at Inference

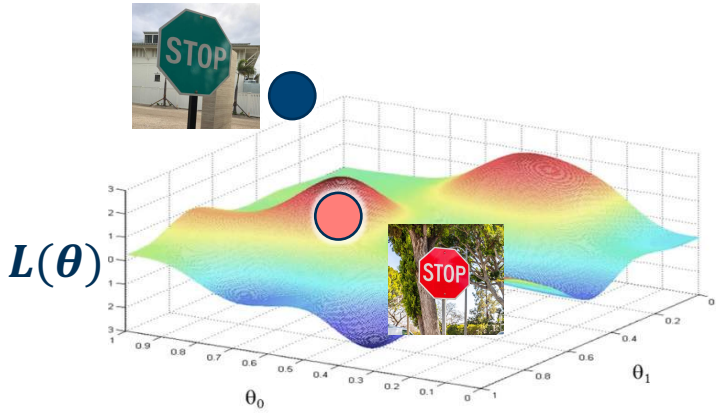
However, at inference only the test data point is available and the underlying structure of the manifold is unknown

In Practice



Existing methodologies estimate this manifold using surrogate networks and validation data at inference. However, they lose generalization performance.

Ideal Goal

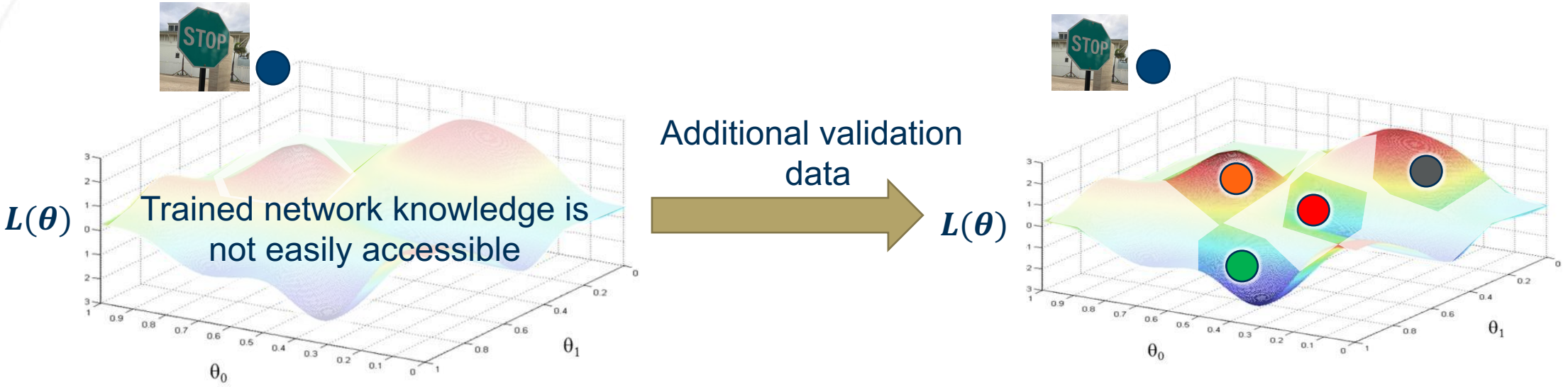


Represent the novel green traffic sign as a function of the learned red traffic sign

Challenges at Inference

Existing Solutions

Kim et.al.[1] use a KNN classifier on validation data at inference to characterize new test data



The surrogate (approximate) manifold is derived from K-Nearest Neighbors search

Cons of surrogates:

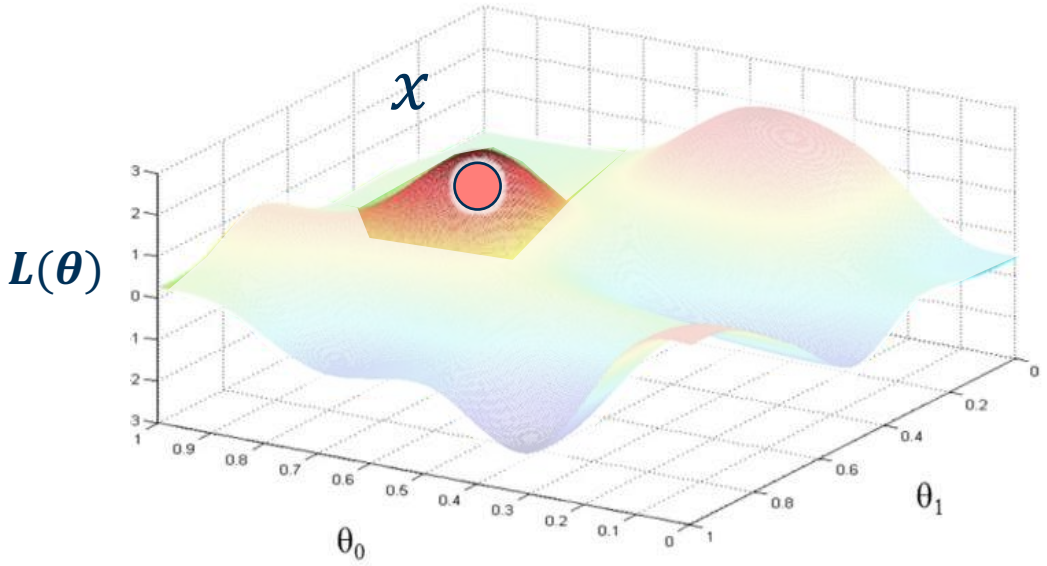
1. Requires a validation set at inference
2. Computationally impractical scale
3. Authors show that performance on anything greater than MNIST is comparable/worse than baseline

[1] Jiang, H., Kim, B., Guan, M., & Gupta, M. (2018). To trust or not to trust a classifier. *Advances in neural information processing systems*, *31*.

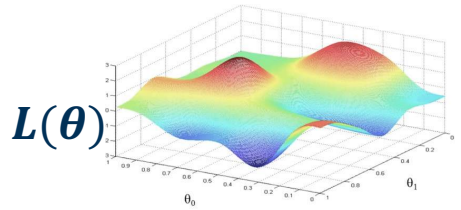
Relevant Properties of Gradients

Local Information

Gradients provide local information around the vicinity of x, even if x is novel. This is because x projects on the learned knowledge



Ideal



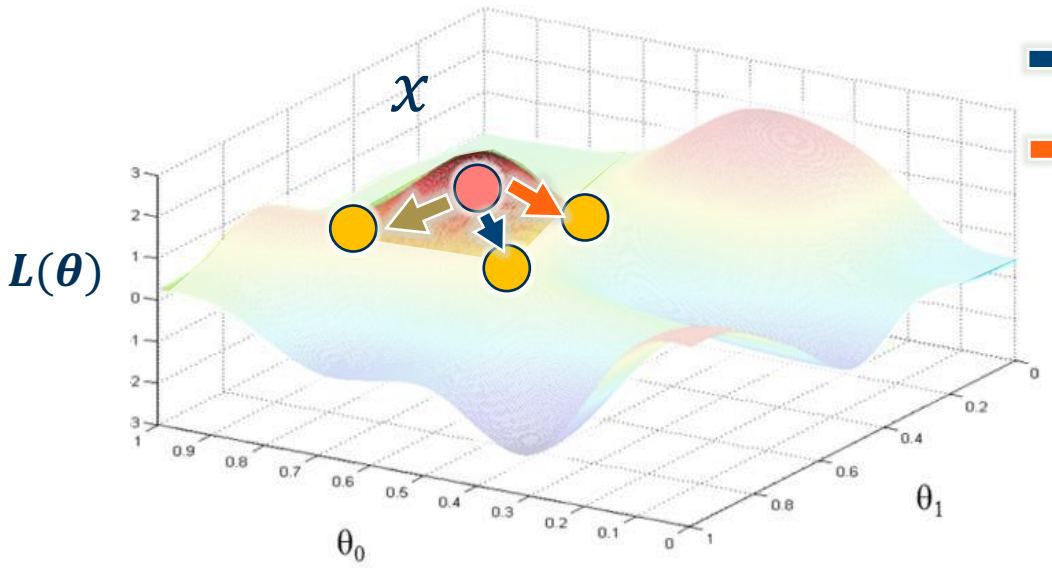
$\alpha \nabla_\theta L(\theta)$ provides local information up to a small distance α away from x

The exact nature and utility of this information is discussed in Part 2

Relevant Properties of Gradients

Direction of Steepest Descent

Gradients allow choosing the fastest direction of descent given a loss function $L(\theta)$



Path 1?

Path 2?

Path 3?

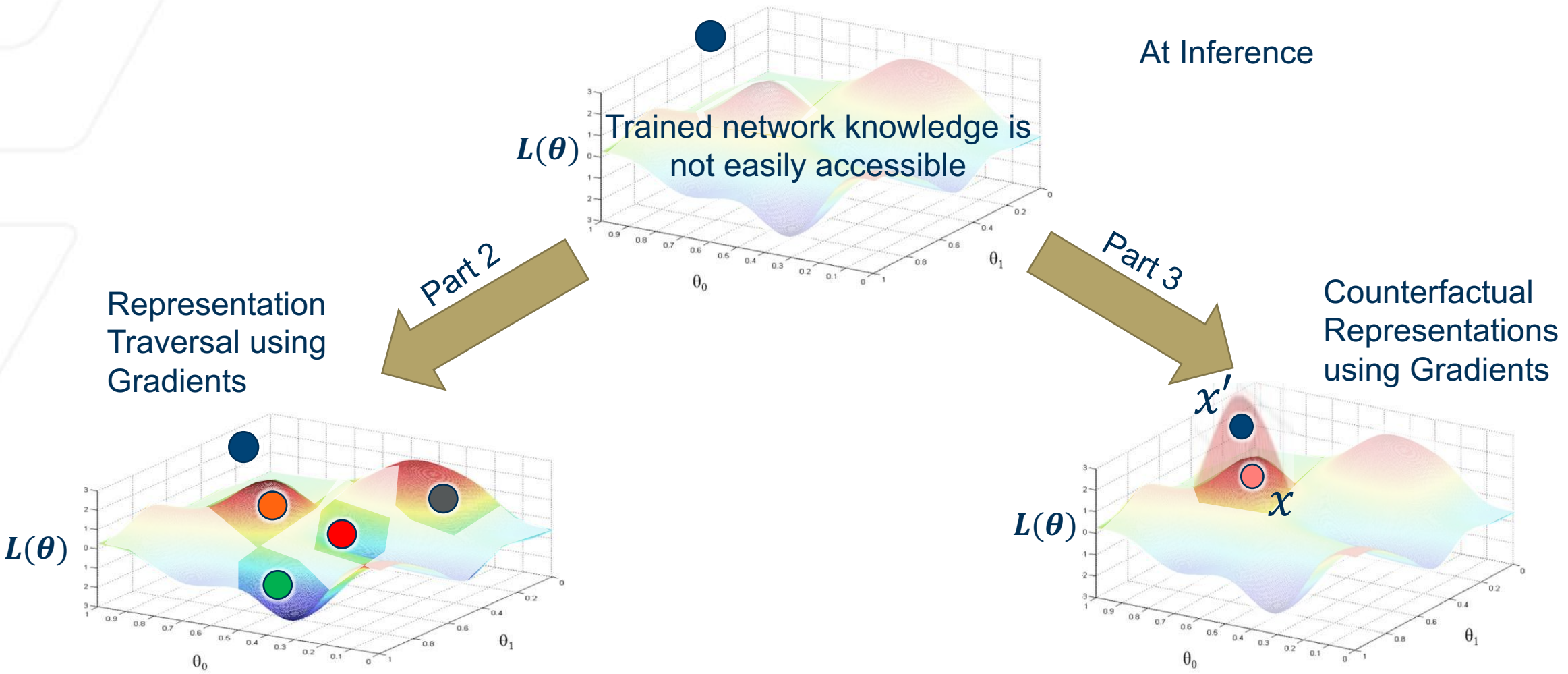
Which direction should we optimize towards (knowing only the local information)?

Negative of the gradient provides the **descent direction** towards the local minima, as measured by $L(\theta)$

The exact nature and utility of this directional information is discussed in Part 3

Our Technical Goal

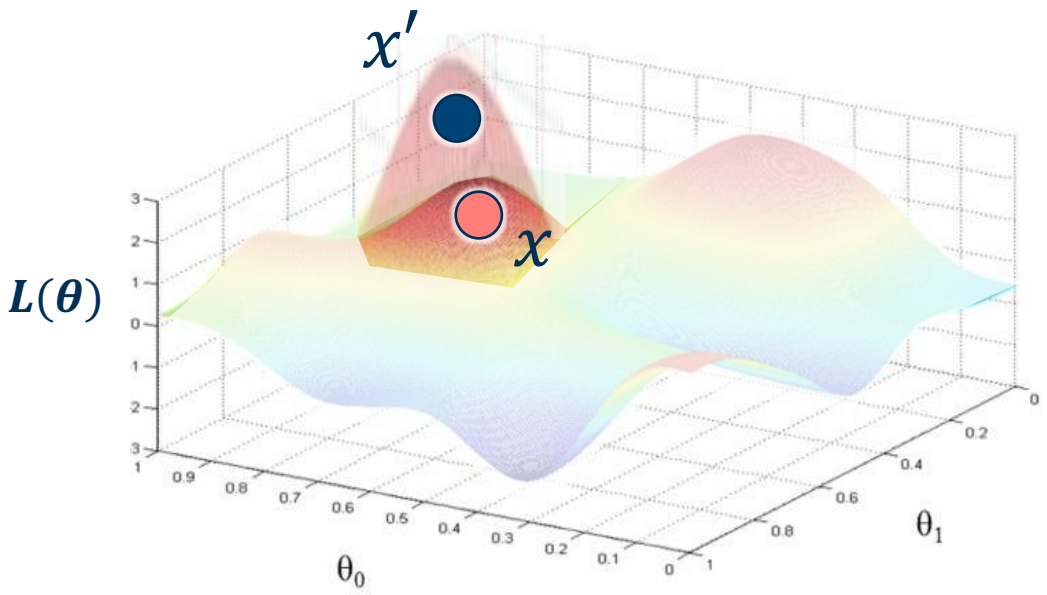
To Characterize the Learned Knowledge



Relevant Properties of Gradients

Counterfactual Manifolds

Gradients allow interventions either on the data or the manifolds to create counterfactuals



- Original manifold with x
- Counterfactual manifold with x'

Counterfactuals can be interpreted as changing the manifold to fit the new data

The exact nature and utility of these counterfactual manifolds is discussed in Part 4

Takeaways

Takeaways from Part 1

- **Part 1: Gradients in Neural Networks**
  - Deep Learning cannot easily generalize to novel data
  - Novel data cannot always be handled during Training
  - Gradients provide local information around the vicinity of $\boldsymbol{x}$
  - Gradients allow choosing the fastest direction of descent given a loss function $\boldsymbol{L(\theta)}$
  - Gradients allow interventions either on the data or the manifolds to create counterfactuals
- Part 2: Gradients as Information
- Part 3: Gradients as Uncertainty
- Part 4: Gradients as Expectancy-Mismatch
- Part 5: Conclusion and Future Directions