Reliability of Artificial Intelligence: Chances and Challenges

Gitta Kutyniok

(Ludwig-Maximilians-Universität München)

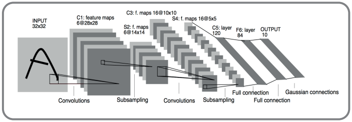
Barcelona Mathematics and Machine Learning Colloquium Series
Universitat Autònoma de Barcelona, March 11, 2024

LUDWIG-
MAXIMILIANS-
UNIVERSITÄT
MÜNCHEN

The Dawn of Artificial Intelligence in Public Life

Self-Driving Cars

Telecommunication/
Speech Recognition



Legal Issues

Health Care

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Artificial Intelligence = Alchemy?

AAAS | Science

AI researchers allege that machine learning is alchemy

By **Matthew Hutson** | May. 3, 2018 , 11:15 AM

Ali Rahimi, a researcher in artificial intelligence (AI) at Google in San Francisco, California, took a swipe at his field last December—and received a 40-second ovation for it. Speaking at an AI conference, Rahimi charged that machine learning algorithms, in which computers learn through trial and error, **have become a form of "alchemy."** Researchers, he said, do not know why some algorithms work and others don't, nor do they have rigorous criteria for choosing one AI architecture over another. Now, in a paper presented on 30 April at the International Conference on Learning Representations in Vancouver, Canada, Rahimi and his collaborators **document examples** of what they see as the alchemy problem and offer prescriptions for bolstering AI's rigor.

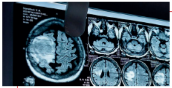
Problem with Reliability

Problems with Safety

Example:
Accidents involving robots

Problems with Security

Example:
Risks in self-driving cars



Problems with Privacy

Example:
Privacy violations of health data

Problems with Responsibility

Example:
Black-box and biased decisions

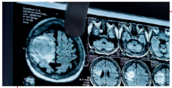
Problem with Reliability

Problems with Safety

Example:
Accidents involving robots

Problems with Security

Example:
Risks in self-driving cars



Problems with Privacy

Example:
Privacy violations of health data

Problems with Responsibility

Example:
Black-box and biased decisions

Current major problem worldwide:
Lack of reliability of AI technology!

Strong Requirements for Reliability

International Position on Reliable AI:

- AI Act of the European Union
- G7 Hiroshima AI Process

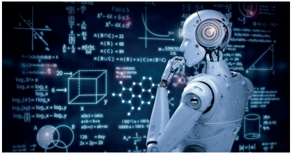
Strong Requirements for Reliability

International Position on Reliable AI:

- AI Act of the European Union
- G7 Hiroshima AI Process

Major Challenge:

Derive a profound mathematical understanding!



LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Delving Deeper into Artificial Intelligence...

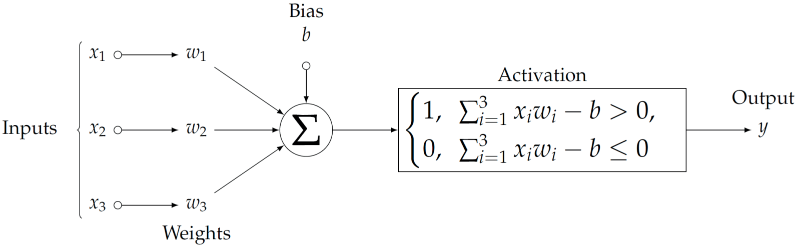
First Appearance of Neural Networks

Key Task of McCulloch and Pitts (1943):

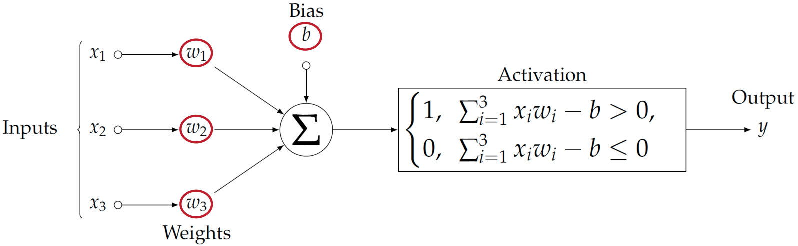
- Develop an algorithmic approach to learning.
- Mimicking the functionality of the human brain.

Goal: Artifical Intelligence!

Artificial Neurons



Artificial Neurons



Artificial Neurons

Definition: An *artificial neuron* with *weights* $w_1, ..., w_n \in \mathbb{R}$, *bias* $b \in \mathbb{R}$ and *activation function* $\rho : \mathbb{R} \to \mathbb{R}$ is defined as the function $f : \mathbb{R}^n \to \mathbb{R}$ given by

$$f(x_1, ..., x_n) = \rho\left(\sum_{i=1}^{n} x_i w_i - b\right) = \rho(\langle x, w \rangle - b),$$

where $w = (w_1, ..., w_n)$ and $x = (x_1, ..., x_n)$.

Artificial Neurons

Definition: An *artificial neuron* with *weights* $w_1, \ldots, w_n \in \mathbb{R}$, *bias* $b \in \mathbb{R}$ and *activation function* $\rho : \mathbb{R} \to \mathbb{R}$ is defined as the function $f : \mathbb{R}^n \to \mathbb{R}$ given by

$$f(x_1, \ldots, x_n) = \rho\left(\sum_{i=1}^{n} x_i w_i - b\right) = \rho(\langle x, w\rangle - b),$$

where $w = (w_1, \ldots, w_n)$ and $x = (x_1, \ldots, x_n)$.

Examples of Activation Functions:

- Heaviside function $\rho(x) = \begin{cases} 1, & x > 0, \\ 0, & x \leq 0. \end{cases}$
- Sigmoid function $\rho(x) = \frac{1}{1+e^{-x}}$.
- *Rectifiable Linear Unit (ReLU)* $\rho(x) = \max\{0, x\}$.

Affine Linear Maps and Weights

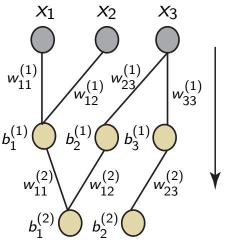
Remark: Concatenating artificial neurons leads to *compositions of affine linear maps and activation functions*.

Example: The following part of a neural network is given by

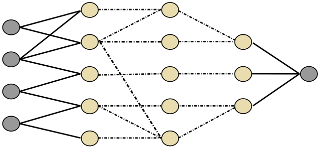
$$\Phi : \mathbb{R}^3 \to \mathbb{R}^2, \quad \Phi(x) = W^{(2)}\rho(W^{(1)}x + b^{(1)}) + b^{(2)}.$$

$$W^{(1)} = \begin{pmatrix} w_{11}^{(1)} & w_{12}^{(1)} & 0 \\ 0 & 0 & w_{23}^{(1)} \\ 0 & 0 & w_{33}^{(1)} \end{pmatrix}$$

$$W^{(2)} = \begin{pmatrix} w_{11}^{(2)} & w_{12}^{(2)} & 0 \\ 0 & 0 & w_{23}^{(2)} \end{pmatrix}$$



Definition of a Deep Neural Network



Definition:
Assume the following notions:

- $d \in \mathbb{N}$: Dimension of input layer.
- L: Number of layers.
- $\rho : \mathbb{R} \to \mathbb{R}$: (Non-linear) function called *activation function*.
- $T_\ell : \mathbb{R}^{N_{\ell-1}} \to \mathbb{R}^{N_\ell}$, $\ell = 1, \ldots, L$, where $T_\ell x = W^{(\ell)} x + b^{(\ell)}$

Then $\Phi : \mathbb{R}^d \to \mathbb{R}^{N_L}$ given by

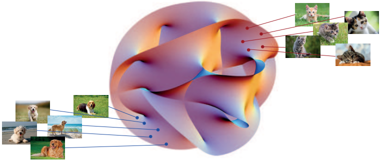
$$\Phi(x) = T_L \rho(T_{L-1} \rho(\ldots \rho(T_1(x)))), \quad x \in \mathbb{R}^d,$$

is called *(deep) neural network (DNN)*.

Training of Deep Neural Networks

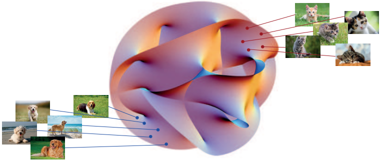
High-Level Set Up:

- Samples $(x_i, f(x_i))_{i=1}^m$ of a function such as $f : \mathcal{M} \to \{1, 2, \ldots, K\}$.
  $\rightsquigarrow$ *Training- and test data set.*

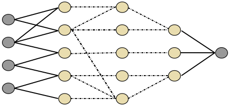


Training of Deep Neural Networks

High-Level Set Up:



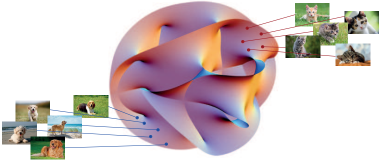
- Samples $(x_i, f(x_i))_{i=1}^m$ of a function such as $f : \mathcal{M} \to \{1, 2, \ldots, K\}$.
  $\rightsquigarrow$ *Training- and test data set.*



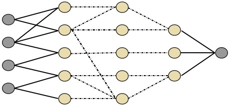
- Select an architecture of a deep neural network, i.e., a choice of d, L, $(N_\ell)_{\ell=1}^L$, and ρ.
  Sometimes selected entries of the matrices $(W^{(\ell)})_{\ell=1}^L$, *i.e., weights, are set to zero at this point.*

Training of Deep Neural Networks

High-Level Set Up:



- Samples $(x_i, f(x_i))_{i=1}^m$ of a function such as $f : \mathcal{M} \to \{1, 2, \ldots, K\}$.
  $\rightsquigarrow$ *Training- and test data set.*



- Select an architecture of a deep neural network, i.e., a choice of d, L, $(N_\ell)_{\ell=1}^L$, and ρ.
  Sometimes selected entries of the matrices $(W^{(\ell)})_{\ell=1}^L$, i.e., weights, are set to zero at this point.
- Learn the affine-linear functions $(T_\ell)_{\ell=1}^L = (W^{(\ell)} \cdot + b^{(\ell)})_{\ell=1}^L$ by

$$\min_{(W^{(\ell)}, b^{(\ell)})_\ell} \sum_{i=1}^m \mathcal{L}(\Phi_{(W^{(\ell)}, b^{(\ell)})_\ell}(x_i), f(x_i))$$

  yielding the network $\Phi_{(W^{(\ell)}, b^{(\ell)})_\ell} : \mathbb{R}^d \to \mathbb{R}^{N_L}$,

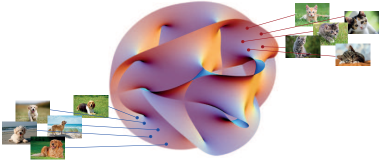
$$\Phi_{(W^{(\ell)}, b^{(\ell)})_\ell}(x) = T_L \rho(T_{L-1} \rho(\ldots \rho(T_1(x)))).$$

  This is often done by stochastic gradient descent.

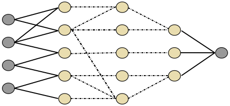
LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Training of Deep Neural Networks

High-Level Set Up:



- Samples $(x_i, f(x_i))_{i=1}^m$ of a function such as $f : \mathcal{M} \to \{1, 2, \ldots, K\}$.
  $\rightsquigarrow$ *Training- and test data set.*



- Select an architecture of a deep neural network, i.e., a choice of d, L, $(N_\ell)_{\ell=1}^L$, and ρ.
  Sometimes selected entries of the matrices $(W^{(\ell)})_{\ell=1}^L$, i.e., weights, are set to zero at this point.
- Learn the affine-linear functions $(T_\ell)_{\ell=1}^L = (W^{(\ell)} \cdot + b^{(\ell)})_{\ell=1}^L$ by

$$\min_{(W^{(\ell)}, b^{(\ell)})_\ell} \sum_{i=1}^m \mathcal{L}(\Phi_{(W^{(\ell)}, b^{(\ell)})_\ell}(x_i), f(x_i))$$

  yielding the network $\Phi_{(W^{(\ell)}, b^{(\ell)})_\ell} : \mathbb{R}^d \to \mathbb{R}^{N_L}$,

$$\Phi_{(W^{(\ell)}, b^{(\ell)})_\ell}(x) = T_L \rho(T_{L-1} \rho(\ldots \rho(T_1(x)))).$$

  This is often done by stochastic gradient descent.

Goal: $\Phi_{(W^{(\ell)}, b^{(\ell)})_\ell}(x_i) \approx f(x_i)$ for the test data!

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Main Research Directions

- **Expressivity:**
  - Which *aspects of a neural network architecture* affect the performance of deep learning?

$\rightsquigarrow$ *Applied Harmonic Analysis, Approximation Theory, ...*

Main Research Directions

- **Expressivity:**
  - Which *aspects of a neural network architecture* affect the performance of deep learning?

$\rightsquigarrow$ *Applied Harmonic Analysis, Approximation Theory, ...*

- **Learning:**
  - Why does *stochastic gradient descent* converge to good local minima despite the non-convexity of the problem?

$\rightsquigarrow$ *Algebraic/Differential Geometry, Optimal Control, Optimization, ...*

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Main Research Directions

- **Expressivity:**
  - Which *aspects of a neural network architecture* affect the performance of deep learning?

  $\rightsquigarrow$ *Applied Harmonic Analysis, Approximation Theory, ...*
- **Learning:**
  - Why does *stochastic gradient descent* converge to good local minima despite the non-convexity of the problem?

  $\rightsquigarrow$ *Algebraic/Differential Geometry, Optimal Control, Optimization, ...*
- **Generalization:**
  - Can we derive overall *success guarantees* (on the test data set)?

  $\rightsquigarrow$ *Learning Theory, Probability Theory, Statistics, ...*

Main Research Directions

- **Expressivity:**
  - Which *aspects of a neural network architecture* affect the performance of deep learning?

$\rightsquigarrow$ *Applied Harmonic Analysis, Approximation Theory, ...*

- **Learning:**
  - Why does *stochastic gradient descent* converge to good local minima despite the non-convexity of the problem?

$\rightsquigarrow$ *Algebraic/Differential Geometry, Optimal Control, Optimization, ...*

- **Generalization:**
  - Can we derive overall *success guarantees* (on the test data set)?

$\rightsquigarrow$ *Learning Theory, Probability Theory, Statistics, ...*

- **Explainability:**
  - Why did a trained deep neural network *reach a certain decision*?

$\rightsquigarrow$ *Information Theory, Uncertainty Quantification, ...*

Main Research Directions

- **Expressivity:**
  - Which *aspects of a neural network architecture* affect the performance of deep learning?

  $\rightsquigarrow$ *Applied Harmonic Analysis, Approximation Theory, ...*
- **Learning:**
  - Why does *stochastic gradient descent* converge to good local minima despite the non-convexity of the problem?

  $\rightsquigarrow$ *Algebraic/Differential Geometry, Optimal Control, Optimization, ...*
- **Generalization:**
  - Can we derive overall *success guarantees* (on the test data set)?

  $\rightsquigarrow$ *Learning Theory, Probability Theory, Statistics, ...*
- **Explainability:**
  - Why did a trained deep neural network *reach a certain decision*?

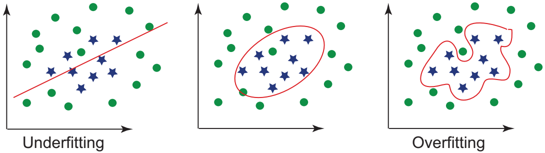
  $\rightsquigarrow$ *Information Theory, Uncertainty Quantification, ...*

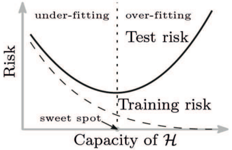
⚠ *Are there fundamental limitations?*

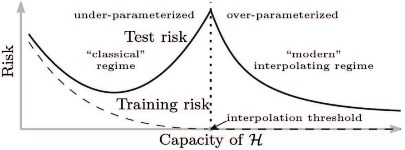
Let's start with generalization!

Generalization: Mysteries

Surprising Phenomenon:



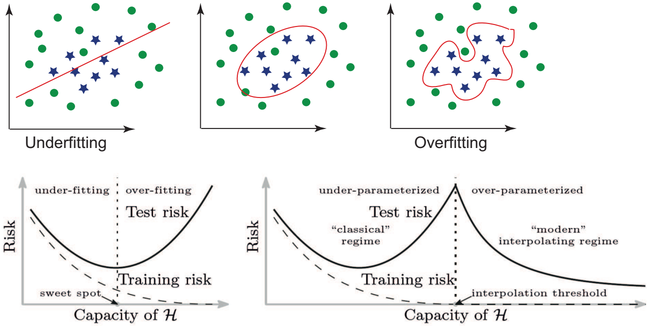




(Source: Belkin, Hsu, Ma, Mandal; 2019)

Generalization: Mysteries

Surprising Phenomenon:



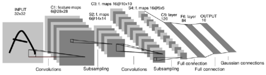
(Source: Belkin, Hsu, Ma, Mandal; 2019)

Common Approaches:

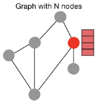
- VC dimension
- Rademacher complexity
- Neural tangent kernels

Some Facts about Graph Convolutional Neural Networks

Graph convolutional neural networks generalize classical CNNs to signals over graph domains. [Sperduti, Starita; 1997], [Gori, Monfardini, Scarselli; 2005], [Bruna, Zaremba, Szlam, LeCun; 2013], [Masci, Boscaini, Bronstein, Vandergheynst; 2015], ...



LeCun et al. 1989

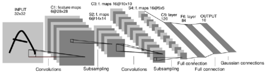


Graph signal: $s :$ graph nodes $\rightarrow \mathbb{R}^c$

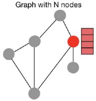
Graph CNN: graph signal $\rightarrow$ convolution $\rightarrow$ activation $\rightarrow$ pooling $\rightarrow \ldots$

Some Facts about Graph Convolutional Neural Networks

Graph convolutional neural networks generalize classical CNNs to signals over graph domains. [Sperduti, Starita; 1997], [Gori, Monfardini, Scarselli; 2005], [Bruna, Zaremba, Szlam, LeCun; 2013], [Masci, Boscaini, Bronstein, Vandergheynst; 2015], ...



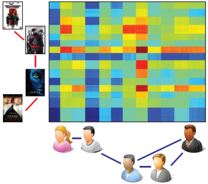
LeCun et al. 1989



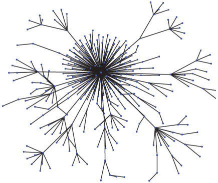
Graph signal: $s:$ graph nodes $\to \mathbb{R}^c$

Graph CNN: graph signal $\to$ convolution $\to$ activation $\to$ pooling $\to \ldots$

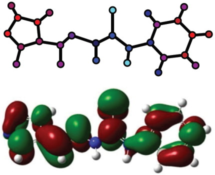
Some Applications:



Recommender system



Fake news detection

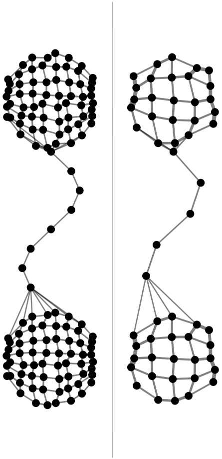


Chemistry

A Special Form of Generalization Capability

Desirable Feature:

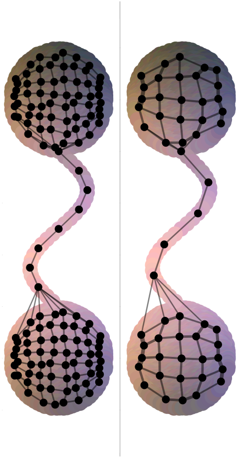
Graph convolutional neural networks should *generalize* to graphs and signals unseen in the training set.



A Special Form of Generalization Capability

Desirable Feature:
Graph convolutional neural networks should *generalize* to graphs and signals unseen in the training set.

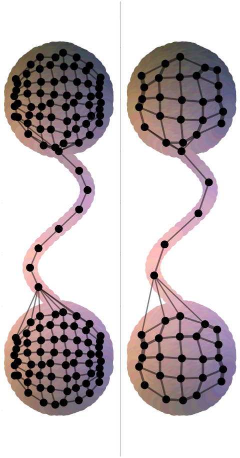
The Concept of Transferability:
If two graphs *model the same phenomenon*, a fixed filter/Graph CNN should have approximately the same repercussion on both graphs.



A Special Form of Generalization Capability

Desirable Feature:
Graph convolutional neural networks should *generalize* to graphs and signals unseen in the training set.

The Concept of Transferability:
If two graphs *model the same phenomenon*, a fixed filter/Graph CNN should have approximately the same repercussion on both graphs.



We prove transferability for spectral graph filters/Graph CNNs!

Graph Laplacian: Oscillations on Graphs

Definition: Let D be the degree matrix and W the adjacency matrix. Then the *unnormalized Graph Laplacian* is defined by

$$\Delta_u = D - W$$

and the *normalized Graph Laplacian* is given by

$$\Delta_n = D^{-1/2} \Delta_u D^{-1/2}.$$

As a generic notation, we will in the following use Δ.

Graph Laplacian: Oscillations on Graphs

Definition: Let D be the degree matrix and W the adjacency matrix. Then the *unnormalized Graph Laplacian* is defined by

$$\Delta_u = D - W$$

and the *normalized Graph Laplacian* is given by

$$\Delta_n = D^{-1/2} \Delta_u D^{-1/2}.$$

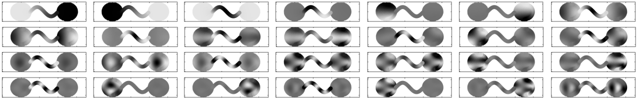
As a generic notation, we will in the following use Δ.

Remark: The Graph Laplacian Δ is self-adjoint. We will denote its

- eigenvalues by $\{\lambda_j\}_j \rightsquigarrow$ *Frequencies*,
- eigenvectors by $\{u_j\}_j \rightsquigarrow$ *Fourier modes*.

The graph Laplacian Δ encapsulates the geometry of the graph!

Graph Laplacian Eigenvectors



Spectral Graph Convolution

Definition:
Letting $\{u_j\}_j$ denote the eigenvectors of the graph Laplacian, we define the *spectral graph convolution operator* by

$$Cf = \sum_j c_j \langle f, u_j \rangle u_j.$$

Spectral Graph Convolution

Definition:

Letting $\{u_j\}_j$ denote the eigenvectors of the graph Laplacian, we define the *spectral graph convolution operator* by

$$Cf = \sum_j c_j \langle f, u_j \rangle u_j.$$

Problem with the Implementation:

- Computationally demanding
  - Eigendecomposition is slow.
  - No general FFT for graphs.
- Not transferable
  - The eigendecomposition is not stable to graph perturbations.
  - A fixed filter has different repercussions on similar graphs.

Spectral Graph Convolution

Definition:

Letting $\{u_j\}_j$ denote the eigenvectors of the graph Laplacian, we define the *spectral graph convolution operator* by

$$Cf = \sum_j c_j \langle f, u_j \rangle u_j.$$

Problem with the Implementation:

- Computationally demanding
  - Eigendecomposition is slow.
  - No general FFT for graphs.
- Not transferable
  - The eigendecomposition is not stable to graph perturbations.
  - A fixed filter has different repercussions on similar graphs.

Solution: Implement convolution using functional calculus!

Functional Calculus

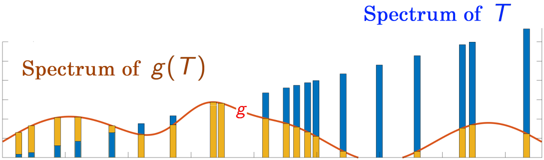
Definition:

Let T be a self-adjoint operator with discrete spectrum

$$Tv = \sum_j \lambda_j \langle v, u_j \rangle u_j.$$

A function $g : \mathbb{R} \to \mathbb{C}$ of T is then defined via

$$g(T)v = \sum_j g(\lambda_j) \langle v, u_j \rangle u_j.$$



Functional Calculus

Definition:

Let T be a self-adjoint operator with discrete spectrum

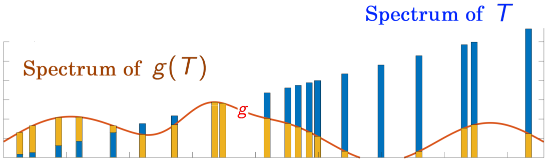
$$Tv = \sum_j \lambda_j \langle v, u_j \rangle u_j.$$

A function $g : \mathbb{R} \to \mathbb{C}$ of T is then defined via

$$g(T)v = \sum_j g(\lambda_j) \langle v, u_j \rangle u_j.$$

Remark:

If $g(\lambda) = \frac{\sum_{l=0}^{L} c_l \lambda^l}{\sum_{l=0}^{L} d_l \lambda^l}$, then $g(T) = \left(\sum_{l=0}^{L} c_l T^l\right) \left(\sum_{l=0}^{L} d_l T^l\right)^{-1}$.



Spectral Filtering using Functional Calculus

Functional Calculus Filters:

The functional calculus for $g : \mathbb{R} \to \mathbb{C}$ applied to the graph Laplacian yields

$$g(\Delta)f = \sum_j g(\lambda_j) \langle f, u_j \rangle u_j.$$

Recall:

The previous implementation used

$$Cf = \sum_j c_j \langle f, u_j \rangle u_j.$$

Spectral Filtering using Functional Calculus

Functional Calculus Filters:

The functional calculus for $g : \mathbb{R} \to \mathbb{C}$ applied to the graph Laplacian yields

$$g(\Delta)f = \sum_j g(\lambda_j) \langle f, u_j \rangle u_j.$$

Recall:

The previous implementation used

$$Cf = \sum_j c_j \langle f, u_j \rangle u_j.$$

Advantages of Functional Calculus Viewpoint:

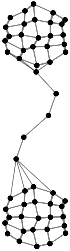
This approach...

- *...solves the instability problem* (Levie, Isufi, K; 2019).
- *...solves the computational problem*, if g is a rational function.

Graphs Modeling the Same Phenomenon

Interpretation:

- Weighted graphs:
  $\rightsquigarrow$ *Points and strength of correspondence between pairs of points.*



Graphs Modeling the Same Phenomenon

Interpretation:

- Weighted graphs:
  $\rightsquigarrow$ *Points and strength of correspondence between pairs of points.*
- Metric spaces:
  $\rightsquigarrow$ *Points and distances.*

Graphs Modeling the Same Phenomenon

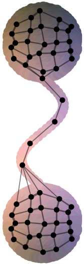
Interpretation:

- Weighted graphs:
  $\rightsquigarrow$ *Points and strength of correspondence between pairs of points.*
- Metric spaces:
  $\rightsquigarrow$ *Points and distances.*

Our Viewpoint:

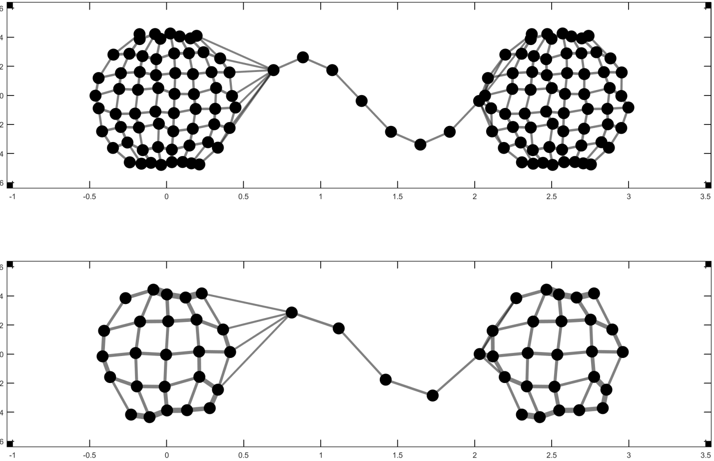
Think of graphs as discretizations of metric spaces

$$\text{distance} \nearrow \iff \text{edge weight} \searrow$$

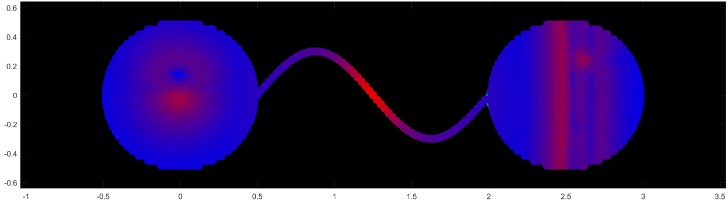


Graphs that represent the same phenomenon are discretizations of the same metric space!

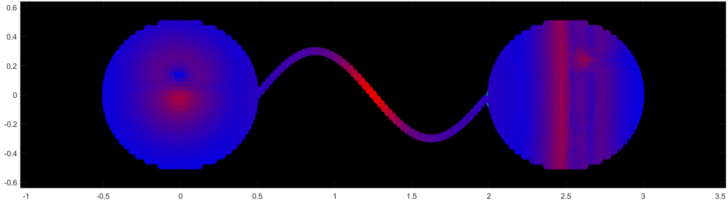
Comparing the Repercussion of a Filter on Two Graphs



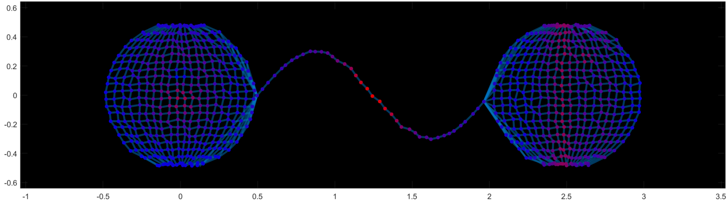
Comparing the Repercussion of a Filter on Two Graphs



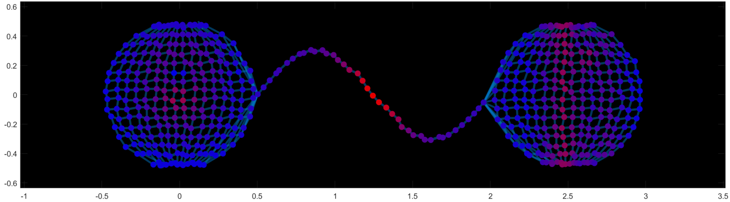
Take a generic signal $f : \mathcal{M} \to \mathbb{C}$



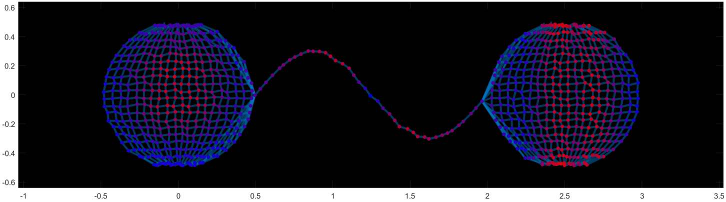
Comparing the Repercussion of a Filter on Two Graphs



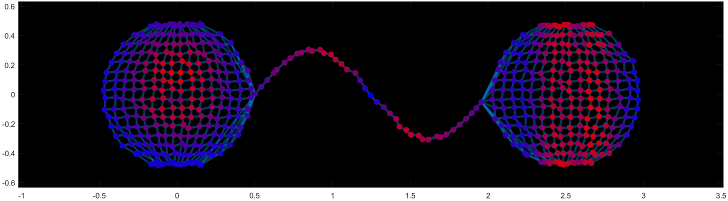
Sample to both graphs $S_1 f : G_1 \to \mathbb{C}, \quad S_2 f : G_2 \to \mathbb{C}$



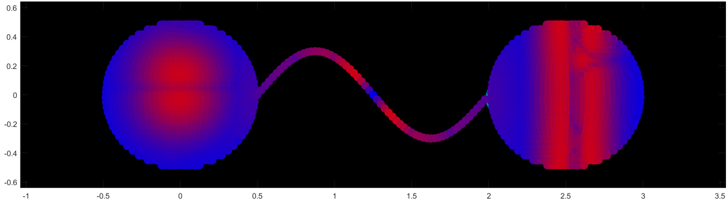
Comparing the Repercussion of a Filter on Two Graphs



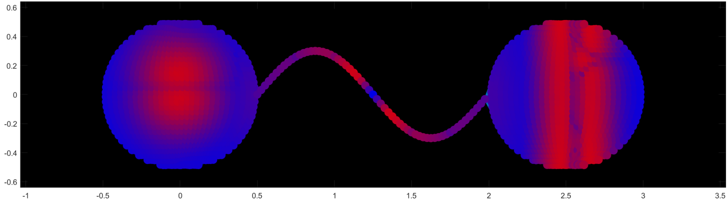
Apply both graph filters $\quad g(\mathbf{\Delta}_1)S_1 f, \quad g(\mathbf{\Delta}_2)S_2 f$



Comparing the Repercussion of a Filter on Two Graphs



Interpolate back to $L^2(\mathcal{M})$ to get $\|R_1 g(\mathbf{\Delta}_1) S_1 f - R_2 g(\mathbf{\Delta}_2) S_2 f\| \approx 0$

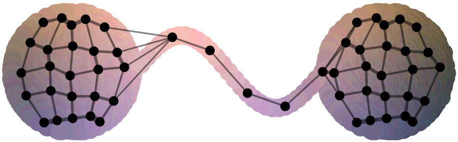


DSP Framework akin to the Nyquist–Shannon Approach

Our New Setting:

- *Analogue domain*: Borel space $\mathcal{M}$, with Laplacian $\mathcal{L}$.
- *Digital domains*: Graphs G with graph Laplacians Δ.
- *Paley Wiener spaces*: Band-limited spaces corresponding to $\mathcal{L}$.
- *Sampling operators*: $S^\lambda : PW(\lambda) \to L^2(G)$.
- *Interpolation operator*:

$$R^\lambda := (S^\lambda P(\lambda))^* := (S^\lambda P_{PW(\lambda)})^* : L^2(G) \to PW(\lambda).$$



What is Transferability precisely?

Definition:
The *transferability error of the filter f* on the signal $s \in L^2(\mathcal{M})$, is now defined by

$$\|f(\mathcal{L})s - R^{\lambda} f(\Delta) S^{\lambda} s\|,$$

the *transferability error of the Laplacian* is defined by

$$\|\mathcal{L}s - R^{\lambda} \Delta S^{\lambda} s\|,$$

and the *consistency error* is defined by

$$\|s - R^{\lambda} S^{\lambda} s\|.$$

Transferability of Functional Calculus Filters

Theorem (Levie, Huang, Bucci, Bronstein, K; 2021): Let

- $\lambda_M > 0$ be a band with $\|R^{\lambda_M}\| < C$,
- $g : \mathbb{R} \to \mathbb{C}$ be Lipschitz continuous with constant D,
- $\|g\|_{\mathcal{L},M} = \max_{0 \leq m \leq M}\{|g(\lambda_m)|\}$.

Then

$$\|g(\mathcal{L})P(\lambda_M) - R^{\lambda_M} g(\Delta) S^{\lambda_M} P(\lambda_M)\|$$
$$\leq DC\sqrt{M}\|S^{\lambda_M}\mathcal{L}P(\lambda_M) - \Delta S^{\lambda_M} P(\lambda_M)\| + \|g\|_{\mathcal{L},M}\|P(\lambda_M) - R^{\lambda_M} S^{\lambda_M} P(\lambda_M)\|$$

and

$$\|g(\mathcal{L})q - R_{\lambda_M} g(\Delta) S^{\lambda_M} q\|$$
$$\leq DC \sum_{m=0}^{M} |c_m| \|S^{\lambda_M}\mathcal{L}\phi_m - \Delta S^{\lambda_M}\phi_m\| + \|g\|_{\mathcal{L},M}\|q - R^{\lambda_M} S^{\lambda_M} q\|,$$

where $q = \sum_{m=0}^{M} c_m \phi_m \in PW(\lambda_M) \subset L^2(\mathcal{M})$.

Transferability of Filter
$\leq$ *Transferability of Laplacian* $+$ *Consistency Error*

Transferability of Functional Calculus CNNs

Theorem (Levie, Huang, Bucci, Bronstein, K; 2021):
Consider two graphs G_j, $j = 1,2$ and two graph Laplacians Δ_j, $j = 1,2$, approximating the same Laplacian $\mathcal{L}$ in $\mathcal{M}$, and consider a ReLU graph CNN with Lipschitz filters. Further, let $G_{j,l}$ be the graph in layer l with graph Laplacians $\Delta_{j,l}$. Also, assume that, for all layers l, bands λ_l, and $j = 1,2$,

$$\|S_{j,l}^{\lambda_l}\mathcal{L}P(\lambda_l) - \Delta_{j,l}S_{j,l}^{\lambda_l}P(\lambda_l)\| \leq \delta$$

and

$$\|P(\lambda_L) - R_{j,L}^{\lambda_L}S_{j,L}^{\lambda_L}P(\lambda_L)\| \leq \delta$$

for some $0 < \delta < 1$. Then, for all output-channels k and mappings $\Phi_{j,L}^k$ given by the graph CNN,

$$\begin{aligned}\|R_{1,L}^{\lambda_L}\Phi_{1,L}^k S_{1,1}^{\lambda_0}P(\lambda_0) - R_{2,L}^{\lambda_L}\Phi_{2,L}^k S_{2,1}^{\lambda_0}P(\lambda_0)\|& \\ \leq 2\Big(LD\sqrt{\dim(PW(\lambda))} + L + 1\Big)\delta.&\end{aligned}$$

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Main Research Directions

- **Expressivity:**
  - Which *aspects of a neural network architecture* affect the performance of deep learning?

  $\rightsquigarrow$ *Applied Harmonic Analysis, Approximation Theory, ...*
- **Learning:**
  - Why does *stochastic gradient descent* converge to good local minima despite the non-convexity of the problem?

  $\rightsquigarrow$ *Algebraic/Differential Geometry, Optimal Control, Optimization, ...*
- **Generalization:**
  - Can we derive overall *success guarantees* (on the test data set)?

  $\rightsquigarrow$ *Learning Theory, Probability Theory, Statistics, ...*
- **Explainability:**
  - Why did a trained deep neural network *reach a certain decision*?

  $\rightsquigarrow$ *Information Theory, Uncertainty Quantification, ...*

!

Are there fundamental limitations?

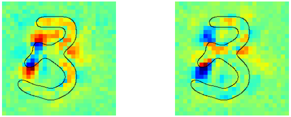
LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

An Applied Harmonic Analysis Approach to Explainability

Explainability

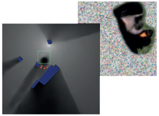
Main Goal: We aim to *understand* decisions of “black-box” predictors!

map for digit 3 map for digit 8



Selected Questions:

- What *exactly* is relevance in a mathematical sense?
- Can we develop a theory for *optimal relevance maps*?
- How to extend to *challenging modalities*?



Vision:

Questioning the AI as a human about the reason for a decision!

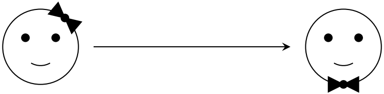
Rate-Distortion Viewpoint

The Setting: Let

- $\Phi\colon [0,1]^d \to [0,1]$ be a *classification function*,
- $x \in [0,1]^d$ be an *input signal*.

Alice

Bob



$\Phi(x) = 0.97$

"Monkey"

Original image x

Partial image S

Random completion y

$\Phi(y) = 0.91$

"Monkey"

Expected Distortion:

$$D(S) = D(\Phi, x, S) = \mathbb{E}\left[\frac{1}{2}\left(\Phi(x) - \Phi(y)\right)^2\right]$$

Rate-Distortion Explanation

Rate-Distortion Function:

$$R(\epsilon) = \min_{S \subseteq \{1,\dots,d\}} \{|S| \,:\, D(S) \leq \epsilon\}$$

$\rightsquigarrow$ *Use this viewpoint for the definition of a relevance map!*

Rate-Distortion Explanation

Rate-Distortion Function:

$$R(\epsilon) = \min_{S \subseteq \{1,\ldots,d\}} \{|S| \,:\, D(S) \leq \epsilon\}$$

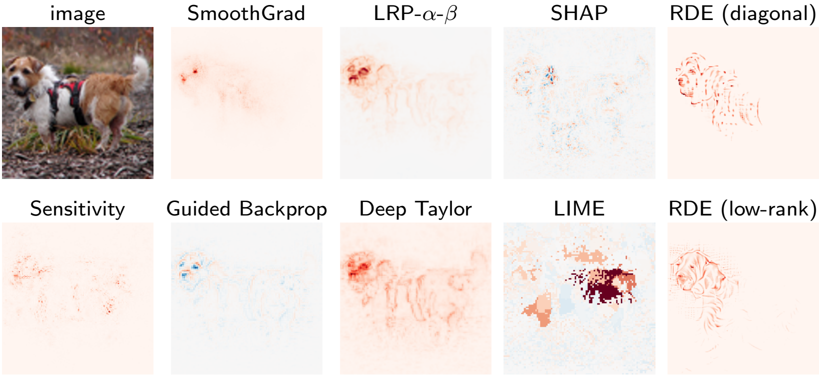
$\rightsquigarrow$ *Use this viewpoint for the definition of a relevance map!*

Finding a minimizer of $R(\epsilon)$ is very hard!
(Wäldchen, Macdonald, Hauch, K, 2020)

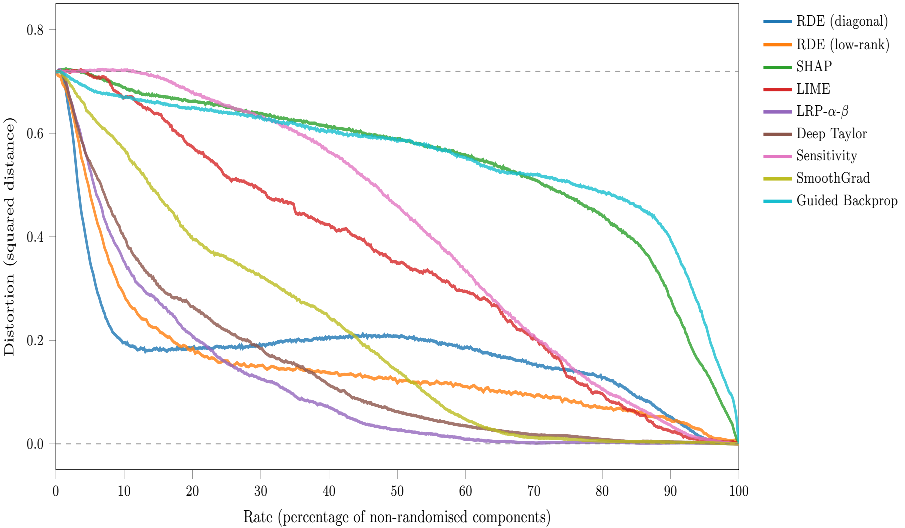
Relaxed (and computable) Variant (RDE) (Macdonald, Wäldchen, Hauch, K, 2020):

$$\boxed{\text{minimize} \quad D(s) + \lambda \|s\|_1 \quad \text{subject to} \quad s \in [0,1]^d}$$

STL-10 Experiment



STL-10 Experiment



SmoothGrad (Smilkov, Thorat, Kim, Viégas, Wattenberg, 2017), Layer-wise Relevance Propagation (Bach, Binder, Montavon, Klauschen, Müller, Samek, 2015), SHAP (Lundberg, Lee, 2017), Sensitivity Analysis (Simonyan, Vedaldi, Zisserman, 2013), Guided Backprop (Springenberg, Dosovitskiy, Brox, Riedmiller, 2015), Deep Taylor Decompositions (Montavon, Samek, Müller, 2018), LIME (Ribeiro, Singh, Guestrin, 2016)

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Desiderata

Problems:

- Modifying the image with random noise or some background color might lead to the obfuscation not being in the domain of the network.
  $\rightsquigarrow$ *Does this give meaningful information about why the network made its decisions?*
- The explanations are pixel-based.
  $\rightsquigarrow$ *Does this lead to useful information for different modalities?*

Desiderata

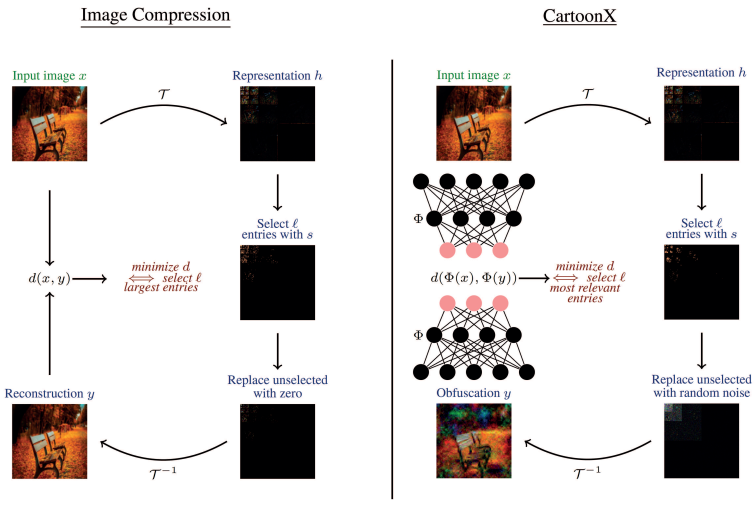
Problems:

- Modifying the image with random noise or some background color might lead to the obfuscation not being in the domain of the network.
  $\rightsquigarrow$ *Does this give meaningful information about why the network made its decisions?*
- The explanations are pixel-based.
  $\rightsquigarrow$ *Does this lead to useful information for different modalities?*

Solutions:

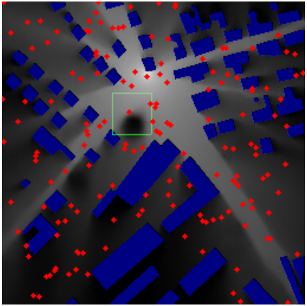
- *Take the conditional data distribution into account using an inpainting GAN!*
- *Use a decomposition of the data and place relevance scores on the (wavelet, etc.) coefficients!*

Cartoon X (Kolek, Nguyen, Levie, Bruna, K; 2022)

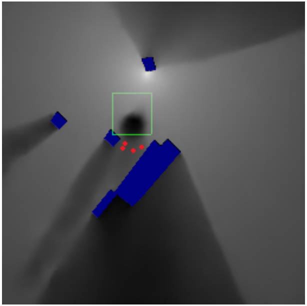


Telecommunication

RadioUNet (Levie, Cagkan, K, Caire; 2021):



Estimated map



Explanation

Detecting Reason for Adversarial Examples

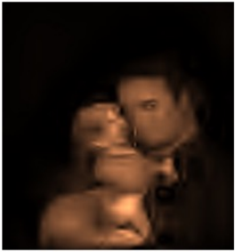
CartoonX (Kolek, Nguyen, Levie, Bruna, K; 2022):

Diaper

Screw

Detecting Reason for Adversarial Examples

CartoonX (Kolek, Nguyen, Levie, Bruna, K; 2022):

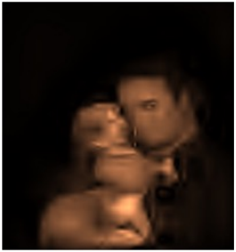


Diaper

Screw

Detecting Reason for Adversarial Examples

CartoonX (Kolek, Nguyen, Levie, Bruna, K; 2022):



Diaper

Screw

$\rightsquigarrow$ *ShearletX* (Kolek, Windesheim, Loarca, K, Levie; 2023)

Main Research Directions

- **Expressivity:**
  - Which *aspects of a neural network architecture* affect the performance of deep learning?

$\rightsquigarrow$ *Applied Harmonic Analysis, Approximation Theory, ...*

- **Learning:**
  - Why does *stochastic gradient descent* converge to good local minima despite the non-convexity of the problem?

$\rightsquigarrow$ *Algebraic/Differential Geometry, Optimal Control, Optimization, ...*

- **Generalization:**
  - Can we derive overall *success guarantees* (on the test data set)?

$\rightsquigarrow$ *Learning Theory, Probability Theory, Statistics, ...*

- **Explainability:**
  - Why did a trained deep neural network *reach a certain decision*?

$\rightsquigarrow$ *Information Theory, Uncertainty Quantification, ...*

Are there fundamental limitations?

Deep Neural Networks are Not a Swiss Army Knife!

They do have Limitations!

A Serious Problem

A Serious Problem

Computability on Digital Machines (informal):

A *computable problem (function)* is one for which the input-output relation can be computed on a digital machine for any given accuracy.

A Serious Problem

Computability on Digital Machines (informal):

A *computable problem (function)* is one for which the input-output relation can be computed on a digital machine for any given accuracy.

Theorem (Boche, Fono, K; 2022):
The solution of a finite-dimensional inverse problem is *not (Turing) computable* (by a deep neural network).

A Serious Problem

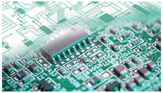
Computability on Digital Machines (informal):

A *computable problem (function)* is one for which the input-output relation can be computed on a digital machine for any given accuracy.

Theorem (Boche, Fono, K; 2022):
The solution of a finite-dimensional inverse problem is *not (Turing) computable* (by a deep neural network).

General Barrier:

- Limits of **computability** on today's hardware



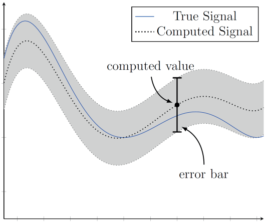
Today computations are performed almost exclusively on digital hardware!

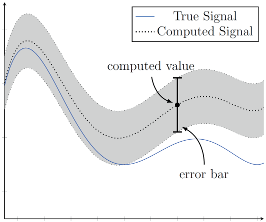
Some Thoughts on the Result

Serious Problems:

- *No algorithm exists*, which on digital hardware derives neural networks approximating the solution for any given accuracy.
- The output of trained neural networks *not reliable (no guarantees)*.
- This result could point towards why *instabilities* and *non-robustness* occurs for deep neural networks.

Illustration of the Problem:





What now? ... Mathematics Tells Us the Answer!

What now? ... Mathematics Tells Us the Answer!

Theorem (Boche, Fono, K; 2022):
The solution of a finite-dimensional inverse problem is *computable* (by a deep neural network) *on an analog (Blum-Shub-Smale) machine!*

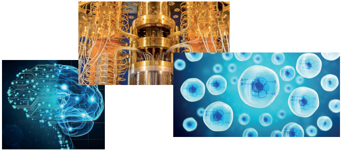
What now? ... Mathematics Tells Us the Answer!

Theorem (Boche, Fono, K; 2022):
The solution of a finite-dimensional inverse problem is *computable* (by a deep neural network) *on an analog (Blum-Shub-Smale) machine!*

Reliability for certain problem settings requires novel hardware!

Possible Future Developments:

- *Neuromorphic computing*
- Biocomputing
- *Quantum computing*



The Situation is Even More Serious!

The Situation is Even More Serious!

Theorem (Boche, Fono, K; 2024): Many classification problems are also *not (Turing) computable*!

The Situation is Even More Serious!

Theorem (Boche, Fono, K; 2024): Many classification problems are also *not (Turing) computable*!

Theorem (Boche, Fono, K; 2024): The Pseudo Inverse is *not (Turing) computable*!

The Situation is Even More Serious!

Theorem (Boche, Fono, K; 2024): Many classification problems are also *not (Turing) computable*!

Theorem (Boche, Fono, K; 2024): The Pseudo Inverse is *not (Turing) computable*!

Theorem (Bacho, Boche, K; 2024): Computing the solutions to the Laplace and the diffusion equation on digital hardware causes a *complexity blowup*.

The Situation is Even More Serious!

Theorem (Boche, Fono, K; 2024): Many classification problems are also *not (Turing) computable*!

Theorem (Boche, Fono, K; 2024): The Pseudo Inverse is *not (Turing) computable*!

Theorem (Bacho, Boche, K; 2024): Computing the solutions to the Laplace and the diffusion equation on digital hardware causes a *complexity blowup*.

Theorem (Lee, Boche, K; 2024): Finding the solution of most optimization problems is *not (Turing) computable*!

The Situation is Even More Serious!

Theorem (Boche, Fono, K; 2024): Many classification problems are also *not (Turing) computable*!

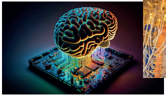
Theorem (Boche, Fono, K; 2024): The Pseudo Inverse is *not (Turing) computable*!

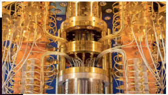
Theorem (Bacho, Boche, K; 2024): Computing the solutions to the Laplace and the diffusion equation on digital hardware causes a *complexity blowup*.

Theorem (Lee, Boche, K; 2024): Finding the solution of most optimization problems is *not (Turing) computable*!

Vision for the Future:

Mathematically Reliable AI...by Analog Computing!



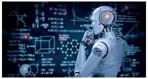


Some Final Thoughts...

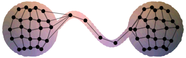
Conclusions

Artificial Intelligence:

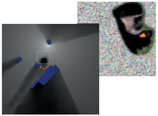
- *Impressive performance* in real-world applications!
- A *mathematical foundation* of it is largely missing!

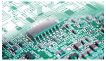


Mathematics for Artificial Intelligence:



- *Expressivity*: Optimal architectures?
- *Learning*: Controllable, efficient algorithms?
- *Generalization*: Performance on test data sets?
- *Explainability*: Explaining network decisions?

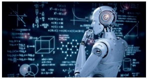




Caution: Problems with computability on digital hardware!

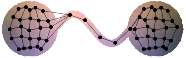
Conclusions

Artificial Intelligence:

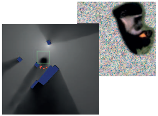


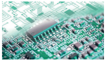
- *Impressive performance* in real-world applications!
- A *mathematical foundation* of it is largely missing!

Mathematics for Artificial Intelligence:



- *Expressivity*: Optimal architectures?
- *Learning*: Controllable, efficient algorithms?
- *Generalization*: Performance on test data sets?
- *Explainability*: Explaining network decisions?





Caution: Problems with computability on digital hardware!

Exciting Future Perspectives for Mathematical Foundations!

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

New AI School in Munich (MSc & PhD)

Konrad Zuse School of Excellence in Reliable AI

(https://zuseschoolrelai.de)

SPONSORED BY THE

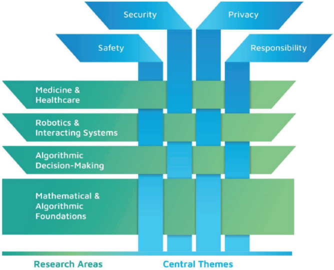
Federal Ministry of Education and Research

DAAD Zuse Schools
Konrad Zuse Schools of Excellence in Artificial Intelligence

relAI
Konrad Zuse School of Excellence in Reliable AI

Technische Universität München — TUM

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN



Mission: Train future generations of AI experts in Germany who combine technical brilliance with awareness of the importance of AI's reliability

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

THANK YOU!

References available at:

www.ai.math.lmu.de/kutyniok

Survey Paper (arXiv:2105.04026):

Berner, Grohs, K, Petersen, *The Modern Mathematics of Deep Learning*.

Check related information on Twitter (@GittaKutyniok) and LinkedIn

Related Book:

- Grohs and K, eds.,
  Mathematical Aspects of Deep Learning
  Cambridge University Press, 2022.