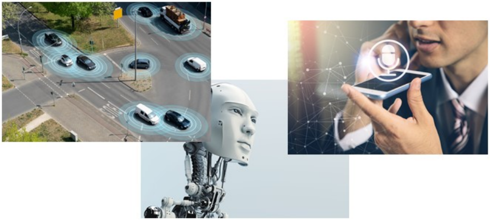
Lecture 2: Reliable AI: Dream or Reality?

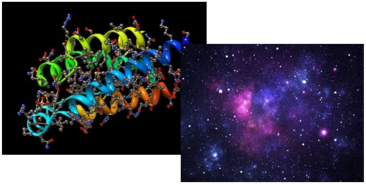
Gitta Kutyniok

(Ludwig-Maximilians-Universität München and University of Tromsø)

Arnold Sommerfeld School "Physics meets Artificial Intelligence"
LMU Munich, September 12 – 16, 2022

Neural Network Successes





LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

... and Failures

Deep neural networks have

- tremendous *success* for problems in scientific computing,
- but serious *downsides*.

... and Failures

Deep neural networks have

- tremendous *success* for problems in scientific computing,
- but serious *downsides*.

Problems/Limitations:

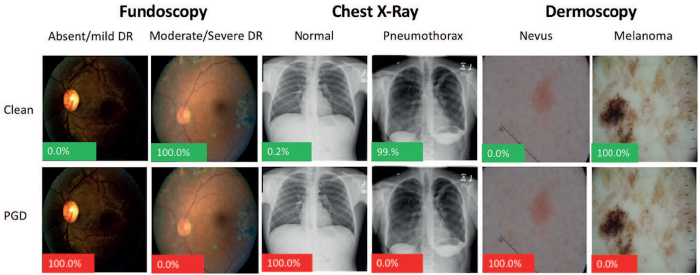
- Robustness
- Explainability
- Severe dependence on data
- Specific task
- Reasoning
- ....

Robustness

Requirements:

- Robustness problems should be immediately detectable or avoidable.
- Heuristic approaches do not satisfy certification standards.

Is complete robustness at all possible?



Source: Finlayson, Chung, Kohane, Beam, Adversarial Attacks Against Medical Deep Learning Systems, arXiv:1804.05296

Explainability

Requirements:

- It should be possible to ask any question about a decision.
- The answer should reason as a human.

Is this achievable by connecting deep learning to natural language processing?

Severe Dependence on Data

Amount of Data:

- Many applications do not have large amounts of training data.
- Methods such as data augmentation do not compensate this fully.

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Severe Dependence on Data

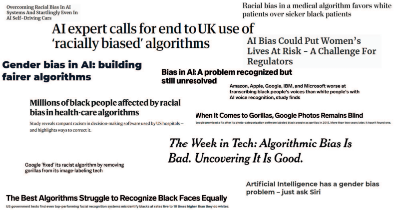
Amount of Data:

- Many applications do not have large amounts of training data.
- Methods such as data augmentation do not compensate this fully.

Problematic Data:

- Training data can unknowingly be biased.
- Uncertainty in the data can occur.

Can these problems be tackled at all?



LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Specific Task and Reasoning

Requirements:

- It should be possible to train for multiple tasks.
- The neural network should also be able to reason.
- Ideally, lifelong learning should be possible.

Are there fundamental limitations that constrain us?

Strong Requirements for Reliability

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

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Strong Requirements for Reliability

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

International Position of Europe and Germany in Reliable AI:

- AI Strategy of the German Federal Government
- AI Act of the European Union

Major Goal:

Introduce Certificates for AI Technology!

Strong Requirements for Reliability

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

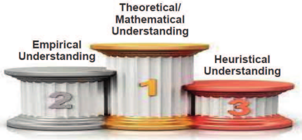
International Position of Europe and Germany in Reliable AI:

- AI Strategy of the German Federal Government
- AI Act of the European Union

Major Goal:

Introduce Certificates for AI Technology!

Types of Understanding:



Reliable AI

Key Requirements for Certificates:

- Bounds for generalization error
- Explainability approach (which is itself reliable)
- Understanding of fundamental problems

Can We Explain Network Decisions ... Reliably?

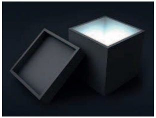
General Problem Setting

Question:

- Given a trained neural network.
- We don't know what the training data was nor how it was trained.

$\rightsquigarrow$ *Can we determine how it operates?*

Opening the Black Box!



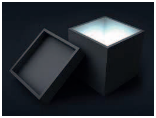
General Problem Setting

Question:

- Given a trained neural network.
- We don't know what the training data was nor how it was trained.

$\rightsquigarrow$ *Can we determine how it operates?*

Opening the Black Box!



Why is this important?

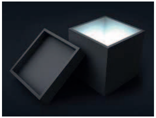
- Reasons for decisions required in various application settings.
- Scientists might get additional insights into their data.
- Trustworthiness can be improved.

General Problem Setting

Question:

- Given a trained neural network.
- We don't know what the training data was nor how it was trained.

⇝ *Can we determine how it operates?*



Opening the Black Box!

Why is this important?

- Reasons for decisions required in various application settings.
- Scientists might get additional insights into their data.
- Trustworthiness can be improved.

Vision for the Future:

- Human-like answer to any question about a decision!

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

History of the Field

Previous Relevance Mapping Methods:

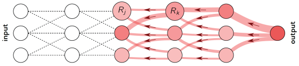
- Gradient based methods:
  - *Sensitivity Analysis* (Baehrens, Schroeter, Harmeling, Kawanabe, Hansen, Müller, 2010)
  - *SmoothGrad* (Smilkov, Thorat, Kim, Viégas, Wattenberg, 2017)

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

History of the Field

Previous Relevance Mapping Methods:

- Gradient based methods:
  - *Sensitivity Analysis* (Baehrens, Schroeter, Harmeling, Kawanabe, Hansen, Müller, 2010)
  - *SmoothGrad* (Smilkov, Thorat, Kim, Viégas, Wattenberg, 2017)
- Backwards propagation based methods:
  - *Guided Backprop* (Springenberg, Dosovitskiy, Brox, Riedmiller, 2015)
  - *Layer-wise Relevance Propagation* (Bach, Binder, Montavon, Klauschen, Müller, Samek, 2015)
  - *Deep Taylor* (Montavon, Samek, Müller, 2018)

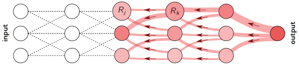


$$R_j = \sum_k \frac{a_j w_{jk}}{\sum_{0,j} a_j w_{jk}} R_k$$

History of the Field

Previous Relevance Mapping Methods:

- Gradient based methods:
  - *Sensitivity Analysis* (Baehrens, Schroeter, Harmeling, Kawanabe, Hansen, Müller, 2010)
  - *SmoothGrad* (Smilkov, Thorat, Kim, Viégas, Wattenberg, 2017)
- Backwards propagation based methods:
  - *Guided Backprop* (Springenberg, Dosovitskiy, Brox, Riedmiller, 2015)
  - *Layer-wise Relevance Propagation* (Bach, Binder, Montavon, Klauschen, Müller, Samek, 2015)
  - *Deep Taylor* (Montavon, Samek, Müller, 2018)



$$R_j = \sum_k \frac{a_j w_{jk}}{\sum_{0,j} a_j w_{jk}} R_k$$

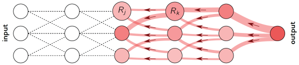
- Surrogate model based methods:
  - *LIME (Local Interpretable Model-agnostic Explanations)* (Ribeiro, Singh, Guestrin, 2016)

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

History of the Field

Previous Relevance Mapping Methods:

- Gradient based methods:
  - *Sensitivity Analysis* (Baehrens, Schroeter, Harmeling, Kawanabe, Hansen, Müller, 2010)
  - *SmoothGrad* (Smilkov, Thorat, Kim, Viégas, Wattenberg, 2017)
- Backwards propagation based methods:
  - *Guided Backprop* (Springenberg, Dosovitskiy, Brox, Riedmiller, 2015)
  - *Layer-wise Relevance Propagation* (Bach, Binder, Montavon, Klauschen, Müller, Samek, 2015)
  - *Deep Taylor* (Montavon, Samek, Müller, 2018)



$$R_j = \sum_k \frac{a_j w_{jk}}{\sum_{0,j} a_j w_{jk}} R_k$$

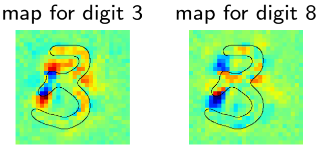
- Surrogate model based methods:
  - *LIME (Local Interpretable Model-agnostic Explanations)* (Ribeiro, Singh, Guestrin, 2016)
- Game theoretic methods:
  - *Shapley values* (Shapley, 1953), (Kononenko, Štrumbelj, 2010)
  - *SHAP (Shapley Additive Explanations)* (Lundberg, Lee, 2017)

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Towards a More Mathematical Understanding

What is Relevance?

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



Classification as a Classical Task for Neural Networks:

- Which features are most relevant for the decision?
  - Treat every pixel separately
  - Consider combinations of pixels
  - Incorporate additional knowledge
- How certain is the decision?

Tasks for Today

Challenges:

- What exactly is relevance in a mathematical sense?
- What is a good relevance map?
- How to compare different relevance maps?
- How to extend to challenging modalities?
- Can we also assign relevance to more complex features?

Tasks for Today

Challenges:

- What exactly is relevance in a mathematical sense?
  $\rightsquigarrow$ *Rigorous definition of relevance by information theory.*
- What is a good relevance map?
- How to compare different relevance maps?
- How to extend to challenging modalities?
- Can we also assign relevance to more complex features?

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Tasks for Today

Challenges:

- What exactly is relevance in a mathematical sense?
  $\rightsquigarrow$ *Rigorous definition of relevance by information theory.*
- What is a good relevance map?
  $\rightsquigarrow$ *Formulation of interpretability as optimization problem.*
- How to compare different relevance maps?
- How to extend to challenging modalities?
- Can we also assign relevance to more complex features?

Tasks for Today

Challenges:

- What exactly is relevance in a mathematical sense?
  $\rightsquigarrow$ *Rigorous definition of relevance by information theory.*
- What is a good relevance map?
  $\rightsquigarrow$ *Formulation of interpretability as optimization problem.*
- How to compare different relevance maps?
  $\rightsquigarrow$ *Canonical framework for comparison.*
- How to extend to challenging modalities?

- Can we also assign relevance to more complex features?

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Tasks for Today

Challenges:

- What exactly is relevance in a mathematical sense?
  $\rightsquigarrow$ *Rigorous definition of relevance by information theory.*
- What is a good relevance map?
  $\rightsquigarrow$ *Formulation of interpretability as optimization problem.*
- How to compare different relevance maps?
  $\rightsquigarrow$ *Canonical framework for comparison.*
- How to extend to challenging modalities?
  $\rightsquigarrow$ *Conceptually general and flexible interpretability approach.*
- Can we also assign relevance to more complex features?

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Tasks for Today

Challenges:

- What exactly is relevance in a mathematical sense?
  $\rightsquigarrow$ *Rigorous definition of relevance by information theory.*
- What is a good relevance map?
  $\rightsquigarrow$ *Formulation of interpretability as optimization problem.*
- How to compare different relevance maps?
  $\rightsquigarrow$ *Canonical framework for comparison.*
- How to extend to challenging modalities?
  $\rightsquigarrow$ *Conceptually general and flexible interpretability approach.*
- Can we also assign relevance to more complex features?
  $\rightsquigarrow$ *Take appropriate decompositions of the data into account.*

The Rate-Distortion Viewpoint

The Relevance Mapping Problem

The Setting: Let

- $\Phi \colon \mathbb{R}^n \to \mathbb{R}^m$ be a *classification function*,
- $x \in \mathbb{R}^n$ be an *input signal*.

$\xrightarrow{\Phi}$ $\Phi(x) = 0.97$ "Monkey"

$\xrightarrow{\Phi}$ $\Phi(x) = 0.07$ "Not a monkey"

The Relevance Mapping Problem

The Task:

- Determine the *most relevant components of x* for the prediction $\Phi(x)$.
- Choose $S \subseteq \{1, \ldots, n\}$ of components that are considered *relevant*.
- S should be small (usually not everything is relevant).
- S^c is considered *non-relevant*.

Original image x

Relevant components S

Non-relevant components S^c

Rate-Distortion Viewpoint

$\Phi(x) = 0.97$

"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!*

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

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!*

Theorem (Wäldchen, Macdonald, Hauch, Kutyniok, 2021):
Given Φ, x, $k \in \{1, \ldots, d\}$, and $\epsilon < \frac{1}{4}$. Deciding whether $R(\epsilon) \leq k$ is $\mathsf{NP}^{\mathsf{PP}}$-complete.

Finding a minimizer of $R(\epsilon)$ is hard!

Rate-Distortion Explanation

Rate-Distortion Function:

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

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

Theorem (Wäldchen, Macdonald, Hauch, Kutyniok, 2021):
Given Φ, x, $k \in \{1, \ldots, d\}$, and $\epsilon < \frac{1}{4}$. Deciding whether $R(\epsilon) \le k$ is $\mathrm{NP}^{\mathrm{PP}}$-complete.

Finding a minimizer of $R(\epsilon)$ is hard!

Theorem (Wäldchen, Macdonald, Hauch, Kutyniok, 2021):
Given Φ, x, and $\alpha \in (0, 1)$. Approximating $R(\epsilon)$ to within a factor of $d^{1-\alpha}$ is NP-hard.

Even the approximation problem of it is hard!

LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

RDE (Macdonald, Wäldchen, Hauch, Kutyniok, 2020)

Problem Relaxation:

	Discrete problem	Continuous problem
Relevant set	$S \subseteq \{1, \dots, d\}$	
Obfuscation	$y_S = x_S$, $y_{S^c} = n_{S^c}$	
Distortion	$D(S)$	
Rate/Size	$\lvert S \rvert$	

Problem Relaxation:

	Discrete problem	Continuous problem
Relevant set	$S \subseteq \{1, \ldots, d\}$	$s \in [0,1]^d$
Obfuscation	$y_S = x_S,\ y_{S^c} = n_{S^c}$	$y = s \odot x + (1-s) \odot n$
Distortion	$D(S)$	$D(s)$
Rate/Size	$\lvert S \rvert$	$\lVert s \rVert_1$

RDE (Macdonald, Wäldchen, Hauch, Kutyniok, 2020)

Problem Relaxation:

	Discrete problem	Continuous problem
Relevant set	$S \subseteq \{1, \dots, d\}$	$s \in [0,1]^d$
Obfuscation	$y_S = x_S,\ y_{S^c} = n_{S^c}$	$y = s \odot x + (1-s) \odot n$
Distortion	$D(S)$	$D(s)$
Rate/Size	$\lvert S \rvert$	$\lVert s \rVert_1$

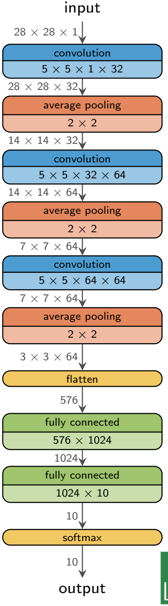
Resulting Minimization Problem:

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

MNIST Experiment

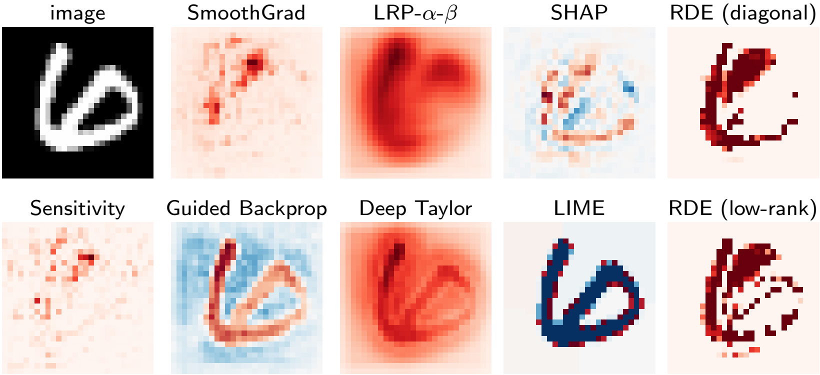
Data Set	
Image size	$28 \times 28 \times 1$
Number of classes	10
Training samples	50000

Test accuracy: 99.1%



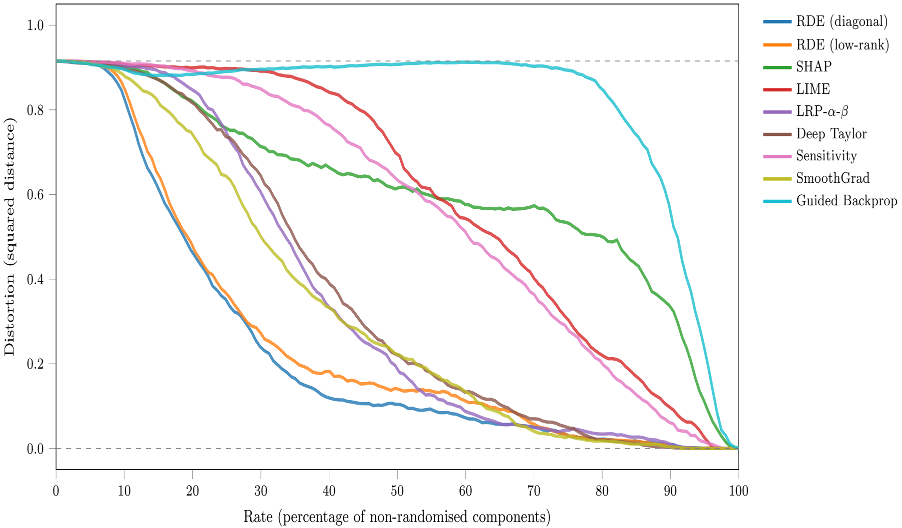
MNIST dataset of handwritten digits (LeCun, Cortes, 1998)

MNIST 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)

MNIST Experiment



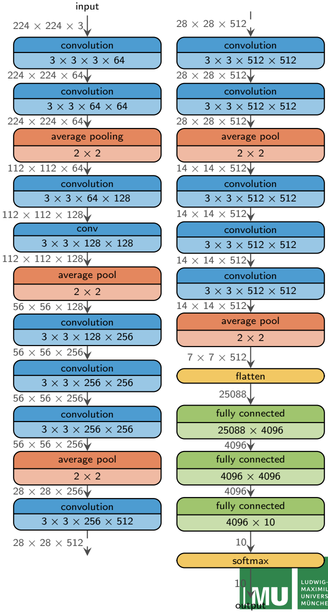
SmoothGrad (Smilkov, Thorat, Kim, Viégas, Wattenberg, 2017), Layer-wise Relevance Propagation (Bach, Binder, Montavon, Klauschen, Müller, Samek, 2015), SHAP (Lundberg, Lee, 2017), Sensitivity Analysis (Simonyan, Vedaldi, Zisserman, 2013), Guided Backprop (Springenberg, Dosovitskiy, Brox, Riedmiller,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

STL-10 Experiment

Data Set	
Image size	$96 \times 96 \times 3$
	$(224 \times 224 \times 3)$
Number of classes	10
Training samples	4000

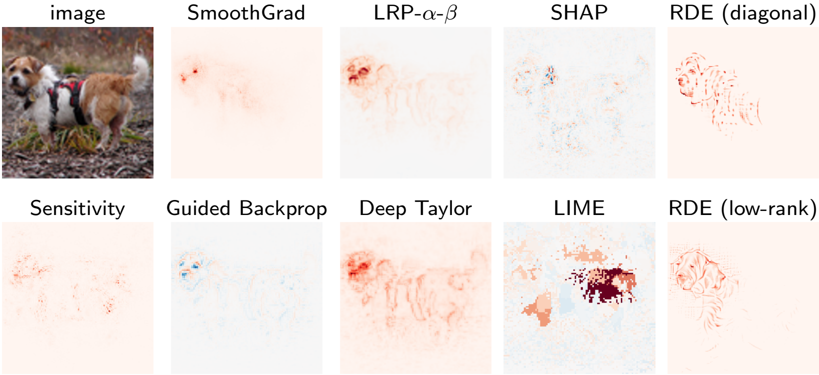
Test accuracy: 93.5%

(VGG-16 convolutions pretrained on Imagenet)



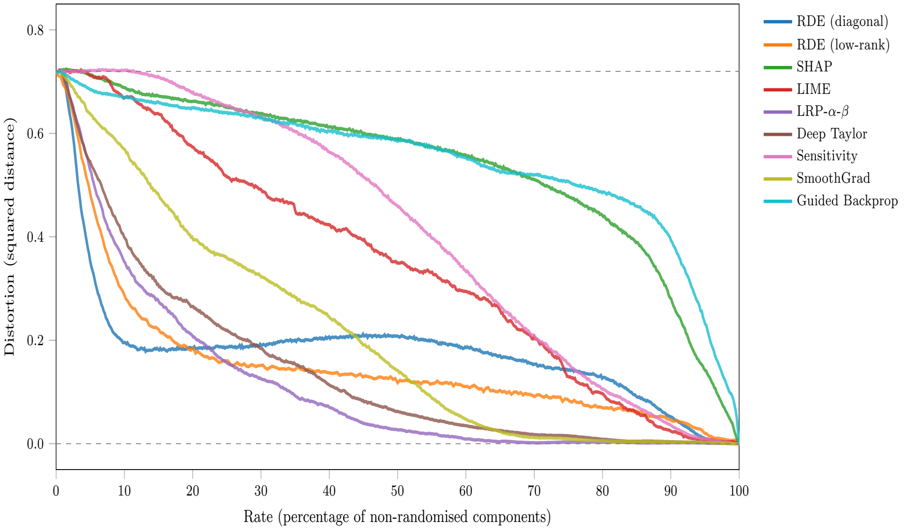
STL-10 dataset (Coates, Lee, Ng, 2011), VGG-16 network (Simonyan, Zisserman, 2014)

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)

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)

Going Further...

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?*

Goal:

- *Take the conditional data distribution into account!*
- *Ensure that specifics of various modalities can be handled!*

General Approach

Optimization Problem:

We consider the following minimization problem:

$$\min_{s \in \{0,1\}^d} \mathbb{E}_{y \sim \Upsilon_s} \left[\frac{1}{2} (\Phi(x) - \Phi(y))^2 \right] + \lambda \|s\|_1,$$

where y is generated by a trained inpainting network G as

$$y := x \odot s + G(x, s, n) \odot (1 - s).$$

General Approach

Optimization Problem:
We consider the following minimization problem:

$$\min_{s\in\{0,1\}^d} \mathbb{E}_{y\sim\Upsilon_s}\left[\frac{1}{2}(\Phi(x)-\Phi(y))^2\right]+\lambda\|s\|_1,$$

where y is generated by a trained inpainting network G as

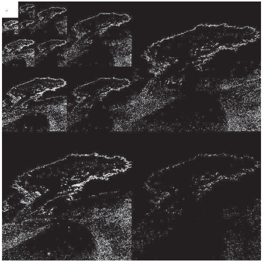
$$y := x \odot s + G(x,s,n) \odot (1-s).$$

Requirements of Different Modalities: Can be applied ...

- ... to images, but also audio data, etc.
- ... after a transform (e.g. *wavelets*) to allow more complex features.

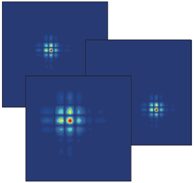
Conceptually general and flexible interpretability approach!

The World is Compressible!



Wavelet Transform (JPEG2000):

$$f \mapsto (\langle f, \psi_{j,m} \rangle)_{j,m}.$$

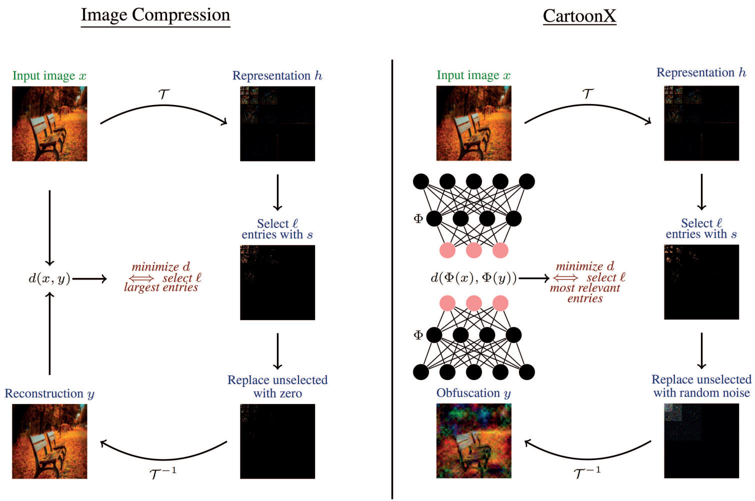


Definition: For a wavelet $\psi \in L^2(\mathbb{R}^2)$, a *wavelet system* is defined by

$$\{\psi_{j,m} : j \in \mathbb{Z}, m \in \mathbb{Z}^2\}, \quad \text{where } \psi_{j,m}(x) := 2^j \psi(2^j x - m).$$

LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

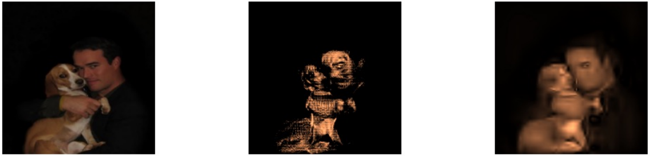
Cartoon X (Kolek, Nguyen, Levie, Bruna, Kutyniok; 2022)



Detecting Reason for Adversarial Examples

CartoonX:

Baby



Screw

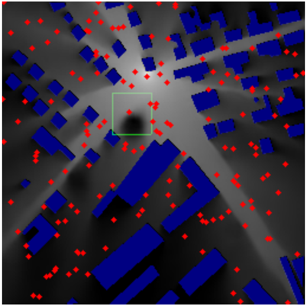
Numerical Experiments:

Other Types of Data

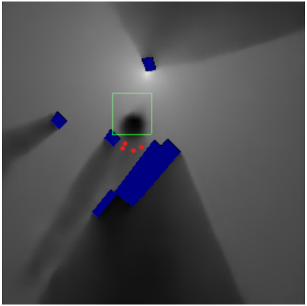
NSynth Dataset:

Instrument	Magnitude Importance	Phase Importance
Organ	0.829	1.0
Guitar	0.0	0.999
Flute	0.092	1.0
Bass	1.0	1.0
Reed	0.136	1.0
Vocal	1.0	1.0
Mallet	0.005	0.217
Brass	0.999	1.0
Keyboard	0.003	1.0
String	1.0	0.0

RadioUNet (Levie, Cagkan, Kutyniok, Caire; 2020):



Estimated map



Explanation

Deep Neural Networks are Not a Swiss Army Knife!

They do have Limitations!

Computability

Theory asserts

- the expressibility of the class of deep neural networks
- convergence of training algorithms
- generalization abilities
- ...

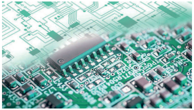
Computability

Theory asserts

- the expressibility of the class of deep neural networks
- convergence of training algorithms
- generalization abilities
- ...

Theory does not sufficiently consider

- practical performance when trained by modern approaches
- required sample complexity
- limits of **computability** on today's hardware



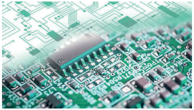
Computability

Theory asserts

- the expressibility of the class of deep neural networks
- convergence of training algorithms
- generalization abilities
- ...

Theory does not sufficiently consider

- practical performance when trained by modern approaches
- required sample complexity
- limits of **computability** on today's hardware



Theory-to-Practice Gap!

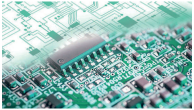
Computability

Theory asserts

- the expressibility of the class of deep neural networks
- convergence of training algorithms
- generalization abilities
- ...

Theory does not sufficiently consider

- practical performance when trained by modern approaches
- required sample complexity
- limits of **computability** on today's hardware



Theory-to-Practice Gap!

Goal: Examine the boundaries imposed by digital computations!

What can actually be computed?

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.

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

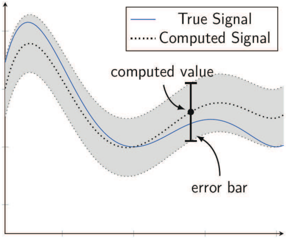
What can actually be computed?

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.

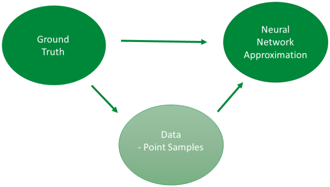
Questions:

- Is the underlying problem feasible?
  → *Computability of the ground truth*
- Are the neural networks computable?
  → *Computability of the network*
- Can the neural networks be found with the minimization problem?
  → *Computability of the mapping from data to approximation*



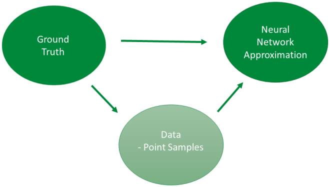
Why does Computability Matter?

- What is the *best* we can hope for?
  - Non-computability of the *ground truth* $\rightarrow$ *No approximation scheme*
  - Non-computability of the *network* $\rightarrow$ *Despite existence, network may not be computable*
  - Non-computability of the *mapping* from data to approximation $\rightarrow$ *Learning not feasible*



Why does Computability Matter?

- What is the *best* we can hope for?
  - Non-computability of the *ground truth* $\rightarrow$ *No approximation scheme*
  - Non-computability of the *network* $\rightarrow$ *Despite existence, network may not be computable*
  - Non-computability of the *mapping* from data to approximation $\rightarrow$ *Learning not feasible*

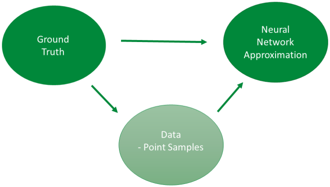


- Can we *trust* the output of a computation?
  - Computability guarantees prescribed error bounds $\rightarrow$ Reliable output

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Why does Computability Matter?

- What is the *best* we can hope for?
  - Non-computability of the *ground truth*
    $\rightarrow$ *No approximation scheme*
  - Non-computability of the *network*
    $\rightarrow$ *Despite existence, network may not be computable*
  - Non-computability of the *mapping* from data to approximation
    $\rightarrow$ *Learning not feasible*

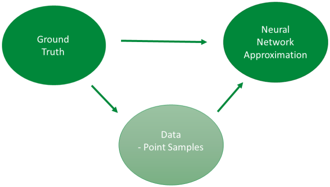


- Can we *trust* the output of a computation?
  - Computability guarantees prescribed error bounds
    $\rightarrow$ Reliable output

Non-computable problems can be tackled successfully in practice, if limited precision suffices!

Why does Computability Matter?

- What is the *best* we can hope for?
  - Non-computability of the *ground truth* $\rightarrow$ *No approximation scheme*
  - Non-computability of the *network* $\rightarrow$ *Despite existence, network may not be computable*
  - Non-computability of the *mapping* from data to approximation $\rightarrow$ *Learning not feasible*



- Can we *trust* the output of a computation?
  - Computability guarantees prescribed error bounds $\rightarrow$ Reliable output

Non-computable problems can be tackled successfully in practice, if limited precision suffices!

But we have no guarantees of correctness!

Specific Example

Consider the modeling of a physical system S on a digital computer.

- Assume a mathematical model S_{mod} for S describes the physical process and allows to predict the output of S for any given input.

How well does S_{mod} describe the real physical process S?

Specific Example

Consider the modeling of a physical system S on a digital computer.

- Assume a mathematical model S_{mod} for S describes the physical process and allows to predict the output of S for any given input.

How well does S_{mod} describe the real physical process S?

- Compute the corresponding output $y = Sx$ for several input signals x.
- Compare these measurements with the theoretical prediction $y_{\text{pred}} = S_{\text{mod}}x$.

However, usually no closed-form solution for the output y_{pred} exists!

Specific Example

Consider the modeling of a physical system S on a digital computer.

- Assume a mathematical model S_{mod} for S describes the physical process and allows to predict the output of S for any given input.

How well does S_{mod} describe the real physical process S?

- Compute the corresponding output $y = Sx$ for several input signals x.
- Compare these measurements with the theoretical prediction $y_{\text{pred}} = S_{\text{mod}}x$.

However, usually no closed-form solution for the output y_{pred} exists!

- Use a computer to determine y_{pred} of the model S_{mod} for an input x.
- A digital computer can only compute an approximation $\tilde{y}_{\text{pred}}$ of y_{pred}.

Control $\|\tilde{y}_{pred} - y_{pred}\|$ algorithmically on the computer!

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Specific Example

Consider the modeling of a physical system S on a digital computer.

- Assume a mathematical model S_{mod} for S describes the physical process and allows to predict the output of S for any given input.

How well does S_{mod} describe the real physical process S?

- Compute the corresponding output $y = Sx$ for several input signals x.
- Compare these measurements with the theoretical prediction $y_{\text{pred}} = S_{\text{mod}}x$.

However, usually no closed-form solution for the output y_{pred} exists!

- Use a computer to determine y_{pred} of the model S_{mod} for an input x.
- A digital computer can only compute an approximation $\tilde{y}_{\text{pred}}$ of y_{pred}.

Control $\|\tilde{y}_{pred} - y_{pred}\|$ algorithmically on the computer!

Otherwise...

- ...the calculated solution $\tilde{y}_{\text{pred}}$ might be far from y_{pred} and comparing measurements of S with $\tilde{y}_{\text{pred}}$ becomes meaningless!
- ...no information about the quality of the mathematical model!

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Theory of Computation

Turing machine $\longleftrightarrow$ Abstract idealization of *digital computer*

$\downarrow$ $\downarrow$

Discrete problems Scientific computing: *Continuous problems*

Theory of Computation

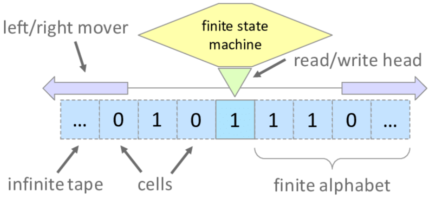
Turing machine $\longleftrightarrow$ Abstract idealization of *digital computer*

$\downarrow$ $\downarrow$

Discrete problems Scientific computing: *Continuous problems*

Definition:

"A Turing machine is a mathematical model of computation that defines an abstract machine that manipulates symbols on a strip of tape according to a table of rules."



Definition:
A *computable real number* r is one for which there is a Turing machine with the following property: Given $n \in \mathbb{N}$ on its initial tape, it terminates with a rational number q such that $|r - q| \leq 2^{-n}$.

Computability

Definition:
A *computable real number* r is one for which there is a Turing machine with the following property: Given $n \in \mathbb{N}$ on its initial tape, it terminates with a rational number q such that $|r - q| \leq 2^{-n}$.

Definition:
A function $f : \mathbb{R} \to \mathbb{R}$ is *computable*, if there exists an algorithm (Turing machine) Γ_f, which gives for all computable $x \in \mathbb{R}_c$ and all $n \in \mathbb{N}$ an approximation to $f(x)$ with

$$|\Gamma_f(x, n) - f(x)| \leq 2^{-n}.$$

A Large Problem Class

Inverse Problem in Imaging

Recall:

Given $A \in \mathbb{C}^{m \times N}$ and $y = Ax + e \in \mathbb{C}^m$ of $x \in \mathbb{C}^N$, recover x.

Properties:

- $A \in \mathbb{C}^{m \times N}$ sampling operator, typically $m < N$ or even $m \ll N$
- successful approaches:
  - Sparse regularization techniques
  - Deep learning techniques or hybrid approaches

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Inverse Problem in Imaging

Recall:

Given $A \in \mathbb{C}^{m \times N}$ and $y = Ax + e \in \mathbb{C}^m$ of $x \in \mathbb{C}^N$, recover x.

Properties:

- $A \in \mathbb{C}^{m \times N}$ sampling operator, typically $m < N$ or even $m \ll N$
- successful approaches:
  - Sparse regularization techniques
  - Deep learning techniques or hybrid approaches

Optimization Problem:

Given $A \in \mathbb{C}^{m \times N}$ and measurements $y \in \mathbb{C}^m$, solve

$$\arg\min_{x \in \mathbb{C}^N} \|x\|_{\ell^1} \text{ such that } \|Ax - y\|_{\ell^2} \leq \varepsilon, \quad \varepsilon > 0.$$

Solution Set

Solution Set:
For $A \in \mathbb{C}^{m \times N}$ and $y \in \mathbb{C}^m$ let

$$\Psi(A, y) := \arg\min_{x \in \mathbb{C}^N} \|x\|_{\ell^1} \text{ such that } \|Ax - y\|_{\ell^2} \leq \varepsilon.$$

Solution Set

Solution Set:
For $A \in \mathbb{C}^{m \times N}$ and $y \in \mathbb{C}^m$ let

$$\Psi(A, y) := \arg\min_{x \in \mathbb{C}^N} \|x\|_{\ell^1} \text{ such that } \|Ax - y\|_{\ell^2} \leq \varepsilon.$$

Fundamental Questions:

What can actually be computed on digital hardware?

Solution Set

Solution Set:
For $A \in \mathbb{C}^{m \times N}$ and $y \in \mathbb{C}^m$ let

$$\Psi(A, y) := \arg\min_{x \in \mathbb{C}^N} \|x\|_{\ell^1} \text{ such that } \|Ax - y\|_{\ell^2} \leq \varepsilon.$$

Fundamental Questions:

What can actually be computed on digital hardware?

What are inherent restrictions of deep learning (performed on digital hardware)?

Solution Set

Solution Set:
For $A \in \mathbb{C}^{m \times N}$ and $y \in \mathbb{C}^m$ let

$$\Psi(A, y) := \arg\min_{x \in \mathbb{C}^N} \|x\|_{\ell^1} \text{ such that } \|Ax - y\|_{\ell^2} \leq \varepsilon.$$

Fundamental Questions:

What can actually be computed on digital hardware?

What are inherent restrictions of deep learning (performed on digital hardware)?

Are we *missing the correct tools and algorithms* to train neural networks adequately on digital machines or do *such algorithms not exist at all*?

A Bit Disappointing News

Non-Computability of Finite Dimensional Inverse Problems

Solution Set:

For $A \in \mathbb{C}^{m \times N}$ and $y \in \mathbb{C}^m$ let

$$\Psi(A, y) := \arg\min_{x \in \mathbb{C}^N} \|x\|_{\ell^1} \text{ such that } \|Ax - y\|_{\ell^2} \leq \varepsilon.$$

Non-Computability of Finite Dimensional Inverse Problems

Solution Set:

For $A \in \mathbb{C}^{m \times N}$ and $y \in \mathbb{C}^m$ let

$$\Psi(A, y) := \arg\min_{x \in \mathbb{C}^N} \|x\|_{\ell^1} \text{ such that } \|Ax - y\|_{\ell^2} \leq \varepsilon.$$

Theorem (Boche, Fono, Kutyniok; 2022):

The function $\Psi : \mathbb{C}^{m \times N} \times \mathbb{C}^m \to \mathbb{C}^N$ for fixed parameters $\epsilon \in (0, 1)$, $N \geq 2$, and $m < N$, is *not computable*.

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Non-Computability of Finite Dimensional Inverse Problems

Solution Set:

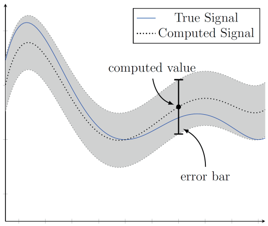
For $A \in \mathbb{C}^{m \times N}$ and $y \in \mathbb{C}^m$ let

$$\Psi(A, y) := \arg\min_{x \in \mathbb{C}^N} \|x\|_{\ell^1} \text{ such that } \|Ax - y\|_{\ell^2} \leq \varepsilon.$$

Theorem (Boche, Fono, Kutyniok; 2022):

The function $\Psi : \mathbb{C}^{m \times N} \times \mathbb{C}^m \to \mathbb{C}^N$ for fixed parameters $\epsilon \in (0, 1)$, $N \geq 2$, and $m < N$, is *not computable*.

Illustration of the Problem:



Some Thoughts on the Result

Corollary:

- *No algorithm exists*, which on digital hardware derives neural networks Φ_A approximating $\Psi(A, \cdot)$ for any given accuracy and all $A \in \mathbb{C}^{m \times N}$.
- The output of trained neural networks is *not reliable (no guarantees)*.
- This result could point towards why *instabilities* and *non-robustness* occurs for deep neural networks.

Some Thoughts on the Result

Corollary:

- *No algorithm exists*, which on digital hardware derives neural networks Φ_A approximating $\Psi(A, \cdot)$ for any given accuracy and all $A \in \mathbb{C}^{m \times N}$.
- The output of trained neural networks is *not reliable (no guarantees)*.
- This result could point towards why *instabilities* and *non-robustness* occurs for deep neural networks.

General Barrier:

This barrier on the capabilities of neural networks for finite-dimensional inverse problems is caused by *a combination* of the following two separate aspects:

- The mathematical structure and properties of *finite-dimensional inverse problems*.
- The mathematical structure and properties of *Turing machines* and thereby also of *digital machines*.

What now?

What now?

New Emerging Hardware:

- *Neuromorphic computing*: Elements of computer modeled after systems in the human brain and nervous system.
- *Biocomputing*: Living cells as the substrate for performing human-defined computations
- *Quantum computing*: Computing units are typically quantum circuits

What now?

New Emerging Hardware:

- *Neuromorphic computing*: Elements of computer modeled after systems in the human brain and nervous system.
- *Biocomputing*: Living cells as the substrate for performing human-defined computations
- *Quantum computing*: Computing units are typically quantum circuits

Key Future Question:

Does the non-computability result also hold for different computation models such as analog computers as well?

Theorem (Boche, Fono, Kutyniok; 2022):
The function $\Psi : \mathbb{C}^{m\times N} \times \mathbb{C}^m \to \mathbb{C}^N$ for fixed parameters $\epsilon \in (0,1)$, $N \geq 2$, and $m < N$, is *computable on a Blum-Shub-Smale machine*.

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Some Final Thoughts...

Mathematics for Deep Learning

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

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

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

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

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

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

THANK YOU!

References available at:

`www.ai.math.lmu.de/kutyniok`

Survey Paper (arXiv:2105.04026):

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

Check related information on Twitter at:

`@GittaKutyniok`

Upcoming Book:

- P. Grohs and G. Kutyniok, eds.
  Mathematical Aspects of Deep Learning
  Cambridge University Press, to appear.