Outline

• general remarks on ML, also on explaining and interpreting
• understanding single decisions of nonlinear learners
• Layer-wise Relevance Propagation (LRP)
• Applications in Neuroscience and Physics
ML in a nutshell

Standard ML

- data
- ML model
- predictions

Generalization error

Kernel Methods: SVM etc.

\[
W(\alpha) = \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{\ell} \alpha_i \alpha_j y_i y_j k(x_i, x_j)
\]

subject to \(0 \leq \alpha_i \leq C, \ i = 1, \ldots, \ell, \) and \(\sum_{i=1}^{\ell} \alpha_i y_i = 0,\)

Deep Neural Networks

input image

hidden layer

hidden layer

hidden layer

class 1 (boat)

class 2 (truck)

class 3 (car)
Tutorial on Methods for Interpreting and Understanding Deep Neural Networks

Wojciech Samek (Fraunhofer HHI)
Grégoire Montavon (TU Berlin)
Klaus-Robert Müller (TU Berlin)
Acknowledgements

We thank our collaborators!

Alexander Binder (SUTD)  Sebastian Lapuschkin (Fraunhofer HHI)

Lecture notes will be online soon at:
http://www.heatmapping.org
Recent ML systems reach superhuman performance

- AlphaGo beats Go human champ
- Deep Net outperforms humans in image classification
- Autonomous search-and-rescue drones outperform humans
- DeepStack beats professional poker players
- Computer out-plays humans in "doom"
- IBM's Watson destroys humans in jeopardy
- Deep Net beats human at recognizing traffic signs

ML in the sciences
From Data to Information

Huge volumes of data

Computing power

Deep Nets / Kernel Machines / ...

Interpretable Information

Solve task

Information (implicit)
From Data to Information

Performace vs Interpretability

AlexNet (16.4%)  Clarifai (11.1%)  VGG (7.3%)  GoogleNet (6.7%)  ResNet (3.57%)

Data → Information → Interpretable for human

Crucial in many applications (industry, sciences...)

ImageNet
Interpretable vs. powerful models?

Linear model

- Poor fit, but easily interpretable
- “global explanation”

Non-linear model

- Can be very complex
- “individual explanation”
Interpretable vs. powerful models?!

60 million parameters
650,000 neurons

We have techniques to interpret and explain such complex models!

Kernel machines
Interpretable vs. powerful models?

train best model $\rightarrow$ interpret it

vs.

train interpretable model

suboptimal or biased due to assumptions (linearity, sparsity ...)

TU Berlin
Different dimensions of interpretability

- **prediction**
  “Explain why a certain pattern $x$ has been classified in a certain way $f(x)$.”

- **model**
  “What would a pattern belonging to a certain category typically look like according to the model.”

- **data**
  “Which dimensions of the data are most relevant for the task.”
Why interpretability?

1) Verify that classifier works as expected

Wrong decisions can be costly and dangerous

“Autonomous car crashes, because it wrongly recognizes ...”

“AI medical diagnosis system misclassifies patient’s disease ...”
Why interpretability?

2) Improve classifier

**Standard ML**
- Data
- ML model
- Predictions
- Generalization error

**Interpretable ML**
- Data
- ML model
- Interpretability
- Human inspection
- Verified predictions
- Generalization error + human experience

model/data improvement
Why interpretability?

3) Learn from the learning machine

"It's not a human move. I've never seen a human play this move." (Fan Hui)

Old promise:
"Learn about the human brain."
Why interpretability?

4) Interpretability in the sciences

Learn about the physical / biological / chemical mechanisms.
(e.g. find genes linked to cancer, identify binding sites ...)

[Graph and molecular structures]
Why interpretability?

5) Compliance to legislation

European Union’s new General Data Protection Regulation → “right to explanation”

Retain human decision in order to assign responsibility.

“Wealth interpretability we can ensure that ML models work in compliance to proposed legislation.”
Why interpretability?

Interpretability as a gateway between ML and society
- Make complex models acceptable for certain applications.
- Retain human decision in order to assign responsibility.
- “Right to explanation”

Interpretability as powerful engineering tool
- Optimize models / architectures
- Detect flaws / biases in the data
- Gain new insights about the problem
- Make sure that ML models behave “correctly”
Techniques of Interpretation

DNN transparency

interpreting models

- activation maximization
  - Berkes 2006
  - Erhan 2010
  - Simonyan 2013
  - Nguyen 2015/16

- data generation
  - Hinton 2006
  - Goodfellow 2014
  - v. den Oord 2016
  - Nguyen 2016

explaining decisions

- sensitivity analysis
  - Khan 2001
  - Gevrey 2003
  - Baehrens 2010
  - Simonyan 2013

- decomposition
  - Poulin 2006
  - Landecker 2013
  - Bach 2015
  - Montavon 2017

focus on model  focus on data
Techniques of Interpretation

Interpreting models (ensemble)
- find prototypical example of a category
- find pattern maximizing activity of a neuron

Explaining decisions (individual)
- “why” does the model arrive at this particular prediction
- verify that model behaves as expected

better understand internal representation

crucial for many practical applications
Techniques of Interpretation

In medical context

• Population view (ensemble)
  • Which symptoms are most common for the disease
  • Which drugs are most helpful for patients

• Patient’s view (individual)
  • Which particular symptoms does the patient have
  • Which drugs does he need to take in order to recover

Both aspects can be important depending on who you are (FDA, doctor, patient).
Techniques of Interpretation

Interpreting models

- find prototypical example of a category
- find pattern maximizing activity of a neuron

\[
\max_{x \in \mathcal{X}} p_\theta(\omega_c \mid x) + \lambda \Omega(x)
\]
Techniques of Interpretation

Explaining decisions

- "why" does the model arrive at a certain prediction
- verify that model behaves as expected

data → ML blackbox → decision

it's a shark
Techniques of Interpretation

Explaining decisions
- “why” does the model arrive at a certain prediction
- verify that model behaves as expected

ML blackbox

<table>
<thead>
<tr>
<th>decision</th>
</tr>
</thead>
<tbody>
<tr>
<td>it's a shark</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensitivity Analysis</td>
</tr>
<tr>
<td>Layer-wise Relevance Propagation (LRP)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>data</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>reasoning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f(x)$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>relevance map</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R(x)$</td>
</tr>
</tbody>
</table>
Techniques of Interpretation

Sensitivity Analysis
(Simonyan et al. 2014)

Explain prediction
(which pixels lead to decrease of prediction score when changed)

\[ \left\| \frac{\partial}{\partial x_p} f(x) \right\| \]
Techniques of Interpretation

Layer-wise Relevance Propagation (LRP) (Bach et al. 2015)

Idea: Decompose function
\[ \sum_i R_i = f(x) \]

"every neuron gets its share of relevance depending on activation and strength of connection."

Theoretical interpretation
Deep Taylor Decomposition (Montavon et al., 2017)
Interpreting models
Interpreting with class prototypes

Goal: Understand what the model considers as a “typical car”.

Activation maximization framework:

$$\max_{x \in \mathcal{X}} \log p(\omega_c | x) - \lambda \|x\|^2$$

class prototype
Examples of Class Prototypes

Activation maximization: \[
\max_{x \in \mathcal{X}} \log p(\omega_c \mid x) - \lambda \| x \|^2
\]

- dumbbell
- cup
- dalmatian

Symonian’13: Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps
Building more natural prototypes

\[ \max_{x \in \mathcal{X}} \log p(\omega_c \mid x) - \lambda \| x \|^2 \]

---

find input pattern that produces the strongest class activation

replace l2-norm by a data-dependent regularizer

\[ \max_{x \in \mathcal{X}} \log p(\omega_c \mid x) + \log p(x) \]

---

apply the Bayes theorem

\[ \max_{x \in \mathcal{X}} \log p(x \mid \omega_c) \]

---

find most likely input pattern for a given class

**Insight:** a good prototype must also depend on the data distribution.

Montavon, Samek, Müller arxiv 2017
Building Prototypes using a generator

Activation maximization in code space:

$$\max_{z \in Z} \log p(\omega_c | g(z)) - \lambda \|z\|^2$$
Building Prototypes using a generator

Nguyen’16: Synthesizing the preferred inputs for neurons in neural networks via deep generator networks.
**Types of Interpretation**

**Global Interpretation:**
Understanding how a lamp typically looks like.

![model's prototypical lamp](image)

Nguyen’16: Synthesizing the preferred inputs for neurons in neural networks via deep generator networks

**Local Interpretation:**
Understanding why *this* image of a lamp contains a lamp.

![some image of a lamp](image)

Approaches to interpretability

**Ante-hoc interpretability:**
Choose a model that is readily interpretable and train it.

*Example:*

\[ f(x) = \sum_{i=1}^{d} g_i(x_i) \]

Is the model expressive enough to predict the data?

**Post-hoc interpretability:**
Choose a model that works well in practice, and develop a special technique to interpret it.

*Example:*

\[ f(x) = \text{DeepNet}(x) \]

How to determine the contribution each input variable?
Explaining models
Explaining Neural Network Predictions

Layer-wise relevance Propagation (LRP, Bach et al 15) first method to explain nonlinear classifiers - based on generic theory (related to Taylor decomposition – deep taylor decomposition M et al 16) - applicable to any NN with monotonous activation, BoW models, Fisher Vectors, SVMs etc.

Explanation: “Which pixels contribute how much to the classification” (Bach et al 2015) (what makes this image to be classified as a car)

\[ f(x) = \sum_p h_p \]

Sensitivity / Saliency: “Which pixels lead to increase/decrease of prediction score when changed” (what makes this image to be classified more/less as a car) (Baehrens et al 10, Simonyan et al 14)

\[ h_p = \left\| \frac{\partial}{\partial x_p} f(x) \right\|_\infty \]

Cf. Deconvolution: “Matching input pattern for the classified object in the image” (Zeiler & Fergus 2014), (relation to f(x) not specified) Activation Maximization

Each method solves a different problem!!!
Explaining Neural Network Predictions

Classification

$$x_j = \sigma \left( \sum_i x_i w_{ij} + b_j \right)$$
Explaining Neural Network Predictions

Explanation

Initialization

\[ r_j = f(x) \]
Explaining Neural Network Predictions

Theoretical interpretation
Deep Taylor Decomposition

\[ r_i = x_i \sum_j \frac{w_{ij} r_j}{\sum_i x_i w_{ij}} = x_i C_i \]

\( r_i \) depends on the activations and the weights
Explaining Neural Network Predictions

Relevance Conservation Property

\[ \sum_p r_p = \ldots = \sum_i r_i = \sum_j r_j = \ldots = f(x) \]
Advantages of LRP over Sensitivity

1. Global explanations: What makes a car a car and not what makes a car less / more a car.

2. No discontinuities: small variations do not result in large changes of the relevance.
Advantages of LRP over both Sensitivity and Deconvolution

Image specific explanations: LRP takes into account the activations.

LRP provides different explanations for different input images.

For NNs without pooling layers Sensitvity and Deconvolution provides the same explanations for different samples.
Advantages of LRP over both Sensitivity and Deconvolution

Positive and Negative Evidence: LRP distinguishes between positive evidence, supporting the classification decision, and negative evidence, speaking against the prediction.

LRP indicates what speaks for class ‘3’ and speaks against class ‘9’.

The sign of Sensitivity and Deconvolution does not have this interpretation.

-> taking norm gives unsigned visualizations.
Male or Female?

http://interpretable-ml.org
Advantages of LRP over both Sensitivity and Deconvolution

Aggregation of Relevance: LRP explanations are normalized (conservation of relevance). This allows to meaningfully aggregate relevance over datasets or regions in an image.
Explaining Neural Network Predictions

<table>
<thead>
<tr>
<th></th>
<th>Sensitivity</th>
<th>Deconvolution</th>
<th>LRP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bird</td>
<td><img src="image1" alt="Bird Image" /></td>
<td><img src="image2" alt="Deconvolution Image" /></td>
<td><img src="image3" alt="LRP Image" /></td>
</tr>
<tr>
<td>Dog</td>
<td><img src="image4" alt="Dog Image" /></td>
<td><img src="image5" alt="Deconvolution Image" /></td>
<td><img src="image6" alt="LRP Image" /></td>
</tr>
</tbody>
</table>
### Application: understanding different DNN Architectures

GoogleNet focuses on the animal faces and only few pixels.

BVLC CaffeNet is less sparse.
Explaining Predictions Pixel-wise

Neural networks

Kernel methods
Understanding learning models for complex gaming scenarios
Analysing Breakout: LRP vs. Sensitivity

LRP  sensitivity

TU Berlin
Perspectives
Is the Generalization Error all we need?
Application: Comparing Classifiers

Test error for various classes:

<table>
<thead>
<tr>
<th></th>
<th>aeroplane</th>
<th>bicycle</th>
<th>bird</th>
<th>boat</th>
<th>bottle</th>
<th>bus</th>
<th>car</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fisher</td>
<td>79.08%</td>
<td>66.44%</td>
<td>45.90%</td>
<td>70.88%</td>
<td>27.64%</td>
<td>69.67%</td>
<td>80.96%</td>
</tr>
<tr>
<td>DeepNet</td>
<td>88.08%</td>
<td>79.69%</td>
<td>80.77%</td>
<td>77.20%</td>
<td>35.48%</td>
<td>72.71%</td>
<td>86.30%</td>
</tr>
<tr>
<td></td>
<td>cat</td>
<td>chair</td>
<td>cow</td>
<td>diningtable</td>
<td>dog</td>
<td>horse</td>
<td>motorbike</td>
</tr>
<tr>
<td>Fisher</td>
<td>59.92%</td>
<td>51.92%</td>
<td>47.60%</td>
<td>58.06%</td>
<td>42.28%</td>
<td>80.45%</td>
<td>69.34%</td>
</tr>
<tr>
<td>DeepNet</td>
<td>81.10%</td>
<td>51.04%</td>
<td>61.10%</td>
<td>64.62%</td>
<td>76.17%</td>
<td>81.60%</td>
<td>79.33%</td>
</tr>
<tr>
<td></td>
<td>person</td>
<td>pottedplant</td>
<td>sheep</td>
<td>sofa</td>
<td>train</td>
<td>tvmonitor</td>
<td>mAP</td>
</tr>
<tr>
<td>Fisher</td>
<td>85.10%</td>
<td>28.62%</td>
<td>49.58%</td>
<td>49.31%</td>
<td>82.71%</td>
<td>54.33%</td>
<td>59.99%</td>
</tr>
<tr>
<td>DeepNet</td>
<td>92.43%</td>
<td>49.99%</td>
<td>74.04%</td>
<td>49.48%</td>
<td>87.07%</td>
<td>67.08%</td>
<td>72.12%</td>
</tr>
</tbody>
</table>

Image  | FV  | DNN  |
Machine Learning in the Sciences
Machine Learning in Neuroscience
BBCI Set-up: *Let the machines learn*

- **multi-channel EEG**
  - FFT based low-pass filter
  - band-pass 4-40 Hz -> AR coeffs.
  - subject-specific band-pass filter, e.g. 7-14Hz, -> multi-class CSP

- **multiple feature extraction**
  - $x_{MRF}$
  - $x_{AR}$
  - $x_{CSP}$

- **classifier**
  - feature combiner 'PROB'

- **continuous feedback**
  - or

- **Artifact removal**

---

Brain Computer Interfacing: 'Brain Pong'

Leitmotiv: »let the machines learn«

Berlin Brain Computer Interface

- ML reduces patient training from 300h -> 5min

Applications

- help/hope for patients (ALS, stroke…)
- neuroscience
- neurotechnology (video coding, gaming, monitoring driving)
DNN Explanation Motor Imagery BCI

Note: Explanation available for single Trial (Sturm et al submitted)
Machine Learning in Chemistry, Physics and Materials

Matthias Rupp, Anatole von Lilienfeld, Alexandre Tkatchenko, Klaus-Robert Müller
Machine Learning for chemical compound space

Ansatz:

\[ \{Z_I, R_I\} \xrightarrow{\text{ML}} E \]

instead of

\[ \hat{H}(\{Z_I, R_I\}) \xrightarrow{\Psi} E \]

\[ \hat{H}\Psi = E\Psi \]

[from von Lilienfeld]
Coulomb representation of molecules

\[ M_{ii} = Z_i^{2.4} \]
\[ M_{ij} = \frac{Z_i Z_j}{|R_i - R_j|} \]

\[ M \in^{23 \times 23} \]

Coulomb Matrix (Rupp, Müller et al 2012, PRL)

\[ d(M, M') = \sqrt{\sum_{IJ} |M_{IJ} - M'_{IJ}|^2} \]
Kernel ridge regression

Distances between $\mathbf{M}$ define Gaussian kernel matrix $\mathbf{K}$

$$k(\mathbf{M}, \mathbf{M}') = \exp\left(-\frac{d(\mathbf{M}, \mathbf{M}')^2}{2\sigma^2}\right)$$

Predict energy as sum over weighted Gaussians

$$E^{est}(\mathbf{M}) = \sum_i \alpha_i k(\mathbf{M}, \mathbf{M}_i) + b$$

using weights that minimize error in training set

$$\min_{\alpha} \sum_i \left( E^{est}(\mathbf{M}_i) - E_i^{ref} \right)^2 + \lambda \sum_i \alpha_i^2$$

Exact solution

$$\alpha = (\mathbf{K} + \lambda \mathbf{I})^{-1} \mathbf{E}^{ref}$$

As many parameters as molecules + 2 global parameters, characteristic length-scale or $kT$ of system ($\sigma$), and noise-level ($\lambda$)

[from von Lilienfeld]
Predicting Energy of small molecules: Results

March 2012
Rupp et al., PRL
9.99 kcal/mol
(kernels + eigenspectrum)

December 2012
Montavon et al., NIPS
3.51 kcal/mol
(Neural nets + Coulomb sets)

2015 Hansen et al 1.3 kcal/mol at 10 million times faster than the state of the art

Prediction considered chemically accurate when MAE is below 1 kcal/mol

Dataset available at http://quantum-machine.org
Learning Atomistic Representations with Deep Tensor Neural Networks

Kristof Schütt, Farhad Arbabzadah, Stefan Chmiela, Alexandre Tkatchenko, Klaus-Robert Müller
Input Representation

\[
Z = \begin{bmatrix}
Z_1 & Z_2 & \cdots & Z_n
\end{bmatrix}
\]

\[
D = \begin{bmatrix}
d_{11} & d_{12} & \cdots & d_{1n} \\
d_{21} & d_{12} & \cdots & d_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
d_{n1} & d_{n2} & \cdots & d_{nn}
\end{bmatrix}
\]
Deep Tensor Neural Network

Atom descriptors:

\[ c_i^{(0)} = c_{Z_i} \in \mathbb{R}^d \]

\[ c_i^{(t+1)} = c_i^{(t)} + \sum_{j \neq i} \tanh(v_{ij}) \]

Interaction correction:

\[ v_{ij} = c_j^{(t)} V_k \hat{d}_{ij} + (W^c c_j^{(t)})_k + (W^d \hat{d}_{ij})_k + b_k \]
\[
\tanh \left( W^c_r \left( (W^c_r \mathbf{c}_j + b^r) \odot (W^d_r \mathbf{d}_{ij} + b^d) \right) \right)
\]
1. **Assign initial atomic descriptors**

   We assign an initial coefficient vector to each atom $i$ of the molecule according to its nuclear charge $Z$:

   $$ \mathbf{c}_i^{(0)} = \mathbf{c}_Z \in \mathbb{R}^B, $$

   where $B$ is the number of basis functions. All presented models use atomic descriptors with 30 coefficients. We initialize each coefficient randomly following $\mathbf{c}_Z \sim \mathcal{N}(0, 1/\sqrt{B})$.

2. **Gaussian feature expansion of the inter-atomic distances**

   The inter-atomic distances $d_{ij}$ are spread across many dimensions by a uniform grid of Gaussians

   $$ \hat{d}_{ij} = \left[ \exp \left( -\frac{(d_{ij} - (\mu_{min} + k\Delta\mu))^2}{2\sigma^2} \right) \right]_{0 \leq k \Delta\mu \leq \mu_{max}/\Delta\mu}, $$

   with $\Delta\mu$ being the gap between two Gaussians of width $\sigma$.

   In our experiments, we set both to 0.2 Å. The center of the first Gaussian $\mu_{min}$ was set to −1, while $\mu_{max}$ was chosen depending on the range of distances in the data (10 Å for GDB-7 and benzene, 15 Å for toluene, malonaldehyde and salicylic acid and 20 Å for GDB-9).

3. **Perform $T$ interaction passes**

   Each coefficient vector $\mathbf{c}_i^{(t)}$, corresponding to atom $i$ after $t$ passes, is corrected by the interactions with the other atoms of the molecule:

   $$ \mathbf{c}_i^{(t+1)} = \mathbf{c}_i^{(t)} + \sum_{j \neq i} v_{ij}. $$

   Here, we model the interaction $v$ as follows:

   $$ v_{ij} = \tanh \left( W_{ij}^f \left( (W_{ij}^{cf} \mathbf{c}_j + \mathbf{b}^f) \circ (W_{ij}^{df} \hat{d}_{ij} + \mathbf{b}^d) \right) \right), $$

   where the circle ($\circ$) represents the element-wise matrix product. The factor representation in the presented models employs 60 neurons.
4. Predict energy contributions

Finally, we predict the energy contributions $E_i$ from each atom $i$. Employing two fully-connected layers, for each atom a scaled energy contribution $\hat{E}_i$ is predicted:

$$o_i = \tanh(W_i^{out1}c_i^{(T)} + b_i^{out1})$$

$$\hat{E}_i = W_i^{out2}o_i + b_i^{out2}$$

In our experiments, the hidden layer $o_i$ possesses 15 neurons. To obtain the final contributions, $\hat{E}_i$ is shifted to the mean $E_\mu$ and scaled by the standard deviation $E_{std}$ of the energy per atom estimated on the training set.

$$E_i = (\hat{E}_i + E_\mu)E_{std}$$

This procedure ensures a good starting point for the training.

5. Obtain the molecular energy $E = \sum_i E_i$
Chemical Compound Space

Mean absolute error [kcal mol⁻¹]

- $T = 1$
- $T = 2$
- $T = 3$

# of training examples

mean abs. error [kcal mol⁻¹]

molecules with ≥ 20 atoms

# atoms

# add. calc. ≤ 15 atoms
Molecular Dynamics Simulations

Mean absolute error [kcal mol$^{-1}$] vs. # of training examples.

- $T = 1$
- $T = 2$
- $T = 3$

Total energy [kcal mol$^{-1}$] vs. time step.
Explaining and Visualizing the learned interactions

Motivation:
- Successive, interaction refinements are hard to grasp
- Look at the effect of these interactions
- Visualize the local potential $\phi_{Z_p}(r)$ learned by the network

Procedure:
- Introduce probe atom with charge $Z_p$ at position $r$
- Predict probe energy $E_p$ by letting the molecule correct the probe descriptor $c_p$ multiple times (but not vice versa)
- Move probe over space
Local 'potentials' for various probes
Quantum Chemical Insights: aromaticity

Figure 3 | Classification of molecular carbon ring stability. Shown are 20 molecules (10 most stable and 10 least stable) with respect to the energy of the carbon ring predicted by the DTNN model. Atom colouring: carbon = black; hydrogen = white; oxygen = red; nitrogen = blue; fluorine = yellow.
Quantum Chemical Insights

Energy prediction:
\[ E = \sum_{i=1}^{n} E_i \]

Learned potential:
\[ \Omega_A^M(r) = E_{probe} \]
Conclusion

• explaining & interpreting nonlinear models is essential
• orthogonal to improving DNNs and other models
• need for opening the blackbox …
• understanding nonlinear models is essential for Sciences & AI
• new theory: LRP is based on deep taylor expansion SAMEK LECTURE

Remark: @NIPS 2017 ML4QC & XPLAINABLE ML workshops
Further Reading I


Further Reading II


Further Reading III


