Tackling the problem of “bad” explanations with the Human-in-the-Loop principle

Dipl. -Ing. Anna Saranti, PhD Student of Prof. Andreas Holzinger

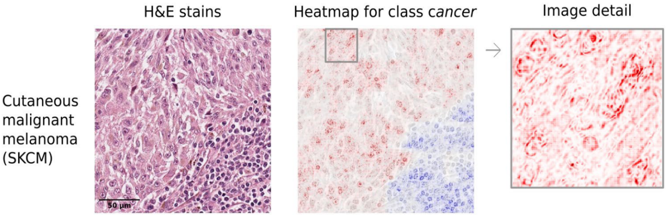
01. April 2022

Outline

1. What is a good explanation?
2. What is a bad explanation?
3. Graphs
4. Graph Neural Networks (GNN)
5. xAI on GNNs
6. Literature
7. Questions

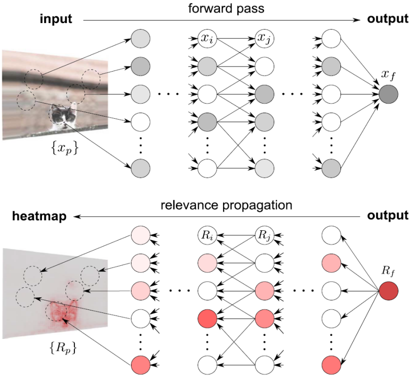
Heatmaps

- Binary classification task
- Cancer or healthy?



Hägele, Miriam, et al. "Resolving challenges in deep learning-based analyses of histopathological images using explanation methods." Scientific reports 10.1 (2020): 1-12.

How does LRP work? - Computational flow



Lapuschkin, Sebastian, et al. “Unmasking clever hans predictors and assessing what machines really learn.” Nature communications 10.1 (2019): 1-8.

LRP vs. SA (1/2)

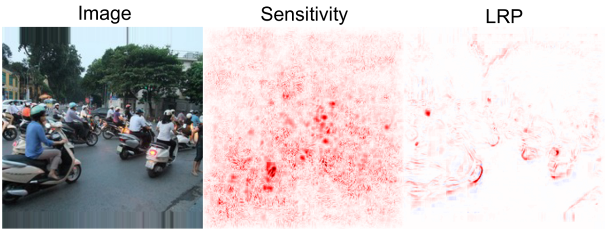
- What is a good heatmap?
- Sensitivity of a pixel p is the norm over all partial derivatives:

  $$h_p = ||\frac{\partial}{\partial x_p} f(x)||$$

- How much a small change in the pixel p affects the prediction (output) of the NN
- The direction of change is lost because of the norm
- Needs (locally) differentiable neurons

LRP vs. SA (2/2)

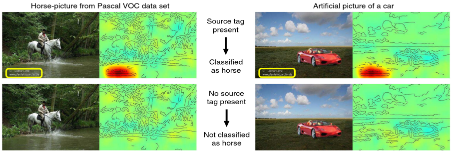
- Blue color denotes negative relevane
  Evidence **against** the predicted class



Samek, Wojciech, et al. “Interpreting the predictions of complex ml models by layer-wise relevance propagation.” arXiv preprint arXiv:1611.08191 (2016).

Whole dataset analysis (1/2)

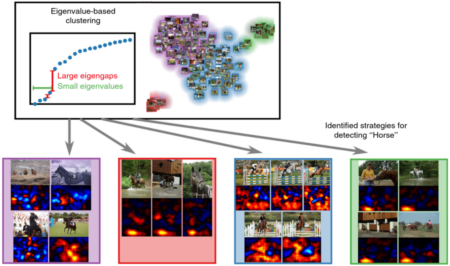
- PASCAL VOC2007 data set: horse images have a tag
- Classification by high-performing NN
- Use LRP and detect Clever Hans predictions



Lapuschkin, Sebastian, et al. “Unmasking clever hans predictors and assessing what machines really learn.” Nature communications 10.1 (2019): 1-8.

Whole dataset analysis (2/2)

- Semi-automated Spectral Relevance Analysis
- Improve the model and the dataset



Lapuschkin, Sebastian, et al. "Unmasking clever hans predictors and assessing what machines really learn." Nature communications 10.1 (2019): 1-8.

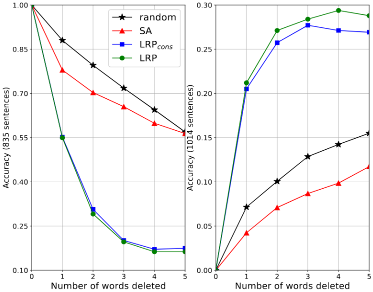
LRP on LSTMs and Perturbation Analysis (1/2)

- Sentiment classification task

true	predicted	N°	Notation: -- very negative, - negative, 0 neutral, + positive, ++ very positive
		1.	do n't waste your money .
		2.	neither funny nor suspenseful nor particularly well-drawn .
		3.	it 's not horrible , just horribly mediocre .
		4.	... too slow , too boring , and occasionally annoying .
		5.	it 's neither as romantic nor as thrilling as it should be .
--	--	6.	the master of disaster - it 's a piece of dreck disguised as comedy .
		7.	so stupid , so ill-conceived , so badly drawn , it created whole new levels of ugly .
		8.	a film so tedious that it is impossible to care whether that boast is true or not .
		9.	choppy editing and too many repetitive scenes spoil what could have been an important documentary about stand-up comedy .
		10.	this idea has lost its originality ... and neither star appears very excited at rehashing what was basically a one-joke picture .
		1.	do n't waste your money .
		2.	neither funny nor suspenseful nor particularly well-drawn .
		3.	it 's not horrible , just horribly mediocre .
		4.	... too slow , too boring , and occasionally annoying .
		5.	it 's neither as romantic nor as thrilling as it should be .
--	--	6.	the master of disaster - it 's a piece of dreck disguised as comedy .
		7.	so stupid , so ill-conceived , so badly drawn , it created whole new levels of ugly .
		8.	a film so tedious that it is impossible to care whether that boast is true or not .
		9.	choppy editing and too many repetitive scenes spoil what could have been an important documentary about stand-up comedy .
		10.	this idea has lost its originality ... and neither star appears very excited at rehashing what was basically a one-joke picture .

Arras, Leila, et al. “Explaining recurrent neural network predictions in sentiment analysis.” arXiv preprint arXiv:1706.07206 (2017)

LRP on LSTMs and Perturbation Analysis (2/2)

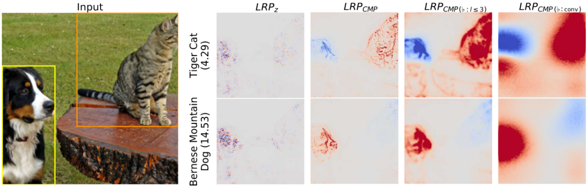


- How does word deleting affect performance?
- Left: Correct classification, decreasing relevance
- Right: Misclassification, increasing relevance

Arras, Leila, et al. "Explaining recurrent neural network predictions in sentiment analysis." arXiv preprint arXiv:1706.07206 (2017)

Positive and negative relevance is important (1/2)

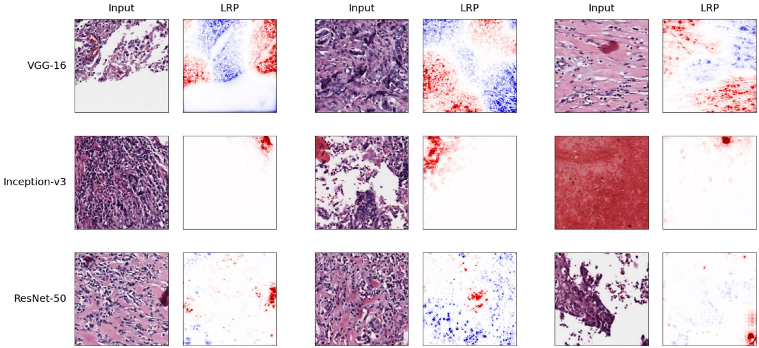
- Classification: is it a cat or a dog?
- What speaks for or against a decision?
- Can a human decide? What would the human say?



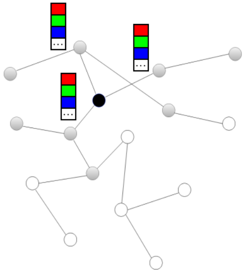
Kohlbrenner, Maximilian, et al. “Towards best practice in explaining neural network decisions with LRP.” 2020 International Joint Conference on Neural Networks (IJCNN). IEEE, 2020.

Positive and negative relevance is important (2/2)

- Do humans trust AI when its prediction is wrong?

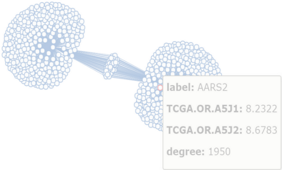


Graph data (1/5)



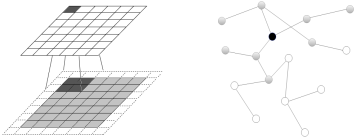
- Not just 3 features, but any number
- Size
  shape
  degree
  type
  ...

Graph data (2/5)



- Sequential, Grid $\leftrightarrow$ Graph data
- Biological data, Drug discovery, Social networks, Maps
- Images, Reinforcement Learning states

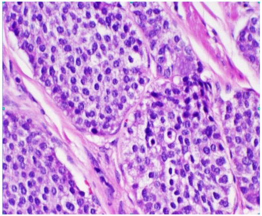
Graph data (3/5)

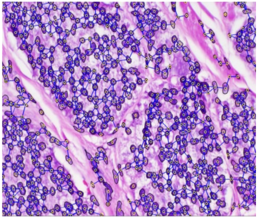


- Each pixel has 3 features (RGB)
- How does a CNN operate?
  Gathers information from the neighborhood

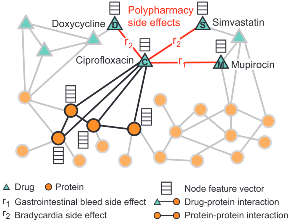
Graph data (4/5)

- `histocartography`: `https://github.com/histocartography/histocartography`
- Centroids and texture features for each node: convex area, length of the major and minor axis, orientation, convex hull perimeter, ellipticity, roudness ...





Graph data (5/5)



Zitnik, Marinka, Monica Agrawal, and Jure Leskovec. “Modeling polypharmacy side effects with graph convolutional networks.” Bioinformatics 34.13 (2018): i457-i466.

- Features on edges: distance, weight, tissue node
- Heterogeneous graphs: nodes and or edges of different type $\rightarrow$ different features
- Multigraphs: many edges between two nodes

Graphs mathematical description

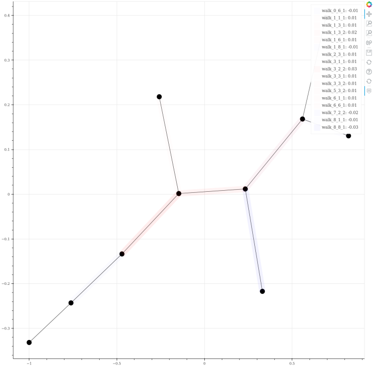
- $\mathcal{G} = (\mathcal{V}, \mathcal{E})$: set of nodes and edges
- Directed vs. undirected, simple vs. multi-relational (heterogeneous), self-loops
- Adjacency matrix: $\mathbf{A} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$
  Laplacian matrix: $\mathbf{L} = \mathbf{D} - \mathbf{A}$
- Shortest path, degree, connected components:

```
nx.connected_components(G),
nx.draw(G, pos=nx.circular_layout(G),
        node_color='r', edge_color='b')
```

Software for graphs and visualization

[Python]

Graph nr. 6, Explanation with gamma: 0.1



- networkx: https://networkx.org/
- matplotlib: https://matplotlib.org/
- pyviz: https://pyviz.org/
- Bokeh: https://bokeh.org/

Define graph(s)

```
import networkx as nx

G=nx.Graph()

G.add_node(1)
G.add_node(2)

G.add_edge(1, 2)
```

```
G_1=nx.complete_graph(9)

G_2=nx.cycle_graph(5)

G_3=nx.star_graph(5)

G_4=barabasi_albert_graph
    (5, 2)
```

Software for GNN

[Python]

- `Pytorch Geometric (PyG)`: `https://pytorch-geometric.readthedocs.io/en/latest/`
- DGL: `https://www.dgl.ai/`

Compatibility with `networkx`:

```
torch_geometric.utils.convert.from_networkx(...)
torch_geometric.utils.convert.to_networkx(...,
                        to_undirected, ...)
```

Graph datasets (benchmarks)

- https://pytorch-geometric.readthedocs.io/en/latest/modules/datasets.html
- Open Graph Benchmark datasets: https://ogb.stanford.edu/

```
dataset = TUDataset(root='data/TUDataset',
                    name='MUTAG')
print(f'Number of graphs: {len(dataset)}')
print(f'Number of node features:
     {dataset.num_features}')
print(f'Number of classes: {dataset.num_classes}')
```

Graph representation in PyG (1/3)

`torch_geometric.data.Data`

- `data.x`: Node feature matrix
  [num_nodes, num_node_features]
- `data.edge_attr`: Edge feature matrix
  [num_edges, num_edge_features]

	color	size	shape
node_0	0.1	0.0	-0.1
node_1	-0.5	0.05	-0.1

https://scikit-learn.org/stable/modules/classes.html#module-sklearn.preprocessing

Graph representation in PyG (2/3)

- `data.edge_index`: [2, num_edges]
  $[[0, 2, 3],$
  $[2, 4, 1]]$
- `data.y`: targets (node or graph classification)
  $[0, 0, 1, 1, 1, 1, ...]$
- `data.pos`: [num_nodes, num_dimensions]

Graph representation in PyG (3/3)

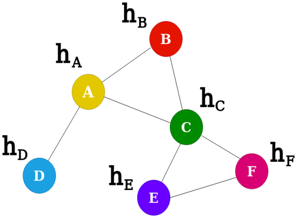
- `**kwargs (optional)`: Additional attributes

```
graph=Data(x=node_attributes_x, y=None,
           edge_index=edge_idx, edge_attr=None,
           pos=None,

           node_labels=node_labels,
           node_ids=node_ids,
           node_feature_labels=node_feature_labels,
           edge_ids=edge_ids,
           edge_attr_labels=edge_attr_labels)
```

Neural message passing (1/6)

Graph with node features/embeddings $h(v)$:

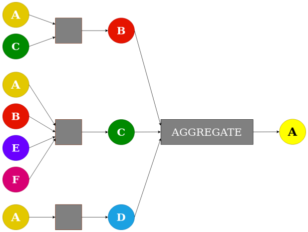


... k times - the initial values of the features are replaced with new ones

Neural message passing (2/6)

Computational graph - $\mathcal{N}(v)$: Neighborhood

$$a_v^{(k)} = \mathrm{AGGREGATE}^{(k)}\left(\left\{h_u^{(k-1)} : u \in \mathcal{N}(v)\right\}\right)$$



Neural message passing (3/6)

$$h_v^{(k)} = \mathrm{COMBINE}^{(k)}\left(h_u^{(k-1)}, a_v^{(k)}\right)$$

What is an appropriate $\mathrm{AGGREGATE}$ function?

- `mean()`
- `max()`
- `sum()`

... reminds Belief Propagation in Conditional Random Fields (CRF)

Neural message passing (4/6)

- $\mathrm{COMBINE}$ is implemented by a Multi-Layer Perceptron (MLP)
- Overall function:

$$h_v^{(k)} = \mathrm{MLP}^{(k)}\left((1 + e^{(k)}) \cdot h_v^{(k-1)} + \sum_{u \in \mathcal{N}(v)} h_u^{(k-1)}\right)$$

- You can write your own module!

Neural message passing (5/6)

```
class GCN(torch.nn.Module):
    def __init__(self, num_node_features: int,
                 hidden_channels: int,
                 num_classes: int):
        super(GCN, self).__init__()
        self.conv1 = GCNConv(num_node_features,
                             hidden_channels)
        self.conv2 = GCNConv(hidden_channels,
                             hidden_channels)
        self.lin = Linear(hidden_channels,
                          num_classes)
```

Neural message passing (6/6)

```
def forward(self, x, edge_index, batch):
    x = self.conv1(x, edge_index)
    x = x.relu()
    x = self.conv2(x, edge_index)
    x = x.relu()
    x = global_mean_pool(x, batch)
    x = F.dropout(x, p=0.2,
                  training=self.training)
    x = self.lin(x)
```

- Use comments and formatting!

GNN Tasks [overview] (1/6)

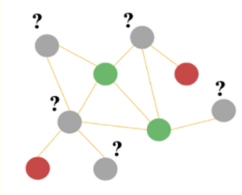
1. Node classification
2. Link prediction
3. Graph classification

What will xAI methods compute?

- Images $\rightarrow$ heatmap
- Graphs $\rightarrow$ relevant subgraphs, walks, and causal structures

GNN Tasks - Node classification (2/6)

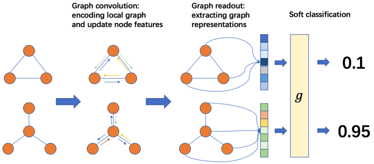
- Input: Graph
- Some nodes labeled
- Label the unlabeled ones



https://docs.dgl.ai/tutorials/blitz/index.html

Tasks - Graph classification (3/6)

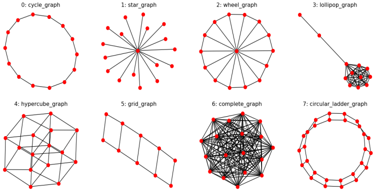
- How is node classification related to graph classification? -
  Use the end values of the node features after the last application of aggregate and combine.



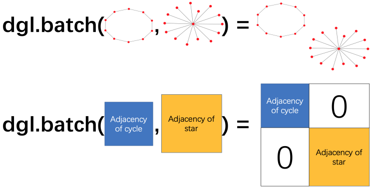
https://docs.dgl.ai/en/0.6.x/guide/training-graph.html

Tasks - Graph classification (4/6)

Batch Adjacency Matrix:



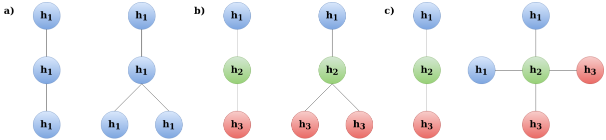
https://docs.dgl.ai/en/0.6.x/guide/training-graph.html



https://docs.dgl.ai/en/0.6.x/guide/training-graph.html

Tasks - Graph classification (5/6)

Graph Isomorphism Network (GIN) architecture:



Can the pairs be differentiated [discrimination]?

- a) mean and maximum of several h_1 same
- b) $\max(h_1, h_2, h_3) = \max(h_1, h_2, h_3, h_3)$
- c) $\frac{1}{2}(h_1 + h_3) = \frac{1}{4}(2 \cdot h_1 + 2 \cdot h_3)$

Tasks - Graph classification (6/6)

- Aggregations implemented by `mean()` and `max()` cannot distinguish between very simple graph structures
- Use `sum()`
- Representationally more powerful - as powerful as the Weisfeiler-Lehman graph isomorphism test

```
torch_geometric.nn.conv.gin_conv
```

GNNExplainer (1/2)

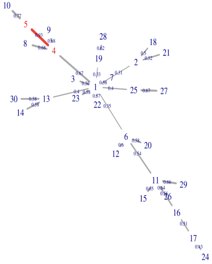
- Compute the important subgraph G_S of the computation graph G_c of input graph G
- Optimization algorithm - iteratively find the substructure that maximizes the mutual information (MI) w.r.t. the prediction score
- X_S: Subset of features of nodes in subgraph G_S.
- **Y**: Predicted label distribution

$$\max_{G_S} \mathrm{MI}\left(\mathbf{Y}, (G_S, X_S)\right) = H(\mathbf{Y}) - H(\mathbf{Y} | \mathbf{G} = G_S, \mathbf{X} = X_S)$$

GNNExplainer (2/2)

Synthetic data

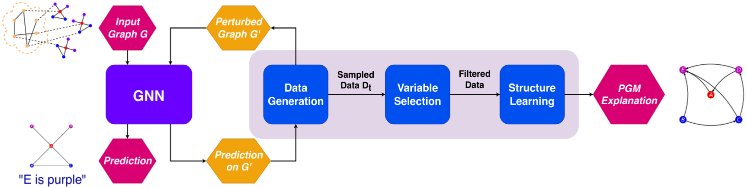
- Barabasi graphs
- Node features: $\mathcal{N}(\mu = 0, \sigma = 0.1)$
- 1000 graphs, same topology
- Edge $4 - 5$: $\mathcal{N}(\mu = -1, \sigma = 0.1)$
- Graph classification



Pfeifer, Bastian, et al. “GNN-SubNet: disease subnetwork detection with explainable Graph Neural Networks.” bioRxiv (2022).

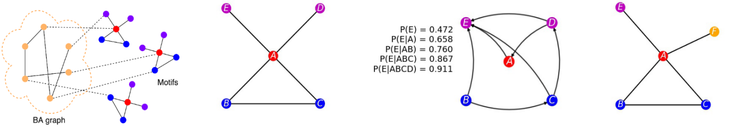
PGMExplainer (1/2)

- Perturb the input to uncover dependencies
- Learn a Bayesian Network (BN) from the generated data $\rightarrow$ structure and parameter learning



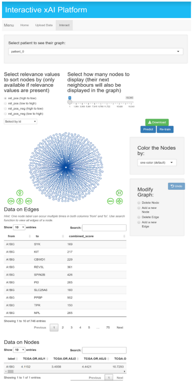
PGMExplainer (2/2)

- Estimate the probability that node E has the predicted role (w.r.t. node classification) given the realization (values of features) of other nodes
- pgmpy: `https://pgmpy.org/`



(a) Input graph. (b) Motif containing E. (c) PGM-Explainer. (d) GNNExplainer.

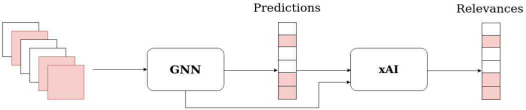
GNN Counterfactuals UI platform (1/3)



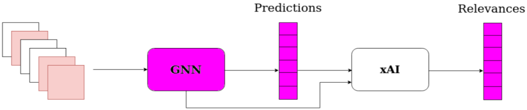
- Add/delete nodes and edges
- Add/delete features
- Predict/Retrain
- Good performance - good explanations?
- Incorporate human domain knowledge

GNN Counterfactuals UI platform (2/3)

- Predict

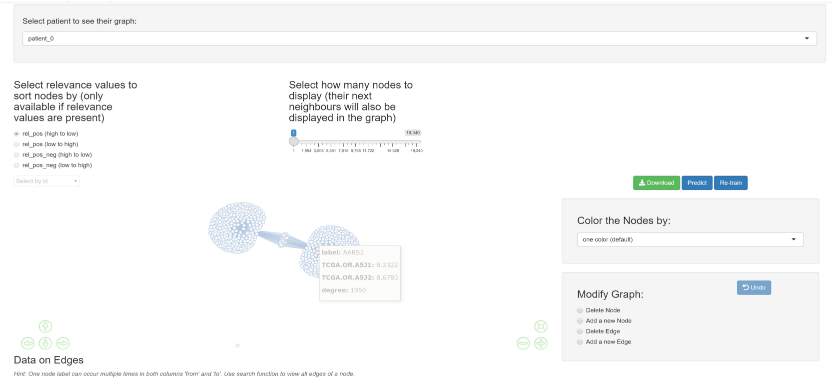


- Retrain



GNN Counterfactuals UI platform (3/3)

Human-in-the-loop in the Causability Lab:



Literature (1/6)

Main LRP paper:

- Montavon, Grégoire, et al. “Explaining nonlinear classification decisions with deep taylor decomposition.” Pattern Recognition 65 (2017): 211-222.

Practical tutorial on xAI techniques:

- Bennetot, Adrien, et al. “A Practical Tutorial on Explainable AI Techniques.” arXiv preprint arXiv:2111.14260 (2021).

Literature (2/6)

Differences with Sensitivity Analysis (SA):

- Montavon, Grégoire, Wojciech Samek, and Klaus-Robert Müller. “Methods for interpreting and understanding deep neural networks.” Digital Signal Processing 73 (2018): 1-15.
- Bach, Sebastian, et al. “On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation.” PloS one 10.7 (2015): e0130140.

Literature (3/6)

Graph datasets:

- Hu, Weihua, et al. “Open graph benchmark: Datasets for machine learning on graphs.” arXiv preprint arXiv:2005.00687 (2020).

Literature (4/6)

GNN:

- William L. Hamilton “Graph Representation Learning”, Synthesis Lectures on Artifical Intelligence and Machine Learning 14.3 (2020): 1-159.
- Geometric Deep Learning -
  Grids, Groups, Graphs, Geodesics, and Gauges
  https://geometricdeeplearning.com/

Literature (5/6)

GNN architectures:

- Xu, Keyulu, et al. “How powerful are graph neural networks?.” arXiv preprint arXiv:1810.00826 (2018).
- Xu, Keyulu, et al. “Representation learning on graphs with jumping knowledge networks.” International Conference on Machine Learning. PMLR, 2018.
- Loukas, Andreas. “What graph neural networks cannot learn: depth vs width.” arXiv preprint arXiv:1907.03199 (2019).

Literature (6/6)

xAI on GNN:

- Ying, Rex, et al. “Gnnexplainer: Generating explanations for graph neural networks.” Advances in neural information processing systems 32 (2019): 9240.
- Vu, Minh N., and My T. Thai. “Pgm-explainer: Probabilistic graphical model explanations for graph neural networks.” arXiv preprint arXiv:2010.05788 (2020).
- Schnake, Thomas, et al. “XAI for graphs: explaining graph neural network predictions by identifying relevant walks.” arXiv e-prints (2020): arXiv-2006.

- Questions?

- Dipl. -Ing. Anna Saranti
  anna.saranti@medunigraz.at