Graph Neural Networks (GNNs)

Anna Saranti & Andreas Holzinger

human-centered.ai

27.04.2021

Outline

Medical data $\leftrightarrow$ Neural Networks

- Neural networks process medical images, text and graph data
- Explanations of AI systems are necessary to shed light on decision criteria, build trust, uncover causes, debugging purposes
- Medical doctors also epxlain their decisions -

  Can neural networks use this knowledge?

  What are the similarities and differences?

Data (1/3)

1. Histopathological images
2. Textual descriptions/explanations
3. Biological *omics data

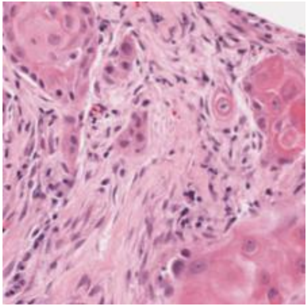

Figure 1: Example histopathological image

Neural Networks (2/3)

- Different CNN architectures for image processing:
  VGG-16, Inception, ResNet-50, ResNet-50 v2
- Bi-LSTMs and BERT for text processing: (dialogue systems)
- Graph Neural Networks for graph processing

```
base_model = ResNet50V2(
    weights="imagenet",
    include_top=False,
    input_tensor=tf.keras.layers.Input(shape=(img_height, img_width, 3)),
)
```

Figure 2: Example code for particular CNN architecture

Explanation methods (3/3)

1. LRP and CAM for CNNs
2. LRP for LSTMs
3. GNNExplainer and LRP for GNNs

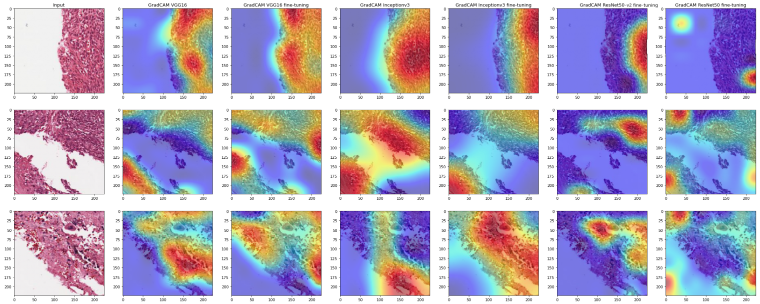

Figure 3: CAM on histopathological image

How Graph Neural Networks (GNN) work (1/2)

- Convolutional Neural Network (CNN) on grid-structured data
- Graph Neural Network (GNN) not necessarily grid-structured input

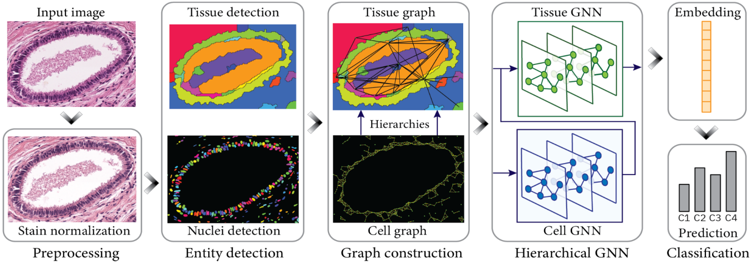

Figure 4: Nuclei Instance Segmentation and Tissue Graph

Pati, Pushpak, et al. "HACT-Net: A Hierarchical Cell-to-Tissue Graph Neural Network for Histopathological Image Classification." Uncertainty for Safe Utilization of Machine Learning in Medical Imaging, and Graphs in Biomedical Image Analysis. Springer, Cham, 2020. 208-219.

How Graph Neural Networks (GNN) work (2/2)

- Graph G = (V, E), nodes and edges that have features that can be of different types (heterogeneous)
- Computer-generated data with known ground truth - Is this going to be uncovered by xAI explanation methods?

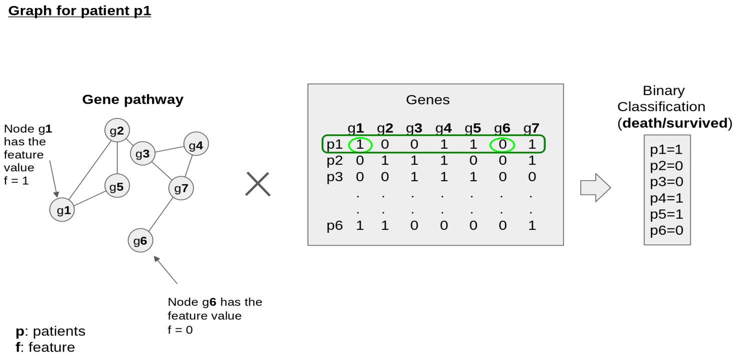

Figure 5: Graph Classification

How does message passing work?

- Training of the GNNs updates the representations of the features of nodes and edges
- Each node will update its own feature with information sent from neighboring nodes

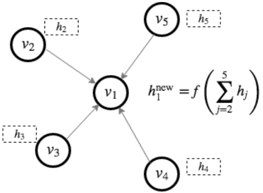

Figure 6: Message passing

What tasks do GNNs solve? - Different aggregation functions (1/3)

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

- After `k` iterations of recursive neighborhood aggregation the node's transformed feature vector captures the structural information of `k`-hop neighborhood

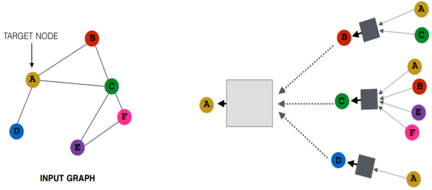

Figure 7: Node classification

What tasks do GNNs solve? - Different aggregation functions (2/3)

- Operators: `SUM`, `MEAN`, `MAX`
- Map two nodes to the same location in the embedding space only if they have identical tree substructures with identical features
- To distinguish different graph structures use injective aggregation operators

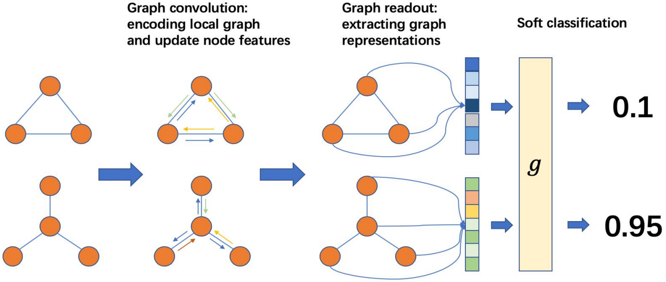

Figure 8: Graph classification

What tasks do GNNs solve? - Different aggregation functions (3/3)

- Representational power of operators
- a) Same features a vs. SUM: 2∗a, 3∗a
- b) $\texttt{MAX}(h_g, h_r)$, $\texttt{MAX}(h_g, h_r, h_r)$ vs. SUM: $\frac{1}{2}(h_g + h_r)$, $\frac{1}{3}(h_g + h_r + h_r)$
- c) $\frac{1}{2}(h_g + h_r) = \frac{1}{4}(h_g + h_g + h_r + h_r)$

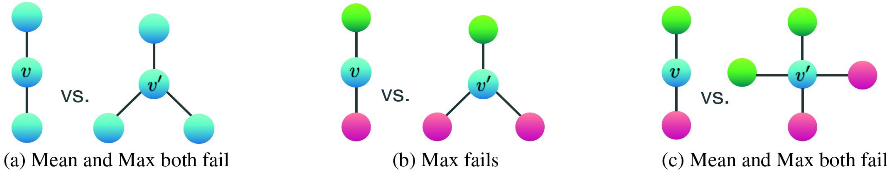

Figure 9: Graph structures where MEAN and MAX aggregation operators fail to distinguish

Code example

```python
class Classifier(nn.Module):

    def __init__(self, in_dim, hidden_dim, n_classes):
        super(Classifier, self).__init__()
        self.conv1 = dglnn.GraphConv(in_dim, hidden_dim)
        self.conv2 = dglnn.GraphConv(hidden_dim, hidden_dim)
        self.classify = nn.Linear(hidden_dim, n_classes)

    def forward(self, g, h):

        # Apply graph convolution and activation.
        h = F.relu(self.conv1(g, h))
        h = F.relu(self.conv2(g, h))

        with g.local_scope():
            g.ndata['h'] = h

            # Calculate graph representation by average readout.
            hg = dgl.mean_nodes(g, 'h')

            return self.classify(hg)


model = Classifier(7, 20, 2)
opt = torch.optim.Adam(model.parameters())
```

Figure 10: Code of GNN with the use of Python's GDL library: `https://www.dgl.ai/`

Explainable AI methods on GNN: GNNExplainer (1/2)

- Features of the nodes that are important
- Subgraphs of the graph that are decisive

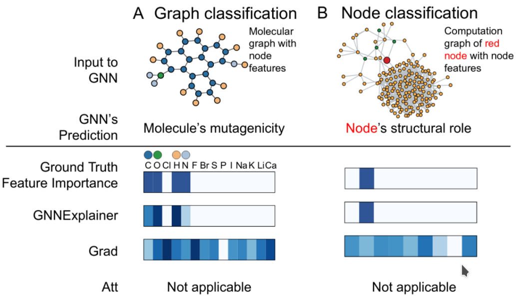

Figure 11: GNNExplainer detecting important features and compared with domain knowledge

Explainable AI methods on GNN: LRP (2/2)

Result of LRP when applied in GNN:

- Collection of relevant walks on the input graph, not individual nodes and edges
- Each walk $\mathcal{W}$: one relevance score $R_{\mathcal{W}}$
- Constraint: Linear or ReLU activation
- Validate that the model uses the graph structure meaningfully
- Search over many possible paths

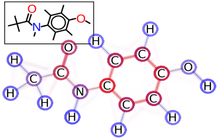

Figure 12: Application of LRP on GNN uncovering relevant paths

Literature (1/3)

Graph extraction from medical images:

- Zhou, Yanning, et al. "Cgc-net: Cell graph convolutional network for grading of colorectal cancer histology images." Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops. 2019.
- Pati, Pushpak, et al. "HACT-Net: A Hierarchical Cell-to-Tissue Graph Neural Network for Histopathological Image Classification." Uncertainty for Safe Utilization of Machine Learning in Medical Imaging, and Graphs in Biomedical Image Analysis. Springer, Cham, 2020. 208-219.

Literature (2/3)

GNNs:

- Kipf, Thomas N., and Max Welling. "Semi-supervised classification with graph convolutional networks." arXiv preprint arXiv:1609.02907 (2016).
- Xu, Keyulu, et al. "How powerful are graph neural networks?." arXiv preprint arXiv:1810.00826 (2018).

- `https://docs.dgl.ai/`
- `https://pytorch-geometric.readthedocs.io/en/latest/`

Literature (3/3)

xAI on GNNs:

- Thomas Schnake, et al. "XAI for Graphs: Explaining Graph Neural Network Predictions by Identifying Relevant Walks", (2020) arXiv: 2006.03589
- Kristina Preuer, et al. "Interpretable Deep Learning in Drug Discovery", Springer LNAI 11700, (2019) 331–345
- Robert Schwarzenberg, et al. "Layerwise Relevance Visualization in Convolutional Text Graph Classifiers", EMNLP Workshop on Graph-Based Natural Language Processing (2019)
- Zhitao Ying, et al. "GNNexplainer: Generating explanations for graph neural networks", Advances in neural information processing systems (2019): 9244-9255.