Project Assignments

Andrea Passerini
andrea.passerini@unitn.it

Advanced Topics in Machine Learning and Optimization

Boost GNN performance

Description

- Message passing exchange node features in the network. What if we do not have node features? Usually, we add an array of ones as node features, **Can we do better?**
- Try a simple GCN and a GAT network on a dataset without node features. Could you explain the performance of GAT?
- Can you try to change those ones in something that can be more useful for the GAT network?

Notes

- **Hint**: try to use topological metrics of the graph as node features
- **Reference**: Tutorials
- **Contact**: Antonio Longa

Open Graph Benchmark

Description

- Chose one dataset from OGB
- Implement your own GNN and compare your results with those reported in the leaderboards. (we do not require your algorithm will be the first on the leaderboard ;))
- Suppose the student decide for arxiv dataset [link]
- Implement your own algorithm, compare your test and validation accuracy with those on the leaderboard [link]
- If you like, follow the instructions on OGB to upload your own model (optional)

Notes

- **Reference**: Dataset,Tutorials
- **Contact**: Antonio Longa
- **Can be selected multiple times**

Graph Autoencoders (learning the decoding)

Description

- The traditional GAE takes in input a graph, apply a GCN in the ecoding part (node embedding $= z$), while in the decoding part they only apply a dot product ($Adj' = z \cdot z^T$).
- Can you increase the performance of GAE using some strategy in the decomposition part?
- Implement a simple GAE on a banchmark dataset.
- Build your own decoder part as you wish
- Compare the performance obtained by your GAE and the original one.

Notes

- **Reference**:GAE paper , Tutorials
- **Contact**: Antonio Longa
- **Extensible to thesis**

GNN for social context classification

Description

- Use GNN to embed **Temporal Graphs** in a latent space, then the student should try to explore the latent space.
- Download two (or more) different dataset from sociopattern.org
- Decide a temporal aggregation to build a sequence of graphs starting from a temporal graph.
- Load the graphs in PyTroch Geometric
- Build a classifier able to distinguish which graphs belongs to its own category

Notes

- **Reference**:Dataset, Tutorials
- **Contact**: Antonio Longa
- **Extensible to thesis**

Explaining GNN Biases

Description

- Pick a number of diverse and representative GNN architectures (e.g. plain GCN, attention models, higher-order models, etc)
- Pick a a number of diverse and representative benchmarks
- Pick a GNN explainability technique that can produce model-based explanations (rather than instance-based only)
- Extract explanations for different architectures on different benchmarks and search for pattern/regularities

Notes

- **Contact**: Andrea Passerini
- **Extensible to thesis**

Knowledge-driven GNN selection

Description

- Choose a predictive problem in computational biology involving graph data (e.g. PPI prediction)
- Using domain knowledge, identify connectivity patterns that can be relevant for the prediction (e.g. motifs)
- Choose the GNN architecture that you believe can best discover and leverage these patterns
- Train the GNN and check the quality of the results and the patterns learned (using tools for explaining GNN)

Notes

- **Contact**: Andrea Passerini
- **Extensible to thesis**

DeepProbLog vs Deep Network

Assignment

- Consider the MNIST multi-digit addition example we saw in the lecture
- Design a purely neural architecture that predicts the result of the addition (e.g. CNN+LSTM, but feel free to invent)
- Compare results of the neural architecture with those of DeepProbLog in the same setting:
  - supervision only on the result of the addition
  - generalization to longer numbers
- Try training the neural network with a larger training set than the one used for DeepProbLog (check how many examples are need to match DeepProbLog performance)

Notes

- **Contact**: Andrea Passerini
- **Can be selected multiple times**

DeepProbLog Tasks

Description

- Start from the MNIST multi-digit addition example we saw in the lecture
- Modify it to address *octal* division between two numbers (assume second is an integer divisor of first)
- Always assume (as in the example) that training instances have one-digit numbers, test instances multiple digit numbers

Notes

- **Contact**: Andrea Passerini

NeurASP vs DeepProbLog

Description

- **NeurASP:** A neuro-symbolic framework that extends Answer Set Programming (ASP) with neural networks
- Compare NeurASP vs DeepProbLog on Sudoku solving
- Retrain the NeurASP model on Sudoku and collect accuracy (as done in the paper)
- Train a DeepProbLog model on the same data and collect accuracy
- Compare accuracy, show predictions, and report training time of both frameworks

Notes

- **Contact**: Gianluca Apriceno
- **Extensible to thesis**
- **References**: paper, repository

Saliency Maps + Example-based Explanations

Description

Influence functions (IFs) identify examples responsible for a prediction, but not what attributes make them influential

- Reference implementation is available
- Uses IFs to compute example-based explanations
- Integrate attribution-based explanations, either post-hoc or into the IF computation

Notes

- **Contact**: Stefano Teso
- **Extensible to thesis**
- **Reference**: paper, code

Explain Learned Concepts

Description

Are the concepts learned by gray-box models truly understandable?

- Pick one or more GBMs and apply it/them to a controlled setting (e.g., birds dataset)
- Implement a strategy to visualize and understand the concepts that it learns
- Optionally, evaluate whether concepts can be improved by supplying extra supervision

Notes

- **Contact**: Stefano Teso
- **Extensible to thesis**
- **Reference**: SENN/CBM/PCN/PPNet paper, ref in slides

Controlled Forgetting for Deep Prototypes

Description

Existing algorithm adapts *k*NN to concept drift by moving examples between classes.

- Adapt code to work with one or more prototype-based NNs
- Evaluate impact of embeddings on quality of the forgetting
- Optionally, if necessary, implement a strategy to update embeddings too

Notes

- **Contact**: Stefano Teso
- **Extensible to thesis**
- **Reference**: paper

Explanation-guided Queries for Active Learning

Description

Improve query selection in active learning using explanations

- It uses uncertainty sampling on the labels to identify query instances, ignores model's explanations, could be sub-optimal
- Define, implement, and evaluate query strategy that uses uncertainty on model's explanations

Notes

- **Contact**: Stefano Teso
- **Extensible to thesis**
- **Reference**: paper

Explanation-aware Drift Detection

Description

Implement existing technique for detecting concept drift that leverages the model's explanations

- Find or write (possibly simplified) implementation
- Evaluate on one of the data sets described in the original paper
- Optionally, design a confounded data set where model's explanations are misaligned with concept drift & the algorithm fails

Notes

- **Contact**: Stefano Teso
- **Extensible to thesis**
- **Reference**: paper

Graybox Models for Medical Decision Making Support

Description

- Ultrasound scans of the lungs can identify patterns that indicate problems due to infections (e.g. by covid-19)
- Deep architectures can be used to predict whether a certain image from a scan shows problems in the lung
- The task is that of training a graybox model (e.g. a PPNet) to predict problems in the lung and:
  - Compare it to (existing) blackbox models
  - Check how the intepretable patterns (e.g. part-prototypes) match what physicians think is relevant

Notes

- **Contact**: Andrea Passerini
- **Extensible to thesis**

Assignment

- Select one of the projects from the previous slides (or discussed with the teacher for custom projects)
- Complete it and prepare a report summarizing the methodology used and the results obtained.
- After completing the assignment send it via email to andrea.passerini@unitn.it
- Subject: ADVML2022
- Attachment: name_surname.zip containing:
  - the report (named report.pdf)
  - the code you wrote
  - the requirements needed to run the code

NOTE

- No group work
- Preliminary versions of the report can be sent for feedback
- The project is discussed asynchronously as soon as it is completed