A Formal Methods Approach Towards Deep Learning Interpretability

Kriten Kessel, Christopher Lazarus, Javier Sagastuy
Stanford University

Summary

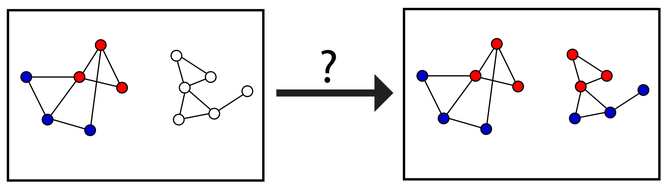
Deep Neural Networks are very useful at classifying tasks but their intrinsic complexity makes it really hard to explain the reasoning behind a classification outcome. In recent work [2], statistical methods were introduced to help assess the influence of human intelligible concepts in classification outcomes. We aim to asses and extend such methods by leveraging formal methods for Neural Network verification.

- **Problem:** Why did the network classify image $\rangle$ with label k?
- **Solution:** Come up with classes TCAV etc bla bla? or more like we tried to test the TCAV method??
- **Preliminary Results:** nothing nothing nothing.

Talk about TCAV ?

Input:

- adjacency matrices A_L and A_U for the two sets of nodes;
- features vectors X_L and X_U;
- labels y_L

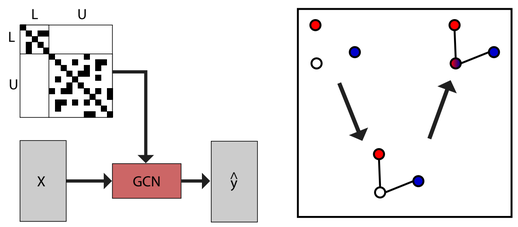


Output:

- predictions for the unlabelled nodes $\hat{y}_L$

Neural Network Verification

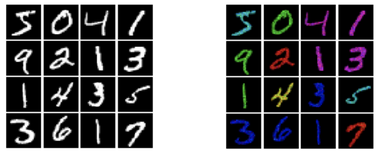
GCN [1] for node classification: $\hat{y} = h(Ah(AXW_1)W_2)$



Approach: TCAV + Verification

Three-step process

- learn low-dimensional node embeddings to encode node similarity (VGAE [2])
- hallucinate edges and complete the adjacency matrix (edges)
- run a GCN with the completed adjacency to predict node labels



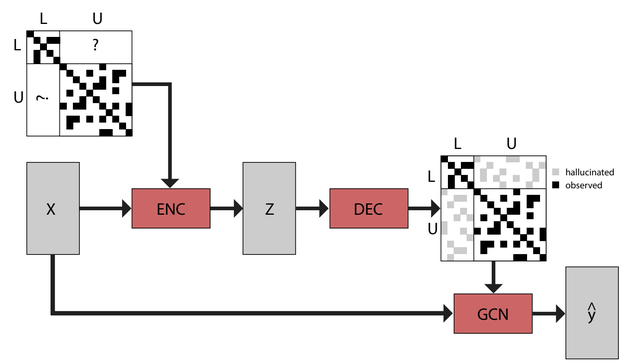
(a) label 1 (b) label 2

2 Figures side by side

1. Link prediction - Variational Graph Auto-Encoder (VGAE):

- $Z \sim \mathcal{N}(\mu_Z, \sigma_Z^2)$ and $\tilde{A} = \sigma(ZZ^T)$.
- with $\mu_Z, \sigma_Z = \mathrm{GCN}(A, X)$

$$\mathcal{L}_{LP} = -\mathbb{E}_{Z\sim q(Z|A,X)}[A_{ij}\log\tilde{A}_{ij} + (1-A_{ij})\log(1-\tilde{A}_{i,j})] + \mathrm{KL}(q(Z|A,X)||p(Z)).$$



2. Edge hallucination produces $\hat{A}$:

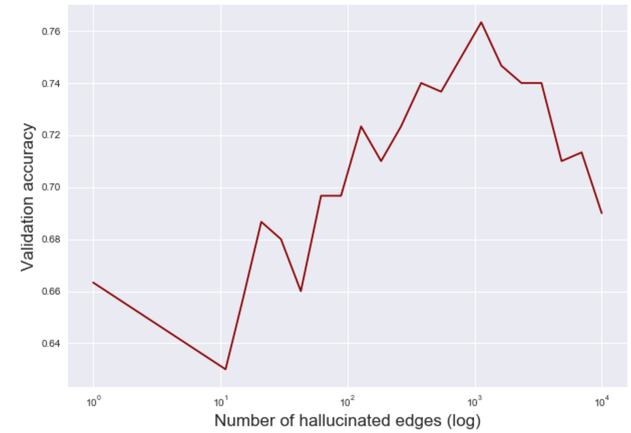
- top$K$ (K hyper-parameter)
- sampling using gumbel softmax [4] trick (allows gradients to flow)

3. Node classification

- $\hat{y} = \mathrm{GCN}(\hat{A}, X)$

Results

Classification performance per number of hallucinated edges (cora)



Node classification results

model	balanced	blue 2	red 2	green 2
balanced_5x50	0.942	0.942	0.941	0.944
blue2_5x50	0.780	0.952	0.682	0.689
balanced_3x50	0.942	0.940	0.942	0.941
blue2_3x50	0.724	0.954	0.675	0.684
balanced_3x20	0.890	0.890	0.891	0.888
blue2_3x20	0.708	0.923	0.663	0.672

Table: Accuracies for various models and datasets. The balanced_DxW models are trained on a color-balanced dataset, and the blue2_DxW models are trained on a color-biased dataset where all 2's are blue, as shown in Fig **??**. Columns indicate validation datasets which are balanced or have a blue, green, or red bias on class 2, respectively.

- As shown in [1], the GCN improves on standalone MLP by leveraging the connections between nodes
- By adding "hallucinated" edges, we improve the connectivity structure, and we obtain more predictive power

References

[1] G. Katz, C. Barrett, D. L. Dill, K. Julian, and M. J. Kochenderfer. Reluplex: An efficient smt solver for verifying deep neural networks. In International Conference on Computer Aided Verification, pages 97–117. Springer, 2017

[2] B. Kim, M. Wattenberg, J. Gilmer, C. J. Cai, J. Wexler, F. B. Viegas, and R. Sayres. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In ICML, 2018

[3] Y. LeCun, C. Cortes, and C. Burges. Mnist handwritten digit database. ATT Labs [Online]. Available: http://yann. lecun. com/exdb/mnist, 2:18, 2010