Graph Neural Networks in Rock-Climbing Classification

Cheng-Hao Tai (c2tai@Stanford.edu)   Aaron Wu (aaron@evisort.com)   Rafael Hinojosa (rahinojo@stanford.edu)

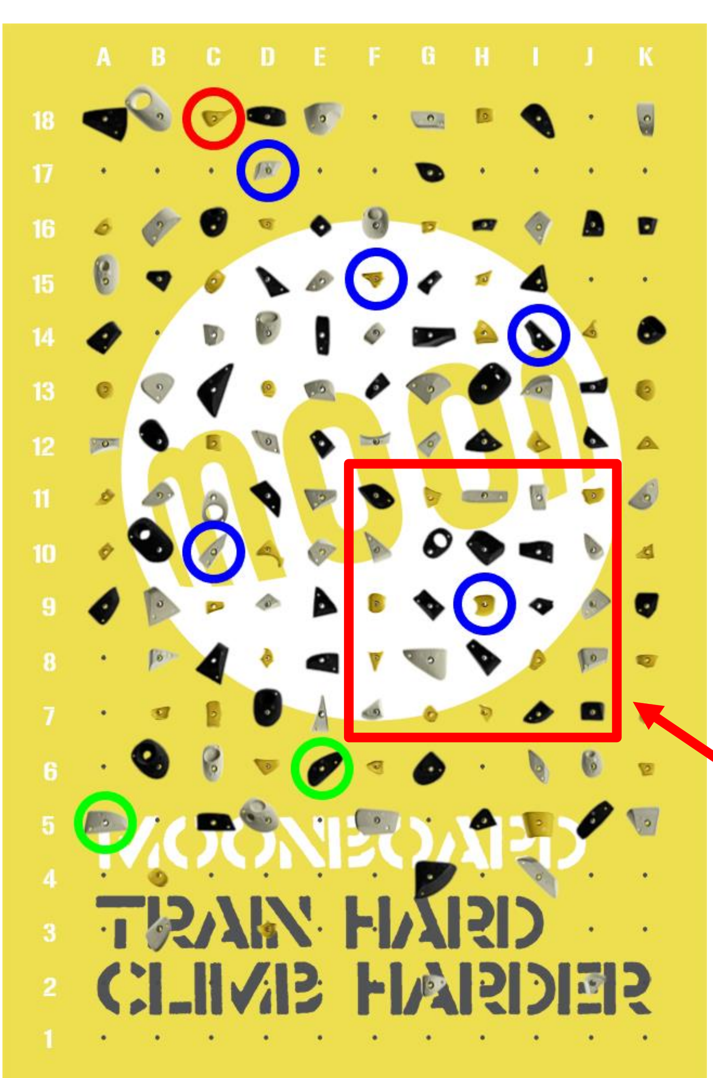
https://youtu.be/8cISaVsOqs0

OVERVIEW

Taking inspiration from the NLP domain, we utilized a graph convolutional network to build a classifier for determining rock climbing problem difficulties. Graph convolutional networks allow for each graph node to be embedded as a nonlinear combination of its k hop neighbors – a characteristic highly desirable in our problem since each climbing route is directly relatable to other climbing routes via shared holds. Our best-performing model achieves 0.73 AUC.

DATA

Data was sourced using Selenium from MoonBoard's website. A total of 13,589 problems were collected and preprocessed into one-hot and multi-hot formats. For multi-hot, vectors are 140-dimensional long where each dimension encodes presence / absence of a hold. We used an 80/20 ratio for train-test split.



$$\mathrm{PMI}(i,j) = \log\left(\frac{p(i,j)}{p(i)p(j)}\right)$$

$$p(i,j) = \frac{\#W(i,j)}{\#W}$$

$$p(i) = \frac{\#W(i)}{\#W}$$

$\#W(i,j)$ is the number of windows in which holds i,j show up

$\#W(i)$ is only for hold i
$\#W(j)$ is only for hold j

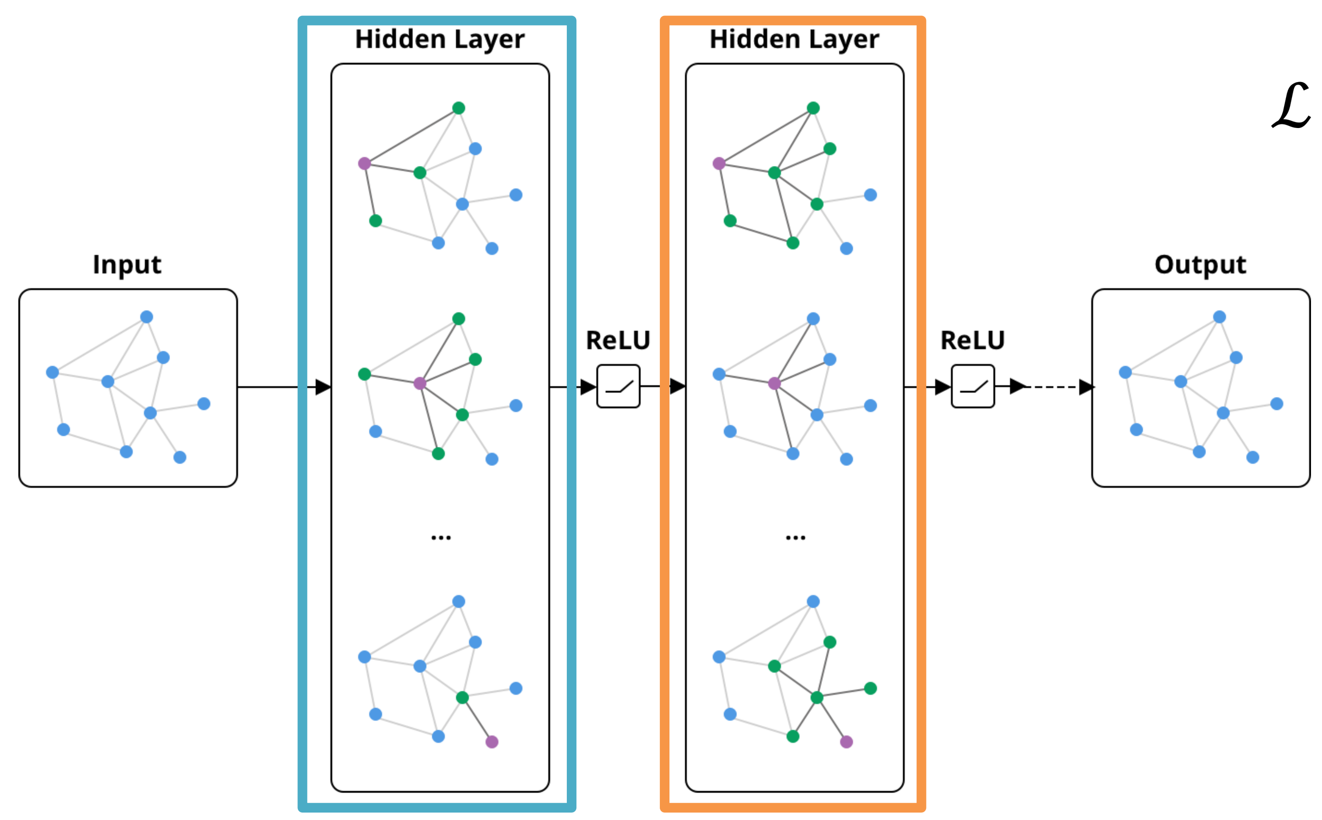
GRAPH CONVOLUTIONAL NETWORKS

With one step of graph convolution, nodes can access information from 1-hop neighbors. Two steps yields 2-hop neighbors. Forward propagation in GCNs are:

$$L^{(j+1)} = \rho(\tilde{A}L^{(j)}W_j)$$

Where $\tilde{A} = D^{-\frac{1}{2}}\,A\,D^{-\frac{1}{2}}$ is the normalized symmetric adjacency matrix. A two-layer GCN's forward pass is

$$Z = \mathrm{softmax}(\tilde{A}\;\mathrm{ReLU}(\tilde{A}\;X\;W_0)W_1)$$



$$\mathcal{L} = \sum_{p \in \mathcal{Y}_P} \sum_{f=1}^{F} Y_{pf} \log(Z_{pf})$$

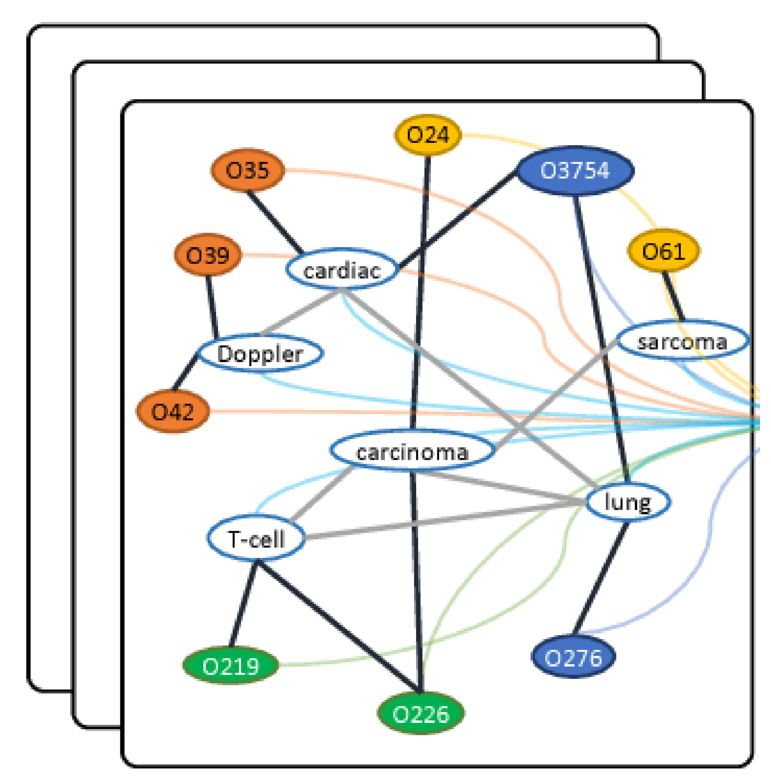
The loss function for this graph-based classification task applies cross-entropy on a subset of labeled nodes $\mathcal{Y}_P$

BUILDING THE GRAPH

A heterogenous graph is used to model the corpus of MoonBoard problems. Adjacency between nodes are modeled as

$$A_{ij} = \begin{cases} \mathrm{PMI}(i,j), \text{if } i,j \text{ holds} \\ \mathrm{IDF}(j), \text{if } i \text{ problem } j \text{ hold} \\ 1, \text{if } i = j \\ 0, \text{otherwise} \end{cases}$$

Both problems and holds are represented as nodes on the heterogenous graph – (right) sample for a heterogeneous document / word graph



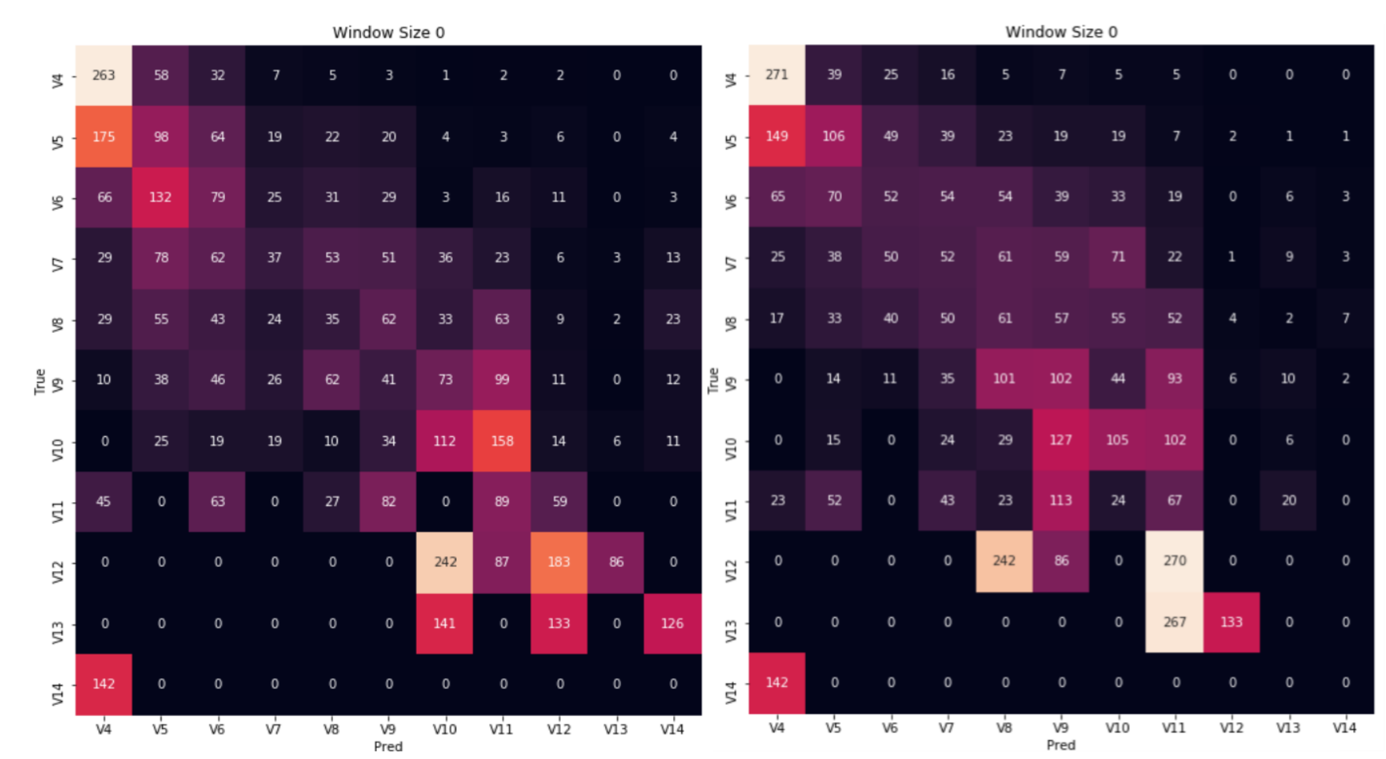
Word Document Graph

DISCUSSION

A total of 17 experiments were run to benchmark the GCN and evaluate its performance. We found that:

- GCN with multi-hot features outperform all baseline models and fully-connected networks
- GCN is less sensitive to extreme data imbalance and produces better predictions across the spectrum of difficulty classes

Confusion matrices – GCN (left), logistic reg. (right)



RESULTS

Summary of experiments including baseline statistical models, fully-connected networks, and GCNs

	Per-Class F1 Scores							Avg.	
Experiment	**V4**	**V5**	**V6**	...	**V12**	**V13**	**V14**	**F1**	**AUC**
Logistic Regression	0.52	0.35	0.25	...	0.22	0.00	0.00	0.23	0.70
SVM	0.53	0.40	0.31	...	0.34	0.33	0.00	0.29	0.66
Random Forest	0.66	0.36	0.26	...	0.24	0.00	0.44	0.28	0.67
Gradient Boosting	0.59	0.35	0.28	...	0.27	0.25	0.00	0.27	0.62
MLP	0.58	0.38	0.34	...	0.00	0.00	0.00	0.22	0.66
Dense Shallow, MH	0.61	0.38	0.32	...	0.37	0.00	0.00	0.26	0.67
Dense Deep, MH	0.48	0.40	0.23	...	0.21	0.00	0.00	0.22	0.65
GCN (S) 2S, OH, PMI	0.36	0.28	0.22	...	0.48	0.24	0.10	0.24	0.63
GCN (L) 2S, OH, PMI	0.33	0.27	0.23	...	0.48	0.25	0.13	0.24	0.64
GCN (S) 2S, MH, PMI	0.57	0.38	0.33	...	0.49	0.00	0.00	**0.29**	**0.73**
GCN (L) 2S, MH, PMI	0.58	0.38	0.33	...	0.45	0.00	0.00	**0.29**	**0.72**
GCN (S) 2S, MH, Binary	0.54	0.38	0.33	...	0.46	0.00	0.00	**0.28**	**0.72**
GCN (S) 2S, MH, Self-Binary	0.58	0.41	0.30	...	0.14	0.00	0.00	0.25	0.70
GCN (S) 2S, MH, Self-PMI	0.59	0.38	0.29	...	0.56	0.00	0.00	0.27	0.68
GCN (L) 4S, MH, PMI	0.52	0.34	0.33	...	0.50	0.00	0.37	**0.31**	**0.73**
GCN (S) 2S, MH, Win-PMI	0.55	0.37	0.33	...	0.48	0.00	0.00	**0.28**	**0.73**
GCN (L) 2S, MH, Win-PMI	0.57	0.38	0.34	...	0.47	0.00	0.00	**0.29**	**0.72**