INSTITUTO
DE INFORMÁTICA
UFRGS

LEARNING CENTRALITY MEASURES WITH GRAPH NEURAL NETWORKS

Pedro Henrique da Costa Avelar
Advisor: Luís da Cunha Lamb

October 28, 2019

Introduction

- Machine Learning subarea:

- Machine Learning subarea:
    - Structures of linear transformations and nonlinear applications.

- Machine Learning subarea:
  - Structures of linear transformations and nonlinear applications.
  - Learning internal representations of data.

- Machine Learning subarea:
  - Structures of linear transformations and nonlinear applications.
  - Learning internal representations of data.
  - "Grown" from Artificial Neural Networks.

- Machine Learning subarea:
    - Structures of linear transformations and nonlinear applications.
    - Learning internal representations of data.
    - “Grown” from Artificial Neural Networks.
- Success generally attributed to the heightened parallel processing capacity with GPUs and due to the high data availability.

- Machine Learning subarea:
    - Structures of linear transformations and nonlinear applications.
    - Learning internal representations of data.
    - “Grown” from Artificial Neural Networks.
- Success generally attributed to the heightened parallel processing capacity with GPUs and due to the high data availability.
- Becoming ubiquitous in our daily lives, with manifold applications on diverse areas.

Figure: Recent success which uses deep learning for image processing – Image generation with the StyleGAN model. Source: KARRAS; LAINE; AILA (2018)

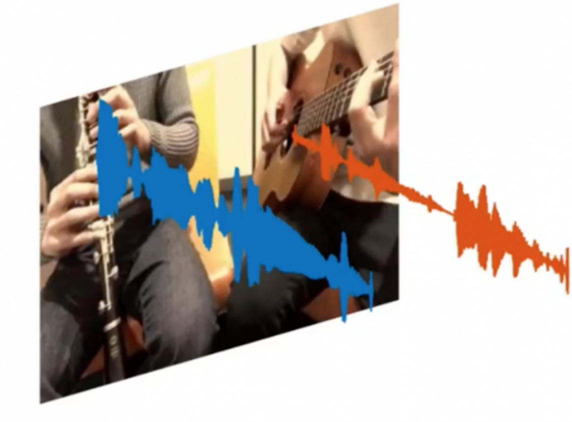

Figure: Recent success which uses deep learning for audio processing – Sound localisation with the PixelPlayer model. Source: ZHAO et al. (2018)

RECENT SUCCESSES – TEXT PROCESSING

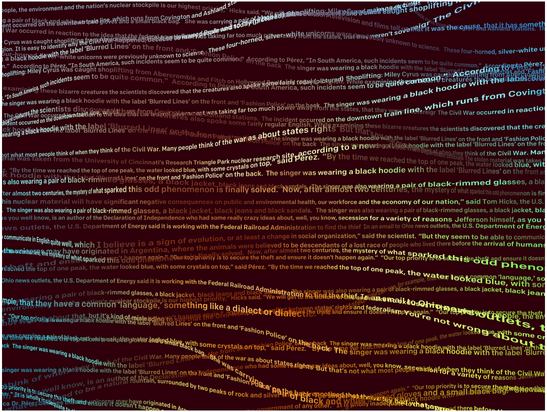

Figure: Recent success which uses deep learning for text processing – OpenAI's GPT-2 model. Source: OpenAI (RADFORD et al., 2019)

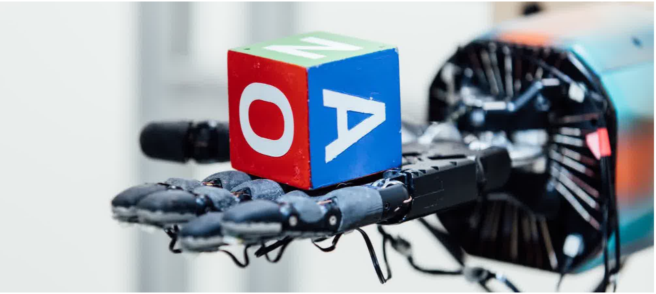

Figure: Recent success which uses deep learning for robotics. Source: OpenAI(OPENAI et al., 2018)

Figure: Recent sucesss that uses deep learning for a relational problem – In this case, playing the chinese boardgame Go. Source: Nature (SILVER et al., 2016)

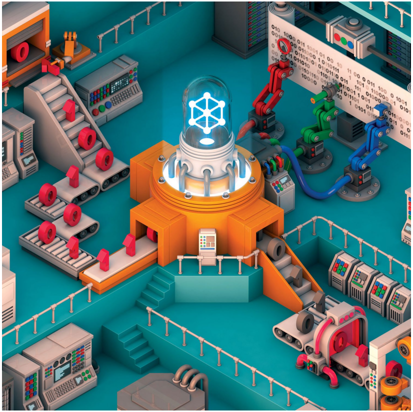

Figure: Recent sucesss that uses deep learning for a relational problem – In this case, the DNC model performs well in question answering and graph processing tasks. Source: Deepmind (GRAVES et al., 2016)

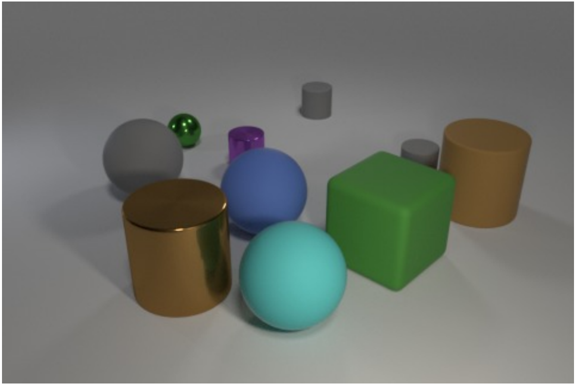

Figure: Recent sucessses that uses deep learning for a relational problem – In this case, answering relational questions about a synthetic image. Source: SANTORO et al. (2017)

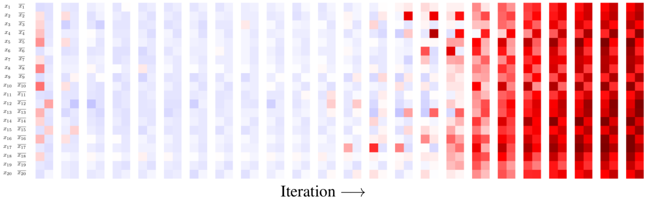

Figure: Recent sucesss that uses deep learning for a relational problem – In this case, solving SAT instances. Source: SELSAM et al. (2018)

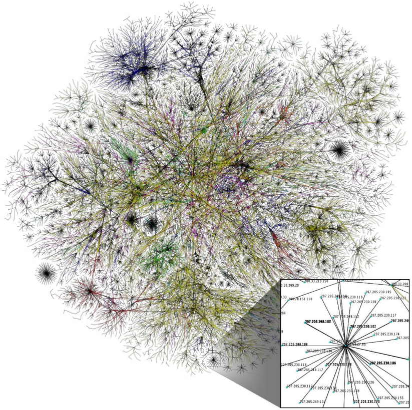

Figure: A partial map of the Internet based in 2005, made by opte.org: The relational structures that support our modern societies have been growing larger and more interconnected by the day. Source: Wikimedia Commons

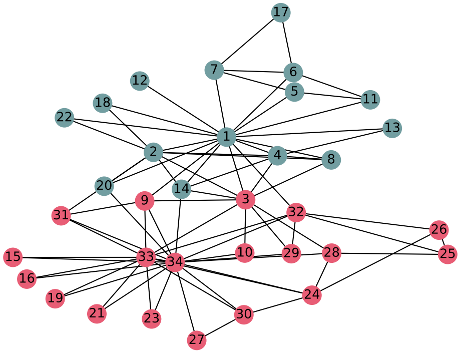

Figure: The connections of the 34 members of Zachary's Karate Club (WAYNE, 1977), a small social network. Here it is easy to see the equivalence between networks and graphs. Source: Author, data from (WAYNE, 1977) plotted using the Networkx Python package (HAGBERG; SWART; CHULT, 2008)

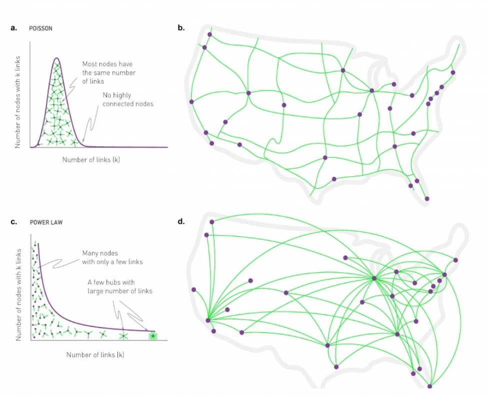

Figure: Two networks consisting of the same vertices, but with different degree distributions, exemplifying the Scale-Free property. Source: BARABÁSI et al. (2016)

- Defines how “important” an entity is

- Defines how “important” an entity is
- Many definitions of importance

- Defines how “important” an entity is
- Many definitions of importance
- Uses in (social) network analysis

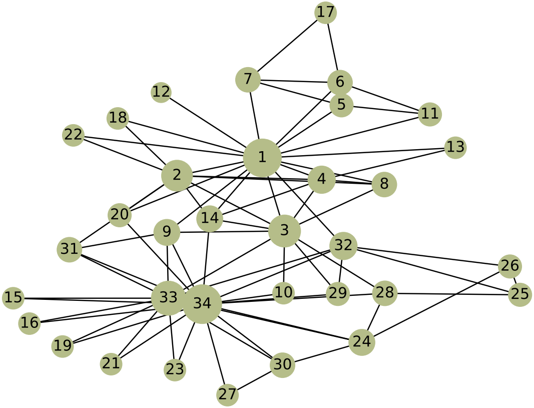

Figure: Zachary's Karate Club with nodes sized by degree. Source: Author, data from (WAYNE, 1977) plotted using the Networkx Python package (HAGBERG; SWART; CHULT, 2008)

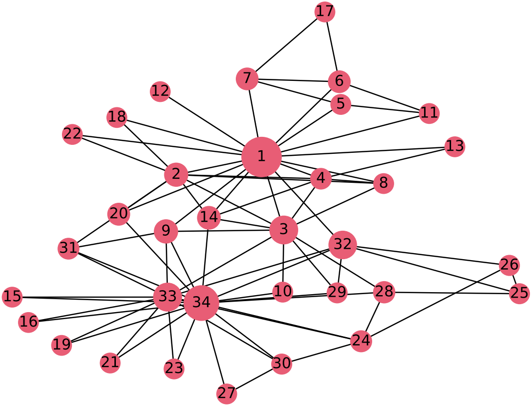

Figure: Zachary's Karate Club with nodes sized by betweenness. Source: Author, data from (WAYNE, 1977) plotted using the Networkx Python package (HAGBERG; SWART; CHULT, 2008)

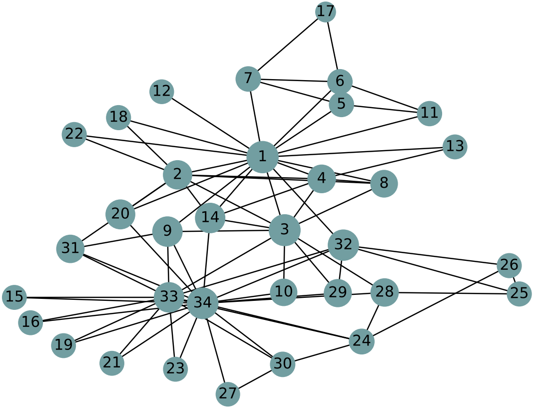

Figure: Zachary's Karate Club with nodes sized by closeness. Source: Author, data from (WAYNE, 1977) plotted using the Networkx Python package (HAGBERG; SWART; CHULT, 2008)

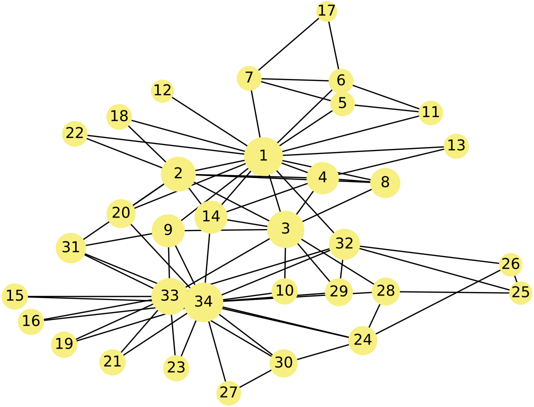

Figure: Zachary's Karate Club with nodes sized by eigenvector. Source: Author, data from (WAYNE, 1977) plotted using the Networkx Python package (HAGBERG; SWART; CHULT, 2008)

Graph Neural Networks

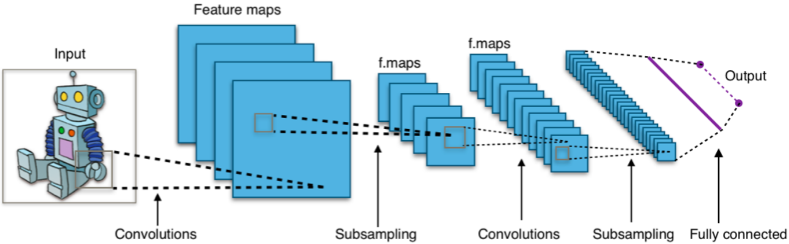

Figure: The typical architecture of a CNN. Source: Wikimedia Commons

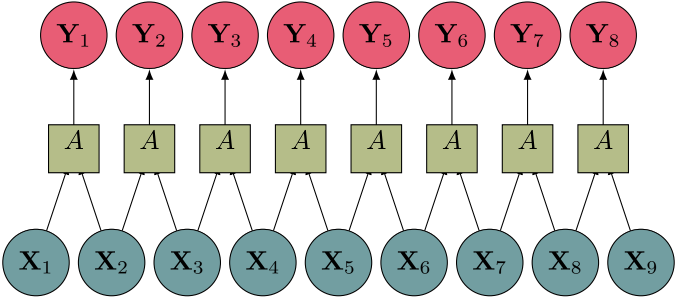

Figure: The application of a 1-dimensional convolutional kernel on a discrete 1-dimensional space. Blue circles are inputs, red circles are outputs and green backgrounds are to represent the whole neural network block. Source: Author, based on (OLAH, 2014)

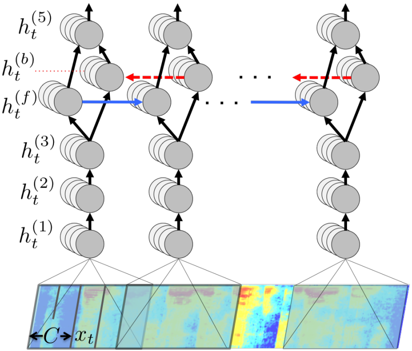

Figure: Deep Speech RNN architecture. Source: HANNUN et al. (2014)

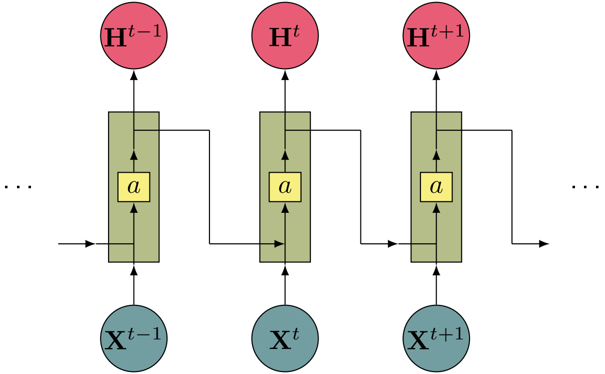

Figure: An unrolled recursive neural network. Yellow squares are neural network layers, blue circles are inputs, red circles are outputs and green backgrounds are to represent the whole neural network block. Source: Author, based on (OLAH, 2015)

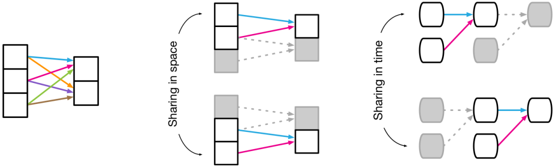

Figure: There is weight reuse across convolutional (middle) and recurrent (right) layers, but not in fully connected (left) layers. Source: BATTAGLIA et al. (2018)

- RNNs work on sequences

- RNNs work on sequences
- CNNs work on discrete spaces

- RNNs work on sequences
- CNNs work on discrete spaces
- What about graphs?

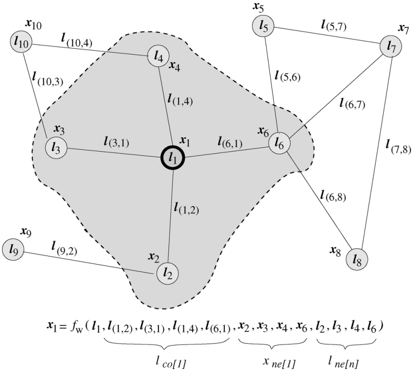

Figure: The representation of the Graph Neural Network Model, with the vertex being updated using the information on its neighbourhood Source: SCARSELLI et al. (2009)

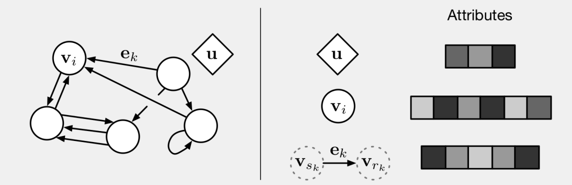

Figure: The representation of the Graph Network Model. Source: BATTAGLIA et al. (2018)

- BATTAGLIA et al. (2018) leaves no space for representing hypergraphs

AND HYPERGRAPHS?

- BATTAGLIA et al. (2018) leaves no space for representing hypergraphs
- The GNN model in SCARSELLI et al. (2009) is more general, and allows such operations

- BATTAGLIA et al. (2018) leaves no space for representing hypergraphs
- The GNN model in SCARSELLI et al. (2009) is more general, and allows such operations
- Reformalisation of the GNN model to generalise the concept of a vertex to a vertex's type.

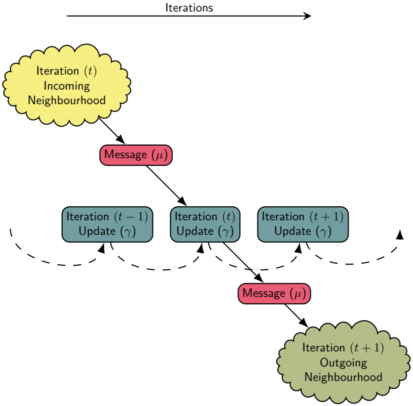

Figure: Pictorial representation of a Typed Graph Network from the perspective of a vertex v. Source: Author

TYPED GRAPH NEURAL NETWORKS I

procedure $\mathrm{TGN}(\mathcal{G} = (\mathcal{V} = \bigcup_{i=1}^{N} \mathcal{V}_i, \mathcal{E} = \bigcup_{k=1}^{K} \mathcal{E}_k), \mathcal{I} = \bigcup_{i=1}^{N} \mathbf{V_i}^{(0)})$

  for $t = 1 \dots t_{max}$ **do**

    for $i = 1 \dots N$ **do**

      Let $K_i \leftarrow \{k \mid \forall k, \pi_k = (s, i)\}$

      for all $v_b \in \mathcal{V}_i$ **do**

        for all $k \in K_i$ **do**

          $\overline{\mu}_{k,b}^{(t)} \leftarrow \{\mu_k(\mathbf{V_s}_{(a)}^{(t-1)}) \mid \forall v_a \in \mathcal{V}_s, (v_a, v_b) \in \mathcal{E}_k\}$

          $\overline{\alpha}_{k,v_t}^{(t)} \leftarrow \alpha_k(\overline{\mu}_{k,b}^{(t)})$

        end for

        $\overline{\rho}_{i,b}^{(t)} = \rho_i(\{\overline{\alpha}_{k,b}^{(t)} \mid \forall k \in K_i\})$

        $\mathbf{V_i}_{(b)}^{(t)} \leftarrow \gamma_i(\mathbf{V_i}^{(t)}, \overline{\rho}_{i,b}^{(t)})$

      end for

    end for

  end for

  return $\{\mathbf{V_i}^{(t_{max})} \mid i = 1 \dots N\}$

end procedure

$$K_i = \{k \mid \forall i, \pi_k = (s, i)\} \tag{1}$$

$$\overline{\mu}_{k,b}^{(t)} = \{\mu_k(\mathbf{V_s}_{(a)}^{(t-1)}) \mid \forall v_a \in \mathcal{V}_s, (v_a, v_b) \in \mathcal{E}_k\} \tag{2}$$

$$\overline{\alpha}_{k,b}^{(t)} = \alpha_k(\overline{\mu}_k^{(t)}) \mid 1 \leq k \leq K \tag{3}$$

$$\overline{\rho}_{i,b}^{(t)} = \rho_i(\overline{\alpha}_{k,b}^{(t)}) \quad \forall 1 \leq i \leq N, v_b \in \mathcal{V}_i \tag{4}$$

$$\mathbf{V_i}_{(b)}^{(t)} = \gamma(\mathbf{V_i}_{(b)}^{(t-1)}, \overline{\rho}_i^{(t)}) \tag{5}$$

Related Work

GRANDO; LAMB

- GRANDO; LAMB (2015) and GRANDO; LAMB (2016) uses neural networks to estimate centrality measures

GRANDO; LAMB

- GRANDO; LAMB (2015) and GRANDO; LAMB (2016) uses neural networks to estimate centrality measures
- Uses a priori knowledge of other centralities to approximate a different one.

- GRANDO; LAMB (2015) and GRANDO; LAMB (2016) uses neural networks to estimate centrality measures
- Uses a priori knowledge of other centralities to approximate a different one.
- GRANDO; LAMB (2018) also produces a ranking of the centrality measures, but again do so using the degree and eigenvector centralities as input.

- KUMAR; MEHROTRA; MOHAN (2015) uses local and global features:

- KUMAR; MEHROTRA; MOHAN (2015) uses local and global features:
    - number of vertices in a network

- KUMAR; MEHROTRA; MOHAN (2015) uses local and global features:
    - number of vertices in a network
    - number of edges in a network

- KUMAR; MEHROTRA; MOHAN (2015) uses local and global features:
  - number of vertices in a network
  - number of edges in a network
  - vertex degree

- KUMAR; MEHROTRA; MOHAN (2015) uses local and global features:
  - number of vertices in a network
  - number of edges in a network
  - vertex degree
  - sum of the degrees on vertex's neighbourhood

SCARSELLI ET AL., 2005

- SCARSELLI et al. (2005) uses GNNs to compute rankings for the PageRank centrality

SCARSELLI ET AL., 2005

- SCARSELLI et al. (2005) uses GNNs to compute rankings for the PageRank centrality
- Does not focus on other centrality measures

SCARSELLI ET AL., 2005

- SCARSELLI et al. (2005) uses GNNs to compute rankings for the PageRank centrality
- Does not focus on other centrality measures
- Does not consider the multitask transfer between centralities.

Experimental Setup and Results

RESEARCH HYPOTHESIS

1. Can a neural network infer a vertex's centrality value only from the network structure?

RESEARCH HYPOTHESIS

1. Can a neural network infer a vertex's centrality value only from the network structure?
2. Can a neural network learn an internal representation that translates into a vertex's centrality in a graph?

RESEARCH HYPOTHESIS

1. Can a neural network infer a vertex's centrality value only from the network structure?
2. Can a neural network learn an internal representation that translates into a vertex's centrality in a graph?
3. Can the representation from such a network benefit from the correlations between centrality measures and hold information about multiple centrality measures?

RESEARCH HYPOTHESIS

1 Can a neural network infer a vertex's centrality value only from the network structure?
2 Can a neural network learn an internal representation that translates into a vertex's centrality in a graph?
3 Can the representation from such a network benefit from the correlations between centrality measures and hold information about multiple centrality measures?
4 Will the algorithm learned by this neural network be scalable and be able to run for more iterations?

RESEARCH HYPOTHESIS

1. Can a neural network infer a vertex's centrality value only from the network structure?
2. Can a neural network learn an internal representation that translates into a vertex's centrality in a graph?
3. Can the representation from such a network benefit from the correlations between centrality measures and hold information about multiple centrality measures?
4. Will the algorithm learned by this neural network be scalable and be able to run for more iterations?
5. Will the algorithm learned by this neural network behave correctly for graphs larger than the ones it was trained?

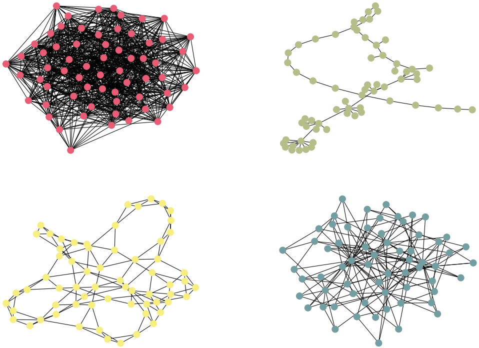

Figure: Examples of training instances with $n = 64$ vertices for each graph distribution, clockwise from the top left: Erdős-Rényi in red, Random power law tree in green, Holme-Kim in blue and Watts-Strogatz in yellow. Source: Author

Dataset	Size Range	Instances per Graph Type
Train	32-128	4096
Test	32-128	4096
Large	128-256	64
Different	32-128	256
Sizes	32-256	$256 \cdot 15$ †
Real	1174-4036	$1*$

Table: Dataset names and sizes. Source: Author.

PARAMETERS FOR GRAPH GENERATION

Graph Distribution	Parameters	Dataset
Erdős-Rényi	$p = 0.25$	Train, Test, Large, Sizes
Random power law tree	$\gamma = 3$	Train, Test, Large, Sizes
Watts-Strogatz	$k = 4, p = 0.25$	Train, Test, Large, Sizes
Holme-Kim	$m = 4, p = 0.1$	Train, Test, Large, Sizes
Circular Shell	$p_{inter} = 0.25, p_{intra} = 0.1$	Different
Barabási-Albert	$m \in \mathcal{U}(2, 5)$	Different

Table: Training instances generation parameters. Source: Author.

Name	Source	Vertices	Edges	Degree Maximum	Degree Average	Degree Minimum
power-eris1176	NR	1174	9861	100	16.8	2
econ-mahindas	NR	1258	7619	206	12.1	2
socfb-haverford76	NR	1446	59590	374	82.4	1
ego-Facebook	SN	4036	88243	1044	43.7	1
bio-SC-GT	NR	1708	33982	549	39.8	1
ca-GrQc	SN	4158	13428	81	6.46	1

Table: Statistics for the real instances and their source, where NR stands for (ROSSI; AHMED, 2015) and SN for (LESKOVEC; KREVL, 2014) Source: Author with data from (ROSSI; AHMED, 2015; LESKOVEC; KREVL, 2014).

- We will always consider a multitasking approach and a non-multitasking

MULTITASKING

- We will always consider a multitasking approach and a non-multitasking
- When not multitasking, each centrality will have a separate model

MULTITASKING

- We will always consider a multitasking approach and a non-multitasking
- When not multitasking, each centrality will have a separate model
- When multitasking, each centrality will share the same TGN block, but will learn different output functions for each centrality

APPROXIMATING CENTRALITY MEASURES

- Used the TGN model to approximate the centralities

- Used the TGN model to approximate the centralities
- Three possible setups were considered:

- Used the TGN model to approximate the centralities
- Three possible setups were considered:
  - C for the pure centrality value

- Used the TGN model to approximate the centralities
- Three possible setups were considered:

C for the pure centrality value
CN1 for the normalised centrality value

- Used the TGN model to approximate the centralities
- Three possible setups were considered:

C for the pure centrality value

CN1 for the normalised centrality value

CN2 for normalised centrality, with normalisation on the model's output

		C	CN1	CN2
"train"	Betweenness(R)	92.7%	119%	$\mathbf{89.6}\%$
	Closeness(R)	77.8%	16.3%	$\mathbf{15.3}\%$
	Degree(R)	55.5%	$\mathbf{38.9}\%$	43.6%
	Eigenvector(A)	0.0438	0.0251	$\mathbf{0.0230}$
"large"	Betweenness(R)	$\mathbf{91.7}\%$	419%	94.2%
	Closeness(R)	274%	85.9%	$\mathbf{75.0}\%$
	Degree(R)	58.9%	210%	$\mathbf{50.6}\%$
	Eigenvector(A)	0.0569	$\mathbf{0.0518}$	0.0734
$\frac{\text{"large"}}{\text{"train"}}$	Betweenness	$\mathbf{0.989}$	3.52	1.05
	Closeness	$\mathbf{3.52}$	5.26	4.90
	Degree	$\mathbf{1.06}$	5.40	1.16
	Eigenvector	$\mathbf{1.3}$	2.06	3.19

Table: Errors (Relative/Absolute) of the multitask learning performance for the proposed models on a sample of the "train" dataset and on the full "large". The best values are in **bold**. Source: Author

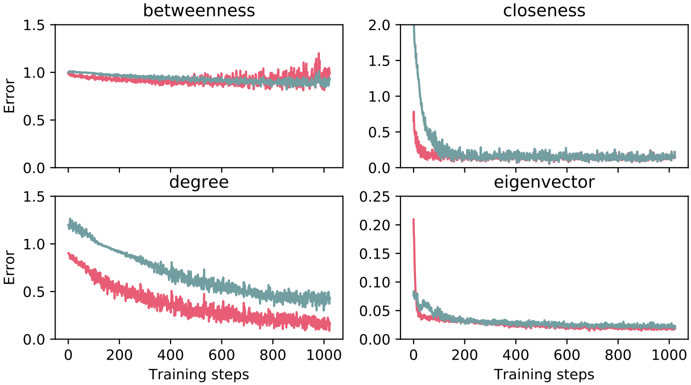

Figure: Training Performance for the CN2 model, with multitasking and without multitasking. Source: Author

RESULTS – TEST

Error Type	Centrality	"test"
Relative (%)	Betweenness	95.96/**89.54**
	Closeness	13.49/**13.38**
	Degree	**16.75**/43.39
	Average	**42.07**/48.77
Absolute	Eigenvector	**0.01946**/0.02286
MSE	Betweenness	**0.01462**/0.01464
	Closeness	0.004785/**0.003710**
	Degree	**0.03465**/0.03705
	Eigenvector	0.01694/**0.008880**
	Average	0.01775/**0.01607**

Table: Loss (MSE) and performance metrics (Relative/Absolute error) for the CN2 model on the "test" dataset (without/with multitasking). The best values are in **bold**. Source: Author

Error Type	Centrality	"large"
Relative (%)	Betweenness	**91.03**/94.17
	Closeness	79.00/**74.76**
	Degree	**27.88**/50.06
	Average	**65.97**/72.99
Absolute	Eigenvector	0.08311/**0.07214**
MSE	Betweenness	**0.01462**/0.01464
	Closeness	0.004785/**0.003710**
	Degree	**0.03465**/0.03705
	Eigenvector	0.01694/**0.008880**
	Average	0.01775/**0.01607**

Table: Loss (MSE) and performance metrics (Relative/Absolute error) for the CN2 model on the "large" dataset (without/with multitasking). The best values are in **bold**. Source: Author

RESULTS – DIFFERENT

Error Type	Centrality	"different"
Relative (%)	Betweenness	99.94/**83.88**
	Closeness	**19.13**/21.35
	Degree	**25.84**/45.32
	Average	**48.30**/50.18
Absolute	Eigenvector	**0.04854**/0.05282
MSE	Betweenness	**0.001376**/0.001445
	Closeness	**0.008956**/0.01079
	Degree	**0.01026**/0.01758
	Eigenvector	**0.003974**/0.004940
	Average	**0.006142**/0.008689

Table: Loss (MSE) and performance metrics (Relative/Absolute error) for the CN2 model on the "different" dataset (without/with multitasking). The best values are in **bold**. Source: Author

COMPARING WITH RELATED WORK

- Results were unsatisfactory.

COMPARING WITH RELATED WORK

- Results were unsatisfactory.
- REPTrees and simpler neural models in GRANDO; LAMB (2015) obtained better relative errors.

COMPARING WITH RELATED WORK

- Results were unsatisfactory.
- REPTrees and simpler neural models in GRANDO; LAMB (2015) obtained better relative errors.
- However, CN2 model performed better in minimising the Mean Squared Error

COMPARING WITH RELATED WORK

- Results were unsatisfactory.
- REPTrees and simpler neural models in GRANDO; LAMB (2015) obtained better relative errors.
- However, CN2 model performed better in minimising the Mean Squared Error
- Maybe further pre-processing could improve performance

COMPARING WITH RELATED WORK

- Results were unsatisfactory.
- REPTrees and simpler neural models in GRANDO; LAMB (2015) obtained better relative errors.
- However, CN2 model performed better in minimising the Mean Squared Error
- Maybe further pre-processing could improve performance
- However, the task here is significantly harder

- Results were unsatisfactory.
- REPTrees and simpler neural models in GRANDO; LAMB (2015) obtained better relative errors.
- However, CN2 model performed better in minimising the Mean Squared Error
- Maybe further pre-processing could improve performance
- However, the task here is significantly harder
- (GRANDO; LAMB, 2016; GRANDO; GRANVILLE; LAMB, 2018; GRANDO; LAMB, 2018) focused on ranking the centrality measures.

- Turning to producing rankings for each centrality measure.

- Turning to producing rankings for each centrality measure.
- Considered a comparison matrix as a form of producing a ranking

COMPARISON MATRIX

$$\begin{pmatrix} P(v_1 >_c v_1) & P(v_2 >_c v_1) & P(v_3 >_c v_1) \\ P(v_1 >_c v_2) & P(v_2 >_c v_2) & P(v_3 >_c v_2) \\ P(v_1 >_c v_3) & P(v_2 >_c v_3) & P(v_3 >_c v_3) \end{pmatrix} \qquad \begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}$$

Figure: Example of a fuzzy comparison matrix. Source: Altered from (AVELAR et al., 2018)

- Also allows one to consider two possible setups

- Also allows one to consider two possible setups
  - RC Which is trained as the CN2 model, but with the performance based on the comparisons

- Also allows one to consider two possible setups
  - RC Which is trained as the CN2 model, but with the performance based on the comparisons
  - RN Which is a model that computes the comparisons “natively”

RESULTS FOR THE APPROXIMATION METHOD

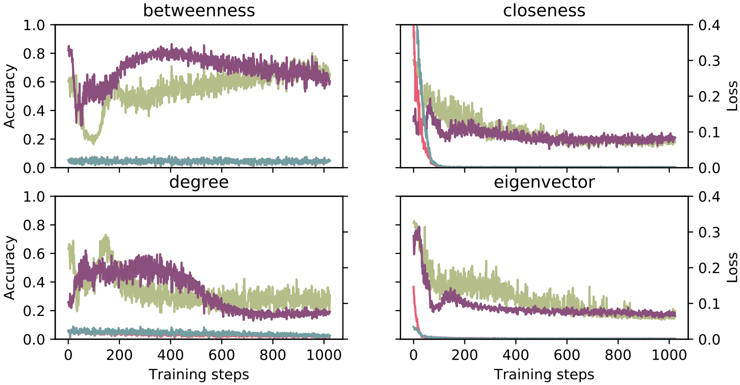

Figure: Loss plotted in red and blue and accuracy in green and purple for training without and with multitasking, respectively. Source: Author

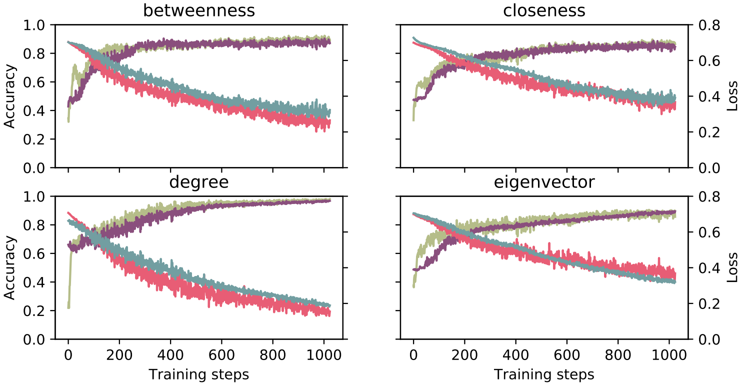

Figure: Loss plotted in red and blue and accuracy in green and purple for training without and with multitasking, respectively. Source: Author

RESULTS – TEST

Model	Centrality	P (%)	R (%)	TN (%)	Acc (%)
RN	Betweenness	**90.26**/87.18	**88.52**/87.24	**90.99**/88.89	**89.83**/87.94
	Closeness	**88.39**/86.88	**84.54**/81.96	**89.75**/88.72	**87.30**/85.52
	Degree	**99.27**/98.31	**94.94**/92.41	**99.44**/98.98	**97.64**/96.38
	Eigenvector	86.24/**89.80**	**90.18**/88.26	82.32/**90.41**	86.28/**89.40**
	Average	**91.04**/90.54	**89.55**/87.47	90.62/**91.75**	**90.26**/89.91
RC	Betweenness	64.04/58.60	64.58/59.25	71.01/65.54	68.73/63.29
	Closeness	13.36/15.84	13.56/16.06	19.77/22.01	16.89/19.25
	Degree	13.23/03.87	14.23/05.30	33.77/24.60	27.16/17.78
	Eigenvector	17.07/14.09	17.07/14.09	21.43/18.58	19.38/16.46
	Average	26.93/23.10	27.36/23.67	36.50/32.68	33.04/29.20

Table: Performance metrics (Precision, Recall, True Negative rate, Accuracy) for both models on the "test" dataset (without/with multitasking). The best values are in **bold**. Source: Author

RESULTS – LARGE

Model	Centrality	P (%)	R (%)	TN (%)	Acc (%)
RN	Betweenness	78.67/**85.12**	**87.39**/75.87	69.44/**88.10**	78.35/**81.83**
	Closeness	63.32/**69.95**	**61.19**/58.36	**89.04**/88.94	**75.60**/74.12
	Degree	**77.16**/76.82	**73.44**/70.85	**99.82**/98.80	**87.42**/85.85
	Eigenvector	70.96/**77.56**	67.66/**87.16**	**89.76**/65.91	**78.79**/76.54
	Average	72.53/**77.36**	72.42/**73.06**	**87.01**/85.44	**80.04**/79.59
RC	Betweenness	64.89/59.87	65.14/60.59	71.45/66.01	69.18/64.09
	Closeness	13.60/15.65	13.61/15.66	16.42/18.35	15.06/17.05
	Degree	24.51/26.58	25.38/28.95	43.82/45.09	37.93/39.90
	Eigenvector	14.78/15.54	14.76/15.53	16.63/17.46	15.72/16.52
	Average	29.87/29.41	29.72/30.18	37.08/36.73	34.48/34.39

Table: Performance metrics (Precision, Recall, True Negative rate, Accuracy) for both models on the "large" dataset (without/with multitasking). The best values are in **bold**. Source: Author

RESULTS – DIFFERENT

Centrality	P (%)	R (%)	TN (%)	Acc (%)
Betweenness	**81.21**/77.92	**77.47**/77.01	**81.82**/78.45	**79.71**/77.75
Closeness	**81.66**/79.57	75.25/**77.47**	**84.22**/81.45	**79.88**/79.52
Degree	86.38/**87.44**	72.46/**74.88**	89.04/**90.98**	82.08/**83.97**
Eigenvector	**84.95**/79.59	**87.95**/80.54	**83.80**/79.95	**85.84**/80.24
Average	**83.55**/81.13	**78.28**/77.48	**84.72**/82.71	**81.88**/80.37

Table: Performance metrics (Precision, Recall, True Negative rate, Accuracy) for the RN model on the "different" dataset (without/with multitasking). The best values are in **bold**. Source: Author

RESULTS – REAL

Centrality	Accuracy (%) PowEris	EconMah	SocHav	SC-GT	GrQc	EGO	Average
Betweenness	64/**66**	77/**81**	84/**85**	**84**/83	**81**/75	**77**/75	**78**/66
Closeness	**71**/65	81/**83**	60/**74**	77/**80**	62/**68**	**64**/58	**69**/61
Degree	78/**82**	**86**/83	67/**73**	80/**80**	82/**84**	**74**/72	**78**/68
Eigenvector	**67**/63	**73**/73	**87**/69	**86**/79	62/**64**	**66**/57	**74**/58
Average	**70**/69	79/**80**	74/**75**	**82**/81	72/**73**	**70**/65	**75**/74

Table: Accuracy for the RN model on the “real” dataset (without/with multitasking). The best values are in **bold**. Source: Author

RESULTS – VARYING SIZES MULTITASKING

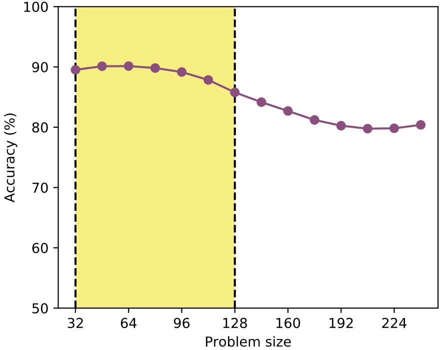

Figure: Overall accuracy multitasking RN model by number of vertices Source: Author

RESULTS – VARYING SIZES NON-MULTITASKING I

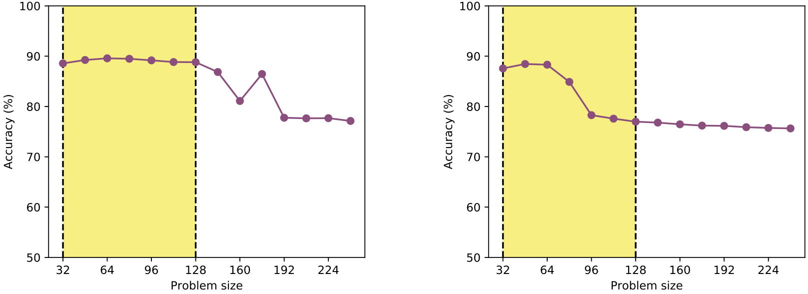

Figure: Overall accuracy non-multitasking RN model by number of vertices. Betweenness (left) and closeness. Source: Author

RESULTS – VARYING SIZES NON-MULTITASKING II

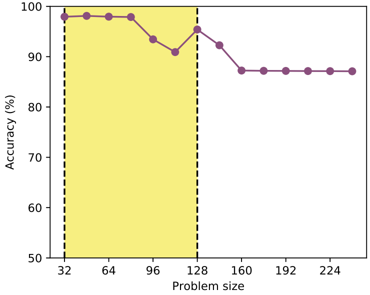

Figure: Overall accuracy non-multitasking RN model by number of vertices. Degree (left) and eigenvector. Source: Author

- Used Principal Component Analysis (PCA) on the Embeddings

- Used Principal Component Analysis (PCA) on the Embeddings
- It was expected that the PCA had a good visual fit with the centrality in question

INTERPRETING THE INTERNAL REPRESENTATION

- Used Principal Component Analysis (PCA) on the Embeddings
- It was expected that the PCA had a good visual fit with the centrality in question
- Many different behaviours were observed, both good and bad

GOOD VISUAL FIT - GOOD ACCURACY

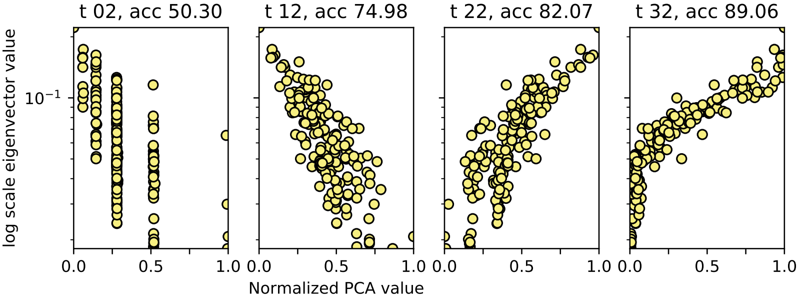

Figure: 1D PCA of a non-multitasking model for the eigenvector centrality of an Watts-Strogatz Small World graph on the “large” dataset. Source: Author

GOOD VISUAL FIT - BAD ACCURACY

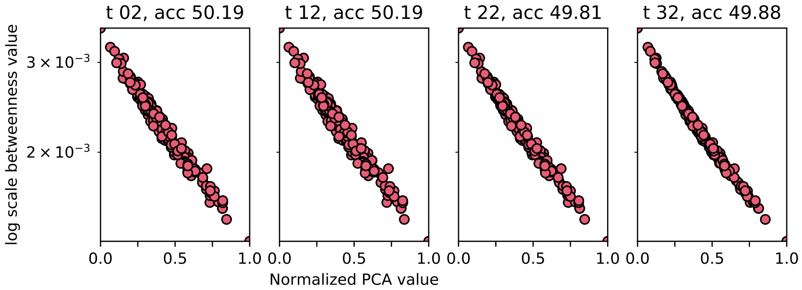

Figure: 1D PCA of a non-multitasking model for the betweenness centrality of an Erdös-Renyi graph on the "large" dataset. Source: Author

BAD VISUAL FIT - GOOD ACCURACY

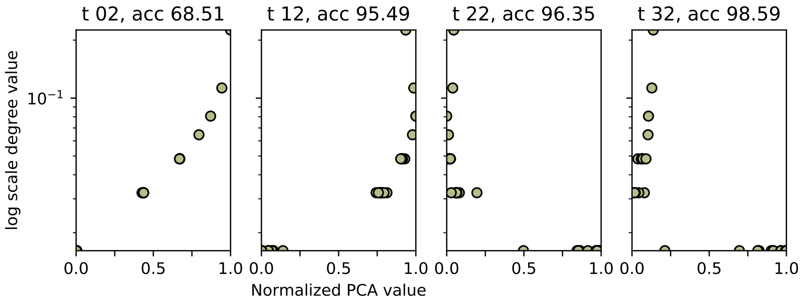

Figure: 1D PCA of a multitasking model for the degree centrality of a Powerlaw-Tree graph on the “small” dataset. Source: Author

BAD VISUAL FIT - BAD ACCURACY

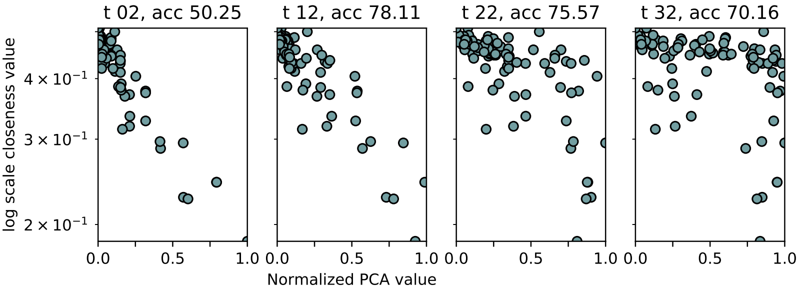

Figure: 1D PCA of a multitasking model for the closeness centrality of a Shell graph on the “different” dataset. Source: Author

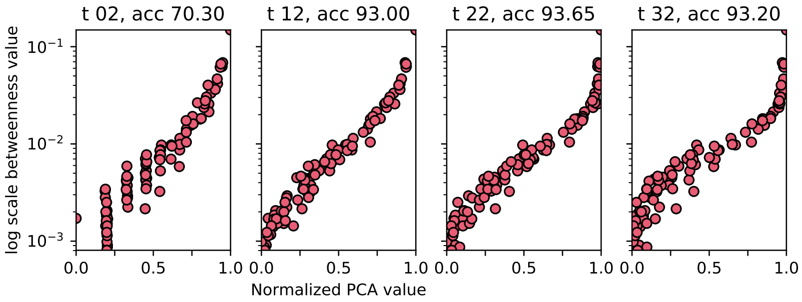

Figure: 1D PCA of a non-multitasking model for the betweenness centrality of an Barabási-Albert graph on the "different". Source: Author

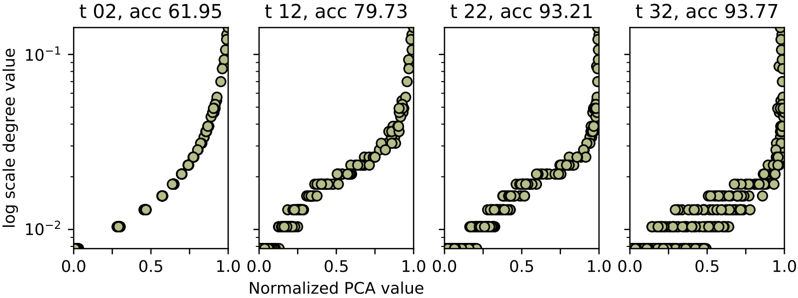

Figure: 1D PCA of a non-multitasking model for the degree centrality of a Holme-Kim graph on the "large". Source: Author

GENERALISING WITH MORE ITERATIONS

- Does the model improve its performance when ran for more iterations?

GENERALISING WITH MORE ITERATIONS

- Does the model improve its performance when ran for more iterations?
- Again, many behaviours on the PCA

GENERALISING WITH MORE ITERATIONS

- Does the model improve its performance when ran for more iterations?
- Again, many behaviours on the PCA
- In the end, the model generally had worse performance than when ran for the usual 32 iterations

GENERALISING WITH MORE ITERATIONS

- Does the model improve its performance when ran for more iterations?
- Again, many behaviours on the PCA
- In the end, the model generally had worse performance than when ran for the usual 32 iterations
- One possibility for this might be the lack of an adversarial training strategy, as done by SELSAM et al. (2018)

STABLE PCA WITH MORE ITERATIONS

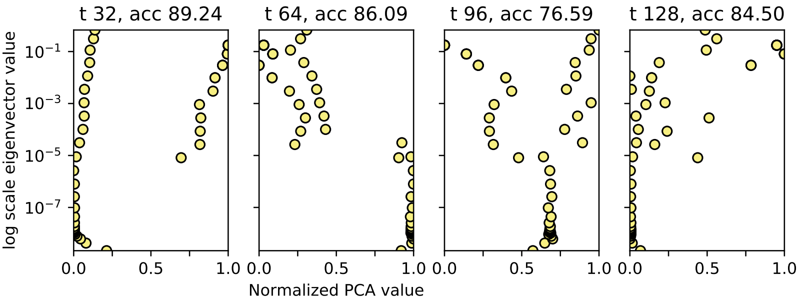

Figure: 1D PCA of a multitasking model for the eigenvector centrality of a Powerlaw Tree graph on the “small”. Source: Author

UNSTABLE PCA WITH MORE ITERATIONS

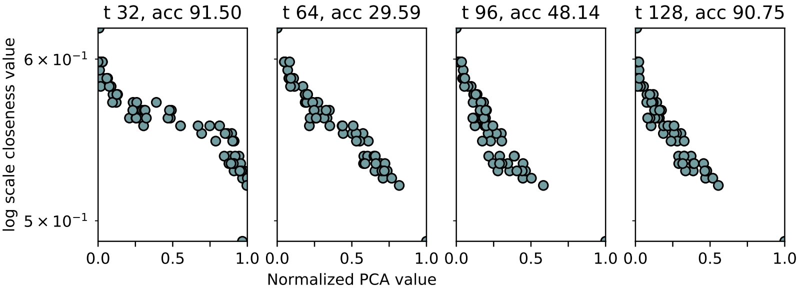

Figure: 1D PCA of a multitasking model for the closeness centrality of an Erdös-Renyi graph on the "small". Source: Author

Conclusions

- Analysis of the performance of a GNN model for 4 different centralities.

- Analysis of the performance of a GNN model for 4 different centralities.
- Analysis of the performance when multitasking.

CONTRIBUTIONS

- Analysis of the performance of a GNN model for 4 different centralities.
- Analysis of the performance when multitasking.
- Analysis of interpretability of the model.

- Analysis of the performance of a GNN model for 4 different centralities.
- Analysis of the performance when multitasking.
- Analysis of interpretability of the model.
- Comparison framework for generating rankings natively.

CONCLUSIONS I

1. Can a neural network infer a vertex's centrality value only from the network structure?

CONCLUSIONS I

1 Can a neural network infer a vertex's centrality value only from the network structure?

- In short – Yes

CONCLUSIONS I

1. Can a neural network infer a vertex's centrality value only from the network structure?
   - In short – Yes
   - Model had worse performance in predicting the centrality than with purely numeric information about the node.

1. Can a neural network infer a vertex's centrality value only from the network structure?
   - In short – Yes
   - Model had worse performance in predicting the centrality than with purely numeric information about the node.
   - However, the task at hand is more difficult

CONCLUSIONS I

1. Can a neural network infer a vertex's centrality value only from the network structure?
   - In short – Yes
   - Model had worse performance in predicting the centrality than with purely numeric information about the node.
   - However, the task at hand is more difficult
   - Model could rank with higher accuracy

CONCLUSIONS I

1. Can a neural network infer a vertex's centrality value only from the network structure?
   - In short – Yes
   - Model had worse performance in predicting the centrality than with purely numeric information about the node.
   - However, the task at hand is more difficult
   - Model could rank with higher accuracy
2. Can a neural network learn an internal representation that translates into a vertex's centrality in a graph?

CONCLUSIONS I

1. Can a neural network infer a vertex's centrality value only from the network structure?
   - In short – Yes
   - Model had worse performance in predicting the centrality than with purely numeric information about the node.
   - However, the task at hand is more difficult
   - Model could rank with higher accuracy
2. Can a neural network learn an internal representation that translates into a vertex's centrality in a graph?
   - Yes

CONCLUSIONS I

1. Can a neural network infer a vertex's centrality value only from the network structure?
   - In short – Yes
   - Model had worse performance in predicting the centrality than with purely numeric information about the node.
   - However, the task at hand is more difficult
   - Model could rank with higher accuracy
2. Can a neural network learn an internal representation that translates into a vertex's centrality in a graph?
   - Yes
   - PCA seemed to show visual correlation with centralities

CONCLUSIONS I

1. Can a neural network infer a vertex's centrality value only from the network structure?
   - In short – Yes
   - Model had worse performance in predicting the centrality than with purely numeric information about the node.
   - However, the task at hand is more difficult
   - Model could rank with higher accuracy
2. Can a neural network learn an internal representation that translates into a vertex's centrality in a graph?
   - Yes
   - PCA seemed to show visual correlation with centralities
3. Can the representation from such a network benefit from the correlations between centrality measures and hold information about multiple centrality measures?

CONCLUSIONS I

1. Can a neural network infer a vertex's centrality value only from the network structure?
   - In short – Yes
   - Model had worse performance in predicting the centrality than with purely numeric information about the node.
   - However, the task at hand is more difficult
   - Model could rank with higher accuracy
2. Can a neural network learn an internal representation that translates into a vertex's centrality in a graph?
   - Yes
   - PCA seemed to show visual correlation with centralities
3. Can the representation from such a network benefit from the correlations between centrality measures and hold information about multiple centrality measures?
   - Apparently it didn't

CONCLUSIONS I

1. Can a neural network infer a vertex's centrality value only from the network structure?
   - In short – Yes
   - Model had worse performance in predicting the centrality than with purely numeric information about the node.
   - However, the task at hand is more difficult
   - Model could rank with higher accuracy
2. Can a neural network learn an internal representation that translates into a vertex's centrality in a graph?
   - Yes
   - PCA seemed to show visual correlation with centralities
3. Can the representation from such a network benefit from the correlations between centrality measures and hold information about multiple centrality measures?
   - Apparently it didn't
   - The performance didn't drop significantly

CONCLUSIONS I

1. Can a neural network infer a vertex's centrality value only from the network structure?
    - In short – Yes
    - Model had worse performance in predicting the centrality than with purely numeric information about the node.
    - However, the task at hand is more difficult
    - Model could rank with higher accuracy
2. Can a neural network learn an internal representation that translates into a vertex's centrality in a graph?
    - Yes
    - PCA seemed to show visual correlation with centralities
3. Can the representation from such a network benefit from the correlations between centrality measures and hold information about multiple centrality measures?
    - Apparently it didn't
    - The performance didn't drop significantly
    - Model is lighter if many centralities are learned jointly than if ran separately

4 Will the algorithm learned by this neural network be scalable and be able to run for more iterations?

4. Will the algorithm learned by this neural network be scalable and be able to run for more iterations?
    - It didn't

4. Will the algorithm learned by this neural network be scalable and be able to run for more iterations?
    - It didn't
    - Model was unstable when ran out of its "comfort zone"

CONCLUSIONS II

4. Will the algorithm learned by this neural network be scalable and be able to run for more iterations?
   - It didn't
   - Model was unstable when ran out of its “comfort zone”
5. Will the algorithm learned by this neural network behave correctly for graphs larger than the ones it was trained?

4. Will the algorithm learned by this neural network be scalable and be able to run for more iterations?
    - It didn't
    - Model was unstable when ran out of its "comfort zone"
5. Will the algorithm learned by this neural network behave correctly for graphs larger than the ones it was trained?
    - It somehow performed well

CONCLUSIONS II

4. Will the algorithm learned by this neural network be scalable and be able to run for more iterations?
   - It didn't
   - Model was unstable when ran out of its "comfort zone"
5. Will the algorithm learned by this neural network behave correctly for graphs larger than the ones it was trained?
   - It somehow performed well
   - There was a drop in performance

CONCLUSIONS II

4 Will the algorithm learned by this neural network be scalable and be able to run for more iterations?
   - It didn't
   - Model was unstable when ran out of its "comfort zone"

5 Will the algorithm learned by this neural network behave correctly for graphs larger than the ones it was trained?
   - It somehow performed well
   - There was a drop in performance
   - Such decay was expected

CONCLUSIONS II

4. Will the algorithm learned by this neural network be scalable and be able to run for more iterations?
   - It didn't
   - Model was unstable when ran out of its "comfort zone"
5. Will the algorithm learned by this neural network behave correctly for graphs larger than the ones it was trained?
   - It somehow performed well
   - There was a drop in performance
   - Such decay was expected
   - May be due to global information and numerical problems

- More centrality measures.

FUTURE WORK

- More centrality measures.
- Deeper analysis of internal embeddings.

- More centrality measures.
- Deeper analysis of internal embeddings.
- Training with an adversarial dataset.

- More centrality measures.
- Deeper analysis of internal embeddings.
- Training with an adversarial dataset.
- Experimenting with Transfer Learning instead of Multitask Learning

AVELAR, P. H. C. et al. Multitask learning on graph neural networks - learning multiple graph centrality measures with a unified network. **CoRR**, abs/1809.07695, 2018. Available from Internet: <http://arxiv.org/abs/1809.07695>.

BARABÁSI, A.-L. et al. **Network science**. [S.l.]: Cambridge university press, 2016.

BATTAGLIA, P. W. et al. Relational inductive biases, deep learning, and graph networks. **CoRR**, abs/1806.01261, 2018. Available from Internet: <http://arxiv.org/abs/1806.01261>.

GRANDO, F.; GRANVILLE, L. Z.; LAMB, L. C. Machine learning in network centrality measures: Tutorial and outlook. **CoRR**, abs/1810.11760, 2018. Available from Internet: <http://arxiv.org/abs/1810.11760>.

GRANDO, F.; LAMB, L. C. Estimating complex networks centrality via neural networks and machine learning. In: **2015 International Joint Conference on Neural Networks, IJCNN 2015, Killarney, Ireland, July 12-17, 2015**. IEEE, 2015. p. 1–8. Available from Internet: <https://doi.org/10.1109/IJCNN.2015.7280334>.

GRANDO, F.; LAMB, L. C. On approximating networks centrality measures via neural learning algorithms. In: **2016 International Joint Conference on Neural Networks, IJCNN 2016, Vancouver, BC, Canada, July 24-29, 2016**. IEEE, 2016. p. 551–557. Available from Internet: <https://doi.org/10.1109/IJCNN.2016.7727248>.

GRANDO, F.; LAMB, L. C. Computing vertex centrality measures in massive real networks with a neural learning model. In: **2018 International Joint Conference on Neural Networks, IJCNN 2018, Rio de Janeiro, Brazil, July 8-13, 2018**. IEEE, 2018. p. 1–8. Available from Internet: <https://doi.org/10.1109/IJCNN.2018.8489690>.

GRAVES, A. et al. Hybrid computing using a neural network with dynamic external memory. **Nature**, v. 538, n. 7626, p. 471–476, 2016. Available from Internet: <https://doi.org/10.1038/nature20101>.

HAGBERG, A.; SWART, P.; CHULT, D. S. **Exploring network structure, dynamics, and function using NetworkX**. [S.l.], 2008.

HANNUN, A. Y. et al. Deep speech: Scaling up end-to-end speech recognition. **CoRR**, abs/1412.5567, 2014. Available from Internet: <http://arxiv.org/abs/1412.5567>.

KARRAS, T.; LAINE, S.; AILA, T. A style-based generator architecture for generative adversarial networks. **CoRR**, abs/1812.04948, 2018. Available from Internet: <http://arxiv.org/abs/1812.04948>.

KUMAR, A.; MEHROTRA, K. G.; MOHAN, C. K. Neural networks for fast estimation of social network centrality measures. In: SPRINGER. **Proceedings of the Fifth International Conference on Fuzzy and Neuro Computing (FANCCO-2015)**. [S.l.], 2015. p. 175–184.

LESKOVEC, J.; KREVL, A. **SNAP Datasets: Stanford Large Network Dataset Collection**. 2014. <http://snap.stanford.edu/data>.

OLAH, C. Conv nets: A modular perspective. Jul 2014. Online; accessed 13-January-2019. Available from Internet: <https://colah.github.io/posts/2014-07-Conv-Nets-Modular/>.

OLAH, C. Understanding lstm networks. Aug 2015. Online; accessed 13-January-2019. Available from Internet: <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>.

OPENAI et al. Learning dexterous in-hand manipulation. **CoRR**, abs/1808.00177, 2018. Available from Internet: <http://arxiv.org/abs/1808.00177>.

RADFORD, A. et al. Language models are unsupervised multitask learners. 2019. Available from Internet: <https://d4mucfpksywv.cloudfront.net/better-language-models/language_models_are_unsupervised_multitask_learners.pdf>.

ROSSI, R. A.; AHMED, N. K. The network data repository with interactive graph analytics and visualization. In: **AAAI**. [s.n.], 2015. Available from Internet: <http://networkrepository.com>.

SANTORO, A. et al. A simple neural network module for relational reasoning. In: GUYON, I. et al. (Ed.). **Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA**. [s.n.], 2017. p. 4974–4983. Available from Internet: <http://papers.nips.cc/paper/7082-a-simple-neural-network-module-for-relational-reasoning>.

SCARSELLI, F. et al. The graph neural network model. **IEEE Trans. Neural Networks**, v. 20, n. 1, p. 61–80, 2009. Available from Internet: <https://doi.org/10.1109/TNN.2008.2005605>.

SCARSELLI, F. et al. Graph neural networks for ranking web pages. In: SKOWRON, A. et al. (Ed.). **2005 IEEE / WIC / ACM International Conference on Web Intelligence (WI 2005), 19-22 September 2005, Compiegne, France**. IEEE Computer Society, 2005. p. 666–672. Available from Internet: <https://doi.org/10.1109/WI.2005.67>.

SELSAM, D. et al. Learning a SAT solver from single-bit supervision. **CoRR**, abs/1802.03685, 2018. Available from Internet: <http://arxiv.org/abs/1802.03685>.

SILVER, D. et al. Mastering the game of go with deep neural networks and tree search. **Nature**, v. 529, n. 7587, p. 484–489, 2016. Available from Internet: <https://doi.org/10.1038/nature16961>.

WAYNE, Z. W. An information flow model for conflict and fission in small groups. **Journal of anthropological research**, v. 33, n. 4, p. 452–473, 1977.

ZHAO, H. et al. The sound of pixels. In: FERRARI, V. et al. (Ed.). **Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part I**. Springer, 2018. (Lecture Notes in Computer Science, v. 11205), p. 587–604. Available from Internet: <https://doi.org/10.1007/978-3-030-01246-5_35>.

Pedro Henrique da Costa Avelar
Advisor: Luís da Cunha Lamb
Instituto de Informática — UFRGS
`pedro.avelar@inf.ufrgs.br`
`lamb@inf.ufrgs.br`

INSTITUTO
DE INFORMÁTICA
UFRGS

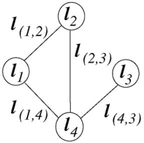

Figure: Source: SCARSELLI et al. (2009)

GNNS IN DETAIL II

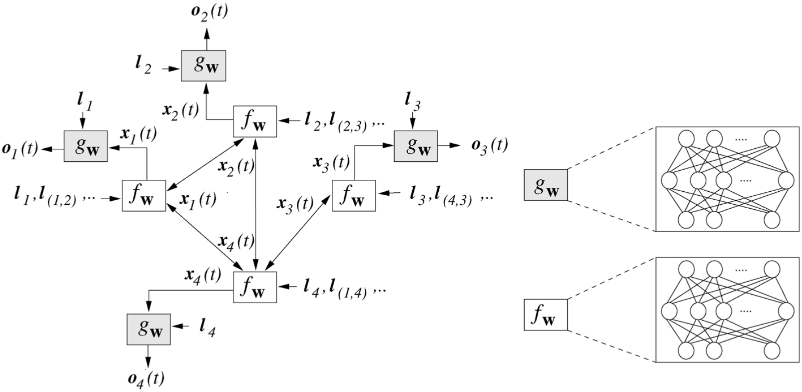

Figure: Source: SCARSELLI et al. (2009)

GNNS IN DETAIL III

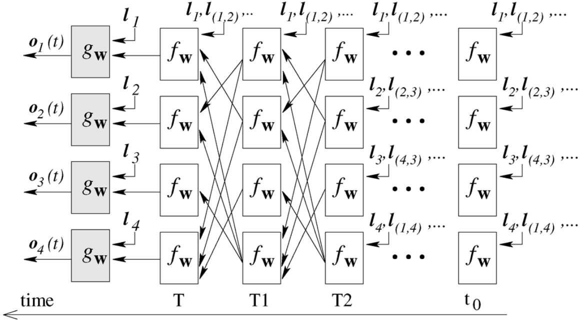

Figure: Source: SCARSELLI et al. (2009)

FITTING THE DIMENSIONALITY OF THE MODEL – APPROXIMATING

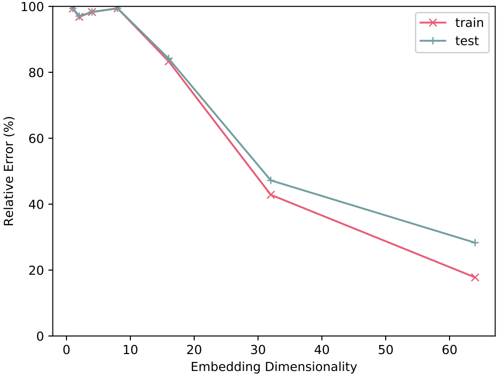

Figure: Relative Error (degree centrality) on the “train” (in red) and “large” datasets (in blue) by embedding dimensionality d for the CN2 model. Source: Author

APPROXIMATING – PROBLEM SIZE INFLUENCE ON MULTITASK MODEL

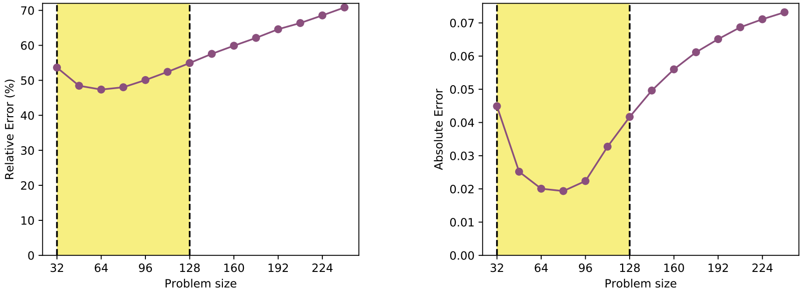

Figure: The overall relative (left) and absolute (right) error for the CN2 multitask model Source: Author

APPROXIMATING – PROBLEM SIZE INFLUENCE ON NON-MULTITASK MODEL I

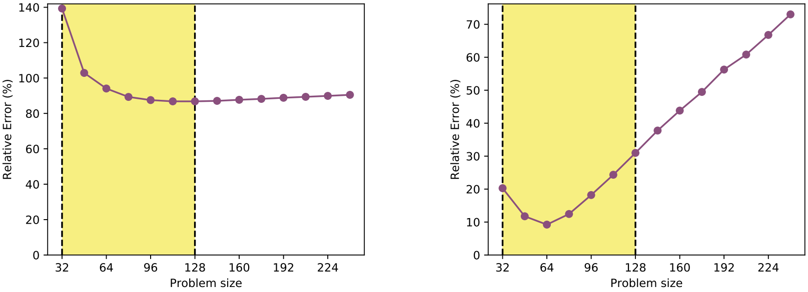

Figure: Overall relative error for the non-multitasking RN model by number of vertices. Betweenness (left) and closeness. Source: Author

APPROXIMATING – PROBLEM SIZE INFLUENCE ON NON-MULTITASK MODEL I

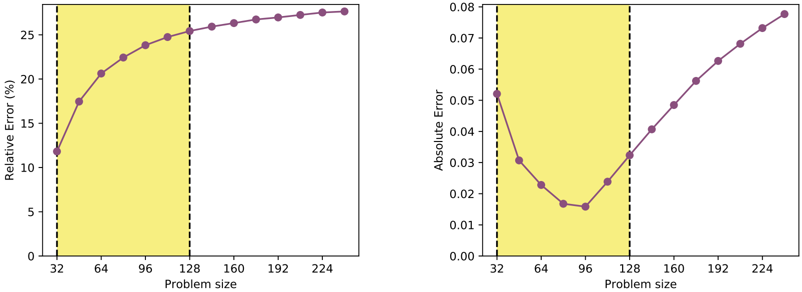

Figure: Overall relative (left) and absolute (right) error for the non-multitasking RN model by number of vertices. Degree (left) and Eigenvector. Source: Author

CENTRALITY-SPECIFIC GNN ALGORITHM

procedure $\textsc{GNN-Centrality}(\mathcal{G} = (\mathcal{V}, \mathcal{E}), \mathcal{C})$

  $\mathbf{M}[i,j] \leftarrow 1 \text{ if } (v_i, v_j) \in \mathcal{E} \text{ else } 0$

  $V^1[i,:] \leftarrow V_{init} \mid \forall v_i \in \mathcal{V}$

  for $t = 1 \ldots t_{max}$ **do**

    $\mathbf{V}^{t+1}, \mathbf{V}_h^{t+1} \leftarrow V_u(\mathbf{V}^t, \mathbf{M} \times \mathrm{src}_{\mathrm{msg}}(\mathbf{V}^t), \mathbf{M}^T \times \mathrm{tgt}_{\mathrm{msg}}(\mathbf{V}^t))$

  end for

  for $c \in \mathcal{C}$ **do**

    $\mathbf{M}_{\gtrapprox_{\mathbf{c}}}[i,j] \leftarrow \mathrm{cmp}_c(\mathbf{V}^{t_{max}}[i,:], \mathbf{V}^{t_{max}}[j,:]) \mid \forall\ v_i, v_j \in \mathcal{V}$

    $\mathbf{M}_{>_{\mathbf{c}}} \leftarrow M_{\gtrapprox_c} > \frac{1}{2}$

  end for

end procedure

FITTING THE DIMENSIONALITY OF THE MODEL – COMPARING

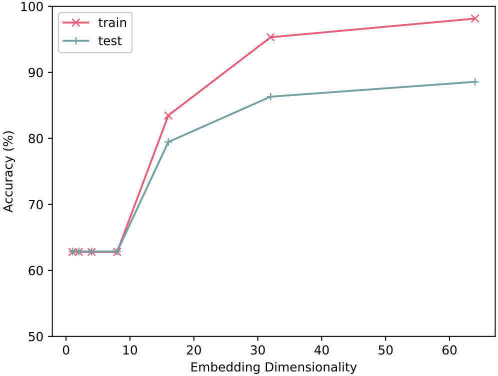

Figure: Accuracy (degree centrality) on the “train” (in red) and “large” datasets (in blue) by embedding dimensionality d for the RN model. Source: Author