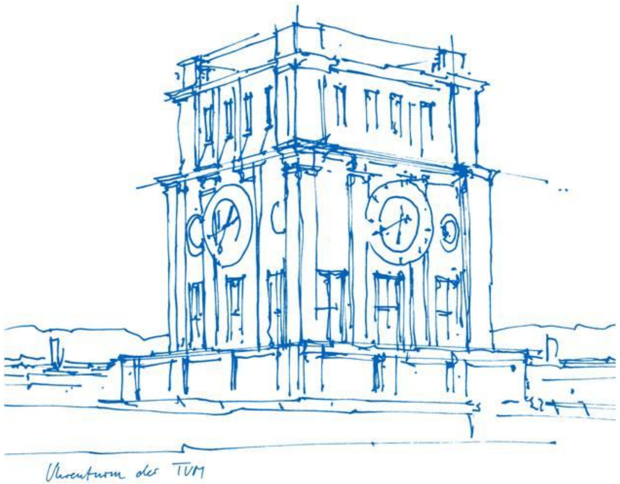
Comparing Graph Architectures for Deep Metric Learning

ADL4CV Final Presentation

09.02.2022

Moritz Schüler
moritz.schueler@tum.de

Luca Eyring
luca.eyring@tum.de



1.1 Deep Metric Learning

A

B

Learn a **similarity** function:

$$d(A, B) > \tau$$

Goal:

- similar objects are close together
- dissimilar far apart

Why not classification?

- Doesn’t scale for new classes

1.2 Research Objective

- Investigate attention mechanisms in metric learning
- Compare:
  - Jenny’s architecture based on traditional attention
  - Graph Attention Network (**GAT**) v1 & v2

- Ablation Study
  - Does Attention benefit metric learning?
  - How important are the linear layers following attention?

1.3 Datasets

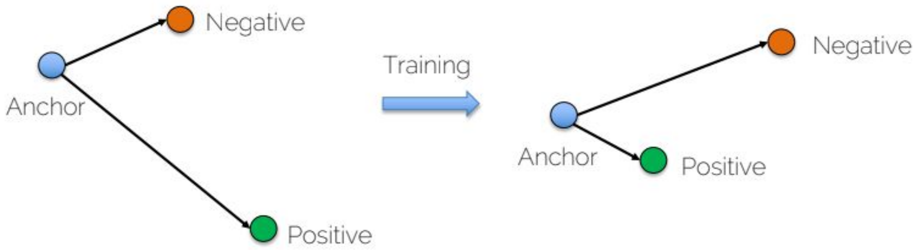
Stanford **CARS** Dataset

- 16.185 images
- 196 classes

Caltech **CUB** Dataset

- 11.788 images
- 200 classes

1.4 Traditional Deep Metric Learning

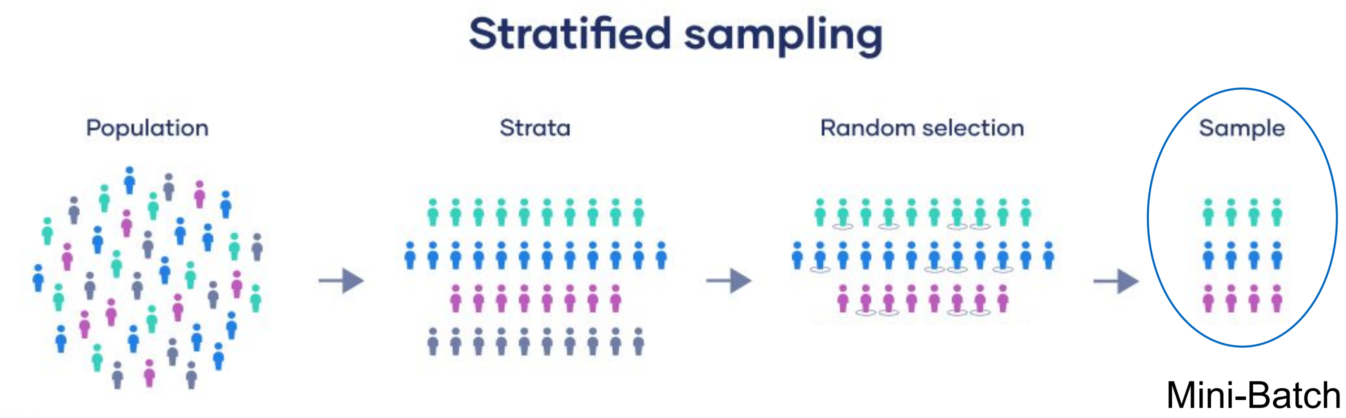


Triplet Loss

- shrink distance of similar samples
- increase distance of dissimilar samples

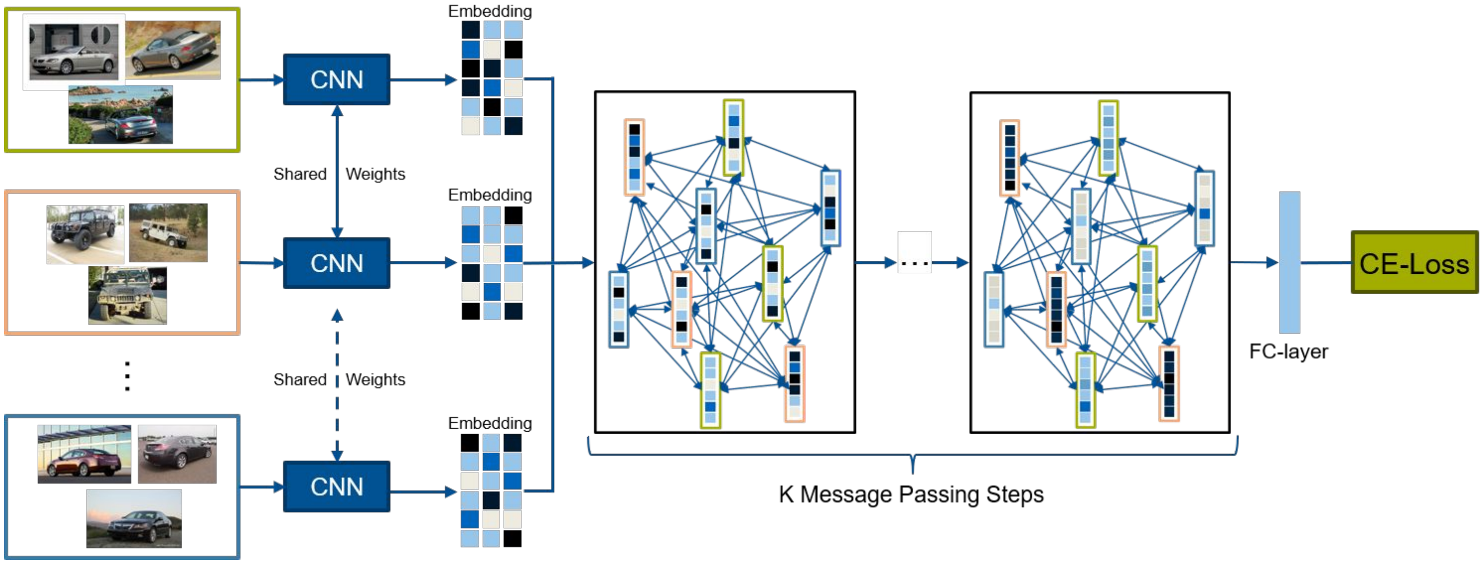
Problem: Which triplets to sample?

2.1 Sampling



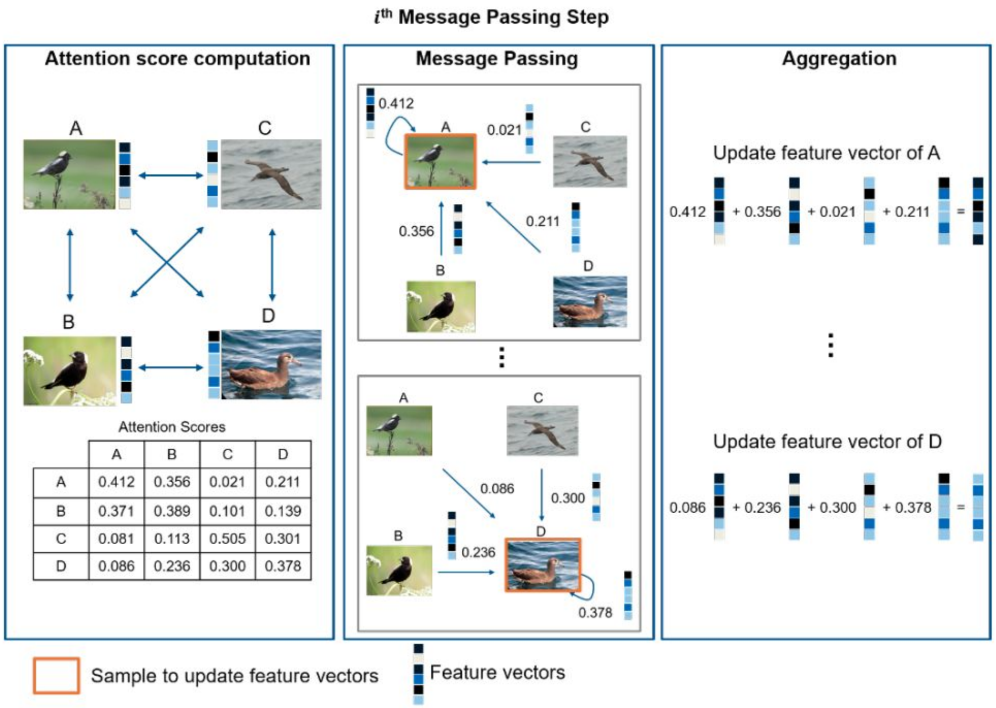
- Each class is a **Strata**
- Randomly choose X classes
- Randomly sample same amount from those classes

2.2 Jenny's Architecture



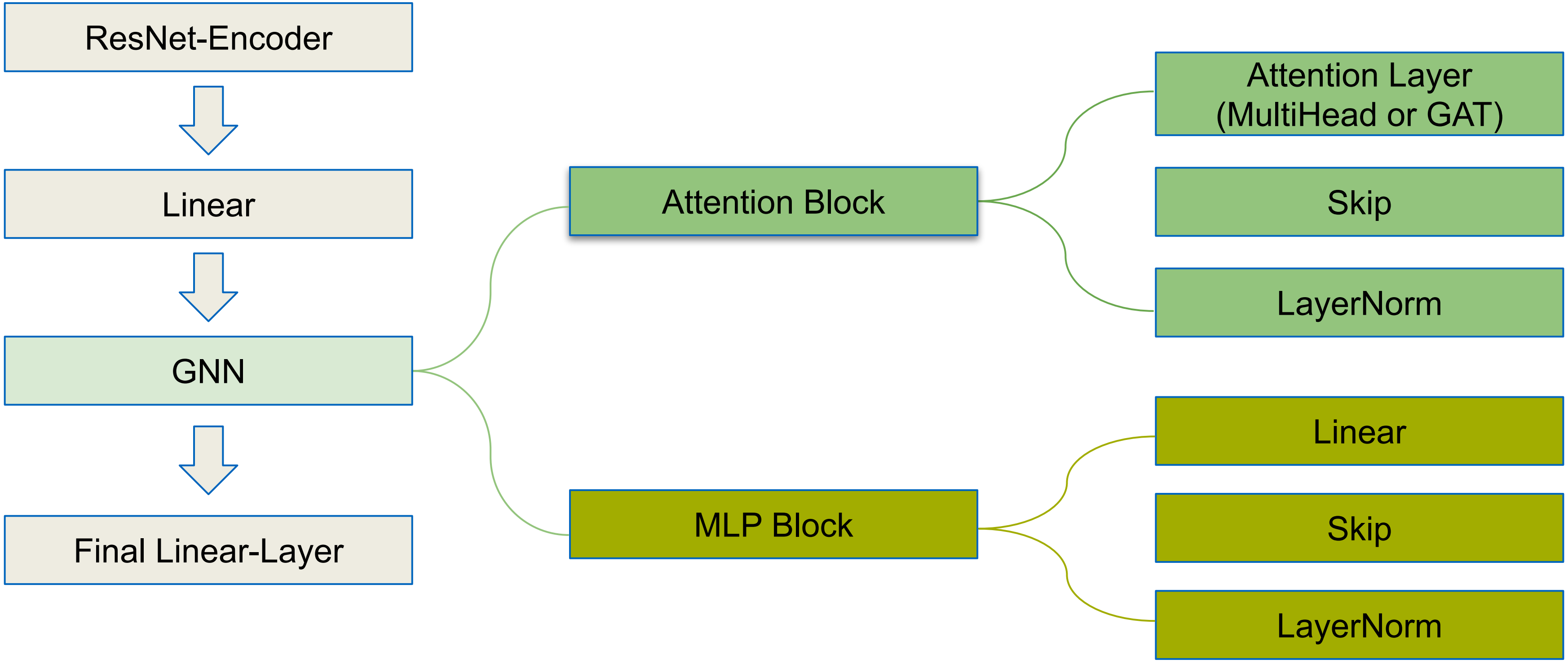
- takes whole (mini-) batch into account
- learns connection between samples

2.3 Message Passing



- Initial embeddings from Backbone
- Calculate temporal embeddings with NN (K times)
- Update node embeddings

2.4 Full Architecture in Detail



3.1 Conducted Experiments

- Train only Encoder
- Remove whole GNN
- Remove Attention
- Remove MLP
- Change Attention Mechanism

ResNet-Encoder

GNN

Attention Block

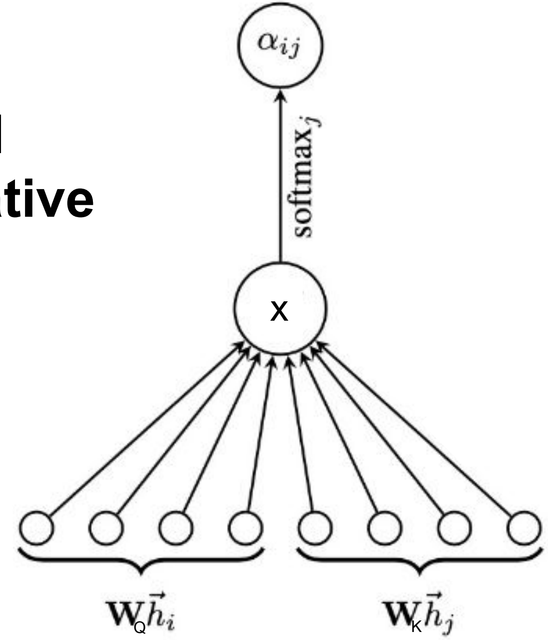
MLP Block

Attention Layer
(MultiHead or GAT)

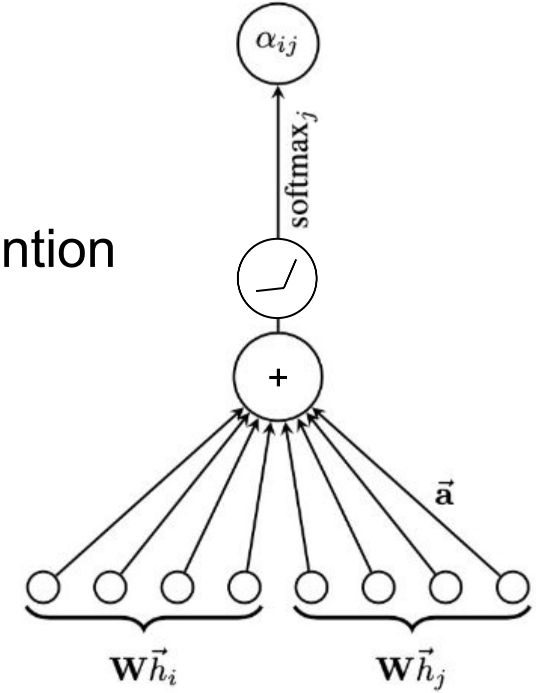
3.2 Attention Mechanisms

Attention between two nodes ***h_i*** and ***h_j***:

Traditional **Multiplicative** Attention:



Additive Attention used in GAT:



3.3 *GAT* vs *GATv2*

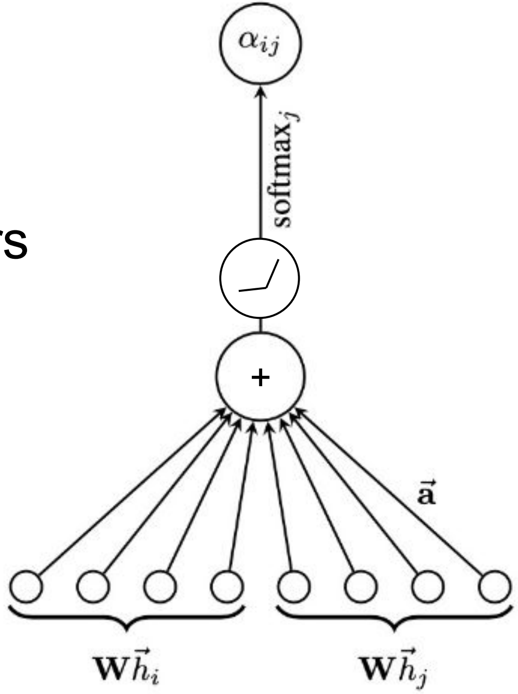
Motivation:

- *GAT* tends to compute a ***global ranking*** of “influential” nodes
- *GATv2* computes ***different rankings*** of neighbors

$$LeakyReLU(\mathbf{a}^T \cdot [\mathbf{W}\mathbf{h}_i || \mathbf{W}\mathbf{h}_j)]$$

$$\Downarrow$$

$$\mathbf{a}^T \cdot LeakyReLU(\mathbf{W}[\mathbf{h}_i || \mathbf{h}_j)]$$



3.3 Reproducibility

Results **varied** a lot across runs
=> Goal: Make everything deterministic

- Rewrote **Attention** Mechanism
- Rewrote ***GAT & GATv2*** Implementation

=> Now: *Deterministic* Training

4.1 Experiment Results on CARS

Method	*Paper*
Baseline	**87.1**
Method	**88.1**
Difference	**+1.0**

s

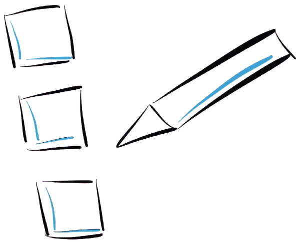
4.2 Experiment Results on CUB

Method	*Paper*	ResNet50 only	No MLP + No Att	No Attention	No MLP	GAT	**GATv2**
Baseline	**69.4**	69.4	69.4	69.4	69.4	69.4	69.4
Method	**70.3**	**67.7**	**69.1**	**69.3**	**67.9**	**68.9**	**69.8**
Difference	**+0.9**	**-1.7**	**-0.3**	**-0.1**	**-1.5**	**-0.5**	**+0.4**

4.3 Experiments over 5 Seeds

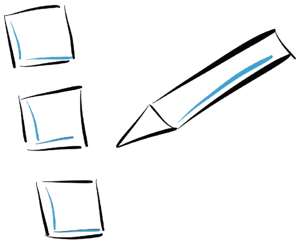
Method	*CARS*	CUB	CARS mAP	CUB mAP	CARS NMI	CUB NMI
Original	**87.48** +- 0.37	69.05 +- 1.04	25.87 +- 0.98	27.15 +- 0.74	71.56 +- 0.83	72.51 +- 0.92
GATv2	87.23 +- 0.38	**69.24** +- 0.30	**26.86** +- 0.50	**27.22** +- 0.39	**72.39** +- 0.40	**72.52** +- 0.74

5. Conclusion



- Attention *beneficial* to Metric Learning in general
- Linear Layers not necessarily needed for Attention
- GATv2 *outperforms* GAT
- GATv2 outperform traditional attention by a small margin

4. Future Work



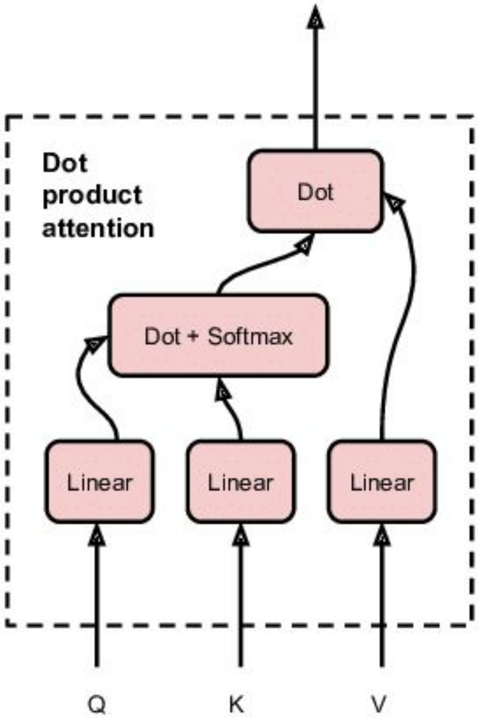
- mAP@R für Gatv2
- tSNE of the embeddings

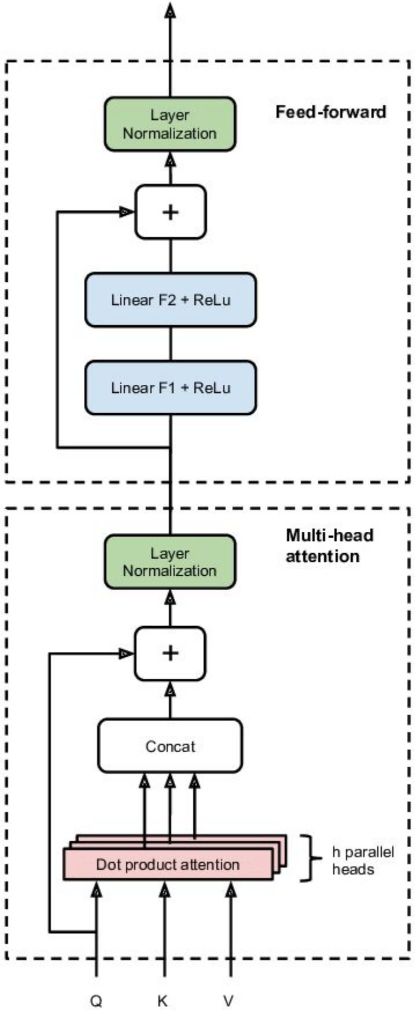
References

- [1] Seidenschwarz et al. “Learning intra-batch connections for deep metric learning”. 2021. https://arxiv.org/pdf/2102.07753.pdf
- [2] Velikovi et al. “Graph attention networks”. 2018. https://arxiv.org/pdf/1710.10903.pdf
- [3] Brody et al. “How attentive are graph attention networks?”. 2021. https://arxiv.org/pdf/2105.14491.pdf
- [4] Dong et al. “Attention is not all you need: Pure attention loses rank doubly exponentially with depth”. 2021. https://arxiv.org/pdf/2103.03404.pdf

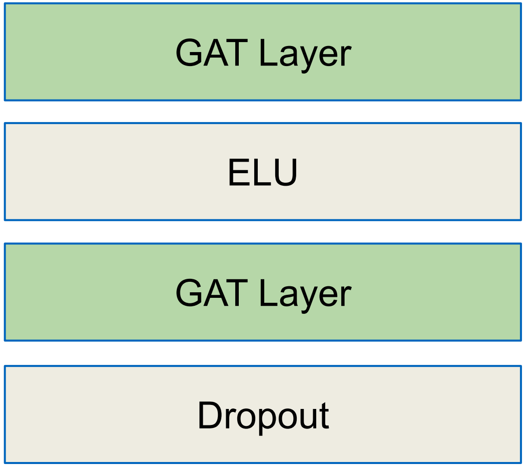
Backup

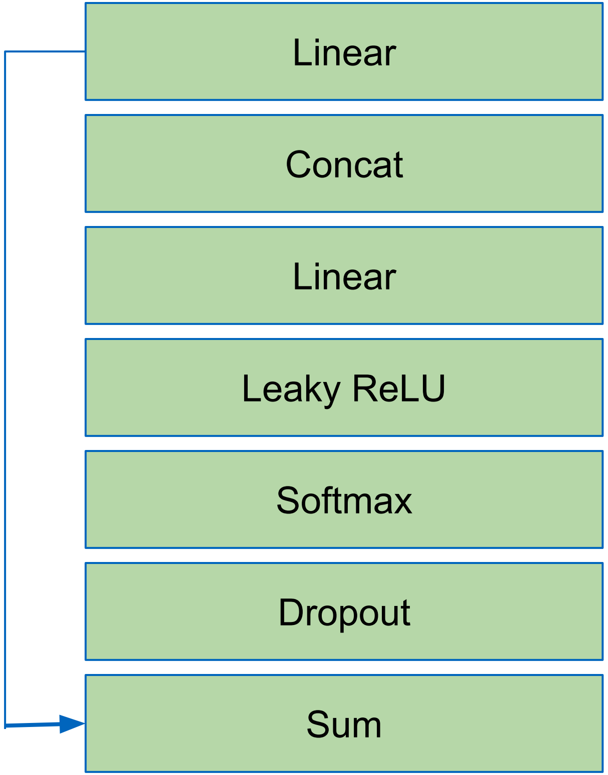
2 Multi Head Attention





2 GAT





Transformation

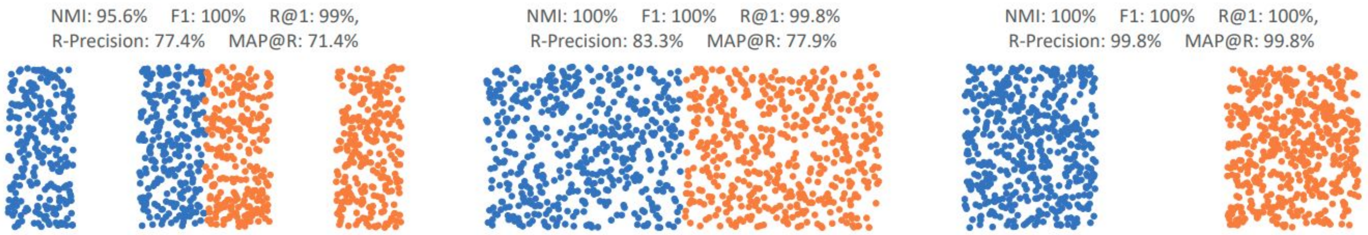
Heads are concatenated

Attention over all heads simult.

3.2 Metrics

Recall@K: 1. k-means clustering
2. Get k nearest neighbors
3. If match: score=1, if not score=0

NMI: 1. Split into clusters (1) and class labels (2)
2. How do (1) and (2) agree?
High:1 , Low: 0



nce

4.1 Message Passing formally

MP & ***Attention*** Weights:

$$\boldsymbol{h}_i^{l+1} = \sum_{j \in N_i} \alpha_{ij}^l \boldsymbol{W}^l \boldsymbol{h}_j^l$$

Attention computation:

$$e_{ij}^l = \frac{W_q^l h_i^l (W_k^l h_j^l)^T}{\sqrt{d}}$$

$$\alpha_{ij}^l = softmax_j(e_{ij}^l)$$

Residual Block:

$$f(\boldsymbol{h}_i^{l+1}) = LayerNorm(\boldsymbol{h}_i^{l+1} + \boldsymbol{h}_i^l)$$

Added ***Linear*** Block:

$$g(\boldsymbol{h}_i^{l+1}) = LayerNorm(FF(f(\boldsymbol{h}_i^{l+1})) + f(\boldsymbol{h}_i^{l+1}))$$

4.2 Graph Attention Network

Most popular framework for attentional *GNNs!*

Difference lies in the ***attention*** computation:

$$e_{ij}^{l} = \frac{W_q^l h_i^l (W_k^l h_j^l)^T}{\sqrt{d}}$$

$$e\left(\boldsymbol{h}_i, \boldsymbol{h}_j\right) = \text{LeakyReLU}\left(\boldsymbol{a}^{\top} \cdot \left[\boldsymbol{W}\boldsymbol{h}_i \| \boldsymbol{W}\boldsymbol{h}_j\right]\right)$$

5.1 First Experiments

- Reproducing Paper
- GAT
- GATv2

Measure the impact of *attention* & *linear* block:

- Remove *linear* Block: $$g(h_i^{l+1}) = f(h_i^{l+1})$$

- Remove *attention*: $$f(h_i^{l+1}) = f(h_i^{l})$$

6. Challenges and Next Steps

- Challenges
  - Setting up Google Colab
  - Getting Google Cloud credits
  - Running experiments (takes a lot of time)
  - Reproducibility (Training still non-deterministic)

- Next Steps
  - Run hyperparameter tuning to improve results
  - Graph construction