Neural Network to Graph

윤 현 석

Vision & Display Systems Lab.

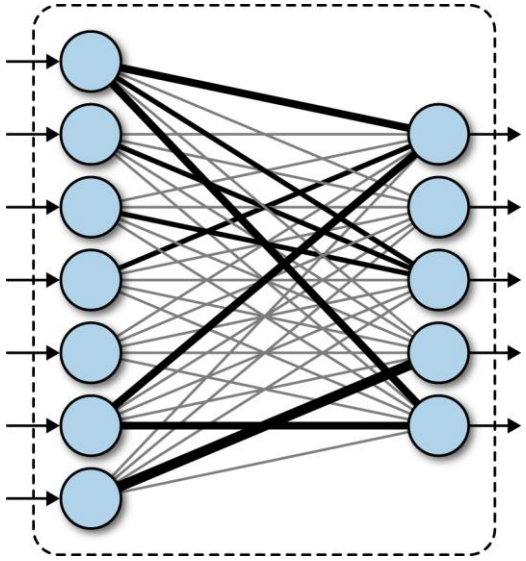
Dept. of Electronic Engineering, Sogang University

Outline

- Introduction
- Neural Networks as Relational Graphs
- Exploring and Generating Relational Graphs
- Experiments
- Discussion and Conclusion

Introduction

- 저자의 연구 이유
  - Neural network의 구조에 따라 performance의 차이가 심함은 많이 알려져 있음
  - Neural network는 많은 경우 graph로 나타내어지며, 이러한 특징이 크게 드러남
  - 그러나 neural network를 형성하는 graph 구조에 따른 정확도에 대한 이해 부족
  - 현재 connectivity에 대한 연구가 활발히 이루어 지고 있음



Graph 형태로 나타낸 neural network의 대표적인 예시

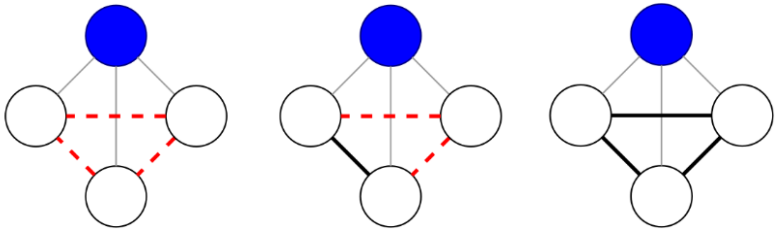
- 저자의 연구 중점
  - Network structure system과 predictive performance간 연관이 존재하는가?
  - 제대로 작동하는 neural network의 구조상 특징은 무엇인가?
  - 그러한 구조상 특징은 어떻게 입력을 task에 일반화하여 답을 도출하는가?
  - Neural network가 믿을 만 한지 알 수 있는 효율적인 방법이 존재하는가?

Introduction

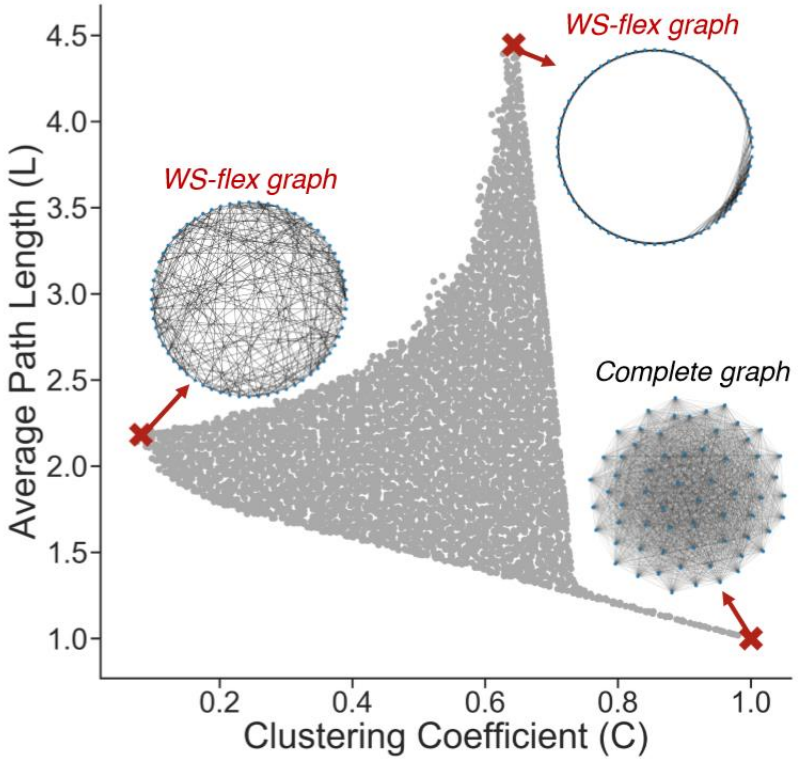
- 기존 기법들의 한계
  - Network architecture와 그것의 정확도간 상관관계를 체계화 하는 것은 쉽지 않음
  - Neural network를 graph로 변환시키거나 그 반대로 하는 확실한 방법이 없었음
  - 기존의 computational graph가 존재하나, 한계점이 다수 존재
    - Generality 부족: 방향성이 존재하고, cyclic한 특성이 없음
    - 생물학/뇌과학과의 괴리: 실제 신경세포의 구조는 제한성이 없음 (template에서 자유로움)
- Relational graph
  - 방향성이 있는 data flow가 아님
  - Neuron(node)들 간의 message exchange에 초점을 맞춤
  - 현재까지 neural network에서 자주 쓰이고 있던 layer들을 표현 가능
  - 기존의 한 layer 통과를 relational graph의 한 round의 message exchange로 취급
  - 여러 round 동안 message exchange를 함으로 써 deep neural network구현
  - Input channel과 output channel을 하나의 node에 한꺼번에 표현 가능

Introduction

- WS-flex
  - System 관점에서 neural network에 접근하는 graph generator
  - Network를 clustering coefficient 및 path length 평균에 따라 characterize
  - Flexible 하고 general한 framework 특성에 따라 relational graph들이 다양한 architecture들로 해석 가능 (MLP, CNN, ResNet 등)



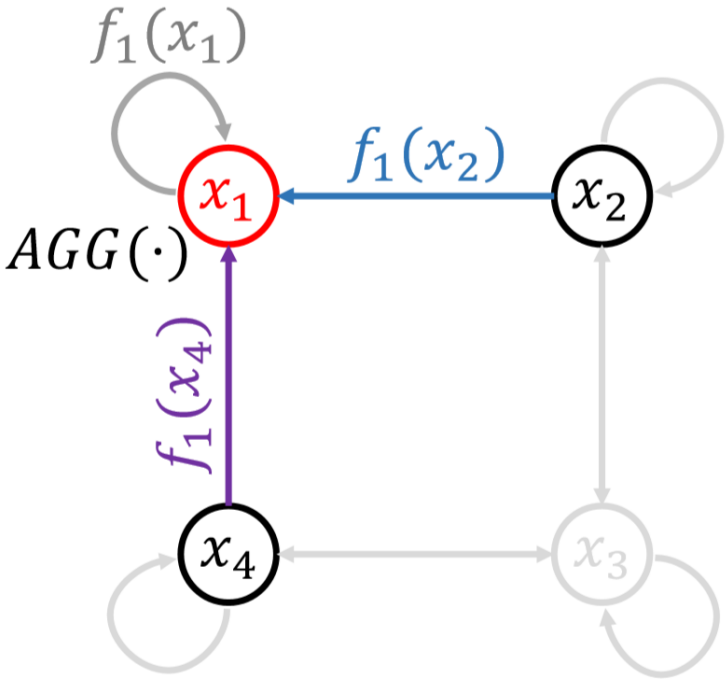
Clustering coefficient의 예시



Relational graph의 예시

Neural Networks as Relational Graphs

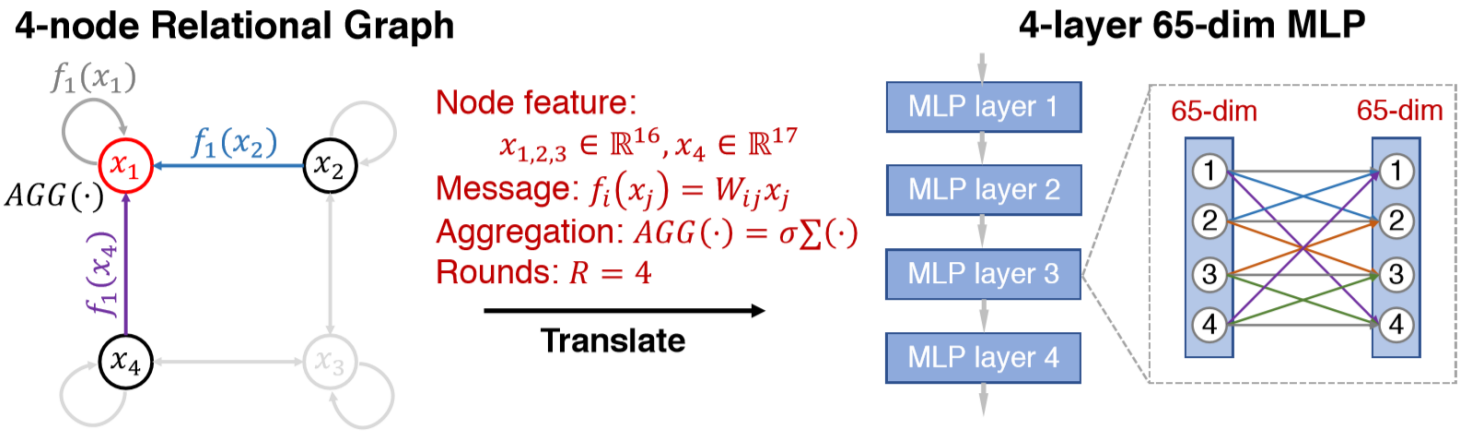
- Graph간 message exchange
  - Node set: $\mathcal{V} = \{v_1, \cdots, v_n\}$. Edge set: $\mathcal{E} \subseteq \{(v_i, v_j) | v_i, v_j \in \mathcal{V}\}$
    - Node v는 scalar/vector x_v를 feature로 가짐
  - 각 message는 edge들을 통해 전달됨
    - Message function: $f(\cdot)$
  - 전달된 message들의 aggregation function 적용
    - $x_v^{(r+1)} = AGG^{(r)}(\{f_v^{(r)}(x_v^{(r)}), \forall u \in N(v)\})$
    - Aggregation function은 각 신호가 수집되어 저장되는 방식을 결정 (이 경우에는 activation function들의 역할과 유사)
  - Message 전달의 반복으로 여러 layer 통과효과



Message exchanging graph

Neural Networks as Relational Graphs

- Width가 고정된 multi-layer perceptron(MLP)의 relational graph화
  - 일정한 개수, 차원의 node들이 반복적으로 데이터를 주고 받는 전형적인 graph 구도 형성
  - MLP의 dimension이 일정하게 유지되는 경우에 한정 됨
    - Input, hidden, output layer의 dimension이 모두 같은 경우
    - Node가 모두 같은 feature개수를 가져야 함



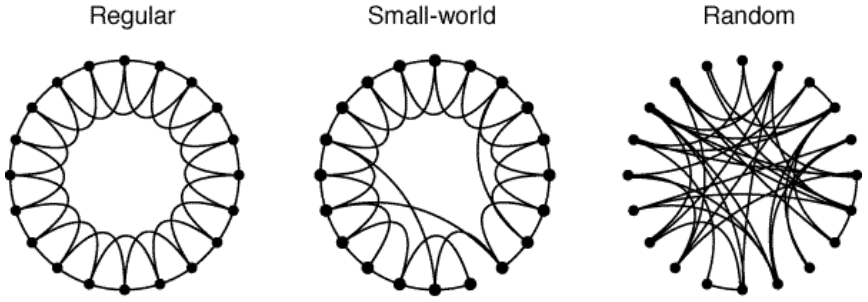
Relational graph의 MLP로의 해석

Neural Networks as Relational Graphs

- General neural network의 relational graph화
  - 가변 width MLP
    - $x_i^{(r+1)} = \sigma\{\sum_{j \in N(i)} W_{ij}^{(r)} x_j^{(r)}\}$ 표현 가능
    - Graph 형성법
      - n개의 node를 가진 relational graph가 존재한다 가정
      - m개의 dimension을 가진 layer를 표현
      - n개의 node들 중 m mod n개의 node는 각각 $\lfloor m/n \rfloor + 1$의 dimension 부여
      - n개의 node들 중 n – (m mod n)개의 node는 각각 $\lfloor m/n \rfloor$의 dimension 부여
    - 예시
      - 4-node relational graph를 이용하여 2-layer neural network 표현
      - 첫 layer의 width : 5, 두번째 layer의 width: 9
      - 4개의 node는 첫번째 round에 {2,1,1,1}개의 feature가 사용
      - 두번째 round에 {3,2,2,2}개의 feature가 사용

Exploring/Generating Relational Graphs

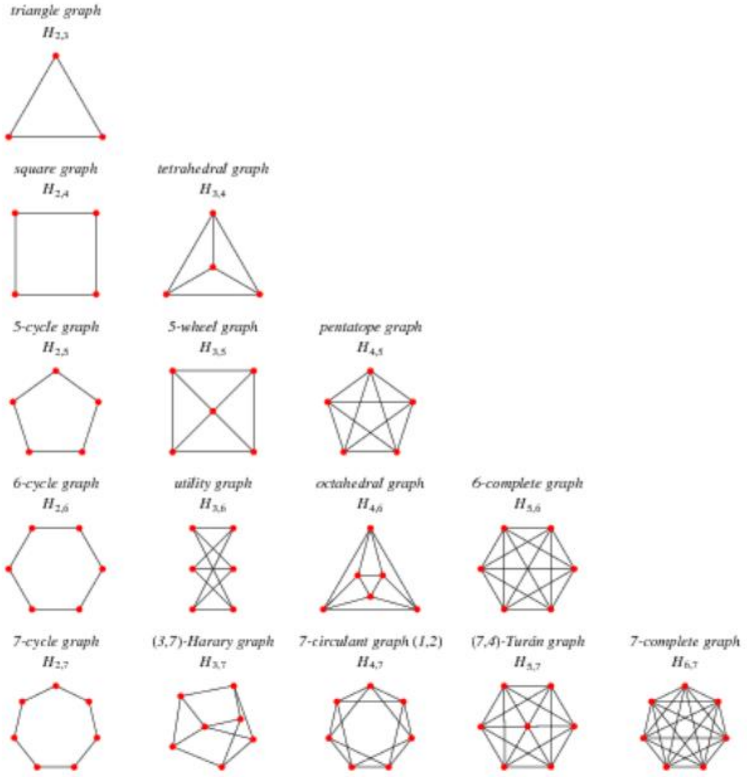
- Graph measures
  - Global graph measure로써 average path measure 이용
  - Local graph measure로써 clustering coefficient 이용
- Graph generator의 design
  - Classic graph generator의 경우 제한된 graph만 생성 가능
    - ER model: Node, edge의 개수를 uniform한 분포로 random하게 생성
    - WS model: Small-word property를 가진 model생성
      - Small-world network: Node간 연결 개수는 적으나, clustering coefficient가 크고, 특정 node의 data가 적은 round의 data exchange만으로 모든 node에 도달할 수 있는 network



Small world network의 예시

Exploring/Generating Relational Graphs

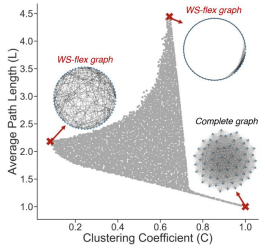
- Graph generator의 design
  - Classic graph generator의 경우 제한된 graph만 생성 가능
    - BA model: scale-free graph 생성 가능
    - Harary model: Connectivity가 maximize된 (최소 개수의 edge를 가지는) graph 생성
    - Ring graph: 모든 node들이 ring 형태로 연결된 graph (harary model에 포함)
    - Complete graph: 모든 node가 모든 다른 node에 연결된 graph
    - 위 모든 classical한 경우, 제한된 개수의 graph가 생성 됨



Harary graph의 형성법

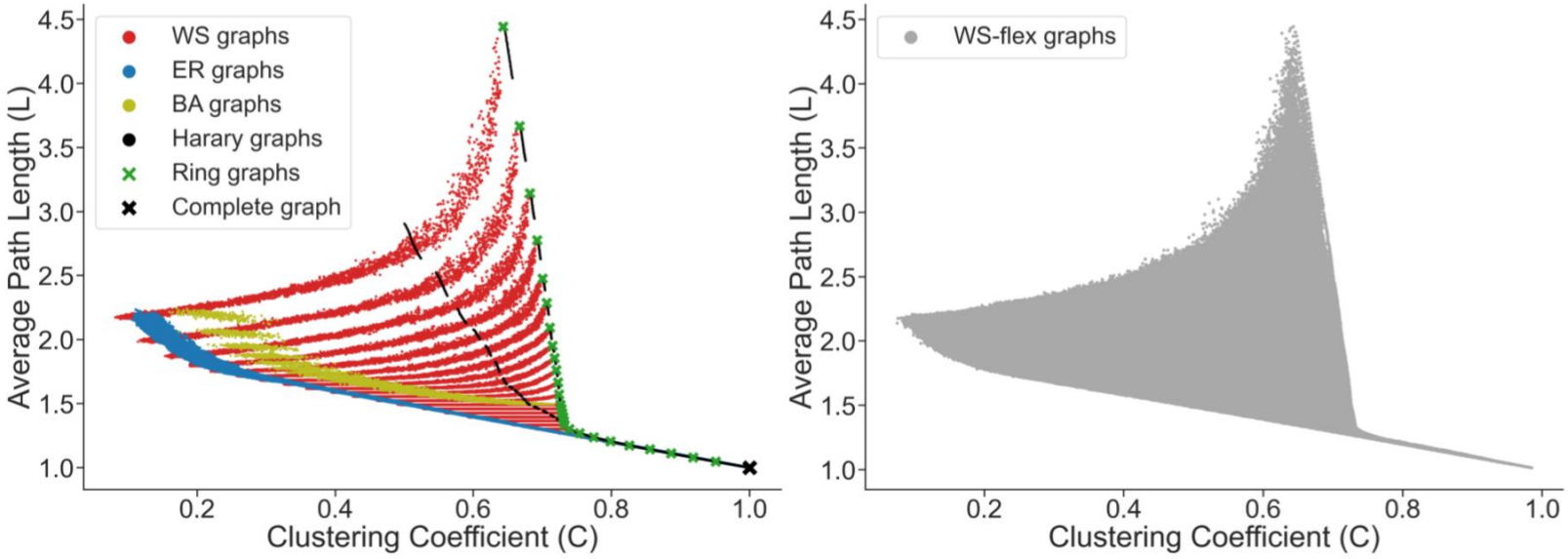
Exploring/Generating Relational Graphs

- Graph generator의 design
  - WS-flex graph generator
    - WS model을 constraint를 relaxing 시킴으로써 generalize
    - 넓은 범위의 graph 형성 가능
    - Node n, average degree k, rewiring probability p로 parameterized됨
    - Edge의 개수는 $e = \lfloor n * k/2 \rfloor$로 설정
    - 첫 과정에서 ring graph를 형성
    - e mod n개의 node를 random하게 선택한 후, 각 node를 가장 가까운 node에 연결
      - Ring 형태로 위치 관계를 설정해 놓은 것
    - 최종적으로 모든 edge들은 probability p의 확률로 재 연결 됨
    - WS-flex를 이용하여 clustering coefficient와 average path length에 대해 sample 제작
    - 생성된 sample들 중 3942개를 subsample



Exploring/Generating Relational Graphs

- Graph generator의 design



- Computational budget controlling
  - Baseline network(complete relational graph를 기준으로 삼는 경우가 있음)의 FLOPS(#of multiply-adds)을 먼저 구하고, 이를 기준으로 비교
  - 기준 network의 0.5% FLOP을 가지도록 width를 변화시킴

Experiments

- Experimental Setup
  - Base Architectures
    - CIFAR10 experiments (50K training images, 10K validation images)
      - 512개의 hidden unit들이 5-layer MLP에 분포하도록 설정
      - Input: 3072-dimension의 vector로 flatten된 (32X32X3)의 image
      - Output: 10-dimension(class 개수)
      - 128의 batch size로 200 epoch 간 train
      - Cosine learning rate scheduling 사용, learning rate: 0.1 -> 0으로 anneal, restarting없음
      - 5개의 random seed를 이용하여 제작하고, 이들의 평균 결과를 report
    - ImageNet experiment
      - ResNet-34, ResNet-34-sep, ResNet-50, EfficientNet-B0, simple 8-layer CNN 이용
      - 256/512의 batch size로 100 epoch간 train
      - Learning rate: 0.1을 initial learning rate로 하는 cosine annealing learning rate
      - 각 model에 대하여 3개의 random seed 이용
      - Model에 따라 reference computational complexity 설정

Experiments

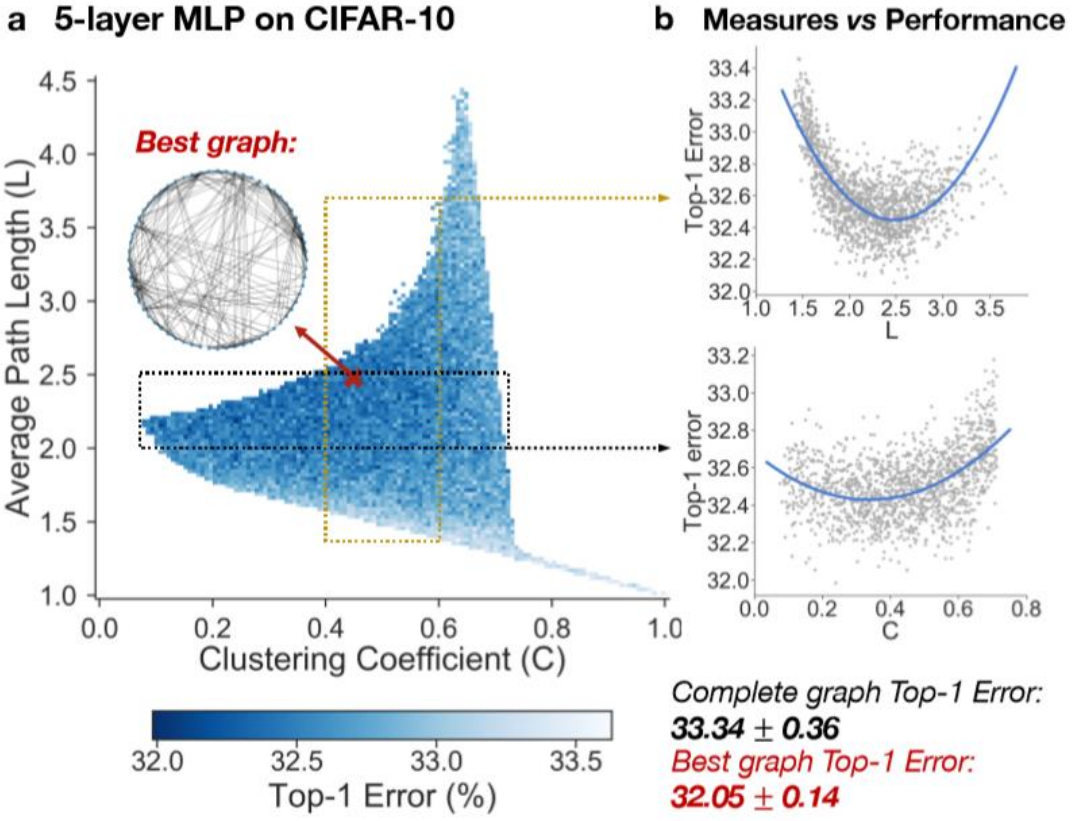
- Experimental Setup
  - Exploration with Relational Graphs
    - 모든 dense layer들(linear layer, 3X3/1X1 convolution layer)을 relational graph로 치환
    - Down-sampling, skip connection등의 설계구조를 남겨두기 위하여 input/output layer는 변화시키지 않음
    - CIFAR-10 MLP 의 경우 64개의 node로 이루어진 3942 graph를 확인
    - ImageNet 의 경우 computational cost를 줄이기 위해 3942 graph들 중 52개의 graph를 uniform하게 subsample함

Graph Structure of Neural Networks

	Fixed-width MLP	**Variable-width MLP**	**ResNet-34**	**ResNet-34-sep**	**ResNet-50**
Node feature $\mathbf{x}_i$	Scalar: 1 dimension of data	Vector: multiple dimensions of data	Tensor: multiple channels of data	Tensor: multiple channels of data	Tensor: multiple channels of data
Message function $f_i(\cdot)$	Scalar multiplication	(Non-square) matrix multiplication	3×3 Conv	3×3 depth-wise and 1×1 Conv	3×3 and 1×1 Conv
Aggregation function AGG$(\cdot)$	$\sigma(\sum(\cdot))$	$\sigma(\sum(\cdot))$	$\sigma(\sum(\cdot))$	$\sigma(\sum(\cdot))$	$\sigma(\sum(\cdot))$
Number of rounds R	1 round per layer	1 round per layer	34 rounds with residual connections	34 rounds with residual connections	50 rounds with residual connections

Experiments

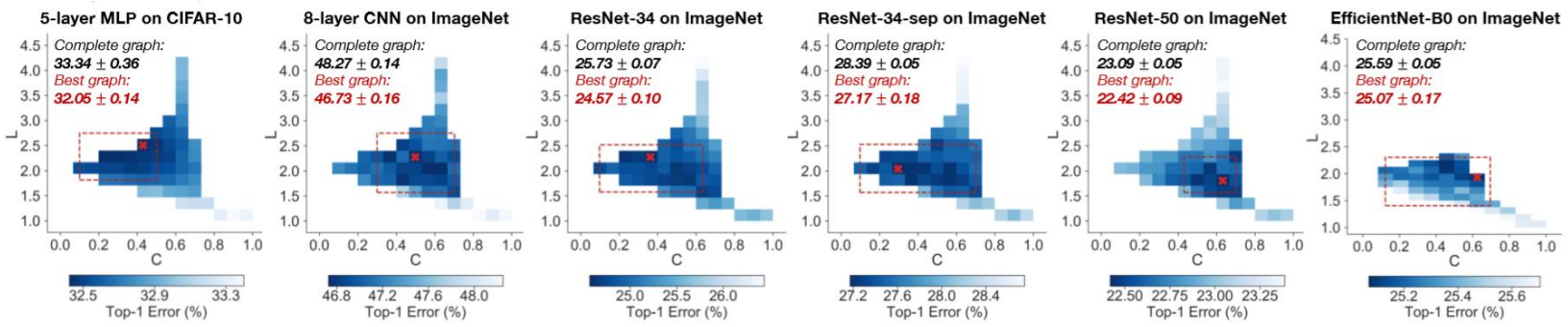
- Results



각 graph 에 Clustering coefficient (C), average path length (L)에 따른 top-1 error heat map 으로 나타냄

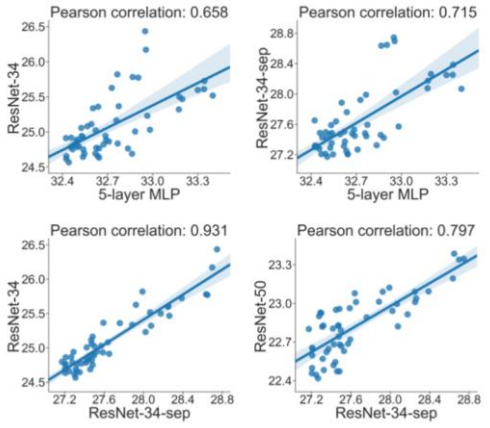
Experiments

- Results
  - A sweet spot for top neural networks
    - 위 그래프는 complete graph에 비해 더 좋은 성능의 graph 구조가 존재함을 나타냄
    - CIFAR-10의 경우 1.4%더 높고, ImageNet의 경우 0.5%에서 1.2%더 높은 정확도를 보임
    - 붉은 직사각형으로 표현된 sweet spot 내부에 높은 performance를 보이는 graph들이 밀집되어 있음
    - 이를 확인하기 위하여 앞에서 보인 3942개의 graph를 52개의 bin으로 heatmap에 대한 downsampling을 진행한 뒤, 최고 평균 performance를 가진 bin에 비해 one-sided t-test 결과 p-value가 0.05 이하인 경우를 rectange 모양으로 설정하여 sweet spot 구함



Experiments

- Results
  - Neural network performance as a smooth function over graph measures
    - 앞의 그림에서 보여진 것과 같이, 같은 clustering coefficient와 average path length에서의 performance의 변화 양상은 smooth graph의 양상을 띔
    - 2차 다항함수를 이용하여 regression한 결과, U-shape가 나타남을 확인하였음.
  - Consistency across architectures
    - Qualitative consistency
      - 다른 architecture들임에도 불구하고 sweet spot들의 위치가 consistent하였음
        - ✓$C \in [0.43, 0.50], L \in [1.82, 2.28]$ 범위안에 존재
      - 모든 architecture에서 U-shape trend가 관찰 됨
    - Quantitative consistency
      - 52개의 각 bin들에 포함되는 heatmap의 평균값을 구하여 consistency 확인
      - Model별로 같은 graph measure들을 가진 경우에 대해 linear한 performance를 보여줌

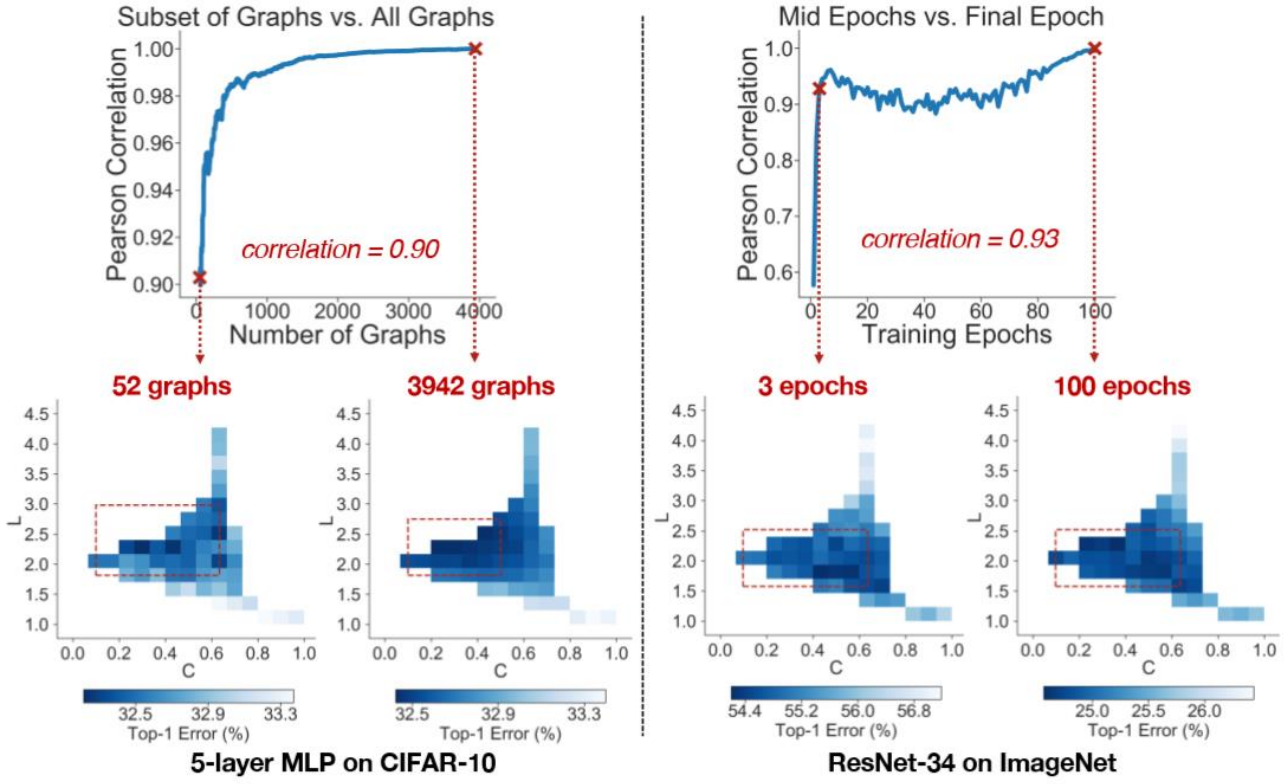


Experiments

- Results
  - Quickly identifying a sweet spot
    - 그래프의 개수를 줄이는 법
      - 52개의 bin을 3942개의 graph의 performance 모두 이용하여 구하지 않는 방법
      - Bin 내부의 heat map중 일부만을 이용하여 52개의 bin 제작
      - 3942개의 graph를 사용한 경우에 비교하여 최소(52개의 sample 이용) 0.90의 큰 Pearson correlation을 얻음
      - 적은 sample만을 이용하여 분석을 하여도 충분히 의미 있는 결과를 얻을 수 있다고 주장
      - 수치상 52개보다 조금만 더 높은 sample 개수를 사용하여도 Pearson correlation은 크게 증가하므로, 더 높은 정확도를 원하더라도 문제 없음
    - Training epoch를 줄이는 법
      - 전체 100epoch의 training이 진행되었지만, 3 epoch만으로도 높은 correlation 확인(0.93)
      - 좋은 relational graph는 초기 training epoch에도 좋은 performance를 보여준다고 해석
      - Computational cost를 크게 낮출 수 있는 단서를 주기에 중요한 발견

Experiments

- Results
  - Quickly identifying a sweet spot



Experiments

- Results
  - Network science and neuroscience connections
    - Network science
      - 평균 path length가 큼으로 인해 message exchange efficiency가 낮아질 수 있음
      - U-shape correlation과의 trade off가 발생할 수 있음
    - Neuroscience
      - Top artificial neural network의 graph measure가 biological neural network와 유사
      - 생물들의 neural network를 5-layer MLP로 해석, complete graph보다 높은 정확도 보임

Graph	Path (L)	Clustering (C)	CIFAR-10 Error (%)
Complete graph	1.00	1.00	33.34 ± 0.36
Cat cortex	1.81	0.55	33.01 ± 0.22
Macaque visual cortex	1.73	0.53	32.78 ± 0.21
Macaque whole cortex	**2.38**	**0.46**	32.77 ± 0.14
Consistent sweet spot across neural architectures	**1.82-2.28**	**0.43-0.50**	32.50 ± 0.33
Best 5-layer MLP	**2.48**	**0.45**	32.05 ± 0.14

Discussion and Conclusion

- Discussion
  - Hierarchical graph structure of neural networks
    - Neural network는 hierarchical graph으로써의 내재된 특성을 generalize하고자 함
  - Efficient implementation
    - 현재 implementation은 CUDA kernel을 이용하여 weight masking에 의존함
    - 이는 complete graph에 비해 느린 속도를 유발함
    - 이론적인 FLOP과 실제 gain과의 괴리를 없애기 위해 연구 중
  - Prior vs Learning
    - Graph structure를 hard-wiring을 통해 제작 시도
    - 이를 통해서 성능이 좋은 MLP를 얻는 방법을 찾기 위한 노력 소개
  - Unified view of Graph Neural Networks(GNN) and general neural architectures
    - Neural network를 message exchange로써 재정의하는 기법은 Kipf의 GCN을 비롯한 여러 GNN에서 영감을 받아 제작되었음
    - GNN은 graph structure가 input으로 사용되고, message functio들이 모든 edge에 대하여 공유되는 특별한 neural architecture라 불 수 있음

Discussion and Conclusion

- Conclusion
  - CIFAR-10, ImageNet dataset을 이용하여 neural network graph의 구조가 predictive performance에 영향을 미치는지 확인하였고, 이는 graph measure들에 영향을 받음
  - Relational graph의 “Sweet spot”을 이용하여 computational budget이 control된 상태에서 높은 performance를 얻음
  - Neural network의 performance는 clustering coefficient와 평균 path length의 smooth function(무한히 미분 가능한 함수)에 근사
  - 다양한 architecture들과 dataset에 적용 가능
  - 최적화된 graph 구조를 graph measure에 따라 묶어 놓은 모임을 sweet spot이라 하고, 이를 찾아내는 데는 몇가지 sample relational graph와 적은 epoch의 training만을 필요로 함
  - Performance가 좋은 neural network는 놀랍게도 실제 생물들의 neural network과 유사