Analyzing Graph Neural Network Architectures through Neural Tangent Kernel

Problem Setup: Node Classification

- Graph $G$ with $n$ nodes
- Adjacency matrix $A \in \{0, 1\}^{n \times n}$
- Degree matrix $D \in \mathbb{R}^{n 	imes n}$
- Feature matrix $X \in \mathbb{R}^{n \times f}$
- $m$ nodes label $Y \in \{1, \ldots, K\}^m$

Predict labels for the unlabeled nodes

Graph Convolution Network

\[
S = S_{\text{conv}} = D^{-1/2}AD^{-1/2} \text{ or } S_{\text{conv}} = D^{-1/2}A, \sigma(\cdot) = \text{Linear or ReLU}, W_i \in \mathbb{R}^{h \times h}
\]

Intriguing Empirical Observations

1. $S_{\text{conv}}$ performs better than $S_{\text{avg}}$ for any depth $d$
2. Performance $\downarrow$ as $d \uparrow$, skip-connections fix it
3. $\sigma(\cdot)$ = Linear performs as good as $\sigma(\cdot)$ = ReLU

Analysis using Graph Neural Tangent Kernel and Degree Corrected Stochastic Block Model (DC-SBM)

Graph Neural Tangent Kernel as $h \to \infty$

\[
\Theta = \sum_{i=1}^{n} \Theta_i \odot (S \Sigma S^T)^{(d+1-i)} \odot \left( \begin{array}{c} d \end{array} \right) \odot E_i
\]

where $\Sigma_i = SX_i S^T$, $\Sigma = \sum_{i=1}^{n} \Sigma_i$, $E_i = \text{influence of } \sigma(\cdot)$.

DC-SBM: Random graph model characterized by $p, q \in [0, 1]$ and degree correction vector $\pi = (\pi_1, \ldots, \pi_n) \in [0, 1]^n$. Then for $K$ latent classes, $\mathcal{C} = \{1, \ldots, K\}$, the population adjacency matrix $M = [A_i]$, is

\[
M_{ij} = \begin{cases} 
\sigma(\pi_j) & \text{if } C_i = C_j \\
\sigma(\pi_j) & \text{if } C_i \neq C_j 
\end{cases}
\]

Visualizations of our Theoretical Results

1. Class structure is preserved in $S_{\text{conv}}$
2. Performance $\downarrow$ as $d \uparrow$
3. Skip-connections retain info even at $d \to \infty$
4. Linear as good as ReLU

Fast Adaptive Test-Time Defense with Robust Features

Problem Statement: Improve Adaptive Test-time Defense

Given a trained neural network, how can we make it robust to adversarial attacks at test-time? Can we efficiently improve the robustness at test-time?

Idea: Project the learned features to the robust subspace

Representation Learning with Tensorized Autoencoder

Problem Statement: Improve representation of multi-modal data

Standard AE learns one representation of the data. How to improve?

\[
\min \left\{ \alpha_i \right\}_{i=1}^{k} \sum_{j=1}^{k} S_{ij} \left\| X_i - C_j \right\|_2 - \lambda \left\| g_j (X_i - C_j) \right\|_2^2
\]

$g_i()$ and $f_j()$ are the encoder and decoder for cluster $j$, $C_j$ is the center of class $j$, $S_{ij}$ assigns a datapoint $i$ to an AE $j$.

Theory: Optimum for Linear TAE

Class Assignment $S_{ij} = 0$ or 1, centers $C_j$ and encoding corresponds to the top $h$ eigenvectors of $\sum_{i=1}^{n} S_{ij} (X_i - C_j)^T (X_i - C_j)$.

Empirical Performance

TAE outperforms other methods in denoising and competitively in clustering