# SEMI-SUPERVISED CLASSIFICATION WITH GRAPH CONVOLUTIONAL NETWORKS

Nick Puglisi, Student B.

# Table of contents

# Motivation

Semi-Supervised vs. Supervised vs. Unsupervised Learning

- Supervised Learning X - Y
- Unsupervised X
- Semi-Supervised X - some Y

Efficient community detection

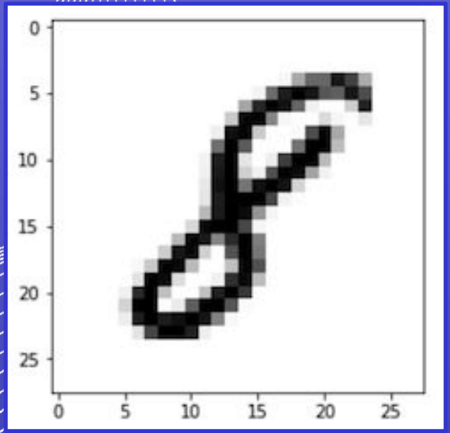
- Only a portion of the data needs to be labeled

# 01

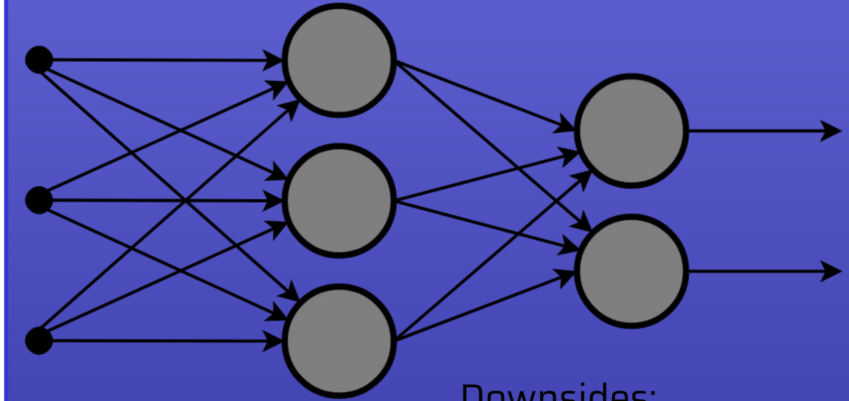
# Convolutions

Why use convolutional network layers?

# Traditional Neural Networks




28x28 Image



784 Input Parameters

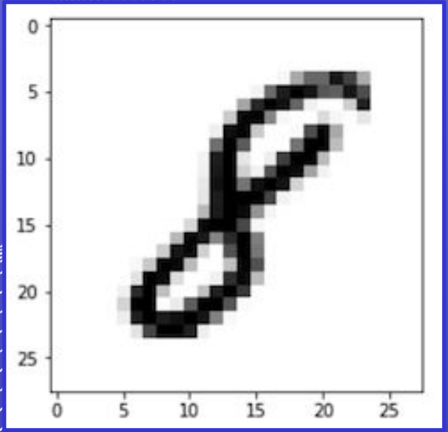
<u>Downsides</u>:

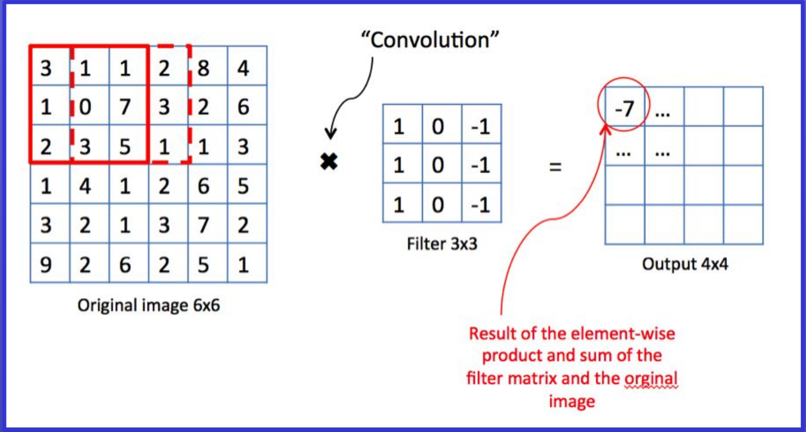
Number of parameters quickly explodes (10 Nodes, 1 Hidden Layer) = 7000+ Parameters

- High Training Costs
- Overfitting

Assumes all pixels are independent

# Convolutional Neural Networks

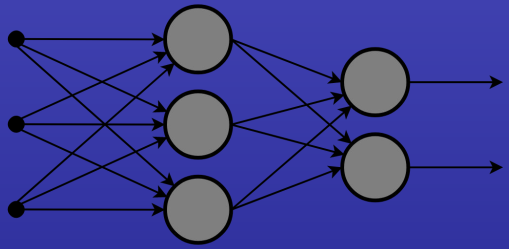




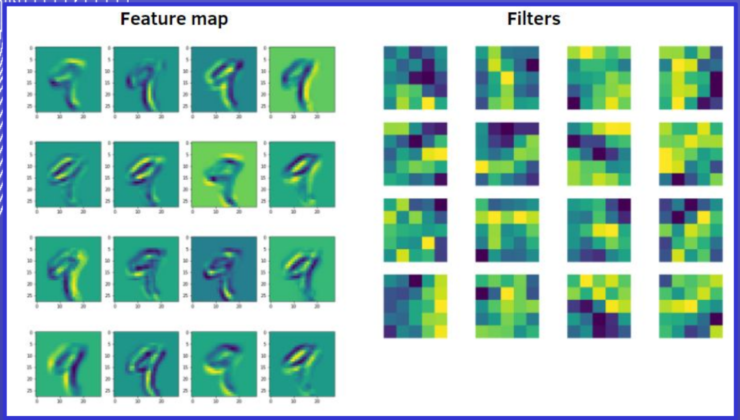

28x28 Image

4x4 Image

16 Inputs



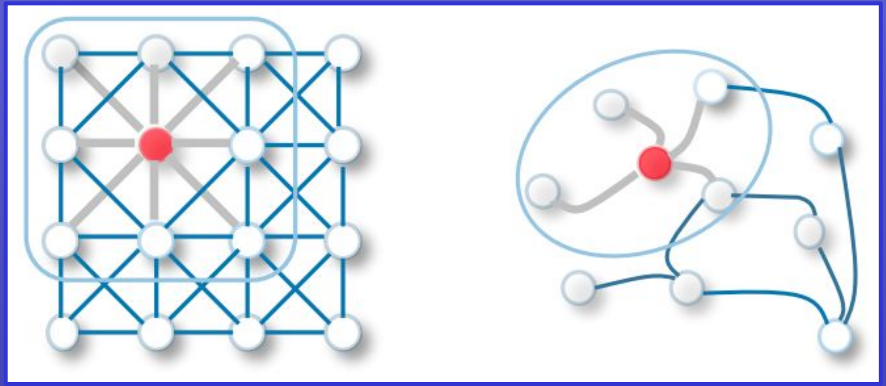
# Convolutional Neural Networks




Advantages

- Reduced parameters, less training time
- Preserves correlation between neighboring pixels
- Greatly increased accuracy
- Literal breakthrough in image recognition technology

# Convolutional Neural Networks



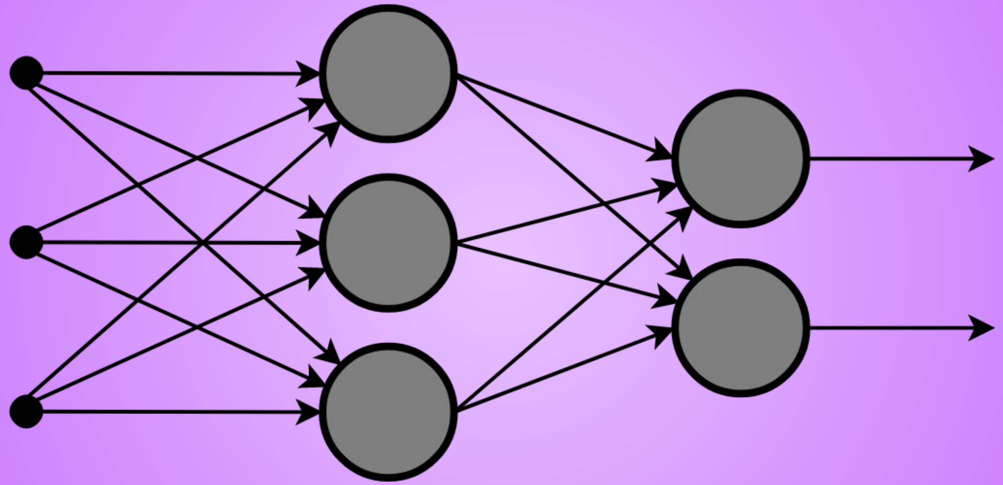
Instead of neighboring pixels, we'll assume that neighboring nodes contain useful information, this is the basis for GCN's

02

# Graph Convolutions

How do convolutions generalize to network data?

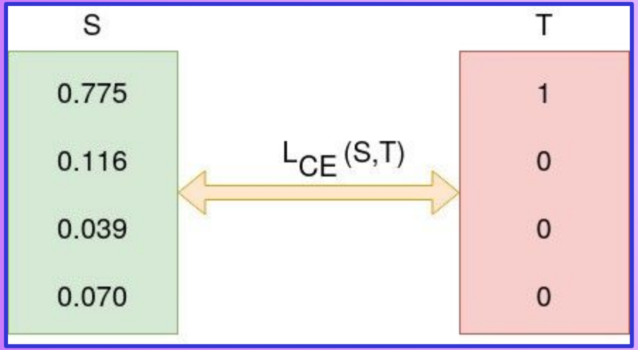
# The Forward Pass

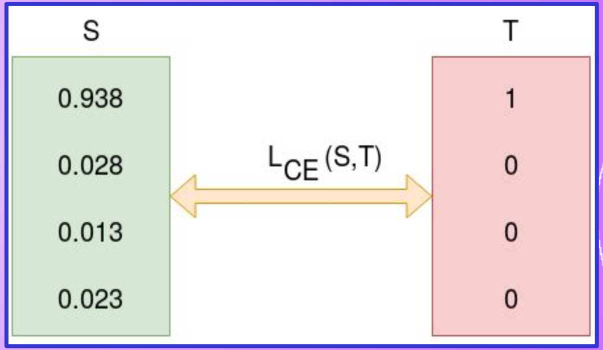


$$Z = \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} X \Theta$$

# Loss Function - Cross Entropy

$$\mathcal{L} = -\sum_{l \in \mathcal{Y}_L} \sum_{f=1}^{F} Y_{lf} \ln Z_{lf}$$



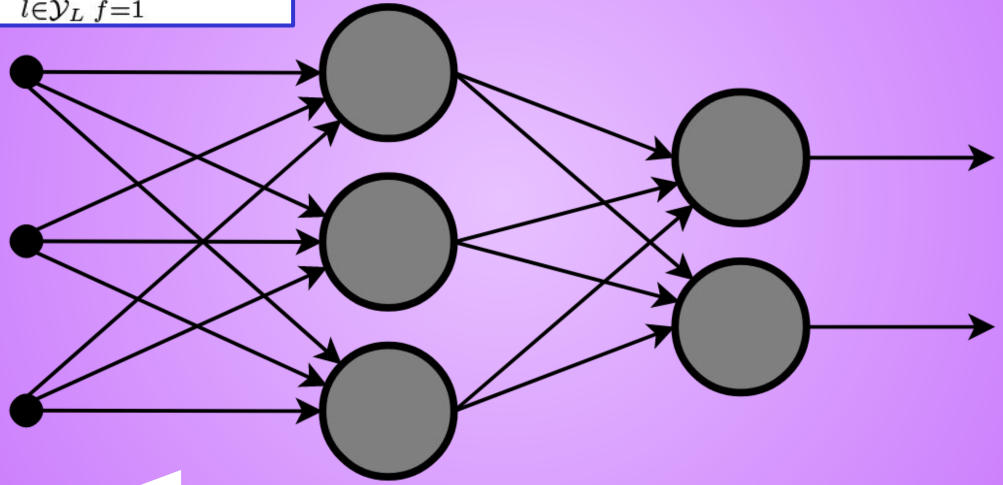




Loss = 0.3677

Loss = 0.095

# Backpropagation

$$\mathcal{L} = -\sum_{l \in \mathcal{Y}_L} \sum_{f=1}^{F} Y_{lf} \ln Z_{lf}$$



$$\mathbf{a}_{n+1} = \mathbf{a}_n - \gamma \nabla F(\mathbf{a}_n)$$

Stochastic Gradient Descent

# 03

# Experimental Design

Applying GCN's to a myriad of datasets.

# Experimental Design

- GCNs were tested in a number of experiments:
  - Semi-supervised document classification
  - Semi-supervised entity classification
    - Bipartite graph extracted from NELL
- Host of datasets are selected for testing

# Datasets and Design

- Datasets used include:
  - Citation Networks: Citeseer (3.6%), Cora (5.2%) , and Pubmed (0.3%)
  - NELL (0.1%)
  - Random Graphs: Training time
- Two-layer GCNs are trained and evaluated on a test set of 1000 labeled examples, with comparisons against:
  - ManiReg, DeepWalk, and Planetoid

# Datasets

| | Nodes | Edges | Classes | Features |
|---|---|---|---|---|
| **Citeseer** | 3327 | 4732 | 6 | 3703 |
| **Cora** | 2708 | 5429 | 7 | 1433 |
| **Pubmed** | 19717 | 44338 | 3 | 500 |
| **NELL** | 65755 | 266144 | 210 | 5414 |

# 04

# Results

How do GCN’s perform?

# Semi-Supervised Classification

| | Citeseer | Cora | Pubmed | NELL |
|---|---|---|---|---|
| ManiReg | 60.1 | 59.5 | 70.7 | 21.8 |
| DeepWalk | 43.2 | 67.2 | 65.3 | 58.1 |
| Planetoid | 64.7 | 75.7 | 77.2 | 61.9 |
| **GCN** | **70.3** | **81.5** | **79.0** | **66.0** |

# Results and Limitations

- GCNs outperformed other methods by a large margin
    - GCNs remain computationally efficient
- Limitations:
    - Memory Requirement
    - Directed Edges
    - Initial Assumptions:
        - Locality (implicit to GCN)

# Thanks!

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Freepik**