Graph Convolutional Networks for Text Classification

A paper by L. Yao, C. Mao, and Y. Lua, 2019

Text classification

Why?

News filtering,

Spam detection,

Opinion mining,

and much more...

But!

Before training a model on textual data, one must first process it.

Text representation

Text can be processed into array representations: Embeddings

It is the main method used today: to embed text features in an array (e.g. via a pre-trained model like GloVe).

Ng, Andrew. *NLP and Word Embeddings, deeplearning.ai, 2021*

The Paper's idea

This approach has limits.

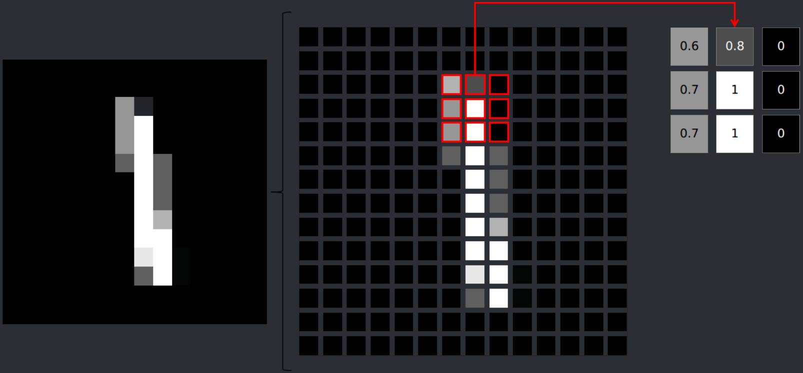
- Embeddings capture word-to-word relations only
- Embeddings do not incorporate a notion of distance between words
- They are two-dimensional

The solution...? a different data structure, and using convolution.

Why?

Convolution

An image is an array



Using an example from MNIST.

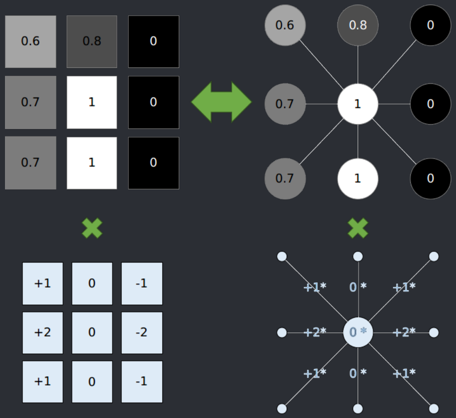
Le Cun, Yann. The MNIST database of handwritten digits, 1998

Convolution

We can perform computation on window-snippets of the array

Convolution

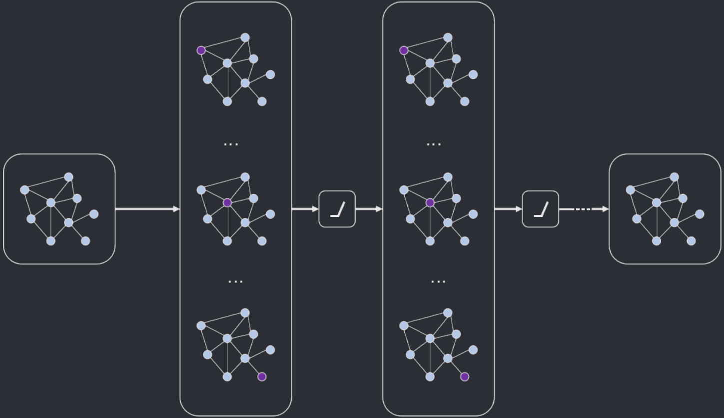
Arrays can be represented as something else: Graphs...



Convolution does not have to apply to pictures and signals only.

Convolution

... and graphs can be used as inputs to neural networks



Kipf, Thomas N. and Welling, Max. *Semi-Supervised Classification with Graph Convolutional Networks, ICLR, 2017*

Graph Convolutional Networks for Text Classification

Graph neural networks, and graph embeddings are recent.

They offer a richer relational representation than usual embeddings, giving less priority to locality and sequentiality.

They capture a higher order neighborhood information.

$\Rightarrow$ The paper proposes a new graph neural network method for text classification.

Battablia et al. *Relational inductive biases, deep learning, and graph networks, 2018*
Yao, L., Mao, C., Luo, Y. *Graph Convolutional Networks for Text Classification, 2019*

Graph Convolutional Networks for Text Classification

How to construct a graph

<u>Usual ML solution:</u> builds a word or document embedding.

<u>Graph neural network solution:</u> learns both at once.

It relies on a **non-grid** or **arbitrarily structured graph**:

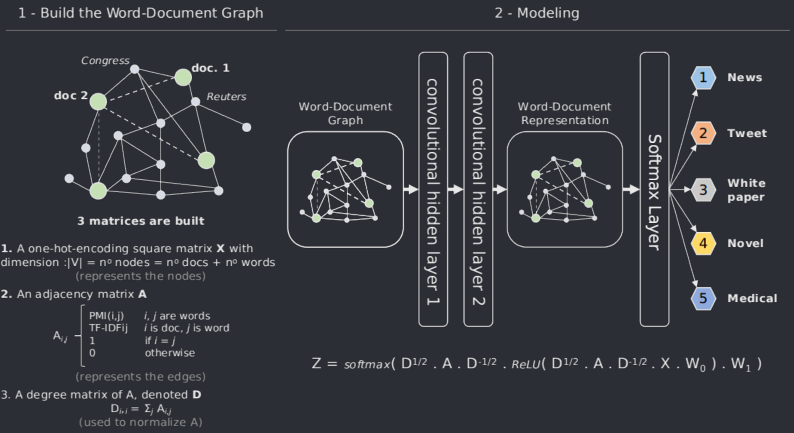
- Nodes represent words <u>or</u> document types
- Edges represent co-occurrences

$\Rightarrow$ Text <u>and</u> document classification is a node classification problem

Bruna, J. et al. *Spectral Networks and Locally Connected Networks on Graphs, 2014*

Graph Convolutional Networks for Text Classification

A 2-Layer Graph Convolutional Network



Note: A fixed size sliding window on all documents in the corpus is used to gather co-occurrence statistics.
TF-IDF: term frequency-inverse document frequency; **PMI**: point-wise mutual information

Graph Convolutional Networks for Text Classification

Datasets

Name	Content	# Docs	Split	# Words	# Nodes	# Classes
20NG	News slips	18,846	60-40	42,757	61,603	20
R8	Reuters cables	7,674	70-30	7,688	15,362	8
R52	Reuters cables	9,100	71-29	8,892	17992	52
Ohsumed	Medical lit.	7,400	45-55	14,157	21,557	23
MR	Movie reviews	10,662	67-33	18,764	29,426	2

Embedding size of the first convolutional layer: 200

Window size: 20

The paper's main question:

Can the model achieve satisfactory results in text classification, even with limited labeled data?

Graph Convolutional Networks for Text Classification

Results

Mean test accuracy - Models were run 10 times

Models	20NG	R8	R52	Ohsumed	MR
Text GCN	0.86	0.97	0.94	0.68	0.77
SWEM	0.85	0.95	0.93	0.63	0.77
TF-IDF + LogReg	0.83	0.94	0.87	0.55	0.75
CNN	0.82	0.96	0.88	0.58	0.78
LEAM	0.82	0.93	0.92	0.59	0.77

"**Text GCN performs the best and significantly outperforms all baseline models**."

Note: 12 models were omitted from this table.
SWEAM: Simple Word Embedding Model. **LEAM**: Label-Embedding Attentive Model.

Graph Convolutional Networks for Text Classification

Further results

Text GCN can capture both document-to-word and global word-to-word relations.

Word nodes capture document label information and act as bridges: label information propagates across the graph.

Text GCN does not outperform CNN and LSTM on the MR dataset: Text GCN *ignores word order*, a key feature in sentiment analysis.

Formulas

TF-IDF:

$$\frac{\#\, word\ occurrences\ in\ the\ document}{log(\#\ of\ documents\ that\ contain\ the\ word)}$$

PMI:

$$PMI(i,j) = log\frac{p(i,j)}{p(i)p(j)}$$

$$p(i,j) = \frac{\#W(i,j)}{\#W}$$

$$p(i) = \frac{\#W(i)}{\#W}$$

with $\#W(i)$, the number of sliding windows in a corpus that contains word i, $\#W(i,j)$, the number of sliding windows that contain both words i and j, and $\#W$ the total number of sliding windows.

Bibliography

[1] Ng, Andrew. *NLP and Word Embeddings - CS230 Deep Learning, Stanford University*. deeplearning.ai, 2021.

[2] Le Cun, Yann. *The MNIST database of handwritten digits*. 1998.

[3] Kipf, Thomas N. and Welling, Max. *Semi-Supervised Classification with Graph Convolutional Networks*. International Conference on Learning Representations (ICLR), 2017.

[4] Battaglia, P. et al. *Relational inductive biases, deep learning, and graph networks*. 2018.

[5] Bruna, J. et al. *Spectral Networks and Locally Connected Networks on Graphs*. 2014.