APS360 Fundamentals of AI

Lisa Zhang

Lecture 9; Feb 7, 2019

Agenda

- Project
- Unsupervised Learning
- Autoencoders (continued)
- Word Embeddings

Group Project

Guidelines

https://www.cs.toronto.edu/~lczhang/360/project.html

Teams

- Teams of 3 **only**
- Teams of 2 or 4 only if there are extrenuous circumstances, or if there is an odd number of people
  - Let me knows ASAP

Examples

- http://www.eecg.utoronto.ca/~jayar/mie324/sortinghat.pdf
- http://www.eecg.utoronto.ca/~jayar/mie324/hawkeye.pdf
- http://www.eecg.utoronto.ca/~jayar/mie324/musicgenre.pdf
- http://www.eecg.utoronto.ca/~jayar/mie324/sparselang.pdf
- http://www.eecg.utoronto.ca/~jayar/mie324/enhance.pdf
- http://www.eecg.utoronto.ca/~jayar/mie324/highhoeps.pdf

Project Uniqueness

- Each project must be **unique**
  - The goal must be unique
  - The dataset used must be unique
- Send an email to Lisa with 1-2 sentence description of topic
  - "you have uniqueness approval"
  - "please try again, that topic is taken"

You must obtain uniqueness approval by Feb 15th 9pm.

Where to get datasets?

- Collect your own
- Kaggle https://www.kaggle.com/
- UCI ML Repo https://archive.ics.uci.edu/ml/index.php
- https://medium.com/datadriveninvestor/the-50-best-public-datasets-for-machine-learning-d80e9f030279

There must be a data cleaning component to your project.

Project Timeline

- Proposal (Feb 24)
- Progress Meeting (Mar 4-11)
- Progress Report (Mar 17)
- Presentation Slides (Mar 29)
- Project Report (Apr 5)

Project Repository

- Open source repository on GitHub
- Speak to be regarding alternatives if you don't want your code to be open source

Project Proposal

- Maximum of 1200 words
- Sections:
  - Introduction
  - Source of Data
  - Overall Structure of your software
  - Plan
  - Risks
  - Things to Learn
  - Ethical Issues
  - References

Project Proposal Word Limit

- There is a 1% penalty for every word in excess of the 1200 limit
- Please count the words in your document, compute the penalty, and put it on the front page.
- If you can present your ideas in much less than 1200 words, please do so.

Office Hours

More to be posted...

Unsupervised Learning

Supervised vs Unsupervised Learning

Supervised Learning:

- Predict an output/target feature given the input features
- Use *labelled* data

Unsupervised Learning:

- *Unlabelled* data, no clear target
- The goal is to find "structure" in the data
  - find clusters
  - generate new data

Classification vs Regression

Classification:

- A type of supervised learning problem where the target feature is **categorical**
  - Cancer vs no cancer
  - Cats vs dogs
  - Digits 0-9

Regression:

- A type of supervised learning problem where the target feature is **continuous**
  - Predict real-estate value
  - Predict stock price

Unsupervised Learning

Building an "Autoencoder" like last week is an example of unsupervised learning

- No target labels
- Find a low-dimensional representation of digits
- Denoising (e.g. removing noise in an image)
- Inpainting (e.g. filling in an image)
- Imputing missing features

Why low-dimensional embedding?

What happens if we used a high-dimensional representation?

Representation Learning

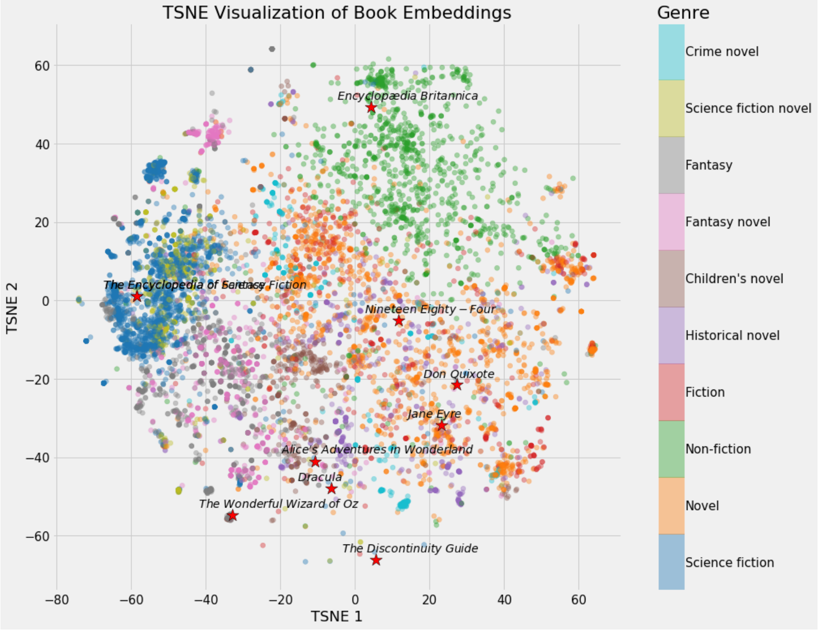
- Finding a low-dimensional embedding of some data is a common unsupervised learning task
- It can be a pre-cursor to other tasks (e.g. clustering)

Denoising Autoencoder Example

`https://cs.stanford.edu/people/karpathy/convnetjs/demo/auto`

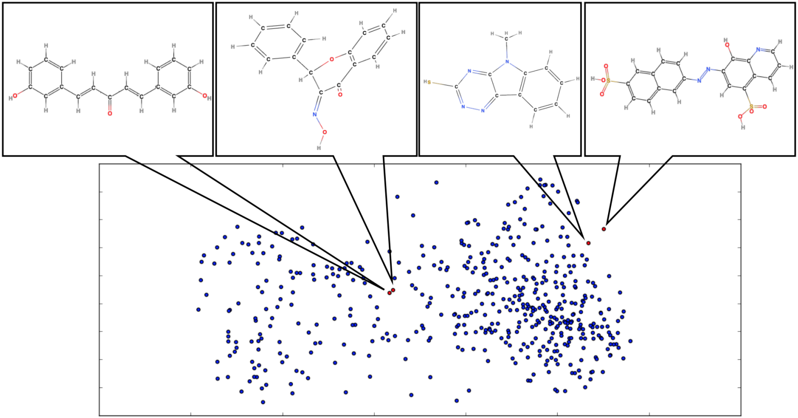
Snapshot of 2D Representation

Embedding of Books



https://towardsdatascience.com/neural-network-embeddings-explained-4d028e6f0526

Embedding of Molecules



https://openreview.net/pdf?id=BkSqjHqxg

How to train embeddings

- **Encoder**: data -> embedding
- **Decoder**: embedding -> data

How to train embeddings (alternative)

- **Encoder**: data -> embedding
- **Decoder**: embedding -> ~~data~~ some *feature* of the data

How to train embeddings (denoising)

- **Encoder**: noisy data -> embedding
- **Decoder**: embedding -> denoised data

Word Embeddings

History

- The term “Word embedding” coined in 2003 (Bengio et al.)
- word2vec model proposed in 2013 (Mikolov et al.)
- GloVe vectors released in 2014 (Pennington et al.)

Training Word Embedding

- **Encoder**: word(??) -> embedding
- **Decoder**: embedding -> ???

How do we encode the word?

What is our target?

One-hot embedding

- Used for categorical features
- Each word has its own “index”

One-hot embedding as input to the encoder

- **Encoder**: one-hot embedding -> low-dim embedding
- **Decoder**: low-dim embedding -> ???

Taking a step back

- What data do we have?
    - Large corpus of text (e.g. wikipedia, news article, tweets, etc)
- What determines the "meaning" of a word?
- What properties do we want our embedding space to have?

Text as sequences

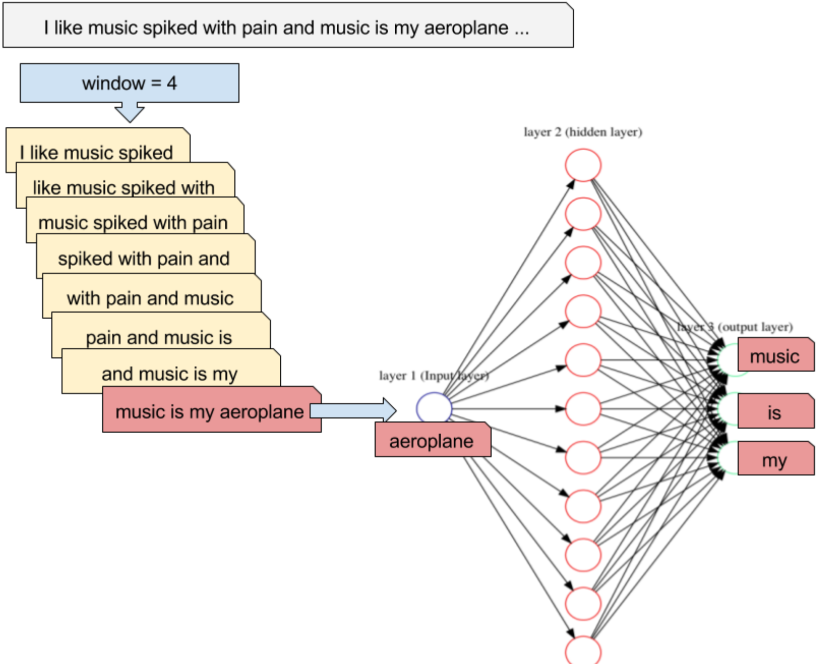
- Hard to find the meaning of a word on its own
- Figure out meaning of words based on its **context** – nearby words

There is evidence that children learns words this way!

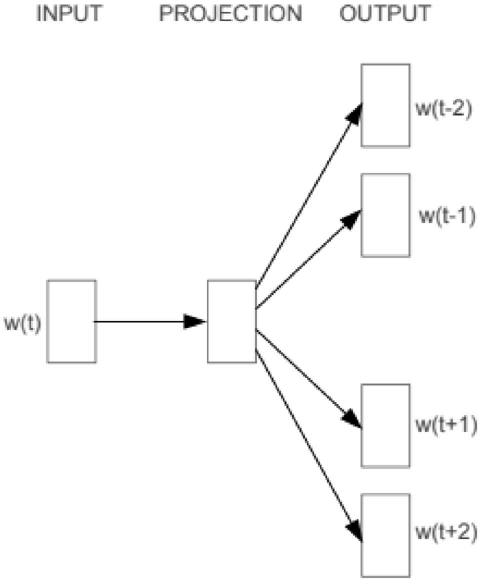
Key idea

- **Encoder**: one-hot embedding -> low-dim embedding
- **Decoder**: low-dim embedding -> **nearby words**

Architecture



Skip-Gram Model



Structure

https://nlp.stanford.edu/projects/glove/
https://github.com/stanfordnlp/GloVe

Language Modelling Tasks

- Text generation, correction, completion, summarization
- Image captioning
- Machine Translation
- Sentiment Analysis
- Named entity detection