Deep Learning
MSDS 631

Images and
Convolutional Neural Networks

Michael Ruddy

Questions?

- From last lecture?
- From the lab assignment?

Overview

- Why imaging? Imaging tasks?
- What/Why is a Convolution?
- Why text? Text tasks?
- Preprocessing Text

Unstructured Data

- Often refer to tabular data as “structured data”

- Thus, we refer to images and text as *unstructured data*
  - Also include things like video (lots of images) and audio (fancy sequence)

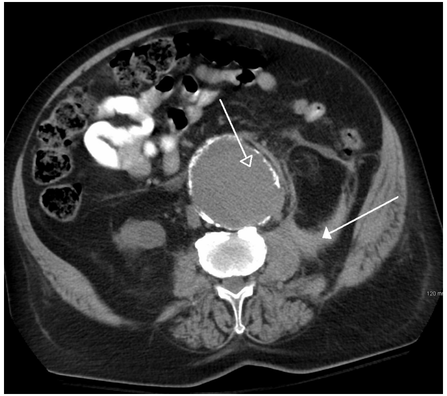
- Structured Data
  - Random Forest vs. Gradient Boosting vs. DL vs. etc
- Unstructured Data
  - Deep Learning reigns supreme

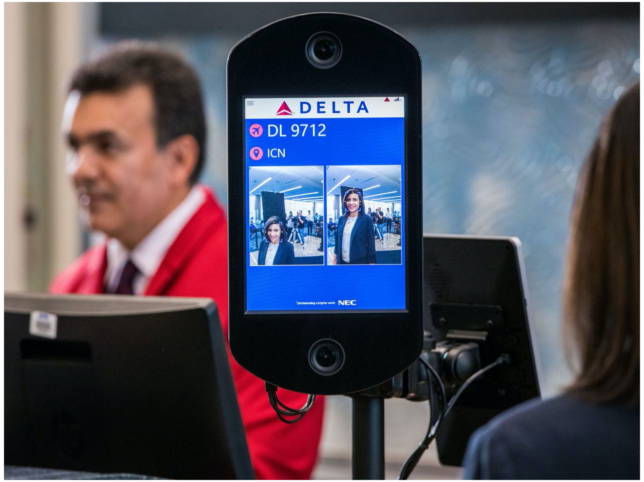
Why Imaging?

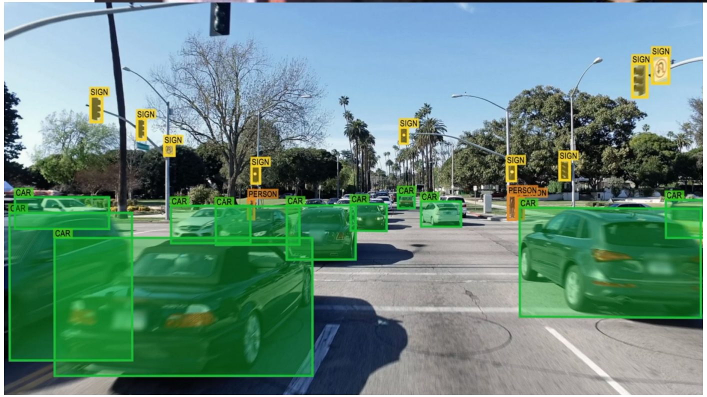
- Humans are really good at looking at things
  - The human eye/brain is an incredibly complicated piece of machinery
- Efforts to recreate vision based on human models of vision have largely been unsuccessful

Why Imaging?

- Humans are really good at looking at things
  - The human eye/brain is an incredibly complicated piece of machinery
- Efforts to recreate vision based on human models of vision have largely been unsuccessful
- Imaging is important!







Why Imaging?

- Humans are really good at looking at things
  - The human eye/brain is an incredibly complicated piece of machinery
- Efforts to recreate vision based on human models of vision have largely been unsuccessful
- Imaging is important!
  - Classification (facial/object recognition, avoid poisonous plants, etc.)
  - Medical Imaging (detecting disease, predicting outcomes of radiation, segmentation of medical images)
  - Autonomous Driving (driver assistance, fully autonomous vehicles)
  - Deepfakes and deepfake detection

Why Imaging?

- Humans are really good at looking at things
  - The human eye/brain is an incredibly complicated piece of machinery
- Efforts to recreate vision based on human models of vision have largely been unsuccessful
- Imaging is important!
  - Classification (facial/object recognition, avoid poisonous plants, etc.)
  - Medical Imaging (detecting disease, predicting outcomes of radiation, segmentation of medical images)
  - Autonomous Driving (driver assistance, fully autonomous vehicles)
  - Deepfakes and deepfake detection
- A lot of these are time-consuming things that human can do really well

Why are images special?

- Images are deceptively hard

Why are images special?

- Images are deceptively hard

This is a cat

Why are images special?

- Images are deceptively hard

This is ??????

$$\begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ .5 & .75 & 1 & \dots & .25 \\ \vdots & \vdots & \vdots & & \vdots \\ .333 & 0 & 1 & \dots & 0 \end{bmatrix}$$

Why are images special?

- Images are deceptively hard
- Images are big

32x32 image
1024 features

512x512 image
262,144 features

Why are images special?

- Images are deceptively hard
- Images are big

32x32 image
1024 features

512x512 image
262,144 features

Fully Connected Layer

- 1024 -> 1024
- 1024^2 = 1,048,576 parameters

- 262,144 -> 262,144
- 68,719,476,736 parameters

Why are images special?

- Images are deceptively hard
- Images are big
- Geometry matters!
  - Pixels near each other interact in different ways to create features than pixels far away

Different relationships

$$\begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ .5 & .75 & 1 & \dots & .25 \\ \vdots & \vdots & \vdots & & \vdots \\ .333 & 0 & 1 & \dots & 0 \end{bmatrix}$$

Why are images special?

- Images are deceptively hard
- Images are big
- Geometry matters!
    - Pixels near each other interact in different ways to create features than pixels far away
    - This is free data that we lose if we simply consider an image as a data vector

Different relationships

$$\begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ .5 & .75 & 1 & \dots & .25 \\ \vdots & \vdots & \vdots & & \vdots \\ .333 & 0 & 1 & \dots & 0 \end{bmatrix}$$

The Convolution

- Fancy **linear** operation useful for spatial data

The Convolution

- Fancy **linear** operation useful for spatial data

$$\begin{bmatrix} 1 & .5 & 1 & 0 \\ 0 & .25 & .5 & 1 \\ 1 & .25 & 0 & 1 \\ .5 & 0 & 1 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$$

The Convolution

- Fancy **linear** operation useful for spatial data

Grayscale Image

Filter (kernel)

$$\begin{bmatrix} 1 & .5 & 1 & 0 \\ 0 & .25 & .5 & 1 \\ 1 & .25 & 0 & 1 \\ .5 & 0 & 1 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$$

The Convolution

- Fancy **linear** operation useful for spatial data

Grayscale Image

Filter (kernel)

$$\begin{bmatrix} 1 & .5 & 1 & 0 \\ 0 & .25 & .5 & 1 \\ 1 & .25 & 0 & 1 \\ .5 & 0 & 1 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} = \begin{bmatrix} \quad & \quad \\ \quad & \quad \end{bmatrix}$$

The Convolution

- Fancy **linear** operation useful for spatial data
- Element-wise product

$$(1 \times 1) + (.5 \times 0) + (0 \times 0) + (.25 \times 2)$$
$$= 1.5$$

Grayscale Image

Filter (kernel)

$$\begin{bmatrix} \boxed{\begin{matrix} 1 & .5 \\ 0 & .25 \end{matrix}} & \begin{matrix} 1 & 0 \\ .5 & 1 \end{matrix} \\ \begin{matrix} 1 & .25 \\ .5 & 0 \end{matrix} & \begin{matrix} 0 & 1 \\ 1 & 1 \end{matrix} \end{bmatrix} * \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} = \begin{bmatrix} \boxed{1.5} & & \\ & & \\ & & \end{bmatrix}$$

The Convolution

- Fancy **linear** operation useful for spatial data
- Element-wise product

$$(.5 \times 1) + (1 \times 0) + (.25 \times 0) + (.5 \times 2) = 1.5$$

Grayscale Image

Filter (kernel)

$$\begin{bmatrix} 1 & .5 & 1 & 0 \\ 0 & .25 & .5 & 1 \\ 1 & .25 & 0 & 1 \\ .5 & 0 & 1 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} = \begin{bmatrix} 1.5 & \boxed{1.5} & \\ & & \\ & & \end{bmatrix}$$

The Convolution

- Fancy **linear** operation useful for spatial data
- Element-wise product

$$(1 \times 1) + (0 \times 0) + (.5 \times 0) + (1 \times 2) = 3$$

Grayscale Image

Filter (kernel)

$$\begin{bmatrix} 1 & .5 & 1 & 0 \\ 0 & .25 & .5 & 1 \\ 1 & .25 & 0 & 1 \\ .5 & 0 & 1 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} = \begin{bmatrix} 1.5 & 1.5 & 3 \\ & & \\ & & \end{bmatrix}$$

The Convolution

- Fancy **linear** operation useful for spatial data
- Element-wise product

$$(0 \times 1) + (.25 \times 0) + (1 \times 0) + (.25 \times 2) = .5$$

Grayscale Image

Filter (kernel)

$$\begin{bmatrix} 1 & .5 & 1 & 0 \\ 0 & .25 & .5 & 1 \\ 1 & .25 & 0 & 1 \\ .5 & 0 & 1 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} = \begin{bmatrix} 1.5 & 1.5 & 3 \\ \boxed{0.5} & & \\ & & \end{bmatrix}$$

The Convolution

- Fancy **linear** operation useful for spatial data
- Element-wise product

Grayscale Image

Filter (kernel)

$$\begin{bmatrix} 1 & .5 & 1 & 0 \\ 0 & .25 & .5 & 1 \\ 1 & .25 & 0 & 1 \\ .5 & 0 & 1 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} = \begin{bmatrix} 1.5 & 1.5 & 3 \\ 0.5 & ? & ? \\ ? & ? & ? \end{bmatrix}$$

The Convolution

- Fancy **linear** operation useful for spatial data
- Element-wise product

Grayscale Image

Filter (kernel)

$$\begin{bmatrix} 1 & .5 & 1 & 0 \\ 0 & .25 & .5 & 1 \\ 1 & .25 & 0 & 1 \\ .5 & 0 & 1 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} = \begin{bmatrix} 1.5 & 1.5 & 3 \\ 0.5 & .25 & 2.5 \\ 1 & 2.25 & 2 \end{bmatrix}$$

The Convolution

- Fancy **linear** operation useful for spatial data
- Element-wise product

Grayscale Image

Unknown Parameters

$$\begin{bmatrix} 1 & .5 & 1 & 0 \\ 0 & .25 & .5 & 1 \\ 1 & .25 & 0 & 1 \\ .5 & 0 & 1 & 1 \end{bmatrix} * \begin{bmatrix} w_{11}^{(1)} & w_{12}^{(1)} \\ w_{21}^{(1)} & w_{22}^{(1)} \end{bmatrix} = \boxed{}$$

“2x2 Filter”

Why Convolution?

- Only four parameters!
    - If input is dimension 16 and output is dimension 9, how many for FC?

Grayscale Image

Unknown Parameters

$$\begin{bmatrix} 1 & .5 & 1 & 0 \\ 0 & .25 & .5 & 1 \\ 1 & .25 & 0 & 1 \\ .5 & 0 & 1 & 1 \end{bmatrix} * \begin{bmatrix} w_{11}^{(1)} & w_{12}^{(1)} \\ w_{21}^{(1)} & w_{22}^{(1)} \end{bmatrix} =$$

"2x2 Filter"

Why Convolution?

- Only four parameters!
- Translational Equivariance
    - If I shift my image, I shift the output!

Grayscale Image

Unknown Parameters

$$\begin{bmatrix} 1 & .5 & 1 & 0 \\ 0 & .25 & .5 & 1 \\ 1 & .25 & 0 & 1 \\ .5 & 0 & 1 & 1 \end{bmatrix} * \begin{bmatrix} w_{11}^{(1)} & w_{12}^{(1)} \\ w_{21}^{(1)} & w_{22}^{(1)} \end{bmatrix} =$$

"2x2 Filter"

Why Convolution?

- Only four parameters!
- Translational Equivariance
- Weight Sharing (detect same feature translated to different parts of the image)

Grayscale Image

Unknown Parameters

$$\begin{bmatrix} 1 & .5 & 1 & 0 \\ 0 & .25 & .5 & 1 \\ 1 & .25 & 0 & 1 \\ .5 & 0 & 1 & 1 \end{bmatrix} * \begin{bmatrix} w_{11}^{(1)} & w_{12}^{(1)} \\ w_{21}^{(1)} & w_{22}^{(1)} \end{bmatrix} = \boxed{}$$

"2x2 Filter"

Why Convolution?

Intuition: [Edge Detection]

- Only four parameters!
- Translational Equivariance
- Weight Sharing (detect same feature translated to different parts of the image)

Grayscale Image

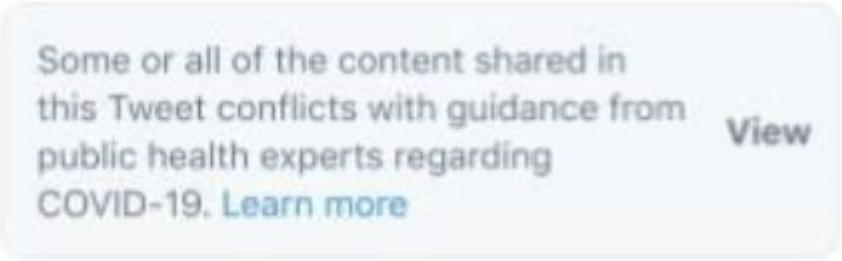
Unknown Parameters

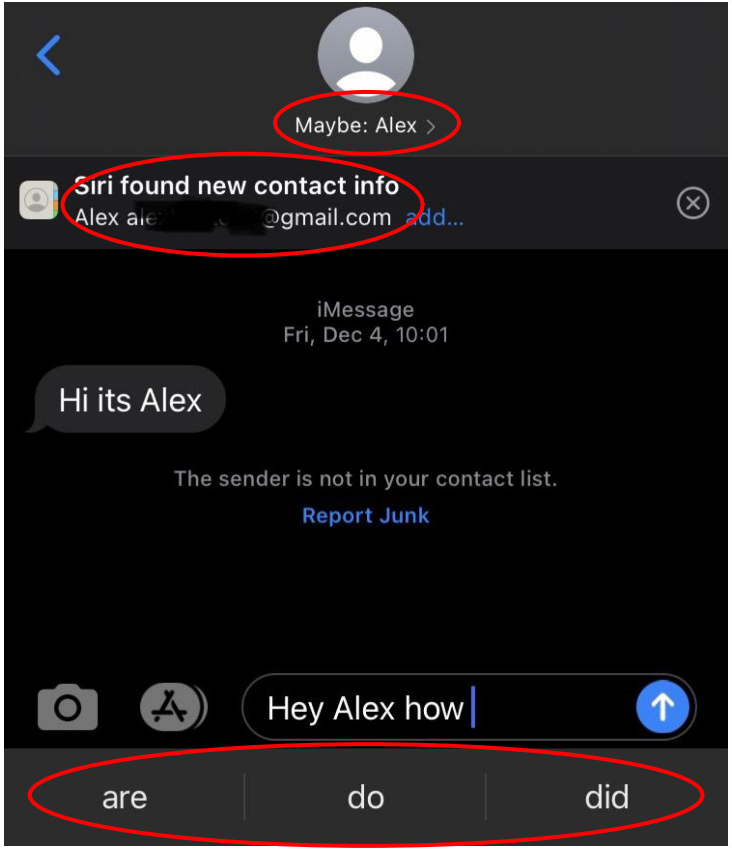
$$\begin{bmatrix} 1 & .5 & 1 & 0 \\ 0 & .25 & .5 & 1 \\ 1 & .25 & 0 & 1 \\ .5 & 0 & 1 & 1 \end{bmatrix} * \begin{bmatrix} w_{11}^{(1)} & w_{12}^{(1)} \\ w_{21}^{(1)} & w_{22}^{(1)} \end{bmatrix} = \boxed{}$$

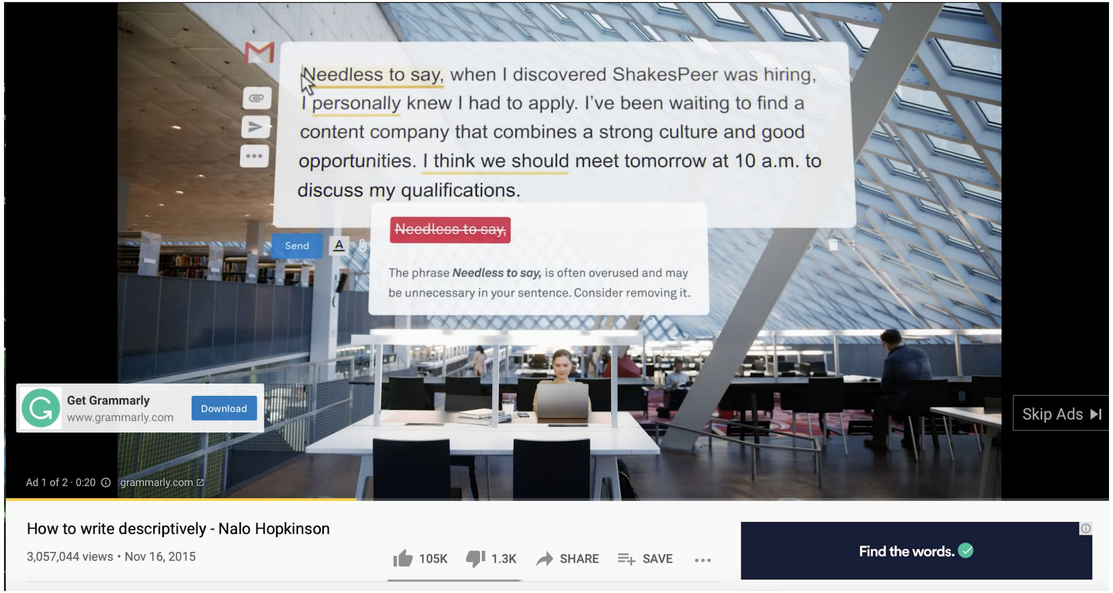
"2x2 Filter"

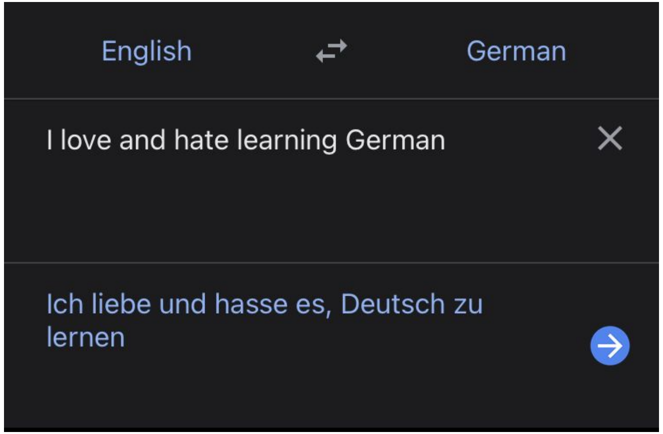
Why Natural Language Processing?

- Understand, analyze, and perform tasks using human language (through text).
- Example Tasks:
    - Sentiment Analysis
    - Auto-complete
    - Translation
    - Question answering
    - Conversation?!






Needless to say, when I discovered ShakesPeer was hiring, I personally knew I had to apply. I've been waiting to find a content company that combines a strong culture and good opportunities. I think we should meet tomorrow at 10 a.m. to discuss my qualifications.
Send
~~Needless to say,~~
The phrase Needless to say, is often overused and may be unnecessary in your sentence. Consider removing it.
Get Grammarly
www.grammarly.com
Download
Skip Ads
Ad 1 of 2 · 0:20
grammarly.com
How to write descriptively - Nalo Hopkinson
3,057,044 views • Nov 16, 2015
105K
1.3K
SHARE
SAVE
Find the words.


English
German
I love and hate learning German
Ich liebe und hasse es, Deutsch zu lernen

NLP is hard!

- How to represent text as data?

$$\begin{bmatrix} 1 & .5 & 1 & 0 \\ 0 & .25 & .5 & 1 \\ 1 & .25 & 0 & 1 \\ .5 & 0 & 1 & 1 \end{bmatrix}$$

NLP is hard!

- How to represent text as data?
- Humans represent text using characters
  - Takes years to learn to read
  - Different peoples do it differently all around the world

$$\longrightarrow \begin{bmatrix} 1 & .5 & 1 & 0 \\ 0 & .25 & .5 & 1 \\ 1 & .25 & 0 & 1 \\ .5 & 0 & 1 & 1 \end{bmatrix}$$

train

brain

head

NLP is hard!

- How to represent text as data?
- Humans represent text using characters
  - Takes years to learn to read
  - Different peoples do it differently all around the world

$$\rightarrow \begin{bmatrix} 1 & .5 & 1 & 0 \\ 0 & .25 & .5 & 1 \\ 1 & .25 & 0 & 1 \\ .5 & 0 & 1 & 1 \end{bmatrix}$$

train $\longrightarrow$ 20-18-1-9-14

brain $\longrightarrow$ 2-18-1-9-14

head $\longrightarrow$ 8-5-1-4

NLP is hard!

- How to represent text as data?
- Humans represent text using characters
    - Takes years to learn to read
    - Different peoples do it differently all around the world
- For most tasks this is not a particularly helpful embedding
    - Intrinsic meaning is largely lost

$$\longrightarrow \begin{bmatrix} 1 & .5 & 1 & 0 \\ 0 & .25 & .5 & 1 \\ 1 & .25 & 0 & 1 \\ .5 & 0 & 1 & 1 \end{bmatrix}$$

train ⟶ 20-18-1-9-14

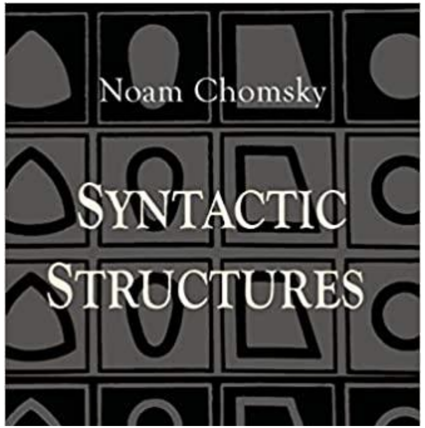
brain ⟶ 2-18-1-9-14

head ⟶ 8-5-1-4

NLP is hard!

- How to represent text as data?
- Humans represent text using characters
    - Takes years to learn to read
    - Different peoples do it differently all around the world
- For most tasks this is not a particularly helpful embedding
    - Intrinsic meaning is largely lost

$$\begin{bmatrix} 1 & .5 & 1 & 0 \\ 0 & .25 & .5 & 1 \\ 1 & .25 & 0 & 1 \\ .5 & 0 & 1 & 1 \end{bmatrix}$$



train → 20-18-1-9-14

brain → 2-18-1-9-14

head → 8-5-1-4

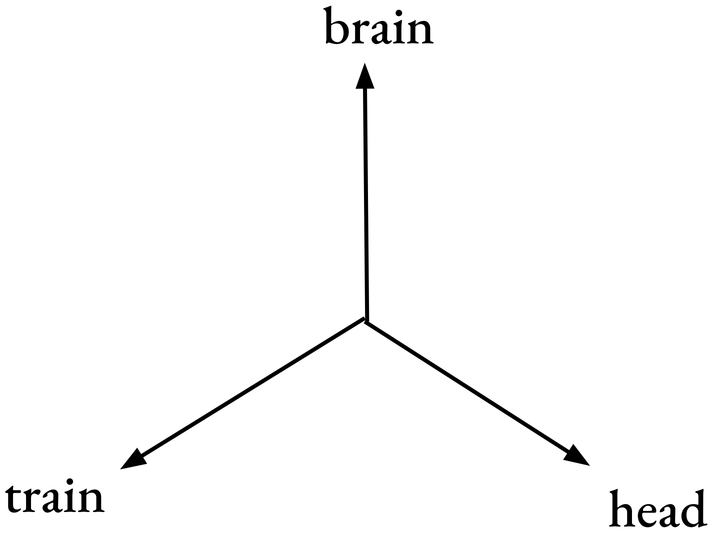
Tokenization

- Idea: Break up text into pieces (tokens) and treat as categorical variables
    - Often these tokens are words

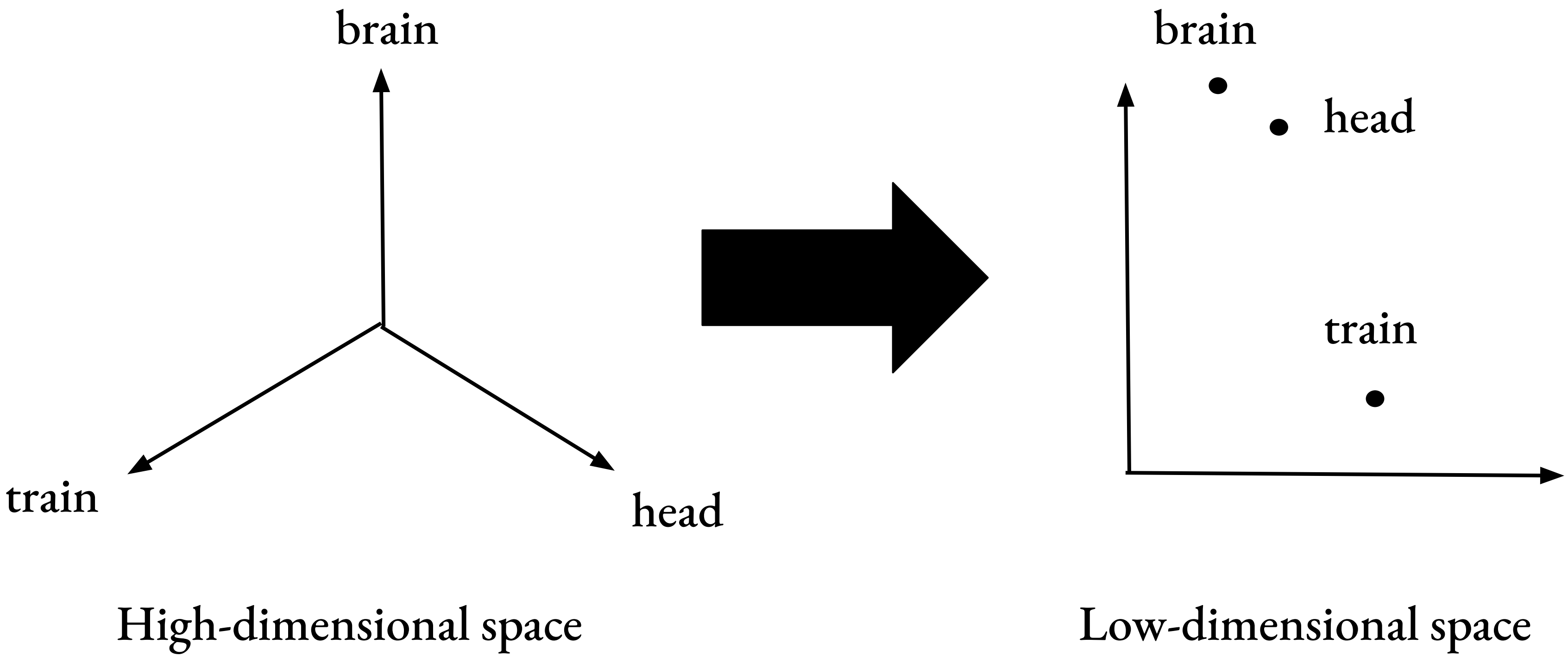
Tokenization

- Idea: Break up text into pieces (tokens) and treat as categorical variables
    - Often these tokens are words

$$\underset{\text{a}}{\begin{bmatrix}1\\0\\0\\\vdots\\0\end{bmatrix}} \quad \underset{\text{aadvark}}{\begin{bmatrix}0\\1\\0\\\vdots\\0\end{bmatrix}} \quad \cdots \quad \underset{\text{zebra}}{\begin{bmatrix}0\\0\\0\\\vdots\\1\end{bmatrix}}$$



Word Embedding



Other Types of Tokenization

- Idea: Break up text into pieces (tokens) and treat as categorical variables
- Words -> Tokens

Other Types of Tokenization

- Idea: Break up text into pieces (tokens) and treat as categorical variables
- Words -> Tokens
  - N-grams: Common phrases as one token instead of separate tokens

data_science vs. data, science

Other Types of Tokenization

- Idea: Break up text into pieces (tokens) and treat as categorical variables
- Words -> Tokens
- Characters -> Tokens

train $\longrightarrow$ 20-18-1-9-14

brain $\longrightarrow$ 2-18-1-9-14

head $\longrightarrow$ 8-5-1-4

Other Types of Tokenization

- Idea: Break up text into pieces (tokens) and treat as categorical variables
- Words -> Tokens
- Characters -> Tokens
- Sub-words -> Tokens
  - Break up words into smaller tokens
  - Smaller dictionary, less total tokens
  - Better at handling unknown, less lemmatization

Unfortunately -> un + fortunate + ly
skiing -> ski + ing

Other Types of Tokenization

- Idea: Break up text into pieces (tokens) and treat as categorical variables
- Words -> Tokens
- Characters -> Tokens
- Sub-words -> Tokens
  - Break up words into smaller tokens
  - Smaller dictionary, less total tokens
  - Better at handling unknown, less lemmatization
  - Many Algorithms: BPE, Unigram, WordPiece

Unfortunately -> un + fortunate + ly

skiing -> ski + ing

Other Types of Tokenization

- Idea: Break up text into pieces (tokens) and treat as categorical variables
- Words -> Tokens
- Characters -> Tokens
- Sub-words -> Tokens
- Sentence Segmentation
    - EOS (End of Sentence) and SOS (Start of Sentence) tokens are common

Other Types of Tokenization

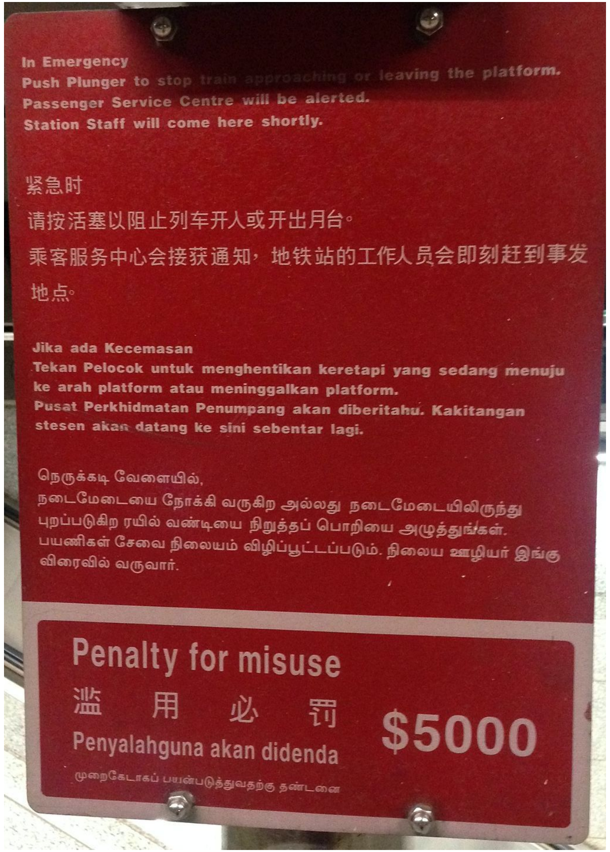
- Idea: Break up text into pieces (tokens) and treat as categorical variables
- Words -> Tokens
- Characters -> Tokens
- Sub-words -> Tokens
- Sentence Segmentation
  - EOS (End of Sentence) and SOS (Start of Sentence) tokens are common
  - Non-trivial to find these!
  - Binary Classifier, complicated logic trees

Can't just rely on periods!

The U.K. exports of goods and services as percent of GDP was 31.6% in 2019.

Other Types of Tokenization

- Idea: Break up text into pieces (tokens) and treat as categorical variables
- Words -> Tokens
- Characters -> Tokens
- Sub-words -> Tokens
- Sentence Segmentation
- Other languages:
    - Chinese languages, Arabic, French, etc.



Word Tokenization Techniques

- Lemmatization
  - Reduce words to their base
  - Shrink dictionary size

running -> run
mice -> mouse

Word Tokenization Techniques

- Lemmatization
- Infrequent words (misspelled or weird words)
  - Remove from text or encode as single UNK token

Word Tokenization Techniques

- Lemmatization
- Infrequent words (misspelled or weird words)
- Cleaning before tokenization
  - Lower case
  - Remove weird characters/numbers/punctuation
  - Remove stop words

the, to, a, an, etc.

Word Tokenization Techniques

- Lemmatization
- Infrequent words (misspelled or weird words)
- Cleaning before tokenization
  - Lower case
  - Remove weird characters/numbers/punctuation
  - Remove stop words
- Named Entity Recognition

Apple vs. apple
Xerox vs. xerox

Deep Learning and NLP

- Sequences
    - Variable length
    - Relationships between elements of sequence
- Continuous Bag of Words (CBOW)
- 1D CNN
- Recurrent Neural Network (RNN)
    - Keep track of a hidden state vector of features as you move along a sequence
    - Sequence length agnostic

Summary

- Images and Text are special
  - Humans are better at seeing than speaking
  - Language is “harder” than vision

- Special architectures exist to take advantage of the unique properties
  - Images: spatial
  - Text: sequences

- NLP requires a lot of preprocessing and thinking deeply about representation