Department of Information Systems
and Computer Science

Data Representation

CS 114

Outline

Text

- How is text represented
- Probabilistic Text
- Sequential Text

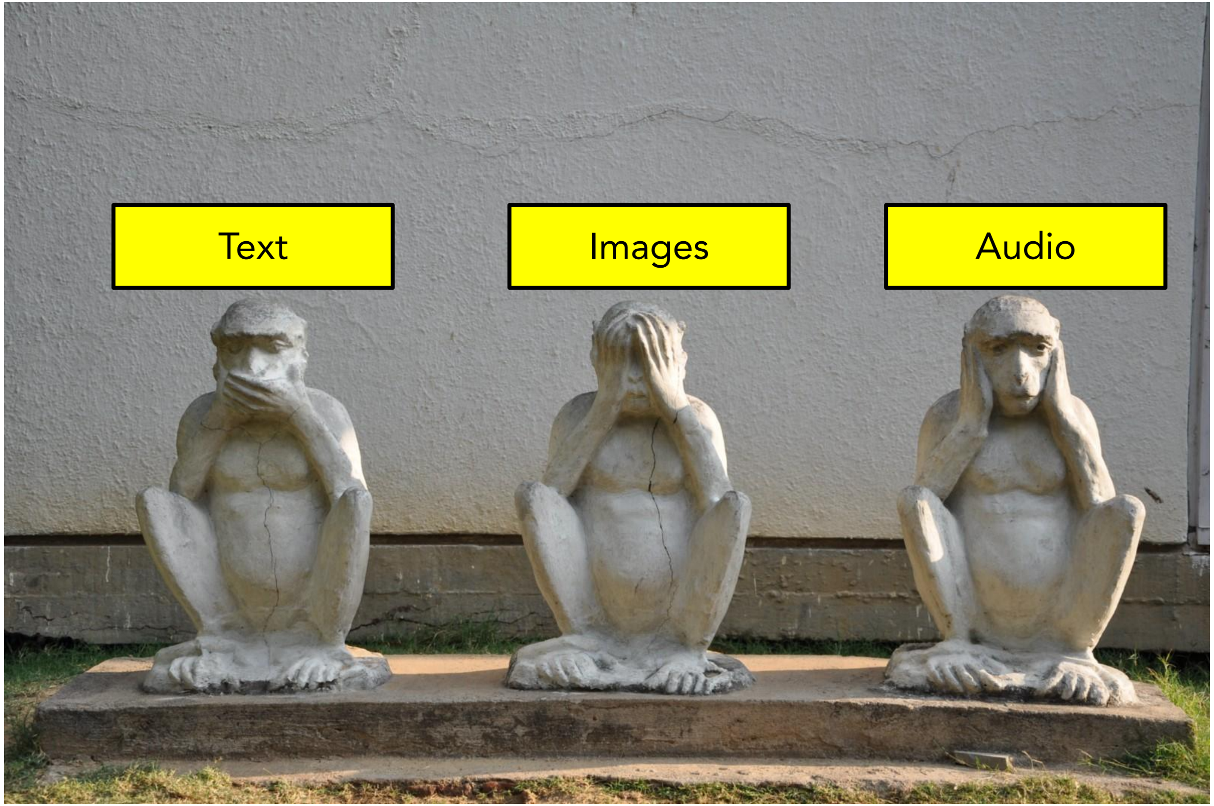
Images

- How are images represented
- Pixel Values
- Convolutions
- Dealing with Video

Audio

- How is audio represented
- Signal Processing

Modes of Data



Text

Text

- Text data can be seen as a sequence of alphanumeric characters in its most granular form
- Treating it as a feature vector poses a challenge because of granularity
  - Letter per letter sequence is too noisy
  - Words are too general without context
  - Sentences are hard to comprehend without granularity
- Text data is often characterized as <u>sparse data</u>
- We often first have to define a vocabulary (letter-wise / word-wise / sentence-wise)

Text and Probability

- An inference to a probabilistic outcome P(y | x) is not scalable when dealing with sparse data
- Bayesian Inference:

```
P(A|B) = (P(B|A) P(A)) / P(B)
```

```
P(y|x) = (P(x|y) P(y)) / P(x)
```

```
P("b will occur" | "a") = P("a occurences" | "b") P("b instances") / P("a")
```

```
P("b will occur" | "a") = P("a occurences" | "b") P("b instances")
P("c will occur" | "a") = P("a occurences" | "c") P("c instances")
```

Text and Probability

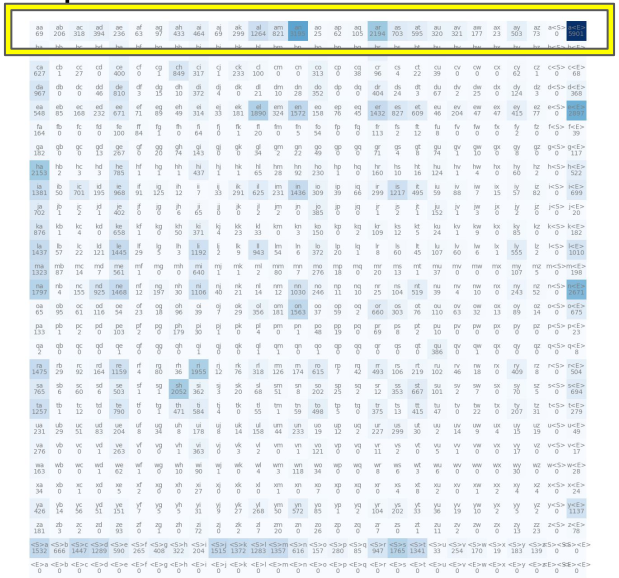
- Bayesian inference on sparse data deals with sparse multiplication operations due to Joint Probability
- Computation on bayesian inference even on just the numerator will be performed over and over again for every instance of "y"

```
P("b will occur" | "a") = P("a occurences" | "b") P("b instances")
P("c will occur" | "a") = P("a occurences" | "c") P("c instances")
```

This is only for the possibilities of "b" vs "c". What about for a vocabulary of 10,000 possibilities?

Text Data for Name Generation

- Bigram Model: Learned weights W is a count probability of two consecutive letters. By simply increasing the window size of previous letters, we run into computationally expensive probabilities.



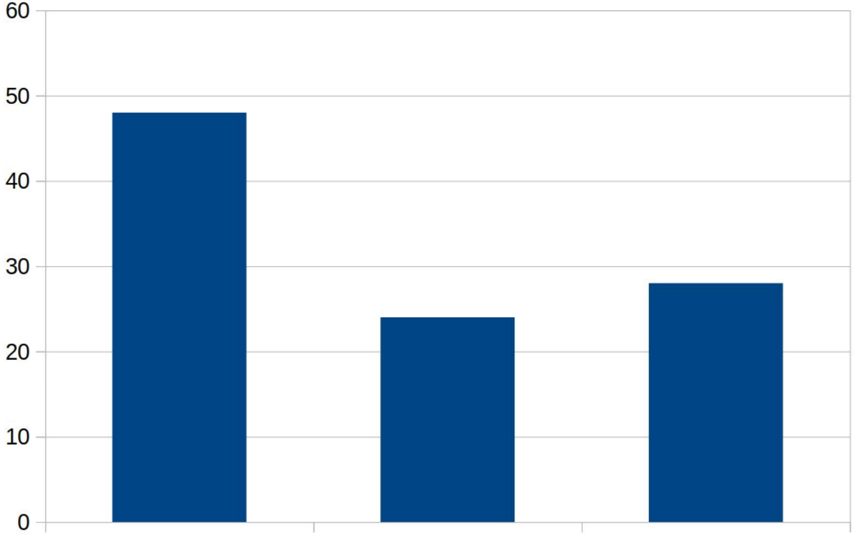
Count Occurences: { “a → b”: 10, “a →c”: 5, “a → d”: 6 }

{ “a → b”: 10, “a →c”: 5, “a → d”: 6 }

{ “a → b”: 48%, “a →c”: 24%, “a → d”: 28% }

Problem with Probability

- Text Generation: Randomly sample an item from the probability distribution with an assumption that expresses a multinomial distribution



{ “a → b”: 48%, “a →c”: 24%, “a → d”: 28% }

There is a 48% chance that a “b” comes after an “a”, a 24% chance that a “c” comes after an “a” and a 28% chance that a “d” comes after an “a”. When randomly sampled, these weighted parameters are applied to generate the next letter.

```
f(x) = x * W
```

Usefulness of Probability

- Modeling the outcome of the next word or token
- Determine if a model f(x) can generate the next letter given a previous (or multiple previous letters):

Vocabulary: [“a”, “b”, “c”]

A model is sure to get an “a” next if the probability outcome is:

[1, 0, 0]

**Notice that all values sum up to 1 or 100%*

f(“b”) = x * W

For some optimized W, when multiplied to x we get some outcome. **However, we are not guaranteed that W will yield a probability distribution (that is, all values sum to 1).** Example:

[0.52, 0.57, 0.58]

Softmax

- A mathematical function that takes a vector of arbitrary values and returns a probability distribution over it.

$$s\left(x_i\right) = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}}$$

[0.52, 0.57, 0.58]

s(0.52) = 1.682 / (1.682 + 1.768 + 1.786) = 0.32
s(0.57) = 1.768 / (1.682 + 1.768 + 1.786) = 0.34
s(0.57) = 1.768 / (1.682 + 1.768 + 1.786) = 0.34

Embeddings

- A way to represent words or phrases or sequences as a vector of numbers
- Instead of treating each “token” in a language as possible x input, we could consider it compressed into an embedding that could also represent other tokens of similar nature
- Objective: Turn a sparse (raw) input into something more compact and understandable. Figure out some W that generalizes the representation.
- Therefore, an embedding is just a smaller representation of some input transformed by W.

Embeddings

- Why are embeddings relevant in text data? Let's take a look at an example:

```
The dog ate my homework
```

```
The cat ate some catfood
```

What word can we substitute to make this sentence make sense?

```
The ______ crossed the road
```

Embeddings

```
The dog ate my homework
```

Some value to say that “The” refers to “dog” which is an animal

Some animal usually comes before a verb like “ate”

“my homework” refers to some subject

```
Embedding: Some set of values that strongly relates to:

[refer, animal, verb, subject]
```

Embeddings

```
The cat ate some catfood
```

Some magical structure W to create the embeddings

```
f([“The”, “cat”, “ate”, “some”, “catfood”]) = x * W

= [refer, animal, verb, subject]
= [P(refer), P(animal), P(verb), P(subject)]
```

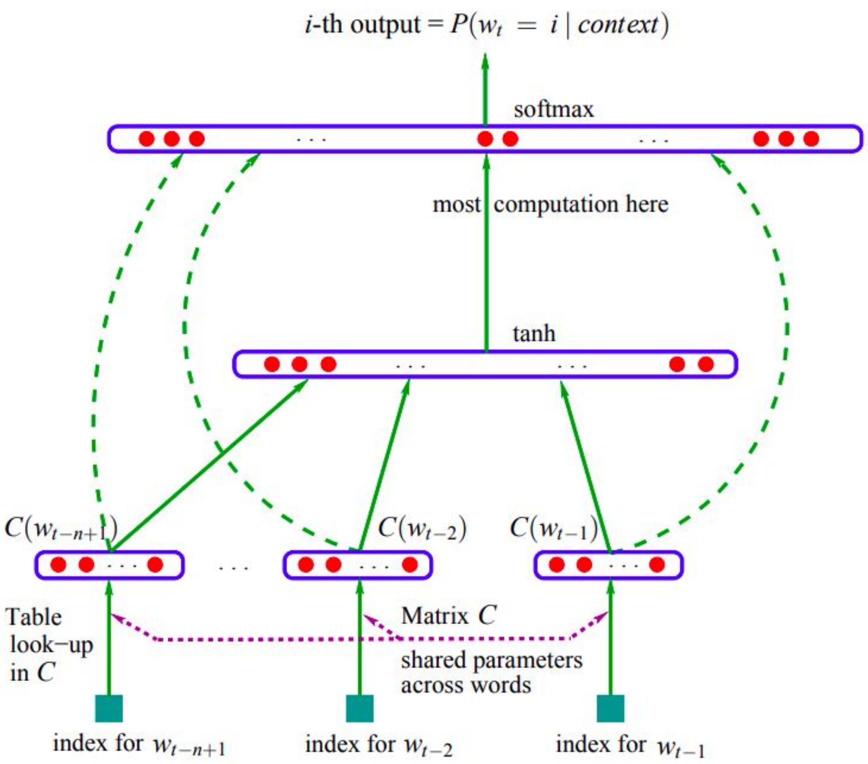
Embeddings

- Given some learned W to represent embeddings and outcomes (and everything in between for that matter), we can then gerate the embedding that most likely will complete the following:

```
The ______ crossed the road
```

Magical structure W should then most likely associate the missing piece with an animal given that most of the parts of the sentence would express similar embedding values. We could sample it from our vocabulary if we know what W looks like.

Bengio's MLP



Softmax layer representing the next part of the sequence that is token at t + 1

Hidden layer that activates [-1, 1] with the tanh function.

An embedding of size 30 that represents each of the input token with shared W in between

A set of 3 tokens to represent the input. Sparsely, input size is 17k.

Comparing Text

- How good is my generated word or sentence?
- COSINE Similarity: If text A and text B can be turned into some form of embeddings, we can apply a difference function to see how close they are to each other

Example

I like programming and pizza.

I like pineapple in pizza.

How similar are these sentences?

Vocabulary

- I
- like
- programming
- pineapple
- and
- in
- pizza

Word Count

Vocabulary	Sentence 1	Sentence 2
I	1	1
like	1	1
programming	1	0
pineapple	0	1
and	1	0
in	1	1
pizza	1	1

Cosine Similarity

- Similarity measure between two non-zero vectors
- Takes the inner product space that measures the cosine of the angle between them

$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{||\mathbf{A}|| \, ||\mathbf{B}||} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}$$

Computation

- Sentence 1 = [1, 1, 1, 0, 1, 1, 1]
- Sentence 2 = [1, 1, 0, 1, 0, 1, 1]

A dot B = (1*1) + (1*1) + (1*0) + (0*1) + (1*1) + (1*1) = 4

||A|| = sqrt(1^2 + 1^2 + 1^2 + 0^2 + 1^2 + 1^2 + 1^2) = 2.4495

||B|| = sqrt(1^2 + 1^2 + 0^2 + 1^2 + 0^2 + 1^2 + 1^2) = 2.2361

Cosine = (A dot B) / (||A|| ||B||) = 4 / (2.4495 * 2.2361) = 0.7302383227

Evaluation of Cosine

- Output will receive a value between -1 to 1
- -1: Completely Dissimilar
  - The cat is sleeping
  - The cat is not sleeping
- 1: Completely Similar
  - The cat is sleeping.
  - The kitten is napping.
- 0: Indicates that they are orthogonal (not similar at all)
  - The cat is sleeping
  - The dog is barking

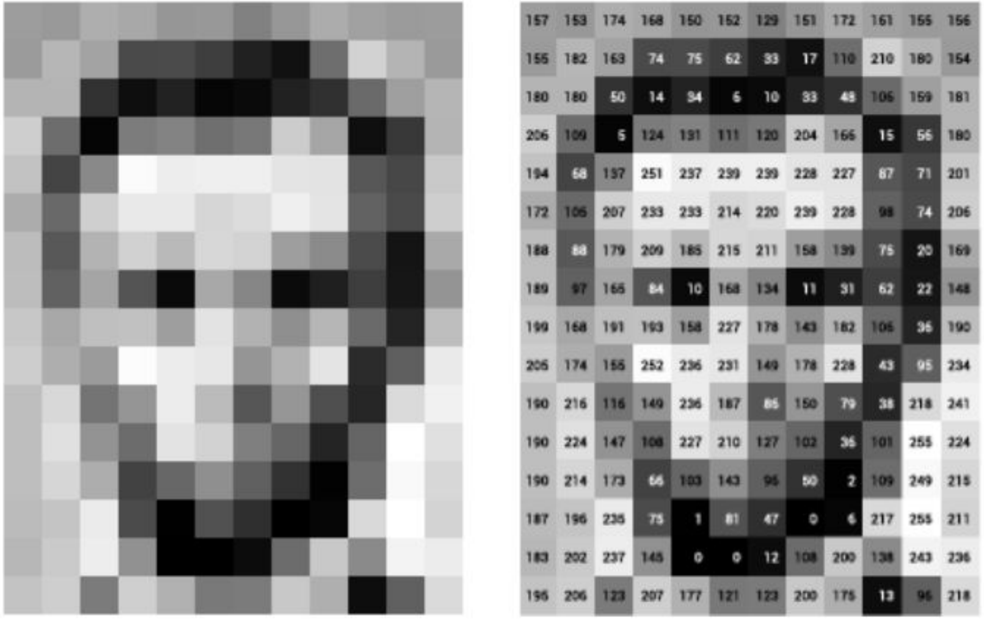
Guide Questions

- What are potential problems of cosine similarity based on count?
- Can similarity measures be applied to a probabilistic framework when dealing with text?

Images

Pixel Data

- Pixel Data: Dimensionality = Length x Width



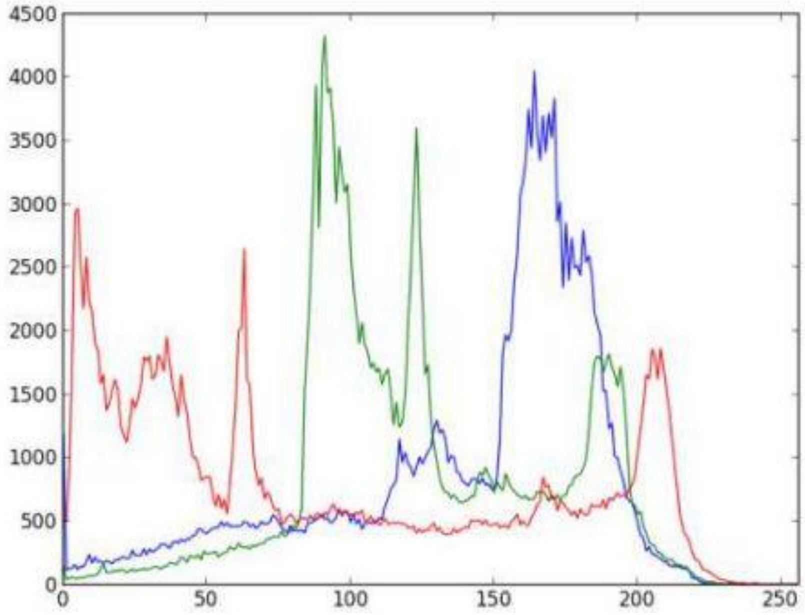
157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

Global Features

- Characteristics or patterns in an image that describe the entire image as a whole
- Although global features can be represented as multi-dimensional data, it only captures the gist of an image as opposed to local characteristics that compose an image
- Examples of Global Features:
    - Histograms
    - Texture

Histograms

- Count of similar pixel values
- Question: What are problems with granularity with histograms?



GLCM Computation

- Gray Level Co-occurrence Matrix

Statistics Group:

GLCM Mean	$\mu_i = \sum_{i,j=1}^{N} i(P_{i,j})$ $\mu_j = \sum_{i,j=1}^{N} j(P_{i,j})$ $\mu = \frac{(\mu_i + \mu_j)}{2}$
GLCM Variance	$\sigma_i^2 = \sum_{i,j=1}^{N} P_{i,j}(i - \mu_i)^2$ $\sigma_j^2 = \sum_{i,j=1}^{N} P_{i,j}(j - \mu_j)^2$ $\sigma^2 = \frac{(\sigma_i^2 + \sigma_j^2)}{2}$
GLCM Correlation	$\sum_{i,j=1}^{N} P_{i,j} \left[\frac{(i - \mu_i)(j - \mu_j)}{\sqrt{\sigma_i^2 \sigma_j^2}} \right]$

Local Binary Patterns

- Like GLCM, LBP is used to determine texture features in a given image
- Captures local texture variations by encoding the relationship between a center pixel and its nth surrounding neighbors
- LBP1:

5	8	1
5	4	1
3	7	2

→

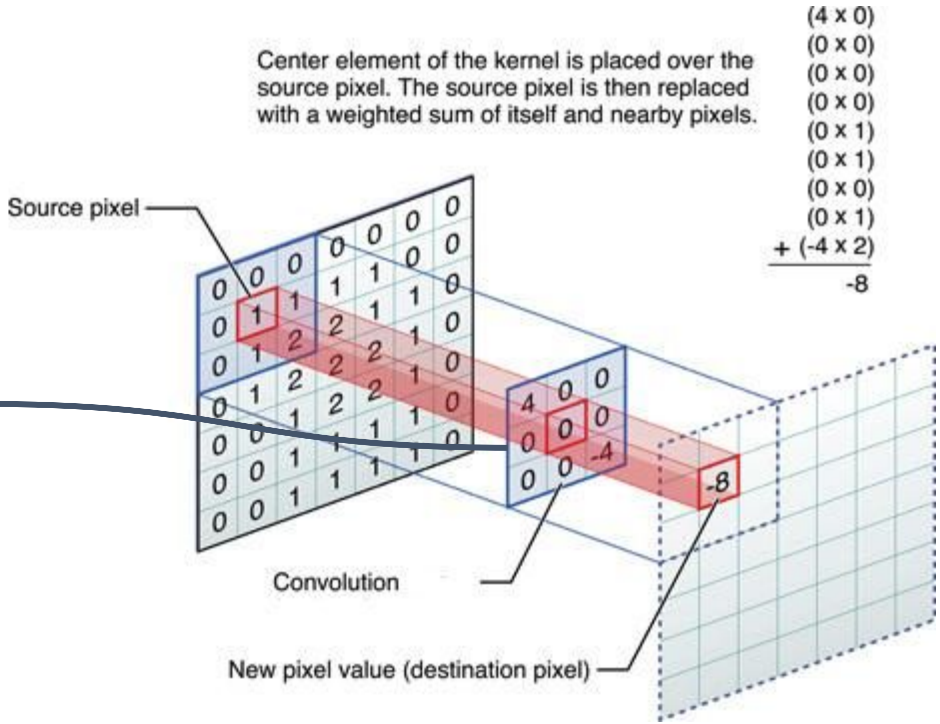
0	0	1
0		1
1	0	1

Deep Learning Features

- High level representation of image data that are learned by deep learning networks
- Creates meaningful patterns, structures and semantic information present in the data
- Semantically Meaningful Features: Encodes information about the content of an image such as the presence (or non-presence) of an object regardless of its location or orientation

Convolutional Features

- Convolution Operation

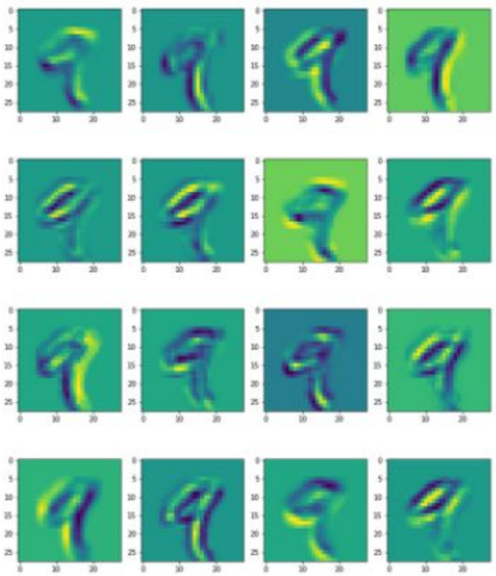


Kernel:
Filter to convolve against a region in the image

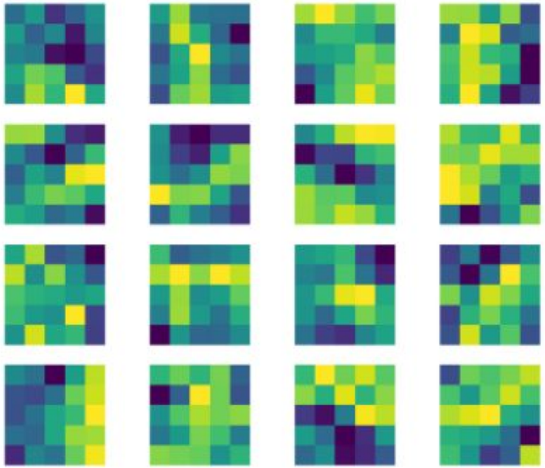
Feature Map:
Resulting signal after convolution.

Convolutional Features

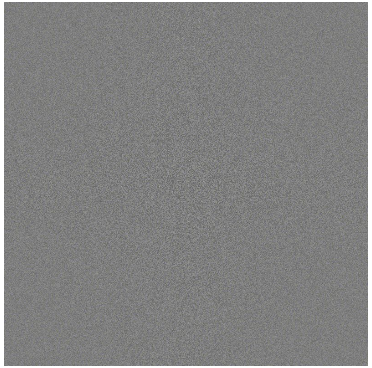
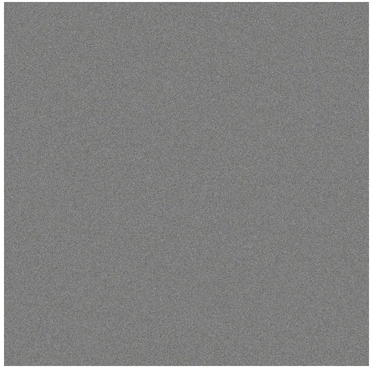
Feature map



Filters



Pixel Dreams

Guide Questions

- What are the potential weaknesses of non-deep learning features?
- What about videos? How do you think videos can be represented as data?

Audio

Audio Signal

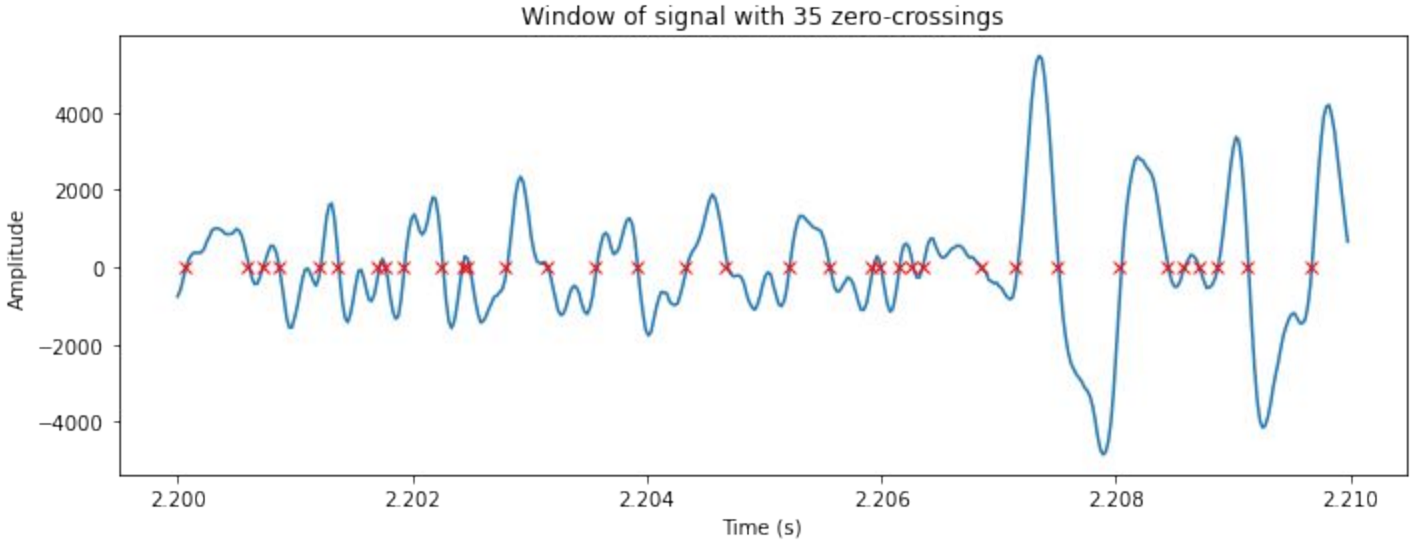
- Representation of sound waves that can be processed and analyzed by a computer
- A sequence of numerical values that represent the instantaneous amplitude of a sound wave at discrete time points

Temporal Features

- Temporal: A feature in relation to time or some chronological order
- Basic Examples:
  - Zero Cross Rate
  - Root Mean Square Energy
  - Temporal Centroid

Zero Cross Rate

- Number of times a signal crosses the zero level within a specified time frame
- Characterizes the frequency content and noisiness of a signal



Department of Information Systems
and Computer Science

Root Mean Square Energy

- Energy of an audio signal over time

$$x_{\mathrm{rms}} = \sqrt{\frac{1}{n}\left(x_1^2 + x_2^2 + \cdots + x_N^2\right)}$$

Temporal Centroid

- Provides insight into where the “center” of the audio’s energy distribution is over time.
- Measured by the temporal location of the average energy of an audio signal

```
total_energy = sum(x ** 2 for x in audio_signal)
temporal_centroid = sum((i / len(audio signal)) * (x ** 2)
for i, x in enumerate(audio_signal)) / total_energy
```

Questions?

Thank you!