Announcements

Exams graded, will release solutions soon

Continue to work on your final projects

Wed 25th pre-recorded lecture
Herding Cats
How modeling turned into modern AI
Recap: Model Fitting

**Fitting:** Observe $x_i$, $y_i$, infer $\theta$

Regression Problems -> Continuous $Y$

Classification Problems -> Discrete $Y$
Recap: Model Fitting - Machine Learning

**Fitting:** Observe $x_i$, $y_i$, infer $\theta$

- Regression Problems $\rightarrow$ Continuous $Y$
- Classification Problems $\rightarrow$ Discrete $Y$
Recap: High Dimensional Data

“More Observations Than Individuals”

**Stocks:** 1000s of securities, Millions of trades

**Genomics:** 6.4 bn base pairs, 100s Clinical Population

**Natural Language:** Inf possible word sequences, Millions of Documents
Principal Component Analysis

Identify low dimensional structure in the data
Spatial Structure

10 Dimensional Vector

Interpret as a sequence!
Spatial Structure

10 Dimensional Vector

Nearby elements are similar
Spatial Structure

10 Dimensional Vector

[Numbers: 1.5, 2.0, 2.2, 2.5, 3.1, 2.9, 2.4, 2.0, 1.3, 0.7]

Seasonality
Spatial Structure

10 Dimensional Vector

Effective degrees of freedom less than possible.
Stock Prices

1.5  2.0  2.2  2.5  3.1  2.9  2.4  2.0  1.3  0.7

1 element for each month, closing price

Predict stock split

“Time” series
Spatial Structure

9 Dimensional Vector

Laid out as a grid
Images

3 2D Grids Representing Pixels

Values 0-255
The bear ran fast

<table>
<thead>
<tr>
<th></th>
<th>Topic1</th>
<th>Topic2</th>
<th>Topic3</th>
</tr>
</thead>
<tbody>
<tr>
<td>The</td>
<td>0.15</td>
<td>0.55</td>
<td>0.3</td>
</tr>
<tr>
<td>bear</td>
<td>0.3</td>
<td>0.6</td>
<td>0.1</td>
</tr>
<tr>
<td>ran</td>
<td>0.4</td>
<td>0.5</td>
<td>0.1</td>
</tr>
<tr>
<td>fast</td>
<td>0.5</td>
<td>0.5</td>
<td>0.0</td>
</tr>
</tbody>
</table>
Recap: Linear Models

\[ y = A \cdot x + \epsilon \]

\[ A \cdot x = a_1 x_1 + a_2 x_2 + \ldots + a_d x_d \]

Positive \( a \) (means positive correlation), negative \( a \) (negative correlation)

Difficult to apply to such data due to dimensionality.
Reduce the size of the model

Convolution

Signal

<table>
<thead>
<tr>
<th></th>
<th>1.5</th>
<th>2.0</th>
<th>2.2</th>
<th>2.5</th>
<th>3.1</th>
<th>2.9</th>
<th>2.4</th>
<th>2.0</th>
<th>1.3</th>
<th>0.7</th>
</tr>
</thead>
</table>

Filter size=2

A1  A2

Output: N-size = 10-2 = 8 dimensional vector
Equivalent For Sequences?

Convolution

Signal

Filter size=4

Output: N-size = 10-4 = 6 dimensional vector
Signal:

<table>
<thead>
<tr>
<th></th>
<th>4</th>
<th>3</th>
<th>1</th>
<th>0</th>
<th>6</th>
</tr>
</thead>
</table>

Filter:

<table>
<thead>
<tr>
<th></th>
<th>-1</th>
<th>1</th>
</tr>
</thead>
</table>

Signal: -4 + 3
Filter: -3 + 1
Result: -1 + 0

Signal: 0 + 6
Filter: 1
Result: 0 + 6
Equivalent for Sequences?

Convolution

\[ A * x \]

Sliding dot product of A across the values of X
Interpretation?

\[ y = A \cdot x + \epsilon \]

\[ A \cdot x = a_1 x_1 + a_2 x_2 + \ldots + a_d x_d \]

Positive \( a \) (means positive correlation),
negative \( a \) (negative correlation)
Signal

4 3 1 0 6

Filter

0.5 0.5

0.5 0.5

0.5 0.5

0.5 0.5

0.5 0.5

3.5 2 0.5 3

“Moving average”
Moving Average

Smooths out the signal
<table>
<thead>
<tr>
<th>Signal</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Filter</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.33</td>
</tr>
<tr>
<td>0.33</td>
</tr>
<tr>
<td>0.33</td>
</tr>
<tr>
<td>0.33</td>
</tr>
<tr>
<td>0.33</td>
</tr>
<tr>
<td>0.33</td>
</tr>
<tr>
<td>0.33</td>
</tr>
</tbody>
</table>

“Moving average”
Moving Average

Smoothes out the signal
<table>
<thead>
<tr>
<th>Signal</th>
<th>Filter</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.5</td>
</tr>
<tr>
<td>1</td>
<td>-0.5</td>
</tr>
<tr>
<td>1</td>
<td>0.5</td>
</tr>
<tr>
<td>0</td>
<td>-0.5</td>
</tr>
<tr>
<td>0</td>
<td>0.5</td>
</tr>
<tr>
<td></td>
<td>-0.5</td>
</tr>
</tbody>
</table>

“Change Detector”
Every Signal Has A “Frequency” Domain

Time domain

Frequency domain
Every Signal Has A “Frequency” Domain

Convolutions filter certain frequencies

Time Domain

\[ x[n] \ast h[k] = y[n] \]

Frequency Domain

\[ X[k] \cdot H[k] = Y[k] \]

“Low Pass Filter”: rejects rapid spikes in the signal

“High Pass Filter”: identifies changes in the signal
Every Signal Has A “Frequency” Domain

“Low Pass Filter”: blurs the image

“High Pass Filter”: detects edges in the image
Anomaly Detection Example

Idea: Apply Convolution Before Classification

Signal: 1.5, 2.0, 2.2, 2.5, 3.1, 2.9, 2.4, 2.0, 1.3, 0.7

Filter: A1, A2, “filters the signal”

Size = 2

Output:

Run output through a classifier
Why Only One?

Signal: 1.5 2.0 2.2 2.5 3.1 2.9 2.4 2.0 1.3 0.7

Filter size=2

A1 A2 “local filters”

Output:

Filter size=4

A1 A2 A3 A4 “global filters”

Output:
Convolutional Neural Network

Cascade of filters that can be trained (fit) as one unit

“Layers” that consist of convolutions, classifications, regressions
“Training”: Provide images and correct values

“Backpropagation”
“Training”: Provide images and correct values

“Backpropagation”

loss 0.35
Idea: Apply Convolution Before Classification

Signal: 1.5 2.0 2.2 2.5 3.1 2.9 2.4 2.0 1.3 0.7

Filter size=2: A1 A2 “filters the signal”

Output: Run output through a classifier
How it works
Powerful and general strategy

Image classification
Powerful and general strategy

Downside? #1 Need lots of data

Image classification datasets
> 100k examples (upto millions)
Powerful and general strategy

Downside? #2 Need lots of compute

Specialized hardware to train (lots of GPUs)
Exactly why it took so long!

2008 first time CNNs actually “worked”

Why?

GPU programming from University of Toronto

Large image dataset from Stanford
Works!
Pre-trained models
Models Trained On Large Datasets Generalize

Filters generally apply well to new data!
Anomaly Detection Example

Idea: Apply Convolution Before Classification

Signal

Filter size=2

Output

Specific to your business

Run output through a classifier

“filters the signal”
Never Have To Implement By Hand

TensorFlow
Never Have To Implement By Hand

Step 1: Define Network Structure

Step 2: Define Training

Step 3: Predict

Data
How fast to update
Training and Testing

Original Data

Randomly Selected Rows

Training Data

Testing Data

Use this data for rule selection

Use this data for rule evaluation
Before Wed

https://www.tensorflow.org/tutorials/keras/classification

Go Through a Tutorial
Why I believe it Works

Identify low dimensional structure in the data