# Extraction of Time-Series Features with Convolutional AutoEncoders

Presented by: Gentry Atkinson

## Introduction:

- The data set was taken from  Mobile Sensor Data Anonymization, IoTDI '19.
- A convolutional autoencoder was trained on this data.
- The hidden layer was used to produce a feature set from the raw data.
- K-Means Clustering was applied to show that distinct groupings form in the data which correlate with the labels. The goal is to identify behavior based on data.

## Pivot from Proposal:

The data set was shifted from EEG to accelerometer and the problem was shifted from seizure recognition to behavioral classification.

## Data Set:

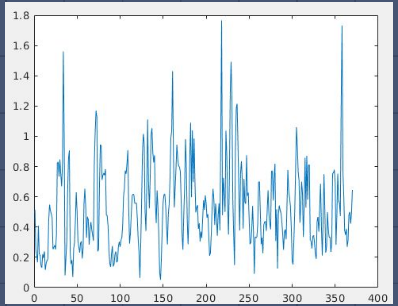
- Mobile Sensor Data Anonymization; M Malekzadeh, RG Clegg, A Cavallaro, H Haddadi; ACM/IEEE Internet of Things Design and Implementation [2019]
- 360 samples of 24 participants performing 6 different activities recorded with an iPhone accelerometer in the front hip pocket @50Hz sample rate.
- 12 features: attitude, gravity, rotation, and user acceleration on 3 axis.

# Data Processing:

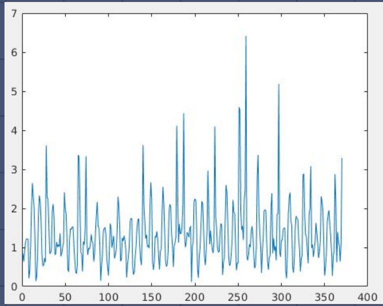
- 9 of the 12 features are derived and were discarded. Only User Acceleration was kept.
- The magnitude of the acceleration on 3 axis was calculated as: $\sqrt{x^2+y^2+z^2}$
- Segments were shortened to 370 values to make a fixed length input.
- Data was normalized before writing final values to CSV

# Data Visualizations:

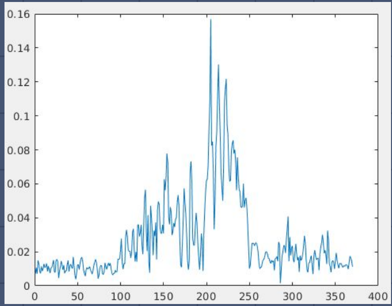
1. Traveling downstairs



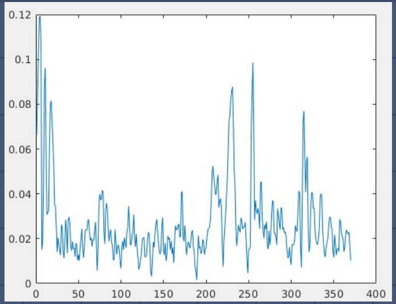

2. Jogging



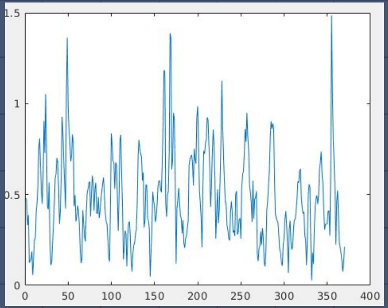

3. Sitting



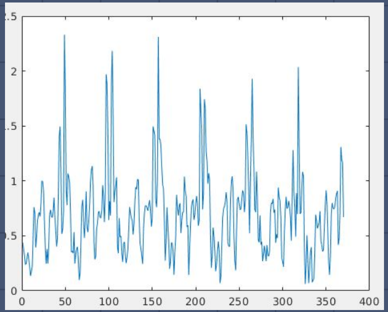

4. Standing




5. Traveling upstairs

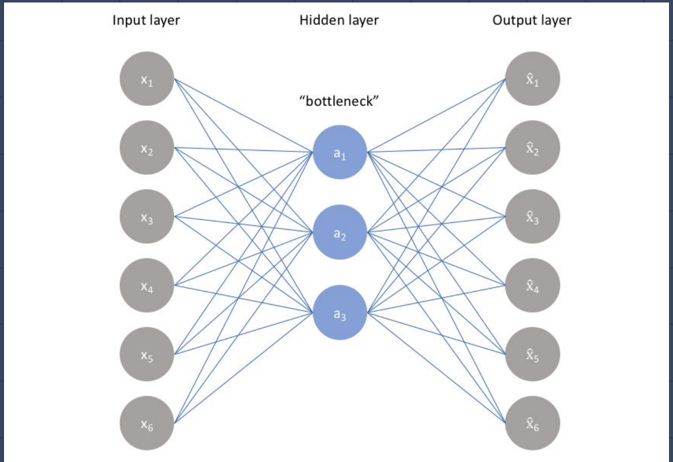



6. Walking



# Autoencoders:

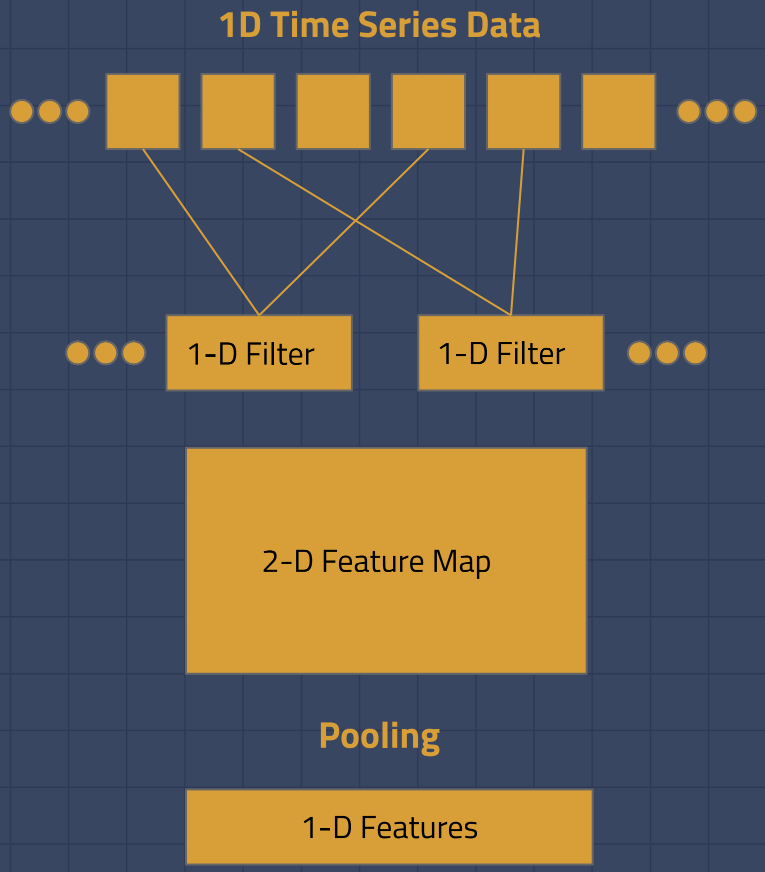
- An ANN is trained to reconstruct its inputs as its outputs.
- Information Theory tells us that the hidden layer must contain all of the information of the input and output layers.
- First used for dimensionality reduction.
- Unsupervised learner.




From: https://www.jeremyjordan.me/autoencoders/

# Convolutional Autoencoder:

- Incorporates both convolutional and dense layers.
- Employs one or more Transposed Convolutional layers in the decoder.
- Hidden layer encodes features learned by the filters.



# Previous Work:

- Stacked Convolutional Auto-Encoders for Hierarchical Feature Extraction; J Masci, U Meier, D Ciresan, J Schmidhuber; ICANN 2011
- Feature Extraction with Stacked Autoencoders for Epileptic Seizure Detection; A Supratak, L Li, Y Guo; 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society 2014

# Implementation:

- Built using Keras with a Tensorflow back-end.
- The encoder and decoder are convolutional while the hidden layer is fully connected.
- MATLAB is used to cluster and visualize the feature set.

```
print("Building input layer.................")
inp = Input(train_set[0].shape)

print("Building encoder.....................")
enc = Reshape((370, 1))(inp)
enc = BatchNormalization(scale=False, center=False)(enc)
enc = Conv1D(filters=339, kernel_size=32, strides=1,
    activation='elu', padding='same', use_bias=True)(enc)
enc = AveragePooling1D(pool_size=370, padding='same')(enc)
print(enc.shape)
enc = Flatten()(enc)
hidden = Dense(128, use_bias=True, activation='sigmoid')(enc)

print("Building decoder.....................")
dec = hidden
dec = Dense(339, activation='sigmoid')(dec)
print(dec.shape)
dec = Reshape((339, 1))(dec)
dec = Conv1DTranspose(dec, filters=370, kernel_size=32, padding='same')
dec = AveragePooling1D(pool_size=678, padding='same')(dec)
print(dec.shape)
dec = Flatten()(dec)

print("Compiling model......................")
autoenc = Model(inp, dec)
autoenc.compile(optimizer='RMSprop', loss='mean_absolute_error', metrics=[])
```

# Challenges:

- MaxPooling is not an invertible function.
- Keras does not offer a 1d Transposed Convolutional Layer yet.
- Some movement types are very similar.
- Many available loss, activation, optimizer functions without "good" choices being readily apparent.

# Results (Effect of Loss Function):

Mean Squared Error | Mean Absolute Error | Cosine Similarity

Clusters

Labels

## Count by label/ cluster

Mean Squared Error

| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | 10 | 8 | 0 | 0 | 39 | 1 |
| 2 | 1 | 20 | 0 | 0 | 16 | 1 |
| 3 | 6 | 9 | 16 | 5 | 2 | 1 |
| 4 | 6 | 8 | 9 | 1 | 6 | 8 |
| 5 | 2 | 25 | 0 | 0 | 29 | 2 |
| 6 | 6 | 19 | 3 | 0 | 19 | 10 |

Mean Absolute Error

| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | 0 | 48 | 0 | 0 | 9 | 1 |
| 2 | 0 | 33 | 0 | 0 | 5 | 0 |
| 3 | 15 | 2 | 7 | 7 | 7 | 1 |
| 4 | 3 | 8 | 10 | 7 | 2 | 8 |
| 5 | 0 | 26 | 4 | 0 | 17 | 11 |
| 6 | 2 | 24 | 2 | 0 | 17 | 12 |

Cosine Similarity

| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | 2 | 33 | 0 | 0 | 0 | 23 |
| 2 | 1 | 30 | 0 | 0 | 0 | 7 |
| 3 | 2 | 8 | 18 | 4 | 0 | 7 |
| 4 | 8 | 11 | 10 | 0 | 1 | 8 |
| 5 | 5 | 41 | 0 | 0 | 0 | 12 |
| 6 | 14 | 28 | 3 | 0 | 0 | 12 |

## % of label for each cluster

Mean Squared Error

| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | 0.1724 | 0.1379 | 0 | 0 | 0.6724 | 0.0172 |
| 2 | 0.0263 | 0.5263 | 0 | 0 | 0.4211 | 0.0263 |
| 3 | 0.1538 | 0.2308 | 0.4103 | 0.1282 | 0.0513 | 0.0256 |
| 4 | 0.1579 | 0.2105 | 0.2368 | 0.0263 | 0.1579 | 0.2105 |
| 5 | 0.0345 | 0.4310 | 0 | 0 | 0.5000 | 0.0345 |
| 6 | 0.1053 | 0.3333 | 0.0526 | 0 | 0.3333 | 0.1754 |

Mean Absolute Error

| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | 0 | 0.8276 | 0 | 0 | 0.1552 | 0.0172 |
| 2 | 0 | 0.8684 | 0 | 0 | 0.1316 | 0 |
| 3 | 0.3846 | 0.0513 | 0.1795 | 0.1795 | 0.1795 | 0.0256 |
| 4 | 0.0789 | 0.2105 | 0.2632 | 0.1842 | 0.0526 | 0.2105 |
| 5 | 0 | 0.4483 | 0.0690 | 0 | 0.2931 | 0.1897 |
| 6 | 0.0351 | 0.4211 | 0.0351 | 0 | 0.2982 | 0.2105 |

Cosine Similarity

| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | 0.0345 | 0.5690 | 0 | 0 | 0 | 0.3966 |
| 2 | 0.0263 | 0.7895 | 0 | 0 | 0 | 0.1842 |
| 3 | 0.0513 | 0.2051 | 0.4615 | 0.1026 | 0 | 0.1795 |
| 4 | 0.2105 | 0.2895 | 0.2632 | 0 | 0.0263 | 0.2105 |
| 5 | 0.0862 | 0.7069 | 0 | 0 | 0 | 0.2069 |
| 6 | 0.2456 | 0.4912 | 0.0526 | 0 | 0 | 0.2105 |

## % of cluster for each label

Mean Squared Error

| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | 0.3226 | 0.0899 | 0 | 0 | 0.3514 | 0.0435 |
| 2 | 0.0323 | 0.2247 | 0 | 0 | 0.1441 | 0.0435 |
| 3 | 0.1935 | 0.1011 | 0.5714 | 0.8333 | 0.0180 | 0.0435 |
| 4 | 0.1935 | 0.0899 | 0.3214 | 0.1667 | 0.0541 | 0.3478 |
| 5 | 0.0645 | 0.2809 | 0 | 0 | 0.2613 | 0.0870 |
| 6 | 0.1935 | 0.2135 | 0.1071 | 0 | 0.1712 | 0.4348 |

Mean Absolute Error

| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | 0 | 0.3404 | 0 | 0 | 0.1579 | 0.0303 |
| 2 | 0 | 0.2340 | 0 | 0 | 0.0877 | 0 |
| 3 | 0.7500 | 0.0142 | 0.3043 | 0.5000 | 0.1228 | 0.0303 |
| 4 | 0.1500 | 0.0567 | 0.4348 | 0.5000 | 0.0351 | 0.2424 |
| 5 | 0 | 0.1844 | 0.1739 | 0 | 0.2982 | 0.3333 |
| 6 | 0.1000 | 0.1702 | 0.0870 | 0 | 0.2982 | 0.3636 |

Cosine Similarity

| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | 0.0625 | 0.2185 | 0 | 0 | 0 | 0.3333 |
| 2 | 0.0313 | 0.1987 | 0 | 0 | 0 | 0.1014 |
| 3 | 0.0625 | 0.0530 | 0.5806 | 1 | 0 | 0.1014 |
| 4 | 0.2500 | 0.0728 | 0.3226 | 0 | 1 | 0.1159 |
| 5 | 0.1563 | 0.2715 | 0 | 0 | 0 | 0.1739 |
| 6 | 0.4375 | 0.1854 | 0.0968 | 0 | 0 | 0.1739 |

# Results (Addition of Bias):

Clusters

Labels

## Count by label/ cluster

Without Bias

| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | 0 | 36 | 0 | 21 | 1 | 0 |
| 2 | 0 | 14 | 0 | 22 | 2 | 0 |
| 3 | 16 | 4 | 6 | 8 | 5 | 0 |
| 4 | 10 | 12 | 1 | 9 | 5 | 1 |
| 5 | 0 | 28 | 0 | 27 | 3 | 0 |
| 6 | 4 | 22 | 0 | 21 | 10 | 0 |

With Bias

| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | 2 | 33 | 0 | 0 | 0 | 23 |
| 2 | 1 | 30 | 0 | 0 | 0 | 7 |
| 3 | 2 | 8 | 18 | 4 | 0 | 7 |
| 4 | 8 | 11 | 10 | 0 | 1 | 8 |
| 5 | 5 | 41 | 0 | 0 | 0 | 12 |
| 6 | 14 | 28 | 3 | 0 | 0 | 12 |

## % of label for each cluster

Without Bias

| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | 0 | 0.6207 | 0 | 0.3621 | 0.0172 | 0 |
| 2 | 0 | 0.3684 | 0 | 0.5789 | 0.0526 | 0 |
| 3 | 0.4103 | 0.1026 | 0.1538 | 0.2051 | 0.1282 | 0 |
| 4 | 0.2632 | 0.3158 | 0.0263 | 0.2368 | 0.1316 | 0.0263 |
| 5 | 0 | 0.4828 | 0 | 0.4655 | 0.0517 | 0 |
| 6 | 0.0702 | 0.3860 | 0 | 0.3684 | 0.1754 | 0 |

With Bias

| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | 0.0345 | 0.5690 | 0 | 0 | 0 | 0.3966 |
| 2 | 0.0263 | 0.7895 | 0 | 0 | 0 | 0.1842 |
| 3 | 0.0513 | 0.2051 | 0.4615 | 0.1026 | 0 | 0.1795 |
| 4 | 0.2105 | 0.2895 | 0.2632 | 0 | 0.0263 | 0.2105 |
| 5 | 0.0862 | 0.7069 | 0 | 0 | 0 | 0.2069 |
| 6 | 0.2456 | 0.4912 | 0.0526 | 0 | 0 | 0.2105 |

## % of cluster for each label

Without Bias

| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | 0 | 0.3103 | 0 | 0.1944 | 0.0385 | 0 |
| 2 | 0 | 0.1207 | 0 | 0.2037 | 0.0769 | 0 |
| 3 | 0.5333 | 0.0345 | 0.8571 | 0.0741 | 0.1923 | 0 |
| 4 | 0.3333 | 0.1034 | 0.1429 | 0.0833 | 0.1923 | 1 |
| 5 | 0 | 0.2414 | 0 | 0.2500 | 0.1154 | 0 |
| 6 | 0.1333 | 0.1897 | 0 | 0.1944 | 0.3846 | 0 |

With Bias

| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | 0.0625 | 0.2185 | 0 | 0 | 0 | 0.3333 |
| 2 | 0.0313 | 0.1987 | 0 | 0 | 0 | 0.1014 |
| 3 | 0.0625 | 0.0530 | 0.5806 | 1 | 0 | 0.1014 |
| 4 | 0.2500 | 0.0728 | 0.3226 | 0 | 1 | 0.1159 |
| 5 | 0.1563 | 0.2715 | 0 | 0 | 0 | 0.1739 |
| 6 | 0.4375 | 0.1854 | 0.0968 | 0 | 0 | 0.1739 |

# Results (Choice of Optimizer):

RMSProp | Adam | Stochastic Gradient Descent

Clusters

Labels

## Count by label/ cluster

**RMSProp**

| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | 2 | 33 | 0 | 0 | 0 | 23 |
| 2 | 1 | 30 | 0 | 0 | 0 | 7 |
| 3 | 2 | 8 | 18 | 4 | 0 | 7 |
| 4 | 8 | 11 | 10 | 0 | 1 | 8 |
| 5 | 5 | 41 | 0 | 0 | 0 | 12 |
| 6 | 14 | 28 | 3 | 0 | 0 | 12 |

**Adam**

| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | 0 | 5 | 31 | 0 | 22 | 0 |
| 2 | 0 | 0 | 27 | 1 | 10 | 0 |
| 3 | 13 | 0 | 8 | 1 | 6 | 11 |
| 4 | 9 | 2 | 13 | 7 | 6 | 1 |
| 5 | 0 | 7 | 35 | 2 | 14 | 0 |
| 6 | 4 | 15 | 14 | 10 | 14 | 0 |

**Stochastic Gradient Descent**

| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | 10 | 0 | 0 | 5 | 43 | 0 |
| 2 | 3 | 0 | 0 | 0 | 35 | 0 |
| 3 | 5 | 13 | 0 | 5 | 9 | 7 |
| 4 | 6 | 2 | 6 | 10 | 8 | 6 |
| 5 | 15 | 0 | 1 | 6 | 35 | 1 |
| 6 | 5 | 2 | 7 | 12 | 31 | 0 |

## % of label for each cluster

**RMSProp**

| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | 0.0345 | 0.5690 | 0 | 0 | 0 | 0.3966 |
| 2 | 0.0263 | 0.7895 | 0 | 0 | 0 | 0.1842 |
| 3 | 0.0513 | 0.2051 | 0.4615 | 0.1026 | 0 | 0.1795 |
| 4 | 0.2105 | 0.2895 | 0.2632 | 0 | 0.0263 | 0.2105 |
| 5 | 0.0862 | 0.7069 | 0 | 0 | 0 | 0.2069 |
| 6 | 0.2456 | 0.4912 | 0.0526 | 0 | 0 | 0.2105 |

**Adam**

| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | 0 | 0.0862 | 0.5345 | 0 | 0.3793 | 0 |
| 2 | 0 | 0 | 0.7105 | 0.0263 | 0.2632 | 0 |
| 3 | 0.3333 | 0 | 0.2051 | 0.0256 | 0.1538 | 0.2821 |
| 4 | 0.2368 | 0.0526 | 0.3421 | 0.1842 | 0.1579 | 0.0263 |
| 5 | 0 | 0.1207 | 0.6034 | 0.0345 | 0.2414 | 0 |
| 6 | 0.0702 | 0.2632 | 0.2456 | 0.1754 | 0.2456 | 0 |

**Stochastic Gradient Descent**

| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | 0.1724 | 0 | 0 | 0.0862 | 0.7414 | 0 |
| 2 | 0.0789 | 0 | 0 | 0 | 0.9211 | 0 |
| 3 | 0.1282 | 0.3333 | 0 | 0.1282 | 0.2308 | 0.1795 |
| 4 | 0.1579 | 0.0526 | 0.1579 | 0.2632 | 0.2105 | 0.1579 |
| 5 | 0.2586 | 0 | 0.0172 | 0.1034 | 0.6034 | 0.0172 |
| 6 | 0.0877 | 0.0351 | 0.1228 | 0.2105 | 0.5439 | 0 |

## % of cluster for each label

**RMSProp**

| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | 0.0625 | 0.2185 | 0 | 0 | 0 | 0.3333 |
| 2 | 0.0313 | 0.1987 | 0 | 0 | 0 | 0.1014 |
| 3 | 0.0625 | 0.0530 | 0.5806 | 1 | 0 | 0.1014 |
| 4 | 0.2500 | 0.0728 | 0.3226 | 0 | 1 | 0.1159 |
| 5 | 0.1563 | 0.2715 | 0 | 0 | 0 | 0.1739 |
| 6 | 0.4375 | 0.1854 | 0.0968 | 0 | 0 | 0.1739 |

**Adam**

| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | 0 | 0.1724 | 0.2422 | 0 | 0.3056 | 0 |
| 2 | 0 | 0 | 0.2109 | 0.0476 | 0.1389 | 0 |
| 3 | 0.5000 | 0 | 0.0625 | 0.0476 | 0.0833 | 0.9167 |
| 4 | 0.3462 | 0.0690 | 0.1016 | 0.3333 | 0.0833 | 0.0833 |
| 5 | 0 | 0.2414 | 0.2734 | 0.0952 | 0.1944 | 0 |
| 6 | 0.1538 | 0.5172 | 0.1094 | 0.4762 | 0.1944 | 0 |

**Stochastic Gradient Descent**

| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | 0.2273 | 0 | 0 | 0.1316 | 0.2671 | 0 |
| 2 | 0.0682 | 0 | 0 | 0 | 0.2174 | 0 |
| 3 | 0.1136 | 0.7647 | 0 | 0.1316 | 0.0559 | 0.5000 |
| 4 | 0.1364 | 0.1176 | 0.4286 | 0.2632 | 0.0497 | 0.4286 |
| 5 | 0.3409 | 0 | 0.0714 | 0.1579 | 0.2174 | 0.0714 |
| 6 | 0.1136 | 0.1176 | 0.5000 | 0.3158 | 0.1925 | 0 |

# Results (Activation Function):

Clusters

Labels

## Count by label/cluster

### Linear

| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | 2 | 33 | 0 | 0 | 0 | 23 |
| 2 | 1 | 30 | 0 | 0 | 0 | 7 |
| 3 | 2 | 8 | 18 | 4 | 0 | 7 |
| 4 | 8 | 11 | 10 | 0 | 1 | 8 |
| 5 | 5 | 41 | 0 | 0 | 0 | 12 |
| 6 | 14 | 28 | 3 | 0 | 0 | 12 |

### Relu

| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | 0 | 3 | 55 | 0 | 0 | 0 |
| 2 | 0 | 3 | 35 | 0 | 0 | 0 |
| 3 | 3 | 3 | 9 | 18 | 2 | 4 |
| 4 | 0 | 10 | 16 | 9 | 1 | 2 |
| 5 | 0 | 9 | 49 | 0 | 0 | 0 |
| 6 | 0 | 13 | 40 | 4 | 0 | 0 |

### Elu

| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | 10 | 0 | 0 | 1 | 0 | 47 |
| 2 | 4 | 0 | 1 | 0 | 0 | 33 |
| 3 | 4 | 6 | 1 | 3 | 17 | 8 |
| 4 | 4 | 1 | 7 | 4 | 9 | 13 |
| 5 | 6 | 0 | 2 | 0 | 0 | 50 |
| 6 | 11 | 0 | 13 | 1 | 4 | 28 |

## % of label for each cluster

### Linear

| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | 0.0345 | 0.5690 | 0 | 0 | 0 | 0.3966 |
| 2 | 0.0263 | 0.7895 | 0 | 0 | 0 | 0.1842 |
| 3 | 0.0513 | 0.2051 | 0.4615 | 0.1026 | 0 | 0.1795 |
| 4 | 0.2105 | 0.2895 | 0.2632 | 0 | 0.0263 | 0.2105 |
| 5 | 0.0862 | 0.7069 | 0 | 0 | 0 | 0.2069 |
| 6 | 0.2456 | 0.4912 | 0.0526 | 0 | 0 | 0.2105 |

### Relu

| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | 0 | 0.0517 | 0.9483 | 0 | 0 | 0 |
| 2 | 0 | 0.0789 | 0.9211 | 0 | 0 | 0 |
| 3 | 0.0769 | 0.0769 | 0.2308 | 0.4615 | 0.0513 | 0.1026 |
| 4 | 0 | 0.2632 | 0.4211 | 0.2368 | 0.0263 | 0.0526 |
| 5 | 0 | 0.1552 | 0.8448 | 0 | 0 | 0 |
| 6 | 0 | 0.2281 | 0.7018 | 0.0702 | 0 | 0 |

### Elu

| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | 0.1724 | 0 | 0 | 0.0172 | 0 | 0.8103 |
| 2 | 0.1053 | 0 | 0.0263 | 0 | 0 | 0.8684 |
| 3 | 0.1026 | 0.1538 | 0.0256 | 0.0769 | 0.4359 | 0.2051 |
| 4 | 0.1053 | 0.0263 | 0.1842 | 0.1053 | 0.2368 | 0.3421 |
| 5 | 0.1034 | 0 | 0.0345 | 0 | 0 | 0.8621 |
| 6 | 0.1930 | 0 | 0.2281 | 0.0175 | 0.0702 | 0.4912 |

## % of cluster for each label

### Linear

| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | 0.0625 | 0.2185 | 0 | 0 | 0 | 0.3333 |
| 2 | 0.0313 | 0.1987 | 0 | 0 | 0 | 0.1014 |
| 3 | 0.0625 | 0.0530 | 0.5806 | 1 | 0 | 0.1014 |
| 4 | 0.2500 | 0.0728 | 0.3226 | 0 | 1 | 0.1159 |
| 5 | 0.1563 | 0.2715 | 0 | 0 | 0 | 0.1739 |
| 6 | 0.4375 | 0.1854 | 0.0968 | 0 | 0 | 0.1739 |

### Relu

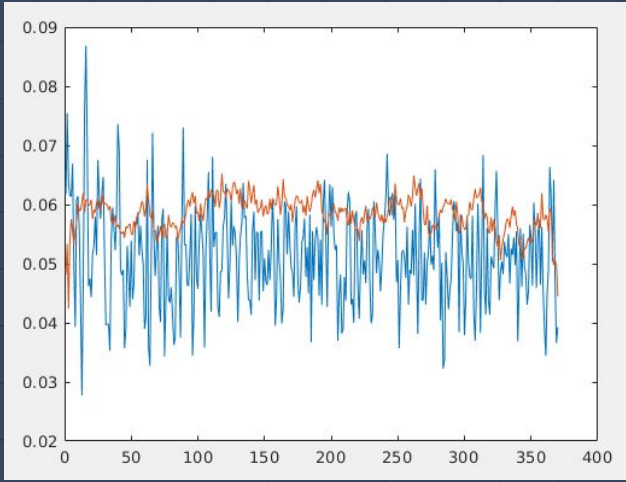
| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | 0 | 0.0732 | 0.2696 | 0 | 0 | 0 |
| 2 | 0 | 0.0732 | 0.1716 | 0 | 0 | 0 |
| 3 | 1 | 0.0732 | 0.0441 | 0.5806 | 0.6667 | 0.6667 |
| 4 | 0 | 0.2439 | 0.0784 | 0.2903 | 0.3333 | 0.3333 |
| 5 | 0 | 0.2195 | 0.2402 | 0 | 0 | 0 |
| 6 | 0 | 0.3171 | 0.1961 | 0.1290 | 0 | 0 |

### Elu

| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | 0.2564 | 0 | 0 | 0.1111 | 0 | 0.2626 |
| 2 | 0.1026 | 0 | 0.0417 | 0 | 0 | 0.1844 |
| 3 | 0.1026 | 0.8571 | 0.0417 | 0.3333 | 0.5667 | 0.0447 |
| 4 | 0.1026 | 0.1429 | 0.2917 | 0.4444 | 0.3000 | 0.0726 |
| 5 | 0.1538 | 0 | 0.0833 | 0 | 0 | 0.2793 |
| 6 | 0.2821 | 0 | 0.5417 | 0.1111 | 0.1333 | 0.1564 |

# Discussion:

- There's still some question of how tightly the autoencoder is fitting the raw data.
- The sample window may be too wide for the autoencoder to reliably train back to.
- What we've seen is the output of a single autoencoder.




Decoded data (red) overlayed on the original input(blue).

# Future Work:

- Adding convolutional layers to the autoencoder or stacking autoencoders.
- A denoising autoencoder is being considered for stronger real world usefulness.
- Work on collected data rather than public.
- Work on running data rather than segmented.
- Classify rather than cluster.

## Classification:

- Strong clusters suggest clear decision boundaries.
- 1d convolutional autoencoders do a good job of maintaining temporal locality in features.
- Step 1: feed features extracted from windows into an SVM or similar classifier.
- Step 2: feed sequences of features extracted in real time into an LSTM or similar classifier.

# Questions or Comments?

Code and data available at:
https://github.com/gentry-atkinson/accel_behavior