MyShake
Building a global smartphone seismic network
http://seismo.berkeley.edu/qingkaikong/
Qingkai Kong
MyShake:

1) When an earthquake occurs, smartphones near the epicenter that are running MyShake detect the shaking.

2) Data are sent to the cloud, combined with information from other smartphones, and the system confirms that an earthquake is underway.

3) An alert could be sent out to phones farther from the epicenter, providing seconds to minutes of warning before the shaking arrives.
MyShake methodology
Is this an Earthquake?
To remove the base-line offset, the high-pass filter $H1$ is applied to compute acceleration as follows:

$$a_j = \frac{1 + q}{2} \frac{\xi_j - \xi_{j-1}}{g_{lg}} + qa_{j-1}$$  \hspace{1cm} (13)

The velocity is obtained by integration and high-pass filtering with $H1$:

$$v_j = \frac{1 + q}{2} \frac{a_j + a_{j-1}}{2} \Delta t + qv_{j-1}.$$  \hspace{1cm} (17)

The displacement, $d_j$, is computed from velocity, $v_j$, by integration and high-pass filtering with $H1$. The high-pass filter is applied to avoid long-term drift of the baseline. Applying $H1$ again, we obtain

$$d_j = \frac{1 + q}{2} \frac{v_j + v_{j-1}}{2} \Delta t + qd_{j-1}.$$  \hspace{1cm} (12)
Recognize Earthquakes

Data Collection
- Acceleration

Windowing
- Sliding window

Feature
- Feature 1
- Feature 2
- Feature 3

Classification
- ANN
DATA
Human activity
Simulated data

Step 1: Get past EQ
Step 2: Downgrade
Step 3: Add noise

Raw data

With phone noise
Shake table data

Record from Loma Prieta earthquake
Data examples

Optimization algorithm

Tunable Model

Trained Model

MAKE THINGS HAPPEN!
MyShake workflow

Steday
Prefilter
STA/LTA
ANN
MyShake workflow

Steday  Prefilter  STA/LTA  ANN
Crash course on ANN
ANN in simple view
YOU'RE IN MY SPOT
Intuitive Artificial Neural Network

Input

\[ F(\text{eye} \times w_1 + \text{nose} \times w_2 + \ldots + \text{mouth} \times w_n) \]

Output
Intuitive
Artificial Neural Network

\[ F(\text{eye} \times w_1 + \text{nose} \times w_2 + \ldots + \text{mouth} \times w_n) \]

Input

\[ w_1 \]
\[ w_2 \]
\[ \ldots \]
\[ w_n \]

Output

feedback

\[ \text{error} \]
Intuitive Artificial Neural Network

\[ F(\text{eye} \times w_1 + \text{nose} \times w_2 + \ldots + \text{mouth} \times w_n) \]

Input

Output

\[ w_1 \]

\[ w_2 \]

\[ w_n \]

error feedback
Intuitive Artificial Neural Network

Output

Input

\[ F(\text{eye} \times w_1 + \text{nose} \times w_2 + \ldots + \text{mouth} \times w_n) \]
Schema of ANN

STA/LTA trigger

2 sec window with 1 sec overlap

Timeline
Features Extracted

We tested 50 different features
Feature selection

- **Forward greedy selection**
  - Step 1: Start with a null model
  - Step 2: Add one variable at a time, and record the improvement (accuracy or AIC).
  - Step 3: Select the feature with the max improvement on the model
  - Step 4: Repeat step 2 and 3 for all the features
3 features

Number of features used

Training score based on forward selection

Feature selection
Calculating features

- **ZC**
  - Get the largest zero crossing on the 3 components

- **IQR**
  - \( \text{IQR} = Q_3 - Q_1 \)

- **CAV**
  - \( CAV = \int_0^2 |a(t)| \, dt \)
ANN structure

Input layer

Hidden layer

Output layer

IQR

ZC

CAV

Σ | f

Σ | f

Σ | f

Σ | f

Σ | f

Σ | f

Σ | f

EQ?
Machine Learning

what society thinks I do
what my friends think I do
what my parents think I do
what other programmers think I do
what I think I do
what I really do

\[ L = \|w\|^2 - \sum_{i} a_i y_i (w \cdot x_i + b) + \sum_i a_i \]

\[ w = \sum_i a_i x_i, \sum_i a_i y_i = 0 \]

\[ \nabla y_i(\theta_0) = \frac{1}{n} \sum_{i=1}^{n} \nabla \ell(x_i, y_i; \theta_0) \]

\[ \theta_{k+1} = \theta_k - \eta_k \nabla \ell(x_i(t); y_i(t); \theta_t) - \eta_k \nabla r(\theta_t) \]

\[ E_{\theta(t)}[\ell(x_i(t), y_i(t); \theta_t)] = \frac{1}{n} \sum_{i} \ell(x_i, y_i; \theta_t) \]

>>> from sklearn import svm
Input $X$  

Output $y$

$\Sigma = \omega_0 x_0 + \omega_1 x_1 + \omega_2 x_2 + \omega_3 x_3 + \ldots + \omega_n x_n$

$f = f(\omega_0 x_0 + \omega_1 x_1 + \omega_2 x_2 + \omega_3 x_3 + \ldots + \omega_n x_n)$

$X$ – input data  
y – output target  
$\omega_i$ – weights  
$\Sigma$ – summation  
f – activation function  
Blue circle – bias
More activation function

\[ z = \omega_0 x_0 + \omega_1 x_1 + \omega_2 x_2 + \omega_3 x_3 + \ldots + \omega_n x_n \]

\[ f(z) = \frac{1}{1 + e^{-z}} \]

\[ \frac{df(z)}{dx} = f(z)(1 - f(z)) \]
X – input data
y – output target
Σ – summation
f – activation function
blue circle - bias
Input

\[ x \]

\begin{align*}
\text{feature1} \\
\text{feature2} \\
\text{feature3}
\end{align*}

\[ \sum f \]

Hidden1

\[ \sum f \]

Hidden2

\[ \sum f \]

Hidden3

\[ \sum f \]

Hidden4

\[ \sum f \]

Output

\[ y \]
Backpropagation training algorithm

- MLP is trained by **backpropagation**.
- **forward pass**
  - present a training sample to the neural network
  - calculate the error (MSE) in each output neuron
- **backward pass**
  - first calculate gradient for hidden-to-output weights
  - then calculate gradient for input-to-hidden weights
    - the knowledge of $\text{grad}_{\text{hidden-output}}$ is necessary to calculate $\text{grad}_{\text{input-hidden}}$
  - update the weights in the network

$$ w_{m+1} = w_m + \Delta w_m \quad \Delta w_m = -\beta d_m $$
\[ \frac{\partial}{\partial W_{i,j}^{(1)}} J(W) = a_j^{(1)} \delta_{i}^{(l+1)} \]  
(compute gradient)

(error term of the output layer)
\[ \delta^{(3)} = a^{(3)} - y \]

\[ \delta^{(2)} = \left(W^{(2)}\right)^T \delta^{(3)} \cdot \frac{\partial g(z^{(2)})}{\partial z^{(2)}} \]  
(error term of the hidden layer)

Input \( x \)

output \( \hat{y} \) → target \( y \)
First layer calculation

\[
\begin{bmatrix} x_1, x_2, x_3 \end{bmatrix} \cdot \begin{bmatrix}
  w_{11}^{(1)}, & w_{12}^{(1)}, & w_{13}^{(1)}, & w_{14}^{(1)}, & w_{15}^{(1)} \\
  w_{21}^{(1)}, & w_{22}^{(1)}, & w_{23}^{(1)}, & w_{24}^{(1)}, & w_{25}^{(1)} \\
  w_{31}^{(1)}, & w_{32}^{(1)}, & w_{33}^{(1)}, & w_{34}^{(1)}, & w_{35}^{(1)}
\end{bmatrix} + \begin{bmatrix} c_1^{(1)}, & c_2^{(1)}, & c_3^{(1)}, & c_4^{(1)}, & c_5^{(1)} \end{bmatrix}
\]

\[
= \begin{bmatrix} a_1, a_2, a_3, a_4, a_5 \end{bmatrix}
\]

Activation function

\[
\frac{1}{1 + e^{-x}}
\]

\[
\begin{bmatrix} A_1, A_2, A_3, A_4, A_5 \end{bmatrix}
\]

EQ?
Second layer calculation

\[
\begin{bmatrix}
A_1, A_2, A_3, A_4, A_5
\end{bmatrix} \cdot \begin{bmatrix}
w_{11}^{(2)} \\
w_{21}^{(2)} \\
w_{31}^{(2)} \\
w_{41}^{(2)} \\
w_{51}^{(2)}
\end{bmatrix} + c_1^{(2)} = B
\]

Activation function

EQ?
Details of the training
Which one is better?
Clearly, black is better.
Train/test dataset split
Train/test dataset split

20% Test  80% Train
Model Complexity vs. Mean Error

- Underfitting
- Overfitting
- Just right
5-fold cross-validation
## Results of 10 fold CV

<table>
<thead>
<tr>
<th>Number of neuron</th>
<th>Length of time window</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>20</td>
<td>2.1</td>
</tr>
<tr>
<td>19</td>
<td>2.06</td>
</tr>
<tr>
<td>18</td>
<td>2.05</td>
</tr>
<tr>
<td>17</td>
<td>2.15</td>
</tr>
<tr>
<td>16</td>
<td>2.26</td>
</tr>
<tr>
<td>15</td>
<td>2.5</td>
</tr>
<tr>
<td>14</td>
<td>2.54</td>
</tr>
<tr>
<td>13</td>
<td>2.38</td>
</tr>
<tr>
<td>12</td>
<td>2.17</td>
</tr>
<tr>
<td>11</td>
<td>2.38</td>
</tr>
<tr>
<td>10</td>
<td>2.21</td>
</tr>
</tbody>
</table>
Imbalanced data

9000 vs 1000

\[
\frac{9000}{1000 + 9000} = 90\%
\]
Address imbalance

• Collecting more data
• Resample the dataset
  o Oversample
  o Downsample
• Generate synthetic samples
• Try algorithms insensitive to imbalanced data
• Penalized Models
• Try a different perspective
  o Anomaly detection
Training data

![Graph showing training data with features 1 (frequency) and 2 (amplitude). The graph includes data points for human, shake table, and simulated categories.](image)
Clustered human data

![Image showing a scatter plot with two features: Feature 1 (Frequency) on the x-axis and Feature 2 (Amplitude) on the y-axis. The plot includes two sets of data points, one labeled 'human' and the other 'clustered human.' The data points are differentiated by color, with blue for 'human' and red for 'clustered human.' The plot visually represents the clustered human data distribution.]
Final dataset

- Feature 1 (Frequency)
- Feature 2 (Amplitude)
Classification

93.2% of the human triggers are correctly classified

Higher frequency

Higher amplitude

IQR

ZC

Earthquakes

Non-Earthquakes
Type I Error
(False Positive)

You’re pregnant

Type II Error
(False Negative)

You’re not pregnant
### Test performance

<table>
<thead>
<tr>
<th>Earthquake</th>
<th>Within 10 km</th>
<th>Within 20 km</th>
<th>Within 30 km</th>
</tr>
</thead>
<tbody>
<tr>
<td>1989 Loma Prieta M7</td>
<td>2/2</td>
<td>4/4</td>
<td>11/11</td>
</tr>
<tr>
<td>1994 Northridge M6.7</td>
<td>4/4</td>
<td>19/19</td>
<td>29/29</td>
</tr>
<tr>
<td>2004 Parkfield M6</td>
<td>19/20</td>
<td>35/39</td>
<td>36/42</td>
</tr>
<tr>
<td>2014 La Habra M5.1</td>
<td>13/13</td>
<td>22/52</td>
<td>30/120</td>
</tr>
<tr>
<td>2014 Napa M6</td>
<td>2/2</td>
<td>6/8</td>
<td>10/24</td>
</tr>
<tr>
<td>Human triggers</td>
<td>Detect as noise</td>
<td>Detect as EQs</td>
<td>Total trigger</td>
</tr>
<tr>
<td>20150201-20150228</td>
<td>3562</td>
<td>261</td>
<td>3823</td>
</tr>
</tbody>
</table>

93.2% of the human triggers are correctly classified
A network of MyShakers
In terms of detect EQ

STA/LTA trigger

2 sec window with 1 sec overlap
In terms of detect EQ

STA/LTA trigger
Crash course on CNN

What the computer sees

82% cat
15% dog
2% hat
1% mug

image classification
Crash course on CNN
Crash course on CNN

Image

Convolved Feature
My final model

Test score: 99.84%
Transfer Learning