Deutsche Telekom, Inc.
Silicon Valley Innovation Center

University of California Berkeley Seismological Laboratory

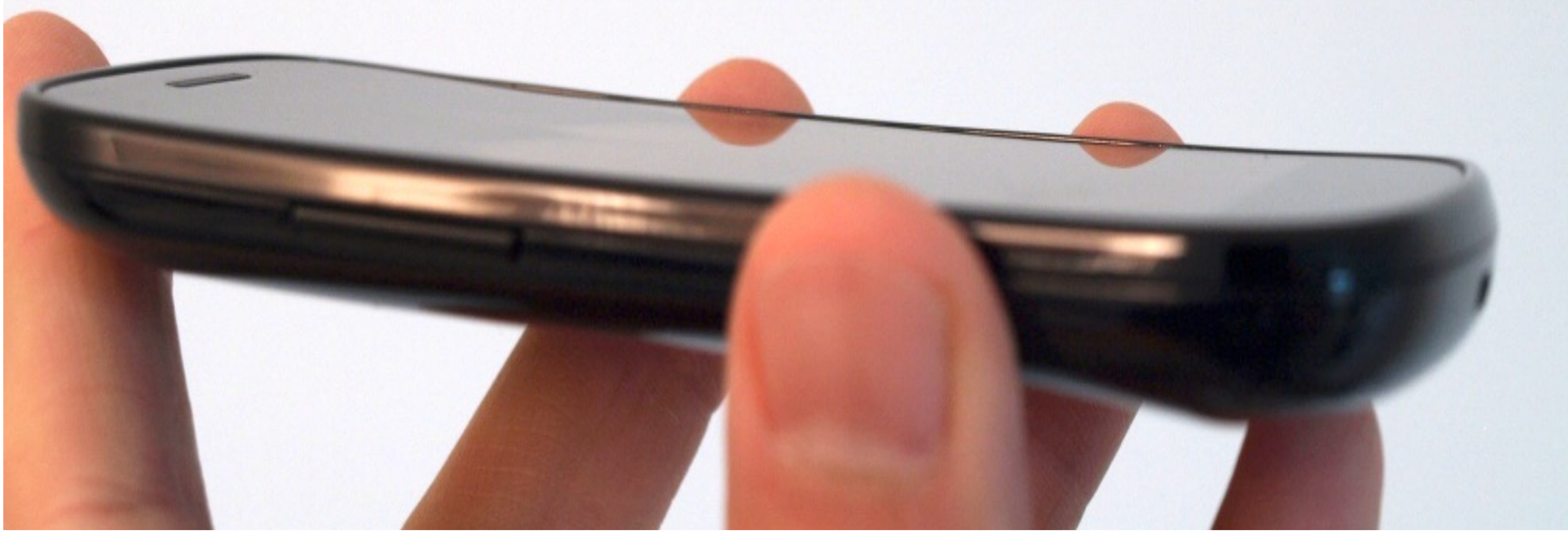
GORDON AND BETTY MOORE FOUNDATION

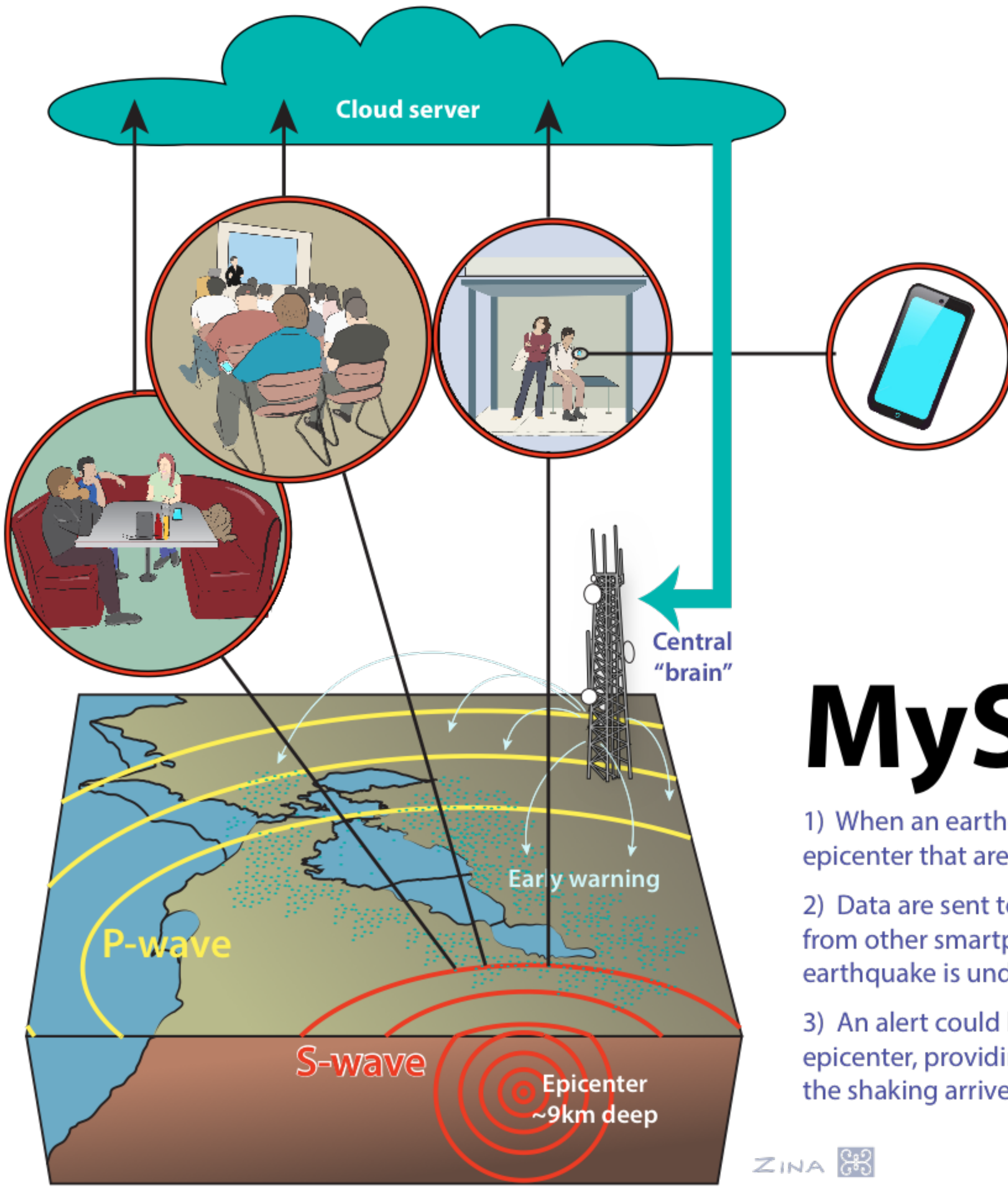
MyShake

http://seismo.berkeley.edu/qingkaikong/

Qingkai Kong

Building a global smartphone seismic network





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:

$$a_j = \frac{1+q}{2} \frac{\xi_j - \xi_{j-1}}{g_{\mathrm{lg}}} + q a_{j-1} \qquad (13)$$

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

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

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

$$d_j = \frac{1+q}{2} \frac{v_j + v_{j-1}}{2} \Delta t + q d_{j-1} . \qquad (12)$$

Kanamori et al 1999

Recognize Earthquakes

Data Collection

- Acceleration

Windowing

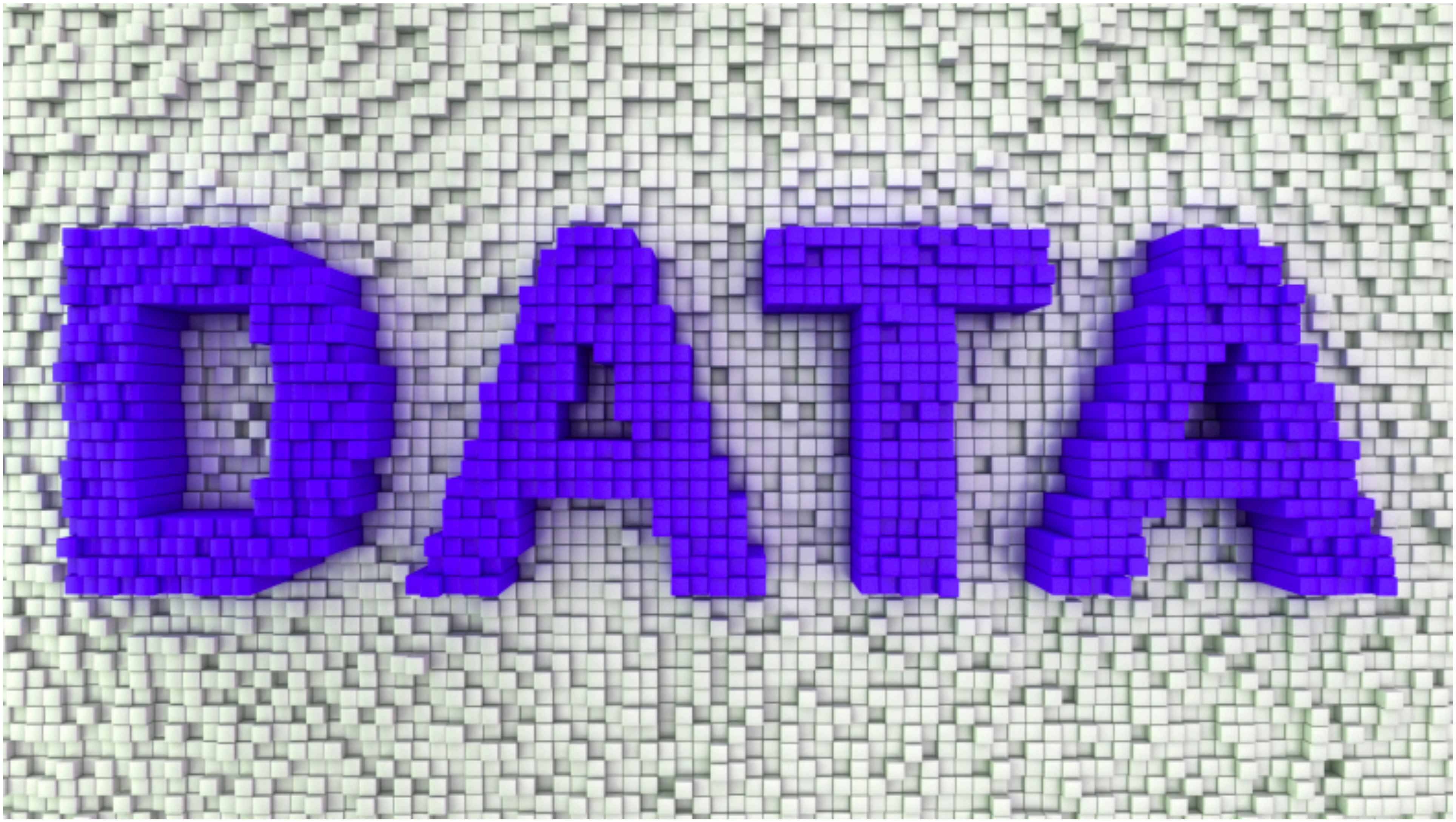
- Sliding window

Feature

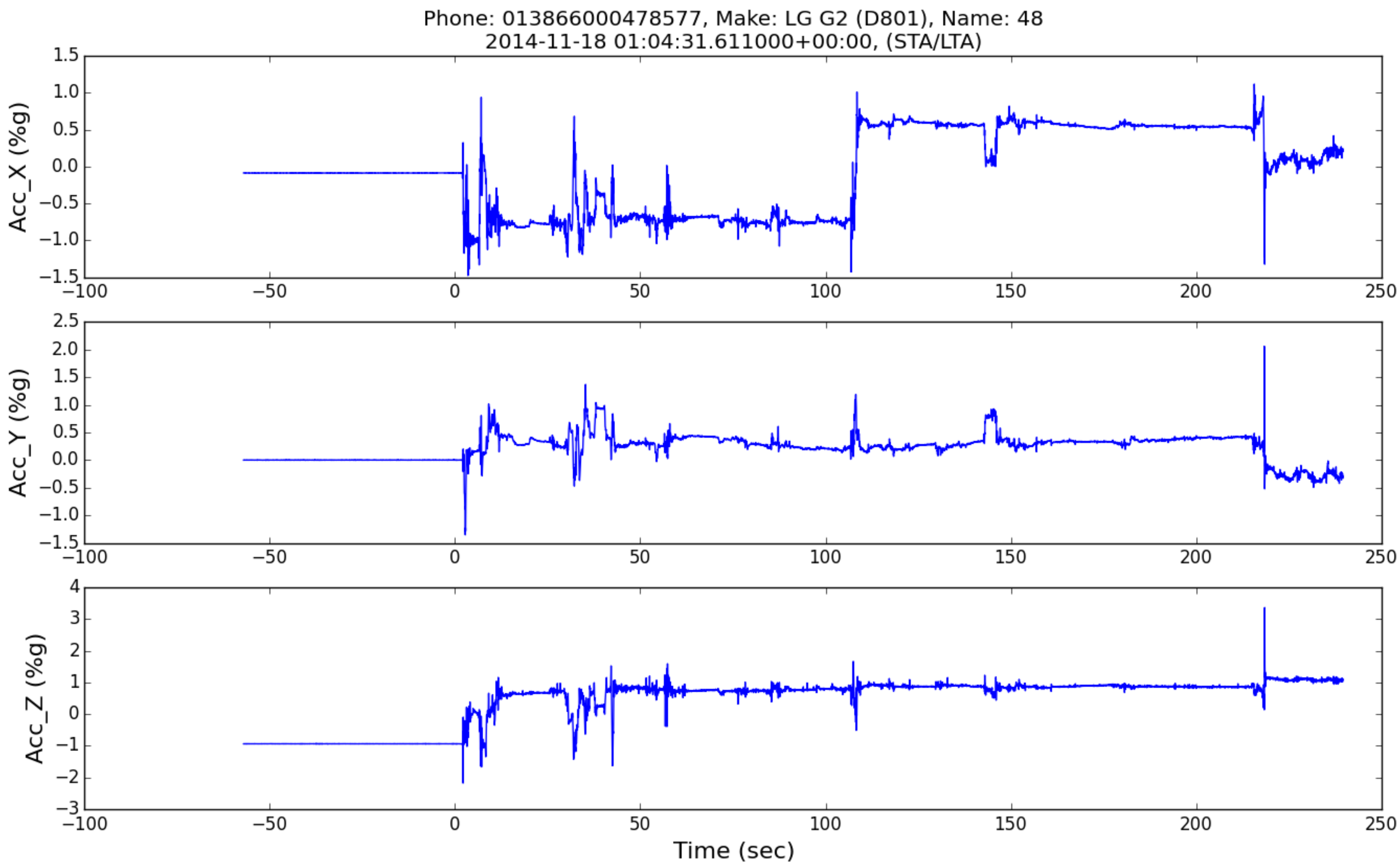
- Feature 1
- Feature 2
- Feature 3

Classification

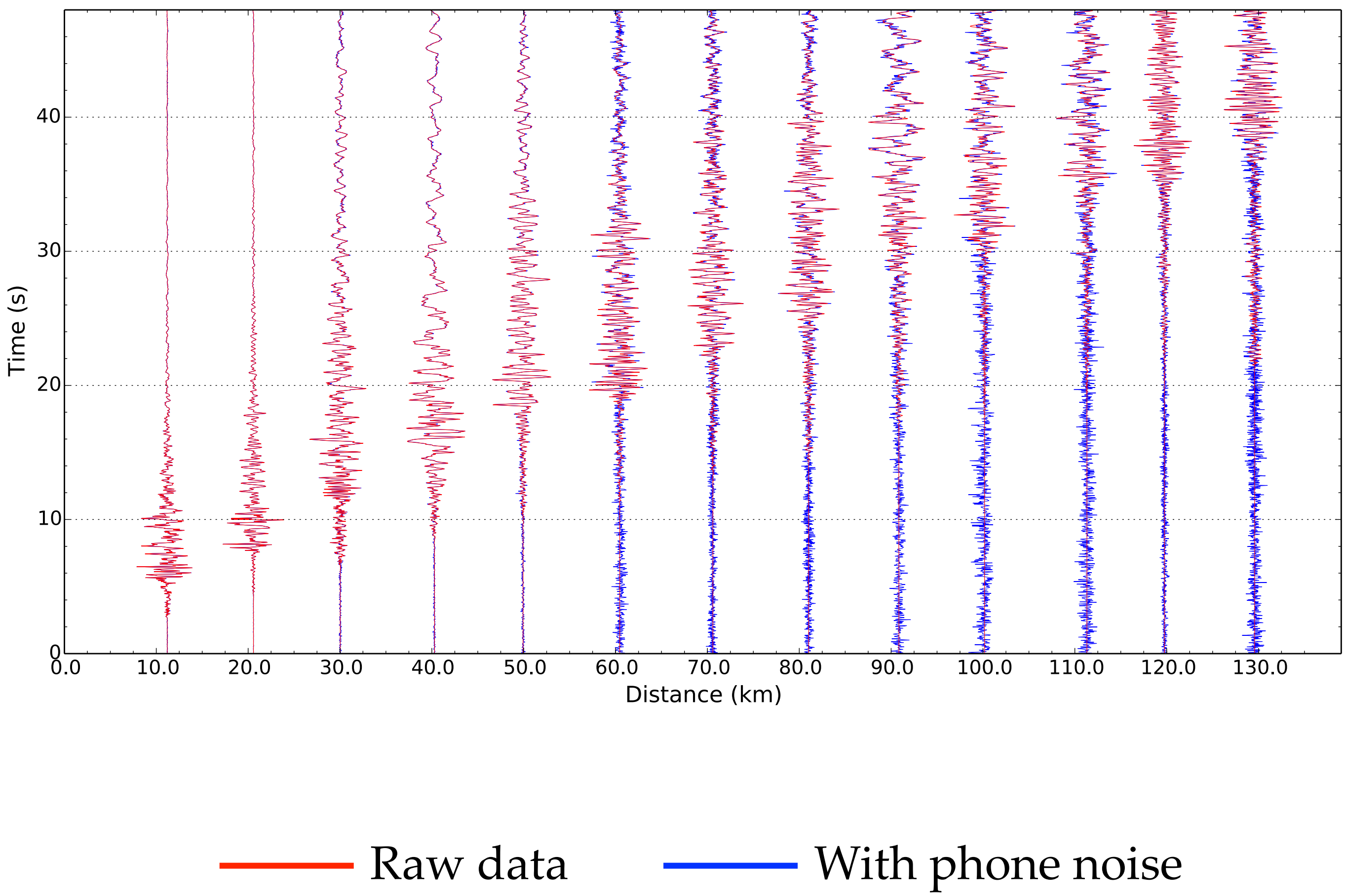
- ANN


DATA

Human activity



Simulated data

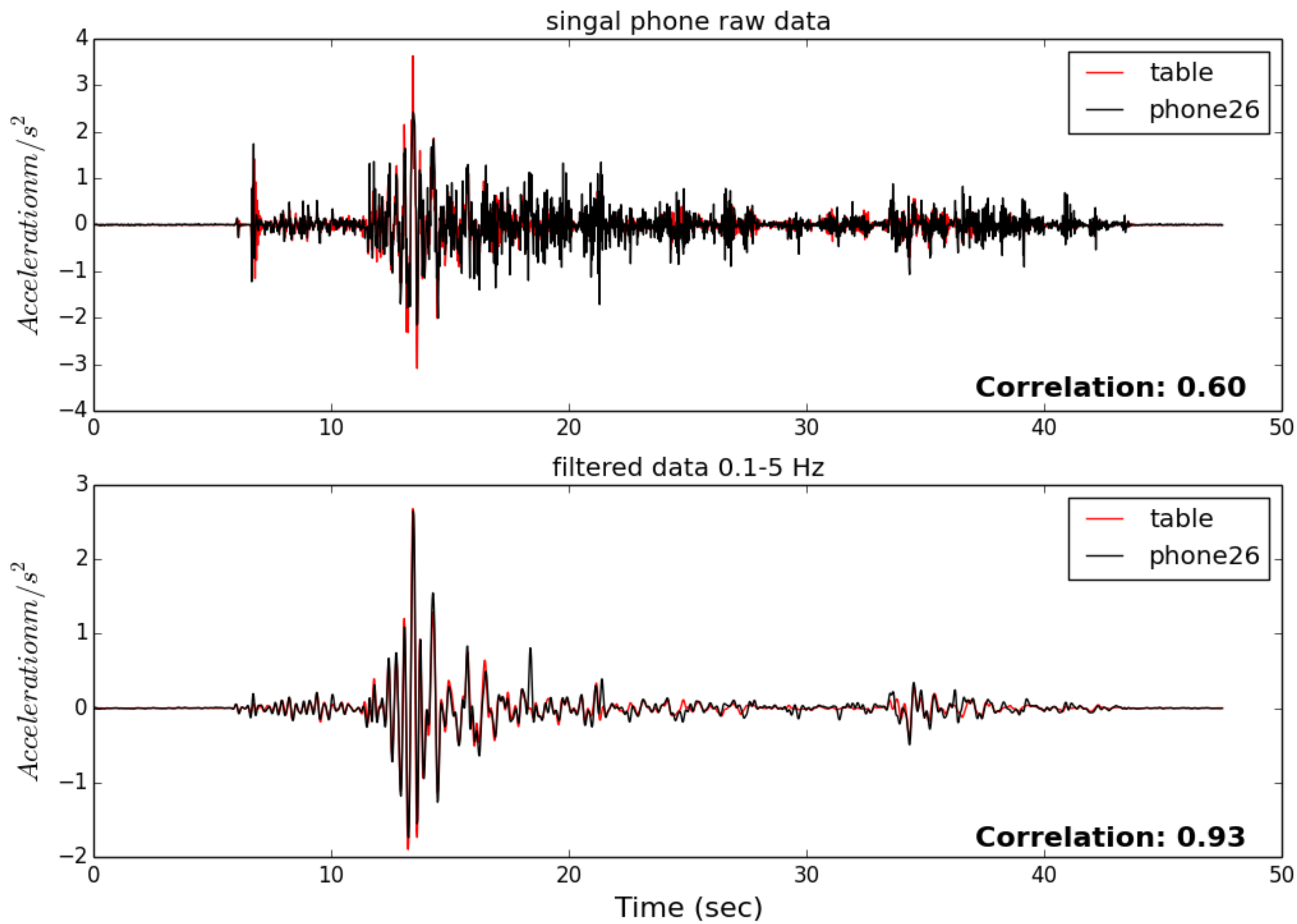


Step 1: Get past EQ

Step 2: Downgrade

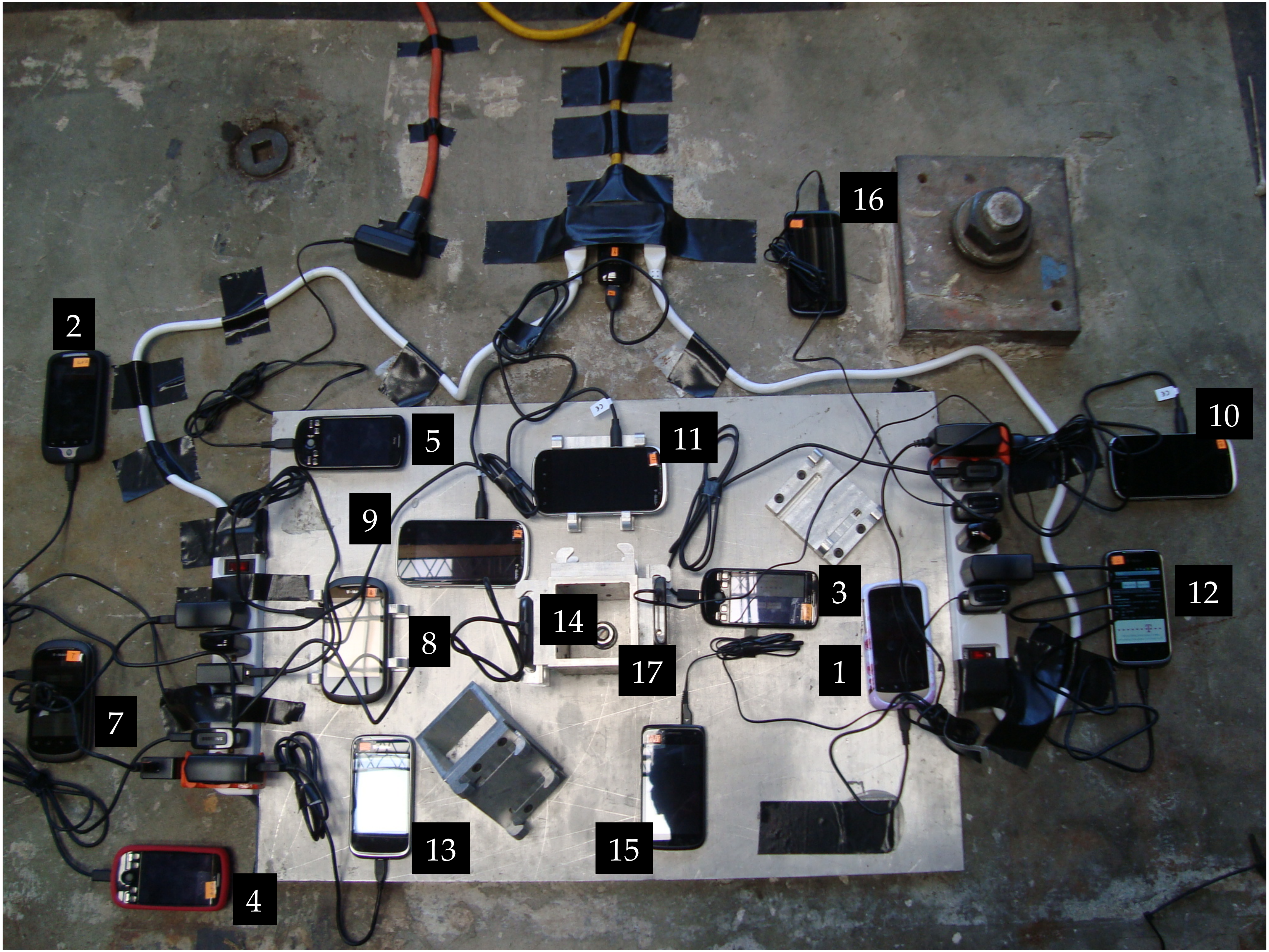
Step 3: Add noise

Shake table data

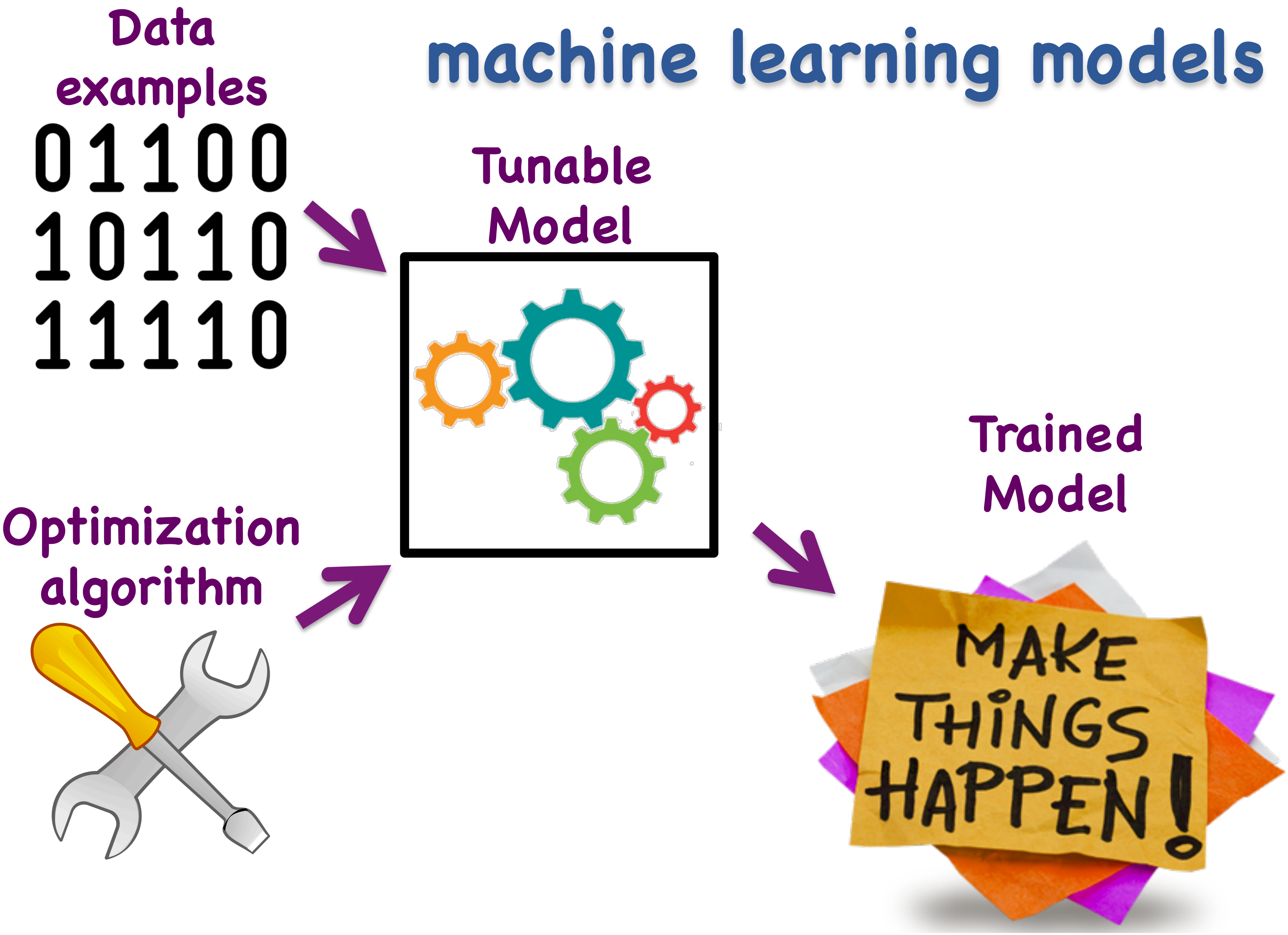


Record from Loma Prieta earthquake


PEER BERKELEY
PEER
PEER

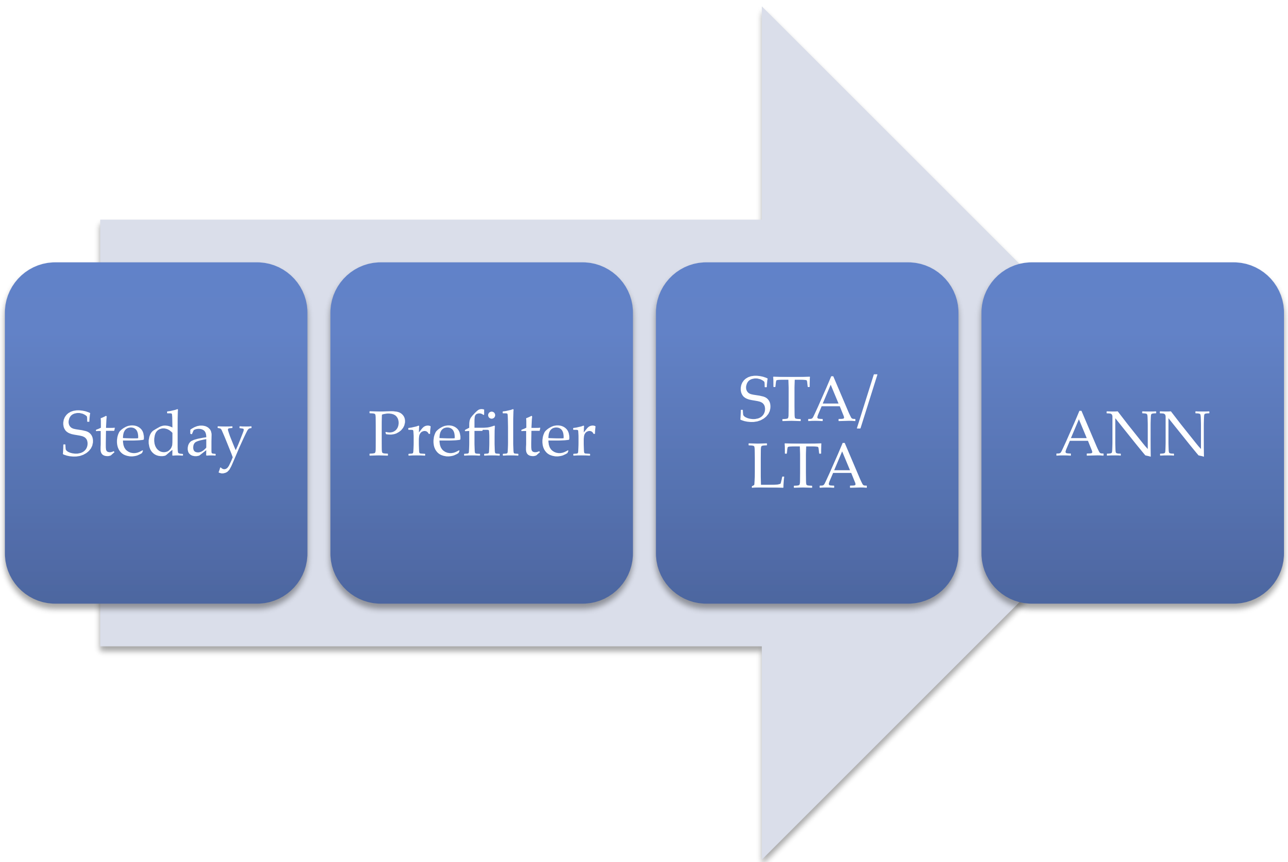

16
2
10
5
11
9
3
12
14
8
17
1
7
13
15
4


PEER
PEER

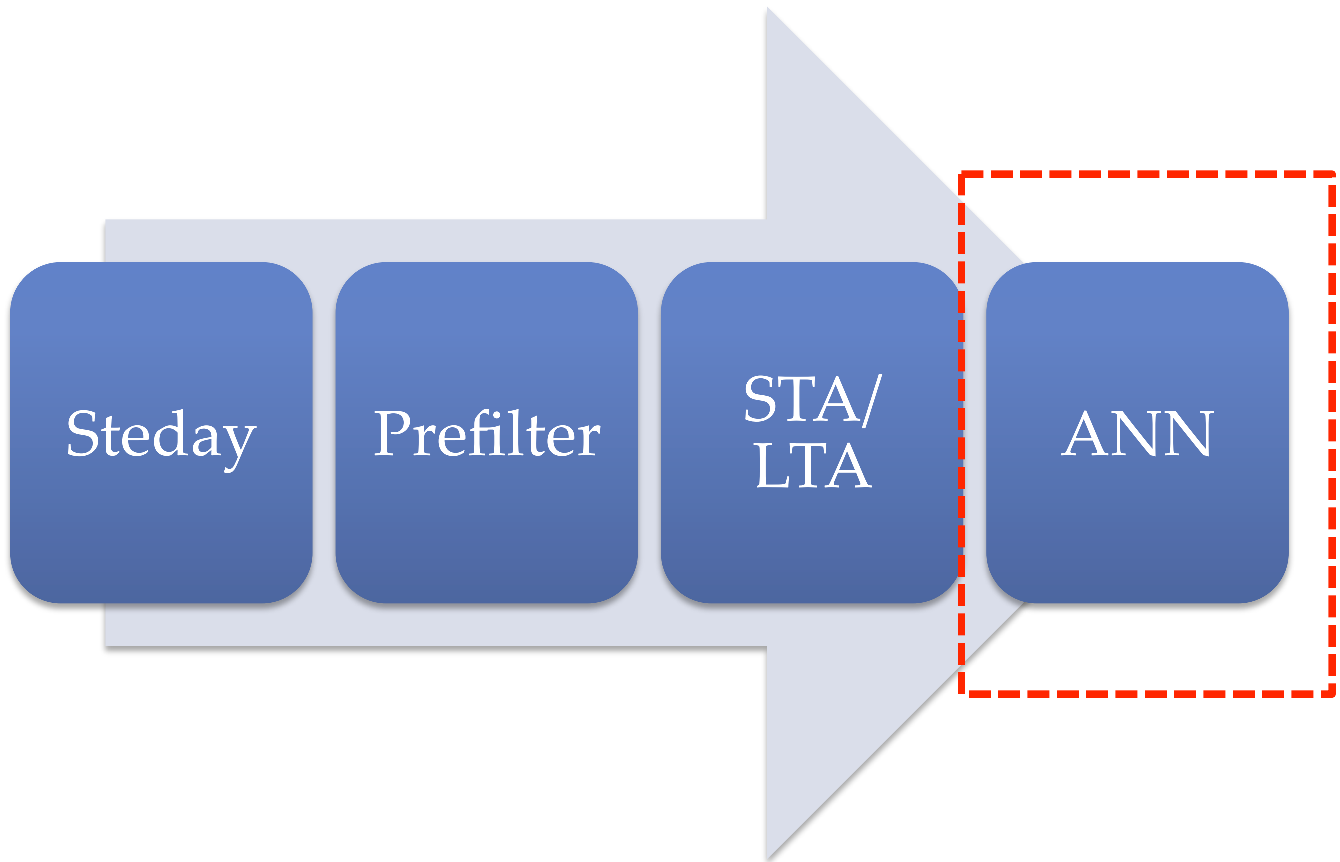


machine learning models
Data examples
01100
10110
11110
Tunable Model
Optimization algorithm
Trained Model
MAKE THINGS HAPPEN!

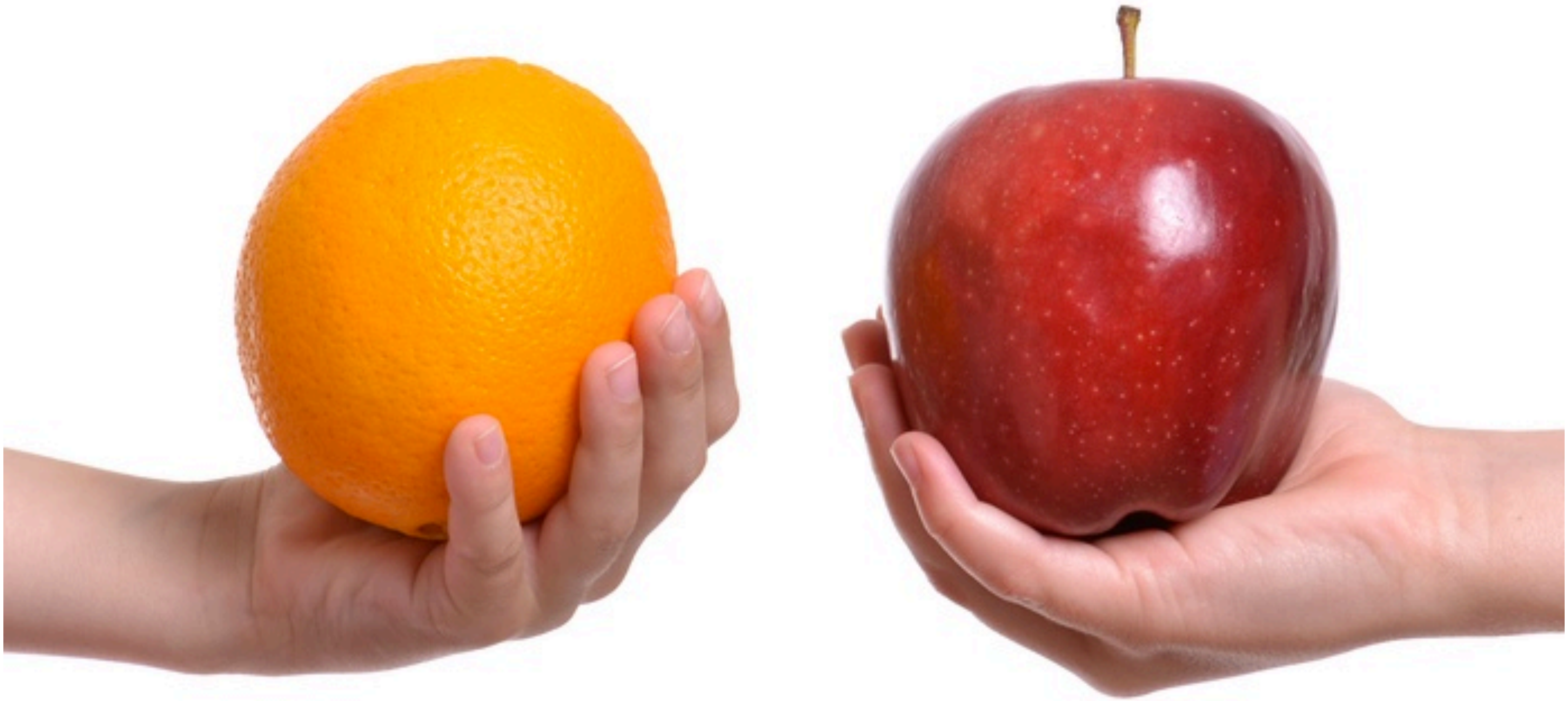
MyShake workflow



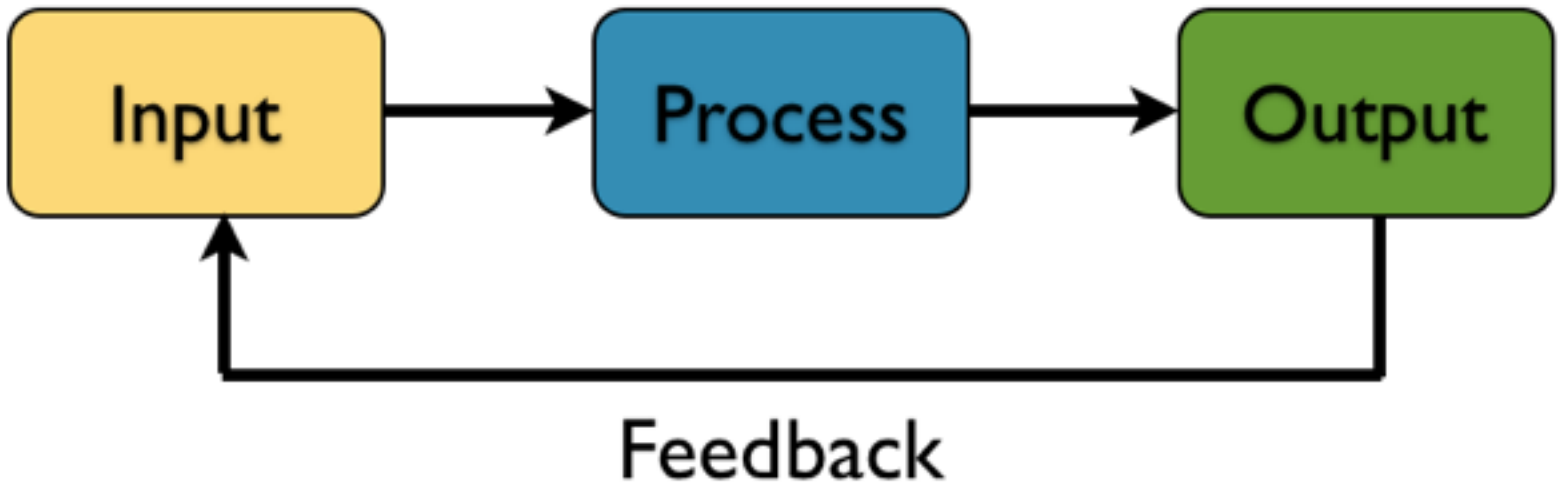
MyShake workflow



Crash course on ANN



ANN in simple view




YOU'RE IN MY SPOT
GRAPHICS GARAGE

Intuitive

Artificial Neural Network

Input

.
.
.

w_1

w_2

w_n

$F(eye \times w_1 + nose \times w_2 + \ldots + mouth \times w_n)$

Output

Intuitive

Artificial Neural Network

Input

.
.
.

w_1

w_2

w_n

F(eye×w_1+nose×w_2+…+mouth×w_n)

Output

error feedback

Intuitive

Artificial Neural Network

Input

.
.
.

w_1

w_2

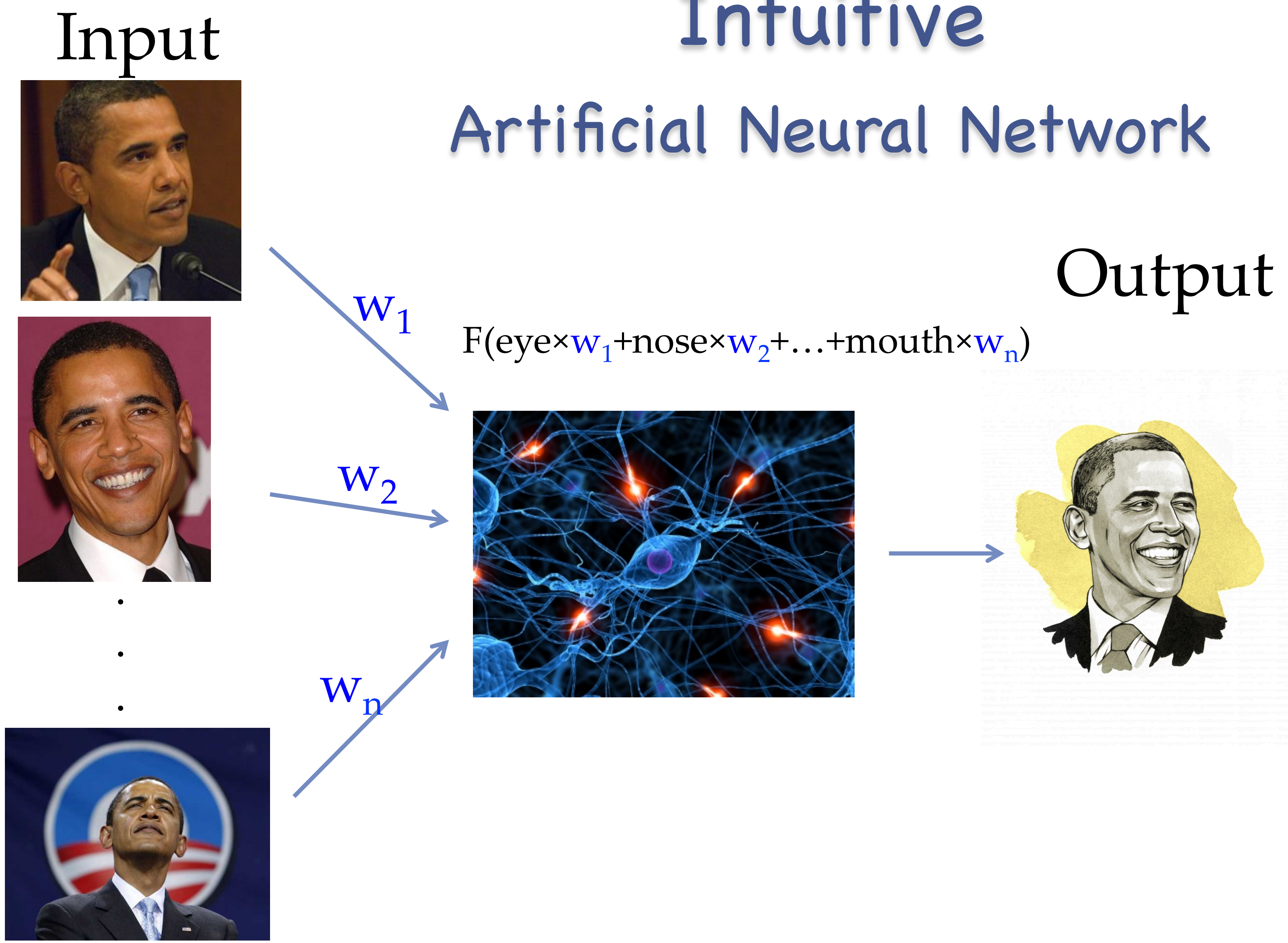
w_n

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

Output

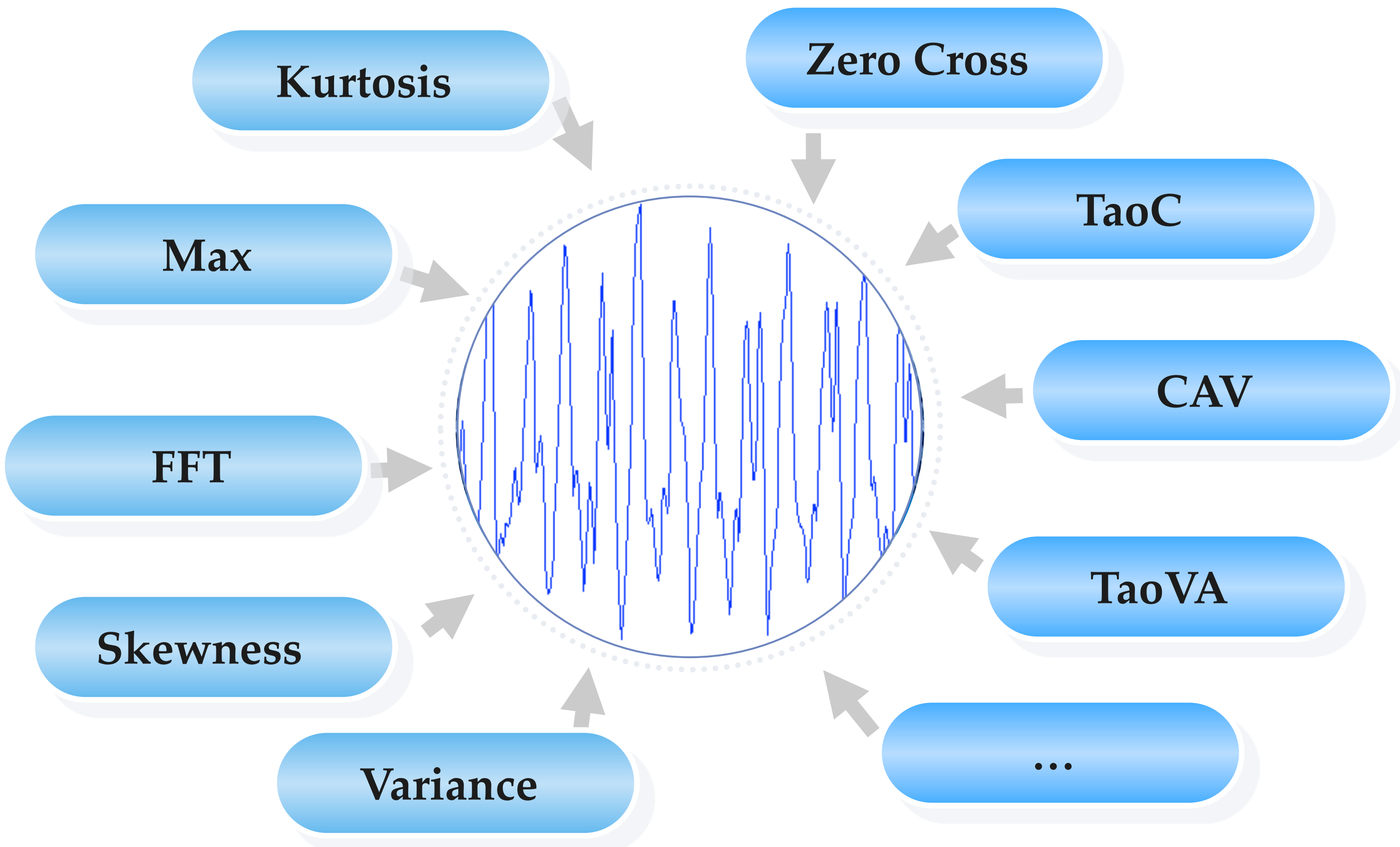
error feedback

Intuitive
Artificial Neural Network



Schema of ANN

Features Extracted

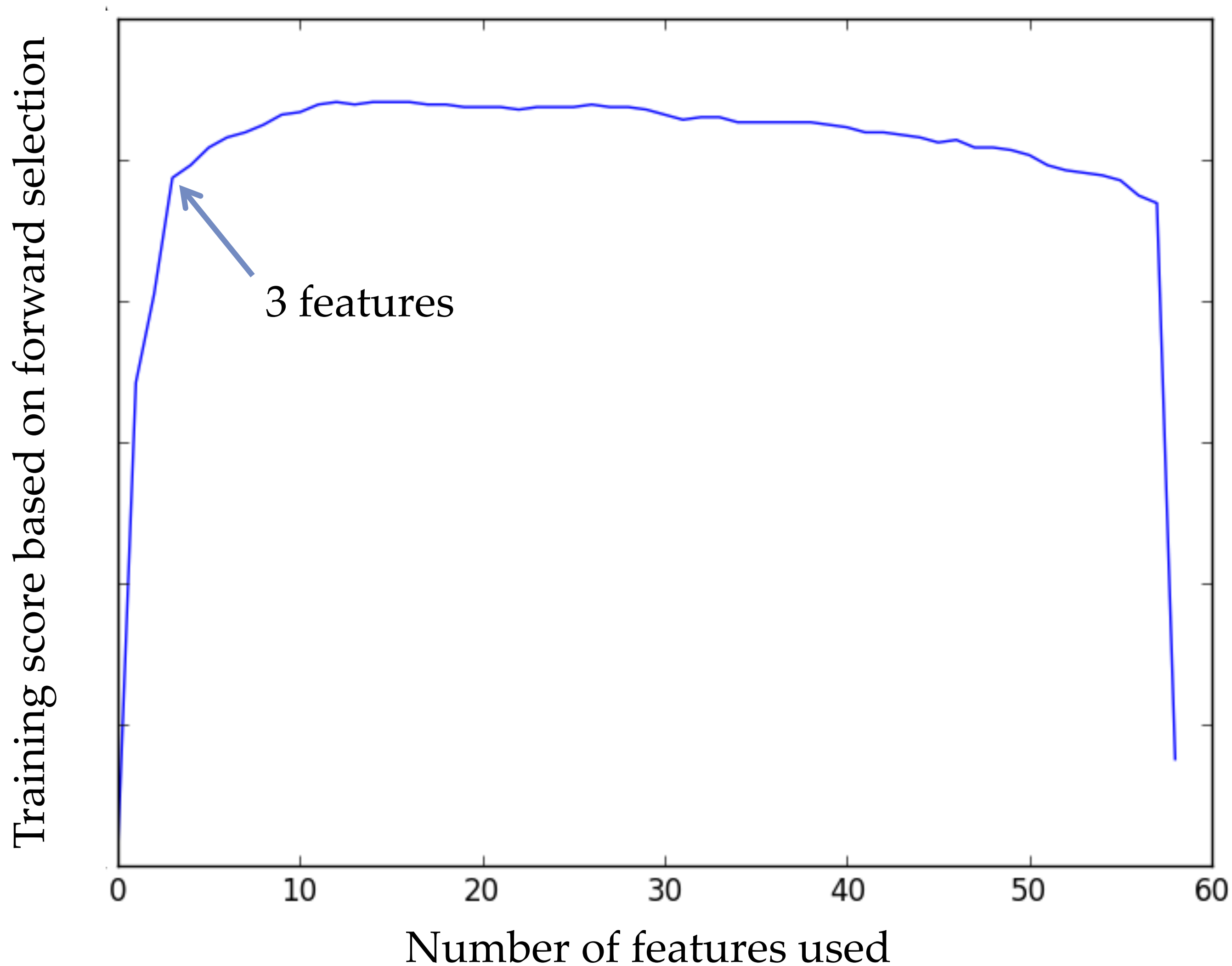


We tested 50 different features

Feature selection

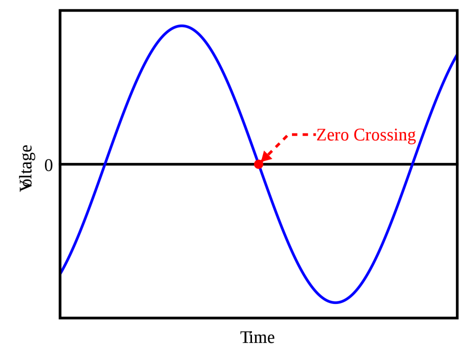
- **Forward greedy selection**
  - Step 1: Start with a null model
  - Step 2: Add one variable at a time, and record the improvement (accuracy or AIC).
  - Step 3: Select the feature with the max improvement on the model
  - Step 4: Repeat step 2 and 3 for all the features

Feature selection

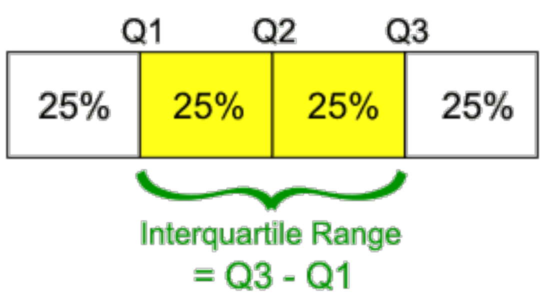


Calculating features

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

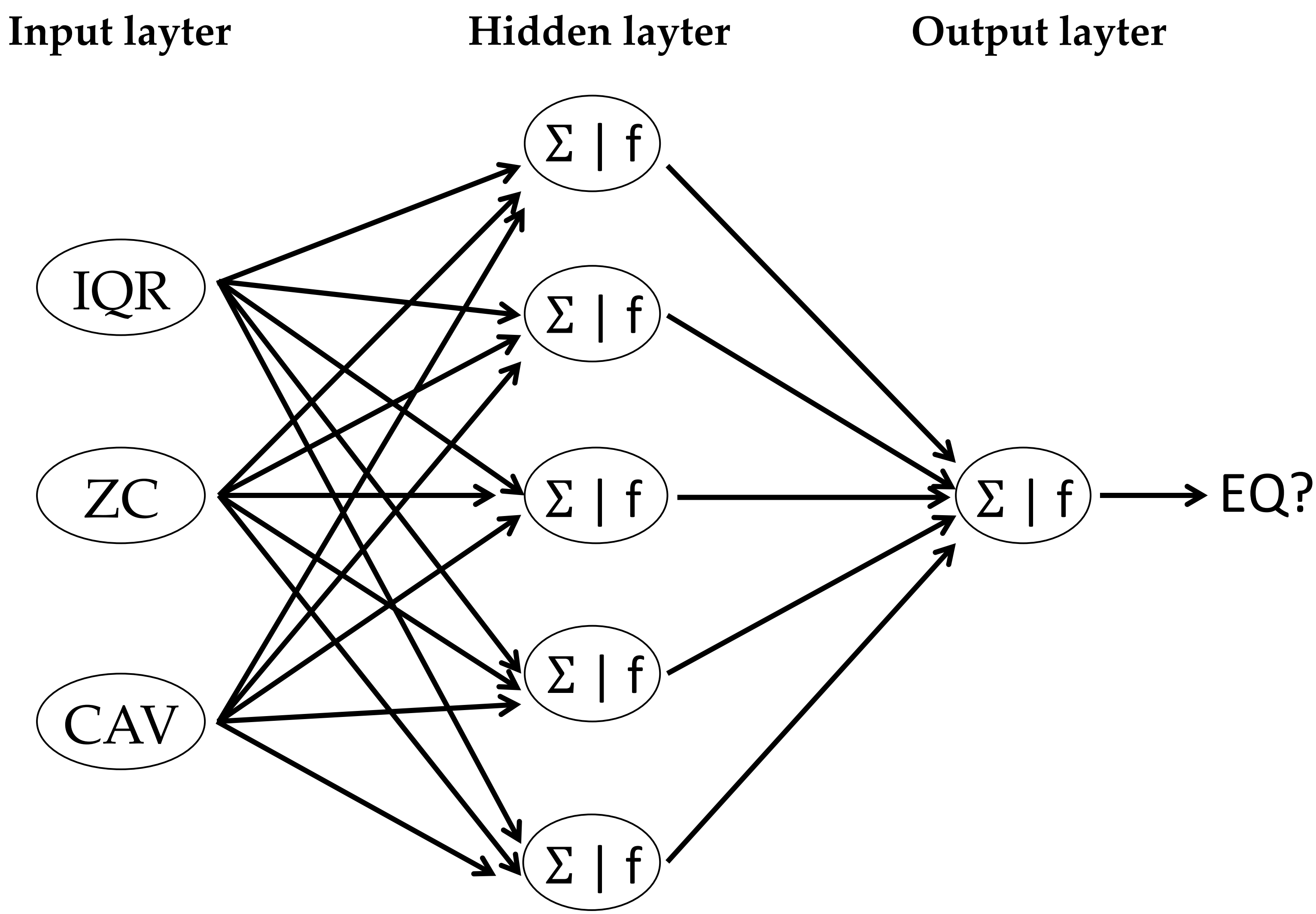


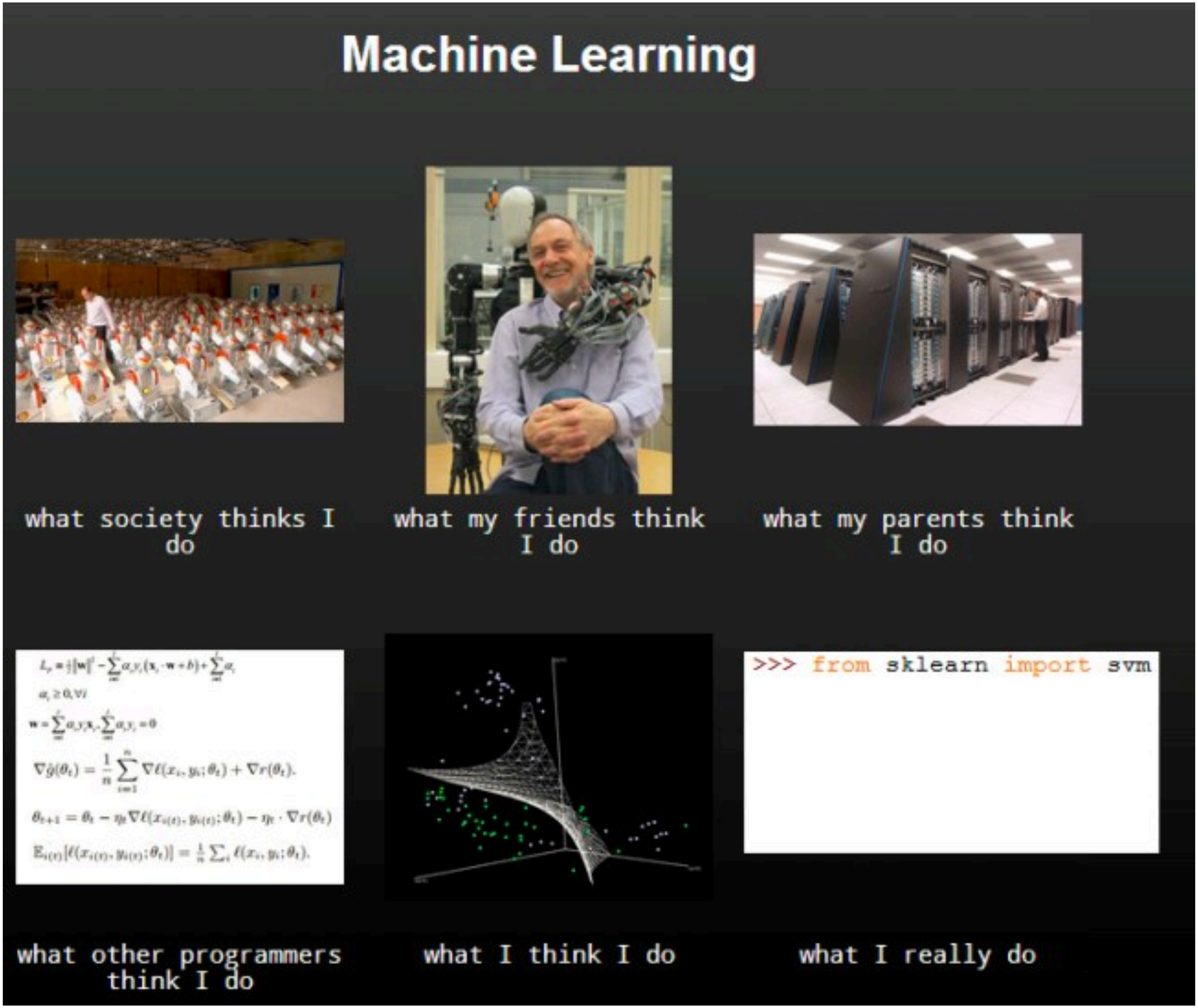
- IQR
  - $IQR = Q_3 - Q_1$

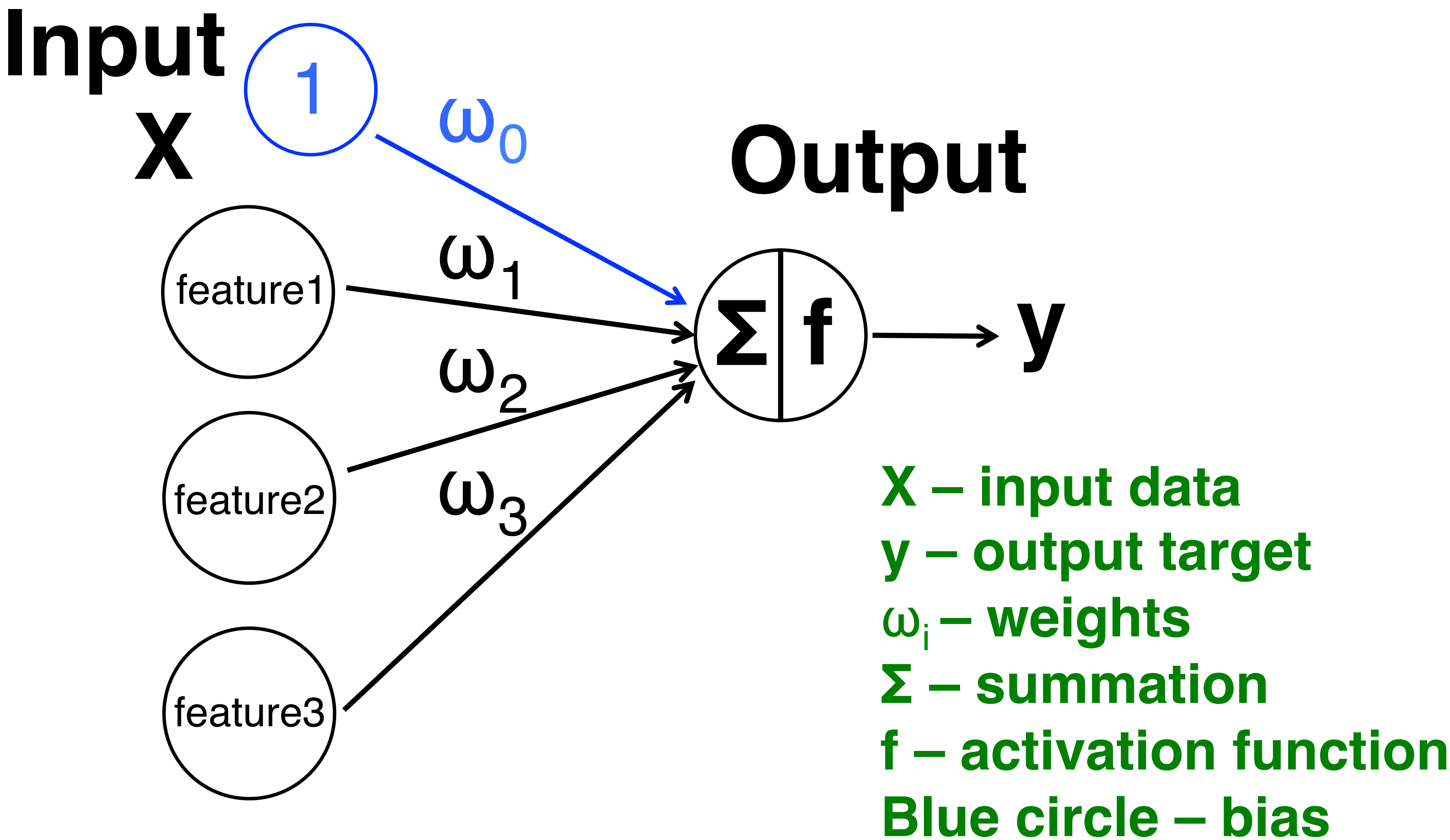


- CAV
  - $CAV = \int_0^2 |a(t)|\, dt$

ANN structure



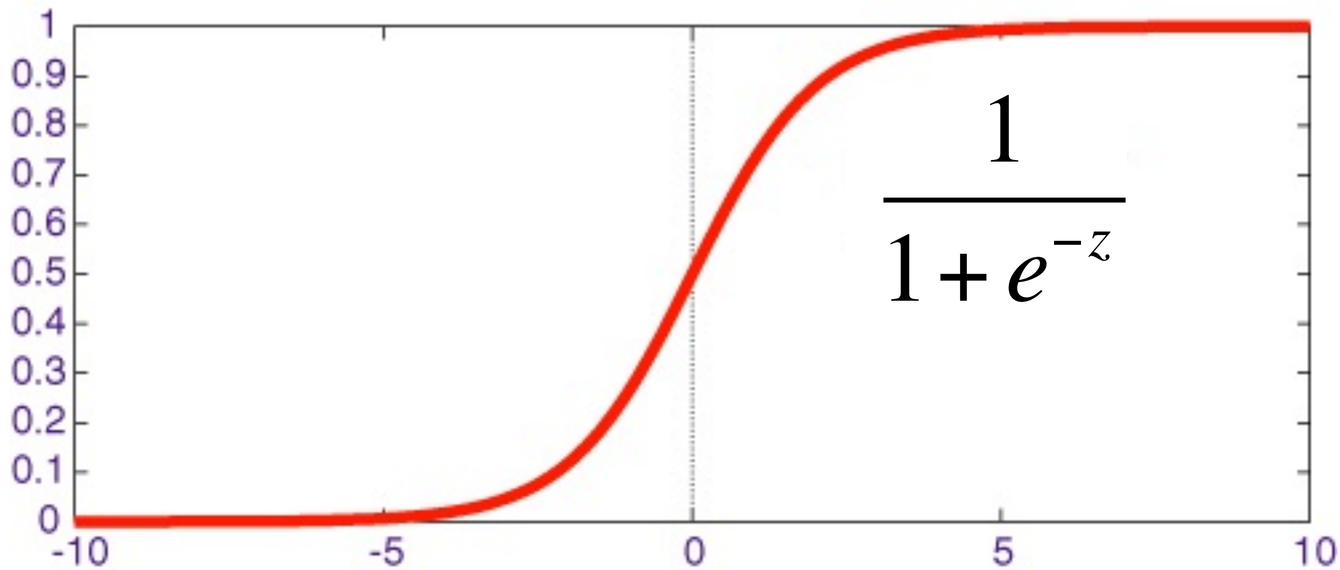

Machine Learning
what society thinks I do
what my friends think I do
what my parents think I do
>>> from sklearn import svm
what other programmers think I do
what I think I do
what I really do



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

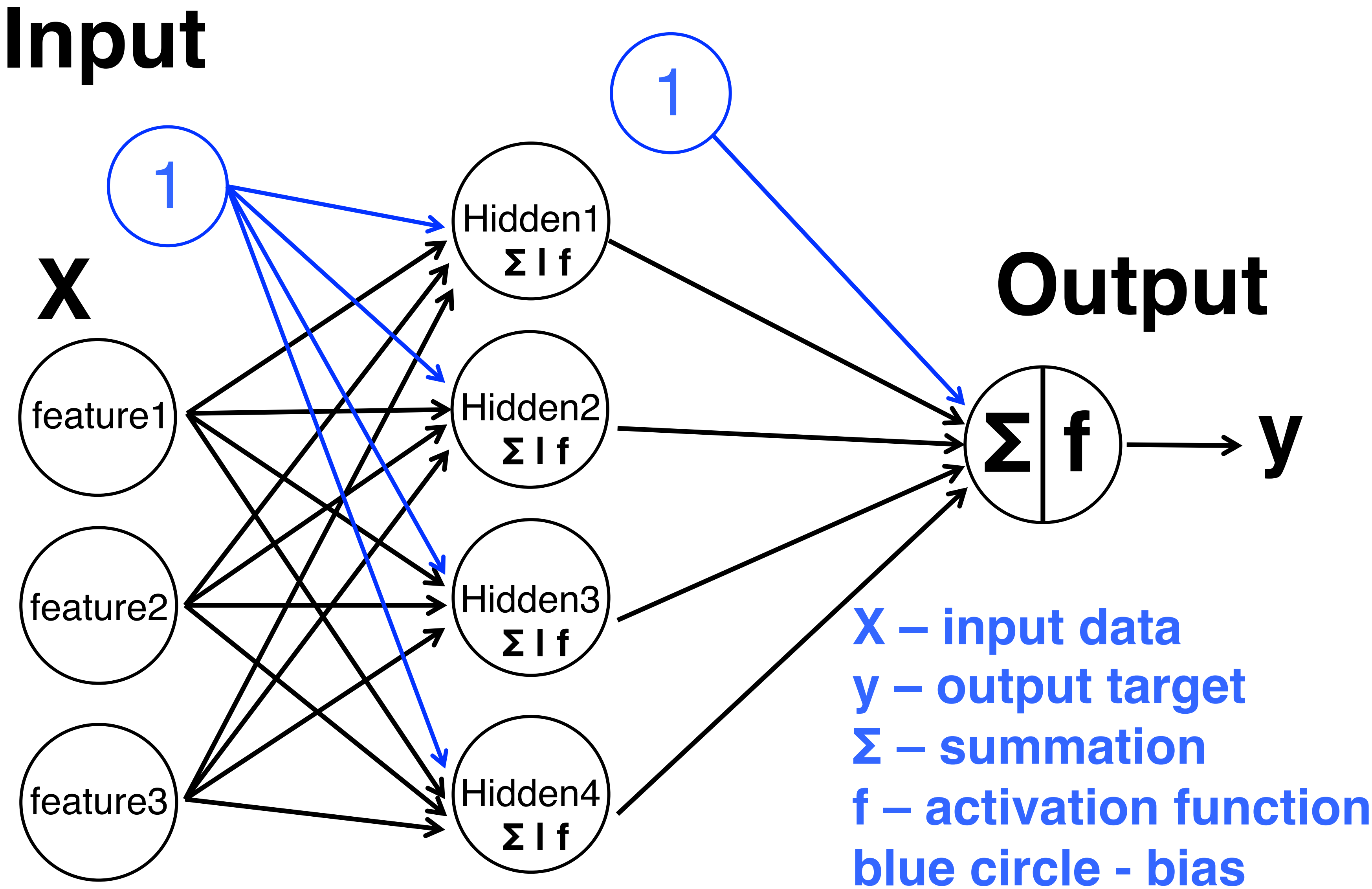
$$f = f(\omega_0 x_0 + \omega_1 x_1 + \omega_2 x_2 + \omega_3 x_3 + \ldots + \omega_n x_n)$$

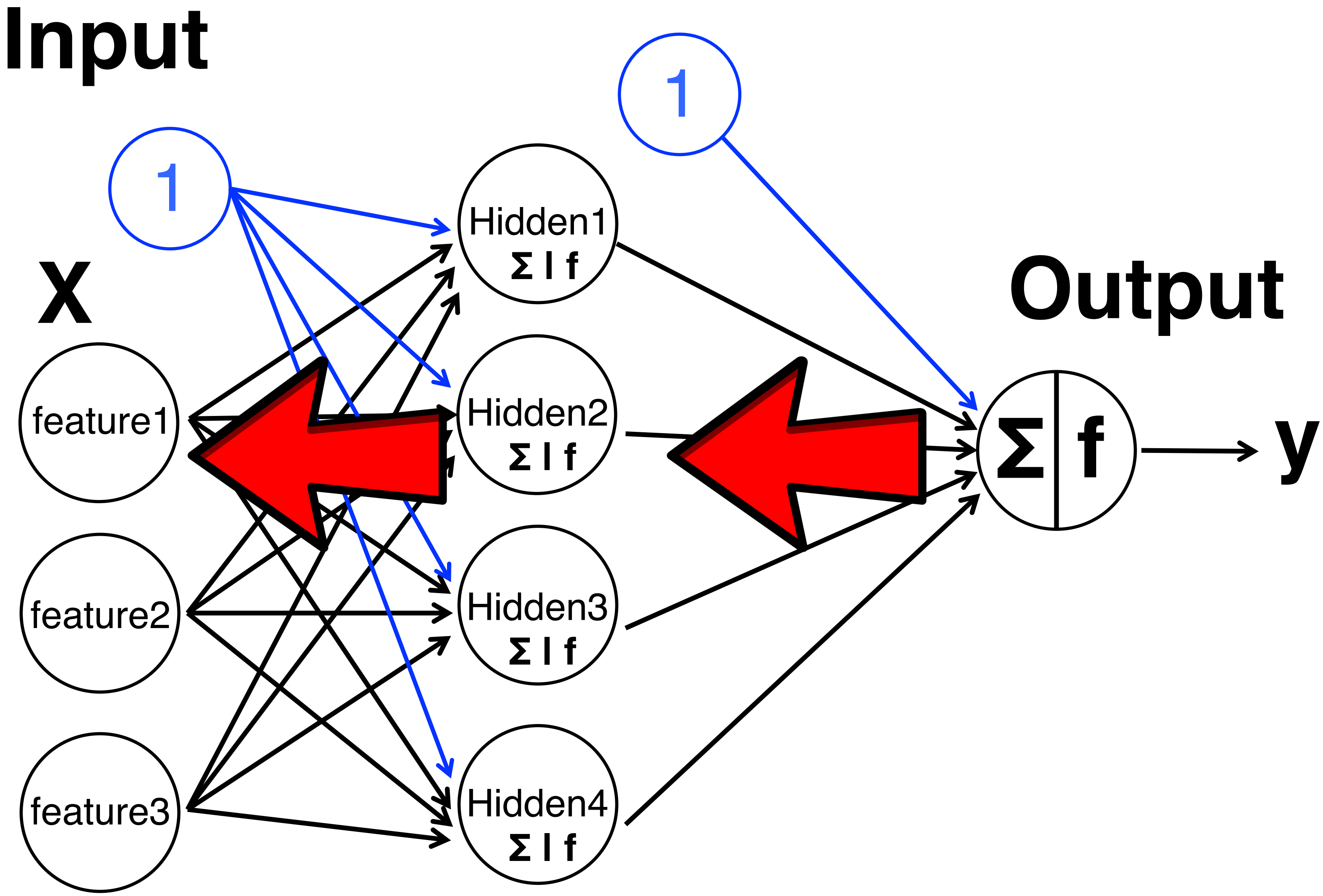
More activation function



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

$$f(z) = \frac{1}{1+e^{-z}} \qquad \frac{df(z)}{dx} = f(z)(1 - f(z))$$

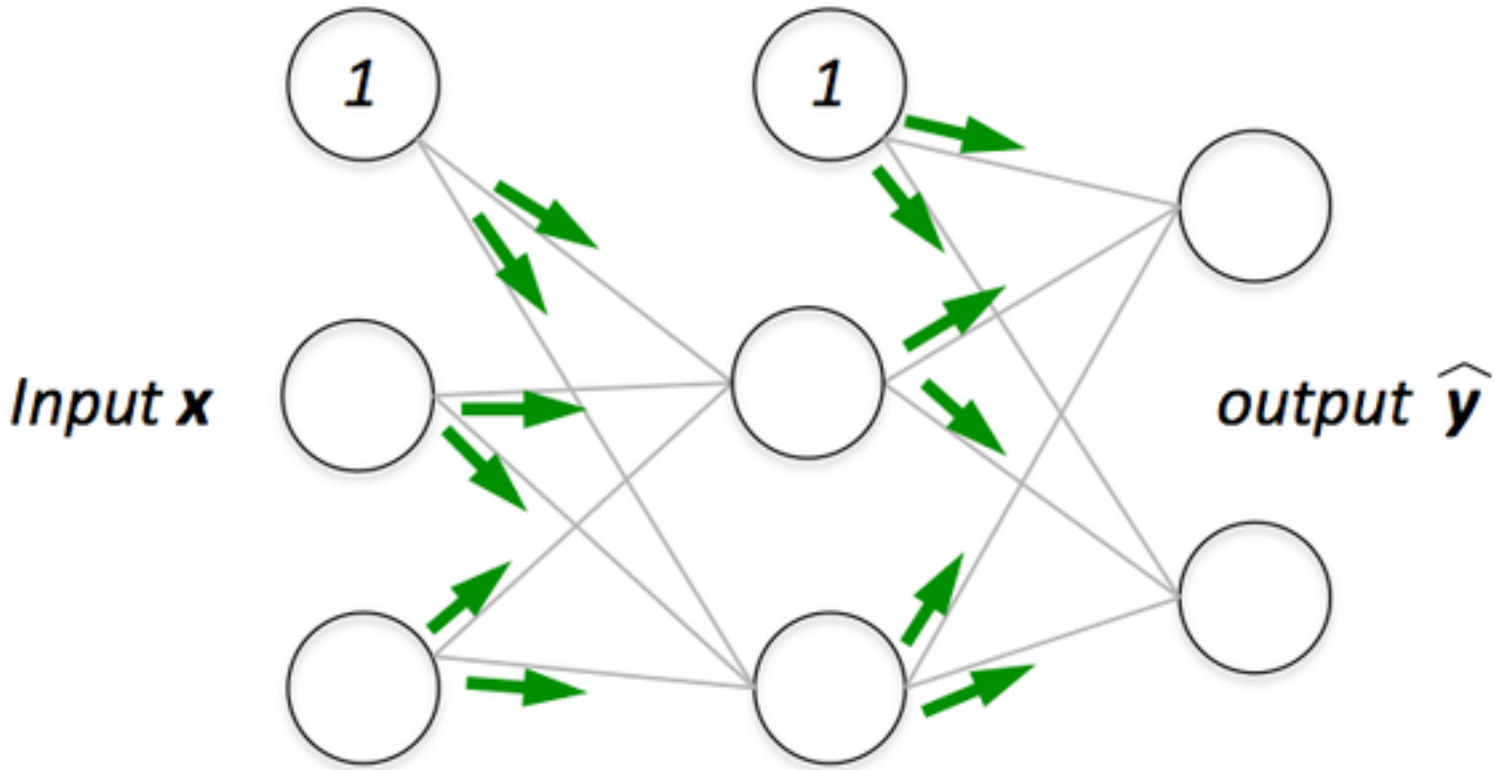

Input
X
1
feature1
feature2
feature3
Hidden1 Σ l f
Hidden2 Σ l f
Hidden3 Σ l f
Hidden4 Σ l f
1
Output
Σ f
y
X – input data
y – output target
Σ – summation
f – activation function
blue circle - bias


Input
X
1
1
feature1
feature2
feature3
Hidden1
Σ | f
Hidden2
Σ | f
Hidden3
Σ | f
Hidden4
Σ | f
Output
Σ
f
y

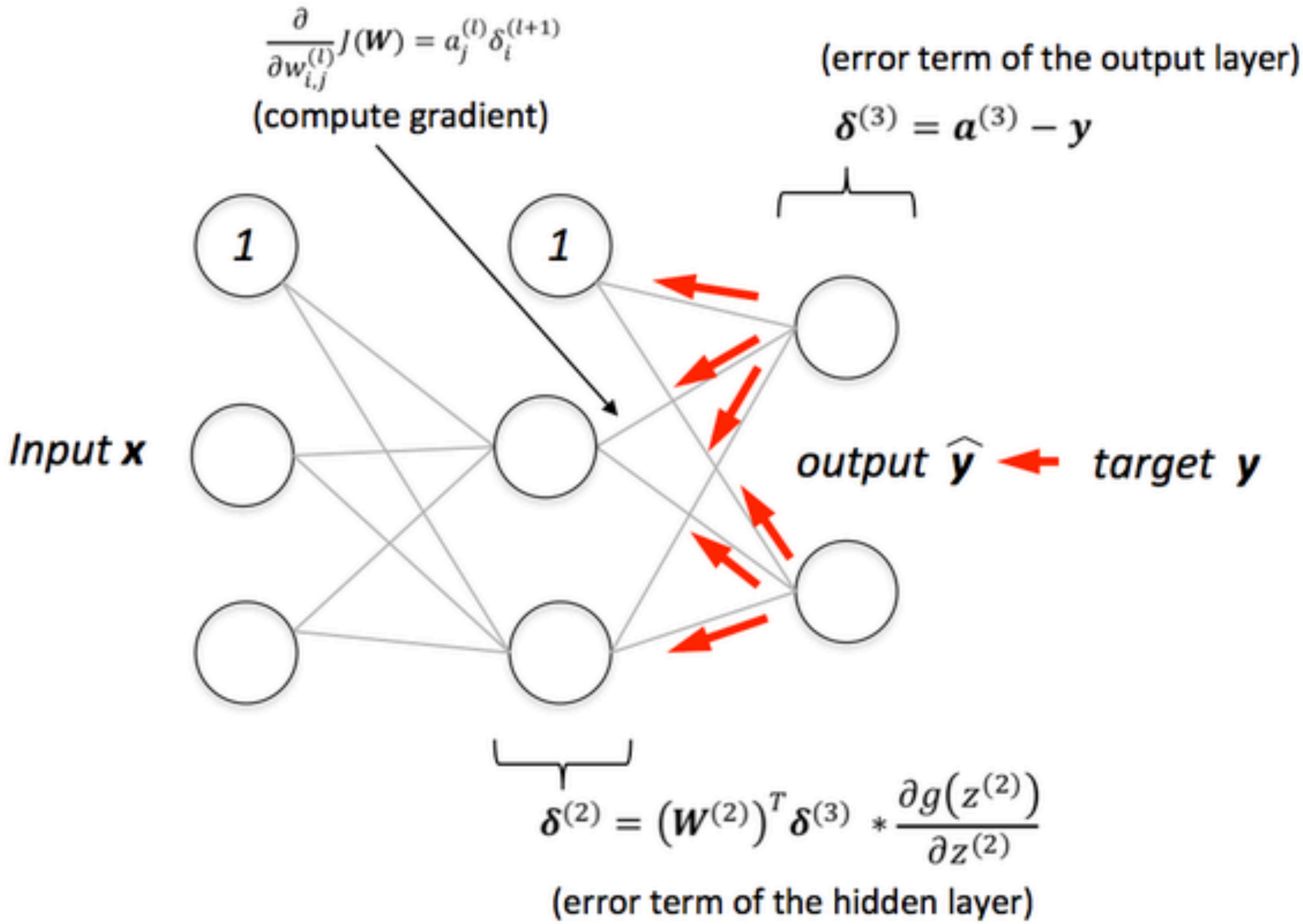
Backpropagation training algorithm

- 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

$$w_{m+1} = w_m + \Delta w_m \qquad \Delta w_m = -\beta \, d_m$$



1
1
Input x
output ŷ


$\frac{\partial}{\partial w_{i,j}^{(l)}} J(\boldsymbol{W}) = a_j^{(l)} \delta_i^{(l+1)}$
(compute gradient)
(error term of the output layer)
$\boldsymbol{\delta}^{(3)} = \boldsymbol{a}^{(3)} - \boldsymbol{y}$
1
1
Input **x**
output $\widehat{\boldsymbol{y}}$
target **y**
$\boldsymbol{\delta}^{(2)} = \left(\boldsymbol{W}^{(2)}\right)^T \boldsymbol{\delta}^{(3)} * \frac{\partial g\left(z^{(2)}\right)}{\partial z^{(2)}}$
(error term of the hidden layer)

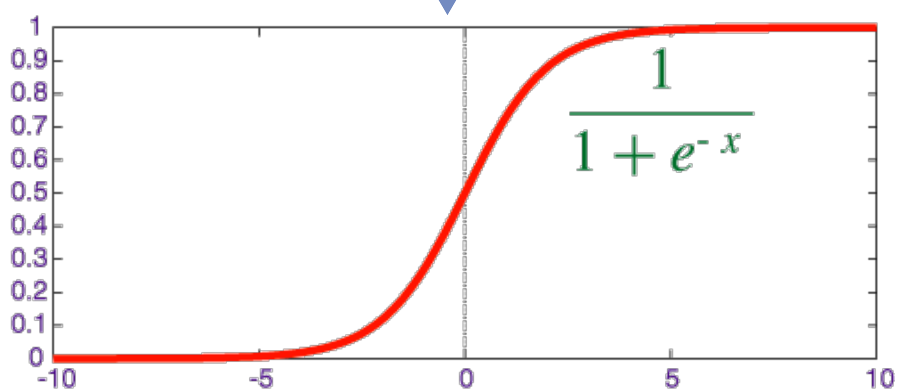
First layer calculation

intercept

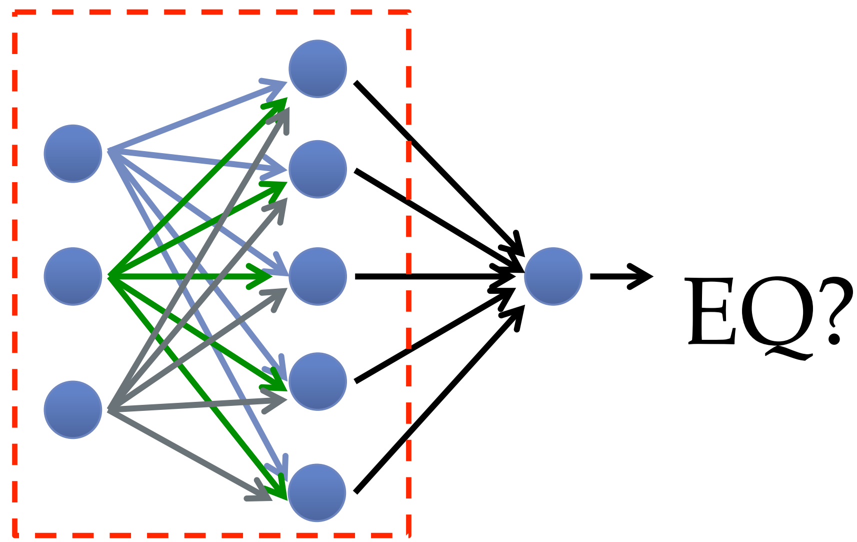
$$[x_1, x_2, x_3] \bullet \begin{bmatrix} w_{11}^{(1)}, w_{12}^{(1)}, w_{13}^{(1)}, w_{14}^{(1)}, w_{15}^{(1)} \\ w_{21}^{(1)}, w_{22}^{(1)}, w_{23}^{(1)}, w_{24}^{(1)}, w_{25}^{(1)} \\ w_{31}^{(1)}, w_{32}^{(1)}, w_{33}^{(1)}, w_{34}^{(1)}, w_{35}^{(1)} \end{bmatrix} + [c_1^{(1)}, c_2^{(1)}, c_3^{(1)},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

$$= [a_1, a_2, a_3, a_4, a_5]$$

Activation function

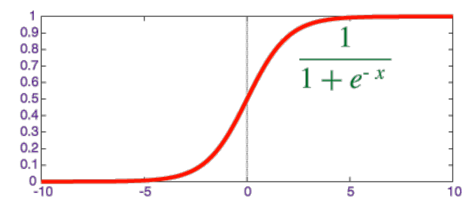


$$[A_1, A_2, A_3, A_4, A_5]$$



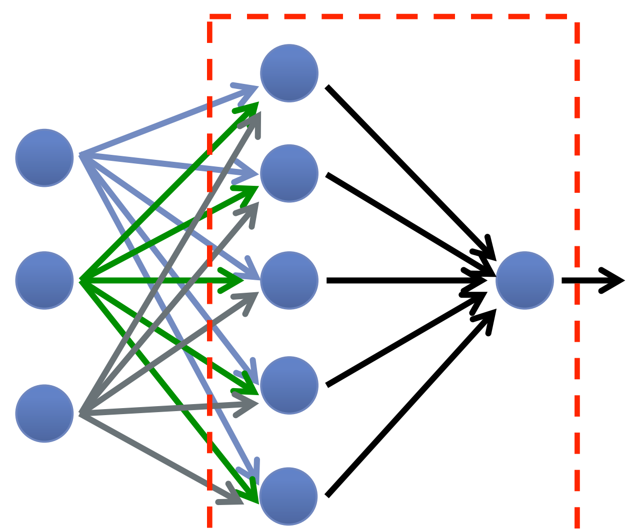
Second layer calculation

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



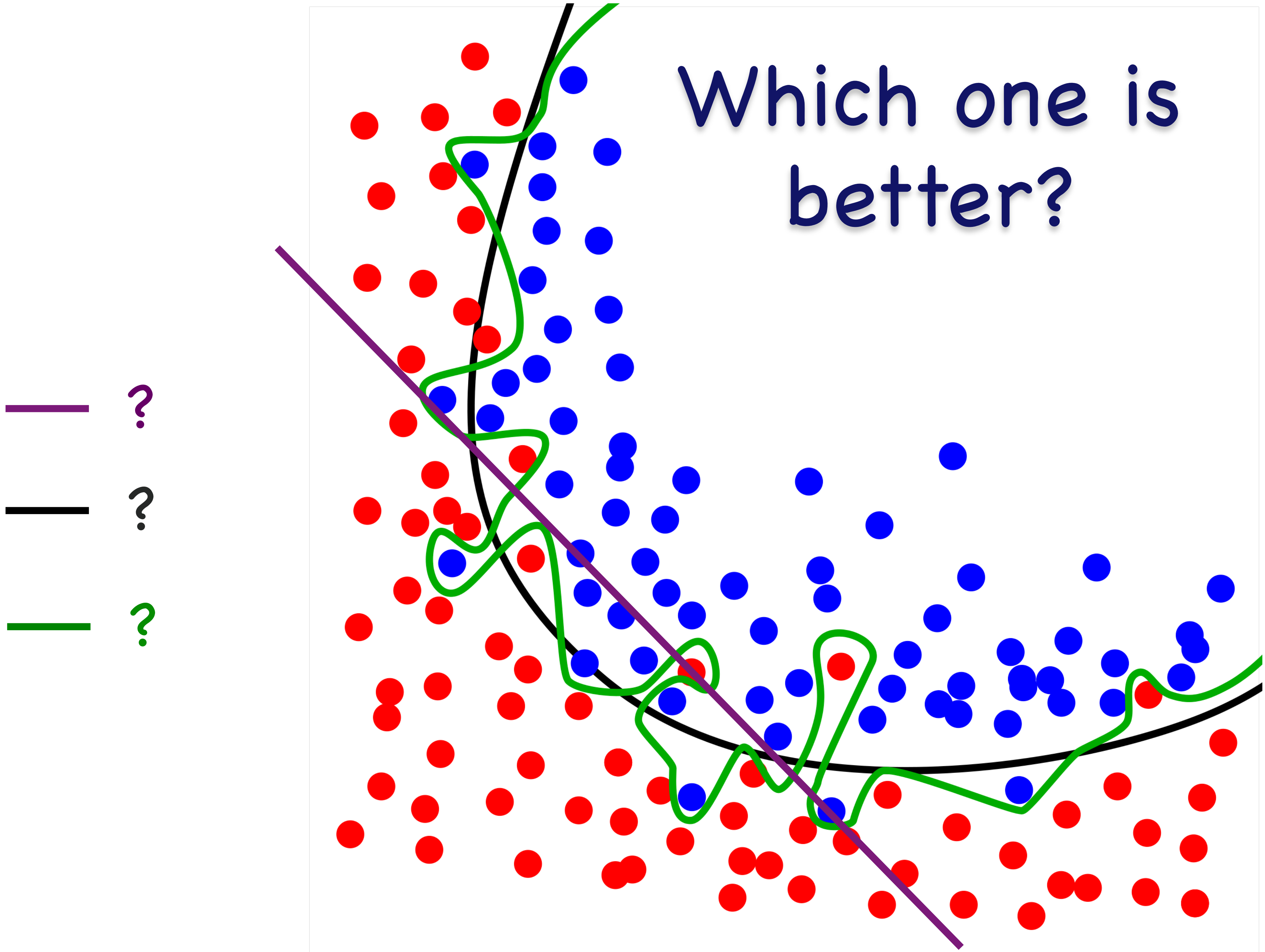
Activation function

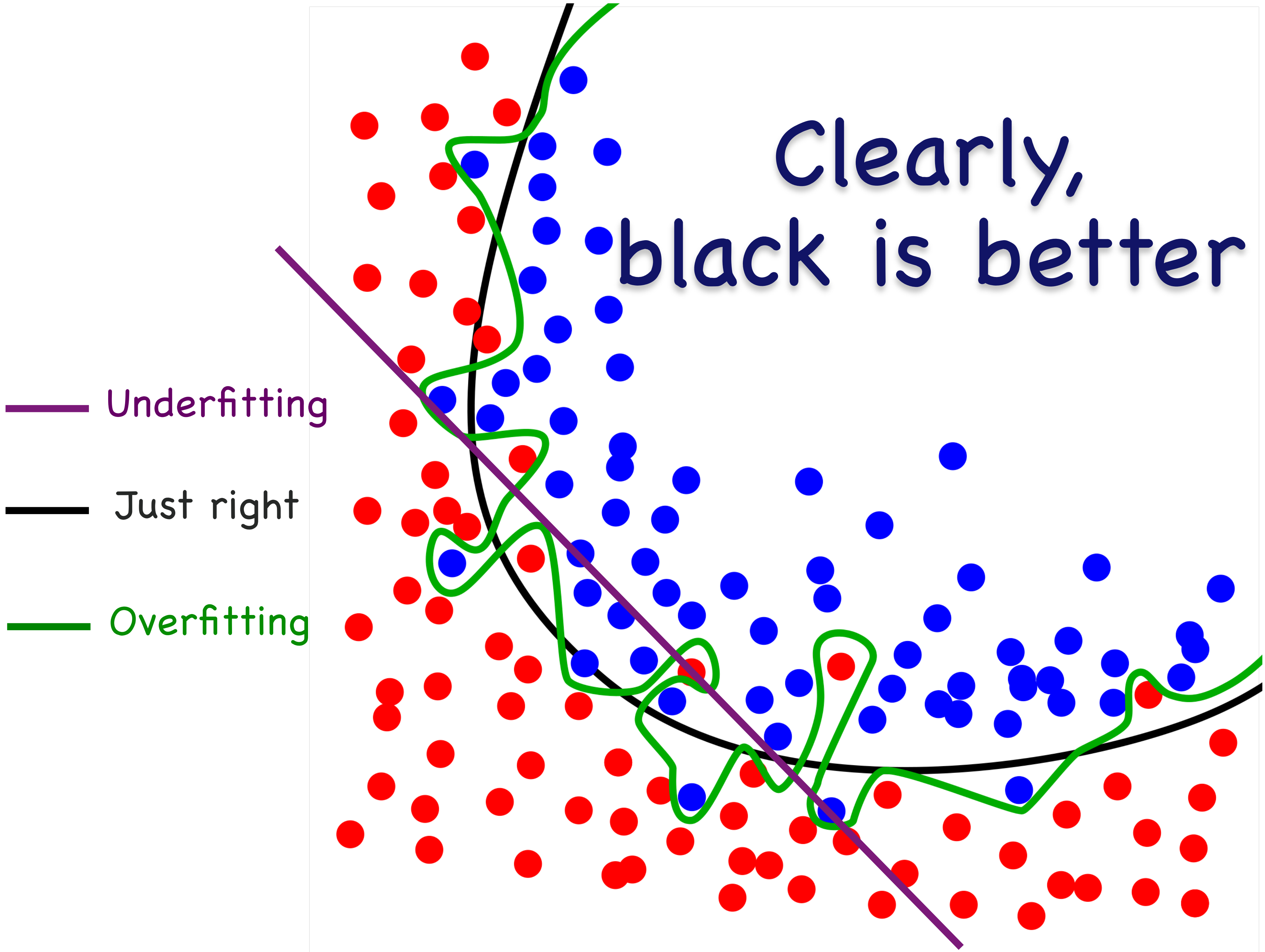
$B \longrightarrow$ EQ?



EQ?

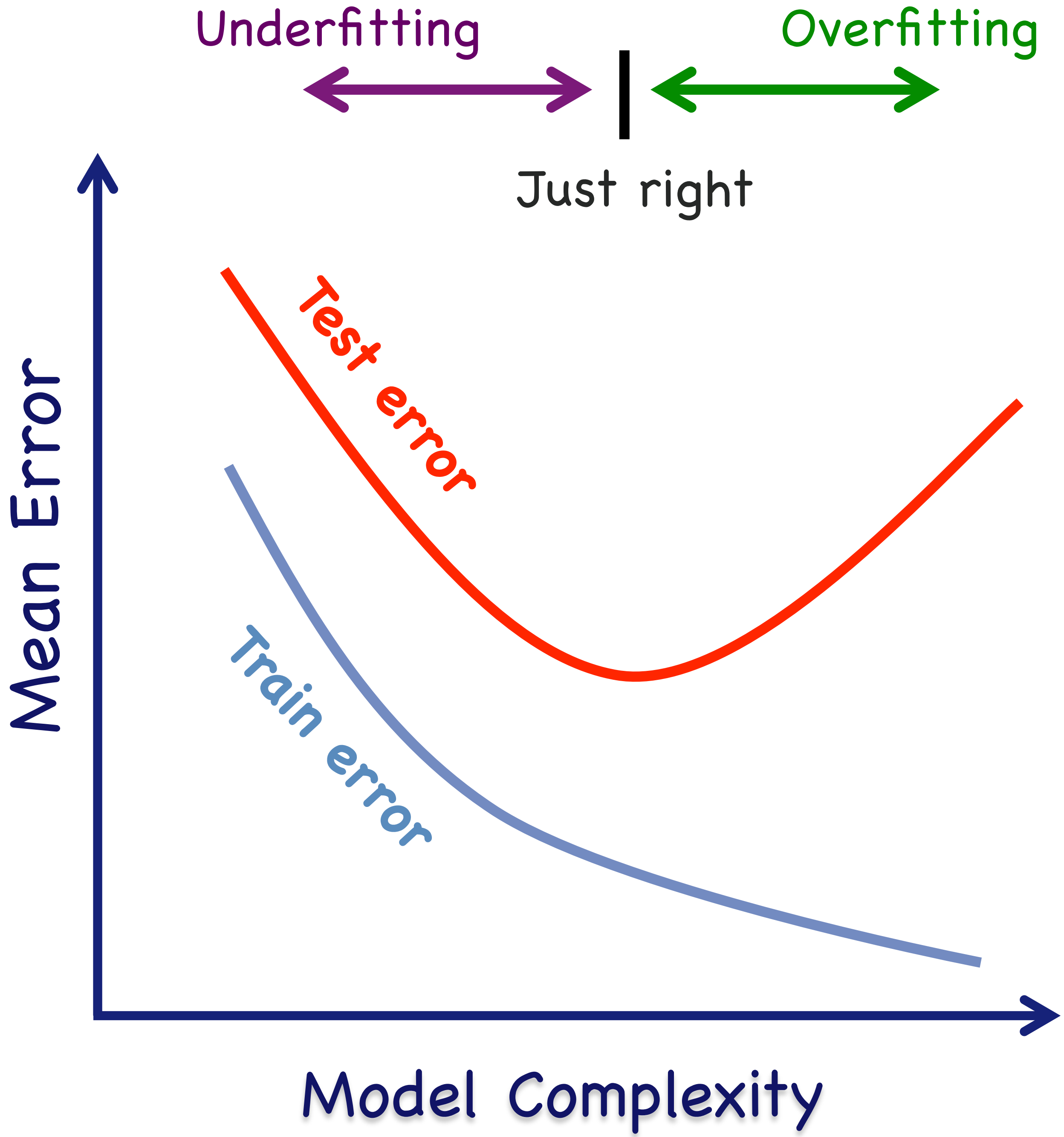
Details of the training


Which one is better?
?
?
?


Underfitting
Just right
Overfitting
Clearly, black is better

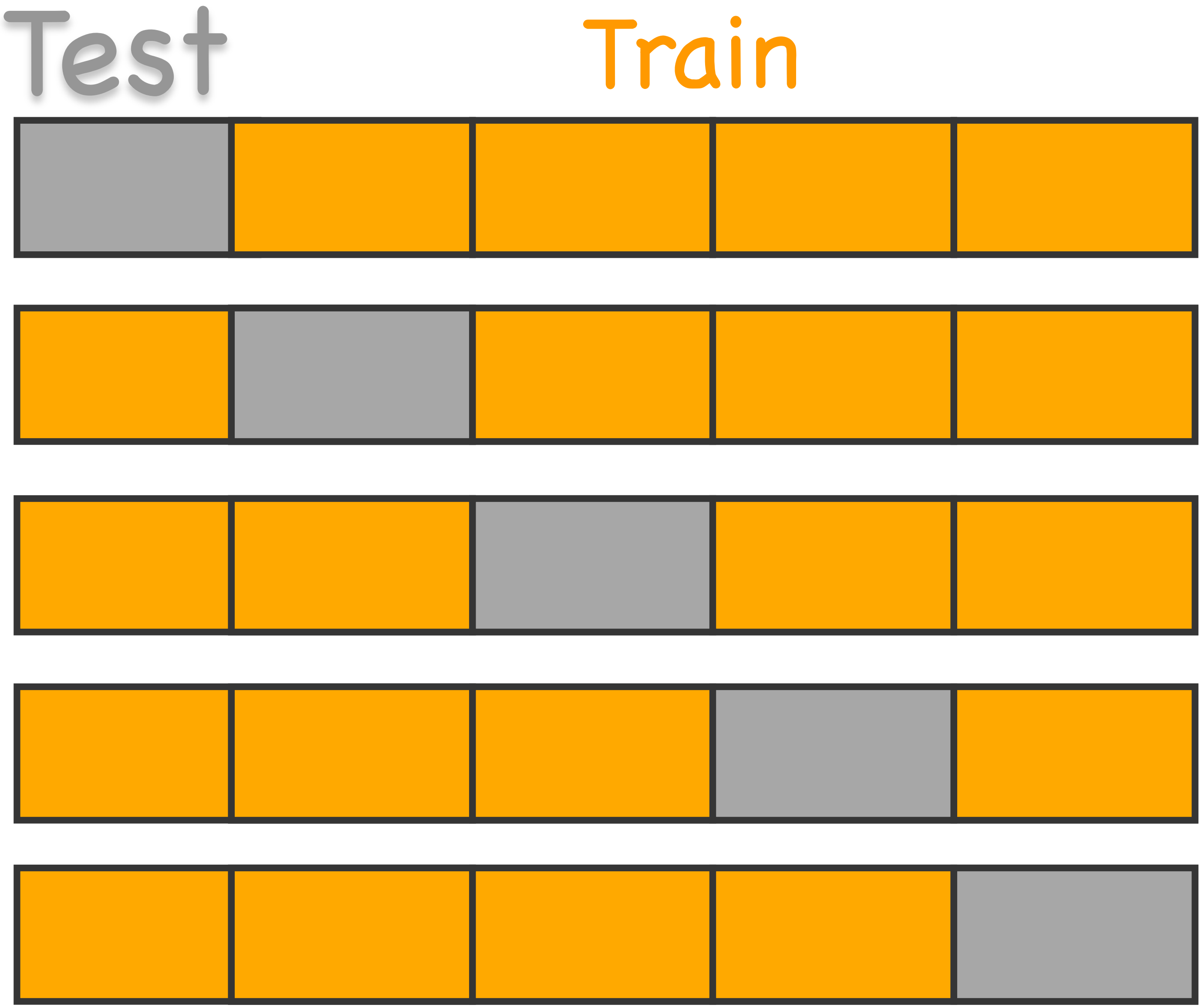
Train/test dataset split

Train/test dataset split



Underfitting
Overfitting
Just right
Test error
Train error
Mean Error
Model Complexity

5-fold cross-validation



Results of 10 fold CV

		Length of time window									
		1	2	3	4	5	6	7	8	9	10
Number of neuron	20	2.1	2.02	2.36	2.08	2.34	2.06	2.76	2.85	3.39	3.74
	19	2.06	2.02	1.72	2.14	3.09	2.71	2.5	2.7	2.59	3.97
	18	2.05	2.08	2.22	2.2	2.42	2.39	2.23	3.45	2.4	4.21
	17	2.15	2.08	2.22	2.2	2.01	2.39	2.37	3.75	2.99	4.91
	16	2.26	1.99	2.09	1.96	1.75	2.49	2.23	3	2.99	3.27
	15	2.5	2.02	1.86	2.44	1.75	2.39	2.1	3.45	3.59	3.74
	14	2.54	2.22	1.95	1.96	2.34	2.82	3.15	3.15	3.39	8.18
	13	2.38	2.28	2	2.14	2.26	2.17	2.89	2.4	4.19	6.78
	12	2.17	2.02	2.04	2.08	2.26	2.28	2.1	2.7	3.99	4.44
	11	2.38	2.05	2.45	2.08	2.09	2.06	2.5	3	2.4	4.44
	10	2.21	2.3	1.9	1.89	2.26	2.06	1.84	3.6	2.79	4.67

Imbalanced data

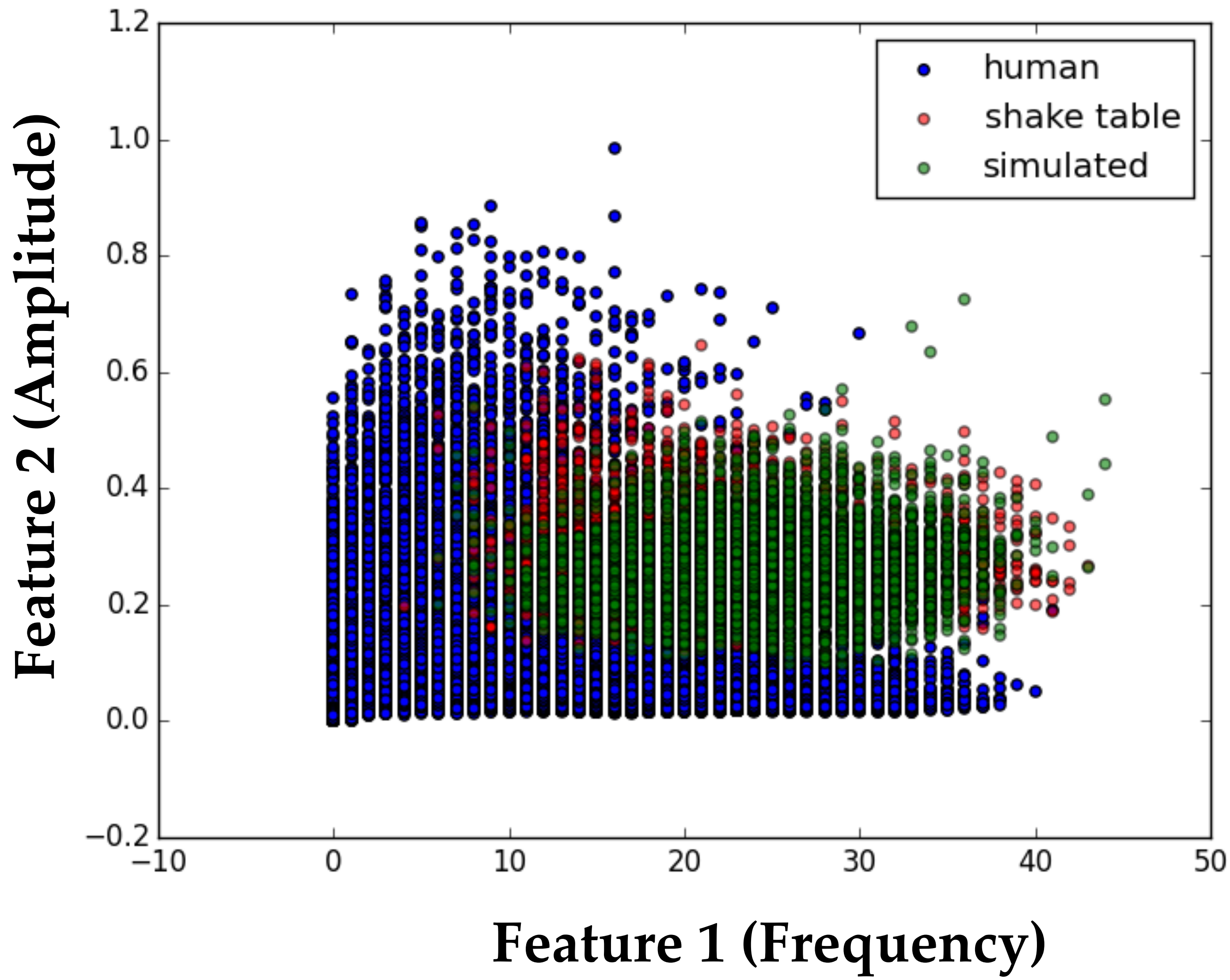
9000 vs 1000

9000 / (1000 + 9000) = 90%

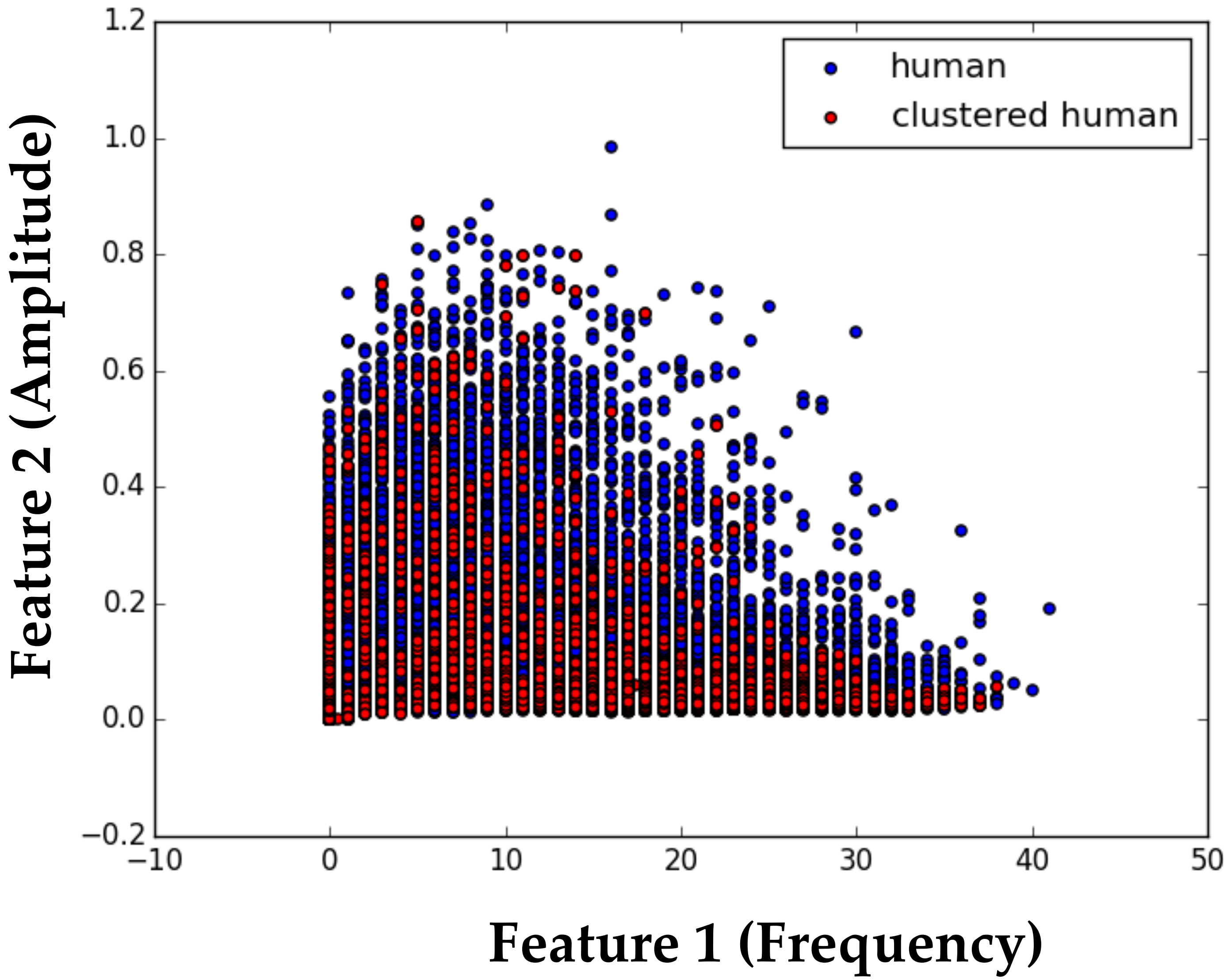
Address imbalance

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

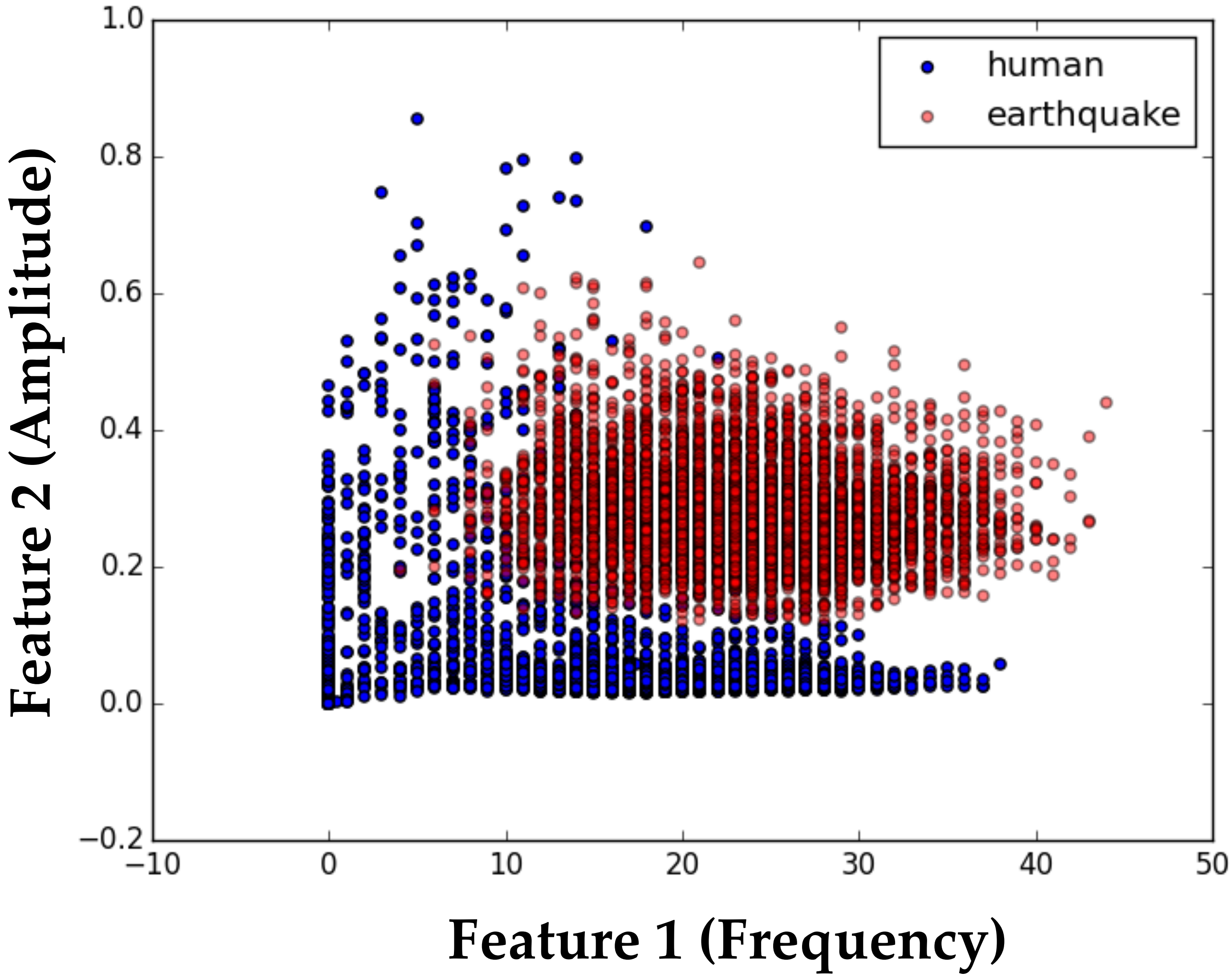
Training data



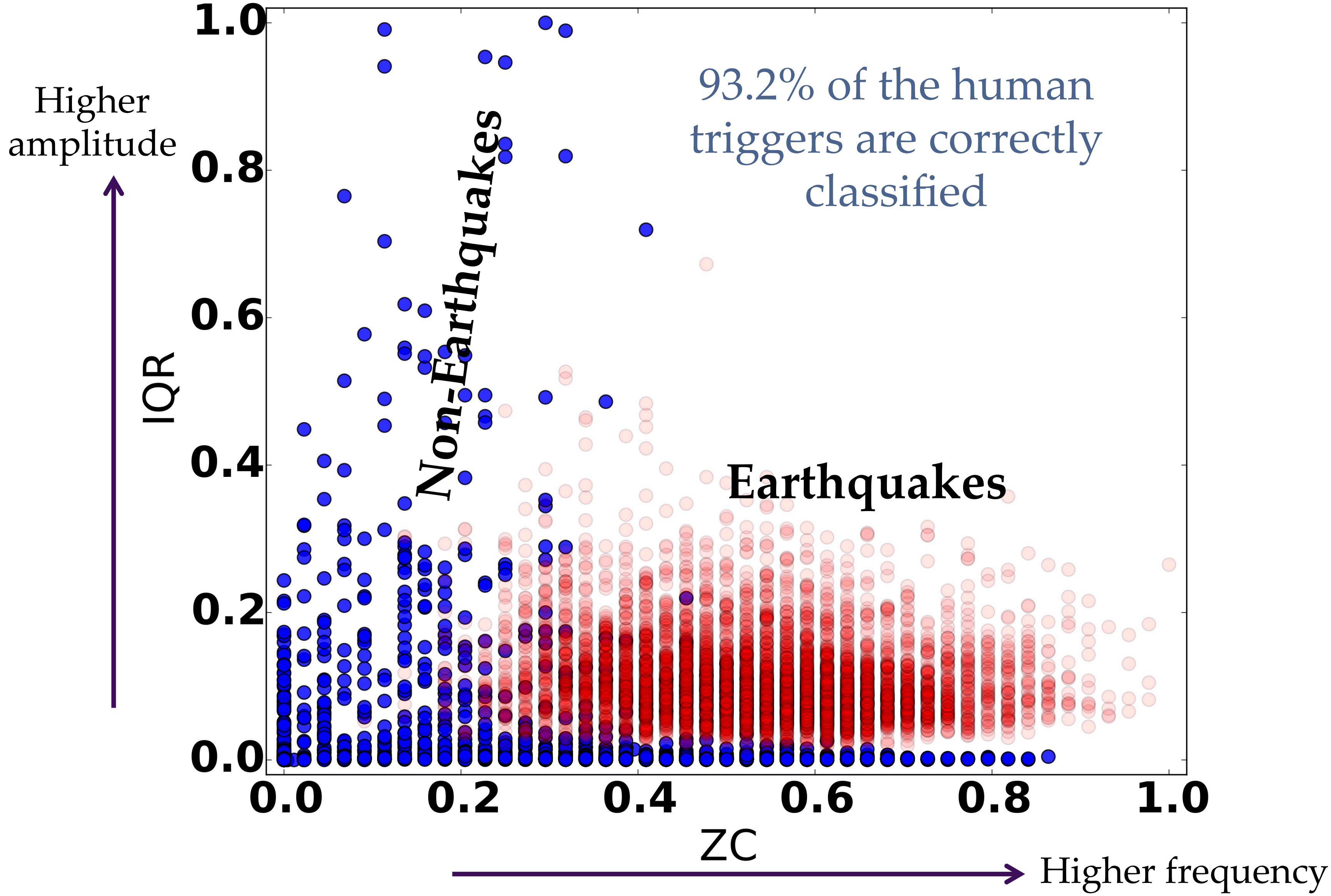
Clustered human data

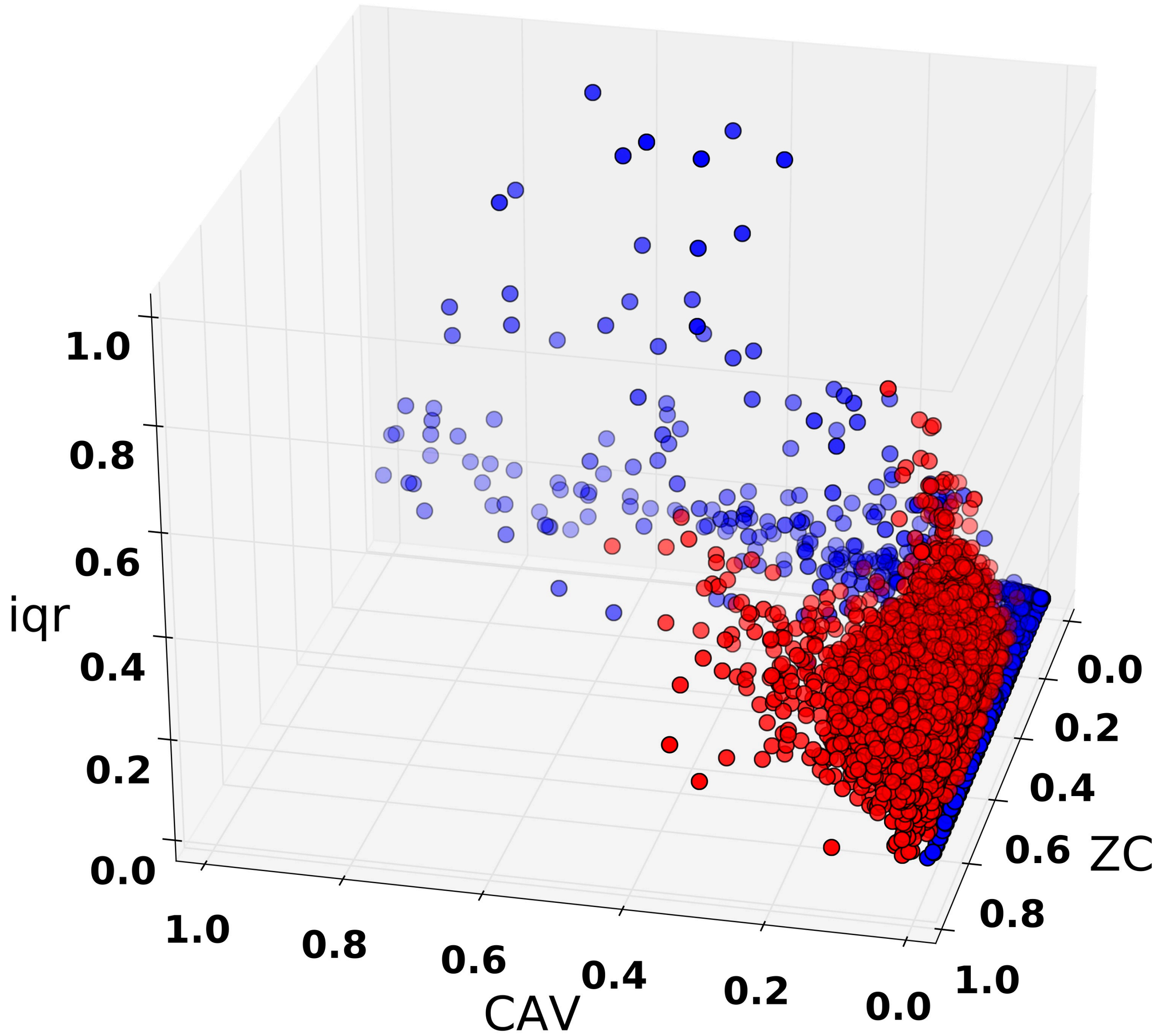


Final dataset



Classification

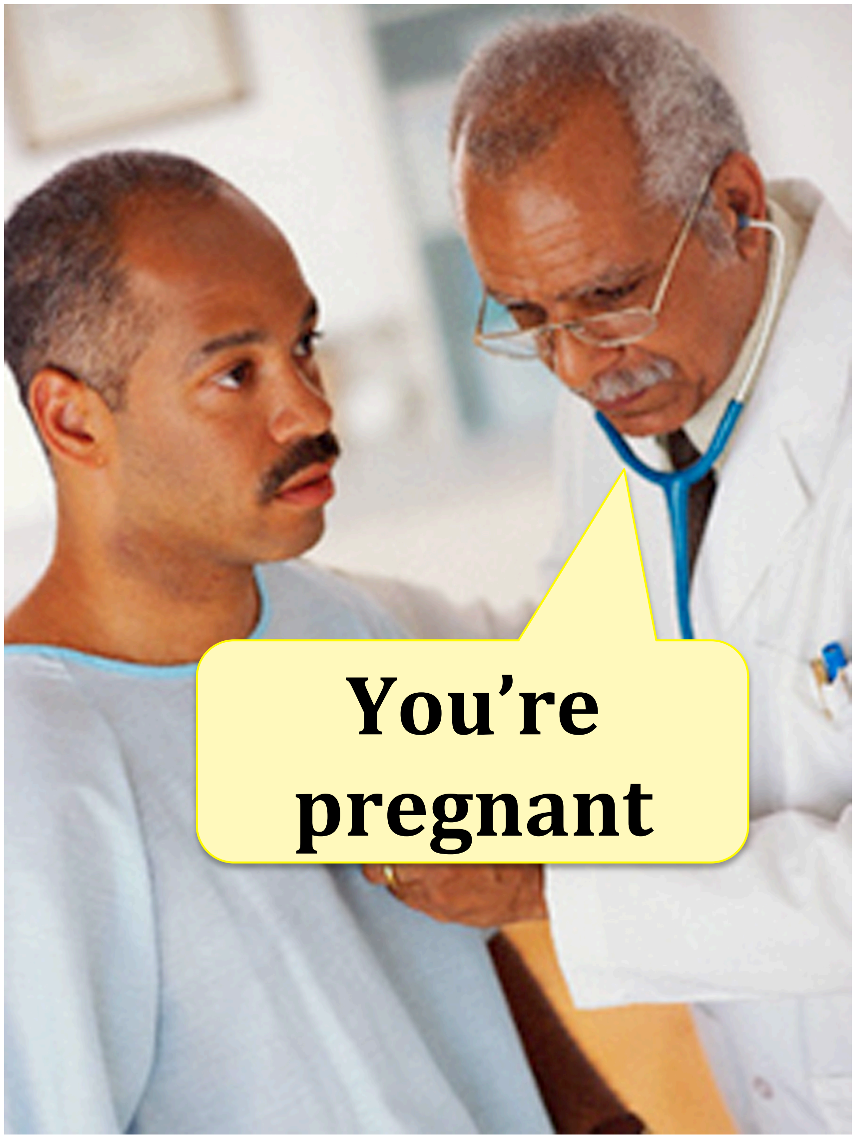




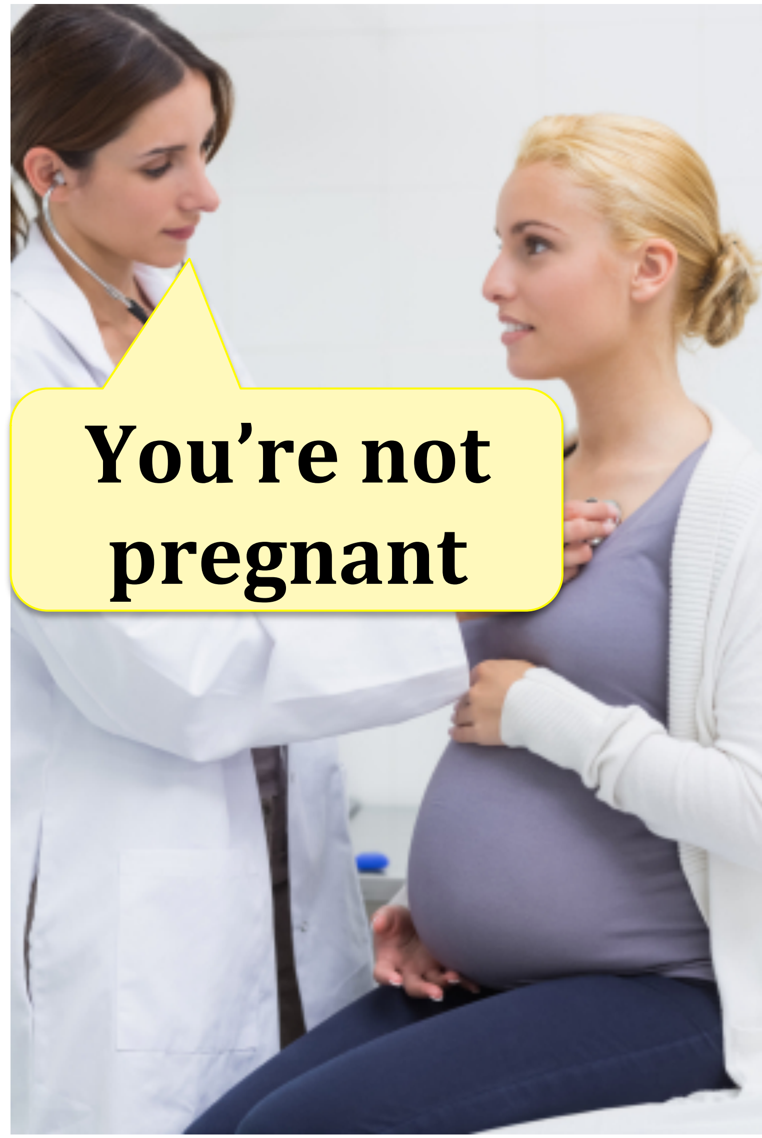
iqr
1.0
0.8
0.6
0.4
0.2
0.0
CAV
1.0
0.8
0.6
0.4
0.2
0.0
ZC
0.0
0.2
0.4
0.6
0.8
1.0


Nina Matthews Photography 2013

Type I Error
(False Positive)



Type II Error
(False Negative)

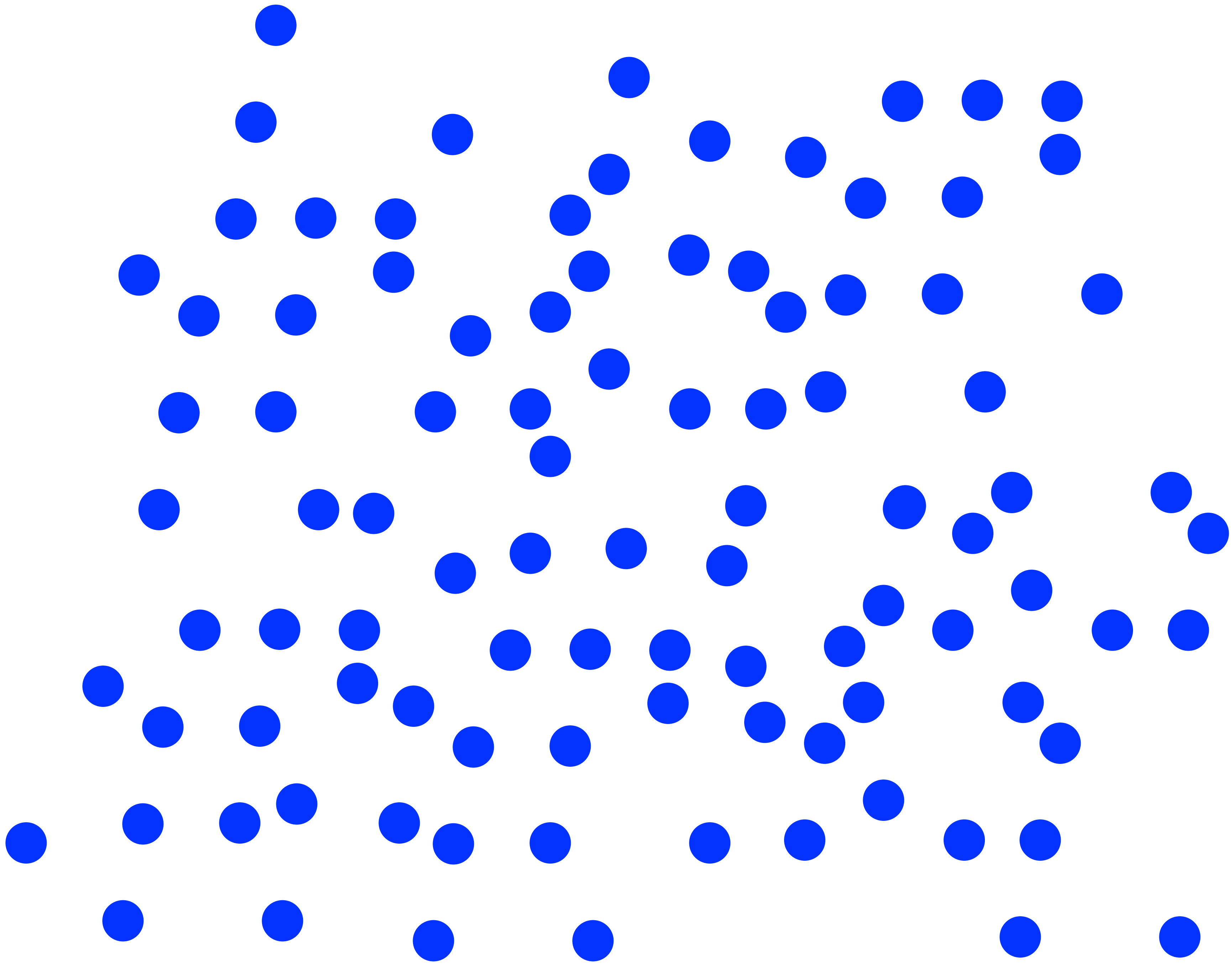


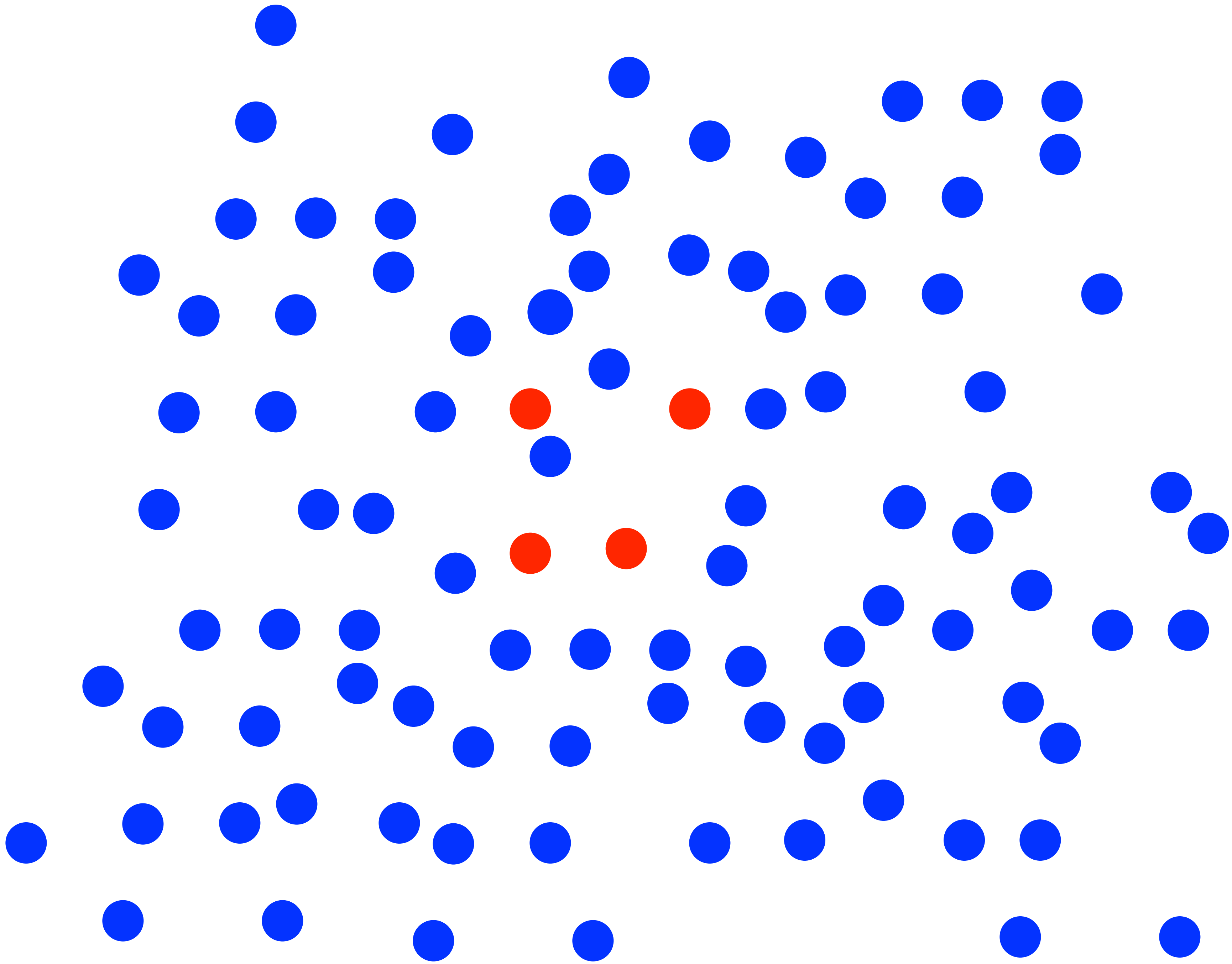
Test performance

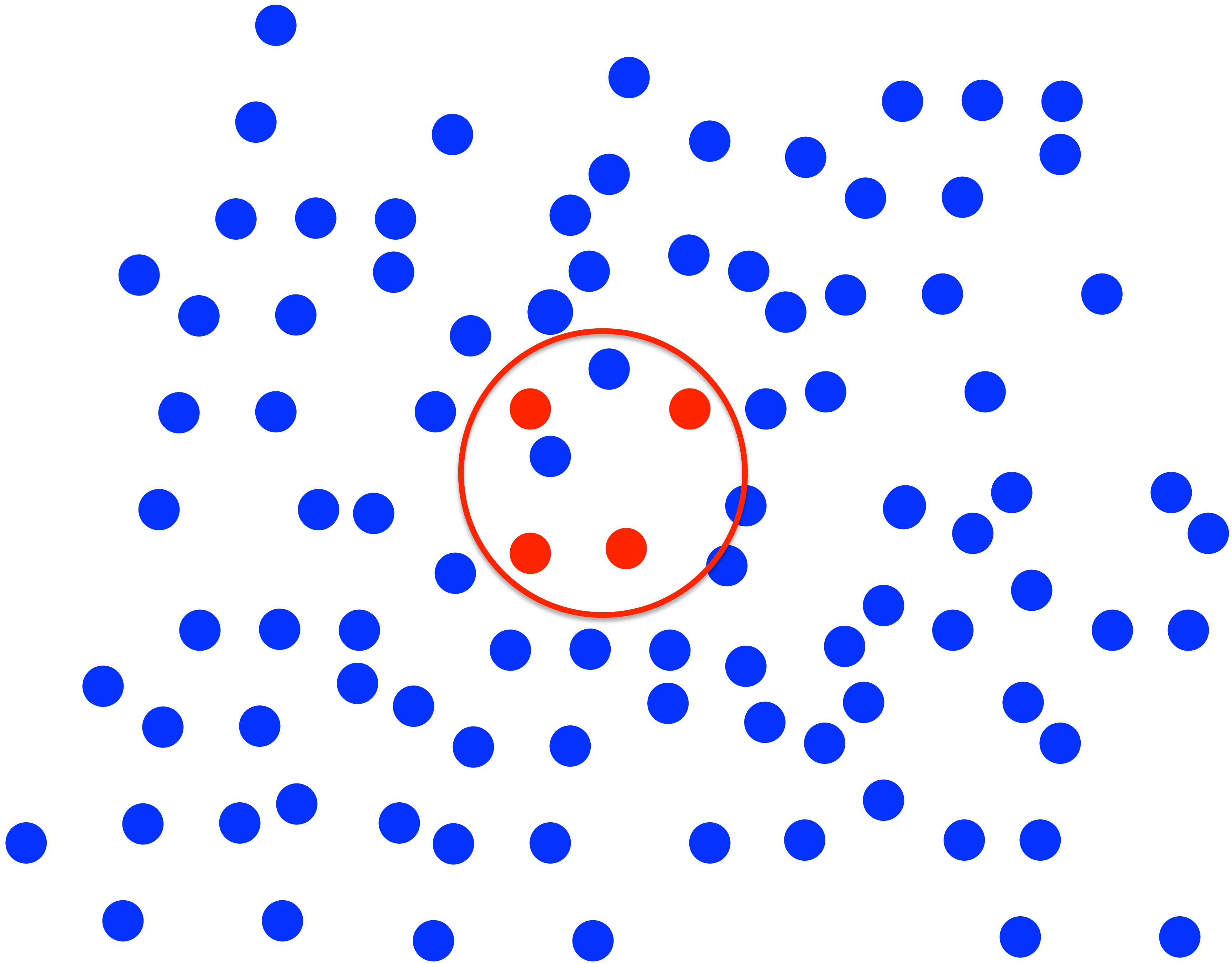
Earthquake	Within 10 km	Within 20 km	Within 30 km
1989 Loma Prieta M7	2/2	4/4	11/11
1994 Northridge M6.7	4/4	19/19	29/29
2004 Parkfield M6	19/20	35/39	36/42
2014 La Habra M5.1	13/13	22/52	30/120
2014 Napa M6	2/2	6/8	10/24
Human triggers	Detect as noise	Detect as EQs	Total trigger
20150201-20150228	3562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

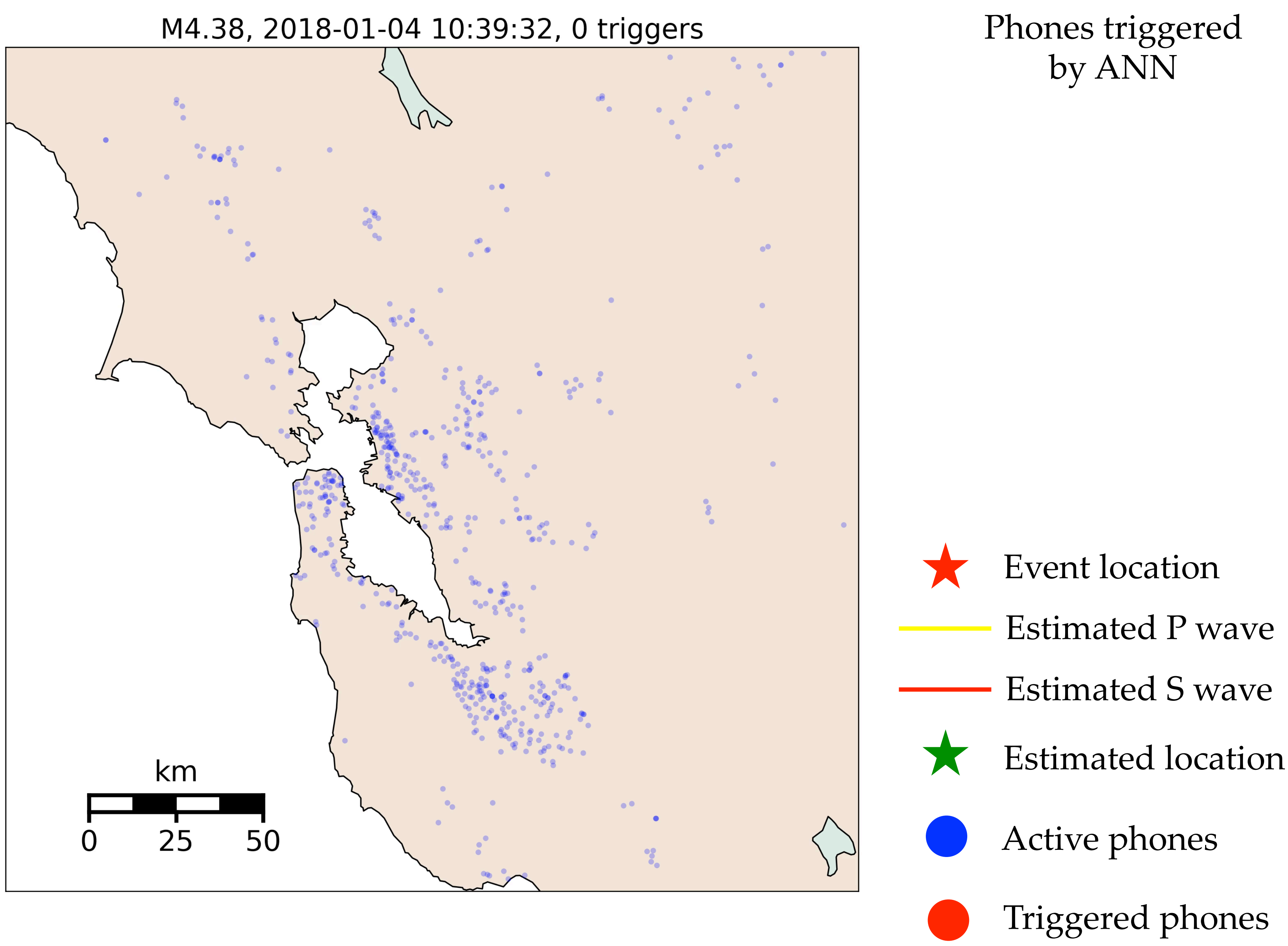
93.2% of the human triggers are correctly classified

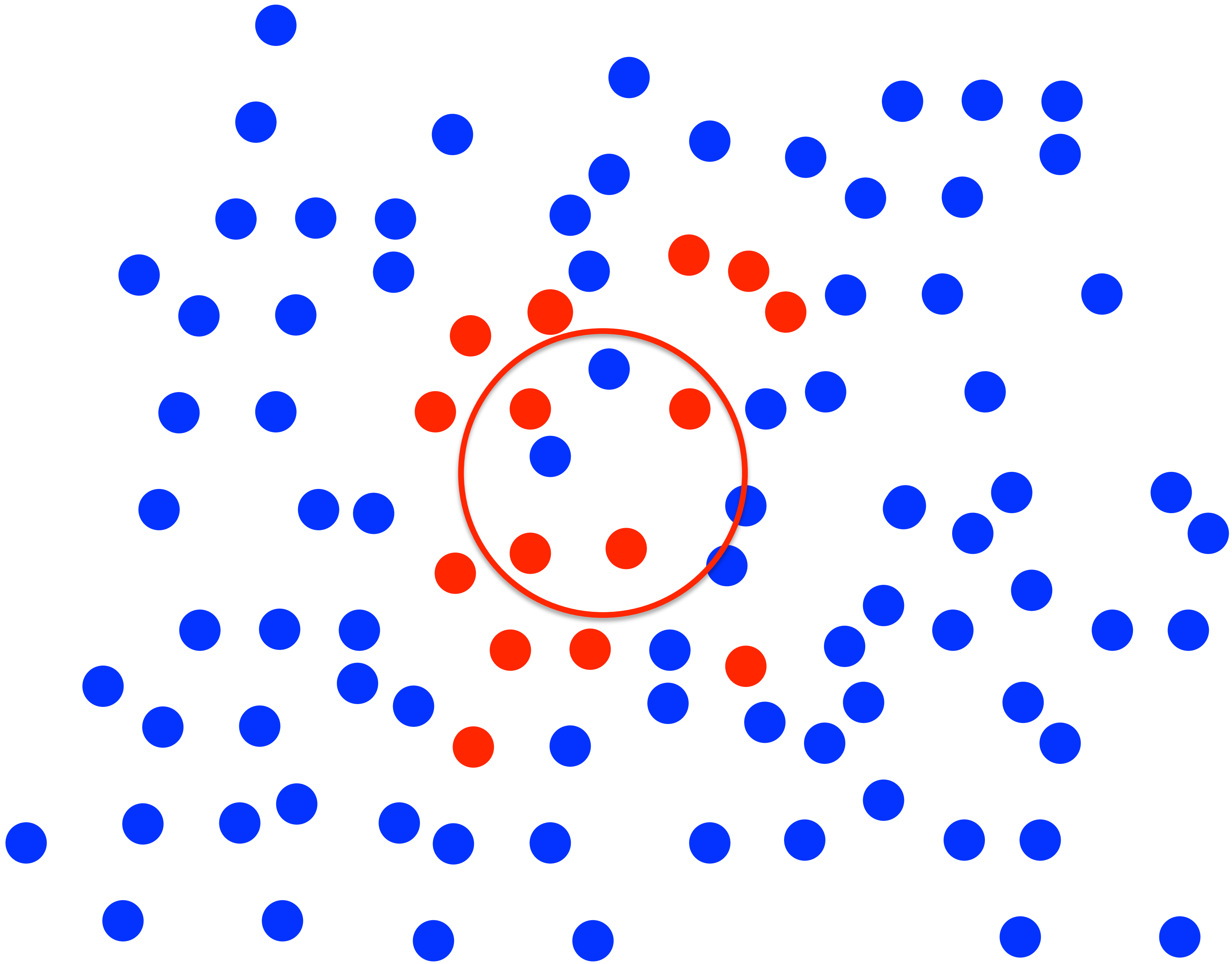
A network of MyShakers

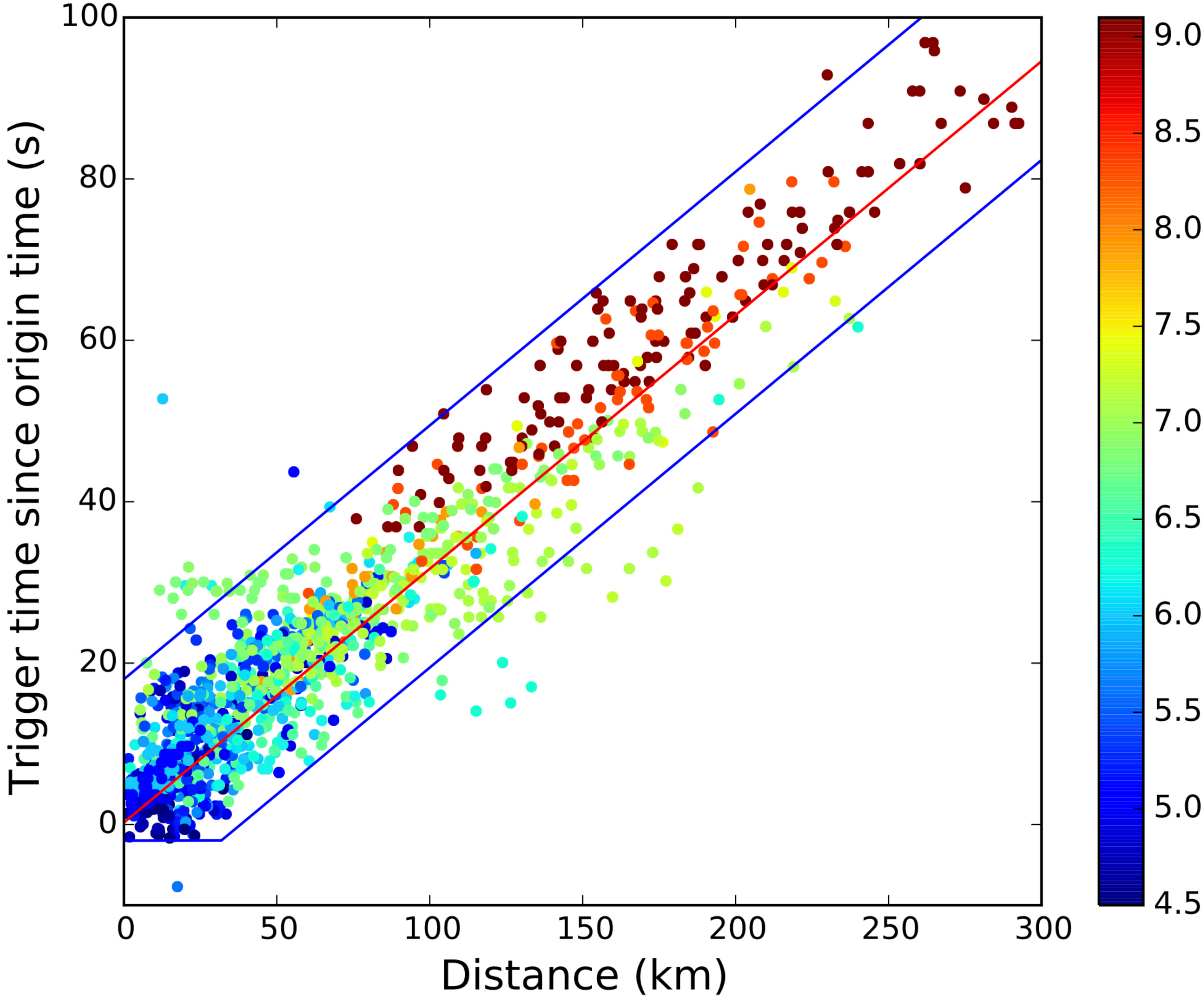




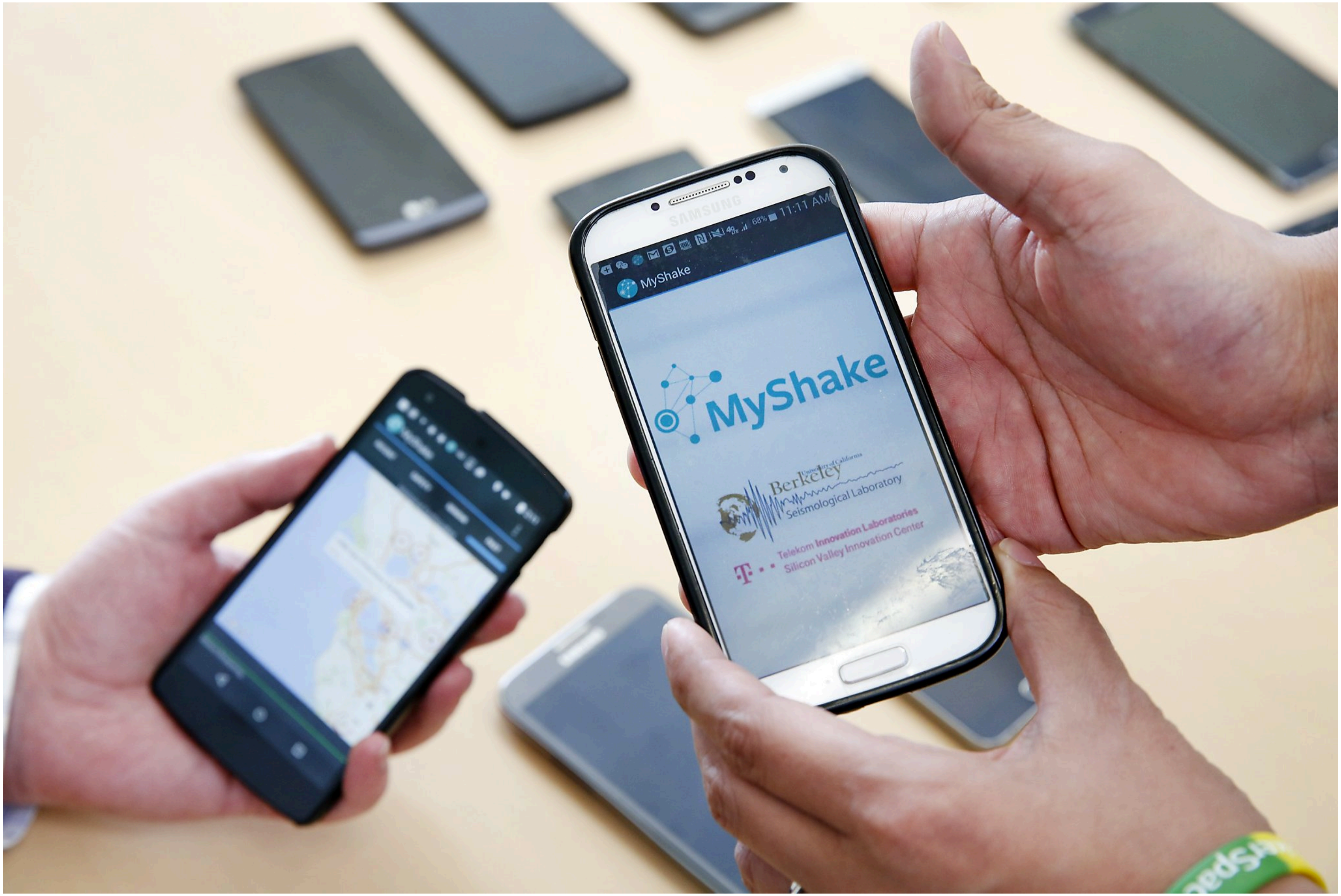



M4.38, 2018-01-04 10:39:32, 0 triggers
km
0
25
50
Phones triggered by ANN
Event location
Estimated P wave
Estimated S wave
Estimated location
Active phones
Triggered phones



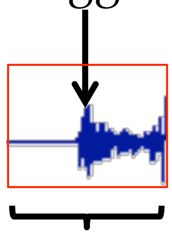


Trigger time since origin time (s)
100
80
60
40
20
0
0
50
100
150
200
250
300
Distance (km)
9.0
8.5
8.0
7.5
7.0
6.5
6.0
5.5
5.0
4.5

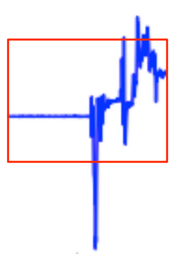

MyShake
MyShake
Berkeley
Seismological Laboratory
Telekom Innovation Laboratories
Silicon Valley Innovation Center
SAMSUNG

In terms of detect EQ

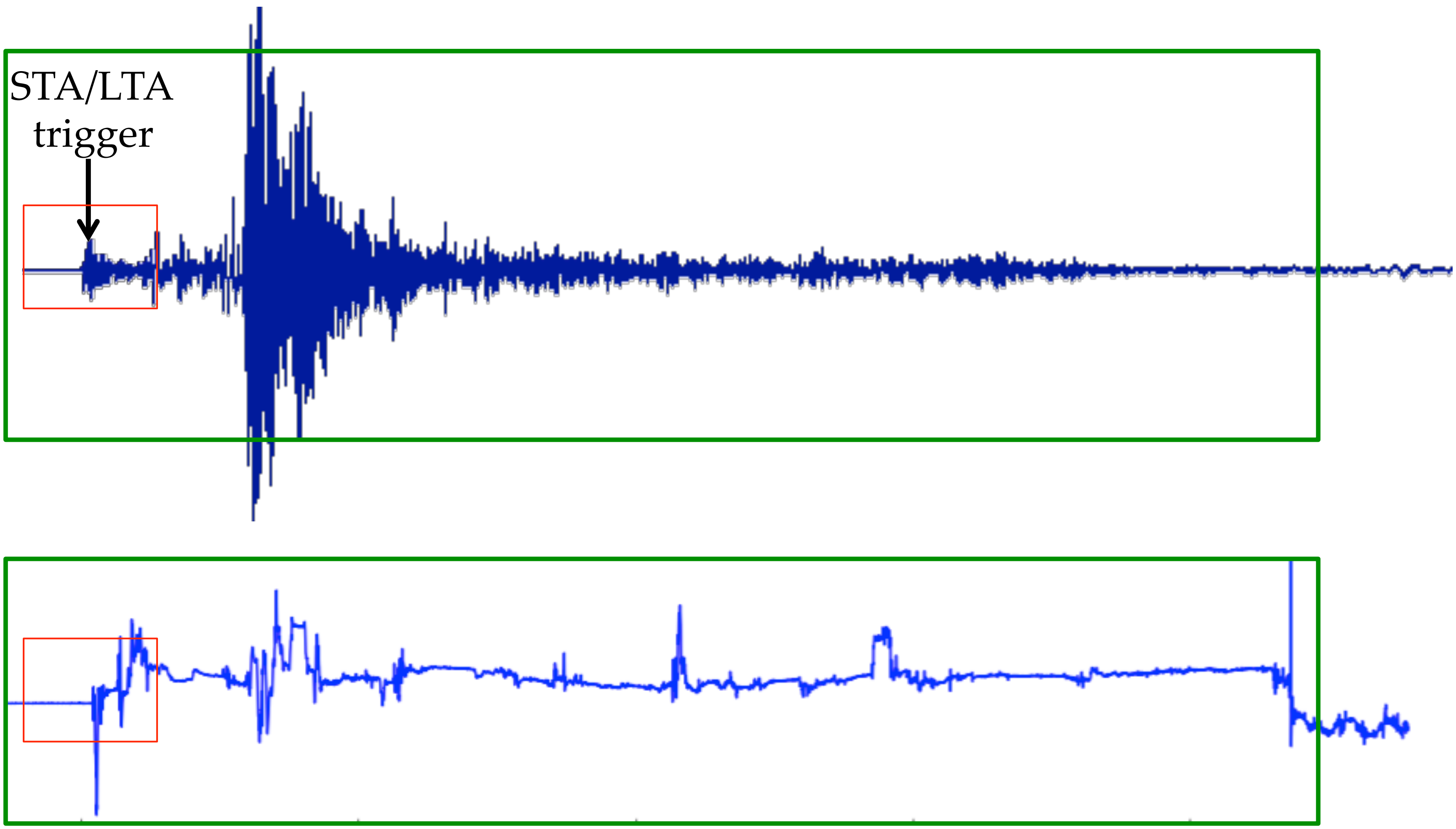
STA/LTA trigger



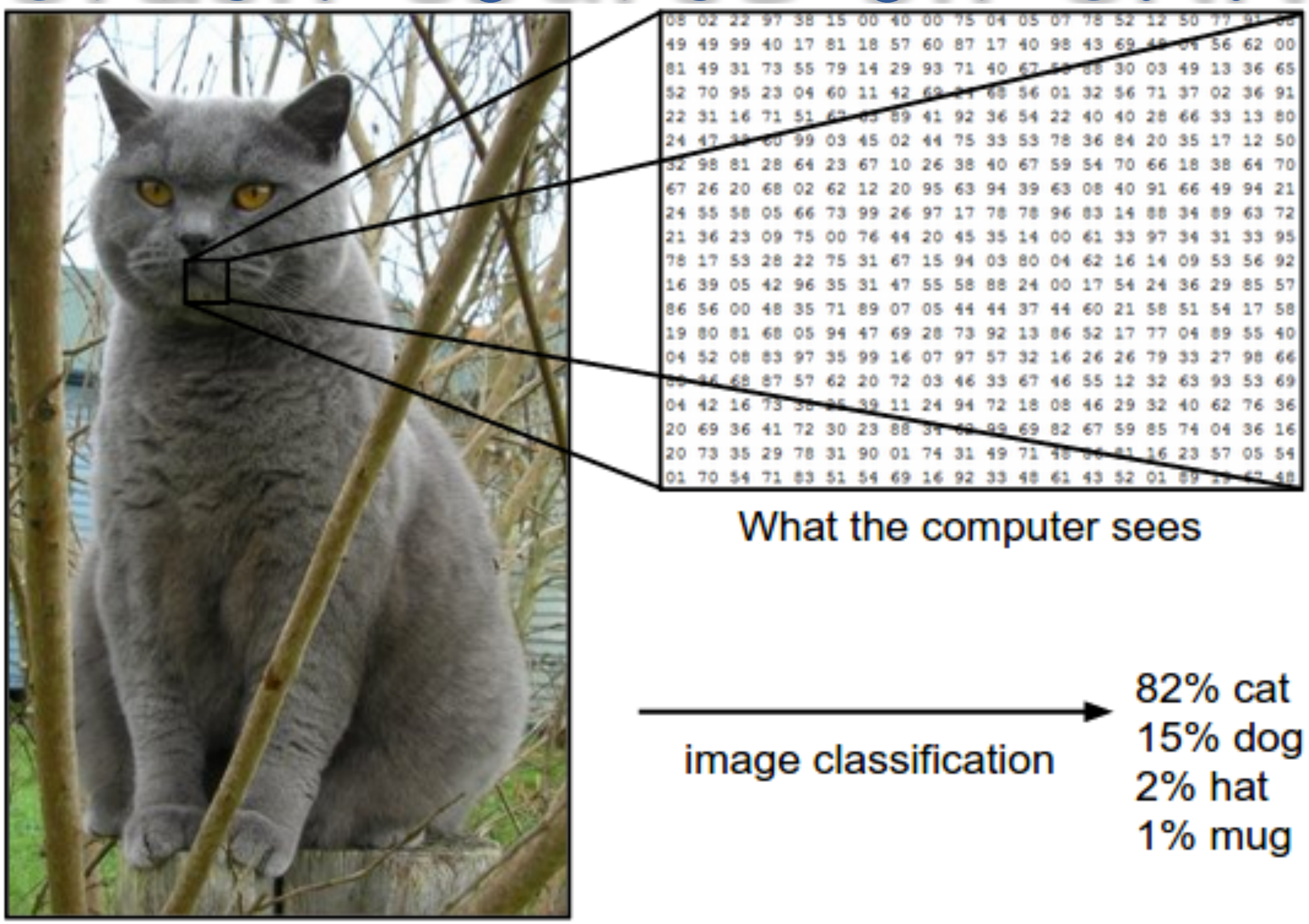
2 sec window with 1 sec overlap



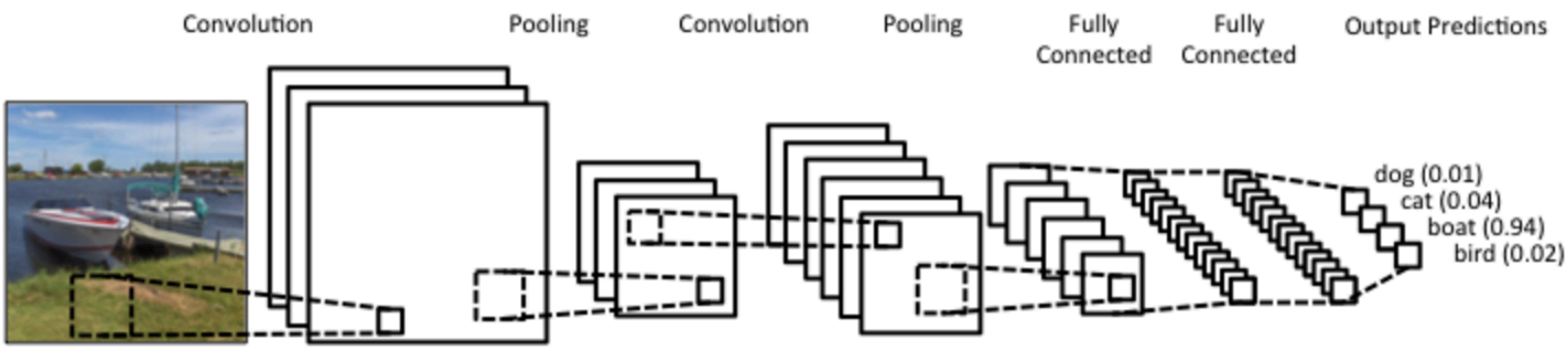
In terms of detect EQ



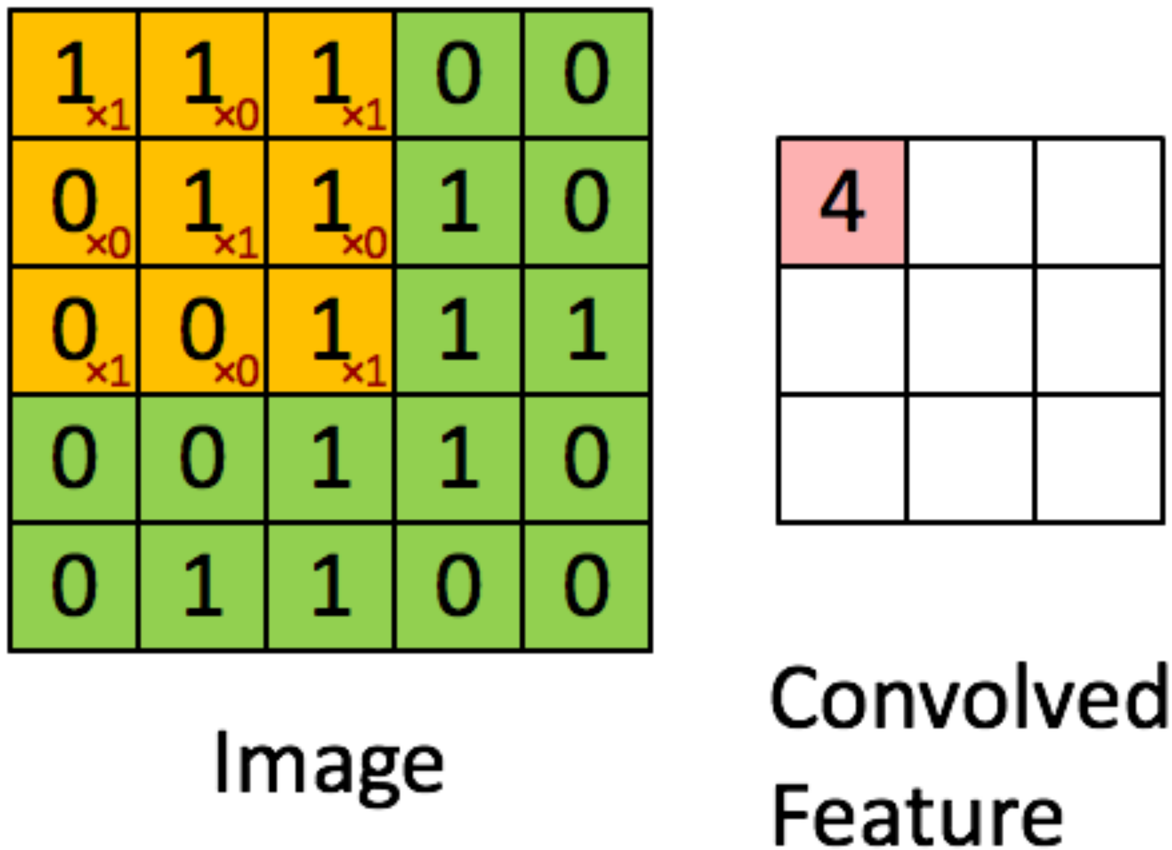
Crash course on CNN

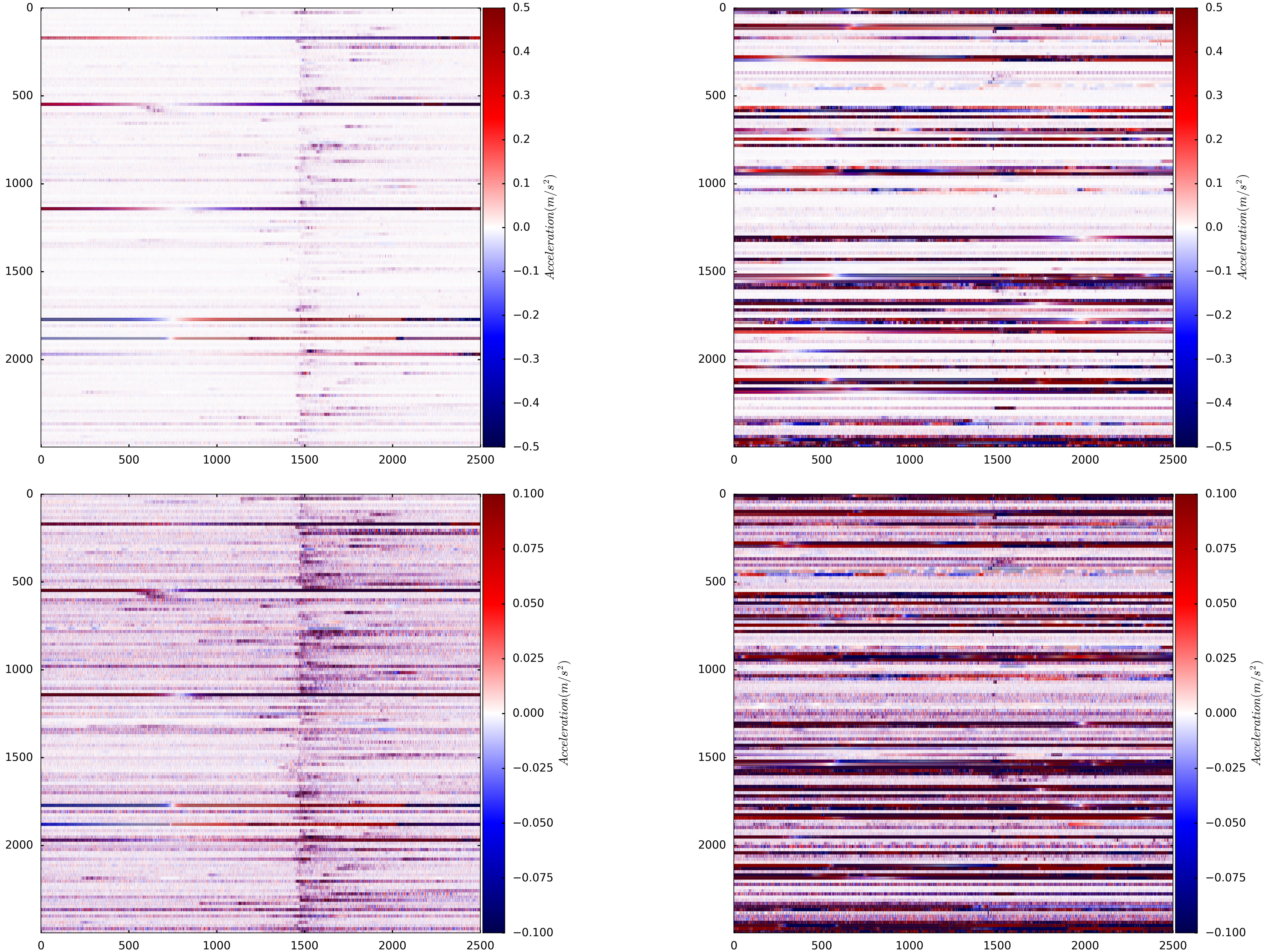


Crash course on CNN



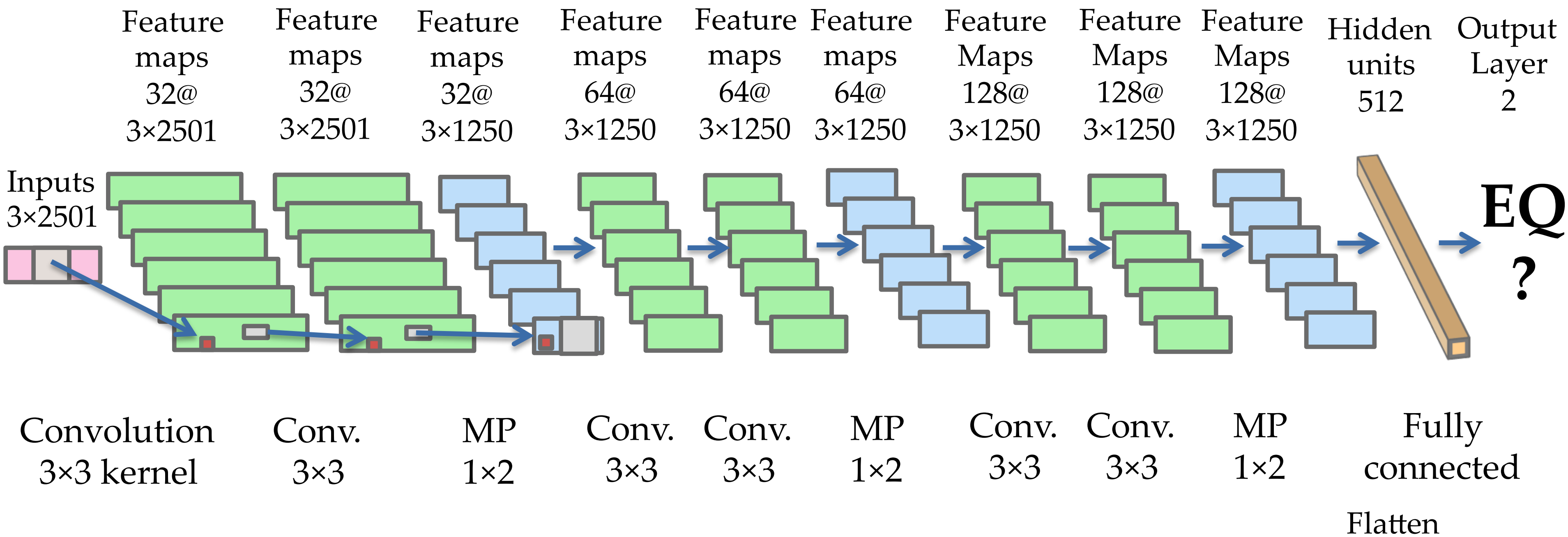
Crash course on CNN





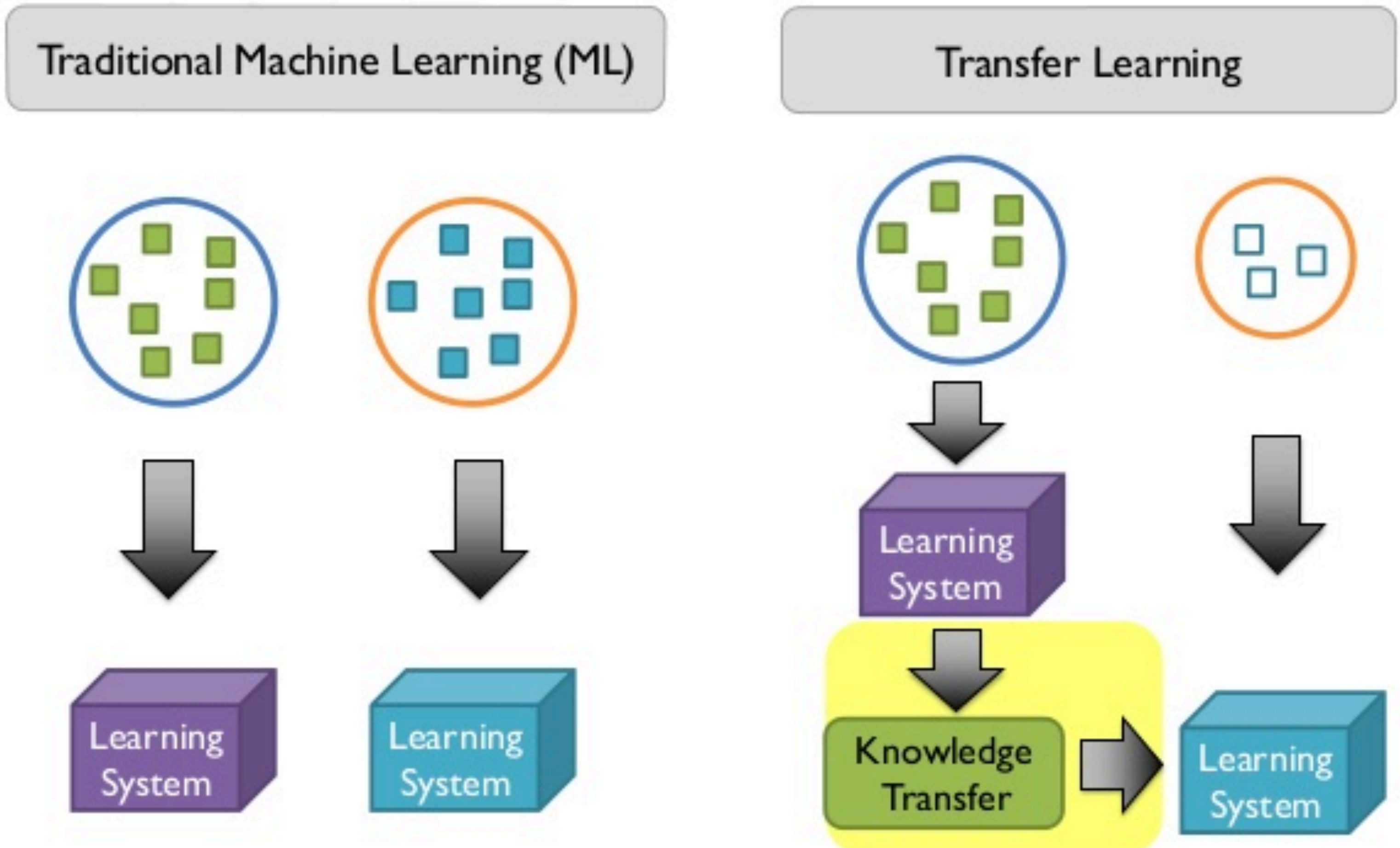
0
500
1000
1500
2000
2500
0.5
0.4
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3
−0.4
−0.5
0.100
0.075
0.050
0.025
0.000
−0.025
−0.050
−0.075
−0.100
$Acceleration(m/s^2)$

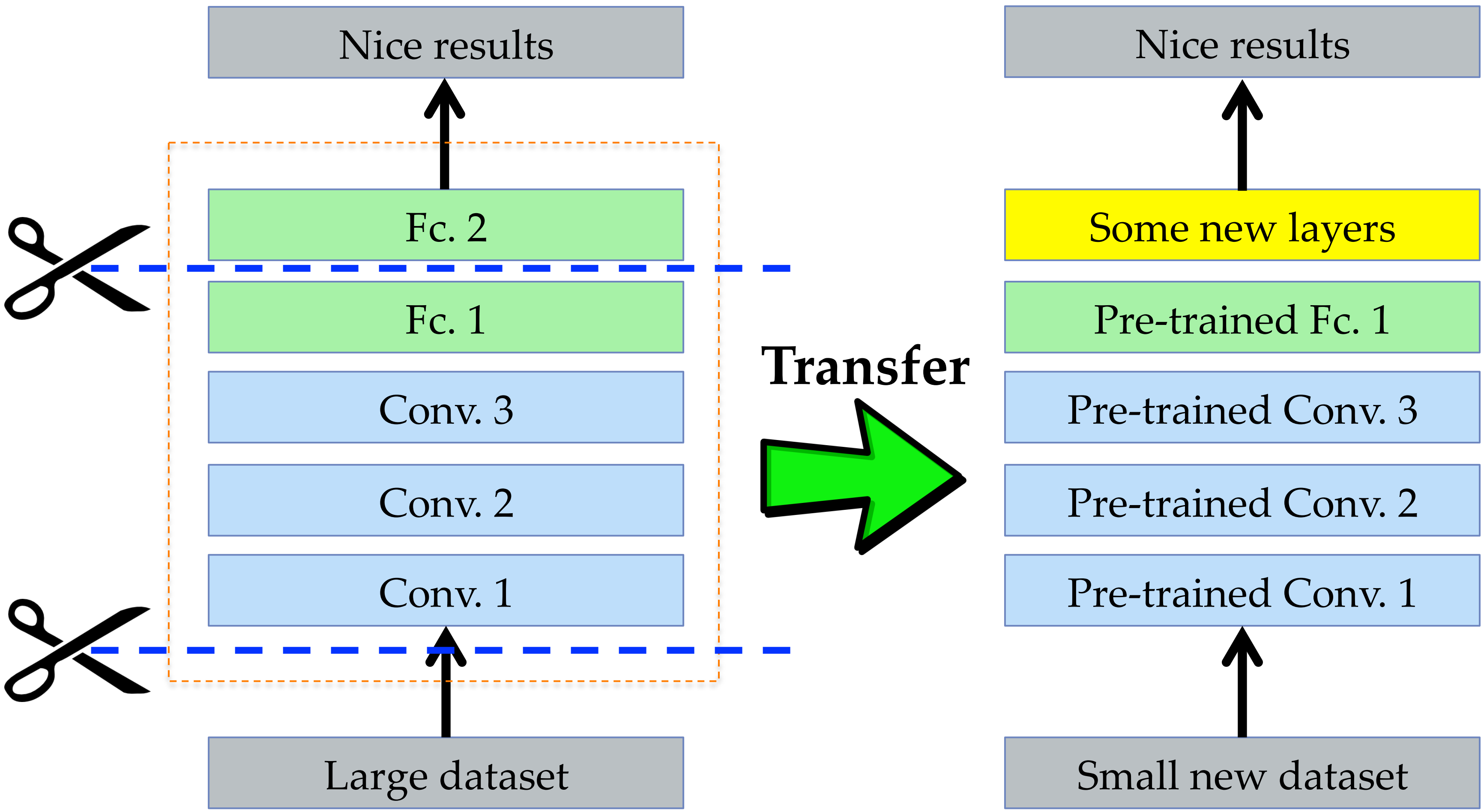
My final model



Test score: 99.84%

Transfer Learning




Nice results
Fc. 2
Fc. 1
Conv. 3
Conv. 2
Conv. 1
Large dataset
Transfer
Nice results
Some new layers
Pre-trained Fc. 1
Pre-trained Conv. 3
Pre-trained Conv. 2
Pre-trained Conv. 1
Small new dataset