Improving Gesture Recognition With Machine Learning: A Comparison of Traditional Machine Learning and Deep Learning.

Reinhard Bacher
Deutsches Elektronen-Synchrotron DESY, Hamburg, Germany

Gestures

Gesture types, e.g.:
- 1D/2D flat gestures including single- and multi-finger gestures,
- 2D/3D spatial gestures including hand and arm gestures,
- 3D head movements including yaw, pitch and roll

User input devices, e.g.:
- Mouse
- Touch-sensitive display
- Leap motion controller
- Myo gesture control armband
- Epson Moverio BT-200 smart glass
- Vuzix M100 smart glass

Myo Gesture Control Armband

Primitive gestures (device-specific), e.g.:
- Mouse: Click, Move
- Touch-sensitive display: Tap, Move / Swipe, Pinch (two fingers)
- Leap motion controller: Key-Tap, Swipe, Open-Hand, Closed-Hand, Circle
- Myo gesture control armband: Double-Tap, Wave-Out / Wave-In, Fingers-Spread, Fist
- Smart glass: Move-Fast / Move-Slow, Roll

Enriched gestures:
- Primitive gestures can be combined with preceding or following linear movements or rotations.
- Each enriched gesture can be performed as a long-or-short and slow-or-fast linear movement or rotation.

Movement Types:
- 32 linear and 8 circular movements
- Diagonal: Upward/Downward - Right/Left - Long/Short - Slow/Fast
- Horizontal: Right/Left - Long/Short - Slow/Fast
- Vertical: Upward/Downward - Long/Short - Slow/Fast
- Circular: Clockwise/Counterclockwise - Long/Short - Slow/Fast

Goal:
Improving the quality and reliability in recognizing enriched gestures based on different finger, hand and head movements using machine learning algorithms

Data Sets

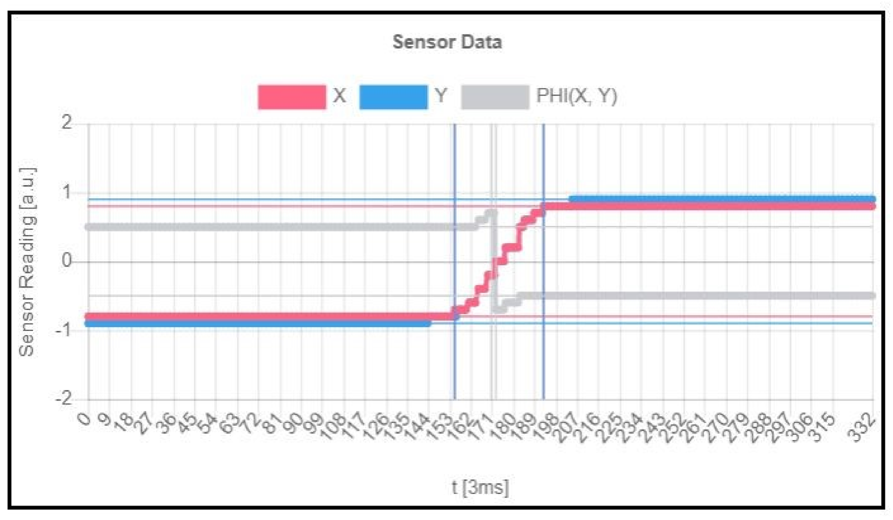
Model:
Generated randomized position data sets simulating movements (i.e. time series of 333 data points at intervals of 3 ms) in cartesian (X, Y) coordinates according to

$$t \le t_{start}: \quad f(t) = s_{start}$$
$$t_{start} < t < t_{end}: \quad f(t) = s_{start} + \left(\left(\frac{s_{end} - s_{start}}{t_{end} - t_{start}} \right) * (t - t_{start}) \right)$$
$$t \ge t_{end}: \quad f(t) = s_{end}$$

Set	t_{jitter}	s_{jitter}	f_{noise}
Set 0	5%	5%	10%
Set 1	2.5%	2.5%	10%
Set 2	7.5%	7.5%	10%
Set 3	5%	5%	5%
Set 4	5%	5%	15%

Legend:
- n_{points}: Number of data points (333)
- t_{jitter}: Jitter of gesture duration
- s_{jitter}: Jitter of gesture length
- f_{noise}: Noise floor of data

Parameter	Value
s_{start} (long signal)	-0.8
s_{end} (long signal)	0.8
s_{start} (short signal)	-0.6
s_{end} (short signal)	0.6
t_{start} (fast signal)	0.35 * n_{points}
t_{end} (fast signal)	0.65 * n_{points}
t_{start} (slow signal)	0.20 * n_{points}
t_{end} (slow signal)	0.80 * n_{points}



Diagonal Upward - Right - Long - Fast Movement

Traditional Machine Learning (TML)

Feature Engineering:
- Set-up a well-engineered mathematical model of the movement (see Data Sets)

Feature Extraction (Nelder-Mead):
- For each coordinate orientation (horizontal, vertical , polar angle):
- Perform a non-linear regression (Nelder-Mead method) and find start / end time and start / end position of movement (features)
- Calculate duration and length of gesture if feature values are confined within reasonable limits

Classification (k-Nearest Neighbor):
- Define gesture orientation (left / right linear movement, up / down linear movement, clockwise / counter-clockwise circular movement) using feature values (t_{start}, t_{end}, s_{start}, s_{end}) found by regression
- Define proper gesture label (Long-Slow, Long-Fast, Short-Slow, Short-Fast) by classifying the gesture duration ($t_{end} - t_{start}$) / gesture length ($s_{end} - s_{start}$) values found using the memorized learned data sets (k-Nearest Neighbor analysis, k = 20).

Long - Fast Movement

Deep Learning (CNN)

Feature Engineering:
- Deep learning does not require any preceding feature engineering or model building.

Feature Extraction and Classification (Convolutional Neural Network):
- Multi-Layer Convolutional Neural Network is well suited to classify time series data
- Input: Series (333 data points) of input sensor data (X, Y)
- Convolution layer filters out the characteristic data features (filter size: 1 x 3)
- Rectified linear activation function (ReLU) provides contrast enhancement
- Pooling halves the size of the resulting data arrays but retains the most important information
- Fully connected layer provides binary classification.
- Softmax activation function delivers a probability distribution
- Output: Prediction (40 classification classes)

Layer (Filters)	Array Size
Input	1 x 333 x 2
Convolution (18)	1 x 331 x 18
ReLU Activation	1 x 331 x 18
Pooling	1 x 165 x 18
Convolution (36)	1 x 163 x 36
ReLU Activation	1 x 163 x 36
Pooling	1 x 81 x 36
Convolution (72)	1 x 79 x 72
ReLU Activation	1 x 79 x 72
Pooling	1 x 39 x 72
Fully Connected	1 x 1 x 1120
ReLU Activation	1 x 1 x 1120
Fully Connected	1 x 1 x 560
ReLU Activation	1 x 1 x 560
Softmax Activation	1 x 1 x 280
Output	1 x 1 x 40

Training and Verification

Training:
- Data sets:
  - t_{jitter} = 5%, s_{jitter} = 5%
  - f_{noise} = 10%
  - Number of movement types simulated: 40
- Number of training steps: 500, 1000, 1500
- Number of verification steps: 100

$n_{training}$	Algorithm	Mean	Min	Max
500	TML	0.936±0.017	0.50	1.00
	CNN	0.997±0.002	0.93	1.00
1000	TML	0.945±0.015	0.58	1.00
	CNN	1.000±0.000	1.00	1.00
1500	TML	0.949±0.015	0.61	1.00
	CNN	1.000±0.000	0.99	1.00

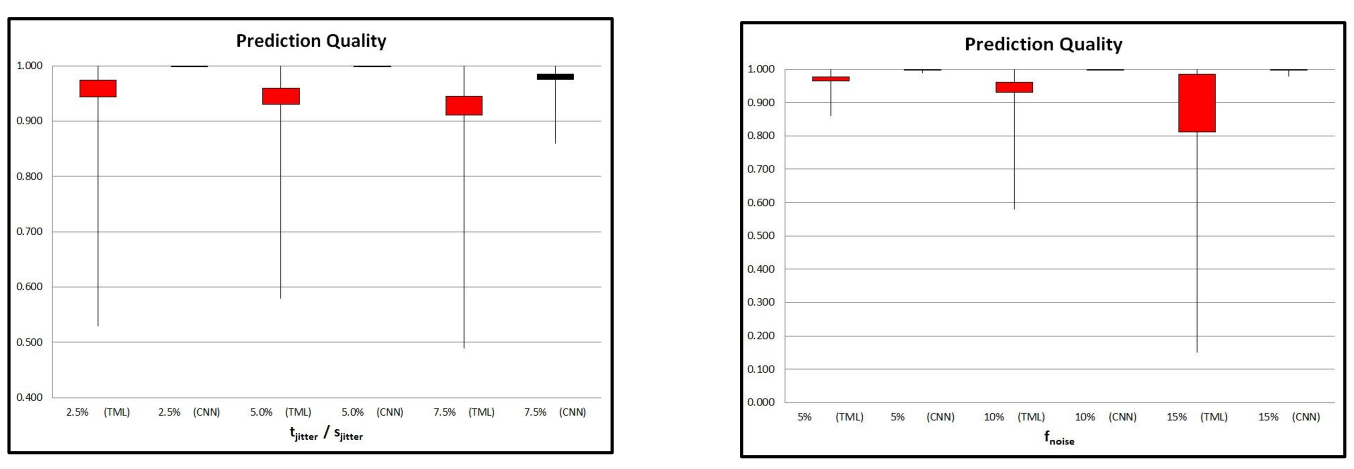
Legend:
- $n_{training}$: Number of training steps
- Mean: Mean training quality
- Min: Worst out of 40 movement types
- Max: Best out of 40 movement types

Performance:
- *Training period:* Due to the inherent complexity of the deep learning algorithm a training step of the convolutional neural network requires considerably more time to be performed as in the case of the traditional machine learning algorithm.
- *Training quality:* The deep learning approach performs better than the traditional machine learning algorithm.

Prediction

Performance:
- *Sensitivity:*
  - The deep learning algorithm is less sensitive to deviations from the standard movement data set (Set 0)
  - The prediction power of the traditional machine learning algorithm degrades rapidly with increasing noise floor
- *Quality:* The deep learning algorithm provides in all tested cases a higher prediction quality

Test	Algorithm	Mean	Min	Max
Set 0	TML	0.945±0.015	0.58	1.00
	CNN	1.000±0.000	1.00	1.00
Set 1	TML	0.959±0.015	0.53	1.00
	CNN	1.000±0.000	1.0	1.0
Set 2	TML	0.928±0.017	0.49	1.0
	CNN	0.980±0.005	0.86	1.0
Set 3	TML	0.971±0.007	0.86	1.0
	CNN	1.000±0.000	0.99	1.0
Set 4	TML	0.849±0.037	0.15	1.0
	CNN	1.000±0.000	0.98	1.0



Prediction Quality ($n_{training}$ = 1000, n_{test} = 100)