

Gait Event Detection Using an LSTM Network

10-701 Project Presentation

Pablo Iturralde
Yin Zhong
Jakob Bauer

April 22, 2015

Introduction

- ▶ Goal: accurately detect gait events (heel strike, toe off) in video-based motion capture data of human walking gait
- ▶ Necessary to analysis of changes in gait that arise from training, disease, aging, etc.

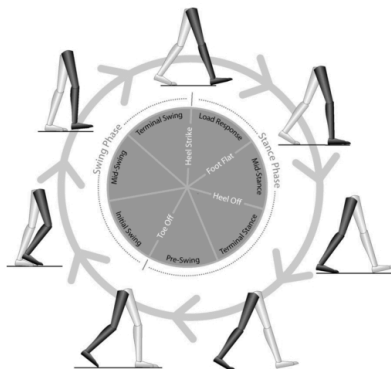


Figure 1: Gait events [Rueterbories et al., 2010]

Data

- ▶ All data are time series
- ▶ **Input:** 3D locus of 18 motion capture markers ($54 \times N$ reals)
- ▶ **Output:**
 - ▶ Raw: Event takes place $\Rightarrow 1$; else $\Rightarrow 0$ ($4 \times N$ bools)
 - ▶ Equivalent: Leg in stance phase $\Rightarrow 1$; else $\Rightarrow 0$ ($2 \times N$ bools)
- ▶ Training dataset:
 - ▶ Sample rate: 100 Hz
 - ▶ 240 000 samples ($8 \text{ subjects} \times 3 \text{ trials} \times 10\,000 \text{ samples}$)
 - ▶ Ground truth from force plates on treadmills (very accurate)

Baseline Methods

- ▶ Signal processing approach [O'Connor et al., 2007]
 - ▶ Heuristic based on position and speed of heel and toe markers
 - ▶ **Pros:**
 - ▶ No training is needed
 - ▶ Good accuracy when input data is clean
 - ▶ **Cons:**
 - ▶ Sensitive to noise in input; resulting in gross mis-predictions
 - ▶ Need to manually specify sensitive thresholds
 - ▶ Heavy pre-filtering helps reduce noise but reduces accuracy
- ▶ Feed-forward Neural Network [Miller, 2009]
 - ▶ Sliding window centered around the desired marker
 - ▶ **Pros:**
 - ▶ TODO
 - ▶ **Cons:**
 - ▶ Heavy preprocessing (dimensionality reduction, windowing, etc.); requires manual picking of parameters
 - ▶ TODO

Our Approach: LSTM

- ▶ Motivation:
 - ▶ Avoid manual picking of sensitive parameters (window size, threshold, filter cutoff, etc.)
 - ▶ Human walking can be modeled as a dynamic system; RNN (Recurrent Neural Network) learns dynamic systems
 - ▶ Any gait cycle may depend on ones preceding it
- ▶ LSTM cell: RNN building block for variable time-dependence
- ▶ Network architecture:
 - ▶ Inputs (54 reals)
 - ▶ n LSTM cells (n reals in $[-1, +1]$)
 - ▶ Output layer (softmax/sigmoid)
 - ▶ Outputs (2 reals in $[0, 1]$)
- ▶ Implementation:
 - ▶ Torch/Lua
 - ▶ LSTM cell by de Freitas (Oxford University, Google Deepmind)
 - ▶ AWS EC2 GPU instance (g2.2xlarge)

Results

Thank you for your attention!