

# LSTM을 활용한 연속보행에서 보행 단계 추정

---

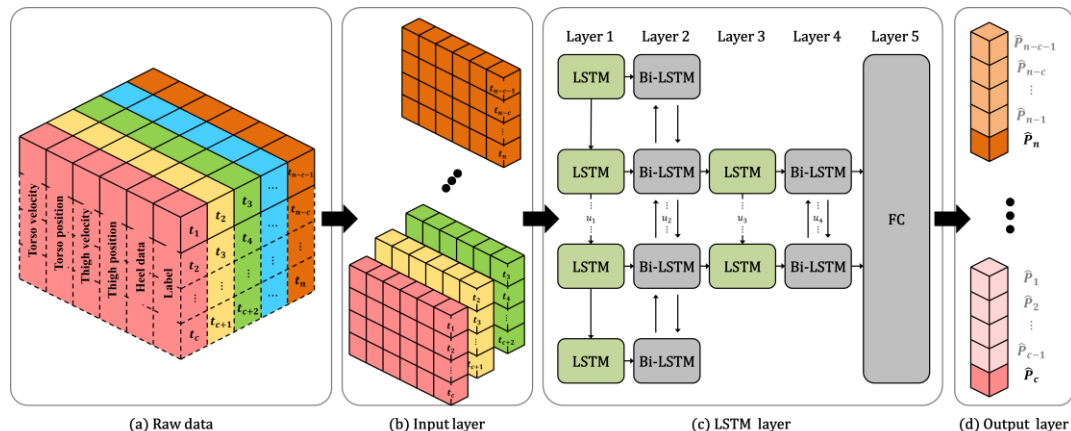
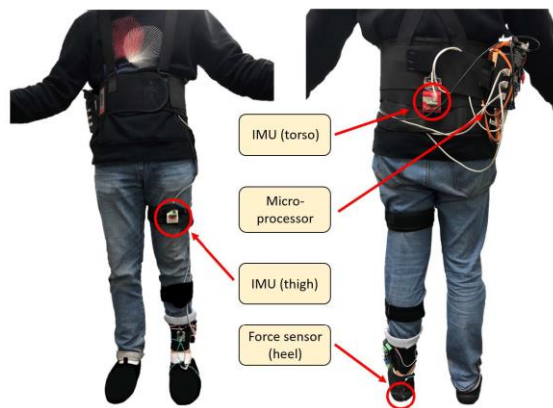
2022. 05. 27.

이진원

고려대학교 기계공학부

# Gait phase estimation with LSTM

- 목적: **LSTM**(Long-Short Term Memory)을 활용하여 보행 단계 추정
- 보행 데이터 수집
  - 3개의 센서(Torso & Thigh IMU, Heel force sensor)
  - 3명의 성인 남성
- 딥러닝 네트워크 구축
  - 입력: Torso position, Torso velocity, Thigh position, Thigh velocity, (Heel force sensor)
    - S1: 2개의 IMU, S2: 2개의 IMU + force sensor
  - 네트워크: 5개의 레이어(LSTM-BiLSTM-LSTM-BiLSTM-FC)
  - 출력: 보행 단계 (sine & cosine label → linear label)



# Gait phase estimation with LSTM

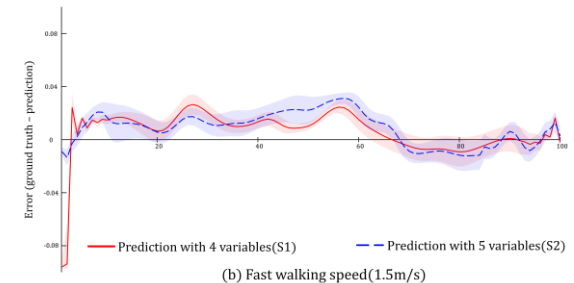
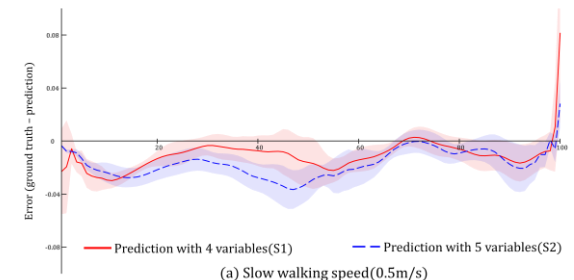
## 결과

- Force sensor를 사용한 S2가 S1보다 6배 예측결과가 좋음
- 두 모델 모두 mid-stance(40%–60%)에서 결과가 좋지 못함

## 한계점 및 향후 연구

- Subject의 숫자가 적음 → **공공 데이터를 활용**하여 신뢰도 있는 많은 데이터 사용
- Mid-stance(Toe-off 부근)에서 정확도가 떨어짐 → **보행 단계를 비선형**으로 표현 시도
- 소형 보드에서 실시간 처리할 수 없음 → 더 적은 연산으로 처리 가능한 **경량 네트워크 모델** 필요

Walking speed	4 variables (S1)	5 variables (S2)
0.5 m/s	18.75 ± 10.68 ms	3.75 ± 3.11 ms
1.0 m/s	17.08 ± 10.75 ms	2.08 ± 2.57 ms
1.5 m/s	20.83 ± 13.29 ms	5.00 ± 3.02 ms
2.0 m/s	22.50 ± 8.12 ms	3.75 ± 3.11 ms

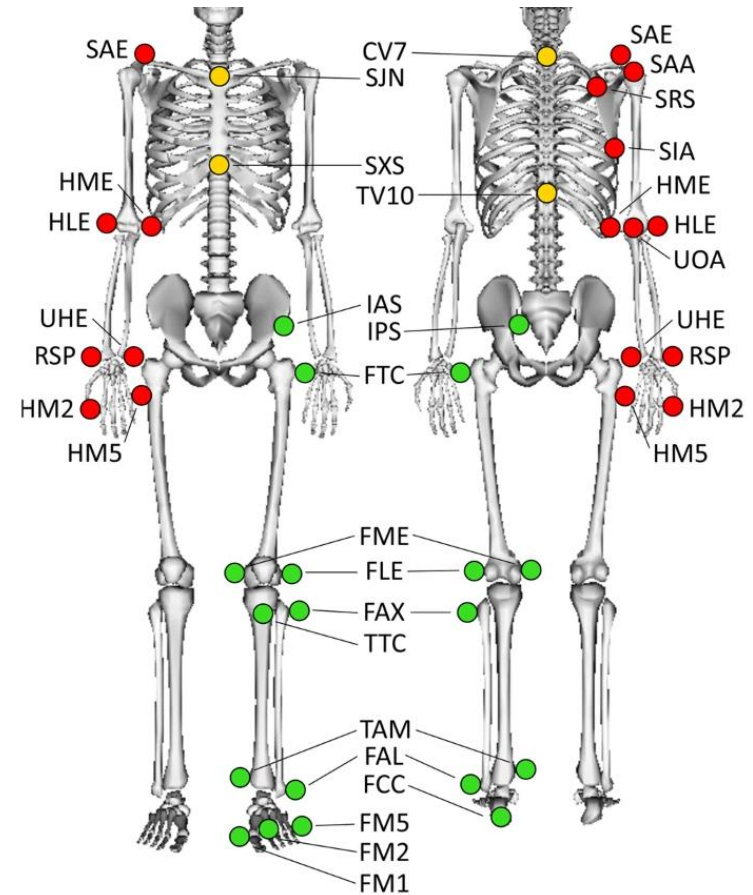


# 보행 단계에 대한 공공 데이터

Schreiber, C., & Moissenet, F. (2019)

데이터 정보

- 50명의 subject
- Marker data(52 points) → (torso & thigh) position, velocity
- Force data(2 force platform)
- Gait event(Left/Right HS&TO) → gait phase labeling
- Speed: C1(0~0.4m/s), C2(0.4~0.8m/s), C3(0.8~1.2m/s), C4(self-selected spontaneous), C5(self-selected fast speed)
- 1개의 데이터 시퀀스마다 1개의 걸음걸이 데이터



# Effect of Torso Kinematics on Gait Phase

- 목적: Torso velocity, position은 보행 단계 예측에 도움이 되는가?
- 보행 데이터 수집
  - Schreiber, C., & Moissenet, F 의 공공 데이터셋을 활용
  - Marker data → 허벅지와 몸통부의 마커 정보를 활용하여 position, velocity 계산
  - 50명의 건강한 성인(42명 학습 데이터, 8명 테스트 데이터)
- 딥러닝 네트워크 구축
  - 입력: Torso position, Torso velocity, Thigh position, Thigh velocity
    - Set1: Thigh position, Thigh velocity, Torso position, Torso velocity
    - Set2: Thigh position, Thigh velocity, Torso velocity
    - Set3: Thigh position, Thigh velocity, Torso position
    - Set4: Thigh position, Thigh velocity
  - 네트워크: 5개의 레이어(LSTM-BiLSTM-LSTM-BiLSTM-FC)
  - 출력: 보행 단계 (sine & cosine label → linear label)



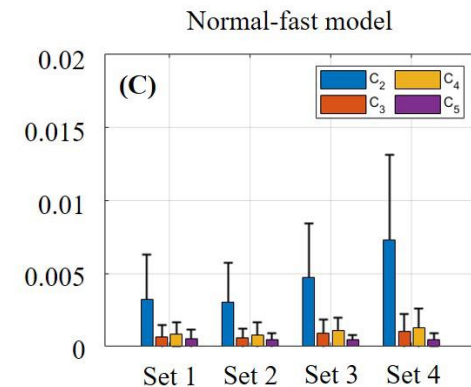
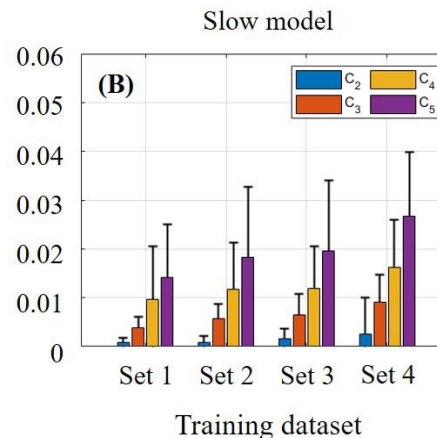
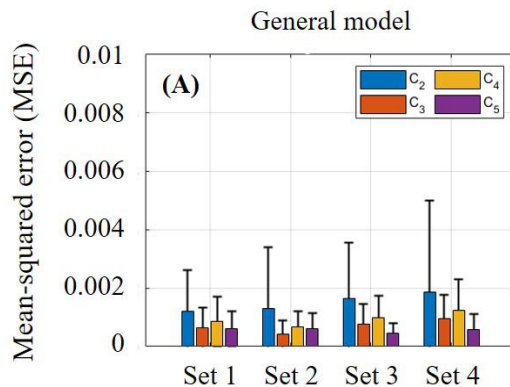
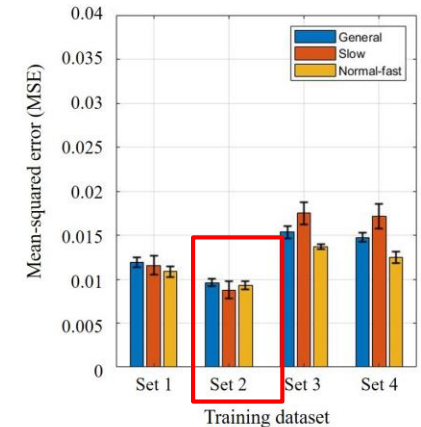
# Effect of Torso Kinematics on Gait Phase

- 결과

- Set 2가 가장 좋음(-Torso position)
- 느린 속도일수록 예측 정확도가 떨어짐

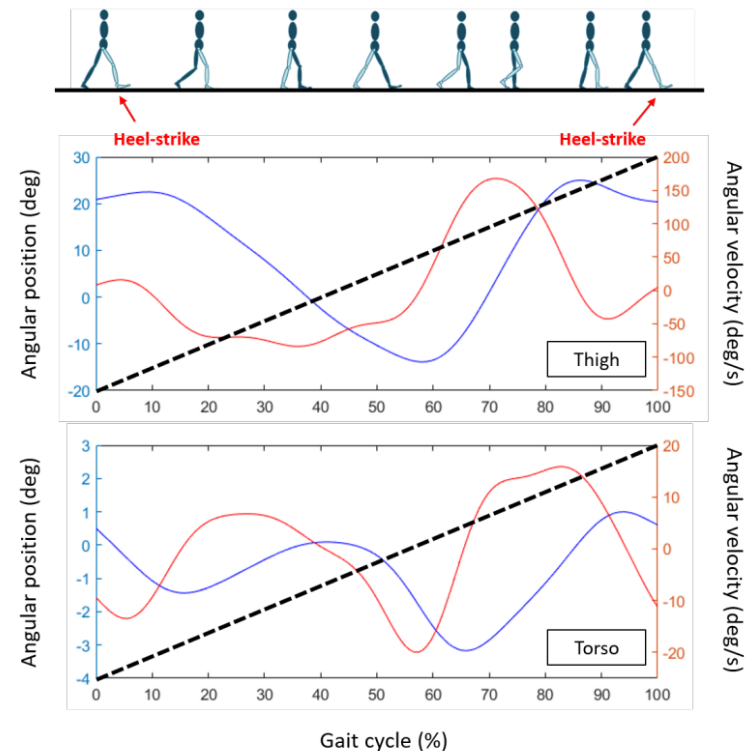
- 한계점 및 향후 연구

- 1개의 시퀀스 데이터에 1개의 걸음걸이 데이터만 있고, 연속 걸음 데이터 부재
- 정확도와 성능을 높일 수 있는 딥러닝 네트워크 필요
- Torso의 velocity만 얻기 위해 Torso IMU 부착이 필요한가?



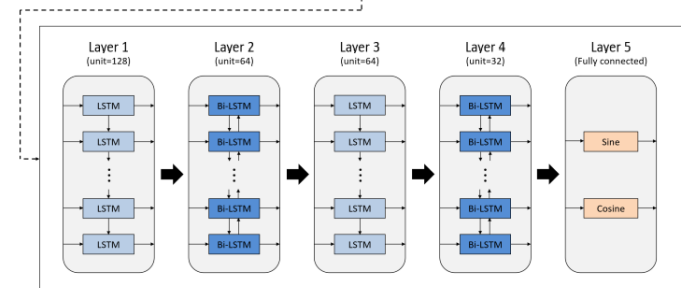
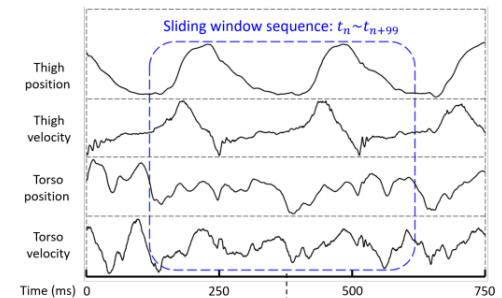
# 보행단계를 비선형 표현

- 보행 단계에 따라 속도가 변화하는데 선형으로 표현이 적절한가?
- 선형으로 표현하지 않는다면 적절한 표현 방식은?
  - 의족 제어에서 중요한 포인트: Heel-strike, Toe-off
  - Heel-strike ~ Toe-off: 0 ~ 50%
  - Toe-off ~ Heel strike: 50% ~ 100%



# Piecewise Linear Labeling

- 목적: 보행 단계의 레이블링 방식은 **선형이 적절한가? 부분선형이 적절한가?**
- 레이블링 방식
  - Linear(기준): heel-strike ~ heel-strike (0~100%)
  - Piecewise linear(제안): heel-strike ~ toe-off(0~50%) / toe-off ~ heel-strike(50~100%)
- 딥러닝 네트워크 구축 (이전과 동일)
  - 입력: Torso position, Torso velocity, Thigh position, Thigh velocity
  - 네트워크: 5개의 레이어(LSTM-BiLSTM-LSTM-BiLSTM-FC)
  - 출력: 보행 단계 (sine & cosine label → linear label)





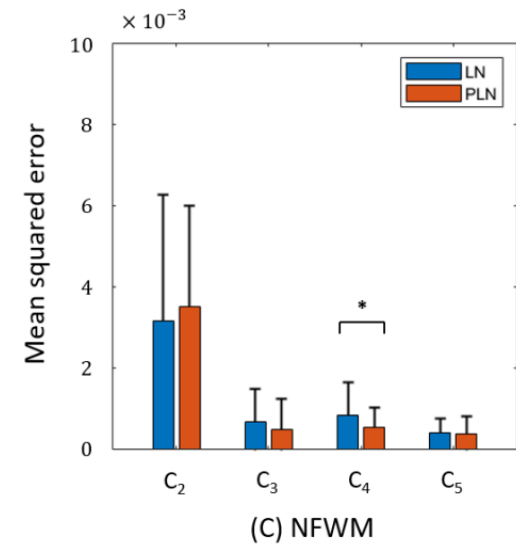
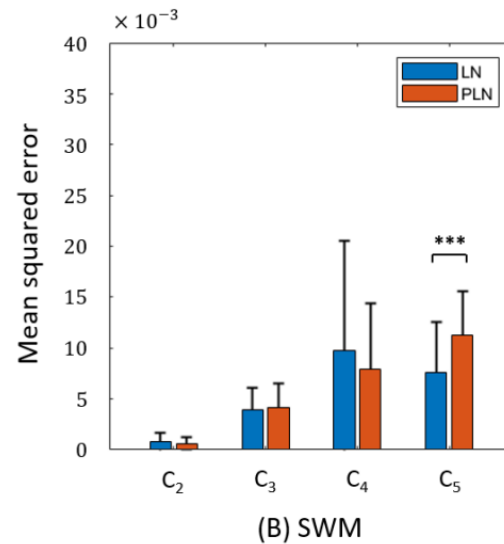
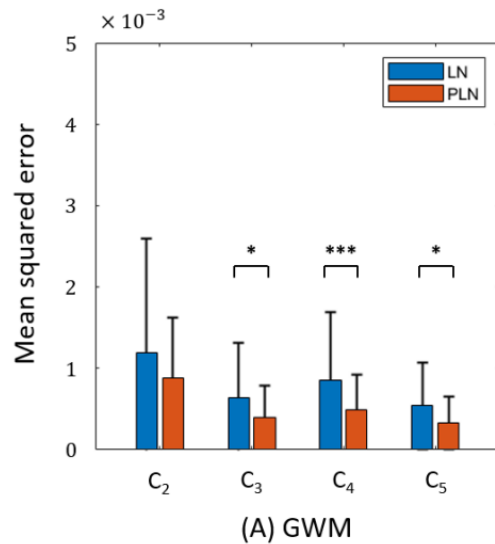
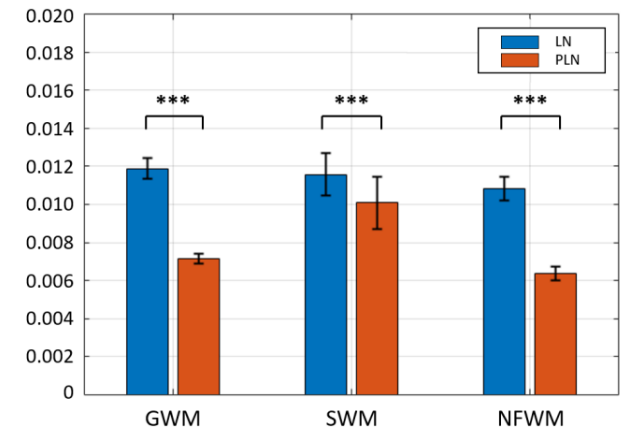
# Piecewise Linear Labeling

## 결과

- 일반적인 속도에서 Piecewise linear가 더 좋은 정확도를 보임
- 느린 속도에서는 비슷하거나 근소하게 **PLN이 좋음**

## 한계점

- 연속걸음이 포함되지 않은 공공 데이터셋
- 느린 걸음 데이터에 대한 실효성

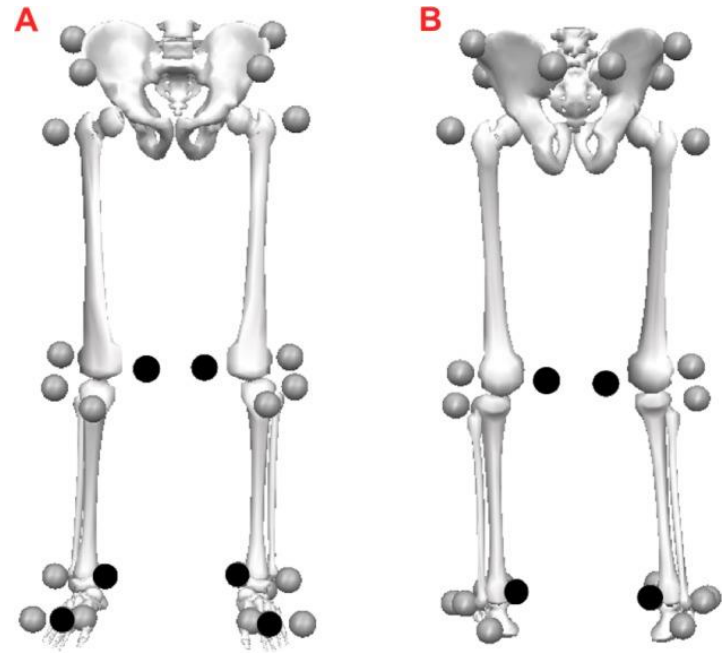


# 연속 걸음이 포함된 새로운 공공데이터

Fukuchi, C. A., Fukuchi, R. K., & Duarte, M. (2018).

데이터 정보

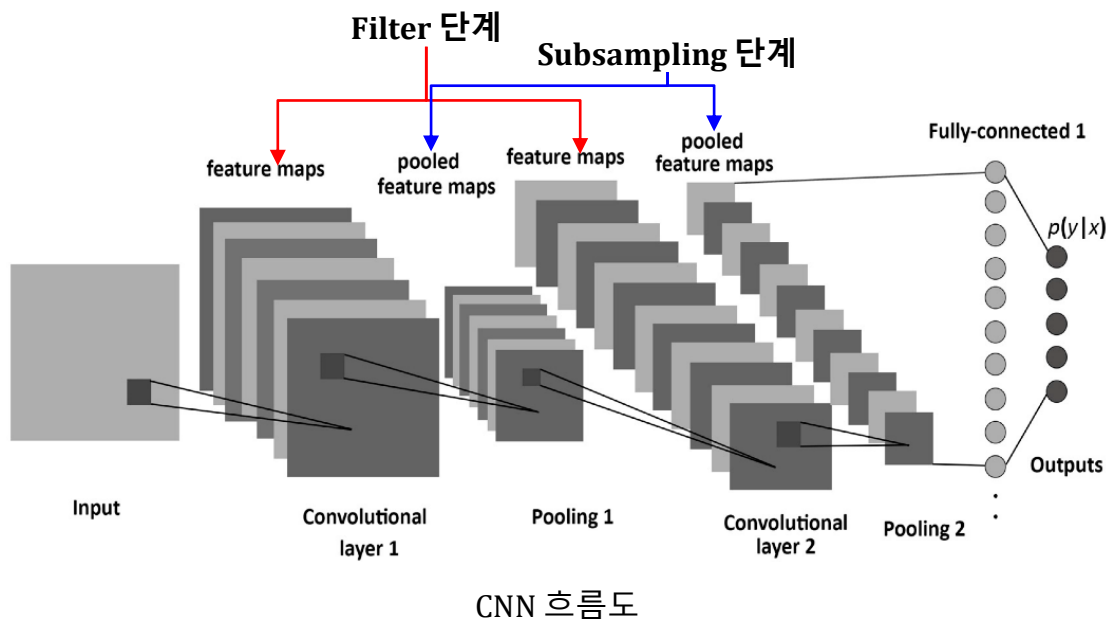
- 42명의 subject
- Marker data(22 points) → **thigh** position, velocity
- Force data(2 force platform)
- Gait event(Left/Right HS&TO) → gait phase labeling
- Speed: T1~T8(0.5, 0.68, 0.87, 1.05, 1.24, 1.43, 1.62\*, 1.8\* m/s)
- 1개의 데이터 시퀀스마다 **200개의 걸음걸이** 데이터



# 고성능 경량 네트워크 모델 (CNN+LSTM)

- Convolutional Neural Network (CNN)

- 특징을 추출하기 위한 단계 (Filter mask, Convolution)
  - 인공지능이 필터 값을 조절하면서 학습
- 추출된 특징 중에서 의미 있는 값을 강화 단계 (Subsampling)
  - 최대값, 평균값을 활용하여 데이터 크기 감소
- 정리된 특징을 클래스에 매칭하는 단계 (Fully Connection)
  - 결과값 (loss)과 객체의 값을 비교



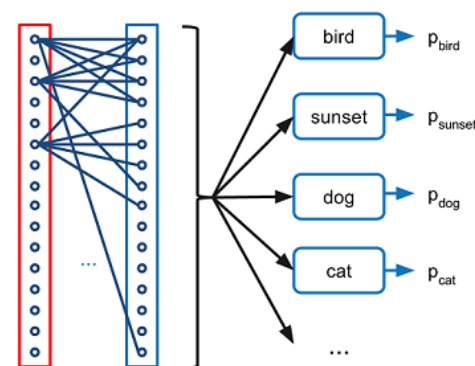
Filter mask example

12	20	30	0
8	12	2	0
34	70	37	4
112	100	25	12

$2 \times 2$  Max-Pool

20	30
112	37

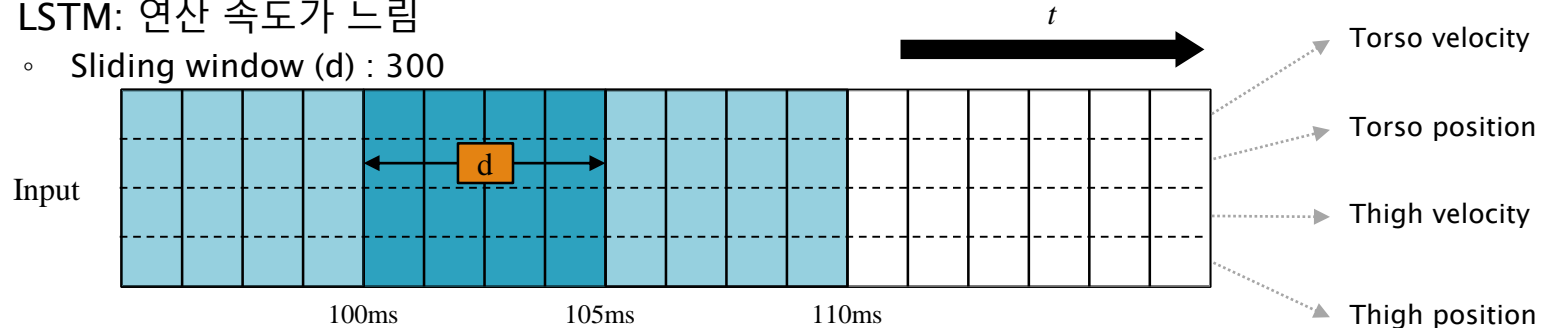
Subsampling example



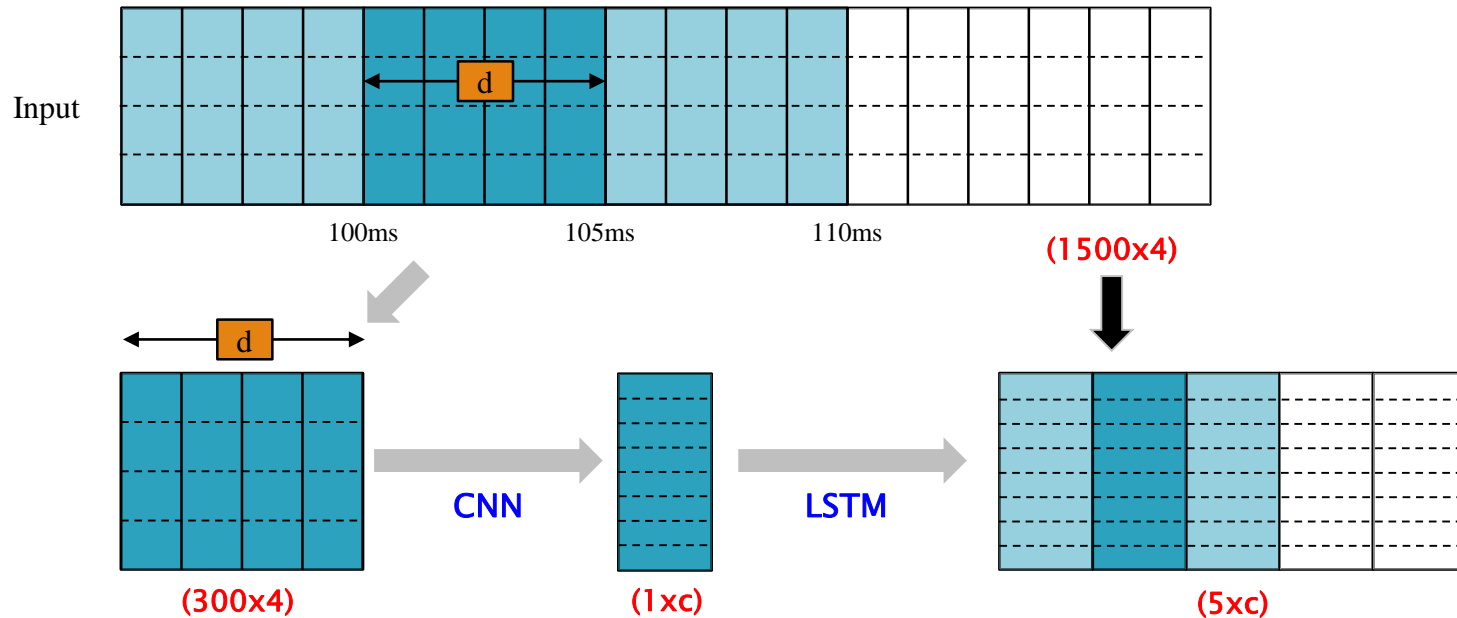
Fully Connection example

# CNN + LSTM

- LSTM: 연산 속도가 느림
  - Sliding window (d) : 300



- CNN(중요한 특징을 학습) + LSTM(전후 특징을 학습)



# LSTM vs CNN+LSTM 성능 비교

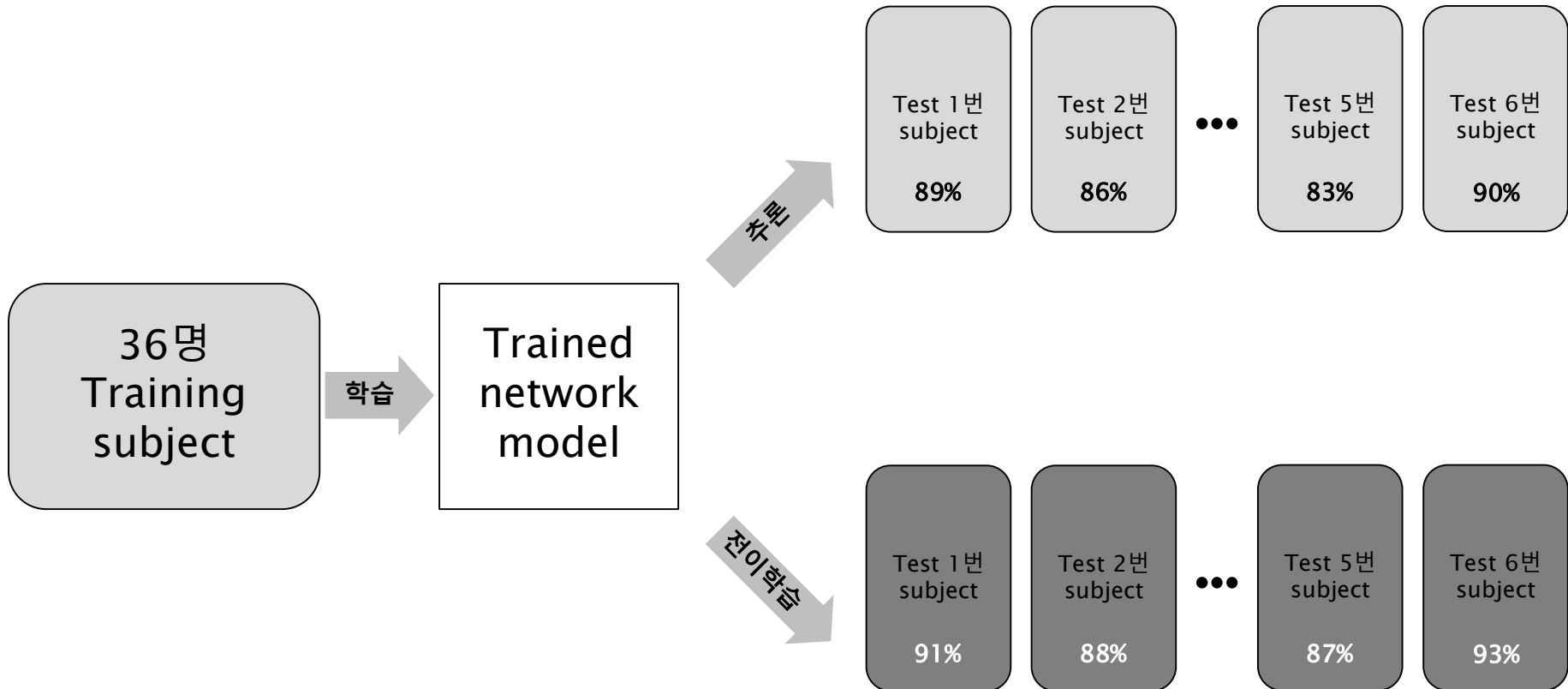
- Gait phase estimation with LSTM dataset (IEEE TNSRE)

	처리시간	파라미터 수	Loss value
LSTM	0.043s (23Hz)	958,722	$4.91 \times 10^{-3}$
CNN + LSTM	0.00310s(320Hz)	85,410	$3.71 \times 10^{-3}$

- 처리시간: 1개 데이터 (300개의 sliding window기준)
- 파라미터 수: 연산시간에 영향
- Loss value (MSE): 성능 지표
- CNN+LSTM 네트워크 구조
  - (300, 5) - CNN(1×1) - (300,64) - BN - CNN - (300,64) - pooling(10) - CNN - (30,64) - pooling (10) - (3, 64) - BN - CNN - (1, 64) - LSTM - (16) - Dropout(0.2) - FC - (2)

# (CNN+LSTM) → Transfer learning

- 연속 걸음 데이터셋 (42명)
  - 학습: 36명 / 테스트: 6명
  - 테스트 데이터에 대해 개별 전이 학습을 통해 커스터마이징하여 성능 비교



**감사합니다!**

---