Predicting Knee Joint Contact Forces using a validated LSTM

This guide provides instructions on how to use the [long-short term memory (LSTM)](https://www.mathworks.com/help/deeplearning/ug/long-short-term-memory-networks.html) neural network to predict medial and lateral joint contact forces from four kinematic variables.

As a note, the LSTM network was built using MATLAB version R2020 (Natick, Massachusetts: The MathWorks Inc.) and [the Deep Learning Toolbox](https://www.mathworks.com/products/deep-learning.html). While previous MATLAB builds can support a variety of neural networks, some functions may not be present in previous builds (e.g., optimizer function, randomizers, etc.).

The output of the LSTM network will be a 2 x 101 matrix: medial (row 1) and lateral (row 2) joint contact forces (in body weights) time normalized to the stance phase (1:101 data points).

Instructions

*Software Requirements*

MATLAB

[Deep Learning Toolbox](https://www.mathworks.com/products/deep-learning.html) (to run [*predict*](https://www.mathworks.com/help/deeplearning/ref/seriesnetwork.predict.html) function)

*Input Variable Preparation*

To use this network, the user will need to import waveforms of the kinematic variables listed below.

1. Hip flexion/extension angle
2. Hip adduction/abduction angle
3. Knee adduction/abduction angle
4. Ankle dorsiflexion/plantarflexion angle

Each variable should be calculated according to the right-hand rule and with anatomical position as the “neutral”, or zero, angle. Each variable will need to be time-normalized to the stance phase. The four kinematic variables should now be combined into a 4 (vars) x 101 (time points) matrix.

Next, the kinematic variables will need to be transformed into Z-scores using the following equation:

where mu and std are the mean and standard deviation derived from the original dataset and X is the kinematic variable.

The two matrices below provide the mu and std data for each kinematic variable, in the order listed above.

The final form of the input variables should now be a matrix that is 4 (vars) x 101 (time points), consisting of time normalized (% stance phase) z-scores of the original kinematic waveforms.

*Execution of LSTM*

After completing the preparation phase of the input variables, the kinematic variables can now be input to the LSTM to predict the medial and lateral contact forces.

First, make sure you have imported the LSTM network into your workspace.

Next, using the [predict function in MATLAB](https://www.mathworks.com/help/deeplearning/ref/seriesnetwork.predict.html), designate the LSTM network and the input matrix. A simple example is shown below.

Variables

*Input\_Matrix* (4x101 double) – contains the z-scores of all four kinematic variables (see list above) time normalized to the stance phase (1:101 data points).

*KJCF\_LSTM* – the provided long-short term memory network (a five-layer neural network)

*Output\_Matrix* (2x101 double) – contains the medial (row 1) and lateral (row 2) joint contact forces (in body weights) time normalized to the stance phase (1:101 data points).

Execution

*Output\_Matrix* = predict (KJCF\_LSTM, Input\_Matrix)

\*\*Please cite the corresponding article any time this network is used\*\*