

---

```

%{
This function takes in the arrayPerActivity struct containing a
single table for each activity.
We now want to label this data through one-hot-encoding so will need a
complementary matrix with labels. We also want to group all of the
data
into a single array after labelling it.

Arguments
- `arrayPerActivity` -> struct containing a single array per
                        activity without any magnetometer data.

Returns:
- `labelledData`      -> array containing all samples for all
                        activities labelled by one-hot encoding
%}

function [labelledData] = getLabelledData(arrayPerActivity)

    % array containing the names of the activities.
    % These names will match the field names in the struct
    sets = ["LGW", "RA", "RD", "SiS", "StS"];

    % -----
    % Label the data by adding an extra 5 columns at the end of
    % each activity array. Only the column corresponding to the
    % activity consists of 1's - the rest are all 0's.
    % -----
    for ff = 1 : length(sets)
        sample = arrayPerActivity(1).(sets{ff});
        temp_labels = zeros(length(sample), 5);
        temp_true_labels = ones(length(sample), 1);
        % horizontally concatenate 5 columns for the labels.
        labelled_activity = horzcat(sample, temp_labels);
        % currently all labels are 0/False
        % depending on the activity, we want to make one column 1/True
        labelled_activity(:, end-(5-ff)) = temp_true_labels;
        labelled_array_per_activity(1).(sets{ff}) = labelled_activity;
    end

    % -----
    % Vertically concatenate the 5 tables into one long table
    % -----
    for ff = 1 : length(sets)
        sample = labelled_array_per_activity(1).(sets{ff});
        if ff == 1
            labelledData = sample;
        else
            labelledData = vertcat(labelledData, sample);
        end
    end
end

```

---

---

```
% -----  
% Split data into inputs and targets for ML  
% -----  
% Define which features to include in the input set.  
final_inputs_nn = labelledData(:,1:end-5)'; % Take all the  
rows, and all the 294 features as inputs.  
  
% Define the target set  
final_targets_nn = labelledData(:, end-4:end)'; % Take all the  
rows, and the last 5 columns as outputs.  
  
end
```

*Published with MATLAB® R2020a*