initTrainDS

Initiates needed data from training dataset to GPR model struct. Builds GPR target vectors depending on which sensor type was used to process the training dataset.

Syntax

```
Mdl = initTrainDS(Mdl, TrainDS)
```

Description

Mdl = initTrainDS(Mdl, TrainDS) attaches regression relevant data information to model struct and initiates the training data with references and regression targets.

Input Argurments

Mdl model struct.

TrainDS training data struct which includes Info and Data struct.

Output Argurments

Mdl with attached dataset information, raw training data, refernce angles and regression targets for cosine and sine predictions.

Requirements

- Other m-files required: None
- Subfunctions: None
- MAT-files required: Train_*.mat

See Also

- initGPR
- Training and Test Datasets

Created on February 20. 2021 by Tobias Wulf. Copyright Tobias Wulf 2021.

```
function Mdl = initTrainDS(Mdl, TrainDS)

    % set model parameters from training dataset and training data dependencies
    % N number of angles and refernces in degree
    Mdl.N = TrainDS.Info.UseOptions.nAngles;
    Mdl.Angles = TrainDS.Data.angles';

    % D sensor array square dimension of DxD sensor array
    Mdl.D = TrainDS.Info.SensorArrayOptions.dimension;

    % P number of predictors in sensor array
    Mdl.P = TrainDS.Info.SensorArrayOptions.SensorCount;

    % get sensor type from dataset
    Mdl.Sensor = TrainDS.Info.UseOptions.BaseReference;

    % choose period factor depending on sensor type
    % how many sinoid periods are abstract on a full rotation by 360°
    switch Mdl.Sensor
        case 'TDK'
            Mdl.PF = 1;
        case 'KMZ60'
            Mdl.PF = 2;

        otherwise
            error('Unkown Sensor %s.', Mdl.Sensor);
    end

    % get reference angles in degree and transpose to column vector
    % get sinoid target vectors depending period factor,
    % transpose because angles2sinoids works with row vectors
    [Mdl.Ysin, Mdl.Ycos] = angles2sinoids(Mdl.Angles, ...
        false, Mdl.PF);

    % attach training data fro cosine and sine to model
    Mdl.Xcos = TrainDS.Data.Vcos;
    Mdl.Xsin = TrainDS.Data.Vsin;
end
```

Published with MATLAB® R2020b