

# Notes from reading existing Matlab code for the paper - Matthias

## Code Structure

All of the code is run from `RUN.m`, which runs some other scripts: `COMPUTE_RESULTS.m` and `CREATE_PLOTS.m`

## Compute Results

### Main Loop

The main loop of the main function runs all of the computations. The order of operations are as follows. 1. Configure (set up fit function, RNG seed) 2. Get data for the relevant dataset (vectors of feature positions and binary labels) 3.

Call ODENet with  $\mathbf{df} = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$  if the neural network being trained is a '`net`',

if not (the type is in '`ResNet`', '`ODENet`', '`ODENetSimplex`') use an empty vector. The result is the network to be trained. 4. Produce plots for initial predictions and transformed points 5. Network training loop

### Train Network

Loop for `niter` iterations: 1. Compute gradient using ODENet gradient function 2.

### ODENet

**The Network Relation** The class relies on the network relation:

$$x^{k+1} = \alpha^k x^k + h^k \cdot \text{link}(K^k x^k + b^k),$$

where the link function is also known as the activation function ##### Gradient