

Supplementary Material S1

```
% Build the structure of the neural network
hiddenLayerSize1 = 80;
hiddenLayerSize2 = 40;
net = feedforwardnet([hiddenLayerSize1, hiddenLayerSize2]);

% Set the activation function for each layer
net.layers{1}.transferFcn = 'tansig';
net.layers{2}.transferFcn = 'tansig';
net.layers{3}.transferFcn = 'purelin';

% Set the learning algorithm and error function
net.trainFcn = 'trainbr';
net.performFcn = 'mse';

% Set training parameters
net.trainParam.epochs = 1000;
net.trainParam.lr = 0.3;
net.trainParam.max_fail = 20;
net.trainParam.min_grad = 1e-7;
net.trainParam.mu=0.005;
net.trainParam.goal = 0;
net.trainParam.mu_inc=10;
net.trainParam.mu_dec=0.1;
net.trainParam.mu_max=1e10;

% 70% of data are used for training and validation test, the remaining
% 30% for an additional final test to calculate the percentage error
c = cvpartition(size(X,1),'HoldOut',0.3);
X_train = X(c.training,:);
Y_train = Y(c.training,:);
X_FinalTest = X(c.test,:);
Y_FinalTest = Y(c.test,:);

% Traing the neural network
[net,tr] = train(net,X_train,Y_train);

% Data training and final test prediction
y_train = net(X_train');
y_val = net(X_FinalTest');

% Visualize training curve
figure;
plot([tr.epoch],[tr.perf]);
hold on;
plot([tr.epoch],[tr.vperf'],'g');
legend('Training Error', 'Validation Error');
title('Training curve');
xlabel('Epoch');
ylabel('Mean Square Error');

% Visualize data training prediction
figure;
scatter(Y_train,y_train);
hold on;
plot(Y_train,Y_train);
title('Data training prediction');
xlabel('Real values');
ylabel('Predicted values');
legend('Thickness', 'Width', 'Length');
```

```
% Visualize final test prediction results
```

```
figure;
```

```
scatter(Y_FinalTest,y_val);
```

```
hold on;
```

```
plot(Y_FinalTest,Y_FinalTest,'r');
```

```
title('Final Test prediction');
```

```
xlabel('Real values');
```

```
ylabel('Predicted values');
```

```
legend('Thickness', 'Width', 'Length');
```

```
% New data prediction
```

```
y_new = net(Xprediction)';
```

```
%calculation of percentage error of both data used for training and for final test
```

```
errore_percentuale_training = mean(abs(Y_train- y_train)./Y_train) * 100;
```

```
errore_percentuale_TestFinale = mean(abs(Y_FinalTest- y_val)./Y_FinalTest) * 100;
```