Model Selection

CE/CZ4042 – Tutorial 6

1. A three-layer perceptron network is used to approximate the following function mapping:

$$y = sin(\pi x_1)cos(2\pi x_2)$$

where $-1.0 \le x_1, x_2 \le +1.0$.

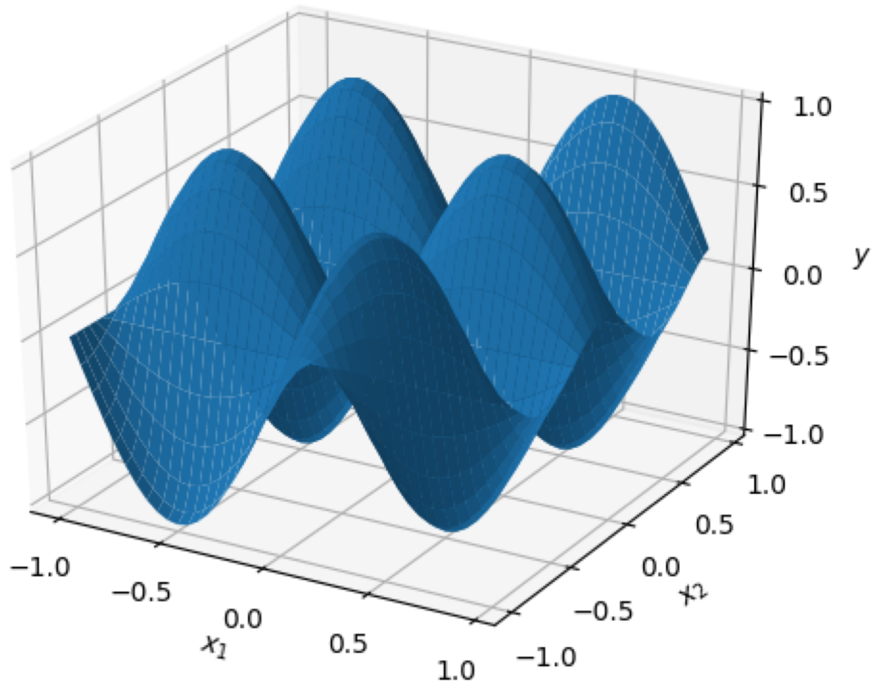
By using 100 data points in an equally spaced 10x10 grid of the input space, find the optimal number of hidden neurons for the approximation by using the following procedures:

a. Random subsampling
b. Five-fold cross validation
c. Three-way data split

Use a learning factor $\alpha = 0.05$ and learning up to 20,000 epochs.

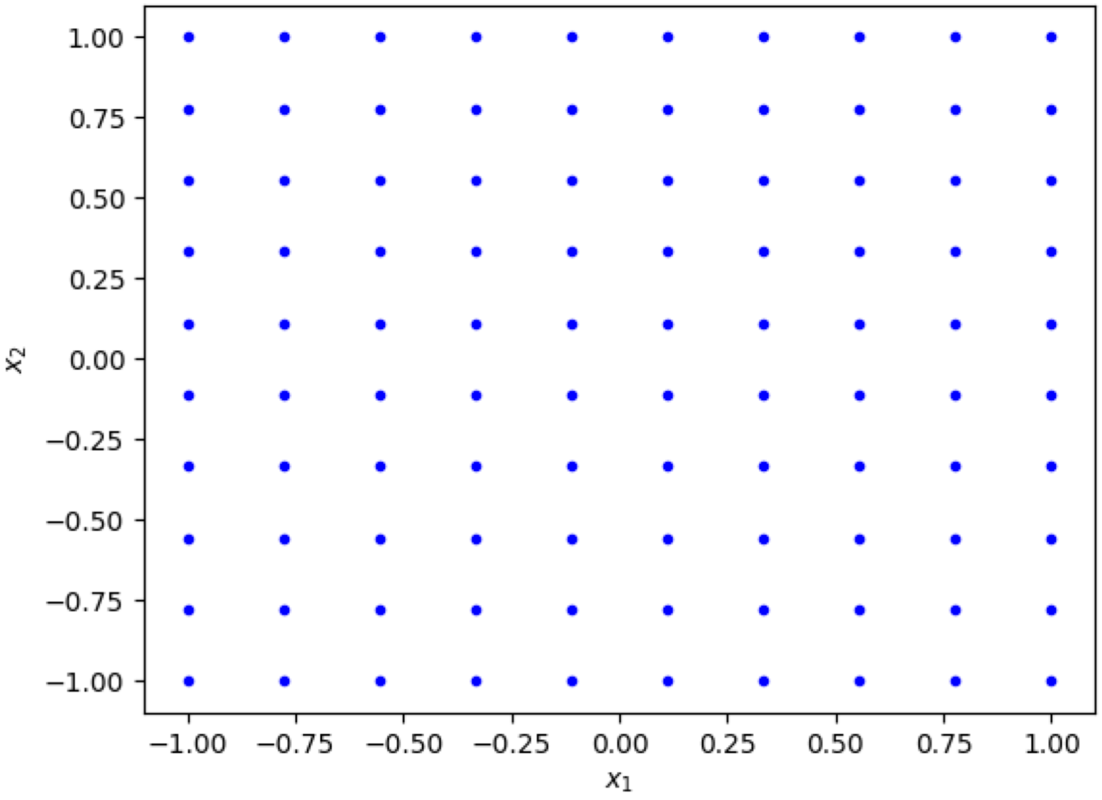
Show learning curves and predicted data points with the optimal number of hidden neurons.

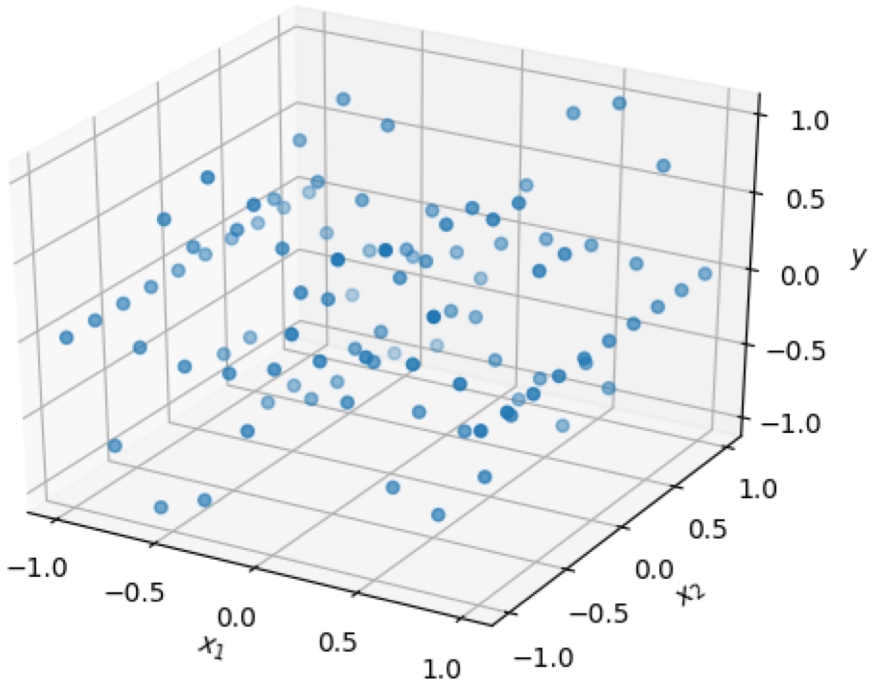
$$y = sin(\pi x_1)\cos(2\pi x_2) \quad \text{where } -1.0 \ \leq x_1, x_2 \leq +1.0.$$



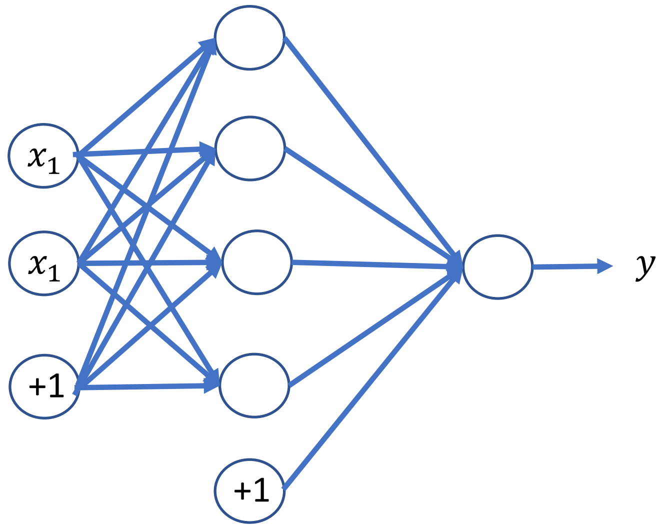
$$y = sin(\pi x_1)\cos(2\pi x_2) \quad \text{where } -1.0 \ \leq x_1, x_2 \leq +1.0.$$

Data is in a grid of 10x10





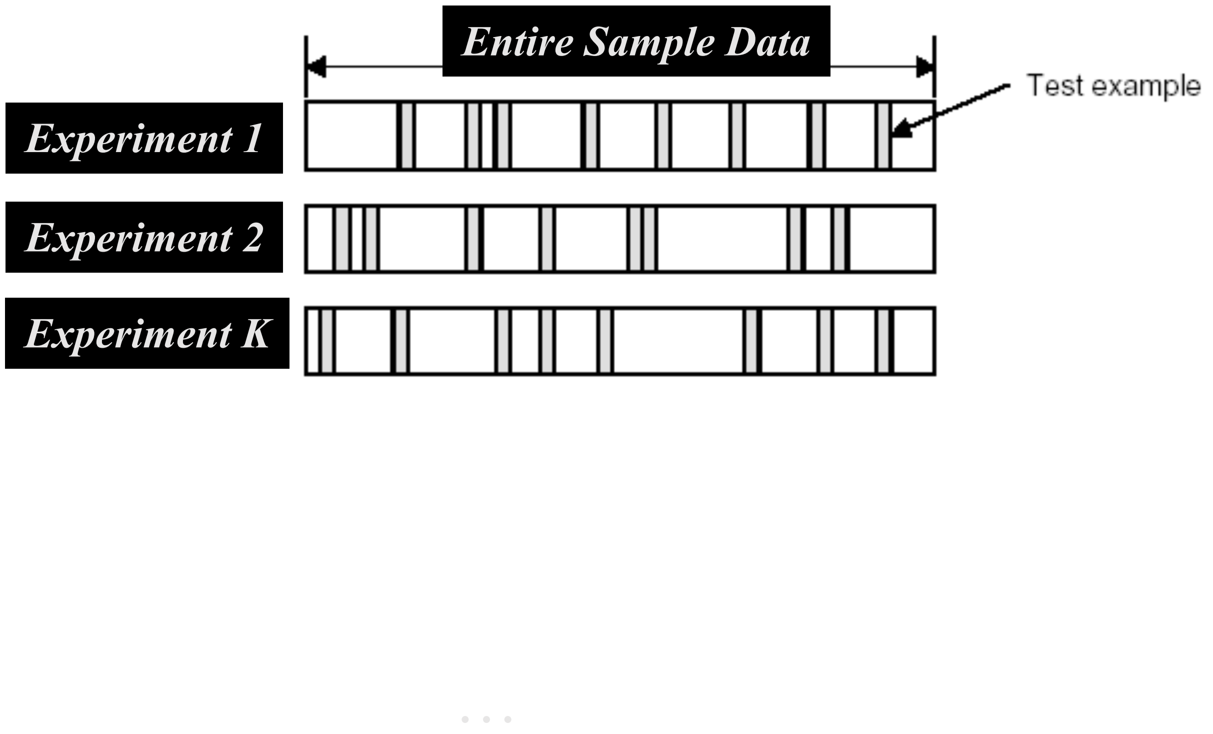
$$y = sin(\pi x_1)\cos(2\pi x_2) \quad \text{where } -1.0 \ \leq x_1, x_2 \leq +1.0.$$



For hidden neurons, let $f(u) = \frac{1}{1+e^{-u}}$

Output neuron is a linear neuron

K Data Splits: Random Subsampling



K Data Splits: Random Subsampling

For each experiment k:

For each model m:

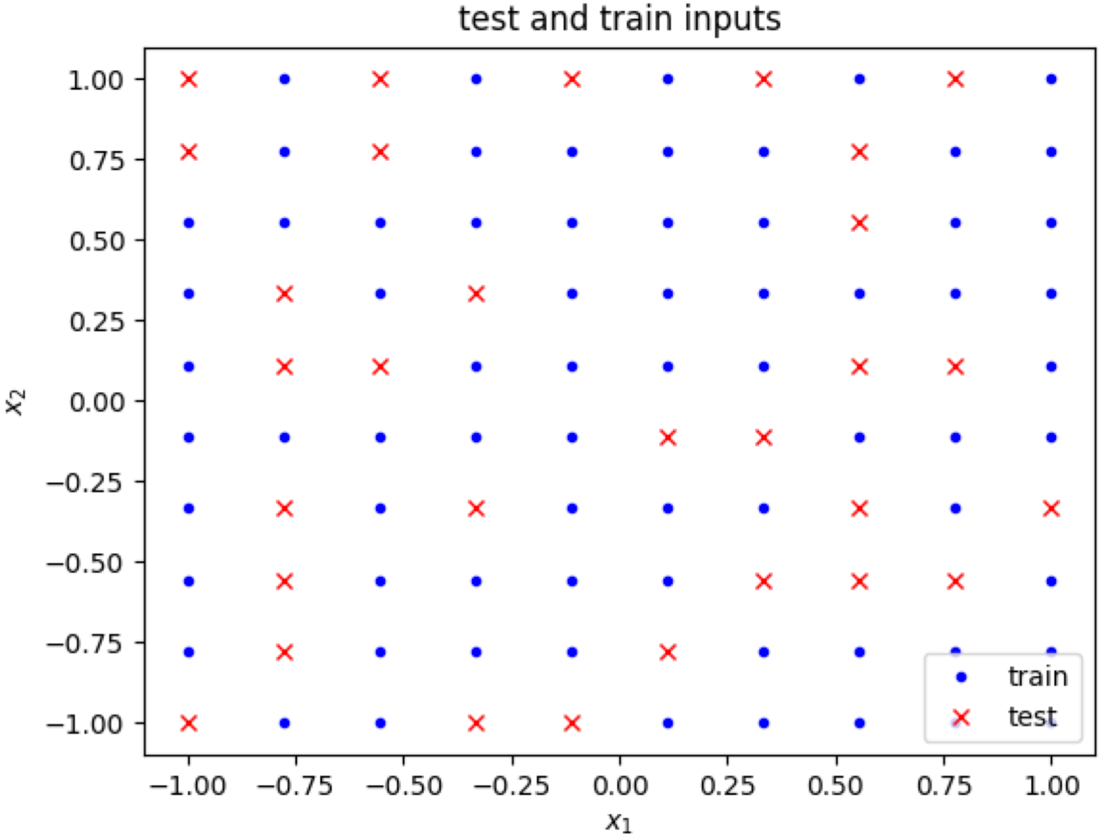
Compute error $e_{k,m}$

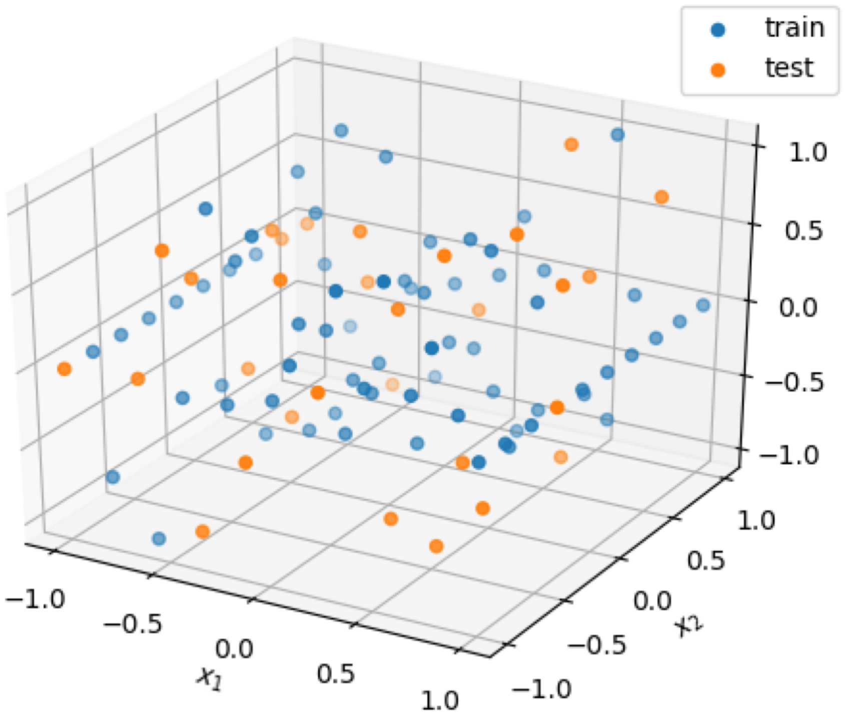
Compute mean error $e_m = \frac{1}{K} \sum_{k=1}^{K} e_{k,m}$

Optimal model with minimum error, $m^* = argmin\ e_m$

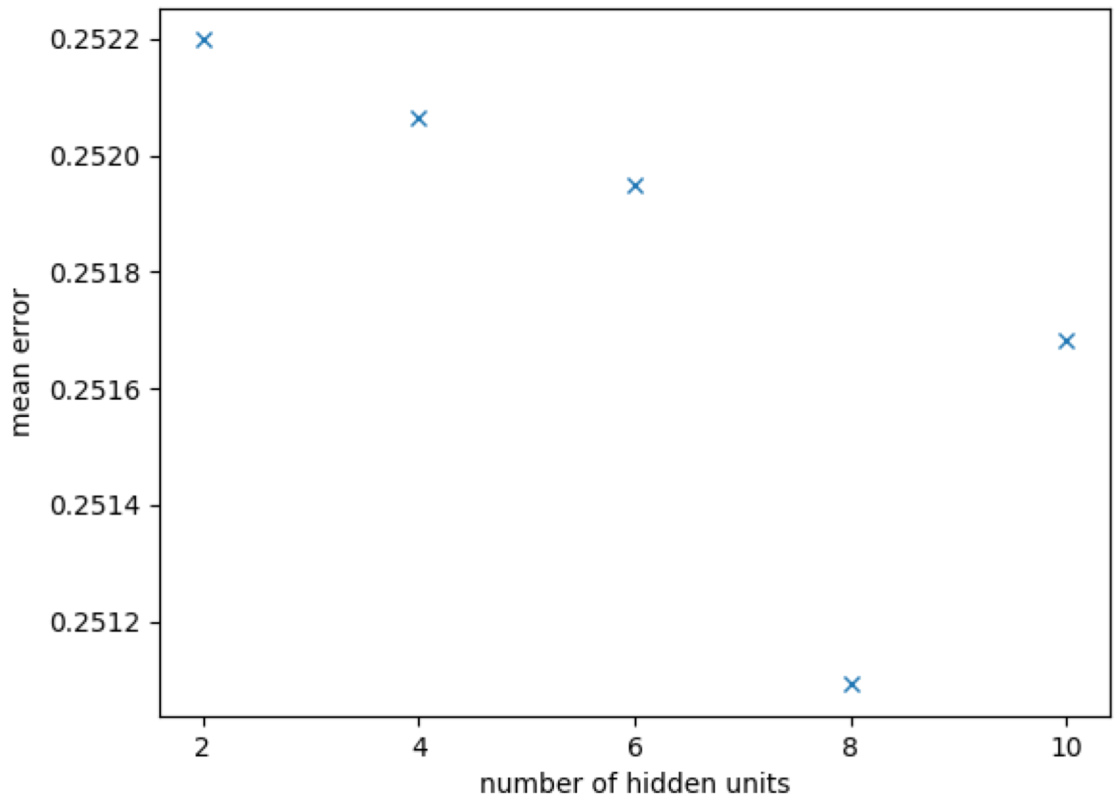
...

Train and test data for one experiment: [30: 70] split



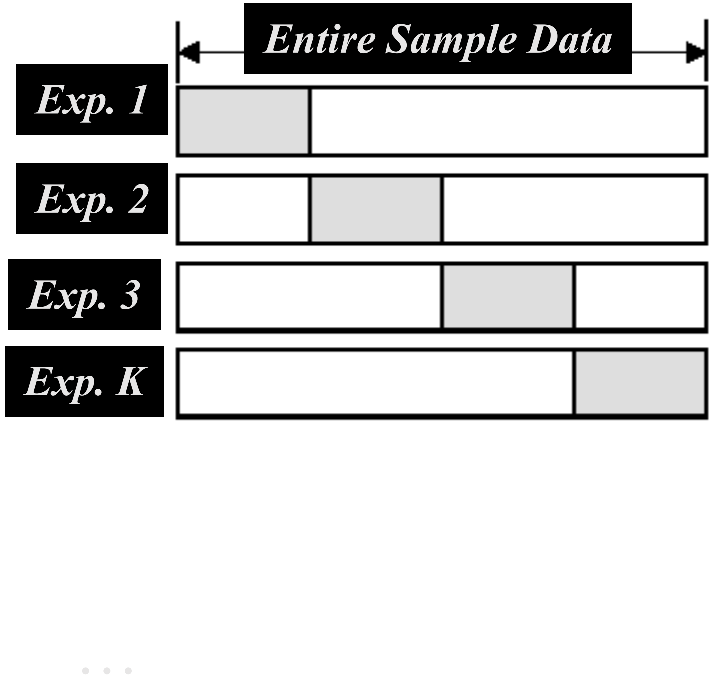


Mean error of 10 experiments



Optimum number of hidden neurons = 8

K-fold Cross Validation



For every fold f:

For every model m

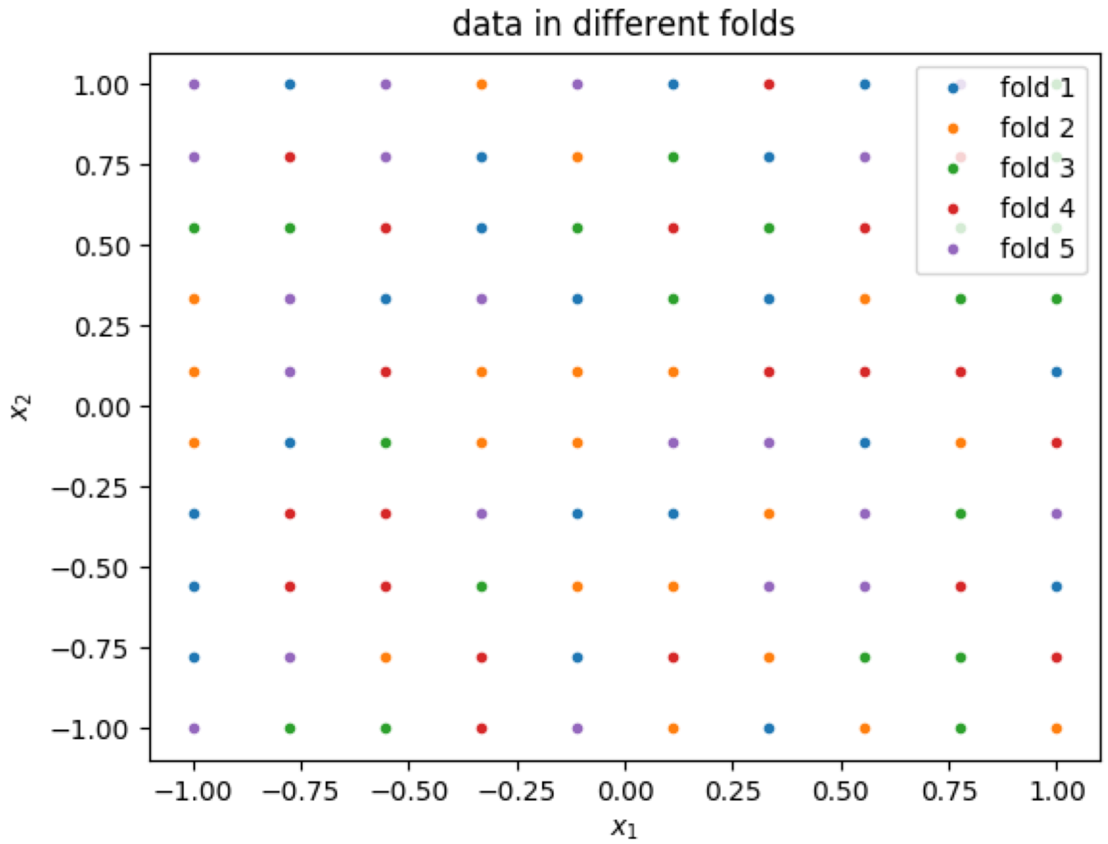
Train the model, using data not in fold f

Error $e_{m,f}$ = error on data in fold f

For every model m

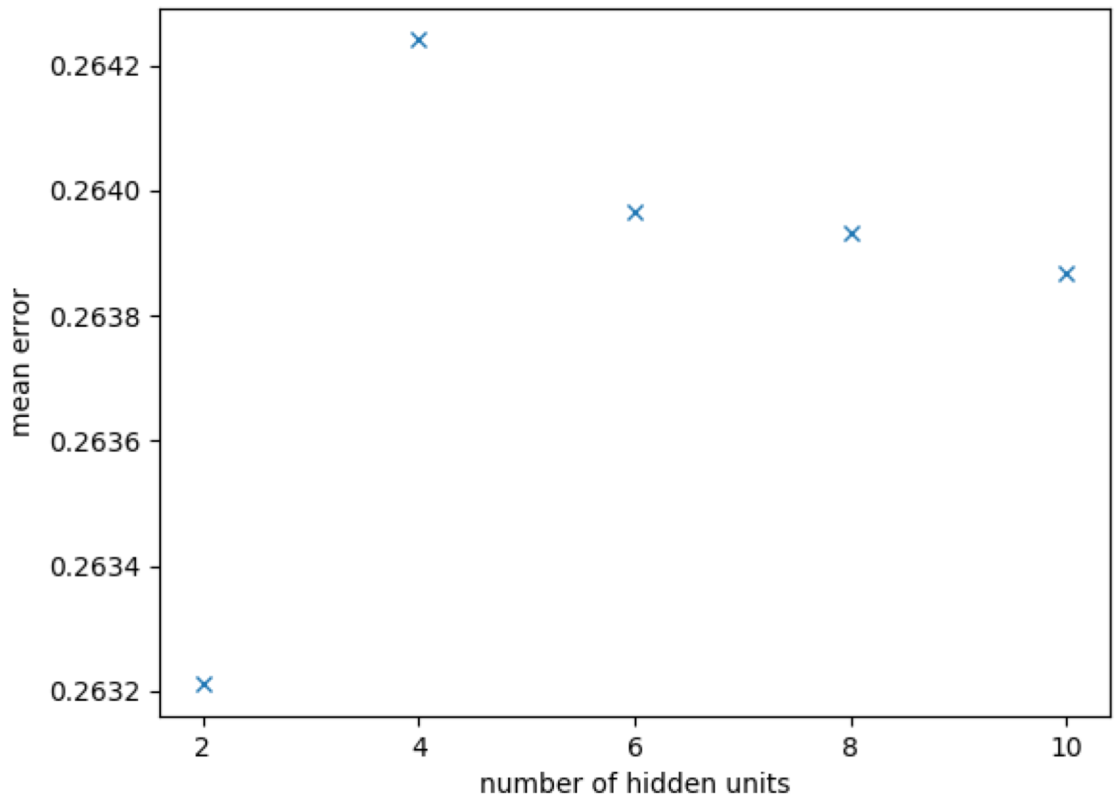
CV error $e_m = \frac{1}{F} \sum_f e_{m,f}$

Select the model with minimum CV error, $m^* = argmin\ e_m$



data in different folds
fold 1
fold 2
fold 3
fold 4
fold 5
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
x_1
x_2

Mean cross-validation error of 10 experiments



Optimum number of hidden neurons = 2

Three-Way Data Splits Method

- **Training set**: examples for *learning* to fit the parameters of several possible classifiers. In the case of DNN, we would use the training set to find the “optimal” weights with the gradient descent rule.

$$\boldsymbol{W}^*, \boldsymbol{b}^* = \underset{\boldsymbol{W},\boldsymbol{b}}{\operatorname{arg}min}\, J(\boldsymbol{W}, \boldsymbol{b})$$

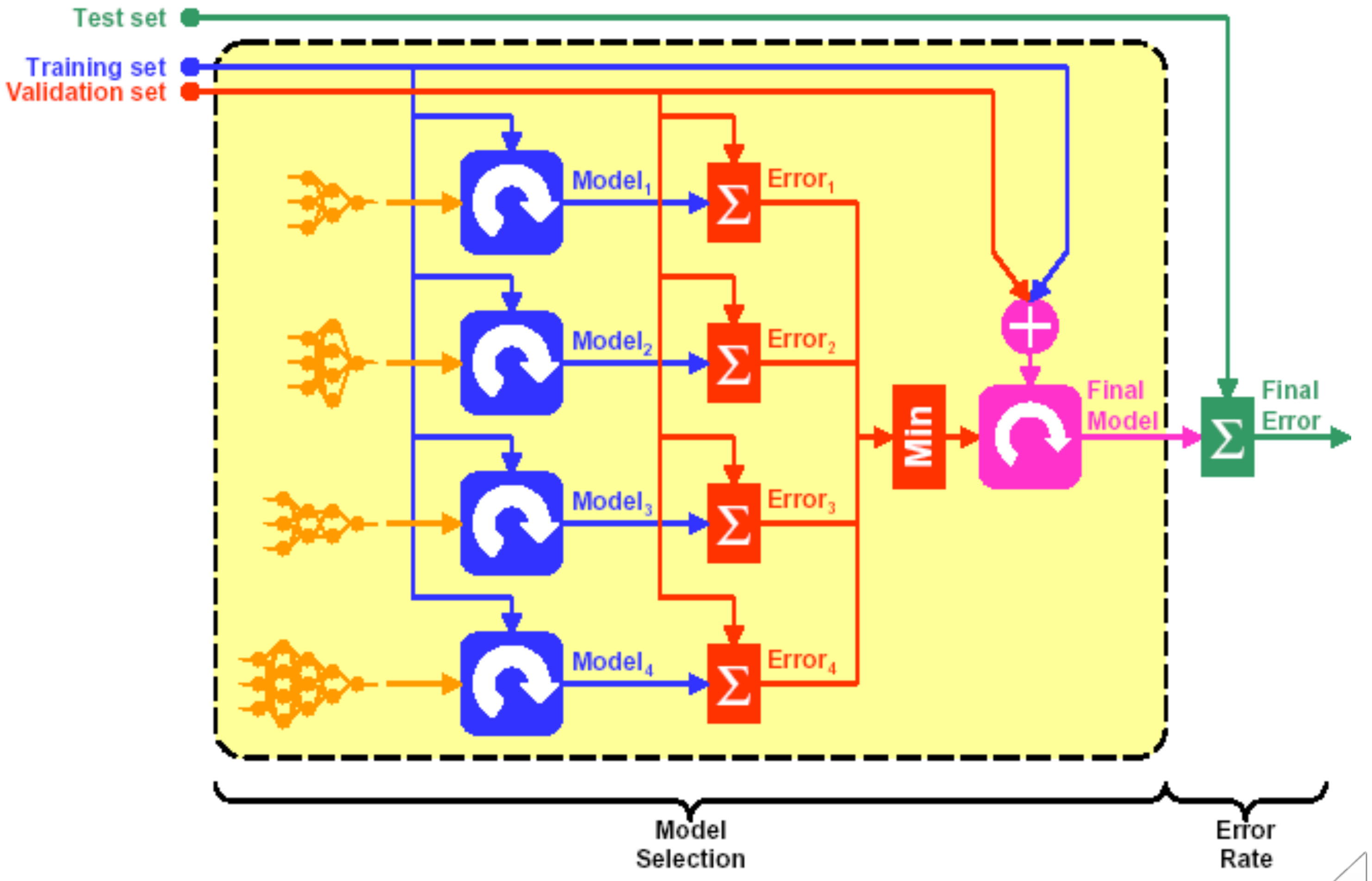
- **Validation set**: examples to *determine* the error J_m of different models m, using the validation set. The optimal model m^* is given by

$$m^* = \underset{m}{\operatorname{arg}min}\, J_m$$

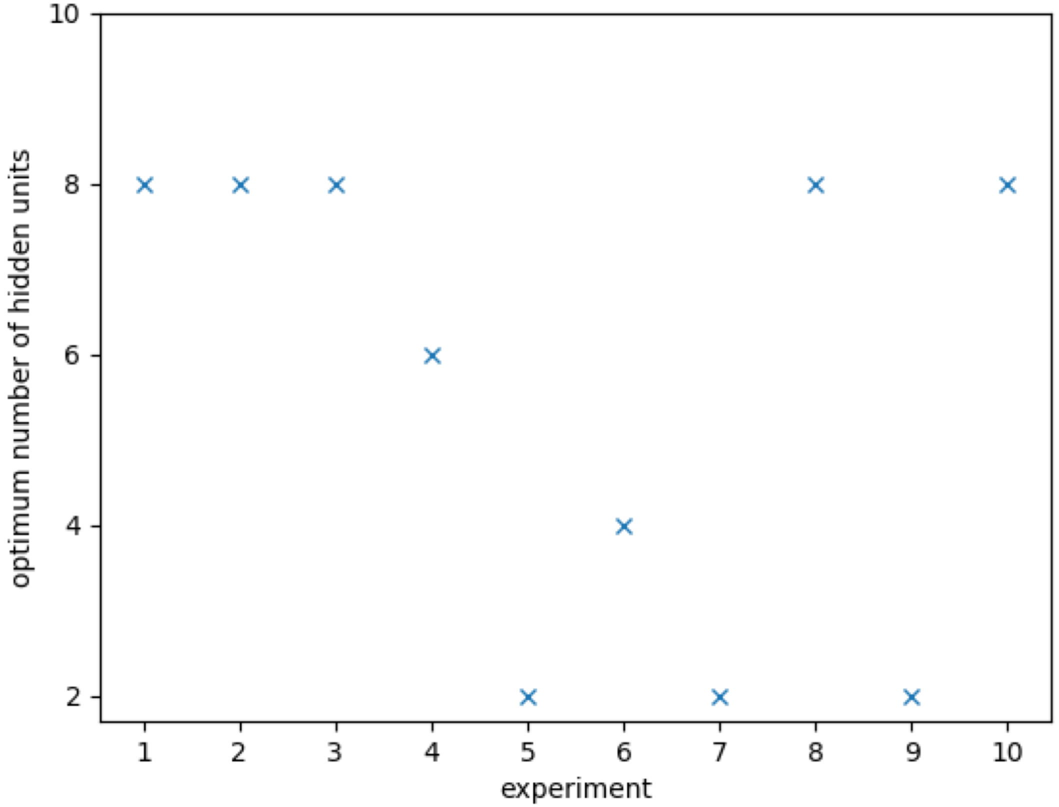
- **Training + Validation set:** combine examples used to re-train/redesign $model_{m^*,}$ and find new “optimal” weights.

- **Test set**: examples used only to *assess* the performance of a *trained model* m^*. We will use the test data to estimate the error rate after we have trained the final model with train + validation data.

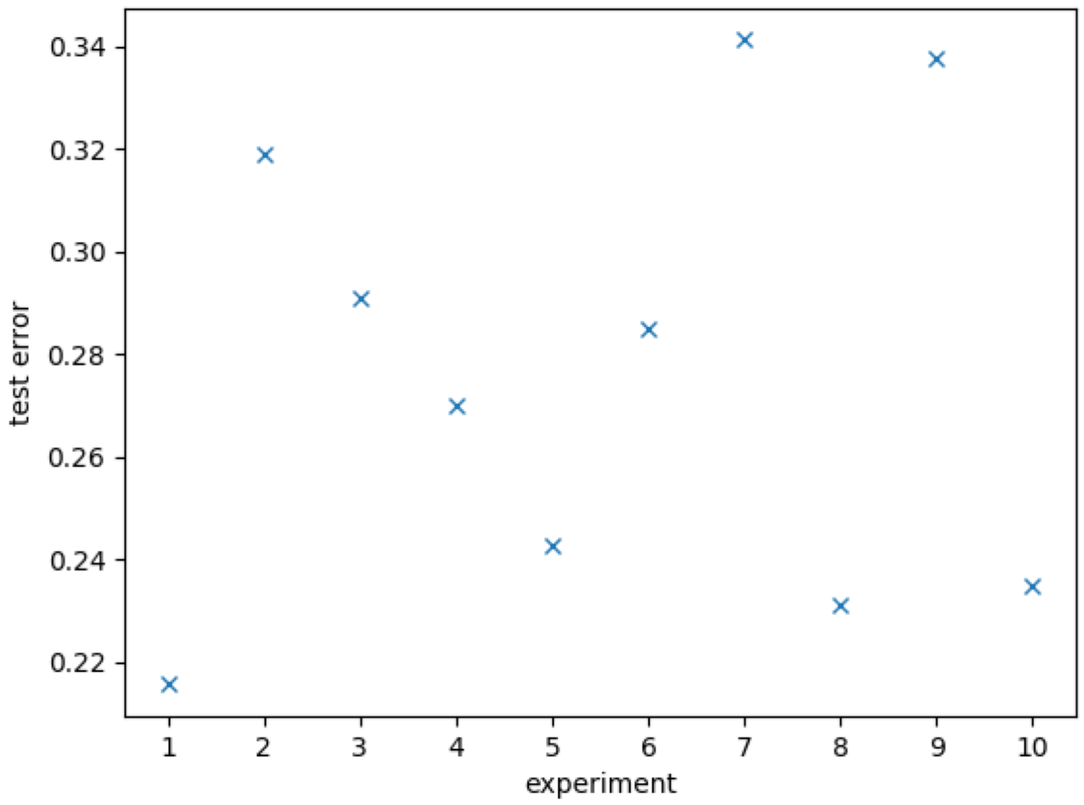
Three-Way Data Splits Method




Training, validation and test inputs
train
validate
test
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00



10
8
6
4
2
optimum number of hidden units
1
2
3
4
5
6
7
8
9
10
experiment



Optimal number of hidden neurons is 8