Cwk Appendix A: NN Datasets

Servo Dataset

Relevant Information

Ross Quinlan: "This is an interesting collection of data provided by Karl Ulrich at MIT. It covers an extremely non-linear phenomenon - predicting the rise time of a servomechanism in terms of two (continuous) gain settings and two (discrete) choices of mechanical linkages."

"*I seem to remember that the data was from a simulation of a servo system involving a servo amplifier, a motor, a lead screw/nut, and a sliding carriage of some sort. It may have been on the translational axes of a robot on the 9th floor of the AI lab. In any case, the output value is almost certainly a rise time, or the time required for the system to respond to a step change in a position set point.*"

Number of Instances: 167

Number of Attributes: 4 (categorical/integer) + numeric class attribute

Attribute information:

1. motor: A, B, C, D, E
2. screw: A, B, C, D, E
3. pgain: 3, 4, 5, 6
4. vgain: 1, 2, 3 ,4, 5
5. class: 0.13 to 7.10

References:

UCI Repository of ML Databases: https://archive.ics.uci.edu/ml/machine-learning-databases/servo/ , retrieved February 2023.

Dorian Suc and Ivan Bratko. Combining Learning Constraints and Numerical Regression. National ICT Australia, Sydney Laboratory at UNSW.

Quinlan, J.R., "Learning with continuous classes", Proc. 5th Australian Joint Conference on AI, Singapore: World Scientific.

Iris Plant Dataset

Relevant Information

The *Iris* classification problem has three classes (categories) to be classified: whether an *Iris* plant sample is from the type *Setosa*; *Versicolour*; or *Virginica*.

The data set contains 3 classes of 50 instances each, where each class refers to a type of *Iris* plant. The first class is linearly separable from the other two and the latter ones are not linearly separable from each other.

Each sample has four attributes characterizing the length/width (in cm) of the flower's sepals and petals.

All the four attributes have continuous values in the following range: sepal length in [4.3, 7.9], sepal width in [2.0, 4.4], petal length in [1.0, 6.9], and petal width in [0.1, 2.5].

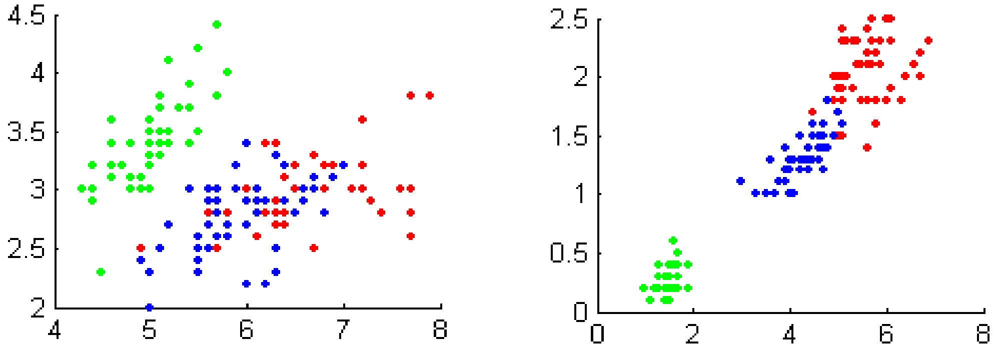

Fig.1a. Pair-wise plots of the class distribution of the three Iris classes (left: sepal width vs. sepal length; right: petal width vs. petal length). It can be seen that two of the classes are not linearly separable.

Fig. 1b. The Iris plant.

This is perhaps the best known database to be found in the pattern recognition literature. The *Iris* classification problem has three classes to be classified: whether an *Iris* sample is from the type *Setosa*; *Versicolour*; or *Virginica*. (Also called *Bearded Irises*, *Aril Irises*, and *Beardless Irises*). *Bearded Iris* is identified by thick, bushy "*beards*" on each of the falls (lower petals) of the blossoms. Two very different types of irises are grouped together under the term "*Aril*". These are the *oncocyclus* and *regelia* irises of the Near East. Although they have beards, they are not classified with the bearded irises because they are so different. Actually, their beards are rather sparse, being long and straggly on the *regelias*, and nothing more than a wide "fuzzy" patch on the *oncocyclus*. The *Arils* show dark signal spots below the beards with much veining and speckling, in an unbelievable range of colours. *Beardless Irises* are mostly native to Asia. The first four types are commonly grown in gardens, and they all bloom after the *Tall Bearded* (TB), extending the iris season even longer. The fifth type, the *Pacific Coast Native*, blooms before the TBs and is native to the western regions of the United States.

Number of Instances: 150 (50 in each of three classes)
Number of Attributes: 4 numeric, predictive attributes and the class label
Attribute Information (columns 1 to 4; all in cm): sepal length; sepal width; petal length; petal width
Iris class: (one of three) -- *Setosa*; -- *Versicolour*; -- *Virginica*.
Summary Statistics:

	Min	Max	Mean	SD	Class Correlation
sepal length:	4.3	7.9	5.84	0.83	0.7826
sepal width:	2.0	4.4	3.05	0.43	-0.4194
petal length:	1.0	6.9	3.76	1.76	0.9490 (high!)
petal width:	0.1	2.5	1.20	0.76	0.9565 (high!)

References (too many to mention - here are just a few):

UCI Repository of ML Databases: https://archive-beta.ics.uci.edu/dataset/53/iris retrieved February 2023.

A. Bortoletti, C. Fiore, S. Fanelli, and P. Zellini, "A new class of quasi-newtonian methods for optimal learning in MLP-networks", *IEEE Trans. Neural Networks,* vol. 14, pp. 263-273, 2003.

M. Rocha, P. Cortez, and J. Neves, "Evolutionary neural network learning", *LNAI 2902*, Springer, pp. 740–746, 2003.

Additional notes

There are generally four steps in the NN training process:

- ✓ Assemble and pre-process the training data;
- ✓ Design and create the network object;
- ✓ Adopt learning technique and train the network;
- ✓ Simulate (test) the network response to new inputs to see if it can generalise.

Once you have chosen the dataset, you may consider the following:

- Representation/Coding the data (input and output)!
- Normalising and standardising the dataset. If you have time, even some basic statistics - Linear Discriminant Analysis, Principal Component Analysis, etc. (not obligatory)!
- Subdividing into three subsets (if decided to use *split sample* training)!
- Subdividing into 10 subsets (if decided to use *cross-validation* training)!

Design the network topology and architecture:

- number of layers/neurons in each;
- activation/transfer functions in each layer;
- training method/function (e.g., BP with *Levenberg-Marquardt*).

Determine the number of neurons in the hidden layer(s):

Some rule-of-thumb approaches for determining the correct number of neurons in the hidden layers:

- in the range between the size of the input layer and the size of the output layer;
- 2/3 of the input layer size, plus the size of the output layer;
- less than twice the input layer size (See also the Appendix of Lecture4).

Experiment with different topologies.

Training strategy

– Split sample:

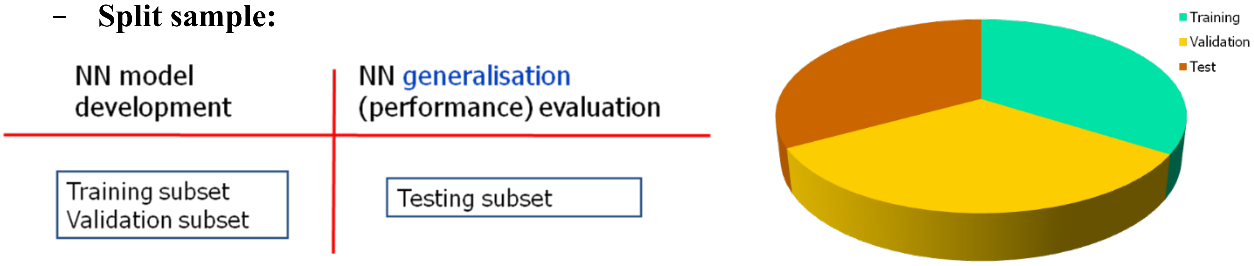

Fig.2. ‘Split sample’ training.

– Cross-validation:

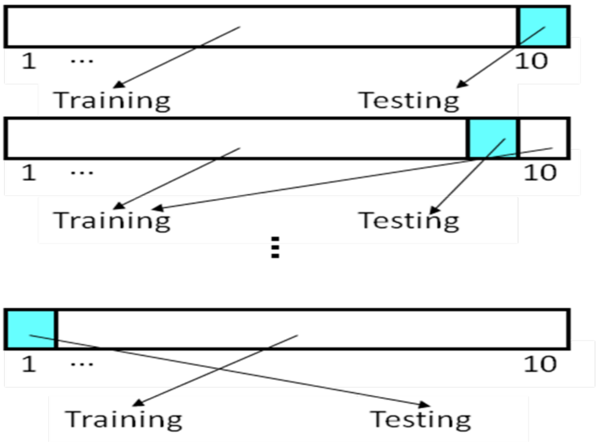

Fig.3. Splitting the Data set for “Cross-validation training”.

Neural Nets Parameters

- number of hidden layers and neurons in them, connectivity;
- training technique;
- transfer functions;
- initial weights;
- learning rate, momentum;
- stopping conditions (min error, number of iterations, training time, etc.).