

# TESTING OF CLASSIFIERS

JAN BIELECKI

## CONTENTS

|     |                                  |   |
|-----|----------------------------------|---|
| 1   | Introduction                     | 1 |
| 2   | Results                          | 1 |
| 2.1 | Multilayer perceptron . . . . .  | 1 |
| 2.2 | K-Nearest Neighbor . . . . .     | 3 |
| 2.3 | Support vector machine . . . . . | 4 |
| 3   | Discussion                       | 5 |

## INTRODUCTION

This report concerns checking the quality of the three classifiers:

1. Multilayer perceptron (MLP) - a class of feedforward artificial neural network.
2. K-Nearest Neighbor (KNN) - classification based on the classes of k-nearest neighbors
3. Support vector machine (SVM) - representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.

The data used for calculations are the results of medical examinations for 568 patients diagnosed with cancer. Every record (patient) consists of numerical results of 10 different observables (for example, the number of lymphocytes). For each patient we have information about the type of the tumour (it is benign - class 'B' or malignant - class 'M').

## RESULTS

### *Multilayer perceptron*

Description of the experiments for MLP:

1. Each result (recognition error as a function of the epoch) is the average of 100 experimental series
2. Calculations were carried out for different sizes of the hidden layer (5, 6, 7, 8 - these sizes bring the best recognition results)
3. Calculations were carried out for different network training functions (Levenberg-Marquardt backpropagation (lm), Scaled conjugate gradient backpropagation (scg) and BFGS quasi-Newton backpropagation (bfg))

4. Data breakdown ratios: train = 450/568, validate = 59/568, test ratio = 59/568.

The results are presented in Figure 1 on the following page, 2 on the next page and 3 on page 3 .

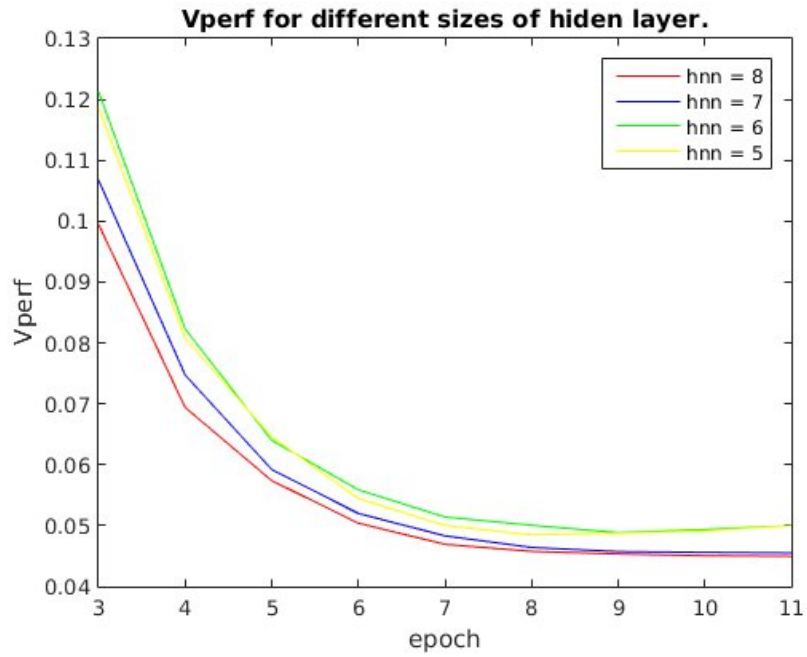


Figure 1: Prediction error for MLP method using Levenberg-Marquardt backpropagation (hnn - hidden layer size).

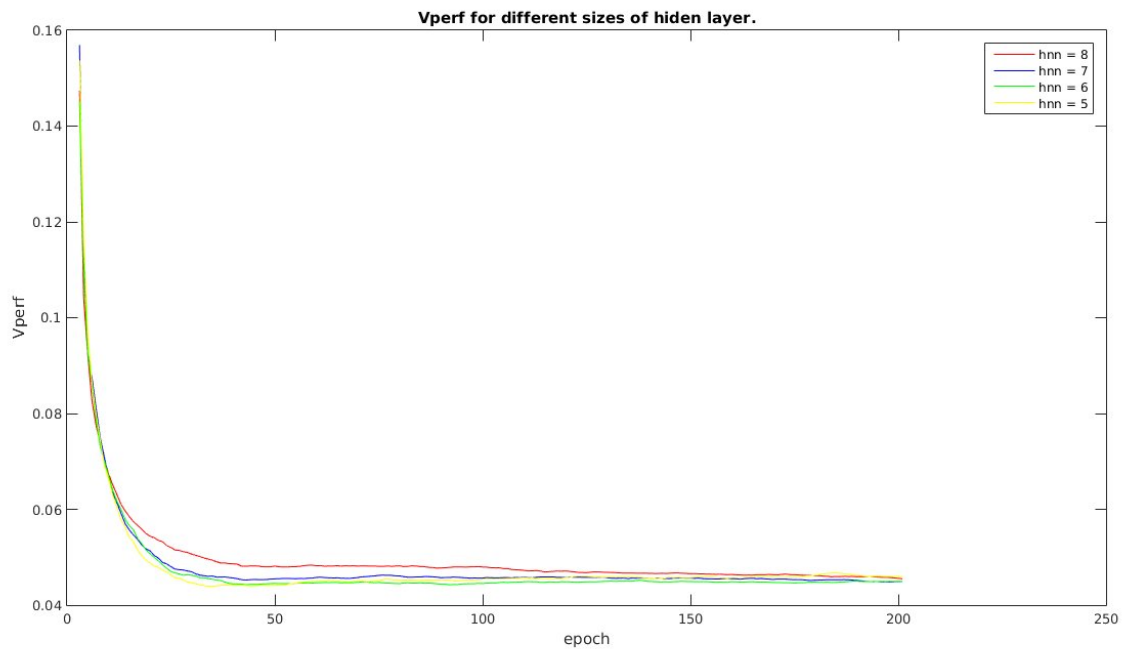


Figure 2: Prediction error for MLP method using Scaled conjugate gradient backpropagation (hnn - hidden layer size).

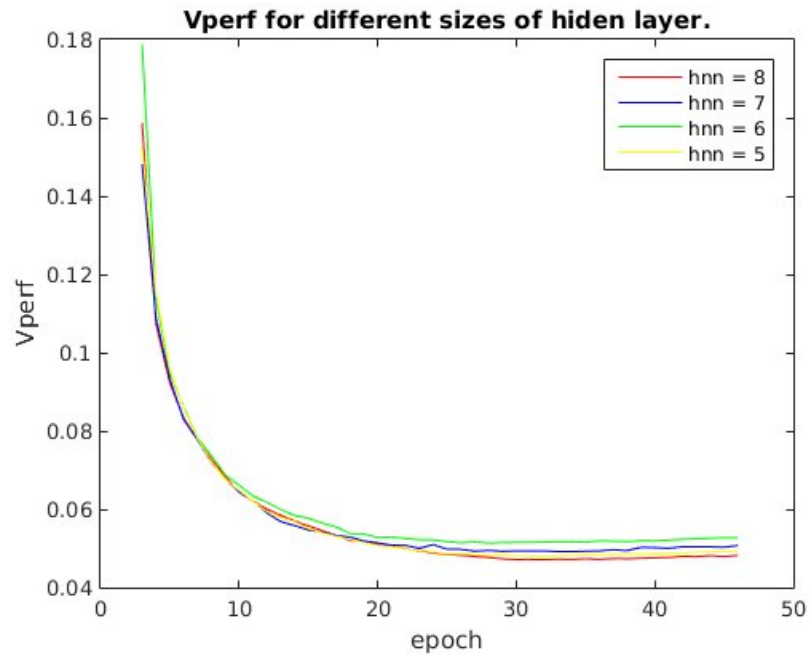


Figure 3: Prediction error for MLP method using BFGS quasi-Newton back-propagation (hnn - hidden layer size).

| fun/hnn | 5      | 6      | 7      | 8      |
|---------|--------|--------|--------|--------|
| lm      | 0.0485 | 0.0488 | 0.0455 | 0.0450 |
| scg     | 0.0439 | 0.0443 | 0.0448 | 0.0455 |
| bfg     | 0.0479 | 0.0513 | 0.0492 | 0.0471 |

Table 1: Minimum prediction errors for MLP method.

Table 1 shows best training results for each training function.

### *K-Nearest Neighbor*

Description of the experiments for KNN:

1. Data (for training the KNN Classifier) have been build with first 450 records
2. Assign the class for each of another 118 records based on k-nearest neighbours and find out how much of them has been correctly identified
3. Calculations were carried out for different k parameter (number of nearest neighbors)

The results are presented in Figure 4 on the following page.

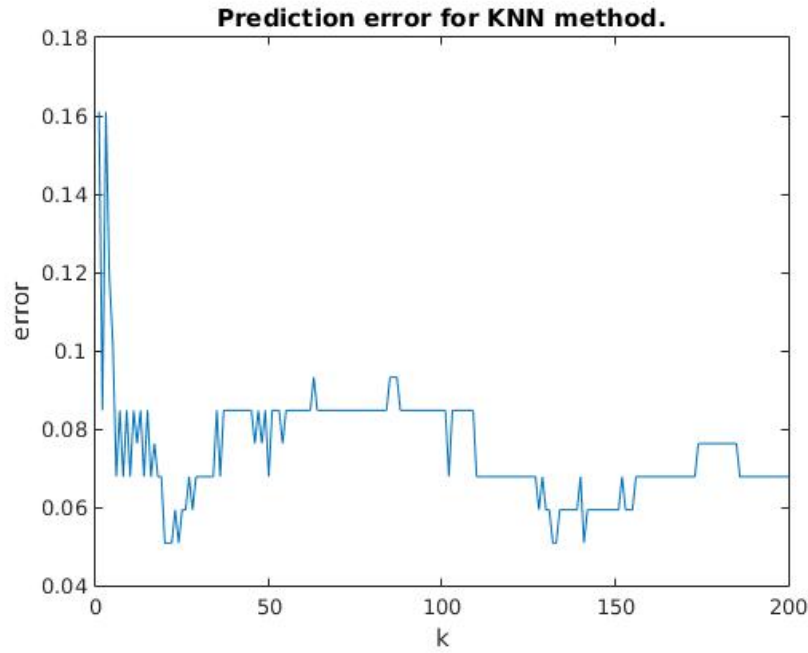


Figure 4: Prediction error for KNN method (k - number of nearest neighbors).

The best prediction result of KNN method:

$$\text{error}_{\text{MIN}} = 0.0471 \quad (1)$$

#### *Support vector machine*

Description of the experiments for SVM:

1. Data (for training the SVM Classifier) have been build with first 450 records
2. Assign the class for each of another 118 records based on SVM classifier and find out how much of them has been correctly identified
3. Calculations were carried out for different kernel functions (linear, gaussian, rbf, polynomial)

The results are presented in Table 2.

| function | linear | gaussian | rbf    | polynomial |
|----------|--------|----------|--------|------------|
| error    | 0.3475 | 0.1271   | 0.1271 | 0.1017     |

Table 2: Prediction errors for SVM method.

Next, we use principal component analysis for reduce data dimension. Our raw data is 10-dimensional space. We compute principal components for our raw data (using 'princomp' matlab function) and we find out that two components account for 99.99% of the variance. The same as before we used 450 records for train and another 118 records for tests.

The minimum prediction error we got by using linear kernel function:

$$\text{error}_{\text{MIN}} = 0.1102 \quad (2)$$

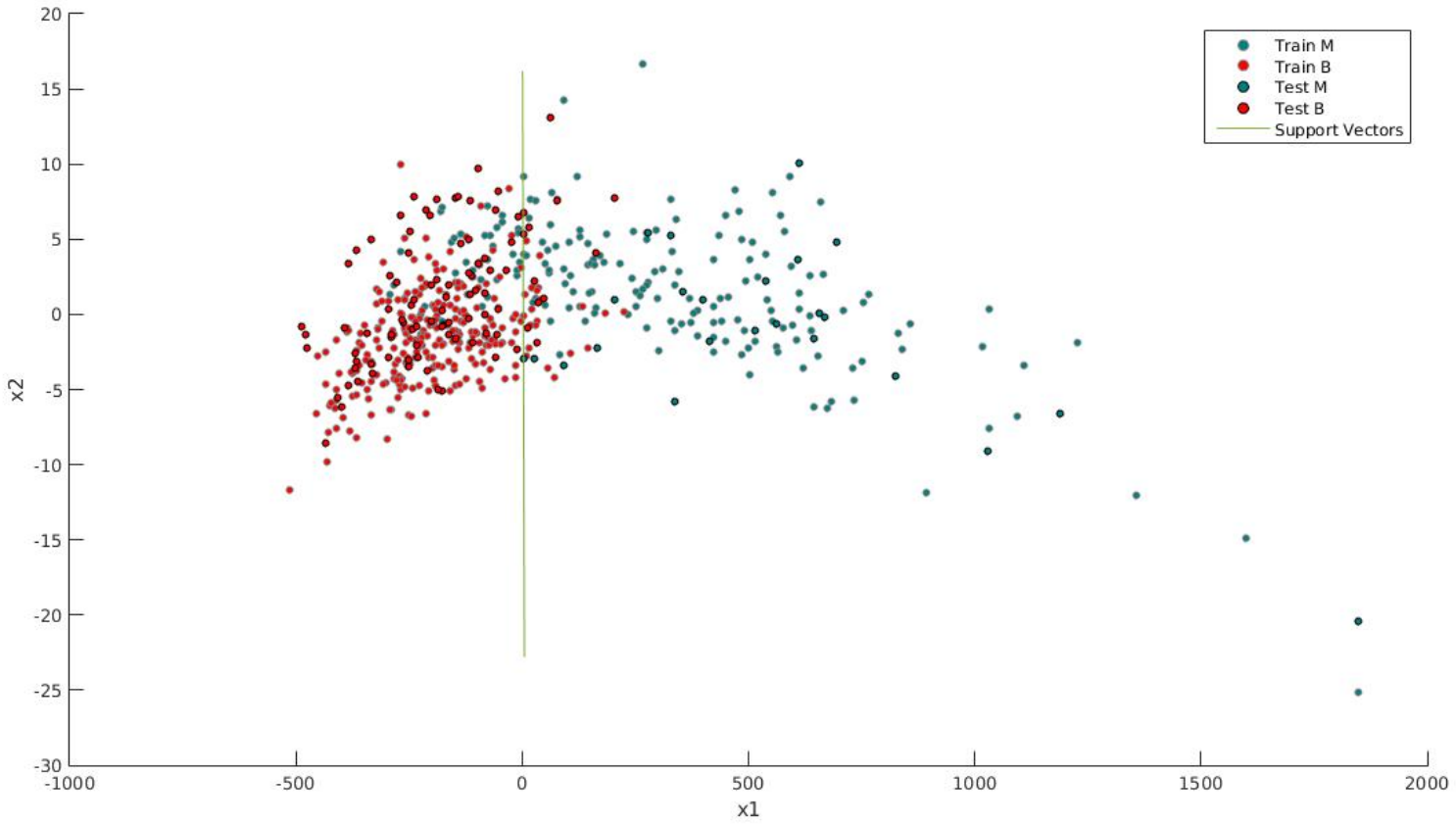


Figure 5: SVM model for reduced state space.

Figure 5 illustrates the SVM method for reduced state space (space created using principal component analysis).

#### DISCUSSION

Table 3 summarizing the best results of the tested classifiers:

| classifier                  | MLP    | KNN    | SVM    |
|-----------------------------|--------|--------|--------|
| $\text{error}_{\text{MIN}}$ | 0.0439 | 0.0471 | 0.1017 |

Table 3: Minimum prediction errors of the tested classifiers.

For our data the best classification method is MLP using Scaled conjugate gradient backpropagation as training function. KNN method got the same results as MLP using BFGS quasi-Newton backpropagation. The worst of the tested classifiers (for our specified data) turned out to be Support Vector Machine.