**104.Connectionist Bench (Vowel Recognition - Deterding Data)**

1. 数据库网址

http://archive.ics.uci.edu/ml/datasets/Connectionist+Bench+(Vowel+Recognition+-+Deterding+Data)

2. 数据库描述

【1.[数据集名称]数据集由[机构名或人名]采集；】The data used in our experiments were collected by E. Alpaydin, C. Kaynak, from Department of Computer Engineering,Bogazici University at July,1998.【2.用于[什么实验目的]】We used preprocessing programs made available by NIST to extract normalized bitmaps of handwritten digits from a preprinted form.【3】

【4】The database has 5620 samples, respectively belong to optdigits.tra with 3823 samples and optidigits.tes with 1797 samples. The categories of network system include seven categories, as shown in Table 1.

Table 1 Category Distribution of Network System [根据数据库绘制]

|  |  |  |  |
| --- | --- | --- | --- |
| Invasion Categories | optdigits.tra | optdigits.tes | Total Number of Samples |
|  |  |  |  |
|  |  |  |  |
| Total number of samples in total |  |  | 528 |

|  |  |
| --- | --- |
| **Abstract**: Speaker independent recognition of the eleven steady state vowels of British English using a specified training set of lpc derived log area ratios. |  |

**Source:**

David Deterding (data and non-connectionist analysis)   
Mahesan Niranjan (first connectionist analysis)   
Tony Robinson (description, program, data, and results) - "ajr **'@'** dsl.eng.cam.ac.uk"

**Data Set Information:**

The problem is specified by the accompanying data file, "vowel.data". This consists of a three dimensional array: voweldata [speaker, vowel, input]. The speakers are indexed by integers 0-89. (Actually, there are fifteen individual speakers, each saying each vowel six times.) The vowels are indexed by integers 0-10. For each utterance, there are ten floating-point input values, with array indices 0-9.   
  
The problem is to train the network as well as possible using only on data from "speakers" 0-47, and then to test the network on speakers 48-89, reporting the number of correct classifications in the test set.   
  
For a more detailed explanation of the problem, see the excerpt from Tony Robinson's Ph.D. thesis in the COMMENTS section. In Robinson's opinion, connectionist problems fall into two classes, the possible and the impossible. He is interested in the latter, by which he means problems that have no exact solution. Thus the problem here is not to see how fast a network can be trained (although this is important), but to maximise a less than perfect performance.

METHODOLOGY:

Report the number of test vowels classified correctly, (i.e. the number of occurences when distance of the correct output to the actual output was the smallest of the set of distances from the actual output to all possible

target outputs).

Though this is not the focus of Robinson's study, it would also be useful

to report how long the training took (measured in pattern presentations or

with a rough count of floating-point operations required) and what level of

success was achieved on the training and testing data after various amounts

of training. Of course, the network topology and algorithm used should be

precisely described as well.

VARIATIONS:

This benchmark is proposed to encourage the exploration of different node

types. Please theorise/experiment/hack. The author (Robinson) will try to

correspond by email if requested. In particular there has been some

discussion recently on the use of a cross-entropy distance measure, and it

would be interesting to see results for that.

RESULTS:

Here is a summary of results obtained by Tony Robinson. A more complete

explanation of this data is given in the exceprt from his thesis in the

COMMENTS section below. The program used to obtain these results is in the

code directory, /afs/cs.cmu.edu/project as "vowel.c".

+-------------------------+--------+---------+---------+

| | no. of | no. | percent |

| Classifier | hidden | correct | correct |

| | units | | |

+-------------------------+--------+---------+---------+

| Single-layer perceptron | - | 154 | 33 |

| Multi-layer perceptron | 88 | 234 | 51 |

| Multi-layer perceptron | 22 | 206 | 45 |

| Multi-layer perceptron | 11 | 203 | 44 |

| Modified Kanerva Model | 528 | 231 | 50 |

| Modified Kanerva Model | 88 | 197 | 43 |

| Radial Basis Function | 528 | 247 | 53 |

| Radial Basis Function | 88 | 220 | 48 |

| Gaussian node network | 528 | 252 | 55 |

| Gaussian node network | 88 | 247 | 53 |

| Gaussian node network | 22 | 250 | 54 |

| Gaussian node network | 11 | 211 | 47 |

| Square node network | 88 | 253 | 55 |

| Square node network | 22 | 236 | 51 |

| Square node network | 11 | 217 | 50 |

| Nearest neighbour | - | 260 | 56 |

+-------------------------+--------+---------+---------+

Notes:

1. Each of these numbers is based on a single trial with random starting

weights. More trials would of course be preferable, but the computational

facilities available to Robinson were limited.

2. Graphs are given in Robinson's thesis showing test-set performance vs.

epoch count for some of the training runs. In most cases, performance

peaks at around 250 correct, after which performance decays to different

degrees. The numbers given above are final performance figures after about

3000 trials, not the peak performance obtained during the run.

COMMENTS:

(By Tony Robinson)

The program supplied is slow. I ran it on several MicroVaxII's for many

nights. I suspect that if I had spent more time on it, it would have been

possible to get better results. Indeed, my later code has a slightly

better adaptive step size algotithm, but the old version is given here for

comatability with the stated performance values. It is interesting that,

for this problem, the nearest neighbour clasification outperforms any of

the connectionist models. This can be seen as a challange to improve the

connectionist performance.

The following problem description results and discussion is taken from my

PhD thesis. The aim was to demonstrate that many types of node can be

trained using gradient descent. The full thesis will be available from me

when it has been examined, say maybe July 1989.

Application to Vowel Recognition

--------------------------------

This chapter describes the application of a variety of feed-forward networks

to the task of recognition of vowel sounds from multiple speakers. Single

speaker vowel recognition studies by Renals and Rohwer [RenalsRohwer89-ijcnn]

show that feed-forward networks compare favourably with vector-quantised

hidden Markov models. The vowel data used in this chapter was collected by

Deterding [Deterding89], who recorded examples of the eleven steady state

vowels of English spoken by fifteen speakers for a speaker normalisation

study. A range of node types are used, as described in the previous section,

and some of the problems of the error propagation algorithm are discussed.

The Speech Data

(An ascii approximation to) the International Phonetic Association (I.P.A.)

symbol and the word in which the eleven vowel sounds were recorded is given in

table 4.1. The word was uttered once by each of the fifteen speakers. Four

male and four female speakers were used to train the networks, and the other

four male and three female speakers were used for testing the performance.

+-------+--------+-------+---------+

| vowel | word | vowel | word |

+-------+--------+-------+---------+

| i | heed | O | hod |

| I | hid | C: | hoard |

| E | head | U | hood |

| A | had | u: | who'd |

| a: | hard | 3: | heard |

| Y | hud | | |

+-------+--------+-------+---------+

Table 4.1: Words used in Recording the Vowels

Front End Analysis

The speech signals were low pass filtered at 4.7kHz and then digitised to 12

bits with a 10kHz sampling rate. Twelfth order linear predictive analysis was

carried out on six 512 sample Hamming windowed segments from the steady part

of the vowel. The reflection coefficients were used to calculate 10 log area

parameters, giving a 10 dimensional input space. For a general introduction

to speech processing and an explanation of this technique see Rabiner and

Schafer [RabinerSchafer78].

Each speaker thus yielded six frames of speech from eleven vowels. This gave

528 frames from the eight speakers used to train the networks and 462 frames

from the seven speakers used to test the networks.

Details of the Models

All the models had common structure of one layer of hidden units and two

layers of weights. Some of the models used fixed weights in the first layer

to perform a dimensionality expansion [Robinson89:sect3.1], and the remainder

modified the first layer of weights using the error propagation algorithm for

general nodes described in [Robinson89:chap2]. In the second layer the hidden

units were mapped onto the outputs using the conventional weighted-sum type

nodes with a linear activation function. When Gaussian nodes were used the

range of influence of the nodes, w\_ij1, was set to the standard deviation of

the training data for the appropriate input dimension. If the locations of

these nodes, w\_ij0, are placed randomly, then the model behaves like a

continuous version of the modified Kanerva model [PragerFallside88]. If the

locations are placed at the points defined by the input examples then the

model implements a radial basis function [BroomheadLowe88]. The first layer

of weights remains constant in both of these models, but can be also trained

using the equations of [Robinson89:sect2.4]. Replacing the Gaussian nodes

with the conventional type gives a multilayer perceptron and replacing them

with conventional nodes with the activation function f(x) = x^2 gives a

network of square nodes. Finally, dispensing with the first layer altogether

yields a single layer perceptron.

The scaling factor between gradient of the energy and the change made to the

weights (the `learning rate', `eta') was dynamically varied during training,

as described in [Robinson89:sect2.5]. If the energy decreased this factor was

increased by 5%, if it increased the factor was halved. The networks changed

the weights in the direction of steepest descent which is susceptible to

finding a local minimum. A `momentum' term [RumelhartHintonWilliams86] is

often used with error propagation networks to smooth the weight changes and

`ride over' small local minima. However, the optimal value of this term is

likely to be problem dependent, and in order to provide a uniform framework,

this additional term was not used.

Recognition Results

This experiment was originally carried out with only two frames of data from

each word [RobinsonNiranjanFallside88-tr]. In the earlier experiment some

problems were encountered with a phenomena termed `overtraining' whereby the

recognition rate on the test data peaks part way through training then decays

significantly. The recognition rates for the six frames per word case are

given in table 4.2 and are generally higher and show less variability than the

previously presented results. However, the recognition rate on the test set

still displays large fluctuations during training, as shown by the plots in

[Robinson89:fig3.2] Some fluctuations will arise from the fact that the

minimum in weight space for the training set will not be coincident with the

minima for the test set. Thus, half the possible trajectories during learning

will approach the test set minimum and then move away from it again on the way

to the training set minima [Mark Plumbley, personal communication]. In

addition, continued training sharpens the class boundaries which makes the

energy insensitive to the class boundary position [Mahesan Niranjan, personal

communiation]. For example, there are a large number planes defined with

threshold units which will separate two points in the input space, but only

one least squares solution for the case of linear units.

+-------------------------+--------+---------+---------+

| | no. of | no. | percent |

| Classifier | hidden | correct | correct |

| | units | | |

+-------------------------+--------+---------+---------+

| Single-layer perceptron | - | 154 | 33 |

| Multi-layer perceptron | 88 | 234 | 51 |

| Multi-layer perceptron | 22 | 206 | 45 |

| Multi-layer perceptron | 11 | 203 | 44 |

| Modified Kanerva Model | 528 | 231 | 50 |

| Modified Kanerva Model | 88 | 197 | 43 |

| Radial Basis Function | 528 | 247 | 53 |

| Radial Basis Function | 88 | 220 | 48 |

| Gaussian node network | 528 | 252 | 55 |

| Gaussian node network | 88 | 247 | 53 |

| Gaussian node network | 22 | 250 | 54 |

| Gaussian node network | 11 | 211 | 47 |

| Square node network | 88 | 253 | 55 |

| Square node network | 22 | 236 | 51 |

| Square node network | 11 | 217 | 50 |

| Nearest neighbour | - | 260 | 56 |

+-------------------------+--------+---------+---------+

Table 4.2: Vowel classification with different non-linear classifiers

Discussion

From these vowel classification results it can be seen that minimising the

least mean square error over a training set does not guarantee good

generalisation to the test set. The best results were achieved with nearest

neighbour analysis which classifies an item as the class of the closest

example in the training set measured using the Euclidean distance. It is

expected that the problem of overtraining could be overcome by using a larger

training set taking data from more speakers. The performance of the Gaussian

and square node network was generally better than that of the multilayer

perceptron. In other speech recognition problems which attempt to classify

single frames of speech, such as those described by McCulloch and Ainsworth

[McCullochAinsworth88] and that of [Robinson89:chap7 and

RobinsonFallside88-neuro], the nearest neighbour algorithm does not perform as

well as a multilayer perceptron. It would be interesting to investigate this

difference and apply a network of Gaussian or square nodes to these problems.

The initial weights to the hidden units in the Gaussian network can be given a

physical interpretation in terms of matching to a template for a set of

features. This gives an advantage both in shortening the training time and

also because the network starts at a point in weight space near a likely

solution, which avoids some possible local minima which represent poor

solutions.

The results of the experiments with Gaussian and square nodes are promising.

However, it has not been the aim of this chapter to show that a particular

type of node is necessarily `better' for error propagation networks than the

weighted sum node, but that the error propagation algorithm can be applied

successfully to many different types of node.