Sascha Richter and Klaus Gottschalk, Potsdam

Blackleg of potatoes

Identification with help from neuronal networks and a computer cluster

Prognosis of blackleg means an increase in quality for the agricultural product potato. Identification of the relevant parameters is decisive for the establishment of a capable neuronal network (NN). Necessary for determining the best parameters is the production of a great number of NN which requires a very high computer performance. The linking of several computers into a cluster represents a solution to this problem.

Sascha Richter is studying for a doctorate and Dr.-Ing. Klaus Gottschalk is a member of the scientific staff of the ATB, Dept. 3, Max-Eyth-Allee 100, 14469 Potsdam

Keywords

Blackleg of potatoes, neuronal networks, cluster

Blackleg in potatoes is a quality fault only measurable through destructive methods. The retail trade, processors and consumers cannot optically identify blackleg damage. A competent diagnosis system would reduce required sampling to a minimum but must also be able to deliver a satisfactory accuracy. To this background, a method based on the learning ability of NN for diagnosing blackleg in potatoes was developed at the Institute for Agricultural Engineering, Bornim e.V. (ATB). At the ATB the contents and the proportion of material with blackleg have been measured since 1995 in 240 potatoes each of the varieties Likaria, Adretta and Koretta. The information on the potato contents was collected in the online databank and used as input for the NN while the blackleg which was actually measured represented the targeted parameter. The deviation between diagnosis and measured blackleg incidence served as error and was used in the NN to correct the internal weighing of the individual neurones. In that we used a back-propagation network as NN, the error was distributed from the output neurones over the neurones in the so-called hidden-layer through to the intake neurones. The calculation of an NN based on the intake of all influencing parameters led to a very imprecise diagnosis because the influencing parameters, which themselves have no influence on the quality of the blackleg diagnosis, falsify the NN. Also it is not probable that, with only one NN, the correct pre-processing of the intake data would be discovered at the first attempt. It is more sensible, therefore, to calculate a large number of NN which differ from one another with regard to the intake parameter and pre-processing.

Neuronal network variants

So that the quality of an influencing parameter with regard to the influences on the blackleg can be evaluated, a classification of the NN was planned. Additionally, every NN was trained in different variants which differed from each other with regard to the pre-processing of the influencing parameters. The neuronal network type used by us – back-propagation – can only depict a linear influence of the intake neurones on the output neurones. Because of this, it is necessary to pre-process and to produce the variants of the NN that only differ with regard to the pre-processing. We limited ourselves to the four pre-processing operators x^-2, x^-1, x^1 and x^2. With this, an NN out of two influencing parameters 16 variants must be trained-through. The combinations of all parameters presents a large sum to the NN for calculation. The calculation effort in this case is dependant on the iteration depth of the learning process, the number of hidden layer strata and the number of neurones in the hidden layer. Added to this is the fact that a degeneration process affects potatoes and because of this the current potato varieties had to be substituted for by others with the result that the NN continually had to be adjusted to match. In practice, this meant that the NN was exposed to a continuous learning process in order to achieve a consistent quality. However, the calculation effort is so substantial that it did not seem practical with the PC technology available at that time. In that the calculations of the NN themselves were themselves completed, it occurred to us to carry out the calculations of different NN parallel to one another on several computers and to assimilate the results in conclusion. The distribution of the measurement data and the synchronisation of the results should then take place automatically. In that the software stood within a dynamic development environment, an automatic equalisation of the data between the involved computers should be suitable not only for the exchange of the data, but also serve to keep consistent the software used.

Conception of the cluster

The cluster is led through a central computer. This computer – the lead wolf – has many tasks:

- user interface
- management of cluster software
- management of data bank
- management of user software
- distribution of tasks for individual computers in cluster

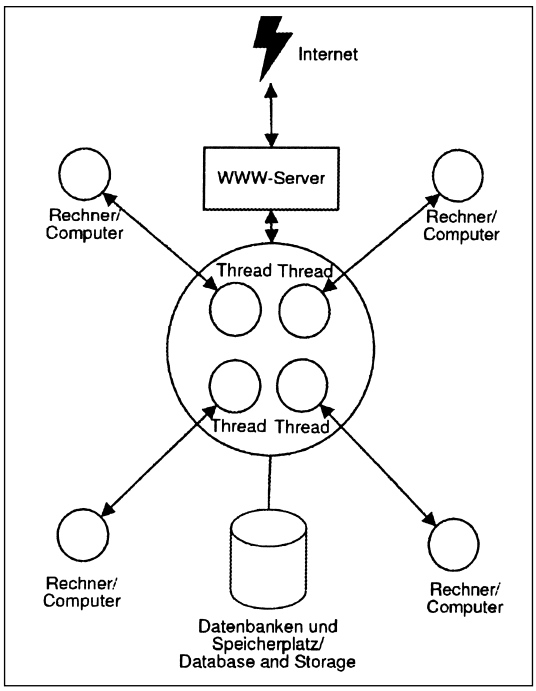

Fig. 1: Structure of the cluster

User interface

A contemporary user interface depends on the utilisation of Internet browser technology which the user of a cluster can operate very easily in that many already have experience in dealing with web browsers such as Internet Explorer or Netscape Communicator and the cluster is not any different to operate than a normal interactive website. Further, the HTTP, the protocol which transports the web contents for the web browser, already offers the necessary technologies in most cases for a successful client-server activity. The necessary extensions were developed in JAVA/RMI/Corba.

Applied methods

The communication between the computers and the modules which was written in the programming language C++ and Java was carried out via object model Corba. Corba enables transparent commands up to, and over, the computer limits.

The actual program which runs on the cluster was programmed in a development environment supported by the program language Prolog, whereby Prolog was extended via cluster-specific commands.

The program language Java was used in the area of the user interface. Linked as so-called servlets and JSP pages on the web server are the project and user management.

Applets were transmitted as programs on the user's web browser. They are able to communicate with the web server in order, e.g., to prepare existing data in graphic form.

Cluster

A cluster is a system comprising several linked-up computers which gives a substantially higher capability compared with that of a standard system. The capacity of a cluster depends on the network speed. A homogenous cluster comprises computers with the same network power and the same computing performance. It depends on the task of the cluster as to whether the cluster may deviate from these tasks without losing too much of its total capacity. Tasks which are only able to be divided in very small parts can lose a lot of the performance through a slow network connection, in that the data is exchanged over the network and because of this a bottleneck is formed. If, in a heterogeneous network, there is a great difference in capacity then it can happen that the weakest member has to deliver an important interim result without the other members being able to work further. This means that the faster computers waste time with unnecessary waiting. A cluster is suitable for tasks:

- which are able to be calculated simultaneously
- which, based on computer effort, can be broken-down well into large parts.

The communication between the central computer and the cluster computers tales place via a light processes, also called threads. These threads run in the program context of the central computer. Each of these processes communicates with a computer (*fig. 1*).

Design of the lead wolf cluster

The lead wolf cluster comprises six computers that which are networked over a 100 MBit-Hub through twisted-pair cabling. The computers in the cluster have a capacity of 400 Mhz CPU and 64 MB RAM. The, at that time, newest SuSE Linux Distribution was used as running system. A web server acted as interface between clients and cluster. We used the open source web server Apache with the Tomcat Java server page extension. In general, attention was paid in the assembly of the cluster to using economic standard components and to do without commercial software. Only open source software such as MICO was used as Corba implementation.

Distributing the data in the cluster

As central meeting point in the communication between the user and the computer cluster, the lead wolf had to deal with the following tasks:

- interaction with the user
- synchronising the cluster software
- translating the user program
- distributing the translated user program within the cluster
- steering the progress of the user program within the cluster
- bringing-together of results
- user and project management

Literature

[1] Informationen zu der Programmiersprache JAVA, JSP, Servlets und RMI: http://java.sun.com
[2] Informationen zu Corba: http://www.omg.org
[3] Informationen zu Cluster-Konzepten : http://www.beowulf.org