MULTILAYER PERCEPTRON ARCHITECTURES FOR DATA COMPRESSION TASKS

Pascal BLANCHET

LIMSI - CNRS, BP 133, 91 403 ORSAY Cedex, FRANCE

ABSTRACT

Different kinds of Multilayer Perceptrons, using a back-propagation learning algorithm, have been used to perform data compression tasks. Depending upon the architecture and the type of problem learned to solve (classification or auto-association), the networks provide different kinds of dimensionality reduction preserving different properties of the data space. Some experiments show that using the non-linearities of the MLP units may improve performances of classical linear dimensionality reduction. All the experiments reported here have been carried out on speech data.

To evaluate the cluster separation capabilities of our networks, we chose to include in the training data set only the frames of stable phonemes pronounced in different contexts by a single speaker (1). Thereby, only one frame instead of a sequence was sufficient to identify one phoneme. Also, the set of frames corresponding to one phoneme should constitute an almost convex cluster in a specific region of the data space.

The training set contained twenty three phonemes and phonetic features, including thirteen vowels, six fricatives, /m/ and /n/, and the voiced and unvoiced silences of plosives. The total number of phonetic segments was 410 which contained 2,855 frames.

THE NETWORKS

The four types of architecture that were used are shown in figure 1. Each one is a simple Multilayer Perceptron containing from one to four successive hidden layers. Each layer is fully connected to the previous and the following one. All cells, except the input ones, have a threshold which is provided by a single unit, with a constant +1 activity value, connected to the other ones through weighted links.

The experiments have been carried out on speech data. The signal has been FFT-analysed, then passed through a Bark-scaled sixteen-channel filter-bank. One sixteen coefficient frame has been computed each 12.8 ms. We then averaged each frame and preserved the mean value as a seventeenth coefficient. The frame values were finally scaled between minus one and plus one to be consistent with the networks' cell activities.

Figure 1: The network architectures used for data compression tasks.
The activity values of the cells range from -1 to +1, and the sigmoid transfer function is:

\[ s = \frac{e^x - 1}{e^x + 1} \]

where \( s \) is the output value, \( x \) is the weighted sum of input values and \( k \) is a positive constant coefficient.

THE TRAINING

The training phase was divided into a sequence of cycles. Each cycle consisted in presenting to the network one randomly chosen phoneme before modifying the weights according to the Back-Propagation algorithm (5,6). Thus, one cycle corresponded to twenty-three frame presentations and one weight modification. During the training, the weights were set to higher values as the number of cycles increased, and the training phase terminated when the output square error stopped decreasing.

The architecture we used, containing three, four or five layers, needed to be initialized carefully before starting the training phase, otherwise the networks would fail in learning anything. The weights were set to higher values as the links were closer to the input layer, so that the cell activity values were about 0.1 when a frame was presented to the initial network.

LINEAR NON-DISCRIMINANT COMPRESSION

In this first experiment, the network had one reduced hidden layer containing two cells and was used for auto-association (network A). With the auto-association training, the desired output is exactly the input data. Then the output space is the same as the input one, and the network has to give a response which is as close as possible to the input vector. When the hidden layer contains two cells, the image of the data set in the output space is in a surface. Thus the network has to learn the surface that best matches the data set. When the units have linear transfer functions, the input data are merely projected in the reduced layer, and the image of this layer in the output space is a plane. Then the best match is given by the Principal Component Analysis (PCA) (2).

Here the transfer functions were sigmoids but the cell activity values were taken in the most linear part of the curve. So the planar representation we obtained after 700 learning cycles is equivalent to a Principal Component Analysis (Figure 2).

To obtain this picture, the activity plane of the reduced layer was divided into a 30x30 grid. As a frame was presented to the network the couple of cell activity values found in the reduced layer were the coordinates of the frame in the plane. As several frames fell into the same square of the grid, this square was labelled with the frames' labels according to a majority voting criteria.

<table>
<thead>
<tr>
<th>API</th>
<th>LIMSI</th>
</tr>
</thead>
<tbody>
<tr>
<td>D</td>
<td>+</td>
</tr>
<tr>
<td>a</td>
<td>A</td>
</tr>
<tr>
<td>i</td>
<td>I</td>
</tr>
<tr>
<td>y</td>
<td>U</td>
</tr>
<tr>
<td>e</td>
<td>(</td>
</tr>
<tr>
<td>oe</td>
<td>E</td>
</tr>
</tbody>
</table>

Table 1: API phonetic codes and corresponding LIMSI codes.

LINEAR DISCRIMINANT COMPRESSION

In this section, we describe how to design a network which will try to learn discriminant axes instead of principal axes. Such a network needs to be trained for a classification problem, so the projection to the first hidden layer will tend to preserve cluster separation. In this experiment, the network output layer contained twenty three cells. The desired outputs were phonetic labels coded with twenty three values, so that each output cell was at one for a given phoneme and minus one for all the others.

With a one hidden layer perceptron, like the previous one, the data would be projected in the hidden layer and then separated in the output layer by a single hyperplan for each phoneme cluster. But we know that this type of separation does not suit any cluster distribution. So we added to our network a second hidden layer containing 20 cells, which enables the MLP to place boundaries around each projected cluster to perform a good classification (network B).

Three MLPs were studied, containing two, three or four cells in the first hidden layer. About 6,000 training cycles were necessary to converge to the minimum output square error and the best classification rate. To measure the amount of information that was lost in the reduced layer, the results were compared to those obtained by a simple two-layer perceptron containing 20 cells in the hidden layer (Table 2). As our goal was to measure the networks' capability in finding good discriminant axes of the training data set, all the results reported here concern this training set.

With three or four cells in the first hidden layer, the networks obtained good results. Most of the phonemes had a recognition rate over 90%. With four cells, only U was at 88% and V at 69%. With three cells, the least score was obtained by Z with 77%. Those results show that each phoneme cluster was located in a specific region of the projection space. This means that the MLP found well discriminant subspaces with three and four dimensions.

On the other hand, with two cells, the recognition rate was very poor. Some phonemes were completely ignored by the network, which means that they overlapped with other phonemes in the activity space of the reduced layer.
Table 2: percentage of good classifications of frames and segments. A segment is said well classified when more than half of its frames are well classified.

**NON-LINEAR DISCRIMINANT COMPRESSION**

The reason why the previous network did not find good discriminant axes may be that there is no plane on which the phoneme clusters could be projected remaining separated. Therefore, to get a discriminant planar representation of this speech data we need to introduce non-linearities before the dimensionality reduction.

To solve this particular problem we built a network with one more hidden layer containing twenty non-linear units between the input layer and the reduced one (network D). An MLP with such architecture is able to transform the data distribution in the first hidden layer so that the phonetic clusters can be projected on a plane remaining separated.

This network was trained with the same data set during 7,000 cycles. It obtained 88.5% of good frame classification and 94.9% of segment classification. Compared to the previous MLP containing two cells in the first hidden layer, the error rate was reduced from 29% to 11.5%.

It seems that there exists a large number of solutions to the problem of placing phonetic clusters in a two dimensional space. Each solution can be found by a network as it falls into a local minimum of the output square error. The result of the training is highly dependent upon the weight initialization.

In our attempt to obtain a better discriminant planar representation we tried to reduce the number of possible cluster distributions by reducing the number of cells in the first hidden layer from twenty to fifteen units. Fortunately the first experiment with this network gave a more relevant representation (figure 4). But this was not the case for other initializations. However it is interesting to see how different those planar representations can be on figure 3 and 4.

**Fig 3:** A two dimensional map obtained by a non-linear discriminant compression.

This two dimensional compression can be visualized on figure 3. As we can see, each phoneme cluster is located in a specific region of the plane. Though most of the cluster distribution in this plane seems consistent with what we know about speech data, some of the topological properties of this representation are peculiar. The silence of plosive P which has a low energy is close to A which is a high energy phoneme, and W is placed between E and =. Those topological aberrations are probably due to the high degree of freedom provided by the first hidden layer where the phonemes can be placed almost anywhere, and the fact that the classification training never constrains the MLP to preserve topology.

Another problem appeared when we used the same architecture with the same data but with a different weight initialization. The planar representation obtained by this new network was very different of the first one, and contained also topological mistakes.

Finally, in our experiments we observed that, reducing the number of cells in the first hidden layer does not seem to improve the topological consistence. Instead, the networks learn harder or fail in learning a good discriminant representation.

**NON-LINEAR NON-DISCRIMINANT COMPRESSION**

The last step of our work was to try the same type of architecture, with non-linearities above and below the reduced layer, for an auto-association task (network C). As with the network architecture A, the image of the two-dimensional activity space of the reduced layer in the output space is still a surface. But this time, the non-linearities contained in the hidden layer between the reduced one and the output one, enable the network to curve this surface.

Minimizing the output square error during the Back-Propagation training will cause the MLP to curve the surface to fit as best as possible the training data cluster.
Moreover, the non-linearities of the first hidden layer enable the network to build any cluster distribution in the reduced layer, as the previous experiments with network D has shown. Then the MLP is able to find the best planar representation according to the shape of the output surface and its position among the training data clusters.

Several networks were trained with this kind of architecture. The convergence rate was very slow compared to our previous experiments. 26,000 learning cycles were necessary to produce the planar representation of figure 5. But the output square error reached the Principal Component Analysis square error after a few hundred cycles. And then decreased slowly to become less than half of this value at the end of the training phase.

It is also noticeable that the best results and the fastest convergence rate were obtained by a network containing four hidden layers (a five-layer perceptron). Those containing only one hidden layer below the reduced one, even with thirty cells, converged twice more slowly, which implies a prohibitive computation time to reach the solution.

Another interesting fact is that all the representations we got were similar and seemed to be a slightly modified PCA instead of a new compressed representation. Nevertheless, some topological features, that are not in figure 2, appear in figure 5: There is a gap between vowels and consonants, the phonemes U, I and ) are well separated from each other and from the nasals M and N, which is not the case in figure 2. On the other hand, the phonemes O and ) are overlapping. To decide whether the non-linear compression improves the PCA or not, besides the reduction of the output square error, we computed the mean overlapping rate among the phonetic clusters of the two planar representations.

This mean overlapping rate is the percentage of frames that fall into a square which has a different label. For consistency, the two planar representations had different grids, so that the number of labelled squares were almost the same for each plane. A 24x24 grid was used for network A while a 20x20 grid was used for network C. The overlapping rate was 29% with the linear auto-association and 20% with the non-linear one. Then, it seems that non-linear compression gives more information than PCA in this two dimensional representation.

Another point may be discussed: The previous planar representation obtained by an unsupervised auto-association training is suggestive of the topological maps obtained by the Kohonen Feature Maps (4). However, KFM performs vector quantization, not a continuous dimensionality reduction found with MLP. To compare those methods, a 15x15 Feature Map was trained with the same data set. After a thousand cycles, the mean square distance between training samples and references was less than the half of the output square error of network C. Moreover, the equivalent overlapping rate was 5.3%, which is much better than with the reduced layer of network C.

These results show that using vector quantization is better to compress the data for speech recognition tasks. But the planar representations provided by KFM are discrete. And the lost of continuity entails lost of topological information. Therefore, MLPs are more interesting for data visualization.

CONCLUSION

In the previous parts we described how to design a Multilayer Perceptron for four types of dimensionality reduction: linear/non-linear and discriminant/non-discriminant. The previous results show that using MLP is another way to find a continuous compressed representation of a data space. The main problem of this approach is the lack of mathematical theory for convergence proof (It is difficult to demonstrate anything about networks with such architectures as B,C or D). Therefore, we do not know how to avoid getting stuck in a local minimum during the training phase. The consequence is that a network can find a new solution at each trial. However, non-linearity and gradient descent seem to be useful for dimensionality reduction. The non-linearity can improve some of the classical linear algorithms, and gradient descent is at the moment the only way to compute such representations.

REFERENCES