Chapter 7

RAN, HOLISM, and HOLISTIC COMPUTATIONS

7.1 Introduction

There are two different views of holism that concern Distributed Representations (DRs) of sequences developed by the Recurrent Autoassociative Networks (RANs were introduced in Chapter 6). The first one is analytical and is related to organizing elements (here, DRs) into a holistic system under a suitable arrangement (Esfeld 1998). This view regards a set of DRs developed by a given \( RAN_L \) as a holistic system \( H \), where the arrangement is imposed by \( RAN_L \) as a recogniser/generator of the elements of \( H \). The other view goes one level below, considering the elements (DRs) alone, although these two views are related as we will see later. This view is associated with a treatment of wholes (here, DRs), which are otherwise not accessible to rational analysis (Esfeld 1998), that is, holistic transformations, or holistic computations – transformations that directly act on the wholes. Considering distributed representations of sequences, structures, etc., Chalmers (1990), Chrisman (1991) and Hammerton (1998a), among others, additionally specified that holistic computations act on DRs as a whole, instead of first decoding them, applying classical (symbolic) techniques on the decoded sequences/structures, and then recoding them again.

I propose that a combination of a set of RAN-developed DRs and holistic computations offers a holistic program. The first part of this program – a holistic system built with RANs – provides the objects of holistic computa-
tions, and it was presented in Chapter 6 from a computational perspective. Here I will only define it more formally, from the holistic standpoint. This chapter will focus on the other view of holism: treating wholes – sequences represented with their DRs – and holistic computations on these DRs.

Holistic operators can be implemented with various connectionist architectures. Given, first, the static nature of the DRs developed by the RANs and second, holistic operators with fixed numbers of parameters, simple, static NNs may be used to implement holistic operators. For example, very appropriate for this purpose are the supervised Multi-Layered Perceptron (MLP) (Rumelhart, Hinton & Williams 1986) and the Radial-Basis Function Networks (Haykin 1994, Reed & Marks II 1999), or more biologically-plausible models such as the ART(MAP) networks (Carpenter & Grossberg 1992), etc. On the other hand, if a given holistic operator has a variable number of parameters, then dynamic Neural Networks such as the Simple Recurrent Networks, could be used, too.

RAN-developed DRs are produced in a systematic way, and therefore I expect that holistic computations on them can be learned easily, using just a subset of them. Nonetheless, in connectionism, due to the computational complexity of the problems, theoretical capacities cannot always be realised computationally – see, for example, the difficulties experienced with RAAM-based holistic computations (Hammerton 1998b), which necessitates more extensive experimental work supporting claims of capacities. Accordingly, the purpose of this chapter is to explore experimentally a few basic holistic operators which act upon DRs developed by the RANs.

The operators which will be presented here are simple in structure, with one or two input parameters, for which I chose the MLP model due to its proven capacity to represent arbitrary continuous mappings. In particular, three types of holistic operators will be explored. The first type is token extraction, implemented with unary and binary operators. The unary holistic operators presented here are a set of neural networks that extract from the DRs the symbols at a pre-specified position \( p \) in the original string (section 7.4). The more-complex binary operators extract symbols at a position entered as an additional parameter. The second type of operators is even more complex – it transforms one DR into another. This operator reverses strings represented by their DRs (section 7.5).

To provide stronger evidence, the above operators will be tested on the distributed representations of the following sets of strings: (1) a small set of phonologically represented English words; (2) a larger set of English words, (3) a complete set of numbers represented as strings of digits (section 6.4.3 for details).
7.2 Holistic Systems and Holistic Computations

This section will present holism from two main standpoints: holism as referring to the view that elements of an organised system are significant only in this system, and holism in terms of treating elements as wholes. Also, RANs and RAN-developed distributed representations will be considered according to these two viewpoints.

Holism and Holistic Systems

According to the Oxford English Dictionary the term holism is “coined by Gen. J. C. Smuts (1870 - 1950) to designate the tendency in nature to produce wholes (i.e. bodies or organisms) from the ordered grouping of unit structures”. Similarly, Block (forthcoming) defines holism as “the view that parts of a system have significance mostly in virtue of their interrelations with other parts”, that is, in a holistic system.

A very comprehensive study on holism from a philosophical point of view is provided by Esfeld (1998), who construed it in terms of a holistic system of constituents, which in turn are characterised with certain holistic relational properties. A relational property is holistic iff: “(1) it belongs to a family of properties which make something a constituent of a holistic system \( H \) in case there is a suitable arrangement, and (2) Nothing can instantiate this property unless there are actually other things together with which this thing is arranged in such a way that there is an \( H \).”

The conclusion from this short review is that a holistic system (of representations, in our context) is characterised by the mutual dependencies among its constituents (here, distributed representations) in such a way that these constituents are considered as a part of the system iff they co-exist and are treated in a similar way with respect to certain set of relational properties. The constituents themselves have a specific internal structure, which may be analytically difficult to interpret, but nevertheless it must conform some rules.

For example, the following set of numbers represented as strings of digits \( N = \{"0000","0001" \ldots "9999"\} \) makes a holistic system under the arrangement “all strings that consist of four digits”. However, if we make a new set containing all, but a few elements of \( N \), then this new set will not be a holistic system under the previous arrangement, since there will be strings that will comply with the arrangement, but will not belong to the system. Nevertheless, this new set may well be a holistic system under some other arrangement(s). Also, if we consider just one string as a system, e.g.,
“1111”, then this “system” will be not a holistic system since it will have one element only, an atom, and no properties will be able to be instantiated so that they specify a holistic system.

**Holism and Holistic computations**

Other dictionary sources define holism as “the process of focusing attention directly on the whole and its characteristics as a whole, without any recourse to consideration of its parts” (Heylighen 1999). In the same vein, Esfeld (1998) also noticed that holism is often associated with “a treatment of wholes that is not accessible to the conceptual tools of rational analysis”, that is, operations on wholes (here, distributed representations) which otherwise do not have reasonable analytical means of processing. Correspondingly, these operators are called holistic operators (Chalmers 1990, Chrisman 1991, Hammerton 1998a).

Holistic computations need a proper set of objects to act on – all these objects have to follow a certain organisation in order to be treated in the same way. As we just saw, the elements of a holistic system follow such an arrangement, and therefore they are candidates for holistic computations. However, not all holistic systems provide such objects: there are systems consisting of complex objects which can not in generally be treated as wholes, e.g., symbolic systems and expressions. Holistic objects must be meaningful as a whole, unlike symbolic expressions which need to be interpreted in order to derive their meaning. Distributed representations fulfil this requirement, and in particular RAN-developed DRs are proper objects for holistic computations: they represent the corresponding sequences as a whole and they form a holistic system. Therefore, I call them holistic representations – a set of static distributed representations that form a holistic system.

Holistic operators are meant to transform holistic representations. A general holistic operator $H(X) = Y$ acts upon the elements of a holistic system $L_1$ – vectors $X = (x_1, x_2 \ldots x_{|H|})$, $X \in L_1$. In generally, the output domain $Y$ of $H$ is also a vector that may belong to the same or other holistic system $L_2$.

For example, given the above hypothetical holistic system of numbers $N = \{"0000","0001" \ldots "9999"\}$, we can develop another holistic system $M$ of DRs of the elements of $N$: $M = \{DR_{N_i}\}_{i=1}^{\lfloor N \rfloor}$, which will contain holistic representations. Then in order to develop some sort of a simple mathematical system, we have to develop holistic mathematical operators, such as comparison $H_<(DR_{N_i}, DR_{N_j}) = \{-,0,+\}$, addition $H_+(DR_{N_i}, DR_{N_j}) =$
$DR_{N_k}$, subtraction $H_\lambda(DR_{N_i}, DR_{N_j}) = DR_{N_k}$, and so on, performing the corresponding mathematical operations.

From a cognitive point of view, holistic operators should be implemented with connectionist models. They can also be implemented with general mathematical models that transform multidimensional data, but then one of the purposes of holistic modeling – modeling human cognition with cognitively plausible architectures – is lost. From a practical standpoint one might directly operate on symbolic structures, which today perform better than distributed models.

Neural Networks are particularly good at learning to perform complex transformations between different distributed representations, which has been exploited successfully in various “low-level” (cognitive) tasks, such as pre-processing in vision and speech. The distributed data in these problems is either the direct source signal, or a vector of features representing this signal. In that respect, there is no theoretical restriction on the type of holistic operations which could be developed, similarly to the capacity of the MLPs and other neural networks to represent and learn any input-output mapping. Still, it is possible that computational restrictions might be imposed by the data or due to some concrete implementations of $H$, such as noise (observed, for example, in the circular convolution operators) or limited capacity (e.g., an MLP with an insufficient number of hidden neurons).

Holistic operators (connectionist models) may be intended for an unlimited number of holistically represented basic elements (sequences, structures). However, the operators can be developed by experiencing (that is, training on) limited amounts of data only. Therefore, holistic operators must rely on the systematic organisation of the holistic representations they work on and on the capability of the neural networks to generalise on unseen data. This strongly relates the problem of holistic computations to the problem of developing proper holistic representations in a holistic system, the two problems representing two different perspectives on the core concept of holism.

**RANs and holistic representations**

If we turn to the RANs as a tool for developing a set of holistic DRs in a holistic system, for a given instantiation $RAN^L$, the set of DRs it develops and decodes constitutes a holistic system $L$, and $RAN^L$ entirely represents this system\(^1\). Elements of $L$ – DRs of sequences $s$ – are characterised with

---

\(^1\)Note that here $L$ denotes the entire holistic system, not only a subset of it, e.g., a training subset.
the following relational property:

\[(DR \in L \Leftrightarrow \exists s : (RAN(s) = DR) \& (RAN^{-1}(DR) = s))\]  (7.1)

where \(RAN(s)\) denotes encoding of a string \(s\) and \(RAN^{-1}(DR)\) denotes decoding of a representation \(DR\).

A distributed representation \(DR^L_i\) taken out of this system is just a vector of numbers. Vectors \(DR^L_i\) are meaningful only within this holistic system, being developed and decoded by the \(RAN^L\). In addition, features (values) \(DR^L_{ij}\) which constitute \(DR^L_i\) do not have a meaning by themselves unless they are bound into that vector \(DR^L_i\) and interpreted within this holistic system \(L\) by the \(RAN^L\) or by a holistic operator \(H\) trained to act upon the set of DRs from \(L\).

The interpretation (meaning) of a vector \(DR^L_i\) depends on its features \(DR^L_{ij}\), but their slight change would not alternate its meaning dramatically due to the generalisation capability of the holistic operators – a general connectionist property. Even more, due to the very important property of RANs that they develop DRs in a very systematic manner, DRs of sequences unseen during the training process of the holistic operator should be developed and treated in the same way as DRs of experienced (training) sequences.

To show that this is valid for the set of DRs developed by RANs, let’s consider the experiments on developing distributed representations of numbers presented in subsection 6.4.3. First, I showed that if an RAN is trained on a considerable subset (10-30%) of a range of numbers represented as a sequence of digits [“1” … “9999”], then the network will treat the rest of the sequences in the same way it will treat the sequences from the training set, developing understandable (i.e., decodable) DRs of these testing sequences. Second, as I will show in the following experimental part of this chapter, holistic operators trained on DRs from the training subset well perform the same operations on the DRs of the testing subset of sequences, that is, these holistic operators treat novel DRs that are part of the holistic system of DRs in the same way they treat the experienced sequences. Hence, RANs develop a holistic system of DRs, given enough training data to provide properties characterising the holistic set.

### 7.2.1 Earlier studies on holistic computations

The line of research on holistic computations roughly follows the line of development of various connectionist architectures aiming at developing distributed representations, as already presented in section 6.3.
The RAAM model was used for this problem most frequently, providing intuitive methodology for developing DRs and the vehicle for the notable work on holistic computations by Chalmers (1990). This work studied the classical problem of converting holistically DRs of active sentences into DRs of passive sentences, demonstrating excellent learnability and generalisation on a relatively small dataset (250 sentences built out of 13 distinct feature-encoded items). However, those results were questioned in a more recent work by Hammerton (1998b), who did not find the initially claimed 100% learning and generalisation.

A sequential version of the RAAM — the SRAAM model (Blank et al. 1991) — was explored more thoroughly for various holistic operators, such as detecting features in DRs; extracting tokens from DRs, and a simple syntactic transformation. The performance of the holistic operators was reasonably good, with some 5-25% error in the different task. Since the purpose of study in this chapter on holistic operations is to demonstrate that RAN-developed DRs allow holistic computations, I will use similar experiments, presenting a few simple operators: holistic symbol extraction operators, and holistic inverse transformation operators.

The theoretically very interesting problem of holistic logical unification was studied by Weber (1992). He used an alternative version of the circular convolution operator to encode logical terms, which he claimed to perform more precisely than the Plate (1994)’s implementation. Then, he trained an MLP to unify terms. The system reportedly performed very well.

An equally interesting and important application for holistic computations is syntactic parsing. In spite of the success of the classical symbolic parsers, they feature poor error recovery due to the rigid rules they normally apply. Connectionist models, on the contrary, are flexible due to the smoothness of the mapping functions they acquire. However, since the parsing process makes extensive computational demands due to the large amount of information processed, only very simple experiments have been reported, e.g., (Ho & Chan 1999) where a number of connectionist parsers have been tested on a very small data set with 112 sentences generated by a simple grammar. Most of the parsers explored there are based on a symbiosis between the RAAM model which is usually used to encode the parse trees and a SRN used as a sequential predictor (Reilly 1992, Berg 1992). A completely holistic parser uses also the SRAAM network to encode the input sequences and then an MLP to holistically map those representations of into holistic representations of parses (Ho & Chan 1994).

Those studies show that syntactic parsing is a promising application for holistic computations, and that research in this area is well worth exploring.
Significant success here would bring substantial evidence in favour of the hypothesis that languages are learned by children with general learning devices using limited linguistic input, but not by using innate language knowledge (according to the Universal Grammar theory by Chomsky (1965)). Yet, since the scope of this thesis is restricted only to lexical modeling, this problem will be explored in the future.

7.3 Implementation of Holistic Computations

The previous section already noted the basic principles of holistic computations. A holistic operator normally consists of two basic blocks: (1) interface modules that produce and interpret holistic representations and (2) the holistic operator itself — usually a static neural network that transforms the input holistic representations into output holistic representation(s) or some other type of data. This implies the coupling of two systems — a generator of holistic representations and a transducer.

As noted, most of the earlier holistic systems used the RAAM as a generator of holistic representations. In this work, the RAN will be used for this purpose instead. The experimental work in the previous section demonstrated that RANs can reliably produce DRs for a substantial number of sequences and a cascade of RANs can develop DRs for hierarchically structured sequences. As for the transducer, an MLP is usually used due to its proven capacity to encode arbitrary mappings given enough hidden neurons. I will also use the MLP.

An example scheme for a holistic operator is given in Fig. 7.1. The operator there is an MLP and it is given in the middle top of the figure. Its input and output data are produced/interpreted in this case by two different RANs, whose DRs for convenience are also stored in an input and output databases, as explained later.

Training

The complete system is trained in the following manner: Firstly, RANs — generators of holistic representations — are trained to encode and decode sequences from the training databases. Usually the RANs learn this task with some error, that is, some strings would not be learnt perfectly. During the evaluation phase, in order to assess independently the performance of the holistic operator, it is better that this erroneous data is filtered out and not used.
Figure 7.1: Diagrammatic representation of a holistic operator $H$ that maps an input holistic DR produced by $RAN_{inp}$ into a DR processed by $RAN_{out}$. The central “bridge” database provides the input/output mapping in terms of sequences, which are further encoded into DRs with the correspondent RANs or are used to access the correspondent DRs from the input and output databases with DRs.
Later, however, the incorrect representations may be used, because although imperfect, they represent the correspondent sequences to some degree.

The second phase of developing a holistic computation system is to train the holistic operator $H$. Since the operator in this case is a regular MLP, I will not go into details about its training. More interesting now is how to access the required input/output training data. For that purpose, two approaches might be used. The first one is simply to use the generators (RANs) any time a representation is required to be encoded or decoded. Although this is the more plausible way, in order to speed up the training process, a second approach based on generating databases with holistic representations is better (as in Figure 7.1). Then, during training, the required input and output exemplars (DRs) are drawn from those databases.

**Evaluation**

As usual, the system is evaluated first, during the training process on the training set after each training epoch, and secondly, after that on a testing set, in order to test the generalisation. As noted earlier, to assess the performance of the holistic operator independently on the encoding/decoding system, only correctly encoded and decoded sentences are used.

The output of the holistic operator is tested for correctness not by computing the distance between the desired vector and the output vector produced, but rather by using the correspondent RANs to decode the DRs produced and then check if the decoded sequences are as expected. Similarly to the evaluation of the RANs, two types of errors are measured: token and string errors; the latter being normally larger than the former because a string error is counted any time an incorrect token is produced, and both errors are measured as a percentage of the total number of tokens/strings.

**7.3.1 Data: RAN-developed holistic representations**

To begin a study on holistic computations, a proper database of holistic DRs must be created. For this purpose, two sets of words were used: a smaller one $A$ with 500 phonologically represented English words and a larger one $B$ with 10,000 words. The larger database was developed in order to test the scalability of the holistic operators—an issue that concerns connectionist modeling. Furthermore, in relation to one of the holistic operators reported later in this paper, each of these databases was extended with the reverse of the original strings. This enlarged the databases only by some 20-30%,
because most of the inverse strings already existed as words in the original corpora.

The words in both corpora are built out of 45 phonemes and have mean length of 4.2 (±σ = 1.65, min = 2, max = 12) and 5.0 (±σ = 1.26, min = 2, max = 13) phonemes, correspondingly.

The first corpus of DRs was developed by an RAN with 150 hidden neurons, which is also the size of the DRs. Some 10% of the strings could not be perfectly encoded and decoded – usually with one misspelled phoneme – and therefore not included in the training database, as discussed earlier. The resulting database $A_{train}$ contained approximately 650 sequences and their DRs. This database was used for training of the holistic operators. Further, the same RAN was tested for generalisation on a larger corpus containing unseen during the training words, which resulted in a second database $A_{test}$ with the DRs of other 1,350 words. $A_{test}$, in turn, was used for testing the generalisation of the holistic operators.

With respect to the larger database, another RAN with 300 hidden neurons developed DRs of some 5,250 words, which resulted in a second training database $B_{train}$. This RAN was also tested for generalisation on even larger word corpus, which in turn resulted in a second testing database $B_{test}$ containing nearly 12,000 DRs. The second database was used to examine the generalisation of the correspondent holistic operators.

Further, to explore holistic computations on a more complete set of DRs of strings, the RAN which developed DRs of numbers ("1" … “9999") (see section 6.4.3) was used to develop a database $M$ of DRs of these numbers. This database was also split into two parts: $M_{Train}$ and $M_{Test}$, each of them containing 50% of the DRs.

An interesting question concerning both the RANs and the holistic operators is the organisation of the RAN-developed DRs. Some knowledge about this would predict the performance of the holistic operators, and also might give some hints about how such a structured data might be organised in humans.

The DRs are high dimensional vectors and difficult for humans to comprehend in their original form. More useful information can be extracted by cluster analysis, which I did on the smaller set of DRs of words. Figure 7.2 shows three fragments from the dendogram containing this cluster analysis. The strings there are apparently clustered by length, initial symbols and final symbols, similarly to the organisation of DRs developed by the SRAAM model (Kwasny & Kalman 1995); see the discussion section for analysis. In the following experiments, clues about such an organisation will be sought, too.
7.4 Extracting tokens from DRs

Extracting tokens from strings is a very intuitive symbolic operation. Still, if we give ourselves such a task as "extract the letter at a position $p$ from a string $w$" we will surprisingly find out that this 'easy' operation is easy for the initial and final symbols, but not for the more central ones. Now, let us develop a holistic operator that does this. This study will have a twofold goal: Firstly, to probe very simple holistic computations on the DRs developed by the Recurrent Autoassociative Networks and secondly, to study the organisation of these DRs.

7.4.1 Holistic operator 1: Extracting symbols at a specific position

The first set of experiments involved training an MLP to associate the DRs of a word $w$ with the phoneme $w_p$ at a specific position $p$, that is, to extract the symbol $w_p$. Note, that this is a unary operator with an only argument the input DR.

The very first task was extracting $w_1$. For this purpose, an MLP with 100 hidden neurons was trained on this task using the $A_{train}$ database. This task was learned by the network easily with very small error. For less than 10 training epochs, the network learned almost perfectly to extract
the first symbol (Table 7.1, operator $E_1$). The network performance on the testing database $A_{test}$ was also good: only 1% of the symbols were incorrectly produced.

Next, another similar experiment: extracting the second symbol from DRs of strings. It was also learned almost perfectly (Table 7.1, operator $E_2$), but this time the generalisation was slightly worse – 4.5% error. When a third operator $E_3$ was trained to extract the third letter, its performance started to decline on the training words (0.6% err) and the generalisation was poorer, too. However, a forth operator $E_{Last}$ trained on extracting the last symbol from the DR of a string performed better than $E_3$, just slightly worse than $E_1$.

Given the good performance and generalisation of these simple holistic unary operators, one can conclude that the sequential items are easily accessible from the DRs and that these DRs are developed systematically with regard to the sequential structure of the sequences. Furthermore, the decreasing performance of the operators $E_i$ as $i$ moves toward the centre of the strings, is an indication that the DRs developed by the RANs are organised hierarchically into clusters keyed on their initial and final symbols. This conclusion fits with the outcome of the DR clustering analysis, presented in the previous section (Fig. 7.2). The strings there are also hierarchically grouped and this organisation is primarily based on their initial symbols, but also, on their last symbols in some cases. Still, further experiments with more complex tasks are necessary in order to confirm this hypothesis. 

<table>
<thead>
<tr>
<th>Err (%)</th>
<th>Holistic Operator (100 hm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>$E_1$</td>
</tr>
<tr>
<td>$A_{train}$</td>
<td>0.2</td>
</tr>
<tr>
<td>$A_{test}$</td>
<td>1.0</td>
</tr>
</tbody>
</table>

Table 7.1: Performance of holistic operators $E_1$, $E_2$, $E_3$ and $E_{Last}$ trained on extracting symbols at specific positions (1, 2, 3 and the last one) from DRs of sequences. The performance is given for the training and the testing databases.
7.4.2 Holistic operator 2: Extracting symbols at a variable position

Having implemented simple unary holistic operators, I examine a more complex, binary holistic operator that extracts the symbol at a position entered as a second parameter to the network. Two versions of such an operator were studied: (a) an operator \( E_B(DR_w, p) = w_p \) where the second parameter \( p \) is an offset from the beginning of the string (e.g., (7.2)) and (b) an operator \( E_E(DR_w, p) = w_{|w| - p + 1} \) where \( p \) is an offset counted backward, from the end of the string (e.g., (7.3)).

\[
E_B(DR_{net}, 1) = 'n' \; ; \; E_B(DR_{net}, 2) = 'e' \; ; \; E_B(DR_{net}, 3) = 't'
\] (7.2)
\[
E_E(DR_{net}, 1) = 't' \; ; \; E_E(DR_{net}, 2) = 'e' \; ; \; E_E(DR_{net}, 3) = 'n'
\] (7.3)

By studying the second operator \( E_E \), evidence was sought for the hypothesis that the DRs of the strings are organised both by initial and final tokens: given the nature of the DRs discovered earlier, the performance of those holistic operators was expected to decrease as the position \( p \) advances toward the middle of the strings.

In this experiments, MLPs with 151 input neurons were employed – one input neuron more represented the position of the token targeted. The position was encoded at the extra neuron as a neural activation - from 1 to 5. Also, the hidden layer was set to 100 hidden neurons. The training was organised in epochs, in the course of which words and targeting positions and correspondent phonemes were selected randomly. Both networks in this experiment were trained on the \( A_{train} \) database and tested for generalisation on \( A_{test} \).

In overall, \( E_B \) produced correctly 91.87% of the symbols, which is worse than the average performance of the first-order extractors, but the task is more complex, too. The performance was further analysed for the influence of the targeting position. As expected, it decreases as the position moves toward the beginning of the string, starting from 4.5% for the first symbol, 5.2% for the second one, and so on (see Table 7.2, first row). However, the error increases almost three times as the targeting position moves forward.

Next, similar to the unary operators, the network \( E_B \) generalised well on \( A_{test} \) with mean performance of 90.69%. Also, factors influencing performance were similar to these in \( A_{train} \) – the error increased for positions further inside. And this is consistent with the idea that the organisation of the DRs is hierarchical and that this hierarchy is keyed on the earlier symbols in the strings. The cluster analysis and the relatively better performance of \( E_{Last} \) than \( E_3 \) were an indication that in this hierarchy both the
Table 7.2: Performance of MLPs trained on extracting symbols from DRs of sequences at positions entered as a parameter. Note that parameter $p$ in $E_B(DR_w, p) = w_p$ counts the position from the beginning, while in $E_E(DR_w, p)$, it counts the position from the end. The mean error and the error dependency on the position $p$ are given.

<table>
<thead>
<tr>
<th>Holistic Operator</th>
<th>Data base</th>
<th>Mean Err(%)</th>
<th>Error/Position(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E_B(DR_w, p) = w_p$</td>
<td>$A_{train}$</td>
<td>8.13</td>
<td>4.5 5.2 9.9 14.6 15.3</td>
</tr>
<tr>
<td>100lm, $A_{train}$</td>
<td>$A_{test}$</td>
<td>9.31</td>
<td>6.6 7.9 11.4 13.6 12.9</td>
</tr>
<tr>
<td>$E_E(DR_w, p) = w_{</td>
<td>w</td>
<td>-p+1}$</td>
<td>$A_{train}$</td>
</tr>
<tr>
<td>100lm, $A_{train}$</td>
<td>$A_{test}$</td>
<td>7.6</td>
<td>4.8 8.7 8.1 9.4 13.4</td>
</tr>
</tbody>
</table>

initial and the final symbols are more prominent than those in the middle of the strings.

Therefore, this hypothesis was tested with the other binary operator — $E_E(DR_w, p) = w_{|w|-p+1}$ — that extracts symbols at a position $p$ counted backward from the end. As expected, the performance of this operator was also position-dependent: see Table 7.2, 3rd and 4th data rows. The error increases as the position $p$ goes toward the middle of the string, which means that the DRs are keyed not only on the anterior symbols, but also on the posterior ones.

### 7.5 Holistic operator 3: Reversing strings

Extracting symbols from DRs is rather simple holistic operation. A more interesting transformation with direct psycholinguistic application is, for example, to transform a sentence from an active into a passive form (Chalmers, 1990), or to transform nouns from singular into plural form, etc. In order to produce the correct word forms independently of the word regularity, those transformations need to extract a number of features from the DRs. In this respect, another interesting transformation is associating the DR of a string $w$ with the DR of its inverse counterpart $\bar{w}$: $R(DR_w) = DR_{\bar{w}}$. Note that this is not simply inverting the order of the features of a DR, but rather, a complex transformation that produces the DR of another string — a holistic transformation of strings into strings.
Table 7.3: Performance of $R(DR_w) = DR_{\bar{w}}$ - a holistic operator reversing strings presented as DRs - on the training and testing databases. The table contains the percentage of incorrectly reproduced phonemes by the decoding RAN: mean error and the error at the different positions of occurrence. Note that the error in the beginning and the end of the strings is smaller than the error in the middle.

Similarly to the previous experiment, the holistic operator is an MLP and the training database is $A_{train}$. However, this time both the input and the output layers contain 150 neurons. The training was organised in epochs, in the course of which all patterns were presented and the MLP was trained to associate the DR of the input string with the DR of the inverse string. After every epoch, the performance of $R$ was tested by using the decoding RAN and comparing the decoded strings with the expected inverse strings. The reported performance gives the percentage of correctly decoded symbols. Finally, the testing database $A_{test} $ was used to examine the generalisation of $R$.

Although $R$ is a more complex operator, its general performance at 2.48% error is even better that the performance of the previous operators (see Table 7.3). The same goes for its generalisation, with 5.28% error. It is also important that the variation of the performance with regard to the position of the error occurrence is the same as before - better at the initial and final positions. This reconfirms the hypothesis that the DRs of strings developed by the RANs are hierarchically clustered and that this hierarchy is keyed on both their initial and final tokens.

7.6 Scaling up

An excellent performance on some hundred or so strings does not impress much nowadays. Although theoretically important, models which do not attempt larger size data sets may not perform well in the real world, which
Table 7.4: Performance of holistic operator $E_1 B(DR_w, p) = w_p$ trained to extract symbols at a position given as a parameter. This time, the operator is trained and tested on the larger databases $B_{\text{train}}$ and $B_{\text{test}}$, containing 5,250 and 11,850 words, correspondingly. The performance of an MLP with the same number of hidden neurons is almost the same as in the earlier experiment (see Table 7.2). Again, the error in the beginning and the end of the strings is smaller than the error in the middle.

is one of the objectives in connectionist modeling. This is because some problems might turn out to be computationally insolvable with certain architectures or for some type of data, as the data size increase – they may be intractable in the sense of theoretical computer science. Especially with learning algorithms based on error minimisation, it might turn out that it is practically impossible to find a near-optimal solution $W^\#$ that minimises a given error function $E_{\text{Data}}(W)$ reasonably well.

One solution to learning very complex problems is to incorporate some basic biases. In the proposed holistic model, an important bias that facilitates the training of a holistic operator $H$ is the systematic development of the data for $H$. If this holistic operator $H$ is trained to learn certain problems represented with either non-systematic or systematic data, it would need many more resources to learn the task in the first case, if possible at all with a given amount of resources (weight space). Systematicity in the input / output data would significantly decrease the efforts needed to find the dependencies between the input and the output. And interestingly, systematicity as a bias here was not deliberately introduced, but it rather is a property of the RAN processing – the RANs transfer systematicity expressed in time (set of sequences) into systematicity represented in space (set of vectors - DRs).

Other facilitations might include more specific network structure, but this is another problem. The basic objective here is to show that the DRs developed by the Recurrent Autoassociative Networks are systematic in a
Table 7.5: Performance of the reversing holistic operator $R1(DR_w)$ trained and tested on the larger databases $B_{train}$ and $B_{test}$, which contain 5,250 and 11,850 words, correspondingly. This time, the MLP with 300 hidden neurons performs even better than the operator trained on a smaller database. Also, the pattern of smaller error around the word edges repeats.

way that allows learning of complex dependencies among larger amounts of data. Even more, I claim that the the larger the corpus is, the more systematically those representations are developed by the RANs. Therefore, larger amounts of data is not expected to hinder the training on the above problems.

Indeed, the same holistic operators explored earlier, but trained and tested on the larger databases $B_{train}$ and $B_{test}$, performed about as well as earlier. The holistic operator extracting symbols $E_{1p}(DR_{wm},p)$ achieved the same performance with the same number of hidden neurons (see table 7.4). However, in order to achieve similar results, the more complex reversing operator $R1(w)$ needed 300 hidden neurons instead of 150 (Table 7.5). However, it works on a database containing over five times as many strings as the smaller one.

**Holistic operations on DRs of numbers**

Although larger and more representative of English, the $B$-dataset is still not complete. Yet, it is also interesting to study holistic operations on complete sets of holistic representations, such as the DRs developed earlier for numbers, externally encoded as sequences of digits (see section 6.4.3). For this purpose, the same two operators were trained on this set. More specifically, a holistic symbol extractor $E_M(DR_M,p)$ with 50 hidden neurons was trained on the $M_{Train}$ set until it learned to decode correctly almost all tokens – with an insignificant 0.03% token error. Then it was tested on $M_{Test}$, resulting also in a very small, 0.07% token error. This is obviously much better than the performance of the previous operators. It means that
the more complete the set used for training is, the more systematically the DRs are developed and the better the performance of the holistic operators is. The reversing operator trained on $M$ continued this tendency. It learned to reverse $M$ – all 10,000 holistically represented numbers – with 0.8% token error, which is also much better than the training of the same operator on the $A$ and the $B$ sets of words.

All this confirms the possibility of processing substantial amounts of data in a holistic way, especially when the holistic representations the operators work on are developed in a very systematic manner – what was showed with a number of demonstrations.

### 7.7 Discussion

*Holism* regards elements, systems, etc. as wholes. From a holistic point of view, first, elements of a holistic system are significant in the system only, and second, elements of a system that are meaningful as wholes can be treated as wholes, from which the notion of holistic computations derives: operations on holistic elements. This chapter argued that RANs develop a holistic system of distributed representations and that these representations are holistic, and therefore they can be used for holistic computations. The latter was examined with a few simple holistic operators implemented with an MLP.

Where Neural Network models are concerned, experimental results are as necessary as theoretical proofs. For example, MLPs can in theory encode any mapping, but in practice the model has limitations in learning. The same goes for the SRNs. In that respect, the experimental research reported on here had three main objectives: firstly, to demonstrate that as constituting a holistic system, the holistic DRs of sequences developed by the RANs can be used for holistic computations; secondly, to probe the structure of these representations and finally, to examine the scalability of the holistic operators. For this purpose, two groups of experiments were conducted: *extracting* tokens of the original strings from their DRs and holistic string *inversion*. The scalability of these two operators was tested by experimenting on two data sets containing words – a smaller database and a larger one – and on a complete set of numbers represented as strings of digits.

The training of all holistic operators – MLPs having as an input DRs of strings (and the targeting position in the symbol-extracting operator) – was easy, quick and showed reasonably small error. They generalised well.
However, when the performance of those operators was projected against position in the original string, it was found that they performed worse when dealing with tokens located internally for the strings. This sort of performance was found independently for three types of holistic operators; it was also observed in a cluster analysis of the DRs. All this suggests that the DRs are organised hierarchically into clusters keyed on both the anterior and posterior tokens of the sequences.

The prominence of the final symbols is easy to explain by looking back to the process of developing of DRs by the RANs: the DRs stand for the content of the hidden layer of the RANs at the moment when the final symbols from the string being encoded had just been processed, so the final positions are more prominent. This is similar to the stack-like organisation of the RAAM model (see Kwasny & Kalman 1995). However, the DRs developed by the RANs also grant prominence to the anterior symbols, which is due to the pressure by the RAN’s learning algorithm to reproduce the processing strings in order, as opposed to RAAMs which decode sequences inversely. Regarding length as being a key factor, too, this could be explained by the size of the vector space needed by the RANs to encode and decode strings: shorter sequences require fewer accessory context states (representations). Also, time and time-related variables are apparently important parameters in the encoding process.

Thinking in terms of cognition, this sort of performance – better in the beginning and in the end of the sequences than in the middle – reminds one of the human memory performance on lists – the well known primacy and recency effects, which consist of better performance for both the earliest and latest items in a list that humans are given to remember and recall later. A further study in this direction might find support for the speculation that the RANs resemble the brain mechanisms for sequential encoding and decoding better than the RAAM does, which model was criticised for the inverse order of sequence decoding. One might go even further, and give an alternative explanation of those memory effects, based on the specificity of the RANs.

Finally, I will finish the discussion with the conclusion that holistic computations over DRs developed by the RANs can be used for holistic linguistic computations and that further research is necessary in order to find more optimal NN learning methods appropriate for complex holistic transformations.