Neural Network Recognition of Spelling Errors

Mark Lewellen
Computational Linguistics, Georgetown University
Washington DC, 20057-1051
mlewellen@apptek.com

Abstract

One area in which artificial neural networks (ANNs) may strengthen NLP systems is in the identification of words under noisy conditions. In order to achieve this benefit when spelling errors or spelling variants are present, variable-length strings of symbols must be converted to ANN input/output form—fixed-length arrays of numbers. A common view in the neural network community has been that different forms of input/output representations have negligible effect on ANN performance. This paper, however, shows that input/output representations can in fact affect the performance of ANNs in the case of natural language words. Minimum properties for an adequate word representation are proposed, as well as new methods of word representation.

To test the hypothesis that word representations significantly affect ANN performance, traditional and new word representations are evaluated for their ability to recognize words in the presence of four types of typographical noise: substitutions, insertions, deletions and reversals of letters. The results indicate that word representations have a significant effect on ANN performance. Additionally, different types of word representation are shown to perform better on different types of error.

Introduction

ANNs are a promising technology for NLP, since a strength of ANNs is their "common sense" ability to make reasonable decisions even when faced with novel data, while a weakness of NLP applications is brittleness in the face of ambiguous situations. One area in which much ambiguity occurs is the identification of words: words may be misspelled, they may have valid spelling variants, and they can be homographic. Robust word recognition capabilities can improve applications which involve text understanding, and are the central component of applications such as spell-checking and name searching.

In order for ANNs to recognize variant and homographic forms of words, however, words must be transformed to a form that is meaningful to ANNs. The interface to ANNs is input and output layers each composed of fixed numbers of nodes. Each node is associated with a numerical value, typically between 0 and 1. Thus, words—variable-length strings of symbols—need to be converted to fixed-length arrays of numbers in order to be processed by ANNs. The resulting word representations should ideally:

1) be in a form which enables an ANN to identify spelling similarities and differences;
2) represent all the letters of words;
3) be concise enough to allow processing of a large number of words in a reasonable time.

To date, research in ANNs has ignored these low-level input issues, even though they critically affect "higher-level" processing. A common view has been that different representation methods do not significantly impact ANN performance. This paper, however, presents word representations that significantly enhance ANN performance on natural language words.

1 Word Representations

To represent words for ANNs, symbols need to be converted to numbers and variable length must be converted to fixed length, ideally under the three constraints listed above.

To handle the variable length of words, recurrent ANNs have sometimes been used. In a recurrent ANN, the values of nodes in the output or hidden layers are recycled to a portion of the input layer nodes. Input to the network thus consists of a letter representation plus the state of the network after all previous letters. Recurrent ANNs have

several drawbacks, though: they require much more training time and use part of their processing capability for the development of representations, rather than the problem to which the network is applied. Importantly, such designs suffer from a primacy effect: the initial letters of a word receive greater emphasis, so that errors at the beginning of words cause much greater problems than errors at the end of words.

1.1 Fixed-Length Letter Buffers

The most common method of representing letters is in a buffer containing a set number of letter representations. For example, space might be allocated for up to 14 letters; if 26 nodes are used to represent each letter, then the input buffer uses a total of 364 nodes. In such "fixed-length letter buffers" (FLLBs), letters are traditionally placed in the buffer one by one from the left, as in writing from left to right. These left-aligned FLLBs suffer from the primacy effect discussed above. To correct this problem, two new FLLB structures are proposed: split and bi-directional.

A split FLLB splits the word in two, left-justifying the first half, and right-justifying the second half, in order to halve the effect of errors which cause position shifts in subsequent letters.

A bi-directional FLLB is similar to a split FLLB, but uses all available space in the FLLB. Instead of leaving certain letter positions blank, as in a split representation, extra letter positions are filled by continuing to add letters from the beginning and end. Such a scheme tends to weight the middle of words more heavily, as that portion of a word is more likely to be represented twice.

Examples of FLLBs for the word "knight" are:

Left	*k*	*n*	*i*	*g*	*h*	*t*				
Split	*k*	*n*	*i*					*g*	*h*	*t*
Bi-D	*k*	*n*	*i*	*g*	*h*	*n*	*i*	*g*	*h*	*t*

1.2 Local vs. Distributed Representations

Each letter of an FLLB needs to be converted to a numeric representation. Letters are symbols, which are adequately represented by binary, rather than continuous values. Consequently, representations become much larger, which may place limitations on the choice of word representation.

Each letter may be represented in a "local" or "distributed" manner. In a local letter representation, 26 nodes could be utilized, one for each letter of the alphabet. Only the node corresponding to a particular letter is assigned a value of 1, while the rest of the nodes have a value of 0.

In a distributed representation, several nodes combine to represent one letter; each node may also participate in different letter representations. Distributed representations are more biologically plausible, and are particularly desirable for their compressive characteristics, as well as the increased error-tolerance of having several, rather than one, nodes contribute to a representation.

2 Test Design and Results

Testing of alternative representations was performed with two variables, FLLB type and local/distributed, for each of four types of error.

A test corpus of similarly-spelled words was developed from a list of American English homophones (Antworth 1993). Homophone groups containing words with apostrophes were removed, yielding a list of 1449 words. Each word was randomly assigned an arbitrary 4-digit symbol.

The training set for the ANN consists of 1449 word/symbol pairs. The words were presented to the network in a 14-letter FLLB, composed in six methods (left-aligned | split | bi-directional X local | distributed). The local method uses 1 of 26 nodes for each letter (total of 364 nodes), while the distributed method uses 4 of 11 nodes for each letter (total of 154 nodes), with no more than two nodes permitted to overlap with any other letter representation. The output of the network is a 4-digit symbol, represented by four 9-node distributed representations (total of 36 nodes).

Four test sets were developed from the word list, each roughly 10% of the list size, resulting in one test set of 150 words for each type of error—substitution, reversal, insertion and deletion. The errors were created by hand, evenly distributed through the beginning, middle and end of words.

Training and testing were performed with an ARTMAP-ICMM ANN, a variant of ARTMAP-IC (Carpenter & Markuzon 1996) specialized for data sets containing many-to-many mappings. The testing phase of ARTMAP-ICMM outputs a rank-ordered list of potential mappings, with the rank of the desired output returned as a score. A score of 1 is optimal; in this case, the worst score is 1449. As scores become larger, they become less meaningful; for example, a difference of 10 is

much more significant between 5 and 15 than between 100 and 110.

To evaluate network performance for a test set, measures of central tendency are computed for the rank scores of the test set. Since large scores become increasingly arbitrary, it is desirable to limit their effect on measures of central tendency. A measure that often fulfills this criterion is the median; however it is somewhat inexact for this purpose. The squared mean root (analagous to the quadratic mean) lessens the influence of large scores while remaining more discriminating:

$$\left(\left(\sum_{j=1}^{N}\sqrt{X_j}\right)/N\right)^2, \text{ or } \overline{\sqrt{X}}^2.$$

The squared mean root is presented first, as a primary indicator, with the median following for comparison. The test results are presented along both test variables for each of four error types.

Substitution	Left	Split	Bi-D
Local	1.7 / 1	1.7 / 1	1.5 / 1
Distributed	1.8 / 1	1.7 / 1	1.6 / 1

Reversal	Left	Split	Bi-D
Local	4.8 / 3	5.8 / 4	3.4 / 2
Distributed	5.9 / 2	7.0 / 3	4.3 / 2

Insertion	Left	Split	Bi-D
Local	99 / 19	5.9 / 2	4.7 / 3
Distributed	97 / 24	7.8 / 3	7.5 / 5.5

Deletion	Left	Split	Bi-D
Local	125 / 64	7.3 / 3	21.7/ 21.5
Distributed	152 / 97	13.5 / 4	38.8/ 39.5

3 Position-maintaining and position-altering errors

The results for the four types of error can be used to create two groupings: position-maintaining and position-altering errors. The position-maintaining errors are substitution and reversal errors, which do not cause other letters to shift to different positions. The position-altering errors (insertions and deletions), however, do cause such a shift. The scores demonstrate that for FLLB representations, position-altering errors cause greater difficulty than position-maintaining errors. The traditional left-aligned FLLB performed dramatically worse on position-altering errors (scores of 99|97 and 125|152) than on position-maintaining errors (1.7|1.8 and 4.8|5.9). Both the split and bi-directional FLLBs display much-improved performance on the position-altering errors. The bi-directional FLLB, however, still has substantially more difficulty with deletion errors than does the split FLLB. The split FLLB thus demonstrated the best overall performance of the three FLLB representations.

Along the local/distributed variable, the local representations consistently equal or surpass the performance of the distributed representations. The advantage, however, is relatively minor, unlike the clear distinctions between FLLB type.

Conclusion

This paper has found that word and letter representations can have a significant effect on ANN recognition of spelling errors. It has specifically found that:

- Methods of word representation can have substantial and measureable effects on ANN performance.
- Position-altering (insertion and deletion) and position-maintaining errors (substitution and reversal) have different effects on ANN recognition of spelling errors.
- An FLLB may, in addition to a traditional left-aligned representation, be organized in split and bi-directional structures. These new FLLBs result in improved performance on position-altering errors, with the split representation offering the best performance.

Research in progress includes development of other ANN word representation methods and testing with data from other languages.

Acknowledgements

Thank-you to Gail Carpenter for suggesting the applicability of ARTMAP-IC, and Donald Loritz and anonymous reviewers for their helpful advice.

References

Antworth E., ed. (1993) List of homophones in General American English. Consortium for Lexical Research. 27 Jan. 1998 <ftp://crl.nmsu.edu/CLR/lexica/homophones/>.

Carpenter G. A. and Markuzon N. (1996) *ARTMAP-IC and Medical Diagnosis: Instance Counting and Inconsistent Cases*. Technical Report CAS/CNS TR-96-017. Boston University, Boston, MA.