УДК 004.738.52

А.О. АНТОНЕНКО

КОДИРОВАНИЕ И ВОССТАНОВЛЕНИЕ ПОСЛЕДОВАТЕЛЬНОСТЕЙ

Abstract: *Methods of the coding and reconstructing the sequences based for use binary vector are considered. The results of experimental investigations conducted for task of searching of words with error are adduced.*
Key words: *coding and reconstructing the sequences, searching of words with errors, binary vector.*

Анотація: *Розглядаються методи кодування й відновлення послідовностей, засновані на використанні бінарних векторів. Наводяться результати експериментальних досліджень, проведені для задачі пошуку слів з помилками.*
Ключові слова: *кодування й відновлення послідовностей, пошук слів з помилками, бінарний вектор.*

Аннотация: *Рассматриваются методы кодирования и восстановления последовательностей, основанные на использовании бинарных векторов. Приводятся результаты экспериментальных исследований, проведенные для задачи поиска слов с ошибками.*
Ключевые слова: *кодирование и восстановление последовательностей, поиск слов с ошибками, бинарный вектор.*

1. Введение

Решение проблемы кодирования и восстановления последовательностей является чрезвычайно важным для решения многих задач, таких, как передача, представление, обработка, поиск, восстановление зашумленной исходной информации и т.д.

В данной работе будут исследованы методы кодирования и восстановления последовательностей, основанные на использовании бинарных векторов. Разработка данных методов велась с целью использования их для работы с ассоциативной памятью в ассоциативно-проективных нейронных сетях АПНС [1, 2]. Поэтому для проверки эффективности методов кодирования и восстановления последовательностей необходимо выбрать задачу, в которой требовалось бы определять схожесть последовательностей. Одной из таких задач является поиск последовательностей, в которых возможны ошибки.

Для определения меры схожести кодируемых последовательностей при использовании методов кодирования, предлагаемых в данной работе, будет использоваться расстояние Хемминга – количество отличающихся позиций в бинарных векторах кодируемых последовательностей [3].

Применять поиск целесообразно для последовательностей символов (слов), так как данная задача достаточно понятна, широко применяема и имеет большое количество стандартных методов решения. Одним из наиболее известных и используемых методов поиска является метод, который основан на измерении расстояния Левенштейна, также часто называемого расстоянием редактирования. Выбор этого метода обусловлен двумя факторами. Во-первых, расстояние Левенштейна формализует интуитивное понятие об «ошибке», а, во-вторых, существует множество алгоритмов эффективного его вычисления [4–8].

2. Векторные представления информации

На эффективность методов обработки данных определяющее влияние оказывает способ их представления. В рамках парадигмы ассоциативно-проективных нейронных сетей [1, 2] данные кодируются бинарными векторами (кодвекторами), где отдельные элементы обычно не имеют однозначного смысла. Этим они отличаются от символьных и локальных представлений, которые однозначно соответствуют тем или иным объектам. Под объектами здесь понимаются данные

различной природы и сложности: числовые и символьные, скалярные, векторные, структурированные и т.д. Данные кодвектора являются примером распределенного представления информации, которое широко применяется как при решении прикладных задач (распознавание различных объектов, классификация, ассоциативная память, управление и др.), так и для моделирования различных когнитивных процессов, таких, как сенсорное восприятие, сенсорно-моторная координация, категоризация, рассуждение по аналогии и др. [9]. К достоинствам распределенных представлений относятся: эффективное использование ресурсов (высокая информационная емкость обеспечивается возможностью представить экспоненциально много объектов различными кодвекторами одинаковой размерности); простота оценки степени сходства (через скалярное произведение кодвекторов распределенных представлений); естественное представление сходных объектов (коррелированными кодвекторами, т.е. кодвекторами с большой величиной скалярного произведения); способность работать в условиях шума, сбоев и неопределенности; параллелизм и др.

Рассматриваются кодвекторы: многомерные, бинарные, псевдослучайные (со случайным, но неизменным для тех же кодируемых данных положением единиц) и разреженные (число единичных элементов значительно меньше числа нулевых). Разреженные бинарные векторы допускают эффективную реализацию векторных и векторно-матричных операций. Благодаря разреженности обеспечивается большая информационная емкость ассоциативной памяти, которая может использоваться для хранения и поиска кодвекторов. Число единичных элементов M в разреженных кодвекторах должно быть значительным для поддержания статистической стабильности числа единиц и уменьшения его дисперсии относительно среднего значения. Это, в свою очередь, необходимо для обеспечения точности кодирования и декодирования (восстановления вектора входного сигнального пространства по его бинарному кодвектору). Для разных значений кодируемых величин M должно быть приблизительно одинаковым, чтобы разная плотность кодов не влияла на величину их перекрытия и на работу последующих алгоритмов.

3. Постановка эксперимента

В исследуемых методах кодирования последовательностей, применяемых в данной работе для решения задачи поиска последовательностей, в которых возможны ошибки, информация представляется в виде бинарных кодвекторов, обеспечивающих большую информационную емкость. Основными параметрами формирования бинарных векторов являются: длина кодвектора N, количество единичных элементов кодвектора M, количество схожих единичных элементов K в кодвекторах расположения определенной буквы на соседних местах в слове. Параметр K определяется в процентах от M. Большая размерность $(N >> 1)$, относительно малое число единичных элементов кодвектора $(M << N - M)$ и поддержание этого параметра в заданных пределах при формировании кодвекторов последовательностей увеличивают информационное содержание кодов, статическую стабильность числа единиц в кодах, дают возможность работать со случайными подмножествами кодов с сохранением свойств всего кода и обеспечивают помехоустойчивость кодирования. Оптимальность значений параметров формирования бинарных

кодвекторов является залогом эффективной работы исследуемых методов кодирования. В соответствии с работами по вычислению оптимальных значений этих параметров, выполненных в [10], для осуществления исследований были выбраны такие значения параметров: N = 10000, M = 0,1 N . Параметр K выбирался по результатам выполненного эксперимента из диапазона от 30 до 100%.

Кодирование происходит следующим образом. Формируется бинарный кодвектор каждой буквы. В зависимости от того, на каком месте в слове располагается буква (кроме первого), кодвектор изменяется следующим образом. В кодвекторе расположения буквы на определенном месте в слове единичные элементы в количестве, соответствующем параметру K, заимствуются из кодвектора расположения этой же буквы на предыдущем месте в слове, а остальное количество единичных элементов $(M-K)$ добавляется при помощи генератора случайных чисел. Таким образом, близость расположений определенного символа в слове отражается в сходстве их кодвекторов. Например, кодвекторы определенного символа, находящиеся на второй и третьей позициях в слове, должны иметь большее количество совпадающих величин (единичных элементов) чем кодвекторы того же символа, находящиеся на второй и четвертой позициях в слове. Кодвекторы слов образуются путем логического сложения бинарных кодвекторов символов, которые входят в состав слова: $B = B_1 \vee B_2 \vee B_3 \vee ... \vee B_j$, где j – количество символов в слове.

В данной работе будем исследовать два метода кодирования и восстановления последовательностей. В первом из них длина каждого кодируемого слова является величиной постоянной и соответствует длине максимально длинного слова в базе исходных данных. В связи с этим мы предлагаем называть данный метод кодирования Encoding by Max Length (EML), кодирование по максимальной длине. Все пустые пробелы в слове кодируются при помощи кодвекторов пробела. При таком способе кодирования коды являются помехозащищенными, но неустойчивыми к сдвигам. Для того чтобы сгладить этот недостаток, мы используем такой метод, как «дисторция» или «подвижка». Сравнение искомого слова будет происходить как с оригинальной базой исходных данных, так и с базами, в которых слова будут смещены на одну и две позиции вправо и влево. Изначально предполагалось проводить сравнения с базами слов с единичным смещением вправо и влево, но в результате экспериментов метод с дополнительным смещением на две позиции оказался намного эффективнее.

Второй метод кодирования Sparse Coding (SC), разреженное кодирование, основывается на прореживании результирующего кодвектора слова до заданного значения количества единичных элементов (параметра M). Длина каждого кодируемого слова, в отличие от метода EML, не является постоянной величиной и соответствует длине данного слова. Для осуществления прореживания необходимо, чтобы количество единичных элементов кодвектора слова было не меньше заданного параметра M . Поэтому параметр M поддерживается в заданных пределах для кодвекторов слова, имеющего наименьшую длину в базе исходных данных. При кодировании слов с длиной, большей минимальной, параметр M кодвектора слова поддерживается путем случайного прореживания кодвектора до заданного значения.

Для проверки эффективности разрабатываемых методов кодирования последовательностей используется метод Левенштейна, который базируется на измерении разницы двух последовательностей символов (строк) относительно минимального количества элементарных операций редактирования. Набор этих операций состоит из операций замены, вставки и удаления одного символа. Фактически при поиске этим методом требуется не столько расстояние между последовательностями символов, сколько знание, превышает ли это расстояние некоторое наперед заданное пороговое значение. Одним из недостатков этого метода является резкое возрастание количества результатов поиска при увеличении порогового значения расстояния между последовательностями символов. В данной работе мы будем применять метод Левенштейна-Дамерау [5, 6], который является усовершенствованным методом Левенштейна, путем расширения набора элементарных операций операцией перестановки соседних символов (при условии, что эти символы являются смежными в обеих строках).

Так как для определения меры схожести кодируемых слов при использовании методов EML и SC используется расстояние Хемминга, то результатом нечеткого поиска будет слово, имеющее с исходным словом минимальное расстояние по Хеммингу. Для увеличения количества результатов поиска свыше одного мы взяли Хеммингово расстояние не минимальное, а увеличенное на 20%.

В качестве исходных данных для эксперимента использовалась база англоязычных названий городов мира, население которых превышает 100 000 человек. Максимальная длина слов составляет 15 символов. Размер базы составляет 3365 наименований.

Эксперименты по сравнению методов кодирования и восстановления последовательностей проводились на ряде текстовых операций, создающих в словах ошибки, которые допускает человек: удаление, вставка, замена, дублирование одного и двух произвольных символов, а также перестановка смежных соседних и соседних через одного символов. Из базы исходных данных псевдослучайным образом выбирались слова, в которых поочередно над каждым символом производились вышеописанные операции редактирования. Для получения статистически достоверных результатов данная операция проводилась 4000 раз. Результаты эксперимента усреднялись.

4. Результаты эксперимента

В первую очередь были проведены эксперименты по выбору оптимального значения параметра K. Как видно на рис. 1, наиболее оптимальным является значение параметра K, равное 80%.

На рис. 2 и рис. 3 представлены результаты нахождения искомого слова в результатах поиска и среднее количество результатов поиска, соответственно, по всем операциям редактирования (уд. 1 – операция удаления одной буквы; уд. 2 – операция удаления двух букв; доб. операции добавления букв; дуб. – операция дублирования букв; зам. – операции замены букв; см. 1 – операция перемены соседних букв местами; см. 2 – операция перемены местами букв, расположенных в слове через одну). Метод Левенштейна-Дамерау дает 100% наличия искомого слова в результатах поиска по всем операциям редактирования.

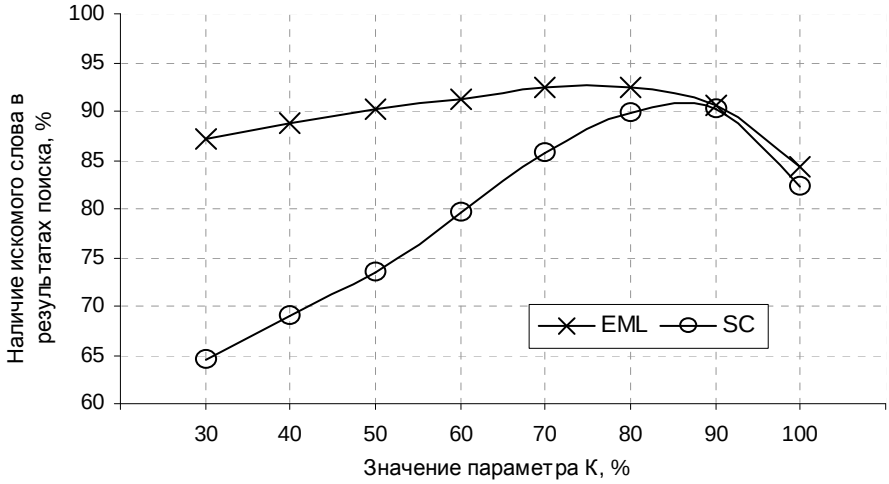

Рис. 1. Зависимость результатов поиска от значения параметра K

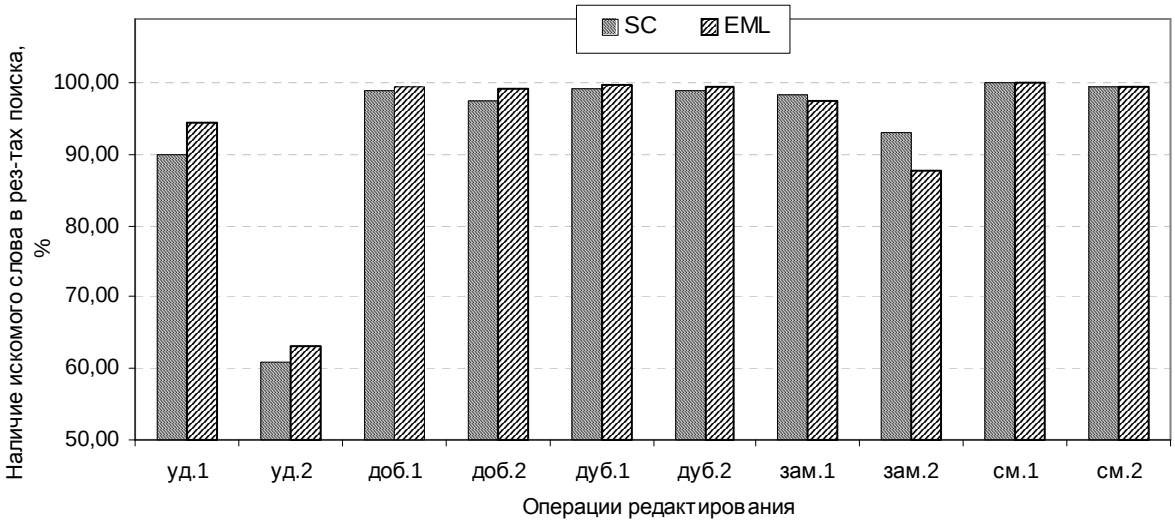

Рис. 2. Нахождение искомого слова в результатах поиска по всем операциям редактирования

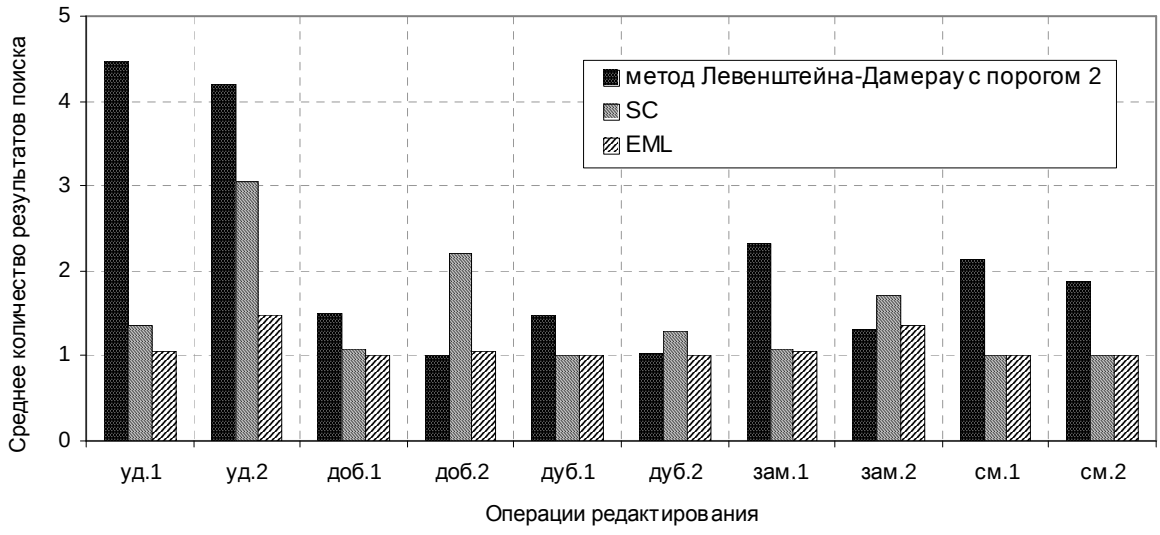

Рис. 3. Среднее количество результатов поиска по всем операциям редактирования

Результаты показывают, что предлагаемые методы кодирования практически не уступают применяемому методу Левенштейна-Дамерау в способности нахождения искомого слова в результатах поиска. Неудовлетворительно они себя проявляют лишь при операции удаления двух букв, что поясняется значительным изменением кодвектора искомого слова в связи с тем, что сдвиги букв в слове имеют различный характер из-за разного расположения удаляемых букв в слове. Однако методы SC и EML, в отличие от метода Левенштейна-Дамерау, выдают в среднем значительно меньшее количество результатов поиска.

5. Выводы

Исследованные методы кодирования были проверены на стандартной задаче поиска слов, в которых возможны ошибки. Они проявили себя не намного хуже, а в некотором смысле и лучше, чем существующие методы, специализированные для данной задачи. Это дает право говорить о возможности применения данных методов кодирования и восстановления последовательностей для задач, не имеющих определенных алгоритмов решения. Исследованные методы также совместимы с форматом данных в АПНС [1, 2]. Это позволяет в дальнейшем использовать АПНС для решения рассматриваемой задачи, что должно значительно улучшить полученные результаты.

В будущем предполагается дальнейшее исследование кодов, применяемых в данных методах, с целью распознавания их уже не при помощи метода Хемминга, а с применением АПНС, что должно отразиться на улучшении результатов при использовании этих кодов для решения широкого спектра задач.

СПИСОК ЛИТЕРАТУРЫ

1. Куссуль Э.М. Ассоциативные нейроподобные структуры. – Киев: Наукова думка, 1991. – 144 с.
2. Рачковский Д.А., Слипченко С.В., Куссуль Э.М., Байдак Т.Н. Процедура связывания для бинарного распределенного представления данных // Кибернетика и системный анализ. – 2005. – № 3. – С. 3–18.
3. Блейхут Р. Теория и практика кодов, контролирующих ошибки: Пер. с англ. – М.: Мир, 1986. – С. 576.
4. Ehrenfeucht A., Haussler D. A New Distance Metric on Strings Computable in Linear Time. Discrete Applied Mathematics. – 1988. – Vol. 20. – P. 191–203.
5. Masek U., Peterson M.S. A faster algorithm for computing string-edit distances // Journal of Computer and System Sciences. – 1980. – Vol. 20(1). – P. 785–807.
6. Sellers P.H. The Theory of Computation of Evolutionary Distances: Pattern recognition // Journal of Algorithms. – 1980. – Vol. 1. – P. 359–373.
7. Ukkonen E. Approximate String Matching over Suffix-Trees // Proc. of the Fourth Annual Symposium on Combinatorial Pattern Matching. – Padova, Italy. – 1993. – P. 229–242.
8. Wagner R.A., Fisher M.J. The String to String Correction Problem // Journal of the ACM. – 1974. – Vol. 21(1). – P.168–173.
9. Rachkovskij D.A., Kussul E.M. Building a world model with structure-sensitive distributed representations. – http://www.bbsonline.org/Preprints/Rachkovskij/Referees/Rachkovskij.pdf.
10. Рачковский Д.А., Слипченко С.В., Куссуль Э.М., Байдак Т.Н. Разреженное бинарное распределенное кодирование скалярных величин // Проблемы управления и информатики. – 2005. – № 3. – С. 89–103.