Institute of Information and Computational Technologies

ISSN : 2788-7677 (Online)
ISSN : 2788-7987 (Print)

ADVANCED TECHNOLOGIES AND COMPUTER SCIENCE

2022
№2

www.atcs.iict.kz

Institute of Information and Computational Technologies

Advanced Technologies and computer science

№2

Almaty 2022

ISSN: 2788-7677 (Online)
ISSN : 2788-7987 (Print)

Institute of Information and Computational Technologies,

Advanced Technologies and computer science

28 Shevchenko str., Almaty,Republic of Kazakhstan
7 (727) 272-37-11
atcs@iict.kz

About the Journal

Advance technologies and computer science is a bilingual scientific peer-reviewed, interdisciplinary, electronic journal of open access, including thematic areas:

- Section **"Applied mathematics, computer science and control theory"** includes papers describing modern problems in these areas.
- Section **"Information and telecommunication technologies"** also includes the following topics:
    - Data transmission systems and networks.
    - Internet technologies.
    - Cloud technologies.
    - Parallel computing.
    - Distributed computing.
    - Supercomputer and cluster systems.
    - Big data processing (Big-data).
    - Geographic Information Systems and Technologies.
- In the section **"Artificial intelligence technologies"** in addition to technology, there are works on topics:
    - Intelligent Management Systems.
    - Speech technology and computer linguistics.
    - Pattern Recognition and Image Processing.
    - Bioinformatics and biometric systems.
    - Human-machine interaction.
    - Machine learning.
    - Intelligent Robotic Systems.
- The section **"Information Security and Data Protection"** also covers topics:
    - Software and hardware information protection.
    - Mathematical methods for ensuring information security of complex systems.
- The section **"Modeling and optimization of complex systems and business processes"** may include:
    - Computational mathematics, numerical analysis and programming, mathematical logic.
    - Theory of Statistics.
    - Statistical Methods.

Contents

UDC 004.02
IRSTI 20.53.15

INTERNAL WORKLOAD OF NOSQL RELATIONAL DATABASE

Mirzakhmet Syzdykov
al-Farabi Kazakh National University, Almaty, Kazakhstan
mspmail598@gmail.com
ORCID ID: https://orcid.org/0000-0002-8086-775X

Abstract. In this article we present the scheme of internal workload of typical relational database supporting NoSQL human interaction protocol, we state that the production consumption of query complexity cannot be avoided and modern techniques like explain plan play vital role in bringing the functionality of database server to the end-user with support of open data storage like CSV file format which, in turn, is textual – thus, the question of open storage is discussed in this work within the unlimited support of table sheet formats like CSV, as to the modern Big Data trends we propose the solution used supposedly empirically by popular SQL database systems – this is a plan to run the incoming query and transform the result to the format which can be adopted for final output; we also develop the end programmable product using C# programming language and .NET framework which are the modern programming environments for proper development of platform independent software – the proof of important theorems of relational database query complexity processing and optimization is also given.

Keywords: NoSQL, data processing, relational database, open storage.

Introduction

In the modern era the structured query language (SQL) became a standard for purpose of querying data to obtain the results as per giving the limits and if-statements to get the more precise result of seeding values. The NoSQL (not only SQL or no SQL) trend today is emerging as an alternative to the general approach and gives the possibility of probable out-performance for the previous generations of SQL database management systems (DBMS).

The work of addressing the OLAP feature in NoSQL databases [1] is solved by our definition of structures along which we achieve the universality of using relational data over production system, the definition of which is give later in this work.

From the other side [2] the NoSQL is giving more broad perspectives for Big Data, according to the authors of referenced work the modern data storages approximate to petabytes. We give the solution according to the optimizing explain plan for the relational database tuple to make the right decision of choosing the data to be processed on server or specially developed RISC-machine.

The comparison of SQL and NoSQL databases is presented in [3]. As previously described we prove by definition of tuple that SQL and NoSQL databases working time observes to be equivalent to the theorem proof.

Java Script object notation (JSON) is discussed in [4]. For this time MongoDB or other systems adopt this format for formal clauses which are equivalent to the SQL queries, the difference is in how they are processed by the “black box” or database engine for query processing and assembling of finite data.

In [5] the way of optimization is presented with respect to the document and relational model. Our approach based upon the law proof which, in turn, is an outcome defined by the main theorem and, thus, the optimization is only a path through tuple elements giving better or best throughput.

The comparison of document-based MongoDB and relational MySQL is given in [6] – it turns out that in some cases the non-relational approach is better for Big Data, however, we define that relational and non-relational databases adhere to the same theorem of the complexity equivalence of this classes of data processors.

For the purpose of our proof of concept we have developed the hybrid database system based on the relational model with the support of non-relational features like NoSQL and typical join operations [7]. In our model the open textual storage is used like comma-separated values (CSV).

SQL DBMS review

SQL database systems are widely used in the modern age, especially for the web applications and

cloud processing of streams of data. They use typical relational database model for operational purposes and to define the final human computer interaction with the end user. The SQL or query by example (QBE) is used as a query language in order to get the results which are seeded by the predicates of the formed query string.

The bottleneck of these systems is that they are highly inefficient and require special hardware resources to process the query and store indexes and data – thus, the question of optimization arises and is solved by introducing explain plan or other alternatives. Most of them are open-source and gives us the possibility to fork the new database engine with the support of extended or NoSQL features.

Thus, the SQL database systems cover approximately more than fifty percents, or half, of the Internet segment of the on-line services and are limited to the gigabytes of data to be stored and queried on budget and on time.

NoSQL DBMS review

Modern trends like NoSQL arise to be an alternative to the industry standard like SQL database. According to our research, most of them adapt JSON notation or any other document-oriented format like XML, for example, which, in turn, is accomplished by the XSL assertion.

Thus, the NoSQL databases which are more efficient than its predecessors are available today in the most common document formats – we consider this as a good addendum, meanwhile, the other side is that the relational model which is simple for understanding cannot be achieved and the broad perspectives of data modeling are omitted.

In the other hand, the relational model is very strict which gives the priority to the non-relational databases to convert data to documents and operate on the higher level rather than typical database schemes and constructed data models.

We define the paradigm between relational database and object-oriented programming (OOP) to be equivalent according to the modern standards of workload of the today developed systems by the use of persistence of objects in the relational model as a single separate table in the database system.

Main theorem and its proof. We define the theorem for the relational data and the data which are document-oriented which are subject of the joining different set of entities in one pre-defined tuple.

By this tuple we define the following expression:

$$<T, Q, R>, (1)$$

where T is a set of tables or document formats in non-relational database and Q is a set of queries to the database engine incoming from end user, and R is a set of optimization rules – these rules can be met as explain plan hints in Oracle database.

Thus, by defining the common model for both SQL and NoSQL databases which can be also divided as relational or non-relational, we get the main theorem which states that the production rule for internal joins adheres to the multiplication of all the tables for the given complexity of each query in Q cannot be avoided and is supposed to be processed by the cursor:

$$O(Q) = |T_1| * |T_2| * \dots * |T_n|. \qquad (2)$$

While the cursor is to be defined as a state in the production rule (2):

$$C(Q) = (t_1, t_2, \dots, t_n): t_i = 1..|T_i|, \qquad (3)$$

where t_i is a current step in the cursor for production consumption within the defined complexity and |...| is a number of rows in the table or document to be processed by the query.

The proof of the theorem for any type of database, which can be both relational or document-based, follows from the fact that the set R of optimization rules is limited to the set of complexity of cursor tuple in (2).

By this proof we get that the join factor of several tables cannot be uniformly optimized in the worst case of evaluation of the truth of the query predicate in Q by simple paths of explain plan, meanwhile, the whole structure is to be preserved, so that any of the acceptable data are covered by each iteration of querying process.

As the cursor is common for both models, we can state that the optimization rules in set *R* are to be constructed according to non-index approach – as we use open data storage in pure textual format. These optimization rules reduce the production explosive complexity to the limited set of operations for which the tuple evaluates to true and the query predicates hold true at the same time. This technique is effectively used in the modern DBMS like Oracle, where user has the possibility to tune the execution plan of the query according to hints which appear in comments for the query string; the explain plan feature is also present in Oracle with respect to the user evaluation and tuning.

Proof of concept software description

For our purpose to prove the product-rule (2) explosion in (1) while evaluating the query, we have developed the software available from GitHub [7].

In this manner we define some SQL-like clauses like logical operators as well as joining operator.

While evaluating the query expression, the optimization rule *r* in *R* can be applied to the cursor and the order of acceptance or reject of current cursor can be determined.

Conclusion

Thus, we have given the proof of product explosion for relational model or document-based model where relational operator cannot be avoided in general, if even we would consider the NoSQL database system.

We have also developed the software for evaluation purposes which gives the outcome as the relational model leads to the product explosion for complexity of processing the query.

We also give the solution for Big Data by applying the optimization rules in definition of tuple. By this fact, we define the set of separated rules which are to minimize the set of elements in cursor and, thus, leading to the more effective process of query evaluation within the product model for both relational or non-relational databases within the entity join concept. This concept leads us to the fact that relational model cannot be avoided as per the complexity of cursor manipulation over the whole set of data stored in table or document storage.

Acknowledgements

The author expresses gratitude to the Free Software Foundation (FSF) for providing with all the necessary software in order to prepare this work for evaluation.

We have also experienced a good work-around using the tools and cloud services described in this work.

References

[1] Banerjee, Shreya. Bhaskar, Sourabh. Sarkar, Anirban. Narayan, C. Debnath. A Formal OLAP Algebra for NoSQL based Data Warehouses. Annals of Emerging Technologies in Computing. 2021. 5. 154-161. 10.33166/AETiC.2021.05.019.

[2] Erraissi, Allae. Hadoop Storage Big Data layer: meta-modeling of key concepts and features. International Journal of Advanced Trends in Computer Science and Engineering. 2019. 8. 646-653. 10.30534/ijatcse/2019/49832019.

[3] Chang, Ming-Li. Chua, Hui Na. SQL and NoSQL Database Comparison. Proceedings of the 2018 Future of Information and Communication Conference (FICC). 2019. 1. 10.1007/978-3-030-03402-3_20.

[4] Lv, Teng. Yan, Ping. He, Weimin. Wang, Tao. On Approximate Querying Large-Scale JSON Data. Journal of Physics: Conference Series. 2020. 1575. 012066. 10.1088/1742-6596/1575/1/012066.

[5] Ha, Muon. Shichkina, Yulia. Translating a Distributed Relational Database to a Document Database. Data Science and Engineering. 2022. 1-20. 10.1007/s41019-022-00181-9.

[6] Matallah, Houcine. Belalem, Ghalem. Bouamrane, K. Comparative Study Between the MySQL Relational Database and the MongoDB NoSQL Database. International Journal of Software Science and Computational Intelligence. 2021. 13. 38-63. 10.4018/IJSSCI.2021070104.

[7] Syzdykov, Mirzakhmet. CSVdb: NoSQL CSV Database System. https://github.com/mirzakhmets/CSVdb. (accessed May 7th, 2022).

Әл-Фараби атындағы Қазақ ұлттық университеті, Алматы, Қазақстан
mspmail598@gmail.com
ORCID ID: https://orcid.org/0000-0002-8086-775X

Аңдатпа. Бұл мақалада біз NoSQL пайдаланушысының өзара әрекеттесу протоколын қолдайтын типтік реляциялық Дерекқордың ішкі жүктеме схемасын ұсынамыз, біз сұраныстардың күрделілігін өндірістік тұтынудан аулақ бола алмайтынымызды және explain plan сияқты заманауи әдістер дерекқор серверінің функционалдығын ашық деректер қоймасын қолдайтын соңғы пайдаланушыға жеткізуде маңызды рөл атқаратынын мәлімдейміз. CSV файл пішімі сияқты, ол өз кезегінде мәтіндік болып табылады – осылайша, ашық репозиторий мәселесі CSV сияқты кесте форматтарын шексіз қолдау аясында осы жұмыста талқыланады, қазіргі заманғы үлкен деректер тенденцияларына келетін болсақ, біз SQL-дің танымал дерекқор жүйелері эмпирикалық түрде қолданатын шешімді ұсынамыз-бұл кіріс сұрауды іске қосу және нәтижені түпкілікті шешім қабылдау үшін қабылдануы мүмкін форматқа айналдыру жоспары шығару; сондай – ақ, біз C# бағдарламалау тілін және.NET framework көмегімен бағдарламаланатын өнімді жасаймыз, олар бағдарламалық жасақтама платформасынан тәуелсіз дұрыс әзірлеу үшін заманауи бағдарламалау ортасы болып табылады-сонымен қатар реляциялық мәліметтер базасына сұраныстардың күрделілігін өңдеу және оңтайландыру үшін маңызды теоремалардың дәлелі.

Түйін сөздер: NoSQL, деректерді өңдеу,реляциялық деректер базасы, ашық сақтау.

ВНУТРЕННЯЯ РАБОЧАЯ НАГРУЗКА РЕЛЯЦИОННОЙ БАЗЫ ДАННЫХ NOSQL

Мырзахмет Сыздыков
Казахский национальный университет имени аль-Фараби, Алматы, Казахстан
mspmail598@gmail.com
ORCID ID: https://orcid.org/0000-0002-8086-775X

Аннотация. В этой статье мы представляем схему внутренней рабочей нагрузки типичной реляционной базы данных, поддерживающей протокол взаимодействия с пользователем NoSQL, мы заявляем, что нельзя избежать производственного потребления сложности запросов, а современные методы, такие как explain plan, играют жизненно важную роль в доведении функциональности сервера баз данных до конечного пользователя с поддержкой открытого хранилища данных, такого как формат файла CSV, который, в свою очередь, является текстовым – таким образом, вопрос открытого хранилища обсуждается в этой работе в рамках неограниченной поддержки форматов таблиц, таких как CSV, что касается современных тенденций в области больших данных, мы предлагаем решение, предположительно эмпирически используемое популярными системами баз данных SQL – это план запуска входящего запроса и преобразования результата в формат, который может быть принят для окончательного вывода; мы также разрабатываем конечный программируемый продукт, используя язык программирования C# и .NET framework, которые являются современными средами программирования для правильной разработки независимого от платформы программного обеспечения – также приводится доказательство важных теорем обработки и оптимизации сложности запросов к реляционным базам данных.

Ключевые слова: NoSQL, обработка данных, реляционная база данных, открытое хранилище.

Сведения об авторе:
Анг.: Syzdykov Mirzakhmet - al-Farabi Kazakh National University, Almaty, Kazakhstan
Каз.: Сыздықов Мырзахмет- әл-Фараби атындағы Қазақ ұлттық университеті, Алматы, Қазақстан.
Рус.: Сыздыков Мырзахмет- Казахский национальный университет имени аль-Фараби, Алматы, Казахстан.

UDC 519.6

NEWTON–COTES FORMULAS FOR NUMERICAL INTEGRATION IN MAPLE

Yalda Qani
Department of Mathematics, Faryab University, Maymana, Afghanistan
e-mail: yalda.qani@gmail.com

Abstract. This article discusses the Newton-Coates formulas for the numerical integration of the Maple package, which have been discussed. In this study, for the purpose of numerical integration of trapezoidal method, Simpson method and Simpson 3/8 method, we examined. We analyzed the function f defined in the interval [a, b] at point f (x) by presenting several analyzes. The Maple package provides the latest applications of numerical analysis in a shorter and more efficient way, which has led to the solution of important scientific and technical problems today. With the help of Maple software, we solved our desired changes on its clear and reliable forms.

Key words: Maple, Newton-Coates formula, Trapezoid, Simpson's Rule, Simpson's 3/8 Rule.

Introduction

Computing software has advanced a lot in recent decades. Today, computing software is used not only in specialized work, but also in educational affairs, academic research and textbook writing. Founded in the early 1980s at the University of Waterloo, Canada, Maple software, in newer versions, offers many computational problems with accurate answers. This software is easy to learn because the same math symbols used in classrooms can be used to enter data. Numerical analysis is one of the most important methods that is very important for performance in many algorithms. Modern numerical analysis tries to understand the data in a shorter or more concise way. Any approximate method must converge to the correct answer [7].

Today, using the latest applications of numerical analysis, they try to calculate the data in a shorter and more efficient way. The Maple package provides numerical methods that lead to the solution of important scientific and technical problems. Maple package also has features for input and output of image and output files. We analyzed the Newton-Coates formula in the Maple package considering other methods. Integration is the technique of calculating the area plotted on a graph using a function [1].

$$I = \int_a^b f(x)\,dx$$

Materials and methods

Because of these tasks, interpolation-based numerical integration using the following methods is accurately demonstrated by Maple.

A. NEWTON–COTES FORMULA

We study Newton-Coates formulas using some Maple computational capabilities with respect to the approximate function at distances [a, b] for the integral f (x) [2].

Examples

```
> int(ln(x), x = 1.1 .. 8.0);
```

$$9.630691136$$

```
> with(Student[Calculus1]):
> ApproximateInt(ln(x), x = 1.1 .. 8.0, method = newtoncotes[1]);
```

$$9.600011676$$

```
> ApproximateInt(ln(x), x = 1.1 .. 8.0, method = newtoncotes[2]);
```

$$9.630586303$$

```
> ApproximateInt(ln(x), x = 1.1 .. 8.0, method = newtoncotes[3]);
```

9.630643941

> $ApproximateInt(\ln(x), x = 1.1..8.0, method = newtoncotes[4]);$

9.630690519

> $ApproximateInt(\ln(x), x = 1.1..8.0, method = newtoncotes[6]);$

9.630691129

B. TRAPEZOIDAL METHODS

The trapezoidal rule is a Newton-Cotes formula for approximating the integral of a function f using linear segments. Let f be tabulated at points x_0 and x_1 spaced by a distance h, and write $f_n = f(x_n)$. then the trapezoidal rule states that $\int_{x_0}^{x_1} f(x)\,dx \approx h(f_0 + f_1)/2$.

Examples

> $polynomial := CurveFitting[PolynomialInterpolation]([x_0, x_1], [f(0), f(1)], z)$:

> $integrated := \int_{x_0}^{x_1} polynomial\,dz$:

> $factor(integrated)$

$$-\frac{1}{2}(x_0 - x_1)(f(1) + f(0))$$

> $with(Student[Calculus1])$:

> $ApproximateInt(\sin(x), x = 3..5, method = trapezoid)$

$$\frac{1}{10}\sin(3) + \frac{1}{5}\sin\left(\frac{16}{5}\right) + \frac{1}{5}\sin\left(\frac{17}{5}\right) + \frac{1}{5}\sin\left(\frac{18}{5}\right) + \frac{1}{5}\sin\left(\frac{19}{5}\right) + \frac{1}{5}\sin(4) + \frac{1}{5}\sin\left(\frac{21}{5}\right) + \frac{1}{5}\sin\left(\frac{22}{5}\right) + \frac{1}{5}\sin\left(\frac{23}{5}\right) + \frac{1}{5}\sin\left(\frac{24}{5}\right) + \frac{1}{10}\sin(5)$$

> $ApproximateInt(\cos(x), 1..100, method = trapezoid, output = animation)$

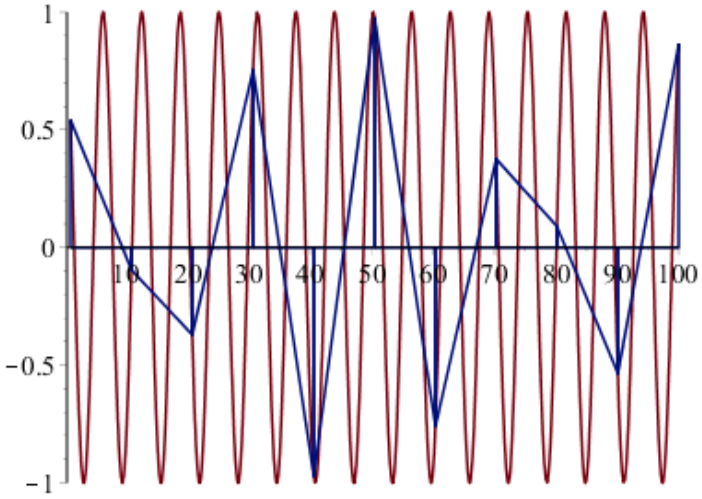

Figure 1 An animated approximation of $\int_1^{100} f(x)\,dx$ using trapezoid rule, where $f(x) = \cos(x)$ and the partition is uniform.
The approximate value of the integral is 1.615815314.
Number of subintervals used: 10.

C. SIMPSON'S RULE

Simpson's Rule is a numerically accurate method of approximating a definite integral usi
ng

a three - point quadrature obtained by integrating the unique quadratic that passes through these points [3,5]. for some c in the interval $[x_0, x_2]$, provided that $f\ (iv)$ exists and is continuous. Concluding the derivation yields Simpson's Rule [2,6]:

$$\int_{x_0}^{x_2} f(x)dx = \frac{h}{2}(y_0 + 4y_1 + y_2) - \frac{h^5}{90}f^{(iv)}(c).$$

Where $h = x_2 - x_1 = x_1 - x_0$ and c between x_0 and x_2.

Examples

> $polynomial := CurveFitting[PolynomialInterpolation]\left(\left[x_0, \frac{x_0 + x_1}{2}, x_1\right], \left[f(0), f\left(\frac{1}{2}\right), f(1)\right], z\right)$

> $integrated := \int_{x_0}^{x_1} polynomial\, dz$:

> $factor(integrated)$

$$-\frac{1}{6}(x_0 - x_1)\left(f(1) + 4f\left(\frac{1}{2}\right) + f(0)\right)$$

> $with\left(Student[Calculus1]\right)$:

> $ApproximateInt(\sin(x), x = 2..-3, method = simpson)$

$$-\frac{1}{6}\sin\left(\frac{5}{2}\right) - \frac{1}{3}\sin\left(\frac{11}{4}\right) - \frac{1}{12}\sin(3) - \frac{1}{12}\sin(2) - \frac{1}{3}\sin\left(\frac{9}{4}\right)$$

> $ApproximateInt\left(\cos(x), 1..100, method = simpson, output = animation\right)$

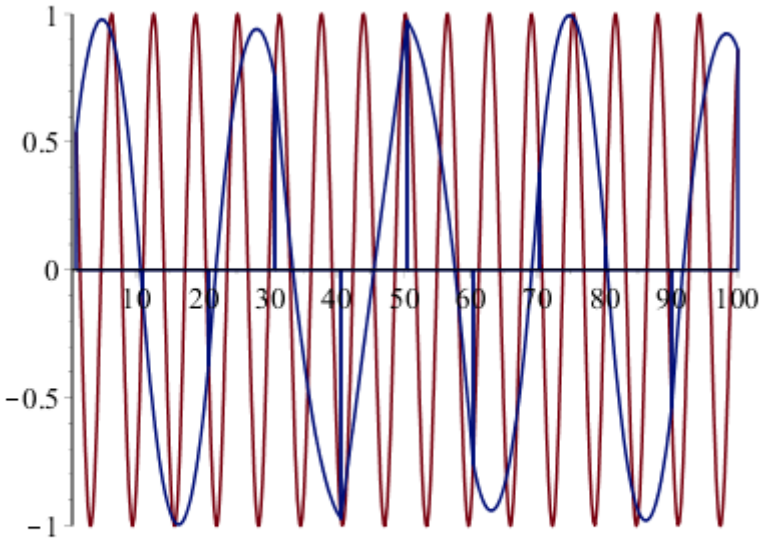

Figure 2 An animated approximation of $\int_1^{100} f(x)\, dx$ using Simpson's rule, where $f(x) = \cos(x)$ and the partition is uniform. The approximate value of the integral is 5.115050025. Number of subintervals used: 10.

D. THE GENERAL FORMULA OF SIMPSON'S 3/8 RULE

Let the values of a function $f(x)$ be tabulated at points x_i equally spaced by $h = x_{i+1} - x_i$, so $f_1 = f(x_1)$, $f_2 = f(x_2)$, ..., $f_4 = f(x_4)$. Then Simpson's 3/8 rule approximating the

integral of $f(x)$ is given by the Newton-Cotes-like formula [7].

$$\int_{x_1}^{x_4} f(x)dx = \frac{3}{8}h(f_1 + 3f_3 + f_4) - \frac{3}{8}h^5(\xi).$$

This method is completely based on cubic interpolation [8]. With this method, we demonstrated some computational capabilities of the Maple package [4].

Examples

```
> polynomial := CurveFitting[PolynomialInterpolation]([x0, (2*x0 + x1)/3, (x0 + 2*x1)/3, x1], [f(0), f(1/3), f(2/3), f(1)], z):
> integrated := int(polynomial, z = x0 .. x1):
> factor(integrated)
```

$$-\frac{1}{8}(x_0 - x_1)\left(f(1) + 3f\left(\frac{2}{3}\right) + 3f\left(\frac{1}{3}\right) + f(0)\right)$$

```
> with(Student[Calculus1]):
> ApproximateInt(sin(x), x = 1 ..6, method = simpson[3/8])
```

$$\frac{1}{8}\sin\left(\frac{11}{2}\right) + \frac{3}{16}\sin\left(\frac{17}{3}\right) + \frac{3}{16}\sin\left(\frac{35}{6}\right) + \frac{1}{16}\sin(6) + \frac{3}{16}\sin\left(\frac{31}{6}\right) + \frac{3}{16}\sin\left(\frac{16}{3}\right) + \frac{1}{8}\sin(5) + \frac{3}{16}\sin\left(\frac{23}{6}\right) + \frac{1}{8}\sin(4) + \frac{3}{16}\sin\left(\frac{25}{6}\right) + \frac{3}{16}\sin\left(\frac{13}{3}\right) + \frac{1}{8}\sin\left(\frac{9}{2}\right) + \frac{3}{16}\sin\left(\frac{14}{3}\right) + \frac{3}{16}\sin\left(\frac{29}{6}\right) + \frac{3}{16}\sin\left(\frac{17}{6}\right) + \frac{1}{8}\sin(3) + \frac{3}{16}\sin\left(\frac{19}{6}\right) + \frac{3}{16}\sin\left(\frac{10}{3}\right) + \frac{1}{8}\sin\left(\frac{7}{2}\right) + \frac{3}{16}\sin\left(\frac{11}{3}\right) + \frac{3}{16}\sin\left(\frac{5}{3}\right) + \frac{3}{16}\sin\left(\frac{11}{6}\right) + \frac{1}{8}\sin(2) + \frac{3}{16}\sin\left(\frac{13}{6}\right) + \frac{3}{16}\sin\left(\frac{7}{3}\right) + \frac{1}{8}\sin\left(\frac{5}{2}\right) + \frac{3}{16}\sin\left(\frac{8}{3}\right) + \frac{3}{16}\sin\left(\frac{7}{6}\right) + \frac{3}{16}\sin\left(\frac{4}{3}\right) + \frac{1}{8}\sin\left(\frac{3}{2}\right) + \frac{1}{16}\sin(1)$$

```
> ApproximateInt(cos(x), 1 ..100, method = simpson[3/8], output = animation)
```

The results show that the Maple package has good computational capabilities and can be very useful for analyzing generalized numerical integration methods of a software. We tested some of the computational capabilities of the Maple package using the Newton-Coates formula for numerical integration.

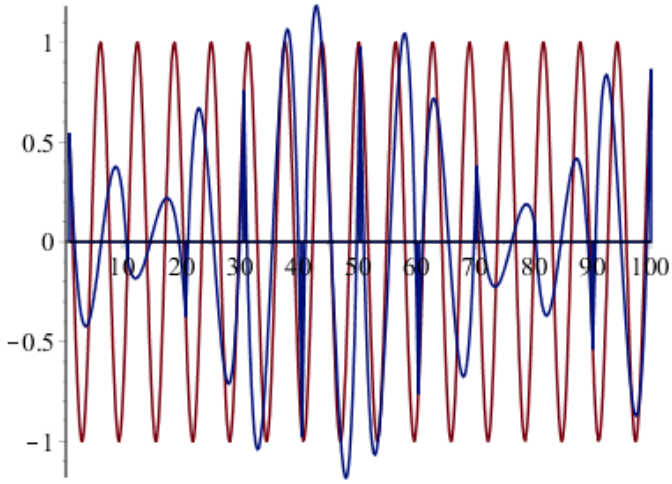

Figure 3 An animated approximation of $\int_1^{100} f(x)\,dx$ using Simpson's 3/8 rule, where $f(x) = \cos(x)$ and the partition is uniform. The approximate value of the integral is −0.0034001 10048. Number of subintervals used: 10.

Conclusion

The results show that the Maple package has good computational capabilities and can be very useful for analyzing generalized numerical integration methods of a software. We tested some of the computational capabilities of the Maple package using the Newton-Coates formula for numerical integration.

References

[1] Stoer, Josef, and Roland Bulirsch. Introduction to numerical analysis. Springer Science & Business Media, 2013. 118. Vol. 12.

[2] Levy, Doron. Introduction to numerical analysis. Department of Mathematics and Center for Scientific Computation and Mathematical Modeling (CSCAMM) University of Maryland. 2010.

[3] Süli, Endre, and David F. Mayers. An introduction to numerical analysis. Cambridge university press, 2003. 201.

[4] Sastry, Shankar S. Introductory methods of numerical analysis. PHI Learning Pvt. Ltd. 2012. 219-222.

[5] Rao, K. Sankara. Numerical methods for scientists and engineers. PHI Learning Pvt. Ltd., 2017.151.

[6] Isaacson, Eugene, and Herbert Bishop Keller. Analysis of numerical methods. Courier Corporation, 2012. 308.

[7] Sauer, Timothy. Numerical Analysis Pearson Addison Wesley. 2012. 254-257.

[8] Heister, Timo, Leo G. Rebholz, and Fei Xue. Numerical Analysis: An Introduction. Walter de Gruyter GmbH & Co KG. 2019. 224.

MAPLE-ГЕ САНДЫҚ ИНТЕГРАЦИЯҒА АРНАЛҒАН НЬЮТОН-КОТС ФОРМУЛАЛАРЫ

Yalda Qani
Фаряб университетінің математика факультеті, Майман, Ауғанстан
e-mail: yalda.qani@gmail.com

Аңдатпа. Бұл мақалада қазірдің өзінде талқыланған Maple пакетін сандық интеграциялауға арналған Ньютон-Коутс формулалары қарастырылады. Бұл зерттеуде біз сандық интеграция мақсатында трапеция әдісін, Симпсон әдісін және Симпсон 3/8 әдісін қарастырдық. Біз f (x) нүктесінде [A, b] интервалында анықталған f функциясын талдадық, бірнеше талдауларды ұсындық. Бүгінгі таңда Maple пакеті сандық талдаудың соңғы қосымшаларын қысқа және тиімді түрде ұсынады, бұл маңызды ғылыми және техникалық мәселелерді шешуге әкелді. Maple

бағдарламалық жасақтамасының көмегімен біз оның түсінікті және сенімді формаларына қажетті өзгерістер енгіздік.

Түйін сөздер: Үйеңкі, Ньютон-Коутс формуласы, Трапеция, Симпсон ережесі, Симпсон ережесі 3/8.

ФОРМУЛЫ НЬЮТОНА–КОТСА ДЛЯ ЧИСЛЕННОГО ИНТЕГРИРОВАНИЯ В MAPLE

Yalda Qani
Математический факультет Фарьябского университета, Маймана, Афганистан
yalda.qani@gmail.com

Абстракт. В этой статье обсуждаются формулы Ньютона-Коутса для численного интегрирования пакета Maple, которые уже обсуждались. В этом исследовании мы рассмотрели трапециевидного метода, метода Симпсона и метода Симпсона 3/8 с целью численного интегрирования. Мы проанализировали функцию f, определенную в интервале [a, b] в точке f (x), представив несколько анализов. Сегодня пакет Maple предоставляет новейшие приложения численного анализа более коротким и эффективным способом, что привело к решению важных научных и технических проблем. С помощью программного обеспечения Maple мы внесли желаемые изменения в его понятные и надежные формы.

Ключевые слова: Клен, формула Ньютона-Коутса, Трапеция, Правило Симпсона, Правило Симпсона 3/8.

Сведения об авторе

Анг.: Yalda Qani- Assistant Professor in the Department of Mathematics, Faryab University, Maymana, Afghanistan

Каз.: Yalda Qani- Фаряб университетінің математика кафедрасының доценті, Майман, Ауғанстан.

Рус.: Yalda Qani- Доцент кафедры математики Фарьябского университета, Маймана, Афганистан.

UDC 519.635
IRSTI 27.35.45

ACCOUNTING FOR THE TEMPERATURE DISTRIBUTION OF A BODY IN THE FORM OF A RECTANGULAR PARALLELEPIPED, TAKING INTO ACCOUNT HEAT TRANSFER USING A VARIATIONAL APPROACH

Z. Omiotec, A.A. Tashev, R.K. Kazykhan
Lublin University of Technology, Lublin, Poland
Institute of Information and Computing Technologies, Almaty, Kazakhstan,
Al-Farabi Kazakh National University, Almaty, Kazakhstan,
z.omiotek@pollub.pl, *azattash@mail.ru*, *rysgulkazykhan@gmail.com*
ORCID ID: https://orcid.org/0000-0002-6614-7799
ORCID ID: https://orcid.org/0000-0003-1427-198X
ORCID ID: https://orcid.org/0000-0002-4350-6322

Abstract. Finite element method is a numerical method solutions of differential equations, found in physics and technology. The emergence of this method associated with solving problems of space research. It was first published in the work of Turner, Cluj, Martin and Topp. This work contributed to the emergence of other works; a number of articles have been published with examples of the finite element method to the problems of construction mechanics of continuous media. The main idea behind the finite element method is that any continuous quantity such,like temperature pressure and displacement, can be approximated by a discrete model, which is built into a set of piece-continuous functions defined on a finite number of subdomains.

The main idea of the finite element method is that any continuous quantity, such as temperature, pressure and displacement, can be approximated by a discrete model, which is built on a set of piece-continuous functions defined on a finite number of subdomains.

Keywords: Finite element, pipe, motion, cross section, compression, lubrication, system, vibration, equilibrium, continuous value, discrete model, node.

Introduction

In the general case, a continuous quantity is not known in advance, and it is necessary to determine the value of this quantity at some internal points of the region. A discrete model, however, is very easy to construct if one first assumes that the numerical value of this quantity at each interior point of the region is known [1]. After that, we can pass to the general case. So, when constructing a discrete model of a continuous quantity, proceed as follows:

1. A finite number of points are fixed in the area under consideration. These points are called anchor points or simply nodes.

2. The value of the continuous quantity at each nodal point is considered a variable to be defined.

3. The domain of definition of a continuous value is divided into a finite number of subdomains, called elements. These elements have common nodal points and together approximate the shape of the regions.

4. A continuous value is approximated at each element by a polynomial, which is determined using the nodal values of this value. For each element, its own polynomial is determined, but the polynomials are selected in such a way that the continuity of the value along the boundaries of the element would be preserved [2].

The finite element method is based on the idea of approximating a continuous function by a discrete model, which is built on a set of piecewise continuous functions defined on a finite

number of subdomains called elements. A polynomial is most often used as a function of an element. The order of the polynomial depends on the number of continuous function data elements used at each node.

Main part

A one-dimensional simplex element is a straight line segment of length L with two nodes, one at each end of the segment [3].

$$T = \varphi_1 T_1 + \varphi_2 T_2$$

$$\varphi_1 = \frac{x_2 - x}{L} \qquad \varphi_2 = \frac{x - x_1}{L}$$

A two-dimensional simplex element is a triangle with straight sides and three nodes, one for each vertex. Logical numbering of element nodes is required.

$$T = \varphi_1 T_1 + \varphi_2 T_2 + \varphi_3 T_3$$

$$\varphi_1 = \frac{1}{2A}(a_1 + b_1 x + c_1 y)$$

$$\begin{cases} a_1 = x_2 y_3 - x_3 y_2 \\ b_1 = y_2 - y_3 \\ c_1 = x_3 - x_2 \end{cases}$$

$$\varphi_2 = \frac{1}{2A}(a_2 + b_2 x + c_2 y)$$

$$\begin{cases} a_2 = x_3 y_1 - y_3 y_1 \\ b_2 = y_3 - y_1 \\ c_2 = x_1 - x_3 \end{cases}$$

$$\varphi_3 = \frac{1}{2A}(a_3 + b_3 x + c_3 y)$$

$$\begin{cases} a_3 = x_1 y_2 - x_2 y_1 \\ b_3 = y_1 - y_2 \\ c_3 = x_2 - x_1 \end{cases}$$

$$2A = \begin{vmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{vmatrix}$$

We consider the three-dimensional simplex element [4]:

$$T = \varphi_1 T_1 + \varphi_2 T_2 + \varphi_3 T_3 + \varphi_4 T_4$$

Consider the three-dimensional function T(x, y, z), whose value is given at the corner points of the parallelepiped (1,2,…8) (1-T, 2-T,…8-T) (Fig. 1)

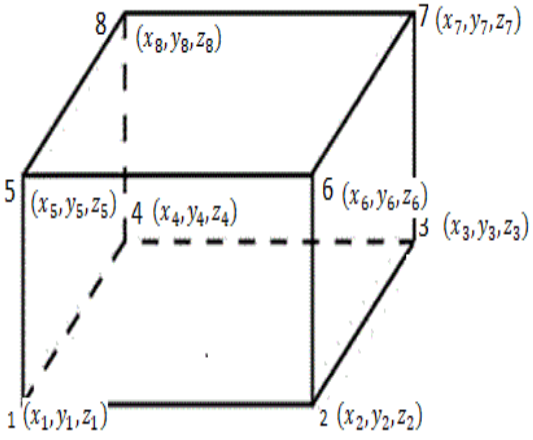

Figure – 1. parallelepiped

If the length of the parallelepiped along the x→ 2a, along the y→ 2b, and along the z→ 2c,then the coordinates of the corner points relative to the center of the parallelepiped will be[5].

1- (-a; в; -c)= (x_1, y_1, z_1)
2- (a; в; -c)= (x_2, y_2, z_2)
3- (a; -в; -c)= (x_3, y_3, z_3)
4- (-a; -в; -c)= (x_4, y_4, z_4)

The value of the function T (x, y, z) to an arbitrary point (x, y, z) inside the parallelepiped isapproximated as follows.

$$\mathrm{T}(\mathrm{x},\mathrm{y},\mathrm{z}) = \lambda_1 + \lambda_2 x_1 + \lambda_3 y_1 + \lambda_4 z_1 \qquad (1)$$

To determine the constants A_i (i = 1; 8), we compose the following control system[6]:

$$T_1 = \mathrm{T}(x_1, y_1, z_1);\ T_2 = \mathrm{T}(x_2, y_2, z_2);\ T_3 = \mathrm{T}(x_3, y_3, z_3);$$
$$T_4 = \mathrm{T}(x_4, y_4, z_4); \qquad (2)$$

Substituting the values of the arguments, we get the system of equations:

$$\begin{cases} \lambda_1 + \lambda_2 x_1 + \lambda_3 y_1 + \lambda_4 z_1 = \mathrm{T}_1 \\ \lambda_1 + \lambda_2 x_2 + \lambda_3 y_2 + \lambda_4 z_2 = \mathrm{T}_2 \\ \lambda_1 + \lambda_2 x_3 + \lambda_3 y_3 + \lambda_4 z_3 = \mathrm{T}_3 \\ \lambda_1 + \lambda_2 x_4 + \lambda_3 y_4 + \lambda_4 z_4 = \mathrm{T}_4 \end{cases} \qquad (3)$$

Solving this system of linear equations, we obtain the values of the coefficients $\lambda_1, \lambda_2 \ldots \lambda_5$

$$\begin{cases} \Lambda_1 = \frac{\mathrm{T}_8+\mathrm{T}_7+\mathrm{T}_6+\mathrm{T}_5+\mathrm{T}_4+\mathrm{T}_3+\mathrm{T}_2+\mathrm{T}_1}{8} \\ \Lambda_2 = \frac{-\mathrm{T}_8+\mathrm{T}_7+\mathrm{T}_6-\mathrm{T}_5-\mathrm{T}_4+\mathrm{T}_3+\mathrm{T}_2-\mathrm{T}_1}{8\mathrm{a}} \\ \Lambda_3 = \frac{-T_8-T_7+T_6+T_5-T_4-T_3+T_2+T_1}{8b} \\ \Lambda_4 = \frac{\mathrm{T}_8+\mathrm{T}_7+\mathrm{T}_6+\mathrm{T}_5-\mathrm{T}_4-\mathrm{T}_3-\mathrm{T}_2-\mathrm{T}_1}{8\mathrm{c}} \end{cases} \qquad (7)$$

(7) substituting these values into equation (2) we obtain:

$$\mathrm{T}(\mathrm{x},\mathrm{y},\mathrm{z}) = \varphi_1(\mathrm{x},\mathrm{y},\mathrm{z}) * T_1 + \varphi_2(\mathrm{x},\mathrm{y},\mathrm{z}) * T_2 + \varphi_3(\mathrm{x},\mathrm{y},\mathrm{z}) * T_3 + \varphi_4 T_4 + \varphi_5(\mathrm{x},\mathrm{y},\mathrm{z}) * T_5 + \varphi_6(\mathrm{x},\mathrm{y},\mathrm{z}) * T_6 + \varphi_7(\mathrm{x},\mathrm{y},\mathrm{z}) * T_7 + \varphi_8(\mathrm{x},\mathrm{y},\mathrm{z}) * T_8; \qquad (4)$$

-a ≤ x ≤a; -b ≤x ≤b; -c ≤ x≤ c;

Here φ_i(i= 1, 8) are defined as follows: [7]

$$\varphi_1(\mathrm{x},\mathrm{y},\mathrm{z}) = \left(\frac{1}{8} - \frac{x}{8a} + \frac{y}{8b} - \frac{z}{8c} - \frac{xy}{8ab} + \frac{xz}{8ac} - \frac{yz}{8bc} + \frac{xyz}{8abc}\right);$$
$$\varphi_2(\mathrm{x},\mathrm{y},\mathrm{z}) = \left(\frac{1}{8} + \frac{x}{8a} + \frac{y}{8b} - \frac{z}{8c} + \frac{xy}{8ab} - \frac{xz}{8ac} - \frac{yz}{8bc} - \frac{xyz}{8abc}\right);$$
$$\varphi_3(\mathrm{x},\mathrm{y},\mathrm{z}) = \left(\frac{1}{8} + \frac{x}{8a} - \frac{y}{8b} - \frac{z}{8c} - \frac{xy}{8ab} - \frac{xz}{8ac} + \frac{yz}{8bc} + \frac{xyz}{8abc}\right);$$
$$\varphi_4(\mathrm{x},\mathrm{y},\mathrm{z}) = \left(\frac{1}{8} - \frac{x}{8a} - \frac{y}{8b} - \frac{z}{8c} + \frac{xy}{8ab} + \frac{xz}{8ac} + \frac{yz}{8bc} - \frac{xyz}{8abc}\right); \qquad (5)$$
$$\varphi_5(\mathrm{x},\mathrm{y},\mathrm{z}) = \left(\frac{1}{8} - \frac{x}{8a} + \frac{y}{8b} + \frac{z}{8c} - \frac{xy}{8ab} - \frac{xz}{8ac} + \frac{yz}{8bc} - \frac{xyz}{8abc}\right);$$
$$\varphi_6(\mathrm{x},\mathrm{y},\mathrm{z}) = \left(\frac{1}{8} + \frac{x}{8a} + \frac{y}{8b} + \frac{z}{8c} + \frac{xy}{8ab} + \frac{xz}{8ac} + \frac{yz}{8bc} + \frac{xyz}{8abc}\right);$$
$$\varphi_7(\mathrm{x},\mathrm{y},\mathrm{z}) = \left(\frac{1}{8} + \frac{x}{8a} - \frac{y}{8b} + \frac{z}{8c} - \frac{xy}{8ab} + \frac{xz}{8ac} - \frac{yz}{8bc} - \frac{xyz}{8abc}\right);$$
$$\varphi_8(\mathrm{x},\mathrm{y},\mathrm{z}) = \left(\frac{1}{8} - \frac{x}{8a} - \frac{y}{8b} + \frac{z}{8c} + \frac{xy}{8ab} - \frac{xz}{8ac} - \frac{yz}{8bc} + \frac{xyz}{8abc}\right);$$

why -a ≤ x ≤a; -b ≤y ≤b; -c ≤ z≤ c.

The value of the functions φ_i(x, y, z) (i=1,8) at the corner points of the parallelepiped isdetermined as follows[8]:

$\varphi_1(x_1, y_1, z_1) = 1;\ \varphi_1(x_1, y_1, z_1) = \varphi_1(x_2, y_2, z_2) = \varphi_1(x_3, y_3, z_3) = \varphi_1(x_4, y_4, z_4) = \varphi_1(x_5,$

$y_5, z_5)= \varphi_1(x_6, y_6, z_6)= \varphi_1(x_7, y_7, z_7)= \varphi_1(x_8, y_8, z_8)=0$

$\varphi_2(x_2, y_2, z_2)= 1;\ \varphi_2(x_1, y_1, z_1)= \varphi_2(x_2, y_2, z_2)= \varphi_2(x_3, y_3, z_3)= \varphi_2(x_4, y_4, z_4)= \varphi_2(x_5, y_5, z_5)= \varphi_2(x_6, y_6, z_6)= \varphi_2(x_7, y_7, z_7)= \varphi_2(x_8, y_8, z_8)=0$

$\varphi_3(x_3, y_3, z_3)= 1;\ \varphi_3(x_1, y_1, z_1)= \varphi_3(x_2, y_2, z_2)= \varphi_3(x_3, y_3, z_3)= \varphi_3(x_4, y_4, z_4)= \varphi_3(x_5, y_5, z_5)= \varphi_3(x_6, y_6, z_6)= \varphi_3(x_7, y_7, z_7)= \varphi_3(x_8, y_8, z_8)=0$

$\varphi_4(x_4, y_4, z_4)= 1;\ \varphi_4(x_1, y_1, z_1)= \varphi_4(x_2, y_2, z_2)= \varphi_4(x_3, y_3, z_3)= \varphi_4(x_4, y_4, z_4)= \varphi_4(x_5, y_5, z_5)= \varphi_4(x_6, y_6, z_6)= \varphi_4(x_7, y_7, z_7)= \varphi_4(x_8, y_8, z_8)=0$

$\varphi_5(x_5, y_5, z_5)= 1;\ \varphi_5(x_1, y_1, z_1)= \varphi_5(x_2, y_2, z_2)= \varphi_5(x_3, y_3, z_3)= \varphi_5(x_4, y_4, z_4)= \varphi_5(x_5, y_5, z_5)= \varphi_5(x_6, y_6, z_6)= \varphi_5(x_7, y_7, z_7)= \varphi_5(x_8, y_8, z_8)=0$

$\varphi_6(x_6, y_6, z_6)= 1;\ \varphi_6(x_1, y_1, z_1)= \varphi_6(x_2, y_2, z_2)= \varphi_6(x_3, y_3, z_3)= \varphi_6(x_4, y_4, z_4)= \varphi_6(x_5, y_5, z_5)= \varphi_6(x_6, y_6, z_6)= \varphi_6(x_7, y_7, z_7)= \varphi_6(x_8, y_8, z_8)=0$

$\varphi_7(x_7, y_7, z_7)= 1;\ \varphi_7(x_1, y_1, z_1)= \varphi_7(x_2, y_2, z_2)= \varphi_7(x_3, y_3, z_3)= \varphi_7(x_4, y_4, z_4)= \varphi_7(x_5, y_5, z_5)= \varphi_7(x_6, y_6, z_6)= \varphi_7(x_7, y_7, z_7)= \varphi_7(x_8, y_8, z_8)=0$

$\varphi_8(x_8, y_8, z_8)= 1;\ \varphi_8(x_1, y_1, z_1)= \varphi_8(x_2, y_2, z_2)= \varphi_8(x_3, y_3, z_3)= \varphi_8(x_4, y_4, z_4)= \varphi_8(x_5, y_5, z_5)= \varphi_8(x_6, y_6, z_6)= \varphi_8(x_7, y_7, z_7)= \varphi_8(x_8, y_8, z_8)=0$ (6)

Conclusion

In the result, there is obtained common variation functional for defining the law of temperature distribution in the body of rectangular 3D cube form. Results of this work can be used to determine the law of temperature distribution in three-dimensional rods in the form of a parallelepiped. There is executed practical implementation of the developed approach as a concrete example, when heat flux is directed to one of 3D cube's facets, and on the opposite facet there occurs heat exchange with the environment. There was shown temperature values proximity of corresponding nodes of rectangular 3D cube upon separation of its facets for one, two and three intervals.

References

[1] Rogié B., et al., Practical analytical modeling of 3D multi-layer Printed Wired Board with buried volumetric heating sources. International Journal of Thermal Sciences. 2018. 129. 404–415.

[2] Xu, G. Y., Wang, J. B. Analytical solution of time fractional Cattaneo heat equation for finite slab under pulse heat flux. Applied Mathematics and Mechanics. 2018. 39(10). 1465–1476.

[2] Forslund R, et al. Analytical solution for heat conduction due to a moving Gaussian heat flux with piecewise constant parameters. Applied Mathematical Modelling. 2019. 66. 227–240.

[3] França, M. V., Orlande, H. R. B. Estimation of parameters of the dual-phase-lag model for heat conduction in metal-oxide-semiconductor field-effect transistors. International Communications in Heat and Mass Transfer. 2018. 92. 107–111.

[4] Shen Y., et al. Effect of non-condensable gas on heat conduction in steam sterilization process. Thermal Science. 2019. 23(4). 2489–2494.

[5] Haji-Sheikh A., Beck J. V., Temperature solution in multi-dimensional multi-layer bodies. International Journal of Heat and Mass Transfer. 2002. 45(9). 1865–1877.

[6] Aviles-Ramos C., et al. Exact solution of heat conduction in composite materials and application to inverse problems. Journal of Heat Transfer.1998. 120(3). 592–599.

[7] Beck J. V., et al. Verification solution for partial heating of rectangular solids. International Journal of Heat and Mass Transfer. 2004. 47(19–20). 4243–4255.

ВАРИАЦИЯЛЫҚ ТӘСІЛДІ ПАЙДАЛАНҒАН ЖЫЛУ ТАРТЫЛУЫН ЕСКЕ АЛУ МЕН ТӨРТ БҰРЫШТЫ ПАРАЛЛЕЛЕППЕТ ПІШІНДЕГІ ДЕНЕ ТЕМПЕРАТУРАСЫНЫҢ БӨЛУІНЕ ЕСЕП

З. Омиотек, А.А. Ташев, Р.Қ. Қазыхан

Люблин технологиялық университеті, Люблин, Польша
Ақпараттық және есептеуіш технологиялар институты, Алматы, Қазақстан,
әл-Фараби атындағы Қазақ ұлттық университеті, Алматы, Қазақстан,

z.omiotek@pollub.pl, *azattash@mail.ru*, *rysgulkazykhan@gmail.com*
ORCID ID: https://orcid.org/0000-0002-6614-7799
ORCID ID: https://orcid.org/0000-0003-1427-198X
ORCID ID: https://orcid.org/0000-0002-4350-6322

Аңдатпа. Ақырлы элементтер әдісі – физика мен техникада кездесетін дифференциалдық теңдеулерді шешудің сандық әдісі. Бұл әдістің пайда болуы ғарышты зерттеу мәселелерін шешумен байланысты. Ол алғаш рет Тернер, Клуж, Мартин және Топптың жұмысында жарияланған. Бұл еңбек басқа еңбектердің пайда болуына ықпал етті; Үздіксіз ортаның құрылыс механикасы мәселелеріне соңғы элементтер әдісінің мысалдарымен бірқатар мақалалар жарияланды. Ақырлы элементтер әдісінің негізгі идеясы - температура қысымы және орын ауыстыру сияқты кез келген үздіксіз шаманы ішкі домендердің шектеулі санында анықталған үзіліссіз функциялар жиынтығына кіріктірілген дискретті модель арқылы жуықтауға болады.

Ақырлы элементтер әдісінің негізгі идеясы температура, қысым және орын ауыстыру сияқты кез келген үздіксіз шаманы ішкі домендердің шектеулі санында анықталған үзіліссіз функциялар жиынтығына құрылған дискретті модель арқылы жуықтауға болады.

Түйін сөздер: Ақырлы элемент, құбыр, қозғалыс, қима, қысу, майлау, жүйе, діріл, тепе-теңдік, үздіксіз шама, дискретті модель, түйін.

УЧЕТ РАСПРЕДЕЛЕНИЯ ТЕМПЕРАТУРЫ ТЕЛА В ВИДЕ ПРЯМОУГОЛЬНОГО ПАРАЛЛЕПИПЕДА С УЧЕТОМ ТЕПЛООТДАЧИ С ИСПОЛЬЗОВАНИЕМ ВАРИАЦИОННОГО ПОДХОДА

З. Омиотек, А.А. Ташев, Р.К. Казыхан
Люблинский технический университет, Люблин, Польша
Институт информационных и вычислительных технологий, Алматы, Казахстан,
Казахский национальный университет им. аль-Фараби, Алматы, Казахстан,
z.omiotek@pollub.pl, *azattash@mail.ru*, *rysgulkazykhan@gmail.com*
ORCID ID: https://orcid.org/0000-0002-6614-7799
ORCID ID: https://orcid.org/0000-0003-1427-198X
ORCID ID: https://orcid.org/0000-0002-4350-6322

Абстракт. Метод конечных элементов представляет собой численный метод решения дифференциальных уравнений, встречающихся в физике и технике. Возникновение этого метода связано с решением задач космических исследований. Впервые он был опубликован в работе Тернера, Клужа, Мартина и Топпа. Эта работа способствовала появлению других работ; опубликован ряд статей с примерами применения метода конечных элементов к задачам строительной механики сплошных сред. Основная идея метода конечных элементов заключается в том, что любая непрерывная величина, такая как температура, давление и перемещение, может быть аппроксимирована дискретной моделью, которая встроена в набор кусочно-непрерывных функций, определенных в конечном числе подобластей.

Основная идея метода конечных элементов состоит в том, что любую непрерывную величину, такую как температура, давление и перемещение, можно аппроксимировать дискретной моделью, построенной на наборе кусочно-непрерывных функций, заданных

на конечном числе подобластей.

Ключевые слова: Конечный элемент, труба, движение, сечение, сжатие, смазка, система, вибрация, равновесие, непрерывная величина, дискретная модель, узел.

Анг.: Zbignev Omiotec - Professor of Lublin University of Technology, Lublin, Poland.

Каз.: Збигнев Омиотек - Люблин технологиялық университетінің профессоры, Люблин, Польша.

Рус.: Збигнев Омиотек – Профессор Люблинского технологического университета, Люблин, Польша.

Анг.: Tashev Azat Aripovich- Professor, leading researcher of Institute of Information and Computational Technologies, Almaty, Kazakhstan.

Каз.: Ташев Азат Әріпұлы – Профессор, Ақпараттық және есептеуіш технологиялар институтының жетекші ғылыми қызметкері, Алматы, Қазақстан.

Рус.: Ташев Азат Арипович – Профессор, ведущий научный сотрудник Института информационных и вычислительных технологий, Алматы, Казахстан.

Анг.: Kazykhan Rysgul Kyzmetkhankyzy- doctoral student of al-Farabi Kazakh National University, Almaty, Kazakhstan.

Каз.: Қазыхан Рысгүл Қызметханқызы– әл-Фараби атындағы Қазақ ұлттық университетінің докторанты, Алматы, Қазақстан.

Рус.: Казыхан Рысгуль Кызметхановна докторант Казахского Национального университета им. аль-Фараби, Алматы, Казахстан.

UDC 004.588
IRSTI 20.00.00

POST-EDITING FOR THE KAZAKH LANGUAGE

Z. Omiotek[1], D.R. Rakhimova[2], A.Zh. Zhunussova[3]
[1]Lublin University of Technology, Lublin, Poland
[2]Al-Farabi Kazakh National University, Almaty, Kazakhstan
[3]Al-Farabi Kazakh National University, Almaty, Kazakhstan
[1]z.omiotek@pollub.pl, [2]di.diva@mail.ru, [3]alia_94-22@mail.ru
[1]ORCID ID: https://orcid.org/0000-0002-6614-7799
[2]ORCID ID: https://orcid.org/0000-0003-1427-198X
[3]ORCID ID: https://orcid.org/0000-0002-4350-6322

Abstract. The modern world and our immediate future depend on applied intelligent systems, as new technologies develop every day. One of the tasks of intelligent systems is machine (automated) translation from one natural language to another. Machine translation (MT) allows people to communicate regardless of language differences, as it removes the language barrier and opens up new languages for communication. Machine translation is a new technology, a special step in human development. This type of translation can help when you need to quickly understand what your interlocutor wrote or said in a letter.

The work of online translators used to translate into Kazakh and vice versa. Translation errors are identified, general advantages and disadvantages of online machine translation systems in Kazakh are given. A model for the development of a post-editing machine translation system for the Kazakh language is presented.

OpenNMT (Open Neural Machine Translation) is an open source system for neural machine translation and neural sequence training. To learn languages in OpenNMT, you need parallel corpuses for language pairs. The advantage of OpenNMT is that it can be applied to all languages and can handle large corpora. Experimental data were obtained for the English-Kazakh language pair. Experimental data were obtained for the English-Kazakh language pair.

Applied intelligent systems play an important role in the modern world. One of their tasks is machine translation (MT) from one language into another one. MT allows people to freely communicate despite language barriers. This new technology is a special step in helping to understand what a companion speaks, or writes to you. Automatic post-editing is the task of correcting errors present in texts as a result of machine translation. Since MT cannot always give the desired result, it becomes necessary to edit the translation. The drawbacks of the translation have to be eliminated by post- editing. This need for post-editing is largely determined by the quality of machine translation: low-quality translation leaves a lot of room for post-editing, and high-quality and human translations require minimal text editing. This work describes the development of the light post-editing module for the English-Kazakh language pairs. The neural network model is trained on pairs mt, pe and triplets src, mt, pe using the OpenNMT framework. Then the results of the BLEU metrics mt - pe and mt - ape are compared, and a conclusion about the quality of post-editing is made.

Key words: Opennmt, neural machine translation, bleu, automatic post-editing, network model

Introduction

Post-editing is one of the first and superficial stages in the ma chine translation (MT) post-editing system. The light post-editing module includes getting the raw MT output and making as few changes to the text as possible to make the translation clear, accurate in facts, and grammatically correct [1]. The light post-editing module includes the following post-editing tasks:

- correction of only the most obvious typos, vocabulary, and grammatical errors
- correction of machine errors
- removal of unnecessary or redundant translation options created by the machine
- agreement on the key terminology, but without careful check ing of terms

When translating from any language into Kazakh, we use different online translators. Very

popular online translators: Yandex, Google.

But even in these online translators there are errors. Basically, errors are obtained when translating complex sentences.

Also when translating phraseological units. In the online translator, phraseological units are translated to as if. Also, some terms are changed, even if this was not taken into account.

The Kazakh language is a complex language in structure. Therefore, after receiving the translation, human editing is required. This is called automatic post-editing.

There are three kinds of automatic post-editing. Such as full post-editing, medium post-editing, light post-editing. All of these edits are included in automatic post-editing. After the model is created, the translation is automatically edited. Created prior to this model, not all the requirement fulfills the post-editing of the Kazakh language.

Therefore, post-editing of the Kazakh language requires the creation of a model. The article created a transformer model. Also created post-editing model for the Kazakh language can be used for other Turkic languages.

Because these languages are similar in vocabulary and also in the structure of languages. In the following table, you can see the similarity of Turkic words.

Table 1. Turkic words

ol	ol	ʻol	ul	ол jol/	ol	oʻl	ol	рук чол)
emiz	emiz	emiz	imez	емиз semiz/	emiz	emiz	emiz	емис
ncient urkic	urkish	urkmen	urkmen	nKazakh	ygyz	nuzbek	nUyghur	yvan

na	na/anne	ne	na	на/ана/	ne	na	na	ва
urun	urun	urun	orın	murın/	urun	urun	urun	умчук
ol	ol	ol	ul	л qol/	ol	o‘l	ol	ол

But languages are different from other languages, they have special characteristics: proximity of the lexical structure;

- the law of harmony;
- agglutination – a series of affixes;
- lack of a category;
- lack of auxiliary words (prepositions);
- special word order;

Therefore, when translating, the Turkic languages give out morphological, lexical, and semantic errors. This article discusses training parallel corpus in Opennmt. The advantage of Opennmt is its universal applicability to different languages, including the Turkic languages. We also get the results of computational experiments for the Kazakh language. The rest of the paper is organised as follows:

- Section 2 provides an overview of previous work carried out in this area.
- Section 3 presents Opennmt for Kazakh-English, English-Kazakh language pairs.
- Section 4 presents experimental NMT results for Kazakh-English, English-Kazakh language pairs.
- Section 5 presents conclusions and suggests directions for future work.

Previous scientific work

In our country, in Kazakhstan, MT (machine translation) of the Kazakh language has been developing since 2000. Professor U. A. Tukeyev was one of the first to study machine translation. He managed to create a scientific school that is actively engaged in research in the field of MT. Among domestic students, one can note the study of models and methods of semantics of machine translation from Russian into Kazakh language [2], a statistical model of the alignment of English-Kazakh words using the machine translation algorithm [3].

To improve the morphologies of the Turkic languages, vocabulary training on a parallel corpus was used [4-6]. Learning on the parallel corpus of the English-German language pair in Opennmt, studied in foreign works. In this work, 50k vocabulary was learned for each pair and it was shown that in Opennmt Bleu was 17.60 [7]. Opennmt showed a better result than in the Nematus Bleu system by 0.5. Also, the architecture and applicability of Opennmt in other areas were considered.

Automatic post-editing (APE) is an area of research aimed at studying methods that apply editing operations to MT output to obtain better translations. APE covers a wide range of post-editing approaches, from regular expressions applied to MT.

APE systems convert mt, a text received by MT, into mt', a corrected version of the translation in the same language. The APE task can be thought of as a monolingual translation task in which the source and target languages are the same. Thus, APE implementations are very similar to MT systems and even use similar methodologies. However, in the MT scenario, the system learns pairs of sentences src, pe, where src is the texts in the source language, pe is the high-quality texts in the target language. In the APE script, training Such data allows you to successfully train the system to identify both simple spelling and syntax errors, and the consistency of words in the original sentences. The learning process depends on the availability of a sufficient amount of data and high-quality texts in the target pe language.

Algorithm for post-editing

An algorithm is used to post-edit a translation from a machine translation. The algorithm is performed as follows: neuron machine translation from the target language x1: s to the desired language w1:T $p(w_{1:T}|x) = \prod_1^T p(w_t|w_{1:t-1}, x; \theta)$ translates with distribution. Post-editing model w1, w2, .., wt-1 translates source vector words into hidden vector words h1,… hs. The target decoder in the model combines the source words with the vector and the corresponding hidden words, and assumes the probability of the next processed words to appear [8,9]. The maximum probability p (w_t | w_ (1: t-1), x; θ) is then used to obtain a sequence of processed words. A hidden word sequence is needed to link the original word sequence to be translated and the sequence of words to be translated. To maximize the probability of the model, it is necessary to read and study the whole sequence of sentences in the case. Figure 2 shows the training of neuron machine translation model in OpenNMT.

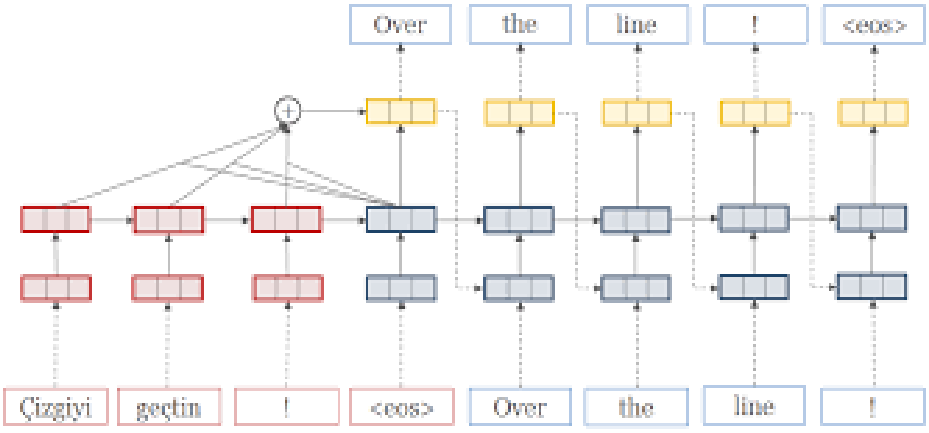

Figure – 1. OpenNMT for learning neuron machine translation model

OpenNMT is an open source generated by neurons. OpenNMT can be downloaded from GitHub (https://github.com/OpenNMT/OpenNMT). Thanks to OpenNMT, data does not take up much memory, machine translation training takes less time. Implementation of training consists of three stages [10]:

1) Data preparation. It consists of words, phrases and sentences in Kazakh and English, created in parallel in two files.

Data preparation is performed using the code onmt_build_vocab -config toy_en_de.yaml -n_sample 10000. The location of the file must be specified. Type in the YAML configuration file:

data:

corpus_1:

```
    path_src: toy-ende/src-train.txt
    path_tgt: toy-ende/tgt-train.txt
  valid:
    path_src: toy-ende/src-val.txt
    path_tgt: toy-ende/tgt-val.txt
```

YAML configuration file Add the following information:

```
Модельді оқыту:
src_vocab: toy-ende/run/example.vocab.src
tgt_vocab: toy-ende/run/example.vocab.tgt
# Train on a single GPU
world_size: 1
gpu_ranks: [0]
# Where to save the checkpoints
save_model: toy-ende/run/model
save_checkpoint_steps: 500
train_steps: 1000
valid_steps: 500
```

In this model we prepare a learning step and a neural model. Then we run this model onmt_train -config toy_en_de.yaml.

3) Translation

In translation, it translates 1000 words or sentences from a pre-rendered file. To translate it, write onmt_translate -model toy-ende / run / model_step_1000.pt -src toy-ende / src-test.txt -output toy-ende / pred_1000.txt -gpu 0 –verbose code.

To create a dictionary for further study, it is necessary to process (tokenize) the data collected in a parallel cas

```
d a t a :
s a v e _ d a t a : OpenNMT / MT_corpus /
en g_ kaz / p r e p r o c e s s e d _ d a t a / m t_kaz/
s r c _ v o c a b : OpenNMT / MT_corpus / en g_ kaz /
p r e p r o c e s s e d _ d a t a / m t_kaz . v ocab . s r c
t g t _ v o c a b : OpenNMT / MT_corpus / en g_ kaz /
p r e p r o c e s s e d _ d a t a / m t_kaz . v ocab . t g t
t r a i n :
p a t h _ s r c : OpenNMT / MT_corpus / APE_da ta /
E n g_ k a z_ d a t a / e n g_ k a z_45000 / m t _ k a z _ t r a i n . t x t
p a t h _ t g t : OpenNMT / MT_corpus / APE_da ta /
E n g_ k a z_ d a t a / e n g_ k a z_45000 / p e _ k a z _ t r a i n . t x t
v a l i d :
p a t h _ s r c : OpenNMT / MT_corpus / APE_da ta /
E n g_ k a z_ d a t a / e n g_ k a z_45000 / m t _ k a z _ v a l i d . t x t
p a t h _ t g t : OpenNMT / MT_corpus / APE_da ta /
E n g_ k a z_ d a t a / e n g_ k a z_45000 / m t _ k a z _ v a l i d . t x t

# S avi n g  c h e c k p o i n t
s a v e _ c h e c k p o i n t _ s t e p s : 2000
```

To obtain a quality result, the t r a n s f o r m e r model was used to train the parallel corps.

50,000 steps were used to teach 45,000 parallel sentences in the case. Commonly used settings can be seen in the following code:

```
# V a l i d a t i o n  s t e p s
v a l i d _ s t e p s : 2000
# T r a i n i n g  s t e p s
t r a i n _ s t e p s : 50000
# O p t i m i z a t i o n
```

```
optim : adam
l e a r n i n g _ r a t e : 2 . 0
# Model
d e c o d e r _ t y p e : t r a n s f o r m e r
 e n c o d e r _ t y p e : t r a n s f o r m e r
```

The experiment was performed on a computer with the following characteristics: CPU - Core i7 4790, 32 GB of RAM, 1 TB SSD and GPU-RTX 2070 Super.

Results

A Kazakh-English language pair was used for training. 109,772 sentences were used in the corpus. These proposals were taken from the website : Akorda, Primeminister, mfa.gov.kz, economy.gov.kz, strategy2050.kz. Of these, it was taken for train 80000, test 20000, validation 9772.It took 36 hours to train at Opennmt.

OpenNMT is a deep learning framework specializing in the Sequence-to-sequence model and covering a wide range of tasks related to machine translation, speech recognition, image- to-text conversion, and more. Corpus training for post-editing is implemented using it. This experiment explores post-editing for an English-Kazakh language pair. Obtained from open sources and using the Bitextor application, 45,000 English-Kazakh parallel sentences were used in this experiment. The parallel corpus for the English-Kazakh language pair was collected from various official sites, and the corpora were aligned and tokenized. The translation As you can see in the table BLEU is less compared to other languages as for English- German, English-French pairs. Because the structure of the language of the Turkic languages is different from these languages. More parallel data is required to improve this metric.

Table 2. The result

Language pair	BLEU(MT)	BLEU(APE)
Kazakh-English	18.45	20.56
English-Kazakh	16.55	20.05

The original src English corpus has been implemented by such machine translation systems as Prompt, Yandex, and Webtran. To train the post-editing model mt - pe , the data was divided into training 80%, validation 10% and test 10%.

The analysis of the results of improving MT was carried out using the bilingual evaluation understudy (BLEU metric). BLEU is an algorithm for evaluating the quality of text translated from one natural language to another. Quality is interpreted as the correspondence between the results of the work of a machine and a person.

Conclusion

Additional experiments showed that OpenNMT improved the quality of translation by creating a model of neural machine translation and processing the Kazakh translation from Google, Yandex online translation. In the article, the model read and processed more than 100 thousand parallel sentences, which showed a much higher rate than after the machine translation. In the article, the Kazakh language was chosen as one of the Turkic-speaking languages as an experiment. As a result of the experiment on the Kazakh-English language pair, after processing, the BLEU value was 20.56. The value of the translation from the unprocessed direct machine translation was 15.70.

Reading the data takes less time and saves memory. As a result of experimental research, it can be seen that the Kazakh-English and English-Kazakh language pairs gave good results in translation. For further research, this model is to obtain translations from other Turkic-speaking languages. For this purpose, a corpus will be assembled for other Turkic-speaking languages.

References

[1] Vyugin V.V. Matematicheskie osnovi mashinnogo obuchenya I prognozirovania. Mathematical Foundations of Machine Learning and Prediction. 2018. 384. (In Russian)

[2] Zhumanov Zh.M., Tukeyev U.A. Development of machine translation software logical model (translation from Kazakh into English language). Reports of the Third Congress of the World Mathematical Society of Turkic Countries, Volume 1 (June 30 – July 4, 2009). – Almaty: Қазақ университеті, 2009. 356-363 p.

[3] Rakhimova D.R. Issledovanie modeli I metodov semantiki mashinnogo perevoda s russkogo na kazakhski yazik. [Study of models and methods of machine translation semantics from Russian into Kazakh]. https://www.kaznu.kz/content/files/pages/folder14360/%D0%94%D0%B8%D1%81%D1%81%D0%B5%D1%80%D1%82%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F%20%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B0%20%D0%A0%D0%B0%D1%85%D0%B8%D0%BC%D0%BE%D0%B2%D0%BE%D0%B9%20%D0%94.%20%D0%A0..pdf (In Russian)

[4] Kartbaev A.Zh. Rasrabotka modeli I metodov statisticheskogo mashinnogo perevoda s prilozheniem k kazakhskomu yaziku. [Development of a model and methods of satistic machine translation with an application to the Kazakh language]. https://www.kaznu.kz/content/files/pages/folder17928/%D0%94%D0%B8%D1%81%D1%81%D0%B5%D1%80%D1%82%D0%B0%D1%86%D0%B8%D1%8F%20%D0%9A%D0%B0%D1%80%D1%82%D0%B1%D0%B0%D0%B5%D0%B2%20%D0%90%D0%96%20%D0%B7%D0%B0%D1%89.pdf (In Russian)

[5] Tukeev U.A., Rakhimova D.R., Baysylbaeva K., Umirbekov N., Orazov B., Abakan M., Kyzyrkanova S.

Kopmaginalik beineleu keste tasili negizinde orys tilinen kazak tiline mashinalik audarmasinin morphologialik analizben sintezin kuru.[Synthesis of morphological analysis of machine translation from Russian into Kazakh on the basis of the method of ambiguous mapping]. 182-191 (in Kazakh)

[6] U. Tukeyev , A. Karibayeva & Z h. Zhumanov|. Morphological segmentation method for Turkic language neural machine translation. 2020

[7] Rakhimova, D., Karyukin, V., Karibayeva, A., Turarbek, A., Turganbayeva, A. The Development of the Light Post-editing Module for English-Kazakh Translation. 2021

[8] Rajen Chatterjee, Gebremedhen Gebremelak, Matteo Negri, and Marco Turchi. Online Automatic Post-editing for MT in a Multi-Domain Translation Environment. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers. Association for Computational Linguistics, Valencia, Spain. 2017. 1. 525–535.

[9] Matteo Negri, Marco Turchi, Rajen Chatterjee, and Nicola Bertoldi. ESCAPE: a large-scale synthetic corpus for automatic post-editing. In In Proceedings of the eleventh international conference on language resources and evaluation (LREC 2018). European Language Resources Association (ELRA), Miyazaki, Japan. 2018. 24–30.

[10] Yu Gong and Demin Yan. A Toolset to Integrate OpenNMT into Production Workflow. In 20th Annual Conference of the European Association for Machine Translation. Czech Republic. 2017.

ҚАЗАҚ ТІЛІНЕ АРНАЛҒАН ПОСТ-РЕДАКТОРЛЕУ

З. Омиотек[1], Д.Р. Рахимова[2], А.Ж. Жунусова[3]

[1]Люблин технологиялық университеті, Люблин, Польша
[2]Әл-Фараби атындағы Қазақ ұлттық университеті, Алматы, Қазақстан
[3]Әл-Фараби атындағы Қазақ ұлттық университеті, Алматы, Қазақстан
[1]z.omiotek@pollub.pl, [2]di.diva@mail.ru, [3]alia_94-22@mail.ru
[1]ORCID ID: https://orcid.org/0000-0002-6614-7799
[2]ORCID ID: https://orcid.org/0000-0003-1427-198X
[3]ORCID ID: https://orcid.org/0000-0002-4350-6322

Аңдатпа. Қазіргі әлем және біздің жақын болашағымыз қолданбалы интеллектуалды жүйелерге байланысты, өйткені күн сайын жаңа технологиялар дамып келеді. Интеллектуалды жүйелердің міндеттерінің бірі – бір табиғи тілден екінші табиғи тілге машиналық (автоматтандырылған) аудару. Машиналық аударма (МТ) адамдарға тілдік айырмашылықтарға

қарамастан сөйлесуге мүмкіндік береді, өйткені ол тілдік кедергіні жояды және қарым-қатынас үшін жаңа тілдерді ашады. Машиналық аударма – бұл жаңа технология, адам дамуындағы ерекше қадам. Аударманың бұл түрі екі түрлі тілде сөйлесетін адамдардың хатта не жазғанын немесе айтқанын тез түсінуге көмектеседі. Интернеттегі онлайн аудармашылардың мақсаты қазақ тіліне және керісінше аудару. Мақалада аударма қателері анықталды, қазақ тіліндегі онлайн машиналық аударма жүйелерінің жалпы артықшылықтары мен кемшіліктері келтірілді. Қазақ тіліне арналған өңдеуден кейінгі машиналық аударма жүйесін әзірлеу үлгісі ұсынылған.

OpenNMT (Open Neural Machine Translation) – нейромашиналық аудармаға және нейрондық тізбекті оқытуға арналған ашық бастапқы жүйе. OpenNMT тілінде тілдерді үйрену үшін тіл жұптары жинақталған параллель корпустар қажет. OpenNMT артықшылығы - оны барлық тілдерге қолдануға болады және үлкен корпустарды өңдей алады. Ағылшын-қазақ тілі жұбы бойынша эксперименттік деректер алынды. Қолданбалы интеллектуалды жүйелер қазіргі заманғы маңызды рөл атқарады. Олардың міндеттерінің бірі - бір тілден екінші тілге машиналық аударма (МТ) қолдану. МТ адамдарға тілдік кедергілерге қарамастан еркін сөйлесуге мүмкіндік береді. Бұл жаңа технология серіктестің сізге не сөйлейтінін немесе жазғанын түсінуге көмектесетін ерекше қадам болып табылады. Автоматты өңдеуден кейінгі өңдеу – машиналық аударма нәтижесінде мәтіндердегі қателерді түзету міндеті. МТ әрқашан қажетті нәтижені бере алмайтындықтан, аударманы өңдеу қажет болады. Аударманың кемшіліктерін кейінгі өңдеу арқылы жою керек. Пост-редакторлеу бұл қажеттілігі негізінен машиналық аударманың сапасымен анықталады: төмен сапалы аударма кейінгі өңдеуге көп орын қалдырады, ал жоғары сапалы және адамдық аудармалар мәтінді ең аз өңдеуді қажет етеді. Бұл жұмыс ағылшын-қазақ тіліндегі жұптарға арналған жеңіл өңдеуден кейінгі модульдің дамуын сипаттайды. Нейрондық желі моделі OpenNMT құрылымы арқылы mt, pe және triplets src, mt, pe жұптарында оқытылады. Содан кейін mt - pe және mt - ape BLEU метрикасының нәтижелері салыстырылады және кейінгі өңдеу сапасы туралы қорытынды жасалады.

Түйін сөздер: Opennmt, нейромашиналық аударма, bleu, автоматты кейінгі өңдеу, желілік модель.

ПОСТРЕДАКТИРОВАНИЕ ДЛЯ КАЗАХСКОГО ЯЗЫКА

O. Zbigniew[1] , Д.Р. Рахимова[2], А.Ж. Жунусова[3]
[1]Люблинский технологический университет, Люблин, Польша
[2]Казахский национальный университет имени аль-Фараби, Алматы, Казахстан
[3]Казахский национальный университет имени аль-Фараби, Алматы, Казахстан
[1]z.omiotek@pollub.pl, [2]di.diva@mail.ru, [3]alia_94-22@mail.ru
[1]ORCID ID: https://orcid.org/0000-0002-6614-7799
[2]ORCID ID: https://orcid.org/0000-0003-1427-198X
[3]ORCID ID: https://orcid.org/0000-0002-4350-6322

Аннотация. Современный мир и наше ближайшее будущее зависят от прикладных интеллектуальных систем, так как новые технологии развиваются каждый день. Одной из задач интеллектуальных систем является машинный (автоматизированный) перевод с одного естественного языка на другой. Машинный перевод (МП) позволяет людям общаться вне зависимости от языковых различий, так как снимает языковой барьер и открывает новые языки для общения. Машинный перевод — это новая технология, особая ступень в развитии человечества. Этот вид перевода может помочь, когда вам нужно быстро понять, что ваш собеседник написал или сказал в письме.

Работа онлайн-переводчиков используется для перевода на казахский язык и наоборот. Выявляются ошибки перевода, приводятся общие преимущества и недостатки онлайн-систем машинного перевода на казахский язык. Представлена модель разработки системы постредактирования машинного перевода для казахского языка.

OpenNMT (Open Neural Machine Translation) — это система с открытым исходным кодом для нейронного машинного перевода и обучения нейронных последовательностей. Для изучения языков в OpenNMT нужны параллельные корпуса для языковых пар. Преимущество OpenNMT в том, что его можно применять ко всем языкам и обрабатывать большие корпуса. Экспериментальные данные были получены для англо-казахской языковой пары. Прикладные интеллектуальные системы играют важную роль в современном мире. Одной из их задач является машинного перевода (МП) перевод с одного языка на другой. МТ позволяет людям свободно

общаться, несмотря на языковые барьеры. Эта новая технология — особый шаг, помогающий понять, что собеседник говорит или пишет вам. Автоматическое постредактирование — это задача исправления ошибок, возникших в текстах в результате машинного перевода. Так как МП не всегда может дать желаемый результат, возникает необходимость редактирования перевода. Недостатки перевода приходится устранять постредактированием. Эта потребность в постредактировании во многом определяется качеством машинного перевода: некачественный перевод оставляет много места для постредактирования, а качественный и человеческий перевод требует минимального редактирования текста. В данной работе описывается разработка легкого модуля постредактирования для англо-казахских языковых пар. Модель нейронной сети обучается на парах mt, pe и триплетах src, mt, pe с использованием фреймворка OpenNMT. Затем сравниваются результаты метрик BLEU mt - pe и mt - ape и делается вывод о качестве постредактирования.

Ключевые слова: Opennmt, нейронный машинный перевод, bleu, автоматическое постредактирование, сетевая модель.

Сведения об авторах

Анг.: Zbignev Omiotec - Professor of Lublin University of Technology, Lublin, Poland

Каз.: Збигнев Омиотек - Люблин технологиялық университетінің профессоры, Люблин, Польша.

Рус.: Збигнев Омиотек – Профессор Люблинского технологического университета, Люблин, Польша.

Анг.: Rakhimova Diana Ramazanovna - Associate Professor of the Department of Information Systems al-Farabi Kazakh National University, Almaty, Kazakhstan.

Каз.: Рахимова Диана Рамазановна- әл-Фараби атындағы Қазақ ұлттық университеті, Ақпараттық жүйелер кафедрасының қауымдастырылған профессоры, Алматы, Қазақстан.

Рус.: Рахимова Диана Рамазановна- ассоциированный профессор кафедры Информационных систем Казахский национальный университет имени аль-Фараби, Алматы, Казахстан

Анг.: Zhunussova Aliya- docoral student of the Department of Information Systems al-Farabi Kazakh National University, Almaty, Kazakhstan.

Рус.: Жунусова Алия- докторант кафедры Информационных систем Казахский национальный университет имени аль-Фараби, Алматы, Казахстан.

Каз.: Жунусова Алия - әл-Фараби атындағы Қазақ ұлттық университеті, Ақпараттық жүйелер кафедрасының докторанты, Алматы, Қазақстан.

UDC 004.93
IRSTI 28.23.37

IMAGE PROCESSING TECHNOLOGY BASED ON CONVOLUTIONAL NEURAL NETWORK

Sh. Bakytnur, P.N. Abilova, N. Tasbolatuly
Astana International University, Nur-sultan, Kazakhstan
shuakbakytnur@gmail.com
ORCID ID:https://orcid.org/0000-0002-0511-7000

Abstract. A convolutional neural network is the structure of a neural network with multiple hidden layers that combines artificial neural network and deep learning. At present, the convolutional neural network, as a deep learning method widely used in various computer fields and effective in processing image information, can learn the features in the image after training, and complete the extraction and classification of image features. Therefore, it is widely used in computer vision research such as image processing and pattern recognition. Although the general structure of the convolutional neural network is very similar to the ordinary artificial neural network, it has a perfect effect on large-scale image datasets. This paper aims to sort out the convolutional neural network and its structural characteristics and advantages, introduce and analyze the basic working principle of the convolutional neural network, and realize the application example of CNN model training in image feature extraction and recognition.

Keywords: Deep learning; Convolutional neural network; image processing; image recognition

1. Introduction

With the rapid development of Internet information technology, processing more and more image data has become one of the popular research directions in the field of computer vision. As one of the ways to process image data, deep learning is also developing rapidly. It has been widely used in speech recognition, natural language processing, computer vision, and other fields, and has achieved great success. This article mainly introduces a neural network that is very effective for processing images-convolutional neural networks. The characteristic of convolutional neural networks is that the characteristics of each layer are obtained by the local area of the previous layer through the convolutional core excitation of shared weights. This feature makes convolutional neural networks more suitable for the learning and expression of image features than other neural network methods, and they have very good results on large-scale image data sets. This article will conduct comprehensive research on convolutional neural networks, which mainly includes the basic structure and working principle of convolutional neural networks, followed by examples of the application of convolutional neural networks.

2. Convolutional neural network and its structure

2.1 Convolutional neural network

Convolutional neural networks originally originated in 1962. Biologists Hubel and Wiesel [1] discovered a cell that covers the entire visual domain and is sensitive to local areas of the visual input space, called the sensory field. In 1980, Fukushima proposed a neurocognitive machine with a similar structure based on the sensory field [2]. Neocognitron is a self-organizing multi-layer neural network model that stimulates the response of the local sensory field of the previous layer to each layer. It is also the main learning method of convolutional neural networks in early learning. After that, in 1998, LeNet-5 proposed by Lecun[3] et al. used a back-propagation algorithm to conduct supervised training of neural network networks. The trained network converts the original image into a series of feature vectors through alternately connected convolutional layers and downsampling layers, and finally classifies the feature expression of the

image through a fully connected neural network. This is the earliest convolutional neural network model. In 2012, A. AlexNet proposed by Krizhevsky [4] and others won the championship in the image classification competition of ImageNet, a large-scale image database, with a huge advantage of accuracy exceeding the second place by 11%, making convolutional neural networks the focus of academia, and it is widely used in optical character recognition, face recognition, image classification, identity recognition, traffic sign recognition, aircraft image recognition and even graph feature analysis.

Convolutional neural networks are a deep learning method specially designed for image classification and recognition developed on the basis of artificial neural networks.

Traditional artificial neural networks consist of three layers: an input layer and an output layer, with multiple hidden layers in the middle. There are several neurons in each layer, and each neuron in the latter layer between the two adjacent layers is connected to each neuron in the previous layer, and there's no association between the neurons within the same layer (*Figure 1*).

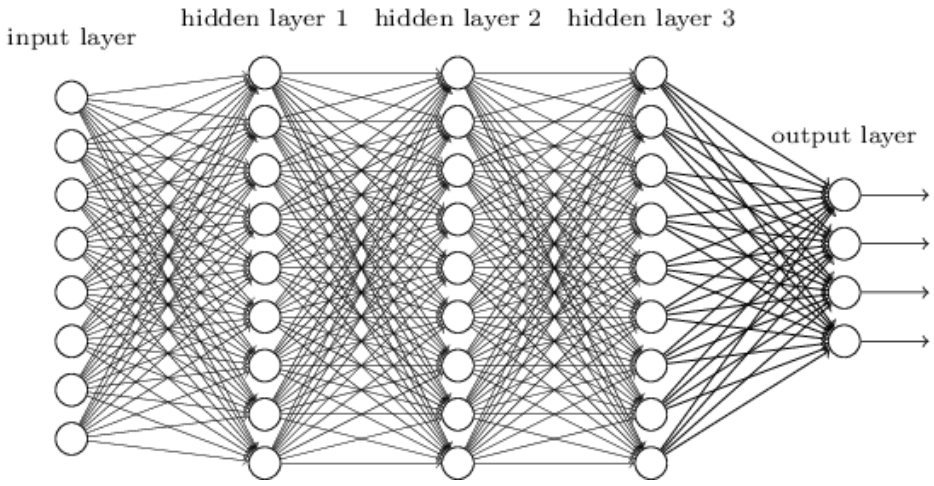

Figure- 1. An example of an artificial neural network model with an input layer, output layer, and 3 hidden layers [5].

When this kind of neural network is trained, each neural network can be used, but because the neurons in the two adjacent layers are all connected, the image processing speed will be limited by the many parameters generated. Therefore, based on artificial neural networks, convolutional neural networks add partially connected convolutional layers and dimension reduction layers in front of the original fully connected layers, and all the upper and lower neurons of CNN are not directly connected, but through the ‘convolution kernel’ as an intermediary. The same convolution kernel is shared among all images, and the image still retains its original positional relationship after the convolution operation. For images, if there is no convolution operation, the number of parameters learned is disaster-level. The reason why CNN is used for image recognition is precise because the CNN model limits the number of parameters and excavates the characteristics of the original structure. In addition to snappily training images through this special convolutional structure, CNN can also reduce the quantum of memory enthralled by the deep network and palliate the overfitting problem of the model.

At the same time, compared with fully connected neural networks similar to CNN, CNN has the characteristics of local connectivity and weight sharing. This kind of neural network not only reduces the training parameters a lot but also conforms to the characteristics that the closer the pixels in the natural image, the greater the impact on the pixels next to it. The weight sharing of the convolutional neural network constitutes the convolutional nucleus, and after it is convoluted with a given image, certain features of the image can be extracted. In the process of image processing, through the training of many different convolution kernel parameters, different image features of the same image can be automatically extracted [6].

2.2 Structure of convolutional neural network

The structure level of convolutional neural networks is more complex than traditional neural networks. The convolutional neural network consists of an input layer, a convolutional layer, a pooling layer, a fully connected layer, and an output layer to form the basic structure. The input and output layers are responsible for receiving and outputting. Convolutional layers and pooling layers generally alternate in pairs. Since the neurons of the characteristic surfaces in the convolutional layer are locally connected to their inputs, the corresponding weights are weighted and summed with the local inputs, and the bias values are added to obtain the input values of the neurons. This process is equivalent to the convolutional process, and the convolutional neural network is also named after it.

Convolutional neural networks belong to feedforward neural networks. Through convolution, pooling, and nonlinear activation perform mapping, high-level semantic information is stripped from the original data layer by layer, which is a feedforward operation. After each convolutional layer, there will be a pooling layer. This is because the information is mapped from low-dimensional to high-dimensional at this time. There are many parameters and the dimension is too high. It is not suitable as the input of the next layer of neurons, so the output of this layer must be processed in a dimensionality reduction, so the pooling operation is introduced. However, it's easy to get overfitting and indeed lead to a dimensional disaster, If you don't reduce the dimension of the data.

3. Principles of convolutional neural network extraction of image features and recognition

Convolutional neural network recognition of an image divides a complete picture into many small parts, extracts the features in each small part, after multiple or parallel automatic feature extraction, and then summarizes each image feature together. The image recognition process can be completed with a high degree of accuracy once the similarity is compared.

3.1. Extraction and recognition of image features

3.1.1Establish a convolutional layer to extract preliminary features.

After the input layer converts an input picture into numerical data that the computer can understand, the convolutional layer will divide the picture data received from the input layer into blocks and extract features from each block. The convolutional nucleus is at the core of the convolutional layer and is responsible for extracting local features from the picture. Each convolution core has a constant bias in the convolution layer, which is a numerical matrix. The elements of the matrix plus the bias contribute to the weight of the convolution layer, and the weight is used for updating the network iteratively. Weight sharing and a local sensory field each convolution operation only needs to consider the colors, contours, textures, and other information present in the part of the area where it is performed; the size of the local sensory field is the scope of the convolution kernel. There is only one convolution layer. In multilayer convolutional networks, it can be fed back layer by layer. As a result of repeated iterations, the size of the sensory field in the original input image, as well as that of the multi-layer convolution layer, can be determined. All previous convolution layers of the layer are related to the convolution kernel size. This is known as weight sharing. Each convolution kernel is unchanged except for the weight update after each iteration. If the value of each convolution kernel is different, the convolution kernel is different as well. The convolution kernels extract features from the image if some convolution kernels extract the color characteristics of the image, contour characteristics, etc. The convolutional layer is made up of multiple feature surfaces. Multiple neurons from each feature surface and each of these neurons are connected to the local area of the feature surface of the previous layer through the convolutional nucleus. The feature map of the previous layer is convoluted by a learnable convolution kernel, and then the output feature map is obtained through an activation function. In the output feature map, the values of multiple feature maps can be combined and convolved.

3.1.2 Create a pooling layer to extract the key features.

After each convolutional layer, there will be a pooling layer. This is because the information is mapped from low-dimensional to high-dimensional at this time. There are many parameters, and the dimension is too high. It is not suitable as the input of the next layer of neurons, so the output of this layer must be processed in a dimensionality reduction, so the pooling operation is introduced. The pooling layer is more effective than convolutional in reducing the dimensions of data. The pooling layer obtains features with spatial immutability by reducing the resolution of the feature surface, which plays the role of secondary feature extraction. Doing so can not only greatly reduce the number of calculations. If the pooling layer does not remove the unimportant feature information in the feature matrix, it can easily cause overfitting and even lead to dimensional disaster. The commonly used downsampling layers are maximum downsampling, random downsampling, mean downsampling, etc. Mean downsampling is to take the average value of the downsampling part to replace all the values in this part; when maximum downsampling is performed, the maximum value is taken in the sampling area; when random downsampling is performed, random values are taken in the sampling area according to certain algorithm criteria.[7] The main roles of downsampling are to reduce the spatial size of the data body, reduce the number of parameters in the network, reduce the overhead of computing resources, and more effectively control overfitting.

3.1.3 Establish a full connection layer to summarize the functions of each part

In the convolutional network structure, at least one fully connected layer is connected after the last pooling layer. Each neuron in the completely connected layer is completely connected to all neurons in the former layer. The fully connected layer can integrate local information with category differentiation in the convolutional layer or the pooling layer. The output value of the last completely connected layer is passed to the output layer. In actual use, the fully connected layer can be realized by a global convolution operation; that is, a convolution kernel of the same size as the image matrix output of the previous layer is used to do convolution operations with the output of the previous layer so that a matrix can be mapped to a number, and a combination of multiple such convolution kernels can map the image matrix output of the previous layer into a fixed-length feature vector. In general, the length of the feature vector corresponds to the number of categories classified. This feature vector is obtained by highly purifying the image features obtained through multiple convolutional layers, pooling layers, and activation functions. This feature vector has high-level feature information, that is, it contains the combined information of all the features of the input image after various operations. This information is the most characteristic feature of the image. Therefore, the image can be classified by outputting the probability value of the specific category to which the image belongs through this information.

From the different and common features of an image, a neural network can be used to identify a specific image among tens of millions. After analyzing the image processing process of convolutional neural networks, the special hierarchical structure of CNN has a very good effect on image feature extraction. In summary, the convolutional layer of CNN is responsible for extracting local features in the image; the pooling layer is used to significantly reduce the order of parameters (dimensionality reduction), and the fully connected layer is similar to the part of a traditional neural network and is used to output the desired results.

3.2 Example of building a simple convolutional neural network model to process images

To realize the training operation of convolutional neural networks, this paper builds an image classification model of convolutional neural networks based on the TensorFlow deep learning framework and studies the steps of data preprocessing, model design, and construction, iterative training, and predictive evaluation. Compared with TensorFlow, Keras uses the least program code and takes the least time to make a deep learning model, train, estimate the delicacy, and make prognostications. Enter the CIFAR-10 data set in the Python environment, normalize the picture, data enhancement, and other preprocessing, and use Keras to construct an improved

VGG16 convolutional neural network structure to model and predict the CIFAR-10 image data set, and finally compare the accuracy and many different models by comparing different batch sizes.

Keras is an open-source advanced deep learning library. Its design refers to Torch, written in Python language, supports GPU and CPU, and is a highly modular neural network library. At present, Keras provides two back-end engines: Theano and TensorFlow. On top of the two, Keras provides APIs that allow users to focus more on model design and conduct model experiments faster. These APIs encapsulate many small components from TensorFlow and Theano in the form of modules, so the network that can be built using these two can also be built through Keras, and there is basically no performance loss. The biggest advantage of using the Keras framework is that it can save more time when building a new network structure.

3.2.1. Development environment

This practice uses the Python + TensorFlow + Keras development environment for programming and model training. Among them, the Python programming language has a clear structure, a rich standard library, and a strong third-party ecosystem, which can efficiently implement complex machine learning algorithms; TensorFlow is a powerful deep learning open-source framework developed by Google, which can easily perform high-performance numerical calculations; Keras belongs to TensorFlow's advanced API, which encapsulates multiple module components for deep learning, which can efficiently and quickly build complex neural network models.

3.2.2 Model design

The CIFAR-10 data set contains 60,000 natural images, which are divided into 10 types. Contains 50,000 training pictures and 10,000 test pictures. The data in the data set exists in an array (stored in rows, each row represents an image). The first 1024 bits are the R-value, the middle 1024 bits are the G value, and the last 1024 bits are the B value. In this article, the experimental data set is simply cut and whitened, and the pixel values are sent to the neural network for training[8]. In this article, a deep convolutional network model is designed based on the CIFAR-10 data set. The architectural parameters of the model are shown in figure 2.

```
def create_model(input_shape):
    # building the model
    model = Sequential()
    model.add(Conv2D(filters=32, kernel_size=(3, 3), padding="same", input_shape=input_shape))
    model.add(Activation("relu"))
    model.add(Conv2D(filters=32, kernel_size=(3, 3), padding="same"))
    model.add(Activation("relu"))
    model.add(MaxPooling2D(pool_size=(2, 2)))
    model.add(Dropout(0.25))
    model.add(Conv2D(filters=64, kernel_size=(3, 3), padding="same"))
    model.add(Activation("relu"))
    model.add(Conv2D(filters=64, kernel_size=(3, 3), padding="same"))
    model.add(Activation("relu"))
    model.add(MaxPooling2D(pool_size=(2, 2)))
    model.add(Dropout(0.25))
    model.add(Conv2D(filters=128, kernel_size=(3, 3), padding="same"))
    model.add(Activation("relu"))
    model.add(Conv2D(filters=128, kernel_size=(3, 3), padding="same"))
    model.add(Activation("relu"))
    model.add(MaxPooling2D(pool_size=(2, 2)))
    model.add(Dropout(0.25))
    # flattening the convolutions
    model.add(Flatten())
    # fully-connected layer
    model.add(Dense(1024))
    model.add(Activation("relu"))
    model.add(Dropout(0.5))
    model.add(Dense(num_classes, activation="softmax"))
    # print the summary of the model architecture
    model.summary()
    # training the model using adam optimizer
    model.compile(loss="sparse_categorical_crossentropy", optimizer="adam", metrics=["accuracy"])
    return model
```

Figure- 2. The CNN model structure in python environment (In order to train the model for accuracy, the Adam optimizer is used)

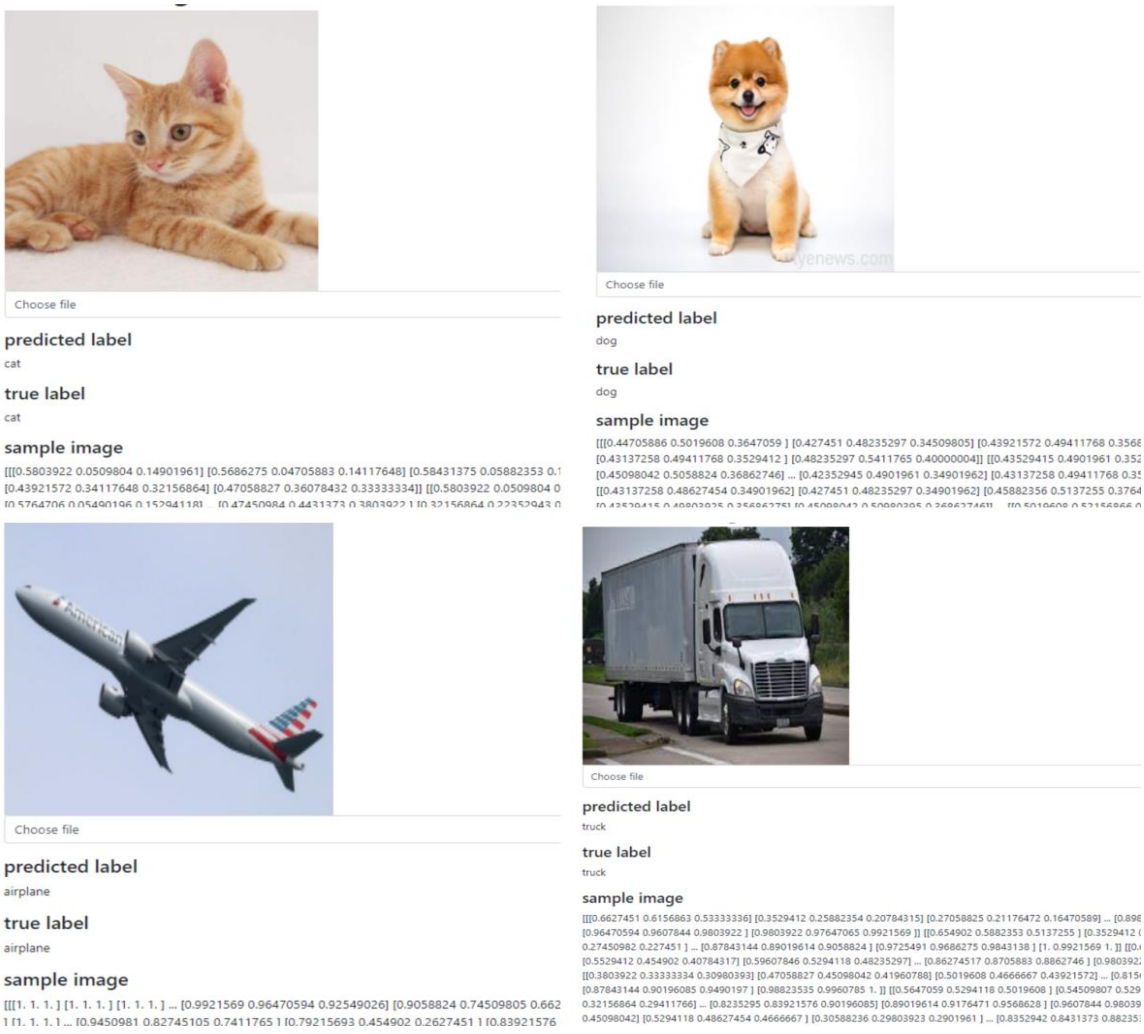

Figure- 3. A simple image recognition Web page based on the Python+TensorFlow+Keras development environment, some of the results obtained are shown

3.2.3 Example analysis

In this paper, the TensorFlow deep learning framework is used to train the CNN model, and it is trained and tested on the CIFAR-10 data set. The training method of cross-verification of the data set is used to iteratively obtain the optimal model (figure 3).Through practice, although the model training method has not reached the high accuracy rate of the CIFAR-10 data set, it provides some ideas for the realization and training of the CNN model.

4. Conclusions

This paper studies the image processing technology grounded on convolutional neural networks, combs the convolutional neural networks and their structural characteristics, builds a simple CNN model, combines the existing image algorithms based on convolutional neural networks, and uses CIFAR-10 as the data set to train its application analysis in image feature extraction and recognition. It can be concluded that compared with traditional neural networks, and convolutional neural networks, the convolutional structure of CNN and its own local connectivity and weight sharing characteristics greatly reduce the complexity of the network model. When entering a multidimensional image, the advantages of this feature are even more obvious. It avoids the process of feature extraction and data reconstruction in traditional recognition algorithms, making the calculation more concise. In general, convolutional neural networks have unique advantages in the field of image processing and computer vision.

References

[1] Hubel D. H, Wiesel T. N. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. The Journal of physiology, 1962, 160(1). 106.

[2] Fukushima K, Miyake S. Neocognitron: A new algorithm for pattern recognition tolerant of deformations and shifts in position[J]. Pattern recognition, 1982. 15(6). 455-469.

[3] Lecun, Y., Bottou, L., Bengio, Y. and Haffner, P. Gradient-Based Learning Applied to Document Recognition. Proceedings of the IEEE, 1998. 86. 2278-2324.

[4] Krizhevsky A, Sutskever I, Hinton G. E. Imagenet classification with deep convolutional neural networks[J]. Advances in neural information processing systems, 2012. 25.

[5] Dürr O, Sick B, Murina E. Probabilistic deep learning: With python, keras and tensorflow probability [M]. Manning Publications, 2020.

[6] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998. *86*(11). 2278-2323. https://doi.org/10.1109/5.726791

[7] Shahid Md. https://towardsdatascience.com/covolutional-neural-network-cb0883dd6529,2021.4.

[8] Krizhevsky A, Hinton G. Learning multiple layers of features from tiny images[J]. 2009.

КОНВУЛЬСИЯЛЫҚ НЕЙРОНДЫҚ ЖЕЛІ НЕГІЗІНДЕ КЕСКІНДЕРДІ ӨҢДЕУ ТЕХНОЛОГИЯСЫ

Бақытнұр Ш., Әбілова П.Н., Тасболатұлы Н.
Астана Халықаралық университеті, Нұрсұлтан, Қазақстан
shuakbakytnur@gmail.com
ORCID ID:https://orcid.org/0000-0002-0511-7000

Аңдатпа. Конволюциялық нейрондық желі жасанды нейрондық желі мен терең оқуды біріктіретін бірнеше жасырын қабаттары бар нейрондық желі құрылымы. Қазіргі уақытта конвульсиялық нейрондық желі әр түрлі компьютерлік салаларда кеңінен қолданылатын және кескін туралы ақпаратты өңдеуде тиімді терең оқыту әдісі ретінде жаттығудан кейін кескіннің ерекшеліктерін зерттеп, кескін белгілерін алу мен жіктеуді аяқтай алады. Сондықтан ол кескіндерді өңдеу және үлгіні тану сияқты компьютерлік көру зерттеулерінде кеңінен қолданылады. Конвульсиялық нейрондық желінің жалпы құрылымы әдеттегі жасанды нейрондық желіге өте ұқсас болғанымен, ол үлкен масштабтағы кескін мәліметтерімен жақсы жұмыс істейді. Бұл мақаланың мақсаты-конвульсиялық нейрондық желіні, оның құрылымдық сипаттамалары мен артықшылықтарын түсіну, конвульсиялық нейрондық желінің негізгі принципін ұсыну және талдау, сонымен қатар кескіннің белгілерін алу және тану үшін CNN моделін оқытудың мысалын қолдану.

Түйін сөздер: терең оқыту; конвульсиялық нейрондық желі; суретті өңдеу; суретті тану.

ТЕХНОЛОГИЯ ОБРАБОТКИ ИЗОБРАЖЕНИЙ НА ОСНОВЕ СВЕРТОЧНОЙ НЕЙРОННОЙ СЕТИ

Бакытнур Ш., Абилова П.Н., Тасболатулы Н.
Международный университет Астана, Нурсултан, Казахстан
shuakbakytnur@gmail.com
ORCID ID:https://orcid.org/0000-0002-0511-7000

Аннотация. Сверточная нейронная сеть структура нейронной сети с несколькими скрытыми слоями, которая сочетает в себе искусственную нейронную сеть и глубокое обучение. В настоящее время сверточная нейронная сеть, как метод глубокого обучения, широко используемый в различных компьютерных областях и эффективный при обработке информации об изображении, может изучать особенности изображения после обучения и завершать извлечение и классификацию признаков изображения. Поэтому он широко используется в исследованиях компьютерного зрения, таких как обработка изображений и распознавание образов. Хотя общая

структура сверточной нейронной сети очень похожа на обычную искусственную нейронную сеть, она прекрасно работает с крупномасштабными наборами данных изображений. Цель этой статьи - разобраться в сверточной нейронной сети, ее структурных характеристиках и преимуществах, представить и проанализировать основной принцип работы сверточной нейронной сети, а также реализовать пример применения обучения модели CNN для извлечения и распознавания признаков изображения.

Ключевые слова: Глубокое обучение; Сверточная нейронная сеть; обработка изображений; распознавание изображений.

Сведения об авторах

Анг.: Bakytnur Shuak - Student's Astana international university, Nur-Sultan, Kazakhstan.

Каз.: Бақытнұр Шуақ- Астана Халықаралық университеті студенті, Нұр-Сұлтан, Қазақстан.

Рус.: Бакытнур Шуак, студент Международного университета Астана, Нур-Султан, Казахстан.

Анг.: Abilova Perizat Nurkhatovna - Senior lecturer, master's degree Astana international university, Nur-Sultan, Kazakhstan.

Каз.: Абилова Перизат Нурхатқызы - Астана Халықаралық университеті аға оқытушысы, Магистр, Нұр-Сұлтан, Қазақстан.

Рус.: Абилова Перизат Нурхатовна- магистр, старший преподаватель Международного университета Астана, Нур-Султан, Казахстан.

Анг.: Abilova Perizat Nurkhatovna - Senior lecturer, PhD, Astana international university, Nur-Sultan, Kazakhstan.

Каз.: Тасболатұлы Нұрболат - Астана Халықаралық университеті аға оқытушысы, PhD, Нұр-Сұлтан, Қазақстан.

Рус.: Тасболатулы Нурболат - PhD, старший преподаватель Международного университета Астана, Нур-Султан, Казахстан.

Responsible for the release: PhD, Shayakhmetova A.S.

Merkebaev A.

Deputy chief editor: PhD, Mamyrbayev O.Zh

Signed in print 03.03.2020
Edition of 50 copies. Format 60x84 1/16. Paper type.
Order No. 4.

Publication of the Institute of Information and Computational Technologies

28 Shevchenko str., Almaty, Republic of Kazakhstan
7 (727) 272-37-11
atcs@iict.kz