

Scancode Keyboard as Knowledge Encoding on Adaptive Resonance Theory (ART)

Wari Nur Raharjo¹, Dr. rer.nat. I Made Wiryana², Dr. Detty Purnamasari^{3*}

¹ nurwary@outlook.com

¹Magister of Information System Management Program, Gunadarma University, Depok 16424, Indonesia

²Doctoral Program of Information Technology, Gunadarma University, Depok 16424, Indonesia

³Doctoral Program of Information Technology, Gunadarma University, Depok 16424, Indonesia

Abstract

This study aims to compare the results obtained, the efficiency and effectiveness of the implementation of the scancode method in the encoding process applied to Adaptive Resonance Theory (ART) which previously used the ASCII method as the default knowledge encoding. The implementation of ART is also carried out to handle Typographical Errors (typo) which is based on the most common method of distinguishing the spelling of two words in spelling, namely Levenshtein distance, which is comparing point by point similarities between two words and also Hamming. The data used as the research object was obtained randomly from several respondents who were documented in real-time so that the respondents could find out what data we would use. This study gives the results that the use of keyboard scancode as knowledge encoding in ART implementation can work as well as using ASCII as well as the clustering results obtained.

Keywords: encoding; scancode; adaptive resonance theory

1. INTRODUCTION

The computer keyboard is designed in such a way with a scancode feature, the ability to perform indexing according to the key position in the its internal matrix. Modestly, scancode produces a number of data that is sent by a keyboard to the computer to report which key has been pressed. The number and/or sequence of numbers is determined for each key on the keyboard. Scancode works by moving one bit in the byte sent to differentiate between a key press (make) or a key release (break) event. The raw scancode bytes are sent to the program by the BIOS. According to IBM, scancode is a tool that makes it easier for computers to translate input given by humans via the keyboard. For example, when you press the 'a' key on the keyboard it will appear on the monitor screen. But behind it all there is a sequence of processes that run, starting from input (keyboard), process (CPU), output (monitor). The 'a' button that is pressed cannot immediately be read and processed by the CPU, but the CPU reads and processes bits of data which then produces output 'a' on the monitor screen. What produces a set of bits is the keyboard scancode which is an encoder or translator. Scancode on the keyboard has an important role in the progress of the process from input provided by the user to output that appears on the monitor screen or to carry out a command or carry out certain program actions. The working process on the

keyboard is of course very dependent on the continued operation of the scancode in it. Scancode consists of 168 hex keys which can be read and understood by a computer. This key will provide orders or commands that are input by sending a linear arrangement of lines that form a unique 8-bit scan code when a keyboard key is pressed. Adaptive Resonance Theory (ART) is one solution in solving the problem of how a learning system is designed to remain adaptive in responding to certain commands while remaining stable in responding to other commands that have no relevance to the previous design or pattern. So this cognitive and neural theory emerged, about how the brain learns independently to categorize, recognize and predict objects and events in a dynamic world.

2. LITERATURE REVIEW

ART is always open to new learning or what is called learning plasticity (with adaptive properties) without losing old patterns or learning stability (resonance). Adaptive resonance theory was developed by Stephen Grossberg and Gail Carpenter in 1987 which is basically a vector classifier that takes an input vector and classifies it into one of the categories depending on its storage pattern. There are several stages in the operation of Adaptive Resonance Theory, as follows.

- (1) Recognition stage, recognizes the input vector with the classification given to each node in the output layer. The output from the neuron will have a value of '1' if there is similarity with the classification applied, and will have a value of '0' if there is no similarity with the classification.
- (2) Comparison stage, comparing the input vector with the comparison screen vector. The condition for reset is that the level or degree of similarity is less than the alert parameter.
- (3) In the search stage, the network will search again for matches made in the previous stage. Therefore, if no re-search is carried out and the comparison is good enough, then the classification is complete. Otherwise, the process will be repeated continuously and another stored pattern will have to be sent to find the correct match in the classification.

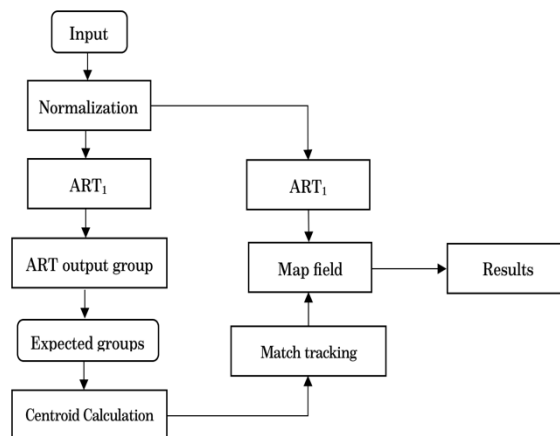


Fig 1: ART implementation flow

The implementation of ART is very influential in the application of Neural Networks because ART has the ability to describe the numbers that need to be processed so that they can be processed further. Thus, Neural

Networks can produce performance not only in pattern recognition or classification tasks, but can also maintain the elasticity of biological networks in learning or in recognizing new patterns in learning without having to delete (forget) or substantially delete the learned patterns previously. In addition, by applying ART theory, Neural Networks can be trained to independently organize themselves without requiring supervision.

- (1) Neural Network is a neural network processing with large numbers that has relationships between its elements. The discovery of neural networks was inspired by biological learning, but not all neural network models were developed by duplicating the operations of the human brain. Some principles in AI related to NNs and their models cannot even be explained from a biological point of view. NN, which is also an adaptive system, can change its structure to solve problems based on external and internal information flowing through the network. Because of its adaptive nature, Neural Networks are also often called adaptive networks. NN has a construction that involves several things, among others. Determine network properties, namely network topology or connectivity, network type, network order and weight coverage.
- (2) Determine the node properties, namely the activation scope and transfer of the activation function.
- (3) Determine the dynamics of movement or operational development of the system, namely the formation of a weight initialization scheme, activation calculation formula and learning rules in the NN.

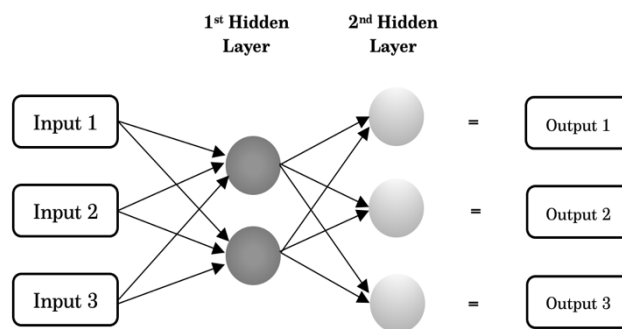


Fig 2: Neural Network with two hidden layers

It can be concluded that NN works by coordinating the relationship between the network and nodes as elements that play an important role in the process of implementing Machine Learning, especially Deep Learning using Neural Networks. One of the most frequently used NN learning models is Adaptive Resonance Theory (ART). An algorithmic method that works by recognizing and identifying objects in general through classification or grouping of data with the ability to remember previously carried out exercises without deleting previously carried out exercises.

The default implementation of ART uses the ASCII (American Standard Code for Information Interchange) method, which is a basic standard code for exchanging information on computers. Initially, ASCII was only composed of 7 bits of code that described each character using binary values which created decimal numbers that every computer understood. ASCII codes are divided into three types, the first is a code that comes from a system with a decimal value between 0 and 31. Next, there are also lowercase characters produced by ASCII with a decimal range between 32 to 127. And the last is the character uppercase letters from ASCII translation results with a range of decimal numbers between 128 to 255. So that this number can fulfill the conversion of binary number values to decimal which is more commonly used and understood by computers.

3. METHODOLOGY

This type of research uses quantitative methods because it has several processes that involve several mathematical algorithms such as the normalization of each hex scancode and ASCII into vector values which are then used to write programs in the Python language.

On the other hand, correcting words that frequently experience typing errors or typos requires data in the form of responses obtained directly from users who operate virtual or physical qwerty keyboards on a daily basis. However, the level of errors that occur for each user can vary based on a person's ability and level of proficiency in using the qwerty keyboard in question. Beginner users may experience more typos because they are not used to using it effectively and efficiently. Meanwhile, users who have mastered the use of the Qwerty keyboard will understand how to use it more efficiently and effectively so that they can prevent more typos from occurring.

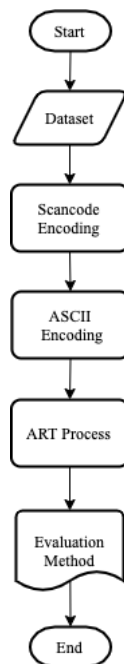


Fig 3: Research stages flow

However, a typo correction development process will be necessary to make it easier to use Qwerty keyboards in the future. Any user of any level of proficiency can use it in an equally effective and efficient manner. And to achieve these development goals, five stages of the development process in question have been determined, the stages in question are as follows.

- (1) Collect data from respondents as users directly via Google Spreadsheet documents which are distributed randomly
- (2) The coding process uses a Scancode keyboard which is then normalized.
- (3) An encoding process with ASCII as the default is applied to the ART, which is then normalized.
- (4) Implementation of Adaptive Resonance Theory Clustering.
- (5) Comparative evaluation of ART implementation using Scancode and ASCII coding methods.

4. RESULTS

This research was carried out with the basic steps of changing text data into numeric data using the keyboard scancode method which was then implemented with the ART algorithm. The ART algorithm itself has a built-in method, namely using the ASCII method. However, in this research a new alternative was created with the aim of facilitating the process of implementing ART in the future.

4.1. Data Collective from Direct Users

Through this process, data has been collected randomly from several respondents. Some samples of this data are as follows.

Table 1: Data Sample from Direct Users

Origin Word	V1	V2	V3	V4	V5
surat	suray	durat	sursy	sirat	surya
pintar	pintaf	pintae	pjntar	pmtar	pnistar
data	daya	dara	dsya	dars	dayta

Table 1 shows that typing errors commonly experienced by keyboard users lie in the position of adjacent keyboard keys. In this case, the keyboard used is of course a qwerty type keyboard which has now been implemented on all virtual and conventional systems. Some users who are still considered 'technically clueless' may have great difficulty operating a Qwerty keyboard which has an arrangement of keys with letters that are not in sequence. Among other things, it can cause a high level of typos, resulting in the formation of letters that are actually the word 'smart', becoming the word 'pnitsr' whose meaning is not recognized. Some of them only cause typos of one or two letters which can still be understood, for example the word 'letter' becomes 'suray' and the word 'data' becomes 'dayta' which still sound almost the same.

4.2. Encoding process using Scancode Keyboard

This process is not much different from the previous one, the next action is converting text characters in the form of letters or symbols based on the scancode code that is owned by normalizing it to get a vector value in the range 0 -1 which begins with adjusting the algorithm. Scancode has a hex value of 195 of the total characters.

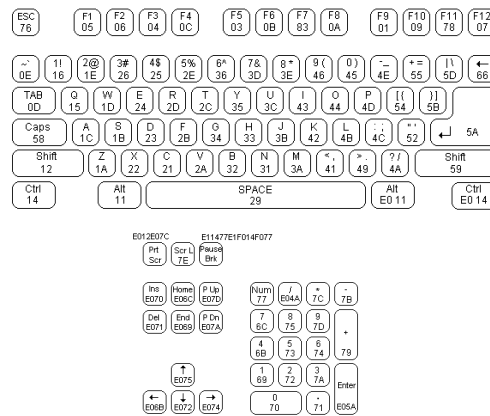


Fig 4: Hex value on Scancode Keyboard (Craig Peacock)

For example, returning to the word 'letter' which is then encoded using scancode, it can be seen that each decimal vector value is '31, 22, 19, 30, 17' so to get the normalized vector by calculating '31/195, 22 /195, 19/195, 30/195, 17/195' so that the final normalized value is '0.158, 0.112, 0.097, 0.153, 0.087' after following the scancode algorithm.

4.3. Encoding process using ASCII Keyboard

Meanwhile ASCII has a total of 128 hex. An example can be taken for the word 'letter' which has the ASCII hex '73, 75, 72, 61, 74' which is then converted to decimal, namely '115, 117, 114, 97, 116' then reprocessed according to the algorithm using the division operation '115/128, 117/128, 114/128, 97/128, 116/128' which has the final result of ASCII normalization being '0.898, 0.914, 0.890, 0.757, 0.906' for the word 'letter ' as one of the results of this normalization.

4.4. Comparison of ASCII Normalization with Scancode

Table 2: Comparison of ASCII Normalization with Scancode

		Origin Word	V1	V2	V3	V4	V5
		surat	suray	durat	sursy	sirat	surya
Normalization	Scancode	0.158,	0.158,	0.194,	0.158,	0.158,	0.158,
		0.112,	0.112,	0.112,	0.112,	0.117,	0.112,
		0.097,	0.097,	0.097,	0.097,	0.097,	0.097,
		0.153,	0.153,	0.153,	0.158,	0.153,	0.107,
		0.087	0.107	0.087	0.107	0.087	0.153
	ASCII	0.898,	0.898,	0.781,	0.898,	0.898,	0.898,
		0.914,	0.914,	0.914,	0.914,	0.820,	0.914,
		0.890,	0.890,	0.890,	0.890,	0.890,	0.890,
		0.757,	0.757,	0.757,	0.898,	0.757,	0.945,
		0.906	0.945	0.906	0.945	0.906	0.757

From the values obtained in Table 3, it can be concluded that the comparison of the use of ASCII and Scancode in the encoding process has final results with different value ranges for each character. On ASCII encoding, the

numbers obtained are quite high in the range of 0.8, 0.7, to 0.9, which should not have a significant influence on the use of encoding related to general needs. However, if studied further, encoding with scancode can of course be more profitable to apply to typo correction development which is being discussed in this research. The vector values produced by the scancode in the normalization process only have the number range 0.0 and 0.1 after going through a division operation process based on the number of hex codes the scancode has which are then converted to decimal.

4.5. ASCII Clustering

Vector values that have gone through a normalization process are then designed in the form of clusters by grouping them based on the data segmentation of errors that occur in each letter that causes typos. The number of clusters for each word that can be used as a benchmark for the next stage of ART is as follows.

(1) To check the suitability of the choice equality criteria, it is necessary to find the best choice

- Parameters:
 - norm** – input and minimal template
 - norml** – standar input
- Returns:
 - return to category locatioin

(2) Re-check the criteria

- Parameters:
 - cmax** – maximum choice (initialization to -1)
 - chmax** – criteria checking (initialization to -1)
 - ch** – template of choice
 - nc** – number of categories
- Returns
 - location template of maximum choice

(3) ART train by making Template Matrix.

- Parameters:
 - I** – input
 - T** – template
 - cmax** – maximum choice (initialization to -1)
 - chmax** – criteria checking (initialization to -1)

(4) Train using main algorithm on ART laterally

- Parameters:
 - xA** – input matrix A-side (float)
 - xB** – input matrix B-side (float)
 - rhoA** – free parameter A-side (float)
 - rhoB** – free parameter B-side (float)
 - beta** – free parameter from study level (float)
 - alpha** – parameter of choice (float)

nep – sum of epoch (integer)
memory_folder – memory storage folder (string)
update_template – command to update or create new template (boolean)

- Return
 Return TA: A-Side template matrix (float)
 Return TB: B-Side template matrix (float)
 Return L: Associator matrix (float)
 Return elapsed_time: time to finish the training (float)

4.6. Knowledge Encoding

The next stage is the encoding stage, which is the process of changing the input data in text form to numeric value (text vector) using scancode encoding. Each character of the text is transformed into a text vector in the form of a number using scancode.

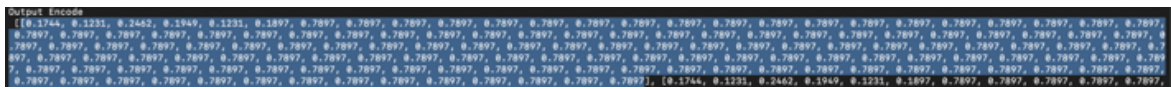


Fig 5: Scancode vector of the word of 'pintar'

4.7. Implement ART

After carrying out hex normalization, clustering and then implementing knowledge encoding, the data is ready to be implemented into ART with the following program writing steps.

- (1) Library import and set the threshold value or vigilance level.

```
import pandas
import csv
import math, numpy

data=[]
memory=[]
panjang=1
nilai_centroid=[]
vigilance_level=60
learning_rate=0.005
t=0
perbandingan=[]
```

Fig 6: Library import and set the threshold value and vigilance level.

- (2) Get the dataset.

```
datanya = pandas.read_csv('dataset_v2.csv')
datanya = datanya.drop(['Kata_Sesungguhnya', 'Versi_1', 'Versi_2', 'Versi_3',
'Versi_4', 'Versi_5', 'Versi_6', 'Versi_7'], axis=1)
data = numpy.array(datanya)
```

Fig 7: Get data

- (3) Calculating the hamming distance (the function to calculate the hamming_distance every memory/ centroid).

```
def hamming_distance(data,centroid):
    print('execute hamming_distance : ' , centroid)
    jumlah=0
    # for i in range(len(data)):
    for i in range(140):
        hasil=(float(data[i])-float(centroid[i]))
        delta=math.pow(hasil,2)
        jumlah+=delta

    hamming=math.sqrt(jumlah)
    return hamming
```

Fig 8: Calculating Hamming Distance

- (4) Update centroid value (the function to update threshold value).

```
def update_centroid(memory,data,learning_rate):
    print('execute update_centroid : ' , memory)
    jumlah=0
    hasil=[]
    #data=[]
    # for i in range(len(data)):
    for i in range(140):
        # delta=float(memory[i])-float(data[i])
        delta=float(data[i])-float(memory[i])
        #print (str(i)+"==delta"+ str(delta))
        final_hasil=float(memory[i]) + (delta*learning_rate)
        hasil.append(final_hasil)

    return hasil
```

Fig 9: Update centroid value

- (5) Input first pattern value as centroid because there's no available centroid yet.

```
memory_awal=numpy.zeros(148)
memory.append(data[1])
jarak=hamming_distance(data[1],memory_awal)
```

Fig 10: Value input

- (6) Run the ART.

```
print('panjang data : ' , len(data))
for y in range (1,len(data)):
    kandidat=[]
    for m in range (len(memory)):
        jarak=hamming_distance(data[y],memory[m])
        if jarak < vigilance_level:
            kandidat.append(m)

    status=0
    index_pilihan=0
    nilai_pilihan=0
    for z in kandidat:
        status=1
        if nilai_pilihan==0:
            jarak = hamming_distance(data[y], memory[z])
            nilai_pilihan=jarak
            index_pilihan=z
        else:
            jarak = hamming_distance(data[y], memory[z])
            if(jarak < nilai_pilihan ):
                nilai_pilihan=jarak
                index_pilihan=z

    if index_pilihan==0 and status==0:
        memory_awal=numpy.zeros(148)
        jarak=hamming_distance(data[y],memory_awal)
        memory.append(data[y])
    else:
        hasil=update_centroid(memory[index_pilihan],data[y],learning_rate)
        memory[index_pilihan]=hasil
```

Fig 11: Run the ART

Based on the results of testing this code, there were 140 clusters, this indicates that ART can run well using vector input using keyboard scancode.

5. CONCLUSION

In this research, an Adaptive Resonance Theory (ART) method was developed which uses vector input resulting from a keyboard scancode encode. Then it can be concluded that the ART implementation using scancode encoding which is first converted into an input vector can produce clusters, as is the case with ART which uses input vectors with ASCII encoding. This indicates that ART with scancode encoding can be implemented to replace the default ART with an ASCII encoded input vector. It is hoped that this method can help as an alternative process for clustering datasets in the form of words or sentences.

References

- Bing Chen Pu Li, dan Tahir Husain. Development of an Integrated Adaptive Resonance Theory Mapping Classification System for Supporting Watershed Hydrological Modeling accessed on Researchgate.net, 2011.
- Fu, Limin. (1994). *Neural Networks in Computer Intelligence*. New York: McGraw Hill.
- Gurney, Kevin. (1997). *An Introduction to Neural Network*. London & New York: Berne Convention.
- i2 Tutorials, "Introduction to Neural Networks" online access on www.i2tutorials.com/introduction-to-neural-networks, accessed 3rd of September 2020.
- Bengio, Y., Courville, A., and Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.*
- Tripathi, Alpika & Srivastava, Geetika & Singh, K.K. & Maurya, Pradeep. (2019). Review of Unsupervised Adaptive Resonance Theory.
- Carpenter, Gail. (2017). *Adaptive Resonance Theory*.
- Perez, A. (2017). *Artificial neural networks: Adaptive Resonance Theory*.
- Barton, Adam & Volna, Eva & Kotyrba, Martin. (2017). *Big Data Filtering Through Adaptive Resonance Theory*. Ostrava: Asian Conference on Intelligent Information and Database Systems.
- Phil Storrs PC Hardware book, "The PC Keyboard Scan Codes" accessed on www.philipstorr.id.au/pcbook/book3/scancode.htm, accessed on 22nd of Februari 2019.
- Raschka, Sebastian. (2015). *Python Machine Learning*. Birmingham, United Kingdom: Packt Publishing.
- Müller, A.C., Guido, S. (2016). *Introduction to Machine Learning with Python: A Guide for Data Scientists*. Massachusetts, United States: O'Reilly Media, Inc.
- ASCII Table, ASCII Code. "The complete table of ASCII characters, letters, codes, symbols and signs." accessed on www.theasciicode.com.ar, accessed on 22nd Februari 2019.
- Sipayung, M.A. (2015). Implementasi dan Perbandingan Pengenalan Pola Tanda Tangan Menggunakan Metode Kohonen dan Metode Adaptive Resonance Theory (ART). Medan: Universitas Sumatera Utara.
- Nurdin. Pratama, K. (2019). Klasifikasi Kecantikan Wanita Aceh Pada Citra Menggunakan Metode Adaptive Resonance Theory (ART1). Lhokseumauwe: TECHSI: Jurnal Penelitian Teknik Informatika.
- Grossberg, S. (2013). *Adaptive Resonance Theory: How a brain learns to consciously attend, learn, and recognize a changing world*. Amsterdam: Elsevier.