

Tribhuvan University
Institute of Engineering
Pulchowk Campus

MULTIMEDIA SYSTEM

LZW Compression Algorithm

SUBMITTED BY:

Bishal Katuwal
075BCT028

SUBMITTED TO:

Department of Electronics and Computer Engineering
Pulchowk Campus

SUBMITTED ON:

9th March, 2023

Title

LZW Algorithm

Background Theory

LZW compression is a lossless data compression method. It was invented by Abraham Lempel, Jacob Ziv and Terry Welch in 1984. The name LZW stands for their names, Lempel(L), Ziv(Z) and Welch(W). It is widely used in modern computer systems, mostly in compressing text files.

The basic idea behind LZW compression is to replace repeated sequences of data with a code that represents that sequence. The working of LZW compression can be summarized in following two steps:

Algorithm

For compression:

1. Initialize a dictionary with all possible input characters as keys.
2. Read in the input data stream, one character at a time.
3. Initialize an empty string and empty output.
4. For each character in stream:
 - (a) Append the character in current string.
 - (b) Check if the current string is in dictionary.
 - If the current is in the dictionary, add next characters from the stream to the dictionary until the string is no longer in the dictionary.
 - If not, add the string to dictionary and output and reset the string to last character of previous string.
5. Output the code for the final string.

For decompression:

1. Initialize a dictionary with all possible input characters' ASCII as keys.
2. Read in the code stream, one code at a time.
3. Initialize a output string with first code's dictionary value.
4. For each code in stream :
 - If the code is in the dictionary, look up the corresponding character and append it to the output string.
 - If not, it represents a combination of the previous code and the first character of the current string. Look up the previous code and the first character of the current string in the dictionary to obtain the corresponding character sequence. Append this sequence to the output string and add it to the dictionary as a new code.

5. Output the final string.

The main advantage of the LZW algorithm is that it adapts to the data being compressed. This means that the size of the dictionary can grow as new codes are added, allowing for better compression of data with repetitive patterns.

Python Code

```
# LZW compression
def compress(data):
    dictionary = {chr(i) : i for i in range(256)}
    pattern = ""
    output = []
    for char in data:
        if pattern + char in dictionary:
            pattern = pattern + char
        else:
            output.append(dictionary[pattern])
            dictionary[pattern + char] = len(dictionary)
            pattern = char
    output.append(dictionary[pattern])
    return output

def decompress(data):
    dictionary = {i : chr(i) for i in range(256)}
    current = data[0]
    output = dictionary[current]
    for i,code in enumerate(data[1:]):
        if code in dictionary:
            temp = dictionary[code]
        else:
            temp = dictionary[current] + dictionary[current][0]
        output += temp
        # print(output)
        dictionary[len(dictionary)] = dictionary[current] + temp[0]
        current = code
    return output

def LZW(input) :
    print('The input string is ' + input + ' with length ' + str(len(input)))
    encoded = compress(input)
    print('The encoded stream is ' + str(encoded) + ' with length ' + str(len(encoded)))
    decoded = decompress(encoded)
    print('The decoded stream is ' + str(decoded))

def main():
    LZW('This string has no repeated pattern.')
    LZW('sadsadsadsad')

if __name__ == '__main__':
    main()
```

Output For 'This string has no repeated pattern.'

The input string is This string has no repeated pattern. with length 36

The encoded stream is [84, 104, 105, 115, 32, 115, 116, 114, 105, 110, 103, 32, 104, 111, 32, 114, 101, 112, 101, 97, 116, 101, 100, 32, 112, 278, 279, 114, 110, 46] with length 33

The decoded stream is This string has no repeated pattern.

For 'sadsadsadsad'

The input string is sadsadsadsad with length 12

The encoded stream is [115, 97, 100, 256, 258, 257, 259] with length 7

The decoded stream is sadsadsadsad

Conclusion

In conclusion, LZW compression algorithm is a powerful lossless compression method. It works by building a dictionary of patterns and encoding the input data as a sequence of codes that reference the patterns in the dictionary. As a dynamic encoding technique, LZW also has no upper limit to number or size of codes, thus for a large system it may consume large memory. Also, the algorithm is subject to some patent restrictions. Overall, the LZW algorithm is a widely used and valuable tool for lossless data compression.