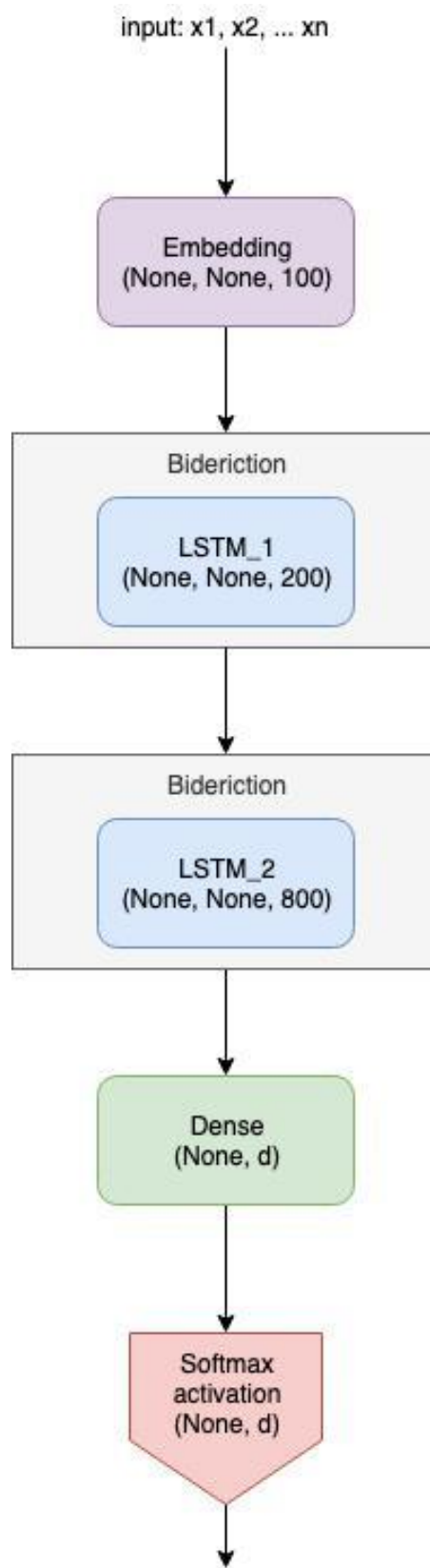


23. Моралдо Х. An Approach for Text SteganographyBased on Markov Chains [Електронний ресурс] / Hernan Moraldo. – 2014. – Режим доступу до ресурсу: <https://arxiv.org/pdf/1409.0915.pdf>.

Додаток А

Схема архітектури нейронної мережі



Додаток Б

Код програми

```
import markovify
import numpy as np
from numpy import cumsum
from markovify.chain import BEGIN,END
from fractions import Fraction
from collections import defaultdict

def find_fraction(input_start, input_end):
    output_fraction = Fraction(0, 1)
    output_denominator = 1

    input_start_fraction = Fraction(input_start)
    input_numerator = input_start_fraction.numerator
    input_denominator = input_start_fraction.denominator

    while not (input_start <= float(output_fraction) < input_end):

        output_numerator = 1 + ((input_numerator * output_denominator) //
input_denominator)
        output_fraction = Fraction(output_numerator, output_denominator)
        output_denominator *= 2

    return output_fraction
def get_next_character(f):
```

```
c = f.read(1)
while c:
yield c
c = f.read(1)
```

```
def calculate_cum_freq(freqTab):
fqa =cumsum(freqTab.values())
length = len(freqTab) + 1
cumFreq = [0] * length
for I in range(1, length):
cumFreq[i] = float(fqa[I - 1]) / float(fqa[-1])
cFreq = (freqTab.keys(), cumFreq)
return cFreq
```

```
def arithmetic (word):
interval_inf = 0
interval_sup = 1
length = len(word)
freqTab = defaultdict(int)
```

```
fixed = True
```

```
for I in range(ord('a'), ord('z')):
freqTab[chr(i)] = 1;
```

```

for i in range(ord('0'), ord('9')):
    freqTab[chr(i)] = 1

if (fixed):
    with open("corpus.txt") as f:
        for c in get_next_character(f):
            string = str(c)
            if(string.isupper()):
                string = string.lower()
            freqTab[string] += 1
            elif (c in freqTab):
                freqTab[c] += 1
            for k, f in freqTab.items():
                if f < 500:
                    del freqTab[k]
            keys,cumFreq = calculate_cum_freq(freqTab)

for i in range(length):
    lengthInterval = interval_sup - interval_inf
    char = entrada[i]
    j = keys.index(char)

interval_inf = cumFreq[j] * lengthInterval
interval_sup = cumFreq[j+ 1] * lengthInterval

```

```

else:
for i in range(length):
lengthInterval = interval_sup - interval_inf
keys, cumFreq = calculate_cum_freq(freqTab)
char = entrada[i]
j = keys.index(char)

interval_inf = cumFreq[j] * lengthInterval
interval_sup = cumFreq[j + 1] * lengthInterval
return (interval_inf + interval_sup)/2

def carrega():
with open('corpus.txt', encoding='utf8') as f:
text = f.read()
return markovify.Text(text,state_size = 4).chain
chain=carrega()

def move(state, w):
return state[1:] + (w,)

def cumfreqs(chain, state):
choices, weights = zip(*chain.model[state].items())
sw = np.cumsum(np.array(weights, dtype=float))
sw /= sw[-1]
return (choices, np.concatenate(([0], sw)))

secret_message = "castanya22"
init_state = 4*(BEGIN,) #El numero depen de state_size
n=arithmetic(secret_message)
word='BEGIN'
while word != END:

```

```
chs, fq = cumfreqs(chain, init_state)
for i in range(len(fq)-1):
    if fq[i] <= n < fq[i+1]: break
    #fq[i], fq[i+1], chs[i]
    s
    init_state = move(init_state, chs[i])
    n = (n - fq[i])/float(fq[i+1]-fq[i])
    word = chs[i]
    print(word)
```

