

УДК 004.85

DOI <https://doi.org/10.32782/2663-5941/2022.5/57>**Борисов Г.О.**Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»**Трапезон К.О.**Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»

## ДОСЛІДЖЕННЯ ОСОБЛИВОСТЕЙ СТВОРЕННЯ ЕЛЕКТРОННИХ СИСТЕМ РОЗПІЗНАВАННЯ МОВИ НА ОСНОВІ НЕЙРОННИХ МЕРЕЖ

Визначено програмні способи реалізації рекурентних нейронних мереж, як основи для створення системи розпізнавання мовних сигналів за візуальними ознаками. Проаналізовано основні ланки моделі нейронної мережі прямого розповсюдження і відмічено, що за архітектурою і принципами функціонування останньої найважливішими елементами в таких мережах є комірки пам'яті та комутаційні пристрої – вентиля двох чи трьох типів за призначенням. Так вентиля дозволяють зберігати інформацію у блоці моделі мережі упродовж значного числа часових кроків, на які за тривалістю розбивається вхідна послідовність. Такий підхід дозволяє нейронній мережі працювати з мовними сигналами, які мають ознаки варіативності за часом. Показано, що для коректної роботи мережі доцільно використовувати петлі зворотного зв'язку для прихованих шарів, оскільки наявність зворотних зв'язків визначає у цьому випадку здатність мережі до навчання.

В якості вхідних даних до створеного алгоритму функціонування рекурентної нейронної мережі обрано три мовних сигнали, які мережа повинна відтворити при мінімальному значенні функції втрат. Відмічено, що в основі роботи програмного алгоритму мережі підключено відповідні бібліотеки-модулі в середовищі об'єктно-орієнтованого програмування *python*, і зокрема використано фреймворк *Keras* з цього середовища. При цьому обрано, що функцією активації у прихованих блоках моделі мережі є сигмоїдні функції, як такі, що дозволяють досить ефективно згладжувати викиди при формулюванні гіпотез в рамках роботи моделі. Представлено програмний алгоритм реалізації рекурентної нейронної мережі для розпізнавання мови. Знайдено, що при аналізі сигналів нейронна мережа при навчанні з вчителем дає високу точність розпізнавання вже на 8 циклі ітерації. При цьому отримано, що ентропійна функція втрат досягає мінімального значення на рівні  $9,389 \times 10^{-5}$  наприкінці функціонування циклів моделі. Це у свою чергу свідчить про високу точність роботи нейронної мережі та підтверджує можливість її впровадження в системах розпізнавання мови.

**Ключові слова:** нейронні мережі, система розпізнавання мови, сигнал, аудіо, оброблення, фреймворк, функція, архітектура.

**Постановка проблеми.** В останні декілька років у численних областях науки і техніки все більшої популярності набуває використання нейронних мереж при вирішенні різноманітних прикладних задач. Так можна відмітити стрімкий розвиток технологій, пов'язаних з нейронними мережами в авіа- та автомобільній промисловостях. Технічні засоби в безпілотних автомобілях для оцінки зображень з вбудованих камер і для оброблення голосових команд використовують такі мережі

в складі бортових електронних комп'ютерів. В авіаційній індустрії бортові навігаційні системи для передавання та аналізу голосових команд також використовують нейронні мережі. Іншими напрямками застосувань нейронних мереж є оброблення кольорових зображень (аналіз тестових підписів до зображень), синтез та розпізнавання мовних сигналів і текстових повідомлень, оптичне розпізнавання графічних символів, створення фото реалістичних електронних портретів

голови людини. Серед наведених прикладних застосунків мереж особливий інтерес викликають електронні системи розпізнавання мови. Задача розпізнавання мовних сигналів на сьогодні вважається досить перспективним науково-технічним напрямком. Розпізнавання мови на основі візуальних ознак використовується, наприклад, при біометричній ідентифікації, а також при реалізації інтерфейсів ведення інформації. Для отримання візуальних ознак використовуються підходи, які є близькими до фізіологічного сприйняття звуків людиною. Проте тут є особливість, яка полягає у тому, що для смуг частот до 1100 Гц суб'єктивне сприйняття звуку при подвоєнні частоти майже лінійно співпадає з фізичним збільшенням частоти у 2 рази. Тоді як для частот вищих за 1100 Гц залежність подвоєння частоти більше нагадує логарифмічну. Для врахування цих особливостей сприйняття звуку рекомендується переходити від частотного діапазону до мел-шкали [1, с. 432]. Тобто значення частоти в Гц перетворюється в безрозмірне значення “висоти” сигналу. Для імітації особливостей сприйняття звуку людиною при виділенні ознак у дослідженні будемо використовувати мел-кепстральні коефіцієнти [2, с. 518]. Ці коефіцієнти визначають застосування до відрізка мовного сигналу дискретного перетворення Фур'є, спектрального згладжування, зведення до логарифмічної шкали і застосування дійсної частини прямого перетворення Фур'є. Така постановка задачі повинна бути врахована при розгляді систем розпізнавання мови. Слід відмітити, що досить популярні серед розробників варіанти реалізації систем розпізнавання мови на основі прихованих марківських моделей мають обмеження, які, зокрема, характеризуються низькою швидкістю при синтезі довгих послідовностей мовних сигналів з часовими параметрами. До того ж, “живий” мовний сигнал значно відрізняється від штучних сигналів своєю складністю за характеристиками. Ця складність обумовлена тим, що насправді існують зв'язки між окремими звуками мови. Тобто, варіативність звукових символів за часом відіграє при цьому не останню роль. Для уникнення наведених обмежень в дослідженні пропонується розглянути рекурентні нейронні мережі. Ці мережі відрізняються від схованих марківських моделей і тим, що в останніх при збільшенні контекстного вікна об'єм розрахунків стає надвеликим і моделювати на їх основі довгострокові залежності стає практично неможливим. Крім цього, рекурентні мережі можуть знаходити довгострокові часові залежності у даних при аналізі.

**Метою статті** є розгляд рекурентної нейронної мережі як основи для створення електронної системи розпізнавання мовних сигналів, а також розроблення програмного алгоритму на основі фреймворку для реалізації нейронної мережі, її тестування та навчання.

Для досягнення поставленої мети були сформульовані наступні завдання:

- визначення ключових особливостей створення рекурентних нейронних мереж для електронних систем розпізнавання мови;
- створення програмного алгоритму з залученням відповідних бібліотек для імітації моделі рекурентної нейронної мережі;
- перевірка програмного алгоритму на основі тестових словосполучень в середовищі об'єктно-орієнтованого програмування `rsucharm`.

**Виклад основного матеріалу дослідження.** Нейронні мережі – це обчислювальні моделі, в яких множина простих блоків (нейронів) паралельно функціонують за умови відсутності центрального керуючого блоку [3, с. 555]. Ваги цих блоків, через відповідні коефіцієнти визначають принципи запам'ятовування інформації в нейронних мережах. Натомість, оновлення значень цих вагових коефіцієнтів а також зміщень – ключ до навчання нової інформації. Нейронні мережі завдяки своїй структурі здатні запам'ятовувати та аналізувати вхідну інформацію. Узагальнено, модель мережі можна представити через матрицю  $A$ , яка описується через систему рівнянь  $Ax=b$ , де сама матриця має вхідні дані, а вектор  $b$  – вихідні мітки для кожного рядка цієї матриці. В якості вектора параметра  $x$  якраз і виступають зазначені вище вагові коефіцієнти зв'язків між блоками нейронної мережі. Через правильно підібрану комбінацію вагових коефіцієнтів можна отримати правильні відповіді від нейронної мережі. Підбір правильної комбінації досягається під час навчання мережі [4, с. 1058]. На основі нейронних мереж з легкістю будуються ієрархічні багаторівневі структури і при цьому зберігається їх прозорість. Так як мова є складальною структурою і її можна розбити на окремі слова, літери, то систему розпізнавання доцільно будувати як ієрархічну структуру. Таки чином модель нейронної мережі являє собою набір взаємодіючих вузлів – нейронів. Вихід нейрону визначається через активаційну функцію. Архітектура нейронної мережі визначається наступними характеристиками: кількість нейронів, кількість шарів, типи зв'язків між шарами.

Найбільш простий тип нейронної мережі за архітектурою – багат шарова мережа прямого

розповсюдження. Ця мережа базується на ідеї біологічного нейрону. Вона має вхідний шар, один чи декілька прихованих шарів та один шар на виході. Кількість нейронів у кожному шарі може бути різною, але кожен нейрон одного шару пов'язаний з усіма сусідніми шарами. Зв'язки між нейронами утворюють граф (рисунок 1).

Через вхідний шар в нейронну мережу надходять вхідні дані. Кількість нейронів у вхідному шарі зазвичай співпадає з кількістю вхідних ознак. У прихованих шарах, а точніше через вагові коефіцієнти між цими шарами закодована інформація, яка вилучається мережею з вхідних даних у процесі навчання мережі. Основне завдання прихованих шарів – виділення специфічних ознак у вихідних даних. Залежно від архітектури використовується певна підмножина функцій активації шару. Для вхідного шару моделі функція активації не задається, оскільки маємо вхідний вектор ознак сигналу без змін.

Результат роботи нейронної мережі отримують через шар на виході. Залежно від конфігурації мережі результат може бути як дійсним числом так і набором ймовірностей і це залежить від функції активації в шарі на виході. Як правило в нейронних мережах для отримання набору ймовірностей використовується в шарі на виході функція softmax або сигмоїда. З позиції математичних співвідношень сигмоїдний шар з одним виходом – теж саме, що softmax-шар з двома нейронами.

Рекурентні нейронні мережі є різновидом нейронних мереж прямого розповсюдження і відрізняються тим, що можуть передавати інформацію між часовими кроками. Тобто, в такій мережі обирається по одному вхідному вектору і проводиться за ним моделювання. В процесі моделювання мережа може зберігати свій стан упродовж деякого часового вікна вхідних векторів. Тобто, між входом і виходом цільових функцій, які використовуються при функціонуванні мережі можуть бути часові залежності. Самі мережі моделюють часовий аспект даних на основі проходження циклів. В рекурентних нейронних мережах можуть бути циклічні зв'язки і вони можуть моделювати поведінку у часі, що є ключовою перевагою при роботі з звуковими даними та при проведенні процедур з розбірливості мови. В таких мережах є петля зворотного зв'язку, яка дозволяє проводити навчання нейронної мережі на

основі послідовностей з часовими параметрами. Крім цього, мережі мають додаткову матрицю зв'язків між часовими кроками і на основі якої можна знайти часові зв'язки, котрі присутні у даних. В рекурентних мережах на вході може бути декілька вхідних векторів даних, по одному для кожного часового кроку і у кожного вектора може бути декілька стовпчиків. Наведемо приклади послідовностей вхідних та вихідних векторів в таких мережах:

- задача підписування зображення. На вхід мережі подається зображення, а на виході маємо послідовність слів;
- задача аналізу тональності звуку. На вході звукові дані, а на виході значення параметру;
- задача класифікації послідовності відеокадрів. На виході отримуємо мітки, які характеризують конкретний кадр зображення.

На рисунку 2 представлено приклад зміни вхідних значень моделі на основі часових кроків. З аналізу даних рисунка 2 виходить, що значення у часових кроках можуть бути відсутні і повинно бути маскування, щоб була інформація де у векторі знаходяться реальні дані. Для цього вводиться додаткова матриця-маска у вигляді рядка, де відмічаються часові кроки (0 або 1), на яких присутні вхідні дані хоча б в одному стовпчику. Кількість станів в рекурентній мережі представлено через прихований шар і ця кількість експоненціально залежить від кількості блоків у шарі.

**LSTM-моделі.** У другій половині 90-х років ХХ століття Сепп Хохрайтер разом з іншими науковцями створили покращені варіанти штучних нейронів, які отримали назву LSTM (Long Short-Term Memory). І LSTM-мережі - це найбільш розповсюджений варіант рекурентних нейронних мереж. Важливими елементами в таких мережах є комірки пам'яті та так звані "вентилі" (вентиль входу та вентиль забуття). Якщо обидва вентилі

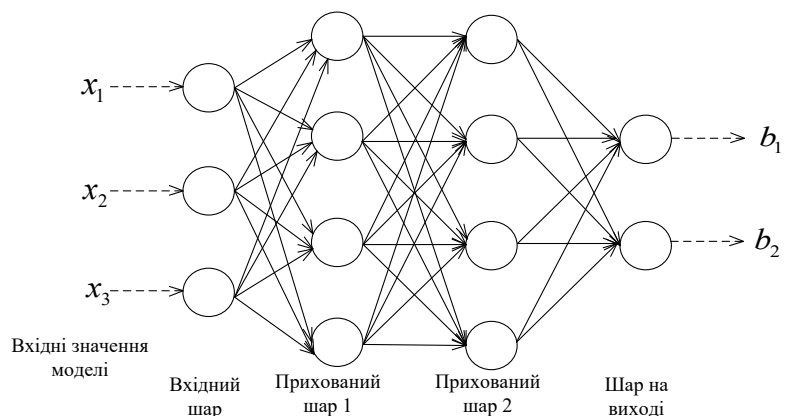


Рис. 1. Типова архітектура нейронної мережі

зачинено, то вміст блоку (нейрону) залишається незмінними при переході від одного часового кроку до іншого. Вентилі дозволяють зберігати інформацію у блоці моделі (комірці пам'яті) упродовж великого числа часових кроків. Моделі на основі LSTM-нейронів здатні запам'ятовувати інформацію на деякому проміжку часу з подальшим прогнозуванням на основі цієї інформації. Тобто, на основі попередніх даних мережа здатна спрогнозувати значення на виході одного з блоків моделі. Ці мережі можуть використовувати власну внутрішню пам'ять для оброблення послідовностей довільної довжини. До цієї послідовності можна віднести словосполучення як складових записаного мовного сигналу.

Вектори-стовпчики	Слова	Значення
	Один	0
	Два	1
	Три	0,5
	Чотири	0



Вектори-стовпчики	Часові кроки					
	Слова	0	1	2	3	4
	Один	0	0	0,2	0	0
	Два	0	0	0	0	0,7
	Три	0	0	0,5	0	0
	Чотири	0	0	0	0,9	0

Рис. 2. Часова залежність появи вхідних даних моделі

Саме такі моделі найчастіше використовуються в системах розпізнавання мови та тексту, а також при поліморфному моделюванні музики. В рекурентних нейронних мережах архітектура з рисунку 1 модернізується через додавання зворотного зв'язку між виходом і входом нейрону мережі. На рисунку 3 наведена спрощена структура такої мережі зі зворотнім зв'язком нейронів у прихованих шарах.

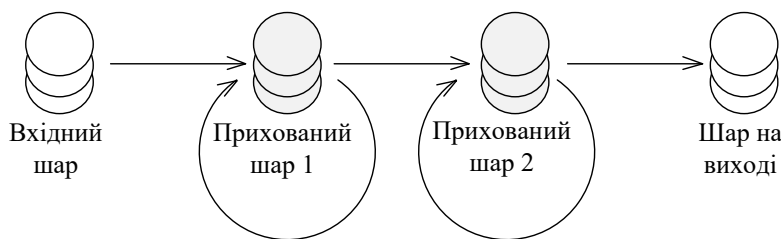


Рис. 3. Архітектура LSTM-моделі нейронної мережі

До структури блоку LSTM-мережі входять три вентилі (вхідний (input gate), вихідний (output gate) та забуття (forget gate)), комірці пам'яті (cell), вхід блоку та вихідна функція активації. Вхідний вентиль захищає блок моделі від несуттєвих вхідних подій. Вентиль забуття дозволяє стирати вміст комірці пам'яті. Вентиль на вході, блокує або ні, дані вмісту комірці пам'яті на виході блоку моделі. Вихід блоку LSTM-моделі рекурентно з'єднується з входом та усіма вентилями блоку.

Навчання нейронної мережі означає, що проводиться пошук найкращої множини значень параметрів моделі. З іншого боку, навчання мережі можна представити, як задача оптимізації, де потрібно мінімізувати функцію втрат відносно параметрів функції передбачення (на основі моделі). Алгоритми оптимізації передбачають розрахунок або матриці Якобі (якобіан, матриця, яка складається з частинних похідних функції втрат по кожному параметру) або похідної якобіана. Функція втрат кількісно визначає відхилення передбаченого від фактичного. Іншими словами, розраховується певна метрика, яка базується на спостереженні помилки передбачення. Далі помилки агрегуються по усьому набору даних і усереднюються. В результаті цієї процедури маємо число, яке показує наскільки близько знаходимось від ідеалу (еталону). Пошук ідеального стану аналогічний пошуку параметрів (вагових коефіцієнтів), які мінімізують "втрату" через помилки. Тим самим функція втрат дозволяє розглядати завдання навчання нейронної мережі, як завдання оптимізації. Як правило, оптимальні значення параметрів нейронної мережі знаходять через алгоритми апроксимації. Значення функції втрат залежить лише від вагових коефіцієнтів та зміщень нейронної мережі, а це визначає стан нейронної мережі.

У добре навченій нейронній мережі вагові коефіцієнти підбираються таким чином, що сигнал підсилюється, а шум навпаки послаблюється. Чим більше вага, тим сильніше кореляція між сигналом та виходом мережі. Входи нейронів з більшими ваговими коефіцієнтами впливають на інтерпретацію даних сильніше, ніж входи з малими ваговими коефіцієнтами. Процес навчання в алгоритмі на основі вагових коефіцієнтів полягає у послідовному перерахунку вагових коефіцієнтів так, що одні збільшуються, а інші зменшуються. У підсумку, значимість одних частин інформації збільшується, а інших зменшується. Це дозволяє моделі визначити,

які ознаки з якими виходами пов'язані і за цим можна переналаштувати вагові коефіцієнти моделі. У більшості наборів даних певні ознаки сильно корелюють з певними мітками. Ці зв'язки нейронна мережа обробляє самостійно і формує гіпотези на основі вхідних даних за якими далі вимірює правильність отриманих результатів.

У всіх трьох вентилях блоку LSTM-мережі використовуються сигмоїдні функції активації. В якості функції активації на вході та на виході використовується функція гіперболічного тангенсу. Шар в нейронних LSTM-мережах передбачає вхідний одновимірний вектор  $x$ , який є не фіксованим. На вихід шару впливає вхід та його історія через використання рекурентних зв'язків. В таких мережах для оновлення вагових коефіцієнтів та їх зміщень використовується навчання з викладачем.

#### **Програмна реалізація рекурентної нейронної мережі на основі фреймворку Keras.**

В рекурентних нейронних мережах враховується послідовний характер вхідних даних. Тобто на вхід такої мережі можна подавати текст, мову, часовий ряд, де поява елемента в послідовності залежить від попередніх елементів цієї послідовності. Задача розпізнавання мови полягає в автоматичному відновленні тексту зі слів, фраз, які промовляються людиною чи голосовою електронною системою.

Структурна схема типової системи розпізнавання мовних сигналів показана на рисунку 4. Мовний сигнал після перетворення в цифрову форму розбивається з деяким сталим кроком на вікна і для кожного блоку в модулі акустичного аналізу вираховується вектор значень деяких спектральних параметрів, і насамперед це стосується розрахунку мел-кепстральних коефіцієнтів. Далі вектори параметрів подаються послідовно на вхід блоку локального розпізнавання, який об'єднує еталонні блоки усіх слів мови. При надходженні на вхід цього модуля кожного нового вікна модифікується орієнтований граф гіпотез розпізнавання. Тобто, додаються нові гіпотези щодо вимовленої послідовності слів мови та видаляються гіпотези, ймовірність котрих менше певного порогу. В результаті такого аналізу, залишаються лише ті гіпотези, які завершуються на певне слово мови. Для ефективної роботи цього модуля суттєву роль відіграє вибір фонетичного алфавіту. І сама модель фактично проходить навчання на основі текстових та акустичних баз даних. Текстові бази

необхідні для навчання та перевірки ефективності мовних моделей, а акустичні – для налаштування параметрів алгоритмів локального розпізнавання. Побудуємо алгоритм функціонування рекурентної нейронної мережі в середовищі `runcharm` враховуючи вище наведені етапи структурної схеми з рисунку 4.

Фреймворк Keras являє собою шар рекурентної мережі SimpleRNN, де використовуються вентиля та комірки пам'яті. Використання можливостей фреймворку Keras дозволяє побудувати систему розпізнавання мови [5, с. 45].

В рамках дослідження побудуємо алгоритм взаємодії ланок рекурентної нейронної мережі для розпізнавання словосполучень на основі інструментів та бібліотек об'єктно-орієнтованого програмування `runcharm`.

Спочатку імпортуємо бібліотеки-модулі `librosa`, а з підключеного таким чином фреймворку Keras завантажуюмо складові нейронної мережі `sequential`, `LSTM`, `Dense`. На основі цих складових підключаються програмні можливості з акустичного аналізу вхідного сигналу.

```
import librosa
from os import listdir
import numpy as np
from sklearn.model_selection import train_test_split
from keras.models import Sequential
from keras.layers import Dense, LSTM
```

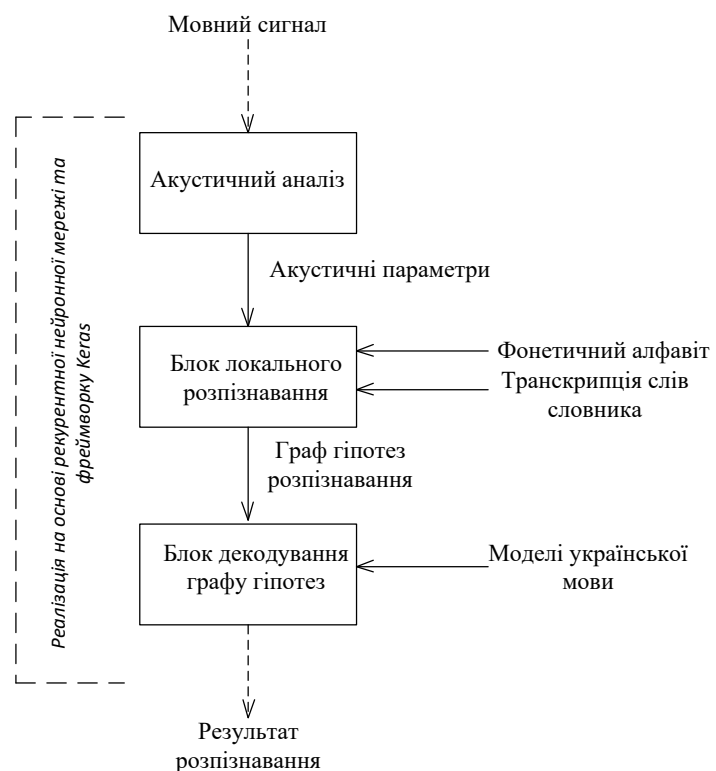


Рис. 4. Варіант системи розпізнавання мовних сигналів

Після цього, завантажуюмо тестові сигнали для роботи нейронної мережі – українські словосполучення, які записано через мікрофон. Але до цього, необхідно розрахувати функцію MFCC (мел-кепстральні коефіцієнти), яка враховує звукове сприйняття сигналів слуховою системою людини:

```
def
loadSound(path):
soundList = listdir(path)
loadedSound = []
for sound in soundList:
Y, sr = librosa.load(path + sound)
loadedSound.append(librosa.feature.mfcc(Y, sr=sr))
return np.array(loadedSound)
```

Тестові сигнали визначаються на основі наступного частину коду алгоритму:

```
first = loadSound('./ukr one/')
two = loadSound('./ukr two/')
three = loadSound('./ukr three/')
X = np.concatenate((first, two, three), axis=0)
first_label = np.concatenate((np.ones(15), np.zeros(15), np.zeros(15)))
two_label = np.concatenate((np.zeros(15), np.ones(15), np.zeros(15)))
three_label = np.concatenate((np.zeros(15), np.zeros(15), np.ones(15)))
y = np.concatenate((first_label[:, None], two_label[:, None], three_label[:, None]), axis=1)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.01, random_state=20)
```

З'єднаємо ці сигнали в один блок даних на основі функції конкатенації і сформуємо мітки. Після цього, створимо два набори вхідних даних – для навчання та тестування. Додамо послідовну модель нейронної мережі LSTM (Dense – повністю з'єднані нейрони). Кількість нейронів складає з 256 до 3 (ємність моделі). Функцією активації в нашому прикладі буде функція softmax, яка нормує оцінки, перетворюючи їх у ймовірності.

```
model = Sequential()
model.add(LSTM(units=256,return_sequences=True))
model.add(LSTM(units=128,return_sequences=True))
model.add(LSTM(units=64,return_sequences=True))
model.add(LSTM(units=32))
model.add(Dense(units=128, activation='relu'))
model.add(Dense(units=64, activation='relu'))
model.add(Dense(units=32, activation='relu'))
model.add(Dense(units=3, activation='softmax'))
```

Оцінка точності роботи моделі в режимі тестування та навчання визначається наступним рядком алгоритму

```
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
```

Запустимо модель рекурентної нейронної мережі з ознаками навчання з вчителем. Кількість циклів моделі обрано на рівні 10000.

```
model.fit(X_train, y_train, epochs=10000, validation_data=(X_test, y_test))
```

```
Epoch 1/10000
1/1 [.....] - 10s 10s/step - loss: 1.2002 - accuracy: 0.2593 - val_loss: 1.8839 - val_accuracy: 0.0047
Epoch 2/10000
1/1 [.....] - 0s 98ms/step - loss: 1.0788 - accuracy: 0.7837 - val_loss: 1.0710 - val_accuracy: 0.0047
Epoch 3/10000
1/1 [.....] - 0s 87ms/step - loss: 1.0442 - accuracy: 0.8289 - val_loss: 1.0599 - val_accuracy: 0.0047
Epoch 4/10000
1/1 [.....] - 0s 88ms/step - loss: 1.0078 - accuracy: 0.8519 - val_loss: 1.0430 - val_accuracy: 0.0047
Epoch 5/10000
1/1 [.....] - 0s 92ms/step - loss: 0.9941 - accuracy: 0.9239 - val_loss: 1.0355 - val_accuracy: 0.0047
Epoch 6/10000
1/1 [.....] - 0s 92ms/step - loss: 0.9153 - accuracy: 0.9430 - val_loss: 1.0108 - val_accuracy: 0.0047
Epoch 7/10000
1/1 [.....] - 0s 88ms/step - loss: 0.8553 - accuracy: 0.9030 - val_loss: 0.9904 - val_accuracy: 0.0047
Epoch 8/10000
1/1 [.....] - 0s 90ms/step - loss: 0.7998 - accuracy: 1.0000 - val_loss: 0.9804 - val_accuracy: 1.0000
Epoch 9/10000
1/1 [.....] - 0s 78ms/step - loss: 0.7221 - accuracy: 1.0000 - val_loss: 0.7608 - val_accuracy: 1.0000
Epoch 10/10000
1/1 [.....] - 0s 123ms/step - loss: 0.6493 - accuracy: 1.0000 - val_loss: 0.6914 - val_accuracy: 1.0000
Epoch 11/10000
1/1 [.....] - 0s 99ms/step - loss: 0.5714 - accuracy: 1.0000 - val_loss: 0.6217 - val_accuracy: 1.0000
Epoch 12/10000
1/1 [.....] - 0s 91ms/step - loss: 0.4876 - accuracy: 1.0000 - val_loss: 0.5638 - val_accuracy: 1.0000
Epoch 13/10000
1/1 [.....] - 0s 91ms/step - loss: 0.4034 - accuracy: 1.0000 - val_loss: 0.4834 - val_accuracy: 0.0047
Epoch 14/10000
1/1 [.....] - 0s 81ms/step - loss: 0.3109 - accuracy: 1.0000 - val_loss: 0.4071 - val_accuracy: 0.0047
Epoch 15/10000
1/1 [.....] - 0s 81ms/step - loss: 0.2102 - accuracy: 1.0000 - val_loss: 0.4035 - val_accuracy: 0.0047
```

Рис. 5. Результат роботи моделі нейронної мережі

```
Epoch 9998/10000
1/1 [.....] - 0s 98ms/step - loss: 9.4757e-05 - accuracy: 1.0000 - val_loss: 0.0897 - val_accuracy: 1.0000
Epoch 9999/10000
1/1 [.....] - 0s 88ms/step - loss: 9.4324e-05 - accuracy: 1.0000 - val_loss: 0.0810 - val_accuracy: 1.0000
Epoch 10000/10000
1/1 [.....] - 0s 35ms/step - loss: 9.3696e-05 - accuracy: 1.0000 - val_loss: 0.0896 - val_accuracy: 1.0000
Process finished with exit code 0.
```

Рис. 6. Результат роботи моделі наприкінці

За умовами побудованого програмного алгоритму мовні сигнали подаються на вхід моделі послідовно. Тобто, три слова: “один”, “два”, “три” з різною інтонацією надходять на вхідний шар нейронної мережі. Враховуючи, що модель працює у двох режимах, і проходячи набори вхідних даних відбувається її навчання, що показує рисунок 5. Відмітимо, що функція втрат поступово зменшується, а максимальна точність розпізнавання циклу з трьох слів вже досягається на 8 циклі. Тобто, можна зробити висновок, що мережа не є навантаженою і сам процес розпізнавання відбувається дуже швидко. На рисунку 6 наведемо останні кроки роботи нейронної мережі.

В результаті отримуємо точність роботи алгоритму на рівні приблизно  $9,3 \times 10^{-5}$ . При цьому, один цикл проходження зменшився до 83 мс (рис.6), тобто в результаті навчання і формування правильних гіпотез нейронна мережа працює з розпізнавання окремих слів швидше.

**Висновки.** Проаналізовано ключові особливості зі створення програмної моделі рекурентної

нейронної мережі для системи розпізнавання мови і визначено ключові переваги цієї архітектури над прихованими марківськими моделями. Знайдено, що для роботи з мовними сигналами необхідно враховувати суб'єктивне сприйняття звучання людиною і тому, для універсальності моделі мережі запропоновано використовувати замість звичайної частотної шкали в Гц мел-шкалу, яка визначає “висоту” сигналу. Побудовано схему електронної системи розпізнавання і на її основі створено програмний алгоритм з підключенням спеціальних бібліотек-модулів. В якості перевірки моделі нейронної мережі обрано три слова і показано, що створений алгоритм досягає цілі за критерієм високої точності та мінімумом функції втрат вже на 8 циклі роботи. На основі проходження 10000 циклів (приблизно 14 хвилин роботи мережі) вдалось отримати точність роботи алгоритму на рівні приблизно  $9,3 \times 10^{-5}$ . Отримані результати дозволяють стверджувати про здатність створеної моделі до розпізнавання складних мовних фонетичних конструкцій з включенням іноземних слів.

#### Список літератури:

1. Beigi H. Fundamentals of speaker recognition. Recognition Technologies, Inc., Yorktown Heights : NY, US, 2011. 942 p.
2. Huang X. Spoken language processing: a guide to theory, algorithm, and system development. Prentice Hall : PTR, 2001. 936 p.
3. Patterson J., Gibson A. Deep Learning : a practitioner's approach. O'Reilly Media, Inc., 2017. 530 p.
4. Shahin I., Nassif A. Speaker identification in stressful talking environments based on convolutional neural network. *International Journal of Speech Technology*. 2021. Vol. 24. P. 1055–1066
5. Gulli A., Pal S. Deep Learning with Keras. Packt Publishing, 2017. 318 p.

#### **Borisov G.O., Trapezon K.O. STUDY OF THE FEATURES OF THE CREATION OF ELECTRONIC LANGUAGE RECOGNITION SYSTEMS BASED ON NEURAL NETWORKS**

*Software methods for implementing recurrent neural networks are defined as the basis for creating a system for recognizing speech signals based on visual features. The main links of the direct propagation neural network model were analyzed and it was noted that according to the architecture and principles of the latter, the most important elements in such networks are memory cells and switching devices - valves of two or three types depending on the purpose. Thus, gates allow you to store information in a block of the network model during a significant number of time steps, into which the input sequence is divided by duration. This approach allows the neural network to work with speech signals that have signs of variability over time. It is shown that for the correct operation of the network, it is advisable to use feedback loops for hidden layers. After all, the presence of feedback determines in this case the ability of the network to learn.*

*Three speech signals, which the network should reproduce at the minimum value of the loss function, are selected as input data to the created algorithm of the functioning of the recurrent neural network. It is noted that the basis of the operation of the software algorithm of the network is connected to the appropriate libraries-modules in the Pycharm object-oriented programming environment, in particular, the Keras framework from this environment is used. At the same time, it was chosen that the activation function in the hidden blocks of the network model are sigmoid functions, as such, which allow you to effectively smooth out emissions when formulating hypotheses within the framework of the model. A software algorithm for implementing a recurrent neural network for speech recognition is presented. It was found that when analyzing signals, the neural network during training with a teacher gives high recognition accuracy already at the 8th cycle of iteration. At the same time, it was found that the entropy function of losses reaches a minimum value at the level of  $9.389 \times 10^{-5}$  at the end of the functioning of the model cycles. This, in turn, testifies to the high accuracy of the neural network and confirms the possibility of its implementation in speech recognition systems.*

**Key words:** neural networks, speech recognition system, signal, audio, processing, framework, function, architecture.