**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА**

Факультет прикладної математики та інформатики

(повне найменування назва факультету)

Кафедра прикладної математики

(повна назва кафедри)

**Магістерська робота**

Дослідження топонімічно-етнічних залежностей на базі нейронної мережі

Виконав: студент групи ПМП-62

спеціальності 113 - прикладна математика

(шифр і назва спеціальності)

Клименко М. В

(підпис) (прізвище та ініціали)

Керівник Ящук Ю. О

(підпис) (прізвище та ініціали)

Рецензент

(підпис) (прізвище та ініціали)

Львів – 2019

***ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА***

Факультет прикладної математики та інформатики

Кафедра прикладної математики

Спеціальність 113 - прикладна математика

(шифр і назва)

**«ЗАТВЕРДЖУЮ»  
Завідувач кафедри**

**" " 20 року**

**ЗАВДАННЯ**

**НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ**

Клименку Миколі Воловимировичу \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

( прізвище, ім'я, по батькові)

1 .Тема роботи

Дослідження топонімічно-етнічних залежностей на базі нейронної мережі

керівник роботи Ящук Юрій Олександрович кандидат фізико-математичних наук

( прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені Вченою радою факультету від " " 20 року №

2. Строк подання студентом роботи 17.05.2019

3. Вхідні дані до роботи

Етнографічні карти(карта лемківщини, карта Європи), топонімічні словники

4. Зміст магістерської роботи (перелік питань, які потрібно розробити)

1. Дослідження можливості встановлення етнічно топонімічної залежності за допомогою нейронної мережі
   1. створення структури нейронної мережі
   2. підготовка тренувальних даних
   3. тестування
2. Адаптація проекту для випадку Лемківщини
3. Розробка механізму візуалізації
4. Обчислювальні експерименти та аналіз результатів

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

6. Консультанти розділів роботи

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Розділ** | **Прізвище, ініціали та посада консультанта** | **Підпис, дата** | | |
|  |  | **завдання видав** | | **завдання прийняв** |
| **1** | Стецюк Валентин Михайлович |  | 05.11.19 | 12.02.19 |
|  |  |  |  |  |

7. Дата видачі завдання

**КАЛЕНДАРНИЙ ПЛАН**

|  |  |  |  |
| --- | --- | --- | --- |
| **№ з/п** | **Назва етапів магістерської роботи** | **Строк виконання етапів роботи** | **Примітка** |
| **1** | Проектування нейронної мережі | 10.11.18-02.12.18 |  |
| 2 | Підготовка тренувальних даних | 02.12.18-16.12.18 |  |
| 3 | Розробка консольного додатку для роботи із нейронною мережею | 18.12.18-03.01.19 |  |
| 4 | Підготовка тестових даних | 03.01.19-05.01.19 |  |
| 5 | Тестування та доопрацювання нейронної мережі | 05.01.19-10.01.19 |  |
| 6 | Розробка візуального представлення отриманого результату | 10.01.19-01.02.19 |  |
| 7 | Реструктуризація нейронної мережі під конкретні етнічні групи | 05.02.19-06.02.19 |  |
| 8 | Підготовка тренувальних даних під конкретну етнічну групу | 08.02.19-01.03.19 |  |
| 9 | Підготовка тестових даних під конкретну етнічну групу | 05.03.19-20.03.19 |  |
| 10 | Тестування програми на конкретній етнічній групі | 01.04.19-25.04.19 |  |
| 11 | Аналіз та візуалізація отриманих результатів | 25.04.19-30.04.19 |  |
|  |  |  |  |



**ОЦІНОЧНИЙ ЛИСТ**

магістерської роботи студента Клименка Миколи Володимировича

прізвище, ім’я, по-батькові

групи ПМП-62 факультету прикладної математики та інформатики

***Відгук наукового керівника***

Тематика

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | |  |  | Комп’ютерне моделювання |
|  |  | |  |  | Чисельні методи |
|  |  | |  |  | Оптимізація процесів |
|  |  | |  |  | Системне програмування |
|  |  | |  |  | Бази даних |
|  |  | |  |  | Навчальні програми |
|  |  | |  |  | Інше |
| **Максимальна**  **кількість балів** | | |  |  | **Зміст** |
| 4 | |  |  |  | Складність, повнота розкриття дослідження |
| 4 | |  |  |  | Наукова новизна, елементи творчості |
| 4 | |  |  |  | Самостійність виконання, систематичність роботи |
| 4 | |  |  |  | Якість та складність програм |
|  | |  |  |  | **Оформлення** |
| 4 | |  |  |  | Стиль, грамотність  Ілюстративний матеріал  Відповідність вимогам до роботи |
|  |  |
|  |  |
| Сума балів | |  |  |  |  |
| **Коментарі**  **Науковий керівник** | |  |  |  |  |

**Оцінка рецензента \_\_\_\_\_б. Оцінка за захист \_\_\_\_\_б. СУМА БАЛІВ \_\_\_\_\_\_\_\_**

**Голова ЕК**

**РЕЦЕНЗІЯ**

на магістерську роботу

студента Клименка Миколи Володимировича

прізвище, ім’я, по-батькові

групи ПМП-62 факультету прикладної математики та інформатики

Тематика

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | Комп’ютерне моделювання |
|  |  | Чисельні методи |
|  |  | Оптимізація процесів |
|  |  | Системне програмування |
|  |  | Бази даних |
|  |  | Навчальні програми |
|  |  | Інше |
| **Максимальна**  **кількість балів** | | **Зміст** |
| 6 |  | Складність |
| 6 |  | Наукова новизна |
| 6 |  | Повнота розкриття дослідження |
| 6 |  | Якість та складність програм |
|  |  | **Оформлення** |
| 2 |  | Стиль, грамотність |
| 2 |  | Ілюстративний матеріал |
| 2 |  | Відповідність вимогам до роботи |
| Сума балів |  |  |
|  |  |  |

**Коментарі**

**Рецензент**

Львівський національний університет імені Івана Франка

Подання

голові екзаменаційної комісії

щодо захисту магістерської роботи

Направляється студент\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(прізвище та ініціали)

до захисту магістерської роботи

за спеціальністю\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(шифр і назва спеціальності)

на тему \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(назва теми)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Магістерська робота і рецензія додаються.

Декан факультету \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис)

Довідка про успішність

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(прізвище та ініціали студента)

за період навчання на факультеті прикладної математики та інформатики

з 20\_\_\_ року до 20\_\_\_ року повністю виконав навчальний план

з таким розподілом оцінок за:

національною шкалою:

відмінно\_\_\_\_\_%, добре\_\_\_\_\_%, задовільно\_\_\_\_\_%;

шкалою ECTS:

А\_\_\_\_\_%; B\_\_\_\_\_%; C\_\_\_\_\_%; D\_\_\_\_\_%; E\_\_\_\_\_%

Секретар факультету\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (прізвище та ініціали)

**Висновок керівника магістерської роботи**

Студент (ка)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Керівник роботи \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис)

" " 20 року

**Висновок кафедри про магістерську роботу**

Магістерська робота розглянута.

Студент(ка) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(прізвище та ініціали)

допускається до захисту даної роботи в Екзаменаційній комісії.

Завідувач кафедри \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (прізвище та ініціали)

" "\_\_\_\_\_\_\_\_\_\_\_\_ 2019 року

**CONTENT**

[**ABSTRACT**](http://../Downloads/Telegram%20Desktop/Diploma_Andrew.docx#__RefHeading___Toc8545509) **9**

[**ABBREVIATIONS**](http://../Downloads/Telegram%20Desktop/Diploma_Andrew.docx#__RefHeading___Toc8545510) **10**

[**INTRODUCTION**](http://../Downloads/Telegram%20Desktop/Diploma_Andrew.docx#__RefHeading___Toc8545511) **11**

[**MAIN PART**](http://../Downloads/Telegram%20Desktop/Diploma_Andrew.docx#__RefHeading___Toc8545512) **12**

[**CHAPTER 1**](http://../Downloads/Telegram%20Desktop/Diploma_Andrew.docx#__RefHeading___Toc8545513) **12**

[**CHAPTER 2**](http://../Downloads/Telegram%20Desktop/Diploma_Andrew.docx#__RefHeading___Toc8545514) **16**

[**CHAPTER 3**](http://../Downloads/Telegram%20Desktop/Diploma_Andrew.docx#__RefHeading___Toc8545514) **23**

[**CONCLUSIONS**](http://../Downloads/Telegram%20Desktop/Diploma_Andrew.docx#__RefHeading___Toc8545515) **26**

[**REFERENCE LIST**](http://../Downloads/Telegram%20Desktop/Diploma_Andrew.docx#__RefHeading___Toc8545516) **27**

# 

**ABSTRACT**

The project is aimed to show history based on math and find historical truth.

This is the investigation of the close connection between language and geographical names. The main goal is to find ethnic roots and movements for each geographical territory and city founders. Obtained results are compared with real historical data.

For example, there is provided a test for ethnic group ‘lemko’ on Ukraine, Poland, and Slovakia territory. The result of this test investigation is a map of cities that are connected to that ethnic group on provided countries territories. Obtained results are compared with real historical ‘lamkoland’ territory. Obtained results will be visualized via google map API.

# 

# ABBREVIATIONS

**NN - Neural Network**

**IPA - International Phonetic Alphabet**

**REST – Representational State Transfer**

**KML – Keyhole Markup Language**

**JSON – JavaScript Object Notation**

**MLP - Multilayer Perceptrons**

**XML - Extensible Markup Language**

**API *-* Application programming interface**

# INTRODUCTION

Very few things happen at the right time, and the rest do not happen at all. The conscientious historian will correct these defects, Herodotus.

The problem of ethnic root for territory is still very important in our lives. Unfortunately, as we can see, controversial territories cause a lot of political conflicts and even war. Investigation of ethnic root and movements helps to find historical truth. The result of this investigation can be applied in ethnography and history. Based on this investigation we can draw borders between nations and prove their rights in a certain area.

Nowadays, historians and geographers rely on their works on old historical documents and vocabulary. Obtained results they use to create the ethnic map and to show ethnic movements in a different historical period of time. They do this researches manually, dividing geography names into parts and searching them in vocabulary or searching for some substrings which are similar to some words in the vocabulary, etc. They use obtained results on a map and create an ethnic territory for that area. Based on that maps, historians create and prove their hypothesizes. These investigations bring results but require a lot of time.

The main aim of my research is the optimization of this process for searching ethnic territory and movements. I am going to show how to emulate this process of search ethnic roots by math. My theory is based on language. Language is the unique value of each ethnic. This is the best identifier, cause for complete this approach we need some unique identifier that we can represent by math. In this case, language is the best choice. Testing of this solution will be provided on ethnic group ‘lemko’(лемки). This ethnic group was spread on Poland, Slovakia, and Ukrainian territory.

I made a research about other similar works related to mine. I found one huge product on the market, which is doing a similar investigation.The “NamSor” is a big product that works with a huge amount of data to find the word’s ethnic root. Their mission is to make sense of the Big Data and help understand international flows of money (diaspora investment, remittances), ideas (brain drain, brain gain) and people (migration, diaspora, expats). They specialized data mining software recognizes the linguistic or cultural origin of personal names in any alphabet/language, with fine grain and high accuracy. Names are meaningful: they use sociolinguistics to extract semantics, deliver actionable intelligence.

## 

## **CHAPTER 1**

Let’s discuss this deeper. First of all, I will define input parameters for this problem. In this case, we have only one important parameter – language vocabulary. This parameter is the unique identifier for ethos. Output parameters are bindings between ethnos and geography name – list of countries and percentage of similarity.

Now, my task looks like so:

Input data:

1) Language vocabularies

2) Training data set of cities and villages for certain area

3) List of countries

4) Testing data set of cities and villages for certain area

Output data:

1) List of cities which are closely connected to some ethnicity

2) Map with markers for those cities

3) Comparison and analysis of this solution with real data

The idea is to create the program which takes names of cities or villages and produces their ethnic roots. Based on the obtained results we will visualize it on the map. These results will help us to determine a group of cities on the map for certain areas. In case some areas will contain a lot of cities for some specific ethnos we can determine that this territory belongs to that ethnic group.

This program is aimed to optimize the process of searching ethnic settlements and movements throw history. Right now, ethnographers solve this issue manually based on vocabulary and documents. This solution is good but requires a lot of time. My project should improve it and bring new theories about old settlements.

In my theory, I rely on such an idea. Let’s imagine that some ethnic groups found a city. Obviously, they will name it based on their language. Each language has some specifics letters, sounds or order of letters, etc. Based on it, we can say that this word sounds like an English word, for example, or other languages. Great, we have found a city who is close to some ethnic group, but one successful experiment can’t prove that the ethnic group has been settled on that territory in some period of history. In case, we have a group of cities and villages close to each other in some area, then it causes a theory that some ethnic group was able to exist and live on that territory.

I am going to solve this problem with feedforward NN. The feedforward neural network was the first and simplest type of artificial neural network devised. In this network, the information moves in only one direction, forward, from the input nodes, through the hidden nodes (if any) and to the output nodes. There are no cycles or loops in the network. I decided to use this type of NN instead of recurrency NN, cause it’s easier and more suitable for this kind of problem.

Now, let’s set up a structure for NN. The input parameter will be the word of geography name(in predefined structure), and output - list of countries and their percentage similarity, where the value in each perceptron is similarity that word to a certain language. A natural way to design the network is to encode each letter number(some uniqueness number value of the symbol in alphabetical order) and order of letter of the words into the input neurons. If the word consists of 60 characters and alphabet consists of 26 characters, then we'd have 1440=60×26 input neurons, with the values 0 and 1. The output layer size will depend on the count of languages that we are checking, with the intensities scaled appropriately between 0 and 1. Each value of output perceptron is characteristic of similarity that word to the specific language.

## While the design of the input and output layers of a neural network is often straightforward, there can be quite an art to the design of the hidden layers. In particular, it's not possible, to sum up, the design process for the hidden layers with a few simple rules of thumb. Instead, neural networks researchers have developed many design heuristics for the hidden layers, which help people get the behavior they want out of their nets. For example, such heuristics can be used to help determine how to trade off the number of hidden layers against the time required to train the network. I’ve decided to use two hidden layers (50 x 70).

## All training and testing data should be on the same alphabet for this structure. I am going to use the English alphabet for this case. So, all training data for NN, initially I will parse to the English alphabet. The obtained result I parse to KML format and upload to the google map.

Let’s show it on an example. For example I took 10 major European ethnic groups (Ukrainian, Belarus, Russian, England, Portugal, Poland, Dutch, German, France, Spain). Training data set contains 10 vocabularies for those ethnic groups. When NN is trained we can start with testing of cities.

I will provide two examples with cities ‘Chernihiv’ and ‘Volgograd’. First city is ukrainian city and other russian.

**Experiment #1**

**Pre Condition:**

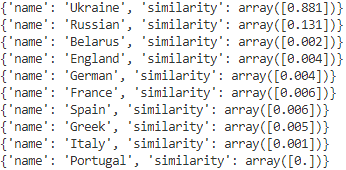
Neural Network is learned on 10 languages:

Ukrainian, Belarus, Russian, England, Portugal, Poland, Dutch, German, France, Spain

**Input Data:**

“Chernihiv”

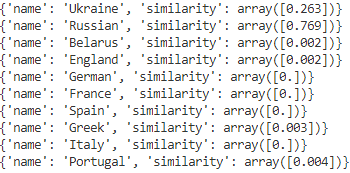
**Output:**



*pic 1*

**Analysis of result:**

So, as you can see, city ‘Chernihiv’ is the most close to Ukrainian ethnos(88%). Historical investigation is saying that city ‘Chernihiv’ is a historic city in northern Ukraine, which serves as the administrative center of the Chernihiv Oblast *(province)*, as well as of the surrounding Chernihiv Raion (*district*) within the oblast. Administratively, it is incorporated as a city of oblast significance. Chernihiv was first mentioned in the Rus'-Byzantine Treaty (907) (as Черниговъ (Chernigov)), but the time of establishment is not known. According to the items uncovered by archaeological excavations of a settlement that included artifacts from the Khazar Khaganate, it seems to have existed at least in the 9th century.

* 1. **Experiment #2**
  2. **Pre Condition:**
  3. Neural Network is learned on 10 languages:
  4. Ukrainian, Belarus, Russian, England, Portugal, Poland, Dutch, German, France, Spain
  5. **Input Data:**
  6. “Volgograd”
  7. **Output:**
  8. 
  9. *pic 2*
  10. **Analysis of result:**
  11. So, as you can see city ‘Volgograd’ is the most close to Russian(77%) ethnos. Let’s take a look in historical papers. This city was two times renamed. It’s initial name was ‘Tsarytsyn’. This city is an industrial city and the administrative centre of Volgograd Oblast, Russia. The city lies on the western bank of the Volga River. Known locally as the "Hero City".

## **CHAPTER 2**

## This program should be defined on the vocabulary of language that we are going to use. For this purpose, train our NN on that vocabularies. The input parameter will be the word of geography name(in predefined structure), and output list of countries and their percentage similarity, where the value in each perceptron is similarity that word to a certain language. NN is learned on languages(polish, ukrainian and slovakei)

## Alright, obtained NN will determine language affiliation for a city name. Now, we need some test data for checking theory. I am going to show it based on three countries, where ‘lemko’ ethnic group was mentioned in history. I took all the cities and villages of Ukrainian, Poland, and Slovakia.

## Now, we obtained a result from NN. What next? I need some graphical representation of my results. Let’s create a map of that NN outcome. The program will mark cities and villages in Poland, Slovakia, and Ukrainian, where the city name is closely connected to the Ukrainian language. It will mean that there had been living ‘lemko’ and that territory is ‘Lemkoland’. I run the list of all of that testing cities via NN and create a KML file(JSON file which allows easing to create google map).

## In the end, check the obtained result and make some consequences.

## So, the thesis consists of five milestones. Let’s decouple it and go throw step by step.

## 1) Populate uniqueness phonetic alphabet for each language (ukrainian, polish and slovakian). This will be a training data set.

## 2) Neural network for this theory

## 3) List of test data

## 4) Console application for maintaining and test NN

## 5) Graphical representation

**Training data**

The first problem appears that a different language has a different alphabet. I’ve decided to use English alphabet. But now we need one extra tool for parsing each ukraine, polish and slovakia word into the English alphabet. I’ve found some REST service which provide this functionality.

Let’s take set up language data set. One notes, we take only words which contains more than 4 letters, in other case small words will interfere to teach NN and cause a huge error. For example, we take the word ‘я’ – ‘ya’. How many words we have in different languages which contain this substring? I think, almost all languages, contain such substrings. That’s why we avoid short words.

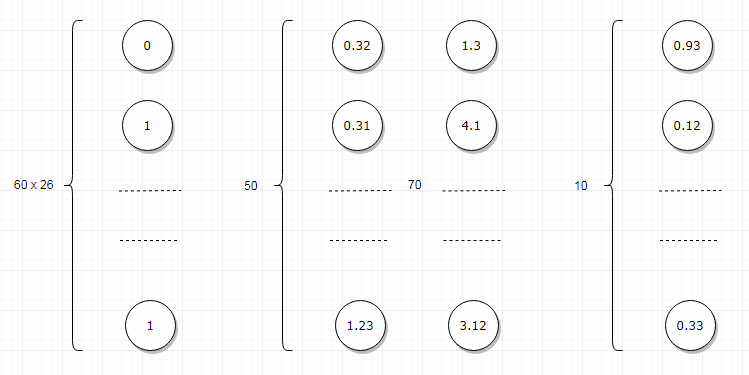
We translate each of those words to English and it produces our training data set for NN.

**Neural Network**

We create feedforward NN. A feedforward neural network is an artificial neural network wherein connections between the nodes do not form a cycle. As such, it is different from recurrent neural networks. I’ve decided to create NN on python because there are a lot of already implemented math functions that work with matrix and vectors. My NN consists of an input layer, two hidden layers, and an output layer.

The input layer contains perceptrons with a value 0 or 1 and the output layer has 3 perceptrons with a decimal value from 0 to 1 – which identify the similarity of the word to certain language(index of output perceptron identify language: Ukrainian – 0, Poland – 1, Slovakia - 2). Based on experiments and practices, two hidden layers (50 x 70) is the best choice to teach NN, for this case.

And here is the NN structure:



*pic 3*

Let’s turn back to the input layer. It consists of 60 x 26 perceptrons. I took this number because 60 – is the longest cities in Europe and English contains 26 symbols of the alphabet. For example, if we take the word ‘так’ – ‘tak’, the input layer will look like so: indexOf(‘t’) multiply 1 - (position in word). The output layer will contain the similarity of this word to language – [0.56, 0.90, 0.4]. In our case, 0.56% similarity – for Ukraine, 0.9% - Poland and 0.4% - Slovakia. So, based on this provided result, we can say that city is close connected to Ukraine with

similarity - 0.9%.

**Testing data**

Now, we need some testing data to analyze results and prove the solution. As we discussed previously, we are going to test our solution on ‘Lamkoland’(territory of Poland, Slovakia, and Ukraine). I’ve used all the cities and villages from those countries. This will be our testing data set.

Now we need to translate each city name to English. We do it via already implemented tools, that I’ve mentioned before. This tool is .NET Console Application which create a request to thirty party REST service, which is responsible for translate each language to English alphabet. For example, ‘Биківці’ will convert to ‘Bykyvtsy’. That’s it. Testing data is ready to use and analysis.

**Console application for testing and maintain NN**

All preparations are ready. We need an application to maintain and test NN. I’ve decided to write the Python Console Application for this purpose. This app consists of many python scripts:

1) **data\_representation.py** – script for the retranslating word to the input layer of NN. It’s responsible for create array of [1,0,…1] which is representing geography name.

2) **data\_sql\_script.py** – get all testing and training data from the database and create testing/training data set for NN.

3) **data\_store.py** – store NN to the JSON file and create NN from the JSON file.

4) **set\_up\_map.py** – create a KML file from the obtained result.

5) **mnist\_loader.py** – parse training and testing data to an appropriate array of items.

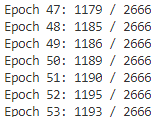
6) **network.py** – NN.

7) **network\_interface.py** – provide an interface for interaction with NN. Here are functions:

1. **learn(training\_data, 500, 100, 0.5, test\_data=test\_data)**
2. **create\_network([1,32,32])**
3. **store\_network(file)**
4. **load\_network(file)**
5. **draw\_kml(data)**

Now, let’s see how to work with API of this created console application.

Output of **learn(training\_data, 500, 100, 0.5, test\_data=test\_data)**:



*pic 4*

Epoch 47 – number of epoch. In other words, the iterator of learning.

1179/2666 – an indicator of how good NN is learned on iteration number 47. The first number ‘1179’ - count of success cases for current weights and biases of NN structure and ‘2667’ - count of all testing data. We see that on iteration 53 probability of NN was 1179/2667 = 45%. We get 500 – epochs. At the end of learning, the probability was 96.23%.

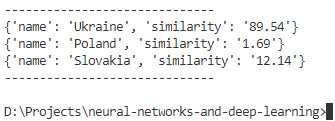
This logger data is aimed to help me better understand how good my NN is learned and gives me control of NN error. When I see the error isn’t good for provided training data set than I will reconsider my choice of training data, epochs count, batch size of training data and learning rate.

For example, if I will call function to learn parameters:

1. epochs = 40,
2. mini\_batch\_size = 100,
3. eta - learning rate = 0.1
4. training data - the same as in previous example

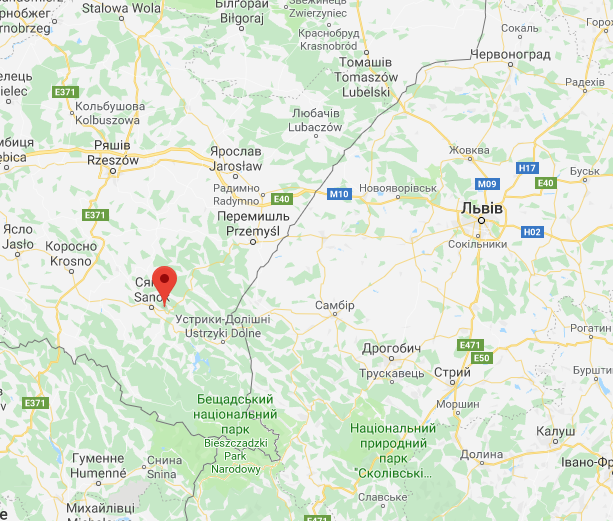
I will get the probability of NN - 39.9%

Now, let’s provide a real example for ‘Lemkoland’ territory and check it with historical ethnicity. There is a village ‘Bykivtsi’ which is recognized as Ukrainian ethnicity on Poland territory. Let’s check it out via the program.

Here is an example of work for the village ‘Bykivtsi’(Бикібці): 

*pic 5*

As you can see, this village is closely connected to Ukraine language (89.54%). This village on the map is located on Poland territory, but his ethnic root is coming from ‘lamkoland’ which is Ukrainian's ethnic.



*pic 6*

**Graphical Representation**

Result of testing will be stored in JSON file with format

{

‘Rudka’: [(0, 0.23), (1, 0.56), (2, 0.97)],

…

‘Dibcha’: [(0, 0.64), (1, 0.53), (2, 0.37)],

…,

‘{City}’: [ ( {Country}, {Similarity} ) ]

}

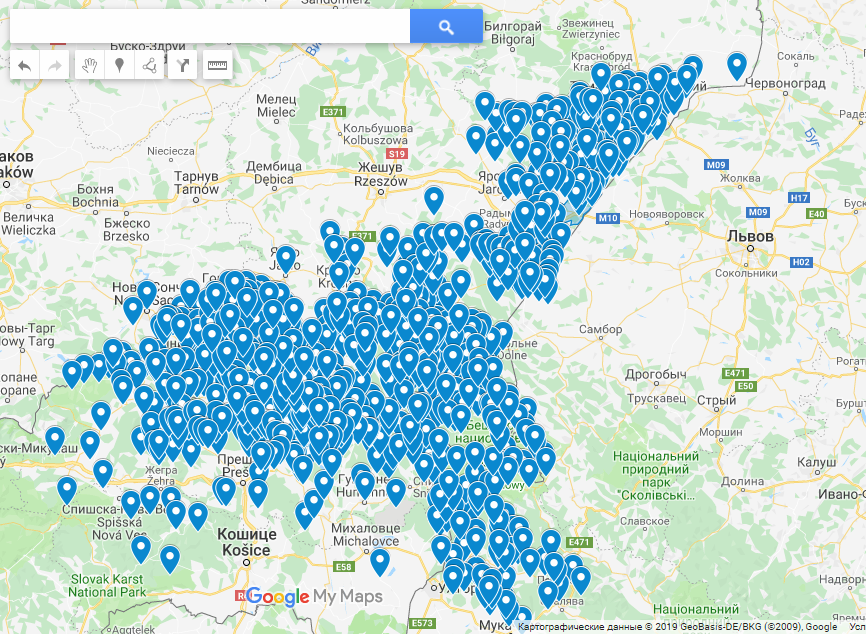
This data is used to create KML file which will be used for google map.

Function draw\_kml(data) – create \*.kml file with format:



*pic 7*

Now, upload this file to google map console and create a new map.

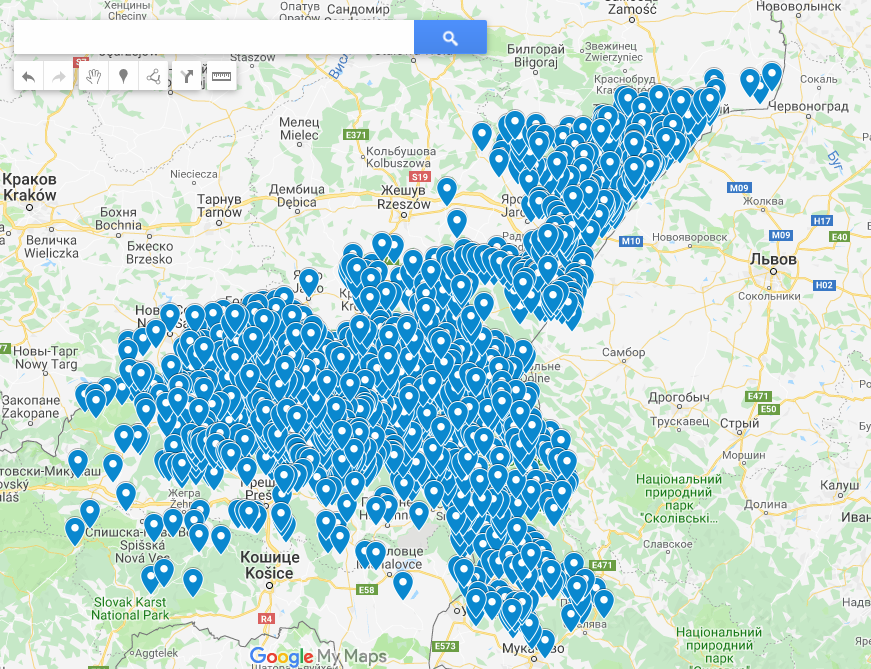


*pic 8*

This is a map obtained by NN for ‘lemko’ roots in Ukrainian, Poland and Slovakia territory. All markers on the map are cities and villages which are recognized as similar to ukraine language by NN. Based on group of markers, we can draw an area of ethnic group on the map.

**CHAPTER 3**

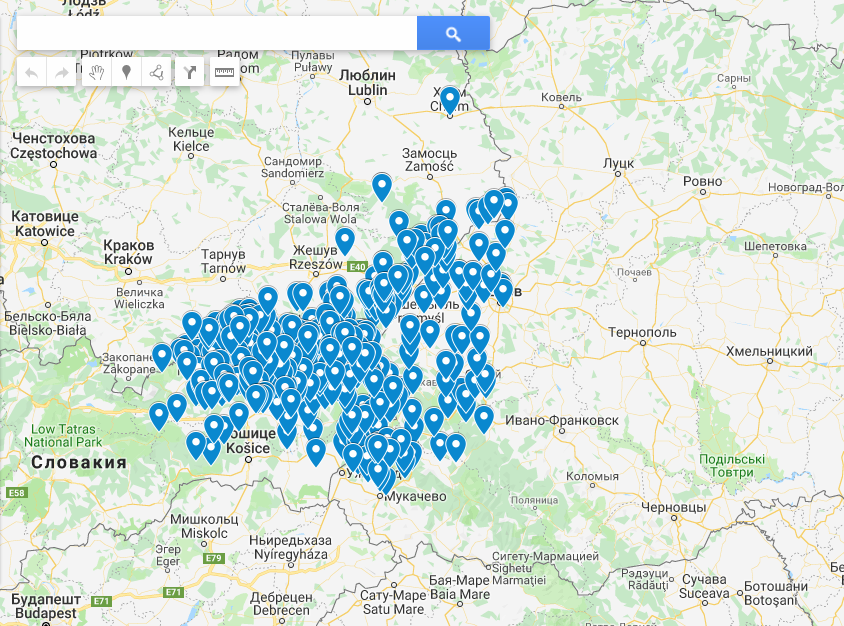
Here is provided historical map for ‘Lamkoland’:



*pic 9*

This map is from web resources which is mentioned in reference list. This is historical investigation of ‘lemko’ ethnicity on other countries territory. I will use this map as a comparison of my theory.

And this one is map created by NN based on Poland, Ukrainian and Slovakia cities:

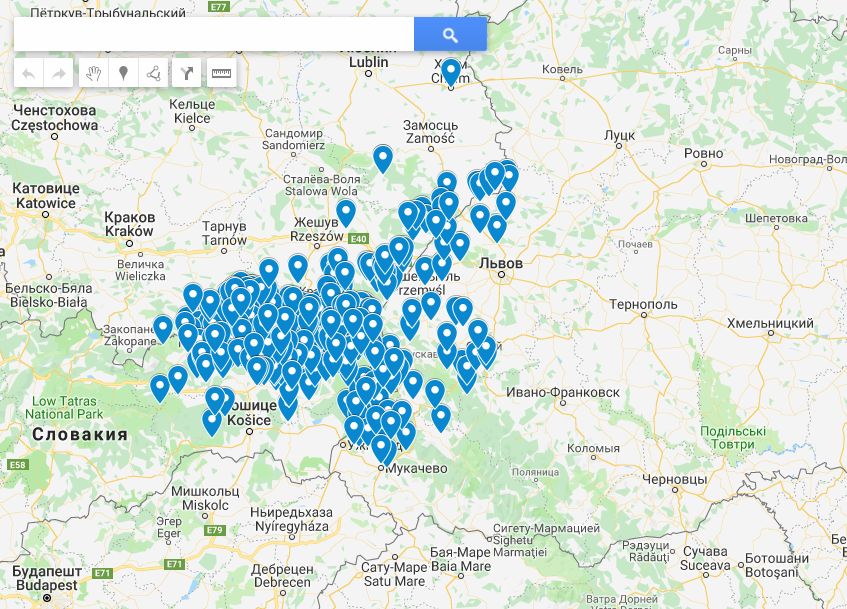


*pic 10*

As you can see, they are quite similar. Map generated by NN contains fewer cities than original and some of their cities are not present in the original historical map. In conclusion, we see that NN draws a pretty similar map to historical. That’s why we are on the right way.

It was really difficult to learn NN on that provided languages because they are similar to each other. Initially, NN has 75% precision. I’ve increased this percentage when I started to use the IPA alphabet and toponyms. Now, precision is close to 94%. It’s obvious because toponyms make my training data more uniqueness per language.

Now I am going to show result on NN with 75%:



*pic 11*

As you can see in both examples, we can detect group of cities and draw territory of ethnic group ‘lemko’. Territory provided by NN is quite similar to historical territory provided in *pic 9.* Yes, of course, on *pic 10* and *pic 11* we have less cities than

on *pic 9,* but we still can find groups of markers on map and their territory will be close to historical.

# Conclusions

This product is going to optimize historians and ethnographers researches. Now they can refer to this application to check ethnic roots for certain areas. They can start to investigate new theories about the area based on the resulted map from this project. The example with ‘Lamkoland’ shows that we are in a good way. The result was close to real historical data. 78% of cities from the original historical map coincided with the NN result.

This system can be improved with the IPA alphabet and initial city names(during history city could be renamed). The IPA will help to avoid the lost spelling of some languages' specific sounds.

Yes, there is still a lot of ways to improve this solution. Sometimes, it makes wrong decisions, but current progress shows that this idea and this implementation produce good results. In my opinion, we should move forward with this and rich better results.

Let’s show history by math.

# References list

1. Michael Nielsen  Using neural nets to recognize handwritten digits[Electronic resource] // October 2018 – Way of access:  WWW URL: http://neuralnetworksanddeeplearning.com/chap1.html
2. Richar Heningston  Neural network in depth[Electronic resource] // October 2012 – Way of access:  WWW URL:https:// worksandneuralnetcom/in-depth
3. NamSor website - Way of access: WWW URL:https://www.namsor.com
4. Lamkoland Map - Way of access: WWW URL: http://molodlem.org.ua/mapa-mist-i-sil-lemkivschyny