2022

Лабораторні роботи

O. Sychov

**Введення**

1. Практичні роботи будуть проходити з використанням бібліотеки **TENSORFLOW (** <https://www.tensorflow.org/>)
2. Програма для практичних занять написана на Python
3. В продовж всього курсу ми будемо вирішувати одну задачу – класифікацію звукових сигналів.

**Для тих, хто не володіє Python**

Крок 1. Завантажуємо <https://www.python.org/ftp/python/3.5.4/python-3.5.4-amd64.exe>

Крок 2. Інсталюємо.

Крок 3. Запускаємо cmd.exe

Крок 4. Pip install notebook

Крок 5. Jupyter notebook

Примітка: Python дуже чутливий до відступів, будьте уважні, коли будете щось міняти в коді.

**Для тих, хто володіє Python**

Ми будемо працювати у «Jupyter notebook», але якщо ви маєте вже інстальоване інше середовище для роботи з Python - працюйте в ньому.

«Jupyter notebook» - не самий гарний варіант, але він самий примітивний у роботі, для тих хто не хоче заглиблюватись у Python .

**Вхідні данні для роботи.**

**https://drive.google.com/drive/folders/1jc-8lTL3qnvB-cftx0kQbgUUkHryIfZc?usp=sharing**

1. Архів з звуковими файлами, зібраними Google для змагання .
2. Wav\_data\_prep.ipynb - Код для підготування даних
3. Wav\_NN\_Learn.ipynb - Код для навчання та тестування нейронних мереж.

**Заняття 1**

Мета: Підготовка робочого середовища, для подальшої роботи.

Необхідні кроки:

1. Інсталюємо Python
2. Jupyter notebook
3. Звантажуємо Wav\_data\_prep.ipynb
4. Запускаємо завантаження бібліотек
5. Інсталюємо всі бібліотеки, що необхідні для роботи Wav\_data\_prep.ipynb

Варіанти: якщо Ви бажаєте працювати в <https://colab.research.google.com/> або PCharmабо в іншому середовищі, це також зараховуються.

Результати заняття 1 окремо не здаються, залік буде відбуватися одразу по заняттям 1-3.

**Заняття 2**

Мета: Підготувати данні.

Пояснення:

**Частина 1:**

Всі звукові файли треба розділити на 3 класи. Кожен студент вибирає собі 2 довільних слова. Перше слово - це буде перший клас, друге – другий клас, решта слів - третій клас.

Звукові файли знаходяться у каталогах, мають різну довжину.

Для того, щоб маркувати файли і вирівняти їх довжину необхідно використати функцію ConvertTo\_8K. Функція і приклад її використання знаходяться у Wav\_data\_prep.ipynb.

Необхідні кроки:

1. Вибрати два слова.
2. За допомогою функції ConvertTo\_8K розділити всі слова на 3 класи.

**Частина 2:**

Мета: Створити Train, Test і Valid послідовності. Створити Blind test.

Пояснення:

Даних третього класу у багато разів більше, ніж даних першого і другого класу.

При навчанні мережі бажано мати збалансовані вибірки, тому данні для третього класу треба випадково прорідити, це можна зробити будь-яким зручним для вас чином.

Blind test створюється для остаточної перевірки якості моделі. Blind test створюють студенти шляхом запису свого голосу.

Необхідні кроки:

1. Збалансувати кількість класів.
2. За допомогою функції Divide\_TrainTestValid згенерувати Train, Test і Valid послідовності.

Результати заняття 2 окремо не здаються, залік буде відбуватися одразу по заняттям 1-3.

**Заняття 3**

**Частина 1:**

Мета: Перше навчання нейронної мережі. .

Необхідні кроки:

1. Завантажуємо Wav\_NN\_Learn.
2. У функції Learn\_NN\_5L\_ задаємо правильні назви каталогів.
3. Запускаємо Learn\_NN\_5L\_. Після закінчення роботи записуємо кількість, часу, необхідного для навчання однієї мережі.
4. У функції TestNN\_ задаємо правильні назви каталогів.
5. Запускаємо TestNN\_.
6. Розбираємо формати файлів результатів.
7. Записуємо перший результат.

**Частина 2:**

Підготовка до створення Blind test

Необхідні кроки:

1. Сформулювати вимоги до Blind test

2. Сформулювати технічне завдання для кожного студента для створення Blind test

Залік по лабораторним 1-3

***Результати роботи: якість класифікації звукових сигналів на базовій архітектурі загорткової мережі.***

**Заняття 4**

Мета: Оптимізація архітектури нейронної мережі прямого поширення.

Необхідні кроки:

1. Формулюємо мету експериментів і що треба зробити для її досягнення..
2. Створюємо програму експериментів.
3. У тілі функції Learn\_NN\_5L\_ міняємо архітектуру нейронної мережі відповідно до плану експериментів.
4. Запускаємо Learn\_NN\_5L\_. Отримані результати вносимо у звіт.
5. Повторюємо кроки 3-4 згідно з програмою екпериментів.
6. Проводимо аналіз отриманих результатів.
7. Коректуємо програму експериментів згідно з отриманими результатами.
8. Повторюємо кроки 3-7 поки не буде знайдена оптимальна архітектура.

***Результати роботи: графіки якості класифікації на Train, Test and Valid data set для різних архітектур нейронної мережі. Вибір оптимальної архітектури.***

**Заняття 5**

Мета: Оптимізація параметрів нейронної мережі .

***Результати роботи: графіки якості класифікації на Train, Test and Valid data set для різних параметрів нейронної мережі. Вибір оптимальної параметрів.***

**Заняття 6**

Мета: Пошук шляхів покращення якості розпізнавання. .

Необхідні кроки:

1. Аналіз існуючих проблем.
2. Генерація гіпотез стосовно причин і можливостей подолання виявлених проблем.
3. Створення програми експериментів для перевірки висунутих гіпотез
4. Перевірка гіпотез
5. Аналіз отриманих результатів

***Результати роботи: Результати аналізу проблем. Перілік висунотих гіпотез. Результати експериментів. Висновки .***

**Заняття 7**

Мета: Вибір і перевірка фінального варіанту нейронної мережі.

Необхідні кроки:

1. З усіх отриманих варіантів нейронних мереж вибираємо 3 найкращих.
2. Самий кращий з обраних варіантів називаємо – фінальним
3. Записуємо результати тестування всіх 3 мереж на Train, Test , Valid data set and **Blind test** *data set*
4. Робимо висновки

***Результати роботи: : таблиця якості класифікації на Train, Test , Valid data set and Blind test data set. Висновки***

**Заняття 8**

Мета: Ознайомлення з бібліотекою https://pytorch.org/

Необхідні кроки:

1. Завантажуємо бібліотеки..
2. Модифікуємо код Learn\_NN\_5L, заміняючи функції **TENSORFLOW на PyTorch**.
3. Проводимо навчання однієї мережі яку в попередньому занятті ви обрали як фінальну..
4. Проводимо 5 запусків навчання з різних стартових точок для бібліотеки **TENSORFLOW і так само 5** запусків навчання з різних стартових точок для бібліотеки **PyTorch. Порівнюємо результати**.

***Результати роботи: таблиця якості класифікації на Train, Test , Valid data set and Blind test data set для 5 запусків з різних стартових точок. Використовуючи* TENSORFLOW та PyTorch. Висновки.**

**Заняття 9**

Мета: Порівняння результатів, отриманих різними студентами. .

***Результати роботи: формулювання best practices по підсумках роботи.***