**Выполнил Ярош Дмитрий**

**dimajarosh@gmail.com**

# Подготовка данных

Мне были необходимы данные по животным и протистам (простейшие).

Данные были загружены [отсюда](http://gtrnadb2009.ucsc.edu/download.html) (источник 1) - это база всех эукариотов (ядерные), в них кроме животных и простейших есть также грибы и растения, поэтому их необходимо было отфильтровать.

Дополнительно были добавлены простейшие [отсюда](http://gtrnadb2009.ucsc.edu/Pfalc/Pfalc-tRNAs.fa) (не включены в общих архив, но есть в базе на сайте). В последствии дополнительно были добавлены данные [отсюда](https://rnacentral.org/expert-databases) - только по простейшим.

Для формирования файлов с данными для дообучения был реализован скрипт dataExtractor.py. Данный скрипт позволяет отфильтровать растения (т.к. часть датасета, загруженная с первого источника содержала данные по животным, растениям и простейшим), удалить дубликаты, а также перевести входной файл в два выходных - data.csv и db.csv содержащие цепочки и метаинформацию о них соответственно.

*Итог:*

29577 животных

2430 протистов

Для разделения данных на наборы для обучения, валидации и теста был реализован скрипт dataDivider.py.

На выходе скрипт выдает три файла с разделением на обучение, валидацию и тест и базу метаинформации с добавленными метками, информирующими в какой набор попал данный образец. Данный скрипт также переиспользует данные по 4м царствам, не входящим в данное исследование (добавляет их в начало каждого файла из соответствующего файла старых данных)

Для обучения сети было подготовлено данные:

* протистов 2430, часть повторена для достижения нужного количества в 2000 в выборке train, животных 3000
* соотношение train/valid/test - 2000/250/750

# Изменения модели

Для решения поставленной задачи последний слой модели “imagesExtended” был изменен для распознавания 6 классов вместо 4х. Впоследствии в некоторых экспериментах был добавлен дополнительный слой, который содержал 32 нейрона и находился перед последним слоем, однако в связи с проявлением переобучения его было решено не включать в итоговую модель.

# Обучение

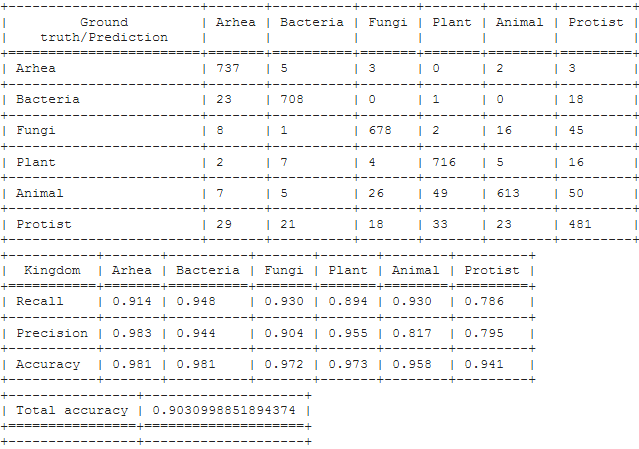
Обучение проводилось на платформе Google Colab. Код обучения можно найти в файле model.py.

Код тестирования можно найти в файле model\_test.py. Код обучения и тестирования базировался на коде обучения модели “imagesExtended”, однако код для обучения был оптимизирован (устранено чтение из файла на каждой эпохе, данные выгружены в память заранее, применен метод model.fit() с автоматическим делением на батчи на стороне keras). Данная оптимизация сократила время обучения одной эпохи со 100 секунд до 8 секунд (gpu-based). Также в коде тестирования был добавлен вывод параметра accuracy для модели и реализовано построение таблицы для демонстрации результатов тестирования.

## Были поставлены следующие эксперименты:

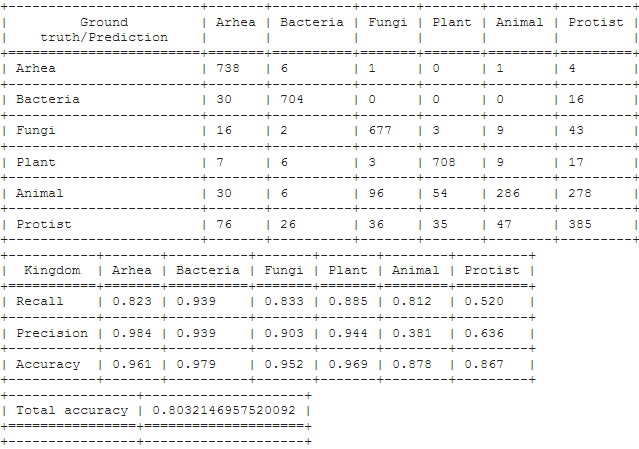
### Обучение в 150 эпох модели (без изменений параметров базовой модели)

#### **Результаты тестирования оказались следующими:**



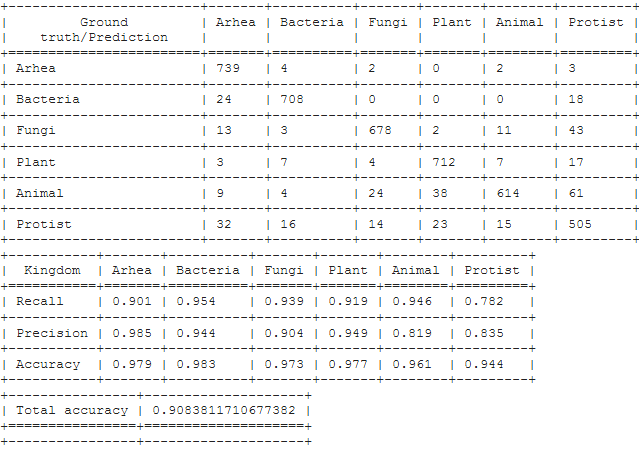
### Обучение в 150 эпох (замена оптимизатора на Adadelta(lr=0.05))

#### **Результаты тестирования оказались следующими:**



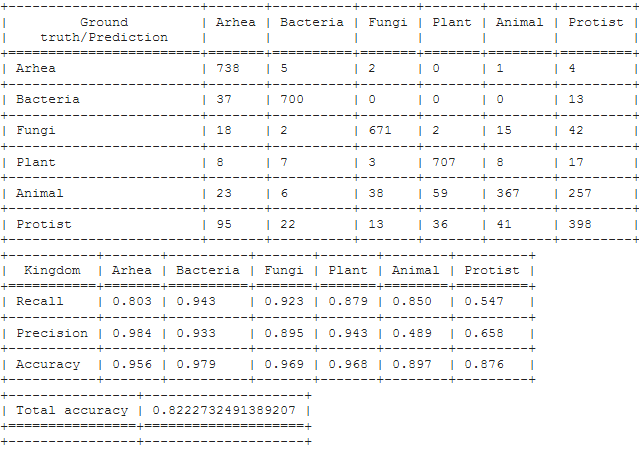
### Дообучение 150-200 эпох (стандартная модель)

#### **Результаты тестирования оказались следующими:**



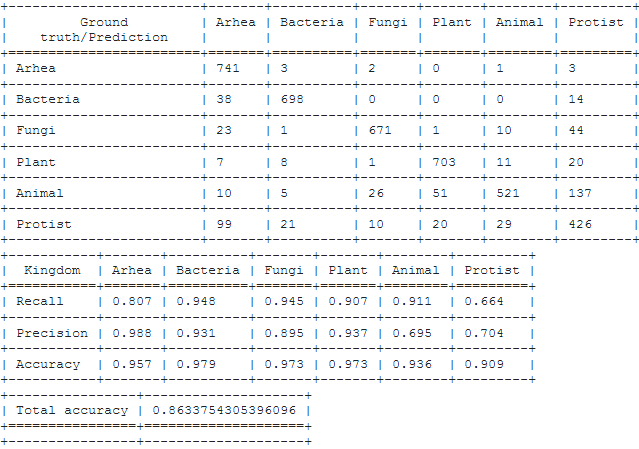
### Стандартная модель с lr=0.01

#### **Результаты тестирования оказались следующими:**



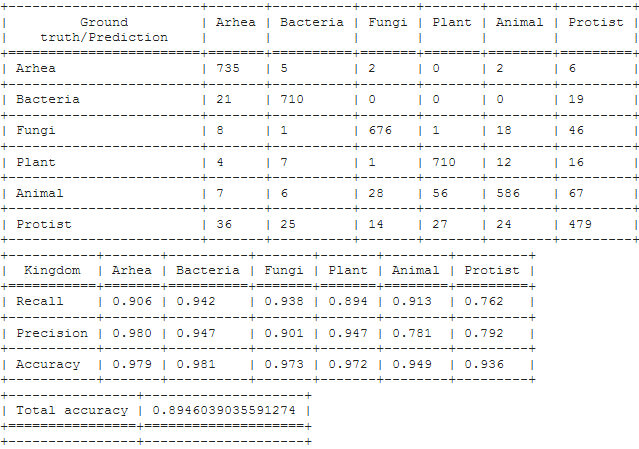
### Дообучение стандартной модели с lr=0.01 с 150 до 250 эпох

#### **Результаты тестирования оказались следующими:**



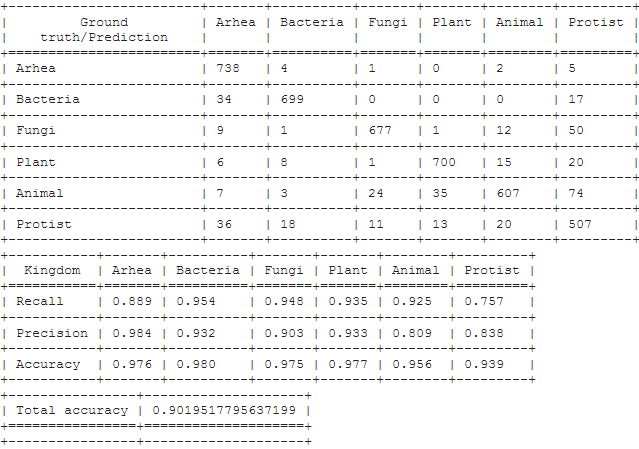
### Стандартная модель с lr = 0.03

#### **Результаты тестирования оказались следующими:**



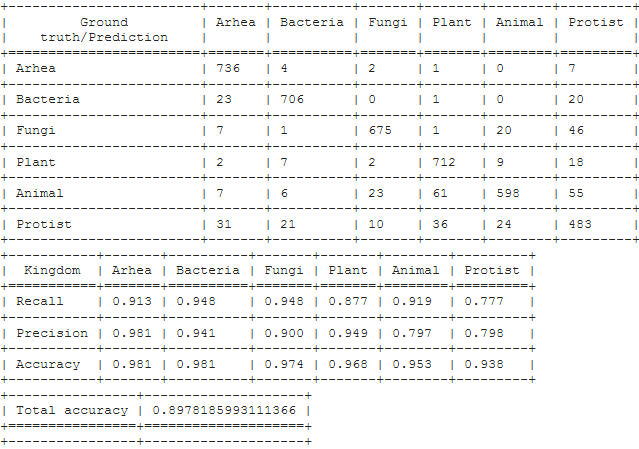
### Дообучение стандартной модели с lr=0.03 с 150 до 250 эпох

#### **Результаты тестирования оказались следующими:**



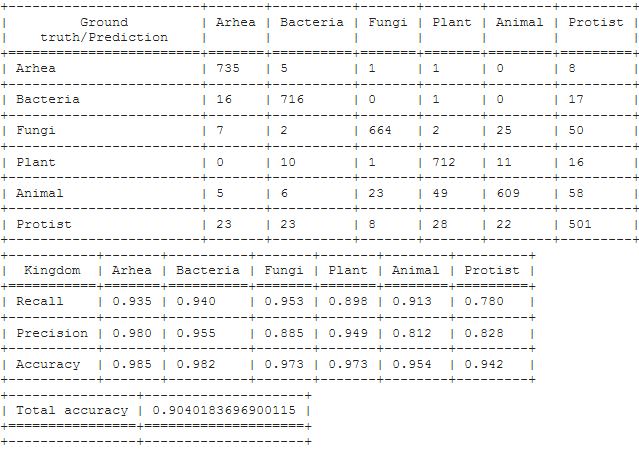
### Стандартная модель с lr = 0.04

#### **Результаты тестирования оказались следующими:**



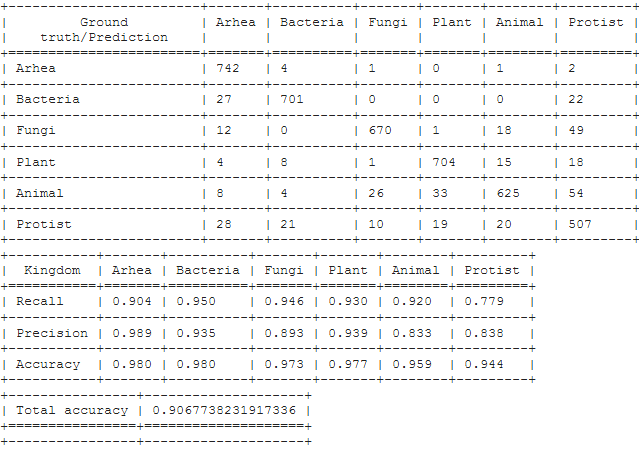
### Стандартная модель с lr = 0.06

#### **Результаты тестирования оказались следующими:**



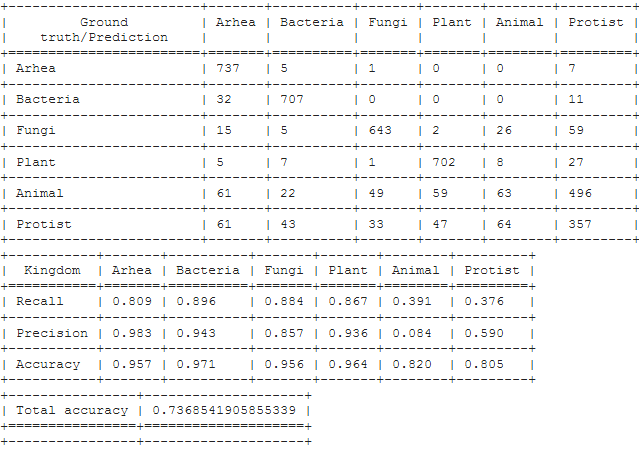
### Дообучение стандартной модели (обученной с lr=0.06) с 150 до 250 эпох с lr=0.02

#### **Результаты тестирования оказались следующими:**



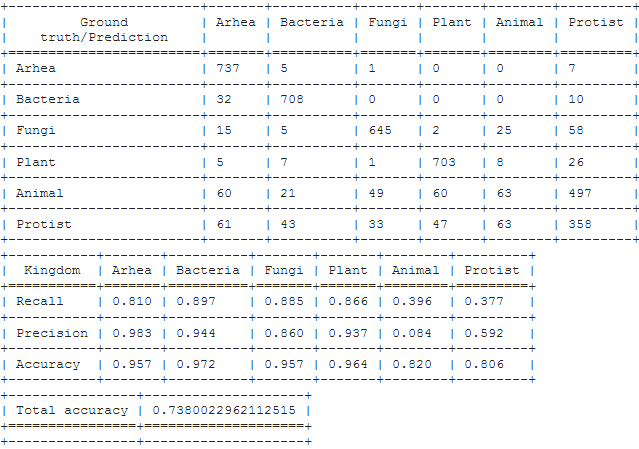
### Обучение стандартной модели с lr=0.05 при замороженных верхних слоях до Dense(1024) включительно, следующего после Dense(30420)

#### **Результаты тестирования оказались следующими:**



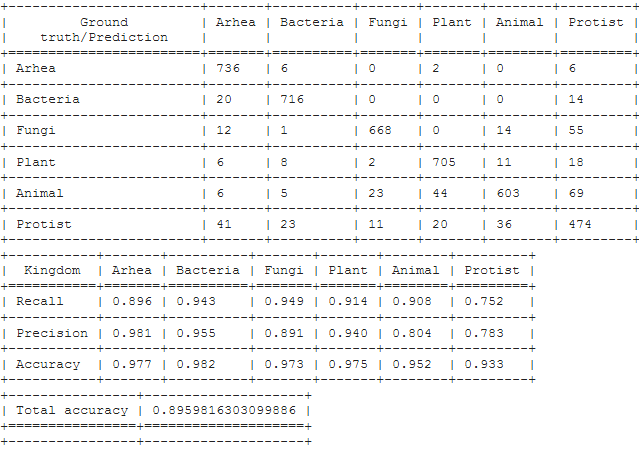
### Обучение стандартной модели с lr=0.05 при замороженных верхних слоях до Dense(30420) включительно

#### **Результаты тестирования оказались следующими:**



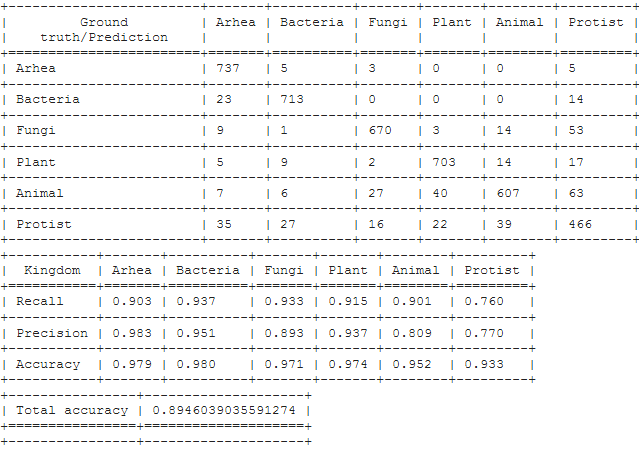
1. Обучение модели с дополнительным слоем c lr=0.05 в 150 эпох с размером батча 32

#### **Результаты тестирования оказались следующими:**



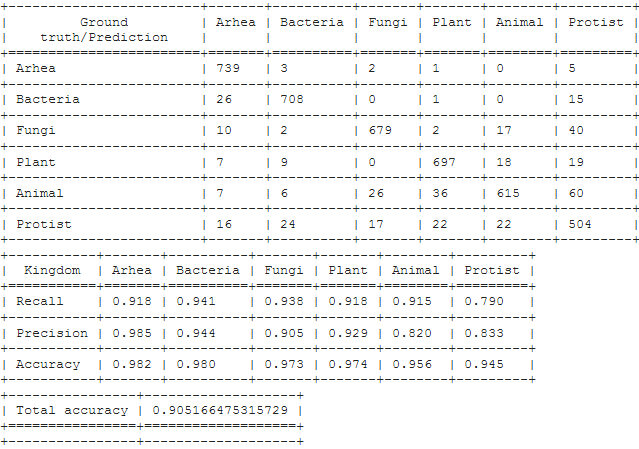
1. Обучение модели с дополнительным слоем c lr=0.04 и размером батча 32 в 150 эпох

#### **Результаты тестирования оказались следующими:**



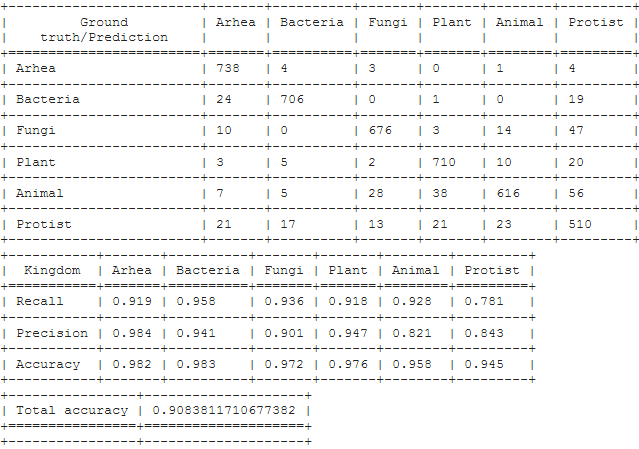
### Стандартная модель с lr = 0.05 с размером батча 32 на 300 эпох

#### **Результаты тестирования оказались следующими:**



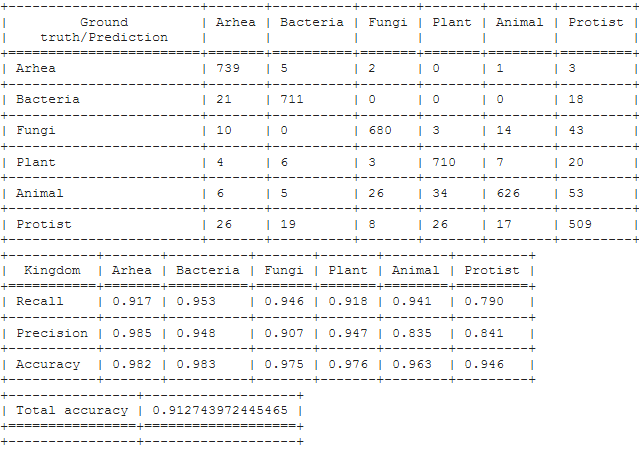
1. Стандартная модель с lr = 0.05 с размером батча 32 и авто понижением lr каждые 25 эпох на 150 эпох

#### **Результаты тестирования оказались следующими:**



### Стандартная модель с lr = 0.05 с размером батча 32 и авто понижением lr каждые 20 эпох на 300 эпох (ЛУЧШАЯ МОДЕЛЬ)

#### **Результаты тестирования оказались следующими:**



## Выводы после обучения

* Наиболее оптимальными являются lr =0.05-0.06
* Добавление новых слоев вызывает переобучение
* Заморозка части слоев не приносит эффекта
* Уменьшение размера батча позитивно сказывается на модели
* Внедрение динамического изменения lr позволяет улучшить качетсво обучения

## Итоговый результат обучения

В ходе обучения было продемонстрирована эффективность дообучения сети, ранее использованной для распознавания 4х царств живых организмов для распознавания других царств (на примере царства животных и простейших). Удалось добиться приемлемых результатов для предсказания всех 6 царств.

# Итоговый вывод

В ходе проведенного эксперимента были подготовлены данные по царствам животных и протистов (простейшие). Поставленные эксперименты позволили продемонстрировать возможность дообучения сети для распознавания новых царств.

Так же в ходе обучения и тестирования было выявлено, что наиболее частой ошибкой сети при распознавании животных является их определение как растений (вероятно это связано с тем что оба этих царства содержит высокоразвитых представителей, обладающих сложной структурой тРНК с которой существующей сети тяжело работать). В качестве возможного решения этой проблемы возможно стоит проверить гипотезу о влиянии увеличения количества данных о растениях и животных на процесс распознавания.