



Задание полуфинала номинации «Python: проектирование нейросети»

Постановка задачи

Оформление проекта

Требования, участвующие в оценке решения

Требования к способности системы

Требования к качеству кода

Требования и рекомендации к коду

Приложение

Критерии и шкала оценки

Сроки для выполнения работы

Предоставление результатов работы

Плагат и несамостоятельное выполнение

Определение финалистов

Постановка задачи

Вы работаете в IT компании, которая занимается разработкой проектов по заказу. В качестве первого задания вы должны написать программу для предсказания прогноза погоды с использованием нейросети.

Информацию о погоде необходимо брать из официальных источников и использовать только реальные данные, а не сгенерированные другой нейросетью. Ссылку на источник укажите в ReadMe.txt в своем репозитории на GitHub.

Для создания данной нейросети необходимо использовать библиотеку TensorFlow и Keras.

Для обучения такой нейросети необходимо собрать статистику средней температуры по месяцам за последние 10 лет. Затем на основе этих данных необходимо обучить нейросеть для предсказания температуры на следующий год.

Напишите программу, оформите и предоставьте результат.

Оформление проекта

Решение необходимо загрузить на GitHub. Оно должно содержать проект приложения и собранный билд формата exe в отдельной папке build. Необходимо также наличие файла ReadMe.txt

Ссылка на репозиторий GitHub, содержащий ваш проект, отправляется в адрес Оргкомитета для оценки работы.

Требования, участвующие в оценке решения

Требования к способности системы

- Нейросеть должна получать на вход датасет из средних температур по месяцам за 2014 - 2024 года.
- Нейросеть должна выдавать массив из 12 чисел (средние температуры по месяцам за 2025 год).
- Должна быть возможность загружать входной массив из файла формата txt.
- Должна быть возможность сохранять полученный результат в файл формата txt.

- По выходному массиву программа должна строить диаграмму (ось x – месяц, ось y – температура).
- Обученная модель должна сохраняться при необходимости и загружаться (в процессе работы программы пользователь выбирает загружать ранее обученную модель или производить обучение на новых данных).
- Программа должна производить оценку точности модели на тестовых данных.
- Модель должна быть обучена, точность должна быть приемлема.
- Для оценки качества модели необходимо построить график потерь (ось x – эпоха, ось y – точность).

Требования к качеству кода

- Приложение должно функционировать в ОС Windows 10
- Рекомендуемая версия Python: 3.10+

Требования и рекомендации к коду


- Для сокращения срока разработки не нужно использовать сложные паттерны программирования.
- У классов, их полей и методов должны быть простые и понятные имена, отражающие их суть.
- Для упрощения разработки модули логирования разрабатывать не нужно.
- Рекомендуемые библиотеки: numpy, tensorflow, keras, matplotlib
- Использование нейросетей для генерации кода запрещено.

Приложение не обязательно должно быть консольным. Сложность и оригинальность исполнения пойдет в плюс.

Приложение

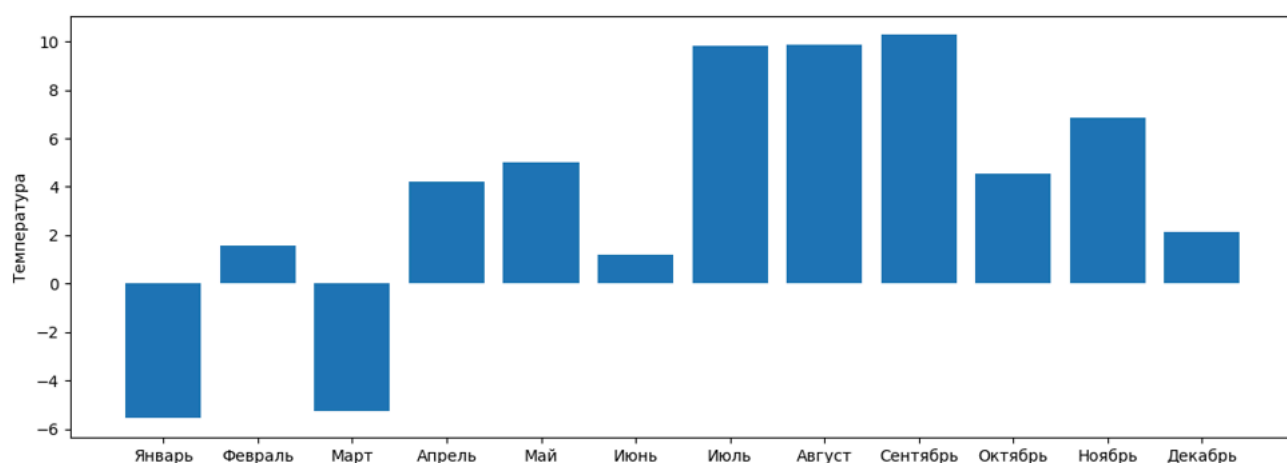
Примерный вид окна программы в ОС Windows

Epoch 150/150

1/1  0s 61ms/step - loss: 0.3628

Save model? (Enter Y to save): Y

Saved (model.weights.h5)



Критерии и шкала оценки

Конечное решение будет оцениваться по критериям, в каждом из которых можно набрать от 0 до 5 баллов. Конечная оценка решения может варьироваться от 0 до 25 баллов:

- Соответствие тематике и оригинальность решения – от 0 до 5 баллов.
- Техническая сложность и оптимальность решения – от 0 до 5 баллов.
- Соответствие требованиям к способностям системы – от 0 до 5 баллов.
У классов, их полей и методов, констант и переменных должны быть простые и понятные имена, отражающие их суть. На русском - наличие комментариев в коде.
- Стиль кода – от 0 до 5 баллов.
- Внимание к деталям и оформлению проекта – от 0 до 5 баллов.

0 баллов в критерии означает полное отсутствие данного элемента

1 балл – элемент в наличии, есть критические ошибки

2 балла – элемент реализован «удовлетворительно», возможно, есть некритические ошибки

3 балла – элемент реализован «хорошо»

4 балла – элемент реализован на «отлично»

5 баллов - получает лучшая(-ие) реализация(-ии) элемента по данному критерию среди всех участников.

В случае предоставления неполного комплекта документации, приведенного в разделе “оформление проекта”, работа не оценивается.

Сроки для выполнения работы

Результаты предоставляются до **19 марта** 2025 года включительно.

Предоставление результатов работы

Отправьте ссылку на репозиторий GitHub **до 19 марта включительно** на почту info@tbolimpiada.ru , указав в теме письма «Полуфинал_ТБ_Python_ФИО(участника)».

Проверьте, что **ссылка на папку дает право просмотра** читателям.

Плагиат и несамостоятельное выполнение

Работы, уличенные в плагиате, оцениваться не будут: участники, приславшие не оригинальные работы — из дальнейшего участия выбывают. Идентичные работы, неоригинальные работы, работы-копии исключаются из соревнований без выяснения причин и соотнесения времени получения.

Участие индивидуальное. Командное участие не допускается.

Выполняйте задание самостоятельно, не привлекая стороннюю помощь.

Оргкомитет оставляет за собой право связаться выборочно с любым из участников полуфинала перед подведением итогов, чтобы убедиться в самостоятельном выполнении задания.

Оргкомитет также оставляет за собой право собрать участников полуфинала в режиме видеоконференции для защиты присланных проектов.

Определение финалистов

Итоги подведем не позднее 26 марта. Результаты полуфинала опубликуем на сайте Олимпиады в виде рейтинговых таблиц. Финалистами станут 15 человек, показавшие лучшие результаты. При определении финалистов будут учтены условия допуска (приложение 4, [Положение](#) об олимпиаде, стр. 25).