ИУ5-83Б Алёшин, Балабас, Коноваликова, Ларионова

Правила:

1. Если нужно определить организацию озера данных, то нужно сравнить три типа хранения данных: реляционный тип, графовой, многомерный.

1. Если нужно оценить реляционный тип хранения данных, то нужно загрузить базу данных в спарк и в постгресс.
2. Если нужно оценить реляционный тип хранения данных в спарк, нужно провести оценку скорости выполнения требуемой команды.
3. Если нужно качественно оценить показатель скорости выполнения требуемой команды для реляционной базы в спарк, нужно составить запросы в трех сложностях: маленькой, средней и большой.
4. Если нужно оценить реляционный тип хранения данных в постресс, нужно провести оценку скорости выполнения требуемой команды.
5. Если нужно качественно оценить показатель скорости выполнения требуемой команды для реляционной базы в посгресс, нужно составить запросы в трех сложностях: маленькой, средней и большой.
6. Если нужно понять, в какой среде выгоднее хранить реляционную базу, нужно сравнить итоги по запросам в спарк и постгресс.

1. Если нужно оценить графовый тип хранения данных, то нужно загрузить базу данных в спарк и неофорджи.
2. Если нужно оценить графовый тип хранения данных в спарке ,нужно провести оценку скорости выполнения требуемой команды.
3. Если нужно качественно оценить показатель скорости выполнения требуемой команды для графовой базы в спарк, нужно составить запросы в трех сложностях: маленькой, средней и большой.
4. Если нужно оценить графовый тип хранения данных в неофорджи, нужно провести оценку скорости выполнения требуемой команды.
5. Если нужно качественно оценить показатель скорости выполнения требуемой команды для графовой базы в неофорджи, нужно составить запросы в трех сложностях: маленькой, средней и большой.
6. Если нужно понять, в какой среде выгоднее хранить графовую базу, нужно сравнить итоги по запросам в спарк и неофорджи.

1. Если нужно оценить многомерный тип хранения данных, то нужно загрузить базу данных в спарк и пентахо.
2. Если нужно оценить многомерный тип хранения данных в спарк, нужно провести оценку скорости выполнения требуемой команды.
3. Если нужно качественно оценить показатель скорости выполнения требуемой команды для многомерной базы в спарк, нужно составить запросы в трех сложностях: маленькой, средней и большой.
4. Если нужно оценить многомерный тип хранения данных в пентахо, нужно провести оценку скорости выполнения требуемой команды.
5. Если нужно качественно оценить показатель скорости выполнения требуемой команды для многомерной базы в пентахо, нужно составить запросы в трех сложностях: маленькой, средней и большой.
6. Если нужно понять, в какой среде выгоднее хранить многомерную базу, нужно сравнить итоги по запросам в спарк и пентахо.
7. Если нужно реализовать и использовать нейросеть с использованием аппаратной платформы Gyrfalcon, то нужно использовать специальное ПО: Gyrfalcon;
8. Если нужно реализовать и использовать нейросеть с использованием аппаратной платформы Khadas, то нужно использовать специальное ПО: Khadas;
9. Если нужно реализовать и использовать нейросеть с использованием аппаратной платформы Hikvision, то нужно использовать специальное ПО: Hikvision;
10. Если нужно реализовать и эффективно использовать нейросеть с использованием аппаратной платформы Nvidia, то нужно использовать ПО: TensorRT;
11. Если нужно реализовать и эффективно использовать нейросеть с использованием аппаратной платформы Intel, то нужно использовать ПО: OpenVino;
12. Если нужно реализовать и эффективно использовать нейросеть с использованием различных аппаратных платформ, то нужно использовать ПО: OpenCV, Tensorflow lite, Tensorflow, PyTorch, ONNX runtime, MNN, Tensorflow.js, ONNX.js.
13. Если нужно реализовать веб-приложение для работы с нейросетью, то нужно использовать ReactJS или Django;
14. Если нужно сделать инференс нейросети, то нужно экспортировать модель в формат .onnx;
15. Если нужно конвертировать модель нейросети, написанной на keras или tensorflow, то необходимо использовать пакет tf2onnx;
16. Если нужно передать в onnx модель нейросети входные данные и получить результат в Django, нужно использовать фреймворки ONNXruntime и ONNX;
17. Если нужно передать в onnx модель нейросети входные данные и получить результат в React.js, нужно использовать фреймворки ONNXruntime.js и ONNX.js;

Описания, по которым составлены правила:

1. Разработать структуру данных реляционной, графовой и многомерной модели данных Microsoft Academic Knowledge Graph.
2. Реализовать разработанную реляционную, графовую и многомерную структуру данных.
3. Загрузить данные в построенную реляционную, графовую и многомерную структуру.
4. Разработать запросы, с помощью которых будет тестироваться скорость обработки реляционных, графовых и многомерных данных.
5. Разработать модуль обработки реляционных, графовых и многомерных данных, который будет вызывать разработанные запросы и фиксировать время их выполнения.
6. Провести тестирование, в ходе которого будет получено время выполнения запросов на обработку реляционных, графовых и многомерных данных.

Сначала нужно выбрать фреймворк для реализации самой нейросети. В зависимости от того, с использованием чего она будет создаваться, будет зависеть выбор фреймворка. Если используется какое-то специальное аппаратное оборудование определённых производителей, то для достижение наибольшей эффективности и быстродействия лучше выбрать их ПО для разработки, так же и с крупными производителями. Но для лучшей универсализации используются OpenCV, Tensorflow lite, Tensorflow, PyTorch, ONNX runtime, MNN.

После выбора фреймворка для разработки и обучения нейронной сети, для последующей работы с моделью, лучше всего конвертировать её в формат .onnx. Поскольку не все фреймворки легко конвертируются в этот формат, то могут понадобиться дополнительные библиотеки, такие как tf2onnx и keras2onnx.

Для написания веб-приложений для нейронных сетей, лучше всего подойдут фреймворки Django и ReactJS.

После того, как была получена модель в .onnx формате и написано веб-приложение, необходимо получить возможность взаимодействовать с моделью, передавать в неё входные данные и получать выходные, для чего необходимы пакеты onnx(.js) и onnxruntime.