**Федеральное государственное автономное образовательное учреждение высшего образования**

**«Национальный исследовательский университет**

**«Высшая школа экономики»**

**Факультет компьютерных наук**

**Основная образовательная программа   
 «Прикладная математика и информатика»**

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

**Исследовательский проект на тему**

**Квантизация нейронных сетей в задачах обработки естественного языка**

**Выполнил студент группы 174, 4 курса,**

**Тропин Федор Владиславович**

**Руководитель ВКР:**

**Доцент, Ильвовский Дмитрий Алексеевич**

**Москва 2021**

Содержание

[1. Аннотация 3](#_Toc72174142)

[1. Annotation 3](#_Toc72174143)

[2. Введение 4](#_Toc72174144)

[3. Обзор литературы 5](#_Toc72174145)

[3.1. Квантизация 5](#_Toc72174146)

[3.2. Xlnet 5](#_Toc72174147)

[4. Datasets 6](#_Toc72174148)

[4.1. Glue 6](#_Toc72174149)

[4.2. Tokens classification 9](#_Toc72174150)

[4.3. IMDb 10](#_Toc72174151)

[5. Методы 10](#_Toc72174152)

[5.1. Huggingface 10](#_Toc72174153)

[5.2. О квантизации в целом 10](#_Toc72174154)

[5.3. Общий пайплайн обучения 12](#_Toc72174155)

[5.4. Qsin 13](#_Toc72174156)

[6. Описание реализованных классов и методов 13](#_Toc72174157)

[6.1. QuantedLayer 14](#_Toc72174158)

[6.2. Quantizer 14](#_Toc72174159)

[7. Способ применения 15](#_Toc72174160)

[8. Использованное оборудование 15](#_Toc72174161)

[9. Результаты 16](#_Toc72174162)

[9.1. CNN 16](#_Toc72174163)

[9.2. Glue 17](#_Toc72174164)

[9.3. Ner 18](#_Toc72174165)

[10. Выводы 19](#_Toc72174166)

[11. Список литературы 20](#_Toc72174167)

# Аннотация

Современные нейронные сети и методы машинного обучения имеют огромное количество параметров. Также на определенных устройствах, в частности смартфонах и планшетах, вычисления значений в типах данных с плавающей точкой производятся крайне медленно, что приводит к полной бесполезности моделей, которые должны работать в реальном времени. В данной работе мы сравниваем эффективность применения различных методов фейковой квантизации. По результатам проделанной работы мы предполагаем получить данные об эффективности примененных алгоритмов на разнообразных задачах и при различных начальных параметрах.

# Annotation

Modern neural networks and machine learning methods have a huge number of parameters. Also, on certain devices, in particular smartphones and tablets, calculations of values in floating-point data types are extremely slow, which leads to the complete uselessness of models that should work in real time. In this paper, we compare the effectiveness of various fake quantization methods. Based on the results of this work, we expect to obtain data on the effectiveness of the applied algorithms on various tasks and at different initial parameters.

# Введение

С каждым годом нейронные сети становятся все больше и тяжелее, количество слоев и параметров в них растет. Это приводит к замедлению расчетов и увеличению занимаемой весами памяти: как оперативной при вычислениях, так и постоянной при сохранении обученных моделей. Причиной этому являются расчеты, производимые при помощи чисел с плавающей точкой. Также с каждым годом растет как рынок смартфонов, так и приложений, разрабатываемых для них. С ростом качества и эффективности нейронных сетей их все чаще интегрируют для решения определенных задач прямо на устройства. При этом смартфоны имеют недостаток – на их процессорах затруднительно производить вычисления в вещественных числах. Обе проблемы возможно решить при помощи технологии приведения данных к целочисленному виду. Однако грубое округление чисел приводит к большим потерям точности. Существует специальный метод под названием квантизация. Данные сначала масштабируются, потом округляются и при выходе значений за пределы используемой битности округляются до минимумов или максимумов соответственно. При некоторых подходах так же возможно обучение самой квантизирующей части модели со своими функциями потерь. Целью нашей работы является определение эффективности применения алгоритма Qsin во время обучения квантизованных моделей на различных задачах. В ходе нашего исследования, перед нами были поставлены задачи изучения и реализация самих способов квантизации, применение полученных инструментов к модели, запуск и сравнение результатов при разнообразных начальных параметрах, а также при различных типах квантизации.

# Обзор литературы

## Квантизация

В работах [1] и [2] приведены результаты общих исследований различных типов квантизации и достижений в этих направлениях. В данных работах было отмечено, что применение квантизации положительно скажется на применении

нейронных сетей в мобильных устройствах. Так же во второй работе производится достаточно обширный анализ литературы по исследуемой нами теме. Если рассматривать применение квантизации конкретно к задачам NLP и трансоформерам в частности, можно обратить внимание на работы [3] и [4]. В первой работе исследуется эффективность квантизации модели BERT, а также применение различной силы сжатия разных слоев. Во второй же статье приводят результаты экспериментов на нескольких представителях трансформеров в 8 бит. Экспериментов, связанных с моделью Xlnet, на данный момент обнаружить не удалось, в связи с чем было принять решение изучить ее поведение.

## Xlnet

В качестве основной модели для исследования влияния квантизации на качество результатов была взята авторегрессионная модель Xlnet [5]. Данная модель является трансформером. Доступна в двух вариантах: базовая версия состоит из 12 входных слоев, 768 скрытых и 12 выходных, в то время как большая модель содержит 24 входных, 1024 скрытых и 16 выходных слоев. Обе версии могут работать с текстами, содержащими заглавные буквы. Версий, для которых текст предварительно нужно приводить к нижнему регистру нет. В ходе исследований было принято решение использовать только обычную версию модели в силу увеличения времени и объема потребляемой памяти при использовании большой версии модели xlnet.

Мы планируем использовать базовую модель для сравнения процентного роста или уменьшения интересующих нас величин. Была выбрана именно эта модель в силу ее достаточно сложного внутреннего устройства, большого количества параметров, а также разнообразия решаемых ею задач. Согласно официальной статистике XLNet показывает результаты лучше, чем BERT [6] в 20 задачах, а добивается эффективности уровня state-of-the-art в 18 задачах. Это позволит получить обширный набор данных для сравнения, продемонстрирует силу сжатия весов и ускорения вычислений, потенциально может помочь реализовать на мобильных устройствах большое количество задач, связанных с обработкой текста и его анализом.

Модель была протестирована на следующим проблемах: классификация текстов и классификации токенов. В качестве используемых тестов были взяты бенчмарк glue [7], как уже описанный в статье об xlnet и с приведенными результатами. Также в силу довольно большой скорости обучения для проверки качества модели была выбрана задача на распознавание именованных сущностей с датасетом CONLL 2003 [8].

Также отдельно была протестирована сверточная сеть на задаче классификации отзывов на датасете IMDb [9]. Данные тесты проводились во время первых тренировок и отладки процесса квантизации на относительно простой архитектуре сети с малым объемом потребляемой видеопамяти и высокой скоростью обучения.

# Datasets

## Glue

Glue это бенчмарк, ориентированный на общую оценку понимания естественного языка, включающий в себе 9 датасетов. Xlnet тестировался на 8 из них, из-за этого модель не имеет суммарной оценки и в лидерборде находится на 59 позиции. Неисследованным датасетом является QNLI

Мы смогли повторить эксперименты только на 5: RTE, SST-2, MRPC, CoLA, STS-B. Остальные датасеты: MNLI, QNLI, QQP обучались чрезвычайно долго.

Почти все используемые датасеты являются задачами бинарной классификации. Из данных предоставляются тестовые и валидационные разметки с правильными ответами, а также тестовые данные без ответов, валидирующиеся на стороне бенчмарка.

RTE – задача на распознавание текстового содержания. Датасет является объединением данных с нескольких соревнований, проводившихся с 2006 года. Данные разбиты в следующем отношении: 3000 обучающих примеров, 277 валидационных и 2490 тестовых примеров с ответами на стороне сервера. На вход подаются два предложения и необходимо определить, является ли второе предложение логическим следствием из первого предложения. Примеры данных приведены в Таблица 4.1. На данной задаче Xlnet занимает 15 место по доступной таблице с качеством модели в 88.5.

Таблица 4.1. Пример данных датасета RTE

|  |  |  |
| --- | --- | --- |
| label | sentence 1 | sentence 2 |
| 1 (not\_entailment) | After years of study, the Vatican's doctrinal congregation has sent church leaders a confidential document concluding that "sex-change" procedures do not change a person's gender in the eyes of the church. | Sex-change operations become more common. |
| 0 (entailment) | The fifth World Social Forum (WSF) has kicked off in Porto Alegre, Rio Grande do Sul state, Brazil. | The WSF takes place in Brazil. |

CoLA и SST-2 – задачи классифицирующие одно предложение. В CoLA необходимо определить, является ли предложение лингвистически приемлемыми, а в SST-2 предсказывается настроение отзыва о фильме – является ли он позитивным или негативным. На обоих датасетах Xlnet достигает 11 позиции согласно эталонному результату. Данные в таблицах распределены в таком соотношении: на обучения выделяется 8551 и 67349, валидацию - 1043 и 872, а тестирование проводится на 1043 и 1821 экземпляре соответственно. Элементы обоих классов для иллюстрации содержания датасетов приведены в Таблица 4.2 и Таблица 4.3.

Таблица 4.2. Пример данных датасета CoLA

|  |  |
| --- | --- |
| label | sentence |
| 0 (unacceptable) | The inspector analyzed the soundness in the building. |
| 1 (acceptable) | Bill will put a picture of her on your desk before tomorrow. |

Таблица 4.3. Пример данных датасета SST-2

|  |  |
| --- | --- |
| label | sentence |
| 0 (negative) | like a giant commercial for universal studios , where much of the action takes place |
| 1 (positive) | like the english patient and the unbearable lightness of being , the hours is one of those reputedly `` unfilmable '' novels that has bucked the odds to emerge as an exquisite motion picture in its own right . |

STS-B – единственная задача с не бинарной формой ответа, а вещественным числом в диапазоне от 1 до 5. Сама задача заключается в измерении семантической близости двух предложений, взятых из различных источников, таких как новости, подписи к видео и изображениями. Данные для этой проблемы приведены в Таблица 4.4 и разделены на 5749, 1500 и 1379 как на тренировочные, валидационные и тестовые соответственно. Сама модель в данной задаче достигает позиции MRPC – задача так же изучающая семантическую близость предложений, извлеченных из новостных источников, но имеющая бинарный тип классификации предложения эквивалентны/неэквивалентны. Данные в данном датасете разбиты в пропорции 1725/3668/408 на тестовые/тренировочные/валидационные соответственно, а их экземпляры приведены в Таблица 4.5.

Таблица 4.4. Пример данных датасета STS-B

|  |  |  |
| --- | --- | --- |
| label | Sentence 1 | Sentence 2 |
| 2.75 | The lawyer representing Torch Concepts, Rich Marsden did not respond to repeated phone calls. | Officials with Torch and JetBlue did not respond to repeated phone calls from the Mercury News seeking comment. |
| 1.25 | A man is cutting a potato. | A woman is cutting a tomato. |
| 4.80 | The inspector analyzed the soundness in the building. | A little girl is sitting on Santa's lap |

Таблица 4.5. Пример данных датасета MRPC

|  |  |  |
| --- | --- | --- |
| label | Sentence 1 | Sentence 2 |
| 0 (not\_equivalent) | Less than 20 percent of Boise 's sales would come from making lumber and paper after the OfficeMax purchase is completed . | Less than 20 percent of Boise 's sales would come from making lumber and paper after the OfficeMax purchase is complete , assuming those businesses aren 't sold . |
| 1 (equivalent) | Those in their twenties who ejaculated more than five times a week were one-third less likely to develop aggressive prostate cancer later in life , they say . | Those who ejaculated more than five times a week were a third less likely to develop serious prostate cancer in later life . |
| 0 (not\_equivalent) | The family stopped for lunch at Freshwater Spit , where several children went to the water 's edge to play in the surf shortly after noon | Several children , including the 8-year-old , went down to the water 's edge to play in the surf . |

## Tokens classification

Для тестирования классификации токенов используется датасет CONLL 2003, содержащий в себе разметки трех типов – Ner, Pos и chunk. Мы применяли модель исключительно к распознаванию ner токенов, в чью разметку входит 4 тип различных сущностей, два префикса начала и продолжения сущности, а также токен обозначающий непричастность слова к какой-либо сущности. Сами сущности включают в себя персоны, организации, локации и другие.

## IMDb

Данные – тексты обзоров фильмов и их бинарная разметка на позитивные и негативные классы. Данные хранятся в единой таблице на 50000 записей и распределены одинаково по классам – по 25000 обзоров каждой метки. Сами обзоры являются параграфами из нескольких предложений с присутствующей пунктуацией, буквы встречаются как в нижнем, так и в верхнем регистре, а также в отзывах иногда присутствуют html теги.

# Методы

## Huggingface

В качестве основного инструмента тестирования был выбран huggingface. Этот фреймворк предоставляет необходимые модули: модуль datasets дает доступ к обширному набору датасетов и метрик, а в модуле transformers есть функционал необходимый для загрузки и сохранения предобученных моделей с необходимыми головами для различных задач. Помимо этого, в модуль transformers входят токенайзеры для моделей, а классы TrainingArguments и Trainer, берущие на себя часть обучения и валидации модели, сохранение логов, вывод промежуточных результатов и многое другое. Так же стоит отметить, что на сейте huggingface выложены подготовленные тетради для запуска большого количества задач, в которых одни и те же задачи можно запускать для любых моделей, включенных в библиотеку huggingface, при условии наличия необходимых голов у предобученной модели. Стоит отметить, что содержание и подход тетрадей не гарантирует оптимальный код для повтора эталонных результатов представленных в статьях сетей, а всего лишь дает удобный и гибкий инструмент для проверки различных моделей.

## О квантизации в целом

Глобально процесс квантизации заключается в следующих трех процедурах: сначала тензор скалируется, т.е. делится поэлементно на определенное число. Далее происходит округление содержащихся в нем значений и в качестве завершающей функции используется обрезание значений вылезающих за пределы допустимых границ. В этих двух операциях так же используется поэлементное действие. Во всей этой последовательности действий стоит выделить два важных параметра – значение используемой битности квантизации и величина, при помощи которой производится скалирование.

Битность задает количество байт, при помощи которой можно закодировать целое число. Исходя из данной величины производится финальное округление. Так же стоит отметить, что обычно битность задается сразу для всей модели, а значит и для всех слоев одинаковой. Зачастую эта константа равняется двойке, возведенной в определенную степень, но не больше, чем 2 в 8 степени.

Способ же расчета и применение делителя тензоров и скаляра является весьма основополагающей идеей и определяет сам тип квантизации, а также имеет центральную роль в этой работе. На данном этапе важно обозначить, что существует два отдельных тензора на каждом слое, которые подлежат квантизации: сами входные данные, которые будут обрабатываться, и взаимодействующие с ними веса предобученной модели. Для обеих этих частей статистика считается или ведется отдельно.

Существует несколько подходов к квантизации моделей машинного обучения, глобально делящиеся на статический тип квантизации и динамический. Динамическая квантизация использует крайние значения из набора поступивших данных, чаще всего – максимум модуля значений тензоров по батчу, в то время как статическая определенным образом сохраняет нужное значение заранее и постоянно хранит его в памяти. В свою очередь статистическая квантизация сама бывает двух разных типов и делится по своим подходам на необучаемые и с возможностью дообучения.

В данной работе мы исследуем дополнительную функцию потерь, разработанную специалистами из Huawei, Qsin. Данный алгоритм предоставляет функцию потерь, рассчитывающий отклонение значений в тензорах от максимально приближенных к целым числам после деления на скаляр.

В целом, как уже описывалось ранее, квантизация состоит из трех частей. Однако при обучении мы не можем работать напрямую с данными в целочисленном виде, поэтому мы используем fake-quantization – ложную квантизацию. То есть мы приводим данные к интовому виду на каждом этапе, но в последствии мы продолжаем работу в модели во флотовом виде. Это связано с тем, что на операционных системах, доступных для серверов и персональных вычислительных машин, нет необходимого программного обеспечения для проведения обучения и тестирования известных нам моделей в целочиленном виде. Решением данной проблемы является добавление четвертого шага в процессе квантизации – домножение полученных данных после на предобученный скаляр в конце. Это позволяет продолжать дообучать исследуемую модель на доступном оборудовании.

## Общий пайплайн обучения

Перед началом квантизации модель обучается до наилучшего результата. После этого слои подлежащие квантизации переключаются в соответствующий режим, задается битность квантования, Сама модель переводится в режим валидации. В случае динамической квантизации слои просто переключаются в соответствующий режим и модель может быть валидирована. Для статической же квантизации выставляется специальный режим, который после поступления данных на слой запоминает и сохраняет скаляр, обычно являющийся максимум модуля тензора. Далее у слоев снова меняется состояние и при квантовании без дообучения может быть валидированна. В случае же обучаемо статической квантизации модель переключается в режим обучение и далее дообучается.

## Qsin

При дообучении статической модели с кусином (Уравнение 1) появляется дополнительный лосс, в свою очередь отдельно складывающийся из кусина входного тензора и кусина весов слоя. Данные значения сохраняются во время обучения модели в отдельные переменные во время форварда слоев и после извлекаются с каждого отдельной функцией и суммируются с необходимыми коэффициентами. Сам кусин считается следующим способом: скалированный тензор разделяется на три части. В первой обрабатываются значения не меньше и не больше, чем заданные битностью границы, во второй соответственно те, что меньше минимума, а в третьей те, что больше максимума. Первая часть домножается на пи, от каждого значения берется квадрат синуса. Из элементов второй и третьей группы вычитают минимум и максимум соответственно, возводят полученные значения в квадрат и домножают их на квадрат пи. Далее все суммируется.

Уравнение 1. Qsin

# Описание реализованных классов и методов

Одной из основных направлений разработки являлось создание удобных классов для быстрого, легкого и интуитивно понятного расширения слоев и обновление их в модели, а так же сопутствующего функционала.

Было создано два основных класса: QuantedLayer и Quantizer. Первый используется как один из классов, от которого наследуется слой с возможностью квантования и на данный момент устарел, делегировав по факту весь свой функционал второму. Внутри второго класса лежат все данные и методы, связанные с квантизацией внутри слоя. Весь вспомогательный функционал вынесен в отдельный файл

## QuantedLayer

В классе находится всего три используемых метода. set\_quantize просто передает квантайзеру полученные данные. Другие две функции возвращают Qsin от весов и от поданного тензора, они использовались во время написания и отладки на собственной моделе и в дальнейшем не использовались, а были заменены обновляемыми полями в квантайзере.

## Quantizer

Основной класс, включающий в себя почти все необходимые данные для квантизации, а также методы, проводящие саму квантизуацию. Начнем с принципа работы и идеи использования.

Данный класс добавляется в инициацию слоя, который будет квантизоватаься. После этого в слое переписывается метод форвард. Он остается практически неизменным относительно наследуемого класса, единственным отличием является пропуск входного вектора и весов слоя через форвард самого квантайзера.

Сам форвард квантайзера устроен следующим образом: если в квантайзере выключена квантизация, то он просто возвращает полученные входной тензор и веса. Если квантизация включена, то в зависимости от одного из трех состояний квантизации происходит следующее: при динамическом типе квантизации просто возвращаются тензоры, отскалированные по максимальному по модулю значению в изначально поступившем тензоре. При статической квантизации происходит тот же процесс, однако в функцию квантизации тензора дополнительно передается заранее подготовленный скаляр.

Если происходит сбор статистики, то сами тензоры не меняются, однако они отправляются в отдельную функцию расчета и накопления статистики. Накопление статистики происходит следующим образом: берется максимум модуля из значений поданного батча, добавляется в массив значений, полученных до этого, после чего обновляются скаляры слоя, присваивая им среднее значение из всех накопленных максимумов.

Так же, при условии, что используется Qsin, дополнительно в поля класса сохраняются последние полученные результаты значений кусина от поступивших данных.

# Способ применения

Помимо переписывания самих слоев, рекомендуется так же написать дополнительно несколько функций для облегчения дальнейшей работы. Из-за вариативности внутреннего устройства моделей и иногда в нестандартных подходах к исследованию квантизации, например, при квантизации весов в 4 бита, а входных данных в 8, автоматизировать или обобщить подход не удалось.

Из дополнительных функций необходимо написать функцию копирования данных из уже предобученных слоев в новосозданные, а также функцию извлечения результатов кусина для весов и активаций слоев.

# Использованное оборудование

В силу того, что на персональном компьютере для обучения модели на большинстве задач не хватало видеопамяти, пайплайн обучения со всеми сопутствующими данными был перенесен на Суперкомпьютер "cHARISMa" НИУ ВШЭ. Для разработки и тестирования наших работ в дальнейшем нам был представлен доступ к видеокартам NVIDIA Tesla v100 с 32 гб видеопамяти. Сами задачи запускались на любом доступном сервере со свободными 4 ядрами процессора Intel Xeon Gold и на одной видеокарте*.*

# Результаты

В целом итоговые результаты модели с квантизацией и Xlnet довольно близки, а иногда и выше качества флотовой модели, однако не больше, чем в пределах погрешности. Если сравнивать получаемое качество при разных типах квантизации xlnet в целом, то можно выделить довольно хорошую работу при любом типе квантизации в 8 бит. Почти на всех задачах модель не только не теряет качества во время квантизации, но при дообучении ей удается результат этот улучшить. Так же стоит отметить, что использование Qsin сильно увеличивает как потребляемое количество видеопамяти, так и время обучения. Почти на всех задачах обучение замедлялось примерно в 10 раз. Весь код и результаты выложены на github[[1]](#footnote-1).

Далее в таблицах зеленым цветом выделены ячейки с лучшим результатом внутри квантизаций с одной битностью.

## CNN

Данная модель использовалась для предварительного тестирования и отладки методов квантизации до начала экспериментов на более тяжелой модели, что так же в дальнейшем повлекло переписывание определенных классов. Сама модель состоит из последовательно соединенных слоя эмбеддингов, трех слоев двухмерной свертки различного размера, переходящих в слои max pooling 1d. Далее после склейки полученных результатов данные пропускают через обычный полносвязный слой и возвращается полученное предсказание. Во время обучения и тестирования на датасете IMDb разбиение данных производилось в процентном соотношении как 70 к 30 соответственно. Модель демонстрирует следующие результаты, приведенные в Таблица 9.1.

Таблица 9.1 – результаты тестов CNN на IMDb

|  |  |
| --- | --- |
|  | NER (F1) |
| CNN - Base | 80.28 |
| Dynamic - 4 bit | 78.2 |
| Static - 4 bit | 77.87 |
| QT - 4 bit | 79.03 |
| Qsin – 4 bit | 80.43 |
| Dynamic - 2 bit | 55.93 |
| Static - 2 bit | 49.73 |
| QT - 2 bit | 68.16 |
| Qsin – 2 bit | 69.08 |

## Glue

В данном наборе датасетов на большинстве задач модель при квантизации в 8 бит удается достичь точности фортовых моделей, за исключением SST-2 и MRPC. Однако в последней задаче у модели квантованной и дообученной с функцией потерь Qsin метрика F1 достаточно сильно превосходит результаты остальных моделей. Так же практически во всех задачах при применении квантизации явно выделяется преимущество дообучаемых квантованных моделей перед недообучаемыми. Так же интересным моментом может быть то, что при 4-х битной квантизации в большинстве задач превосходит квантование с дообучением и применением Qsin. Причины данного нетривиального поведения функции точно не были установлены и подлежат дальнейшему изучению. Полный список результатов можно увидеть в Таблица 9.2.

Отдельно стоит отметить датасет CoLA по нескольким причинам. Во-первых, это единственный датасет, на котором отклонение результатов от предоставленных в статье Xlnet отличается на порядок. Так же квантизация без дообучения полностью зануляет результаты модели. Причина данной проблемы исследована не была, есть вероятность, что данная задача требует особой настройки параметров запуска дообучения и прочих тонкостей подбора настроек для запуска. Так же отмечу, что процесс обучения всегда начинается с нуля, и это единственный датасет, при котором для достижения стабильного качества пришлось применить количество циклов обучения больше, чем 4. Для достижения стабильного результата пришлось потратить около 20 итераций в среднем, в то время как другие задачи сходились к стабильному результату примерно к третьему циклу.

Таблица 9.2 – результаты тестов на glue

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | RTE | SST-2 | MRPC (acc) | MRPC (F1) | CoLA | STS-B (pear) | STS-B (spea) |
| XLNet-Base | 69.3 | 93.1 | 89.2 | 92.3 | 48.4 | 87.6 | 87.2 |
| Dynamic - 8 bit | 65.3 | 92.8 | 87.7 | 91.5 | 46.8 | 87.4 | 87.3 |
| Static - 8 bit | 67.1 | 92.5 | 86.7 | 90.8 | 47.3 | 87.4 | 87.3 |
| QT – 8 bit | 70.7 | 93 | 88.4 | 91.8 | 50.2 | 88.1 | 87.7 |
| Qsin – 8 bit | 71.8 | 90.7 | 88.7 | 97.8 | 46.1 | 87.8 | 87.4 |
| Dynamic - 4 bit | 48.7 | 77 | 58 | 63.5 | 0 | 27.8 | 28.8 |
| Static - 4 bit | 49.8 | 78.4 | 59.8 | 65.5 | 0 | 32 | 32.7 |
| QT - 4 bit | 58.4 | 89.4 | 81.3 | 87 | 19.7 | 60.3 | 60.5 |
| Qsin – 4 bit | 56.3 | 90 | 80.8 | 86.2 | 21.6 | 70.5 | 73.3 |

## Ner

Данная задача, как и класс задач по классификации токенов в целом для модели Xlnet в оффициальной статье приведены не были, однако эта задача была выбрана как одна из уже имеющих реализацию и при этом достаточно быстро обучающуюся. На данной задаче достичь качества флотовой модели не удалось, однако и сильного понижения точности не последовало. Отдельно стоит обратить внимание на результаты квантизованной модели в 4 бита. Без дообучения качестово не сдвинулось от оценки в 0, однако при дообучении модели качество резко растет и даже приближается к результатам квантованных моделей в 8 бит, хотя и не добирая до них явную пару процентов. Итоги тестирования представлены в Таблица 9.3.

Таблица 9.3 – результаты тестов Xlnet на Ner

|  |  |
| --- | --- |
|  | NER (F1) |
| XLNet-Base | 94.5 |
| Dynamic - 8 bit | 93.1 |
| Static - 8 bit | 92.9 |
| QT - 8 bit | 95.3 |
| Qsin -8bit | 92.3 |
| Dynamic - 4 bit | 0 |
| Static - 4 bit | 0 |
| QT - 4 bit | 90.3 |
| Qsin -4bit | 90.7 |

# Выводы

Подводя итоги, хочется отметить, что xlnet достаточно хорошо работает после квантизации, практически всегда даже повышая свое качество при дообучаемых типах квантизации на несколько процентов, однако тип квантизации нужно подбирать индивидуально под задачу. Отдельно стоит отметить квантизацию с дополнительной функцией потерь Qsin. Для данной функции по наблюдениям оптимально считать отдельно функции потерь для весов и входов слоя в силу их сильного отличия по результатам суммирования. Так, например, во многих задачах сумма функции от входных векторов достегает порядка , а иногда и превосходит несколько миллионов. В это же время для весов результат меньше в десятки или даже в сотни раз. Из-за этого для работы функции необходимо тщательно подбирать коэффициенты для домножения полученных сумм, так как они изменяются довольно нелинейно. При применении слишком маленького делителя, Qsin явно начинает занижать качество модели. Так же было замечено, что во время начала обучения с применением выше указанной функции, обычно первые несколько эпох предпочтительно обучать с дополнительно заниженным коэффициентом. То есть, если основное дообучение происходит с домножением на коэффициент, превращающий изначально существовавшие значения в десятые, то в первые две эпохи коэффициенты лучше дополнительно занизить на порядок. Так же на бенчмарке glue видно, что Qsin зачастую показывает более хорошее качество работы на более низких битностях, чем обычная дообучаемая статическая квантизация. Предполагается что это обоснованно недостаточно эффективным подбором коэффициентов для домножения. При низкой битности квантизации подходы без дообучения в некоторых задачах, таких как распознавание сущностей, а также при распознавании классов предложений на датасете cola показывают зануление качества модели.

В качестве дальнейшего потенциального развития проекта рассматривается добавление функции оценки сжатия данных после квантизации, возможно добавлении функции автоматически подбирающей битность квантизации для конкретного слоя. Так же планируется продолжить эксперименты по дообучению Qsin, исследовать скорость сходимости и изучать способы подборов коэффициентов.

# Список литературы

1. Guo Y. A survey on methods and theories of quantized neural networks //arXiv preprint arXiv:1808.04752. – 2018.
2. Gholami A. et al. A Survey of Quantization Methods for Efficient Neural Network Inference //arXiv preprint arXiv:2103.13630. – 2021.
3. Shen S. et al. Q-bert: Hessian based ultra low precision quantization of bert //Proceedings of the AAAI Conference on Artificial Intelligence. – 2020. – Т. 34. – №. 05. – С. 8815-8821.
4. Bhandare A. et al. Efficient 8-bit quantization of transformer neural machine language translation model //arXiv preprint arXiv:1906.00532. – 2019.
5. Yang Z. et al. Xlnet: Generalized autoregressive pretraining for language understanding //arXiv preprint arXiv:1906.08237. – 2019.
6. Devlin J. et al. Bert: Pre-training of deep bidirectional transformers for language understanding //arXiv preprint arXiv:1810.04805. – 2018.
7. Wang A. et al. GLUE: A multi-task benchmark and analysis platform for natural language understanding //arXiv preprint arXiv:1804.07461. – 2018.
8. Sang E. F., De Meulder F. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition //arXiv preprint cs/0306050. – 2003.
9. Maas A. et al. Learning word vectors for sentiment analysis //Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies. – 2011. – С. 142-150.

1. https://github.com/Ph4nt0mm/Final\_grade\_project [↑](#footnote-ref-1)