ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
ФГАОУ ВО НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет компьютерных наук
Образовательная программа «Прикладная математика и информатика»

УДК 004.8

**Отчет об исследовательском проекте на тему:**

# Методы обучения с учителем для классификации последовательностей событий

**Выполнил студент:**

группы БПМИ211, 3 курса Кадейшвили Полина Алексеевна

**Принял руководитель проекта:**

Савченко Андрей Владимирович
Управляющий директор по исследованию данных
ПАО Сбербанк

Москва 2024

# Содержание

# Аннотация

Классификация последовательностей является важной областью для исследований во многих направлений нашей жизни. Так, человек, совершая любую цепочку действий, создает последовательность, которая может многое о нем рассказать. Например, что он любит больше всего покупать в супермаркете, сколько ему лет, его семейное положение. Большая часть данных в мире хранится в виде таблиц, потому что это самый удобный способ получить информация об объекте по его каким-то отличительными характеристикам. А во многих табличных данных можно найти структуру последовательности, поэтому задача классификации последовательностей является весьма актуальной проблемой. Есть много разных варинатов того, как решать поставленную задачу. В Sber AI Lab разработали метод, который позволяет эффективно проводить классификацию последовательностей путем контрастного обучения. Плюсом такого метода является то, что не у всех данных должны быть истинные метки классов, а лишь у маленького процента. Эта работа нацелена на улучшения качества классификации последовательностей, показываемых придуманными в лаборатории методами, в задачах обучения с учителем в условиях ограниченного количества табличных данных.

# Ключевые слова

Глубинное обучение, машинное обучение, обучение с учителем, качество классификации, последовательность транзакций, бустинг, эмббединги, контрастное обучение, аугментация данных.

# 1 Введение

## 1.1 Описание предметной области

Машинное обучение уже давно используется для решения множетсва задач в нашей жизни. Оно показывает хорошие результаты в задачах классификации, значительно упрощая жизнь людям. Так, с помощью машинного обучения можно классфицировать табличные данные, с помощью глубинного обучения - тексты, картинки, аудио дорожки. Но если рассматривать последовательности, то это сложная структура, которая одновременно может содержать и зависимые наблюдения, и независимые (если говорить о покупках, то это могут быть стандартные продукты, которые человек потребляет каждый день, и покупка машины, которую человек может менять раз в 10 лет). Поэтому для классификации цепочек событий не подойдут стандартные методы из машинного обучения. Часто для решения этой задачи используются подходы из сферы обработки естественного языка (NLP) и работы с картинками (CV). Существует много вариантов, как проводить классификацию при работе с картинками или текстом. Так, для текстовых данных с этой задачей хорошо справляется BERT [13]. Для картинок существует много архитектур, которые хорошо справляются с задачей классификации, например, ResNet [9]. Но такие методы классификации в основном основываются на предположении, что соседние пиксели будут похожи друг на друга, или что контекст соседних слов будет близок. Однако для последовательностей, записанных в таблицы, такое утверждение не является верным. Так, люди, которых модель может считать соседями, а значит рассчитывать, что их покупки будут похожими, могут совершать импульсивные действия, которые будут непоследовательными и нерегулярными. Так же выглядят мошеннические транзакции, которые могут происходить не часто и выглядеть как очень нестандратное поведение клиента. В данный момент не существует гарантий, что модели из NLP и CV смогут отличить мошеннические транзакции от обычных транзакций клиента, которые отличаются от большинства предыдущих платежей. Более того, на практике данных обычно бывает слишком мало, чтобы использовать большие модели, требующие большого количества входных данных, из-за чего подходы из NLP и CV могут показывать не лучшие результаты.

## 1.2 Постановка задачи

В Sber AI Lab был придуман новый алгоритм классификации последовательностей CoLES (Contrastive Learning for Event Sequences) [2]. Он основан на контрастном обучении

(специальном способе аугментации данных), то есть все объекты разбиваются на 2 категории: позитивные и негативные пары. Позитивные пары - это семантически близкие объекты. Негативные пары - это объекты разных классов, то есть достаточно непохожие объекты. Целью такого контрастного обучения является расположить объект в новом эмбеддинговом пространстве близко к его положительным парам и далеко от его негативных пар. CoLES позволяет адаптировать контрастное обучение для дискретных цепочек событий и демонстрирует результаты лучше, чем supervised, self-supervised и semi-supervised алгоритмы.

Однако у метода CoLES есть значительный минус: при кластеризации модель никак не берет во внимание истинные значения целевой переменной. То есть во время создания позитивных и негативных пар алгоритм может уменьшать расстояние между объектами, которые на самом деле относятся к разным классам и, наоборот, увеличивать расстояние между объектами, которые принадлежат одному классу. Чтобы избежать этой проблемы, нам было необходимо исследовать другие supervised, semi-supervised и self-supervised методы, чтобы либо изменить модель CoLES так, чтобы она стала основываться на истинных метках классов при создании пар, либо придумать новые архитектуры, которые бы показывали сопоставимое с CoLES качество. Единственной большой проблемой стояло то, что неразмеченных данных в разы больше, чем размеченных, то есть на вход CoLES поступало в несколько раз больше входных данных, чем на вход любой модели, которая проводила классификацию с учителем. Поэтому было необходимо придумать такую архитектуру, которая могла бы с небольшим количеством данных давать хороший результат классификации, превосходящий имеющийся на тот момент результат работы модели обучения с учителем.

## 1.3 Актуальность задачи

Классификация последовательностей событий, особенно транзакций, является очень востребованной задачей, например, для банков. Так, банкам важно понимать, сможет ли человек вернуть кредит, основываясь на его тратах, стоит ли ему предлагать какие-то акции. Зная пол, возраст и предыдущие транзакций, можно значительно улучшить систему рекомендаций услуг, что в дальнейшем может привести к росту прибыли. Более того, все банки страдают от мошенников, которые обманывают их клиентов. Если мошенники оформят кредит в финансовой организации на человека, который не сможет его выплатить, то пострадает в том числе банк. Поэтому проблема выявления мошеннических транзакций является актуальной и требует подробных исследований. Хорошо работающий способ классификации последовательностей событий позволит банкам избежать много проблем, таких как невы-

плаченный кредит, и привлечь новых клиентов благодаря хорошим персонализированным рекомендациям.

# 2 Обзор литературы

## 2.1 Gradient boosted decision trees (GBDT)

Существует много разных подходов машинного обучения, которые хорошо работают с табличными данными, однако не всегда такие методы подходят для классификации последовательностей.Основным подходом для классификации табличных данных является градиентный бустинг над решающими деревьями (XGBoost) [19].Однако, этот метод нельзя напрямую применять к последовательностям, потому что XGBoost никак не учитывает структуру данных, а именно, что какие-то транзакции могут зависеть от предыдущих. Поэтому, прежде чем применять GBM к данным для классификации, их нужно предобработать.

## 2.2 TabNet

В работе Sercan O.Arık,TomasPfiste [17] представлена архитектура трансформера, которая способна работать с табличными данными. Как у всех трансформеров, у этой модели есть encoder и decoder слои, такая сеть способна работать как в случаях, когда метки классов есть, так и когда они отсутствуют. В экспериментах, проведенных авторами статьи, TabNet показывал результаты лучше XGBoost [19], LightGBM [14], CatBoost [5] и Auto ML. Однако эта архитектура тоже не приспособлена к работе с последовательностями событий, поэтому данные необходимо сначала обработать, чтобы TabNet показал какой-то результат.

## 2.3 TabR

В статье Yury Gorishniy, Ivan Rubachev, Nikolay Kartashev [23] был предложен новый метод классификации табличных данных, который использовал генерацию ответа пользователю с учетом дополнительно найденной релевантной информации (retrieval-augmented generation). Этот способ предсказания объединяет в себе преимущества трансформеров, подходов из обработки естественного языка и удобство табличных данных. В экспериментах, проведенных авторами статьи, результаты работы TabR превзошли результаты различных видов бустинга: XGBoost [19], LightGBM [14], CatBoost [5]. Но с таким подходом возникает проблема, что методы из NLP основываются на контексте соседних событий, что для тран-

закций может быть неверно.

## 2.4 Обучение табличных данных с помощью перевода в текстовый вид

В работе Keshav Ramani, Daniel Borrajo [15] описан подход, который позволяет перенести методы классификации текстов для классификации табличных данных. Для этого все данные в таблице сначала кодируются соответсвующими словами, а затем из таких закодированных слов создается предложение с добавлением специального символа в его начало и конец. После этого для предсказания класса используется последовательная реккурентная сеть, слоями которой является LSTM (long short-term memory) [11]. Проблемой такого подхода тоже является то, что в последовательностях не всегда нужно смотреть на соседние транзакции, чтобы сделать правильную классификацию.

## 2.5 BERT

В статье Jacob Devlin, Ming-Wei Chang, Kenton Lee и Kristina Toutanova [13] описана архитектура сети, позволяющая угадывать слова из контеста и производить классификацию текстов. Эта модель обходит большую часть бейзлайнов, поскольку обучена на огромных корпусах текстов. Поскольку текст - это тоже некая последовательность символов, такая архитектура могла бы подойти под решение нашей задачи, но, так как в тексте контекст обязательно согласовывается, а просто в последовательностях могут быть элементы, которые не очень согласуются с остальной последовательностью, такая архитектура не подойдет.

## 2.6 Revisiting Deep Learning Models for Tabular Data

В статье Юрия Горишного и Ивана Рубачева [7] описываются подходы к применению неройнных сетей к табличным данным. Обычно для табличных данных используются разные виды бустинга, но в этой работе описано, как можно добиться результатов лучше, чем у бустинга, используя нейронные сети. Эта работа показывает, что глубинное обучение можно успешно применять к табличным данным, но в ней не рассматривается применение к последовательностям, потому что у них более сложная структура.

# 3 Описание предлагаемого метода

Для решения задачи классификации последовательностей в Sber AI Lab разработали метод, который назвали CoLES [2]. Это алгоритм, который работает на основе контрастного обучения. То есть для каждого объекта подбираются позитивные и негативные пары. Позитивными парами являются данные, которые похожи на наш исходный объект, а негативными, наоборот, непохожие. Целью работы модели является уменьшить расстояние между позитивными парами и увеличить расстояние до негативных пар. Чтобы отправить данные, представленные в табличном виде в такую модель, нужно сначала предобработать их, то есть закодировать категориальные признаки, и сделать из таблицы список словарей, где в каждой ячейке списка будет лежать словарь, который соберет в себе всю информацию о конкретном пользователе. Ключами в таком словаре выступают признаки, а значениями - все числа из таблицы, относящиеся к этому пользователю. В итоге получается, что все последовательности представлены в одной структуре, но разной длины, поскольку люди могли совершить разное количество транзакций. Затем этот список словарей преобразовывается в батч с паддингом. После этого этапа данные уже подготовлены для обучения. В начале обучения данные попадают в рекуррентную нейронную сеть: LSTM [11] или GRU [4]. Это необходимо, потому что последовательности можно рассматривать как временной ряд, где часто следующая транзакция может зависеть от предыдущей. Обучение в рекуррентной нейронной сети позволяет из последовательности событий вычленить действия, которые повлияли на следующие транзакции. На выходе получаются логиты, которые передаются в модель CoLES, алгоритм которой описан выше. После того, как CoLES создала позитивные и негативные пары, мы получаем новые эмбеддинги. Уже после обучения к полученным эмбеддингам конкатинируются имя пользователя и истинная метка класса для этого человека. Затем, чтобы произвести уже классификацию, необходимо запустить градиентный бустинг, подав в качестве данных новые признаки от алгоритма CoLES. Такой подход значительно превосходит supervised, self-supervised и semi-supervised алгоритмы. Результаты можно увидеть в таблице 3.1.

В качестве данных для экспериментов было использовано 4 датасета: 'Age', 'Churn', 'Assess' и 'Retail'. Во всех них было необходимо предсказывать пол человека, возраст либо принадлежность к какой-то целевой группе. В качества метрики качества классификации использовалось accuracy или AUC-ROC. 'Unsupervised' в таблице означает результат работы CoLES + GBM. 'Downstream' означает, что сначала последовательности прошли через модель CoLES, а затем к полученным эмбеддингам были добавлены метки классов и они бы-

Таблица 3.1: Результаты классификации модели CoLES

| | Age | Churn | Assess | Retail |
|---|---|---|---|---|
| **Metric** | **Accuracy** | **AUROC** | **Accuracy** | **Accuracy** |
| **Unsupervised** | 0.638 | 0.843 | 0.601 | 0.539 |
| **Downstream** | 0.644 | 0.827 | 0.615 | 0.552 |

ли переданы как входные данные модели, отвечающей за классификацию, то есть обучение второй сети происходит с помощью дообучения эмбеддингов от CoLES.

# 4 Экспериментальное исследование

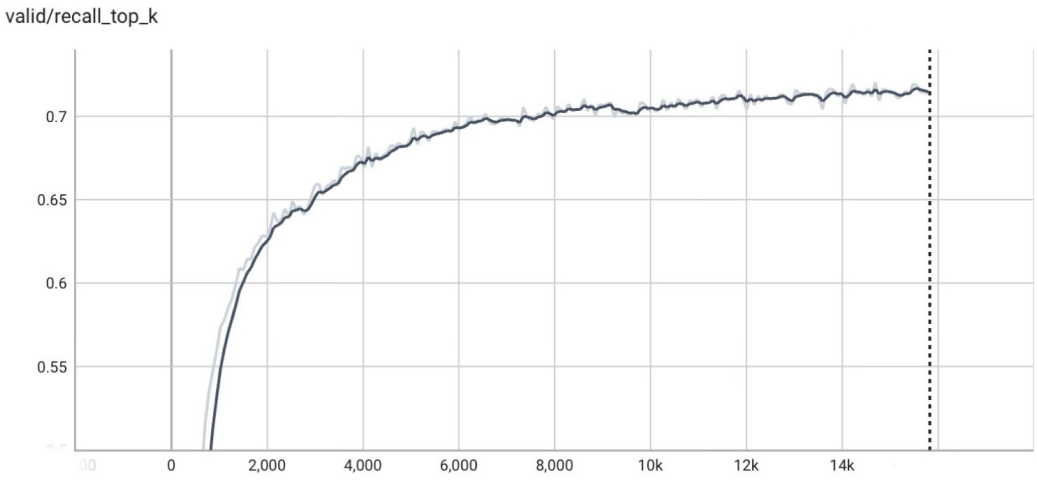
Задачей нашей работы было исследование различных способов улучшения итогового качества классификации. В начале работы было необходимо повторить эксперименты, приведенные в статье о CoLES и посмотреть, что заявленные значения метрик действительно достижимы.

## 4.1 Работа с моделью CoLES

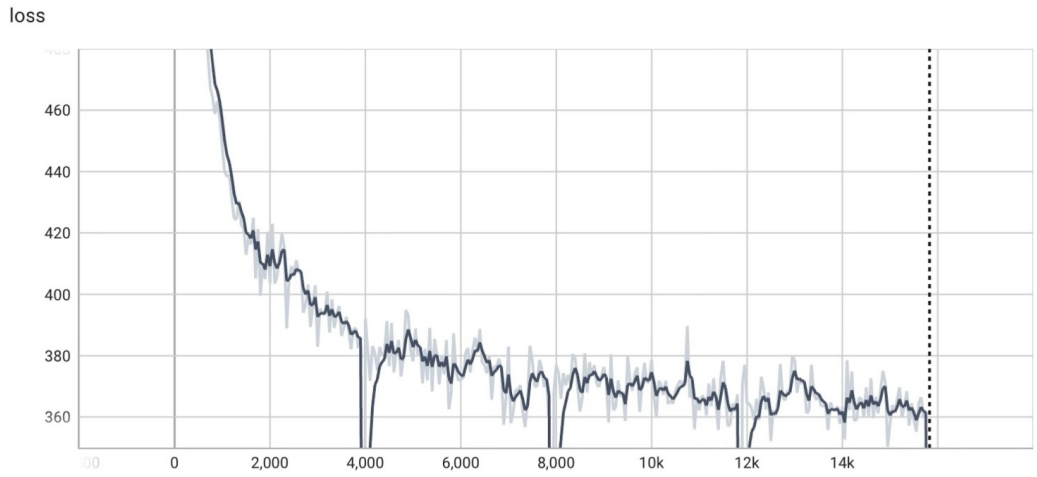
После того, как мы проверили, что результаты из статьи можно действительно получить, встал вопрос, можно ли как-то улучшить эту модель, чтобы повысить качество классификации. Результаты модели CoLES можно посмотреть на графиках 4.1. Модель CoLES + GBM дает accuracy равное 0.639.

В начале работы были исследованы известные способы классификации из машинного обучения поверх полученных из CoLES эмбеддингов. Были рассмотрены такие алгоритмы многоклассовой классификации как случайный лес, все против всех, один против всех, логистическая регрессия, разные варианты градиентного бустинга. Для всех этих методов были подобраны гиперпараметры, чтобы качество было лучшим из возможных. После того, как стало понятно, что лучший результат все равно дает бустинг, мы попробовали использовать нейронные сети, на вход которым подавались эмбеддинги, полученные после контрастного обучения. В качестве таких сетей были реализованы MLP, LSTM и Transformer. Поскольку размеченных данных в несколько раз меньше чем неразмеченных, то количество данных, которые превращались в эмбеддинги было очень мало, из-за чего качество классификации не стало лучше, чем у CoLES + GBM.

В исходном виде модель CoLES обучается на функцию потерь, описанную в статье Jane Bromley, Isabelle Guyon, Yann LeCun [3] для контрастного обучения. Для достижения




(a) График recall для модели CoLES на валидации

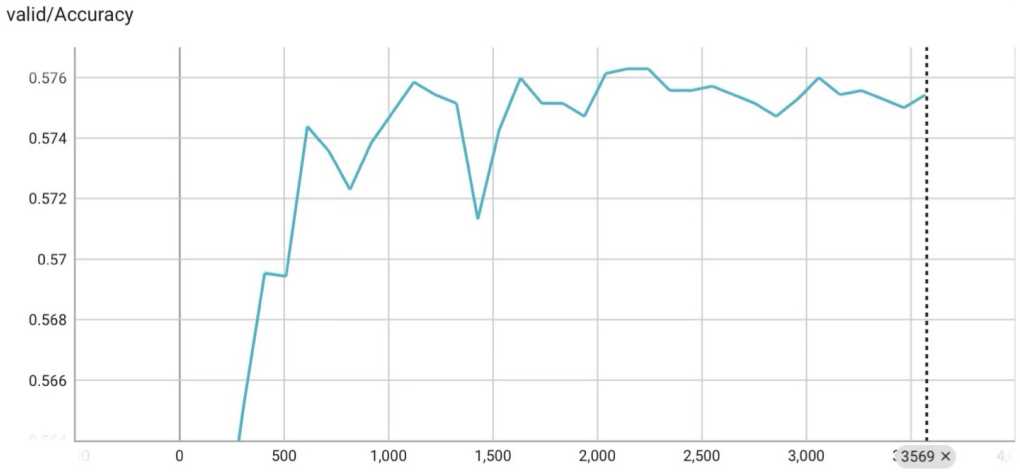



(b) График функции потерь для модели CoLES на валидации

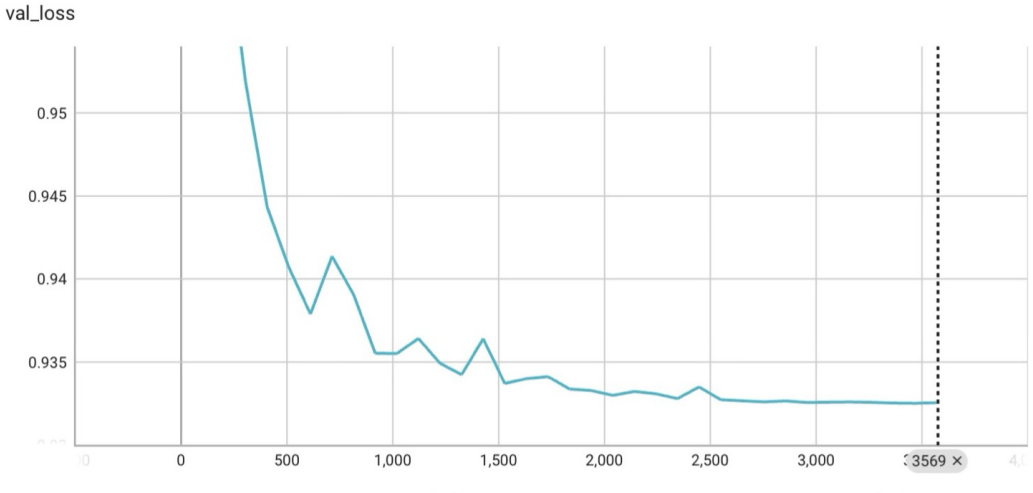
Рисунок 4.1: Результаты работы модели CoLES

лучших результатов было попробовано изменить функцию потерь, чтобы создавать позитивные и негативные пары несколько иначе. Так, были рассмотрены такие функции потерь, как Barlow Twin loss [24], Binomial Deviance loss [22], Centroid loss [21], Centroid Softmax loss, NCE loss [16], Triplets loss [6]. Для всех этих функций потерь также перебирались различные стратегии создания позитивных и негативых пар. Однако эти изменения несильно помогли улучшить качество предсказания.

В Sber AI Lab также работали над созданием End2End модели, которая могла бы хорошо проводить классификацию, ее назвали SequenceToTarget. Но результаты работы этой нейронной сети были значительно хуже, чем у CoLES. Поскольку бустинг, на вход которому подавались эмбеддинги, построенные CoLES, работал лучше остальных алгоритмов, мы решили попробовать то же самое, но вместо бустинга использовать более сложную нейрон-




(a) График accuracy для модели Seq2Target с входными эмбеддингами от CoLES на валидации




(b) График функции потерь для модели Seq2Target с входными эмбеддингами от CoLES на валидации

Рисунок 4.2: Результаты работы модели Seq2Target на основе эмбеддингов от CoLES

ную сеть, ею стала SequenceToTarget модель. Изначально качество работы этой End2End сети было 0.562, а при обучении на эмбеддингах, построенных после контрастного обучения, качество классификации End2End модели улучшилось 0.567. Графики такого процесса обучения можно посмотреть тут 4.2 Но этот результат все равно не превзошел результат работы CoLES + GBM.

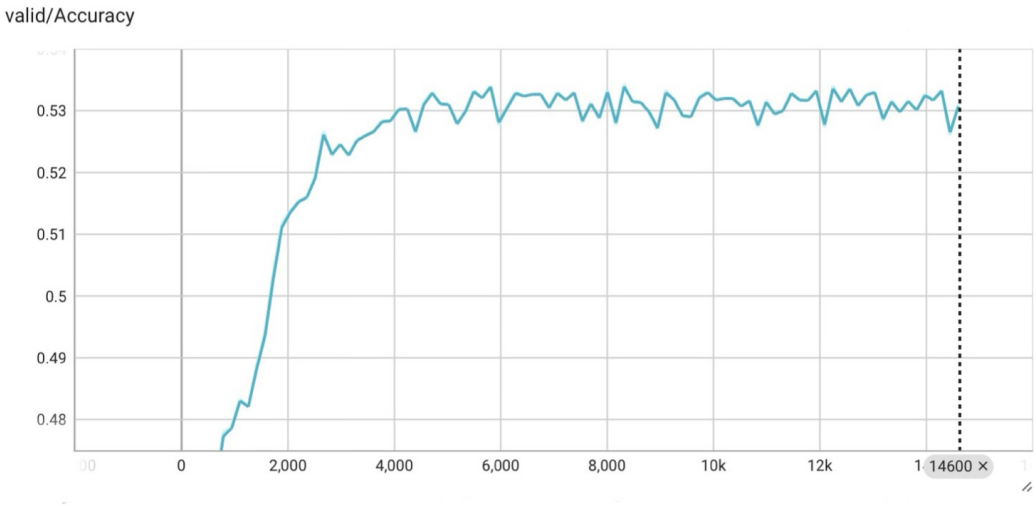
## 4.2 Улучшение End2End модели для классификации последовательностей

Вторым этапом исследования стал вопрос, можно ли как-то улучшить модель, которая работает только с размеченными данными и сразу же проводит классификацию без

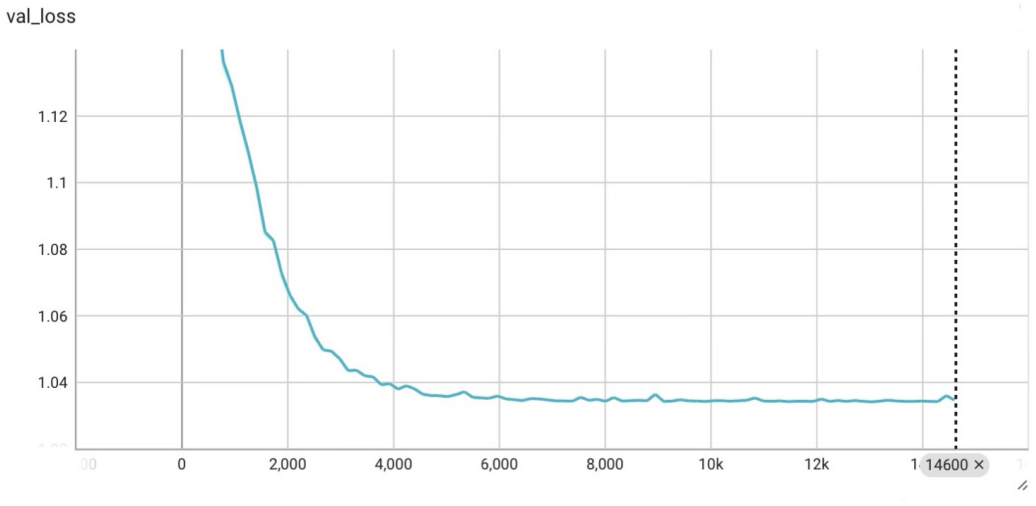
дополнительных алгоритмов, то есть, как улучшить SequenceToTarget.

В качестве такой нейронной сети стала модель, основанная на рекуррентной нейронной сети LSTM и на классифицирующей голове, которой может выступать как один линейный слой, так и более сложная архитектура.

В базовой версии такой сети используется LSTM и один линейный слой для классификации. Нашей задачей стояло поднять качество классификации такой сети, несильно меняя ее архитектуру. Результат работы базовой версии можно увидеть на рисунке 4.3.




(a) График accuracy для модели Seq2Target на валидации




(b) График функции потерь для модели Seq2Target на валидации

Рисунок 4.3: Результаты работы модели Seq2Target в базовом виде

Для этого, в первую очередь, было добавлено несколько вариантов нормализации в рекуррентную часть и в классифицирующую голову. Такое изменение позволило дольше учить нейронную сеть, избегая быстрого переобучения. В качестве регуляризации выступили Batch Normalization [12], L2 regularization, Dropout [18].

Затем было попробовано использовать вместо LSTM GRU и RNN, но такие изменения ничего особо не дали, потому что LSTM уже достаточно хорошо улавливала структуру последовательностей и выделяла основные элементы.

После этого стало понятно, что нужно менять голову, усложняя ее. Необходимо было выбрать такую архитектуру, чтобы она показывала большое качество в работе с табличными данными. В статье Revisiting Deep Learning Models for Tabular Data [7] как раз предлагаются такие методы.

В первую очередь, мы попробовали реализовать более сложную односвязную нейронную сеть. Проблема с исходной головой заключалась в том, что сеть достаточно быстро начинала переобучаться. Поэтому была выбрана архитектура сети SNN [8], которая использует функцию активации SELU, позволяющую обучать более глубокие сети. Однако, такие изменения не помогли улучшить качество, и оно стало равным 0.542. Результаты работы SNN, отраженные на графиках 4.4.

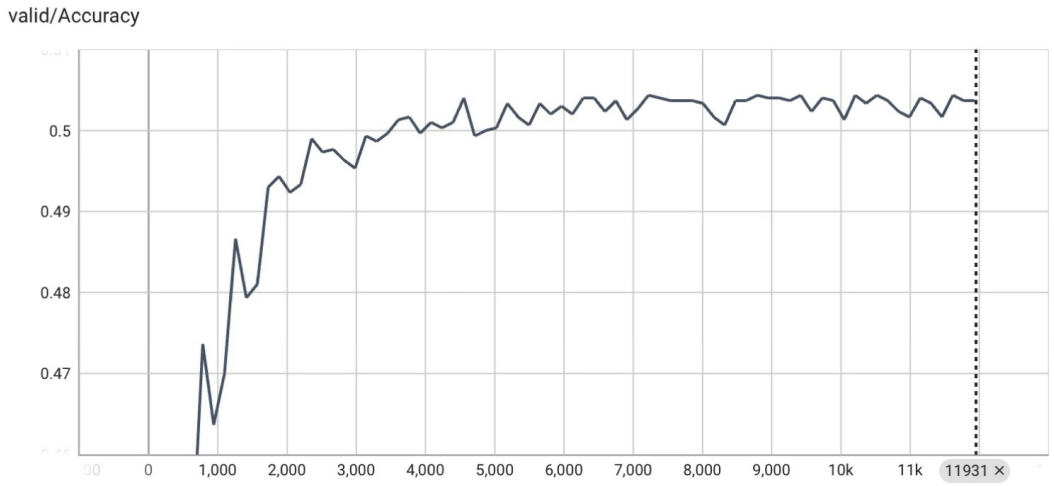
Затем мы реализовали ResNet [9], но для табличных данных. В отличие от нейронной сети для картинок, тут нет сверточных слоев, только линейные слои, функции активации, различные способы регуляризации. Так же как и у ResNet, у нашего табличного аналога есть блоки. Такая сеть сильно улучшила качество классификации, и accuracy стало равно 0.597. На рисунке 4.5 представлены результаты работы ResNet.

После того, как более сложная архитектура позволила улучшить результат, мы попробовали усложнить ее еще сильнее, а именно: взять трансформер [20] FT-Transformer. Первым шагом работы этой сети выступает преобразование входных числовых и категориальных признаков в токены. Для этого создается специальный модуль Tokenizer. Так как в нашем случае FT-Transformer это всего лишь голова, то на вход ей подаются логиты, которые являются выходом рекуррентной нейронной сети.

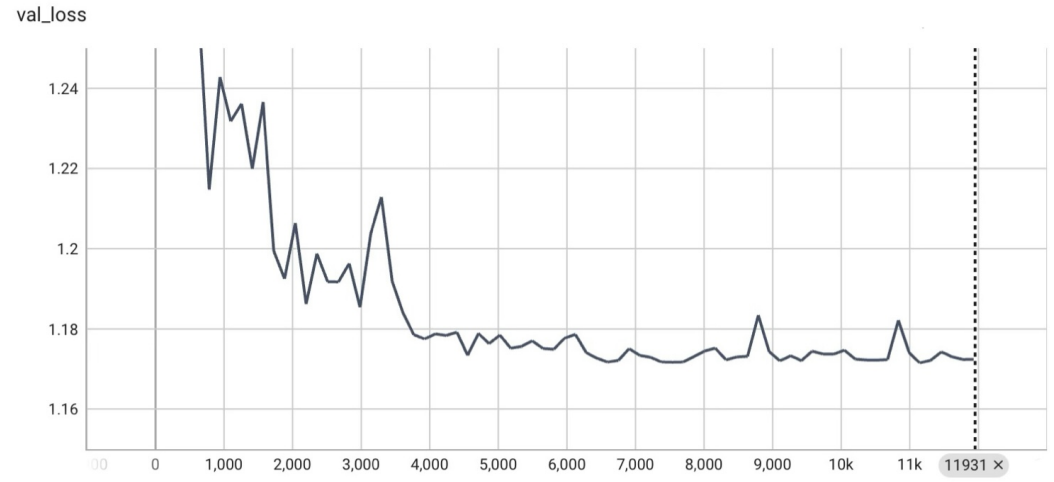
Затем, как в любом трансформере, идет блок MultiheadAttention. Для извлечения зависимостей между токенами используется модуль внимания с несколькими головами.

После этого идет модуль Transformer. Основная часть обработки структурированных данных осуществляется через модуль трансформера. Он включает несколько слоев, каждый из которых состоит из модуля внимания, полносвязных слоев и функций активации (например, ReLU [1] или GELU [10]). Все слои могут быть нормализованы перед или после применения операций.

Поскольку трансформер может хорошо улавливать контекст, он показал результаты сильно лучше, чем все предыдущие варианты. Но все-таки не превзошел контрастное обучение. Так, качество классификации с трансформером получилось 0.603. Результат работы




(a) График accuracy для модели Seq2Target с головой SNN на валидации




(b) График функции потерь для модели Seq2Target с головой SNN на валидации

Рисунок 4.4: Результаты работы модели Seq2Target с головой SNN

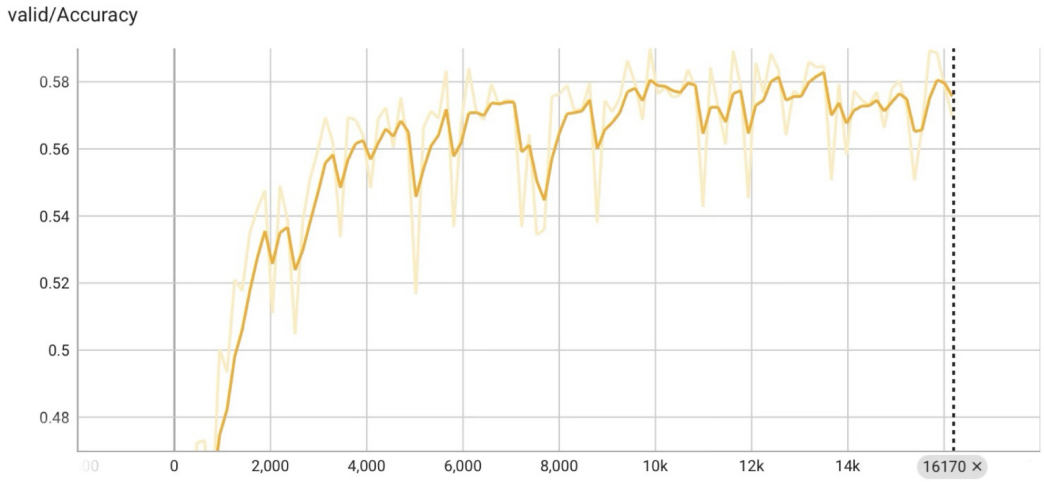
трансформера можно увидеть на графиках 4.6.

Для всех этих голов было необходимо подобрать гиперпараметры. Для SNN - это количество и размеры слоев, коэффициент регуляризации и размер эмбеддингов.

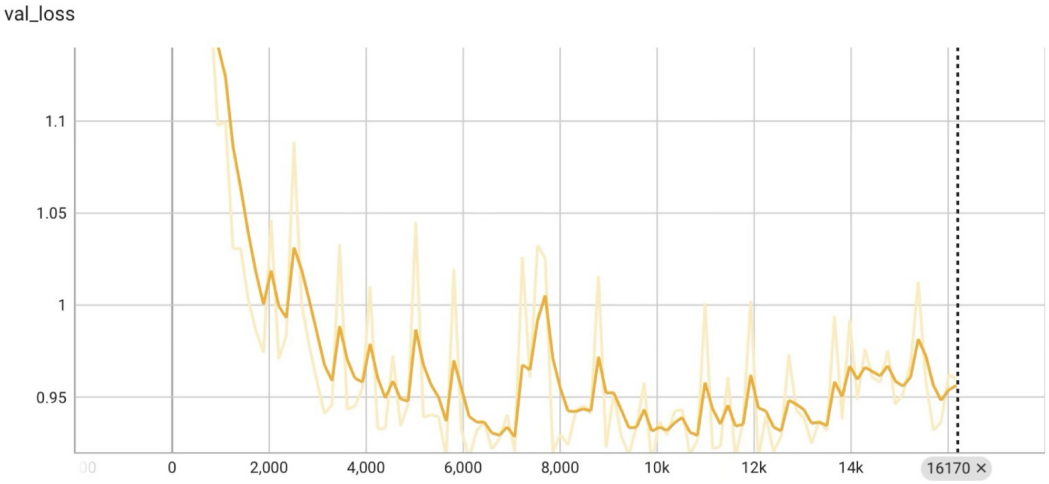
Для ResNet - это количество блоков, размеры блоков, рармерность скрытого слоя и коэффициенты регуляризации.

Для FT-Transformer гиперпараметрами выступают количество слоев, размерности токенов, количество голов, размер скрытого слоя и коэффициенты регуляризации.

Для подбора всех гиперпараметров использовался фреймворк optuna.




(a) График accuracy для модели Seq2Target с головой ResNet на валидации

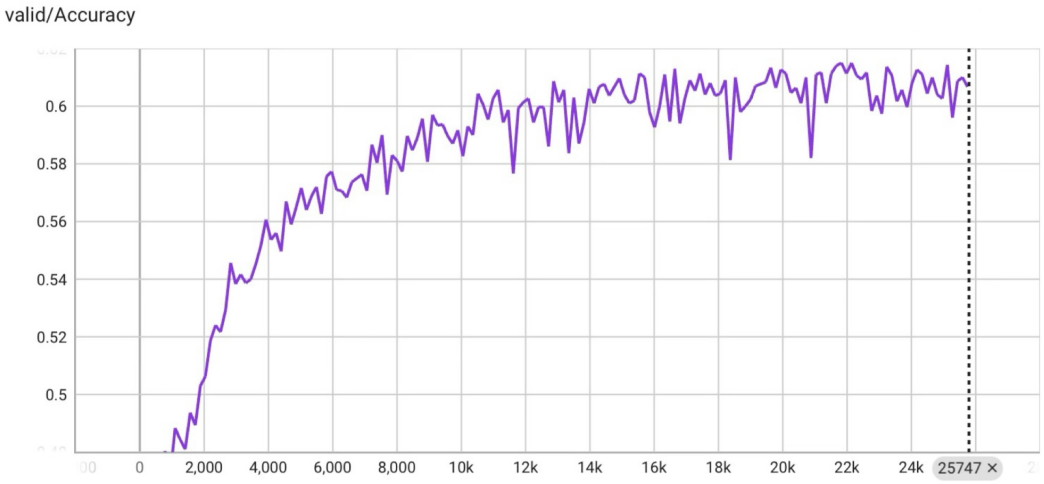



(b) График функции потерь для модели Seq2Target с головой ResNet на валидации

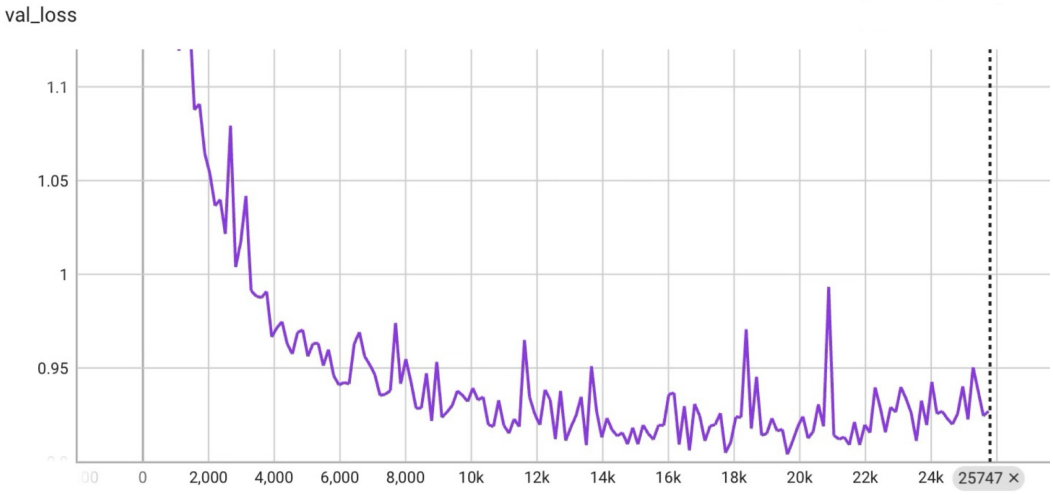
Рисунок 4.5: Результаты работы модели Seq2Target с головой ResNet

## 4.3 Решение Downstream задачи

Лучше всего после CoLES + GBM показал себя Downstream метод, который использует обученную CoLES рекуррентную сеть. После того, как self-supervised обучение заканчивается, начинается обучение SequenceToTarget сети, которая использует эту самую рекуррентную сеть и классифицирующую голову. В наших экспериментах downstream задача со стандартной односвязной классифицирующей головой показала качество 0.609. Если же вместо простой головы использоваь ResNet, то качество возрастает до 0.642. Для FT-transformer в качестве головы accuracy стало равным 0.643.




(a) График accuracy для модели Seq2Target с головой Transformer на валидации




(b) График функции потерь для модели Seq2Target с головой Transformer на валидации

Рисунок 4.6: Результаты работы модели Seq2Target с головой Transformer

# 5 Выводы и результаты

Мы исследовали методы self-supervised, supervised классификации. Посмотрели варианты улучшения CoLES, SequenceToTarget. Попробовали новые архитектуры, такие как ResNet, Transformer для табличных данных. В итоге, не получилось превзойти результат классификации CoLES + GBM, но удалось сильно улучшить результат supervised метода, хоть данных для обучения с учителем на датасете, с которым происходила работа, в несколько раз меньше, чем для обучения без учителя. Лучшим же способом классификации оказалось решение downstream задачи, где также удалось улучшить качество относительно базового алгоритма. В таблице 5.1 приведены получившиеся результаты.

Таблица 5.1: Итоговые результаты работы

| Name of the experiment | CoLES + GBM | Seq2Target base | Seq2Target CoLES embeddings | Seq2Target SNN | Seq2Target ResNet | Seq2Target Transformer |
|---|---|---|---|---|---|---|
| accuracy | 0.639 | 0.562 | 0.567 | 0.542 | 0.597 | 0.603 |

Таблица 5.2: Результаты Downstream задач

| Name of the experiment | Basic Downstream | ResNet Downstream | transformer Downstream |
|---|---|---|---|
| accuracy | 0.609 | 0.642 | 0.639 |

# Список литературы


[1] Abien Fred Agarap. “Deep Learning using Rectified Linear Units”. В: *CoRR* abs/1803.08375 (2018). arXiv: 1803.08375. URL: http://arxiv.org/abs/1803.08375.

[2] Dmitrii Babaev, Nikita Ovsov, Ivan Kireev, Maria Ivanova, Gleb Gusev, Ivan Nazarov и Alexander Tuzhilin. “CoLES: Contrastive Learning for Event Sequences with Self-Supervision”. В: *Proceedings of the 2022 International Conference on Management of Data*. SIGMOD/PODS ’22. ACM, июнь 2022. DOI: 10.1145/3514221.3526129. URL: http://dx.doi.org/10.1145/3514221.3526129.

[3] Jane Bromley, I. Guyon, Yann Lecun, Eduard Sackinger и R. Shah. “Signature verification using a Siamese time delay neural network”. English (US). В: *Advances in neural information processing systems (NIPS 1993)*. Т. 6. Morgan Kaufmann, 1993.

[4] Rahul Dey и Fathi M. Salem. “Gate-variants of Gated Recurrent Unit (GRU) neural networks”. В: *2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)*. 2017, с. 1597—1600. DOI: 10.1109/MWSCAS.2017.8053243.

[5] Anna Veronika Dorogush, Vasily Ershov и Andrey Gulin. “CatBoost: gradient boosting with categorical features support”. В: *CoRR* abs/1810.11363 (2018). arXiv: 1810.11363. URL: http://arxiv.org/abs/1810.11363.

[6] Nir Ailon Elad Hoffer. “Deep metric learning using Triplet network”. В: *arXiv preprint, arXiv:1412.6622, version 4* (2018).

[7] Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov и Artem Babenko. “Revisiting Deep Learning Models for Tabular Data”. В: *NeurIPS*. 2021.

[8] Yufei Guo, Xuhui Huang и Zhe Ma. *Direct Learning-Based Deep Spiking Neural Networks: A Review*. 2023. arXiv: 2305.19725 [cs.CV].

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren и Jian Sun. "Deep Residual Learning for Image Recognition". В: *CoRR* abs/1512.03385 (2015). arXiv: 1512.03385. URL: http://arxiv.org/abs/1512.03385.

[10] Dan Hendrycks и Kevin Gimpel. "Bridging Nonlinearities and Stochastic Regularizers with Gaussian Error Linear Units". В: *CoRR* abs/1606.08415 (2016). arXiv: 1606.08415. URL: http://arxiv.org/abs/1606.08415.

[11] Sepp Hochreiter и Jürgen Schmidhuber. "Long short-term memory". В: *Neural computation* 9.8 (1997), с. 1735—1780.

[12] Sergey Ioffe и Christian Szegedy. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". В: *CoRR* abs/1502.03167 (2015). arXiv: 1502.03167. URL: http://arxiv.org/abs/1502.03167.

[13] Kenton Lee Jacob Devlin Ming-Wei Chang и Kristina Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". В: *Conference of the North American Chapter of the Association for Computational Linguistics* 1 (2019), с. 4171—4186.

[14] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye и Tie-Yan Liu. "Lightgbm: A highly efficient gradient boosting decision tree". В: *Advances in neural information processing systems* 30 (2017), с. 3146—3154.

[15] Daniel Borrajo Keshav Ramani. "Classification of Tabular Data by Text Processing". В: *arXiv preprint, arXiv:/2311.12521, version 1* (2023).

[16] Zhuang Ma и Michael Collins. "Noise Contrastive Estimation and Negative Sampling for Conditional Models: Consistency and Statistical Efficiency". В: *CoRR* abs/1809.01812 (2018). arXiv: 1809.01812. URL: http://arxiv.org/abs/1809.01812.

[17] TomasPfiste Sercan O.Arık. "TabNet: Attentive Interpretable Tabular Learning". В: *arXiv preprint, arXiv:/1908.07442, version 5* (2020).

[18] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever и Ruslan Salakhutdinov. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". В: *Journal of Machine Learning Research* 15.56 (2014), с. 1929—1958. URL: http://jmlr.org/papers/v15/srivastava14a.html.

[19] Carlos Guestrin Tianqi Chen. "XGBoost: A Scalable Tree Boosting System". В: *arXiv preprint, arXiv:/1603.02754, version 3* (2016).

[20] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser и Illia Polosukhin. "Attention Is All You Need". В: *CoRR* abs/1706.03762 (2017). arXiv: 1706.03762. URL: http://arxiv.org/abs/1706.03762.

[21] Kai Yao, Alberto Ortiz и Francisco Bonnín-Pascual. "A Centroid Loss for Weakly Supervised Semantic Segmentation in Quality Control and Inspection Application". В: *CoRR* abs/2010.13433 (2020). arXiv: 2010.13433. URL: https://arxiv.org/abs/2010.13433.

[22] Dong Yi, Zhen Lei и Stan Z. Li. "Deep Metric Learning for Practical Person Re-Identification". В: *CoRR* abs/1407.4979 (2014). arXiv: 1407.4979. URL: http://arxiv.org/abs/1407.4979.

[23] Nikolay Kartashev Yury Gorishniy Ivan Rubachev. "TabR: TABULAR DEEP LEARNING MEETS NEAREST NEIGHBORS IN 2023". В: *arXiv preprint, arXiv:/2307.14338, version 2* (2023).

[24] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun и Stéphane Deny. "Barlow Twins: Self-Supervised Learning via Redundancy Reduction". В: *arXiv preprint arXiv:2103.03230* (2021).