Методы глубокого обучения как инструмент преодоления кризиса трекинга частиц в экспериментах на коллайдерах высокой светимости

Ососков Геннадий Алексеевич

Лаборатория информационных технологий им. М.Г.Мещерякова ОИЯИ
Профессор университета «Дубна»

email: gososkov@gmail.com
https://gososkov.ru/u/UNI-DUBNA/

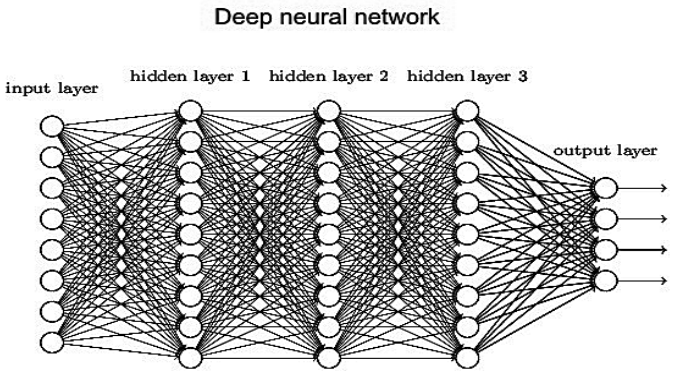
Различные типы глубоких нейросетей

1. Многослойная прямоточная нейронная сеть

Задана обучающая выборка (X_i,Z_i). Инициируем веса w_{ij}, выбираем активационную функцию $g(x)$ (обычно сигмоид $\sigma(x)$) и обучаем сеть.

Прежний опыт: чтобы обучить сеть применяют метод обратного распространения ошибки, когда методом градиентного спуска минимизируют по всем весам квадратичную функцию ошибки сети:

$$E=\Sigma_m\Sigma_{ij}\,(y_i^{(m)}-z_i^{(m)})^2\rightarrow min_{\{wij\}}$$

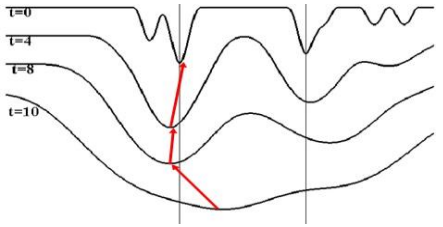


Возникающие проблемы: проклятье размерности, застревание E в ложном минимуме, переобучение, затухающий градиент и др.

Как их решают

1. **Уменьшение размерности сети, методы: PCA, автоэнкодер, dropout при обучении.**
2. **Размерность и ложные минимумы, методы: Имитация отжига**

$$g(u)=\frac{1}{1+e^{-\lambda u}}$$



Стохастический градиентный спуск (SGD)- требуется только один проход по обучающим данным, когда градиент вычисляется **всего на одном элементе обучения, выбираемом случайно,** но попеременно из разных классов, чтобы выскочить из локального минимума. Чтобы не «промахнуться» веса обновляют с «моментом инерции» $\Delta w_i=\eta\bullet Gradw+\alpha\bullet\Delta w_{i-1}$.

$\lambda=1/t$ **E=E(t,W)**

Для больших массивов данных **SGD работает много быстрее**.

2. Сверточные нейросети для распознавания изображений

Мотивация: 1. Время обучения многослойных персептронов очень велико

2. Прямое применение регулярных ИНС к распознаванию изображений бесполезно из-за двух основных факторов: (i) входное 2D-изображение в виде сканированного 1D-вектора означает потерю топологии пространства изображения; (ii) полносвязность ИНС, где каждый нейрон полностью связан со всеми нейронами предыдущего слоя, слишком расточительна из-за проклятия размерности, кроме того, огромное количество параметров быстро приводит к переобучению

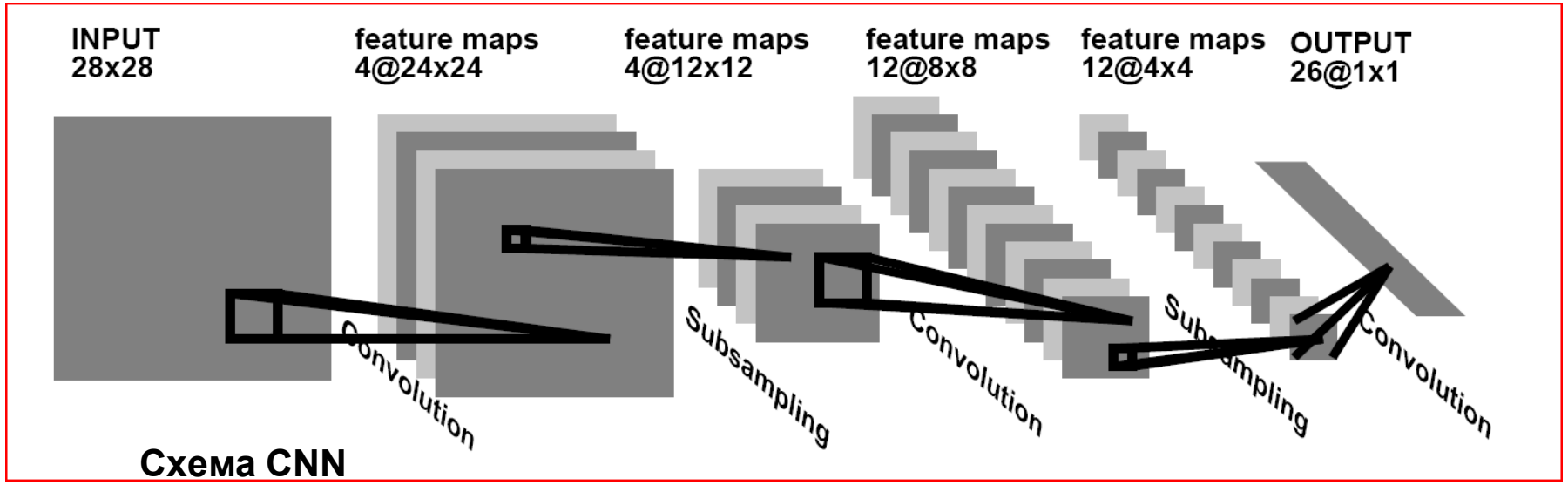


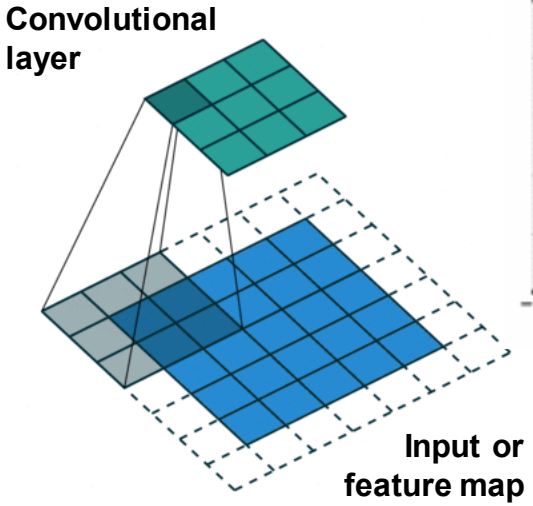
Схема CNN

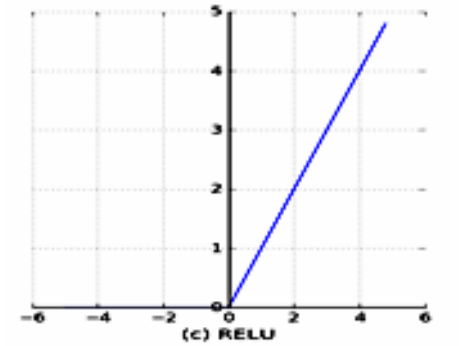
Поясняющая аналогия

Вместо этого, сверточные сети - Convolutional Neural Networks (CNN) принимают на вход двумерные цветные изображения, а нейроны в слое CNN связаны только с малой областью предыдущего слоя

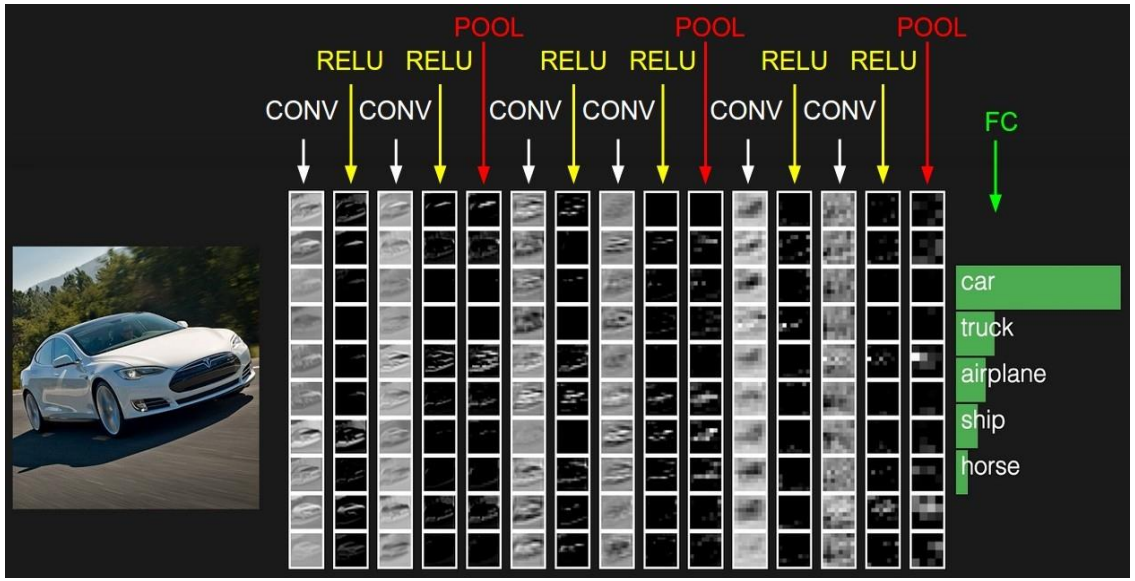
Основы архитектуры CNN

Архитектура CNN: последовательность слоев, каждый слой преобразует один набор активаций в другой через **фильтр-свертку с ядром.** Основные типы слоев для CNN: **Сверточный слой, Слой объединения** (pooling) и **скрытый слой персептрона с обучением backprop**). Также существуют слои **RELU** (rectified linear unit), выполняющие операцию **max (0, x) .**

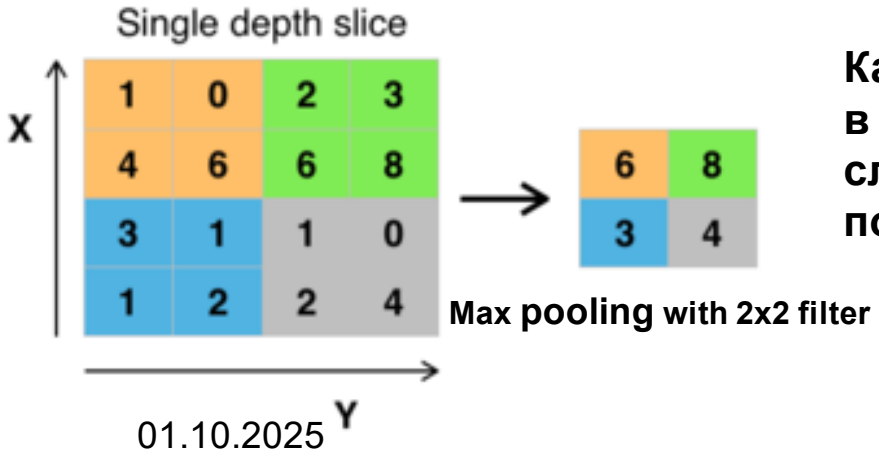




Activation: ReLU



Example of classifying by CNN



Max pooling with 2x2 filter

Каждый слой принимает входные 3D данные (x, y, RGB) и преобразует их в выходные 3D данные. Чтобы построить все фильтры сверточных слоев, CNN должна быть обучена на помеченных изображениях с помощью метода обратного распространения ошибки.

3. Изменяющийся мир требует рекуррентных нейросетей

- В жизни мы имеем дело с объектами, изменяющимися во времени, и наш мозг, воспринимая обстановку и решая, что делать дальше, всегда исходит из знания того, что уже было.
- Однако обычные нейросети с одним скрытым слоем, глубокие сети и даже такие продвинутые сети, как сверточные, предназначены для работы со статическими объектами. **Обычное обучение не поможет традиционной нейросети смоделировать будущее состояние объекта.**
- Для описания динамического объекта **нейронная сеть должна обладать некоей памятью**, чтобы исходя не только из настоящего его состояния, но и из прошлого, нейросеть могла бы моделировать его последующее состояние.

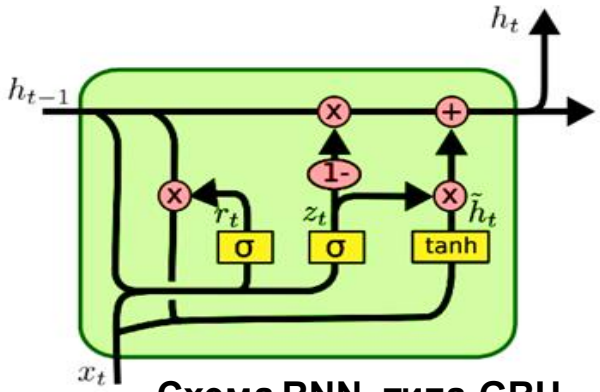


Схема RNN типа GRU

Эту проблему решает семейство новых глубоких нейросетей, называемых рекуррентными (Recurrent Neural Networks - RNN).
В практике ЛИТ применяются многослойные RNN сети типа **Управляемый Рекуррентный Модуль (Gated Recurrent Unit - GRU).**
GRU содержат три взаимодействующих слоя, позволяющих удалять и изменять информацию с помощью **фильтров – gates,** которые состоят из слоя сигмоидальной нейронной сети и операции поточечного умножения и позволяют менять информацию на основании задаваемых условий.

. **Основной компонент GRU – это своеобразная память сети –линия, проходящая по верхней части схемы.**
В GRU есть три фильтра для контроля состояние ячейки.
(1) фильтр «обновления» (update gate) определяет какую часть входной информации выбросить и что сохранить.
(2) (2) фильтр, объединяющий состояние ячейки со скрытым состоянием. (3) формирование выхода.

С помощью нейросети, основанной на GRU нам удалось решить задачу восстановления траекторий элементарных частиц в детекторе GEM эксперимента BM@N в ОИЯИ.

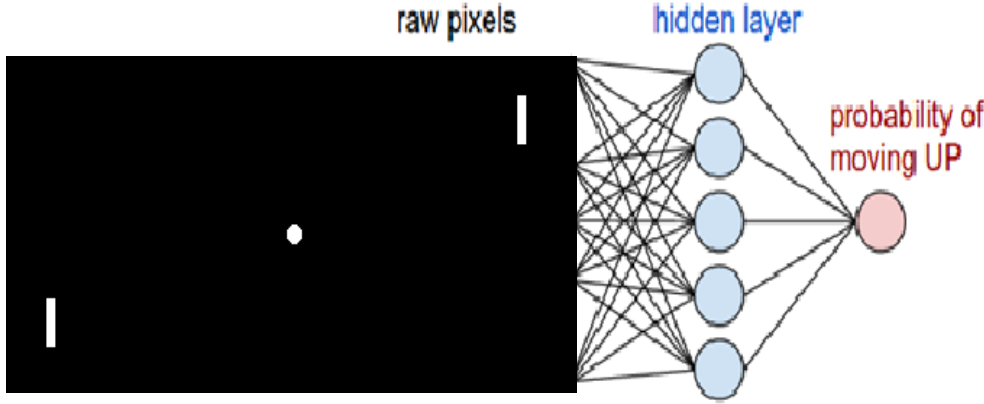
4. Трансфо́рмер — нейросетевая архитектура, **сочетающая преимущества как сверточных, так и рекуррентных глубоких нейронных сетей**. Трансформеры предназначены для обработки таких последовательностей, как текст на естественном языке, и решения задач машинного перевода, автоматического реферирования и обработки изображений. https://arxiv.org/abs/1706.03762
Архитектура трансформера подобна автоэнкодеру и состоит из кодировщика и декодировщика.
Самый важный механизм в архитектуре трансформера — это внимание (attention), который в процессе обучения повышает вес соответствия одного слова другому в предложении. Кодировщик получает на вход векторизованую последовательность и состоит из слоев самовнимания (вход из предыдущего слоя) с последующими слоями МСП, декодировщик состоит из аналогичных слоев. Эта конструкция позволяет обученной нейросети-переводчику правильно расположить слова в выходном переведенном тексте.
В отличие от RNN, трансформеры не требуют обработки последовательностей по порядку. Например, для текста трансформеру не требуется обрабатывать конец текста после обработки его начала. Благодаря этому **трансформеры распараллеливаются легче, чем RNN, и могут быть быстрее обучены**

Мы использовали недавние результаты, когда механизм внимания был реализован для обработки данных в виде облака точек с помощью фреймворка Point Cloud Transformer (PCT) и применили PCT, названный Perceiver, для решения важной задачи отсева зашумляющего фона фейковых измерений в экспериментах с полосковыми и strow-tube трековыми детекторами.
В настоящее время мы занимаемся разработкой метода , когда измеренные хиты используются непосредственно для восстановления параметров треков без предварительной подгонки траектории в магнитном поле, чтобы затем использовать потенциал трансформеров для нахождения параметры треков прямо из сырых данных, минуя времяемкий этап кластеризации измерений для формирования хитов.

Другие типы глубоких нейросетей

5. Нейросети Обучаемые с подкреплением Reinforcement learning Networks. Они реализуют такое обучение, когда агент нейронной сети, находясь в некотором состоянии, взаимодействует с окружающей средой, которая вознаграждает агента за его действия и сообщает, в какое состояние агент перешел после этого, чтобы увеличить общее вознаграждение.

Применения обучения с подкреплением: **роботы, самоуправляемые автомобили**, торговые боты для игры на фондовом рынке, чат-боты, которые обучаются от диалога к диалогу, разработка игровых программ и т.д.



Example. Ping-pong game http://karpathy.github.io/2016/05/31/rl/

6. Генеративные Состязательные сети (Generative Adversarial Networks, GAN). GAN реализует принцип состязательности между генеративной сетью и сетью дискриминации. Генеративная сеть G генерирует наиболее реалистичный образец, а дискриминационная сеть D обучается различать подлинные и поддельные образцы.

Применения GAN:

- Получения фотореалистичных изображений и картин;
- **Написание стихов, текстов статей и даже диссертаций**
- Создание фильмов и мультипликаций.
- Создание трёхмерной модели объекта с помощью фрагментарных изображений
- **Моделирование сложных физических процессов** в детекторах экспериментальной физики

Этические проблемы GAN приложений!

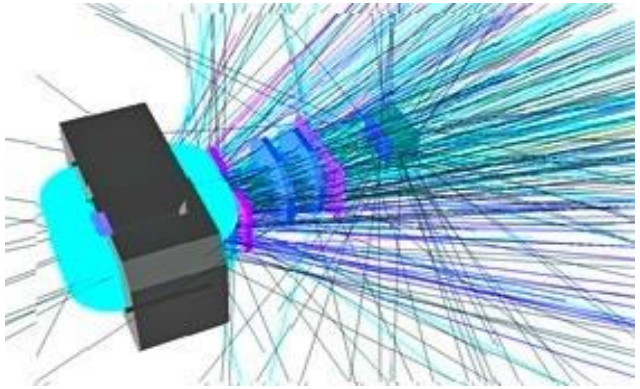
Опасность дипфейков: политика, мошенничество и шантаж с использованием дипфейков

https://www.kaspersky.ru/resource-center/threats/protect-yourself-from-deep-fake

Данные, технологии и проблемы в экспериментальной ФВЭ в наше время

- Коллайдеры LHC и NICA
- Электронный съем данных
- Пиксельные и полосковые трековые детекторы
- Всемирная паутина - интернет
- Компьютерные фермы и суперкомпьютеры
- Распределенные вычисления, GRID, WLCG
- Машинное и глубокое обучение

Современные эксперименты с электронным съемом данных



Эксперимент BM@N. Стриповый GEM-детектор внутри магнита

Трековый детектор TPC внутри магнита MPD. Показано смоделированное событие от взаимодействия ионов золота, порождающее тысячи треков

Схема комплекса NICA с экспериментами MPD, SPD, BM@N

Задачи: реконструкция событий по данным измерения в трековых и других детекторах

Данные, измеренные в экспериментах, и постановки задач

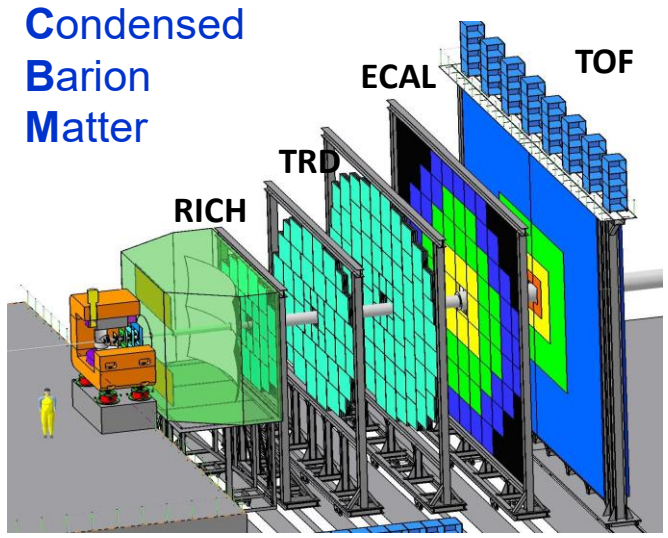


Схема установки CBM

CBM эксперимент (Германия, GSI, будет запущен в 2024 году)

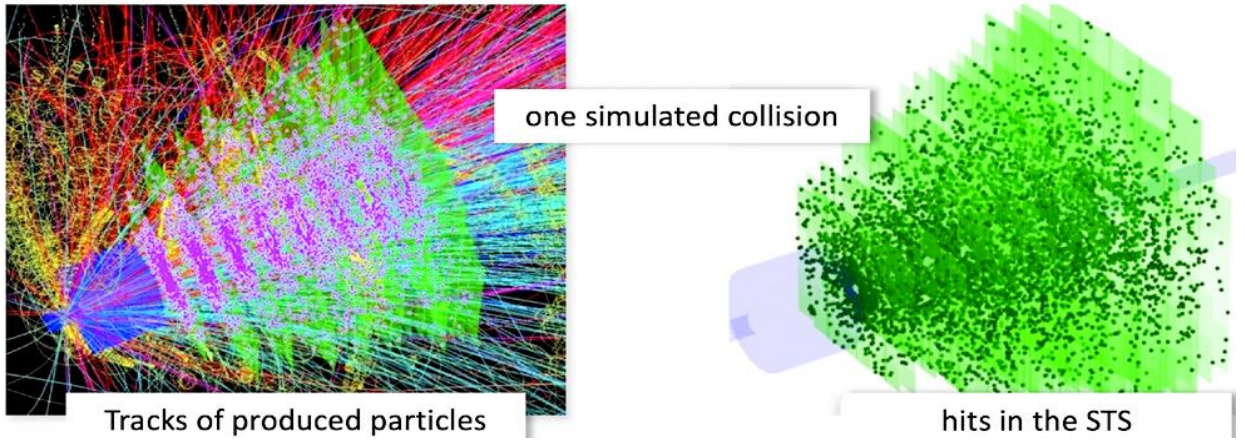
Скорость передачи данных:

10^7 событий в сек,

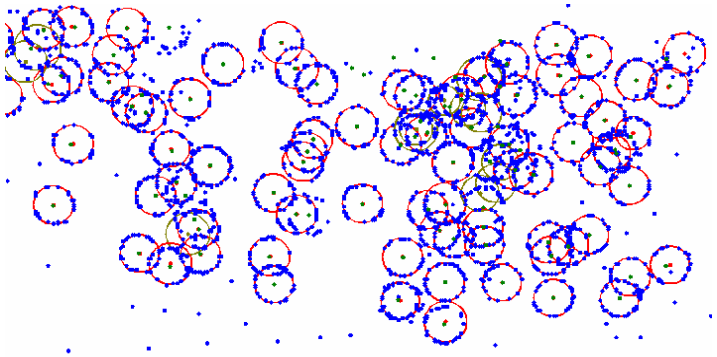
~1000 треков на событие

~100 чисел на трек

Итого: 1 терабайт/сек!



Вид модельного событий взаимодействия Au+Au в вершинном детекторе



Фрагмент данных фотодетектора. В среднем 1200 точек, образующих 75 колец

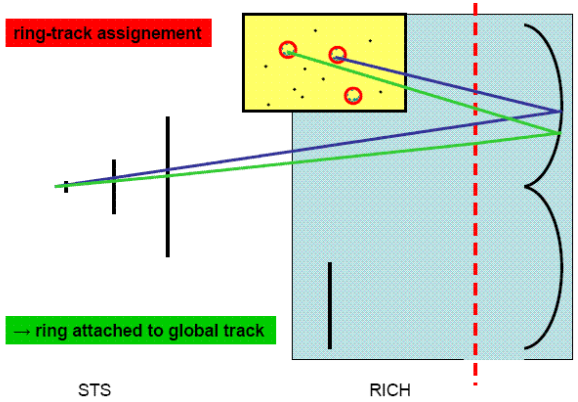
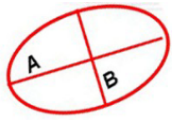


Схема детектораRICH черенковского излучения

Проблемы CBM, решаемые методами машинного обучения: распознавание всех этих треков и колец RICH и оценка их параметров, с учетом их перекрытий, шумов и оптических искажений, ведущим к эллиптическим формам колец (подгонка эллипса), идентификация частиц, анализ спектров инвариантных масс короткоживущих частиц, поиск резонансов.



До 2015 года все эти задачи решались с помощью персептронов с одним скрытым слоем, нейросетей Хопфилда, фильтра Калмана, робастными методами и применением вейвлет-анализа.

Глубокое обучение ждало новых компьютерных технологий

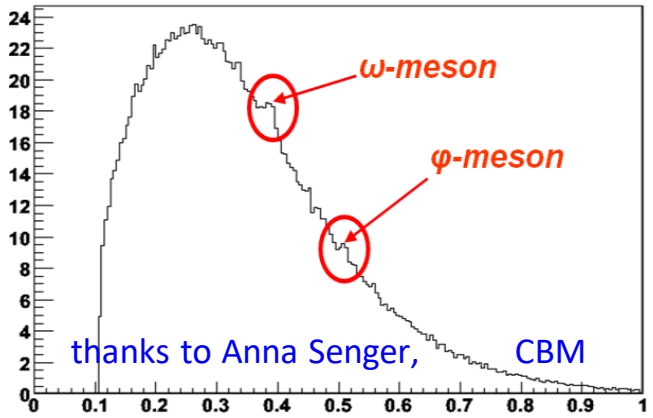
Основные этапы анализа данных в текущих экспериментах ФВЭ

- Сбор данных со многих каналов на многих субдетекторах (млн/сек)
- Решить, считывать или отбросить событие (триггеры разных уровней)
- Реконструировать событие (собрать всю информацию)
- Отправить данные на хранение
- Анализировать их
  - корректировка данных с учетом искажений детектора: калибровка, алайнмент
  - нахождение хитов, трекинг, поиск вершин, распознавание черенковских колец,
  - удаление ложных объектов (фейков)
  - алгоритмы анализа от физиков-пользователей
  - уменьшение объема данных

Применяемые методы машинного обучения

- Преобразования Хафа,
- клеточные автоматы,
- фильтр Калмана,
- искусственные нейронные сети,
- робастное оценивание,
- вейвлет-анализ и т.д.

- Детальное моделирование всех процессов эксперимента
  - взаимодействия пучка с мишенью или налетающей частицей
  - рассеяния при прохождении частиц через детекторы
  - искажений при оцифровке и т. д.
- Сравнение теории и физических параметров, полученных в эксперименте
  - анализ спектров инвариантных масс короткоживущих частиц резонансов



- Использовать современные средства компьютинга для достижения наивысшей скорости и масштабируемости обработки

Неизбежность создания всемирной интернет-сети распределенных вычислений (Worldwide LHC Computing Grid -WLCG)

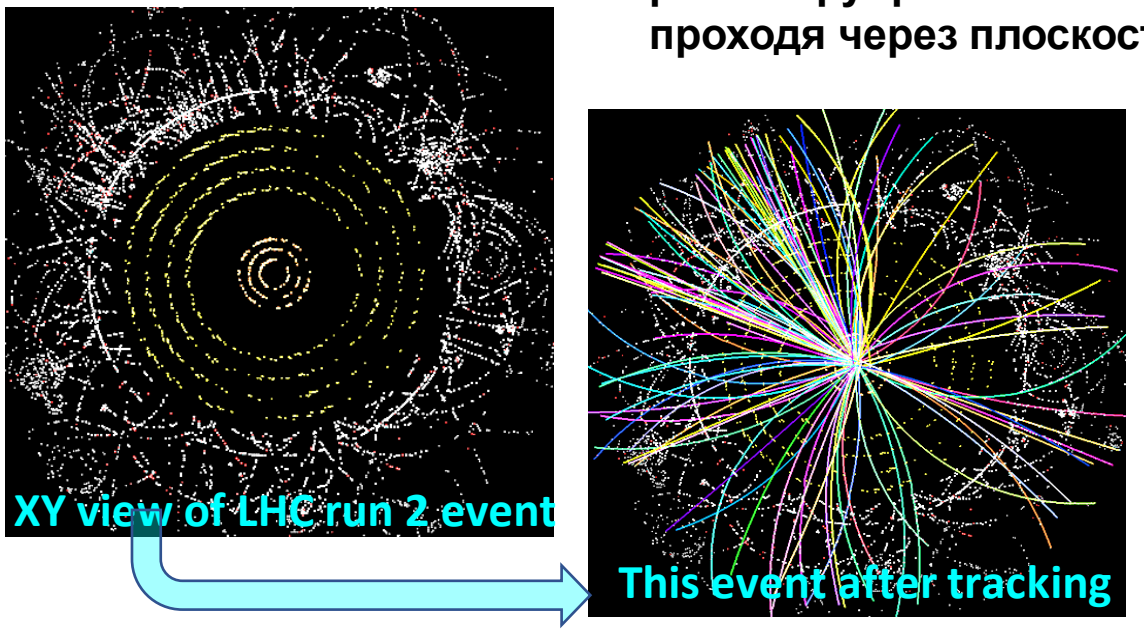
Parallel programming of optimized algorithms Grid-cloud technologies which changed considerably HEP data processing concept
See *Scientific data management in the coming decade* https://dl.acm.org/doi/10.1145/1107499.1107503

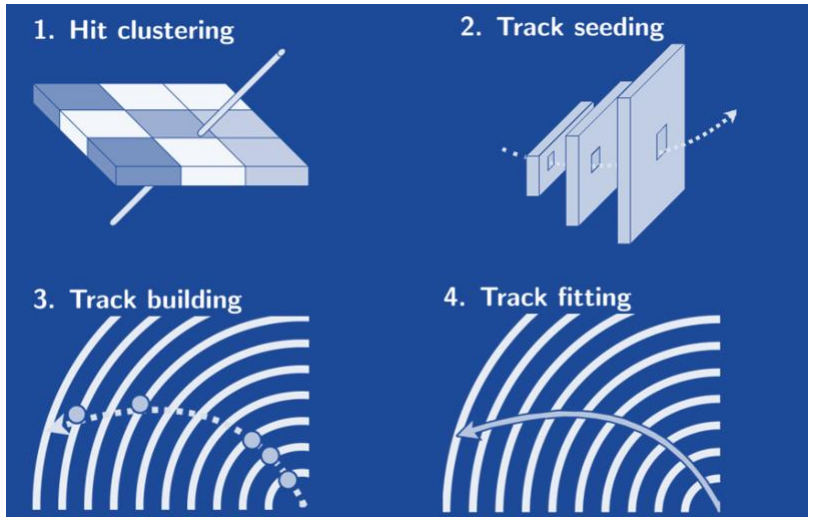
О методах машинного обучения на примере задачи трекинга, как ключевой проблеме реконструкции событий в ФВЭ

Реконструкция должна определять параметры вершин и траекторий (треки) частиц для каждого события.

Что такое трекинг?

Трекинг или распознавание треков - это процесс восстановления траекторий частиц в детекторе ФВЭ путем прослеживания и соединения точек- хитов (*хит* – это реконструированный отклик детектора), которые каждая частица оставляет, проходя через плоскости детектора.



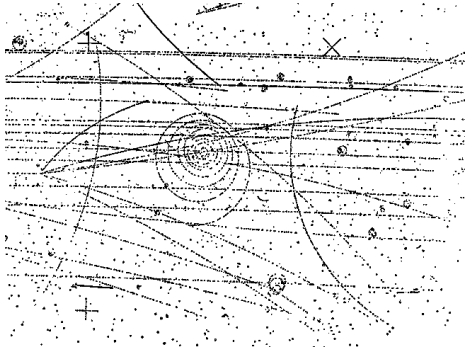


Главная проблема современного трекинга- высокая светимость пучков ускорителей, т.е. мегагерцовый темп поступления данных и банчевая структура пучка

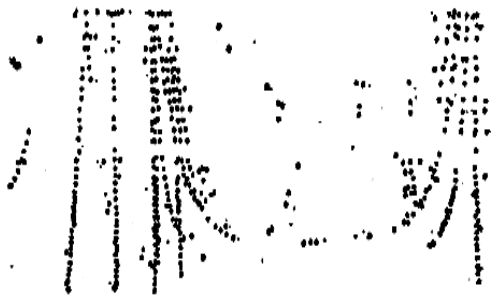
Процедура трекинга включает в себя фазы: (1) получения хитов (hit clustering), (2) построения треков-кандидатов - наборов хитов с вычисленными параметрами (*англ. seeds*), (3) прослеживания треков и (4) их подгонки уравнением движения частицы в магнитном поле.

Эволюция методов трекинга

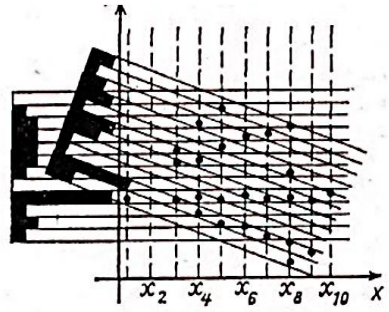
Началось еще в эпоху **пузырьковых камер**, когда события регистрировались на стереофотографиях и **вводились в компьютер вручную**, полуавтоматами или с помощью **сканирующих устройств** типа «Спиральный измеритель», в котором оператор ставил точку в вершину события, откуда шло сканирование снимка по спирали



Снимок события.



Его оцифровка в полярных коорд.

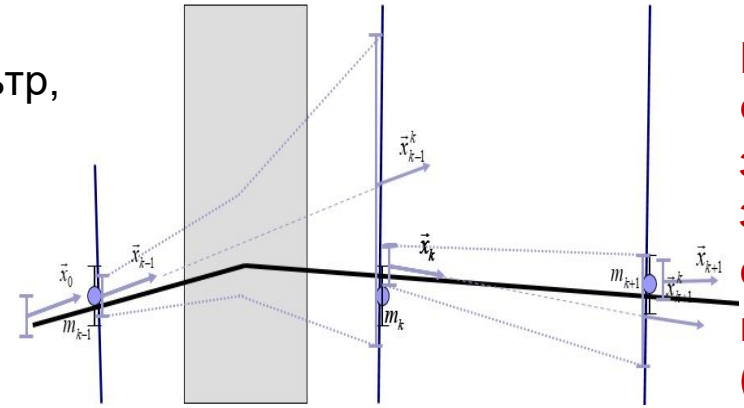


Поворотные гистограммы

Когда пришла **эра электронных экспериментов**, данные измерений стали оцифровываться и сразу поступать **прямо в компьютер**. После многоэтапной фильтрации и процедур алайнмента, наступало время трекинга. Среди многих методов трекинга, самым эффективным оказался метод, использующий **фильтр Калмана**, поскольку он позволяет легко учитывать неоднородность магнитного поля, многократное рассеяние и потери энергии

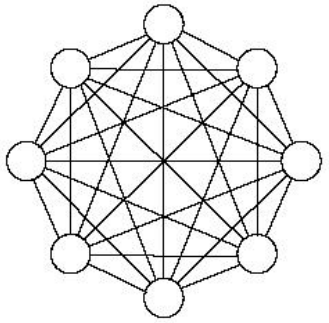
Фильтр Калмана (ФК) – это эффективный рекурсивный фильтр, оценивающий состояние **линейной динамической системы**, используя ряд неточных измерений

Вектор состояния $\vec{x} = (x, y, t_x, t_y, q/p)^T$ итеративно оценивается для предсказания позиции трека на след. координатной плоскости с учетом изменения ковариационной матрицы и коридоров ошибок.



Главный недостаток ФК – необходимость знать начальное значение вектора состояния $\vec{X}$, выполнить «сидинг» (англ. seed- семя

Однако ФК медленный, плохо распараллеливается и масштабируется!



Распознавание треков с помощью сети Хопфилда.

Одно из первых применений нейросетей в ФВЭ -1988 г.

ЗАДАЧА. **Имеется множество N экспериментальных точек на плоскости. Требуется выбрать (распознать) среди них те, по которым проходит некоторое число непрерывных гладких кривых (треков).**

Нейросеть Хопфилда (ХНС) - это полносвязная сеть из бинарных нейронов s_i с симметричной весовой матрицей $w_{ij} = w_{ji}$, $w_{ii} = 0$. Эволюция ХНС приводит ее в некоторое состояние устойчивого равновесия. Функционал энергии сети – это билинейная функция Ляпунова $E(s) = -\frac{1}{2} \Sigma_{ij} s_i w_{ij} s_j$. Теорема Хопфилда: в результате эволюции $E(s)$ убывает в локальные минимумы, соответствующие точкам стабильности сети.

Для нахождения глобального минимума E используется теория среднего поля, термализация сети и механизм «имитационного отжига» (simulated annealing).

Энергетический функционал (Денби и Петерсон, 1988) состоит из двух частей: $E = E_{cost} + E_{constraint}$,

где

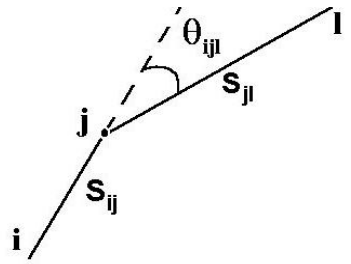
$$E_{cost} = -\frac{1}{2} \sum_{ijkl} \delta_{jk} \frac{\cos^m \theta_{ijl}}{r_{ij} r_{jl}} v_{ij} v_{kl},$$

w_{ijkl}

поощряет связи нейронов принадлежащих одному и тому же треку, т.е. короткие смежные сегменты с малым углом между ними.

Метод сегментов.

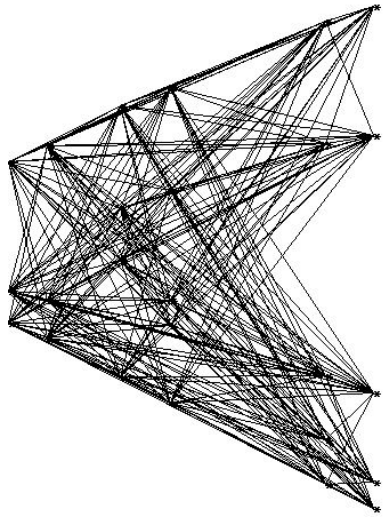
Вводится нейрон s_{ij} как направленный сегмент, соединяющий точки i, j ..



$E_{constraint}$ запрещает как межтрековые связи (бифуркации), так и чрезмерный рост числа самих треков.

Пример применения ХНС для распознавания событий с короткоживущими частицами

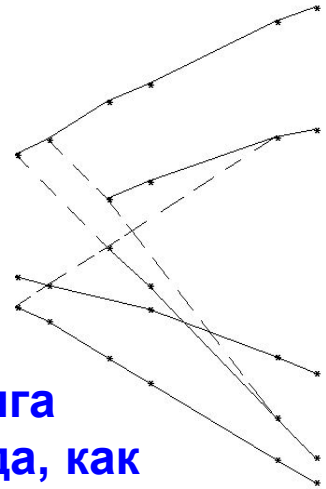
Эксперимент EXCHARM (Протвино 90-е годы) - проблема: в отличие от Денби-Петерсена разрешить бифуркации, но не допустить массовых ветвлений треков



на нулевой итерации

всего 244 нейрона

Заметим: появление даже единственной шумовой точки привело бы к появлению ~80 дополнительных мешающих нейронов



на 30-ой итерации

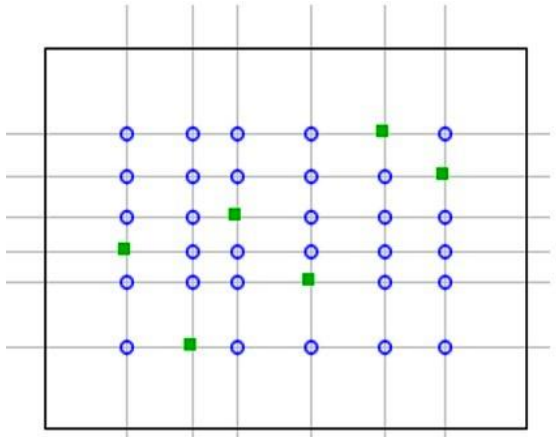
$V_{ij} > 0.5$ у 26 нейронов

Однако, именно практическое применение ХНС для трекинга показало такие малоприемлемые недостатки этого подхода, как медленностью процесса эволюции сети, высокая вероятность попадания функции энергии сети в локальный минимум, чрезмерная чувствительность ХНС к шумам. Кроме того, не учитывалось известное уравнение движения частицы в магнитном поле.

Более удачными были попытки преодолеть эти трудности с помощью использующих ХНС методов «эластичного трекинга», в которых объединялись этапы распознавания и фитирования искомых треков, но и эти методы перестали быть эффективными с ростом сложности экспериментов и множественности событий в них.

Проблемы трекинга для современных детекторов типа GEM и straw-tube

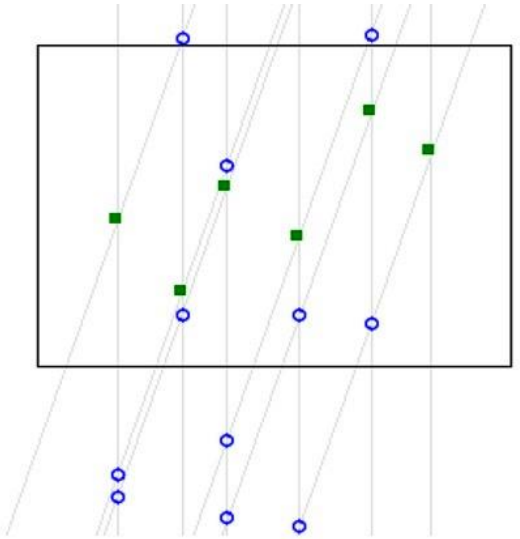
Главная трудность, вызванная спецификой **GEM** и **straw-tube детекторов** – появление ложных отсчетов из-за лишних пересечений стрипов. Для **n** истинных хитов имеем **n^2- n фейков!**



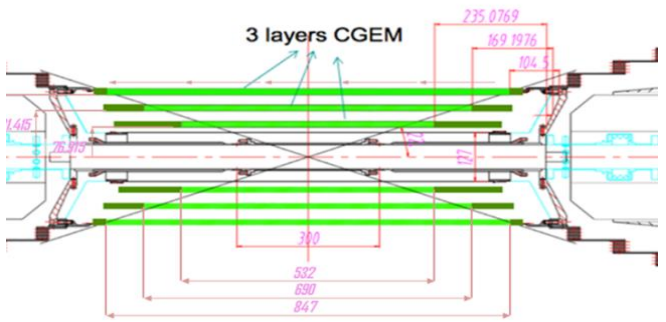
- истинный хит

- ложное пересечение

Можно уменьшить количество фейков – повернуть слой стрипов на маленький угол (5-15 градусов) по отношению к другому слою

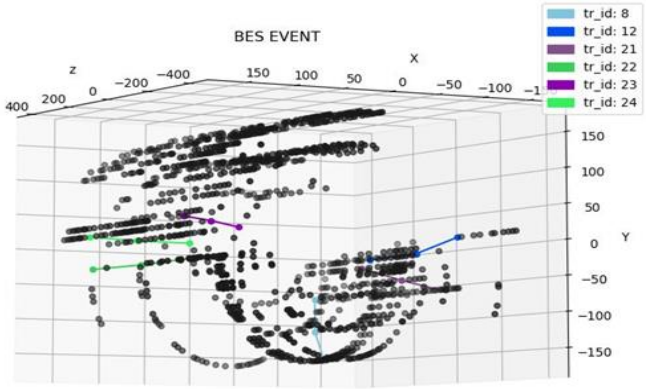


Наклон 15° избавляет нас от трети фейков, но большая их часть остаётся.



Внутренний детектор CGEM-IT эксперимента BESIII, состоящий из трех детектирующих цилиндров

Вторая проблема - <u>пропуски отсчетов</u> из-за неэффективности детекторов. Для детекторов с малым числом станций это вызывает ошибки прослеживания, ведущие к появлению ложно-положительных треков (hosts) . В детекторах с малым числом станций станций пропуск одного хита их трех не даёт восстановить трек в магнитном поле.



Все хиты модельного события

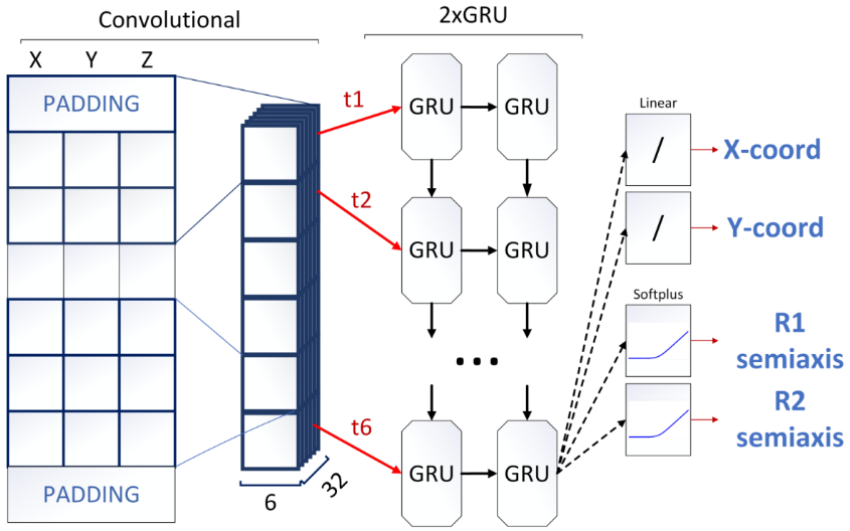
Эти проблемы в условиях сверхвысокого темпа поступления данных из-за высокой светимости новых экспериментов неизбежно потребовали разработки <u>новых методов трекинга с использованием глубоких нейронных сетей</u>

Локальный и пособытийный подходы к трекингу

Два подхода к реализации «глубокого трекинга»

1. Локальный трекинг, когда треки восстанавливаются один за другим, как в алгоритме фильтра Калмана.
Недостатки: медленно, нет возможности увидеть зависимость между отдельными треками или группами треков и такие явления как вторичные вершины, необходимость реализации специального этапа для поиска вторичной вершины.
2. Пособытийный трекинг, при котором распознавание треков среди шумов происходит сразу по всему событию

1. Локальный трекинг для детектора GEM эксперимента BM@N особенно сложен из-за наличия гигантского количества фейковых хитов, что крайне затрудняет поиск тех хитов на последующих станциях детектора, которые являются продолжением обрабатываемого трека.

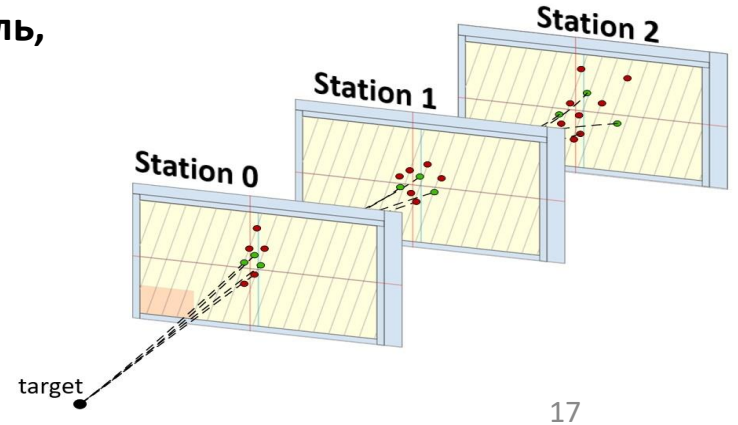


Scheme of the recurrent TrackNETv2 neural network

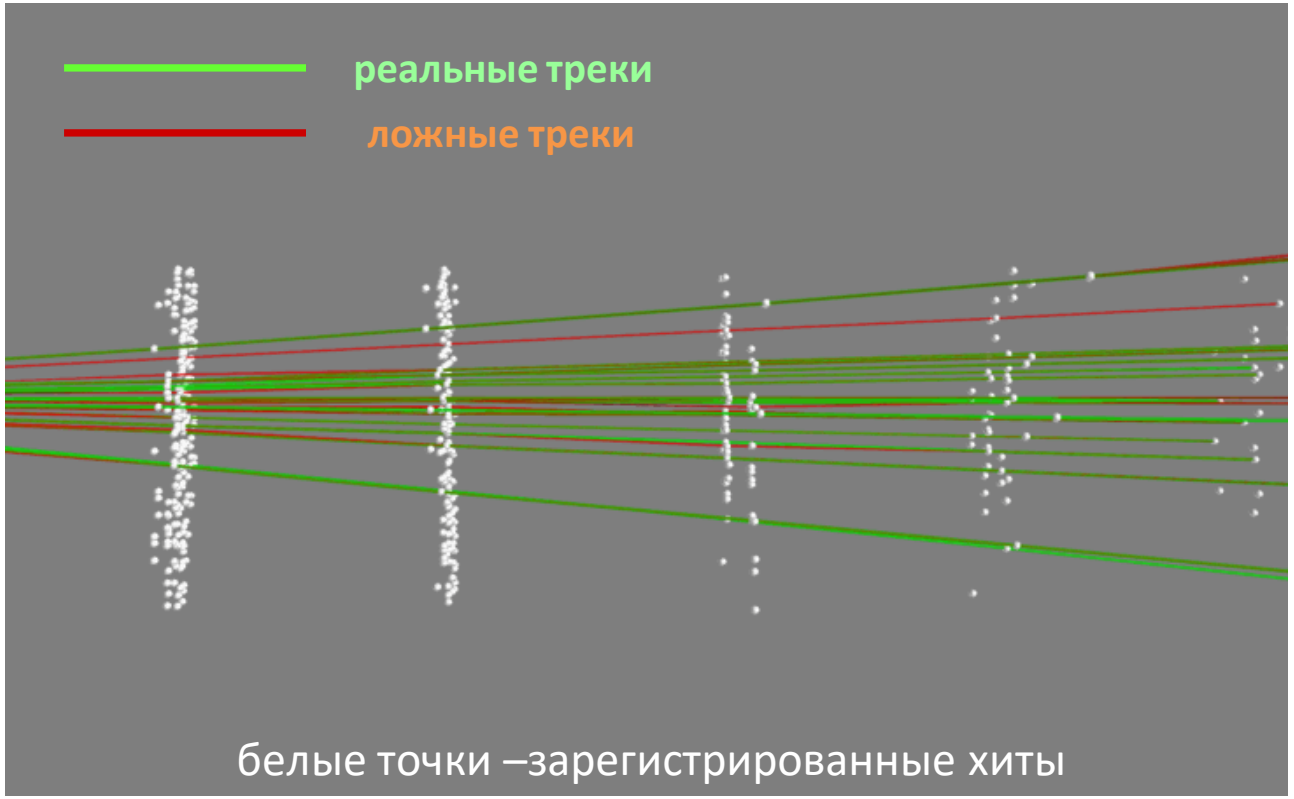
See https://doi.org/10.1063/1.5130102

Гибкость конструкции RNN позволила нам преодолеть эти трудности и придумать новую сквозную нейронную сеть TrackNET с регрессионной частью из четырех нейронов, два из которых предсказывают точку центра эллипса на следующей координатной плоскости, где нужно искать продолжение трека-кандидата, а еще два - определяют полуось этого эллипса.

Это дает нам возможность обучить нашу модель, используя только истинные треки, которые можно извлечь из симуляции Монте-Карло. Таким образом, мы получили нейронную сеть, выполняющую прослеживание трека подобно фильтру Калмана, хотя и без той его части, где выполняется подгонка трека



Проблемы при применения локального трекинга



Пример результатов поиска треков-кандидатов

Проблемы с фейками и пропусками хитов для наиболее употребительных типов трековых детекторов типа GEM или Strow Tubes приводят к тому, что в процессе прослеживания возникает большое число ложных трек-кандидатов, образованных переходом при продолжении на соседний трек или проходом по шумовым хитам.

Поэтому GRU нейросеть TrackNET, обученная на истинных монте-карловских треках, при тестировании не всегда отсеивала эти ложные треки и часть из них неверно распознавались как настоящие.

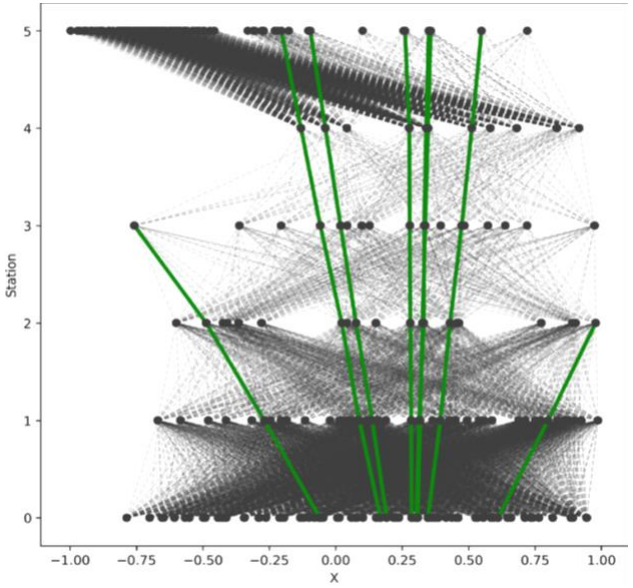
Физики называют их «призраками» (ghosts) – гостами.

Например для модельных данных 7-го рана BM@N обученная TrackNET нашла реальные треки с достаточной эффективностью 98%, однако доля гостов при этом превысила 50%, что недопустимо.
Поэтому потребовалась ввести второй этап трекинга, глобальный, принимающий а вход все треки, распознанные на 1-м этапе, и учитывающий взаимосвязь всех треков каждого события. В итоге доля гостов стала менее 10%.

Пособытийный трекинг

1. Применение графовых нейронных сетей. Эксперимент BM@N

Рассмотрим событие как граф, в котором вершины являются хитами. Узлы между соседними станциями могут быть соединены ребрами, которые являются возможными сегментами треков. Узлы не связаны внутри одного слоя детектора. Задачу трекинга для графовых нейронных сетей (GNN) можно сформулировать как задачу классификации ребер графа – определить, какие из сегментов относятся к реальным трекам, а какие нужно отбросить, как ложные.

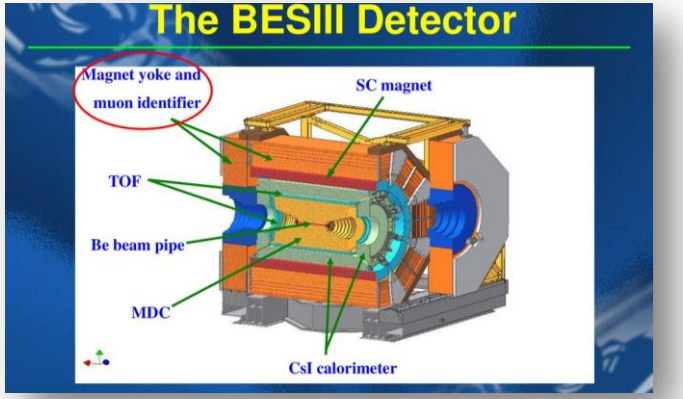


Графическое представление события C + C, 4 ГэВ эксперимента BM@N. Черные узлы и ребра соответствуют фейкам, зеленые узлы и ребра - найденным трекам

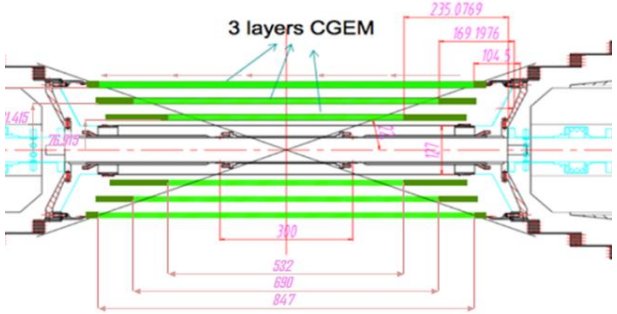
Эта схема похожа уже известный глобальный подход Денби-Петерсона с сегментной нейросетью Хопфилда, где нейросеть подолгу самообучалась отдельно для каждого события, в то время как GNN, где надо найти те ребра, что являются сегментами реальных треков можно обучить на выборке из графов событий, где эти искомые ребра снабжены метками в виде бинарного вектора, указывающего, является ли конкретное ребро истинным (1) или нет (0). Такой подход был успешно реализован в ЦЕРНе для модельных событий с пиксельного детектора, но наши попытки адаптировать их GNN для BM@N событий с огромным фейковым фоном потерпели неудачу из-за возникших проблем с объемом памяти для загрузки графа.

Эти проблемы отпали, когда на втором этапе трекинга GNN была применена к данным на выходе TrackNET. Получая на вход событие, представленное в виде графа треков-кандидатов, сформированных на первом этапе, GNN выдавала в итоге приемлемую эффективность трекинга

2. Применение графовых нейронных сетей, эксперимент BES-III

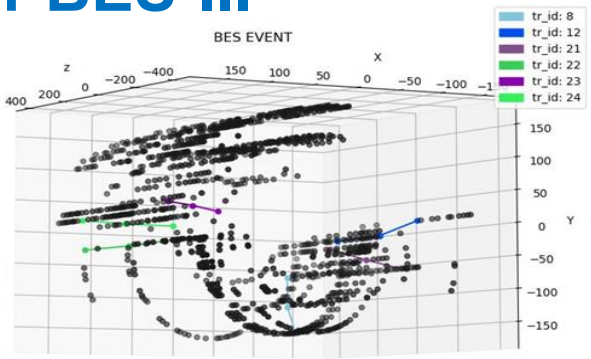


BESIII – коллайдерный эксперимент



Внутренний детектор CGEM-IT эксперимента BESIII, состоящий из трех детектирующих цилиндров

Наличие фейков и пропусков хитов потребовало спользовать другой тип GNN



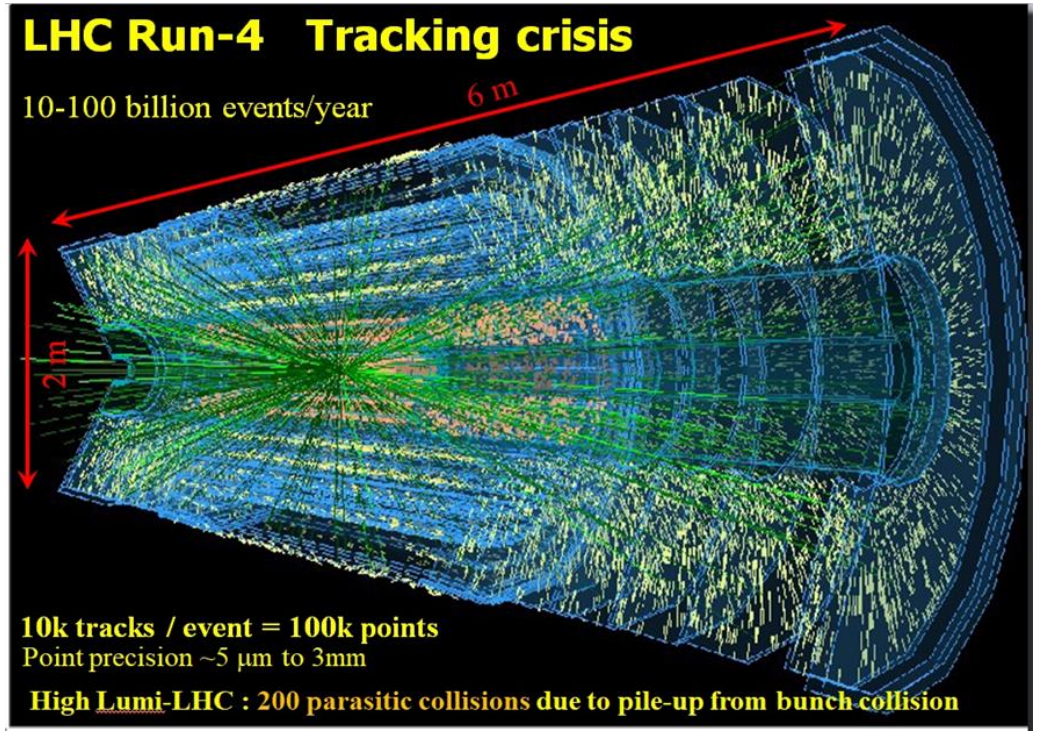
Все хиты модельного события

Граф события инвертируется в линейный дираф, когда ребра представляются узлами, а узлы исходного графа - ребрами. В этом случае информация о кривизне сегментов трека встраивается в ребра графа, что упрощает распознавание треков в море фейков и шумов. **В процессе обучения сеть получает на вход инверсный диграф с метками истинных ребер - сегментов реальных путей. Уже обученная нейронная сеть GraphNet в результате связывает каждое ребро со значением *x ϵ [0,1]* на выходе. Истинные ребра пути - это те ребра, для которых *x* больше некоторого заданного порога *(> 0,5).*** (http://ceur-ws.org/Vol-2507/280-284-paper-50.pdf)

Оценки эффективности трекинга. Оценка **accuracy** как доля найденных треков к общему числу треков-кандидатов – бесполезна и даже опасна, т.к. наша выборка очень сильно несбалансированна. Принято использовать две метрики – **recall и precision. Recall** – это доля истинных треков, которые модель смогла верно реконструировать, найдя все его хиты. **Precision (чистота)** – это доля истинных треков среди тех, которые модель реконструировала

GraphNet	recall	precision
BES-III	96.23	90.64

Эксперименты с высокой светимостью. Кризис трекинга

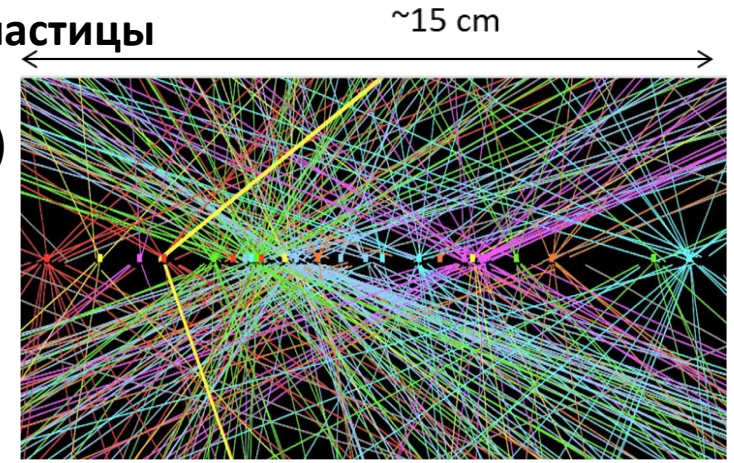


Модельное события в HL-LHC

Для достижения намеченных ультимативных целей светимость Большого адронного коллайдера в ЦЕРНе будет увеличена, так что количество дополнительных столкновений достигнет уровня 200 взаимодействий на пересечение пучка. Это станет вызовом для экспериментов ATLAS и CMS, в частности для алгоритмов реконструкции треков.

Аналогичные планы есть в мегасайнс проекте NICA в ОИЯИ

В условиях большой светимости частицы ускоряются не по отдельности, а группами - банчами (англ. bunch)

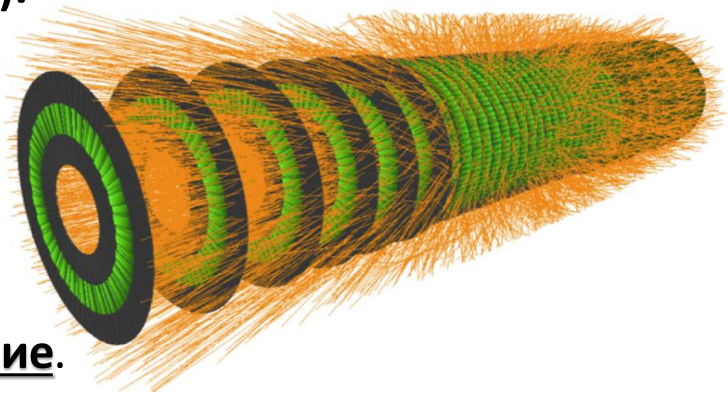


Поэтому моменты столкновений происходят так близко, что треки событий сильно перекрываются в 15 сантиметровой области встречи пучков. Это может дать, в среднем, 10000 треков (100000 хитов) в каждом событии.

Таким образом, реконструкция треков частиц в плотных средах, таких как детекторы БАК высокой светимости (HL-LHC) и NICA, представляет собой сложную проблему распознавания образов для решения которой необходимо развитие новых алгоритмов глубокого трекинга и их распараллеливание на суперкомпьютерах

Эксперименты с высокой светимостью
Что уже сделано по трекингу в мире

- В 2018 году физики из ЦЕРНа и др. физических центров мира, включая Россию, устроили соревнование - TrackML challenge по решении проблемы машинного обучения для трекинга частиц в физике высоких энергий в условиях высокой светимости (DOI 10.1109/eScience.2018.00088).
- Для этого на платформе Kaggle была сделана программа-симулятор, где типичный полностью пиксельный трекинг-детектор БАК из 10 слоев генерирует физические события, из 200 перекрывающихся взаимодействий, как это показано на предыдущем слайде

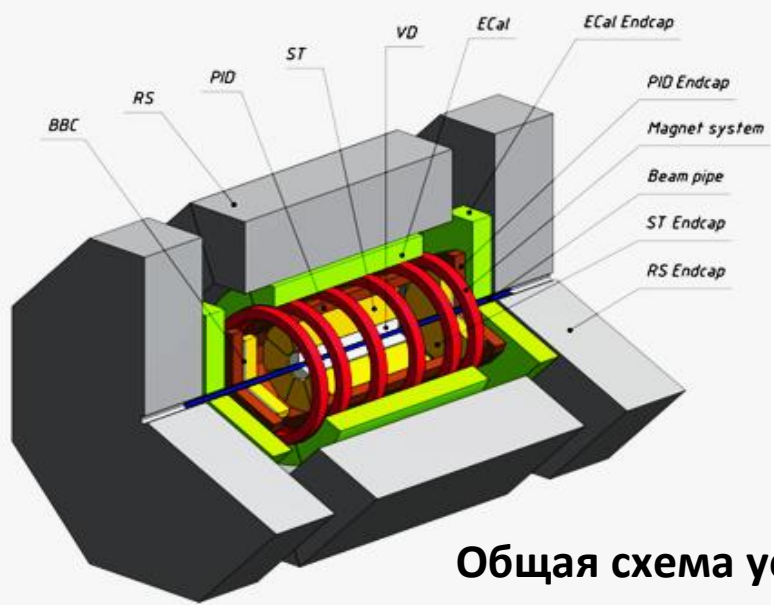


- Победила в 2019 г. команда «Top Quarks» с решением типа ускоренного фильтра Калмана, но не используя при этом глубокое обучение.
- Более значимо влияние TrackML на методы глубокого трекинга показали доклады на CHEP-2024, где превалировали следующие подходы для трекинга с применением глубоких нейросетей:
  - нейромодели на облаках точек - Point cloud transformer (PCT) (https://doi.org/10.1007/s41095-021-0229-5);
  - графовые нейронные сети (GNN), особенно те, что используют механизм внимания(GATNN) (https://arxiv.org/abs/2007.13681) ;
  - Большое внимание было уделено методам реконструкции событий на основе квантового машинного обучения (https://doi.org/10.1007/s42484-021-00054-w)

Эти работы во многом стимулировали новые и вполне перспективные исследования по глубокому трекингу, проведенные с 2018 года в МЛИТ ОИЯИ для экспериментов проектов NICA и BES-III

Трекинг для данных экспериментов высокой светимости. SPD NICA

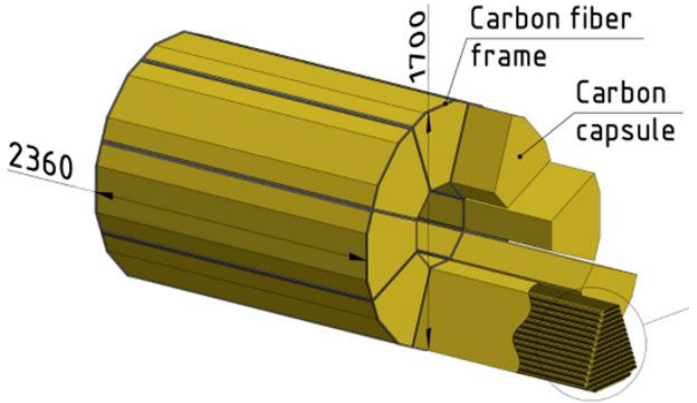
SPD (Spin Physics Detector) разрабатывается для изучения спиновой структуры протона, дейтрона и других явлений, связанных со спином, с помощью поляризованных пучков протонов и дейтронов при энергии столкновения до 27 ГэВ и светимости до 10^{32} cm $^{-2}$ s $^{-1}$.



Общая схема установки SPD.

Данные о событиях из SPD будут поступать со скоростью 3 МГц в виде тайм-слайсов в 10 мс, в каждом из которых будет происходить до 40 событий, т.е. один тайм-слайс будет содержать до 200 треков на одну станцию (при этом к хитам от треков добавится множество фейковых).

Требуется реконструкция событий из массива данных временных кадров. Для этого планируется разработать алгоритм для онлайн фильтра, чтобы обрабатывать не менее 100 тайм-слайсов в секунду



ST - Straw-Tracker. Его основной модуль состоит из 35 двойных слоев строу-трубок

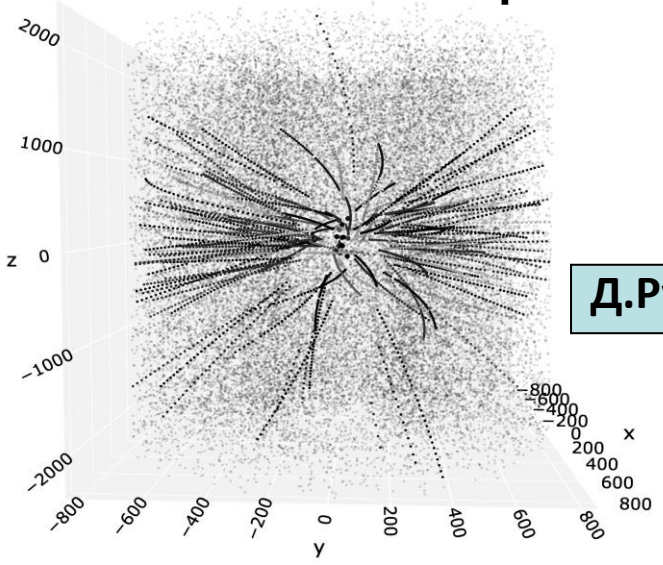
Глубокий трекинг для данных тайм-слайсов SPD NICA

Основные проблемы при трекинге SPD это, **огромное количество фейковых сигналов, пропуски отсчетов из-за неэффективности детекторов и “лево-право” неопределенность строу-трубок**

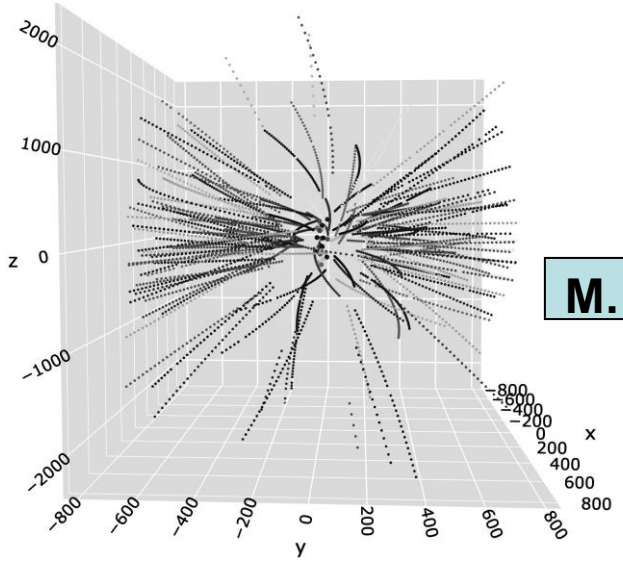
Внесение соответствующих усложнений в программу TrackNET неизбежно замедляет ее работу и снижает эффективность.

Реконструкция событий из массива данных тайм-слайса выполнялась в два этапа

1. Онлайн трекинг (TrackNET)

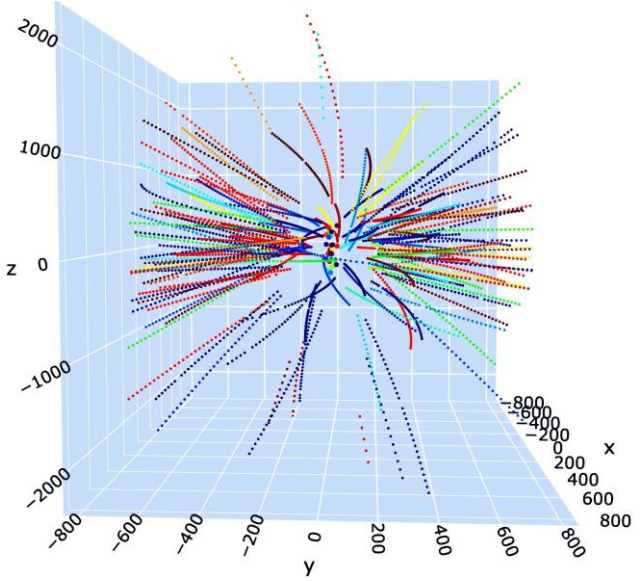


Д.Русов



М. Борисов

2. Распутывание массива треков по событиям



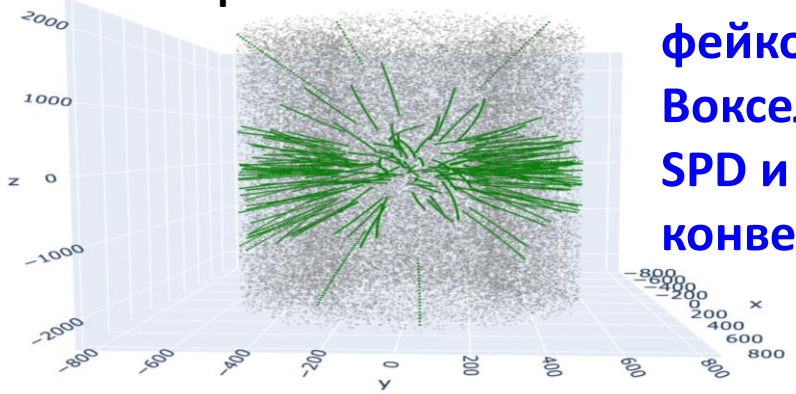
С помощью тонкой настройки TrackNET на суперкомпьютере ГОВОРУН достигнута скорость обработки ~ 2000 модельных событий в секунду при допустимой эффективности трекинга

Алгоритм распутывания событий основан на кластеризации векторов признаков, полученных с использованием сиамской нейросети. Результат вполне перспективен, но требует доработки из-за недостаточно высокой эффективности.

Новые подходы с применением нейросети Point cloud transformer (PCT)

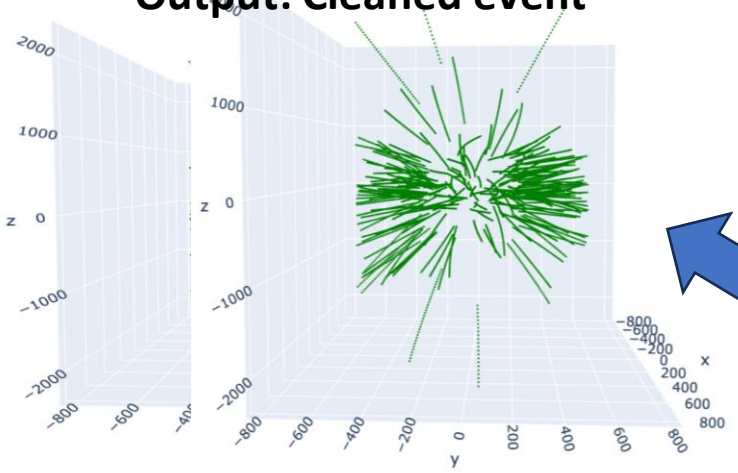
1. PCT для определения числа треков в событии.

Input: Raw event

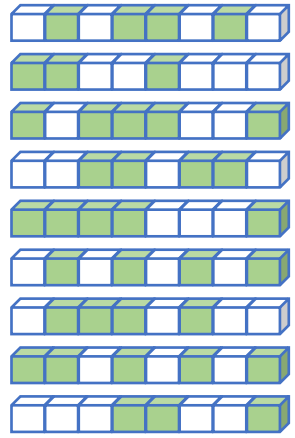


2.PCT для удаления фейков в тайм-слайсе. Вокселизация событий SPD и программный конвейер

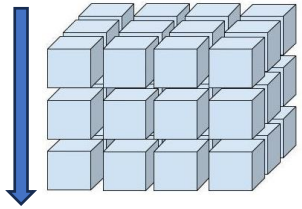
Output: Cleaned event

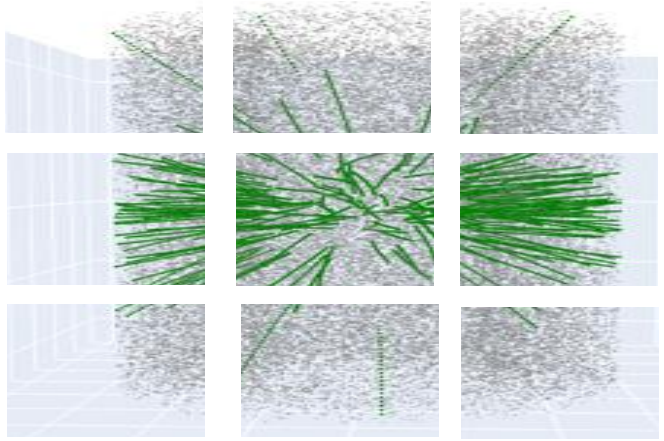


Объединяем все в одно событие

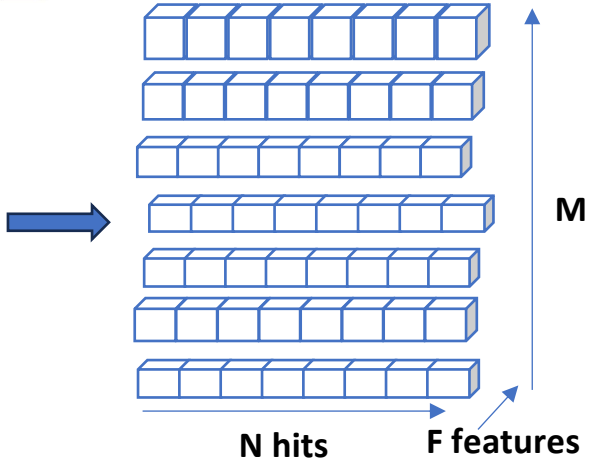


Делим пространство детектора на M вокселей, т.е. на более мелкие подпространства

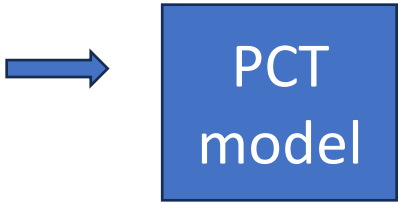




Берем хиты из каждого вокселя и формируем батч из M x N x F подвыборок, считая, что каждая подвыборка - это мини-событие



PCT= Point cloud transformer



Классифицируем хиты на истинные и фейки

РЕЗУЛЬТАТЫ ЧИСЛЕННОГО ЭСПЕРИМЕНТА

Число событий	Неэффек	Число вокселей	Precision	Recall
40	2%	512	0.98	0.98

Новое - графовая нейросеть с механизмом внимания

1. Как работала GNN проекта HEPTrkX https://arxiv.org/pdf/2003.11603

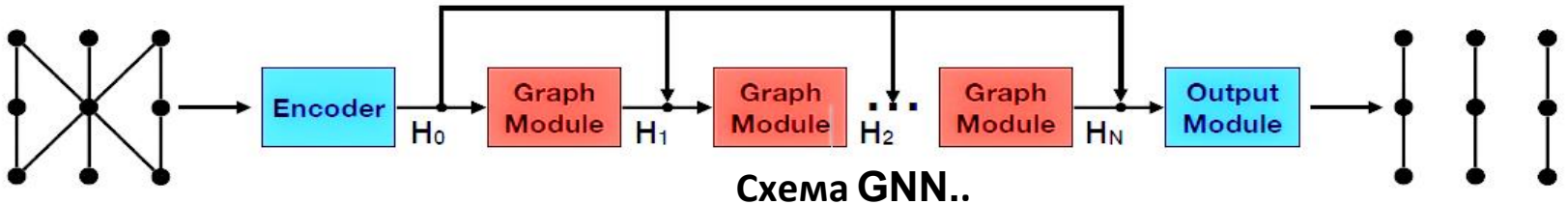


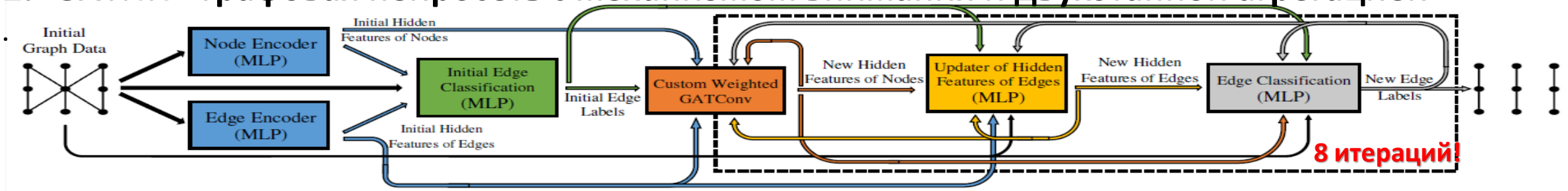
Схема GNN..

Архитектура GNN состоит из трёх двухслойных MLP.

Encoder вычисляет свойства узлов и ребер (H0). Graph module итеративно преобразует эти свойства, аггрегируя их с текущими Hi. После 8 итераций Output module берет последние скрытые признаки HN и выдает оценки классификации - веса для каждого ребра.

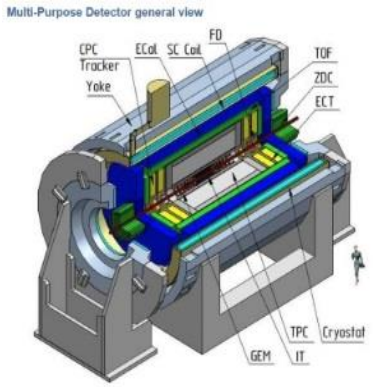
Однако, как показал опыт ЛИТ непосредственное применение GNN к данным BM@N и BES-III было невозможным без предварительного этапа фильтрации или значительных структурных изменений.

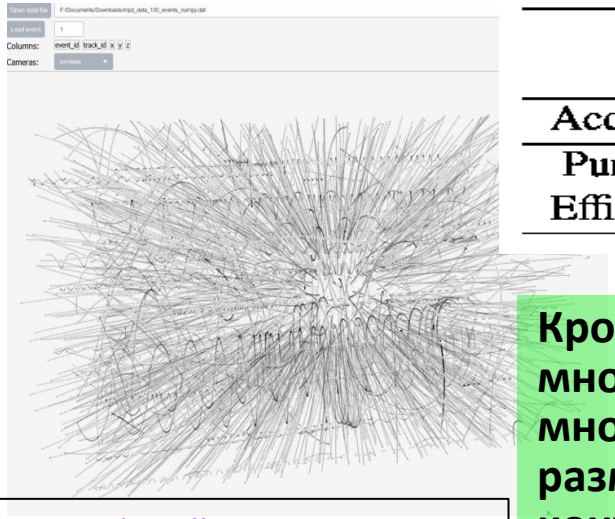
2. GATNN - графовая нейросеть с механизмом внимания и двухэтапной агрегацией



Архитектура GATNN: Два MLP Encoders узлов и рёбер графа события; начальный классификатор рёбер; графовый свёрточный слой - GATConv с вниманием для обновления скрытых признаков узлов; блок обновления скрытых признаков и меток рёбер. Механизм внимания позволяет динамически взвешивать вклады соседних узлов, а двухэтапная агрегация позволяет учитывать кривизну треков по информации от ±1 и ±2 слоёв детектора

1. GATNN для реконструкции треков в эксперименте MPD

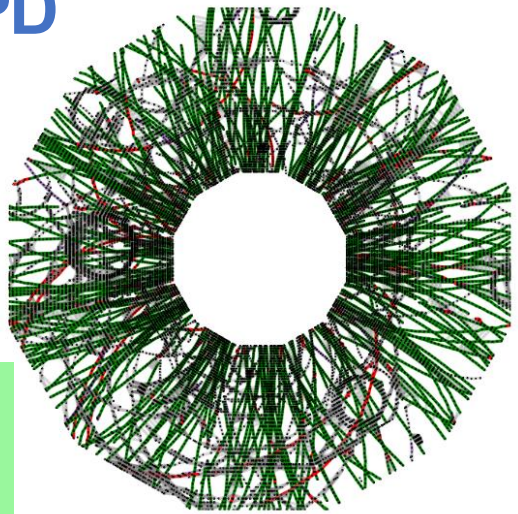




	Simple GNN	GNN with attention	Full GNN
Accuracy	95.9 %	96.1 %	**96.2 %**
Purity & Efficiency	91.8 %	92.2 %	**92.6 %**

Таблица с результатами реконструкции

Кроме того для размещения графа событий с множественностью 200 и больше удалось многократно уменьшить объем памяти для, размещения графа и разработать быстрый дата-контейнер



X-Y прооекция события с 200 треками

Датасет: 1000 Au-Au событий, p_t < 150 MeV
80% - обучение, 20% - тестирование

2. GATNN на втором этапе распутывания по событиям треков в тайм-слайсе SPD

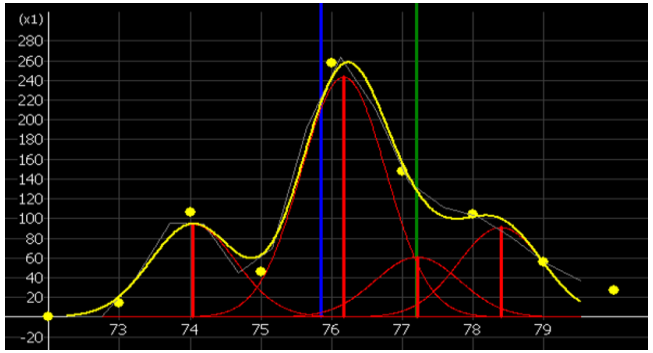
На 1-м этапе **TrackNET распознает треки всех событий в таймслайсе и выдает их в виде наборов хитов.**
Ключевая идея продолжения второго этапа – не просто преобразовать каждый трек в граф (узлы=хиты, ребра соединяют соседние хиты), но и закодировать все эти треки в виде узлов большого суперграфа, где ребра соединяют каждый узел со всеми другими узлами с метками 1 для истинных и 0 для ложных ребер.
При обучении GATNN происходит классификация ребер для определения положительных связей между треками из одного события, необходимое, чтобы разбить их по событиям - подграфам. К сожалению, при этом из-за ошибок сети могут появляться единичные ложные перемычки между подграфами, снижающие достигнутые 90% успешной классификации. В наших планах - разработка алгоритма обнаружения и устранения ложных ребер.

Нейросети Колмогорова-Арнольда для задач комплекса NICA

- Аппроксимационные свойства многослойных персептронов (МСП) обосновываются знаменитой теоремой Колмогорова и его ученика Арнольда: любая непрерывная функция нескольких переменных представима в виде суммы функций одной переменной.
- На основе этой теоремы В.И. Арнольд предложил совершенно новую концепцию обучения нейронных сетей, в которых в процессе обучения подстраивались не просто веса межнейронных связей, как в МСП, а сами активационные функции, определяющие выходной сигнал нейрона.
- Позже в начале 90-х В. Куркова доказала универсальную теорему аппроксимации для многослойных нейронных сетей, названных нейросетями Колмогорова-Арнольда (КАН).
- Однако их практическое применение КАН затянулось на три десятилетия в ожидании появления алгоритмов глубокого обучения и многоядерных компьютеров, позволяющих распараллеливать вычисления при обучении нейросетей. Только в 2024 году Лю и др. выпустили пакет Python для реализации КАН, использующих B-сплайны для представления обучаемых активационных функций и показали значительное повышение точности, достигнутое при применении KAN в задачах обработки физических данных, но столкнулись с проблемами при их обучении из-за застревания в локальных минимумах.
- Эти проблемы удалось недавно преодолеть в ЛИТ ОИЯИ путем усовершенствования КАН за счет замены функций активации с B-сплайнов (они не везде дифференцируемы и плохо распараллеливаются) на базисные функции в виде суперпозиции асимметричных супергауссовых компонент, а также путем инициализации их весов близкими к нулю, совместимой с оптимизатором Adam.
- Новая КАН была применена для решения задачи деконволюции многогауссовых сигналов и задачи подгонки трехмерного распределения магнитного поля в детекторе BM@N ускорителя NICA.

КАН подходы для задач комплекса NICA

Предложена, улучшена и реализована нейронная сеть Колмогорова-Арнольда (КАН), совместимая с оптимизаторами семейства Adam.

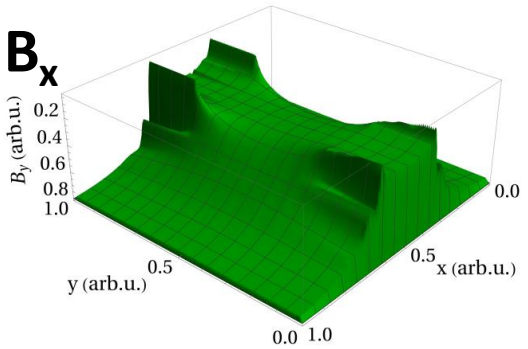


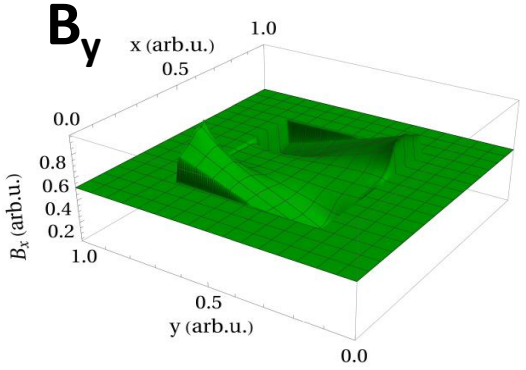
1. Разработанный подход был применен к задаче разрешение неизвестного числа перекрывающихся сигналов в полосковых трековых детекторов (типа CSC CMS) для обнаружения близко летящих частиц. Алгоритмы КАН показали точность > 90 %.

2. Разработанный подход может быть применен для реконструкции свойств частиц и струй, получаемых на ускорителе NICA.

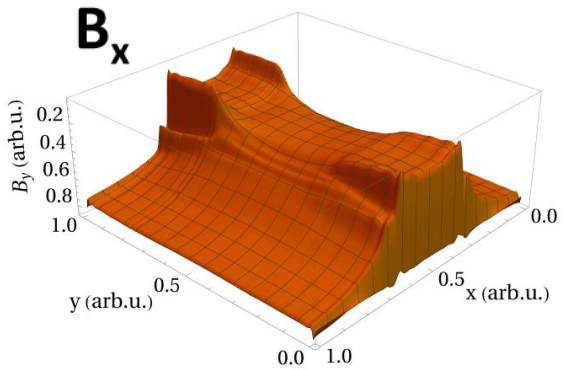
3. Выполнена быстрая аппроксимация распределения магнитного поля в детекторе BM@N

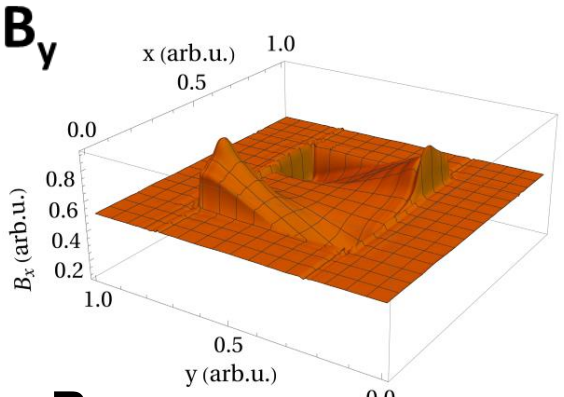
Интерполяция сплайном





Фитирование с помощью KAN fit





Значения поля аппроксимируются КАН со средней ошибкой 0,43% для B_x компоненты поля и 1% для компонента B_y.

Квантовый QUBO трекинг данных TrackML

После не вполне удачной попытки применить квантовый отжиг для данных SPD (см. *М.Буреш и др., Сообщ. ОИЯИ Р11-2024-5*) было решено усложнить схему трекинга, используя решение в режиме симуляции на кластере HybriLIT методом квадратичной бинарной оптимизации без ограничений (QUBO), но уже на более сложном хорошо известном датасете TrackML, опубликованном на платформе Kaggle.

Новизна состояла в использовании в качестве бинарных переменных триплетов хитов, а не только дублетов, а также квадруплетов путем объединения двух триплетов. Это позволило ослабить геометрические предположения о вершинах, запретить зигзагообразные квадруплеты и включить в формулировку QUBO более физические и геометрические аспекты, как например, учет кривизны треков и заряда частиц.

При вычислениях ипользовались библиотеки neal и qbsolv. Последняя позволил применить в качестве решателя современный квантовый отжиг D-Wave в режиме симуляции на кластере HybriLIT.

Из-за большого количества хитов в алгоритме трекинга вычисления удалось провести только для сокращенных вдвое полных событий (обычное для участников соревнования TrackML), что показало хорошую эффективность трекинга (Recall=97%. Precision=98%), но к сожалению, и многократную потерю времени на построение графа событий и манипуляций с триплетами, превышавшую на три порядка время самого трекинга.

В дальнейшем для применения данного подхода для эксперимента SPD NICA с учётом его специфики в отношении шума планируется разработка алгоритмов более качественной фильтрации сегментов, которая, возможно, уменьшит влияние шумовых хитов.

Итоги и перспективы

- **Применение методов машинного обучения было эффективным на всех стадиях развития систем обработки экспериментальных данных ФВЭ, прогрессируя вместе с развитием вычислительных технологий и алгоритмической базы.**
- **Радикальные проекты последних лет для экспериментов с высокой светимостью (HL-LHC) и NICA, ставят сложную проблему реконструкция треков частиц в плотных средах, для решения которой необходимо развитие новых алгоритмов глубокого трекинга и их распараллеливания на суперкомпьютерах.**
- **Помимо уже опробованных методов глубокого трекинга (TrackNet, GraphNet, Loot) следует отметить перспективность исследований по применению нейросетевых моделей трансформеров, позволяющих, в частности, эффективно отфильтровывать фейковые измерения и выполнять трекинг на сырых данных, минуя этап с получением хитов.**
- **В более далекой перспективе следует также уделять внимание методам квантового отжига в приложениях как к глобальному трекингу, так и локальным методам прослеживания, обобщающих алгоритмы фильтра Калмана.**
- **На волне успеха генеративных нейросетей типа CHatGPT-4 в создании картин и диссертаций следует отметить публикации об их успешном применении для симуляции взаимодействий в экспериментах ФВЭ**

Примерные темы ВКР на 2024-25 уч. год

1. **Разработка алгоритмов для реконструкции событий в трековых детекторах физики высоких энергий в условиях экспериментов с высокой светимостью.**
2. **Разделение и параметризация перекрывающихся сигналов на основе вейвлет-анализа и других аппроксимационных методов, в том числе КАН нейросетей**
3. **Разработка алгоритмов классификации изображений графовыми свёрточными нейронными сетями с механизмом внимания в задачах классификации частиц**
4. **Применение методов глубокого обучения нейросетевого классификатора в условиях сильного дисбаланса обучающей выборки на примерах конкретных задач ФВЭ или биологии**
5. **Применение информационных методов анализа социальных сетей для определения структуры малых социальных групп.**

Требования к студентам:

1. **Представление об искусственных нейронных сетях и теории графов**
2. **Умение программировать на Python и/или C++**
3. **Представление о пользовании библиотеками PyTorch и NumPy.**
4. **Знание английского языка, хотя бы на уровне беглого чтения.**

Для тех, кто выберет темы, связанные с ФВЭ, см. Трекинг в ФВЭ — Глоссарий.doc
на сайте https://gososkov.ru/u/UNI-DUBNA/Machine%20Learning/

Г.А.Ососков

Методы глубокого обучения как инструмент преодоления кризиса трекинга частиц в экспериментах на коллайдерах высокой светимости

Эту лекцию, файл с темами ВКР, учебники и курс из 6 лекций по глубокому обучению вы найдете на сайте

https://gososkov.ru/u/UNI-DUBNA/Machine%20Learning/

Спасибо за внимание!

email: gososkov@gmail.com
https://gososkov.ru/u/UNI-DUBNA/