Машинное обучение для трекинга в ФВЭ

Статус и перспективы работ группы Г.А.Ососкова

Геннадий Алексеевич Ососков

Гл.н.с. МЛИТ ОИЯИ

email: gososkov@gmail.com

https://gososkov.ru/u/UNIDUBNA/Machine%20Learning/

Трекинг ДО 2017.
Многообразие экспериментов, задач и методов

ARES,
NA-45,
Ионосфера,
EXCHARM,
HYPERON,
STAR,
CBM,
BM@N

Нейросети Хопфилда, роторные нейросети, клеточные автоматы, робастная подгонка, эластичный трекинг, вейвлет-анализ, фильтр Калмана, быстрое преобразование Хафа

Глубокий трекинг. Многообразие экспериментов, задач, глубоких нейронный сетей и исполнителей

BM@N,

BES-III,

SPD,

MPD?

Г.А. Ососков,
Павел Гончаров,
Настя Никольская,
Даниил Русов,
Максим Борисов,
Дима Стариков,
Иван Кадочников,
Мартин Буреш

TrackNetv1-3, GraphNet-RdgGraphNet, U-LOOT-Tracking- U-LOOT-Vertexing,
Transformers-Percivers, Квантовый отжиг

ARIADNE: PYTORCH LIBRARY FOR PARTICLE TRACK RECONSTRUCTION USING DEEP LEARNING

Pavel Goncharov, Egor Schavelev, Anastasia Nikolskaya, Daniil Rusov, Gennady Ososkov

Что имеем

1. **Группу объединяет только интерес к исследованиям, энтузиазм и желание видеть результаты работы**
2. **Результаты активной деятельности группы вполне удачны, опубликованы и перспективны**

Проблемы

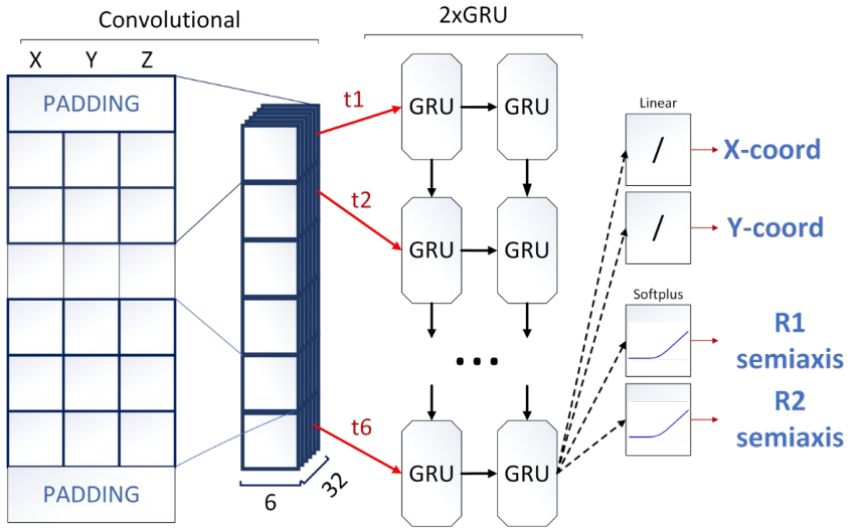
1. **Группа не объединена и не подкреплена административными рамками,**
2. **Нет уверенного обеспечения компьютерной базой ,**
3. **Исследования выполняются в духе "холодного сапожника" по спонтанным заказам лабораторий, без учета перспектив развития науки,**
4. **Группа непрерывно меняется по составу, т.к. выпускники не получают достойной оплаты**
5. **Неопределенность перспективы в университета «Дубна» дальнейшей качественной подготовке выпускников в области методов глубокого обучения, применяемых в экспериментальной физике.**
6. **Руководитель ждёт замену**

Всё это - не позволяет стать этой группе значимой силой в ОИЯИ, серьезно влияющей на широкое внедрение методов МО в повседневную практику всех лабораторий, без чего трудно ожидать радикального развития научных исследований.

Локальный и пособытийный подходы к трекингу

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 всей картине события

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.

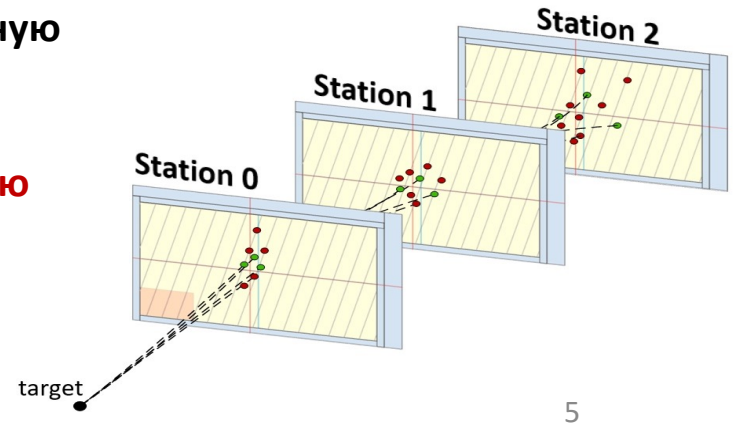


Scheme of the recurrent TrackNETv2 neural network

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

Однако гибкость конструкции RNN позволила нам преодолеть эти трудности и придумать новую сеть, которая объединяет оба этапа в одну сквозную 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 одну сквозную 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 его части подгонки трека

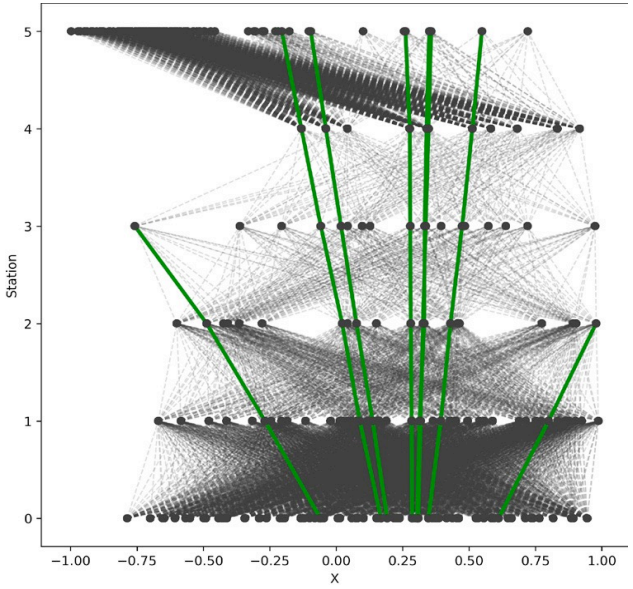


Пособытийный трекинг

Выделим три метода, когда распознавание треков среди шумов осуществляется сразу по всей картине события.

2.1. 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, от англ. networks) можно сформулировать как задачу классификации ребер графа – определить, какие из сегментов относятся к реальным трекам, а какие нужно отбросить, как ложные.

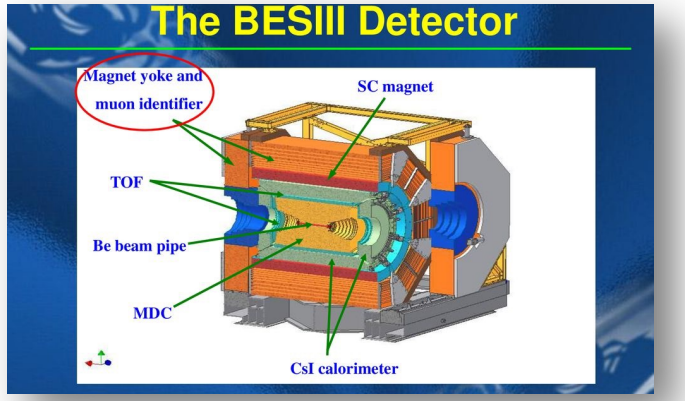


Графическое представление события C + C, 4 ГэВ эксперимента BM@N. Черные узлы и ребра соответствуют фейкам, зеленые узлы и ребра - найденным трекам

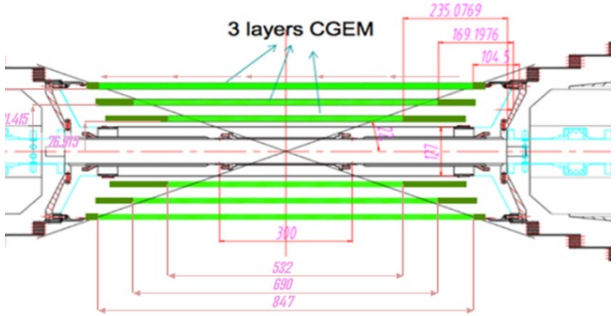
Эта схема похожа уже известный подход Денби-Петерсона с сегментной нейросетью Хопфилда, где нейросеть подолгу обучалась отдельно для каждого события, в то время как GNN, где надо найти те ребра, что являются сегментами реальных треков можно обучить на выборке из графов событий, где эти искомые ребра 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, что дало в итоге приемлемую эффективность трекинга

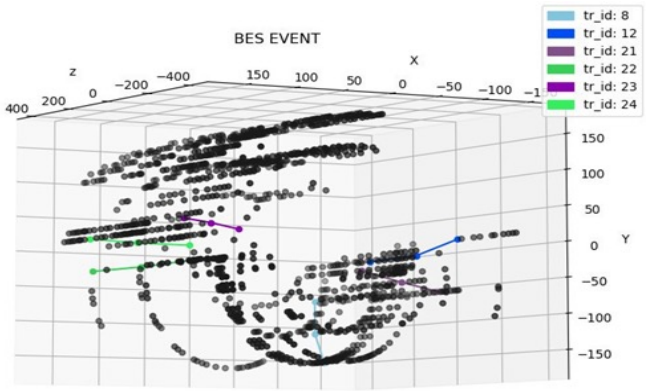
2.2. Применение графовых нейронных сетей, эксперимент BES-III



BESIII – коллайдерный эксперимент



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



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 несбалансирована. 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

Вершинный детектор BES-III имеет три цилиндрических станции типа GEM. Отсюда множества фейковых хитов, а также то , что пропуск одного хита их трех не даёт восстановить трек в магнитном без знания координата вершины.

Пособытийый подход LOOT, эксперимент BES-III

См. Goncharov et al http://ceur-ws.org/Vol-2507/130-134-paper-22.pdf

Событие, как 3D изображение в CNN.

- **В CNN Изображения имеют формат 3d: высота + ширина + RGB;**
- **У нас данные с каждой станции - разреженная матрица нулей и единиц, где единицы указывают на появление хитов;**
- **События также имеют формат 3D: Высота + Ширина + Станции.**

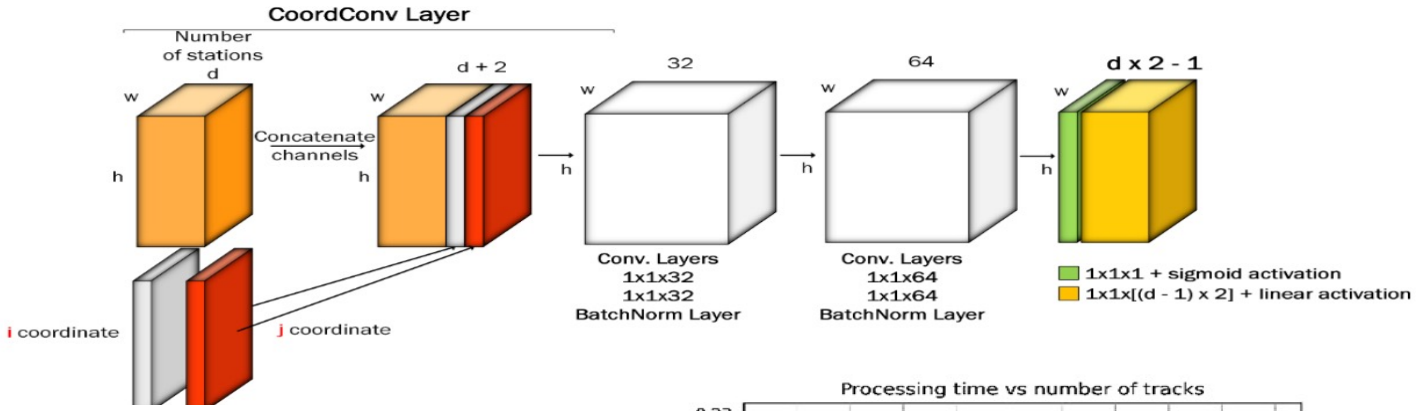
Высота и Ширина - это размеры самой большой из станций (обычно это последняя).

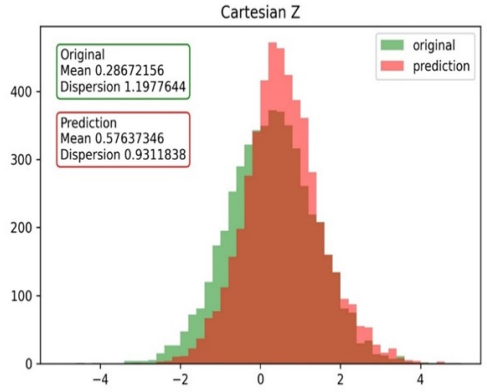
Наша основная идея – использовать размер OZ вместо RGB каналов.

Это радикально новый подход, позволяющий найти координаты вершины события

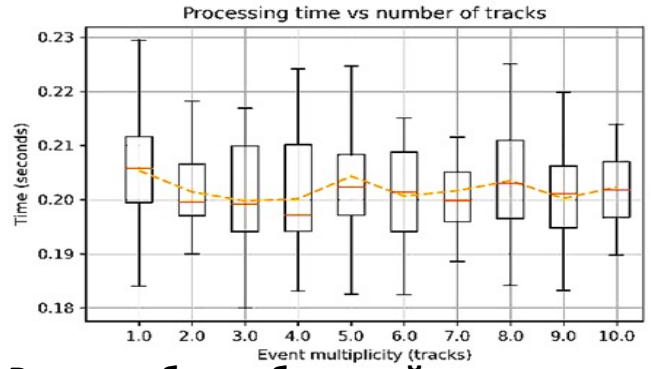
Используется новая нейросетевая модель **Look Once On Tracks (LOOT)**

Поскольку обычные сверточные нейросети не могут при обучении научиться находить координаты из входных данных, их подают на вход и преобразуют потом в индексы ячеек. Сеть обучается предсказывать продолжения треков на следующие слои с помощью процедуры сдвигов





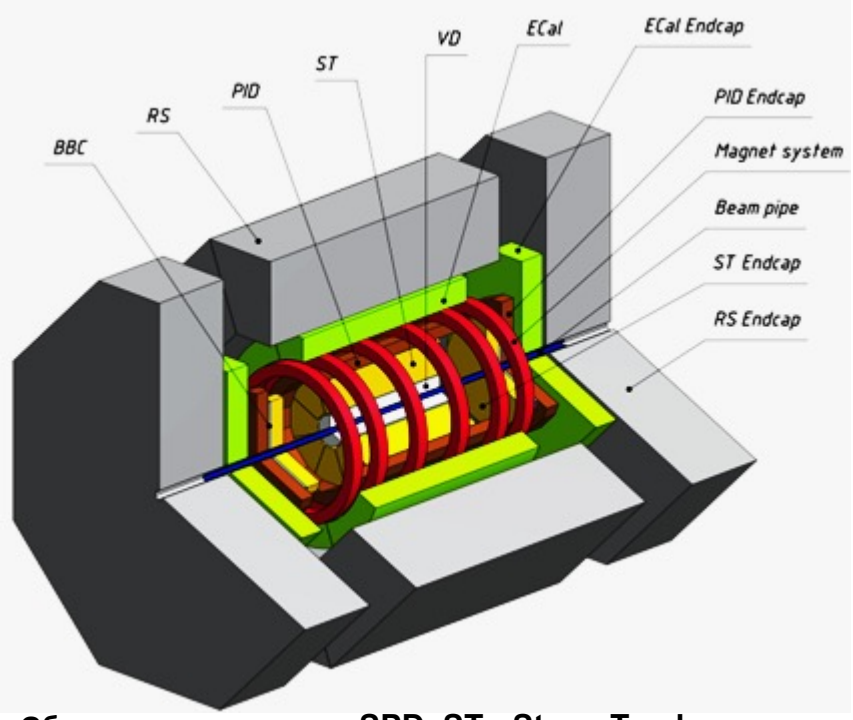
Хотя на модельных данных без фейков результаты были хорошие, учет проблем с фейками потребовал введения новой архитектуры U-Net. В результате работы модель после обучения предсказывает координату Z первичной вершины события с приемлемой среднеквадратичной ошибкой в 1 см



Время работы обученной модели не зависит от множественности события

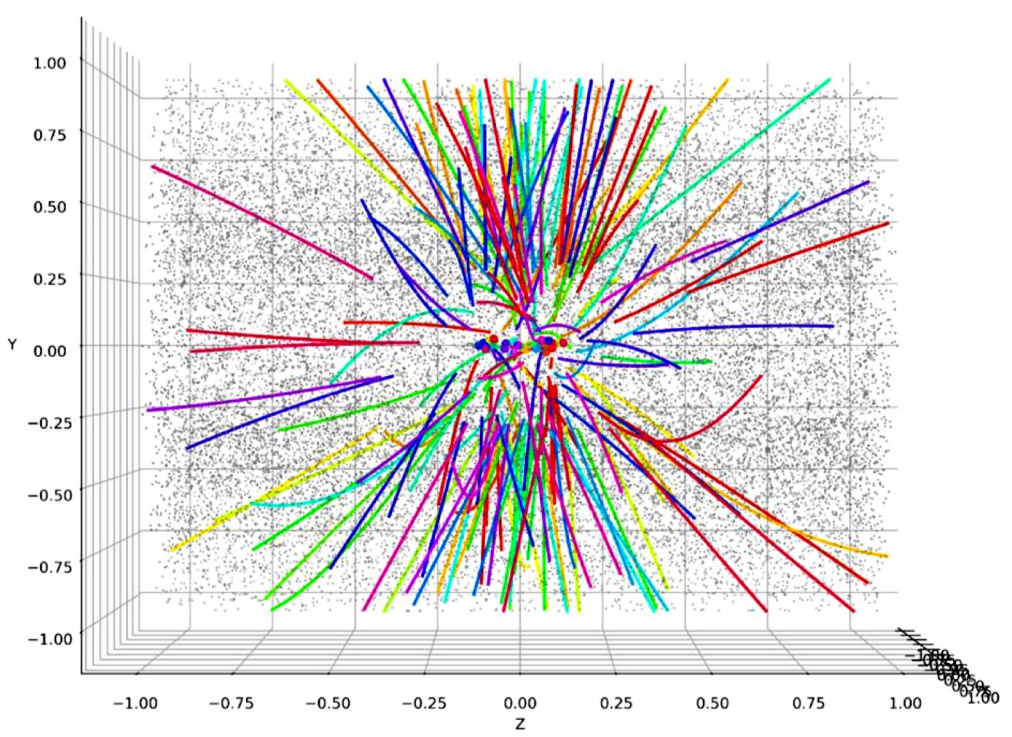
Трекинг для данных экспериментов высокой светимости. SPD NICA

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 $^{-1}$. Данные о событиях из SPD будут поступать со скоростью 3 МГц в виде тайм-слайсов в 10 мс, в каждом из которых будет происходить в среднем 40 событий, т.е. один тайм-слайс будет содержать в среднем 200 треков и 1100 хитов на одну станцию (причем 82,26% всех хитов являются фейками). Планируется разработать алгоритм для онлайн фильтра, чтобы обрабатывать не менее 100 тайм-слайсов в секунду.



Общая схема установки SPD. ST - Straw-Trecker. Его основной модуль состоит из 31 двойного слоя строу-трубок

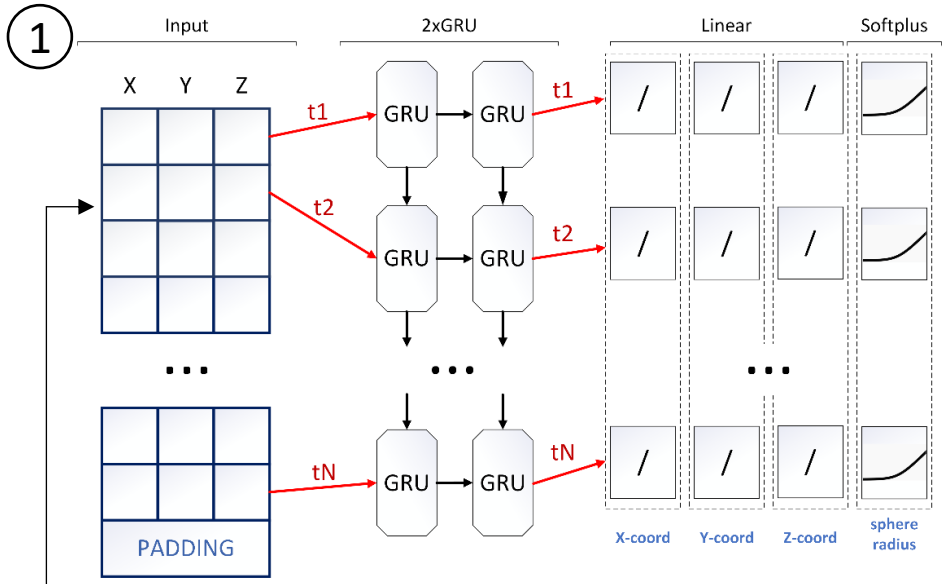
Основные проблемы при трекинге это “лево-право” неопределенность строу-трубок, огромное количество фейковых сигналов и пропуски отсчетов из-за неэффективности детекторов.
Внесение соответствующих усложнений в программу TrackNET неизбежно замедляет ее работу и снижает эффективность



Пример тайм-слайса в эксперименте SPD. Треки показаны цветными линиями, их первичные вершины – точками соответствующего цвета. Фейковые хиты показаны серыми точками.

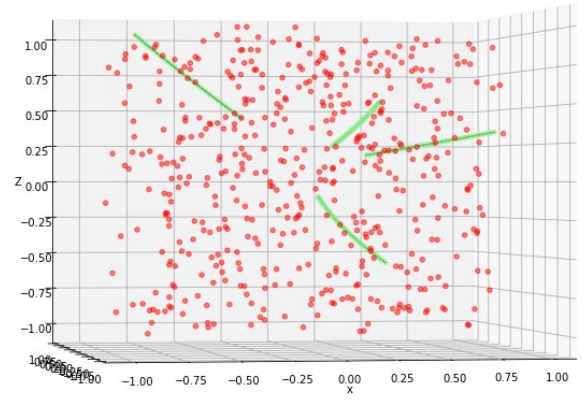
SPD тайм-слайс Tracking with TrackNET. Inference optimisation

Доложено Д.Русовым на SPD митинге



Testing setup:

- 25 000 generated events (625 time slices)
- Xeon(R) Gold 6148 CPU @ 2.40GHz
- Single Nvidia V100 32Gb GPU



3 Search for the sphere centers (x, y, z)

4 1 nearest hit with distance to each sphere center

2 Sphere predictions (x, y, z, r)

5 Check sphere attendance

Prolong candidates and pass them to the model 6

Впечатляющие результаты удалось показать Д.Русову при прогоне 25 000 модельных событий SPD, представляющих 625 тайм-слайсов с 40 событиями в каждом, на суперкомпьютере ГОВОРУН (Xeon(R) Gold 6148 CPU @ 2.40GHz, Single Nvidia V100 32Gb GPU). Была достигнута скорость обработки ~ 1800 событий в секунду

Трансфо́рмер — **архитектура глубоких нейронных сетей, позволяющая сочетать в себе преимущества как сверточных (CNN), так и рекуррентных нейронных сетей (RNN). Трансформеры предназначены для обработки последовательностей, таких 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.

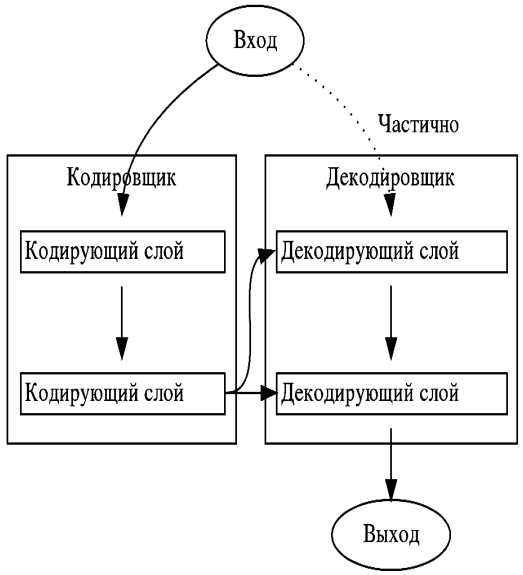
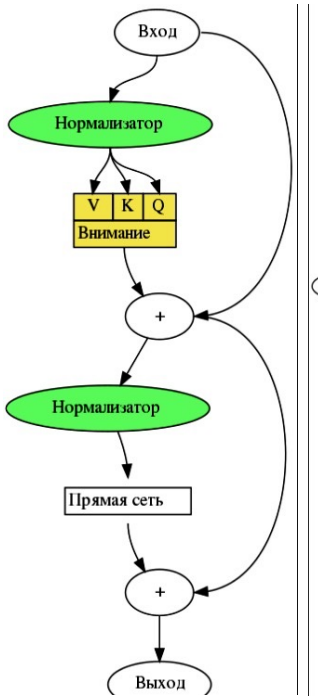
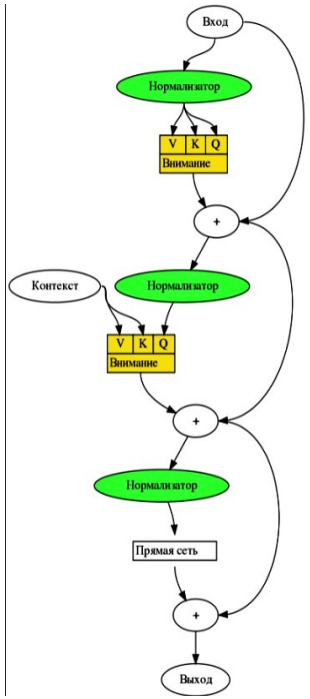


Схема трансформера



Кодировщик



Декодер

Кодировщик получает на вход векторизованую последовательность и состоит из слоя самовнимания (вход из предыдущего слоя) с последующими слоями МСП.
Декодировщик состоит из аналогичных слоев.
Таких слоев и в кодировщике и в декодировщике может быть много в зависимости от решаемой задачи.
Механизм attention повышает вес соответствия одного слова другому в предложении, если они чем-то близки .

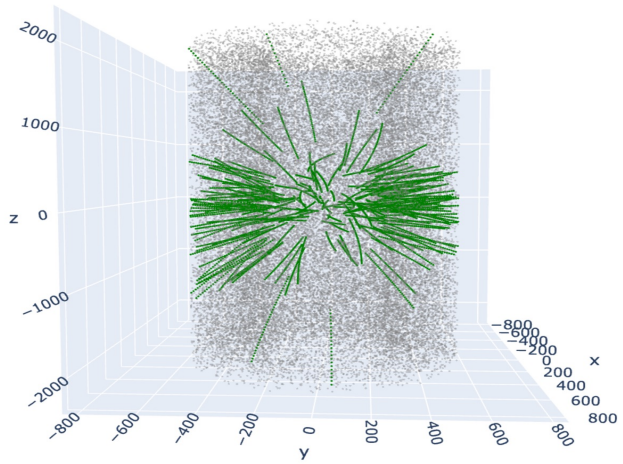
Механизм внимания параметризован матрицами весов запросов W_Q, весов ключей W_K , весов значений W_v. Внимание входного вектора X к вектору Y, вычисляются по формуле:

$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^{\mathrm{T}}}{\sqrt{d_k}}\right)V$$

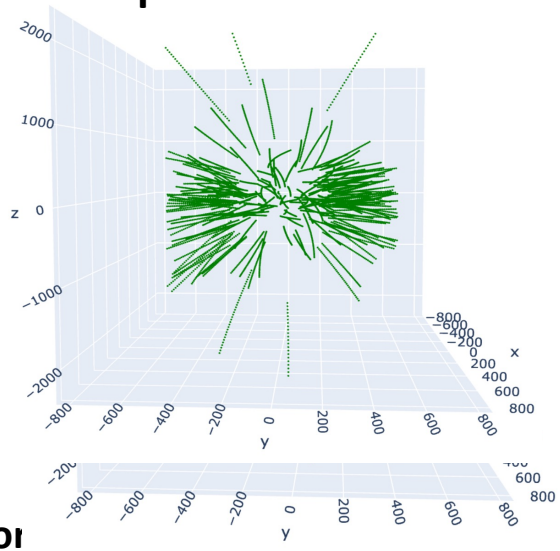
где вектора $Q= W_qX$, $K=W_kX$, $V=W_vY$ вычислены, как скалярные произведения. В отличие от RNN, трансформеры не требуют обработки последовательностей по порядку. Например, если входные данные — это текст, то трансформеру не требуется обрабатывать конец текста после обработки его начала. Благодаря этому трансформеры распараллеливаются легче, чем RNN, и могут быть быстрее обучены.

SPD event segmentation (voxelization) pipeline

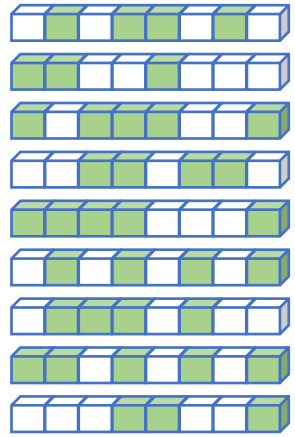
Input: Raw event



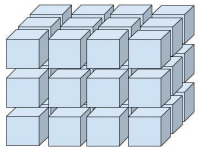
Output: Cleaned event



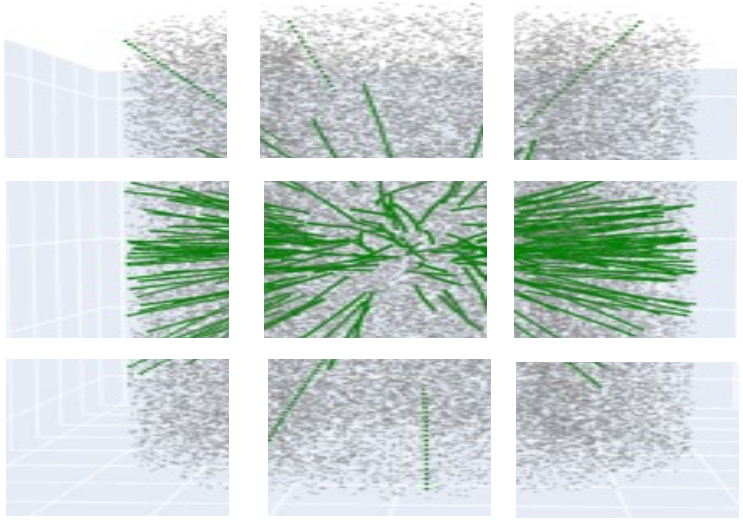
Combine into one event

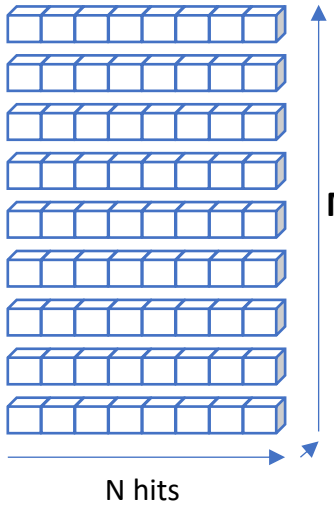


Divide the detector space into M voxels, i.e. smaller subspaces



Take hits from each voxel and for batch of M x N x F samples pretending that each sample is a mini-event





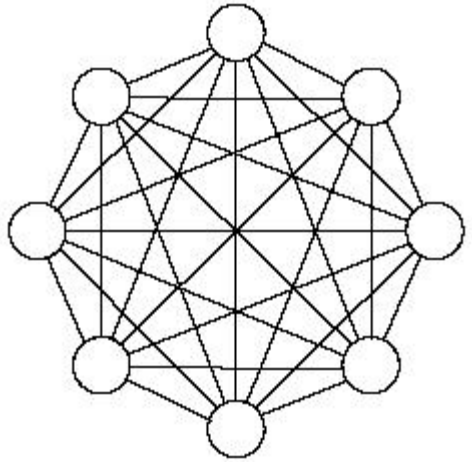
PCT= Point cloud transformer

PCT model

Classify hits on true and fakes

Experimental results

Events per time slice	Number of parts	Precision	Recall	Speed (events/sec)
1	16	0.96	0.98	0.06
1	32	0.97	0.97	0.7
1	64	0.96	0.99	0.07
2	64	0.97	0.95	0.17 (0.34)
5	64	0.93	0.97	0.27 (1.37)
10	64	0.90	0.97	0.59 (5.96)



Нейронная сеть 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

$$E(s) = -\tfrac{1}{2} \Sigma_{ij} s_i w_{ij} s_j.$$

Теорема Хопфилда: в результате эволюции $E(s)$ убывает в локальные минимумы, соответствующие точкам стабильности сети.

Для нахождения глобального минимума E сеть термализуется.

В соответствии с теорией среднего поля состояния нейронов $v_i = \langle s_i \rangle_T$ усредняются по температуре T. Эволюция сети определяется уравнением динамики среднего поля: $v_i = 1/2(1+\tanh(-\partial E/\partial v_i \, 1/T) = 1/2(1+\tanh(H_i/T))$,

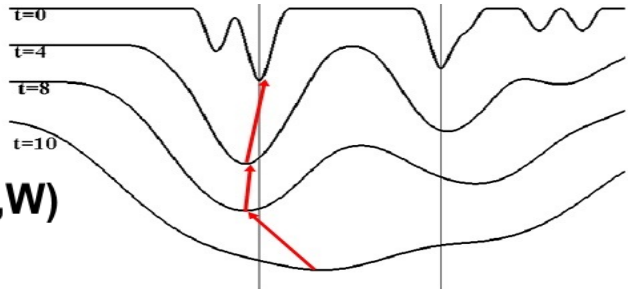
где $H_i = \langle \Sigma_j w_{ij} s_j \rangle_T$ – локальное среднее поле нейрона.

Значения v_i , переставшие быть целочисленными, определяют уровень активности нейрона, т.е. в случае $v_i > v_{min}$ нейрон считается активным.

Температура убывает по схеме «имитационного отжига» (simulated annealing).

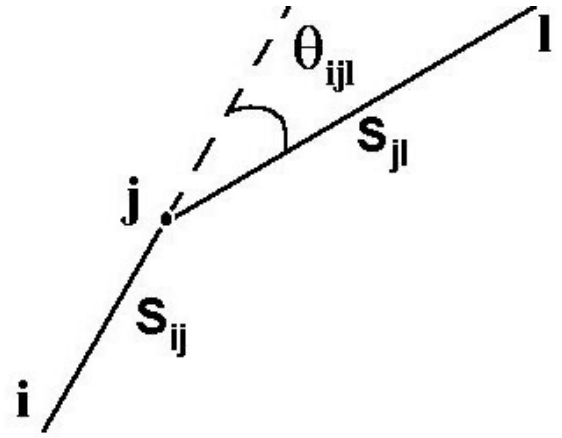
$$g(u) = \frac{1}{1+e^{-\lambda u}}$$

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



Распознавание треков. Метод сегментов.

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



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}

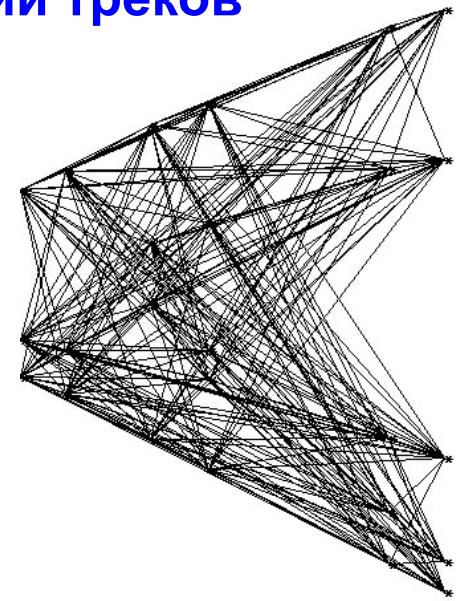
Вводится нейрон s_{ij} как направленный сегмент, соединяющий точки i, j ..

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

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

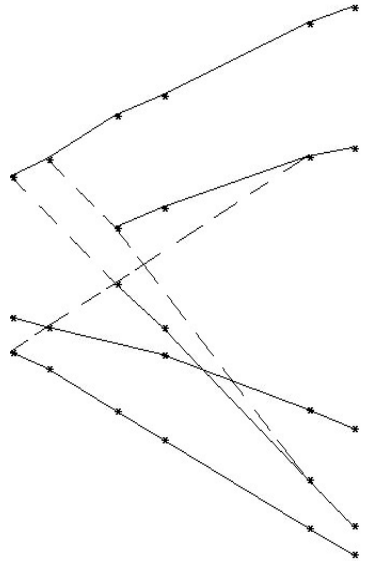
Пример применения для распознавания событий с короткоживущими частицами

Эксперимент EXCHARM - проблема: разрешить бифуркации, но не допустить массовых ветвлений треков



на нулевой итерации

всего 244 нейрона



на 30-ой итерации

$V_{ij} > 0.5$ у 26 нейронов

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

Однако чрезмерная чувствительность к шумам и такие недостатки применения полносвязных нейросетей, как слишком медленная сходимость и то, что не учитывается известное уравнение движения частицы в магнитном поле, потребовало поиска новых подходов к проблеме трекинга с применением глубоких нейросетей