



1956

Секция Международной конференции «Математика в созвездии наук», приуроченная к юбилею академика В.А. Садовниченко. ОИЯИ, МЛИТ, 2 апр. 2024 г



Методы машинного обучения и интеллектуального анализа экспериментальных данных. История, проблемы, перспективы

Ососков Геннадий Алексеевич

Окончил МГУ с отличием в 1953 г.

Работает в Объединенном институте ядерных исследований с 1961 г.

Главный научный сотрудник Лаборатории информационных технологий им. М.Г.Мещерякова

Профессор университета «Дубна»

email: gososkov@gmail.com

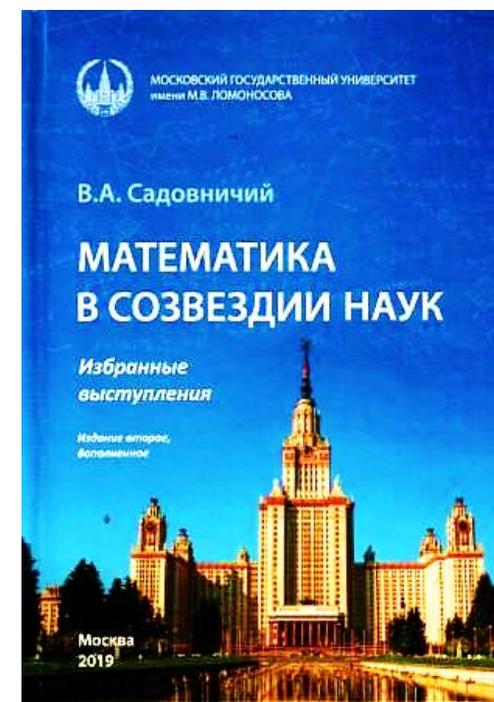
<http://gososkov.ru>

«Математика в созвездии наук»!

Международная конференция, 1-2 апр. 2024 г.

Среди основных направлений работ:

- Математические методы искусственного интеллекта.
- Математическая теория управления и оптимизация.
- Математическое обеспечение ЭВМ.
- Суперкомпьютерные вычисления.
- Искусственный интеллект и большие языковые модели.
- Математика в фундаментальных и прикладных космических исследованиях.
- Моделирование и прогнозирование глобальных процессов.
- Актуальные проблемы преподавания математики в школе и в вузе.



Виктор Антонович
Садовничай



01.04.2024

Можно вспомнить:

Михаил Васильевич Ломоносов :

«Математика – царица всех наук,
но служанка физики»



Карл Фридрих
Гаусс (1777-1855)

2

Эволюция экспериментальной техники, измерительных и вычислительных средств ДО эры электронных измерений

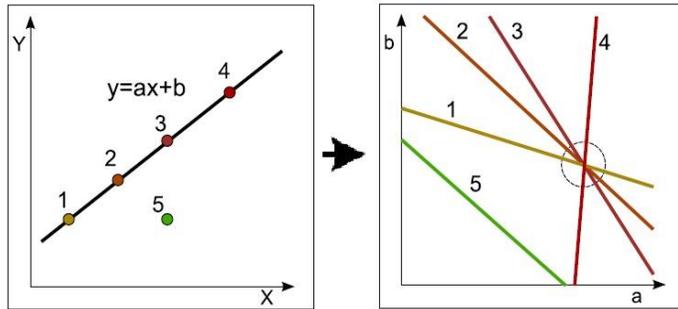
- Пузырьковые камеры с фотосъемом и ручными измерениями
- Измерительные полуавтоматы и просмотрные столы
- Сканирующие автоматы: HPD, Spiral Reader, Sweepnik, АЭЛТ-2
- В ЦЕРНе IBM-360/44 потом CDC-6600, программирование на языке ФОРТРАН
- В ОИЯИ ламповые КИЕВ, М-20, Урал-2, полупроводниковая Минск-2 , программы в машинном коде

Возникшие тогда математические проблемы остаются еще актуальными

- Применение теории массового обслуживания для расчета буферной памяти при импульсной загрузке
- Обеспечение программ моделирования физических процессов генераторами случайных чисел с заданными распределениями
- Распознавание гладких линий в шумовом фоне
- Отсев выпадающих измерений при подгонке кривых
- Математический аппарат калибровочных преобразований
- Разделение близких перекрывающихся сигналов
- Нахождение малых резонансных пиков на большой шумовой подложке

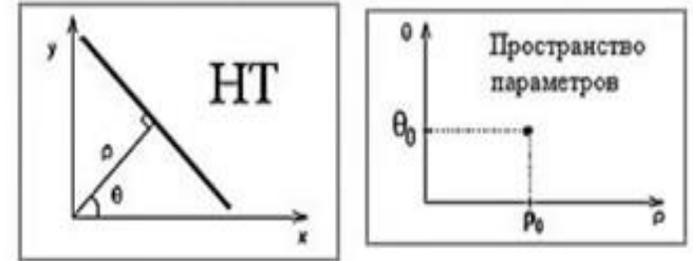
Преобразования Радона-Хафа (Hough Transform) для поиска линий

При этом точки прямых или окружностей из пространства координат преобразуются в точки в пространстве параметров кривой



Прямая $y=ax+b$ и её вид в пространстве параметров (ab).

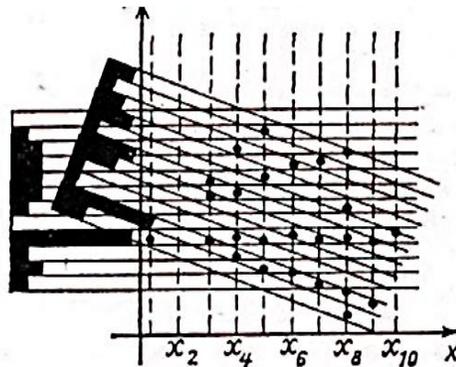
Аналогично для окружности $(x-x_0)^2+(y-y_0)^2=R^2$ получаем точку (X_0, Y_0, R) в трехмерном пространстве параметров



$$x \cos \theta + y \sin \theta = \rho$$

Физик П.С. Хаф использовал параметры (ρ, θ) , чтобы избежать обращения в бесконечность параметра a в уравнении $y=ax+b$ при приближении прямой к вертикали.

В применениях, гораздо практичнее оказался метод поворотных гистограмм.



Поворотные гистограммы

Суть метода состоит в фрагментации исследуемого региона на узкие параллельные полосы, в каждой из которых подсчитывается количество попаданий. Меняя угол наклона полос α от α_{\min} до α_{\max} , находим максимум в гистограмме, соответствующий группе точек лежащих близко к прямой. Критерий обнаружения линии, - максимум должен превысить заданный порог

Задача – найти 20 точек, лежащих вблизи прямой, среди множества из 200 других точек

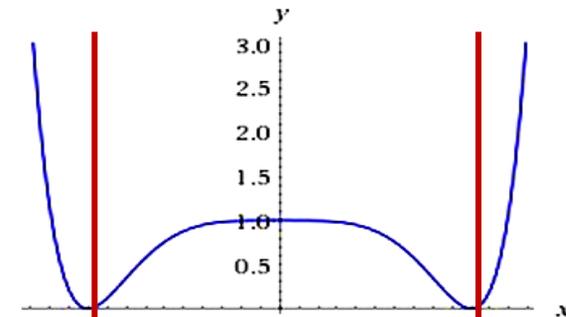
Робастная регрессия

Это специальный итеративный вариант метода наименьших квадратов, где вместо постоянных весов $w=1/\sigma^2$ вводится весовая функция, учитывающая расстояние от i -й точки до подгоняемой кривой. Для прямой это будет

$$\sum_i w_i (y_i - ax_i - b)^2 = \sum_i w_i e_i^2 \rightarrow \min_{a,b},$$

В качестве весовой функции чаще всего используют бивеса Тьюки

$$w(e) = \begin{cases} \left(1 - \left(\frac{e}{c_T \sigma}\right)^2\right)^2, & \text{если } |e| < c_T \sigma, \\ 0 & \text{в остальных случаях} \end{cases}, \quad \text{где обычно } c_T = 3 \div 5.$$



Итерации начинаются с постоянного веса при σ , равному среднеквадратичному разбросу всех точек, но потом значение подлежит перевычислению на каждой k -й итерации

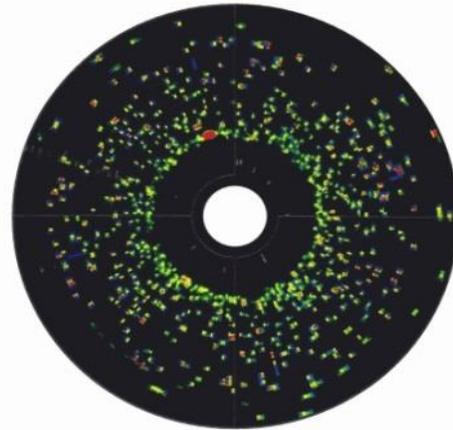
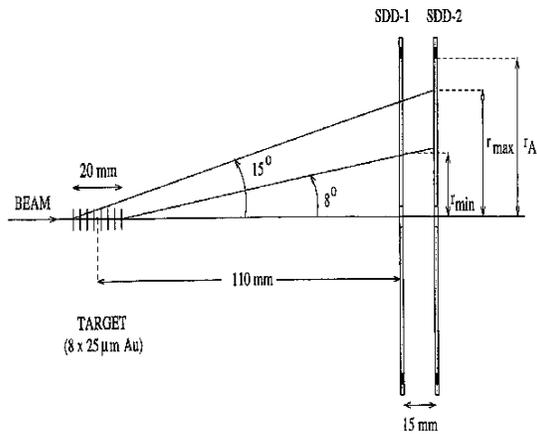
по формуле

$$\sigma^{(k)} = \sqrt{\frac{\sum_i w_i^{(k-1)} (e_i^{(k-1)})^2}{\sum_i w_i^{(k-1)}}}.$$

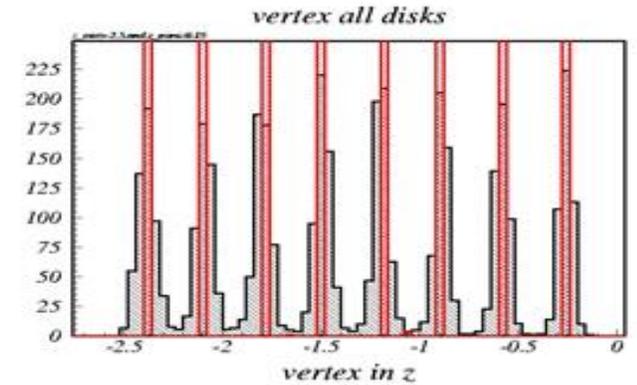
Коридор $c_T \sigma$ быстро сужается, так что подгонка сходится за 4-5 итераций

Примеры применения методов робастной подгонки

1. NA-45. Определение координат вершин событий по только двум многократно измеренным точкам



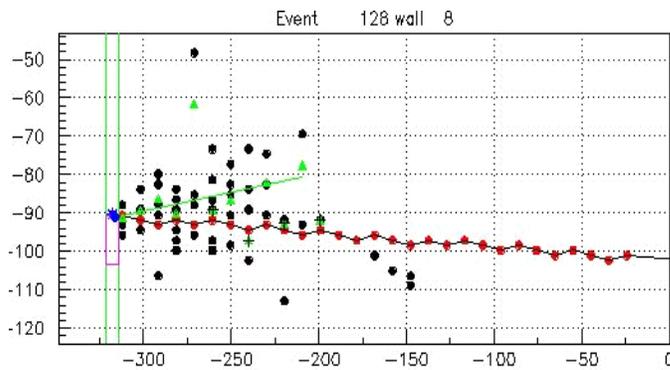
Мишень состояла из восьми 25-мк золотых дисков. 700 трековых событий в узком угловом приеме и большое количество шумовых отсчетов не позволили распознать отдельные треки. Однако метод робастной подгонки сходил за всего за пять итераций, хотя за начальную аппроксимацию была грубо взята середина всей области мишени.



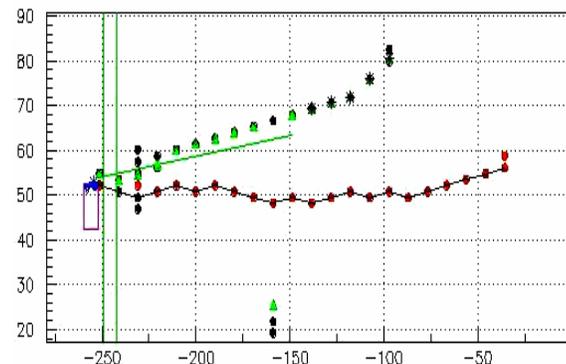
NA-45 координатный детектор

Один из двух силиконовых дисков

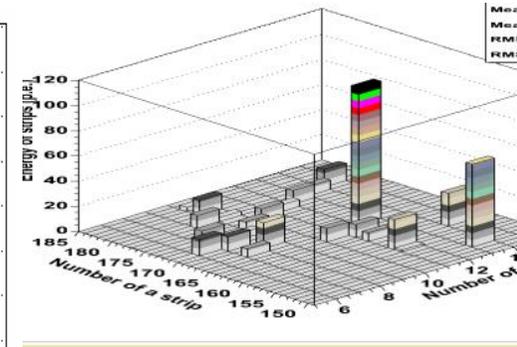
2. Орега. Робастная подгонка для адронных ливней и мюонных треков с 2D весами, зависящими не только от расстояний, но и амплитуд трековых хитов



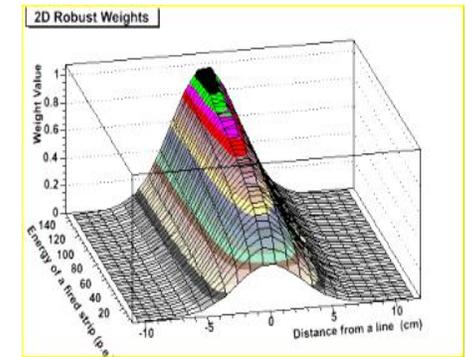
Адронный ливень



1D мюонные треки



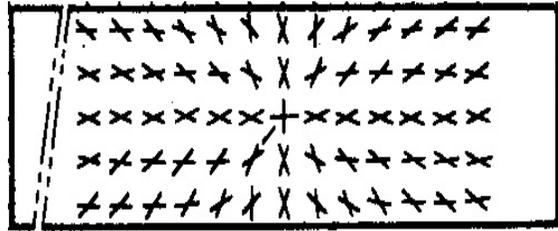
2D мюонные треки



Двумерная весовая функция

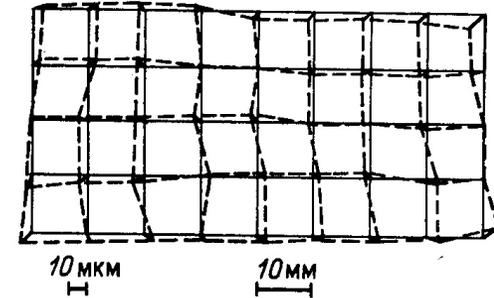
Применения математических методов для коррекции аппаратных измерений

1. Калибровка сканирующего автомата «Спиральный измеритель»



reference plate

Проводилось сканирование калибровочных измерений эталонной пластины со специально расположенными крестами. Для распознавания крестов использовался **метод распознавания образов**. Их центры вычислялись и сравнивались с идеальными положениями для построения карты остатков, с помощью которой строилось **калибровочное преобразование**, устраняющее оптические искажения при измерениях.



10 мкм 10 мм

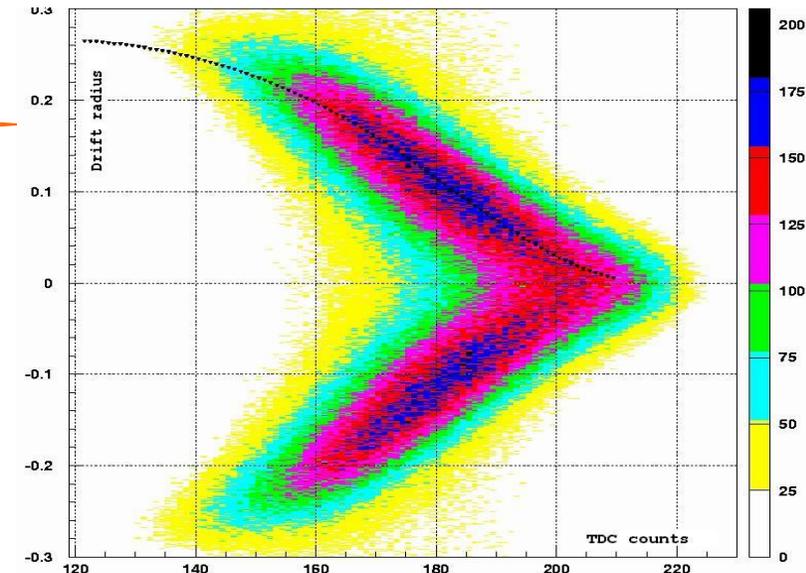
map of residuals

2. Калибровка дрейфовых камер

Пример калибровки дрейфовой камеры HERA-B OTR:  Время дрейфа, измеренное в отсчетах TDC (время-цифровой преобразователь), следует перевести в радиусы дрейфа для получения калибровочной функции $r(t)$.

$r(t)$ была получена путем **робастной подгонки** кубических сплайнов непосредственно к 2D гистограмме отсчетов TDC со многими тысячами бинов

drift radii versus TDC counts



3. Вейвлет-анализ

Одномерное вейвлет-преобразование сигнала $f(x)$ является **двумерной функцией частоты и времени**,

$$W_{\Psi}(a, b) f = \frac{1}{\sqrt{C_{\Psi}}} \int_{-\infty}^{\infty} \frac{1}{\sqrt{|a|}} \Psi\left(\frac{b-x}{a}\right) f(x) dx,$$

где функция Ψ - вейвлет, b - сдвиг (shift), a – масштаб (scale). Условие $C_{\Psi} < \infty$ обеспечивает существование Ψ и обратного вейвлет-преобразования. Произвол в выборе Ψ позволил предложить много разных типов вейвлетов. Семейство **непрерывных вейвлетов** можно получить как вторую производную гауссиана

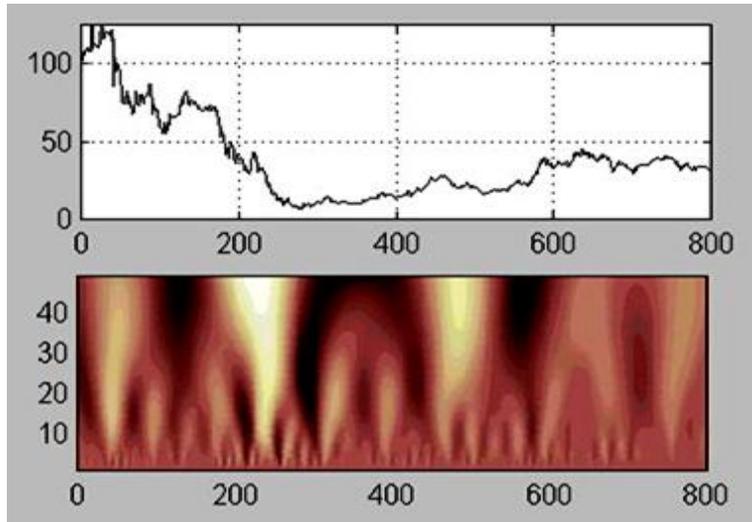
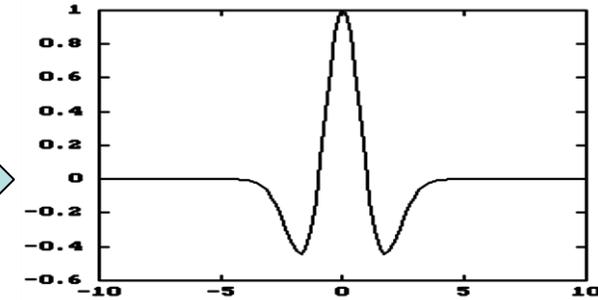
Полученный **гауссов вейвлет** $g_2(x) = (1 - x^2)e^{-\frac{x^2}{2}}$ стал известен, как

“Мексиканская шляпа”

На этом примере видно, что разные вертикальные уровни вейвлет-спектра соответствуют разным частотам сигнала, а по горизонтали видно в какое время сигнал изменял свое поведение.

Непрерывные вейвлеты очень **устойчивы к шумам**, однако они **не ортогональны**. поэтому после обратного преобразования

происходят **недопустимые искажения сигнала**. Кроме того, реальные сигналы, подлежащие компьютерному анализу, всегда дискретны. **Поэтому непрерывные вейвлеты практически не используют, а применяют ортогональные дискретные вейвлеты**.



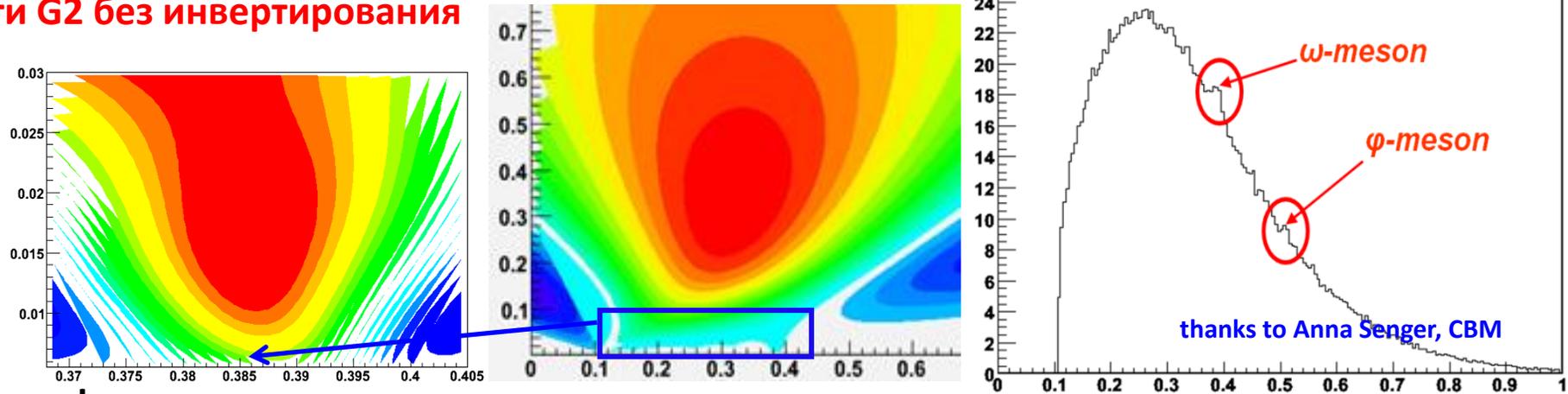
Непрерывные вейвлет-преобразования

это путь к вейвлет- домену, где проблемы МО могут быть решены гораздо проще
они нужны для работы с инвариантными масс-спектрами, когда отношение $S/B \ll 1$

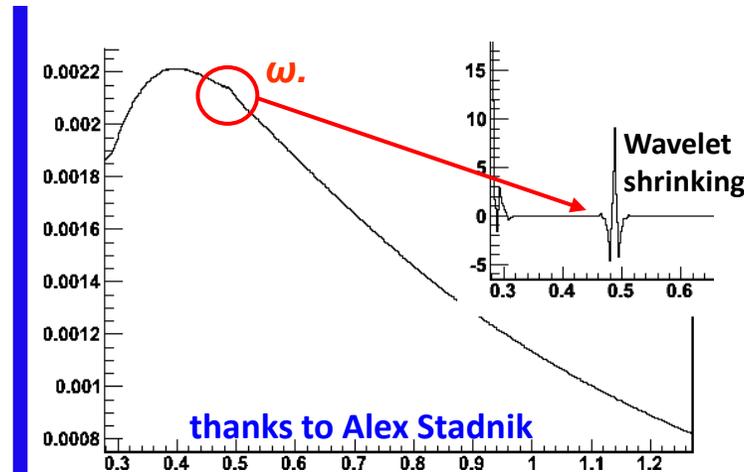
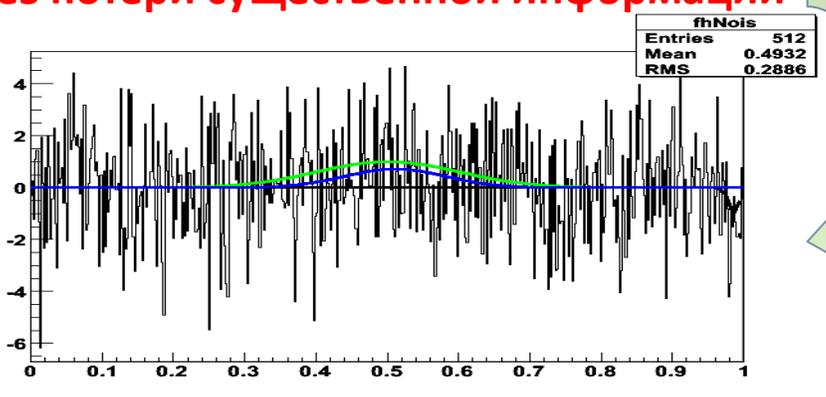
Непрерывный вейвлет G_2 преобразует гауссиан $g(x;A,x_0,\sigma)$ в вейвлет того же порядка, но с параметрами этого гауссиана. Это справедливо для любого порядка n и позволяет находить параметры пика непосредственно в области G_2 без инвертирования

$$W_{G_2}(a, b)g = \frac{Aa^{5/2}\sigma}{(a^2+\sigma^2)^{3/2}} G_2\left(\frac{b-x_0}{\sqrt{a^2+b^2}}\right)$$

1. Оценка параметров пиков по спектру инвариантных масс



2. Сглаживание после вычитания фона без потери существенной информации

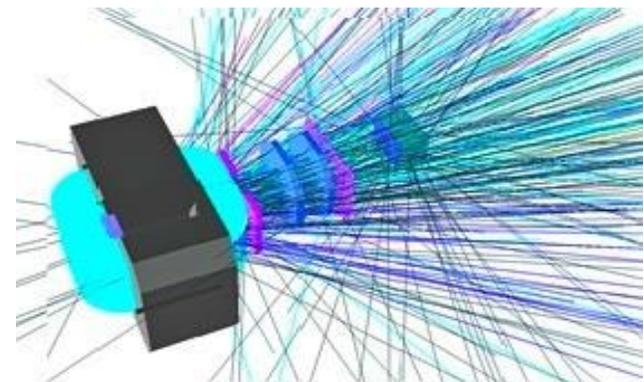


3. Дискретные вейвлеты

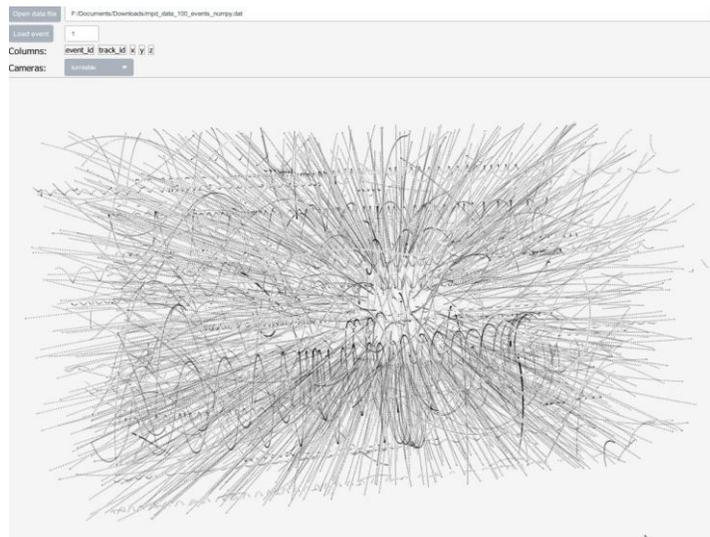
Применяют для выявления резонанса даже в присутствии массивного фона



Современные эксперименты с электронным съемом данных

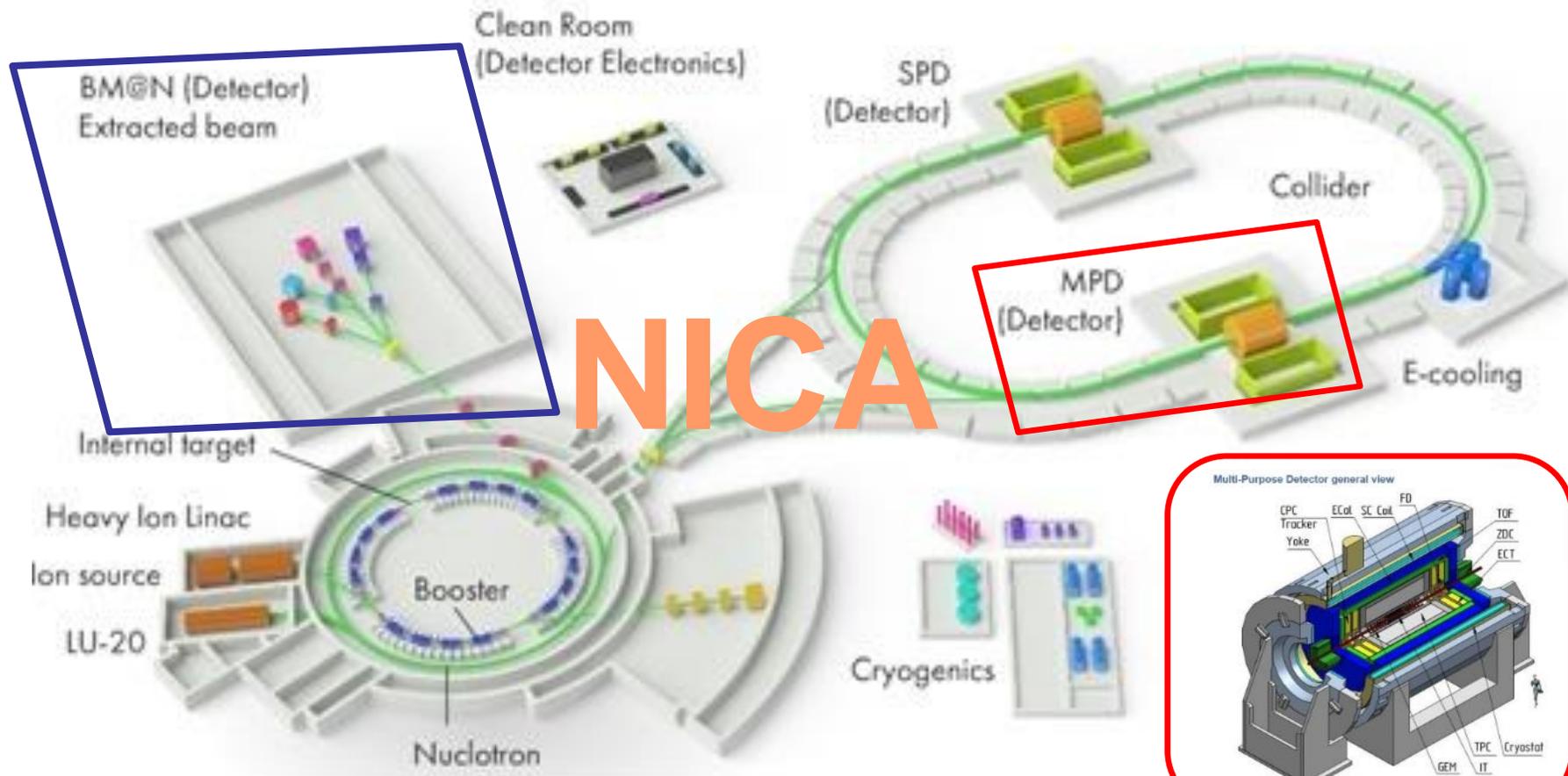


Эксперимент BM@N. Стриповый GEM-детектор внутри магнита



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

01.04.2024



NICA

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

Condensed
Barion
Matter

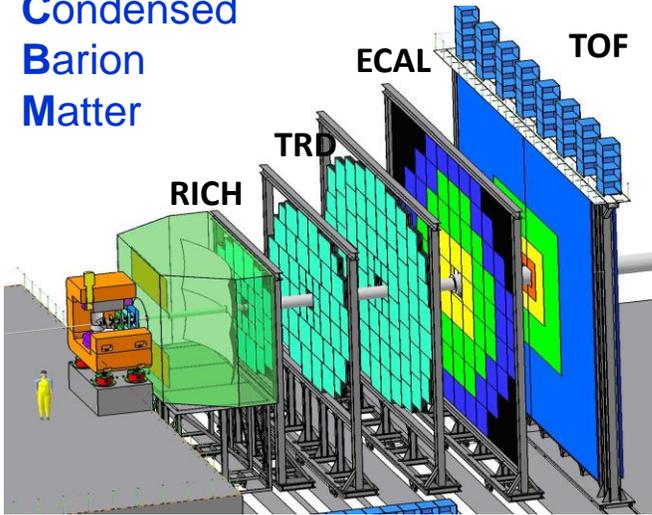


Схема установки CBM

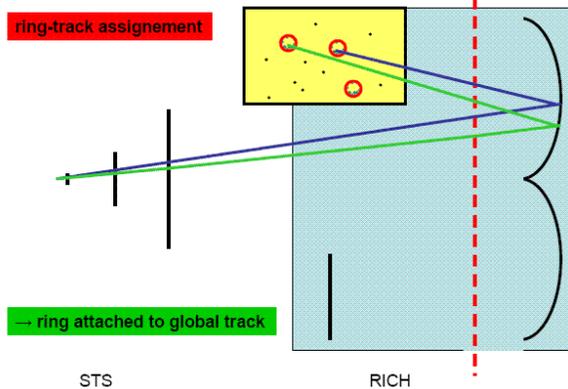


Схема детектора RICH
черенковского излучения

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

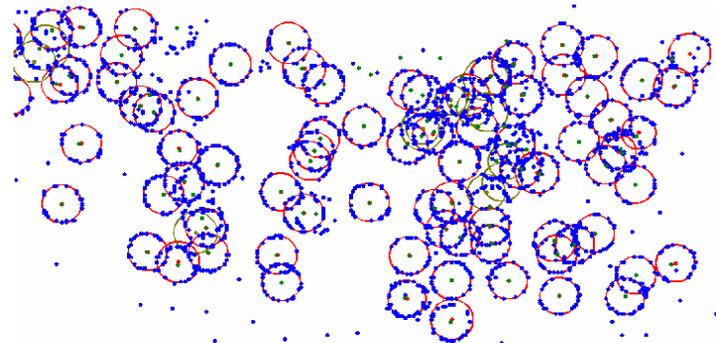
**Скорость передачи
данных:**

10^7 событий в сек,

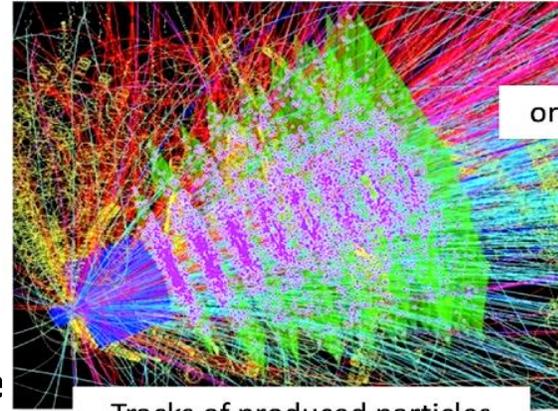
~1000 треков на событие

~100 чисел на трек

Итого: 1 терабайт/сек!

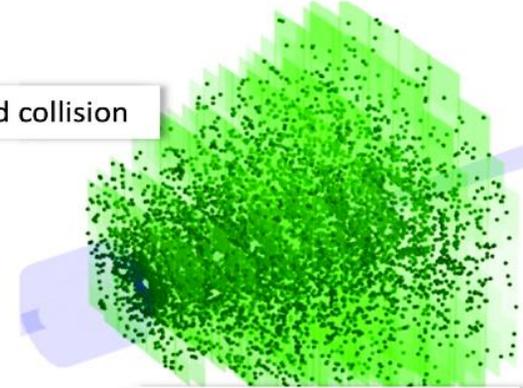


Фрагмент данных фотодетектора. В
среднем 1200 точек, образующих 75 колец



Tracks of produced particles

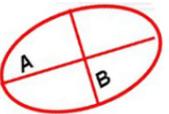
Вид модельного события взаимодействия Au+Au в вершинном детекторе



hits in the STS

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 перцептронов с одним скрытым слоем, нейросетей Хопфилда, фильтра Калмана, робастными методами и применением вейвлет-анализа.

Глубокое обучение ждало новых компьютерных технологий

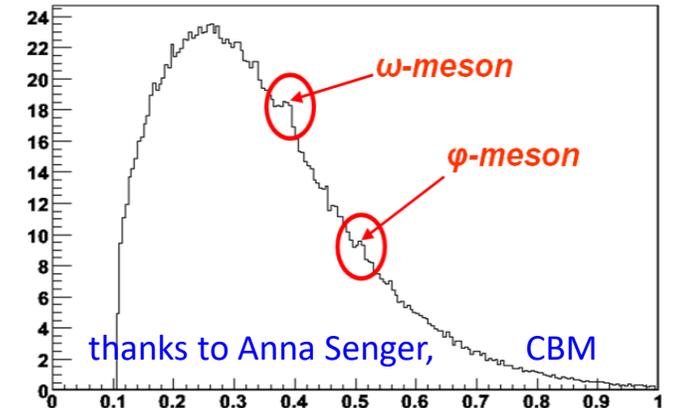
Основные этапы анализа данных в текущих экспериментах ФВЭ

- ❖ 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 компьютеринга для достижения наивысшей скорости и масштабируемости обработки



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

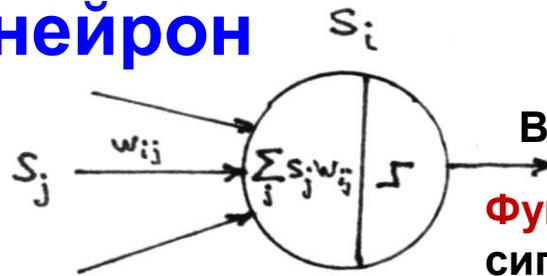
Parallel programming of optimized algorithms Grid-cloud technologies which changed considerably HEP data processing concept
See *Scientific data management in the coming decade* <https://dl.acm.org/doi/10.1145/1107499.1107503>

Применение ИНС в экспериментальной физике

Началось уже в конце 80-х, раньше, чем в других областях науки, поскольку именно физика могла на основе современной физической теории реализовать возможность монте-карловской генерации обучающей выборки любой требуемой длины.

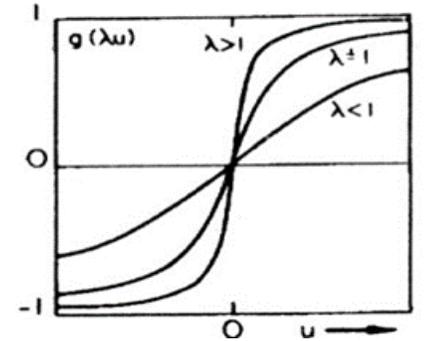
Именно физики написали в 80-х программный нейропакет JetNet и были одними из первых пользователей нейрончиков.

Искусственный нейрон



Выходной сигнал $h_i = g(\sum_j w_{ij} s_j)$

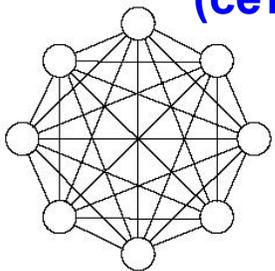
Функция активации $g(x)$. Как обычно, это сигмоид $g(x) = 1/(1 + \exp(-\lambda x))$, но не только



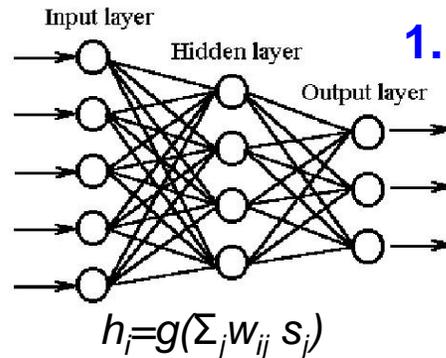
Связь между i-м и j-м нейронами характеризуется синаптическим весом w_{ij}

Выделим три типа нейросетей, с которых все начиналось

2. Полносвязная ИНС (сеть Хопфилда)



Самообучение



1. Прямоточная ИНС или многослойный персептрон (МСП)

$$y_j = f(\sum_k w_{kj} h_k)$$

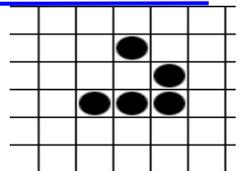
Обучение с учителем (backpropagation)

Цель обучения – определить веса так, чтобы обученная сеть решала задачу распознавания или классификации.

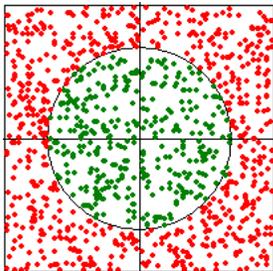
3. Клеточные автоматы

можно рассматривать как сети с локальными связями

Саморазвитие



Тайны обучения. Что внутри черного ящика ИНС?

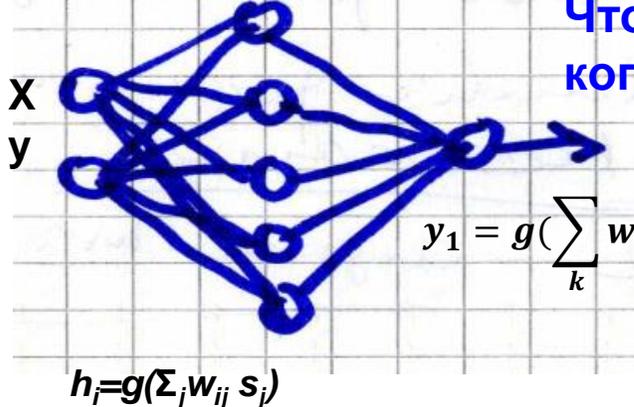


Простой пример: обучить сеть определять, где точка, - внутри круга или вне него.

Обучающая выборка из 1000 троек чисел: на вход сети подают по 3 числа (X, Y, Z) : координаты точки X, Y и признак Z (Z = 1 - внутри круга или Z = 0 – вне его).

Решение: ИНС с одним скрытым слоем из 5 нейронов, два входных и один выходной

Всего 15 весов



Чтобы обучить МСП применяют метод обратного распространения ошибки, когда минимизируют по всем весам функцию ошибки сети:

$$E = \sum_m \sum_{ij} (y_i^{(m)} - z_i^{(m)})^2 \rightarrow \min_{\{w_{ij}\}}$$

Т.о. надо решать систему из 15 уравнений

$$\frac{\partial E}{\partial w_{ij}} = 0$$

с 15 неизвестными значениями весов w_{ij} w_{jk} , для чего требуется дифференцируемость активационной функции $g(x)$, определяющей выход каждого нейрона. Выбор сигмоидальной функции

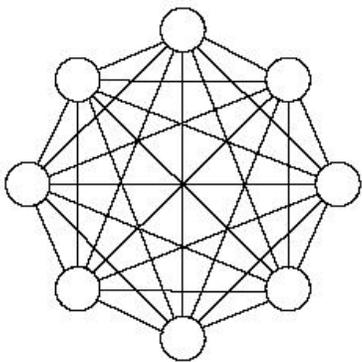
$$g(x) = \frac{1}{1 + e^{-\lambda x}}$$

обеспечивает к тому же простое выражение и для ее

производных, $g'(x) = \lambda g(x)(1 - g(x))$, входящих в формулы для итеративной (по эпохам обучения)

подстройки весов. Для весов выходного слоя имеем $\Delta w_{kj}(t+1) = -\eta (y_j^t - z_j^t) g'(y_j^t) h_k^t$ для скрытого слоя - $\Delta w_{ik}(t+1) = -\eta \sum_j w_{kj} g'(y_j^t) g'(h_k^t) x_k^t$, где η - параметр скорости обучения. Сеть считается обученной, когда в эпохе обучения t максимальная ошибка обучения $E = \max_{t,j} |y_{t,j} - z_{t,j}|$ уменьшится до заданной точности.

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

$$E(s) = - \frac{1}{2} \sum_{ij} s_i w_{ij} s_j$$

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

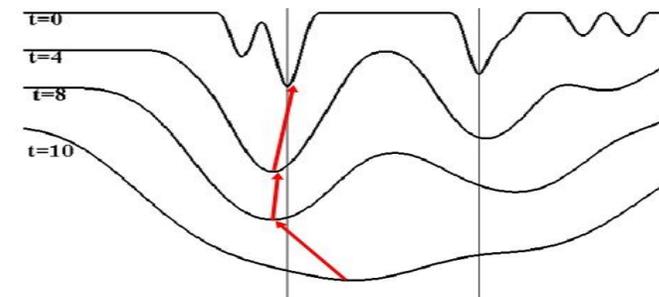
Для нахождения глобального минимума E сеть термализуется.

$$g(u) = \frac{1}{1 + e^{-2u}} \quad \lambda = 1/t \quad E = E(t, W)$$

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:
 $v_i = 1/2(1 + \tanh(-\partial E / \partial v_i \cdot 1/T)) = 1/2(1 + \tanh(H_i / T))$, где $H_i = \langle \sum_j w_{ij} s_j \rangle_T$ – 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 итеративно убывает по схеме **«имитационного отжига» (simulated annealing)**.

При этом на каждой итерации сеть эволюционирует всякий раз, начиная вблизи окрестности глобального минимума.



ХНС применяются для построения ассоциативной памяти и решения оптимизационных задач типа задачи коммивояжера. Особенно успешными были применения для реконструкции событий в трековых детекторах

Различные типы глубоких нейросетей и проблемы их обучения

1. Многослойная прямооточная нейронная сеть

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

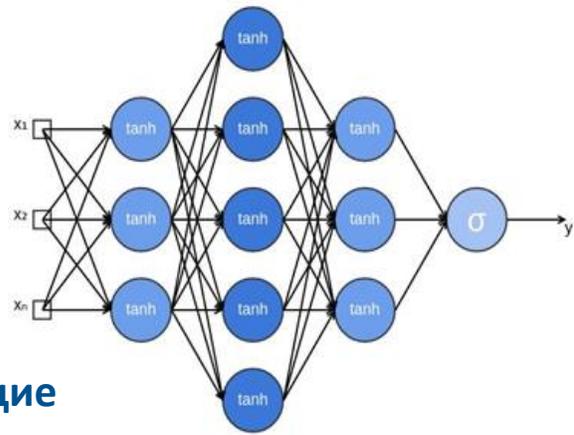
Прежний опыт: чтобы обучить сеть применяют **метод обратного распространения ошибки**, когда методом градиентного спуска минимизируют по всем весам **квадратичную функцию ошибки сети**:

Возникающие проблемы:

$$E = \sum_m \sum_{ij} (y_i^{(m)} - z_i^{(m)})^2 \rightarrow \min_{\{w_{ij}\}}$$

- 1) проклятье размерности
- 2) переобучение
- 3) **застревание E в ложном минимуме**
- 4) затухающий или взрывной градиент
- 5) Выбор функции активации
- 6) Инициализация значений весов нейросети
- 7) Выбор адекватной функции потерь

Проблемы эти удалось решить только в 20-х годах этого века, когда были придуманы алгоритмы минимизации многомерных функций ошибки нейросети и появились компьютеры, позволяющие реализовать эти алгоритмы.



https://www.researchgate.net/publication/322673516_Shallow_and_deep_learning_for_image_classification

Например, для для решения проблемы ложных минимумов можно применить уже упоминавшийся метод отжига. Однако более эффективен **Стохастический градиентный спуск (Stochastic Gradient Descent – SGD)**, при котором требуется только один проход по обучающим данным, когда значение градиента аппроксимируются градиентом функции ошибки, вычисленном только на одном элементе обучения, что работает много быстрее. Выбор точки обучения в SGD происходит случайно, но попеременно из разных классов, что также повышает вероятность выхода из локального минимума.

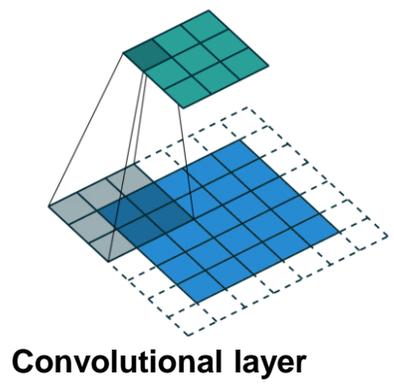
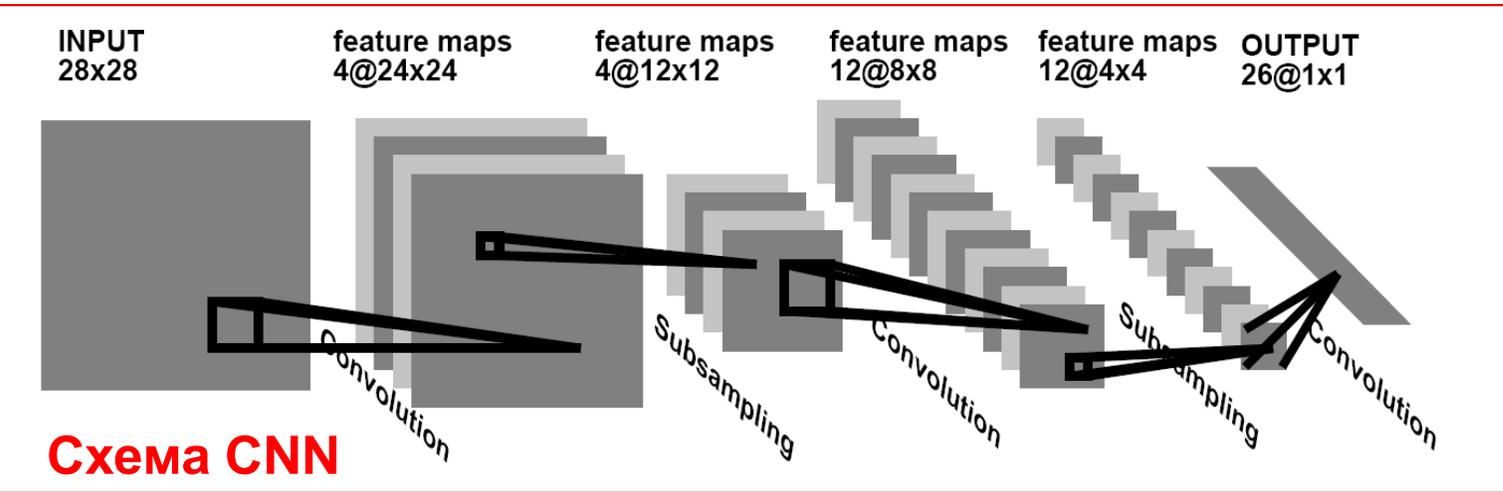
Все эти возможности включает метод ADAM (Adaptive Moment Estimation)

2. Сверточные нейросети для распознавания изображений

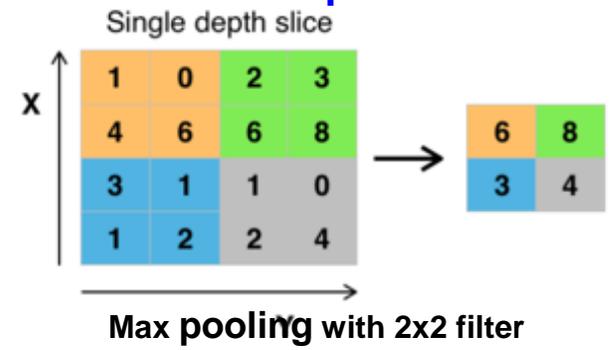
Мотивация: Прямое применение регулярных ИНС к распознаванию изображений бесполезно из-за двух основных факторов: (i) входное 2D-изображение в виде сканированного 1D-вектора означает потерю топологии пространства изображения; (ii) полносвязность ИНС, где каждый нейрон полностью связан со всеми нейронами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



Поясняющая аналогия

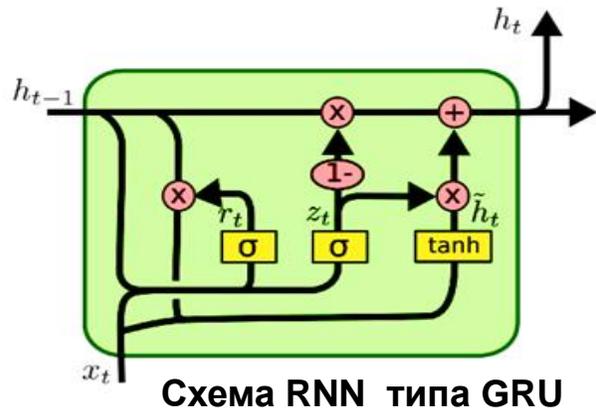


Вместо этого, сверточные сети - Convolutional Neural Networks (CNN) принимают в сверточных слоях на вход 3D двумерные цветные данные (x, y, RGB) и преобразует их в выходные 3D карты признаков. Нейроны в слое CNN связаны только с малой областью предыдущего слоя. CNN должна быть обучена на помеченных изображениях с помощью метода обратного распространения ошибки



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.



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) 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 VM@N в ОИЯИ.

Трансфóрмер — 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 векторизованную 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 параметров треков прямо из сырых данных, минуя времязатратный этап кластеризации измерений для формирования ХИТОВ.

Другие типы глубоких нейросетей

3. 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.

4. 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.

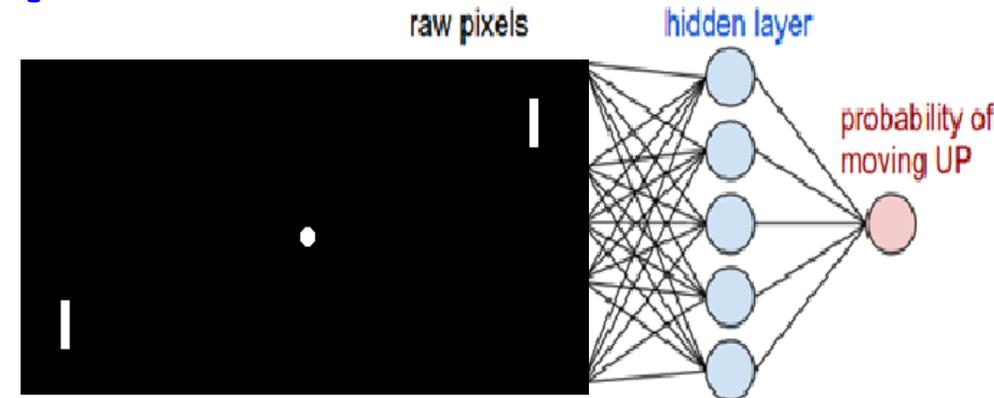
Применения GAN:

- 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>



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



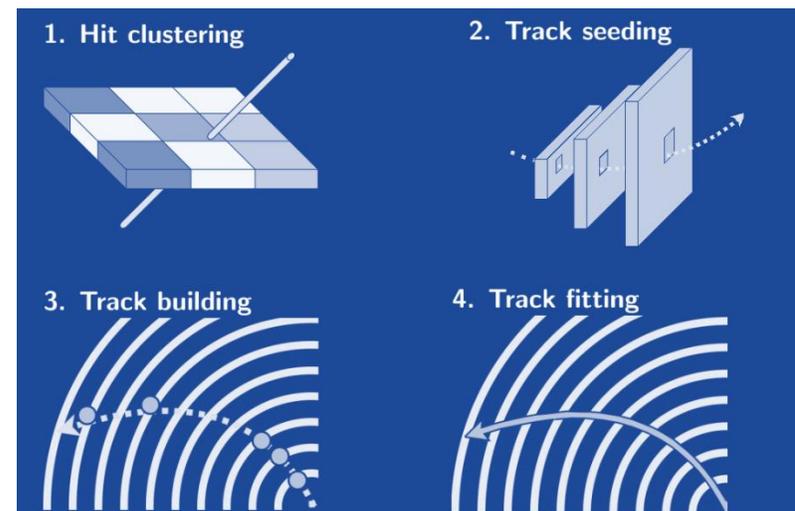
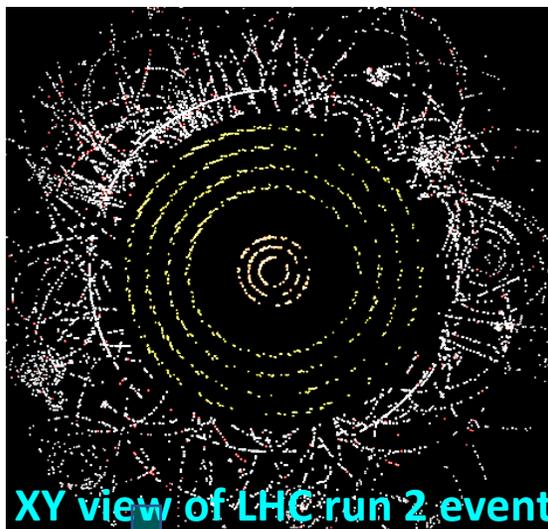
Восстановление треков

– ключевая проблема реконструкции событий ФВЭ

Реконструкция должна определять параметры вершин и траекторий частиц (треков) для каждого события. Традиционно **алгоритмы трекинга, основанные на комбинаторном фильтре Калмана**, с большим успехом использовались в экспериментах ФВЭ в течение многих лет.

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.

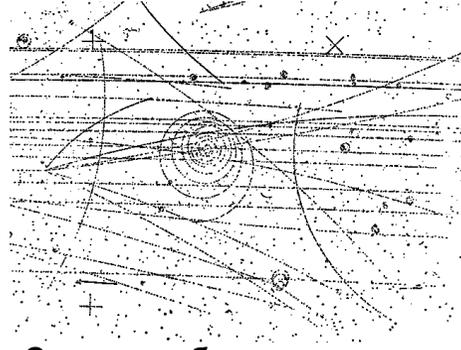


Процедура трекинга включает в себя фазы: (1) получения хитов (hit clustering), (2) построения треков-кандидатов - наборов хитов с вычисленными параметрами (*англ. seeds*), (3) прослеживания треков и (4) их подгонки уравнением движения частицы в магнитном поле.

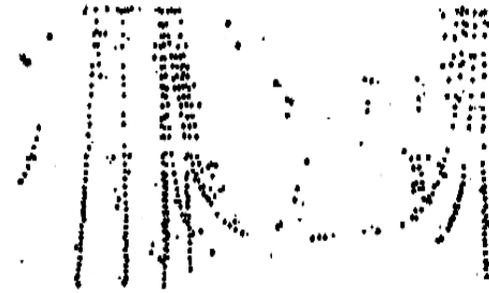
Главная проблема современного трекинга- высокая светимость пучков ускорителей, т.е. мегагерцовый темп поступления данных и банчевая структура пучка

Эволюция методов трекинга

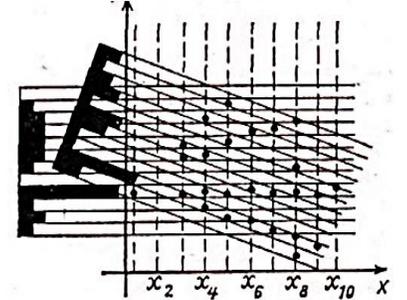
Началось еще в эпоху **пузырьковых камер**, когда события регистрировались на стереофотографиях и **вводились в компьютер вручную**, полуавтоматами или с помощью **сканирующих устройств** типа «Спиральный измеритель», в котором оператор ставил точку в вершину события, откуда шло сканирование снимка по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



Снимок события.



Его оцифровка в полярных коорд.

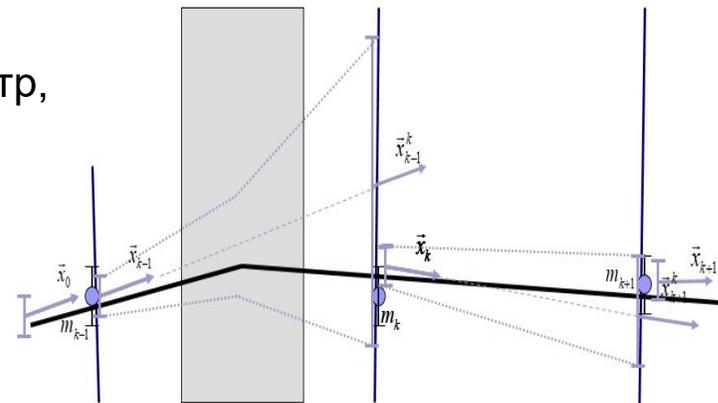


Поворотные гистограммы

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.

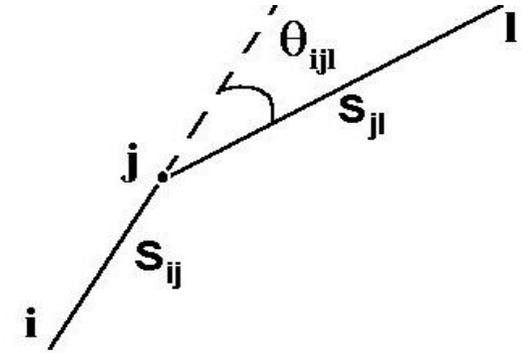


Главный недостаток ФК – необходимость знать начальное значение вектора состояния \vec{X} , выполнить «СИДИНГ» (англ. seed-семя)

Однако ФК медленный, плохо распараллеливается и масштабируется!

Распознавание треков с помощью сети Хопфилда. Метод сегментов.

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



А теперь вспомним про нейросеть Хопфилда!

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}

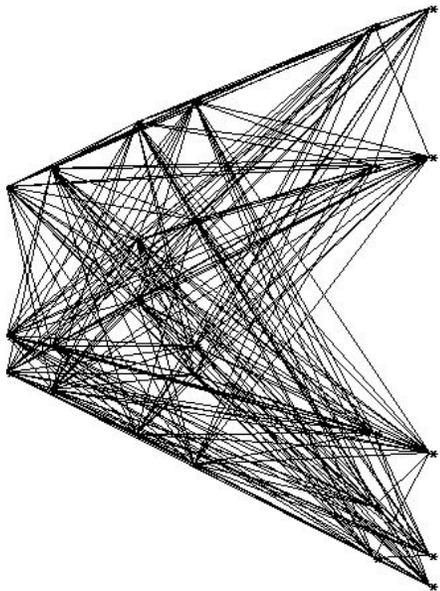
поощряет связи нейронов принадлежащих одному и тому же треку, т.е. короткие смежные сегменты с малым углом между ними.

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 для распознавания событий с короткоживущими частицами

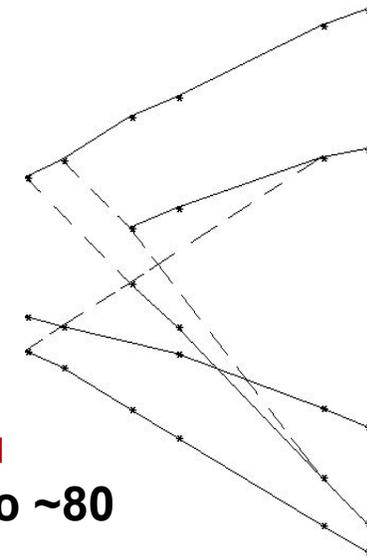
Эксперимент EXCHARM (Протвино 90-е годы) - проблема: в отличие от Денби-Петерсена разрешить бифуркации, но не допустить массовых ветвлений треков



на нулевой итерации

всего 244 нейрона

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



на 30-ой итерации

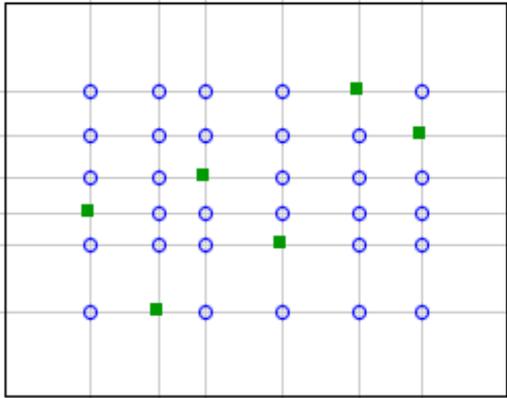
$V_{ij} > 0.5$
у 26

нейронов

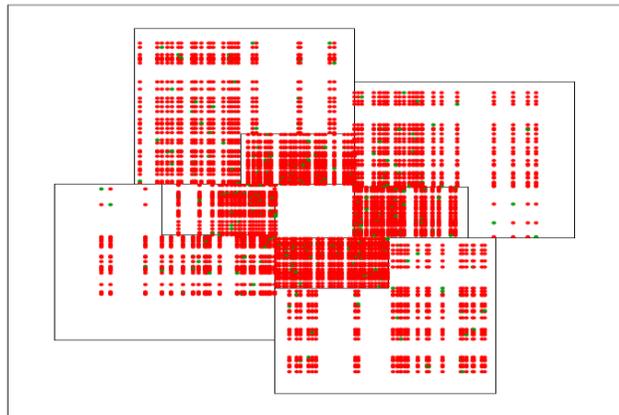
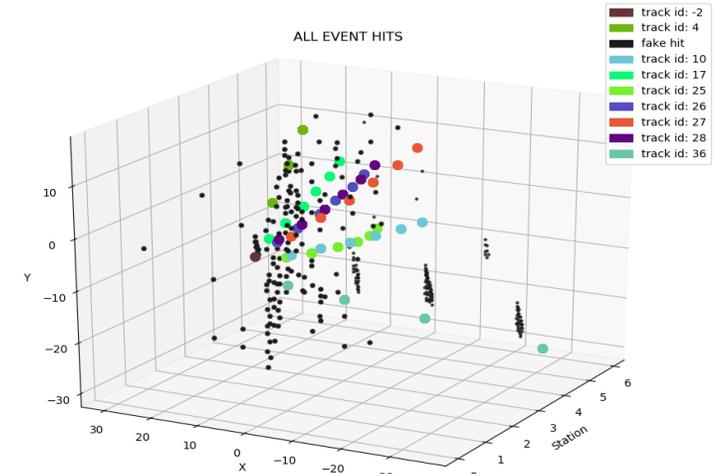
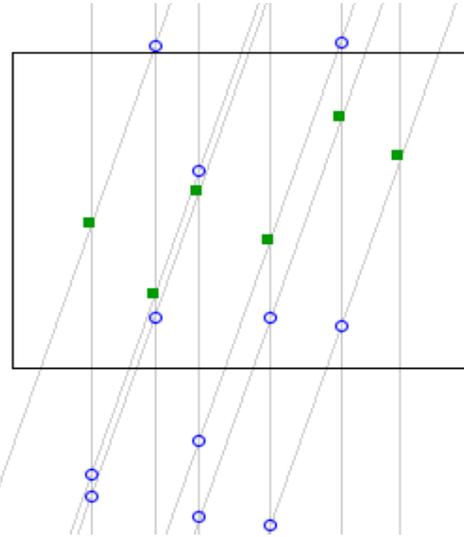
Однако чрезмерная чувствительность к шумам и такие недостатки применения полносвязных нейросетей, как слишком медленная сходимость и то, что не учитывается известное уравнение движения частицы в магнитном поле, потребовало поиска новых подходов к проблеме трекинга с **применением глубоких нейросетей**

Проблемы трекинга для GEM и полосковых детекторов

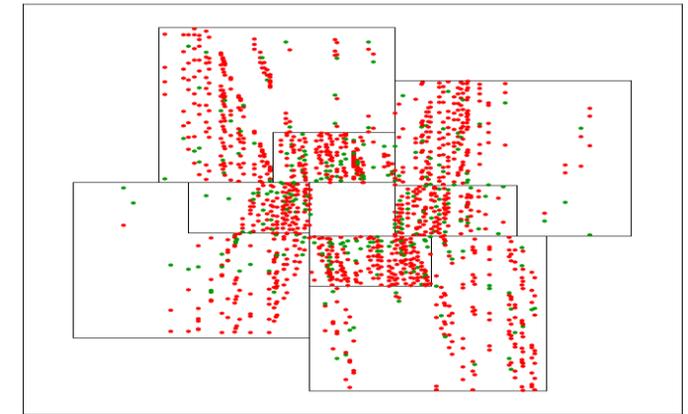
Главная трудность, вызванная спецификой GEM детектора – появление ложных отсчетов из-за лишних пересечений стрипов. Для n истинных хитов имеем $n^2 - n$ фейков!



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.



Две основных проблемы - наличие фейковых засорений данных и, главное, сверхвысокий темп их поступления из-за высокой светимости неизбежно требуют разработки новых методов трекинга с использованием глубоких нейронных сетей

Локальный и событийный подходы к трекингу

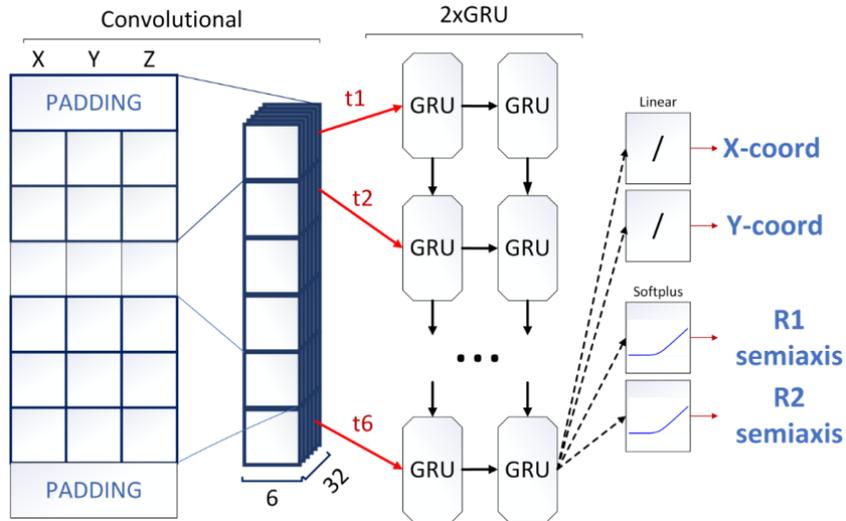
Два подхода к реализации «глубокого трекинга»

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. Событийный 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.

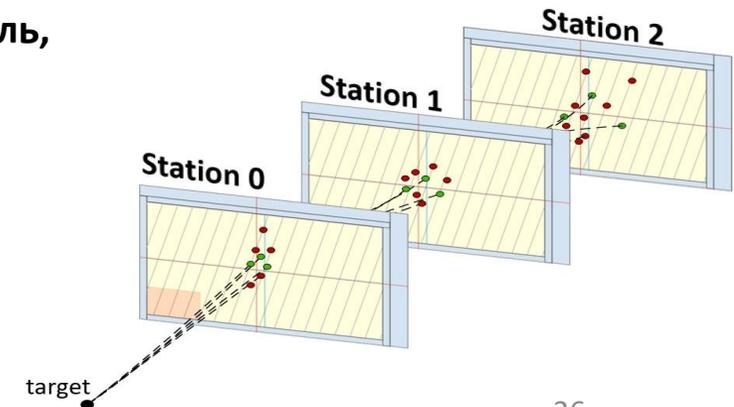


Scheme of the recurrent TrackNETv2 neural network

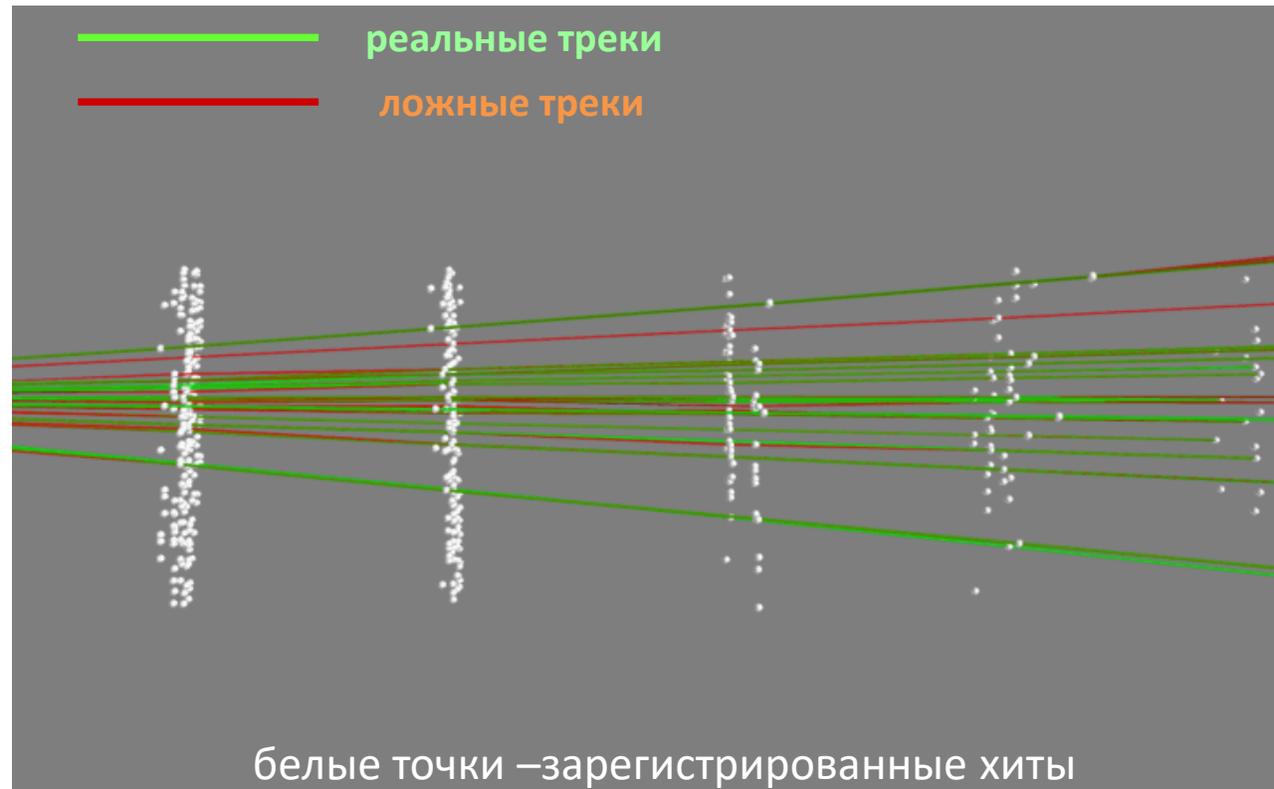
See <https://doi.org/10.1063/1.5130102>

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

Для наиболее употребительных типов трековых детекторов типа GEM или Straw Tubes характерно неизбежное **появление огромного количества ложных отсчетов - фейков**, число которых на два порядка выше реальных измерений, Кроме того, из-за неэффективности самих станций детектора **появляются пропуски среди хитов, образующих треки**. Всё это приводит 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) – гостями.

Например для симулированных данных 7-го рана VM@N обученная TrackNET нашла реальные треки с достаточной эффективностью 97%, однако доля гостей при этом превысила 50%, что недопустимо.

Поэтому потребовалась модификация TrackNET с заменой области предсказания с эллиптической на трехмерную сферу и, главное был введен **второй этап трекинга, глобальный, учитывающий взаимосвязь всех треков каждого**

Пособытийный трекинг

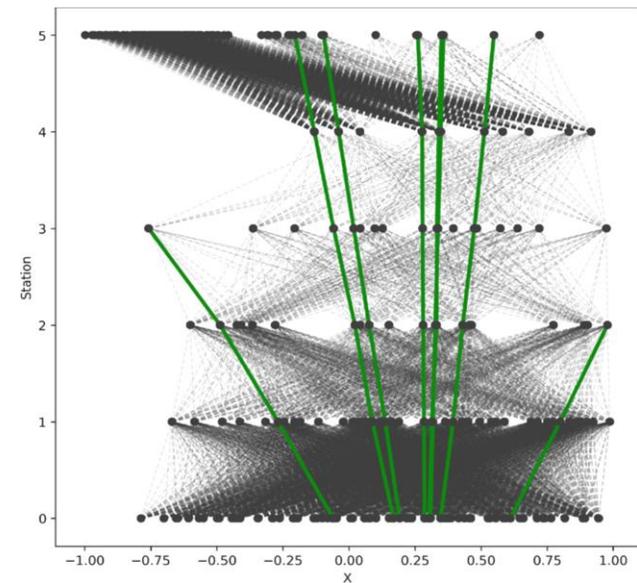
Выделим три метода, когда распознавание треков среди шумов осуществляется сразу по всей картине события.

2.1. Применение графовых нейронных сетей. Эксперимент VM@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.

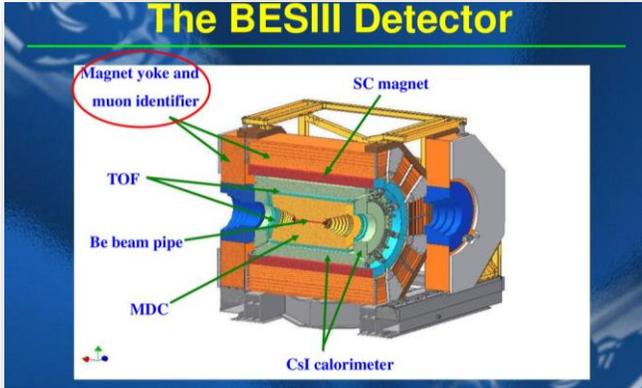
Эта схема похожа уже известный глобальный подход Денби-Петерсона с сегментной нейросетью Хопфилда, где нейросеть подолгу самообучалась отдельно для каждого события, в то время как GNN, где надо найти те ребра, что являются сегментами реальных треков можно обучить на выборке из графов событий, где эти 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 VM@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

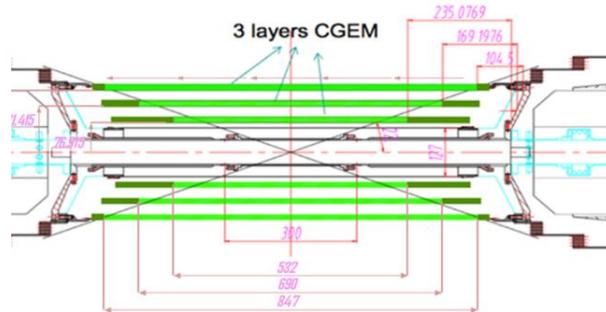


Графическое представление события C + C, 4 ГэВ эксперимента VM@N. Черные узлы и ребра соответствуют фейкам, зеленые узлы и ребра - найденным трекам

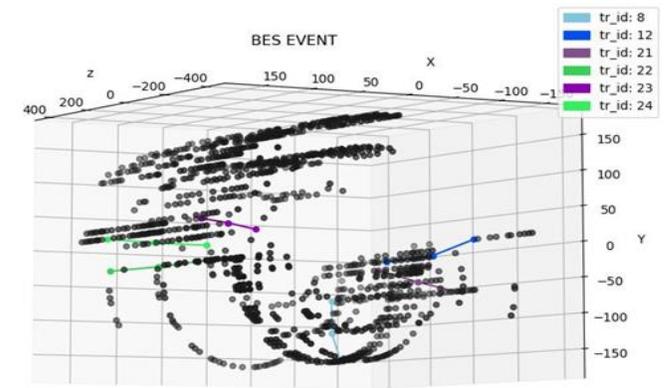
2.2. Применение графовых нейронных сетей, эксперимент BES-III



BESIII – коллайдерный эксперимент



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



Все хиты модельного события

Граф события инвертируется в линейный диграф, 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 $x \in [0,1]$ 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

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**.

Пособытийный 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 сверточных нейросетях 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.

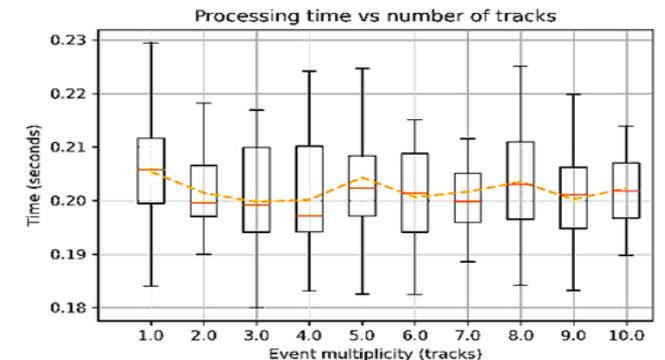
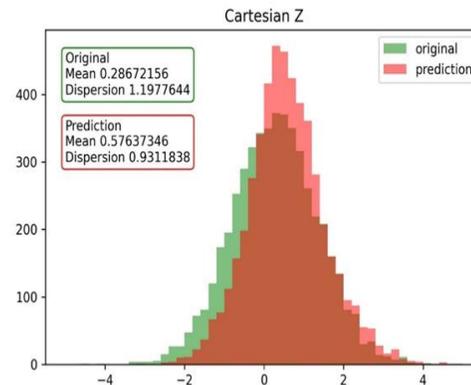
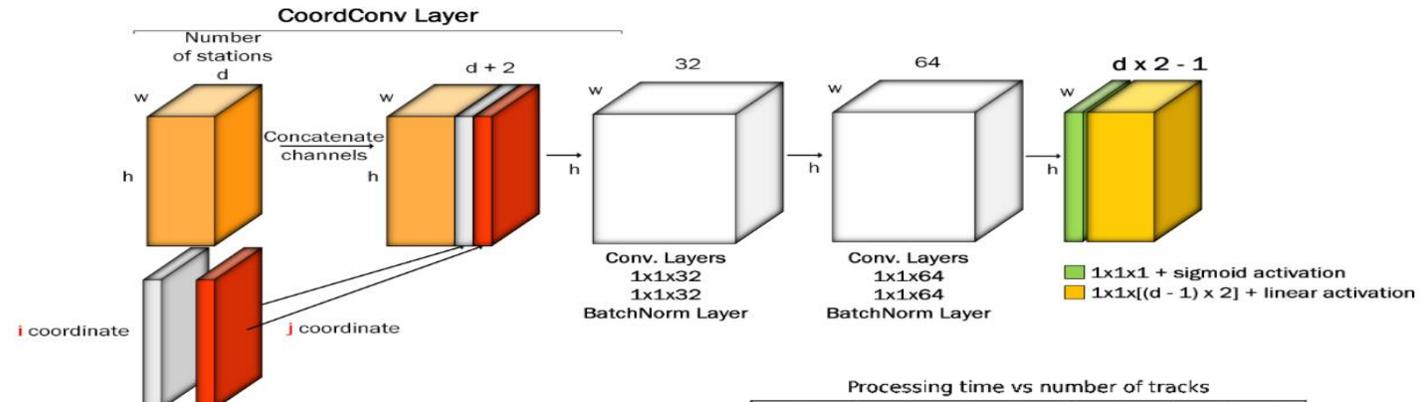
Это радикально новый подход, позволяющий найти координаты вершины события

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**

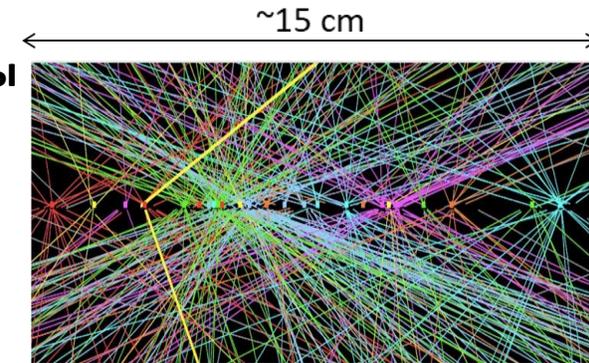


Время работы обученной модели не зависит от множественности события

Эксперименты с высокой светимостью. Кризис трекинга

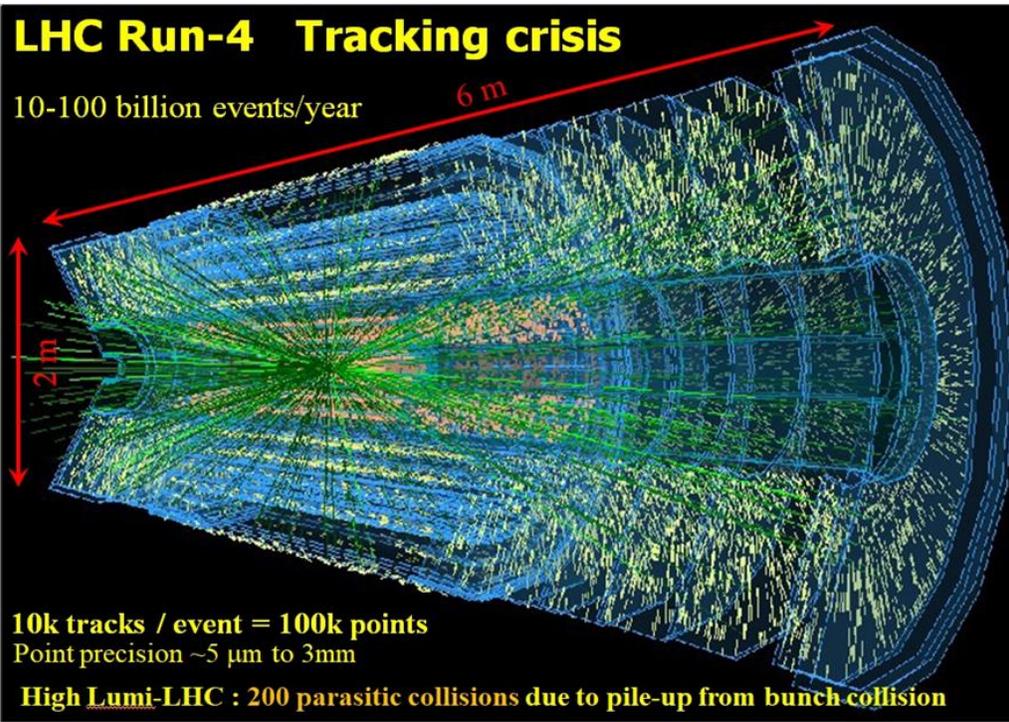
Для достижения намеченных ультимативных целей светимость Большого адронного коллайдера в ЦЕРНе будет увеличена, так что количество дополнительных столкновений достигнет уровня 200 взаимодействий на пересечение пучка, что в 7 раз превышает текущую (2017 г.) светимость. Это станет вызовом для экспериментов 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)



LHC Run-4 Tracking crisis

10-100 billion events/year



Модельное события в HL-LHC

Поэтому моменты столкновений происходят так близко, что треки событий сильно перекрываются в 15 сантиметровой области встречи пучков.

Текущая ситуация: 20 паразитных столкновений, High Lumi-LHC: 200 паразитных столкновений.

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

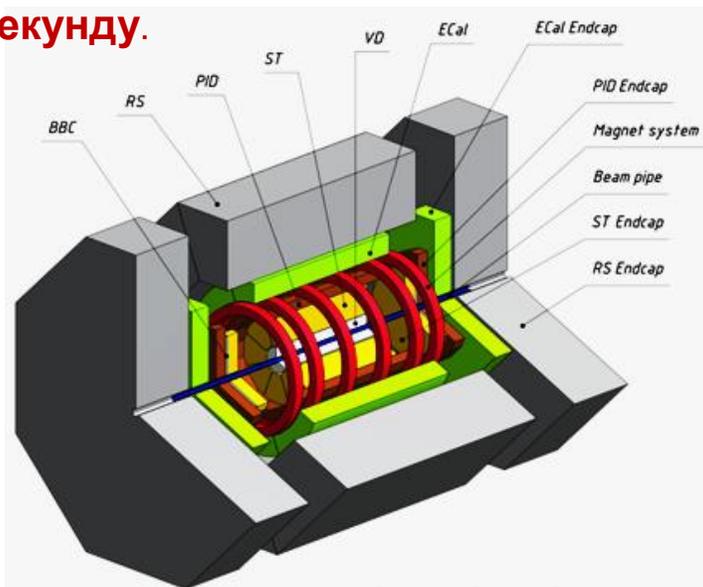
Эксперименты с высокой светимостью

Что уже сделано по трекингу

- В 2018 году физики из ЦЕРНа и др. физических центров мира, включая Россию, устроили соревнование - **TrackML challenge** по решению проблемы машинного обучения для трекинга частиц в физике высоких энергий в условиях высокой светимости ([DOI 10.1109/eScience.2018.00088](https://doi.org/10.1109/eScience.2018.00088))
- Для этого на платформе Kaggle сделана программа-симулятор с исходным кодом, где типичный полностью пиксельный трекинг-детектор БАК из 10 слоев генерирует физические события (Pythia ttbar), наложенные на 200 дополнительных столкновений. Это дает, как правило, 10000 треков (100000 хитов) в каждом событии.
- Соревнование **TrackML** стимулировало много исследований, где TrackML dataset использовался для обучения и верификации разных нейромоделей трекинга
 - Много программ для графовых нейросетей, например <https://arxiv.org/pdf/1810.06111.pdf>
 - Возник также интерес к применению нейросетей Хопфилда, но совсем в другом аспекте, медленную эволюцию сети предлагается резко ускорить путем квантового отжига, выполненного на квантовом D-Wave компьютере <https://doi.org/10.1007/s42484-021-00054-w>
 - Более того предлагается также применить квантовый отжиг и для ускорения работы графовых нейросетей [arXiv:2109.12636v1 \[quant-ph\] 26 Sep 2021](https://arxiv.org/abs/2109.12636)
 - **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**
- Появились уже вне рамок соревнования отчеты о выполнении реальных задач трекинга по данным LHC RAN 2 и 3 (см. например, [arXiv:2308.09471v1 \[hep-ex\] 18 Aug 2023](https://arxiv.org/abs/2308.09471))

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 $10^{32} \text{ cm}^{-2} \text{ s}^{-1}$. Данные о событиях из SPD будут поступать со скоростью 3 МГц в виде тайм-слайсов в 10 мс, в каждом из которых будет происходить до 40 событий, т.е. один тайм-слайс будет содержать до 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.**



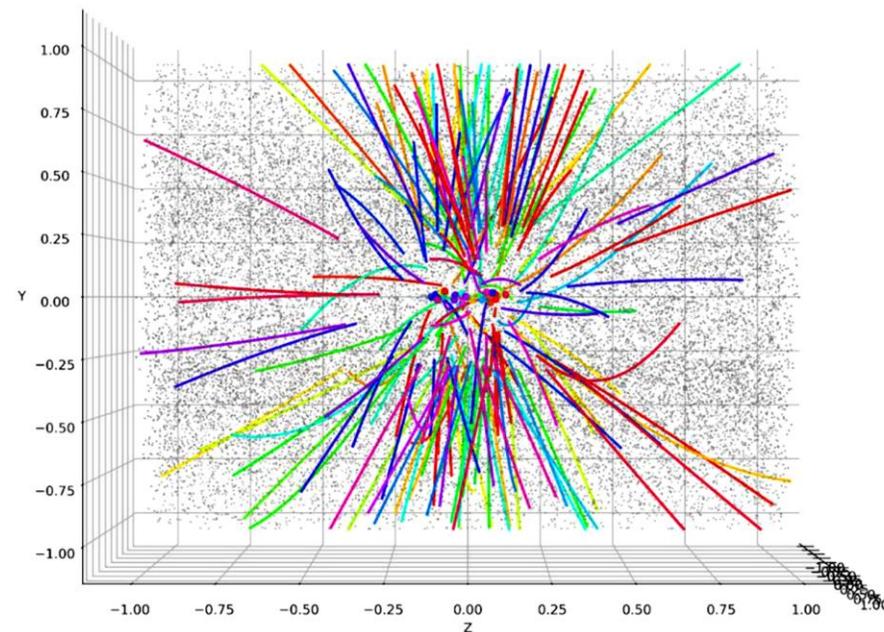
Общая схема установки SPD. ST - Straw-Trecker. Его основной модуль состоит из 31 двойного слоя строу-трубок

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.

Была достигнута скорость обработки ~ 1800 событий в секунду при допустимой эффективности трекинга

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

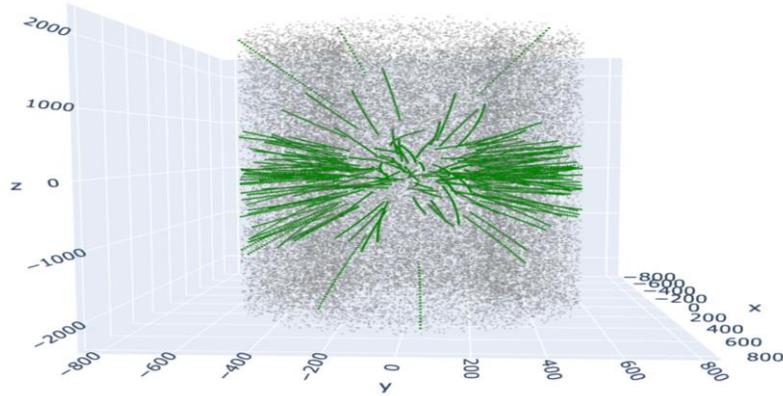
Внесение соответствующих усложнений в программу TrackNET неизбежно замедляет ее работу и снижает эффективность



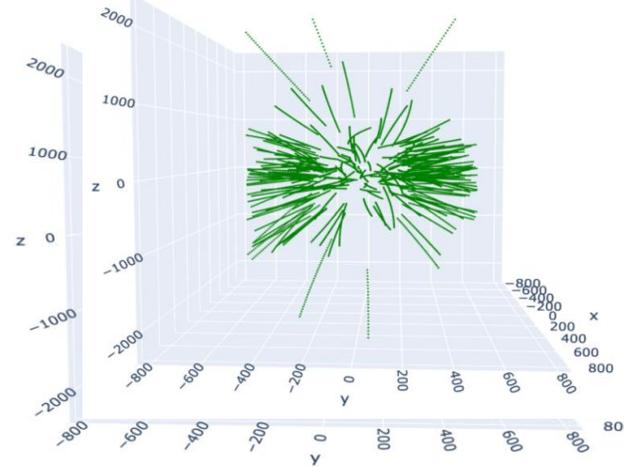
Пример тайм-слайса в эксперименте SPD. Треки показаны цветными линиями, их первичные вершины – точками соответствующего цвета. Фейковые хиты показаны серыми точками.

Вокселизация событий SPD. Программный конвейер

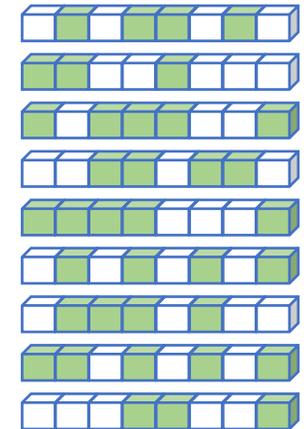
Input: Raw event



Output: Cleaned event

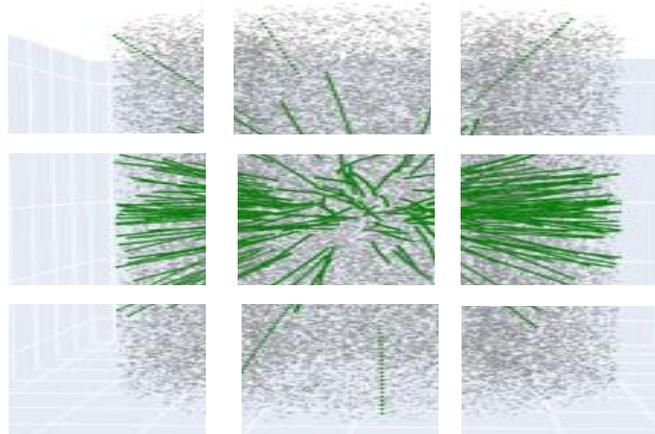
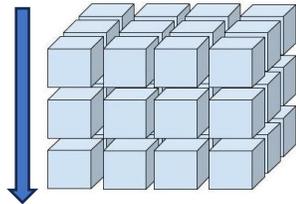


Объединяем
все в одно
событие

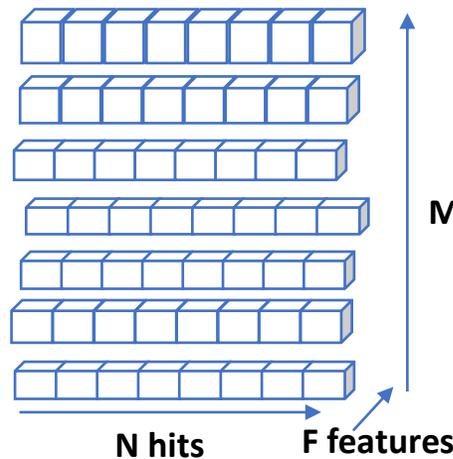


Классифицируем
хиты на истинные и
фейки

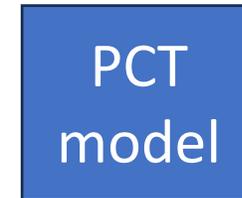
Делим пространство
детектора на M вокселей,
т.е. на более мелкие
подпространства



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



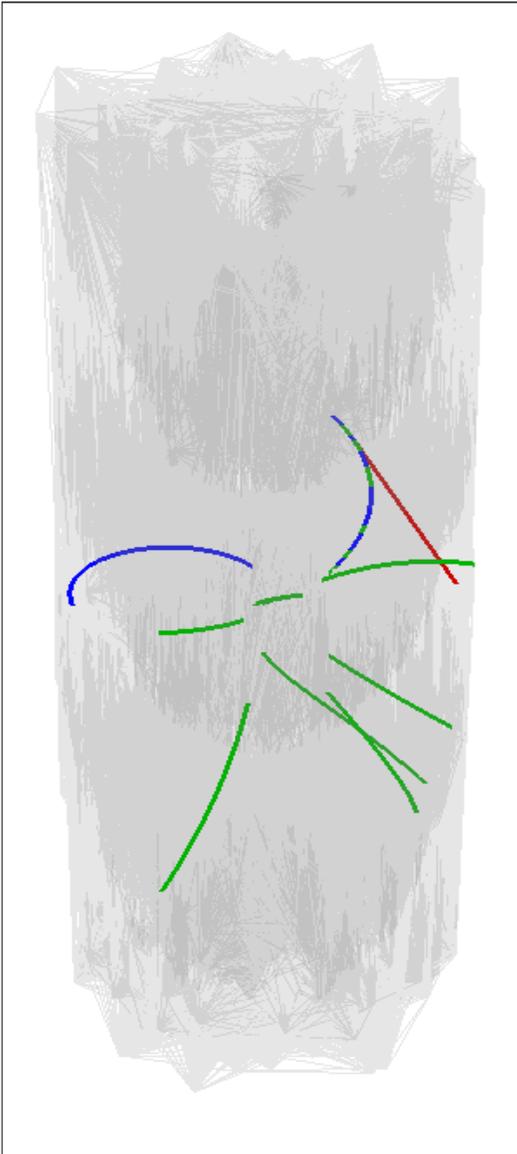
PCT= Point cloud transformer



РЕЗУЛЬТАТЫ ЧИСЛ. ЭСПЕРИМЕНТА

Число событий	Неэффект	Число вокселей	Precision	Recall
40	2%	1000	0.94	0.96

Пример применения улучшенного алгоритма сети Хопфилда для модельных данных SPD.



Подход - похож на классический метод Денби-Петерсона, то есть поиск глобального минимума для сети Хопфилда методом среднего поля, но с внесением дополнительных параметров в формулу энергии.

Результат трекинга события с 10 треками, 1750 шумовых хитов.

- **Активные трековые сегменты (true positive) - зеленые**
- Неактивные нетрековые сегменты (true negative) - серые
- **Активные нетрековые сегменты (false positive) - красные**
- **Неактивные трековые сегменты (false negative) – синие**

Однако эффективность на большой выборке оказалась недостаточной.

Необходимо исследовать расширение функциональных средств нейросетевого подхода Хопфилда, основанного на нейронах-дуплетах из двух соседних измерений на треке, до новых понятий, объединяющий уже три и четыре последовательных измерения, - триплетов и квадруплетов,

Это можно сделать в рамках квантового отжига с использованием современных квантовых компьютеров и переформулировки задачи трекинга, как квадратичной неограниченной двоичной оптимизации (QUBO), решаемой на квантовом вычислителе типа D-Wave

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 генеративно-сопоставительных нейросетей в создании картин и диссертаций следует отметить публикации об их успешном применении для симуляции взаимодействий в экспериментах ФВЭ



Г.А.Ососков

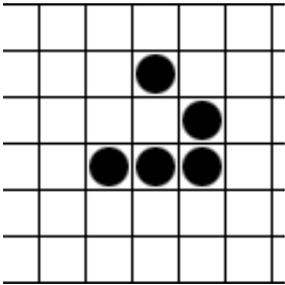
**Методы машинного обучения и
интеллектуального анализа
экспериментальных данных.
История, проблемы и перспективы**

Спасибо за внимание!

email: gososkov@gmail.com
<http://gososkov.ru>

Клеточные автоматы (КА)

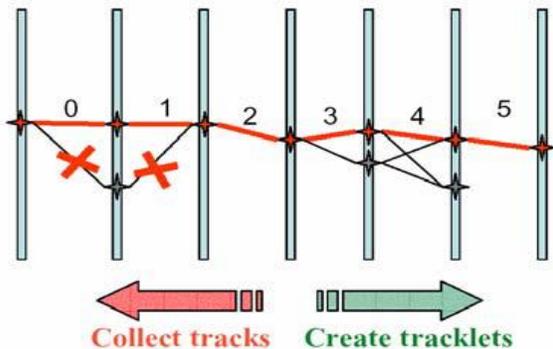
КА – это дискретная динамическая система, образованная регулярной решеткой ячеек, каждая имеет нескольких **состояний**, например 1 и 0. Для каждой ячейки определены ее **соседи и правила перехода из одного состояния в другое**. Правила **локальны**, т.е. зависят только от соседей. Изменения значений всех клеток происходят **одновременно**. Дж.Конвей предложил правила КА для игры **«Жизнь»**:



если клетка имеет двух "живых" соседей, она остается в прежнем состоянии. Если клетка имеет трех "живых" соседей, она переходит в "живое" состояние. В остальных случаях клетка "умирает".

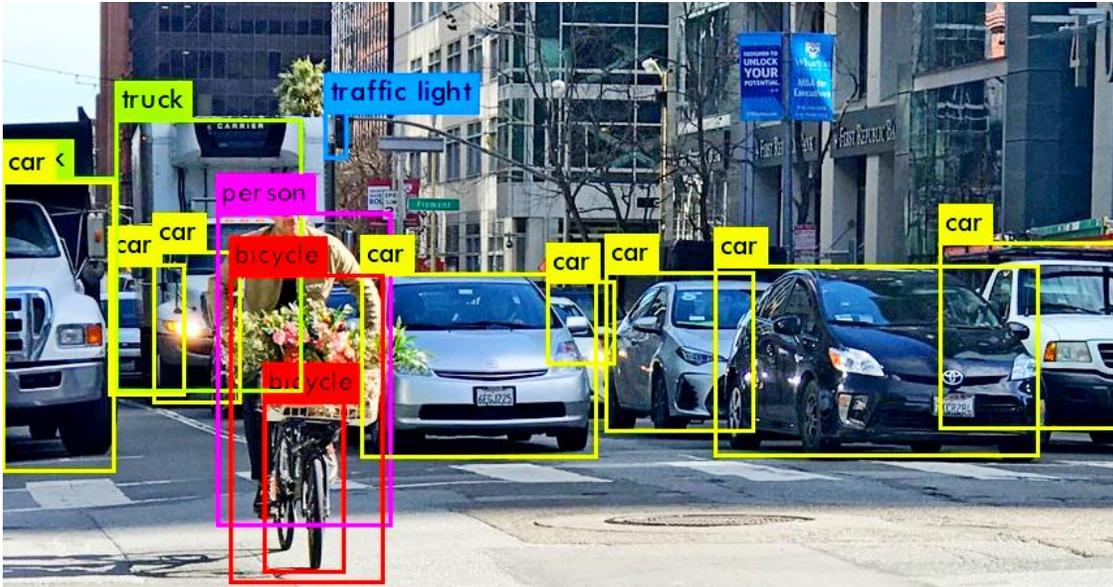
Применение подобных правил к зашумленным данным измерений для организации «вымирания» изолированных шумовых точек оказалось весьма эффективным способом

Еще более полезным оказалось применение КА для **реконструкции треков**. В качестве клеток берут сегменты (tracklets), соединяющие экспериментальные отсчеты на соседних координатных плоскостях. Клетка =1, если на данном этапе сегмент считается частью трека, и 0, если отрезок



соединяет точки, не лежащие на одном треке. Соседство устанавливается по совпадению конечной и начальной точек сегментов и их близости по направлению (по малости угла между ними).

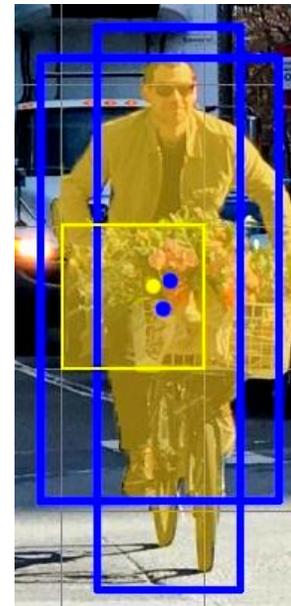
Детектирование объектов - важный аспект взаимодействия со средой



Философия **YOLO** (от англ. **you only live once** — «ты живешь только один раз») — **стремление исполнять желания и получать удовольствие прямо сейчас, не отказываясь от этого ради будущего благополучия.**

YOLO — современный алгоритм глубокого обучения “You Only Look Once” (“Стоит только раз взглянуть”), который широко используется для обнаружения объектов. Называется так, потому что подразумевает ввод сразу всего изображения, которое проходит

через сверточную нейросеть (СНС) **только один раз**. Именно поэтому он очень быстро **опознает объекты в режиме реального времени**. Для детектирования изображение разбивают на ячейки и обычная СНС по многу раз смотрит наличие объекта в каждой ячейке, в YOLO же это делается всего один раз. Вокруг клетки рисуются несколько прямоугольников для определения объекта (их рисуют сразу несколько и разных форм), и их позиции, ширина и высота вычисляются относительно центра этой клетки. После чего специальный блок анализирует все возможные прямоугольники и выбирает объединение тех, куда объект вписывается наилучшим образом (см. <https://habr.com/ru/post/514450/>) Алгоритм YOLOv3 был обучен на датасете “Coco”, состоящем из 80 различных классов



Проблемы нефизических приложений глубокого обучения

- Эра Больших Данных сегодня охватила все области науки и приложений: биологию, экономику, медицину и т.д. Поэтому общий спрос на новые методы обработки данных, такие как глубокое обучение, неизбежно стал актуальным и в этих областях.
- Однако почти фатальным отличием этих областей от физики является отсутствие знаменитой физической "стандартной модели", позволяющей генерировать обучающие выборки любой желаемой длины для надежного обучения разрабатываемых глубоких нейронных сетей.
- **Отсутствие маркированных данных для обучения становится одной из главных проблем.** Помимо отсутствия теоретических предпосылок, ограниченный набор обучающих данных объясняется как высокой стоимостью самих экспериментов, так и стоимостью работы экспертов, маркирующих полученные данные. В качестве примера рассмотрим проблему некоторых сельскохозяйственных приложений CNN

Определение болезней растений по фото их листьев



Подробности можно найти в статье <https://www.mdpi.com/2673-9909/3/4/43>, где описана управляющая ИТ-среда для фермерского хозяйства, использующая веб-платформу <http://pdd.jinr.ru>, чтобы по видео больных растений выяснять диагноз, причину и способы лечения заболевания.

Для преодоления препятствия в виде малой обучающей выборки мы используем **сиамскую CNN со специальной триплетной функцией потерь и персептроном в качестве классификатора на выходном слое**, который показывает 98% точности для 25 классов пяти культур.