Осенняя Школа по информационным технологиям

Октябрь 16-20 2023 МЛИТ ОИЯИ

О машинном обучении и его применении к задачам физики высоких энергий

Ососков Геннадий Алексеевич

Объединенный институт ядерных исследований
Лаборатория информационных технологий им. М.Г.Мещерякова
email: gososkov@gmail.com
http://gososkov.ru

Искусственный интеллект -

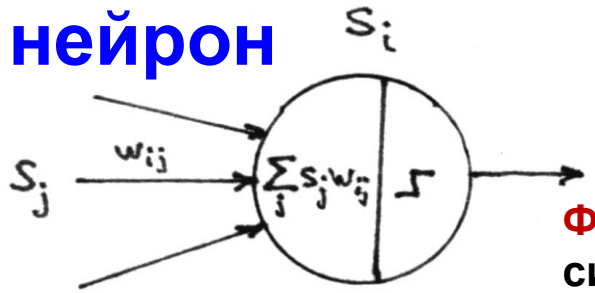
– Машинное обучение –

Машинное обучение (Machine Learning-ML), это когда компьютер не просто использует заранее написанный алгоритм, а используя данные, сам обучается решению поставленной задачи.

- Искусственные нейронные сети

– Глубокое обучение

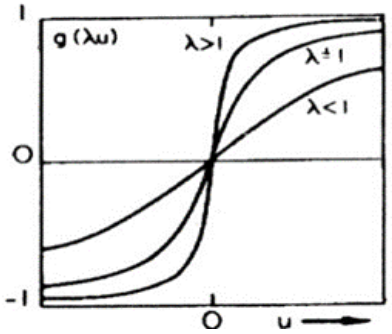
Искусственный нейрон

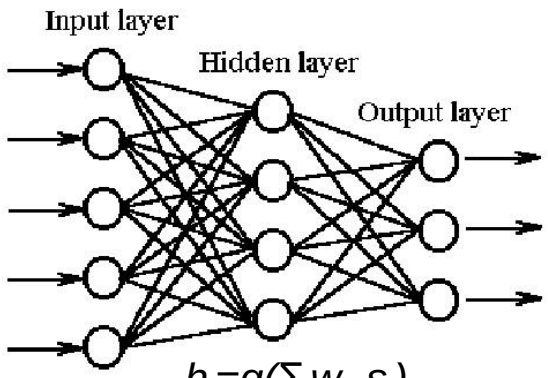


Связь между i^{th} и j^{th} нейронами выражена как синаптический вес w_{ij}

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

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





$h_i=g(\Sigma_j w_{ij}\, s_j)$

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

ИНС в экспериментальной физике

1. Прямоточная ИНС или многослойный персептрон (МСП)

Обучение с учителем.

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

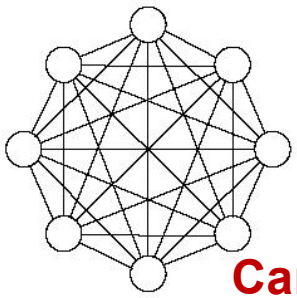
Этапы применения МСП: 1. Создать обучающую выборку как набор пар (X_i,Z_i), где X_i – входное значение, Z_i – его целевое значение.

2. **Обучить МСП, т.е. так подстроить веса w_{ij}, чтобы сеть определяла правильный выход для входов, не использованных при обучении**

3. **Протестировать МСП на тестовой выборке**

4. **Обученная сеть реализуется как программа или нейрочип для очень быстрого применения**

2. Полносвязная ИНС (сеть Хопфилда)

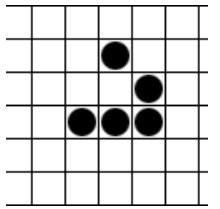


Самообучение

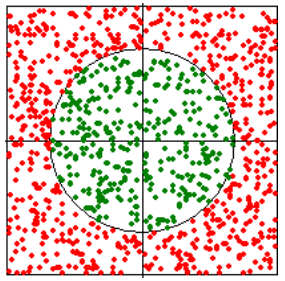
3. Клеточные автоматы

Саморазвитие

можно рассматривать как сети с локальными связями

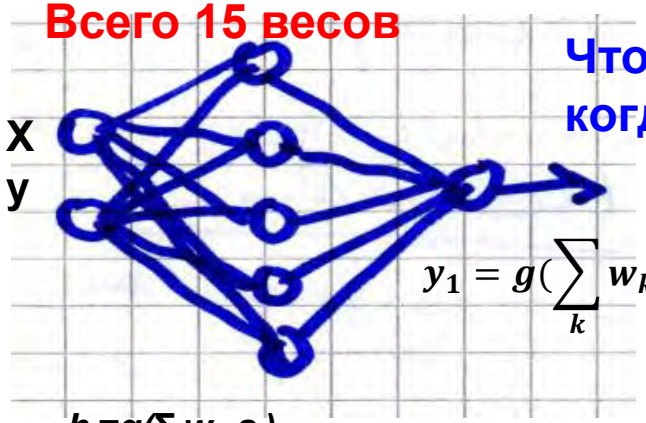


Тайны обучения. Что внутри черного ящика ИНС?



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:

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

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 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. После этого следует обязательно протестировать работу сети на тестовом множестве без меток.**

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

Почему ИНС так востребованы в физике

Нейросети с их способностью к обучению и самообучению явились весьма эффективным средством решения многих экспериментальных задач, так что **физики накопили большой опыт в применении ИНС** во многих экспериментах для распознавания изображений, траекторий элементарных частиц и проверки физических гипотез.

В физике были особенно популярны, в частности, МСП. Именно физики написали в 80-х программный **нейропакет JetNet** и были одними из первых пользователей **нейрочипов**.

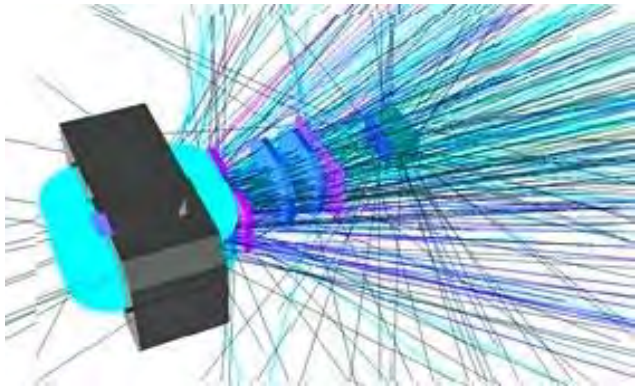
Причины исторические:

- **возможность монте-карловской генерации обучающей выборки любой требуемой длины на основе современной физической теории** ;
- появление в то время на рынке нейрочипов, реализующих обученную нейросеть для ее применения, как сверхбыстрого триггера;
- появление удобных в использовании программ для конструирования нейросетей и реализации их обучения типа черновской программы TMVA – the Toolkit for Multivariate Data Analysis with ROOT.

Причины современные:

- **Наступление эпохи Больших Данных.**
- **Новые технологические достижения:**
- **Суперкомпьютеры, карты GPU, облачные технологии.**
- **Биологические открытия о мозге**

Данные, измеренные в экспериментах и постановки задач -1



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-2

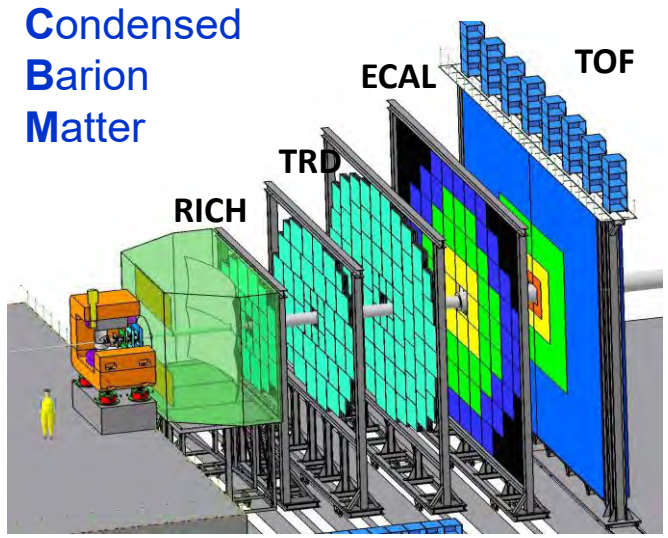


Схема установки CBM

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

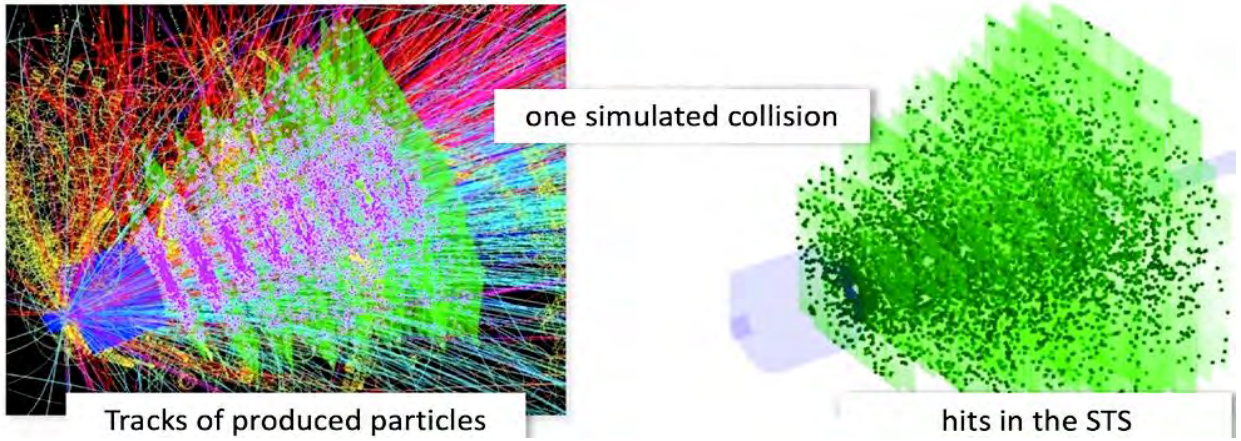
Скорость передачи данных:

10^7 событий в сек,

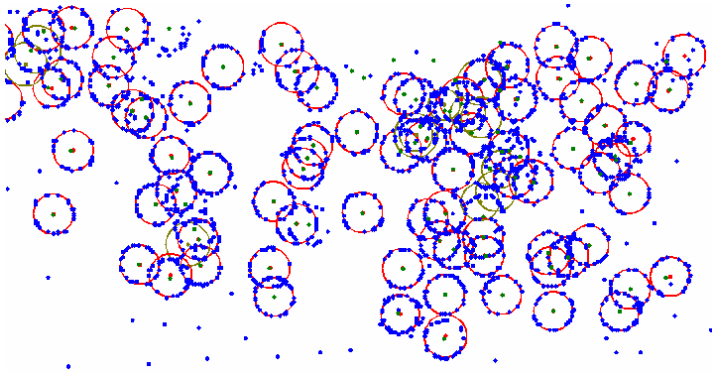
~1000 треков на событие

~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

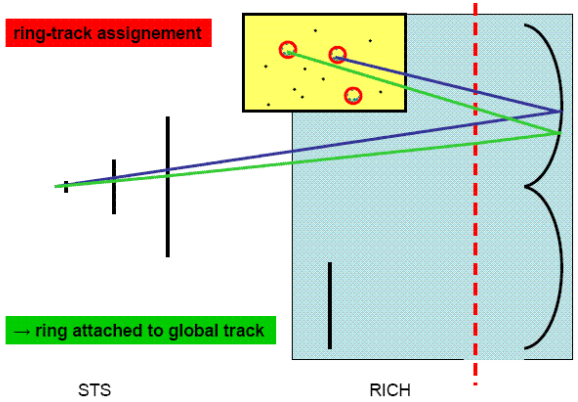
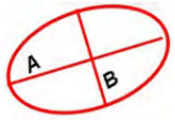


Схема детектораRICH черенковского излучения

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

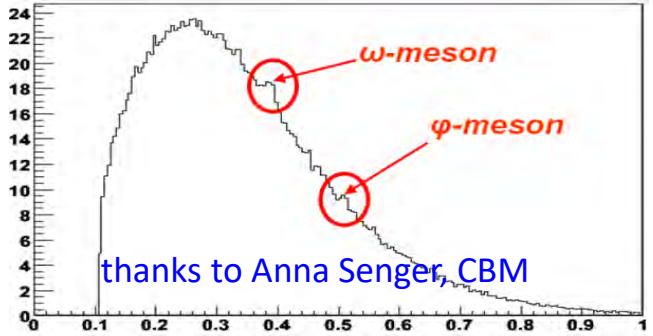
Основные этапы анализа данных в экспериментах ФВЭ

- 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)

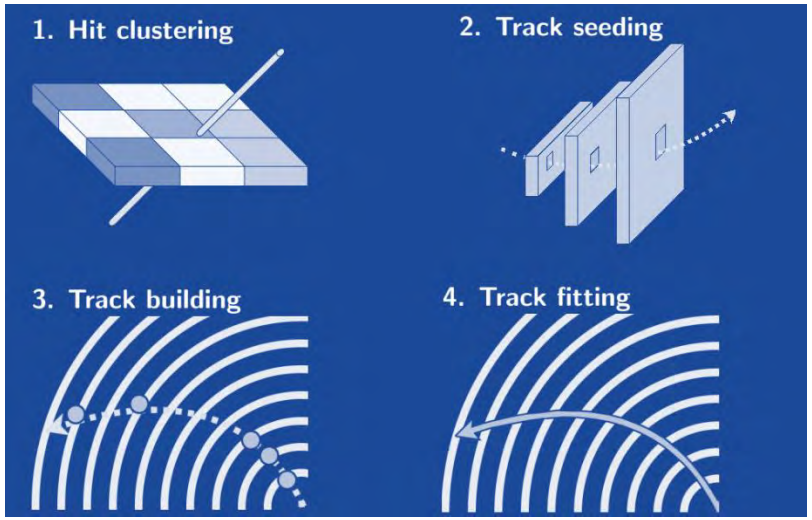
Parallel programming of optimized algorithms Grid-cloud technologies which changed considerably HEP data processing concept
See *Scientific data management in the coming decade* https://dl.acm.org/doi/10.1145/1107499.1107503

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.

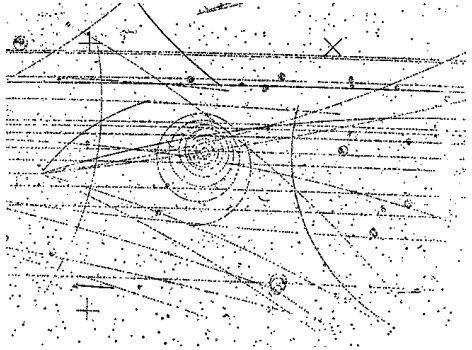


Процедура трекинга включает в себя фазы: (1) получения хитов (hit clustering), (2) построения треков-кандидатов - наборов хитов с вычисленными параметрами (*англ. seeds*), (3) прослеживания треков и (4) их подгонки уравнением движения частицы в магнитном поле.

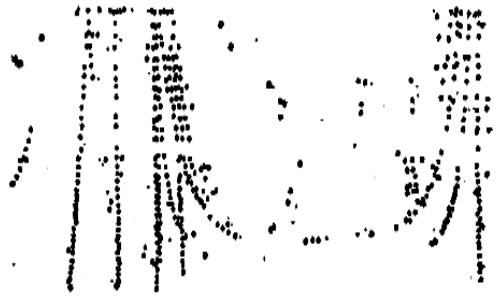
Главная проблема современного трекинга- высокая светимость пучков ускорителей, т.е. мегагерцовый темп поступления данных и банчевая структура пучка

Эволюция методов трекинга

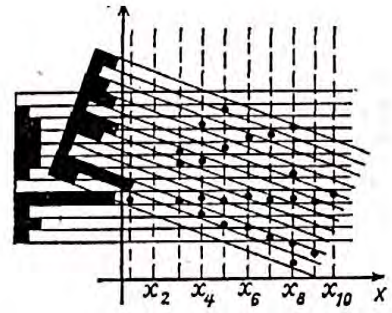
Началось еще в эпоху **пузырьковых камер**, когда события регистрировались на стереофотографиях и вводились в компьютер вручную, полуавтоматами или с помощью сканирующих устройств типа «Спиральный измеритель», в котором оператор ставил точку в вершину события, откуда шло сканирование снимка по спирали



Снимок события.



Его оцифровка в полярных коорд.

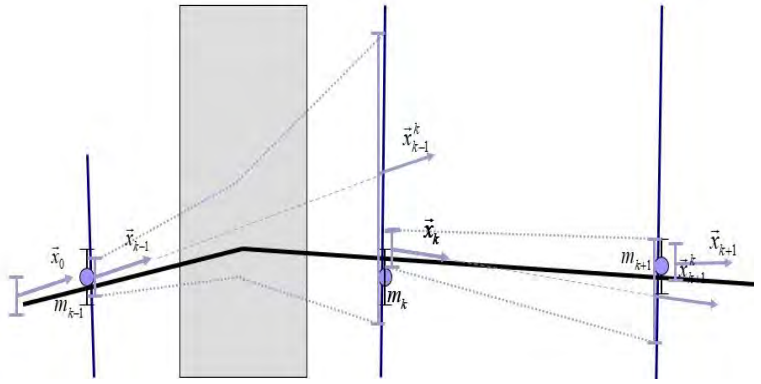


Поворотные гистограммы

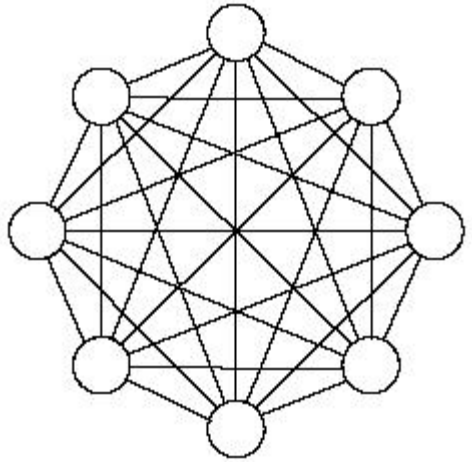
Когда пришла **эра электронных экспериментов**, данные измерений стали оцифровываться и сразу поступать прямо в компьютер. После многоэтапной фильтрации и процедур алайнмента, наступало время трекинга. Среди многих методов трекинга, самым эффективным оказался метод, использующий **фильтр Калмана**, поскольку 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



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

$$E(s) = - ½ \Sigma_{ij} s_i w_{ij} s_j.$$

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 $v_i = < s_i >T$ усредняются по температуре T. Эволюция сети определяется уравнением динамики среднего поля: $v_i = 1/2(1+\tanh(-\partial E/\partial v_i \, 1/T) = 1/2(1+\tanh(H_i /T))$,

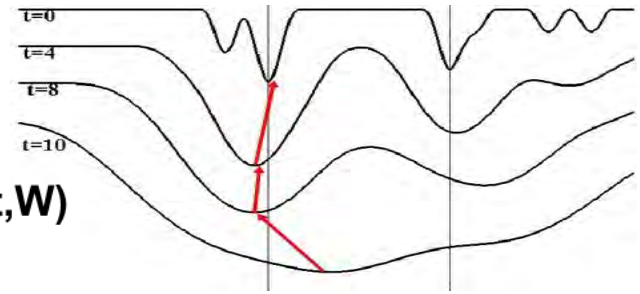
где $H_i = <\Sigma_j w_{ij} s_j >T$ – локальное среднее поле нейрона.

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).

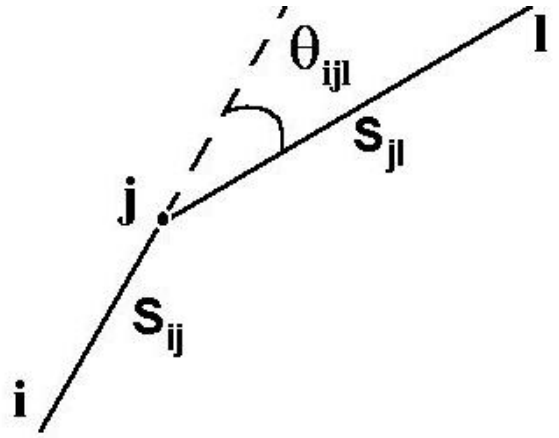
$$g(u) = \frac{1}{1+e^{-\lambda u}}$$

λ=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}

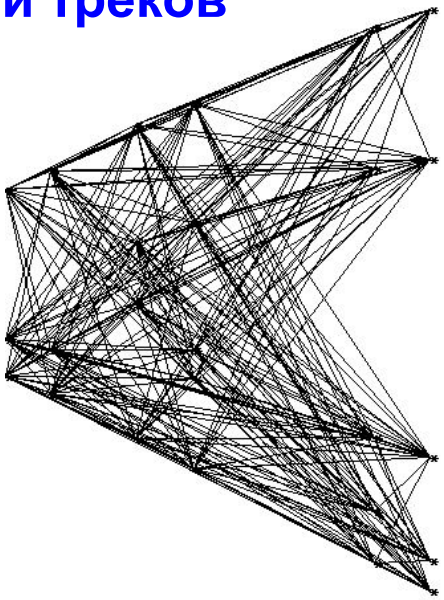
Вводится нейрон s_{ij} как направленный сегмент, соединяющий точки i, j ..

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.**

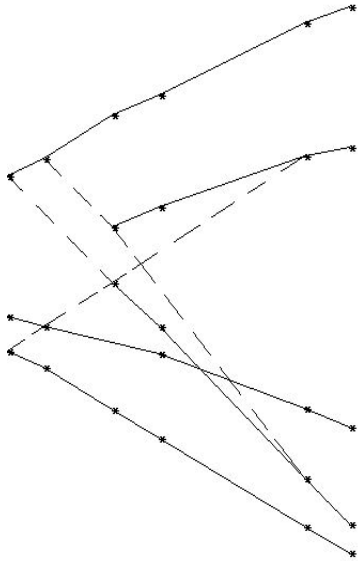
Пример применения для распознавания событий с короткоживущими частицами

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

Пример применения МСП в ФВЭ: идентификация частиц по данным детектора RICH

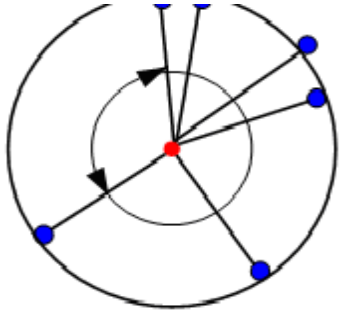
Следовало выбрать наиболее значимые характеристики полученных колец, чтобы

- **удалить ложно найденные кольца, - это делает первая нейросеть, после которой среди «хороших» колец вторая нейросеть выполняет**
- **идентификацию частиц**

В итоге выбрали 10 характеристик:

1. количество точек в найденном кольце
2. расстояние от центра кольца до ближайшего трека
3. сумма трех наибольших углов между соседними точкам
4. радиальная позиция на плоскости фотодетектора
5. χ^2 эллиптической подгонки кольца
6. большая (A) и (8) малая (B) полуоси эллипса
7. угол поворота эллипса φ относительно оси абсцисс
8. азимутальный угол трека
9. импульс трека

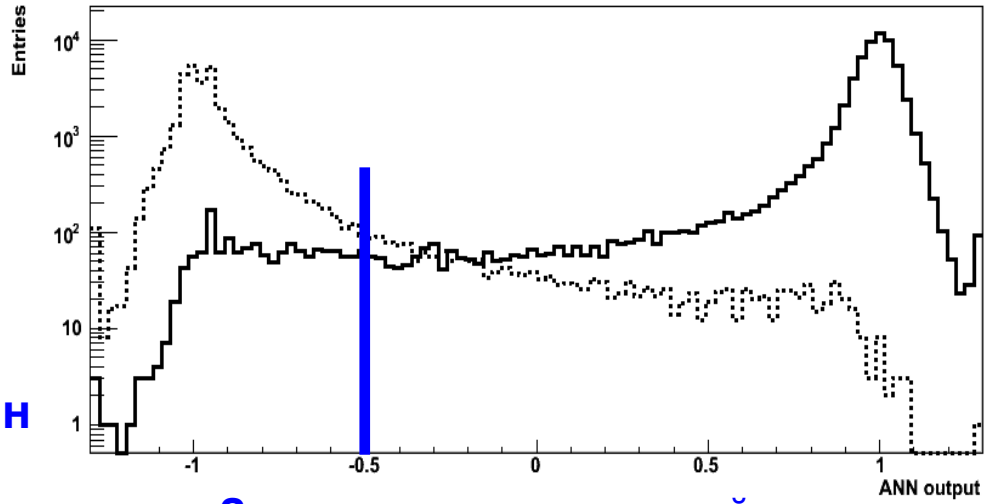
Первые 5 признаков использовались для ввода в первую нейросеть



Обучение 1-й сети велось на Монте-Карло выборке из событий с 3000 электронов (+1) и 3000 пи-мезонов (-1) и показало 93%-й эффективность отбора хороших колец

Вторая ИНС имела все 9 входных нейронов, 20 скрытых и один выходной нейрон, обучалась на выборке из хороших колец

Порог -0.5 для выхода сети обеспечил идентификацию электронов с приемлемым уровнем подавления пионов



Значения выходного нейрона Для идетификации частиц

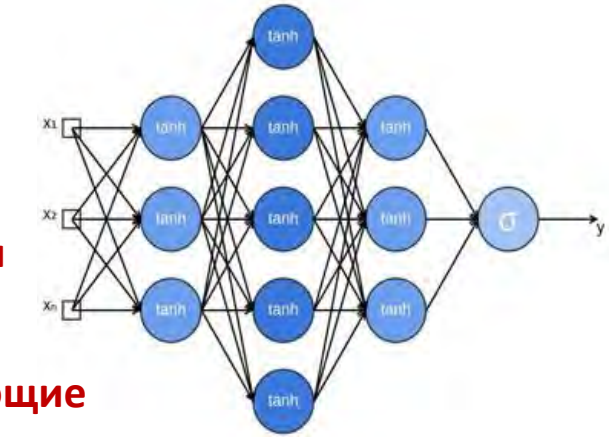
Различные типы глубоких нейросетей и проблемы их обучения

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:

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

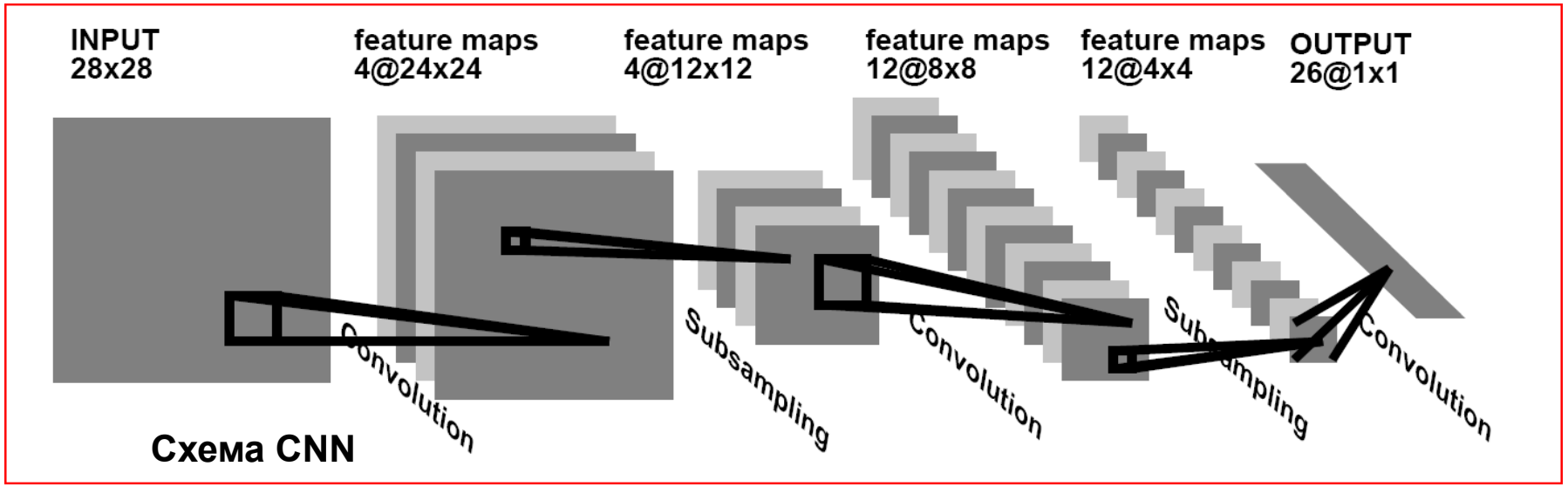


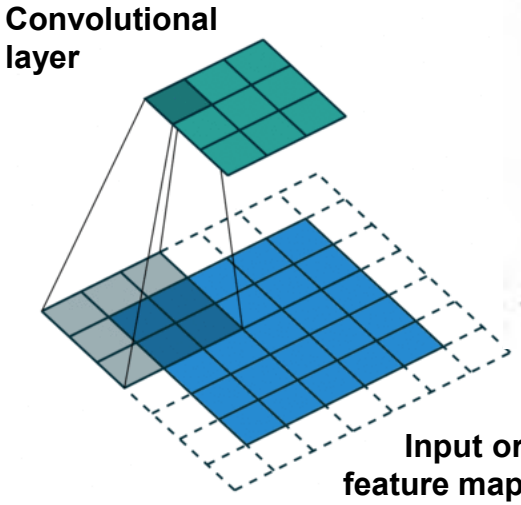
Схема CNN

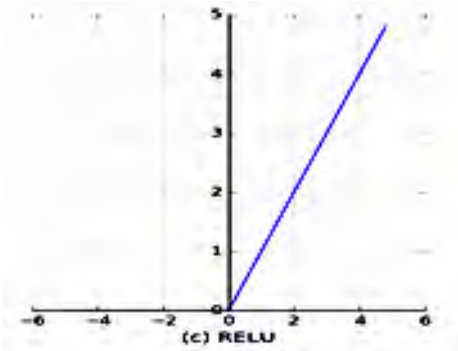
Поясняющая аналогия

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

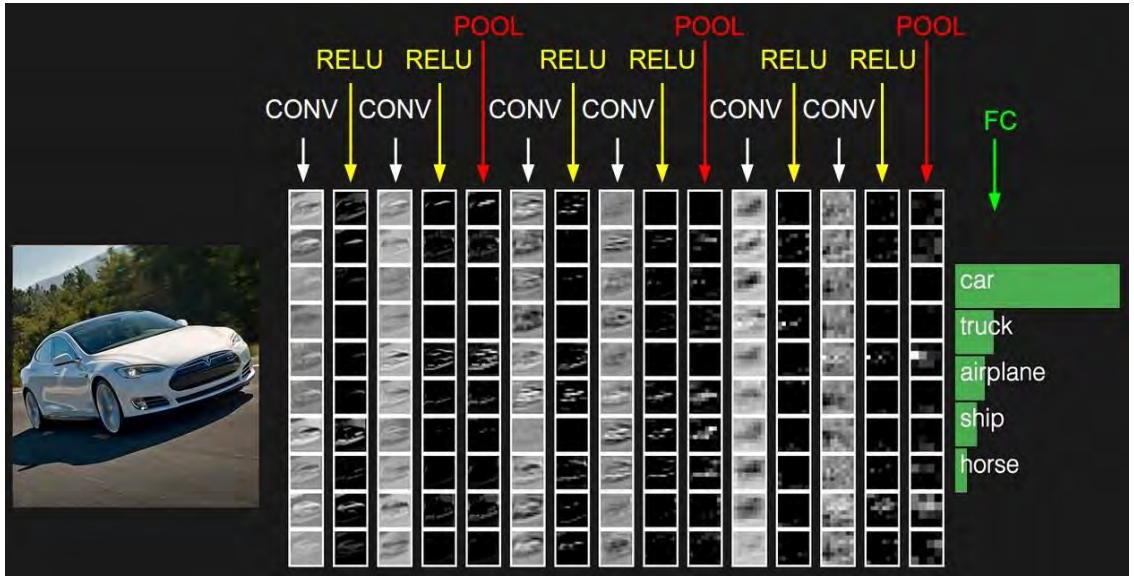
Основы архитектуры CNN

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) .**

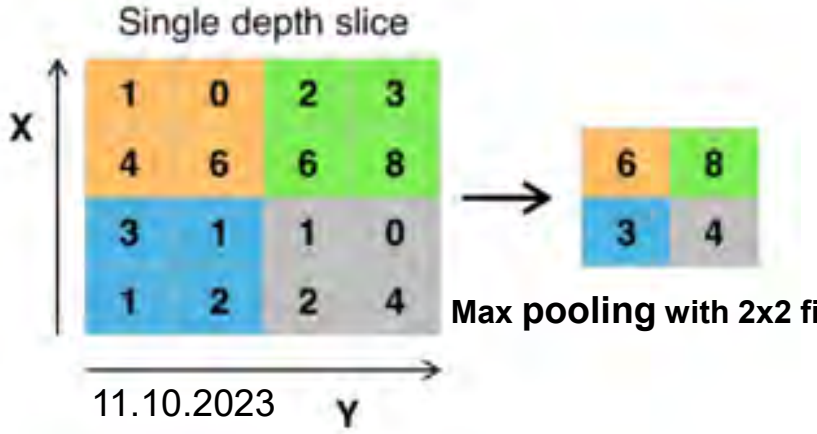




Activation: ReLU



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. Обучение с подкреплением

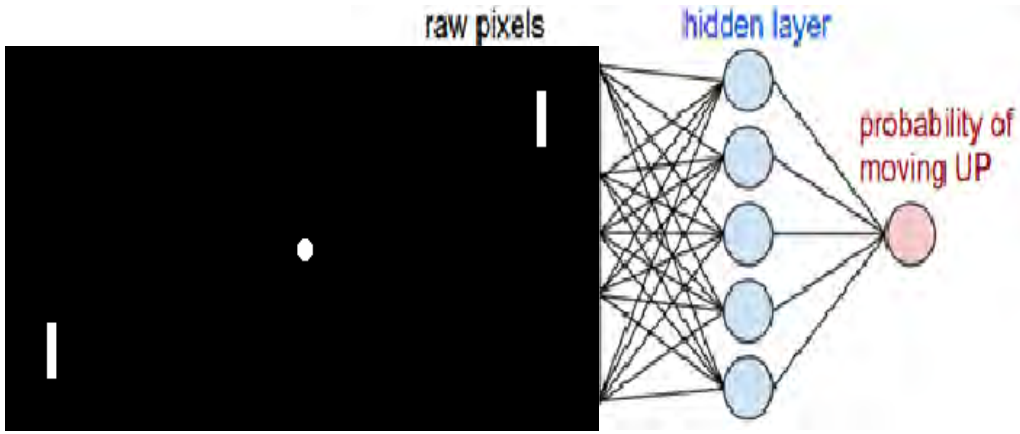
До сих пор нейросети обучались на тренировочной выборке с «правильными» ответами либо с учителем, либо самообучались, но в реальной жизни мы редко знаем набор правильных ответов, а просто делаем то или иное действие и ведем себя дальше в зависимости от полученного результата.

Отсюда идея обучения с подкреплением Reinforcement learning, - нейросетевой агент, находясь в состоянии S, взаимодействует с окружающей средой, а она его за эти действия поощряет и сообщает в какое состояние агент после этого перешел так, чтобы увеличивать общую награду.

Агент не знает, какое действие предпринять, как при обучении с учителем, но зато узнает, какое действие принесет максимальное поощрение. Действия могут повлиять не только на немедленное вознаграждение, но и на следующую ситуацию и все последующие награды. Такое обучение является частным случаем обучения с учителем, но учителем является среда или её модель.

Пример. Игра в пин-понг

http://karpathy.github.io/2016/05/31/rl/



Две основные цели обучения в практических применениях

1. минимизировать ошибки. Машина учится анализировать информацию перед каждым следующим ходом в виртуальной модели города со случайными пешеходами и другими участниками дорожного движения.

2. получить от выполнения задания максимальную выгоду, - максимально быстрое время прохождения маршрута, оптимальный расход ресурсов, обслуживание как можно большего количества клиентов.

Области практического применения reinforcement learning:

роботы, беспилотные авто, трейдинговые боты для игры на бирже, чат боты, которые учатся от диалога к диалогу, разработка игровых программ

7. Генеративные Состязательные Сети - Generative Adversarial Networks (GAN)

GAN реализует принцип состязания между сетью генерации и сетью дискриминации. Генеративная сеть **G** генерирует максимально реалистичный образец, а дискриминативная сеть **D** обучается различать подлинные и поддельные образцы

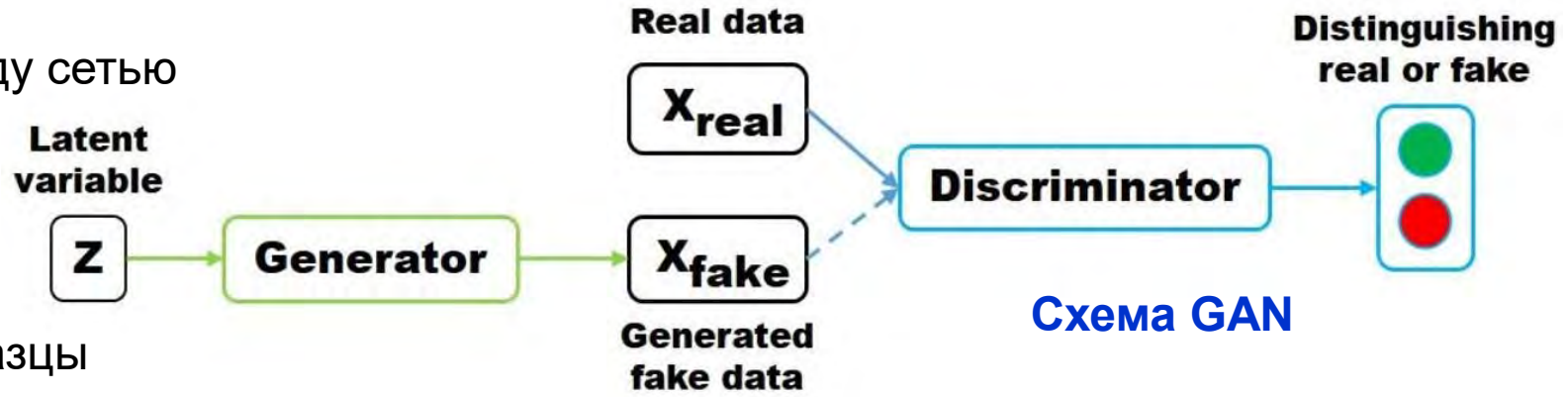


Схема GAN

Потом результаты различения подаются на вход сети G так, чтобы она смогла подобрать лучший набор латентных параметров, смешивая исходные образцы, чтобы сеть D **уже не смогла бы отличить подлинные образцы от поддельных.** Сеть D реализуется как свёрточная нейронная сеть, в то время как сеть G наоборот разворачивает изображение на базе скрытых параметров.

Применения GAN

- получения фотореалистичных изображений и картин;
- автоматическое редактирование изображений
- создание фильмов и мультипликаций.
- создание трёхмерной модели объекта с помощью фрагментарных изображений
- **моделирование сложных физических процессов в детекторах экспериментальной физики**

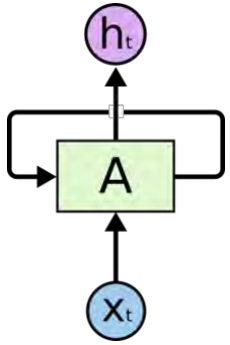
Этические проблемы GAN приложений!

Опасность дипфейков: политика, мошенничество и шантаж с использованием дипфейков

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

Изменяющийся мир требует рекуррентных нейросетей

- **В жизни мы имеем дело с объектами, изменяющимися во времени, и наш мозг, работая, 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.**



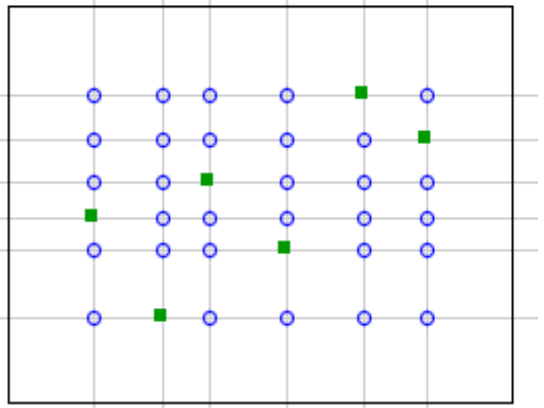
Эту проблему решает семейство новых глубоких нейросетей, называемых рекуррентными (Recurrent Neural Networks - RNN). Они содержат в себе обратные связи, позволяющие сохранять прошлую информацию. Так модуль *A* принимает входное значение x_t и возвращает значение h_t. Внутри этой ячейки обычная нейросеть с одним скрытым слоем.

Чтобы RNN могла связывать предыдущую информацию с текущей задачей, требуется ее усовершенствование до LSTM сети (Long short-term memory - долгая краткосрочная память) с четырьмя взаимодействующих слоями, позволяющими удалять информацию из состояния ячейки с помощью фильтров - gates. Фильтры состоят из слоя сигмоидальной нейронной сети и операции поточечного умножения и позволяют обрабатывать информацию на основании задаваемых условий.

С помощью упрощенной версией LSTM GRU (Gated Recurrent Unit) нам удалось решить задачу восстановления траекторий элементарных частиц в детекторе GEM эксперимента BM@N в ОИЯИ.

Проблемы трекинга для GEM и полосковых детекторов

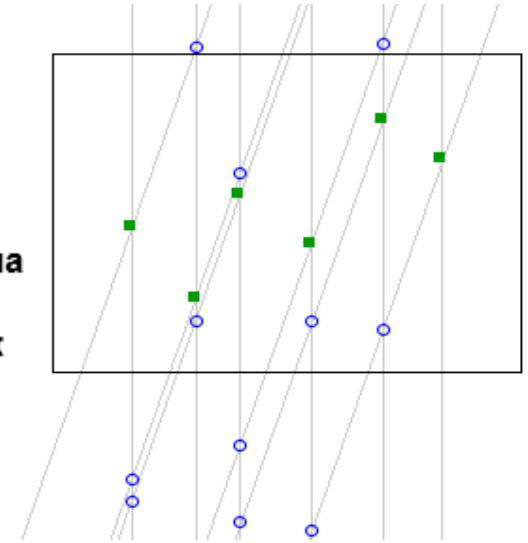
Главная трудность, вызванная спецификой **GEM детектора** – появление ложных отсчетов из-за лишних пересечений стрипов. Для **n** истинных хитов имеем **n^2- n фейков!**

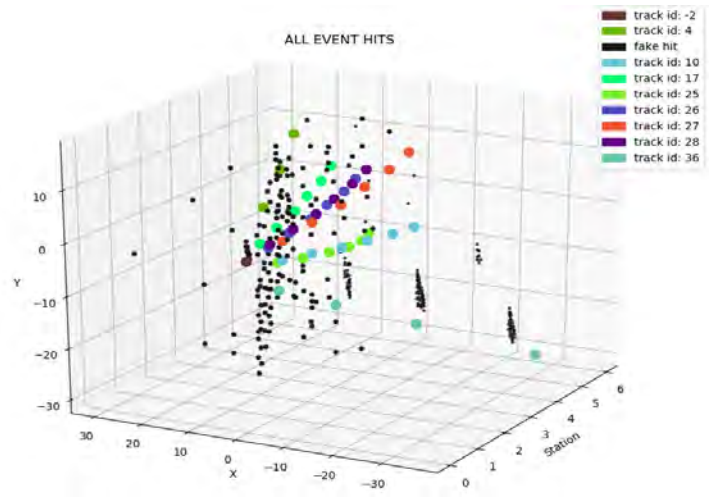


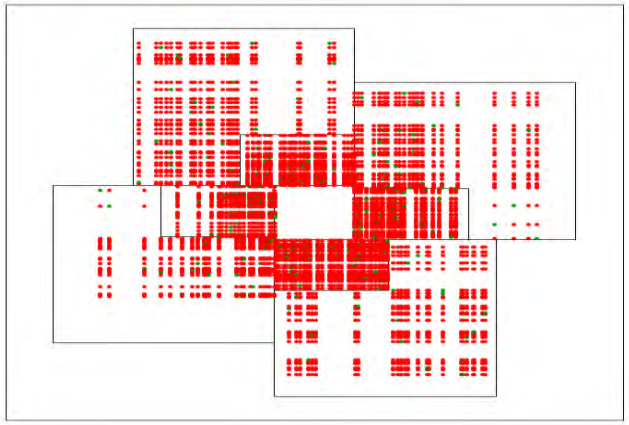
- истинный хит

- ложное пересечение

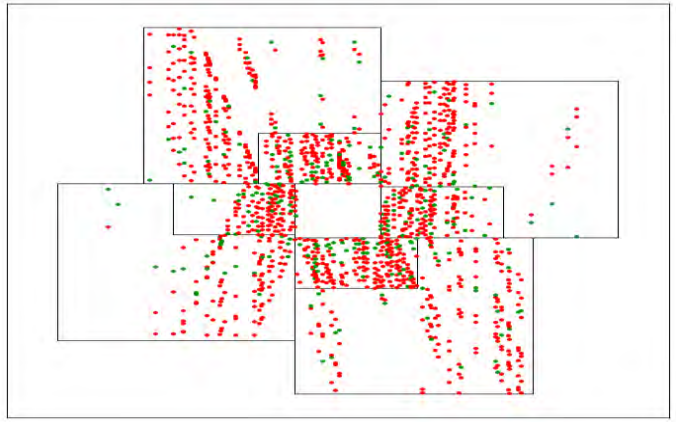
Можно уменьшить количество фейков – повернуть слой стрипов на маленький угол (5-15 градусов) по отношению к другому слою







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



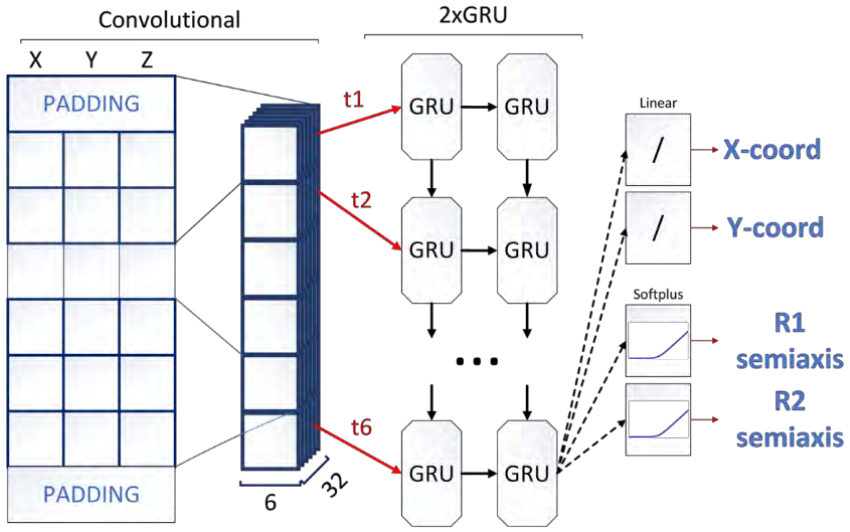
Две основных проблемы - наличие фейковых засорений данных и, главное, сверхвысокий темп их поступления из-за высокой светимости неизбежно требуют разработки новых методов трекинга с использованием глубоких нейронных сетей

Локальный и глобальный 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. Глобальный 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.

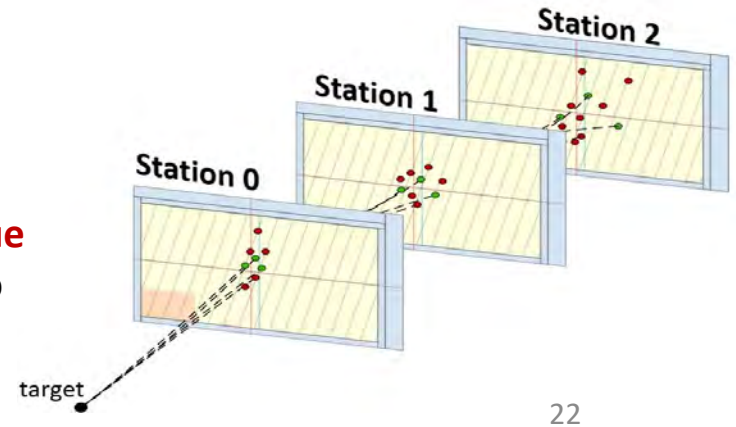


Scheme of the recurrent TrackNETv2 neural network

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

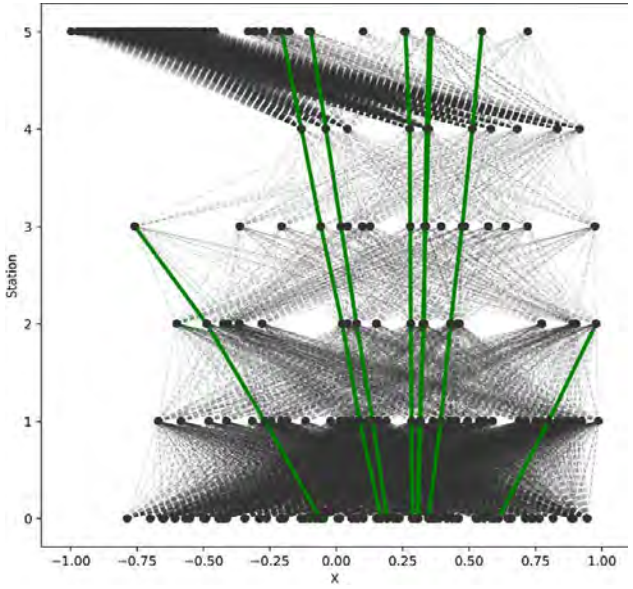


2. Глобальный трекинг

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.

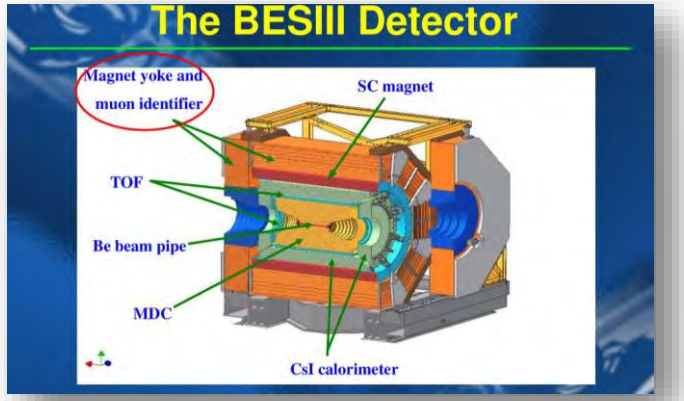


Графическое представление события C + C, 4 ГэВ эксперимента BM@N. Черные узлы и ребра соответствуют фейкам, зеленые узлы и ребра - найденным трекам

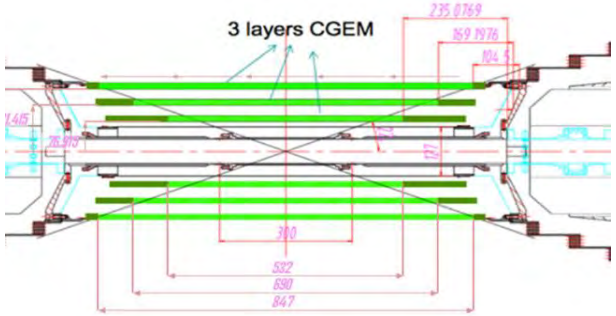
Эта схема похожа уже известный глобальный подход Денби-Петерсона с сегментной нейросетью Хопфилда, где нейросеть подолгу обучалась отдельно для каждого события, в то время как GNN, где надо найти те ребра, что являются сегментами реальных треков можно обучить на выборке из графов событий, где эти искомые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

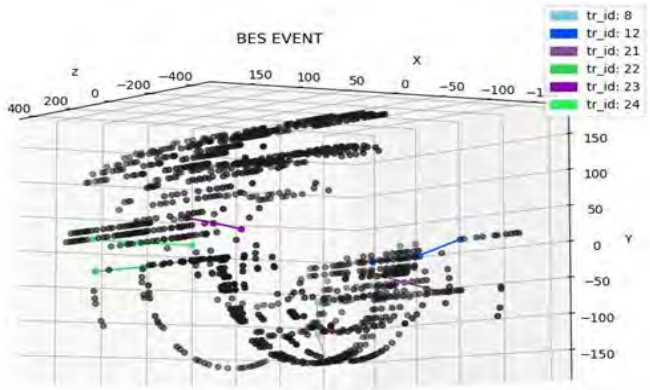
2.2. Применение графовых нейронных сетей, эксперимент BES-III



BESIII – коллайдерный эксперимент



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



Все хиты модельного события

Граф события <u>инвертируется</u> в линейный дограф, 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 дограф 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)**

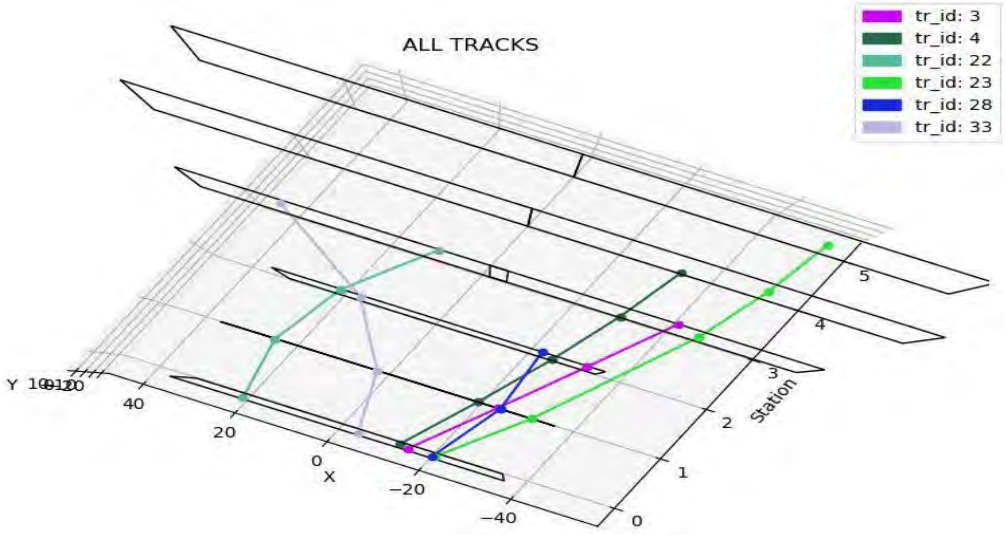
<u>Оценки эффективности трекинга</u>. Оценка **accuracy** как доля найденных треков к общему числу треков-кандидатов – бесполезна и даже опасна, т.к. наша выборка очень сильно несбалансирована. Принято использовать две метрики – **recall и precision. Recall** – это доля истинных треков, которые модель 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.

2.3. Глобальный 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.

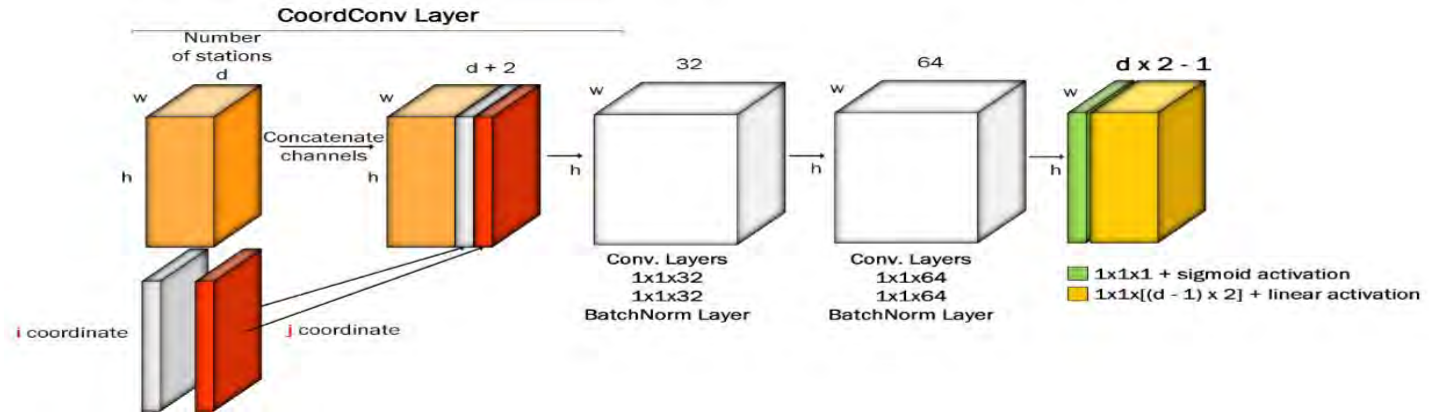
- **В CNN Изображения имеют формат 3d: высота + ширина + RGB;**
- **У нас данные с каждой станции - разреженная матрица нулей и единиц, где единицы указывают на появление хитов;**
- **События также имеют формат 3D: Высота + Ширина + Станции.**

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

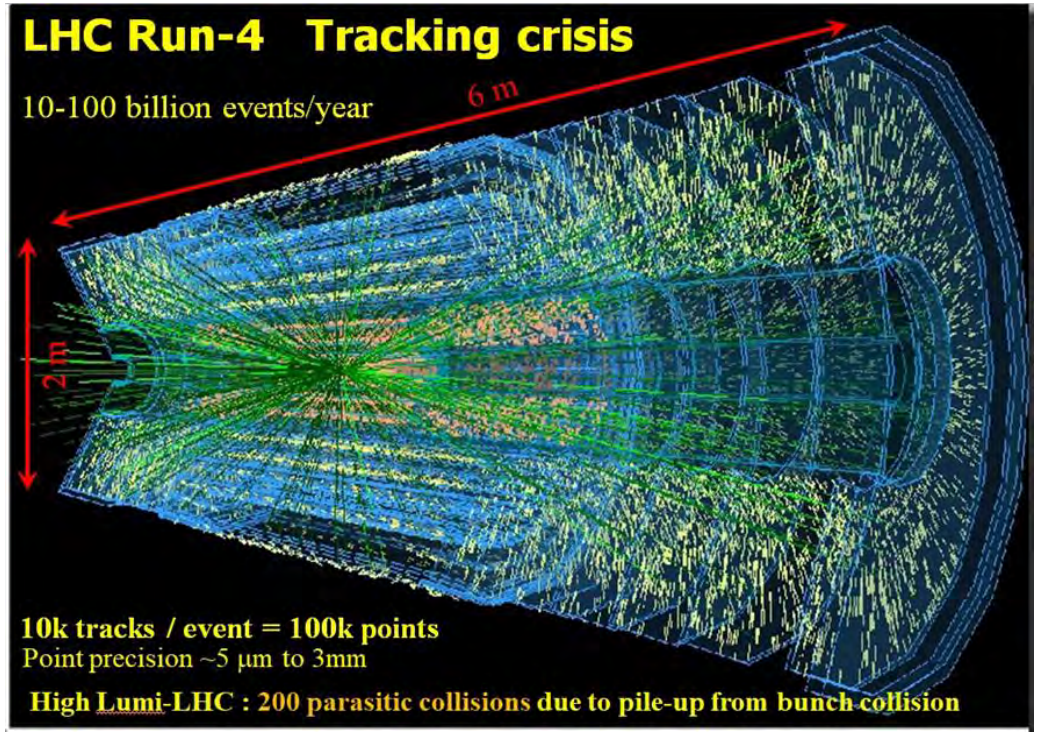
Используется новая нейросетевая модель **Look Once On Tracks (LOOT)**

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, причем время работы обученной модели не зависит от множественности события

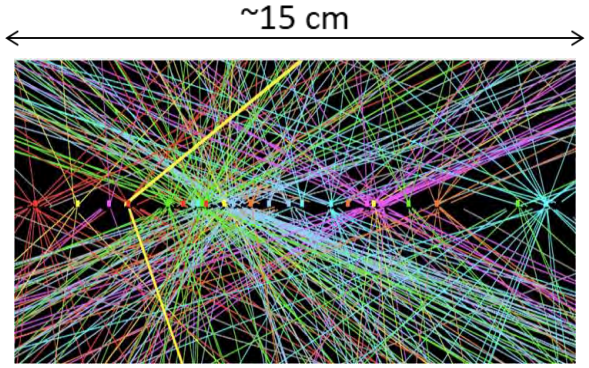
Эксперименты с высокой светимостью. Кризис трекинга



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, 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)



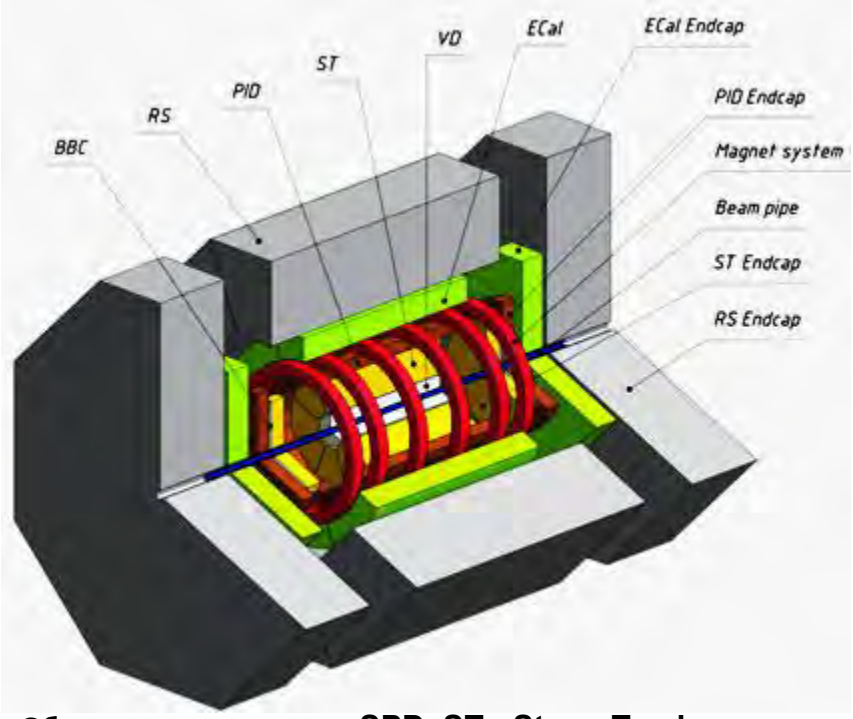
Поэтому моменты столкновений происходят так близко, что треки событий сильно перекрываются в 15 сантиметровой области встречи пучков.

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

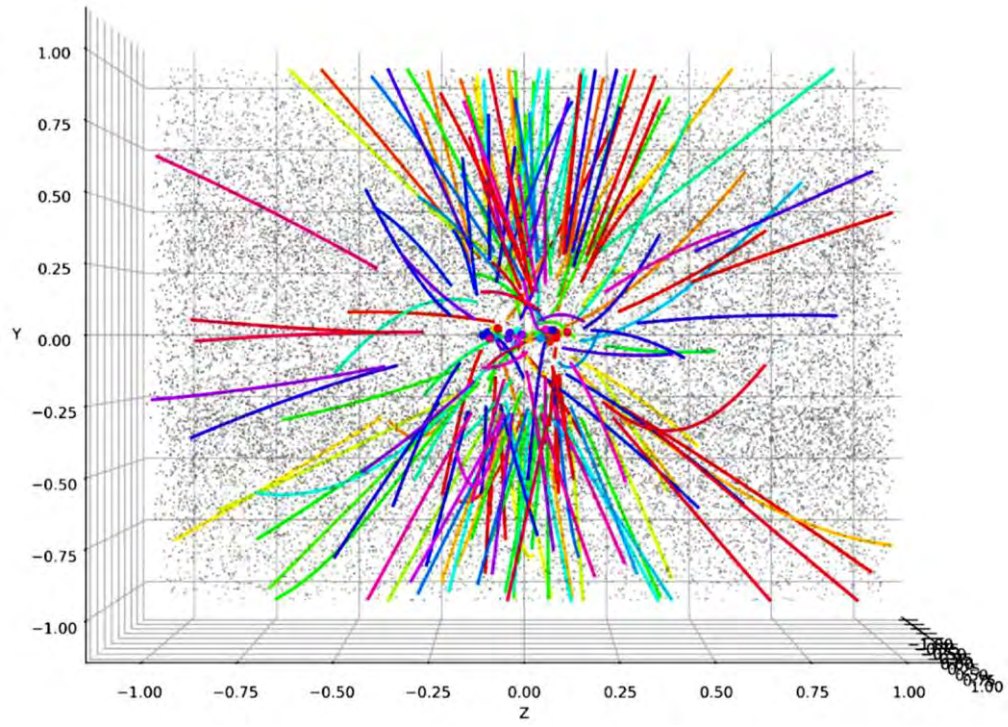
Трекинг для данных экспериментов высокой светимости. SPD NICA

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

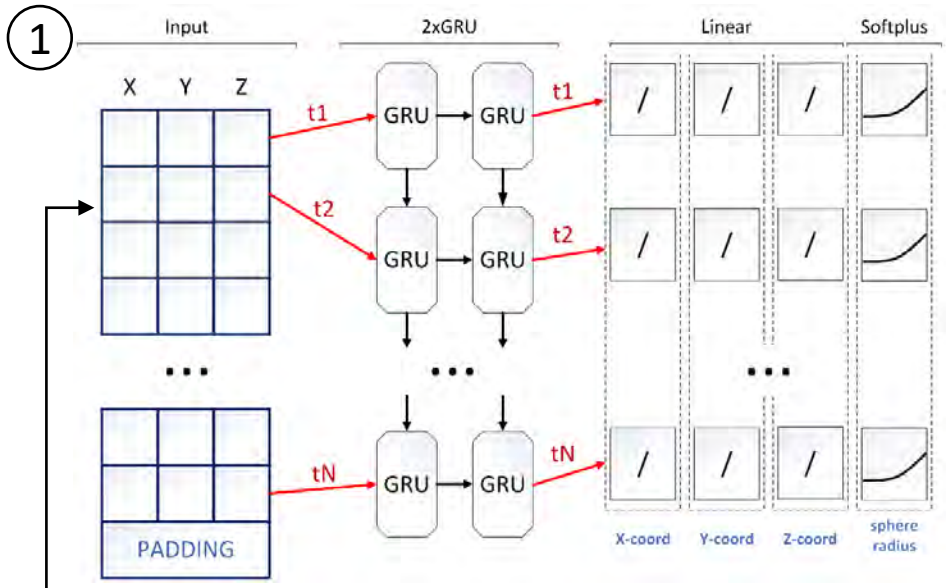
Основные проблемы при трекинге это “лево-право” неопределенность строу-трубок, огромное количество фейковых сигналов и пропуски отсчетов из-за неэффективности детекторов.
Внесение соответствующих усложнений в программу TrackNET неизбежно замедляет ее работу и снижает эффективность



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

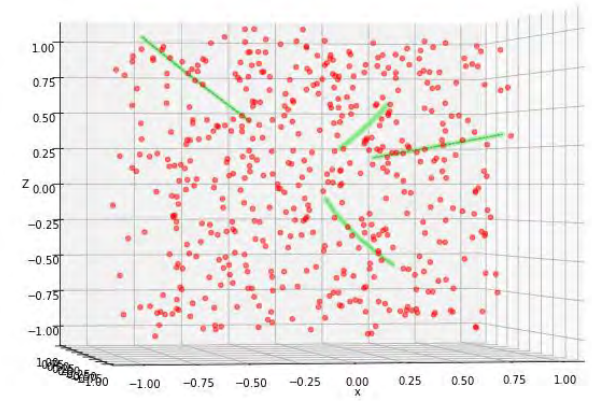
SPD тайм-слайс Tracking with TrackNET. Inference optimisation

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

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

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 ~ 2500 событий в секунду

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

Г.А.Ососков

О машинном обучении и его применении к задачам физики высоких энергий

Спасибо за внимание!

Могу ответить на вопросы также по почте

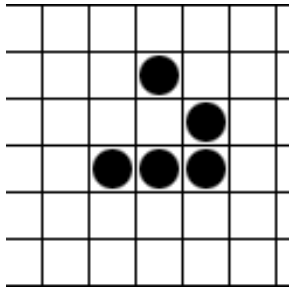
email: gososkov@gmail.com

Много материалов на моем сайте

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).