

Министерство образования и науки Российской Федерации
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Иркутский государственный университет»
(ФГБОУ ВО «ИГУ»)
Физический факультет

Кафедра радиопизики и радиоэлектроники
Допускается к защите
И.О. зав. кафедрой,
Доцент, к.ф.м.н. _____ Колесник С.Н.
«_____» _____ 20__ г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА
по направлению 03.03.03 Радиофизика
направленность (профиль)
«Телекоммуникационные системы и информационные технологии»

ПРИМЕНЕНИЕ НЕЙРОСЕТЕЙ ДЛЯ ВОССТАНОВЛЕНИЯ
РАДИОСИГНАЛОВ ОТ ШИРОКИХ АТМОСФЕРНЫХ ЛИВНЕЙ

Нормоконтролёр:

(подпись) Ланин Б.Ю.

Студент 4 курса очного отделения,
Группа 01421 - ДБ

(подпись) Шипилов Дмитрий Андреевич
Руководитель: д.ф.м.н., профессор

(подпись) Афанасьев Н.Т.
Консультант: к.ф.м.н., СНС

(подпись) Казарина Ю.А.

Работа защищена:
«_____» _____ 20__ г.
С оценкой _____
Протокол № _____

Иркутск 2018

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1. Широкие атмосферные ливни и Tunka-Rex	4
1.1 Широкий атмосферный ливень (ШАЛ).....	4
1.2 Излучение Вавилова-Черенкова.....	5
1.3 Основные механизмы радиоизлучения ШАЛ.....	6
1.4 Установка Tunka-Rex	7
1.5 Восстановление параметров сигнала	10
2. Нейронные сети и автоэнкодеры.....	12
2.1 Искусственные нейронные сети.....	12
2.2 Сверточные нейронные сети	14
2.3 Сети типа «автокодировщик».....	16
2.4 Проблемы переобучения и регуляризация.....	18
2.5 Используемые метрики качества	19
3. Практическая часть	22
3.1 Стратегия обучения	22
3.2 Конвертер данных из ROOT	23
3.3 Обучение и оценка.....	24
3.4 Результаты	26
Рис. 6. Диаграмма рассеяния	28
Рис.7. Порог восстановления.....	28
ЗАКЛЮЧЕНИЕ	30
Список использованных источников	31
Приложение 1	32
Приложение 2	34

ВВЕДЕНИЕ

Космические лучи являются уникальным природным источником частиц высоких и сверхвысоких энергий и позволяют получать информацию как о микромире – природе и взаимодействиях элементарных частиц и ядер, так и о крупномасштабных процессах, происходящих не только в нашей Галактике, но и за ее пределами. Несмотря на то, что с момента открытия первичных космических лучей (ПКЛ) прошло более 100 лет, вопросы, связанные с этим явлением, не теряют своей актуальности. Постоянно уточняются и углубляются наши представления о составе ПКЛ, происхождении, механизмах распространения в межзвездной и межпланетной средах, характере ядерных взаимодействий, в которых они участвуют. Одним из основных способов исследования ПКЛ является изучение широких атмосферных ливней (ШАЛ).

В последнее время активно развивается радиометод исследования ШАЛ, так как имеет значительные преимущества – независимость от погодных условий и относительная легкость в изготовлении и размещении детекторов.

В эксперименте Tunka-Rex, действующем с 2012 года, детектирование радиоизлучения производится с помощью антенных станций, подключенных к центрам кластеров черенковских детекторов Tunka-133 и антенных станций, подключенных к сцинтилляционным детекторам Tunka-Grande.

В процессе измерения исследователи сталкиваются с влиянием внешних шумов различного характера, затрудняющих получение необходимой информации из сигналов.

Основной целью этой работы являлась разработка программного обеспечения, основанного на нейронных сетях типа «автокодировщик», позволяющих решать задачу восстановления радиосигналов от ШАЛ.

1. Широкие атмосферные ливни и Tunka-Rex

1.1 Широкий атмосферный ливень (ШАЛ)

При прохождении космических лучей через атмосферу Земли в результате ядерных и электромагнитных взаимодействий происходит каскадное рождение элементарных частиц – развивается широкий атмосферный ливень [1]. Это название связано с тем, что вторичные частицы, возникающие в результате взаимодействий и распадов, могут быть зарегистрированы на достаточно больших расстояниях от оси ШАЛ – прямой, совпадающей с направлением движения первичной частицы. В зависимости от первичной энергии, регистрация ШАЛ может происходить на расстояниях порядка сотен и даже тысяч метров от оси, так что эффективная площадь может достигать десятков квадратных километров. В процессе развития ливня в атмосфере число вторичных частиц увеличивается до глубины, где средняя энергия вторичных частиц равняется критической энергии. Ниже этого уровня число частиц уменьшается приблизительно по экспоненте. Глубина атмосферы, на которой в каскаде достигается наибольшее число заряженных частиц, называется глубиной максимума ливня [2].

В ходе развития каскада образуются компоненты ШАЛ – адронная, электронно-фотонная, мюонная, нейтринная, а также генерируются оптические (черенковское и флуоресцентное) и радио излучения. Наиболее многочисленными среди заряженных частиц ШАЛ являются электроны и позитроны. Количество мюонов составляет примерно 10% от числа электронов. Число гамма-квантов примерно в два раза превышает число электронов, а адроны составляют около 1% от полного числа частиц в ШАЛ. Именно вторичные частицы ШАЛ, а также излучения, создаваемые заряженными частицами ливня, и используются для исследования ПКЛ высоких, сверхвысоких и предельно высоких энергий. Размеры ливня зависят

от энергии первичной частицы. Для первичной частицы с энергией $E_0 > 10^{16}$ эВ характерный поперечный размер ливня на поверхности Земли достигает нескольких сотен метров, а число составляющих его вторичных частиц, в основном электронов и гамма-квантов, – более 10^7 .

1.2 Излучение Вавилова-Черенкова

Черенковское излучение возникает при движении заряженной частицы в прозрачной среде со скоростью v большей скорости света в этой среде, т.е. при $v > c/n$, где c – скорость света в вакууме, а n – показатель преломления среды. Черенковское излучение является совместным излучением множества атомов среды, расположенных вдоль траектории движения частицы и поляризованных её электрическим полем. Таким образом, непосредственно излучает не частица, а среда, в которой движется частица. Волновой фронт этого излучения представляет собой поверхность конуса, вершиной которого является частица, а осью – её траектория. Угол раствора конуса θ фиксирован и определяется скоростью частицы v и свойствами среды (n – показатель преломления среды, $\theta = c/vn$.) [3].

Энергия частицы, конвертируемая в черенковское излучение, мала по сравнению с энергией, которую она тратит на ионизацию и возбуждение атомов среды. Число фотонов, излучаемых на 1 см пути, в зависимости от среды колеблется от нескольких единиц до нескольких сот. Это излучение можно наблюдать визуально и регистрировать с помощью фотоплёнки или фотоэлектронного умножителя (ФЭУ).

1.3 Основные механизмы радиоизлучения ШАЛ

Радиоизлучение ШАЛ это сложное явление. Теоретические модели выделяют несколько видов генерации радиоизлучения: черенковское излучение, эффект Аскарьяна, геоэлектрический механизм, геомагнитный механизм и др. На сегодняшний день обнаружено, что основным видом генерации является геомагнитный механизм, который вносит наибольший вклад в радиоизлучение (~90%).

Когда частица ультравысокой энергии входит в верхние слои атмосферы, развивается каскад вторичных частиц высоких энергий [4]. Из-за высоких скоростей большая часть частиц сосредоточена в относительно тонком фронте ливня, который называют диском. Диск, который предполагается электронейтральным, содержит большое количество электронов и позитронов. Близко к ядру ливня диск имеет типичную толщину порядка метра и движется к поверхности Земли со скоростью близкой к скорости света. Релятивистские электроны и позитроны в диске под действием силы Лоренца отделяются друг от друга из-за магнитного поля Земли.

Это накладывает на независимое от заряда поперечное перемещение частиц направленное движение, перпендикулярное и к оси ливня, и к магнитному полю, причем электроны и позитроны движутся по взаимно противоположным направлениям. Такое явление имеет формальное сходство с поведением свободных электронов в проводнике при создании в нем постоянной разности потенциалов.

В обоих случаях такое направленное движение частиц представляет собой электрический ток, значение которого определяется величиной заряда, прошедшего через условную площадь поперечного сечения. Такой поперечный ток в ливне можно рассматривать как причину поляризации нейтрального в целом диска. При движении в магнитном поле диск

позитронов и диск электронов раздвигаются во взаимно противоположных направлениях, образуя электрический диполь, направление которого совпадает с направлением поперечного тока. На том участке движения ливня, где число частиц близко к максимальному, можно считать, что в поперечном направлении имеет место динамическое квазиравновесие, при котором число рождающихся в стволе частиц равно числу частиц, выбывающих из ливня.

Поэтому широкий атмосферный ливень можно уподобить движущемуся квазистатическому диполю, в котором поперечный ток поддерживает дипольный момент.

Поляризация геомагнитного радиоизлучения перпендикулярна к магнитному полю Земли, поэтому, излучение наблюдается преимущественно в направлении восток-запад.

1.4 Установка Tunka-Rex

На данный момент в Тункинской долине активно ведут работу следующие установки:

- Tunka-133, изучает черенковское излучение ШАЛ в диапазоне энергий $10^{15} - 10^{18}$ эВ
- Tunka-Grande, занимается сбором данных о мюонной и электронной компонентах ШАЛ
- Tunka-HiSCORE, ориентированна на черенковское излучение от ШАЛ, однако с порогом в несколько раз ниже, чем Tunka-133.
- Tunka-Rex

Основная цель Tunka-Rex — выяснить предел точности восстановления параметров породившей каскад частицы по измерениям радиоизлучения ливня и, таким образом, понять возможности использования радиоантенн как альтернативы классическим методам исследования ШАЛ с помощью черенковского излучения.

К преимуществам Tunka-Rex стоит отнести низкий уровень шума в Тункинской долине, низкую стоимость антен и простоту их установки.

Совместные измерения с установкой Тунка-133 дают возможность проведения калибровки детекторов [5]. В частности, с помощью черенковских данных может быть проверена точность восстановления энергии и положения максимума, полученных по радиоизмерениям. Таким образом, это уникальная не имеющая аналогов гибридная система по регистрации черенковского света и радиоизлучения от ШАЛ.

В свою очередь, радиоантенны, в отличие от детекторов черенковского света, не зависят от атмосферы условий и погоды условий, они намного более просты в установке и обслуживании. С 2014 года было установлено 19 дополнительных антенных станций, подключенных к станциям сцинтилляторов. Это позволит существенно увеличить эффективное время работы радиоустановки [6].

Установка Tunka-Rex (Рис. 1) детектирует радиоизлучение ШАЛ в диапазоне 30-80 МГц.



Рис.1. Экспериментальная установка Tunka-Rex

Данный диапазон является оптимальным для регистрации когерентного радиоизлучения ШАЛ. Ограничение снизу обусловлено интенсивным использованием коротковолнового диапазона вещательными, любительскими и профессиональными радиостанциями.

Сверху — характерными размерностями распределения зарядов в стволе ШАЛ, определяющих область когерентности радиоизлучения [7].

Размер области распространения ливня определяется плотностью среды. В атмосфере поперечные размеры ливня составляют примерно 100 м, а длины излучаемых волн — несколько метров. Кроме того, на частотах до 20 МГц из-за наличия больших естественных радиошумов преимущественно грозового происхождения не представляется возможным выделять радиоимпульсы ШАЛ с достаточной эффективностью.

В качестве детектирующего элемента Tunka-Rex используется 2 петлевые антенны типа SALLA (Short Aperiodic Loaded Loop Antenna), ориентируемые на северо-западное и северо-восточное направление. Такой тип антенны обладает двумя основными преимуществами: данные антенны имеют низкую цену и просты в изготовлении, их диаграмма направленности мало зависит от подстилающей поверхности.

Для восстановления сигнала используется следующая схема: в начале производится преобразование отсчётов АЦП в напряжение. Затем, используя измеренные параметры усилителей и кабелей, восстанавливается напряжение, наведённое радиоимпульсом на антенну.

Для сравнения результатов эксперимента Tunka-Rex с результатами других радио детекторов или моделирования, электрическое поле должно быть переведено в абсолютные единицы.

Преобразование входного сигнала может быть описано с помощью свертки в частотной области выходного сигнала с передаточными характеристиками тракта. Далее из входного сигнала, используя данные о диаграмме направленности, можно получить электромагнитное поле.

Схема восстановления изображена на рис. 2

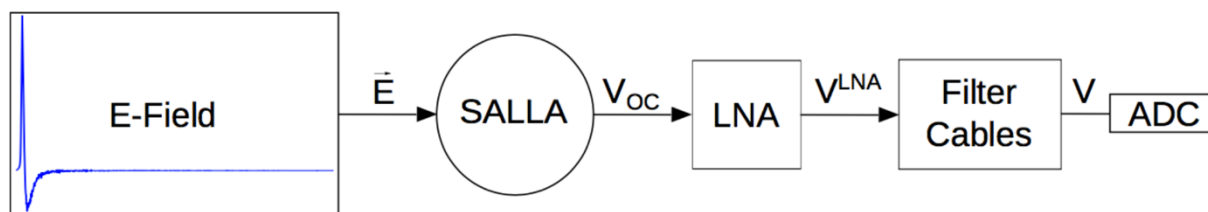


Рис.2. Схема восстановления сигнала

1.5 Восстановление параметров радиосигнала ШАЛ

Для анализа радиосигналов используется программное обеспечение Offline разработанное группой Pier Aguer.

Установки Tunka-Rex и Tunka-133 имеет общую систему сбора данных, и распознавание событий происходит следующим образом: Tunka-133 формирует триггер, событие регистрируется, если в течении определенного времени сработало более фиксируется более 3-ех сигналов ФЭУ, превышающих порог. Время каждого сигнала пересчитывается с учетом задержки, возникающей из-за прохождения по оптическому волокну. Далее объединяются отдельные события.

Для определения окна, в котором осуществляется поиск сигнала, считается, что сигнал находится примерно посередине записанной дорожки, при этом учитывается несколько дополнительных факторов.

Как минимум – возможны задержки из-за разной длины кабелей (сигнал с ФЭУ медленнее чем сигнал с радиоантенны), задержки, связанные с особенностями геометрии детектора, задержки, связанные с разницей во времени между приходом ливня и реакцией ФЭУ.

Для определения амплитуды сигнала считается максимум от аналитической огибающей (огибающей Гильберта). Для установление уровня шума, считается среднеквадратичное отклонение в той части дорожки, где

появление ложных импульсов наименее вероятно. Соотношение сигнал/шум определяется по отношению квадрата амплитуды к квадрату шума.

2. Нейронные сети и автоэнкодеры

2.1 Искусственные нейронные сети

Искусственная нейронная сеть (ИНС) - математическая модель, использующая технологии машинного обучения. Основой для ИНС является попытка повторить взаимодействие между биологическими нейронами в нервной системе живого существа. ИНС используются для решения большого класса задач – прогнозирование, распознавание образов, классификация.

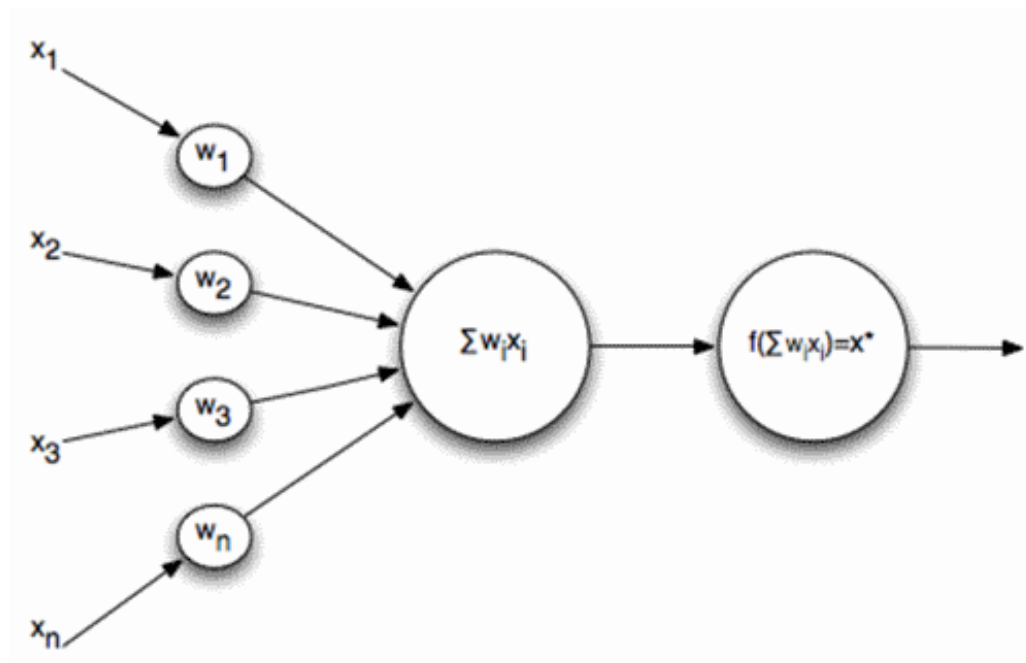


Рис.3. Искусственный нейрон

Стандартно сеть состоит из входного слоя, активационного слоя нейронов и выходного слоя.

Сигналы, поступающие со входов, суммируются с входными весами, далее к этой сумме применяется преобразование (или функция активации), полученный результат называется ответом сети. Если функция активации

линейная, то нейронная сеть вырождается в линейную регрессию. Но чаще всего используют нелинейные преобразования - сигмоида, гиперболический тангенс, пороговая функция.

Традиционно для нейронных сетей применяется обучение с учителем. Существует размеченное ранее множество примеров, на котором известны правильные ответы. Входные веса сети инициализируются случайным образом и далее меняются в зависимости от ответов сети на обучающей выборке. Данная задача решается как минимизацию функции ошибки. В качестве функции ошибки часто используют среднеквадратичное отклонение предсказанных ответов от правильных.

Для минимизации ошибки используют численные методы оптимизации: градиентный спуск (стандартный или случайный), его модификации, генетические алгоритмы, алгоритм имитации отжига, муравьиный алгоритм.

Стоит отметить, что у одного из самых популярных методов оптимизации – градиентного спуска, есть недостатки. В частности, градиент может сходиться к локальному минимуму, а не глобальному. Проблемой является так же и подбор шага спуска: слишком большие шаги могут «промахнуться» мимо минимума, а малый шаг существенно увеличивает время схождения к оптимуму. Для обхода этих минусов и улучшения результатов зачастую используются различные модификации – адаптивный градиент, ускоренный градиент Нестерова, оценщик адаптивного момента.

Нейронная сеть, состоящая из нескольких связанных между собой слоев, называется глубокой сетью.

Подобная сеть осуществляет последовательные нелинейные преобразования, применяемые к входным данным. Позволяя таким образом найти более абстрактные взаимодействия между отдельными признаками. Примером подобной сети является многослойный персептрон (Рис. 2).

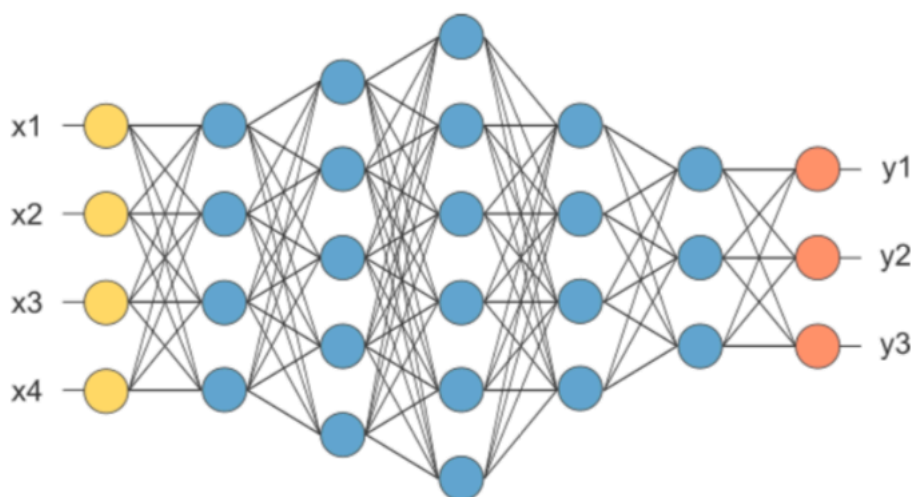


Рис.4. Сеть типа «многослойный персептрон»

Основная сложность при обучении глубоких сетей – временные и вычислительные затраты на многомерную оптимизацию. Для каждого слоя нейронов приходится считать свой градиент, что существенно с точки зрения процессорного времени и ресурсов.

Подобные вычисления требуют высокой скорости и параллелизма, по этой причине застой в активном использовании и развитии глубоких сетей наблюдался до недавнего времени.

Данный способ обучения получил название «метод обратного распространения ошибки». Условие правильной работы данного способа – дифференцируемость передаточной функции.

2.2 Сверточные нейронные сети

Важный вид глубоких нейронных сетей - сверточные нейронные сети (convolutional neural networks), которые хорошо себя зарекомендовали в рамках задач распознавания и классификации изображений. Данные сети имеют особый входной слой – вместо обычного входа, принимающего параметры, в них используются сверточные слои. Сверточный слой

представляет собой матрицу, с помощью которой реализуется операция математической операции свертки, применяющейся к входным данным.

Типичная архитектура сети - несколько сверточных слоев, чередующихся со слоями субдискретизации (pooling), так же используются слои со случайным отключением связей (dropout) и полносвязные слои. В качестве функции активации чаще всего используется ReLu, которая следует за каждым сверточным слоем. На выходе слоя выделяется так называемая карта признаков, над которой мы можем проводить дальнейшие операции свертки и субдискретизации (Рис. 3). После нескольких слоев получается конечная карта признаков, которая подается на вход полносвязной сети. Для обучения нейронной сети необходимо подобрать веса для каждого из фильтров сверточного слоя. А для итоговой классификации подобрать веса последних полносвязных слоев. Метод обучения - обратное распространение ошибки.

Подобная архитектура сети была предложена Яном Лекуном [8], который обратил внимание на некоторые особенности человеческого зрения. В частности – переход от анализа более простых деталей объекта к более общим. Сверточная сеть работает подобным образом с каждым слоем, выделяя все более абстрактные признаки входящего объекта. При этом стоит отметить, что выделяемые таким образом признаки, трудно поддаются интерпретации человеком и фактически не улучшаются напрямую. По этой причине для улучшения работы сети рекомендуется обратить внимание на настройку гиперпараметров (параметры, настраиваемые вручную) – количество слоев, уровень регуляризации.

Важным элементов сверточной сети является слой пулинга, это операция уплотнения карты признаков, появляющейся после слоя свертки. Для осуществления данной операции выбирается размер окна пулинга и признак, по которому будет вестись уплотнение. Наиболее частым подходом является выбор максимального элемента в рамках окна (MaxPooling).

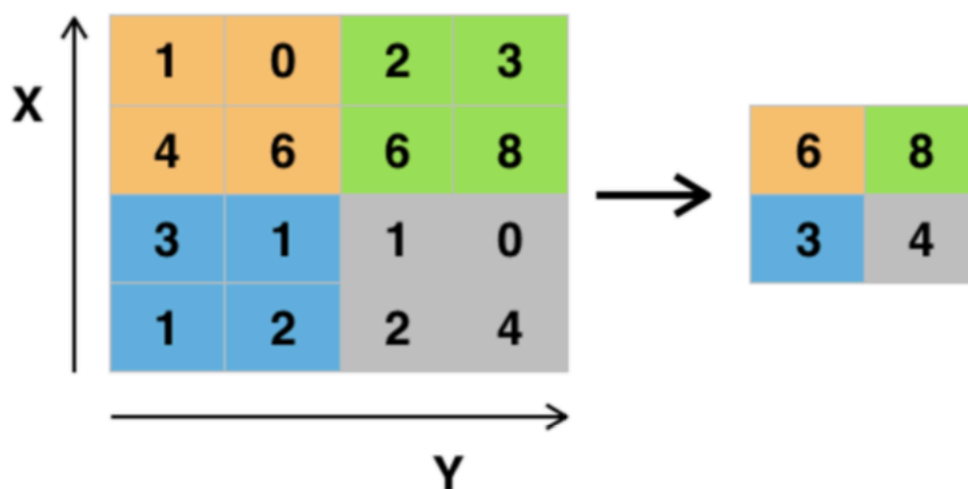


Рис.3. Операция субдискретизации

Подобный подход к построению сети имеет несколько важных достоинств: вычисления легко поддаются распараллеливанию, появляется возможность считать веса нейронной сети на GPU, сеть устойчива к поворотам и сдвигам входных данных. Легко обучается классическим методом обратного распространения ошибки.

К недостаткам сети можно отнести слишком большое количество настраиваемых гиперпараметров: количество слоев, количество фильтров на каждом слое, функции активации, параметры полносвязной сети. Все параметры подбираются эмпирически исследователем, при этом каждый может существенно влиять на итоговый результат [9].

2.3 Сети типа «автокодировщик»

Автокодировщики это еще один важный класс архитектур нейронных сетей, позволяющий решать задачи «обучения без учителя» или unsupervised learning (Рис. 4).

Архитектура подразумевает, что сеть состоит из двух частей: энкодер и декодер. Количество выходных нейронов данной сети должно быть тождественно количеству входов.

Основная задача данного типа сети: получить на выходе отклик максимально близкий ко входу. С целью избежать тривиальное решение используется следующее граничное условие: количество нейронов на внутреннем слое должно быть меньшим, чем количество входных нейронов. Таким образом, сеть учится искать корреляции между входными признаками и сжимать входные данные, это называется «внутреннее представление».

На вход сети так же можно подавать данные со сверточных слоев, подобная архитектура называется «сверточный автокодировщик».

Обычно кодировщики используют для следующих целей: первые слои для обучения нейронных сетей без учителя, аналог метода главных компонент для понижения размерности данных, очистка данных от шума.

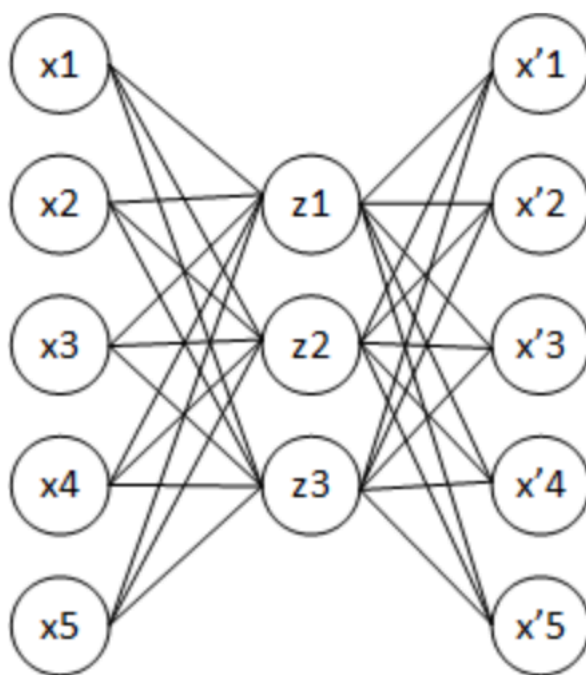


Рис.4. Архитектура типа «автокодировщик»

2.4 Проблемы переобучения и регуляризация

При работе с любыми методами машинного обучения, мы можем столкнуться с проблемами недообучения или переобучения. Каждая из этих проблем проявляется как плохой результат при реальной работе (при этом показывая хорошие результаты при обучении). Недообученной является сеть, которая тренировалась на слишком маленьком тренировочном множестве. Данная проблема решается относительно просто – тренировочное множество необходимо расширить за счет новых примеров, сеть необходимо переобучить на данном расширенном множестве.

Вторая проблема - переобучение. Сеть слишком сильно подстраивается под обучающую выборку, показывая на ней идеальное качество, однако теряет свою обобщающую способность на новых данных, показывая существенно более худшее качество. Для борьбы с этим эффектом применяется регуляризация - ограничения накладываемые на веса, которые подбирает сеть в процессе обучения. Есть несколько способов регуляризации - основные, так называемые, L1 и L2 регуляризаторы. В первом случае к функционалу ошибки добавляется модуль вектора весов, во втором - квадрат этого вектора.

Полносвязная нейронная сеть, которая находится на выходе сверточной подвержена переобучению. Для борьбы с этим явлением используется так называемый dropout. Это случайное отключение связей между частью нейронов. Количество нейронов, которые будут подвержены данной процедуре, выбирается исследователем, исходя из полученных результатов, обычно доля отключаемых нейронов устанавливается в размере 0.3-0.4.

Особенно данная технология актуальна для глубоких сетей, т.к отключение чаще всего используются на скрытых (внутренних слоях), использование dropout на входном слое - редкая практика, и если и

применяется то для долей намного меньших чем 0.5, во избежание потерь информации.

Отличие полносвязной сети, от сети с dropout можно увидеть на рис. 5

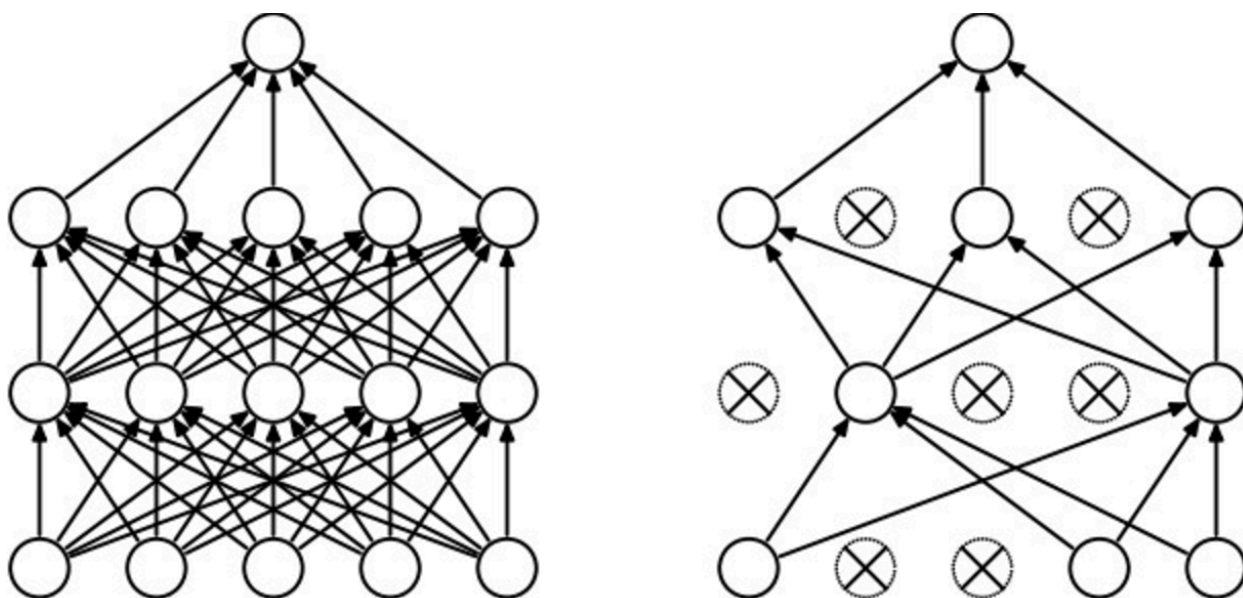


Рис.5. Пример работы Dropout

2.5 Используемые метрики качества

Для обучения сети необходимо определиться с двумя метриками, первая – метрика функции ошибки, которая будет оптимизироваться при градиентном спуске, вторая – контрольная метрика, по которой мы будем оценить успешность работы сети, понимать насколько хорошо она решает свою задачу. В качестве контрольной метрики для регрессии используют среднеквадратичное или абсолютно отклонение. Для задач классификации есть ряд метрик, связанный с точностью отнесения к классу: бинарная кросс энтропия, категориальная кросс энтропия, широко используется «ассигасу». Это доля объектов, которых мы верно отнесли к своему классу. Данная метрика проста, однако имеет ряд существенных недостатков, к примеру,

если в данных существенно преобладает один класс, а наш константный классификатор предсказывает только его, то «ассигасу» будет высокий, несмотря на то, что он практически бесполезный.

Для более полного учета результата обычно используют метрики – точность и полнота. Данные метрики строятся на основе так называемой матрицы ошибок, которая состоит из нескольких ячеек. В первой строке: первая ячейка – примеры, которые сеть правильно отнесла первому классу (true positive), вторая ячейка, те примеры, которая сеть ошибочно отвергла (false negative). Вторая строка: первая ячейка, это примеры, которая сеть ошибочно отнесла к первому классу (false positive), вторая ячейка, те примеры, что сеть правильно отнесла к негативному классу (true negative)

		Actual Class	
		p	n
Predicted Class	Y	True Positives	False Positives
	N	False Negatives	True Negatives
Totals:		P	N

Рис.5. Матрица ошибок

Точность – это доля правильных ответов, среди всех ответов, которые сеть отнесла к первому классу, фактически эта метрика показывает насколько можно доверять предсказанию классификатора. Полнота – это отношение правильно предсказанных ответов к ответам, которые действительно принадлежат классу первому, эта метрика показывает, на какой доле истинных объектов первого класса алгоритм срабатывает.

Для тестирования сети применяется отложенная выборка, примеры из которой не участвовали в обучении, следовательно являются новыми для

нашей сети. Только такие данные подходят для измерения контрольной метрики и внесения изменений, основанных на полученных результатах.

Однако не стоит рассчитывать, что качество на тестовой выборке сохранится и при внедрении сети непосредственно в процесс обработки данных, т.к. реальные данные отличается от ситуаций, которые мы встречаем при обучении модели. По этому после внедрения, необходимо отслеживать результаты сети и вносить необходимые корректировки.

3. Практическая часть

3.1 Создание необходимого окружения

В качестве инструмента для восстановления параметров сигнала был выбран автокодировщик, получающий признаки из дорожек с помощью сверточных слоев.

Для успешного выполнения данной цели необходимо было запрограммировать соответствующее программное обеспечение, выполняющее несколько основных задач, с которыми нам приходится сталкиваться в процессе обучения сетей.

В частности были разработаны:

- converter.py – конвертер из формата ADST в бинарный формат NumPy
- creator.py – инструмент для создания обучающих выборок
- trainer.py – инструмент для обучения нейронных сетей и получения натренированной модели
- denoiser.py – инструмент для очистки входных дорожек от шума
- estimator.py – инструмент для первичной оценки результатов
- plotter.py – инструмент для построения различных гистограмм

Кроме того нейронная сеть была интегрирована в процесс обработки данных, которая производится с помощью программы Offline.

В качестве основного программного обеспечения для построения нейронных сетей были выбран Keras, который работает на основе Tensorflow от Google.

Первоначально процесс обучения проходил на сервере с несколькими CPU, однако позже процедура обучения была перестроена под GPU. Для этого использовалась виртуализация, в частности программное обеспечение

uDocker, которое позволяет создавать задачи в виртуальных контейнерах и отправлять их на выполнение на удаленном сервере. Существует несколько готовых контейнеров, которые содержат различные версии необходимых нам программ.

Такой подход позволил ускорить время обучения одной сети до 20-и раз.

3.2 Конвертер данных из ROOT

ROOT – программное обеспечение, разработанное в CERN, предназначенное для обработки экспериментальных данных в физике высоких энергий и содержит несколько различных библиотек, которые могут быть использованы в том числе и для астрофизических задач. Кроме того, данное программное обеспечение предлагает несколько структур для хранения данных – массивы, деревья.

В нашем случае ROOT контейнер с данными представляет собой массив и содержит в себе :

- объект event — зафиксированные (в нашем случае сгенерированные) события
- объекты station для каждого события — 18 станций
- объект track для каждой станции — дорожка из 4096 отчетов

У нас есть два контейнера, первый содержит события с различным уровнем SNR (отношение сигнал/шум), второй контейнер содержит аналогичные события с наложенными поверх шумами, сгенерированными программно.

Программа основана на функции `HasPulse()`, входящей в пакет ROOT. Данная функция возвращает `True`, если отношение SNR в дорожке превышает необходимый порог. На ее основе мы строим цикл по всем событиям, станциям и дорожкам и оставляем только те пары дорожек, которые реально использовать в дальнейшем. Данные дорожки мы конвертируем в Numpy массив, на основе которого можно обучать нейронные сети.

Код конвертера представлен в Приложении 1.

3.3 Процесс обучения

Для процесса обучения, с помощью ПО Corsika, было сгенерировано более 20 тысяч модельных пар из симуляций сигнала (дорожек) и наложенного на него шума.

Пример пары дорожек на рис. 5.

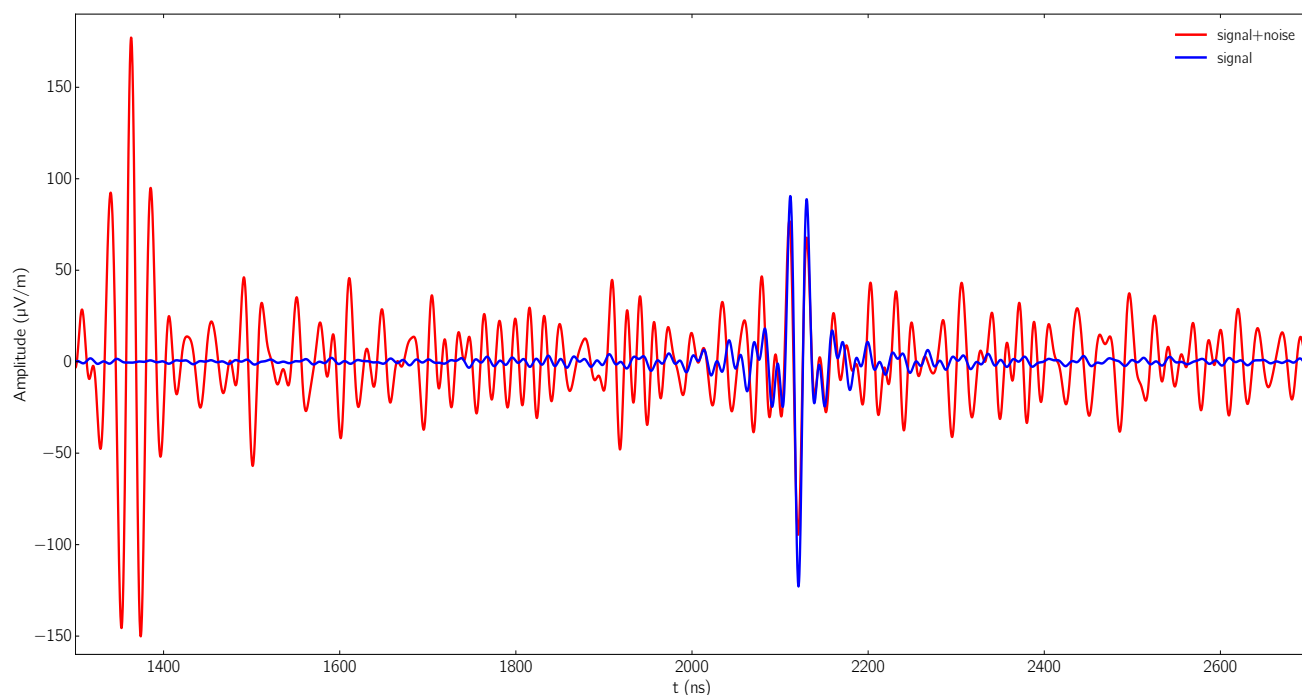


Рис.5. Пример пары дорожек для обучения

При условии, что upsampling равен 16, первоначальная дорожка состоит из 16 384 отсчетов.

После обработки, дорожка для обучения собой последовательность из 4096 отсчетов, каждый отчет соответствует 0.325 наносекунде. Сигнал расположен в центре дорожки с возможным отклонением ± 100 наносекунд.

Все дорожки были поделены по уровню амплитуды: 0-100 мкВ/м, 100-200 мкВ/м, 200-300 мкВ/м. Для каждого диапазона амплитуд было использованы собственные примеры для обучения.

Структурно сеть состоит нескольких кодирующих и несколько декодирующих слоев (их число должно совпадать).

Было создано более 20 различных архитектур сетей, они отличались количеством слоев, способами субдискретизации и т.д

Пример архитектуры нашей нейронной сети представлена на рис. 6.

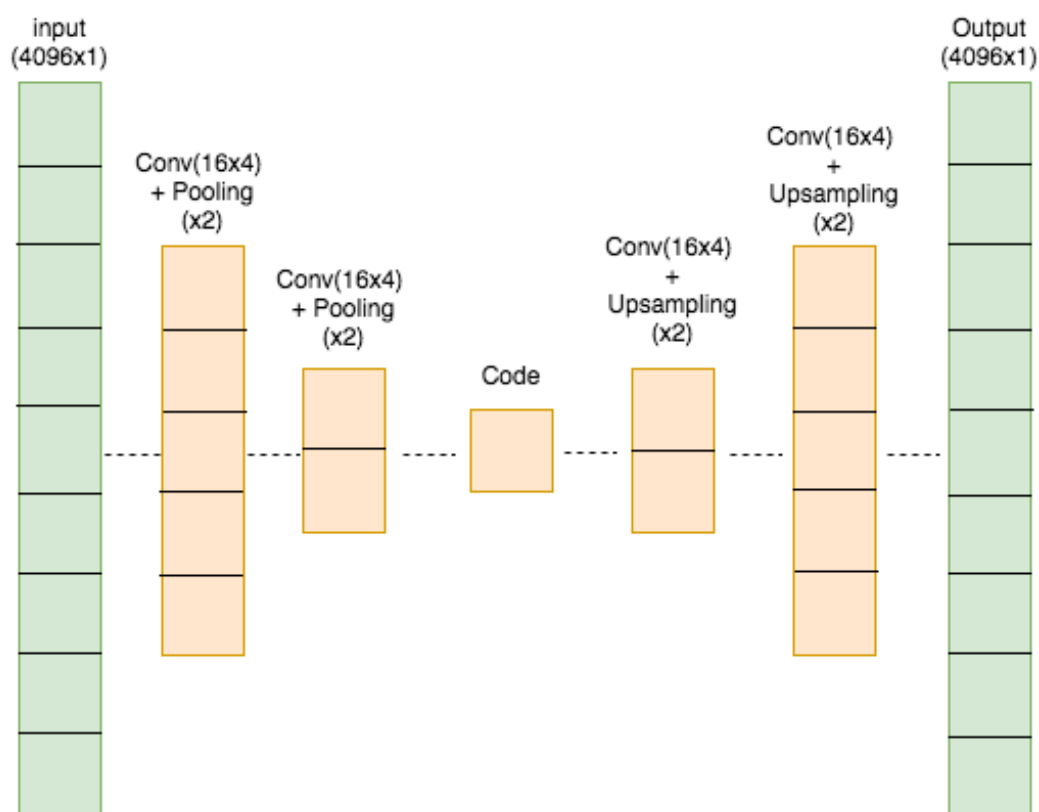


Рис.5. Архитектура нейронной сети

Для получения обученной модели используется программа на вход которой подается пара массивов, содержащих дорожки с шумом и без. Так же параметром для программы, является диапазон амплитуд, которые будут использоваться для обучения. Процесс обучения одной сети занимает порядка получаса, в зависимости от количества используемых слоев и фильтров, а так же количества примеров, попавших в окно амплитуд.

Сохраненную модель можно использовать в программе для очистки подготовленных дорожек с шумом, входные параметры «denoiser.py»: массив с зашумленными дорожками, модель для очистки данных.

Код программы «denoiser.py» представлен в Приложении 2.

Для оценки работы нейросети создается таблица, которая содержит в себе данные по истинному и восстановленному положению пика сигнала, истинной и восстановленной амплитуде. На основе этой таблицы можно построить необходимые гистограммы.

Для использования нейросети в процессе восстановления, используется interface.py, который позволяет автоматически поучать данные для очистки из программы Offline и передавать их обратно.

3.4 Результаты

Первоначальная оценка сети производится на основе оптимизируемой метрики, а так же на визуальном анализе полученных результатов и гистограмм.

Если нейронная сеть удовлетворяет необходимым требованиям, ее тестируют в полноценном процессе восстановления с помощью Offline.

Так же в процессе анализа результатов очистки на малых амплитудах, были найдены интересные случаи, где нейронная сеть обрабатывает лучше стандартного метода. Так же нейронная адекватно работает на тех случаях, где сигнал фактически отсутствует из-за сильного воздействия шума.

На рис. 5 представлена работа программы при поступлении на вход сигнала с большой амплитудой.

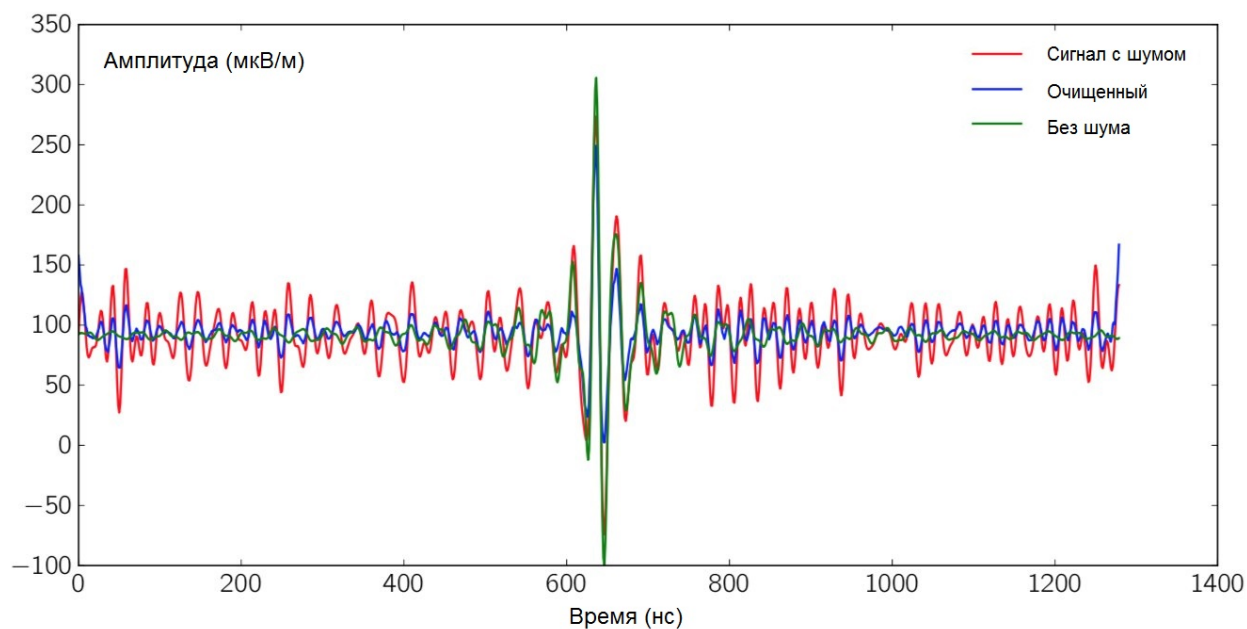


Рис.5. Результат работы автокодировщика

Работа программы на фактически «шуме» изображена на рис. 6.

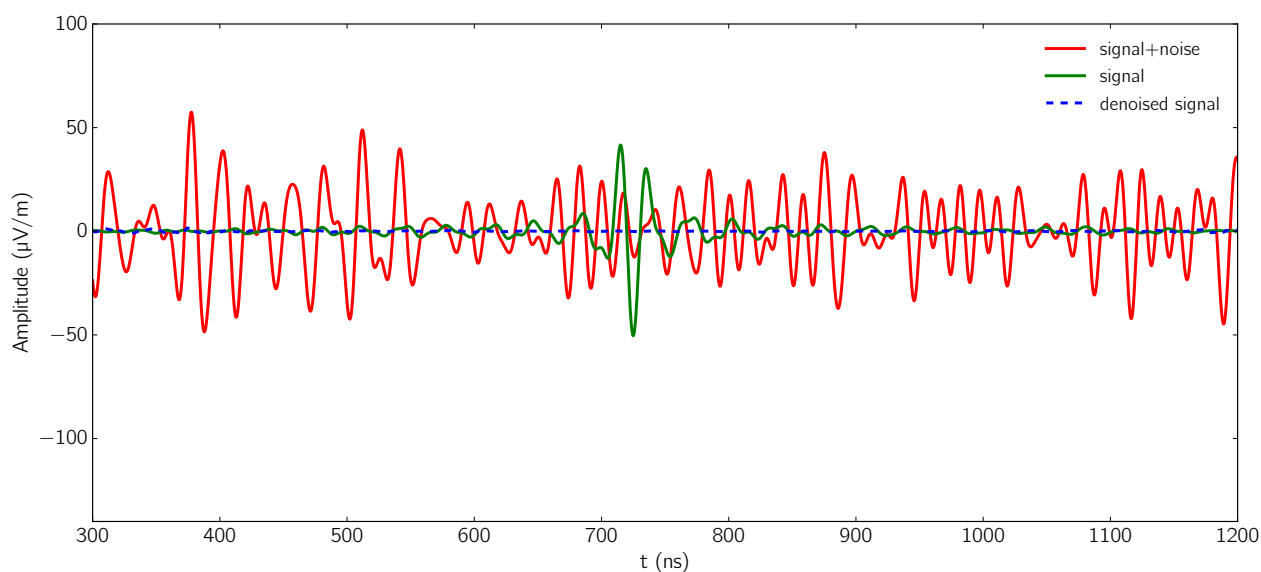


Рис.5. Результат работы автокодировщика на «шуме»

Хорошо восстанавливается положение пика сигнала для низких амплитуд (10-100 мкВ/м), диаграмма рассеяния представлена на рис. 6.

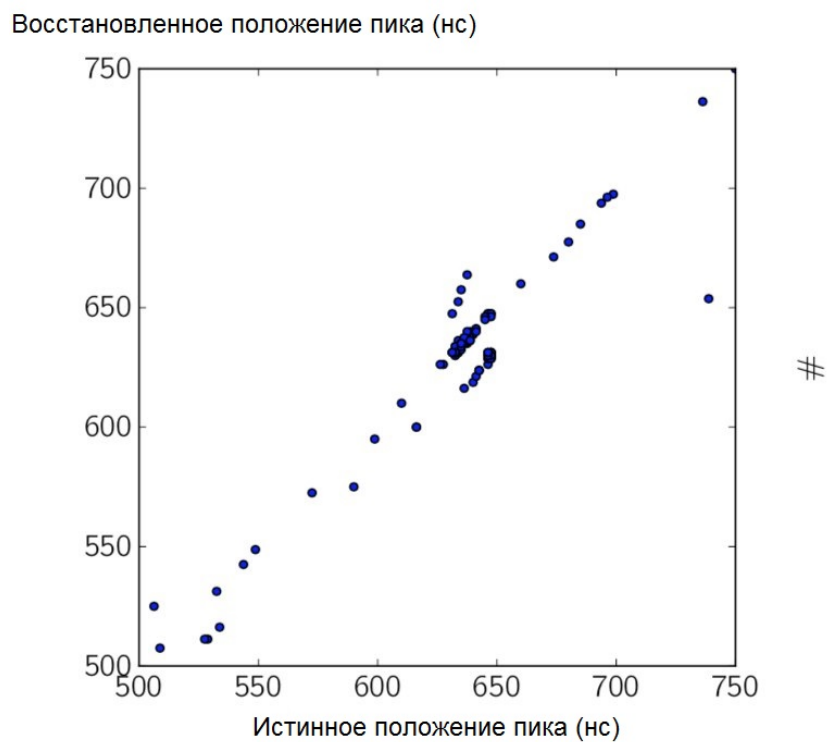


Рис. 6. Диаграмма рассеяния

Количество восстановленных дорожек зависит от амплитуды (Рис.7).

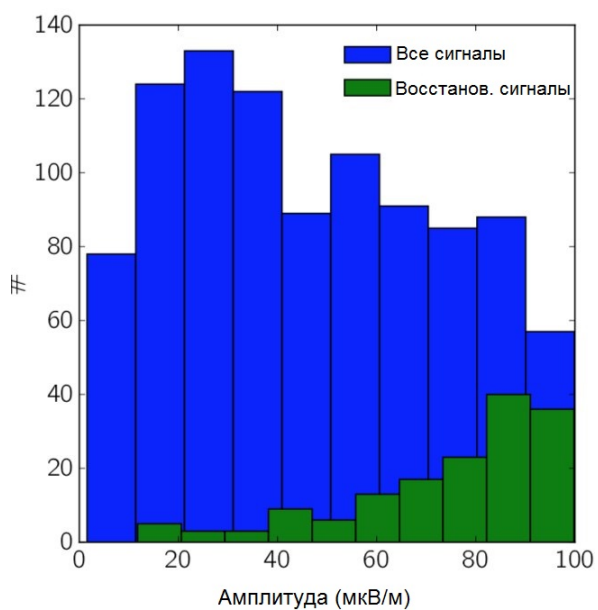


Рис.7. Распределение восстановленных сигналов

На реальных данных, использование нейросети позволило восстановить больше событий чем классический метод (85 против 54), а так же не уступило согласованной фильтрации.

ЗАКЛЮЧЕНИЕ

В ходе работы поставленная цель была выполнена, все поставленные задачи были решены.

Изучены механизмы генерации ШАЛ и методы детектирования их радиоизлучения, изучены способы текущей обработки сигналов и восстановления событий на установке Tunka-Rex.

Помимо изучения вышеописанной информации был написан комплекс программного обеспечения для создания и обучения нейронных сетей на основе как генерированных, так и реальных данных эксперимента Tunka-Rex. Разработан интерфейс для взаимодействия с программой Offline, появилась возможность использовать нейросети в существующей процедуре восстановления событий.

Протестировано более 20 различных архитектур автокодировщиков и получены результаты, сравнимые с классическими методами восстановления, а так же с согласованной фильтрацией. Это доказывает перспективность для дальнейшего использования нейронных сетей в качестве инструмента в цепочке обработки событий на Tunka-Rex.

В дальнейшем, планируется продолжение работы в данном направлении, в частности доработка различных отобранных архитектур автокодировщиков. Кроме того планируется тестирование принципиально других архитектур: LSTM сети, капсульные сети, генеративные сети для решения задач восстановления параметров радиосигнала ШАЛ.

Список использованных источников

1. Б.С. Ишханов, И.М. Капитонов, Н.П. Юдин. Частицы и атомные ядра. М.: Изд. Московского университета, 2005.
2. И.М. Капитонов. Введение в физику ядра и частиц. М.: МГУ, 2000.
3. Черенковское излучение / Cherenkov radiation [Электронный ресурс] - URL: <http://nuclphys.sinp.msu.ru/enc/e186.htm/> (дата обращения: 30.09.17).
4. Кочаров Г. Е. Космические лучи ультравысокой энергии и реликтовое излучение во вселенной / Г. Е. Кочаров // Соросовский образовательный журнал. – 2001. – Том 7, No7. – С. 83–87
5. TUNKA EAS CHERENKOV LIGHT ARRAY [Электронный ресурс] – URL: <http://dec1.sinp.msu.ru/~tunka/> (дата обращения: 15.09.17).
6. Kostunin D. The Tunka Radio Extension: Latest Analysis Results / D. Kostunin, P. A. Bezyazeekov, N. M. Budnev // J. Phys. Conf. Ser. – 2015. – V.632, n. 1. – P. 1–7
7. Казарина Ю.А. Исследование структуры радиоизлучения каскадного ливня от космических лучей высоких энергий в Тункинском эксперименте.: Дис. ... канд. физ-мат. Наук: Иркутск – 2015/ Ю.А.Казарина; Ирк.гос.универ. – Иркутск, 2015. – 104 с.
8. Pierre Sermanet. Convolutional Neural Networks Applied to House Numbers Digit Classification / Pierre Sermanet, Soumith Chintala and Yann LeCun. // International Conference on Pattern Recognition – 2012.
9. Honglak Lee. Unsupervised feature learning for audio classification using convolutional deep belief networks / Honglak Lee, Yan Largman, Peter Pham and Andrew Y. Ng. // NIPS – 2009.

Листинг 1

```

import ROOT
import numpy as np
from scipy import signal
from pyik.adst import RecEventProvider,
RecEventHandler, MICRO, FULL, ROOT
import matplotlib.pyplot as plt
%matplotlib inline
# Читаем ROOT файлы и создаем объекты ROOT Event Vector
file_name_nonoise = "/media/windata/dataset/ADST/ADST-
nonoise-nocorr.root"
file_name_noise = "/media/windata/dataset/ADST/ADST-
noise-nocorr.root"
adstHandler_nonoise =
RecEventHandler(file_name_nonoise, mode = MICRO)
adstEvent_nonoise = ROOT.RecEvent()
adstHandler_nonoise.SetBuffers(adstEvent_nonoise)
nEvents = adstHandler_nonoise.GetNEvents()
adstHandler_noise = RecEventHandler(file_name_noise,
mode = MICRO)
adstEvent_noise = ROOT.RecEvent()
adstHandler_noise.SetBuffers(adstEvent_noise)
# Создаем пустые массивы для наших данных
noise_list = []
nonoise_list = []
for event in xrange(nEvents):
    adstHandler_nonoise.ReadEvent(event)
    adstHandler_noise.ReadEvent(event)
    for station_nonoise, station_noise in
zip(adstEvent_nonoise.GetRdEvent().GetRdStationVector()
,\ adstEvent_noise.GetRdEvent().GetRdStationVector()):
        if not station_nonoise.HasPulse(): continue
        trace_nonoise =
station_nonoise.GetRdTrace(0)
        trace_noise = station_noise.GetRdTrace(0)
        samples_nonoise =
trace_nonoise.GetTimeTrace()
        samples_noise = trace_noise.GetTimeTrace()
        signal_nonoise = [x for x in
samples_nonoise]
        signal_noise = [x for x in samples_noise]
        nonoise_list.append(signal_nonoise)
        noise_list.append(signal_noise)

```



```
# Трансформируем их в объекты типа numpy array  
nonoise_list = np.array(nonoise_list)  
noise_list = np.array(noise_list)
```

Результатом работы программы является numpy массивы, содержащие все необходимые нам данные для дальнейшей обработки. Данные массивы можно записать в файл с расширением `npy` для дальнейшего сохранения на жестком диске.

Листинг 1

```

# Подключаем важные библиотеки
import sys
import numpy as np
# Определяем необходимые функции
def processing_data(array):
    shifted_signals = []
    for signal in array:
        xmin = np.abs(np.min(list(signal)))
        signal = np.asarray(signal) + xmin
        shifted_signals.append(signal)
    maxs = np.max(shifted_signals)
    shifted_signals = shifted_signals/maxs
    return shifted_signals, maxs

# Choose the model
model = str(sys.argv[3])
# Загружаем данные
noised_signals = np.load(sys.argv[1])
noised_signals, maxs = processing_data(noised_signals)
# Получаем результат
from keras.models import load_model
autoencoder = model
predict = np.reshape(noised_signals,
                     (noised_signals.shape[0], 1024, 1))
predict = np.asarray(predict)
predict = predict*maxs
# Сохраняем результат
np.save(sys.argv[2], predict)

```

Результатом работы программы является numpy массив, который содержит очищенные от шума дорожки.