

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ЯДЕРНЫЙ УНИВЕРСИТЕТ «МИФИ»

УДК 539.12.01

ОТЧЕТ
О НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ

**ПРИМЕНЕНИЕ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ ДЛЯ
ИДЕНТИФИКАЦИИ СТРУЙ, ОБРАЗОВАННЫХ W-БОЗОНОМ**

Научный руководитель

к.ф.-м.н.

Студент

_____ Мягков Алексей Григорьевич

_____ Ван Алина Маошэновна

Москва
2024

Содержание

Введение	3
1 Машинное обучение с учителем	4
1.1 Бинарная классификация	4
1.2 Многослойный персептрон (MLP)	5
1.2.1 Функции активации	6
1.2.2 Метод обратного распространения ошибки	7
1.2.3 Гиперпараметры модели	7
1.3 Переобучение и недообучение	7
1.4 Свёрточные нейросети	8
2 Дискриминирующие переменные	10
3 Процесс работы и результаты	11
3.1 Подготовка данных	11
3.2 Обучение и тестирование MLP	11
3.3 Обучение и тестирование CNN	14
4 Заключение	18

Введение

Стандартная модель – это современная теория в физике элементарных частиц, описывающая сильное, слабое и электромагнитное взаимодействия. Однако, несмотря на все свои преимущества, Стандартная модель не считается полной теорией всего, так как она не включает в себя гравитационное взаимодействие и не описывает некоторые экспериментальные факты, такие как скрытая масса, темная энергия, барионная асимметрия и т.д. Предполагается, что Стандартная модель является частью более общей теории. Поэтому одной из главных задач на ЛНС является поиск новой физики за рамками Стандартной модели.

Идентификация толстых струй, образованных W -бозонами, распавшимися по адронной моде, является одним из главных составляющих этапов анализа данных с экспериментов по поиску новой физики. К примеру, в эксперименте по поиску возбужденного лептона с конечным состоянием $e\nu J$ процессом, приносящем основной вклад в фон, является образование пары топ-анти топ. Так как топ-кварк распадается преимущественно по слабому взаимодействию, идентификация струй, образованных W -бозоном, является необходимой для подавления фона.

В предыдущей работе для идентификации W -бозона использовалось ограничение по переменной D_2 и инвариантной массе струи. Однако для выделения струй от векторных бозонов существуют и другие дискриминирующие переменные. Для улучшения точности идентификации толстой струи рассмотрено решение этой задачи с помощью методов машинного обучения. В контексте машинного обучения задача идентификации частицы – это задача бинарной классификации, т.е. предсказание по полученным признакам к какому из двух классов принадлежит объект: фону или сигналу.

Ранее проводилось изучение таких моделей машинного обучения, как бустированные деревья решений с разными алгоритмами бустинга: градиентным и адаптивным. В рамках изучения моделей сделаны выводы, что несмотря на хорошую интерпретируемость и быстрый темп обучения, данные модели имеют свойство быстро переобучаться.

Оценка качества предыдущих моделей AUC варьировалась от 0.815 до 0.820 при различных настройках гиперпараметров.

Цель работы.

Целью работы является использование различных моделей глубокого обучения для решения задачи идентификации толстых струй, образованных W -бозоном.

В соответствии с поставленной целью задачами данной работы были:

- Ознакомление с различными моделями глубокого обучения для решения задачи бинарной классификации;
- Выбор дискриминирующих переменных (признаков) для объектов и формирование сигнального и фонового деревьев для данной задачи
- Ознакомление с библиотекой глубокого обучения Keras;
- Обучение и тестирование моделей;
- Резюме результатов.

1 Машинное обучение с учителем

Машинное обучение – это область прикладной математики, изучающая методы решения задач с использованием обучающих данных. В данной работе используются методы машинного обучения с учителем.

Алгоритмы машинного обучения с учителем изучают соответствие между входными и выходными данными. Постановка задачи для обучения с учителем следующая: на вход алгоритма подаются размеченные данные вида (матрица признаков; отклик). Эти данные называются обучающей выборкой, они используются для настройки модели. В зависимости от задачи обучающие данные также могут быть разделены и на валидационную выборку, которая необходима для оценки обобщающих способностей обученных моделей и выбора лучшей модели. Тестовая выборка используется для финальной оценки качества модели. Цель машинного обучения с учителем состоит не в моделировании данных из пространства признаков, а в предсказании откликов на новых данных.

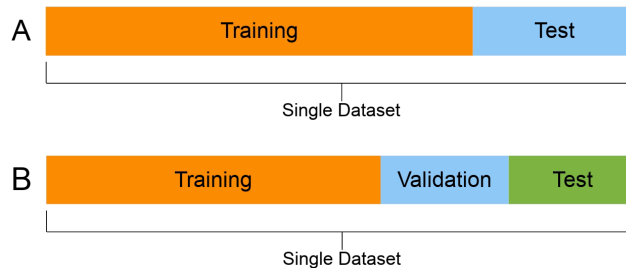


Рис. 1: Разделение датасета

Основными задачами машинного обучения с учителем являются задачи регрессии и задачи классификации. При регрессии алгоритм учится предсказывать непрерывное числовое значение. То есть цель алгоритма для задачи регрессии установить функциональную зависимость между независимыми переменными (признаками) и откликом. При классификации модель учится предсказывать класс принадлежности события (метку класса), то есть целевая переменная – это категориальное значение. Основной целью моделей для решения задачи классификации является обобщение обучения таким образом, чтобы делать точные прогнозы на тестовых данных.

Оценка качества модели связана с функцией потерь. Функция потерь – это мера неточности модели, она сравнивает цели и выходы модели. Задача алгоритма машинного обучения с учителем – минимизация функции потерь.

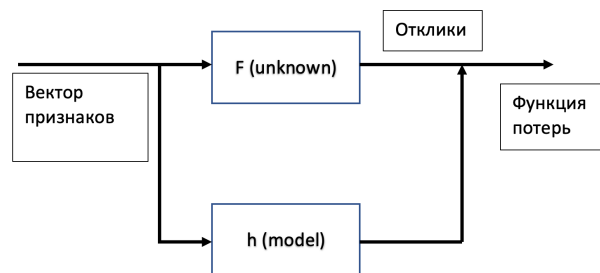


Рис. 2: Схема алгоритма обучения с учителем

1.1 Бинарная классификация

Задача идентификации частицы – это задача бинарной классификации. То есть отклики в данной модели дискретны, они могут принимать только два значения, в нашем случае – сигнал или фон. Задача бинарной классификации [1] состоит в нахождении гиперплоскости, размерность которой равна $N - 1$, где N – количество признаков. Эта гиперплоскость должна оптимально разделять объекты одного класса от другого, то есть отличать сигнал от фона.

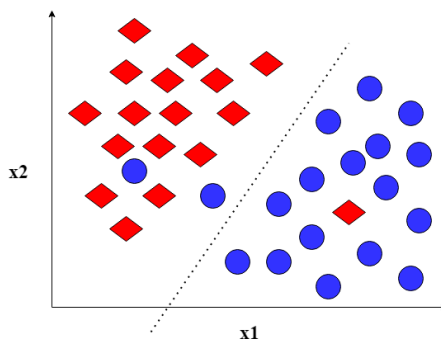


Рис. 3: Бинарная классификация

Функция потерь в данной задаче – это некоторая мера неточности классификатора в предсказании метки класса для объекта. Меньшие потери означают большую точность в предсказании метки класса. В задаче бинарной классификации функция потерь определяется как функция от разности между целями и выходами модели. То есть если объект классифицирован верно – вклад его в функцию потерь будет минимален, если классификатор ошибся, то вклад в потери будет большим. Однако среднего значения функции потерь по выборке (эмпирического риска) не достаточно для оценки ошибки модели из-за дискретного характера откликов. Поэтому необходимы дополнительные метрики для оценивания качества классификатора.

Одними из таких характеристик качества классификации являются ROC-кривая и AUC. ROC-кривая – это график зависимости доли истинных положительно определенных результатов от доли ложных положительно определенных результатов для различных значений контрольных точек проверки гипотезы. AUC – это площадь под ROC-кривой, которая является статистической характеристикой того, насколько доля истинных положительно определенных результатов превышает ложные положительно определенные результаты. То есть чем лучше классификатор разделяет два класса, тем больше AUC и тем выше ROC-кривая.

Для решения задачи бинарной классификации существует множество алгоритмов: от простых моделей, как логистическая регрессия, до глубоких нейронных сетей. В работе использованы алгоритмы MLP и CNN, речь о которых пойдет далее.

1.2 Многослойный персептрон (MLP)

Нейронная сеть – это сложная дифференцируемая функция, задающая отображение из признакового пространства в пространство откликов, все параметры которой могут настраиваться одновременно и взаимосвязано. Сложную функцию обычно представляют в виде композиции простых функций, которые называют слоями. Общая длина цепочки слоев определяет глубину модели.

Нейронную сеть, в которой есть только линейные слои и различные функции активации, называют многослойным персептроном (MLP). Многослойный персептрон состоит из входного, скрытого и выходного уровней взаимосвязанных нейронов.

Архитектура MLP следующая:

- Структура нейронов в входном слое определяется количеством признаков в данных. На входном уровне каждому нейрону сопоставляется один из признаков обучающих данных, далее эти значения распределяются по нейронам скрытых слоев.
- На скрытых слоях происходят сами вычисления. Каждый нейрон в скрытом слое анализирует информацию, полученную из предыдущего слоя. Происходит преобразования данных из пространства одной размерности в пространство другой размерности.
- В выходном слое нейроны генерируют предсказания модели. Структура выходного слоя зависит от поставленной задачи: для задачи бинарной классификации нейронов в выходном слое 2.

Структура слоев в MLP:

- Линейный слой — линейное преобразование над входящими данными (его обучаемые параметры — это матрица W и вектор b). Такой слой преобразует d -мерные векторы в k -мерные.
- Функция активации[2] — нелинейное преобразование, поэлементно применяющееся к пришедшим на вход данным. Благодаря функциям активации нейронные сети способны порождать более информативные признаковые описания, преобразуя данные нелинейным образом.

1.2.1 Функции активации

Главные требования для функции активации: быть монотонной и иметь первую производную почти всюду (необходимо для обратного распространения ошибки). В качестве функции активации могут использоваться разные функции, у каждой из которых есть свои плюсы и минусы.

Сигмоида

Функция сигмоиды преобразовывает поступающие в неё значения в вещественный диапазон $[0, 1]$. То есть, если входные данные окажутся большими положительными значениями, то после преобразования они будут равны примерно единице, а отрицательные числа станут близки к нулю. Это довольно популярная функция, которую можно интерпретировать как частоту возбуждения нейрона. Однако сигмоида имеет несколько недостатков. Во-первых, область значений данной функции смещена относительно 0. Во-вторых, на хвостах сигмоиды происходит затухание градиента, что неприятно для обратного распространения ошибки. В-третьих, просчет экспоненты вычислительно сложен.

Гиперболический тангенс

Гиперболический тангенс используется в случаях, когда необходимо ограничить выходные данные в диапазоне от -1 до 1. Он часто применяется в скрытых слоях нейронных сетей. В отличие от сигмоиды он не смещен относительно 0. Однако значения градиента при обратном распространении по-прежнему могут обнуляться. Тем не менее, использование тангенса обычно более предпочтительно.

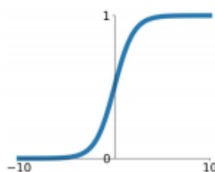
ReLU

ReLU является наиболее популярной функцией активации и широко используется в современных нейронных сетях. Она позволяет избежать проблемы затухания градиента. Она вычисляет функцию $f(x) = \max(0, x)$, то есть просто выдаёт значения «ноль» и «не ноль».

Кроме того, ReLU очень просто вычисляется: примерно в шесть раз быстрее сигмоиды и тангенса. Однако, в ней отсутствует нулевое центрирование.

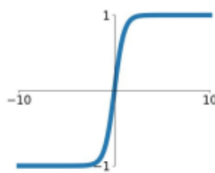
Sigmoid

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



tanh

$$\tanh(x)$$



ReLU

$$\max(0, x)$$

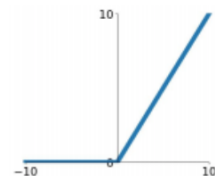


Рис. 4: Функции активации

1.2.2 Метод обратного распространения ошибки

Нейронная сеть обучается с помощью какой-либо модификации градиентного спуска. А для этого ей необходимо вычислять градиенты от функции потерь по всем обучающим параметрам. Нейронную сеть, как сложную функцию, можно представить в виде вычислительного графа, в которых узлы – это вычислительные операции или простые функции.

Применение нейронной сети к данным (вычисление выхода по заданному входу) называют прямым проходом. На этом этапе происходит преобразование исходного представления данных в выходное и последовательно строятся промежуточные представления данных — результаты применения слоёв к предыдущим представлениям. При обратном проходе информация движется от финального представления (от функции потерь) к исходному через все преобразования. Механизм обратного распространения ошибки, играющий важнейшую роль в обучении нейронных сетей, как раз предполагает обратное движение по вычислительному графу сети.

Метод обратного распространения ошибки заключается в рекурсивном использовании правила дифференцирования сложной функции для вычисления градиента в каждом узле, начиная с конечного узла. Информацию о величине градиента для каждого из узлов мы берем из forward pass.

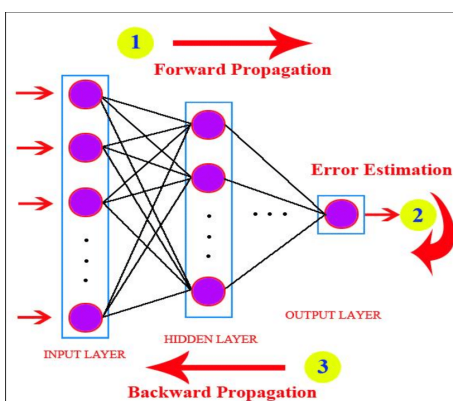


Рис. 5: Метод обратного распространения ошибки

1.2.3 Гиперпараметры модели

Выше была упомянута такая настройка MLP, как выбор функции активации для слоев сети. Однако не только выбор функции активации является гиперпараметром модели. Реализация градиентного спуска на больших наборах данных является затратной операцией как для оперативной памяти, так и по времени, поэтому существуют различные алгоритмы оптимизации весов: стохастический градиентный спуск, Adam и т.д. Выбор алгоритма оптимизации также является настройкой модели, вместе с выбором лосс-функции, которую надо минимизировать. По умолчанию лосс-функция для задачи бинарной классификации - это бинарная кросс-энтропия. Ширина шага, с которой градиент спускается в поисках минимума, называется скоростью обучения. Обычно скорость обучения варьируется от 0.001 до 0.1.

Пропустить через всю сеть целую обучающую выборку будет не только затратным, но и неоптимальным решением. Поэтому данные делят на маленькие части - батчи. Размер батча также является гиперпараметром: чем меньше батч, тем больше количество итераций. Для обновления весов модели необходимо более чем раз провести данные через сеть в прямом и обратном направлении. Эпоха - это один цикл, когда вся обучающая выборка проходит в прямом и обратном направлении через нейронную сеть. Обычно оптимальное количество эпох для модели - это плато между состояниями переобучения и недообучения.

1.3 Переобучение и недообучение

При самом процессе обучения модель может быть недостаточно обучена или переобучена. Недообучение возникает в том случае, когда модель не способна точно предсказывать отклики и моделировать закономерности в данных. Когда модель настроена на избыточную классификацию, тогда можно сказать, что модель переобучена.

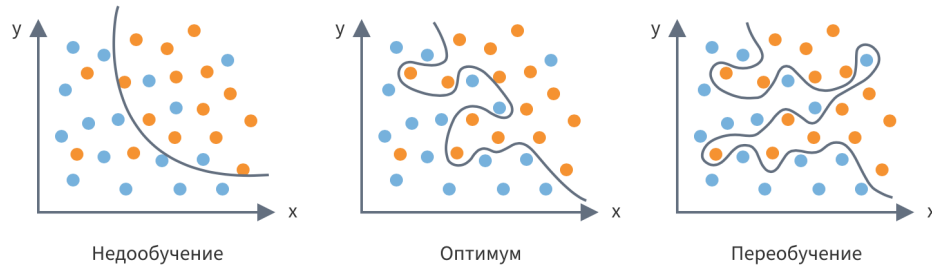


Рис. 6: Примеры недообучения и переобучения

Зафиксировать состояние недообучения или переобучения можно с помощью графиков зависимости ошибок на обучающих данных (далее $loss$), ошибок на валидационной выборке (val_loss) и номеру эпохи. Для устранения от недообучения применяют следующие подходы: увеличивают количество эпох, меняют архитектуру модели, увеличивают обучающую выборку, усложняют модель. Для устранения переобучения также существуют решения:

- Останавливают процесс обучения на эпохе, при которой val_loss еще не начинает возрастать;
- Уменьшают количество параметров модели, т.е. делают модель менее сложной;
- Применяют dropout: отключают указанную часть нейронов в слое, таким образом прореживая слой;
- Регуляризация: добавляют в функцию потерь дополнительное слагаемое, которое ставит дополнительные ограничения на вектор весов.

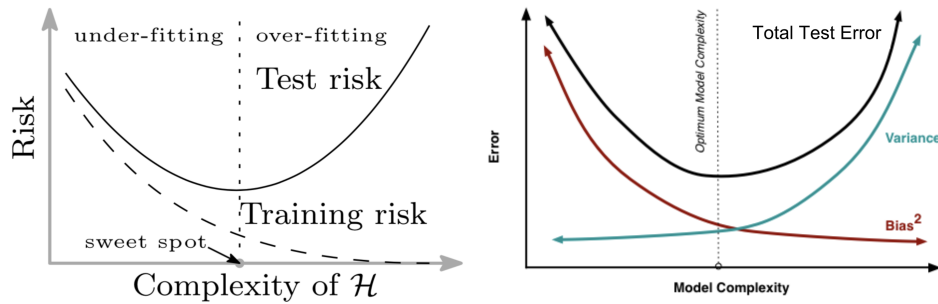


Рис. 7: История обучения

1.4 Свёрточные нейросети

Свёрточные нейронные сети [3] - это класс моделей глубокого обучения, ориентированный на классификацию изображений. Основными компонентами данной архитектуры являются свёрточные, пулинговые и полносвязные слои. Основное отличие полносвязного слоя от свёрточного заключается в следующем: слои MLP изучают глобальные шаблоны в пространстве входных признаков, тогда как свёрточные слои изучают локальные паттерны. Вместе со свёрточными и пулинговыми слоями добавляются и разные гиперпараметры, речь о которых пойдет далее.

Идея свёрточного слоя - нахождение паттернов в локальном подпространстве признаков. Причем эти паттерны инвариантны относительно переноса. То есть если в верхнем углу картинке сеть опознала некоторый шаблон, далее она может распознавать его повсюду. Для нахождения некоторых шаблонов CNN использует фильтры: фильтр проходит по входному тензору, записывая результат своего прохождения в карту ответов. К примеру, для задачи классификации цифр, нарисованных на бумаге, фильтр, настроенный на нахождение паттерна скругления, проходя по картинке с изображением "8" записывает в

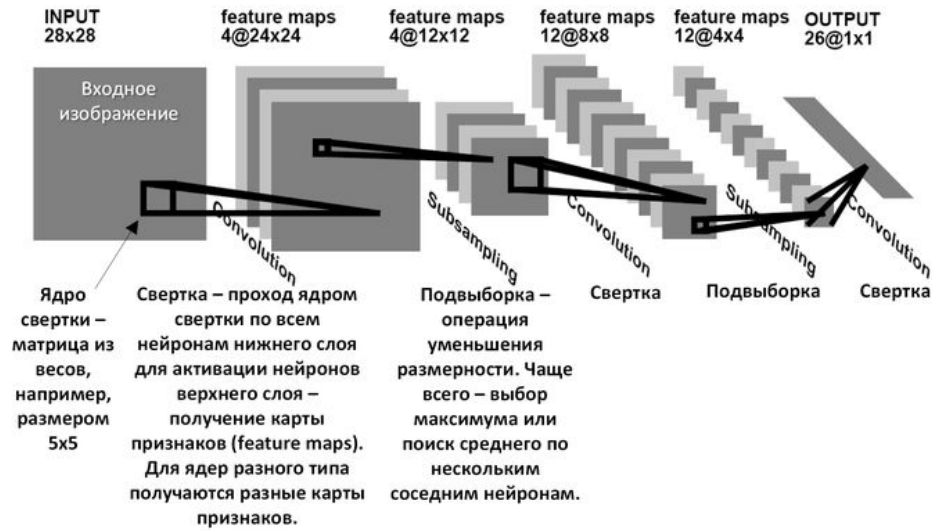


Рис. 8: Схема CNN

карту ответов 1, где есть места скругления, или 0, где таких мест нет. Полученные карты ответов, к примеру двумерные матрицы, для каждого фильтра создают вместе трехмерный тензор, глубина которого равна количеству фильтров. Фильтр - это параметр сети, который она настраивает в процессе обучения. Однако количество фильтров и их размерность - это гиперпараметры, которые уже нужно настраивать.

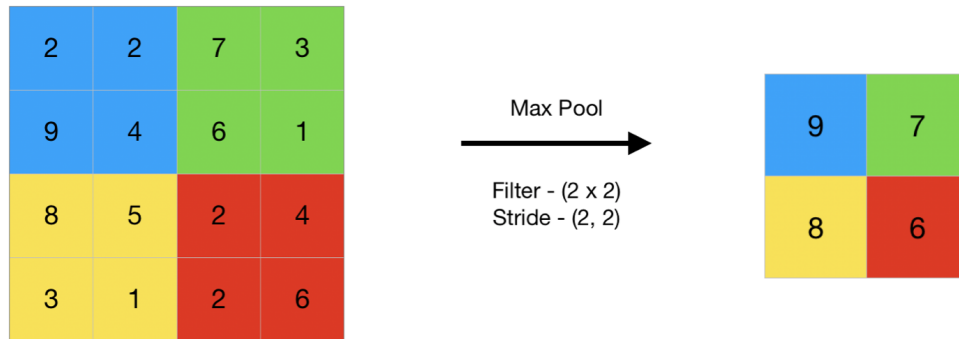


Рис. 9: Max Pooling

Далее идет пулинговый слой. Его идея заключается в уменьшении пространства полученной на сверточном слое карте признаков. Есть разные стратегии уменьшения пространства: выбор максимального значения в подматрице в карте ответов, выбор среднего и т.д. В этом слое настраиваемый гиперпараметр - размер окна/подматрицы, из которого в выходной тензор записывается максимальное значение.

Архитектура полносвязных слоев аналогична MLP.

2 Дискриминирующие переменные

Первые два выбранных признака [4], которые будут поданы на вход алгоритмов МО – это инвариантная масса струи и поперечный импульс струи. Ранее для выделения W-струй из большого фона КХД использовалась переменная D2. Определение переменной:

$$D_2 = E_{CF3} \times \left(\frac{E_{CF1}}{E_{CF2}} \right)^3, \quad (1)$$

где энергетические корреляционные функции задаются формулами:

$$E_{CF1} = \sum_i^n p_{T,i}; E_{CF2} = \sum_{i,j}^n p_{T,i} p_{T,j} \Delta R_{ij}; E_{CF3} = \sum_{i,j,k}^n p_{T,i} p_{T,j} p_{T,k} \Delta R_{ij} \Delta R_{jk} \Delta R_{ki}. \quad (2)$$

ΔR_{ij} определяется как:

$$\Delta R_{ij}^2 = (\eta_i - \eta_j)^2 + (\phi_i - \phi_j)^2. \quad (3)$$

Также иногда вместо переменной D2 для выделения толстой струи используют переменную C2, которая задана следующей формулой:

$$C_2 = \frac{E_{CF1} \times E_{CF2}}{E_{CF3}^2} \quad (4)$$

В работе использованы данные переменные:

1. Энергетические корреляционные функции $D2, C2, ECF2, ECF3$
2. Масса и поперечный импульс струи p_t, m
3. Planar flow (измеряет степень, в которой энергия струи равномерно распределяется по плоскости поперек поверхности струи по сравнению с линейной распределением по поверхности струи)
4. N-subjettiness (для эффективного подсчета подструй в струе) τ_1
5. Апланарность A
6. Момент Фокса-Вольфрама R_2
7. Angularity (переменная, чувствительная к степени симметрии потока энергии внутри струи)
8. KtDR
9. Разделяющая мера d_{12}

3 Процесс работы и результаты

Процесс обучения модели машинного обучения состоит из следующих этапов: обработка и стандартизация данных, разбиение данных на тестовую, обучающую и валидационную выборки, подбор оптимальных гиперпараметров и обучение, тестирование модели и оценка ее результативности с помощью выбранной метрики.

3.1 Подготовка данных

Работа проводится с данными, полученными методом МК моделирования протон-протонного столкновения в детекторе ATLAS на LHC с энергией в системе центра масс 13 ТэВ.

Сигнальное дерево строится на данных фонового процесса образования пары топ-анти топ, так как топ преимущественно распадается через W -бозон. Ограничения, поставленные на сигнальное дерево, схожи с контрольной областью пары топ-анти топ: отбираются события с как минимум одной b -меченной струей, поперечный импульс которой больше 30 ГэВ и модуль псевдобыстроты меньше 2.5.

В фоновом дереве не должно быть W , распадающегося по адронной моде. Поэтому фоновое дерево строится на данных фонового процесса распада Z бозона в электрон-позитрон. Также поставлены ограничения на отсутствие b -меченных струй в событиях данного дерева.

И на фоновое, и на сигнальные деревья наложены ограничения на поперечный импульс толстой струи и на ее инвариантную массу. Ограничения представлены ниже:

$$60\text{GeV} < m < 110\text{GeV}, p_t > 200\text{GeV}$$

Для некоторых событий вычисление ECF3 не возможно. Переменные D2, C2 задавались по формулам, поэтому для событий с ECF3 равным 0 значения этих переменных были не определены. Для дальнейшей работы с данными и для последующего обучения моделей необходимо было удалить события, содержащие неопределенные значения.

Перед обучением проведена стандартизация датасета.

3.2 Обучение и тестирование MLP

Процесс обучения полносвязной нейронной сети заключался в поэтапном усложнении модели, то есть увеличения слоев и нейронов сети, а также в введении L2 регуляризации для устранения переобучения. Подобраны оптимальные значения батча, скорости обучения и функции активации для данной архитектуры.

$$\text{Batch} = 512$$

$$\text{LearningRate} = 0.01$$

$$\text{activation} - \text{ReLU}$$

Изначально была задана однослойная архитектура сети. Ниже представлены зависимости метрик и функций ошибок для однослойных сетей от количества нейронов. Показатели функции ошибок и точности были взяты как для тестовой выборки, так и для обучающей выборки.

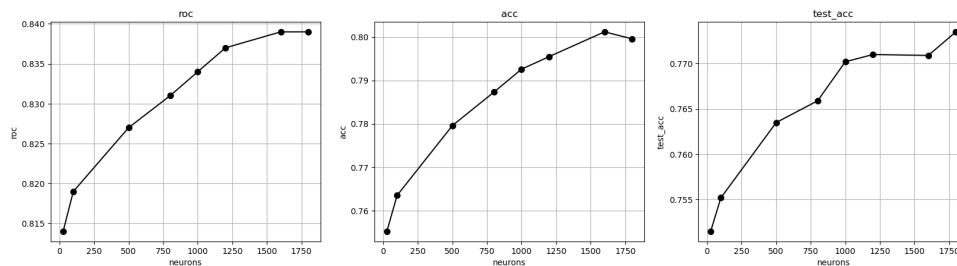


Рис. 10: Зависимости AUC, точности на обучающей и на тестовой выборках от количества нейронов в слое

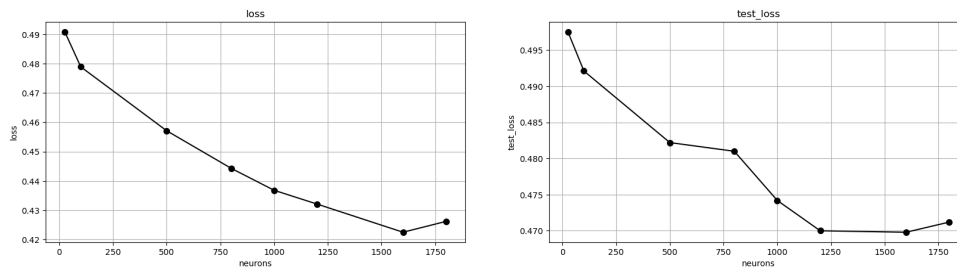


Рис. 11: Зависимости функций ошибок на обучающей и на тестовой выборках от количества нейронов в слое

Для улучшения результатов модели было решено добавить второй скрытый слой с той же функцией активации. При добавлении нового слоя увеличивается количество настраиваемых моделью параметров и, как следствие, сложность модели. Сложность модели - одна из главных причин переобучения, поэтому для устранения переобучения была добавлена L2-регуляризация.

Подбор коэффициента регуляризации заключался в нахождении значения, при котором зависимости между лосс-функцией на обучающей и на валидационной выборке были близки друг к другу, а также сама модель обучалась, а не штрафовала все события. По данным зависимостям можно сделать вывод, что подходящим коэффициентом регуляризации является $L2=0.0001$.

Для двухслойной нейронной сети выбраны аналогичные размер батча и скорость обучения, что и для сети с одним скрытым слоем. Максимальное значение AUC для данной сети 0.842, далее при увеличении сложности сети, уменьшается и количество эпох обучения, то есть сеть стремится переобучиться быстрее. Слагаемое, связанное с коэффициентом регуляризации, при его увеличении замедляет процесс обучения и уменьшает значения оценочных метрик. Поэтому еще большее усложнение двухслойной сети не приводит к хорошим результатам. Далее представлены графики зависимости метрик и функций ошибок для двухслойной сети с различным объемом параметров.

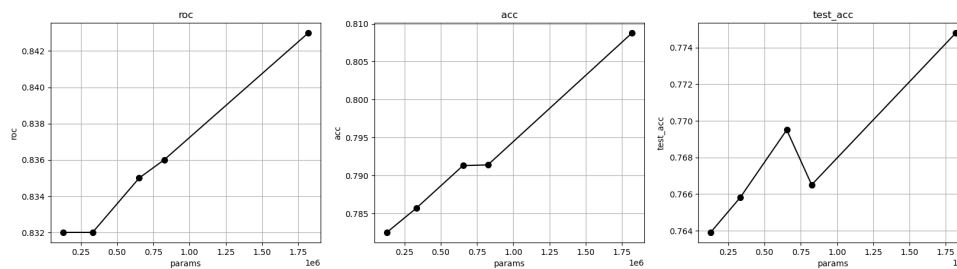


Рис. 12: Зависимости AUC, точности на обучающей и на тестовой выборках от количества параметров сети

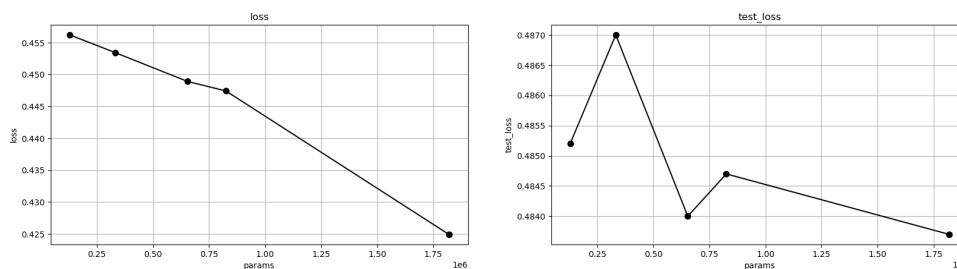


Рис. 13: Зависимости функций ошибок на обучающей и на тестовой выборках от количества параметров сети

Так как при увеличении слоев на данном этапе наблюдалась тенденция роста оценочных метрик, в архитектуру сети был добавлен третий скрытый слой. Также были выбраны аналогичный размер батча и та же функция активации в каждом скрытом слое.

Чем больше слоев в нейронной сети, тем больше ее сложность, как было сказано выше. Для устранения проблемы переобучения вместе с L2-регуляризацией был применен метод прореживания (dropout), который заключается в установлении каждому нейрону сети определенной вероятности того, будет ли работать нейрон в этой эпохе. При отключении нейрона его выход равен 0. Устранение переобучения с помощью комбинации L2-регуляризации вместе с методом прореживания довольно частая практика в машинном обучении. Поэтому следующий шаг - настройка коэффициента L2-регуляризации и dropout.

На данных рисунках изображены зависимости валидационной функции ошибки и функции ошибки на обучающей выборке от количества эпох, то есть разные истории обучения для одной и той же архитектуры - трехслойной сети с 1000 нейронами в каждом слое.

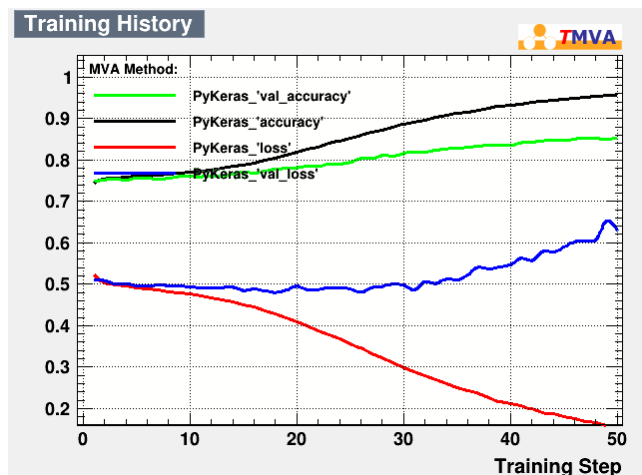


Рис. 14: История обучения для MLP без dropout и L2=0.0001

На графике 14 представлена наглядная иллюстрация переобучения. Несмотря на введенную регуляризацию, по разнице между потерями на валидационной и обучающей выборках можно сделать вывод, что модель не обобщается, а обучает нейроны запоминать обучающую выборку.

При введении dropout = 0.45 до 60 эпох разницу между валидационной функцией ошибки и функцией ошибки при обучении можно считать незначительной, полученная история обучения представлена на рисунке 15.

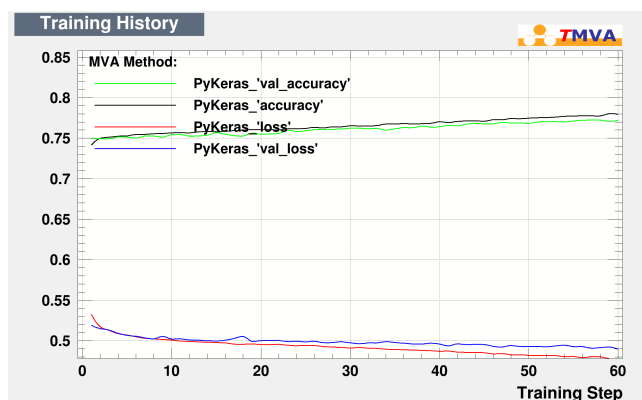


Рис. 15: История обучения для MLP с dropout = 0.45 и L2=0.0001

Для разных количеств нейронов в слоях обучены несколько моделей трехслойной сети, для каждой модели отдельно подбирались оптимальные гиперпараметры. Зависимости метрик и ошибок от количества параметров модели представлены далее.

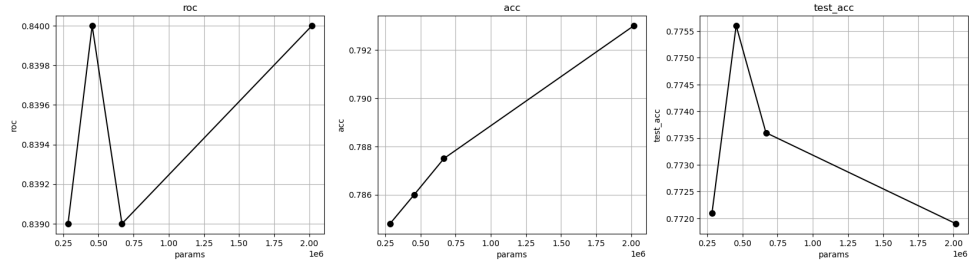


Рис. 16: Зависимости AUC, точности на обучающей и на тестовой выборках от количества нейронов в сети

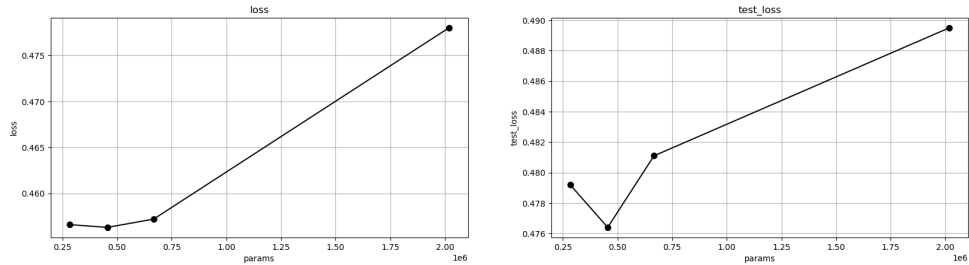


Рис. 17: Зависимости функций ошибок на обучающей и на тестовой выборках от количества нейронов в сети

При добавлении еще одного слоя или увеличении количества нейронов результат моделей не становится лучше, поэтому решено остановиться на трехслойной сети. Наилучший результат продемонстрировала модель с тремя слоями. В первом скрытом слое 200 нейронов, во втором слое 500 нейронов, в третьем - 700. AUC данной модели равен 0.840.

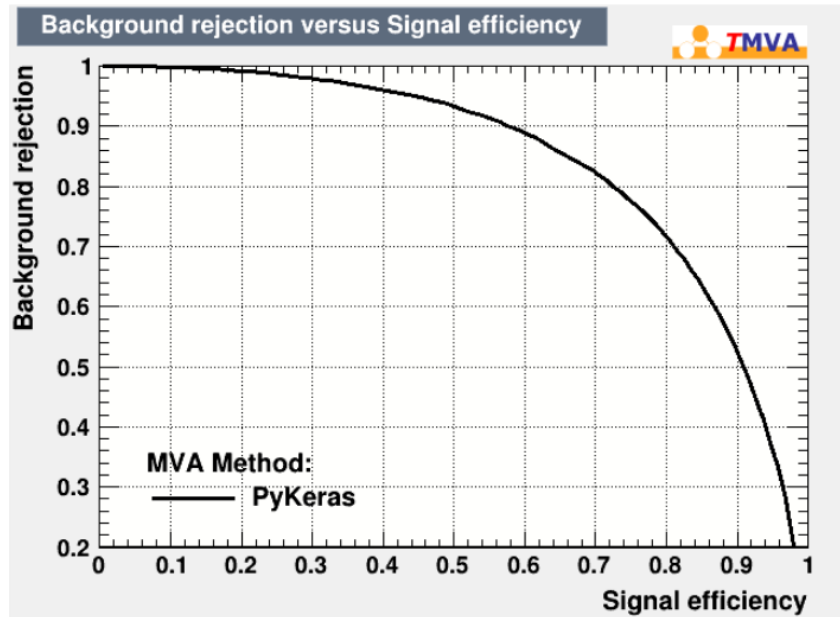


Рис. 18: ROC-кривая наилучшей модели MLP

3.3 Обучение и тестирование CNN

Главным преимуществом сверточных нейронных сетей является то, что модель потенциально должна давать хорошие результаты при ее малой сложности. Были настроены более оптимальные гипер-

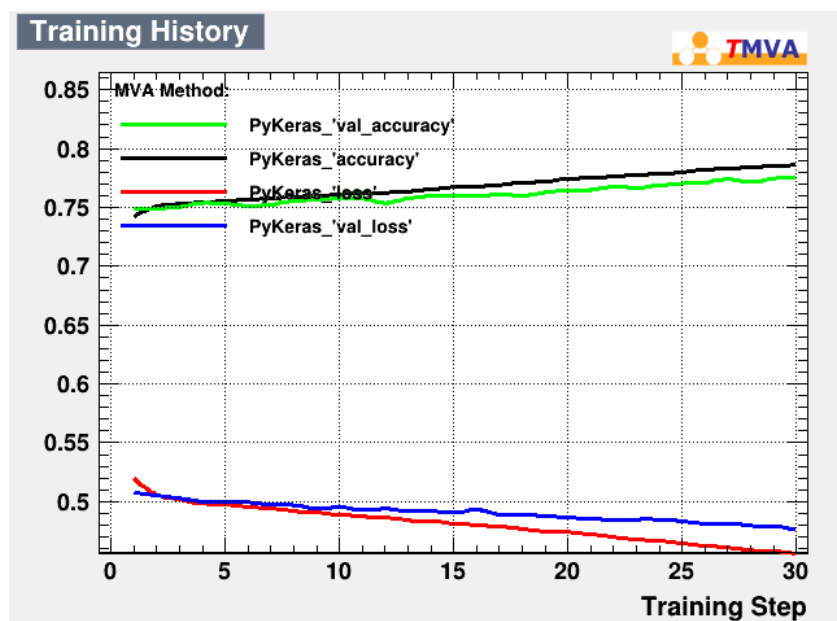


Рис. 19: История обучения модели

параметры для сверточных слоев: количество фильтров - 10, размерность фильтра - 4, смещение(stride) выбрано равным 2, функция активации - ReLU. В пулинговом слое выбрана стратегия выбора максимального значения из карты признаков, размер окна был выбран равным 2. Всего в данных моделях 1 сверточный слой. Размер батча - 512, скорость обучения равна 0.001.

Для конфигурации 1 сверточный слой и 1 скрытый полносвязный слой на рисунках (20-21) представлены зависимости метрик и функции ошибок на обучающих и тестовых данных. В сравнении с MLP CNN уже с одним полносвязным слоем дает результаты лучше по каждому метрикам, чем MLP. Однако сеть склонна к переобучению при увеличении числа нейронов, поэтому в архитектуру добавлено прореживание. Также увеличилось время обучения для сети, максимальное количество эпох обучения для одной из моделей равно 350.

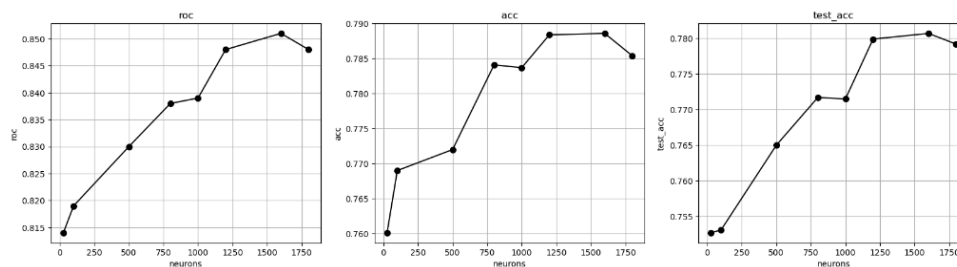


Рис. 20: Зависимости AUC, точности на обучающей и на тестовой выборках от количества параметров сети

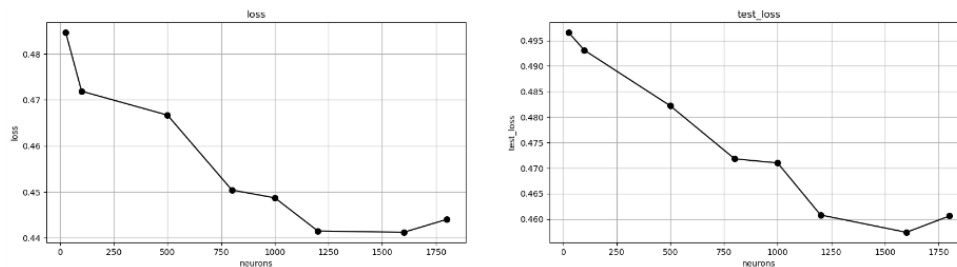


Рис. 21: Зависимости функций ошибок на обучающей и на тестовой выборках от количества параметров сети

Для конфигурации 1 сверточный слой и 2 скрытых полносвязных слоя на рисунках (22-23) представлены зависимости метрик и функции ошибок на обучающих и тестовых данных. Модели сильнее склонны к переобучению, поэтому применялась комбинация регуляризации и метода прореживания нейронов в сети. Дальнейшее увеличение нейронов в слоях лишь увеличивает количество эпох на обучение, а не улучшает оценочные метрики.

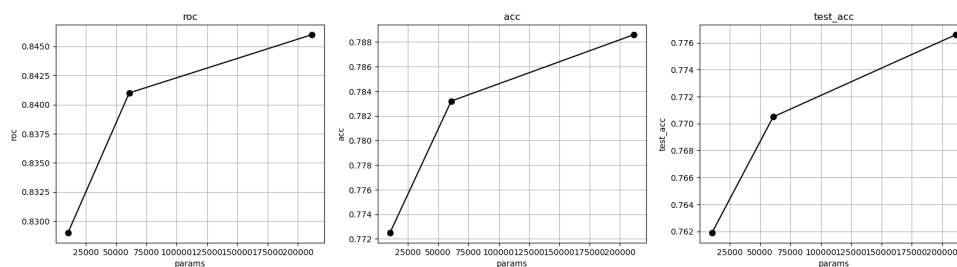


Рис. 22: Зависимости AUC, точности на обучающей и на тестовой выборках от количества параметров сети

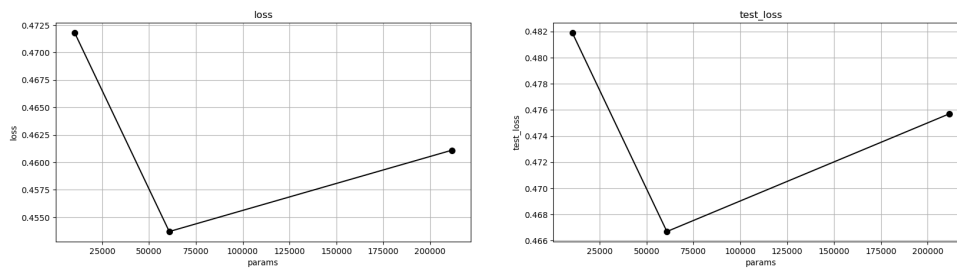


Рис. 23: Зависимости функций ошибок на обучающей и на тестовой выборках от количества параметров сети

Наилучший результат продемонстрировала модель с конфигурацией 1 сверточный слой и 1 скрытый полносвязный слой с 1600 нейронами. AUC данной модели равен 0.850.

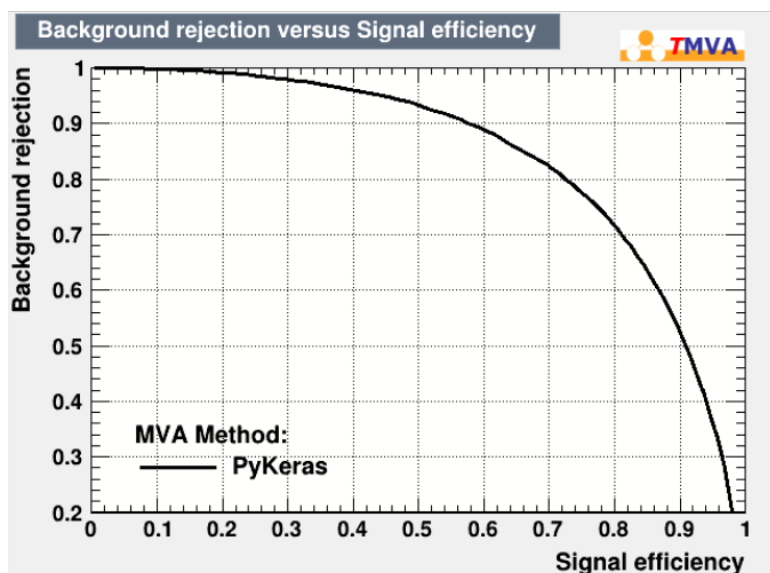


Рис. 24: ROC-кривая наилучшей модели CNN

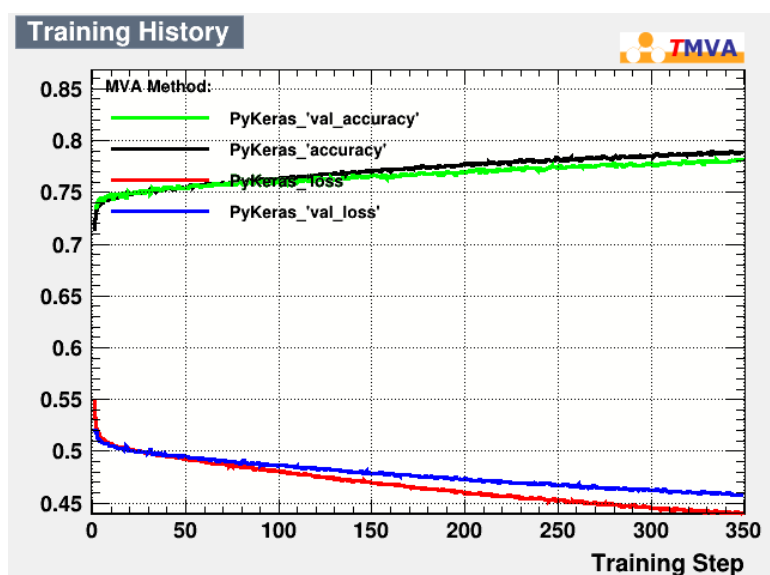


Рис. 25: История обучения

4 Заключение

В рамках НИР за семестр проведено ознакомление с библиотекой глубокого обучения Keras для более гибкого процесса обучения сетей, также проведено ознакомление с архитектурой многослойного перцептрона и сверточной нейронной сети для решения задачи бинарной классификации; сформированы сигнальные и фоновые деревья, содержащие выбранные дискриминирующие переменные для данной задачи. Изучены различные методы предотвращения процесса переобучения. Проведено обучение и тестирование MLP и CNN для разных наборов гиперпараметров и архитектур.

Максимальное значение AUC в прошлом семестре составляло 0.820 у модели BDT с градиентным бустингом. В данной работе получилось улучшить данную оценочную метрику, теперь максимальное значение $AUC = 0.850$ достигается в модели CNN с одним сверточным и одним полносвязным слоями.

В дальнейшей работе будут использоваться результаты модели, которая дала наилучшую оценочную метрику, с помощью данной модели будут классифицироваться события из реальных данных. Задача на будущее будет в сравнении эффективностей классификации данных с помощью выбранной модели и с помощью используемой ранее в анализе дискриминирующей переменной `fat_jet0_wtag50`.

Список литературы

- [1] Школа Анализа Данных. Учебник по машинному обучению. <https://education.yandex.ru/handbook/ml/article/about>.
- [2] Stanford University School of Engineering. Свёрточные нейронные сети для визуального распознавания. <https://www.reg.ru/blog/stenfordskij-kurs-лекция-1-vvedenie/>.
- [3] Francois Chollet. *Deep learning with python*. Manning Publications, New York, NY, 2017.
- [4] Identification of Hadronically-Decaying W Bosons and Top Quarks Using High-Level Features as Input to Boosted Decision Trees and Deep Neural Networks in ATLAS at $\sqrt{s} = 13$ TeV. Technical report, CERN, Geneva, 2017. All figures including auxiliary figures are available at <https://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/PUBNOTES/ATL-PHYS-PUB-2017-004>.