САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Направление: 01.03.02 Прикладная математика и информатика

ООП: Прикладная математика, фундаментальная информатика и программирование

**ОТЧЕТ О НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ**

**Тема задания:** gредсказывание наличия болезни сердца у человека с помощью машинного обучения.

Выполнил: Хлопин Игорь Владимирович 19Б05

Руководитель научно- исследовательской работы: Гришкин Валерий Михайлович

Санкт-Петербург

2021

**Содержание:**

1. **Введение** ………………………………………………………………...………….........2
   1. Что такое машинное обучение……….……………………………………………...2
   2. Виды задач в машинном обучении ………………………………............................2
2. **Проведение исследования**………………………………………………..…………….5
   1. Исследуемая задача……………………………………………………......................5
   2. Работа с данными ……………………………………………………........................6
   3. Преобразование данных………….…………………………………….....................7
   4. Архитектура нейронной сети, выбор оптимизатора, функции ошибки и метрики……………………………………………………………………………….7
   5. Обработка и исследование данных для обучения………………………………….9
   6. Обучение…………………………………………………………………………….10

**Ссылки**…………………………………………………………………………..…………17

Код программы (нейронной сети) - <https://github.com/IgorvH-Developer/ColaboratoryFailes/blob/main/Heart_Failure_Prediction.ipynb>

1. **Введение**

Машинное обучение с каждым днем занимает всё большее место в нашей жизни ввиду огромного спектра его применений. Начиная от анализа пробок и заканчивая самоуправляемыми автомобилями, всё больше задач перекладывается на самообучаемые машины.

Машинное обучение может выполнять те действия, которые не под силу человеку (или на данное действие у человека уйдет огромное кол-во времени). Если развивать данное направление ИИ, то можно достичь неописуемых результатов в строительстве, науке, медицине и т.д

Развитие современной технологии показывает недостаток знаний человеком и необходимость создания искусственного интеллекта, который бесспорно бы явился помощником человека, в решении ряда проблем, в том числе глобального характера. В наши дни информация вышла на приоритетное место среди критериев прогресса, как и средства ее получения, обработке и использования — компьютер и компьютерные технологии, с помощью которых это осуществляется, активно развиваются.

* 1. **Что такое машинное обучение**

Машинное обучение — это направление в науке, а с недавних пор и в технологиях, которое решает задачу обучения компьютеров. Под этим понимают передачу аппаратно-программным комплексам какого-то сугубо ограниченного набора знаний с возможностью их последующего накопления. В данном контексте не предполагается полноценное обучение, сравнимое с человеческим. Это совсем не то, что поанглийски называют by heart («с глубоким пониманием»), а, скорее, в результате компьютер приобретает машинное ролевое знание. ML реализуется на вычислительных и статистических принципах, объединяя самые разные подходы, включая теорию вероятностей, статистику, логику, вычислительную оптимизацию, поисковые методы, обучение с подкреплением, теорию управления и, возможно, что-то иное. Область применения ML с каждым днем растёт.

* 1. **Виды задач в машинном обучении**

1. Обучение с учителем.

Наиболее распространённый случай. Каждый объект представляет собой пару «набор признаков, ответ». Требуется найти функциональную зависимость ответов от описаний объектов и построить алгоритм, принимающий на входе описание объекта и выдающий на выходе ответ.

x — многомерный вектор данных, характеризующих объект

y — значение, которое нужно предсказать

Для предсказания строим зависимость f(x) = y.

Так, например, программы учат распознавать объекты на фотографиях – программа просматривает миллионы изображений с описанием того, что на них изображено (дерево или облако). Она находит общие черты и уже сама учится давать описания изображениям. Учитель показывает изображение без описания, а программа выдаёт своё предсказание. Если предсказание совпадает с ответом, то программа понимает, что сделала правильные выводы.

Например, поиск по фото работает по такому же принципу. Сеть, обученная на огромном количестве картинок, научилась выделять на них объекты и понимать что это за объекты.

Похожим образом работает программа, если ей показывать вместо изображения вектор признаков объекта.

Несколько видов задач решающихся методом “с учителем”:

* 1. Классификация. Отличается тем, что множество допустимых ответов конечно. Их называют метками классов. Класс — это множество всех объектов с данным значением метки.
  2. Регрессия. Отличается тем, что допустимым ответом является действительное число или числовой вектор.
  3. Ранжирование. Отличается тем, что ответы надо получить сразу на множестве объектов, после чего отсортировать их по значениям ответов. Может сводиться к задачам классификации или регрессии. Часто применяется в информационном поиске и анализе текстов.

1. Обучение без учителя

Изучает широкий класс задач обработки данных, в которых известны только описания множества объектов, и требуется обнаружить внутренние взаимосвязи, зависимости, закономерности, существующие между объектами. Обучение без учителя часто противопоставляется обучению с учителем.

Такой подход изучается для выполнения тех задач, где присутствует неочевидное решение. Например, в том же маркетинге. ИИ не понимает, что предлагать похожий товар человеку, который в нем не нуждается, нелогично.

Также нейросети могут обучаться не самостоятельно, а в паре. Так работает генеративно-состязательная сеть (GAN). Она состоит из сетей G и D – первая на основе реальных изображений генерирует образцы, а вторая пытается отличить подлинные образцы от неправильных. Одна из компаний, которая использует GAN, – Facebook.

Типы задач решающиеся “без учителя”:

* 1. Кластеризация. Задача заключается в том, чтобы сгруппировать объекты в кластеры, используя данные о попарном сходстве объектов.
  2. Поиск ассоциативных правил. Исходные данные представляются в виде признаковых описаний. Требуется найти такие наборы признаков, и такие значения этих признаков, которые особенно часто (неслучайно часто) встречаются в признаковых описаниях объектов.
  3. Сокращения размерности. Главная задача – это исходным признакам с помощью некоторых функций преобразования перейти к наименьшему числу новых признаков, не потеряв при этом никакой существенной информации об объектах выборки.

1. Глубокое обучение

Глубокое обучение может быть как с учителем, так и без, но оно подразумевает под собой анализ Big Data – настолько большой информации, что одного компьютера будет недостаточно. Поэтому Deep Learning использует для работы нейронные сети.

Нейронные сети позволяют разделить одну большую задачу на несколько маленьких и делегировать их другим устройствам. Например, один процессор собирает информацию и передает ее двум другим. Те, в свою очередь, анализируют ее и передают еще четырем, которые выполняют еще какие-то задачи и передают следующим процессорам.

1. **Проведение исследования.**
   1. **Исследуемая задача**

**Цель исследования:** используя методы машинного обучения с помощью нейронной сети добиться наилучшей точности предсказаний наличия болезней сердца, обучаясь на данном датасете.

**Проблемы:** основная проблема - это маленький объем данных. В наборе данных для обучения примерно 900 объектов, а исключив объекты с незаполненными признаками, в таблице останется 746 строк. Из-за маленького количества объектов качество предсказаний будет хуже.

**Используемые методы:** среда разработки – Colaboratory, язык программирования – Python, для предсказывания используем нейронную сеть вида – многослойный персептрон, состоящий из двух скрытых слоёв, в каждом из которых по 5 нейронов.

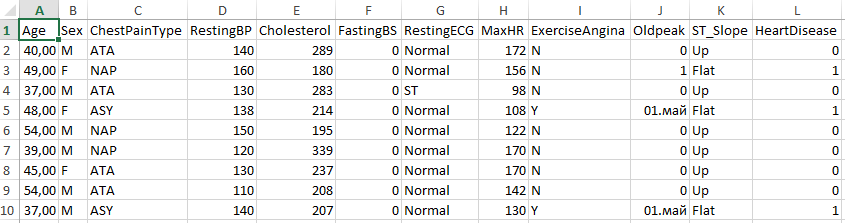
**Возможные применения:** обученная нейронная сеть для предсказывания наличия болезни сердца может облегчить процесс постановки диагноза, сохранив время врачей.

**Подобные задачи:** в той же медицине, нейронные сети выполняют задачи распознавания объектов на рентгеновских снимках ( например опухолей), эта задача так же как и наша является задачей классификации, но вместо многослойного персептрона используются свёрточные нейронные сети( как для большинства задач работающих с фото).

**Алгоритм обучения:**

1. Загрузка базы данных.
2. Изменение признаков объектов для более комфортного обучения.(от 11 признаков до 19).
3. Создание матрицы корреляции и выделение двух наборов признаков (исходный – 19 признаков, новый – 7 признаков).
4. Создание архитектуры нейронной сети, выбор функции ошибки, оптимизатора и метрики.
5. Обучение на наборе данных из 19 признаков.
6. Обучение на наборе данных из 7 признаков.
7. Сравнение результатов и выбор лучшего набора.
   1. **Работа с данными.**

Чтобы понять к какому виду относиться данная задача рассмотрим набор данных для этой задачи.



Каждый объект представлен парой <x, y> (набор признаков, ответ).

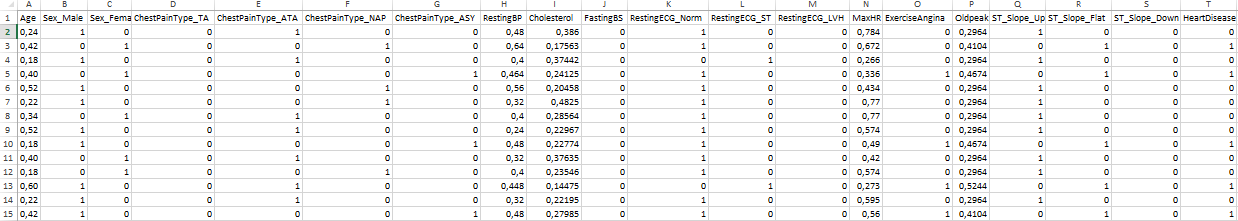
Вектор описания объекта x состоит из 11 признаков:

1. Age: возраст пациента [лет]
2. Sex: пол пациента [M: Мужчина, F: Женщина]
3. ChestPainType: тип боли в груди [TA: Типичная стенокардия, ATA: Атипичная стенокардия, NAP: Неангинальная боль, ASY: Бессимптомный]
4. RestingBP: артериальное давление в состоянии покоя [мм рт.ст.]
5. Cholesterol: сывороточный холестерин [мм/дл]
6. FastingBS: уровень сахара в крови натощак [1: если BS натощак > 120 мг/дл, 0: в противном случае]
7. RestingECG: результаты электрокардиограммы в состоянии покоя [Normal: Норма, ST: наличие аномалии зубца ST-T (инверсии зубца T и/или повышение или понижение ST > 0,05 мВ), LVH: наличие вероятной или определенной гипертрофии левого желудочка]
8. MaxHR: достигнутая максимальная частота сердечных сокращений [Числовое значение от 60 до 202]
9. ExerciseAngina: стенокардия, вызванная физической нагрузкой [Y: Да, N: Нет]
10. Oldpeak: oldpeak = ST [Числовое значение, измеренное в депрессии]
11. ST\_Slope: наклон сегмента ST пикового упражнения [Up, Flat, Down]

HeartDisease - это значение y (1- есть болезнь сердца, 0 - норма)

В данном наборе данных каждый объект описывают 11 признаков, которым соответствует значение(1 либо 0), показывающее наличие заболевания сердца. Следовательно, данную задачу можно отнести к задачам **классификации**.

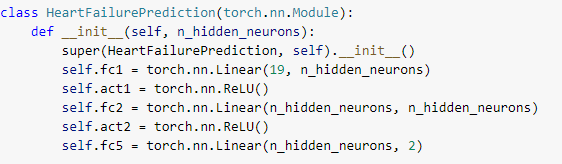
* 1. **Преобразование данных**



Для удобства работы с данными:

1. Столбики с признаками принимающие произвольные значение были отображены на отрезок [0,1], где 0 – минимальное значение в столбике, 1 – максимальное значение в столбике.
2. Столбики с признаками принимающие значение из заранее известного небольшого множества вариантов были разделены на отдельные столбики(отдельные столбик для каждого варианта значения признака). Например: признак “пол” принимающий значение либо M либо F был разделён на 2 столбика sex\_Male, sex\_Female, каждый их которых теперь принимает значение либо 1 либо 0.
   1. **Архитектура нейронной сети, выбор оптимизатора, функции ошибки и метрики**

1. Архитектура нейронной сети



Инициализируем сеть:

сеть

Схема многослойного персептрона:

x1 y1,1=Relu(x1;x2;..;xn+b1,1) y2,1=Relu(y1,1;y1,2;..;y1,5+b2,1)

x2 y1,2=Relu(x1;x2;..;xn+b1,2) y2,2=…

x3 y1,3=Relu(x1;x2;..;xn+b1,3) y2,3=… y3,1

. y3,2

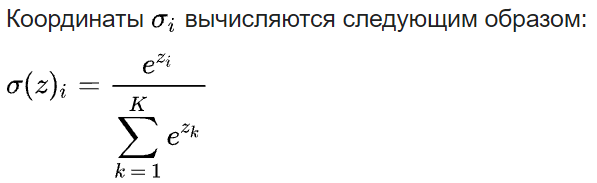
.

xn y1,5=Relu(x1;x2;..;xn+b1,5) y2,5=…

На вход нейронной сети подаётся вектор из n признаков(рассмотрим случаи 19 и 7 признаков), этот вектор проходит через 2 полносвязных нейронный слоя(по 5 нейронов в каждом). А на выходе получаем вектор из 2 элементов, которые отправляем в функцию активации Softmax, после этого сумма значений вектора равняется единице, в итоге получаем наш prediction. Если обозначить наибольшее значение вектора за 1, а наименьшее за 0, то можем сравнить наш prediction([1,0] либо [0,1]) с целевой функцией([1,0] – есть болезнь сердца, [0,1] - норма) и проверить точность предсказания.

prediction = Softmax([y3,1;y3,2])

Формула softmax:



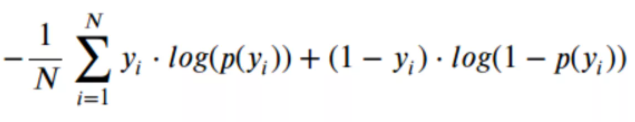
1. Функция потерь - бинарная кросс-энтропия



Pred = Softmax ( [y3,1;y3,2] )

Target = либо [1;0] – есть болезнь сердца, либо [0;1] – норма

Формула бинарной кросс-энтропии:



y = target

p(y) = pred

1. Оптимизатор

Я выбрал оптимизатор Adam с шагом обучения равным 0.0001:

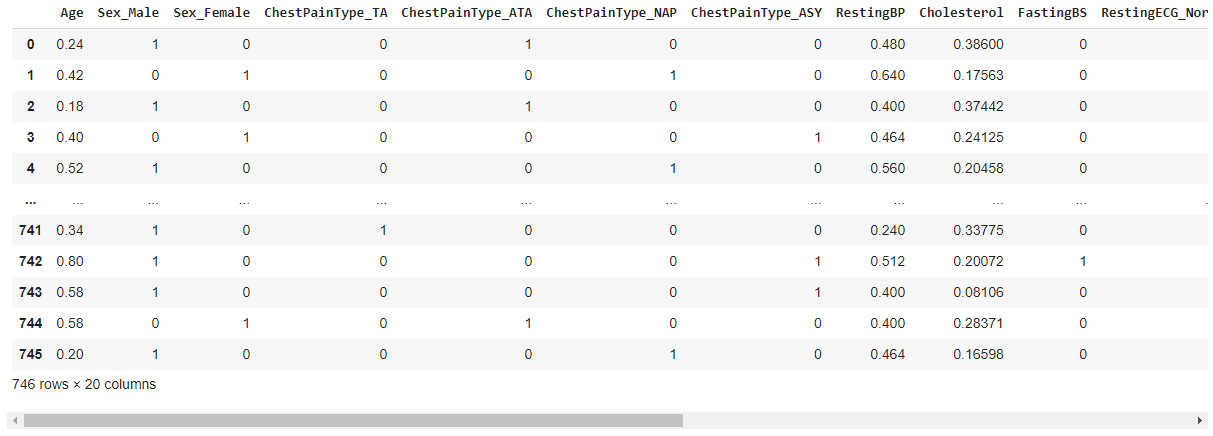


1. Метрика

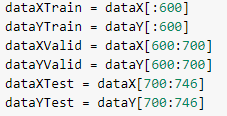
Метрика будет показывать насколько качественно обучена сеть. Моя метрика – это отношение правильных предсказаний к общему количеству предсказаний (правильные / всего).

* 1. **Обработка и исследование данных для обучения**

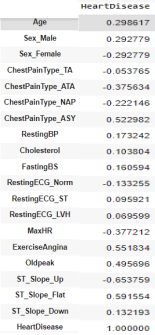
Загружаем базу данных:

****

В ней 746 объектов. Разделим их на 3 группы (train, valid, test).



Далее рассмотрим как каждый из признаков коррелируется с наличием заболевания.



Можно сделать вывод, что наличие болезни больше всего коррелируется с признаками: ChestPainType\_ATA, ChestPainType\_ASY, MaxHR, ExerciseAngine, Oldpeak, ST\_Slope\_Up, ST\_Slope\_Flat.

Обучим сеть с этим набором признаков и с набором из всех признаков и сравним результаты.

* 1. **Обучение**

Создаём нейронную сеть:



Создаём алгоритм обучения:



Будем обучать нейронную сеть на тренировочном датасете(dataTrain), проверять качество её работы на валидационном датасете(dataTrain) в процессе обучения, а на тестовом датасете(dataTest) проверять после завершения обучения.

На каждой эпохе подаём сети на вход батч размером с сам тренировочный датасет(600 объектов), ищем функцию потерь на этом батче, далее с помощью оптимизатора и функции потерь изменяем весы нейронной сети. Каждую тысячную эпоху проверяем как изменились функции потерь на тренировчном и валидационных датасетах и метрика на тренировочном датасете.

1. Обучение на наборе всех признаков.

Рассмотрим как менялись метрики и функций потерь в процессе обучения. Запускаем обучение в 70000 эпох.

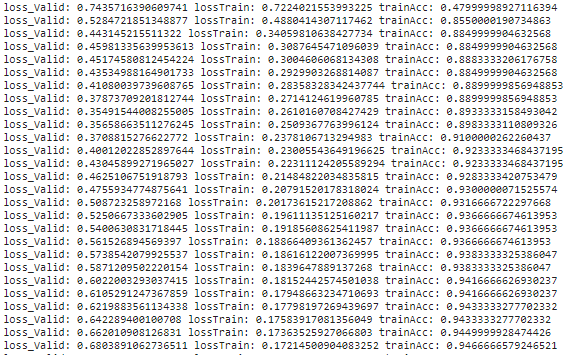
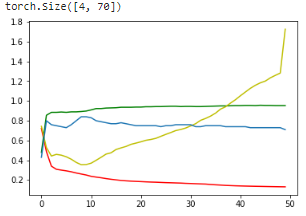


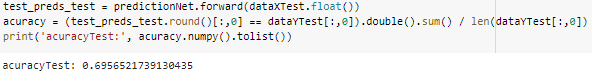
График:



гор. – эпохи(в тысячах), верт. – значения функций потерь и метрик

красный–функ. потерь(train), жёлтый – функ. потерь (valid), синий–метрика(valid), зелёный– метрика(train)

Значение метрики на тестовом датасете (после обучения в 70000 эпох) = 0.69:



Посмотрев на список и на график можно понять, что на тестовом датасете сеть показывает хороший результат, почти безошибочно предсказывая наличие болезни сердца. Обратив внимание на метрику и функцию потерь валидационного датасета замечаем, что метрика достигает своего пика примерно на 10 тысячах эпох, а после этого ситуация только ухудшается. Поэтому при обучении сети остановимся на 10 тысячах эпох.

Результат после обучения в 10 тысячах эпох:

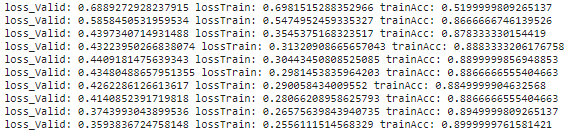
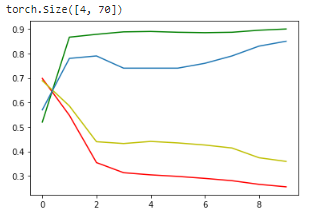
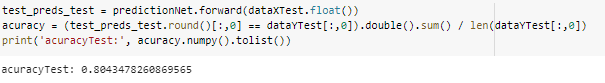


График:



Значение метрики на тестовом датасете = 0.8:



Значения метрики на валидационном и тестовом датасетах увеличились и теперь оба примерно равняются 0.8, следовательно мы улучшили качество работы нейройнной сети. На этом обучений заканчивается.

Вывод: нейронная сеть обученная на полном наборе признаков предсказывает наличие болезни сердца в 80% случаев.

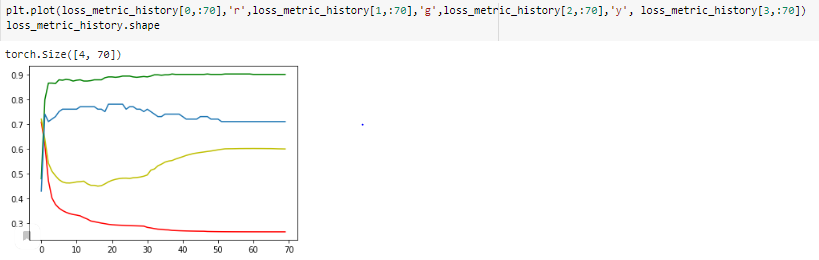
1. Обучение на наборе из 7 признаков.

Сравним качество обучения, что выбрать лучший способ.

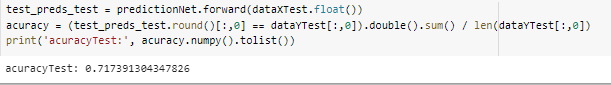
Процесс обучения:



График изменения метрик и функций потерь:



Значение метрики на тестовом датасете (после обучения в 70000 эпох) = 0.71:



Как и в случае с 19 признаками метрика со временем незначительно уменьшается, а метрика возрастает, но не так стремительно по сравнению с сетью с входом из 19 признаков. Возможно не такой значительный рост связан с тем, что сеть с входом из 7 признаков больше выявляет закономерности в них, чем подстраивается под данные тренировочного датасета.

Как мы видим на графики, наилучший результат сеть показывает в районе двадцати эпох, поэтому обучим её снова, но на этот раз остановимся на 20-ой эпохе.

Процесс обучения:

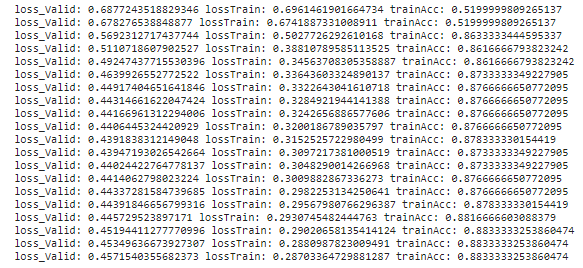
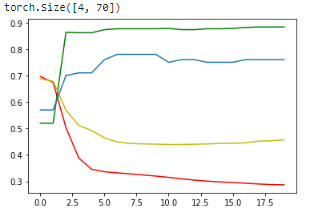
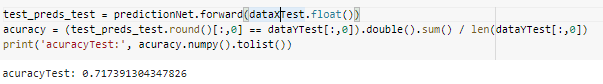


График:



Метрика на тестовом датасете = 0.71



Функция потерь на валидационном датасете уменьшилась, но метрика на тестовом датасете как была 0.71, так и осталась.

Вывод: Нейронная сеть с семью входными признаками верно предсказывает наличие болезни сердца в 71% процентах случаев.

**Вывод по всему процессу обучения:** лучший результат показала нейронная сеть с 19 входными признаками. Обучив эту сеть стало ясно,что нейронная сеть с 19-ю входными признаками, с двумя скрытыми слоями (по 5 нейронов каждый), с бинарной кросс-энтропией в роли функции потерь и с оптимизатором Adam может предсказывать наличие болезни сердца в 80% случаев.

Ссылки:

1. Моя работа в Colaboratory - <https://github.com/IgorvH-Developer/ColaboratoryFailes/blob/main/Heart_Failure_Prediction.ipynb>
2. Исходный датасет - <https://www.kaggle.com/fedesoriano/heart-failure-prediction>
3. Типы задач машинного обучения (30с) - <https://www.hse.ru/data/2017/05/14/1171296413/Григорий%20Сапунов%20—%20Введение%20в%20машинное%20обучение.pdf>
4. Искусственный интеллект и машинное обучение - <https://ru.wikipedia.org/wiki/Искусственный_интеллект>
5. Многослойный персептрон - <https://ru.wikipedia.org/wiki/Многослойный_перцептрон_Румельхарта>
6. Оптимизаторы(Adam) - <https://habr.com/ru/post/318970/>