Изображение выглядит как текст

Автоматически созданное описание

**Задание 1**

Построим нейронную сеть из 1 нейрона и обучим ее на 2 датасетах. Воспользуемся библиотекой pytorch для построения нейронной сети. Возьмем некоторые функции, в качестве функции активации:



Но для функций tanh и atan необходимо сделать дополнительные преобразования:

Изображение выглядит как текст

Автоматически созданное описание

Оптимизаторы:

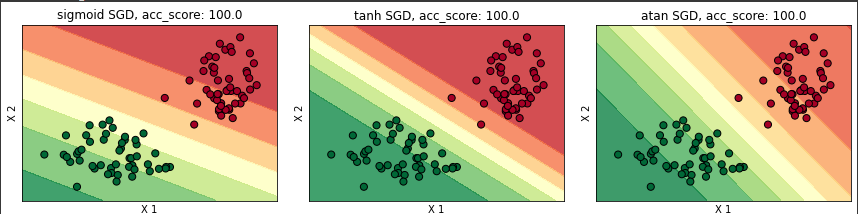


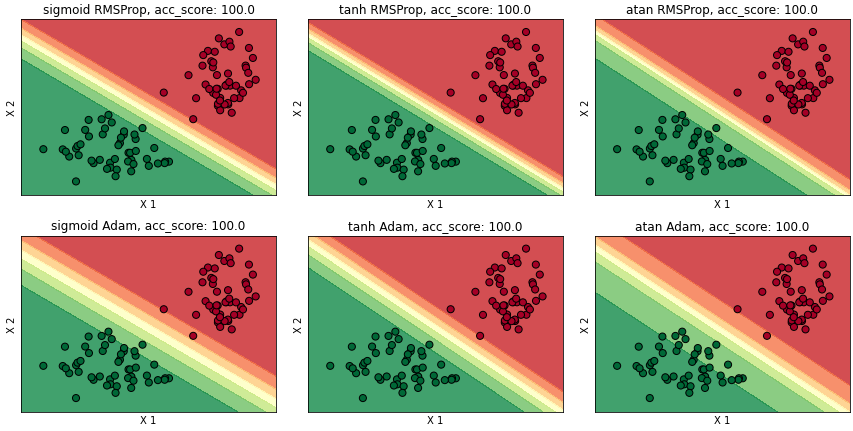
Разобьем обучающую и тестовую выборки в соотношении 80%:20%. Для датасета **nn\_0** получены сл. Результаты:

При количестве эпох для обучения 10, все модели сходятся с точностью в среднем 77%.

При количестве эпох для обучения 100, все модели сходятся с точностью в среднем 100%.

Представим графическое разбиение пространства признаков для различных функций оптимизации и оптимизаторов (при количестве эпох для обучения 100):





Как заметим по графикам, то лучшее разбиение происходит при функции активации tanh с оптимизатором RMSProp.

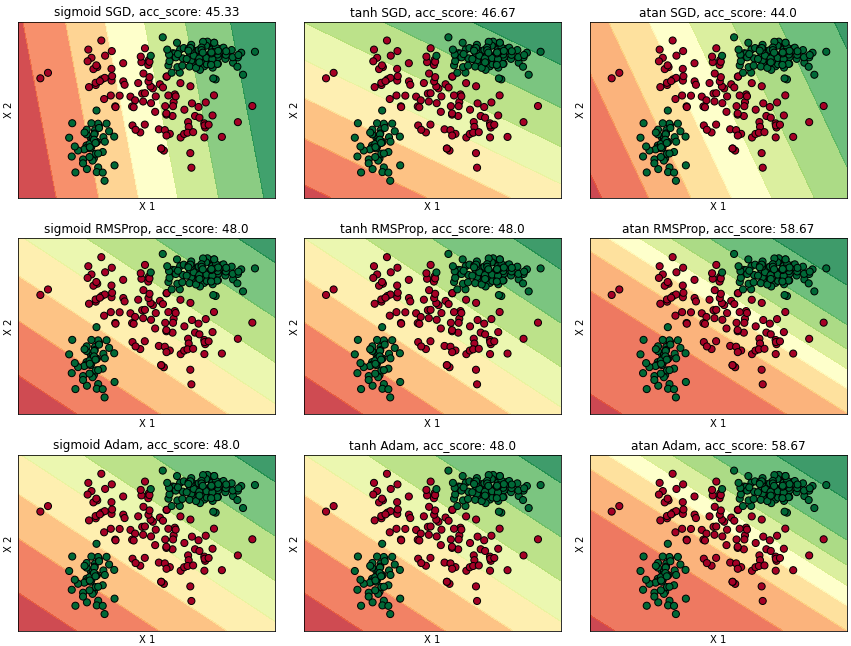
Для датасета **nn\_1** получены следующие результаты:

При количестве эпох для обучения 10, все модели сходятся с точностью в среднем 48%

При количестве эпох для обучения 100, все модели сходятся с точностью 47%

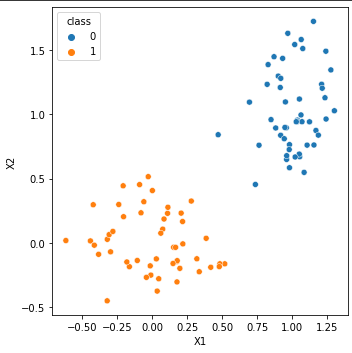
Точность моделей при данном датасете сильно хуже, чем при предыдущем.

Представим графическое разбиение пространства признаков для различных функций оптимизации и оптимизаторов (при количестве эпох для обучения 10):

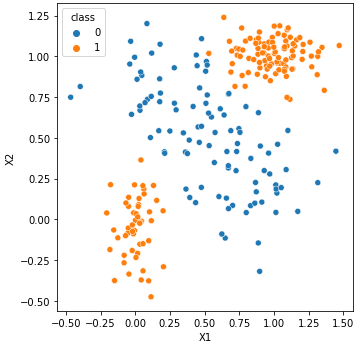


Можем заметить, что лучшее разбиение происходит при функции активации atan с оптимизаторами RMSProp/Adam.

Для того, чтобы понять почему результаты так сильно отличаются, достаточно взглянуть на изначальные данные. Выведем данные из датасета **nn\_0:**



**Nn\_1:**



Можно сделать вывод, что данные из датасета nn\_1 невозможно разделить линейно, а вот из nn\_0 хорошо разделяются линейно. Для nn\_1 стоит использовать нейронную сеть с количеством нейронов > 1.

**Задание 2**

Для модификации нейронной сети из пункта 1 с целью достижения минимальной ошибки на датасете nn\_1.csv мы добавили еще один слой в нашу нейронную сеть.

Количество нейронов = 10

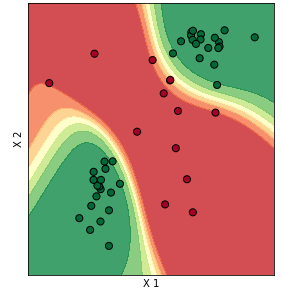
Количество эпох для обучения = 100

В качестве функции активации используем atan

В качестве оптимизатора используем Adam

Наша точность составила 100%

Выведем график разбиения пространства:



Изменения были сделаны из-за того, что данные из датасета nn\_1 не разделимы линейно, и нейронная сеть из 1 нейрона не справлялась со своей задачей. Мы увеличили количество нейронов. После изменений, мы доказали, что результат улучшился, а точнее точность модели увеличилась.

**Задание 3**

Я уже делал подомное задание по другому предмету.

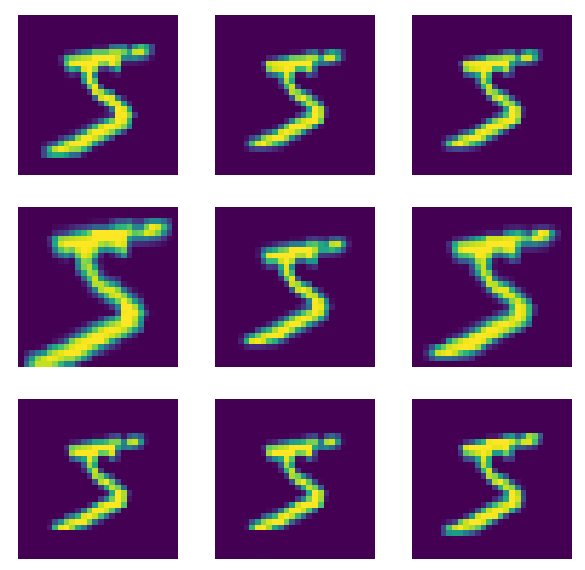
Воспользуемся библиотекой Tensorflow.

Перед началом работы для нашего датасета произведем аугментацию данных. Для этого воспользуемся следующей функцией

Изображение выглядит как текст

Автоматически созданное описание

Она случайно увеличит или уменьшит исходное изображение на случайную величину. Таким образом, в нашем датасете окажется больше данных, что более благоприятно скажется на результате. Приведем пример, как изменяются изображения:



Создадим модель со следующими параметрами:

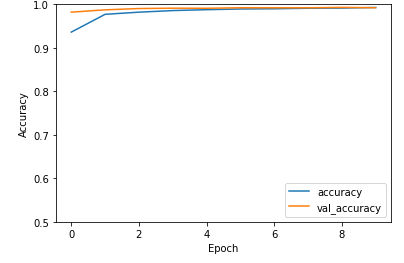
Изображение выглядит как текст

Автоматически созданное описание

В качестве оптимизатора будем использовать оптимизатор Adam.

Обучим модель с количеством эпох 10.

Оценим полученные результаты:  
Изменение точности в зависимости от эпохи обучения:



Окончательная точность:

Изображение выглядит как текст

Автоматически созданное описание

Мы создали и обучили нейронную сеть, которая способна классифицировать рукописные цифры с точностью > 99%.