Федеральное государственное бюджетное образовательное учреждение высшего образования

«Московский Авиационный Институт»

(Национальный Исследовательский Университет)

Факультет №8 «Информационные технологии и прикладная математика»

Кафедра 805 «Прикладная математика»

**Реферат**

**по курсу «Языки и методы программирования»**

**2 семестр**

**Тема:**

**Создание нейронной сети на языке C++**

**Автор:**

Студент 1 курса, гр. М80-103Б-21

Белоносов Кирилл Алексеевич

**Руководитель:**

Севастьянов Виктор Сергеевич

Москва, 2022

Оглавление

[**Введение** 3](#_Toc100993843)

[**Знакомство с нейронными сетями** 4](#_Toc100993844)

[**Написание простейшей нейронной сети с использованием сигмоидальной функции** 6](#_Toc100993845)

[**Написание классификатора для распознавания цифр** 9](#_Toc100993846)

[**Написание нейронной сети на Python с использованием библиотеки Keras** 12](#_Toc100993847)

[**Итоги** 14](#_Toc100993848)

[**Список литературы:** 15](#_Toc100993849)

[**Ссылки** 16](#_Toc100993850)

# **Введение**

Основной целью данного проекта является знакомство с архитектурой нейронных сетей, написание простейшей нейронной сети для распознавания цифр.

Основные задачи:

1. Знакомство с устройством нейронных сетей
2. Написание простейшей нейронные сети с использованием сигмоидной функции активации
3. Написание классификатора для распознавания цифр
4. Написание нейронной сети на Python с использованием библиотеки Keras
5. Подвести итоги

# **Знакомство с нейронными сетями**

**Нейронная сеть** (также искусственная нейронная сеть, ИНС) — математическая модель, а также её программное или аппаратное воплощение, построенная по принципу организации и функционирования биологических нейронных сетей — сетей нервных клеток живого организма. Это понятие возникло при изучении процессов, протекающих в мозге, и при попытке смоделировать эти процессы.

ИНС представляет собой систему соединённых и взаимодействующих между собой простых процессоров (искусственных нейронов). Такие процессоры обычно довольно просты (особенно в сравнении с процессорами, используемыми в персональных компьютерах). Каждый процессор подобной сети имеет дело только с сигналами, которые он периодически получает, и сигналами, которые он периодически посылает другим процессорам. И, тем не менее, будучи соединёнными в достаточно большую сеть с управляемым взаимодействием, такие по отдельности простые процессоры вместе способны выполнять довольно сложные задачи.

Несмотря на большое разнообразие вариантов нейронных сетей, все они имеют общие черты. Так, все они, так же, как и мозг человека, состоят из большого числа связанных между собой однотипных элементов – нейронов, которые имитируют нейроны головного мозга. Нейрон состоит из синапсов, связывающих входы нейрона с ядром; ядра нейрона, которое осуществляет обработку входных сигналов и аксона, который связывает нейрон с нейронами следующего слоя. Каждый синапс имеет вес, который определяет, насколько соответствующий вход нейрона влияет на его состояние. Состояние нейрона определяется по формуле:

Затем определяется значение аксона нейрона по формуле:

Где f – некоторая функция активации

Нейронные сети обратного распространения – это мощнейший инструмент поиска закономерностей, прогнозирования, качественного анализа. Такое название – сети обратного распространения (back propagation) они получили из-за используемого алгоритма обучения, в котором ошибка распространяется от выходного слоя к входному, т. е. в направлении, противоположном направлению распространения сигнала при нормальном функционировании сети.

Нейронная сеть обратного распространения состоит из нескольких слоев нейронов, причем каждый нейрон слоя i связан с каждым нейроном слоя i+1, т. е. речь идет о полносвязной НС.

В общем случае задача обучения НС сводится к нахождению некой функциональной зависимости Y=F(X) где X – входной, а Y – выходной векторы. В общем случае такая задача, при ограниченном наборе входных данных, имеет бесконечное множество решений. Для ограничения пространства поиска при обучении ставится задача минимизации целевой функции ошибки НС.

Обучение нейросети производится методом градиентного спуска, т. е. на каждой итерации изменение веса производится по формуле:

Где – параметр отвечающий за скорость обучения

При этом множитель:

Введем вспомогательную переменную:

Тогда изменение веса равно:

Алгоритм обучения нейронной сети:

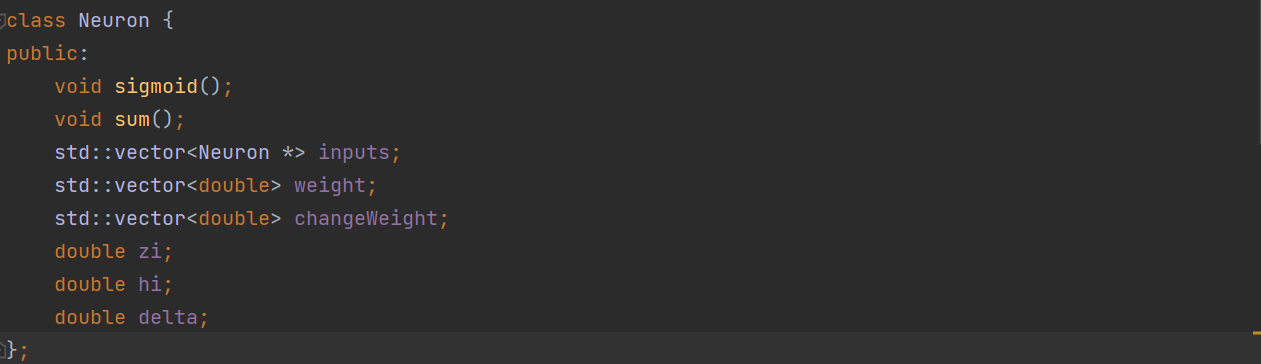
1. подать на вход НС один из требуемых образов и определить значения выходов нейронов нейросети
2. рассчитать для выходного слоя НС по формуле (12) и рассчитать изменения весов выходного слоя N по формуле (13)
3. Рассчитать по формулам (11) и (13) соответственно и Δw(N)ij для остальных слоев НС, n = N-1..1
4. Скорректировать все веса НС
5. Если ошибка существенна, то перейти на шаг 1

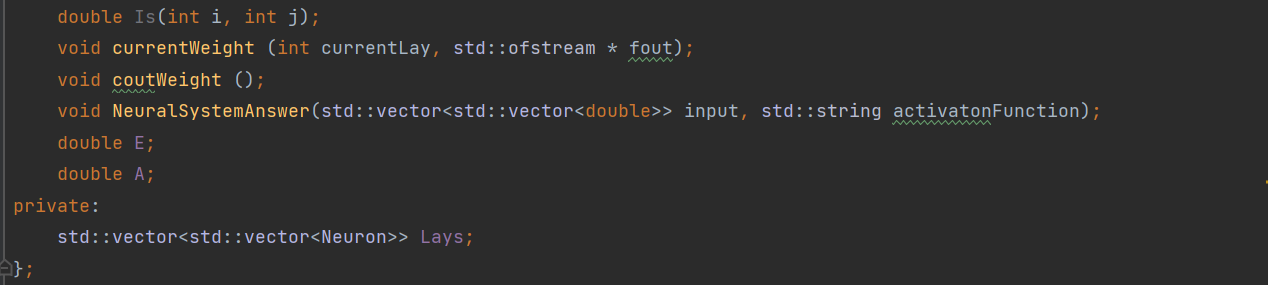
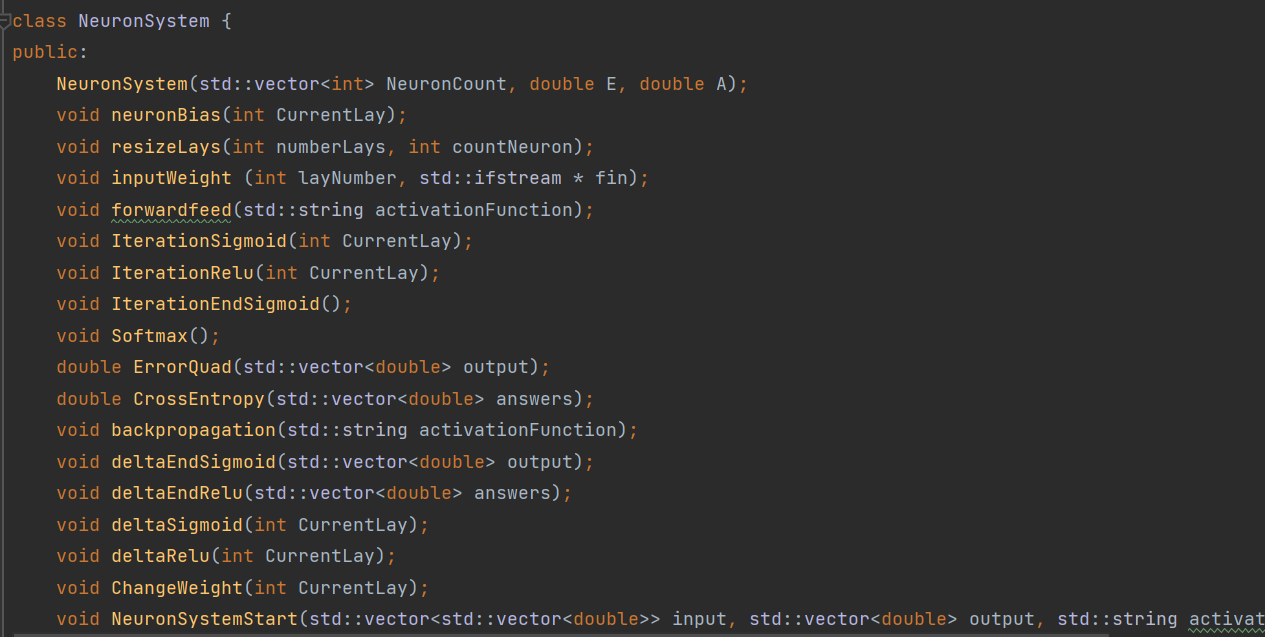
# **Написание простейшей нейронной сети с использованием сигмоидальной функции**

Функция активации — это способ нормализации входных данных То есть, если на входе у нас будет большое число, пропустив его через функцию активации, мы получите выход в нужном вам диапазоне.

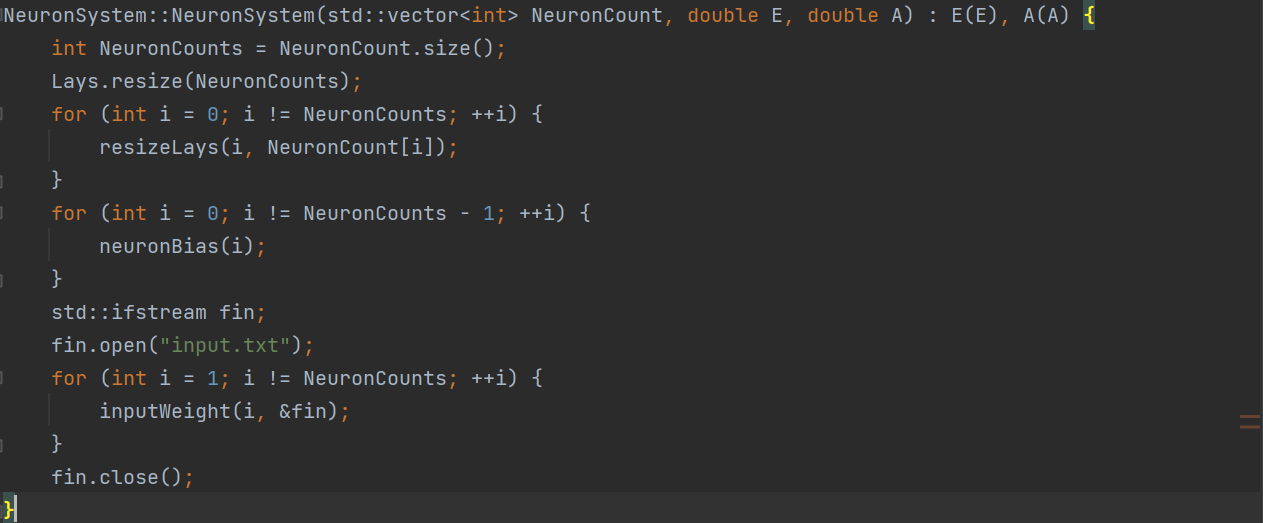
В данном сети я использую сигмоидальную (логистическую) функцию

Для начала создам класс нейрона и Нейронной сети.





Создам конструктор нейронной сети:



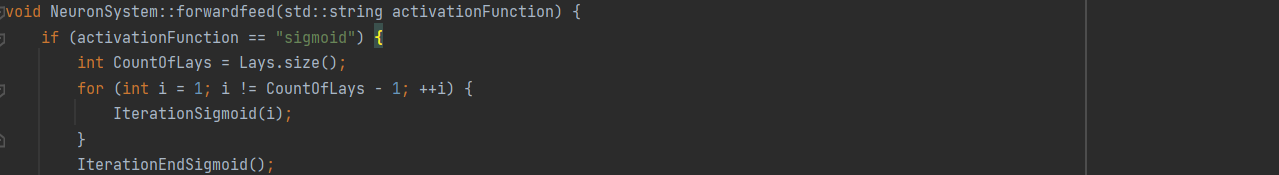
При создании нейронной сети количество слоев и нейронов будет импортироваться из вектора NeuronCount. Переменные E и А будут задавать гиперпараметры сети. В результате создается структура нейронной сети, где синапсы имеют веса из файла.

Далее создам функцию старта нейронной сети:

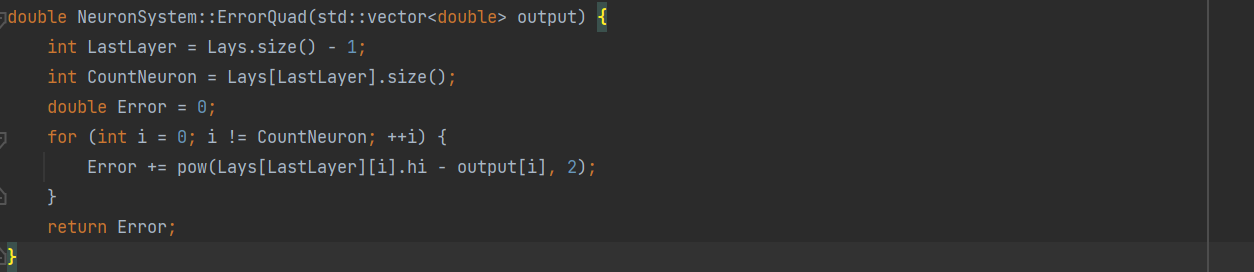


Происходит установка количества эпох и в цикле происходит полный алгоритм обучения.

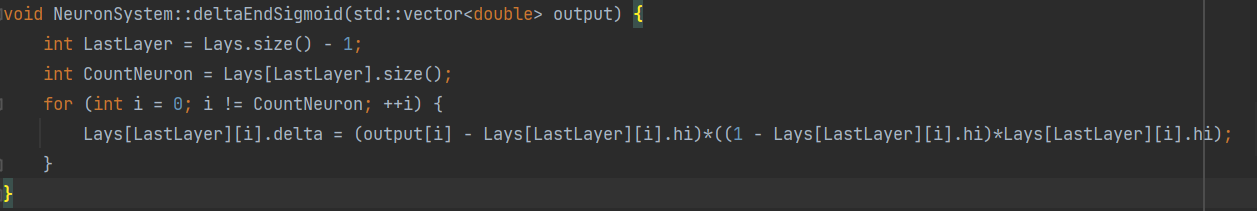
В функции forwardfeed происходит прямой проход по нейронной сети.



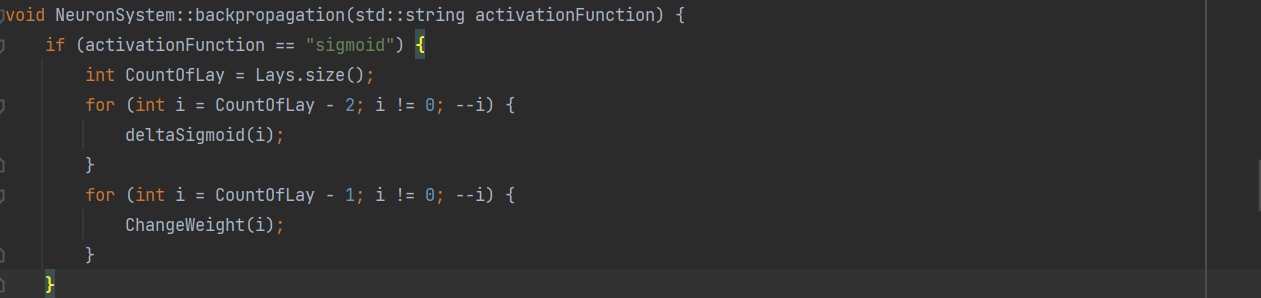
В функции ErrorQuad вычисляется среднеквадратичная ошибка:



Функция deltaEndSigmoid вычисляет значение на выходных нейронных



В функции backpropagation происходит вычисление ошибок на каждом нейроне и вычисление изменения весов.



В итоге полученная нейронная сеть может классифицировать 2 вида предмета. Например, её можно использовать для XOR сети.

Данная сеть имеет мало практического применения и служит чаще всего для ознакомления с устройством нейронной сети.

# **Написание классификатора для распознавания цифр**

Данная нейронная сеть использует 2 функции активации relu и softmax

ReLU:

Softmax:

На вход нейронной сети будет поступать изображение 28x28, а на выходном слое будет 10 нейронов, где будет находиться вероятность принадлежности к определенной цифре.

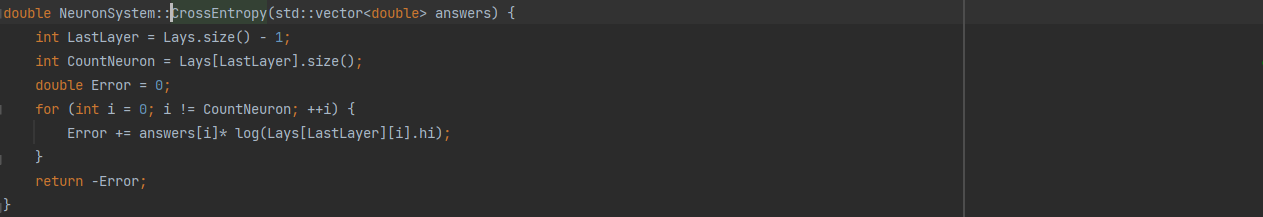
Конструктор нейронной сети останется из предыдущей сети

Напишем функцию старта нейронной сети

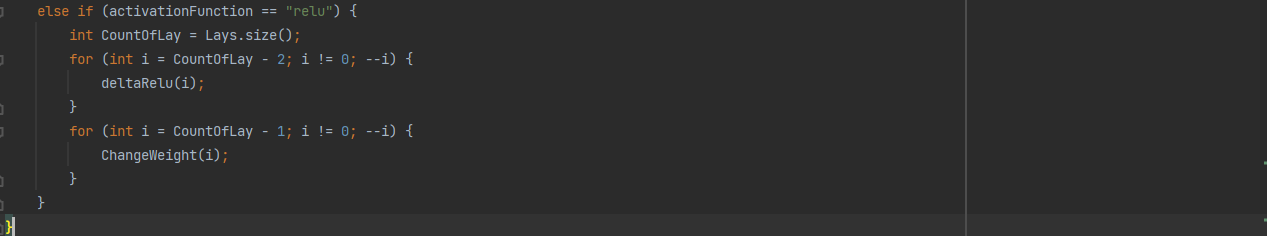


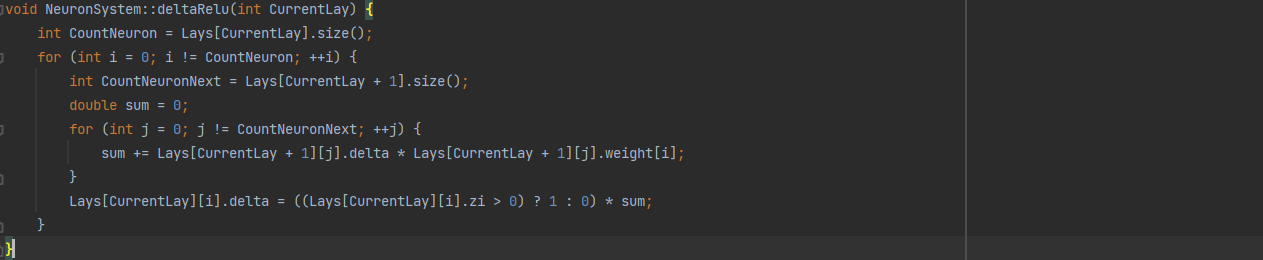
Перепишем forwardfeed для функции ReLU, на выходных нейронах вызывается функция Softmax



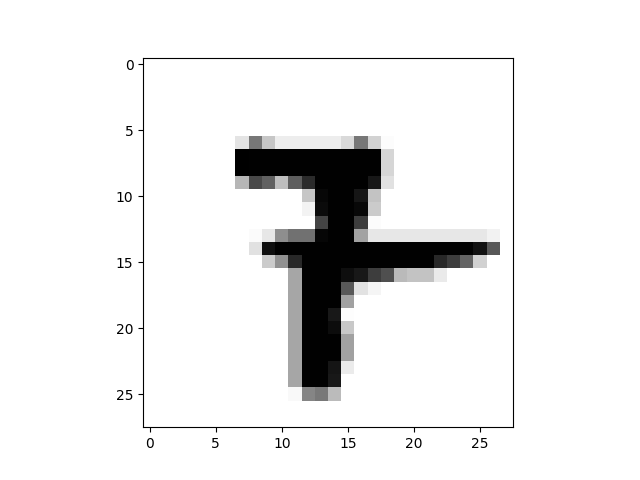
Реализую логарифмическую функцию потерь (кросс-энтропию) 

Также перепишу backpropagation и напишу deltaRelu





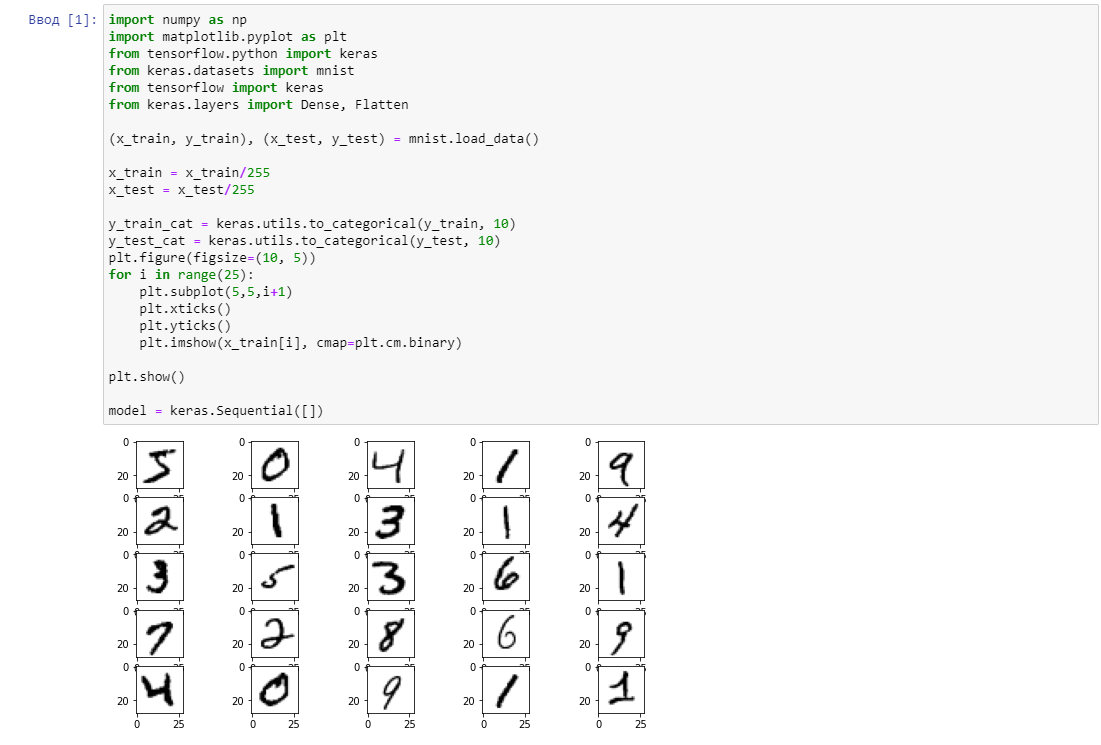
Для обучения нейронной сети используем dataset mnist содержащий изображения рукописных цифр.

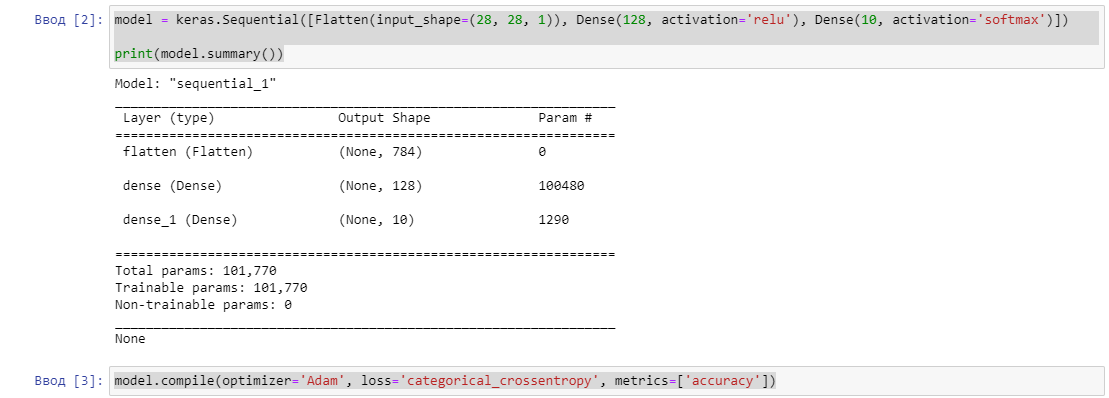


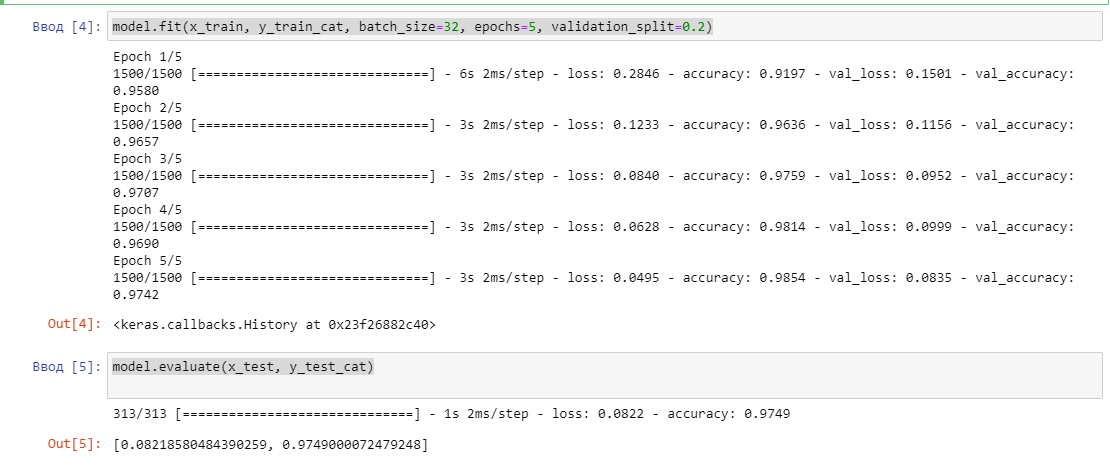
Погрешность данной нейронной сети составляет примерно 97%.

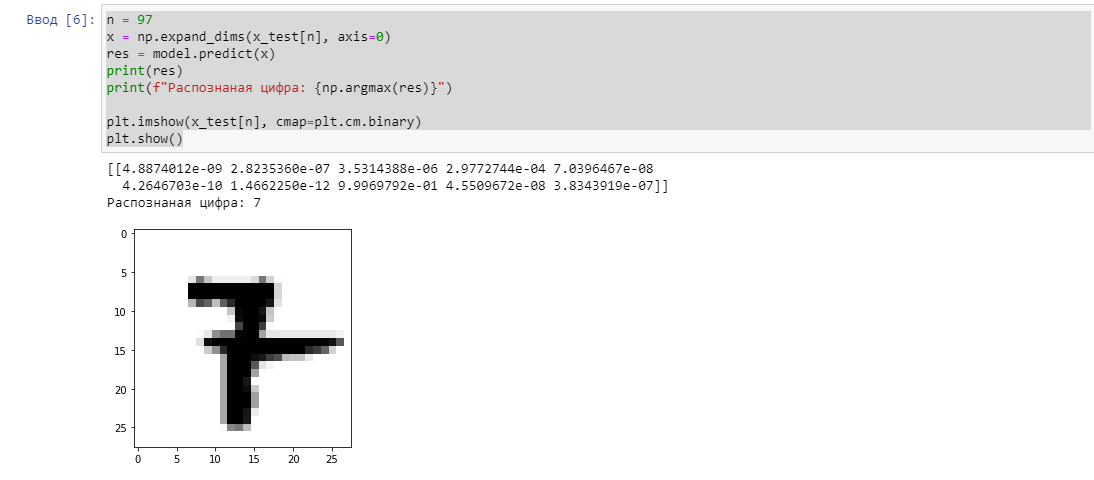
# **Написание нейронной сети на Python с использованием библиотеки Keras**

Реализуем данную нейронную сеть на языке Python, используя библиотеку Keras. Напишу программу используя Jupyter Notebook









Точность также составила примерно 97%

# **Итоги**

Данная работа была очень интересной, благодаря ей я познакомился с основами нейронных сетей, реализовал простейшие нейронные сети, также я познакомился с библиотекой keras для разработки нейронных сетей. Как можно заметить, скорость разработки НС намного выше при использовании библиотек. Работа имеет пути дальнейшего развития, например, создание сверточной нейронной сети.

# **Список литературы:**

1. <https://habr.com/ru/post/312450/>
2. <https://ru.wikipedia.org/wiki/%D0%9D%D0%B5%D0%B9%D1%80%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B5%D1%82%D1%8C>
3. <https://basegroup.ru/community/articles/math>
4. <https://www.monographies.ru/ru/book/section?id=2465>
5. <http://neuralnetworksanddeeplearning.com/>
6. <https://www.youtube.com/watch?v=5031rWXgdYo>

# **Ссылки**

1. https://github.com/KiRiLLBEL/Projects