Моделирование аффинных преобразований над признаковым пространством

Подготовил: Клинцов С. М.

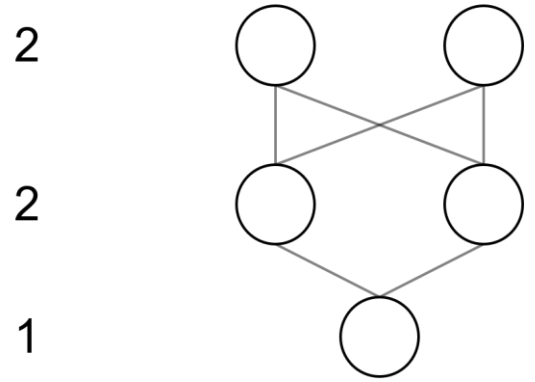
Целью данной работы является исследование возможности применения аффинных преобразований к признаковым пространствам обученной нейронной сети и разработка на этой основе алгоритма, позволяющего адаптировать веса модели к изменённым данным без изменения её выходной точности.

Задачи работы:

- провести теоретический анализ структуры признаковых пространств глубоких нейронных сетей и свойств аффинных преобразований;
- сформулировать строгие математические условия, при которых аффинное преобразование, применённое к весам одного или нескольких слоёв сети, не изменяет итоговое предсказание для измененных данных;
- разработать и формализовать алгоритм применения аффинных преобразований к весам полносвязных слоёв предобученной модели;
- реализовать спроектированный алгоритм в виде программы;
- экспериментально проверить корректность работы алгоритма.

Средства разработки

Разработка модели



Архитектура модели

Модель “Треугольник”

Модель решает бинарную задачу классификации: определение принадлежности точки с координатами (x, y) заданной геометрической области - треугольнику.

Визуализация модели

В общем случае получение значения активации нейрона на каждом слое полносвязной сети можно описать следующим уравнением:

$$\sigma(w_1 a_1 + w_2 a_2 + \cdots + w_n a_n + b) = a \text{ , где}$$

σ – функция активации;
w_n – вес n-го нейрона прошлого слоя;
a_n– активация нейрона прошлого слоя;
b – смещение;
a – новое значение активации нейрона.

Вспомним как выглядит уравнение линии на двумерной плоскости:

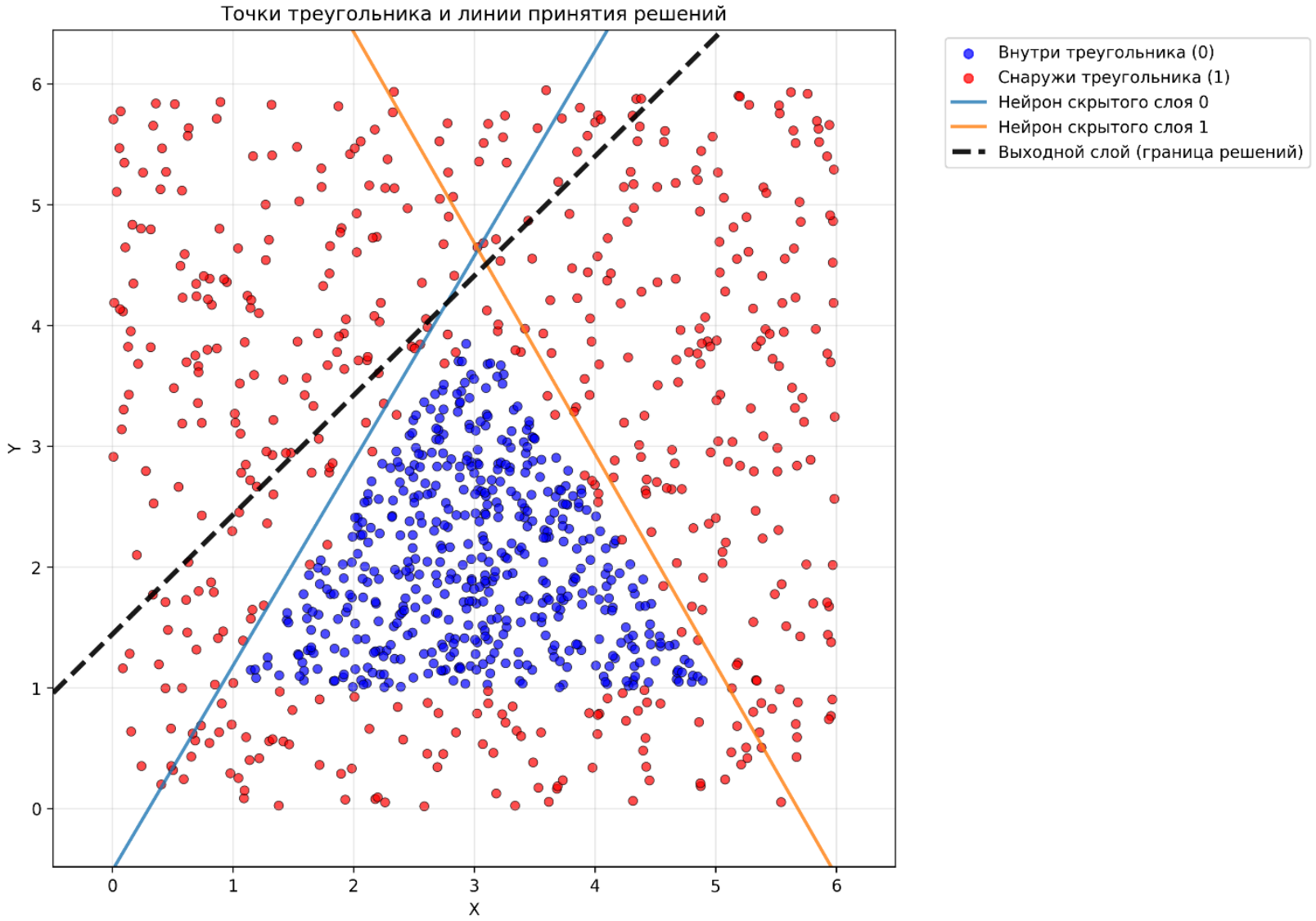
$$ax + by + c = 0$$

Преобразуем левое уравнение к уравнению линии:

$$w_1 a_1 + w_2 a_2 + b$$

По этому уравнению мы и сможем построить линии сети, ограничивающие область решений.

Разработка модели



Разработка модели

Нормали линий нейронов

Вектор n = $[w_1, w_2]$ является нормалью (перпендикуляром) к линии $w_1a_1 + w_2a_2 + b$

Зная направление нормали линии нейрона, мы сможем узнать, где находиться точка, передавая значения её координат.

Если результат уравнения $w_1x + w_2y + b > 0$, то нормаль указывает в сторону искомой области. Если $w_1x + w_2y + b < 0$, то область находиться в противоположном направлении от неё.

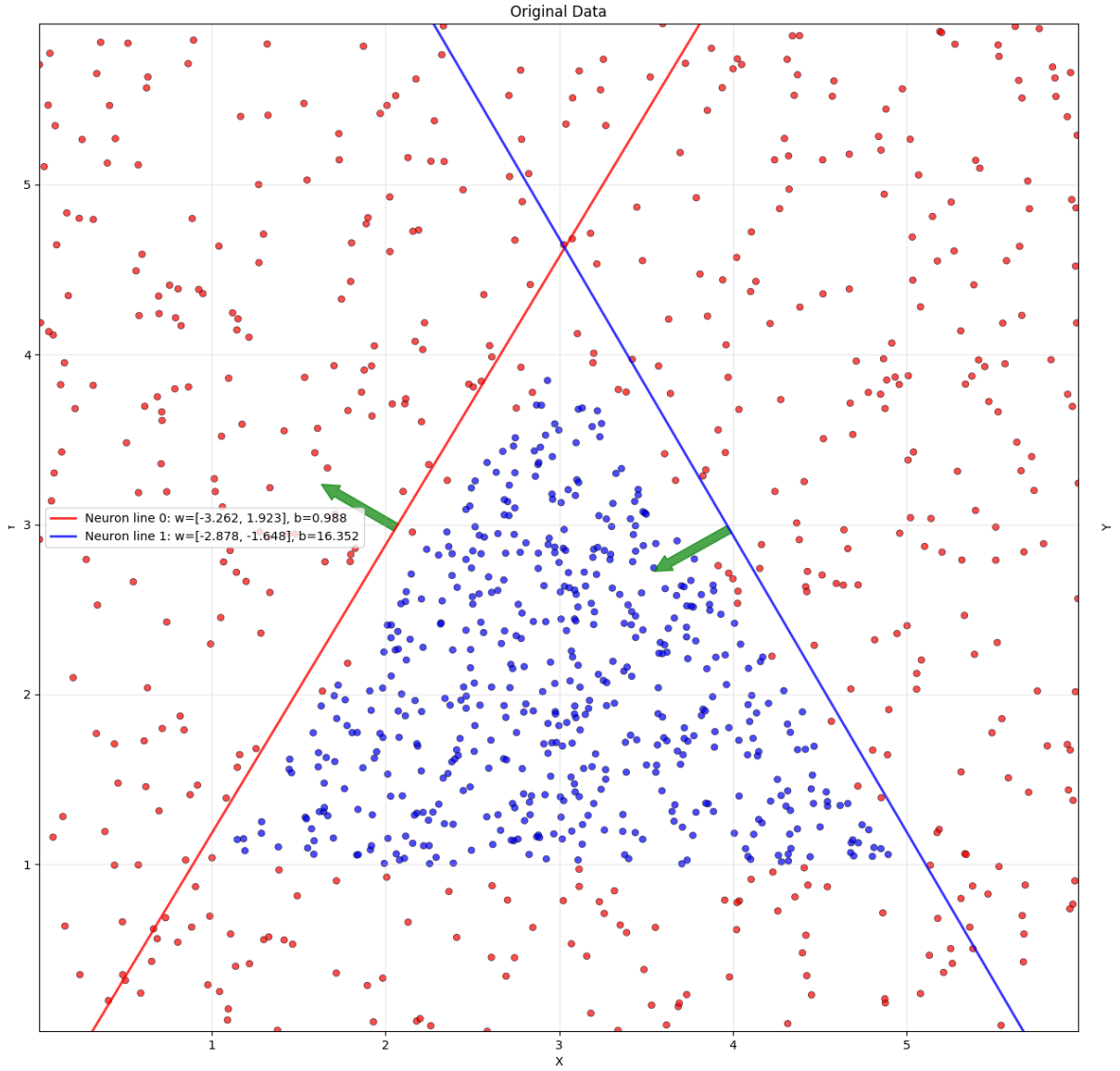
Возьмём точку с $x = 3$, $y = 3$

Для линии 0 (красная) это будет:

```
-3.26*3.0 + 1.92*3.0 + 0.98 = -3.04
```

Для линии 1 (синяя) это будет:

```
-2.87*3.0 - 1.64*3.0 + 16.35 = 2.82
```



Обновлённое отображение модели “Треугольник” с нормалями

Разработка модели

Знакомство с аффинными преобразованиями

Аффинное преобразование (от лат. affinis «соприкасающийся, близкий, смежный») — отображение плоскости или пространства в себя, при котором параллельные прямые переходят в параллельные прямые, пересекающиеся — в пересекающиеся, скрещивающиеся — в скрещивающиеся.

«Отображение в себя» означает, что если мы находились в пространстве Rn, то после образования мы должны остаться в нем же. Например: если мы применили какое-то преобразование к прямоугольнику и получили параллелепипед, то мы вышли из R2 в R3. А вот если из прямоугольника у нас получился другой прямоугольник, то все хорошо, мы отобразили исходное пространство в себя.

Каждое преобразование представляет из себя матрицу n * n.

Применение аффинного преобразования к вектору можно описать формулой:

$$M * v, \text{ где}$$

M - аффинная матрица, а v - n-мерный вектор

Разработка модели

Для дальнейшей реализации было выбрано 3 аффинных преобразования: поворот, растяжение/сжатие и сдвиг

$$\begin{pmatrix} \cos(a) & \sin(a) \\ -\sin(a) & \cos(a) \end{pmatrix}$$

поворот

$$\begin{pmatrix} k_x & 0 \\ 0 & k_y \end{pmatrix}$$

растяжения/сжатия

$$\begin{pmatrix} 0 & k_{xy} \\ k_{yx} & 0 \end{pmatrix}$$

расширение

Разработка модели

$$\begin{pmatrix} \cos(a) & \sin(a) \\ -\sin(a) & \cos(a) \end{pmatrix}$$

поворот

$$\begin{pmatrix} k_x & 0 \\ 0 & k_y \end{pmatrix}$$

растяжения/сжатия

нужно инвертировать !

$$\begin{pmatrix} 0 & k_{xy} \\ k_{yx} & 0 \end{pmatrix}$$

расширение

нужно инвертировать !

Простая аналогия

При повороте картинки на 30 градусов против часовой стрелки, требует от наблюдателя “повернуть” свой взгляд также на 30 градусов против часовой, чтобы увидеть такую же картинку.
Но, например, при растяжении изображения в 5 раз, если наблюдатель также “увеличит” взгляд в 5 раз (например, посмотрит через увеличительное стекло), то он просто ничего не увидит, так как картинка увеличиться уже в 25 раз. В данном преобразовании, взгляд наблюдателя нужно пропорционально “уменьшить” в 5 раз, чтобы увидеть ту же картинку.

Разработка модели

Обобщение матриц до n-мерного случая

```
double[][] matrix =
createIdentityMatrix(dimensions);

matrix[axis][axis] = scaleFactor;
```

растяжения/сжатия

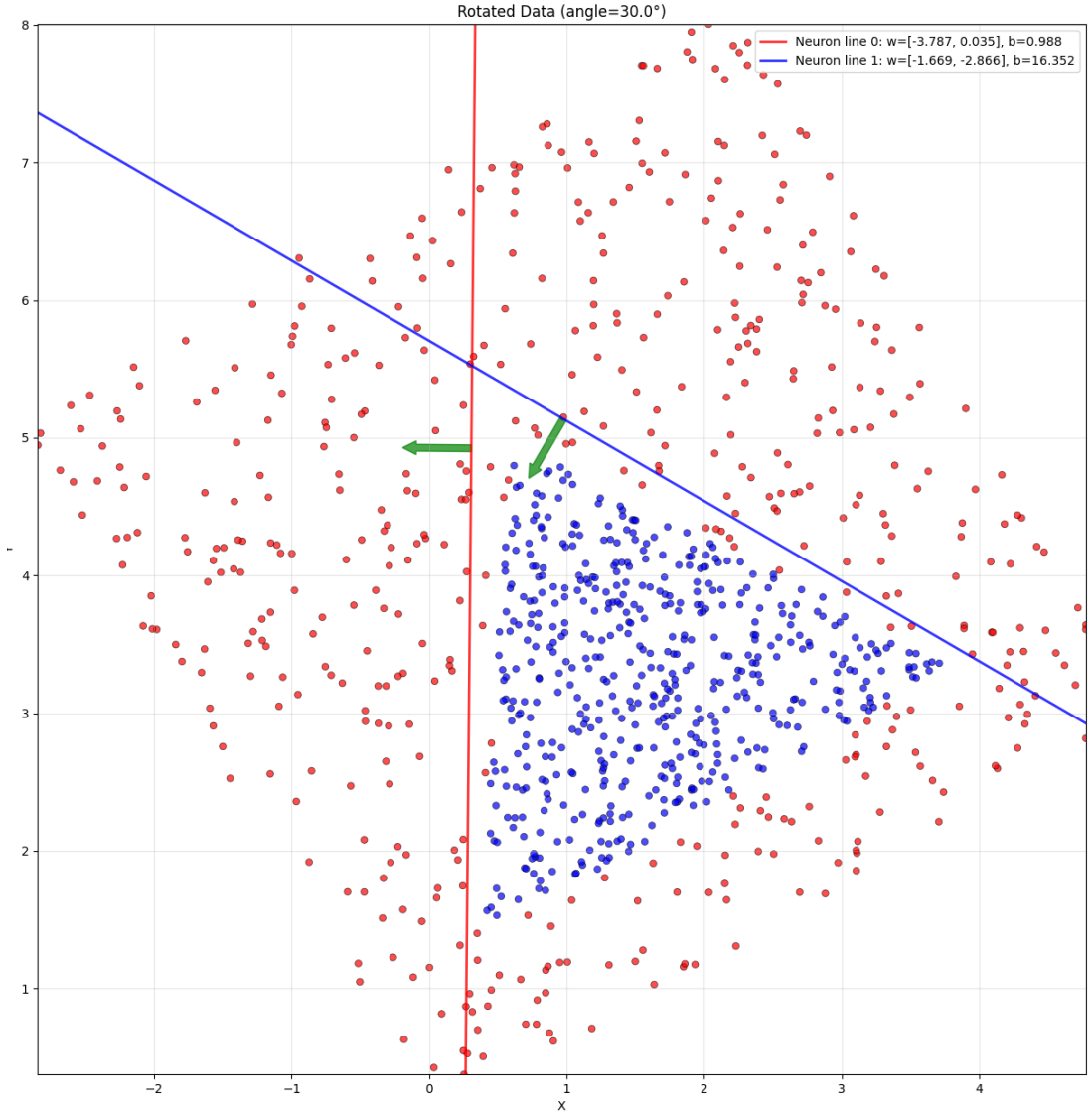
```
double[][] matrix =
createIdentityMatrix(dimensions);

matrix[axis1][axis2] = shear;
matrix[axis2][axis1] = shear;
```

расширение

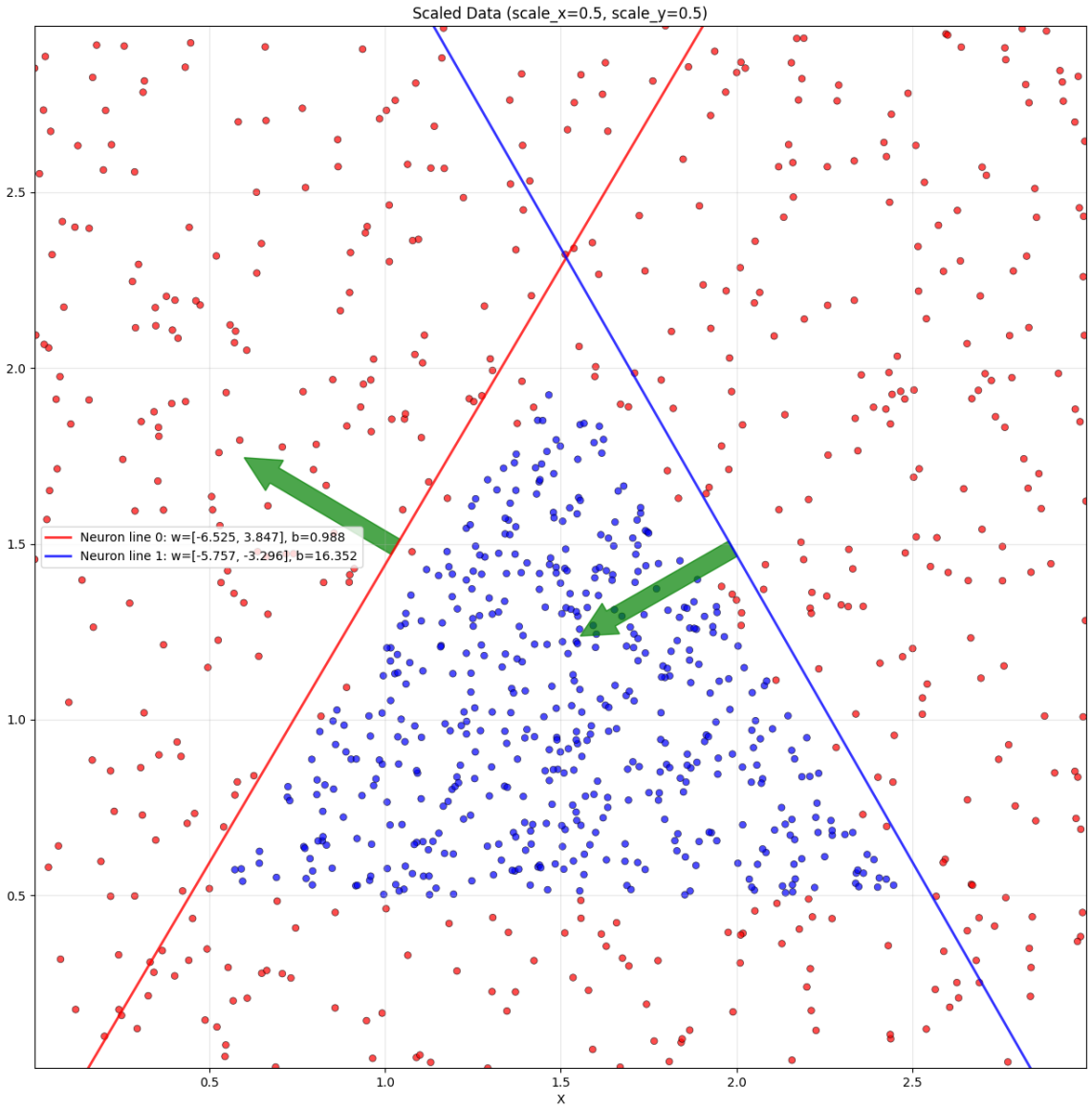
```
double[][] matrix =
createIdentityMatrix(dimensions);

matrix[axis1][axis1] = cos;
matrix[axis1][axis2] = -sin;
matrix[axis2][axis1] = sin;
matrix[axis2][axis2] = cos;
```

Поворот осуществляется одновременным применением растяжения и сжатия, используя для этого тригонометрию

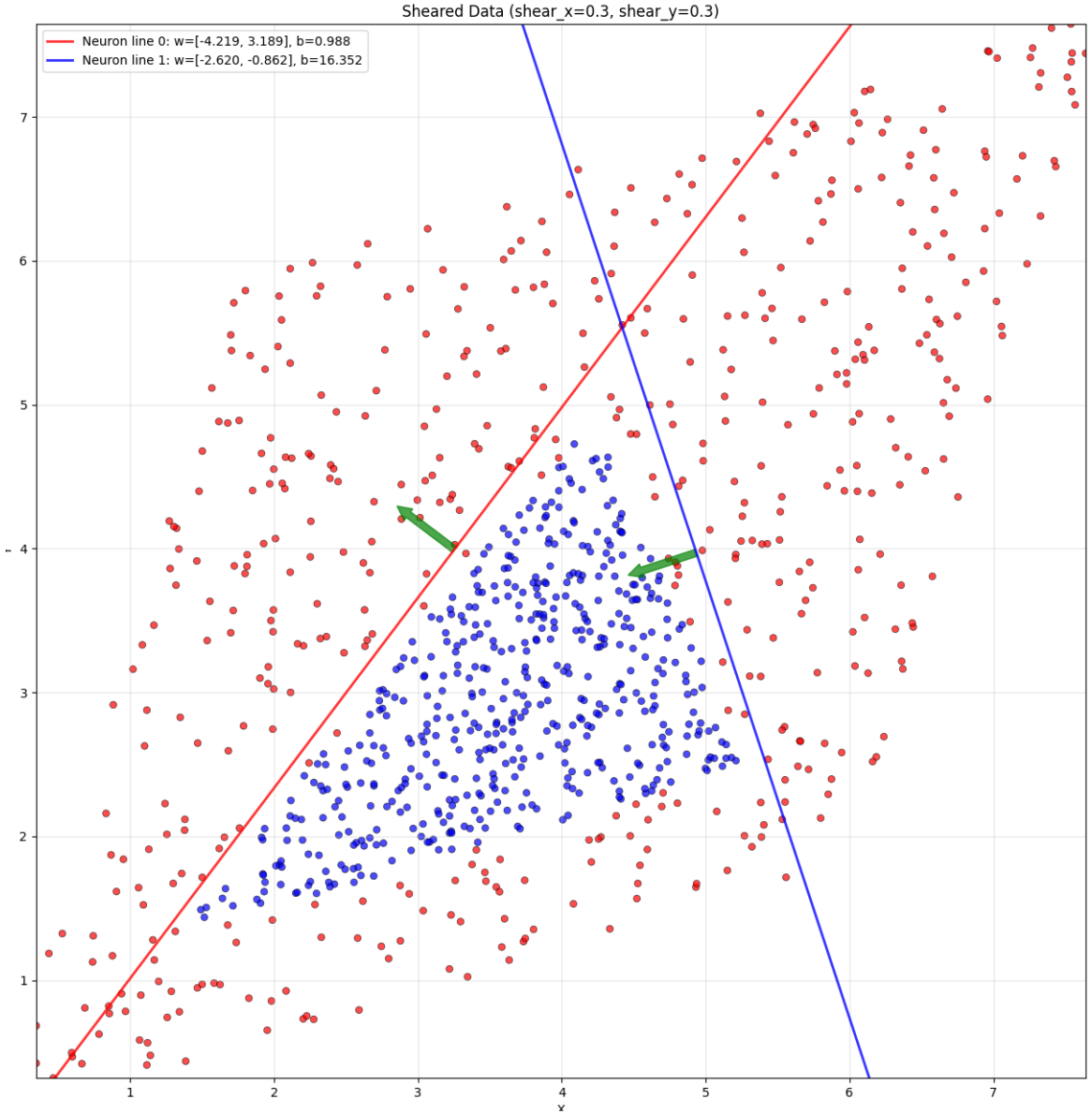
Разработка модели



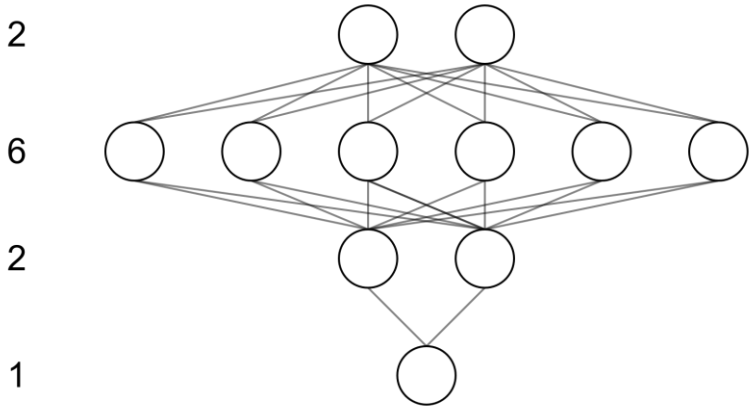
Разработка модели



Разработка модели



Разработка модели



Архитектура модели

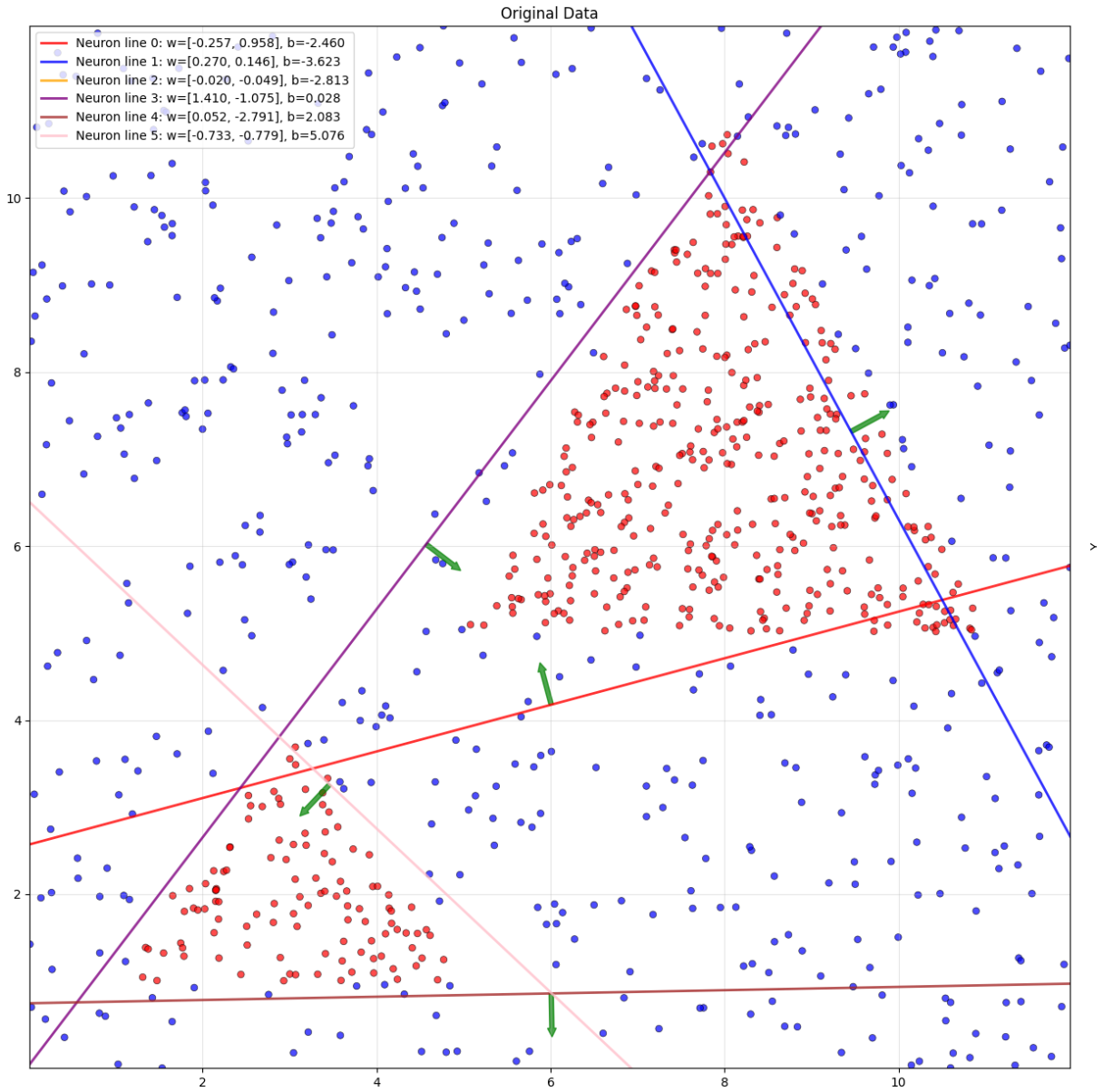
Модель “Два Треугольник”

После проведения всех экспериментов с преобразованиями на однослойной модели “Треугольник”, нужно выяснить, будет ли всё работать также хорошо и для многослойных моделей.

Аффинные преобразования будут применяться только к первому скрытому слою.
Или можно сказать иначе, только к слою, который непосредственно работает с входными данными.

Каждый полносвязный слой в нейронной сети выполняет преобразование по формуле $\sigma(w_1a_1 + w_2a_2 + \cdots + w_na_n + b) = a$. Из неё видно, что активация нейрона вычисляется по значению активации предыдущего нейрона, кроме первого слоя. Он работает непосредственно с исходными данными. Отсюда можно сделать вывод, что если при применении преобразования, первый скрытый слой выдаст тот же результат, как если бы этих преобразований не было, то все остальные слои автоматически сработают правильно и результат будет корректным.
Что бы это доказать, нужно сравнить активации нейронов первых скрытых слоёв до всех преобразований и после изменения весов и входных данных.

Разработка модели



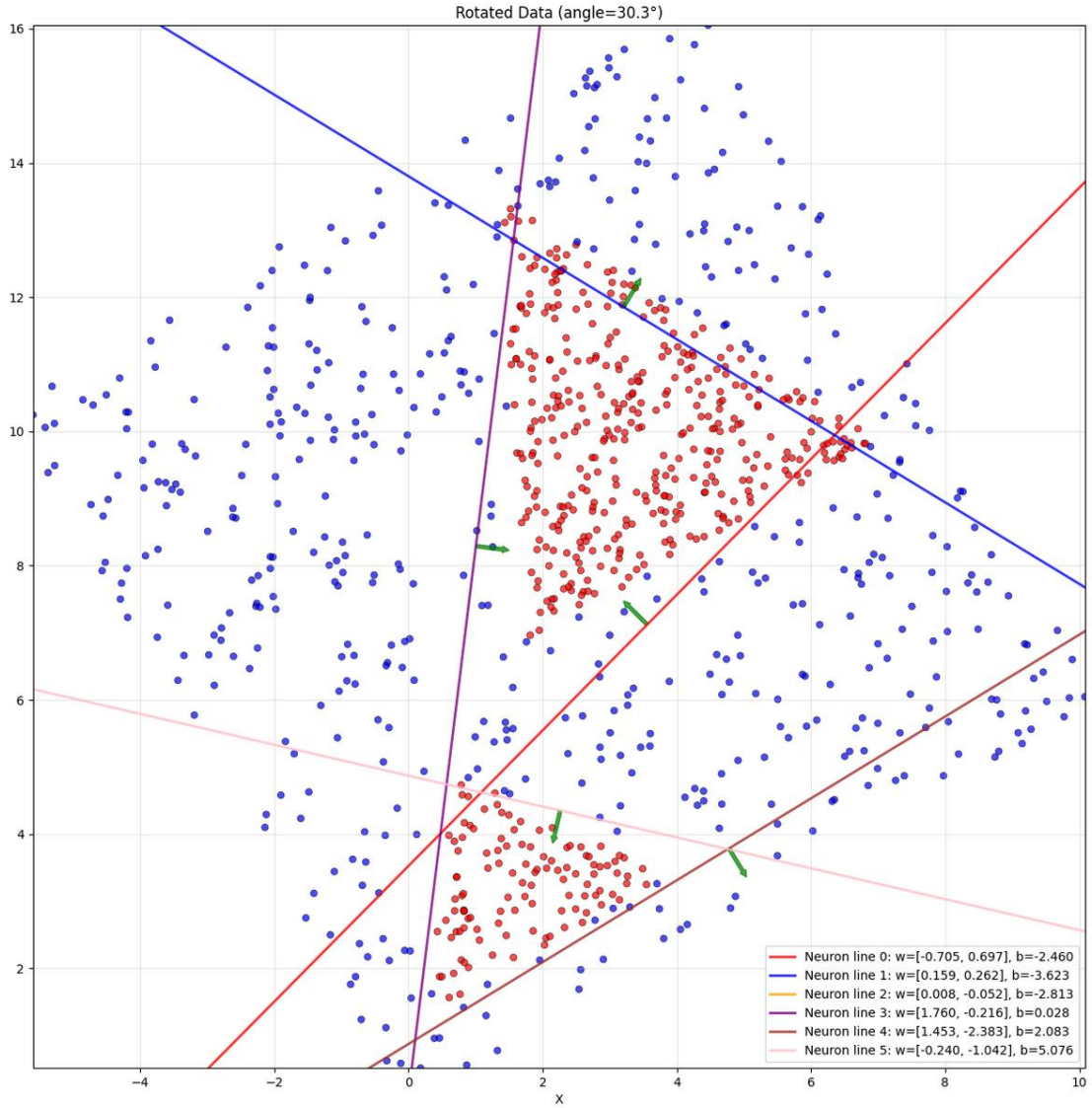
Разработка модели

Применим преобразование поворота на 30 градусов к модели, со входной точкой 3, 3:

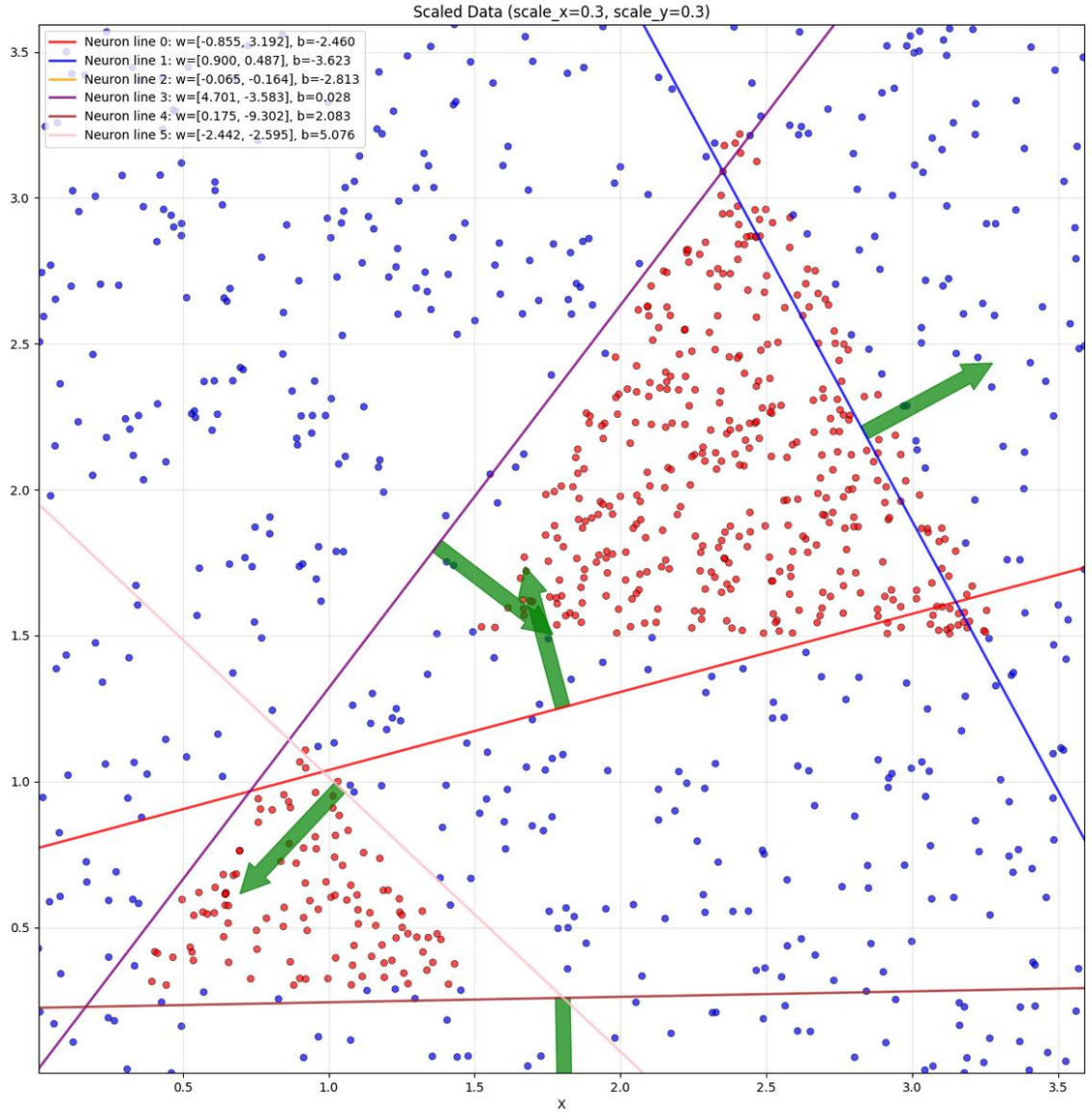
```
====== Before weight and data transformation ======
------ First hiden layer weights ------
-0.2565350830554962 0.9576932787895203
0.27009227871894836 0.1462019383907318
-0.019547274336218834 -0.04933701455593109
1.410359501838684 -1.0748653411865234
0.05246511474251747 -2.7907469272613525
-0.7326761484146118 -0.7785971164703369
------ Neuron activations ------
3.0 3.0
-0.3422616422176361 -0.9828152060508728 -0.9952481389045715 0.7757573723793 03 -0.9999905228614807 -0.49449196457862854
-0.9999547600746155 -0.7523653507232666
0.985088050365448
------ Prediction ------
Confidence: 0.985088050365448
```

```
====== After weight and data transformation ======
------ First hiden layer weights ------
0.7476058476903883 0.6512082321163086
0.24999965462541984 -0.1783956796000739
-0.05290834426082423 -0.004116314716547248
-0.3459362576915701 -1.7391887567843332
-2.4839285710086605 -1.2732320504960983
-1.0213258509573662 0.3161033286724542
------ Neuron activations ------
4.012990772626067 -1.3769186827180608
-0.3422614336013794 -0.9828152060508728 -0.9952481389045715 0.7757572531700134 -0.999990522861 4807 0.4944923520088196
-0.9999547600746155 -0.75236576795578
0.9850881695747375
------ Prediction ------
Confidence: 0.9850881695747375
```

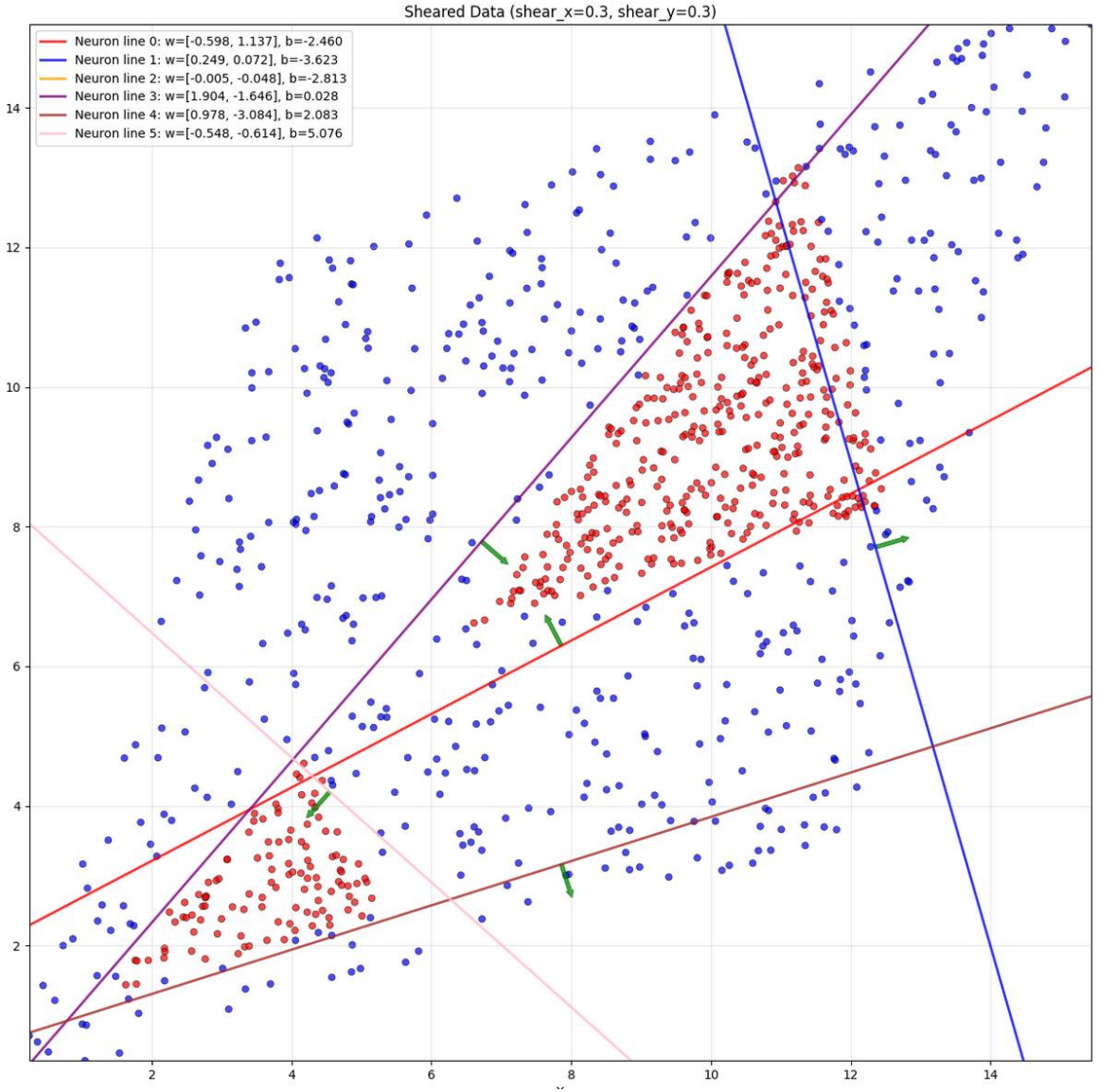
Разработка модели



Разработка модели



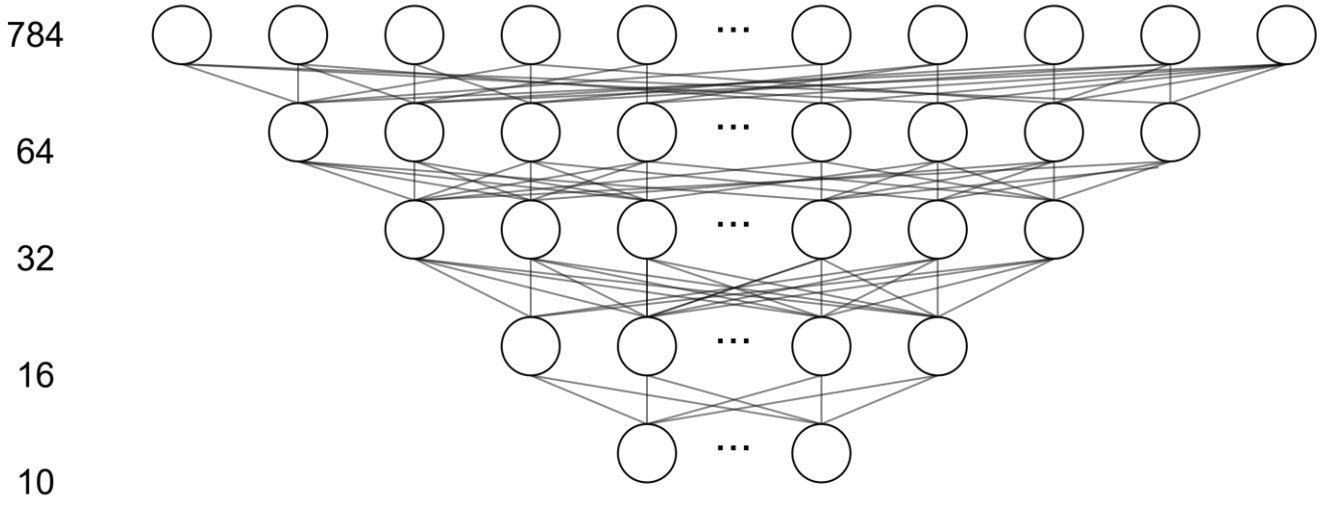
Разработка модели



Разработка модели

Модель “MNIST”

Заключительной экспериментальной моделью послужит нейронная сеть на базе датасета чисел “MNIST”, где каждая цифра от 0-9 представлена изображение 28*28 пикселей



Архитектура модели

Разработка модели

Результаты после поворота изображения цифры 3 и весов первого скрытого слоя:

```
====== Before weight and data transformation ======

------ First hiden layer weights ------

5.77050783973259 56E-39 4.703835244793152E-39 -
8.87386489726088E-38 -6.339193992912607E-41 5.253545014169277E-
39 1.5848125112127951E-40 3.264153814232014E-39 ...

...

------ Neuron activations ------

... 241.0 255.0 255.0 255.0 255.0 255.0 244.0 237.0 128.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
50.0 235.0 255.0 245.0 254.0 246.0 245.0 245.0 245.0 248.0 ...

232.97018432617188 627.9869995117188 0.0 0.0
1112.656982421875 629.7344970703125 0.0 0.0 261.0110168457031
879.9109497070312 278.156005859375 209.4010 1623535156 ...

...

0.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0

------ Prediction ------

Predicted digit: 3

Confidence: 1.0
```

```
====== After weight and data transformation ======

------ First hiden layer weights ------

1.839428899626901E-4 8.9961941361909E-5 1.0014484522300589E-4
1.1148036461767837E-4 1.240989655295359E-4 1.381458815488908E-4
1.537827854364992E-4 ...

...

------ Neuron activations ------

... 6.764818237218361E-8 3.308506143967021E-8
3.6830000630381116E-8 4.099883413870403E-8 4.56395430888585E-8
5.0805539648099246E-8 5.655628177322188E-8 6.295795754098908E-8 ...

232.97021484375 627.9871826171875 0.0 0.0 1112.6568603515625
629.7346191406 25 0.0 0.0 261.0109558105469 879.9109497070312
278.15557312011719 209.4010467529297 ...

...

0.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0

------ Prediction ------

Predicted digit: 3

Confidence: 1.0
```

Исследование работы модели

Таблица экспериментов с поворотом на 256 градусов для модели “Два треугольника”:

№	x	y	Ответ	До преобразований (prediction)	До преобразований (confidence)	После преобразования точки (prediction)	После преобразования точки (confidence)	После преобразования весов (prediction)	После преобразования весов (confidence)	После всех преобразований (prediction)	После всех преобразований (confidence)	Совпадение после преобразования точки	Изменение уверенности после преобразования точки	Совпадение после преобразования весов	Изменение уверенности после преобразования весов	Совпадение после всех преобразований	Изменение уверенности после всех преобразований
1	0.748883	4.466553	out	out	0.013652	out	0.011325	out	0.012876	out	0.013652	ИСТИНА	-0.002327	ИСТИНА	-0.000776	ИСТИНА	0.000000
2	2.915012	7.792783	out	out	0.011009	out	0.029955	out	0.013105	out	0.011009	ИСТИНА	0.018946	ИСТИНА	0.002096	ИСТИНА	0.000000
3	10.806968	8.793428	out	out	0.100910	out	0.027732	out	0.012856	out	0.100910	ИСТИНА	-0.073178	ИСТИНА	-0.088054	ИСТИНА	0.000000
4	9.720528	5.146038	in	in	0.897826	out	0.012521	out	0.012846	in	0.897826	ЛОЖЬ	-0.885305	ЛОЖЬ	-0.884980	ИСТИНА	-0.000000
5	6.421433	5.919338	in	in	0.945953	out	0.029565	out	0.012831	in	0.945953	ЛОЖЬ	-0.916389	ЛОЖЬ	-0.933122	ИСТИНА	-0.000000
6	1.552912	1.323665	in	in	0.926742	out	0.014082	out	0.012797	in	0.926742	ЛОЖЬ	-0.912660	ЛОЖЬ	-0.913945	ИСТИНА	0.000000
7	10.725274	11.075804	out	out	0.074198	out	0.029556	out	0.012955	out	0.074198	ИСТИНА	-0.044641	ИСТИНА	-0.061243	ИСТИНА	0.000000
8	9.355639	5.716580	in	in	0.977043	out	0.019840	out	0.012838	in	0.977043	ЛОЖЬ	-0.957203	ЛОЖЬ	-0.964205	ИСТИНА	0.000000
9	3.381909	3.165688	in	in	0.953030	out	0.029044	out	0.012811	in	0.953030	ЛОЖЬ	-0.923986	ЛОЖЬ	-0.940219	ИСТИНА	0.000000
10	8.025612	8.196489	in	in	0.984033	out	0.029589	out	0.012871	in	0.984033	ЛОЖЬ	-0.954444	ЛОЖЬ	-0.971161	ИСТИНА	0.000000

Исследование работы модели

Таблица экспериментов с поворотом на 256 градусов для модели “MNIST”:

№	Цифра	До преобразований (prediction)	До преобразований (confidence)	После преобразования изображения (prediction)	После преобразования изображения (confidence)	После преобразования весов (prediction)	После преобразования весов (confidence)	После всех преобразований (prediction)	После всех преобразований (confidence)	Совпадение после преобразования изображения	Изменение уверенности после преобразования изображения	Совпадение после преобразования весов	Изменение уверенности после преобразования весов	Совпадение после всех преобразований	Изменение уверенности после всех преобразований2
1	0	0	1.000000	0	1.000000	0	1.000000	0	1.000000	ИСТИНА	0.000000	ИСТИНА	0.000000	ИСТИНА	0.000000
2	1	1	1.000000	7	1.000000	4	1.000000	1	1.000000	ЛОЖЬ	0.000000	ЛОЖЬ	0.000000	ИСТИНА	0.000000
3	2	2	1.000000	2	1.000000	1	1.000000	2	1.000000	ИСТИНА	0.000000	ЛОЖЬ	0.000000	ИСТИНА	0.000000
4	3	3	1.000000	2	1.000000	0	1.000000	3	1.000000	ЛОЖЬ	0.000000	ЛОЖЬ	0.000000	ИСТИНА	0.000000
5	4	4	1.000000	3	1.000000	2	1.000000	4	1.000000	ЛОЖЬ	0.000000	ЛОЖЬ	0.000000	ИСТИНА	0.000000
6	5	5	1.000000	6	1.000000	5	1.000000	5	1.000000	ЛОЖЬ	0.000000	ИСТИНА	0.000000	ИСТИНА	0.000000
7	6	6	1.000000	2	1.000000	2	1.000000	6	1.000000	ЛОЖЬ	0.000000	ЛОЖЬ	0.000000	ИСТИНА	0.000000
8	7	7	1.000000	1	1.000000	7	1.000000	7	1.000000	ЛОЖЬ	0.000000	ИСТИНА	0.000000	ИСТИНА	0.000000
9	8	8	1.000000	0	1.000000	5	1.000000	8	1.000000	ЛОЖЬ	0.000000	ЛОЖЬ	0.000000	ИСТИНА	0.000000
10	9	9	1.000000	0	1.000000	0	1.000000	9	1.000000	ЛОЖЬ	0.000000	ЛОЖЬ	0.000000	ИСТИНА	0.000000

Заключение

Таким образом, работа продемонстрировала принципиальную возможность и практическую реализуемость безопасного преобразования весов обученных нейронных сетей через аффинные преобразования. Полученные результаты подтверждают, что нейронные сети, несмотря на свою сложность, обладают определённой алгебраической структурой, позволяющей осуществлять контролируемые манипуляции с их параметрами при сохранении функциональной целостности.