Обучение глубинных нейронных сетей на основе методов байесовской фильтрации

Сергей Павлов

Институт проблем передачи информации РАН
Московский физико-технический институт

2017

Оглавление

Постановка задачи

- Измеримое пространство $(\Omega, \mathcal{F})$ с фильтрацией $\mathcal{F} := (\mathcal{F}_r)_{r=0,1,2,\dots T}$, $T \in N_+$
- Случайный процесс(управляющий процесс) $\mathbf{a}$: $\Omega \times \{0, \dots, T-1\} \to A$, где $(A, \mathcal{B})$ - измеримое пространство
- $\mathbf{a} = (a_0, a_1, \dots, a_{T-1}) \in \mathcal{A}-$допустимое множество значений управлений; задавшись таким значением процесса $\mathbf{a}$, определим
- Марковский процесс $X_r : (\Omega, \mathcal{F}, P^{\mathbf{a}}) \to (S, \mathcal{S})$, $X_0 = x_0$ п.н., с

$$\mathbf{P}^{\mathbf{a}}(X_{r+1} \in dy | X_r = x) = P^{a_r}(x, dy), \quad 0 \le r < T.$$

Пусть $\mathcal{T} \subset \{0, \ldots, T\}$. Зададимся измеримыми функциями

$$f_r : S \times A \to \mathbb{R}$$

$$g_r : S \to \mathbb{R}$$

Рассмотрим оптимизационную задачу:

$$Y_0^* := \sup_{\mathbf{a} \in \mathcal{A}, \tau \in \mathcal{T}} E^{\mathbf{a}} \left[\sum_{r=0}^{\tau-1} f_r(X_r, a_r) + g_\tau(X_\tau) \right].$$

Интерпретируем данную общую задачу в терминах обучения нейронных сетей.

Рассмотрим нейронную сеть, состоящую из T слоёв с номерами $r = 0, \ldots, T-1$. $\mathcal{T} = \{T\}$ При этом:

- X_r отвечает за значение нейронов слоя r;
- a_r соответствует паре $(\mathbf{W^r}, \mathbf{B^r})$ - матрица переходов между слоями r и $r+1$, и вектор порогов активаций слоя $r+1$;
- Значения процессов $\mathbf{W}$ и $\mathbf{B}$ имеют независимые нормальные распределения:

$$\mathbf{B^i} \sim \mathcal{N}(\mathbf{b^i}, \mathbb{I}),$$

$$\mathbf{W^i} \sim \mathcal{N}(\mathbf{w^i}, \mathbb{I})$$

- Обозначим функцию потерь сети $L'(X_{T-1}, Y)$;
- $g_T(X_{T-1}) := -L'(X_{T-1}, Y) = L(X_{T-1}, Y)$
- $f_r = 0, r = 0, \ldots, T-1$

Оптимизационная задача переформулируется таким образом:

$$Y_0^* := \sup_{\mathbf{a}\in\mathcal{A}} E^{\mathbf{a}}\left[L(X_{T-1}, Y)\right]$$

Напомним, что здесь L есть функция потерь сети, взятая с обратным знаком($L < 0$).
Обозначим A_r допустимое множество управлений $\mathbf{a}$:

$$\Omega \times \{r, \ldots, T-1\} \to A$$

Определим случайные величины:

$$Y_r^* := \sup_{\mathbf{a}\in\mathcal{A}_r} E^{\mathbf{a}}\left[L(X_{T-1}, Y)|X_r\right], \quad 0 \le r \le T.$$

Зададим функции:

$$h_r^*(x) = \sup_{\mathbf{a}\in\mathcal{A}_r} E^{\mathbf{a}}\left[L(X_{T-1}, Y)|X_r = x\right], \quad 0 \le r \le T$$

Решение задачи

В соответствии с принципом Беллмана:

$$Y_r^* = h_r^*(X_r).$$

Если вычислена зависимость $h_{r+1}^*(x'), \forall x'$, то задача поиска зависимости $h_r^*(x)$ сводится к оптимизации по единственному параметру a_r:

$$h_r^*(x) = \sup_{a_r} \int P^{a_r}(x, dy) h_{r+1}^*(y)$$

Алгоритм обучения для фиксированного Y

- Значения X_0 и $Y_0(X_0)$ берутся из тренировочного набора данных;
- На шаге $i \in [1, \ldots, T-1]$ оптимизационная задача решается перебором по следующей схеме:
  - Сгенерировать $\{X^j_{T-i-1}\}_{j=0}^N$ из гиперкуба $[0,1]^{dim(X_{T-i-\mathbf{1}})}$;
  - Для каждого из X^j_{T-i-1} генерировать по M значений параметров $\mathbf{w_{jk}^{T-i-1}}, \mathbf{b_{jk}^{T-i-1}}$, $k \in \{1, \ldots, M\}$ из гиперкуба с базой [-2,2];
  - Для каждой пары параметров $\mathbf{w_{jk}^{T-i-1}}, \mathbf{b_{jk}^{T-i-1}}$ получить K реализаций случайных величин: $B^{T-i-1}_{jkl}, W^{T-i-1}_{jkl}, l = 1, \ldots, K$
  - $$h^{*,emp}_{T-i-1,jk}(X^j_{T-i-1}) := \frac{1}{K}\sum_{l=1}^{K} h^*_{T-i}(\sigma(X^j_{T-i-1}W^{T-i-1}_{jkl} + B^{T-i-1}_{jkl}))$$
  - Выбрать пару параметров $(w_j^{*,T-i-1}, b_j^{*,T-i-1}) = (w^{T-i-1}_{jk^*}, b^{T-i-1}_{jk^*})$, $k^* := \text{argmax}_k\, h^{*,emp}_{T-i-1,jk}(X^j_{T-i-1})$;

- Построить регрессионную модель M^Y_{T-i-1} зависимости $(w_j^{*,T-i-1}, b_j^{*,T-i-1})$ от значения X^j_{T-i-1};

Результат обучения: $M^Y = (M^Y_0, \ldots, M^Y_{T-2})$

Вычисление прогноза

Имея обучение $M^Y = (M_0^Y, \ldots, M_{T-2}^Y)$ для **фиксировнного** Y, опишем процесс построение прогноза сети.
При вычислении прогноза будем полагать случайные величины $\mathbf{W^i}, \mathbf{B^i}$ равными своим мат.ожиданиям $\mathbf{w^i(X_{i-1})}, \mathbf{b^i(X_{i-1})}$;
Чтобы сделать прогноз по заданному входу X_0 необходимо произвести цепочку вычислений:

$$\overline{X_0} := X_0 \xrightarrow{M_0^Y} \mathbf{w^0}, \mathbf{b^0} \to \overline{X}_1 := \sigma(\overline{X_0}\mathbf{w^0} + \mathbf{b^0})$$

$$\xrightarrow{M_1^Y} \mathbf{w^1}, \mathbf{b^1} \to \overline{X}_2 \ldots, \overline{X}_{T-1}$$

В зависимости от задачи(регрессия/классификация) прогнозом объявляется $\overline{X}_{T-1}$ или $\sigma(\overline{X}_{T-1})$.

Объединение обучений для различных Y

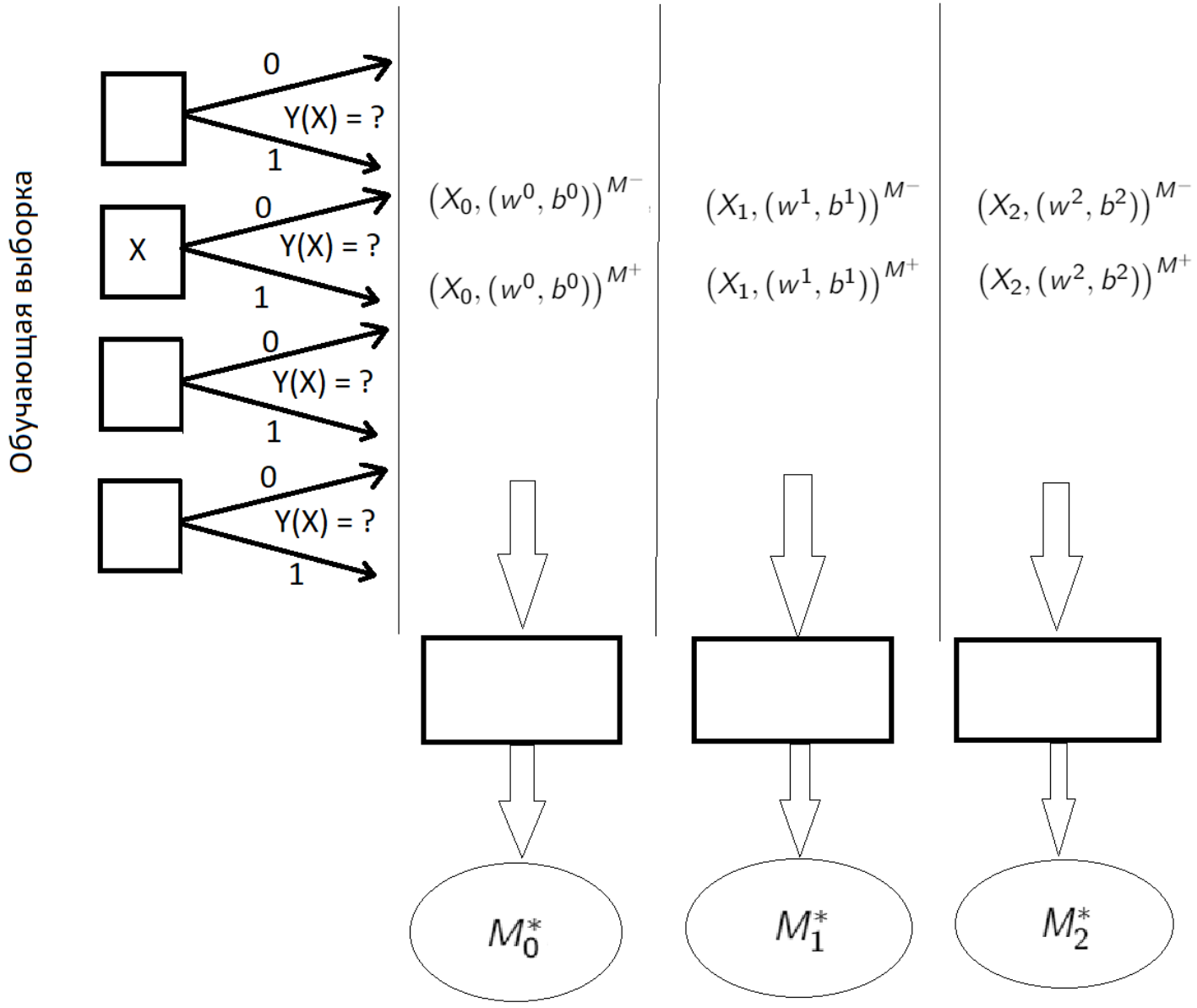
Перейдём к задаче бинарной классификации(0/1).
$\{(X_r, Y(X_r))\}_{r=1}^{Q}$ - обучающая выборка; $Y(X_r) \in \{0, 1\}$

$$\text{Обучение для } 1 : M^+ = (M_0^+, \ldots, M_{T-2}^+)$$

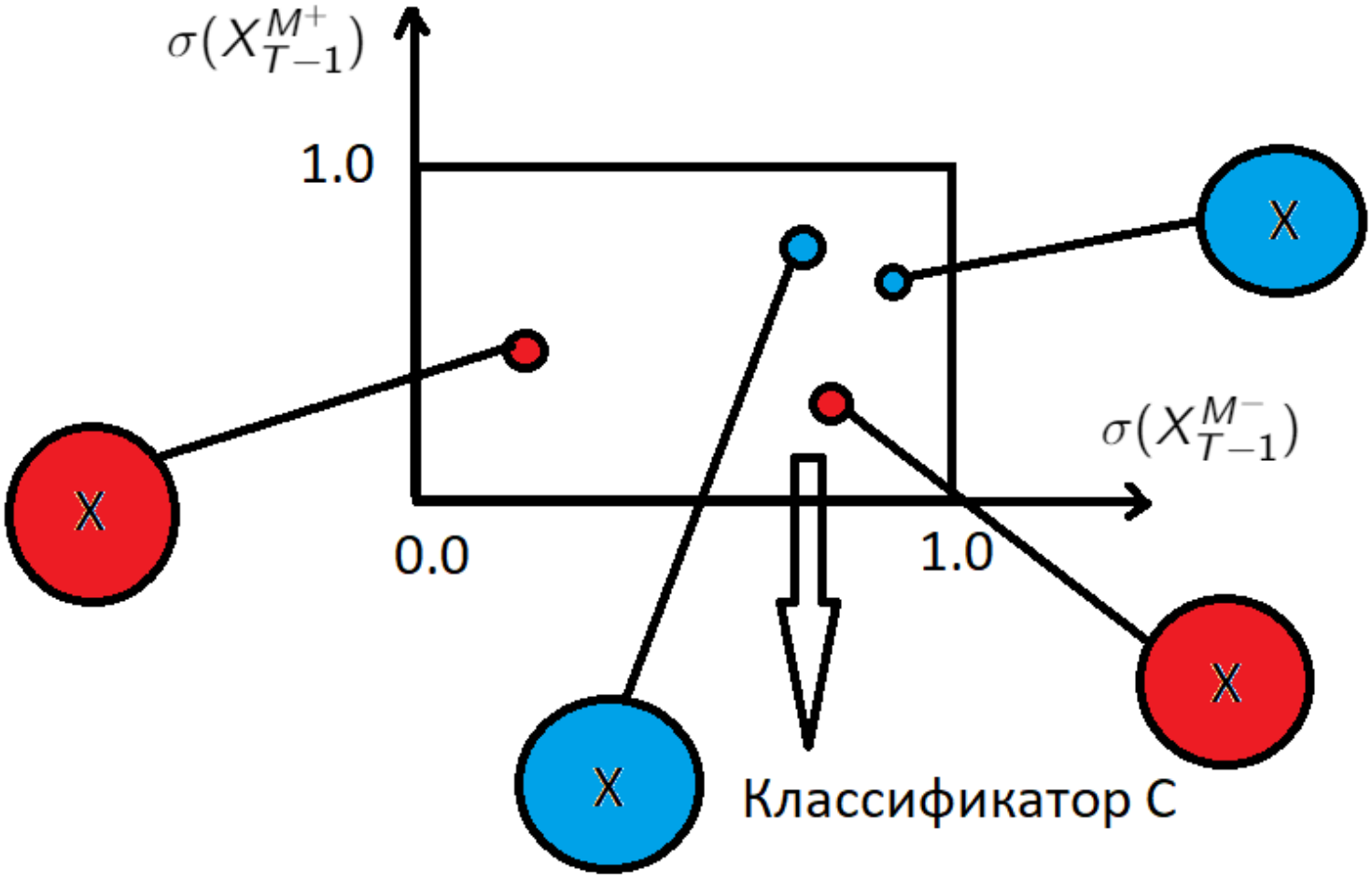
$$\text{Обучение для } 0 : M^- = (M_0^-, \ldots, M_{T-2}^-)$$

Цель: Получить обучение $M^* = (M_0^*, \ldots, M_{T-2}^*)$, отражающее зависимость $Y(X_{in})$

Подход 1


Обучающая выборка
X
0
Y(X) = ?
1
0
Y(X) = ?
1
0
Y(X) = ?
1
0
Y(X) = ?
1
$(X_0, (w^0, b^0))^{M^-}$
$(X_0, (w^0, b^0))^{M^+}$
$(X_1, (w^1, b^1))^{M^-}$
$(X_1, (w^1, b^1))^{M^+}$
$(X_2, (w^2, b^2))^{M^-}$
$(X_2, (w^2, b^2))^{M^+}$
M_0^*
M_1^*
M_2^*

Подход 2



Эксперименты

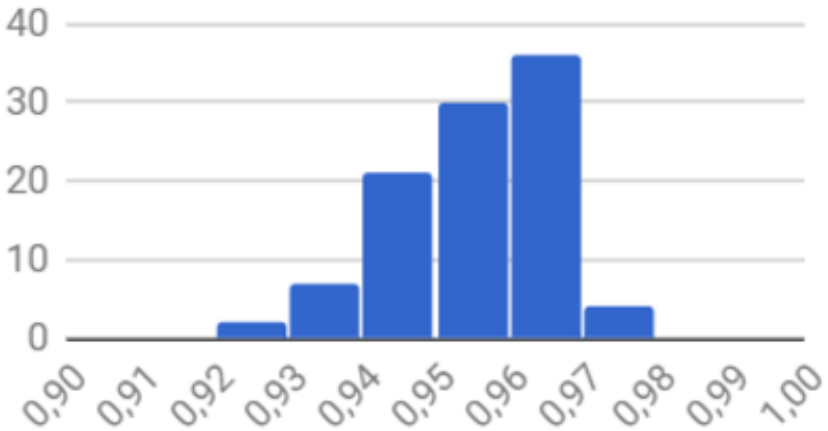
$N = 15, M = 15, K = 5$, архитектура сети: $[X, 5, 5, 5, 5, 5, 1]$.
Данные:

1. "diabetes"
2. "breast-cancer"
3. "fourclass"
4. "german.numer"
5. "heart"
6. "liver-disorders"

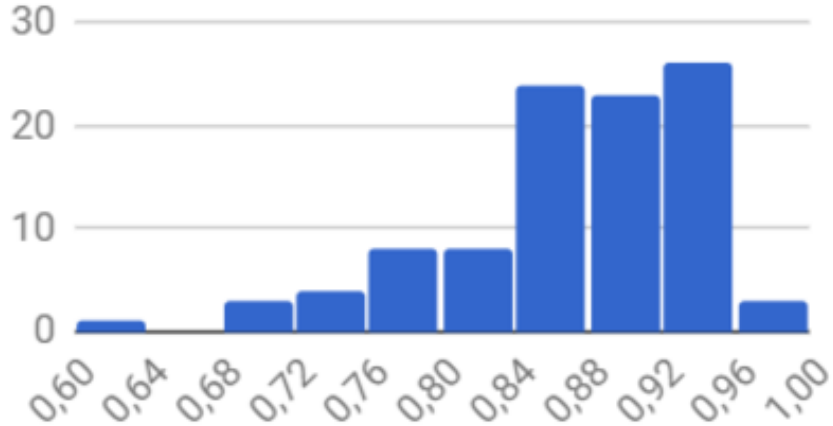
Результаты

Набор данных	Среднее(о/с)	Дисперсия(о/с)	Среднее(#1)	Дисперсия(#1)	Среднее(#2)	Дисперсия(#2)	Размер обуч. выб	Размер тест. выб
1	0,72	0,0006	0,69	0,003	0,54	0,014	400	368
2	0,96	0,0001	0,87	0,004	0,7	0,04	350	333
3	0,93	0,005	0,69	0,004	0,58	0,02	550	312
4	0,73	0,006	0,68	0,009	0,51	0,02	900	100
5	0,78	0,003	0,67	0,01	0,52	0,02	170	100
6	0,68	0,004	0,63	0,01	0,58	0,02	100	45

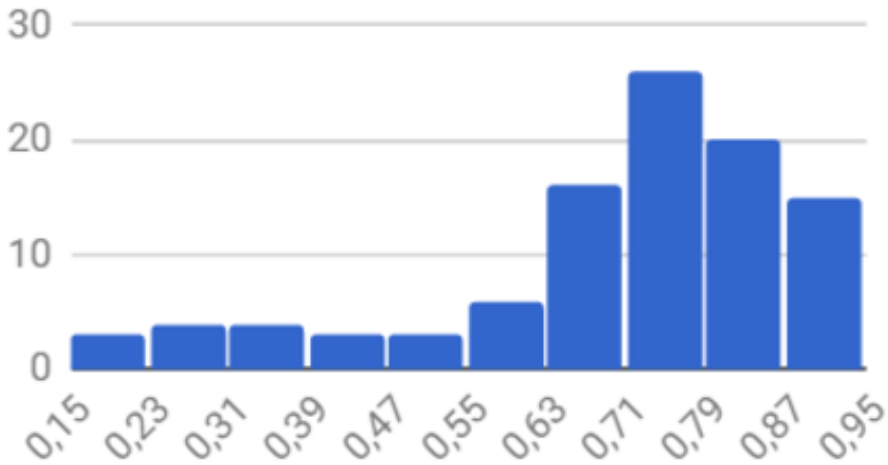
Обычная нейросеть

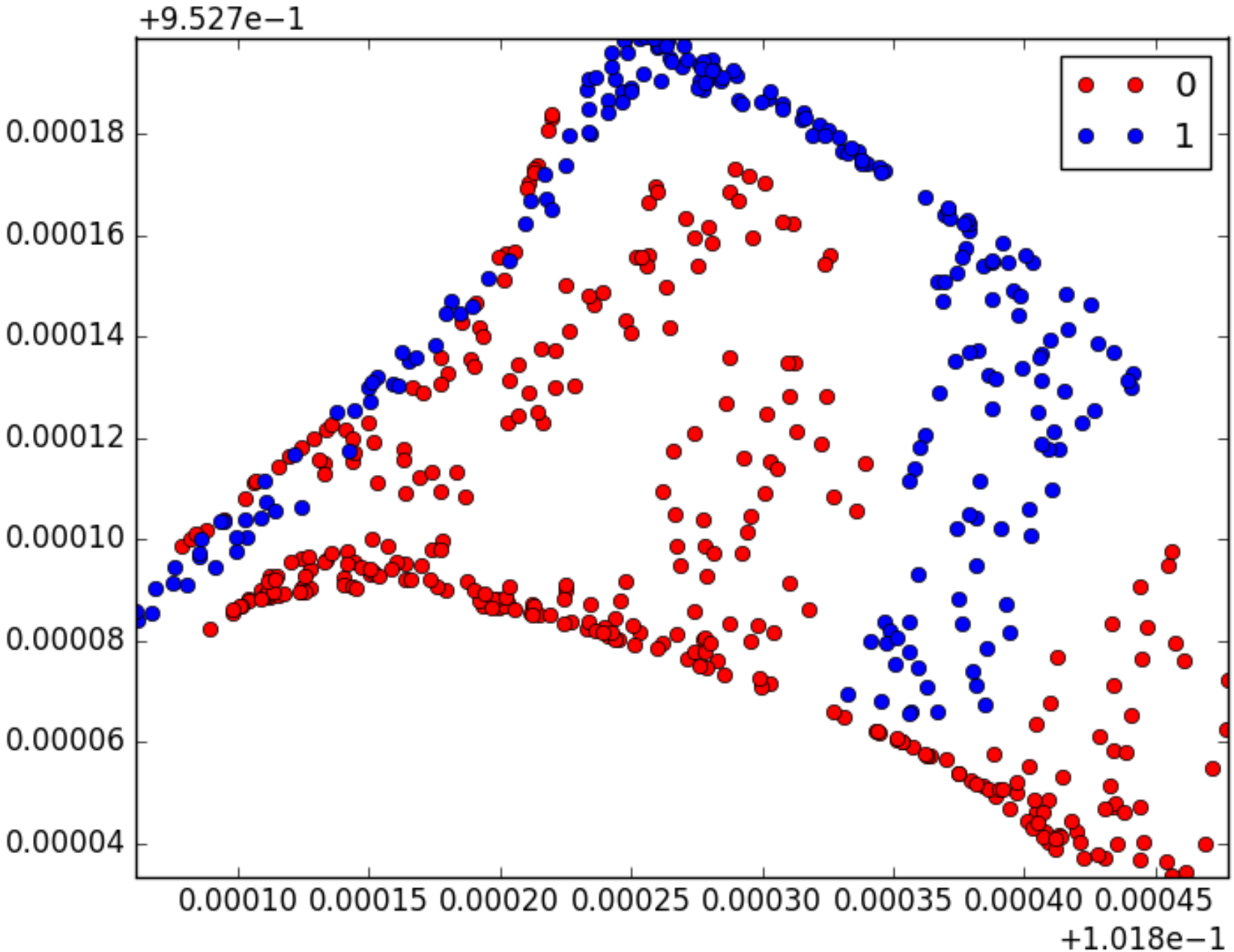


Подход #1

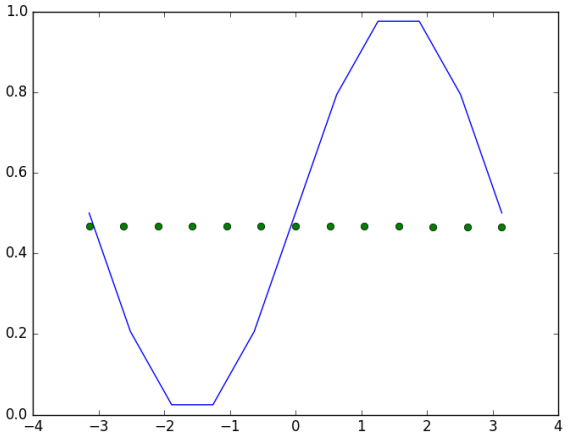


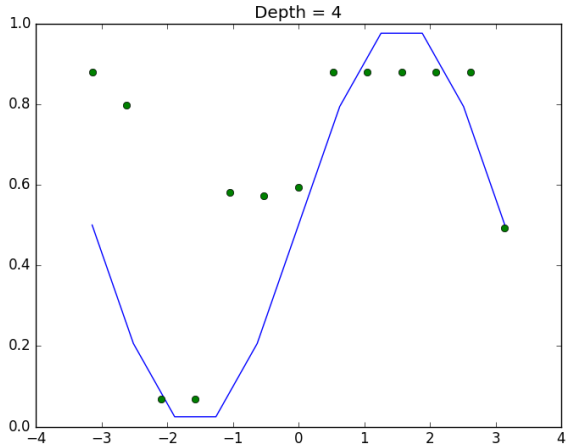
Подход #2

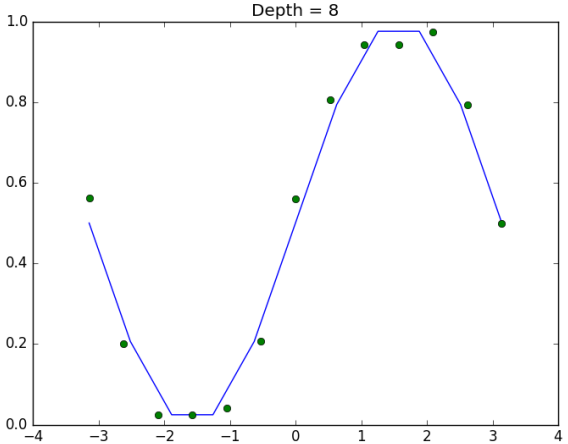


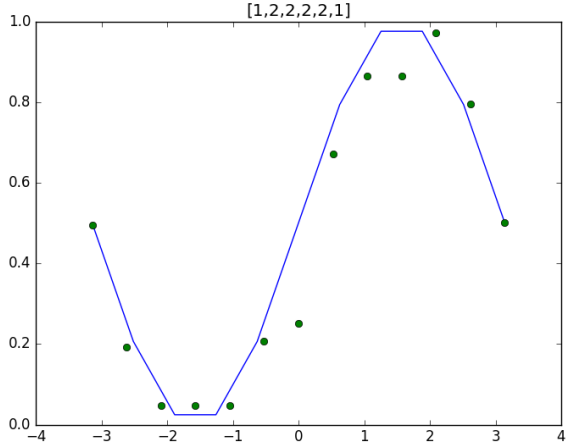

+9.527e−1
0.00018
0.00016
0.00014
0.00012
0.00010
0.00008
0.00006
0.00004
0.00010 0.00015 0.00020 0.00025 0.00030 0.00035 0.00040 0.00045
+1.018e−1
0
1

Спасибо за внимание!!


1.0
0.8
0.6
0.4
0.2
0.0
−4
−3
−2
−1
0
1
2
3
4


Depth = 4
1.0
0.8
0.6
0.4
0.2
0.0
−4
−3
−2
−1
0
1
2
3
4


Depth = 8
1.0
0.8
0.6
0.4
0.2
0.0
−4
−3
−2
−1
0
1
2
3
4


[1,2,2,2,2,1]
1.0
0.8
0.6
0.4
0.2
0.0
−4
−3
−2
−1
0
1
2
3
4