



Московский государственный университет имени М.В. Ломоносова

Факультет вычислительной математики и кибернетики

Кафедра нелинейных динамических систем и процессов управления

Сивков Антон Александрович

Применение нейронных сетей для моделирования объектов управления

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Научный руководитель:

д.ф-м.н., профессор

В. В. Фомичев

Москва, 2019

Оглавление

1	Введение	3
1.1	Постановка задачи	3
1.2	Виды идентификации	3
1.3	Перцептрон Розенблатта	4
1.4	Объект управления, использованный для экспериментов	6
2	Моделирование с помощью многослойного перцептрона	7
2.1	Структура сети	7
2.2	Код и схема модели	8
2.3	Эксперименты	10

Глава 1

Введение

1.1 Постановка задачи

Данная работа ставит целью сравнительный анализ способности нейронных сетей с различными архитектурами аппроксимировать объекты управления для задач идентификации, основное сравнение происходит в контексте способности сетей экстраполировать поведение объектов управления

1.2 Виды идентификации

В теории управления одной из основных задач является задача идентификации систем, задача построения математической модели объекта управления, способной адекватно воспроизводить реальное поведение объекта управления.

Задачи идентификации приблизительно можно разделить на 3 вида:

- Модель 'белого ящика', модель применяется, когда точно известен основополагающий принцип, лежащий в основе устройства системы. Модель может применяться например для систем, описывающих взаимодействия в рамках законов Ньютона. Зачастую такие системы имеют сложную структуру, и для построения точной модели требуются неадекватные затраты времени.
- Модель 'серого ящика', модель применяется, когда известны или имеются достаточно сильные предположения о характере соотношений между входом и выходом системы, но остается некоторое количество 'свободных' параметров

системы, которые приближаются с использованием данных выходов и входов системы. Модель может применяться, например для приближения параметров уравнения Моно при моделировании микробных популяций.

- Модель 'чёрного ящика', модель применяется, когда сложно сделать какие-либо априорные предположения о структуре системы, в этой модели используются только данные о входах и выходах системы, модель применяется к различным нелинейным системам. В этой работе используется эта модель, нейронные сети обучаются моделировать 'черный ящик'.

Сама идея применения неронных сетей для моделирования объектов управления не нова, примеры можно найти в публикации [1], датированной 1990 годом, где простейшие сети используются для моделирования простых дискретных линейных объектов управления.

1.3 Перцептрон Розенблатта

Имеет смысл дать краткую историю создания неронных сетей, и объяснить почему нейронные сети адекватно применять в широком спектре задач.

Одной из первых искусственных сетей, способных к перцепции (восприятию) и формированию реакции на воспринятый стимул, явился PERCEPTRON Фрэнка Розенблатта, предложенный им в 1957 году. Персептрон рассматривался его автором не как конкретное техническое вычислительное устройство, а как модель работы мозга. Нужно заметить, что после нескольких десятилетий исследований современные работы по искусственным нейронным сетям редко преследуют такую цель.

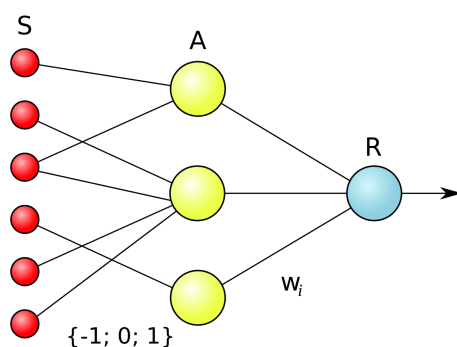


Рис. 1.1: Схема перцептрона.

Структура перцептрона представлена на рисунке 1.1. Веса S–A связей могут иметь значения -1, +1 или 0 (то есть отсутствие связи), S-элементы воплощают собой слой сенсоров (рецепторов). В физическом воплощении они соответствуют, например, светочувствительным клеткам сетчатки глаза или фоторезисторам матрицы камеры. A-элементы называются ассоциативными, потому что каждому такому элементу, как правило, соответствует целый набор (ассоциация) S-элементов. A-элемент активизируется, как только количество сигналов от S-элементов на его входе превысило некоторую величину θ . Сигналы от возбужденных A-элементов, в свою очередь, передаются в сумматор R. Веса A–R связей W могут быть любыми. R-элемент, а вместе с ним и элементарный перцептрон, выдаёт «1», если линейная форма превышает порог θ , иначе на выходе будет «-1». Перцептрон обучается правильно воспроизводить заданную функцию с помощью метода коррекции ошибок, в котором при ошибке веса изменяются на некоторую дельту, которая уменьшает ошибку на заданных данных. Розенблатт показал, что элементарный перцептрон, обучаемый по методу коррекции ошибки, независимо от начального состояния весовых коэффициентов и последовательности появления стимулов всегда приведёт к достижению решения за конечный промежуток времени.

Важнейшим утверждением, которое частично объясняет и обосновывает сходимость большей части используемых сегодня нейронных сетей, является теорема об универсальной аппроксимации. Она формулируется так:

Теорема 1 Пусть $\phi(\cdot)$ — ограниченная, непостоянная монотонно возрастающая непрерывная функция. Пусть I_{m_0} — m_0 -мерный единичный гиперкуб $[0, 1]^{m_0}$. Пусть пространство непрерывных на I_{m_0} функций обозначается символом $C(I_{m_0})$.

Тогда $\forall f \in C(I_{m_0})$ и $\varepsilon > 0$

$\exists m_1 \in \mathbb{Z}$ и множества $\{a_i\}, \{b_i\}, \{w_{ij}\} \subset \mathbb{R}, i = 1, \dots, m_1, j = 1, \dots, m_0$, что:

$$F(x_1, \dots, x_{m_0}) = \sum_{i=1}^{m_1} a_i \phi \left(\sum_{j=1}^{m_0} w_{ij} x_j + b_i \right)$$

является реализацией аппроксимации функции $f(\cdot)$, т.е.

$$|F(x_1, \dots, x_{m_0}) - f(x_1, \dots, x_{m_0})| < \varepsilon \quad \forall x_1, x_2, \dots, x_{m_0} \in I_{m_0}$$

Теорема об универсальной аппроксимации непосредственно применима к многослойному перцептрону.

Хотя для простейших архитектур нейронных сетей сходимость доказуема,

множество современных архитектур нейросетей не имеют под собой теоретической базы, которая обосновывала бы их сходимость, тем не менее, они сходятся, и показывают впечатляющие результаты в задачах обработки речи, изображений, естественных языков, и многих других.

1.4 Объект управления, использованный для экспериментов

Для экспериментов использовался простейший объект управления, который можно описать следующей дискретной передаточной функцией:

$$H = \frac{z}{(z - \frac{1}{2})(z - \frac{1}{3})(z - \frac{1}{4})} \quad (1.1)$$

Объект устойчив, в экспериментах $dt = 1 \text{ sec}$. Выглядит нелогичным проведение экспериментов, которые имеют смысл для сложных нелинейных объектов, на простейшем линейном дискретном объекте, однако данная работа имеет цель провести сравнительный анализ различных архитектур нейронных сетей, и, как показывает практика, даже такой простой объект позволяет сделать однозначные выводы относительно преимуществ одной архитектуры над другой.

Глава 2

Моделирование с помощью многослойного перцептрона

2.1 Структура сети

Многослойный перцептрон получил слово 'перцептрон' в названии в силу исторических причин, и его структура существенно отличается от структуры перцептрона Розенблатта, прежде всего тем, что он может иметь сколь угодно много слоев, и иметь обучаемые веса в каждом слое.

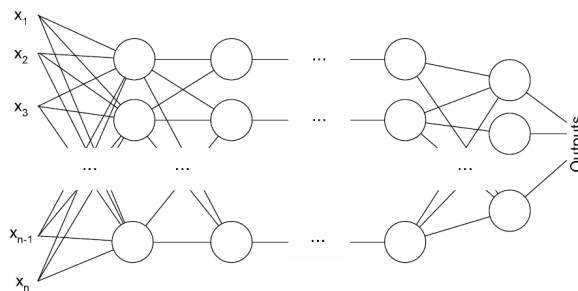


Рис. 2.1: Многослойный перцептрон.

Устройство перцептрона схематично представлено на рисунке ??, x_i обозначает i -й вход сети. Каждый из обозначенных на рисунке узлов сети выполняет следующее преобразование: $y = f(< w, x > + b)$, т. е. функция активации f (как правило сигмоида, тангенс или ReLu), примененная к скалярному произведению вектора входов узла x на вектор весов узла w плюс сдвиг b (bias). При обучении сети с помощью стохастического градиентного спуска веса изменяются на каждой итерации

в направлении антиградиента функции ошибки предсказания сети $L(x, y, w)$ на текущих данных. Значение градиента вычисляется с помощью метода *обратного распространения ошибки*. Этот метод вкратце можно описать так:

Пусть дан некий узел сети, для удобства отбросим функцию активации, т.к. она не влияет существенно на принцип вычислений. Пусть y — выход узла, w_i — веса узла, x_i — входы $i = 1, \dots, n$, тогда справедливы следующие формулы:

$$\frac{\partial L}{\partial w_i} = \frac{\partial L}{\partial y} * \frac{\partial y}{\partial w_i} = \frac{\partial L}{\partial y} * x_i \quad (2.1)$$

$$\frac{\partial L}{\partial x_i} = \frac{\partial L}{\partial y} * \frac{\partial y}{\partial x_i} = \frac{\partial L}{\partial y} * w_i \quad (2.2)$$

Формула 1.1 позволяет вычислить производную ошибки по конкретному весу сети, если известна производная ошибки по выходу узла $\frac{\partial L}{\partial y}$, а формула 2.1 позволяет вычислить производную ошибки по выходам узлов, не являющихся выходными, путем рекурсивного расчета производных по выходам, при этом расчет происходит от выходов сети ко входам, в направлении, *обратном* прямому вычислению.

2.2 Код и схема модели

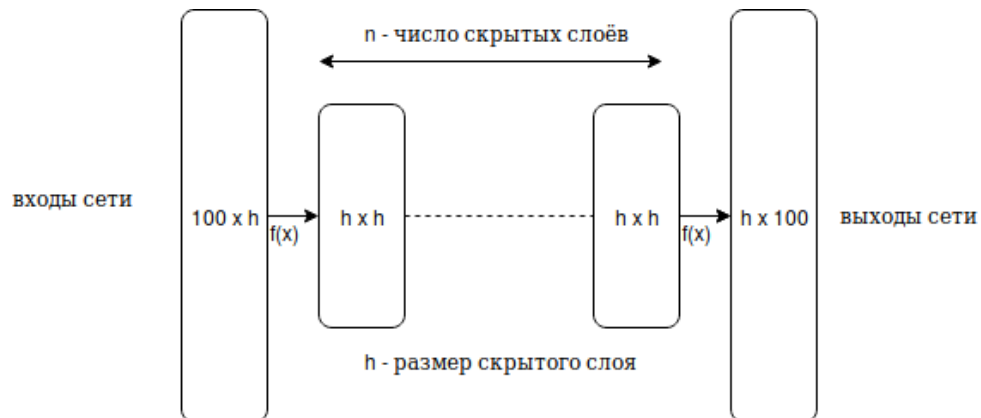


Рис. 2.2: Используемая для экспериментов архитектура

Один из вариантов многослойного перцептрона, использованного для моделирования объекта управления 1.1 изображен на рисунке 2.2, такой перцептрон может быть реализован на языке python с помощью библиотеки pytorch:


```

class MultiPercNet(nn.Module):

    def __init__(self, input_size, output_size, n_layers, layer_size):
        super().__init__()
        self.layers = nn.ModuleList([nn.Linear(input_size, layer_size)])
        self.layers.extend(nn.Linear(layer_size, layer_size) for _ in range(n_layer-1))
        self.layers.append(nn.Linear(layer_size, output_size))
        self.layer_size = layer_size
        for layer in self.layers:
            nn.init.xavier_uniform_(layer.weight, gain=1)
        self.input_size = input_size
        self.layer_size = layer_size
        self.tanh = nn.Hardtanh(min_val=-1, max_val=1)

    def forward(self, x):
        y = self.tanh(self.layers[0](x))
        for i in range(1, len(self.layers)):
            y = self.tanh(self.layers[i](y))
        return y

```

Сеть обучалась с использованием среднеквадратичной ошибки в качестве функции ошибки.

2.3 Эксперименты

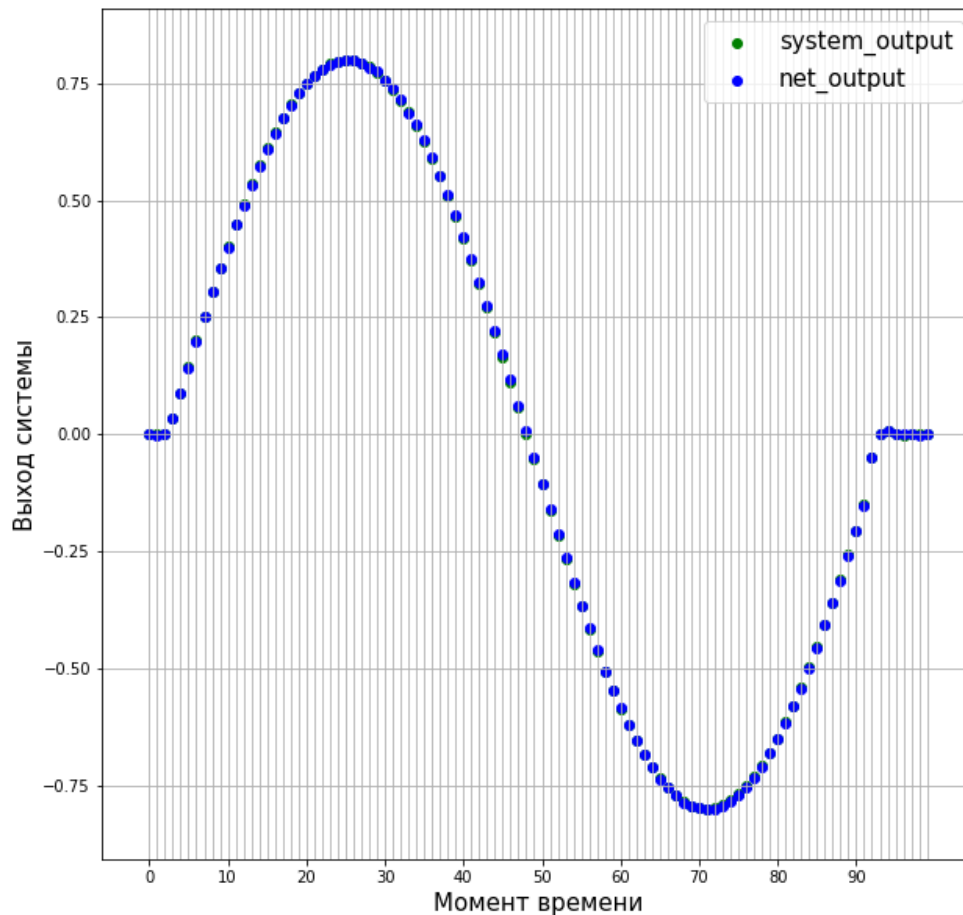


Рис. 2.3: Предсказание при отсутствии сдвига частот.

Во всех экспериментах сеть обучается на множестве входных и выходных сигналов системы, полученном при подаче на вход системы сигналов со случайной частотой, взятой из равномерного распределения на отрезке $[0.05 \text{ Гц}, 0.1 \text{ Гц}]$, входные и выходные сигналы берутся за первые 100 секунд. На рисунке 2.3 показано предсказание и реальный выход системы при подаче на вход системы сигнала со случайной частотой, полученной аналогично частоте входных сигналов, использованных при обучении. Среднеквадратичная ошибка при отсутствии сдвига частот: $1.1196042208666767e - 05$. Нетрудно заметить, что сеть предсказывает выход объекта управления практически идеально при сохранении той же частоты входных

сигналов, что использовалась при обучении. Это объясняется тем, что объект управления имеет простейшую структуру, и сеть без проблем аппроксимирует выходной сигнал объекта управления при подаче входных сигналов, не выходящих за область данных, использованных для обучения.

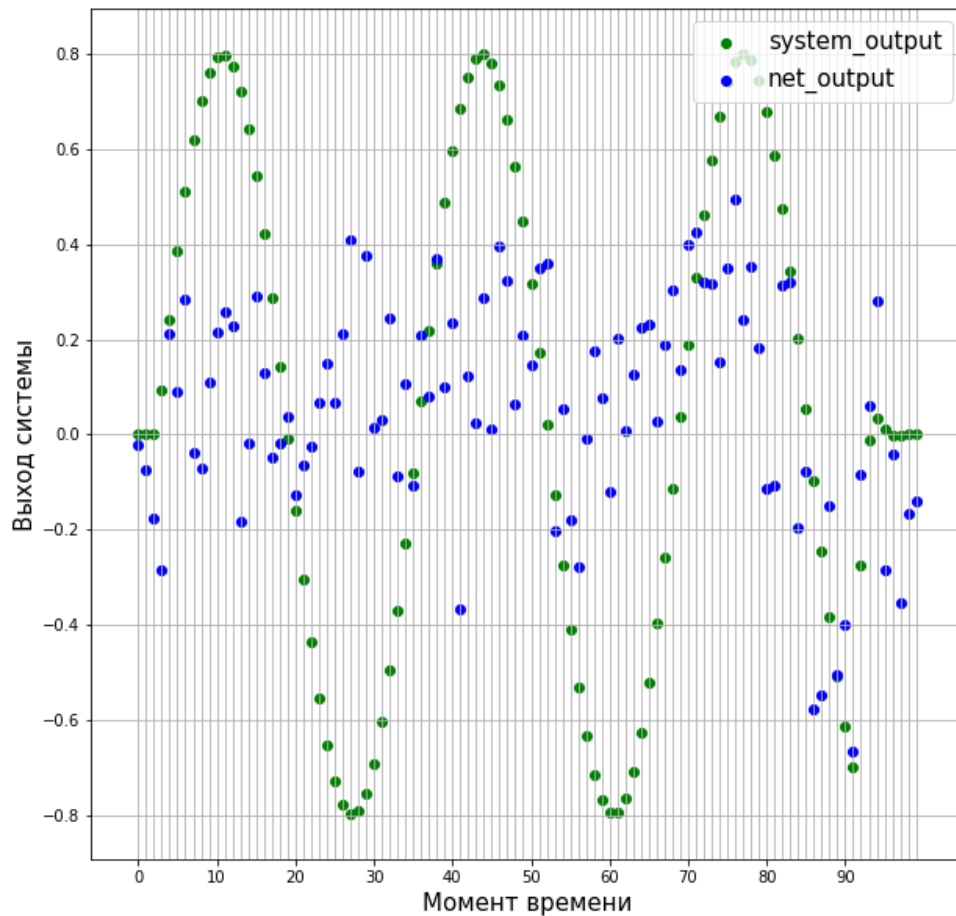
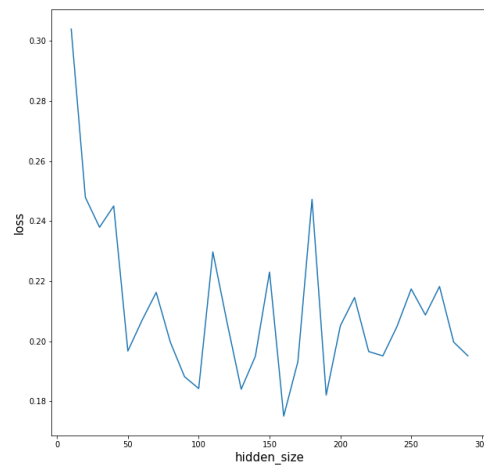


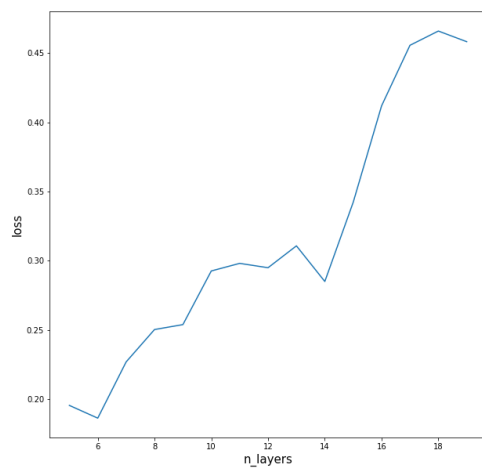
Рис. 2.4: Предсказание при частоте 0.2 Гц.

На рисунке 2.4 изображено предсказание выхода системы сетью, обученной на входных и выходных данных, полученных при подаче на вход системы сигналов из того же диапазона частот $[0.05 \text{ Гц}, 0.1 \text{ Гц}]$, при подаче на вход сигнала с частотой 0.2 Гц. Среднеквадратичная ошибка при подаче сигналов со случайной частотой из равномерного распределения $[0.1 \text{ Гц}, 0.2 \text{ Гц}]$: 0.19516661121453216. Из показателей и рисунка 2.4 можно сделать вывод о том что сеть не способна предсказать в

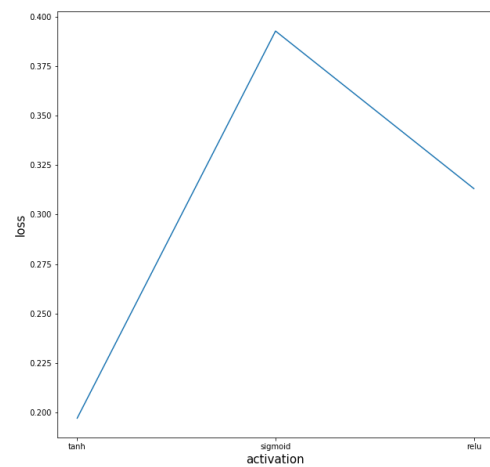
какой-либо степени релевантные данных при выходе частоты входного сигнала за пределы параметров обучающей выборки. Это также позволяет сделать вывод о том, что сеть переобучается на данных обучающей выборки, и теряет способность к какой-либо экстраполяции поведения объекта управления.



(а) Перебор размера скрытого слоя сети



(б) Перебор количества внутренних слоев



(в) Перебор функций активации

Рис. 2.5: Предсказание поведения объекта

Литература

- [1] KUMPATI S. NARENDRA FELLOW, IEEE. AND KANNAN PARTHASARATHY,
Identification and control of dynamical systems using neural networks, 1990,
<https://ieeexplore.ieee.org/abstract/document/80202>