

**мФТиАД ФКН ВШЭ, 1 курс, 4 модуль**

## **Задание 7. Нелинейные модели.**

### **Полносвязные нейронные сети**

**Прогнозирование временных данных и случайных процессов,  
весна 2018**

Время выдачи задания: 31 мая (четверг).

Срок сдачи: **17 июня (воскресенье), 23:59.**

Среда для выполнения практического задания – PYTHON 2.x.

## **Правила сдачи**

### **Выполнение работы в команде**

1. Домашнее задание допускается выполнять в команде от 1 до 4 человек.
2. Командное решение достаточно загрузить в AnyTask только один раз. При этом в посылке следует указать состав команды.
3. Баллы, набранные командой, выставляются всем членам команды одинаковыми. Бонусные баллы выставляются всем членам команды одинаковыми. Это означает, что каждый член команды получает баллы, набранные его командой, независимо от его вклада в решение работы.

### **Инструкция по отправке:**

1. Решения задач следует присылать единым файлом формата .pdf, набранным в L<sup>A</sup>T<sub>E</sub>X. Допускается отправка отдельных практических задач в виде отдельных файлов (ipython-тетрадок или исходных файлов с кодом на языке python).

### **Оценивание и штрафы:**

1. Максимально допустимая оценка за работу – 10 баллов. В этом задании в 2018 году нет бонусных баллов.
2. Дедлайн жесткий. Сдавать задание после указанного срока сдачи нельзя.
3. Задание выполняется командой независимо от других команд. «Похожие» решения считаются плагиатом и все студенты обеих команд (в том числе те, у кого списали) не могут получить за него больше 0 баллов (подробнее о плагиате см. на странице курса). Если вы нашли решение какого-то из заданий (или его часть) в открытом источнике, необходимо указать ссылку на этот источник в отдельном блоке в конце вашей работы (скорее всего вы будете не единственным, кто это нашел, поэтому чтобы исключить подозрение в плагиате, необходима ссылка на источник).

# Вариант 1

1. (4 балла) Рассмотрим полносвязную нейронную сеть с одним скрытым слоем, в которой выход  $\hat{y} \in \mathbb{R}$  вычисляется по входу  $\mathbf{x} \in \mathbb{R}^n$  согласно соотношениям

$$\hat{y} = \sum_{j=1}^{n_2} W_j^{(2)} \sigma\left(\sum_{i=1}^n W_{ji}^{(1)} x_i + b_j^{(1)}\right) + b^{(2)},$$

где  $n_1$  – размерность скрытого слоя,  $\mathbf{W}^{(1)} \in (\mathbb{R}^n \rightarrow \mathbb{R}^{n_1})$  и  $\mathbf{b}^{(1)} \in \mathbb{R}^{n_1}$  – параметры, определяющие преобразование входа в активации скрытого слоя,  $\mathbf{W}^{(2)} \in (\mathbb{R}^{n_1} \rightarrow \mathbb{R})$  – вектор-строка, и  $b^{(2)} \in \mathbb{R}$  – параметры преобразования активаций скрытого слоя в выход.

Рассмотрим робастную регрессионную целевую функцию:

$$L(y, \hat{y}) = \|y - \hat{y}\|_1.$$

- (а) (1 балл) Выпишите уравнения алгоритма обратного распространения ошибки (включая явные соотношения для производных  $\frac{\partial L}{\partial W_j^{(2)}}, \frac{\partial L}{\partial b^{(2)}} \frac{\partial L}{\partial W_{ij}^{(1)}}, \frac{\partial L}{\partial b_j^{(1)}}$ ).
- (б) (1 балл) Не используя фреймворки автоматического дифференцирования, реализуйте на языке `python` алгоритм обратного распространения ошибки для численной оптимизации параметров сети. В отчете приведите код (или явно укажите ячейку `ipython`-тетрадки) с реализацией алгоритма.
- (с) (1 балл) Провалидируйте корректность реализации алгоритма на следующем простом примере<sup>1</sup>.

Пусть  $\mathbf{x} \in \mathbb{R}^2$ , а функциональная зависимость  $y = f(\mathbf{x})$  выражается функцией

$$f(\mathbf{x}) = 20 + x_1^2 + x_2^2 - 2 \cos(2\pi x_1) - 2 \cos(2\pi x_2).$$

---

<sup>1</sup>См. [https://en.wikipedia.org/wiki/Rastrigin\\_function](https://en.wikipedia.org/wiki/Rastrigin_function).

Сгенерируйте  $\ell = 1000$  точек  $\mathbf{x}_i = (x_{i1}, x_{i2})$ ,  $x_{i1,2} \sim U[-5, 5]$  и соответствующих им  $y_i = f(\mathbf{x}_i)$ . Выберите какое-нибудь число  $n_1$  нейронов скрытого слоя. Обучите методом обратного распространения ошибки нейронную сеть приближать зависимость  $y = f(\mathbf{x})$  по выборке  $\mathbf{X}^\ell = (\mathbf{x}_i, y_i)_{i=1}^\ell$ . Визуализируйте процесс обучения с помощью графика зависимости средней ошибки на выборке  $\sum_{i=1}^\ell L(y_i, \hat{y}_i)$  от количества просмотренных обучающих примеров.

- (d) (1 балл) Векторизуйте вычисления в алгоритме обратного распространения ошибки для ускорения вычислений (используйте только векторные операции с объектами типа `numpy.ndarray`).
2. (3 балла) Создайте алгоритм прогнозирования следующего значения временного ряда на основе архитектуры, описанной в задаче 1 (и кода, полученного в результате ее выполнения). Для этого скачайте данные о количестве пятен на Солнце по ссылке [http://www.sidc.be/silso/DATA/SN\\_ms\\_tot\\_V2.0.txt](http://www.sidc.be/silso/DATA/SN_ms_tot_V2.0.txt). Формат этих данных описан на странице <http://www.sidc.be/silso/newdataset>. В качестве целевой переменной выберите `SNvalue`, в качестве переменных-регрессоров – предыдущие  $n$  значений переменной `SNvalue`.
- (a) (1 балл) Используя подходы, описанные в литературе и упомянутые на лекции, создайте алгоритм прогнозирования следующего значения временного ряда. В отчете приведите код алгоритма обучения и алгоритма прогнозирования следующего значения.
  - (b) (1 балл) Обучите модель прогнозирования следующего значения на скачанных данных о пятнах на Солнце. При этом используйте для обучения параметров модели случайно вырезанные сегменты  $([s_i, s_i + n])_{i=1}^\ell$  временного ряда, а для оценки

качества – независимые сегменты  $([t_i, t_i + \ell])_{i=1}^{\ell}$ , где  $\min_i t_i > \max_i s + n$  (обучающая и тестовая выборка не должны пересекаться).

- (с) (1 балл) Исследуйте зависимость качества прогнозирования от длины контекста  $n$  и количества нейронов скрытого слоя  $n_1$ . Для этого задайте  $n \in \{32, 64, 128, 256, 512, 1024\}$  и постройте зависимость ошибки модели на валидационной выборке от значения  $n$  (при фиксированном  $n_1$ ). То же самое проделайте с  $n_1$  (при фиксированном  $n$ ). Эксперименты для каждого значения  $n$  повторите 10 раз и приведите на графике в виде box-plot<sup>2</sup>.
3. (3 балла) Для нейронной сети с архитектурой, описанной в задаче 1, реализуйте алгоритм Левенберга-Марквардта для оптимизации параметров сети.
- (а) (1 балл) Выпишите уравнения алгоритма Левенберга-Марквардта для вычисления градиента функции штрафа по параметрам  $\mathbf{W}^{(1)}$ ,  $\mathbf{b}^{(1)}$ ,  $\mathbf{W}^{(2)}$ ,  $\mathbf{b}^{(2)}$ .
- (б) (1 балл) Не используя фреймворки автоматического дифференцирования, реализуйте на языке `python` (векторизованный) алгоритм Левенберга-Марквардта для численной оптимизации параметров сети. В отчете приведите код (или явно укажите ячейку `ipython`-тетрадки) с реализацией алгоритма.
- (с) (1 балл) Примените реализованный алгоритм для решения задачи прогнозирования количества пятен на Солнце на следующий день. В отчете приведите код алгоритма обучения и алгоритма прогнозирования следующего значения.

---

<sup>2</sup>[https://en.wikipedia.org/wiki/Box\\_plot](https://en.wikipedia.org/wiki/Box_plot).

## Вариант 2

1. (3 балла) Рассмотрим полносвязную нейронную сеть с одним скрытым слоем, в которой выход  $\hat{y} \in \mathbb{R}^m$  вычисляется по входу  $\mathbf{x} \in \mathbb{R}^n$  согласно соотношениям

$$\begin{aligned}\mathbf{a}^{(1)} &= \tanh(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}), \\ \hat{\mathbf{y}} &= \mathbf{W}^{(2)}\mathbf{a}^{(1)} + \mathbf{b}^{(2)},\end{aligned}$$

где  $n_1$  – размерность скрытого слоя,  $\mathbf{W}^{(1)} \in (\mathbb{R}^n \rightarrow \mathbb{R}^{n_1})$  и  $\mathbf{b}^{(1)} \in \mathbb{R}^{n_1}$  – параметры, определяющие преобразование входа в активации скрытого слоя,  $\mathbf{W}^{(2)} \in (\mathbb{R}^{n_1} \rightarrow \mathbb{R}^m)$ , и  $\mathbf{b}^{(2)} \in \mathbb{R}^m$  – параметры преобразования активаций скрытого слоя в выход.

Рассмотрим регрессионную целевую функцию:

$$L(\mathbf{y}, \hat{\mathbf{y}}) = \sum_{i=1}^m \text{smooth}(y_i - \hat{y}_i),$$

где

$$\text{smooth}(x) = \begin{cases} 0.5x^2 & \text{если } |x| < 1, \\ |x| - 0.5 & \text{иначе.} \end{cases}$$

- (а) (1 балл) Выпишите уравнения алгоритма обратного распространения ошибки (включая явные соотношения для производных  $\frac{\partial L}{\partial W_{ij}^{(2)}}$ ,  $\frac{\partial L}{\partial b_i^{(2)}}$ ,  $\frac{\partial L}{\partial W_{ij}^{(1)}}$ ,  $\frac{\partial L}{\partial b_i^{(1)}}$ ).
- (б) (1 балл) Не используя фреймворки автоматического дифференцирования, реализуйте на языке `python` алгоритм обратного распространения ошибки для численной оптимизации параметров сети. В отчете приведите код (или явно укажите ячейку `ipython`-тетрадki) с реализацией алгоритма.
- (с) (1 балл) Провалидируйте корректность реализации алгоритма на следующем простом примере<sup>3</sup>.

---

<sup>3</sup>См. [https://en.wikipedia.org/wiki/Rastrigin\\_function](https://en.wikipedia.org/wiki/Rastrigin_function).

Пусть  $\mathbf{x} \in \mathbb{R}^2, y \in \mathbb{R}^1$ , а функциональная зависимость  $y = f(\mathbf{x})$  выражается функцией

$$f(\mathbf{x}) = 20 + x_1^2 + x_2^2 - 2 \cos(2\pi x_1) - 2 \cos(2\pi x_2).$$

Сгенерируйте  $\ell = 1000$  точек  $\mathbf{x}_i = (x_{i1}, x_{i2}), x_{i1,i2} \sim U[-5, 5]$  и соответствующих им  $y_i = f(\mathbf{x}_i)$ . Выберите какое-нибудь число  $n_1$  нейронов скрытого слоя. Обучите методом обратного распространения ошибки нейронную сеть приближать зависимость  $y = f(\mathbf{x})$  по выборке  $\mathbf{X}^\ell = (\mathbf{x}_i, y_i)_{i=1}^\ell$ . Визуализируйте процесс обучения с помощью графика зависимости средней ошибки на выборке  $\sum_{i=1}^\ell L(y_i, \hat{y}_i)$  от количества просмотренных обучающих примеров.

- (d) (1 балл) Векторизуйте вычисления в алгоритме обратного распространения ошибки для ускорения вычислений (используйте только векторные операции с объектами типа `numpy.ndarray`).

2. (3 балла) Создайте алгоритм прогнозирования следующего значения временного ряда на основе архитектуры, описанной в задаче 1 (и кода, полученного в результате ее выполнения). Для этого скачайте данные о количестве пятен на Солнце по ссылке [http://www.sidc.be/silso/DATA/SN\\_ms\\_tot\\_V2.0.txt](http://www.sidc.be/silso/DATA/SN_ms_tot_V2.0.txt). Формат этих данных описан на странице <http://www.sidc.be/silso/newdataset>. В качестве целевой переменной выберите  $m$  значений `SNvalue`, в качестве переменных-регрессоров – предыдущие  $n$  значений переменной `SNvalue`.

- (a) (1 балл) Используя подходы, описанные в литературе и упомянутые на лекции, создайте алгоритм прогнозирования следующего значения временного ряда. В отчете приведите код

алгоритма обучения и алгоритма прогнозирования следующего значения.

- (b) (1 балл) Обучите модель прогнозирования следующего значения на скачанных данных о пятнах на Солнце. При этом используйте для обучения параметров модели случайно вырезанные сегменты  $([s_i, s_i + n])_{i=1}^{\ell}$  временного ряда, а для оценки качества – независимые сегменты  $([t_i, t_i + \ell])_{i=1}^{\ell}$ , где  $\min_i t_i > \max_i s + n$  (обучающая и тестовая выборка не должны пересекаться).
  - (c) (1 балл) Исследуйте зависимость качества прогнозирования от длины контекста  $n$  и количества нейронов скрытого слоя  $n_1$ . Для этого задайте  $n \in \{32, 64, 128, 256, 512, 1024\}$  и постройте зависимость ошибки модели на валидационной выборке от значения  $n$  (при фиксированном  $n_1$ ). То же самое проделайте с  $n_1$  (при фиксированном  $n$ ). Эксперименты для каждого значения  $n$  повторите 10 раз и приведите на графике в виде box-plot<sup>4</sup>.
3. (3 балла) Для нейронной сети с архитектурой, описанной в задаче 1, реализуйте алгоритм Левенберга-Марквардта для оптимизации параметров сети.
- (a) (1 балл) Выпишите уравнения алгоритма Левенберга-Марквардта для вычисления градиента функции штрафа по параметрам  $\mathbf{W}^{(1)}$ ,  $\mathbf{b}^{(1)}$ ,  $\mathbf{W}^{(2)}$ ,  $\mathbf{b}^{(2)}$ .
  - (b) (1 балл) Не используя фреймворки автоматического дифференцирования, реализуйте на языке `python` (векторизованный) алгоритм Левенберга-Марквардта для численной оптимизации параметров сети. В отчете приведите код (или явно

---

<sup>4</sup>[https://en.wikipedia.org/wiki/Box\\_plot](https://en.wikipedia.org/wiki/Box_plot).



укажите ячейку `ipython`-тетрадки) с реализацией алгоритма.

- (с) (1 балл) Примените реализованный алгоритм для решения задачи прогнозирования количества пятен на Солнце на следующий день. В отчете приведите код алгоритма обучения и алгоритма прогнозирования следующего значения.