

УДК 004.89

<https://doi.org/10.36906/AP-2020/34>

## ИССЛЕДОВАНИЕ ВРЕМЕННЫХ РЯДОВ ПРИ ПОМОЩИ НЕЙРОННЫХ СЕТЕЙ

**Тагиров Т. М.***Нижневартровский государственный университет  
г. Нижневартовск, Россия***Тагиров К. М.***Нижневартровский государственный университет  
г. Нижневартовск, Россия***Катермина Т. С.***канд. техн. наук  
Нижневартровский государственный университет  
г. Нижневартовск, Россия*

**Аннотация.** В данной статье рассматривается краткий обзор прогнозирования временных рядов с использованием рекуррентных нейронных сетей на примере одномерно и многомерных рядов данных.

**Ключевые слова:** искусственные нейронные сети; математическая модель; временной ряд, машинное обучение.

В современном мире, существует множество задач, связанных с обработкой большого количества данных. С развитием компьютерных технологий приходит и решение задачи с прогнозированием временных рядов. Например, прогноз температуры или биржевых счетов, которые являются не постоянными и меняются в определенный момент времени.

Дадим некоторое определение временному ряду. Временной ряд — это значения признака, измеренные через постоянные временные интервалы [1]. То есть это некоторый ряд чисел. Главной особенностью временного ряда являются постоянные временные интервалы с определенно заданным значением, например, один день или месяц.

Задачу прогнозирования временных рядов можно исследовать с помощью рекуррентных нейронных сетей. В этом случае берется множество значений  $X$  для обучения и множество  $Y$  для проверки обученной нейронной сети. Изначально, перед обучением масштабируют данные (множество значений  $X$ ), стандартизируют данные посредством вычитания среднего значения деленное на стандартное отклонение для каждого признака. На рисунке 1 схематично представлена подготовка данных для модели с одномерным входом.

Рассмотрим базовое решение (без машинного обучения), которое заключается в следующем: для заданного входного вектора прогнозирует следующее значение как среднее из последних  $n$  значений. Рекуррентная нейронная сеть в свою очередь, шаг за шагом обрабатывает временной ряд данных, перебирая значения и сохраняя внутреннее состояние, полученное при обработке предыдущих значений. Воспользуемся специализированным слоем PNC (<https://clck.ru/T7t5o>) [2], *Long Short-Term Memory (LSTM*, с англ. Долгая краткосрочная память). Далее с помощью функции *tf.data* выполняется перемешивание (*shuffle*), пакетирование (*batch*) и кэширование (*cache*) набора данных. Получим следующий пакет: временной интервал на количество признаков (рис. 2).



Рис. 1. Подготовка данных для прогнозирования

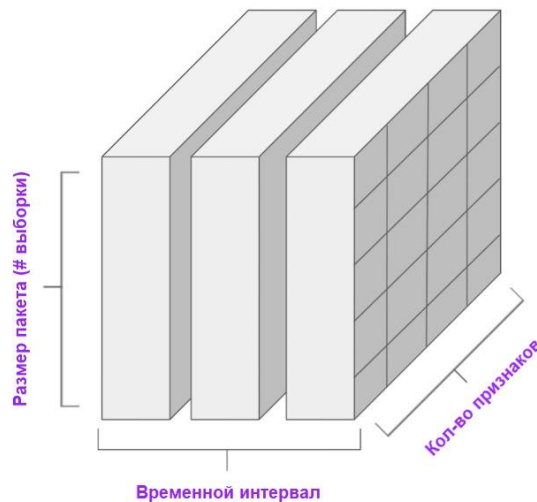


Рис. 2. Временной интервал на количество признаков

Стандартизированные данные можно применять для обучения модели искусственной нейронной сети. Рассмотрим прогнозирование с использованием простой модели *LSTM* ( см. листинг 1.)

Листинг 1

```

for x, y in val_univariate.take(3):
    plot = show_plot([x[0].numpy(), y[0].numpy(),
                    simple_lstm_model.predict(x)[0]], 0, 'Simple LSTM model')
    plot.show()
    
```

Данный код представлен для одномерного временного ряда. Для многомерных рядов первым шагом, как и в одномерном ряде, будет стандартизированные набора данных с вычислением среднего значения и стандартного отклонения обучающих данных. Далее

рассмотрим два прогнозирования точечное и интервальное. Для точечного прогнозирования в конечном итоге получается одно значение в предсказании, а для интервального в результате будет последовательность значений.

Ниже представлена функция (листинг 2), выполняющая задачу организации временных интервалов лишь с тем отличием от одномерной, что здесь отбираются, на основе заданного размера шага, последние наблюдения.

Листинг 2

```
def multivariate_data(dataset, target, start_index, end_index, history_size,
                      target_size, step, single_step=False):
    data = []
    labels = []

    start_index = start_index + history_size
    if end_index is None:
        end_index = len(dataset) - target_size

    for i in range(start_index, end_index):
        indices = range(i-history_size, i, step)
        data.append(dataset[indices])

        if single_step:
            labels.append(target[i+target_size])
        else:
            labels.append(target[i:i+target_size])

    return np.array(data), np.array(labels)
```

В заключении отметим, что исследование временных рядов как одномерных, так и многомерных, можно реализовать с помощью рекуррентных нейронных сетей. И в зависимости от условия задачи, нейронная сеть может предсказывать как точечное значение, так и последовательный ряд данных.

### Литература

1. Бокс Дж. Дженкинс Г. М. Анализ временных рядов, прогноз и управление. М.: Мир, 1974. 406 с.
2. Шолле Ф. Глубокое обучение на Python. СПб. Питер, 2018. 400 с.

©Тагиров Т. М., Тагиров К. М., Катермина Т. С., 2020