

НЕЙРОСЕТЕВЫЕ МОДЕЛИ ДЛЯ БИНАРНОЙ КЛАССИФИКАЦИИ ДАННЫХ ТЕЛЕМЕТРИЧЕСКОЙ ИНФОРМАЦИИ МАЛЫХ КОСМИЧЕСКИХ АППАРАТОВ

Скобцов В. Ю.

Объединённый институт проблем информатики Национальной академии наук Беларуси
Кафедра ПОИТ, Белорусский государственный университет информатики и радиоэлектроники
Минск, Республика Беларусь
E-mail: vasko_vasko@mail.ru

Разработаны и исследованы нейросетевые модели на основе современных архитектур глубокого обучения для решения задачи бинарной классификации данных телеметрической информации МКА, позволяющие определять штатное и нештатное состояние функционирования МКА или некоторых его подсистем. Проведены компьютерные эксперименты по обучению, валидации и тестированию построенных нейросетевых моделей, показавшие их достаточно высокую точность в решении рассматриваемой задачи.

ВВЕДЕНИЕ

Одной из важнейших задач на всех этапах жизненного цикла малых космических аппаратов (МКА) является анализ их телеметрической информации (ТМИ) о функционировании МКА с точки зрения определения их технического состояния для обеспечения их корректного и безопасного функционирования. Их актуальность обусловлена прежде всего тем, что одной из основных причин потерь МКА являются отказы, сбой и некорректная работа систем МКА.

Большое количество информации, поступающей и накапливающейся в специализированных банках данных с МКА, может быть эффективно использовано для определения технического состояния МКА и его подсистем. В современных условиях для решения задач обеспечения требуемой степени автономности, качества и оперативности управления такими сложными объектами как МКА необходимо выполнить комплексную автоматизацию и интеллектуализацию процессов многомодельного анализа данных ТМИ МКА. При этом в соответствии с ГОСТ 1410-002-2010 и Стратегией цифровой трансформации ракетно-космической отрасли до 2025 г. и перспективой до 2030г. госкорпорации «Роскосмос» важной задачей является создание так называемой системы информации о техническом состоянии и надежности космических комплексов и входящих в их состав изделий [1].

При этом разработка и применение методов анализа данных ТМИ МКА на основе моделей искусственного интеллекта и машинного обучения систем позволяет на новом научно-техническом уровне решить поставленную задачу и повысить эффективность принимаемых управленческих и эксплуатационных решений наземных комплексов управления (НКУ) МКА.

I. ЗАДАЧА БИНАРНОЙ КЛАССИФИКАЦИИ ДАННЫХ ТМИ МКА

Исходные данные ТМИ являются временным рядом, который можно представить как

матрицу, где i -я строка является анализируемым вектором показателей ТМИ в i -й момент времени, индекс j соответствует j -му показателю ТМИ в i -м векторе.

Определение 1. Одномерный временной ряд $\mathbf{X} = (x_1, x_2, \dots, x_T)$ – упорядоченный набор вещественных значений. Длина X равна количеству вещественных значений T .

Определение 2. M -мерный временной ряд $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_M)$ состоит из M различных одномерных временных рядов $\mathbf{X}_j \in R^T$.

Рассматриваемый временной ряд данных ТМИ является M -мерным временным рядом $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_M)$. Для каждого вектора показателей ТМИ в i -й момент времени \mathbf{X}_i в соответствие поставлена метка класса $y_i \in \mathbf{Y}$, который характеризует состояние функционирования анализируемого по данным ТМИ МКА или его подсистемы. В исследуемой задаче мы рассматриваем случай бинарной классификации, поскольку конечной целью является определить к штатному или нештатному состоянию принадлежит анализируемый вектор \mathbf{X}_i M -мерного временного ряда \mathbf{X} . В этом случае число классов $K = 2$ и, следовательно, $\mathbf{Y} \in \{0, 1\}$, где 0 обозначает штатное состояние и 1 – нештатное состояние анализируемой подсистемы МКА. Таким образом, стоит задача нахождения модели следующего отображения: $y : \mathbf{X} \rightarrow \mathbf{Y}$.

При этом будем использовать так называемое One Hot кодирование. В последнем случае вектору \mathbf{X}_i M -мерного временного ряда \mathbf{X} соответствует вектор $\mathbf{Y}_i = (y_{i0}, y_{i1})$ размерности $K = 2$. При этом в векторе \mathbf{Y}_i присутствует только одно значение 1, соответствующее метке класса 0 : (1, 0) или 1 : (0, 1).

II. АНАЛИЗ НЕЙРОСЕТЕВЫХ МОДЕЛЕЙ

В течение последних двух десятилетий классификация временных рядов считается одной из самых сложных проблем в области интеллектуального анализа данных [2,3]. Фактически, любая проблема классификации, использующая

данные, которые регистрируются с учетом некоторого понятия упорядочивания можно рассматривать как проблему классификации временного ряда. Временные ряды встречаются во многих реальных приложениях: электронных медицинских карт, распознавания человеческой деятельности, классификации акустических сцен, кибербезопасности, анализа состояний функционирования МКА по данным ТМИ [3,4].

Определение 3. Искусственной нейронной сетью (нейросетевой моделью) называется система, состоящая из совокупности связанных между собой по типу узлов направленного графа элементарных процессоров, называемых искусственными или формальными нейронами, и способная генерировать выходную информацию в ответ на входное воздействие.

Искусственный нейрон имитирует свойства биологического нейрона. Здесь множество входных сигналов, обозначенных $x_i, i = 1, \dots, n$, поступает на искусственный нейрон. Эти входные сигналы, в совокупности обозначаемые вектором $X = (x_1, x_2, \dots, x_n)$, соответствуют сигналам приходящим в синапсы биологического нейрона. Каждый синапс характеризуется величиной синаптической связи или ее весом w_i . Каждый входной сигнал умножается на соответствующий вес w_i , и поступает на суммирующий блок. Множество весов в совокупности обозначаются вектором W . Суммирующий блок, соответствующий телу биологического нейрона, складывает взвешенные входы алгебраически, создавая величину S . Полученная величина S подается на активационную выходную функцию нейрона a , моделируя процесс активации или нелинейную передаточную характеристику биологического нейрона. Таким образом, математическую модель искусственного нейрона можно представить выражением

$$y = a(S) = a\left(\sum_{i=1}^n w_i x_i + b\right),$$

где y - выходной сигнал нейрона, b - начальное возбуждение или смещение.

В анализе и разработке нейросетевых моделей мы будем рассматривать нейросетевые модели от простого к более сложному, начиная с основных на текущий момент нейросетевых моделей:

- полносвязных нейронных сетей/слоев (многослойных перцептронов, MLP);
- одномерных (1D) сверточных нейронных сетей/слоев (1D CNN);
- рекуррентных нейронных сетей/слоев типа Long Short-Term Memory (LSTM) и Gated Recurrent Units (GRU), и продолжая их комбинациями, в том числе, на основе архитектур ResNet и Inception [5].

В качестве активационных функций будем использовать следующие широко используемые активационные функции [6,7]:

- функция relu - rectified linear unit, линейный выпрямитель $\text{relu}(z) = \max(0, z)$;
- обобщение логистической функции для One-Hot кодирования меток классов $\text{softmax}(z_i) = e^{z_i} / \sum_{i=0}^1 e^{z_i}$.

В качестве функции ошибки будем использовать функцию двоичной кроссэнтропии, поскольку мы решаем задачу бинарной классификации [5].

Основываясь на описанных выше, нейросетевых архитектурах, были предложены и исследованы 3-слойная MLP модель, 5-слойная сверточная 1D CNN модель, 4-слойные рекуррентные LSTM и GRU модели. Каждая из них использовала финальный классификационный полносвязный слой из 2 нейронов с логистической функцией.

Компьютерные эксперименты проводились на реальных данных ТМИ одной из подсистем МКА. Каждый вектор матрицы ТМИ X_i имеет размерность 9 и помечается 0 в случае штатного состояния и 1 в случае нештатного состояния подсистемы. Общая размерность 9-мерного временного ряда X составляет 121690 векторов, из которых 77881 вектор составляет обучающий набор данных, 19471 вектор составляет валидационный набор данных, 24338 векторов составляет тестовый набор данных.

Для приведенной выше группы нейросетевых моделей обучение и валидация проводились при следующих значениях гиперпараметров: метод обучения adam (как один из наиболее эффективных на текущий момент), функция потерь - binary crossentropy, число эпох обучения - 500, размер мини-пакета - 128. Механизм раннего останова [5] не использовался и процесс обучения и валидации проходил на всех 500 эпохах.

Как показывают данные экспериментов, наименьшей точностью, как на этапе обучения - 0.8816, этапе валидации - 0.8810 и тестирования - 0.8809 обладает полносвязная модель, при этом время одной эпохи обучения и валидации является - 3 секунды. Более точной моделью (0.9065, 0.9037, 0.8999) является одномерная сверточная модель, причем время эпохи обучения и валидации модели такое же - 3 сек. Лидерами по точности, очевидно являются рекуррентные модели: модель GRU - (0.9485, 0.9358, 0.9336), модель LSTM - (0.9617, 0.9487, 0.9485). При этом по времени эпохи обучения и валидации модель GRU достаточно выигрывает у LSTM - 26 < 31 сек. Поэтому делаем вывод о лидерстве LSTM модели с точки зрения точности в данной серии экспериментов. Если же точность модели достаточно иметь больше 0.9 и важен фактор времени обучения и легковесности модели, то более привле-

кательной является сверточная модель 1D CNN. Увеличение числа слоев и нейронов в слоях моделей к увеличению качества моделей не привело, зачастую наблюдался обратный эффект выхода на плато и понижения точности в процессе обучения.

Дальнейшей целью исследований было с одной стороны повысить точность модели, с другой стороны уменьшить ее время обучения и валидации, то есть получить более легковесную модель по сравнению с рекуррентными. Для этого были последовательно соединены три блока слоев: сверточный 1D CNN, рекуррентный GRU или LSTM и в итоге как дискриминантный классификатор полносвязный блок:

```
Z = Conv1D(filters=512, kernel size=4,
activation='relu') Xi
Z = Conv1D(filters=512, kernel size=4,
activation='relu') (Z)
Z = Conv1D(filters=512, kernel size=4,
activation='relu') (Z)
Z = Pooling1D(2) (Z)
Z = Conv1D(filters=256, kernel size=2,
activation='relu') (Z)
Z = Conv1D(filters=256, kernel size=2,
activation='relu') (Z)
Z = Conv1D(filters=256, kernel size=2,
activation='relu') (Z)
Z = RNN(units=64) (Z)
Z = Dense(2, activation='softmax') (Z)
```

На основе данной архитектуры были получены несколько нейросетевых моделей путем использования в слое агрегации методов AveragePooling и MaxPooling, в рекуррентных слоях ячейки типа GRU и LSTM. Рассматривался входной временной ряд с исходными и нормированными значениями в диапазоне от 0 до 1, функцией MinMaxScaler. Обучение проводилось также на 500 эпохах, но при этом использовался механизм раннего останова в случае выхода на плато значения точности валидации в течение 10 итераций. Как показали эксперименты в этом случае обучение выполнялось не более 160 эпох. В рекуррентном слое применялась $l1$ и $l2$ регуляризация.

Как показали данные анализа, по точности и времени одной эпохи обучения и валидации в данной группе моделей с небольшим преимуществом лидирует модель с параметрами AveragePooling, GRU и показателями точности на обучающем, валидационном и тестовом этапах: 0.9850, 0.9668, 0.9661, и временем одной эпохи обучения и валидации 14 сек.

Следующая группа моделей строилась на основе архитектуры предыдущей модели и методики добавления остаточных связей, развитие которой началось с появления семейства сетей ResNet [5]. Также было увеличено число сверточных слоев и уменьшено число фильтров в них.

```
Z1 = Conv1D(filters=64, kernel size=4,
```

```
activation='relu') (Xi)
```

```
Z1 = Conv1D(filters=64, kernel size=4,
activation='relu') (Z1) * 9 слоев
```

```
Z2 = add([Z1, Xi]) – пропуск остаточной связи
входа Xi
```

```
Z2 = Pooling1D(2) (Z2)
```

```
Z2 = Conv1D(filters=64, kernel size=2,
activation='relu') (Z2) * 9 слоев
```

```
Z3 = Conv1D(filters=64, kernel size=2,
activation='relu') (Z2)
```

```
Z4 = Pooling1D(2) (Xi)
```

```
Output = add([Z2, Z3, Z4]) – пропуск остаточных
связей Z2 и Z3
```

```
Output = Dense(32, activation='relu') (Output)
```

```
Output = Dense(2, activation='softmax') (Output)
```

По данным экспериментов, исходя из соотношения значения точности и функции потерь, на этапах обучения, валидации и тестирования, а также времени одной эпохи обучения и валидации лидером является модель с параметрами: AveragePooling, GRU, нормированные данные с показателями точности на обучающем, валидационном и тестовом этапах: 0.9821, 0.9665, 0.9690, и временем обучения эпохи 12 сек.

Также исследовалась группа моделей на основе архитектуры семейства Inception [5]. К сожалению, прироста в значениях точности и уменьшения значений потерь не произошло. При этом время одной эпохи обучения и валидации возросло более чем в 2 раза.

Машинный анализ разработанных моделей показал преимущество нейросетевых глубоких моделей, представляющих собой последовательное соединение трех блоков слоев: сверточного 1D CNN, рекуррентного GRU и итогового полносвязного блока- классификатора, с применением слоя агрегации AveragePooling, метода пропуска остаточных связей семейства архитектур ResNet и нормирования исходного набора данных в силу небольшого преимущества по точности на этапе тестирования и времени обучения и валидации. Также хорошо себя показала подобная модель без остаточных данных и нормирования входных данных, но с немного более длительным временем эпохи обучения и валидации.

1. ГОСТ РО 1410-002-2010. Ракетно-космическая техника. Система информации о техническом состоянии и надежности космических комплексов и входящих в их состав изделий.
2. Yang Q. 10 challenging problems in data mining research /Yang Q., Wu X. // Inf. Technol. Decis. Mak. – 2006. – 05(04). – pp.597–604.
3. Hassan Ismail Fawaz, Deep learning for time series classification: a review /Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, Pierre-Alain Muller // Data Mining and Knowledge Discovery. – 2019. – 33. – pp.917–963.
4. Скобцов В. Ю. Нейросетевой анализ данных телеметрической информации бортовой аппаратуры космических аппаратов. /Скобцов В. Ю., Архипов В. И. // Космическая техника и технологии. – 2021. – С.111-124.
5. Шолле Ф. Глубокое обучение на Python / Шолле Ф. // Издательство: МЦНМО, 2006. – 448 с.