УДК 136.88:004.942

R.R. MUKHAMETZYANOV
KSPEU, Kazan, Republic of Tatarstan
Scientific advisor, PHD associate professor, G. Z. GILYAZIEVA, KPEU, Kazan

NEURAL NETWORK FOR DETERMINING THE STATE OF LI – ION BATTERIES

Abstract: This comprehensive exploration delves deeply into the intricacies of deploying a sophisticated two-layer neural network, serving as an advanced and indispensable tool for the meticulous monitoring and predictive analysis of battery conditions. The versatile application of this program spans a broad spectrum, ranging from its pivotal role in the dynamic realm of electric vehicles to its strategic integration within power storage systems relying on cutting-edge lithium-ion battery technology.

The ubiquity of rechargeable batteries in contemporary society, powering everything from electric vehicles to portable electronic devices and renewable energy storage, underscores their pivotal role in our daily lives. However, the omnipresence of these energy storage units is not without its challenges, with issues such as inhomogeneous cell wear and the potential failure of specific cells taking center stage [1]. A nuanced portrayal of the intricate cell structure of a battery is meticulously illustrated in Figure 1, emphasizing the need for advanced monitoring and predictive tools.

Fig. 1. The cell structure of a battery is shown

As the global demand for rechargeable batteries continues its meteoric rise, the imperative for sustainable methods and innovative technologies for their recycling and reuse becomes increasingly pronounced. Concurrently, a deeper understanding of the multifaceted reasons contributing to uneven battery cell utilization and failures emerges as a critical area of focus [2].

To address this complex landscape and propel the optimization of battery cell utilization to new heights, a sophisticated program grounded in a two-layer neural network paradigm has been meticulously developed. The intricacies of its architecture and functionality are visually depicted in the elucidating Figures 2a and 2b [3]. This program, developed using the versatile Python programming language, harnesses the power of widely adopted libraries such as pandas, numpy, and sklearn. These libraries contribute to seamless data handling, precise numerical operations, efficient array manipulation, and the implementation of cutting-edge machine learning algorithms [4]. The utilization of the StandardScaler class emerges as a pivotal component, ensuring meticulous data normalization and, by extension, the accurate processing of information.

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
# Загрузка данных
data = pd.read_csv('battery_dataset.csv')
# Удаление столбцов, которые не будут использоваться в модели
data.drop(['cycle', 'time'], axis=1, inplace=True)
# Разделение на тренировочный и тестовый наборы
train_data, test_data = train_test_split(data, test_size=0.2, random_state=42)
# Нормализация данных
scaler = StandardScaler()
train_data = scaler.fit_transform(train_data)
test_data = scaler.transform(test_data)
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout
# Определение архитектуры нейросети
model = Sequential()
model.add(Dense(64, activation='relu', input_shape=(train_data.shape[1],)))
model.add(Dropout(0.2))
model.add(Dense(32, activation='relu'))
model.add(Dropout(0.2))
model.add(Dense(1, activation='sigmoid'))
# Компиляция модели
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
```

```python
# Обучение модели
history = model.fit(train_data, train_labels, epochs=100, batch_size=64, validation_split=0.2)
# Предсказание на тестовых данных
predictions = model.predict(test_data)
# Сохранение результатов в файл
output = pd.DataFrame({'prediction': predictions.reshape(-1)})
output.to_csv('predictions.csv', index=False)
# Предсказание на тестовых данных
predictions = model.predict(test_data)
# Сохранение результатов в файл
output = pd.DataFrame({'prediction': predictions.reshape(-1)})
output.to_csv('predictions.csv', index=False)

#Загрузка итоговых данных
final_data = pd.read_csv('final_battery_data.csv')
#Удаление столбцов, которые не будут использоваться в модели
final_data.drop(['cycle', 'time'], axis=1, inplace=True)
#Нормализация данных
final_data = scaler.transform(final_data)
#Предсказание результатов на итоговых данных
final_predictions = model.predict(final_data)
#Анализ результатов
final_output = pd.DataFrame({'prediction': final_predictions.reshape(-1)})
final_output.hist(bins=20)
#Сохранение результатов в файл
final_output.to_csv('final_predictions.csv', index=False)
```

Fig. 2 Program code listing

However, the efficacy of this program hinges on the existence of a well-curated database, a repository rich in detailed information pertaining to each individual battery cell. Critical parameters, including but not limited to the number of charge-discharge cycles, operating time, temperature fluctuations, capacity metrics, voltage profiles, and operational currents, form the bedrock of this comprehensive database. The structured representation of these parameters within a matrix, as exemplified in Figure 3, further underscores the meticulous approach undertaken.

```
cycle, time, temperature, voltage, current, capacity
1,0,25,3.744,0.0,1.8563
1,1,26,3.748,0.0,1.8563
1,2,27,3.752,0.0,1.8563
1,3,28,3.756,0.0,1.8563
2,0,25,3.743,0.0,1.8564
2,1,26,3.747,0.0,1.8564
2,2,27,3.751,0.0,1.8564
2,3,28,3.755,0.0,1.8564
```

Fig.3. Massive example

At the core of the developed program lies the primary objective of generating a multifaceted array encapsulating exhaustive information regarding the status of each battery cell and its potential for continued use. Figure 4 serves as an illustrative vignette, offering a glimpse into the practical operation of the program. This innovative methodology proves instrumental in the early identification of cells susceptible to inhomogeneous wear or potential failure, enabling timely interventions such as replacement or repair.

Fig.4. Example of output data

In the realm of practical application, the implementation of this cutting-edge program transcends mere optimization; it becomes a cornerstone for extending the service life of batteries across diverse sectors. It is paramount to acknowledge that non-uniform cell wear, a pervasive issue, stems from a myriad of factors, including but not limited to uneven load distribution, suboptimal operating conditions, or inherent defects within the cells themselves. This comprehensive exploration seeks not only to address these challenges but also to provide a robust framework for ushering in a new era of sustainable and efficient battery utilization.

REFFERENCES:

1. *Евдокимов Ю.К.* Современные литий-ионные аккумуляторы. Батареи на их основе// Компоненты и технологии. 2015. № 5. С. 81–85.

2. *Губенков Андрей Олегович ЭЛЕКТРОМОБИЛИ: ГАРАНТИЯ ЭКОЛОГИЧЕСКОЙ БЕЗОПАСНОСТИ ИЛИ МИФ? УТИЛИЗАЦИЯ ЛИТИЙ-ИОННЫХ АККУМУЛЯТОРОВ ЭЛЕКТРОМОБИЛЕЙ ПРОБЛЕМА ЭКОЛОГИИ ИЛИ СОВРЕМЕННОЙ ПРОМЫШЛЕННОСТИ?//Автономия личности. 2022. №1 (27). URL: https://cyberleninka.ru/article/n/elektromobili-garantiya-ekologicheskoy-bezopasnosti-ili-mif-utilizatsiya-litiy-ionnyh-akkumulyatorov-elektromobiley-problema (дата обращения: 18.05.2023).*

3. Свидетельство о государственной регистрации программы для ЭВМ 2023660200 Российская Федерация. Программа для мониторинга и прогнозирования состояния аккумуляторных батарей электромобилей с использованием двухслойной нейронной сети/ П. П. Павлов, А. Н. Хуснутдинов, Р.С. Литвиненко, В. М. Бутаков, Л.Н. Киснеева, Р.Р. Мухаметзянов, Д.О. Марков; заявитель и правообладатель Федеральное государственное бюджетное образовательное учреждение высшего образования

«Казанский государственный энергетический университет». – № 2023660200 ; заявл. 28.04.2023 ; опубл 18.05.2023.

4. Альбовский Александр Владимирович, Егоров Никита Андреевич, Романюк Анастасия Георгиевна Реализация нейронной сети с помощью языка программирования Python // Colloquium-journal. 2020. №9 (61). URL: https://cyberleninka.ru/article/n/realizatsiya-neyronnoy-seti-s-pomoschyu-yazyka-programmirovaniya-python (дата обращения: 18.05.2023).