Gökhan Laçin 21MECT1001 28.03.2025

ENGR4450 Aplied Artifical Inteligence
Homework 2 Report

1. FFNN Prediction of SoC and SoH

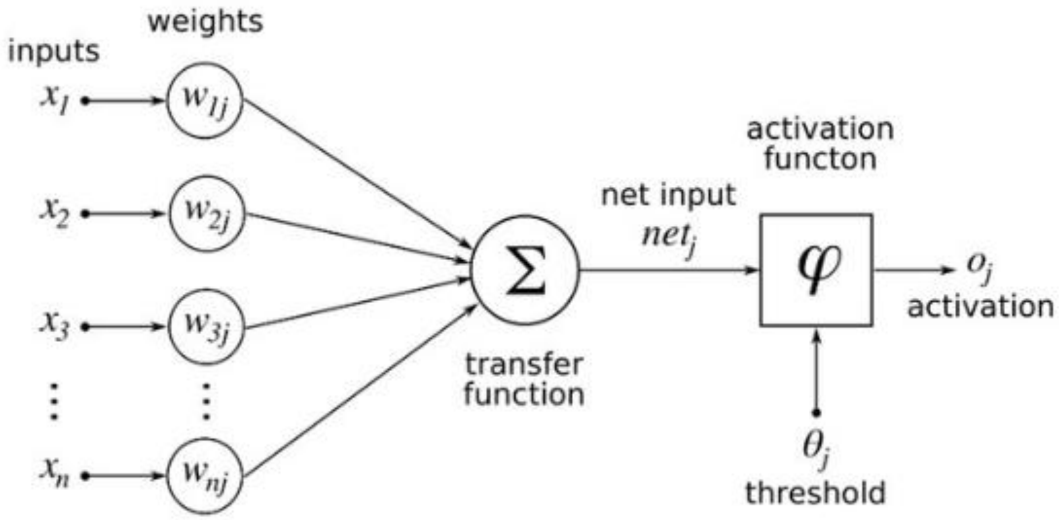


Figure 1. Structure of a single neuron in a feedforward neural network [1]

Artificial Neural Networks (ANNs) have become essential tools for solving complex problems in pattern recognition and machine learning. Among the most fundamental architectures is the Feedforward Neural Network (FFNN), which is widely used in supervised learning tasks such as classification, regression, and function approximation.

A Feedforward Neural Network is composed of layers of interconnected neurons, where data flows in one direction only: from the input layer, through one or more hidden layers, and finally to the output layer. Unlike recurrent neural networks, FFNNs do not contain feedback loops or cycles, which makes their structure simple and efficient for forward data propagation.

The diagram above illustrates the inner workings of a single neuron within an FFNN. Each neuron receives multiple input values, labeled as x1, x2, ..., xn. Each input is multiplied by an associated weight, denoted w1j, w2j, ..., wnj. These weighted inputs are summed to produce a value known as the net input (net j). This can be written as:

$$\textbf{net j = (x1 * w1j) + (x2 * w2j) + ... + (xn * wnj)}$$

Optionally, a threshold value (theta j) may be subtracted from this sum to regulate neuron activation. The resulting value is then passed through an activation function, such as the sigmoid or ReLU function, which determines whether the neuron "fires" and produces an output. This output is then transmitted to the next layer of the network or used as a final result.

Thanks to its feedforward structure and the ability to adjust weights during training, the FFNN is capable of learning from data and making accurate predictions. These features make FFNNs a foundational model in the fields of artificial intelligence and deep learning.

Yapay Sinir Ağları (Artificial Neural Networks – ANN), karmaşık desen tanıma ve makine öğrenmesi problemlerinin çözümünde güçlü araçlar olarak öne çıkmaktadır. Bu yapılar arasında en temel ve yaygın kullanılan mimarilerden biri İleri Beslemeli Sinir Ağı (Feedforward Neural Network – FFNN) olup, sınıflandırma, regresyon ve fonksiyon yaklaşımı gibi denetimli öğrenme görevlerinde etkin şekilde kullanılmaktadır.

İleri beslemeli sinir ağları, birbirine bağlı nöron katmanlarından oluşur ve veri akışı tek yönlüdür: giriş katmanından başlayarak bir veya birden fazla gizli katman üzerinden geçerek çıkış katmanına ulaşır. Geri besleme ya da döngü içermediğinden, yapısı sade ve veri iletimi açısından etkilidir.

Yukarıda verilen görsel, FFNN yapısı içindeki tek bir nöronun nasıl çalıştığını göstermektedir. Her bir nöron, x1, x2, ..., xn şeklinde ifade edilen birden fazla girdi alır. Bu girdilerin her biri, w1j, w2j, ..., wnj şeklinde gösterilen ağırlıklarla çarpılır. Ardından bu değerler toplanarak net giriş değeri (net j) elde edilir. Bu hesaplama, genellikle şu şekilde ifade edilir:

$$\textbf{net } j = (x1 \times w1j) + (x2 \times w2j) + \ldots + (xn \times wnj)$$

veya daha genel olarak, tüm girişlerin ağırlıklarıyla çarpılıp toplanmasıyla elde edilir.

Bu net girişten, varsa bir eşik değeri (theta j) çıkarılır. Elde edilen sonuç, aktivasyon fonksiyonu adı verilen ve genellikle doğrusal olmayan bir işlevden geçirilir. Bu fonksiyon (örneğin sigmoid ya da ReLU gibi), nöronun çıktısını belirler. Elde edilen çıktı, bir sonraki katmana aktarılır ya da çıkış olarak kullanılır.

İleri beslemeli yapısı ve eğitim sırasında ağırlıkların ayarlanabilmesi sayesinde FFNN'ler veriden öğrenme yeteneğine sahiptir ve doğru tahminlerde bulunabilir. Bu özellikleriyle FFNN'ler, yapay zekâ ve derin öğrenme uygulamalarında temel bir yapı taşıdır.

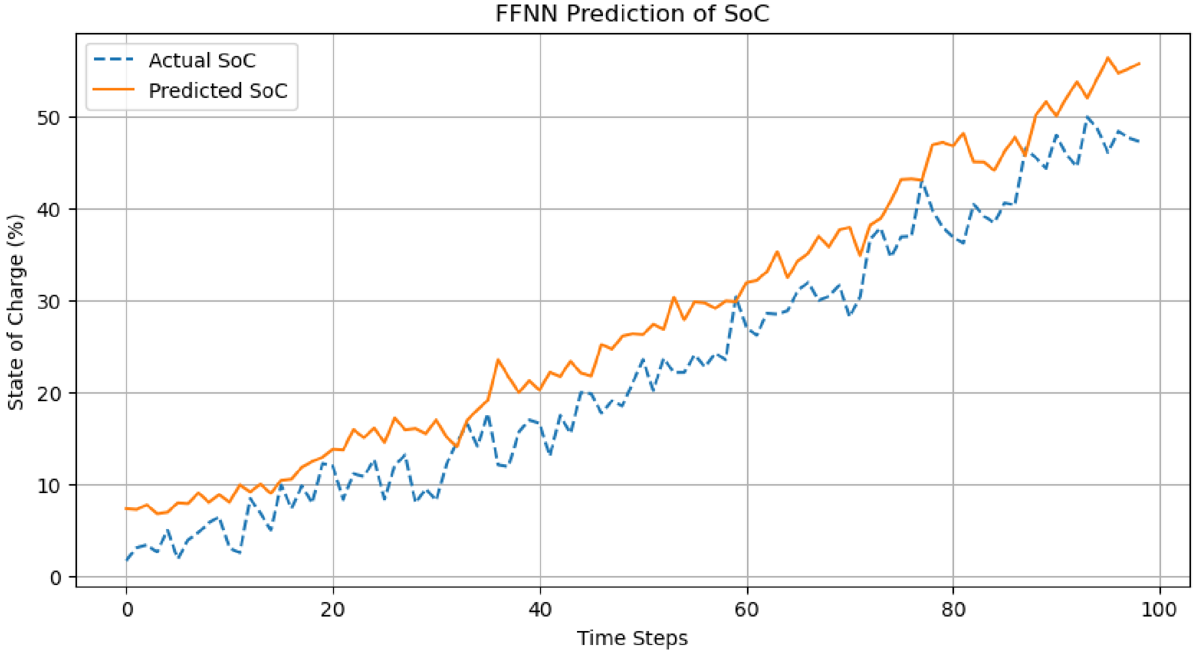
1.1 Engineering Context of the Code

Here, SoC (State of Charge) indicates how full the battery is, expressed as apercentage. In the automotive domain, it's a critical parameter for vehicle range, instantaneous performance, and energy management.

SoH (State of Health) shows the battery's aging/capacity loss condition. It decreases over time with charge-discharge cycles. It is important for long battery life and safe driving.

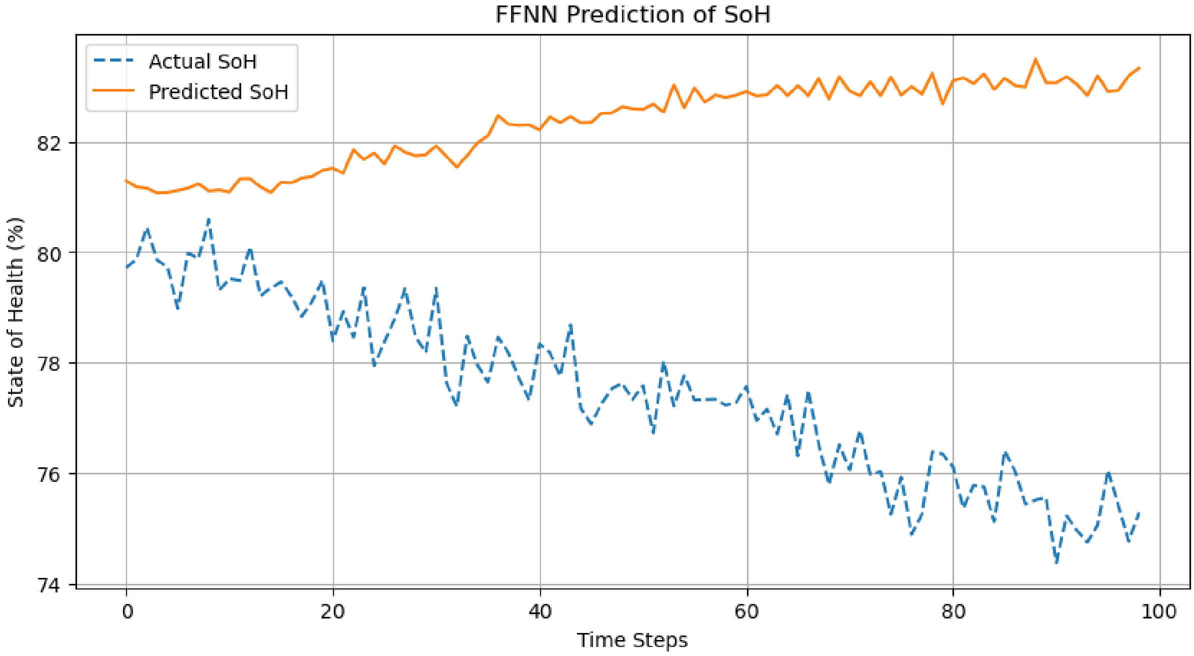
1.2 Prediction Graphs

In the graph, the orange line is “Predicted” (the model's forecast), and the blue dashed line is “Actual” (the real value).



***Graph 1**: Actual vs Predicted SoC Graphic [2]*

In the graph 1, the model captures the sine wave's fluctuation trend, but there are occasional discrepancies (especially at peaks and troughs).



***Graph 2**: Actual vs Predicted SoH Graphic [2]*

In the graph 2, the model predicts a higher tendency (around 85–88), whereas the actual value goes down to the 80–75 range. The model might be overestimating (bias).

1.3 Erol and Improvement Methods

Even though the MSE loss decreases, it may not be a perfect fit. One could try longer training, different numbers of layers, or different hyperparameters.

The dropout rate could be increased or decreased.

Other activation functions (for example, using Leaky ReLU instead of ReLU) could be tested.

1.4 Connection to Automotive Dynamics

Essentially, estimating SoC and SoH is critical for electric powertrains in EVs or hybrids, as it affects range management and battery power.

If the battery SoH is low, the vehicle could experience reduced torque distribution or less efficient regenerative braking on long drives, partially impacting the "friction circle" dynamics. For instance, if the motor power is insufficient, torque vectoring in a turn may not be as effective.

Also, as SoC decreases, less support is available from electric braking, forcing the vehicle to rely more on mechanical brakes, which in turn can affect tire-road friction differently.

2. What Is LSTM (Long Short-Term Memory)?

LSTM is a special type of Recurrent Neural Network (RNN) designed primarily for **time-series or sequential data**, where capturing long-term dependencies is crucial. It addresses the vanishing gradient problem often seen in standard RNNs by incorporating gate mechanisms (**input gate, forget gate, output gate**) and a **cell state**. These gates regulate how information flows in and out of the memory cell, allowing the network to retain or discard information as needed.

Typical Uses of LSTM

- **Time-series forecasting** (e.g., financial data, IoT sensors, automotive sensor data)
- **Natural Language Processing** (due to the sequential nature of text)
- **Speech recognition and audio processing**
- **Video analysis** (where consecutive frames have temporal dependencies)

2.1 Why Choose LSTM?

- **Time Series**: In problems such as **SoC (State of Charge)** and **SoH (State of Health)**, which vary over time based on vehicle usage, LSTM layers typically outperform simpler feed-forward models.

- **Long-Context Memory**: An LSTM can preserve relevant information from the previous 30 time steps (or more) of voltage, current, temperature, SoC, and SoH, thereby making more accurate future predictions than a basic feed-forward neural network that lacks recurrent connections.

2.2 Why Are SoC and SoH Important?

SoC (State of Charge)

- Represents how full the battery is, in percentage terms.
- Vital for vehicle range, performance, and overall energy management.

SoH (State of Health)

- Reflects the battery's aging or capacity loss.
- Highly relevant for long-distance trips, fleet vehicles, and any scenario where battery health significantly impacts operations.

2.3 Key Points in the Code

2.3.1 Normalization

- Scaling data to the 0–1 range is crucial for stable training, especially in LSTM networks.

2.3.2 Sequence Length (seq_length=30)

- Using the last 30 time steps lets the model incorporate short- to medium-term historical data.
- Increasing the sequence length can improve performance but also complicates training.

2.3.3 EarlyStopping, ReduceLROnPlateau

Useful techniques to avoid overfitting or unnecessarily long training.

If the validation loss plateaus or worsens, they can reduce the learning rate (ReduceLROnPlateau) or stop the training early (EarlyStopping).

2.3.4 Inverse Transformation

After prediction, reverting the normalized outputs back to the original 0–100 range is essential for interpreting real-world SoC and SoH values.

2.4 Automotive Dynamics Context

High or Low SoC

- Affects vehicle traction performance (acceleration, torque) and regenerative braking efficiency.

Low SoH

- Decreased battery life or capacity, reducing potential driving range and increasing risks during long journeys.

Temperature/Voltage/Current Fluctuations

- Critical for battery safety and efficiency. An LSTM can capture the temporal patterns of these parameters, thus improving predictive accuracy.

This code demonstrates how an **LSTM-based model** can predict SoC and SoH using synthetic data, applying best practices such as normalization, sequence creation, regularization (dropout, batch normalization), early stopping, and learning rate reduction (ReduceLROnPlateau).

- **Key Idea**: Feed time-based battery data (SoC, SoH, voltage, current, temperature) into an LSTM to forecast future SoC and SoH.
- **Engineering Use**: Real-time battery management systems (BMS) and condition monitoring.
- **Further Work**: Training on real data, adjusting sequence length or number of layers, adding additional sensors (e.g., internal resistance, driver behavior) are all possible enhancements.

LSTM vs. FFNN Comparison

Below is a brief overview of **LSTM (Long Short-Term Memory)** versus **FFNN (Feed-Forward Neural Network)**, along with a table of their advantages and disadvantages.

Structural Differences

- **FFNN (Feed-Forward Neural Network)**
  - Data flows from the input layer to the output layer **in a single forward direction**.
  - No direct temporal or sequential component; each input sample is processed independently.
- **LSTM (Long Short-Term Memory)**
  - Designed for **time series or sequential data**, using **recurrent connections** and a **cell state**.
  - Memory gates (forget, input, output) enable the network to maintain or discard information over long sequences.

Model	Advantages	Disadvantages
FFNN	- Simple structure, fast training - Good performance if there is no temporal dependence - Relatively easier hyperparameter tuning	- Not suitable for sequential data unless additional feature engineering is done - Cannot capture long-term dependencies directly - Lacks a recurrent mechanism
LSTM	- Strong at modeling long-term dependencies - Mitigates vanishing gradient issues - Excels in tasks with sequential/time-series data	- More complex, slower training - Higher parameter count, increased memory/compute demands - Tuning hyperparameters can be more difficult

○

REFERENCES

[1] Leong, Y. K., Chang, C.-K., Arumugasamy, S. K., Lan, J. C.-W., Loh, H.-S., Muhammad, D., & Show, P. L. (2018). Statistical design of experimental and bootstrap neural network modelling approach for thermoseparating aqueous two-phase extraction of polyhydroxyalkanoates. *Polymers, 10*(2), 132. https://doi.org/10.3390/polym10020132

[2] Laçin, G. (2025, March 28). *hw2_FFNN.ipynb* [Jupyter Notebook]. GitHub. https://github.com/gokhanlcn/FFNN-Predictions-SoC-and-SoH