Personalized Conputational Casual Model based on Neural Networks

Yilin Li

In this note, we would illustrate the inspiring progression on PCCMNN verson3. More specifically, focus on the approach we construct populational model of Alzheimer's Disease, and prospects for future research.

March 04, 2025

Contents

1. Background

There is a large amount of research which focus on constructing data-driven approaches to construct and parameterize causal models[1], while there is less amount of research for Alzheimer's disease (AD) biomarker trajectories.

In the article[2], Zheng et al proposed a polynomial-based approach to construct population and personalized models for AD biomarker trajectories and stage recognition without relying on prior hypothesis bias, where they found that the dynamics in the 4 variables, including 2 protein biomarkers: A_β, τ, the volume of brain: N and one's intellective behavior: the intelligence test scores, could be modeled by the following ODE system:

$$\begin{aligned}
\frac{\mathrm{d}A_\beta}{\mathrm{d}s} &= w_{10} + w_{11}A_\beta + w_{12}A_\beta^2; \\
\frac{\mathrm{d}T}{\mathrm{d}s} &= w_{20} + w_{21}T + w_{22}T^2 + w_{23}A_\beta + w_{24}A_\beta^2 + w_{25}A_\beta T; \\
\frac{\mathrm{d}N}{\mathrm{d}s} &= w_{30} + w_{31}N + w_{32}N^2 + w_{33}T + w_{34}T^2 + w_{35}TN; \\
\frac{\mathrm{d}C}{\mathrm{d}s} &= w_{40} + w_{41}C + w_{42}C^2 + w_{43}N + w_{44}N^2 + w_{45}NC
\end{aligned} \tag{1}$$

For briefness, this system could be described as:

$$\begin{aligned}
\frac{\mathrm{d}r}{\mathrm{d}s} &= p(r); \\
r &:= \left(A_\beta, \tau, N, C\right)
\end{aligned} \tag{2}$$

In this article, we would replace polynomial function p with 4 neural network functions:

$$\begin{aligned}
\frac{\mathrm{d}A_\beta}{\mathrm{d}s} &= f_1\left(A_\beta\right); \\
\frac{\mathrm{d}T}{\mathrm{d}s} &= f_2\left(A_\beta, \tau\right); \\
\frac{\mathrm{d}N}{\mathrm{d}s} &= f_3(\tau, N); \\
\frac{\mathrm{d}C}{\mathrm{d}s} &= f_4(N, C)
\end{aligned} \tag{3}$$

2. Data Reduction

To process data efficiently, we first apply data reduction:

1. Using dataset: ANDI Org. and CSF Biomarker
2. Calculating each patient's age at the drwdate, keeping one decimal place
3. Combining data of the same patient and age to one data, i.e. collecting A_β, τ, N and C data of same patient and age together.
4. Similar to process in [2], removing samples which do not provide at least two measurements for any one of the four biomarkers.
5. Calculating each biomarkers' population 5 quartile x_5 and 95 quartile x_{95}, conducting normalize based on 5 and 95 percentile:

$$\hat{x} = \frac{x - x_5}{x_{95} - x_5} \tag{4}$$

where $\hat{x}$ denotes the normalized data.

3. Model Construction

3.1. DPS

Similar to the process in article [2], Page 9, equation(3), we apply linear regression to each patient's age:

$$s_i = a_i t_i + b_i \tag{5}$$

where t denotes patient i's age vector (age of each data points) and s denotes the **DPS (Disease progression scores)**.

When constructing A_β's model, we set $a_i = 0.1$ and $b_i = -10$ as the initial value. In the training procedure, a_i and b_i will be optimized to each samples. The initial value a_i and b_i may make a great influence to the model behavior, so we would discuss their influence in dynamical systems in the future.

Before training τ, N and C's model, we set A_β, τ, A_β's corresponding training result as a_i and b_i's initial values, respectively. Therefore, each model's corresponding parameter set would be different.

In [2], they restricted DPS score in $[-10, 20]$. In our work, we does not restrict as this, but we would calculate the average number of indices where DPS exceeds the limit during training process, and put a little penalty to a_i and b_i.

3.2. Training Process

3.2.1. Training A_β Model

1. Initialization
   - For each patient i, obtain the A_β data $A_{\beta i}$.
   - Set initial value $A'_{\beta i0} = A_{\beta i0}$ as input to model f_1.
2. Forward Euler Iteration
   - For each time step s_{ij}:
     1. Compute the approximate derivative $f_1\left(A'_{\beta i,j-1}\right)$.
     2. Predict $A'_{\beta ij}$ using the Euler method: $A'_{\beta ij} = A'_{\beta i,j-1} + \left(s_{ij} - s_{i,j-1}\right) f_1\left(A'_{\beta i,j-1}\right)$
   - Repeat until predictions for all s_{ij} are obtained.
3. Loss Calculation and Parameter Update
   - Compute the L^2 loss between predicted $A'_{\beta i}$ and actual $A_{\beta i}$.
   - Perform backpropagation and gradient descent to update parameters a_i, b_i.
4. Global Optimization
   - Repeat Steps 1-3 for all patients.
   - Compute the average L^2 loss across all patients.
   - Update f_1 parameters using gradient descent.
5. Convergence Check
   - Repeat training until the loss converges or a stopping criterion is met.

3.2.2. Training τ, N, C Model

It's important to recognize that for the same patient, their time line s_i can vary across the four variants. This occurs because for each variant, we remove missing values, which also adjusts the time points in s_i. Therefore, when predicting τ, N, and C values, we must predict A_β, τ, and N values on their respective new time lines.

Specifically:

1. Variable Time Line Differences:
   - Due to potential differences in missing values across variables like τ, N, C, and A_β in the raw data, the time line s_i for each of the four variants can be different for the same patient i.
2. Prediction Dependencies:
   - Predicting τ, N, and C values requires using A_β values at the same time points.
   - Directly using the original A_β time series can lead to time point mismatches.
3. Re-predicting former variants:
   - To address time point mismatches, we must first use the former trained model to predict former variants' values on the respective time lines of τ, N, and C.
   - For example, when predicting N, we need to re-predict τ on the N time line, thus we need to predict A_β on the same time line.

Here is the algorithm to train τ's model f_2, same for N and C.

1. Initialization
   - For each patient i, obtain time line s_i and A_β initial value $A_{\beta i}$. Predicting $A'_{\beta i,j}$ value at each time point s_{ij} with A_β's model.
   - Obtain the τ data τ_i.
   - Set initial value $\left(A'_{\beta i0}, \tau'_{i0}\right) = \left(A'_{\beta i,0}, \tau_{i,0}\right)$ as input to model f_2.
2. Forward Euler Iteration
   - For each time step s_{ij}:
     1. Compute the approximate derivative $f_2\left(A'_{\beta i,j-1}, \tau_{i,j-1}\right)$.
     2. Predict τ'_{ij} using the Euler method: $\tau'_{ij} = \tau'_{i,j-1} + \left(s_{ij} - s_{i,j-1}\right) f_2\left(A'_{\beta i,j-1}, \tau'_{i,j-1}\right)$
   - Repeat until predictions for all s_{ij} are obtained.
3. Loss Calculation and Parameter Update
   - Compute the L^2 loss between predicted τ'_i and actual τ_i.
   - Perform backpropagation and gradient descent to update parameters a_i, b_i.
4. Global Optimization
   - Repeat Steps 1-3 for all patients.
   - Compute the average L^2 loss across all patients.
   - Update f_2 parameters using gradient descent.
5. Convergence Check
   - Repeat training until the loss converges or a stopping criterion is met.

3.2.3. Training Settings

For each model's training process, we use same training settings as shown below:

Neural Network Architecture	1-32-32-1
Activation Functions	ReLU
Neural Network initialization	Kaiming
Optimizer of Neural Network	Adam
Optimizer of a and b	Adam
Learning rate of 2 optimizers	0.01
Epoch	200

4. Results

In this section, for each variants:

1. The first graph includes result in which we sampled 6 patients who has data with at least 6 time points randomly, predicted their time series of the variant with our model and compare with real data.
2. The second graph includes loss curve during training. The ab loss describes average number of indices where DPS exceeds the limit: $[-10, 20]$.
3. The third graph includes graph of model f_k with polynomial model p_k in the [2], Page4, Table1.

4.1. A_β Results

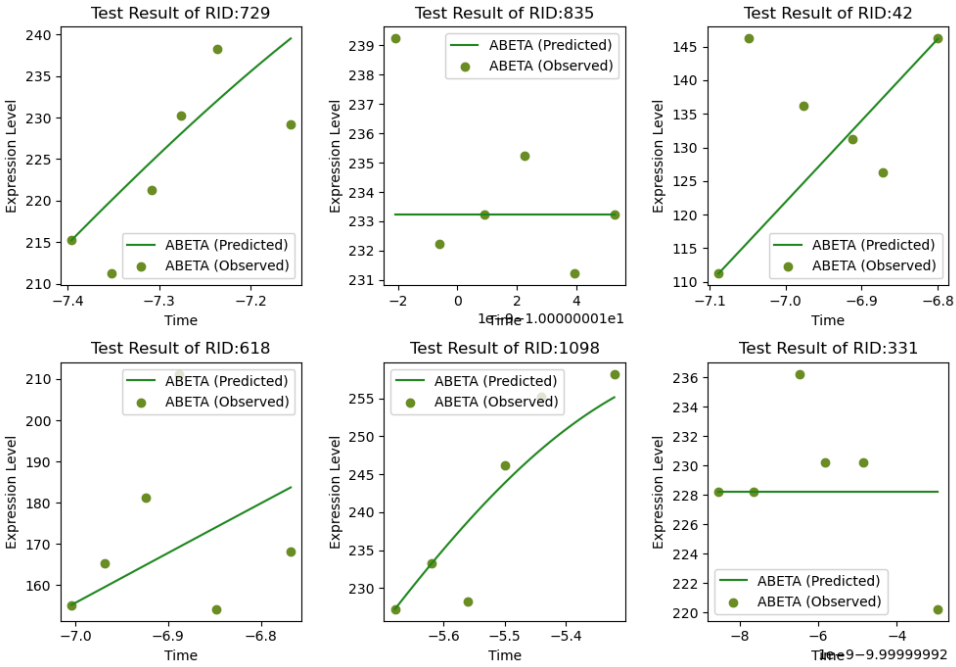

Figure 1: A_β model's behavior

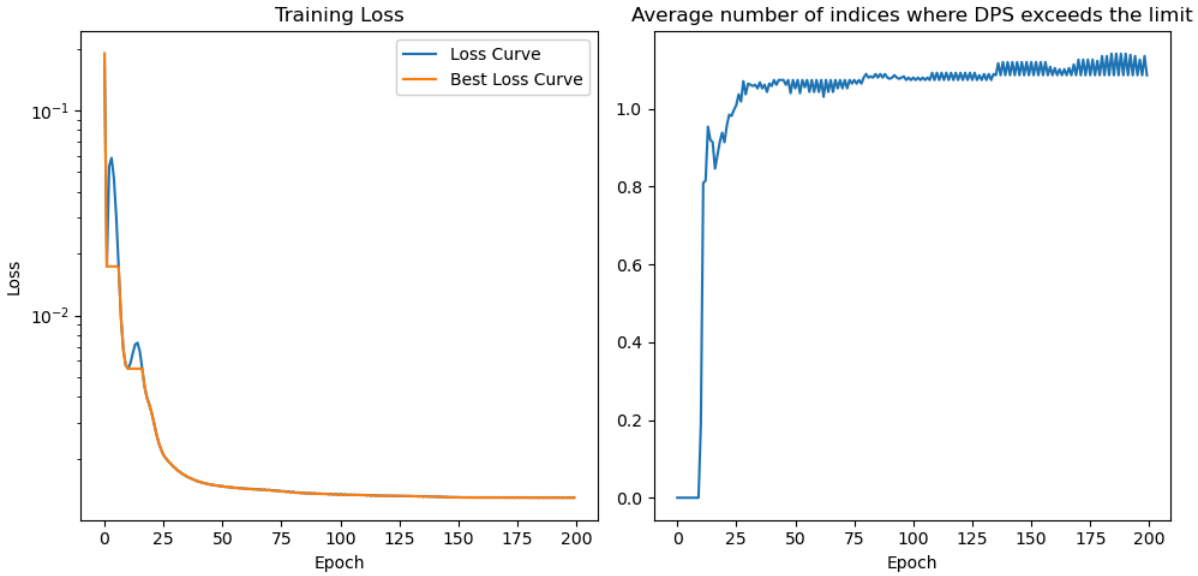

Figure 2: Training loss of A_β

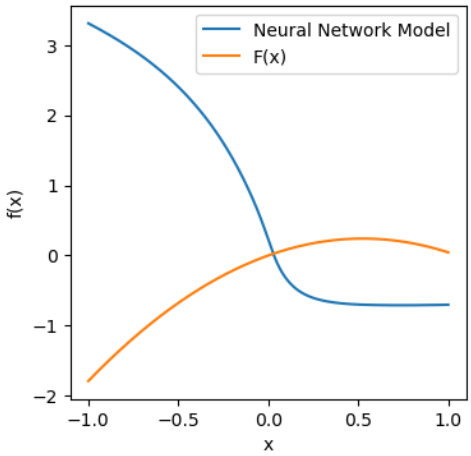

Figure 3: Graph of f_1 and p_1

Here $p_1 = 0.917A_\beta - 0.873A_\beta^2$.

4.2. τ Results

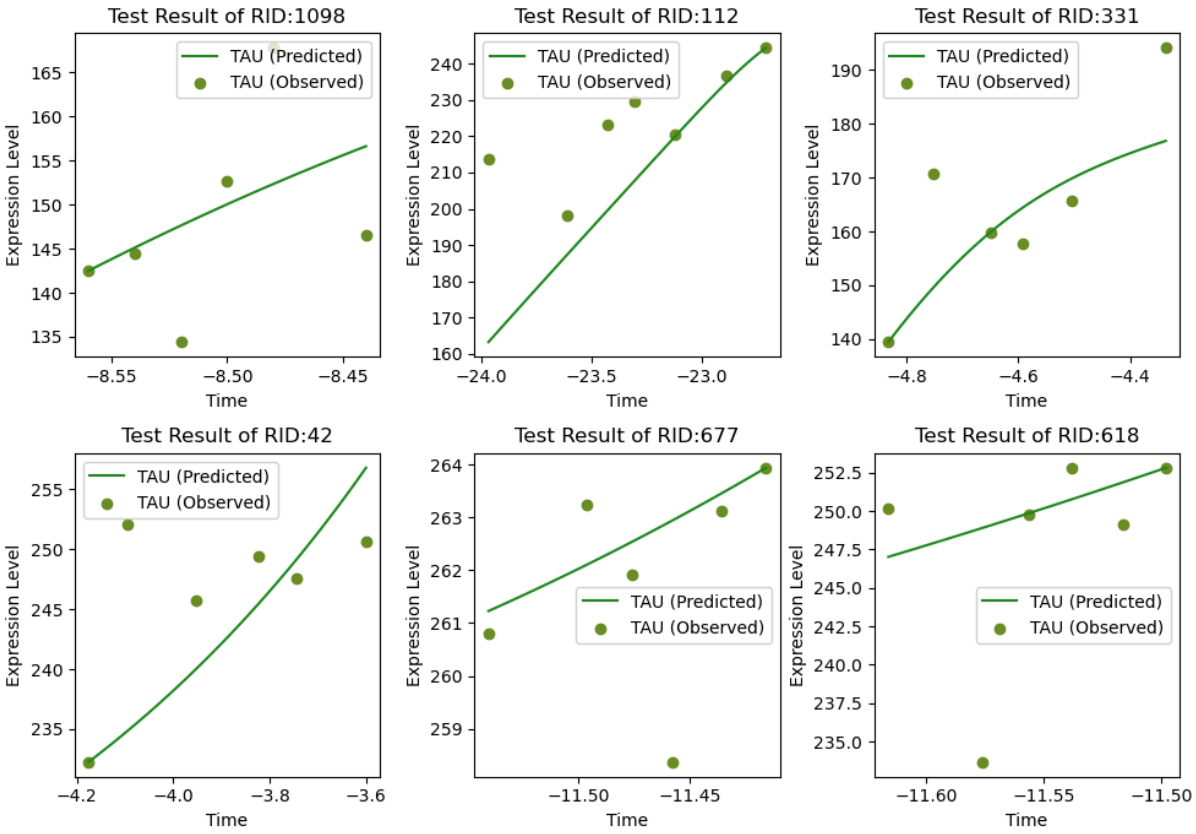

Figure 4: τ model's behavior

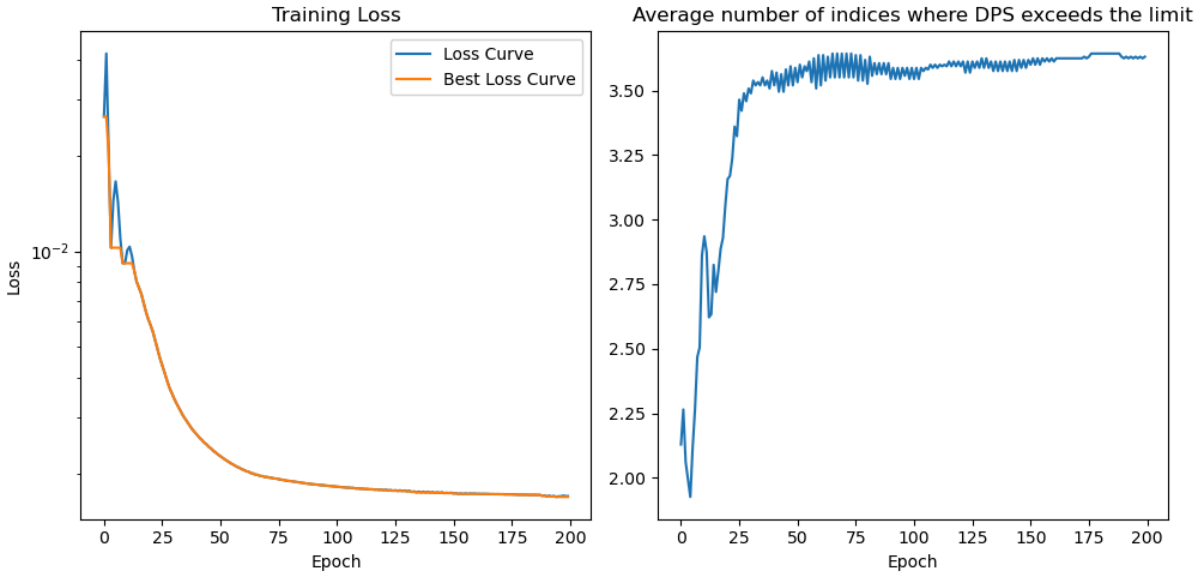

Figure 5: Training loss of τ

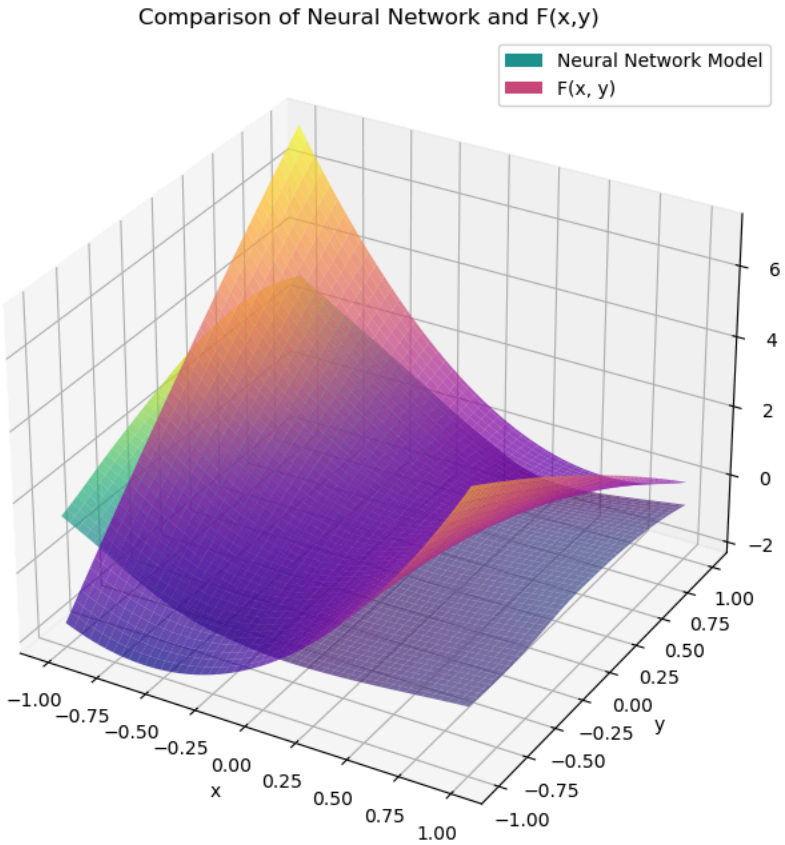

Figure 6: Graph of f_2 and p_2

Here $p_2 = 0.788\tau - 0.246\tau^2 + 0.002A_\beta + 3.066A_\beta^2 - 3.650A_\beta\tau$

4.3. N Results

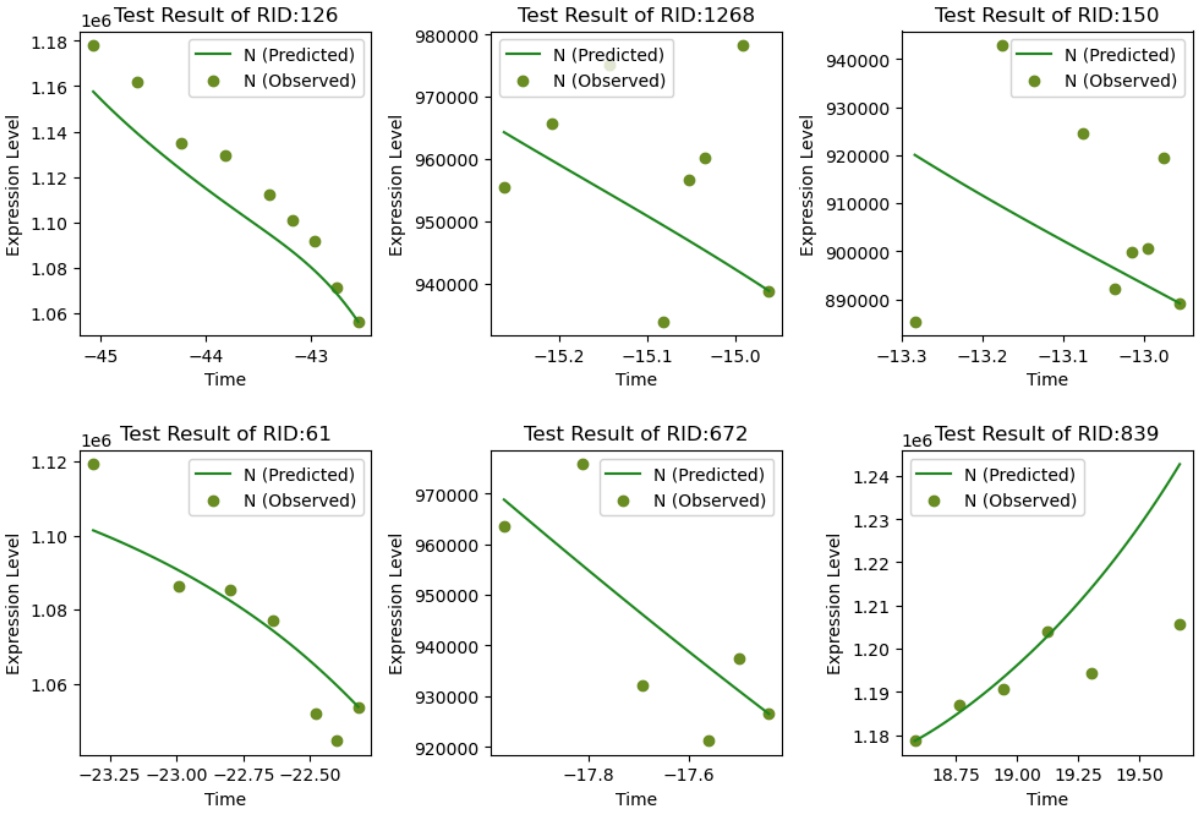

Figure 7: N model's behavior

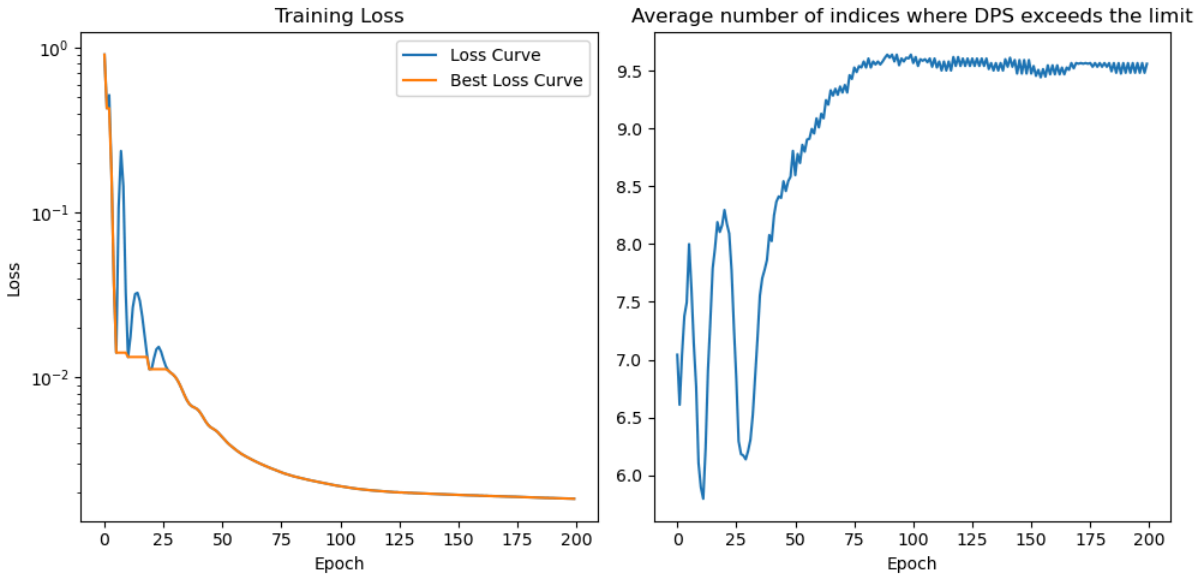

Figure 8: Training loss of N

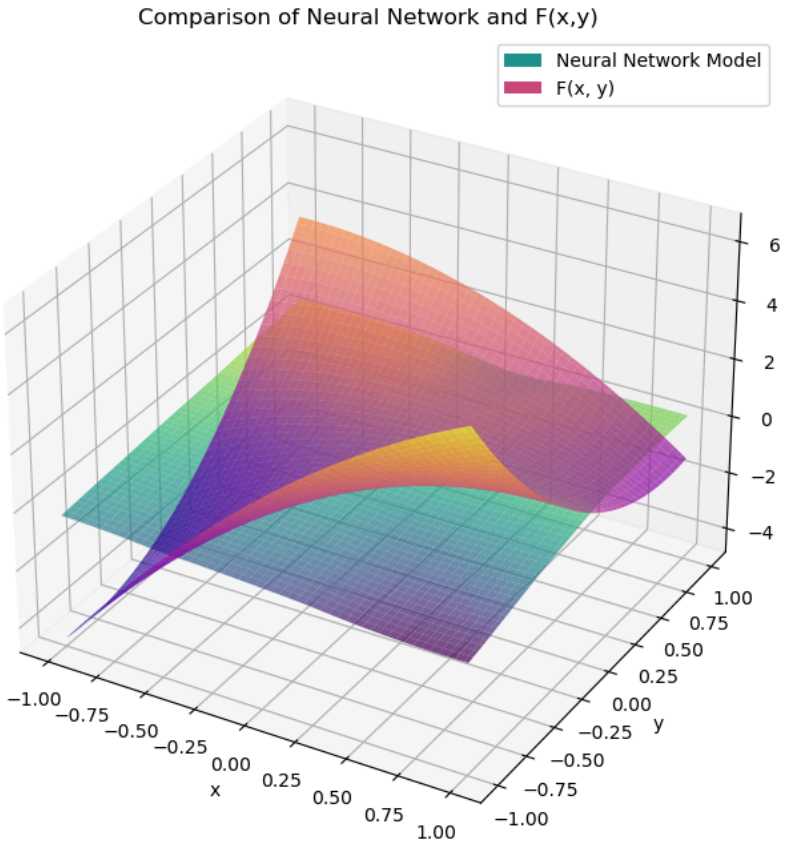

Figure 9: Graph of f_3 and p_3

Here $p_3 = 1.627N - 1.253N^2 + 0.018\tau + 2.342\tau^2 - 4.015\tau N$

4.4. C Results

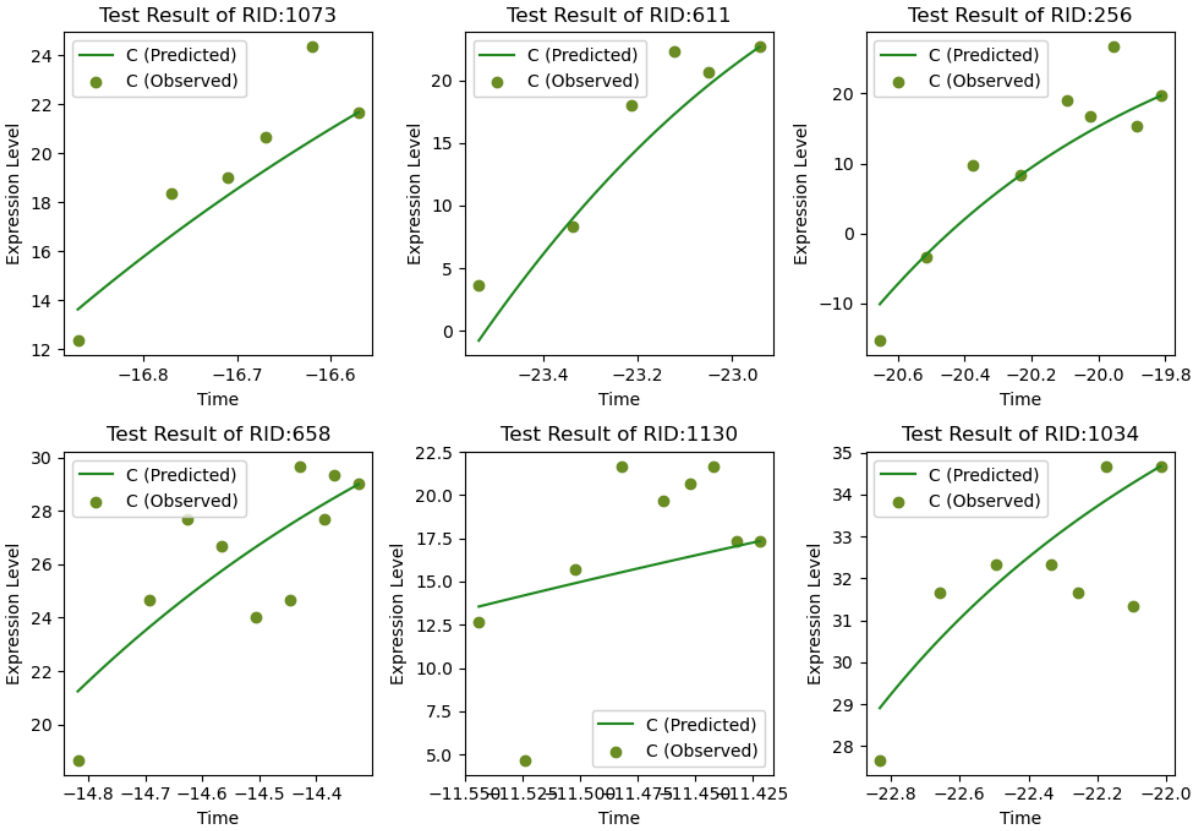

Figure 10: C model's behavior

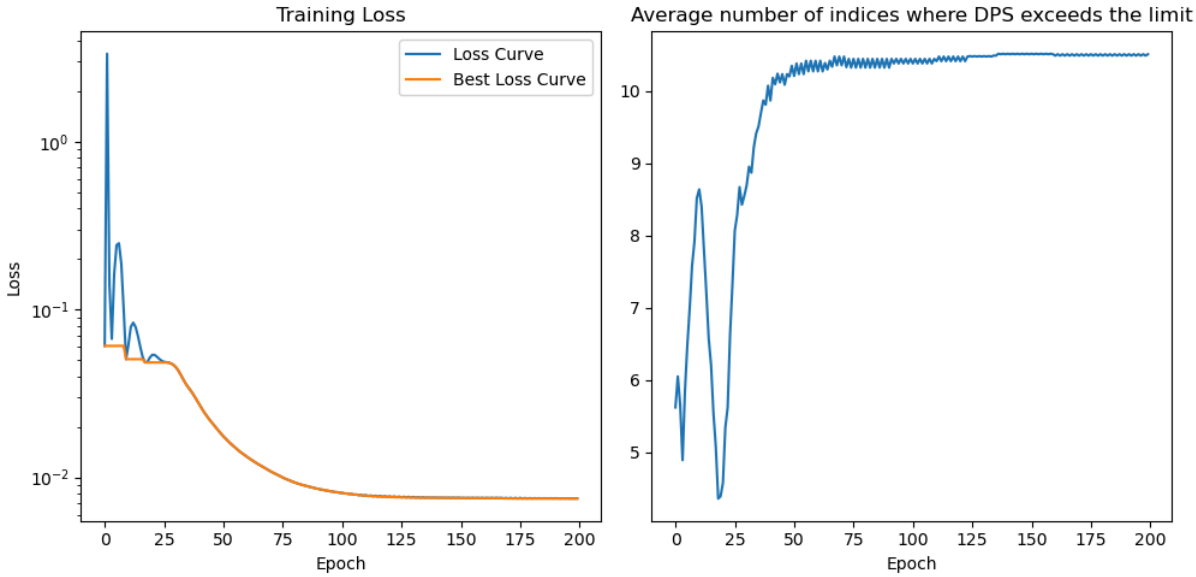

Figure 11: Training loss of C

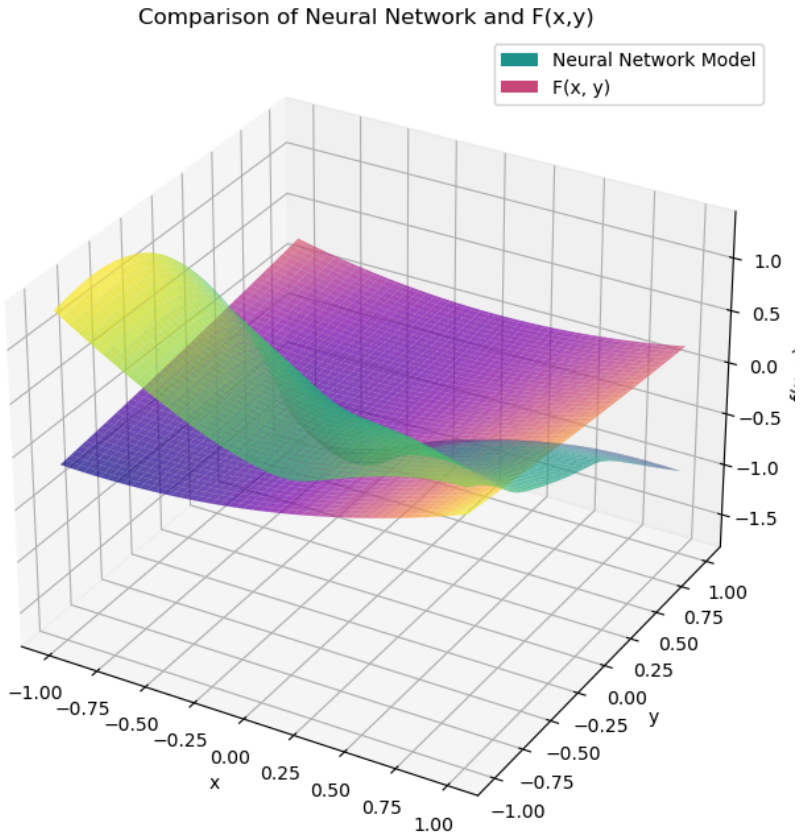

Figure 12: Graph of f_4 and p_4

Here $p_4 = 0.159C + 0.202C^2 + 0.010N + 0.019N^2 - 0.176NC$

5. Prospective Research Aspects

1. Constructing personalized model based on populational model, with technique of transfer learning, and recognizing disease stages.
2. Improve integration precision, changing Euler method to Runge-Kutta methods for example.
3. Change loss function: approximating numerical derivative of data
4. Compare model efficiency with NeuralODE
5. Conducting purning and sparse, reducing model scale
6. Getting rid of the priori hypotheses: the dependency chain of $A_\beta \to \tau \to N \to C$, developing a general approach to modeling disease progress and mechanism.

Bibliography

[1] J. Petrella, J. Jiang, K. Sreeram, S. Dalziel, P. Doraiswamy, and W. Hao, "Personalized Computational Causal Modeling of the Alzheimer Disease Biomarker Cascade," *J Prev Alzheimers Dis*, vol. 11, no. 2, pp. 435–444, 2024, doi: 10.14283/jpad.2023.134°.

[2] H. Zheng, J. Petrella, P. Doraiswamy, and others, "Data-driven causal model discovery and personalized prediction in Alzheimer's disease," *npj Digital Medicine*, vol. 5, p. 137, 2022, doi: 10.1038/s41746-022-00632-7°.

Index of Figures