Neural Network Model for Alzheimer's Disease

Yilin Li

In this note, we would illustrate the progression on ADNN populational model. More specifically, focus on the model structure and their behaviors.

August 14, 2025

Contents

1. Introduction

The main focus is to solve the problem that the trajectory function diverges at both ends and does not exhibit an S-shaped curve.

This attempt explored four architectures:

1. PINN + FNN
2. GRU-RNN
3. LSTM
4. ResNet+Fine Tunning

2. PINN + FNN

The neural network is an ordinary FNN with two hidden layers, using Tanh or ReLU as activation at first, then Sigmoid.

Specifically, the loss applies the idea of PINN: encourage the ends of the trajectory to be smooth (f approaches 0 when $\|y\| \to y_5, y_{95}$) and the gradient in the middle to be as large as possible (f larger when $\|y\| \to 0$, resulting in smaller loss).

More specifically, the loss function is defined as:

$$L = \left(L_g + R\right) * \frac{e^{\|f(y_5, s_5)\| + \|f(y_{95}, s_{95})\|}}{\|f(y_0, s_0)\| + 1e - 5} \tag{1}$$

L_g was introduced in the previous document — it is the loss of the entire trajectory for a specific patient. R is the residual loss.

In addition, a **product-form** PINN was attempted: dy/ds = y(1 - y) f(y, s) (I did not directly replace the left polynomial with the actual polynomial model because it would be meaningless.)

Directly using $\frac{\mathrm{d}y}{\mathrm{d}s} = y(1-y)f(y,s)$ produced highly unstable results, prone to explosion.

To address this, a sigmoid constraint was used:

$$\frac{\mathrm{d}y}{\mathrm{d}s} = \sigma(y)(A - \sigma(y))f(y,s) \tag{2}$$

where $A \in R^4$ represents the upper limits of each biomarker in logistic regression.

The results from the above approaches were consistent: the network learned a straight or nearly straight line.

3. Gated-RNN

In the following structures, Equation 1 and Equation 2 are applied

A gated neural network, with the structure as follows:

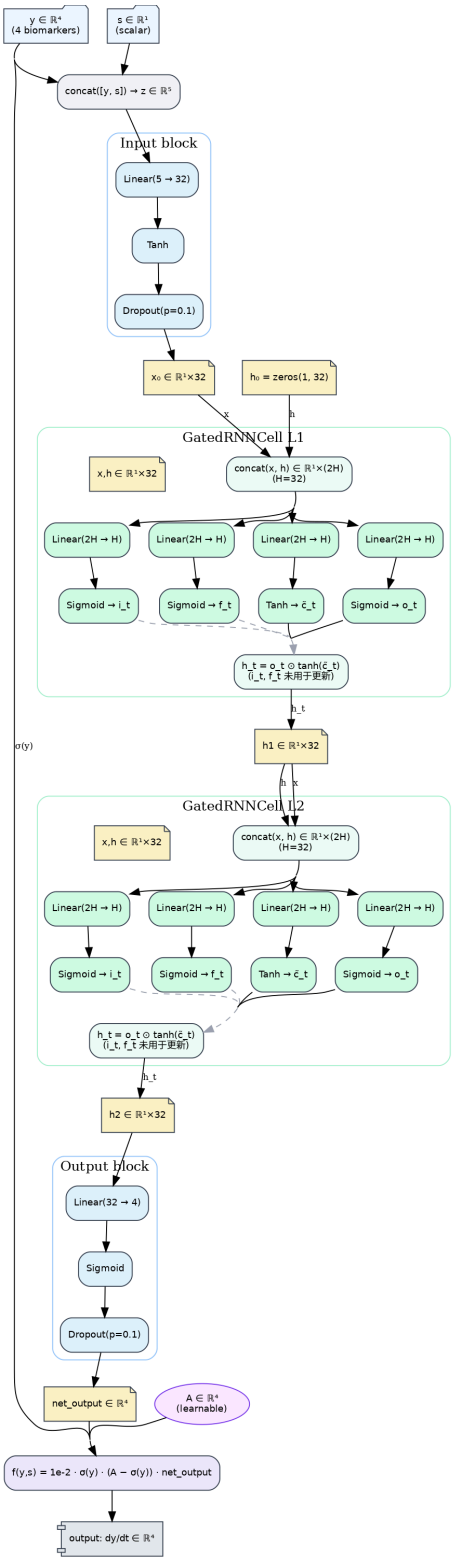

Figure 1: Gated-RNN Structure

This structure can learn a small amount of curvature but not an S-shaped curve.

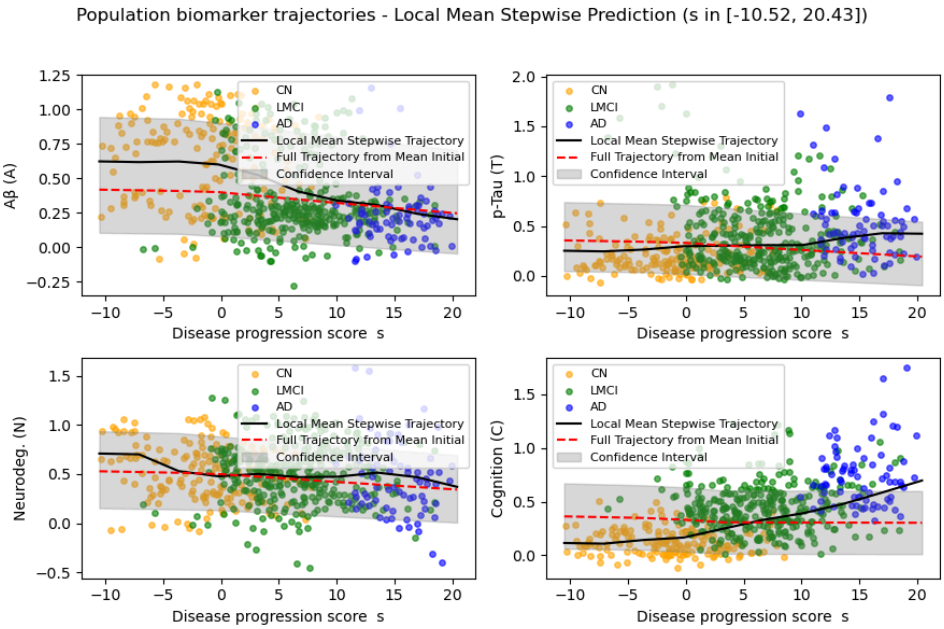

Figure 2: Gated-RNN Result

4. LSTM

An evolved version of GRU-RNN, with similar results.

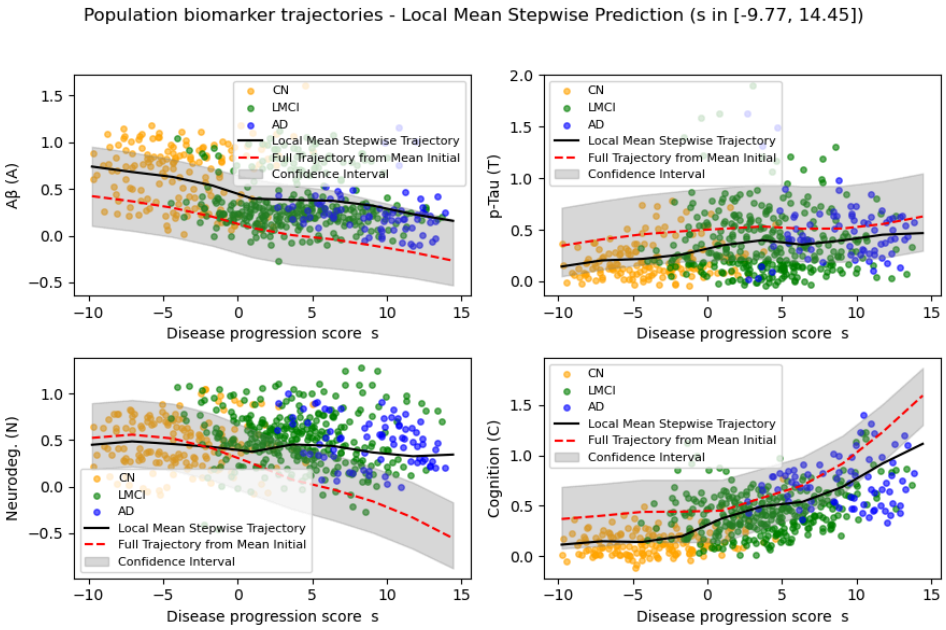

Figure 3: LSTM Result

5. Residual Neural Network + Fine Tuning

Result is as follow:

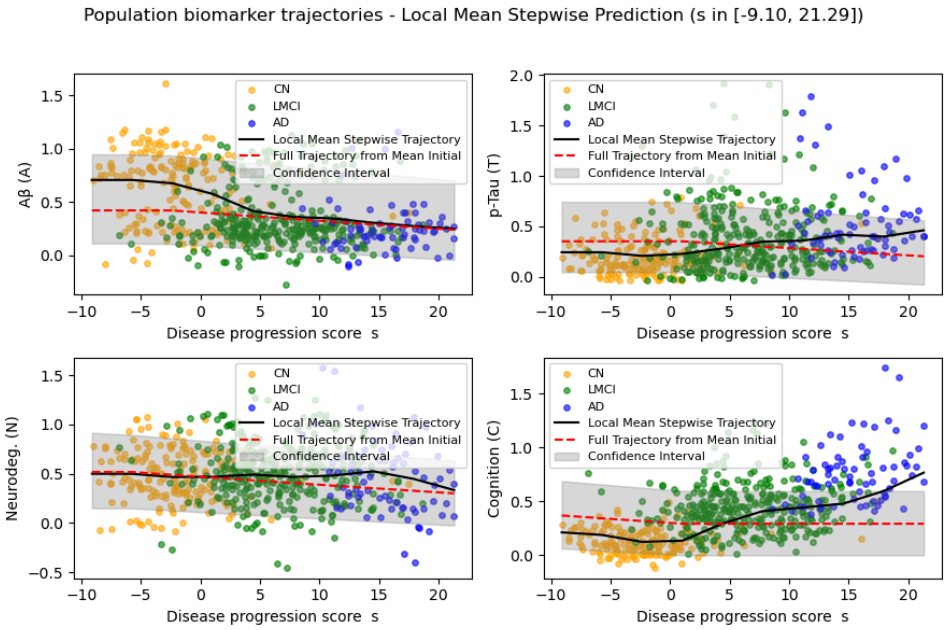

Figure 4: ResNet Result

Fine-tunning: In each round of training, 5% of the parameters are randomly selected and updated with a larger learning rate (1). The new loss is checked to see if it is lower than the original loss. If so, it is accepted. If not, the update is discarded. [1]

Model structure is shown as follow:

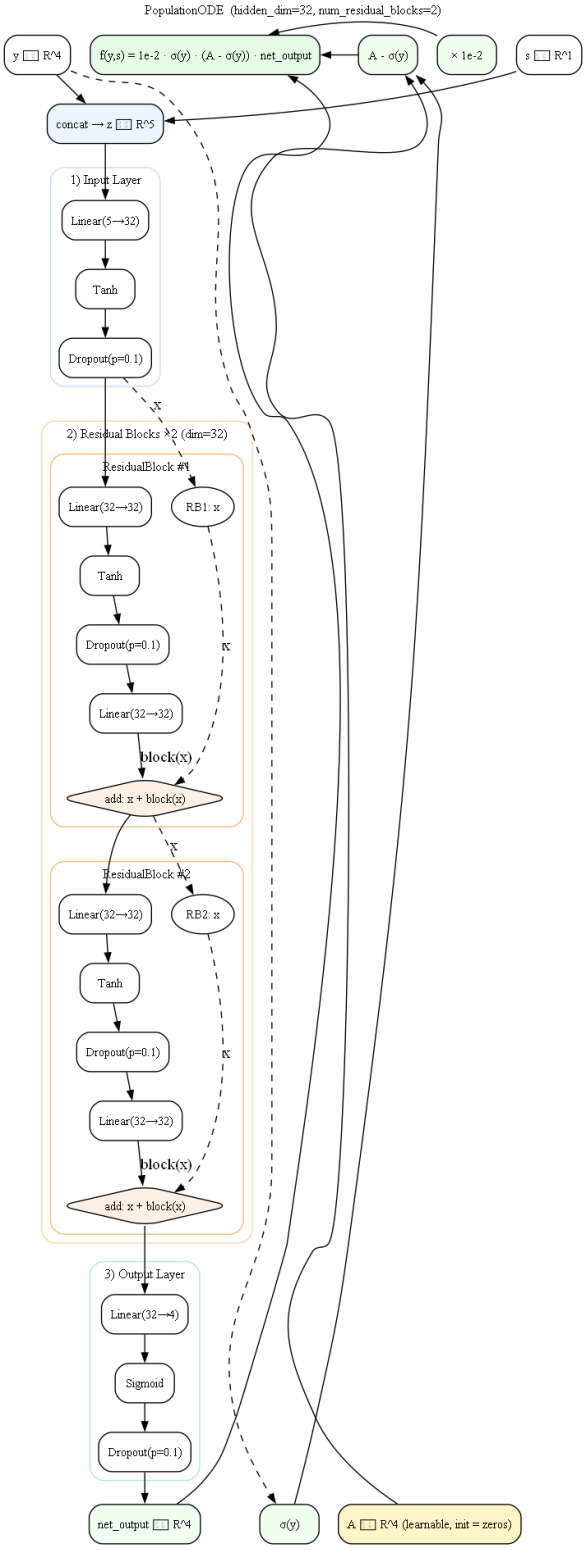

Figure 5: resnet_structure

6. Ideas for Further Improvement

Mainly based on other related work, especially discussions on activation functions.

6.1. 1. Multi-head + Classification Model

Jung et al., 2021, MedIA — **Deep Recurrent Model for Individualized Trajectories**[2]

Goal: Jointly model longitudinal continuous measures and discrete diagnosis for individuals, capable of handling irregular sampling/missing data.

Core algorithm: embed disease progression into latent state z_t and update it with RNN:

$$z_t = f_\theta(z_{t-1}, x_t, \Delta t) \tag{3}$$

Multi-head decoding predicts continuous variables (Gaussian likelihood / MSE) and classification (cross-entropy); missingness is explicitly modeled via masks/time intervals into f_θ. Implemented as RNN/GRU/LSTM backbone + multi-task heads; activations tanh/sigmoid/softmax; authors provide GitHub (model.py etc.).

6.2. 2. BINN

$\frac{\mathrm{d}y}{\mathrm{d}t} = f(x) + g(x)$ Here, $f(x)$ comes from a classical polynomial ODE system in the literature, and $g(x)$ denotes a neural network to learn the residual of polynomial model.

Need to read the paper to see what results generally make sense.

6.3. 3. Abandon layer by layer

The model may be too complex. We can try setting the width of the last few hidden layers to 5, that is, the same dimension as the input and output, and then check the variable values layer by layer, try to discard them layer by layer, and examine the transformation of data as it flows through each hidden layer.

Bibliography

[1] J. Xu and J. Zhang, "Random Masking Finds Winning Tickets for Parameter Efficient Fine-tuning." [Online]. Available: https://arxiv.org/abs/2405.02596°

[2] W. Jung, E. Jun, and H.-I. Suk, "Deep recurrent model for individualized prediction of Alzheimer's disease progression," *NeuroImage*, vol. 237, p. 118143, 2021, doi: https://doi.org/10.1016/j.neuroimage.2021.118143°.

Index of Figures