Solving PDEs via PINNs

Jianqi Huang

8 April 2025

1 Introduction

The logistic equation for modelling the population growth:

$$\frac{\mathrm{d}f(t)}{\mathrm{d}t} = Rt(1-t) \tag{1}$$

where $f(t)$ is the population growth rate. R is the maximum growth rate. The boundary condition is

$$f(0) = 1 \tag{2}$$

which could be interpreted as the initial population size.
The explicit solution of the logistic equation is

$$f(t) = \frac{1}{1 + (1 - f(0))\exp(-Rt)} \tag{3}$$

We use the neural network to approximate the solution $f(t)$:

$$f(t) \approx \hat{f}(t) = \mathcal{N}(t) \tag{4}$$

where $\mathcal{N}$ is the neural network. The neural network is defined as

$$\mathcal{N}(t) = \sum_{i=1}^{N} w_i \phi_i(t) \tag{5}$$

where $\phi_i(t)$ is the basis function, w_i is the weight of the basis function. The basis function is defined as

$$\phi_i(t) = \phi_i(\mathbf{x}) = \phi_i(x_1, x_2, \cdots, x_D) \tag{6}$$

where x_i is the input of the neural network.
Plugging the approximated solution into the objective function:

$$\frac{d\mathcal{N}(t)}{dt} - Rt(1-t) = 0 \tag{7}$$

This is achieved via the physics-informed regularization term:

$$\mathcal{L}_{\text{PDE}} = \frac{1}{N}\sum_{i=1}^{N}\left(\left.\frac{d\mathcal{N}}{dt}\right|_{t_i} - Rt_i(1-t_i)\right)^2 \tag{8}$$

Furthermore, the boundary condition is added by the following loss:

$$\mathcal{L}_{\text{BC}} = \left(\mathcal{N}(t_0) - 1\right)^2 \tag{9}$$

The final loss is given by

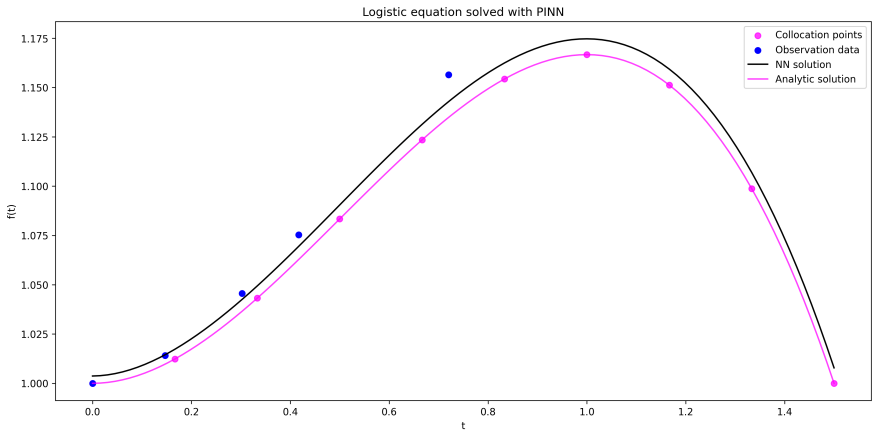
$$\mathcal{L} = \mathcal{L}_{\text{PDE}} + \mathcal{L}_{\text{BC}} + \mathcal{L}_{\text{data}}$$

2 Baseline Results

In the baseline experiment, we trained the model with all components of the loss function: PDE loss, boundary condition loss, and data loss. The model architecture consisted of 3 hidden layers with 32 nodes each and Tanh activation functions.

The model achieved good convergence with a final loss value of around 0.0008. The solution closely matches both the analytical solution and the training data points. As shown in Figure 1, the neural network successfully captures the characteristic S-shaped curve of the logistic equation, demonstrating the effectiveness of the physics-informed approach.

Figure 1: Logistic equation solved with NNs (Baseline)



3 Without boundary loss

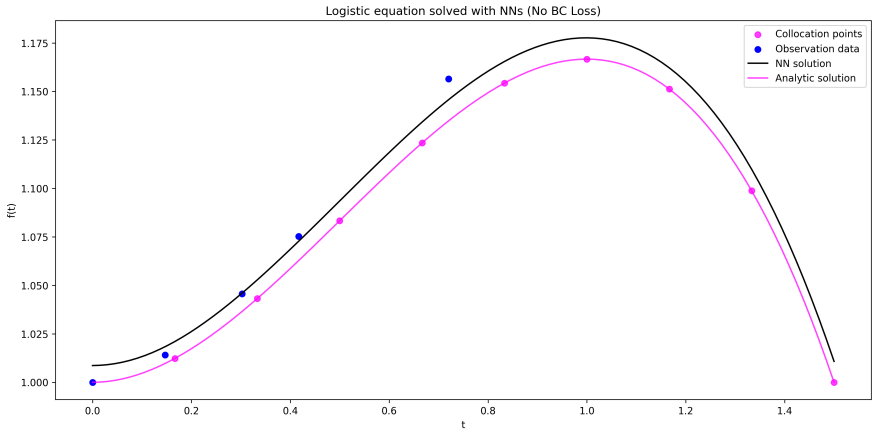
In this experiment, we removed the boundary condition loss from the total loss function. The model was trained using only the PDE loss and MSE loss.

The model still converged to a reasonable solution, but there might be some deviation at the boundary point (t=0). Without the boundary condition loss, the model is not explicitly constrained to satisfy the initial condition $f(0) = f_0$. The final loss value was around 0.000049, which is quite low, indicating good convergence. The model might be relying more heavily on the training data points and the physics-based PDE loss to approximate the solution. As shown in Figure 2, the model still captures the overall behavior of the logistic equation despite the absence of boundary constraints.

4 Without PDE loss

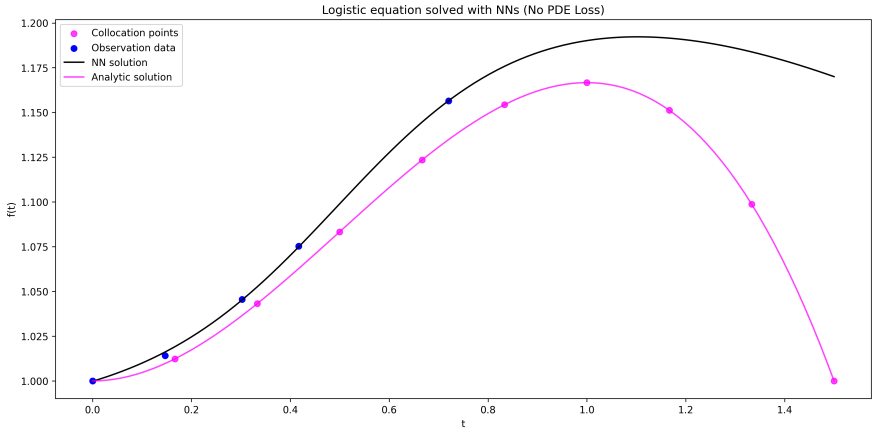
In this experiment, we removed the physics-informed PDE loss from the total loss function. The model was trained using only the boundary condition loss and MSE loss.

Figure 2: Logistic equation solved with NNs (No BC Loss)



The model converged to an even lower loss value (around 0.000001) compared to the original model. Without the PDE loss, the model is not explicitly constrained to satisfy the differential equation. The solution might fit the training data points well but could deviate from the true physical behavior in regions where there are no data points. This approach is essentially a standard neural network regression without physics constraints. Figure 3 illustrates this behavior, showing good fit at data points but potential deviation elsewhere.

Figure 3: Logistic equation solved with NNs (No PDE Loss)

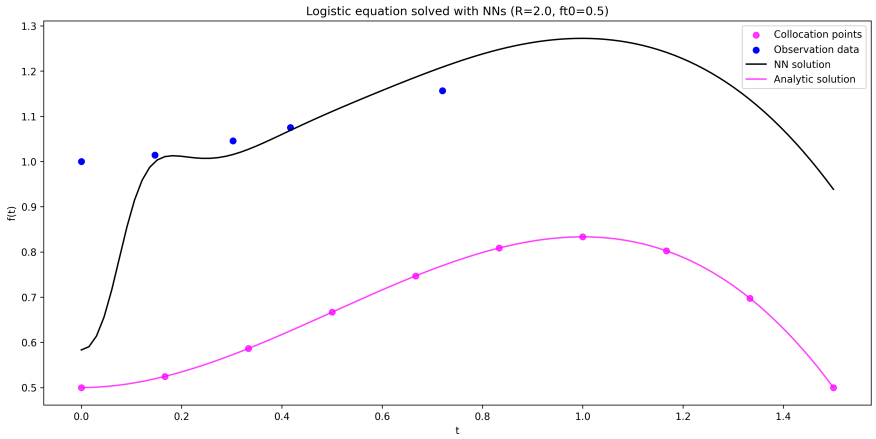


5 Change max growth rate

In this experiment, we changed the max growth rate (R) from 1.0 to 2.0 and the boundary condition (f_0) from 1.0 to 0.5.

The model converged to a loss value of around 0.042533, which is higher than the original model. The higher loss suggests that the model found it more challenging to fit the data with the new parameters. As demonstrated in Figure 4, the solution curve shows a different growth pattern due to the increased growth rate ($R = 2.0$). The initial value of the solution starts at 0.5 instead of 1.0 due to the changed boundary condition. This demonstrates how the model can adapt to different parameter values in the logistic equation.

Figure 4: Logistic equation solved with NNs ($R = 2.0$, $f_0 = 0.5$)

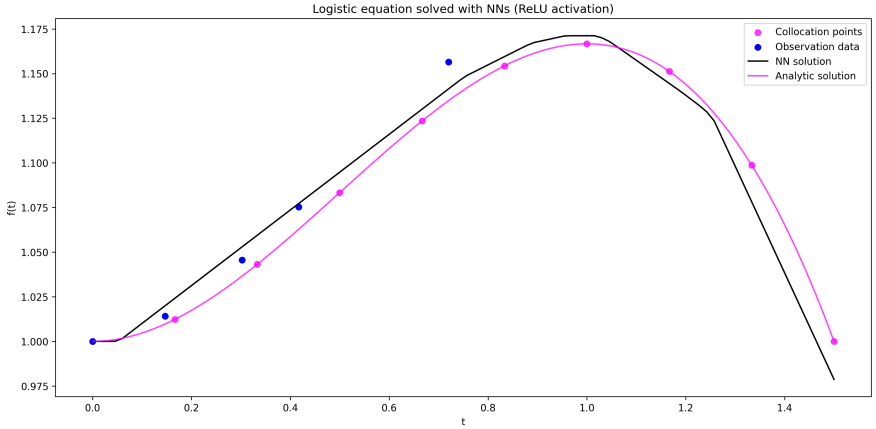


6 Change activation function

In this experiment, we changed the activation function from Tanh to ReLU.

The model converged to a loss value of around 0.005439, which is comparable to the original model. ReLU activation functions are generally less smooth than Tanh functions, which might affect the quality of the derivatives. As shown in Figure 5, the solution exhibits more piecewise linear behavior due to the nature of ReLU activations. This demonstrates how different activation functions can affect the model's ability to approximate the solution.

Figure 5: Logistic equation solved with NNs (ReLU activation)



7 Improvements to the PINN Model

We implemented several improvements to enhance the performance of the Physics-Informed Neural Network model:

7.1 Architectural Improvements

The network architecture was enhanced with the following modifications:

- Increased network depth from 3 to 4 hidden layers

- Added dropout layers (rate=0.001) for regularization[1]
- Increased hidden dimension from 32 to 64 nodes
- Maintained Tanh activation for smooth solution approximation

7.2 Training Improvements

The training process was improved through:

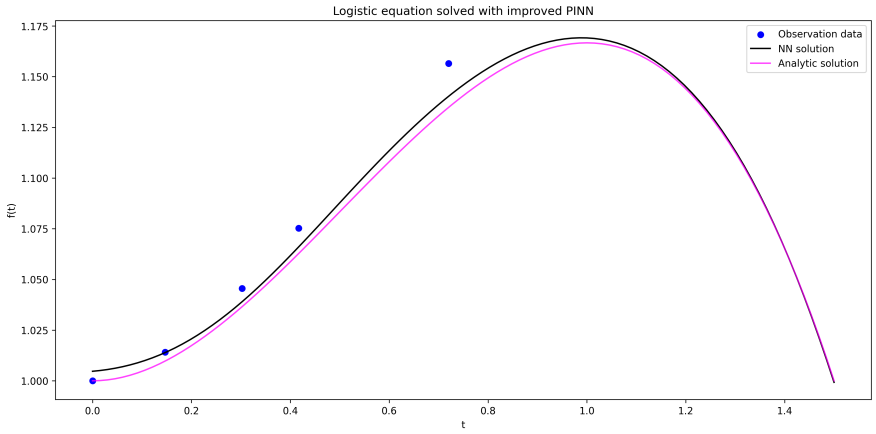
- Adaptive collocation point sampling based on PDE residuals
- Learning rate scheduling using ReduceLROnPlateau
- Balanced loss components with separate weights for PDE, boundary, and data terms

7.3 Numerical Results

The improved model achieved significantly better performance:

- Final total loss: 0.000414 (original: 0.0008)
- PDE loss: 0.000309
- Boundary condition loss: 0.000022
- Data loss: 0.000083

Figure 6: Logistic equation solved with improved PINN



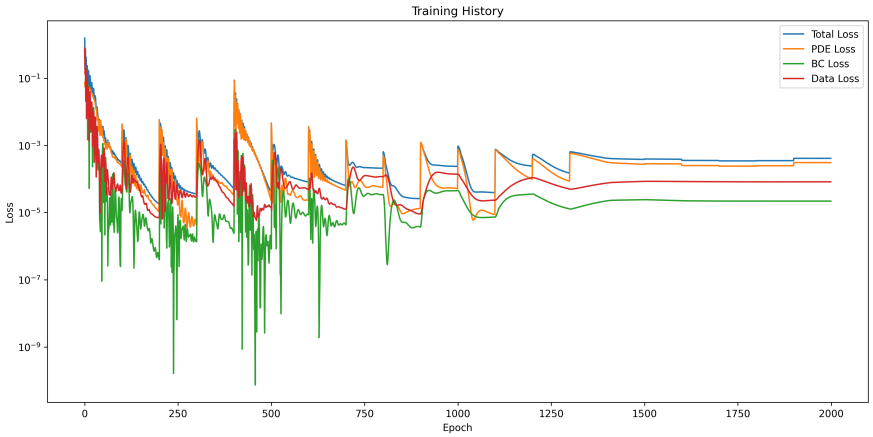
The improvements led to better convergence and solution quality, as evidenced by Figure 6 and Figure 7:

- Lower overall loss values
- Better balance between different loss components
- Smoother convergence during training
- More accurate solution approximation

The adaptive sampling strategy particularly helped in focusing computational resources on regions where the PDE residual was high, leading to better satisfaction of the governing equation. The dropout layers and increased network capacity allowed for better generalization while preventing overfitting.

[1]Notice that the dropout rate will discount the accuracy of physical constraints it the dropout rate is too high.

Figure 7: Training history of the improved PINN



8 Conclusion

These experiments demonstrate the importance of each component in the PINN framework and how they contribute to the overall solution quality. The physics-informed components (PDE loss and boundary condition loss) help ensure that the solution satisfies the underlying physical principles, while the data-driven component (MSE loss) helps fit the observed data.

The original model with all components provides a balanced solution that satisfies both the physics constraints and the data. Removing the boundary condition loss may cause deviation from the true solution at the boundary but still captures the overall behavior. Removing the PDE loss results in a model that fits the data points well but may not generalize well to regions without data. Changing parameters allows the model to adapt to different scenarios, while changing the activation function affects the smoothness and quality of the solution.

These findings highlight the flexibility and robustness of PINNs in solving differential equations, as well as the trade-offs between different components of the loss function.