Diffusion (Heat transfer) equation and machine learning

1 Introduction

Research on machine learning has got a great progess in recent years, and its results have also been applied in many areas, including image recognition, cognitive cognitive, and genomics. Despite all this, the potential for machine learning is still far from being tapped. There is a same situation in physics. Partial differential equation is the research object of physical mathematics method, We study some important physical properties by solving partial differential equations. The traditional method to solve those equations are numerical calculation. Such methods have meet the robustness and computational efficiency standards required in practice. However, Machine learning brings us a new idea for solving partial differential equations. We will use the machine learning method to solve the diffusion (heat transfer) equation, below.

2 Theoretical Section

Let us consider a one-dimension diffusion (hear transfer) equation with periodic boundary:

$$\begin{cases} u_t(x,t) = u_{xx}(x,t) + c\sin(2\pi x)u(x,t), x \in [0,1], t \in [0,1] & (1.1) \\ u(x,0) = 1, & (1.2) \\ u(0,t) = u(1,t), & (1.3) \end{cases} \tag{1}$$

Here, $u_t(x,t)$ is derivative of $u(x,t)$ with respect to t, $u_{xx}(x,t)$ is second derivative of $u(x,t)$ with respect to x. We replace $u_t(x,t)$ in (1.1) by former differential approximation $\frac{u(x,t_{n+1})-u(x,t_n)}{\Delta t}$, and approximating the right part of (1.1) with Crank-Nicolson scheme leaves us with

$$\begin{aligned} \frac{u(x,t_{n+1}) - u(x,t_n)}{\Delta t} &= \frac{1}{2}\big[u_{xx}(x,t_{n+1}) + u_{xx}(x,t_n)\big] \\ &\quad + \frac{1}{2}c\sin(2\pi x)\big[u(x,t_{n+1}) + u(x,t_n)\big] \end{aligned} \tag{2}$$

From this formula, we can design a cost function given by

$$SSE = SSE_u + SSE_b \tag{3}$$

where

$$
\begin{aligned}
SSE_u = \frac{\alpha_1}{N_t(N_x+1)} \sum_{n=0}^{N_t-1} \sum_{i=0}^{N_x} \Big\{ & u(x_i, t_{n+1}) - u(x_i, t_n) \\
& - \frac{\triangle t}{2} \big[u_{xx}(x_i, t_{n+1}) + u_{xx}(x_i, t_n) \big] \\
& - \frac{\Delta t}{2} c \sin(2\pi x_i) \big[u(x_i, t_{n+1}) + u(x_i, t_n) \big] \Big\}^2
\end{aligned} \tag{4}
$$

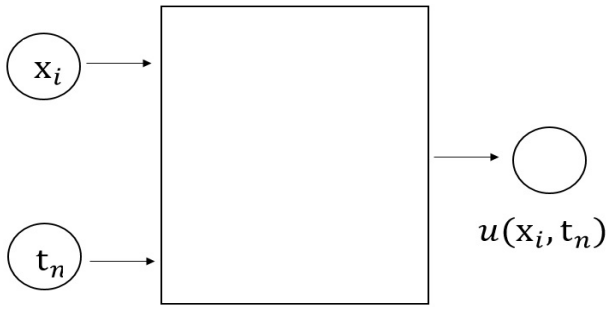
and

$$
\begin{aligned}
SSE_b = & \frac{\beta_1}{N_t+1} \sum_{n=0}^{N_t} \big[u(0, t_n) - u(1, t_n) \big]^2 \\
& + \frac{\beta_2}{N_x+1} \sum_{i=0}^{N_x} \big[(u(x_i, 0) - 1) \big]^2
\end{aligned} \tag{5}
$$

Here, x_i^{n+1} corresponds to the data time t_{n+1}, similar to x_i^n. (4) is obtained through simplification of (3). The first sum of (5) originated in periodic boundary (1.3),and the second come form initial condition (1.2).

3 Implementation

Machine learning works through a neural network, which consists of three layers of neurons: the input layer, the hidden layer and the output layer.Our design for the network is like this:



In the first place, we input t_n and x_i^n corresponding to t_n, and get $u(x_i^n, t_n)$ from output layer. The network can calculate the derivative of $u_{xx}(x_i^n, t_n)$. Next, we can get $u(x_i^{n+1}, t_{n+1})$ and $u_{xx}(x_i^{n+1}, t_{n+1})$ in the same way through the same network.

So far, we have already got all the unknowns in the cost function (3). Next, we will train this network by a training set collected in the domain of equation (1). The network will continuously adjust the weights and deviations through the cost function(3), and get the ability to solve this equation, finally.