# MASC 520 Project 2 Report
# Solve Differential Equation using Neutral Network

**Ruoxi Li**
5817808450

## Abstract

Nowadays, Automatic differentiation makes it possible to approach calculus problems from a machine learning perspective. In this project, a simple differential equation is solved by a neural network. I used a neural nework to find the solution of $dy/dx = -2xy$ with initial conditions, y(0)=1. To keep it simple, I solved the problem with a neural network having a single hidden layer with 10 nodes. After training, the neural network can output almost same result compared to the analytical solution.

## 1 Method

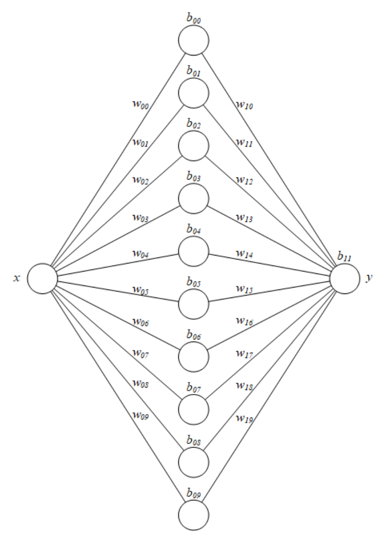

Figure 1: Neural Network Structure

The architecture of the neural network is shown in Figure 1. It has one input layer, one hidden layer with 10 nodes and one output layer. In total, the network has 31 trainable parameters: the weights and biases for the hidden layer, plus the weights and bias for the output layer. The sigmoid function is used as the activation function. At the beginning of the training, the bias and weights can be initialized randomly, using a normal distribution.

We can rewrite the differential equation and initial condition as $dy/dx + 2xy = 0$ and $y(0) - 1 = 0$. In each iteration, we can compute the mean squere of the residuals $dy/dx + 2xy$ and $y(0) - 1$, and use it as the loss function to do backpropergation. To solve the ODE in a certain domain, for example in $-2 \leq x \leq 2$. We can create an array of input values in that range.

Preprint. Under review.

To accelerate the training, I use gradient descent with Nesterov momentum. The learning rate is set to 0.1 and the momentum is set to 0.99.

## 2 Result

I trained the network 1000 epochs for two case $-2 \leq x \leq 2$ and $-3 \leq x \leq 3$. The training loss is calculate every 100 iterations and is shown in Figure 2. We can see that the loss converge nearly to 0, which means that the output should be very close to the analytical result.

The output of the trained model is shown in Figure 3. We can see that the output of the neural network is very close to the exact result as we expected for both cases.

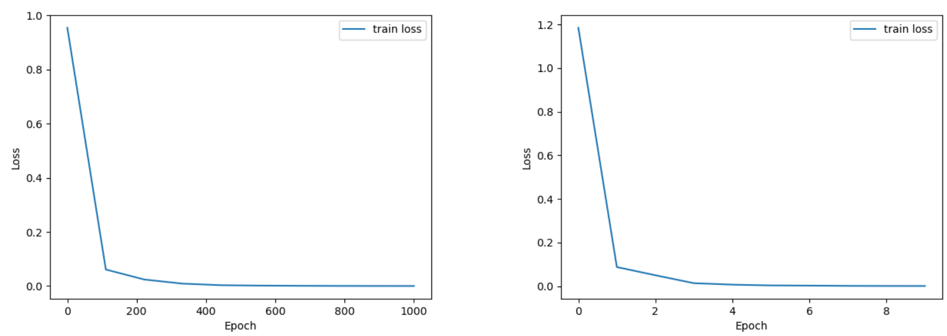


Figure 2: Training loss for $-2 \leq x \leq 2$ (left) and $-3 \leq x \leq 3$ (right)

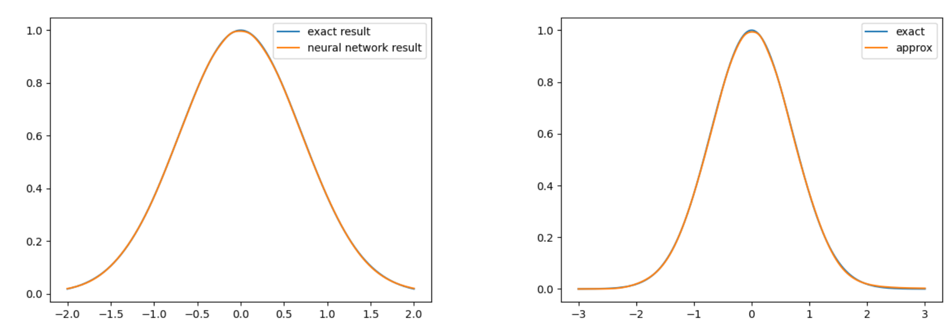


Figure 3: Solution of the differential equation for $-2 \leq x \leq 2$ (left) and $-3 \leq x \leq 3$ (right)

## 3 Conclusion

In this project, I have shown that neural network can be used to solve differential equation. The neural network only has one layer with Sigmoid activation function, but after 1000 training epochs, it can return almost exact result compare to the analytical solution. By changing the range of the input, we can solve the equation in any range. Here, the simple neural network is good enough to solve this easy differential equation. To solve more complicated differential equation, we should add more hidden layers to improve the power of the neural network.