A Machine learning approach to partial differential equations

Hans Mathias Mamen Vege

December 12, 2018

Abstract

nan

1 Introduction

There exists several different approaches that can be used for investigating one of the fundamental building blocks of natural sciences, partial differential equations. The PDEs can appear in almost any field of science, and can give us information about a wide range of topics, such as heat dispersion, wave dynamics, quantum mechanical systems[see 2, ch. 10].

We will be investigating two different approaches to solving partial differential equations(PDEs). The first method being Deep Neural Networks, and for the second approach will be looking at finite differences. For the latter, we will be using Forward Euler. We will first and foremost be investigating solutions with Deep Neural Networks, and use Forward Euler as a method for comparison.

We will begin in section 2 to look at the PDE we are investigating and quickly derive its analytical solution. Then we will quickly go through the finite differences method of Forward Euler to solve the same PDE, and finally go through the basics of a Deep Neural Network(DNN). In section 3 we will go through the results for different combinations of hyper parameters used in the DNN, and compare it with both the analytical results and the Forward Euler results. Then, we will proceed with discussing these results in section 4, and finally in section 5 we will summarize our findings and make appropriate conclusions.

2 Theory

2.1 A solution to the heat equation

The partial differential equation we will be investigating is commonly dubbed the heat equation, and has the following shape

$$\frac{\partial^2 u(x,y)}{\partial x^2} = \frac{\partial u(x,t)}{\partial t} \tag{1}$$

given that $t > 0$, $x \in [0, L]$. As a shorthand notation, we can write this system as $u_{xx} = u_t$. The initial conditions are given as,

$$u(x, 0) = \sin(\pi x) \tag{2}$$

for $0 < x < L$ and $L = 1$. The boundary conditions are given as,

$$\begin{aligned} u(0,t) &= 0 \\ u(0,L) &= 0 \end{aligned} \tag{3}$$

To derive an analytical solution for our PDE, we begin by assuming that the variables are separable,

$$u(x,t) = X(x)T(t).$$

Inserting this into the PDE (1), we get

$$\frac{\partial^2 \left(X(x)T(t)\right)}{\partial x^2} = \frac{\partial \left(X(x)T(t)\right)}{\partial t}.$$

This becomes,

$$\frac{1}{X(x)}\frac{\partial^2 X(x)}{\partial x^2} = \frac{1}{T(t)}\frac{\partial T(t)}{\partial t},$$

in which we see that each side does not depend on the other, such that they are equal to a constant. Calling this constant $-k^2$, we can solve each of them separately. We begin with the temporal part,

$$\frac{1}{T(t)}\frac{\partial T(t)}{\partial t} = -k^2,$$

which has the solution

$$V(t) = a\mathrm{e}^{-k^2 t}.$$

For the spatial part, we get

$$\frac{1}{X(x)}\frac{\partial^2 X(x)}{\partial x^2} = -k^2 \tag{4}$$

which we can see has a solution given as[1],

$$X(x) = b\sin(kx). \tag{5}$$

We now need to find a, b and k. From the initial conditions requirement(2), we get

$$u(x, t=0) = b\sin(kx)\cdot 1$$

Requiring $b = 1$ fulfills the initial condition requirement(2). If we then look at the boundary conditions(3),

$$\begin{aligned} u(0,t) &= \sin(k\cdot 0)\mathrm{e}^{-k^2 t} = 0 \\ u(L,t) &= \sin(k\cdot L)\mathrm{e}^{-k^2 t} = 0, \end{aligned}$$

with the last line giving us that,

$$\begin{aligned} kL &= \sin(0) \\ kL &= n\pi \\ k &= \frac{n\pi}{L}. \end{aligned}$$

Summing up, we then have our analytical solution

$$u(x,t) = \sin\left(\frac{n\pi x}{L}\right)\mathrm{e}^{-\left(\frac{n\pi}{L}\right)t} \tag{6}$$

We can further set $n = 1$ and $L = 1$, as that will match the initial conditions in equation (2).

2.2 Forward Euler and finite differences

Using an explicit scheme, namely forward Euler[2, ch. 10.2.2], the basis for this scheme is in how to approximate the derivative. Using the familiar forward formula for the derivative, we can write the right hand side of (1) as,

$$\frac{\partial u(x,t)}{\partial t} \approx \frac{u(x_i, t_j + \Delta t) - u(x_i, t_j)}{\Delta t}$$
$$u_t = \frac{u_{i,j+1} - u_{ij}}{\Delta t}.$$

Rewriting the left hand side of (1) using a second derivative approximation, we get

$$\frac{\partial^2 u(x,t)}{\partial x^2} \approx \frac{u(x_i + \Delta x, t_j) - 2u(x_i, t_j) + u(x_i - \Delta x, t_j)}{\Delta x^2}$$
$$u_{xx} = \frac{u_{i+1,j} - 2u_{ij} + u_{i-1,j}}{\Delta x^2}$$

2.3 Deep Neural Networks and PDEs

A full derivation of DNNs will not be given here[1], as we will mainly focus on how to apply DNNs to solving PDEs.

3 Implementation

4 Results

5 Discussion

6 Conclusion

Appendices

pass

References

[1] Mary L. Boas. *Mathematical methods in the physical sciences*. Wiley, Hoboken, NJ, 3rd ed edition, 2006. ISBN 978-0-471-19826-0 978-0-471-36580-8.

[2] Morten Hjorth-Jensen. *Computational Physics, Lecture Notes Fall 2015*. 2015. URL `https://github.com/CompPhysics/ComputationalPhysics/blob/master/doc/Lectures/lectures2015.pdf`.

[1] A *more* complete derivation can be found here.