Table of Contents

Neural network learning differential equation

Differential Equation

Let's say we want to our neural network to learn the solution of a differential equation.

$$u' = x^2 + cos(x)$$

with $u(0) = 1$.

```
f (generic function with 1 method)
```

```
f(x) = cos(2π*x) + x^2
```

We can approximate the solution of the equation $u(x)$ with a neural network

Neural Network

```
Chain(#1, Dense(1, 32, tanh), Dense(32, 1), first)
```

```
begin
    using Flux
    NN = Chain(x -> [x],
               Dense(1,32,tanh),
               Dense(32,1),
               first)
end
```

We can encode the initial vaue in by considering $u(x)$ as following.

u (generic function with 1 method)

```
u(x) = x*NN(x) + 1
```

ϵ = 0.00034526698f0

```
ϵ = sqrt(eps(Float32))
```

Let's calculate the derivative. Note that we can also use gradient function using Automatic Differentiation to do this step.

du (generic function with 1 method)

```
du(x) = (u(x+ϵ)-u(x))/ϵ
```

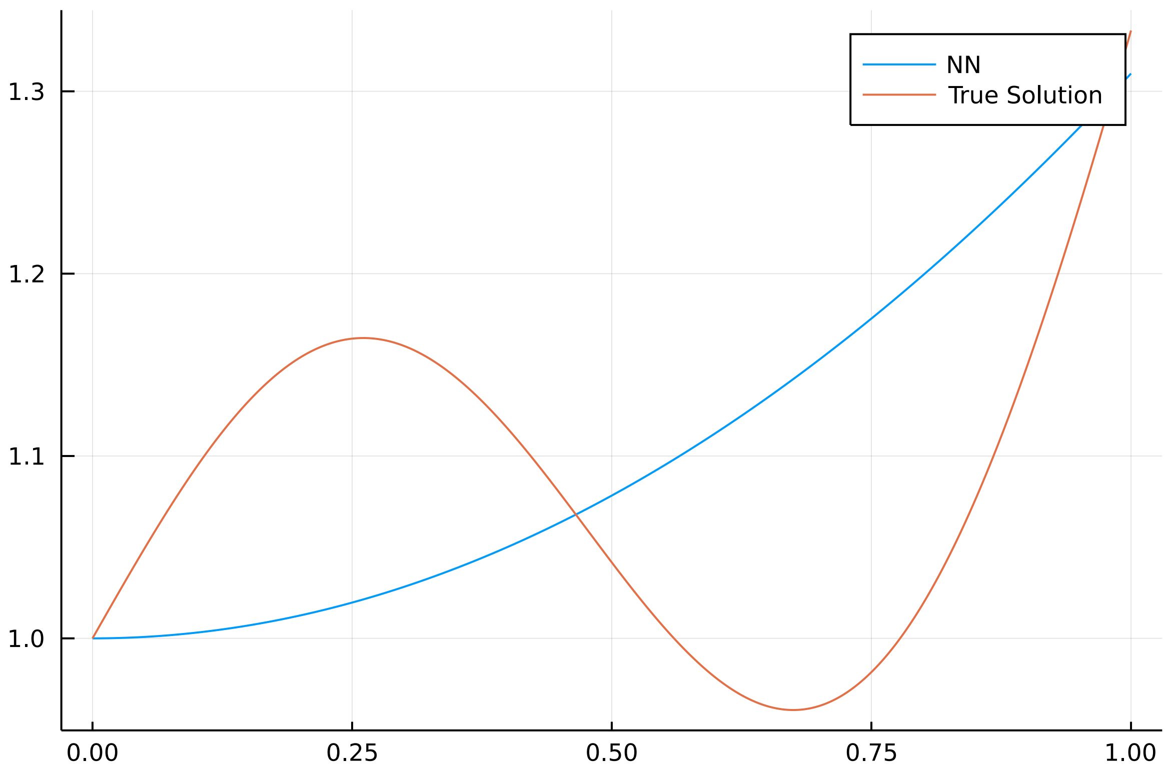
```
using Statistics
```

loss (generic function with 1 method)

```
loss() = mean(abs2(du(x) - f(x)) for x in 0:1f-2:1f0)
```

```
Flux.train!(loss, Flux.params(NN), Iterators.repeated((), 5000), Flux.Descent(0.01))
```

This equation can be solved analytically. Let's compare how is approximation doing compared to the exact solution.



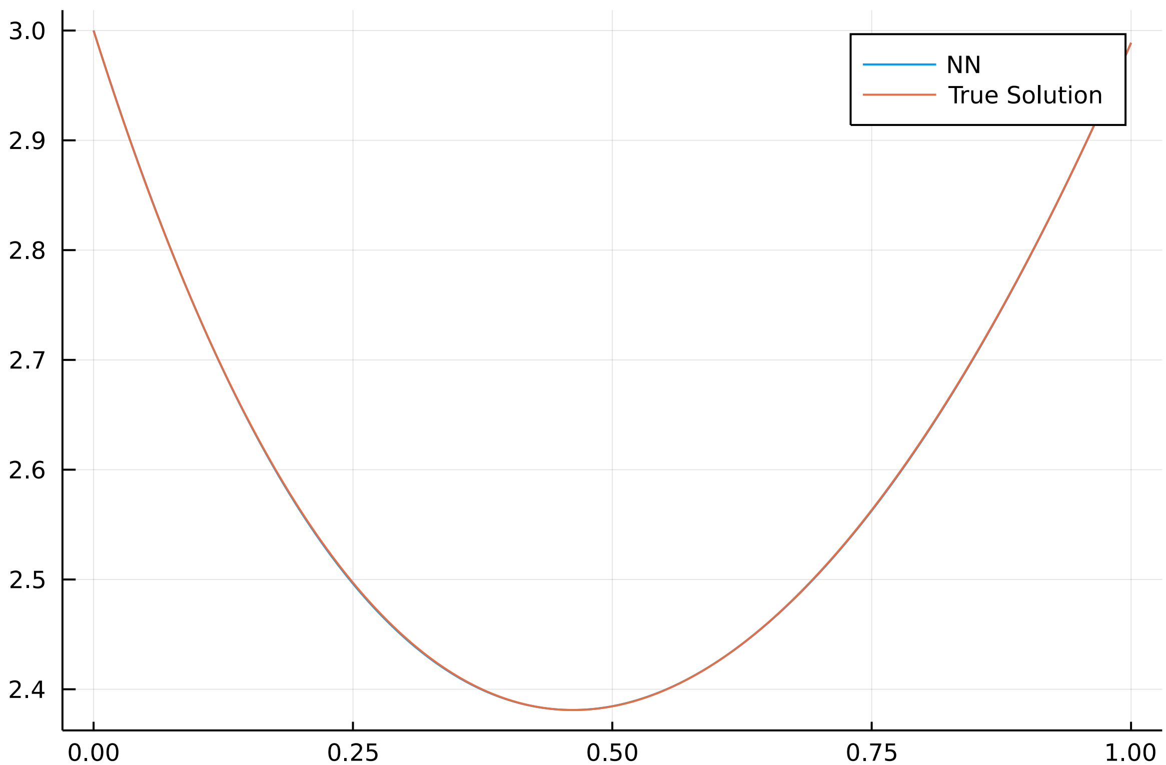

```
begin
    using Plots
    t = 0:0.001:1.0
    plot(t,u.(t),label="NN")
    plot!(t,1.0 .+ sin.(2π.*t)/2π + (t.^3)/3 , label = "True Solution")
end
```

Another Equation

Let us consider another equation

$$y' + 2y = 3e^t$$

with initial condition $y(0) = 3$, and solution $y = 2e^{-2t} + e^t$




```
begin
    NN2 = Chain(t -> [t],
                Dense(1,32,tanh),
                Dense(32,1),
                first)


    g(t) = t*NN2(t) + 3f0
    #using Statistics
    #ϵ = sqrt(eps(Float32))
    loss2() = mean(abs2(((g(t+ϵ)-g(t))/ϵ) - (3ℯ^t - 2g(t))) for t in 0:1f-2:1f0)
    loss2()
    data = Iterators.repeated((), 5000)
    Flux.train!(loss2, Flux.params(NN2), data,ADAM(0.1))

    #using Plots
    #t = 0:0.001:1.0
    plot(t,g.(t),label="NN")
    plot!(t,2ℯ.^(-2t) + ℯ.^t, label = "True Solution")
end
```