

Easy Torch in R

Akanksha Koshti

2025-03-06

```
# Load required libraries
library(torch)
library(plotly)

# Generate a 3D grid using torch tensors
steps = 50
x = torch_linspace(-3, 3, steps = steps)
y = torch_linspace(-3, 3, steps = steps)

# Create a meshgrid using torch
meshgrid = torch_meshgrid(list(x, y), indexing = "ij")
X = meshgrid[[1]]
Y = meshgrid[[2]]

# 3D Gaussian function:
#  $Z = \exp(-((X-\mu)^2 + (Y-\mu)^2) / (2\sigma^2)) / (2\pi\sigma^2)$ 
sigma = torch_tensor(1.0)
mu = torch_tensor(0.0)

Z = torch_exp(-((X - mu)^2 + (Y - mu)^2) / (2 * sigma^2)) / (2 * pi * sigma^2)

# Convert the tensors to matrices using torch operations
X_mat = as.matrix(as_array(X))
Y_mat = as.matrix(as_array(Y))
Z_mat = as.matrix(as_array(Z))

# Convert matrices into a structured data frame
surface_data = expand.grid(x = as.numeric(X_mat[,1]),
                           y = as.numeric(Y_mat[1,])),
  surface_data$z = as.numeric(Z_mat)

# Print a sample of the structured data
print(head(surface_data))
```

```
##           x  y           z
## 1 -3.000000 -3 1.964128e-05
## 2 -2.877551 -3 2.814820e-05
## 3 -2.755102 -3 3.973926e-05
## 4 -2.632653 -3 5.526845e-05
## 5 -2.510204 -3 7.572219e-05
## 6 -2.387755 -3 1.022015e-04
```

```

# Create a 3D surface plot using plotly
fig = plot_ly(
  x = X_mat[,1],
  y = Y_mat[,1],
  z = Z_mat,
  type = "surface"
) %>%
  layout(
    title = "3D Gaussian Distribution using Torch",
    scene = list(
      xaxis = list(title = "X-Axis"),
      yaxis = list(title = "Y-Axis"),
      zaxis = list(title = "Probability Density")
    )
  )

# Show the interactive 3D plot
fig

```

Explanation: 3D Gaussian Distribution using Torch in R

This code creates a 3D normal distribution (bell curve) using `torch` and visualizes it with `plotly`.

1. Generate a 3D Grid (X, Y)

- We create X and Y coordinates using `torch_linspace()` (from -3 to 3).
- `torch_meshgrid()` arranges them into a grid for 3D plotting.

2. Compute Z Values (Height of the Surface)

- We calculate Z using the Gaussian function:

$$Z = \frac{e^{-((X-\mu)^2 + (Y-\mu)^2)/2\sigma^2}}{2\pi\sigma^2}$$

- This creates a smooth 3D surface with a peak at the center.

3. Convert Data for Plotly

- We convert `torch` tensors to matrices using `as_array()`.
- The data is structured correctly for `plot_ly()` to generate a 3D interactive plot.

4. Display the 3D Surface

- We use `plot_ly(type = "surface")` to create the final 3D visualization.

- The result is a smooth, interactive 3D plot of the normal distribution.

