BIOS694 Final Project - Softmax Plot

2025-04-17

```
library(torch)
library(torchvision)
library(luz)
library(reshape2)
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tibble)
library(caret)
```

```
## Loading required package: lattice
```

```
library(here)
```

```
## here() starts at C:/Users/mince/Desktop/McGill/Courses/BIOS694/Project
```

```
library(tidyr)
```

```
##
## Attaching package: 'tidyr'

## The following object is masked from 'package:reshape2':
##
##     smiths
```

MNIST dataset

```r
torch_manual_seed(42)
dir <- "./dataset/mnist"

train_ds <- mnist_dataset(
  dir,
  download = TRUE,
  transform = transform_to_tensor
)

test_ds <- mnist_dataset(
  dir,
  train = FALSE,
  transform = transform_to_tensor
)

# Select one test image
test_sample <- test_ds[77]
image <- test_sample$x$unsqueeze(1)
label <- test_sample$y - 1
```

Load the Student models

```r
# The students trained with T = 1, 3, 10
fitted_student1 <- luz_load(here("model/mnist-cnn-student1.pt"))
fitted_student3 <- luz_load(here("model/mnist-cnn-student3.pt"))
fitted_student10 <- luz_load(here("model/mnist-cnn-student10.pt"))
```

```r
# Set models to eval mode
fitted_student1$model$eval()
fitted_student3$model$eval()
fitted_student10$model$eval()
```

```r
logits1 <- predict(fitted_student1, image)
probs1 <- nnf_softmax(logits1, dim = 2)

logits3 <- predict(fitted_student3, image)
probs3 <- nnf_softmax(logits3, dim = 2)

logits10 <- predict(fitted_student10, image)
probs10 <- nnf_softmax(logits10, dim = 2)
```

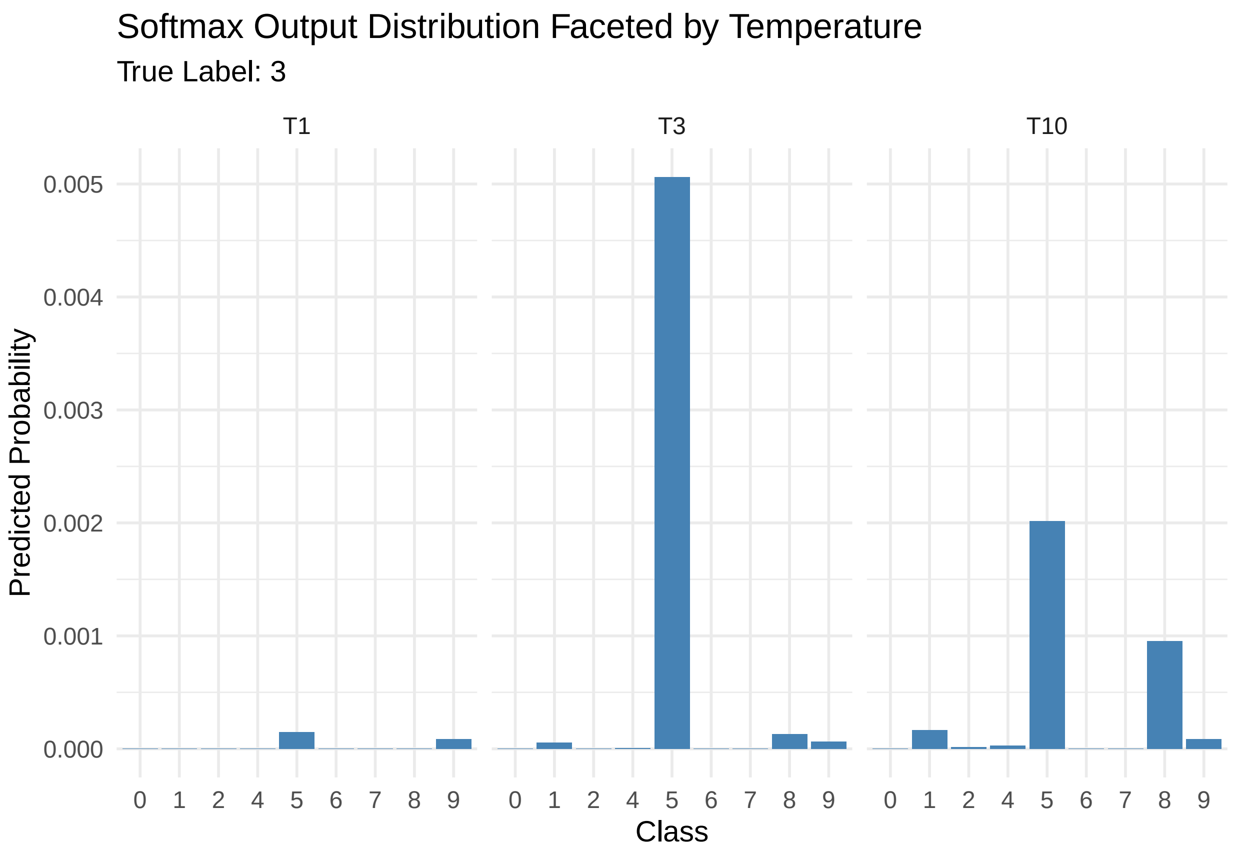
```r
# Convert tensors to arrays
probs1_array <- as_array(probs1)[1, ]
probs3_array <- as_array(probs3)[1, ]
probs10_array <- as_array(probs10)[1, ]
```

Plot the soft targets distribution

```r
# Create tidy data frame
df <- data.frame(
  class = factor(0:9),
  `T1` = probs1_array,
  `T3` = probs3_array,
  `T10` = probs10_array
) %>%
  pivot_longer(
    cols = starts_with("T"),
    names_to = "Temperature",
    values_to = "Probability"
  ) %>%
  group_by(Temperature) %>%
  filter(Probability < max(Probability)) %>%
  ungroup()

# Set the facet order explicitly
df$Temperature <- factor(df$Temperature, levels = c("T1", "T3", "T10"))

# Plot
ggplot(df, aes(x = class, y = Probability)) +
  geom_col(fill = "steelblue") +
  facet_wrap(~Temperature) +
  labs(
    title = "Softmax Output Distribution Faceted by Temperature",
    subtitle = paste("True Label:", as.numeric(label)),
    x = "Class",
    y = "Predicted Probability"
  ) +
  theme_minimal()
```


Softmax Output Distribution Faceted by Temperature
True Label: 3
T1
T3
T10
Predicted Probability
0.005
0.004
0.003
0.002
0.001
0.000
0
1
2
4
5
6
7
8
9
Class