Today, I started learning deep learning in earnest using Pytorch. First, I recapitalized what the neural network is, what the batch size is, and the reason for flattening. In the code to make the neural network model, it was amazing because I felt that the layers deepened as the code increased. The training, validation, and training_for_epochs functions were long, so it did not reach me at once, but I understood from the big frame that there was a process for setting which loss function was used and which optimizer was used, and that there was a process necessary for learning. Through the given code, training acuity and validation acuity could be checked, and execise proceeded in the same process. In execise, fashion MNIST data was used instead of handwriting data. I thought that this series of processes could be used in applications such as optical character recognition.

After that, I learned CNN. It was impressive to share the same parameters through the calculation process of creating a new matrix using kernel. I also learned about dropout, and found that there is a generation effect such as sensorble learning. I also learned the concept of data augmentation and its necessity.

Before starting CNN execise in earnest, I experienced the process of allocating and using gpu. Originally, I launched and used anaconda in VSCode, but I had a hard time setting up a new environment by changing it to a jumper notebook. Still, it was nice to finally complete the configuration. It was not easy to match the existing code to the new dataset in execise, but I was able to understand how to configure the CNN code through the professor's explanation.