### Question 2 (2 points)

**- 2a. Take a look at the [Pytorch documentation for DataLoaders](
https://pytorch.org/tutorials/beginner/basics/data_tutorial.html). In your
own words, why is it important that we use a `DataLoader` object to
container our data, rather than a different data structure?**

DataLoader simplifies the process of iterating through the dataset in training loops and integrates seamlessly with PyTorch's training routines. It allows us to divide the dataset into batches. This is important because processing the entire dataset at once may not fit in memory, and batching enables efficient processing by loading only a portion of the data into memory at a time.

**- 2b. Try running `python -m free_response.batch_sizes <num_examples>
<batch_size>` with different values of `num_examples` and `batch_size`. In
your own words, what trends do you notice?**

Increasing the num_examples while keeping batch_size constant leads to more epochs required to process the entire dataset, resulting in longer training times.

**- 2c. Using `free_response/batch_sizes.py` as above, find a *single value* of
`num_examples` and *two values* of `batch_size` such that the **LARGER**
value of `batch_size` allows the model to train *faster*. Say which values
you used for this to be true, and explain why this happens.**

num_examples = 1000, batch_size = 64
large batch sizes often result in more stable gradients and less frequent weight updates, which can lead to faster training convergence.

**- 2d. Using `free_response/batch_sizes.py` as above, find a *single value* of
`num_examples` and *two values* of `batch_size` such that the **SMALLER**
value of `batch_size` allows the model to train *faster*. Say which values
you used for this to be true, and explain why this happens.**

num_examples = 1000 , batch_size = 16
Smaller batch sizes result in more frequent weight updates, which can lead to faster convergence in some cases.