Saaif Ahmed

saaifza2@illinois.edu

Chapter 2 Reflection

1. Please give a brief summary of the chapter?

The chapter begins with an introduction to neural networks, and their history and variations. The main variations discussed are general Deep Neural Networks, Convolutional Neural Networks, and Recurrent Neural Networks. With the advent of the GPU it became feasible to run the massive parallel operations necessary to facilitate running and training a neural network model. A neural network is made of many neurons; thus, we need to figure out how to train a single neuron. Each neuron takes in an input parameter, multiplies them by a weight value, and then places them into an activation function. There are many activations functions, but the chapter covers 5 functions: Sigmoid, ReLuU, Tanh, Leaky ReLU, and ELU. Next is an example and walkthrough of applying gradient descent on a single neuron using the sigmoid activation function.

Moving on, the chapter then discusses how to expand this to a DNN or deep neural network. This time, we will have many layers in between input and output and each layer will contain many neurons each with their own weight. Each layer will have a weight matrix, and a vector containing the values of the constants associated with the bias terms. We can use back propagation starting from the output moving backwards to the input, to calculate all the gradient of weight matrices for each layer and the same for bias vectors. By then applying gradient descent we can optimize parameters. The chapter then goes onto to discuss the main example of the chapter which is using a PyHealth database to construct a model of whether a patient will revisit. It is best reviewed on its own to describe the process because it is specific to this dataset and machine learning library.

2. What improvements do you want to see in this chapter? Please elaborate on them

There are two main improvements I would like to see in this chapter. I think at the beginning of the book it would be helpful to include the hard calculations for the gradients. I understand that this book is meant to be for those who are extremely familiar with the linear algebra needed to do machine learning work, however in the context of students, especially graduate students it has been a while since we've had to take multiple chain rule derivatives. If we want to get a true appreciation for the mathematics in back propagation, it would be helpful to at least show the process once such that people can reference back at it.

The other topic that I would like to see improved is the way in which the code is presented. I think it is very difficult to follow the chapter when the code blocks are split by pages. I think it is much more worthwhile for the code to be referenced in figures and left in their whole image rather than them being split apart across pages. Especially for a language like python where indexes and spacing matters, it can become difficult to read. If this is meant to be printed in a

real textbook, using both pages at once is very helpful, and makes for a better learning experience.

3. What are the typos in this chapter?

- Page 19 of the shared pdf: “Then we call $get_dataloader$” is a typo.

Otherwise no other typos were noticed.

4. Which part of the chapter do you like most?

The part of the chapter I liked the most was the mathematical breakdown of how back propagation worked in one neuron up to the whole DNN. I enjoy understanding mathematics, as it becomes much easier to implement it in code. It also allows the developers and engineers to analyze the performance of the model. Knowing what each step of the model training is responsible for creates the opportunity to catch problems before they arise. Furthermore, foundational math is quite interesting as it uses linear algebra in ways unusual by general computer science standards. In a typical course of study, a CS student will probably not use calculus again after they have taken their early year courses. Machine learning reintroduces these rules and techniques into the lives of these students.

5. What are the most useful things you learned from this chapter?

The most useful thing I learned in this chapter is the example of building a deep neural network from scratch. PyTorch makes it relatively simple to build a decent model and run the parameter optimization over many epochs. However, the breakdown of the data is usually the hardest part. The written-out and step by step instructions portion of the example were very helpful. Having the proper code as well makes testing it ourselves quite easy. Structured data can get very complicated, therefore these types of examples teach us how to analyze these data sets so we can make relevant AI models for them.

References

Xiao, C., Sun, J. (2021). Introduction. In: Introduction to Deep Learning for Healthcare. Springer, Cham. https://doi-org.proxy2.library.illinois.edu/10.1007/978-3-030-82184-5_1