

I have been going through the book ***"Deep Learning for Coders with fastai &PyTorch" by Jeremy Howard and Sylvian Gugger.*** I found this book fascinating, and easy to comprehend. I also found that the code snippets provided in the book, are easy to understand and try out.

Deep Learning is full of jargons and complex mathematical formulae. I have tried several top Neural Network books, but halfway through, I used to feel lost. I am practitioner and not an academician who is constantly in touch with calculus and matrix algebra. I am more interested in finding solutions to business problems. I would like to have **code/SDK** that can be applied to the problems I am trying to solve. It is more of a combination of AI concepts and Applications that I am interested in, not just theory.

This book is what I was just looking for! It starts with a note that Deep Learning is indeed not as complicated as it is made to look. And that one does not require a PhD in Data Science or a Degree in Math to build state-of-the-art, novel Deep Learning models. It then goes on to introduce Neural net, and its architecture. It briefly introduces Machine Learning and emphasises on the need to have a holistic view of Deep Learning (including solving for ***"How will the end-user consume the output?"****).*

The Book argues that Jupyter is good enough to build and deploy production grade model, and can be used to build basic Applications with User interface. Most of the Neural Network terms are explained in simple English. For example: Parameters and Activations are just numbers. Activations are the output of each layer of the neural net, and parameters are just coefficients of weights of each independent variable. One concept that caught my mind is ***universal approximation theorem*** which states that any function can be approximated to any degree of accuracy by using a combination of 1 Linear and 1 Non-Linear layers of Neural network. This is the basis for Neural Network

The book contains examples and sample code for a variety of problems in Cognitive learning space involving text, images and video. At the same time, there are several examples that shows how to use Deep Learning on tabular data to solve problems such ass forecasting, regression and classification. Concepts are explained in simple language, and code snippets are provided at the same time. I think it is very important to be able read and comprehend code written in Python/Pytorch and fastai. It reinforces theoretical concepts that we just learnt.

The authors of the book are creators of fastai. Here what the authors have to say about fastai (source: docs.fast.ai)

*fastai is a deep learning library which provides practitioners with high-level components that can quickly and easily provide state-of-the-art results in standard deep learning domains, and provides researchers with low-level components that can be mixed and matched to build new approaches. It aims to do both things without substantial compromises in ease of use, flexibility, or performance. This is possible thanks to a carefully layered architecture, which expresses common underlying patterns of many deep learning and data processing techniques in terms of decoupled abstractions. These abstractions can be expressed concisely and clearly by leveraging the dynamism of the underlying Python language and the flexibility of the PyTorch library. fastai includes:*

* *A new type dispatch system for Python along with a semantic type hierarchy for tensors*
* *A GPU-optimized computer vision library which can be extended in pure Python*
* *An optimizer which refactors out the common functionality of modern optimizers into two basic pieces, allowing optimization algorithms to be implemented in 4–5 lines of code*
* *A novel 2-way callback system that can access any part of the data, model, or optimizer and change it at any point during training*
* *A new data block API*
* *And much more...*

*fastai is organized around two main design goals: to be approachable and rapidly productive, while also being deeply hackable and configurable. It is built on top of a hierarchy of lower-level APIs which provide composable building blocks. This way, a user wanting to rewrite part of the high-level API or add particular behavior to suit their needs does not have to learn how to use the lowest level.*

The examples in the book show how easy it is to build a neural net and deploy it in production - just by writing few lines of codes using fastai on pytorch.

There are several documents, and videos that help people at all levels to get onboard with fastai and PyTorch. Here are some useful links

* <https://www.fast.ai/>
* <https://course.fast.ai/>
* <https://course19.fast.ai/part2>

One of my favourites (and probably yours too, if you want to get deep into the algebra behind neural nets)

* <https://github.com/fastai/numerical-linear-algebra/blob/master/README.md>

I urge you to give it a try, and see if it helps in expediting some of your Deep learning projects. One note: PyTorch will require GPU machines. Since fastai works on PyTorch, the same applies for fastai too. So make sure you have access to GPU machines. One way is to use Paperspace.com which provides servers with GPU for free. If you have Cloud Credits such as GCP, Azure or AWS, you can spin up machines there and install fastai libraries there. I think AWS and GCP provide fastai images too - which can be deployed on a GPU machine.

Another interesting fact is that the book was written on Jupyter notebook. Therefore, free version (jupyter notebook version) is available on fast.ai website. The hardcopy is available on Amazon.