A framework to design your own learning pathway in Data Science

In this article I will be talking about perhaps the most important aspect of getting into AI or Data Science – a framework for deciding how, what and from where to learn your data science concepts.
To be more specific, I will try and answer these questions from my own experience:

- Identifying what you need to learn.
- What you do not need to learn and some prerequisites.
- Choosing the right online courses and identifying the bad ones.
- Links to popular courses for various pathways
- Where do books fit in all of this?

> My focus will be on online courses. Since offline courses differ wildly in quality between colleges, and a reader still in school may not even have access to those, it doesn't make sense to talk about them in a generalized way.

Since there's a lot of ground to cover in this article, I'll take help of bullet points wherever possible to save everyone's time.

Identifying what you need to learn

First of all, go read the section "But then, how do I choose what to do ?" of the linked article if you haven't already. We need to have a good starting point before we start this discussion. I'll be assuming you have done so going forward in this article.
Typically, most people's interests or end goals would fall into these categories:

1. Computer Vision: generating art, face detection, identifying diseases from medical imagery, measuring crop yields and air pollution from satellite imagery, VR/AR applications, robot vision, etc.
2. NLP/NLU: Voice assistants (Siri, Alexa, etc.), chatbots, AI models that can write code, etc.
3. Traditional Machine Learning: estimating stock prices, customer inflow, life expentancy, loan eligibility, etc. from thousands or millions of rows of structured data
4. Data Analytics: using tools like SQL, PowerBI, Tableau, MS Excel, etc. to analyse huge amounts of data and gather useful insights for an organization
5. Reinforcement Learning: robots that can (might) walk naturally, intelligent agents that beat humans at computer games, autopilot military aircraft that defeats real pilots, etc.

Try and think of which of these application areas you might want to end up in and read the rest of the article in that context. Don't worry if you don't know what you want, you're in good company (that's most people) and it's completely okay!
I won't talk a lot about data analytics a lot because I do not have much experience doing that. Now, for other 4 categories, typically, you will start with learning the same basic concepts of traditional machine learning that will be a common theme across these domains. That is to say, you will pretty much need to start with a course that focuses on ML and not DL regardless of what you want to do and diverge from it later down the line based on your specific interests.

- If you're aiming to get into CV, NLP or RL, then the best approach is to choose an introductory ML course and complete it quickly before moving into domain specific courses.
- If you aim to get into ML only, then your first ML course should have enough rigor (like CS229 from Stanford University)

- Whatever is your first course, you should do it thoroughly and at a comfortable pace. Make sure you understand every little detail that you can. The reasoning is that everything going forward will ***heavily*** draw from these foundational ideas ***again and again and again*** and you will slow yourself down later if you rushed through these concepts early on.
- PLEASE DO YOUR ASSIGNMENTS (no matter how easy). I cannot stress this enough. I frequently come across people who claim to have completed CS231n in a weekend and can't even write a binary classifier or wade through an elementary Kaggle dataset because they didn't do any assignments.

Once you're beyond your first hurdle of learning some basic ML (for those wanting to get into CV, NLP or RL), that's when you try and dig into specific areas of your interest. By this time, you will hopefully have sufficient background to make an educated choice for yourself.

What you do not need to learn, and some prerequisites

Quite often, people ask me about a host of other topics related to computer science which they feel they might need to learn before getting into AI. I'll try and answer some of the most common doubts people have here:

1. No, you do not need to learn data structures and algorithms or have any competitive coding experience.
2. No, you do not need to learn web development or have any other software development experience.
3. No, you do not need to learn C, C++, Fortran or JAVA.
4. No, you especially do not need to learn CUDA, MPI or OpenMP.
5. No, you do not need to be an expert coder like that wizard friend of yours. As you'll see when you start learning, you just need some basic math and a more than average attention span.
6. No, you do not need a few courses in statistics or advanced linear algebra up your sleeve.

You can read a very good blog post on fast.ai that talks about some of these things here

Typically, most courses you'll do will generally cover these prerequisites. But in case they don't, here's what you need to learn, in order:

1. Python. 99% of your time will be spent with this language so you should spend some time learning it. There's a very good course on **Kaggle Learn** that gets you started with just enough python skills to get you started in AI. It won't take a lot of your time and is to the point : https://www.kaggle.com/learn/python
2. Numpy
   a. Introductory: CS231n Numpy Tutorial
   b. Comprehensive (recommended): Link
3. Pandas, Matplotlib, Seaborn:
   This is a great series of notebooks to get an ML-focused practical introduction to Pandas and data vizualization:
   a. Exploratory data analysis with Pandas
   b. Visual data analysis in Python
   c. Seaborn & Plotly (can skip this)
4. Matrix Calculus Refresher (by FastAI founder – Jeremy Howard): https://explained.ai/matrix-calculus/index.html

You can also read Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython by Wes McKinney, the creator of Pandas himself. This book gives an in-depth introduction to all the libraries that are generally involved in a data science workflow with an assumption that you have basic Python skills. The book is designed in such a way that you can read individual parts of the book without necessarily reading the previous chapters to learn only what you need. Also, chapters for most libraries are split into two – introduction and an in-depth dive. For now, you just need the introduction parts to start doing some ML.

That's about it! Note that the links I've provided are by no means entirely comprehensive for the respective topics (except the book). They are designed to teach you *just enough* to get started with data science as quickly as possible without any friction. At the same time they aren't too introductory to leave you filling gaps on the internet. Most people get lost in trying to cover these topics in detail and never end up starting an ML course. Please try and avoid that and learn what you need only when you need it.
If you spend a few hours every day, you'll be done with all of these in a comfortable 2 weeks (as opposed to doing that random 10 week python course on coursera followed by everything else that you also need to learn)

Choosing the right online courses and identifying the bad ones.

Before I start recommending online courses for all your specific needs, there are some hard learned observations or truths that I think you should know about before choosing courses. Online courses can roughly be divided into these categories:

1. Courses that were originally taken in a university or classroom setting and later uploaded online as is (eg: CS229 form Stanford, Princeton's Algorithm's course). These courses are great, but sometimes a little inconvenient to follow along without proper material.
2. Courses that were originally taken in a university or classroom setting and later ***adapted*** for online audiences without losing any of the content (popular example: CS50 by Harvard University, Fast.ai). These are by far the best kind of courses, generally.
3. Courses that were made for an online audience from the start:
    a. Made for specific online audiences with a certain background or motivation: these vary in quality but are generally good. Some good examples are: Algorithmic Toolbox, Machine Learning, courses on Udacity, etc.
    b. Made to be accessible for everyone on the internet: these are the most basic, introductory courses that give you a false sense of satisfaction by reiterating over and over again that you're doing great and say that you now know everything at the end of the course. These courses form the major chunk of online education. Generally try and avoid these if you can if you're looking to build deep expertise in a subject. Examples of these are those "Introduction to XYZ" courses that you find all over Coursera and Udemy, and sometimes on edX.

The kind of courses you should generally prefer, in order, are: 2 → 1 → 3.a → 3.b
Explanation: In my experience, as far as your ability permits, you should always take more rigorous courses as opposed to the ones you find in category *3.b*. Very frequently, you'll find that a lot of the simplistic online courses leave you with huge gaps in your knowledge (both breadth and depth) and you're always left wanting a bit more. You eventually find yourself spend *a lot more time* wading through the internet (blogs, other more rigorous courses, etc.) to fill in those gaps. This realization generally sets in when you start working on some project and realize that you don't have any practical skills yet.

So rather, it's best to simply take rigorous, and possibly longer courses in the first place. You'll need to (and should) spend significant time with the first 2-3 of these courses to get your foundations right after which you'll never have to take an online course again. *You will simply be able to read research papers and documentation to quickly gather new skills exactly when you need them without needing to spend a few weeks on another course****.*** **This last bit is what you should aim for**. This is way more important than what you'll realize when your'e starting out. Things move fast in the data world, books get outdated in a few months, and online courses are simply not built to keep up with that. So you'll need to be able to read papers and documentation (both of which are to the point, and the fastest, most accurate way to learn new things) to gather new skills.

But what do I mean by rigour? A rigorous course will have these characteristics:

1. A good breadth of content (number of topics) with a good focus on latest developments.
2. For important parts of the course (foundations, established methods, etc.), there will be no compromise on depth. For relatively niche, unestablished or new topics, they might skimp on the depth, which is fine.
3. The assignments and tests will be made to make sure you understand everything. Less work will be already done for you and you will be responsible for most of the code. This is probably the most important part of a good course.

   a. In the best courses, the evaluation will not only test your theoretical understanding, but also the implementation aspects.
   b. In some courses, there will be a heavy focus on implementation (code) and less on the theory. Such courses will generally give you a very good intuition of the theory that might be better than the theory focused courses (ex: FastAI). Depending on your goals, you might actually prefer this.
   c. Some courses won't have formal assignments. Instead, the instructor will recommend things you should try out in your free time (like in FastAI). For some people, this can be more rewarding than the alternative. As long as you have the motivation to follow through, this is still fine and probably better than the above points.
4. They will spend time on foundational topics without rushing to the more advanced stuff.
5. They will have you do very impressive projects that you could even proudly put on your resume.
6. Almost always, it will be a course that was originally taken offline in a university setting. If not that, it will have a very accomplished instructor.

A lot of courses on the internet ***seem*** to satisfy these requirements. But very often, in the name of breadth, they will teach you just one aspect of each topic and quickly move on. They will then claim that they covered dozens of topics when all they did was barely touch the surface. In their seemingly comprehensive assignments, most of the times they will have a major chunk of the work done for you already. Some friends of mine who've done such courses later reported that they're unable to do any projects entirely on their own. You don't want to end up like that.

With all of this out of the way, I believe you now have a good understanding of what to look for in any learning resource you follow on the internet.

Popular courses for different learning pathways

There are a lot of online courses on the internet these days, each of which are great in their own way. I'll try and stick to the popular courses because they generally have a community built around them and it's easier to get support when you're stuck. I will also try and not recommend more than one or two items for each category since that tends to confuse beginners.
As mentioned before, in almost all cases, you should do an ML (or related) course first before dabbling into anything else. Also, as we talked about in the previous section, depending on your exact aims, you may need to do an entry level course or a more in-depth course. Here are my recommendations:
Introductory courses (best for people wanting to jump into deep learning quickly):

1. Andrew Ng's classic ML course on Coursera: Although the course is taught in Octave (~ MATLAB), it doesn't matter as much because as you'll see, all the ML libraries in Python carry their syntax from MATLAB itself.
2. First 2 courses of the deeplearning.ai specialization on coursera.

> Note that I **do not** recommend doing the deeplearning.ai specialization in it's entirety. It's very introductory in a lot of places, and almost everyone I know in college (BITS) who has done the specialization had to do more rigorous courses to actually be able to apply their skills. The specialization is spread out over a few months with not a lot of content or depth to show for it. Please try and avoid it.

In depth ML courses:

1. CS229 from Stanford University: lot of focus on the theory and really understanding everything
2. https://mlcourse.ai: heavy focus on implementation without dumbing down on the theory. This will help you actually win those competitions and build great projects

If you do not want to go into deep learning, then you can stop here after having done one of the previous 2 courses (or both) and start making great projects, participating in competitions, etc.

After this, for people wanting to enter deep learning but aren't sure exactly what subdomain they want to focus in:

1. MIT's Intro to deep learning: this course moves very fast, has a great breadth of content, amazing projects, and has good focus on the theory. If you want a quick (and yet comprehensive) introduction to deep learning to figure out whether it's the right thing for you, then go for it. You do not specifically need to do ML basics courses before this course as they try and cover those parts where needed.
2. fast.ai: The latest version of this course now covers theory in depth, has you make the best projects and take them to an actual, minimal web app, and has by far the best community forums on the internet. FastAI alumni regularly win Kaggle competitions, publish papers, get hired at top tech companies and research labs, and break state of the art records. The course also has the best instructions for all kinds of setting up you will need to get into deep learning, and the forums are filled with answers to every doubt you'll ever have. Again, they also cover the ML in sufficient depth so you do not need to do and previous courses before starting this one (except learning Python). The only caveat is that this course needs more motivation from your side. The amount of content covered is a lot and there's a lot of code to go through. There's also a lot of abstraction in the fastai library (which some people find uncomfortable), but it doesn't matter as long as you know what's going on without having written the code for it. If you prefer books, you can read the recently released Fastbook (freely available on GitHub) which has the same content as the video course. If you were to take just one advice from this whole article, I'd say you should do this course. Things covered in this course: Neural networks, computer vision, NLP, ML, tricks to get the best results like data augmentation, learning rate scheduling, progressive resizing, presizing, writing your own optimizers, looking inside deep models, and so much more.

For people wanting to enter Computer Vision:

1. The Ancient Secrets of Computer Vision: This is the most well-rounded computer vision course I know of as it not only teaches you the deep learning side of CV but "older" methods like SIFT and optical flow as well. The instructor is the creator of the popular object detection framework called YOLO. He recently left computer vision research after seeing his research being used for military applications.
2. Deep Learning for Computer Vision: this course is pretty much an updated replica of the popular CS231n course (by the same instructor). Unlike the course above, this one is focused only on the deep learning side of computer vision. Also, the assignments are challenging for most people and you will learn a lot in the process.
3. ADL4CV - Advanced Deep Learning for Computer Vision - Technical University Munich: This course is a phenomenal next step for anyone who has already taken an introductory CV or DL course and wants to explore ideas like neural rendering, interpretability and GANs further.
4. Notable mentions: The MIT and especially the FastAI course mentioned above also have a significant focus on computer vision along with other things. Also, they have much better projects for you to showcase.

For people wanting to enter NLP/NLU:

1. CS224n (latest lectures): this course from Stanford university has been a popular choice for people wanting to enter NLP. The currently online available online version of the course was last updated in Winter 2019. There is a very good focus on the theory and all the assignments are challenging enough just like in CS231n.
2. Code-First Introduction to Natural Language Processing by Rachel Thomas from FastAI: This course carries the same philosophy as that of the previous FastAI course — code first, implementation focused, huge amount of content, great projects from the first lecture, and a focus on getting the state of the art results with the latest techniques. Unlike other fast.ai courses though, this course covers a lot more theory.

For people wanting to enter data analytics:
This is where you take huge troves of data and use statistical techniques to derive useful insights from the data. For instance, you can look at thousands of waist measurements of people in a region to decide what a clothing company should standardize the S, M, L and XL sizes of shirts to for that region.
As I previously said, I do not personally have a lot of experience in this area and will refrain from giving any advice here. In my knowledge, there are no established courses as of now for this domain. Although I would say that the mlcourse.ai

website has it's ML course structured in a way that will end up teaching you a lot of what you're looking for (since data exploration is a huge part of ML projects anyway).

For people wanting to enter Reinforcement Learning:
Again, I personally do not have much experience in RL, but some of my friends from the Meta Daisy club @ BITS Pilani (only club in BITS that's focused purely on RL, yet) were happy to share their insights on the topic. Reinforcement learning, due to its highly mathematical nature, still doesn't have as many good resources on the internet as other established areas like ML, CV and NLP. Also, the subject area has yet to prove itself significantly useful outside of carefully simulated environments. That is to say, RL hasn't entered our daily lives yet. The recommended pathway to enter RL includes a lot of focus and hard work, along with these courses:
Getting the basics right (do one of these courses):

1. Spinning Up in Deep RL (by OpenAI): This isn't exactly a "course". This is a carefully curated and organized group of resources, implementations, exercises, algorithms and papers for anyone to get started with RL. For people that prefer text based resources, this is a good course to start with.
2. David Silver's course on RL: David Silver is a leading authority in the RL space and his course is currently considered a staple in RL. If you're not comfortable with openai's offering and prefer video lectures, this course if for you.

After you've got your basics covered, my friends at Meta Daisy suggest just one course that you should be going through:

- CS285 Deep Reinforcement Learning (UC Berkeley): This course also starts from the basics, but I've been told that it might be too much for a beginner and so you should complete one of the above courses before jumping into this one.

What about books?

There are a lot of great books on the subject for those who prefer text based resources. There are some caveats though. For example, books that promise to teach you machine learning using a certain library or framework (scikit-learn, PyTorch, TensorFlow, Keras, etc.) get outdated pretty quickly within a year or so. The reason being the fact that these libraries are not mature yet and their APIs change very frequently. They're starting to become stable now, but there's still a long way to go. For scikit-learn, the API has remained pretty much constant over the years because of it being a very old project (2007). Make sure that the books you're getting are as current as possible.
On the other hand, there are books that are not based on any particular ML/DL framework, and focus more on the theory. If you choose to read such a book, you can always go ahead to any ML/DL library's documentation and learn the implementation from the examples there. I'll try and recommend a very small selection of books so as to not confuse beginners with a myriad of choices. I will also try and only recommend books that I'm fairly certain will still be relevant after 2-3 years from now.
Here are some of the book which I've heard a lot of good things about:

1. Deep Learning by Ian Goodfellow: Although this book is from 2015, it's sort of timeless (as of now) as it focuses significantly on the foundations and the theory behind the subject. The book is quite mathematical and is a slow read if you want to extract the maximum value out of it. But to be honest I have never read anything quite like it. Ian Goodfellow doesn't dumb down anything and I recommend this book only for people who want to go into research (a must read).
2. Deep Learning with PyTorch: This book comes right from the creators of PyTorch and is freely available for a limited time. Since it's coming from the source, expect this book to stay updated and relevant for quite some time. The book covers concepts in a simple and very accessible way for readers of all levels and at the same time gets you comfortable with a major deep learning library.
3. Python Machine Learning (3e): The author (Sebastian Raschka) has won several awards for this book including one from ACM. It introduces you to ML using scikit-learn and moves on to using TensorFlow for deep learning. The book has been getting regular updates since it's first release in 2016 and so it should stay relevant for quite a while.

There are several books for CV, NLP and RL. But in my opinion it's best to go through the courses that I've mentioned above for these areas when you're starting out. I'll refrain from recommending anything in this article that I myself wouldn't do.

In the next article, I'll talk a lot about setting up your computer or server for ML or DL. It will mostly be a compilation from different sources on the internet since a lot of people on the internet have already gone through the trouble of writing about it.