Self-Learning Machine Learning

Given the vast amount of resources available online on this topic, the list below is merely a curated collection of thoughts—put together by me and my students—rather than a definitive guide. It should be supplemented with critical thinking, independent research, and continuous feedback on learning. This list reflects our perspective on "How we would approach self-learning ML today," based on our current understanding and its applications in our research area. Please let us know if you think there are better resources, lists, or step-by-step guides to do this.

Probability & Statistics (recommended before linear algebra)

1. **Stat 110 — Probability (Harvard), Prof. Joe Blitzstein**
   The gold standard introductory probability course. Exceptional treatment of Bayesian reasoning, conditional probability, and distributions. Highly recommended as the primary starting point for probability. `[link]`
2. **CS109 — Probability for Computer Scientists (Stanford)** `[link]`
   A highly recommended introductory probability course from a CS perspective. Any equivalent course would also suffice.
3. **CS229 Probability Review Notes** `[link]`
   Concise, self-contained notes for revisiting core probability concepts quickly before or during an ML course.

Linear Algebra

4. **All Things Linear Algebra — Prof. Gilbert Strang (MIT)** `[link]`
   The best person to learn linear algebra from. Work through as much material as you can.
5. **Essence of Linear Algebra — 3Blue1Brown (Grant Sanderson)** `[link]`
   Stunning geometric intuition for every core concept. Watch alongside or right after Strang. Non-negotiable.
6. **Matrix Methods for ML — Prof. Gilbert Strang (MIT 18.065)** `[link]`
   A specialized follow-up covering SVD, PCA, and optimization in the ML context.
7. **Interactive Linear Algebra (UBC)** `[link]`
   Excellent interactive book for geometric revision. Consult as needed after Strang's courses.

Programming

8. **Python and Standard Libraries**
   Learn Python and standard scientific libraries: `NumPy`, `Pandas`, `Matplotlib`, `SciPy`. Fluency here is prerequisite to all practical ML work.

Machine Learning

9. **Machine Learning Specialization — Prof. Andrew Ng (Coursera)** `[link]`
   Ideal first exposure: builds intuition for terms, ideas, and workflow before diving into theory.
10. **CS4780/5780 — Machine Learning (Cornell), Prof. Kilian Weinberger**
    Arguably the best rigorous ML course available freely online. More theoretically deep and engaging than CS229, with excellent lecture notes. `[link]` `[link]`
11. **CS229 — Machine Learning (Stanford), Prof. Andrew Ng** `[link]`
    The classical ML course. COL774 by Prof. Parag Singla (IIT Delhi) is also a strong alternative. `[link]`
    Lecture notes: `github.com/mxc19912008/Andrew-Ng-Machine-Learning-Notes`

Deep Learning (after core ML)

12. **EECS 498 — Deep Learning for Computer Vision (UMich), Prof. Justin Johnson**
    Begin with the *first 10 lectures*: covers neural networks, CNNs, GPU training, Attention, and Transformers at depth. One of the finest DL courses available. `[link]` `[link]`

Practice & Research

13. **Build Projects**
    Work on end-to-end ML projects to solidify understanding. Implement papers, compete on Kaggle, or reproduce key results from scratch.
14. **Approach a Professor for a Research Project**
    Once the foundations are in place, research experience is the most accelerating step. Seek out a lab working on problems that genuinely interest you.

If you know of better resources, step-by-step guides, or alternatives to any of the above, please share them.

Prepared for $\mathcal{P}^2$-Lab, EE, IIT Roorkee `https://psquare-lab.github.io`. Can be shared with Acknowledgment.