You have 10,000 hours during your undergraduate years. (50*52*4)
How will you spend it?

Undergraduates at elite schools are generally good at book learning, but doing well in class is often not enough to secure hypercompetitive positions or develop an extreme talent. Most students burn their undergraduate hours by wandering into areas, but students should instead align with an optimized strategy.

Let us say an undergraduate student is interested in helping with ML Safety.
What should they do? Work through the MIRI reading list? Take advanced mathematics courses, since some people wish they took more maths? To get into an ML PhD graduate school a different strategy is needed.

First, read about how to get into CS graduate school.
https://www.cs.cmu.edu/~harchol/gradschooltalk.pdf
You will need to do research and get 2 strong letters of recommendation (plus another good letter), all while getting a good GPA (hopefully >=3.8). If your GPA is <3.6 you'll have a hard time getting into grad school.
During the admissions process, they will look at your application for a few minutes, as there are so many to sift through. When they submit your letter to different schools, your letter writers will enter a value on a numerical scale (for Berkeley, it's insufficient opportunity to observe, 4, 5, 6, 7—the mean is ~6 for most letter writers). Reviewers will look at your publications too.

Roughly, the objective is to produce or assist with lots of research and get good letters, subject to maintaining good grades.

This roughly means you should try to get familiar with deep learning, start researching, and get your application ready to send off in your fourth year.

Things to do:

Certainly do research during the summer of your second year and third year, possibly with different people to get two strong letters.

Learn your basics in your first year. Learn linear algebra, probability with random variables, multivariable calculus, and programming. While you can understand DL without it, at some point you will need to learn it for job interviews. You will also lack confidence if you have shaky foundations. Learn how to prove things only because your GPA will suffer if you are not good at writing small proofs. Learn deep learning and learn content as needed, such as facets of optimization. Try to come up with many intuitions for what you learn; this will require months of ruminating on content and hearing different people present it. After you learn the necessary background knowledge, your goal must move beyond being smart and must now be to come up with good ideas.
Take a deep learning course, such as https://atcold.github.io/pytorch-Deep-Learning/ or CS231n

If your university has renewable research credits, strategizing is much easier. At my undergraduate university, I could only get research credits for one quarter.

Take project graduate courses. These types of graduate courses are graded very leniently (A is the most common grade, then A-, and then B+s; profs don't want to give grad students bad grades; it's not normally distributed like in undergraduate courses—its histogram is like a wedge with A being the most frequent).
Line up project-based courses in your second and third year, while you should do much book learning in your first year.

If the course is time-consuming and will not help you with research, and if you can put it off until your fourth year, then look to put it off. I put off difficult C programming courses in my fourth year.

It is useful to take a heavy course load in your first year though, because this will force you to become more conscientious. Just choose your first-year courses wisely.

Maximize the amount of time you can research in your third year. Research performed in your fourth year likely won't be very reflected in your application.

The NeurIPS deadline is at the end of the school year, and you hear back whether papers are accepted in fall. The last time you can have a top ML conference paper on your grad school application CV is by submitting to NeurIPS. That means you'll need your paper squared away in your third year. Even if you have a prodigious summer after your third year, it will mostly be reflected through letters.

Carefully choose your advisor. They might be a lazy tenured professor or a weak researcher who got lucky and happens to be a prof (this happens at all places, including the top schools). The best single metric is "Highly Influential Citations" on Semantic Scholar, but this doesn't show how they're currently doing. See if they're writing many big influential papers in the past five years. Do not just weigh the total h-index; that says what they've accumulated all time, not whether they're currently connecting with the research field. If they're not with the field in recent years, then you'll probably wind up on one of their dud papers. It's OK to work with one researcher who's not at the forefront/whose prime has past if they advise well and spend time with you, but in your first you need to work with one highly active researcher who is connecting with the research field. Avoid being paired with a weak graduate student, as most grad students do not know what they are doing and are bad managers and lead you to think you dislike research.

It's useful to have an advisor who will spend time with you, if possible. If they're not available, befriend a competitive PhD student or a postdoc. Nearly all of my advising experiences happened as an undergraduate.

Course selection and course sequence planning is the easiest simple structure to optimize. It will determine how many free hours you have a week and also influence your GPA. In my second year, I planned out some tentative specific schedules for the next three years.

Initially it may make sense to get on a larger paper or try to shoot for a nice workshop contribution. Start small. Only a handful of incoming PhD students have a first author conference paper—most do not.

Follow ML news sources (Andrew Ng's newsletter, Import AI, Paperswithcode newsletter, ML Safety Newsletter, https://twitter.com/popular_ml) and https://paperswithcode.com/ and especially http://www.arxiv-sanity.com/top
It will take over six months to get a sense of what's going on. Even so, following these will help you with self-supervised learning. At some point it'll become easier for you to skim papers.

Figure out if you like research. If you don't, optimize for becoming a research engineer (and take more programming courses and practice leetcode).
Some people do much better with clearer directions and tighter feedback loops, and some have a very hard time with sparse feedback signals on long time horizons.

Take a personality survey. https://www.outofservice.com/bigfive/
If you are very high in neuroticism or agreeableness (85+ percentile), or if you are below average in conscientiousness or openness (<50th percentile), then this is some evidence that you may *not* be temperamentally suited for research. If you're at an elite school you're probably very conscientious and probably like learning and have high openness. You can probably afford to be in the bad direction for one of these traits. For example, a hands-on advisor could mitigate your low conscientiousness, and an extremely structured environment could reduce the impact of neuroticism. Also, fortunately conscientiousness is somewhat manageable with routine.

Allocate as much time as you can to research. Humans are remarkably adaptable, so you can progressively work harder. I used to be afraid of becoming a prof since they seem busy, but then I became progressively stronger. Get in a group where working hard and succeeding is socially rewarded. You will be a combination of the people you hang around: if your friends are goofing around and if you are burning much time on hedonistic activities (wasting weekends, juggling polyamorous relationships, dating app addiction, getting trashed consistently, youtube or netflix or video game addiction, etc.), and if you are not trying to fix it, I would heavily bet that you will fail to have an outsized impact. "Be regular and orderly in your life, so that you may be violent and original in your work."

Common failure modes:

Avoid tough needless courses and take easy or highly relevant courses. Only put substantial time in highly instrumentally useful courses. No "Honors Advanced Algorithms" or "Math 55." This is the easiest way people burn time. Students love comparing themselves to their peers.

This means kids start taking hard undergraduate courses that are a waste of time ("wow, you're in <really hard maths class that you're getting a B+ in and is requiring 25 hours a week>? you must be a big man on campus; you are high status"). I almost took the honors introduction to computer science, which would have made me learn things I did not need to know such as Haskell. I already knew how to program, and this course would have just stolen time from learning relevant content. I would not take "honors" courses when I could instead just work on research. Again, this document is for deep learning graduate school. For maths grad school, you'll need to take all of the hardest courses, but for CS grad school, research is what matters. In my first year I was known to be taking all the hard courses, and then I stopped and went off everyone's radar for the remainder of my undergraduate; I rarely saw other undergraduates and removed myself from their prestige games, as I was hanging around the graduate students.
I can't see much reason for a double or triple major since those will force you to take many more courses that you otherwise would not take.
Again, it may be useful to have a heavy course load in your first year though, as it will force you to become more conscientious.
The application readers will not look at your specific courses that closely. It's too difficult and time-intensive to weigh this across tons of universities. They'll probably just look at your GPA; if they do skim it, they might just look for red flags. If they like you a lot, and if they're an ML theory type and happened to go to your same undergraduate university, they might look closely, but this is unlikely to be decisive.

Generally avoid time-intensive extracurricular activities. I have a good friend who wisely but painfully quit his competitive A Capella group for research (and he got into UW and CMU).

A student comes in with an extreme talent in something that is not very useful for ML research, such as mathematics, so they burn many of their hours pursuing this interest. This only makes sense if you intend to do ML theory, which is not neglected and unusually low in tractability. I initially took many maths courses, but fortunately I was willing to stop being notable for being good at maths. Students wanting to stay king of a molehill will lose in the long run. Again, this is very common among people who have an extensive background in maths. I've seen so much motivated cognition for their taking that graduate functional analysis course. They enjoy maths, and their ego and sense of self is based on their maths ability; doing more useful things might be somewhat less fun and they would need to voluntarily start as a lowly newbie at the bottom of a prestige hierarchy.

You do not need to take an "AI" course unless you intend to research RL (an example AI course: https://stanford-cs221.github.io/spring2021/#schedule). The content is largely outdated. I have never taken an "AI" course, but as an undergraduate I took six ML courses. Of course, more knowledge is better, but this isn't as useful as a deep learning course that covers a specific area such as NLP.

As of 2021, do not learn a lot about RL unless you will have a very strong advisor in the area (Finn, Abbeel, etc.) or are in a very strong group. It is too hard to get oriented in RL without

being in a competent research group. This may change when sequential decision making methods become less fragile.

Be somewhat leery of unusual strategic advice from very old profs. They were operating in an environment where a majority of their friends became professors and when PhDs were much shorter. The days of writing one paper during your PhD and getting a top professorship are well over. ML is substantially more competitive than it was 10 years ago, back when there were 15 people or fewer in each ML class.

My trajectory:
In the first year, I tried gaining background to become a competent researcher. This meant taking preliminary courses such as probability and increasing my mathematical maturity to some extent. The summer after my first year, I started self-studying many DL/ML courses and working through various DL/ML course problem sets so that I would get a good grade in my ML graduate course in the fall of my second year. I kept studying until I didn't have any lingering questions about the course material (including having an intuition or two for everything presented). (I did not take tough graduate courses that I suspected I would do poorly in, even if I prepared beforehand. Try to learn some of the content before taking the course. If it's too hard, consider not taking the course or taking it later. When you've self-studied a course beforehand, take the actual course and it can serve as a review. If you know the content, you can spend more time on research.) During winter break I self-studied more ML courses, some of which were not so useful such as probabilistic graphical models. In the winter I took an NLP graduate course which had a course project component. The project component meant I had time to practice doing research. In my spring semester I practiced doing research independently for maybe 40 hours a week, and I spent spring break constantly running experiments. The person who taught my NLP course let me research under him during the summer, and in that time span I became sufficiently skilled and wrote several papers. In my third year I caught academic neuroses, failed to get in a relationship, and was always lightheaded due to needing to drink soylent all the time (I was really poor)—this flattened me and my research really stalled. I did not fully recover until starting graduate school. These days, I could not afford to slip up like this since admissions is more competitive. Still I did some research with Tom Dietterich the summer after my third year to get my second main letter. I also took lots of ML project courses in my third year and did some research at the time.

Note:
We will want some students to deviate from these paths so that the strategy can evolve and so that the pipeline is diverse.
Since students are starting to learn ML as an undergraduate, starting research at the end of one's first year might now become possible. Students could also start research even earlier than that, but they may get higher returns from gaining various competencies through courses.
These are very off-the-cuff remarks.