Calculus machine learning pdf

Originally published in Andreessen Horowitz's AI Playbook.There are four main ways to teach deep learning networks: controlled, uncontrollable, semi-controlled, and enhanced learning. We will explain the intuition of each of these methods. Along the way, we'll share the terms you'll read in literature in parentheses and point out more resources for the mathematically inclined. By the way, these categories cover both traditional machine learning algorithms and newer, more bizarre deep learning algorithms. For math is inclined, see this Stanford textbook that covers controlled and uncontrolled learning and includes code samples. Controlled learning mix-up learning trains networks using examples on which we already know the correct answer. Imagine that we are interested in teaching the network to recognize the photos from your photo library that your parents are in them. These are the steps we would take in this hypothetical scenario. Step 1: Creating a dataset and categorizing, we'd start the process by looking at your photos (dataset) and identifying all the photos your parents have, marking them. Then we would take the whole stack of photos and divide them into two piles. We would use the first bunch to teach the network (data training) and the second bunch to see how accurate the model is when choosing photos with our parents (data checks). Once the datasets are ready, we will feed the photos of the model. Mathematically, our goal is to have a deep network to find a feature that enters a photo and whose output is 0 when your parents are not in a photo or one when they are there. This step is commonly referred to as a categorization task. In this case, we're charred for results that are yes-no, but controlled learning can also be used to deduce a set of values, not just 0 or 1. For example, we can train the network to exit the probability that someone will repay a credit card loan, in which case the yield is from 0 to 100. These tasks are called regressions. Step 2: TrainingTo continue the process, the model makes a prediction for each photo by following the rules (activation function) to decide whether to sunbathe a particular node in the work. The model works from left to right one layer of time - we will ignore the more complex networks at the moment. Once the network calculates this for each site on the network, we will get to the right node (exit node) that lights up or not. Since we already know what photos your parents have in them, we could tell the model whether his prediction is right or wrong. We will then return this information to the network. The algorithm uses this feedback, which is the result of a function that quantifies how far from the real response from the prediction model. This is called the cost function, and as an objective objective utility or fitness function functions. The result of the feature is then used to change the strength of connections and biases between nodes in a process called reverse offer, as the information moves backwards from the result nodes. We repeat this for each of the images, and in each case the algorithms try to minimize the cost of the function. There are various mathematical methods to use this knowledge about whether the model was correct or wrong back into the model, but a very common method is gradient descent. Algobeans has a good explanation of the non-specialists on how it works. Michael Nielsen adds a math that includes calculus and linear algebra (and a friendly demon!). Step 3: VerifyOnce we have processed all the photos from our first stack we will be ready to test the model. We would like to capture a second stack of photos and use them to see how accurately a trained model can pick up photos of your parents. Steps 2 and 3 are usually repeated by tweaking different things about the model (hyperparameters), such as how many nodes there are, how many layers there are, what mathematical functions to use to decide whether the knot lights up, how to aggressively train weights during the reverse procrastination phase, and so on. This quar response has a good explanation of the stick you can turn. Step 4: Use it until you have an accurate model, you deploy this model in the app. You're calling the API, such as ParentsInPicture (pictured), and you can name this method from your software, resulting in the model's conclusion and results. We will go through this exact process later to write an iPhone app that recognizes business cards. It can be difficult (i.e., expensive) to get a tagged data set, so you need to make sure that the cost of the forecast justifies the cost of getting tagged data and learning the model first. For example, getting labeled X-rays of people who may have cancer is expensive, but the value of the exact model that generates several false positives and a few false negatives is obviously very high. Uncontrolled training is unsupervised in training for situations where you have a data set but no tags. Uncontrolled learning takes inputs and tries to find patterns in the data, such as organizing it into groups (clustering) or detecting emissions (detecting anomalies). For example: Imagine that you are a manufacturer of T-shirts and you have a bunch of people measuring the body. You want a clustering algorithm that groups these measurements into a set of clusters so you can decide how big to make your XS, S, M, L, and XL shirts. You are a CTO security startup and you want to find anomalies in history Connections between computers: Network traffic that looks unusual can help you find an employee by downloading your entire CRM history because they are they quit smoking or someone transferring an abnormally large amount of money into a new bank account. If you're interested in this kind of thing, you'll like this review of uncontrolled anomaly detection algorithms. You are on the Google Brain team and you are wondering what is in the YouTube video. This is a very real story of YouTube cat finder studies that have ignited public enthusiasm for AI. In this article, the Google Brain team teamed up with Stanford researcher Kok Le and Andrew Ng to describe an algorithm that groups YouTube videos into a bunch of categories, including one that contained cats. They didn't intend to find cats, but the algorithm automatically grouped videos containing cats (and thousands of other objects from the 22,000 categories of objects identified in ImageNet) together without any explicit training data. Some uncontrollable learning methods you'll read about in literature include: AutoencodingPrincipal components of analysisRandom forestsK-means clustering To learn more about uncontrolled learning, try this class Udacity. One of the most promising recent developments in uncontrolled learning is the idea of Ian Goodfellow (who at the time worked in the Iosua Bengio laboratory) called generative adversarial networks in which we contrast two neural networks: one network, called a generator, is responsible for generating data designed to try to deceive another network called a discriminator. This approach achieves some amazing results, such as AI, which can generate photorealistic images from text lines or hand-drawn sketches. LearningSemi's training combines a large amount of unsadhed data with a small amount of tagged data during the training phase. Trained models obtained from this training kit can be very accurate and less expensive to learn than using all the tagged data. For example, our friend Delip Rao of the artificial intelligence consulting company Joostware created a solution using semi-controlled training using only 30 grades per class that received the same accuracy as a controlled training model that required 1,360 class labels. This allowed their client to scale their forecasting capabilities from 20 categories to 110 very quickly. One intuition is why using obsessable data can sometimes help make models more accurate: even if you don't know the answer, you learn something about what possible values are and how often specific values appear. Mathematics Fans: Try this Xiaojin Chu epic 135-slide tutorial and accompanying paper that surveys literature back in 2008.Reinforcement LearningReinforcement for situations where you are again not labeled datasets, but you have a way of saying whether you are getting closer to your goal (reward function). Classic children's game is hotter or colder (a Geckle Buckle Bean Stalk) is a good illustration of the concept. Your job is to find a hidden object, and your friends will call whether you're getting hotter (closer to) or colder (further from) the object. Hotter/colder is a reward function, and the purpose of the algorithm is to maximize the reward function. You may be thinking about the reward function being delayed and the rare form of tagged data: instead of getting a specific correct/wrong answer with each data point, you will get a delayed reaction and just a hint of whether you are going in the right direction. DeepMind has published an article in Nature describing a system that combines strengthening learning with deep learning to learn to play a set of Atari video games, some with great success (such as Breakout) and others terribly (such as Montezum's Revenge). The Nervana team (now at Intel) has published an excellent explanatory blog that goes through the techniques in detail. Stanford's very creative student project, Russell Kaplan, Christopher Sauer, Alexander Sosa illustrates one of the problems with strengthening learning and offers a smart solution. In the DeepMind article, you'll see that algorithms haven't been able to learn how to play Montezum's Revenge. The reason for this is that, as Stanford University students describe, learning agents are still trying to learn in a rare-award environment. When you don't get enough hot or cold cues, you have a hard time finding a hidden key. Stanford students have mostly been taught the system to understand and respond to hints of natural language, such as going down stairs or getting a key, making the system an algorithm for higher scoring in an OpenAI gym. Watch the algorithm video in action. Richard Sutton and Andrew Barto have written a book about learning. Check out the 2nd edition project. Originally published as Andreessen Horowitz's AI Playbook.Subscribe to get daily overtaking of top tech history! History! calculus machine learning pdf. machine learning algorithms calculus. calculus machine learning reddit. vector calculus machine learning. multivariate calculus machine learning. stochastic calculus machine learning. differential calculus machine learning. tensor calculus machine learning