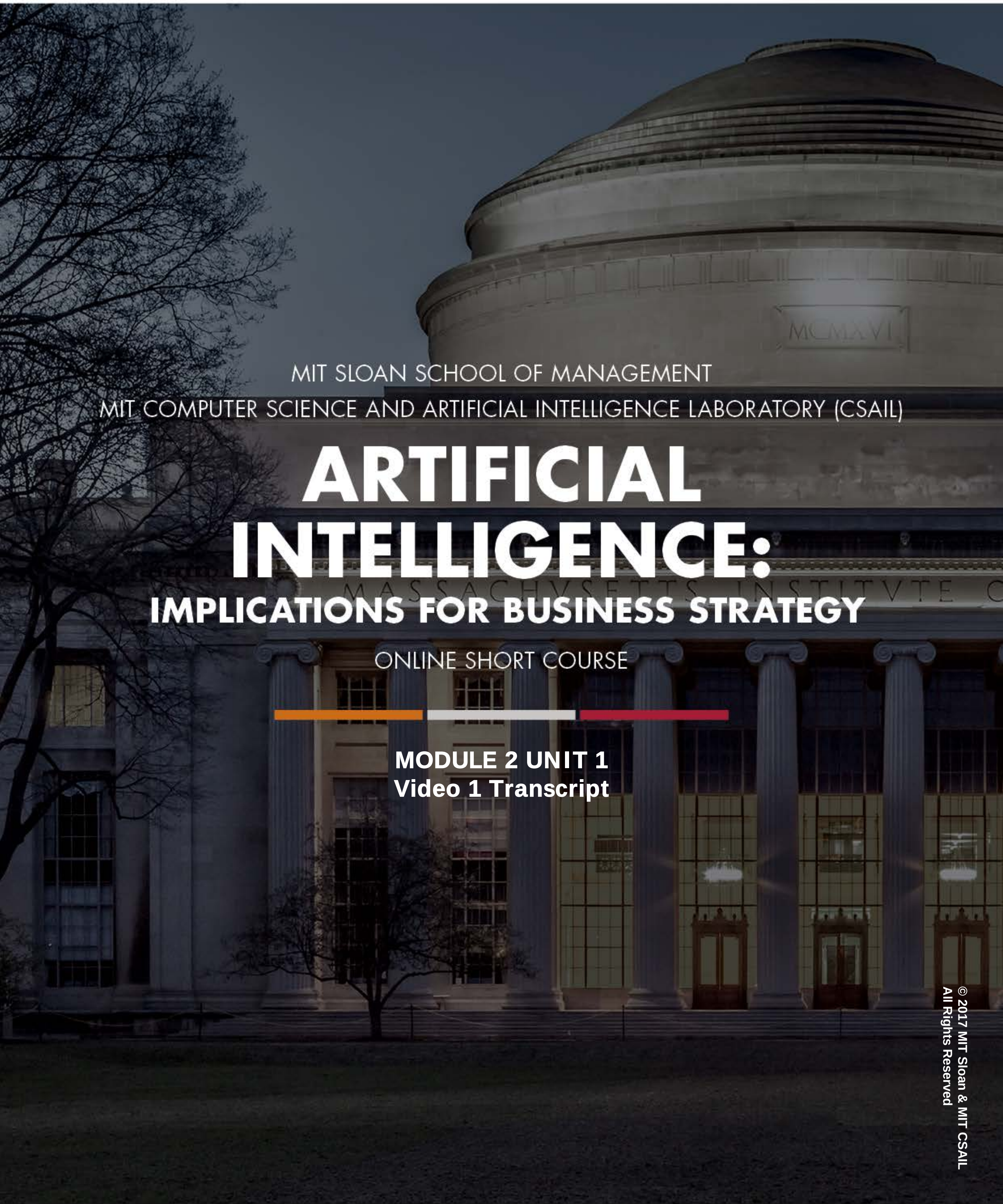
MIT MANAGEMENT EXECUTIVE EDUCATION

MIT CSAIL



MIT SLOAN SCHOOL OF MANAGEMENT

MIT COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE LABORATORY (CSAIL)

ARTIFICIAL INTELLIGENCE: IMPLICATIONS FOR BUSINESS STRATEGY

ONLINE SHORT COURSE

MODULE 2 UNIT 1
Video 1 Transcript

DELIVERED IN COLLABORATION WITH getsmarter

Module 2 Unit 1 Video 1 Transcript

THOMAS MALONE: One of the hardest parts of using computers is that you somehow have to tell the machines in painstaking detail how to do what you want them to do. The way we usually do this is by hiring programmers to write programs that include these very detailed instructions. This is often a very difficult task, and programmers have developed tools to help them do it, such as programs that automatically translate a high-level description of what needs to be done into the much more detailed instructions that a computer actually needs.

But there's another very promising approach. That is, instead of trying to get people to write detailed instructions for each different kind of problem, just write a very general program that includes instructions for how to learn from experience. Then you can teach the machines what to do by giving them lots of experience and often by telling them the right and wrong answers along the way. But you don't need to actually tell the machines the steps they actually have to go through to solve the problems, you just let these very general learning programs figure that out for themselves from the examples they see. This is what is called machine learning, and this is what Professor Tommi Jaakkola is going to tell you about next.

TOMMI JAAKKOLA: Let's start with a brief definition of what machine learning is. As a field, it deals with computer programs that try to learn from experience, and it is for a purpose, and typically it is prediction, modeling, trying to understand data, or trying to control something. Today, I will primarily focus on prediction problems. So, let's take a couple of examples of the type of prediction problems that I will deal with.

So, you can predict about future events, for example, how the market is going to change, whether there is a collision about to happen, monitor, say, a manufacturing plant when it needs service, or predict something about medical events – the risk of getting a certain disease, for example. There are other types of prediction problems as well, more about properties of things that we would like to know. So, I might wish to know whether I would like to see a particular movie, what a particular image is about, take a sentence in one language and predict what it would be in another language. What is common about these problems is that they are very hard to solve as is. It is very difficult to come up with a solution, write down that explicitly, but it is much easier to give examples of correct behavior. I can easily give examples of movies that I like, I can give examples of images, what the content is, I can give you translated sentences. So, this is how we formulate machine learning problems. We formulate them as learning from examples. So, what machine learning is trying to do, is not to specify the solution directly, but try to automate the process of finding the solution, based on examples.

So, let's say I'm trying to predict whether I would like movies. Okay. It's very easy for me to specify now a few examples of movies that I do and don't like. So, let's say I give the method access to some description like a synopsis of a movie. And I give a corresponding label of plus/minus one, depending on whether I like it or not. Okay. I have four examples of that kind here. Now, the task that I wish to solve is to learn, somehow, from these set of examples, whether I would like one of tens of thousands of available movies. So, what I have now is a training set of examples, labeled examples, and I wish to make predictions about whether I would like future movies.

The issue now is that the computer doesn't know anything about the movies, so we need to translate the problem into a form that the computer can understand. So, we're going to take a movie, its description here, and translate it into a feature vector. Vectors are things that computers can handle. And one possible way that we can make this translation is to ask questions about these movies, descriptions, and then tabulate those answers to those questions into a vector. So, what is the genre of a movie? Whether there is a famous lead, and so on. I can have a fixed number of these questions, and I concatenate the answers into now a binary vector. And I can do that for each of the movies in my training set here that I have also labeled, and I can follow the same process of translating the movie into a vector for movies that I have yet not seen. So as a result, we have translated the problem of having a set of movies, into having a set of vectors. And in the training set, I have a set of labeled vectors, and in the test set, which represents the problem that I really wish to solve, I have a set of just vectors, for which I will need to provide the label for.

So, now as we are dealing with vectors and labels, we can also translate the problem into a geometric form. So now I can take each one of those vectors and put them as a point in space, and there is a corresponding label with these points that I wish to learn something from. So, this is the machine-learning problem that the computer must solve. What makes this problem interesting is that even though this is given as a training set here, the problem that I really wish to solve, is how to classify those test examples. I don't have them, but this is the problem that I wish to solve. So, the training set just gives me clues about what the solution might be like, rather than giving it directly. Okay. So now I have to find some way of separating these examples, following, sensibly, the rule that the simplest explanation is best, if it works. So, here is a very simple explanation: division of space into two halves. However, it does not work. It doesn't even work on the training set. It's a linear separation, but it doesn't solve the training problem very well either. Here is the same solution, but now it solves, nicely, the problem on the training set. So, this is the solution that we would select, and there are many algorithms that can find such a solution. And once we have that, then we can bring back the question marks, now the test examples, movies translated into vectors, and use this classifier to classify them correctly.

Now, that's not a very interesting machine learning problem, so let's look at a little bit more of realistic problem: trying to classify images into content categories. So, in this case, the objects that I'm dealing with are much more complex. They are images, so I need to understand what the content of the images is. And what I'm trying to predict is also more complicated. Now I have a thousand different categories that I have to assign the image to, rather than just having a plus/minus label for each one. I can formulate this as a machine learning problem, exactly in the same way as before. I can give you examples of correct behavior, I can give you an image and the corresponding content category, and I can give you lots of these examples. I am, in this way, formulating the problem as a machine learning problem, as a problem of learning from examples. Now the problem is more complicated, so the solution will be more complicated as well, and we cannot just specify how to featurize the images by hand. And, in fact, Computer Vision, the old version of Computer Vision, tried to do this, but nowadays this is not how these are learned. In fact, you try to learn also the features that you pay attention to in images, in solving this problem.

So, here is an example of what that solution looks like, and it's motivated by how our visual system processes the images. If you look at how the signal goes through different transformation layers, first you recognize little edges, then combinations, then parts, and then, finally, objects. And now, the machine learning slash computer vision solution to this problem is to specify these layers of transformation from small features – how they are combined – lots of different layers, so you get this deep model, and it's fully primarized. So, you need to learn what those features are, as well as how they are combined, to ultimately make that category distinction. So, for example, this is a particular solution from already a few years back. It has tens of millions of parameters that are learned entirely from just having those examples of an image and corresponding category.

Now this is just an illustration of how these complicated models, that are learned end to end, have overtaken the field of computer vision. So, there are two points that I wish to make from this slide. One is that if you look at the top rows here, you see the error of these methods on a standard large-scale benchmark task. The error has gone down 50 to 100% per year, in a short amount of time. That's point number one. Point number two is that if you look at the distinction here between blue and red, blue are the older-style computer-vision methods, and the red ones are these complicated deep learning architectures that are learned entirely from examples. And the field was overtaken by these deep learning architectures in the course of just a few years.

Once you can solve image categorization problems, these have parallel developments in all kinds of other interesting tasks, for example, scene analysis for self-driving vehicles. This development did not happen only in computer vision. You can take a, say, machine translation problem entirely just as a machine learning problem, given parallel sentences – a sentence in one language, a sentence in another language – and giving lots of these examples, and then asking the method just to learn from these examples, to translate from one language to another. And in fact, now when you try query to a Google Translate, you're actually using these neural machine translation methods that are learned entirely from this example. I would like you to appreciate the complexity of this problem. Not only do you have to map a sentence into some semantic representation that captures what the sentence is about, you also have to now predict a much more complex object, which is the natural language sentence, take that representation and unravel it into a different language. On a high level, it is entirely the same kind of problem. You're giving an example of correct behavior, and you're trying to automate the process of finding a solution to this problem.

Now the last example on these high-level advances that have happened over the past few years, is playing games. And a typical thing that comes to mind when you are talking about computers playing against humans is, say, a game of chess, where you can essentially beat a human player by just thinking more moves ahead. So, it is a search problem. Now that's not true for the game of Go, and, in fact, it was thought, and that the method that beats a human master-level player was at least a decade away. Because for the game of Go, the number of possible moves that you have, at any given point in the game, is quite a bit more than, say, for chess. You have about 250 possible moves that you can make. So, if you think of it as a search problem, its ConvNetworks explode very quickly. And in fact, you have to learn to pay attention visually to the game board and map that visual description to actions that you take. This is how AlphaGo deep learning architecture solved this problem. So, essentially you learned, similarly to mapping images to categories, now you take an image of a game board and you learn to map that image to actions that you take. And you can learn that from just having access to expert play, what they moved in a particular context. Now you can make this even better by having the computer play against itself and learn from that as well, as well as incorporating some local search strategies as well. DeepMind AlphaGo architecture beat a European master a couple of years ago, then the Korean World Championship, and now very recently also the Chinese master.

THOMAS MALONE: Did you understand all the concepts covered in this video? If you'd like to go over any of the sections again, please click on the relevant button.