[Music]

Welcome to this seventh week of the course in machine learning. This week we will have two separate themes, the first lecture will be on tools and resources and the second lecture will be on interdisciplinary inspiration for the field of machine learning. So we will start with this lecture and talk about a number of tools and resources that can be used for the implementation of machine learning systems. This week has a number of sub themes and I will shortly go through them with you now. So the first theme is about industrial contributions to the area and roughly one can say that from the late 1950s to 2010 machine learning was more or less a research area in artificial intelligence and the bulk of results came from the research groups. There was a modest industrial instance over time but what happened in 2010 was more like an explosion of interest and because this change of interest was so dramatical I want to talk specifically about this and how the various companies are engaged from that point onwards. Having said that we will start with I would say with a natural and simple corner (02:07). Machine learning is about algorithms I hope you have understood that during this six week we have to be together, algorithms working on representations then essentially this is the classical scenario in computer science, one talk about algorithms and data structures and the next step is implementation, and implementation typically happens through programming in a programming language that's the classic scenario. So it's very natural start this discussion with this scenario but you want to build a machine learning and you're more or less program your system from scratch in a particular programming language. The next theme is about the excess of data and also about some general resources for computing, so if you want to do with any kind of serious machine learning you need a good supply and a large supply of the kind of data you need to work, and you also need reasonable resources for computation, for the execution of your algorithms. And happy enough not seems long but slightly longer than this explosive interest in machine learning most of the big software companies have put a lot of effort in develop them software support for what is now typically called cloud computing. So there are a lot of tools to support systems around for both getting services from the so called cloud for having helped for development and for also doing computations. So it's appropriate also to bring that in the picture here pretty early. The next step sub team number four, is about what I call here software development enablers, that could be libraries, there could be tools, and different kinds it could be API's etc so I will discuss that and I will go through a number of examples of such software tools existing at the moment. The fifth is about a category of systems called technical computing systems where Mathematica is such a system and of course it's also clearly possible to implement a machine learning system in terms of such tools. The sixth theme is about enhanced hardware support more powerful computers architectures more powerful processors dedicated to this kind of tasks. And the final theme for this week is about datasets and repositories for datasets because that's also a crucial component if you want to do applied and series work and in this field.

Let's start with a short discussion about the industrial interest in the area. So let's talk about companies with substantial research and development with this focus. As I said traditionally development of techniques in this area has been driven by research groups at university research each with a few exceptions. Companies like IBM actually have been very interested on a moderate level for a very long time so there may be a few other such example but not many. In the 1990s there was an increased interest in machine learning so the area of data mining was formed and data mining was primarily very applied and driven by industry, however the magnitude and the size of work in data mining at that time was I would say on a normal level, considering that machine learning already done was a serious technical area. However what has happened since 2010 is something different. It's a clear trend shift and it's an explosion of interest in this area and not so many years have gone since that that time. So what happened is that suddenly almost all influential software companies but also more and more hardware oriented companies have made large investments in building up competence in artificial intelligence and machine learning. Also officially stating their belief that this competence with subsequent effects on their products and services will be pivotal for the further successes and competitor's of this company. So not only making this silently in the background but rather very cleanly and efficiently stating this is as a way I keep the main way forward. Yeah also apart from large companies one can have observed during the same period an extensive set of successful startups coming up. Some of them grown to be independent companies but also many of them were doing in a moderate sized doing profitable exits when acquired by the competing industrial giants who want to really expand that territory. So why did this happen? So it's not easy to tell but you can at least you can mention I can mention a few contributing factors. So just before this happened we could see some really clear success stories for this field specifically related to improvements or performance in image processing and in speech processing. Of you also we could see an awareness of the possibility of optimizing performance because performance have been a huge problem historically for this area specifically for the field of neural networks. So essentially one could also say that neural network have benefited most as a subarea due to the extended presence and availability of better hardware support. Also there are some areas that has been growing and been developing and during this period and is still developing and whether prognosis are also very good for further expansion, and that are those areas are robotics and autonomous vehicles. Also we in a broader sense we can see that the upcoming presence of cyber-physical systems where we're more and more tied together digital systems, cyber systems with physical systems and very close to that the concept of Internet of Things. So because we got very much more of this interwoven systems both of Hardware and software, the impact of AI and machine learning algorithms are scaled up. We can have one algorithm that existed for 30 years but 30 years ago the overall technological presence in various sectors of society was not so big, so therefore the effect of applying this algorithm is smaller but today the impact has grown. So just to sum up the industrial interest of course had a big effect, it has an effect on research of development but it particularly also have a very strong effect of the availability of tools or resources because if you want to have extensive development of this kind of system in industry, industry needs tools and resources and a lot of resources have to be put into that development.

Let us now look at some of these companies that has been engaged in this area since roughly 2012. And up the top you have Google. So Google has worked very hard to establish themselves as a key player on this arena, So even since 2010 they have a deep learning AI research team for Google brain, they also recently in 2017 announced a whole new division of the company dedicated solely to artificial intelligence and finally the alphabet holding company for Google has also bought and taking over smaller companies one example of an important company being taking over is the deep mind technology company also focused in 2010 that have shown immense success in various kind of game playing applications. So this is Google. Yeah another company has also been very early on this but a little later than Google, is Amazon. So initially the focus was very much on the improvements of Alexa, Amazon language assistant system which they also worked hard to integrate into their eco speaker series system but soon they also looked more and more on to their what they call the cloud platform of Amazon Web Services and put a lot of efforts of augmenting that platform with machine learning components. So Amazon have a lot of power obviously and even if they started a little later they are an important player in this game. IBM has been very active since the 1950s so they are an exception they've been active for a long time and they are still active, and they created a pretty famous system called IBM Watson was originally more of a question answering systems, problem style solving type but what they have done now is also more and more created software tools that various kinds of user can access sub functionalities of the original Watson system and they also seen to that these kind of functionalities are available into also the IBM cloud services. We should not forget Microsoft also had a well-known and very much use cloud service called Azure and also here Microsoft have worked hard to use machine learning components and of course also Microsoft has been interesting in having their own to digital assistant system in this way called Cortana and Microsoft also tried to strengthen their position here also by purchasing a lot of smaller AI companies, 5 companies only in 2018. Facebook is also engaged has an AI research group called FAIR and they recently recruited top people specifically from neural network like Yann LeCun which you heard about in earlier lectures and they also developed and support competitive software packages - for example Google. Also a European actor SAP are engaged they also have a cloud platform which they in the same fashion developed in the machine learning direction. Finally Apple has been very late obviously in this but as it seems they try also now to get into this at this game. They're also hardware companies that that are important but are will take them up later in the lecture.

Let's start with a scenario where you want to develop a machine learning system from start from scratch in general programming language the classical scenario. So which programming language should you choose. Two languages that been used a lot in artificial intelligence for a long time, Lisp and Prolog typical representatives for the functional style of programming and the logical programs that style of programs. However if we look at the situation today for industrial machine learning applications these languages are not a strong rule. Actually for real-life machine learning systems we can observe the dominance of languages that support the object-oriented paradigm. During the last twenty five years I would say the triad of languages C or indeed (19:33) with C++ to become Oregon (19:39) Java and finally Python have dominated to scene, however over this period changing their relative top position as indicated on the slide where you can see that originally C, C++ was very dominating over time, Java became a more dominant and finally at this moment the most popular and most widely used general programming languages Python. And then there may be many reasons for that I mean python is a really good language so of course that can explain don't think it's a very clear and simple and straightforward language but also it's the fact that python is a language that easily lend itself to programming in many paradigm. So probably for machine learning and the future lies in multi-paradigm languages because very much of the history because payment of the history as I already told were lying in the in functional programming logic programming so of course it's very nice if you have a language that could be efficient like C but not only imperative also object-oriented functional logic oriented. So this is some kind of explanation where things are moving. There are also other general languages that are coming up, once of language is the R language where there is a very strong specialization in the language towards statistical computing. One can also say that for certain applications where efficiency and real-time issues are still very important, this combination of C and C++ still serve us a strong position. So let's now go and look at through a list of this kind of languages. So as you see here currently Python as at the top. Most of these languages are open-source this means that you can access this language you may have a license but it would not cost you anything and many times the restriction what you can do or cannot do is pretty liberal. Second place at the moment still comes Java and third place as already C/C++. Also in the kind of if you look at top group for popularity this language it is included. Then there are variety of other languages that are in this ballpark (23:09) so you have languages JavaScript and which is also popular the of languages Scala used in some context, you are other special languages like Julia coming up but as you see at the bottom here Prolog and Lisp are still there but not very much used in practice. As I already said machine learning is not only about algorithms. Very essential aspects are the access to relevant data and the access to necessary computational resources. So as a consequence the natural basis for machine learning systems are the system's for distributed cloud computing provided by at this moment all majors of related companies and the common term here are cloud systems, cloud platforms and essentially what you can get out from the cloud is as a end user you can get services, this means that you can come this all the distance(24:43), you can acquire services you don't have to worry about how these services are realized, so you only have an end-user experience so this is what people in this field abbreviated SAAS, software as a service. And the next step which is not so much for end-users but for developers that people talk about PAAS which are platform as a service. So not only specific services are provided but of platforms for development of new applications are provided via the cloud system. And finally you have what is called abbreviated as IAAS which is that you can also get access to more or less infinity infrastructures, computational and data storage resources via the cloud the system. So I think the trouble of taking this up here this is not machine learning this is distribute computing, this is cloud computing. However it's a very very clear tendency and I already touched it a number of times in this lecture that almost all of these cloud platforms are now or from last few year's and presently augmented by components that supports machine building development of machine learning systems.

So look at the various frequently used cloud computing platforms, the same both players occur. I guess everybody has heard about the iCloud but also Microsoft has a very ambitious cloud computing system call Azur, Amazon Web Services in a similar way and SAP Leonardo so all these big software companies have their own solutions here, but they're also as you can see below some more clearly open source oriented even if some of them is also now our open source even if they are developed and promoted by these companies there are also these kind of solutions that originally also come from a purely open source environment. An example of that is our system related to the foundation of Spark which is open-source software foundation but also today there are new alliances form actually, so for example and this Hadoop System often occurs in the context of collaborations with IBM. So this is an arena and yeah in a minute you will see also more couplings coming up to the extensions to machine learning.

Just recently I started with talking about developing machine learning systems from scratch in a general programming language. Actually this is today not the typical thing it is not only for machine learning for any kind of software development the typical thing is not to develop any code from scratch. Typically what you do is you combine existing, I would say atom or molecules of software code from libraries that already are program and tested and developed to perform some functionality to the task of programming today is rather combining pre-existing modules, of course you have to program somewhat typically I think it's rare than you can only build something from this pieces and put them together the typical thing is that some programming is still needed, but the bulk of the work and a bulk of the system functionality comes from the combined functionality of the atomic block blocks. And today therefore there is a whole range of what are call here software development enablers of various kinds named with a lot of different names. But a pretty straightforward though to start with something his software library so software library and a set of consists of subroutines for important algorithm function for targeted application areas and it could be an narrow or it could be broad. So this is always important and it's been for a long time and it's still there. But today there are also other tools so people talk about integrated development environments IDEs which is our tools that not only provide access to a library with a subroutine but also with some software components that automate some useful processes of the development such as debugging, code generation etc. And then on slightly larger scale, people talking about software development kits where we essentially can combine a number of this IDEs together. Even larger than people talk about framework. And then you also have this important concept of application programming interface API where an API as should be because of the name, for the name interface is rather the interface to these tools and to these libraries, so an API one could say the logical representation of what is in all these toolboxes that you want to access. It's not a priority here of this like to classify name is all these enablers that exist in an area for machine learning according to these categories because the borderlines many times are blurred there may be many reasons why people call something what they saw that, so it's not really meaningful to try to achieve very short boundaries here, especially not in a state where we have a very intensive in development going in these areas. Unfortunately one can say that because of all these concepts and because of that so many things are developed and so many things are interrelated it's not entirely trivial to understand that this fast growing forest, I have used an analogy of software tools. But I will mention a few, so in the in the next few slides I will mention a few of the enablers that are relevant if you want to build machine learning systems.

Let's look at a few of the software development enablers and there are various kinds, so you can see on the top of the list you can see a system called anaconda which is basically a distribution channel or a platform for distributing Python and R programming language in a convenient way for the users. And also handling the program libraries that comes with these languages in an efficient way. Then there is a category of support systems that is strongly focused on support for building artificial neural networks. So now say not only one can say that many of these tools are support tools for doing efficient multi array calculations because as you have understood from earlier lectures it doesn't matter where you start from if you have a vector machine or artificial neural network many times you can transform those systems onto computational problem expressed on multi-dimensional arrays. So anyway Tensorflow is that the system it is one of the primary tools that is pushed by Google and probably is one of the most popular tools at the moment. There is a competitor to Tensorflow called Pytorch which is sponsored by Facebook and both these are open source so even though they have a very strong company coupling there they are easily available. And both languages are very very clearly coupled to programming in Python, primarily not only but primarily. Then we have a similar Microsoft system called Cognitive Toolkit which have a mixed Python C connections. Then you have other systems. One issue here is that many of these system are interlinked which may it's probably good idea because it's useful but when you start to understand what's going on it's a problem because a system like Keras it's a system of its own but it is a system that could run on top of some of the others like Tensorflow, Microsoft cognitive toolkit and so on. Okay so these are the systems there are similar systems Amazon has also a toolkit of this kind and as I already said even IBM now tries to more and more and offer at least a lot of sub functionality to the original Watson system in various contexts for their customers.

So here you find more of the system. I like to focus mostly here on the top two on this slide because this system not only should need to provide efficient access to all the range or machine learning algorithms, many times it's very important when you when you run a machine learning systems project to be able to coordinate a number of resources and a number of steps, and one such support system is this Jupiter notebook which is essentially which emanates from a similar project developed product even this project called ipython, so they these two go together. Essentially what this these kind of support system try to do is to help you to coordinate work maybe in different languages because you don't necessarily have to just work in one languages but in a variety of toolboxes and also to coordinate your data and also to help you to present the results in an efficient way. So this is like a very popular complementary support system in the machine learning context as this very mean. So the rest of the systems here are more just more of the same. I mean for good or for bad there are many powerful actors here who want to have their market shares and all the factors develop their own systems in a way competition is healthy because it's sharpens and push the functionalities but it's also I would say somewhat confusing and because it's not so that there is one tool that is best for that another tool for that and there are just a bunch of tools for each of these actors and some are more popular, some are less popular and this develops over time. So simply my advice actually here is that very much what you chose is depending on the context yard I mean if you're an independent person, if you're a student and you have no special connect you I think everybody has a connection even you're a student you have a context you have your University and it may be so that your university or your department or in the research group you work together with they have made some choices, so they decide if they want to work in this environment and not in that, so and then it can be what doing to make something useful it's always clever to use the competence of the cop of the context you are in. Even more if you come to a company it's very clear that most companies may made some choices here, so this means that the actual choice you probably have as a person in this kind of setting is rather limited but because other people have made a number of choices for you.

I also want to introduce your little to another realm of support system what is called here technical computing systems and so technical building system is the application of the mathematical computational principles of scientific computing to solve practical problems of industrial interest. So it is not scientific computing its technical computing. So this kind of system are dedicated software system for support of this kind of computing, so typically they comprise implementations of a great variety of key algorithms in applied mathematics and some extensions of that, but also some general program compatibilities. So even these systems have some kind of programming languages built in for doing this extra tailoring needed to glue the pieces together. Increasingly so now technical computing systems comprised implementations directly explicitly of key machine learning algorithms. I would say some time ago you could use this kind of system for implementing machine learning algorithms but they were no pre implemented such algorithms you had to do it yourself in terms of the mathematical tools, applied mathematical tools away but today there are more and more of these classical machine learning algorithms built in. So essentially there are three categories of systems, so for a long time there been a numerical analysis software system that really supports numerical analysis problems solving but there also a category of system that started developed in the 60s called symbol manipulation systems where essentially solve mathematical problems symbolically, not numerically. I would say today the most widely used systems are not extreme in any of these sense the system use today at least for machine learning are hybrid systems. In this stronger (44:39) it's more common that the systems are proprietary than open-source in a sense that you have to pay not just little but probably reasonable sum to use these systems. And there are a lot of restrictions also then of course off of the chose.

I'd like to say a few words about some of the most widespread technical computing systems. I believe one of the most well-known and popular are still Mathematica but it's a proprietary system, it's provided by an organization called Wolfram Research and it's based on its own general programming language called the Wolfram language. So this is very much useful system it's a closed system. Maybe as a reaction to that at least some open source alternative have been developed, so an example of that is the system at the bottom called SageMath which is open source and that system was released in 2005. Otherwise the competitors in the original shiner of systems to Mathematica I would say primarily our MAPLE and MATLAB. So MATLAB in contrast to Mathematica is Primarily a numerical computing system in contrast to Mathematica who is also so it's a balance hybrid and what we say on the other side of the camp you are the system like MAXIMA who are really developed from the symbolic side the computer algebra systems. One general comment here that I made also earlier is that for this kind of genre you are probably very much constrained in your choice by the moment, your a typically University design typically company decide whether you want to use one of these systems either you are an organization that was Mathematica or use Maple or any MATLAB, you don't typically use all these system means because I'm kind of mix it, so depending on where you are now the University and our nature at the company we will probably them yeah the choice you have is to use what the company already and decide it for you.

So the next theme and it's about hardware. So one very important factor in the success of the recent successes of machine learning is the better hardware support, so what is mentioned a lot today is the existence of what called AI chips or AI accelerators and machine learning algorithm particularly there artificial neural network once, demands more computational power than what can provide it by conventional CPUs. So new computing architectures are needed and these terms are used AI chips AI accelerator or neural network processors because a lot of the efforts recently have been on efficient computation for neural networks not only but primarily. So what I mention here on this slide are kind of four trends for streams of work, so there is something called heterogeneous computing which has been going on for a long time where you actually design complex compute computer architectures where you combine various kinds of specialized processors with conventional ones and you even embed them on the same chip. So this is a kind of multi-core tailor-made tailored processor structures and it has been one way of achieving better performance, what is talked about mostly actually and you probably have heard about that other is the use of graphics processing units GPUs and essentially the story is that these processing units were developed with because in the gaming industry primarily the way you have a lot of visualization a lot of graphics you need very high performance. So this kind of architectures was developed in that field however it turns out that and hopefully you have got a feeling for that is that mathematics behind this are pretty similar when we talk about neural networks and image manipulation so therefore it wasn't a big step to realize that it would be a good idea to use or not only just to use but to adapt these kind of processors for their machine learning tasks. So GPU is still probably the main trend in development here but there are also other avenues there's something called field programmable gate arrays, which is another architecture philosophy for processors and then something called application-specific integrated circuits. So one can say that the GPUs have triggered the process and they are still dominating but computer architecture is a broad field and there are many ways forward and not everything happens on the graphic processing tracks also the other streams I mentioned here where our work coming on. This leads to the hardware companies in their own, so what one can see is that hardware companies that traditionally wasn't really interested in artificial intelligence and machine learning suddenly when they realized that this technical sector could be a really good avenue for increased and strengthening of their customer base, they also to some extent also engage themselves not only in the hardware part but also in the software aspects. So you can see that companies like Nvidia who is the key player on the GPU side, companies like Intel, Qualcomm etc. they also lift their eyes to the more tool to the software aspects, also one can see the opposite way is that companies traditionally so specialized on tech focused on specialized hardware Apple Samsung Google Microsoft have started to interest themselves in these specialized processing processor architectures, because of course for the reason that they do not want to be entirely in the hands of the other can't category of companies. So there is a jungle of systems coming up here so they're all I mean of course NVIDIA is very well known and they all all the time come up with new more efficient versions or models. But also you can see on the list here Google themselves produce processing units, so obviously now this has become a field where the software and hardware companies be. So there is another aspect which is important here for this meeting between hardware and software it's because when this development started it was a very tricky to use this special kind of processor, so this means when you build the system you have to kind of tailor your solution and your system for the explicit use of the GPUs which is possibly tricky, so there is actually a category or programming language and tools developed now you can call the GPU languages but you can also call them general gpgpu which means general programming based on GPUs and there are a few of those who are well known I think the most well-known is called CUDA and CUDA is very much coupled to an Nvidia but there are also others coming up OpenCL, Harlan and so on and it turns out of course that if you have a tandem development a very good specialized processor systems and also that you are the driving force and you are monitoring the development of a popular language of this kind, you have a competitive advantage but because your language quotation mark in a way is always slightly biased to the kind type of professors to you push for, so therefore we can also see now that all the hardware companies have understood this and also engage themselves therefore in this kind of languages that way you have a strong support for getting help on how to utilize the specialized processes in your system for the purposes you want.

So these were the repositories where you can assume that the data sets are have a more guaranteeing level of quality, however there are now a lot of other open data sets, so down up close you can go to them and back of course there is no guarantee that for every data set in these repositories it's a trivial task to just use the data normally in many cases you have to have some pre-processing of the data from these data sets, so on this last slide you can see some more examples you can see some example the meaning of the big software companies like Amazon Google Twitter etc. They collect data and they also exhibit this data in an open fashion but there are also large organizations like US government World Bank who also make datasets available in an open fashion. So this was the end of this lecture thanks for your attention, so the next lecture in 7.2 for this and the last one for this week will be on interdisciplinary inspiration sources for machine learning thank you