**Interviewer 1:** [00:00:00] Okay. Okay, so I will start with some background questions. Could you tell us what is your current position at your job or?

**Interviewee:** [00:00:13] Okay, for about one year I am in the machine learning business. Okay before that I was a WordPress developer.

**Interviewer 1:** [00:00:27] Okay.

**Interviewee:** [00:00:28] I was relatively successful as WordPress developer, but I decided to create some turnaround in my career and, so I investigated in machine learning my time, and I decided to dedicate everything that I have at this moment, and this is why I'm here right now and I believe it's the future, right?

**Interviewer 1:** [00:01:06] Okay. So do you work in industry right now?

**Interviewee:** [00:01:10] No, at this moment I'm still like prospective candidate for some job, but I don't think that I do have enough experience to compete or to get the job right now. I'm almost there, so I am almost there. I do have some background in mathematics and engineering industry. Because I do have this faculty of electronic engineering and stuff like that. And this is my basics.

**Interviewer 1:** [00:01:57] I see. So good luck in your new career. Yeah. So could you tell us your, like you told that you have experience in machine learning for a year? Could you tell us about your overall work experience?

**Interviewee:** [00:02:11] So my overall work experience would be, I started as a C developer and C++ developer. And after that I worked in PHP as a programming language. And it was almost natural to go to the WordPress, because it was the part of the industry where PHP is based, and I deal with websites and things like that, like browsers, everything that goes with it, with the design of the website. And right now I decided to go to the machine learning.

**Interviewer 1:** [00:03:04] Okay. So how many years of overall experience do you have? Work experience.

**Interviewee:** [00:03:08] Well, let's say, 12 years.

**Interviewer 1:** [00:03:14] 12 years. Okay, thank you. Could you tell us what type of deep learning or machine learning networks have you developed? And by that I mean supervised/unsupervised/reinforcement learning, things like that.

**Interviewee:** [00:03:28] So I worked on the, at first it was a little strange, but I worked on the LSTM networks, this VDRNNS(?) and this was part because I was really interested in that. And after that I switched to the convolutional networks. And I think I'm right now at that point. Of course, in the between I learned about some other techniques like the tabular data and things like that, how to manage that, actually I worked a little in Pandas in order to prepare myself for that job. And also because I worked on Python, on my Python knowledge, to improve that, because it was important for me to have a good background in there.

**Interviewer 1:** [00:04:29] Okay. Okay, and which problems you were trying to tackle using this machine learning networks, like image classification, speech recognition, any other.

**Interviewee:** [00:04:44] So I worked on the, for instance, on the RNNs. I tried to, I was on the Romeo and Juliet novel, and I tried to create something like the novel of my own. So, I created some sentences from there and it was interesting at that point, but then. Can you repeat the question, please?

**Interviewer 1:** [00:05:11] Yes, I was asking what kind of problems you were trying to tackle using machine learning or deep learning networks. So what problems were you trying to solve with the help of this networks?

**Interviewee:** [00:05:24] So. The problem, I didn't work actually on the problem when I was dealing with the RNNs, because I was just experimenting with the RNNs. I followed the ideas I had online and started to open the first problems, basically because it is important in what software you work. At that time I was on Tensorflow and I thought this is the best environment for me to work with, you know.

**Interviewer 1:** [00:06:12] Okay.

**Interviewee:** [00:06:13] After that I switched to Pytorch. Maybe this will be your next question. But I switched to Pytorch, because I liked it. Yeah, I can compare this two.

**Interviewer 1:** [00:06:27] Good. So, yeah, you were kind of right, my next question was which programming languages and frameworks you have been using. So you told us that it was Pytorch and Tensorflow, and I guess as a language you were using Python, right?

**Interviewee:** [00:06:41] My deep understanding is that you must decide on the language you'll be working on and currently Python is the best choice. So you can work in R, you can work in Lua, maybe Julia, if you heard that language, but it's somehow Python is currently the best choice, as I would say.

**Interviewer 1:** [00:07:12] Okay, thank you. So, as I said when I was talking about the objectives of the interview, we have one general question from this interview, which is what kind of bugs/problems/challenges you have faced When developing this machine learning systems. And usually when we interview developers, we ask them about the only their experience, but as you also very active in StackOverflow, maybe you could tell us about your own experience and also about the type of problems people ask about on StackOverflow very often. So if you could start from there, that would be very useful.

**Interviewee:** [00:07:49] Okay, I will start by comparing Pytorch and Python, this is very important and maybe this will shed little light to the problem that I had. When I started with, sorry, with PyTorch and Tensorflow, when I started with Tensorflow I saw that they have great examples. Their website was okay, relatively okay, pretty slow for the today's standards. Right? But they haven't used good engines to make the site work better, and I saw the great examples, and I saw they have that thing called TensorBoard. Okay, which is a graphical environment where you can work and present your work, whatever you do, you can create these diagrams and you know things important. So it makes things easier, so you can experiment little better. This was a great thing at the moment, and when I first created some TensorBoard graphs, I was a little out of myself, because it was really because, you know comparison what of the strategies is very easy with that. But soon after that I realized one important thing that Tensorflow is not having good terms. When I said the terms, I mean terminology they are using, the name of the functions. It looks like, confusing to me, and after that I investigated and I saw that the other people also seeing these problems, leading to the very bad terminology in overall. Some have even compare this with like, well, devil terminology. Really, for some people this is very important, because when you do things you would like to have the best possible names you can use for the job that you do, and they don't care about that. Actually, even when I saw on GitHub some requests, issues to alter the names they set at the first point, and they are pretty rigid at that, so they will not alter the names afterwards, which is not okay, because some names do not have any sense. And this was a part of the destruction that I had at this moment. And after that I saw that you must prepare, you must be absolute expert to set the system, unless you inherit some system from some other guy that created that, and he is the expert. But you never know that, because this is a problem, right problem you can emphasize, that online on internet you can see every kind of the worksheets or programs, but they may be wrong.

**Interviewer 1:** [00:11:22] Okay.

**Interviewee:** [00:11:23] You must see every bit of the program to be sure, or absolutely sure, almost sure that this is a good stuff. So I saw many things and really many of these things do not have any sense. They are trying to achieve that in the wrong way and so on, so it is very important to have the expertise in literally every part of the work you're doing. Starting from the loading of data and then how do you process that, the format of your tensors, everything must be very precise.

**Interviewer 1:** [00:12:10] Okay.

**Interviewee:** [00:12:10] A small coefficient of two may create some big difference on the outputs, randomizing the inputs, shuffling everything must fit to have a great model. So I was not very happy with Tensorflow, because even though I have great examples, I find it somehow hard to work with and I cannot now explain in detail, but just that information may be okay for you now.

**Interviewer 1:** [00:12:46] Yeah

**Interviewee:** [00:12:47] Then I started to analyze the PyTorch literally by accident, because I saw one great example, this happened for instance five months ago. And I analyzed it, and I saw how nicely written, how close to python it is, close to NumPy and so on. And I love, like I actually understood that PyTorch is the thing for me, because it is simple, it has great syntax and so on. So this was the main reason and also at that time I saw an interview, or not an interview, but an article from, let me just put the information for you from the, you know he is now the the main guy in the Tesla, he created the Quantum Net JS(?) . I don't know, I cannot recall his name. But generally he said that PyTorch is also astonishing experience for him. He wrote something on the Twitter like, I couldn't believe this is so good and something like that. So he triggered me also.

**Interviewer 1:** [00:14:23] I see.

**Interviewee:** [00:14:24] And. So this is the most important thing about the Tensorflow and PyTorch. I think it's easier working with PyTorch, probably just like natural.

**Interviewer 2:** [00:14:43] One little question. So as I noticed on Stackoverflow, there are much more issues for tensorflow tag rather than for PyTorch. Do you think it's because less people now use PyTorch than Tensorflow or because it's simpler so people face like few issues.

**Interviewee:** [00:15:08] So the reason for that is because currently the community doesn't know about PyTorch much. Okay, it's like hidden somehow and maybe this was the intention. When I analyzed the later studies from DeepMind, they never work in Tensorflow, for instance. For that project that they had, about Go game, if you've heard of, so they work it in PyTorch. This is completely done in PyTorch. The code is online. So why do they work in PyTorch? Because they find, even though Tensorflow is Google's technique, you know, but they are the Google main guys for the brains. They're reducing the overheating of the computers by using these techniques, you may heard of. So these DeepMind guys, they are using PyTorch. So it's great choice for... It's a great that I think that you understand that the best guys in industry are using PyTorch at this point. The number of the questions on StackOverflow. I really don't know why, but I think that they're following Google and Google advertises Tensorflow and somehow they get in there. So the greater number of people making greater number of questions.

**Interviewer 2:** [00:16:49] Thank you. Thank you.

**Interviewer 1:** [00:16:52] So could you tell us more about like specific errors, bugs that you have faced very often, like in your own experience and people ask a lot about these things on StackOverflow.

**Interviewer 2:** [00:17:07] Even small bugs.

**Interviewer 1:** [00:17:08] Yes.

[Long explanation of Interviewee’s ideas on how to build a model that will determine if the person will conduct a crime again.]

**Interviewee:** [00:22:17] You were asking in general or you know any technology like Tensorflow or PyTorch, or just in PyTorch.

**Interviewer 1:** [00:22:28] Any! Any bugs, problems that you had or you saw people having quite often.

**Interviewee:** [00:22:33] Well, at this point all the people, because this is very new, all the people are just learning things. This is my experience. And the ones who know actually do not go to the StackOverflow. So I believe that the people that are in the industry do not find the reasoning to go to the StackOverflow. My primary reason to go to StackOverflow is to learn at the same time and also to find possible job opportunities for me, where I could, you know, based on my answers get some career change, some job. So the problems are currently totally unexperienced people go and ask questions and for some, for most of the problems. So, these are pretty easy to understand and to solve. You, basically, just how to solve these questions, you basically go to the PyTorch forum where you can list all the solutions and you can create some mark of the overall problems that we have in PyTorch. And there you can find by the error codes or error names. You create, for instance, some analysis and then you can answer pretty much all the questions on the StackOverflow. So this is the idea. After that, there are few questions. They are like methodology, how they will work, and they are just asking what we should do and things like that pretty much. Although these questions do not have the answer or do not have the correct answer, because nobody knows that, because if you see, if you are just watching in backwards, the idea of GPUs and using the GPUs is relatively new, it's related to NVidia and the Cuda software, that apparently just recently a few hours ago started to go on live and to have some significant impact. Right now we have the technology, actually at this point the state of the machine learning is shaping, the best results from yesterday are now easily achieved with some software tools. The "state of the art" so called, that the academic society had, are just overcrossed with literally one notebook or Jupiter notebook. For instance, using the Fast Tie(?), I think you heard about Fast Tie(?) . It's a technique based on PyTorch, that people are using to create fast solutions for the machine learning problems. And also Allen and NLP, if you saw that, is another technique that can boost your natural language processing career. So, by using these libraries you can go and fast forward all the problems, but these create the emptiness and voids in your knowledge, because you do not understand, you use that, you create the solutions, but you do not understand how this works. I think the best engineering practice is to have the solid understanding of how the things work and then build on that and to have, you know, like if you do not understand something, do not eat for a day or two, but just think about it problem and it will open for you at some point.

**Interviewer 1:** [00:26:48] Okay, but that's an interesting advice. Okay, so you told that the people who ask on StackOverflow are mostly beginners, but like even for these people who are beginners, do you see any categories of bugs and problems that they ask about very often.

**Interviewee:** [00:27:12] Well, the missing dimensions would be one. For instance, you're creating the convolutional network, you have the input and you have the output of the network, and then the model does require some input shape that you do not provide, and this creates the so-called M1-M2 problem. Okay, you literally just need to understand what is M1 and what is in M2 dimension and to match these dimensions at some point and you are done. So there are the problems like that. Also, you know the problem of saving the model, for instance, in these StackOverflow question that appears at the very top. They're appearing at the top just because often there is a need to save the model, and nobody knew how to save the model at first point, and they just find that on StackOverflow, and this is why this is growning into, you know, many clicks up up and.

**Interviewer 2:** [00:28:24] What more could you think of?

**Interviewee:** [00:28:27] But, well this would be one biggy. Very important question - missing dimensions or non-matching dimensions. The second thing would be, for instance, not using the right optimizer or not using the right loss function. Because the loss functions influence on the strategy how you solve the problem. For instance, you cannot use MSE function all the time. You must use some other criterions. So, it's basically how you understand how your network should learn. And you need to select the correct loss function and the people are using one, because they think this is okay, so I can select any of the functions, but it's not like that. You must really understand why do you pick a function, the loss function of the criterion. So the non-matching dimension problem is also not picking the right criterion. Because, it is involved based on the what loss function you will have the other dimension for the input.

**Interviewer 1:** [00:30:05] Okay, anything else?

**Interviewee:** [00:30:09] I would need to analyze my answers in the other questions too. Right now, there is nothing popping up.

**Interviewer 1:** [00:30:17] Okay, so. In your experience when you are training networks, as you said, you're experimenting. So I would guess that you've been using existing datasets. So you were not collecting information of your own?

**Interviewee:** [00:30:33] No no, you were asking me if I was using only the well-known datasets, right?

**Interviewer 1:** [00:30:42] Exactly, yes.

**Interviewee:** [00:30:44] Well, I used the well-known datasets. But they are often wrong, if you know. Maybe many people do not know that, but they are often wrong.

**Interviewer 1:** [00:30:56] Could you give us examples on how wrong they are?

**Interviewee:** [00:31:00] MNIST. The only thing why they do not have the 100% precision accuracy is because there are some wrong cifars in their.

**Interviewer 2:** [00:31:10] Labels, you mean?

**Interviewee:** [00:31:12] For instance, there are five and the label is six. Things like that. This is actually five and you see, you know, the labeler was or lablerian was incorrect when set the label, so this is one thing. And...

**Interviewer 1:** [00:31:31] So, just out of curiosity, of how many wrong labels you know in MNIST. Like how many wrong labels are there?

**Interviewee:** [00:31:41] You know, just if you are asking about the number of labels, they're just 10.

**Interviewer 1:** [00:31:47] No, no.

**Interviewer 2:** [00:31:47] How many mistakes she means.

**Interviewee:** [00:31:50] Four or five. But this leads to that nine point nine nine nine six result. Because this is one thing that people do not know, for instance. But the careful observer will find this, when, you know, you go and create the model, and test, and use the best technique, and then still not having one. Then when you are checking this, you will see that obviously this was the error in labeling. So, things like that. So this often happened, but it's not terrible. It's just okay. But the idea currently, the most famous idea is to create your own labels from Google. Not your own, images, I think, when you're working on the convolution to create your own data set from Google. So you have the images from Google, you take these images, happened to be that I worked on similar problem and, you create for instance... I created a problem, like American football or not, no, soccer or American football, you know. So based on that I have pretty much 99.9 success. When I provide the image from American football, it classifies as American football, this is the basic classification problem. So I took about, less than 100 images from Google, and just like that and using the so-called, using the idea of pre-trained networks, I created the model. And based on that I get this accuracy. So, just people are thinking they need a huge amount of data, they do not in most cases. In your case you do [talking about PreCrime].

**Interviewer 1:** [00:34:13] Okay, so about training data, is there any other problems about training data that come to mind that people often face or ask about?

**Interviewee:** [00:34:25] Yes, yes. The problem is for instance they are missing manual some manuals in PyTorch. Nobody created a manual on how to load MNIST, for instance, correct?

**Interviewer 1:** [00:34:36] Okay.

**Interviewee:** [00:34:39] Even the manuals that you can find online are totally wrong. They are using some normalizations that nobody understands why they do that. They're trying to create some standard deviations of one and mean of zero. And someone evaluated some coefficients to use in PyTorch and based on that to work, but this is totally wrong. For instance, nobody actually answers on this question, even on the PyTorch forum. Nobody understands why they are doing that, this cannot be measured, this can be repeated somehow. So there are many voids in the knowledge in there, but they just told you just use that. For instance, what is missing what I see that is missing is loading data lazily. So when you're on the GPU for instance, you have a big problem and you need a lot of memory, but you can not fit all the images in the memory. So, you need to load images in batches from the CPU to the GPU. Using that CUDA stuff and you will need to understand that you need to load things lazily or on-demand, which will definitely slow up your process of learning, but you need to have the Python knowledge how to do that. So, I asked it, for instance, between the main guys on the forum on the Pytorch, if you can help with that. Nobody actually answered these questions. I don't know why. But they said it is possible. Okay. I saw some other guy said it is possible and I know it is possible, because I am working on this problem and I almost created solid solution and so on. But nobody have this, like in TensorFlow there are these great starter examples where you can just open this notebook, and you can start and evaluate and everything will be just fine. You will have the great accuracy in the end. In PyTorch... This is called the problem of the missing manual. Right?

**Interviewer 1:** [00:37:04] Okay.

**Interviewee:** [00:37:05] What is missing? It is missing the code, how to do that in the optimal, in the best way. You can find manuals online on the internet, where you know picking this problem, but these are often wrong approaches and you will really believe that this is good approach and it is actually wrong. Because it is using some normalization technique and trying to do something that is not smart.

**Interviewer 1:** [00:37:33] Okay. So you said that you have this system to identify whether it is soccer or American football. Did you label that data by yourself?

**Interviewee:** [00:37:42] Well, definitely, when I asked Google, I asked, for instance, 50 images of American football and he provided me. And I just, I labeled these images and then I asked...

**Interviewer 1:** [00:37:55] Sorry. So you were the only person who did the labeling, right?

**Interviewee:** [00:38:00] Yes.

**Interviewer 1:** [00:38:01] Okay. Thank you. So another question I have is about pre-processing of training data. Did you ever had problems because of missing pre-processing or wrong pre-processing. And do you think in general based on your experience on StackOverflow that this is a problem for the developers to find out how to correctly pre-process?

**Interviewee:** [00:38:23] Definitely. This is the great problem, because everything is in these preparations that you do, everything. If you normalize the data well at the very start, then you will start with good. The first epoch will have the good results immediately. Okay, it's not that big problem, as you may assume, because even if you start with unnormalized data, after few epochs you will get there, if everything is correct. And if you're using the right model, even if you're having the big numbers and things like that, but often times you can go to these infinities or NaNs, and this may stuck at some point, and this may, it is good at the very start to have the normalized data. And I think I planned to create you know, like question and answer on how to normalize data, because I do that regularly and I may put it on StackOverflow but for many problems this is the most important to start well.

**Interviewer 1:** [00:39:40] Well. Okay, thank you. So you mentioned that even if you do not normalize, if you have a right model structure, it will converge after a few epochs. And I wanted to ask if you ever had problems related to wrong models structure. Like you and people on StackOverflow. Do you witness it quite often? And if yes, what kind of problems about model structure?

**Interviewee:** [00:40:03] Just, please, repeat the first part of the question.

**Interviewer 1:** [00:40:08] So you told us that even if you do not normalize at the beginning, if you have a right model structure your model will get a high accuracy after a few epochs anyway, so I wanted to ask whether you had problems related to model structure. Should I repeat again?

**Interviewee:** [00:40:34] No no. You do not have actually problems with model structure. The model that you are creating is your idea how the things should work.

**Interviewer 1:** [00:40:47] Okay.

**Interviewee:** [00:40:48] So if you create the model, then this means you intended to read this way. If you are picking the bugs that may arise when you're creating the model, this is another thing, but the model is like, it is doing what it is supposed to do so, I do not find...

**Interviewer 1:** [00:41:10] But isn't it possible that if you change a layer type or a layer dimension and etc., you get a better model that gives you a higher accuracy.

**Interviewee:** [00:41:25] Well, definitely it is. It is part. I don't know how to answer this question.

**Interviewer 1:** [00:41:36] No, if you just remember any specific cases when changing a type of layer, changing the dimensions of the layer helped you could just mention them. If not, we can just move forward.

**Interviewee:** [00:41:49] You know, I'm thinking about the reasoning. Why do you asking this question?

**Interviewer 1:** [00:41:55] Okay.

**Interviewee:** [00:41:56] At this time all the answers I do have would be, for instance, like a song from Lady Gaga. I just do not know how to answer.

**Interviewer 1:** [00:42:07] Okay, that's fine. So could you tell us about, so you mentioned that picking a wrong loss function is a problem and I wanted to ask whether you usually use a predefined loss function or did you ever write a custom written one?

**Interviewee:** [00:42:39] I do write some custom functions, but usually this is not necessary. Because, the existing amount of functions is just great for 90% of the problems. But it is really important to write at some point some custom functions to tackle some important problems that you may actually test, like random forest tests, if this would be better to do, it is another approach.

**Interviewer 1:** [00:43:16] Okay, and about hyperparameter tuning. Did you remember cases when there were problems because of the wrongly picked hyperparameters. If yes, which hyperparameters you think people are having more trouble with...

**Interviewee:** [00:43:34] Well, I will tell you this. We almost crossed the time I hoped for this interview.

**Interviewer 1:** [00:43:42] So yes, that's true.

**Interviewee:** [00:43:44] So if you have some other important

questions. Okay after that. Basically, the questions that you are asking me are not so about you know... When I prepared for this interview. I thought this would be something like providing the idea how to tackle this specific problem of yours, you know, okay this PreCrime problem. So I prepared a little more in that...

**Interviewer 1:** [00:44:19] No, but so far the things you told are very useful for our objectives...

**Interviewer 2:** [00:44:26] Current study.

**Interviewee:** [00:44:27] I can tell you, let me little advertise myself and I can tell you I do have the clear idea how you can get the latent factor. If a person do have the repeating crime, you know inside the repeating crime factor inside.

[Some ideas on how to predict whether the person will conduct a crime again]

**Interviewer 2:** [00:47:34] Thank you very much.

**Interviewee:** [00:47:35] I can't hear you. Okay? Okay. Bye and have a great day.

**Interviewer 2:** [00:47:40] You too, thank you. Good luck with your job.