**Interviewer 1:** [00:00:00] Okay. Great! So could you confirm it on record that you allowed us to tape this interview and share it's transcribed version.

**Interviewee:** [00:00:10] Okay. I confirm.

Okay, great!

**Interviewer 1:** [00:00:12] Thank you. So the first questions I have are about your background. Could you tell us what is your current position at your job? I would guess that you're a freelance developer, but...

**Interviewee:** [00:00:21] Okay, my current job, I am basically a student at [University: Removed for Anonymity]. Before that I was like working for 4 years in the industry, but I started to change to academia, and besides academia, I am working as a freelancer in Upwork. Yeah, that's a short story about me.

**Interviewer 1:** [00:00:46] Okay. Thank you. So could you tell us what is your overall work experience and your experience specifically in deep learning systems in terms of years?

**Interviewee:** [00:00:56] Okay, in terms of years, I have 4 years of experience. I started working in [Tech Company: Removed for Anonymity] in natural language processing. I was creating like natural virtual bots(?) for customer support and then I worked for [Team Inside the Tech Company: Removed for Anonymity] team for working on machine translation and speech recognition, which is [Product Name: Removed for Anonymity] translator. It is an app you can use to translate your language to another language. I was the responsible for developing the [Language: Removed for Anonymity] model. Because I was working in [Tech Company: Removed for Anonymity] in [Country: Removed for Anonymity], then I moved to academia. I did my master's for 2 years in the [University: Removed for Anonymity], which is a university in [Country: Removed for Anonymity]. I was applying like deep learning and neural networks on bioinformatics. Then I just joined a new PhD program, I was applying deep learning in cancer problems. So, overall, I think I have experience in, practical experience in neural networks and machine learning, in natural language processing, speech recognition and bioinformatics. Besides that I did like several project on Upwork. You can see it like more than 32 jobs, they have different domains. One of them is time series prediction for company called [Company Name: Removed for Anonymity]. And I did face recognition and speech identification for a biometric company called [Company Name: Removed for Anonymity] and there are other projects, smaller projects I did in Upwork as well.

**Interviewer 1:** [00:02:56] Okay, so just for statistics, could you tell us how many years of experience you have in deep learning?

**Interviewee:** [00:03:03] Deep learning, I think, from 3-4 years.

**Interviewer 1:** [00:03:07] 4 years, right? 3 to 4.

**Interviewee:** [00:03:10] Yeah.

**Interviewer 1:** [00:03:11] Okay. Thank you. So could you tell us what type of deep learning networks have you developed/implemented and by that I mean supervised/unsupervised/reinforcement learning?

**Interviewee:** [00:03:22] Okay. I did like convolutional networks and the recurrent neural networks with LSTM and GRU and RNM. And I did unsupervised deep learning for [inaudible] auto-encoders, like variational autoencoders I mean. So I didn't do any reinforcement learning, I didn't do, I think, firstly I didn't visit reinforcement learning. I think this is the only domain that I didn't visit and start lately.

**Interviewer 1:** [00:04:01] Okay, thank you. And could you tell us which languages, programming languages and frameworks have you been using so far in your experience.

**Interviewee:** [00:04:15] I use Python, but when I was in [Tech Company: Removed for Anonymity], there was some deep learning models built on other pipelines. So I was using, sometimes I was using like C#. And the framework was CNTK, but mainly like most of the four years I am using Python and Tensorflow, Keras and PyTorch.

**Interviewer 1:** [00:04:46] Okay. Thank you. So, for this interview we have one general question, which is what types of problems/challenges/bugs have you faced while developing the systems that use deep or machine learning. And we're interested in all kinds of problems and bugs. So we're interested in conceptual bugs, we're interested in just small programming bugs, any error messages that you got and etc. So if you could start from there and tell us whatever comes to your mind about this, that would be great.

**Interviewee:** [00:05:20] Okay. Let me think first. The early thing I remember, Tensorflow, the last monthes I was trying to use Tensorflow GS trying to implement and deploy deep learning models on the browser. Because Tensorflow GS is a new thing is in Tensorflow, I faced a lot of problems regarding the compiler compatibility issues. And sometimes I am using custom layers and they didn't implement it in Tensorflow GS. So, this to me, in this problem, I was like contacting Tensorflow a lot and asking them for fixes and something like that. I think this is the main problem I remember, but let me think more to find another problems. Some problems are related to data pipelines, like I am using custom Python functions and I need this custom function to be ready to be consumed in the training pipelines.

**Interviewer 1:** [00:06:45] Okay.

**Interviewee:** [00:06:46] But because of this custom Python functions, and the training pipeline are customized somehow to deal with GPUs. So I did like a bottleneck problems, related to transfering the data from the memory and the memory is in the GPU. So it gave me like a very slow behavior, which is like painful for deep learning training.

**Interviewer 1:** [00:07:18] Okay.

**Interviewee:** [00:07:19] It like reduce speed by 100x, so it needs to be customized and fix the problem to deal with custom Python functions as well.

**Interviewer 1:** [00:07:40] Okay. So do you plan to do this customization or what? Have you done it already?

**Interviewee:** [00:07:48] I did like some tricks and fixes for that, but I don't think that it is the perfect way for doing that. By the end of the day I tried to reduce the time as much as I can. But it is not a good situation. It's not the best solution. I think they have to talk more on providing some layers to be used for Python functions, because I think mainly Tensorflow or any other deep learning frameworks are built for computer vision. So, when I'm using it for different problems, like speech recognition, I'm implementing my function individually, like if you checked Tensorflow, you will find they implemented many functions related to image processing and doing that efficiently with the GPU. But in my case, if you try to work on speech recognition, you don't find any function or you don't find the most important function you can use in speech recognition. So you have to implement it yourself and when you implement it yourself, it won't be efficient and it won't be suitable for GPU pipelines, something like that.

**Interviewer 1:** [00:09:31] So, if I understand correctly you are telling that Tensorflow or any of the other frameworks, they have limited support for some kind of problems, right? That you want to address.

**Interviewee:** [00:09:41] Exactly, they are mainly focused on image processing and computer vision problems.

**Interviewer 1:** [00:09:47] Okay, I see. Thank you. Anything else you want to add?

**Interviewee:** [00:09:55] Let me think. Some problems are related to how to present the model, because our models we want to load it to like microcontrollers or browsers. So we need some of those to be more compressed and give the same accuracy. And, I don't think that they have, they don't have all the tricks for compressions. They just have quantization, which is losing a lot of accuracy for the model. So we need more support for model compression.

**Interviewer 1:** [00:10:41] Okay. So the things you were telling us so far, I feel like they are related mostly to the problems with the frameworks themselves. So is there any problems/bugs and etc. that appeared because of your wrong implementation in a way, or the wrong choices that you made for the specific model. Do you remember anything about that?

**Interviewee:** [00:11:04] I think wrong choices. And mainly it's because there are some layers are implemented in different ways, like you can implement it in a Lambda format and you can implement it using like contribution repo in Tensorflow. So you have to be careful and use the appropriate layer that can be used in different frameworks and different platforms. One day I remember that I was trying to convert one of deep learning models from Python to C++.

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

**Interviewee:** [00:12:01] It was a ResNet model and I think I faced like plugged in this issue and we didn't fix it. Because we don't know how we can convert this Python to C++. Like we have the same model and we need just a prediction, the code for prediction to be converted from Python to C++. And we know that we can do that in Tensorflow. But when w do that regarding the Resnet models, we found the problem. Yeah.

**Interviewer 1:** [00:12:43] Okay. So you were telling that you may select the wrong type of layers, if I got it correctly, but is it only because you cannot convert it between different programming languages or just because some specific type of layer is better for a given problem.

**Interviewee:** [00:13:03] No, so the problem appears when you try to convert from one platform to another platform. So there is no problem with implementing any layer in your current platform. But once you try to convert like from desktop to browsers, from desktop to another, from language to another language, when you try to do this conversion you face an issue.

**Interviewer 1:** [00:13:28] Okay. Yeah. So in your experience, does it happen very often that you have to do with these types of conversions?

**Interviewee:** [00:13:37] No, I think that it happens like 5% from my time. Some developers didn't face these issues, because they don't have to convert, do conversion, but because I was working on different projects and many projects need special things to do.

**Interviewer 1:** [00:14:01] I see. Okay. Do you remember any kind of more general problems that you have faced and you think that they are general across the developers who use this type of systems?

**Interviewee:** [00:14:14] I don't remember. I don't have anything in my mind now.

**Interviewer 1:** [00:14:25] Okay, that's fine. So I wanted to ask about the, so when you're training your models, do you usually use existing datasets or did you ever have to collect your own dataset for training?

**Interviewee:** [00:14:39] Most of the problems I worked on, I did my datasets, because like most of these problems are unique and most of these problems no one visited before. So, we have to collect our own datasets and try to make it suitable for the training.So, I think I'm facing like 60% of my time the dataset and the pre-processing and all that stuff.

**Interviewer 1:** [00:15:11] Okay, so could you give us more details? What kind of bugs/problems you face when you collect the data, when you pre-process it? Or any kind of problems related to training data?

**Interviewee:** [00:15:23] There's a lot.

**Interviewer 1:** [00:15:24] That is good!

**Interviewee:** [00:15:25] First of them, all these datasets do not have a standard format or the same format and everything. So, you have to do like a specific processing for each one of them, to make them unified to single standard formats. And then some problem for creating like, because some of these datasets are very huge and cannot fit in the memory, like we have, most of implementation of neural networks or machine learning models, you will find it is using memory a lot, like a CPU memory is loading all the data, did the shuffling, did everything and then trained. But in my problems that because of the data cannot fit in the memory, you have to convert it and into a good format, for like records that can be pipelines or fit or generate biased training. So, I think it is a problem, because most of them, I think the main problem is that there is no unified format for that, or a unified recipe for that. Even the documentations, because the documentation itself is not unique and is not helpful a lot. So you have to invent your own recipes and your own stuff.

**Interviewer 1:** [00:17:13] Okay. So you you told us that you have to do a lot of pre-processing and I guess the things that you mentioned, bringing it to the right format, is part of it. So do you remember any cases, when you did not do some kind of pre-processing, and you had problems in training, or your training failed because of this missing step? And if yes, could you give us details about that.

**Interviewee:** [00:17:36] Sometimes you would like, you had a problem in conversion, like I remember that some day I had a data and I gave it to the conversion and saved it into 32 format, like 32

**Interviewer 1:** [00:17:55] bits.

Okay.

**Interviewee:** [00:17:55] And then I was consuming it at 64 bits. So, the data like, in streaming, was reduced to half, but I didn't realzie that until I get, I realized that really because of the accuracy. Like, at first, you didn't realize that, because these bugs, there is no one tell you that you have a bug in this item. You have to realize it yourself. So yeah, if you have any problem in conversion, if you have any problem in your data, you realize it really late and sometimes you cannot realize that. Like you did the training and get accuracies and you are trying how to make these accuracies much better. And you don't know that there's a problem in the data, like until you realize that I have to revisit the data pipelines and see if this is something wrong and then you will realize that there is something wrong.

**Interviewer 1:** [00:19:01] Okay. Any other problems you remember about pre-processing?

**Interviewee:** [00:19:06] Eee, bugs related to pre-processing. Sometimes, I'm yes, I would like to do like shuffling, because if you have like if you have your data without shuffling, you will get biased models.

**Interviewer 1:** [00:19:27] Okay.

**Interviewee:** [00:19:28] So you want to do shuffling. And when you try to do shuffling, sometimes my data is very big to fit in the memory like terabytes of data.

**Interviewer 1:** [00:19:38] Okay.

**Interviewee:** [00:19:39] When you face like these terabytes of data, you will get like some recipes or some solutions that won't give you a full randomness and/or a full shuffling process. So, when you finish this problem, the only solution for you is using like a big data pipelines, such as such Hadoop or Spark. But sometimes you don't have these options and don't you don't have these items. So, you will create your own solutions, which is not very good with pre-processing. I hope that, if there are any implemented pipelines for pre-processing in Tensorflow itself, that can give you like a complete, full randomness for this big data.

**Interviewer 1:** [00:20:43] Okay, that's very interesting. Thank you. Do you remember any other kind of problems related to training data? Like if it's wrong in some way, like labels are wrong, or cases when you did not have enough training data? Like any kinds of problems about training data that you ever had.

**Interviewee:** [00:21:03] Labels are wrong? In the annotation part...

**Interviewer 1:** [00:21:15] Okay.

**Interviewee:** [00:21:17] Yeah, because...

**Interviewer 1:** [00:21:19] The... Sorry, go ahead.

**Interviewee:** [00:21:22] Go ahead.

**Interviewer 1:** [00:21:23] No. No you go ahead.

**Interviewee:** [00:21:26] I forgot what I wanted to say.

**Interviewer 1:** [00:21:32] Sorry. I wanted to ask about annotation part anyway. So do you do annotation only by yourself? Did you ever had cases when you had more than one annotater?

**Interviewee:** [00:21:43] No. I didn't do annotation myself.

**Interviewer 1:** [00:21:48] Okay.

**Interviewee:** [00:21:50] Sometimes I did that for small number of data, but I think it's like five percent of my time. Like if I would like to tune a model with a small number of data, like create, I remember that I tried to train YoLo(?) model to detect specific things in images. So, I just annotated like 100 images or less than that. So, I face the annotation in tuning the models only, using transfer learning, but because most of my problems were not implemented before in speech recognition, in biology. Because my PhD is trying to integrate deep learning/machine learning in biology. So, we don't have like transfer learning concepts in this in these domains. But in image recognition or in computer vision, when I try to do transfer learning, I'm doing tha annotation. And by doing the annotation, I'm trying to find any app or any open source app for doing that annotation very fast. I didn't face any problems in annotations. I think it's an easy job. But if you have a lot of data, it would be problematic.

**Interviewer 1:** [00:23:34] Okay, I see. Thank you. Okay, so moving from training data, I wanted to ask about the model structure. Did you ever face problems related to the wrong model structure and by that, I mean the number of layers, types of layers. Like do you remember cases when you changed the model in some way, and your performance or accuracy has increased, let's say.

**Interviewee:** [00:23:59] I think, this is specifically our job, like trying to change the number of layers, and train again, and see what are the accuracies. So, mainly I'm doing that stuff, and because of it is very time consuming, because of hyperparameters, lots of hyperparameters number, of layer number, of neurons, the type of activation functions. I'm doing that in random way, I know that there are a lot of people doing like bayesian search or grid search. But because, mainly when I was in industry, we need the models very fast. So you just like, you don't have this type of hyperparameter optimization, and I'm trying to increase or decrease the layers. Mainly we are trying to reuse model implemented in any research paper. And after that like doing some iterations, increasing and decreasing, and you will see if the, like I'm trying to increase one layer if the model has become much better I increase other one, until it's like saturated, leads to a saturation.

**Interviewer 1:** [00:25:29] Okay. Yeah, like what you said is totally true that there are different cases, you can add, you can remove and there is a lot of room, like there are a lot of different things you can do. But do you remember...

**Interviewee:** [00:25:43] But mainly doing that stuff just increases like 5% or maximum 5 %. I think that the main gain you will get is from your data. But the architecture itself, it can like increase up to 5-10%. Like I don't remember if I get 10% before by increasing the parameters, but I think that's the maximum I can get 5% of gain.

**Interviewer 1:** [00:26:10] So do you remember what you did to get the 5% improvement or not?

**Interviewee:** [00:26:14] It is just the parameters, like sometimes, it is appropriate to use less numbers of layers. Sometimes, it's appropriate to use like different kind of architecture, like you have to move too far from feed-forward neural networks to recurrent neural networks, because the problem, the type of problem is different.

**Interviewer 1:** [00:26:38] Yeah. Okay, thank you. And did you ever had problems about model structure in terms that there was a mismatch between the components of the model, let's say like layers, or, I don't know, different kind of networks put together.

**Interviewee:** [00:26:55] I faced these kind of problems when I started deep learning, but I think it's like because less of experience. The more experience you have, the better you can solve these kind of issues, like compatibility between layers and architecture and all of that.

**Interviewer 1:** [00:27:17] Okay. Thank you. So you mentioned hyperparameters and I wanted to ask if you ever had problems related to hyperparameter tuning, like changing the value of some hyperparameter and again gaining some improvement in performance or accuracy. And, in general, which kind of hyperparameter you think causes the most of the problems.

**Interviewee:** [00:27:42] I don't have any special answer for that. But every problem, it is a problem-based, so sometimes the number of layers is important, sometimes the activation function is much important, sometimes increasing the weights, which is number of neurons and dropout, like because to avoid overfitting it's important. So I think it's a problem-based, based on your problem, based on your domain, based on the data you have.

**Interviewer 1:** [00:28:28] So do you remember anything specific, some example, when you changed something?

**Interviewee:** [00:28:32] I remember that when I changed sigmoid activations into linear activations in the speech recognition, it gave me a gain. Yeah, and you don't realize that because no one telling you that in the research papers you read.

**Interviewer 1:** [00:29:01] Okay.

**Interviewee:** [00:29:01] And I think maybe we were more focused on layers and number of layers. I didn't focus on activation functions. So, I did that by accident, like someone told me you have your data, it is a nonlinear data and you need this embedding in a nonlinear format. So, don't do any sigmoid, don't do any squash(?) for the values. You have to get it using a linear activation. And when I did that, it gave a gain.

**Interviewer 1:** [00:29:40] Okay, great. Thank you. So the next question I have is about loss function. So do you usually use a predefined or a custom-written loss function and did you ever had any problems because of the use of wrong loss function, or because a loss function was implemented in a wrong way?

**Interviewee:** [00:29:58] No, I didn't have any problems with that. I'm using, I did customified loss functions, like triplet(?) loss and the triplet loss(?) with softmax and all of them are custom functions, you have implemented your way and I didn't face any issues with that. I think it would be better if they are implementing these custom functions, because it is well-known and many people know that stuff. So I think it's much better to be implemented in the frameworks, instead of my implementation. Because with my implementation, it might be not so efficient. So I think it's the only problem.

**Interviewer 1:** [00:30:47] Okay. Thank you. So and one more question about hardware. So where do you train your models and have you ever encountered problems that are related to hardware?

**Interviewee:** [00:31:00] No, mainly I'm training my models, I use Google Cloud and I use AWS machines, virtual machines. When I was in [Tech Company: Removed for Anonymity], I was using their own internal servers. And in Upwork, some people, some clients don't have their own platforms or hardware to train. So, this type of jobs I didn't take, I didn't take these jobs. I have like conditions: you have to provide these data.

**Interviewer 1:** [00:31:37] Okay.

**Interviewee:** [00:31:38] All right, you have to well know your data or your problem. Second, you have, you must have a server or must provide me like an access to any cloud to train your models. I had the client before, he has his own server in his house. Like we have like five machines and all these machines have good GPUs, like Tesla GPUs. So I was using this stuff.

**Interviewer 1:** [00:32:15] Okay, so you have used different things and you never had problems related to hardware?

**Interviewee:** [00:32:20] No.

**Interviewer 1:** [00:32:21] You're very lucky!

**Interviewer 2:** [00:32:23] Yeah.

**Interviewee:** [00:32:24] You see the problem isn't hardware, what I face is related to memory, like the normal memory. Like I have terabytes of data and the machine I have is like 32 gigabyte memory. And when I asked for more RAM for pre-eprocessing and all that, they replied that memory is very, very expensive to them.

**Interviewer 1:** [00:32:55] And how did you deal with this problem?

**Interviewee:** [00:32:57] Just split your jobs into several tasks, to run on different machines and we waited for days.

**Interviewer 1:** [00:33:05] Okay, patience. Okay, so do you have a question, Interviewer 2?

**Interviewer 2:** [00:33:14] A little one. Have you tried to train your models on several GPUs at one time? And did you face any problems related to concurrent things?

**Interviewee:** [00:33:25] I did I did that only in [Tech Company: Removed for Anonymity] and they have their own solution for using different GPUs, which I don't see. Like they have like their own cluster, you have to submit your job and ask for any number of GPUs, you have up to 8 GPUs. And when I use that, I didn't face any problem, after I left [Tech Company: Removed for Anonymity] and worked for different companies, I just use like one GPU for all the data and all the training.

**Interviewer 2:** [00:34:00] Thank you.

**Interviewer 1:** [00:34:01] Yeah, thank you. So, one of the last questions, do you remember, like while we were talking, did you remember any other kinds of problems and bugs that we have not discussed?

**Interviewee:** [00:34:14] I think there are more of that, but I don't remember, I didn't prepare for a while.

**Interviewer 1:** [00:34:28] No, you did not need to prepare for this interview.

**Interviewee:** [00:34:38] I can send you like, if I remember anything I can send you by Skype or an email.

**Interviewer 1:** [00:34:39] As you like.

Yes, that would be very useful. Thank you. So I think, yeah, we can wrap up here ...

**Interviewer 2:** [00:34:45] One little question. Do you practice using optimizers like Adam Optimizer or any others and did you ever had problems related to this?

**Interviewee:** [00:34:55] No. Sometimes, I didn't experience a lot for changing the optimizers in the training, from the SGD to Adam, but I don't focus in, I didn't focus on changing the optimizer before, like just using the suitable optimizer for a suitable architecture and that's it.

**Interviewer 2:** [00:35:25] Okay, thank you.

**Interviewer 1:** [00:35:28] Okay, so thank you for your time. Thanks a lot. It was very nice to talk to you.

**Interviewee:** [00:35:42] Thank you too.

**Interviewer 1:** [00:35:43] Okay, bye-bye, have a nice day.