1. What are the ways over-fitting can be mitigated in DNN Training? (10 Points)

Overfitting means that the model would not learn the data, but the model chooses to memorize the data.

We could use three ways to mitigate the over-fitting issues. First, we could use dropout layers, we could delete some feature randomly. In this way, the model would not memorize the data easily. Second, we could apply regularization. Regularization is a form of regression which would restrict the coefficient to zero. In this way, we could avoid the model to be flexible. Thirdly, we could do early stop. By reducing the capacity of each layer, the model could not memorize the data easily. The model could only study the important pattern of the data.

2. What is the effect of learning rate on DNN training? Explain what can happen if the learning rate

is too high or low. (5 Points)

The learning rate is a hyper-parameter used to control the speed that algorithm updates or learns the pattern to regulate the gradient.

If the learning rate is too high, the learning process would have drastic changes. In this way, the divergent behaviors would be happened.

If the learning rate is too low, the learning process would need to spend a lot of time on learning patterns. The whole process would be slow.

3. Why can’t multi-layer perceptron (dense layers) be used directly on images? Why is the

convolution layer more suitable for extracting features in computer vision? (15 Points)

The multi-layer perception is not suitable for the image. By using dense layers, we need to flatten the layers. When the matrix is flattened, three things would happen. 1. The number of features would too many. 2. The multi-layer perception would not have good built-in invariance with local distortion. The model would not recognize the image with the different directions. 3. After flattening, the different features would be independent. In this way, it is hard for model to learn the images.

For the first reason, after the convolution operation, the parameters would be filtered and have the common features. For the second reason, the pooling would help CNN to have a similar pattern. The model could find the object in the image with the different directions. For the third reason, the activation model would help model generate the feature map which make the connections among features.

4. Why was the AlexNet architecture split in half? Do we still need to split AlexNet if trained on

modern GPUs like NVIDIA A100? (5 Points)

The reason why AlexNet needed to split in half is that single GPU would not fit entire network. The AlexNet is trained on GTX 580 which is only 3 GB large. In this way, each GPU would have half of neurons. If we could train AlexNet by modern GPUs, we do not need to spilt in halt. The modern GPUs would have larger storage for AlexNet to train.

5.What are some of the limitations of the RNN architecture? How does the Transformer

architecture overcome these limitations? (10 Points)

1.RNN could only capture the long-term dependencies. It could process the sentences word-by-word.

2.RNN could only use learn embeddings. There are no weight matrices about self-attention in RNN.

3.RNNs would need recurrence to train. Therefore, it is hard for GPU to do the parallel computing.

The transformer architecture could overcome these limitations easily.

1. Transformer could put the input sentence as a whole input. In this way, transformer would not need to capture the long-term dependencies.

2. Transformer would introduce the self-attention to find the relations between the words.

3. Transformer would have positional embeddings without the necessary of recurrence. In this way, it is possible for GPU to compute in parallel ways.

6. Define Scalability of a parallel system and describe what is weak scaling and strong scaling? (10 points)

Scalability: A parallel system’s ability to describe increase of parallel computing with much resources.

Weak scalability: The problem size is fixed for each processor. We could add more processors to decrease running time. In this way, we could run the larger task in the same amount of time.

Strong scalability: The total problem size is fixed. We could add more processors to distribute the whole problem. In this way, we could run the same problem faster.

7. Assume 23% of the runtime of a program is not parallelizable. This program is run on 128 cores

of a state-of-the-art 128-core AMD Milan system. Under the assumption that the program runs at

the same speed on all those cores, and there are no additional overheads, what is the parallel

speedup? (10 points)

Speed ==0.00601times

Speedup in 128 cores==4.2371

The parallel speedup would be 4.24.

8. Given an algorithm where the fraction that the program is serial is 0.25, what is the efficiency on

56, 64, 128 and an infinite number of processors. You may ignore communication costs and the

problem size is fixed. (20 points)

Speed for 56===3.797

Efficiency for 56==3.797/56=0.68

Speed for 64===3.821

Efficiency for 64==3.821 /64=0.60

Speed for 128===3.908

Efficiency for 128==3.908 /128=0.31

If there is infinite number of processors, the efficiency would be approach to 0.

9.What are the main differences between a Shared Memory Model and Distributed Memory

Model? (10 points)

Shared Memory Model: The memory that all processors could access the same resources. The user could use the global address space for the memory.

Distributed Memory Model: The tasks would compute on its own memory. If the data is needed to transfer, they could communicate by sending or receiving data.

The differences are that the shared memory model would need access the same memory resource. The distributed memory model would not need to access the same resource. The distributed memory model needs to send or receive data. The shared memory model would not need to do this.

10. Why is CPU utilization an important metric for high-performance interconnects? (5 points)

CPU utilization means the amount of CPU time to send or receive data. When the machine is interconnecting, CPU need to process the commands from each machine to finish the actions. The importance of CPU utilization for high-performance interconnects is that CPU would take a big part of the time about interconnection. In this way, if the CPU utilization is high, the interconnection would be terrible for users.