Hi, everyone, today I am going to talk about a deep frequency principle, which provides an understanding of why deeper learning is faster.

This is our machine group at Shanghai Jiao Tong University. This DFP work is joint with Hanxu Zhou, an undergraduate student working with me.

Depth is very important in modern machine learning. By increasing the **depth,** we can improve the learning much better.

For example, let’s take a look in this example of the original resnet paper. The red curves are for network with 34 layers, while the cyan ones are for the 18 layers. The Thick curves are training errors and the **thin** ones are for the test errors.

As we can see, either the training or the test case, the error of the 34 layers decrease faster. And finally, the deeper network achieves a lower test error. In this case, the 34-layer network has a faster training and better generalization. In this talk, we would focus on why deep can increases the training.

Currently, there are very few works on studying the effect of the depth. A 2018 paper studies linear network and found the depth acts as a preconditioner which may accelerate convergence. The preconditioning can be seen as an acceleration procedure that combines momentum with adaptive learning rates. However, it is still very unclear how to understand real non-linear deep network.

It is well known that DNNs and the real data are very complicated. A common sense that MNIST data is the simplest data. If we study the DNN in learning a much simpler data, such as one-dimensional data, others may think we are not doing a real DNN problem. The follwoing comic is a pretty good example to show this situation.

A: I am looking for my quarter I dropped.

B: Did you drop it here?

A: No, I dropped it two blocks down the street.

B: Then why are you looking for it here?

A: Because the light is better here.

This may implicate people are doing something they can but not something matters.

I would like to change the last sentence by

A: **Because I need to get familiar with the road structure first.**

Similarly for deep learning, which is far too complicated, even MNIST is a too high-dimensional data set.

Therefore, we reproduce the deep phenomenon by learning a 1d function. Cos3x+cos5x. We define the speed as the training steps to achieve a fixed error. As shown here, as we increasing the number of layers, the required step decreases significantly.

Here, we provide a learning component view to think about this problem. Consider a deep cnn, if we regard the last fully-connected (FC) layer as a learning component, and study the effective target function of this learning component. The input to this learning component is the output of the previous layer.

Now consider two structures and regard the last fully-connected (FC) layer as a learning component and examine their effective target functions. If the deeper hidden layer‘s target function is one easier to be learned, then, we can understand why deeper is faster.

Therefore, we need to consider what kind of function DNN can learn easier, which may explain why we need less training steps when increasing layers. For a fixed network, when fitting different target functions, our previous works show that DNN learns low frequency function faster. As shown in here, lower frequency function requires less training steps. We guess the effect of depth maybe relate to frequency. Let us recall the frequency principle first.

To clearly show the training behavior in the Fourier domain, it is very necessary to learn a 1d function, which is easy to be visualized in the Fourier domain, a light place. The plot is the amplitude vs. Frequency. The red is the DFT of the target function and the blue is the DFT of the DNN output. This is a movie, each frame is after several training steps. As we can see, the DNN learns the first frequency peak first, followed by higher and higher frequencies. This is called frequency principle: DNN learn low frequency first.

This F-Principle is first observed in 2018. We have two papers to demonstrate this phenomenon. There is an interesting story about the connection between the third paper and ours, which we claim both independently found this phenomenon. A series of works have theoretically shown the validity of the F-Principle. There are several algorithms are developed to overcome the weakness of learning high frequency in deep learning. In addition, the F-Principle explains many phenomena in practical applications. In this talk, we focus on using F-Principle to understand the effect of depth.

By the way, the F-Principle is a central work of our series work on studying deep learning from implicit bias view point.

Now, let’s get back to the study of depth. Increasing depth may lead to an easier function. Therefore, we have the following conjecture: The effective target function for a deeper hidden layer towards a lower frequency function during training. There are two things we need to figure out. One is what is the effective target function, the other is we need a quantity to characterize the frequency distribution of high-dimensional functions.

Now consider the effective target function of the green layer when doing back propagation. Its input is the output of the previous layer and the error signal propagates from the difference between the DNN output and the training data.

Therefore, when considering an interested hidden layer during analysis stage, we actually decompose the network into two parts, one is the pre-condition part, the other is the learning component. The effective target function for the learning component or the green layer is the mapping from the output of l-1 layer to the true labels. This is the one we need study in the following for each layer.

Next, we need to find a quantity to characterize the frequency distribution of a high-dimensional function. We decompose the function into a low-frequency part and a high-frequency part in radial direction. This can be done by the product with an indicator function in Fourier space. Then, we can define the low frequency ratio as the ratio of the energy or L2 norm of the low-frequency part over the original function. However, this is difficult since the computation in Fourier space suffers from the curse of dimensionality. Therefore, we use a Gaussian function to approximate the indicator function. Actually, this is a common gaussian filter, through which, we can obtain the low-frequency part. Then, we can define an approximate LFR.

I would like to talk a bit more about why we use Gaussian filter. For a Gaussian function with standard deviation delta, its Fourier transform is still a Gaussian function with the standard deviation 1/delta, which can be regarded as the frequency width kept in the low frequency part.

LFR is similar as the cumulative density function in probability. Then, we can define a Ratio distribution function to characterize the energy of each frequency, similar to the probability density function. Now, we apply these two quantities to simple function sinkx. For LFR, as we can see, for low-frequency function, the width does not have to be a large value, the LFR can achieve 1, while for high frequency function, the width has to be large when LFR approaching 1. For the RDF, it is much more clear that the peak locates at a larger value when the frequency is larger. Now, we can forget all technical details, and only remember peak more right, frequency higher. Since this is done in radial direction, these quantities apply for high-dimensional functions.

Now we examine the deep frequency principle in variants of resnet18. Take an example, we fixed the last FC layers and examine its target function’s frequency distribution. We expect to see the lower structure, which has more layer, would have a low frequency peak during training.

We use four variant network structures of resnet18, from the first one to the last one, we eliminate one by one sub-convolution part.

First, we reproduce the observed phenomenon. The training accuracy of the deeper network increases faster and the validation accuracy is higher at the end of training.

Next, we examine the frequency distribution of the last FC layer in each structure during the training. At the beginning before training, we can see the deepest one has the highest frequency peak. This is definitely not good, since neural network prefer to learn low-frequency fast. However, during the training, the peak of the deepest structure goes towards lowest frequency. Relatively, the shallow ones peak at higher frequency positions. At epoch 3, the deepest structure peaks at the lowest frequency position. At epoch 15, we can see there is monotonical order that deeper structure possesses lower-frequency peak. And this monotonical order is preserved until the end of the training.

Based on this experiment, we can find a deep frequency principle: The effective target function for a deeper hidden layer has a bias towards a function with more low frequency during the training. If we want to do a further theoretical analysis, this would be a bit difficult since the comparison crosses different network structures. A natural question is that how about the frequency distribution of different hidden layers in a network?

Therefore, we consider to use a FC network to learn MNIST. There is only one network, but we consider the RDF of different hidden layers. We should expect to observe that for deeper layers, their effective target functions should have more low-frequency during the training.

Here are the results. Similarly, at the beginning, the deepest layer, the red one, peak at the highest frequency position. Actually, this is not a stable phenomenon if we run more experiments. However, as the training goes on, we can see that the deeper layer goes towards low-frequency position. At epoch 100, we can see the trend. At epoch 200, this trend is much more clear. At epoch 400, the two most deepest layer peak at lowest-frequency positions and keep their situation throughout the training. We can see they still have a monotonical order, such as at epoch 600 and 800. Therefore, the neural network transforms the target function to be an easier and easier function that the neural network can learn.

Our summary is an empirical observation: The effective target function for a deeper hidden layer has a bias towards a function with more low frequency during the training. Further theoretical study: start from one deep network.

This idea is a bit similar as the kernel methods, such as SVM or random feature model. However, the NN combines multiple kernel and also learn the kernel to make the target much easier.

Previous studies have shown some connection between the generalization and the low-frequency bias, we guess we can further study the generalization effect of depth.

The other important thing is how much deep should we design in a practical task. As we can see that, deeper network is better to learn difficult task. But the effect of depth can saturate. Further study may give some guidance for consider how many layers are required.

Finally, more information can be found in my homepage at SJTU.