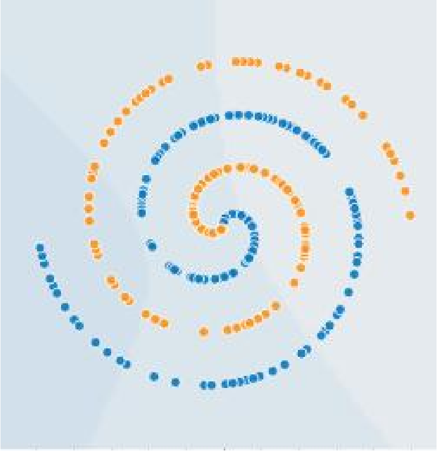
From discriminant model to understand neural network

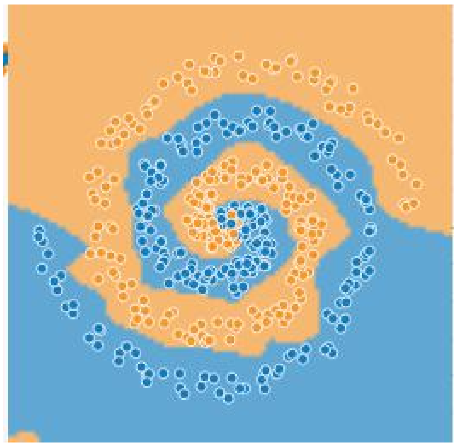
Neural network is one kind of discriminant model meaning we draw boundaries through the whole input dimensional space and put a label to each point. Although the training set become larger and larger and the final boundary can become extremely complex,due to the use of multi layers and nonlinear activation function,we can still approximate that boundary well by increasing the trainable parameters.Any two-layer(or greater) nonlinear neural network can compute any mathematical function which has been proven by 'Universal Approximation Theorem'. This character leads neural network to outperform any other traditional machine learning models and keep shining with the growing amount of available data.But, as discriminant model, neural network directly map input data to labels will always suffer some inborn deficiencies.Concretely, the requirment of large amount of training set, generalize badly to new data from different distribution and easily cheated by adversarial samples. I will first discuss the discriminant characters of neural network and then elaborate understandings of the above mentioned deficiencies.

The discriminant character is elaborated in figures below:

E.g we want to train a neural network to approximate the function of y=f(x) and use the trained function to predict with new samples.There are two classes colored by blue and orange. Because they are 2-d dots, the input are the location of each dot (x1,x2).



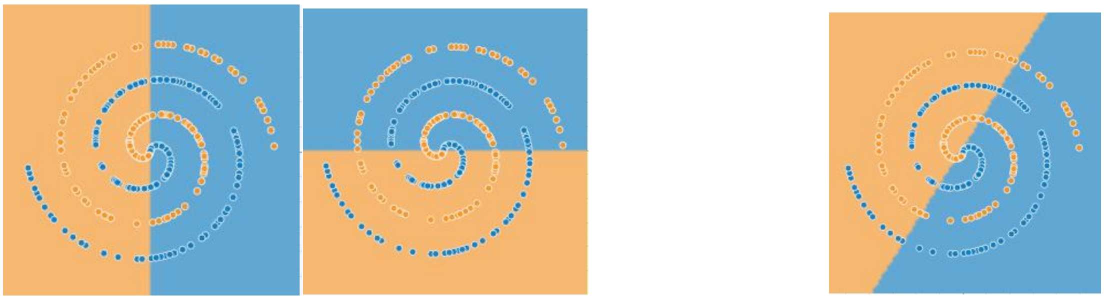
If we add some noise to it and use neural nets (here I use 3 layers with 8 units each layer and tanh activation function), we can get one decision boundary shown below:



Units in the same layer are weighted summed(we can treat the bias term as one additional input with weight -1) then followed by nonlinear activation. The weights between units of this layer and units in the following layer represent ‘how much ’ we use the output results from previous units.We can imageine each output from previous unit is a ‘wall’,the absolute value of the weight is the height of the wall, the sign of the weight is the direction of the wall(up or down),we add all the different walls followed by tanh nonlinear mapping to build our final decision wall. So the boundary generated by unit in the later layer is the weightd sum of boundaries from previous layer.

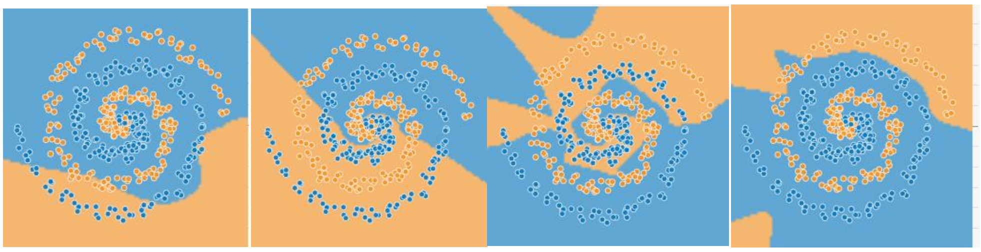
Elaborations are shown below(Tanh activation function):

Input boundaries Output boundary

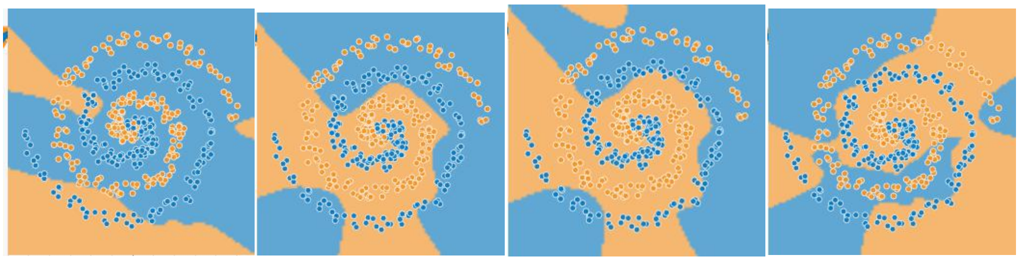


Weights: 0.41 -0.26

We now back to the boundary generated by the three layer neural network. The final well-shaped boundary generated by the output unit is weighted sum of the third layers output boundaries. I list all the 8 output boundaries below and their weights to verify.



-0.43 -2.3 -1.3 2.6

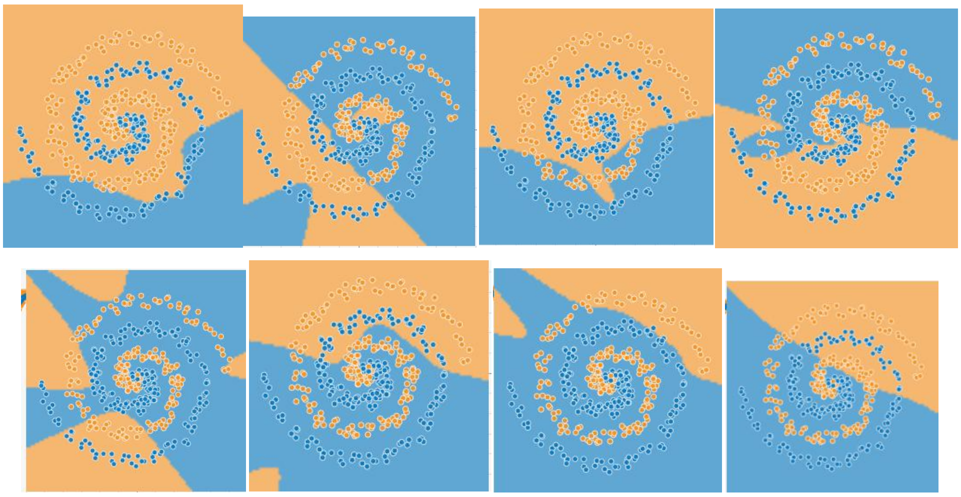


-0.29 2.6 1.2 -3.1

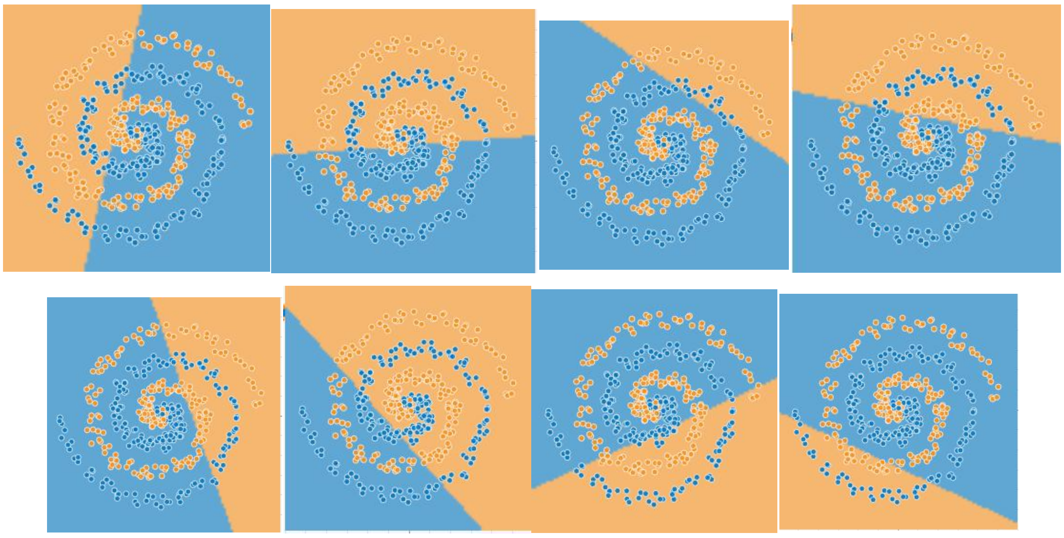
Of course the weird boundaries of each unit is generated by the weighted sum of second layer units outputs.

And from that, we can then discuss the explainations of why neural network works and their deficiencies. Each unit output boundary in intermediate layers looks ‘nonsense’ to the input data and can’t seperate the data while every of them is somehow ‘similar’ to the finally decision boundary. I am not sure whether such phenomena have connections with the visiualization of CNN units.I will not discuss that part here. During the training process, we can only determine the final decision boundary given the labels of training data. The weights of all layers are learnt by gradient descent to minimize the differences of outputs and labels. In discriminant models, we approximate the decision boundaries(will exist many in one continuous narrow area) that can perfectly seperate all the data. But the trainable parameters are huge and the input dimensions are always much higher than 2, it's impossible to visualize the training process and the outputs perfectly. The training process may fall into different minimum with different paths due to the initialization. But,from the view of discriminant models, the whole process is still draw boundaries and weighted sum them to approximate the final boundary.

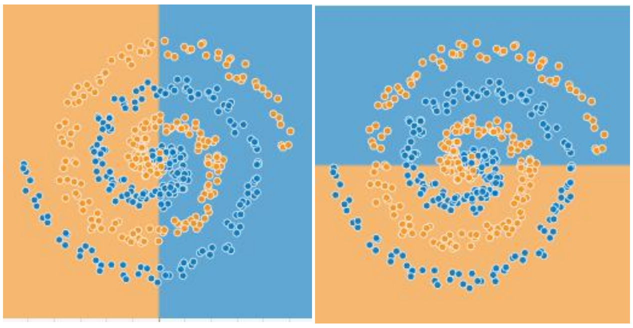
I also list all the outputs of the other layers to help elaborate this idea.



Layer 2



Layer 1

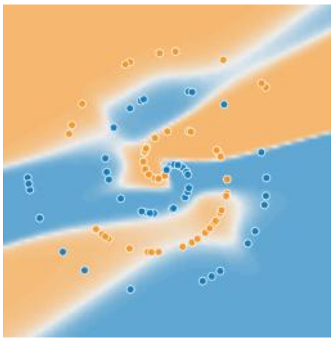


Input layer

Next I will elaborate my explainations of the deficiencies existed in neural networks.

1. Requirment of large training samples

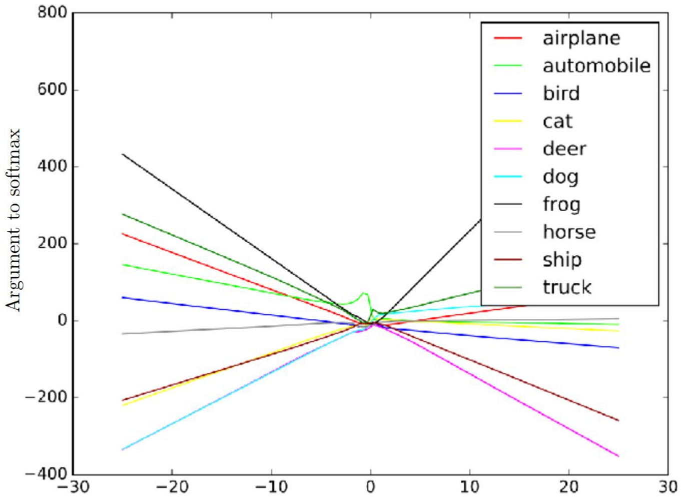
The different classes of input data inherently have some connections and form different 'clusters' in the whole input dimensional space.But due to the complex structures of each class(we can take MNIST for example,it's very hard to describe what structure form the different dights) the clusters maybe not like 'sphere' surrounding to exact one point in all dimensions,they may only 'close' in some dimensions and far away(or have no relations) in other dimensions(image pixels are very rudundant, using dimension reduction methods like T-SNE/PCA we can already seperate digits well only rely on several dimensions). In neural networks, we can approximate any mathematical functions.But we can't choose how we approximate the boundary and the boundary is described by the data.So the whole input dimensional space is very redundant(for image data), if the amount of data can't describe the structures and relations of different classes well in the high dimensions, the boundary will be teared apart to fragments and causing the space labeled messily. Elaboration is shown below.



But we can still use neural network to approximate that 'broken' boundary well which is different from overfitting or underfitting but due to the lack of training samples. And when we use test samples even from the same distribution of input data, if it happened lie in mislabeled regions, the output label will be totally different.Because all the space is labeled by the decision boundary and we can't decide the process to approximate the boundary. The generality will also be ugly if training samples are too few and neural networks will be inferior compared to other traditional machine learning models if the amount of samples can't match the high dimensions and can't describe the relations in the 'needed' dimensions.

2. Generalization

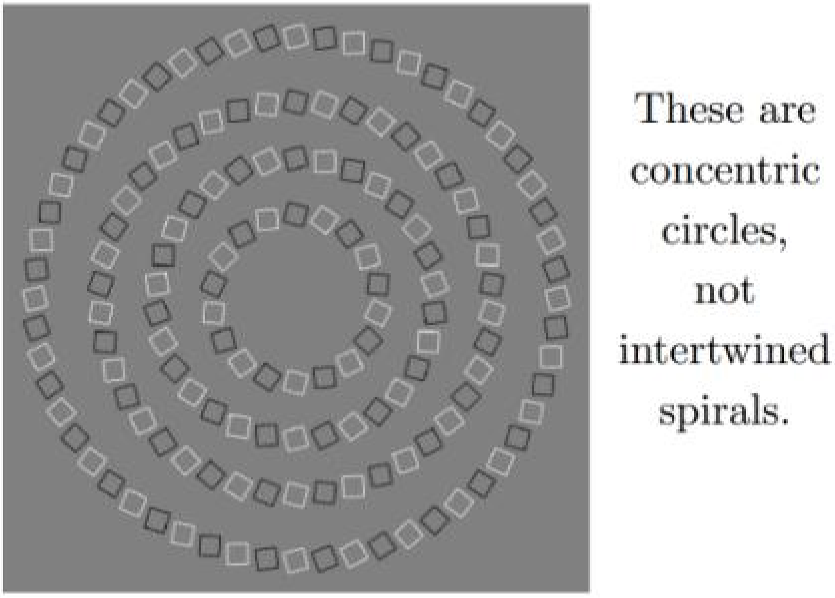
The final decision boundary and the label of the space can only determined by the training data in neural network.The only job for the model is to successfully draw boundary to classify the different classes.But the way to draw the boundary may tricky and totally different from human.For examples, in CIFAR-10, we classify different classes by the shape of the object shown in the image.But the input data to neural network are just pixels and neural network has no preformed concepts of each object and how to classify them like us.But because the size of CIFAR-10 is 32*32,the total input dimensions are very high and redundant.Even can’t be perfect,neural network can still find some boundaries with very small loss using the extra unimportant dimensions (the background or the some specific color pixels shown in fixed positions in one class). In short, neural network may following the wrong cues to classify in some datasets. The figure below shows the likelihood of each class by disturbing the trained network following one specific direction(1024d-vector) in the whole input space.



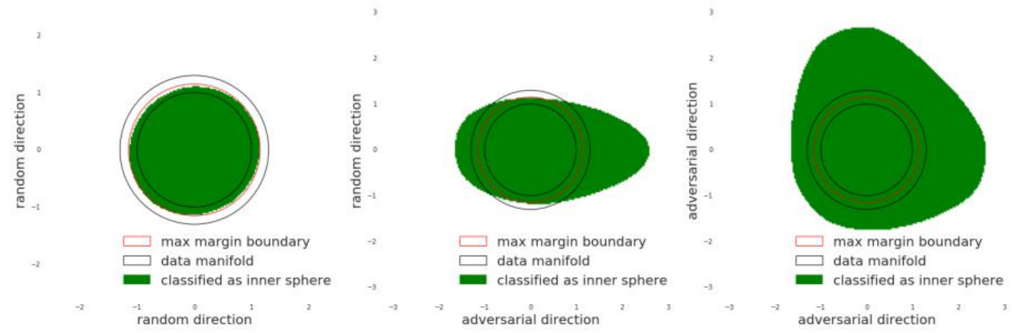
As we see, the correct label(automobile) is only within a small area of the whole space(no disturb).if we follow one direction of the whole space, the output label becomes incorrect.More data from different distributions will alleviate this problem, but it’s extremly difficult to make sure the exact amount of per class to describle them perfectly using pixels.

3. Adversarial samples

Adversarial samples exist in every kind of discriminant models and our human brains also suffer from it. Onel sample is shown below.



In discriminant models, we can accredit it due to the mistakenly label the space by carelessly drawing the decision boundaries.Like the elaboration image in Sec.2, the whole space is randonly labeled if we can't using enough proper amount of data to cover the whole space.Because the only target we try to approximate is the final decision boundary. This paper(adversarial sphere): https://arxiv.org/pdf/1801.02774.pdf shows that even a perfect approximated boundary with tons of data(500 dimensions with 20 milion data), we still hard to make sure every dimension is correct just relying on the cost function.



So, these adversarial dimensions cause the vulnerability of the discriminant model. Especially in spaces where no data coverd, the label is less likely to be correct. Still, discriminant models will always suffer adversarial samples because brutally draw boundaries through the whole space and label them.

In conclusion, discriminant models are easier to train and employ, but the inborn deficiencies can't be overcomed.Maybe by changing the cost function to control the training process and more training data will alleviate the problems, but we can't eliminate them.The problems are caused by discriminant models themselves.

P.S During training, I find one interesting phenomenon. In above 2-D dots, if we not only use (x1,x2) but with some nonlinear function directly to the input (sin(x) or X^2) together with input (x1,x2) , the training process will be much faster and easier.