RESNET Structure Improvement

Rudra Prakash Sarkar

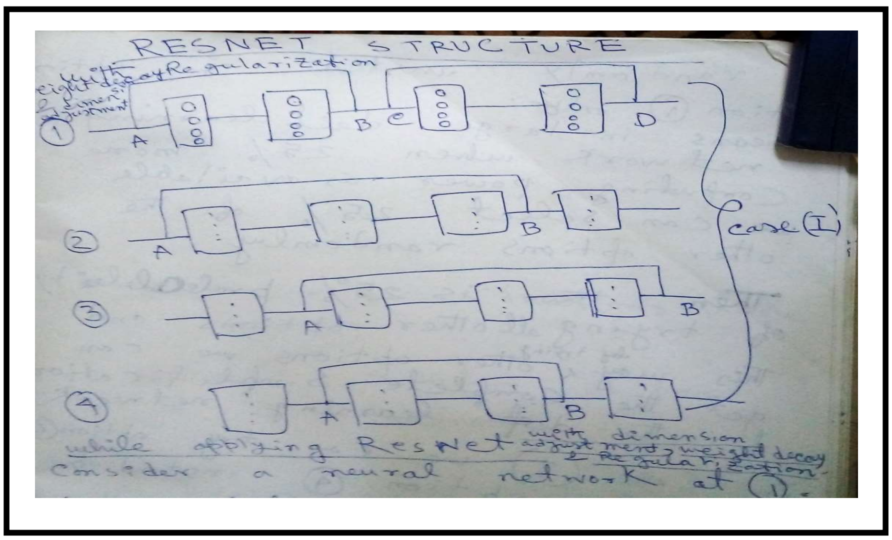
[B.tech(Electronics and telcomm enggr) from IEM,Saltlake affiliated to

university of kalyani, West Bengal]

Abstract : Machine learning and deep learning has tremendous potential that can transform our healthcare system,research in astrophysics,simplification of computing in neural network etc.Here I have modified traditional algorithms in machine learning and deep learning to apply in above mention areas to serve humanity.My approaches are simple and if implemented by code and training the model using dataset they will certainly make great products to help in healthcare,astrophysics and hardware utilization areas.

IDEA - I

Consider weight decay regularization and dimension adjustment While applying RESNET



Consider a neural network at

(1) ,layers between AB or CD can vanish.

In option (2) layers between AB can vanish.

In option (3) layers between AB can vanish after RESNET.

In option (4) layers between AB can vanish after applying RESNET.

So there are 4 ways.

Suppose our deep learning network is connected to cloud. If 25% more computing power is available at the same cost due to drop in demand in the cloud resources, we may select option (2),(3),(4) randomly instead of option (1) only. Means, in large deep learning network when 25% more computing power is available we can select 25% of the other options randomly.

Then there is 25% probability of trying all other options and in this way by trying other options we can get the complete simplification of the deep learning network.

In case (I) in the picture we can try option (2), option (3),option (4) instead of trying option (1) only.

We are selecting other options randomly, so there is a chance by trying only 25% more options.

Of all the options we may get a complete simplification(in the best situation or case.

When 25% more computing power is available at the same case we can do that.

Note: If we are trying option

(2)(for example),and layers between AB vanishes ,we will not try that option (2) again.

Say, total number of situations like (1),(2),(3),(4)….. is 6 and probable solution arises in 3 situations.

Suppose we try 2 more cases, in best case probability of complete utilization to get solution is

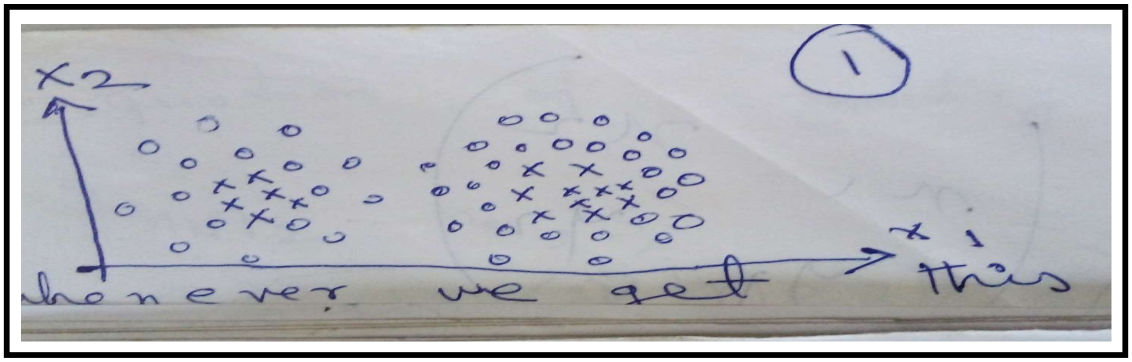
$(\frac{3}{6}).(\frac{2}{5})$.100%=20%.In worst case 20% probability of no solution.

If we try more than worst case possibility , say 25% (which is more than 20%=worst case probability) we get at least 5% probability of more solution at the same cost.

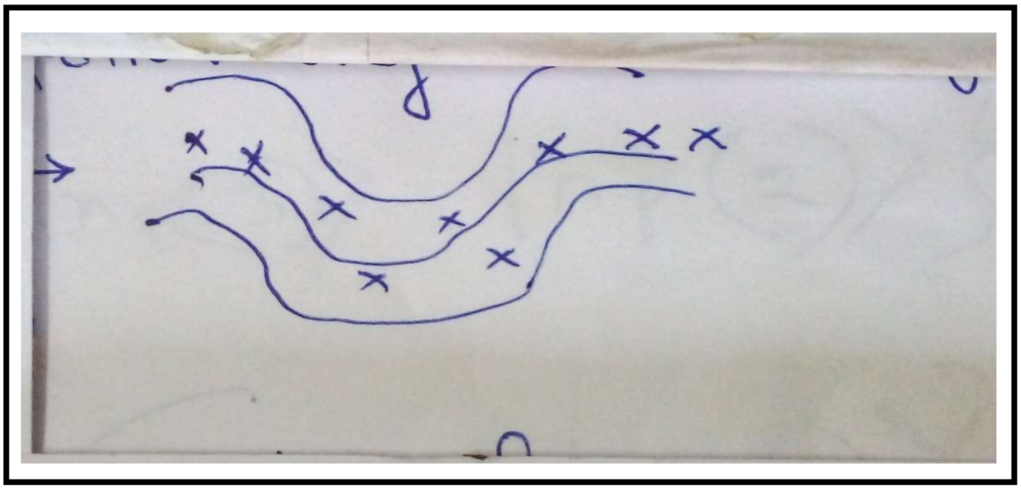
IDEA - II

Finding out pattern of the universes by Kernel SVM

Whenever we get this kind of peculiar(difficult to process) distribution of data (see fig(i)) we apply kernel SVM. We first convert the 2D distribution of data in three dimensional format then apply 3D hyper plane to net one type of data points then project it onto 2D plane. We get following kind of distribution (see fig(ii)) in 11 dimensional space. 4D(x,y,z,t) or 4 dimensional universes are floating in 11 dimensional space. This is a fact which is peculiar and difficult to fathom



Fig(i)



Fig(ii)

If we project 11 Dimensional space in 12 dimensional space (like we projected 2 dimensional data points in 3 dimensional space in kernel SVM) and net the data (4 dimensional universe) by 11 dimensional hyperplane in 12 dimensional space then project them into 11 dimensional space we will catch a glimpse of the distribution of 4 dimensional universe in a pattern(pattern in 11 dimensional space).In 11 dimensional string theory we have already got more than quadrillion 4 dimensional universes as solution. Treat them as data points in 11 dimensional space and apply kernel SVM like I have just described so that we can catch a glimpse of pattern of 4 dimensional universes in 11 dimensional space. So that we can predict the characteristics of more universes whose characters are still unsolved by string theory. The computing power will grow exponentially so we can apply kernel SVM in finding the distribution of 4 dimensional universes in 11 dimensional space.

References :

1) Deep Learning By Ian Goodfellow,YoshuaBengio And Aaron Courville
Published In 2016 ,November By MIT Press, ISBN- 978-0262035613

2) Deep learning with TensorFlow 2 and Keras by Antonio Gulli,AmitaKapoor,Sujit Pal
ISBN-13 978-1838823412, Packt Publishing, December 2019.