ALMA MATER STUDIORUM – UNIVERSITA' DI BOLOGNA

DEPARTMENT OF STATISTICAL SCIENCES

Master Degree in Statistical Science

Deep learning algorithms for the analysis of x-rays and gamma-rays

Presented by :

Zhe Luo

819973

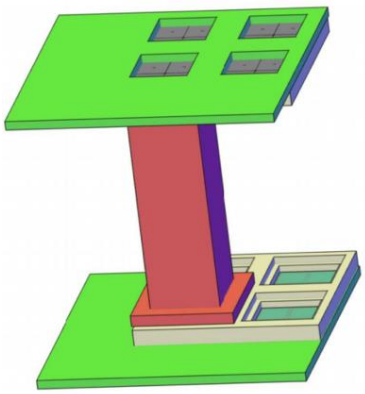
Supervisor:

Prof Stefano Lodi

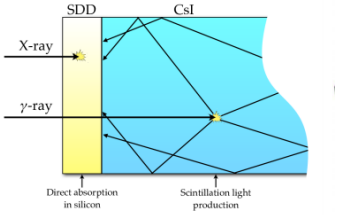
II.c SESSION

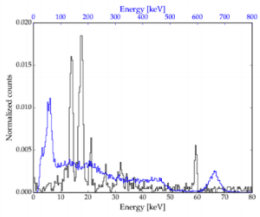
ACADEMIC YEAR 2019/2020

The project i paticipated is part of The National Institute for Astrophysics's program on Gamma Ray Burst phenomenon. In order to better study the gamma-ray burst, the X and Gamma ray spectrometer has been developed:



They improved the device in oder to absorb the X ray in the top side Silicon detector and can absorb gamma-rays in the crystal and SDD.the signals received were then registered according to the rise time. As shown below:

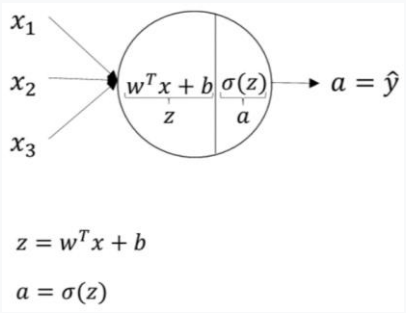




The black curve represents the spectrum of X-events. The blue curve represents the spectrum of gamma -event. those time series data had been stored by the National Institue for Astrophysics.

Our work was to use this data set to train an deep learning algorithm model ,which would be able to effectively identify X-ray and gamma-ray signals and simplified the process of spectral classification

I will briefly introduce the theoretical of deep learning .The Deep learning actually refers to the artifical neural network . it mimics the structure of biological neural network cells. Like this



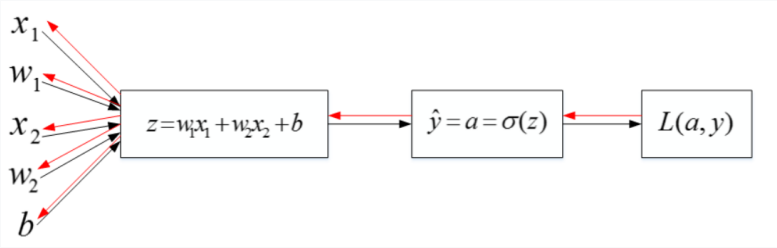
the neuron will perform a linear transformation on the input information.this W is called weight. The weight value is the memory of the network. each node has an activation function.It is a screening mechanism that controls the passage of information. Eventually we will get an estimate y hat

$$L(\hat{y}, y) = -(y log\, \hat{y} + (1 - y) log\, (1 - \hat{y}))$$

We usually use Cross entropy function as loss function. to show the difference between the estimated value and the true value.

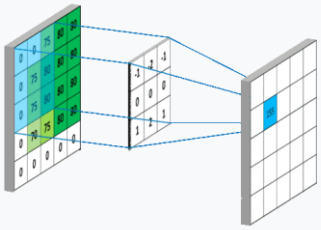
Due to the Cross entropy function is a concave function, the optimal point is in its bottom . We can use the gradient descent algorithm to keep the parameters close to the optimal state through lots of iteration.

The process of solving the parameters is called the back-propagation process. like is



Starting from the LOSS function and layer by layer, performing partial derivative operations on the parameters， This process has a lot of calculations. We can only do it through the host

Neural networks has many different types. Two of the most famous types are convolutional neural networks and recurrent neural networks.the first are usually used in the fields of computer vision. The most famous example is face recognition because it has the structure like this:

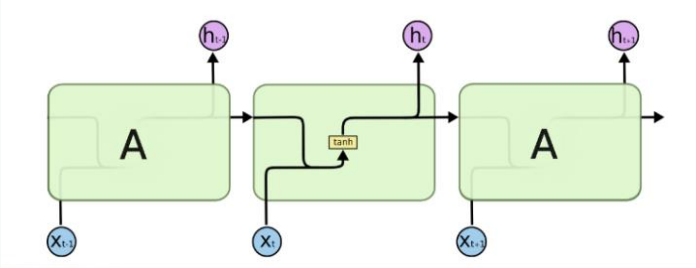


http://blog.csdn.net/weixin_36604953

It uses a convolution kernel as a medium for extracting information. If the input information is a picture, it will traverse all the picture. The traversed area will be multiplied with the parameters of the kernel and summed to form an output.

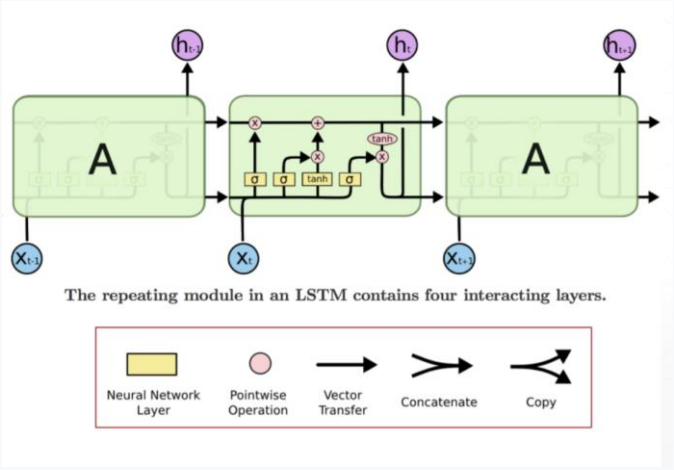
The structure of the recurrent neural network is slightly different.

1：20



Where X represents input. The most notable feature is that it contains a hidden layer A, which keeps the previous information and provides assistance for the subsequent analysis.

In the recurrent neural networks , the long short term momery model is the most widely used



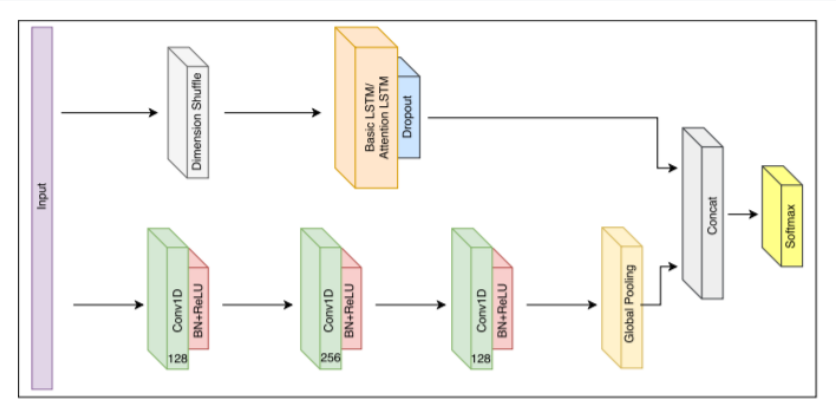
The repeating module in an LSTM contains four interacting layers.

It contains two hidden layers to save and transfer the information, and uses three sigmoid activation functions as the gate structure to control the information passing ratio. Because of this structure, the LSTM model is very useful for solving the gradient vanishing problem.

Next I will introduce our research process

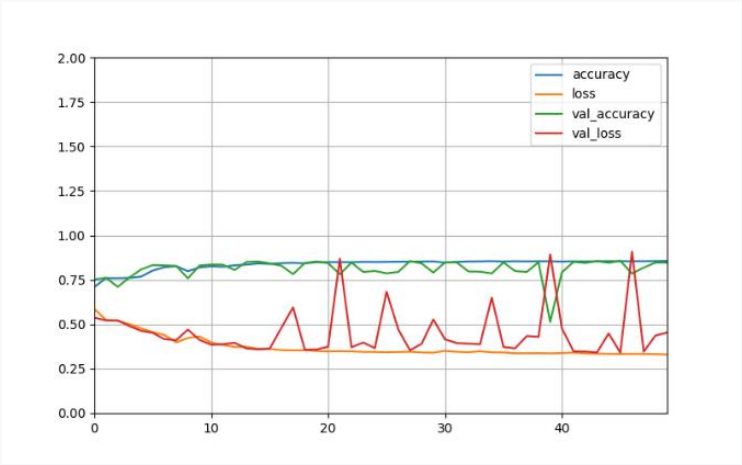
I mainly used tensorflow 2 and keras modules of Python language to build models. They are very convenience and ease – using while providing the GPU acceleration.

In terms of model selection, We first borrowed a model named LSTM Fully Convolutional Networks model. like this:



This is a mixed model,one channel is LSTM model, and another channel is three layers of normal convolutional neural network, We restored this model and adjusted the parameters to the limit that our host can carry, bring in the data set and did 50 rounds of traversal.

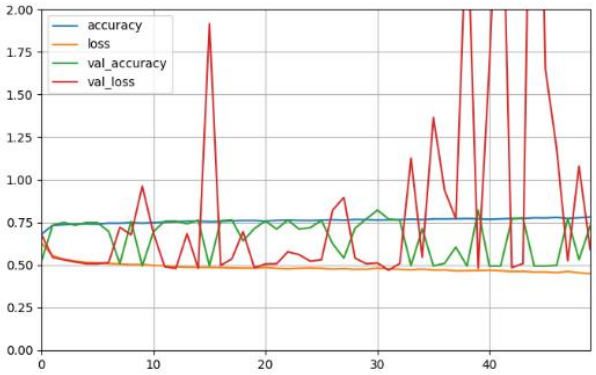
By the way,we would do the same in subsequent experiments in order to compare different models in the same experimental environment. After the calculation, the results is this:



Val-loss = 0.4483 Val-accuracy = 0.8437 Test _loss = 0.4507 Test_accuracy = 0.848

We used the Cross validation algorithm to devide the data set. The traing process is shown in the graph. The test set only be used for the evaluation of the error .We can see that the test accuracy is 84.8 percent. But we can also find that the validation loss

function has been turbulent and it wasn't convergent in the end , so we re-calculated it. 1:20

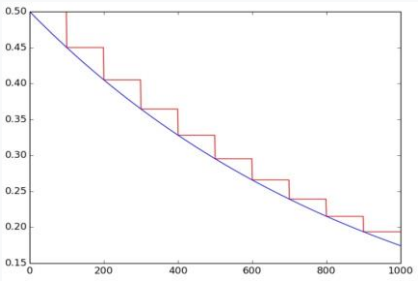


Val-loss = 0.5987 **Val-accuracy** =0.7240 **Test _loss** = 0.5878 **Test_accuracy** = 0.7195

As we see, we got a very different results, It could prove that the model was unstable, so we couldn't accept the previous result.

There are many reasons that cause the model to be unstable, but the most important factor is learning rate. Previously, our learning rate was fixed at 0.0001, and it was no longer suitable for any change. So we decided to use dynamic learning rate to deal with this problem.

We successfully implemented a dynamic adjustment of the learning rate , it can slowly decay with each iteration




```
decayed_learning_rate = learning_rate *
                        decay_rate ^ (global_step / decay_steps)
```

We upgraded the model and recalculated it

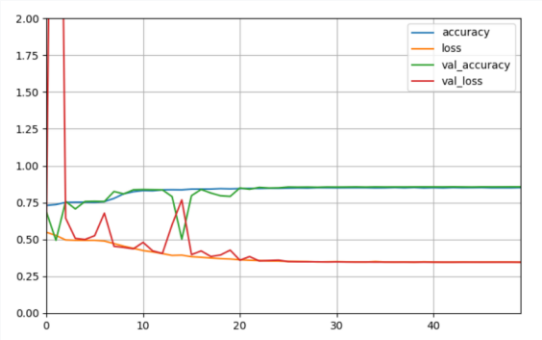


Figure : LFCN model

Val-loss = 0.3124 **Val-accuracy =0.8570** **Test _loss = 0.3120** **Test_accuracy = 0.860**

Time =22640

We can find that the dynamic learning rate has played a huge role, it not only made the model converged, but also gained 86% accuracy. It was a successful experimental result,But we wanted to ask :Which of these two channels is more important?So we make each channel to a complete independent model.

We performed these 2 new models. The results are as follows:

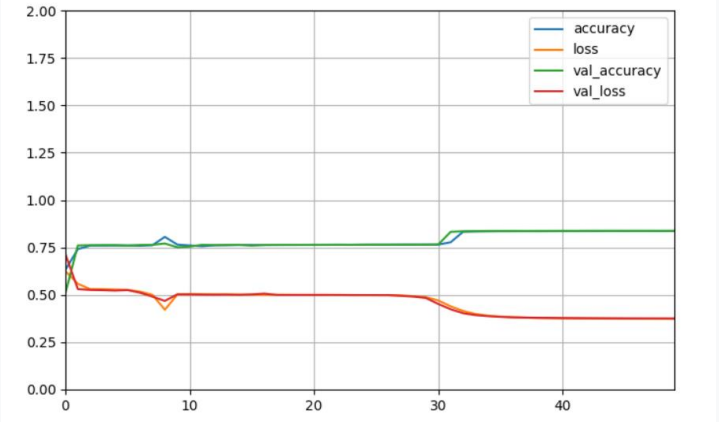


Figure : Long short term memory model

Val-loss = 0.3732 Val-accuracy =0.8365 Test _loss = 0.3663 Test_accuracy = 0.8412

Time = 36224

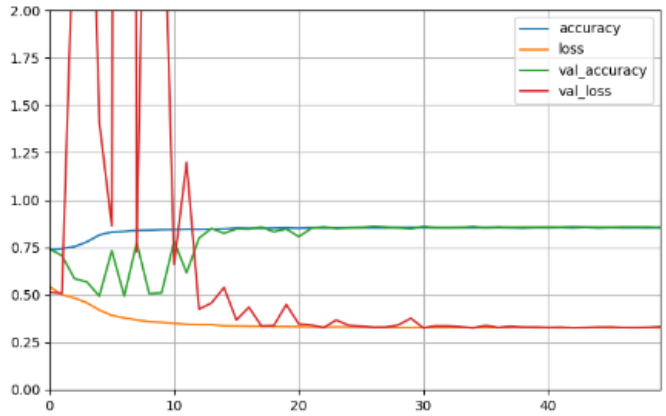


Figure : normal convolutional neural networks

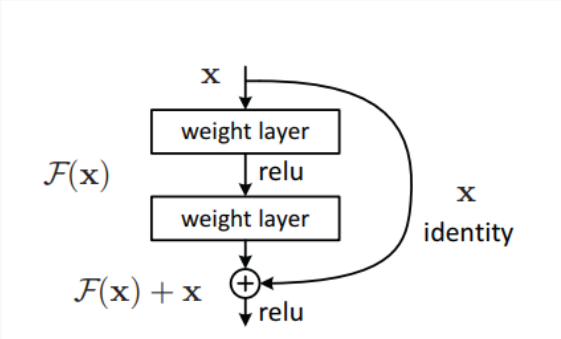
Val_ loss :0.3292 Val_accuracy:0.8561 Test_loss: 0.3251 Test_accuracy: 0.8585

Time = 8445

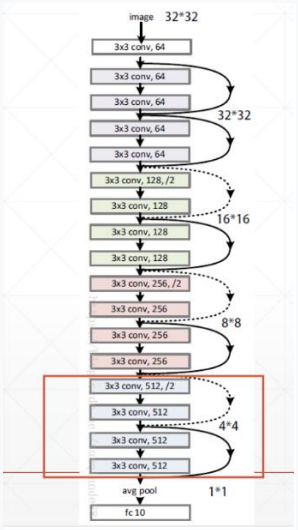
We can find that the LSTM model is very robust and NCNN model has a higher accuracy,so we have reason to believe that the LSTM model provided better robustness for the entire model, while the NCNN model made it close to higher accuracy.

We noticed that the CNN performed well in previous work. Maybe we can find a better model in CNN, so after screening .Due to the superior performance of Resnet, we decided to try it .

Resnet has a structure that add the identity mapping into the normal CNN to eliminate the adverse effects of the redundant network layer.

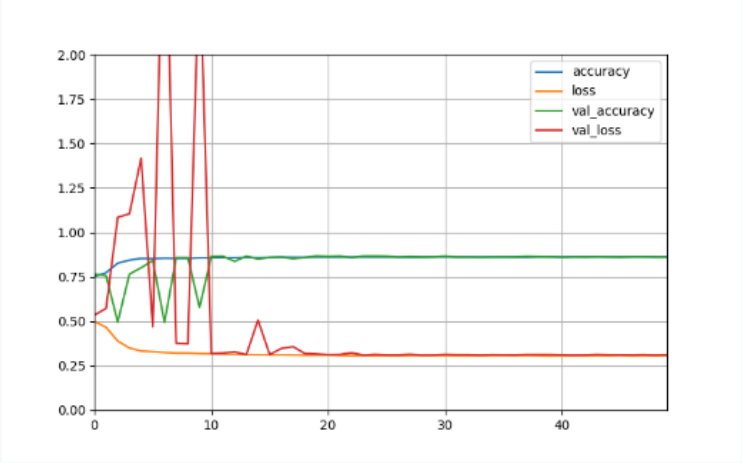


This identity mapping is also called short cut, It makes the neural network not degenerate even if it is stacked multiple layers。 So resnet is a construction concept that obtains higher accuracy by overlaying network layers.



This is a model structure of resnet 16. We can find that in the middle ,there are 16 CNN layers connected by short cuts ,so we called it resnet 16.

The resnet model we designed can change the number of intermediate model layers by adjusting parameters. In view of the huge amount of data and the limited computing power of the host, we can use resnet 10 as the experimental model, and we used it to perform the data set .



Figure： Resnet 10 model

Val-loss = 0.3097 Val-accuracy =0.8593 Test _loss = 0.3023 Test_accuracy = 0.863

Time = 10957

We can see that the experimental results was ideal. not only it had 86.3% accuracy, but also that the model had almost converged after 10 rounds traversal. It shows that the robustness of the model was very high.

We integrate all the models we used to make a chart:

Comprehensive experiment table

	Val- loss	Val -accuracy	Test - loss	Test-accuracy	Time
Resnet 10	**0.3087**	**0.8593**	**0.3023**	**0.863**	**10957**
LFCA	**0.3124**	**0.8570**	**0.3120**	**0.860**	**22640**
NCNN	**0.3292**	**0.8561**	**0.3251**	**0.8585**	**8445**
LSTM	**0.3732**	**0.8365**	**0.3663**	**0.8412**	**36224**

In the table we can see that under the same experimental conditions, the resnet 10 obtained the highest accuracy and used less time.so the Resnet 10 model is the best

We increased the number of training rounds , uesd resnet 10 to perform 100 traversals .

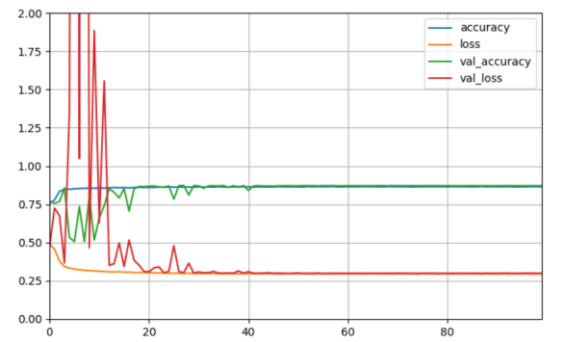


Figure : Resnet 10 model

Val-loss = 0.2937 Val-accuracy =0.8668 Test _loss = 0.2923 Test_accuracy = 0.8705

Time = 22964

Finally we achieved 87% accuracy. This is the best result we can achieve.