Case study: Improving Deep Learning in Inputs and Parameter Selecting

Yue Lian,

Australian National University
Yue Lian, u6175160@anu.edu.au

Abstract. This study describes the deep learning in GIS system and analysis how to change input to improve the learning effects with the data from UCI Repository. Back propagation network with pruning can perform properly in classical problem, classification. Also, evolution algorithm as a new method can perform as well as NN without not many parameters settled

Keywords: Pytorch; Variables; Input; Parameters; Prune; Evolution

1 Introduction

There is a significant increase in calculating speed of CPU and GPU between every generations of that. According to UserBenchmark (2018), a common website of CPU data, the increase in calculating speed is 11% between I7 7700k and 8700k. The number can represent that the time to train same data by the CPUs is decreased by 11%. It is notable that the publishing gap between the above two CPU model is 3 quarter and since 2005, main home-using CPUs' speed (from Pentium 4 to I7 8700k) dramatically jump by 700% (Intel (2018) & UserBenchmark (2018)). Therefore, the possibility dealing with problem by machine learning has been improved and machine learning, especially deep learning can be predicated having a relatively positive future.

Pytorch, a convenient learning library is introduced recently and gradually becomes a famours ML framework. With easier to use existed code frame instead of coding by people themselves, how to adjust input to obtain acceptable results and prune network to reduce the learning time becomes the main problem in ML. In the following part, some discussion will be based on "Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables" (Blackard & Dean, 1999) and involved dataset from UCI Machine Learning Repository (1999).

This study will be divided into two parts. First one focusses on input dealing, hidden neuron number (including depth of network), selecting some parameters choosing for optimiser model and pruning. Although it is difficult to predict why the deep learning is effective, experiences show that changing parameters or strategies in Deep Learning can give an effective model to predict. Additionally, this report attempts to utilize basic concepts of evolution algorithm to obtain better result than NN.

2 Dataset

Dataset is originally from Rawah, Comanche Peak, Neota, and Cache la Poudre areas of the Roosevelt National Forest in northern Colorado for these areas is natural without many human activities (Blackard & Dean,1999). The uploaded dataset from UCI is modified to some degree and this dataset is what the article originally used. The description will be divided into two parts, what is modified properly by the author, Blackard & Dean, and what is the continuous enhancement for this article. For the data selection of original picture, this is ignored for not the topic which are focused.

2.1 Source dataset (original author's input)

According to UCI (1999) and Blackard & Dean (1999), the originally consists of 54 attributes ,12 types and significant different numbers in different class.as shown below
1. Elevation (m),
2. Aspect (azimuth from true north),
3. Slope (°),
4. Horizontal distance to nearest surface water feature (m),
5. Vertical distance to nearest surface water feature (m),
6. Horizontal distance to nearest roadway (m),
7. A relative measure of incident sunlight at 09:00 h on the summer solstice(index),
8. A relative measure of incident sunlight at noon on the summer solstice (index),
9. A relative measure of incident sunlight at 15:00 h on the summer solstice(index),
10. Horizontal distance to nearest historic wildfire ignition point (m),
11. Wilderness area designation (four binary values, one for each wilderness area),

12. Soil type designation (40 binary values, one for each soil type).

Class	Data Number
1	211 840
2	283 301
3	35754
4	2747
5	9493
6	17367
7	20510

Figure 2.1

2.2 Modified dataset

Bustos and Gedeon (1995) introduce a method to classify slope and aspect. It will be clear to use 10 to 80 to represent the slope from 0 degree to 90 degrees. However, the whole dataset is mainly focus on the value <30, the largest degree is about 60. So, only 10 to 60 is used. This form can keep many features of number, ex. Sin and cos is also supported and have clear classification.

Slope	Converted number
<1%	10
<2.15%	20
<4.64%	30
<10	40
<21.5	*50*
others	60

Figure 2.2

As for aspects, the introduced method is make 4 variables (west, north, east and south) to present closed (1), relatively closed (0.5) and far (0). For every direction, if included angel is less than 45, this will be treat as 1. As same, less than 90 but not less than 45 will be treat as 0.5. Same with slope, the advantage of this change can keep main feature but more clear presentation: just 3 possibilities of variables.

For distance, all remains input is distance apart from soil type and wildness area designation. Distance is difficult to deal for too many possibilities in this area. In this dataset, the max distance for 10th input is 7117 but the lowest is 0. The method after consideration is using percentage to present it. The entire formula is (value – min_value) / (max_value – min_value). When most distances are converted, some inputs' percentage is extremely intensive on 80% approximately. The further step is changing formula: get a number that appears frequently and not large in whole frequently appearing's number set; (value – chosen number)/modified_max_value. After that, we can get a distributing averagely value from -100 to 100 and make it natural number as final result; This form can keep basic feature of distance, like larger distance have larger percentage and this convent do not change the basic relation. However, this method has some drawbacks. The largest one, percentages will lose some math feature and add some noises.

The Final change is most important one. The original data use 40 0/1 binary number to present the soil type or wilderness area designation. However, the special design from original author actually damages the training. Too sharp change of input and too many this change will seriously decrease the accuracy of final training. In a trial, 30 neurons in 1-layer increases to 20*120 and the best result is 50% with 4,5,6 and 7 types never predicted. From right to left, add value * I (I is auto-increase 1). We can get 1-40 by summing up the 40 variables.

For testing and training, the experiment will use two modes and 3 different change on dataset. Tow mode: 50000/861012 will be used as training set and the other will be test set; as introduced by original author, 1500 data for every class is used to be consisted as 7X1500 test and the others test the training effects. The 3 different changes are with all changes above, with only binary converting and keeping source dataset.

3 Parameters

In setting parameters, the consideration is adopting more common number to test rather than find an absolute best model. In common use, 0.001, 0.01, 0.05, 0.1 and 0.2 are relatively common learning rate with 0,0.1, 0.4, 0.9 momentum rate.

Xu and Chen (2008) claimed that neuron number and layers selecting should consider input data pairs, input dimensions. N/d <30, larger neuron gets better result but for >30, the formula should be n = C (N/ (d log N))1/2 (Here, n is number of layers, N is data pairs, d is input dimensions and C is a constant number decided on different model). Based on used data set, predicated best layer is 8 on

original dataset, 3 on remained dataset. Therefore, 1,3,5,8,10 will be the tested neuron layers number to verify the correct of this concept.
Due to enormous number of data pairs in one epoch, the option in the number of epoch will be relatively small as 1, 2, 3,5,10,20,30.

4 Experiment

The experiment is run on I7 6700k and NVIDIA 1070. We mainly focus on stable time and accuracy. All result is selected as best result or some featured result from 1000+ test. It is notable that all test has ±3% error.

4.1 original dataset

For this dataset, the effect is relative lower, and the output approximately does not have 4-7 output in lower layers. Adding neuron improve it slowly and need 20 or 30 epochs to train. Therefore, the method is not ideal.

Neuron number	Neuron Layer	Training accuracy	Test accuracy
30	1	60	61
60	10	66-81(not stable)	64
120	10	61-75(not stable)	60

Figure 3.1

4.2 dataset with only binary converting

For this dataset, when increasing number of neuron or increase the depth. The ability of memory is significant increased. The output will be stable around 60%. However, overfitting can be observed. More complex network produces more overfitting result and to get stable result need 8 epochs or more in complex network.

Neuron number	Neuron Layer	Training accuracy	Test accuracy	loss
30	3	70	66	-3.24
60	3	66	68	-4.59
120	8	90	60	-6.5

Figure 3.2

4.3 dataset with all changes

The convergence of data will be extremely fast. The stable result can be obtained after 1 epoch. The all data below is tested in 1-3 epoch and use learning rate decay to avoid too much data training destroy the result as shown in 3.2. However, more training or more complex network have not improved the effects due to lost information. But the modified percentage of distances indeed improve the training speed and get test result same as only binary converting.

Neuron number	Neuron Layer	Training accuracy	Test accuracy	loss
30	1	72	66	-3.20
60	2	72	65	-3.7
120	1	72	65	-3.9

Figure 3.3

4.4 1500x7 pairs dataset

This dataset has not proper result due to few data training. In training period, it is easy to get over 90%, even 95% accuracy but not over 50% in test. The reason why lead to this situation is that the original author’s method against a basic concept: more data give better result as a hard difficulty for our world to obtain a positive ML result is lacking data, especially for recognize cover type which is a complex topic. Thus, the method is not recommended to utilize.

4.5 Parameter

Above all experiments, the changed parameters indeed present different results. For momentum, 0 momentum is worst leading to that the learning will stop on a local minimum and limited result in 40% approximately. 0.1 and 0.4 give relatively same result which is better than 0.9.

In learning rates, 0.001 is best one and the predicted reason is that the data pairs are enormous. Zeiler (2012) illustrates that from 0.01 to 0.0000001, in his model, the training effect are increased. If dataset is enormous, lower learning rate is better.

Neuron layers' number and neuron's number is involved with epoch. Deeper layer need more larger width. In this test (Figure 3.2), 3 layers need 30 or 60 neurons to get best accuracy in 8 epochs, but 8 layers need 120 neurons to get it in 15 epochs or more. Meanwhile, the concepts of Xu and Chen is identified effective. About 8 epochs, the original dataset gets best result.

5 Pruning

After achieving the target goal, how to reduce the complex network architecture should be another method to accelerate the training method. When the neuron should be cut is introduced as when two neurons' output is same, adding two neurons' weight in one and pop the other off network (Gedeon & Harris, 1995).

Same introduced by Gedeon and Harris (1995), calculating the angle between two vectors of outputs can be implemented by cos θ = (a * b)/|a|*|b| (should expanded to n-d vector and move all value to 0.5 0.5 to 0 – 180 instead of just 90). To guarantee the misprune, the pruning will begin after training 1 epoch because in the beginning, output is in completely random.

5.1 Result

Neuron number	Neuron Layer	Training accuracy(best-worst)	Test accuracy (best-worst)	Pruned number(mean)
30	1	71 -60	61 - 58	30 – 14
60	1	65 -20	60 -22	60 – 29
120	1	65-20	63 -21	120 -76
120	3	55- 25	53-20	120*3-200(about)

Figure 5.1

5.2 Analysis

The results are completely different. If after pruning the result can be kept, the total result will be acceptable, verse vice.

The two reasons can be predicated that some well-trained neuron has been cut and after cutting, learning rate decay obstacle the remains to recovery original result. Generally, single layer has larger possible to obtain acceptable result than low-result

For multi-layer network, pruning cause more serious and negative result. The reason is no limitation to stop prune one layer: there is a test in total testing that 3x120 hidden layers will be reduce to 61-27-40. A larger reducing in the middle layer will cause serious loss of information, even other layers are keep in larger degree.

6 Evolution Algorithm

6.1 Algorithm

In the final part, a basic, fixed topological evolution algorithm has been implemented and tested. The initiate state is one person(network) in population(networks) with fixed weight and no backward. The initial action use production of person1*random (0, 0.5, 1, 1.5, -0.5, -1) x person1*random (0, 0.5, 1, 1.5, -0.5, -1) to get new weights and using these weights to add new pair of networks. Every pair of networks consists of one male and one female. Reproducing action select a male and a female randomly to reproduce new pair until the population reach limitation. After reaching limitation, a starvation will reduce a half number of people in male and female both. This algorithm reduce parameters setting and not too complex.

6.2 validation (fitness function)

In EA, there is no general training step; every adjusted weight will be produced by reproduction method in new generation. Based on natural selection and fitness survival, a number of random batchs in about 80% dataset will test the fitness and after starvation, the unsuitable networks will be reduced. After reproducing several generations, 20% of original dataset will validate if overfitting or not.

6.3 Result and analysis

Same with normal NN, more testing batchs will reduce overfitting because testing bias from few datasets will cut some strong people and keep some fake-strong networks. Larger limitation of number of networks will improve the whole result but need more time to train, test and validate.

If with whole 80% dataset to test every person, the result will be 65% as same as NN (not stable), but with a few random dataset, 38% validation result and 65% testing result can be found

7 Conclusion

From this experiment, the some dealing to input can improve the learning effects of deep learning and in large degree speed up the training effects. However, this method may break some unpredicted relation of numbers and decrease the memory ability. In other side to consider the memory ability, the dealing can decrease overfitting according to results. For parameter selecting, it is recommended to use Xu and Chen’s formula to choose layers’ number. Also, evolution algorithm is simple way to get acceptable result without complex parameter setting.

References

Blackard, J. A., & Dean, D. J. (1999). *Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables.*

Gedeon, T. D., & Harris, D. (1995). *NETWORK REDUCTION TECHNIQUES.*

Gedeon, T., & Bustos, R. (1995). *DECRYPTING NEURAL NETWORK DATA: A GIS CASE STUDY.*

Intel. (2018). *Legacy Intel® Pentium® Processor*. Retrieved from Intel: https://ark.intel.com/products/series/78132/Legacy-Intel-Pentium-Processor

UserBenchmark. (2018). *Compare*. Retrieved from UserBenchmark: http://cpu.userbenchmark.com/Compare/Intel-Core-i7-8700K-vs-Intel-Pentium-4-380GHz/3937vsm11003

Xu, S., & Chen, L. (2008). *A Novel Approach for Determining the Optimal Number of Hidden Layer Neurons for FNN's and Its Application in Data Mining.*

Zeiler, M. D. (2012). *ADADELTA: AN ADAPTIVE LEARNING RATE METHOD.*