Refining Input in Neural Network for

Rule-based Data Understanding

Zhaohui Zhang

Research School of Computer Science, Australian National University

u6342695@anu.edu.au

Abstract:

The problem of processing features and finding the best input to feed neural network is well-known, which will contribute to the performance of neural network. In practice, black-box prediction is not always satisfactory, understanding the input and organizing the input are important when training neural network. Data can be analyzed based on visualization or graphical method, which could improve performance on predicting. In addition, in many cases, large neural network architecture is required, followed by heavy calculations and overfitting problem. Neural network pruning is a good way to solve these problems. Thus, decrypting neural network data and pattern reduction are chosen as the improvement method. Compared with the result using method nearest neighbor (NN) classifier [3], the result of decrypting data method performs better. Method is implemented on Thyroid disease dataset [1] from the UCI repository. For CNN, I choose Cifar-10 as dataset for experiment.

Keywords: Input Processing; Convolutional Neural Network; Neural Network; Neural Network Pruning

1 Introduction

1.1 Background and motivation

In supervised learning, training set is given containing labelled instance. For classification, the task is to induce a predictor that accurately predicts the label of the test set. For input of neural network, processing features is significant in real-world applications of learning task [2], which will directly affect the performance of predictor. Thus, appropriate encoding pattern for each feature is pivotal. Thus, feature encoding and dropping irrelevant features are both crucial to neural network, which could contribute to good learning and prediction. We need to understand and analyze the data, to do this, visualization and some basic calculation are required. Additionally, for convolutional neural network, we usually take images as input. One single image can be regarded as hundreds or thousands features, so it is significant to properly process the input to improve the performance of convolutional neural network and speed up the training procedure.

In practical, it is common to use large neural network architectures to solve the problems. With the increasement of task complexity, more complex neural network structure is needed. For instance, AlexNet is proposed in 2012, which is the champion of ImageNet 2012. AlexNet is a simple network nowadays, which contains 5 convolution layers and 3 fully connected layers. However, even it is a simple network structure, 60 million parameters are required in AlexNet. In 2016, Kaiming He proposed ResNet-1001 which contains 1001 layers [9], which is calculation infeasible for a normal computer. The idea is that among the many parameters in the network, some are redundant and don't contribute a lot to the output.Thus, it is an important technique pruning the network, which could save time for training. In neural network, we expect that neurons from the same layer have different behaviors, in other words, find different features of input. To achieve that, we need to remove the neurons which have similar or same behaviors with other neurons. If we find [10]:

1. Weights between two units are similar or identical, one of them is regarded as redundant
2. Unit performs no function can arise in a number of ways
3. Group units together produce a constant effect across the pattern set

All these possibilities are the targets of network pruning problem.

1.2 Dataset

For different neural network architecture, I choose different dataset for experiment. The datasets are described as followed.

1.2.1 Simple neural network

The dataset for this study is Thyroid disease dataset from the Garavan Institute in Sydney, Australia for classification problem. For the dataset, it contains 2800 training (data) instances and 972 test instances, with plenty of missing data (signified by '?'). There are 29 attributes and 3 classes, whose data type is Boolean, categorical or continuously-valued.

There are 29 features in the dataset, but not all of them are useful. Among these features, several of them make no sense which need to be removed. After that, 23 or so features will stay.

In this paper, I present a classification learning that achieves high accuracy, comparable to method nearest neighbor (NN) classifier [3]. First, we need to detect the irrelevant features by summarizing data to minimize use of input features e.g. when features are tests that have an associated cost or when features have extreme unbalanced value. Then, decision will be made on encoding features according to the characteristic of the feature value. Other than these method, for training process, gradient descent method is chosen to correct the parameters between each layer. We use loss and accuracy to evaluate the result. After modifying with the methods above, the accuracy of test set can reach 97.63%.

1.2.2 *Convolutional neural network*

The CIFAR-10 dataset consists of 60000 32x32 colour images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images.

The dataset is divided into five training batches and one test batch, each with 10000 images. The test batch contains exactly 1000 randomly-selected images from each class. The training batches contain the remaining images in random order, but some training batches may contain more images from one class than another. Between them, the training batches contain exactly 5000 images from each class.

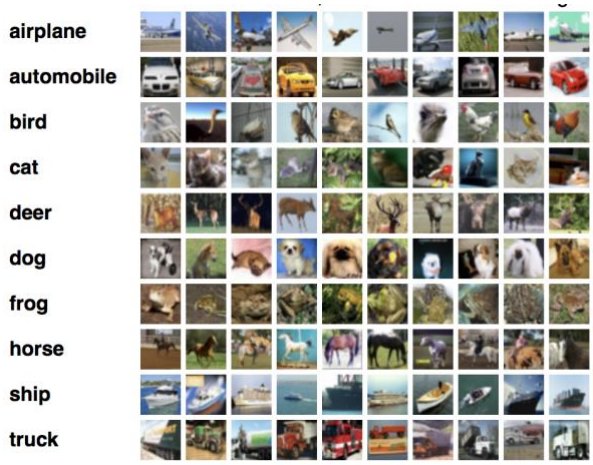


Fig. Sample images from Cifar-10

In paper, I will present a classification problem on Cifar-10 dataset. The neural network to solve this problem is convolutional neural network. To improve the performance, process the input is necessary. In addition, usually convolutional neural network has high dimensional parameters. So, both process input method and network pruning method are applied in this paper.

2 Method

2.1 Simple neural network

Data Pre-process

After observing the data set, there are many missing values. If I delete all rows which include missing values, the dataset will shrink a lot. Thus, we need to process the missing value (i.e. '?') rather than delete them.

First, convert the missing value into -1.

Data Analysis and Encoding [3] (improvement method)

Here I list the contributions of some features as examples.

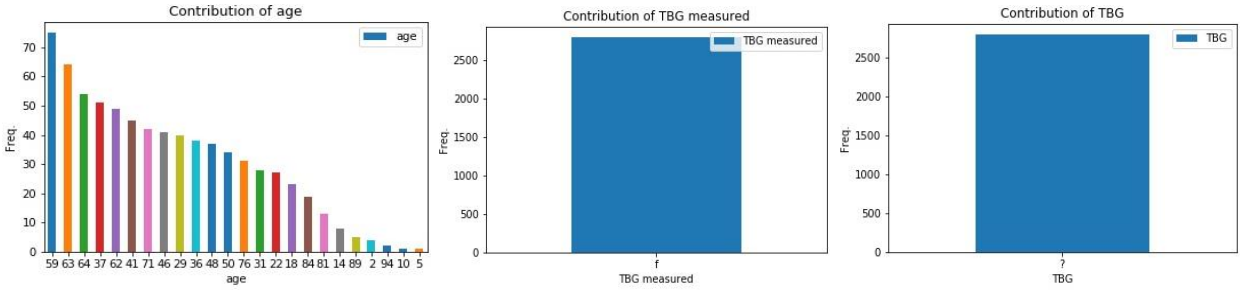


Fig. contribution of age, TBG measured and TBG

- As for age, which is continuous value, for *age* attribute, we need to normalize data over the range 0 - 1 for the network, for the logistic function.
- We can see that all value of TBG measured is 'f' and TBG is '?'. No doubt that TBG measured and TBG need dropping.
- Attributes TBG measured and TBG can be regarded as irrelevant attributes, which make no sense to the learning process.

Other Boolean type and categorical type values:

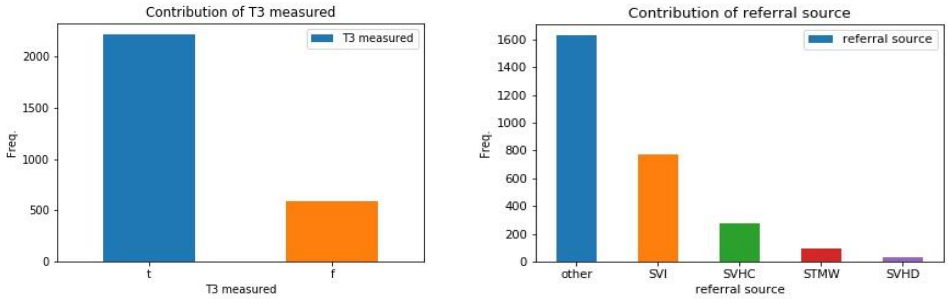


Fig. Sample of Boolean type attribute and categorical attribute

- For the values with Boolean type (only list three of them), simply convert t into 1, f into 0.
- The value of referral source is categorical, thus, new value should be assigned to them.
- Other <-0; SVI <-1; SVHC <-2; STMW <-3; SVHD <-4
- As the values of them are simple, no more encoding needed just simple squashing function.

Structure of neural network

Three-layer fully connected neural network is adopted for this study: Input layer -> hidden layer -> output layer
Two fully connected layers are included.
Number of hidden unit: 32 hidden unit perform best.

Number of hidden unit	testing accuracy
8	93.29%
18	97.04%
28	96.91%
32	97.46%
48	96.91%

Tab. performance of different number of hidden unit

Activation function: ReLU (Rectified Linear Unit) Activation Function
The ReLU is the most used activation function in the world right now. Since, it is used in almost all the convolutional neural networks or deep learning.

$$F(x)= max(x,0)$$

Back propagation:

Neural networks can learn their weights and biases using the gradient descent algorithm. The fast algorithm for computing such gradients is known as *backpropagation*, which is for the minimum of the error function in weight space using the method of gradient descent.

I choose backpropagation to train the neural network. The formula of backpropagation is as followed:

repeat until convergence {

$\theta_0 := \theta_0 - \alpha \partial J(\theta_0, \theta_1) \partial \theta_0$

$\theta_1 := \theta_1 - \alpha \partial J(\theta_0, \theta_1) \partial \theta_1$

}

2.2 Convolutional neural network

Input processing:

In machine learning and data mining, several steps such as data preparation, data preprocessing and feature extraction occupy almost half of the data engineers' work time. At the same time, the result of data processing can directly affect the efficiency of models.
Images in Cifar-10 are 32*32 colour images. Before putting the data into convolutional neural network, images need processing in order to get the better performance. Data processing is an important step which ensures that each input (i.e. pixel) has a similar data distribution, which makes convergence faster while training the network. Data normalization is done by subtracting the mean from each pixel, and then dividing the result by the standard deviation. The distribution of such data would resemble a Gaussian curve centered at zero. For image inputs we need the pixel numbers to be positive, so we might choose to scale the normalized data in the range [0,1] or [0, 255]. For my dataset example, I choose 2 ways to process the data:

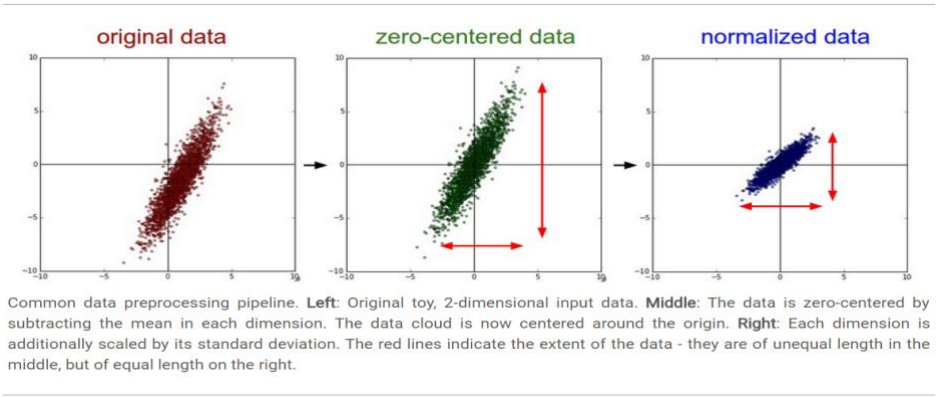
1. Z-score standardization: subtracting the mean of each dimension

$z = \frac{x - \mu}{\sigma}$ x: origin data u: mean

2. Normalization: scale the image to [-1,1]

   Each dimension is additionally scaled by its standard derivation.

$$\hat{v} = \frac{v - mean}{std} = \frac{v - 0.5}{0.5}$$



Common data preprocessing pipeline. **Left**: Original toy, 2-dimensional input data. **Middle**: The data is zero-centered by subtracting the mean in each dimension. The data cloud is now centered around the origin. **Right**: Each dimension is additionally scaled by its standard deviation. The red lines indicate the extent of the data - they are of unequal length in the middle, but of equal length on the right.

Fig. Different processing methods

The figure shows the different input process methods. The left picture indicates the original data, which has no rules to follow. The middle one shows Z-score standardization, which is to subtract the mean of each dimension. The right picture shows the normalized data. Each dimension is additionally scaled by its standard derivation.

Network Pruning:

Recently, it is common to use large neural network architectures to solve the problems. With the increasement of task complexity, more complex neural network structure is needed. To implement network pruning method, we need to know which neurons should be removed.

1. Weights between two units are similar or identical, one of them is regarded as redundant
2. Unit performs no function can arise in a number of ways
3. Group units together produce a constant effect across the pattern set

The ranking can be done according to the L1/L2 mean of neuron weights, their mean activations, the number of times a neuron wasn't zero on some validation set, and other creative methods. After the pruning, the accuracy will drop (hopefully not too much), and the network is usually trained more to recover. If we prune too much at once, the network might be damaged so much it won't be able to recover.
We need to calculate the angles of weights of each two neurons from the same layer. If the angle is less than 15 (or even 30), the neuron is regarded as redundant, which should be discarded.

In this work, we need first calculate cosine, then get the angle which is used as the similarity measurements and for two vector a and b.

$$cdist(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a} \cdot \mathbf{b}}{|\mathbf{a}||\mathbf{b}|}$$

After getting cosine value, we use arccos function to get the angle.

Structure of neural network

ResNet 18 is chosen as model for training Cifar-10. ResNet is simulated by the structure shown below.

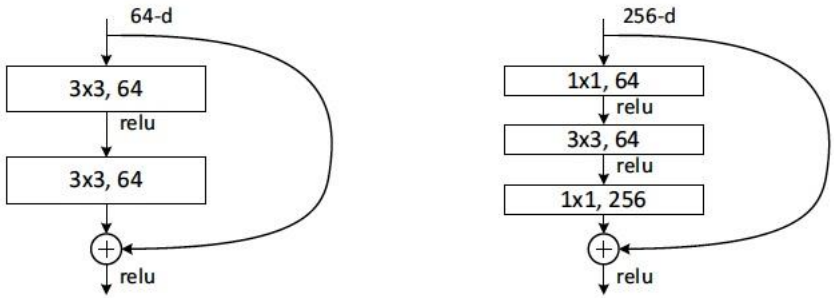


Fig. architecture of residual network

3 Results and Discussion

3.1 Simple Neural Network

Evaluation method:

Loss function:

It describes how far off the result your network produced is from the expected result - it indicates the magnitude of error your model made on its prediction.

We could see the error of the result from loss.

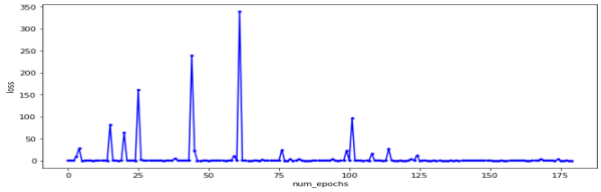


Fig. loss with different number of epoch

Loss falls down and become stable after under training, at the end of 180 epochs, loss is 0.0417. It is quite good with the value 0.0417, which indicates that the error is small, in other words, accuracy is high.

Confusion matrix:

A confusion matrix is a table that is often used to describe the performance of a classification model on a set of test data for which the true values are known. The confusion matrix itself is relatively simple to understand, but the related terminology can be confusing.

Confusion matrix for training set and test set are as followed:

```
Confusion matrix for training:

 2656     10     1
   73     51     0
    7      0     2
[torch.FloatTensor of size 3x3]
```

```
Confusion matrix for testing:

 937   10    1
  13   12    0
   5    0    0
[torch.FloatTensor of size 3x3]
```

From the matrix, we can see that the prediction of the first class is quite good. However, the accuracy of predicting second and third class is poor, less than half of the values are correct. Although the accuracy is high, balanced accuracy will be low for this model of classifier.

Comparison:

Compared with kNN method [3] given by P. Viswanath, M. Narasimha Murty& Shalabh Bhatnagar, the accuracy in this report is higher. With KNN method, the testing accuracy is 94.40%, however, the accuracy in this report is 97.63%.
Learning in NN classifiers consists of simply storing the training instances in memory, leaving all the computation to the classification phase. For that reason, these kinds of algorithms are called lazy learners [5]. The kNN algorithm is a generalization of the NN algorithm, where the prediction is based on a majority voting of the nearest k neighbors. [3]
Compared with kNN algorithm, method adopted for this study is smarter, which could get result via calculation according to the features and functions not just simply compare with the previous results.

3.2 Convolutional Neural Network

Loss function and test accuracy of data processing:

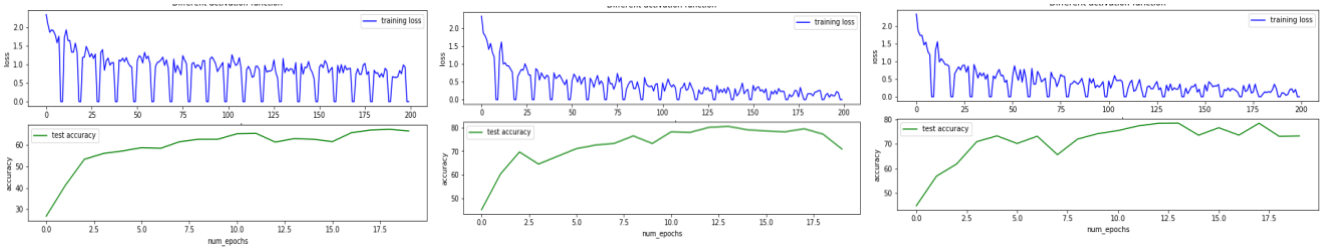


Fig. comparison of origin, Z-score standardization and normalization

The left figure indicates the data without processing. The middle figure shows the performance of normalization while the right one shows Z-score standardization.
As I observed, the performance of using data without processing is poor. The overall accuracy is under 70% and the loss is quite large compared with others. As for normalization, it performs best. The highest accuracy is over 83%. Accuracy of Z-score standardization is also higher than origin data, but not as good as normalization, which is almost 80%.
Thus, in this case, normalization performs best. However, result might be different on different datasets and models, so experiment is needed.
Loss function and test accuracy of pruning:

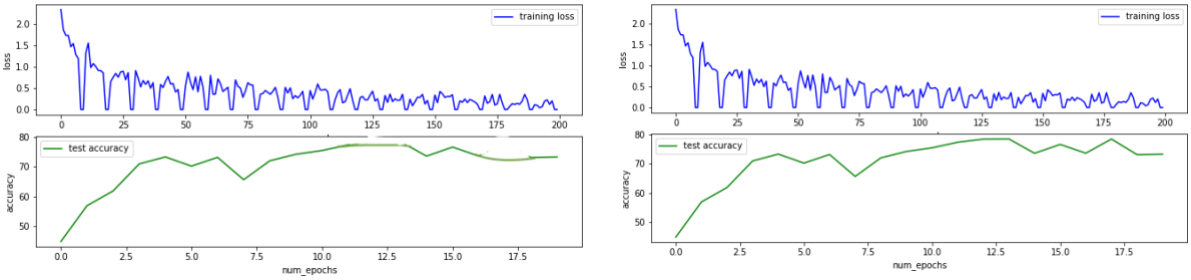


Fig. comparison of origin network and pruning network

The left figure indicates loss and accuracy of pruning network, while the right figure shows loss and accuracy of the network without pruning. We can see that, after pruning, the accuracy falls slightly. Before pruning, the accuracy is 83%, while after pruning, accuracy falls to 79%. However, the training time shrinks a little.
Comparison:
In paper [10], authors built a second-layer fully-connected RBM. This model classifies the CIFAR-10 test set with 78.9% accuracy. Convolutional DBN, one must decide what to do with the edge pixels of teh images, using a combination of locally-connected convolutional units and globally-connected units, as well as a few tricks to reduce the effects of overfitting. While in our study, the accuracy of normalization can reach 83%, which is higher than the accuracy in paper ‘Convolutional Deep Belief Networks on CIFAR-10’ [10].

Model	Two-layer performance
3	77.46%
4	**78.90%**
5	77.56%
6	77.27%

Fig. Result of Convolutional DBN on Cifar-10

4 Conclusion and Future Work

In this report, we have shown the example using Thyroid disease dataset to properly encode and find irrelevant features. Normalization, squashing function and logarithm function etc. could be used as method of encoding features. After analysis, we can see that appropriate encoding pattern for each feature is significant and irrelevant features could hurt the overall accuracy. Thus, feature encoding and dropping irrelevant features are both crucial to neural network, which could contribute to good learning and prediction. Other than the method above, in order to produce a good neural network, we also adopt backpropogation for better performance. For CNN, properly processing the data could contributes to better performance.
For pruning, by discarding redundant neurons, we can get smaller neural network. In the result, we can see that with a good neural network architecture, the accuracy only falls slightly. However, with minor accuracy loss we can save training time.

Future work:

1. Improve the accuracy of prediction of class with few instances. As mentioned above, the accuracy of class with large number is almost 100%, however, for class with few instances, the accuracy is less than 50%.
2. The model should be more robust to irrelevant features. The effect of irrelevant features can be avoided if the model itself can ignore the irrelevant features.
3. Improve balance of bias & variance.
4. Network reduction technique should be adopted to more layers.
5. When pruning the convolutional neural network, another option would be to reduce the weights in each filter, or remove a specific dimension of a single kernel.

5 References

[1] Dua, D. and Karra Taniskidou, E. (2017). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

[2] Güvenir, H. Altay (1998). A Classification Learning Algorithm Robust To Irrelevant Features. Artificial Intelligence: Methodology, Systems, and Applications (pp. 281-290). Berlin, Heidelberg

[3] P. Viswanath and M. Narasimha Murty and Shalabh Bhatnagar (2006). Partition based pattern synthesis technique with efficient algorithms for nearest neighbor classification. Pattern Recognition Letters (pp. 1714-1724). India

[4] Bustos, R. A., & Gedeon, T. D. (1995). Decrypting Neural Network Data: A GIS Case Study. In Artificial Neural Nets and Genetic Algorithms (pp. 231-234). Springer, Vienna.

[5] Fix, E., Hodges Jr., J., 1951. Discriminatory analysis: non-parametric discrimination: Consistency properties. Report No. 4, USAF School of Aviation Medicine, Randolph Field, TX.

[6] George H. John, Ron Kohavi and Karl Pfleger (1994). Irrelevant Features and the Subset Selection Problem. Machine Learning Proceedings 1994 (pp. 121-129). San Francisco (CA)

[7] Kaiming He, et al (2015). Deep residual learning for image recognition. arXiv preprint arXiv:1512.03385.

[8] Kaiming He and Jian Sun (2015). Convolutional neural networks at constrained time cost. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

[9] He K., Zhang X., Ren S., Sun J. (2016) Identity Mappings in Deep Residual Networks. In: Leibe B., Matas J., Sebe N., Welling M. (eds) Computer Vision – ECCV 2016. ECCV 2016. Lecture Notes in Computer Science, vol 9908. Springer, Cham

[10] A. Krizhevsky (2010). Convolutional Deep Belief Networks on CIFAR-10

[11] T. D. Gedeon. Network reduction techniques. In Proceedings International Conference on Neural Networks Methodologies and Applications, volume 1, pages 119–126, 1991.