Accuracy improvement utilizing restarts on Convolutional neural network based on Fashion MINST

Yanpeng Zhao
University of Wisconsin-Madison
Zhao374@wisc.edu

Abstract

The goal of this research is to improve the accuracy on convolutional neural network based on fashion MINST database. More specifically, the research is aiming to find the relationship between restart numbers on training process and the total extend of learning improvement. At the same time, several algorithms on utilizing these restart numbers is to be compared and selected. Finally, the conclusion is drawn that more restart we make on training process on convolutional neural network, the less profit on accuracy improvement we gain form restart process.

Introduction

1.1 Overview

Artificial intelligence has been applied to more and more areas at the basement of hardware technology mature. Among artificial intelligence including PAC learning, RC learning, Online learning, Neural network or convolutional neural networking which is applied in this research is especially popular and being used in different classifying, predicting scenarios. Fashion MINST database (figure 1) is dataset consisting 60000 of training images including labels, and 10000 test images accompanied with labels. The goal of this neural network is to classify those images recognizing for example which one is high heel shoes, which one trousers.

1.2 Neural Network

For studying the relationship between learning profit and restart numbers, we choose simply constructed convolutional neural network. The CNN is constructed with two two dimensional layers. The first layer is in shape 24*24*32 and using 32 5*5 filters. The shape is 24*24 because the input is reshaped to 28*28 pixels images, and filter is 5*5. Multiplying each 5*5 pixels sub-image in original input makes the output of first layer diminished by 4 in each side. 32 different filters make the first layer 32 different filtered 24*24 image. After the first convolutional 2-dimension layer, a max pooling which selects 1 of each 2*2 pixels in first convolutional 2-dimensional layer output is added. Dimension of first pooling output is diminished to 1/4 which is 12*12*32. Same process is gone through in second convolutional two-dimensional layer. The output of second convolutional layer becomes 8*8*64. Same pooling mechanism is utilized after second convolutional layer. The dimension is further step down to 4*4*64 dimension. Now 4*4*64 output is transformed into one dimensional linear 1024 inputs for a multilayer perceptron with activation function of SoftMax. (Shown after figure 2) And the final classifying result should be a one-hot label with length 10 for there are in total 10 different classes. (Figure 2)

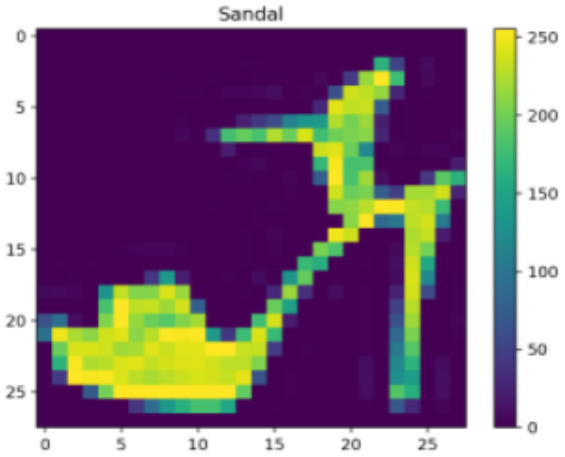


Figure 1: Fashion MINST example

Layer (type)	Output Shape	Param #
conv2d_2 (Conv2D)	(None, 24, 24, 32)	832
max_pooling2d_2 (MaxPooling2	(None, 12, 12, 32)	0
conv2d_3 (Conv2D)	(None, 8, 8, 64)	51264
max_pooling2d_3 (MaxPooling2	(None, 4, 4, 64)	0
flatten_1 (Flatten)	(None, 1024)	0
dense_1 (Dense)	(None, 10)	10250

Total params: 62,346
Trainable params: 62,346
Non-trainable params: 0

Figure 2: The verbal structure of CNN used for the research

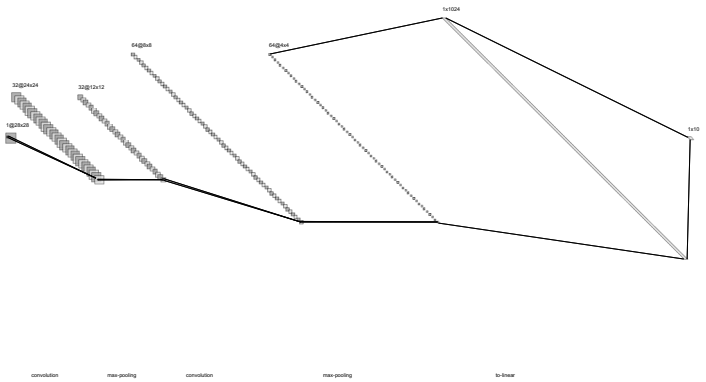


Figure 2: neural network diagram

1.3 Activation, Loss Function

RELU Max $(0, x)$ which when the input is lower or equal to 0 is outputted as 0, or output is input itself is used in both 2 convolutional layers (Figure 3). And for the last multilayer perceptron, we use SoftMax to output out one-hot result. Loss function is categorical cross entropy that is the sum of log predicted output times the real label.

$$\text{Loss} = -\sum_{j} d_j \log P_j$$

$$\text{SoftMax: } P_k = \frac{\exp(z_k)}{\sum_i \exp(zi)}$$

Using chain rule, we can get: $\frac{\partial \text{Loss}}{\partial w_j} = \frac{\partial \text{Loss}}{\partial p_i} * \frac{\partial p_i}{\partial x_j} * \frac{\partial x_j}{\partial w_j}$

The Gradient of loss which is essential to observe global optima or local minima is

$$\frac{\partial \text{Loss}}{\partial z_k} = \sum_t \frac{P_i}{t_i} * P_i P_k (i \neq k) + \frac{t_k}{P_k} P_k * (P_k\text{-}1) = P_k\text{-}t_k$$

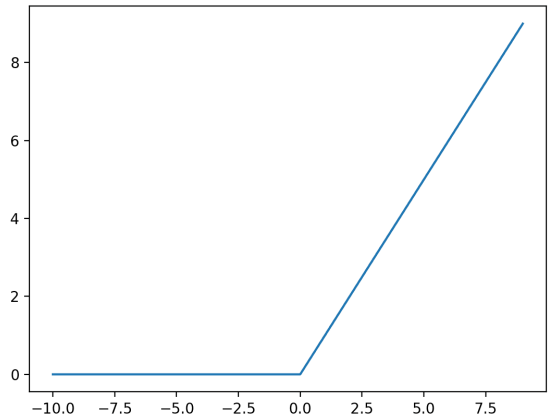


Figure 3: RELU plot

The accuracy improvement observation is due to the decrease of PE (acronym of percentage error) value of a trained model. When PE value decreases, the accuracy of an expert improves.

Restart enables training process randomly to start at the point of function. This randomly restart is functioned to escape local minima and ideally can approach to global minima between predicted label and truth label. In this case, more times the training process restarts, the more possible we can get to global minimal. But the truth is, the improvement of possibility finding global minima decreases along with more restarts. In the rest part of this essay, ideal restart number and algorithm utilizing those restarts is going to be introduced. A function between restart number and accuracy, relationship between restart number and learning rate would also been drawn.

Epoch and accuracy

An expert is trained using validation split of 0.25 which avoid repeat training on same training and test dataset. Batch is set to 100 which 100 different training images plus labels are put in single epoch. Before finding out the relation between accuracy and restart number, we need avoid overfitting problem first. Overfitting is when training set is overly trained in convolutional neural network model and hypothesis is too much based on training set which includes noise. It leads test dataset is terribly predicted. The accuracy of the expert predicting training set and test set differed a lot.

First, I use 10 epochs and see the trend. Under descripted convolutional neural network setting, the graph shows an increasing trend on both predicting on training set and testing set: the more epochs applied to the model, the more accurate it can predict on both training set and test set. The graph is shown in figure (Figure 6). But the trend is not always increasing as the epoch number increases. When epoch number passes the critic point where training on epochs gives positive gain to test set prediction accuracy, overfitting occurs. As the training set accuracy improves and reach to 0.97 after training 100 epochs, the test set accuracy only stays at 0.89 (Figure 7). Cross-Entropy loss of test set prediction approaches to 0.9. By observing when the test set accuracy can reach the vertex, we can select an epoch training number to let training process stop early. (Figure 4) After 20 arbitrary restarts of our learning model on same training dataset, the mode perfect epoch between these 20 trials is 13. (Figure 5) So, we will select 13 for number of epochs in further probing to relation between restarts and accuracy.

```
low = 999
indexx=0
for i in range(len(history.history['val_loss'])):
    if history.history['val_loss'][i]<low:
        low=history.history['val_loss'][i]
        indexx=i
```

Figure 4: Find out epoch avoid overfitting

```
[13, 13, 13, 13, 13, 13, 12, 12, 12, 13, 13]
```

Figure 5: Mode of 10 trials is 13

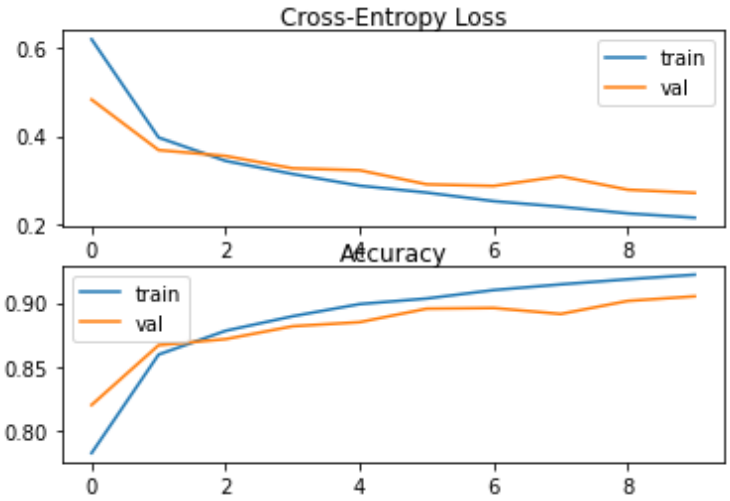


Figure 6: epoch and accuracy/Loss over 10 epochs

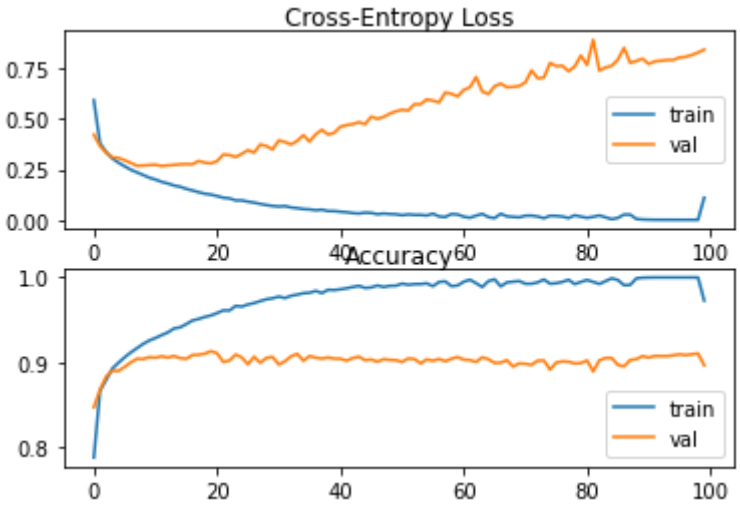


Figure 7: Overfitting occurs when more epochs involve in training process

Restart Algorithms

3.1 Simple version algorithm

It is not easy to acquire a perfect way to get global minima in loss function which reaches highest accuracy. While there are some optimizers enable us jump out of local minima in search of global minima. And Adam is relatively more effective in optimizer and more popular comparing to other optimizer functions like SGD or Gradient descendent. Adam automatically adjusts learning rate for each weight. But powerful optimizer at the same time weakens the usefulness of restart itself. Directly gradient descent might be a better way to understanding how restart improves the accuracy, but not many people applying gradient descent as optimizer in their neural network model. Gradient descent is the easiest way to adjust different weight and very easy to understand and implement. The problem is it can be stuck at local minima or saddle point. Gradient descendent

also require memory to compute derivative of entire dataset. Such terrible optimizer does not worth for research in this case. The algorithm for best utilizing restart is wrote below:

3.1.1 Algorithm steps

Input our desired low PE value for the trained model
Have ten restarts on the model with 13 epochs each
If single restart reaches the PE value we want, the iteration is jumped out
Else it selects the best or lowest PE value among ten restarts

The restart number doesn't have to be exactly 10 if we want to observe relation between profit gain of accuracy and restart number. The conjecture (Figure 8) is that the more restart number we have on our trained model, the lower profit we gain from the learning. But the accuracy inevitably goes higher accompanied with restart number. A function between accuracy improved and restart number is shown in figure (Figure 9). We now take the derivative or gradient of the curve. The value finally converges to zero as the restart number goes high.
There are smarter and more directly related to loss function local minima restart algorithms existed in artificial intelligence area.

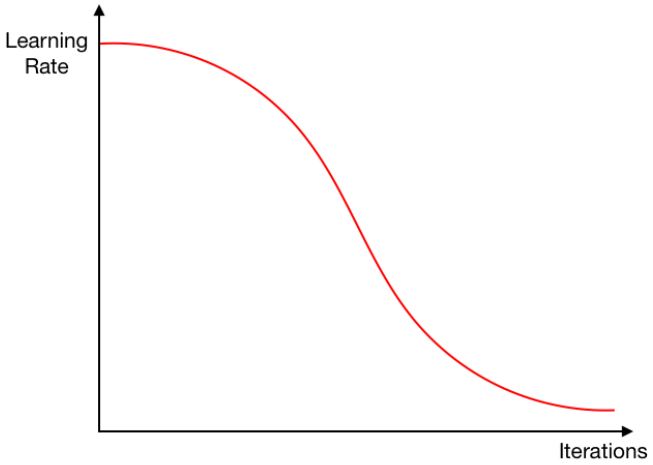


Figure 8: The conjecture on restart number and learning profit

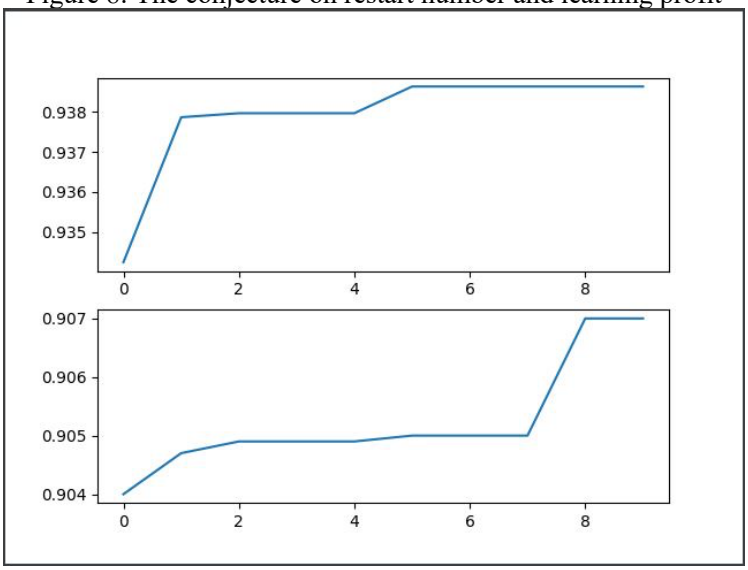


Figure 9: The above is relationship between train set accuracy and restart
below is relationship between test set accuracy and restart

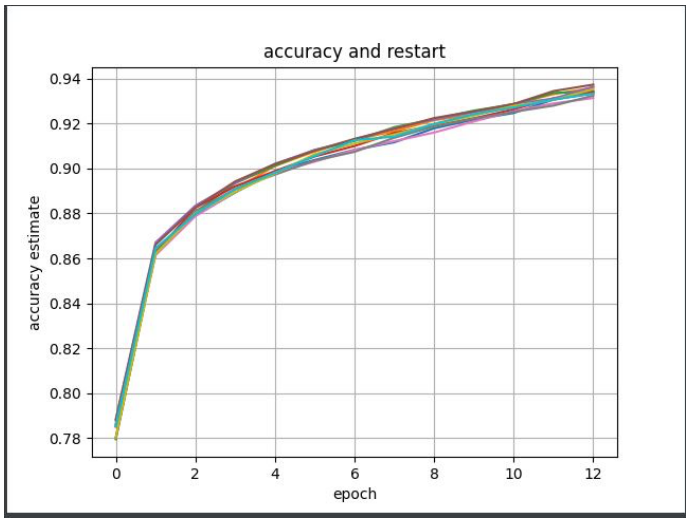


Figure 10: The performance of 10 restarts
*Ten restart is not enough to see salient discrepancies between restarts.
Once the number goes higher, we can see more significant improvement

3.2 Bayesian stopping rules

Bayesian stopping method consists of three parts that is prior function, likelihood function and loss function that premised in every restart algorithm. The likelihood relates the unknown true discrete distribution over local optima to the observed quantities. In combination of prior function on the unknowns, the posterior probability to unknown true distribution can be pertained. And this algorithm stops whenever the expected loss under current posterior would increase with further starts.
In conclusion in equation:

Present Uncertainty = Past Uncertainty * predictive updating factor

That:

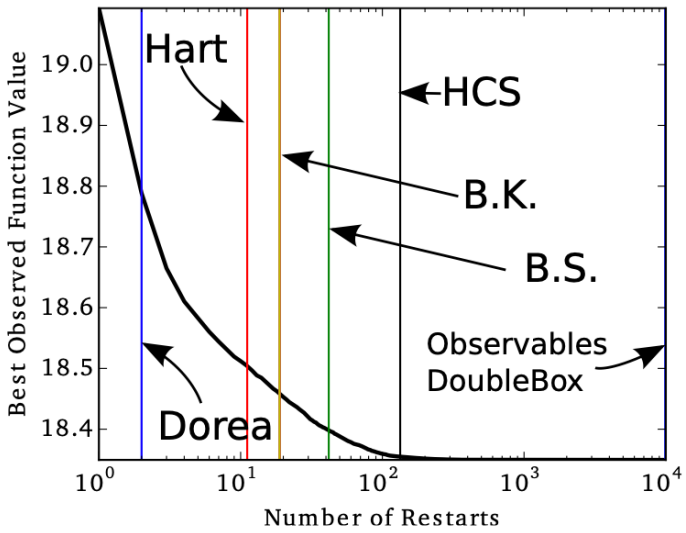
$$\underset{\text{Posterior}}{\mathrm{P}(\theta|data)} = \underset{\text{prior}}{\mathrm{P}(\theta)} * \frac{\mathrm{P}(\text{data}|\theta)}{\underset{\text{predictive}}{\mathrm{p}(\text{data})}}$$

3.3 Heuristic stopping rule

The mechanism of heuristic stopping rule is that there are finite minima on the range of function. They stop once a new minimum has not been found through V restart. The number of restarts allowed to find the next minimum depends on the convergence rate of the sample variance of different statistics, and usually increases with the number of minimum values found.

3.4 Dorea's stopping rule

This rule can estimate the probability that the function value of one local optimal is in the range of global optimal. And there are two stopping conditions: it stops at the same time the probability of the local optima is in the range of global optimal is high enough. Second, the iteration stops when there's not improvement on the last several steps (defined by user).
Among these 3 stopping rules Dorea's stopping rule usually stops below ten restarts while other stopping rule mentions passes 10 and one rule utilizing Bayesian stopping rule even uses 10^2 to reach global minima. The average of those different restart rule on finding minimum function value if shown by *Dick et al*. And it is coincidence with our research figure output. Taking the derivative of the averaged line in 8000 trials, the figure should also converge to zero as restart number approaches to infinity.



Discussion

Because of existed experiment statistic done by three researchers in Carnegie Mellon University, the code and graph are not further drowned. However, those restart mechanisms should be contained in code and output should be compared. Different stopping rule itself effect accuracy given by the model. As a research finding relation between restart number and model accuracy, stopping rule or restart algorithm cannot be more important. Fortunately, the average of serval restart algorithm gives the relation between restart number and their ability to find global optima. The selecting of epoch number also requires more advanced algorithms. The choice of loss function also matters a lot when computing accuracy of trained. Analysis on the effect on different loss function on accuracy should also be included on further research. Influences affects accuracy perform in systematic way which means the bad selection of loss function affect the performance of optimizer as well, in this way destruct whole neural network model. This research is very limited in probing relationship between restart number and accuracy. Only convolutional neural network with 2 different layers followed by same constructed pooling layer is covered, more neural network like DNN, simple multi-layer perceptron should also be considered. The multi-layer perceptron that follows CNN in our research to classify to one hot classis can also vary to see the relation with restarts and with accuracy.

Conclusion

A simple relation between restart number and accuracy is drawn: the more restart number we have, the more accurate on the trained expert, but with less improvement of the accuracy. I first constructed a simple version restart algorithm to output the graph. Then, the research on several different are analyzed and compared with my figure. These two are coincident with each other which testified my hypothesis. And perfect epoch number is selected using iteration through 100 epochs and see which one can gives highest test set prediction accuracy. Mode of 20 trials suggested 13 as epoch number.

Overall, this research support the existed essay on restart number and it's indirectly relation with expert accuracy in way finding out global optima. The research shows accuracy can be improved with those various restart algorithms.

Reference

T. Dick, E. Wong, C. Dann. {CMU}How many random restarts are enough? 2014. URL: https://www.cs.cmu.edu/~epxing/Class/10715-14f/project-reports/DannDickWong.pdf

E.J. Wagenmakers. {U of Amsterdam} Stopping Rules and Their Irrelevance for Bayesian Inference: Online Appendix to "A Practical Solution to the Pervasive Problems of p–Values", to appear in Psychonomic Bulletin & Review. 2007. URL: https://ejwagenmakers.com/2007/StoppingRuleAppendix.pdf

R. Kuc. {Yale}Machine Learning for Spoken Digit Classification.