How is the choice of the activation function used between layers impacting CIFAR-10 dataset training performance?

Machine Learning Practical

Muhamad Iqbal Rosiadi

Coursework 3
Machine Learning Practical (INF11119)

s1635468@ed.ac.uk

Introduction

During computation, Neural Network produces a linear combination from the input signals. In this case, the output from this computation is not differentiable and linear. Element-wise non-linear function is needed to make the output of computation to reduce the overfitting and improve generalization. There are several, usually it's called activation function, non-linear function available to complete this task, such as logistic sigmoid, hyperbolic tangent, rectified linear, and softplus. This paper compares impact of those three to the set performance. Each implementation of non-linear function was tested to several different depth of the affine layer networks.

Comparison between Activation Function

Every activation function has their own characteristic to be explored. Logistic Sigmoid pulling a value towards between 0 and 1. It depends on how far the threshold, whether above or below threshold. And also it allows for a gradient to be discovered. The other activation which is similar to sigmoid is Hyperbolic Tangent (Tanh) by makes activation saturated. But unlike sigmoid, it has a real-valued number to the range [-1,1] and its output is zero centered. Other activation in this experiment is Rectified Linier(ReLU). This function is the identity for positive arguments and zero otherwise. While sigmoid and Tanh suffer with saturated activation, ReLU does not suffer from saturating. But ReLU units can be fragile during training and can "die". If this happens, then the gradient flowing through the unit will forever be zero from that point on. That is, the ReLU units can irreversibly die during training since they can get knocked off the data manifold (2). The other activation in which similar to ReLU is softplus. This activation is more likely ReLU but with a smooth approximation.

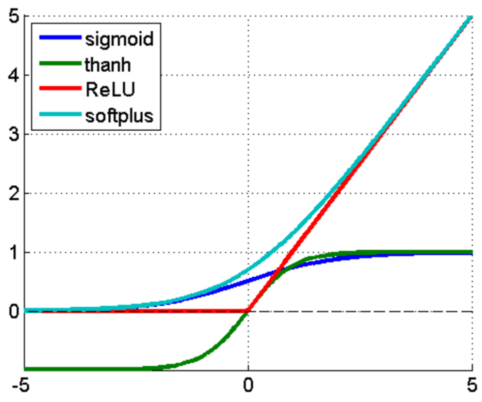


Figure 1 Comparison of Activation functions

Experiments Setting

In this experiment, we assess the performance of four activation function in feed-forward networks with multiple affine layers, comparing between (i) Sigmoid; (ii) Hyperbolic Tangent (Tanh); (iii) Rectified Linear Units (ReLUs); and (iv) Softplus. The following benchmark datasets was used CIFAR-10 (color images in 10 classes, 50k train and 10k test). Number of hidden unit is fixed 200. Adam Optimizer rule and Cross Entropy Softmax were used in this experiment. Learning rate value for this case was set to default from tensorflow.

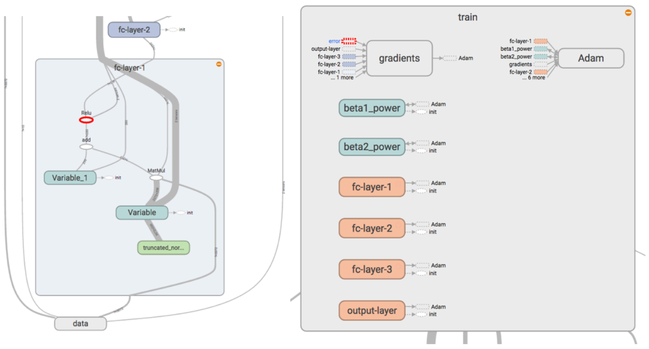


Figure 2 Implementation activation function on each layer and Adam optimizer for each model

The depth variation of affine layers are 1, 2, 3, 4, and 5 layers with different implementation of four non-linear function in each experiment.

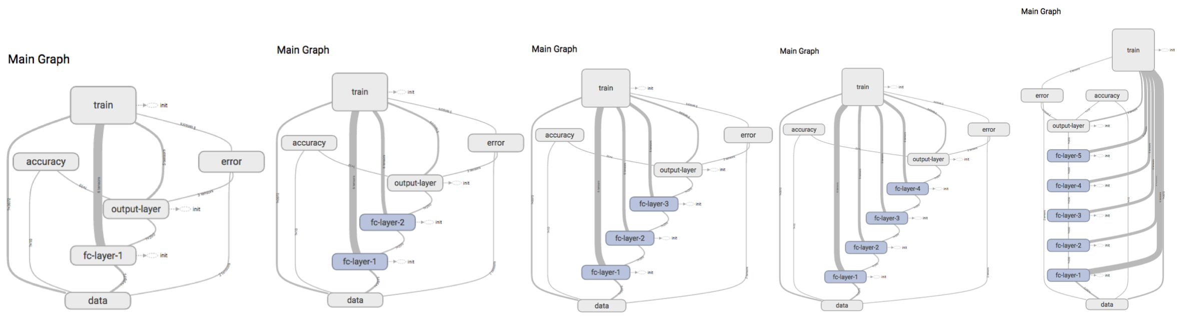
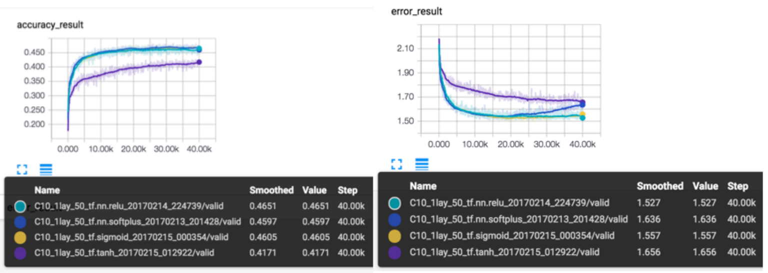


Figure 3 Architecture of multiple layer implementation

This setup is applied for all four non-linear function experiments and all implementation using Tensorflow framework.

Experiment Results on CIFAR10

One Affine Layer

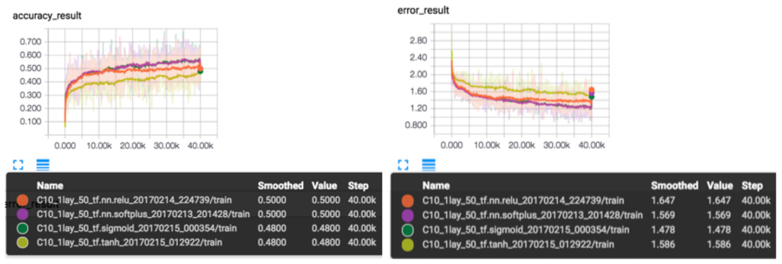


	Name	Smoothed	Value	Step
●	C10_1lay_50_tf.nn.relu_20170214_224739/valid	0.4651	0.4651	40.00k
●	C10_1lay_50_tf.nn.softplus_20170213_201428/valid	0.4597	0.4597	40.00k
●	C10_1lay_50_tf.sigmoid_20170215_000354/valid	0.4605	0.4605	40.00k
●	C10_1lay_50_tf.tanh_20170215_012922/valid	0.4171	0.4171	40.00k

	Name	Smoothed	Value	Step
●	C10_1lay_50_tf.nn.relu_20170214_224739/valid	1.527	1.527	40.00k
●	C10_1lay_50_tf.nn.softplus_20170213_201428/valid	1.636	1.636	40.00k
●	C10_1lay_50_tf.sigmoid_20170215_000354/valid	1.557	1.557	40.00k
●	C10_1lay_50_tf.tanh_20170215_012922/valid	1.656	1.656	40.00k

Figure 4 Validation data result comparison of activation function for one layer

Based on validation data result, only tanh produced the smallest value for accuracy and error rate. And also softplus value on error rate starts to decrease on epoch 25. If we compare figure 4 with figure 5, we will see all model are seem overfitting. In this model, Relu has the best accuracy at 0.4651 with the low level of error rate at 1.527.

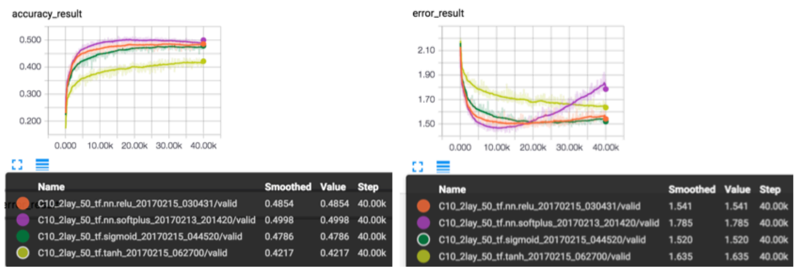


	Name	Smoothed	Value	Step
●	C10_1lay_50_tf.nn.relu_20170214_224739/train	0.5000	0.5000	40.00k
●	C10_1lay_50_tf.nn.softplus_20170213_201428/train	0.5000	0.5000	40.00k
●	C10_1lay_50_tf.sigmoid_20170215_000354/train	0.4800	0.4800	40.00k
●	C10_1lay_50_tf.tanh_20170215_012922/train	0.4800	0.4800	40.00k

	Name	Smoothed	Value	Step
●	C10_1lay_50_tf.nn.relu_20170214_224739/train	1.647	1.647	40.00k
●	C10_1lay_50_tf.nn.softplus_20170213_201428/train	1.569	1.569	40.00k
●	C10_1lay_50_tf.sigmoid_20170215_000354/train	1.478	1.478	40.00k
●	C10_1lay_50_tf.tanh_20170215_012922/train	1.586	1.586	40.00k

Figure 5 Train data result comparison of activation function for one layers

Two Affine Layer

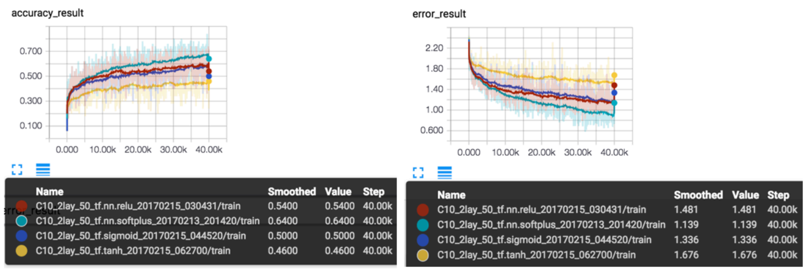


	Name	Smoothed	Value	Step
●	C10_2lay_50_tf.nn.relu_20170215_030431/valid	0.4854	0.4854	40.00k
●	C10_2lay_50_tf.nn.softplus_20170213_201420/valid	0.4998	0.4998	40.00k
●	C10_2lay_50_tf.sigmoid_20170215_044520/valid	0.4786	0.4786	40.00k
●	C10_2lay_50_tf.tanh_20170215_062700/valid	0.4217	0.4217	40.00k

	Name	Smoothed	Value	Step
●	C10_2lay_50_tf.nn.relu_20170215_030431/valid	1.541	1.541	40.00k
●	C10_2lay_50_tf.nn.softplus_20170213_201420/valid	1.785	1.785	40.00k
●	C10_2lay_50_tf.sigmoid_20170215_044520/valid	1.520	1.520	40.00k
●	C10_2lay_50_tf.tanh_20170215_062700/valid	1.635	1.635	40.00k

Figure 6 Validation data result comparison of activation function for two layer

As it can be seen on figure 6, tanh is still the least performance while other functions could push their value above 0.45. While softplus seems so promising, but its error rate starts to decreased on epoch 15 . On this model, relu and sigmoid are slightly same on the consistency of performance until the end of the 50 epoch. Also in this model performances are overfitting. In this model, softplus has the best accuracy at 0.4998 but with the high level of error rate at 1.785.

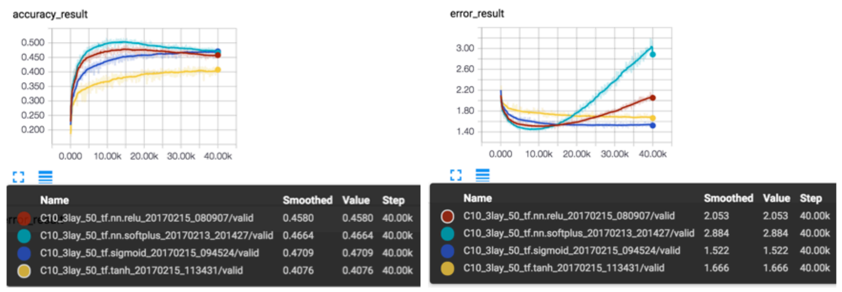


	Name	Smoothed	Value	Step
●	C10_2lay_50_tf.nn.relu_20170215_030431/train	0.5400	0.5400	40.00k
●	C10_2lay_50_tf.nn.softplus_20170213_201420/train	0.6400	0.6400	40.00k
●	C10_2lay_50_tf.sigmoid_20170215_044520/train	0.5000	0.5000	40.00k
●	C10_2lay_50_tf.tanh_20170215_062700/train	0.4600	0.4600	40.00k

	Name	Smoothed	Value	Step
●	C10_2lay_50_tf.nn.relu_20170215_030431/train	1.481	1.481	40.00k
●	C10_2lay_50_tf.nn.softplus_20170213_201420/train	1.139	1.139	40.00k
●	C10_2lay_50_tf.sigmoid_20170215_044520/train	1.336	1.336	40.00k
●	C10_2lay_50_tf.tanh_20170215_062700/train	1.676	1.676	40.00k

Figure 7 Train data result comparison of activation function for two layer

Three Affine Layer

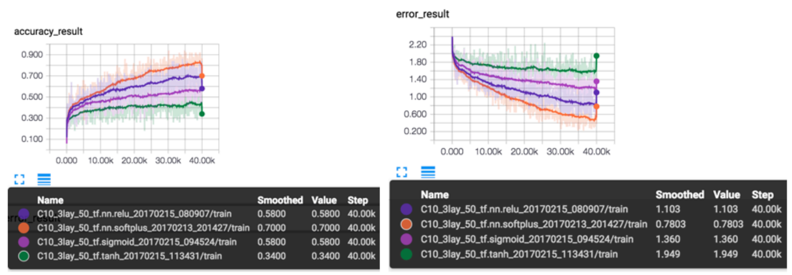


	Name	Smoothed	Value	Step
●	C10_3lay_50_tf.nn.relu_20170215_080907/valid	0.4580	0.4580	40.00k
●	C10_3lay_50_tf.nn.softplus_20170213_201427/valid	0.4664	0.4664	40.00k
●	C10_3lay_50_tf.sigmoid_20170215_094524/valid	0.4709	0.4709	40.00k
●	C10_3lay_50_tf.tanh_20170215_113431/valid	0.4076	0.4076	40.00k

	Name	Smoothed	Value	Step
●	C10_3lay_50_tf.nn.relu_20170215_080907/valid	2.053	2.053	40.00k
●	C10_3lay_50_tf.nn.softplus_20170213_201427/valid	2.884	2.884	40.00k
●	C10_3lay_50_tf.sigmoid_20170215_094524/valid	1.522	1.522	40.00k
●	C10_3lay_50_tf.tanh_20170215_113431/valid	1.666	1.666	40.00k

Figure 8 Validation data result comparison of activation function for three layer

From figure 8, we can see softplus value on error rate performance decrease sharply if its compared to figure 6. And also ReLUs has similar behavior with softplus. Its error rate performance starts to decrease on epoch 20, while Tanh and Sigmoid are steady. The decrease of ReLU is not significant compare to softplus, though. On other hand, Sigmoid has the best accuracy compared to other function with stable error rate until the end of epoch 50. In this model, sigmoid has the best accuracy at 0.4709 with the low level of error rate at 1.522.

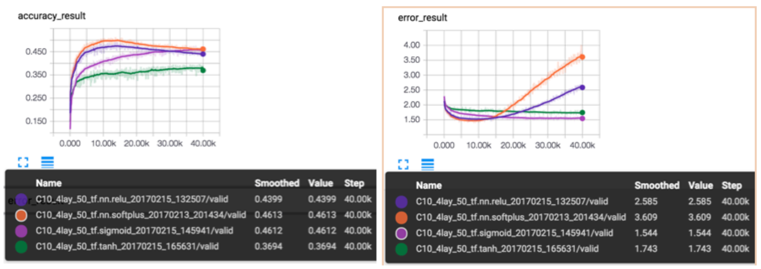


	Name	Smoothed	Value	Step
●	C10_3lay_50_tf.nn.relu_20170215_080907/train	0.5800	0.5800	40.00k
●	C10_3lay_50_tf.nn.softplus_20170213_201427/train	0.7000	0.7000	40.00k
●	C10_3lay_50_tf.sigmoid_20170215_094524/train	0.5800	0.5800	40.00k
●	C10_3lay_50_tf.tanh_20170215_113431/train	0.3400	0.3400	40.00k

	Name	Smoothed	Value	Step
●	C10_3lay_50_tf.nn.relu_20170215_080907/train	1.103	1.103	40.00k
●	C10_3lay_50_tf.nn.softplus_20170213_201427/train	0.7803	0.7803	40.00k
●	C10_3lay_50_tf.sigmoid_20170215_094524/train	1.360	1.360	40.00k
●	C10_3lay_50_tf.tanh_20170215_113431/train	1.949	1.949	40.00k

Figure 9 Train data result comparison of activation function for three layer

Four Affine Layer

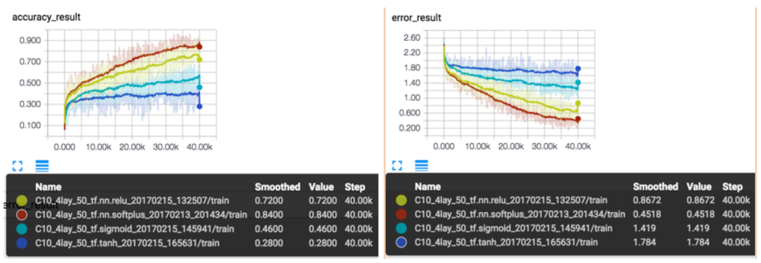


	Name	Smoothed	Value	Step
●	C10_4lay_50_tf.nn.relu_20170215_132507/valid	0.4399	0.4399	40.00k
●	C10_4lay_50_tf.nn.softplus_20170213_201434/valid	0.4613	0.4613	40.00k
●	C10_4lay_50_tf.sigmoid_20170215_145941/valid	0.4612	0.4612	40.00k
●	C10_4lay_50_tf.tanh_20170215_165631/valid	0.3694	0.3694	40.00k

	Name	Smoothed	Value	Step
●	C10_4lay_50_tf.nn.relu_20170215_132507/valid	2.585	2.585	40.00k
●	C10_4lay_50_tf.nn.softplus_20170213_201434/valid	3.609	3.609	40.00k
●	C10_4lay_50_tf.sigmoid_20170215_145941/valid	1.544	1.544	40.00k
●	C10_4lay_50_tf.tanh_20170215_165631/valid	1.743	1.743	40.00k

Figure 10 Validation data result comparison of activation function for four layer

Model with four layer are become more complex with depth dimension of layer. It might affect the performance of each function. As it can be seen on figure 10, sigmoid has positive trend compared to relu and softplus. While sigmoid performance on error rate is remain stable, softplus and relu error rate performance are significantly decreased. In this model, softplus has the best accuracy with 0.4613 but still with the lowest performance in error rate at 3.609. the second place is sigmoid, it has slight different accuracy performance with softplus but with stable and better performance in error rate with 1.544.

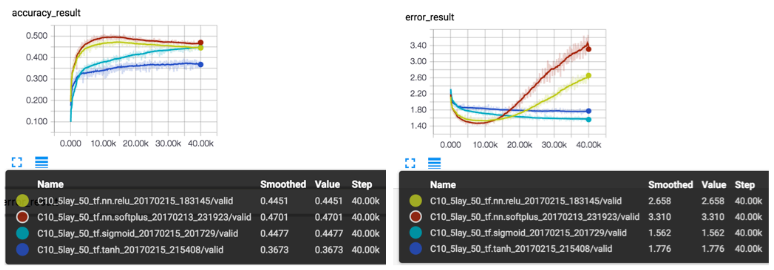


	Name	Smoothed	Value	Step
●	C10_4lay_50_tf.nn.relu_20170215_132507/train	0.7200	0.7200	40.00k
●	C10_4lay_50_tf.nn.softplus_20170213_201434/train	0.8400	0.8400	40.00k
●	C10_4lay_50_tf.sigmoid_20170215_145941/train	0.4600	0.4600	40.00k
●	C10_4lay_50_tf.tanh_20170215_165631/train	0.2800	0.2800	40.00k

	Name	Smoothed	Value	Step
●	C10_4lay_50_tf.nn.relu_20170215_132507/train	0.8672	0.8672	40.00k
●	C10_4lay_50_tf.nn.softplus_20170213_201434/train	0.4518	0.4518	40.00k
●	C10_4lay_50_tf.sigmoid_20170215_145941/train	1.419	1.419	40.00k
●	C10_4lay_50_tf.tanh_20170215_165631/train	1.784	1.784	40.00k

Figure 11 Train data result comparison of activation function for four layer

Five Affine Layer

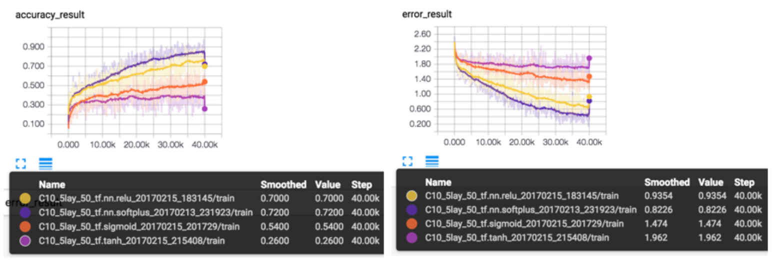


	Name	Smoothed	Value	Step
●	C10_5lay_50_tf.nn.relu_20170215_183145/valid	0.4451	0.4451	40.00k
●	C10_5lay_50_tf.nn.softplus_20170213_231923/valid	0.4701	0.4701	40.00k
●	C10_5lay_50_tf.sigmoid_20170215_201729/valid	0.4477	0.4477	40.00k
●	C10_5lay_50_tf.tanh_20170215_215408/valid	0.3673	0.3673	40.00k

	Name	Smoothed	Value	Step
●	C10_5lay_50_tf.nn.relu_20170215_183145/valid	2.658	2.658	40.00k
●	C10_5lay_50_tf.nn.softplus_20170213_231923/valid	3.310	3.310	40.00k
●	C10_5lay_50_tf.sigmoid_20170215_201729/valid	1.562	1.562	40.00k
●	C10_5lay_50_tf.tanh_20170215_215408/valid	1.776	1.776	40.00k

Figure 12 Validation data result comparison of activation function for five layer

Based on figure 12, softplus and relu behavior almost same. It is obvious because softplus is "smooth" version of relu. While sigmoid and tanh seem stable on both accuracy and error rate performance, relu and softplus performance on error rate are decreased sharply on epoch 15. In this model, softplus is the best model with 0.4701 but it has very low level of error rate performance with 3.310. Still, the fair model is sigmoid with 0.4477 on accuracy performance and 1.562 on error rate which is the lowest value of error rate.



	Name	Smoothed	Value	Step
●	C10_5lay_50_tf.nn.relu_20170215_183145/train	0.7000	0.7000	40.00k
●	C10_5lay_50_tf.nn.softplus_20170213_231923/train	0.7200	0.7200	40.00k
●	C10_5lay_50_tf.sigmoid_20170215_201729/train	0.5400	0.5400	40.00k
●	C10_5lay_50_tf.tanh_20170215_215408/train	0.2600	0.2600	40.00k

	Name	Smoothed	Value	Step
●	C10_5lay_50_tf.nn.relu_20170215_183145/train	0.9354	0.9354	40.00k
●	C10_5lay_50_tf.nn.softplus_20170213_231923/train	0.8226	0.8226	40.00k
●	C10_5lay_50_tf.sigmoid_20170215_201729/train	1.474	1.474	40.00k
●	C10_5lay_50_tf.tanh_20170215_215408/train	1.962	1.962	40.00k

Figure 13 Train data result comparison of activation function for five layer

Conclusion

From overall validation data result, we can see that the number of layers and type activation function has different impact on the performance. More deep layer the model will be more complex. If we use ReLU and Softplus the units of the function can die in order to respond zero to anything and will give low performance on error rate. While Sigmoid might has better performance on deep feed-forward model in this setting of experiment, ReLU and Softplus would give better performance while it has less deep feed-forward.

While the final training set error performance increasing through the increased number of affine layers, the final validation set error performances seem tend to decrease. If we refer to all figure of train data, ReLU has the best performance during the training process, this makes wider gap between final training set and final validation set. Hence, ReLU has wider gap in overfitting if it is compared with Sigmoid and Tanh. On the other side, sigmoid has narrow gap between training and validation data. Even though, overfitting for each function occurs we might could overcome this issue using early stopping.

Further Work

The Author interesting to explore more about how convolution neural network will give better result on CIFAR dataset. The CNN for next CIFAR experiments will reconstruct experiment of Clevert on paper "FAST AND ACCURATE DEEP NETWORK LEARNING BY EXPONENTIAL LINEAR UNITS (ELUS)". So, the author will try to construct CNN of 11 convolutional layers arranged in stacks of ([1 × 192 × 5], [1 × 192 × 1, 1 × 240 × 3], [1 × 240 × 1, 1 × 260 × 2], [1 × 260 × 1, 1 × 280 × 2],[1 × 280 × 1,1 × 300 × 2],[1 × 300 × 1],[1 × 100 × 1]) layers × units × receptive fields. 2×2 max-pooling with a stride of 2 was applied after each stack. For network regularization we used the following drop-out rate for the last layer of each stack (0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.0). The L2-weight decay regularization term was set to 0.0005. The following learning rate schedule was applied (0 − 35k[0.01], 35k − 85k[0.005], 85k − 135k[0.0005], 135k − 165k[0.00005]) (iterations [learning rate]).

Bibliography

1. A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In NIPS, 2012
2. CS231n: Convolutional Neural Networks for Visual Recognition of Stanford University, held by Prof. Fei-Fei Li and Andrej Karpathy.
3. Dugas, Y. Bengio, F. Bélisle, C. Nadeau, R. Garcia. Incorporating Second-Order Functional Knowledge for Better Option Pricing. In NIPS, 2000
4. Clevert, D., Unterthiner, D., Hochreiter, S. Fast and Accurate Deep Network Learning by Exponential Linear Units(ELUS). In ICLR, 2016