Adaptive Signal Variances: CNN Initialization Through Modern Architectures

Poster No: 1187

Takahiko Henmi*, Esmeraldo Ronnie Rey Zara*, Yoshihiro Hirohashi†, Tsuyoshi Kato*

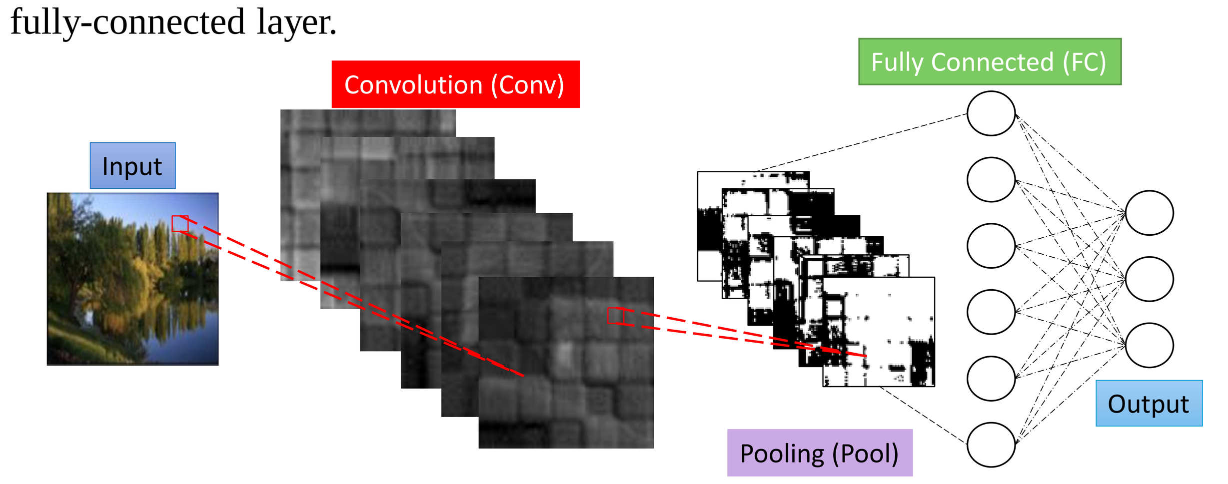
*Faculty of Science and Technology, Gunma University; †Individual

Objectives

1. Is it possible to derive an initialization method from a more precise CNN model?
2. How does the gradient descent work when using a more precise initialization model?

Convolutional Neural Networks (CNN)

The basic CNN architecture consists of a convolution layer, pooling layer and fully-connected layer.

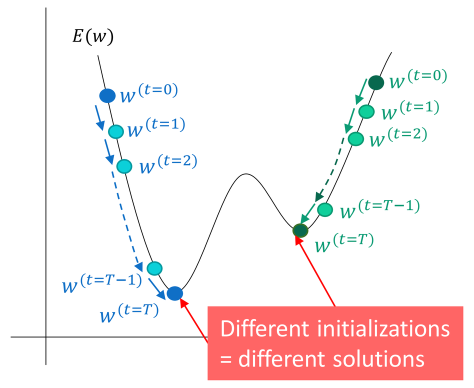


Learning CNN can be formulated as the problem:

$$\min_{w} E(w)$$ Non-convex problem

with weight w updates

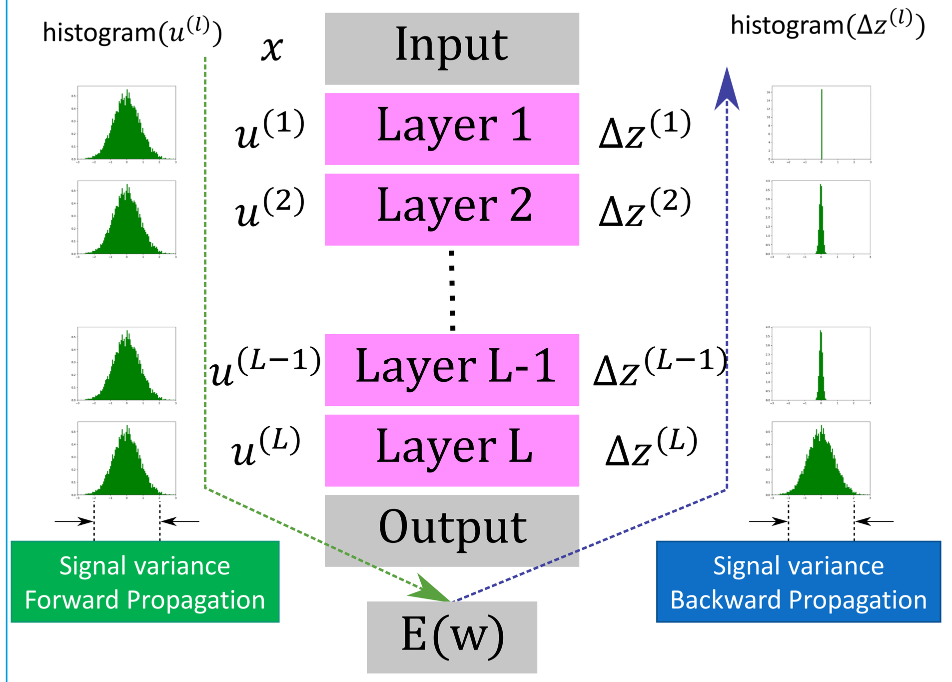
$$w^{(t=T+1)} = w^{(t=T)} - \eta \left.\frac{\partial E}{\partial w}\right|_{w=w^{(t=T)}}$$



Due to non-convexity, using the gradient method, the optimal solution will depend on the initial solution. This poses a challenge in learning CNNs.

$E(w)$ Loss function and error function; η learning rate

Initialization on CNN and Adaptive Signal Variances (ASV)



$$u_i^{(l)} = \left\langle w_{c(l,i),a(l,i)}^{(l)}, z_{s(l,i)}^{(l)} \right\rangle + b_{c(l,i)}^{(l)}$$

$$v_i^{(l)} = f_l\left(u_i^{(l)}\right),\ f_l\text{: activation function}$$

$$z_i^{(l)} = g_l\left(v_{t(l,i)}^{(l)}\right),\ g_l\text{: pooling function}$$

Classical Initialization Methods:

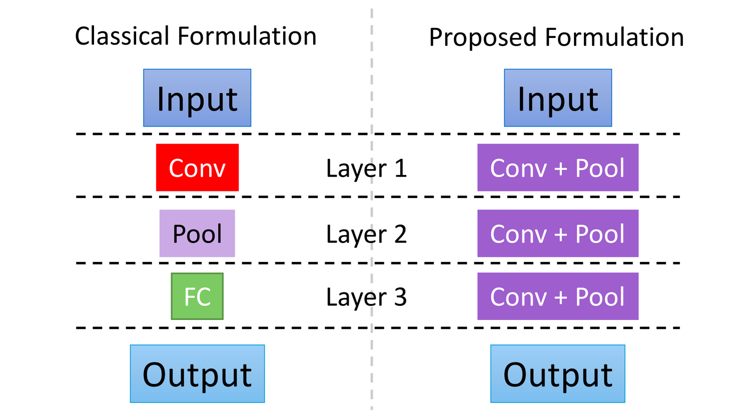
Xavier method: (Glorot et al, 2010):

$$\sigma_{w^{(l)}}^2 = \frac{2}{M_{l-1}+M_l}$$

Kaiming method: (He et al, 2015):

$$\sigma_{w^{(l)}}^2 = \frac{2}{S_l} \ \ or \ \ \sigma_{w^{(l)}}^2 = \frac{2}{J_l}$$

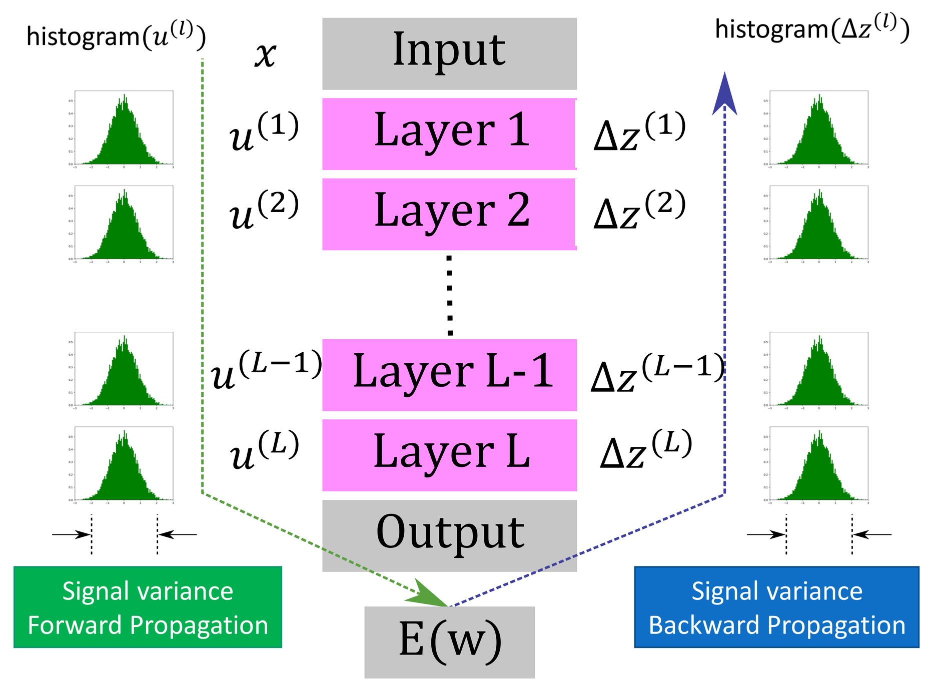
$b^{(l)}$: bias term
M_l: Size of signals in l layer
S_l: Receptive field size
J_l: Size of backward weights for convolution



Methods	FC	ReLU	Conv	Pooling
Xavier	✓			
Kaiming	✓	✓	✓	
Proposed	✓	✓	✓	✓

To support all the components in CNN, we formulated a new "layer" considering the convolution and pooling layer together.

In this new mathematical formulation, it is possible to express the convolution and fully-connected, but also pooling operation which has been ignored in the classical initialization methods.



For an input x and layer l, consider $u^{(l)}$ be the convolution output and $z^{(l)}$ be the pooling output with $\Delta z^{(l)} = \frac{\partial E}{\partial z^{(l)}}$.

Looking at the histogram on the forward propagation variance, the parameters are rarely updated in the first few layers resulting to **vanishing gradients**. On the other hand, if the variance becomes extremely large as it is backpropagated, this will result to the **exploding gradients** problem.

This poses another challenge in learning CNNs.

Initialization methods were introduced as way to solve these problems simultaneously.

Random number-based initialization method introduces an initialization parameter $\sigma_{w^{(l)}}^2$ where for each layer l, initial values are generated with zero-mean normal distributions with $\sigma_{w^{(l)}}^2$ as variance.

The Xavier and Kaiming methods are typically used but there are problems using these such as:

- Both ignore the pooling operation
- Insufficient theoretical background to support these methods

Adaptive Signal Variance (ASV)

ASV Forward Method:

$$\sigma_{w^{(\ell)}}^2 = \frac{M'_\ell}{\tau_{\ell-1}\varepsilon_\ell}$$

ASV Backward Method:

$$\sigma_{w^{(\ell)}}^2 = \frac{M_{\ell-1}}{\gamma_\ell\varepsilon_\ell}$$

The Adaptive Signal Variance (ASV) method preserves the variance of the forward and backward propagation.

With this new initialization method, the vanishing and exploding gradient problems are suppressed and since it is theoretically supporting the pooling and other operations of a CNN architecture, ASV initialization method is a theoretically supported initialization method for CNNs.

Details on the formulation can be found in our arXiv paper. Click on this link or scan the QR Code on the right.

Experimental Setting and Results

Below is the 34-layer architecture used for the experiment. Max Pooling is used in layer $l = 1$, Global Average Pooling is used in the last layer of the feature extraction $l = 33$

Component `InputBlock`(c)
Conv(7x7,channels=c,padding=3,stride=2), MaxPool(3x3,padding=1,stride=2)

Component `ConvBlock`($c, s = 1$)
Conv(3x3,channels=c,padding=1,stride=s), Conv(3x3,channels=c,padding=1,stride=1)

ℓ-th Layer	Output Shape	Model F34
	(3,224,224)	Input Image
1	(64,112,112)	`InputBlock`($c = 64$)
$2 \sim 7$	(64,56,56)	`ConvBlock`($c = 64$)×3
$8 \sim 15$	(128,28,28)	`ConvBlock`($c = 128$, $s = 2$) `ConvBlock`($c = 128$)×3
$16 \sim 27$	(256,14,14)	`ConvBlock`($c = 256$, $s = 2$) `ConvBlock`($c = 256$)×5
$28 \sim 33$	(512,7,7) (512,1,1)	`ConvBlock`($c = 512$, $s = 2$) `ConvBlock`($c = 512$)×2 Global Average Pooling
34	10	Linear
	2.11×10^7	#Parameters

The proposed initialization methods were tested on three subsets of public datasets:

- **Car** dataset from VMMRdb
- **Food** dataset from iFood 2019
- **Fungi** dataset from iNaturalist 2019

The 34-layer architecture was implemented with cross-entropy loss was used as the loss function and one thousand epochs of the gradient descent based on Adam with mini-batch size of 64. The learning rate was varied with four values $10^{-6}, 10^{-5}, 10^{-4}$ and 10^{-3}. ReLU is used for all layers except for the output layer.

The numbers in **bold face** represent the best performance for the entire table. Results on show that the proposed method, especially ASV backward, has a higher accuracy rate than the other methods for all the datasets.

For a 50-layer architecture, described in detail in the arXiv paper, the results on Tables 2 and 3 still show that the proposed method improved the performance in pattern recognition even for different optimization algorithms.

Table 1: Validation accuracies of 34-layer architecture with Adam

Datasets	Initialization Methods				
	Xavier	Kaiming (forward)	Kaiming (backward)	ASV (forward)	ASV (backward)
Car	71.95	70.52	73.10	72.74	**81.49**
Food	72.83	75.72	69.75	76.49	**78.81**
Fungi	65.23	68.16	66.99	67.97	**69.73**

Table 2: Validation accuracies of 50-layer architecture on Car dataset with Adam

Xavier	Kaiming (forward)	Kaiming (backward)	ASV (forward)	ASV (backward)
69.23	69.87	71.23	69.58	**80.77**

Table 3: Validation accuracies of 50-layer architecture on Car dataset

	Xavier	Kaiming (forward)	Kaiming (backward)	ASV (forward)	ASV (backward)
SGD	15.78	35.15	31.78	**35.29**	24.39
Adadelta	15.78	38.52	39.53	42.04	**50.22**
Adagrad	27.19	36.37	34.79	**43.33**	35.58
RMSprop	28.98	41.75	42.40	**42.47**	35.80
Adam	31.56	45.05	48.06	44.76	**51.15**

Conclusion

We have presented a reformulation of the CNN structure to introduce our proposed initialization methods that take account all the components of CNN. Our proposed initialization method, not only support the CNN structure theoretically, it also significantly improves the recognition performance.

References

[1]Vinod Nair and Geoffrey E. Hinton, "Rectified linear units improve restricted boltzmann machines.," in ICML, Johannes Fürnkranz and Thorsten Joachims, Eds. 2010, pp. 807–814, Omnipress.

[2] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich, "Going deeper with convolutions," in 2015 IEEE CVPR June 2015, IEEE.

[3] Ross Girshick, "Fast r-CNN," in 2015 IEEE ICCV. Dec. 2015, IEEE.

[4] Xavier Glorot and Yoshua Bengio, "Understanding the difficulty of training deep feedforward neural networks," in 13th International Conference on Artificial Intelligence and Statistics. 2010, pp. 249–256, PMLR.

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in 2015 IEEE ICCV, 2015, pp. 1026–1034.

Acknowledgements

This research was supported by the Environment Research and Technology Development Fund JPMEERF20205006 of the Environmental Restoration and Conservation Agency of Japan and JSPS KAKENHI Grant Number 19K04661.