JDCS
디지털콘텐츠학회논문지
Journal of Digital Contents Society
Vol. 22, No. 10, pp. 1721-1731, Oct. 2021

Check for updates

Glorot과 He의 초기화를 이용한 Mobilenet 및 Resnet 모델 레이어의 형상도 및 경사도 분포 분석

보 호앙 트롱[1] · 배 지 훈[2] · 유 광 현[1] · 김 진 영[3*]
[1]전남대학교 ICT융합시스템공학과 박사과정
[2]전남대학교 ICT융합시스템공학과 석사과정
[3*]전남대학교 ICT융합시스템공학과 교수

Distribution Analysis of Feature Map and Gradients in Mobilenet and Resnet Model Layers using Glorot and He`s initialization

Vo Hoang Trong[1] · Ji-Hoon Bae[2] · Gwang-Hyun Yu[1] · Jin-Young Kim[3*]

[1]Ph.D Course, Department of ICT Convergence System Engineering, Chonnam National University, Gwangju, Republic of Korea
[2]Master's Course, Department of ICT Convergence System Engineering, Chonnam National University, Gwangju, Republic of Korea
[3*]Professor, Department of ICT Convergence System Engineering, Chonnam National University, Gwangju, Republic of Korea

[요 약]

합성곱 신경망 모델에서 가중치를 초기화하는 것은 모델 성능 향상에 필수적인 역할을 한다. 본 논문은 Mobilenet 및 Resnet 기반 잡초 분류를 위한 합성곱 신경망 모델에 Glorot 및 He 초기화 방법을 분석한다. 본 실험 결과 Mobilenet의 Pointwise와 Depthwise 컨벌루션이 초기 레이어의 특징 맵의 변화를 줄이고, Resnet의 단축경로 연결은 He 초기화를 사용할 때 로지스틱 분류 레이어에서의 값을 포화시킨다. Glorot 초기화 방법을 사용한 Mobilenet 및 Resnet 기반 잡초 분류 합성곱 신경망 모델의 정확도는 각각 0.9568과 0.9711이고, He 초기화 방법을 사용한 경우에는 각각 0.9568과 0.9711이다. 따라서 Glorot 초기화 방법을 사용할 경우 He 초기화 방법보다 합성곱 신경망 모델이 더 빠르게 학습되고 일반화에 좋다.

[Abstract]

Initializing the weights plays an essential role in a convolutional neural network model. This paper investigates how Glorot and He's initialization methods behave in Mobilenet and Resnet models on the weeds classification problem. Experiments show that pointwise and depthwise convolution in Mobilenet reduces the variance of feature maps from earlier layers. Using the He's method, shortcut connection in Resnet saturate values in logistic classify layer. The accuracy of Mobilenet and Resnet, using Glorot's method, are 0.9568 and 0.9711, respectively. While using He's method, we obtain 0.9471 using Mobilenet and 0.9645 using Resnet. Also, both models converge faster and better generalization using Glorot's method than using He's method.

색인어 : 가중치 초기화, Glorot 초기화, He 초기화, 컨볼루션 신경망 네트워크, 잡초 분류
Keyword : Weights initialization, Glorot initialization, He initialization, Convolutional neural network, Weeds classification

http://dx.doi.org/10.9728/dcs.2021.22.10.1721

Received 08 September 2021; **Revised** 19 October 2021
Accepted 19 October 2021

***Corresponding Author; Jin-Young Kim**

Tel: +82-062-530-1757
E-mail: beyondi@chonnam.ac.kr

Ⅰ. INTRODUCTION

Many researchers have used convolutional neural network (CNN) models and achieved state-of-the-art performances in many computer vision problems, such as object detection, classification, and retrieval. These models are either trained from scratch or used transfer learning to initialize weights in models. Transfer learning applies weights trained from a source domain as initial parameters to a target domain. In practice, many deep learning frameworks such as Tensorflow [1], Keras [2], or PyTorch [3] support CNN models trained on the ImageNet dataset for transfer learning purposes. As shown in [4], we can apply transfer learning to guide the model to converge quickly, as long as the information features in the target domain are similar to those in the source domain. Otherwise, training the model from scratch is a reasonable solution. But this type of training requires careful initialization of learnable weights so that the model can learn features efficiently.

From a classical shallow LeNet-5 model to the deep VGG, Mobilenet, and Resnet CNN models, in general, a CNN model architecture consists of many “stacks” convolutional layers, where the output feature maps of a layer are the input of other layers. A matrix multiplication operates the convolution operators between the feature map and filter (followed by an activation function, such as sigmoid or rectified linear unit (ReLU)). The distribution of learnable parameters in filters affects the distribution of values in the output feature maps. Therefore, large weights tend to saturate feature maps towards 0 and 1 using sigmoid or lead to computational overflow using ReLU. Conversely, small weights may make the distribution of feature maps tight around zero in deeper layers, leading to small gradients and harmful model performance.

Probabilistic analysis of initialization of weights has been studied to avoid saturation or small gradient problems. LeCun *et al*. [5] showed that initializing the weights by randomly drawn from a zero-mean distribution makes the distribution of output nodes in the artificial neural network (ANN) model approximate a standard deviation of 1. This unit standard deviation places the node value in the linear curve of the sigmoid function so that the model can converge efficiently. However, the sigmoid function reduces the model’s ability to learn features (due to its non-zero mean) and may induce important singular values in the Hessian matrix. Glorot and Yoshua [6] clarified this phenomenon by experimenting with a multi-layer perceptron with a sigmoid activation function. They found that the sigmoid activation values in all layers are saturated to 0. To solve this problem, they studied and proposed a probabilistic approach to initialized weights so that variances of these weights were similar across layers (we call it Glorot's method). This method helps the gradients propagate with similar variance from lower to upper layers so that the model may learn features efficiently. But their approach is based on symmetric activation functions such as sigmoid, hyperbolic tangent, and softsign, which may be unsuitable for ReLU. He *et al.* [7] followed the ideas of Glorot and Yoshua [6] but examined the rectifier nonlinearities functions such as ReLU. They proposed a weights initialization method that works for nonlinear rectifiers (He's method). Given a Resnet model with ReLU as the activation function, Glorot's method makes the model stall, especially on extremely deep models. In contrast, He's initialization can make the model well converged.

In this paper, we investigate and compare the quality of Glorot and He's methods on Mobilenet and Resnet models. We show weight distributions, feature map values, and backpropagation gradients in both models on the CNU weeds dataset. By visualizing the feature map distribution, we realize that pointwise and depthwise convolution in Mobilenet reduces the variance of feature maps in deeper layers. Using the He's method, the shortcut connection in Resnet causes the last layer to saturate, and both models converge slower and generalize worse than using the Glorot's method. In experiments, Mobilenet and Resnet have an accuracy of 0.9568 and 0.9711 using Glorot's method; and 0.9471 and 0.9645 using He's method.

Ⅱ. CNU WEEDS DATASET

We used the CNU weeds dataset for experiments. This dataset was constructed by the Department of Electronics and Computer Eng, CNU, Gwangju, under the supervision of Rural Development Administration, Republic of Korea. Weeds images were captured and collected by Korean plant taxonomists, whose majors are botany. They collected weeds images on farms, fields in the Republic of Korea by using high-definition resolution cameras. Those species were classified and grouped by the international standard of plant taxonomy. This dataset has 21 species; it has 208,477 images in total. As shown in Figure 2, the CNU weeds dataset is an imbalanced dataset, where the species that has the largest number of images is *Galinsoga quadriradiata* Ruiz & Pav., around 24300 images. In contrast, species *Bidens bipinnata* L. has the smallest number of images, about 800 images.

Figure 1 shows an example of images on the CNU weeds dataset. From an agricultural perspective, weeds belonging to a particular family share some common characteristics in their structure, making their morphology quite similar even though they belong to two different species. On the other hand, weeds of different species are entirely different in appearance.

그림 1. CNU 잡초 데이터 세트에 있는 21종의 이미지 예제.
Fig. 1. Image examples of 21 species in the CNU weeds dataset.

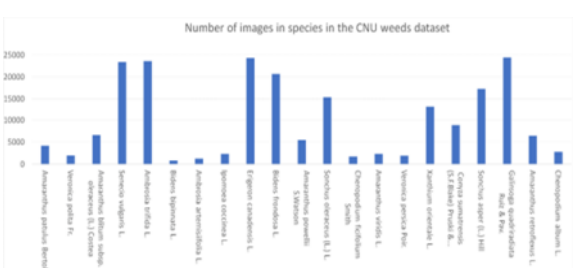

그림 2. CNU 잡초 데이터 세트의 종 분포.
Fig. 2. Distribution of species in the CNU weeds dataset.

Ⅲ. CNNs

3-1 Mobilenet

Howard *et al.* [8] developed a CNN model capable of deploying in low-memory devices called Mobilenet. This model has two core layers: Depthwise and pointwise convolution.

In standard convolution, current feature maps are formed by a multiplication and summation between previous feature maps and current filters. However, many filters lead to a huge number of learnable weights. To reduce the number of weights, Mobilenet split this convolution into two types of convolutions: First, a depthwise convolution applies a single convolutional filter for each channel in the feature maps. Mathematically, we can formulate depthwise convolution as

$$\hat{G}_{k,l,m} = \sum_{i,j} \hat{K}_{i,j,m} \bullet F_{k+i-1,l+j-1,m} \quad (1)$$

where $\hat{K}$ is the depthwise convolutional filter of size $D_k \times D_k \times M$. The m^{th} filter in $\hat{K}$ is applied to the m^{th} channel in F.

Then, a pointwise convolution (or 1×1 convolution) is applied to aggregate information in the feature maps channel-wise. A 3×3 depthwise convolutional filter in Mobilenet reduces 8 to 9 times computation than standard convolution.

In experiment, Howard *et al.* [8] showed that Mobilenet got a high accuracy on fine-grained object classification. So Mobilenet is sufficient for the weeds classification problem.

3-2 Resnet

Typically, CNN architecture models have “stacked” layers: One layer is placed on top of each layer. This type of deep architecture results in vanishing or exploding gradient flows from the bottom to the top layers, thus preventing the model from converging. He *et al.* [9] presented a residual learning framework to conveniently train an extremely deep model. Mathematically, a building block of the Resnet model is formulated as

$$y = F(x, (W_i)) + x \quad (2)$$

where x and y are the input and output feature maps of residual block, F is the residual mapping. In practice, this function is a stack of convolutional layers, each layers contain filters, and each i-th filter has a weights matrix W_i. The summation operation is performed by an element-wise addition of the input feature maps x to the stacked layers and the output feature maps of these stacked layers.

Resnet contains many building blocks. As explained in [10], implementing Eq. (2) recursively, we have

$$x_{l+2} = x_{l+1} + F(x_{l+1}, W_{l+1}) \quad (3)$$

$$= x_l + F(x_l, W_l) + F(x_{l+1}, W_{l+1}) \quad (4)$$

$$= x_l + \sum_{i=l}^{L-1} F(x_i, W_i) \quad (5)$$

for any shallow layer l. Assign Eq. (5) as x_L and denote the loss function as ϵ, the gradient of ϵ with respect to the feature map x_L is

$$\frac{\delta\epsilon}{\delta x_l} = \frac{\delta\epsilon}{\delta x_L}\frac{\delta x_L}{\delta x_l} \quad (6)$$

$$= \frac{\delta\epsilon}{\delta x_L}(1 + \frac{\delta}{\delta x_l}\sum_{i=l}^{L-1} F(x_i, W_i)) \quad (7)$$

$\delta\epsilon/\delta x_L$ is unlikely to be zero, so the gradient does not get vanish. With this building blocks, Resnet contains many deep layers and avoid making the gradient vanish or explode.

Ⅳ. WEIGHT INITIALIZATION

4-1 Glorot

Glorot and Bengio [6] analyzed layers in an ANN model by estimating the distribution of activation values (output of symmetry, nonlinear activation function) and gradients. They initialized weights followed by a random distribution and used the standard stochastic gradient descent (SGD) to optimize weighs in this model. They found that weights distribution in deeper layers saturate toward 0. With the sigmoid activation function, this behavior prevents the gradients from flowing backward so that deeper layers may not learn valuable features. This phenomenon also occurs for softsign and hyperbolic tangent but is less extreme for sigmoid. In common, SGD from random initialization is inefficient for a deep neural network.

To keep the information flow from saturating, they proposed a new initialization method to preserve the variance of output activation values between layers.

$$\forall\ (i,i'),\ Var[z^i] = Var[z^{i'}] \quad (8)$$

where $z^i, z^{i'}$ is the activation outputs of layer i and i' respectively. Finally Eq. (8) led to

$$\forall\ i,\ Var[z^i] = Var[x] \quad (9)$$

where x was the network input.

Assume we use a symmetry, zero-centered activation function with unit derivative at 0. Using 1st order Taylor function, we can approximate as a linear function as Eq. (10)

$$f(x) \approx f(0) + \frac{f'(0)}{1!}(x-0) = x \quad (10)$$

In common, at layer i, ANN models calculate a linear combination s^i between weights W^i and output of the activation function z^i, and the bias term b^i is added. That is, $s^i = z^i W^i + b^i$, and $z^{i+1} = f(s^i)$. Assume weights are initialized independently. To obtain similar variance, we calculate

$$Var[z^i] = Var[f(s^{i-1})] \quad (11)$$

$$= Var[z^{i-1}W^{i-1} + b^{i-1}] \quad (12)$$

$$= Var[z^{i-1}W^{i-1}] + Var[b^{i-1}] \quad (13)$$

Initializing the bias term as zero lead to $Var[b^{i-1}] = 0$, we have

$$Var[z^i] = Var[z^{i-1}W^{i-1}] \quad (14)$$

$$= Var[\sum_{k=0}^{n_{i-1}-1} z_k^{i-1} W_k^{i-1}] \quad (15)$$

$$= \sum_{k=0}^{n_{i-1}-1} Var[z_k^{i-1} w_k^{i-1}] \quad (16)$$

$$= \sum_{k=0}^{n_{i-1}-1} ([E(z_k^{i-1})]^2 Var(w_k^{i-1}) + [E(w_k^{i-1})]^2 Var(z_k^{i-1}) + Var(w_k^{i-1}) Var(z_k^{i-1})) \quad (17)$$

where is the number of nodes in layer. Assume we initialize weights by using a zero-mean distribution function, then

$$Var[z^i] = \sum_{k=0}^{n_{i-1}-1} Var(w_k^{i-1}) Var(z_k^{i-1}) \quad (18)$$

$$= n_{i-1} Var(W^{i-1}) Var(z^{i-1}) \quad (19)$$

Eq. (19) is a recursive formular. Expanding to all layers, we have

$$Var[z^i] = Var[x] \prod_{i'=0}^{i-1} n_{i'} Var[W^{i'}] \quad (20)$$

Setting $n_{i'}\,Var[\,W^{i'}]=1$ satisfy Eq. (8).

We also want similar gradient variance across layers, means that

$$Var[\frac{\delta Cost}{\delta s^{i}}]=\ Var[\frac{\delta Cost}{\delta s^{j}}] \tag{21}$$

where is a cost function of the ANN model. Using chain rule, we have

$$Var[\frac{\delta Cost}{\delta s^{i}}]$$

$$=\ Var[(\sum_{k-0}^{n_{i+1}} w_k^{i+1}\frac{\delta Cost}{\delta s_k^{i+1}})f'(s_j^i)] \tag{22}$$

$$=\sum_{k=0}^{n_{i+1}} Var(w_k^{i+1})\,Var(\frac{\delta Cost}{\delta s_k^{i+1}}) \tag{23}$$

$$=n_{i+1}\,Var(\,W^{i+1})\,Var(\frac{\delta Cost}{\delta s^{i+1}}) \tag{24}$$

Eq. (24) is also having a recursive form. Expanding to all layers, we have

$$Var[\frac{\delta Cost}{\delta s^{i}}]=\ Var[\frac{\delta Cost}{\delta s^{d}}]\prod_{i'=i}^{d} n_{i'+1}\,Var[\,W^{i'}] \tag{25}$$

Setting $n_{i'+1}\,Var[\,W^{i'}]=1$ satisfy Eq. (21). In total, for all, we have the following system of equation.

$$\left\{\begin{matrix} n_i\,Var[\,W^i]=1 \\ n_{i+1}\,Var[\,W^i]=1 \end{matrix}\right\} \tag{26}$$

$$\Leftrightarrow Var[w^i]=\frac{2}{n_i+n_{i+1}} \tag{27}$$

Eq. (27) show that the variance of weights in layer i depend on the number of nodes in that layer and the next layer. Assume we initialize weights followed by a uniform distribution, $W^i\sim\ U(-a,a)$, then

$$a=\sqrt{\frac{6}{n_i+n_{i+1}}} \tag{28}$$

4-2 He

Glorot's method is based on the assumption of using a zero-centered symmetric function as the activation function. However, this function is unsuitable for deep neural networks because of the gradient vanish problem during backpropagation [5]. Many popular CNN models such as VGG [11], Resnet [9], Mobilenet [8] used rectifier nonlinear activation functions because this function accelerates the training convergence [12] and more generalizes the model than symmetry function [13]. He *et al.* [7] investigated the variance condition of the output of rectifier nonlinear activation function at layer in a stacked CNN model, based on the procedure of Glorot's method.

$$y_l=\ W_l x_l+b_l \tag{29}$$

where x is $k^2c\times 1$ vector, c is the number of channels and k is the filter size. W is a $d\times n$ weight matrix containing d filters, $n=k^2c$ is the number of connections. Because we stack layers, the number of channels in current layer are equal to the number of filters used in the previous layer, $c_l=d_{l-1}$.

At the beginning, we set the bias term $b_l=0$ and $x_l=f(y_{l-1})$. Assume we initialize weights W_l followed by a zero-mean distribution ($E(\,W_l)=0$), then

$$Var(y_l)=\ Var(\,W_l x_l+b_l) \tag{30}$$

$$=\ Var(\sum_{i=1}^{n_l} w_l^i x_l^i) \tag{31}$$

$$=n_l(\,Var(\,W_l)\,Var(x_l)+(E(\,W_l))^2\,Var(x_l) \tag{32}$$
$$+\ Var(\,W_l)(E(x_l))^2)$$

$$=n_l\,Var(\,W_l)(\,Var(x_l)+(E(x_l))^2) \tag{33}$$

$$=n_l\,Var(\,W_l)E(x_l^2) \tag{34}$$

Using rectified linear unit function, we have $x=f(y)$ $=\max(0,y)$, so

$$E(x^2)=\int_{-\infty}^{+\infty} x^2P(x)dx \tag{35}$$

$$=\int_{-\infty}^{+\infty}\max(0,y)^2P(y)dy \tag{36}$$

$$=\int_{0}^{+\infty} y^2P(y)dy \tag{37}$$

Since y^2 is a symmetric function around 0 and the input y is zero-mean, we have

$$\int_{0}^{+\infty} y^2P(y)dy=\frac{1}{2}\int_{-\infty}^{+\infty} y^2P(y)dy \tag{38}$$

$$= \frac{1}{2}\int_{-\infty}^{+\infty}(y-E(y))^2 P(y)dy \quad (39)$$

$$= \frac{1}{2}E[(y-E(y))^2] \quad (40)$$

$$= \frac{1}{2}Var(y) \quad (41)$$

So we have

$$Var(y_l) = n_l Var(W_l)E(x_l^2) \quad (42)$$

$$= n_l Var(W_l)\frac{1}{2}Var(y_{l-1}) \quad (43)$$

Expand recursive Eq. (43) to all L layers, we have

$$Var(y_L) = Var(y_1)(\prod_{l=2}^{L}\frac{1}{2}n_l Var(w_l)) \quad (44)$$

Similar to Glorot's method, He's method maintains variance in the activation output across all layers by setting a condition for all layers,

$$\frac{1}{2}n_l Var(W_l) = 1 \quad (45)$$

Denote $\Delta x = \delta\epsilon/\delta x$, $\Delta y = \delta\epsilon/\delta y$, where Δx is a $c\times 1$ vector, represents gradient at a pixel of this layer, Δy represents $k\times k$ pixels in d channelks, and ϵ is an objective function. We have

$$y_l = W_l x_l + b_l \quad (46)$$

$$\Rightarrow \Delta x_l = \widehat{W}_l \Delta y_l \quad (47)$$

where $\widehat{W}_l$ is a $c\times\hat{n}$ matrix, and $\hat{n}= k^2 d$. Assume that W_l is initialized by a symmetric distribution around zero, then $E[\Delta x_l]$ for all l. By using ReLU as the activation function, $f'(y_l)$ = 1 or 0 with equal probability, so

$$E(\Delta y_l) = \frac{E(\Delta x_{l+1})}{2} = 0 \quad (48)$$

$$\Rightarrow E((\Delta y_l)^2) = Var(\Delta y_l) \quad (49)$$

$$= \frac{1}{2}Var(\Delta x_{l+1}) \quad (50)$$

So, calculate variance in Eq. (47), we have

$$Var(\Delta x_l) = \hat{n_l} Var(w_l) Var(\Delta y_l) \quad (51)$$

$$= \frac{1}{2}\hat{n_l} Var(w_l) Var(\Delta x_{l+1}) \quad (52)$$

Put L layers together, we have

$$Var(\Delta x_k) = Var(\Delta x_{L+1})(\prod_{l=k}^{L}\frac{1}{2}\hat{n_l} Var(w_l)) \quad (53)$$

To maintain gradient variance in backpropagation, we set $0.5\hat{n_l} Var(w_l) = 1$ for all layers l. He *et al.* [7] stated that using either n_l or $\hat{n_l}$ were sufficient for training the model. This condition aims to keep the gradient from being exponentially large or small.

He's method show that the weights in layer are initialized with a variance that depend only on the filter size in that layer. If we initialize $W_l \sim U(-a, a)$, then

$$a = \begin{bmatrix} -\sqrt{6/n_l} \\ \sqrt{6/\hat{n_l}} \end{bmatrix} \quad (54)$$

Mathematically, a filter of size n_l applied to layer l has a larger variance using He`s method than that of Glorot. By considering Eq. (28) and Eq. (54), Glorot requires an additional filter of size n_{l+1} in the next layer $l+1$, which increases the denominator value, hence smaller variance than He.

V. EXPERIMENTS

We used the CNU weeds dataset to train Mobilenet (88 layers) and Resnet model (50 layers). We trained on 100 epochs, using SGD with Nesterov momentum to optimize the cross-entropy loss function. The learning rate was initialized as , and decay to and at the 30th and 60th epoch. The batch size was 128 for Mobilenet and 32 for Resnet. We finetuned the model by adding a 256 dimension fully connected layer to avoid overfitting problems, followed by a batch normalization layer and ReLU activation function. We set the size of input color images to and normalized to using Mobilenet, and per-pixel mean subtraction using Resnet, as applied in [8] and [9].

We divided 60% of images in each species on the CNU Weeds dataset as training set, 20% as validation set, and the remaining 20% as test set. During model training, we selected the weights that had the highest accuracy on the validation set.

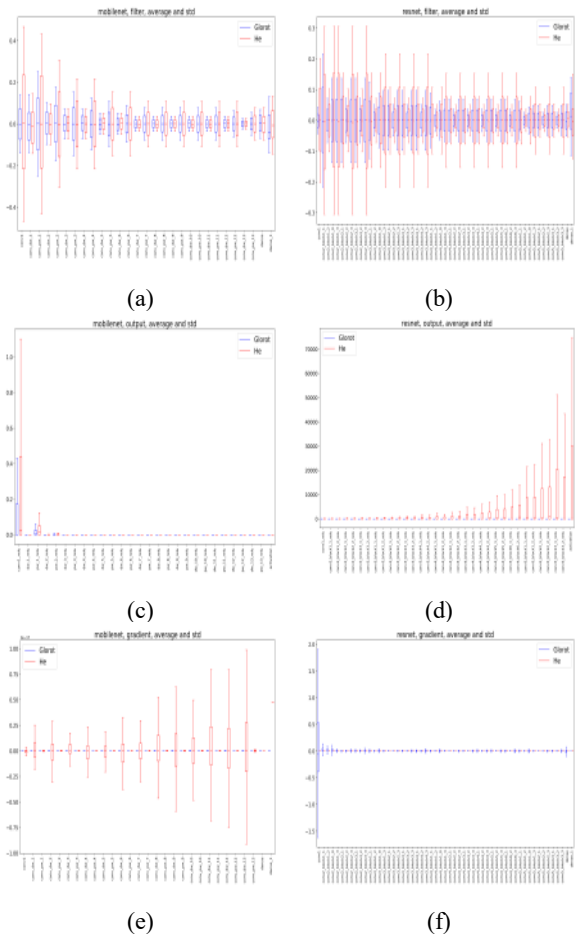

그림 3. Glorot 및 He의 방법을 사용하여 필터의 값, 활성화 레이어의 출력 및 여러 레이어의 그라데이션 상자 그림. 왼쪽 열: Mobilenet. 오른쪽 열: Resnet

Fig. 3. Box plot of values in filters, output of activation layers, and gradients across layers, using Glorot and He's methods. Left column: Mobilenet. Right column: Resnet

5-1 Weight Initialization

We initialized weights using a zero-mean Uniform distribution with variance followed by Glorot and He's methods. Figures (a) and (b) in Figure 3 show the box plot of values in filters, output of activation layers, and gradients between layers. It turns out that these values, using He's method, had a higher variance than Glorot's method because mathematically, He's method only needs the filter size in the current layer, while Glorot's method requires the current and next layer. Both Mobilenet and Resnet have large filter sizes at deeper layers, thus reducing its variance then. In particular, variances of the output of activation layers in Mobilenet decreased faster than those in Resnet, starting from the depthwise and pointwise convolution layers. The large filter size configuration caused this issue in the earlier layers in Mobilenet.

The first two rows of Figure 6 show the Mobilenet output activation values distribution, using Glorot and He's methods. The first layer in this model is a standard convolution. The left-most figures in 2 rows show that the output of this layer had a higher variance using He's method than Glorot's method. The last two figures of Figure 6 and figure (c) in Figure 3 show that, on average, using both initialization methods brought weights approaching zero. Both methods had a small variance when we went to deeper layers, mainly affected by two reasons: small input values after normalization ($[-1,1]$) and small variance of the zero-mean depthwise and pointwise convolution layers.

The last two rows of Figure 6 show the distribution of Resnet output activations. Like Mobilenet using Glorot's method, the variance decreased as we went to deeper layers, as shown in figure (d) of Figure 3. In contrast, using He's method increased the variance rapidly. In the last layer, many weights are larger than 10^5, resulting in the maximum value of softmax activation function saturated to 1. Thus, the gradient of all weights (after applying chain rule) was approximate zero. Those large weight values in deeper layers came from 2 reasons: For one layer, the distribution variance using He's method is larger than using Glorot's method, and the skip connection by a summation in Resnet architecture increases the output values in deeper layers.

The distribution of gradients in Mobilenet is shown in the first two rows in Figure 7, using Glorot and He's methods. Figure (e) in Figure 3 in both methods shows a lower variance in gradient in deeper layers for Glorot's method. In contrast, gradients have large values in deep layers for He's method, except for the values in pointwise convolution layers.

$$s(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \tag{55}$$

$$\frac{\delta s(z)_i}{\delta z_j} = s(z)_j(\delta_{ij} - s(z)_i) \tag{56}$$

The last two rows in Figure 7 show the distributions of gradient values in the Resnet model, using Glorot and He's methods. It shows that using Glorot's method, many weights had non-zero gradients. Except for the first layer, all layers had similar output variance, as shown in figure (f) of Figure 3. Using He's method to initialize weights in the Resnet model, all the gradients across layers were very close to zero. As explained above, large variance in He's method and shortcut connection led to enormous value in deeper layers. Therefore, $\max(s)$ in Eq. (55) (softmax function, K is the number of classes) was

approximately 1, and other elements in the softmax output vector were close to 0. In this case, the partial derivative in Eq. (56), where δ_{ij} is the Kronecker delta function, was approximately zero with respect to any z_j.

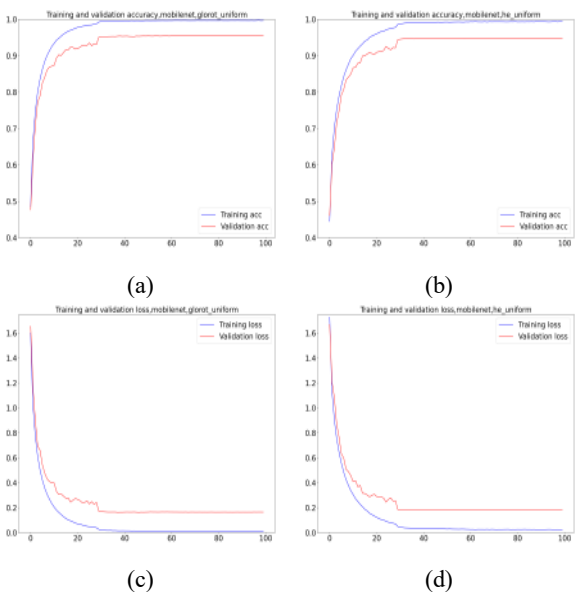

그림 4. Mobilenet의 훈련 및 검증 곡선. (a) Glorot 정확도. (b) He 정확도. (c) Glorot 손실 곡선. (d) He 손실 곡선.

Fig. 4. Training and validation curve on Mobilenet. (a) Accuracy, Glorot. (b) Accuracy, He. (c) Loss, Glorot. (d) Loss, He.

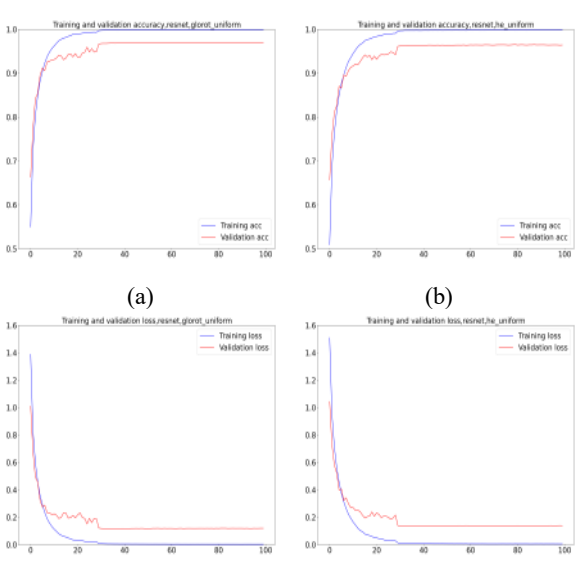

그림 5. Resnet의 교육 및 검증 곡선. (a) Glorot 정확도. (b) He 정확도. (c) Glorot 손실 곡선 (d) He 손실 곡선

Fig. 5. Training and validation curve on Resnet. (a) Accuracy, Glorot. (b) Accuracy, He. (c) Loss, Glorot. (d) Loss, He.

표 1. Glorot(Glo.)과 Mobilenet에서의 He(Mob.) 및 Resnet(Res.)을 사용한 초기화 방법의 성능. 네 가지 평가 지표: 정확도(Ac.) 정밀도(Pre.) 재현율(Rec.) F1 점수(F1) 및 손실.

Table 1. Performances of initialization methods, using Glorot (Glo.) and He on Mobilenet (Mob.) and Resnet (Res.). Four evaluation metrics: Accuracy (Acc.), precision (Pre.), recall (Rec.), F1 score (F1), and loss.

	Acc.	Pre.	Rec.	F1	Loss
Mob., Glo.	0.9568	0.9418	0.9413	0.9413	0.1541
Mob., He	0.9471	0.9347	0.9286	0.9313	0.1822
Res., Glo.	0.9711	0.9591	0.9614	0.9601	0.1119
Res., He	0.9645	0.9503	0.9517	0.9508	0.1356

5-2 Performances

Figure 4 shows performances of training the Mobilenet, where the weights were initialized using the Glorot and He's methods. It indicates that Glorot‘s method helped the model to converge faster than using He. By using Glorot's method, the model increased in accuracy quickly, faster than He‘s method. In both methods, the model converged at the 30th epoch.

Figure 5 shows performances of training the Resnet. Like what was in Mobilenet, Glorot‘s method helped the model converge a bit faster, starting in the 10th epoch. The model converged at 30th epoch in both methods.

Table 1 shows performances of models using Glorot and He’s methods as initialization methods. In particular, Glorot‘s method showed the ability to generalize the model better than He‘s method when all metrics using Glorot‘s method were 0.01 points higher than He‘s method, except for the loss value showing that Glorot‘s method was lower than He‘s method by nearly 0.03 points. Besides, Resnet had higher performance than Mobilenet, regardless of the initialization method used.

5-3 Analysis

Experiences have shown that, for a model, using Glorot's method to initialize weights makes the model converge faster and more general than He's method. The first few epochs in Figure 4 and Figure 5 indicate that, for the CNU weeds dataset, Glorot's method to initialize the weights placed these weights in appropriate positions on the loss surface in both models. This result is in contrast to [9], where Glorot's method (based on symmetric activation functions) is more suitable for training a weeds classification model than using He's method (based on rectifier nonlinear activation functions). Remind that both models use ReLU as the activation function. This problem occurs when the architectures of both models are not deep enough to show the advantage of He‘s method over Glorot‘s method.

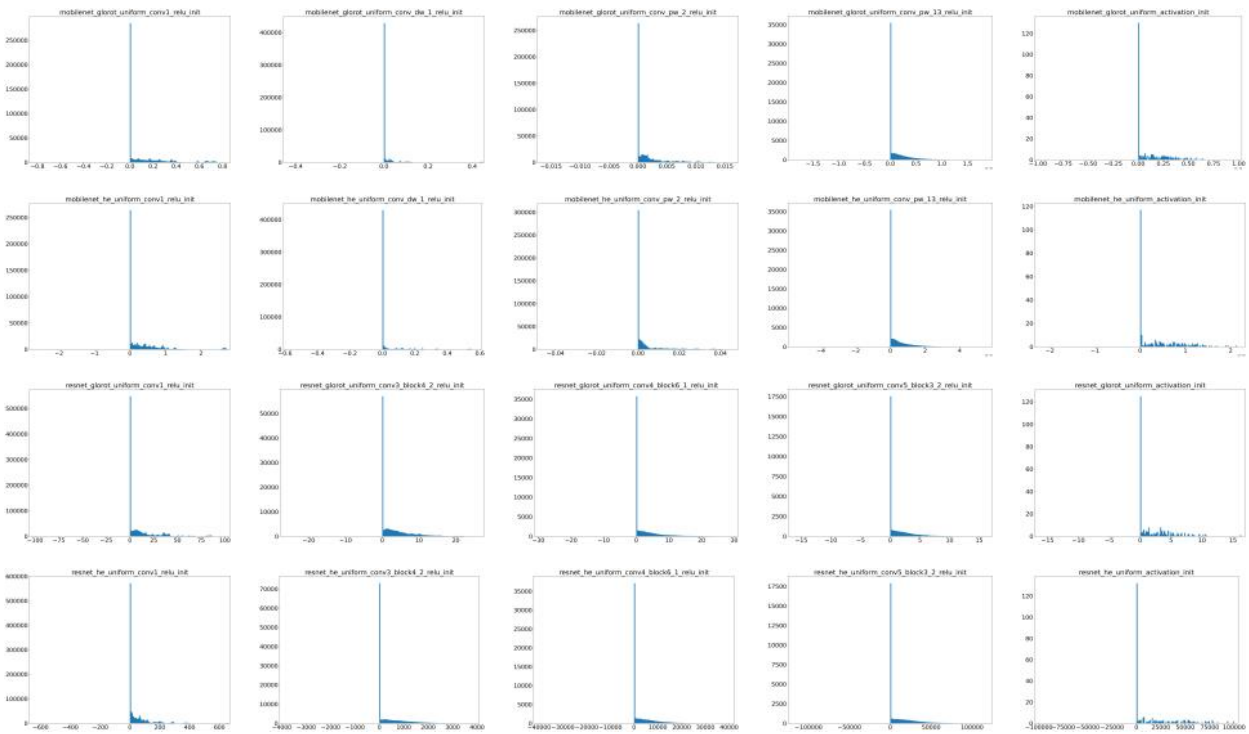

그림 6. 활성화 함수의 출력 분포. 위에서 아래로 1열과 2열은 Glorot과 He를 Mobilenet에서, 3열과 4열은 Glorot과 He를 Resnet에서 사용한 결과. 왼쪽에서 오른쪽으로는 가장 이른 계층에서 가장 낮은 계층까지의 결과.

Fig. 6. Distribution of output of activation function. From top to down, 1st and 2nd row use Glorot and He on Mobilenet, 3rd and 4th row use Glorot and He on Resnet. From left to right is the result from earliest to lowest layers.

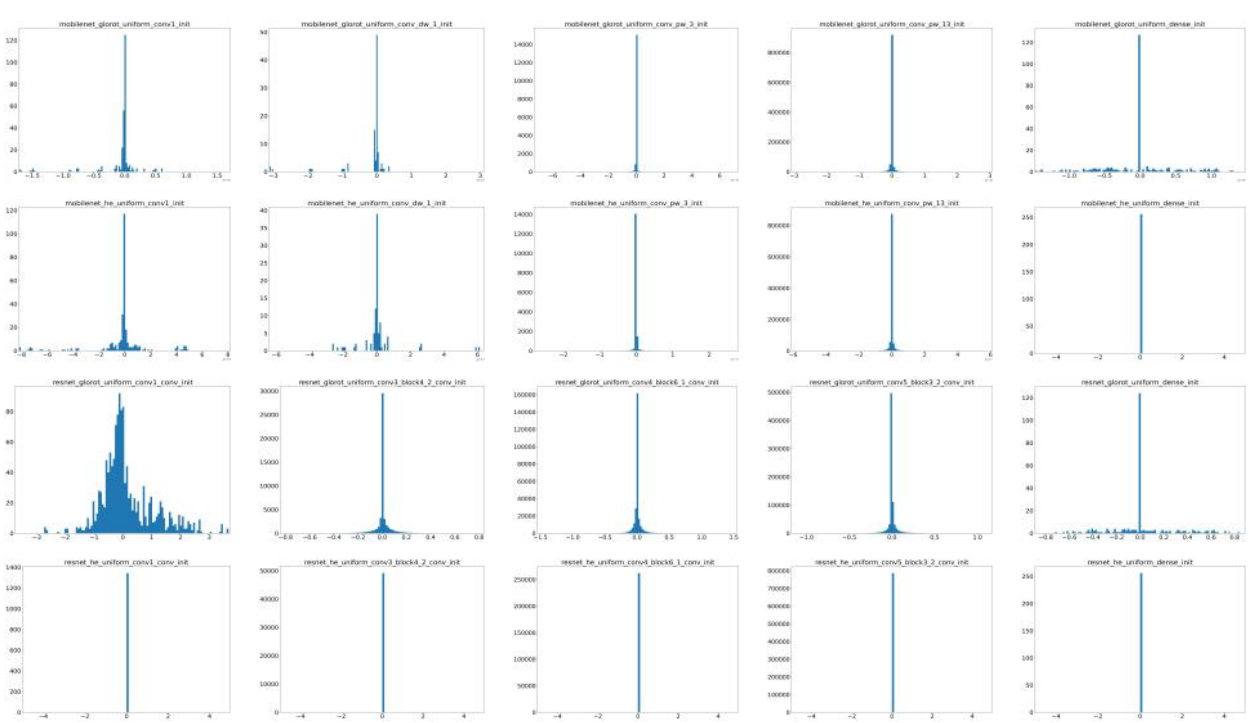

그림 7. 그라데이션의 출력 분포. 위에서 아래로 1열과 2열은 Glorot과 He를 Mobilenet에서, 3열과 4열은 Glorot과 He를 Resnet에서 사용한 결과. 왼쪽에서 오른쪽으로는 가장 이른 계층에서 가장 낮은 계층까지의 결과.

Fig. 7. Distribution of output of gradient. From top to down, 1st and 2nd row use Glorot and He on Mobilenet, 3rd and 4th row use Glorot and He on Resnet. From left to right is the result from earliest to lowest layers.

Glorot and He's methods is founded based on the assumption of stacking standard convolution layers such as VGG or conventional ANN model. However, both model architectures have its characteristic that may not fit that assumption. As shown in Figure 6, pointwise and depthwise convolution in Mobilenet reduces the variance of the output of activation function, resulting in the output in deeper layers close to zero. Shortcut connection in Resnet enlarged variance when using He's method, but not in Glorot's method because the variance in Glorot's method is smaller than He's method. This means that we might deal with large weights initialization, resulting in saturation in Glorot's method is smaller than He's method.

Ⅵ. CONCLUSION

In this paper, we compared and analyzed the performance of CNN models in the CNU Weeds dataset for weeds classification. We trained the Mobilenet and Resnet models from scratch, using Glorot and He's methods to initialize the weights. These authors studied these methods in conventional ANN and CNN models to maintain the variance of output of activation function across layers. But in practice, specific layers in Resnet and Mobilenet made these methods behave unexpectedly. Pointwise and depthwise layers in Mobilenet reduced the variance to near zero, and shortcut connection in He's method expanded the variance in deeper layers leading to saturation in softmax layer. Experiments show that using Glorot's method placed weights in both models in convenient positions so that models converged more quickly and generally than using He's method.

ACKNOWLEDGMENT

This work was carried out with the support of "Cooperative Research Program for Agriculture Science and Technology Development (Project No. PJ01385501)" Rural Development Administration, Republic of Korea.

REFERENCES

[1] TensorFlow, "Transfer learning and fine-tuning," TensorFlow, 17 06 2021. [Online]. Available: https://www.tensorflow.org/tutorials/images/transfer learning.

[2] F. Chollet, "Transfer learning & fine-tuning," Keras, 12 05 2020. [Online]. Available: https://keras.io/guides/transfer_learning/#transfer-learning-amp-finetuning.

[3] S. Chilamkurthy, "TRANSFER LEARNING FOR COMPUTER VISION TUTORIAL," Pytorch, [Online]. Available: https://pytorch.org/tutorials/beginner/transfer learning tutorial.html.

[4] B. Neyshabur, H. Sedghi and C. Zhang, "What is being transferred in transfer learning?," Advances in Neural Information Processing Systems, vol. 33, pp. 512-523, 2020.

[5] Y. LeCun, L. Bottou, G. B. Orr and K. R. Müller, "Efficient BackProp," Neural Networks: Tricks of the Trade, pp. 9-50, 1998. https://doi.org/10.1007/978-3-642-35289-8_3

[6] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in Proceedings of the 13th international conference on artificial intelligence and statistics, 2010.

[7] K. He, X. Zhang, S. Ren and J. Sun, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification," in 2015 IEEE International Conference on Computer Vision (ICCV), 2015. https://doi.org/10.1109/ICCV.2015.123

[8] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Wetand, ... and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," in arXiv preprint arXiv:1704.04861, 2017.

[9] K. He, X. Zhang, S. Ren and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016. https://doi.org/10.1109/CVPR.2016.90

[10] K. He, X. Zhang, S. Ren and J. Sun, "Identity mappings in deep residual networks," in European conference on computer vision, 2016. https://doi.org/10.1007/978-3-319-46493-0_38

[11] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for large-Scale Image Recognition," in 3rd International Conference on Learning Representations, San Diego, 2015.

[12] A. Krizhevsky, I. Sutskever and G. E. Hinton, "Imagenet classification with deep convolutional neural networs," in Advances in neural information processing systems, 2012. https://doi.org/10.1145/3065386

[13] X. Glorot, A. Bordes and Y. Bengio, "Deep sparse rectifier neural networks," in Proceedings of the 14th international conference on artificial intelligence and statistics, 2011.

보 호앙 트롱(Vo Hoang Trong)

2017년 : Department of Mathematics, Computer Science, Ho Chi Minh University of Science (공학학사)
2020년 : 전남대학교 전자공학과 (공학석사)

2020년～현 재: 전남대학교 ICT융합시스템공학과 박사과정
※관심분야 : 디지털 신호처리(Digital signal processing), 영상 처리(Image processing), 머신러닝(Machine learning), 딥러닝(Deep learning) 등

배지훈(Ji-Hoon Bae)

2021년 : 전남대학교 전자공학과 (공학학사)

2021년～현 재: 전남대학교 ICT융합시스템공학과 석사과정
※관심분야 : 영상 처리(Image processing), 머신러닝(Machine learning), 딥러닝(Deep learning) 등

유광현(Gwagn-Hyun Yu)

2018년 : 전남대학교 전자공학과 (공학석사)

2006년～현 재: Insectpedia 대표
2018년～현 재: 전남대학교 ICT융합시스템공학과 박사과정
※관심분야 : 디지털 신호처리(Digital signal processing), 영상 처리(Image processing), 머신러닝(Machine learning), 딥러닝(Deep learning) 등

김진영(Jin-Young Kim)

1986년 : 서울대학교 전자공학과 (공학학사)
1988년 : 서울대학교 전자공학과 (공학석사)
1994년 : 서울대학교 전자공학과 (공학박사)

1995년～현 재: 전남대학교 ICT융합시스템공학과 교수
※관심분야 : 디지털 신호처리(Digital signal processing), 영상 처리(Image processing), 음성 신호처리(Audio signal processing), 머신러닝(Machine learning), 딥러닝(Deep learning) 등