

정규화 및 항등사상이 활성화함수 성능에 미치는 영향

류서현*, 윤재복
국방기술품질원

The Effect of regularization and identity mapping on the performance of activation functions

Seo-Hyeon Ryu*, Jae-Bok Yoon

Defense Agency of Technology and Quality

요약 본 논문에서는 딥러닝에서 활용되는 정규화(regularization) 및 항등사상(identity mapping)이 활성화함수(activation function) 성능에 미치는 영향에 대해 설명한다. 딥러닝에서 활성화함수는 비선형 변환을 위해 사용된다. 초기에는 sigmoid 함수가 사용되었으며, 기울기가 사라지는 기존의 활성화함수의 문제점을 극복하기 위해 ReLU(Rectified Linear Unit), LReLU(Leaky ReLU), PReLU(Parametric ReLU), ELU(Exponential Linear Unit)이 개발되었다. 활성화함수와의 연구와는 별도로 과적합(Overfitting)문제를 해결하기 위해, Dropout, 배치 정규화(Batch normalization) 등의 정규화 방법들이 개발되었다. 추가적으로 과적합을 피하기 위해, 일반적으로 기계학습 분야에서 사용되는 data augmentation 기법이 활용된다. 딥러닝 구조의 측면에서는 기존에 단순히 컨볼루션(Convolution) 층을 쌓아올리는 구조에서 항등사상을 추가하여 순방향, 역방향의 신호흐름을 개선한 residual network가 개발되었다. 위에서 언급된 활성화함수들은 각기 서로 다른 특성을 가지고 있으나, 새로운 정규화 및 딥러닝 구조 연구에서는 가장 많이 사용되는 ReLU에 대해서만 검증되었다. 따라서 본 논문에서는 정규화 및 항등사상에 따른 활성화함수의 성능에 대해 실험적으로 분석하였다. 분석을 통해, 정규화 및 항등사상 유무에 따른 활성화함수 성능의 경향을 제시하였으며, 이는 활성화함수 선택을 위한 교차검증 횟수를 줄일 수 있을 것이다.

Abstract In this paper, we describe the effect of the regularization method and the network with identity mapping on the performance of the activation functions in deep convolutional neural networks. The activation functions act as nonlinear transformation. In early convolutional neural networks, a sigmoid function was used. To overcome the problem of the existing activation functions such as gradient vanishing, various activation functions were developed such as ReLU, Leaky ReLU, parametric ReLU, and ELU. To solve the overfitting problem, regularization methods such as dropout and batch normalization were developed on the sidelines of the activation functions. Additionally, data augmentation is usually applied to deep learning to avoid overfitting. The activation functions mentioned above have different characteristics, but the new regularization method and the network with identity mapping were validated only using ReLU. Therefore, we have experimentally shown the effect of the regularization method and the network with identity mapping on the performance of the activation functions. Through this analysis, we have presented the tendency of the performance of activation functions according to regularization and identity mapping. These results will reduce the number of training trials to find the best activation function.

Keywords : Activation function, Convolutional neural network, Deep learning, Identity mapping, Regularization

1. 서론

딥러닝은 영상분류, 얼굴인식, 물체인식 등 다양한 컴

퓨터비전 분야에서 탁월한 성능을 보이고 있다.

하지만 이러한 우수한 성능에도 불구하고 과적합(overfitting), 그래디언트 소실/폭발(gradient vanishing/

*Corresponding Author : Seo-Hyeon Ryu(Defense Agency for Technology and Quality

Tel: +82-54-469-6527 email: rsh87@dtq.re.kr

Received July 27, 2017

Revised (1st September 13, 2017, 2nd October 12, 2017)

Accepted October 13, 2017

Published October 31, 2017

exploding) 등의 문제를 가지고 있다. 이러한 문제를 해결하기 위해 많은 연구가 진행되어 왔다. 우선, 과적합 문제를 해결하기 위해, dropout[1], dropconnect[2]등이 제안되었다. dropout은 학습 간 임의로 노드를 0으로, dropconnect는 학습 간 임의로 파라미터를 0으로 만드는 것으로, 두 방법 모두 학습 간 서로 다른 모델이 학습되어, 모델 앙상블의 효과를 가져 온다. 그래디언트 소실/폭발 문제를 해결하기 위해, 배치 정규화(batch normalization)[3]연구되었다. 딥러닝에서 전체 데이터를 한 번에 최적화하는 것은 메모리 문제에 의해 힘들기 때문에, 기본적으로 미니배치(mini batch) 최적화를 수행한다. 이때 미니배치 구성이 매번 달라지므로, 각 미니배치의 분포가 다르게 되어 최적화 시 방해가 되는 요소가 된다. 이를 극복하기 위해, 배치정규화는 각 미니배치마다 평균이 0, 분산이 1이 되도록 정규화를 수행하여, 여러 미니배치간 분포의 차이를 최소화 시켰다. 또한, 분포를 정규화 시킴으로써, 더 큰 학습률 에도 최적화가 될 수 있도록 하여 학습시간을 단축시켰으며, 미니배치마다 다른 평균 및 분산을 가지므로, 학습 간 무작위 요소가 가미되어 과적합 문제를 해결하는 효과도 가지고 있다. 기존에는 주로 과적합을 피하기 위해, dropout이 사용되었지만, 최근 추세는 기본적으로 배치 정규화를 모든 컨볼루션층에 적용시키고, 추가적으로 정규화가 필요하다면 dropout방법을 적용시키고 있다.

딥러닝 구조적으로 그래디언트 소실/폭발을 해결한 방법은 residual network[4]이다. 이 구조는 단순히 컨볼루션층을 쌓아 올리는 방식에서는 일정 깊이 이상(약 20)에서는 최적화가 되지 않는 문제를 해결하기 위해, 추가적으로 항등사상을 추가하여, 순방향/역방향으로 신호의 흐름을 개선하여, 기존에 비해 매우 깊은 약 1100 이상의 깊이에서도 학습이 가능하도록 하였다.

또한, 그래디언트 소실/폭발 문제를 해결하기 위해, 활성화함수를 개선하는 방향으로도 연구가 진행되었다. 딥러닝 초기에는 활성화함수로 sigmoid 또는 tangent hyperbolic 함수가 사용되었으나, 두 함수 모두 입력이 어느 값 이상 커지거나 또는 작아지게 되면 기울기가 매우 작아져, 그래디언트 소실 문제가 발생하였다. 이를 개선하기 위해 ReLU(Rectified Linear Unit)[5]가 제안되었으며, 추가적인 개선을 위해, ReLU의 변형인 LReLU(Leaky ReLU)[6], PReLU(Parametric ReLU)[7], ELU(Exponential Linear Unit)[8]이 연구되었다. 위의

활성함수 중 가장 기본적으로 많이 사용되는 것은 ReLU이며, 따라서 최근 개발되는 정규화 및 각종 딥러닝 구조 개선 논문에서는 ReLU에 대해서만 검증이 실시되었다. 하지만 각 활성화함수는 서로 다른 특성을 가지고 있기 때문에, 본 논문에서는 각종 정규화 및 항등사상을 사용하는 residual network에 대해 활성화함수가 어떤 성능을 나타내는지 실험적으로 분석하고 경향을 제시한다. 이러한 분석을 바탕으로, 정규화 및 항등사상의 적용 여부에 따라 어떤 활성화함수를 선택하는 것이 좋은지 제시한다.

본 논문은 다음과 같이 구성된다. 2장에서는, 각종 정규화 방법, residual network, 각종 활성화함수에 대해 설명한다. 3장에서는 정규화 및 항등사상의 적용 유무에 따른 활성화함수의 성능을 검증하기 위한 실험 방법, 실험결과를 제시한다. 5장에서는 이 실험 결과를 바탕으로 활성화함수 선택 방법에 대해 제시한다.

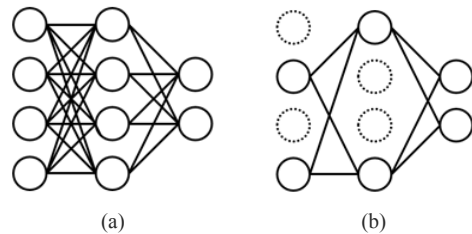


Fig. 1. Dropout (a) fully connected layer, (b) fully connected layer with dropout

2. 관련 연구

2.1 정규화

딥러닝에서 과적합 문제를 해결하기 위한 기본 방법은 학습 간 무작위 요소가 포함되어, 서로 다른 구조가 학습되는 효과를 이용한다. 대표적으로 많이 사용되어온 정규화 방법은 dropout이 있다. 이는 그림1의 (b)와 같이 학습 간 임의로 노드를 0으로 만들어, 여러 모델이 학습되는 효과를 가져 오며, 테스트를 할 때에는 평균을 취한다. 그리고 최근에 가장 많이 사용되고 있는 정규화 방법은 배치 정규화이다. 원래 목적은 과적합을 피하기 위한 목적은 아니었으나, 이 또한 학습 간 무작위 요소가 포함되어, 정규화에 많은 도움을 준다. 원래 목적은 미니배치 최적화를 수행 시, 각 미니배치는 서로 다른 분포를 가져 최적화에 방해되는 효과를 가져 온다. 이를 극복하기 위

해, 각 미니배치가 유사한 분포를 가지도록 아래 식(1)과 같이 n 차 입력 $x = [x_1, x_2, \dots, x_n]$ 의 각 요소에 대해 평균이 0, 분산이 1이 되도록 만들어 주었다.

$$\hat{x}_k = \frac{x_k - E[x_k]}{\text{Var}[x_k]} \quad (1)$$

이를 통해 얻은 효과로는 학습시간이 기존보다 빨라졌으며, 추가적으로 과적합을 예방하는 효과를 보였다. 기존 정규화 방법으로는 주로 dropout이 사용되었으나, 최근에는 기본적으로 배치 정규화를 모든 컨볼루션층에 적용시키고, 추가적으로 정규화가 필요하다면 dropout을 사용하는 추세이다.

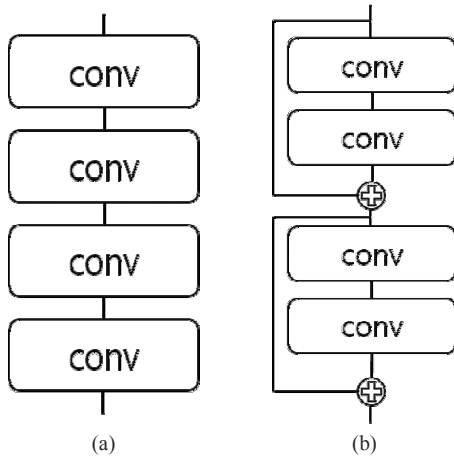


Fig. 2. conventional network and residual network (a) conventional convolutional network, (b) residual network

2.2 Residual Network

기존의 딥러닝 구조는 그림 2의 (a)와 같이 단순히 컨볼루션층을 쌓아가는 방식이었다. 하지만 일정이상의 깊이(약 20) 이상이 되면, 학습 가능한 파라미터수가 많아짐에도 불구하고 학습오류가 커지는 문제가 발생하였다. 이러한 문제를 해결하기 위해 그림 2의 (b)와 같은 구조를 가지는 residual network가 제안되었다. 이 구조는 순방향/역방향 신호의 흐름을 원활하도록 기존 딥러닝 구조에서 2개의 컨볼루션층 마다 입력을 바로 출력으로 연결 시켜주었다. 이를 통해 기존에는 불가능 했던 깊은 구조를 학습할 수 있게 해주었으며 무려 깊이가 1100이상의 깊은 구조를 학습할 수 있게 되었다.

2.3 활성화함수

2.3.1 ReLU(Rectified Linear Unit)

ReLU는 기존에 사용되던 활성화함수인 sigmoid 또는 tangent hyperbolic 함수의 그래디언트 소실문제를 해결하기 위해 아래의 식(2)와 같이 제안되었다.

$$\text{ReLU}(x) = \max(0, x) \quad (2)$$

ReLU는 입력이 0보다 클 때는 선형으로 기울기가 1이 보장되기 때문에 그래디언트가 사라지는 효과를 방지하였으며, 음의 입력에 대해 출력이 0이 되도록 하여 희소성(sparsity)에 도움을 주었다. 하지만 음의 입력에 대해 기울기가 0이 되므로, 학습기 기울기가 특정 경로를 통해서만 학습되는 문제를 가지고 있다.

2.3.2 Leaky ReLU(LReLU)

LReLU는 위에서 언급된 음의 입력에 대해 기울기가 0이 되는 문제를 해결하기 위해 아래 식(3)과 같이 제안되었다.

$$\text{LReLU}(x) = \max(ax, x) \quad (3)$$

음의 입력에 대해 작은 기울기를 가지게 함으로써, 학습이 더 원활하게 만들어 주었으며, a 값은 주로 0.01과 같이 작은 값을 사용하였다.

2.3.3 Parametric ReLU(PReLU)

PReLU는 LReLU의 음의 기울기를 사용자가 직접 정해줘야 한다는 문제를 해결하기 위해 제안되었으며 아래의 식(4)와 같다.

$$\text{PReLU}(x) = \max(px, x) \quad (4)$$

수식의 형태는 LReLU와 동일하지만, p 는 학습 가능한 파라미터이다.

2.3.4 ELU(Exponential Linear Unit)

ELU는 학습을 빠르게 하기 위해 제안된 활성화함수로, 활성화함수의 출력이 음의 값을 가지며, 컨볼루션층의 출력이 활성화함수를 거친 출력의 평균이 0이 가까워지도록 만든 활성화함수로 아래의 식(5)와 같다.

$$ELU(x) = \begin{cases} x & x \geq 0 \\ \alpha (\exp(x) - 1) & x < 0 \end{cases} \quad (5)$$

α 값은 주로 1이 사용되며, 음의 값을 가진다는 것은 ReLU와 다르나, 지수함수의 특성에 따라 대해 빠르게 기울기 감소하는 특징을 가지고 있다.

위의 4개의 활성화함수에 대해 특성을 정리하면 다음과 같이 분류할 수 있다. 첫째, 음의 입력에 대해 음의출력 여부이다. ReLU를 제외한 나머지 활성화함수들은 음의 출력을 가지고 있다. 둘째, 음의 입력에 대한 기울기이다. LReLU와 PReLU는 일정한 기울기를 가지며, ReLU는 0, ELU는 초기에는 기울기를 가지지만 빠르게 0으로 수렴한다. 셋째, 추가적인 학습 가능한 파라미터의 보유여부이다. 오직 PReLU만 학습 가능한 파라미터를 가지고 있다. 본 논문에서는 이러한 특성이 정규화 및 residual network에 대해 어떠한 성능 변화가 있는지 연구를 수행하였다.

Table 1. VGG16

layer name	output size (h x w x c)	layer
conv1	32x32x64	(conv 3x3, 64) X 2
pool1	16x16x64	maxpool 2x2 / 2
conv2	16x16x128	(conv 3x3, 128) X 2
pool2	8x8x256	maxpool 2x2 / 2
conv3	8x8x256	(conv 3x3, 256) X 3
pool3	4x4x256	maxpool 2x2 / 2
conv4	4x4x512	(conv 3x3, 512) X 3
pool4	2x2x512	maxpool 2x2 / 2
conv5	2x2x512	(conv 3x3, 512) X 2
pool5	1x1x512	maxpool 2x2 / 2
fc	10(or 100)	fully connected 10(or 100)

Table 2. Resnet18

layer name	output size (h x w x c)	layer
conv1	32x32x64	conv 3x3, 64
res1	16x16x64	[conv 3x3, 64]X 2 conv 3x3, 64
res2	8x8x128	[conv 3x3, 128]X 2 conv 3x3, 128
res3	4x4x256	[conv 3x3, 256]X 2 conv 3x3, 256
res4	2x2x512	[conv 3x3, 512]X 2 conv 3x3, 512
pool1	1x1x512	average pool 2x2 / 2
fc	10(or 100)	fully connected 10(or 100)

3. 실험

3.1 실험 구성

3.1.1 데이터 세트

본 실험에서는 영상분류에서 벤치마크 데이터세트로 활용되는 CIFAR10, 100에 대해서 실험을 실시하였다. 두 데이터세트 모두 총 50,000장의 학습데이터와 10,000장의 테스트 데이터가 있으며, 총 클래스의 개수는 각각 10개, 100개이며, 영상크기는 32 x 32의 RGB 영상이다. 입력의 전처리 방법은 RGB 각 채널에 대해 평균을 0, 분산을 1로 정규화를 수행하였다.

3.1.2 정규화

본 실험에서는 정규화 방법으로 1) 배치 정규화 + dropout, 2) 배치 정규화 + dropout + data augmentation 을 고려한다. 일반적으로 dropout은 배치 정규화를 적용 후 추가적으로 필요하다면 적용하기 때문에, 배치 정규화와 dropout을 하나의 정규화 방법으로써 실험을 실시한다. data augmentation은 일반적으로 많이 사용하는 horizontal random flip과 random crop을 적용하였다. random crop은 32 x 32 영상에 상하좌우 4픽셀씩 추가한 후 무작위로 32 x 32 이미지를 추출하였으며, 테스트 시에는 원본 이미지를 그대로 활용한다.

3.1.3 구조

항등사상이 적용유무에 따른 차이를 알아보기 위해 1) 항등사상이 적용되지 않는 VGG16[9], 2) 항등사상이 적용된 Resnet18에 대해 실험을 수행하였다. VGG16의 구조는 표1에 제시되어 있으며, (conv 3x3, 64) X 2는 커널(kernel) 크기가 3x3, 출력 피쳐맵(feature map)이 64, 그리고 이 컨볼루션층이 2번 반복됨을 의미하며, maxpool 2x2 / 2는 커널의 크기가 2x2이며, 스트라이드(stride)가 2를 의미한다. Resnet18의 구조는 표2에 제시되어 있으며 res1는 residual block1을 의미하며, 하나의 residual block이 그림1의 (b)와 같이 2개의 컨볼루션층, 1개의 항등사상으로 구성되고, 이것이 2개가 있음을 의미한다. VGG16의 경우 모든 컨볼루션층뒤에 배치 정규화를 적용하였으며, dropout은 컨볼루션층 사이에 0이 될 확률을 0.4로 하여 적용하였으며, 마지막 max pool 뒤에 0이 될 확률이 0.5로 하여 적용하였다. Resnet18의 경우 dropout은 적용하지 않았으며, 모든 컨볼루션층 뒤

에 배치 정규화를 적용하였다.

3.1.4 활성화함수

본 실험에서는 1) ReLU, 2) PReLU, 3) ELU 총 3가지 활성화함수에 대해 실험을 수행하였다. PReLU는 LReLU와 유사한 특성을 가지고 있으며, 더 일반화된 것이므로 PReLU에 대해서만 실험을 실시하였다.

3.1.5 최적화

최적화 방법으로는 그래디언트 하강 방식을 사용하였다. 미니배치의 크기는 128, 모멘트는 0.9을 사용하였다. 초기 학습율은 0.1로 설정하였고, 더 이상 학습이 진행되지 않을 때 마다 0.1배씩 학습율을 줄였다.

실험은 데이터셋 2가지, 정규화 2가지, 구조 2가지, 활성화함수 3가지의 모든 조합, 즉 24가지에 대해 실시하였다.

Table 3. classification accuracy for CIFAR10

	batch normalization + dropout		batch normalization + dropout + augmentation	
	VGG16	Resnet18	VGG16	Resnet18
ReLU	91.61%	88.95%	93.80%	95.36%
PReLU	92.22%	90.19%	93.81%	95.08%
ELU	91.38%	88.37%	93.99%	95.23%

Table 4. classification accuracy for CIFAR100

	batch normalization + dropout		batch normalization + dropout + augmentation	
	VGG16	Resnet18	VGG16	Resnet18
ReLU	67.71%	61.97%	70.82%	77.37%
PReLU	69.20%	66.24%	72.06%	76.48%
ELU	67.56%	61.38%	73.25%	77.80%

3.2 실험 결과

3.2.1 CIFAR10

CIFAR10에 대한 영상분류 정확도 결과는 표3에 제시되어 있다. 정규화가 배치 정규화 및 dropout이 적용됐을 때에는 항등사상과 상관없이, PReLU가 상당히 좋은 성능을 보인다. 하지만 data augmentation까지 추가로 적용됐을 때에는 활성화함수 간 성능격차가 상당히 줄어들었다. 이는 ReLU 및 ELU는 음의 입력에 대해 기울기가 0 또는 빠르게 0으로 근접하는 특성을 가지고 있어,

고정된 입력에 대해 특정 경로로 학습이 되지 않는 문제가 있다. 하지만 random flip, random crop을 통해 입력이 달라지는 효과를 주어 기존에는 학습되지 않는 경로에도 학습이 가능하게 해준다. 따라서 data augmentation은 모든 활성화함수에 대해 성능향상을 가져다주지만, 특히 ReLU, ELU의 성능향상이 두드러진다. 항등사상의 효과를 가장 많이 받는 활성화함수는 ReLU임을 알 수 있다. 이는 ReLU는 음의 입력에 대해 출력 및 기울기가 모두 0이므로, 순방향/역방향 신호흐름이 방해되는 문제가 있는데, 항등사상을 통해 이를 개선함으로써 성능향상이 있음을 알 수 있다.

3.2.2 CIFAR100

CIFAR100에 대한 영상분류 정확도 결과는 표4에 제시되어 있다. 배치 정규화 및 dropout만 적용되었을 때에는 CIFAR10과 동일하게 PReLU가 상당한 성능을 보임을 알 수 있다. 하지만 data augmentation이 적용되었을 때에는, 항등사상의 유무에 따라 CIFAR10과는 다른 경향이 보임을 알 수 있다. 우선 항등사상이 적용되지 않은 VGG16의 경우 음의 출력을 가지는 PReLU, ELU가 ReLU에 비해 상당히 잘 나옴을 알 수 있으며, 특히 음의 입력에 대해 기울기가 빠르게 0으로 근접하는 ELU가 상당히 좋은 성능을 보임을 알 수 있다. 항등사상이 적용되는 Resnet18의 경우에는 ReLU가 상당한 성능향상이 있다. 이는 앞에서 설명했듯이 항등사상이 순방향/역방향으로의 신호흐름을 개선함으로써, ReLU의 단점을 보완하여 얻은 결과로 볼 수 있다.

결론적으로 data augmentation이 적용되지 않았을 때에는 항등사상 유무와 상관없이 PReLU가 상당히 좋은 성능을 보인다. 하지만 data augmentation이 적용되었을 때에는, 항등사상이 없는 경우 모든 데이터셋에 대해 ELU가 가장 좋은 성능이 나왔으며, 항등사상이 있는 경우에는 ReLU, ELU가 좋은 성능을 보였다. 일반적으로 좋은 성능을 얻기 위해서는 data augmentation을 활용한다. 그리고 비교적 깊이가 얇은 구조의 경우 항등사상을 사용하지 않기 때문에, ELU를 선택, 깊은 구조의 경우 항등사상을 활용해야하기 때문에, ReLU, ELU를 검증해 보는 것을 추천한다.

4. 결론

본 논문에서는 딥러닝을 활용한 영상분류에 있어서, 정규화 및 항등사상이 활성화수에 미치는 영향에 대해 실험을 실시 및 분석을 실시하였다. 정규화 및 항등사상 적용 유무에 따른 영상분류 성능을 제시하고 경향을 분석하여 각 경우에 따라 어떤 활성화수를 활용하여 검증 을 시도해보는 것이 좋은지를 제시하였다. 이는 영상분 류를 위한 모델 개발 시 활성화수 선택에 대한 교차검증 횟수를 줄일 수 있을 것이다.

References

- [1] Srivastava, Nitish, et al., "Dropout: a simple way to prevent neural networks from overfitting," Journal of Machine Learning Research 15.1, pp. 1929-1958, 2014.
- [2] Wan, Li, et al., "Regularization of neural networks using dropconnect." Proceedings of the 30th international conference on machine learning (ICML-13), 2013.
- [3] Ioffe, Sergey, and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift." International Conference on Machine Learning, 2015.
- [4] He, Kaiming, et al., "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition, 2016.
DOI: <https://doi.org/10.1109/CVPR.2016.90>
- [5] Nair, Vinod, and Geoffrey E. Hinton, "Rectified linear units improve restricted boltzmann machines." Proceedings of the 27th international conference on machine learning (ICML-10), 2010.
- [6] Maas, Andrew L., Awni Y. Hannun, and Andrew Y. Ng., "Rectifier nonlinearities improve neural network acoustic models." Proc. ICML, vol. 30, no. 1, 2013.
- [7] He, Kaiming, et al., "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification." Proceedings of the IEEE international conference on computer vision, 2015.
DOI: <https://doi.org/10.1109/ICCV.2015.123>
- [8] Clevert, Djork-Arné, Thomas Unterthiner, and Sepp Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," arXiv preprint arXiv: 1511.07289, 2015.
- [9] Simonyan, Karen, and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, 2014.

류 서 현(Seo-Hyeon Ryu)

[정회원]



- 2013년 2월 : 경북대학교 전자공학과 (전자공학학사)
- 2015년 2월 : 한국과학기술원 공과대학원 전기 및 전자공학과(전자 및 전자공학석사)
- 2014년 12월 ~ 현재 : 국방기술품질원 연구원

<관심분야>
레이더, 영상처리

윤 재 복(Jae-Bok Yoon)

[정회원]



- 2011년 2월 : 경북대학교 전자공학과 (전자공학학사)
- 2013년 2월 : 광주과학기술원 의료시스템공학과(전자공학석사)
- 2013년 2월 ~ 현재 : 국방기술품질원 연구원

<관심분야>
레이더, 신호처리