Surpassing Human-Level Performance On ImageNet Classification

SKT Fellowship

LEE JINKYU

Department of Civil, Environmental and Architectural Engineering, Korea University

February 19, 2023

SKT AI Fellowship

Introduction

About PReLU & He initialization

1. Background – why PReLU & initialization

2. PReLU

- Definition
- Coefficient Table
- Analysis

3. Initialization

- History of initialization
- He initialization
- Comparison with Xavier

Background – Why PReLU & Initialization

Neccessity of PReLU and Initialization

기존 모델링 :
- 모델의 Complexity를 높임
- New nonlinear activation F를 제안 (ReLU)
- Sophisticated layer design
- Generalization & Regulation (Augmentation, large scale data)

→ ReLU의 우수성에도 불구하고 이를 중점적으로 연구가 이루어지지 않음

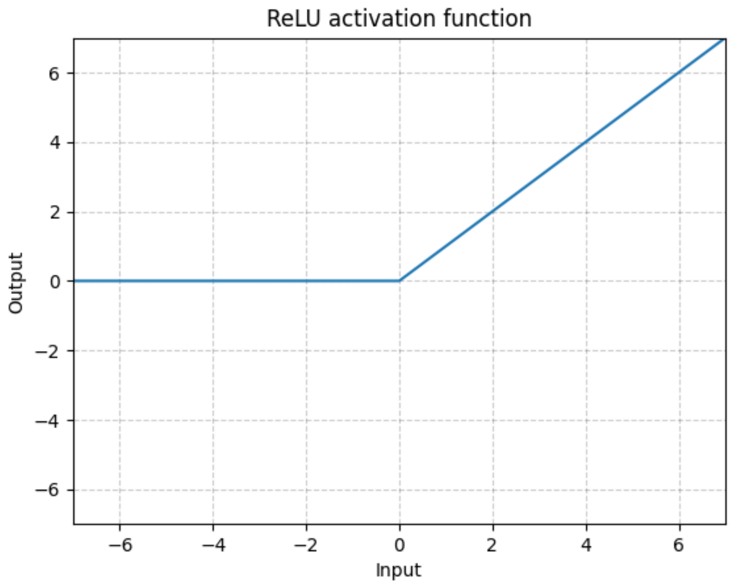
ReLU :
- Not symmetric function
- 대칭 분포에 대해 학습하고 싶어도 ReLU 특성에 의해 불가능

→ PReLU를 제안, 학습 파라미터를 설정해 문제 해결 + 적은 추가 Cost

Initialization :
- Rectified Models를 통해 Deep Model을 만들기 어렵다

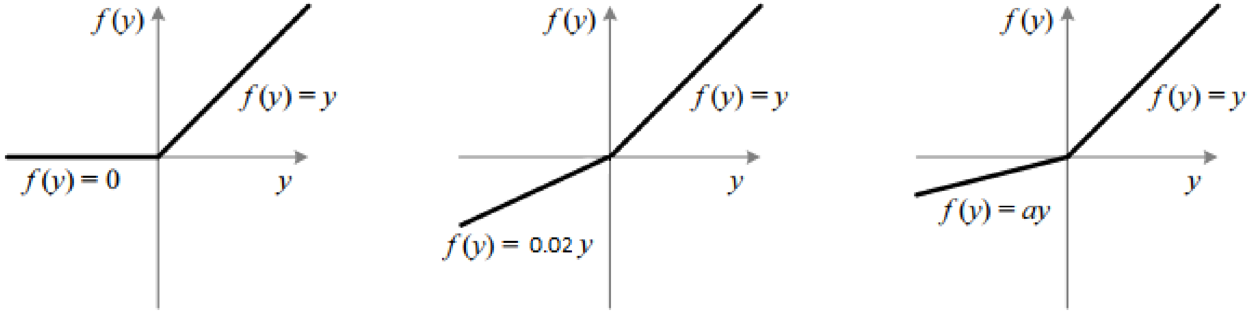
→ Initialized method derive

→ **이 모든 과정을 통해 Human Level Performance Surpass !!**



PReLU - Definition

What is PReLU



$$f(y_i) = \begin{cases} y_i, & \text{if } y_i > 0 \\ a_i y_i, & \text{if } y_i \leq 0 \end{cases}$$

$$\frac{\partial \mathcal{E}}{\partial a_i} = \sum_{y_i} \frac{\partial \mathcal{E}}{\partial f(y_i)} \frac{\partial f(y_i)}{\partial a_i},$$

$$\Delta a_i := \mu \Delta a_i + \epsilon \frac{\partial \mathcal{E}}{\partial a_i}.$$

Optimizing
= momentum.

추가되는 parameters가 적다 :

- No extra computation cost
- Less overfitting risk

+ Channel Shared Variant로도 사용 가능 (Only single variant add)

Optimization :

- 순전파, 역전파 시 다른 gradients과 함께 update

ReLU & Leaky ReLU & PReLU :

- ReLU보다 Leaky ReLU가 좋은 성능을 내려면 반복적인 작업(grid search)를 통해 a를 잘 구해줘야 하지만 PReLU는 학습을 통해 최적의 a를 도출한다

→ 성능 개선 + Computation Cost 적으로 부담 x

PReLU – Coefficient Table

Analysis of coefficient a

Conv 1의 coefficient는 0보다 훨씬 큰 값을 갖는다

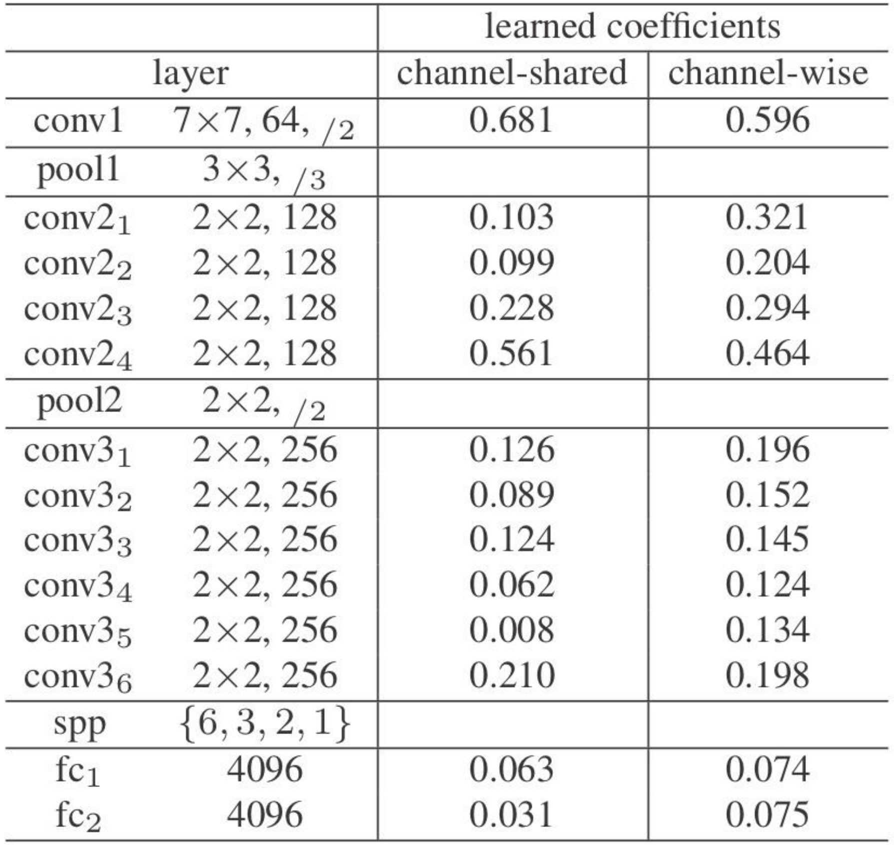
→ Conv 1의 필터는 edge, texture 등 이미지의 윤곽선을 인식하는 필터이다

→ Both positive and negative responses of filter are respected, 그렇기에 PReLU가 더 경제적인 방법 ! (Negative responses를 고려하므로)

Network가 깊어질수록 더 작은 coefficient를 갖는다

→ 깊어질수록 더 non-linear 해진다

→ 초기에는 이미지의 정보를 더 저장하다가 깊어질수록 더 discriminative 해진다



		learned coefficients	
layer		channel-shared	channel-wise
conv1	7×7, 64, /2	0.681	0.596
pool1	3×3, /3		
conv2_1	2×2, 128	0.103	0.321
conv2_2	2×2, 128	0.099	0.204
conv2_3	2×2, 128	0.228	0.294
conv2_4	2×2, 128	0.561	0.464
pool2	2×2, /2		
conv3_1	2×2, 256	0.126	0.196
conv3_2	2×2, 256	0.089	0.152
conv3_3	2×2, 256	0.124	0.145
conv3_4	2×2, 256	0.062	0.124
conv3_5	2×2, 256	0.008	0.134
conv3_6	2×2, 256	0.210	0.198
spp	{6, 3, 2, 1}		
fc_1	4096	0.063	0.074
fc_2	4096	0.031	0.075

PReLU – Analysis

Advantages of PReLU

PReLU의 training effect를 FIM(fisher information method)를 통해 확인

- FIM이 0에 가까워질수록 converge가 빨라진다

→ PReLU는 off-diagonal blocks of FIM을 0에 근사시킨다 (ReLU는 x)
→ PReLU가 ReLU에 비해 converge 속도가 빠르다
→ PReLU는 대칭분포의 학습을 용이하게 한다 (ReLU는 non-symmertric)

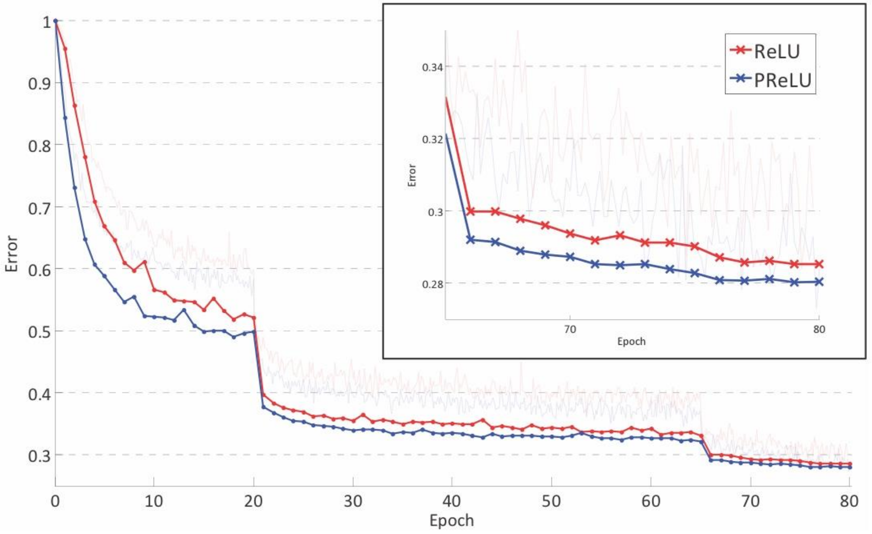


Figure 4. Convergence of ReLU (red) *vs*. PReLU (blue) of model A on ImageNet. Light lines denote the training error of the current mini-batch, and dark lines denote validation error of the center crops. In the zoom-in is the last few epochs. Learning rates are switched at 20 and 65 epochs.

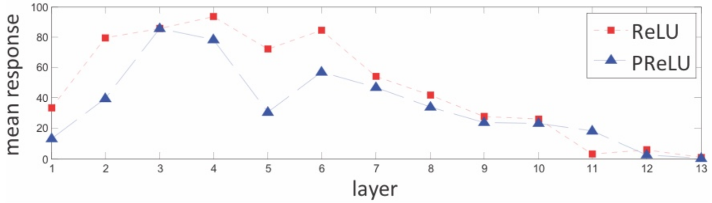


Figure 2. Mean responses of each layer for the trained models in Table 1. PReLU in general has smaller mean responses.

Initialization – History of initialization

Many initialization methods

Bad initialization hamper the learning of highly non-linear system

→ Propose robust initialize methods for extremely deep NN

History :

- Gaussian Random initialization → Deep한 모델에서는 잘 fitting x
- Pretraining → More training time & poorer local optima
- Auxiliary converge
- Xavier initialization → Based on “linear” activation
- **He initialization !**

Initialization – He initialization

Definition of He initialization

He initialization :

- Investigate the variance of the responses in each layer
- 앞 층의 노드가 n개 일 때, 표준편차가 (2/n)^(1/2)인 정규분포를 이용 → Xavier ((1/n)^(1/2)보다 2배 큰데, ReLU의 Negative part 때문)

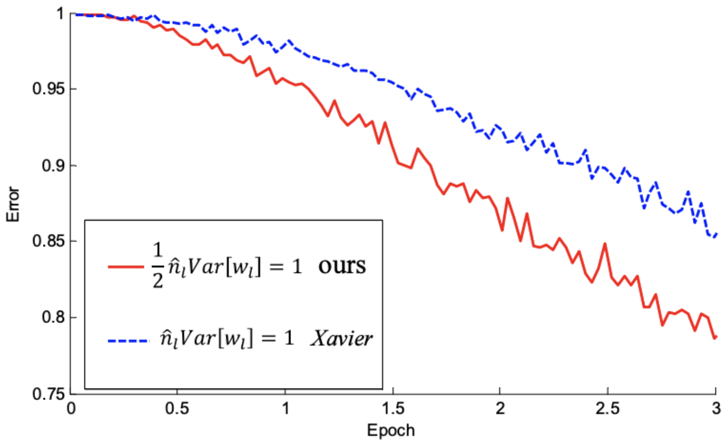
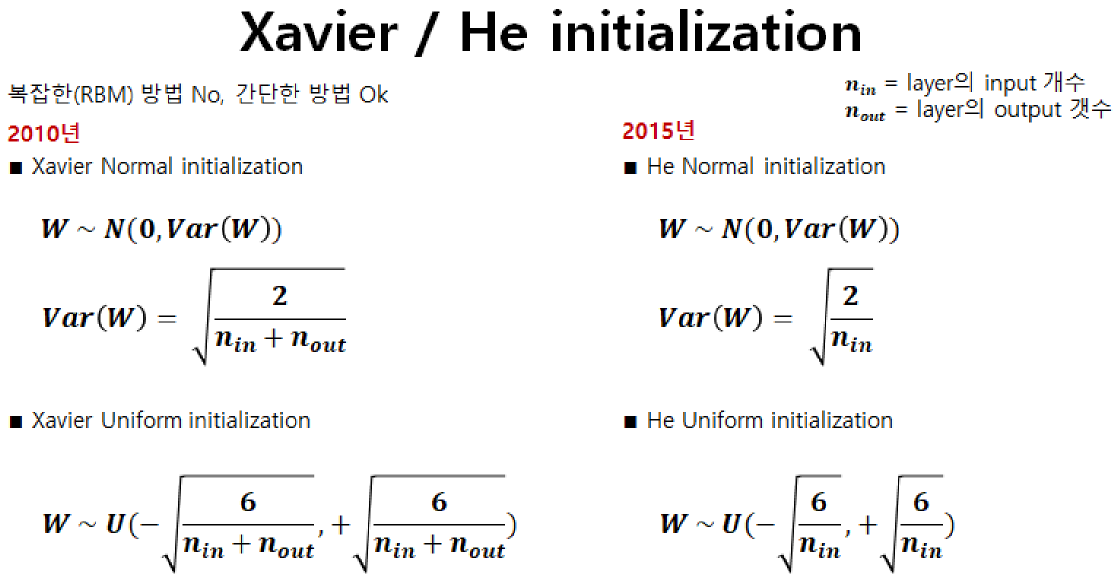


Figure 2. The convergence of a **22-layer** large model (B in Table 3). The x-axis is the number of training epochs. The y-axis is the top-1 error of 3,000 random val samples, evaluated on the center crop. We use ReLU as the activation for both cases. Both our initialization (red) and "*Xavier*" (blue) [7] lead to convergence, but ours starts reducing error earlier.



Xavier / He initialization

복잡한(RBM) 방법 No, 간단한 방법 Ok

n_{in} = layer의 input 개수
n_{out} = layer의 output 갯수

2010년

■ Xavier Normal initialization

$$W \sim N(0, Var(W))$$

$$Var(W) = \sqrt{\frac{2}{n_{in} + n_{out}}}$$

■ Xavier Uniform initialization

$$W \sim U(-\sqrt{\frac{6}{n_{in} + n_{out}}}, +\sqrt{\frac{6}{n_{in} + n_{out}}})$$

2015년

■ He Normal initialization

$$W \sim N(0, Var(W))$$

$$Var(W) = \sqrt{\frac{2}{n_{in}}}$$

■ He Uniform initialization

$$W \sim U(-\sqrt{\frac{6}{n_{in}}}, +\sqrt{\frac{6}{n_{in}}})$$

Initialization – Comparison with Xavier

Compare he with Xavier

He initialization :

- Linear case만 고려한 경우, Xavier의 std (1/n)^(1/2)의 분포는 가우시안 분포로 근사되고 수렴하기에 충분히 작지 않다 (…?)

→ 적당히 깊은 모델에 대해서는 둘다 수렴 but, he가 조금 더 빠르다, 그렇지민 성능 면에서는 큰 차이를 보이지 않는다

→ Extremely deep model에서는 he >>> Xavier, 하지만 extremely deep model은 여러 이유로 (degradation, saturation) 사용되지 않는 추세. 정말 깊은 모델에 효과적이라는 것 자체에 대한 의의 !

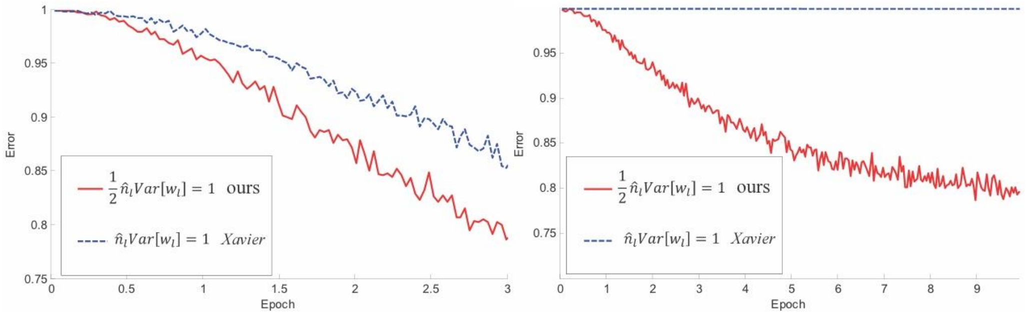


Figure 3. **Left**: convergence of a **22-layer** model (B in Table 3). The x-axis is training epochs. The y-axis is the top-1 val error. Both our initialization (red) and "*Xavier*" (blue) [8] lead to convergence, but ours starts reducing error earlier. **Right**: convergence of a **30-layer** model. Our initialization is able to make it converge, but "*Xavier*" completely stalls. We use ReLU in both figures.