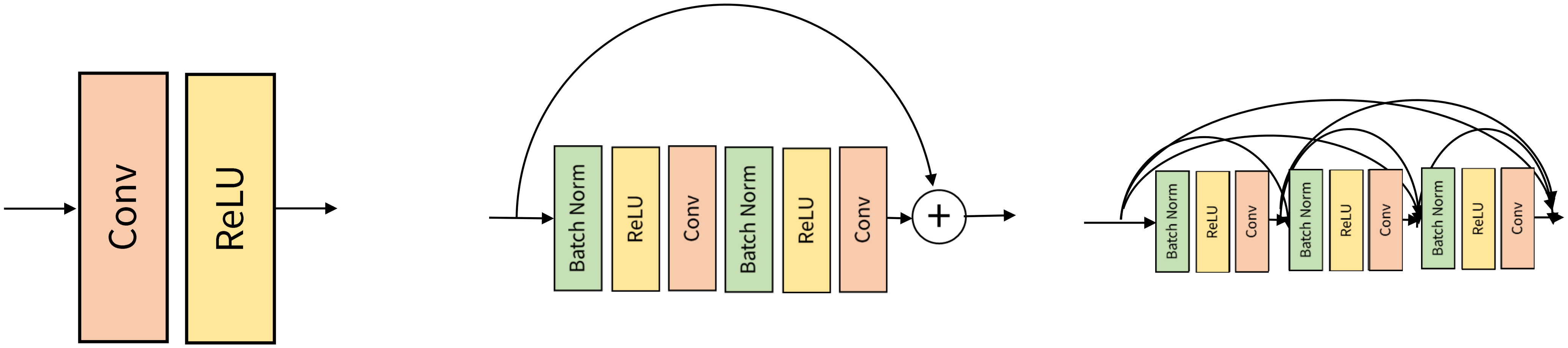
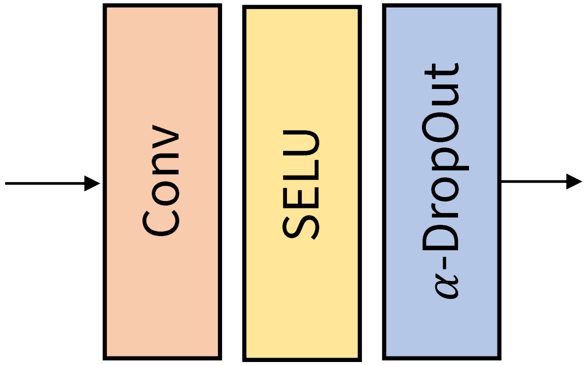
CNN의 발전



Vanilla CNN　　ResNet　　DenseNet

ResNet, DenseNet 등 다양한 네트워크가 등장하면서, Vanilla CNN은 설 자리를 잃었다.

자가 정규화 신경망 Self-Normalizing Neural Network



Self-Normalizing NN

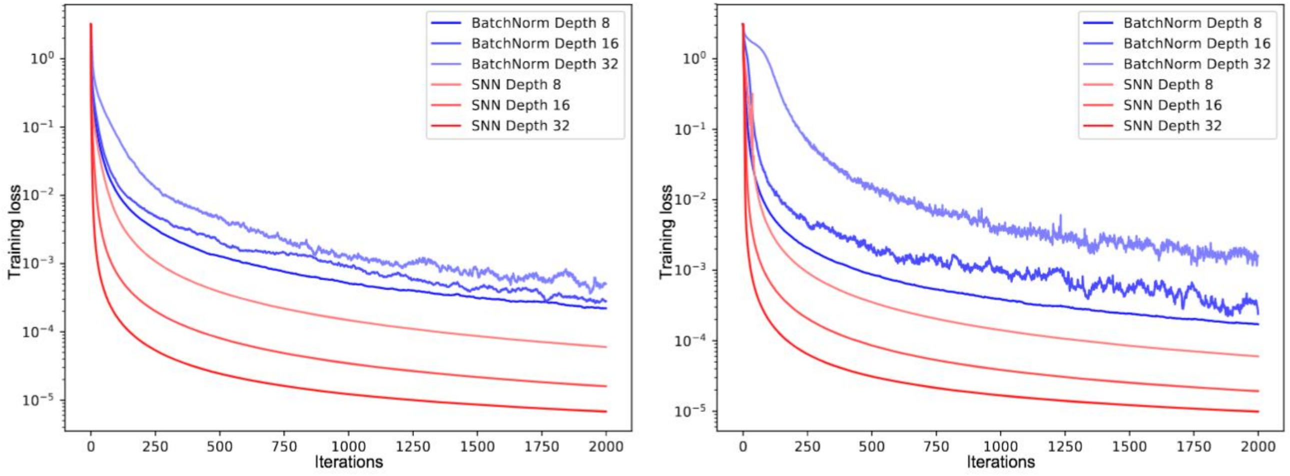


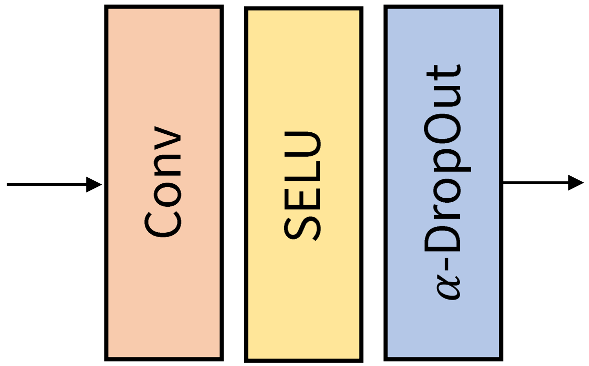
Figure 1: The left panel and the right panel show the training error (y-axis) for feed-forward neural networks (FNNs) with batch normalization (BatchNorm) and self-normalizing networks (SNN) across update steps (x-axis) on the MNIST dataset the CIFAR10 dataset, respectively. We tested networks with 8, 16, and 32 layers and learning rate 1e-5. FNNs with batch normalization exhibit high variance due to perturbations. In contrast, SNNs do not suffer from high variance as they are more robust to perturbations and learn faster.

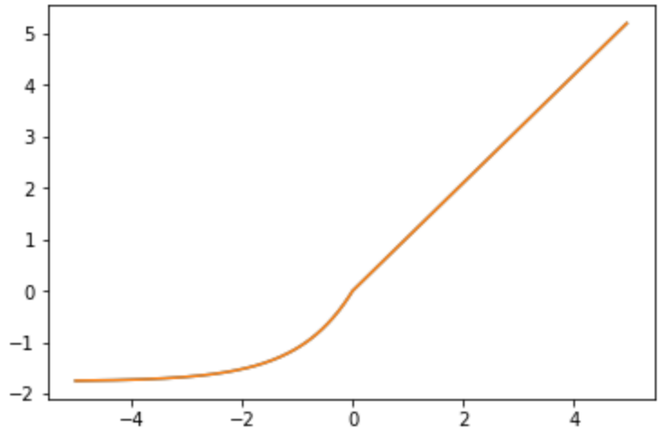
SNN은 바닐라 CNN에 약간의 변화를 줌으로써, 스스로 정규화하는 계층을 형성한다.

BatchNorm 대비 훨씬 더 안정적이면서도 좋은 성능을 보인다.

SELU Scaled Exponential Linear Unit

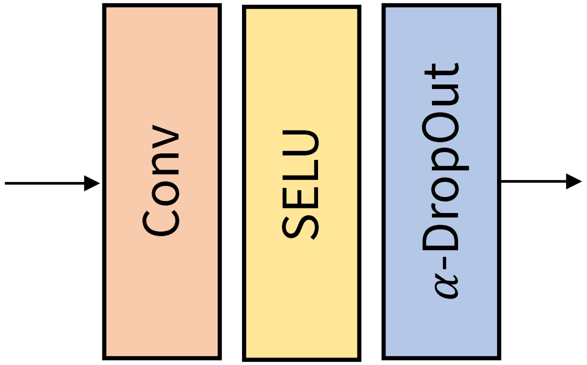
$$\mathrm{selu}(x) = \lambda \begin{cases} x & , \text{if } x > 0 \\ \alpha e^{x} - \alpha & , \text{if } x \leq 0 \end{cases}$$





SELU는 ReLU와 달리, 음수 값을 Exponential하게 활성화 하는 특징이 있다. λ와 α의 값에 따라 특성이 결정된다. 자가 정규화를 위한 이 값은 $\lambda = 1.0507, \alpha = 1.67326$으로 알려져 있다.

α-DropOut



$$\boxed{\alpha\mathrm{DropOut}(x) = a\big(xd + \alpha'(1-d)\big) + b}$$

$$a = (q + \alpha'^2 q(1-q))^{-\frac{1}{2}}$$

$$b = -(q + \alpha'^2 q(1-q))^{-\frac{1}{2}}\big((1-q)\alpha'\big)$$

$$\alpha' = -\lambda \times \alpha = -1.7581$$

$1 - q$: DropOut Rate (0.05 ~ 0.1)

DropOut이 ReLU에 잘 동작하는 것을 반영하여, SELU에 적합하도록 변형한 버전.

복잡한 Derivation을 통해 α'값을 결정해 두었으며, dropout-rate는 5~10%로 비교적 작다.

SNN Results

Table 2: Comparison of FNNs at the Tox21 challenge dataset in terms of AUC. The rows represent different methods and the columns different network depth and for ResNets the number of residual blocks ("na": 32 blocks were omitted due to computational constraints). The deeper the networks, the more prominent is the advantage of SNNs. The best networks are SNNs with 8 layers.

method	2	3	4	6	8	16	32
	#layers / #blocks						
SNN	*83.7* ± 0.3	**84.4** ± 0.5	**84.2** ± 0.4	**83.9** ± 0.5	**84.5** ± 0.2	**83.5** ± 0.5	**82.5** ± 0.7
Batchnorm	80.0 ± 0.5	79.8 ± 1.6	77.2 ± 1.1	77.0 ± 1.7	75.0 ± 0.9	73.7 ± 2.0	76.0 ± 1.1
WeightNorm	*83.7* ± 0.8	82.9 ± 0.8	82.2 ± 0.9	*82.5* ± 0.6	*81.9* ± 1.2	78.1 ± 1.3	56.6 ± 2.6
LayerNorm	**84.3** ± 0.3	*84.3* ± 0.5	*84.0* ± 0.2	*82.5* ± 0.8	80.9 ± 1.8	78.7 ± 2.3	78.8 ± 0.8
Highway	83.3 ± 0.9	83.0 ± 0.5	82.6 ± 0.9	82.4 ± 0.8	80.3 ± 1.4	80.3 ± 2.4	79.6 ± 0.8
MSRAinit	82.7 ± 0.4	81.6 ± 0.9	81.1 ± 1.7	80.6 ± 0.6	80.9 ± 1.1	80.2 ± 1.1	80.4 ± 1.9
ResNet	82.2 ± 1.1	80.0 ± 2.0	80.5 ± 1.2	81.2 ± 0.7	81.8 ± 0.6	*81.2* ± 0.6	na

Table 3: Comparison of FNNs and reference methods at HTRU2 in terms of AUC. The first, fourth and seventh column give the method, the second, fifth and eight column the AUC averaged over 10 cross-validation folds, and the third and sixth column the p-value of a paired Wilcoxon test of the AUCs against the best performing method across the 10 folds. FNNs achieve better results than Naive Bayes (NB), C4.5, and SVM. SNNs exhibit the best performance and set a new record.

method	FNN methods AUC	p-value	method	FNN methods AUC	p-value	ref. methods method	AUC
SNN	0.9803 ± 0.010						
MSRAinit	0.9791 ± 0.010	3.5e-01	LayerNorm	0.9762* ± 0.011	1.4e-02	NB	0.976
WeightNorm	0.9786* ± 0.010	2.4e-02	BatchNorm	0.9760 ± 0.013	6.5e-02	C4.5	0.946
Highway	0.9766* ± 0.009	9.8e-03	ResNet	0.9753* ± 0.010	6.8e-03	SVM	0.929

아직 딥러닝에서도 세심한 수학적 연구가 직관을 뛰어넘을 수 있다는 것을 보여 준 훌륭한 사례