
AReLU: Attention-based Rectified Linear Unit

supplemental materials

Anonymous Author(s)

Affiliation

Address

email

1 Details of ConvMNIST

ConvMNIST is a VGG [18]-like network but with fewer layers, as shown in Figure 1. The activation layers will be placed with specified activation functions while experiments.

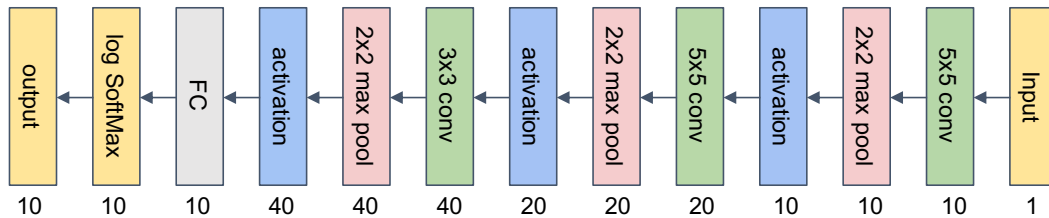


Figure 1: The network architecture of ConvMNIST. The number under the block indicates that output channels of current layer.

2 More results on MNIST

In Table 1, best testing accuracy on MNIST for five trainings of MNIST-Conv after the *first epoch* with different optimizers and learning rates are reported. In Table 3, mean testing accuracy of five-time training of MNIST-Conv trained for 20 epochs with different learning rates on MNIST are reported. In Table 2, best testing accuracy of five-time training of MNIST-Conv trained for 20 epochs with different learning rates on MNIST are reported. We compare AReLU with 13 non-learnable and 5 learnable activation functions. The number of parameters per activation unit are listed beside the name of the learnable activation functions. The best numbers are shown in bold text with blue color for non-learnable methods and red for learnable ones. At the bottom of the table, we report the improvement of AReLU over the best among other non-learnable and learnable methods, in blue and red respectively.

At the meantime, we also plot the mean training loss and testing accuracy of five runs with different optimizers and learning rates in Figure 2 and Figure 3.

3 Full result of transfer learning

The full table of transfer learning is shown in Table 4.

Table 1: Best testing accuracy (%) on MNIST for five trainings of MNIST-Conv after the *first epoch* with different optimizers and learning rates. We compare AReLU with 13 non-learnable and 5 learnable activation functions. The number of parameters per activation unit are listed beside the name of the learnable activation functions. The best numbers are shown in bold text with blue color for non-learnable methods and red for learnable ones. At the bottom of the table, we report the improvement of AReLU over the best among other non-learnable and learnable methods, in blue and red color respectively.

Learning Rate	1×10^{-2}		1×10^{-3}		1×10^{-4}		1×10^{-5}	
Optimizer	Adam	SGD	Adam	SGD	Adam	SGD	Adam	SGD
CELU [2]	98.49	96.56	96.41	75.67	85.62	17.64	34.23	10.79
ELU [3]	98.36	96.64	96.34	65.66	86.77	22.61	28.91	11.36
GELU [8]	98.68	96.02	96.33	14.44	84.45	14.61	20.64	13.30
LeakyReLU [12]	98.29	95.93	96.19	44.76	84.86	13.20	20.55	11.49
Maxout [5]	97.79	96.09	96.45	79.21	85.62	13.99	22.05	10.55
ReLU [15]	98.13	96.33	96.07	49.78	86.07	12.67	19.87	10.24
ReLU6 [11]	98.18	96.05	96.55	56.07	83.42	13.13	17.42	10.27
RReLU [19]	98.52	96.30	95.98	61.78	86.97	10.36	20.61	11.35
SELU [10]	97.72	96.88	97.01	83.85	87.53	22.09	37.98	10.59
Sigmoid	97.62	11.35	85.29	11.35	11.47	11.35	11.35	10.28
Softplus [4]	97.80	93.83	94.58	11.35	75.05	11.35	11.35	10.32
Swish [16]	98.32	95.28	96.47	12.38	85.52	11.82	15.51	10.27
Tanh	97.32	94.40	96.84	69.29	81.50	16.32	29.92	11.35
APL [1] (2)	98.48	96.25	95.50	27.75	80.56	10.28	19.50	15.47
Comb [13] (1)	98.42	96.54	96.07	57.88	85.59	11.67	25.40	10.72
PAU [14] (10)	98.42	97.94	97.07	76.69	89.71	11.35	18.11	14.54
PReLU [7] (1)	98.52	96.10	96.33	61.72	87.24	15.86	18.31	11.40
SLAF [6] (2)	96.69	97.27	95.76	84.13	76.84	15.60	11.19	13.09
AReLU (2)	98.46	97.60	97.29	93.83	90.91	61.06	48.06	19.84
Improvement	-0.22	+0.72	+0.28	+9.98	+3.38	+38.45	+10.08	+6.54
Improvement	-0.02	-0.34	+0.22	+9.70	+1.20	+45.20	+22.66	+4.37

4 More results on Segmentation

We visualize the learning procedure of UNet [17] on a testing MRI image in Figure. 4. AReLU can learn a better silhouette information than ReLU.

References

- [1] Forest Agostinelli, Matthew Hoffman, Peter Sadowski, and Pierre Baldi. Learning activation functions to improve deep neural networks. *arXiv preprint arXiv:1412.6830*, 2014.
- [2] Jonathan T Barron. Continuously differentiable exponential linear units. *arXiv preprint arXiv:1704.07483*, 2017.
- [3] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015.
- [4] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323, 2011.
- [5] Ian J Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron Courville, and Yoshua Bengio. Maxout networks. *arXiv preprint arXiv:1302.4389*, 2013.
- [6] Mohit Goyal, Rajan Goyal, and Brejesh Lall. Learning activation functions: A new paradigm for understanding neural networks, 2019.
- [7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

Table 2: Best testing accuracy (%) of five-time training of MNIST-Conv trained for 20 epochs with different learning rates on MNIST. We compare ARELU with 13 non-learnable and 5 learnable activation functions. The number of parameters per activation unit are listed beside the name of the learnable activation functions. The best numbers are shown in bold text with blue color for non-learnable methods and red for learnable ones. At the bottom of the table, we report the improvement of ARELU over the best among other non-learnable and learnable methods, in blue and red respectively.

Learning Rate	1×10^{-2}		1×10^{-3}		1×10^{-4}		1×10^{-5}	
Optimizer	Adam	SGD	Adam	SGD	Adam	SGD	Adam	SGD
CELU [2]	98.67	99.05	99.14	97.98	97.88	90.89	91.33	17.39
ELU [3]	98.70	99.04	99.10	97.93	97.96	90.84	91.40	20.51
GELU [8]	99.03	98.99	99.14	97.65	98.04	85.60	90.19	13.30
LeakyReLU [12]	98.79	99.04	99.10	97.85	97.91	90.00	90.90	16.63
Maxout [5]	98.30	98.86	98.82	97.98	97.69	89.98	91.04	23.90
ReLU [15]	98.80	99.06	99.17	97.64	97.85	86.53	89.98	13.95
ReLU6 [11]	98.55	99.01	99.17	98.03	98.14	86.57	88.98	18.79
RReLU [19]	98.98	99.05	99.19	97.90	97.76	88.23	90.31	18.31
SELU [10]	98.53	98.96	98.91	98.04	98.06	92.09	92.26	37.85
Sigmoid	98.99	96.37	98.78	11.35	94.02	11.35	11.35	11.35
Softplus [4]	99.07	98.86	99.04	97.52	96.86	11.88	80.60	16.34
Swish [16]	98.83	98.85	99.09	97.65	97.80	36.30	89.21	10.29
Tanh	97.96	98.89	99.02	97.27	98.09	78.30	88.17	17.76
APL [1] (2)	98.80	99.02	99.00	97.73	97.35	49.37	87.24	16.14
Comb [13] (1)	99.03	99.10	99.16	97.71	97.90	86.52	89.52	12.48
PAU [14] (10)	99.19	99.07	99.18	98.82	98.28	95.75	92.98	15.82
PReLU [7] (1)	98.97	99.11	99.11	97.93	98.07	90.34	91.31	17.82
SLAF [6] (2)	98.88	98.97	98.82	98.50	97.82	94.88	87.67	25.35
ARELU (2)	99.08	99.07	99.05	98.60	98.40	96.32	93.75	85.45
Improvement	+0.01	+0.01	-0.14	+0.56	+0.26	+4.23	+1.49	+47.60
Improvement	-0.11	-0.04	-0.13	-0.22	+0.12	+0.57	+0.77	+60.10

- [8] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.
- [9] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus), 2016.
- [10] Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. Self-normalizing neural networks. In *Advances in neural information processing systems*, pages 971–980, 2017.
- [11] Alex Krizhevsky and Geoff Hinton. Convolutional deep belief networks on cifar-10. *Unpublished manuscript*, 40(7):1–9, 2010.
- [12] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3, 2013.
- [13] Franco Manessi and Alessandro Rozza. Learning combinations of activation functions. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 61–66. IEEE, 2018.
- [14] Alejandro Molina, Patrick Schramowski, and Kristian Kersting. Pad\`e activation units: End-to-end learning of flexible activation functions in deep networks. *arXiv preprint arXiv:1907.06732*, 2019.
- [15] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.
- [16] Prajit Ramachandran, Barret Zoph, and Quoc V Le. Searching for activation functions. *arXiv preprint arXiv:1710.05941*, 2017.
- [17] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.
- [18] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Table 3: Mean testing accuracy (%) of five-time training of MNIST-Conv trained for 20 epochs with different learning rates on MNIST. We compare AReLU with 13 non-learnable and 5 learnable activation functions. The number of parameters per activation unit are listed beside the name of the learnable activation functions. The best numbers are shown in bold text with blue color for non-learnable methods and red for learnable ones. At the bottom of the table, we report the improvement of AReLU over the best among other non-learnable and learnable methods, in blue and red respectively.

Learning Rate	1×10^{-2}		1×10^{-3}		1×10^{-4}		1×10^{-5}	
Optimizer	Adam	SGD	Adam	SGD	Adam	SGD	Adam	SGD
CELU [2]	98.62	98.93	99.05	97.73	97.70	89.58	90.58	14.96
ELU [3]	98.55	98.94	99.02	97.82	97.70	89.24	90.46	15.41
GELU [9]	98.85	98.93	99.08	97.51	97.67	51.19	88.94	10.94
LeakyReLU [12]	98.66	98.92	98.96	97.74	97.61	74.01	89.21	13.27
Maxout [5]	98.23	98.78	98.76	97.67	97.46	89.52	90.04	14.85
ReLU [15]	98.72	98.98	99.05	97.57	97.58	81.63	88.88	11.03
ReLU6 [11]	98.51	98.93	99.02	97.79	97.96	81.96	88.14	12.44
RReLU [19]	98.78	98.94	99.06	97.77	97.62	87.64	89.59	13.37
SELU [10]	98.34	98.91	98.84	97.98	97.88	91.56	90.91	33.48
Sigmoid	81.05	96.24	98.72	11.35	92.96	11.35	11.35	10.67
Softplus [4]	98.95	98.78	98.93	97.30	96.57	11.52	78.36	12.50
Swish [16]	98.77	98.80	99.02	97.44	97.51	23.77	88.53	10.05
Tanh	97.91	98.86	98.96	96.94	97.97	75.66	86.99	14.76
APL [1] (2)	98.72	98.92	98.94	97.56	97.22	37.67	84.95	13.52
Comb [13] (1)	98.88	99.01	99.04	97.56	97.55	85.60	88.39	10.94
PAU [14] (10)	99.17	99.01	99.07	98.78	98.15	95.21	92.22	12.86
PRReLU [7] (1)	98.89	98.86	99.01	97.81	97.77	88.67	89.69	13.36
SLAF [6] (2)	98.80	98.86	98.67	98.37	97.60	94.61	86.10	18.91
AReLU (2)	98.94	99.01	98.97	98.46	98.22	96.00	93.48	73.00
Improvement	-0.01	+0.03	-0.11	+0.48	+0.25	+4.44	+2.57	+39.52
Improvement	-0.23	+0.00	-0.10	-0.32	+0.07	+0.79	+1.26	+54.09

Table 4: Test accuracy (%) on SVHN by models (with different activation functions) trained directly on SVHN (w/o pretrain), trained on MNIST but not finetuned (w/o finetune), as well as pretrained on MNIST and finetuned on SVHN (pretrain+finetune).

	CELU	ELU	GELU	LReLU	Maxout	RReLU	ReLU	ReLU6	SELU	Sigmoid	Softplus	Swish	Tanh	APL	Comb	PAU	PRReLU	SLAF	AReLU
w/o pretrain	15.59	19.59	19.59	19.58	23.01	19.58	19.58	19.96	19.58	19.58	19.58	19.58	19.58	19.58	19.58	19.58	19.58	19.58	24.95
w/o finetune	28.56	31.95	37.38	28.11	36.52	33.38	36.87	31.74	32.57	15.73	14.39	27.23	21.92	36.20	35.89	24.67	33.45	35.74	31.91
w/ finetune	71.11	72.39	71.64	72.18	71.88	69.87	70.58	69.90	73.43	33.83	69.18	67.75	64.78	74.21	69.92	74.70	71.15	73.12	76.77

- 63 [19] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations
64 in convolutional network. *arXiv preprint arXiv:1505.00853*, 2015.

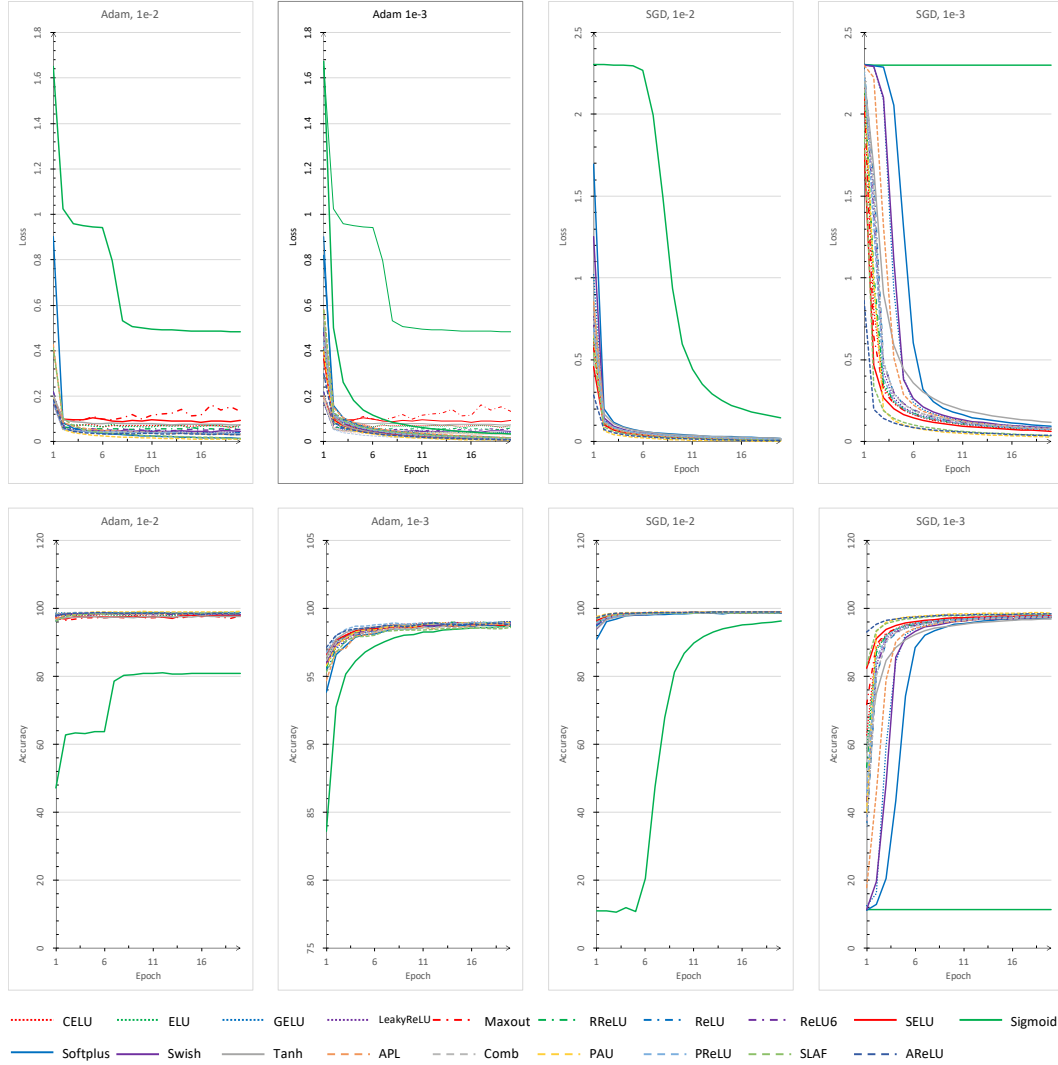


Figure 2: The plots of mean training loss and testing accuracy (%) on MNIST for five-time trainings of MNIST-Conv over increasing training epochs with different optimizers and learning rates.

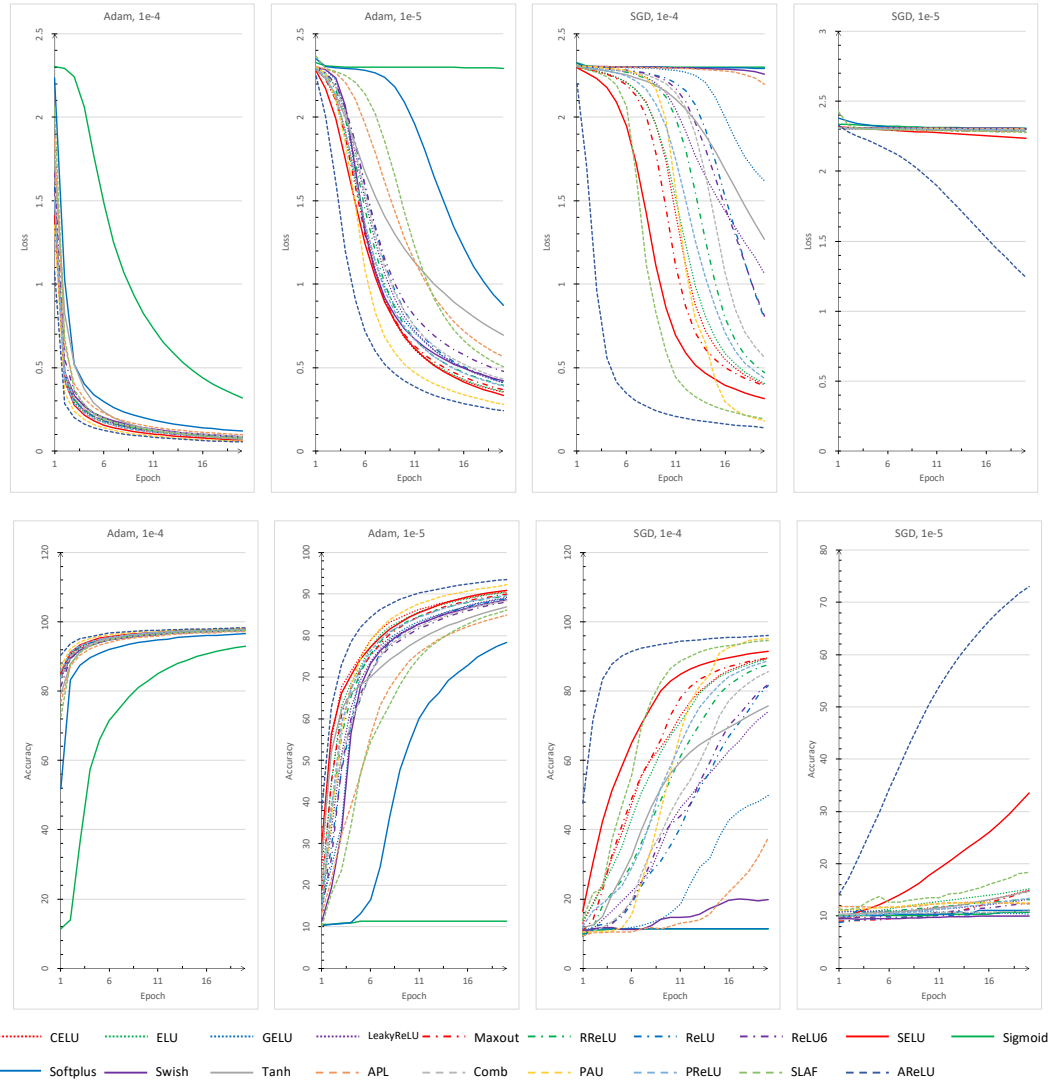


Figure 3: The plots of mean training loss and testing accuracy (%) on MNIST for five-time trainings of MNIST-Conv over increasing training epochs with different optimizers and learning rates.

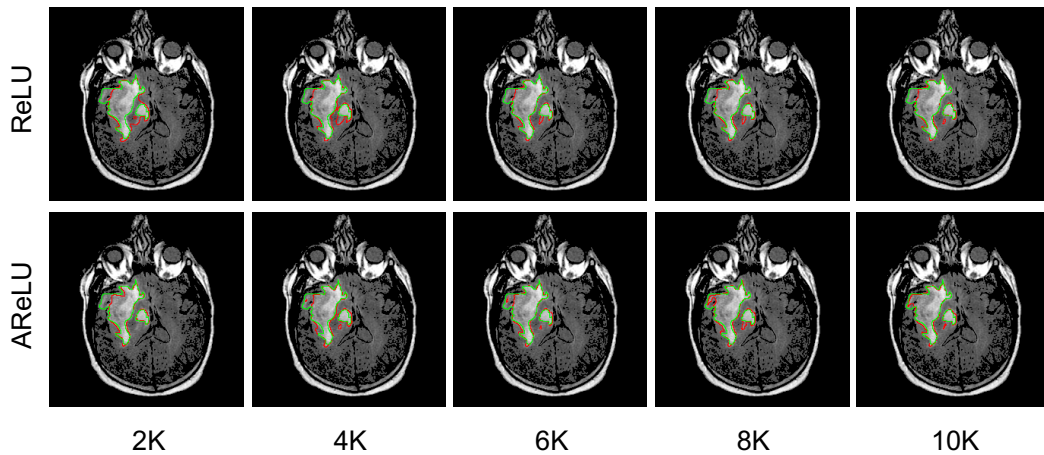


Figure 4: The learning procedure of UNet at 2k, 4k, 6k, 8k, 10k iterations.