A comparison of activation functions for deep learning on Fashion-MNIST

markers:note that this paper came out of previous work on political science datasets with ANNs, but the dataset has changed to allow for consideration of deep learning. That old paper is cited anyway, since it was inspirational. However, the benchmark paper is cited below at Han Xiao, Kashif Rasul, and Roland Vollgraf. 2017. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747 (2017), and additional benchmarks can be found at the Fashion-MNIST github page.

Michael McKenna[1],

[1] School of Politics and International Relations, Australian National University, Canberra, Australia; Research School of Computer Science, Australian National University, Canberra, Australia. ORCID http://orcid.org/0000-0002-8124-591X {Michael.mckenna@anu.edu.au}

Abstract. MNIST is a well known dataset for benchmarking, but it is almost 20 years old. Some within the deep learning community have called for MNIST to be put to bed, and Fashion-MNIST has been proposed as a structurally similar but more difficult alternative. Our objectives here are twofold. Firstly, we provide a benchmark for currently missing multilayer non-convolutional feed-forward neural networks on Fashion-MNIST. Secondly, we test the effectiveness of modern activation functions (comparing ELU [Exponential Linear Units], ReLUs, and sigmoid functions). Both objectives are novel, since Fashion-MNIST has a large number of benchmarks but none with non-convolutional deep architectures, and ELUs are rarely used outside of convolutional networks. Our performance is substantially worse than the convolutional benchmarks, and we did observe some benefits of ELUs and ReLUs, when the network had been partially trained.

Keywords: classification**,** units, ELU, ReLU, sigmoid, activation_function, MNIST, Fashion_MNIST

1. Introduction and Dataset Choice

The perennial overuse of MNIST has attracted ire from many machine learning experts. There are three common reasons for this:

1. “MNIST is too easy. Convolutional nets can achieve 99.7% on MNIST. Classic machine learning algorithms can also achieve 97% easily. Check out our side-by-side benchmark for Fashion-MNIST vs. MNIST, and read "Most pairs of MNIST digits can be distinguished pretty well by just one pixel."
2. MNIST is overused. In this April 2017 Twitter thread, Google Brain research scientist and deep learning expert Ian Goodfellow calls for people to move away from MNIST.
3. MNIST can not represent modern CV tasks, as noted in this April 2017 Twitter thread, deep learning expert/Keras author François Chollet.” (Xiao, Rasul, Vollgraf, 2017; Zalandoresearch, 2017)

A suggested replacement for MNIST is entitled Fashion-MNIST, which is a demonstrably harder classification problem that is nevertheless similar to MNIST in its structure. Like the original MNIST, the task is to classify black and white images into ten classes. Fashion-MNIST images are taken from amazon.com . However, most classifiers perform substantially worse on Fashion-MNIST, with 5-10% less accuracy. While Fashion-MNIST is extensively benchmarked, the benchmarks use either convolutional neural networks or non-neural network machine learning models, and this is a gap we aim to fill.

In 2017 Xiao, Rasul and Vollgraf published some benchmarks for this dataset. Additional benchmarks can be found at https://github.com/zalandoresearch/fashion-mnist . We will be revisiting this paper and trying to add to its results. We do not expect to beat the results in this paper, given that some classifiers such as RandomForest are competitive with shallow convolutional networks that may outperform ordinary multilayer networks on MNIST. We will also be testing whether or not the recently invented *ELU* or the *ReLU* activation functions outperform the contemporary sigmoid function. Importantly, Fashion-MNIST does not have benchmarks for performance on non-convolutional deep architectures for any of these activation functions, and this is particularly reflective of a lack of information on the literature on how *ELU*s perform outside of convolutional neural networks, where they were invented (Clevert et al, 2016). The (non-NN) classifiers in the original paper have performance ranging from 51% to 86.8%, while the deep convolutional classifiers submitted by others have accuracy up to 94.9% (with ResNet18).

Investigation Aims

Our aims are twofold. We aim to add to the extensive benchmarks on Fashion-MNIST, and also to test out modern activation functions (ELUs and ReLUs) in comparison to sigmoid functions commonly used on MNIST.

2 Method

Summary of parameter choices

Dependent Variables:

Activation Functions - ELUs (Exponential Linear Units), ReLUs (Rectified Linear Unit), Sigmoid function.

Parameters taken from Fashion-MNIST:

We haven't used cross-validation, as the dataset is sufficiently large and cross-validation does not seem to have been used in any of the other benchmarks. Most benchmarks with a specified number of epochs have 2 epochs, so that is our upper limit (although there is nothing to say that the network will not perform better with more than two epochs). Normalisation did not improve initial results on any of the architectures, and many of the other benchmarks did not normalise the data (see Zalandoresearch, 2017) so we have not used it.

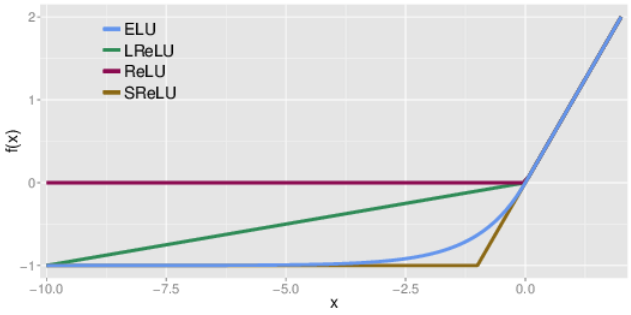
Other parameters and performance measurements
After some brief testing, we determined that a learning rate of 0.01 would be appropriate for all three activation functions as applied. We picked a batch size of 100 because for a 10-class problem there is sufficient information for the network to learn. We chose the RPROP optimizer for its speed and lack of sensitivity to the size of the gradient. For our final activation and loss function we chose softmax / CrossEntropyLoss, which are well suited to classification problems with multiple classification.

Model Design Principles

The dataset is our dependent variables, while we are changing the activation function (our independent variable). We normalised the data to its mean and standard deviation. Though many benchmarks have transformed and normalised the data, many have not, and because we are trying to take a new (if somewhat antiquated) approach to this dataset we can make an independent choice.

We used PyTorch, to compare RELUs, ELUs, and sigmoid activation functions.
To make comparison at the early stages of training easier, we believe batch learning (with a batch size of 100) with equal samples of each class is appropriate.

Understanding ReLUs and ELUs; and their benefits over Sigmoid/tanh functions
ReLUs were invented in the late 1990s as a solution to the problems caused by the Sigmoid/tanh functions (Hahnloser et al, 2000). They have been found to be useful in larger networks because they mitigate the "vanishing gradient problem", in which the gradient of sigmoid and tanh is disproportionately low at some inputs. Here, we are testing on a four-layer network, so the "vanishing gradient problem" may be present and observable - but it may also be the case that the disproportionately uninformative gradients of sigmoid may lead to less efficient training.



Recent improvements on ReLUs include ELUs, which are the activation function we are using which most closely models the biological activation function. ELUs avoid circumstances where the ReLU fails to ever fire: “For example, a large gradient flowing through a ReLU neuron could cause the weights to update in such a way that the neuron will never activate on any datapoint again ” (Karpathy, 2017; for more detail see Hochreiter, 1998). As ELUs avoid zero-derivative outputs, as shown on the graph above, this is unlikely to happen. ELUs have been shown to be highly successful in deep learning contexts, and we have found they have some benefits in training on Fashion-MNIST with our architecture as well.

Data Pre-processing:

We normalised the data, but otherwise were able to use the PyTorch Fashion-MNIST loader which divides it into 50,000 training samples and 10,000 test samples. The data can also be found at https://github.com/zalandoresearch/fashion-mnist , and the paper makes clear that the original images were converted to PNG, trimmed, resized, sharpened, extended, partially negated, and converted to 8-bit grayscale pixels.

Model Design:
We have chosen a learning rate of 0.01, on the basis that the problem is not overwhelmingly large and more time can be taken on ensuring optimal learning, provided that 0.01 is not too low for the activation functions to learn. Our model is built in PyTorch, using the inbuilt training set (80% of the data) and test set (20% of the data).

Our architecture takes a 28x28 image as an input layer, progressively decreasing over three intermediate/hidden layers (outputs 200, 200, 10) before an output layer of 10. The relevant activation function is the same for all layers (chosen from ELU, ReLU, or Sigmoid).

We train the model in batches of 100, being 50 randomly selected data points without conflict and 50 randomly selected data points with conflict. We selected the optimizer (RPROP), and final activation and loss function (softmax / CrossEntropyLoss) as well-trodden choices in batch-learning and classification problems respectively. Fortunately, ELU, ReLU, and sigmoid activation functions (our independent variables) are built into PyTorch.

Because there is so much data, we did not expect epochs would be necessary, but in light of a lower risk of overfitting from batch training we used two epochs each corresponding to the size of the test set.

Performance Measure:

In an earlier version of this paper, we used ROC curves to determine overall performance, but this is not possible here because it is a multi-class problem. Hence we will be graphing overall accuracy progressively at each batch as a way to compare the learning different activation functions as they cycle through the data. This will give us some insight into the amount of data the different activation functions can learn from. As a final benchmark, we will submit the accuracy after 2 epochs for the best activation function.

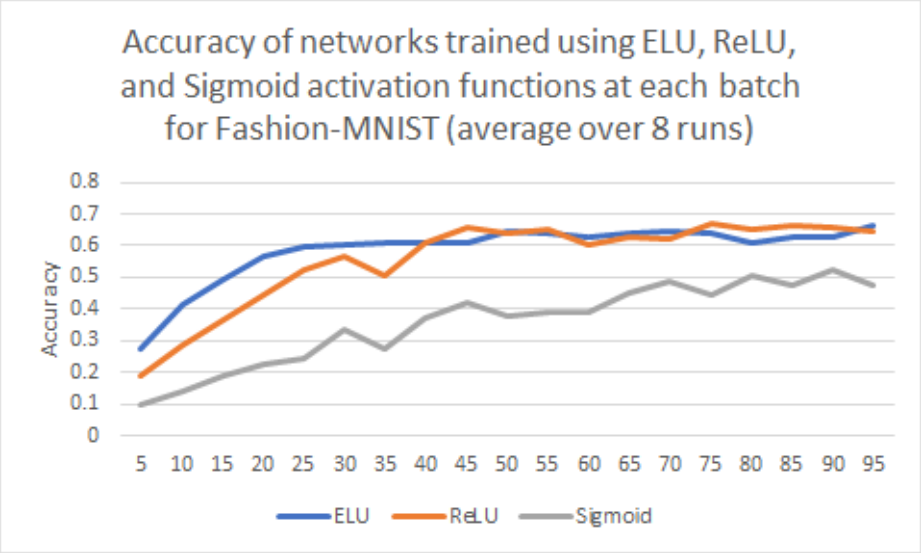
3 Results

In summary, while we did not determine that ELU or RELU led to superior results on a fully trained network, we found better performance by ELU, followed by RELU, over the sigmoid function, when the network was fully trained. Once

the networks were trained over two epochs, the results were as follows (averaged over 8 iterations). The difference is, in our view, not significant.

Sigmoid	ELU	ReLU
0.66233	0.67362	0.65889

Hence, once fully trained, we do not see a substantial difference between the information preserved by the different activation functions. However, following the network through its' early training steps suggests that ELUs are superior to ReLUs, which are in turn superior to sigmoid curves, when data is limited.



Considering that each batch represents 100 samples, ELUs achieved almost triple the performance of sigmoid curves once 2,000 samples had been seen, while ReLUs achieved almost double the performance of sigmoid curves until 3,500 samples had been seen. It is clear from our final results that the sigmoid curve catches up to the ELU and ReLU curves.

4 Discussion

In this paper we had two objectives - to add a missing benchmark to the Fashion-MNIST dataset; and to compare sigmoid functions, ELUs, and ReLUs. We have achieved both objectives. But what is interesting about the Fashion-MNIST dataset is that the relatively large amount of data would have obscured the difference between the performance of the different activation functions. Only when we look at very early on in the training does a difference emerge. The difference between ELUs, ReLUs and sigmoid function' performance could be better demonstrated on datasets with relatively small amounts of labelled data. Additionally, the fact that our network is outperformed by state-of-the-art non-NN models such as a range of SVMs as well as deeper convolutional networks suggests that our benchmark is not of particular use besides contributing to a better understanding of Fashion-MNIST by adding a currently missing benchmark. We were surprised by the relatively poor performance of this NN on the dataset, given that performances of similar architectures on MNIST are much higher, but we are confident in our parameter choice and ultimately believe this reflects substantially on the difficulty of Fashion-MNIST.

4 Conclusion and future work

We have achieved both of our aims and can officially submit our benchmark to Fashion-MNIST as well as conclude that on this dataset ELUs will outperform ReLUs which will outperform sigmoid curves at the start of training; while performance will level out as more batches are learned. This corresponds with earlier work by the author using ANNs for prediction.

In terms of future work, Fashion-MNIST is already well benchmarked in most respects. However, it would be interesting to have a similar comparison of activation functions on this or other datasets using convolutional nets, or using a wider range of algorithms (for example including SELUs and parametric ReLUs).

References

1. Nathaniel, B. King, G. Zeng, L. 2000. Improving quantitative studies of international conflict: A conjecture. American Political Science Review 94(1): 21-35.
2. R Hahnloser, R. Sarpeshkar, M A Mahowald, R. J. Douglas, H.S. Seung (2000). Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. Nature. 405(1): pp. 947–951.
3. Clevert, D. Unterthiner, T. Hochreiter, S. Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs). In: International Conference on Learning Representation (2016).
4. Hochreiter, S. The vanishing gradient problem during learning recurrent neural nets and problem solutions. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 6(2):107–116, 1998.
5. Han Xiao, Kashif Rasul, and Roland Vollgraf. 2017. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747 (2017)
6. Hochreiter, S. The vanishing gradient problem during learning recurrent neural nets and problem solutions. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 6(2):107–116, 1998.
7. LeCun Y, Bengio Y, Hinton GE (2015). “Deep learning”, Nature 521: 436–444.
8. Ludovic Trottier, Philippe Gigu, Brahim Chaib-draa, et al. 2017. Parametric exponential linear unit for deep convolutional neural networks. In Machine Learning and Applications (ICMLA), 2017 16th IEEE International Conference on. IEEE, 207–214
9. Zalandoresearch. 2017. zalandoresearch/fashion-mnist. (Dec 2017). https:// github.com/zalandoresearch/fashion-mnist
10. LeCun, Yann. The MNIST Dataset Of Handwritten Digits (Images). 1999.
11. Glorot, Xavier and Bengio, Yoshua. Understanding the difficulty of training deep feedforward neural networks. In Teh, Yee Whye and Titterington, Mike (eds.), Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, volume 9 of Proceedings of Machine Learning Research, pp. 249–256. PMLR, 13–15 May 2010.
12. Nair, Vinod and Hinton, Geoffrey E. Rectified linear units improve restricted boltzmann machines. pp. 807–A ¸S814. ˇ In Proc. ICML, volume 30, 2010.