PeLU: The Porcelain-emulated Linear Unit

Kartik Chandra and Jeffrey Chang

March 2022

Abstract

Inspired by the nonlinear dynamics of a flush toilet, we introduce a novel activation function for deep learning: the Porcelain-emulated Linear Unit (PeLU). PeLUs allow for cheap, low-power inference on an unconventional hardware target.

1 Introduction

Artificial neural networks were originally inspired by biological neural networks. For example, the structure of modern convolutional neural networks (which have made a big splash in computer vision) was directly inspired by the local and hierarchical nature of the human visual processing system. Even the artificial "neurons" themselves were designed by analogy to biological neurons, borrowing key properties such as the nonlinear response of the activation function.

Another system that was designed by analogy to the biological neuron is the flush toilet. Notice, for example, how both toilets and neurons exhibit all-or-nothing behavior. If the handle of a toilet is depressed slightly, a negligible response is elicited in the bowl, but once the magnitude of the stimulus increases beyond a critical threshold, a large gush of water suddenly exits the tank and flushes the bowl with great force. Immediately following a flush, there is a refractory period wherein further action yields no response; after a minute or so the tank refills and the toilet becomes once again responsive to flushing. (The analogy actually extends much further — for instance, both toilets and neurons exhibit sub-threshold phenomena.)

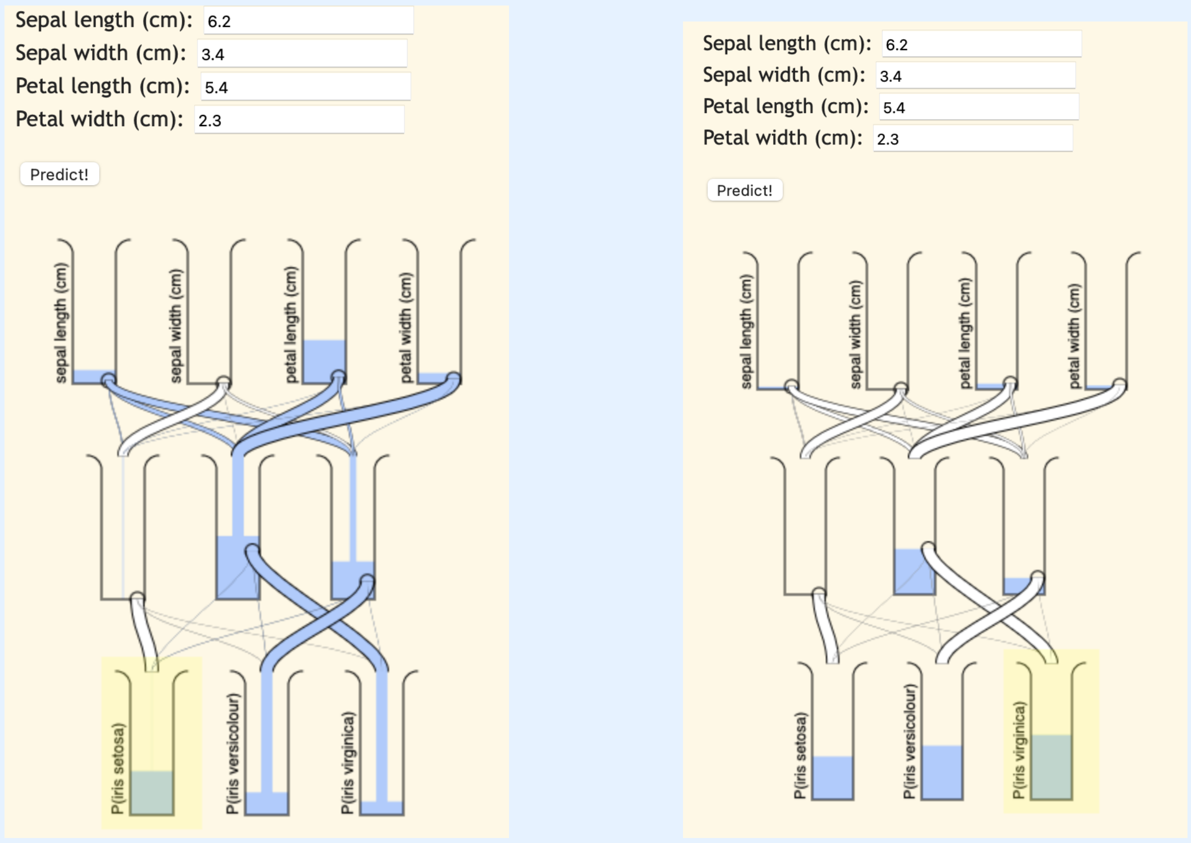
If artifical neurons are like biological neurons and biological neurons are like flush toilets, it is natural to ask if we can transitively implement artificial neurons using plumbing parts. In this paper, we answer this question in the affirmative, opening the floodgates to a new wave of hardware for deep learning.

2 Mathematical description of the PeLU

To get our feet wet with PeLUs, let us consider the following highly simplified model of a flush toilet's nonlinear behavior: a cylindrical bucket with cross-sectional base area B, and a small hole in its wall at height $h \geq 0$. If a volume v of water is poured into the bucket, the water level begins to rise, and if $hB < v$, then some water spills out of the hole. The amount of water that spills out is the output of this system, measured by $\mathrm{ReLU}(v - hB)$. From here on, we will work in units where $B = 1$ so that the B term can be dropped from equations.

We refer to one such bucket as a Porcelain-emulated Linear Unit (PeLU). The output of one PeLU can be distributed into multiple downstream buckets by attaching a branching pipe to the hole at height h; the proportions in which the output is distributed is determined by the relative cross-sectional areas of the pipes. If the areas are A_i such that $\sum_i A_i = 1$, then the amount of water distributed to branch i is $\mathrm{ReLU}(v - h) \cdot A_i$.

Notice that this simplified model of fluid dynamics in the PeLU is differentiable in both h and $\vec{A}$ — hence, those two parameters can be optimized by gradient descent. However, note that if we ever accidentally make h negative, or make the $\vec{A}$ sum to more or less than 1, then the system will become un-physical and we will end up in hot water. To make this robust, therefore, we need to enforce two key constraints: first, that h is posi-


Sepal length (cm): 6.2
Sepal width (cm): 3.4
Petal length (cm): 5.4
Petal width (cm): 2.3

Sepal length (cm): 6.2
Sepal width (cm): 3.4
Petal length (cm): 5.4
Petal width (cm): 2.3

Figure 1: A PeLU-network classifying the Irises dataset, in action (left) and completed (right).

tive, and second, that $\sum_j A_j = 1$. We account for these constraints by reparametrizing: $h = h'^2$ and $\vec{A} = \text{softmax}(\vec{A}')$. The trainable parameters are thus h' and $\vec{A}'$. This completes our description of the PeLU:

$$\text{PeLU}(v, \vec{A}', h') = \text{ReLU}(v - h'^2) \cdot \text{softmax}(\vec{A}')$$

We note that the PeLU unit can have multiple inputs as well, because multiple pipes can empty into the same bucket. The inputs are implicitly summed, because the volumes of water add up inside the bucket.

3 Experiments

We implemented PeLU as a module in PyTorch and used it to build a deep neural Irises classifier with one hidden layer of size 3. Using a standard gradient descent optimizer and training on a TPU (of course), were able to obtain a train error of 2% in a matter of minutes. While we have not yet been able to fabricate the trained network in porcelain,[1] Figure 1 shows a rendered snapshot of our resulting PeLU network running in an in-silico simulation.

4 Conclusion

In this paper we introduced the PeLU, a novel activation function for deep learning. We demonstrated its utility on the Irises dataset (but we emphasize that this is just the tip of the iceberg).

[1]A key challenge in fabricating and operating physical PeLU networks is emptying the buckets in the hidden layers between inference runs (we will cross that bridge when we get there).