Is a single layered ReLu network still a universal approximator?

This question previously had details. They are now in a comment.

Conner Davis, Math PhD candidate
Answered Feb 25 · Upvoted by Robby Goetschalckx, 11+ years as researcher in Machine Learning.

Yes it is!

Technically, “universal approximator” is misleading. I can construct a function that a neural network cannot approximate to arbitrary precision. It was proven to approximate continuous functions to arbitrary accuracy. In reality, a function that is continuous at almost all points can be approximated arbitrarily closely.

Let’s look at why you can approximate any reasonable function

Here we have some arbitrary function. You’ll see below that we can approximate it arbitrarily closely using rectangles.

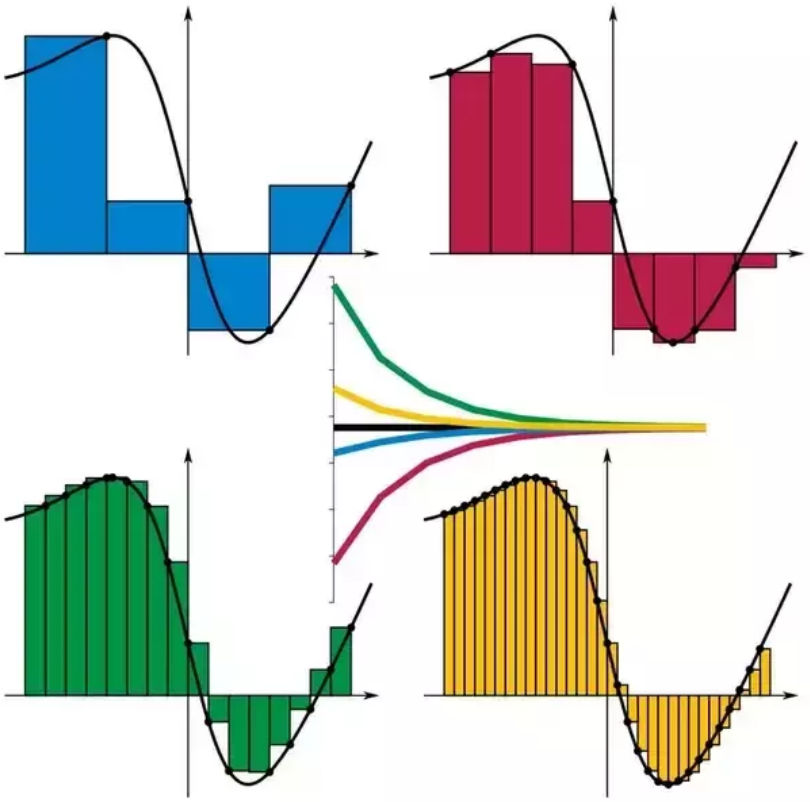
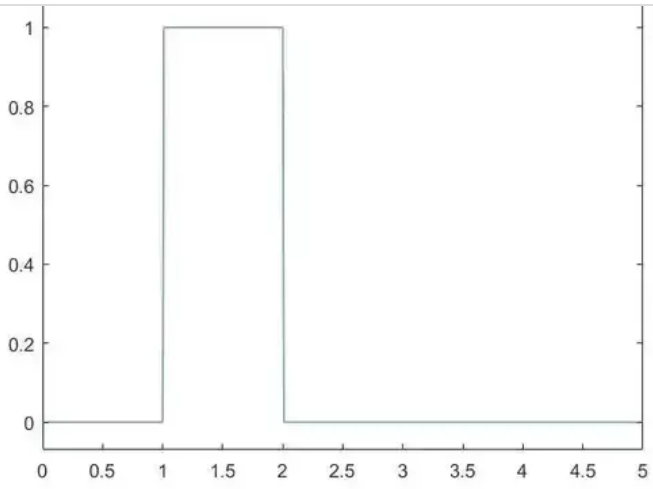


Image from: Riemann sum - Wikipedia

That means that if we can approximate these rectangles arbitrarily closely, we can approximate nearly anything arbitrarily closely.

Oh, look! Here’s one of those rectangles now!



Actually, I lied. That's not a rectangle. That is actually a linear combination of four ReLU activations.

```
x=0:0.01:5;
plot(x,100*(max(0,x-1)-max(0,x-1.01)-(max(0,x-2)-max(0,x-2.01))))
```

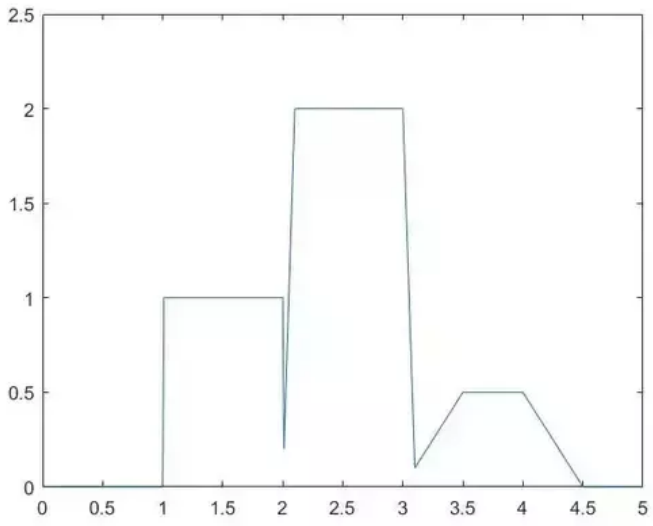
Two lines of MATLAB code is all it takes.

In fact, if you want a rectangle of height h on $[a, b]$, then you can just do the following:

$$\frac{h}{\epsilon} * (max(0, x - a) - max(0, x - (a + \epsilon)) - max(0, x - b) + max(0, x - (b + \epsilon)))$$

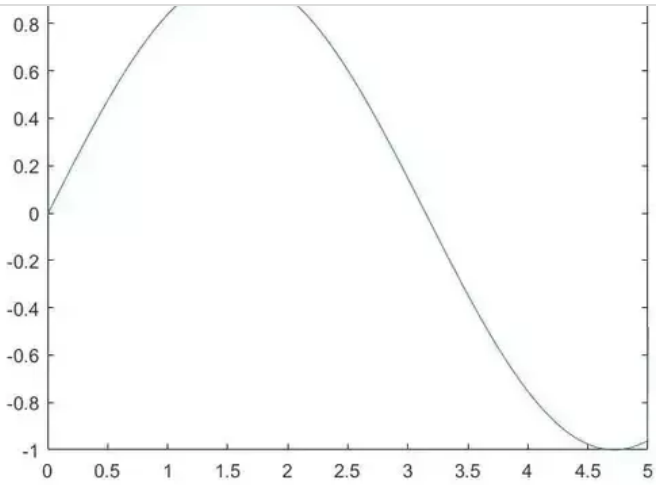
And that's it!

Here you can see that they aren't actually rectangles. For the third one, I made epsilon larger and it made a huge difference.




```
rect = @(h,a,b,epsilon) h/epsilon*(max(0,x-a)-max(0,x-(a+epsilon))-max(0
plot(x, rect(1,1,2,.01) + rect(2,2,3,0.1) + rect(0.5,3,4,0.5))
```

Watch, we can even approximate $\sin(x)$



I lied again. That **WAS** the approximation. I used a lot of ReLUs to get it and it's far nicer than it needs to be, but that's exactly the point of arbitrary approximation.

The MATLAB code for the fake $\sin(x)$

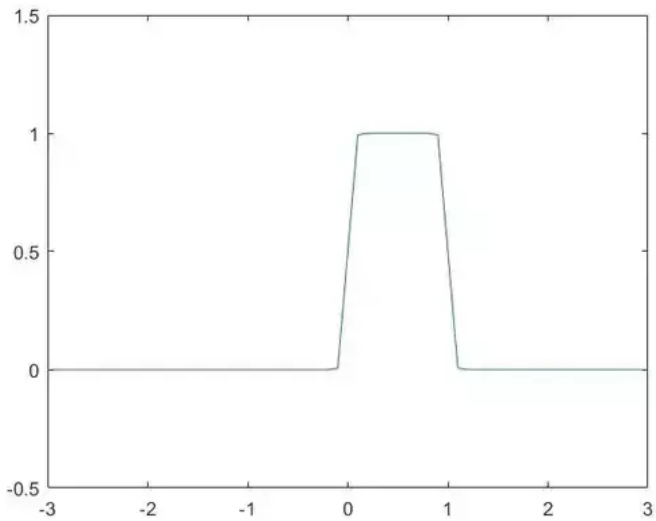
```
x=0:0.01:5;
z=0:0.001:5;
sin_approx = zeros(size(z));
rect = @(h,a,b,epsilon) h/epsilon*(max(0,z-a)-max(0,z-(a+epsilon))-max(0
for i=2:(numel(x)-1)
    sin_approx = sin_approx + rect(sin(x(i)),(x(i)+x(i-1))/2, (x(i)+x(i+
end
plot(z,sin_approx)
```

WAIT! But you've only been using a single variable! My dataset has thousands of them!

That's true, but it's actually not hard to extend these same results into arbitrarily many dimensions. You just acknowledge that rectangular prisms now do the job of rectangles and show you can approximate those arbitrarily closely.

I'd love to provide pretty pictures for those, but my 253453-dimensional rendering software is broken. I'll update this answer when it's fixed.

Oh, and a bonus:



We needed four ReLUs, but this was made with only two sigmoids. That does NOT mean that sigmoid is a better activation function. If your network is actually doing the approximation this way, then it won't generalize well. In practice, your network

```
4  ylim([-0.5,1.5])
```

notice the '50's? Make those bigger to get a better approximation of the rectangle.

2.5k Views · 69 Upvotes · Answer requested by Brando Miranda

Sahil Juneja
So what significance does this have when working with NNs or DL models?

1 more comments from Yevgeniy Grechka

View 2 Other Answers to this Question

About the Author

Conner Davis

Researcher at The University of Texas at Dallas

PhD student Mathematics, The University of Texas at Dallas

Lives in The United States of America

6.1m answer views
94.4k this month

Follow 6.6k Turn On Notifications

More Answers from Conner Davis

View More

What are some pranks that don't hurt anyone?
321.1k Views

Have you ever scammed a scammer?
243.4k Views

How would you explain to a young child that they are dying?
9k Views

What is the most ridiculous thing that you have ever convinced someone to believe?
13.8k Views

What is an insane senior prank you took part in during high school?
219.8k Views