

Elyanah

Tom Walsh, NYC Data Science Academy



Elixir

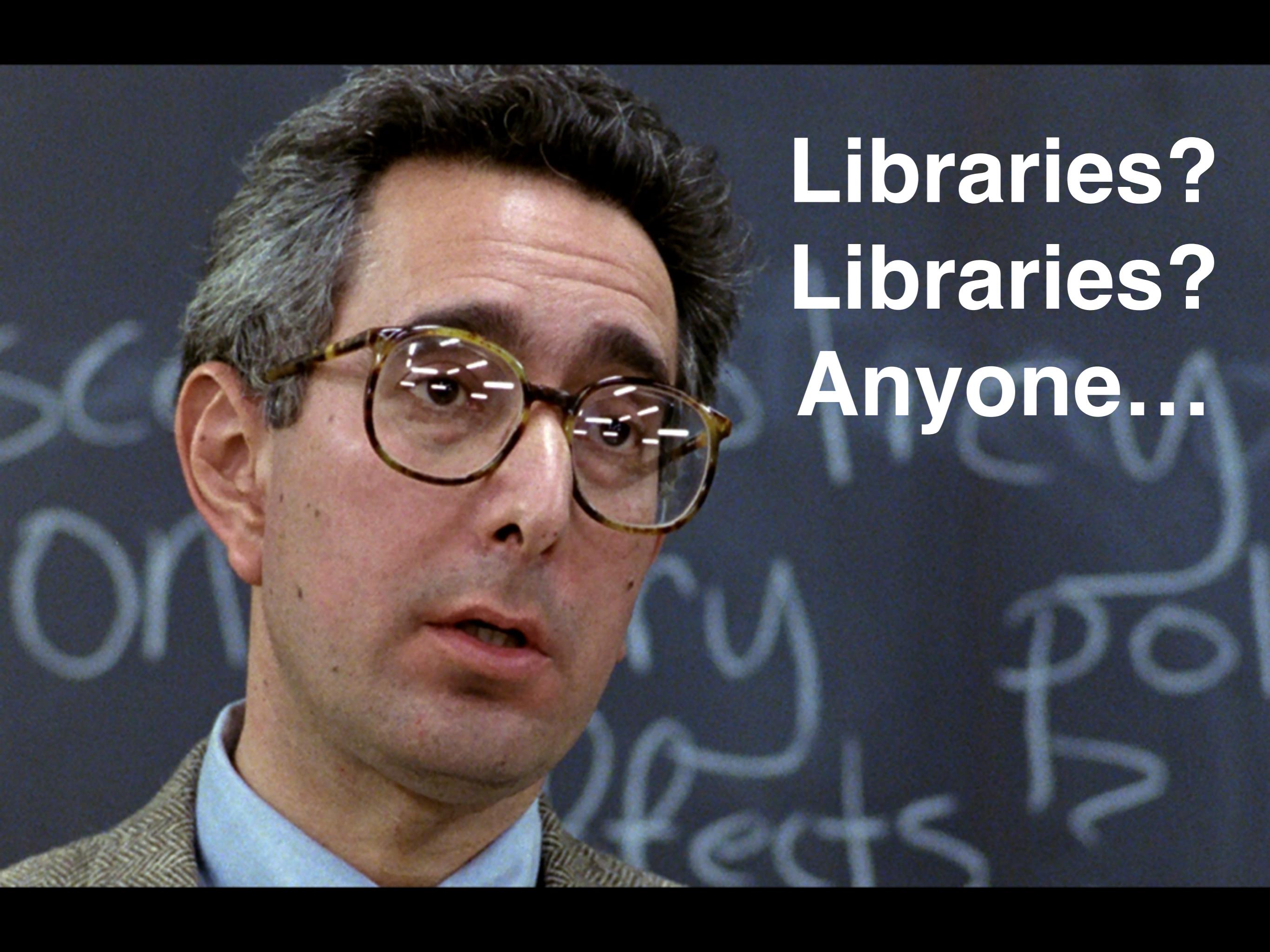
- Scalable
 - Agents, Tasks, Servers
 - Lightweight Processes
- Fault Tolerant
 - Supervisors
- Functional
 - Immutable State
 - Pattern Matching
 - Data Flow with Pipes

“Thus, there’s a lot of energy devoted to creating the Next Big Thing; the framework that gives us what we love about Rails; the language that enables it like Ruby did while addressing their shortcomings and setting us up for the future.

That framework is Phoenix and that language is Elixir.”

— Ken Miller

<https://medium.com/infinite-red/phoenix-is-rails-5-f6d28e57395>

A close-up photograph of Michael Scott, played by Steve Carell, from the television show "The Office". He is wearing his signature round, brown-rimmed glasses and has a look of surprise or confusion on his face. His hair is slightly messy, and he is wearing a light blue button-down shirt under a textured, olive-green sweater. The background is dark and out of focus, showing some faint, glowing blue text that appears to be part of a screen or a wall.

Libraries?
Libraries?
Anyone...

“Elixir is only a few years old. On the one hand this means that there are not many quantitative libraries - but it also means you can get in on the ground floor.

How about being the guy who kicks off 'Elixir Pandas', a statistics or machine learning library?

In this environment a little can go a surprisingly long way.”

–John Orford

<http://blog.johnorford.com/2015/11/01/x-reasons-to-use-elixir-in-finance/>

Elyanah

A Very Early-Stage Machine Learning Library for Elixir

Neural Networks are Exciting...

...and I would like to understand how they work.

Design

- Process layers sequentially
- Process neurons within a layer in parallel
- Arbitrary numbers of layers, neurons.
- No skipping layers.
- Automatically convert factors
- Randomize initial weights intelligently
- Try not to **** up backpropagation

Usage

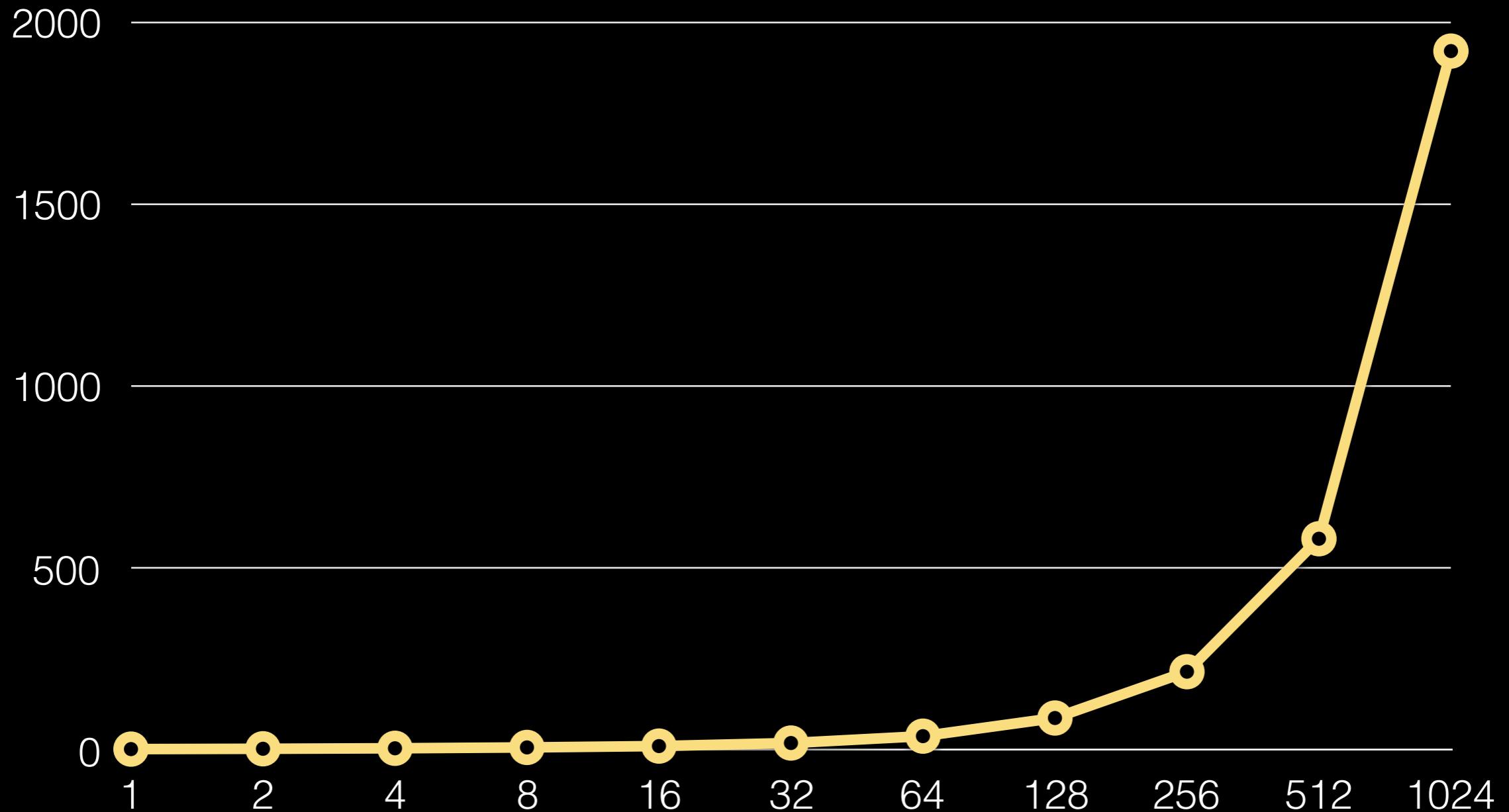
```
{:ok, net} = GenServer.start_link(Elyanah.NeuralNet, [5,5,1])
```

```
[prediction] = GenServer.call(net, {:predict, input})  
GenServer.call(net, {:backprop, output, eta})
```

Does it Work?

[1,0] vs. [0,1]	YES
random permutations of 10 variables	YES
Given a word, is the word count even or odd	NO
Predicting NBA shots	YES

Scalability?



Time in seconds to train 10000 observations of 10 variables as a function of the number of neurons in the single hidden layer on a 2014 Macbook Air.

Next Steps

- Code cleanup
- Supervision Tree
- Documentation
- Test cases
- Read more so I actually know what I'm doing
- Implement other statistics/machine learning methods
- Use variables in back propagation that make sense to people
- Agents vs. servers?
- Escape from local minimums?
- Other activation functions?
- More flexible topology?
- Specifying initial weights?
- Inspection?
- Cloning and persistence?
- Search for params/topology?