

Elyanah.Numeric

An Elixir Linear Algebra Library

Last Episode...

- Neural Network was a fun proof and concept, but...
- It was pretty rudimentary:
 - Limited options.
- It wasn't much use on its own.
- It was a bit hastily constructed:
 - No documentation.
 - No test cases.

Started from the Bottom...

- Just as NumPy forms the foundation for Pandas and scikit-learn, the goal is to build a foundation I can build upon.
- Some major differences from NumPy
 - Entirely in Elixir (no native code)
 - No new classes/structs (operate on Elixir Lists).

...Now We Here

<http://allintheeyes.github.io/elyanah>

```

def multiply([[_|_] | _] = a, [[_|_] = h | _] = b) do
  for aa <- a, bb <- transpose(b) do
    Array.dot(aa, bb)
  end
  |> Enum.chunk(length(h))
end
def multiply([_|_] = a, [[_|_] | _] = b), do: multiply([a], b)
def multiply([[_|_] | _] = a, [_|_] = b), do: multiply(a, transpose([b]))
def multiply(a, [[_|_] | _] = b) do
  Enum.map(b, &(Array.multiply(a, &1)))
end
def multiply([[_|_] | _] = a, b), do: multiply(b, a)

```

```
iex(1)> alias Elyanah.Numeric.Matrix, as: Matrix  
nil  
iex(2)> Matrix.multiply([[1,2],[3,4],[5,6]], [[1,2,3],[4,5,6]])  
[[9, 12, 15], [19, 26, 33], [29, 40, 51]]
```

That's cool and all, but we can do better...

```
iex(3)> use Elyanah.Numeric  
nil  
iex(4)> [[1,2],[3,4],[5,6]] * [[1,2,3],[4,5,6]]  
[[9, 12, 15], [19, 26, 33], [29, 40, 51]]
```

Infix Operators

```
def ([[_|_] | _] = a) * b, do: Matrix.multiply(a, b)
def a * ([[_|_] | _] = b), do: Matrix.multiply(a, b)
def ([_|_] = a) * b, do: Array.multiply(a, b)
def a * ([_|_] = b), do: Array.multiply(a, b)
def a * b, do: Kernel.*(a,b)

defmacro __using__(_opts) do
  quote do
    import Kernel, except: [*: 2, +: 2, -: 2, /: 2]
    import Elyanah.Numeric
  end
end
```

Supported Operations

- Array Operations
- Matrix Operations
- Infix Operators

Continuing Work

- Creating a DataFrame equivalent
 - Support for row-wise and column-wise data storage
 - Support for row-wise and column-wise data manipulation
 - Stream-based data transformation for model inputs.