

Matlab mode

in  My Hub

Notebook dependencies and setup

Evaluated

Module wrapper

```
1 # defmodule Foo do
2 #   import Nx.Defn
3
4 #   defn run() do
```

Evaluated

nil

```
1 # m = 5
2 # x = Nx.tensor([1,2,3])
3 # b = Nx.tensor([4,5,6])
4
5 # (m * x) + b
```

Evaluated

nil

```
1 #   end
2 # end
3 # Foo.run()
```

Evaluated

nil

Nx Operators

```
1 use Nx.Defn.Kernel
```

Evaluated

Nx.Defn.Kernel

```
1 # defmodule Foo do
2 #   import Nx.Defn
3
```

```

4  # defn run() do
5  #     m = 5
6  #     x = Nx.tensor([1,2,3])
7  #     b = Nx.tensor([4,5,6])
8  #     (m * x) + b
9
10 #     # Nx.dot(x, Nx.transpose(b))
11 #     # (x == 2)
12 # end
13 # end
14 # Foo.run()

```

Evaluated

```
nil
```

```

1  # import Nx.Defn
2
3  m = 5
4  x = Nx.tensor([1,2,3])
5  b = Nx.tensor([4,5,6])
6
7  (m * x) + b

```

Evaluated

```

#Nx.Tensor<
s64[3]
[9, 15, 21]
>

```

```

1  m = 5
2  x = Nx.tensor([1,2,3])
3  b = Nx.tensor([4,5,6])
4
5  Nx.multiply(m, x)
6  |> Nx.add(b)

```

Evaluated

```

#Nx.Tensor<
s64[3]
[9, 15, 21]
>

```

```

1  mA = ( Nx.Random.key(42)
2      |> Nx.Random.randint(0,5, shape: {5,5})
3      |> elem(0)
4      ) [[..., ...]]
5
6  mB = ( Nx.Random.key(4)
7      |> Nx.Random.randint(0,5, shape: {5,5})
8      |> elem(0)
9      ) [[..., ...]]

```

```

9      , [L..., ...]]
10
11
12     xVals = 0..5 |> Enum.to_list |> Nx.tensor()
13     yVals = Nx.dot(mA, mB)[[0, ...]]

```

Evaluated

```

#Nx.Tensor<
  s64[5]
  [6, 8, 18, 10, 12]
>

```

```

1  my_data = %{
2    a: xVals |> Nx.to_list,
3    b: yVals |> Nx.to_list
4  }

```

Evaluated

```
%{a: [0, 1, 2, 3, 4, 5], b: [6, 8, 18, 10, 12]}
```

Evaluated

```

1  # random matrix
2  # increasing value matrix
3  # identity matrix
4  # slicing
5  mA = (Nx.Random.key(42)

```

```
6     |> Nx.Random.randint(0,5, shape: {5,5})
7     |> elem(0)
8     )[[..., ...]]
9
10    (
11      Nx.iota({5,5}, axis: 0) + Nx.iota({5,5}, axis: 1) + (10 * Nx.eye({5,5}))
12    )[[..., ...]]
```

Evaluated

#Nx.Tensor<

s64[5][5]

[

[10, 1, 2, 3, 4],

[1, 12, 3, 4, 5],

[2, 3, 14, 5, 6],

[3, 4, 5, 16, 7],

[4, 5, 6, 7, 18]

]

>

```
1  # get color channel
2  # crop an image
3  {:ok, img} = StbImage.read_file("/Users/davidsarmaholdings/Dropbox/Public,
4
5  (img |> StbImage.to_nx())[[0..400, 0..300, 0..0]]
6  |> StbImage.from_nx()
```

Evaluated

Image Raw