

# Examine saved models

---

The purpose of this notebook is to examine the learned representations and accuracies of saved models.

## Setup

---

### Add NeuralODE code

```
train (generic function with 1 method)
```

### Load model

---

```
loadModel (generic function with 1 method)
```

```
(Chain(
  layer 1 = Scale((100.)).           # 100 parameters
```

### Load data

---

```
(ArrayAndFuncs([1.0, 1.0, 1.0, 1.0, 1.0,   more ,1.0], [Interpolate([   more], [   more
```

### Add display method for Interpolate

```
1 function (itp::Interpolate)(t::Float64)
2     i = searchsortedfirst(itp.locations, t)
3     @inbounds itp.SET[i] - 1
4 end
```

## Test model

---

## Training data

1.0

## Testing data

1.0

## What is the testing SET number?

---

SET\_num = 12

```
1 SET_num = 12
```

## Visualize low-dimensional dynamics

---

input\_to\_mat (generic function with 1 method)

This is data.

```

1 begin
2     time_range = 0.01f0:0.01f0:0.50f0
3     input = input_to_mat(reduce(vcat, testing_input_funcs[SET_num].
4         (0.01:0.01:0.50)))
5     (y_out_train, r_out_train), st_out = model(training_data[1], ps, st)
6     (y_out_test, r_out_test), st_out = model(testing_data[1], ps, st)
7     cat_r_out = reduce(hcat, r_out_train[:, :, i] for i in 1:size(r_out_train)[3])
8     M = fit(PCA, cat_r_out;)
9     pc_rates = predict(M, r_out_test[:, :, SET_num])
10    y = y_out_test[1, :, SET_num]
11    df_rates = DataFrame(
12        time = time_range,
13        out = y,
14        pc_1= pc_rates[1, :],
15        pc_2 = pc_rates[2, :],
16        pc_3 = pc_rates[3, :],
17        pc_4 = pc_rates[4, :],
18        pc_5 = pc_rates[5, :],
19    )
20    md"This is data."
21 end

```













