

```
In [ ]: from machine_learning_functions import *
        from itertools import product
```

```
In [ ]: def experiment_linear(a, b, learning_rate, epochs, random_x_function):
        a, b = float(a), float(b)
        model = Model(
            FFN = FFN(
                neurons_per_layer_list=[1, 1],
                activation_functions_list=[None,],
                cost_function=MSE()
            ),
            data_set=create_1_input_1_output_XY_data(
                function=lambda x: a*x+b,
                num_data_items=10000,
                random_x_function=random_x_function
            )
        )
        mean_cost, variance_cost = model.train_and_evaluate(
            learning_rate=learning_rate,
            epochs=epochs,
            batch_size=50
        )
        print(f"Experiment results were: mean_cost={mean_cost:.4f} with varian

        model.print_FFN_parameters()
        return model
```

```
In [ ]: def experiment_generator_factory_linear(coefficients, epochs_set, learning_rate_
        for (a, b), learning_rate, epochs, x_interval in product(coefficients, learn
            parameters = {
                "coefficients": (a, b),
                "learning_rate": learning_rate,
                "epochs": epochs,
                "x_interval": x_interval,
            }
            def experiment():
                experiment_linear(
                    a=a, b=b,
                    learning_rate=learning_rate,
                    epochs=epochs,
                    random_x_function=lambda: random.uniform(*x_interval)
                )
            yield experiment, parameters
```

```
In [ ]: experiemnt_generator = experiment_generator_factory_linear(
        coefficients=(
            (2, 5),
            (-10, 6),
            (1/2, -5/4),
            (100, -50),
            (1/100, 20)
        ),
        epochs_set=(
            # 1, 5, 10, 20,
            5,
        ),
        learning_rate_set=(
```

```

        # 10**-2, 10**-3, 10**-4, 10**-5
        10**-2, 10**-3, 10**-4,
    ),
    x_intervals = (
        (-1, 1),
        (-10, 10),
        (-100, 100),
    ),
)

for i, (experiment, parameters) in enumerate(experiemnt_generator):
    print(f"Experiment {i} with parameters:")
    print(parameters)
    experiment()
    print("\n")

```

Experiment 0 with parameters:

```
{'coefficients': (2, 5), 'learning_rate': 0.01, 'epochs': 5, 'x_interval': (-1, 1)}
```

Experiment results were: mean_cost=0.0298 with variance=0.0007

Parameters of network

```
{'W1': array([[1.70981529]])}
```

```
{'B1': array([4.97257316])}
```

Experiment 1 with parameters:

```
{'coefficients': (2, 5), 'learning_rate': 0.01, 'epochs': 5, 'x_interval': (-10, 10)}
```

Experiment results were: mean_cost=0.0015 with variance=0.0000

Parameters of network

```
{'W1': array([[2.00041316]])}
```

```
{'B1': array([4.96104802])}
```

Experiment 2 with parameters:

```
{'coefficients': (2, 5), 'learning_rate': 0.01, 'epochs': 5, 'x_interval': (-100, 100)}
```

c:\Users\Henry\Documents\comp sci presentations\machine_learning_functions.py:332:

RuntimeWarning: overflow encountered in scalar add

```
total_cost += cost
```

c:\Users\Henry\Documents\comp sci presentations\machine_learning_functions.py:155:

RuntimeWarning: overflow encountered in matmul

```
dcdAp = dZdAp.T @ dcdZ
```

c:\Users\Henry\Documents\comp sci presentations\machine_learning_functions.py:343:

RuntimeWarning: invalid value encountered in scalar subtract

```
loss_change = new_loss - old_loss
```

c:\Users\Henry\Documents\comp sci presentations\machine_learning_functions.py:339:

RuntimeWarning: overflow encountered in add

```
total_param_cost_gradients[param_name] += param_gradients[param_name]
```

c:\Users\Henry\Documents\comp sci presentations\machine_learning_functions.py:339:

RuntimeWarning: invalid value encountered in add

```
total_param_cost_gradients[param_name] += param_gradients[param_name]
```

c:\Users\Henry\Documents\comp sci presentations\machine_learning_functions.py:167:

RuntimeWarning: invalid value encountered in add

```
self.weights += weights_change
```

```
Experiment results were: mean_cost=nan with variance=nan
Parameters of network
{'W1': array([[nan]])}
{'B1': array([nan])}
```

Experiment 3 with parameters:

```
{'coefficients': (2, 5), 'learning_rate': 0.001, 'epochs': 5, 'x_interval': (-1, 1)}
```

```
Experiment results were: mean_cost=8.5479 with variance=20.9740
```

Parameters of network

```
{'W1': array([[0.57836567]])}
```

```
{'B1': array([2.20027431])}
```

Experiment 4 with parameters:

```
{'coefficients': (2, 5), 'learning_rate': 0.001, 'epochs': 5, 'x_interval': (-10, 10)}
```

```
Experiment results were: mean_cost=7.8882 with variance=0.0048
```

Parameters of network

```
{'W1': array([[2.00218823]])}
```

```
{'B1': array([2.1908033])}
```

Experiment 5 with parameters:

```
{'coefficients': (2, 5), 'learning_rate': 0.001, 'epochs': 5, 'x_interval': (-10, 100)}
```

```
c:\Users\Henry\Documents\compsci presentations\machine_learning_functions.py:387:
RuntimeWarning: invalid value encountered in scalar subtract
  (sum(cost**2 for cost in costs) / self.num_test_data_items)
```

Experiment results were: mean_cost=inf with variance=nan
 Parameters of network
 {'W1': array([[-1.98234099e+181]])}
 {'B1': array([3.15438628e+178])}

Experiment 6 with parameters:
 {'coefficients': (2, 5), 'learning_rate': 0.0001, 'epochs': 5, 'x_interval': (-1, 1)}
 Experiment results were: mean_cost=18.3241 with variance=25.0563
 Parameters of network
 {'W1': array([[0.99568691]])}
 {'B1': array([0.70627023])}

Experiment 7 with parameters:
 {'coefficients': (2, 5), 'learning_rate': 0.0001, 'epochs': 5, 'x_interval': (-10, 10)}
 Experiment results were: mean_cost=19.7214 with variance=139.0039
 Parameters of network
 {'W1': array([[1.73973192]])}
 {'B1': array([0.70787085])}

Experiment 8 with parameters:
 {'coefficients': (2, 5), 'learning_rate': 0.0001, 'epochs': 5, 'x_interval': (-100, 100)}
 Experiment results were: mean_cost=19.4862 with variance=45.5783
 Parameters of network
 {'W1': array([[1.98738495]])}
 {'B1': array([0.70859671])}

Experiment 9 with parameters:
 {'coefficients': (-10, 6), 'learning_rate': 0.01, 'epochs': 5, 'x_interval': (-1, 1)}
 Experiment results were: mean_cost=1.4818 with variance=1.5737
 Parameters of network
 {'W1': array([[-7.92298175]])}
 {'B1': array([5.95755549])}

Experiment 10 with parameters:
 {'coefficients': (-10, 6), 'learning_rate': 0.01, 'epochs': 5, 'x_interval': (-10, 10)}
 Experiment results were: mean_cost=0.0022 with variance=0.0000
 Parameters of network
 {'W1': array([[-9.99867146]])}
 {'B1': array([5.9535317])}

Experiment 11 with parameters:
 {'coefficients': (-10, 6), 'learning_rate': 0.01, 'epochs': 5, 'x_interval': (-100, 100)}

c:\Users\Henry\AppData\Local\Programs\Python\Python311\Lib\site-packages\numpy\core\numeric.py:925: RuntimeWarning: overflow encountered in multiply
 return multiply(a.ravel()[:, newaxis], b.ravel()[newaxis, :], out)

Experiment results were: mean_cost=nan with variance=nan
Parameters of network
{'W1': array([[nan]])}
{'B1': array([nan])}

Experiment 12 with parameters:

{'coefficients': (-10, 6), 'learning_rate': 0.001, 'epochs': 5, 'x_interval': (-1, 1)}

Experiment results were: mean_cost=36.9090 with variance=1670.5115

Parameters of network

{'W1': array([[-1.64173433]])}

{'B1': array([2.55751786])}

Experiment 13 with parameters:

{'coefficients': (-10, 6), 'learning_rate': 0.001, 'epochs': 5, 'x_interval': (-10, 10)}

Experiment results were: mean_cost=11.8649 with variance=0.6781

Parameters of network

{'W1': array([[-9.97780547]])}

{'B1': array([2.54745428])}

Experiment 14 with parameters:

{'coefficients': (-10, 6), 'learning_rate': 0.001, 'epochs': 5, 'x_interval': (-100, 100)}

Experiment results were: mean_cost=inf with variance=nan

Parameters of network

{'W1': array([[4.60613373e+17]])}

{'B1': array([1.45117871e+175])}

Experiment 15 with parameters:

{'coefficients': (-10, 6), 'learning_rate': 0.0001, 'epochs': 5, 'x_interval': (-1, 1)}

Experiment results were: mean_cost=55.8378 with variance=3674.3449

Parameters of network

{'W1': array([[-0.81450249]])}

{'B1': array([0.74585777])}

Experiment 16 with parameters:

{'coefficients': (-10, 6), 'learning_rate': 0.0001, 'epochs': 5, 'x_interval': (-10, 10)}

Experiment results were: mean_cost=211.1997 with variance=41443.1945

Parameters of network

{'W1': array([[-7.72827929]])}

{'B1': array([0.73176144])}

Experiment 17 with parameters:

{'coefficients': (-10, 6), 'learning_rate': 0.0001, 'epochs': 5, 'x_interval': (-100, 100)}

Experiment results were: mean_cost=27.4441 with variance=0.8080

Parameters of network

{'W1': array([[-9.99850079]])}

{'B1': array([0.76852942])}

Experiment 18 with parameters:
{'coefficients': (0.5, -1.25), 'learning_rate': 0.01, 'epochs': 5, 'x_interval': (-1, 1)}
Experiment results were: mean_cost=0.0066 with variance=0.0000
Parameters of network
{'W1': array([[0.36284787]])}
{'B1': array([-1.23609084])}

Experiment 19 with parameters:
{'coefficients': (0.5, -1.25), 'learning_rate': 0.01, 'epochs': 5, 'x_interval': (-10, 10)}
Experiment results were: mean_cost=0.0002 with variance=0.0000
Parameters of network
{'W1': array([[0.49959156]])}
{'B1': array([-1.23533])}

Experiment 20 with parameters:
{'coefficients': (0.5, -1.25), 'learning_rate': 0.01, 'epochs': 5, 'x_interval': (-100, 100)}
Experiment results were: mean_cost=nan with variance=nan
Parameters of network
{'W1': array([[nan]])}
{'B1': array([nan])}

Experiment 21 with parameters:
{'coefficients': (0.5, -1.25), 'learning_rate': 0.001, 'epochs': 5, 'x_interval': (-1, 1)}
Experiment results were: mean_cost=1.4286 with variance=1.1980
Parameters of network
{'W1': array([[-0.42958543]])}
{'B1': array([-0.16960566])}

Experiment 22 with parameters:
{'coefficients': (0.5, -1.25), 'learning_rate': 0.001, 'epochs': 5, 'x_interval': (-10, 10)}
Experiment results were: mean_cost=1.2213 with variance=0.0142
Parameters of network
{'W1': array([[0.49053729]])}
{'B1': array([-0.15060354])}

Experiment 23 with parameters:
{'coefficients': (0.5, -1.25), 'learning_rate': 0.001, 'epochs': 5, 'x_interval': (-100, 100)}
Experiment results were: mean_cost=inf with variance=nan
Parameters of network
{'W1': array([[-8.5415591e+178]])}
{'B1': array([-3.42490362e+175])}

Experiment 24 with parameters:
{'coefficients': (0.5, -1.25), 'learning_rate': 0.0001, 'epochs': 5, 'x_interval': (-1, 1)}
Experiment results were: mean_cost=3.3377 with variance=3.6916
Parameters of network
{'W1': array([[-0.46912092]])}

```
{'B1': array([0.41923213])}
```

Experiment 25 with parameters:

```
{'coefficients': (0.5, -1.25), 'learning_rate': 0.0001, 'epochs': 5, 'x_interval': (-10, 10)}
```

Experiment results were: mean_cost=2.7869 with variance=0.0074

Parameters of network

```
{'W1': array([[0.50453547]])}
```

```
{'B1': array([0.41959838])}
```

Experiment 26 with parameters:

```
{'coefficients': (0.5, -1.25), 'learning_rate': 0.0001, 'epochs': 5, 'x_interval': (-100, 100)}
```

Experiment results were: mean_cost=2.7887 with variance=0.3421

Parameters of network

```
{'W1': array([[0.49687399]])}
```

```
{'B1': array([0.42012712])}
```

Experiment 27 with parameters:

```
{'coefficients': (100, -50), 'learning_rate': 0.01, 'epochs': 5, 'x_interval': (-1, 1)}
```

Experiment results were: mean_cost=163.6281 with variance=20009.9989

Parameters of network

```
{'W1': array([[79.01480757]])}
```

```
{'B1': array([-49.7440074])}
```

Experiment 28 with parameters:

```
{'coefficients': (100, -50), 'learning_rate': 0.01, 'epochs': 5, 'x_interval': (-10, 10)}
```

Experiment results were: mean_cost=0.1705 with variance=0.0004

Parameters of network

```
{'W1': array([[100.00435216]])}
```

```
{'B1': array([-49.59025883])}
```

Experiment 29 with parameters:

```
{'coefficients': (100, -50), 'learning_rate': 0.01, 'epochs': 5, 'x_interval': (-100, 100)}
```

Experiment results were: mean_cost=nan with variance=nan

Parameters of network

```
{'W1': array([[nan]])}
```

```
{'B1': array([nan])}
```

Experiment 30 with parameters:

```
{'coefficients': (100, -50), 'learning_rate': 0.001, 'epochs': 5, 'x_interval': (-1, 1)}
```

Experiment results were: mean_cost=3215.8271 with variance=12306704.7825

Parameters of network

```
{'W1': array([[15.62694879]])}
```

```
{'B1': array([-19.10231304])}
```

Experiment 31 with parameters:

```
{'coefficients': (100, -50), 'learning_rate': 0.001, 'epochs': 5, 'x_interval': (-10, 10)}
```

Experiment results were: mean_cost=983.0724 with variance=673.5098
Parameters of network
{'W1': array([[100.07108071]])}
{'B1': array([-18.63668096])}

Experiment 32 with parameters:
{'coefficients': (100, -50), 'learning_rate': 0.001, 'epochs': 5, 'x_interval': (-100, 100)}
Experiment results were: mean_cost=inf with variance=nan
Parameters of network
{'W1': array([[-2.50844807e+178]])}
{'B1': array([1.65126148e+175])}

Experiment 33 with parameters:
{'coefficients': (100, -50), 'learning_rate': 0.0001, 'epochs': 5, 'x_interval': (-1, 1)}
Experiment results were: mean_cost=5203.2417 with variance=35943051.9581
Parameters of network
{'W1': array([[1.89894827]])}
{'B1': array([-1.77763419])}

Experiment 34 with parameters:
{'coefficients': (100, -50), 'learning_rate': 0.0001, 'epochs': 5, 'x_interval': (-10, 10)}
Experiment results were: mean_cost=16986.2349 with variance=276342240.9549
Parameters of network
{'W1': array([[79.43943908]])}
{'B1': array([-1.94710922])}

Experiment 35 with parameters:
{'coefficients': (100, -50), 'learning_rate': 0.0001, 'epochs': 5, 'x_interval': (-100, 100)}
Experiment results were: mean_cost=2340.6108 with variance=35635.1680
Parameters of network
{'W1': array([[99.96667362]])}
{'B1': array([-1.55094823])}

Experiment 36 with parameters:
{'coefficients': (0.01, 20), 'learning_rate': 0.01, 'epochs': 5, 'x_interval': (-1, 1)}
Experiment results were: mean_cost=0.0550 with variance=0.0030
Parameters of network
{'W1': array([[-0.23203314]])}
{'B1': array([19.81419204])}

Experiment 37 with parameters:
{'coefficients': (0.01, 20), 'learning_rate': 0.01, 'epochs': 5, 'x_interval': (-10, 10)}
Experiment results were: mean_cost=0.0281 with variance=0.0000
Parameters of network
{'W1': array([[0.008139]])}
{'B1': array([19.83138432])}

Experiment 38 with parameters:

```
{'coefficients': (0.01, 20), 'learning_rate': 0.01, 'epochs': 5, 'x_interval': (-100, 100)}
```

Experiment results were: mean_cost=nan with variance=nan

Parameters of network

```
{'W1': array([[nan]])}
```

```
{'B1': array([nan])}
```

Experiment 39 with parameters:

```
{'coefficients': (0.01, 20), 'learning_rate': 0.001, 'epochs': 5, 'x_interval': (-1, 1)}
```

Experiment results were: mean_cost=148.4125 with variance=20.2849

Parameters of network

```
{'W1': array([[ -0.30087084]])}
```

```
{'B1': array([7.82555553])}
```

Experiment 40 with parameters:

```
{'coefficients': (0.01, 20), 'learning_rate': 0.001, 'epochs': 5, 'x_interval': (-10, 10)}
```

Experiment results were: mean_cost=148.3524 with variance=12.3900

Parameters of network

```
{'W1': array([[0.03505832]])}
```

```
{'B1': array([7.82765982])}
```

Experiment 41 with parameters:

```
{'coefficients': (0.01, 20), 'learning_rate': 0.001, 'epochs': 5, 'x_interval': (-100, 100)}
```

Experiment results were: mean_cost=inf with variance=nan

Parameters of network

```
{'W1': array([[ -6.42619083e+176]])}
```

```
{'B1': array([3.83397167e+173])}
```

Experiment 42 with parameters:

```
{'coefficients': (0.01, 20), 'learning_rate': 0.0001, 'epochs': 5, 'x_interval': (-1, 1)}
```

Experiment results were: mean_cost=345.5897 with variance=84.4915

Parameters of network

```
{'W1': array([[ -0.42686317]])}
```

```
{'B1': array([1.39562938])}
```

Experiment 43 with parameters:

```
{'coefficients': (0.01, 20), 'learning_rate': 0.0001, 'epochs': 5, 'x_interval': (-10, 10)}
```

Experiment results were: mean_cost=350.1916 with variance=1515.3733

Parameters of network

```
{'W1': array([[ -0.18214431]])}
```

```
{'B1': array([1.39331274])}
```

Experiment 44 with parameters:

```
{'coefficients': (0.01, 20), 'learning_rate': 0.0001, 'epochs': 5, 'x_interval': (-100, 100)}
```

Experiment results were: mean_cost=348.0788 with variance=1241.8302

Parameters of network

```
{'W1': array([[0.0268159]])}
```

```
{'B1': array([1.39257003])}
```

```
In [ ]: def experiment_learn_FFN(learning_rate, epochs, random_x_function, neurons_per_l
target_FFN = FFN(
    neurons_per_layer_list=neurons_per_layer_list,
    activation_functions_list=activation_functions_list,
    cost_function=cost_function
)

data_set = create_a_inputs_b_outputs_XY_data(
    a=neurons_per_layer_list[0],
    b=neurons_per_layer_list[-1],
    num_data_items=10_000,
    random_x_function= random_x_function,
    function= lambda X: target_FFN.forward_propagate(X)[0]
)

model_to_train = FFN(
    neurons_per_layer_list=neurons_per_layer_list,
    activation_functions_list=activation_functions_list,
    cost_function=cost_function
)

model = Model(
    FFN=model_to_train,
    data_set=data_set
)

mean_cost, variance_cost = model.train_and_evaluate(
    learning_rate=learning_rate,
    epochs=epochs,
    batch_size=50
)

print(f"Experiment results were:   mean_cost={mean_cost:.4f}   with   varian

# model.print_FFN_parameters()

return (model, target_FFN)
```

```
In [ ]: def experiment_generator_factory_FFN(
    learning_rates, epochs_sets, x_intervals, neurons_and_activation_layer_l
):
    for learning_rate, epochs, x_interval, (neurons_per_layer_list, activation_f
    learning_rates, epochs_sets, x_intervals, neurons_and_activation_layer_l
    ):
        parameters = {
            "learning_rate": learning_rate,
            "epochs": epochs,
            "x_interval": x_interval,
            "neurons_per_layer_list": neurons_per_layer_list,
            "activation_functions_list": [type(activation).__name__ for activati
            "cost_function": type(cost_function).__name__,
        }
        def experiment():
            return experiment_learn_FFN(
                learning_rate=learning_rate,
                epochs=epochs,
```

```

        random_x_function=lambda: random.uniform(*x_interval),
        neurons_per_layer_list=neurons_per_layer_list,
        activation_functions_list=activation_functions_list,
        cost_function=cost_function
    )
    yield experiment, parameters

```

```

In [ ]: experiemnt_generator = experiment_generator_factory_FFN(
    neurons_and_activation_layer_lists=(
        (
            [5, 5, 5, 5],
            [Sigmoid(), Sigmoid(), None]
        ),
        (
            [1, 10, 1],
            [RELU(), Sigmoid()]
        ),
        (
            [10, 20, 20, 10],
            [RELU(), RELU(), Sigmoid()]
        ),
        (
            [1, 1],
            [None]
        )
    ),
    cost_functions=(
        MSE(),
    ),
    epochs_sets=(
        # 5, 10, 20,
        5,
    ),
    learning_rates=(
        10**-i for i in range(2, 5)
    ),
    x_intervals = (
        (-1, 1),
        (0, 1),
        (-10, 10),
    ),
)

trained_models = []
traget_FFNs = []
for i, (experiment, parameters) in enumerate(experiemnt_generator):
    print(f"Experiment {i} with parameters:")
    print(parameters)
    trained_model, traget_FFN = experiment()
    trained_models.append(trained_model)
    traget_FFNs.append(traget_FFN)
    # model[i].print_parameters()
    print("\n")

```

Experiment 0 with parameters:

```
{'learning_rate': 0.01, 'epochs': 5, 'x_interval': (-1, 1), 'neurons_per_layer_list': [5, 5, 5, 5], 'activation_functions_list': ['Sigmoid', 'Sigmoid', 'NoneType'], 'cost_function': 'MSE'}
```

Experiment results were: mean_cost=1.3627 with variance=3438.6313

Experiment 1 with parameters:

```
{'learning_rate': 0.01, 'epochs': 5, 'x_interval': (-1, 1), 'neurons_per_layer_list': [1, 10, 1], 'activation_functions_list': ['RELU', 'Sigmoid'], 'cost_function': 'MSE'}
```

Experiment results were: mean_cost=26.3690 with variance=288712.8200

Experiment 2 with parameters:

```
{'learning_rate': 0.01, 'epochs': 5, 'x_interval': (-1, 1), 'neurons_per_layer_list': [10, 20, 20, 10], 'activation_functions_list': ['RELU', 'RELU', 'Sigmoid'], 'cost_function': 'MSE'}
```

Experiment results were: mean_cost=10.4138 with variance=90995.8591

Experiment 3 with parameters:

```
{'learning_rate': 0.01, 'epochs': 5, 'x_interval': (-1, 1), 'neurons_per_layer_list': [1, 1], 'activation_functions_list': ['NoneType'], 'cost_function': 'MSE'}
```

Experiment results were: mean_cost=0.0000 with variance=0.0000

Experiment 4 with parameters:

```
{'learning_rate': 0.01, 'epochs': 5, 'x_interval': (0, 1), 'neurons_per_layer_list': [5, 5, 5, 5], 'activation_functions_list': ['Sigmoid', 'Sigmoid', 'NoneType'], 'cost_function': 'MSE'}
```

Experiment results were: mean_cost=0.0000 with variance=0.0000

Experiment 5 with parameters:

```
{'learning_rate': 0.01, 'epochs': 5, 'x_interval': (0, 1), 'neurons_per_layer_list': [1, 10, 1], 'activation_functions_list': ['RELU', 'Sigmoid'], 'cost_function': 'MSE'}
```

Experiment results were: mean_cost=0.8162 with variance=0.0462

Experiment 6 with parameters:

```
{'learning_rate': 0.01, 'epochs': 5, 'x_interval': (0, 1), 'neurons_per_layer_list': [10, 20, 20, 10], 'activation_functions_list': ['RELU', 'RELU', 'Sigmoid'], 'cost_function': 'MSE'}
```

Experiment results were: mean_cost=2.8083 with variance=7265.0272

Experiment 7 with parameters:

```
{'learning_rate': 0.01, 'epochs': 5, 'x_interval': (0, 1), 'neurons_per_layer_list': [1, 1], 'activation_functions_list': ['NoneType'], 'cost_function': 'MSE'}
```

Experiment results were: mean_cost=0.0003 with variance=0.0000

Experiment 8 with parameters:

```
{'learning_rate': 0.01, 'epochs': 5, 'x_interval': (-10, 10), 'neurons_per_layer_list': [5, 5, 5, 5], 'activation_functions_list': ['Sigmoid', 'Sigmoid', 'NoneType'], 'cost_function': 'MSE'}
```

Experiment results were: mean_cost=0.1186 with variance=24.1015

Experiment 9 with parameters:

```
{'learning_rate': 0.01, 'epochs': 5, 'x_interval': (-10, 10), 'neurons_per_layer_list': [1, 10, 1], 'activation_functions_list': ['RELU', 'Sigmoid'], 'cost_function': 'MSE'}
```

Experiment results were: mean_cost=0.3614 with variance=71.4001

Experiment 10 with parameters:

```
{'learning_rate': 0.01, 'epochs': 5, 'x_interval': (-10, 10), 'neurons_per_layer_list': [10, 20, 20, 10], 'activation_functions_list': ['RELU', 'RELU', 'Sigmoid'], 'cost_function': 'MSE'}
```

Experiment results were: mean_cost=0.4338 with variance=97.8740

Experiment 11 with parameters:

```
{'learning_rate': 0.01, 'epochs': 5, 'x_interval': (-10, 10), 'neurons_per_layer_list': [1, 1], 'activation_functions_list': ['NoneType'], 'cost_function': 'MSE'}
```

Experiment results were: mean_cost=0.0000 with variance=0.0000

Experiment 12 with parameters:

```
{'learning_rate': 0.001, 'epochs': 5, 'x_interval': (-1, 1), 'neurons_per_layer_list': [5, 5, 5, 5], 'activation_functions_list': ['Sigmoid', 'Sigmoid', 'NoneType'], 'cost_function': 'MSE'}
```

Experiment results were: mean_cost=1.4221 with variance=1766.6873

Experiment 13 with parameters:

```
{'learning_rate': 0.001, 'epochs': 5, 'x_interval': (-1, 1), 'neurons_per_layer_list': [1, 10, 1], 'activation_functions_list': ['RELU', 'Sigmoid'], 'cost_function': 'MSE'}
```

Experiment results were: mean_cost=0.2378 with variance=0.1880

Experiment 14 with parameters:

```
{'learning_rate': 0.001, 'epochs': 5, 'x_interval': (-1, 1), 'neurons_per_layer_list': [10, 20, 20, 10], 'activation_functions_list': ['RELU', 'RELU', 'Sigmoid'], 'cost_function': 'MSE'}
```

Experiment results were: mean_cost=21.2195 with variance=543041.7857

Experiment 15 with parameters:

```
{'learning_rate': 0.001, 'epochs': 5, 'x_interval': (-1, 1), 'neurons_per_layer_list': [1, 1], 'activation_functions_list': ['NoneType'], 'cost_function': 'MSE'}
```

Experiment results were: mean_cost=0.0534 with variance=0.0023

Experiment 16 with parameters:

```
{'learning_rate': 0.001, 'epochs': 5, 'x_interval': (0, 1), 'neurons_per_layer_list': [5, 5, 5, 5], 'activation_functions_list': ['Sigmoid', 'Sigmoid', 'NoneType'], 'cost_function': 'MSE'}
```

Experiment results were: mean_cost=0.1667 with variance=31.3099

Experiment 17 with parameters:

```
{'learning_rate': 0.001, 'epochs': 5, 'x_interval': (0, 1), 'neurons_per_layer_list': [1, 10, 1], 'activation_functions_list': ['RELU', 'Sigmoid'], 'cost_function': 'MSE'}
```

Experiment results were: mean_cost=0.0000 with variance=0.0000

Experiment 18 with parameters:

```
{'learning_rate': 0.001, 'epochs': 5, 'x_interval': (0, 1), 'neurons_per_layer_list': [10, 20, 20, 10], 'activation_functions_list': ['RELU', 'RELU', 'Sigmoid'], 'cost_function': 'MSE'}
```

Experiment results were: mean_cost=0.0776 with variance=3.5353

Experiment 19 with parameters:

```
{'learning_rate': 0.001, 'epochs': 5, 'x_interval': (0, 1), 'neurons_per_layer_list': [1, 1], 'activation_functions_list': ['NoneType'], 'cost_function': 'MSE'}
```

Experiment results were: mean_cost=0.0516 with variance=0.0022

Experiment 20 with parameters:

```
{'learning_rate': 0.001, 'epochs': 5, 'x_interval': (-10, 10), 'neurons_per_layer_list': [5, 5, 5, 5], 'activation_functions_list': ['Sigmoid', 'Sigmoid', 'NoneType'], 'cost_function': 'MSE'}
```

Experiment results were: mean_cost=0.0162 with variance=0.1314

Experiment 21 with parameters:

```
{'learning_rate': 0.001, 'epochs': 5, 'x_interval': (-10, 10), 'neurons_per_layer_list': [1, 10, 1], 'activation_functions_list': ['RELU', 'Sigmoid'], 'cost_function': 'MSE'}
```

Experiment results were: mean_cost=0.0194 with variance=0.1468

Experiment 22 with parameters:

```
{'learning_rate': 0.001, 'epochs': 5, 'x_interval': (-10, 10), 'neurons_per_layer_list': [10, 20, 20, 10], 'activation_functions_list': ['RELU', 'RELU', 'Sigmoid'], 'cost_function': 'MSE'}
```

Experiment results were: mean_cost=1.4488 with variance=3723.7081

Experiment 23 with parameters:

```
{'learning_rate': 0.001, 'epochs': 5, 'x_interval': (-10, 10), 'neurons_per_layer_list': [1, 1], 'activation_functions_list': ['NoneType'], 'cost_function': 'MSE'}
```

Experiment results were: mean_cost=0.0000 with variance=0.0000

Experiment 24 with parameters:

```
{'learning_rate': 0.0001, 'epochs': 5, 'x_interval': (-1, 1), 'neurons_per_layer_list': [5, 5, 5, 5], 'activation_functions_list': ['Sigmoid', 'Sigmoid', 'NoneType'], 'cost_function': 'MSE'}
```

Experiment results were: mean_cost=0.0257 with variance=0.0530

Experiment 25 with parameters:

```
{'learning_rate': 0.0001, 'epochs': 5, 'x_interval': (-1, 1), 'neurons_per_layer_list': [1, 10, 1], 'activation_functions_list': ['RELU', 'Sigmoid'], 'cost_function': 'MSE'}
```

Experiment results were: mean_cost=0.0304 with variance=0.0200

Experiment 26 with parameters:

```
{'learning_rate': 0.0001, 'epochs': 5, 'x_interval': (-1, 1), 'neurons_per_layer_list': [10, 20, 20, 10], 'activation_functions_list': ['RELU', 'RELU', 'Sigmoid']}
```

```
d'], 'cost_function': 'MSE'}
```

Experiment results were: mean_cost=8.9681 with variance=133081.8454

Experiment 27 with parameters:

```
{'learning_rate': 0.0001, 'epochs': 5, 'x_interval': (-1, 1), 'neurons_per_layer_list': [1, 1], 'activation_functions_list': ['NoneType'], 'cost_function': 'MSE'}
```

Experiment results were: mean_cost=0.0294 with variance=0.0007

Experiment 28 with parameters:

```
{'learning_rate': 0.0001, 'epochs': 5, 'x_interval': (0, 1), 'neurons_per_layer_list': [5, 5, 5, 5], 'activation_functions_list': ['Sigmoid', 'Sigmoid', 'NoneType'], 'cost_function': 'MSE'}
```

Experiment results were: mean_cost=0.7180 with variance=996.7866

Experiment 29 with parameters:

```
{'learning_rate': 0.0001, 'epochs': 5, 'x_interval': (0, 1), 'neurons_per_layer_list': [1, 10, 1], 'activation_functions_list': ['RELU', 'Sigmoid'], 'cost_function': 'MSE'}
```

Experiment results were: mean_cost=0.0067 with variance=0.0002

Experiment 30 with parameters:

```
{'learning_rate': 0.0001, 'epochs': 5, 'x_interval': (0, 1), 'neurons_per_layer_list': [10, 20, 20, 10], 'activation_functions_list': ['RELU', 'RELU', 'Sigmoid'], 'cost_function': 'MSE'}
```

Experiment results were: mean_cost=4.4068 with variance=16482.6493

Experiment 31 with parameters:

```
{'learning_rate': 0.0001, 'epochs': 5, 'x_interval': (0, 1), 'neurons_per_layer_list': [1, 1], 'activation_functions_list': ['NoneType'], 'cost_function': 'MSE'}
```

Experiment results were: mean_cost=0.1532 with variance=0.0275

Experiment 32 with parameters:

```
{'learning_rate': 0.0001, 'epochs': 5, 'x_interval': (-10, 10), 'neurons_per_layer_list': [5, 5, 5, 5], 'activation_functions_list': ['Sigmoid', 'Sigmoid', 'NoneType'], 'cost_function': 'MSE'}
```

Experiment results were: mean_cost=0.0250 with variance=0.2429

Experiment 33 with parameters:

```
{'learning_rate': 0.0001, 'epochs': 5, 'x_interval': (-10, 10), 'neurons_per_layer_list': [1, 10, 1], 'activation_functions_list': ['RELU', 'Sigmoid'], 'cost_function': 'MSE'}
```

Experiment results were: mean_cost=0.0011 with variance=0.0000

Experiment 34 with parameters:

```
{'learning_rate': 0.0001, 'epochs': 5, 'x_interval': (-10, 10), 'neurons_per_layer_list': [10, 20, 20, 10], 'activation_functions_list': ['RELU', 'RELU', 'Sigmoid'], 'cost_function': 'MSE'}
```

Experiment results were: mean_cost=0.0796 with variance=2.9101

Experiment 35 with parameters:

```
{'learning_rate': 0.0001, 'epochs': 5, 'x_interval': (-10, 10), 'neurons_per_layer_list': [10, 20, 20, 10], 'activation_functions_list': ['RELU', 'RELU', 'Sigmoid'], 'cost_function': 'MSE'}
```

```
r_list': [1, 1], 'activation_functions_list': ['NoneType'], 'cost_function': 'MSE'}
```

Experiment results were: mean_cost=0.0000 with variance=0.0000

In []: *# check result of experiment that supposedly went well*

```
experiment_index = 4
x_range = (0, 1)

X = np.array([random.uniform(*x_range) for _ in range(5)])
Y, _ = target_FFNS[experiment_index].forward_propagate(X)
P, _ = trained_models[experiment_index].FFN.forward_propagate(X)
cost = MSE()(P, Y)

print(X)
print(Y)
print(P)
print(cost)
```

```
[0.3676792  0.25133716 0.04669012 0.53569267 0.69325618]
[0.59594197 0.49997183 0.64087      0.50858402 0.52346084]
[0.59550865 0.50032766 0.64012531 0.50931319 0.5234302 ]
2.803158360779426e-07
[0.5 0.5 0.5 0.5 0.5]
[0.5 0.5 0.5 0.5 0.5]
```

In []: `print(target_FFNS[experiment_index].transformation_layers[2].bias)`
`print(target_FFNS[experiment_index].transformation_layers[2].bias)`
`print(target_FFNS[experiment_index].transformation_layers[2].bias - target_FFNS[exp`
`)`

```
[0.5 0.5 0.5 0.5 0.5]
[0.5 0.5 0.5 0.5 0.5]
[0. 0. 0. 0. 0.]
```

In []: `for i in range(3):`
`print(target_FFNS[experiment_index].transformation_layers[i].weights`
`-`
`target_FFNS[experiment_index].transformation_layers[i].weights`
`)`
`print(target_FFNS[experiment_index].transformation_layers[i].bias`
`-`
`target_FFNS[experiment_index].transformation_layers[i].bias`
`)`


```
[[0. 0. 0. 0. 0.]  
 [0. 0. 0. 0. 0.]  
 [0. 0. 0. 0. 0.]  
 [0. 0. 0. 0. 0.]  
 [0. 0. 0. 0. 0.]]  
[0. 0. 0. 0. 0.]  
[[0. 0. 0. 0. 0.]  
 [0. 0. 0. 0. 0.]  
 [0. 0. 0. 0. 0.]  
 [0. 0. 0. 0. 0.]  
 [0. 0. 0. 0. 0.]]  
[0. 0. 0. 0. 0.]  
[[0. 0. 0. 0. 0.]  
 [0. 0. 0. 0. 0.]  
 [0. 0. 0. 0. 0.]  
 [0. 0. 0. 0. 0.]  
 [0. 0. 0. 0. 0.]]  
[0. 0. 0. 0. 0.]
```