NNC_Final_Project

December 15, 2019

Code

[4]:
```
import numpy as np


class Model:
    def __init__(self, n0, n1, n2):
        self.n_0 = n0
        self.n_1 = n1
        self.n_2 = n2

        self.reinitialize_weights = True

        # weights and biases initialized as placeholders.
        # reinitialized later
        self.weights_1 = np.zeros((self.n_0, self.n_1), dtype=int)
        self.weights_2 = np.zeros((self.n_1, self.n_2), dtype=int)
        self.biases_1 = np.zeros((1, self.n_1), dtype=int)
        self.biases_2 = np.zeros((1, self.n_2), dtype=int)

        self.activations_1 = np.zeros((1, self.n_1), dtype=int)
        self.activations_2 = np.zeros((1, self.n_2), dtype=int)
        self.sensitivities_1 = np.zeros((1, self.n_1), dtype=int)
        self.sensitivities_2 = np.zeros((1, self.n_2), dtype=int)

        self.hyper_params = HyperParams()

        self.model_info = Info()


class HyperParams:
    def __init__(self):
        self.alpha_list = [0.1, 0.2, 0.3]
        self.zeta_list = [0.5, 1, 1.5]
        self.x0_list = [0.5, 1, 1.5]
        self.max_epochs = 700  # empirically chosen
        self.tolerance = 0.05
        self.learning_rate = self.alpha_list[1]
```

```
        self.zeta = self.zeta_list[1]
        self.x0 = self.x0_list[1]
        self.cost_fn = 0  # {0: quadratic, 1: cross-entropy}


class Info:
    def __init__(self, total_epochs=0, last_epoch_error=0.0, convergence=False):
        self.total_epochs_req = total_epochs
        self.last_epoch_error = last_epoch_error
        self.converged = convergence
```

[3]:
```
import pandas as pd

sheets = {}


def update_sheet(writer, sheet_name, sheet_obj):
    df_obj = {
        'Model architecture': sheet_obj['model_arch_list'],
        'Model weights': sheet_obj['model_weight_list'],
        'Model biases': sheet_obj['model_bias_list'],
        '# Training epochs': sheet_obj['total_epochs_req_list'],
        'Learning Rate': sheet_obj['learning_rate_list'],
        'Zeta': sheet_obj['zeta_list'],
        'X0': sheet_obj['x0_list'],
        'Cost Function': sheet_obj['cost_fn_list'],
        'Last epoch error': sheet_obj['last_epoch_error_list'],
        'Did converge?': sheet_obj['converged_list'],
    }
    df = pd.DataFrame(df_obj)
    df.to_excel(writer, sheet_name=sheet_name, index=False)


def save_data(sheet_name, model):
    try:
        sheet_obj = sheets[sheet_name]
    except KeyError:
        sheet_obj = {}
    if not sheet_obj:
        sheet_obj = {'model_arch_list': [], 'model_weight_list': [],␣
↪'model_bias_list': [],
                     'total_epochs_req_list': [], 'learning_rate_list': [],␣
↪'zeta_list': [],
                     'x0_list': [], 'cost_fn_list': [], 'last_epoch_error_list':␣
↪[],
                     'converged_list': []}
```

```python
    sheet_obj['model_arch_list'].append("[ " + str(model.n_0) + ", " + str(model.
 ↪n_1) +
                                        ", " + str(model.n_2) + "]")
    sheet_obj['model_weight_list'].append("Weights1 = " + str(model.weights_1) +
                                          "\nWeights2 = " + str(model.weights_2))
    sheet_obj['model_bias_list'].append("Biases1 = " + str(model.biases_1) +
                                        "\nBiases2 = " + str(model.biases_2))
    sheet_obj['total_epochs_req_list'].append(model.model_info.total_epochs_req)
    sheet_obj['learning_rate_list'].append(model.hyper_params.learning_rate)
    sheet_obj['zeta_list'].append(model.hyper_params.zeta)
    sheet_obj['x0_list'].append(model.hyper_params.x0)
    sheet_obj['cost_fn_list'].append("Quadratic" if model.hyper_params.cost_fn␣
 ↪== 0
                                     else "Cross-Entropy")
    sheet_obj['last_epoch_error_list'].append(model.model_info.last_epoch_error)
    sheet_obj['converged_list'].append("Yes" if model.model_info.converged else␣
 ↪"No")
    sheets[sheet_name] = sheet_obj


def export_data():
    print("Starting export")
    writer = pd.ExcelWriter('Results.xlsx', engine='xlsxwriter')
    if not writer:
        print("Error while opening writer. Exiting.")
        return
    for sheet_name in sheets:
        sheet_obj = sheets[sheet_name]
        if not sheet_obj:
            print("Skipping sheet - %s " % sheet_name)
            continue
        print("Updating sheet - %s " % sheet_name)
        update_sheet(writer, sheet_name, sheet_obj)
    writer.save()
    print("Data exported")
```

[1]:

```python
import getopt
import sys
import numpy as np

from model import Model, HyperParams
from save import save_data, export_data

export_to_excel = False


def transfer_ftn(n_l, x0):
```

```
    a_l = np.tanh(n_l / (2 * x0))
    return a_l


# we only save a_l NOT n_l if using bipolar sigmoid transfer function
def derivative_transfer_ftn(a_l, x0):
    derivative = ((1 + a_l) * (1 - a_l)) / (2 * x0)
    return derivative


def init_weights_biases(model):
    if model.reinitialize_weights:
        model.weights_1 = np.random.uniform(-1 * model.hyper_params.zeta,
                                            model.hyper_params.zeta, model.
 ↪weights_1.shape)

        model.weights_2 = np.random.uniform(-1 * model.hyper_params.zeta,
                                            model.hyper_params.zeta, model.
 ↪weights_2.shape)

        model.biases_1 = np.random.uniform(-1 * model.hyper_params.zeta,
                                           model.hyper_params.zeta, model.
 ↪biases_1.shape)

        model.biases_2 = np.random.uniform(-1 * model.hyper_params.zeta,
                                           model.hyper_params.zeta, model.
 ↪biases_2.shape)

    return [model.weights_1, model.weights_2], [model.biases_1, model.biases_2]


def train_nn(x_train, y_train, model):
    Q = len(x_train)
    weight_list, bias_list = init_weights_biases(model)
    weight_list_len = len(weight_list)
    for epoch in range(model.hyper_params.max_epochs):
        epoch_error = 0
        for iteration in range(Q):
            x = x_train[iteration]
            y = y_train[iteration]
            x = np.array(x).reshape((1, len(x)))
            y = np.array(y).reshape((1, len(y)))

            # Calculate activations for all layers
            # don't need to save n_l if we are using bipolar sigmoid transfer␣
 ↪function
            a_l_list = [x]
```

```
            for i in range(len(weight_list)):
                n_l = np.matmul(a_l_list[-1], weight_list[i]) + bias_list[i]
                a_l = transfer_ftn(n_l, model.hyper_params.x0)
                a_l_list.append(a_l)

            # calculating the error for this example
            y_hat = a_l_list[-1]  # activation of the last layer
            example_error = np.matmul(y_hat - y, (y_hat - y).T)
            example_error = np.asscalar(example_error)
            epoch_error = epoch_error + example_error

            # Calculate sensitivities for last layer. Performs element-wise␣
↪multiplication.
            # quadratic cost function
            if model.hyper_params.cost_fn == 0:
                s_L = np.multiply((y_hat - y), derivative_transfer_ftn(y_hat,␣
↪model.hyper_params.x0))
            # cross entropy cost function
            elif model.hyper_params.cost_fn == 1:
                s_L = y_hat - y

            # Calculate sensitivites for other layers
            sensitivities_list = [s_L]

            for l in range(weight_list_len - 1, 0, -1):
                s_l = np.multiply(np.matmul(sensitivities_list[0],␣
↪weight_list[l].T), \
                                  derivative_transfer_ftn(a_l_list[l], model.
↪hyper_params.x0))
                sensitivities_list.insert(0, s_l)

            # Update weights and biases
            for l in range(weight_list_len):
                weight_list[l] = weight_list[l] - \
                                 (model.hyper_params.learning_rate *
                                  np.matmul(a_l_list[l].T,␣
↪sensitivities_list[l]))

                bias_list[l] = bias_list[l] - \
                               (model.hyper_params.learning_rate *␣
↪sensitivities_list[l])

        # epoch error is not normalized (not divided by number of examples)
        if epoch_error < model.hyper_params.tolerance:
            break
```

```python
    num_training_epochs = epoch + 1
    if num_training_epochs < model.hyper_params.max_epochs:
        convergence = True
    else:
        convergence = False

    update_model_info(model, weight_list, bias_list, num_training_epochs,␣
 ↪epoch_error, convergence)

    return model


def update_model_info(model, weight_list, bias_list, num_training_epochs,␣
 ↪epoch_error, convergence):
    model.weights_1 = weight_list[0]
    model.weights_2 = weight_list[1]
    model.biases_1 = bias_list[0]
    model.biases_2 = bias_list[1]
    model.model_info.total_epochs_req = num_training_epochs
    model.model_info.last_epoch_error = epoch_error
    model.model_info.converged = convergence


def extract_model_info(model, sheet_name, verbose=True):
    if verbose:
        ␣
 ↪print("--------------------------------------------------------------------------------")
        print(f"Learning rate = {model.hyper_params.learning_rate} | "
              f"Zeta = {model.hyper_params.zeta} | "
              f"x0 = {model.hyper_params.x0}")
        print(f"Convergence = {model.model_info.converged} | "
              f"Training Epochs = {model.model_info.total_epochs_req} | "
              f"Squared Error = {model.model_info.last_epoch_error}")
        ␣
 ↪print("--------------------------------------------------------------------------------")

    if export_to_excel:
        save_data(sheet_name, model)


def part_2a(x_train, y_train, model, sheet_name):
    # TODO
    # Look for patterns when do we get non-convergent results
    # Try all 3X3X3=27 hyper parameter combinations of alpha, zeta and x0
    num_convergence = 0
    for alpha in model.hyper_params.alpha_list:
        for zeta in model.hyper_params.zeta_list:
```

```
            for x0 in model.hyper_params.x0_list:
                model.hyper_params.learning_rate = alpha
                model.hyper_params.zeta = zeta
                model.hyper_params.x0 = x0
                model = train_nn(x_train, y_train, model)
                if model.model_info.converged:
                    num_convergence += 1
                extract_model_info(model, sheet_name, verbose=True)
    
 ↪print("-----------------------------------------------------------------------")
    print(f"Number of convergent hyper parameter combinations = 
 ↪{num_convergence} (out of 27)")


def part_2b(x_train, y_train, cost_fn, sheet_name):
    N1_list = [1, 2, 4, 6, 8, 10]
    convergence_list = []
    for i in range(len(N1_list)):
        model = Model(2, N1_list[i], 1)
        model.hyper_params.cost_fn = cost_fn
        num_convergence = 0
        for iters in range(100):
            model = train_nn(x_train, y_train, model)
            if model.model_info.converged:
                num_convergence += 1
            extract_model_info(model, sheet_name, verbose=False)
        convergence_list.append(num_convergence)
        print(f"Convergence for N1 = %d -> %d" % (N1_list[i], num_convergence))

    print(f"Convergence results for N1 = [1,2,4,6,8,10] (out of 100): 
 ↪{convergence_list}")

    # Results mostly converge for N1=4 and above. For N1=2, almost 70% of the 
 ↪times,
    # it converges. For N1=1, it doesn't converge at all.
    # This is probably because the XOR problem is not linearly separable and we 
 ↪need a higher
    # number of neurons in the hidden layer to approximate the function (see 
 ↪universality theorem).


def xor_weight_validation(x_train, y_train, model, sheet_name):
    model.hyper_params.max_epochs = 1

    # Setting initial weights and biases for xor weight validation
    model.weights_1 = np.array([[0.197, 0.3191, -0.1448, 0.3594],
```

```
                            [0.3099, 0.1904, -0.0347, -0.4861]]).
↪reshape(model.weights_1.shape)
    model.weights_2 = np.array([0.4919, -0.2913, -0.3979, 0.3581]).reshape(model.
↪weights_2.shape)
    model.biases_1 = np.array([-0.3378, 0.2771, 0.2859, -0.3329]).reshape(model.
↪biases_1.shape)
    model.biases_2 = np.array([-0.1401]).reshape(model.biases_2.shape)
    model.reinitialize_weights = False

    model = train_nn(x_train, y_train, model)

    print("W1=", model.weights_1, sep="\n")
    print("b1=", model.biases_1, sep="\n")
    print("W2=", model.weights_2, sep="\n")
    print("b2=", model.biases_2, sep="\n")

    extract_model_info(model, sheet_name, verbose=False)

    # np.savez('xor_weight_validation.npz', model=model)
    # results can be loaded from the xor_weight_validation.npz file by␣
↪uncommenting the following
    # data = np.load('Part1_results.npz')
    # model = data['model']


def main():
    # # uncomment this if data needs to be stored in excel
    # global export_to_excel
    # export_to_excel = True

    x_train = [[1, 1], [1, -1], [-1, 1], [-1, -1]]
    y_train = [[-1], [1], [1], [-1]]

    model = Model(2, 4, 1)
    xor_weight_validation(x_train, y_train, model, sheet_name="XOR weights␣
↪validation")

    # using quadratic cost ftn
    model = Model(2, 4, 1)
    model.hyper_params.cost_fn = 0
    part_2a(x_train, y_train, model, sheet_name="A-Z-X0 variations (Quad)")
    part_2b(x_train, y_train, cost_fn=0, sheet_name="N1 variations (Quad)")

    # using cross entropy cost ftn
    model = Model(2, 4, 1)
    model.hyper_params.cost_fn = 1
    part_2a(x_train, y_train, model, sheet_name="A-Z-X0 variations (CrsEnt)")
```

```python
    part_2b(x_train, y_train, cost_fn=1, sheet_name="N1 variations (CrsEnt)")

    model = Model(2, 4, 1)
    model.hyper_params.learning_rate = 0.2
    model.hyper_params.zeta = 1.0
    model.hyper_params.x0 = 1.0
    model.hyper_params.cost_fn = 1
    model.hyper_params.max_epochs = 1
    model = train_nn(x_train, y_train, model)
    extract_model_info(model, sheet_name="Final verification")

    # should be set to true above
    if export_to_excel:
        export_data()

if __name__ == "__main__":
    main()
```

```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:62:
DeprecationWarning: np.asscalar(a) is deprecated since NumPy v1.16, use a.item()
instead

W1=
[[ 0.19347555  0.31675476 -0.144748    0.36374521]
 [ 0.30686735  0.18845288 -0.03301594 -0.48859019]]
b1=
[[-0.32243413  0.26504268  0.27330512 -0.32503622]]
W2=
[[ 0.47534885]
 [-0.27642811]
 [-0.38395025]
 [ 0.34801327]]
b2=
[[-0.08027444]]
--------------------------------------------------------------------------------
Learning rate = 0.1 | Zeta = 0.5 | x0 = 0.5
Convergence = True | Training Epochs = 144 | Squared Error =
0.049873341492876956
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Learning rate = 0.1 | Zeta = 0.5 | x0 = 1
Convergence = False | Training Epochs = 700 | Squared Error =
0.055927975170455696
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Learning rate = 0.1 | Zeta = 0.5 | x0 = 1.5
Convergence = False | Training Epochs = 700 | Squared Error = 4.0477393613429244
--------------------------------------------------------------------------------
```

```
-------------------------------------------------------------------------------
Learning rate = 0.1 | Zeta = 1 | x0 = 0.5
Convergence = True | Training Epochs = 97 | Squared Error = 0.049637115432790604
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
Learning rate = 0.1 | Zeta = 1 | x0 = 1
Convergence = True | Training Epochs = 640 | Squared Error = 0.04986490264371217
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
Learning rate = 0.1 | Zeta = 1 | x0 = 1.5
Convergence = False | Training Epochs = 700 | Squared Error = 0.3720739550348952
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
Learning rate = 0.1 | Zeta = 1.5 | x0 = 0.5
Convergence = False | Training Epochs = 700 | Squared Error = 4.000705200611622
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
Learning rate = 0.1 | Zeta = 1.5 | x0 = 1
Convergence = True | Training Epochs = 458 | Squared Error =
0.049987887286957175
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
Learning rate = 0.1 | Zeta = 1.5 | x0 = 1.5
Convergence = False | Training Epochs = 700 | Squared Error =
0.14315651838094595
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
Learning rate = 0.2 | Zeta = 0.5 | x0 = 0.5
Convergence = False | Training Epochs = 700 | Squared Error = 4.862384115264075
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
Learning rate = 0.2 | Zeta = 0.5 | x0 = 1
Convergence = False | Training Epochs = 700 | Squared Error = 4.204958347834154
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
Learning rate = 0.2 | Zeta = 0.5 | x0 = 1.5
Convergence = False | Training Epochs = 700 | Squared Error = 4.094065865517321
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
Learning rate = 0.2 | Zeta = 1 | x0 = 0.5
Convergence = True | Training Epochs = 52 | Squared Error = 0.048779595735270326
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
Learning rate = 0.2 | Zeta = 1 | x0 = 1
Convergence = True | Training Epochs = 408 | Squared Error = 0.04991055755562201
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
Learning rate = 0.2 | Zeta = 1 | x0 = 1.5
```

```
Convergence = True | Training Epochs = 631 | Squared Error = 0.04988909483521971
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Learning rate = 0.2 | Zeta = 1.5 | x0 = 0.5
Convergence = True | Training Epochs = 69 | Squared Error = 0.04907998529277527
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Learning rate = 0.2 | Zeta = 1.5 | x0 = 1
Convergence = True | Training Epochs = 224 | Squared Error = 0.04987026445288031
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Learning rate = 0.2 | Zeta = 1.5 | x0 = 1.5
Convergence = True | Training Epochs = 553 | Squared Error = 0.04999293222262656
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Learning rate = 0.3 | Zeta = 0.5 | x0 = 0.5
Convergence = True | Training Epochs = 54 | Squared Error = 0.049039109630381855
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Learning rate = 0.3 | Zeta = 0.5 | x0 = 1
Convergence = False | Training Epochs = 700 | Squared Error = 4.310545405999978
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Learning rate = 0.3 | Zeta = 0.5 | x0 = 1.5
Convergence = False | Training Epochs = 700 | Squared Error = 4.138021047560265
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Learning rate = 0.3 | Zeta = 1 | x0 = 0.5
Convergence = True | Training Epochs = 31 | Squared Error = 0.04995314267186744
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Learning rate = 0.3 | Zeta = 1 | x0 = 1
Convergence = True | Training Epochs = 185 | Squared Error =
0.049735165791450764
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Learning rate = 0.3 | Zeta = 1 | x0 = 1.5
Convergence = True | Training Epochs = 605 | Squared Error = 0.04984200046100855
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Learning rate = 0.3 | Zeta = 1.5 | x0 = 0.5
Convergence = True | Training Epochs = 27 | Squared Error = 0.04787247678823797
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Learning rate = 0.3 | Zeta = 1.5 | x0 = 1
Convergence = True | Training Epochs = 137 | Squared Error =
0.049822583171420486
--------------------------------------------------------------------------------
```

```
--------------------------------------------------------------------------------
Learning rate = 0.3 | Zeta = 1.5 | x0 = 1.5
Convergence = True | Training Epochs = 391 | Squared Error = 0.04992550078242809
--------------------------------------------------------------------------------
-----------------------------------------------------------------------
Number of convergent hyper parameter combinations = 17 (out of 27)
Convergence for N1 = 1 -> 0
Convergence for N1 = 2 -> 82
Convergence for N1 = 4 -> 100
Convergence for N1 = 6 -> 100
Convergence for N1 = 8 -> 100
Convergence for N1 = 10 -> 98
Convergence results for N1 = [1,2,4,6,8,10] (out of 100): [0, 82, 100, 100, 100,
98]
--------------------------------------------------------------------------------
Learning rate = 0.1 | Zeta = 0.5 | x0 = 0.5
Convergence = True | Training Epochs = 130 | Squared Error = 0.04818403472549024
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Learning rate = 0.1 | Zeta = 0.5 | x0 = 1
Convergence = True | Training Epochs = 273 | Squared Error = 0.04920549727660134
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Learning rate = 0.1 | Zeta = 0.5 | x0 = 1.5
Convergence = True | Training Epochs = 537 | Squared Error =
0.049217856264777524
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Learning rate = 0.1 | Zeta = 1 | x0 = 0.5
Convergence = True | Training Epochs = 59 | Squared Error = 0.04812216331925205
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Learning rate = 0.1 | Zeta = 1 | x0 = 1
Convergence = True | Training Epochs = 189 | Squared Error =
0.048827557727704765
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Learning rate = 0.1 | Zeta = 1 | x0 = 1.5
Convergence = True | Training Epochs = 481 | Squared Error = 0.04918584143788117
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Learning rate = 0.1 | Zeta = 1.5 | x0 = 0.5
Convergence = True | Training Epochs = 32 | Squared Error = 0.04730144103192096
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Learning rate = 0.1 | Zeta = 1.5 | x0 = 1
Convergence = True | Training Epochs = 93 | Squared Error = 0.04935925793673039
--------------------------------------------------------------------------------
```

```
--------------------------------------------------------------------------------
Learning rate = 0.1 | Zeta = 1.5 | x0 = 1.5
Convergence = True | Training Epochs = 169 | Squared Error = 0.04935280121677305
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Learning rate = 0.2 | Zeta = 0.5 | x0 = 0.5
Convergence = True | Training Epochs = 48 | Squared Error = 0.04729646661090205
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Learning rate = 0.2 | Zeta = 0.5 | x0 = 1
Convergence = False | Training Epochs = 700 | Squared Error = 4.4181942951797675
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Learning rate = 0.2 | Zeta = 0.5 | x0 = 1.5
Convergence = False | Training Epochs = 700 | Squared Error = 4.275074610028772
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Learning rate = 0.2 | Zeta = 1 | x0 = 0.5
Convergence = True | Training Epochs = 30 | Squared Error = 0.04450862905880065
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Learning rate = 0.2 | Zeta = 1 | x0 = 1
Convergence = True | Training Epochs = 61 | Squared Error = 0.04999468286965289
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Learning rate = 0.2 | Zeta = 1 | x0 = 1.5
Convergence = True | Training Epochs = 143 | Squared Error =
0.049281296908125924
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Learning rate = 0.2 | Zeta = 1.5 | x0 = 0.5
Convergence = True | Training Epochs = 15 | Squared Error = 0.04682004586369068
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Learning rate = 0.2 | Zeta = 1.5 | x0 = 1
Convergence = True | Training Epochs = 56 | Squared Error = 0.04877838809480512
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Learning rate = 0.2 | Zeta = 1.5 | x0 = 1.5
Convergence = True | Training Epochs = 174 | Squared Error =
0.048719322156361364
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Learning rate = 0.3 | Zeta = 0.5 | x0 = 0.5
Convergence = False | Training Epochs = 700 | Squared Error = 5.302362039146319
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Learning rate = 0.3 | Zeta = 0.5 | x0 = 1
```

```
Convergence = False | Training Epochs = 700 | Squared Error = 4.638025288751011
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Learning rate = 0.3 | Zeta = 0.5 | x0 = 1.5
Convergence = False | Training Epochs = 700 | Squared Error = 4.41819429503458
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Learning rate = 0.3 | Zeta = 1 | x0 = 0.5
Convergence = True | Training Epochs = 15 | Squared Error = 0.04426839330093334
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Learning rate = 0.3 | Zeta = 1 | x0 = 1
Convergence = True | Training Epochs = 26 | Squared Error = 0.043646433081242504
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Learning rate = 0.3 | Zeta = 1 | x0 = 1.5
Convergence = True | Training Epochs = 118 | Squared Error = 0.04929856228031971
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Learning rate = 0.3 | Zeta = 1.5 | x0 = 0.5
Convergence = True | Training Epochs = 11 | Squared Error = 0.049505345251627336
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Learning rate = 0.3 | Zeta = 1.5 | x0 = 1
Convergence = True | Training Epochs = 31 | Squared Error = 0.049029234653004204
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Learning rate = 0.3 | Zeta = 1.5 | x0 = 1.5
Convergence = True | Training Epochs = 82 | Squared Error = 0.04785224419009246
--------------------------------------------------------------------------------
-----------------------------------------------------------------------
Number of convergent hyper parameter combinations = 22 (out of 27)
Convergence for N1 = 1 -> 0
Convergence for N1 = 2 -> 59
Convergence for N1 = 4 -> 90
Convergence for N1 = 6 -> 96
Convergence for N1 = 8 -> 97
Convergence for N1 = 10 -> 100
Convergence results for N1 = [1,2,4,6,8,10] (out of 100): [0, 59, 90, 96, 97,
100]
--------------------------------------------------------------------------------
Learning rate = 0.2 | Zeta = 1.0 | x0 = 1.0
Convergence = False | Training Epochs = 1 | Squared Error = 5.3801314005663095
--------------------------------------------------------------------------------
```

[ ]: