Laporan Tugas Besar
IF3270 - Pembelajaran Mesin

Artificial Neural Network Part 2:
Backpropagation Mini-batch Gradient Descent

Dipersiapkan oleh

Angger Ilham Amanullah	13521001
Kelvin Rayhan Alkarim	13521005
Ditra Rizqa Amadia	13521019
Bernardus Willson	13521021

I. Penjelasan implementasi

Berikut ini merupakan implementasi serta penjelasan algoritma utama yang telah kami buat.

```python
class FFNN:
    # Initialize model
    def __init__(self, model):
        try:
            # Attributes
            self._layers = model["case"]["model"]["layers"]
            self._weights = model["case"]["initial_weights"]
            self._inputs = model["case"]["input"]
            self._outputs = []
            self._outputs
            self._targets = model["case"]["target"]
            self._layer_objects = []
            self._error_node_output = []
            self._loss = 0
            self._predictions = []
            self._stopped_by = "max_iteration"

            # Learning Parameters
            self._learning_rate = model["case"]["learning_parameters"]["learning_rate"]
            self._batch_size = model["case"]["learning_parameters"]["batch_size"]
            self._max_iteration = model["case"]["learning_parameters"]["max_iteration"]
            self._error_threshold = model["case"]["learning_parameters"]["error_threshold"]

            # Expected
            self._expect = model["expect"]
```

Hal pertama yang dilakukan setelah membaca input .json adalah menginisiasi class FFNN. Berikut adalah penjelasan setiap atributnya:

- layers : atribut ini mempresentasikan lapisan-lapisan neuron dengan jumlah neuron pada setiap layer dan jenis fungsi aktivasi pada layer tersebut (dapat berupa relu, sigmoid, linear, dan softmax).
- weights : atribut ini mempresentasikan bobot yang digunakan untuk menghubungkan jaringan neuron antar layer dengan layer selanjutnya.
- inputs : atribut ini digunakan untuk menerima data nilai input yang akan diproses.
- outputs : atribut ini digunakan untuk menyimpan hasil dari pemrosesan Feed Forward Neural Network.

- targets: atribut ini merepresentasikan target atau output sebenarnya dari data yang kita proses.
- layer_objects: atribut ini digunakan untuk menyimpan objek-objek layer yang telah dibuat untuk nantinya digunakan kembali dalam proses backpropagation.
- error_node_ouput: atribut ini merepresentasikan nilai-nilai error node output yang digunakan dalam update weights yang baru setelah dilakukan backpropagation.
- loss: atribut ini merepresentasikan nilai loss yang akan di-update untuk setiap iterasi atau epoch.
- predictions: atribut ini merepresentasikan hasil prediksi data input berdasarkan weight dari hasil training.
- stopped_by: atribut ini merepresentasikan kondisi yang menyebabkan proses training berhenti, antara jumlah iterasi sudah mencapai maksimum, atau nilai loss sudah kurang dari nilai error threshold.
- learning _rate: atribut ini merepresentasikan nilai dari learning rate sebuah model.
- batch_size: atribut ini merepresentasikan jumlah data yang ingin diproses dalam 1 batch dalam pemrosesan mini batch.
- max_iteration: atribut ini merepresentasikan jumlah iterasi maksimal yang dapat dilakukan, jika melebihi jumlah iterasi maksimum maka pemrosesan akan berhenti.
- error_threshold: atribut ini merepresentasikan batas nilai error, jika perhitungan error sudah dibawah batas maka pemrosesan akan berhenti.
- expect : atribut ini digunakan untuk menyimpan hasil output yang diharapkan dari proses jaringan neuron yang telah terjadi. Selain itu, atribut ini juga menyimpan kondisi berhenti proses train yang diharapkan.

```python
# Build the network
def build(self):
    # Loop until max epoch
    for epoch in range(self._max_iteration):
        print(f"[•] Epoch {epoch + 1} ========================================\n")
        current_batch = 0
        self._loss = 0

        # Split data into batches
        for batch_start in range(0, len(self._inputs), self._batch_size):
            current_batch += 1
            print(f"[•] Batch {current_batch} ====================\n")

            X_batch = self._inputs[batch_start:batch_start + self._batch_size]
            y_batch = self._targets[batch_start:batch_start + self._batch_size]

            self._outputs = []
            self._error_node_output = []
            self._layer_objects = []
            batch_loss = 0

            print(f"[•] Building batch {current_batch}")

            # For each data in batch, calculate output
            for i in range(len(X_batch)):
                _in = X_batch[i]
                layers = []

                print(f"[•] Evaluating data {i + 1}")

                # Feed forward
                for j in range(len(self._layers)):

                    # Create layer
                    layer = Layer(j, self._layers[j], self._weights[j], _in)

                    # Calculate forward
                    _in = layer._calculate_forward()

                    layers.append(layer)
                    self._outputs.append(_in)
```

Pertama-tama kita akan membagi data-data yang ada menjadi beberapa batch, sesuai dengan batch_size yang diminta. Setiap batch akan dievaluasi / diproses per input dengan membuat objek ‘Layer’ pada setiap layer. Objek layer dibuat dengan menggunakan konfigurasi layer, bobot yang sesuai, dan data masukan dari iterasi sebelumnya, karena

seperti yang sudah dijelaskan di atas bahwa setiap layer akan saling terhubung dengan layer selanjutnya.

```python
class Layer:
    # Initialize layer
    def __init__(self, id, layer, weight, input):
        try:
            self._id = id
            self._num_of_neuron = layer["number_of_neurons"]
            self._activation_function = layer["activation_function"]
            self._weight = weight
            self._input = input
            self._input = np.insert(self._input, 0, 1) # Append bias of 1
            self._sigma = np.zeros(self._num_of_neuron)
            self._output = np.zeros(self._num_of_neuron)
            self._error = np.zeros(self._num_of_neuron)
            self._error_node_output = np.zeros(self._num_of_neuron)
            self._loss = 0
        except:
            print(f"[✕] Error initialiazing layer")
            print("")

            Driver.terminate(1)
```

Lakukan inisiasi layer dengan membentuk beberapa atribut. Berikut adalah penjelasan setiap atributnya:

- id: atribut ini mempresentasikan urutan layer yang sedang ditinjau.
- num_of_neuron: atribut ini merepresentasikan jumlah neuron yang ada di layer tersebut.
- activation_function: atribut ini merepresentasikan jenis activation function yang digunakan pada layer tersebut.
- weight : atribut ini mempresentasikan bobot yang digunakan untuk menghubungkan jaringan neuron antar layer dengan layer selanjutnya.
- input : atribut ini merepresentasikan data input (dapat berupa data input pertama kali, atau hasil dari fungsi aktivasi pada layer sebelumnya).

- sigma : atribut ini merepresentasikan jumlah bobot setelah dikalikan dengan data input berdasarkan atribut ‘input’.
- output: atribut ini merepresentasikan data yang dihasilkan setelah melalui pemrosesan pada layer tersebut.
- error: atribut ini merepresentasikan nilai error pada output layer.
- error_node_output: atribut ini merepresentasikan nilai error node output dalam suatu layer.
- loss: atribut ini merepresentasikan nilai loss dalam suatu layer, atau bisa dibilang dalam suatu data.

```python
# Calculate forward
def _calculate_forward(self):
    print(f"[•] Building layer {self._id}")

    # For each neuron, calculate its activation
    for i in range(self._num_of_neuron):
        self._calculate_sigma(i)
        self._calculate_activation(i)

    print(f"[✓] Successfully built layer {self._id}")
    print(f"Number of neurons: {self._num_of_neuron}")
    print(f"Activation function: {self._activation_function}")
    for i in range(len(self._output)):
        print(f"h{self._id}{i}: {self._output[i]}")
    print("")

    return self._output
```

Setelah menginisiasi layer, program ini akan melakukan perhitungan sigma dan activation untuk setiap neuron dengan penggunaan for loop dan pemanggilan fungsi calculate_sigma dan calculate_activation.

```python
# Calculate sigma
def _calculate_sigma(self, i):
    for j in range(len(self._input)):
        self._sigma[i] += self._input[j] * self._weight[j][i]
```

Fungsi ini digunakan untuk menghitung sigma pada setiap neuron dengan menggunakan rumus $w \bullet x$ atau $w_0 \cdot bias + w_1 \cdot x_1 + \ldots + w_n \cdot x_n$

```python
# Calculate activation
def _calculate_activation(self, i):
    if self._activation_function == 'relu':
        self._output[i] = max(0, self._sigma[i])
    elif self._activation_function == 'sigmoid':
        self._output[i] = 1 / (1 + np.exp(-self._sigma[i]))
    elif self._activation_function == 'linear':
        self._output[i] = self._sigma[i]
    elif self._activation_function == 'softmax':
        exp_values = np.exp(self._sigma - np.max(self._sigma))
        self._output = exp_values / np.sum(exp_values)
```

Setelah mendapatkan nilai sigma, hitung nilai aktivasi sesuai dengan fungsi aktivasinya:

- relu: $max(0, sigma)$
- sigmoid: $\frac{1}{1 + e^{-(sigma)}}$

- linear: $sigma$
- softmax: $\frac{e^{sigma}}{\Sigma i\ e^{sigma(i)}}$

Hasil dari fungsi aktivasi kemudian akan dipakai menjadi input pada layer selanjutnya. Kemudian proses-proses di atas diulang secara iteratif sampai mencapai output layer. Hasil fungsi aktivasi pada output layer inilah yang menjadi hasil dari pemrosesan FFNN.

```python
# Calculate Loss
loss = layer._calculate_loss(y_batch[i], self._layers[-1]["activation_function"] == 'softmax')
batch_loss += loss
```

```python
# Calculate loss
def _calculate_loss(self, target, isSoftmax):
    print("[•] Calculating Loss")

    if isSoftmax:
        self._loss = -np.sum(target * np.log(self._output) + 1e-9)
    else:
        self._loss = np.sum(target - self._output) ** 2
        self._loss = self._loss / 2

    print(f"[✓] Successfully calculated Loss")
    print(f"Loss: {self._loss}")
    print("")

    return self._loss
```

Untuk perhitungan loss, dimulai dari kita mencari loss masing-masing data dari sebuah batch terlebih dahulu. Setelah didapat nilai loss untuk masing-masing data, nilai loss pada sebuah batch dijumlahkan menjadi batch loss. Untuk perhitungan loss masing-masing data, kami menggunakan rumus dari spesifikasi dengan sedikit modifikasi pada perhitungan loss softmax. Rumus dari spek adalah sebagai berikut.

ReLU, sigmoid, dan linear	$E = \frac{1}{2} \sum_{k \in output} (t_k - o_k)^2$
Softmax	$E = -log(p_k),\ \text{k=target}$

```python
print(f"[•] Calculating batch loss")
batch_loss /= len(X_batch)
print(f"[✓] Successfully calculated batch loss")
print(f"Batch Loss: {batch_loss}")
print("")

self._loss += batch_loss
```

Setelah proses perhitungan batch selesai, jumlah loss per batch tadi dirata-ratakan dengan jumlah data di dalam batch tersebut. Kemudian untuk setiap batch loss dari 1 epoch dijumlahkan juga ke dalam loss.

```python
# Calculate loss
print(f"[•] Calculating epoch loss")
self._loss /= math.ceil(len(self._inputs) / self._batch_size)
print(f"[✓] Successfully calculated epoch loss")
print(f"Epoch Loss: {self._loss}")
if self._loss <= self._error_threshold:
    self._stopped_by = "error_threshold"
    print("")
    print(f"[✓] Loss threshold reached")
    print("\n")
    return
```

Setelah semua batch sudah dihitung, akumulasi nilai batch loss tadi dirata-ratakan dengan jumlah batch yang ada di satu epoch, kemudian kita akan mendapatkan nilai loss dalam 1 epoch. Di akhir perhitungan dalam 1 epoch, kita mengecek apakah nilai loss tersebut sudah di bawah threshold atau belum. Jika sudah dibawah threshold, maka proses iterasi akan berhenti.

```python
# Calculate Error Node Output
for i in range(len(X_batch)):
    for j in range(len(self._layers) - 1, -1, -1):

        print(f"[•] Evaluating data {i + 1}")

        if (j == len(self._layers) - 1):
            self._error_node_output.append(self._layer_objects[i][j]._calculate_error_node_ouput(y_batch[i], None))
        else:
            self._error_node_output.append(self._layer_objects[i][j]._calculate_error_node_ouput(self._layer_objects[i][j + 1]._error_node_output, self._layer_objects[i][j + 1]._weight))

        print(f"[✓] Successfully evaluated data {i + 1}")
        print("")
```

Setelah mendapatkan hasil FFNN pada sebuah batch, kita memulai langkah backpropagation kita dengan menghitung error node output yang ada di setiap data dan di setiap layer dengan memanggil fungsi calculate_error_node_output. Untuk output layer, parameter perhitungannya adalah berdasarkan target dari data, sedangkan untuk yang

selain output layer, parameter perhitungannya adalah berdasarkan error node output dan weights dari layer setelahnya.

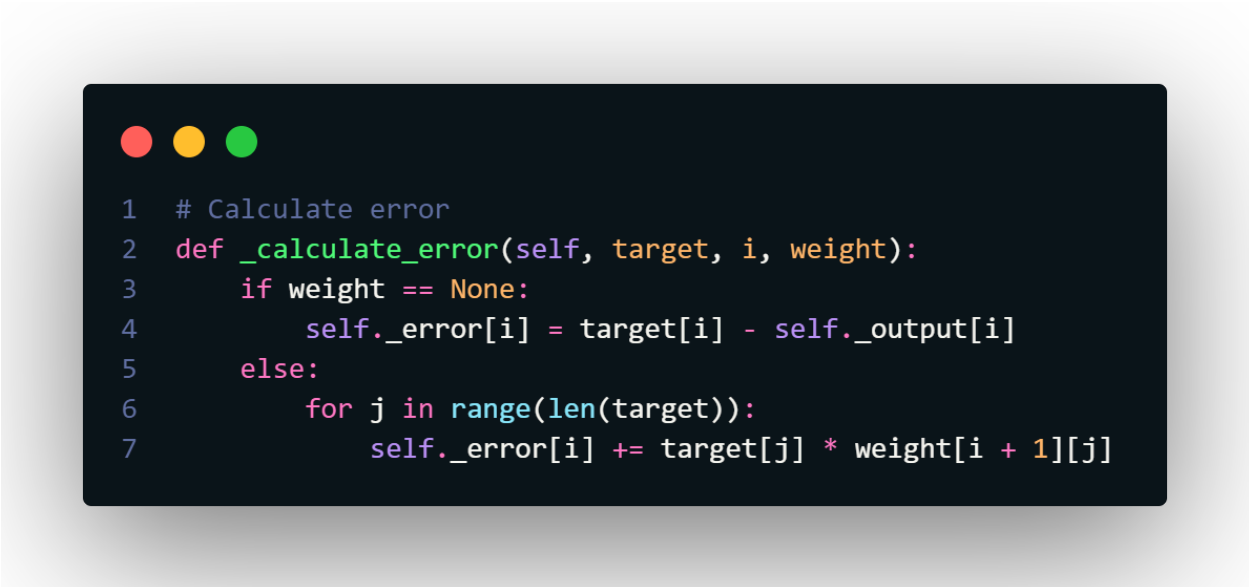
```python
# Calculate Error Node Output
def _calculate_error_node_ouput(self, target, weight):
    print(f"[•] Calculating Error Node Output for layer {self._id}")

    for i in range(self._num_of_neuron):
        self._calculate_error(target, i, weight)
        self._calculate_derivative(i)

    print(f"[✓] Successfully calculated Error Node Output for layer {self._id}")
    print(f"Derivative function: {self._activation_function}")
    for i in range(len(self._error_node_output)):
        print(f"e{self._id}{i}: {self._error_node_output[i]}")
    print("")

    return self._error_node_output
```

Pada fungsi ini, program akan melakukan perhitungan error dan nilai derivative untuk setiap neuron dengan penggunaan for loop dan pemanggilan fungsi calculate_error dan calculate_derivative.



```python
# Calculate error
def _calculate_error(self, target, i, weight):
    if weight == None:
        self._error[i] = target[i] - self._output[i]
    else:
        for j in range(len(target)):
            self._error[i] += target[j] * weight[i + 1][j]
```

Fungsi calculate_error menerima parameter target dari data untuk output layer, sedangkan untuk yang selain output layer, parameternya adalah berdasarkan error node output dan weights dari layer setelahnya. Untuk perhitungan error di output layer, cukup mengurangkan nilai target dengan nilai output yang didapat dari FFNN. Sedangkan untuk perhitungan error di selain output layer, lakukan dot product untuk nilai error node output layer setelahnya terhadap weight terkait pada layer setelahnya.

```python
# Calculate ENO
def _calculate_derivative(self, i):
    if self._activation_function == 'relu':
        self._error_node_output[i] = self._error[i] if self._sigma[i] > 0 else 0
    elif self._activation_function == 'sigmoid':
        self._error_node_output[i] = self._error[i] * (1 / (1 + np.exp(-self._sigma[i]))) * (1 - (1 / (1 + np.exp(-self._sigma[i]))))
    elif self._activation_function == 'linear':
        self._error_node_output[i] = self._error[i]
    elif self._activation_function == 'softmax':
        self._error_node_output[i] = self._error[i]
```

Setelah mendapatkan hasil error, kita akan mencari nilai error node output dengan mengalikan nilai hasil error dengan nilai derivative sigma pada layer tersebut. Berikut

adalah rumus untuk perhitungan nilai derivative-nya dengan x adalah sigma.

ReLU	$\frac{d}{dx}ReLU(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geqslant 0 \end{cases}$
Sigmoid	$\frac{d}{dx}\sigma(x) = \sigma(x) \times (1 - \sigma(x))$
Linear	$\frac{d}{dx}x = 1$
Softmax*	$\frac{\partial E_d}{\partial net(x)} = \begin{cases} p_j, & j \neq \text{target} \\ -(1 - p_j), & j = \text{target} \end{cases}$

```python
# Update weights
m = len(self._layers) - 1
print(f"[•] Updating weights")
for i in range(len(X_batch)):
    for j in range(len(self._layers)):
        for k in range(len(self._weights[j])):
            for l in range(len(self._weights[j][k])):
                if k == 0:
                    self._weights[j][k][l] += self._learning_rate * self._error_node_output[m][l]
                else:
                    if (j == 0):
                        self._weights[j][k][l] += self._learning_rate * self._error_node_output[m][l] * X_batch[i][k - 1]
                    else:
                        self._weights[j][k][l] += self._learning_rate * self._error_node_output[m][l] * self._layer_objects[i][j - 1]._output[k - 1]
        m -= 1
```

Kemudian kita akan melakukan updating weights sesuai dengan aturan mini batch, yaitu update weights setelah 1 batch selesai. Perhitungan nilai weights baru dipengaruhi oleh hasil error node output yang kita hitung pada perhitungan sebelumnya. Rumus yang dipakai adalah sebagai berikut.

$$\Delta w = -\text{gradient} \times \text{learning rate}$$

$$w_{new} = w_{old} + \Delta w$$

II. Hasil pengujian

1. linear.json

```
[✔] Successfully loaded model from 'models/linear.json'

[✔] Successfully initialized model
Number of features: 2
Number of layers: 1
Number of data: 2


[●] Epoch 1 ========================================

[●] Batch 1 ====================

[●] Building batch 1
[●] Evaluating data 1
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: linear
h00: 1.4000000000000004
h01: 0.10000000000000009
h02: -1.3999999999999997

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 0.04499999999999981

[✔] Successfully evaluated data 1

[●] Evaluating data 2
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: linear
h00: 0.7
h01: -1.1
h02: 0.5

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 0.18000000000000013
```

```
[✔] Successfully evaluated data 2

[●] Evaluating data 1
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: linear
e00: 0.5999999999999996
e01: 0.1999999999999999
e02: -0.5000000000000002

[✔] Successfully evaluated data 1

[●] Evaluating data 2
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: linear
e00: 0.6000000000000001
e01: 0.40000000000000013
e02: -0.4

[✔] Successfully evaluated data 2

[●] Updating weights
[✔] Successfully updated weights for batch 1
[[[0.21999999999999997, 0.36, 0.10999999999999999], [0.64,
0.30000000000000004, -0.8900000000000001], [0.28, -0.7,
0.36999999999999994]]]
[●] Calculating batch loss
[✔] Successfully calculated batch loss
Batch Loss: 0.11249999999999998

[✔] Successfully built batch 1

[●] Calculating epoch loss
[✔] Successfully calculated epoch loss
Epoch Loss: 0.11249999999999998


[●] Asserting output ========================================

Stopped by max_iteration
Test status: PASS
Test result: SSE (1.9451892438311082e-32) < MAX SSE (1e-07)
```

```
Final Weights :
[[[0.21999999999999997, 0.36, 0.10999999999999999], [0.64,
0.30000000000000004, -0.8900000000000001], [0.28, -0.7,
0.36999999999999994]]]
```

2. linear_two_iteration.json

```
[✔] Successfully loaded model from
'models/linear_two_iteration.json'

[✔] Successfully initialized model
Number of features: 2
Number of layers: 1
Number of data: 2


[●] Epoch 1 ========================================

[●] Batch 1 ====================

[●] Building batch 1
[●] Evaluating data 1
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: linear
h00: 1.4000000000000004
h01: 0.10000000000000009
h02: -1.3999999999999997

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 0.04499999999999981

[✔] Successfully evaluated data 1

[●] Evaluating data 2
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: linear
h00: 0.7
h01: -1.1
h02: 0.5

[●] Calculating Loss
```

```
[✔] Successfully calculated Loss
Loss: 0.18000000000000013

[✔] Successfully evaluated data 2

[●] Evaluating data 1
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: linear
e00: 0.5999999999999996
e01: 0.1999999999999999
e02: -0.5000000000000002

[✔] Successfully evaluated data 1

[●] Evaluating data 2
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: linear
e00: 0.6000000000000001
e01: 0.40000000000000013
e02: -0.4

[✔] Successfully evaluated data 2

[●] Updating weights
[✔] Successfully updated weights for batch 1
[[[0.21999999999999997, 0.36, 0.10999999999999999], [0.64,
0.30000000000000004, -0.8900000000000001], [0.28, -0.7,
0.36999999999999994]]]
[●] Calculating batch loss
[✔] Successfully calculated batch loss
Batch Loss: 0.11249999999999998

[✔] Successfully built batch 1

[●] Calculating epoch loss
[✔] Successfully calculated epoch loss
Epoch Loss: 0.11249999999999998


[●] Epoch 2 ======================================

[●] Batch 1 ====================

[●] Building batch 1
[●] Evaluating data 1
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: linear
```

```
h00: 2.42
h01: 0.5600000000000003
h02: -2.1900000000000004

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 0.07604999999999988

[✔] Successfully evaluated data 1

[●] Evaluating data 2
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: linear
h00: 1.42
h01: -0.7399999999999999
h02: -0.04000000000000026

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 0.0018000000000000182

[✔] Successfully evaluated data 2

[●] Evaluating data 1
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: linear
e00: -0.41999999999999993
e01: -0.2600000000000003
e02: 0.2900000000000005

[✔] Successfully evaluated data 1

[●] Evaluating data 2
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: linear
e00: -0.11999999999999988
e01: 0.039999999999999925
e02: 0.14000000000000026

[✔] Successfully evaluated data 2

[●] Updating weights
[✔] Successfully updated weights for batch 1
[[[0.166, 0.33799999999999997, 0.15300000000000008], [0.502,
0.22599999999999995, -0.7889999999999999], [0.21400000000000008,
-0.718, 0.42700000000000005]]]
[●] Calculating batch loss
```

```
[✔] Successfully calculated batch loss
Batch Loss: 0.03892499999999995

[✔] Successfully built batch 1

[●] Calculating epoch loss
[✔] Successfully calculated epoch loss
Epoch Loss: 0.03892499999999995


[●] Asserting output ========================================

Stopped by max_iteration
Test result: SSE (3.543711097672514e-32) < MAX SSE (1e-07)
Test status: PASS

Final Weights :
[[[0.166, 0.33799999999999997, 0.15300000000000008], [0.502,
0.22599999999999995, -0.7889999999999999], [0.21400000000000008,
-0.718, 0.4270000000000005]]]
```

3. linear_small_lr.json

```
[✔] Successfully loaded model from 'models/linear_small_lr.json'

[✔] Successfully initialized model
Number of features: 2
Number of layers: 1
Number of data: 2


[●] Epoch 1 ========================================

[●] Batch 1 ====================

[●] Building batch 1
[●] Evaluating data 1
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: linear
h00: 1.4000000000000004
h01: 0.10000000000000009
h02: -1.3999999999999997

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 0.04499999999999981
```

```
[✔] Successfully evaluated data 1

[●] Evaluating data 2
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: linear
h00: 0.7
h01: -1.1
h02: 0.5

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 0.18000000000000013

[✔] Successfully evaluated data 2

[●] Evaluating data 1
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: linear
e00: 0.5999999999999996
e01: 0.1999999999999999
e02: -0.5000000000000002

[✔] Successfully evaluated data 1

[●] Evaluating data 2
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: linear
e00: 0.6000000000000001
e01: 0.40000000000000013
e02: -0.4

[✔] Successfully evaluated data 2

[●] Updating weights
[✔] Successfully updated weights for batch 1
[[[0.10120000000000001, 0.3006, 0.1991], [0.40240000000000004,
0.201, -0.7018999999999999], [0.10180000000000002, -0.799,
0.4987]]]
[●] Calculating batch loss
[✔] Successfully calculated batch loss
Batch Loss: 0.11249999999999998

[✔] Successfully built batch 1

[●] Calculating epoch loss
[✔] Successfully calculated epoch loss
```

```
Epoch Loss: 0.11249999999999998


[●] Asserting output ========================================

Stopped by max_iteration
Test status: FAIL

Final Weights :
[[[0.10120000000000001, 0.3006, 0.1991], [0.40240000000000004, 0.201,
-0.7018999999999999], [0.10180000000000002, -0.799, 0.4987]]]
```

4. mlp.json

```
[✔] Successfully loaded model from 'models/mlp.json'

[✔] Successfully initialized model
Number of features: 2
Number of layers: 2
Number of data: 2


[●] Epoch 1 ========================================

[●] Batch 1 ====================

[●] Building batch 1
[●] Evaluating data 1
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 2
Activation function: linear
h00: 0.48000000000000004
h01: -0.19999999999999998

[●] Building layer 1
[✔] Successfully built layer 1
Number of neurons: 2
Activation function: relu
h10: 0.252
h11: 0.0

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 0.359552

[✔] Successfully evaluated data 1
```

```
[●] Evaluating data 2
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 2
Activation function: linear
h00: -0.36
h01: -0.19999999999999996

[●] Building layer 1
[✔] Successfully built layer 1
Number of neurons: 2
Activation function: relu
h10: 0.0
h11: 0.12000000000000005

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 0.48019999999999985

[✔] Successfully evaluated data 2

[●] Evaluating data 1
[●] Calculating Error Node Output for layer 1
[✔] Successfully calculated Error Node Output for layer 1
Derivative function: relu
e10: 0.748
e11: 0.0

[✔] Successfully evaluated data 1

[●] Evaluating data 1
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: linear
e00: 0.2992
e01: 0.5236

[✔] Successfully evaluated data 1

[●] Evaluating data 2
[●] Calculating Error Node Output for layer 1
[✔] Successfully calculated Error Node Output for layer 1
Derivative function: relu
e10: 0.0
e11: 0.8799999999999999

[✔] Successfully evaluated data 2

[●] Evaluating data 2
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
```

```
Derivative function: linear
e00: -0.43999999999999995
e01: 0.704

[✔] Successfully evaluated data 2

[●] Updating weights
[✔] Successfully updated weights for batch 1
[[[0.08592000000000001, 0.32276000000000005],
[-0.33871999999999997, 0.46171999999999996], [0.449984, 0.440072]],
[[0.27480000000000004, 0.188], [0.435904, -0.53168], [0.68504,
0.78240000000000001]]]
[●] Calculating batch loss
[✔] Successfully calculated batch loss
Batch Loss: 0.4198759999999999

[✔] Successfully built batch 1

[●] Calculating epoch loss
[✔] Successfully calculated epoch loss
Epoch Loss: 0.4198759999999999


[●] Asserting output ========================================

Stopped by max_iteration
Test result: SSE (2.484449628259534e-32) < MAX SSE (1e-07)
Test status: PASS

Final Weights :
[[[0.08592000000000001, 0.32276000000000005], [-0.33871999999999997,
0.46171999999999996], [0.449984, 0.440072]], [[0.27480000000000004,
0.188], [0.435904, -0.53168], [0.68504, 0.78240000000000001]]]
```

5. relu_b.json

```
[✔] Successfully loaded model from 'models/relu_b.json'

[✔] Successfully initialized model
Number of features: 2
Number of layers: 1
Number of data: 2


[●] Epoch 1 ========================================

[●] Batch 1 ====================
```

```
[●] Building batch 1
[●] Evaluating data 1
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: relu
h00: 0.0
h01: 0.0
h02: 1.9

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 0.2449999999999998

[✔] Successfully evaluated data 1

[●] Evaluating data 2
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: relu
h00: 0.20999999999999996
h01: 1.05
h02: 0.35

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 0.08405000000000001

[✔] Successfully evaluated data 2

[●] Evaluating data 1
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: relu
e00: 0.0
e01: 0.0
e02: -1.7999999999999998

[✔] Successfully evaluated data 1

[●] Evaluating data 2
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: relu
e00: -0.10999999999999996
e01: -0.9500000000000001
e02: 0.65

[✔] Successfully evaluated data 2
```

```
[●] Updating weights
[✔] Successfully updated weights for batch 1
[[[-0.21100000000000002, 0.105, 0.885], [0.3033, 0.5285, 0.3005],
[-0.489, -0.905, 0.291]]]
[●] Calculating batch loss
[✔] Successfully calculated batch loss
Batch Loss: 0.16452499999999992

[✔] Successfully built batch 1

[●] Calculating epoch loss
[✔] Successfully calculated epoch loss
Epoch Loss: 0.16452499999999992


[●] Asserting output ========================================

Stopped by max_iteration
Test result: SSE (7.703719777548943e-34) < MAX SSE (1e-07)
Test status: PASS

Final Weights :
[[[-0.21100000000000002, 0.105, 0.885], [0.3033, 0.5285, 0.3005],
[-0.489, -0.905, 0.291]]]
```

6. softmax.json

```
[✔] Successfully loaded model from 'models/softmax.json'

[✔] Successfully initialized model
Number of features: 8
Number of layers: 1
Number of data: 3


[●] Epoch 1 ========================================

[●] Batch 1 ====================

[●] Building batch 1
[●] Evaluating data 1
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: softmax
h00: 0.752992052861625
h01: 0.24470531086173575
h02: 0.0023026362766392482
```

```
[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 1.4077006022065623

[✔] Successfully evaluated data 1

[●] Evaluating data 1
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: softmax
e00: -0.752992052861625
e01: 0.7552946891382643
e02: -0.0023026362766392482

[✔] Successfully evaluated data 1

[●] Updating weights
[✔] Successfully updated weights for batch 1
[[[0.09247007947138376, 0.9075529468913827, -0.1000230263627664],
[-0.18192819073132102, 0.7818729274606817, 0.20005526327063936],
[0.32093317906955315, -0.7209971923580437, 0.3000640132884906],
[0.40451795231716975, 0.5954682318651704, -0.39998618418234017],
[0.497213929404412, 0.5027945903498116, 0.49999148024577644],
[-0.6185236045003959, 0.4185802493528013, 0.5999433551475947],
[-0.693072473113673, -0.306948711140072, 0.7000211842537745],
[0.7792174193410192, 0.2208461334202161, -0.8000635527612353],
[0.8802716082150255, -0.08021127914457749,
-6.032907044794831e-05]]]
[●] Calculating batch loss
[✔] Successfully calculated batch loss
Batch Loss: 1.4077006022065623

[✔] Successfully built batch 1

[●] Batch 2 ====================

[●] Building batch 2
[●] Evaluating data 1
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: softmax
h00: 0.004457910204553203
h01: 0.0007803907177990602
h02: 0.9947616990776477

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 5.413075183762103
```

```
[✔] Successfully evaluated data 1

[●] Evaluating data 1
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: softmax
e00: 0.9955420897954468
e01: -0.0007803907177990602
e02: -0.9947616990776477

[✔] Successfully evaluated data 1

[●] Updating weights
[✔] Successfully updated weights for batch 2
[[[0.10242550036933823, 0.9075451429842046, -0.10997064335354288],
[-0.19974839413865952, 0.7818868964545304, 0.21786149768412924],
[0.33735962355117805, -0.7210100688048874, 0.2836504452537094],
[0.3968522782257448, 0.5954742408736975, -0.3923265190994423],
[0.4869598458795189, 0.5028026283742049, 0.5102375257462762],
[-0.6175280624106004, 0.4185794689620835, 0.598948593448517],
[-0.6719669808100096, -0.30696525542328934, 0.6789322362332989],
[0.7557226260218466, 0.22086455064115618, -0.7765871766630028],
[0.8927158843374685, -0.08022103402854998, -0.01249485030891854 7]]]
[●] Calculating batch loss
[✔] Successfully calculated batch loss
Batch Loss: 5.413075183762103

[✔] Successfully built batch 2

[●] Batch 3 ====================

[●] Building batch 3
[●] Evaluating data 1
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: softmax
h00: 0.4600442160486412
h01: 0.28690869720037165
h02: 0.25304708675098697

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 1.374179690931347

[✔] Successfully evaluated data 1

[●] Evaluating data 1
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: softmax
```

```
e00: -0.4600442160486412
e01: -0.28690869720037165
e02: 0.746952913249013

[✔] Successfully evaluated data 1

[●] Updating weights
[✔] Successfully updated weights for batch 3
[[[0.09782505820885182, 0.9046760560122009, -0.10250111422105275],
[-0.2073391237034621, 0.7771529029507243, 0.23018622075273795],
[0.32659458889563986, -0.727723732319376, 0.3011291434237363],
[0.39561015884241346, 0.5946995873912565, -0.39030974623366993],
[0.4761948112239807, 0.49608896485971615, 0.527716223916303 1],
[-0.6199202923340533, 0.41708754373664153, 0.6028327485974119],
[-0.678269586569876, -0.31089590457493443, 0.6891654911448104],
[0.7475798433977856, 0.2157862667007096, -0.7633661100984953],
[0.8898636101979669, -0.08199986795119228, -0.007863742246774667]]]
[●] Calculating batch loss
[✔] Successfully calculated batch loss
Batch Loss: 1.374179690931347

[✔] Successfully built batch 3

[●] Calculating epoch loss
[✔] Successfully calculated epoch loss
Epoch Loss: 2.7316518256333375


[●] Epoch 2 ========================================

[●] Batch 1 ====================

[●] Building batch 1
[●] Evaluating data 1
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: softmax
h00: 0.629657124207718
h01: 0.3676680802540851
h02: 0.002674795538196836

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 1.0005747006514105

[✔] Successfully evaluated data 1

[●] Evaluating data 1
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
```

```
Derivative function: softmax
e00: -0.629657124207718
e01: 0.632331919745915
e02: -0.002674795538196836

[✔] Successfully evaluated data 1

[●] Updating weights
[✔] Successfully updated weights for batch 1
[[[0.09152848696677464, 0.91099937520966, -0.10252786217643471],
[-0.19222735272247687, 0.7619769368768223, 0.23025041584565467],
[0.3440990569486144, -0.7453025596883125, 0.3012035027396982],
[0.39938810158765975, 0.590905595872781, -0.39029369746044074],
[0.4738650798644122, 0.49842859296277603, 0.5277063271728119],
[-0.6354098575895631, 0.43264290896239105, 0.6027669486271723],
[-0.672476741027165, -0.31671335823659685, 0.6891900992637618],
[0.7302013067696526, 0.23323862768569686, -0.7634399344553495],
[0.8733665935437247, -0.06543277165384931, -0.007933821889875424]]]
[●] Calculating batch loss
[✔] Successfully calculated batch loss
Batch Loss: 1.0005747006514105

[✔] Successfully built batch 1

[●] Batch 2 ====================

[●] Building batch 2
[●] Evaluating data 1
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: softmax
h00: 0.006920014108788463
h01: 0.0009379466819150773
h02: 0.9921420392092966

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 4.973337467512374

[✔] Successfully evaluated data 1

[●] Evaluating data 1
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: softmax
e00: 0.9930799858912115
e01: -0.0009379466819150773
e02: -0.9921420392092966

[✔] Successfully evaluated data 1
```

```
[●] Updating weights
[✔] Successfully updated weights for batch 2
[[[0.10145928682568675, 0.9109899957428409, -0.112449282568527 68],
[-0.21000348446992956, 0.7619937261224287, 0.248009758347501 08],
[0.3604848767158194, -0.745318035808564, 0.2848331590927448],
[0.3917413856962974, 0.5909128180622317, -0.382654203758529 15],
[0.4636363560097327, 0.49843825381359974, 0.5379253901766676],
[-0.6344167776036719, 0.4326419710157091, 0.601774806587963],
[-0.6514234453262714, -0.31673324270625347, 0.6681566880325247],
[0.70676461910262, 0.23326076322739006, -0.7400253823300101],
[0.8857800933673649, -0.06544449598737324, -0.020335597379991632]]]
[●] Calculating batch loss
[✔] Successfully calculated batch loss
Batch Loss: 4.973337467512374

[✔] Successfully built batch 2

[●] Batch 3 ====================

[●] Building batch 3
[●] Evaluating data 1
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: softmax
h00: 0.42635894507323585
h01: 0.2709072904840207
h02: 0.3027337644427434

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 1.1949015219307813

[✔] Successfully evaluated data 1

[●] Evaluating data 1
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: softmax
e00: -0.42635894507323585
e01: -0.2709072904840207
e02: 0.6972662355572565

[✔] Successfully evaluated data 1

[●] Updating weights
[✔] Successfully updated weights for batch 3
[[[0.0971956973749544, 0.9082809228380007, -0.10547662021295512],
[-0.21703840706363794, 0.7575237558294423, 0.2595146512341958],
[0.3505080774011057, -0.7516572664058901, 0.3011491890047846],
```

```
[0.3905902165445997, 0.5901813683779249, -0.38077158492252455],
[0.45365955669501895, 0.49209902321627363, 0.5542414200887075],
[-0.6366338441180527, 0.4312332531051922, 0.6054005910128607],
[-0.6572645628737748, -0.32044467258588455, 0.6777092354596591],
[0.6992180657748237, 0.2284657041858229, -0.7276837699606467],
[0.8831366679079108, -0.06712412118837417, -0.01601254671953664]]]
[●] Calculating batch loss
[✔] Successfully calculated batch loss
Batch Loss: 1.1949015219307813

[✔] Successfully built batch 3

[●] Calculating epoch loss
[✔] Successfully calculated epoch loss
Epoch Loss: 2.389604563364855


[●] Epoch 3 ========================================

[●] Batch 1 ====================

[●] Building batch 1
[●] Evaluating data 1
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: softmax
h00: 0.5088455218820439
h01: 0.488279043866845
h02: 0.002875434251111076

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 0.7168682223481698

[✔] Successfully evaluated data 1

[●] Evaluating data 1
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: softmax
e00: -0.5088455218820439
e01: 0.511720956133155
e02: -0.002875434251111076

[✔] Successfully evaluated data 1

[●] Updating weights
[✔] Successfully updated weights for batch 1
[[[0.09210724215613396, 0.9133981323993322, -0.10550537455546623],
[-0.20482611453846888, 0.7452424528822466, 0.2595836616562225],
```

```
[0.3646539829094265, -0.7658831089863919, 0.3012291260769655],
[0.39364328967589196, 0.587111042641126, -0.3807543323170179],
[0.4517768282640554, 0.4939923907539663, 0.5542307809819783],
[-0.6491514439563509, 0.44382158862606785, 0.6053298553302834],
[-0.65258318407246, -0.32515250538230955, 0.6777356894547694],
[0.6851739293708793, 0.24258920257509797, -0.7277631319459773],
[0.8698049152346012, -0.05371703213768551, -0.0160878830969157511]]
[●] Calculating batch loss
[✔] Successfully calculated batch loss
Batch Loss: 0.7168682223481698

[✔] Successfully built batch 1

[●] Batch 2 ====================

[●] Building batch 2
[●] Evaluating data 1
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: softmax
h00: 0.01065268090138435
h01: 0.0011299678773485809
h02: 0.9882173512212671

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 4.54194368767532

[✔] Successfully evaluated data 1

[●] Evaluating data 1
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: softmax
e00: 0.9893473190986156
e01: -0.0011299678773485809
e02: -0.9882173512212671

[✔] Successfully evaluated data 1

[●] Updating weights
[✔] Successfully updated weights for batch 2
[[[0.10200071534712012, 0.9133868327205588, -0.1153875480676789],
[-0.22253543155033412, 0.7452626793072511, 0.2772727522430832],
[0.3809782136745537, -0.7659017534563681, 0.2849235397818146],
[0.3860253153188326, 0.5871197433937816, -0.3731450587126141],
[0.44158655087733967, 0.49400402942310295, 0.5644094196995574],
[-0.6481620966372523, 0.44382045865819053, 0.6043416379790622],
[-0.6316090209075693, -0.32517646070130934, 0.6567854816088785],
[0.661825332640152, 0.2426158698170034, -0.7044412024571554],
```

```
[0.8821717567233339, -0.05373115673615236, -0.028440599987181591]]]
[●] Calculating batch loss
[✔] Successfully calculated batch loss
Batch Loss: 4.54194368767532

[✔] Successfully built batch 2

[●] Batch 3 ====================

[●] Building batch 3
[●] Evaluating data 1
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: softmax
h00: 0.39185585226410424
h01: 0.25411408040018124
h02: 0.35403006733571446

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 1.0383734304786238

[✔] Successfully evaluated data 1

[●] Evaluating data 1
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: softmax
e00: -0.39185585226410424
e01: -0.25411408040018124
e02: 0.6459699326642856

[✔] Successfully evaluated data 1

[●] Updating weights
[✔] Successfully updated weights for batch 3
[[[0.09808215682447907, 0.910845691916557, -0.10892784874103605],
[-0.22900105311269184, 0.7410697969806481, 0.2879312561320439],
[0.37180878673157364, -0.7718480229377324, 0.3000392362061589],
[0.38496730451771954, 0.5864336353767011, -0.3714009398944205],
[0.4324171239343596, 0.4880577599417387, 0.5795251161239017],
[-0.6501997470690257, 0.4424990654401096, 0.6077006816289164],
[-0.6369774460835875, -0.32865782360279183, 0.6656352696863792],
[0.6548894840550773, 0.2381180505939202, -0.6930075346489974],
[0.8797422504392964, -0.05530666403463349, -0.024435586404663021]]]
[●] Calculating batch loss
[✔] Successfully calculated batch loss
Batch Loss: 1.0383734304786238

[✔] Successfully built batch 3
```

```
[●] Calculating epoch loss
[✔] Successfully calculated epoch loss
Epoch Loss: 2.0990617801673714


[●] Epoch 4 ======================================

[●] Batch 1 ====================

[●] Building batch 1
[●] Evaluating data 1
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: softmax
h00: 0.40783829781820774
h01: 0.5892232495550347
h02: 0.0029384526267574296

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 0.5289501326422388

[✔] Successfully evaluated data 1

[●] Evaluating data 1
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: softmax
e00: -0.40783829781820774
e01: 0.4107767504449653
e02: -0.0029384526267574296

[✔] Successfully evaluated data 1

[●] Updating weights
[✔] Successfully updated weights for batch 1
[[[0.09400377384629699, 0.9149534594210066, -0.10895723326730362],
[-0.21921293396505484, 0.7312111549699689, 0.28800177899508606],
[0.38314669141091984, -0.7832676166001024, 0.30012092518918276],
[0.3874143343046288, 0.5839689748740313, -0.37138330917865997],
[0.43090812223243224, 0.4895776339183851, 0.5795142438491827],
[-0.6602325691953537, 0.45260417350105575, 0.6076283956942982],
[-0.63322533374366, -0.3324369697068855, 0.6656623034505453],
[0.6436331470352947, 0.24945548890620123, -0.693088635941496],
[0.8690568870364594, -0.0445443131729754, -0.024512573863484065]]]
[●] Calculating batch loss
[✔] Successfully calculated batch loss
Batch Loss: 0.5289501326422388
```

```
[✔] Successfully built batch 1

[●] Batch 2 ====================

[●] Building batch 2
[●] Evaluating data 1
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: softmax
h00: 0.016256744217055554
h01: 0.001362131726108723
h02: 0.9823811240568359

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 4.119247424565116

[✔] Successfully evaluated data 1

[●] Evaluating data 1
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: softmax
e00: 0.9837432557829444
e01: -0.001362131726108723
e02: -0.9823811240568359

[✔] Successfully evaluated data 1

[●] Updating weights
[✔] Successfully updated weights for batch 2
[[[0.10384120640412643, 0.9149398381037455, -0.11878104450787198],
[-0.23682193824356954, 0.7312355371278663, 0.3055864011157034],
[0.3993784551313384, -0.7832900917735831, 0.28391163664224495],
[0.3798395112351001, 0.5839794632883224, -0.3638189745234223],
[0.4207755666978679, 0.489591663875164, 0.5896327694269681],
[-0.6592488259395707, 0.4526028113693296, 0.6066460145702414],
[-0.6123699767210615, -0.332465846899479, 0.6448358236205404],
[0.6204168061988172, 0.2494876352149374, -0.6699044414137546],
[0.8813536777337463, -0.04456133981955176, -0.03679233791419451]]]
[●] Calculating batch loss
[✔] Successfully calculated batch loss
Batch Loss: 4.119247424565116

[✔] Successfully built batch 2

[●] Batch 3 ====================

[●] Building batch 3
[●] Evaluating data 1
```

```
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: softmax
h00: 0.35785416814659776
h01: 0.2370430187438402
h02: 0.405102813109562

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 0.9036143815626934

[✔] Successfully evaluated data 1

[●] Evaluating data 1
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: softmax
e00: -0.35785416814659776
e01: -0.2370430187438402
e02: 0.594897186890438

[✔] Successfully evaluated data 1

[●] Updating weights
[✔] Successfully updated weights for batch 3
[[[0.10026266472266046, 0.912569407916307, -0.1128320726389676],
[-0.2427265320179884, 0.7273243273185929, 0.3154022046993956],
[0.39100466759670804, -0.788836898412189, 0.2978322308154812],
[0.37887330498110433, 0.583339447137714, -0.3622127521188l815],
[0.41240177916323756, 0.48404485723655816, 0.6035533636002043],
[-0.661109667613933, 0.45137018767186166, 0.6097394799420717],
[-0.6172725788246699, -0.3357133362562696, 0.6529859150809394],
[0.6140827874226225, 0.2452919737831714, -0.6593747612057939],
[0.8791349818912374, -0.046031006535763565, -0.0331039753554738]]]
[●] Calculating batch loss
[✔] Successfully calculated batch loss
Batch Loss: 0.9036143815626934

[✔] Successfully built batch 3

[●] Calculating epoch loss
[✔] Successfully calculated epoch loss
Epoch Loss: 1.850603979590016


[●] Epoch 5 ========================================

[●] Batch 1 ====================

[●] Building batch 1
```

```
[●] Evaluating data 1
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: softmax
h00: 0.3299927429125378
h01: 0.6670810524472469
h02: 0.002926204640215245

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 0.4048437195373022

[✔] Successfully evaluated data 1

[●] Evaluating data 1
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: softmax
e00: -0.3299927429125378
e01: 0.3329189475527531
e02: -0.002926204640215245

[✔] Successfully evaluated data 1

[●] Updating weights
[✔] Successfully updated weights for batch 1
[[[0.09696273729353508, 0.9158985973918345, -0.11286133468536975],
[-0.2348067061880875, 0.7193342725773268, 0.31547243361076077],
[0.4001784658496766, -0.7980920451541555, 0.2979135793044792],
[0.38085326143857956, 0.5813419334523975, -0.36219519489097685],
[0.41118080601446116, 0.48527665734250336, 0.6035425366430355],
[-0.6692274890895814, 0.4595599937816594, 0.6096674953079224],
[-0.6142366455898746, -0.3387761905737549, 0.6530128361636294],
[0.6049749877182364, 0.25448053673562737, -0.6594555244538638],
[0.8704891720269289, -0.03730853010988143, -0.03318064191704743 5]]]
[●] Calculating batch loss
[✔] Successfully calculated batch loss
Batch Loss: 0.4048437195373022

[✔] Successfully built batch 1

[●] Batch 2 ====================

[●] Building batch 2
[●] Evaluating data 1
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: softmax
h00: 0.02457693026874535
```

```
h01: 0.0016392930128811658
h02: 0.9737837767183736

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 3.7059470670011203

[✔] Successfully evaluated data 1

[●] Evaluating data 1
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: softmax
e00: 0.9754230697312547
e01: -0.0016392930128811658
e02: -0.9737837767183736

[✔] Successfully evaluated data 1

[●] Updating weights
[✔] Successfully updated weights for batch 2
[[[0.10671696799084762, 0.9158822044617058, -0.12259917245255349],
[-0.25226677913627693, 0.7193636159222574, 0.33290316321401964],
[0.41627294650024227, -0.7981190934888681, 0.28184614698862603],
[0.3733425038016489, 0.5813545560085966, -0.35469705981024535],
[0.40113394839622923, 0.48529354206053604, 0.6135725095432347],
[-0.6682520660198501, 0.4595583544886465, 0.608693711531204],
[-0.593557676511572, -0.33881094358562797, 0.6323686200971999],
[0.5819550032725789, 0.25451922405073135, -0.6364742273233102],
[0.8826819603985696, -0.037329021272542444, -0.0453529391260271]]]
[●] Calculating batch loss
[✔] Successfully calculated batch loss
Batch Loss: 3.7059470670011203

[✔] Successfully built batch 2

[●] Batch 3 ====================

[●] Building batch 3
[●] Evaluating data 1
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: softmax
h00: 0.32544493405274133
h01: 0.22018416214708847
h02: 0.4543709038001702

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 0.7888414427552469
```

```
[✔] Successfully evaluated data 1

[●] Evaluating data 1
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: softmax
e00: -0.32544493405274133
e01: -0.22018416214708847
e02: 0.5456290961998298

[✔] Successfully evaluated data 1

[●] Updating weights
[✔] Successfully updated weights for batch 3
[[[0.1034625186503202, 0.9136803628402349, -0.11714288149055518],
[-0.2576366205481472, 0.7157305772468304, 0.34190604330131685],
[0.4086575350434081, -0.8032714028831099, 0.29461386783970206],
[0.3724638024797065, 0.5807600587707995, -0.3532238612505058],
[0.3935185369393951, 0.48014123266629416, 0.6263402303943107],
[-0.6699443796769243, 0.45841339684548166, 0.6115309828314431],
[-0.5980162721080945, -0.34182746660704305, 0.6398437387151376],
[0.5761946279398453, 0.25062196438072787, -0.6268165923205732],
[0.8806642018074426, -0.03869416307785439, -0.04197003872958815 5]]]
[●] Calculating batch loss
[✔] Successfully calculated batch loss
Batch Loss: 0.7888414427552469

[✔] Successfully built batch 3

[●] Calculating epoch loss
[✔] Successfully calculated epoch loss
Epoch Loss: 1.63321074309789


[●] Epoch 6 ======================================

[●] Batch 1 ====================

[●] Building batch 1
[●] Evaluating data 1
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: softmax
h00: 0.27148438673174335
h01: 0.7256321838376995
h02: 0.0028834294305571987

[●] Calculating Loss
[✔] Successfully calculated Loss
```

```
Loss: 0.32071202337199456

[✔] Successfully evaluated data 1

[●] Evaluating data 1
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: softmax
e00: -0.27148438673174335
e01: 0.27436781616230055
e02: -0.0028834294305571987

[✔] Successfully evaluated data 1

[●] Updating weights
[✔] Successfully updated weights for batch 1
[[[0.10074767478300277, 0.9164240410018579, -0.11717171578486076],
[-0.2511209952665853, 0.7091457496589352, 0.34197524560765025],
[0.4162048009945506, -0.8108988281724219, 0.2946940271778716],
[0.374092708800097, 0.5791138518738257, -0.3532065606739225],
[0.39251404470848766, 0.4811563935860947, 0.6263295617054176],
[-0.6766228955905252, 0.46516284512307426, 0.6114600504674514],
[-0.5955186157501624, -0.34435165051573624, 0.6398702662658987],
[0.5687016588660492, 0.2581945161068074, -0.6268961749728565],
[0.873551310875071, -0.03150572629440212, -0.04204558458066875]]]
[●] Calculating batch loss
[✔] Successfully calculated batch loss
Batch Loss: 0.32071202337199456

[✔] Successfully built batch 1

[●] Batch 2 ====================

[●] Building batch 2
[●] Evaluating data 1
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: softmax
h00: 0.03675176080742558
h01: 0.0019644734472237762
h02: 0.9612837657453506

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 3.3035691383438617

[✔] Successfully evaluated data 1

[●] Evaluating data 1
[●] Calculating Error Node Output for layer 0
```

```
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: softmax
e00: 0.9632482391925744
e01: -0.0019644734472237762
e02: -0.9612837657453506

[✔] Successfully evaluated data 1

[●] Updating weights
[✔] Successfully updated weights for batch 2
[[[0.11038015717492851, 0.9164043962673857, -0.12678455344231426],
[-0.2683631387481324, 0.7091809137336405, 0.35918222501449204],
[0.4320983969412281, -0.8109312419843011, 0.27883284504307326],
[0.3666756973583142, 0.5791289783193693, -0.34580467567768325],
[0.38259258784480416, 0.4811766276626011, 0.6362307844925947],
[-0.6756596473513327, 0.465160880649627, 0.6104987667017061],
[-0.5750977530792798, -0.3443932973528174, 0.6194910504320973],
[0.5459690004211044, 0.25824087768016185, -0.6042098781012663],
[0.8855919138649782, -0.03153028221249241, -0.05406163165248563 4]]]
[●] Calculating batch loss
[✔] Successfully calculated batch loss
Batch Loss: 3.3035691383438617

[✔] Successfully built batch 2

[●] Batch 3 ====================

[●] Building batch 3
[●] Evaluating data 1
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: softmax
h00: 0.295359646773063
h01: 0.20395973667308892
h02: 0.5006806165538481

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 0.6917868700901251

[✔] Successfully evaluated data 1

[●] Evaluating data 1
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: softmax
e00: -0.295359646773063
e01: -0.20395973667308892
e02: 0.4993193834461519
```

```
[✔] Successfully evaluated data 1

[●] Updating weights
[✔] Successfully updated weights for batch 3
[[[0.10742656070719787, 0.9143647989006548, -0.12179135960785274],
[-0.27323657291988795, 0.7058155780785345, 0.36742099484135354],
[0.4251869812067384, -0.8157038998224514, 0.2905169186157132],
[0.36587822631202693, 0.578578287030352, -0.34445651334237865],
[0.3756811721103145, 0.4764039698244508, 0.6479148580652347],
[-0.6771955175145525, 0.46410029001892694, 0.6130952274956261],
[-0.5791441802400707, -0.3471875457452387, 0.6263317259853096],
[0.5407411346732212, 0.2546307903410482, -0.5953719250142694],
[0.8837606840549852, -0.03279483257986556, -0.05096585147511949]]]
[●] Calculating batch loss
[✔] Successfully calculated batch loss
Batch Loss: 0.6917868700901251

[✔] Successfully built batch 3

[●] Calculating epoch loss
[✔] Successfully calculated epoch loss
Epoch Loss: 1.4386893439353272


[●] Epoch 7 ========================================

[●] Batch 1 ====================

[●] Building batch 1
[●] Evaluating data 1
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: softmax
h00: 0.22734599370301667
h01: 0.7698208004228146
h02: 0.002833205874168632

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 0.2615975149430911

[✔] Successfully evaluated data 1

[●] Evaluating data 1
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: softmax
e00: -0.22734599370301667
e01: 0.23017919957718536
e02: -0.002833205874168632
```

```
[✔] Successfully evaluated data 1

[●] Updating weights
[✔] Successfully updated weights for batch 1
[[[0.1051531007701677, 0.9166665908964267, -0.12181969166659443],
[-0.26778026907101554, 0.700291277288682, 0.3674889917823336],
[0.4315071998316823, -0.8221028815706971, 0.2905956817390151],
[0.36724230227424504, 0.5771972118328889, -0.34443951410713364],
[0.37483999193361334, 0.4772556328628864, 0.6479043752035003],
[-0.6827882289596467, 0.4697626983285257, 0.6130255306311215],
[-0.577052597098003, -0.34930519438134877, 0.626357791479352],
[0.534466385247018, 0.2609837362493785, -0.5954501214963964],
[0.8778042190199662, -0.026764137550943305, -0.05104008146902271]]]
[●] Calculating batch loss
[✔] Successfully calculated batch loss
Batch Loss: 0.2615975149430911

[✔] Successfully built batch 1

[●] Batch 2 ====================

[●] Building batch 2
[●] Evaluating data 1
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: softmax
h00: 0.054219854367190075
h01: 0.002337053032434812
h02: 0.9434430926003751

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 2.9147081178763075

[✔] Successfully evaluated data 1

[●] Evaluating data 1
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: softmax
e00: 0.9457801456328099
e01: -0.002337053032434812
e02: -0.9434430926003751

[✔] Successfully evaluated data 1

[●] Updating weights
[✔] Successfully updated weights for batch 2
[[[0.11461090222649581, 0.9166432203661024, -0.13125412259259817],
```

```
[-0.28470973367784286, 0.7003331105379625, 0.3843766231398803],
[0.44711257223462364, -0.8221414429457323, 0.27502887071110893],
[0.3599597951528724, 0.5772152071412386, -0.337175002294110741],
[0.3650984564335954, 0.4772797045091205, 0.6576218390572842],
[-0.6818424488140139, 0.46976036127549325, 0.6120820875385211],
[-0.5570020580105874, -0.34935473990563637, 0.6063567979162241],
[0.5121459738100836, 0.26103889070094394, -0.5731848645110276],
[0.8896264708403763, -0.02679335071384874, -0.06283312012652741]]]
[●] Calculating batch loss
[✔] Successfully calculated batch loss
Batch Loss: 2.9147081178763075

[✔] Successfully built batch 2

[●] Batch 3 ====================

[●] Building batch 3
[●] Evaluating data 1
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: softmax
h00: 0.2679820240557137
h01: 0.18867992515386597
h02: 0.5433380507904204

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 0.6100235884507742

[✔] Successfully evaluated data 1

[●] Evaluating data 1
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: softmax
e00: -0.2679820240557137
e01: -0.18867992515386597
e02: 0.4566619492095796

[✔] Successfully evaluated data 1

[●] Updating weights
[✔] Successfully updated weights for batch 3
[[[0.11193108198593867, 0.9147564211145637, -0.12668750310050236],
[-0.28913143707476213, 0.6972198917729238, 0.39191154530183836],
[0.44084179287171993, -0.8265565531943327, 0.2857147603226131],
[0.359236243687922, 0.5767057713433232, -0.3359420150312449],
[0.3588276770706917, 0.47286459426052, 0.6683077286687884],
[-0.6832359553391036, 0.46877922566469316, 0.6144567296744109],
[-0.5606734117401507, -0.3519396548802443, 0.6126130666203954],
```

```
[0.5074026919842975, 0.2576992560257205, -0.565101948010018],
[0.8879649822912309, -0.02796316624980271, -0.06000181604142801]]]
[●] Calculating batch loss
[✔] Successfully calculated batch loss
Batch Loss: 0.6100235884507742

[✔] Successfully built batch 3

[●] Calculating epoch loss
[✔] Successfully calculated epoch loss
Epoch Loss: 1.262109740423391


[●] Epoch 8 ======================================

[●] Batch 1 ====================

[●] Building batch 1
[●] Evaluating data 1
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: softmax
h00: 0.1935462003266874
h01: 0.8036684404380204
h02: 0.0027853592352923007

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 0.21856847937318644

[✔] Successfully evaluated data 1

[●] Evaluating data 1
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: softmax
e00: -0.1935462003266874
e01: 0.19633155956197956
e02: -0.0027853592352923007

[✔] Successfully evaluated data 1

[●] Updating weights
[✔] Successfully updated weights for batch 1
[[[0.1099956199826718, 0.9167197367101835, -0.12671535669285527],
[-0.28448632826692166, 0.6925079343434363, 0.3919783939234854],
[0.4462223772408018, -0.8320145705501558, 0.2857919330935424],
[0.36039752088988214, 0.5755277819859513, -0.3359253028758331],
[0.3581115561294829, 0.47359102103089934, 0.6682974228396178],
[-0.6879971918671401, 0.47360898202991786, 0.6143882098372228],
```

```
[-0.5588927866971451, -0.35374590522821453, 0.61263869192536],
[0.5020608168552809, 0.2631180070696312, -0.565178823924912],
[0.8828940718426717, -0.022819279389278847, -0.06007479245339267]]]
[●] Calculating batch loss
[✔] Successfully calculated batch loss
Batch Loss: 0.21856847937318644

[✔] Successfully built batch 1

[●] Batch 2 ====================

[●] Building batch 2
[●] Evaluating data 1
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: softmax
h00: 0.07862092628573392
h01: 0.002750251942117924
h02: 0.9186288217721481

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 2.543117374249968

[✔] Successfully evaluated data 1

[●] Evaluating data 1
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: softmax
e00: 0.9213790737142661
e01: -0.002750251942117924
e02: -0.9186288217721481

[✔] Successfully evaluated data 1

[●] Updating weights
[✔] Successfully updated weights for batch 2
[[[0.11920941071981446, 0.9166922341907623, -0.13590164491057677],
[-0.300979013686407, 0.6925571638532002, 0.40842184983320684],
[0.46142513195708723, -0.8320599497072008, 0.27063481775011138],
[0.3533029020222823, 0.5755489589259056, -0.3288518609481876],
[0.348621351670226, 0.47361934862590316, 0.6777592997038709],
[-0.6870758127934259, 0.47360623177797573, 0.6134695810154506],
[-0.5393595503344027, -0.35380421056938743, 0.5931637609037905],
[0.4803162707156242, 0.26318291301546515, -0.5434991837310893],
[0.8944113102641, -0.02285365753855532, -0.07155765272554453]]]
[●] Calculating batch loss
[✔] Successfully calculated batch loss
Batch Loss: 2.543117374249968
```

```
[✔] Successfully built batch 2

[●] Batch 3 ====================

[●] Building batch 3
[●] Evaluating data 1
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: softmax
h00: 0.24342456534744028
h01: 0.17453170755771164
h02: 0.5820437270948481

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 0.5412096986076268

[✔] Successfully evaluated data 1

[●] Evaluating data 1
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: softmax
e00: -0.24342456534744028
e01: -0.17453170755771164
e02: 0.4179562729051519

[✔] Successfully evaluated data 1

[●] Updating weights
[✔] Successfully updated weights for batch 3
[[[0.11677516506634006, 0.9149469171151852, -0.13172208218152526],
[-0.3049955190146398, 0.689677390678498, 0.41531812833614185],
[0.4557289971279571, -0.8361439916640512, 0.2804149945360943],
[0.3526456556958442, 0.5750777233154998, -0.3277233790113437],
[0.3429252168410959, 0.4695353066690527, 0.6875394764898515],
[-0.6883416205332326, 0.4726986668986756, 0.6156429536345573],
[-0.5426944668796627, -0.35619529496292807, 0.5988897618425911],
[0.47600765590897454, 0.26009370179169367, -0.5361013577006681],
[0.8929020779589459, -0.023935754125413135, -0.068966323833532591]]]
[●] Calculating batch loss
[✔] Successfully calculated batch loss
Batch Loss: 0.5412096986076268

[✔] Successfully built batch 3

[●] Calculating epoch loss
[✔] Successfully calculated epoch loss
Epoch Loss: 1.100965184076927
```

```
[●] Epoch 9 ========================================

[●] Batch 1 ====================

[●] Building batch 1
[●] Evaluating data 1
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: softmax
h00: 0.16720155513676802
h01: 0.8300553683744649
h02: 0.0027430764887669203

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 0.18626286853153468

[✔] Successfully evaluated data 1

[●] Evaluating data 1
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: softmax
e00: -0.16720155513676802
e01: 0.16994463162553508
e02: -0.0027430764887669203

[✔] Successfully evaluated data 1

[●] Updating weights
[✔] Successfully updated weights for batch 1
[[[0.11510314951497237, 0.9166463634314406, -0.13174951294641293],
[-0.30098268169135733, 0.6855987195194851, 0.41538396217187223],
[0.46037720036075924, -0.840868452423241, 0.28049125206248204],
[0.3536488650266648, 0.5740580555257465, -0.3277069205524111],
[0.34230657108708984, 0.47016410180606716, 0.687529327106843],
[-0.692454778789597, 0.4768793048366638, 0.6155754739529337],
[-0.5411562125724044, -0.35775878557388296, 0.5989149981462878],
[0.4713928929871997, 0.2647841736245584, -0.5361770666117581],
[0.8885213972143626, -0.019483204776824116, -0.06903819243753828]]]
[●] Calculating batch loss
[✔] Successfully calculated batch loss
Batch Loss: 0.18626286853153468

[✔] Successfully built batch 1

[●] Batch 2 ====================
```

```
[●] Building batch 2
[●] Evaluating data 1
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: softmax
h00: 0.11151401543523436
h01: 0.003188580731722332
h02: 0.8852974038330433

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 2.1936049940107054

[✔] Successfully evaluated data 1

[●] Evaluating data 1
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: softmax
e00: 0.8884859845647657
e01: -0.003188580731722332
e02: -0.8852974038330433

[✔] Successfully evaluated data 1

[●] Updating weights
[✔] Successfully updated weights for batch 2
[[[0.12398800936062003, 0.9166144776241233, -0.14060248698474337],
[-0.31688658081506665, 0.685655795114583, 0.43123078570048373],
[0.4750372191060779, -0.8409210640053144, 0.26588384489923683],
[0.34680752294551614, 0.5740826075973807, -0.3208901305428966],
[0.33315516544607277, 0.4701969441876039, 0.6966478903663234],
[-0.6915662928050322, 0.47687611625593207, 0.6146901765491006],
[-0.5223203096996314, -0.35782638348539547, 0.5801466931850273],
[0.45042462375147124, 0.26485942412982705, -0.5152840478812983],
[0.8996274720214222, -0.019523062035970644, -0.08010440998545132]]]
[●] Calculating batch loss
[✔] Successfully calculated batch loss
Batch Loss: 2.1936049940107054

[✔] Successfully built batch 2

[●] Batch 3 ====================

[●] Building batch 3
[●] Evaluating data 1
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: softmax
```

```
h00: 0.22161705358936498
h01: 0.16159528148028532
h02: 0.6167876649303498

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 0.48323045242464063

[✔] Successfully evaluated data 1

[●] Evaluating data 1
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: softmax
e00: -0.22161705358936498
e01: -0.16159528148028532
e02: 0.38321233506965025

[✔] Successfully evaluated data 1

[●] Updating weights
[✔] Successfully updated weights for batch 3
[[[0.12177183882472638, 0.9149985248093204, -0.13677036363404688],
[-0.3205432621992912, 0.6829894729701582, 0.43755378922913296],
[0.46985138005208676, -0.8447023935919531, 0.27485101353986663],
[0.3462091569008249, 0.573646300337384, -0.3198554572382086],
[0.32796932639208165, 0.4664156146009652, 0.7056150590069532],
[-0.692718701483697, 0.4760358207922346, 0.6166828806914628],
[-0.5253564633338057, -0.36004023884167535, 0.5853967021754816],
[0.4465020019029395, 0.261999187647626, -0.5085011895505656],
[0.8982534462891681, -0.020524952781148412, -0.07772849350801948]]]
[●] Calculating batch loss
[✔] Successfully calculated batch loss
Batch Loss: 0.48323045242464063

[✔] Successfully built batch 3

[●] Calculating epoch loss
[✔] Successfully calculated epoch loss
Epoch Loss: 0.9543661049889604


[●] Epoch 10 ========================================

[●] Batch 1 ====================

[●] Building batch 1
[●] Evaluating data 1
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
```

```
Activation function: softmax
h00: 0.14631957017015298
h01: 0.8509739735367845
h02: 0.0027064562930624258

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 0.16137373126447657

[✔] Successfully evaluated data 1

[●] Evaluating data 1
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: softmax
e00: -0.14631957017015298
e01: 0.1490260264632155
e02: -0.0027064562930624258

[✔] Successfully evaluated data 1

[●] Updating weights
[✔] Successfully updated weights for batch 1
[[[0.12030864312302485, 0.9164887850739526, -0.1367974281969775],
[-0.3170315925152075, 0.679412848335041, 0.43761874418016644],
[0.47391906410281703, -0.8488453171276304, 0.27492625302481377],
[0.34708707432184577, 0.5727521441786046, -0.3198392185004502],
[0.32742794398245206, 0.4669670108988791, 0.7056050451186688],
[-0.6963181629098827, 0.4797018610432297, 0.6166163018666535],
[-0.5240103232882403, -0.3614112782851369, 0.5854216015733777],
[0.4424635817662433, 0.2661123059780107, -0.5085758877442541],
[0.8944198735507102, -0.016620470887812166, -0.07779940266289771]]]
[●] Calculating batch loss
[✔] Successfully calculated batch loss
Batch Loss: 0.16137373126447657

[✔] Successfully built batch 1

[●] Batch 2 ====================

[●] Building batch 2
[●] Evaluating data 1
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: softmax
h00: 0.15387763177403485
h01: 0.0036268128754966427
h02: 0.8424955553504685

[●] Calculating Loss
```

```
[✔] Successfully calculated Loss
Loss: 1.8715975882994262

[✔] Successfully evaluated data 1

[●] Evaluating data 1
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: softmax
e00: 0.8461223682259651
e01: -0.0036268128754966427
e02: -0.8424955553504685

[✔] Successfully evaluated data 1

[●] Updating weights
[✔] Successfully updated weights for batch 2
[[[0.1287698668052845, 0.9164525169451976, -0.1452223837504822],
[-0.3321771829064523, 0.6794777682855124, 0.4526994146209398],
[0.4878800831785455, -0.8489051595400762, 0.261025076361531O5],
[0.3405719320865058, 0.5727800706377459, -0.3133520027242516],
[0.3187128835897246, 0.46700436707149673, 0.7142827493387787],
[-0.6954720405416567, 0.4796982342303542, 0.6157738063113031],
[-0.5060725290818499, -0.36148816671809747, 0.5675606957999478],
[0.4224950938761105, 0.26619789876187244, -0.48869299263798305],
[0.9049964031535347, -0.016665806048755873, -0.08833059710477857]]]
[●] Calculating batch loss
[✔] Successfully calculated batch loss
Batch Loss: 1.8715975882994262

[✔] Successfully built batch 2

[●] Batch 3 ====================

[●] Building batch 3
[●] Evaluating data 1
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: softmax
h00: 0.20238121656597066
h01: 0.14987172540112167
h02: 0.6477470580329077

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 0.4342549983751524

[✔] Successfully evaluated data 1

[●] Evaluating data 1
```

```
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: softmax
e00: -0.20238121656597066
e01: -0.14987172540112167
e02: 0.35225294196709234

[✔] Successfully evaluated data 1

[●] Updating weights
[✔] Successfully updated weights for batch 3
[[[0.1267460546396248, 0.9149537996911864, -0.14169985433081125],
[-0.3355164729797908, 0.6770048848163939, 0.4585115881633968],
[0.4831443627109018, -0.8524121579144625, 0.26926779520356103],
[0.3400255028017777, 0.5723754169791629, -0.3124009197809405],
[0.3139771631220809, 0.4634973686971105, 0.7225254681808086],
[-0.6965244228677997, 0.47891890125826836, 0.617605521609532],
[-0.5088451517488036, -0.36354140935609286, 0.572386561104897],
[0.41891294634289283, 0.2635451692222726, -0.4824581155651655],
[0.9037416396108257, -0.017595010746242826, -0.0861466288645826]]]
[●] Calculating batch loss
[✔] Successfully calculated batch loss
Batch Loss: 0.4342549983751524

[✔] Successfully built batch 3

[●] Calculating epoch loss
[✔] Successfully calculated epoch loss
Epoch Loss: 0.8224087726463517


[●] Asserting output ========================================

Stopped by max_iteration
Test result: SSE (1.9504576876404183e-16) < MAX SSE (1e-07)
Test status: PASS

Final Weights :
[[[0.1267460546396248, 0.9149537996911864, -0.14169985433081125],
[-0.3355164729797908, 0.6770048848163939, 0.4585115881633968],
[0.4831443627109018, -0.8524121579144625, 0.26926779520356103],
[0.3400255028017777, 0.5723754169791629, -0.3124009197809405],
[0.3139771631220809, 0.4634973686971105, 0.7225254681808086],
[-0.6965244228677997, 0.47891890125826836, 0.617605521609532],
[-0.5088451517488036, -0.36354140935609286, 0.572386561104897],
[0.41891294634289283, 0.2635451692222726, -0.4824581155651655],
[0.9037416396108257, -0.017595010746242826, -0.0861466288645826]]]
```

7. softmax_two_layer.json

```
[✔] Successfully loaded model from 'models/softmax_two_layer.json'

[✔] Successfully initialized model
Number of features: 2
Number of layers: 2
Number of data: 8


[●] Epoch 1 ========================================

[●] Batch 1 ====================

[●] Building batch 1
[●] Evaluating data 1
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 4
Activation function: relu
h00: 0.0
h01: 0.595
h02: 0.0
h03: 0.0030000000000000058

[●] Building layer 1
[✔] Successfully built layer 1
Number of neurons: 2
Activation function: softmax
h10: 0.5868503570425038
h11: 0.4131496429574962

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 0.883945418031139

[✔] Successfully evaluated data 1

[●] Evaluating data 1
[●] Calculating Error Node Output for layer 1
[✔] Successfully calculated Error Node Output for layer 1
Derivative function: softmax
e10: -0.5868503570425038
e11: 0.5868503570425039

[✔] Successfully evaluated data 1

[●] Evaluating data 1
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: relu
```

```
e00: 0.0
e01: -0.12910707854935083
e02: 0.0
e03: -0.011737007140850075

[✔] Successfully evaluated data 1

[●] Updating weights
[✔] Successfully updated weights for batch 1
[[[0.1, -0.1129107078549351, 0.1, -0.10117370071408502], [-0.1,
0.048486275658809024, -0.1, 0.09531693415080082], [0.1,
0.06178430474939216, -0.1, -0.10347415411369162]],
[[0.06131496429574962, -0.041314964295749615], [-0.12, 0.1],
[0.08508240375597102, -0.06508240375597102], [-0.12, 0.1],
[0.019823944892887246, 0.00017605510711127546]]]
[●] Calculating batch loss
[✔] Successfully calculated batch loss
Batch Loss: 0.883945418031139

[✔] Successfully built batch 1

[●] Batch 2 ====================

[●] Building batch 2
[●] Evaluating data 1
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 4
Activation function: relu
h00: 0.45099999999999996
h01: 0.025660089725608543
h02: 0.0
h03: 0.0

[●] Building layer 1
[✔] Successfully built layer 1
Number of neurons: 2
Activation function: softmax
h10: 0.5018157847740381
h11: 0.49818421522596196

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 0.6895221872391776

[✔] Successfully evaluated data 1

[●] Evaluating data 1
[●] Calculating Error Node Output for layer 1
[✔] Successfully calculated Error Node Output for layer 1
Derivative function: softmax
```

```
e10: 0.4981842152259619
e11: -0.49818421522596196

[✔] Successfully evaluated data 1

[●] Evaluating data 1
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: relu
e00: -0.10960052734971162
e01: 0.07480973678489447
e02: 0.0
e03: 0.0

[✔] Successfully evaluated data 1

[●] Updating weights
[✔] Successfully updated weights for batch 2
[[[0.08903994726502884, -0.10542973417644565, 0.1,
-0.10117370071408502], [-0.09221836255817048, 0.04317478434708152,
-0.1, 0.09531693415080082], [0.06931185234208075,
0.0827310310491626, -0.1, -0.10347415411369162]],
[[0.11113338581834581, -0.09113338581834582],
[-0.09753189189330912, 0.07753189189330913], [0.08636074892222903,
-0.06636074892222903], [-0.12, 0.1], [0.019823944892887246,
0.00017605510711127546]]]
[●] Calculating batch loss
[✔] Successfully calculated batch loss
Batch Loss: 0.6895221872391776

[✔] Successfully built batch 2

[●] Batch 3 ====================

[●] Building batch 3
[●] Evaluating data 1
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 4
Activation function: relu
h00: 0.29926819781296693
h01: 0.0
h02: 0.36300000000000004
h03: 0.0

[●] Building layer 1
[✔] Successfully built layer 1
Number of neurons: 2
Activation function: softmax
h10: 0.5174967899863836
h11: 0.4825032100136164
```

```
[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 0.7287677033472654

[✔] Successfully evaluated data 1

[●] Evaluating data 1
[●] Calculating Error Node Output for layer 1
[✔] Successfully calculated Error Node Output for layer 1
Derivative function: softmax
e10: -0.5174967899863836
e11: 0.5174967899863836

[✔] Successfully evaluated data 1

[●] Evaluating data 1
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: relu
e00: 0.09059494615244525
e01: 0.0
e02: 0.1138492937970044
e03: 0.0

[✔] Successfully evaluated data 1

[●] Updating weights
[✔] Successfully updated weights for batch 3
[[[0.09809944188027336, -0.10542973417644565, 0.11138492937970045,
-0.101173700714085021, [-0.11423293447321467, 0.04317478434708152,
-0.12766537839267209, 0.09531693415080082], [0.06749995341903184,
0.0827310310491626, -0.1022769858759401, -0.10347415411369162]],
[[0.05938370681970745, -0.03938370681970746],
[-0.11301892506463117, 0.09301892506463118], [0.08636074892222903,
-0.06636074892222903], [-0.13878513347650573, 0.11878513347650574],
[0.019823944892887246, 0.00017605510711275461]]]
[●] Calculating batch loss
[✔] Successfully calculated batch loss
Batch Loss: 0.7287677033472654

[✔] Successfully built batch 3

…

[●] Batch 5 ====================

[●] Building batch 5
[●] Evaluating data 1
[●] Building layer 0
[✔] Successfully built layer 0
```

```
Number of neurons: 4
Activation function: relu
h00: 0.6189771368528977
h01: 4.917374081169825
h02: 1.3691493502711627
h03: 0.0

[●] Building layer 1
[✔] Successfully built layer 1
Number of neurons: 2
Activation function: softmax
h10: 0.993298187753054
h11: 0.006701812246946138

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 0.00672436823339215

[✔] Successfully evaluated data 1

[●] Evaluating data 1
[●] Calculating Error Node Output for layer 1
[✔] Successfully calculated Error Node Output for layer 1
Derivative function: softmax
e10: 0.006701812246945993
e11: -0.006701812246946138

[✔] Successfully evaluated data 1

[●] Evaluating data 1
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: relu
e00: -0.006616385594294059
e01: 0.01668687750462991
e02: -0.015550561141353702
e03: 0.0

[✔] Successfully evaluated data 1

[●] Updating weights
[✔] Successfully updated weights for batch 5
[[[-0.28747623359290725, -0.2877821415172352, -0.7055609150557745,
0.4201791094471029], [-0.578482154041294, -1.18515594551451,
-1.3405295364202199, 0.6948726658361675], [-0.4143002385575707,
1.5130365900232106, -0.9782811220400641, -1.3025856174844246]],
[[-1.7179035638544469, 1.7379035638544473], [-0.5032117737186412,
0.4832117737186408], [1.2582481572648112, -1.23824815726481],
[-1.169258249509587, 1.1492582495095867], [1.0892683204083213,
-1.0692683204083209]]]
[●] Calculating batch loss
```

```
[✔] Successfully calculated batch loss
Batch Loss: 0.00672436823339215

[✔] Successfully built batch 5

[●] Batch 6 ====================

[●] Building batch 6
[●] Evaluating data 1
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 4
Activation function: relu
h00: 1.800292614408124
h01: 3.3990436040917316
h02: 4.137025675375596
h03: 0.0

[●] Building layer 1
[✔] Successfully built layer 1
Number of neurons: 2
Activation function: softmax
h10: 0.0017652068919563801
h11: 0.9982347931080436

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 0.0017667647055080928

[✔] Successfully evaluated data 1

[●] Evaluating data 1
[●] Calculating Error Node Output for layer 1
[✔] Successfully calculated Error Node Output for layer 1
Derivative function: softmax
e10: -0.0017652068919563801
e11: 0.0017652068919563613

[✔] Successfully evaluated data 1

[●] Evaluating data 1
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: relu
e00: 0.0017412416443243423
e01: -0.004406832500151368
e02: 0.004092661303183202
e03: 0.0

[✔] Successfully evaluated data 1
```

```
[●] Updating weights
[✔] Successfully updated weights for batch 6
[[[-0.28730210942847484, -0.2882228247672503, -0.7051516489254561,
0.42017910944710291], [-0.5790793999252972, -1.183644401966958,
-1.3419333192472118, 0.6948726658361675], [-0.4143437695986788,
1.5131467608357143, -0.9783834385726436, -1.3025856174844246]],
[[-1.7180800845436426, 1.738080084543643], [-0.5035295626116904,
0.48352956261168994], [1.2576481557452108, -1.2376481557452097],
[-1.1699885201330242, 1.149988520133024], [1.0892683204083213,
-1.0692683204083209]]]
[●] Calculating batch loss
[✔] Successfully calculated batch loss
Batch Loss: 0.0017667647055080928

[✔] Successfully built batch 6

[●] Batch 7 ====================

[●] Building batch 7
[●] Evaluating data 1
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 4
Activation function: relu
h00: 0.0
h01: 0.0
h02: 0.0019069426573303794
h03: 4.215229595372872

[●] Building layer 1
[✔] Successfully built layer 1
Number of neurons: 2
Activation function: softmax
h10: 0.9964531537478781
h11: 0.0035468462521219286

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 0.0035531492242159325

[✔] Successfully evaluated data 1

[●] Evaluating data 1
[●] Calculating Error Node Output for layer 1
[✔] Successfully calculated Error Node Output for layer 1
Derivative function: softmax
e10: 0.0035468462521218713
e11: -0.0035468462521219286

[✔] Successfully evaluated data 1
```

```
[●] Evaluating data 1
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: relu
e00: 0.0
e01: 0.0
e02: -0.008228601870276492
e03: 0.007655997594548302

[✔] Successfully evaluated data 1

[●] Updating weights
[✔] Successfully updated weights for batch 7
[[[-0.28730210942847484, -0.2882228247672503, -0.7059745091124837,
0.42094470920655774], [-0.5790793999252972, -1.183644401966958,
-1.3428796084622936, 0.6957531055595405], [-0.4143437695986788,
1.5131467608357143, -0.9764908601424801, -1.3043464969311707]],
[[-1.7177253999184303, 1.7377253999184308], [-0.5035295626116904,
0.48352956261168994], [1.2576481557452108, -1.2376481557452097],
[-1.1699878437697826, 1.1499878437697824], [1.0907633975375395,
-1.070763397537539]]]
[●] Calculating batch loss
[✔] Successfully calculated batch loss
Batch Loss: 0.0035531492242159325

[✔] Successfully built batch 7

[●] Batch 8 ====================

[●] Building batch 8
[●] Evaluating data 1
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 4
Activation function: relu
h00: 0.0
h01: 0.0
h02: 0.0
h03: 0.0

[●] Building layer 1
[✔] Successfully built layer 1
Number of neurons: 2
Activation function: softmax
h10: 0.030606719962503852
h11: 0.9693932800374961

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 0.03108488567917136
```

```
[✔] Successfully evaluated data 1

[●] Evaluating data 1
[●] Calculating Error Node Output for layer 1
[✔] Successfully calculated Error Node Output for layer 1
Derivative function: softmax
e10: -0.030606719962503852
e11: 0.03060671996250386

[✔] Successfully evaluated data 1

[●] Evaluating data 1
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: relu
e00: 0.0
e01: 0.0
e02: 0.0
e03: 0.0

[✔] Successfully evaluated data 1

[●] Updating weights
[✔] Successfully updated weights for batch 8
[[[-0.28730210942847484, -0.2882228247672503, -0.7059745091124837,
0.42094470920655774], [-0.5790793999252972, -1.183644401966958,
-1.3428796084622936, 0.6957531055595405], [-0.4143437695986788,
1.5131467608357143, -0.9764908601424801, -1.3043464969311707]],
[[-1.7207860719146808, 1.7407860719146813], [-0.5035295626116904,
0.48352956261168994], [1.2576481557452108, -1.2376481557452097],
[-1.1699878437697826, 1.1499878437697824], [1.0907633975375395,
-1.070763397537539]]]
[●] Calculating batch loss
[✔] Successfully calculated batch loss
Batch Loss: 0.03108488567917136

[✔] Successfully built batch 8

[●] Calculating epoch loss
[✔] Successfully calculated epoch loss
Epoch Loss: 0.009936240476110797

[✔] Loss threshold reached


[●] Asserting output ========================================

Stopped by error_threshold
Test result: SSE (1.584550445837967e-16) < MAX SSE (1e-07)
Test status: PASS
```

```
Final Weights :
[[[-0.28730210942847484, -0.2882228247672503, -0.7059745091124837,
0.42094470920655774], [-0.5790793999252972, -1.183644401966958,
-1.3428796084622936, 0.6957531055595405], [-0.4143437695986788,
1.5131467608357143, -0.9764908601424801, -1.3043464969311707]],
[[-1.7207860719146808, 1.7407860719146813], [-0.5035295626116904,
0.48352956261168994], [1.2576481557452108, -1.2376481557452097],
[-1.1699878437697826, 1.1499878437697824], [1.0907633975375395,
-1.0707633975375397]]]
```

8. sigmoid.json

```
[✔] Successfully loaded model from 'models/sigmoid.json'

[✔] Successfully initialized model
Number of features: 2
Number of layers: 1
Number of data: 2


[●] Epoch 1 ========================================

[●] Batch 1 ====================

[●] Building batch 1
[●] Evaluating data 1
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 2
Activation function: sigmoid
h00: 0.598687660112452
h01: 0.598687660112452

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 0.0194785085169417

[✔] Successfully evaluated data 1

[●] Evaluating data 2
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 2
Activation function: sigmoid
h00: 0.6681877721681662
h01: 0.5621765008857981

[●] Calculating Loss
```

```
[✔] Successfully calculated Loss
Loss: 0.026533849149840694

[✔] Successfully evaluated data 2

[●] Evaluating data 1
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: sigmoid
e00: -0.14384114368486883
e01: 0.0964196020566603

[✔] Successfully evaluated data 1

[●] Evaluating data 2
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: sigmoid
e00: 0.07356704242638361
e01: -0.13837079738215854

[✔] Successfully evaluated data 2

[●] Updating weights
[✔] Successfully updated weights for batch 1
[[[0.2929725898741514, 0.09580488046745018], [0.19280794281575658,
0.604820980102833], [0.8036783521213192, 0.2930814601308920411]]]
[●] Calculating batch loss
[✔] Successfully calculated batch loss
Batch Loss: 0.0230061788333912

[✔] Successfully built batch 1

[●] Calculating epoch loss
[✔] Successfully calculated epoch loss
Epoch Loss: 0.0230061788333912


[●] Epoch 2 ========================================

[●] Batch 1 ====================

[●] Building batch 1
[●] Evaluating data 1
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 2
Activation function: sigmoid
h00: 0.5961326101130702
h01: 0.5982588082864907
```

```
[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 0.018894011773696564

[✔] Successfully evaluated data 1

[●] Evaluating data 2
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 2
Activation function: sigmoid
h00: 0.6670364718469761
h01: 0.5602916068936388

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 0.025839027691949602

[✔] Successfully evaluated data 2

[●] Evaluating data 1
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: sigmoid
e00: -0.14352400569334628
e01: 0.09655656971975084

[✔] Successfully evaluated data 1

[●] Evaluating data 2
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: sigmoid
e00: 0.07395080573121061
e01: -0.13803619810702872

[✔] Successfully evaluated data 2

[●] Updating weights
[✔] Successfully updated weights for batch 1
[[[0.28601526987793785, 0.0916569176287224], [0.18563174253108927,
0.6096488085888205], [0.8073758924078798, 0.2861796502255406]]]
[●] Calculating batch loss
[✔] Successfully calculated batch loss
Batch Loss: 0.02236651973282308

[✔] Successfully built batch 1

[●] Calculating epoch loss
[✔] Successfully calculated epoch loss
Epoch Loss: 0.02236651973282308
```

```
[●] Epoch 3 ========================================

[●] Batch 1 ====================

[●] Building batch 1
[●] Evaluating data 1
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 2
Activation function: sigmoid
h00: 0.5935911574864398
h01: 0.5978419670992413

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 0.018323320594318446

[✔] Successfully evaluated data 1

[●] Evaluating data 2
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 2
Activation function: sigmoid
h00: 0.6659009024398505
h01: 0.5584186633490336

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 0.025159633797856733

[✔] Successfully evaluated data 2

[●] Evaluating data 1
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: sigmoid
e00: -0.14319834352055189
e01: 0.09668962905685466

[✔] Successfully evaluated data 1

[●] Evaluating data 2
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: sigmoid
e00: 0.07432932836730595
e01: -0.13769892800106723
```

```
[✔] Successfully evaluated data 2

[●] Updating weights
[✔] Successfully updated weights for batch 1
[[[0.27912836836261323, 0.087555987734301 14], [0.17847182535506168,
0.6144832900416632], [0.8110923588262451, 0.2792947038254872]]]
[●] Calculating batch loss
[✔] Successfully calculated batch loss
Batch Loss: 0.02174147719608759

[✔] Successfully built batch 1

[●] Calculating epoch loss
[✔] Successfully calculated epoch loss
Epoch Loss: 0.02174147719608759


[●] Epoch 4 ========================================

[●] Batch 1 ====================

[●] Building batch 1
[●] Evaluating data 1
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 2
Activation function: sigmoid
h00: 0.5910636720142057
h01: 0.5974370962477745

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 0.01776626981767839

[✔] Successfully evaluated data 1

[●] Evaluating data 2
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 2
Activation function: sigmoid
h00: 0.6647812081941112
h01: 0.5565577450137487

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 0.024495466103575596

[✔] Successfully evaluated data 2

[●] Evaluating data 1
```

```
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: sigmoid
e00: -0.14286446791231275
e01: 0.09681879867121264

[✔] Successfully evaluated data 1

[●] Evaluating data 2
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: sigmoid
e00: 0.07470255352887506
e01: -0.13735913129296862

[✔] Successfully evaluated data 2

[●] Updating weights
[✔] Successfully updated weights for batch 1
[[[0.27231217692426946, 0.08350195447212554], [0.17132860195944605,
0.6193242299752238], [0.8148274865026889, 0.27242674726083876]]]
[●] Calculating batch loss
[✔] Successfully calculated batch loss
Batch Loss: 0.021130867960626992

[✔] Successfully built batch 1

[●] Calculating epoch loss
[✔] Successfully calculated epoch loss
Epoch Loss: 0.021130867960626992


[●] Epoch 5 ======================================

[●] Batch 1 ====================

[●] Building batch 1
[●] Evaluating data 1
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 2
Activation function: sigmoid
h00: 0.5885505083149751
h01: 0.5970441519963781

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 0.017222688968043288

[✔] Successfully evaluated data 1
```

```
[●] Evaluating data 2
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 2
Activation function: sigmoid
h00: 0.6636775228823234
h01: 0.5547089202701808

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 0.02384631927640098

[✔] Successfully evaluated data 2

[●] Evaluating data 1
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: sigmoid
e00: -0.14252268923363043
e01: 0.09694409812832029

[✔] Successfully evaluated data 1

[●] Evaluating data 2
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: sigmoid
e00: 0.07507042862758022
e01: -0.13701694968216874

[✔] Successfully evaluated data 2

[●] Updating weights
[✔] Successfully updated weights for batch 1
[[[0.2655669508636644, 0.07949466931674068], [0.16420246749776451,
0.6241714348816398], [0.8185810079340679, 0.265575899776735]]]
[●] Calculating batch loss
[✔] Successfully calculated batch loss
Batch Loss: 0.020534504122222134

[✔] Successfully built batch 1

[●] Calculating epoch loss
[✔] Successfully calculated epoch loss
Epoch Loss: 0.020534504122222134


[●] Epoch 6 ========================================

[●] Batch 1 ====================
```

```
[●] Building batch 1
[●] Evaluating data 1
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 2
Activation function: sigmoid
h00: 0.586052005784154
h01: 0.5966630872743749

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 0.01669240261569343

[✔] Successfully evaluated data 1

[●] Evaluating data 2
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 2
Activation function: sigmoid
h00: 0.6625899698689619
h01: 0.5528722512345812

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 0.023211984361436042

[✔] Successfully evaluated data 2

[●] Evaluating data 1
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: sigmoid
e00: -0.1421733169940338
e01: 0.09706554791841407

[✔] Successfully evaluated data 1

[●] Evaluating data 2
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: sigmoid
e00: 0.0754329052541561
e01: -0.1366725222794302

[✔] Successfully evaluated data 2

[●] Updating weights
[✔] Successfully updated weights for batch 1
[[[0.25889290968967665, 0.07553397188063907], [0.15709380164806283,
0.6290247122775605], [0.8223526531967756, 0.2587422736627588]]]
```

```
[●] Calculating batch loss
[✔] Successfully calculated batch loss
Batch Loss: 0.019952193488564737

[✔] Successfully built batch 1

[●] Calculating epoch loss
[✔] Successfully calculated epoch loss
Epoch Loss: 0.019952193488564737


[●] Epoch 7 ========================================

[●] Batch 1 ====================

[●] Building batch 1
[●] Evaluating data 1
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 2
Activation function: sigmoid
h00: 0.5835684885641302
h01: 0.5962938517745601

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 0.016175230736055 43

[✔] Successfully evaluated data 1

[●] Evaluating data 2
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 2
Activation function: sigmoid
h00: 0.6615186621927844
h01: 0.5510477938741284

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 0.02259224912242339

[✔] Successfully evaluated data 2

[●] Evaluating data 1
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: sigmoid
e00: -0.14181665939207475
e01: 0.09718316941927956
```

```
[✔] Successfully evaluated data 1

[●] Evaluating data 2
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: sigmoid
e00: 0.07578993913321067
e01: -0.13632598555362443

[✔] Successfully evaluated data 2

[●] Updating weights
[✔] Successfully updated weights for batch 1
[[[0.2522902376637903, 0.07161969026720458], [0.15000296867845908,
0.6338838707485245], [0.8261421501534362, 0.2519259743850776]]]
[●] Calculating batch loss
[✔] Successfully calculated batch loss
Batch Loss: 0.01938373992923941

[✔] Successfully built batch 1

[●] Calculating epoch loss
[✔] Successfully calculated epoch loss
Epoch Loss: 0.01938373992923941


[●] Epoch 8 ========================================

[●] Batch 1 ====================

[●] Building batch 1
[●] Evaluating data 1
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 2
Activation function: sigmoid
h00: 0.5811002655438001
h01: 0.5959363920520812

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 0.015670989066360653

[✔] Successfully evaluated data 1

[●] Evaluating data 2
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 2
Activation function: sigmoid
h00: 0.6604637026641766
```

```
h01: 0.5492355981275022

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 0.021986898376259504

[✔] Successfully evaluated data 2

[●] Evaluating data 1
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: sigmoid
e00: -0.14145302287968342
e01: 0.09729698485943264

[✔] Successfully evaluated data 1

[●] Evaluating data 2
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: sigmoid
e00: 0.07614149007159855
e01: -0.1359774732845473

[✔] Successfully evaluated data 2

[●] Updating weights
[✔] Successfully updated weights for batch 1
[[[0.24575908438298177, 0.06775164142469312], [0.1429303175344749,
0.6387487199914961], [0.8299492246570161, 0.2451271007208502]]]
[●] Calculating batch loss
[✔] Successfully calculated batch loss
Batch Loss: 0.01882894372131008

[✔] Successfully built batch 1

[●] Calculating epoch loss
[✔] Successfully calculated epoch loss
Epoch Loss: 0.01882894372131008


[●] Epoch 9 ======================================

[●] Batch 1 ====================

[●] Building batch 1
[●] Evaluating data 1
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 2
Activation function: sigmoid
```

```
h00: 0.5786476303863605
h01: 0.5955906516236025

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 0.015179489458891688

[✔] Successfully evaluated data 1

[●] Evaluating data 2
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 2
Activation function: sigmoid
h00: 0.6594251839767168
h01: 0.5474357080286119

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 0.02139571432062012

[✔] Successfully evaluated data 2

[●] Evaluating data 1
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: sigmoid
e00: -0.1410827117469736
e01: 0.09740701728171822

[✔] Successfully evaluated data 1

[●] Evaluating data 2
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: sigmoid
e00: 0.0764875219007618
e01: -0.13562711652159448

[✔] Successfully evaluated data 2

[●] Updating weights
[✔] Successfully updated weights for batch 1
[[[0.2392995653983606, 0.06392963150070549], [0.13587618194712622,
0.643619070855582], [0.8337736007520542, 0.2383457448947048]]]
[●] Calculating batch loss
[✔] Successfully calculated batch loss
Batch Loss: 0.018287601889755905

[✔] Successfully built batch 1
```

```
[●] Calculating epoch loss
[✔] Successfully calculated epoch loss
Epoch Loss: 0.018287601889755905


[●] Epoch 10 ========================================

[●] Batch 1 ====================

[●] Building batch 1
[●] Evaluating data 1
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 2
Activation function: sigmoid
h00: 0.5762108615842368
h01: 0.5952565710665944

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 0.014700540229933666

[✔] Successfully evaluated data 1

[●] Evaluating data 2
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 2
Activation function: sigmoid
h00: 0.6584031888321861
h01: 0.5456481618331449

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 0.020818476854172933

[✔] Successfully evaluated data 2

[●] Evaluating data 1
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: sigmoid
e00: -0.1407060277279717
e01: 0.09751329050736961

[✔] Successfully evaluated data 1

[●] Evaluating data 2
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: sigmoid
```

```
e00: 0.07682800241343894
e01: -0.1352750435481178

[✔] Successfully evaluated data 2

[●] Updating weights
[✔] Successfully updated weights for batch 1
[[[0.23291176286690732, 0.06015345619663067], [0.12884088056072765,
0.6484947353809505], [0.8376150008727261, 0.2315819927173646]]]
[●] Calculating batch loss
[✔] Successfully calculated batch loss
Batch Loss: 0.0177595085420533

[✔] Successfully built batch 1

[●] Calculating epoch loss
[✔] Successfully calculated epoch loss
Epoch Loss: 0.0177595085420533


[●] Asserting output ========================================

Stopped by max_iteration
Test result: SSE (2.058977452871475e-08) < MAX SSE (1e-07)
Test status: PASS

Final Weights :
[[[0.23291176286690732, 0.06015345619663067], [0.12884088056072765,
0.6484947353809505], [0.8376150008727261, 0.2315819927173646]]]
```

9. iris.csv

```
[✔] Successfully loaded data from 'models/iris.csv'

[●] Initializing random weights
[✔] Successfully initialized new model
[●] Epoch 1 ========================================

[●] Batch 1 ====================

[●] Building batch 1
[●] Evaluating data 1
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: softmax
h00: 0.5948988539823914
```

```
h01: 0.014269865598738638
h02: 0.39083128041887005

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 4.249605262986335

[✔] Successfully evaluated data 1

[●] Evaluating data 2
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: softmax
h00: 0.2859720215709368
h01: 0.38303286747012233
h02: 0.33099511095894096

…

[✔] Successfully built batch 21

[●] Batch 22 ====================

[●] Building batch 22
[●] Evaluating data 1
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: softmax
h00: 0.9161715991161592
h01: 0.08361372218261826
h02: 0.0002146787012223591

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 0.08755159357341934

[✔] Successfully evaluated data 1

[●] Evaluating data 2
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: softmax
h00: 0.02200685713091903
h01: 0.5661161620827295
h02: 0.41187698078635143

[●] Calculating Loss
[✔] Successfully calculated Loss
```

```
Loss: 0.5689559854747266

[✔] Successfully evaluated data 2

[●] Evaluating data 3
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: softmax
h00: 0.9583604152963772
h01: 0.04160709989354349
h02: 3.248481007933329e-05

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 0.04253135237838846

[✔] Successfully evaluated data 3

[●] Evaluating data 4
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: softmax
h00: 0.05145089480224031
h01: 0.6270493146751434
h02: 0.32149979052261646

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 0.4667300866476195

[✔] Successfully evaluated data 4

[●] Evaluating data 5
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: softmax
h00: 0.9007313592281357
h01: 0.09899728563104382
h02: 0.00027135514082038425

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 0.10454822129120862

[✔] Successfully evaluated data 5

[●] Evaluating data 1
[●] Calculating Error Node Output for layer 0
```

```
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: softmax
e00: 0.0838284008838408
e01: -0.08361372218261826
e02: -0.0002146787012223591

[✔] Successfully evaluated data 1

[●] Evaluating data 2
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: softmax
e00: -0.02200685713091903
e01: 0.4338838379172705
e02: -0.41187698078635143

[✔] Successfully evaluated data 2

[●] Evaluating data 3
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: softmax
e00: 0.041639584703622834
e01: -0.04160709989354349
e02: -3.248481007933329e-05

[✔] Successfully evaluated data 3

[●] Evaluating data 4
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: softmax
e00: -0.05145089480224031
e01: 0.37295068532485665
e02: -0.32149979052261646

[✔] Successfully evaluated data 4

[●] Evaluating data 5
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: softmax
e00: 0.09926864077186426
e01: -0.09899728563104382
e02: -0.00027135514082038425

[✔] Successfully evaluated data 5

[●] Updating weights
[✔] Successfully updated weights for batch 22
[array([[ 0.62944682,  0.90386608,  0.52393546],
```

```
       [ 1.03434509,  0.32306767, -0.44674112],
       [ 1.31594214,  0.78109442, -0.57166179],
       [-1.1356926 ,  0.67294505,  2.16131448],
       [-0.06079051,  0.13622747,  1.15116976]])]
[●] Calculating batch loss
[✔] Successfully calculated batch loss
Batch Loss: 0.25406344787307245

[✔] Successfully built batch 22

[●] Batch 23 ====================

[●] Building batch 23
[●] Evaluating data 1
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: softmax
h00: 0.8654277721618292
h01: 0.13411632246581956
h02: 0.0004559053723513823

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 0.14453135700090808

[✔] Successfully evaluated data 1

[●] Evaluating data 2
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: softmax
h00: 0.016317507365582386
h01: 0.4820313352336739
h02: 0.5016511574007436

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 0.6898503034254092

[✔] Successfully evaluated data 2

[●] Evaluating data 3
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: softmax
h00: 0.08302839321652722
h01: 0.7563155050540498
h02: 0.16065610172942296
```

```
[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 0.2792966521349411

[✔] Successfully evaluated data 3

[●] Evaluating data 4
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: softmax
h00: 0.0015528141632570604
h01: 0.18614937495704073
h02: 0.8122978108797022

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 0.20788824089414135

[✔] Successfully evaluated data 4

[●] Evaluating data 5
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: softmax
h00: 0.012305776354547205
h01: 0.43541879440993153
h02: 0.5522754292355212

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 0.5937083881408073

[✔] Successfully evaluated data 5

[●] Evaluating data 1
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: softmax
e00: 0.1345722278381708
e01: -0.13411632246581956
e02: -0.0004559053723513823

[✔] Successfully evaluated data 1

[●] Evaluating data 2
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: softmax
```

```
e00: -0.016317507365582386
e01: -0.4820313352336739
e02: 0.4983488425992564

[✔] Successfully evaluated data 2

[●] Evaluating data 3
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: softmax
e00: -0.08302839321652722
e01: 0.24368449494595024
e02: -0.16065610172942296

[✔] Successfully evaluated data 3

[●] Evaluating data 4
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: softmax
e00: -0.0015528141632570604
e01: -0.18614937495704073
e02: 0.1877021891202978

[✔] Successfully evaluated data 4

[●] Evaluating data 5
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: softmax
e00: -0.012305776354547205
e01: -0.43541879440993153
e02: 0.4477245707644788

[✔] Successfully evaluated data 5

[●] Updating weights
[✔] Successfully updated weights for batch 23
[array([[ 0.62946819,  0.90287205,  0.52490813],
       [ 1.03426169,  0.31671372, -0.44030377],
       [ 1.31603997,  0.77815125, -0.56881644],
       [-1.13609179,  0.66869345,  2.16596527],
       [-0.0610061 ,  0.13460702,  1.1530058 ]])]
[●] Calculating batch loss
[✔] Successfully calculated batch loss
Batch Loss: 0.3830549883192414

[✔] Successfully built batch 23

[●] Batch 24 ====================
```

```
[●] Building batch 24
[●] Evaluating data 1
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: softmax
h00: 0.01867366915788892
h01: 0.5236450953033196
h02: 0.4576812355387913

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 0.6469411201711751

[✔] Successfully evaluated data 1

[●] Evaluating data 2
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: softmax
h00: 0.16820039241679066
h01: 0.7147536445553183
h02: 0.11704596302789115

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 0.3358173457296613

[✔] Successfully evaluated data 2

[●] Evaluating data 3
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: softmax
h00: 0.8554620442902288
h01: 0.1431424664644254
h02: 0.0013954892453458252

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 0.15611355033169005

[✔] Successfully evaluated data 3

[●] Evaluating data 4
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: softmax
```

```
h00: 0.0005399880465785823
h01: 0.13598369309615063
h02: 0.8634763188572707

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 0.14678880337219374

[✔] Successfully evaluated data 4

[●] Evaluating data 5
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: softmax
h00: 0.0046092752721218965
h01: 0.3370175079219408
h02: 0.6583732168059372

[●] Calculating Loss
[✔] Successfully calculated Loss
Loss: 0.41798330666003825

[✔] Successfully evaluated data 5

[●] Evaluating data 1
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: softmax
e00: -0.01867366915788892
e01: 0.4763549046966804
e02: -0.4576812355387913

[✔] Successfully evaluated data 1

[●] Evaluating data 2
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: softmax
e00: -0.16820039241679066
e01: 0.2852463554446817
e02: -0.11704596302789115

[✔] Successfully evaluated data 2

[●] Evaluating data 3
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: softmax
e00: 0.14453795570977124
e01: -0.1431424664644254
```

```
e02: -0.0013954892453458252

[✔] Successfully evaluated data 3

[●] Evaluating data 4
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: softmax
e00: -0.0005399880465785823
e01: -0.13598369309615063
e02: 0.13652368114272928

[✔] Successfully evaluated data 4

[●] Evaluating data 5
[●] Calculating Error Node Output for layer 0
[✔] Successfully calculated Error Node Output for layer 0
Derivative function: softmax
e00: -0.0046092752721218965
e01: -0.3370175079219408
e02: 0.3416267831940628

[✔] Successfully evaluated data 5

[●] Updating weights
[✔] Successfully updated weights for batch 24
[array([[ 0.6294207 ,  0.90301751,  0.52481016],
       [ 1.03414528,  0.31776269, -0.44123633],
       [ 1.31587779,  0.7788403 , -0.56934333],
       [-1.13616987,  0.67021402,  2.16452277],
       [-0.0610451 ,  0.13519838,  1.15245343]])]
[●] Calculating batch loss
[✔] Successfully calculated batch loss
Batch Loss: 0.34072882525295173

[✔] Successfully built batch 24

[●] Calculating epoch loss
[✔] Successfully calculated epoch loss
Epoch Loss: 0.3007429329703777


[●] Predicting output ========================================

[●] Evaluating data 1
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: softmax
h00: 0.9052989050355136
h01: 0.09447970583074798
```

```
h02: 0.00022138913373845887

[●] Evaluating data 2
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: softmax
h00: 0.8974906066512021
h01: 0.10223853910415229
h02: 0.00027085424464564164

[●] Evaluating data 3
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: softmax
h00: 0.004759771932482818
h01: 0.3212212188393534
h02: 0.6740190092281638

[●] Evaluating data 4
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: softmax
h00: 0.10201923398091554
h01: 0.7163201863401496
h02: 0.1816605796789349

[●] Evaluating data 5
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: softmax
h00: 0.8907495032090677
h01: 0.10885447641344734
h02: 0.0003960203774849613

[●] Evaluating data 6
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: softmax
h00: 0.8968730391692853
h01: 0.10286138329907167
h02: 0.00026557753164298514

[●] Evaluating data 7
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
```

```
Activation function: softmax
h00: 0.0016437523352074112
h01: 0.1830731832530613
h02: 0.8152830644117314

[●] Evaluating data 8
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: softmax
h00: 0.8674962771143114
h01: 0.1320463208521802
h02: 0.0004574020335085012

[●] Evaluating data 9
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: softmax
h00: 0.0017687896215197443
h01: 0.21336315302095524
h02: 0.7848680573575251

[●] Evaluating data 10
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: softmax
h00: 0.0051094853297573625
h01: 0.472422827974002
h02: 0.5224676866962407

[●] Evaluating data 11
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: softmax
h00: 0.027373708238179358
h01: 0.5898911847977456
h02: 0.3827351069640749

[●] Evaluating data 12
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: softmax
h00: 0.9294023884646948
h01: 0.07050064552275903
h02: 9.696601254621613e-05

[●] Evaluating data 13
```

```
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: softmax
h00: 0.008561852620741741
h01: 0.3952986728273347
h02: 0.5961394745519235

[●] Evaluating data 14
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: softmax
h00: 0.13525751607189254
h01: 0.6915413374687779
h02: 0.17320114645932955

[●] Evaluating data 15
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: softmax
h00: 0.05522084382256771
h01: 0.7090215194434316
h02: 0.2357576367340006

[●] Evaluating data 16
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: softmax
h00: 0.06884158182019609
h01: 0.6984143475659508
h02: 0.23274407061385308

[●] Evaluating data 17
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: softmax
h00: 0.003498280162074948
h01: 0.34410100997976645
h02: 0.6524007098581586

[●] Evaluating data 18
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: softmax
h00: 0.0016846646757323947
h01: 0.21746361201568656
```

```
h02: 0.780851723308581

[●] Evaluating data 19
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: softmax
h00: 0.002919208899098641
h01: 0.2367640362756197
h02: 0.7603167548252817

[●] Evaluating data 20
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: softmax
h00: 0.8974906066512021
h01: 0.10223853910415229
h02: 0.00027085424464564164

[●] Evaluating data 21
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: softmax
h00: 0.014604843166121309
h01: 0.4854071331939647
h02: 0.49998802363991396

[●] Evaluating data 22
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: softmax
h00: 0.9572451471835489
h01: 0.042716873369129806
h02: 3.797944732115115e-05

[●] Evaluating data 23
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: softmax
h00: 0.8199129097609896
h01: 0.1793984113961256
h02: 0.0006886788428846635

[●] Evaluating data 24
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
```

```
Activation function: softmax
h00: 0.07136807250881273
h01: 0.6042853752110324
h02: 0.32434655228015496

[●] Evaluating data 25
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: softmax
h00: 0.07772118021519786
h01: 0.6898235492628468
h02: 0.2324552705219554

[●] Evaluating data 26
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: softmax
h00: 0.9248793263533573
h01: 0.07498660664945651
h02: 0.0001340669971861989

[●] Evaluating data 27
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: softmax
h00: 0.0055627310213398114
h01: 0.2701990706444096
h02: 0.7242381983342505

[●] Evaluating data 28
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: softmax
h00: 0.026691785557810856
h01: 0.599056123272674
h02: 0.3742520911695152

[●] Evaluating data 29
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: softmax
h00: 0.001622235110563521
h01: 0.2173830658440945
h02: 0.780994699045342

[●] Evaluating data 30
```

```
[●] Building layer 0
[✔] Successfully built layer 0
Number of neurons: 3
Activation function: softmax
h00: 0.024331698083059863
h01: 0.5514528359756196
h02: 0.4242154659413206

[✔] Successfully predicted output
[0 0 2 1 0 0 2 0 2 2 1 0 2 1 1 1 2 2 2 0 2 0 0 1 1 0 2 1 2 1]
Accuracy: 1.0
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        10
           1       1.00      1.00      1.00         9
           2       1.00      1.00      1.00        11

    accuracy                           1.00        30
   macro avg       1.00      1.00      1.00        30
weighted avg       1.00      1.00      1.00        30
```

III. Perbandingan dengan hasil perhitungan library

1. sigmoid.json

```
Final Weights :
[[[0.23291176286690732, 0.06015345619663067], [0.12884088056072765,
0.6484947353809505], [0.8376150008727261, 0.2315819927173646]]]
```

```
Final Weights:
[[[-1.1860206, 0.5108522], [1.7645199, -1.3445257]], [-0.11916187,
0.11916197]]
```

Tabel di atas merupakan perhitungan perubahan weights sigmoid.json dengan menggunakan algoritma dari scratch, sedangkan tabel di bawah merupakan perhitungan weights dengan menggunakan library. Hasil perhitungan berdasarkan scratch pada sigmoid.json cukup berbeda dengan hasil perhitungan dengan library. Hasil ini dapat disebabkan oleh beberapa faktor, di antaranya adalah jumlah iterasi yang sedikit yaitu hanya 10. Kemudian hal ini dipengaruhi oleh nilai random initial weights pada library. Berbeda dengan initial weights pada scratch, sudah ditentukan di awal nilainya. Untuk

perhitungan file json lainnya juga berlaku demikian, maka dari itu kami hanya menganalisis 1 hasil dari file sigmoid.json saja.

2. iris.csv

```
[0 0 2 1 0 0 2 0 2 2 1 0 2 1 1 1 2 2 2 0 2 0 0 1 1 0 2 1 2 1]
Accuracy: 1.0
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        10
           1       1.00      1.00      1.00         9
           2       1.00      1.00      1.00        11

    accuracy                           1.00        30
   macro avg       1.00      1.00      1.00        30
weighted avg       1.00      1.00      1.00        30
```

```
[0 0 2 1 0 0 2 0 2 2 1 0 2 1 1 1 2 2 2 0 2 0 0 1 1 0 2 1 2 1]
Accuracy: 1.00
                 precision    recall  f1-score   support

    Iris-setosa       1.00      1.00      1.00        10
Iris-versicolor       1.00      1.00      1.00         9
 Iris-virginica       1.00      1.00      1.00        11

       accuracy                           1.00        30
      macro avg       1.00      1.00      1.00        30
   weighted avg       1.00      1.00      1.00        30
```

Tabel di atas merupakan perhitungan prediksi iris.csv dengan menggunakan algoritma dari scratch, sedangkan tabel di bawah merupakan perhitungan prediksi dengan menggunakan library. Hasil perhitungan berdasarkan scratch pada iris.csv sudah sama dengan hasil perhitungan dengan library. Hasil ini dapat disebabkan oleh beberapa faktor, salah satunya adalah jumlah iterasi yang cukup banyak yaitu maksimal 300 kali. Walaupun nilai weights awal pada library random, tapi hal itu dapat diatasi karena banyaknya iterasi yang dibutuhkan.

IV. Pembagian tugas

- 13521001 Angger Ilham Amanullah: Kode FFNN, Laporan
- 13521005 Kelvin Rayhan Alkarim : Kode FFNN, Laporan
- 13521019 Ditra Rizqa Amadia : Kode FFNN, Laporan
- 13521021 Bernardus Willson : Kode FFNN, Laporan

V. Lampiran

- Github