In [25]:

```
import os
import time
import shelve
import random
import numpy as np
import pandas as pd
import tensorflow as tf
from pandas import DataFrame
import matplotlib.pyplot as plt
```

In [26]:

```
def load_data(name):

    if name == 'mnist':
        (X_train, y_train), (X_test, y_test) = tf.keras.datasets.mnist.load_data()
    elif name == 'fashion_mnist':
        (X_train, y_train), (X_test, y_test) = tf.keras.datasets.fashion_mnist.load_data()
    else:
        print('Only mnist or fashion_mnist.')
        return False

    imageSize  = X_train.shape[1]*X_train.shape[2]
    numClasses = np.max(y_train)+1

    X_train = np.reshape(X_train.astype(float)/255.0, (-1, 784))
    X_test  = np.reshape(X_test.astype(float)/255.0, (-1, 784))

    y_train = tf.keras.utils.to_categorical(y_train, num_classes=numClasses)
    y_test  = tf.keras.utils.to_categorical(y_test, num_classes=numClasses)

    X_val   = X_train[-10000:]
    y_val   = y_train[-10000:]
    X_train = X_train[:-10000]
    y_train = y_train[:-10000]

    print('Data Split: ')
    print(f'X_train: {X_train.shape}, y_train: {y_train.shape}')
    print(f'X_test : {X_test.shape }, y_test : {y_test.shape }')
    print(f'X_val  : {X_val.shape  }, y_val  : {y_val.shape  }')

    data = {}
    data['X_train'] = X_train
    data['y_train'] = y_train
    data['X_val']   = X_val
    data['y_val']   = y_val
    data['X_test']  = X_test
    data['y_test']  = y_test

    data['imageSize'] = imageSize

    return data
```

In [27]:

```
class MLP(object):

    def __init__(self, name, size_input, size_hidden, size_output, learning_rate=0.01, optimizer='SGD', weight_coeff=1,\
                 Reg=None, RegC=0, training=None, validation=None, accuracy=0, device=None):

        self.name          = name
        self.size_input    = size_input
        self.size_hidden   = size_hidden
        self.size_output   = size_output
        self.learning_rate = learning_rate
        self.optimizer     = optimizer
        self.Reg           = Reg
        self.RegC          = RegC
        self.training      = training
        self.validation    = validation
        self.accuracy      = accuracy
        self.device        = device
        self.weight_coeff  = weight_coeff

        self.W1 = self.initWeights(self.size_input, self.size_hidden[0], self.weight_coeff)
        self.b1 = self.initWeights(1, self.size_hidden[0], self.weight_coeff)

        self.W2 = self.initWeights(self.size_hidden[0], self.size_hidden[1], self.weight_coeff)
        self.b2 = self.initWeights(1, self.size_hidden[1], self.weight_coeff)

        self.W3 = self.initWeights(self.size_hidden[1], self.size_hidden[2], self.weight_coeff)
        self.b3 = self.initWeights(1, self.size_hidden[2], self.weight_coeff)

        self.W4 = self.initWeights(self.size_hidden[2], self.size_output, self.weight_coeff)
        self.b4 = self.initWeights(1, self.size_output, self.weight_coeff)

        self.varibles = [self.W1, self.b1, self.W2, self.b2, self.W3, self.b3, self.W4, self.b4]


    def initWeights(self, rows, columns, multFactor=1):
        return tf.Variable(multFactor*tf.random.normal([rows, columns]))

    def forward(self, X):

        if self.device is not None:
            with tf.device('gpu:0' if self.device=='gpu' else 'cpu'):
                self.y = self.compute_output(X)
        else:
            self.y = self.compute_output(X)

        return self.y
    def getRegLoss(self, X_train):
        
        if self.Reg=='L2':
            return (self.RegC/X_train.shape[0])*(tf.reduce_sum(tf.math.square(self
.W1)) +
                                                 tf.reduce_sum(tf.math.square(self
.W2)) +
                                                 tf.reduce_sum(tf.math.square(self
.W3)) +
                                                 tf.reduce_sum(tf.math.square(self
.W4)))
        
        elif self.Reg=='L1':
            return (self.RegC/X_train.shape[0])*tf.abs(tf.reduce_sum(self.W1) +
                                                       tf.reduce_sum(self.W2) +
                                                       tf.reduce_sum(self.W3) +
                                                       tf.reduce_sum(self.W4))
        
        elif self.Reg=='L1+L2':
            L2 =  (self.RegC/X_train.shape[0])*(tf.reduce_sum(tf.math.square(self.
W1)) +
                                                tf.reduce_sum(tf.math.square(self.
W2)) +
                                                tf.reduce_sum(tf.math.square(self.
W3)) +
                                                tf.reduce_sum(tf.math.square(self.
W4)))
            
            L1 = (self.RegC/X_train.shape[0])*tf.abs(tf.reduce_sum(self.W1) +
                                                     tf.reduce_sum(self.W2) +
                                                     tf.reduce_sum(self.W3) +
                                                     tf.reduce_sum(self.W4))
            return L1+L2
        
        else:
            return 0
    
    def loss(self, y_pred, y_true):
        
        y_true_tf = tf.cast(tf.reshape(y_true, (-1, self.size_output)), dtype=tf.f
loat32)
        y_pred_tf = tf.cast(y_pred, dtype=tf.float32)
        
        loss = tf.keras.losses.CategoricalCrossentropy()(y_true_tf, y_pred_tf)
        return loss
    
    def backward(self, X_train, y_train):
        
        if self.optimizer=='SGD':
            optimizer = tf.keras.optimizers.SGD(learning_rate=self.learning_rate)
        
        elif self.optimizer=='Adam':
            optimizer = tf.keras.optimizers.Adam(learning_rate=self.learning_rate)
        
        elif self.optimizer=='RMSProp':
            optimizer = tf.keras.optimizers.RMSprop(learning_rate=self.learning_rate)

        else:
            pass

        if self.Reg is not None and self.RegC==0:
            print('Regularization coffecient argument was 0, seeting it to default lamda=0.01')
            self.RegC = 0.01;

        with tf.GradientTape() as tape:
            predicted = self.forward(X_train)
            current_loss = self.loss(predicted, y_train)
            current_loss += self.getRegLoss(X_train)

        grads = tape.gradient(current_loss, self.varibles)
        optimizer.apply_gradients(zip(grads, self.varibles))

    def compute_output(self, X):

        X_tf = tf.cast(X, dtype=tf.float32)

        w1Hat = tf.matmul(X_tf,  self.W1) + self.b1
        h1Hat = tf.nn.relu(w1Hat)

        w2Hat = tf.matmul(h1Hat, self.W2) + self.b2
        h2Hat = tf.nn.relu(w2Hat)

        w3Hat = tf.matmul(h2Hat, self.W3) + self.b3
        h3Hat = tf.nn.relu(w3Hat)

        w4Hat = tf.matmul(h3Hat, self.W4) + self.b4
        output = tf.nn.softmax(w4Hat)

        return output

    def getAccuracy(self, predictions, outputs):
        preds  = np.argmax(predictions, axis=1)
        y_true = np.argmax(outputs, axis=1)

        return (preds==y_true).mean()
```

In [28]:

```
def trainModel(model, data, NUM_EPOCHS=10, batchSize=50, seedVal=1234):

    X_train = data['X_train']
    y_train = data['y_train']
    X_val   = data['X_val']
    y_val   = data['y_val']

    training = np.zeros(shape=(NUM_EPOCHS, 3))
    validation = np.zeros(shape=(NUM_EPOCHS, 3))

    train_ds = tf.data.Dataset.from_tensor_slices((X_train, y_train)).batch(batchS
ize)
    val_ds   = tf.data.Dataset.from_tensor_slices((X_val, y_val)).batch(batchSize)

    print(f'\n\n**************** Training model: {model.name} with optimizer: {mod
el.optimizer} and seed: {seedVal} ****************\n')
    time_start = time.time()
    for epoch in range(NUM_EPOCHS):
        train_loss = tf.zeros([1, 1], dtype=tf.float32)
        val_loss   = tf.zeros([1, 1], dtype=tf.float32)

        train_ds = tf.data.Dataset.from_tensor_slices((X_train, y_train)).shuffle(
25, seed = epoch*(seedVal)).batch(batchSize)
        val_ds   = tf.data.Dataset.from_tensor_slices((X_val, y_val)).shuffle(25,
seed = epoch*(seedVal)).batch(batchSize)

        for inputs, outputs in train_ds:
            train_pred = model.forward(inputs)
            train_loss = train_loss + model.loss(train_pred, outputs)
            model.backward(inputs, outputs)
            train_acc = model.getAccuracy(train_pred, outputs)

        for inputs, outputs in val_ds:
            val_pred = model.forward(inputs)
            val_loss = val_loss + model.loss(val_pred, outputs)
            val_acc  = model.getAccuracy(val_pred, outputs)

        # train_loss = np.array(train_loss)
        # val_loss = np.array(val_loss)

        training[epoch] = [epoch+1, train_acc, np.sum(train_loss)/X_train.shape[0
]]
        validation[epoch] = [epoch+1, val_acc, np.sum(train_loss)/X_train.shape[0
]]

        print('# Epoch:={}/{}  - train loss:={:.4f} - val loss:={:.4f}, train acc:
={:.2f} - val acc:={:.2f}'\
              .format(epoch+1, NUM_EPOCHS, np.sum(train_loss)/X_train.shape[0], np
.sum(val_loss)/X_val.shape[0], train_acc, val_acc))

    time_taken = time.time()-time_start
    print(f'\nTotal time taken (in seconds): {time_taken: .2f}')
    print(f'\nFinished training model: {model.name}\n')
    model.training = training
    model.validation = validation

def testModel(model, data):

    X_test = data['X_test']
    y_test = data['y_test']

    preds = model.forward(X_test)

    pred = np.argmax(preds, axis=1)
    y_true= np.argmax(y_test, axis=1)

    model.accuracy = (pred==y_true).mean()*100

    print(f'**************** Testing ****************')
    print(f'{model.name} model accuracy = {model.accuracy:.2f}%')
    print(f'*****************************************')

def plotAccuracyAndLoss(model):

    training = model.training
    validation = model.validation
    fig, (ax1, ax2) = plt.subplots(1, 2)
    training[:, -1] = training[:, -1]/np.linalg.norm(training[:, -1])
    ax1.plot(training[:,0], training[:,1], 'g')
    ax1.plot(training[:,0], training[:,2], 'b')
    ax1.set_title('Training')
    ax1.legend(["Accuracy", "Loss"])

    validation[:, -1] = validation[:, -1]/np.linalg.norm(validation[:, -1])
    ax2.plot(validation[:,0], validation[:,1], 'g')
    ax2.plot(validation[:,0], validation[:,2], 'b')
    ax2.set_title('Validation')
    ax2.legend(["Accuracy", "Loss"])
    plt.show()
```

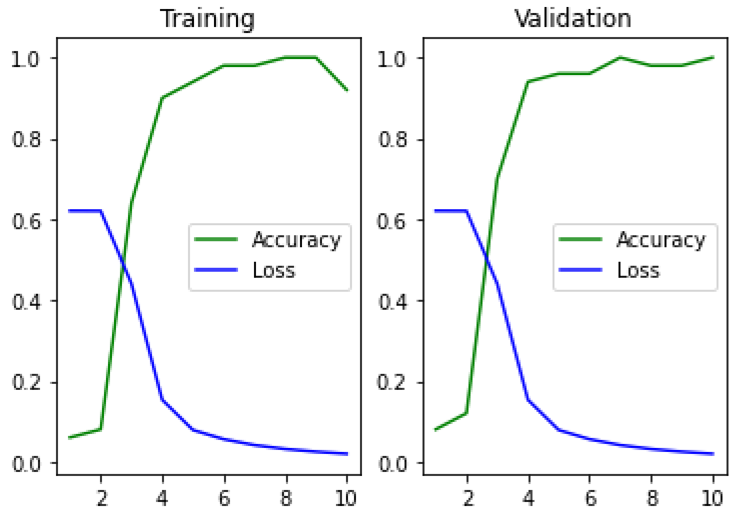
In [29]:

```
def main():

    for j in range(2):
        if j==0:
            data = load_data('mnist')
            size_hidden = [512, 256, 64]
            learning_rate = 0.05
            weight_coeff = 0.01

        if j==1:
            data = load_data('fashion_mnist')
            size_hidden = [1024, 512, 256]
            learning_rate = 0.09
            weight_coeff = 0.05

        for k in range(3):
            if k==0:
                opt = 'SGD'
            elif k==1:
                opt = 'Adam'
            elif k==2:
                opt = 'RMSProp'
            else:
                pass

            imageSize = data['imageSize']

            size_input  = imageSize
            size_output = 10

            allModels = {}
            allModels['mlp_on_gpu_default'] = {}
            allModels['mlp_on_gpu_RegL1']   = {}
            allModels['mlp_on_gpu_RegL2']   = {}

            for model_name in allModels:
                model = allModels[model_name]

                cnt = -1

                numEpochs = 10
                batchSize = 50
                numTrials = 10

                seeds = random.sample(range(1000, 9999), numTrials)

                # loss     = np.zeros(shape=(numEpochs, 1))
                accuracy = np.zeros(shape=(numTrials, 1))

                for i in seeds:
                    cnt += 1

                    np.random.seed(i)
                    tf.random.set_seed(i)

                    print(f'Count: {cnt}, j=: {j}')
                    if model_name == 'mlp_on_gpu_default':
                        model['name'] = MLP('mlp_on_gpu_default', size_input, size
_hidden, size_output, learning_rate, opt, weight_coeff,\
                                             device='gpu')

                    elif model_name == 'mlp_on_gpu_RegL1':
                        model['name'] = MLP('mlp_on_gpu_RegL1', size_input, size_h
idden, size_output, learning_rate, opt, weight_coeff,\
                                            'L1', 0.01, device='gpu')

                    elif model_name == 'mlp_on_gpu_RegL2':
                        model['name'] = MLP('mlp_on_gpu_RegL2', size_input, size_h
idden, size_output, learning_rate, opt, weight_coeff,\
                                            'L2', 0.01, device='gpu')

                    else:
                        pass

                    trainModel(model['name'], data, numEpochs, batchSize, i)
                    testModel(model['name'], data)

                    accuracy[cnt] = model['name'].accuracy

                    plotAccuracyAndLoss(model['name'])

                    allModels[model_name][i] = model['name']
                    allModels[model_name]['Accuracy'] = [np.mean(accuracy), np.var
(accuracy)]

        if j==0:
            mnist = allModels
        elif j==1:
            fashion_mnist = allModels
        else:
            pass



    return mnist, fashion_mnist
```

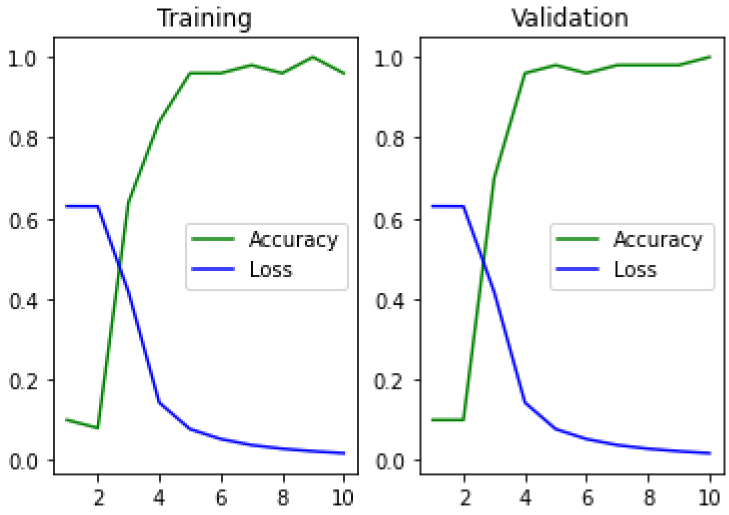
In [ ]:

```
if __name__ == "__main__":
    mnist, fashion_mnist = main()
```

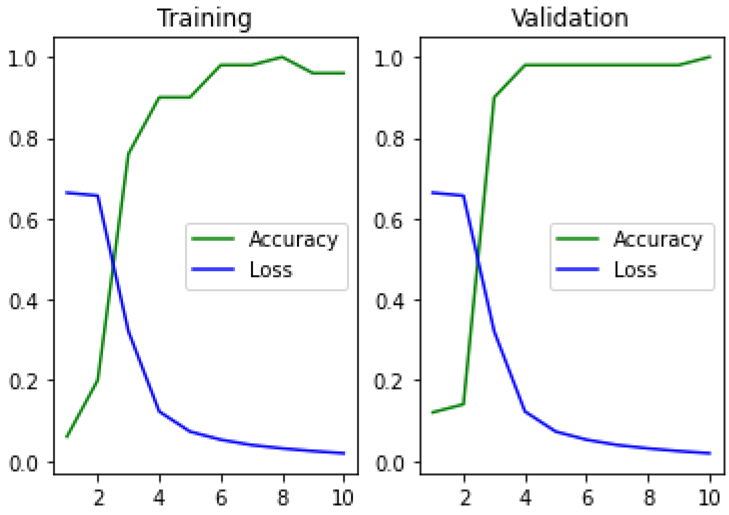
```
Data Split:
X_train: (50000, 784), y_train: (50000, 10)
X_test : (10000, 784), y_test : (10000, 10)
X_val  : (10000, 784), y_val  : (10000, 10)
Count: 0, j=: 0


**************** Training model: mlp_on_gpu_default with optimizer: SG
D and seed: 2191 ****************

# Epoch:=1/10  - train loss:=0.0460 - val loss:=0.0460, train acc:=0.0
6 - val acc:=0.08
# Epoch:=2/10  - train loss:=0.0460 - val loss:=0.0459, train acc:=0.0
8 - val acc:=0.12
# Epoch:=3/10  - train loss:=0.0327 - val loss:=0.0159, train acc:=0.6
4 - val acc:=0.70
# Epoch:=4/10  - train loss:=0.0114 - val loss:=0.0068, train acc:=0.9
0 - val acc:=0.94
# Epoch:=5/10  - train loss:=0.0058 - val loss:=0.0047, train acc:=0.9
4 - val acc:=0.96
# Epoch:=6/10  - train loss:=0.0041 - val loss:=0.0038, train acc:=0.9
8 - val acc:=0.96
# Epoch:=7/10  - train loss:=0.0030 - val loss:=0.0033, train acc:=0.9
8 - val acc:=1.00
# Epoch:=8/10  - train loss:=0.0023 - val loss:=0.0030, train acc:=1.0
0 - val acc:=0.98
# Epoch:=9/10  - train loss:=0.0018 - val loss:=0.0027, train acc:=1.0
0 - val acc:=0.98
# Epoch:=10/10  - train loss:=0.0014 - val loss:=0.0025, train acc:=0.
92 - val acc:=1.00

Total time taken (in seconds):  147.20

Finished training model: mlp_on_gpu_default

**************** Testing ****************
mlp_on_gpu_default model accuracy = 96.70%
*****************************************
```



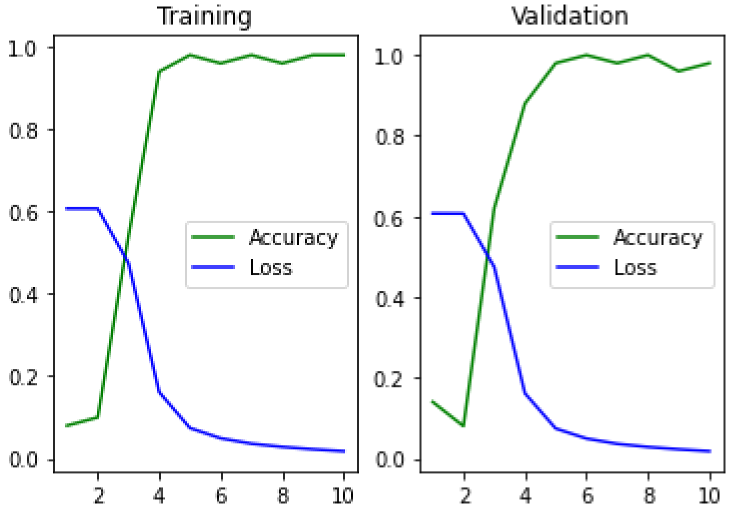
```
Count: 1, j=: 0


**************** Training model: mlp_on_gpu_default with optimizer: SG
D and seed: 7240 ****************

# Epoch:=1/10  - train loss:=0.0460 - val loss:=0.0460, train acc:=0.1
0 - val acc:=0.10
# Epoch:=2/10  - train loss:=0.0460 - val loss:=0.0459, train acc:=0.0
8 - val acc:=0.10
# Epoch:=3/10  - train loss:=0.0305 - val loss:=0.0154, train acc:=0.6
4 - val acc:=0.70
# Epoch:=4/10  - train loss:=0.0104 - val loss:=0.0065, train acc:=0.8
4 - val acc:=0.96
# Epoch:=5/10  - train loss:=0.0056 - val loss:=0.0045, train acc:=0.9
6 - val acc:=0.98
# Epoch:=6/10  - train loss:=0.0038 - val loss:=0.0035, train acc:=0.9
6 - val acc:=0.96
# Epoch:=7/10  - train loss:=0.0028 - val loss:=0.0032, train acc:=0.9
8 - val acc:=0.98
# Epoch:=8/10  - train loss:=0.0021 - val loss:=0.0028, train acc:=0.9
6 - val acc:=0.98
# Epoch:=9/10  - train loss:=0.0016 - val loss:=0.0028, train acc:=1.0
0 - val acc:=0.98
# Epoch:=10/10  - train loss:=0.0013 - val loss:=0.0026, train acc:=0.
96 - val acc:=1.00

Total time taken (in seconds):  147.88

Finished training model: mlp_on_gpu_default

**************** Testing ****************
mlp_on_gpu_default model accuracy = 96.30%
*****************************************
```



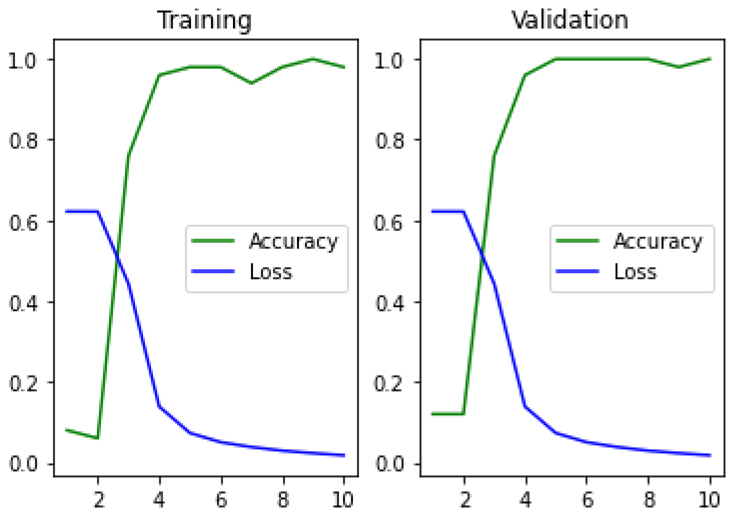
```
Count: 2, j=: 0


**************** Training model: mlp_on_gpu_default with optimizer: SG
D and seed: 8662 ****************

# Epoch:=1/10  - train loss:=0.0460 - val loss:=0.0460, train acc:=0.0
6 - val acc:=0.12
# Epoch:=2/10  - train loss:=0.0455 - val loss:=0.0388, train acc:=0.2
0 - val acc:=0.14
# Epoch:=3/10  - train loss:=0.0223 - val loss:=0.0117, train acc:=0.7
6 - val acc:=0.90
# Epoch:=4/10  - train loss:=0.0085 - val loss:=0.0057, train acc:=0.9
0 - val acc:=0.98
# Epoch:=5/10  - train loss:=0.0050 - val loss:=0.0046, train acc:=0.9
0 - val acc:=0.98
# Epoch:=6/10  - train loss:=0.0036 - val loss:=0.0035, train acc:=0.9
8 - val acc:=0.98
# Epoch:=7/10  - train loss:=0.0027 - val loss:=0.0032, train acc:=0.9
8 - val acc:=0.98
# Epoch:=8/10  - train loss:=0.0021 - val loss:=0.0029, train acc:=1.0
0 - val acc:=0.98
# Epoch:=9/10  - train loss:=0.0017 - val loss:=0.0029, train acc:=0.9
6 - val acc:=0.98
# Epoch:=10/10  - train loss:=0.0013 - val loss:=0.0028, train acc:=0.
96 - val acc:=1.00

Total time taken (in seconds):  147.44

Finished training model: mlp_on_gpu_default

**************** Testing ****************
mlp_on_gpu_default model accuracy = 96.39%
*****************************************
```



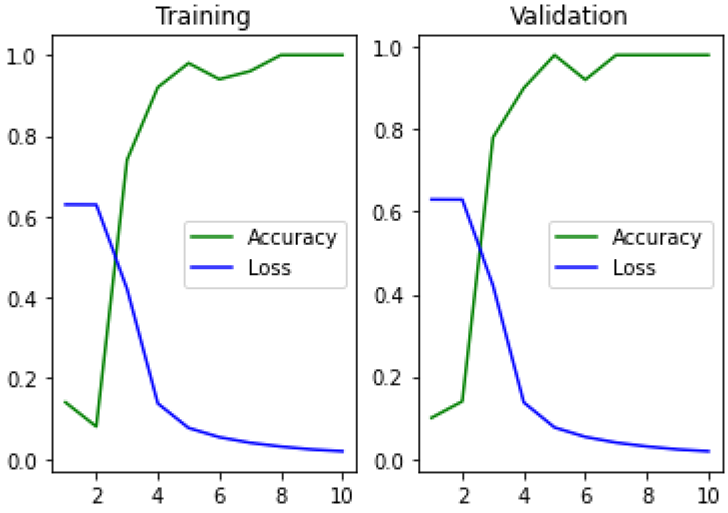
```
Count: 3, j=: 0


**************** Training model: mlp_on_gpu_default with optimizer: SG
D and seed: 2208 ****************

# Epoch:=1/10  - train loss:=0.0460 - val loss:=0.0460, train acc:=0.0
8 - val acc:=0.14
# Epoch:=2/10  - train loss:=0.0460 - val loss:=0.0460, train acc:=0.1
0 - val acc:=0.08
# Epoch:=3/10  - train loss:=0.0360 - val loss:=0.0193, train acc:=0.5
4 - val acc:=0.62
# Epoch:=4/10  - train loss:=0.0122 - val loss:=0.0066, train acc:=0.9
4 - val acc:=0.88
# Epoch:=5/10  - train loss:=0.0056 - val loss:=0.0042, train acc:=0.9
8 - val acc:=0.98
# Epoch:=6/10  - train loss:=0.0037 - val loss:=0.0033, train acc:=0.9
6 - val acc:=1.00
# Epoch:=7/10  - train loss:=0.0027 - val loss:=0.0029, train acc:=0.9
8 - val acc:=0.98
# Epoch:=8/10  - train loss:=0.0021 - val loss:=0.0024, train acc:=0.9
6 - val acc:=1.00
# Epoch:=9/10  - train loss:=0.0017 - val loss:=0.0024, train acc:=0.9
8 - val acc:=0.96
# Epoch:=10/10  - train loss:=0.0014 - val loss:=0.0023, train acc:=0.
98 - val acc:=0.98

Total time taken (in seconds):  147.42

Finished training model: mlp_on_gpu_default

**************** Testing ****************
mlp_on_gpu_default model accuracy = 96.42%
*****************************************
```



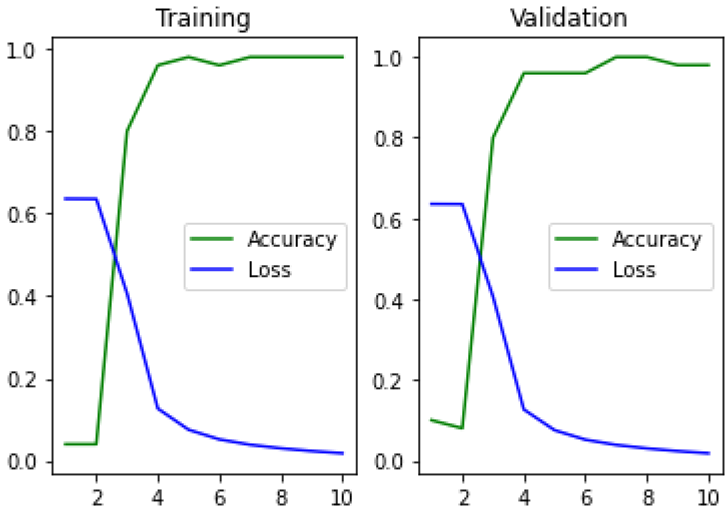
```
Count: 4, j=: 0


**************** Training model: mlp_on_gpu_default with optimizer: SG
D and seed: 1724 ****************

# Epoch:=1/10  - train loss:=0.0460 - val loss:=0.0460, train acc:=0.0
8 - val acc:=0.12
# Epoch:=2/10  - train loss:=0.0460 - val loss:=0.0460, train acc:=0.0
6 - val acc:=0.12
# Epoch:=3/10  - train loss:=0.0329 - val loss:=0.0156, train acc:=0.7
6 - val acc:=0.76
# Epoch:=4/10  - train loss:=0.0103 - val loss:=0.0065, train acc:=0.9
6 - val acc:=0.96
# Epoch:=5/10  - train loss:=0.0054 - val loss:=0.0041, train acc:=0.9
8 - val acc:=1.00
# Epoch:=6/10  - train loss:=0.0037 - val loss:=0.0033, train acc:=0.9
8 - val acc:=1.00
# Epoch:=7/10  - train loss:=0.0028 - val loss:=0.0028, train acc:=0.9
4 - val acc:=1.00
# Epoch:=8/10  - train loss:=0.0022 - val loss:=0.0025, train acc:=0.9
8 - val acc:=1.00
# Epoch:=9/10  - train loss:=0.0017 - val loss:=0.0027, train acc:=1.0
0 - val acc:=0.98
# Epoch:=10/10  - train loss:=0.0013 - val loss:=0.0023, train acc:=0.
98 - val acc:=1.00

Total time taken (in seconds):  148.01

Finished training model: mlp_on_gpu_default

**************** Testing ****************
mlp_on_gpu_default model accuracy = 96.55%
*****************************************
```



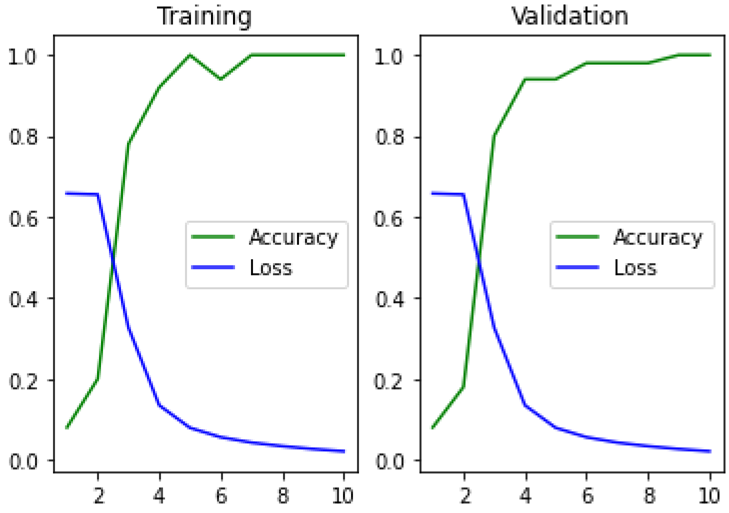
```
Count: 5, j=: 0


**************** Training model: mlp_on_gpu_default with optimizer: SG
D and seed: 7423 ****************

# Epoch:=1/10  - train loss:=0.0460 - val loss:=0.0460, train acc:=0.1
4 - val acc:=0.10
# Epoch:=2/10  - train loss:=0.0460 - val loss:=0.0459, train acc:=0.0
8 - val acc:=0.14
# Epoch:=3/10  - train loss:=0.0308 - val loss:=0.0150, train acc:=0.7
4 - val acc:=0.78
# Epoch:=4/10  - train loss:=0.0100 - val loss:=0.0066, train acc:=0.9
2 - val acc:=0.90
# Epoch:=5/10  - train loss:=0.0056 - val loss:=0.0046, train acc:=0.9
8 - val acc:=0.98
# Epoch:=6/10  - train loss:=0.0039 - val loss:=0.0036, train acc:=0.9
4 - val acc:=0.92
# Epoch:=7/10  - train loss:=0.0029 - val loss:=0.0033, train acc:=0.9
6 - val acc:=0.98
# Epoch:=8/10  - train loss:=0.0022 - val loss:=0.0029, train acc:=1.0
0 - val acc:=0.98
# Epoch:=9/10  - train loss:=0.0017 - val loss:=0.0026, train acc:=1.0
0 - val acc:=0.98
# Epoch:=10/10  - train loss:=0.0014 - val loss:=0.0026, train acc:=1.
00 - val acc:=0.98

Total time taken (in seconds):  147.74

Finished training model: mlp_on_gpu_default

**************** Testing ****************
mlp_on_gpu_default model accuracy = 96.16%
*****************************************
```



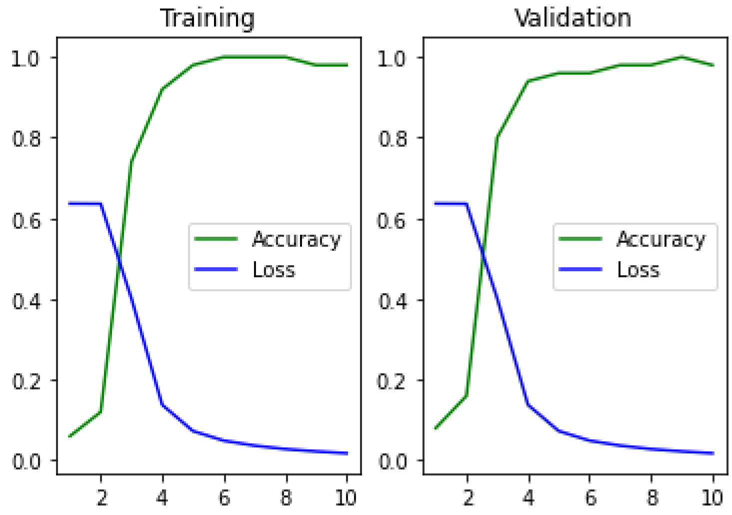
```
Count: 6, j=: 0


**************** Training model: mlp_on_gpu_default with optimizer: SG
D and seed: 7351 ****************

# Epoch:=1/10  - train loss:=0.0460 - val loss:=0.0460, train acc:=0.0
4 - val acc:=0.10
# Epoch:=2/10  - train loss:=0.0460 - val loss:=0.0459, train acc:=0.0
4 - val acc:=0.08
# Epoch:=3/10  - train loss:=0.0294 - val loss:=0.0138, train acc:=0.8
0 - val acc:=0.80
# Epoch:=4/10  - train loss:=0.0092 - val loss:=0.0065, train acc:=0.9
6 - val acc:=0.96
# Epoch:=5/10  - train loss:=0.0054 - val loss:=0.0043, train acc:=0.9
8 - val acc:=0.96
# Epoch:=6/10  - train loss:=0.0037 - val loss:=0.0033, train acc:=0.9
6 - val acc:=0.96
# Epoch:=7/10  - train loss:=0.0028 - val loss:=0.0029, train acc:=0.9
8 - val acc:=1.00
# Epoch:=8/10  - train loss:=0.0022 - val loss:=0.0026, train acc:=0.9
8 - val acc:=1.00
# Epoch:=9/10  - train loss:=0.0017 - val loss:=0.0025, train acc:=0.9
8 - val acc:=0.98
# Epoch:=10/10  - train loss:=0.0013 - val loss:=0.0025, train acc:=0.
98 - val acc:=0.98

Total time taken (in seconds):  147.08

Finished training model: mlp_on_gpu_default

**************** Testing ****************
mlp_on_gpu_default model accuracy = 96.48%
*****************************************
```



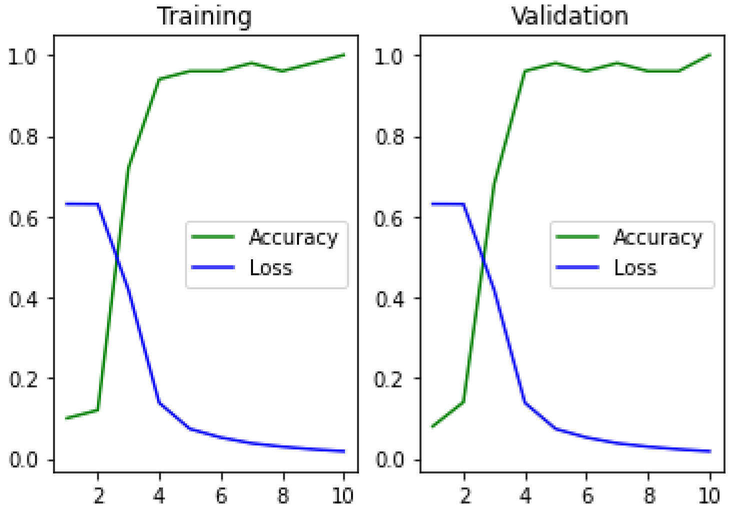
```
Count: 7, j=: 0


**************** Training model: mlp_on_gpu_default with optimizer: SG
D and seed: 3598 ****************

# Epoch:=1/10  - train loss:=0.0460 - val loss:=0.0460, train acc:=0.0
8 - val acc:=0.08
# Epoch:=2/10  - train loss:=0.0459 - val loss:=0.0429, train acc:=0.2
0 - val acc:=0.18
# Epoch:=3/10  - train loss:=0.0229 - val loss:=0.0119, train acc:=0.7
8 - val acc:=0.80
# Epoch:=4/10  - train loss:=0.0094 - val loss:=0.0066, train acc:=0.9
2 - val acc:=0.94
# Epoch:=5/10  - train loss:=0.0055 - val loss:=0.0047, train acc:=1.0
0 - val acc:=0.94
# Epoch:=6/10  - train loss:=0.0039 - val loss:=0.0042, train acc:=0.9
4 - val acc:=0.98
# Epoch:=7/10  - train loss:=0.0030 - val loss:=0.0035, train acc:=1.0
0 - val acc:=0.98
# Epoch:=8/10  - train loss:=0.0024 - val loss:=0.0034, train acc:=1.0
0 - val acc:=0.98
# Epoch:=9/10  - train loss:=0.0019 - val loss:=0.0031, train acc:=1.0
0 - val acc:=1.00
# Epoch:=10/10  - train loss:=0.0015 - val loss:=0.0030, train acc:=1.
00 - val acc:=1.00

Total time taken (in seconds):  146.29

Finished training model: mlp_on_gpu_default

**************** Testing ****************
mlp_on_gpu_default model accuracy = 96.05%
*****************************************
```



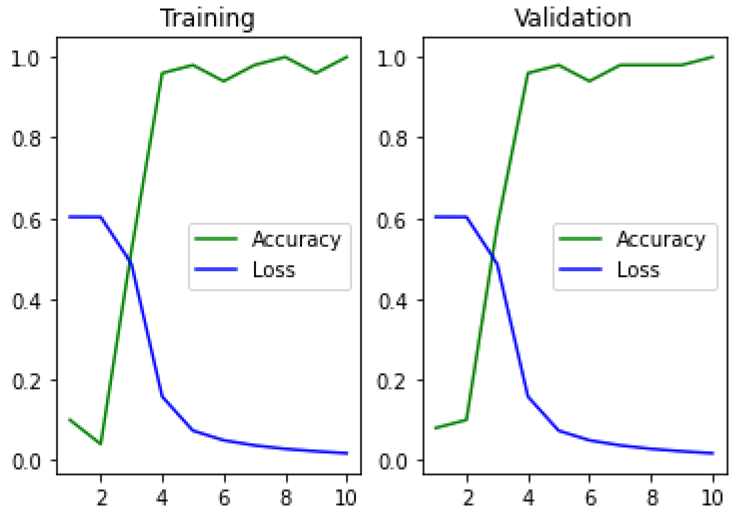
```
Count: 8, j=: 0


**************** Training model: mlp_on_gpu_default with optimizer: SG
D and seed: 7007 ****************

# Epoch:=1/10  - train loss:=0.0460 - val loss:=0.0460, train acc:=0.0
6 - val acc:=0.08
# Epoch:=2/10  - train loss:=0.0460 - val loss:=0.0458, train acc:=0.1
2 - val acc:=0.16
# Epoch:=3/10  - train loss:=0.0291 - val loss:=0.0131, train acc:=0.7
4 - val acc:=0.80
# Epoch:=4/10  - train loss:=0.0100 - val loss:=0.0064, train acc:=0.9
2 - val acc:=0.94
# Epoch:=5/10  - train loss:=0.0053 - val loss:=0.0042, train acc:=0.9
8 - val acc:=0.96
# Epoch:=6/10  - train loss:=0.0035 - val loss:=0.0034, train acc:=1.0
0 - val acc:=0.96
# Epoch:=7/10  - train loss:=0.0026 - val loss:=0.0029, train acc:=1.0
0 - val acc:=0.98
# Epoch:=8/10  - train loss:=0.0020 - val loss:=0.0027, train acc:=1.0
0 - val acc:=0.98
# Epoch:=9/10  - train loss:=0.0016 - val loss:=0.0025, train acc:=0.9
8 - val acc:=1.00
# Epoch:=10/10  - train loss:=0.0013 - val loss:=0.0025, train acc:=0.
98 - val acc:=0.98

Total time taken (in seconds):  146.17

Finished training model: mlp_on_gpu_default

**************** Testing ****************
mlp_on_gpu_default model accuracy = 96.75%
*****************************************
```



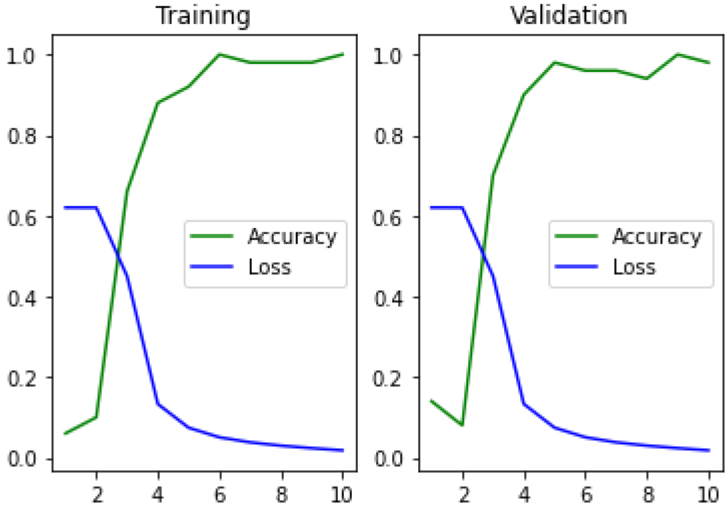
```
Count: 9, j=: 0


**************** Training model: mlp_on_gpu_default with optimizer: SG
D and seed: 8841 ****************

# Epoch:=1/10  - train loss:=0.0460 - val loss:=0.0460, train acc:=0.1
0 - val acc:=0.08
# Epoch:=2/10  - train loss:=0.0460 - val loss:=0.0458, train acc:=0.1
2 - val acc:=0.14
# Epoch:=3/10  - train loss:=0.0305 - val loss:=0.0148, train acc:=0.7
2 - val acc:=0.68
# Epoch:=4/10  - train loss:=0.0101 - val loss:=0.0061, train acc:=0.9
4 - val acc:=0.96
# Epoch:=5/10  - train loss:=0.0054 - val loss:=0.0044, train acc:=0.9
6 - val acc:=0.98
# Epoch:=6/10  - train loss:=0.0038 - val loss:=0.0040, train acc:=0.9
6 - val acc:=0.96
# Epoch:=7/10  - train loss:=0.0028 - val loss:=0.0032, train acc:=0.9
8 - val acc:=0.98
# Epoch:=8/10  - train loss:=0.0021 - val loss:=0.0029, train acc:=0.9
6 - val acc:=0.96
# Epoch:=9/10  - train loss:=0.0017 - val loss:=0.0028, train acc:=0.9
8 - val acc:=0.96
# Epoch:=10/10  - train loss:=0.0013 - val loss:=0.0026, train acc:=1.
00 - val acc:=1.00

Total time taken (in seconds):  146.27

Finished training model: mlp_on_gpu_default

**************** Testing ****************
mlp_on_gpu_default model accuracy = 96.67%
*****************************************
```



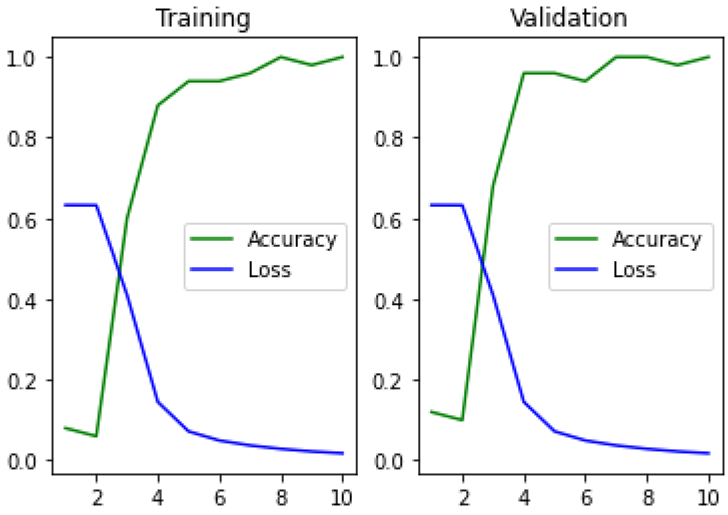
```
Count: 0, j=: 0


**************** Training model: mlp_on_gpu_RegL1 with optimizer: SGD
and seed: 1575 ****************

# Epoch:=1/10  - train loss:=0.0460 - val loss:=0.0460, train acc:=0.1
0 - val acc:=0.08
# Epoch:=2/10  - train loss:=0.0460 - val loss:=0.0460, train acc:=0.0
4 - val acc:=0.10
# Epoch:=3/10  - train loss:=0.0372 - val loss:=0.0205, train acc:=0.5
2 - val acc:=0.58
# Epoch:=4/10  - train loss:=0.0121 - val loss:=0.0065, train acc:=0.9
6 - val acc:=0.96
# Epoch:=5/10  - train loss:=0.0056 - val loss:=0.0044, train acc:=0.9
8 - val acc:=0.98
# Epoch:=6/10  - train loss:=0.0038 - val loss:=0.0034, train acc:=0.9
4 - val acc:=0.94
# Epoch:=7/10  - train loss:=0.0028 - val loss:=0.0030, train acc:=0.9
8 - val acc:=0.98
# Epoch:=8/10  - train loss:=0.0021 - val loss:=0.0028, train acc:=1.0
0 - val acc:=0.98
# Epoch:=9/10  - train loss:=0.0017 - val loss:=0.0025, train acc:=0.9
6 - val acc:=0.98
# Epoch:=10/10  - train loss:=0.0013 - val loss:=0.0025, train acc:=1.
00 - val acc:=1.00

Total time taken (in seconds):  173.11

Finished training model: mlp_on_gpu_RegL1

**************** Testing ****************
mlp_on_gpu_RegL1 model accuracy = 96.32%
*****************************************
```



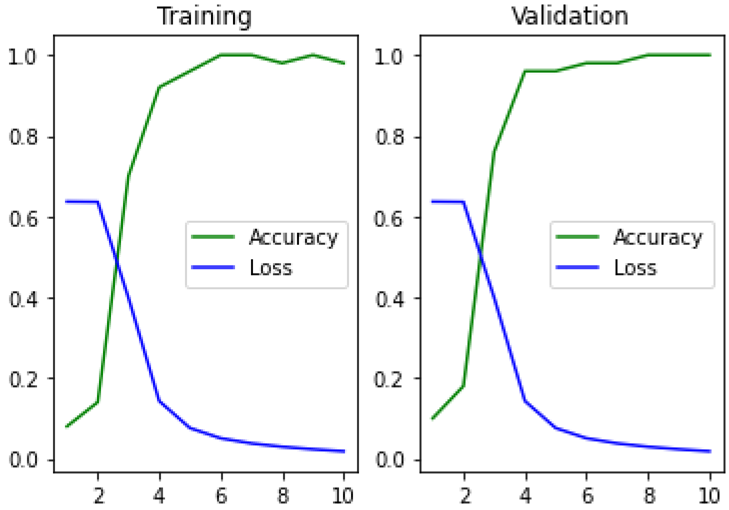
```
Count: 1, j=: 0


**************** Training model: mlp_on_gpu_RegL1 with optimizer: SGD
and seed: 6829 ****************

# Epoch:=1/10  - train loss:=0.0460 - val loss:=0.0460, train acc:=0.0
6 - val acc:=0.14
# Epoch:=2/10  - train loss:=0.0460 - val loss:=0.0460, train acc:=0.1
0 - val acc:=0.08
# Epoch:=3/10  - train loss:=0.0335 - val loss:=0.0150, train acc:=0.6
6 - val acc:=0.70
# Epoch:=4/10  - train loss:=0.0099 - val loss:=0.0068, train acc:=0.8
8 - val acc:=0.90
# Epoch:=5/10  - train loss:=0.0055 - val loss:=0.0043, train acc:=0.9
2 - val acc:=0.98
# Epoch:=6/10  - train loss:=0.0037 - val loss:=0.0034, train acc:=1.0
0 - val acc:=0.96
# Epoch:=7/10  - train loss:=0.0028 - val loss:=0.0030, train acc:=0.9
8 - val acc:=0.96
# Epoch:=8/10  - train loss:=0.0022 - val loss:=0.0027, train acc:=0.9
8 - val acc:=0.94
# Epoch:=9/10  - train loss:=0.0017 - val loss:=0.0026, train acc:=0.9
8 - val acc:=1.00
# Epoch:=10/10  - train loss:=0.0013 - val loss:=0.0024, train acc:=1.
00 - val acc:=0.98

Total time taken (in seconds):  171.62

Finished training model: mlp_on_gpu_RegL1

**************** Testing ****************
mlp_on_gpu_RegL1 model accuracy = 96.57%
*****************************************
```



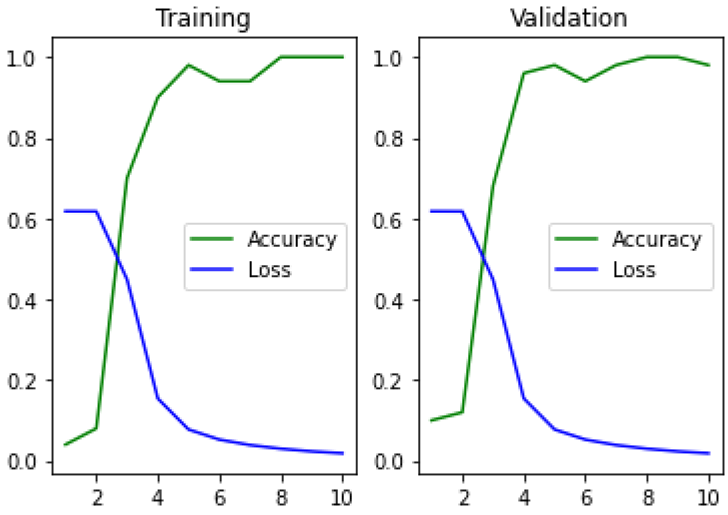
```
Count: 2, j=: 0


**************** Training model: mlp_on_gpu_RegL1 with optimizer: SGD
and seed: 3136 ****************

# Epoch:=1/10  - train loss:=0.0460 - val loss:=0.0460, train acc:=0.0
8 - val acc:=0.12
# Epoch:=2/10  - train loss:=0.0460 - val loss:=0.0459, train acc:=0.0
6 - val acc:=0.10
# Epoch:=3/10  - train loss:=0.0299 - val loss:=0.0166, train acc:=0.6
0 - val acc:=0.68
# Epoch:=4/10  - train loss:=0.0106 - val loss:=0.0064, train acc:=0.8
8 - val acc:=0.96
# Epoch:=5/10  - train loss:=0.0052 - val loss:=0.0043, train acc:=0.9
4 - val acc:=0.96
# Epoch:=6/10  - train loss:=0.0036 - val loss:=0.0034, train acc:=0.9
4 - val acc:=0.94
# Epoch:=7/10  - train loss:=0.0027 - val loss:=0.0031, train acc:=0.9
6 - val acc:=1.00
# Epoch:=8/10  - train loss:=0.0021 - val loss:=0.0028, train acc:=1.0
0 - val acc:=1.00
# Epoch:=9/10  - train loss:=0.0016 - val loss:=0.0027, train acc:=0.9
8 - val acc:=0.98
# Epoch:=10/10  - train loss:=0.0013 - val loss:=0.0025, train acc:=1.
00 - val acc:=1.00

Total time taken (in seconds):  171.97

Finished training model: mlp_on_gpu_RegL1

**************** Testing ****************
mlp_on_gpu_RegL1 model accuracy = 96.15%
*****************************************
```



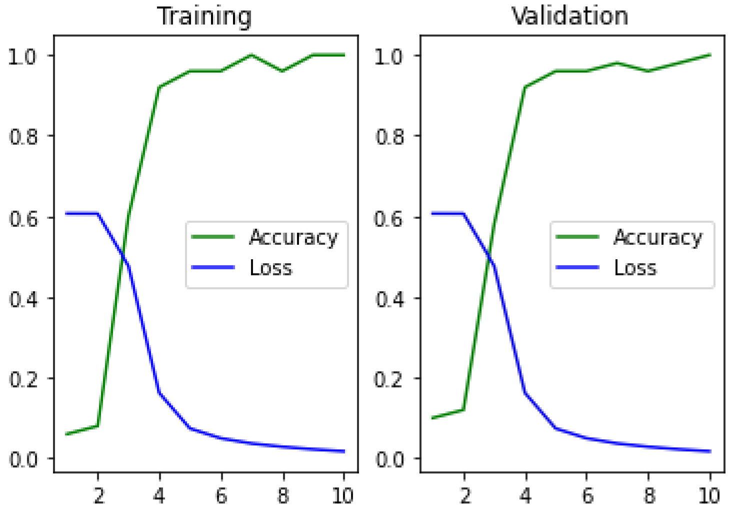
```
Count: 3, j=: 0


**************** Training model: mlp_on_gpu_RegL1 with optimizer: SGD
and seed: 7927 ****************

# Epoch:=1/10  - train loss:=0.0460 - val loss:=0.0460, train acc:=0.0
8 - val acc:=0.10
# Epoch:=2/10  - train loss:=0.0460 - val loss:=0.0456, train acc:=0.1
4 - val acc:=0.18
# Epoch:=3/10  - train loss:=0.0289 - val loss:=0.0136, train acc:=0.7
0 - val acc:=0.76
# Epoch:=4/10  - train loss:=0.0103 - val loss:=0.0067, train acc:=0.9
2 - val acc:=0.96
# Epoch:=5/10  - train loss:=0.0055 - val loss:=0.0044, train acc:=0.9
6 - val acc:=0.96
# Epoch:=6/10  - train loss:=0.0036 - val loss:=0.0037, train acc:=1.0
0 - val acc:=0.98
# Epoch:=7/10  - train loss:=0.0027 - val loss:=0.0033, train acc:=1.0
0 - val acc:=0.98
# Epoch:=8/10  - train loss:=0.0021 - val loss:=0.0027, train acc:=0.9
8 - val acc:=1.00
# Epoch:=9/10  - train loss:=0.0017 - val loss:=0.0026, train acc:=1.0
0 - val acc:=1.00
# Epoch:=10/10  - train loss:=0.0013 - val loss:=0.0028, train acc:=0.
98 - val acc:=1.00

Total time taken (in seconds):  171.85

Finished training model: mlp_on_gpu_RegL1

**************** Testing ****************
mlp_on_gpu_RegL1 model accuracy = 96.11%
*****************************************
```



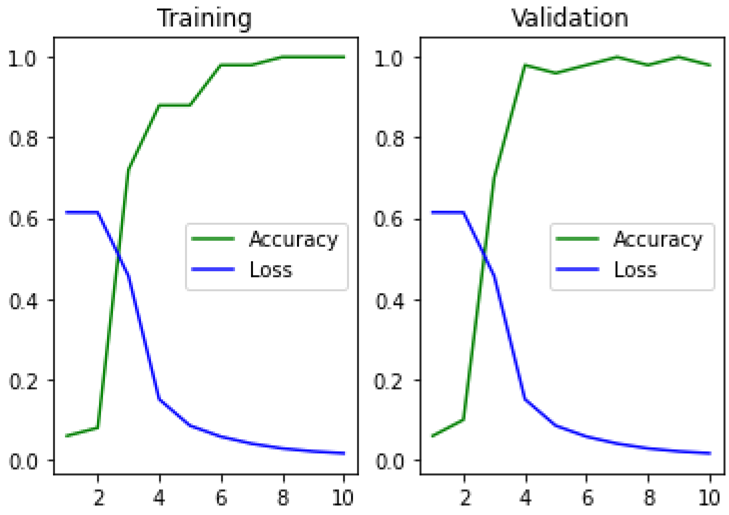
```
Count: 4, j=: 0


**************** Training model: mlp_on_gpu_RegL1 with optimizer: SGD
and seed: 7149 ****************

# Epoch:=1/10  - train loss:=0.0460 - val loss:=0.0460, train acc:=0.0
4 - val acc:=0.10
# Epoch:=2/10  - train loss:=0.0460 - val loss:=0.0460, train acc:=0.0
8 - val acc:=0.12
# Epoch:=3/10  - train loss:=0.0335 - val loss:=0.0168, train acc:=0.7
0 - val acc:=0.68
# Epoch:=4/10  - train loss:=0.0115 - val loss:=0.0068, train acc:=0.9
0 - val acc:=0.96
# Epoch:=5/10  - train loss:=0.0058 - val loss:=0.0047, train acc:=0.9
8 - val acc:=0.98
# Epoch:=6/10  - train loss:=0.0039 - val loss:=0.0037, train acc:=0.9
4 - val acc:=0.94
# Epoch:=7/10  - train loss:=0.0029 - val loss:=0.0031, train acc:=0.9
4 - val acc:=0.98
# Epoch:=8/10  - train loss:=0.0022 - val loss:=0.0029, train acc:=1.0
0 - val acc:=1.00
# Epoch:=9/10  - train loss:=0.0017 - val loss:=0.0028, train acc:=1.0
0 - val acc:=1.00
# Epoch:=10/10  - train loss:=0.0014 - val loss:=0.0025, train acc:=1.
00 - val acc:=0.98

Total time taken (in seconds):  172.63

Finished training model: mlp_on_gpu_RegL1

**************** Testing ****************
mlp_on_gpu_RegL1 model accuracy = 96.41%
*****************************************
```



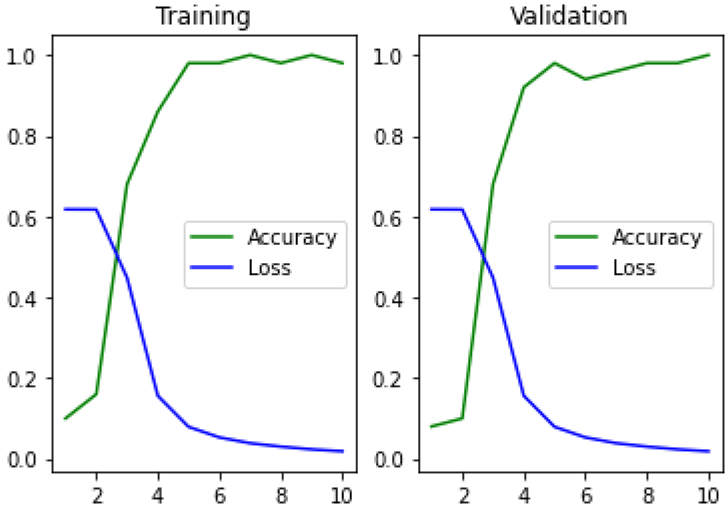
```
Count: 5, j=: 0


**************** Training model: mlp_on_gpu_RegL1 with optimizer: SGD
and seed: 9716 ****************

# Epoch:=1/10  - train loss:=0.0460 - val loss:=0.0460, train acc:=0.0
6 - val acc:=0.10
# Epoch:=2/10  - train loss:=0.0460 - val loss:=0.0460, train acc:=0.0
8 - val acc:=0.12
# Epoch:=3/10  - train loss:=0.0362 - val loss:=0.0202, train acc:=0.6
0 - val acc:=0.58
# Epoch:=4/10  - train loss:=0.0123 - val loss:=0.0066, train acc:=0.9
2 - val acc:=0.92
# Epoch:=5/10  - train loss:=0.0056 - val loss:=0.0045, train acc:=0.9
6 - val acc:=0.96
# Epoch:=6/10  - train loss:=0.0038 - val loss:=0.0035, train acc:=0.9
6 - val acc:=0.96
# Epoch:=7/10  - train loss:=0.0028 - val loss:=0.0031, train acc:=1.0
0 - val acc:=0.98
# Epoch:=8/10  - train loss:=0.0022 - val loss:=0.0028, train acc:=0.9
6 - val acc:=0.96
# Epoch:=9/10  - train loss:=0.0017 - val loss:=0.0025, train acc:=1.0
0 - val acc:=0.98
# Epoch:=10/10  - train loss:=0.0013 - val loss:=0.0026, train acc:=1.
00 - val acc:=1.00

Total time taken (in seconds):  175.22

Finished training model: mlp_on_gpu_RegL1

**************** Testing ****************
mlp_on_gpu_RegL1 model accuracy = 96.39%
*****************************************
```



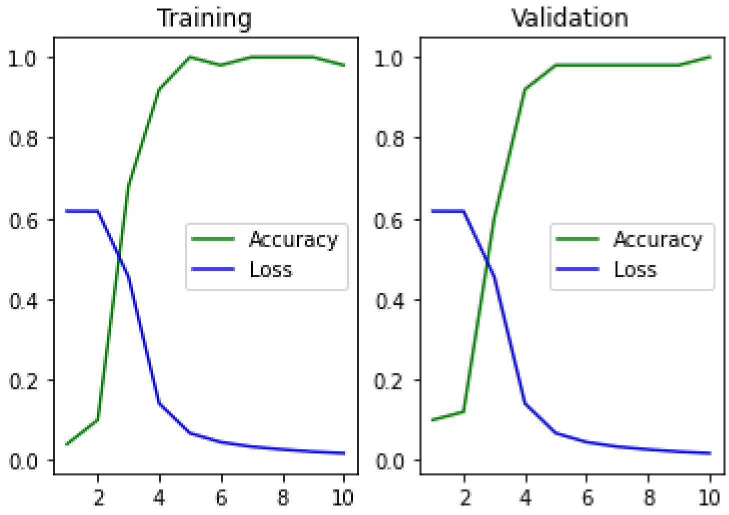
```
Count: 6, j=: 0


**************** Training model: mlp_on_gpu_RegL1 with optimizer: SGD
and seed: 7288 ****************

# Epoch:=1/10  - train loss:=0.0460 - val loss:=0.0460, train acc:=0.0
6 - val acc:=0.06
# Epoch:=2/10  - train loss:=0.0460 - val loss:=0.0460, train acc:=0.0
8 - val acc:=0.10
# Epoch:=3/10  - train loss:=0.0343 - val loss:=0.0156, train acc:=0.7
2 - val acc:=0.70
# Epoch:=4/10  - train loss:=0.0113 - val loss:=0.0072, train acc:=0.8
8 - val acc:=0.98
# Epoch:=5/10  - train loss:=0.0064 - val loss:=0.0051, train acc:=0.8
8 - val acc:=0.96
# Epoch:=6/10  - train loss:=0.0044 - val loss:=0.0041, train acc:=0.9
8 - val acc:=0.98
# Epoch:=7/10  - train loss:=0.0030 - val loss:=0.0031, train acc:=0.9
8 - val acc:=1.00
# Epoch:=8/10  - train loss:=0.0022 - val loss:=0.0029, train acc:=1.0
0 - val acc:=0.98
# Epoch:=9/10  - train loss:=0.0016 - val loss:=0.0025, train acc:=1.0
0 - val acc:=1.00
# Epoch:=10/10  - train loss:=0.0013 - val loss:=0.0023, train acc:=1.
00 - val acc:=0.98

Total time taken (in seconds):  175.90

Finished training model: mlp_on_gpu_RegL1

**************** Testing ****************
mlp_on_gpu_RegL1 model accuracy = 96.40%
*****************************************
```



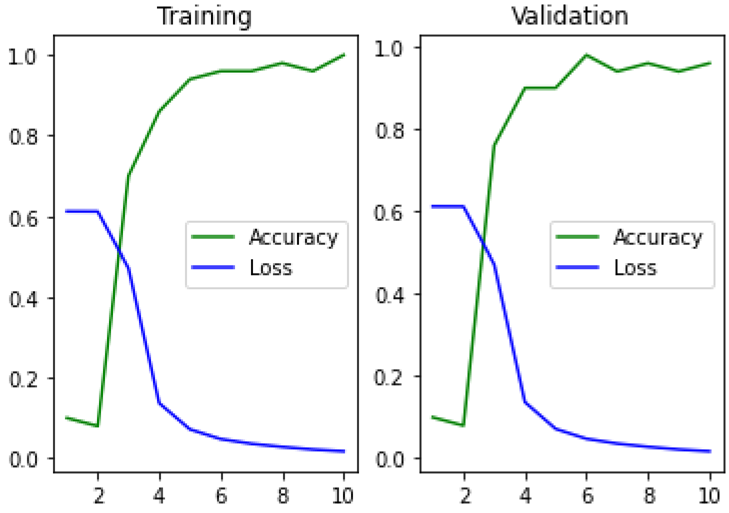
```
Count: 7, j=: 0


**************** Training model: mlp_on_gpu_RegL1 with optimizer: SGD
and seed: 9321 ****************

# Epoch:=1/10  - train loss:=0.0460 - val loss:=0.0460, train acc:=0.1
0 - val acc:=0.08
# Epoch:=2/10  - train loss:=0.0460 - val loss:=0.0458, train acc:=0.1
6 - val acc:=0.10
# Epoch:=3/10  - train loss:=0.0335 - val loss:=0.0188, train acc:=0.6
8 - val acc:=0.68
# Epoch:=4/10  - train loss:=0.0116 - val loss:=0.0070, train acc:=0.8
6 - val acc:=0.92
# Epoch:=5/10  - train loss:=0.0059 - val loss:=0.0046, train acc:=0.9
8 - val acc:=0.98
# Epoch:=6/10  - train loss:=0.0039 - val loss:=0.0038, train acc:=0.9
8 - val acc:=0.94
# Epoch:=7/10  - train loss:=0.0029 - val loss:=0.0035, train acc:=1.0
0 - val acc:=0.96
# Epoch:=8/10  - train loss:=0.0022 - val loss:=0.0030, train acc:=0.9
8 - val acc:=0.98
# Epoch:=9/10  - train loss:=0.0017 - val loss:=0.0029, train acc:=1.0
0 - val acc:=0.98
# Epoch:=10/10  - train loss:=0.0014 - val loss:=0.0028, train acc:=0.
98 - val acc:=1.00

Total time taken (in seconds):  175.09

Finished training model: mlp_on_gpu_RegL1

**************** Testing ****************
mlp_on_gpu_RegL1 model accuracy = 96.45%
*****************************************
```



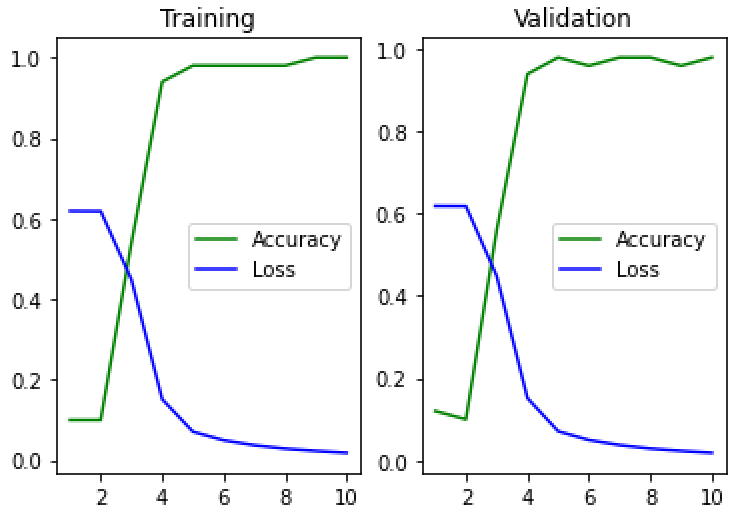
```
Count: 8, j=: 0


**************** Training model: mlp_on_gpu_RegL1 with optimizer: SGD
and seed: 8150 ****************

# Epoch:=1/10  - train loss:=0.0460 - val loss:=0.0460, train acc:=0.0
4 - val acc:=0.10
# Epoch:=2/10  - train loss:=0.0460 - val loss:=0.0460, train acc:=0.1
0 - val acc:=0.12
# Epoch:=3/10  - train loss:=0.0339 - val loss:=0.0164, train acc:=0.6
8 - val acc:=0.60
# Epoch:=4/10  - train loss:=0.0105 - val loss:=0.0062, train acc:=0.9
2 - val acc:=0.92
# Epoch:=5/10  - train loss:=0.0050 - val loss:=0.0040, train acc:=1.0
0 - val acc:=0.98
# Epoch:=6/10  - train loss:=0.0033 - val loss:=0.0032, train acc:=0.9
8 - val acc:=0.98
# Epoch:=7/10  - train loss:=0.0025 - val loss:=0.0027, train acc:=1.0
0 - val acc:=0.98
# Epoch:=8/10  - train loss:=0.0020 - val loss:=0.0025, train acc:=1.0
0 - val acc:=0.98
# Epoch:=9/10  - train loss:=0.0016 - val loss:=0.0024, train acc:=1.0
0 - val acc:=0.98
# Epoch:=10/10  - train loss:=0.0013 - val loss:=0.0022, train acc:=0.
98 - val acc:=1.00

Total time taken (in seconds):  174.48

Finished training model: mlp_on_gpu_RegL1

**************** Testing ****************
mlp_on_gpu_RegL1 model accuracy = 96.50%
*****************************************
```



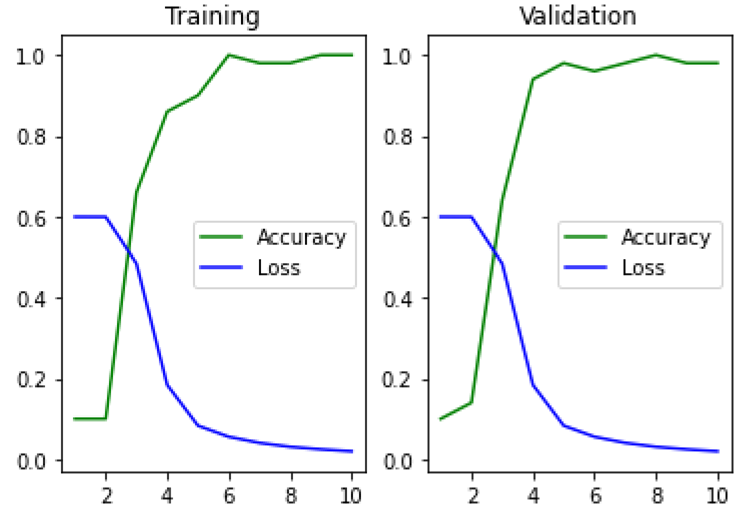
```
Count: 9, j=: 0


**************** Training model: mlp_on_gpu_RegL1 with optimizer: SGD
and seed: 7198 ****************

# Epoch:=1/10  - train loss:=0.0460 - val loss:=0.0460, train acc:=0.1
0 - val acc:=0.10
# Epoch:=2/10  - train loss:=0.0460 - val loss:=0.0460, train acc:=0.0
8 - val acc:=0.08
# Epoch:=3/10  - train loss:=0.0355 - val loss:=0.0163, train acc:=0.7
0 - val acc:=0.76
# Epoch:=4/10  - train loss:=0.0103 - val loss:=0.0066, train acc:=0.8
6 - val acc:=0.90
# Epoch:=5/10  - train loss:=0.0054 - val loss:=0.0043, train acc:=0.9
4 - val acc:=0.90
# Epoch:=6/10  - train loss:=0.0036 - val loss:=0.0032, train acc:=0.9
6 - val acc:=0.98
# Epoch:=7/10  - train loss:=0.0027 - val loss:=0.0027, train acc:=0.9
6 - val acc:=0.94
# Epoch:=8/10  - train loss:=0.0021 - val loss:=0.0025, train acc:=0.9
8 - val acc:=0.96
# Epoch:=9/10  - train loss:=0.0016 - val loss:=0.0024, train acc:=0.9
6 - val acc:=0.94
# Epoch:=10/10  - train loss:=0.0013 - val loss:=0.0024, train acc:=1.
00 - val acc:=0.96

Total time taken (in seconds):  174.64

Finished training model: mlp_on_gpu_RegL1

**************** Testing ****************
mlp_on_gpu_RegL1 model accuracy = 96.38%
*****************************************
```



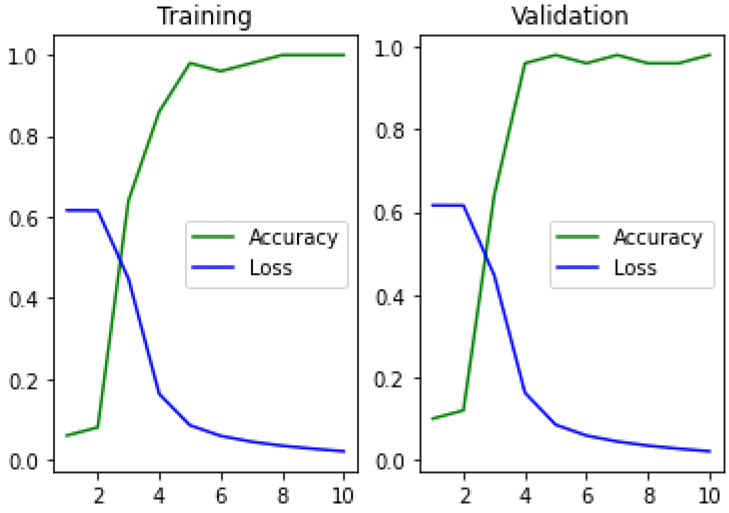
```
Count: 0, j=: 0


**************** Training model: mlp_on_gpu_RegL2 with optimizer: SGD
and seed: 9800 ****************

# Epoch:=1/10  - train loss:=0.0460 - val loss:=0.0460, train acc:=0.1
0 - val acc:=0.12
# Epoch:=2/10  - train loss:=0.0460 - val loss:=0.0459, train acc:=0.1
0 - val acc:=0.10
# Epoch:=3/10  - train loss:=0.0334 - val loss:=0.0187, train acc:=0.5
4 - val acc:=0.56
# Epoch:=4/10  - train loss:=0.0113 - val loss:=0.0062, train acc:=0.9
4 - val acc:=0.94
# Epoch:=5/10  - train loss:=0.0053 - val loss:=0.0045, train acc:=0.9
8 - val acc:=0.98
# Epoch:=6/10  - train loss:=0.0037 - val loss:=0.0034, train acc:=0.9
8 - val acc:=0.96
# Epoch:=7/10  - train loss:=0.0028 - val loss:=0.0029, train acc:=0.9
8 - val acc:=0.98
# Epoch:=8/10  - train loss:=0.0021 - val loss:=0.0026, train acc:=0.9
8 - val acc:=0.98
# Epoch:=9/10  - train loss:=0.0017 - val loss:=0.0027, train acc:=1.0
0 - val acc:=0.96
# Epoch:=10/10  - train loss:=0.0014 - val loss:=0.0025, train acc:=1.
00 - val acc:=0.98

Total time taken (in seconds):  194.03

Finished training model: mlp_on_gpu_RegL2

**************** Testing ****************
mlp_on_gpu_RegL2 model accuracy = 96.62%
*****************************************
```



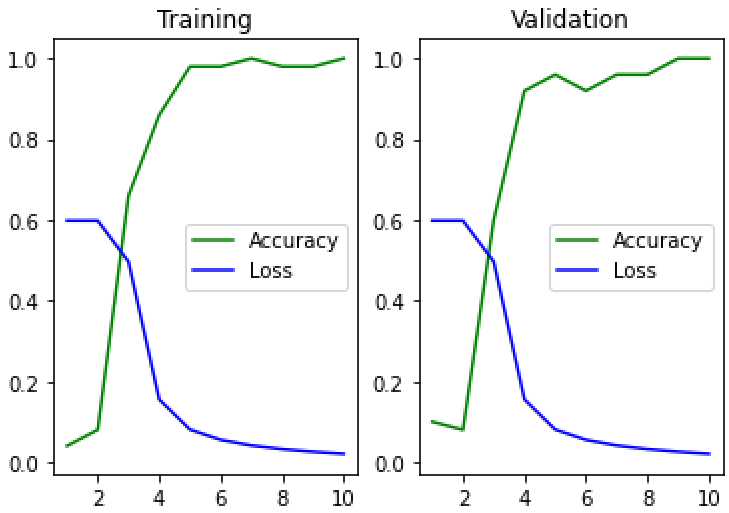
```
Count: 1, j=: 0


**************** Training model: mlp_on_gpu_RegL2 with optimizer: SGD
and seed: 8573 ****************

# Epoch:=1/10  - train loss:=0.0460 - val loss:=0.0460, train acc:=0.1
0 - val acc:=0.10
# Epoch:=2/10  - train loss:=0.0460 - val loss:=0.0460, train acc:=0.1
0 - val acc:=0.14
# Epoch:=3/10  - train loss:=0.0372 - val loss:=0.0195, train acc:=0.6
6 - val acc:=0.64
# Epoch:=4/10  - train loss:=0.0141 - val loss:=0.0072, train acc:=0.8
6 - val acc:=0.94
# Epoch:=5/10  - train loss:=0.0064 - val loss:=0.0053, train acc:=0.9
0 - val acc:=0.98
# Epoch:=6/10  - train loss:=0.0043 - val loss:=0.0036, train acc:=1.0
0 - val acc:=0.96
# Epoch:=7/10  - train loss:=0.0031 - val loss:=0.0029, train acc:=0.9
8 - val acc:=0.98
# Epoch:=8/10  - train loss:=0.0024 - val loss:=0.0026, train acc:=0.9
8 - val acc:=1.00
# Epoch:=9/10  - train loss:=0.0019 - val loss:=0.0025, train acc:=1.0
0 - val acc:=0.98
# Epoch:=10/10  - train loss:=0.0015 - val loss:=0.0024, train acc:=1.
00 - val acc:=0.98

Total time taken (in seconds):  193.45

Finished training model: mlp_on_gpu_RegL2

**************** Testing ****************
mlp_on_gpu_RegL2 model accuracy = 96.26%
*****************************************
```



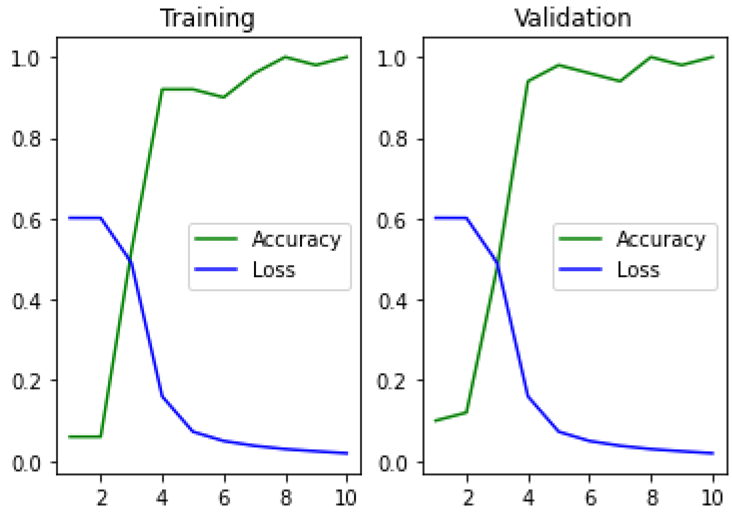
```
Count: 2, j=: 0


**************** Training model: mlp_on_gpu_RegL2 with optimizer: SGD
and seed: 1288 ****************

# Epoch:=1/10  - train loss:=0.0460 - val loss:=0.0460, train acc:=0.0
6 - val acc:=0.10
# Epoch:=2/10  - train loss:=0.0460 - val loss:=0.0459, train acc:=0.0
8 - val acc:=0.12
# Epoch:=3/10  - train loss:=0.0335 - val loss:=0.0173, train acc:=0.6
4 - val acc:=0.64
# Epoch:=4/10  - train loss:=0.0122 - val loss:=0.0075, train acc:=0.8
6 - val acc:=0.96
# Epoch:=5/10  - train loss:=0.0064 - val loss:=0.0050, train acc:=0.9
8 - val acc:=0.98
# Epoch:=6/10  - train loss:=0.0044 - val loss:=0.0043, train acc:=0.9
6 - val acc:=0.96
# Epoch:=7/10  - train loss:=0.0033 - val loss:=0.0035, train acc:=0.9
8 - val acc:=0.98
# Epoch:=8/10  - train loss:=0.0026 - val loss:=0.0032, train acc:=1.0
0 - val acc:=0.96
# Epoch:=9/10  - train loss:=0.0020 - val loss:=0.0029, train acc:=1.0
0 - val acc:=0.96
# Epoch:=10/10  - train loss:=0.0016 - val loss:=0.0026, train acc:=1.
00 - val acc:=0.98

Total time taken (in seconds):  193.56

Finished training model: mlp_on_gpu_RegL2

**************** Testing ****************
mlp_on_gpu_RegL2 model accuracy = 96.45%
*****************************************
```



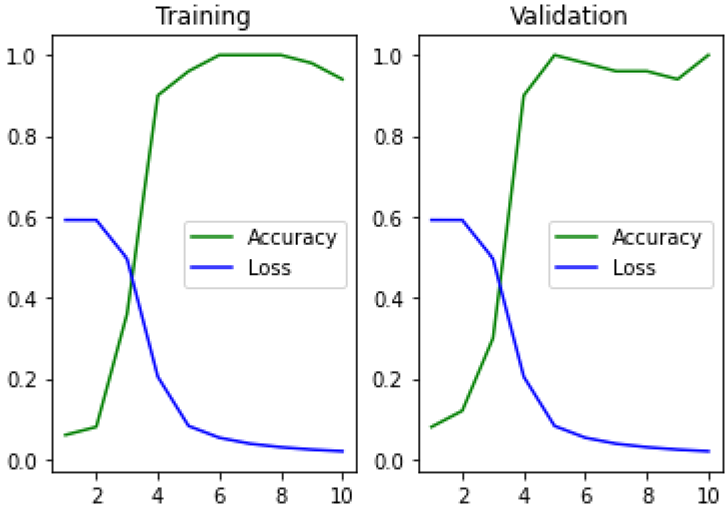
```
Count: 3, j=: 0


**************** Training model: mlp_on_gpu_RegL2 with optimizer: SGD
and seed: 3549 ****************

# Epoch:=1/10  - train loss:=0.0460 - val loss:=0.0460, train acc:=0.0
4 - val acc:=0.10
# Epoch:=2/10  - train loss:=0.0460 - val loss:=0.0460, train acc:=0.0
8 - val acc:=0.08
# Epoch:=3/10  - train loss:=0.0382 - val loss:=0.0196, train acc:=0.6
6 - val acc:=0.60
# Epoch:=4/10  - train loss:=0.0119 - val loss:=0.0074, train acc:=0.8
6 - val acc:=0.92
# Epoch:=5/10  - train loss:=0.0062 - val loss:=0.0049, train acc:=0.9
8 - val acc:=0.96
# Epoch:=6/10  - train loss:=0.0042 - val loss:=0.0037, train acc:=0.9
8 - val acc:=0.92
# Epoch:=7/10  - train loss:=0.0032 - val loss:=0.0032, train acc:=1.0
0 - val acc:=0.96
# Epoch:=8/10  - train loss:=0.0025 - val loss:=0.0027, train acc:=0.9
8 - val acc:=0.96
# Epoch:=9/10  - train loss:=0.0020 - val loss:=0.0025, train acc:=0.9
8 - val acc:=1.00
# Epoch:=10/10  - train loss:=0.0016 - val loss:=0.0024, train acc:=1.
00 - val acc:=1.00

Total time taken (in seconds):  193.93

Finished training model: mlp_on_gpu_RegL2

**************** Testing ****************
mlp_on_gpu_RegL2 model accuracy = 96.44%
*****************************************
```



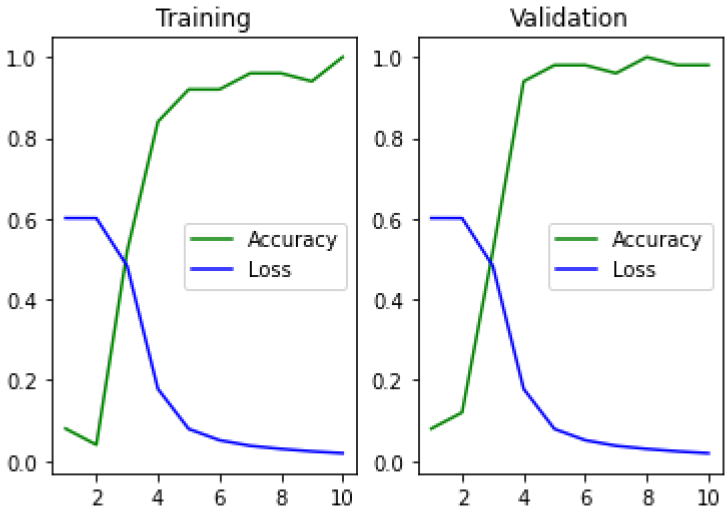
```
Count: 4, j=: 0


**************** Training model: mlp_on_gpu_RegL2 with optimizer: SGD
and seed: 6213 ****************

# Epoch:=1/10  - train loss:=0.0460 - val loss:=0.0460, train acc:=0.0
6 - val acc:=0.10
# Epoch:=2/10  - train loss:=0.0460 - val loss:=0.0460, train acc:=0.0
6 - val acc:=0.12
# Epoch:=3/10  - train loss:=0.0375 - val loss:=0.0206, train acc:=0.5
2 - val acc:=0.48
# Epoch:=4/10  - train loss:=0.0123 - val loss:=0.0065, train acc:=0.9
2 - val acc:=0.94
# Epoch:=5/10  - train loss:=0.0056 - val loss:=0.0047, train acc:=0.9
2 - val acc:=0.98
# Epoch:=6/10  - train loss:=0.0038 - val loss:=0.0034, train acc:=0.9
0 - val acc:=0.96
# Epoch:=7/10  - train loss:=0.0029 - val loss:=0.0029, train acc:=0.9
6 - val acc:=0.94
# Epoch:=8/10  - train loss:=0.0023 - val loss:=0.0026, train acc:=1.0
0 - val acc:=1.00
# Epoch:=9/10  - train loss:=0.0018 - val loss:=0.0028, train acc:=0.9
8 - val acc:=0.98
# Epoch:=10/10  - train loss:=0.0015 - val loss:=0.0024, train acc:=1.
00 - val acc:=1.00

Total time taken (in seconds):  193.40

Finished training model: mlp_on_gpu_RegL2

**************** Testing ****************
mlp_on_gpu_RegL2 model accuracy = 96.64%
*****************************************
```



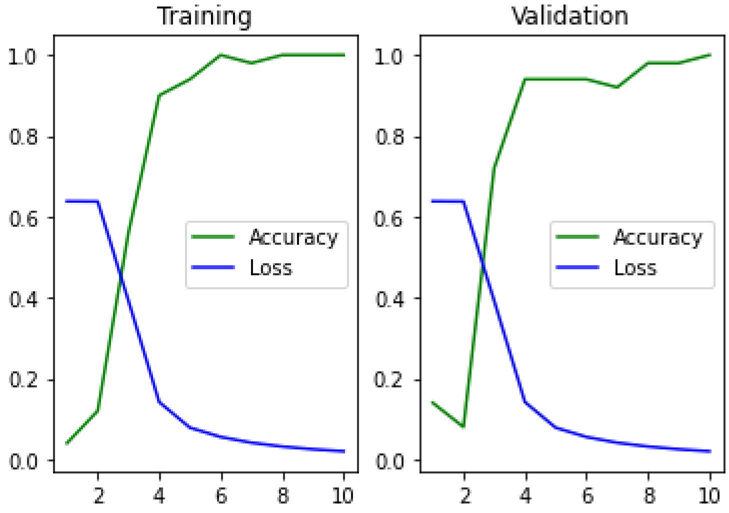
```
Count: 5, j=: 0


**************** Training model: mlp_on_gpu_RegL2 with optimizer: SGD
and seed: 3320 ****************

# Epoch:=1/10  - train loss:=0.0460 - val loss:=0.0460, train acc:=0.0
6 - val acc:=0.08
# Epoch:=2/10  - train loss:=0.0460 - val loss:=0.0460, train acc:=0.0
8 - val acc:=0.12
# Epoch:=3/10  - train loss:=0.0386 - val loss:=0.0277, train acc:=0.3
6 - val acc:=0.30
# Epoch:=4/10  - train loss:=0.0159 - val loss:=0.0079, train acc:=0.9
0 - val acc:=0.90
# Epoch:=5/10  - train loss:=0.0064 - val loss:=0.0048, train acc:=0.9
6 - val acc:=1.00
# Epoch:=6/10  - train loss:=0.0041 - val loss:=0.0037, train acc:=1.0
0 - val acc:=0.98
# Epoch:=7/10  - train loss:=0.0030 - val loss:=0.0031, train acc:=1.0
0 - val acc:=0.96
# Epoch:=8/10  - train loss:=0.0023 - val loss:=0.0028, train acc:=1.0
0 - val acc:=0.96
# Epoch:=9/10  - train loss:=0.0018 - val loss:=0.0028, train acc:=0.9
8 - val acc:=0.94
# Epoch:=10/10  - train loss:=0.0015 - val loss:=0.0029, train acc:=0.
94 - val acc:=1.00

Total time taken (in seconds):  193.74

Finished training model: mlp_on_gpu_RegL2

**************** Testing ****************
mlp_on_gpu_RegL2 model accuracy = 95.83%
*****************************************
```



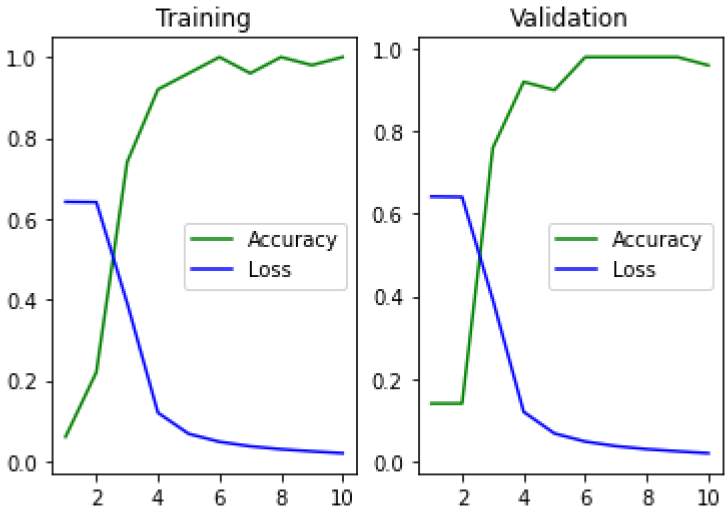
```
Count: 6, j=: 0


**************** Training model: mlp_on_gpu_RegL2 with optimizer: SGD
and seed: 3789 ****************

# Epoch:=1/10  - train loss:=0.0460 - val loss:=0.0460, train acc:=0.0
8 - val acc:=0.08
# Epoch:=2/10  - train loss:=0.0460 - val loss:=0.0460, train acc:=0.0
4 - val acc:=0.12
# Epoch:=3/10  - train loss:=0.0370 - val loss:=0.0199, train acc:=0.5
2 - val acc:=0.52
# Epoch:=4/10  - train loss:=0.0136 - val loss:=0.0079, train acc:=0.8
4 - val acc:=0.94
# Epoch:=5/10  - train loss:=0.0061 - val loss:=0.0045, train acc:=0.9
2 - val acc:=0.98
# Epoch:=6/10  - train loss:=0.0039 - val loss:=0.0034, train acc:=0.9
2 - val acc:=0.98
# Epoch:=7/10  - train loss:=0.0029 - val loss:=0.0028, train acc:=0.9
6 - val acc:=0.96
# Epoch:=8/10  - train loss:=0.0023 - val loss:=0.0026, train acc:=0.9
6 - val acc:=1.00
# Epoch:=9/10  - train loss:=0.0018 - val loss:=0.0026, train acc:=0.9
4 - val acc:=0.98
# Epoch:=10/10  - train loss:=0.0015 - val loss:=0.0025, train acc:=1.
00 - val acc:=0.98

Total time taken (in seconds):  193.00

Finished training model: mlp_on_gpu_RegL2

**************** Testing ****************
mlp_on_gpu_RegL2 model accuracy = 96.36%
*****************************************
```



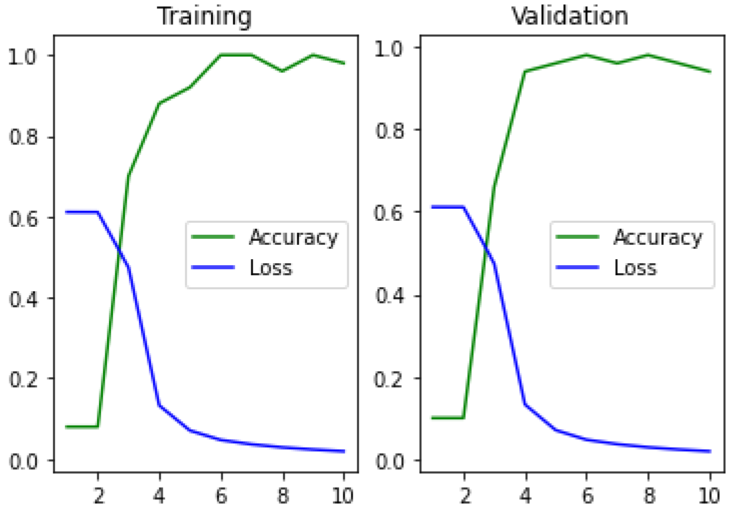
```
Count: 7, j=: 0


**************** Training model: mlp_on_gpu_RegL2 with optimizer: SGD
and seed: 2773 ****************

# Epoch:=1/10  - train loss:=0.0460 - val loss:=0.0460, train acc:=0.0
4 - val acc:=0.14
# Epoch:=2/10  - train loss:=0.0460 - val loss:=0.0459, train acc:=0.1
2 - val acc:=0.08
# Epoch:=3/10  - train loss:=0.0283 - val loss:=0.0151, train acc:=0.5
6 - val acc:=0.72
# Epoch:=4/10  - train loss:=0.0102 - val loss:=0.0065, train acc:=0.9
0 - val acc:=0.94
# Epoch:=5/10  - train loss:=0.0057 - val loss:=0.0046, train acc:=0.9
4 - val acc:=0.94
# Epoch:=6/10  - train loss:=0.0040 - val loss:=0.0037, train acc:=1.0
0 - val acc:=0.94
# Epoch:=7/10  - train loss:=0.0030 - val loss:=0.0038, train acc:=0.9
8 - val acc:=0.92
# Epoch:=8/10  - train loss:=0.0023 - val loss:=0.0029, train acc:=1.0
0 - val acc:=0.98
# Epoch:=9/10  - train loss:=0.0018 - val loss:=0.0026, train acc:=1.0
0 - val acc:=0.98
# Epoch:=10/10  - train loss:=0.0014 - val loss:=0.0025, train acc:=1.
00 - val acc:=1.00

Total time taken (in seconds):  191.77

Finished training model: mlp_on_gpu_RegL2

**************** Testing ****************
mlp_on_gpu_RegL2 model accuracy = 96.40%
*****************************************
```



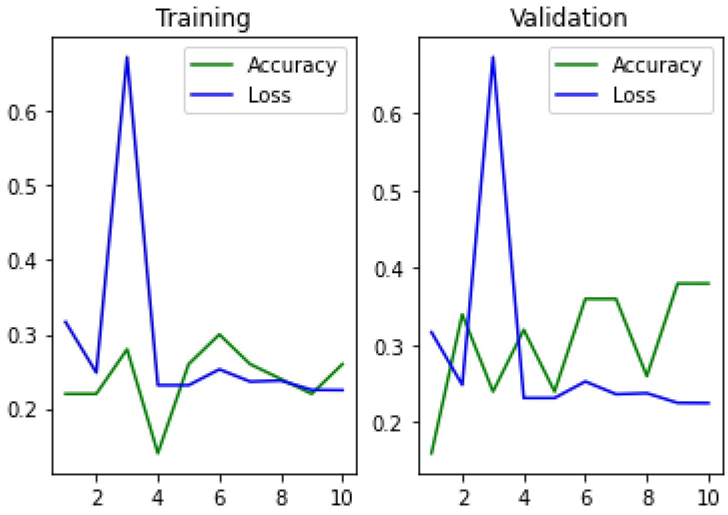
```
Count: 8, j=: 0


**************** Training model: mlp_on_gpu_RegL2 with optimizer: SGD
and seed: 5196 ****************

# Epoch:=1/10  - train loss:=0.0460 - val loss:=0.0460, train acc:=0.0
6 - val acc:=0.14
# Epoch:=2/10  - train loss:=0.0460 - val loss:=0.0453, train acc:=0.2
2 - val acc:=0.14
# Epoch:=3/10  - train loss:=0.0281 - val loss:=0.0121, train acc:=0.7
4 - val acc:=0.76
# Epoch:=4/10  - train loss:=0.0086 - val loss:=0.0058, train acc:=0.9
2 - val acc:=0.92
# Epoch:=5/10  - train loss:=0.0048 - val loss:=0.0037, train acc:=0.9
6 - val acc:=0.90
# Epoch:=6/10  - train loss:=0.0034 - val loss:=0.0030, train acc:=1.0
0 - val acc:=0.98
# Epoch:=7/10  - train loss:=0.0026 - val loss:=0.0025, train acc:=0.9
6 - val acc:=0.98
# Epoch:=8/10  - train loss:=0.0021 - val loss:=0.0025, train acc:=1.0
0 - val acc:=0.98
# Epoch:=9/10  - train loss:=0.0017 - val loss:=0.0024, train acc:=0.9
8 - val acc:=0.98
# Epoch:=10/10  - train loss:=0.0014 - val loss:=0.0023, train acc:=1.
00 - val acc:=0.96

Total time taken (in seconds):  191.76

Finished training model: mlp_on_gpu_RegL2

**************** Testing ****************
mlp_on_gpu_RegL2 model accuracy = 96.42%
*****************************************
```



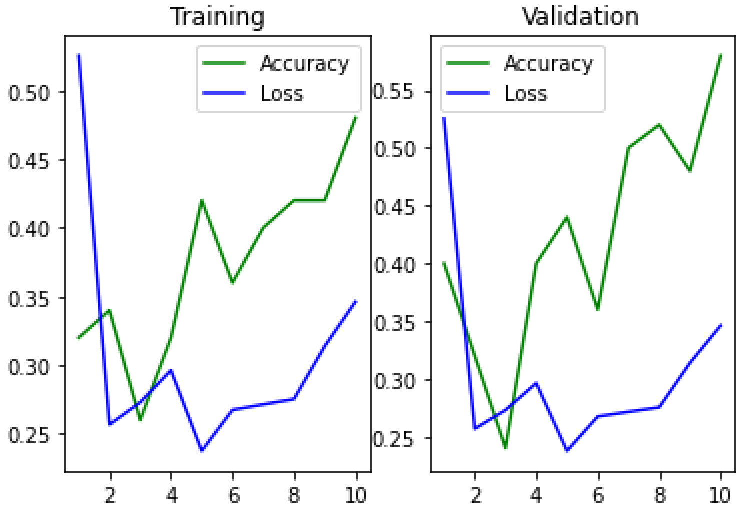
```
Count: 9, j=: 0


**************** Training model: mlp_on_gpu_RegL2 with optimizer: SGD
and seed: 6211 ****************

# Epoch:=1/10  - train loss:=0.0460 - val loss:=0.0460, train acc:=0.0
8 - val acc:=0.10
# Epoch:=2/10  - train loss:=0.0460 - val loss:=0.0460, train acc:=0.0
8 - val acc:=0.10
# Epoch:=3/10  - train loss:=0.0358 - val loss:=0.0152, train acc:=0.7
0 - val acc:=0.66
# Epoch:=4/10  - train loss:=0.0100 - val loss:=0.0064, train acc:=0.8
8 - val acc:=0.94
# Epoch:=5/10  - train loss:=0.0053 - val loss:=0.0044, train acc:=0.9
2 - val acc:=0.96
# Epoch:=6/10  - train loss:=0.0036 - val loss:=0.0032, train acc:=1.0
0 - val acc:=0.98
# Epoch:=7/10  - train loss:=0.0028 - val loss:=0.0029, train acc:=1.0
0 - val acc:=0.96
# Epoch:=8/10  - train loss:=0.0022 - val loss:=0.0025, train acc:=0.9
6 - val acc:=0.98
# Epoch:=9/10  - train loss:=0.0018 - val loss:=0.0025, train acc:=1.0
0 - val acc:=0.96
# Epoch:=10/10  - train loss:=0.0015 - val loss:=0.0027, train acc:=0.
98 - val acc:=0.94

Total time taken (in seconds):  189.31

Finished training model: mlp_on_gpu_RegL2

**************** Testing ****************
mlp_on_gpu_RegL2 model accuracy = 95.70%
*****************************************
```



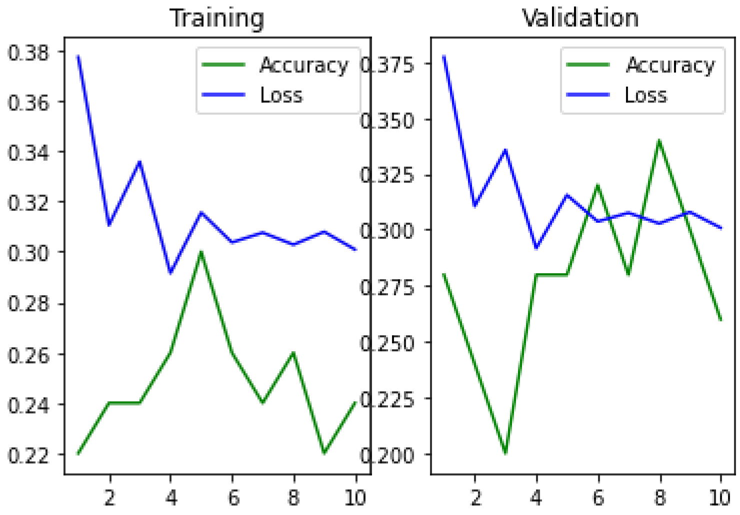
```
Count: 0, j=: 0


**************** Training model: mlp_on_gpu_default with optimizer: Ad
am and seed: 6568 ****************

# Epoch:=1/10  - train loss:=0.0624 - val loss:=0.0384, train acc:=0.2
2 - val acc:=0.16
# Epoch:=2/10  - train loss:=0.0490 - val loss:=0.0409, train acc:=0.2
2 - val acc:=0.34
# Epoch:=3/10  - train loss:=0.1326 - val loss:=0.0386, train acc:=0.2
8 - val acc:=0.24
# Epoch:=4/10  - train loss:=0.0456 - val loss:=0.0910, train acc:=0.1
4 - val acc:=0.32
# Epoch:=5/10  - train loss:=0.0456 - val loss:=0.0392, train acc:=0.2
6 - val acc:=0.24
# Epoch:=6/10  - train loss:=0.0499 - val loss:=0.0389, train acc:=0.3
0 - val acc:=0.36
# Epoch:=7/10  - train loss:=0.0466 - val loss:=0.0369, train acc:=0.2
6 - val acc:=0.36
# Epoch:=8/10  - train loss:=0.0469 - val loss:=0.0385, train acc:=0.2
4 - val acc:=0.26
# Epoch:=9/10  - train loss:=0.0444 - val loss:=0.0370, train acc:=0.2
2 - val acc:=0.38
# Epoch:=10/10  - train loss:=0.0443 - val loss:=0.0496, train acc:=0.
26 - val acc:=0.38

Total time taken (in seconds):  241.50

Finished training model: mlp_on_gpu_default

**************** Testing ****************
mlp_on_gpu_default model accuracy = 35.83%
*****************************************
```



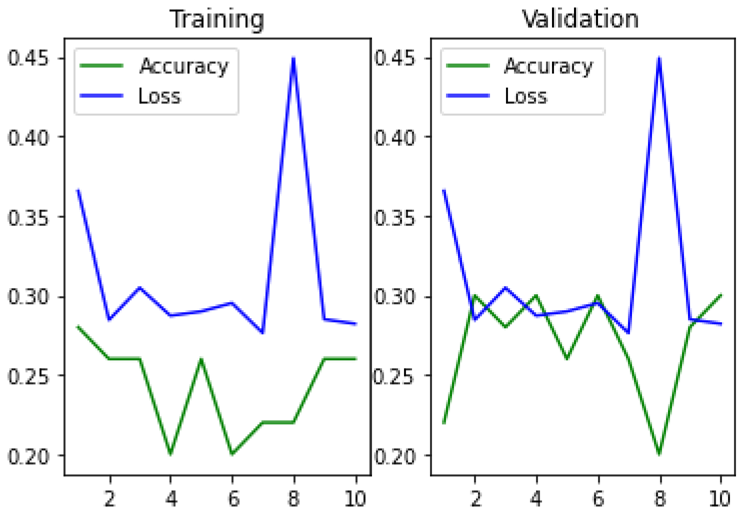
```
Count: 1, j=: 0


**************** Training model: mlp_on_gpu_default with optimizer: Ad
am and seed: 7854 ****************

# Epoch:=1/10  - train loss:=0.0914 - val loss:=0.0514, train acc:=0.3
2 - val acc:=0.40
# Epoch:=2/10  - train loss:=0.0447 - val loss:=0.0358, train acc:=0.3
4 - val acc:=0.32
# Epoch:=3/10  - train loss:=0.0475 - val loss:=0.0444, train acc:=0.2
6 - val acc:=0.24
# Epoch:=4/10  - train loss:=0.0516 - val loss:=0.0364, train acc:=0.3
2 - val acc:=0.40
# Epoch:=5/10  - train loss:=0.0414 - val loss:=0.0337, train acc:=0.4
2 - val acc:=0.44
# Epoch:=6/10  - train loss:=0.0465 - val loss:=0.0456, train acc:=0.3
6 - val acc:=0.36
# Epoch:=7/10  - train loss:=0.0472 - val loss:=0.0420, train acc:=0.4
0 - val acc:=0.50
# Epoch:=8/10  - train loss:=0.0479 - val loss:=0.0452, train acc:=0.4
2 - val acc:=0.52
# Epoch:=9/10  - train loss:=0.0546 - val loss:=0.0847, train acc:=0.4
2 - val acc:=0.48
# Epoch:=10/10  - train loss:=0.0602 - val loss:=0.0497, train acc:=0.
48 - val acc:=0.58

Total time taken (in seconds):  240.89

Finished training model: mlp_on_gpu_default

**************** Testing ****************
mlp_on_gpu_default model accuracy = 48.04%
*****************************************
```



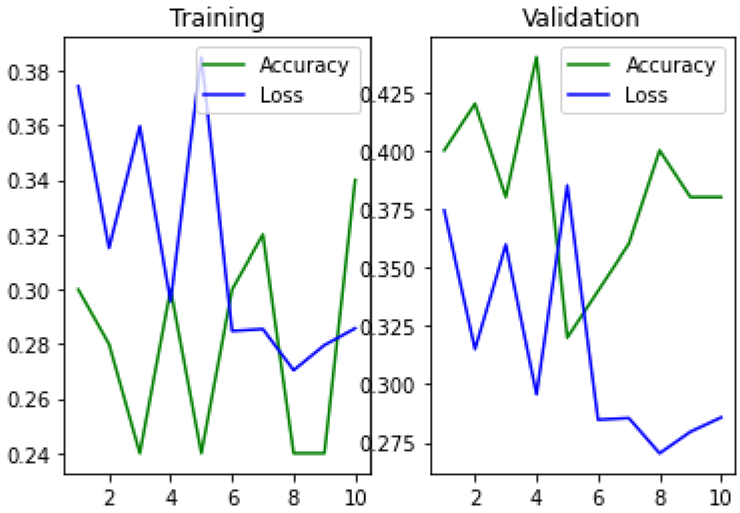
```
Count: 2, j=: 0


**************** Training model: mlp_on_gpu_default with optimizer: Ad
am and seed: 4061 ****************

# Epoch:=1/10  - train loss:=0.0520 - val loss:=0.0412, train acc:=0.2
2 - val acc:=0.28
# Epoch:=2/10  - train loss:=0.0428 - val loss:=0.0389, train acc:=0.2
4 - val acc:=0.24
# Epoch:=3/10  - train loss:=0.0463 - val loss:=0.0386, train acc:=0.2
4 - val acc:=0.20
# Epoch:=4/10  - train loss:=0.0402 - val loss:=0.0383, train acc:=0.2
6 - val acc:=0.28
# Epoch:=5/10  - train loss:=0.0435 - val loss:=0.0480, train acc:=0.3
0 - val acc:=0.28
# Epoch:=6/10  - train loss:=0.0418 - val loss:=0.0394, train acc:=0.2
6 - val acc:=0.32
# Epoch:=7/10  - train loss:=0.0424 - val loss:=0.0428, train acc:=0.2
4 - val acc:=0.28
# Epoch:=8/10  - train loss:=0.0417 - val loss:=0.0390, train acc:=0.2
6 - val acc:=0.34
# Epoch:=9/10  - train loss:=0.0424 - val loss:=0.0400, train acc:=0.2
2 - val acc:=0.30
# Epoch:=10/10  - train loss:=0.0414 - val loss:=0.0393, train acc:=0.
24 - val acc:=0.26

Total time taken (in seconds):  241.28

Finished training model: mlp_on_gpu_default

**************** Testing ****************
mlp_on_gpu_default model accuracy = 28.55%
*****************************************
```



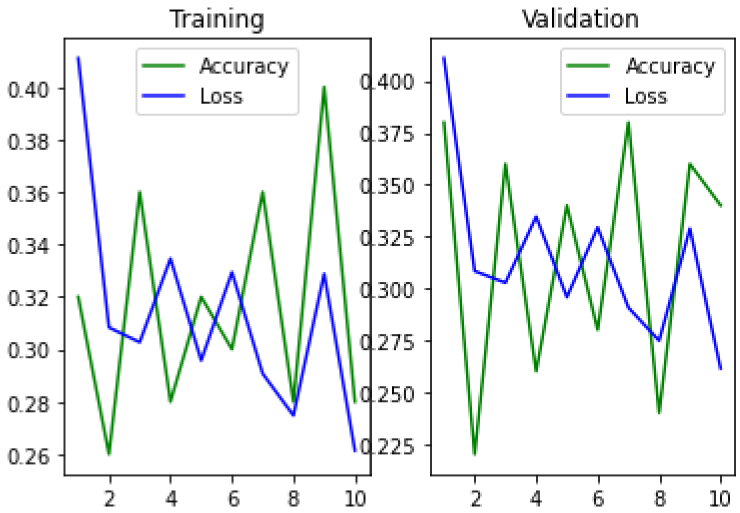
```
Count: 3, j=: 0


**************** Training model: mlp_on_gpu_default with optimizer: Ad
am and seed: 9321 ****************

# Epoch:=1/10  - train loss:=0.0526 - val loss:=0.0387, train acc:=0.2
8 - val acc:=0.22
# Epoch:=2/10  - train loss:=0.0410 - val loss:=0.0397, train acc:=0.2
6 - val acc:=0.30
# Epoch:=3/10  - train loss:=0.0439 - val loss:=0.0467, train acc:=0.2
6 - val acc:=0.28
# Epoch:=4/10  - train loss:=0.0414 - val loss:=0.0369, train acc:=0.2
0 - val acc:=0.30
# Epoch:=5/10  - train loss:=0.0417 - val loss:=0.0378, train acc:=0.2
6 - val acc:=0.26
# Epoch:=6/10  - train loss:=0.0425 - val loss:=0.0381, train acc:=0.2
0 - val acc:=0.30
# Epoch:=7/10  - train loss:=0.0398 - val loss:=0.0385, train acc:=0.2
2 - val acc:=0.26
# Epoch:=8/10  - train loss:=0.0647 - val loss:=0.0405, train acc:=0.2
2 - val acc:=0.20
# Epoch:=9/10  - train loss:=0.0410 - val loss:=0.0464, train acc:=0.2
6 - val acc:=0.28
# Epoch:=10/10  - train loss:=0.0406 - val loss:=0.0385, train acc:=0.
26 - val acc:=0.30

Total time taken (in seconds):  239.52

Finished training model: mlp_on_gpu_default

**************** Testing ****************
mlp_on_gpu_default model accuracy = 30.22%
*****************************************
```



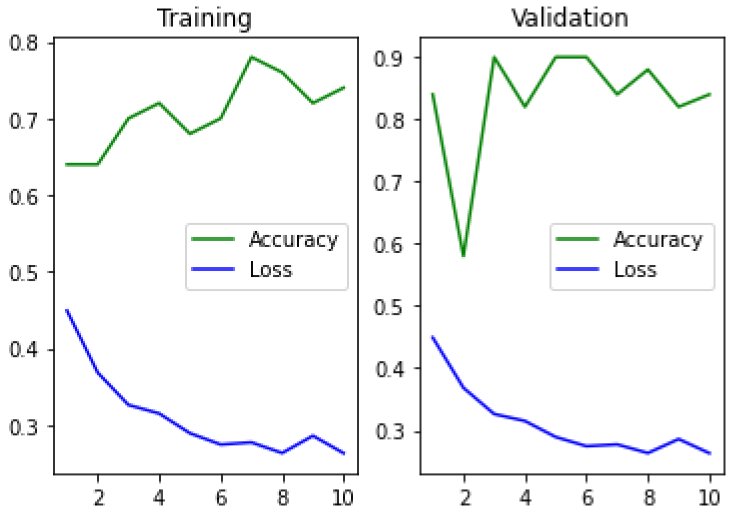
```
Count: 4, j=: 0


**************** Training model: mlp_on_gpu_default with optimizer: Ad
am and seed: 2481 ****************

# Epoch:=1/10  - train loss:=0.0622 - val loss:=0.0438, train acc:=0.3
0 - val acc:=0.40
# Epoch:=2/10  - train loss:=0.0523 - val loss:=0.0380, train acc:=0.2
8 - val acc:=0.42
# Epoch:=3/10  - train loss:=0.0598 - val loss:=0.0535, train acc:=0.2
4 - val acc:=0.38
# Epoch:=4/10  - train loss:=0.0491 - val loss:=0.0412, train acc:=0.3
0 - val acc:=0.44
# Epoch:=5/10  - train loss:=0.0640 - val loss:=0.0411, train acc:=0.2
4 - val acc:=0.32
# Epoch:=6/10  - train loss:=0.0473 - val loss:=0.0447, train acc:=0.3
0 - val acc:=0.34
# Epoch:=7/10  - train loss:=0.0474 - val loss:=0.0364, train acc:=0.3
2 - val acc:=0.36
# Epoch:=8/10  - train loss:=0.0449 - val loss:=0.0406, train acc:=0.2
4 - val acc:=0.40
# Epoch:=9/10  - train loss:=0.0465 - val loss:=0.0724, train acc:=0.2
4 - val acc:=0.38
# Epoch:=10/10  - train loss:=0.0475 - val loss:=0.0378, train acc:=0.
34 - val acc:=0.38

Total time taken (in seconds):  240.76

Finished training model: mlp_on_gpu_default

**************** Testing ****************
mlp_on_gpu_default model accuracy = 34.13%
*****************************************
```



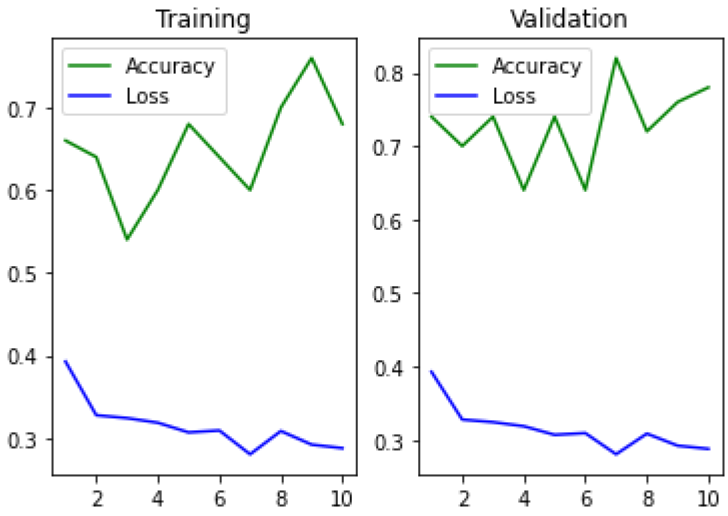
```
Count: 5, j=: 0


**************** Training model: mlp_on_gpu_default with optimizer: Ad
am and seed: 5305 ****************

# Epoch:=1/10  - train loss:=0.0594 - val loss:=0.0649, train acc:=0.3
2 - val acc:=0.38
# Epoch:=2/10  - train loss:=0.0446 - val loss:=0.0385, train acc:=0.2
6 - val acc:=0.22
# Epoch:=3/10  - train loss:=0.0438 - val loss:=0.0515, train acc:=0.3
6 - val acc:=0.36
# Epoch:=4/10  - train loss:=0.0484 - val loss:=0.0394, train acc:=0.2
8 - val acc:=0.26
# Epoch:=5/10  - train loss:=0.0428 - val loss:=0.0504, train acc:=0.3
2 - val acc:=0.34
# Epoch:=6/10  - train loss:=0.0476 - val loss:=0.0376, train acc:=0.3
0 - val acc:=0.28
# Epoch:=7/10  - train loss:=0.0420 - val loss:=0.0430, train acc:=0.3
6 - val acc:=0.38
# Epoch:=8/10  - train loss:=0.0397 - val loss:=0.0362, train acc:=0.2
8 - val acc:=0.24
# Epoch:=9/10  - train loss:=0.0476 - val loss:=0.0355, train acc:=0.4
0 - val acc:=0.36
# Epoch:=10/10  - train loss:=0.0378 - val loss:=0.0363, train acc:=0.
28 - val acc:=0.34

Total time taken (in seconds):  240.34

Finished training model: mlp_on_gpu_default

**************** Testing ****************
mlp_on_gpu_default model accuracy = 33.33%
*****************************************
```



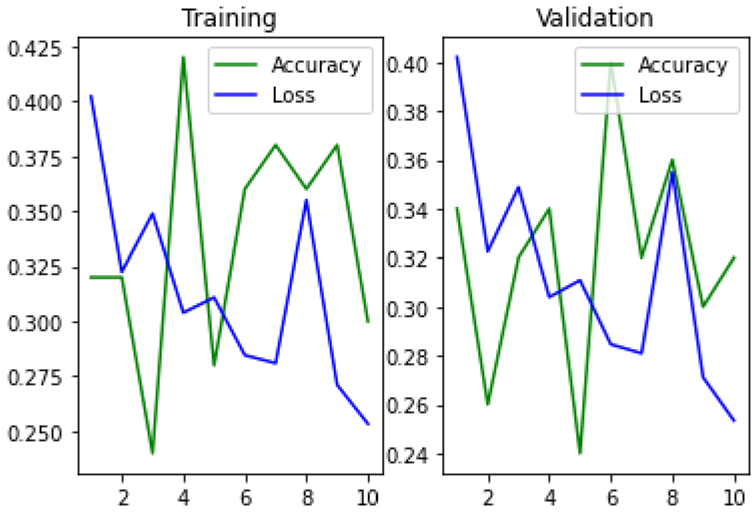
```
Count: 6, j=: 0


**************** Training model: mlp_on_gpu_default with optimizer: Ad
am and seed: 1665 ****************

# Epoch:=1/10  - train loss:=0.1287 - val loss:=0.0969, train acc:=0.6
4 - val acc:=0.84
# Epoch:=2/10  - train loss:=0.1056 - val loss:=0.0996, train acc:=0.6
4 - val acc:=0.58
# Epoch:=3/10  - train loss:=0.0935 - val loss:=0.0832, train acc:=0.7
0 - val acc:=0.90
# Epoch:=4/10  - train loss:=0.0903 - val loss:=0.0772, train acc:=0.7
2 - val acc:=0.82
# Epoch:=5/10  - train loss:=0.0829 - val loss:=0.0793, train acc:=0.6
8 - val acc:=0.90
# Epoch:=6/10  - train loss:=0.0787 - val loss:=0.0735, train acc:=0.7
0 - val acc:=0.90
# Epoch:=7/10  - train loss:=0.0795 - val loss:=0.0753, train acc:=0.7
8 - val acc:=0.84
# Epoch:=8/10  - train loss:=0.0755 - val loss:=0.0662, train acc:=0.7
6 - val acc:=0.88
# Epoch:=9/10  - train loss:=0.0820 - val loss:=0.0796, train acc:=0.7
2 - val acc:=0.82
# Epoch:=10/10  - train loss:=0.0754 - val loss:=0.0809, train acc:=0.
74 - val acc:=0.84

Total time taken (in seconds):  241.49

Finished training model: mlp_on_gpu_default

**************** Testing ****************
mlp_on_gpu_default model accuracy = 73.98%
*****************************************
```



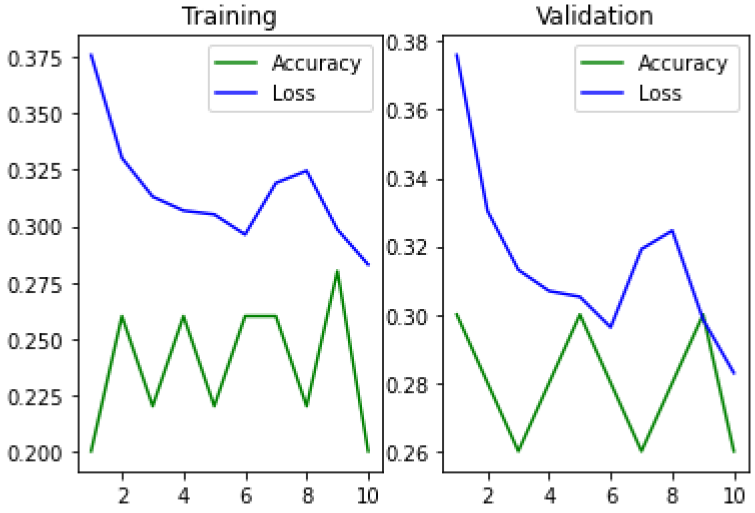
```
Count: 7, j=: 0


**************** Training model: mlp_on_gpu_default with optimizer: Ad
am and seed: 4308 ****************

# Epoch:=1/10  - train loss:=0.1142 - val loss:=0.0896, train acc:=0.6
6 - val acc:=0.74
# Epoch:=2/10  - train loss:=0.0952 - val loss:=0.0853, train acc:=0.6
4 - val acc:=0.70
# Epoch:=3/10  - train loss:=0.0942 - val loss:=0.0988, train acc:=0.5
4 - val acc:=0.74
# Epoch:=4/10  - train loss:=0.0926 - val loss:=0.0956, train acc:=0.6
0 - val acc:=0.64
# Epoch:=5/10  - train loss:=0.0892 - val loss:=0.0801, train acc:=0.6
8 - val acc:=0.74
# Epoch:=6/10  - train loss:=0.0899 - val loss:=0.0867, train acc:=0.6
4 - val acc:=0.64
# Epoch:=7/10  - train loss:=0.0815 - val loss:=0.0831, train acc:=0.6
0 - val acc:=0.82
# Epoch:=8/10  - train loss:=0.0897 - val loss:=0.0877, train acc:=0.7
0 - val acc:=0.72
# Epoch:=9/10  - train loss:=0.0849 - val loss:=0.0699, train acc:=0.7
6 - val acc:=0.76
# Epoch:=10/10  - train loss:=0.0836 - val loss:=0.0801, train acc:=0.
68 - val acc:=0.78

Total time taken (in seconds):  240.95

Finished training model: mlp_on_gpu_default

**************** Testing ****************
mlp_on_gpu_default model accuracy = 75.13%
*****************************************
```



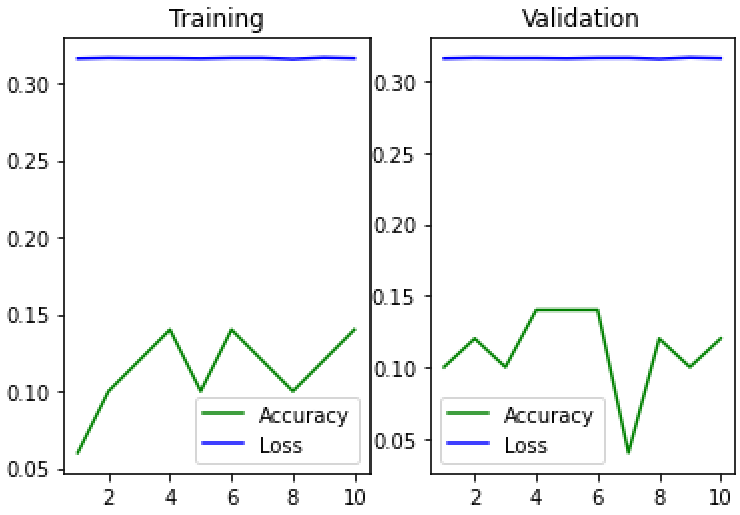
```
Count: 8, j=: 0


**************** Training model: mlp_on_gpu_default with optimizer: Ad
am and seed: 2905 ****************

# Epoch:=1/10  - train loss:=0.0605 - val loss:=0.0400, train acc:=0.3
2 - val acc:=0.34
# Epoch:=2/10  - train loss:=0.0486 - val loss:=0.0356, train acc:=0.3
2 - val acc:=0.26
# Epoch:=3/10  - train loss:=0.0525 - val loss:=0.0373, train acc:=0.2
4 - val acc:=0.32
# Epoch:=4/10  - train loss:=0.0458 - val loss:=0.0367, train acc:=0.4
2 - val acc:=0.34
# Epoch:=5/10  - train loss:=0.0468 - val loss:=0.1222, train acc:=0.2
8 - val acc:=0.24
# Epoch:=6/10  - train loss:=0.0429 - val loss:=0.0491, train acc:=0.3
6 - val acc:=0.40
# Epoch:=7/10  - train loss:=0.0423 - val loss:=0.0371, train acc:=0.3
8 - val acc:=0.32
# Epoch:=8/10  - train loss:=0.0534 - val loss:=0.0385, train acc:=0.3
6 - val acc:=0.36
# Epoch:=9/10  - train loss:=0.0408 - val loss:=0.0344, train acc:=0.3
8 - val acc:=0.30
# Epoch:=10/10  - train loss:=0.0382 - val loss:=0.0366, train acc:=0.
30 - val acc:=0.32

Total time taken (in seconds):  240.37

Finished training model: mlp_on_gpu_default

**************** Testing ****************
mlp_on_gpu_default model accuracy = 36.13%
*****************************************
```



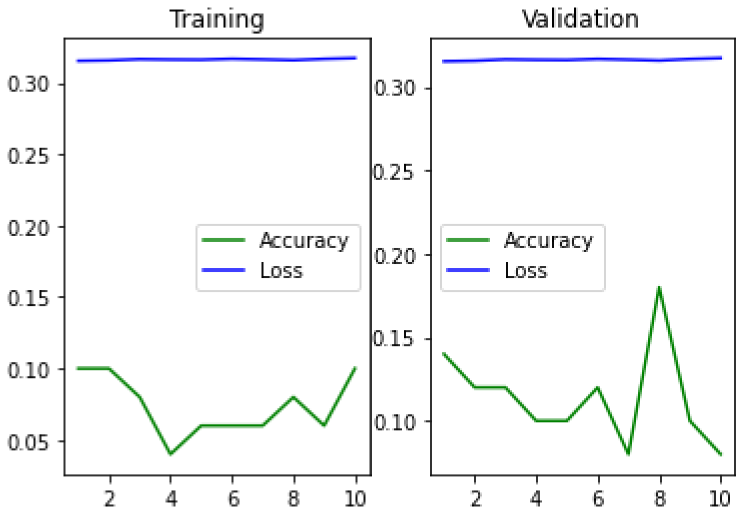
```
Count: 9, j=: 0


**************** Training model: mlp_on_gpu_default with optimizer: Ad
am and seed: 3652 ****************

# Epoch:=1/10  - train loss:=0.0523 - val loss:=0.0393, train acc:=0.2
0 - val acc:=0.30
# Epoch:=2/10  - train loss:=0.0460 - val loss:=0.0402, train acc:=0.2
6 - val acc:=0.28
# Epoch:=3/10  - train loss:=0.0436 - val loss:=0.0414, train acc:=0.2
2 - val acc:=0.26
# Epoch:=4/10  - train loss:=0.0427 - val loss:=0.0411, train acc:=0.2
6 - val acc:=0.28
# Epoch:=5/10  - train loss:=0.0425 - val loss:=0.0424, train acc:=0.2
2 - val acc:=0.30
# Epoch:=6/10  - train loss:=0.0412 - val loss:=0.0377, train acc:=0.2
6 - val acc:=0.28
# Epoch:=7/10  - train loss:=0.0444 - val loss:=0.0418, train acc:=0.2
6 - val acc:=0.26
# Epoch:=8/10  - train loss:=0.0452 - val loss:=0.0395, train acc:=0.2
2 - val acc:=0.28
# Epoch:=9/10  - train loss:=0.0415 - val loss:=0.0373, train acc:=0.2
8 - val acc:=0.30
# Epoch:=10/10  - train loss:=0.0394 - val loss:=0.0375, train acc:=0.
20 - val acc:=0.26

Total time taken (in seconds):  241.24

Finished training model: mlp_on_gpu_default

**************** Testing ****************
mlp_on_gpu_default model accuracy = 27.69%
*****************************************
```



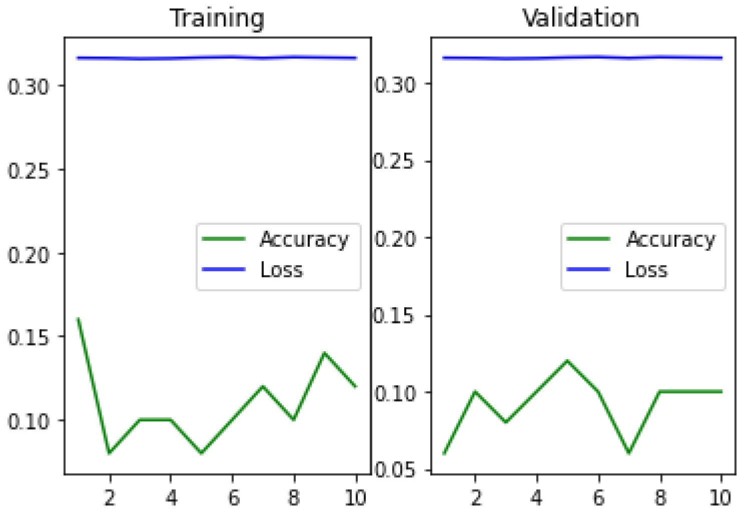
```
Count: 0, j=: 0


**************** Training model: mlp_on_gpu_RegL1 with optimizer: Adam
and seed: 9523 ****************

# Epoch:=1/10  - train loss:=0.1677 - val loss:=0.0462, train acc:=0.0
6 - val acc:=0.10
# Epoch:=2/10  - train loss:=0.1679 - val loss:=0.0461, train acc:=0.1
0 - val acc:=0.12
# Epoch:=3/10  - train loss:=0.1678 - val loss:=0.0461, train acc:=0.1
2 - val acc:=0.10
# Epoch:=4/10  - train loss:=0.1678 - val loss:=0.0461, train acc:=0.1
4 - val acc:=0.14
# Epoch:=5/10  - train loss:=0.1677 - val loss:=0.0462, train acc:=0.1
0 - val acc:=0.14
# Epoch:=6/10  - train loss:=0.1679 - val loss:=0.0461, train acc:=0.1
4 - val acc:=0.14
# Epoch:=7/10  - train loss:=0.1679 - val loss:=0.0461, train acc:=0.1
2 - val acc:=0.04
# Epoch:=8/10  - train loss:=0.1675 - val loss:=0.0461, train acc:=0.1
0 - val acc:=0.12
# Epoch:=9/10  - train loss:=0.1681 - val loss:=0.0461, train acc:=0.1
2 - val acc:=0.10
# Epoch:=10/10  - train loss:=0.1678 - val loss:=0.0462, train acc:=0.
14 - val acc:=0.12

Total time taken (in seconds):  269.34

Finished training model: mlp_on_gpu_RegL1

**************** Testing ****************
mlp_on_gpu_RegL1 model accuracy = 11.35%
*****************************************
```



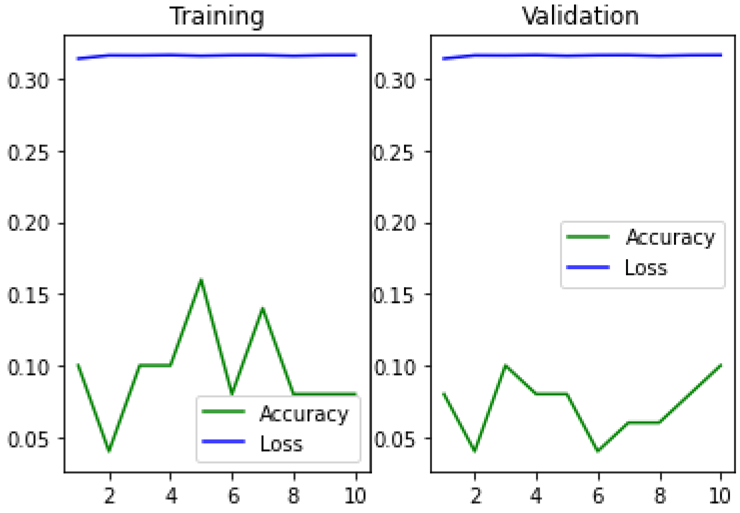
```
Count: 1, j=: 0


**************** Training model: mlp_on_gpu_RegL1 with optimizer: Adam
and seed: 3791 ****************

# Epoch:=1/10  - train loss:=0.1654 - val loss:=0.0461, train acc:=0.1
0 - val acc:=0.14
# Epoch:=2/10  - train loss:=0.1655 - val loss:=0.0461, train acc:=0.1
0 - val acc:=0.12
# Epoch:=3/10  - train loss:=0.1660 - val loss:=0.0461, train acc:=0.0
8 - val acc:=0.12
# Epoch:=4/10  - train loss:=0.1659 - val loss:=0.0461, train acc:=0.0
4 - val acc:=0.10
# Epoch:=5/10  - train loss:=0.1658 - val loss:=0.0463, train acc:=0.0
6 - val acc:=0.10
# Epoch:=6/10  - train loss:=0.1661 - val loss:=0.0461, train acc:=0.0
6 - val acc:=0.12
# Epoch:=7/10  - train loss:=0.1659 - val loss:=0.0462, train acc:=0.0
6 - val acc:=0.08
# Epoch:=8/10  - train loss:=0.1656 - val loss:=0.0462, train acc:=0.0
8 - val acc:=0.18
# Epoch:=9/10  - train loss:=0.1661 - val loss:=0.0461, train acc:=0.0
6 - val acc:=0.10
# Epoch:=10/10  - train loss:=0.1664 - val loss:=0.0461, train acc:=0.
10 - val acc:=0.08

Total time taken (in seconds):  269.20

Finished training model: mlp_on_gpu_RegL1

**************** Testing ****************
mlp_on_gpu_RegL1 model accuracy = 8.92%
*****************************************
```



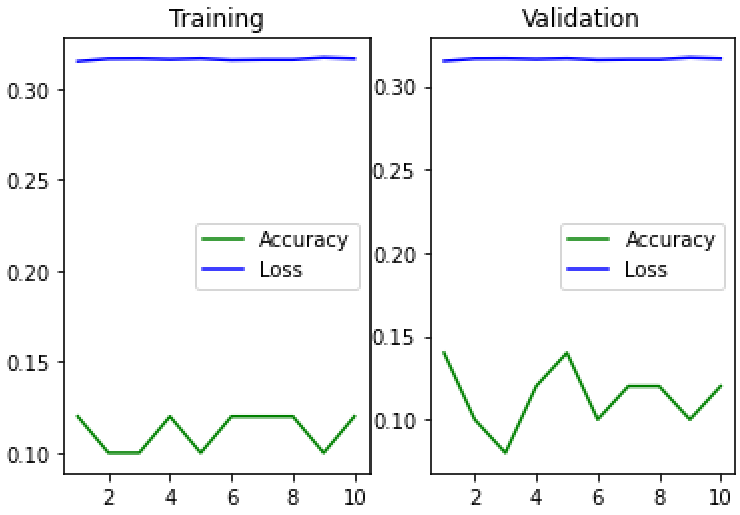
```
Count: 2, j=: 0


**************** Training model: mlp_on_gpu_RegL1 with optimizer: Adam
and seed: 3393 ****************

# Epoch:=1/10  - train loss:=0.1679 - val loss:=0.0461, train acc:=0.1
6 - val acc:=0.06
# Epoch:=2/10  - train loss:=0.1678 - val loss:=0.0462, train acc:=0.0
8 - val acc:=0.10
# Epoch:=3/10  - train loss:=0.1676 - val loss:=0.0462, train acc:=0.1
0 - val acc:=0.08
# Epoch:=4/10  - train loss:=0.1677 - val loss:=0.0461, train acc:=0.1
0 - val acc:=0.10
# Epoch:=5/10  - train loss:=0.1680 - val loss:=0.0461, train acc:=0.0
8 - val acc:=0.12
# Epoch:=6/10  - train loss:=0.1682 - val loss:=0.0461, train acc:=0.1
0 - val acc:=0.10
# Epoch:=7/10  - train loss:=0.1678 - val loss:=0.0461, train acc:=0.1
2 - val acc:=0.06
# Epoch:=8/10  - train loss:=0.1681 - val loss:=0.0461, train acc:=0.1
0 - val acc:=0.10
# Epoch:=9/10  - train loss:=0.1680 - val loss:=0.0461, train acc:=0.1
4 - val acc:=0.10
# Epoch:=10/10  - train loss:=0.1679 - val loss:=0.0461, train acc:=0.
12 - val acc:=0.10

Total time taken (in seconds):  269.56

Finished training model: mlp_on_gpu_RegL1

**************** Testing ****************
mlp_on_gpu_RegL1 model accuracy = 10.28%
*****************************************
```



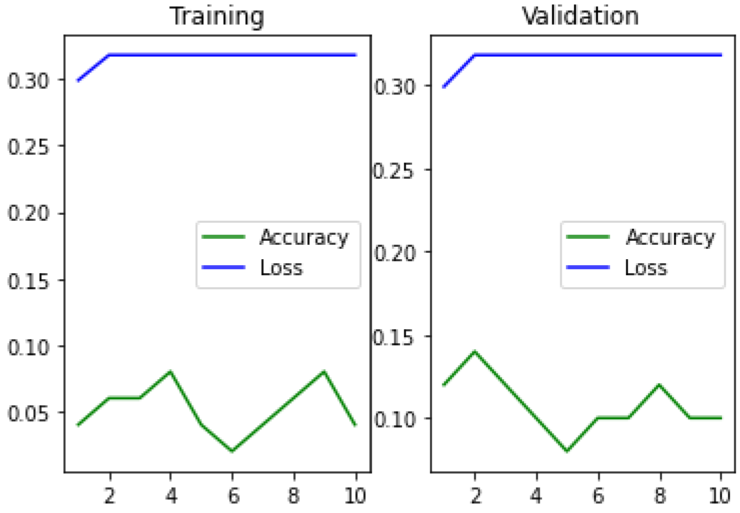
```
Count: 3, j=: 0


**************** Training model: mlp_on_gpu_RegL1 with optimizer: Adam
and seed: 5118 ****************

# Epoch:=1/10  - train loss:=0.1663 - val loss:=0.2872, train acc:=0.1
0 - val acc:=0.08
# Epoch:=2/10  - train loss:=0.1675 - val loss:=0.2872, train acc:=0.0
4 - val acc:=0.04
# Epoch:=3/10  - train loss:=0.1675 - val loss:=0.2872, train acc:=0.1
0 - val acc:=0.10
# Epoch:=4/10  - train loss:=0.1676 - val loss:=0.2872, train acc:=0.1
0 - val acc:=0.08
# Epoch:=5/10  - train loss:=0.1673 - val loss:=0.2872, train acc:=0.1
6 - val acc:=0.08
# Epoch:=6/10  - train loss:=0.1676 - val loss:=0.2872, train acc:=0.0
8 - val acc:=0.04
# Epoch:=7/10  - train loss:=0.1676 - val loss:=0.2872, train acc:=0.1
4 - val acc:=0.06
# Epoch:=8/10  - train loss:=0.1673 - val loss:=0.2872, train acc:=0.0
8 - val acc:=0.06
# Epoch:=9/10  - train loss:=0.1676 - val loss:=0.2872, train acc:=0.0
8 - val acc:=0.08
# Epoch:=10/10  - train loss:=0.1676 - val loss:=0.2872, train acc:=0.
08 - val acc:=0.10

Total time taken (in seconds):  269.51

Finished training model: mlp_on_gpu_RegL1

**************** Testing ****************
mlp_on_gpu_RegL1 model accuracy = 10.28%
*****************************************
```



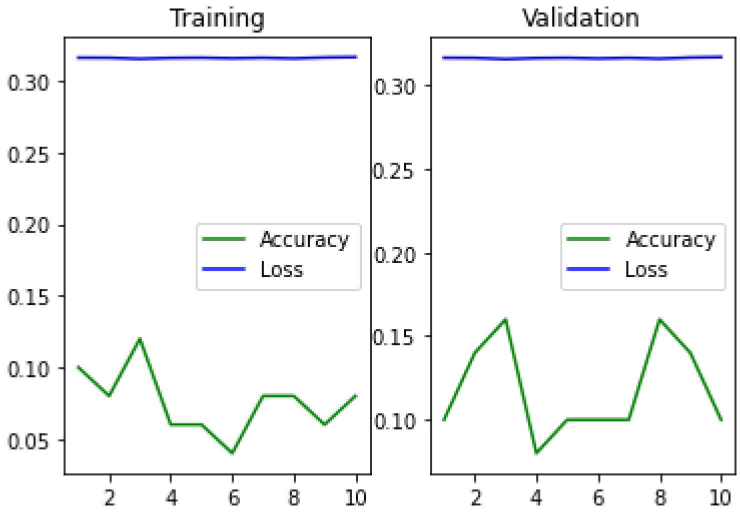
```
Count: 4, j=: 0


**************** Training model: mlp_on_gpu_RegL1 with optimizer: Adam
and seed: 5211 ****************

# Epoch:=1/10  - train loss:=0.1675 - val loss:=0.0462, train acc:=0.1
2 - val acc:=0.14
# Epoch:=2/10  - train loss:=0.1683 - val loss:=0.0461, train acc:=0.1
0 - val acc:=0.10
# Epoch:=3/10  - train loss:=0.1684 - val loss:=0.0461, train acc:=0.1
0 - val acc:=0.08
# Epoch:=4/10  - train loss:=0.1681 - val loss:=0.0463, train acc:=0.1
2 - val acc:=0.12
# Epoch:=5/10  - train loss:=0.1684 - val loss:=0.0462, train acc:=0.1
0 - val acc:=0.14
# Epoch:=6/10  - train loss:=0.1679 - val loss:=0.0462, train acc:=0.1
2 - val acc:=0.10
# Epoch:=7/10  - train loss:=0.1680 - val loss:=0.0461, train acc:=0.1
2 - val acc:=0.12
# Epoch:=8/10  - train loss:=0.1680 - val loss:=0.0461, train acc:=0.1
2 - val acc:=0.12
# Epoch:=9/10  - train loss:=0.1686 - val loss:=0.0462, train acc:=0.1
0 - val acc:=0.10
# Epoch:=10/10  - train loss:=0.1683 - val loss:=0.0461, train acc:=0.
12 - val acc:=0.12

Total time taken (in seconds):  269.76

Finished training model: mlp_on_gpu_RegL1

**************** Testing ****************
mlp_on_gpu_RegL1 model accuracy = 9.74%
*****************************************
```



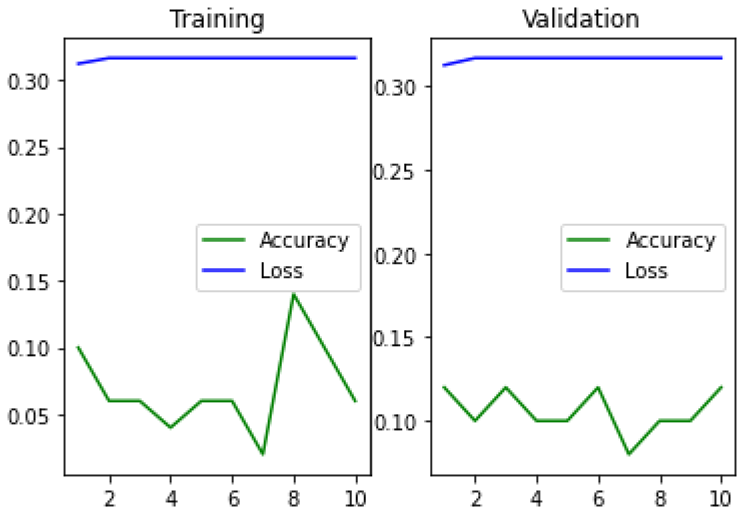
```
Count: 5, j=: 0


**************** Training model: mlp_on_gpu_RegL1 with optimizer: Adam
and seed: 3926 ****************

# Epoch:=1/10  - train loss:=0.2687 - val loss:=0.2881, train acc:=0.0
4 - val acc:=0.12
# Epoch:=2/10  - train loss:=0.2858 - val loss:=0.2881, train acc:=0.0
6 - val acc:=0.14
# Epoch:=3/10  - train loss:=0.2858 - val loss:=0.2881, train acc:=0.0
6 - val acc:=0.12
# Epoch:=4/10  - train loss:=0.2858 - val loss:=0.2881, train acc:=0.0
8 - val acc:=0.10
# Epoch:=5/10  - train loss:=0.2858 - val loss:=0.2881, train acc:=0.0
4 - val acc:=0.08
# Epoch:=6/10  - train loss:=0.2858 - val loss:=0.2881, train acc:=0.0
2 - val acc:=0.10
# Epoch:=7/10  - train loss:=0.2858 - val loss:=0.2881, train acc:=0.0
4 - val acc:=0.10
# Epoch:=8/10  - train loss:=0.2858 - val loss:=0.2881, train acc:=0.0
6 - val acc:=0.12
# Epoch:=9/10  - train loss:=0.2858 - val loss:=0.2881, train acc:=0.0
8 - val acc:=0.10
# Epoch:=10/10  - train loss:=0.2858 - val loss:=0.2881, train acc:=0.
04 - val acc:=0.10

Total time taken (in seconds):  272.92

Finished training model: mlp_on_gpu_RegL1

**************** Testing ****************
mlp_on_gpu_RegL1 model accuracy = 11.35%
*****************************************
```



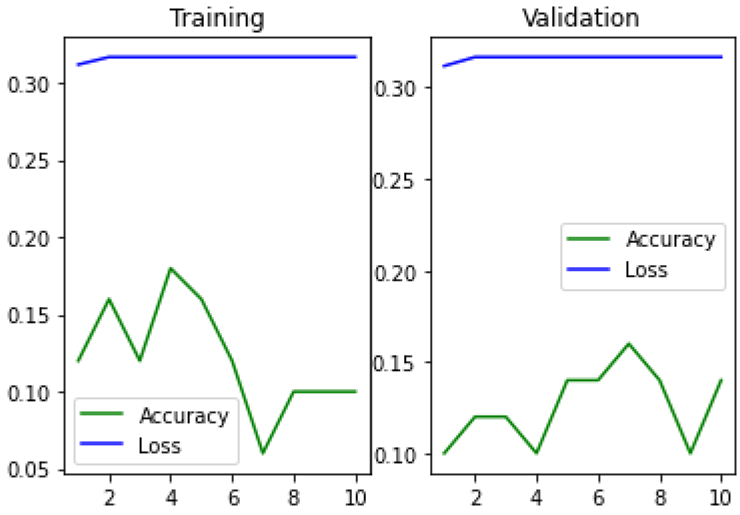
```
Count: 6, j=: 0


**************** Training model: mlp_on_gpu_RegL1 with optimizer: Adam
and seed: 9681 ****************

# Epoch:=1/10  - train loss:=0.1661 - val loss:=0.0462, train acc:=0.1
0 - val acc:=0.10
# Epoch:=2/10  - train loss:=0.1660 - val loss:=0.0461, train acc:=0.0
8 - val acc:=0.14
# Epoch:=3/10  - train loss:=0.1657 - val loss:=0.0461, train acc:=0.1
2 - val acc:=0.16
# Epoch:=4/10  - train loss:=0.1660 - val loss:=0.0461, train acc:=0.0
6 - val acc:=0.08
# Epoch:=5/10  - train loss:=0.1661 - val loss:=0.0461, train acc:=0.0
6 - val acc:=0.10
# Epoch:=6/10  - train loss:=0.1659 - val loss:=0.0461, train acc:=0.0
4 - val acc:=0.10
# Epoch:=7/10  - train loss:=0.1660 - val loss:=0.0461, train acc:=0.0
8 - val acc:=0.10
# Epoch:=8/10  - train loss:=0.1658 - val loss:=0.0461, train acc:=0.0
8 - val acc:=0.16
# Epoch:=9/10  - train loss:=0.1662 - val loss:=0.0462, train acc:=0.0
6 - val acc:=0.14
# Epoch:=10/10  - train loss:=0.1663 - val loss:=0.0461, train acc:=0.
08 - val acc:=0.10

Total time taken (in seconds):  272.12

Finished training model: mlp_on_gpu_RegL1

**************** Testing ****************
mlp_on_gpu_RegL1 model accuracy = 11.35%
*****************************************
```



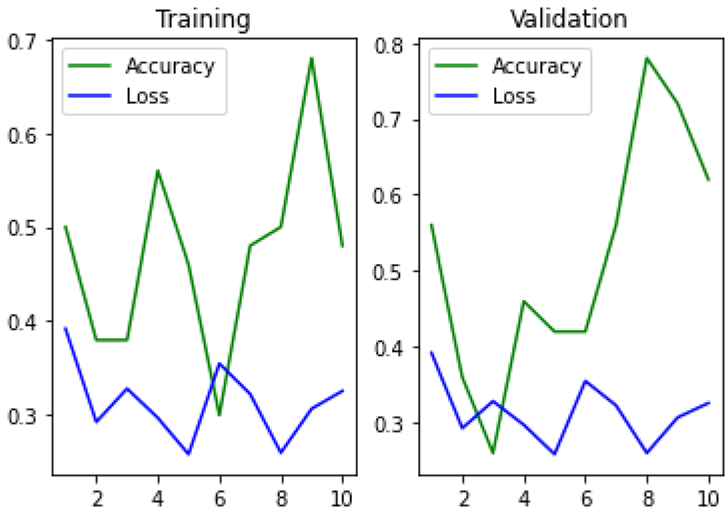
```
Count: 7, j=: 0


**************** Training model: mlp_on_gpu_RegL1 with optimizer: Adam
and seed: 3416 ****************

# Epoch:=1/10  - train loss:=0.2819 - val loss:=0.2881, train acc:=0.1
0 - val acc:=0.12
# Epoch:=2/10  - train loss:=0.2858 - val loss:=0.2881, train acc:=0.0
6 - val acc:=0.10
# Epoch:=3/10  - train loss:=0.2858 - val loss:=0.2881, train acc:=0.0
6 - val acc:=0.12
# Epoch:=4/10  - train loss:=0.2858 - val loss:=0.2881, train acc:=0.0
4 - val acc:=0.10
# Epoch:=5/10  - train loss:=0.2858 - val loss:=0.2881, train acc:=0.0
6 - val acc:=0.10
# Epoch:=6/10  - train loss:=0.2858 - val loss:=0.2881, train acc:=0.0
6 - val acc:=0.12
# Epoch:=7/10  - train loss:=0.2858 - val loss:=0.2881, train acc:=0.0
2 - val acc:=0.08
# Epoch:=8/10  - train loss:=0.2858 - val loss:=0.2881, train acc:=0.1
4 - val acc:=0.10
# Epoch:=9/10  - train loss:=0.2858 - val loss:=0.2881, train acc:=0.1
0 - val acc:=0.10
# Epoch:=10/10  - train loss:=0.2858 - val loss:=0.2881, train acc:=0.
06 - val acc:=0.12

Total time taken (in seconds):  271.40

Finished training model: mlp_on_gpu_RegL1

**************** Testing ****************
mlp_on_gpu_RegL1 model accuracy = 11.35%
*****************************************
```



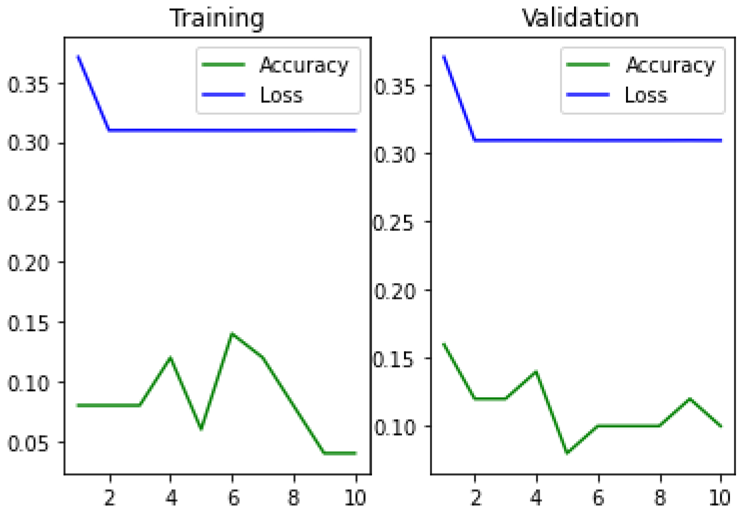
```
Count: 8, j=: 0


**************** Training model: mlp_on_gpu_RegL1 with optimizer: Adam
and seed: 7676 ****************

# Epoch:=1/10  - train loss:=0.2850 - val loss:=0.2892, train acc:=0.1
2 - val acc:=0.10
# Epoch:=2/10  - train loss:=0.2895 - val loss:=0.2892, train acc:=0.1
6 - val acc:=0.12
# Epoch:=3/10  - train loss:=0.2895 - val loss:=0.2892, train acc:=0.1
2 - val acc:=0.12
# Epoch:=4/10  - train loss:=0.2895 - val loss:=0.2892, train acc:=0.1
8 - val acc:=0.10
# Epoch:=5/10  - train loss:=0.2895 - val loss:=0.2892, train acc:=0.1
6 - val acc:=0.14
# Epoch:=6/10  - train loss:=0.2895 - val loss:=0.2892, train acc:=0.1
2 - val acc:=0.14
# Epoch:=7/10  - train loss:=0.2895 - val loss:=0.2892, train acc:=0.0
6 - val acc:=0.16
# Epoch:=8/10  - train loss:=0.2895 - val loss:=0.2892, train acc:=0.1
0 - val acc:=0.14
# Epoch:=9/10  - train loss:=0.2895 - val loss:=0.2892, train acc:=0.1
0 - val acc:=0.10
# Epoch:=10/10  - train loss:=0.2895 - val loss:=0.2892, train acc:=0.
10 - val acc:=0.14

Total time taken (in seconds):  271.09

Finished training model: mlp_on_gpu_RegL1

**************** Testing ****************
mlp_on_gpu_RegL1 model accuracy = 10.10%
*****************************************
```



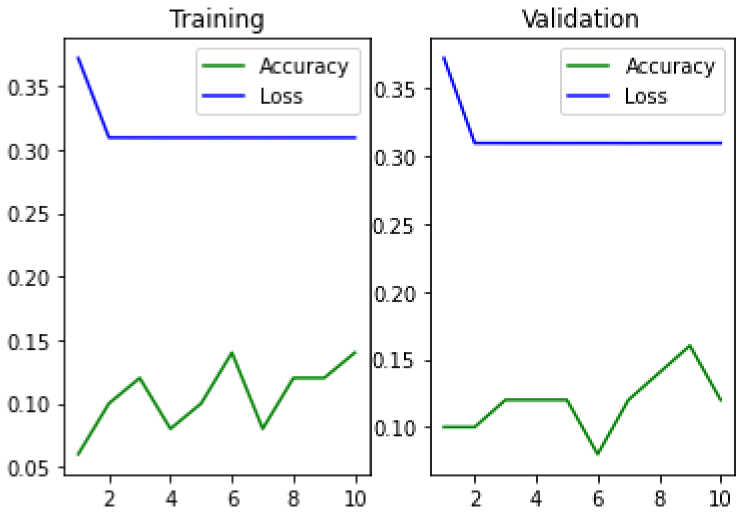
```
Count: 9, j=: 0


**************** Training model: mlp_on_gpu_RegL1 with optimizer: Adam
and seed: 7146 ****************

# Epoch:=1/10  - train loss:=0.1158 - val loss:=0.0920, train acc:=0.5
0 - val acc:=0.56
# Epoch:=2/10  - train loss:=0.0865 - val loss:=0.1256, train acc:=0.3
8 - val acc:=0.36
# Epoch:=3/10  - train loss:=0.0970 - val loss:=0.2020, train acc:=0.3
8 - val acc:=0.26
# Epoch:=4/10  - train loss:=0.0878 - val loss:=0.1093, train acc:=0.5
6 - val acc:=0.46
# Epoch:=5/10  - train loss:=0.0763 - val loss:=0.0670, train acc:=0.4
6 - val acc:=0.42
# Epoch:=6/10  - train loss:=0.1049 - val loss:=0.1938, train acc:=0.3
0 - val acc:=0.42
# Epoch:=7/10  - train loss:=0.0953 - val loss:=0.1381, train acc:=0.4
8 - val acc:=0.56
# Epoch:=8/10  - train loss:=0.0768 - val loss:=0.0771, train acc:=0.5
0 - val acc:=0.78
# Epoch:=9/10  - train loss:=0.0907 - val loss:=0.0825, train acc:=0.6
8 - val acc:=0.72
# Epoch:=10/10  - train loss:=0.0963 - val loss:=0.0968, train acc:=0.
48 - val acc:=0.62

Total time taken (in seconds):  270.11

Finished training model: mlp_on_gpu_RegL1

**************** Testing ****************
mlp_on_gpu_RegL1 model accuracy = 63.19%
*****************************************
```



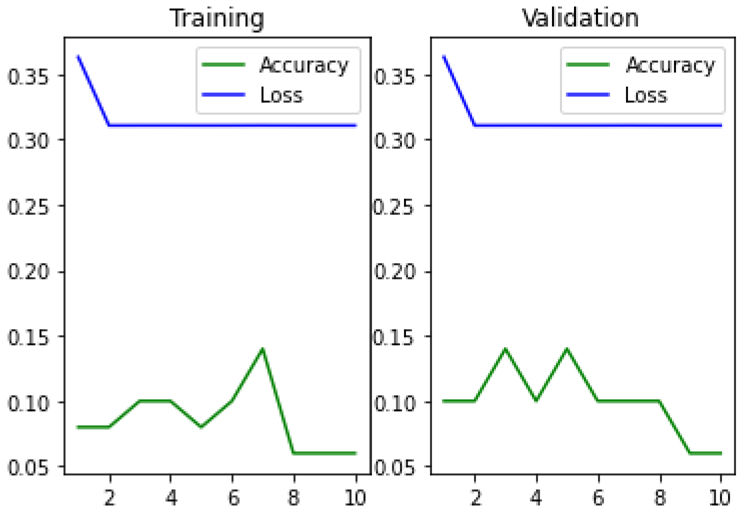
```
Count: 0, j=: 0


**************** Training model: mlp_on_gpu_RegL2 with optimizer: Adam
and seed: 7474 ****************

# Epoch:=1/10  - train loss:=0.0553 - val loss:=0.0461, train acc:=0.0
8 - val acc:=0.16
# Epoch:=2/10  - train loss:=0.0462 - val loss:=0.0461, train acc:=0.0
8 - val acc:=0.12
# Epoch:=3/10  - train loss:=0.0462 - val loss:=0.0461, train acc:=0.0
8 - val acc:=0.12
# Epoch:=4/10  - train loss:=0.0462 - val loss:=0.0461, train acc:=0.1
2 - val acc:=0.14
# Epoch:=5/10  - train loss:=0.0462 - val loss:=0.0461, train acc:=0.0
6 - val acc:=0.08
# Epoch:=6/10  - train loss:=0.0462 - val loss:=0.0461, train acc:=0.1
4 - val acc:=0.10
# Epoch:=7/10  - train loss:=0.0462 - val loss:=0.0461, train acc:=0.1
2 - val acc:=0.10
# Epoch:=8/10  - train loss:=0.0462 - val loss:=0.0462, train acc:=0.0
8 - val acc:=0.10
# Epoch:=9/10  - train loss:=0.0462 - val loss:=0.0462, train acc:=0.0
4 - val acc:=0.12
# Epoch:=10/10  - train loss:=0.0462 - val loss:=0.0461, train acc:=0.
04 - val acc:=0.10

Total time taken (in seconds):  291.97

Finished training model: mlp_on_gpu_RegL2

**************** Testing ****************
mlp_on_gpu_RegL2 model accuracy = 11.35%
*****************************************
```



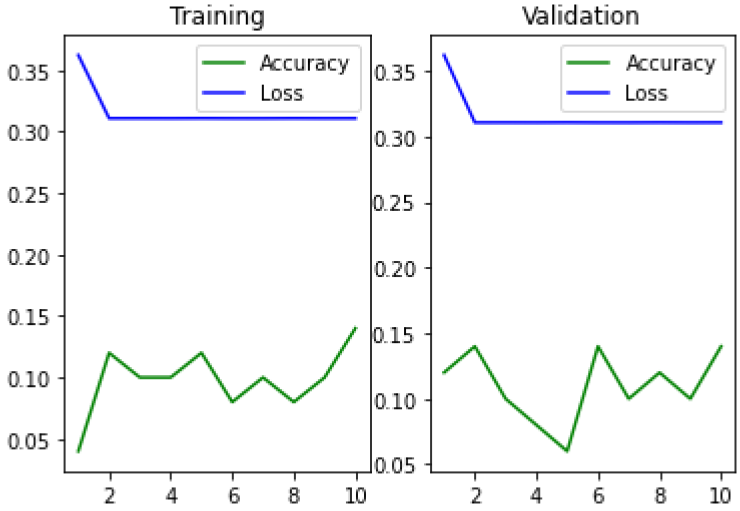
```
Count: 1, j=: 0


**************** Training model: mlp_on_gpu_RegL2 with optimizer: Adam
and seed: 9991 ****************

# Epoch:=1/10  - train loss:=0.0555 - val loss:=0.0461, train acc:=0.0
6 - val acc:=0.10
# Epoch:=2/10  - train loss:=0.0462 - val loss:=0.0461, train acc:=0.1
0 - val acc:=0.10
# Epoch:=3/10  - train loss:=0.0462 - val loss:=0.0461, train acc:=0.1
2 - val acc:=0.12
# Epoch:=4/10  - train loss:=0.0462 - val loss:=0.0461, train acc:=0.0
8 - val acc:=0.12
# Epoch:=5/10  - train loss:=0.0462 - val loss:=0.0461, train acc:=0.1
0 - val acc:=0.12
# Epoch:=6/10  - train loss:=0.0462 - val loss:=0.0461, train acc:=0.1
4 - val acc:=0.08
# Epoch:=7/10  - train loss:=0.0462 - val loss:=0.0461, train acc:=0.0
8 - val acc:=0.12
# Epoch:=8/10  - train loss:=0.0462 - val loss:=0.0461, train acc:=0.1
2 - val acc:=0.14
# Epoch:=9/10  - train loss:=0.0462 - val loss:=0.0461, train acc:=0.1
2 - val acc:=0.16
# Epoch:=10/10  - train loss:=0.0462 - val loss:=0.0461, train acc:=0.
14 - val acc:=0.12

Total time taken (in seconds):  291.51

Finished training model: mlp_on_gpu_RegL2

**************** Testing ****************
mlp_on_gpu_RegL2 model accuracy = 10.10%
*****************************************
```



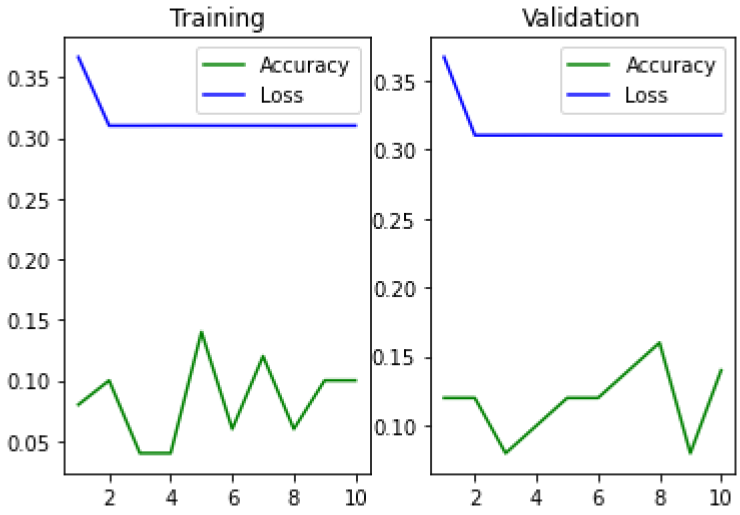
```
Count: 2, j=: 0


**************** Training model: mlp_on_gpu_RegL2 with optimizer: Adam
and seed: 1400 ****************

# Epoch:=1/10  - train loss:=0.0540 - val loss:=0.0462, train acc:=0.0
8 - val acc:=0.10
# Epoch:=2/10  - train loss:=0.0462 - val loss:=0.0461, train acc:=0.0
8 - val acc:=0.10
# Epoch:=3/10  - train loss:=0.0462 - val loss:=0.0462, train acc:=0.1
0 - val acc:=0.14
# Epoch:=4/10  - train loss:=0.0462 - val loss:=0.0461, train acc:=0.1
0 - val acc:=0.10
# Epoch:=5/10  - train loss:=0.0462 - val loss:=0.0461, train acc:=0.0
8 - val acc:=0.14
# Epoch:=6/10  - train loss:=0.0462 - val loss:=0.0461, train acc:=0.1
0 - val acc:=0.10
# Epoch:=7/10  - train loss:=0.0462 - val loss:=0.0461, train acc:=0.1
4 - val acc:=0.10
# Epoch:=8/10  - train loss:=0.0462 - val loss:=0.0461, train acc:=0.0
6 - val acc:=0.10
# Epoch:=9/10  - train loss:=0.0462 - val loss:=0.0461, train acc:=0.0
6 - val acc:=0.06
# Epoch:=10/10  - train loss:=0.0462 - val loss:=0.0461, train acc:=0.
06 - val acc:=0.06

Total time taken (in seconds):  291.21

Finished training model: mlp_on_gpu_RegL2

**************** Testing ****************
mlp_on_gpu_RegL2 model accuracy = 10.10%
*****************************************
```



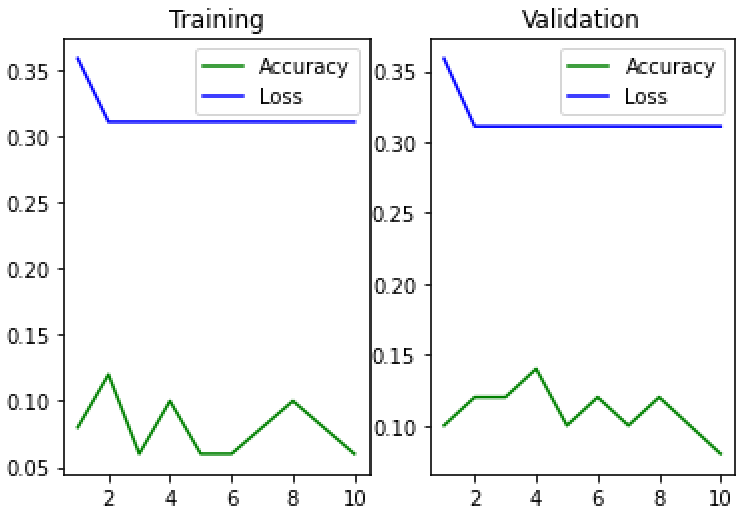
```
Count: 3, j=: 0


**************** Training model: mlp_on_gpu_RegL2 with optimizer: Adam
and seed: 5786 ****************

# Epoch:=1/10  - train loss:=0.0538 - val loss:=0.0461, train acc:=0.0
4 - val acc:=0.12
# Epoch:=2/10  - train loss:=0.0462 - val loss:=0.0462, train acc:=0.1
2 - val acc:=0.14
# Epoch:=3/10  - train loss:=0.0462 - val loss:=0.0461, train acc:=0.1
0 - val acc:=0.10
# Epoch:=4/10  - train loss:=0.0462 - val loss:=0.0462, train acc:=0.1
0 - val acc:=0.08
# Epoch:=5/10  - train loss:=0.0462 - val loss:=0.0462, train acc:=0.1
2 - val acc:=0.06
# Epoch:=6/10  - train loss:=0.0462 - val loss:=0.0461, train acc:=0.0
8 - val acc:=0.14
# Epoch:=7/10  - train loss:=0.0462 - val loss:=0.0461, train acc:=0.1
0 - val acc:=0.10
# Epoch:=8/10  - train loss:=0.0462 - val loss:=0.0461, train acc:=0.0
8 - val acc:=0.12
# Epoch:=9/10  - train loss:=0.0462 - val loss:=0.0461, train acc:=0.1
0 - val acc:=0.10
# Epoch:=10/10  - train loss:=0.0462 - val loss:=0.0461, train acc:=0.
14 - val acc:=0.14

Total time taken (in seconds):  290.44

Finished training model: mlp_on_gpu_RegL2

**************** Testing ****************
mlp_on_gpu_RegL2 model accuracy = 10.10%
*****************************************
```



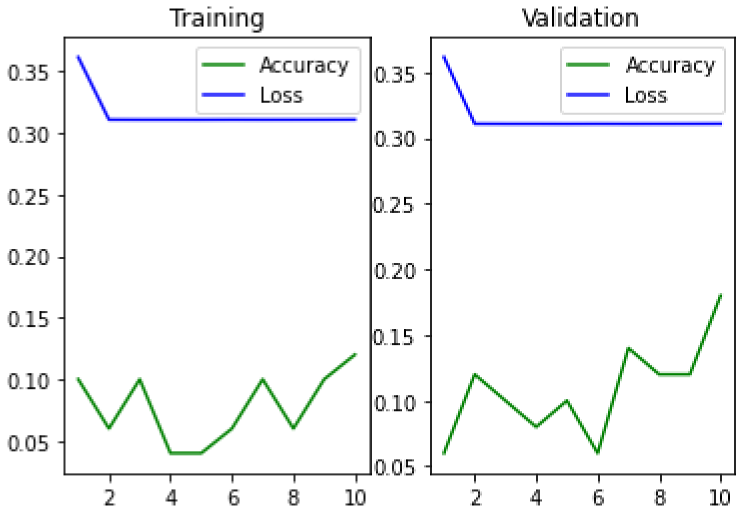
```
Count: 4, j=: 0


**************** Training model: mlp_on_gpu_RegL2 with optimizer: Adam
and seed: 5720 ****************

# Epoch:=1/10  - train loss:=0.0546 - val loss:=0.0461, train acc:=0.0
8 - val acc:=0.12
# Epoch:=2/10  - train loss:=0.0462 - val loss:=0.0462, train acc:=0.1
0 - val acc:=0.12
# Epoch:=3/10  - train loss:=0.0462 - val loss:=0.0462, train acc:=0.0
4 - val acc:=0.08
# Epoch:=4/10  - train loss:=0.0462 - val loss:=0.0461, train acc:=0.0
4 - val acc:=0.10
# Epoch:=5/10  - train loss:=0.0462 - val loss:=0.0461, train acc:=0.1
4 - val acc:=0.12
# Epoch:=6/10  - train loss:=0.0462 - val loss:=0.0461, train acc:=0.0
6 - val acc:=0.12
# Epoch:=7/10  - train loss:=0.0462 - val loss:=0.0461, train acc:=0.1
2 - val acc:=0.14
# Epoch:=8/10  - train loss:=0.0462 - val loss:=0.0461, train acc:=0.0
6 - val acc:=0.16
# Epoch:=9/10  - train loss:=0.0462 - val loss:=0.0461, train acc:=0.1
0 - val acc:=0.08
# Epoch:=10/10  - train loss:=0.0462 - val loss:=0.0461, train acc:=0.
10 - val acc:=0.14

Total time taken (in seconds):  289.27

Finished training model: mlp_on_gpu_RegL2

**************** Testing ****************
mlp_on_gpu_RegL2 model accuracy = 9.58%
*****************************************
```



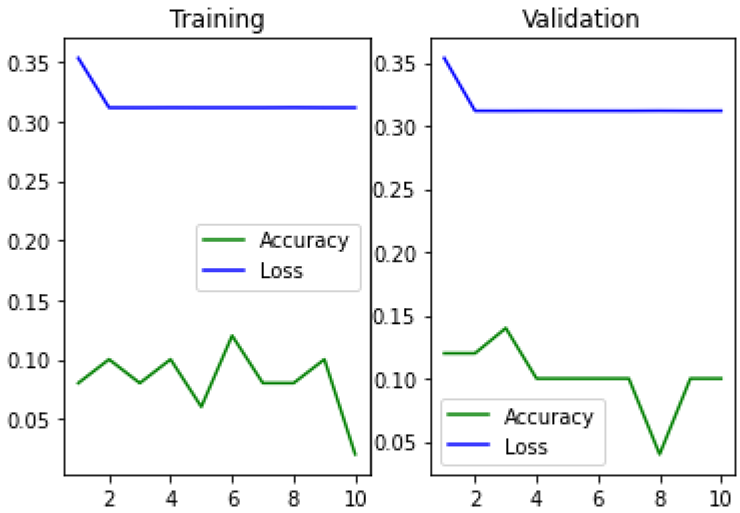
```
Count: 5, j=: 0


**************** Training model: mlp_on_gpu_RegL2 with optimizer: Adam
and seed: 3272 ****************

# Epoch:=1/10  - train loss:=0.0533 - val loss:=0.0461, train acc:=0.0
8 - val acc:=0.10
# Epoch:=2/10  - train loss:=0.0462 - val loss:=0.0461, train acc:=0.1
2 - val acc:=0.12
# Epoch:=3/10  - train loss:=0.0462 - val loss:=0.0461, train acc:=0.0
6 - val acc:=0.12
# Epoch:=4/10  - train loss:=0.0462 - val loss:=0.0461, train acc:=0.1
0 - val acc:=0.14
# Epoch:=5/10  - train loss:=0.0462 - val loss:=0.0462, train acc:=0.0
6 - val acc:=0.10
# Epoch:=6/10  - train loss:=0.0462 - val loss:=0.0462, train acc:=0.0
6 - val acc:=0.12
# Epoch:=7/10  - train loss:=0.0462 - val loss:=0.0461, train acc:=0.0
8 - val acc:=0.10
# Epoch:=8/10  - train loss:=0.0462 - val loss:=0.0462, train acc:=0.1
0 - val acc:=0.12
# Epoch:=9/10  - train loss:=0.0462 - val loss:=0.0461, train acc:=0.0
8 - val acc:=0.10
# Epoch:=10/10  - train loss:=0.0462 - val loss:=0.0461, train acc:=0.
06 - val acc:=0.08

Total time taken (in seconds):  289.53

Finished training model: mlp_on_gpu_RegL2

**************** Testing ****************
mlp_on_gpu_RegL2 model accuracy = 10.09%
*****************************************
```



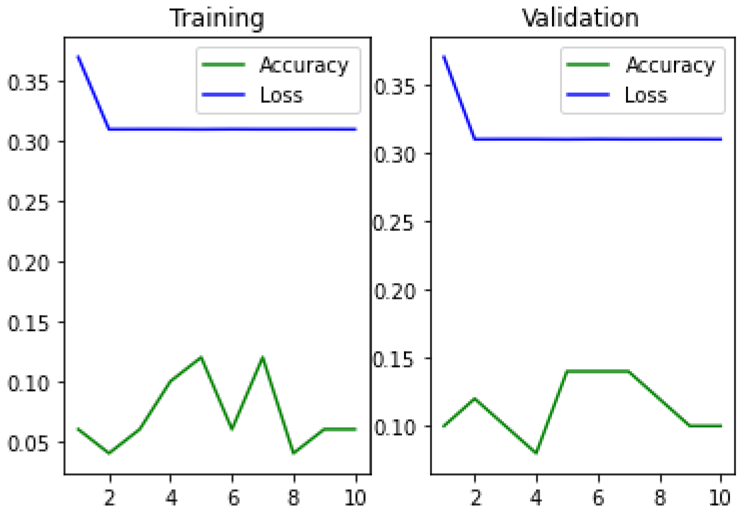
```
Count: 6, j=: 0


**************** Training model: mlp_on_gpu_RegL2 with optimizer: Adam
and seed: 1777 ****************

# Epoch:=1/10  - train loss:=0.0537 - val loss:=0.0461, train acc:=0.1
0 - val acc:=0.06
# Epoch:=2/10  - train loss:=0.0462 - val loss:=0.0461, train acc:=0.0
6 - val acc:=0.12
# Epoch:=3/10  - train loss:=0.0462 - val loss:=0.0461, train acc:=0.1
0 - val acc:=0.10
# Epoch:=4/10  - train loss:=0.0462 - val loss:=0.0461, train acc:=0.0
4 - val acc:=0.08
# Epoch:=5/10  - train loss:=0.0462 - val loss:=0.0461, train acc:=0.0
4 - val acc:=0.10
# Epoch:=6/10  - train loss:=0.0462 - val loss:=0.0461, train acc:=0.0
6 - val acc:=0.06
# Epoch:=7/10  - train loss:=0.0462 - val loss:=0.0462, train acc:=0.1
0 - val acc:=0.14
# Epoch:=8/10  - train loss:=0.0462 - val loss:=0.0461, train acc:=0.0
6 - val acc:=0.12
# Epoch:=9/10  - train loss:=0.0462 - val loss:=0.0461, train acc:=0.1
0 - val acc:=0.12
# Epoch:=10/10  - train loss:=0.0462 - val loss:=0.0461, train acc:=0.
12 - val acc:=0.18

Total time taken (in seconds):  290.63

Finished training model: mlp_on_gpu_RegL2

**************** Testing ****************
mlp_on_gpu_RegL2 model accuracy = 10.10%
*****************************************
```



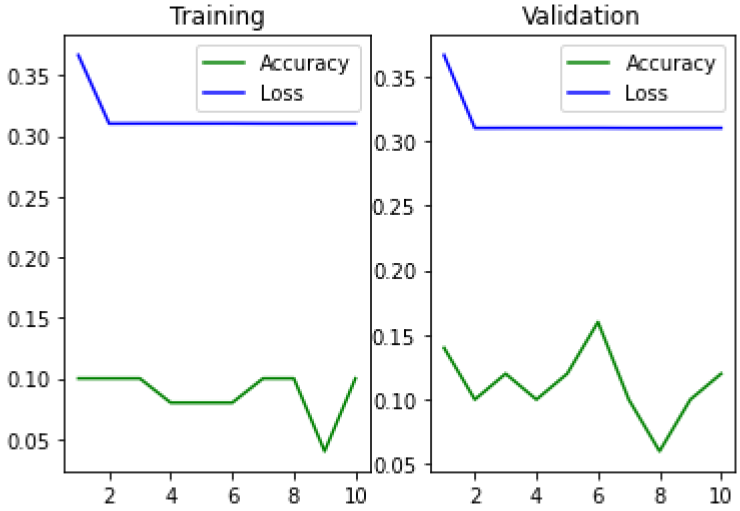
```
Count: 7, j=: 0


**************** Training model: mlp_on_gpu_RegL2 with optimizer: Adam
and seed: 3305 ****************

# Epoch:=1/10  - train loss:=0.0524 - val loss:=0.0461, train acc:=0.0
8 - val acc:=0.12
# Epoch:=2/10  - train loss:=0.0462 - val loss:=0.0461, train acc:=0.1
0 - val acc:=0.12
# Epoch:=3/10  - train loss:=0.0462 - val loss:=0.0461, train acc:=0.0
8 - val acc:=0.14
# Epoch:=4/10  - train loss:=0.0462 - val loss:=0.0462, train acc:=0.1
0 - val acc:=0.10
# Epoch:=5/10  - train loss:=0.0462 - val loss:=0.0462, train acc:=0.0
6 - val acc:=0.10
# Epoch:=6/10  - train loss:=0.0462 - val loss:=0.0461, train acc:=0.1
2 - val acc:=0.10
# Epoch:=7/10  - train loss:=0.0462 - val loss:=0.0461, train acc:=0.0
8 - val acc:=0.10
# Epoch:=8/10  - train loss:=0.0462 - val loss:=0.0461, train acc:=0.0
8 - val acc:=0.04
# Epoch:=9/10  - train loss:=0.0462 - val loss:=0.0460, train acc:=0.1
0 - val acc:=0.10
# Epoch:=10/10  - train loss:=0.0462 - val loss:=0.0461, train acc:=0.
02 - val acc:=0.10

Total time taken (in seconds):  289.48

Finished training model: mlp_on_gpu_RegL2

**************** Testing ****************
mlp_on_gpu_RegL2 model accuracy = 11.35%
*****************************************
```



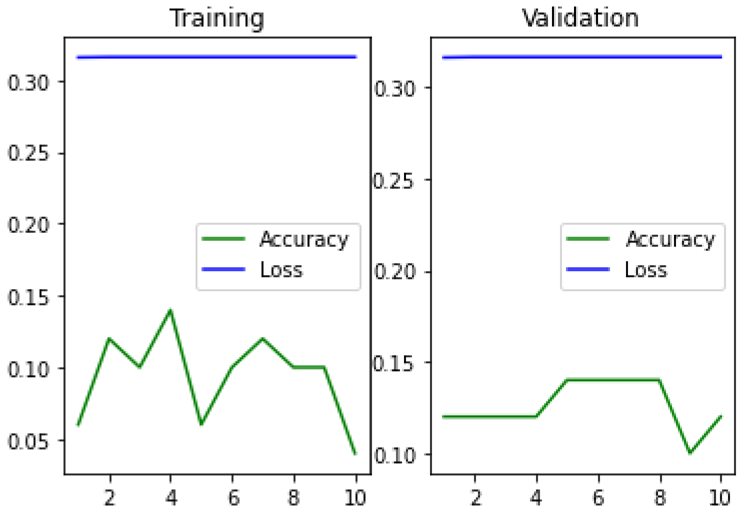
```
Count: 8, j=: 0


**************** Training model: mlp_on_gpu_RegL2 with optimizer: Adam
and seed: 4161 ****************

# Epoch:=1/10  - train loss:=0.0552 - val loss:=0.0461, train acc:=0.0
6 - val acc:=0.10
# Epoch:=2/10  - train loss:=0.0462 - val loss:=0.0461, train acc:=0.0
4 - val acc:=0.12
# Epoch:=3/10  - train loss:=0.0462 - val loss:=0.0461, train acc:=0.0
6 - val acc:=0.10
# Epoch:=4/10  - train loss:=0.0462 - val loss:=0.0461, train acc:=0.1
0 - val acc:=0.08
# Epoch:=5/10  - train loss:=0.0462 - val loss:=0.0461, train acc:=0.1
2 - val acc:=0.14
# Epoch:=6/10  - train loss:=0.0462 - val loss:=0.0461, train acc:=0.0
6 - val acc:=0.14
# Epoch:=7/10  - train loss:=0.0462 - val loss:=0.0461, train acc:=0.1
2 - val acc:=0.14
# Epoch:=8/10  - train loss:=0.0462 - val loss:=0.0462, train acc:=0.0
4 - val acc:=0.12
# Epoch:=9/10  - train loss:=0.0462 - val loss:=0.0461, train acc:=0.0
6 - val acc:=0.10
# Epoch:=10/10  - train loss:=0.0462 - val loss:=0.0461, train acc:=0.
06 - val acc:=0.10

Total time taken (in seconds):  289.32

Finished training model: mlp_on_gpu_RegL2

**************** Testing ****************
mlp_on_gpu_RegL2 model accuracy = 9.80%
*****************************************
```



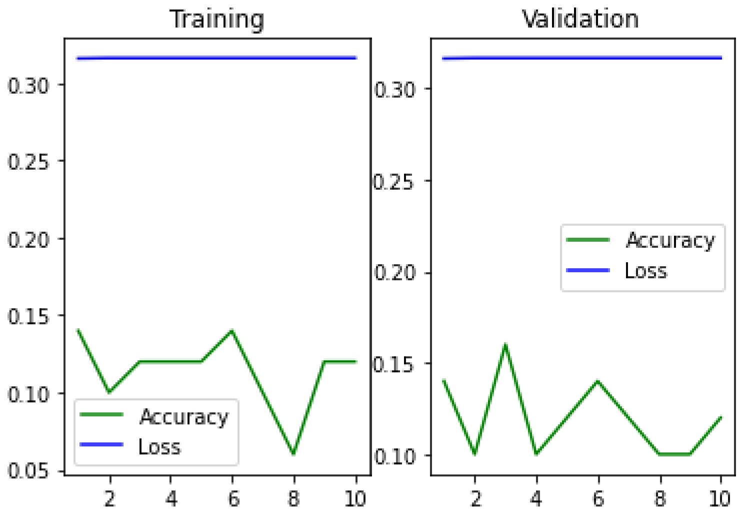
```
Count: 9, j=: 0


**************** Training model: mlp_on_gpu_RegL2 with optimizer: Adam
and seed: 7727 ****************

# Epoch:=1/10  - train loss:=0.0546 - val loss:=0.0461, train acc:=0.1
0 - val acc:=0.14
# Epoch:=2/10  - train loss:=0.0462 - val loss:=0.0461, train acc:=0.1
0 - val acc:=0.10
# Epoch:=3/10  - train loss:=0.0462 - val loss:=0.0462, train acc:=0.1
0 - val acc:=0.12
# Epoch:=4/10  - train loss:=0.0462 - val loss:=0.0461, train acc:=0.0
8 - val acc:=0.10
# Epoch:=5/10  - train loss:=0.0462 - val loss:=0.0461, train acc:=0.0
8 - val acc:=0.12
# Epoch:=6/10  - train loss:=0.0462 - val loss:=0.0462, train acc:=0.0
8 - val acc:=0.16
# Epoch:=7/10  - train loss:=0.0462 - val loss:=0.0461, train acc:=0.1
0 - val acc:=0.10
# Epoch:=8/10  - train loss:=0.0462 - val loss:=0.0462, train acc:=0.1
0 - val acc:=0.06
# Epoch:=9/10  - train loss:=0.0462 - val loss:=0.0461, train acc:=0.0
4 - val acc:=0.10
# Epoch:=10/10  - train loss:=0.0462 - val loss:=0.0461, train acc:=0.
10 - val acc:=0.12

Total time taken (in seconds):  288.70

Finished training model: mlp_on_gpu_RegL2

**************** Testing ****************
mlp_on_gpu_RegL2 model accuracy = 10.10%
*****************************************
```



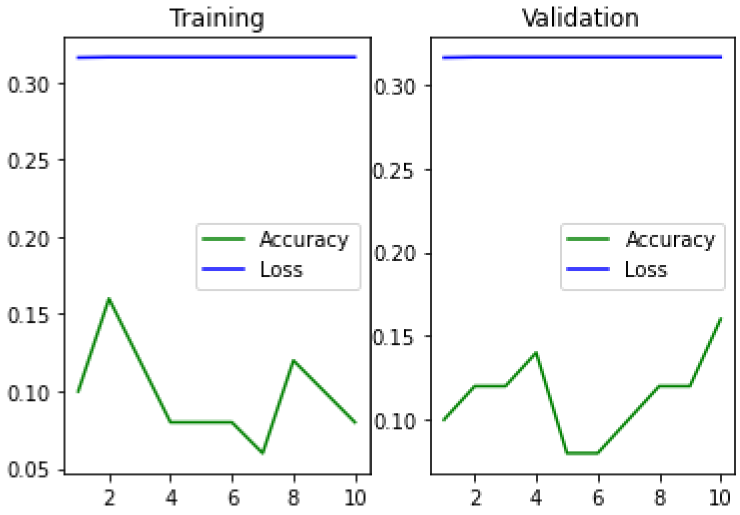
```
Count: 0, j=: 0


**************** Training model: mlp_on_gpu_default with optimizer: RM
SProp and seed: 8392 ****************

# Epoch:=1/10  - train loss:=0.2892 - val loss:=0.2892, train acc:=0.0
6 - val acc:=0.12
# Epoch:=2/10  - train loss:=0.2895 - val loss:=0.2892, train acc:=0.1
2 - val acc:=0.12
# Epoch:=3/10  - train loss:=0.2895 - val loss:=0.2892, train acc:=0.1
0 - val acc:=0.12
# Epoch:=4/10  - train loss:=0.2895 - val loss:=0.2892, train acc:=0.1
4 - val acc:=0.12
# Epoch:=5/10  - train loss:=0.2895 - val loss:=0.2892, train acc:=0.0
6 - val acc:=0.14
# Epoch:=6/10  - train loss:=0.2895 - val loss:=0.2892, train acc:=0.1
0 - val acc:=0.14
# Epoch:=7/10  - train loss:=0.2895 - val loss:=0.2892, train acc:=0.1
2 - val acc:=0.14
# Epoch:=8/10  - train loss:=0.2895 - val loss:=0.2892, train acc:=0.1
0 - val acc:=0.14
# Epoch:=9/10  - train loss:=0.2895 - val loss:=0.2892, train acc:=0.1
0 - val acc:=0.10
# Epoch:=10/10  - train loss:=0.2895 - val loss:=0.2892, train acc:=0.
04 - val acc:=0.12

Total time taken (in seconds):  263.50

Finished training model: mlp_on_gpu_default

**************** Testing ****************
mlp_on_gpu_default model accuracy = 10.10%
*****************************************
```



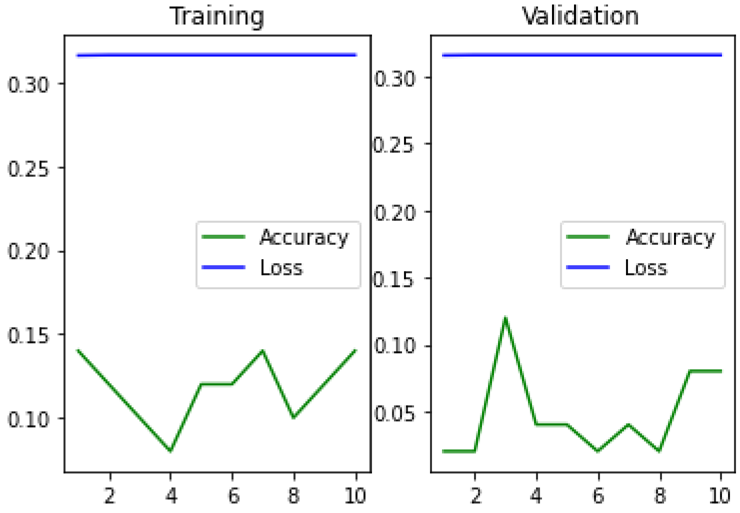
```
Count: 1, j=: 0


**************** Training model: mlp_on_gpu_default with optimizer: RM
SProp and seed: 8247 ****************

# Epoch:=1/10  - train loss:=0.2892 - val loss:=0.2892, train acc:=0.1
4 - val acc:=0.14
# Epoch:=2/10  - train loss:=0.2895 - val loss:=0.2892, train acc:=0.1
0 - val acc:=0.10
# Epoch:=3/10  - train loss:=0.2895 - val loss:=0.2892, train acc:=0.1
2 - val acc:=0.16
# Epoch:=4/10  - train loss:=0.2895 - val loss:=0.2892, train acc:=0.1
2 - val acc:=0.10
# Epoch:=5/10  - train loss:=0.2895 - val loss:=0.2892, train acc:=0.1
2 - val acc:=0.12
# Epoch:=6/10  - train loss:=0.2895 - val loss:=0.2892, train acc:=0.1
4 - val acc:=0.14
# Epoch:=7/10  - train loss:=0.2895 - val loss:=0.2892, train acc:=0.1
0 - val acc:=0.12
# Epoch:=8/10  - train loss:=0.2895 - val loss:=0.2892, train acc:=0.0
6 - val acc:=0.10
# Epoch:=9/10  - train loss:=0.2895 - val loss:=0.2892, train acc:=0.1
2 - val acc:=0.10
# Epoch:=10/10  - train loss:=0.2895 - val loss:=0.2892, train acc:=0.
12 - val acc:=0.12

Total time taken (in seconds):  262.76

Finished training model: mlp_on_gpu_default

**************** Testing ****************
mlp_on_gpu_default model accuracy = 10.10%
*****************************************
```



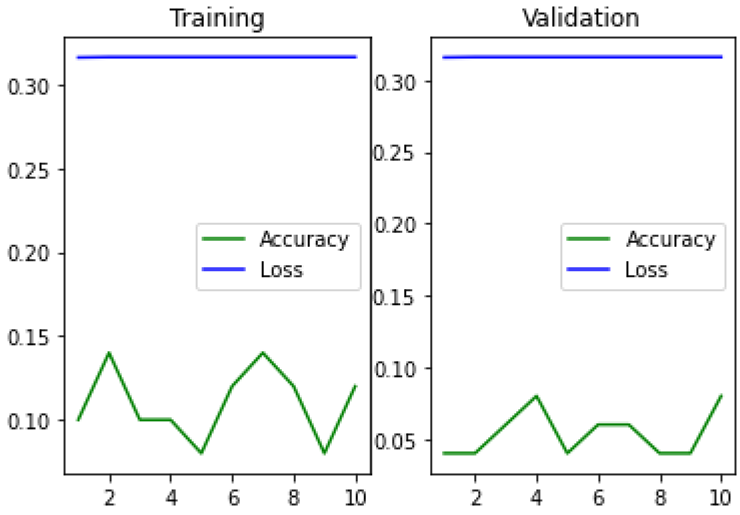
```
Count: 2, j=: 0


**************** Training model: mlp_on_gpu_default with optimizer: RM
SProp and seed: 6253 ****************

# Epoch:=1/10  - train loss:=0.2892 - val loss:=0.2892, train acc:=0.1
0 - val acc:=0.10
# Epoch:=2/10  - train loss:=0.2895 - val loss:=0.2892, train acc:=0.1
6 - val acc:=0.12
# Epoch:=3/10  - train loss:=0.2895 - val loss:=0.2892, train acc:=0.1
2 - val acc:=0.12
# Epoch:=4/10  - train loss:=0.2895 - val loss:=0.2892, train acc:=0.0
8 - val acc:=0.14
# Epoch:=5/10  - train loss:=0.2895 - val loss:=0.2892, train acc:=0.0
8 - val acc:=0.08
# Epoch:=6/10  - train loss:=0.2895 - val loss:=0.2892, train acc:=0.0
8 - val acc:=0.08
# Epoch:=7/10  - train loss:=0.2895 - val loss:=0.2892, train acc:=0.0
6 - val acc:=0.10
# Epoch:=8/10  - train loss:=0.2895 - val loss:=0.2892, train acc:=0.1
2 - val acc:=0.12
# Epoch:=9/10  - train loss:=0.2895 - val loss:=0.2892, train acc:=0.1
0 - val acc:=0.12
# Epoch:=10/10  - train loss:=0.2895 - val loss:=0.2892, train acc:=0.
08 - val acc:=0.16

Total time taken (in seconds):  262.45

Finished training model: mlp_on_gpu_default

**************** Testing ****************
mlp_on_gpu_default model accuracy = 10.10%
*****************************************
```



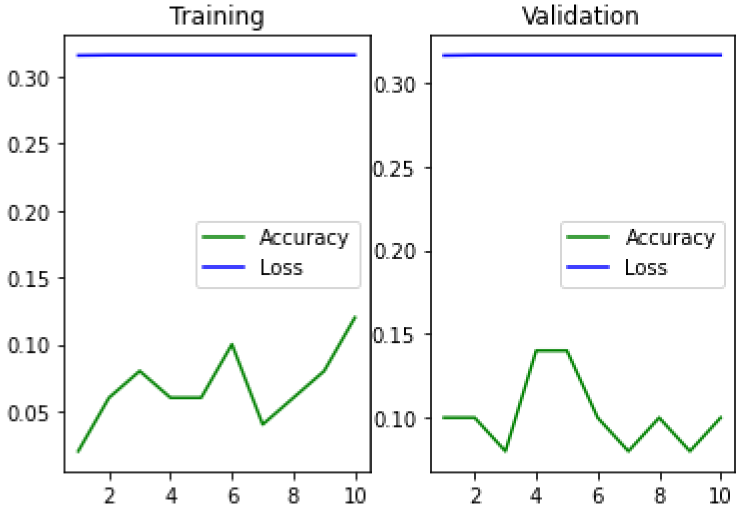
```
Count: 3, j=: 0


**************** Training model: mlp_on_gpu_default with optimizer: RM
SProp and seed: 8530 ****************

# Epoch:=1/10  - train loss:=0.2908 - val loss:=0.2907, train acc:=0.1
4 - val acc:=0.02
# Epoch:=2/10  - train loss:=0.2910 - val loss:=0.2907, train acc:=0.1
2 - val acc:=0.02
# Epoch:=3/10  - train loss:=0.2910 - val loss:=0.2907, train acc:=0.1
0 - val acc:=0.12
# Epoch:=4/10  - train loss:=0.2910 - val loss:=0.2907, train acc:=0.0
8 - val acc:=0.04
# Epoch:=5/10  - train loss:=0.2910 - val loss:=0.2907, train acc:=0.1
2 - val acc:=0.04
# Epoch:=6/10  - train loss:=0.2910 - val loss:=0.2907, train acc:=0.1
2 - val acc:=0.02
# Epoch:=7/10  - train loss:=0.2910 - val loss:=0.2907, train acc:=0.1
4 - val acc:=0.04
# Epoch:=8/10  - train loss:=0.2910 - val loss:=0.2907, train acc:=0.1
0 - val acc:=0.02
# Epoch:=9/10  - train loss:=0.2910 - val loss:=0.2907, train acc:=0.1
2 - val acc:=0.08
# Epoch:=10/10  - train loss:=0.2910 - val loss:=0.2907, train acc:=0.
14 - val acc:=0.08

Total time taken (in seconds):  261.61

Finished training model: mlp_on_gpu_default

**************** Testing ****************
mlp_on_gpu_default model accuracy = 9.82%
*****************************************
```



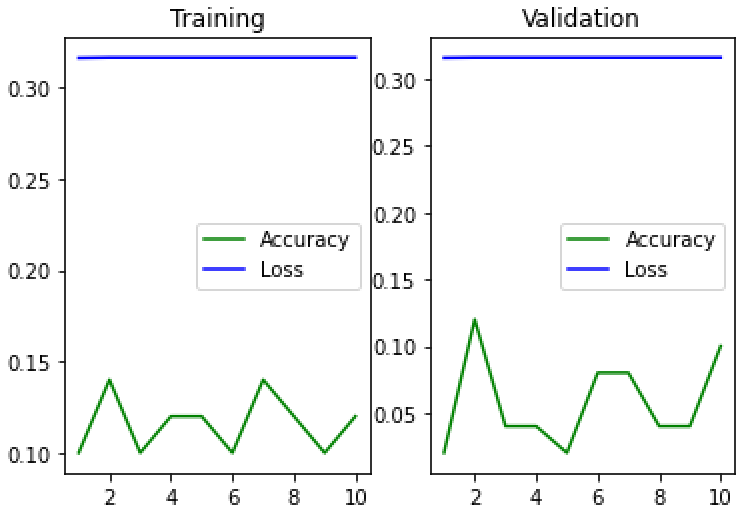
```
Count: 4, j=: 0


**************** Training model: mlp_on_gpu_default with optimizer: RM
SProp and seed: 4385 ****************

# Epoch:=1/10  - train loss:=0.2908 - val loss:=0.2907, train acc:=0.1
0 - val acc:=0.04
# Epoch:=2/10  - train loss:=0.2910 - val loss:=0.2907, train acc:=0.1
4 - val acc:=0.04
# Epoch:=3/10  - train loss:=0.2910 - val loss:=0.2907, train acc:=0.1
0 - val acc:=0.06
# Epoch:=4/10  - train loss:=0.2910 - val loss:=0.2907, train acc:=0.1
0 - val acc:=0.08
# Epoch:=5/10  - train loss:=0.2910 - val loss:=0.2907, train acc:=0.0
8 - val acc:=0.04
# Epoch:=6/10  - train loss:=0.2910 - val loss:=0.2907, train acc:=0.1
2 - val acc:=0.06
# Epoch:=7/10  - train loss:=0.2910 - val loss:=0.2907, train acc:=0.1
4 - val acc:=0.06
# Epoch:=8/10  - train loss:=0.2910 - val loss:=0.2907, train acc:=0.1
2 - val acc:=0.04
# Epoch:=9/10  - train loss:=0.2910 - val loss:=0.2907, train acc:=0.0
8 - val acc:=0.04
# Epoch:=10/10  - train loss:=0.2910 - val loss:=0.2907, train acc:=0.
12 - val acc:=0.08

Total time taken (in seconds):  261.71

Finished training model: mlp_on_gpu_default

**************** Testing ****************
mlp_on_gpu_default model accuracy = 9.82%
*****************************************
```



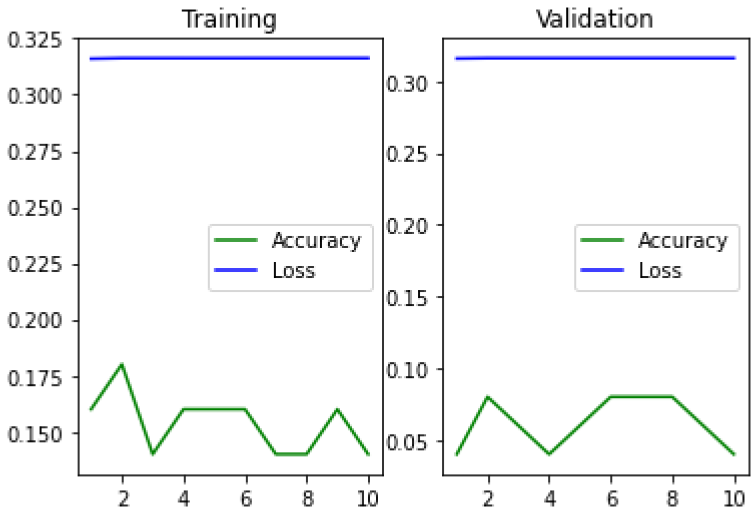
```
Count: 5, j=: 0


**************** Training model: mlp_on_gpu_default with optimizer: RM
SProp and seed: 2969 ****************

# Epoch:=1/10  - train loss:=0.2855 - val loss:=0.2881, train acc:=0.0
2 - val acc:=0.10
# Epoch:=2/10  - train loss:=0.2858 - val loss:=0.2881, train acc:=0.0
6 - val acc:=0.10
# Epoch:=3/10  - train loss:=0.2858 - val loss:=0.2881, train acc:=0.0
8 - val acc:=0.08
# Epoch:=4/10  - train loss:=0.2858 - val loss:=0.2881, train acc:=0.0
6 - val acc:=0.14
# Epoch:=5/10  - train loss:=0.2858 - val loss:=0.2881, train acc:=0.0
6 - val acc:=0.14
# Epoch:=6/10  - train loss:=0.2858 - val loss:=0.2881, train acc:=0.1
0 - val acc:=0.10
# Epoch:=7/10  - train loss:=0.2858 - val loss:=0.2881, train acc:=0.0
4 - val acc:=0.08
# Epoch:=8/10  - train loss:=0.2858 - val loss:=0.2881, train acc:=0.0
6 - val acc:=0.10
# Epoch:=9/10  - train loss:=0.2858 - val loss:=0.2881, train acc:=0.0
8 - val acc:=0.08
# Epoch:=10/10  - train loss:=0.2858 - val loss:=0.2881, train acc:=0.
12 - val acc:=0.10

Total time taken (in seconds):  261.01

Finished training model: mlp_on_gpu_default

**************** Testing ****************
mlp_on_gpu_default model accuracy = 11.35%
*****************************************
```



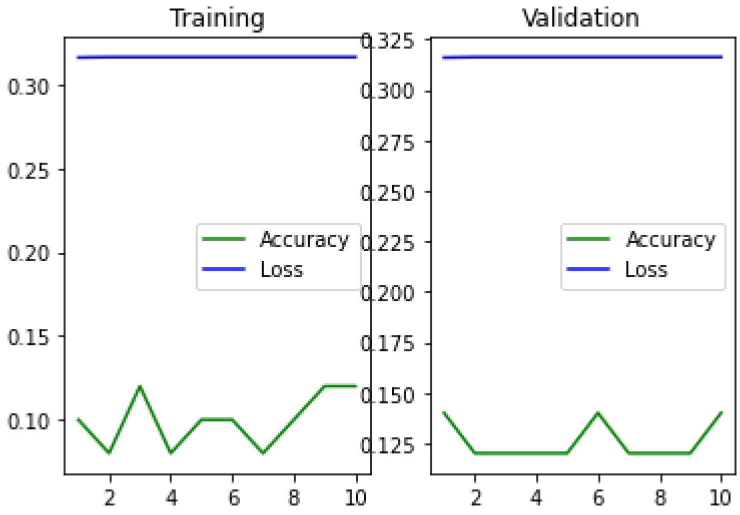
```
Count: 6, j=: 0


**************** Training model: mlp_on_gpu_default with optimizer: RM
SProp and seed: 4828 ****************

# Epoch:=1/10  - train loss:=0.2908 - val loss:=0.2907, train acc:=0.1
0 - val acc:=0.02
# Epoch:=2/10  - train loss:=0.2910 - val loss:=0.2907, train acc:=0.1
4 - val acc:=0.12
# Epoch:=3/10  - train loss:=0.2910 - val loss:=0.2907, train acc:=0.1
0 - val acc:=0.04
# Epoch:=4/10  - train loss:=0.2910 - val loss:=0.2907, train acc:=0.1
2 - val acc:=0.04
# Epoch:=5/10  - train loss:=0.2910 - val loss:=0.2907, train acc:=0.1
2 - val acc:=0.02
# Epoch:=6/10  - train loss:=0.2910 - val loss:=0.2907, train acc:=0.1
0 - val acc:=0.08
# Epoch:=7/10  - train loss:=0.2910 - val loss:=0.2907, train acc:=0.1
4 - val acc:=0.08
# Epoch:=8/10  - train loss:=0.2910 - val loss:=0.2907, train acc:=0.1
2 - val acc:=0.04
# Epoch:=9/10  - train loss:=0.2910 - val loss:=0.2907, train acc:=0.1
0 - val acc:=0.04
# Epoch:=10/10  - train loss:=0.2910 - val loss:=0.2907, train acc:=0.
12 - val acc:=0.10

Total time taken (in seconds):  260.91

Finished training model: mlp_on_gpu_default

**************** Testing ****************
mlp_on_gpu_default model accuracy = 9.82%
*****************************************
```



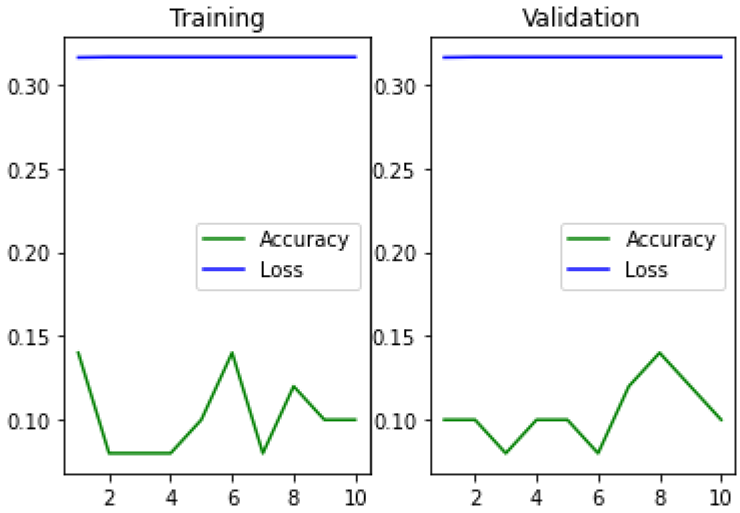
```
Count: 7, j=: 0


**************** Training model: mlp_on_gpu_default with optimizer: RM
SProp and seed: 8984 ****************

# Epoch:=1/10  - train loss:=0.2903 - val loss:=0.2904, train acc:=0.1
6 - val acc:=0.04
# Epoch:=2/10  - train loss:=0.2906 - val loss:=0.2904, train acc:=0.1
8 - val acc:=0.08
# Epoch:=3/10  - train loss:=0.2906 - val loss:=0.2904, train acc:=0.1
4 - val acc:=0.06
# Epoch:=4/10  - train loss:=0.2906 - val loss:=0.2904, train acc:=0.1
6 - val acc:=0.04
# Epoch:=5/10  - train loss:=0.2906 - val loss:=0.2904, train acc:=0.1
6 - val acc:=0.06
# Epoch:=6/10  - train loss:=0.2906 - val loss:=0.2904, train acc:=0.1
6 - val acc:=0.08
# Epoch:=7/10  - train loss:=0.2906 - val loss:=0.2904, train acc:=0.1
4 - val acc:=0.08
# Epoch:=8/10  - train loss:=0.2906 - val loss:=0.2904, train acc:=0.1
4 - val acc:=0.08
# Epoch:=9/10  - train loss:=0.2906 - val loss:=0.2904, train acc:=0.1
6 - val acc:=0.06
# Epoch:=10/10  - train loss:=0.2906 - val loss:=0.2904, train acc:=0.
14 - val acc:=0.04

Total time taken (in seconds):  260.94

Finished training model: mlp_on_gpu_default

**************** Testing ****************
mlp_on_gpu_default model accuracy = 9.80%
*****************************************
```



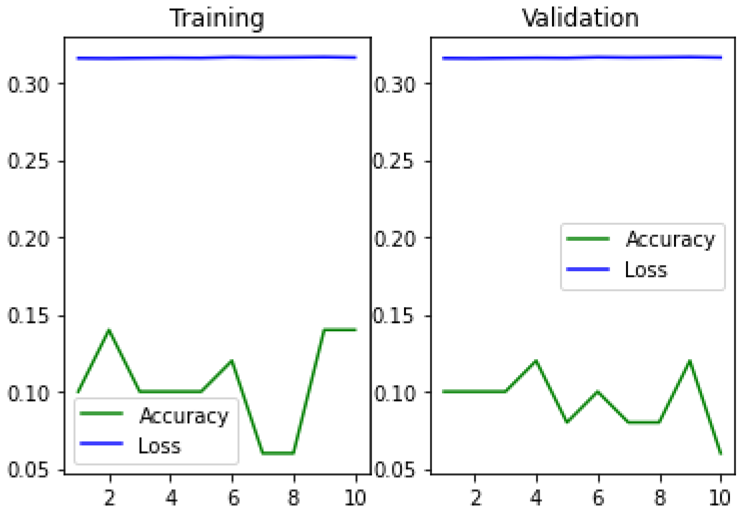
```
Count: 8, j=: 0


**************** Training model: mlp_on_gpu_default with optimizer: RM
SProp and seed: 7319 ****************

# Epoch:=1/10  - train loss:=0.2902 - val loss:=0.2912, train acc:=0.1
0 - val acc:=0.14
# Epoch:=2/10  - train loss:=0.2904 - val loss:=0.2912, train acc:=0.0
8 - val acc:=0.12
# Epoch:=3/10  - train loss:=0.2904 - val loss:=0.2912, train acc:=0.1
2 - val acc:=0.12
# Epoch:=4/10  - train loss:=0.2904 - val loss:=0.2912, train acc:=0.0
8 - val acc:=0.12
# Epoch:=5/10  - train loss:=0.2904 - val loss:=0.2912, train acc:=0.1
0 - val acc:=0.12
# Epoch:=6/10  - train loss:=0.2904 - val loss:=0.2912, train acc:=0.1
0 - val acc:=0.14
# Epoch:=7/10  - train loss:=0.2904 - val loss:=0.2912, train acc:=0.0
8 - val acc:=0.12
# Epoch:=8/10  - train loss:=0.2904 - val loss:=0.2912, train acc:=0.1
0 - val acc:=0.12
# Epoch:=9/10  - train loss:=0.2904 - val loss:=0.2912, train acc:=0.1
2 - val acc:=0.12
# Epoch:=10/10  - train loss:=0.2904 - val loss:=0.2912, train acc:=0.
12 - val acc:=0.14

Total time taken (in seconds):  260.54

Finished training model: mlp_on_gpu_default

**************** Testing ****************
mlp_on_gpu_default model accuracy = 9.58%
*****************************************
```



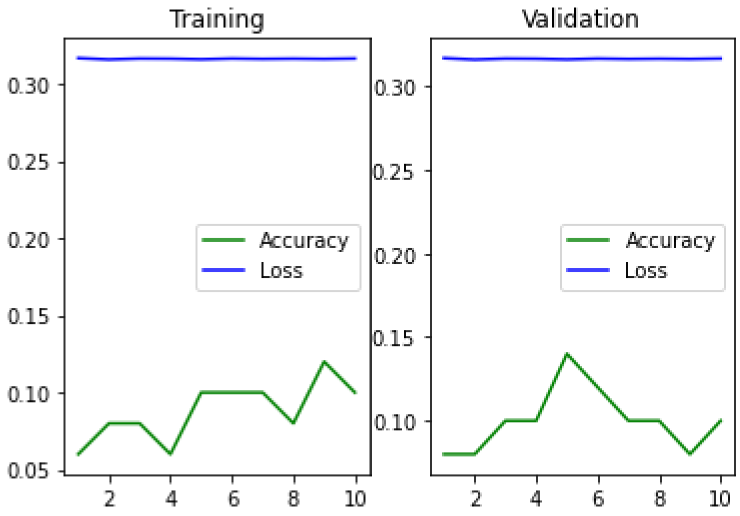
```
Count: 9, j=: 0


**************** Training model: mlp_on_gpu_default with optimizer: RM
SProp and seed: 5344 ****************

# Epoch:=1/10  - train loss:=0.2892 - val loss:=0.2892, train acc:=0.1
4 - val acc:=0.10
# Epoch:=2/10  - train loss:=0.2895 - val loss:=0.2892, train acc:=0.0
8 - val acc:=0.10
# Epoch:=3/10  - train loss:=0.2895 - val loss:=0.2892, train acc:=0.0
8 - val acc:=0.08
# Epoch:=4/10  - train loss:=0.2895 - val loss:=0.2892, train acc:=0.0
8 - val acc:=0.10
# Epoch:=5/10  - train loss:=0.2895 - val loss:=0.2892, train acc:=0.1
0 - val acc:=0.10
# Epoch:=6/10  - train loss:=0.2895 - val loss:=0.2892, train acc:=0.1
4 - val acc:=0.08
# Epoch:=7/10  - train loss:=0.2895 - val loss:=0.2892, train acc:=0.0
8 - val acc:=0.12
# Epoch:=8/10  - train loss:=0.2895 - val loss:=0.2892, train acc:=0.1
2 - val acc:=0.14
# Epoch:=9/10  - train loss:=0.2895 - val loss:=0.2892, train acc:=0.1
0 - val acc:=0.12
# Epoch:=10/10  - train loss:=0.2895 - val loss:=0.2892, train acc:=0.
10 - val acc:=0.10

Total time taken (in seconds):  261.33

Finished training model: mlp_on_gpu_default

**************** Testing ****************
mlp_on_gpu_default model accuracy = 10.10%
*****************************************
```



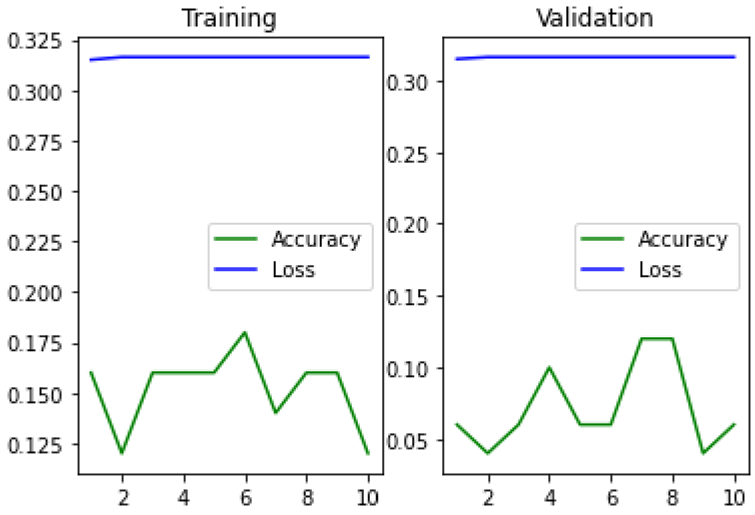
```
Count: 0, j=: 0


**************** Training model: mlp_on_gpu_RegL1 with optimizer: RMSP
rop and seed: 9383 ****************

# Epoch:=1/10  - train loss:=0.1697 - val loss:=0.2929, train acc:=0.1
0 - val acc:=0.10
# Epoch:=2/10  - train loss:=0.1696 - val loss:=0.2929, train acc:=0.1
4 - val acc:=0.10
# Epoch:=3/10  - train loss:=0.1697 - val loss:=0.2929, train acc:=0.1
0 - val acc:=0.10
# Epoch:=4/10  - train loss:=0.1698 - val loss:=0.2929, train acc:=0.1
0 - val acc:=0.12
# Epoch:=5/10  - train loss:=0.1698 - val loss:=0.2929, train acc:=0.1
0 - val acc:=0.08
# Epoch:=6/10  - train loss:=0.1700 - val loss:=0.2929, train acc:=0.1
2 - val acc:=0.10
# Epoch:=7/10  - train loss:=0.1699 - val loss:=0.2929, train acc:=0.0
6 - val acc:=0.08
# Epoch:=8/10  - train loss:=0.1700 - val loss:=0.2929, train acc:=0.0
6 - val acc:=0.08
# Epoch:=9/10  - train loss:=0.1701 - val loss:=0.2929, train acc:=0.1
4 - val acc:=0.12
# Epoch:=10/10  - train loss:=0.1699 - val loss:=0.2929, train acc:=0.
14 - val acc:=0.06

Total time taken (in seconds):  288.83

Finished training model: mlp_on_gpu_RegL1

**************** Testing ****************
mlp_on_gpu_RegL1 model accuracy = 8.92%
*****************************************
```



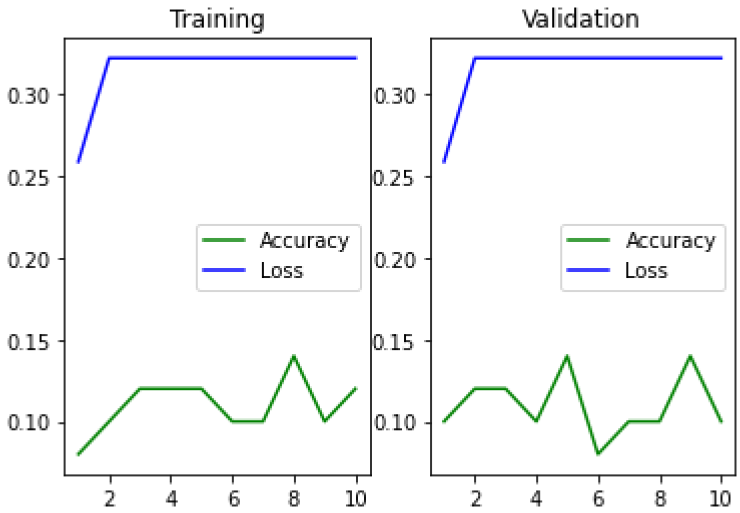
```
Count: 1, j=: 0


**************** Training model: mlp_on_gpu_RegL1 with optimizer: RMSP
rop and seed: 1332 ****************

# Epoch:=1/10  - train loss:=0.1680 - val loss:=0.0463, train acc:=0.0
6 - val acc:=0.08
# Epoch:=2/10  - train loss:=0.1675 - val loss:=0.0465, train acc:=0.0
8 - val acc:=0.08
# Epoch:=3/10  - train loss:=0.1679 - val loss:=0.0462, train acc:=0.0
8 - val acc:=0.10
# Epoch:=4/10  - train loss:=0.1678 - val loss:=0.0462, train acc:=0.0
6 - val acc:=0.10
# Epoch:=5/10  - train loss:=0.1676 - val loss:=0.0466, train acc:=0.1
0 - val acc:=0.14
# Epoch:=6/10  - train loss:=0.1679 - val loss:=0.0464, train acc:=0.1
0 - val acc:=0.12
# Epoch:=7/10  - train loss:=0.1677 - val loss:=0.0464, train acc:=0.1
0 - val acc:=0.10
# Epoch:=8/10  - train loss:=0.1678 - val loss:=0.0463, train acc:=0.0
8 - val acc:=0.10
# Epoch:=9/10  - train loss:=0.1677 - val loss:=0.0466, train acc:=0.1
2 - val acc:=0.08
# Epoch:=10/10  - train loss:=0.1679 - val loss:=0.0465, train acc:=0.
10 - val acc:=0.10

Total time taken (in seconds):  289.46

Finished training model: mlp_on_gpu_RegL1

**************** Testing ****************
mlp_on_gpu_RegL1 model accuracy = 8.92%
*****************************************
```



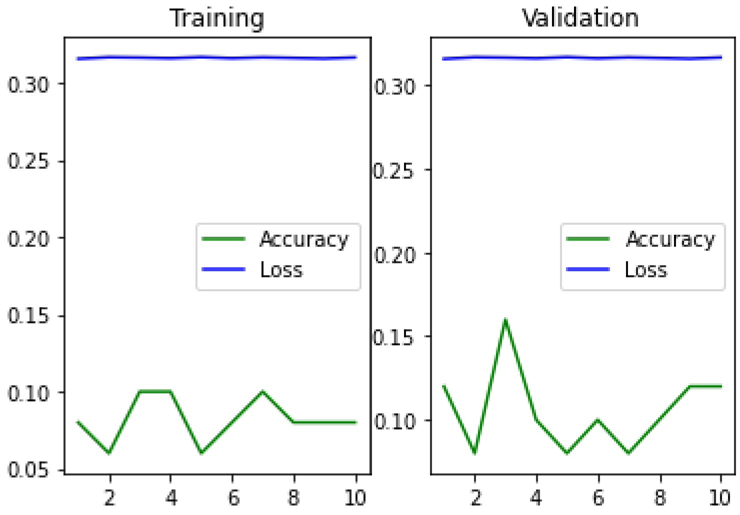
```
Count: 2, j=: 0


**************** Training model: mlp_on_gpu_RegL1 with optimizer: RMSP
rop and seed: 8948 ****************

# Epoch:=1/10  - train loss:=0.2893 - val loss:=0.2904, train acc:=0.1
6 - val acc:=0.06
# Epoch:=2/10  - train loss:=0.2906 - val loss:=0.2904, train acc:=0.1
2 - val acc:=0.04
# Epoch:=3/10  - train loss:=0.2906 - val loss:=0.2904, train acc:=0.1
6 - val acc:=0.06
# Epoch:=4/10  - train loss:=0.2906 - val loss:=0.2904, train acc:=0.1
6 - val acc:=0.10
# Epoch:=5/10  - train loss:=0.2906 - val loss:=0.2904, train acc:=0.1
6 - val acc:=0.06
# Epoch:=6/10  - train loss:=0.2906 - val loss:=0.2904, train acc:=0.1
8 - val acc:=0.06
# Epoch:=7/10  - train loss:=0.2906 - val loss:=0.2904, train acc:=0.1
4 - val acc:=0.12
# Epoch:=8/10  - train loss:=0.2906 - val loss:=0.2904, train acc:=0.1
6 - val acc:=0.12
# Epoch:=9/10  - train loss:=0.2906 - val loss:=0.2904, train acc:=0.1
6 - val acc:=0.04
# Epoch:=10/10  - train loss:=0.2906 - val loss:=0.2904, train acc:=0.
12 - val acc:=0.06

Total time taken (in seconds):  288.74

Finished training model: mlp_on_gpu_RegL1

**************** Testing ****************
mlp_on_gpu_RegL1 model accuracy = 9.80%
*****************************************
```



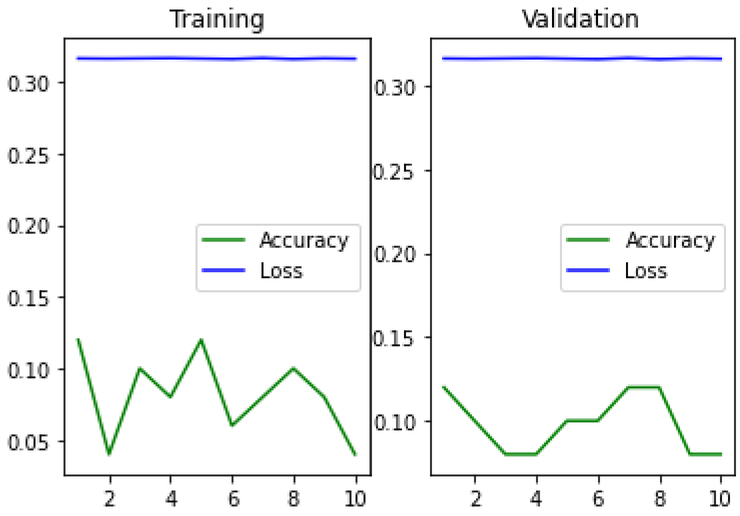
```
Count: 3, j=: 0


**************** Training model: mlp_on_gpu_RegL1 with optimizer: RMSP
rop and seed: 7195 ****************

# Epoch:=1/10  - train loss:=0.2332 - val loss:=0.2914, train acc:=0.0
8 - val acc:=0.10
# Epoch:=2/10  - train loss:=0.2902 - val loss:=0.2914, train acc:=0.1
0 - val acc:=0.12
# Epoch:=3/10  - train loss:=0.2902 - val loss:=0.2914, train acc:=0.1
2 - val acc:=0.12
# Epoch:=4/10  - train loss:=0.2902 - val loss:=0.2914, train acc:=0.1
2 - val acc:=0.10
# Epoch:=5/10  - train loss:=0.2902 - val loss:=0.2914, train acc:=0.1
2 - val acc:=0.14
# Epoch:=6/10  - train loss:=0.2902 - val loss:=0.2914, train acc:=0.1
0 - val acc:=0.08
# Epoch:=7/10  - train loss:=0.2902 - val loss:=0.2914, train acc:=0.1
0 - val acc:=0.10
# Epoch:=8/10  - train loss:=0.2902 - val loss:=0.2914, train acc:=0.1
4 - val acc:=0.10
# Epoch:=9/10  - train loss:=0.2902 - val loss:=0.2914, train acc:=0.1
0 - val acc:=0.14
# Epoch:=10/10  - train loss:=0.2902 - val loss:=0.2914, train acc:=0.
12 - val acc:=0.10

Total time taken (in seconds):  289.39

Finished training model: mlp_on_gpu_RegL1

**************** Testing ****************
mlp_on_gpu_RegL1 model accuracy = 10.09%
*****************************************
```



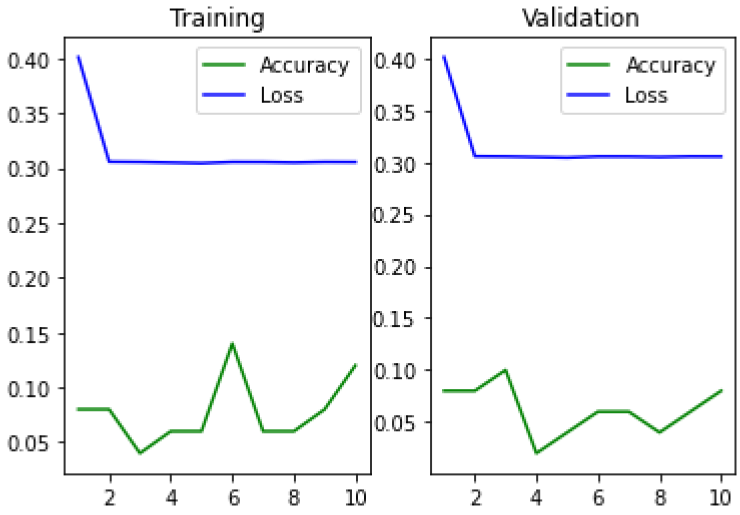
```
Count: 4, j=: 0


**************** Training model: mlp_on_gpu_RegL1 with optimizer: RMSP
rop and seed: 9616 ****************

# Epoch:=1/10  - train loss:=0.1674 - val loss:=0.0464, train acc:=0.0
8 - val acc:=0.12
# Epoch:=2/10  - train loss:=0.1679 - val loss:=0.0464, train acc:=0.0
6 - val acc:=0.08
# Epoch:=3/10  - train loss:=0.1678 - val loss:=0.0464, train acc:=0.1
0 - val acc:=0.16
# Epoch:=4/10  - train loss:=0.1676 - val loss:=0.0462, train acc:=0.1
0 - val acc:=0.10
# Epoch:=5/10  - train loss:=0.1680 - val loss:=0.0466, train acc:=0.0
6 - val acc:=0.08
# Epoch:=6/10  - train loss:=0.1676 - val loss:=0.0462, train acc:=0.0
8 - val acc:=0.10
# Epoch:=7/10  - train loss:=0.1679 - val loss:=0.0464, train acc:=0.1
0 - val acc:=0.08
# Epoch:=8/10  - train loss:=0.1677 - val loss:=0.0462, train acc:=0.0
8 - val acc:=0.10
# Epoch:=9/10  - train loss:=0.1675 - val loss:=0.0466, train acc:=0.0
8 - val acc:=0.12
# Epoch:=10/10  - train loss:=0.1678 - val loss:=0.0463, train acc:=0.
08 - val acc:=0.12

Total time taken (in seconds):  288.98

Finished training model: mlp_on_gpu_RegL1

**************** Testing ****************
mlp_on_gpu_RegL1 model accuracy = 11.35%
*****************************************
```



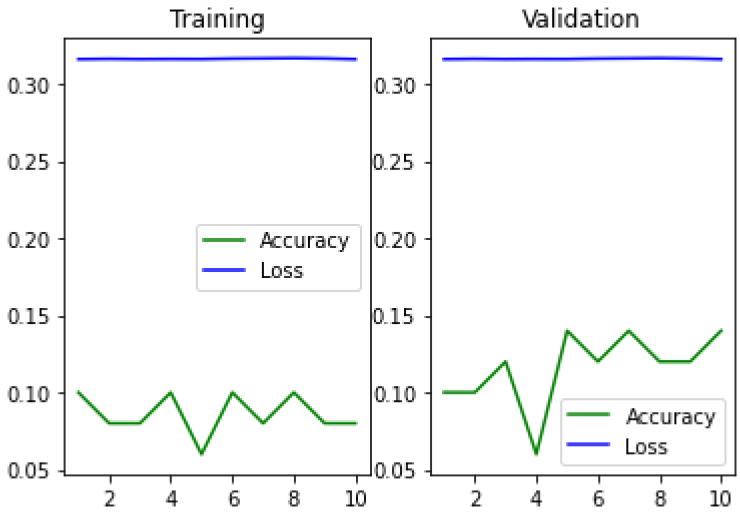
```
Count: 5, j=: 0


**************** Training model: mlp_on_gpu_RegL1 with optimizer: RMSP
rop and seed: 2112 ****************

# Epoch:=1/10  - train loss:=0.1683 - val loss:=0.2914, train acc:=0.1
2 - val acc:=0.12
# Epoch:=2/10  - train loss:=0.1683 - val loss:=0.2914, train acc:=0.0
4 - val acc:=0.10
# Epoch:=3/10  - train loss:=0.1684 - val loss:=0.2914, train acc:=0.1
0 - val acc:=0.08
# Epoch:=4/10  - train loss:=0.1685 - val loss:=0.2914, train acc:=0.0
8 - val acc:=0.08
# Epoch:=5/10  - train loss:=0.1683 - val loss:=0.2914, train acc:=0.1
2 - val acc:=0.10
# Epoch:=6/10  - train loss:=0.1681 - val loss:=0.2914, train acc:=0.0
6 - val acc:=0.10
# Epoch:=7/10  - train loss:=0.1685 - val loss:=0.2914, train acc:=0.0
8 - val acc:=0.12
# Epoch:=8/10  - train loss:=0.1681 - val loss:=0.2914, train acc:=0.1
0 - val acc:=0.12
# Epoch:=9/10  - train loss:=0.1684 - val loss:=0.2914, train acc:=0.0
8 - val acc:=0.08
# Epoch:=10/10  - train loss:=0.1682 - val loss:=0.2914, train acc:=0.
04 - val acc:=0.08

Total time taken (in seconds):  292.81

Finished training model: mlp_on_gpu_RegL1

**************** Testing ****************
mlp_on_gpu_RegL1 model accuracy = 10.09%
*****************************************
```



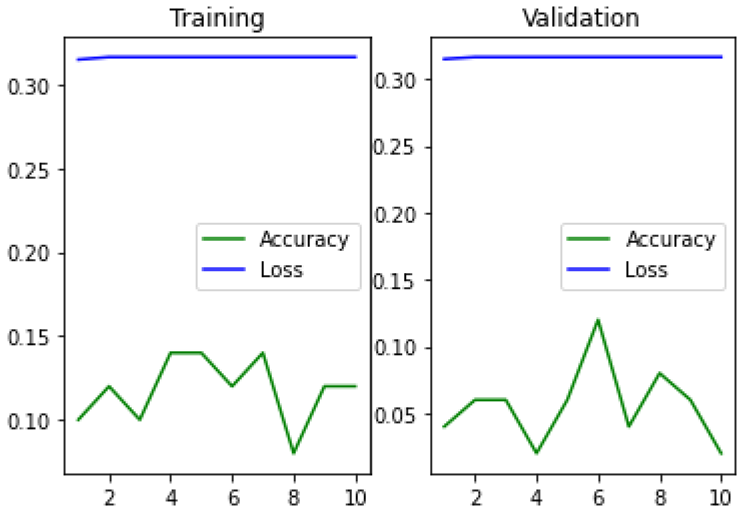
```
Count: 6, j=: 0


**************** Training model: mlp_on_gpu_RegL1 with optimizer: RMSP
rop and seed: 5047 ****************

# Epoch:=1/10  - train loss:=0.2218 - val loss:=0.2907, train acc:=0.0
8 - val acc:=0.08
# Epoch:=2/10  - train loss:=0.1691 - val loss:=0.2907, train acc:=0.0
8 - val acc:=0.08
# Epoch:=3/10  - train loss:=0.1690 - val loss:=0.2907, train acc:=0.0
4 - val acc:=0.10
# Epoch:=4/10  - train loss:=0.1687 - val loss:=0.2907, train acc:=0.0
6 - val acc:=0.02
# Epoch:=5/10  - train loss:=0.1684 - val loss:=0.2907, train acc:=0.0
6 - val acc:=0.04
# Epoch:=6/10  - train loss:=0.1690 - val loss:=0.2907, train acc:=0.1
4 - val acc:=0.06
# Epoch:=7/10  - train loss:=0.1690 - val loss:=0.2907, train acc:=0.0
6 - val acc:=0.06
# Epoch:=8/10  - train loss:=0.1687 - val loss:=0.2907, train acc:=0.0
6 - val acc:=0.04
# Epoch:=9/10  - train loss:=0.1690 - val loss:=0.2907, train acc:=0.0
8 - val acc:=0.06
# Epoch:=10/10  - train loss:=0.1689 - val loss:=0.2907, train acc:=0.
12 - val acc:=0.08

Total time taken (in seconds):  292.09

Finished training model: mlp_on_gpu_RegL1

**************** Testing ****************
mlp_on_gpu_RegL1 model accuracy = 9.82%
*****************************************
```



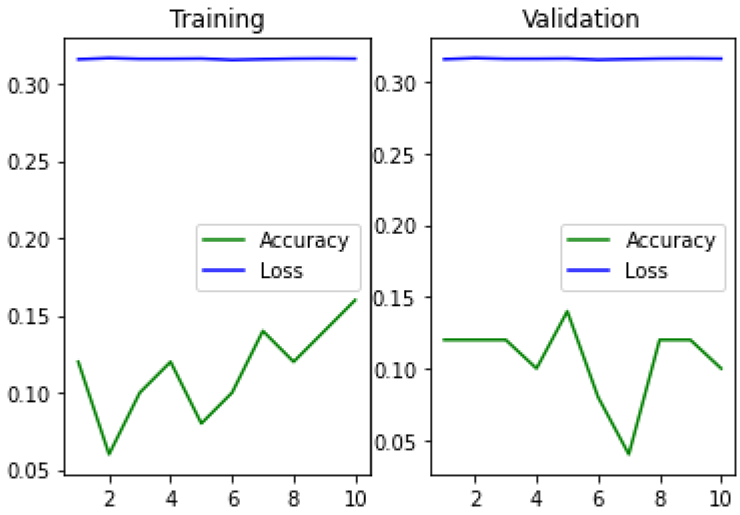
```
Count: 7, j=: 0


**************** Training model: mlp_on_gpu_RegL1 with optimizer: RMSP
rop and seed: 2013 ****************

# Epoch:=1/10  - train loss:=0.1677 - val loss:=0.0462, train acc:=0.1
0 - val acc:=0.10
# Epoch:=2/10  - train loss:=0.1678 - val loss:=0.0465, train acc:=0.0
8 - val acc:=0.10
# Epoch:=3/10  - train loss:=0.1677 - val loss:=0.0462, train acc:=0.0
8 - val acc:=0.12
# Epoch:=4/10  - train loss:=0.1677 - val loss:=0.0461, train acc:=0.1
0 - val acc:=0.06
# Epoch:=5/10  - train loss:=0.1677 - val loss:=0.0465, train acc:=0.0
6 - val acc:=0.14
# Epoch:=6/10  - train loss:=0.1679 - val loss:=0.0466, train acc:=0.1
0 - val acc:=0.12
# Epoch:=7/10  - train loss:=0.1679 - val loss:=0.0462, train acc:=0.0
8 - val acc:=0.14
# Epoch:=8/10  - train loss:=0.1680 - val loss:=0.0463, train acc:=0.1
0 - val acc:=0.12
# Epoch:=9/10  - train loss:=0.1679 - val loss:=0.0464, train acc:=0.0
8 - val acc:=0.12
# Epoch:=10/10  - train loss:=0.1677 - val loss:=0.0466, train acc:=0.
08 - val acc:=0.14

Total time taken (in seconds):  291.15

Finished training model: mlp_on_gpu_RegL1

**************** Testing ****************
mlp_on_gpu_RegL1 model accuracy = 10.10%
*****************************************
```



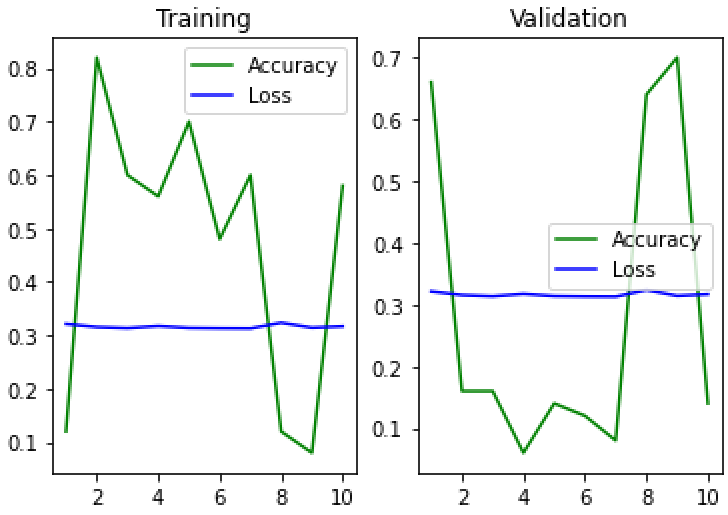
```
Count: 8, j=: 0


**************** Training model: mlp_on_gpu_RegL1 with optimizer: RMSP
rop and seed: 1137 ****************

# Epoch:=1/10  - train loss:=0.2897 - val loss:=0.2907, train acc:=0.1
0 - val acc:=0.04
# Epoch:=2/10  - train loss:=0.2910 - val loss:=0.2907, train acc:=0.1
2 - val acc:=0.06
# Epoch:=3/10  - train loss:=0.2910 - val loss:=0.2907, train acc:=0.1
0 - val acc:=0.06
# Epoch:=4/10  - train loss:=0.2910 - val loss:=0.2907, train acc:=0.1
4 - val acc:=0.02
# Epoch:=5/10  - train loss:=0.2910 - val loss:=0.2907, train acc:=0.1
4 - val acc:=0.06
# Epoch:=6/10  - train loss:=0.2910 - val loss:=0.2907, train acc:=0.1
2 - val acc:=0.12
# Epoch:=7/10  - train loss:=0.2910 - val loss:=0.2907, train acc:=0.1
4 - val acc:=0.04
# Epoch:=8/10  - train loss:=0.2910 - val loss:=0.2907, train acc:=0.0
8 - val acc:=0.08
# Epoch:=9/10  - train loss:=0.2910 - val loss:=0.2907, train acc:=0.1
2 - val acc:=0.06
# Epoch:=10/10  - train loss:=0.2910 - val loss:=0.2907, train acc:=0.
12 - val acc:=0.02

Total time taken (in seconds):  290.32

Finished training model: mlp_on_gpu_RegL1

**************** Testing ****************
mlp_on_gpu_RegL1 model accuracy = 9.82%
*****************************************
```



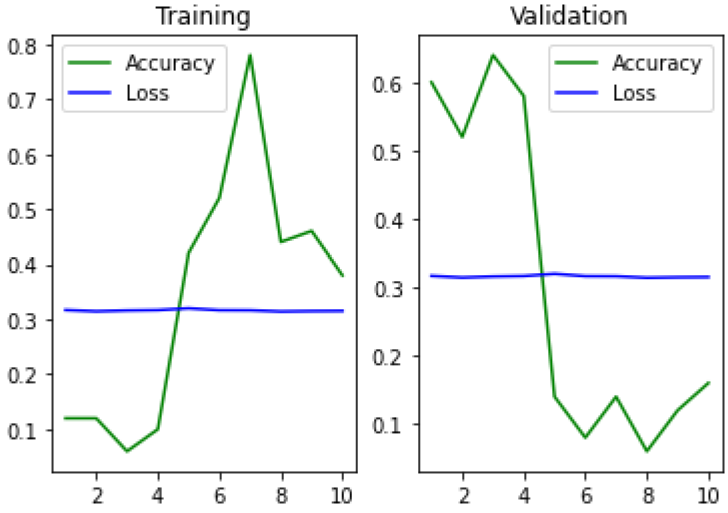
```
Count: 9, j=: 0


**************** Training model: mlp_on_gpu_RegL1 with optimizer: RMSP
rop and seed: 5413 ****************

# Epoch:=1/10  - train loss:=0.1678 - val loss:=0.0462, train acc:=0.1
2 - val acc:=0.12
# Epoch:=2/10  - train loss:=0.1682 - val loss:=0.0465, train acc:=0.0
6 - val acc:=0.12
# Epoch:=3/10  - train loss:=0.1679 - val loss:=0.0462, train acc:=0.1
0 - val acc:=0.12
# Epoch:=4/10  - train loss:=0.1679 - val loss:=0.0465, train acc:=0.1
2 - val acc:=0.10
# Epoch:=5/10  - train loss:=0.1680 - val loss:=0.0462, train acc:=0.0
8 - val acc:=0.14
# Epoch:=6/10  - train loss:=0.1676 - val loss:=0.0465, train acc:=0.1
0 - val acc:=0.08
# Epoch:=7/10  - train loss:=0.1678 - val loss:=0.0464, train acc:=0.1
4 - val acc:=0.04
# Epoch:=8/10  - train loss:=0.1680 - val loss:=0.0463, train acc:=0.1
2 - val acc:=0.12
# Epoch:=9/10  - train loss:=0.1680 - val loss:=0.0463, train acc:=0.1
4 - val acc:=0.12
# Epoch:=10/10  - train loss:=0.1680 - val loss:=0.0464, train acc:=0.
16 - val acc:=0.10

Total time taken (in seconds):  289.70

Finished training model: mlp_on_gpu_RegL1

**************** Testing ****************
mlp_on_gpu_RegL1 model accuracy = 11.35%
*****************************************
```



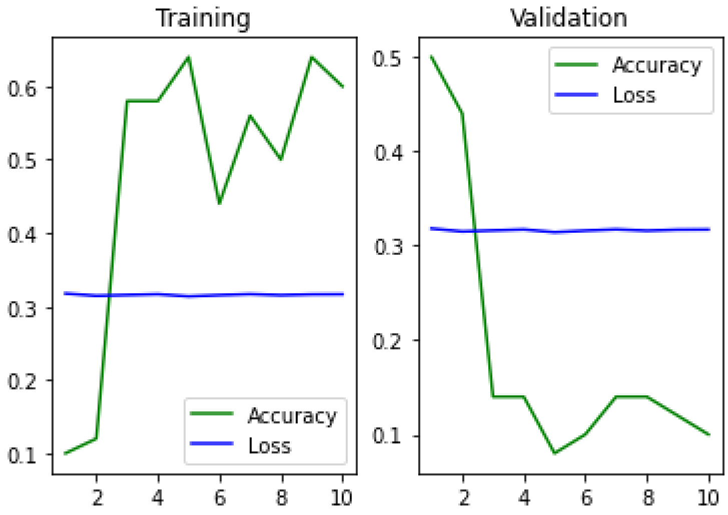
```
Count: 0, j=: 0


**************** Training model: mlp_on_gpu_RegL2 with optimizer: RMSP
rop and seed: 8417 ****************

# Epoch:=1/10  - train loss:=0.1611 - val loss:=0.0210, train acc:=0.1
2 - val acc:=0.66
# Epoch:=2/10  - train loss:=0.1582 - val loss:=0.2904, train acc:=0.8
2 - val acc:=0.16
# Epoch:=3/10  - train loss:=0.1572 - val loss:=0.2912, train acc:=0.6
0 - val acc:=0.16
# Epoch:=4/10  - train loss:=0.1591 - val loss:=0.2881, train acc:=0.5
6 - val acc:=0.06
# Epoch:=5/10  - train loss:=0.1574 - val loss:=0.2892, train acc:=0.7
0 - val acc:=0.14
# Epoch:=6/10  - train loss:=0.1571 - val loss:=0.2904, train acc:=0.4
8 - val acc:=0.12
# Epoch:=7/10  - train loss:=0.1569 - val loss:=0.2904, train acc:=0.6
0 - val acc:=0.08
# Epoch:=8/10  - train loss:=0.1623 - val loss:=0.0172, train acc:=0.1
2 - val acc:=0.64
# Epoch:=9/10  - train loss:=0.1576 - val loss:=0.0197, train acc:=0.0
8 - val acc:=0.70
# Epoch:=10/10  - train loss:=0.1589 - val loss:=0.2904, train acc:=0.
58 - val acc:=0.14

Total time taken (in seconds):  310.72

Finished training model: mlp_on_gpu_RegL2

**************** Testing ****************
mlp_on_gpu_RegL2 model accuracy = 10.32%
*****************************************
```



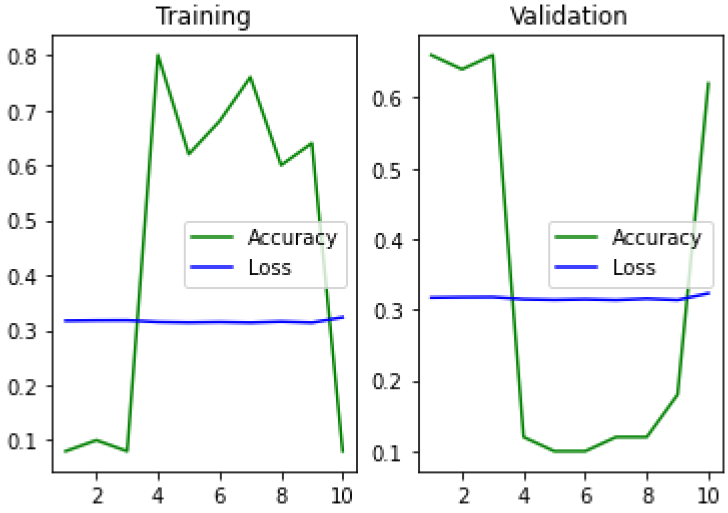
```
Count: 1, j=: 0


**************** Training model: mlp_on_gpu_RegL2 with optimizer: RMSP
rop and seed: 3147 ****************

# Epoch:=1/10  - train loss:=0.1582 - val loss:=0.0215, train acc:=0.1
2 - val acc:=0.60
# Epoch:=2/10  - train loss:=0.1570 - val loss:=0.0248, train acc:=0.1
2 - val acc:=0.52
# Epoch:=3/10  - train loss:=0.1578 - val loss:=0.0220, train acc:=0.0
6 - val acc:=0.64
# Epoch:=4/10  - train loss:=0.1581 - val loss:=0.0231, train acc:=0.1
0 - val acc:=0.58
# Epoch:=5/10  - train loss:=0.1598 - val loss:=0.2912, train acc:=0.4
2 - val acc:=0.14
# Epoch:=6/10  - train loss:=0.1580 - val loss:=0.2887, train acc:=0.5
2 - val acc:=0.08
# Epoch:=7/10  - train loss:=0.1580 - val loss:=0.2892, train acc:=0.7
8 - val acc:=0.14
# Epoch:=8/10  - train loss:=0.1569 - val loss:=0.2904, train acc:=0.4
4 - val acc:=0.06
# Epoch:=9/10  - train loss:=0.1572 - val loss:=0.2904, train acc:=0.4
6 - val acc:=0.12
# Epoch:=10/10  - train loss:=0.1574 - val loss:=0.2912, train acc:=0.
38 - val acc:=0.16

Total time taken (in seconds):  310.10

Finished training model: mlp_on_gpu_RegL2

**************** Testing ****************
mlp_on_gpu_RegL2 model accuracy = 9.58%
*****************************************
```



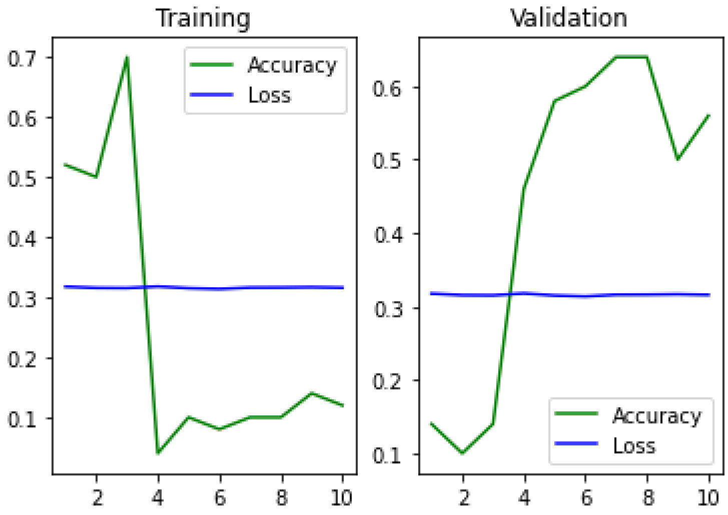
```
Count: 2, j=: 0


**************** Training model: mlp_on_gpu_RegL2 with optimizer: RMSP
rop and seed: 4129 ****************

# Epoch:=1/10  - train loss:=0.1592 - val loss:=0.0225, train acc:=0.1
0 - val acc:=0.50
# Epoch:=2/10  - train loss:=0.1576 - val loss:=0.0235, train acc:=0.1
2 - val acc:=0.44
# Epoch:=3/10  - train loss:=0.1582 - val loss:=0.2904, train acc:=0.5
8 - val acc:=0.14
# Epoch:=4/10  - train loss:=0.1587 - val loss:=0.2904, train acc:=0.5
8 - val acc:=0.14
# Epoch:=5/10  - train loss:=0.1572 - val loss:=0.2898, train acc:=0.6
4 - val acc:=0.08
# Epoch:=6/10  - train loss:=0.1581 - val loss:=0.2904, train acc:=0.4
4 - val acc:=0.10
# Epoch:=7/10  - train loss:=0.1588 - val loss:=0.2892, train acc:=0.5
6 - val acc:=0.14
# Epoch:=8/10  - train loss:=0.1580 - val loss:=0.2912, train acc:=0.5
0 - val acc:=0.14
# Epoch:=9/10  - train loss:=0.1586 - val loss:=0.2892, train acc:=0.6
4 - val acc:=0.12
# Epoch:=10/10  - train loss:=0.1587 - val loss:=0.2881, train acc:=0.
60 - val acc:=0.10

Total time taken (in seconds):  310.17

Finished training model: mlp_on_gpu_RegL2

**************** Testing ****************
mlp_on_gpu_RegL2 model accuracy = 11.35%
*****************************************
```



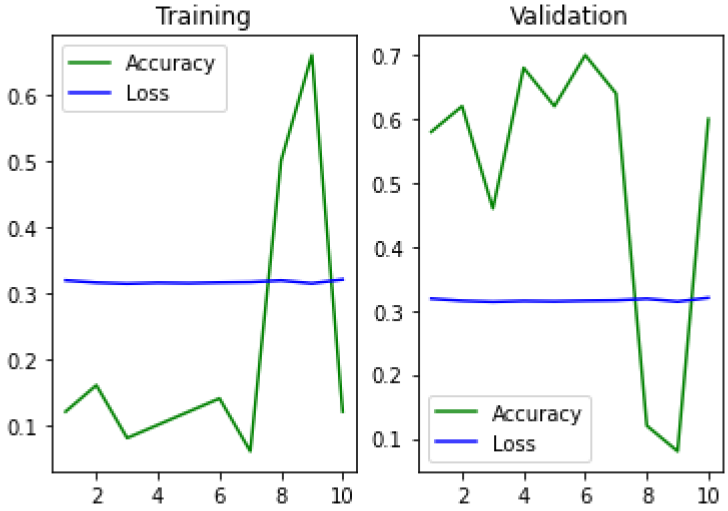
```
Count: 3, j=: 0


**************** Training model: mlp_on_gpu_RegL2 with optimizer: RMSP
rop and seed: 1725 ****************

# Epoch:=1/10  - train loss:=0.1591 - val loss:=0.0199, train acc:=0.0
8 - val acc:=0.66
# Epoch:=2/10  - train loss:=0.1594 - val loss:=0.0205, train acc:=0.1
0 - val acc:=0.64
# Epoch:=3/10  - train loss:=0.1595 - val loss:=0.0204, train acc:=0.0
8 - val acc:=0.66
# Epoch:=4/10  - train loss:=0.1580 - val loss:=0.2904, train acc:=0.8
0 - val acc:=0.12
# Epoch:=5/10  - train loss:=0.1575 - val loss:=0.2892, train acc:=0.6
2 - val acc:=0.10
# Epoch:=6/10  - train loss:=0.1579 - val loss:=0.2904, train acc:=0.6
8 - val acc:=0.10
# Epoch:=7/10  - train loss:=0.1574 - val loss:=0.2904, train acc:=0.7
6 - val acc:=0.12
# Epoch:=8/10  - train loss:=0.1584 - val loss:=0.2904, train acc:=0.6
0 - val acc:=0.12
# Epoch:=9/10  - train loss:=0.1574 - val loss:=0.2904, train acc:=0.6
4 - val acc:=0.18
# Epoch:=10/10  - train loss:=0.1622 - val loss:=0.0191, train acc:=0.
08 - val acc:=0.62

Total time taken (in seconds):  310.32

Finished training model: mlp_on_gpu_RegL2

**************** Testing ****************
mlp_on_gpu_RegL2 model accuracy = 67.36%
*****************************************
```



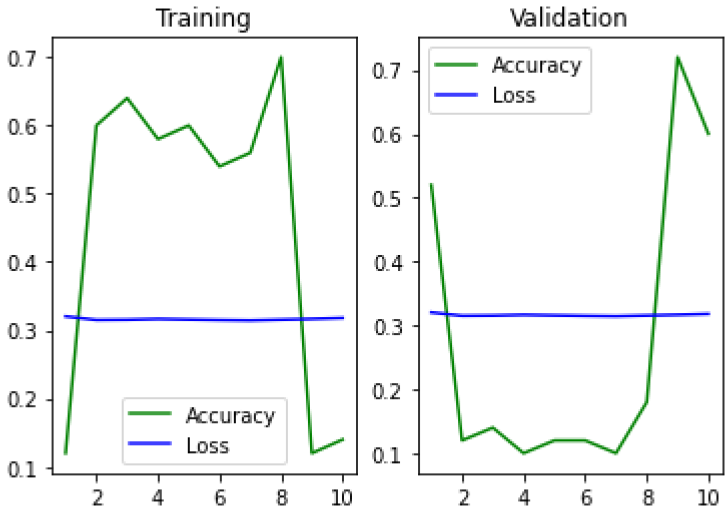
```
Count: 4, j=: 0


**************** Training model: mlp_on_gpu_RegL2 with optimizer: RMSP
rop and seed: 9844 ****************

# Epoch:=1/10  - train loss:=0.1587 - val loss:=0.2912, train acc:=0.5
2 - val acc:=0.14
# Epoch:=2/10  - train loss:=0.1577 - val loss:=0.2872, train acc:=0.5
0 - val acc:=0.10
# Epoch:=3/10  - train loss:=0.1575 - val loss:=0.2892, train acc:=0.7
0 - val acc:=0.14
# Epoch:=4/10  - train loss:=0.1589 - val loss:=0.0242, train acc:=0.0
4 - val acc:=0.46
# Epoch:=5/10  - train loss:=0.1574 - val loss:=0.0218, train acc:=0.1
0 - val acc:=0.58
# Epoch:=6/10  - train loss:=0.1568 - val loss:=0.0217, train acc:=0.0
8 - val acc:=0.60
# Epoch:=7/10  - train loss:=0.1580 - val loss:=0.0201, train acc:=0.1
0 - val acc:=0.64
# Epoch:=8/10  - train loss:=0.1580 - val loss:=0.0203, train acc:=0.1
0 - val acc:=0.64
# Epoch:=9/10  - train loss:=0.1583 - val loss:=0.0237, train acc:=0.1
4 - val acc:=0.50
# Epoch:=10/10  - train loss:=0.1579 - val loss:=0.0214, train acc:=0.
12 - val acc:=0.56

Total time taken (in seconds):  309.37

Finished training model: mlp_on_gpu_RegL2

**************** Testing ****************
mlp_on_gpu_RegL2 model accuracy = 59.00%
*****************************************
```



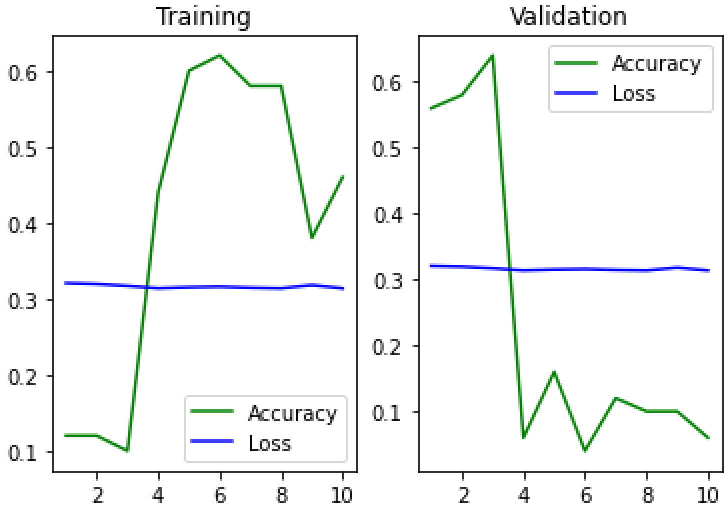
```
Count: 5, j=: 0


**************** Training model: mlp_on_gpu_RegL2 with optimizer: RMSP
rop and seed: 7336 ****************

# Epoch:=1/10  - train loss:=0.1598 - val loss:=0.0218, train acc:=0.1
2 - val acc:=0.58
# Epoch:=2/10  - train loss:=0.1581 - val loss:=0.0197, train acc:=0.1
6 - val acc:=0.62
# Epoch:=3/10  - train loss:=0.1573 - val loss:=0.0205, train acc:=0.0
8 - val acc:=0.46
# Epoch:=4/10  - train loss:=0.1580 - val loss:=0.0177, train acc:=0.1
0 - val acc:=0.68
# Epoch:=5/10  - train loss:=0.1576 - val loss:=0.0189, train acc:=0.1
2 - val acc:=0.62
# Epoch:=6/10  - train loss:=0.1581 - val loss:=0.0190, train acc:=0.1
4 - val acc:=0.70
# Epoch:=7/10  - train loss:=0.1585 - val loss:=0.0214, train acc:=0.0
6 - val acc:=0.64
# Epoch:=8/10  - train loss:=0.1597 - val loss:=0.2881, train acc:=0.5
0 - val acc:=0.12
# Epoch:=9/10  - train loss:=0.1575 - val loss:=0.2904, train acc:=0.6
6 - val acc:=0.08
# Epoch:=10/10  - train loss:=0.1605 - val loss:=0.0207, train acc:=0.
12 - val acc:=0.60

Total time taken (in seconds):  309.73

Finished training model: mlp_on_gpu_RegL2

**************** Testing ****************
mlp_on_gpu_RegL2 model accuracy = 63.55%
*****************************************
```



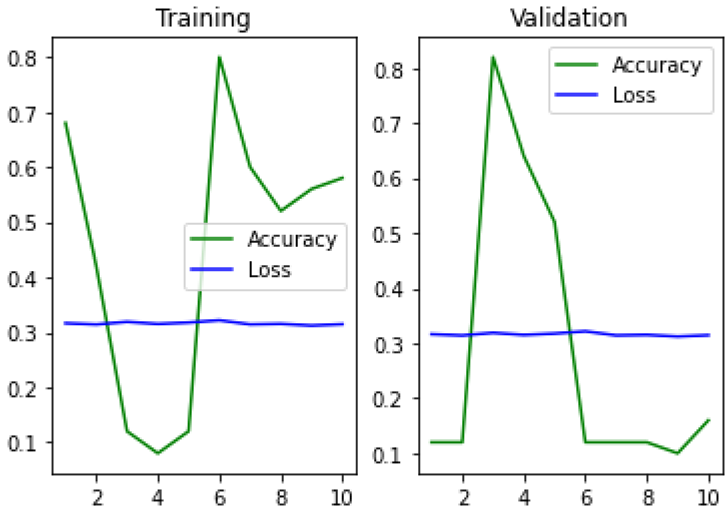
```
Count: 6, j=: 0


**************** Training model: mlp_on_gpu_RegL2 with optimizer: RMSP
rop and seed: 7014 ****************

# Epoch:=1/10  - train loss:=0.1600 - val loss:=0.0217, train acc:=0.1
2 - val acc:=0.52
# Epoch:=2/10  - train loss:=0.1573 - val loss:=0.2904, train acc:=0.6
0 - val acc:=0.12
# Epoch:=3/10  - train loss:=0.1575 - val loss:=0.2898, train acc:=0.6
4 - val acc:=0.14
# Epoch:=4/10  - train loss:=0.1581 - val loss:=0.2892, train acc:=0.5
8 - val acc:=0.10
# Epoch:=5/10  - train loss:=0.1577 - val loss:=0.2904, train acc:=0.6
0 - val acc:=0.12
# Epoch:=6/10  - train loss:=0.1573 - val loss:=0.2904, train acc:=0.5
4 - val acc:=0.12
# Epoch:=7/10  - train loss:=0.1570 - val loss:=0.2881, train acc:=0.5
6 - val acc:=0.10
# Epoch:=8/10  - train loss:=0.1576 - val loss:=0.2892, train acc:=0.7
0 - val acc:=0.18
# Epoch:=9/10  - train loss:=0.1581 - val loss:=0.0962, train acc:=0.1
2 - val acc:=0.72
# Epoch:=10/10  - train loss:=0.1588 - val loss:=0.0211, train acc:=0.
14 - val acc:=0.60

Total time taken (in seconds):  308.96

Finished training model: mlp_on_gpu_RegL2

**************** Testing ****************
mlp_on_gpu_RegL2 model accuracy = 64.98%
*****************************************
```



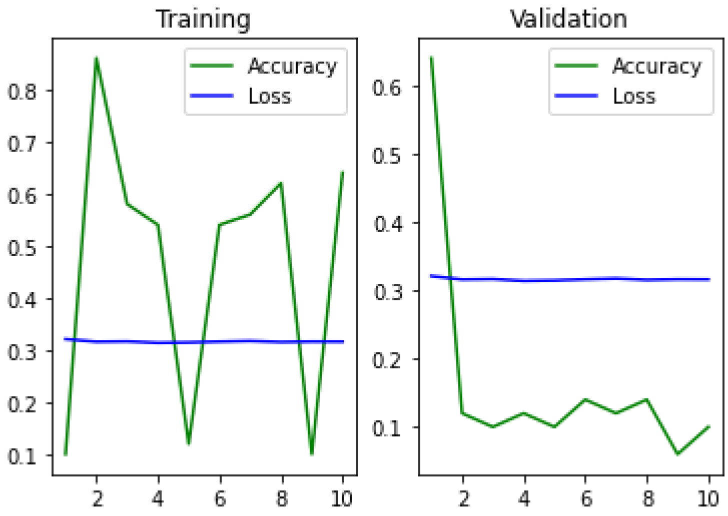
```
Count: 7, j=: 0


**************** Training model: mlp_on_gpu_RegL2 with optimizer: RMSP
rop and seed: 4800 ****************

# Epoch:=1/10  - train loss:=0.1607 - val loss:=0.0214, train acc:=0.1
2 - val acc:=0.56
# Epoch:=2/10  - train loss:=0.1601 - val loss:=0.0174, train acc:=0.1
2 - val acc:=0.58
# Epoch:=3/10  - train loss:=0.1588 - val loss:=0.0202, train acc:=0.1
0 - val acc:=0.64
# Epoch:=4/10  - train loss:=0.1573 - val loss:=0.2904, train acc:=0.4
4 - val acc:=0.06
# Epoch:=5/10  - train loss:=0.1579 - val loss:=0.2912, train acc:=0.6
0 - val acc:=0.16
# Epoch:=6/10  - train loss:=0.1583 - val loss:=0.2904, train acc:=0.6
2 - val acc:=0.04
# Epoch:=7/10  - train loss:=0.1576 - val loss:=0.2892, train acc:=0.5
8 - val acc:=0.12
# Epoch:=8/10  - train loss:=0.1572 - val loss:=0.2892, train acc:=0.5
8 - val acc:=0.10
# Epoch:=9/10  - train loss:=0.1594 - val loss:=0.2904, train acc:=0.3
8 - val acc:=0.10
# Epoch:=10/10  - train loss:=0.1572 - val loss:=0.2904, train acc:=0.
46 - val acc:=0.06

Total time taken (in seconds):  308.67

Finished training model: mlp_on_gpu_RegL2

**************** Testing ****************
mlp_on_gpu_RegL2 model accuracy = 9.80%
*****************************************
```



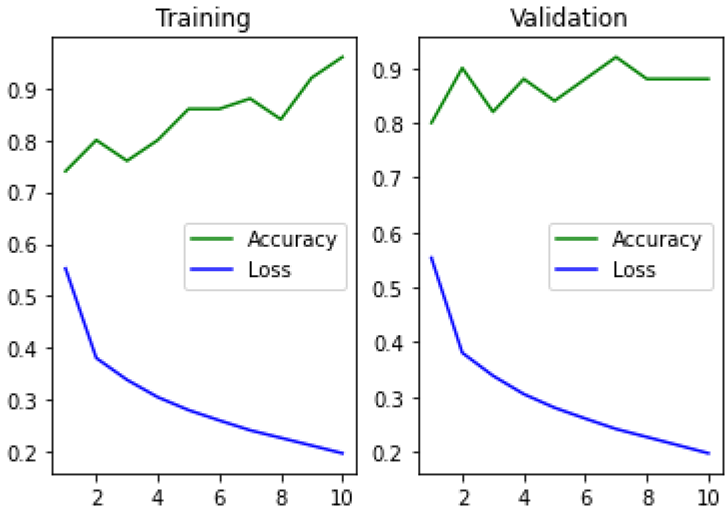
```
Count: 8, j=: 0


**************** Training model: mlp_on_gpu_RegL2 with optimizer: RMSP
rop and seed: 9250 ****************

# Epoch:=1/10  - train loss:=0.1580 - val loss:=0.2904, train acc:=0.6
8 - val acc:=0.12
# Epoch:=2/10  - train loss:=0.1569 - val loss:=0.2904, train acc:=0.4
2 - val acc:=0.12
# Epoch:=3/10  - train loss:=0.1593 - val loss:=0.0578, train acc:=0.1
2 - val acc:=0.82
# Epoch:=4/10  - train loss:=0.1574 - val loss:=0.0201, train acc:=0.0
8 - val acc:=0.64
# Epoch:=5/10  - train loss:=0.1587 - val loss:=0.0200, train acc:=0.1
2 - val acc:=0.52
# Epoch:=6/10  - train loss:=0.1607 - val loss:=0.2892, train acc:=0.8
0 - val acc:=0.12
# Epoch:=7/10  - train loss:=0.1570 - val loss:=0.2892, train acc:=0.6
0 - val acc:=0.12
# Epoch:=8/10  - train loss:=0.1575 - val loss:=0.2904, train acc:=0.5
2 - val acc:=0.12
# Epoch:=9/10  - train loss:=0.1559 - val loss:=0.2881, train acc:=0.5
6 - val acc:=0.10
# Epoch:=10/10  - train loss:=0.1572 - val loss:=0.2904, train acc:=0.
58 - val acc:=0.16

Total time taken (in seconds):  308.42

Finished training model: mlp_on_gpu_RegL2

**************** Testing ****************
mlp_on_gpu_RegL2 model accuracy = 10.32%
*****************************************
```



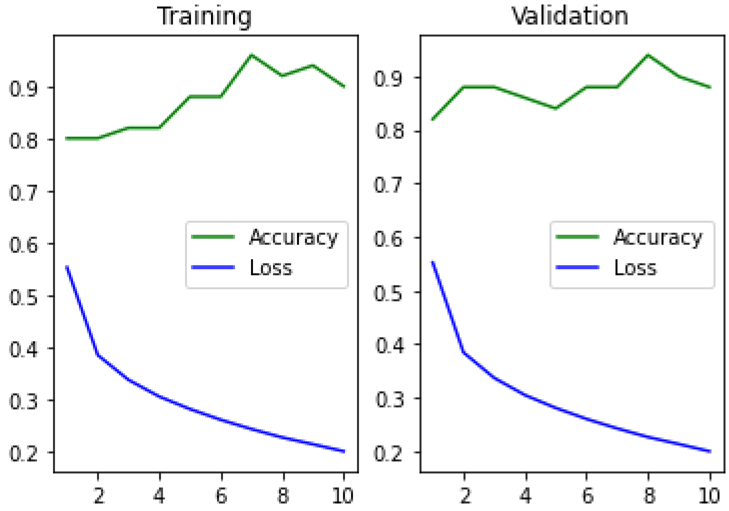
```
Count: 9, j=: 0


**************** Training model: mlp_on_gpu_RegL2 with optimizer: RMSP
rop and seed: 3319 ****************

# Epoch:=1/10  - train loss:=0.1596 - val loss:=0.0175, train acc:=0.1
0 - val acc:=0.64
# Epoch:=2/10  - train loss:=0.1571 - val loss:=0.2904, train acc:=0.8
6 - val acc:=0.12
# Epoch:=3/10  - train loss:=0.1574 - val loss:=0.2892, train acc:=0.5
8 - val acc:=0.10
# Epoch:=4/10  - train loss:=0.1561 - val loss:=0.2904, train acc:=0.5
4 - val acc:=0.12
# Epoch:=5/10  - train loss:=0.1564 - val loss:=0.2914, train acc:=0.1
2 - val acc:=0.10
# Epoch:=6/10  - train loss:=0.1572 - val loss:=0.2892, train acc:=0.5
4 - val acc:=0.14
# Epoch:=7/10  - train loss:=0.1578 - val loss:=0.2892, train acc:=0.5
6 - val acc:=0.12
# Epoch:=8/10  - train loss:=0.1568 - val loss:=0.2904, train acc:=0.6
2 - val acc:=0.14
# Epoch:=9/10  - train loss:=0.1572 - val loss:=0.2929, train acc:=0.1
0 - val acc:=0.06
# Epoch:=10/10  - train loss:=0.1572 - val loss:=0.2892, train acc:=0.
64 - val acc:=0.10

Total time taken (in seconds):  308.49

Finished training model: mlp_on_gpu_RegL2

**************** Testing ****************
mlp_on_gpu_RegL2 model accuracy = 10.10%
*****************************************
```



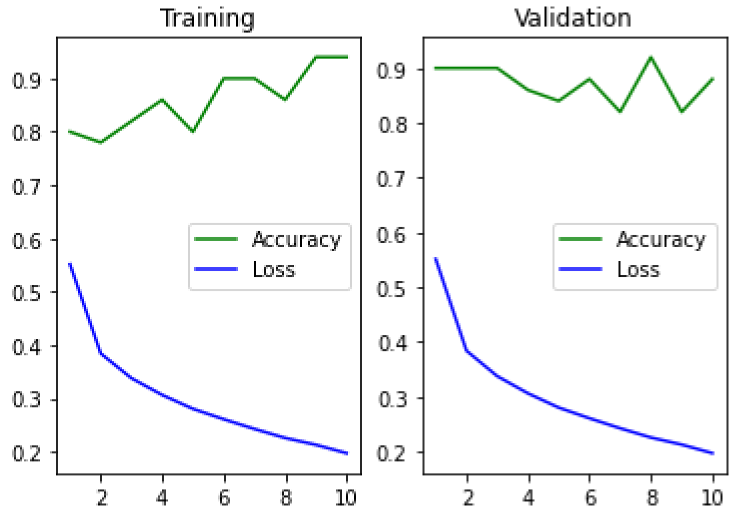
```
Data Split:
X_train: (50000, 784), y_train: (50000, 10)
X_test : (10000, 784), y_test : (10000, 10)
X_val  : (10000, 784), y_val  : (10000, 10)
Count: 0, j=: 1


**************** Training model: mlp_on_gpu_default with optimizer: SG
D and seed: 2417 ****************

# Epoch:=1/10  - train loss:=0.0109 - val loss:=0.0084, train acc:=0.7
4 - val acc:=0.80
# Epoch:=2/10  - train loss:=0.0075 - val loss:=0.0075, train acc:=0.8
0 - val acc:=0.90
# Epoch:=3/10  - train loss:=0.0067 - val loss:=0.0078, train acc:=0.7
6 - val acc:=0.82
# Epoch:=4/10  - train loss:=0.0060 - val loss:=0.0069, train acc:=0.8
0 - val acc:=0.88
# Epoch:=5/10  - train loss:=0.0055 - val loss:=0.0066, train acc:=0.8
6 - val acc:=0.84
# Epoch:=6/10  - train loss:=0.0052 - val loss:=0.0067, train acc:=0.8
6 - val acc:=0.88
# Epoch:=7/10  - train loss:=0.0048 - val loss:=0.0070, train acc:=0.8
8 - val acc:=0.92
# Epoch:=8/10  - train loss:=0.0045 - val loss:=0.0064, train acc:=0.8
4 - val acc:=0.88
# Epoch:=9/10  - train loss:=0.0042 - val loss:=0.0065, train acc:=0.9
2 - val acc:=0.88
# Epoch:=10/10  - train loss:=0.0039 - val loss:=0.0068, train acc:=0.
96 - val acc:=0.88

Total time taken (in seconds):  194.21

Finished training model: mlp_on_gpu_default

**************** Testing ****************
mlp_on_gpu_default model accuracy = 87.84%
*****************************************
```



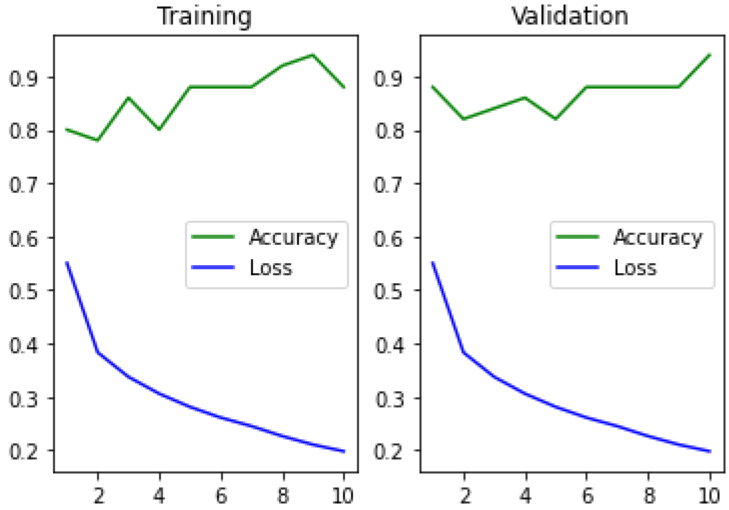
```
Count: 1, j=: 1


**************** Training model: mlp_on_gpu_default with optimizer: SG
D and seed: 9992 ****************

# Epoch:=1/10  - train loss:=0.0110 - val loss:=0.0082, train acc:=0.8
0 - val acc:=0.82
# Epoch:=2/10  - train loss:=0.0076 - val loss:=0.0074, train acc:=0.8
0 - val acc:=0.88
# Epoch:=3/10  - train loss:=0.0067 - val loss:=0.0079, train acc:=0.8
2 - val acc:=0.88
# Epoch:=4/10  - train loss:=0.0060 - val loss:=0.0075, train acc:=0.8
2 - val acc:=0.86
# Epoch:=5/10  - train loss:=0.0056 - val loss:=0.0067, train acc:=0.8
8 - val acc:=0.84
# Epoch:=6/10  - train loss:=0.0051 - val loss:=0.0068, train acc:=0.8
8 - val acc:=0.88
# Epoch:=7/10  - train loss:=0.0048 - val loss:=0.0065, train acc:=0.9
6 - val acc:=0.88
# Epoch:=8/10  - train loss:=0.0045 - val loss:=0.0067, train acc:=0.9
2 - val acc:=0.94
# Epoch:=9/10  - train loss:=0.0042 - val loss:=0.0066, train acc:=0.9
4 - val acc:=0.90
# Epoch:=10/10  - train loss:=0.0039 - val loss:=0.0069, train acc:=0.
90 - val acc:=0.88

Total time taken (in seconds):  193.39

Finished training model: mlp_on_gpu_default

**************** Testing ****************
mlp_on_gpu_default model accuracy = 87.41%
*****************************************
```



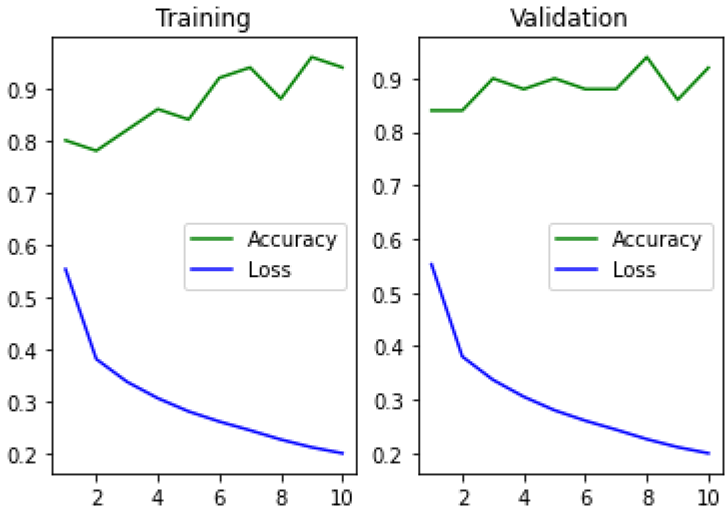
```
Count: 2, j=: 1


**************** Training model: mlp_on_gpu_default with optimizer: SG
D and seed: 1255 ****************

# Epoch:=1/10  - train loss:=0.0110 - val loss:=0.0083, train acc:=0.8
0 - val acc:=0.90
# Epoch:=2/10  - train loss:=0.0076 - val loss:=0.0074, train acc:=0.7
8 - val acc:=0.90
# Epoch:=3/10  - train loss:=0.0067 - val loss:=0.0073, train acc:=0.8
2 - val acc:=0.90
# Epoch:=4/10  - train loss:=0.0061 - val loss:=0.0066, train acc:=0.8
6 - val acc:=0.86
# Epoch:=5/10  - train loss:=0.0056 - val loss:=0.0071, train acc:=0.8
0 - val acc:=0.84
# Epoch:=6/10  - train loss:=0.0052 - val loss:=0.0062, train acc:=0.9
0 - val acc:=0.88
# Epoch:=7/10  - train loss:=0.0048 - val loss:=0.0065, train acc:=0.9
0 - val acc:=0.82
# Epoch:=8/10  - train loss:=0.0045 - val loss:=0.0066, train acc:=0.8
6 - val acc:=0.92
# Epoch:=9/10  - train loss:=0.0042 - val loss:=0.0062, train acc:=0.9
4 - val acc:=0.82
# Epoch:=10/10  - train loss:=0.0039 - val loss:=0.0067, train acc:=0.
94 - val acc:=0.88

Total time taken (in seconds):  193.30

Finished training model: mlp_on_gpu_default

**************** Testing ****************
mlp_on_gpu_default model accuracy = 88.03%
*****************************************
```



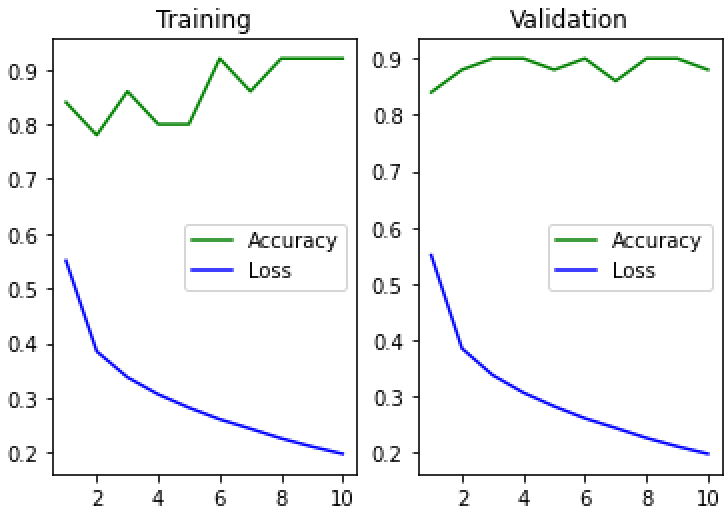
```
Count: 3, j=: 1


**************** Training model: mlp_on_gpu_default with optimizer: SG
D and seed: 8492 ****************

# Epoch:=1/10  - train loss:=0.0109 - val loss:=0.0084, train acc:=0.8
0 - val acc:=0.88
# Epoch:=2/10  - train loss:=0.0076 - val loss:=0.0078, train acc:=0.7
8 - val acc:=0.82
# Epoch:=3/10  - train loss:=0.0067 - val loss:=0.0069, train acc:=0.8
6 - val acc:=0.84
# Epoch:=4/10  - train loss:=0.0061 - val loss:=0.0067, train acc:=0.8
0 - val acc:=0.86
# Epoch:=5/10  - train loss:=0.0056 - val loss:=0.0066, train acc:=0.8
8 - val acc:=0.82
# Epoch:=6/10  - train loss:=0.0052 - val loss:=0.0066, train acc:=0.8
8 - val acc:=0.88
# Epoch:=7/10  - train loss:=0.0049 - val loss:=0.0065, train acc:=0.8
8 - val acc:=0.88
# Epoch:=8/10  - train loss:=0.0045 - val loss:=0.0064, train acc:=0.9
2 - val acc:=0.88
# Epoch:=9/10  - train loss:=0.0042 - val loss:=0.0064, train acc:=0.9
4 - val acc:=0.88
# Epoch:=10/10  - train loss:=0.0039 - val loss:=0.0064, train acc:=0.
88 - val acc:=0.94

Total time taken (in seconds):  192.85

Finished training model: mlp_on_gpu_default

**************** Testing ****************
mlp_on_gpu_default model accuracy = 88.06%
*****************************************
```



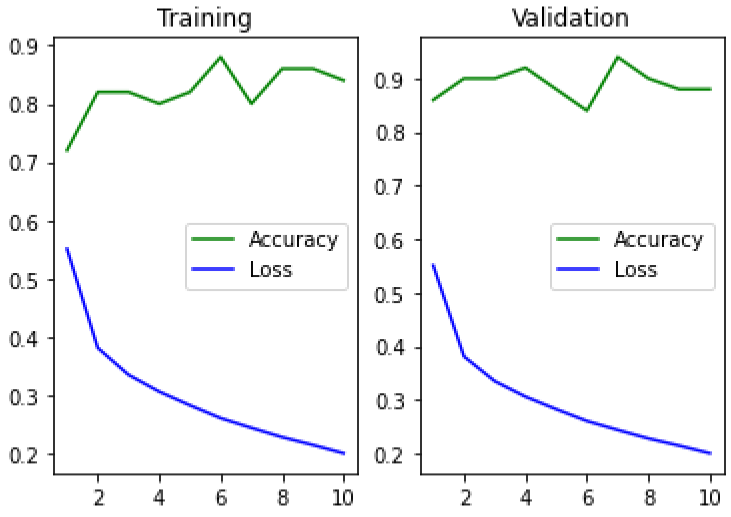
```
Count: 4, j=: 1


**************** Training model: mlp_on_gpu_default with optimizer: SG
D and seed: 1566 ****************

# Epoch:=1/10  - train loss:=0.0110 - val loss:=0.0082, train acc:=0.8
0 - val acc:=0.84
# Epoch:=2/10  - train loss:=0.0076 - val loss:=0.0079, train acc:=0.7
8 - val acc:=0.84
# Epoch:=3/10  - train loss:=0.0067 - val loss:=0.0071, train acc:=0.8
2 - val acc:=0.90
# Epoch:=4/10  - train loss:=0.0061 - val loss:=0.0069, train acc:=0.8
6 - val acc:=0.88
# Epoch:=5/10  - train loss:=0.0056 - val loss:=0.0070, train acc:=0.8
4 - val acc:=0.90
# Epoch:=6/10  - train loss:=0.0052 - val loss:=0.0064, train acc:=0.9
2 - val acc:=0.88
# Epoch:=7/10  - train loss:=0.0049 - val loss:=0.0064, train acc:=0.9
4 - val acc:=0.88
# Epoch:=8/10  - train loss:=0.0045 - val loss:=0.0065, train acc:=0.8
8 - val acc:=0.94
# Epoch:=9/10  - train loss:=0.0042 - val loss:=0.0062, train acc:=0.9
6 - val acc:=0.86
# Epoch:=10/10  - train loss:=0.0040 - val loss:=0.0063, train acc:=0.
94 - val acc:=0.92

Total time taken (in seconds):  194.92

Finished training model: mlp_on_gpu_default

**************** Testing ****************
mlp_on_gpu_default model accuracy = 88.08%
*****************************************
```



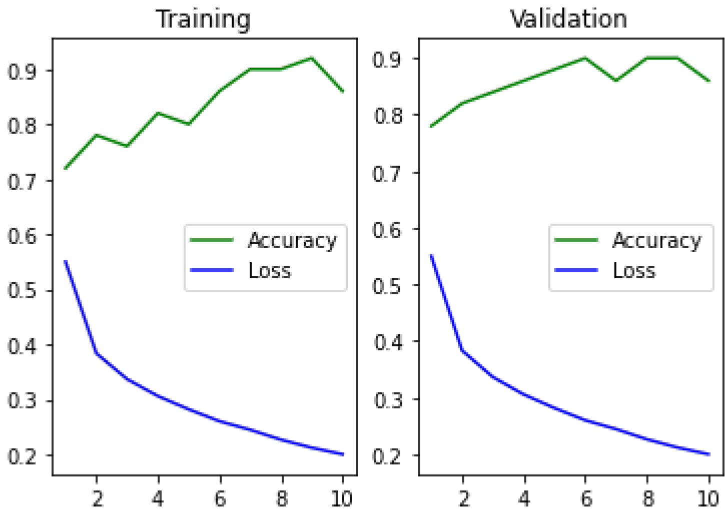
```
Count: 5, j=: 1


**************** Training model: mlp_on_gpu_default with optimizer: SG
D and seed: 1276 ****************

# Epoch:=1/10  - train loss:=0.0109 - val loss:=0.0082, train acc:=0.8
4 - val acc:=0.84
# Epoch:=2/10  - train loss:=0.0076 - val loss:=0.0076, train acc:=0.7
8 - val acc:=0.88
# Epoch:=3/10  - train loss:=0.0067 - val loss:=0.0072, train acc:=0.8
6 - val acc:=0.90
# Epoch:=4/10  - train loss:=0.0061 - val loss:=0.0070, train acc:=0.8
0 - val acc:=0.90
# Epoch:=5/10  - train loss:=0.0056 - val loss:=0.0067, train acc:=0.8
0 - val acc:=0.88
# Epoch:=6/10  - train loss:=0.0052 - val loss:=0.0065, train acc:=0.9
2 - val acc:=0.90
# Epoch:=7/10  - train loss:=0.0048 - val loss:=0.0067, train acc:=0.8
6 - val acc:=0.86
# Epoch:=8/10  - train loss:=0.0045 - val loss:=0.0065, train acc:=0.9
2 - val acc:=0.90
# Epoch:=9/10  - train loss:=0.0042 - val loss:=0.0065, train acc:=0.9
2 - val acc:=0.90
# Epoch:=10/10  - train loss:=0.0039 - val loss:=0.0064, train acc:=0.
92 - val acc:=0.88

Total time taken (in seconds):  194.39

Finished training model: mlp_on_gpu_default

**************** Testing ****************
mlp_on_gpu_default model accuracy = 88.36%
*****************************************
```



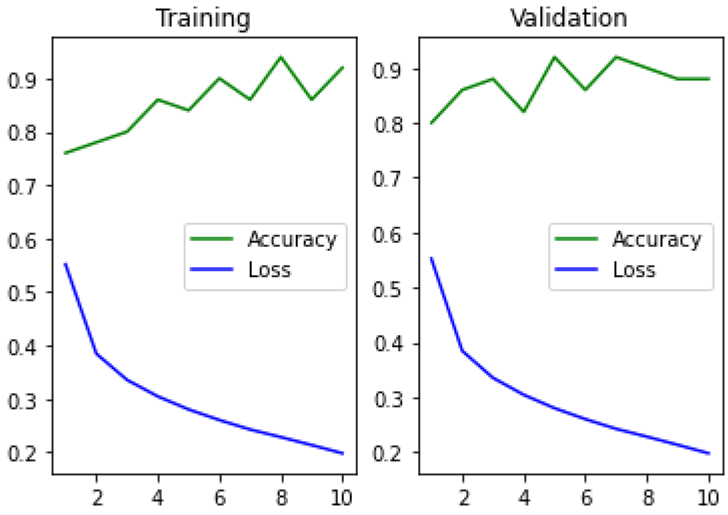
```
Count: 6, j=: 1


**************** Training model: mlp_on_gpu_default with optimizer: SG
D and seed: 3499 ****************

# Epoch:=1/10  - train loss:=0.0110 - val loss:=0.0084, train acc:=0.7
2 - val acc:=0.86
# Epoch:=2/10  - train loss:=0.0076 - val loss:=0.0076, train acc:=0.8
2 - val acc:=0.90
# Epoch:=3/10  - train loss:=0.0067 - val loss:=0.0068, train acc:=0.8
2 - val acc:=0.90
# Epoch:=4/10  - train loss:=0.0061 - val loss:=0.0065, train acc:=0.8
0 - val acc:=0.92
# Epoch:=5/10  - train loss:=0.0056 - val loss:=0.0065, train acc:=0.8
2 - val acc:=0.88
# Epoch:=6/10  - train loss:=0.0052 - val loss:=0.0064, train acc:=0.8
8 - val acc:=0.84
# Epoch:=7/10  - train loss:=0.0049 - val loss:=0.0065, train acc:=0.8
0 - val acc:=0.94
# Epoch:=8/10  - train loss:=0.0046 - val loss:=0.0064, train acc:=0.8
6 - val acc:=0.90
# Epoch:=9/10  - train loss:=0.0043 - val loss:=0.0062, train acc:=0.8
6 - val acc:=0.88
# Epoch:=10/10  - train loss:=0.0040 - val loss:=0.0074, train acc:=0.
84 - val acc:=0.88

Total time taken (in seconds):  193.93

Finished training model: mlp_on_gpu_default

**************** Testing ****************
mlp_on_gpu_default model accuracy = 87.12%
*****************************************
```



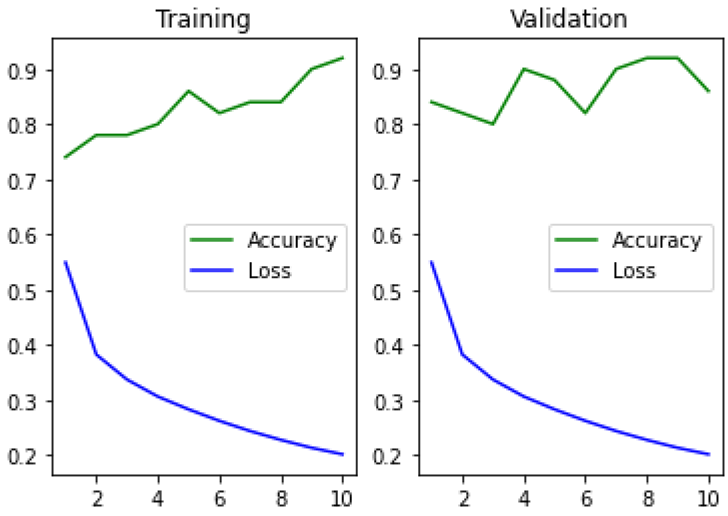
```
Count: 7, j=: 1


**************** Training model: mlp_on_gpu_default with optimizer: SG
D and seed: 6753 ****************

# Epoch:=1/10  - train loss:=0.0110 - val loss:=0.0087, train acc:=0.7
2 - val acc:=0.78
# Epoch:=2/10  - train loss:=0.0077 - val loss:=0.0084, train acc:=0.7
8 - val acc:=0.82
# Epoch:=3/10  - train loss:=0.0067 - val loss:=0.0071, train acc:=0.7
6 - val acc:=0.84
# Epoch:=4/10  - train loss:=0.0061 - val loss:=0.0075, train acc:=0.8
2 - val acc:=0.86
# Epoch:=5/10  - train loss:=0.0056 - val loss:=0.0066, train acc:=0.8
0 - val acc:=0.88
# Epoch:=6/10  - train loss:=0.0052 - val loss:=0.0067, train acc:=0.8
6 - val acc:=0.90
# Epoch:=7/10  - train loss:=0.0049 - val loss:=0.0063, train acc:=0.9
0 - val acc:=0.86
# Epoch:=8/10  - train loss:=0.0045 - val loss:=0.0066, train acc:=0.9
0 - val acc:=0.90
# Epoch:=9/10  - train loss:=0.0042 - val loss:=0.0062, train acc:=0.9
2 - val acc:=0.90
# Epoch:=10/10  - train loss:=0.0040 - val loss:=0.0065, train acc:=0.
86 - val acc:=0.86

Total time taken (in seconds):  193.40

Finished training model: mlp_on_gpu_default

**************** Testing ****************
mlp_on_gpu_default model accuracy = 88.31%
*****************************************
```



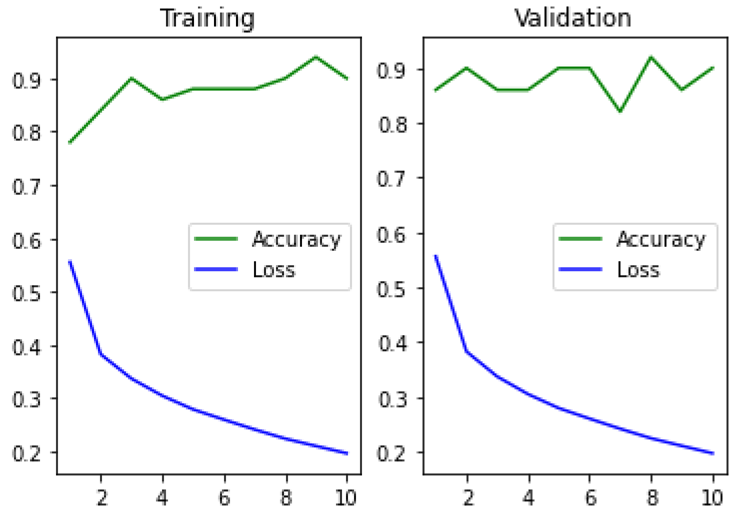
```
Count: 8, j=: 1


**************** Training model: mlp_on_gpu_default with optimizer: SG
D and seed: 6860 ****************

# Epoch:=1/10  - train loss:=0.0109 - val loss:=0.0082, train acc:=0.7
6 - val acc:=0.80
# Epoch:=2/10  - train loss:=0.0076 - val loss:=0.0082, train acc:=0.7
8 - val acc:=0.86
# Epoch:=3/10  - train loss:=0.0066 - val loss:=0.0069, train acc:=0.8
0 - val acc:=0.88
# Epoch:=4/10  - train loss:=0.0060 - val loss:=0.0069, train acc:=0.8
6 - val acc:=0.82
# Epoch:=5/10  - train loss:=0.0056 - val loss:=0.0068, train acc:=0.8
4 - val acc:=0.92
# Epoch:=6/10  - train loss:=0.0052 - val loss:=0.0069, train acc:=0.9
0 - val acc:=0.86
# Epoch:=7/10  - train loss:=0.0048 - val loss:=0.0068, train acc:=0.8
6 - val acc:=0.92
# Epoch:=8/10  - train loss:=0.0045 - val loss:=0.0064, train acc:=0.9
4 - val acc:=0.90
# Epoch:=9/10  - train loss:=0.0042 - val loss:=0.0064, train acc:=0.8
6 - val acc:=0.88
# Epoch:=10/10  - train loss:=0.0039 - val loss:=0.0066, train acc:=0.
92 - val acc:=0.88

Total time taken (in seconds):  192.68

Finished training model: mlp_on_gpu_default

**************** Testing ****************
mlp_on_gpu_default model accuracy = 87.77%
*****************************************
```



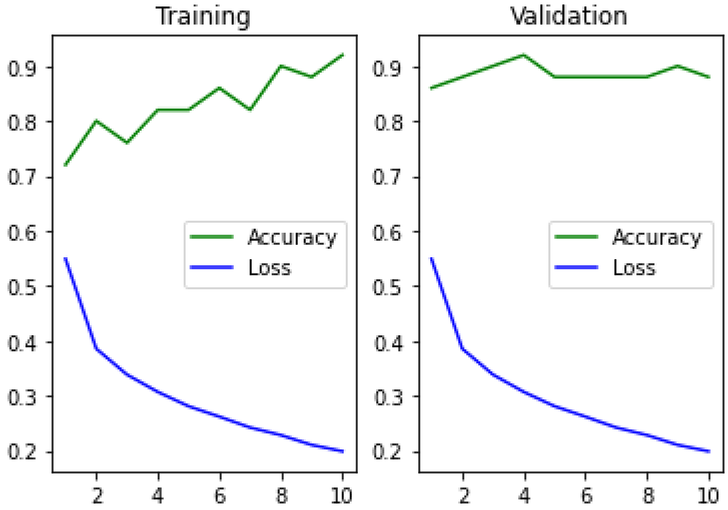
```
Count: 9, j=: 1


**************** Training model: mlp_on_gpu_default with optimizer: SG
D and seed: 5776 ****************

# Epoch:=1/10  - train loss:=0.0109 - val loss:=0.0084, train acc:=0.7
4 - val acc:=0.84
# Epoch:=2/10  - train loss:=0.0076 - val loss:=0.0078, train acc:=0.7
8 - val acc:=0.82
# Epoch:=3/10  - train loss:=0.0067 - val loss:=0.0075, train acc:=0.7
8 - val acc:=0.80
# Epoch:=4/10  - train loss:=0.0061 - val loss:=0.0070, train acc:=0.8
0 - val acc:=0.90
# Epoch:=5/10  - train loss:=0.0056 - val loss:=0.0064, train acc:=0.8
6 - val acc:=0.88
# Epoch:=6/10  - train loss:=0.0052 - val loss:=0.0072, train acc:=0.8
2 - val acc:=0.82
# Epoch:=7/10  - train loss:=0.0048 - val loss:=0.0065, train acc:=0.8
4 - val acc:=0.90
# Epoch:=8/10  - train loss:=0.0045 - val loss:=0.0068, train acc:=0.8
4 - val acc:=0.92
# Epoch:=9/10  - train loss:=0.0042 - val loss:=0.0068, train acc:=0.9
0 - val acc:=0.92
# Epoch:=10/10  - train loss:=0.0040 - val loss:=0.0063, train acc:=0.
92 - val acc:=0.86

Total time taken (in seconds):  193.26

Finished training model: mlp_on_gpu_default

**************** Testing ****************
mlp_on_gpu_default model accuracy = 88.12%
****************************************
```



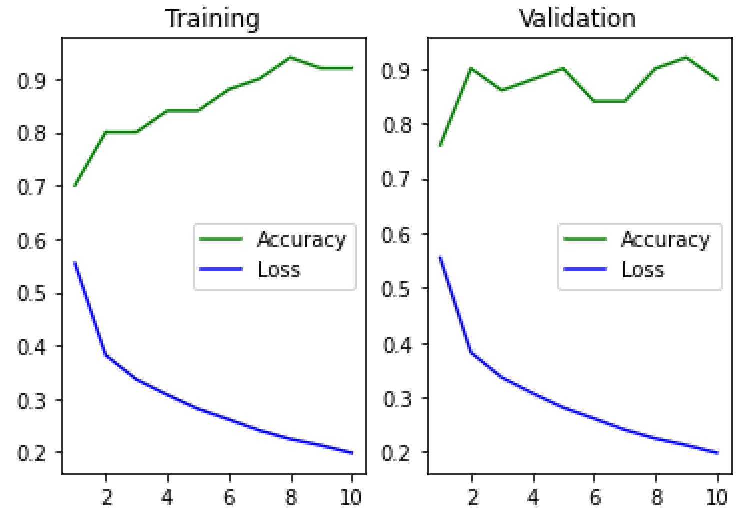
```
Count: 0, j=: 1


**************** Training model: mlp_on_gpu_RegL1 with optimizer: SGD
and seed: 6955 ****************

# Epoch:=1/10  - train loss:=0.0111 - val loss:=0.0082, train acc:=0.7
8 - val acc:=0.86
# Epoch:=2/10  - train loss:=0.0076 - val loss:=0.0074, train acc:=0.8
4 - val acc:=0.90
# Epoch:=3/10  - train loss:=0.0067 - val loss:=0.0071, train acc:=0.9
0 - val acc:=0.86
# Epoch:=4/10  - train loss:=0.0061 - val loss:=0.0069, train acc:=0.8
6 - val acc:=0.86
# Epoch:=5/10  - train loss:=0.0056 - val loss:=0.0066, train acc:=0.8
8 - val acc:=0.90
# Epoch:=6/10  - train loss:=0.0052 - val loss:=0.0067, train acc:=0.8
8 - val acc:=0.90
# Epoch:=7/10  - train loss:=0.0048 - val loss:=0.0064, train acc:=0.8
8 - val acc:=0.82
# Epoch:=8/10  - train loss:=0.0045 - val loss:=0.0070, train acc:=0.9
0 - val acc:=0.92
# Epoch:=9/10  - train loss:=0.0042 - val loss:=0.0068, train acc:=0.9
4 - val acc:=0.86
# Epoch:=10/10  - train loss:=0.0039 - val loss:=0.0066, train acc:=0.
90 - val acc:=0.90

Total time taken (in seconds):  237.93

Finished training model: mlp_on_gpu_RegL1

**************** Testing ****************
mlp_on_gpu_RegL1 model accuracy = 88.25%
*****************************************
```



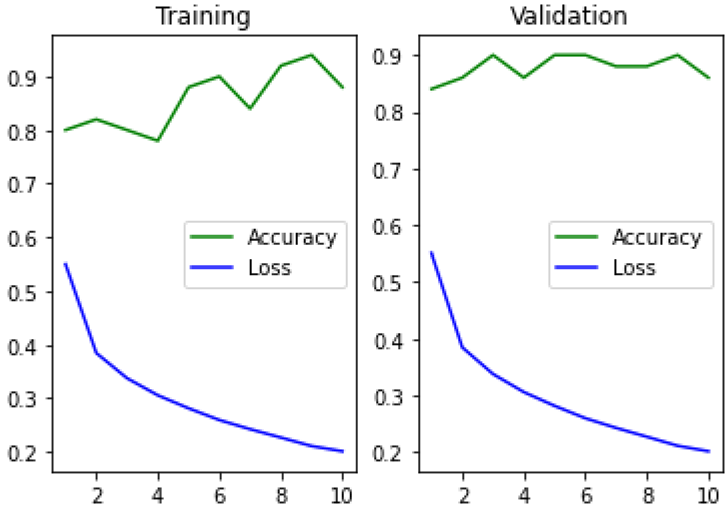
```
Count: 1, j=: 1


**************** Training model: mlp_on_gpu_RegL1 with optimizer: SGD
and seed: 3079 ****************

# Epoch:=1/10  - train loss:=0.0109 - val loss:=0.0081, train acc:=0.7
2 - val acc:=0.86
# Epoch:=2/10  - train loss:=0.0076 - val loss:=0.0077, train acc:=0.8
0 - val acc:=0.88
# Epoch:=3/10  - train loss:=0.0067 - val loss:=0.0085, train acc:=0.7
6 - val acc:=0.90
# Epoch:=4/10  - train loss:=0.0061 - val loss:=0.0068, train acc:=0.8
2 - val acc:=0.92
# Epoch:=5/10  - train loss:=0.0055 - val loss:=0.0067, train acc:=0.8
2 - val acc:=0.88
# Epoch:=6/10  - train loss:=0.0052 - val loss:=0.0062, train acc:=0.8
6 - val acc:=0.88
# Epoch:=7/10  - train loss:=0.0048 - val loss:=0.0068, train acc:=0.8
2 - val acc:=0.88
# Epoch:=8/10  - train loss:=0.0045 - val loss:=0.0062, train acc:=0.9
0 - val acc:=0.88
# Epoch:=9/10  - train loss:=0.0042 - val loss:=0.0063, train acc:=0.8
8 - val acc:=0.90
# Epoch:=10/10  - train loss:=0.0039 - val loss:=0.0062, train acc:=0.
92 - val acc:=0.88

Total time taken (in seconds):  238.07

Finished training model: mlp_on_gpu_RegL1

**************** Testing ****************
mlp_on_gpu_RegL1 model accuracy = 88.49%
*****************************************
```



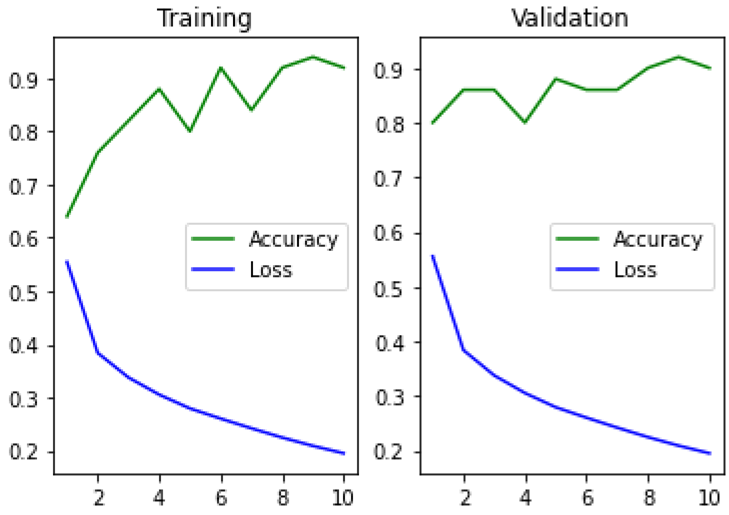
```
Count: 2, j=: 1


**************** Training model: mlp_on_gpu_RegL1 with optimizer: SGD
and seed: 9781 ****************

# Epoch:=1/10  - train loss:=0.0110 - val loss:=0.0086, train acc:=0.7
0 - val acc:=0.76
# Epoch:=2/10  - train loss:=0.0076 - val loss:=0.0073, train acc:=0.8
0 - val acc:=0.90
# Epoch:=3/10  - train loss:=0.0067 - val loss:=0.0073, train acc:=0.8
0 - val acc:=0.86
# Epoch:=4/10  - train loss:=0.0061 - val loss:=0.0068, train acc:=0.8
4 - val acc:=0.88
# Epoch:=5/10  - train loss:=0.0056 - val loss:=0.0065, train acc:=0.8
4 - val acc:=0.90
# Epoch:=6/10  - train loss:=0.0052 - val loss:=0.0068, train acc:=0.8
8 - val acc:=0.84
# Epoch:=7/10  - train loss:=0.0048 - val loss:=0.0063, train acc:=0.9
0 - val acc:=0.84
# Epoch:=8/10  - train loss:=0.0045 - val loss:=0.0065, train acc:=0.9
4 - val acc:=0.90
# Epoch:=9/10  - train loss:=0.0042 - val loss:=0.0065, train acc:=0.9
2 - val acc:=0.92
# Epoch:=10/10  - train loss:=0.0039 - val loss:=0.0065, train acc:=0.
92 - val acc:=0.88

Total time taken (in seconds):  238.40

Finished training model: mlp_on_gpu_RegL1

**************** Testing ****************
mlp_on_gpu_RegL1 model accuracy = 88.03%
*****************************************
```



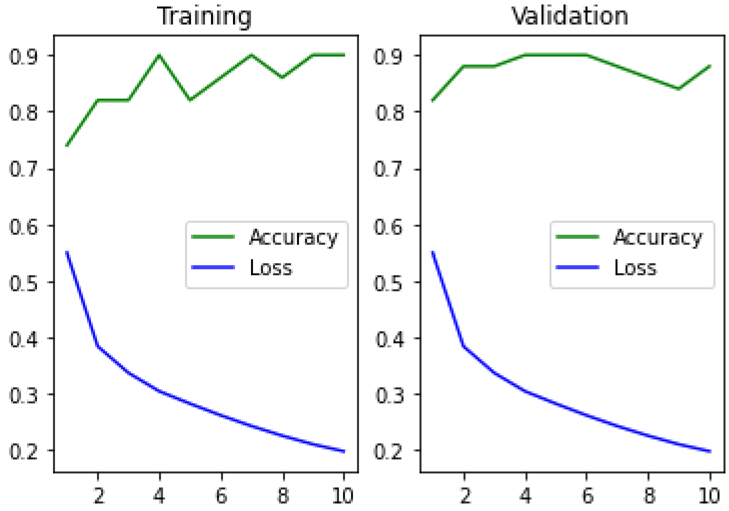
```
Count: 3, j=: 1


**************** Training model: mlp_on_gpu_RegL1 with optimizer: SGD
and seed: 1357 ****************

# Epoch:=1/10  - train loss:=0.0110 - val loss:=0.0085, train acc:=0.8
0 - val acc:=0.84
# Epoch:=2/10  - train loss:=0.0077 - val loss:=0.0075, train acc:=0.8
2 - val acc:=0.86
# Epoch:=3/10  - train loss:=0.0067 - val loss:=0.0072, train acc:=0.8
0 - val acc:=0.90
# Epoch:=4/10  - train loss:=0.0061 - val loss:=0.0076, train acc:=0.7
8 - val acc:=0.86
# Epoch:=5/10  - train loss:=0.0056 - val loss:=0.0065, train acc:=0.8
8 - val acc:=0.90
# Epoch:=6/10  - train loss:=0.0052 - val loss:=0.0066, train acc:=0.9
0 - val acc:=0.90
# Epoch:=7/10  - train loss:=0.0048 - val loss:=0.0073, train acc:=0.8
4 - val acc:=0.88
# Epoch:=8/10  - train loss:=0.0045 - val loss:=0.0062, train acc:=0.9
2 - val acc:=0.88
# Epoch:=9/10  - train loss:=0.0042 - val loss:=0.0062, train acc:=0.9
4 - val acc:=0.90
# Epoch:=10/10  - train loss:=0.0040 - val loss:=0.0065, train acc:=0.
88 - val acc:=0.86

Total time taken (in seconds):  238.41

Finished training model: mlp_on_gpu_RegL1

**************** Testing ****************
mlp_on_gpu_RegL1 model accuracy = 88.10%
*****************************************
```



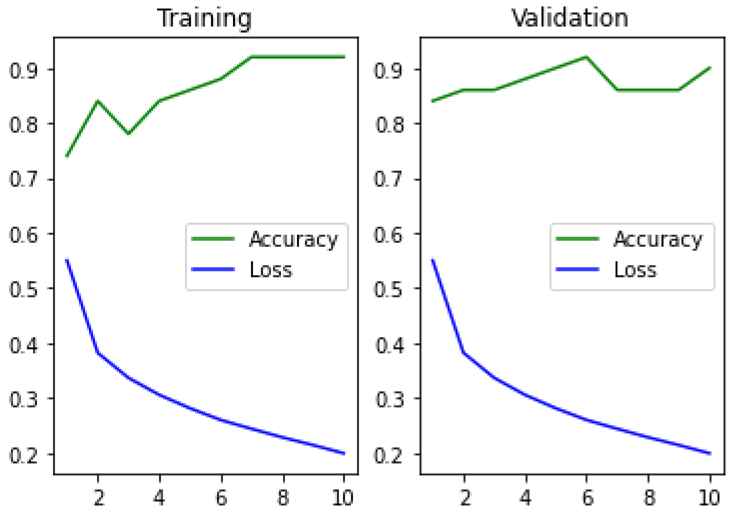
```
Count: 4, j=: 1


**************** Training model: mlp_on_gpu_RegL1 with optimizer: SGD
and seed: 6333 ****************

# Epoch:=1/10  - train loss:=0.0111 - val loss:=0.0087, train acc:=0.6
4 - val acc:=0.80
# Epoch:=2/10  - train loss:=0.0076 - val loss:=0.0076, train acc:=0.7
6 - val acc:=0.86
# Epoch:=3/10  - train loss:=0.0067 - val loss:=0.0073, train acc:=0.8
2 - val acc:=0.86
# Epoch:=4/10  - train loss:=0.0061 - val loss:=0.0068, train acc:=0.8
8 - val acc:=0.80
# Epoch:=5/10  - train loss:=0.0056 - val loss:=0.0068, train acc:=0.8
0 - val acc:=0.88
# Epoch:=6/10  - train loss:=0.0052 - val loss:=0.0065, train acc:=0.9
2 - val acc:=0.86
# Epoch:=7/10  - train loss:=0.0048 - val loss:=0.0066, train acc:=0.8
4 - val acc:=0.86
# Epoch:=8/10  - train loss:=0.0045 - val loss:=0.0063, train acc:=0.9
2 - val acc:=0.90
# Epoch:=9/10  - train loss:=0.0042 - val loss:=0.0064, train acc:=0.9
4 - val acc:=0.92
# Epoch:=10/10  - train loss:=0.0039 - val loss:=0.0069, train acc:=0.
92 - val acc:=0.90

Total time taken (in seconds):  242.83

Finished training model: mlp_on_gpu_RegL1

**************** Testing ****************
mlp_on_gpu_RegL1 model accuracy = 87.36%
*****************************************
```



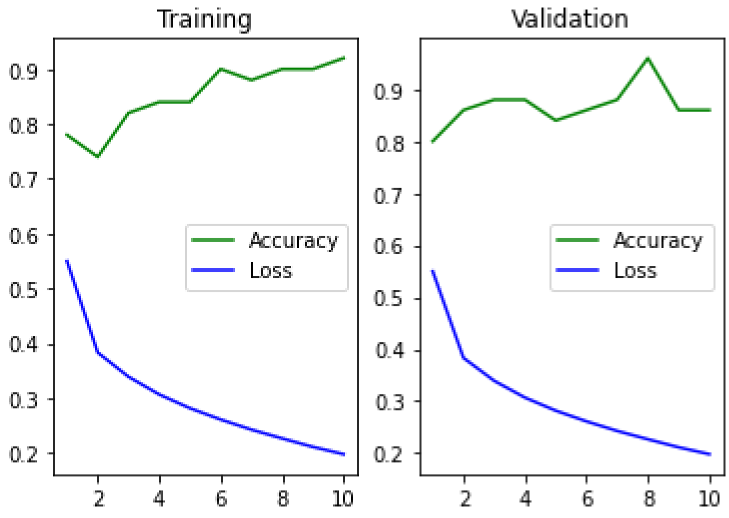
```
Count: 5, j=: 1


**************** Training model: mlp_on_gpu_RegL1 with optimizer: SGD
and seed: 5649 ****************

# Epoch:=1/10  - train loss:=0.0109 - val loss:=0.0084, train acc:=0.7
4 - val acc:=0.82
# Epoch:=2/10  - train loss:=0.0076 - val loss:=0.0073, train acc:=0.8
2 - val acc:=0.88
# Epoch:=3/10  - train loss:=0.0067 - val loss:=0.0070, train acc:=0.8
2 - val acc:=0.88
# Epoch:=4/10  - train loss:=0.0060 - val loss:=0.0065, train acc:=0.9
0 - val acc:=0.90
# Epoch:=5/10  - train loss:=0.0056 - val loss:=0.0064, train acc:=0.8
2 - val acc:=0.90
# Epoch:=6/10  - train loss:=0.0052 - val loss:=0.0063, train acc:=0.8
6 - val acc:=0.90
# Epoch:=7/10  - train loss:=0.0048 - val loss:=0.0063, train acc:=0.9
0 - val acc:=0.88
# Epoch:=8/10  - train loss:=0.0045 - val loss:=0.0062, train acc:=0.8
6 - val acc:=0.86
# Epoch:=9/10  - train loss:=0.0042 - val loss:=0.0061, train acc:=0.9
0 - val acc:=0.84
# Epoch:=10/10  - train loss:=0.0039 - val loss:=0.0066, train acc:=0.
90 - val acc:=0.88

Total time taken (in seconds):  260.34

Finished training model: mlp_on_gpu_RegL1

**************** Testing ****************
mlp_on_gpu_RegL1 model accuracy = 87.73%
*****************************************
```



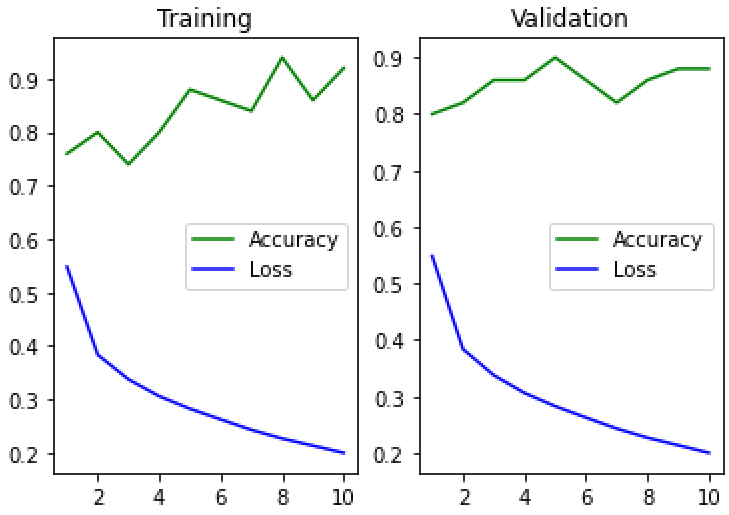
```
Count: 6, j=: 1


**************** Training model: mlp_on_gpu_RegL1 with optimizer: SGD
and seed: 9604 ****************

# Epoch:=1/10  - train loss:=0.0109 - val loss:=0.0084, train acc:=0.7
4 - val acc:=0.84
# Epoch:=2/10  - train loss:=0.0076 - val loss:=0.0072, train acc:=0.8
4 - val acc:=0.86
# Epoch:=3/10  - train loss:=0.0067 - val loss:=0.0070, train acc:=0.7
8 - val acc:=0.86
# Epoch:=4/10  - train loss:=0.0061 - val loss:=0.0071, train acc:=0.8
4 - val acc:=0.88
# Epoch:=5/10  - train loss:=0.0056 - val loss:=0.0068, train acc:=0.8
6 - val acc:=0.90
# Epoch:=6/10  - train loss:=0.0051 - val loss:=0.0069, train acc:=0.8
8 - val acc:=0.92
# Epoch:=7/10  - train loss:=0.0048 - val loss:=0.0063, train acc:=0.9
2 - val acc:=0.86
# Epoch:=8/10  - train loss:=0.0045 - val loss:=0.0062, train acc:=0.9
2 - val acc:=0.86
# Epoch:=9/10  - train loss:=0.0042 - val loss:=0.0060, train acc:=0.9
2 - val acc:=0.86
# Epoch:=10/10  - train loss:=0.0039 - val loss:=0.0065, train acc:=0.
92 - val acc:=0.90

Total time taken (in seconds):  242.66

Finished training model: mlp_on_gpu_RegL1

**************** Testing ****************
mlp_on_gpu_RegL1 model accuracy = 88.31%
*****************************************
```



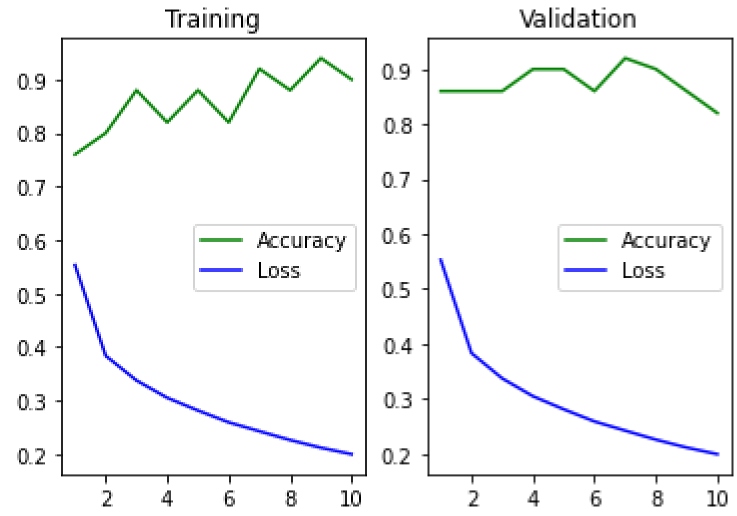
```
Count: 7, j=: 1


**************** Training model: mlp_on_gpu_RegL1 with optimizer: SGD
and seed: 1477 ****************

# Epoch:=1/10  - train loss:=0.0109 - val loss:=0.0084, train acc:=0.7
8 - val acc:=0.80
# Epoch:=2/10  - train loss:=0.0076 - val loss:=0.0076, train acc:=0.7
4 - val acc:=0.86
# Epoch:=3/10  - train loss:=0.0067 - val loss:=0.0069, train acc:=0.8
2 - val acc:=0.88
# Epoch:=4/10  - train loss:=0.0061 - val loss:=0.0066, train acc:=0.8
4 - val acc:=0.88
# Epoch:=5/10  - train loss:=0.0056 - val loss:=0.0065, train acc:=0.8
4 - val acc:=0.84
# Epoch:=6/10  - train loss:=0.0052 - val loss:=0.0066, train acc:=0.9
0 - val acc:=0.86
# Epoch:=7/10  - train loss:=0.0048 - val loss:=0.0064, train acc:=0.8
8 - val acc:=0.88
# Epoch:=8/10  - train loss:=0.0045 - val loss:=0.0063, train acc:=0.9
0 - val acc:=0.96
# Epoch:=9/10  - train loss:=0.0042 - val loss:=0.0064, train acc:=0.9
0 - val acc:=0.86
# Epoch:=10/10  - train loss:=0.0039 - val loss:=0.0063, train acc:=0.
92 - val acc:=0.86

Total time taken (in seconds):  242.49

Finished training model: mlp_on_gpu_RegL1

**************** Testing ****************
mlp_on_gpu_RegL1 model accuracy = 88.39%
*****************************************
```



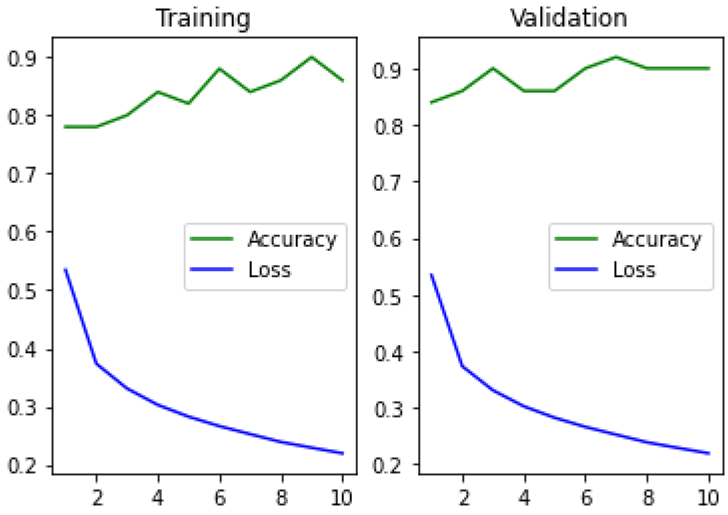
```
Count: 8, j=: 1


**************** Training model: mlp_on_gpu_RegL1 with optimizer: SGD
and seed: 8149 ****************

# Epoch:=1/10  - train loss:=0.0109 - val loss:=0.0083, train acc:=0.7
6 - val acc:=0.80
# Epoch:=2/10  - train loss:=0.0076 - val loss:=0.0077, train acc:=0.8
0 - val acc:=0.82
# Epoch:=3/10  - train loss:=0.0067 - val loss:=0.0074, train acc:=0.7
4 - val acc:=0.86
# Epoch:=4/10  - train loss:=0.0061 - val loss:=0.0070, train acc:=0.8
0 - val acc:=0.86
# Epoch:=5/10  - train loss:=0.0056 - val loss:=0.0067, train acc:=0.8
8 - val acc:=0.90
# Epoch:=6/10  - train loss:=0.0052 - val loss:=0.0068, train acc:=0.8
6 - val acc:=0.86
# Epoch:=7/10  - train loss:=0.0048 - val loss:=0.0070, train acc:=0.8
4 - val acc:=0.82
# Epoch:=8/10  - train loss:=0.0045 - val loss:=0.0065, train acc:=0.9
4 - val acc:=0.86
# Epoch:=9/10  - train loss:=0.0042 - val loss:=0.0067, train acc:=0.8
6 - val acc:=0.88
# Epoch:=10/10  - train loss:=0.0040 - val loss:=0.0065, train acc:=0.
92 - val acc:=0.88

Total time taken (in seconds):  242.18

Finished training model: mlp_on_gpu_RegL1

**************** Testing ****************
mlp_on_gpu_RegL1 model accuracy = 87.80%
*****************************************
```



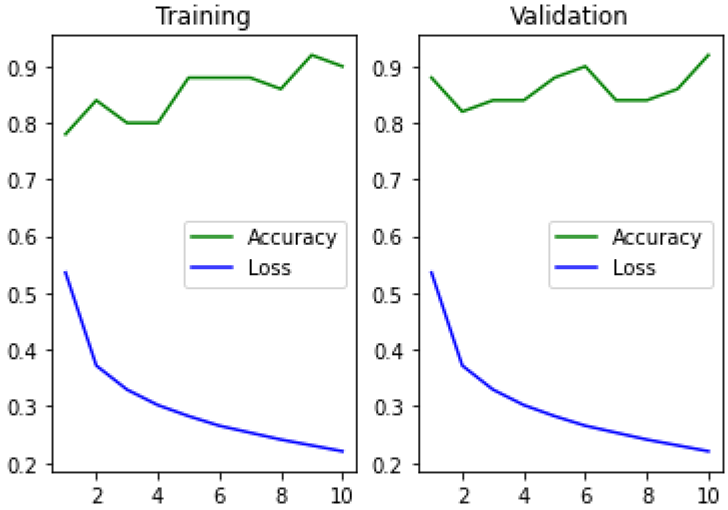
```
Count: 9, j=: 1


**************** Training model: mlp_on_gpu_RegL1 with optimizer: SGD
and seed: 7004 ****************

# Epoch:=1/10  - train loss:=0.0110 - val loss:=0.0083, train acc:=0.7
6 - val acc:=0.86
# Epoch:=2/10  - train loss:=0.0076 - val loss:=0.0074, train acc:=0.8
0 - val acc:=0.86
# Epoch:=3/10  - train loss:=0.0067 - val loss:=0.0070, train acc:=0.8
8 - val acc:=0.86
# Epoch:=4/10  - train loss:=0.0061 - val loss:=0.0066, train acc:=0.8
2 - val acc:=0.90
# Epoch:=5/10  - train loss:=0.0056 - val loss:=0.0066, train acc:=0.8
8 - val acc:=0.90
# Epoch:=6/10  - train loss:=0.0052 - val loss:=0.0068, train acc:=0.8
2 - val acc:=0.86
# Epoch:=7/10  - train loss:=0.0048 - val loss:=0.0064, train acc:=0.9
2 - val acc:=0.92
# Epoch:=8/10  - train loss:=0.0045 - val loss:=0.0062, train acc:=0.8
8 - val acc:=0.90
# Epoch:=9/10  - train loss:=0.0042 - val loss:=0.0065, train acc:=0.9
4 - val acc:=0.86
# Epoch:=10/10  - train loss:=0.0040 - val loss:=0.0065, train acc:=0.
90 - val acc:=0.82

Total time taken (in seconds):  242.40

Finished training model: mlp_on_gpu_RegL1

**************** Testing ****************
mlp_on_gpu_RegL1 model accuracy = 87.89%
*****************************************
```



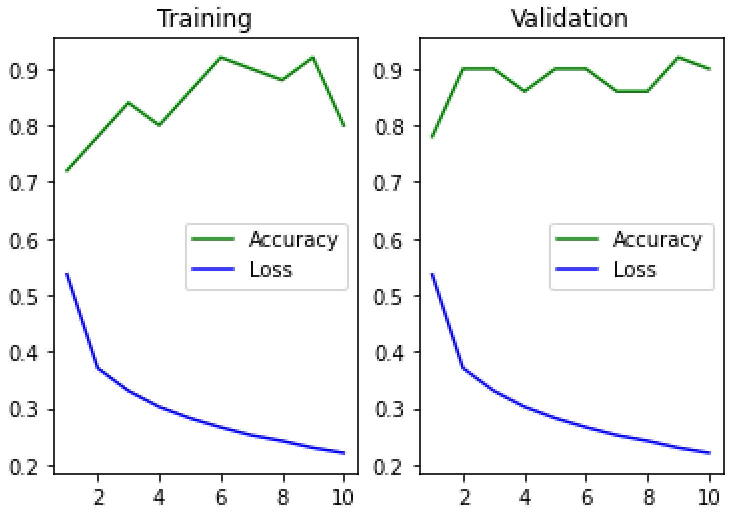
```
Count: 0, j=: 1


**************** Training model: mlp_on_gpu_RegL2 with optimizer: SGD
and seed: 1603 ****************

# Epoch:=1/10  - train loss:=0.0111 - val loss:=0.0085, train acc:=0.7
8 - val acc:=0.84
# Epoch:=2/10  - train loss:=0.0077 - val loss:=0.0078, train acc:=0.7
8 - val acc:=0.86
# Epoch:=3/10  - train loss:=0.0069 - val loss:=0.0071, train acc:=0.8
0 - val acc:=0.90
# Epoch:=4/10  - train loss:=0.0063 - val loss:=0.0067, train acc:=0.8
4 - val acc:=0.86
# Epoch:=5/10  - train loss:=0.0059 - val loss:=0.0065, train acc:=0.8
2 - val acc:=0.86
# Epoch:=6/10  - train loss:=0.0055 - val loss:=0.0065, train acc:=0.8
8 - val acc:=0.90
# Epoch:=7/10  - train loss:=0.0052 - val loss:=0.0064, train acc:=0.8
4 - val acc:=0.92
# Epoch:=8/10  - train loss:=0.0050 - val loss:=0.0062, train acc:=0.8
6 - val acc:=0.90
# Epoch:=9/10  - train loss:=0.0047 - val loss:=0.0064, train acc:=0.9
0 - val acc:=0.90
# Epoch:=10/10  - train loss:=0.0046 - val loss:=0.0061, train acc:=0.
86 - val acc:=0.90

Total time taken (in seconds):  282.30

Finished training model: mlp_on_gpu_RegL2

**************** Testing ****************
mlp_on_gpu_RegL2 model accuracy = 88.07%
*****************************************
```



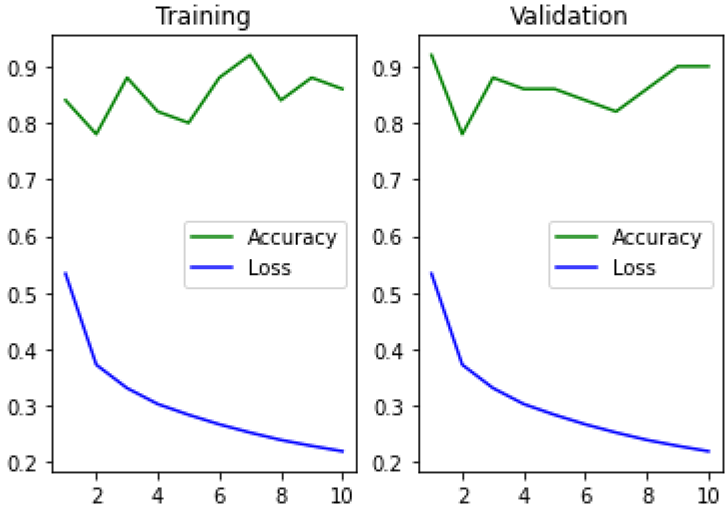
```
Count: 1, j=: 1


**************** Training model: mlp_on_gpu_RegL2 with optimizer: SGD
and seed: 2235 ****************

# Epoch:=1/10  - train loss:=0.0111 - val loss:=0.0084, train acc:=0.7
8 - val acc:=0.88
# Epoch:=2/10  - train loss:=0.0077 - val loss:=0.0075, train acc:=0.8
4 - val acc:=0.82
# Epoch:=3/10  - train loss:=0.0068 - val loss:=0.0073, train acc:=0.8
0 - val acc:=0.84
# Epoch:=4/10  - train loss:=0.0063 - val loss:=0.0073, train acc:=0.8
0 - val acc:=0.84
# Epoch:=5/10  - train loss:=0.0059 - val loss:=0.0067, train acc:=0.8
8 - val acc:=0.88
# Epoch:=6/10  - train loss:=0.0055 - val loss:=0.0064, train acc:=0.8
8 - val acc:=0.90
# Epoch:=7/10  - train loss:=0.0053 - val loss:=0.0064, train acc:=0.8
8 - val acc:=0.84
# Epoch:=8/10  - train loss:=0.0050 - val loss:=0.0064, train acc:=0.8
6 - val acc:=0.84
# Epoch:=9/10  - train loss:=0.0048 - val loss:=0.0062, train acc:=0.9
2 - val acc:=0.86
# Epoch:=10/10  - train loss:=0.0046 - val loss:=0.0063, train acc:=0.
90 - val acc:=0.92

Total time taken (in seconds):  282.14

Finished training model: mlp_on_gpu_RegL2

**************** Testing ****************
mlp_on_gpu_RegL2 model accuracy = 87.87%
*****************************************
```



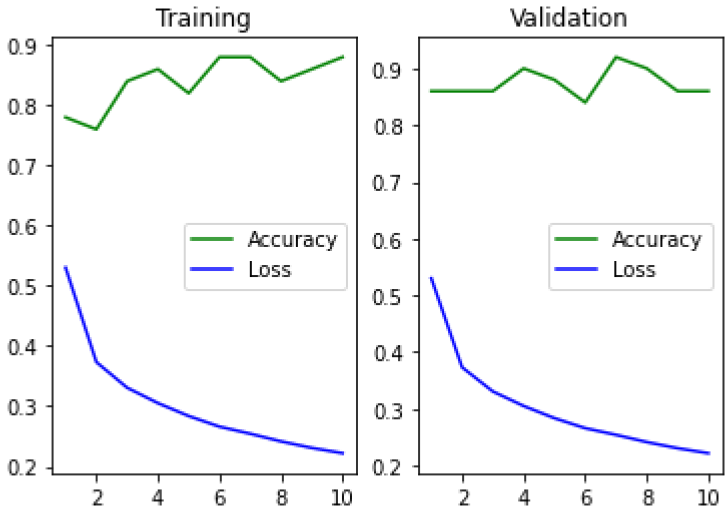
```
Count: 2, j=: 1


**************** Training model: mlp_on_gpu_RegL2 with optimizer: SGD
and seed: 1879 ****************

# Epoch:=1/10  - train loss:=0.0111 - val loss:=0.0085, train acc:=0.7
2 - val acc:=0.78
# Epoch:=2/10  - train loss:=0.0077 - val loss:=0.0077, train acc:=0.7
8 - val acc:=0.90
# Epoch:=3/10  - train loss:=0.0069 - val loss:=0.0073, train acc:=0.8
4 - val acc:=0.90
# Epoch:=4/10  - train loss:=0.0063 - val loss:=0.0068, train acc:=0.8
0 - val acc:=0.86
# Epoch:=5/10  - train loss:=0.0059 - val loss:=0.0067, train acc:=0.8
6 - val acc:=0.90
# Epoch:=6/10  - train loss:=0.0055 - val loss:=0.0064, train acc:=0.9
2 - val acc:=0.90
# Epoch:=7/10  - train loss:=0.0052 - val loss:=0.0066, train acc:=0.9
0 - val acc:=0.86
# Epoch:=8/10  - train loss:=0.0050 - val loss:=0.0066, train acc:=0.8
8 - val acc:=0.86
# Epoch:=9/10  - train loss:=0.0048 - val loss:=0.0064, train acc:=0.9
2 - val acc:=0.92
# Epoch:=10/10  - train loss:=0.0046 - val loss:=0.0067, train acc:=0.
80 - val acc:=0.90

Total time taken (in seconds):  280.83

Finished training model: mlp_on_gpu_RegL2

**************** Testing ****************
mlp_on_gpu_RegL2 model accuracy = 87.43%
*****************************************
```



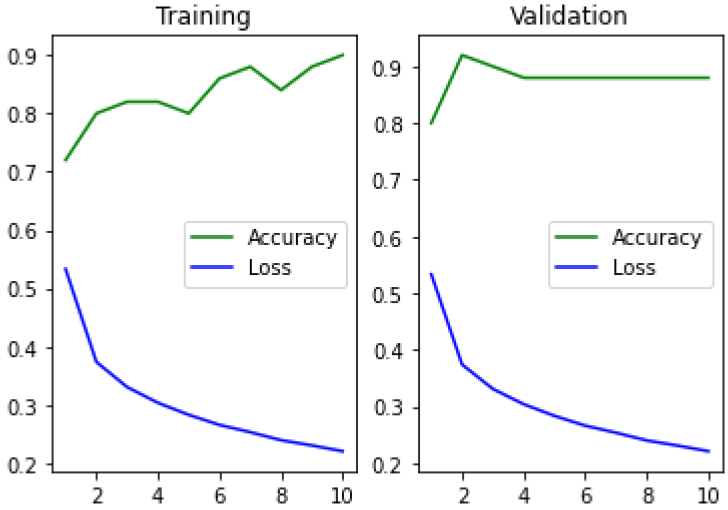
```
Count: 3, j=: 1


**************** Training model: mlp_on_gpu_RegL2 with optimizer: SGD
and seed: 7354 ****************

# Epoch:=1/10  - train loss:=0.0110 - val loss:=0.0083, train acc:=0.8
4 - val acc:=0.92
# Epoch:=2/10  - train loss:=0.0077 - val loss:=0.0083, train acc:=0.7
8 - val acc:=0.78
# Epoch:=3/10  - train loss:=0.0068 - val loss:=0.0069, train acc:=0.8
8 - val acc:=0.88
# Epoch:=4/10  - train loss:=0.0063 - val loss:=0.0068, train acc:=0.8
2 - val acc:=0.86
# Epoch:=5/10  - train loss:=0.0059 - val loss:=0.0063, train acc:=0.8
0 - val acc:=0.86
# Epoch:=6/10  - train loss:=0.0055 - val loss:=0.0063, train acc:=0.8
8 - val acc:=0.84
# Epoch:=7/10  - train loss:=0.0052 - val loss:=0.0068, train acc:=0.9
2 - val acc:=0.82
# Epoch:=8/10  - train loss:=0.0049 - val loss:=0.0061, train acc:=0.8
4 - val acc:=0.86
# Epoch:=9/10  - train loss:=0.0047 - val loss:=0.0065, train acc:=0.8
8 - val acc:=0.90
# Epoch:=10/10  - train loss:=0.0045 - val loss:=0.0063, train acc:=0.
86 - val acc:=0.90

Total time taken (in seconds):  280.99

Finished training model: mlp_on_gpu_RegL2

**************** Testing ****************
mlp_on_gpu_RegL2 model accuracy = 88.07%
*****************************************
```



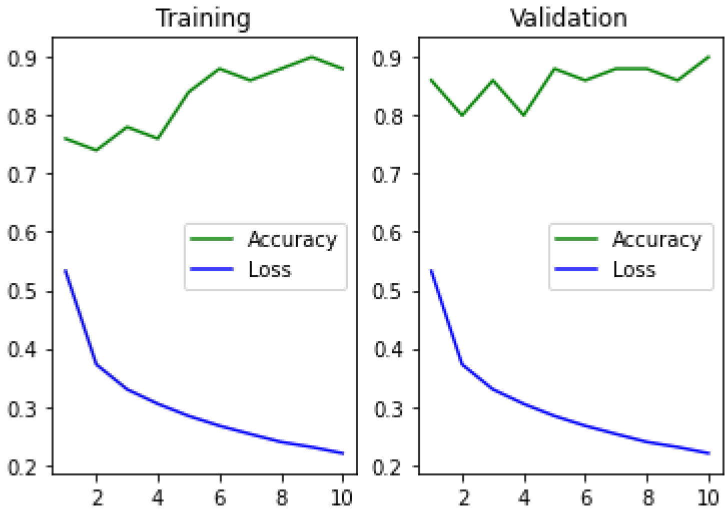
```
Count: 4, j=: 1


**************** Training model: mlp_on_gpu_RegL2 with optimizer: SGD
and seed: 1056 ****************

# Epoch:=1/10  - train loss:=0.0110 - val loss:=0.0081, train acc:=0.7
8 - val acc:=0.86
# Epoch:=2/10  - train loss:=0.0077 - val loss:=0.0074, train acc:=0.7
6 - val acc:=0.86
# Epoch:=3/10  - train loss:=0.0068 - val loss:=0.0072, train acc:=0.8
4 - val acc:=0.86
# Epoch:=4/10  - train loss:=0.0063 - val loss:=0.0070, train acc:=0.8
6 - val acc:=0.90
# Epoch:=5/10  - train loss:=0.0059 - val loss:=0.0068, train acc:=0.8
2 - val acc:=0.88
# Epoch:=6/10  - train loss:=0.0055 - val loss:=0.0066, train acc:=0.8
8 - val acc:=0.84
# Epoch:=7/10  - train loss:=0.0053 - val loss:=0.0063, train acc:=0.8
8 - val acc:=0.92
# Epoch:=8/10  - train loss:=0.0050 - val loss:=0.0065, train acc:=0.8
4 - val acc:=0.90
# Epoch:=9/10  - train loss:=0.0048 - val loss:=0.0066, train acc:=0.8
6 - val acc:=0.86
# Epoch:=10/10  - train loss:=0.0046 - val loss:=0.0063, train acc:=0.
88 - val acc:=0.86

Total time taken (in seconds):  280.91

Finished training model: mlp_on_gpu_RegL2

**************** Testing ****************
mlp_on_gpu_RegL2 model accuracy = 87.75%
*****************************************
```



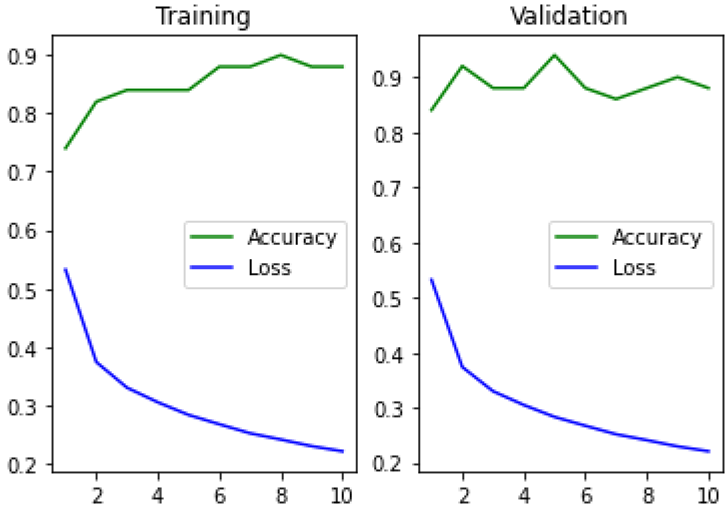
```
Count: 5, j=: 1


**************** Training model: mlp_on_gpu_RegL2 with optimizer: SGD
and seed: 9057 ****************

# Epoch:=1/10  - train loss:=0.0110 - val loss:=0.0082, train acc:=0.7
2 - val acc:=0.80
# Epoch:=2/10  - train loss:=0.0077 - val loss:=0.0077, train acc:=0.8
0 - val acc:=0.92
# Epoch:=3/10  - train loss:=0.0068 - val loss:=0.0074, train acc:=0.8
2 - val acc:=0.90
# Epoch:=4/10  - train loss:=0.0062 - val loss:=0.0066, train acc:=0.8
2 - val acc:=0.88
# Epoch:=5/10  - train loss:=0.0058 - val loss:=0.0067, train acc:=0.8
0 - val acc:=0.88
# Epoch:=6/10  - train loss:=0.0055 - val loss:=0.0066, train acc:=0.8
6 - val acc:=0.88
# Epoch:=7/10  - train loss:=0.0052 - val loss:=0.0064, train acc:=0.8
8 - val acc:=0.88
# Epoch:=8/10  - train loss:=0.0049 - val loss:=0.0062, train acc:=0.8
4 - val acc:=0.88
# Epoch:=9/10  - train loss:=0.0047 - val loss:=0.0064, train acc:=0.8
8 - val acc:=0.88
# Epoch:=10/10  - train loss:=0.0045 - val loss:=0.0066, train acc:=0.
90 - val acc:=0.88

Total time taken (in seconds):  280.41

Finished training model: mlp_on_gpu_RegL2

**************** Testing ****************
mlp_on_gpu_RegL2 model accuracy = 87.57%
*****************************************
```



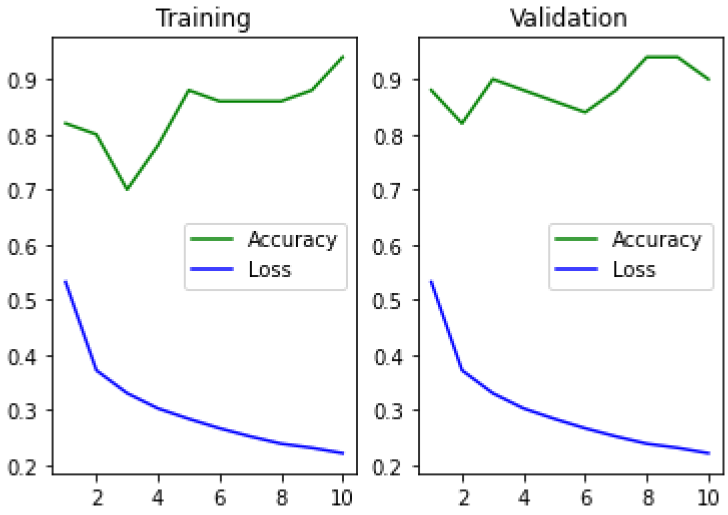
```
Count: 6, j=: 1


**************** Training model: mlp_on_gpu_RegL2 with optimizer: SGD
and seed: 5885 ****************

# Epoch:=1/10  - train loss:=0.0110 - val loss:=0.0085, train acc:=0.7
6 - val acc:=0.86
# Epoch:=2/10  - train loss:=0.0077 - val loss:=0.0077, train acc:=0.7
4 - val acc:=0.80
# Epoch:=3/10  - train loss:=0.0068 - val loss:=0.0074, train acc:=0.7
8 - val acc:=0.86
# Epoch:=4/10  - train loss:=0.0063 - val loss:=0.0074, train acc:=0.7
6 - val acc:=0.80
# Epoch:=5/10  - train loss:=0.0059 - val loss:=0.0065, train acc:=0.8
4 - val acc:=0.88
# Epoch:=6/10  - train loss:=0.0055 - val loss:=0.0064, train acc:=0.8
8 - val acc:=0.86
# Epoch:=7/10  - train loss:=0.0052 - val loss:=0.0062, train acc:=0.8
6 - val acc:=0.88
# Epoch:=8/10  - train loss:=0.0050 - val loss:=0.0063, train acc:=0.8
8 - val acc:=0.88
# Epoch:=9/10  - train loss:=0.0048 - val loss:=0.0067, train acc:=0.9
0 - val acc:=0.86
# Epoch:=10/10  - train loss:=0.0046 - val loss:=0.0063, train acc:=0.
88 - val acc:=0.90

Total time taken (in seconds):  279.93

Finished training model: mlp_on_gpu_RegL2

**************** Testing ****************
mlp_on_gpu_RegL2 model accuracy = 87.29%
*****************************************
```



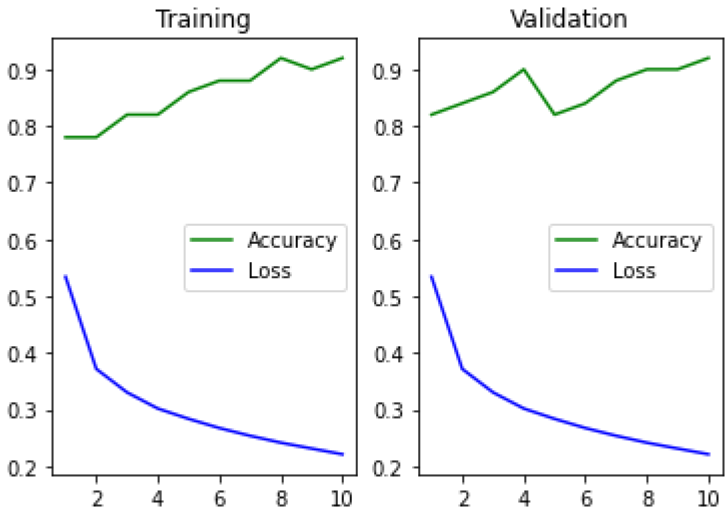
```
Count: 7, j=: 1


**************** Training model: mlp_on_gpu_RegL2 with optimizer: SGD
and seed: 3697 ****************

# Epoch:=1/10  - train loss:=0.0110 - val loss:=0.0085, train acc:=0.7
4 - val acc:=0.84
# Epoch:=2/10  - train loss:=0.0077 - val loss:=0.0079, train acc:=0.8
2 - val acc:=0.92
# Epoch:=3/10  - train loss:=0.0068 - val loss:=0.0071, train acc:=0.8
4 - val acc:=0.88
# Epoch:=4/10  - train loss:=0.0063 - val loss:=0.0070, train acc:=0.8
4 - val acc:=0.88
# Epoch:=5/10  - train loss:=0.0058 - val loss:=0.0067, train acc:=0.8
4 - val acc:=0.94
# Epoch:=6/10  - train loss:=0.0055 - val loss:=0.0064, train acc:=0.8
8 - val acc:=0.88
# Epoch:=7/10  - train loss:=0.0052 - val loss:=0.0064, train acc:=0.8
8 - val acc:=0.86
# Epoch:=8/10  - train loss:=0.0050 - val loss:=0.0060, train acc:=0.9
0 - val acc:=0.88
# Epoch:=9/10  - train loss:=0.0047 - val loss:=0.0062, train acc:=0.8
8 - val acc:=0.90
# Epoch:=10/10  - train loss:=0.0045 - val loss:=0.0067, train acc:=0.
88 - val acc:=0.88

Total time taken (in seconds):  280.06

Finished training model: mlp_on_gpu_RegL2

**************** Testing ****************
mlp_on_gpu_RegL2 model accuracy = 87.11%
*****************************************
```



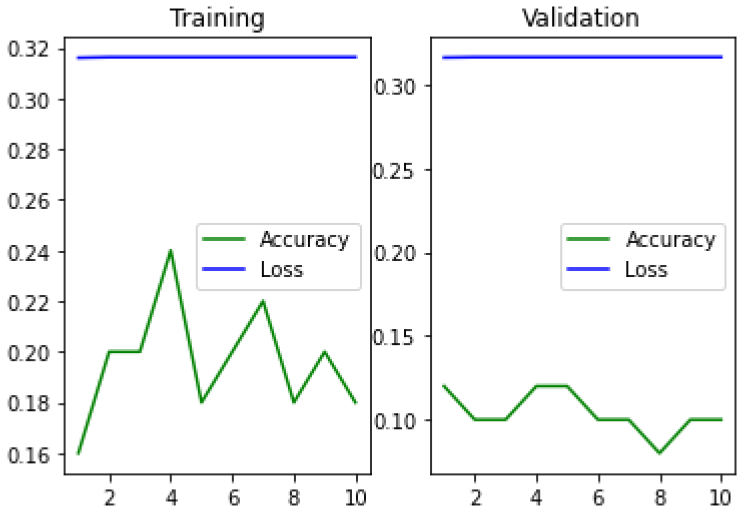
```
Count: 8, j=: 1


**************** Training model: mlp_on_gpu_RegL2 with optimizer: SGD
and seed: 2053 ****************

# Epoch:=1/10  - train loss:=0.0110 - val loss:=0.0081, train acc:=0.8
2 - val acc:=0.88
# Epoch:=2/10  - train loss:=0.0077 - val loss:=0.0078, train acc:=0.8
0 - val acc:=0.82
# Epoch:=3/10  - train loss:=0.0068 - val loss:=0.0077, train acc:=0.7
0 - val acc:=0.90
# Epoch:=4/10  - train loss:=0.0063 - val loss:=0.0068, train acc:=0.7
8 - val acc:=0.88
# Epoch:=5/10  - train loss:=0.0059 - val loss:=0.0066, train acc:=0.8
8 - val acc:=0.86
# Epoch:=6/10  - train loss:=0.0055 - val loss:=0.0063, train acc:=0.8
6 - val acc:=0.84
# Epoch:=7/10  - train loss:=0.0052 - val loss:=0.0065, train acc:=0.8
6 - val acc:=0.88
# Epoch:=8/10  - train loss:=0.0050 - val loss:=0.0065, train acc:=0.8
6 - val acc:=0.94
# Epoch:=9/10  - train loss:=0.0048 - val loss:=0.0062, train acc:=0.8
8 - val acc:=0.94
# Epoch:=10/10  - train loss:=0.0046 - val loss:=0.0064, train acc:=0.
94 - val acc:=0.90

Total time taken (in seconds):  283.77

Finished training model: mlp_on_gpu_RegL2

**************** Testing ****************
mlp_on_gpu_RegL2 model accuracy = 87.58%
*****************************************
```



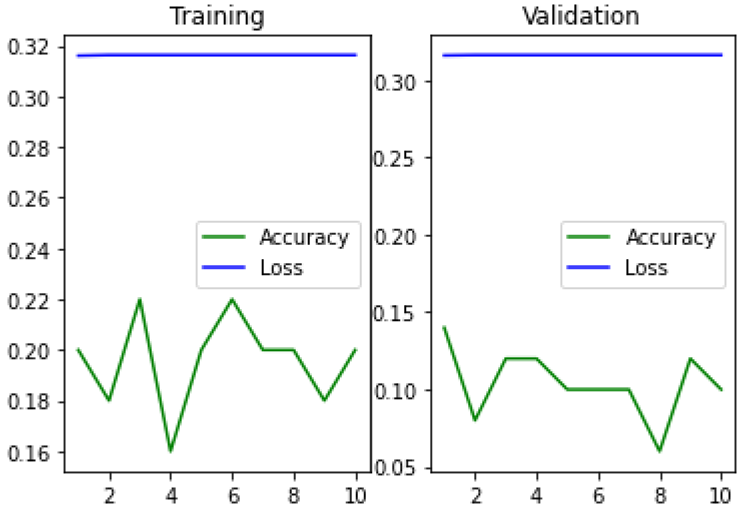
```
Count: 9, j=: 1


**************** Training model: mlp_on_gpu_RegL2 with optimizer: SGD
and seed: 6857 ****************

# Epoch:=1/10  - train loss:=0.0111 - val loss:=0.0085, train acc:=0.7
8 - val acc:=0.82
# Epoch:=2/10  - train loss:=0.0077 - val loss:=0.0078, train acc:=0.7
8 - val acc:=0.84
# Epoch:=3/10  - train loss:=0.0068 - val loss:=0.0071, train acc:=0.8
2 - val acc:=0.86
# Epoch:=4/10  - train loss:=0.0062 - val loss:=0.0067, train acc:=0.8
2 - val acc:=0.90
# Epoch:=5/10  - train loss:=0.0059 - val loss:=0.0071, train acc:=0.8
6 - val acc:=0.82
# Epoch:=6/10  - train loss:=0.0055 - val loss:=0.0069, train acc:=0.8
8 - val acc:=0.84
# Epoch:=7/10  - train loss:=0.0053 - val loss:=0.0068, train acc:=0.8
8 - val acc:=0.88
# Epoch:=8/10  - train loss:=0.0050 - val loss:=0.0062, train acc:=0.9
2 - val acc:=0.90
# Epoch:=9/10  - train loss:=0.0048 - val loss:=0.0065, train acc:=0.9
0 - val acc:=0.90
# Epoch:=10/10  - train loss:=0.0046 - val loss:=0.0062, train acc:=0.
92 - val acc:=0.92

Total time taken (in seconds):  283.40

Finished training model: mlp_on_gpu_RegL2

**************** Testing ****************
mlp_on_gpu_RegL2 model accuracy = 88.15%
*****************************************
```



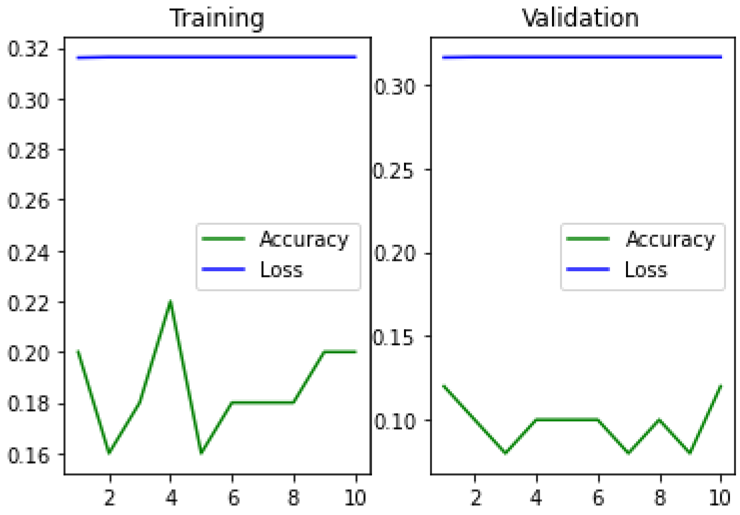
```
Count: 0, j=: 1


**************** Training model: mlp_on_gpu_default with optimizer: Ad
am and seed: 8532 ****************

# Epoch:=1/10  - train loss:=0.2899 - val loss:=0.2903, train acc:=0.1
6 - val acc:=0.12
# Epoch:=2/10  - train loss:=0.2901 - val loss:=0.2903, train acc:=0.2
0 - val acc:=0.10
# Epoch:=3/10  - train loss:=0.2901 - val loss:=0.2903, train acc:=0.2
0 - val acc:=0.10
# Epoch:=4/10  - train loss:=0.2901 - val loss:=0.2903, train acc:=0.2
4 - val acc:=0.12
# Epoch:=5/10  - train loss:=0.2901 - val loss:=0.2903, train acc:=0.1
8 - val acc:=0.12
# Epoch:=6/10  - train loss:=0.2901 - val loss:=0.2903, train acc:=0.2
0 - val acc:=0.10
# Epoch:=7/10  - train loss:=0.2901 - val loss:=0.2903, train acc:=0.2
2 - val acc:=0.10
# Epoch:=8/10  - train loss:=0.2901 - val loss:=0.2903, train acc:=0.1
8 - val acc:=0.08
# Epoch:=9/10  - train loss:=0.2901 - val loss:=0.2903, train acc:=0.2
0 - val acc:=0.10
# Epoch:=10/10  - train loss:=0.2901 - val loss:=0.2903, train acc:=0.
18 - val acc:=0.10

Total time taken (in seconds):  315.20

Finished training model: mlp_on_gpu_default

**************** Testing ****************
mlp_on_gpu_default model accuracy = 10.00%
*****************************************
```



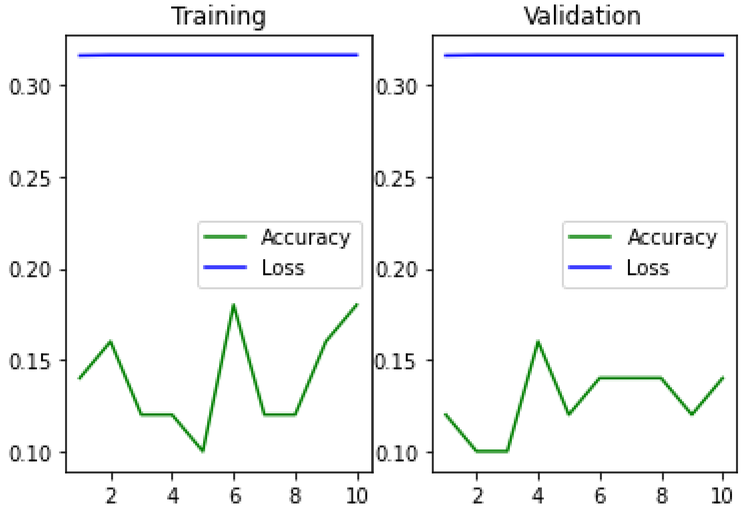
```
Count: 1, j=: 1


**************** Training model: mlp_on_gpu_default with optimizer: Ad
am and seed: 3298 ****************

# Epoch:=1/10  - train loss:=0.2899 - val loss:=0.2903, train acc:=0.2
0 - val acc:=0.14
# Epoch:=2/10  - train loss:=0.2901 - val loss:=0.2903, train acc:=0.1
8 - val acc:=0.08
# Epoch:=3/10  - train loss:=0.2901 - val loss:=0.2903, train acc:=0.2
2 - val acc:=0.12
# Epoch:=4/10  - train loss:=0.2901 - val loss:=0.2903, train acc:=0.1
6 - val acc:=0.12
# Epoch:=5/10  - train loss:=0.2901 - val loss:=0.2903, train acc:=0.2
0 - val acc:=0.10
# Epoch:=6/10  - train loss:=0.2901 - val loss:=0.2903, train acc:=0.2
2 - val acc:=0.10
# Epoch:=7/10  - train loss:=0.2901 - val loss:=0.2903, train acc:=0.2
0 - val acc:=0.10
# Epoch:=8/10  - train loss:=0.2901 - val loss:=0.2903, train acc:=0.2
0 - val acc:=0.06
# Epoch:=9/10  - train loss:=0.2901 - val loss:=0.2903, train acc:=0.1
8 - val acc:=0.12
# Epoch:=10/10  - train loss:=0.2901 - val loss:=0.2903, train acc:=0.
20 - val acc:=0.10

Total time taken (in seconds):  313.55

Finished training model: mlp_on_gpu_default

**************** Testing ****************
mlp_on_gpu_default model accuracy = 10.00%
*****************************************
```



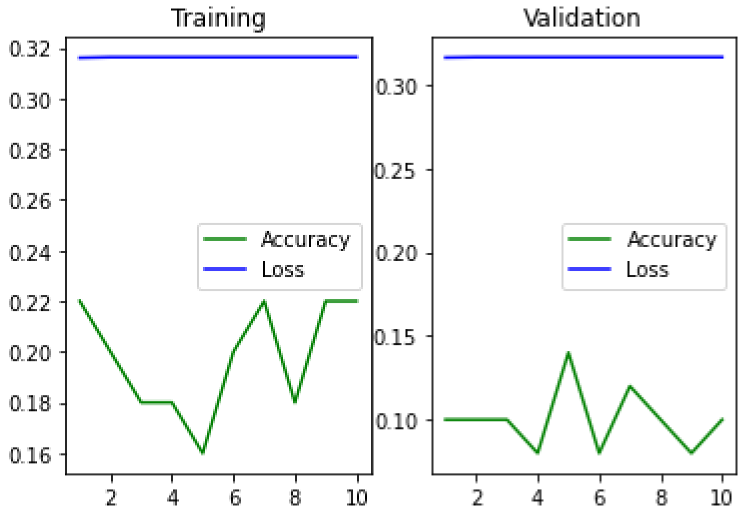
```
Count: 2, j=: 1


**************** Training model: mlp_on_gpu_default with optimizer: Ad
am and seed: 8562 ****************

# Epoch:=1/10  - train loss:=0.2899 - val loss:=0.2903, train acc:=0.2
0 - val acc:=0.12
# Epoch:=2/10  - train loss:=0.2901 - val loss:=0.2903, train acc:=0.1
6 - val acc:=0.10
# Epoch:=3/10  - train loss:=0.2901 - val loss:=0.2903, train acc:=0.1
8 - val acc:=0.08
# Epoch:=4/10  - train loss:=0.2901 - val loss:=0.2903, train acc:=0.2
2 - val acc:=0.10
# Epoch:=5/10  - train loss:=0.2901 - val loss:=0.2903, train acc:=0.1
6 - val acc:=0.10
# Epoch:=6/10  - train loss:=0.2901 - val loss:=0.2903, train acc:=0.1
8 - val acc:=0.10
# Epoch:=7/10  - train loss:=0.2901 - val loss:=0.2903, train acc:=0.1
8 - val acc:=0.08
# Epoch:=8/10  - train loss:=0.2901 - val loss:=0.2903, train acc:=0.1
8 - val acc:=0.10
# Epoch:=9/10  - train loss:=0.2901 - val loss:=0.2903, train acc:=0.2
0 - val acc:=0.08
# Epoch:=10/10  - train loss:=0.2901 - val loss:=0.2903, train acc:=0.
20 - val acc:=0.12

Total time taken (in seconds):  313.80

Finished training model: mlp_on_gpu_default

**************** Testing ****************
mlp_on_gpu_default model accuracy = 10.00%
*****************************************
```



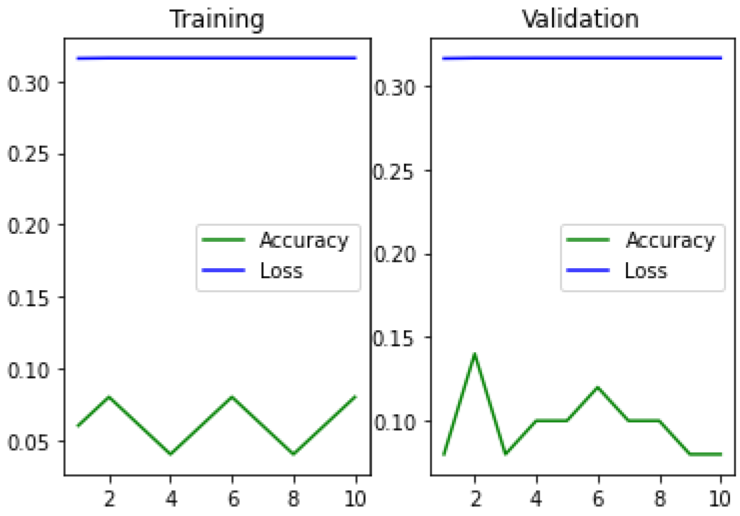
```
Count: 3, j=: 1


**************** Training model: mlp_on_gpu_default with optimizer: Ad
am and seed: 6144 ****************

# Epoch:=1/10  - train loss:=0.2897 - val loss:=0.2911, train acc:=0.1
4 - val acc:=0.12
# Epoch:=2/10  - train loss:=0.2899 - val loss:=0.2911, train acc:=0.1
6 - val acc:=0.10
# Epoch:=3/10  - train loss:=0.2899 - val loss:=0.2911, train acc:=0.1
2 - val acc:=0.10
# Epoch:=4/10  - train loss:=0.2899 - val loss:=0.2911, train acc:=0.1
2 - val acc:=0.16
# Epoch:=5/10  - train loss:=0.2899 - val loss:=0.2911, train acc:=0.1
0 - val acc:=0.12
# Epoch:=6/10  - train loss:=0.2899 - val loss:=0.2911, train acc:=0.1
8 - val acc:=0.14
# Epoch:=7/10  - train loss:=0.2899 - val loss:=0.2911, train acc:=0.1
2 - val acc:=0.14
# Epoch:=8/10  - train loss:=0.2899 - val loss:=0.2911, train acc:=0.1
2 - val acc:=0.14
# Epoch:=9/10  - train loss:=0.2899 - val loss:=0.2911, train acc:=0.1
6 - val acc:=0.12
# Epoch:=10/10  - train loss:=0.2899 - val loss:=0.2911, train acc:=0.
18 - val acc:=0.14

Total time taken (in seconds):  324.20

Finished training model: mlp_on_gpu_default

**************** Testing ****************
mlp_on_gpu_default model accuracy = 10.00%
*****************************************
```



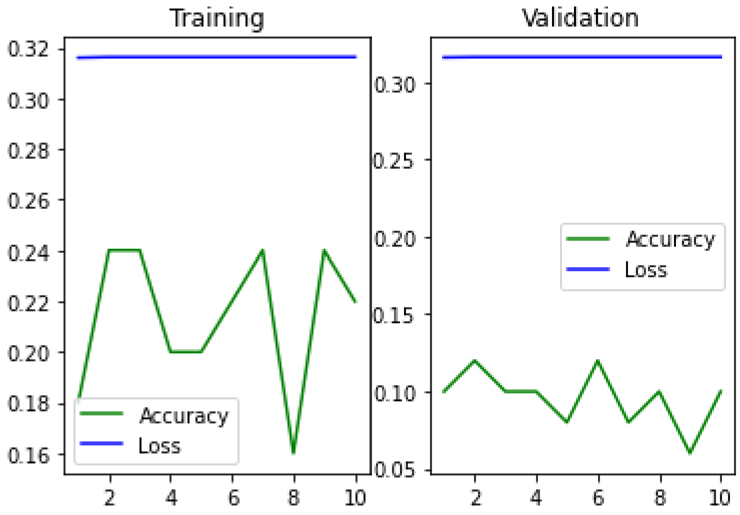
```
Count: 4, j=: 1


**************** Training model: mlp_on_gpu_default with optimizer: Ad
am and seed: 4574 ****************

# Epoch:=1/10  - train loss:=0.2899 - val loss:=0.2903, train acc:=0.2
2 - val acc:=0.10
# Epoch:=2/10  - train loss:=0.2901 - val loss:=0.2903, train acc:=0.2
0 - val acc:=0.10
# Epoch:=3/10  - train loss:=0.2901 - val loss:=0.2903, train acc:=0.1
8 - val acc:=0.10
# Epoch:=4/10  - train loss:=0.2901 - val loss:=0.2903, train acc:=0.1
8 - val acc:=0.08
# Epoch:=5/10  - train loss:=0.2901 - val loss:=0.2903, train acc:=0.1
6 - val acc:=0.14
# Epoch:=6/10  - train loss:=0.2901 - val loss:=0.2903, train acc:=0.2
0 - val acc:=0.08
# Epoch:=7/10  - train loss:=0.2901 - val loss:=0.2903, train acc:=0.2
2 - val acc:=0.12
# Epoch:=8/10  - train loss:=0.2901 - val loss:=0.2903, train acc:=0.1
8 - val acc:=0.10
# Epoch:=9/10  - train loss:=0.2901 - val loss:=0.2903, train acc:=0.2
2 - val acc:=0.08
# Epoch:=10/10  - train loss:=0.2901 - val loss:=0.2903, train acc:=0.
22 - val acc:=0.10

Total time taken (in seconds):  325.47

Finished training model: mlp_on_gpu_default

**************** Testing ****************
mlp_on_gpu_default model accuracy = 10.00%
*****************************************
```



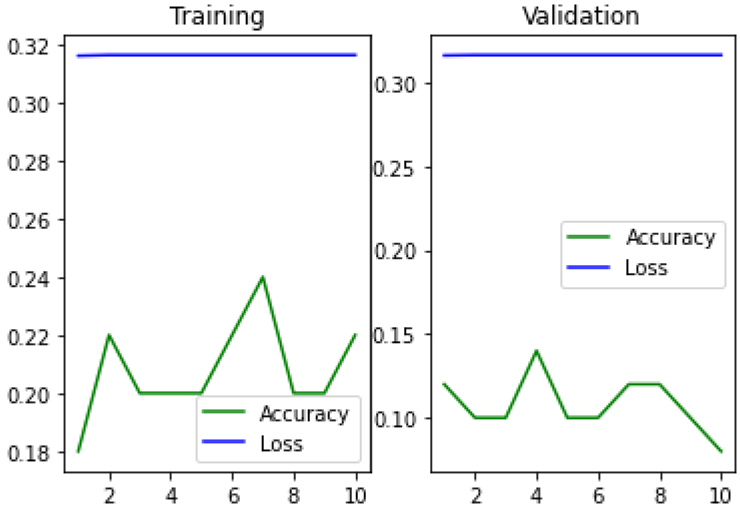
```
Count: 5, j=: 1


**************** Training model: mlp_on_gpu_default with optimizer: Ad
am and seed: 2230 ****************

# Epoch:=1/10  - train loss:=0.2901 - val loss:=0.2894, train acc:=0.0
6 - val acc:=0.08
# Epoch:=2/10  - train loss:=0.2903 - val loss:=0.2894, train acc:=0.0
8 - val acc:=0.14
# Epoch:=3/10  - train loss:=0.2903 - val loss:=0.2894, train acc:=0.0
6 - val acc:=0.08
# Epoch:=4/10  - train loss:=0.2903 - val loss:=0.2894, train acc:=0.0
4 - val acc:=0.10
# Epoch:=5/10  - train loss:=0.2903 - val loss:=0.2894, train acc:=0.0
6 - val acc:=0.10
# Epoch:=6/10  - train loss:=0.2903 - val loss:=0.2894, train acc:=0.0
8 - val acc:=0.12
# Epoch:=7/10  - train loss:=0.2903 - val loss:=0.2894, train acc:=0.0
6 - val acc:=0.10
# Epoch:=8/10  - train loss:=0.2903 - val loss:=0.2894, train acc:=0.0
4 - val acc:=0.10
# Epoch:=9/10  - train loss:=0.2903 - val loss:=0.2894, train acc:=0.0
6 - val acc:=0.08
# Epoch:=10/10  - train loss:=0.2903 - val loss:=0.2894, train acc:=0.
08 - val acc:=0.08

Total time taken (in seconds):  320.05

Finished training model: mlp_on_gpu_default

**************** Testing ****************
mlp_on_gpu_default model accuracy = 10.00%
*****************************************
```



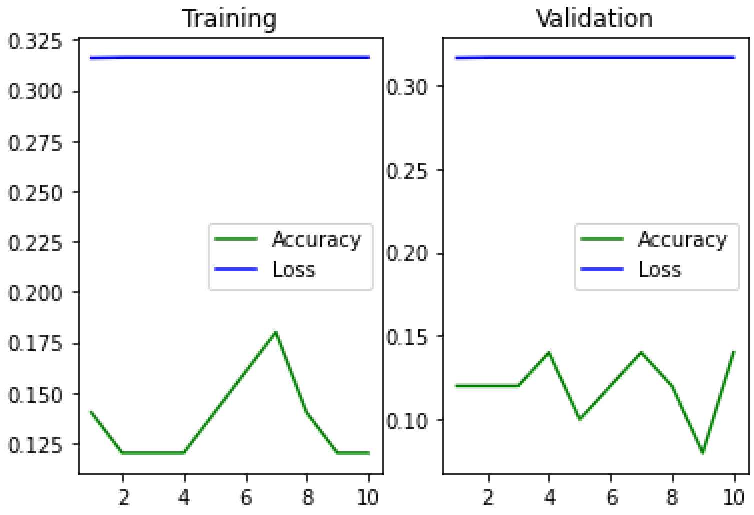
```
Count: 6, j=: 1


**************** Training model: mlp_on_gpu_default with optimizer: Ad
am and seed: 5566 ****************

# Epoch:=1/10  - train loss:=0.2899 - val loss:=0.2903, train acc:=0.1
8 - val acc:=0.10
# Epoch:=2/10  - train loss:=0.2901 - val loss:=0.2903, train acc:=0.2
4 - val acc:=0.12
# Epoch:=3/10  - train loss:=0.2901 - val loss:=0.2903, train acc:=0.2
4 - val acc:=0.10
# Epoch:=4/10  - train loss:=0.2901 - val loss:=0.2903, train acc:=0.2
0 - val acc:=0.10
# Epoch:=5/10  - train loss:=0.2901 - val loss:=0.2903, train acc:=0.2
0 - val acc:=0.08
# Epoch:=6/10  - train loss:=0.2901 - val loss:=0.2903, train acc:=0.2
2 - val acc:=0.12
# Epoch:=7/10  - train loss:=0.2901 - val loss:=0.2903, train acc:=0.2
4 - val acc:=0.08
# Epoch:=8/10  - train loss:=0.2901 - val loss:=0.2903, train acc:=0.1
6 - val acc:=0.10
# Epoch:=9/10  - train loss:=0.2901 - val loss:=0.2903, train acc:=0.2
4 - val acc:=0.06
# Epoch:=10/10  - train loss:=0.2901 - val loss:=0.2903, train acc:=0.
22 - val acc:=0.10

Total time taken (in seconds):  320.46

Finished training model: mlp_on_gpu_default

**************** Testing ****************
mlp_on_gpu_default model accuracy = 10.00%
*****************************************
```



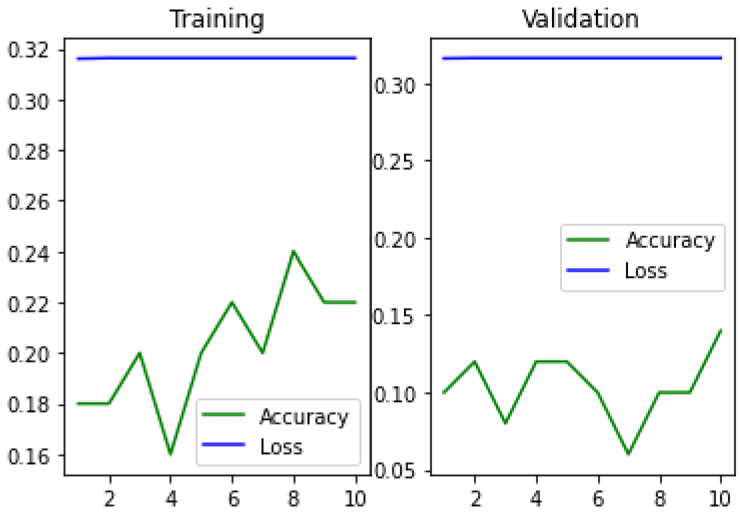
```
Count: 7, j=: 1


**************** Training model: mlp_on_gpu_default with optimizer: Ad
am and seed: 8194 ****************

# Epoch:=1/10  - train loss:=0.2899 - val loss:=0.2903, train acc:=0.1
8 - val acc:=0.12
# Epoch:=2/10  - train loss:=0.2901 - val loss:=0.2903, train acc:=0.2
2 - val acc:=0.10
# Epoch:=3/10  - train loss:=0.2901 - val loss:=0.2903, train acc:=0.2
0 - val acc:=0.10
# Epoch:=4/10  - train loss:=0.2901 - val loss:=0.2903, train acc:=0.2
0 - val acc:=0.14
# Epoch:=5/10  - train loss:=0.2901 - val loss:=0.2903, train acc:=0.2
0 - val acc:=0.10
# Epoch:=6/10  - train loss:=0.2901 - val loss:=0.2903, train acc:=0.2
2 - val acc:=0.10
# Epoch:=7/10  - train loss:=0.2901 - val loss:=0.2903, train acc:=0.2
4 - val acc:=0.12
# Epoch:=8/10  - train loss:=0.2901 - val loss:=0.2903, train acc:=0.2
0 - val acc:=0.12
# Epoch:=9/10  - train loss:=0.2901 - val loss:=0.2903, train acc:=0.2
0 - val acc:=0.10
# Epoch:=10/10  - train loss:=0.2901 - val loss:=0.2903, train acc:=0.
22 - val acc:=0.08

Total time taken (in seconds):  315.58

Finished training model: mlp_on_gpu_default

**************** Testing ****************
mlp_on_gpu_default model accuracy = 10.00%
*****************************************
```



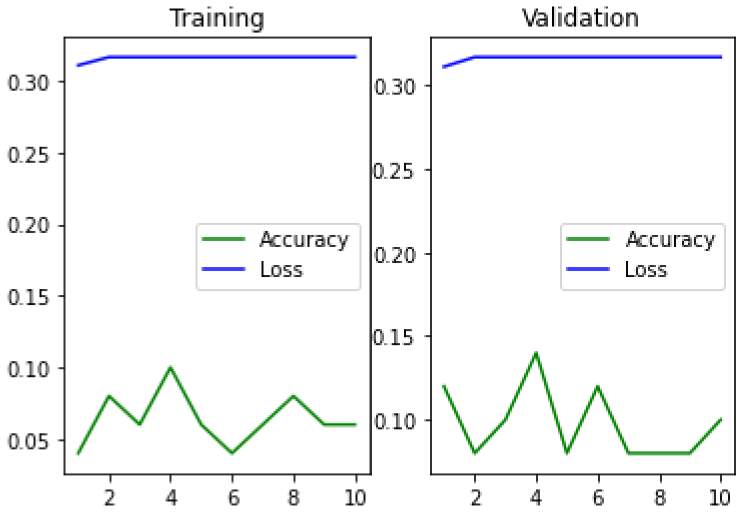
```
Count: 8, j=: 1


**************** Training model: mlp_on_gpu_default with optimizer: Ad
am and seed: 1399 ****************

# Epoch:=1/10  - train loss:=0.2897 - val loss:=0.2911, train acc:=0.1
4 - val acc:=0.12
# Epoch:=2/10  - train loss:=0.2899 - val loss:=0.2911, train acc:=0.1
2 - val acc:=0.12
# Epoch:=3/10  - train loss:=0.2899 - val loss:=0.2911, train acc:=0.1
2 - val acc:=0.12
# Epoch:=4/10  - train loss:=0.2899 - val loss:=0.2911, train acc:=0.1
2 - val acc:=0.14
# Epoch:=5/10  - train loss:=0.2899 - val loss:=0.2911, train acc:=0.1
4 - val acc:=0.10
# Epoch:=6/10  - train loss:=0.2899 - val loss:=0.2911, train acc:=0.1
6 - val acc:=0.12
# Epoch:=7/10  - train loss:=0.2899 - val loss:=0.2911, train acc:=0.1
8 - val acc:=0.14
# Epoch:=8/10  - train loss:=0.2899 - val loss:=0.2911, train acc:=0.1
4 - val acc:=0.12
# Epoch:=9/10  - train loss:=0.2899 - val loss:=0.2911, train acc:=0.1
2 - val acc:=0.08
# Epoch:=10/10  - train loss:=0.2899 - val loss:=0.2911, train acc:=0.
12 - val acc:=0.14

Total time taken (in seconds):  314.74

Finished training model: mlp_on_gpu_default

**************** Testing ****************
mlp_on_gpu_default model accuracy = 10.00%
*****************************************
```



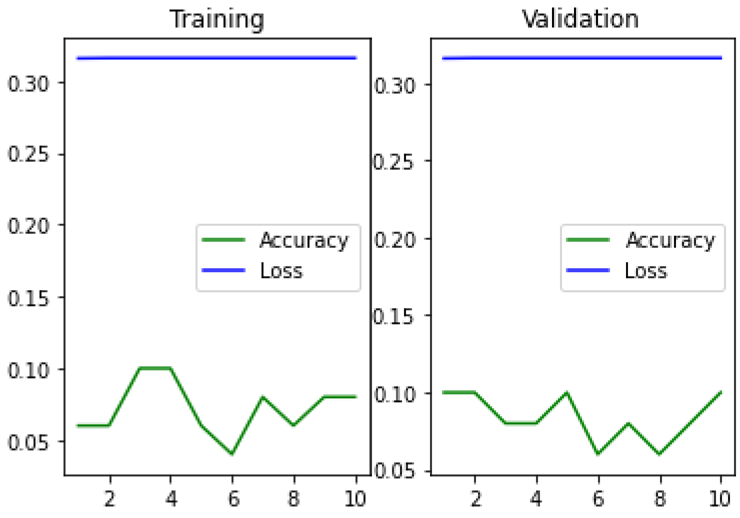
```
Count: 9, j=: 1


**************** Training model: mlp_on_gpu_default with optimizer: Ad
am and seed: 6249 ****************

# Epoch:=1/10  - train loss:=0.2899 - val loss:=0.2903, train acc:=0.1
8 - val acc:=0.10
# Epoch:=2/10  - train loss:=0.2901 - val loss:=0.2903, train acc:=0.1
8 - val acc:=0.12
# Epoch:=3/10  - train loss:=0.2901 - val loss:=0.2903, train acc:=0.2
0 - val acc:=0.08
# Epoch:=4/10  - train loss:=0.2901 - val loss:=0.2903, train acc:=0.1
6 - val acc:=0.12
# Epoch:=5/10  - train loss:=0.2901 - val loss:=0.2903, train acc:=0.2
0 - val acc:=0.12
# Epoch:=6/10  - train loss:=0.2901 - val loss:=0.2903, train acc:=0.2
2 - val acc:=0.10
# Epoch:=7/10  - train loss:=0.2901 - val loss:=0.2903, train acc:=0.2
0 - val acc:=0.06
# Epoch:=8/10  - train loss:=0.2901 - val loss:=0.2903, train acc:=0.2
4 - val acc:=0.10
# Epoch:=9/10  - train loss:=0.2901 - val loss:=0.2903, train acc:=0.2
2 - val acc:=0.10
# Epoch:=10/10  - train loss:=0.2901 - val loss:=0.2903, train acc:=0.
22 - val acc:=0.14

Total time taken (in seconds):  313.54

Finished training model: mlp_on_gpu_default

**************** Testing ****************
mlp_on_gpu_default model accuracy = 10.00%
*****************************************
```



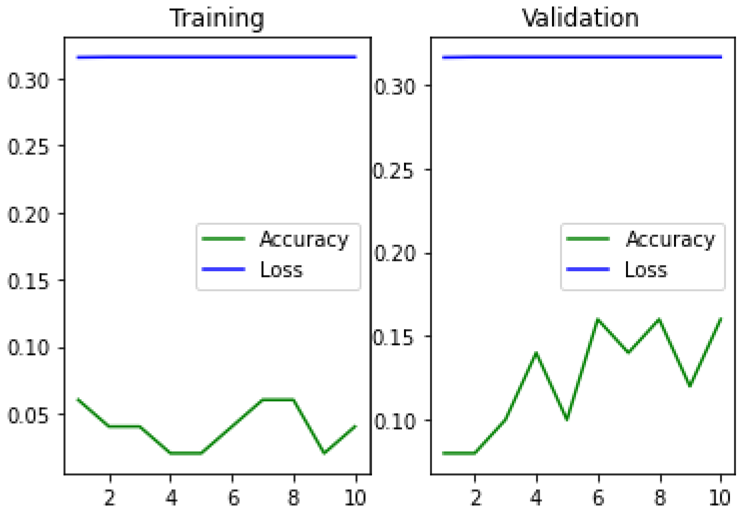
```
Count: 0, j=: 1


**************** Training model: mlp_on_gpu_RegL1 with optimizer: Adam
and seed: 2717 ****************

# Epoch:=1/10  - train loss:=0.2850 - val loss:=0.2894, train acc:=0.0
4 - val acc:=0.12
# Epoch:=2/10  - train loss:=0.2903 - val loss:=0.2894, train acc:=0.0
8 - val acc:=0.08
# Epoch:=3/10  - train loss:=0.2903 - val loss:=0.2894, train acc:=0.0
6 - val acc:=0.10
# Epoch:=4/10  - train loss:=0.2903 - val loss:=0.2894, train acc:=0.1
0 - val acc:=0.14
# Epoch:=5/10  - train loss:=0.2903 - val loss:=0.2894, train acc:=0.0
6 - val acc:=0.08
# Epoch:=6/10  - train loss:=0.2903 - val loss:=0.2894, train acc:=0.0
4 - val acc:=0.12
# Epoch:=7/10  - train loss:=0.2903 - val loss:=0.2894, train acc:=0.0
6 - val acc:=0.08
# Epoch:=8/10  - train loss:=0.2903 - val loss:=0.2894, train acc:=0.0
8 - val acc:=0.08
# Epoch:=9/10  - train loss:=0.2903 - val loss:=0.2894, train acc:=0.0
6 - val acc:=0.08
# Epoch:=10/10  - train loss:=0.2903 - val loss:=0.2894, train acc:=0.
06 - val acc:=0.10

Total time taken (in seconds):  357.64

Finished training model: mlp_on_gpu_RegL1

**************** Testing ****************
mlp_on_gpu_RegL1 model accuracy = 10.00%
*****************************************
```



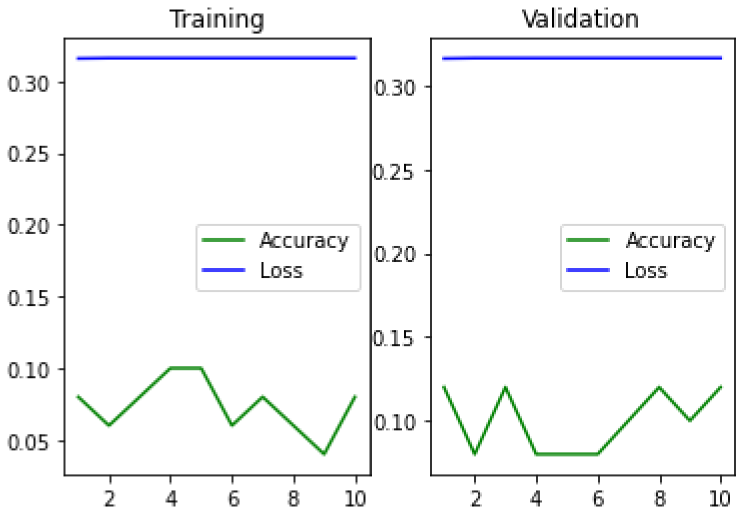
```
Count: 1, j=: 1


**************** Training model: mlp_on_gpu_RegL1 with optimizer: Adam
and seed: 1888 ****************

# Epoch:=1/10  - train loss:=0.2901 - val loss:=0.2894, train acc:=0.0
6 - val acc:=0.10
# Epoch:=2/10  - train loss:=0.2903 - val loss:=0.2894, train acc:=0.0
6 - val acc:=0.10
# Epoch:=3/10  - train loss:=0.2903 - val loss:=0.2894, train acc:=0.1
0 - val acc:=0.08
# Epoch:=4/10  - train loss:=0.2903 - val loss:=0.2894, train acc:=0.1
0 - val acc:=0.08
# Epoch:=5/10  - train loss:=0.2903 - val loss:=0.2894, train acc:=0.0
6 - val acc:=0.10
# Epoch:=6/10  - train loss:=0.2903 - val loss:=0.2894, train acc:=0.0
4 - val acc:=0.06
# Epoch:=7/10  - train loss:=0.2903 - val loss:=0.2894, train acc:=0.0
8 - val acc:=0.08
# Epoch:=8/10  - train loss:=0.2903 - val loss:=0.2894, train acc:=0.0
6 - val acc:=0.06
# Epoch:=9/10  - train loss:=0.2903 - val loss:=0.2894, train acc:=0.0
8 - val acc:=0.08
# Epoch:=10/10  - train loss:=0.2903 - val loss:=0.2894, train acc:=0.
08 - val acc:=0.10

Total time taken (in seconds):  357.74

Finished training model: mlp_on_gpu_RegL1

**************** Testing ****************
mlp_on_gpu_RegL1 model accuracy = 10.00%
*****************************************
```



```
Count: 2, j=: 1


**************** Training model: mlp_on_gpu_RegL1 with optimizer: Adam
and seed: 9527 ****************

# Epoch:=1/10  - train loss:=0.2900 - val loss:=0.2894, train acc:=0.0
6 - val acc:=0.08
# Epoch:=2/10  - train loss:=0.2903 - val loss:=0.2894, train acc:=0.0
4 - val acc:=0.08
# Epoch:=3/10  - train loss:=0.2903 - val loss:=0.2894, train acc:=0.0
4 - val acc:=0.10
# Epoch:=4/10  - train loss:=0.2903 - val loss:=0.2894, train acc:=0.0
2 - val acc:=0.14
# Epoch:=5/10  - train loss:=0.2903 - val loss:=0.2894, train acc:=0.0
2 - val acc:=0.10
# Epoch:=6/10  - train loss:=0.2903 - val loss:=0.2894, train acc:=0.0
4 - val acc:=0.16
# Epoch:=7/10  - train loss:=0.2903 - val loss:=0.2894, train acc:=0.0
6 - val acc:=0.14
# Epoch:=8/10  - train loss:=0.2903 - val loss:=0.2894, train acc:=0.0
6 - val acc:=0.16
# Epoch:=9/10  - train loss:=0.2903 - val loss:=0.2894, train acc:=0.0
2 - val acc:=0.12
# Epoch:=10/10  - train loss:=0.2903 - val loss:=0.2894, train acc:=0.
04 - val acc:=0.16

Total time taken (in seconds):  361.89

Finished training model: mlp_on_gpu_RegL1

**************** Testing ****************
mlp_on_gpu_RegL1 model accuracy = 10.00%
*****************************************
```



```
Count: 3, j=: 1


**************** Training model: mlp_on_gpu_RegL1 with optimizer: Adam
and seed: 7862 ****************

# Epoch:=1/10  - train loss:=0.2900 - val loss:=0.2894, train acc:=0.0
8 - val acc:=0.12
# Epoch:=2/10  - train loss:=0.2903 - val loss:=0.2894, train acc:=0.0
6 - val acc:=0.08
# Epoch:=3/10  - train loss:=0.2903 - val loss:=0.2894, train acc:=0.0
8 - val acc:=0.12
# Epoch:=4/10  - train loss:=0.2903 - val loss:=0.2894, train acc:=0.1
0 - val acc:=0.08
# Epoch:=5/10  - train loss:=0.2903 - val loss:=0.2894, train acc:=0.1
0 - val acc:=0.08
# Epoch:=6/10  - train loss:=0.2903 - val loss:=0.2894, train acc:=0.0
6 - val acc:=0.08
# Epoch:=7/10  - train loss:=0.2903 - val loss:=0.2894, train acc:=0.0
8 - val acc:=0.10
# Epoch:=8/10  - train loss:=0.2903 - val loss:=0.2894, train acc:=0.0
6 - val acc:=0.12
# Epoch:=9/10  - train loss:=0.2903 - val loss:=0.2894, train acc:=0.0
4 - val acc:=0.10
# Epoch:=10/10  - train loss:=0.2903 - val loss:=0.2894, train acc:=0.
08 - val acc:=0.12

Total time taken (in seconds):  365.30

Finished training model: mlp_on_gpu_RegL1

**************** Testing ****************
mlp_on_gpu_RegL1 model accuracy = 10.00%
*****************************************
```



```
Count: 4, j=: 1


**************** Training model: mlp_on_gpu_RegL1 with optimizer: Adam
and seed: 5226 ****************
```