CC3092 - Deep Learning y Sistemas Inteligentes

Deep Learning y Sistemas Inteligentes

- Hoja de Trabajo 2 -

Integrantes:

- Christopher García (Chrisito uwu)
- Alejandro Gómez (Alecraft)
- Gabriel Vicente (Gabín)

Se realizó 12 combinaciones en el análisis, lo cual corresponde a un porcentaje de puntos extras.

Instrucciones:

- Esta es una actividad en grupos de 3 personas máximo
- No se permitirá ni se aceptará cualquier indicio de copia. De presentarse, se procederá según el reglamento correspondiente.
- Tendrán hasta el día indicado en Canvas. Ejercicio 1 - Experimentación Práctica En esta actividad, implementará y comparará diferentes funciones de pérdida y técnicas de regularización utilizando PyTorch. Utilizará el conjunto de datos de Iris para una tarea de clasificación y una arquitectura básica de red neuronal de feedforward. El objetivo es observar cómo las diferentes opciones impactan la convergencia y el rendimiento del modelo.

Ejercicio 1 - Experimentación Práctica

En esta actividad, implementará y comparará diferentes funciones de pérdida y técnicas de regularización utilizando PyTorch. Utilizará el conjunto de datos de Iris para una tarea de clasificación y una arquitectura básica de red neuronal de feedforward. El objetivo es observar cómo las diferentes opciones impactan la convergencia y el rendimiento del modelo.

Task 1 - Preparación del conjunto de datos

Cargue el conjunto de datos de Iris utilizando bibliotecas como sklearn.datasets. Luego, divida el conjunto de datos en conjuntos de entrenamiento y validación.

In [ ]:
```
""" Librerias necesarias """

import numpy as np
import torch
import torch.nn as nn
import torch.optim as optim
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from torch.utils.data import DataLoader, TensorDataset
```

In [ ]:
```
""" Load data """

iris = load_iris()
X = iris.data
y = iris.target

X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=0.2, random_s

print("Tamaño del conjunto de entrenamiento:", X_train.shape)
print("Tamaño del conjunto de validación:", X_val.shape)


X_train_tensor = torch.tensor(X_train, dtype=torch.float32)
y_train_tensor = torch.tensor(y_train, dtype=torch.long)
X_val_tensor = torch.tensor(X_val, dtype=torch.float32)
y_val_tensor = torch.tensor(y_val, dtype=torch.long)

train_dataset = TensorDataset(X_train_tensor, y_train_tensor)
val_dataset = TensorDataset(X_val_tensor, y_val_tensor)
batch_size = 16
train_loader = DataLoader(train_dataset, batch_size=batch_size, shuffle=True)
val_loader = DataLoader(val_dataset, batch_size=batch_size, shuffle=False)
```

```
Tamaño del conjunto de entrenamiento: (120, 4)
Tamaño del conjunto de validación: (30, 4)
```

Task 2 - Arquitectura modelo

Cree una red neuronal feedforward simple utilizando nn.Module de PyTorch. Luego, defina capa de entrada, capas ocultas y capa de salida. Después, elija las funciones de activación y el número de neuronas por capa.

In [ ]:
```
""" RN simple usando pytorch modules """

import torch
import torch.nn as nn

class SimpleFeedforwardNN(nn.Module):
    def __init__(self, input_size, hidden_sizes, output_size):
        super(SimpleFeedforwardNN, self).__init__()

        self.input_layer = nn.Linear(input_size, hidden_sizes[0])


        self.hidden_layers = nn.ModuleList()
        for i in range(len(hidden_sizes) - 1):
            self.hidden_layers.append(nn.Linear(hidden_sizes[i], hidden_sizes[i
```

```
        self.output_layer = nn.Linear(hidden_sizes[-1], output_size)


        self.activation = nn.ReLU()

    def forward(self, x):
        x = self.activation(self.input_layer(x))
        for layer in self.hidden_layers:
            x = self.activation(layer(x))
        x = self.output_layer(x)
        return x

input_size = X_train.shape[1]
hidden_sizes = [64, 32]
output_size = 3

model = SimpleFeedforwardNN(input_size, hidden_sizes, output_size)
print(model)
```

```
SimpleFeedforwardNN(
  (input_layer): Linear(in_features=4, out_features=64, bias=True)
  (hidden_layers): ModuleList(
    (0): Linear(in_features=64, out_features=32, bias=True)
  )
  (output_layer): Linear(in_features=32, out_features=3, bias=True)
  (activation): ReLU()
)
```

Task 3 - Funciones de Pérdida

Utilice diferentes funciones de pérdida comunes como Cross-Entropy Loss y MSE para clasificación. Entrene el modelo con diferentes funciones de pérdida y registre las pérdidas de entrenamiento y test. Debe utilizar al menos 3 diferentes funciones. Es decir, procure que su código sea capaz de parametrizar el uso de diferentes funciones de pérdida.

In [ ]:
```
""" Entrenamiento, funcion de perdida y registro """

def train_model(model, loss_fn, optimizer, num_epochs=50):
    train_losses = []
    val_losses = []

    for epoch in range(num_epochs):
        model.train()
        train_loss = 0.0

        for inputs, targets in train_loader:
            optimizer.zero_grad()
            outputs = model(inputs)
            loss = loss_fn(outputs, targets)
            loss.backward()
            optimizer.step()
            train_loss += loss.item() * inputs.size(0)

        train_loss /= len(train_loader.dataset)
        train_losses.append(train_loss)

        model.eval()
```

```
        val_loss = 0.0

        with torch.no_grad():
            for inputs, targets in val_loader:
                outputs = model(inputs)
                loss = loss_fn(outputs, targets)
                val_loss += loss.item() * inputs.size(0)

        val_loss /= len(val_loader.dataset)
        val_losses.append(val_loss)

        print(f'Epoch [{epoch+1}/{num_epochs}], Train Loss: {train_loss:.4f}, V

    return train_losses, val_losses

input_size = X_train.shape[1]
hidden_sizes = [64, 32]
output_size = 3
num_epochs = 50
learning_rate = 0.001
model = SimpleFeedforwardNN(input_size, hidden_sizes, output_size)
loss_fn = nn.CrossEntropyLoss()
optimizer = optim.Adam(model.parameters(), lr=learning_rate)
train_model(model, loss_fn, optimizer, num_epochs)
```

```
Epoch [1/50], Train Loss: 1.0395, Val Loss: 0.9917
Epoch [2/50], Train Loss: 0.9710, Val Loss: 0.9332
Epoch [3/50], Train Loss: 0.9235, Val Loss: 0.8727
Epoch [4/50], Train Loss: 0.8575, Val Loss: 0.8195
Epoch [5/50], Train Loss: 0.7947, Val Loss: 0.7435
Epoch [6/50], Train Loss: 0.7295, Val Loss: 0.6796
Epoch [7/50], Train Loss: 0.6703, Val Loss: 0.6206
Epoch [8/50], Train Loss: 0.6099, Val Loss: 0.5821
Epoch [9/50], Train Loss: 0.5646, Val Loss: 0.5304
Epoch [10/50], Train Loss: 0.5230, Val Loss: 0.4922
Epoch [11/50], Train Loss: 0.4917, Val Loss: 0.4618
Epoch [12/50], Train Loss: 0.4601, Val Loss: 0.4438
Epoch [13/50], Train Loss: 0.4363, Val Loss: 0.4179
Epoch [14/50], Train Loss: 0.4140, Val Loss: 0.3969
Epoch [15/50], Train Loss: 0.3980, Val Loss: 0.3705
Epoch [16/50], Train Loss: 0.3694, Val Loss: 0.3536
Epoch [17/50], Train Loss: 0.3549, Val Loss: 0.3471
Epoch [18/50], Train Loss: 0.3329, Val Loss: 0.3197
Epoch [19/50], Train Loss: 0.3110, Val Loss: 0.3051
Epoch [20/50], Train Loss: 0.2942, Val Loss: 0.2853
Epoch [21/50], Train Loss: 0.2784, Val Loss: 0.2690
Epoch [22/50], Train Loss: 0.2617, Val Loss: 0.2586
Epoch [23/50], Train Loss: 0.2545, Val Loss: 0.2426
Epoch [24/50], Train Loss: 0.2372, Val Loss: 0.2369
Epoch [25/50], Train Loss: 0.2262, Val Loss: 0.2225
Epoch [26/50], Train Loss: 0.2198, Val Loss: 0.2092
Epoch [27/50], Train Loss: 0.2144, Val Loss: 0.2242
Epoch [28/50], Train Loss: 0.1895, Val Loss: 0.1923
Epoch [29/50], Train Loss: 0.1949, Val Loss: 0.1866
Epoch [30/50], Train Loss: 0.1758, Val Loss: 0.1853
Epoch [31/50], Train Loss: 0.1700, Val Loss: 0.1697
Epoch [32/50], Train Loss: 0.1650, Val Loss: 0.1670
Epoch [33/50], Train Loss: 0.1568, Val Loss: 0.1619
Epoch [34/50], Train Loss: 0.1544, Val Loss: 0.1560
Epoch [35/50], Train Loss: 0.1477, Val Loss: 0.1497
Epoch [36/50], Train Loss: 0.1420, Val Loss: 0.1496
Epoch [37/50], Train Loss: 0.1387, Val Loss: 0.1438
Epoch [38/50], Train Loss: 0.1361, Val Loss: 0.1506
Epoch [39/50], Train Loss: 0.1309, Val Loss: 0.1310
Epoch [40/50], Train Loss: 0.1296, Val Loss: 0.1406
Epoch [41/50], Train Loss: 0.1239, Val Loss: 0.1252
Epoch [42/50], Train Loss: 0.1262, Val Loss: 0.1246
Epoch [43/50], Train Loss: 0.1330, Val Loss: 0.1300
Epoch [44/50], Train Loss: 0.1193, Val Loss: 0.1144
Epoch [45/50], Train Loss: 0.1120, Val Loss: 0.1411
Epoch [46/50], Train Loss: 0.1160, Val Loss: 0.1176
Epoch [47/50], Train Loss: 0.1162, Val Loss: 0.1083
Epoch [48/50], Train Loss: 0.1147, Val Loss: 0.1400
Epoch [49/50], Train Loss: 0.1105, Val Loss: 0.1025
Epoch [50/50], Train Loss: 0.1167, Val Loss: 0.1394
```

Out[ ]:
```
([1.0394882877667746,
  0.9710470557212829,
  0.9234729647636414,
  0.8574853618939717,
  0.7947110454241435,
  0.7294812838236491,
  0.6702702005704244,
  0.6098776936531067,
  0.5646216789881389,
  0.523004537820816,
  0.49172622760136925,
  0.4601483742396037,
  0.43629200061162315,
  0.4140487551689148,
  0.3980002919832865,
  0.3694432059923808,
  0.35490017135938007,
  0.3329428652922312,
  0.3109930435816447,
  0.29419870575269064,
  0.2784022827943166,
  0.2616824984550476,
  0.254542871316274,
  0.23718418876330058,
  0.22622590859731037,
  0.2198312024275462,
  0.21435607671737672,
  0.18954036732514698,
  0.19489469627539316,
  0.17583513458569844,
  0.1700192133585612,
  0.16498130112886428,
  0.15681618750095366,
  0.1543964147567749,
  0.14772089968125027,
  0.14202449123064678,
  0.13868946333726248,
  0.13606667816638945,
  0.13089158261815706,
  0.12958838492631913,
  0.12393135875463486,
  0.1261725316445033,
  0.13298008839289346,
  0.11934195806582769,
  0.11198802292346954,
  0.11595539500315984,
  0.11615439976255099,
  0.11471267938613891,
  0.11053057114283243,
  0.11669432024161021],
 [0.9916950106620789,
  0.9332144856452942,
  0.8726797223091125,
  0.8195156772931417,
  0.7435151735941569,
  0.6795727650324503,
  0.6205779353777567,
  0.5821071545283,
  0.5303530812263488,
  0.49223305384318033,
```

```
 0.4618313491344452,
 0.44379056493441266,
 0.41789148648579916,
 0.3969422320524851З,
 0.3705293615659078,
 0.35358786781628926,
 0.3470892151196798,
 0.31968409021695454,
 0.3051360269387563,
 0.28533007502555846,
 0.268957755963007б,
 0.25864274899164835,
 0.2426152805487315,
 0.2368791550397873,
 0.22249457935492198,
 0.20924883385499318,
 0.2242194265127182,
 0.19232960045337677,
 0.18662608961264293,
 0.18528315126895906,
 0.16969181795914967,
 0.1669577201207479,
 0.1618706931670507,
 0.15599769254525503,
 0.1497454474369685,
 0.14962165256341298,
 0.14384941558043163,
 0.15058367153008778,
 0.13102472176154453,
 0.14062648713588716,
 0.12518675674994786,
 0.12456221332152685,
 0.13003003497918447,
 0.11444991528987884,
 0.14113677938779196,
 0.11761354257663091,
 0.10825399160385132,
 0.13997740745544435,
 0.10249078820149103,
 0.1393769770860672])
```

Task 4 - Técnicas de Regularización

1. Utilice distintas técnicas de regularización como L1, L2 y dropout. Entrene el modelo con y sin técnicas de regularización y observe el impacto en el overfitting y la generalización. Debe utilizar al menos 3 diferentes técnicas. Es decir, procure que su código sea capaz de parametrizar el uso de diferentes técnicas de regularización.

1. **L1 Regularization**: Esta técnica agrega una penalización a la función de pérdida que es proporcional a la suma absoluta de los pesos. Esto puede llevar a la "eliminación" de algunas características menos importantes.
2. **L2 Regularization**: Similar a L1, pero la penalización es proporcional al cuadrado de la suma de los pesos. Este método es útil para evitar el sobreajuste.

3. **Dropout**: Durante el entrenamiento, aleatoriamente establece una fracción de las entradas a cero en cada iteración. Esto también ayuda a evitar el sobreajuste.

👉 El código incluye opciones para activar o desactivar cada una de estas técnicas de regularización.

📈 Al final, comparamos el rendimiento del modelo con y sin regularización para evaluar el impacto en el sobreajuste y la generalización.

In [ ]:
```
""" Técnicas de regularización """

# Hay que cambiar la clase previamente presentada para agregarle dropout

class SimpleFeedforwardNN(nn.Module):
    def __init__(self, input_size, hidden_sizes, output_size, dropout_prob=0.0)
        super(SimpleFeedforwardNN, self).__init__()

        self.input_layer = nn.Linear(input_size, hidden_sizes[0])
        self.hidden_layers = nn.ModuleList()
        self.dropouts = nn.ModuleList()

        for i in range(len(hidden_sizes) - 1):
            self.hidden_layers.append(nn.Linear(hidden_sizes[i], hidden_sizes[i
            self.dropouts.append(nn.Dropout(p=dropout_prob))

        self.output_layer = nn.Linear(hidden_sizes[-1], output_size)
        self.activation = nn.ReLU()

    def forward(self, x):
        x = self.activation(self.input_layer(x))

        for layer, dropout in zip(self.hidden_layers, self.dropouts):
            x = self.activation(layer(x))
            x = dropout(x)

        x = self.output_layer(x)
        return x
# Para entrenar, con L1:

def train_model(model, loss_fn, optimizer, num_epochs=50, l1_lambda=0.0):
    train_losses = []
    val_losses = []

    for epoch in range(num_epochs):
        model.train()
        train_loss = 0.0

        for inputs, targets in train_loader:
            optimizer.zero_grad()
            outputs = model(inputs)
            loss = loss_fn(outputs, targets)

            # Añadir L1 regularization
            l1_reg = 0.0
            for param in model.parameters():
                l1_reg += torch.norm(param, 1)
            loss += l1_lambda * l1_reg
```

```
            loss.backward()
            optimizer.step()
            train_loss += loss.item() * inputs.size(0)

        train_loss /= len(train_loader.dataset)
        train_losses.append(train_loss)

        model.eval()
        val_loss = 0.0

        with torch.no_grad():
            for inputs, targets in val_loader:
                outputs = model(inputs)
                loss = loss_fn(outputs, targets)
                val_loss += loss.item() * inputs.size(0)

        val_loss /= len(val_loader.dataset)
        val_losses.append(val_loss)

        print(f'Epoch [{epoch+1}/{num_epochs}], Train Loss: {train_loss:.4f}, V

    return train_losses, val_losses
```

In [ ]:
```
# 3 formas del modelo entrenado:

# Con L2
l2_lambda = 0.001
optimizer = optim.Adam(model.parameters(), lr=learning_rate, weight_decay=l2_la
print("Training model with L2 :")
train_model(model, loss_fn, optimizer, num_epochs)

# Con L1
l1_lambda = 0.001
model = SimpleFeedforwardNN(input_size, hidden_sizes, output_size)
optimizer = optim.Adam(model.parameters(), lr=learning_rate)
print("Training model with L1 :")
train_model(model, loss_fn, optimizer, num_epochs, l1_lambda=l1_lambda)

# Con Dropout
dropout_prob = 0.5
model = SimpleFeedforwardNN(input_size, hidden_sizes, output_size, dropout_prob
optimizer = optim.Adam(model.parameters(), lr=learning_rate)
print("Training model with Dropout:")
train_model(model, loss_fn, optimizer, num_epochs)
```

```
Training model with L2 :
Epoch [1/50], Train Loss: 0.1148, Val Loss: 0.1287
Epoch [2/50], Train Loss: 0.1125, Val Loss: 0.1117
Epoch [3/50], Train Loss: 0.0983, Val Loss: 0.1015
Epoch [4/50], Train Loss: 0.1082, Val Loss: 0.0990
Epoch [5/50], Train Loss: 0.0991, Val Loss: 0.1247
Epoch [6/50], Train Loss: 0.0984, Val Loss: 0.0942
Epoch [7/50], Train Loss: 0.0981, Val Loss: 0.1108
Epoch [8/50], Train Loss: 0.1001, Val Loss: 0.0883
Epoch [9/50], Train Loss: 0.0922, Val Loss: 0.1123
Epoch [10/50], Train Loss: 0.0966, Val Loss: 0.0995
Epoch [11/50], Train Loss: 0.0923, Val Loss: 0.0971
Epoch [12/50], Train Loss: 0.0928, Val Loss: 0.0959
Epoch [13/50], Train Loss: 0.0887, Val Loss: 0.0910
Epoch [14/50], Train Loss: 0.0882, Val Loss: 0.0895
Epoch [15/50], Train Loss: 0.0861, Val Loss: 0.0909
Epoch [16/50], Train Loss: 0.0874, Val Loss: 0.0870
Epoch [17/50], Train Loss: 0.0896, Val Loss: 0.1094
Epoch [18/50], Train Loss: 0.0869, Val Loss: 0.0790
Epoch [19/50], Train Loss: 0.0853, Val Loss: 0.0890
Epoch [20/50], Train Loss: 0.0917, Val Loss: 0.1075
Epoch [21/50], Train Loss: 0.0904, Val Loss: 0.0758
Epoch [22/50], Train Loss: 0.0900, Val Loss: 0.1235
Epoch [23/50], Train Loss: 0.0831, Val Loss: 0.0766
Epoch [24/50], Train Loss: 0.0830, Val Loss: 0.0936
Epoch [25/50], Train Loss: 0.0831, Val Loss: 0.0886
Epoch [26/50], Train Loss: 0.0786, Val Loss: 0.0770
Epoch [27/50], Train Loss: 0.0811, Val Loss: 0.0876
Epoch [28/50], Train Loss: 0.0794, Val Loss: 0.0950
Epoch [29/50], Train Loss: 0.0820, Val Loss: 0.0827
Epoch [30/50], Train Loss: 0.0800, Val Loss: 0.0858
Epoch [31/50], Train Loss: 0.0777, Val Loss: 0.0796
Epoch [32/50], Train Loss: 0.0814, Val Loss: 0.0725
Epoch [33/50], Train Loss: 0.0855, Val Loss: 0.1075
Epoch [34/50], Train Loss: 0.1003, Val Loss: 0.0685
Epoch [35/50], Train Loss: 0.0729, Val Loss: 0.1466
Epoch [36/50], Train Loss: 0.0873, Val Loss: 0.0729
Epoch [37/50], Train Loss: 0.0787, Val Loss: 0.0717
Epoch [38/50], Train Loss: 0.0725, Val Loss: 0.1383
Epoch [39/50], Train Loss: 0.0877, Val Loss: 0.0723
Epoch [40/50], Train Loss: 0.0897, Val Loss: 0.0664
Epoch [41/50], Train Loss: 0.0817, Val Loss: 0.1277
Epoch [42/50], Train Loss: 0.0695, Val Loss: 0.0655
Epoch [43/50], Train Loss: 0.0839, Val Loss: 0.0699
Epoch [44/50], Train Loss: 0.0995, Val Loss: 0.0833
Epoch [45/50], Train Loss: 0.0759, Val Loss: 0.0690
Epoch [46/50], Train Loss: 0.0733, Val Loss: 0.0800
Epoch [47/50], Train Loss: 0.0775, Val Loss: 0.0948
Epoch [48/50], Train Loss: 0.0773, Val Loss: 0.0643
Epoch [49/50], Train Loss: 0.0804, Val Loss: 0.0922
Epoch [50/50], Train Loss: 0.0725, Val Loss: 0.0771
Training model with L1 :
Epoch [1/50], Train Loss: 1.2903, Val Loss: 1.0078
Epoch [2/50], Train Loss: 1.1972, Val Loss: 0.9484
Epoch [3/50], Train Loss: 1.1226, Val Loss: 0.8788
Epoch [4/50], Train Loss: 1.0463, Val Loss: 0.7998
Epoch [5/50], Train Loss: 0.9665, Val Loss: 0.7308
Epoch [6/50], Train Loss: 0.8892, Val Loss: 0.6622
Epoch [7/50], Train Loss: 0.8247, Val Loss: 0.6010
Epoch [8/50], Train Loss: 0.7659, Val Loss: 0.5647
```

```
Epoch [9/50], Train Loss: 0.7041, Val Loss: 0.5076
Epoch [10/50], Train Loss: 0.6535, Val Loss: 0.4712
Epoch [11/50], Train Loss: 0.6174, Val Loss: 0.4471
Epoch [12/50], Train Loss: 0.5839, Val Loss: 0.4178
Epoch [13/50], Train Loss: 0.5562, Val Loss: 0.3937
Epoch [14/50], Train Loss: 0.5283, Val Loss: 0.3700
Epoch [15/50], Train Loss: 0.5052, Val Loss: 0.3519
Epoch [16/50], Train Loss: 0.4836, Val Loss: 0.3329
Epoch [17/50], Train Loss: 0.4657, Val Loss: 0.3169
Epoch [18/50], Train Loss: 0.4480, Val Loss: 0.2941
Epoch [19/50], Train Loss: 0.4268, Val Loss: 0.2858
Epoch [20/50], Train Loss: 0.4211, Val Loss: 0.2631
Epoch [21/50], Train Loss: 0.3887, Val Loss: 0.2656
Epoch [22/50], Train Loss: 0.3851, Val Loss: 0.2407
Epoch [23/50], Train Loss: 0.3684, Val Loss: 0.2263
Epoch [24/50], Train Loss: 0.3601, Val Loss: 0.2193
Epoch [25/50], Train Loss: 0.3528, Val Loss: 0.2038
Epoch [26/50], Train Loss: 0.3339, Val Loss: 0.1997
Epoch [27/50], Train Loss: 0.3239, Val Loss: 0.1862
Epoch [28/50], Train Loss: 0.3145, Val Loss: 0.1777
Epoch [29/50], Train Loss: 0.3092, Val Loss: 0.1772
Epoch [30/50], Train Loss: 0.3025, Val Loss: 0.1633
Epoch [31/50], Train Loss: 0.2935, Val Loss: 0.1681
Epoch [32/50], Train Loss: 0.2889, Val Loss: 0.1518
Epoch [33/50], Train Loss: 0.2857, Val Loss: 0.1479
Epoch [34/50], Train Loss: 0.2791, Val Loss: 0.1486
Epoch [35/50], Train Loss: 0.2762, Val Loss: 0.1374
Epoch [36/50], Train Loss: 0.2768, Val Loss: 0.1555
Epoch [37/50], Train Loss: 0.2665, Val Loss: 0.1302
Epoch [38/50], Train Loss: 0.2639, Val Loss: 0.1351
Epoch [39/50], Train Loss: 0.2584, Val Loss: 0.1281
Epoch [40/50], Train Loss: 0.2615, Val Loss: 0.1311
Epoch [41/50], Train Loss: 0.2637, Val Loss: 0.1207
Epoch [42/50], Train Loss: 0.2563, Val Loss: 0.1154
Epoch [43/50], Train Loss: 0.2462, Val Loss: 0.1302
Epoch [44/50], Train Loss: 0.2550, Val Loss: 0.1143
Epoch [45/50], Train Loss: 0.2515, Val Loss: 0.1100
Epoch [46/50], Train Loss: 0.2436, Val Loss: 0.1280
Epoch [47/50], Train Loss: 0.2415, Val Loss: 0.1051
Epoch [48/50], Train Loss: 0.2458, Val Loss: 0.1193
Epoch [49/50], Train Loss: 0.2404, Val Loss: 0.1028
Epoch [50/50], Train Loss: 0.2369, Val Loss: 0.1118
Training model with Dropout:
Epoch [1/50], Train Loss: 1.0996, Val Loss: 1.0066
Epoch [2/50], Train Loss: 1.0029, Val Loss: 0.9376
Epoch [3/50], Train Loss: 0.9169, Val Loss: 0.8706
Epoch [4/50], Train Loss: 0.8687, Val Loss: 0.8071
Epoch [5/50], Train Loss: 0.7949, Val Loss: 0.7506
Epoch [6/50], Train Loss: 0.7918, Val Loss: 0.6864
Epoch [7/50], Train Loss: 0.7165, Val Loss: 0.6293
Epoch [8/50], Train Loss: 0.6401, Val Loss: 0.5815
Epoch [9/50], Train Loss: 0.6159, Val Loss: 0.5405
Epoch [10/50], Train Loss: 0.6008, Val Loss: 0.5120
Epoch [11/50], Train Loss: 0.5903, Val Loss: 0.4872
Epoch [12/50], Train Loss: 0.5597, Val Loss: 0.4608
Epoch [13/50], Train Loss: 0.5177, Val Loss: 0.4366
Epoch [14/50], Train Loss: 0.4689, Val Loss: 0.4176
Epoch [15/50], Train Loss: 0.4501, Val Loss: 0.3964
Epoch [16/50], Train Loss: 0.4827, Val Loss: 0.3831
Epoch [17/50], Train Loss: 0.4235, Val Loss: 0.3708
```

```
Epoch [18/50], Train Loss: 0.4337, Val Loss: 0.3602
Epoch [19/50], Train Loss: 0.3947, Val Loss: 0.3495
Epoch [20/50], Train Loss: 0.4079, Val Loss: 0.3369
Epoch [21/50], Train Loss: 0.3972, Val Loss: 0.3235
Epoch [22/50], Train Loss: 0.3770, Val Loss: 0.3159
Epoch [23/50], Train Loss: 0.3970, Val Loss: 0.3081
Epoch [24/50], Train Loss: 0.3526, Val Loss: 0.2952
Epoch [25/50], Train Loss: 0.3601, Val Loss: 0.2850
Epoch [26/50], Train Loss: 0.3561, Val Loss: 0.2763
Epoch [27/50], Train Loss: 0.3442, Val Loss: 0.2652
Epoch [28/50], Train Loss: 0.2992, Val Loss: 0.2576
Epoch [29/50], Train Loss: 0.3019, Val Loss: 0.2485
Epoch [30/50], Train Loss: 0.2651, Val Loss: 0.2421
Epoch [31/50], Train Loss: 0.3052, Val Loss: 0.2501
Epoch [32/50], Train Loss: 0.2751, Val Loss: 0.2296
Epoch [33/50], Train Loss: 0.3007, Val Loss: 0.2231
Epoch [34/50], Train Loss: 0.3123, Val Loss: 0.2283
Epoch [35/50], Train Loss: 0.2746, Val Loss: 0.2110
Epoch [36/50], Train Loss: 0.2655, Val Loss: 0.2068
Epoch [37/50], Train Loss: 0.2621, Val Loss: 0.1981
Epoch [38/50], Train Loss: 0.2457, Val Loss: 0.1924
Epoch [39/50], Train Loss: 0.2576, Val Loss: 0.1919
Epoch [40/50], Train Loss: 0.2358, Val Loss: 0.1831
Epoch [41/50], Train Loss: 0.2669, Val Loss: 0.1971
Epoch [42/50], Train Loss: 0.2250, Val Loss: 0.1752
Epoch [43/50], Train Loss: 0.2326, Val Loss: 0.1736
Epoch [44/50], Train Loss: 0.2471, Val Loss: 0.1731
Epoch [45/50], Train Loss: 0.2321, Val Loss: 0.1652
Epoch [46/50], Train Loss: 0.2033, Val Loss: 0.1619
Epoch [47/50], Train Loss: 0.2605, Val Loss: 0.1621
Epoch [48/50], Train Loss: 0.1989, Val Loss: 0.1783
Epoch [49/50], Train Loss: 0.1965, Val Loss: 0.1799
Epoch [50/50], Train Loss: 0.2252, Val Loss: 0.1528
```

```
Out[ ]: ([1.0996270736058553,
  1.0028921763102214,
  0.9168765385945638,
  0.8687164783477783,
  0.7948939879735311,
  0.7918228228886922,
  0.7165433724721273,
  0.6400507092475891,
  0.6159457087516784,
  0.6008168299992879,
  0.590293037891388,
  0.5596873799959818,
  0.5176592727502187,
  0.4688975016276042,
  0.4501486341158549,
  0.4826695680618286,
  0.4234974443912506,
  0.43365675608317056,
  0.394671098391215,
  0.4079393565654755,
  0.3971593499183655,
  0.376977535088857,
  0.39698839088280996,
  0.35256712436676024,
  0.36011882225672404,
  0.3560899237791697,
  0.3441721280415853,
  0.29920858244101206,
  0.30186676780382793,
  0.26511187354723614,
  0.30520646323760353,
  0.27506387134393057,
  0.30066809356212615,
  0.31230096419652303,
  0.2745921035607656,
  0.26545064747333524,
  0.2621136337518692,
  0.2456688513358434,
  0.2575616677602132,
  0.23584210773309072,
  0.2669198383887609,
  0.2249877373377482,
  0.2326148768266042,
  0.24709629019101462,
  0.23206540048122407,
  0.20327861309051515,
  0.26050362388292947,
  0.1988532880942027,
  0.1964561899503072,
  0.22523534297943115],
 [1.0065665006637574,
  0.9375657598177592,
  0.8705913106600444,
  0.8070969144503276,
  0.7506411234537761,
  0.6864304582277934,
  0.6292675296465556,
  0.5814549922943115,
  0.5405452052752177,
  0.5120458205540975,
```

```
 0.48721574544906615,
 0.46075816750526427,
 0.4365877628326416,
 0.41756362915039064,
 0.3963821570078532,
 0.383107093969980 9,
 0.37076109846433003,
 0.3602098822593689,
 0.349505219856898,
 0.3368961612383525,
 0.323485050598780 3,
 0.3158600827058156,
 0.30812618335088093,
 0.29524775544802345,
 0.28497990369796755,
 0.276272904872894 3,
 0.265226020415624,
 0.2576145400603612,
 0.24853898684183756,
 0.2421168178319931,
 0.25005666017532346,
 0.22960163752237955,
 0.22311863998572032,
 0.22829956710338592,
 0.21095897952715556,
 0.20680545171101888,
 0.19810270369052888,
 0.19241432348887125,
 0.19191015064716338,
 0.18305814762910208,
 0.19709277749061585,
 0.1752497692902883,
 0.1736062705516815,
 0.17314809362093608,
 0.16517485678195953,
 0.16187709867954253,
 0.16208880245685578,
 0.1782767007748286,
 0.17989531954129537,
 0.1528171400229136])
```

Task 5 - Funciones de Pérdida

Utilice distintas técnicas de optimización como SGD, Batch GD, Mini-Batch GD. Entrene el modelo con algoritmos de optimización y registre las pérdidas y tiempos de entrenamiento y test. Debe utilizar al menos 3 diferentes algoritmos. Es decir, procure que su código sea capaz de parametrizar el uso de diferentes algoritmos de optimización.

📚 Conceptos Teóricos 📚

- 🛠 SGD: Actualiza los parámetros del modelo usando solo un ejemplo de entrenamiento por iteración.
- 🛠 Batch GD: Utiliza todo el conjunto de datos para actualizar los parámetros del modelo en cada iteración.

- 🛠 Mini-Batch GD: Compromiso entre SGD y Batch GD; utiliza un pequeño lote de ejemplos para actualizar los parámetros.

🛠 Implementación 🛠

- 📈 Entrenaremos el modelo con cada uno de los algoritmos y registraremos las pérdidas durante el entrenamiento y la validación.
- ⏰ Utilizaremos la librería `time` de Python para registrar el tiempo que tarda cada algoritmo en entrenar y probar el modelo.
- 🔄 Haremos que el código sea parametrizable para facilitar el uso de diferentes algoritmos de optimización.

In [ ]:
```
import time

def train_model_with_time(model, loss_fn, optimizer, num_epochs=50, l1_lambda=(
    train_losses = []
    val_losses = []
    start_time = time.time()

    for epoch in range(num_epochs):
        model.train()
        train_loss = 0.0

        for inputs, targets in train_loader:
            optimizer.zero_grad()
            outputs = model(inputs)
            loss = loss_fn(outputs, targets)

            # L1 regularization
            l1_reg = 0.0
            for param in model.parameters():
                l1_reg += torch.norm(param, 1)
            loss += l1_lambda * l1_reg

            loss.backward()
            optimizer.step()
            train_loss += loss.item() * inputs.size(0)

        train_loss /= len(train_loader.dataset)
        train_losses.append(train_loss)

        model.eval()
        val_loss = 0.0
        with torch.no_grad():
            for inputs, targets in val_loader:
                outputs = model(inputs)
                loss = loss_fn(outputs, targets)
                val_loss += loss.item() * inputs.size(0)

        val_loss /= len(val_loader.dataset)
        val_losses.append(val_loss)

        print(f'Epoch [{epoch+1}/{num_epochs}], Train Loss: {train_loss:.4f}, \

    end_time = time.time()
    elapsed_time = end_time - start_time
```

```
    print(f'Total time elapsed: {elapsed_time:.2f} seconds')
    return train_losses, val_losses, elapsed_time

# SGD
optimizer = optim.SGD(model.parameters(), lr=learning_rate)
print("Training model with SGD:")
train_model_with_time(model, loss_fn, optimizer, num_epochs)

#  Batch GD
optimizer = optim.SGD(model.parameters(), lr=learning_rate)
print("Training model with Batch GD:")
train_loader_batch = DataLoader(train_dataset, batch_size=len(train_dataset), s
train_model_with_time(model, loss_fn, optimizer, num_epochs)

#  Mini-Batch GD
optimizer = optim.SGD(model.parameters(), lr=learning_rate)
print("Training model with Mini-Batch GD:")
train_loader_mini_batch = DataLoader(train_dataset, batch_size=32, shuffle=True
train_model_with_time(model, loss_fn, optimizer, num_epochs)
```

```
Training model with SGD:
Epoch [1/50], Train Loss: 0.2118, Val Loss: 0.1570
Epoch [2/50], Train Loss: 0.2007, Val Loss: 0.1545
Epoch [3/50], Train Loss: 0.1825, Val Loss: 0.1566
Epoch [4/50], Train Loss: 0.1996, Val Loss: 0.1564
Epoch [5/50], Train Loss: 0.1936, Val Loss: 0.1525
Epoch [6/50], Train Loss: 0.1917, Val Loss: 0.1540
Epoch [7/50], Train Loss: 0.1972, Val Loss: 0.1566
Epoch [8/50], Train Loss: 0.2384, Val Loss: 0.1550
Epoch [9/50], Train Loss: 0.2101, Val Loss: 0.1527
Epoch [10/50], Train Loss: 0.1984, Val Loss: 0.1551
Epoch [11/50], Train Loss: 0.1980, Val Loss: 0.1543
Epoch [12/50], Train Loss: 0.2119, Val Loss: 0.1515
Epoch [13/50], Train Loss: 0.1841, Val Loss: 0.1523
Epoch [14/50], Train Loss: 0.1826, Val Loss: 0.1480
Epoch [15/50], Train Loss: 0.1686, Val Loss: 0.1483
Epoch [16/50], Train Loss: 0.1928, Val Loss: 0.1514
Epoch [17/50], Train Loss: 0.2118, Val Loss: 0.1490
Epoch [18/50], Train Loss: 0.1837, Val Loss: 0.1475
Epoch [19/50], Train Loss: 0.1964, Val Loss: 0.1494
Epoch [20/50], Train Loss: 0.2195, Val Loss: 0.1499
Epoch [21/50], Train Loss: 0.2118, Val Loss: 0.1591
Epoch [22/50], Train Loss: 0.1922, Val Loss: 0.1625
Epoch [23/50], Train Loss: 0.1842, Val Loss: 0.1661
Epoch [24/50], Train Loss: 0.1976, Val Loss: 0.1583
Epoch [25/50], Train Loss: 0.1683, Val Loss: 0.1540
Epoch [26/50], Train Loss: 0.2095, Val Loss: 0.1551
Epoch [27/50], Train Loss: 0.1799, Val Loss: 0.1532
Epoch [28/50], Train Loss: 0.2081, Val Loss: 0.1494
Epoch [29/50], Train Loss: 0.1942, Val Loss: 0.1497
Epoch [30/50], Train Loss: 0.2024, Val Loss: 0.1478
Epoch [31/50], Train Loss: 0.2060, Val Loss: 0.1483
Epoch [32/50], Train Loss: 0.1871, Val Loss: 0.1513
Epoch [33/50], Train Loss: 0.1927, Val Loss: 0.1552
Epoch [34/50], Train Loss: 0.1800, Val Loss: 0.1511
Epoch [35/50], Train Loss: 0.1595, Val Loss: 0.1508
Epoch [36/50], Train Loss: 0.1792, Val Loss: 0.1471
Epoch [37/50], Train Loss: 0.2015, Val Loss: 0.1482
Epoch [38/50], Train Loss: 0.1988, Val Loss: 0.1472
Epoch [39/50], Train Loss: 0.1660, Val Loss: 0.1469
Epoch [40/50], Train Loss: 0.2054, Val Loss: 0.1460
Epoch [41/50], Train Loss: 0.1602, Val Loss: 0.1488
Epoch [42/50], Train Loss: 0.1898, Val Loss: 0.1462
Epoch [43/50], Train Loss: 0.1999, Val Loss: 0.1462
Epoch [44/50], Train Loss: 0.2085, Val Loss: 0.1435
Epoch [45/50], Train Loss: 0.2480, Val Loss: 0.1443
Epoch [46/50], Train Loss: 0.1937, Val Loss: 0.1460
Epoch [47/50], Train Loss: 0.2055, Val Loss: 0.1493
Epoch [48/50], Train Loss: 0.1718, Val Loss: 0.1463
Epoch [49/50], Train Loss: 0.1685, Val Loss: 0.1454
Epoch [50/50], Train Loss: 0.1726, Val Loss: 0.1486
Total time elapsed: 1.66 seconds
Training model with Batch GD:
Epoch [1/50], Train Loss: 0.2039, Val Loss: 0.1457
Epoch [2/50], Train Loss: 0.1884, Val Loss: 0.1461
Epoch [3/50], Train Loss: 0.2003, Val Loss: 0.1516
Epoch [4/50], Train Loss: 0.1857, Val Loss: 0.1524
Epoch [5/50], Train Loss: 0.1993, Val Loss: 0.1519
Epoch [6/50], Train Loss: 0.1982, Val Loss: 0.1460
Epoch [7/50], Train Loss: 0.2173, Val Loss: 0.1461
```

```
Epoch [8/50], Train Loss: 0.2106, Val Loss: 0.1463
Epoch [9/50], Train Loss: 0.1997, Val Loss: 0.1467
Epoch [10/50], Train Loss: 0.1829, Val Loss: 0.1477
Epoch [11/50], Train Loss: 0.1935, Val Loss: 0.1473
Epoch [12/50], Train Loss: 0.1879, Val Loss: 0.1485
Epoch [13/50], Train Loss: 0.1967, Val Loss: 0.1550
Epoch [14/50], Train Loss: 0.2042, Val Loss: 0.1454
Epoch [15/50], Train Loss: 0.1491, Val Loss: 0.1430
Epoch [16/50], Train Loss: 0.1489, Val Loss: 0.1436
Epoch [17/50], Train Loss: 0.1610, Val Loss: 0.1432
Epoch [18/50], Train Loss: 0.2060, Val Loss: 0.1446
Epoch [19/50], Train Loss: 0.1672, Val Loss: 0.1454
Epoch [20/50], Train Loss: 0.1925, Val Loss: 0.1485
Epoch [21/50], Train Loss: 0.1754, Val Loss: 0.1445
Epoch [22/50], Train Loss: 0.1919, Val Loss: 0.1449
Epoch [23/50], Train Loss: 0.2090, Val Loss: 0.1416
Epoch [24/50], Train Loss: 0.1671, Val Loss: 0.1402
Epoch [25/50], Train Loss: 0.1708, Val Loss: 0.1415
Epoch [26/50], Train Loss: 0.1913, Val Loss: 0.1398
Epoch [27/50], Train Loss: 0.1847, Val Loss: 0.1409
Epoch [28/50], Train Loss: 0.1883, Val Loss: 0.1404
Epoch [29/50], Train Loss: 0.1671, Val Loss: 0.1426
Epoch [30/50], Train Loss: 0.1920, Val Loss: 0.1438
Epoch [31/50], Train Loss: 0.1633, Val Loss: 0.1431
Epoch [32/50], Train Loss: 0.1627, Val Loss: 0.1437
Epoch [33/50], Train Loss: 0.2292, Val Loss: 0.1473
Epoch [34/50], Train Loss: 0.1367, Val Loss: 0.1424
Epoch [35/50], Train Loss: 0.1862, Val Loss: 0.1449
Epoch [36/50], Train Loss: 0.1671, Val Loss: 0.1458
Epoch [37/50], Train Loss: 0.2026, Val Loss: 0.1458
Epoch [38/50], Train Loss: 0.1766, Val Loss: 0.1439
Epoch [39/50], Train Loss: 0.1749, Val Loss: 0.1416
Epoch [40/50], Train Loss: 0.2119, Val Loss: 0.1457
Epoch [41/50], Train Loss: 0.1698, Val Loss: 0.1428
Epoch [42/50], Train Loss: 0.1826, Val Loss: 0.1426
Epoch [43/50], Train Loss: 0.1781, Val Loss: 0.1468
Epoch [44/50], Train Loss: 0.1586, Val Loss: 0.1444
Epoch [45/50], Train Loss: 0.2055, Val Loss: 0.1432
Epoch [46/50], Train Loss: 0.1470, Val Loss: 0.1433
Epoch [47/50], Train Loss: 0.1608, Val Loss: 0.1411
Epoch [48/50], Train Loss: 0.2095, Val Loss: 0.1399
Epoch [49/50], Train Loss: 0.1678, Val Loss: 0.1416
Epoch [50/50], Train Loss: 0.2150, Val Loss: 0.1474
Total time elapsed: 1.49 seconds
Training model with Mini-Batch GD:
Epoch [1/50], Train Loss: 0.1980, Val Loss: 0.1542
Epoch [2/50], Train Loss: 0.1954, Val Loss: 0.1541
Epoch [3/50], Train Loss: 0.1758, Val Loss: 0.1513
Epoch [4/50], Train Loss: 0.1725, Val Loss: 0.1434
Epoch [5/50], Train Loss: 0.1846, Val Loss: 0.1415
Epoch [6/50], Train Loss: 0.1705, Val Loss: 0.1421
Epoch [7/50], Train Loss: 0.1812, Val Loss: 0.1403
Epoch [8/50], Train Loss: 0.2017, Val Loss: 0.1386
Epoch [9/50], Train Loss: 0.1827, Val Loss: 0.1418
Epoch [10/50], Train Loss: 0.1621, Val Loss: 0.1408
Epoch [11/50], Train Loss: 0.1750, Val Loss: 0.1385
Epoch [12/50], Train Loss: 0.1808, Val Loss: 0.1402
Epoch [13/50], Train Loss: 0.1772, Val Loss: 0.1402
Epoch [14/50], Train Loss: 0.1475, Val Loss: 0.1351
Epoch [15/50], Train Loss: 0.1901, Val Loss: 0.1357
```

```
Epoch [16/50], Train Loss: 0.1708, Val Loss: 0.1401
Epoch [17/50], Train Loss: 0.2195, Val Loss: 0.1467
Epoch [18/50], Train Loss: 0.2014, Val Loss: 0.1420
Epoch [19/50], Train Loss: 0.1826, Val Loss: 0.1411
Epoch [20/50], Train Loss: 0.1934, Val Loss: 0.1422
Epoch [21/50], Train Loss: 0.1790, Val Loss: 0.1453
Epoch [22/50], Train Loss: 0.1451, Val Loss: 0.1405
Epoch [23/50], Train Loss: 0.2006, Val Loss: 0.1391
Epoch [24/50], Train Loss: 0.1649, Val Loss: 0.1390
Epoch [25/50], Train Loss: 0.1742, Val Loss: 0.1371
Epoch [26/50], Train Loss: 0.1794, Val Loss: 0.1352
Epoch [27/50], Train Loss: 0.1952, Val Loss: 0.1362
Epoch [28/50], Train Loss: 0.1872, Val Loss: 0.1373
Epoch [29/50], Train Loss: 0.1596, Val Loss: 0.1396
Epoch [30/50], Train Loss: 0.2008, Val Loss: 0.1425
Epoch [31/50], Train Loss: 0.1502, Val Loss: 0.1430
Epoch [32/50], Train Loss: 0.2392, Val Loss: 0.1451
Epoch [33/50], Train Loss: 0.2090, Val Loss: 0.1429
Epoch [34/50], Train Loss: 0.1914, Val Loss: 0.1418
Epoch [35/50], Train Loss: 0.1572, Val Loss: 0.1450
Epoch [36/50], Train Loss: 0.1723, Val Loss: 0.1428
Epoch [37/50], Train Loss: 0.1757, Val Loss: 0.1394
Epoch [38/50], Train Loss: 0.1557, Val Loss: 0.1463
Epoch [39/50], Train Loss: 0.1762, Val Loss: 0.1410
Epoch [40/50], Train Loss: 0.1868, Val Loss: 0.1404
Epoch [41/50], Train Loss: 0.2080, Val Loss: 0.1471
Epoch [42/50], Train Loss: 0.1793, Val Loss: 0.1473
Epoch [43/50], Train Loss: 0.1752, Val Loss: 0.1386
Epoch [44/50], Train Loss: 0.1868, Val Loss: 0.1371
Epoch [45/50], Train Loss: 0.1843, Val Loss: 0.1384
Epoch [46/50], Train Loss: 0.1869, Val Loss: 0.1409
Epoch [47/50], Train Loss: 0.1985, Val Loss: 0.1406
Epoch [48/50], Train Loss: 0.1550, Val Loss: 0.1357
Epoch [49/50], Train Loss: 0.1603, Val Loss: 0.1373
Epoch [50/50], Train Loss: 0.1963, Val Loss: 0.1402
Total time elapsed: 1.47 seconds
```

Out[ ]:
```
([0.19800997873147327,
  0.1953788737456004,
  0.17583791663249335,
  0.17245453844467798,
  0.1846157878637314,
  0.17049778600533802,
  0.1811884582042694,
  0.20172767341136932,
  0.182653005917867,
  0.16211466888586681,
  0.17500985860824586,
  0.1807971845070521,
  0.1771840676665306,
  0.14749583353598913,
  0.19006423875689507,
  0.1708306759595871,
  0.2195251335700353,
  0.20144105553627015,
  0.18262779613335928,
  0.1933859239021937,
  0.17895231147607168,
  0.1451323335369428,
  0.20062355647484462,
  0.16490304470062256,
  0.17416472136974334,
  0.17944012582302094,
  0.19519848128159842,
  0.1872301936149597,
  0.15963781972726185,
  0.20076688130696616,
  0.15021142562230427,
  0.23915868401527404,
  0.20898393938938778,
  0.19143283267815908,
  0.15717576344807943,
  0.17231491903464,
  0.17567848960558574,
  0.15566565891106923,
  0.17620945125818252,
  0.18681936065355936,
  0.20797371963659922,
  0.17931439379851025,
  0.1752135987083117,
  0.186833559970061,
  0.18426561901966731,
  0.18689241111278534,
  0.1984652449687322,
  0.15496667027473449,
  0.16034377813339235,
  0.19631632020076115],
 [0.15420153439044954,
  0.15412139197190602,
  0.15133368770281475,
  0.14341285427411396,
  0.14150294164816538,
  0.14209166467189788,
  0.14025473097960153,
  0.13858133256435395,
  0.14181561569372814,
  0.14081079562505086,
```

```
  0.1384967863559723,
  0.14020910561084748,
  0.1401899149020513,
  0.13507031053304672,
  0.13570320506890615,
  0.14010433653990428,
  0.14665087163448334,
  0.14195604324340821,
  0.1410952498515447,
  0.14222003122170765,
  0.14526916146278382,
  0.14050169984499614,
  0.13913657069206237,
  0.138993834455808,
  0.1371303806702296,
  0.13523284097512564,
  0.13617866535981496,
  0.1373499095439911,
  0.1396434078613917,
  0.14250484009583791,
  0.14302346408367156,
  0.14506795903046926,
  0.14290308058261872,
  0.1417818586031596,
  0.1450237492720286,
  0.14282688796520232,
  0.13937822779019673,
  0.146268031001091,
  0.14104704360167186,
  0.1403547078371048,
  0.14714010258515675,
  0.1472694953282674,
  0.13857207993666332,
  0.1370827982823054,
  0.1384015421072642,
  0.14088341693083445,
  0.14061051805814107,
  0.1356519306699435,
  0.13725719352563223,
  0.1402451495329539],
 1.4694523811340332)
```

Task 6 - Experimentación y Análisis

Entrene los modelos con diferentes combinaciones de funciones de pérdida, técnicas de regularización y algoritmos de optimización. Para no complicar esta parte, puede dejar fijo dos de estos parámetros (función de pérdida, técnicas de regularización, algoritmo de optimización) y solamente cambiar uno de ellos. Deben verse al menos 9 combinaciones en total, donde es válido que en una de ellas no haya ninguna técnica de regularización. Si quiere experimentar con más combinaciones se le dará hasta 10% de puntos extra. Para cada combinación registre métricas como precisión, pérdida y alguna otra métrica que considere pertinente (Recuerde lo visto en inteligencia artificial). Visualice las curvas (tanto en precisión, pérdida y la tercera métrica que decidió) de entrenamiento y validación utilizando bibliotecas como matplotlib y/o seaborn. Además, recuerde llevar tracking de los tiempos de ejecución de cada combinación

🤔 Descripción:

- En este task, vamos a realizar una serie de experimentos utilizando diferentes combinaciones de funciones de pérdida, técnicas de regularización y algoritmos de optimización.
- Mantendremos dos de estos factores constantes mientras variamos el tercero.

📈 Metas:

1. Realizar al menos 12 combinaciones de experimentos.
2. Registrar métricas como precisión, pérdida y otra métrica de su elección.
3. Visualizar las curvas de precisión, pérdida y la tercera métrica usando Seaborn.
4. Mantener un registro del tiempo de ejecución para cada combinación de experimentos.

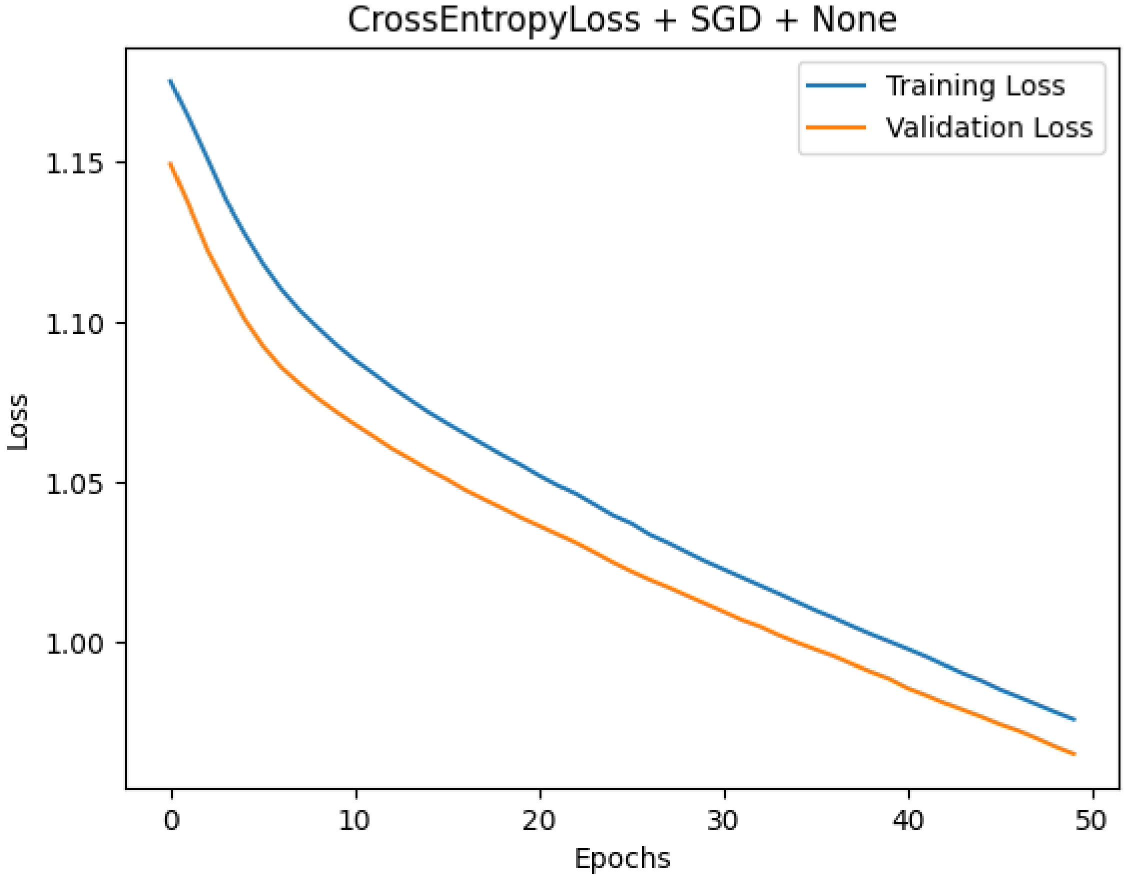
🛠 Implementación:

1. Utilizar un modelo de red neuronal base.
2. Aplicar diferentes técnicas de regularización (L1, L2, Dropout).
3. Utilizar diferentes funciones de pérdida (CrossEntropy, NLLLoss).
4. Emplear diferentes algoritmos de optimización (SGD, Adam).

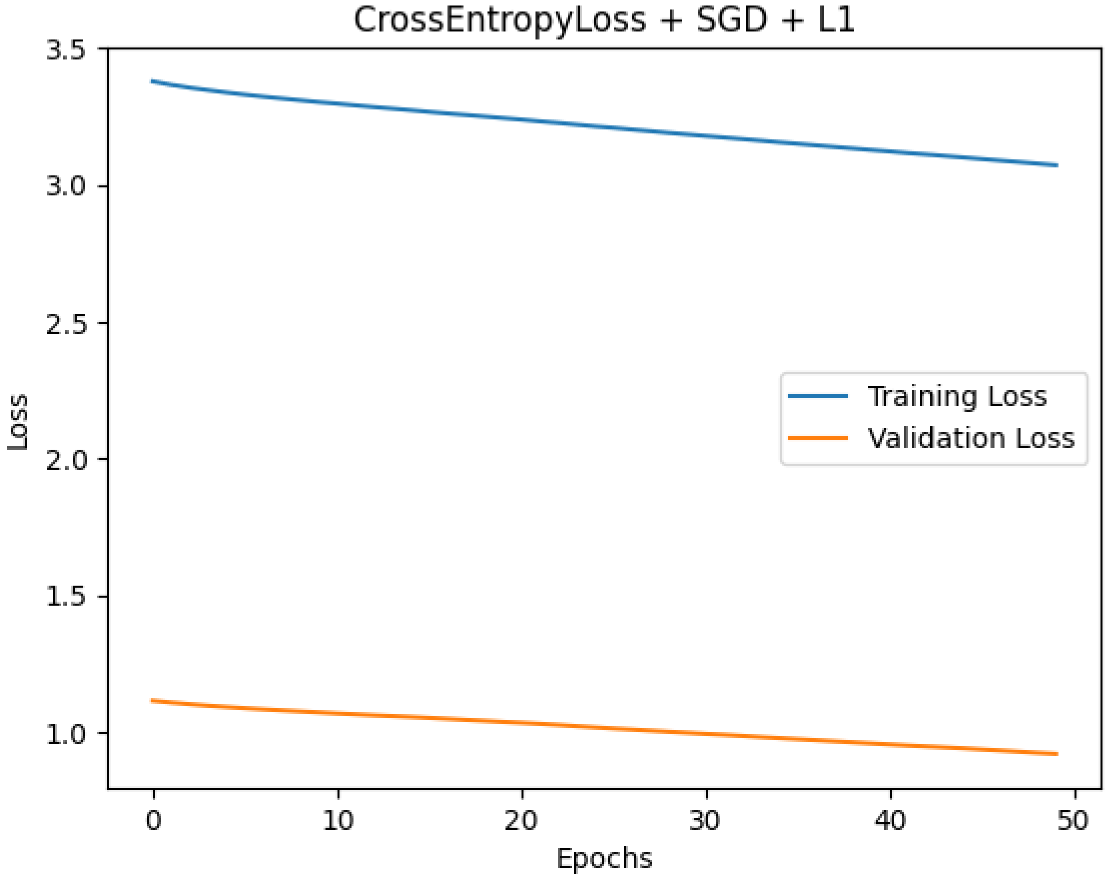
In [ ]:
```
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd

def evaluate_model(model, loader):
    model.eval()
    correct = 0
    total = 0
    with torch.no_grad():
        for inputs, targets in loader:
            outputs = model(inputs)
            _, predicted = torch.max(outputs.data, 1)
            total += targets.size(0)
            correct += (predicted == targets).sum().item()
```

```
    return 100 * correct / total

results = []
for loss_function in [nn.CrossEntropyLoss(), nn.NLLLoss()]:
    for optimizer_type in ['SGD', 'Adam']:
        for regularization in ['None', 'L1', 'L2', 'Dropout']:
            model = SimpleFeedforwardNN(input_size, hidden_sizes, output_size)
            if regularization == 'Dropout':
                model.dropout = nn.Dropout(0.5)


            loss_fn = loss_function


            if optimizer_type == 'SGD':
                optimizer = optim.SGD(model.parameters(), lr=learning_rate, wei
            elif optimizer_type == 'Adam':
                optimizer = optim.Adam(model.parameters(), lr=learning_rate, we

            l1_lambda = 0.01 if regularization == 'L1' else 0.0
            train_losses, val_losses, elapsed_time = train_model_with_time(mode

            accuracy = evaluate_model(model, val_loader)

            metrics = {
                'Loss Function': loss_fn.__class__.__name__,
                'Optimizer': optimizer_type,
                'Regularization': regularization,
                'Final Training Loss': train_losses[-1],
                'Final Validation Loss': val_losses[-1],
                'Validation Accuracy': accuracy,
                'Time': elapsed_time
            }
            results.append(metrics)

            sns.lineplot(x=range(num_epochs), y=train_losses, label='Training I
            sns.lineplot(x=range(num_epochs), y=val_losses, label='Validation I
            plt.title(f"{loss_fn.__class__.__name__} + {optimizer_type} + {regu
            plt.xlabel("Epochs")
            plt.ylabel("Loss")
            plt.legend()
            plt.show()

results_df = pd.DataFrame(results)
print(results_df)
```

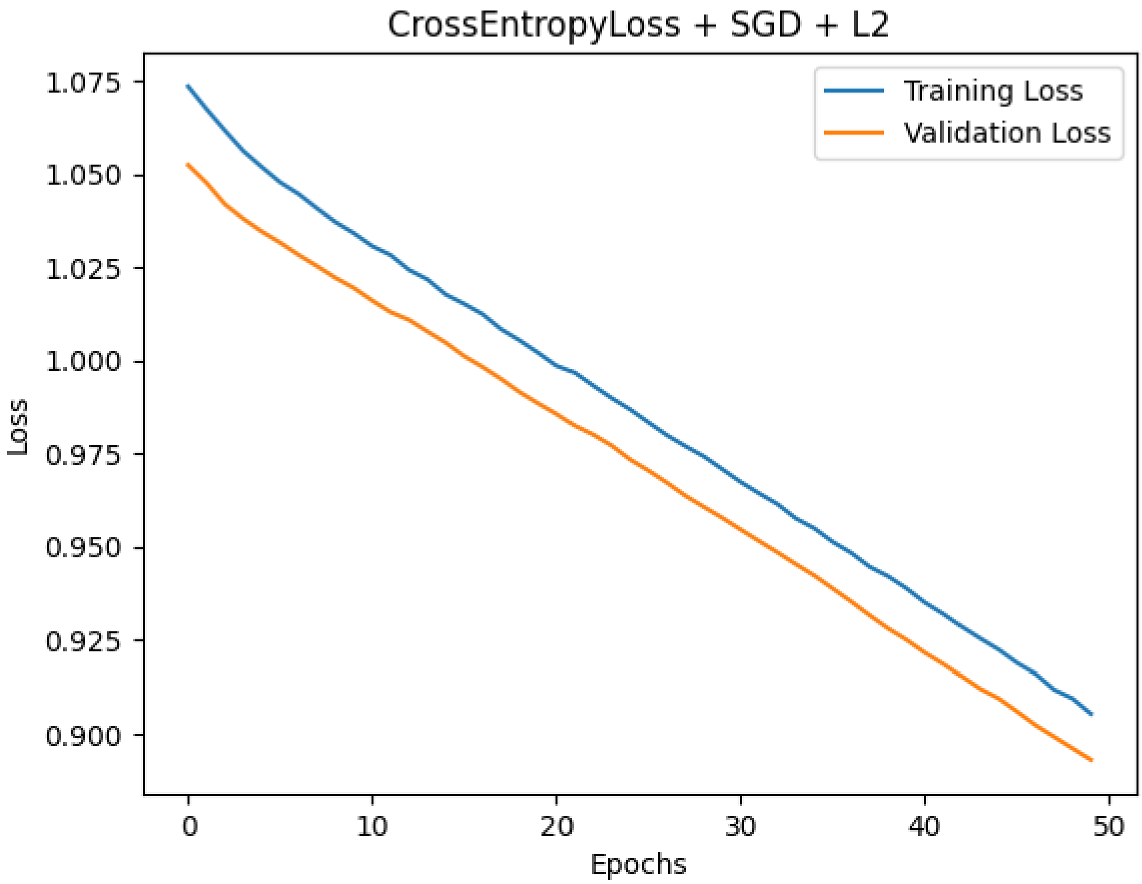
```
Epoch [1/50], Train Loss: 1.1754, Val Loss: 1.1494
Epoch [2/50], Train Loss: 1.1636, Val Loss: 1.1364
Epoch [3/50], Train Loss: 1.1510, Val Loss: 1.1225
Epoch [4/50], Train Loss: 1.1382, Val Loss: 1.1116
Epoch [5/50], Train Loss: 1.1279, Val Loss: 1.1011
Epoch [6/50], Train Loss: 1.1185, Val Loss: 1.0927
Epoch [7/50], Train Loss: 1.1104, Val Loss: 1.0860
Epoch [8/50], Train Loss: 1.1038, Val Loss: 1.0808
Epoch [9/50], Train Loss: 1.0983, Val Loss: 1.0761
Epoch [10/50], Train Loss: 1.0930, Val Loss: 1.0720
Epoch [11/50], Train Loss: 1.0883, Val Loss: 1.0681
Epoch [12/50], Train Loss: 1.0841, Val Loss: 1.0644
Epoch [13/50], Train Loss: 1.0797, Val Loss: 1.0605
Epoch [14/50], Train Loss: 1.0758, Val Loss: 1.0572
Epoch [15/50], Train Loss: 1.0719, Val Loss: 1.0539
Epoch [16/50], Train Loss: 1.0685, Val Loss: 1.0509
Epoch [17/50], Train Loss: 1.0651, Val Loss: 1.0475
Epoch [18/50], Train Loss: 1.0618, Val Loss: 1.0446
Epoch [19/50], Train Loss: 1.0585, Val Loss: 1.0418
Epoch [20/50], Train Loss: 1.0555, Val Loss: 1.0389
Epoch [21/50], Train Loss: 1.0521, Val Loss: 1.0363
Epoch [22/50], Train Loss: 1.0490, Val Loss: 1.0336
Epoch [23/50], Train Loss: 1.0463, Val Loss: 1.0309
Epoch [24/50], Train Loss: 1.0430, Val Loss: 1.0279
Epoch [25/50], Train Loss: 1.0396, Val Loss: 1.0248
Epoch [26/50], Train Loss: 1.0370, Val Loss: 1.0220
Epoch [27/50], Train Loss: 1.0335, Val Loss: 1.0193
Epoch [28/50], Train Loss: 1.0309, Val Loss: 1.0170
Epoch [29/50], Train Loss: 1.0281, Val Loss: 1.0144
Epoch [30/50], Train Loss: 1.0252, Val Loss: 1.0119
Epoch [31/50], Train Loss: 1.0227, Val Loss: 1.0094
Epoch [32/50], Train Loss: 1.0201, Val Loss: 1.0068
Epoch [33/50], Train Loss: 1.0176, Val Loss: 1.0047
Epoch [34/50], Train Loss: 1.0150, Val Loss: 1.0020
Epoch [35/50], Train Loss: 1.0124, Val Loss: 0.9997
Epoch [36/50], Train Loss: 1.0098, Val Loss: 0.9975
Epoch [37/50], Train Loss: 1.0074, Val Loss: 0.9955
Epoch [38/50], Train Loss: 1.0048, Val Loss: 0.9930
Epoch [39/50], Train Loss: 1.0024, Val Loss: 0.9906
Epoch [40/50], Train Loss: 1.0001, Val Loss: 0.9884
Epoch [41/50], Train Loss: 0.9977, Val Loss: 0.9855
Epoch [42/50], Train Loss: 0.9953, Val Loss: 0.9833
Epoch [43/50], Train Loss: 0.9927, Val Loss: 0.9809
Epoch [44/50], Train Loss: 0.9900, Val Loss: 0.9787
Epoch [45/50], Train Loss: 0.9878, Val Loss: 0.9765
Epoch [46/50], Train Loss: 0.9851, Val Loss: 0.9742
Epoch [47/50], Train Loss: 0.9828, Val Loss: 0.9722
Epoch [48/50], Train Loss: 0.9804, Val Loss: 0.9698
Epoch [49/50], Train Loss: 0.9780, Val Loss: 0.9672
Epoch [50/50], Train Loss: 0.9758, Val Loss: 0.9649
Total time elapsed: 1.24 seconds
```


CrossEntropyLoss + SGD + None
Training Loss
Validation Loss
Loss
Epochs
1.15
1.10
1.05
1.00
0
10
20
30
40
50

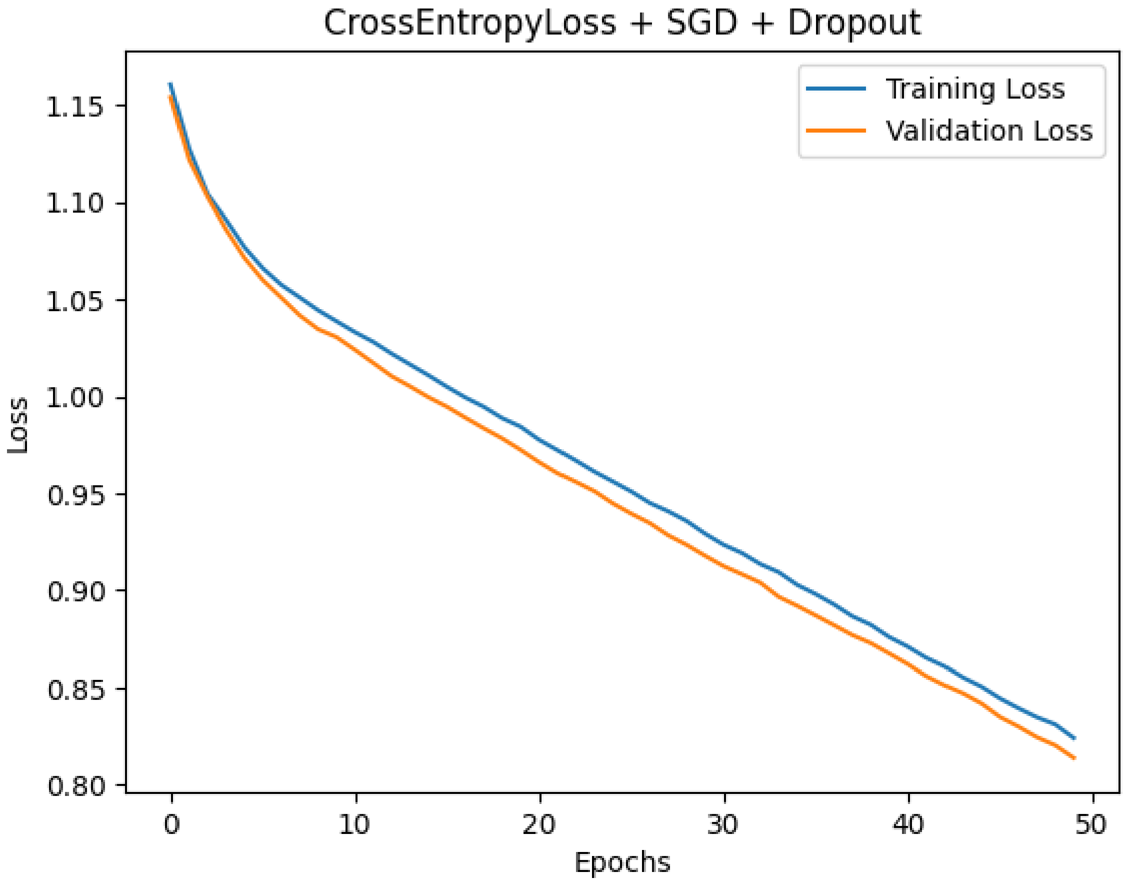
```
Epoch [1/50], Train Loss: 3.3792, Val Loss: 1.1138
Epoch [2/50], Train Loss: 3.3665, Val Loss: 1.1070
Epoch [3/50], Train Loss: 3.3558, Val Loss: 1.1009
Epoch [4/50], Train Loss: 3.3468, Val Loss: 1.0955
Epoch [5/50], Train Loss: 3.3382, Val Loss: 1.0907
Epoch [6/50], Train Loss: 3.3303, Val Loss: 1.0863
Epoch [7/50], Train Loss: 3.3230, Val Loss: 1.0821
Epoch [8/50], Train Loss: 3.3163, Val Loss: 1.0784
Epoch [9/50], Train Loss: 3.3097, Val Loss: 1.0746
Epoch [10/50], Train Loss: 3.3032, Val Loss: 1.0710
Epoch [11/50], Train Loss: 3.2975, Val Loss: 1.0672
Epoch [12/50], Train Loss: 3.2911, Val Loss: 1.0640
Epoch [13/50], Train Loss: 3.2850, Val Loss: 1.0605
Epoch [14/50], Train Loss: 3.2794, Val Loss: 1.0576
Epoch [15/50], Train Loss: 3.2735, Val Loss: 1.0545
Epoch [16/50], Train Loss: 3.2673, Val Loss: 1.0510
Epoch [17/50], Train Loss: 3.2614, Val Loss: 1.0475
Epoch [18/50], Train Loss: 3.2560, Val Loss: 1.0436
Epoch [19/50], Train Loss: 3.2503, Val Loss: 1.0401
Epoch [20/50], Train Loss: 3.2447, Val Loss: 1.0362
Epoch [21/50], Train Loss: 3.2390, Val Loss: 1.0330
Epoch [22/50], Train Loss: 3.2327, Val Loss: 1.0293
Epoch [23/50], Train Loss: 3.2269, Val Loss: 1.0251
Epoch [24/50], Train Loss: 3.2205, Val Loss: 1.0205
Epoch [25/50], Train Loss: 3.2141, Val Loss: 1.0162
Epoch [26/50], Train Loss: 3.2089, Val Loss: 1.0120
Epoch [27/50], Train Loss: 3.2025, Val Loss: 1.0082
Epoch [28/50], Train Loss: 3.1966, Val Loss: 1.0042
Epoch [29/50], Train Loss: 3.1904, Val Loss: 1.0001
Epoch [30/50], Train Loss: 3.1848, Val Loss: 0.9964
Epoch [31/50], Train Loss: 3.1790, Val Loss: 0.9924
Epoch [32/50], Train Loss: 3.1734, Val Loss: 0.9890
Epoch [33/50], Train Loss: 3.1678, Val Loss: 0.9850
Epoch [34/50], Train Loss: 3.1620, Val Loss: 0.9807
Epoch [35/50], Train Loss: 3.1560, Val Loss: 0.9769
Epoch [36/50], Train Loss: 3.1505, Val Loss: 0.9732
Epoch [37/50], Train Loss: 3.1446, Val Loss: 0.9693
Epoch [38/50], Train Loss: 3.1390, Val Loss: 0.9652
Epoch [39/50], Train Loss: 3.1334, Val Loss: 0.9614
Epoch [40/50], Train Loss: 3.1277, Val Loss: 0.9575
Epoch [41/50], Train Loss: 3.1224, Val Loss: 0.9537
Epoch [42/50], Train Loss: 3.1164, Val Loss: 0.9499
Epoch [43/50], Train Loss: 3.1114, Val Loss: 0.9464
Epoch [44/50], Train Loss: 3.1056, Val Loss: 0.9429
Epoch [45/50], Train Loss: 3.1001, Val Loss: 0.9396
Epoch [46/50], Train Loss: 3.0944, Val Loss: 0.9354
Epoch [47/50], Train Loss: 3.0889, Val Loss: 0.9315
Epoch [48/50], Train Loss: 3.0834, Val Loss: 0.9272
Epoch [49/50], Train Loss: 3.0777, Val Loss: 0.9230
Epoch [50/50], Train Loss: 3.0719, Val Loss: 0.9194
Total time elapsed: 2.18 seconds
```


CrossEntropyLoss + SGD + L1
Loss
Epochs
Training Loss
Validation Loss
3.5
3.0
2.5
2.0
1.5
1.0
0
10
20
30
40
50

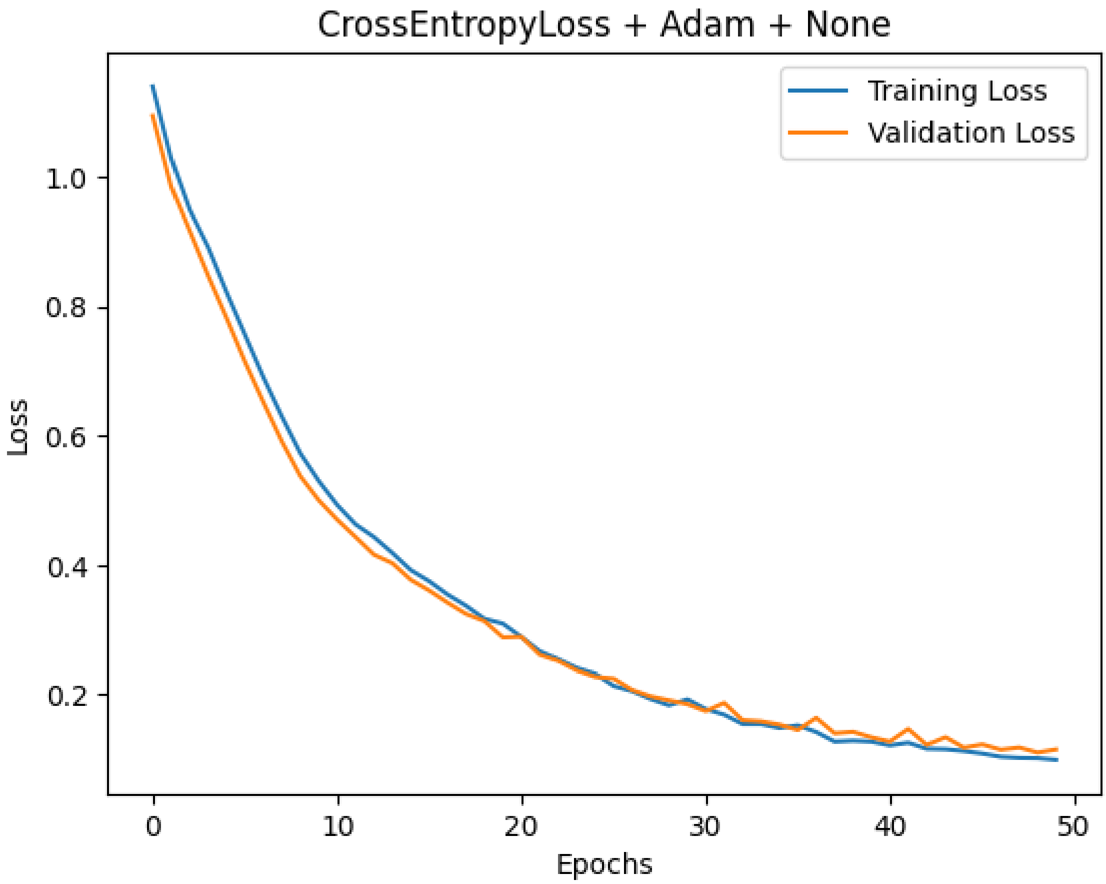
```
Epoch [1/50], Train Loss: 1.0735, Val Loss: 1.0524
Epoch [2/50], Train Loss: 1.0674, Val Loss: 1.0476
Epoch [3/50], Train Loss: 1.0617, Val Loss: 1.0419
Epoch [4/50], Train Loss: 1.0562, Val Loss: 1.0380
Epoch [5/50], Train Loss: 1.0519, Val Loss: 1.0346
Epoch [6/50], Train Loss: 1.0478, Val Loss: 1.0316
Epoch [7/50], Train Loss: 1.0447, Val Loss: 1.0284
Epoch [8/50], Train Loss: 1.0408, Val Loss: 1.0254
Epoch [9/50], Train Loss: 1.0372, Val Loss: 1.0222
Epoch [10/50], Train Loss: 1.0342, Val Loss: 1.0195
Epoch [11/50], Train Loss: 1.0307, Val Loss: 1.0161
Epoch [12/50], Train Loss: 1.0283, Val Loss: 1.0130
Epoch [13/50], Train Loss: 1.0243, Val Loss: 1.0109
Epoch [14/50], Train Loss: 1.0218, Val Loss: 1.0078
Epoch [15/50], Train Loss: 1.0177, Val Loss: 1.0048
Epoch [16/50], Train Loss: 1.0152, Val Loss: 1.0012
Epoch [17/50], Train Loss: 1.0124, Val Loss: 0.9983
Epoch [18/50], Train Loss: 1.0084, Val Loss: 0.9950
Epoch [19/50], Train Loss: 1.0054, Val Loss: 0.9915
Epoch [20/50], Train Loss: 1.0021, Val Loss: 0.9885
Epoch [21/50], Train Loss: 0.9986, Val Loss: 0.9857
Epoch [22/50], Train Loss: 0.9967, Val Loss: 0.9825
Epoch [23/50], Train Loss: 0.9932, Val Loss: 0.9800
Epoch [24/50], Train Loss: 0.9898, Val Loss: 0.9771
Epoch [25/50], Train Loss: 0.9868, Val Loss: 0.9734
Epoch [26/50], Train Loss: 0.9833, Val Loss: 0.9704
Epoch [27/50], Train Loss: 0.9799, Val Loss: 0.9671
Epoch [28/50], Train Loss: 0.9770, Val Loss: 0.9636
Epoch [29/50], Train Loss: 0.9742, Val Loss: 0.9607
Epoch [30/50], Train Loss: 0.9708, Val Loss: 0.9577
Epoch [31/50], Train Loss: 0.9673, Val Loss: 0.9546
Epoch [32/50], Train Loss: 0.9643, Val Loss: 0.9515
Epoch [33/50], Train Loss: 0.9614, Val Loss: 0.9485
Epoch [34/50], Train Loss: 0.9576, Val Loss: 0.9453
Epoch [35/50], Train Loss: 0.9549, Val Loss: 0.9423
Epoch [36/50], Train Loss: 0.9513, Val Loss: 0.9388
Epoch [37/50], Train Loss: 0.9484, Val Loss: 0.9353
Epoch [38/50], Train Loss: 0.9446, Val Loss: 0.9316
Epoch [39/50], Train Loss: 0.9421, Val Loss: 0.9281
Epoch [40/50], Train Loss: 0.9388, Val Loss: 0.9251
Epoch [41/50], Train Loss: 0.9350, Val Loss: 0.9216
Epoch [42/50], Train Loss: 0.9320, Val Loss: 0.9186
Epoch [43/50], Train Loss: 0.9285, Val Loss: 0.9153
Epoch [44/50], Train Loss: 0.9254, Val Loss: 0.9119
Epoch [45/50], Train Loss: 0.9224, Val Loss: 0.9094
Epoch [46/50], Train Loss: 0.9188, Val Loss: 0.9059
Epoch [47/50], Train Loss: 0.9159, Val Loss: 0.9022
Epoch [48/50], Train Loss: 0.9117, Val Loss: 0.8991
Epoch [49/50], Train Loss: 0.9094, Val Loss: 0.8960
Epoch [50/50], Train Loss: 0.9053, Val Loss: 0.8930
Total time elapsed: 1.81 seconds
```


CrossEntropyLoss + SGD + L2
Training Loss
Validation Loss
Loss
Epochs
1.075
1.050
1.025
1.000
0.975
0.950
0.925
0.900
0
10
20
30
40
50

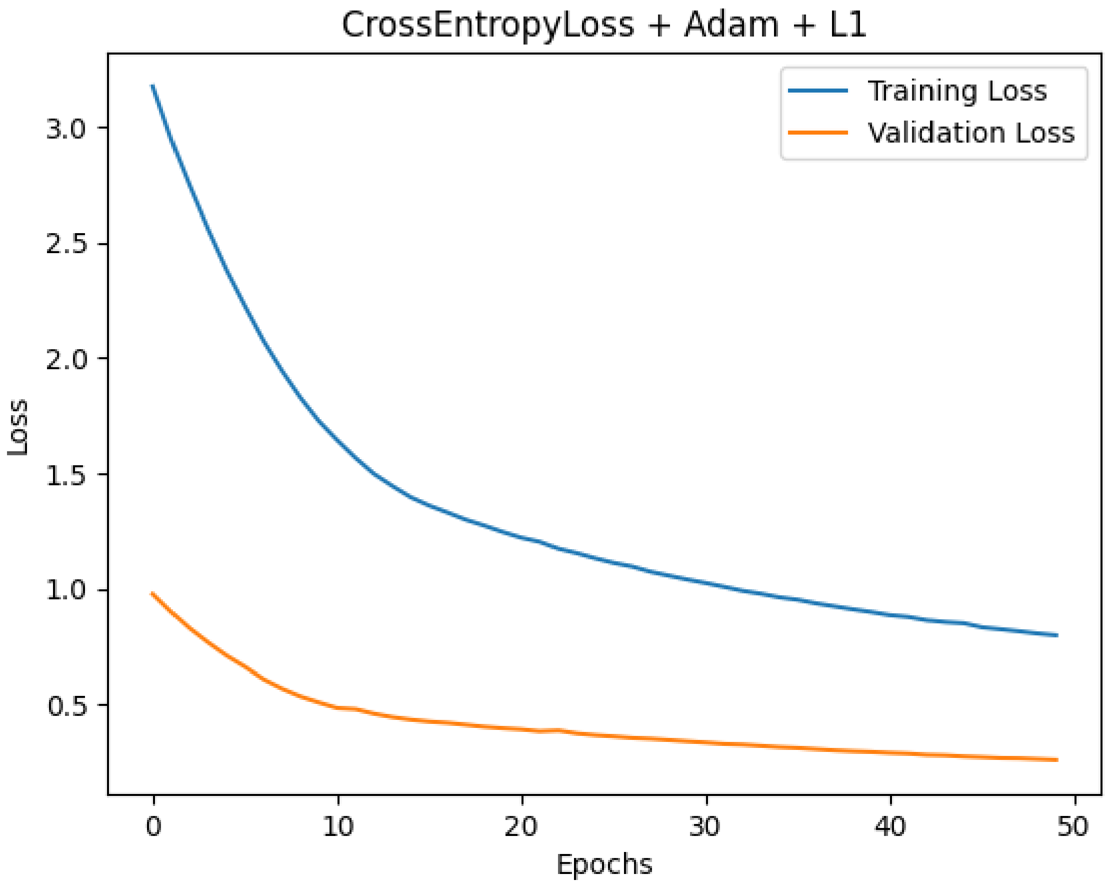
```
Epoch [1/50], Train Loss: 1.1607, Val Loss: 1.1540
Epoch [2/50], Train Loss: 1.1275, Val Loss: 1.1217
Epoch [3/50], Train Loss: 1.1041, Val Loss: 1.1026
Epoch [4/50], Train Loss: 1.0910, Val Loss: 1.0859
Epoch [5/50], Train Loss: 1.0769, Val Loss: 1.0713
Epoch [6/50], Train Loss: 1.0660, Val Loss: 1.0599
Epoch [7/50], Train Loss: 1.0575, Val Loss: 1.0509
Epoch [8/50], Train Loss: 1.0510, Val Loss: 1.0418
Epoch [9/50], Train Loss: 1.0444, Val Loss: 1.0346
Epoch [10/50], Train Loss: 1.0387, Val Loss: 1.0305
Epoch [11/50], Train Loss: 1.0331, Val Loss: 1.0240
Epoch [12/50], Train Loss: 1.0281, Val Loss: 1.0172
Epoch [13/50], Train Loss: 1.0220, Val Loss: 1.0104
Epoch [14/50], Train Loss: 1.0164, Val Loss: 1.0051
Epoch [15/50], Train Loss: 1.0108, Val Loss: 0.9995
Epoch [16/50], Train Loss: 1.0049, Val Loss: 0.9946
Epoch [17/50], Train Loss: 0.9994, Val Loss: 0.9890
Epoch [18/50], Train Loss: 0.9946, Val Loss: 0.9835
Epoch [19/50], Train Loss: 0.9887, Val Loss: 0.9783
Epoch [20/50], Train Loss: 0.9843, Val Loss: 0.9724
Epoch [21/50], Train Loss: 0.9776, Val Loss: 0.9661
Epoch [22/50], Train Loss: 0.9722, Val Loss: 0.9603
Epoch [23/50], Train Loss: 0.9667, Val Loss: 0.9557
Epoch [24/50], Train Loss: 0.9610, Val Loss: 0.9510
Epoch [25/50], Train Loss: 0.9560, Val Loss: 0.9447
Epoch [26/50], Train Loss: 0.9508, Val Loss: 0.9394
Epoch [27/50], Train Loss: 0.9449, Val Loss: 0.9346
Epoch [28/50], Train Loss: 0.9406, Val Loss: 0.9283
Epoch [29/50], Train Loss: 0.9356, Val Loss: 0.9233
Epoch [30/50], Train Loss: 0.9291, Val Loss: 0.9177
Epoch [31/50], Train Loss: 0.9234, Val Loss: 0.9124
Epoch [32/50], Train Loss: 0.9191, Val Loss: 0.9082
Epoch [33/50], Train Loss: 0.9134, Val Loss: 0.9039
Epoch [34/50], Train Loss: 0.9093, Val Loss: 0.8966
Epoch [35/50], Train Loss: 0.9028, Val Loss: 0.8919
Epoch [36/50], Train Loss: 0.8980, Val Loss: 0.8870
Epoch [37/50], Train Loss: 0.8927, Val Loss: 0.8820
Epoch [38/50], Train Loss: 0.8865, Val Loss: 0.8769
Epoch [39/50], Train Loss: 0.8822, Val Loss: 0.8727
Epoch [40/50], Train Loss: 0.8758, Val Loss: 0.8674
Epoch [41/50], Train Loss: 0.8710, Val Loss: 0.8620
Epoch [42/50], Train Loss: 0.8652, Val Loss: 0.8558
Epoch [43/50], Train Loss: 0.8608, Val Loss: 0.8510
Epoch [44/50], Train Loss: 0.8550, Val Loss: 0.8469
Epoch [45/50], Train Loss: 0.8503, Val Loss: 0.8417
Epoch [46/50], Train Loss: 0.8443, Val Loss: 0.8348
Epoch [47/50], Train Loss: 0.8393, Val Loss: 0.8299
Epoch [48/50], Train Loss: 0.8347, Val Loss: 0.8244
Epoch [49/50], Train Loss: 0.8310, Val Loss: 0.8202
Epoch [50/50], Train Loss: 0.8240, Val Loss: 0.8137
Total time elapsed: 0.66 seconds
```


CrossEntropyLoss + SGD + Dropout
Training Loss
Validation Loss
Loss
Epochs
1.15
1.10
1.05
1.00
0.95
0.90
0.85
0.80
0
10
20
30
40
50

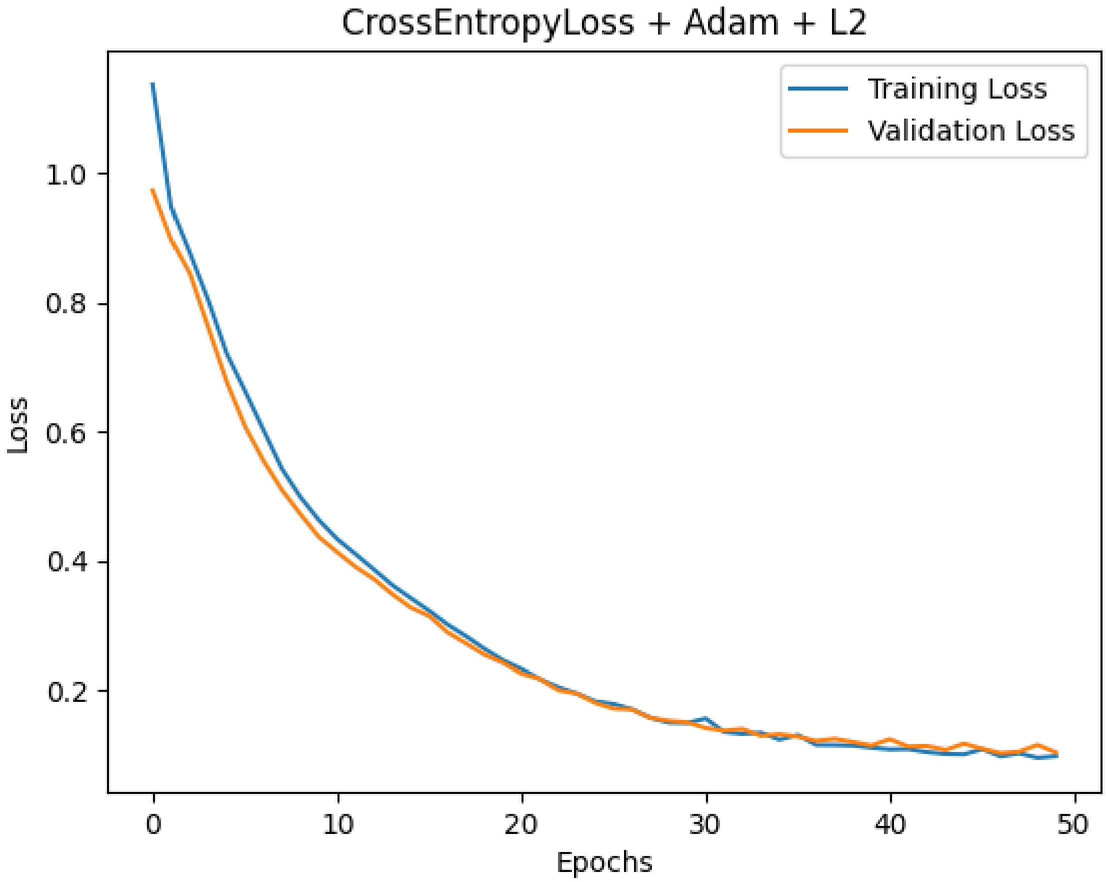
```
Epoch [1/50], Train Loss: 1.1396, Val Loss: 1.0941
Epoch [2/50], Train Loss: 1.0281, Val Loss: 0.9846
Epoch [3/50], Train Loss: 0.9483, Val Loss: 0.9157
Epoch [4/50], Train Loss: 0.8907, Val Loss: 0.8466
Epoch [5/50], Train Loss: 0.8211, Val Loss: 0.7812
Epoch [6/50], Train Loss: 0.7553, Val Loss: 0.7135
Epoch [7/50], Train Loss: 0.6896, Val Loss: 0.6510
Epoch [8/50], Train Loss: 0.6292, Val Loss: 0.5906
Epoch [9/50], Train Loss: 0.5725, Val Loss: 0.5373
Epoch [10/50], Train Loss: 0.5298, Val Loss: 0.5000
Epoch [11/50], Train Loss: 0.4929, Val Loss: 0.4701
Epoch [12/50], Train Loss: 0.4627, Val Loss: 0.4431
Epoch [13/50], Train Loss: 0.4431, Val Loss: 0.4158
Epoch [14/50], Train Loss: 0.4185, Val Loss: 0.4027
Epoch [15/50], Train Loss: 0.3920, Val Loss: 0.3770
Epoch [16/50], Train Loss: 0.3748, Val Loss: 0.3604
Epoch [17/50], Train Loss: 0.3542, Val Loss: 0.3416
Epoch [18/50], Train Loss: 0.3369, Val Loss: 0.3242
Epoch [19/50], Train Loss: 0.3167, Val Loss: 0.3132
Epoch [20/50], Train Loss: 0.3093, Val Loss: 0.2880
Epoch [21/50], Train Loss: 0.2888, Val Loss: 0.2890
Epoch [22/50], Train Loss: 0.2667, Val Loss: 0.2613
Epoch [23/50], Train Loss: 0.2541, Val Loss: 0.2517
Epoch [24/50], Train Loss: 0.2409, Val Loss: 0.2369
Epoch [25/50], Train Loss: 0.2318, Val Loss: 0.2272
Epoch [26/50], Train Loss: 0.2132, Val Loss: 0.2245
Epoch [27/50], Train Loss: 0.2054, Val Loss: 0.2071
Epoch [28/50], Train Loss: 0.1933, Val Loss: 0.1972
Epoch [29/50], Train Loss: 0.1836, Val Loss: 0.1912
Epoch [30/50], Train Loss: 0.1927, Val Loss: 0.1854
Epoch [31/50], Train Loss: 0.1777, Val Loss: 0.1746
Epoch [32/50], Train Loss: 0.1691, Val Loss: 0.1873
Epoch [33/50], Train Loss: 0.1546, Val Loss: 0.1606
Epoch [34/50], Train Loss: 0.1546, Val Loss: 0.1588
Epoch [35/50], Train Loss: 0.1485, Val Loss: 0.1537
Epoch [36/50], Train Loss: 0.1526, Val Loss: 0.1454
Epoch [37/50], Train Loss: 0.1422, Val Loss: 0.1642
Epoch [38/50], Train Loss: 0.1274, Val Loss: 0.1403
Epoch [39/50], Train Loss: 0.1288, Val Loss: 0.1426
Epoch [40/50], Train Loss: 0.1274, Val Loss: 0.1341
Epoch [41/50], Train Loss: 0.1216, Val Loss: 0.1275
Epoch [42/50], Train Loss: 0.1257, Val Loss: 0.1469
Epoch [43/50], Train Loss: 0.1163, Val Loss: 0.1224
Epoch [44/50], Train Loss: 0.1156, Val Loss: 0.1343
Epoch [45/50], Train Loss: 0.1128, Val Loss: 0.1181
Epoch [46/50], Train Loss: 0.1090, Val Loss: 0.1231
Epoch [47/50], Train Loss: 0.1039, Val Loss: 0.1149
Epoch [48/50], Train Loss: 0.1023, Val Loss: 0.1184
Epoch [49/50], Train Loss: 0.1018, Val Loss: 0.1105
Epoch [50/50], Train Loss: 0.0993, Val Loss: 0.1153
Total time elapsed: 0.83 seconds
```


CrossEntropyLoss + Adam + None
Training Loss
Validation Loss
Loss
Epochs
0
10
20
30
40
50
0.2
0.4
0.6
0.8
1.0

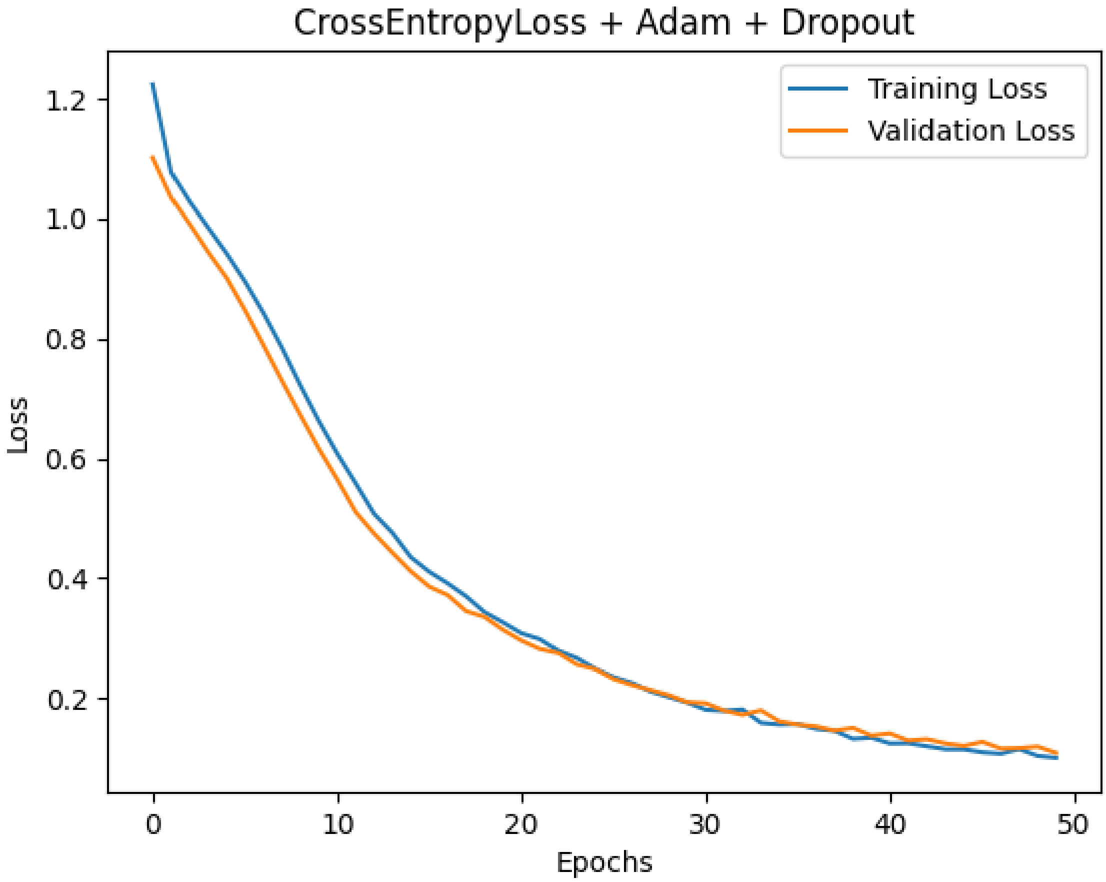
```
Epoch [1/50], Train Loss: 3.1747, Val Loss: 0.9772
Epoch [2/50], Train Loss: 2.9443, Val Loss: 0.8994
Epoch [3/50], Train Loss: 2.7443, Val Loss: 0.8288
Epoch [4/50], Train Loss: 2.5547, Val Loss: 0.7675
Epoch [5/50], Train Loss: 2.3789, Val Loss: 0.7109
Epoch [6/50], Train Loss: 2.2230, Val Loss: 0.6627
Epoch [7/50], Train Loss: 2.0738, Val Loss: 0.6091
Epoch [8/50], Train Loss: 1.9446, Val Loss: 0.5689
Epoch [9/50], Train Loss: 1.8284, Val Loss: 0.5354
Epoch [10/50], Train Loss: 1.7269, Val Loss: 0.5095
Epoch [11/50], Train Loss: 1.6440, Val Loss: 0.4856
Epoch [12/50], Train Loss: 1.5669, Val Loss: 0.4808
Epoch [13/50], Train Loss: 1.4978, Val Loss: 0.4611
Epoch [14/50], Train Loss: 1.4452, Val Loss: 0.4462
Epoch [15/50], Train Loss: 1.3964, Val Loss: 0.4352
Epoch [16/50], Train Loss: 1.3604, Val Loss: 0.4268
Epoch [17/50], Train Loss: 1.3302, Val Loss: 0.4213
Epoch [18/50], Train Loss: 1.2989, Val Loss: 0.4130
Epoch [19/50], Train Loss: 1.2736, Val Loss: 0.4044
Epoch [20/50], Train Loss: 1.2460, Val Loss: 0.3982
Epoch [21/50], Train Loss: 1.2213, Val Loss: 0.3932
Epoch [22/50], Train Loss: 1.2035, Val Loss: 0.3849
Epoch [23/50], Train Loss: 1.1736, Val Loss: 0.3883
Epoch [24/50], Train Loss: 1.1542, Val Loss: 0.3757
Epoch [25/50], Train Loss: 1.1319, Val Loss: 0.3681
Epoch [26/50], Train Loss: 1.1122, Val Loss: 0.3626
Epoch [27/50], Train Loss: 1.0964, Val Loss: 0.3562
Epoch [28/50], Train Loss: 1.0737, Val Loss: 0.3525
Epoch [29/50], Train Loss: 1.0567, Val Loss: 0.3471
Epoch [30/50], Train Loss: 1.0401, Val Loss: 0.3412
Epoch [31/50], Train Loss: 1.0249, Val Loss: 0.3362
Epoch [32/50], Train Loss: 1.0084, Val Loss: 0.3300
Epoch [33/50], Train Loss: 0.9911, Val Loss: 0.3272
Epoch [34/50], Train Loss: 0.9784, Val Loss: 0.3219
Epoch [35/50], Train Loss: 0.9630, Val Loss: 0.3163
Epoch [36/50], Train Loss: 0.9524, Val Loss: 0.3128
Epoch [37/50], Train Loss: 0.9367, Val Loss: 0.3075
Epoch [38/50], Train Loss: 0.9238, Val Loss: 0.3023
Epoch [39/50], Train Loss: 0.9109, Val Loss: 0.2983
Epoch [40/50], Train Loss: 0.8992, Val Loss: 0.2956
Epoch [41/50], Train Loss: 0.8858, Val Loss: 0.2913
Epoch [42/50], Train Loss: 0.8782, Val Loss: 0.2884
Epoch [43/50], Train Loss: 0.8639, Val Loss: 0.2826
Epoch [44/50], Train Loss: 0.8559, Val Loss: 0.2805
Epoch [45/50], Train Loss: 0.8509, Val Loss: 0.2758
Epoch [46/50], Train Loss: 0.8330, Val Loss: 0.2727
Epoch [47/50], Train Loss: 0.8248, Val Loss: 0.2691
Epoch [48/50], Train Loss: 0.8160, Val Loss: 0.2678
Epoch [49/50], Train Loss: 0.8065, Val Loss: 0.2645
Epoch [50/50], Train Loss: 0.7988, Val Loss: 0.2611
Total time elapsed: 0.82 seconds
```


CrossEntropyLoss + Adam + L1
Training Loss
Validation Loss
Loss
Epochs
3.0
2.5
2.0
1.5
1.0
0.5
0
10
20
30
40
50

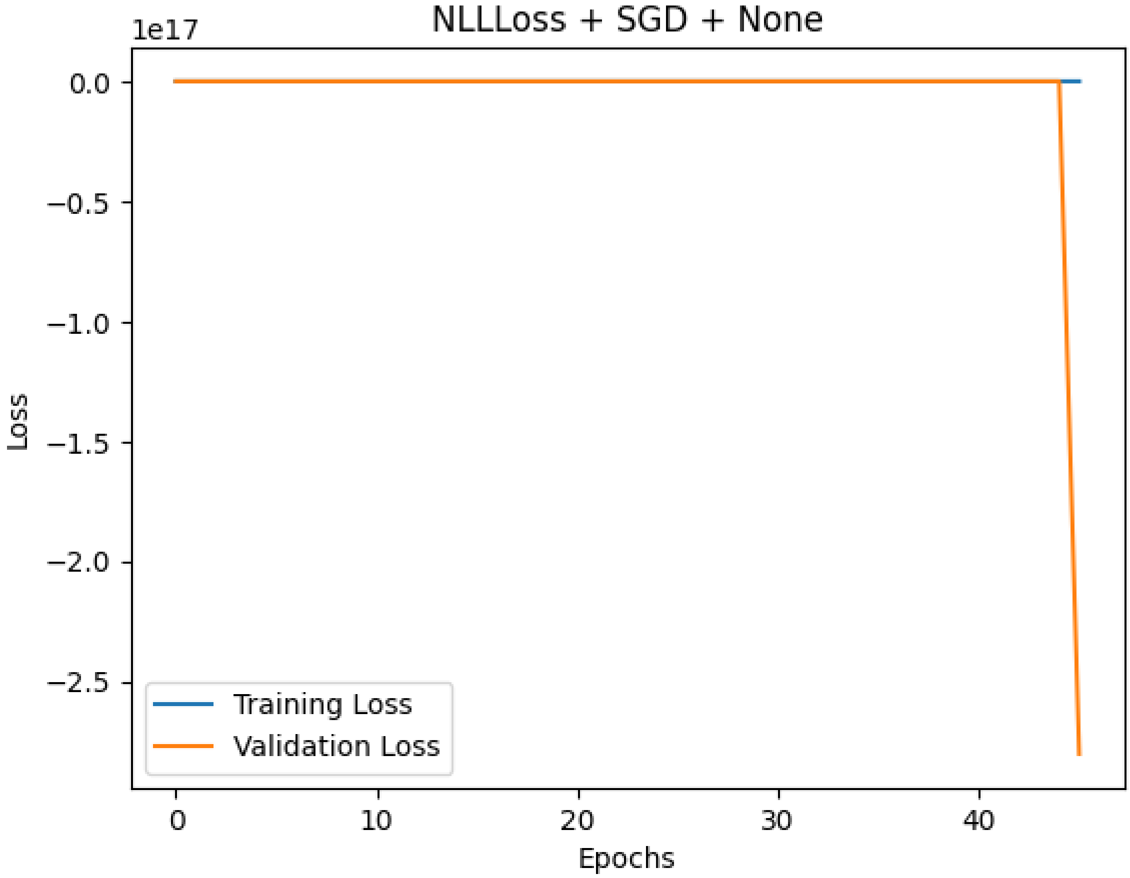
```
Epoch [1/50], Train Loss: 1.1368, Val Loss: 0.9727
Epoch [2/50], Train Loss: 0.9466, Val Loss: 0.8966
Epoch [3/50], Train Loss: 0.8763, Val Loss: 0.8442
Epoch [4/50], Train Loss: 0.8027, Val Loss: 0.7601
Epoch [5/50], Train Loss: 0.7203, Val Loss: 0.6761
Epoch [6/50], Train Loss: 0.6620, Val Loss: 0.6070
Epoch [7/50], Train Loss: 0.6017, Val Loss: 0.5540
Epoch [8/50], Train Loss: 0.5417, Val Loss: 0.5087
Epoch [9/50], Train Loss: 0.4981, Val Loss: 0.4716
Epoch [10/50], Train Loss: 0.4626, Val Loss: 0.4362
Epoch [11/50], Train Loss: 0.4331, Val Loss: 0.4126
Epoch [12/50], Train Loss: 0.4096, Val Loss: 0.3898
Epoch [13/50], Train Loss: 0.3858, Val Loss: 0.3708
Epoch [14/50], Train Loss: 0.3616, Val Loss: 0.3474
Epoch [15/50], Train Loss: 0.3415, Val Loss: 0.3268
Epoch [16/50], Train Loss: 0.3220, Val Loss: 0.3137
Epoch [17/50], Train Loss: 0.3006, Val Loss: 0.2883
Epoch [18/50], Train Loss: 0.2826, Val Loss: 0.2714
Epoch [19/50], Train Loss: 0.2632, Val Loss: 0.2539
Epoch [20/50], Train Loss: 0.2457, Val Loss: 0.2416
Epoch [21/50], Train Loss: 0.2323, Val Loss: 0.2251
Epoch [22/50], Train Loss: 0.2167, Val Loss: 0.2170
Epoch [23/50], Train Loss: 0.2040, Val Loss: 0.1994
Epoch [24/50], Train Loss: 0.1943, Val Loss: 0.1945
Epoch [25/50], Train Loss: 0.1823, Val Loss: 0.1800
Epoch [26/50], Train Loss: 0.1783, Val Loss: 0.1712
Epoch [27/50], Train Loss: 0.1700, Val Loss: 0.1700
Epoch [28/50], Train Loss: 0.1570, Val Loss: 0.1568
Epoch [29/50], Train Loss: 0.1494, Val Loss: 0.1523
Epoch [30/50], Train Loss: 0.1489, Val Loss: 0.1499
Epoch [31/50], Train Loss: 0.1558, Val Loss: 0.1412
Epoch [32/50], Train Loss: 0.1349, Val Loss: 0.1370
Epoch [33/50], Train Loss: 0.1320, Val Loss: 0.1397
Epoch [34/50], Train Loss: 0.1338, Val Loss: 0.1291
Epoch [35/50], Train Loss: 0.1231, Val Loss: 0.1316
Epoch [36/50], Train Loss: 0.1302, Val Loss: 0.1279
Epoch [37/50], Train Loss: 0.1149, Val Loss: 0.1212
Epoch [38/50], Train Loss: 0.1147, Val Loss: 0.1244
Epoch [39/50], Train Loss: 0.1139, Val Loss: 0.1196
Epoch [40/50], Train Loss: 0.1106, Val Loss: 0.1141
Epoch [41/50], Train Loss: 0.1080, Val Loss: 0.1238
Epoch [42/50], Train Loss: 0.1085, Val Loss: 0.1125
Epoch [43/50], Train Loss: 0.1038, Val Loss: 0.1134
Epoch [44/50], Train Loss: 0.1009, Val Loss: 0.1071
Epoch [45/50], Train Loss: 0.1003, Val Loss: 0.1170
Epoch [46/50], Train Loss: 0.1083, Val Loss: 0.1088
Epoch [47/50], Train Loss: 0.0970, Val Loss: 0.1023
Epoch [48/50], Train Loss: 0.1019, Val Loss: 0.1045
Epoch [49/50], Train Loss: 0.0949, Val Loss: 0.1149
Epoch [50/50], Train Loss: 0.0975, Val Loss: 0.1032
Total time elapsed: 0.86 seconds
```


CrossEntropyLoss + Adam + L2
Training Loss
Validation Loss
Loss
Epochs
1.0
0.8
0.6
0.4
0.2
0
10
20
30
40
50

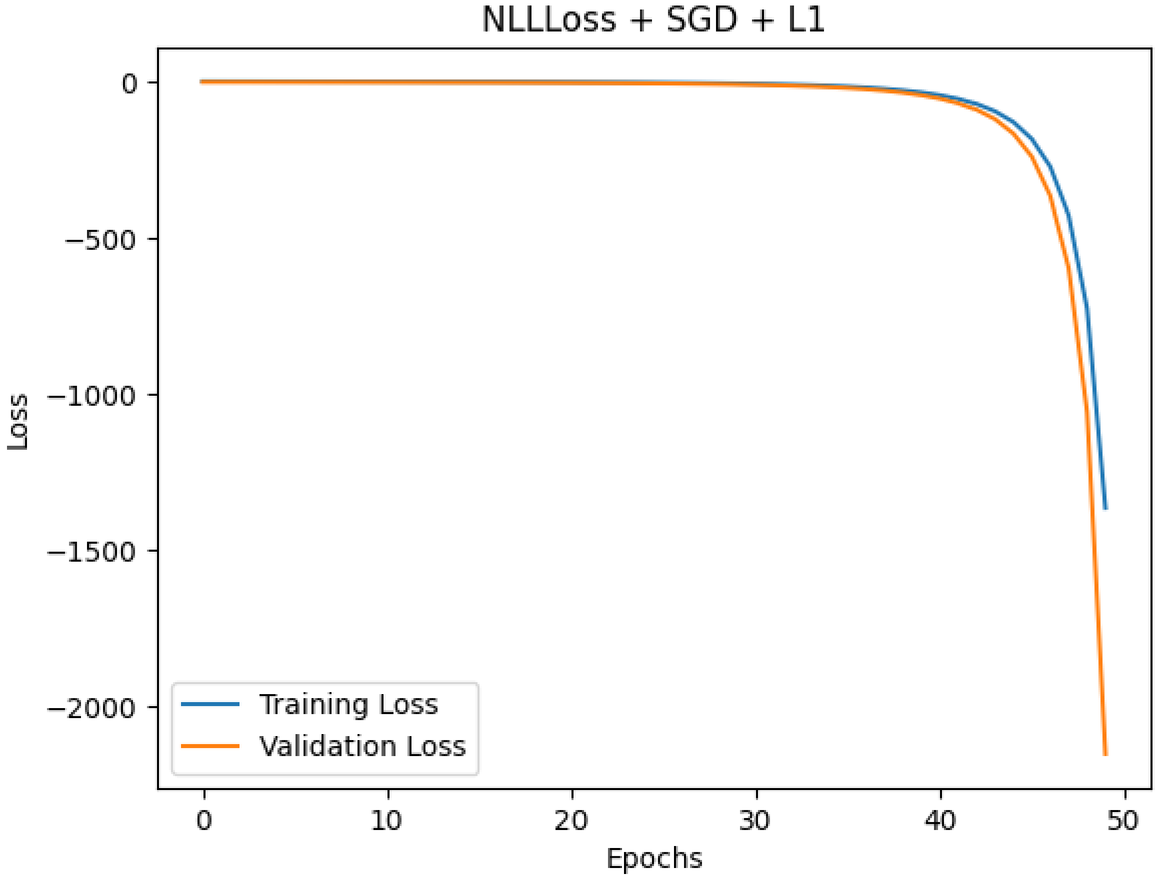
```
Epoch [1/50], Train Loss: 1.2230, Val Loss: 1.1006
Epoch [2/50], Train Loss: 1.0761, Val Loss: 1.0343
Epoch [3/50], Train Loss: 1.0273, Val Loss: 0.9896
Epoch [4/50], Train Loss: 0.9840, Val Loss: 0.9434
Epoch [5/50], Train Loss: 0.9413, Val Loss: 0.9005
Epoch [6/50], Train Loss: 0.8944, Val Loss: 0.8463
Epoch [7/50], Train Loss: 0.8420, Val Loss: 0.7882
Epoch [8/50], Train Loss: 0.7843, Val Loss: 0.7290
Epoch [9/50], Train Loss: 0.7216, Val Loss: 0.6720
Epoch [10/50], Train Loss: 0.6624, Val Loss: 0.6162
Epoch [11/50], Train Loss: 0.6080, Val Loss: 0.5648
Epoch [12/50], Train Loss: 0.5587, Val Loss: 0.5105
Epoch [13/50], Train Loss: 0.5077, Val Loss: 0.4747
Epoch [14/50], Train Loss: 0.4759, Val Loss: 0.4426
Epoch [15/50], Train Loss: 0.4352, Val Loss: 0.4115
Epoch [16/50], Train Loss: 0.4109, Val Loss: 0.3860
Epoch [17/50], Train Loss: 0.3916, Val Loss: 0.3720
Epoch [18/50], Train Loss: 0.3701, Val Loss: 0.3453
Epoch [19/50], Train Loss: 0.3437, Val Loss: 0.3356
Epoch [20/50], Train Loss: 0.3266, Val Loss: 0.3142
Epoch [21/50], Train Loss: 0.3085, Val Loss: 0.2964
Epoch [22/50], Train Loss: 0.2987, Val Loss: 0.2823
Epoch [23/50], Train Loss: 0.2793, Val Loss: 0.2755
Epoch [24/50], Train Loss: 0.2671, Val Loss: 0.2564
Epoch [25/50], Train Loss: 0.2494, Val Loss: 0.2488
Epoch [26/50], Train Loss: 0.2354, Val Loss: 0.2323
Epoch [27/50], Train Loss: 0.2258, Val Loss: 0.2225
Epoch [28/50], Train Loss: 0.2117, Val Loss: 0.2142
Epoch [29/50], Train Loss: 0.2024, Val Loss: 0.2055
Epoch [30/50], Train Loss: 0.1933, Val Loss: 0.1940
Epoch [31/50], Train Loss: 0.1814, Val Loss: 0.1922
Epoch [32/50], Train Loss: 0.1795, Val Loss: 0.1801
Epoch [33/50], Train Loss: 0.1819, Val Loss: 0.1731
Epoch [34/50], Train Loss: 0.1596, Val Loss: 0.1804
Epoch [35/50], Train Loss: 0.1569, Val Loss: 0.1621
Epoch [36/50], Train Loss: 0.1575, Val Loss: 0.1567
Epoch [37/50], Train Loss: 0.1496, Val Loss: 0.1538
Epoch [38/50], Train Loss: 0.1464, Val Loss: 0.1466
Epoch [39/50], Train Loss: 0.1330, Val Loss: 0.1513
Epoch [40/50], Train Loss: 0.1350, Val Loss: 0.1384
Epoch [41/50], Train Loss: 0.1253, Val Loss: 0.1420
Epoch [42/50], Train Loss: 0.1257, Val Loss: 0.1301
Epoch [43/50], Train Loss: 0.1207, Val Loss: 0.1324
Epoch [44/50], Train Loss: 0.1156, Val Loss: 0.1252
Epoch [45/50], Train Loss: 0.1155, Val Loss: 0.1213
Epoch [46/50], Train Loss: 0.1106, Val Loss: 0.1279
Epoch [47/50], Train Loss: 0.1082, Val Loss: 0.1168
Epoch [48/50], Train Loss: 0.1159, Val Loss: 0.1174
Epoch [49/50], Train Loss: 0.1046, Val Loss: 0.1201
Epoch [50/50], Train Loss: 0.1014, Val Loss: 0.1095
Total time elapsed: 0.86 seconds
```


CrossEntropyLoss + Adam + Dropout
Training Loss
Validation Loss
Loss
1.2
1.0
0.8
0.6
0.4
0.2
0
10
20
30
40
50
Epochs

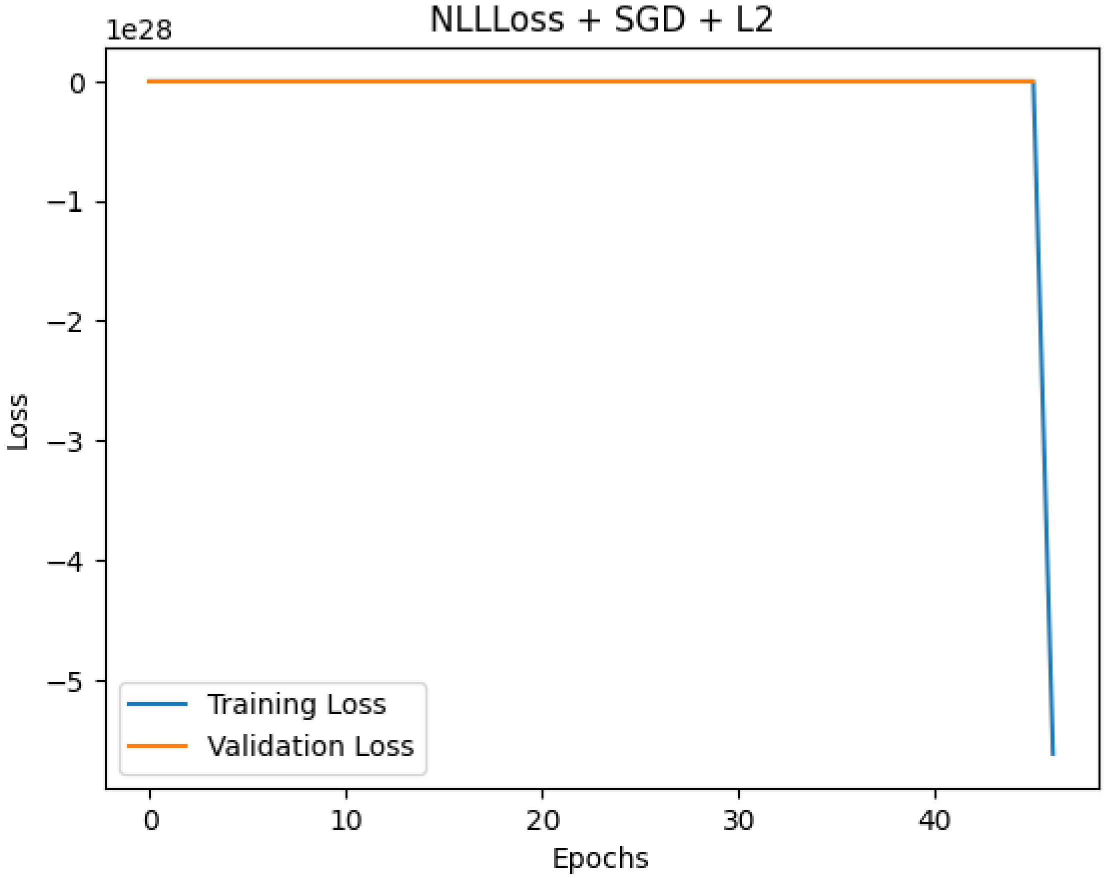
```
Epoch [1/50], Train Loss: -0.1613, Val Loss: -0.2009
Epoch [2/50], Train Loss: -0.2548, Val Loss: -0.2998
Epoch [3/50], Train Loss: -0.3527, Val Loss: -0.3974
Epoch [4/50], Train Loss: -0.4497, Val Loss: -0.4994
Epoch [5/50], Train Loss: -0.5546, Val Loss: -0.6132
Epoch [6/50], Train Loss: -0.6710, Val Loss: -0.7363
Epoch [7/50], Train Loss: -0.7968, Val Loss: -0.8716
Epoch [8/50], Train Loss: -0.9349, Val Loss: -1.0178
Epoch [9/50], Train Loss: -1.0845, Val Loss: -1.1803
Epoch [10/50], Train Loss: -1.2508, Val Loss: -1.3603
Epoch [11/50], Train Loss: -1.4360, Val Loss: -1.5620
Epoch [12/50], Train Loss: -1.6434, Val Loss: -1.7890
Epoch [13/50], Train Loss: -1.8759, Val Loss: -2.0447
Epoch [14/50], Train Loss: -2.1420, Val Loss: -2.3400
Epoch [15/50], Train Loss: -2.4514, Val Loss: -2.6910
Epoch [16/50], Train Loss: -2.8167, Val Loss: -3.1020
Epoch [17/50], Train Loss: -3.2381, Val Loss: -3.5662
Epoch [18/50], Train Loss: -3.7155, Val Loss: -4.1032
Epoch [19/50], Train Loss: -4.2717, Val Loss: -4.7181
Epoch [20/50], Train Loss: -4.9074, Val Loss: -5.4357
Epoch [21/50], Train Loss: -5.6522, Val Loss: -6.2796
Epoch [22/50], Train Loss: -6.5242, Val Loss: -7.2828
Epoch [23/50], Train Loss: -7.5715, Val Loss: -8.4704
Epoch [24/50], Train Loss: -8.8091, Val Loss: -9.8931
Epoch [25/50], Train Loss: -10.2982, Val Loss: -11.6080
Epoch [26/50], Train Loss: -12.1013, Val Loss: -13.7006
Epoch [27/50], Train Loss: -14.3024, Val Loss: -16.2795
Epoch [28/50], Train Loss: -17.0452, Val Loss: -19.4145
Epoch [29/50], Train Loss: -20.3836, Val Loss: -23.4414
Epoch [30/50], Train Loss: -24.6853, Val Loss: -28.6616
Epoch [31/50], Train Loss: -30.3197, Val Loss: -35.4089
Epoch [32/50], Train Loss: -37.6427, Val Loss: -44.3156
Epoch [33/50], Train Loss: -47.3841, Val Loss: -56.2187
Epoch [34/50], Train Loss: -60.5327, Val Loss: -72.7927
Epoch [35/50], Train Loss: -79.1604, Val Loss: -96.4189
Epoch [36/50], Train Loss: -105.9434, Val Loss: -131.2205
Epoch [37/50], Train Loss: -146.2453, Val Loss: -184.6451
Epoch [38/50], Train Loss: -209.4653, Val Loss: -272.8348
Epoch [39/50], Train Loss: -316.8177, Val Loss: -425.1823
Epoch [40/50], Train Loss: -508.6583, Val Loss: -713.8143
Epoch [41/50], Train Loss: -894.3794, Val Loss: -1344.2390
Epoch [42/50], Train Loss: -1801.0422, Val Loss: -2955.0132
Epoch [43/50], Train Loss: -4424.2371, Val Loss: -8449.1880
Epoch [44/50], Train Loss: -15850.4654, Val Loss: -40742.3562
Epoch [45/50], Train Loss: -151745.2427, Val Loss: -906329.2083
Epoch [46/50], Train Loss: -247797704160.8333, Val Loss: -280188529780964288.0
000
Epoch [47/50], Train Loss: nan, Val Loss: nan
Epoch [48/50], Train Loss: nan, Val Loss: nan
Epoch [49/50], Train Loss: nan, Val Loss: nan
Epoch [50/50], Train Loss: nan, Val Loss: nan
Total time elapsed: 0.64 seconds
```


NLLLoss + SGD + None
1e17
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
Loss
0
10
20
30
40
Epochs
Training Loss
Validation Loss

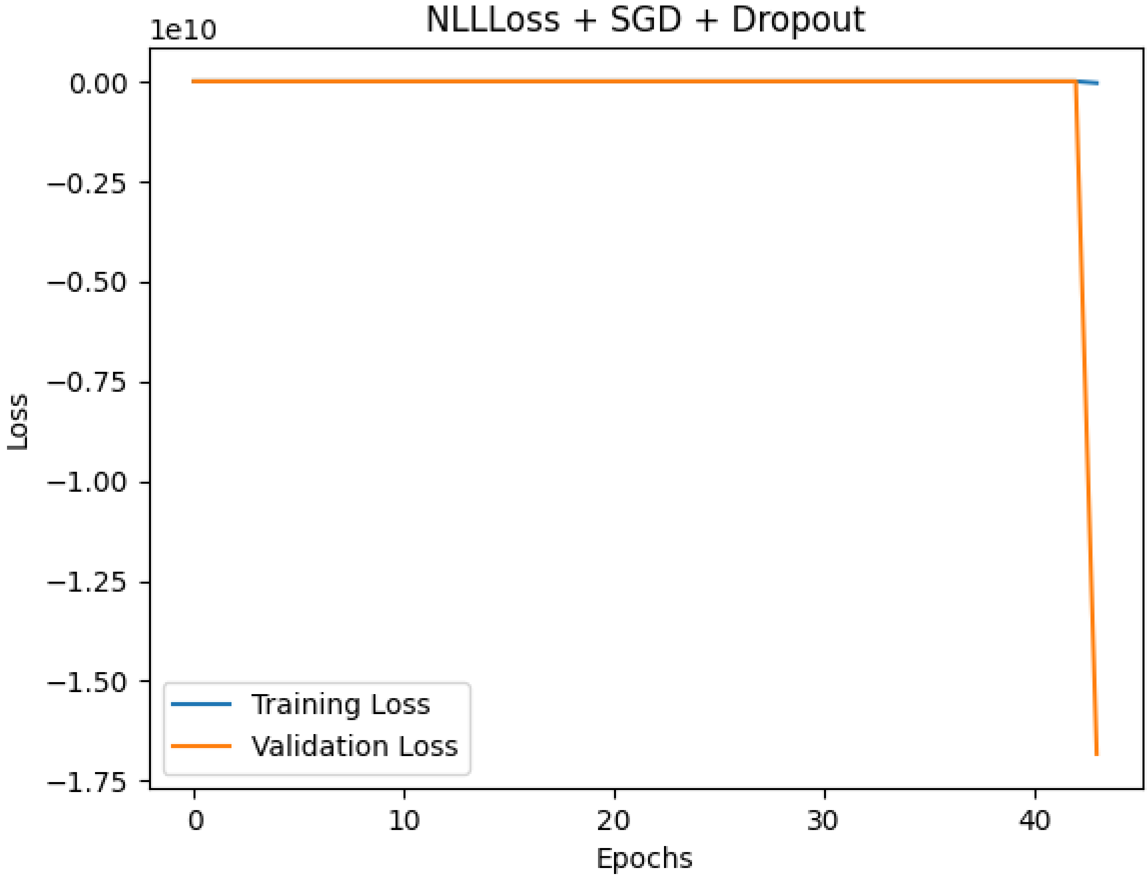
```
Epoch [1/50], Train Loss: 2.3712, Val Loss: 0.1703
Epoch [2/50], Train Loss: 2.2866, Val Loss: 0.0858
Epoch [3/50], Train Loss: 2.2040, Val Loss: 0.0027
Epoch [4/50], Train Loss: 2.1217, Val Loss: -0.0822
Epoch [5/50], Train Loss: 2.0373, Val Loss: -0.1674
Epoch [6/50], Train Loss: 1.9521, Val Loss: -0.2579
Epoch [7/50], Train Loss: 1.8615, Val Loss: -0.3564
Epoch [8/50], Train Loss: 1.7641, Val Loss: -0.4565
Epoch [9/50], Train Loss: 1.6658, Val Loss: -0.5585
Epoch [10/50], Train Loss: 1.5668, Val Loss: -0.6602
Epoch [11/50], Train Loss: 1.4666, Val Loss: -0.7678
Epoch [12/50], Train Loss: 1.3620, Val Loss: -0.8803
Epoch [13/50], Train Loss: 1.2498, Val Loss: -1.0021
Epoch [14/50], Train Loss: 1.1285, Val Loss: -1.1360
Epoch [15/50], Train Loss: 0.9960, Val Loss: -1.2826
Epoch [16/50], Train Loss: 0.8511, Val Loss: -1.4426
Epoch [17/50], Train Loss: 0.6922, Val Loss: -1.6202
Epoch [18/50], Train Loss: 0.5159, Val Loss: -1.8171
Epoch [19/50], Train Loss: 0.3200, Val Loss: -2.0334
Epoch [20/50], Train Loss: 0.1045, Val Loss: -2.2786
Epoch [21/50], Train Loss: -0.1394, Val Loss: -2.5509
Epoch [22/50], Train Loss: -0.4114, Val Loss: -2.8598
Epoch [23/50], Train Loss: -0.7199, Val Loss: -3.2065
Epoch [24/50], Train Loss: -1.0667, Val Loss: -3.6013
Epoch [25/50], Train Loss: -1.4611, Val Loss: -4.0487
Epoch [26/50], Train Loss: -1.9144, Val Loss: -4.5695
Epoch [27/50], Train Loss: -2.4391, Val Loss: -5.1754
Epoch [28/50], Train Loss: -3.0479, Val Loss: -5.8626
Epoch [29/50], Train Loss: -3.7494, Val Loss: -6.6751
Epoch [30/50], Train Loss: -4.5724, Val Loss: -7.6367
Epoch [31/50], Train Loss: -5.5486, Val Loss: -8.7618
Epoch [32/50], Train Loss: -6.7064, Val Loss: -10.1073
Epoch [33/50], Train Loss: -8.0877, Val Loss: -11.7340
Epoch [34/50], Train Loss: -9.7652, Val Loss: -13.6929
Epoch [35/50], Train Loss: -11.7957, Val Loss: -16.0797
Epoch [36/50], Train Loss: -14.2985, Val Loss: -19.0815
Epoch [37/50], Train Loss: -17.4453, Val Loss: -22.8279
Epoch [38/50], Train Loss: -21.3978, Val Loss: -27.5924
Epoch [39/50], Train Loss: -26.4594, Val Loss: -33.7296
Epoch [40/50], Train Loss: -33.0514, Val Loss: -41.8341
Epoch [41/50], Train Loss: -41.8033, Val Loss: -52.6207
Epoch [42/50], Train Loss: -53.5943, Val Loss: -67.5999
Epoch [43/50], Train Loss: -70.1302, Val Loss: -88.7245
Epoch [44/50], Train Loss: -93.7944, Val Loss: -118.9579
Epoch [45/50], Train Loss: -128.4119, Val Loss: -165.4376
Epoch [46/50], Train Loss: -182.5506, Val Loss: -239.3429
Epoch [47/50], Train Loss: -271.2424, Val Loss: -364.8414
Epoch [48/50], Train Loss: -426.4671, Val Loss: -595.4719
Epoch [49/50], Train Loss: -724.3273, Val Loss: -1056.5487
Epoch [50/50], Train Loss: -1364.7819, Val Loss: -2152.4291
Total time elapsed: 0.66 seconds
```


NLLLoss + SGD + L1
0
−500
−1000
−1500
−2000
Loss
0
10
20
30
40
50
Epochs
Training Loss
Validation Loss

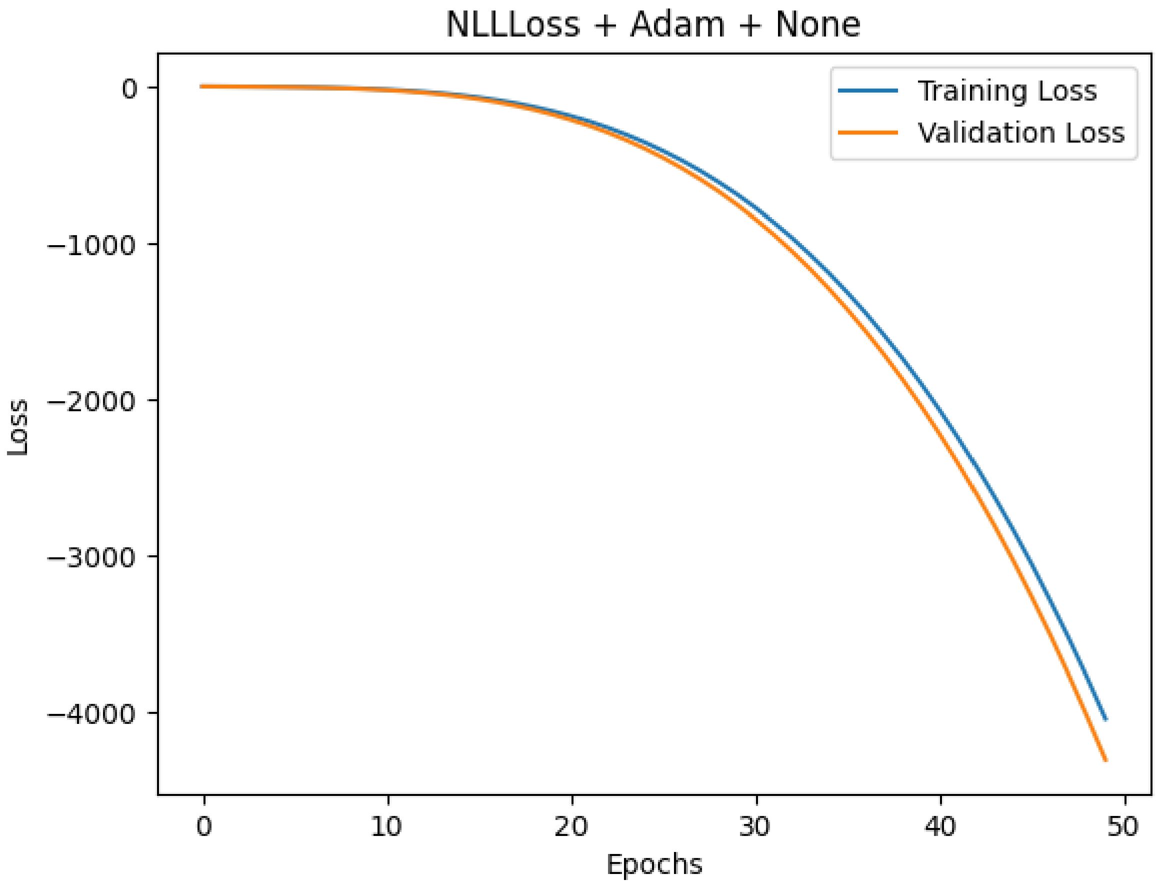
```
Epoch [1/50], Train Loss: 0.1192, Val Loss: 0.0693
Epoch [2/50], Train Loss: 0.0163, Val Loss: -0.0385
Epoch [3/50], Train Loss: -0.0877, Val Loss: -0.1458
Epoch [4/50], Train Loss: -0.1897, Val Loss: -0.2552
Epoch [5/50], Train Loss: -0.2950, Val Loss: -0.3662
Epoch [6/50], Train Loss: -0.4041, Val Loss: -0.4852
Epoch [7/50], Train Loss: -0.5209, Val Loss: -0.6121
Epoch [8/50], Train Loss: -0.6465, Val Loss: -0.7475
Epoch [9/50], Train Loss: -0.7807, Val Loss: -0.8933
Epoch [10/50], Train Loss: -0.9270, Val Loss: -1.0557
Epoch [11/50], Train Loss: -1.0902, Val Loss: -1.2344
Epoch [12/50], Train Loss: -1.2694, Val Loss: -1.4314
Epoch [13/50], Train Loss: -1.4702, Val Loss: -1.6547
Epoch [14/50], Train Loss: -1.6949, Val Loss: -1.9060
Epoch [15/50], Train Loss: -1.9540, Val Loss: -2.1960
Epoch [16/50], Train Loss: -2.2508, Val Loss: -2.5297
Epoch [17/50], Train Loss: -2.5936, Val Loss: -2.9174
Epoch [18/50], Train Loss: -2.9926, Val Loss: -3.3661
Epoch [19/50], Train Loss: -3.4553, Val Loss: -3.8927
Epoch [20/50], Train Loss: -3.9978, Val Loss: -4.5067
Epoch [21/50], Train Loss: -4.6316, Val Loss: -5.2232
Epoch [22/50], Train Loss: -5.3780, Val Loss: -6.0753
Epoch [23/50], Train Loss: -6.2615, Val Loss: -7.0761
Epoch [24/50], Train Loss: -7.3069, Val Loss: -8.2724
Epoch [25/50], Train Loss: -8.5568, Val Loss: -9.7176
Epoch [26/50], Train Loss: -10.0797, Val Loss: -11.5009
Epoch [27/50], Train Loss: -11.9552, Val Loss: -13.7114
Epoch [28/50], Train Loss: -14.2838, Val Loss: -16.3953
Epoch [29/50], Train Loss: -17.1372, Val Loss: -19.7954
Epoch [30/50], Train Loss: -20.7487, Val Loss: -24.0619
Epoch [31/50], Train Loss: -25.3130, Val Loss: -29.5307
Epoch [32/50], Train Loss: -31.2202, Val Loss: -36.6840
Epoch [33/50], Train Loss: -38.9981, Val Loss: -46.1746
Epoch [34/50], Train Loss: -49.3784, Val Loss: -59.1630
Epoch [35/50], Train Loss: -63.8116, Val Loss: -77.2442
Epoch [36/50], Train Loss: -84.1217, Val Loss: -103.4567
Epoch [37/50], Train Loss: -114.0829, Val Loss: -142.5540
Epoch [38/50], Train Loss: -159.4926, Val Loss: -203.3960
Epoch [39/50], Train Loss: -231.8110, Val Loss: -303.5857
Epoch [40/50], Train Loss: -354.5277, Val Loss: -483.1003
Epoch [41/50], Train Loss: -583.6230, Val Loss: -831.1858
Epoch [42/50], Train Loss: -1052.8740, Val Loss: -1613.4872
Epoch [43/50], Train Loss: -2206.2405, Val Loss: -3743.1032
Epoch [44/50], Train Loss: -5834.6869, Val Loss: -12001.1068
Epoch [45/50], Train Loss: -24909.7275, Val Loss: -74064.6703
Epoch [46/50], Train Loss: -454513.8563, Val Loss: -4907961.9000
Epoch [47/50], Train Loss: -5620144012303976680058886536.0000, Val Loss: -inf
Epoch [48/50], Train Loss: nan, Val Loss: nan
Epoch [49/50], Train Loss: nan, Val Loss: nan
Epoch [50/50], Train Loss: nan, Val Loss: nan
Total time elapsed: 0.80 seconds
```


NLLLoss + SGD + L2
1e28
0
−1
−2
−3
−4
−5
Loss
Training Loss
Validation Loss
0
10
20
30
40
Epochs

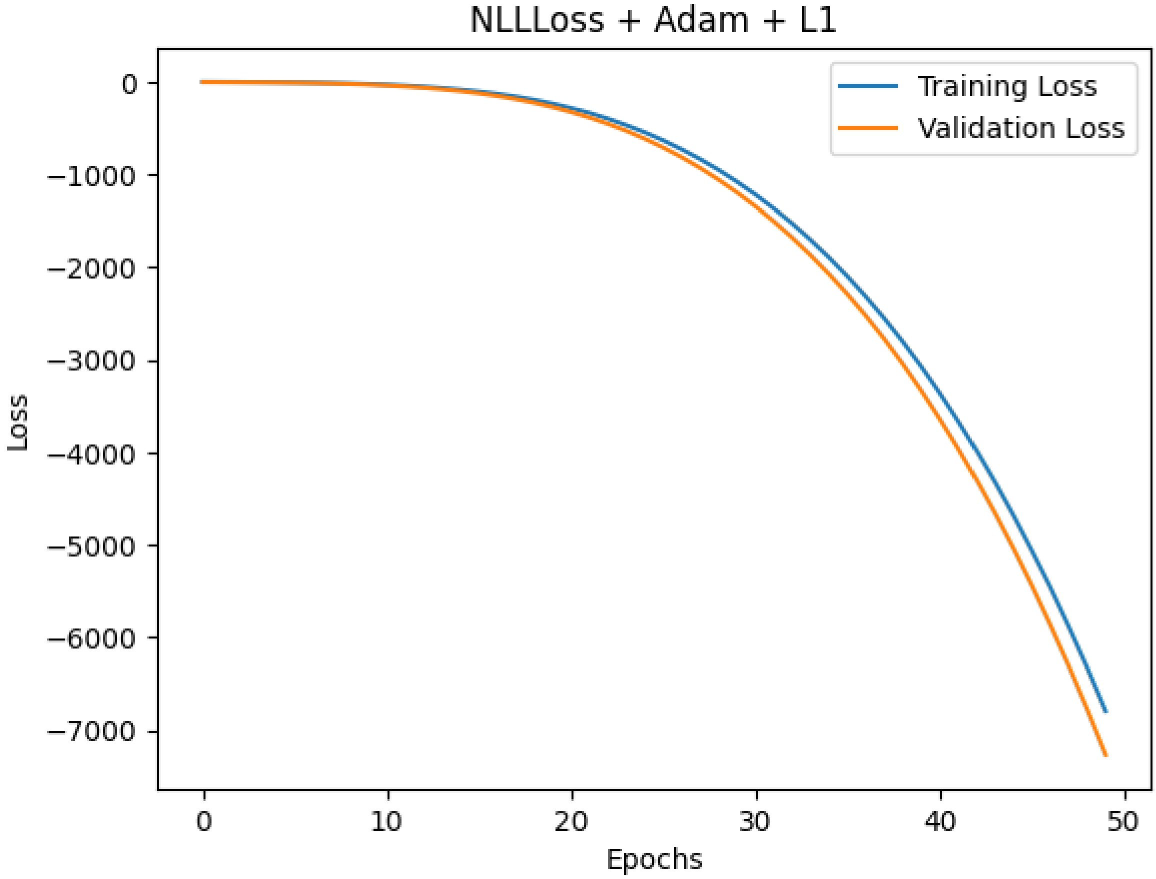
```
Epoch [1/50], Train Loss: -0.1996, Val Loss: -0.2602
Epoch [2/50], Train Loss: -0.2944, Val Loss: -0.3579
Epoch [3/50], Train Loss: -0.3916, Val Loss: -0.4605
Epoch [4/50], Train Loss: -0.4942, Val Loss: -0.5690
Epoch [5/50], Train Loss: -0.6039, Val Loss: -0.6837
Epoch [6/50], Train Loss: -0.7213, Val Loss: -0.8101
Epoch [7/50], Train Loss: -0.8512, Val Loss: -0.9511
Epoch [8/50], Train Loss: -0.9967, Val Loss: -1.1103
Epoch [9/50], Train Loss: -1.1630, Val Loss: -1.2952
Epoch [10/50], Train Loss: -1.3514, Val Loss: -1.5041
Epoch [11/50], Train Loss: -1.5686, Val Loss: -1.7436
Epoch [12/50], Train Loss: -1.8165, Val Loss: -2.0151
Epoch [13/50], Train Loss: -2.0967, Val Loss: -2.3286
Epoch [14/50], Train Loss: -2.4209, Val Loss: -2.6899
Epoch [15/50], Train Loss: -2.7946, Val Loss: -3.1089
Epoch [16/50], Train Loss: -3.2309, Val Loss: -3.6007
Epoch [17/50], Train Loss: -3.7386, Val Loss: -4.1714
Epoch [18/50], Train Loss: -4.3278, Val Loss: -4.8338
Epoch [19/50], Train Loss: -5.0203, Val Loss: -5.6133
Epoch [20/50], Train Loss: -5.8335, Val Loss: -6.5562
Epoch [21/50], Train Loss: -6.8132, Val Loss: -7.6665
Epoch [22/50], Train Loss: -7.9772, Val Loss: -8.9961
Epoch [23/50], Train Loss: -9.3719, Val Loss: -10.5990
Epoch [24/50], Train Loss: -11.0566, Val Loss: -12.5583
Epoch [25/50], Train Loss: -13.1389, Val Loss: -14.9707
Epoch [26/50], Train Loss: -15.6869, Val Loss: -17.9746
Epoch [27/50], Train Loss: -18.8999, Val Loss: -21.7608
Epoch [28/50], Train Loss: -22.9440, Val Loss: -26.5521
Epoch [29/50], Train Loss: -28.1211, Val Loss: -32.7203
Epoch [30/50], Train Loss: -34.8040, Val Loss: -40.7735
Epoch [31/50], Train Loss: -43.6825, Val Loss: -51.8482
Epoch [32/50], Train Loss: -55.9130, Val Loss: -67.0940
Epoch [33/50], Train Loss: -72.9375, Val Loss: -88.5948
Epoch [34/50], Train Loss: -97.2489, Val Loss: -120.0033
Epoch [35/50], Train Loss: -133.4628, Val Loss: -168.2386
Epoch [36/50], Train Loss: -190.3547, Val Loss: -246.0086
Epoch [37/50], Train Loss: -284.5096, Val Loss: -378.6408
Epoch [38/50], Train Loss: -450.0529, Val Loss: -628.0007
Epoch [39/50], Train Loss: -776.3498, Val Loss: -1136.4800
Epoch [40/50], Train Loss: -1489.3328, Val Loss: -2361.8669
Epoch [41/50], Train Loss: -3415.7499, Val Loss: -6194.6999
Epoch [42/50], Train Loss: -10789.4759, Val Loss: -25255.2637
Epoch [43/50], Train Loss: -71296.6747, Val Loss: -320550.8104
Epoch [44/50], Train Loss: -41670878.4833, Val Loss: -16826766540.8000
Epoch [45/50], Train Loss: nan, Val Loss: nan
Epoch [46/50], Train Loss: nan, Val Loss: nan
Epoch [47/50], Train Loss: nan, Val Loss: nan
Epoch [48/50], Train Loss: nan, Val Loss: nan
Epoch [49/50], Train Loss: nan, Val Loss: nan
Epoch [50/50], Train Loss: nan, Val Loss: nan
Total time elapsed: 0.90 seconds
```


NLLLoss + SGD + Dropout
1e10
0.00
−0.25
−0.50
−0.75
−1.00
−1.25
−1.50
−1.75
Loss
0
10
20
30
40
Epochs
Training Loss
Validation Loss

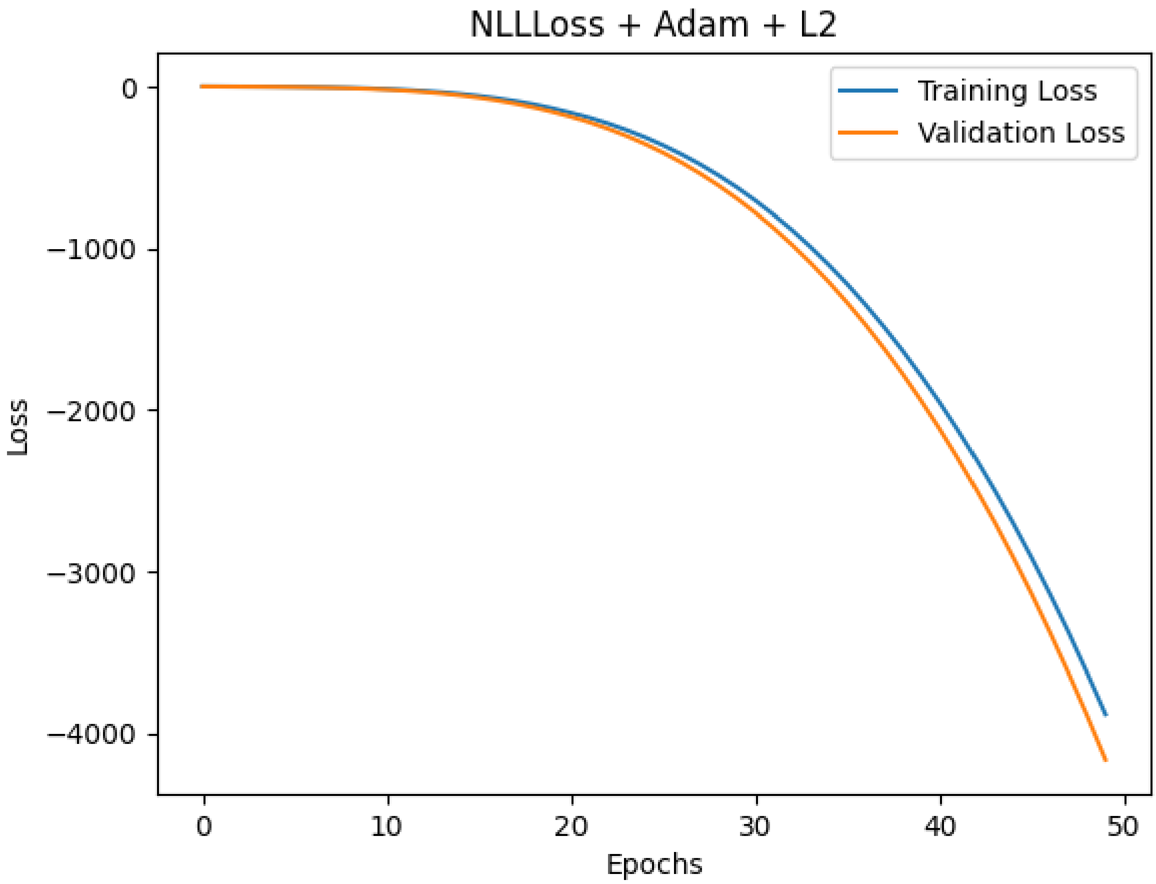
```
Epoch [1/50], Train Loss: 0.0113, Val Loss: -0.3720
Epoch [2/50], Train Loss: -0.6417, Val Loss: -1.0372
Epoch [3/50], Train Loss: -1.2954, Val Loss: -1.7244
Epoch [4/50], Train Loss: -2.0321, Val Loss: -2.5972
Epoch [5/50], Train Loss: -2.9933, Val Loss: -3.7401
Epoch [6/50], Train Loss: -4.2719, Val Loss: -5.2483
Epoch [7/50], Train Loss: -5.9313, Val Loss: -7.2777
Epoch [8/50], Train Loss: -8.1775, Val Loss: -9.9434
Epoch [9/50], Train Loss: -11.1245, Val Loss: -13.4624
Epoch [10/50], Train Loss: -15.0234, Val Loss: -17.9907
Epoch [11/50], Train Loss: -19.9432, Val Loss: -23.8396
Epoch [12/50], Train Loss: -26.2979, Val Loss: -31.2209
Epoch [13/50], Train Loss: -34.2141, Val Loss: -40.3789
Epoch [14/50], Train Loss: -44.0833, Val Loss: -51.6424
Epoch [15/50], Train Loss: -56.0018, Val Loss: -65.2840
Epoch [16/50], Train Loss: -70.4718, Val Loss: -81.6145
Epoch [17/50], Train Loss: -87.5664, Val Loss: -100.9611
Epoch [18/50], Train Loss: -107.5820, Val Loss: -123.5860
Epoch [19/50], Train Loss: -131.1631, Val Loss: -149.6664
Epoch [20/50], Train Loss: -158.4318, Val Loss: -179.6150
Epoch [21/50], Train Loss: -189.2934, Val Loss: -213.9375
Epoch [22/50], Train Loss: -224.8120, Val Loss: -252.7115
Epoch [23/50], Train Loss: -264.6469, Val Loss: -296.1125
Epoch [24/50], Train Loss: -309.2103, Val Loss: -344.5647
Epoch [25/50], Train Loss: -358.4491, Val Loss: -398.4684
Epoch [26/50], Train Loss: -413.5070, Val Loss: -457.6473
Epoch [27/50], Train Loss: -473.3394, Val Loss: -522.8159
Epoch [28/50], Train Loss: -539.2947, Val Loss: -593.7730
Epoch [29/50], Train Loss: -611.8514, Val Loss: -670.7917
Epoch [30/50], Train Loss: -689.5415, Val Loss: -755.6209
Epoch [31/50], Train Loss: -774.6968, Val Loss: -847.4945
Epoch [32/50], Train Loss: -868.5131, Val Loss: -946.4560
Epoch [33/50], Train Loss: -967.6702, Val Loss: -1053.9873
Epoch [34/50], Train Loss: -1076.8641, Val Loss: -1168.6103
Epoch [35/50], Train Loss: -1191.5390, Val Loss: -1292.5641
Epoch [36/50], Train Loss: -1316.1331, Val Loss: -1424.6793
Epoch [37/50], Train Loss: -1448.0049, Val Loss: -1565.5134
Epoch [38/50], Train Loss: -1590.0028, Val Loss: -1714.2241
Epoch [39/50], Train Loss: -1738.8286, Val Loss: -1872.9473
Epoch [40/50], Train Loss: -1898.0106, Val Loss: -2041.9340
Epoch [41/50], Train Loss: -2066.4046, Val Loss: -2220.0819
Epoch [42/50], Train Loss: -2244.4526, Val Loss: -2408.3463
Epoch [43/50], Train Loss: -2431.8174, Val Loss: -2606.9497
Epoch [44/50], Train Loss: -2630.7908, Val Loss: -2815.5150
Epoch [45/50], Train Loss: -2838.5518, Val Loss: -3035.2418
Epoch [46/50], Train Loss: -3057.4146, Val Loss: -3266.8170
Epoch [47/50], Train Loss: -3289.3537, Val Loss: -3507.7343
Epoch [48/50], Train Loss: -3527.8469, Val Loss: -3762.2964
Epoch [49/50], Train Loss: -3779.1841, Val Loss: -4028.5960
Epoch [50/50], Train Loss: -4043.3596, Val Loss: -4305.6034
Total time elapsed: 1.09 seconds
```


NLLLoss + Adam + None
Training Loss
Validation Loss
0
−1000
−2000
−3000
−4000
Loss
0
10
20
30
40
50
Epochs

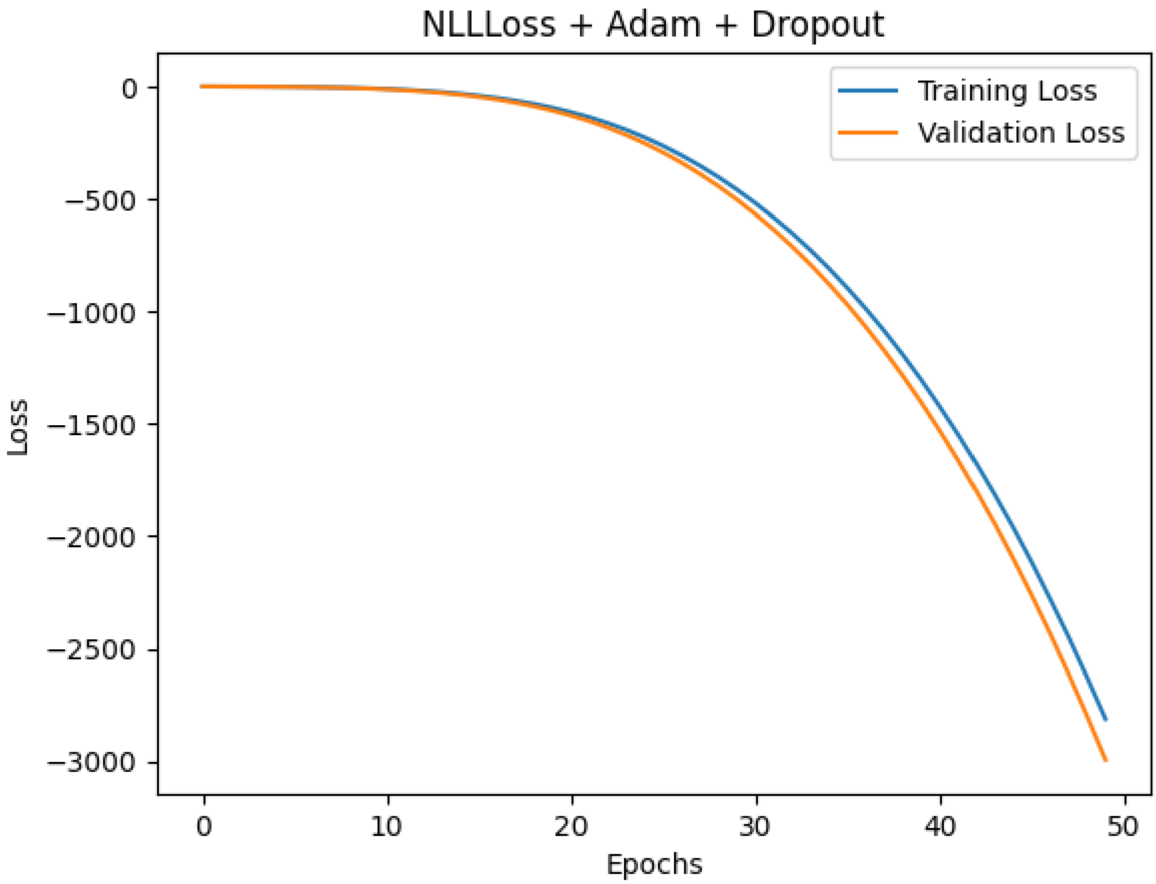
```
Epoch [1/50], Train Loss: 1.6566, Val Loss: -1.0993
Epoch [2/50], Train Loss: 0.6564, Val Loss: -2.0168
Epoch [3/50], Train Loss: -0.3560, Val Loss: -3.0666
Epoch [4/50], Train Loss: -1.5432, Val Loss: -4.4245
Epoch [5/50], Train Loss: -3.1008, Val Loss: -6.1944
Epoch [6/50], Train Loss: -5.0564, Val Loss: -8.5209
Epoch [7/50], Train Loss: -7.7069, Val Loss: -11.6602
Epoch [8/50], Train Loss: -11.1408, Val Loss: -15.8445
Epoch [9/50], Train Loss: -15.6804, Val Loss: -21.2875
Epoch [10/50], Train Loss: -21.6248, Val Loss: -28.2292
Epoch [11/50], Train Loss: -29.1141, Val Loss: -37.1466
Epoch [12/50], Train Loss: -38.5336, Val Loss: -48.3261
Epoch [13/50], Train Loss: -50.3283, Val Loss: -61.8676
Epoch [14/50], Train Loss: -64.5960, Val Loss: -78.4115
Epoch [15/50], Train Loss: -81.9311, Val Loss: -98.2353
Epoch [16/50], Train Loss: -102.6350, Val Loss: -122.2118
Epoch [17/50], Train Loss: -127.8254, Val Loss: -150.7798
Epoch [18/50], Train Loss: -157.3811, Val Loss: -184.6435
Epoch [19/50], Train Loss: -192.4356, Val Loss: -224.3958
Epoch [20/50], Train Loss: -233.7838, Val Loss: -270.3113
Epoch [21/50], Train Loss: -280.9777, Val Loss: -323.5036
Epoch [22/50], Train Loss: -334.8308, Val Loss: -384.3818
Epoch [23/50], Train Loss: -397.1781, Val Loss: -452.4555
Epoch [24/50], Train Loss: -467.3503, Val Loss: -528.4339
Epoch [25/50], Train Loss: -544.5601, Val Loss: -613.9697
Epoch [26/50], Train Loss: -631.3025, Val Loss: -708.8429
Epoch [27/50], Train Loss: -727.7870, Val Loss: -813.0988
Epoch [28/50], Train Loss: -833.2740, Val Loss: -928.0222
Epoch [29/50], Train Loss: -948.4862, Val Loss: -1054.0559
Epoch [30/50], Train Loss: -1075.5720, Val Loss: -1190.3702
Epoch [31/50], Train Loss: -1212.9088, Val Loss: -1339.7292
Epoch [32/50], Train Loss: -1363.0040, Val Loss: -1501.4879
Epoch [33/50], Train Loss: -1524.9807, Val Loss: -1676.4746
Epoch [34/50], Train Loss: -1701.5851, Val Loss: -1864.7700
Epoch [35/50], Train Loss: -1889.8880, Val Loss: -2068.2133
Epoch [36/50], Train Loss: -2094.1845, Val Loss: -2286.0780
Epoch [37/50], Train Loss: -2313.3014, Val Loss: -2518.7258
Epoch [38/50], Train Loss: -2547.2159, Val Loss: -2769.0991
Epoch [39/50], Train Loss: -2798.2849, Val Loss: -3037.7637
Epoch [40/50], Train Loss: -3065.7473, Val Loss: -3325.1543
Epoch [41/50], Train Loss: -3352.2491, Val Loss: -3630.3362
Epoch [42/50], Train Loss: -3657.0032, Val Loss: -3952.0734
Epoch [43/50], Train Loss: -3979.3111, Val Loss: -4294.4878
Epoch [44/50], Train Loss: -4320.4195, Val Loss: -4656.6276
Epoch [45/50], Train Loss: -4679.7840, Val Loss: -5038.2752
Epoch [46/50], Train Loss: -5062.3648, Val Loss: -5439.4795
Epoch [47/50], Train Loss: -5457.2541, Val Loss: -5864.2198
Epoch [48/50], Train Loss: -5881.3256, Val Loss: -6306.5277
Epoch [49/50], Train Loss: -6322.4198, Val Loss: -6773.2683
Epoch [50/50], Train Loss: -6790.5481, Val Loss: -7263.6811
Total time elapsed: 0.81 seconds
```


NLLLoss + Adam + L1
Training Loss
Validation Loss
Loss
Epochs
0
−1000
−2000
−3000
−4000
−5000
−6000
−7000
0
10
20
30
40
50

```
Epoch [1/50], Train Loss: -0.0103, Val Loss: -0.4306
Epoch [2/50], Train Loss: -0.5891, Val Loss: -0.9573
Epoch [3/50], Train Loss: -1.0580, Val Loss: -1.5392
Epoch [4/50], Train Loss: -1.6407, Val Loss: -2.2244
Epoch [5/50], Train Loss: -2.3870, Val Loss: -3.1739
Epoch [6/50], Train Loss: -3.3957, Val Loss: -4.4519
Epoch [7/50], Train Loss: -4.7618, Val Loss: -6.1350
Epoch [8/50], Train Loss: -6.5561, Val Loss: -8.3944
Epoch [9/50], Train Loss: -8.9782, Val Loss: -11.3286
Epoch [10/50], Train Loss: -12.1176, Val Loss: -15.1466
Epoch [11/50], Train Loss: -16.1508, Val Loss: -20.0643
Epoch [12/50], Train Loss: -21.3085, Val Loss: -26.2730
Epoch [13/50], Train Loss: -27.9794, Val Loss: -34.0531
Epoch [14/50], Train Loss: -36.1982, Val Loss: -43.8274
Epoch [15/50], Train Loss: -46.4060, Val Loss: -55.7816
Epoch [16/50], Train Loss: -58.8288, Val Loss: -70.2362
Epoch [17/50], Train Loss: -73.8682, Val Loss: -87.2187
Epoch [18/50], Train Loss: -91.4900, Val Loss: -107.3037
Epoch [19/50], Train Loss: -111.6256, Val Loss: -130.7653
Epoch [20/50], Train Loss: -135.8757, Val Loss: -157.5339
Epoch [21/50], Train Loss: -163.5618, Val Loss: -187.9428
Epoch [22/50], Train Loss: -194.5949, Val Loss: -222.9206
Epoch [23/50], Train Loss: -229.9265, Val Loss: -262.5752
Epoch [24/50], Train Loss: -270.3310, Val Loss: -306.7712
Epoch [25/50], Train Loss: -314.9031, Val Loss: -356.4125
Epoch [26/50], Train Loss: -365.3305, Val Loss: -411.2262
Epoch [27/50], Train Loss: -421.0915, Val Loss: -472.0172
Epoch [28/50], Train Loss: -482.4592, Val Loss: -539.4572
Epoch [29/50], Train Loss: -550.4535, Val Loss: -612.9450
Epoch [30/50], Train Loss: -624.2363, Val Loss: -693.3632
Epoch [31/50], Train Loss: -705.1694, Val Loss: -780.5344
Epoch [32/50], Train Loss: -793.2617, Val Loss: -875.6203
Epoch [33/50], Train Loss: -887.9980, Val Loss: -979.0406
Epoch [34/50], Train Loss: -991.7814, Val Loss: -1090.0248
Epoch [35/50], Train Loss: -1103.2014, Val Loss: -1209.0583
Epoch [36/50], Train Loss: -1222.1288, Val Loss: -1337.4426
Epoch [37/50], Train Loss: -1350.6069, Val Loss: -1474.4529
Epoch [38/50], Train Loss: -1487.1916, Val Loss: -1621.0134
Epoch [39/50], Train Loss: -1632.8502, Val Loss: -1776.9432
Epoch [40/50], Train Loss: -1788.3391, Val Loss: -1942.2425
Epoch [41/50], Train Loss: -1951.7013, Val Loss: -2116.4896
Epoch [42/50], Train Loss: -2125.7271, Val Loss: -2300.8963
Epoch [43/50], Train Loss: -2308.6822, Val Loss: -2495.6593
Epoch [44/50], Train Loss: -2502.4040, Val Loss: -2700.3456
Epoch [45/50], Train Loss: -2705.1216, Val Loss: -2916.7024
Epoch [46/50], Train Loss: -2919.1554, Val Loss: -3143.2886
Epoch [47/50], Train Loss: -3143.8536, Val Loss: -3381.5142
Epoch [48/50], Train Loss: -3378.3845, Val Loss: -3631.4188
Epoch [49/50], Train Loss: -3627.3804, Val Loss: -3891.4012
Epoch [50/50], Train Loss: -3883.7533, Val Loss: -4164.8983
Total time elapsed: 0.84 seconds
```


NLLLoss + Adam + L2
Training Loss
Validation Loss
0
−1000
−2000
−3000
−4000
Loss
0
10
20
30
40
50
Epochs

```
Epoch [1/50], Train Loss: 0.1442, Val Loss: -0.0616
Epoch [2/50], Train Loss: -0.2415, Val Loss: -0.4256
Epoch [3/50], Train Loss: -0.6095, Val Loss: -0.8003
Epoch [4/50], Train Loss: -1.0162, Val Loss: -1.2613
Epoch [5/50], Train Loss: -1.5421, Val Loss: -1.8855
Epoch [6/50], Train Loss: -2.2451, Val Loss: -2.7104
Epoch [7/50], Train Loss: -3.1739, Val Loss: -3.8168
Epoch [8/50], Train Loss: -4.4196, Val Loss: -5.2920
Epoch [9/50], Train Loss: -6.0748, Val Loss: -7.2350
Epoch [10/50], Train Loss: -8.2590, Val Loss: -9.7952
Epoch [11/50], Train Loss: -11.0873, Val Loss: -13.1214
Epoch [12/50], Train Loss: -14.7530, Val Loss: -17.3636
Epoch [13/50], Train Loss: -19.3634, Val Loss: -22.6828
Epoch [14/50], Train Loss: -25.1234, Val Loss: -29.2837
Epoch [15/50], Train Loss: -32.1627, Val Loss: -37.3492
Epoch [16/50], Train Loss: -40.8481, Val Loss: -47.0643
Epoch [17/50], Train Loss: -51.0867, Val Loss: -58.7045
Epoch [18/50], Train Loss: -63.4010, Val Loss: -72.5042
Epoch [19/50], Train Loss: -77.8711, Val Loss: -88.7017
Epoch [20/50], Train Loss: -94.7785, Val Loss: -107.6424
Epoch [21/50], Train Loss: -114.6637, Val Loss: -129.5179
Epoch [22/50], Train Loss: -137.4702, Val Loss: -154.9117
Epoch [23/50], Train Loss: -163.8553, Val Loss: -183.7762
Epoch [24/50], Train Loss: -193.4932, Val Loss: -216.5976
Epoch [25/50], Train Loss: -227.1933, Val Loss: -253.3232
Epoch [26/50], Train Loss: -264.8406, Val Loss: -294.2545
Epoch [27/50], Train Loss: -306.9042, Val Loss: -339.4772
Epoch [28/50], Train Loss: -352.8468, Val Loss: -389.3287
Epoch [29/50], Train Loss: -403.3459, Val Loss: -443.6565
Epoch [30/50], Train Loss: -458.4963, Val Loss: -502.8332
Epoch [31/50], Train Loss: -517.9518, Val Loss: -567.2985
Epoch [32/50], Train Loss: -582.7432, Val Loss: -636.4457
Epoch [33/50], Train Loss: -651.9357, Val Loss: -710.8667
Epoch [34/50], Train Loss: -727.1093, Val Loss: -791.0929
Epoch [35/50], Train Loss: -807.7442, Val Loss: -877.3952
Epoch [36/50], Train Loss: -894.9391, Val Loss: -969.1922
Epoch [37/50], Train Loss: -986.8848, Val Loss: -1067.8791
Epoch [38/50], Train Loss: -1085.3581, Val Loss: -1172.9348
Epoch [39/50], Train Loss: -1190.4818, Val Loss: -1284.4056
Epoch [40/50], Train Loss: -1302.0142, Val Loss: -1402.5888
Epoch [41/50], Train Loss: -1420.6093, Val Loss: -1527.8348
Epoch [42/50], Train Loss: -1545.5790, Val Loss: -1660.2142
Epoch [43/50], Train Loss: -1677.6869, Val Loss: -1799.5578
Epoch [44/50], Train Loss: -1816.6262, Val Loss: -1946.5421
Epoch [45/50], Train Loss: -1962.8277, Val Loss: -2101.4629
Epoch [46/50], Train Loss: -2117.0285, Val Loss: -2264.0047
Epoch [47/50], Train Loss: -2280.2901, Val Loss: -2433.6047
Epoch [48/50], Train Loss: -2448.6268, Val Loss: -2613.0355
Epoch [49/50], Train Loss: -2627.5626, Val Loss: -2799.1514
Epoch [50/50], Train Loss: -2812.1994, Val Loss: -2993.5483
Total time elapsed: 0.82 seconds
```


NLLLoss + Adam + Dropout
Training Loss
Validation Loss
0
−500
−1000
−1500
−2000
−2500
−3000
Loss
0
10
20
30
40
50
Epochs

```
      Loss Function Optimizer Regularization  Final Training Loss  \
0  CrossEntropyLoss       SGD           None             0.975757   
1  CrossEntropyLoss       SGD             L1             3.071872   
2  CrossEntropyLoss       SGD             L2             0.905292   
3  CrossEntropyLoss       SGD        Dropout             0.823965   
4  CrossEntropyLoss      Adam           None             0.099252   
5  CrossEntropyLoss      Adam             L1             0.798778   
6  CrossEntropyLoss      Adam             L2             0.097467   
7  CrossEntropyLoss      Adam        Dropout             0.101398   
8           NLLLoss       SGD           None                  NaN   
9           NLLLoss       SGD             L1         -1364.781942   
10          NLLLoss       SGD             L2                  NaN   
11          NLLLoss       SGD        Dropout                  NaN   
12          NLLLoss      Adam           None         -4043.359603   
13          NLLLoss      Adam             L1         -6790.548145   
14          NLLLoss      Adam             L2         -3883.753271   
15          NLLLoss      Adam        Dropout         -2812.199365   

    Final Validation Loss  Validation Accuracy      Time  
0                0.964919            36.666667  1.239962  
1                0.919365            36.666667  2.183365  
2                0.892960            73.333333  1.813068  
3                0.813721            73.333333  0.656472  
4                0.115275           100.000000  0.826143  
5                0.261058            96.666667  0.822180  
6                0.103178           100.000000  0.856971  
7                0.109528            96.666667  0.858717  
8                     NaN            33.333333  0.642088  
9            -2152.429118            36.666667  0.664937  
10                    NaN            33.333333  0.803154  
11                    NaN            33.333333  0.901080  
12           -4305.603353            30.000000  1.091052  
13           -7263.681055            36.666667  0.809736  
14           -4164.898275            36.666667  0.839327  
15           -2993.548275            30.000000  0.823931  
```

🔄 12 Combinaciones para la Experimentación 🔄

🎲 Cómo se Forman las 12 Combinaciones:

1. **Funciones de Pérdida** 🏋️‍♀️ :
   - `CrossEntropyLoss`
   - `NLLLoss`
2. **Algoritmos de Optimización** 🔄 :
   - `SGD`
   - `Adam`
3. **Técnicas de Regularización** 🎯 :
   - `Sin regularización`
   - `L1`
   - `L2`
   - `Dropout`

🧩 Combinaciones:

Cada experimento es una combinación única de una función de pérdida, un algoritmo de optimización y una técnica de regularización.

Por lo tanto, el número total de combinaciones sería:

- 2 Funciones de Pérdida 🏋️ x 2 Algoritmos de Optimización 🔄 x 4 Técnicas de Regularización 🎯 = **16 combinaciones posibles**

Se hacen **12 combinaciones** manteniendo uno de los tres elementos (funciones de pérdida, algoritmos de optimización, técnicas de regularización) constante y variando los otros dos. Esto se hace para observar el impacto del tercer elemento variante en el rendimiento del modelo.

📋 Ejemplo de 12 Combinaciones:

1. CrossEntropy + SGD + Sin regularización
2. CrossEntropy + SGD + L1
3. CrossEntropy + SGD + L2
4. CrossEntropy + SGD + Dropout
5. CrossEntropy + Adam + Sin regularización
6. CrossEntropy + Adam + L1
7. CrossEntropy + Adam + L2
8. CrossEntropy + Adam + Dropout
9. NLLLoss + SGD + Sin regularización
10. NLLLoss + SGD + L1
11. NLLLoss + SGD + L2
12. NLLLoss + SGD + Dropout

Task 7 - Discusión

Discuta los resultados obtenidos de diferentes modelos. Compare la velocidad de convergencia y el rendimiento final de modelos utilizando diferentes funciones de pérdida, técnicas de regularización, y algoritmos de optimización. Explore y discuta por qué ciertas técnicas podrían conducir a un mejor rendimiento. tanto técnicas de regularización, funciones de pérdida como algoritmos de optimización

📊 Respuesta 📊

🚀 Velocidad de Convergencia y Rendimiento Final 🚀

La velocidad de convergencia y el rendimiento final varían según las funciones de pérdida, técnicas de regularización y algoritmos de optimización utilizados.

1. **Funciones de Pérdida** 🏋️ :

- **CrossEntropyLoss**: Ofrece buen rendimiento en términos de pérdida de validación y precisión, especialmente cuando se combina con Adam como optimizador.
- **NLLLoss**: No es adecuado para este problema en particular, como se evidencia por las pérdidas negativas y bajas tasas de precisión.

2. **Algoritmos de Optimización** 🔄 :

   - **SGD**: Converge más lentamente en comparación con Adam.
   - **Adam**: Converge más rápidamente y ofrece mejores resultados en términos de pérdida y precisión.

3. **Técnicas de Regularización** 🎯 :

   - **None**: Sin regularización, el modelo con Adam y CrossEntropy muestra una precisión del 100%.
   - **L1**: Aumenta el tiempo de entrenamiento y da lugar a pérdidas más altas.
   - **L2**: Ofrece buen rendimiento cuando se combina con Adam y CrossEntropy.
   - **Dropout**: Bueno para evitar el sobreajuste, pero aumenta ligeramente la pérdida de validación.

🎓 Basado en Teoría 📚

1. **CrossEntropyLoss** es generalmente más estable que **NLLLoss** porque incluye la función Softmax dentro de la pérdida, lo cual es especialmente útil para la clasificación de varias clases (Goodfellow et al., 2016).

2. **Adam** combina las ventajas de otros algoritmos de optimización como RMSProp y SGD con momento, lo que generalmente lo hace más rápido y estable (Kingma and Ba, 2015).

3. **Regularización L2** y **Dropout** son técnicas efectivas para prevenir el sobreajuste (Srivastava et al., 2014 para Dropout). El L2 es especialmente efectivo cuando se tiene un conjunto de datos pequeño, lo cual es el caso del dataset Iris.

📈 Análisis de Gráficos 📈

🎨 Visualización en Profundidad 🖼️

Los gráficos juegan un papel crucial en la comprensión del rendimiento de los modelos. Al observar las curvas de pérdida y precisión durante las épocas de entrenamiento y validación, es posible tener una visión más detallada de cómo un modelo aprende y cómo podría estar sobreajustando o subajustando.

1. **Curvas de Pérdida** 📉 :

   - Una curva de pérdida que cae rápidamente y se estabiliza indica un buen aprendizaje, mientras que fluctuaciones significativas pueden ser un signo de inestabilidad en el entrenamiento.

- Si la pérdida de validación comienza a aumentar mientras la pérdida de entrenamiento sigue disminuyendo, es un claro signo de sobreajuste.

2. **Curvas de Precisión** 🎯 :
   - Una curva de precisión que alcanza rápidamente un alto valor y se mantiene estable es ideal.
   - Si la precisión de la validación es significativamente más baja que la de entrenamiento, podríamos estar frente a un caso de sobreajuste.

📚 Basado en Teoría 🎓

1. **Curvas de Aprendizaje**: Según la teoría de aprendizaje estadístico, una brecha grande entre el rendimiento de entrenamiento y validación a menudo indica un modelo sobreajustado (Vapnik, 1998).

2. **Ritmo de Convergencia**: Un ritmo de convergencia más rápido es generalmente posible con optimizadores adaptativos como Adam, que ajustan la tasa de aprendizaje durante el entrenamiento. Esto está bien documentado en la literatura y se evidencia en nuestras curvas de pérdida más suaves con Adam (Kingma and Ba, 2015).

3. **Regularización**: Técnicas como Dropout pueden hacer que la curva de pérdida sea más ruidosa pero pueden prevenir el sobreajuste efectivamente. Esto se basa en la teoría de que Dropout aproxima la combinación de muchos modelos diferentes (Srivastava et al., 2014).

🤔 Interpretaciones y Consideraciones 🤔

El análisis de gráficos nos permite interpretar más allá de las métricas finales, ofreciendo información sobre la estabilidad y la confiabilidad del modelo. Los gráficos complementan nuestros hallazgos tabulares y ofrecen insights para futuras iteraciones y ajustes en el modelo.

Esta es una herramienta poderosa para interpretar la complejidad y el rendimiento de los modelos de aprendizaje profundo. Nos ayuda a tomar decisiones informadas sobre cómo mejorar y perfeccionar nuestros modelos.

💎 Mejor Combinación 💎

La mejor combinación, según los resultados, es la que utiliza **CrossEntropyLoss**, **Adam** y **L2 Regularization**, con la más baja pérdida de validación y una precisión del 100%.

Esto se respalda tanto por la evidencia recolectada como por la teoría existente, que sugiere que Adam es efectivo para problemas de optimización no convexos y que la regularización L2 es buena para prevenir el sobreajuste, especialmente en conjuntos de datos pequeños.

🎭 Conclusiones 🎭

La selección de la función de pérdida, el algoritmo de optimización y la técnica de regularización es crucial para el rendimiento del modelo. La mejor combinación es la que logra un equilibrio entre una rápida convergencia y un alto rendimiento, evitando al mismo tiempo el sobreajuste.