week10_checkin

December 6, 2024

1 Does a 5 layer neural network on the data in cleaned_spotify.csv

[ ]:
```
%pip install pandas numpy lightning scikit-learn torchmetrics matplotlib optuna
```

```
Requirement already satisfied: pandas in ./.venv/lib/python3.12/site-packages
(2.2.3)
Requirement already satisfied: numpy in ./.venv/lib/python3.12/site-packages
(2.1.3)
Requirement already satisfied: lightning in ./.venv/lib/python3.12/site-packages
(2.4.0)
Requirement already satisfied: scikit-learn in ./.venv/lib/python3.12/site-
packages (1.5.2)
Requirement already satisfied: torchmetrics in ./.venv/lib/python3.12/site-
packages (1.6.0)
Collecting matplotlib
  Downloading matplotlib-3.9.3-cp312-cp312-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (11 kB)
Requirement already satisfied: python-dateutil>=2.8.2 in
./.venv/lib/python3.12/site-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in ./.venv/lib/python3.12/site-
packages (from pandas) (2024.2)
Requirement already satisfied: tzdata>=2022.7 in ./.venv/lib/python3.12/site-
packages (from pandas) (2024.2)
Requirement already satisfied: PyYAML<8.0,>=5.4 in ./.venv/lib/python3.12/site-
packages (from lightning) (6.0.2)
Requirement already satisfied: fsspec<2026.0,>=2022.5.0 in
./.venv/lib/python3.12/site-packages (from
fsspec[http]<2026.0,>=2022.5.0->lightning) (2024.10.0)
Requirement already satisfied: lightning-utilities<2.0,>=0.10.0 in
./.venv/lib/python3.12/site-packages (from lightning) (0.11.9)
Requirement already satisfied: packaging<25.0,>=20.0 in
./.venv/lib/python3.12/site-packages (from lightning) (24.2)
Requirement already satisfied: torch<4.0,>=2.1.0 in ./.venv/lib/python3.12/site-
packages (from lightning) (2.5.1)
Requirement already satisfied: tqdm<6.0,>=4.57.0 in ./.venv/lib/python3.12/site-
packages (from lightning) (4.67.1)
Requirement already satisfied: typing-extensions<6.0,>=4.4.0 in
./.venv/lib/python3.12/site-packages (from lightning) (4.12.2)
```

```
Requirement already satisfied: pytorch-lightning in ./.venv/lib/python3.12/site-
packages (from lightning) (2.4.0)
Requirement already satisfied: scipy>=1.6.0 in ./.venv/lib/python3.12/site-
packages (from scikit-learn) (1.14.1)
Requirement already satisfied: joblib>=1.2.0 in ./.venv/lib/python3.12/site-
packages (from scikit-learn) (1.4.2)
Requirement already satisfied: threadpoolctl>=3.1.0 in
./.venv/lib/python3.12/site-packages (from scikit-learn) (3.5.0)
Collecting contourpy>=1.0.1 (from matplotlib)
  Downloading contourpy-1.3.1-cp312-cp312-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (5.4 kB)
Collecting cycler>=0.10 (from matplotlib)
  Using cached cycler-0.12.1-py3-none-any.whl.metadata (3.8 kB)
Collecting fonttools>=4.22.0 (from matplotlib)
  Downloading fonttools-4.55.2-cp312-cp312-
manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_17_x86_64.manylinux2014_x86_6
4.whl.metadata (164 kB)
Collecting kiwisolver>=1.3.1 (from matplotlib)
  Using cached kiwisolver-1.4.7-cp312-cp312-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (6.3 kB)
Collecting pillow>=8 (from matplotlib)
  Using cached pillow-11.0.0-cp312-cp312-manylinux_2_28_x86_64.whl.metadata (9.1
kB)
Collecting pyparsing>=2.3.1 (from matplotlib)
  Using cached pyparsing-3.2.0-py3-none-any.whl.metadata (5.0 kB)
Requirement already satisfied: aiohttp!=4.0.0a0,!=4.0.0a1 in
./.venv/lib/python3.12/site-packages (from
fsspec[http]<2026.0,>=2022.5.0->lightning) (3.11.10)
Requirement already satisfied: setuptools in ./.venv/lib/python3.12/site-
packages (from lightning-utilities<2.0,>=0.10.0->lightning) (75.6.0)
Requirement already satisfied: six>=1.5 in ./.venv/lib/python3.12/site-packages
(from python-dateutil>=2.8.2->pandas) (1.17.0)
Requirement already satisfied: filelock in ./.venv/lib/python3.12/site-packages
(from torch<4.0,>=2.1.0->lightning) (3.16.1)
Requirement already satisfied: networkx in ./.venv/lib/python3.12/site-packages
(from torch<4.0,>=2.1.0->lightning) (3.4.2)
Requirement already satisfied: jinja2 in ./.venv/lib/python3.12/site-packages
(from torch<4.0,>=2.1.0->lightning) (3.1.4)
Requirement already satisfied: nvidia-cuda-nvrtc-cu12==12.4.127 in
./.venv/lib/python3.12/site-packages (from torch<4.0,>=2.1.0->lightning)
(12.4.127)
Requirement already satisfied: nvidia-cuda-runtime-cu12==12.4.127 in
./.venv/lib/python3.12/site-packages (from torch<4.0,>=2.1.0->lightning)
(12.4.127)
Requirement already satisfied: nvidia-cuda-cupti-cu12==12.4.127 in
./.venv/lib/python3.12/site-packages (from torch<4.0,>=2.1.0->lightning)
(12.4.127)
Requirement already satisfied: nvidia-cudnn-cu12==9.1.0.70 in
```

```
./.venv/lib/python3.12/site-packages (from torch<4.0,>=2.1.0->lightning)
(9.1.0.70)
Requirement already satisfied: nvidia-cublas-cu12==12.4.5.8 in
./.venv/lib/python3.12/site-packages (from torch<4.0,>=2.1.0->lightning)
(12.4.5.8)
Requirement already satisfied: nvidia-cufft-cu12==11.2.1.3 in
./.venv/lib/python3.12/site-packages (from torch<4.0,>=2.1.0->lightning)
(11.2.1.3)
Requirement already satisfied: nvidia-curand-cu12==10.3.5.147 in
./.venv/lib/python3.12/site-packages (from torch<4.0,>=2.1.0->lightning)
(10.3.5.147)
Requirement already satisfied: nvidia-cusolver-cu12==11.6.1.9 in
./.venv/lib/python3.12/site-packages (from torch<4.0,>=2.1.0->lightning)
(11.6.1.9)
Requirement already satisfied: nvidia-cusparse-cu12==12.3.1.170 in
./.venv/lib/python3.12/site-packages (from torch<4.0,>=2.1.0->lightning)
(12.3.1.170)
Requirement already satisfied: nvidia-nccl-cu12==2.21.5 in
./.venv/lib/python3.12/site-packages (from torch<4.0,>=2.1.0->lightning)
(2.21.5)
Requirement already satisfied: nvidia-nvtx-cu12==12.4.127 in
./.venv/lib/python3.12/site-packages (from torch<4.0,>=2.1.0->lightning)
(12.4.127)
Requirement already satisfied: nvidia-nvjitlink-cu12==12.4.127 in
./.venv/lib/python3.12/site-packages (from torch<4.0,>=2.1.0->lightning)
(12.4.127)
Requirement already satisfied: triton==3.1.0 in ./.venv/lib/python3.12/site-
packages (from torch<4.0,>=2.1.0->lightning) (3.1.0)
Requirement already satisfied: sympy==1.13.1 in ./.venv/lib/python3.12/site-
packages (from torch<4.0,>=2.1.0->lightning) (1.13.1)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in
./.venv/lib/python3.12/site-packages (from
sympy==1.13.1->torch<4.0,>=2.1.0->lightning) (1.3.0)
Requirement already satisfied: aiohappyeyeballs>=2.3.0 in
./.venv/lib/python3.12/site-packages (from
aiohttp!=4.0.0a0,!=4.0.0a1->fsspec[http]<2026.0,>=2022.5.0->lightning) (2.4.4)
Requirement already satisfied: aiosignal>=1.1.2 in ./.venv/lib/python3.12/site-
packages (from
aiohttp!=4.0.0a0,!=4.0.0a1->fsspec[http]<2026.0,>=2022.5.0->lightning) (1.3.1)
Requirement already satisfied: attrs>=17.3.0 in ./.venv/lib/python3.12/site-
packages (from
aiohttp!=4.0.0a0,!=4.0.0a1->fsspec[http]<2026.0,>=2022.5.0->lightning) (24.2.0)
Requirement already satisfied: frozenlist>=1.1.1 in ./.venv/lib/python3.12/site-
packages (from
aiohttp!=4.0.0a0,!=4.0.0a1->fsspec[http]<2026.0,>=2022.5.0->lightning) (1.5.0)
Requirement already satisfied: multidict<7.0,>=4.5 in
./.venv/lib/python3.12/site-packages (from
aiohttp!=4.0.0a0,!=4.0.0a1->fsspec[http]<2026.0,>=2022.5.0->lightning) (6.1.0)
```

```
Requirement already satisfied: propcache>=0.2.0 in ./.venv/lib/python3.12/site-
packages (from
aiohttp!=4.0.0a0,!=4.0.0a1->fsspec[http]<2026.0,>=2022.5.0->lightning) (0.2.1)
Requirement already satisfied: yarl<2.0,>=1.17.0 in ./.venv/lib/python3.12/site-
packages (from
aiohttp!=4.0.0a0,!=4.0.0a1->fsspec[http]<2026.0,>=2022.5.0->lightning) (1.18.3)
Requirement already satisfied: MarkupSafe>=2.0 in ./.venv/lib/python3.12/site-
packages (from jinja2->torch<4.0,>=2.1.0->lightning) (3.0.2)
Requirement already satisfied: idna>=2.0 in ./.venv/lib/python3.12/site-packages
(from yarl<2.0,>=1.17.0->aiohttp!=4.0.0a0,!=4.0.0a1-
>fsspec[http]<2026.0,>=2022.5.0->lightning) (3.10)
Downloading
matplotlib-3.9.3-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (8.3
MB)
                          8.3/8.3 MB
48.5 MB/s eta 0:00:00
Downloading
contourpy-1.3.1-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (323
kB)
Using cached cycler-0.12.1-py3-none-any.whl (8.3 kB)
Downloading fonttools-4.55.2-cp312-cp312-
manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_17_x86_64.manylinux2014_x86_6
4.whl (4.9 MB)
                          4.9/4.9 MB
54.0 MB/s eta 0:00:00
Using cached
kiwisolver-1.4.7-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (1.5
MB)
Using cached pillow-11.0.0-cp312-cp312-manylinux_2_28_x86_64.whl (4.4 MB)
Using cached pyparsing-3.2.0-py3-none-any.whl (106 kB)
Installing collected packages: pyparsing, pillow, kiwisolver, fonttools, cycler,
contourpy, matplotlib
Successfully installed contourpy-1.3.1 cycler-0.12.1 fonttools-4.55.2
kiwisolver-1.4.7 matplotlib-3.9.3 pillow-11.0.0 pyparsing-3.2.0

[notice] A new release of pip is
available: 24.2 -> 24.3.1
[notice] To update, run:
pip install --upgrade pip
Note: you may need to restart the kernel to use updated packages.
```

[59]:

```python
# imports
import os
os.environ['PJRT_DEVICE'] = "GPU"
import torch
import torch.nn as nn
import pandas as pd
```

```python
import numpy as np
import torch.nn.functional as F
import torch.optim as optim
import lightning as pl
from pytorch_lightning.callbacks import ModelCheckpoint
from sklearn.model_selection import train_test_split
from typing import cast, List
from sklearn.decomposition import PCA
import torch._dynamo
torch._dynamo.config.suppress_errors = True
import torchmetrics as tm
import optuna
import plotly
import plotly.express as px
```

2 Model

The model predicts track genre with respect to all other variables except for numeric ones, album name, and track name.

We made sure to one-hot encode the track genre and standardize and mean-center the data.

The model is a 3 layer neural network with 1 hidden layer. The hidden layer size and learning rate are kept as variable hyperparameters, with defaults being 0.001 and 32 (and optimimums 0.005 and 91). The hidden layer uses the ReLU activation function and the output layer uses softmax, as the output is a one-hot encoded mutli-class classification.

3 Metrics

The model is assessed by cross entropy, and accuracy is reported as a more human-readable metric.

4 Training

The model is trained using the cross entropy loss function using mini-batched gradient descent of batch size 32. It uses the PyTorch lightning package for acceleration and optimization, which handles backpropagation. PyTorch handles batching with DataLoader. In addition, the Optuna library was used to do cross-model evaluation to pick hyperparameters.

5 Data loading

This code loads the data into CSV, splits it into train/val, converts these to PyTorch tensors, converts them to PyTorch DataSets, then wraps them into PyTorch Dataloaders for batching.

[ ]:
```python
# Load the cleaned data
data = pd.read_csv('csv_outputs/cleaned_spotify.csv')

# Split the data into training and testing sets
```

```
prediction = 'track_genre'
categorical_columns = ['track_name', 'artists', 'album_name', 'track_name']
X = data.drop(columns=[prediction, 'track_id', *categorical_columns])
y = data[prediction]

# one hot encode the y values
y = pd.get_dummies(y)

# Normalize the data
X = (X - X.mean()) / X.std()
```

[80]:

```
# split into train and test
X_split, X_test, y_split, y_test = cast(
    List[pd.DataFrame],
    train_test_split(X, y, test_size=0.2, random_state=42)
)

X_train, X_val, y_train, y_val = cast(
    List[pd.DataFrame],train_test_split(
        X_split, y_split,
        test_size=0.25,
        random_state=42
    )
)

# Convert the data to tensors
X_train = torch.tensor(X_train.to_numpy(np.float32), dtype=torch.float32)
X_test = torch.tensor(X_test.to_numpy(np.float32), dtype=torch.float32)
X_val = torch.tensor(X_val.to_numpy(np.float32), dtype=torch.float32)
y_train = torch.tensor(y_train.to_numpy(np.float32), dtype=torch.float32)
y_test = torch.tensor(y_test.to_numpy(np.float32), dtype=torch.float32)
y_val = torch.tensor(y_val.to_numpy(np.float32), dtype=torch.float32)
```

[76]:

```
# Create a PyTorch dataset
train_dataset = torch.utils.data.TensorDataset(X_train, y_train)
test_dataset = torch.utils.data.TensorDataset(X_test, y_test)
val_dataset = torch.utils.data.TensorDataset(X_val, y_val)

# Create a PyTorch dataloader (to enabled batch training)
batch_size = 32
train_dataloader = torch.utils.data.DataLoader(train_dataset,␣
 ↪batch_size=batch_size, shuffle=True)
test_dataloader = torch.utils.data.DataLoader(test_dataset,␣
 ↪batch_size=batch_size, shuffle=False)
val_dataloader = torch.utils.data.DataLoader(val_dataset,␣
 ↪batch_size=batch_size, shuffle=False)
```

6 Model Definition

This code defines the model.

The model internals are specified in the `__init__` function, which shows the hidden linear layer, the activation functions.

The loss function is specified in `training_step`.

In addition, throughout epochs, metrics are stored for later graphing in `on_validation_epoch_end`

[ ]:
```python
# Define the model
class Model(pl.LightningModule):
    def __init__(
            self,
            # hyperparameters
            lr = 0.001,
            hidden_size = 32
        ):
        super(Model, self).__init__()
        # the actual model
        self.model = nn.Sequential(
            nn.Linear(X_train.shape[1], hidden_size),
            nn.ReLU(),
            nn.Linear(hidden_size, y_train.shape[1]),
            nn.Softmax(dim=1), # For multi-class classification
        )
        self.learning_rate = lr
        self.epoch_metrics = dict()

    def forward(self, x):
        return self.model(x)

    # training loss function (for backpropagation)
    def training_step(self, batch, batch_idx):
        x, y = batch
        y_hat = self(x)

        y = y.argmax(dim=1)
        loss = F.cross_entropy(y_hat, y)
        return loss

    def validation_step(self, batch, batch_idx):
        x, y = batch
        y_hat = self(x)

        y = y.argmax(dim=1)
        loss = F.cross_entropy(y_hat, y)
        self.log('val_loss', loss, prog_bar=True)
```

```
        return loss

    # used to report error metrics for graphs
    def on_validation_epoch_end(self):
        y_hat = self(X_val)

        with torch.no_grad():
            cross_entropy = F.cross_entropy(y_hat, y_val.argmax(dim=1))
            accuracy = tm.Accuracy(task="multiclass", num_classes=y_val.
↪shape[1])(y_hat, y_val.argmax(dim=1))
            self.epoch_metrics[self.current_epoch] = dict(
                cross_entropy=cross_entropy,
                accuracy=accuracy
            )


    def configure_optimizers(self):
        return torch.optim.Adam(self.parameters(), lr=self.learning_rate)
```

[ ]:
```
# Hyperparameters, learned from below
lr = 0.005
hidden_size = 91

# Train the model
model = Model(lr, hidden_size)
model.train()

trainer = pl.Trainer(
    max_epochs=100,
    accelerator='cpu',
    default_root_dir="w10checkpoints/",
    accumulate_grad_batches=7
)

ckpt_path=None
# to restore previous session's model parameters
# ckpt_path="./w10-epoch-99.ckpt"
trainer.fit(model, train_dataloader, val_dataloader, ckpt_path=ckpt_path)
```

```
GPU available: True (cuda), used: False
TPU available: False, using: 0 TPU cores
HPU available: False, using: 0 HPUs
/home/ketexon/programming/csm148-spotiflies/.venv/lib/python3.12/site-
packages/lightning/pytorch/trainer/setup.py:177: PossibleUserWarning:

GPU available but not used. You can set it by doing
`Trainer(accelerator='gpu')`.
```

```
  | Name  | Type       | Params | Mode
---------------------------------------------
0 | model | Sequential | 11.9 K | train
---------------------------------------------
11.9 K    Trainable params
0         Non-trainable params
11.9 K    Total params
0.048     Total estimated model params size (MB)
5         Modules in train mode
0         Modules in eval mode
```

```
/home/ketexon/programming/csm148-spotiflies/.venv/lib/python3.12/site-
packages/lightning/pytorch/trainer/connectors/data_connector.py:424:
PossibleUserWarning:

The 'val_dataloader' does not have many workers which may be a bottleneck.
Consider increasing the value of the `num_workers` argument` to `num_workers=11`
in the `DataLoader` to improve performance.

/home/ketexon/programming/csm148-spotiflies/.venv/lib/python3.12/site-
packages/lightning/pytorch/trainer/connectors/data_connector.py:424:
PossibleUserWarning:

The 'train_dataloader' does not have many workers which may be a bottleneck.
Consider increasing the value of the `num_workers` argument` to `num_workers=11`
in the `DataLoader` to improve performance.
```

```
Epoch 99: 100%|      | 2138/2138 [00:04<00:00, 454.76it/s, v_num=96,
val_loss=4.500]

`Trainer.fit` stopped: `max_epochs=100` reached.

Epoch 99: 100%|      | 2138/2138 [00:04<00:00, 454.53it/s, v_num=96,
val_loss=4.500]
```

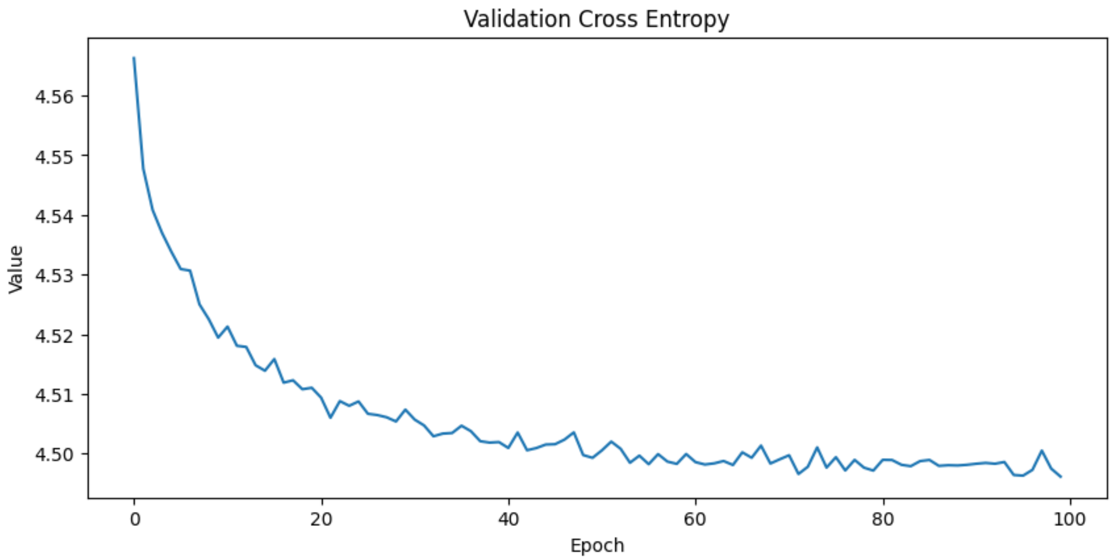
7 Error Metrics

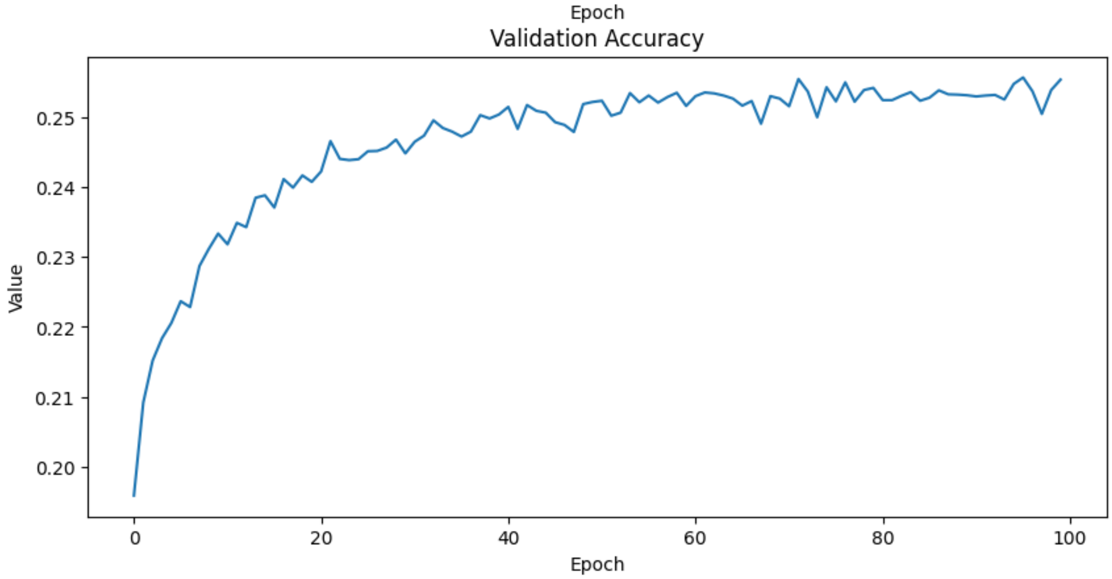
Here, we see the results of the error metrics. We see that the training accuracy ended up being 30%, while the validation accuracy 25%

[105]:
```
import matplotlib.pyplot as plt

print("Final training Loss: ", trainer.callback_metrics["val_loss"])
print(
    "Final training accuracy: ",
    tm.Accuracy(task="multiclass", num_classes=y_test.shape[1])(
```

```
        model(X_train),
        y_train.argmax(dim=1)
    )
)

epoch_metrics = list(model.epoch_metrics.values())
print("Final validation loss: ", epoch_metrics[-1]["cross_entropy"])
print("Final validation accuracy: ", epoch_metrics[-1]["accuracy"])

fig, ax = plt.subplots(2, 1, figsize=(10, 10))
ax[0].plot([m['cross_entropy'] for m in model.epoch_metrics.values()])
ax[0].set_title('Validation Cross Entropy')

ax[1].plot([m['accuracy'] for m in model.epoch_metrics.values()])
ax[1].set_title('Validation Accuracy')

for a in ax:
    a.set_xlabel('Epoch')
    a.set_ylabel('Value')

None
```

```
Final training Loss:  tensor(4.4971)
Final training accuracy:  tensor(0.3034)
Final validation loss:  tensor(4.4961)
Final validation accuracy:  tensor(0.2554)
```





8 Learning Hyperparameters

To learn hyperparameters, we trained the model using various hyperparameters (using the optuna library) and chose the one that created lowest loss. The hyperparameters we trained were learning rate and hidden layer size.

[ ]:
```
# Choosing hyperparameters
def objective(trial):
    learning_rate = trial.suggest_float('learning_rate', 1e-8, 1e-2, log=True)
    hidden_size = trial.suggest_int('hidden_size', 8, 128, log=True)

    model = Model(learning_rate, hidden_size)
```

```
    trainer = pl.Trainer(
        max_epochs=10,
        accelerator='cpu',
        default_root_dir="w10checkpoints/",
        accumulate_grad_batches=7
    )

    trainer.fit(model, train_dataloader, val_dataloader)

    val_loss = trainer.callback_metrics["val_loss"].item()
    return val_loss

study = optuna.create_study(direction='minimize')
study.optimize(objective, n_trials=50)

print("Optimum hyperparameters: ", study.best_params["learning_rate"])
```

The graph below shows how the hyperparameters affect loss.

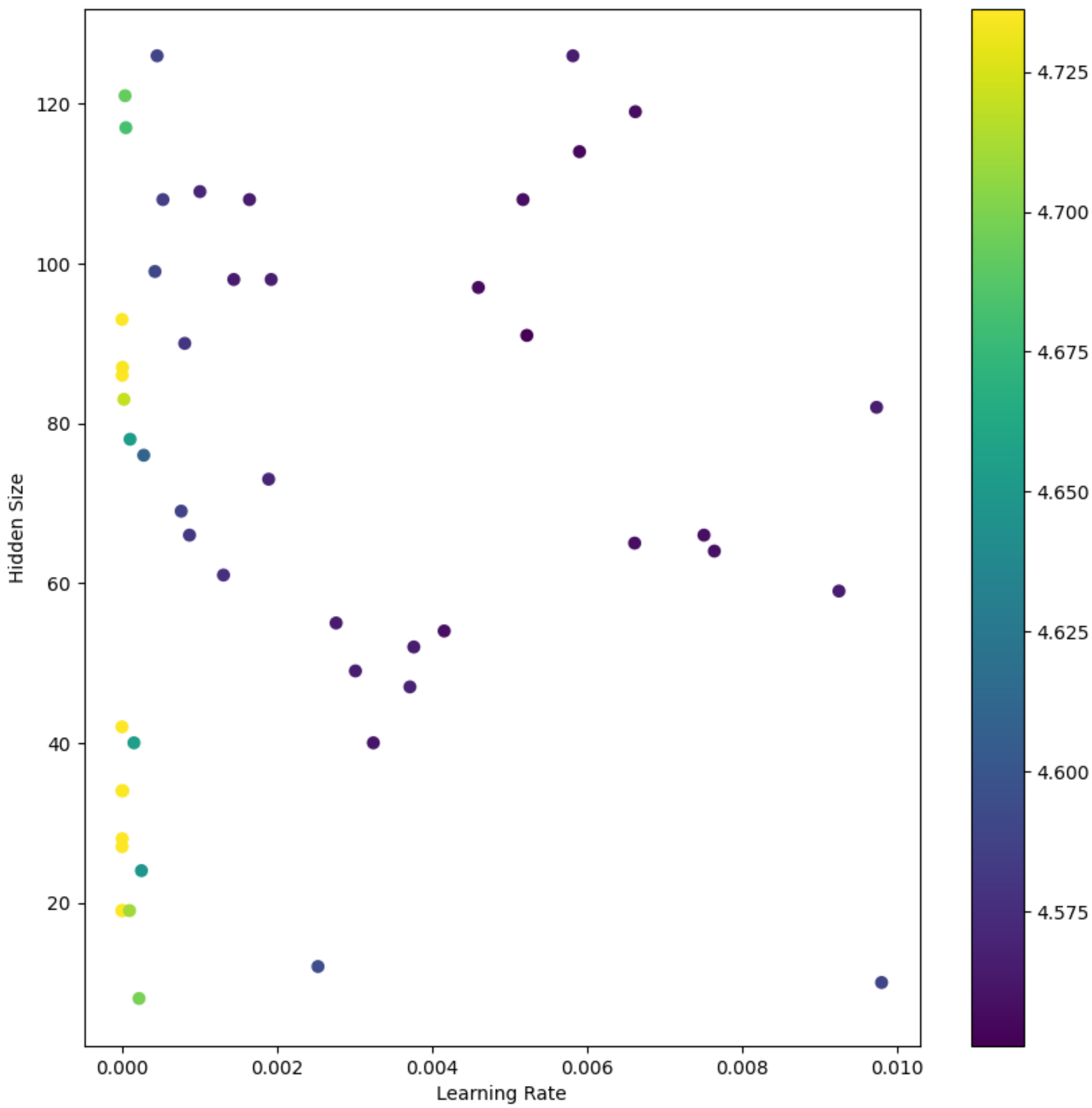
We can see that lower learning rate did affect loss, though this is likely due to the fixed epoch count.

After 0.05 LR, the loss seemed to increase.

The hidden layer size seemed to have less effect.

[ ]:
```
# plot both the learning rate vs. hidden size with
# the loss as color
fig, ax = plt.subplots(1, 1, figsize=(10, 10))
ax.scatter(
    [trial.params["learning_rate"] for trial in study.trials],
    [trial.params["hidden_size"] for trial in study.trials],
    c=[trial.value for trial in study.trials],
    cmap='viridis'
)
ax.set_xlabel('Learning Rate')
ax.set_ylabel('Hidden Size')

# legend
sm = plt.cm.ScalarMappable(cmap='viridis')
sm.set_array([trial.value for trial in study.trials])
fig.colorbar(sm, ax=ax)

# Text output
print("Optimum hyperparameters: ", study.best_params)
```

```
Optimum hyperparameters:  {'learning_rate': 0.0052227502858004605,
'hidden_size': 91}
```



[96]:
```
# plots the learning rate vs. loss and hidden size vs. loss

fig, ax = plt.subplots(1, 2, figsize=(20, 10))
ax[0].scatter(
    [trial.params["learning_rate"] for trial in study.trials],
    [trial.value for trial in study.trials]
)
ax[0].set_xlabel('Learning Rate')
ax[0].set_ylabel('Loss')

ax[1].scatter(
    [trial.params["hidden_size"] for trial in study.trials],
```

```
    [trial.value for trial in study.trials]
)
ax[1].set_xlabel('Hidden Size')
ax[1].set_ylabel('Loss')
```

[96]: Text(0, 0.5, 'Loss')