Description of the Task

Predict county level presidential election results based on historical economic data. Take 2024 as an example. The goal is to build a model that takes the economic data of 2021, 2022, 2023 of a county as input, and gives "R" or "D" as output.

Results

1. The specification of the neural network:

```
model = torch.nn.Sequential(
        torch.nn.Linear(72, 256, bias = True),
        torch.nn.LeakyReLU(),
        torch.nn.Linear(256, 512),
        torch.nn.LeakyReLU(),
        torch.nn.Linear(512, 256),
        torch.nn.LeakyReLU(),
        torch.nn.Linear(256, 256),
        torch.nn.LeakyReLU(),
        torch.nn.Linear(256, 1),
        torch.nn.Sigmoid(),
)
```

2. Hyperparameters:

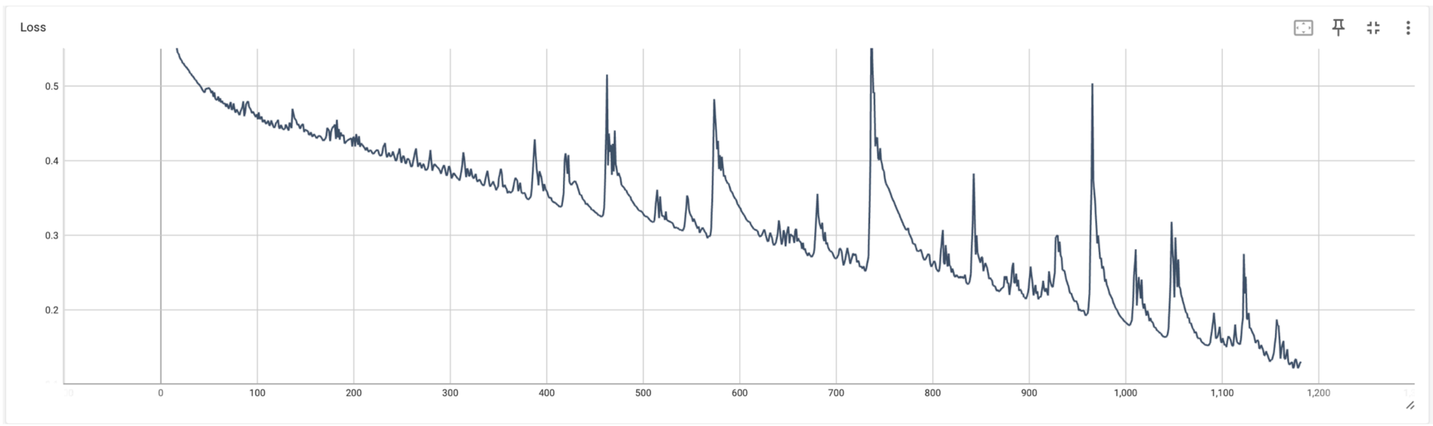
```
lr = 0.0025, epochs = 3000, training_accuracy_threshold(stops after) = 0.95, batchsize = len(training_set)
```

3. Training Method:

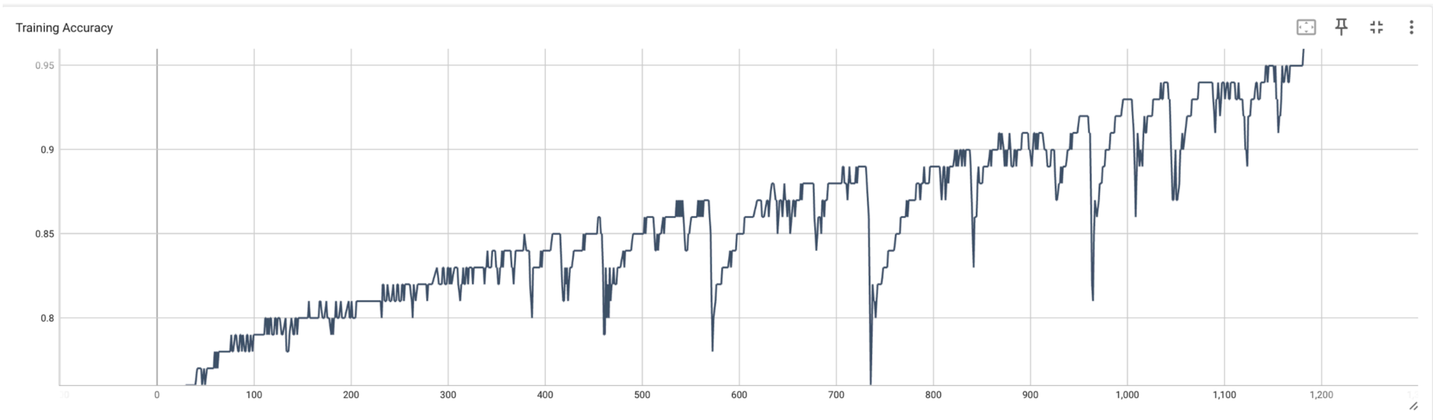
```
optimizer = torch.optim.Adam(list(nn.parameters()), lr=lr)
dataLoader = torch.utils.data.DataLoader(training_set, batch_size=batch_size, shuffle=True)

for epoch in range(epochs):
    for d, l in dataLoader:

        y_pred = nn(d)
        loss = loss_fn(y_pred, l)
        optimizer.zero_grad()
        loss.backward()
        optimizer.step()

        test_accuracy = training_accuracy(input_training_set, nn)
        test_accuracy = np.round(test_accuracy,2)
        valid_accuracy = training_accuracy(input_valid_set, nn)
```

Our training data so far consists of presidential election results and 24 economic indicators from 1969 up to 2008. We have yet to incoporate the same data from 2008 to 2024, which we already possess, into our training data. There are also a couple dozen other economic indicators we haven't used. Currently, at best, we have `0.96` training accuracy and `0.83` validation accuracy (see the training curves below drawn using TensorBoard). 90% is a priori a high level of performance, therefore it suffices as a baseline.There is also a model that takes similar data as input, which has 0.93 for "overall accuracy": https://scholar.smu.edu/cgi/viewcontent.cgi?article=1005&context=datasciencereview. But `0.83` is not far below either baselines, even without accounting the fact that we haven't used all the data we have.

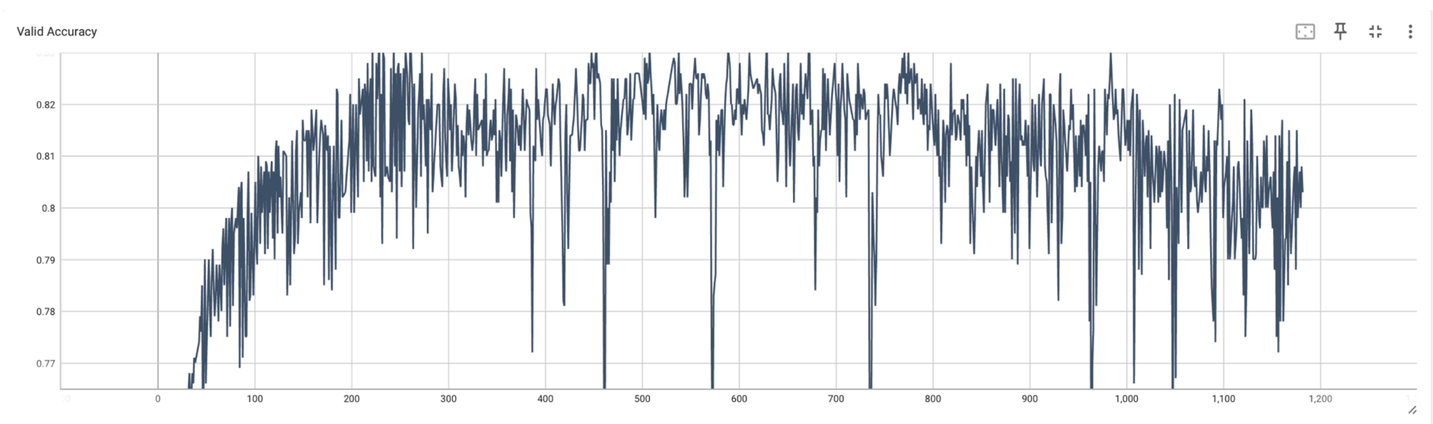
Loss



Training Accuracy



Validation Accuracy



Analysis

1. No major issue is with training method or the (hyper-)parameters of the model, as the loss and training accuracy asymptotically improves.
2. What accounts for the current result being below the baseline of 0.9-0.93 could be because validation accuracy seems to go down after 1000 epochs, while training accuracy keeps going up after. This suggests overfitting. However, this could be improved by adding more data. This is reasonable because we have yet to incorporate another .csv file containing other economic indicators for all counties in the US into our training data, and currently all our data are up to 2008.