Imports & private packages

[ ] ↳ *8 cells hidden*

Training routine

[ ] ↳ *2 cells hidden*

Run

Network definition

[ ] ↳ *3 cells hidden*

Train

```
elements = None

train_loader, test_loader, val_loader = create_data_loaders(elements, (0.8, 0.2), seq_l
```

```
Dataset shape:  (411319, 5, 3)
```

```
best_net = train(net, optimizer, train_loader, test_loader, batch_size, seq_len, measur
test(best_net, test_loader)
```

```
Loss:  0.7276880381432986
Training accuracy:  0.7241678599821549
Epoch  70
Loss:  0.7166662973206311
Training accuracy:  0.7271096156511127
Epoch  80
Loss:  0.7047234161176812
Training accuracy:  0.7317532134426078
Epoch  90
Loss:  0.6953449757872863
Training accuracy:  0.7347654739994992
Epoch  100
Loss:  0.6895580910327054
Training accuracy:  0.7357744232578607
Epoch  110
Loss:  0.6845859598783358
Training accuracy:  0.7375443390652997
Epoch  120
Loss:  0.6790249567726939
Training accuracy:  0.7401894879643294
```

```
Training accuracy:  0.7401894879531...
Epoch  130
Loss:  0.6721746895575864
Training accuracy:  0.7427398199451034
Epoch  140
Loss:  0.6706536595744974
Training accuracy:  0.7432066109272851
Epoch  150
Loss:  0.66629133738378
Training accuracy:  0.7446118462798946
Epoch  160
Loss:  0.6632844073270223
Training accuracy:  0.7452245094440082
Epoch  170
Loss:  0.658924456609282
Training accuracy:  0.7471013981848638
Epoch  180
Loss:  0.6564312622776369
Training accuracy:  0.7477675478156857
Epoch  190
Loss:  0.6529772575589983
Training accuracy:  0.7489953053469448
Epoch  200
Loss:  0.6517828110589251
Training accuracy:  0.7487181482012745
Epoch  210
Loss:  0.6506194582843009
Training accuracy:  0.7504759079935525
Epoch  220
Loss:  0.6470646033968774
Training accuracy:  0.750680129048257
Epoch  230
Loss:  0.6446645664145061
Training accuracy:  0.7522798606434422
Epoch  240
Loss:  0.6429920962732922
Training accuracy:  0.7528511933560084
Test accuracy:  0.9102632627476895
0.9102632627476895
```

```
1 test(best_net, test_loader)
```

```
1 torch.save(best_net, 'ElementClassifier_var.pth')
```

Test on Barin data

```
1 net = torch.load('/content/ElementClassifier_9782_3.pth').to(device)
```

```
1 inp = torch.tensor([[[ .3,   11.403],
2                      [ .7,   22.25],
3                      [ .8,   23.364],
4                      [.9,   24.248],
5                      [1.0,   24.979]]]).to(device)
```

```
1 out = net(inp)
```

```
1 print(Encoder()(out.argmax(dim=-1).item()))
2 print(Softmax(dim=-1)(out).amax(dim=-1))
```

```
B
tensor([0.9999], grad_fn=<AmaxBackward0>)
```

```
1 test(net, test_loader)
```

```
Test accuracy:  0.9772111058201572
0.9772111058201572
```

✓ 0s completed at 9:08 PM