

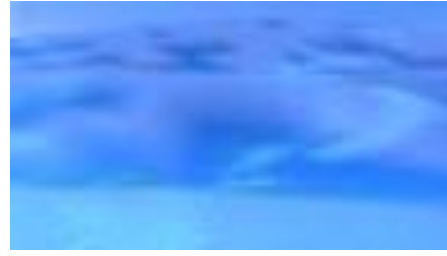
Presentation 11/03

Data discontinuity

- Episodes don't follow chronologically



Ep1: 399th frame



Ep1: 400th frame



Ep2: 1st frame



Ep2: 2nd frame

Data discontinuity

- Episodes don't follow chronologically
 - **Add episode number & position**
 - Batch size = 400?



Ep1: 400th frame



Ep2: 1st frame

	pitch	roll	episode	position_within_episode
0	15.915869	2.358416	1	0
1	14.295925	1.700391	1	1
2	6.213590	1.611325	1	2
3	1.235337	6.175461	1	3
4	1.341678	3.695468	1	4
...
215995	0.428497	-0.663964	540	395
215996	0.485498	-1.376229	540	396
215997	-0.658033	-1.328761	540	397
215998	-1.322262	-1.650465	540	398
215999	-3.697282	-1.758844	540	399

216000 rows × 4 columns

Data discontinuity

- Episodes don't follow chronologically
 - Add episode number & position
 - Batch size = 400?
 - **Stack data: each row = 1 episode**



Ep1: 400th frame



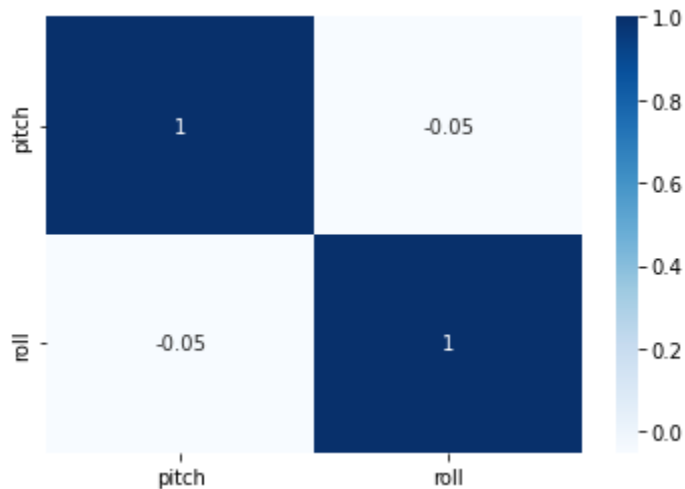
Ep2: 1st frame

episode		pr
0	1	[(15.92, 2.36), (14.3, 1.7), (6.21, 1.61), (1....
1	2	[(15.98, 2.27), (9.86, 1.36), (6.11, 1.56), (1...
2	3	[(15.96, 2.07), (6.18, 0.12), (6.08, 1.68), (1...
3	4	[(7.7, 1.58), (8.48, 0.06), (3.16, 0.86), (1.0...
4	5	[(7.81, 1.49), (7.87, 0.18), (3.13, 0.91), (0....
...
535	536	[(27.03, -7.86), (19.12, -5.18), (12.65, -3.41...
536	537	[(18.02, -5.29), (18.95, -5.79), (10.58, -2.69...
537	538	[(21.82, -6.14), (14.1, -3.58), (9.7, -2.57), ...
538	539	[(21.23, -6.12), (16.44, -4.69), (10.11, -2.68...
539	540	[(-0.65, -0.5), (-0.81, -0.24), (2.55, 0.03), ...

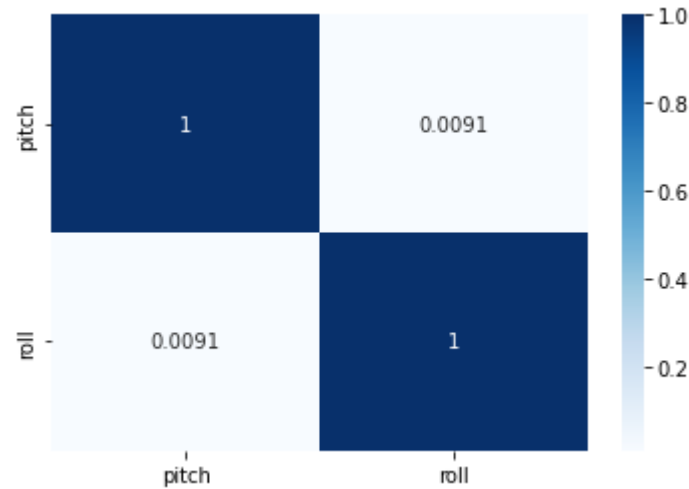
540 rows × 2 columns

Data analysis

- Low correlation between Pitch & Roll
 - Use both to predict both at the same time
 - Use both to predict one
 - One LSTM model for pitch, one for roll
 - Only use pitch to predict pitch, roll to predict roll



Episode 1



All episodes

Multivariate single output LSTM model

- Use **P**, **R**, **episode** and **position** to predict **P**
- Normalized data
 - $PR = \frac{angle_{original} + 90}{180} \cdot 2 - 1$
 - Episode & position
 - MinMaxScaler

	pitch	roll	episode	position_within_episode
0	0.176843	0.026205	-1.0	-1.000000
1	0.158844	0.018893	-1.0	-0.994987
2	0.069040	0.017904	-1.0	-0.989975
3	0.013726	0.068616	-1.0	-0.984962
4	0.014908	0.041061	-1.0	-0.979950
...
215995	0.004761	-0.007377	1.0	0.979950
215996	0.005394	-0.015291	1.0	0.984962
215997	-0.007311	-0.014764	1.0	0.989975
215998	-0.014692	-0.018338	1.0	0.994987
215999	-0.041081	-0.019543	1.0	1.000000

216000 rows × 4 columns

Multivariate single output LSTM model

- Use P, R, episode and position to predict P
- Normalized data
- Sequence length = 30
- Single Pitch value output
 - Not yet multi-step LSTM

```
for item in train_dataset:  
    print(item["sequence"].shape)  
    print(item["label"].shape)  
    print(item["label"])  
    break
```

✓ 0.4s

```
torch.Size([30, 4])  
torch.Size([])  
tensor(0.0972)
```

	pitch	roll	episode	position_within_episode
0	0.176843	0.026205	-1.0	-1.000000
1	0.158844	0.018893	-1.0	-0.994987
2	0.069040	0.017904	-1.0	-0.989975
3	0.013726	0.068616	-1.0	-0.984962
4	0.014908	0.041061	-1.0	-0.979950
...
215995	0.004761	-0.007377	1.0	0.979950
215996	0.005394	-0.015291	1.0	0.984962
215997	-0.007311	-0.014764	1.0	0.989975
215998	-0.014692	-0.018338	1.0	0.994987
215999	-0.041081	-0.019543	1.0	1.000000

216000 rows × 4 columns

Multivariate single output LSTM model

- 4 input features
- 128 hidden features
- Stacked LSTM
- 1 output value
- MSE Loss for evaluation

```
class PRPredictionModel(nn.Module):  
    def __init__(self, n_features, n_hidden=128, n_layers=2):  
        super().__init__()  
  
        self.n_hidden = n_hidden  
  
        self.lstm = nn.LSTM(  
            input_size=n_features,  
            hidden_size=n_hidden,  
            batch_first=True,  
            num_layers=n_layers,  
            dropout=0.2  
        )  
  
        self.regressor = nn.Linear(n_hidden, 1)  
  
    def forward(self, x):  
        self.lstm.flatten_parameters()  
  
        _, (hidden, _) = self.lstm(x)  
        out = hidden[-1]  
  
        return self.regressor(out)
```


Results

- No results yet
- Problems with datamodule/dataloader/...
 - Train_dataloader stuck in iteration
- Above my capabilities
 - Multiple inputs, multiple outputs
 - hybrid models
 - encoder/decoder
 - discontinuous training data
 - lightweight, ...
 - not just keras.VGG16(...)
- Pytorch Udemy course

Planning March 2022

- 14/03-20/03
 - Continue Udemy course
 - Debug, train, test and evaluate LSTM PR model
- 21/03-27/03
 - Continue Udemy course
 - Create simple CNN model with only images
- 28/03-03/04
 - Train, test and evaluate CNN model
 - Combine with PR input
 - Hybrid with LSTM
 - Encoder-Decoder
 - Try to recreate Nazar's best models if possible

Planning April 2022

- 04/04-10/04
 - Recreate Nazar's best models
 - Train, test and evaluate with simulated (and real if available)
- 11/04-17/04
 - Debugging
 - Optimizing
 - Buffer for problems/delays occurred in previous weeks
- 18/04-24/04
 - Compare results simulation-real vs. criteria
 - Create new model if necessary:
 - Transformer model
- 25/04-01/05
 - Continue new model or optimize old model
 - Extensive testing

Planning May 2022

- Depends on progression
- Writing paper

Planning June 2022

- Writing paper
- Debugging
- Optimizations

Comments

- Select sequences in dataloader only from the same episodes
 - Batch size 60, sequence length 30,
 - 2 seq/batch from different random episode
- Test set for optimizations, validation set for final performance analysis
- Pitch & Roll: angular = cyclical values: $359 = -1^\circ$
 - Big impact on Loss
- Use both pitch and roll to predict both
- Testing, analysis
 - Prediction vs. truth
 - Prediction vs. error
 - Convergence, loss vs. epochs
 - MSE vs. #predicted frames
 - Roll MSE vs. Pitch MSE