Thesis Report 8 : 13 April - 20 April

Goals

- Variational Autoencoder to be able to generate new LMA features from Latent Space ✅
- PAD-Latent Space Regression ✅
- Continue working on PAD-LMA feature generation ✅

Last Week Leftovers:

None

Done

- Tweaked and improved the VAE in order to generate new LMA features
- Managed to create a VAE that uses a latent space of size 1 (a single value) in order to generate features
    - Played around with the network configurations of both the Encoder/Decoder, the weight of the reconstruction error, learning rates, types of activation functions and so on
    - Encoding/Decoding errors were okay but not great.
    - Overall average MAE over all 27 features is 0.15
    - Noted that certain features struggle much more than others in being regenerated (have way higher MAEs, whilst others have really low ones)

```
Overall MAE: 0.15670356043772807

==max_hand_distance==
MSE: 0.06012
MAE: 0.19185
Example [Regen-Real]: 0.5500726 - 0.4307294347930162

==avg_l_hand_hip_distance==
MSE: 0.01924
MAE: 0.10951
Example [Regen-Real]: 0.32496446 - 0.4309957160177905

==avg_r_hand_hip_distance==
MSE: 0.02399
MAE: 0.12261
Example [Regen-Real]: 0.33140257 - 0.574602981247248

==max_stride_length==
MSE: 0.01940
MAE: 0.10684
Example [Regen-Real]: 0.36242947 - 0.5436291098827964

==avg_l_hand_chest_distance==
MSE: 0.01152
MAE: 0.08035
Example [Regen-Real]: 0.42536807 - 0.4366265481499257

==avg_r_hand_chest_distance==
MSE: 0.01141
MAE: 0.08069
Example [Regen-Real]: 0.4126603 - 0.4853309428154969

==avg_l_elbow_hip_distance==
MSE: 0.00531
MAE: 0.05266
Example [Regen-Real]: 0.34532416 - 0.3809397134466188

==avg_r_elbow_hip_distance==
MSE: 0.00556
MAE: 0.05451
Example [Regen-Real]: 0.33311942 - 0.3194540220825596

==avg_chest_pelvis_distance==
MSE: 0.00000
MAE: 0.00007
Example [Regen-Real]: 0.28606498 - 0.286150999997311
```

```
==avg_neck_chest_distance==
MSE: 0.00000
MAE: 0.00140
Example [Regen-Real]: 0.27782476 - 0.27862898765383

==avg_neck_rotation_w==
MSE: 0.00725
MAE: 0.05805
Example [Regen-Real]: -0.0029133298 - 0.012165701914984

==avg_neck_rotation_x==
MSE: 0.06104
MAE: 0.16637
Example [Regen-Real]: -0.11023945 - 0.181021728424847

==avg_neck_rotation_y==
MSE: 0.01392
MAE: 0.08414
Example [Regen-Real]: -0.0953259 - -0.1191623970033042

==avg_neck_rotation_z==
MSE: 0.00953
MAE: 0.05798
Example [Regen-Real]: 0.9575454 - 0.977094760455076

==avg_total_body_volume==
MSE: 0.02173
MAE: 0.11463
Example [Regen-Real]: 0.31134456 - 0.5240143166405168

==avg_triangle_area_hands_neck==
MSE: 0.00365
MAE: 0.05104
Example [Regen-Real]: 0.11289087 - 0.1630560309683537

==avg_triangle_area_feet_hips==
MSE: 0.00300
MAE: 0.04344
Example [Regen-Real]: 0.1420451 - 0.1287968879463306

==l_hand_speed==
MSE: 0.08892
MAE: 0.22292
Example [Regen-Real]: 0.35017312 - 0.0511732290229953
```

```
==r_hand_speed==
MSE: 0.09465
MAE: 0.23104
Example [Regen-Real]: 0.3297714 - 0.3677404667259225

==l_foot_speed==
MSE: 0.06727
MAE: 0.18389
Example [Regen-Real]: 0.5712544 - 0.6261659226438303

==r_foot_speed==
MSE: 0.06975
MAE: 0.18928
Example [Regen-Real]: 0.47807708 - 0.7034740979538282

==neck_speed==
MSE: 0.05520
MAE: 0.17583
Example [Regen-Real]: 0.5674466 - 0.2718643096912089

==l_hand_acceleration_magnitude==
MSE: 0.32162
MAE: 0.39444
Example [Regen-Real]: 0.9785443 - 1.027498684585981

==r_hand_acceleration_magnitude==
MSE: 0.31643
MAE: 0.38389
Example [Regen-Real]: 2.8770142 - 2.02100332166924

==l_foot_acceleration_magnitude==
MSE: 0.24712
MAE: 0.34882
Example [Regen-Real]: 0.61287016 - 0.3309134284388706

==r_foot_acceleration_magnitude==
MSE: 0.28307
MAE: 0.36753
Example [Regen-Real]: 0.7664799 - 0.6173527436973426

==neck_acceleration_magnitude==
MSE: 0.23917
MAE: 0.35721
Example [Regen-Real]: 0.7739915 - 0.1702744257349307
```

- Tried to create a regression between PAD coordinates and the latent space
  - The idea behind this was that, the decoder is good enough to transform the latent space value into a set of LMA features. On the other hand, using the encoder we can get the latent space representation of our dataset, and hopefully train a regressor to go from PAD coordinates to that latent space, and then using the decoder, from that latent space into actual features.
  - However, this didn't go too well. Using XGBoostRegression the model showcased an MAE of 0.55 which is really high
  - And even in cases where the regressed latent space value is very close to the actual

value, the generated features don't get predicted as having the intended emotion. **I believe this is due to the fact that, since the decoder uses a single value for decoding, that value is super sensitive, in the sense that even if the regressed value was 0.64 and the real value was 0.69, the decoder generates wildly different LMA features!**. So unless our PAD-Latent space regression is super accurate, it won't work.

- This can possibly be alleviated by having a larger latent space (rather than just using a single value). This may help both the VAE's reconstruction accuracy, but also the whole PAD-Latent space regression. I'll be trying this approach during this following week

```
Real: [0.6967958]
Predicted: [0.647487]
```

```
model_p = xgb.XGBRegressor(verbosity=0)
model_p.load_model("../../emotion_classifier/model_training/models/l2p_dan

model_a = xgb.XGBRegressor(verbosity=0)
model_a.load_model("../../emotion_classifier/model_training/models/l2a_dan

model_d = xgb.XGBRegressor(verbosity=0)
model_d.load_model("../../emotion_classifier/model_training/models/l2d_dan

scaler = joblib.load('../../emotion_classifier/model_training/datasets/sca
```

```
regen = decoder.predict(predicted)
print(regen)
```

```
[[ 0.5518506   0.3396051   0.35146886  0.34499368  0.41565     0.4207828
   0.33057868  0.33588266  0.28606454  0.27774927  0.00991045 -0.09116153
  -0.07128949  0.95351416  0.29623687  0.11524868  0.14204371  0.43524724
   0.4640965   0.2898261   0.30749616  0.31924236  1.1339846   1.1546081
   0.72193986  0.7683582   0.70810694]]
```

```
scaled_regen = scaler.transform(regen)
generated_coordinates = (
    model_p.predict(scaled_regen),
    model_a.predict(scaled_regen),
    model_d.predict(scaled_regen)
)


print('Real: %s' % np.asarray(reg_test_X.iloc[index]))
print('Predicted: %s' % [generated_coordinates[0][0], generated_coordinate
```

```
Real: [-0.4   0.25 -0.1 ]
Predicted: [-0.14019483, -0.23435989, 0.06783479]
```

- Tried to continue using XGBoostRegression to go directly from PAD to LMA features
    - Results were ok. Some features get regressed with MAE as low as 0.05, whilst others do much poorer with MAE over 0.25.

    - Obviously each of the features has a different range of values so an MAE of 0.25 here is not as bad as an MAE of 0.25 in, for example, the LMA-PAD regression, but still
    - Worst features tend to be those associated with acceleration.
    - In a worst case scenario we could reduce the set of generated LMA features to discard those acceleration related (since they're also not the most impactful in discerning the animation's emotion)

- **Tested out an idea** - Each LMA feature correlates differently to each emotional coordinate (and vice versa). For example:

> ===CORRELATION max_stride_length=== EMOTION_P - 0.099946 EMOTION_A - 0.218093 EMOTION_D - 0.151456

- As we can see here, max stride is highly correlated to the Arousal and Dominance axis, but not correlated at all to the Pleasure axis. My idea was: **what if we select, for each LMA feature we generate, their most correlated emotional axis, and then regress using only those**. So for example, for max stride, our regressor would only consider the Arousal and Dominance values when fitting.
    - This was implemented and tested and gridsearch was used to find the optimal hyper parameters, however results weren't better than our previous regressions, and in some cases (especially accelerations and speed features) did even worse!
- Did some outlier removing data preparation on our datasets. Initially, this was done for the LMA generation, but I then went back to the **LMA-PAD model and noted that, using this new dataset without outliers, we managed to reduce the MAE of all PAD coordinates to about 0.05!!**

Left Undone

Problems

- **[IDEA]** Due to the fact that the decoder uses a single value for decoding (really small latent space), that value is super sensitive to changes, in the sense that even if the regressed value was 0.64 and the real value was 0.69, the decoder generates wildly different LMA features. As such, unless our PAD-Latent space regression is super accurate, the generated LMA features will not depict the intended emotion.
    - This can possibly be alleviated by having a larger latent space (rather than just using a single value). This may help both the VAE's reconstruction accuracy and the whole PAD-Latent space regression. I'll be trying this approach during this following week

Notes

Thoughts

This week can be summarized by: "Tried a lot of things, saw a lot of things fail". On the VAE side, I was happy to be able to build a model that manages to +- accurately encode/decode our set of LMA features using a latent space of just 1. But this was only achieved by spending literally days of trial and error, building and changing the network's architecture, trying a lot of implementations, changing epochs, batch sizes, learning rates, optimizers and so on and so forth.

Then, when trying to go from PAD coordinates to the latent space value (which I was confident was going to work), I was met with really, really poor values.

On the regression forefront, going from PAD to LMA features directly, I kept trying a lot of things in order to improve the generation but didn't really get far. I had an idea I thought was pretty good of using only the emotional axis most correlated to each feature to generate that feature, but that didn't really work (may still look further into it).

On the flip side, managed to very slightly improve the LMA-PAD regression once again, by adding an extra step of outlier removing to our data preparation. So that was nice.

That was that for this week. I do have some news I wanted to share about the upcoming weeks:

- There is a very high possibility that one of my projects is going to get officially published to a scientific magazine. I'm currently working with the Realidade Virtual teacher - Daniel Simões Lopes - to publish the project me and my group created for the course.
- I wanted to double check with you, Some of my colleagues have told me that this year **there won't be any thesis defenses in July**, with the normal delivery having shifted to October and defenses being around November. I've checked the course page and found nothing there, so I was wondering if you have some insight into this?
- **I will be out of the country for about 1 week starting the 28th of April** to go participate in a GameJam in Denmark alongside other selected members of GameDevTecnico. All expenses are being paid by the association and I was one of only 10 selected from over 60 people in the organization. This and the fact that it's a great addition to my CV and a great networking event (with speeches from people from the industry) make this an unmissable opportunity for me. **It does mean that I won't be able to have a meeting on the 4th of May**

Work Hours

- Worked Thursday-Saturday from 1pm to 6-8pm
- Worked Monday-Thursday from 1pm to 6pm