UDACITY Logout

PROJECT

Build a Sign Language Recognizer

A part of the Artificial Intelligence Program

PROJECT REVIEW | CODE REVIEW 3 | NOTES

Meets Specifications

SHARE YOUR ACCOMPLISHMENT

All done! 🌬

Congratulations, and good luck with the rest of your nanodegree!

PART 1: Data

1. **Student provides correct alternate feature sets: delta, polar, normalized, and custom.**
2. **Student passes unit tests.**
3. **Student provides a reasonable explanation for what custom set was chosen and why (Q1).**

Interesting ideas for custom features! Using the area is an idea I hadn't seen before!

Here are some other thoughts I've collected over the months grading this project:

- Normalizing deltas doesn't work so well
- Rescaling x and y values works well
- Normalizing only the r in polar coordinates works well
- Distance between the hands is a good feature to add

In general, these selectors seem to prefer more features (cardinality, I mean). This could be because the current state isn't expressive enough. So you can concatenate features together and the performance will improve significantly. Just some thoughts 😉

PART 2: Model Selection

1. **Student correctly implements CV, BIC, and DIC model selection techniques in "my_model_selectors.py".**
2. **Student code runs error-free in notebook, passes unit tests and code review of the algorithms.**
3. **Student provides a brief but thoughtful comparison of the selectors (Q2).**

Your DIC is indeed, correct 😉

7 is quite a high number of CV. Many words don't have enough sequences to use that many splits. You are returning `n_constant` for those words, which means basically CV is not really optimised for those words. Your results are great, but will probably improve even more if you use a lower number of sequences where required.

PART 3: Recognizer

1. **Student implements a recognizer in "my_recognizer.py" which runs error-free in the notebook and passes all unit tests**
2. **Student provides three examples of feature/selector combinations in the submission cells of the notebook.**
3. **Student code provides the correct words within <60% WER for at least one of the three examples student provided.**
4. **Student provides a summary of results and speculates on how to improve the WER.**

Great WERs, and good explanation to go along with it!

DOWNLOAD PROJECT

3 CODE REVIEW COMMENTS ›

RETURN TO PATH

Rate this review

Student FAQ

Rate this review