Computational Psycholinguistics — Assignment 2

Daan Brugmans (S1080742)

I. Introduction

This report is the realization of the Assignment 2 project for the Radboud University course Computational Psycholinguistics. For this assignment, students must investigate whether the gradients computed from a recurrent neural network correlate with measured P600 component activity from a controlled experiment. The reasoning behind this assignment is that recent research ([1], [2]) has shown that the P600 component may be the backpropagation of prediction errors in the human language system. Since neural language models also backpropagate their prediction errors using gradients, there may exist similarities between the language error backpropagation of human and artificial neural language systems. This report contains the findings found by me for the Assignment 2 project.

The relevant code for this assignment can be found at the following URL: https://github.com/daanbrugmans/ru-computational-psycholinguistics-23-24/tree/main/assignment-2/code.

II. Related Work

III. Methodology

All the code I have written can be found in the Jupyter Notebook called `main.ipynb`, which should thusly contain all results also shown in this report. It can be found at the following URL: https://github.com/daanbrugmans/ru-computational-psycholinguistics-23-24/blob/main/assignment-2/code/main.ipynb I have placed the code in the `get_predictions.py` file in a function called `get_predictions`, so that I can easily call this code from the notebook.

My results are a collection of scatterplots that visualize the relationship between the surprisal and gradients of the models. Every plot also contains a quadratic function fitted to the data to show the trend of the relationship between the variables. I have chosen for a quadratic function as opposed to a linear function, as I found that this fits the data better. Included in every plot is also the Pearson Correlation Coefficient r between variables. Since surprisal values and gradients are calculated for multiple models, I provide a plot for every model. For every pair of variables, I also provide a plot of how the Pearson Correlation Coefficient changes as the model is trained on more data, which will be the main focus of my results.

Before producing results, I set a universal seed for Python itself, NumPy, and PyTorch of `3131`. I do this in order to improve the reproducibility of my results. This is why I also set some settings for CUDA regarding PyTorch's randomness when computing on GPU.

IV. Results

A. Comparing Surprisal vs Gradients

B. Comparing Conditions

V. Discussion

A. Correlations between Surprisal and Gradients

B. Correlations between Conditions

C. Correlations with ERP Components

VI. Conclusions

References

[1] Hartmut Fitz and Franklin Chang. Language erps reflect learning through prediction error propagation. *Cognitive Psychology*, 111:15–52, 2019.

[2] Stefan L Frank. Neural language model gradients predict event-related brain potentials, Jan 2024.

APPENDIX