## Output of check\_gradients

The loss derivative looks OK.
The gradient for word\_embedding\_weights looks OK.
The gradient for embed\_to\_hid\_weights looks OK.
The gradient for hid\_to\_output\_weights looks OK.
The gradient for hid\_bias looks OK.
wThe gradient for output\_bias looks OK.
loss\_derivative[2, 5] 0.001112231773782498
loss\_derivative[2, 121] -0.9991004720395987
loss\_derivative[5, 33] 0.0001903237803173703
loss\_derivative[5, 31] -0.7999757709589483

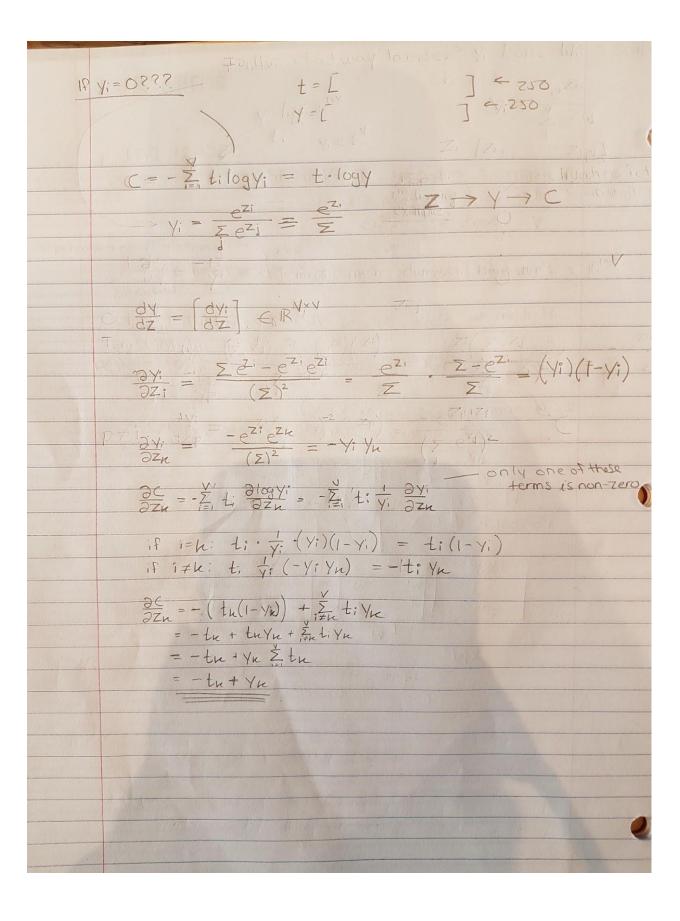
param\_gradient.word\_embedding\_weights[27, 2] -0.27199539981936866 param\_gradient.word\_embedding\_weights[43, 3] 0.8641722267354154 param\_gradient.word\_embedding\_weights[22, 4] -0.2546730202374648 param\_gradient.word\_embedding\_weights[2, 5] 0.0

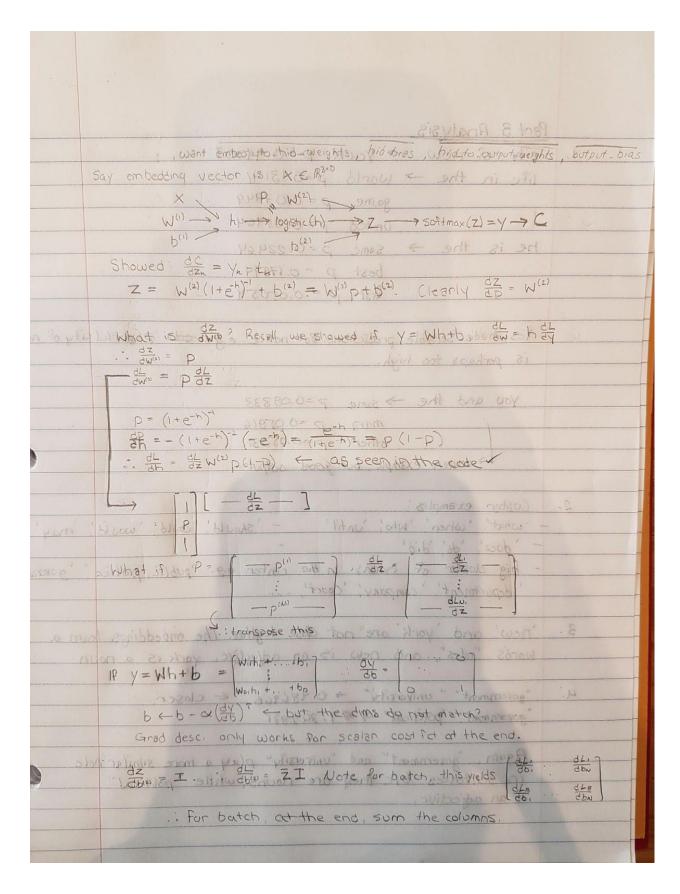
param\_gradient.embed\_to\_hid\_weights[10, 2] -0.6526990313918257 param\_gradient.embed\_to\_hid\_weights[15, 3] -0.13106433000472612 param\_gradient.embed\_to\_hid\_weights[30, 9] 0.11846774618169399 param\_gradient.embed\_to\_hid\_weights[35, 21] -0.10004526104604386

param\_gradient.hid\_bias[10] 0.2537663873815642 param\_gradient.hid\_bias[20] -0.03326739163635357

param\_gradient.output\_bias[0] -2.0627596032173052 param\_gradient.output\_bias[1] 0.0390200857392169 param\_gradient.output\_bias[2] -0.7561537928318482 param\_gradient.output\_bias[3] 0.21235172051123635

|         | 2019年05月12日   |
|---------|---|
|         |   |
|         |   |
|         | CSC421 DAI  |
|         | 1. Params in embedding: DXV madrix                                |
|         | D= word embedding size=16   |
|         | V= Vocab size = 250   |
|         | ## wnits in hidden layer  |
|         | Embed to hidden parameters: 3D × 16 + 16                          |
|         | Labiases  |
|         | Hidden to Oupot Params: 16 x250 +250                              |
|         |   |
| (000)   | Total: (250×16)+ (48×16+16)+ (16×25+250) = 5434                   |
|         |   |
|         | 2. 4-gram eg. p(x4/x5, x2, x1)                                    |
|         | 2504 entries :. 4×2504=1.5625×1010                                |
| 12077   |   |
| - North |   |
|         | W +> I  |
|         | For batch input $X = \begin{bmatrix} -x^{\alpha} - \end{bmatrix}$ |
|         |   |
|         |   |
|         |   |
|         | 1 W. Z.   |
|         | : XWT   |
|         |   |
|         |   |
|         |   |
| V       |   |
|         |   |
|         |   |
| 1       |   |
|         |   |
|         |   |





Part 3 Analysis 1. city of new - York P=0.99007 life in the > world p=0.13141 game p=0.07949 united p = 0.05964 he is the > some p=0.22464 best p = 0.17594 first p = 0.05054 Yes, made sensible predictions. But notice e.g. cits p(york/city of new) is perhaps too high. You and the -> same p=0.08333 man p=0.05816 other p=0.05031 Li unseen example, but good output. Custer examples: what', 'when', 'who', 'until' - 'should', 'could', 'would', 'may' 'does', 'do', 'did' - big cluster of nouns in the center. e.g. 'public, police' government! 'department', 'company', 'court', ... 'new' and 'york' are not close together. The embeddings learn a words "roles", and new is an adjective, york is a noun "government" " university" > 0.986966 & closer. "government", "political" > \$ 1.342831 Again, "government" and "university" play a more similar role in sentences b/c they are nouns, while "political" is an adjective.