

## Question 1.2

Provide a table that compares the 10-most cosine-similar words to the word 'dog', in order, alongside to the 10 closest words computed using euclidean distance. Give the same kind of table for the word 'computer.'

"dog"

Cosine similariy		Euclidean distance	
cat	0.922	cat	1.88
dogs	0.851	dogs	2.65
horse	0.791	puppy	3.15
puppy	0.775	rabbit	3.18
pet	0.772	pet	3.23
rabbit	0.772	horse	3.25
pig	0.749	pig	3.39
snake	0.740	pack	3.43
baby	0.740	cats	3.44
bite	0.739	bite	3.46

"computer"

Cosine similarity		Euclidean distance	
computers	0.917	computers	2.44
software	0.881	software	2.93
technology	0.853	technology	3.19
electronic	0.813	electronic	3.51
internet	0.806	computing	3.60
computing	0.803	devices	3.67
devices	0.802	hardware	3.68
digital	0.799	internet	3.69
applications	0.791	applications	3.69
pc	0.788	digital	3.70

Looking at the two lists, does one of the metrics (cosine similarity or Euclidean distance) seem to be better than the other? Explain your answer.

For the dog example Euclidean distance ranks puppy above horse whereas cosine similarity does the opposite. In terms of different words in the 2 groups, personally I think "pack, cats" are closer to "dog" than "snake, baby". For computer, the word "computing" which it shares same word root is ranked higher based on Euclidean distance. Overall, I would consider Euclidean distance to be the better metric.

### Question 1.3

Choose one of these relationships, but not one of the ones already shown in the starter notebook, and report which one you chose. Write and run code that will generate the second word given the first word. Generate 10 more examples of that same relationship from 10 other words, and comment on the quality of the results.

I chose nationality adjective.

finland	finnish	1.82
denmark	danish	1.73
iceland	icelandic	2.03
niger	niger	2.86
georgia	georgia	2.86
singapore	malaysian	2.76
ecuador	peruvian	2.34
peru	peruvian	1.32
philippines	philippine	2.30
south-africa	soldiery	2.24

I used the average of Mexico and Austria and their nationality adjectives, which couldn't be mistaken for the language noun since there isn't one, yet it still yields surprisingly bad results.

For finland, Denmark and Iceland the output is the language instead of the people. For Niger and Georgia the output is it self. For Singapore and Ecuador the output maps to their neighbor. For the Philippines the output is philippine which refers to the islands. South Africa is the worst as the output is simply wrong. However, there could be a bias in my country choices though.

### Question 1.4

Choose a context that you're aware of (different from those already in the notebook), and see if you can find evidence of a bias that is built into the word vectors. Report the evidence and the conclusion you make from the evidence.

I found that some vectors representing field of studies, when moved along the "intelligence" index, will produce another field of study as the closest word, which is counter intuitive

considering the relationship between the input-output pairs. This shows that there exists a bias on how the model perceives the "intelligence content" (whatever that means) of these fields.

Input	Add $1.5 * (\text{dumb} - \text{smart})$	Add $1.5 * (\text{smart} - \text{dumb})$
archeology	epigraphy	biomedical
anthropology	ethnology	sciences
psychiatry	psychosomatic	clinical

### Question 1.5

How does the euclidean difference change between the various words in the notebook when switching from  $d=50$  to  $d=300$ ? How does the cosine similarity change? Does the ordering of nearness change? Is it clear that the larger size vectors give better results - why or why not?

The Euclidean distance went up, and cosine similarity is reduced. The ordering of nearness also changed. From my observations larger sized vectors doesn't necessarily mean better results, this might be due to less significant dimensions having the same weight as the more significant ones, since both metrics give the same weight for all dimensions.

### Question 1.6

Modify the notebook to use the FastText embeddings. State any changes that you see in the Bias section of the notebook.

There seems to be less bias using fasttext embeddings. However it also seems like fasttext includes some made up words.

### Question 2.2

Compute the similarity (using both methods (a) and (b) above) for each of these words: "greenhouse", "sky", "grass", "azure", "scissors", "microphone", "president" and present them in a table. Do the results for each method make sense? Why or why not? What is the apparent difference between method 1 and 2?

word	avg of cos similarity	cos similarity with avg
greenhouse	0.183	0.202
sky	0.602	0.670

grass	0.506	0.558
azure	0.408	0.456
scissors	0.289	0.320
microphone	0.308	0.343
president	0.299	0.329

The results makes sense since the words that are closely tied with one particular color has a higher similarity. Apparently, the avg of similarities from method 1 is smaller than the similarity given by method 2.

### Question 2.3

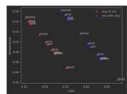
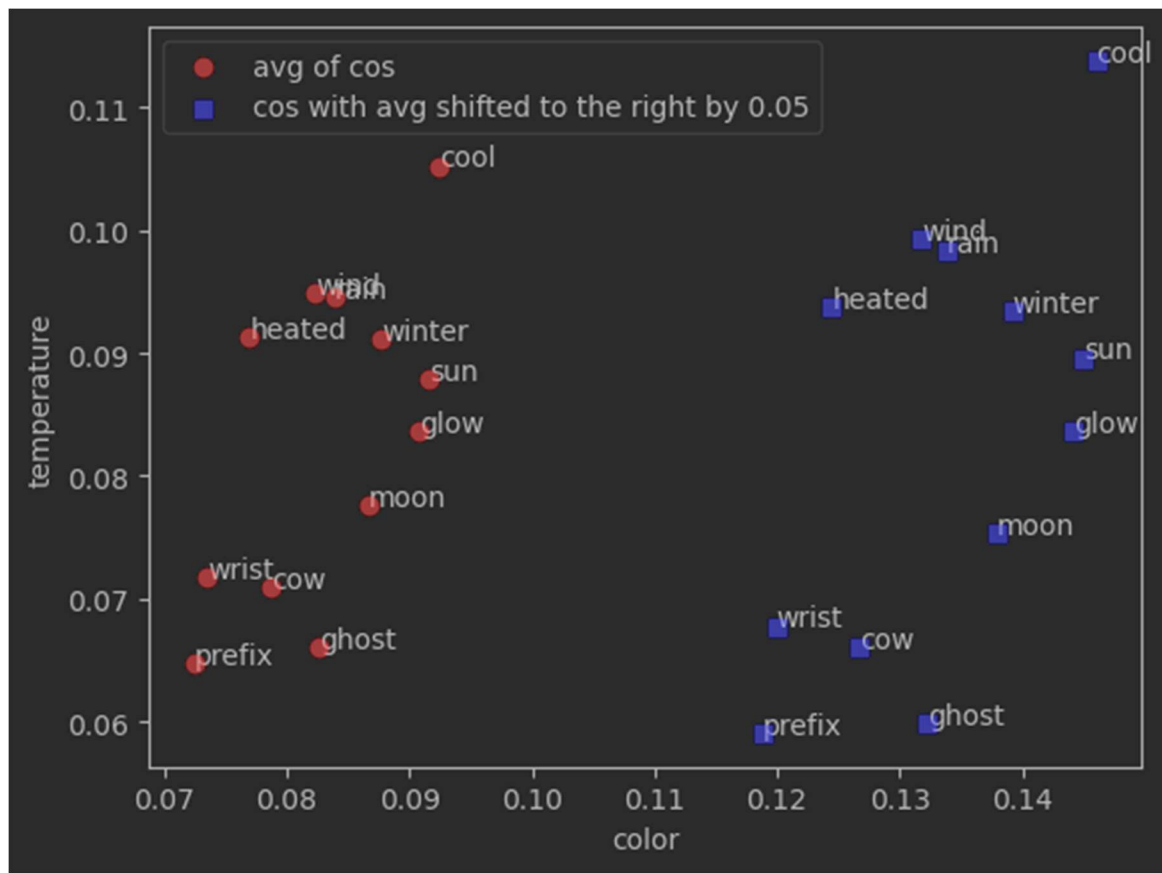
Create a similar table for the meaning category temperature by defining your own set of category words, and test a set of 10 words that illustrate how well your category works as a way to determine how much temperature is "in" the words. You should explore different choices and try to make this succeed as much as possible. Comment on how well your approach worked.

word	avg of cos similarity	cos similarity with avg
molten	0.372	0.496
greenhouse	0.305	0.412
shower	0.430	0.573
arctic	0.341	0.458
republic	0.151	0.202
placebo	0.214	0.298
tectonics	0.144	0.197
resistor	0.020	0.034
locomotive	0.076	0.100
zebra	-0.012	-0.025

Overall my set of words almost has similar performance compared to question 2.2, however it wasn't able to detect some words that are half-way between "having" and "not having" "temperature" very well, such as "placebo" and "locomotive".

### Question 2.4

Plot each of the words in two dimensions (one for colour and one for temperature) using matplotlib. Do the words that are similar end up being plotted close together? Why or why not?



They did because words being similar implies that they would have similar "color" and "temperature" content, which gives them a similar vector and thus a similar coordination on the plot.

### Question 3.1

Find three pairs of words that this corpora implies have similar or related meanings.

hold-rub

a-the

cat-dog

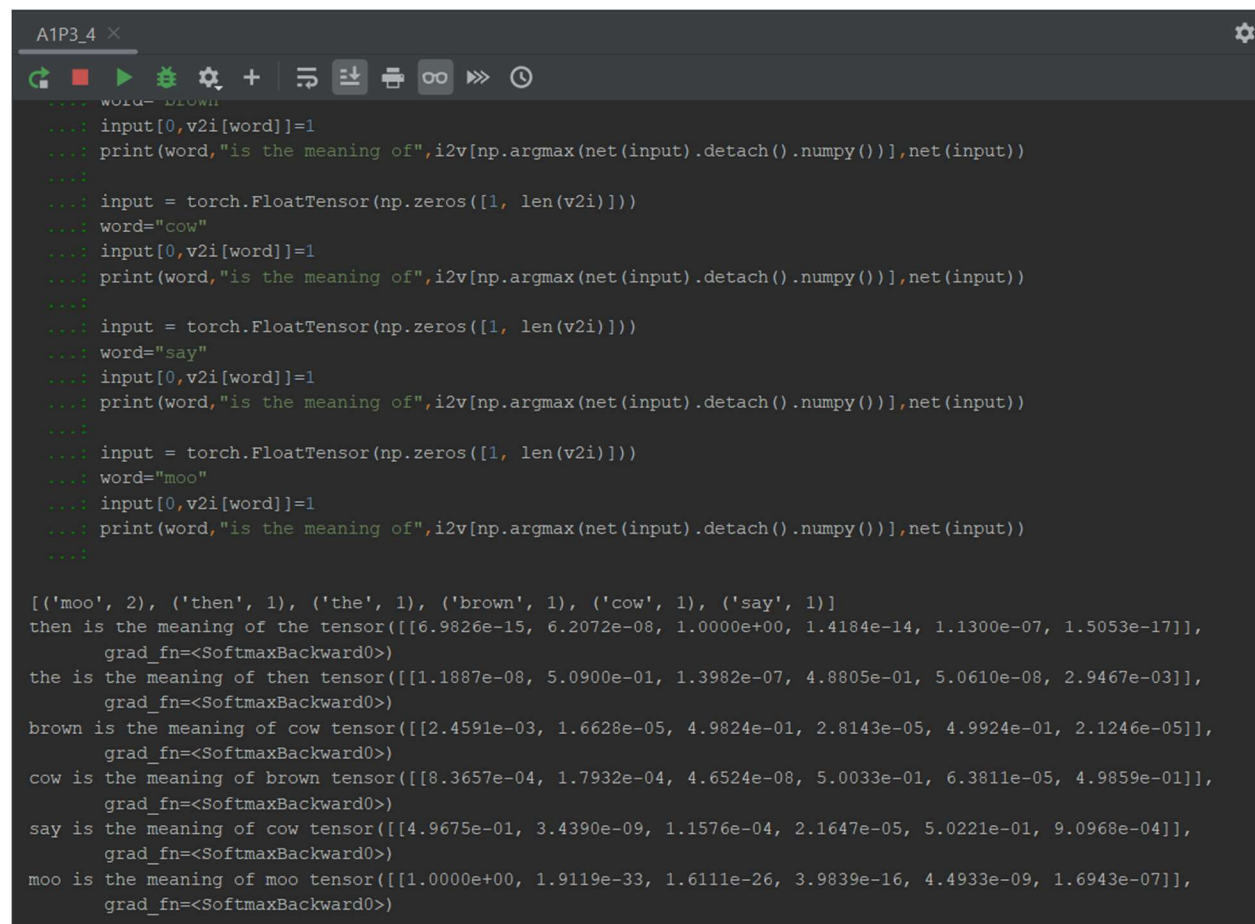
### Question 3.2

Which is the most frequent word in the corpus, and the least frequent word? What purpose do the v2i and i2v functions serve?

The most frequent word is "and", and the least frequent word is "I". The purpose of the v2i and i2v functions are to create forward and backward mapping between words and numbers.

### Question 3.3

Test that your function works, and show with examples of output (submitted) that it does.



```
A1P3.4 x [icon]
[icon] [icon] [icon] [icon] [icon] [icon] [icon] [icon] [icon] [icon]
... word="brown"
... input[0,v2i[word]]=1
... print(word,"is the meaning of",i2v[np.argmax(net(input).detach().numpy())],net(input))
...
... input = torch.FloatTensor(np.zeros([1, len(v2i)]))
... word="cow"
... input[0,v2i[word]]=1
... print(word,"is the meaning of",i2v[np.argmax(net(input).detach().numpy())],net(input))
...
... input = torch.FloatTensor(np.zeros([1, len(v2i)]))
... word="say"
... input[0,v2i[word]]=1
... print(word,"is the meaning of",i2v[np.argmax(net(input).detach().numpy())],net(input))
...
... input = torch.FloatTensor(np.zeros([1, len(v2i)]))
... word="moo"
... input[0,v2i[word]]=1
... print(word,"is the meaning of",i2v[np.argmax(net(input).detach().numpy())],net(input))
...

[('moo', 2), ('then', 1), ('the', 1), ('brown', 1), ('cow', 1), ('say', 1)]
then is the meaning of the tensor([[6.9826e-15, 6.2072e-08, 1.0000e+00, 1.4184e-14, 1.1300e-07, 1.5053e-17]],
  grad_fn=<SoftmaxBackward0>)
the is the meaning of then tensor([[1.1887e-08, 5.0900e-01, 1.3982e-07, 4.8805e-01, 5.0610e-08, 2.9467e-03]],
  grad_fn=<SoftmaxBackward0>)
brown is the meaning of cow tensor([[2.4591e-03, 1.6628e-05, 4.9824e-01, 2.8143e-05, 4.9924e-01, 2.1246e-05]],
  grad_fn=<SoftmaxBackward0>)
cow is the meaning of brown tensor([[8.3657e-04, 1.7932e-04, 4.6524e-08, 5.0033e-01, 6.3811e-05, 4.9859e-01]],
  grad_fn=<SoftmaxBackward0>)
say is the meaning of cow tensor([[4.9675e-01, 3.4390e-09, 1.1576e-04, 2.1647e-05, 5.0221e-01, 9.0968e-04]],
  grad_fn=<SoftmaxBackward0>)
moo is the meaning of moo tensor([[1.0000e+00, 1.9119e-33, 1.6111e-26, 3.9839e-16, 4.4933e-09, 1.6943e-07]],
  grad_fn=<SoftmaxBackward0>)
```

### Question 3.4

What is the total number of parameters in this model with an embedding size of 2 - counting all the weights and biases?

I added 2 hidden layers before and after the embedding layer with 5 nodes each. The total number of free parameters is therefore  $12*5+6*2+3*5+6*11=153$ .

### Question 3.5

Find a suitable learning rate, and report what that is. Show the training and validation curves (loss vs. Epoch), and comment on the apparent success (or lack thereof) that these curves suggest.

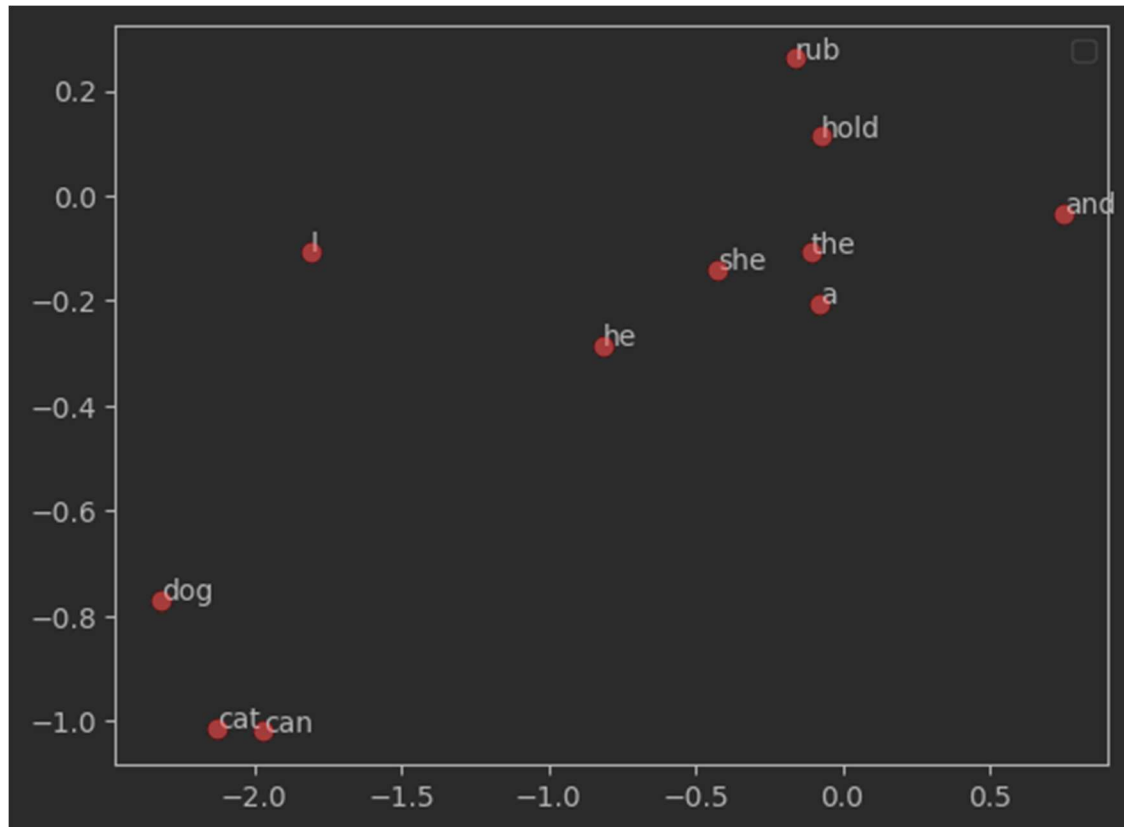


I used leaky ReLU with 0.3 and a learning rate of  $6e-3$ . I was able to get validation loss, showing the the network is indeed working.

### Question 3.6

Do the results make sense, and confirm your choices from part 1 of this Section? What would happen when the window size is too

large? At what value would window become too large for this corpus?



The results make sense. If window size is too large the model will be less able to differentiate between a word and it's neighbors. For this corpus where sentences can be as short as 4 words, a window of 7 will already be too large.

#### Question 4.1

Give a 3 sentence summary of the subject of the document.

The document talked about National Mint for the United States, as well as other nations, and includes some figures to back up its claims.

#### Question 4.2

What are the functional differences between this code and that of the same function in Section 3?



This code can deal with more punctuation and special characters instead of space and period only. It also labels some infrequent words as OOV so that time is not wasted on training these.

#### Question 4.3

Determine the number of words in the text, and the size of the filtered vocabulary, and the most frequent 20 words in the filtered vocabulary, and report those. Of those top 20 most frequent words, which one(s) are unique to the subject of this particular text?

There are 62255 words in the text, and the size of the filtered vocab is 2568. The most frequent 20 words in the filtered vocabulary are 0: 'the', 1: 'of', 2: 'be', 3: 'and', 4: 'in', 5: 'to', 6: 'a', 7: 'for', 8: 'as', 9: 'by', 10: 'he', 11: 'with', 12: 'coin', 13: 'this', 14: 'on', 15: 'his', 16: 'which', 17: 'at', 18: 'it', 19: 'from'. All of these word, with the exception of "coin", are also high-frequency words in any ordinary English text.

#### Question 4.4

How many total examples were created?

I created 498028 samples.

#### Question 4.5

State how many examples remain for the corpus using this reduction.

114120 samples remain.

#### Question 4.7

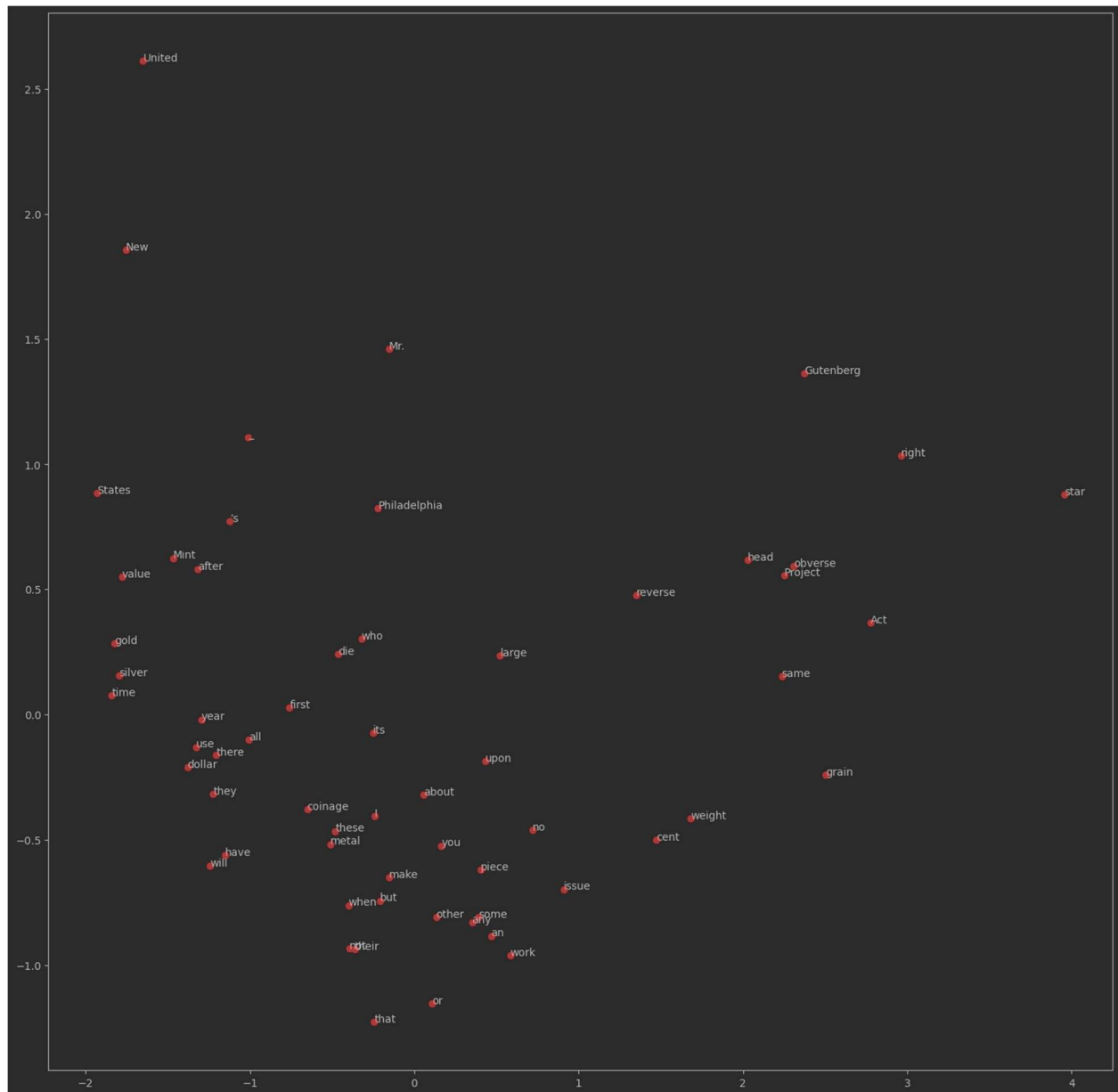
Show the training and validation curves vs. Epoch, and comment on the apparent success (or lack thereof) that these curves suggest.



I used MSELoss and a training rate of  $1e-4$ . Judging from the curves it could be said that the training is descending in the correct directions, and could be improved if given more epochs to train.

#### Question 4.8

Comment on how well the embeddings worked, finding two examples each of embeddings that appear correctly placed in the plot, and two examples where they are not.



I don't think an embedding of size 2 can represent the words very well. From the graph we can see that "gold" and "silver" are placed together, as well as "observe" and "project". However the words "dollar" and "cent" are only similar in 1 of the 2 dimensions. "United" and "States" separates by a lot also.