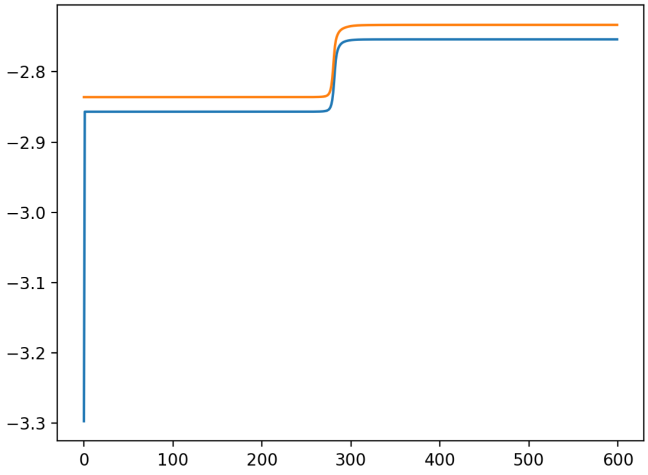
Part 1:

1. HMM is sensitive to different initialization. Different initializations will affect the model's performance.  Here is the comparison:

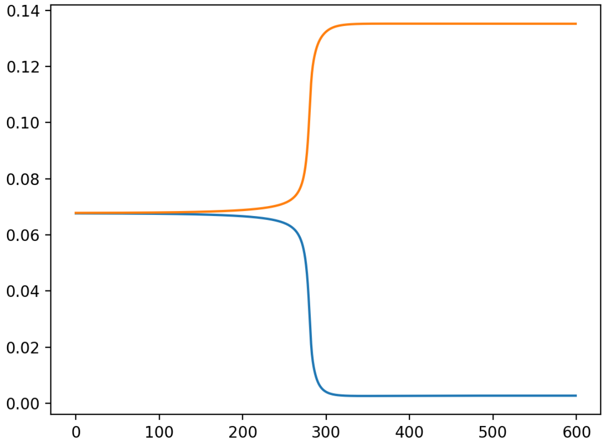
| Initialization Setting | Likelihood_training | likelihood_testing |
|---|---|---|
| Uniform initialization | -2.8567940501278706 | -2.8360197676670067 |
| Slightly different from uniform initialization | -2.7538860454958565 | -2.7332487699330454 |

   As you might notice, we can conclude that when we set the emission and transition matrix to have slightly different entries, the model actually performs **better than** the model which adopts uniform initialization.

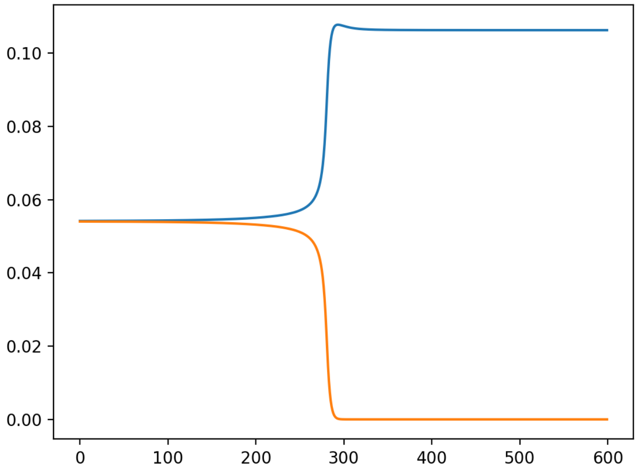
2. Here is the plot of the average log likelihood of the training data and the test data. Notice the blue line corresponds to the average log likelihood of training data, whereas the orange one denotes that of the test data.



3. **Blue is for state 0 and orange is for state 1.**
   Here is the emission probability of state 0 generating the letter A and probability of state 1 generating letter A:  We can observe that for state 0, it is less and less likely to generate letter A as time elapses, whereas the state 1 is more and more much likely to generate A as time elapses.



Here is the emission probability of generating letter N. We see the trend is completely opposite from what we can see in the previous graph. State 0 is more probable of generating letter N than state 1 does as time elapses.



4. Here are the emission matrix at 600 iteration:

```
[[2.68768450e-003 2.44182603e-002 5.61829867e-002 7.05707363e-002
  6.38854994e-044 3.57436197e-002 2.86734532e-002 7.20028036e-002
  4.12316100e-076 5.36808940e-003 8.10094565e-003 7.67898524e-002
  3.77075548e-002 1.06314356e-001 4.61227171e-037 3.79039483e-002
  1.63661262e-003 1.04743208e-001 1.10307691e-001 1.45166275e-001
  1.88169520e-052 1.58424102e-002 2.59894085e-002 5.49901841e-003
  2.74296276e-002 9.16503069e-004 4.95509039e-006]
 [1.35212612e-001 1.92675168e-101 1.20845823e-004 2.07415485e-046
  2.13928490e-001 8.90484358e-155 6.94845502e-013 1.16299154e-003
  1.17966282e-001 4.63220485e-188 7.64327882e-004 1.22931953e-008
  2.55509778e-092 2.64058083e-068 1.32499836e-001 5.84820572e-020
  9.75374805e-188 2.70944916e-071 2.84029053e-046 8.11701176e-003
  4.63172159e-002 1.04400395e-175 3.20573787e-133 2.87980407e-117
  1.07170189e-018 4.75734153e-201 3.43910375e-001]]
```

Here is the transition matrix:

```
[[0.28622091 0.71377909]
 [0.74048455 0.25951545]]
```

Observation:

Just by observing the emission probability and transition probabilities. We could found several interesting things:

1) Different states have their own preferences generating the observation. We can decode these preferences by just comparing the emission values. Some states are more likely to generate specific letters than other states.
2) The updated transition and emission matrix is far different from what we gave to the initial value .
3) For state 0, its emission distribution is more stable than that of state 1: It only has few small emission probabilities as opposed to state 1. For state 2, many of its emission values are vanishingly small, which means once the HMM model enters state 2, it is very very unlikely to generate these observations.

In conclusion, our HMM learns the underlying composition of the words and grammatical structure of the sentences by learning to judge what letter should appear next after seeing the current one.

Part II

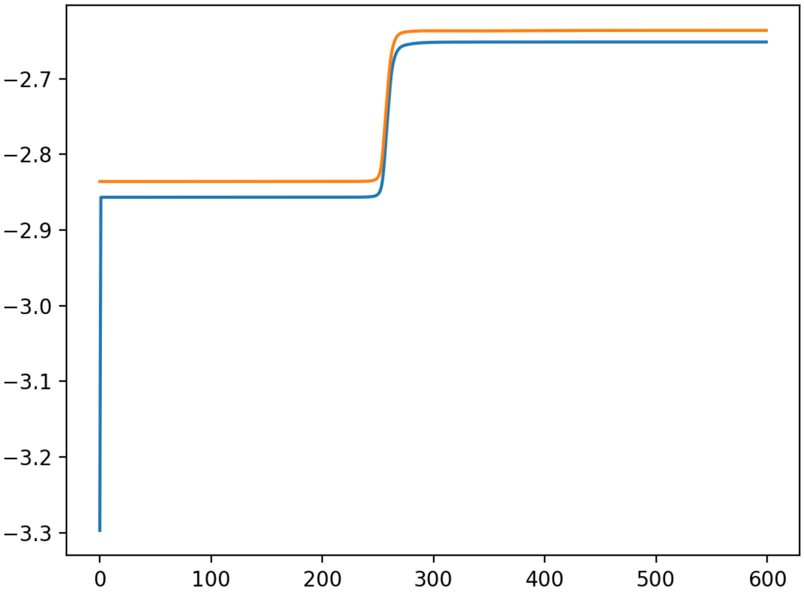
a)

We can use the same approach to make comparisons. Again, we could see that an initialization method which initializes the probabilities with different values, performs much much better than uniform initialization does since the HMM returns higher data likelihood. Also as you would see from the results in 2b), we can see the likelihood jumps significantly during the training when we are not using uniform initialization.

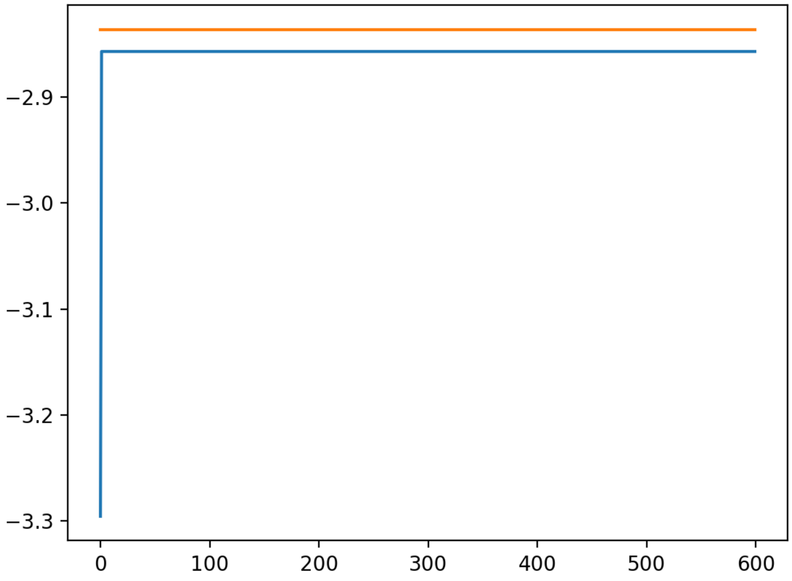
| Initialization Setting | Likelihood_training | likelihood_testing |
|---|---|---|
| Uniform initialization | -2.8567940501278706 | -2.8360197676670067 |
| Slightly different from uniform initialization | -2.65168815011314 | -2.636356392140911 |

b) The orange line is the average log likelihood for testing, and the blue line is the average log likelihood for training. **Again, uniform initialization is really bad here.**

Un-uniform initialization:

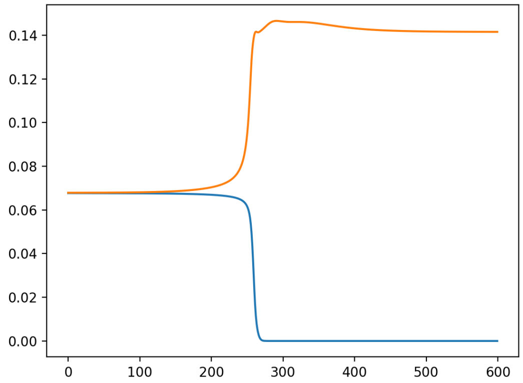


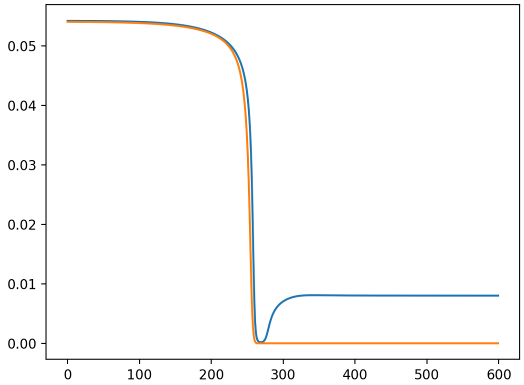
Uniform distribution:



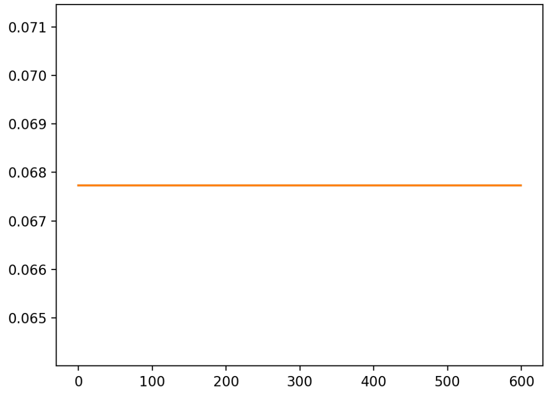
c)
Un-uniform initialization:

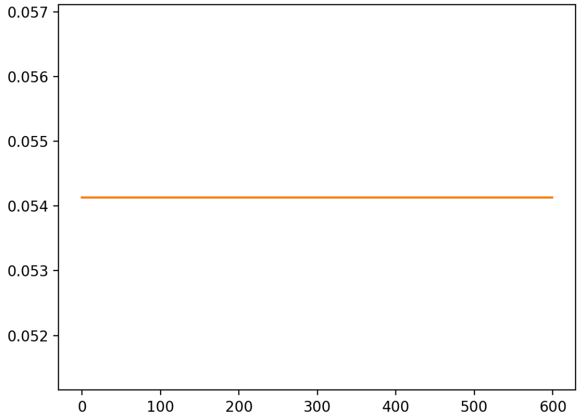
As you might notice in the second picture(exhibits a different pattern that we never saw in the previous part), which is the emission probabilities of N for state 0 and state 1. Their emission probabilities are roughly the same from the beginning to the middle of the training loop, but state 0 is more probable of emitting N than state 1 is.





Uniform initialization:  Very bad. Not changing at all.





d)

For **uniform initialization**( the transition matrix **is not updating** at all ):

Emission (the emission probabilities are the same for each state, which is bad):

```
[[0.06773333 0.01243333 0.02866667 0.03593333 0.105      0.0182
  0.0146     0.03723333 0.0579     0.00273333 0.0045     0.0391
  0.0192     0.05413333 0.06503333 0.0193     0.00083333 0.05333333
  0.05616667 0.0779     0.02273333 0.00806667 0.01323333 0.0028
  0.01396667 0.00046667 0.1688    ]
 [0.06773333 0.01243333 0.02866667 0.03593333 0.105      0.0182
  0.0146     0.03723333 0.0579     0.00273333 0.0045     0.0391
  0.0192     0.05413333 0.06503333 0.0193     0.00083333 0.05333333
  0.05616667 0.0779     0.02273333 0.00806667 0.01323333 0.0028
  0.01396667 0.00046667 0.1688    ]
 [0.06773333 0.01243333 0.02866667 0.03593333 0.105      0.0182
  0.0146     0.03723333 0.0579     0.00273333 0.0045     0.0391
  0.0192     0.05413333 0.06503333 0.0193     0.00083333 0.05333333
  0.05616667 0.0779     0.02273333 0.00806667 0.01323333 0.0028
  0.01396667 0.00046667 0.1688    ]
 [0.06773333 0.01243333 0.02866667 0.03593333 0.105      0.0182
  0.0146     0.03723333 0.0579     0.00273333 0.0045     0.0391
  0.0192     0.05413333 0.06503333 0.0193     0.00083333 0.05333333
  0.05616667 0.0779     0.02273333 0.00806667 0.01323333 0.0028
  0.01396667 0.00046667 0.1688    ]]
```

Transition; (**bad, not updating at all**)

```
[[0.25 0.25 0.25 0.25]
 [0.25 0.25 0.25 0.25]
 [0.25 0.25 0.25 0.25]
 [0.25 0.25 0.25 0.25]]
```

Un-uniform initialization:

Emission matrix(**tells us the probability of generating some letter once we are at certain state**):

```
[[1.34833110e-100 1.86053228e-003 2.85485400e-016 6.26218041e-003
  1.11546964e-002 6.80444640e-003 3.34402058e-003 3.39758398e-001
  2.55506422e-002 1.93113891e-003 2.35948970e-003 7.44109370e-002
  2.80680596e-002 8.00920717e-003 1.95969250e-021 3.41140613e-003
  3.73465150e-043 1.08419063e-001 9.60158288e-002 7.48306638e-002
  7.71265611e-003 1.41421604e-003 6.73858979e-003 1.78364874e-230
  7.59013807e-002 4.06398639e-004 1.15636049e-001]
 [1.41517008e-001 6.73487748e-157 2.18205832e-032 3.12577050e-109
  2.78098977e-001 3.41400244e-168 4.01605241e-173 5.36192021e-037
  1.24058945e-001 4.46833877e-132 1.35327057e-004 6.08657546e-107
  1.64453141e-202 4.77017965e-134 1.38401626e-001 1.60888075e-077
  2.41035257e-256 5.43508708e-166 1.13544204e-015 3.22046578e-007
  3.40825472e-002 8.53182872e-284 6.25638111e-106 0.00000000e+000
  2.37560171e-003 1.95969320e-214 2.81329646e-001]
 [8.35718157e-063 3.41358047e-002 7.50348742e-002 9.72438818e-002
  2.37206267e-070 4.81511624e-002 4.05013339e-002 4.13161364e-053
  2.47148522e-143 7.22803384e-003 1.20125817e-002 5.61227567e-002
  3.90197970e-002 5.92134987e-002 2.00882417e-229 4.87425407e-002
  2.38860740e-003 7.07914126e-002 9.56113545e-002 1.99781292e-001
  1.51466051e-171 2.26774941e-002 3.58143979e-002 4.43501604e-003
  1.29865172e-002 1.20996451e-003 3.68976784e-002]
 [8.86385893e-002 1.90180532e-003 1.47810294e-002 7.84476598e-003
  2.50187387e-019 3.89282345e-003 6.14726941e-004 2.36248037e-059
  5.22976764e-002 5.83913651e-011 1.20759044e-099 6.75044326e-002
  1.49134759e-002 1.93609617e-001 7.95070684e-002 1.14092744e-002
  0.00000000e+000 9.95113652e-002 7.29824983e-002 6.58920461e-013
  5.44623079e-002 0.00000000e+000 8.93626842e-052 7.44044234e-003
  1.37556333e-003 0.00000000e+000 2.27312537e-001]]
```

Transition matrix:

**If you are at state 1, then it is unlikely that you will enter state 4.**
**If you are at state 3, then it is unlikely that you will enter state 4.**
**If you are at state 4, then it is unlikely that you will enter state 1 and state 2.**

```
[[1.27137257e-02 9.61630364e-01 2.56559101e-02 5.66517091e-63]
 [3.27664991e-02 4.60243316e-02 5.47918787e-01 3.73290382e-01]
 [2.75078483e-01 7.18322658e-01 6.59885912e-03 6.30947938e-46]
 [5.12214405e-38 1.73255854e-22 8.27361626e-01 1.72638374e-01]]
```

Part III:

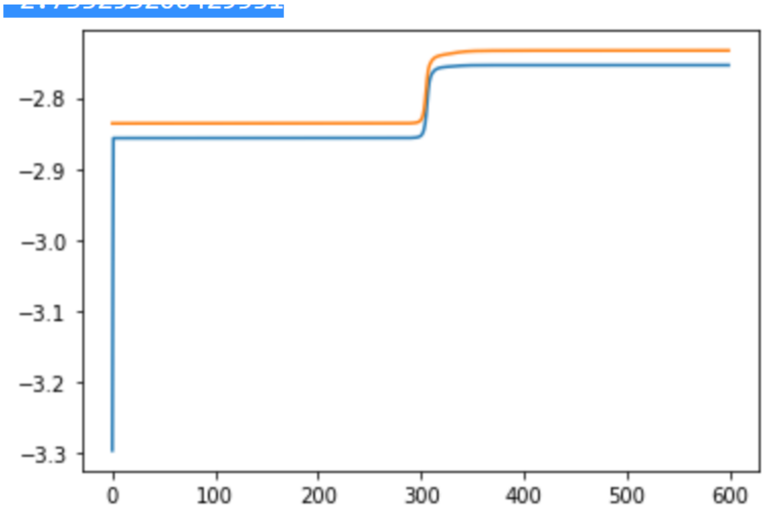
a) Relative frequency in training corpus:

```
[0.06773333, 0.01243333, 0.02866667, 0.03593333, 0.105     ,
       0.0182    , 0.0146    , 0.03723333, 0.0579    , 0.00273333,
       0.0045    , 0.0391    , 0.0192    , 0.05413333, 0.06503333,
       0.0193    , 0.00083333, 0.05333333, 0.05616667, 0.0779    ,
       0.02273333, 0.00806667, 0.01323333, 0.0028    , 0.01396667,
       0.00046667, 0.1688    ]
```

b) Zero_mean random vector that I have

```
array([[ 0.14107157,  0.54326375, -0.23451668, -0.16674527, -0.390442  ,
        -0.16912168,  0.05401647,  0.12671911,  0.51535747,  0.2319119 ,
        -0.17835103, -0.36968423, -0.28112477,  0.17979391,  0.3315305 ,
         0.34907856, -0.29932801, -0.04960838, -0.09141588, -0.37717775,
         0.11859362, -0.17305512,  0.04569592, -0.28255877, -0.13886274,
         0.04059285,  0.52436666]])
```

c) I found the jumping time (drastic increase in likelihood) postones several iterations if we initialize the probabilities matrix this way. Performance-wice, the log likelihood is slightly worse than the original initialization method.
**training_likelihood**:-2.753886014422116
**test_likelihood**: -2.733253206429531


−2.8
−2.9
−3.0
−3.1
−3.2
−3.3
0
100
200
300
400
500
600