1. **Experiment**: Run a simple Artificial Neural Network (ANN) model 10 times using different random seeds. Use seeds ranging from 1 to 10 for these runs

   Question:

   1.1 Result Analysis: Record the output of each trial. Present this data in a tabular format, summarizing the convergence value for each seed. (1pt)

   Answer:

seeds/epoch	0	50	100	150	200	250	300
1	2.337935	0.354601	0.407567	0.393525	0.210256	0.379416	0.209338
2	2.305132	0.481161	0.220649	0.214474	0.214594	0.208037	0.209025
3	2.305132	0.481161	0.220649	0.214474	0.214594	0.208037	0.212019
4	2.293585	0.368123	0.421418	0.403463	0.401345	0.417959	0.210538
5	2.359769	0.418165	0.499559	0.217204	0.215842	0.210331	0.392910
6	2.310882	0.373581	0.227843	0.214972	0.210251	0.209413	0.208573
7	2.279085	0.340240	0.234737	0.415527	0.214810	0.214058	0.208971
8	2.310026	0.366771	0.222349	0.220332	0.382671	0.429902	0.212254
9	2.314622	0.302922	0.218075	0.213723	0.210246	0.449849	0.214507
10	2.307677	0.615157	0.227124	0.216396	0.210159	0.210810	0.214379

   1.2 Convergence Assessment: Analyze the convergence of the model across the different trials. Do the trials converge to similar values? Describe any patterns or variations you observe. (1pt)

seeds/delta	Δ_1	Δ_2	Δ_3	Δ_4	Δ_5	Δ_6
1	-1.98333	0.05297	-0.01404	-0.18327	0.16916	-1.7008
2	-1.82397e+00	-2.60512e-01	-6.17474e-03	1.20007e-04	-6.55782e-03	9.88472e-04
3	-1.92546	0.05329	-0.01795	-0.00212	0.01661	-0.20594
4	-1.86122e+00	-2.07326e-01	-8.09717e-03	2.36416e-01	-2.39698e-01	1.15305e-03
5	-1.94160e+00	8.13943e-02	-2.82355e-01	-1.36128e-03	-5.51175e-03	1.82579e-01
6	-1.93730e+00	-1.45739e-01	-1.28702e-02	-4.72132e-03	-8.38023e-04	-8.39703e-04
7	-1.93884e+00	-1.05503e-01	1.80790e-01	-2.00717e-01	-7.52437e-04	-5.08620e-03
8	-1.94326	-0.14442	-0.00202	0.16234	0.04723	-0.21765
9	-2.0117	-0.08485	-0.00435	-0.00348	0.2396	-0.23534
10	-1.69252e+00	-3.88033e-01	-1.07279e-02	-6.23736e-03	6.51025e-04	3.56926e-03

   a) **Stricly decreasing loss** (seed 6) this is rare in the pattern where each 50 epoch the loss are stricly decreasing
   b) **Converge to 0.21** (all seed except seed 5) all seed on the pattern are decreasing to some number i.e 0.21
   c) **Seed 5 have jumping loss at the end**

   1.3 Statistical Summary: Calculate and discuss the mean and standard deviation of the convergence values. What does this statistical summary tell you about the stability and reliability of the ANN model under different initial conditions? (1pt)

   Answer

```
converges_value_epoch 300 = [0.209338, 0.209025, 0.212019, 0.210538, 0.392910,
                             0.208573, 0.208971, 0.212254, 0.214507, 0.214379]
sample_mean = 0.22925139999999997
std = 0.05459241992291604
KV = std/sample_mean = 23.813342000492057 %
```

   the statistical tells us the ANN model tends to a certain value on neighborhood of average, the standard deviation iq quitely small with KV = 23%. the stability and reliability is determine by the initial condition, but there is one value i.e 0.382910 which is an outliers, so sometimes the initial condition can lead some loss to be far from the average

1.4 Discussion of Results:

a) Why Variation Occurs: Based on your observations, why do you think there are differences in the results from trial to trial? How does the choice of seed affect the outcomes? (1pt)

Answer

different seed meaning that we have different initialization random weight, it can lead to diverse convergence paths and final outcomes. the randomness is contributes with variation. The outcomes by determining the initial state of the random number generator used during weight initialization. Using the same seed ensures reproducibility, as it initializes the training process in a consistent manner over the trials.

b) Impact of Initialization: Discuss how initialization can impact the learning and performance of neural networks. Refer to specific instances from your experiment results to support your discussion. (1pt)

Answer

the initialization can impact to convergence speed, we can see from the example, for every seeds have different convergenve epoch, for seed 1 it can be done at 200 epoch, but on seed 2 it can be done at 100 epoch refer to the first table.

2. **Experiment**: Train two types of Recurrent Neural Network (RNN) models—one with a single layer and the other with two layers. For each model, run experiments with varying numbers of training epochs (5, 50, 500) and different context sizes (1, 2, 3).

Question:

2.1 Convergence Analysis: At which epoch numbers do the single-layer and two-layer RNNs begin to converge for each context size? Record the earliest epoch at which each configuration appears to stabilize. (1pt)

Answer:

for this example we set the thresshold for loss is 0.22, so whenever the loss achieve loss lower than 0.22 we take that epoch, and write on the table

context size / layer	one layer	two layer
1	379	344
2	354	319
3	385	395

2.2 Performance Comparison: this time table below are comparing to computed for epoch 500

context size / layer	one layer	two layer
1	6.97	9.4
2	7.5	10.7
3	7.65	11.4

a) Which model configuration (layer count and context size) achieves the best performance at the shortest training time? (1pt)

Answer: from the table we can conclude the shortest time training are for 1 context size and one layer

b) How does increasing the context size affect the performance and convergence of each model? (1pt)

Answer more context size and more layer on model it lead more complex model which is, affect the performance and the convergence of our model

it can seen, if we increase the context size so there will more loop and more possible input, that's why it will be longer.

also, if we increase the layer it leads us to do more computation. more computation mean more time.

2.3 Result Summary: Summarize the final accuracy or loss values for each configuration at the different epochs. How do these results vary between the single-layer and two layer models? (2pts)

for epoch 5,50,500 the loss are given by:

context size/layer	1-layer	2-layer
1	2.14328	2.19803
	0.96867	0.84945
	0.21449	0.21321
2	2.10417	2.18303
	0.73292	0.64828
	0.21363	0.21242
3	2.15295	2.20717
	0.94749	0.71965
	0.21578	0.21740

commonly the loss result with same epoch and context size, the 2-layer model are gives more lower loss at the end, it could because more complex usually brings lower loss, since we have more parameters here (more layer more parameters) and for all convergence value is always less than 0.22.