

1. Predict 2022 citations based on all the 2017-2021 citations.

Please refer to the last column (predicts).

	univ_rank	first_initial	last_initial	cit_2017	cit_2018	cit_2019	cit_2020	cit_2021	cit_2022	h_index	i_10_index	predicts
0	51	I	P	38	102	159	245	277	381	16	24	326.906952
1	51	S	M	153	333	510	749	963	1048	31	59	1145.870972
2	51	A	B	5524	8950	12526	14204	16734	17508	60	88	17913.009766
3	51	W	H	161	183	206	215	179	262	22	33	193.432526
4	51	F	N	70	96	88	133	157	156	16	24	170.850189
5	51	M	I	238	386	641	602	1025	1249	41	110	1199.789185
6	51	R	F	41	115	210	312	473	554	15	21	592.749207
7	51	S	J	54	72	113	139	144	141	9	9	159.353165
8	51	S	Z	135	92	160	184	238	332	24	34	249.772537
9	51	J	Z	1678	2066	2635	3253	4319	4125	23	30	4723.891113
10	52	E	G	151	147	156	152	169	167	31	85	172.762985
11	52	M	C	85	121	202	264	376	383	28	65	443.748505
12	52	W	E	1375	1264	1038	998	947	784	37	65	935.400757
13	52	A	D	183	286	356	395	449	490	22	31	474.249207
14	52	R	C	89	128	103	109	108	103	20	39	113.343437
15	52	V	C	19	22	52	116	172	188	13	21	220.804123
16	52	T	B	503	463	584	722	945	893	46	163	985.492126
17	52	W	A	47	82	98	128	178	346	17	35	208.224319
18	52	K	A	139	125	84	80	74	47	16	24	73.160461
19	52	S	H	205	201	220	210	202	187	21	29	208.292938

2. Try a 1-hidden layer neural network (5-3-1 architecture) using the backpropagation algorithm.

Begin by making design decisions: node functions, data normalization, output interpretation, optimizer choice, etc.

```
activation='relu'  
loss='mean_absolute_error'  
optimizer=keras.optimizers.Adam(lr)
```

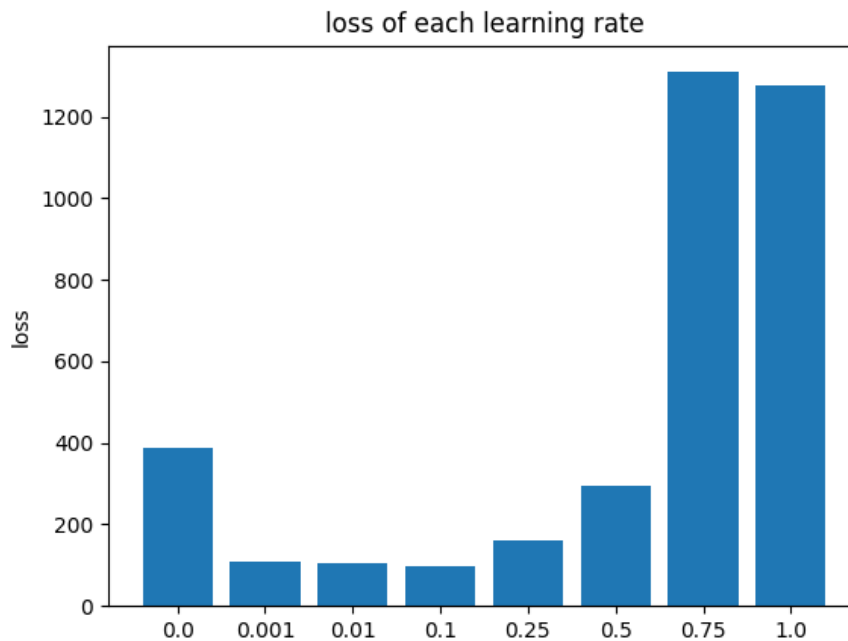
Model: "sequential"

Layer (type)	Output Shape	Param #
hidden (Dense)	(None, 3)	18
output (Dense)	(None, 1)	4

```
=====  
Total params: 22 (88.00 Byte)  
Trainable params: 22 (88.00 Byte)  
Non-trainable params: 0 (0.00 Byte)
```

3. Play with different values of the learning rate to see what works best for this problem.

The optimal learning rate for minimizing loss varies between different experiments, but typically, the loss is minimized when the learning rate falls within the range of 0.001 to 0.25. In this specific case, the learning rate that minimizes the loss is 0.1.



4. Evaluate and compare with the results of HW3.

I've discovered that the neural network (NN) predictions outperform the other three predictions made in HW3 by a significant margin. In this experiment, where the learning rate was set to 0.1, the loss (mean of absolute errors) achieved with the NN was 99.12. In contrast, the previous losses were considerably higher at 866.80, 961.10, and 940.43.

Loss of NN in HW4

```
learning rate: 0.1
WARNING:absl:At this time,
s.legacy.Adam`.
2023-10-28 23:30:39.048
loss: 99.11775207519531
```

Loss of predictions from HW3.

```
Average difference of prediction 1: 866.80
Average difference of prediction 2: 961.10
Average difference of prediction 3: 940.43
```