# ASSIGNMENT 1 (DEEP LEARNING)

Done By : Sara Alsanajleh & Ghada Abu Shaqra

ID : 163418 & 164188

Supervisor :  Rasha Obaiedat

# DEFINITION OF LOGISTIC REGRESSION & MINI BATCH GRADIENT DESCENT

- Logistic regression is a supervised machine learning algorithm used for binary classification tasks. It predicts probabilities for a binary outcome (0 or 1) using the sigmoid function.

- **Mini-Batch Gradient Descent** divides the dataset into small, random batches, reducing computation time compared to full-batch gradient descent and smoothing out noisy updates from stochastic gradient descent.

# LOSS FUNCTION & REGULARIZATION & HYPERPARAMETER TUNING

- The **binary cross-entropy loss** is used in logistic regression as loss function .
- **L2 Regularization** helps prevent overfitting by penalizing large weights.
- **Hyperparameter Tuning :**

1. Learning Rate (lr)
2. Regularization Parameter (lambda_reg)
3. Batch Size
4. epochs

# Q1 : REPORT YOUR OBSERVATION ON THE CONVERGENCE BEHAVIOR OF THE GRADIENT DESCENT. USE A FIXED NUMBER OF DR. RASHA OBEIDAT AI342 (DEEP LEARNING) JORDAN UNIVERSITY OF SCIENCE AND TECHNOLOGY ITERATIONS (E.G., 1000) AS YOUR STOPPING CONDITION. PLEASE MENTION YOUR CHOICE CLEARLY IN YOUR REPORTPLEASE MENTION THE LEARNING RATES THAT CAUSE AN OVERFLOW. REPORT THE TRAINING AND DEV ACCURACY FOR EACH LEARNING RATE VALUE IN A TABLE, WHICH ONE ARE YOU GOING TO USE FOR THE FINAL MODEL?

- Model behavior is strong : Training Accuracy is (80%) & Validation Accuracy is (85%).
- The data is divided into 3 sections: the training section, which we trained the model on, the validation section, which we modified the hyper parameters through, and the testing section for the prediction of  unseen data.
- We noticed in the model that the larger the epochs , the more stable the results became.
- We also noticed that the LR, which has a large value, gave good results on the data we are working on.

# …………… Q1 …………..

- We added something simple to our code, which is that the lr is big at first, then the lr gets smaller as we go through the epochs . This improved the results.
- The learning rates that cause an overflow are : (1000 & 10000).
- The fixed number of epochs is : 589
- which one are you going to use for the final model ? LR = 0.01 The best in my model.

# ……………. Q1 …………….

| Learning Rate | Training Accuracy | Validation Accuracy | Test Accuracy |
|---|---|---|---|
| 10000 | 78% | 81% | 81% |
| 1000 | 77% | 79% | 80% |
| 100 | 74% | 76% | 82% |
| 10 | 78% | 79% | 82% |
| 1 | 79% | 84% | 82% |
| 0.1 | 79% | 83% | 82% |
| 0.01 | 79% | 85% | 82% |
| 0.001 | 71% | 78% | 69% |
| 0.0001 | 68% | 70% | 67% |
| 0.00001 | 68% | 70% | 67% |

**We adopted the lr value of 0.01 because it is the highest accuracy in the validation.**

# Q2: PLOT THE TRAINING ACCURACY, THE VALIDATION ACCURACY OF YOUR MODEL AS A FUNCTION OF THE NUMBER OF GRADIENT DESCENT ITERATIONS FOR EACH LEARNING RATE OF THE THREE LEARNING RATES: 0.1, 0.001 .00001. THE Y-AXIS IS THE ACCURACY, THE X-AXIS IS THE NUMBER OF ITERATIONS. WHAT HAVE YOU OBSERVED? WHY? IF ANY OF THE LEARNING RATE VALUES ABOVE LEAD TO DIVERGENCE, CHOOSE AN ALTERNATIVE ONE.

There is no divergence in any of the values (0.1 , 0.001 , 0.00001)

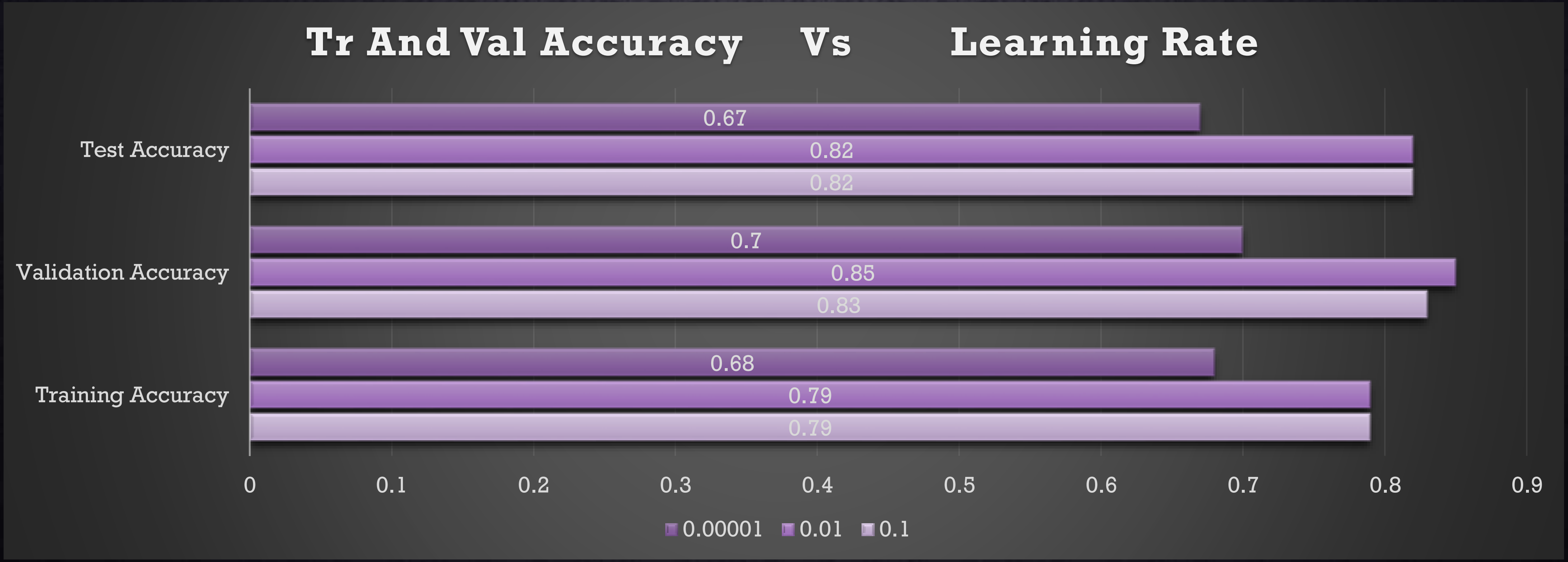
When lr = 0.1, learning was better and accuracy was higher. The solution was to make it large at first, and then we use the epoxy to reduce it.

When lr=0.01 the learning rate is slower because it is walking in baby steps.

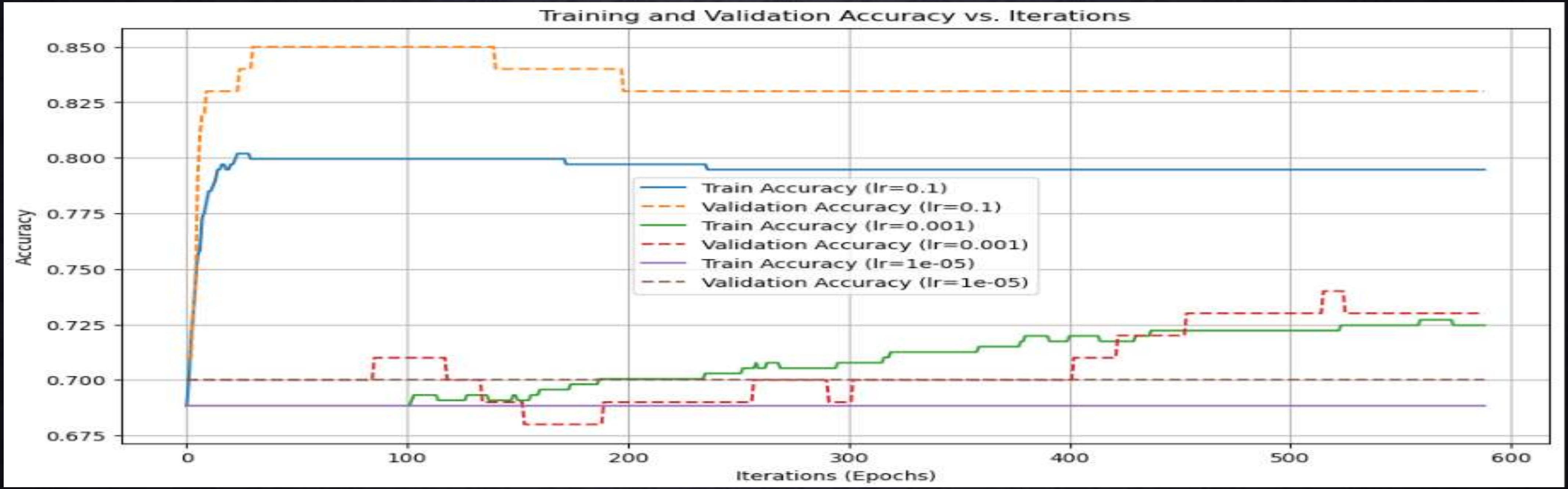
When lr = 0.00001 the learning rate is too slow for the model to learn patterns efficiently during a small number of iterations.

# Note : The answer of the above questions in the next slide .

# ………….. Q2 ……………..


Tr And Val Accuracy Vs Learning Rate
Test Accuracy
0.67
0.82
0.82
Validation Accuracy
0.7
0.85
0.83
Training Accuracy
0.68
0.79
0.79
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
▨ 0.00001 ▨ 0.01 ▨ 0.1

# ……………. Q2 ………………….



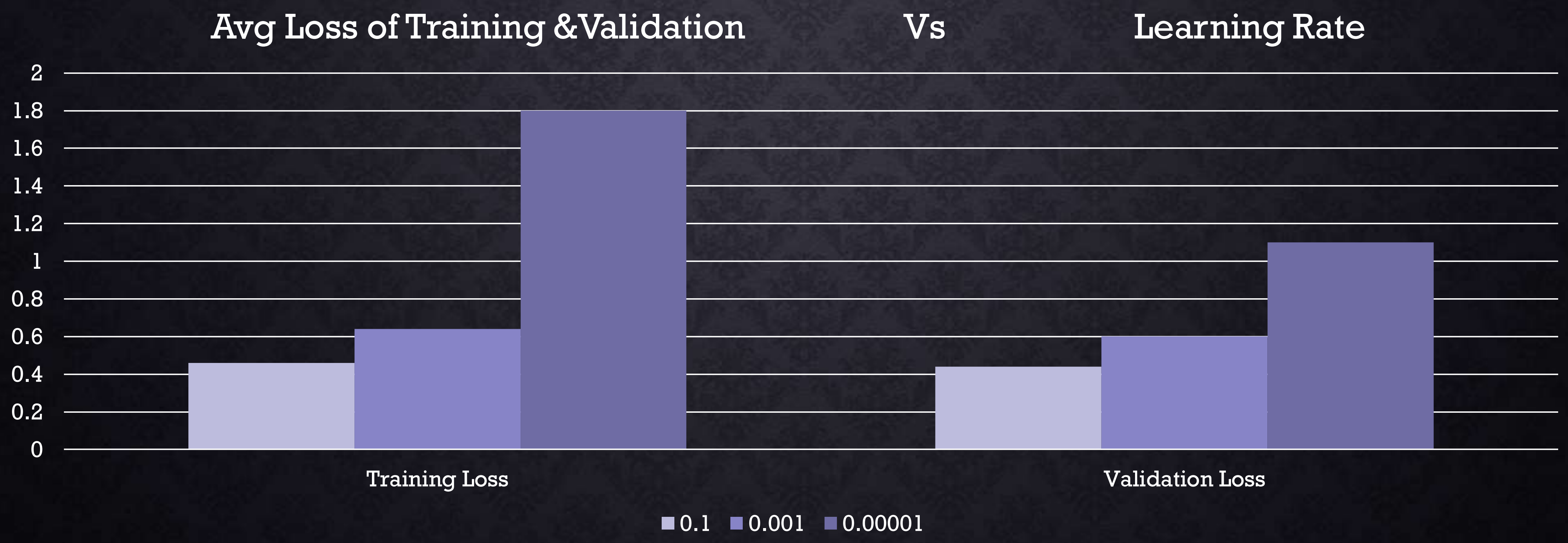
What have you observed?

My note is that 0.1 is the best thing because the accuracy is higher and the loss is lower. I will prove for the loss in the third question.

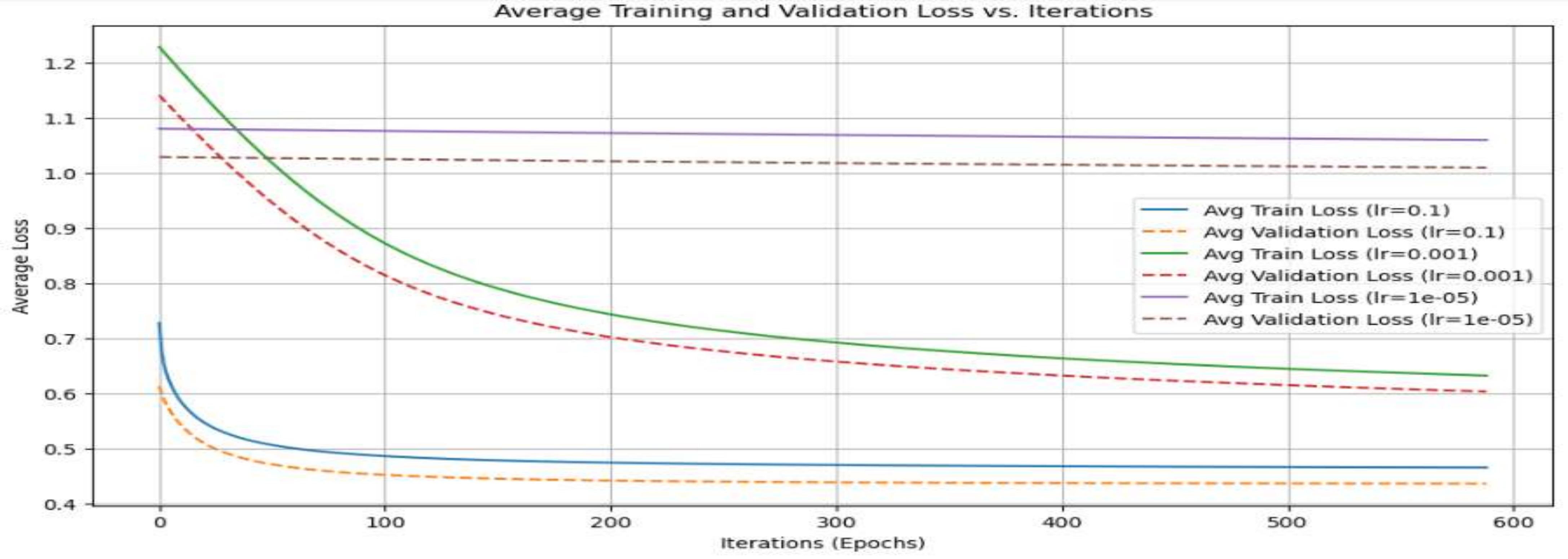
And lr = 0.001 is better than 0.00001

Why ? When the model walks as a baby in steps, it needs more training to learn,
so if I give him a few epochs, the strength will remain low,
but if the learning rate is very high, he may be kept away from the global minimum
, so I want to do a hyper parameter tuning for checking the best value.

# Q3 : REPEAT 2 BY PLOTTING THE AVERAGE TRAINING LOSS AND THE DEV LOSS INSTEAD OF THE ACCURACY. THE TRAINING AND THE DEV LOSSES ARE COMPUTED BY DIVIDING THE LOSS SUMS BY THE NUMBER OF SAMPLES.

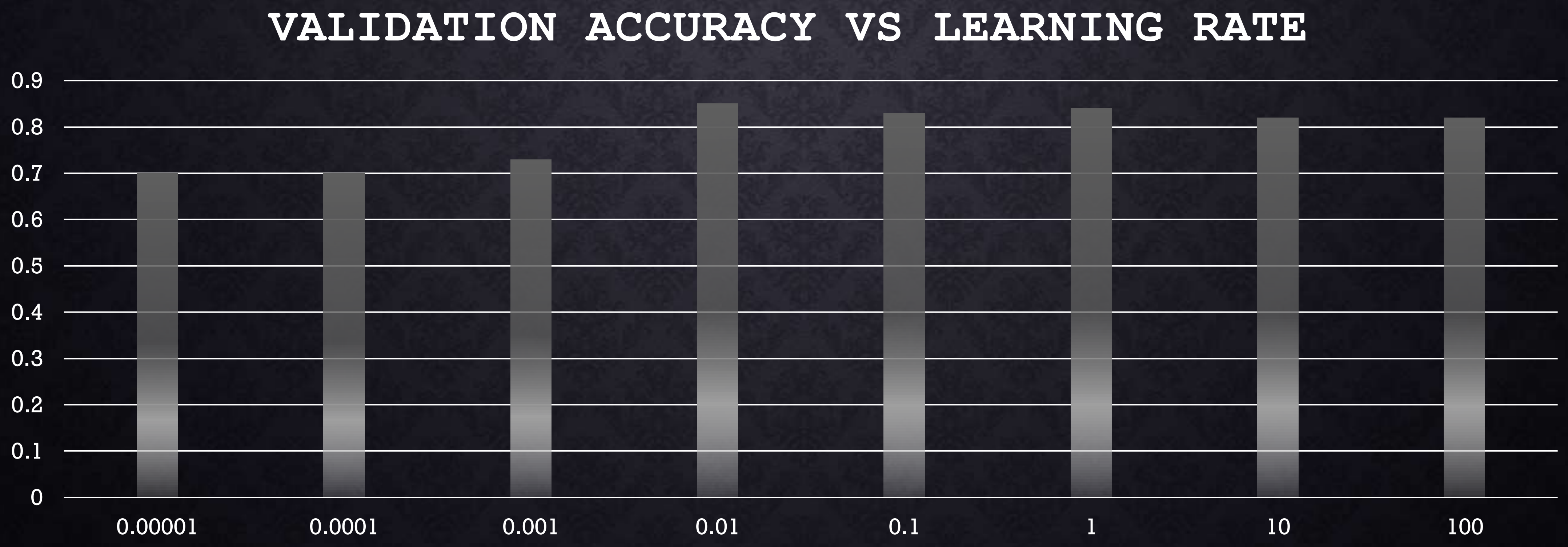


# …………….. Q3 ………………..



The lowest loss is at lr = 0.1, which is the best among the 3 learning rates above .

# Q4 : PLOT THE VALIDATION ACCURACY AS A FUNCTION OF LEARNING RATE (THE ONES THAT DID NOT LEAD TO OVERFLOW)? WHAT DO YOU OBSERVE? WHICH LEARNING RATE GAVE YOU THE BEST RESULTS. THE Y-AXIS IS THE ACCURACY, THE X-AXIS IS THE LEARNING RATE (BAR CHARTS OR DOT CHARTS ARE GOOD CHOICES FOR THIS QUESTION).

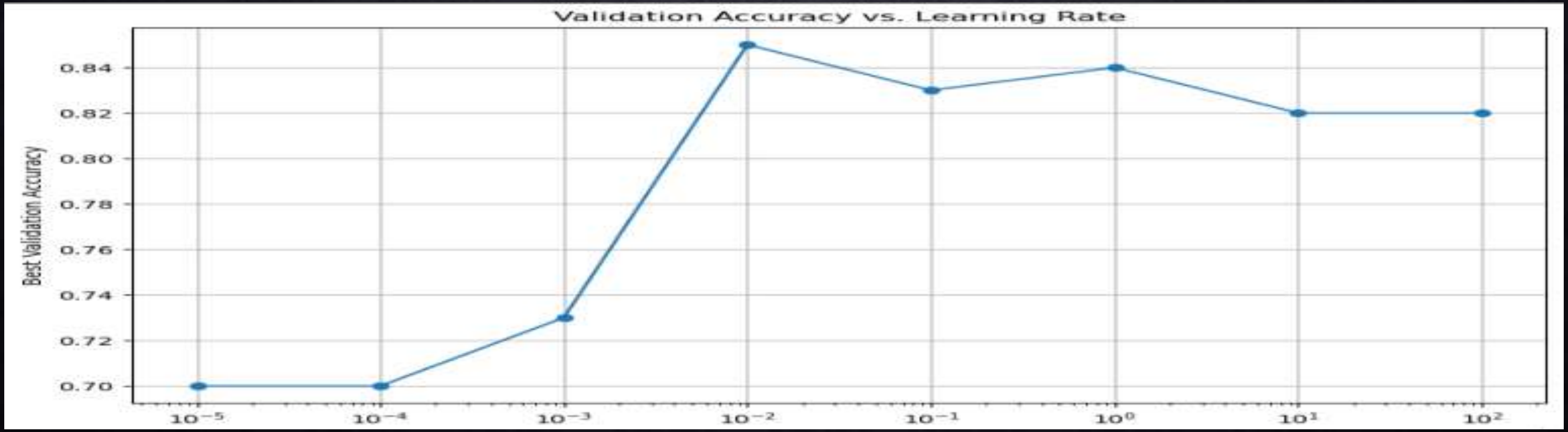


# ………………..Q4 ………………..

- What do you observe ?

I noticed, according to the results I got, that the small learning rate often had bad results on the data we had, while the high learning rate had good results on the same data.

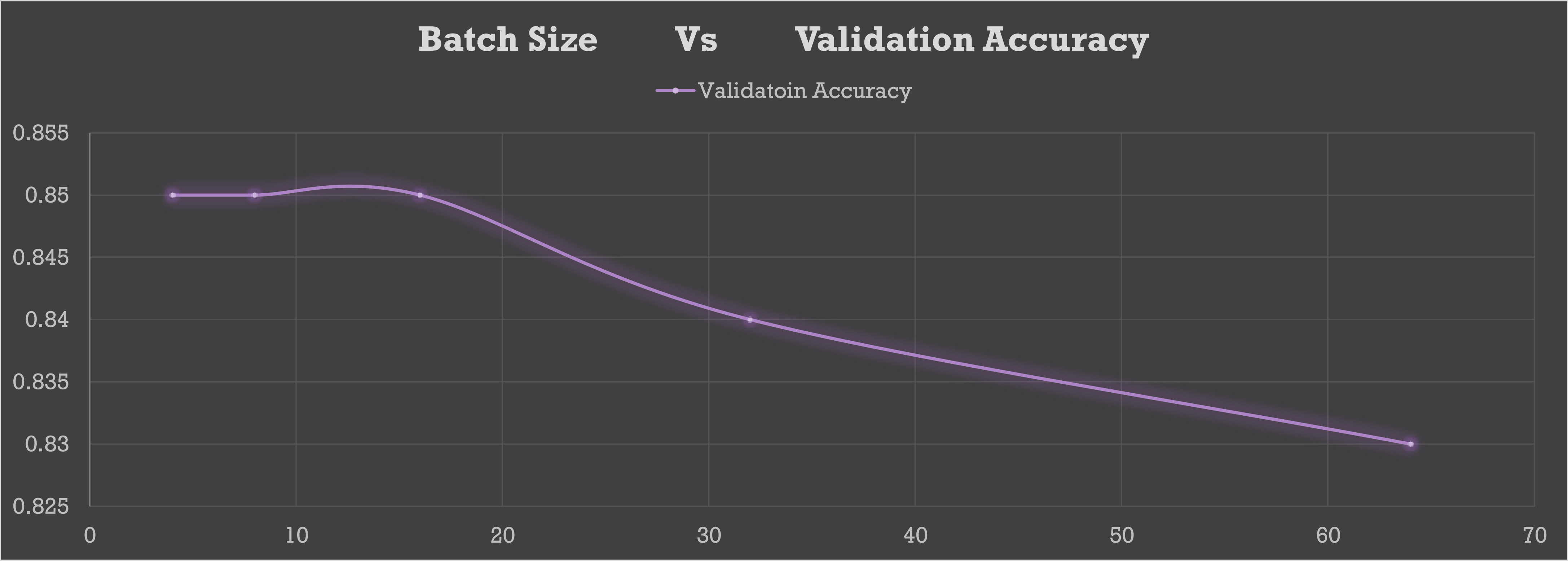
Which learning rate gave you the best results ? 0.01



# Q5 : REPORT THE TESTING ACCURACY THAT CORRESPONDS TO THE BEST LEARNING RATE (YOU DECIDED FROM THE PREVIOUS STEP).

- **Testing Accuracy with best learning rate (0.01) :** 0.82

# Q6 : TRY DIFFERENT BATCH SIZES (4, 8, 16, 32, 64) AND PLOT THE VALIDATION ACCURACY FOR THESE VALUES IN ONE FIGURE.



Which batch size gave the best performance? (4 , 8 , 16) The Same Result all of them are good

# Q7 : ADD L2 REGULARIZATION TO THE IMPLEMENTATION OF GRADIENT DECENT, WHAT DID YOU ADD TO YOUR IMPLEMENTATION? DID ADDING L2 REGULARIZATION CHANGE TO THE BEST VALIDATION AND TESTING ACCURACY YOU HAVE GOT? WHAT IS THE VALUE OF $\lambda$ YOU HAVE CHOSEN AND WHY? DO YOU RECOMMEND USING REGULARIZATION? PLOT THE TRAINING AND THE VALIDATION ACCURACY FOR DIFFERENT VALUES OF $\lambda$, YOU THE BEST LEARNING RATE VALUE YOU HAVE OBTAINED TO DO THIS PART

- what did you add to your implementation? I add penality term → → →

$\text{Loss} = \text{Original Loss}+2\lambda\sum w2$ and the GD will be $\nabla\text{Loss}+\lambda w$ .

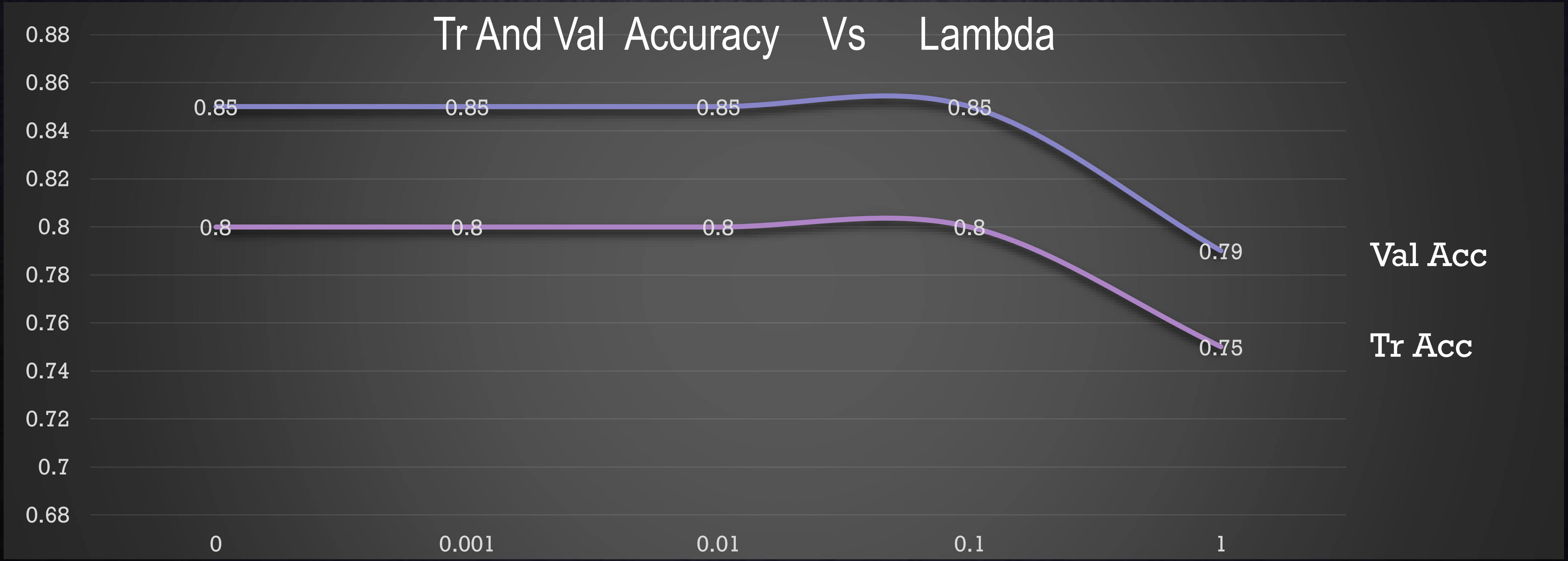
Did adding L2 regularization change to the best validation and testing accuracy you have got? In my code no, the results remains as before because I basically didn't have over fitting and the main functionality of regularization is prevent the overfitting .

What is the value of $\lambda$ you have chosen and why ? 0.01 because it is the most appropriate penality term for data and the validation accuracy is high through it ….I did a hyper parameter tuning to know who is the best.

Do you recommend using regularization? Yes, I recommend using **regularization**, particularly L2 regularization as it offers several advantages especially in models that may be prone to **overfitting**.

(**Reduces overfitting** …. improves model stability …. enhanced interpretability).

# PLOT THE TRAINING AND THE VALIDATION ACCURACY FOR DIFFERENT VALUES OF $\lambda$, YOU THE BEST LEARNING RATE VALUE YOU HAVE OBTAINED TO DO THIS PART



# Q8 : TRAIN A MODEL WITHOUT DATA NORMALIZATION, DOES IT WORK? IS IT EASY TO TRAIN? DID YOU NEED MORE ITERATIONS? EXPLAIN YOUR OBSERVATION? WHICH RATE OF LEARNING WORKS THE BEST FOR UNNORMALIZED DATA?

- **In my code Just I didn't run the cell of Normalization.**
- does it work? **Yes, a model can still work without data normalization** but in slowely way it have learn and **the performance is often suboptimal.**
- Is it easy to train? No , **training becomes more challenging without normalization**
- Did you need more iterations? Yes , because it learns slowely

Explain your observation? Without normalization there will be a divergence in the range of values so there will be a higher weight for the featurers that have a high range of values. Therefore, not all featurers have the same weight so the value of the Accuracy will be lower and the loss will be higher.

**Which rate of learning works the best for unnormalized data? 0.001**

**( Do similar plot for 3 learning rate values of your choice like the plots in part Q1.**

**I did the thing with my code by not running the normalization cell and all the plots after it ..... are running**