Tecnológico de Monterrey

Final project
Deep learning

Team 13:

A01566271 Eduardo Blanco
A01338798 Edgar Rubén Salazar Lugo

06/june/2021

A. Introduction

We were motivated in being able to apply all knowledge that we could get during the whole semester. We have already worked with different data sets in class and made some predictions but finding our own result, interpreting them and evaluating if we are following the correct path was the best part of the process.

Our goal with this project was to find a data set capable of making a binary classification or being able to modify it in order to create a model for two new data (dev and test).

B. Data Set

We struggled finding the final data set because the websites from the instructions were not helpful and finding a data set with binary classification was even harder. However, in the end we found a data set in kaggle that already had three folders with our training, dev and test data. The train data had a total of 1219 cases, dev data had only 20 cases and the test had 267 cases.

In our data set we can determine if the woman has makeup or not. So, in the folder of “makeup” there are images with models, magazine photos and this kind. In the folder of “no_makeup” we could see women with natural poses and ordinary photos.

As the data set is made of images and not arrays, we had to resize the images to the same shapes and convert that image to an numpy array. In that way, we could train our model and make the predictions.

C. Methodology

The first steps we did were to run the two models we already had for the homework (2 layer model and 4 layer model).

Then, we created a generic method that could receive the iterations, learning rate and layers. The only thing we had to do was to call the method with the corresponding parameters.

Then, what we tried to do was to increase the number of iterations, decrease the learning rate and increase the number of layers to five. With these steps we got better results, however, we tried to improve our predictions. That’s why we conserve the same amount of layers but modify the learning rate and number of iterations.

```
### CONSTANTS ###
layers_dims = [12288, 20, 14, 7, 5, 1] #  5-layer model
```

```
parameters = us_layer_model(train_x, train_y, layers_dims, learning_rate = 0.005, num_iterations = 5000, print_cost = True)
```

We observed that the iterations we had setted took a lot of time to train the data. That's why we decided to decrease that number and test it with a lower learning rate. However, our predictions were worse.

```
In [26]: ### CONSTANTS ###
         layers_dims = [12288, 20, 14, 7, 5, 1] #  5-layer model
```

```
In [27]: parameters = us_layer_model(train_x, train_y, layers_dims, learning_rate = 0.0005, num_iterations = 3000, print_cost = True)
```

After some tests with the same amount of layers, we decided to modify that parameter and test it with the previous learning rates and iterations. However, our results were getting worse and worse. That's why we came back to five layers but with a learning rate equal to the first two given by the teacher with a low amount of iterations. The results were better but not enough.

D. Results

The first results we got based on the models that were given, were these:

```
predictions_train = predict(train_x, train_y, parameters)
```

```
Accuracy: 0.8301886792452831
```

```
predictions_dev = predict(dev_x, dev_y, parameters)
```

```
Accuracy: 0.5
```

2 layer model

```
pred_train = predict(train_x, train_y, parameters)
```

```
Accuracy: 0.9105824446267432
```

```
pred_dev = predict(dev_x, dev_y, parameters)
```

```
Accuracy: 0.6500000000000001
```

4 layer model

Even though 0.65 is an acceptable model and following our first methodology we got our better result:

```
pred_train = predict(train_x, train_y, parameters)
Accuracy: 0.9122231337161608
```

```
pred_dev = predict(dev_x, dev_y, parameters)
Accuracy: 0.7500000000000002
```

5 layer model - 0.005 leaning rate - 5,000 iterations

Surprisingly, our first attempt to improve the hyper parameters was our best result for the dev set data with a final result of 0.75. Based on this final result, we were satisfied but we followed our methodology and found a better model but we could not find it. Some of the best results with our differents hyper parameters are the following ones.

```
pred_train = predict(train_x, train_y, parameters)
Accuracy: 0.7038556193601313
```

```
pred_dev = predict(dev_x, dev_y, parameters)
Accuracy: 0.5
```

5 layers - 0.0005 learning rate - 3,000 iterations

```
pred_train = predict(train_x, train_y, parameters)
Accuracy: 0.8621821164889254
```

```
pred_dev = predict(dev_x, dev_y, parameters)
Accuracy: 0.7000000000000002
```

3 layers - 0.0075 learning rate - 2,500 iterations

```
pred_train = predict(train_x, train_y, parameters)
Accuracy: 0.7038556193601313
```

```
pred_dev = predict(dev_x, dev_y, parameters)
Accuracy: 0.5
```

5 layers - 0.005 learning rate - 5,000 iterations

E. Final results

Finally, as we saw that our best model was the one with five layers, a 0.005 learning rate and 5,000 iterations. we decided to make a prediction with our test data and determinate if the result could be similar:

```
pred_dev = predict(dev_x, dev_y, parameters)
```

```
Accuracy: 0.7500000000000002
```

```
pred_test = predict(test_x, test_y, parameters)
```

```
Accuracy: 0.6629213483146068
```

On one hand, as we can see, the result with the test data is lower but bigger than the middle point (0.5). We know that an accuracy of 0.66 is not the expected but based on the previous models, we inferred that it's the best prediction.

On the other hand, we also did a prediction for the other test we did and the biggest was with 0.72

```
pred_dev = predict(dev_x, dev_y, parameters)
```

```
Accuracy: 0.5
```

```
pred_test = predict(test_x, test_y, parameters)
```

```
Accuracy: 0.7265917602996256
```

We do not consider this model trustworthy because its dev data has a 0.50 accuracy.

F. Conclusions

We conclude that making a model is not as simple as we could think. It requires time, effort and a lot of analysis. With this project, we could apply all knowledge during the whole semester and determine if some parameter could impact and how.

Also, we conclude that what we learned in this project is very useful for our future projects because deep learning is such a support for the new technologies that are coming in the following years.

G. References

makeup or not makeup. (2021). Retrieved 6 June 2021, from https://www.kaggle.com/gurusabarish/makeup-or-not-makeup