

- Describe what problems you encountered and how did you solve them when implementing the basic and advanced functions. (2%)  
When I was writing model.backward(), I kept having error like

```

69 """
70 A_prev, W, b = self.cache
--> 71 m = A_prev.shape[1]
72 # GRADED FUNCTION: linear_backward
73 ### START CODE HERE ### (~ 3 lines of code)
75 self.dW = 1/m * np.matmul(dZ, A_prev.transpose())

AttributeError: 'str' object has no attribute 'shape'

```

After spending lots of time debugging, I found that I should write like this

```

#self.cache = {"A": A.copy(), "W": self.parameters["W"].copy(), "b": self.parameters["b"].copy()}
self.cache = (A.copy(), self.parameters["W"].copy(), self.parameters["b"].copy())

```

If I write something like the line that I comment, the string A,W,b will be store into cache.

- Briefly describe the structure of your binary and multi-class classifiers. (2%)

Basic

```

layers_dims = [30,64,1]
activation_fn = ["relu", "sigmoid"]
learning_rate = 0.4
num_iterations = 1000
print_cost = True

```

Advanced

```

layers_dims = [28*28,512,64,10]
activation_fn = ["relu", "relu", "softmax"]
learning_rate = 0.01
num_iterations = 103
batch_size = 100
print_cost = True
classes = 10

```

- Describe effort you put to improve your model (e.g., hyperparameter finetuning). (1%)  
Test the layer of the model and activation function to determine which is better.