Improve deep neural network

# Methods

## Initialize W and b

### Initialize parameters zeros

W and b are both initialize to zero, this causes the neural network symmetry, which is not beneficial to improve neural network no matter how many training data.

### Initialize parameters random

Initialize W with random, which has a normal distribution. It had the same function with normalization. Whereas, b is the same with former, all zeros.

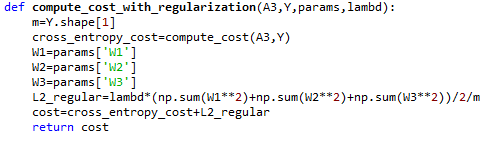
### Initialize parameters he

‘He’ is not the true meaning at here, it means we use the method that is not convenient to give a definition. It can represent everyone who make little changes of random.

## overfit

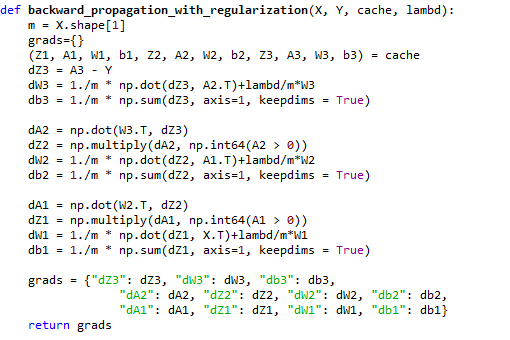
When the training data are not enough, the model may present with over fit. Regularizations, dropout may solve this sort of problems.

### Regularization:



Lambda is the parameters in regularization, it improves neural network by reducing the W and b smaller, which make the learning smooth and reach the lowest point quickly.

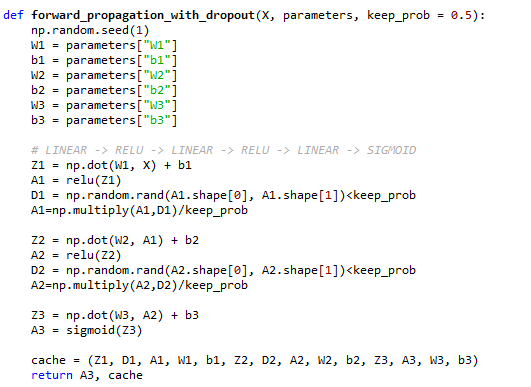
Since the forward propagation add lambda parameters, the backward propagation should be change synchronic.

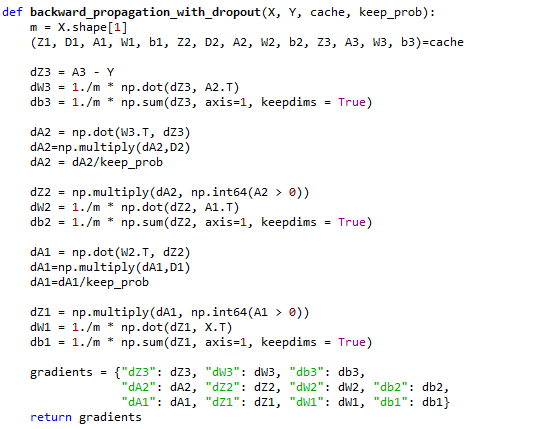


### Dropout:

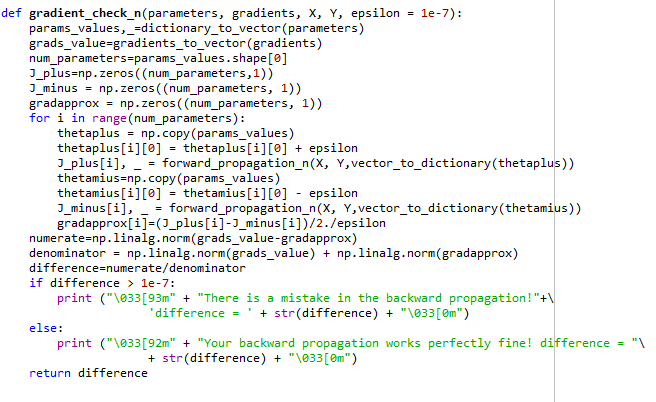
The main idea if dropout is choosing partly neural network randomly based on probably. If the data is large enough, we can get right classification for every input.

We should remember, that if the forward propagation changed, backward propagation also need change, in practice, we need to change cache.





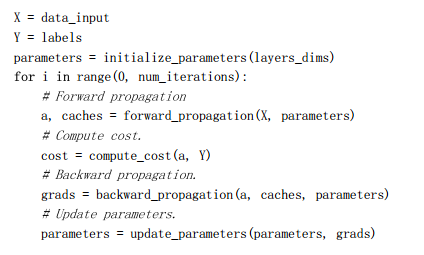
## gradient check

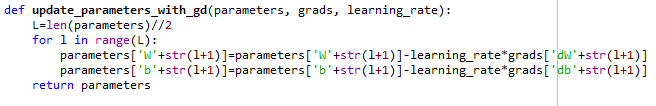


## optimization

Until now, we always use Gradient Descent to update the parameters and minimize the cost. In this notebook, will will learn more advanced optimization methods that can speed up learning and perhaps even get a better final value for the cost function.

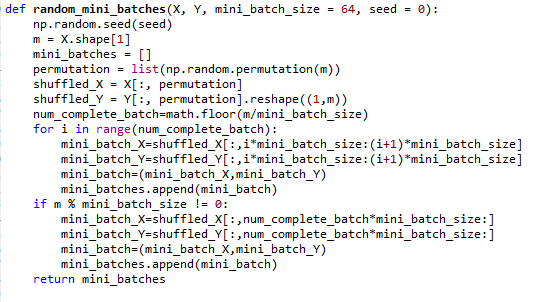
### Gradient Descent





### Mini-Batch Gradient descent

1. the first step is permutation
2. partition



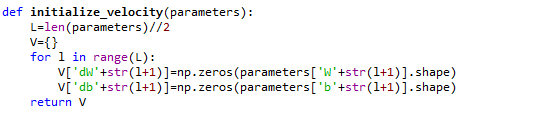
What you should remember:

Shuffling and Partitioning are the two steps required to build mini-batches;

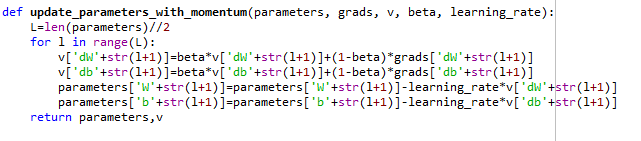
Powers of two are often chosen to be the mini-batch size, e.g., 16, 32, 64, 128.

### Momentum

1. initialize velocity



1. compute parameters, gradients and velocity, then update the parameters



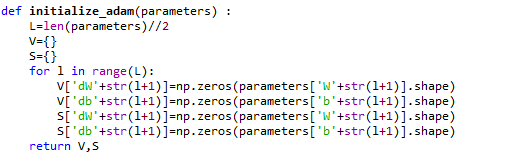
Momentum takes past gradients into account to smooth out the steps of gradient descent. It can be applied with batch gradient descent, mini-batch gradient descent or stochastic gradient descent.

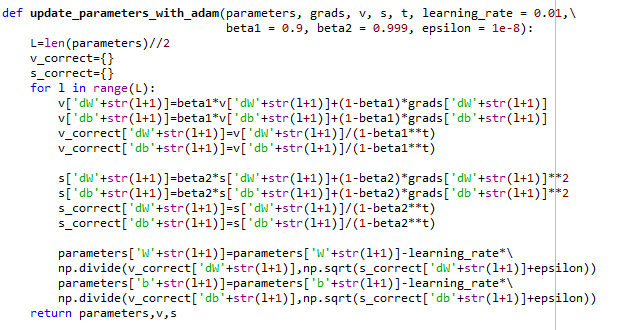
You have to tune a momentum hyper parameter β and a learning rate α.

### Adam

Adam is one of the most effective optimization algorithms for training neural networks, it combines the idea from RMSProp and momentum.

1. Initialize Adam parameters, including v, s
2. Correct v and s;
3. Update parameters;

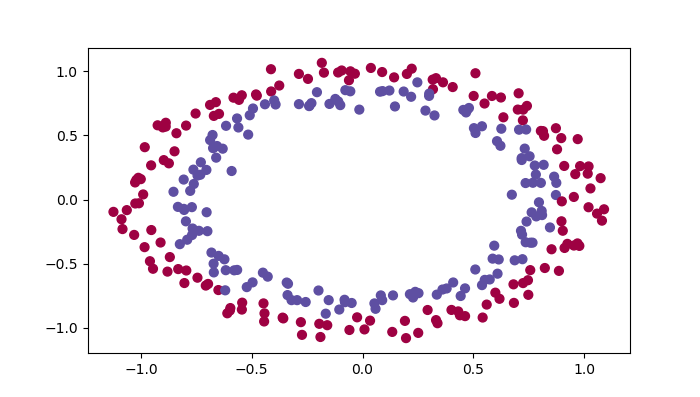




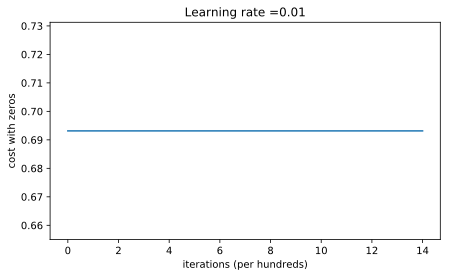
# Result

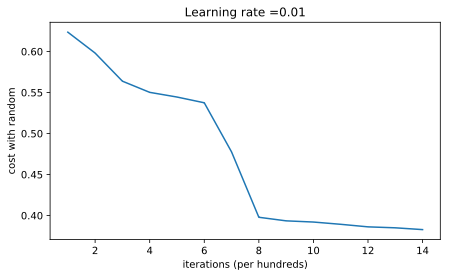
## Initialize result:

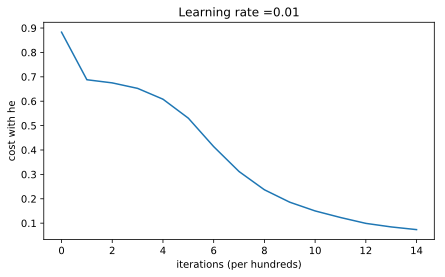
### Original data:



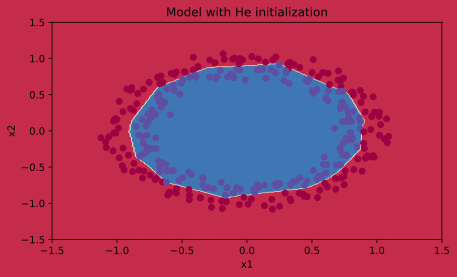
### Cost with different initialize methods







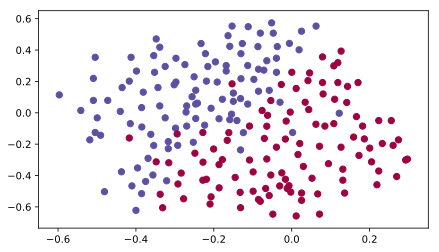
### Accuracy



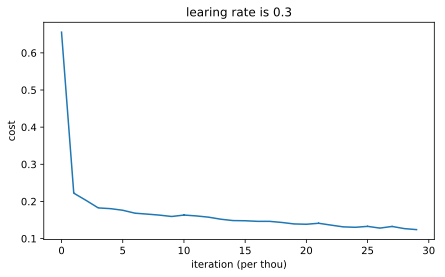
|  |  |  |
| --- | --- | --- |
| Initialize methods | Train accuracy | Test accuracy |
| Zeros | 0.5 | 0.5 |
| Random | 0.83 | 0.86 |
| He | 0.9933333333333333 | 0.96 |

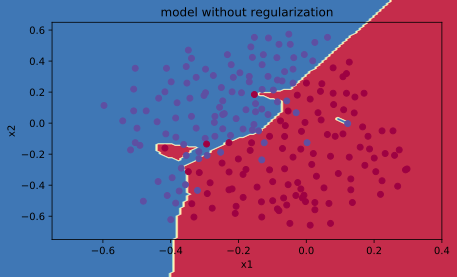
## Regularization and dropout result:

### Original figure

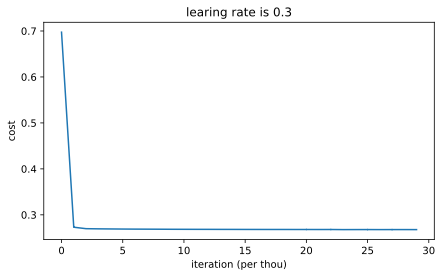


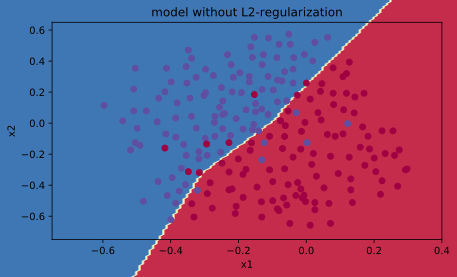
### Without regularization:



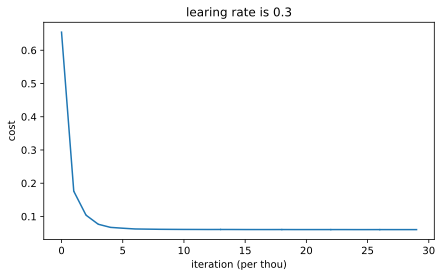


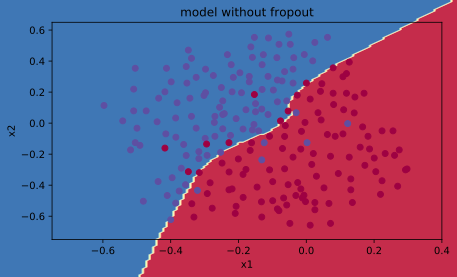
### regularization:





### dropout:



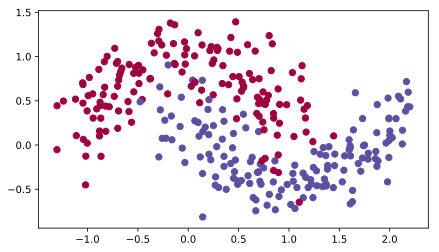


### Accuracy:

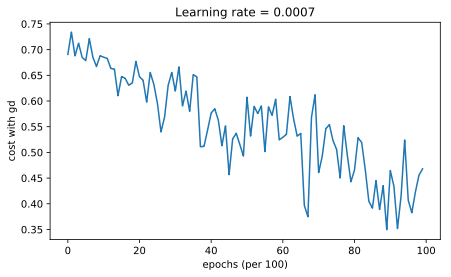
|  |  |  |
| --- | --- | --- |
| methods | Train accuracy | Test accuracy |
| Without regularization | 0.948 | 0.915 |
| Regularization | 0.9384 | 0.93 |
| dropout | 0.9289 | 0.95 |

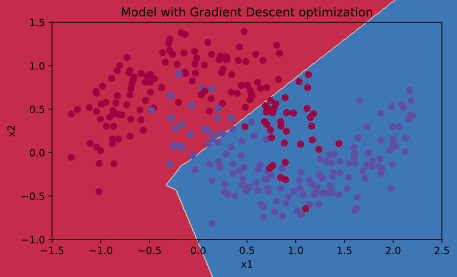
## Optimization

### Original figure:

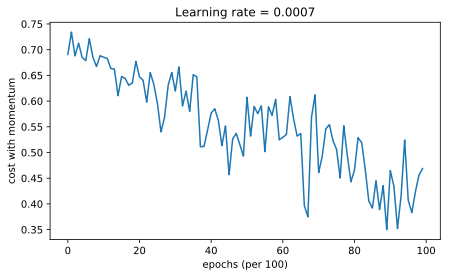


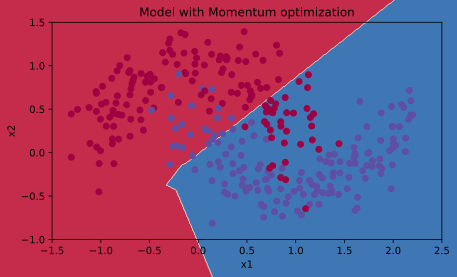
### Gradient check:



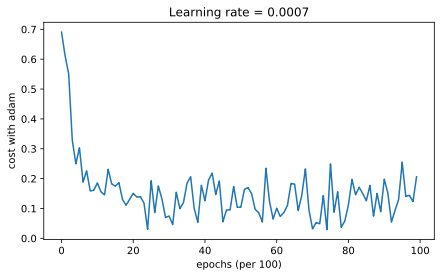


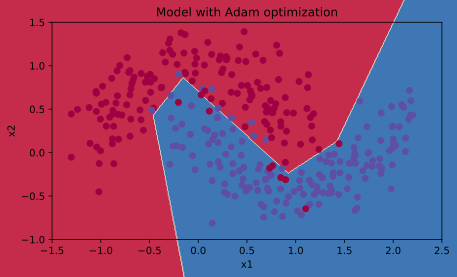
### Momentum:





### Adam:





### Accuracy:

|  |  |
| --- | --- |
| methods | Training data |
| Gradient | 0.797 |
| Momentum | 0.797 |
| Adam | 0.94 |

# Conclusion

In this scenario, it is apparently that in initialize process, he initialized method works better than two others. The regularization and dropout methods can both improve the test accuracy.

Momentum usually helps, but given the small learning rate and the simplistic dataset, its impact is almost negligible. Also, the huge oscillations you see in the cost come from the fact that some mini batches are more difficult than others for the optimization algorithm. Adam on the other hand, clearly outperforms mini-batch gradient descent and Momentum. If you run the model for more epochs on this simple dataset, all three methods will lead to very good results. However, you've seen that Adam converges a lot faster. Some advantages of Adam include: Relatively low memory requirements (though higher than gradient descent and gradient descent with momentum).