Initialization

A well chosen initialization can:

1. speed up the convergence of gradient descent
2. Increase the odds of gradient descent converging to a lower training (and generalization) error

- Zeros initialization
- Random initialization
- He initialization Best!

Initialize parameters **zeros**:

```
for l in range(1, L):
    parameters['W' + str(l)] = np.zeros((layers_dims[l],
                                         layers_dims[l-1]))
    parameters['b' + str(l)] = np.zeros((layers_dims[l], 1))
```

random:

```
parameters['W' + str(l)] = np.random.randn(layers_dims[l],
                                           layers_dims[l-1])*10
parameters['b' + str(l)] = np.zeros((layers_dims[l], 1))
```

He initialization:

```
random.randn(layers_dims[l], layers_dims[l-1]) *
        np.sqrt(2. / layers_dims[l-1])
```

2 / dimension of previous layer