

# MATLAB常见的学习率下降策略

凯鲁嘎吉 - 博客园 <http://www.cnblogs.com/kailugaji/>

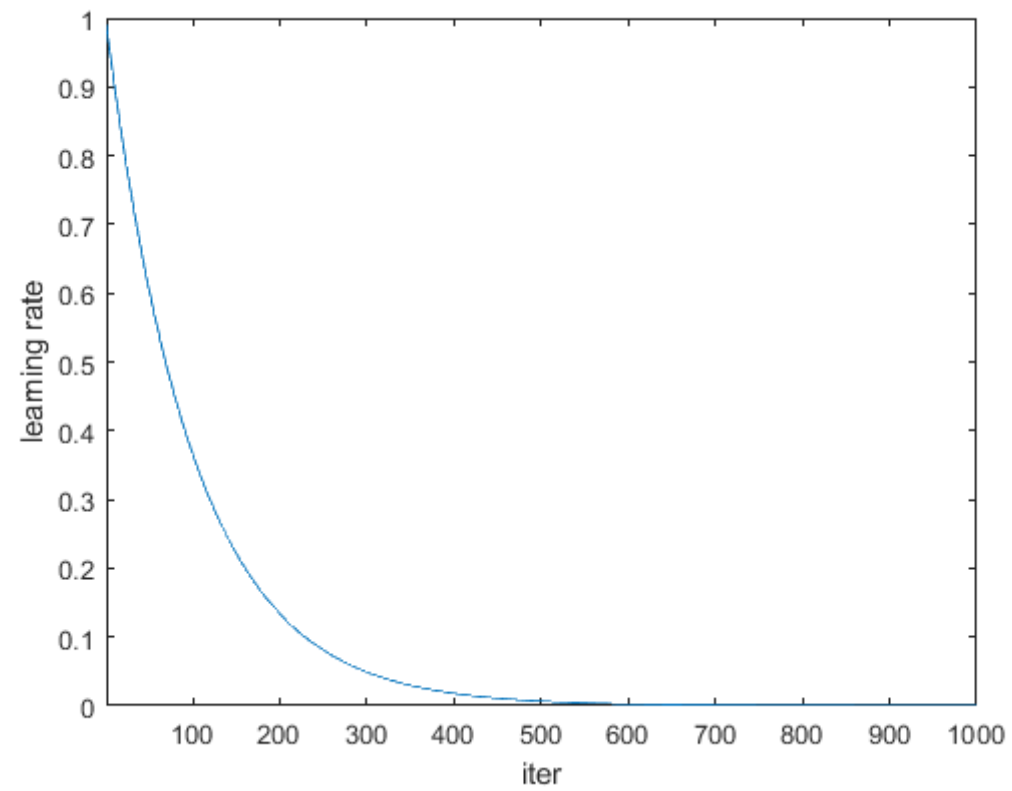
## 1. 几种常见的学习率下降策略 (learning rate decay strategy)

t: 自变量, 迭代次数,  $\lambda(t)$ : 因变量, 学习率, T: 常量, 最大迭代次数, 其他参数均为常量, 可自行设定。可以设定初始学习率 $\lambda(0)$ :

$$\lambda'(t) = \lambda(0)\lambda(t)$$

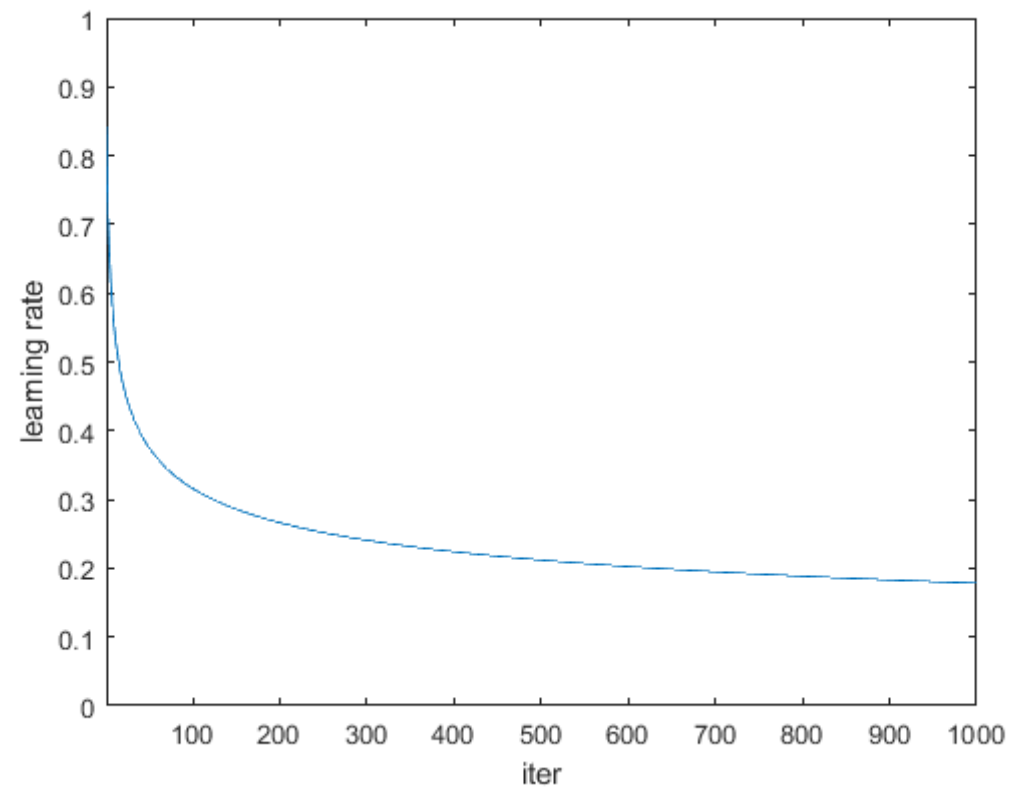
1) exp

$$\lambda(t) = \gamma^t$$



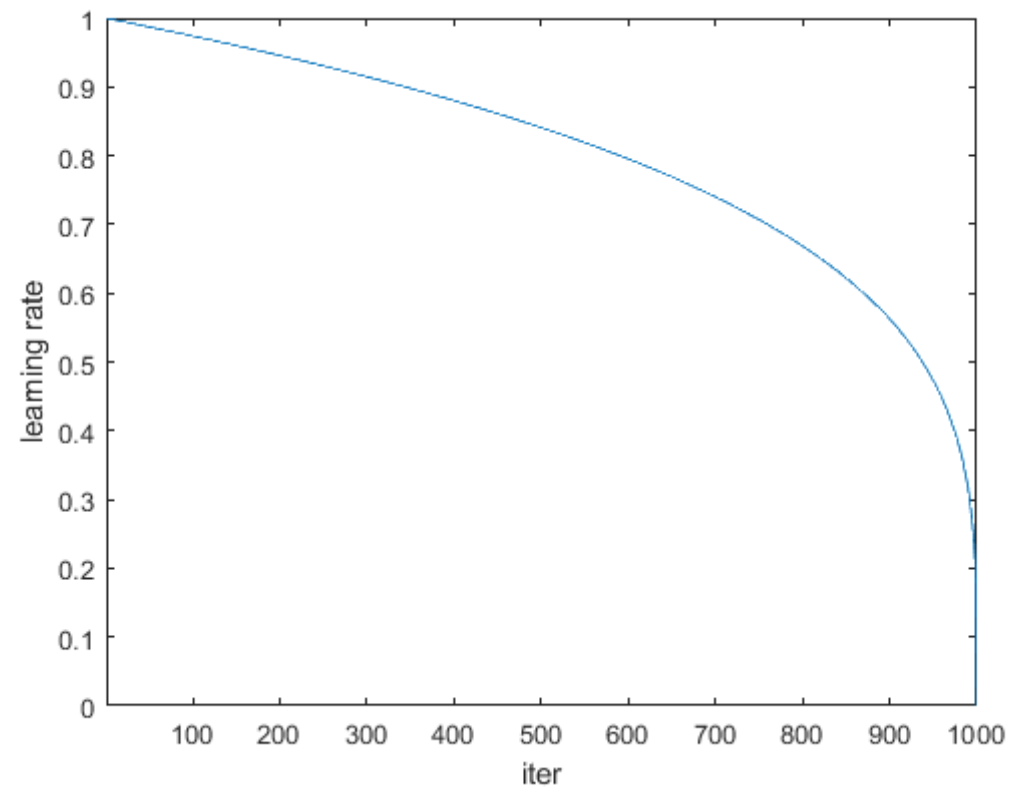
2) inv

$$\lambda(t) = (1 + \gamma \cdot t)^{-p}$$



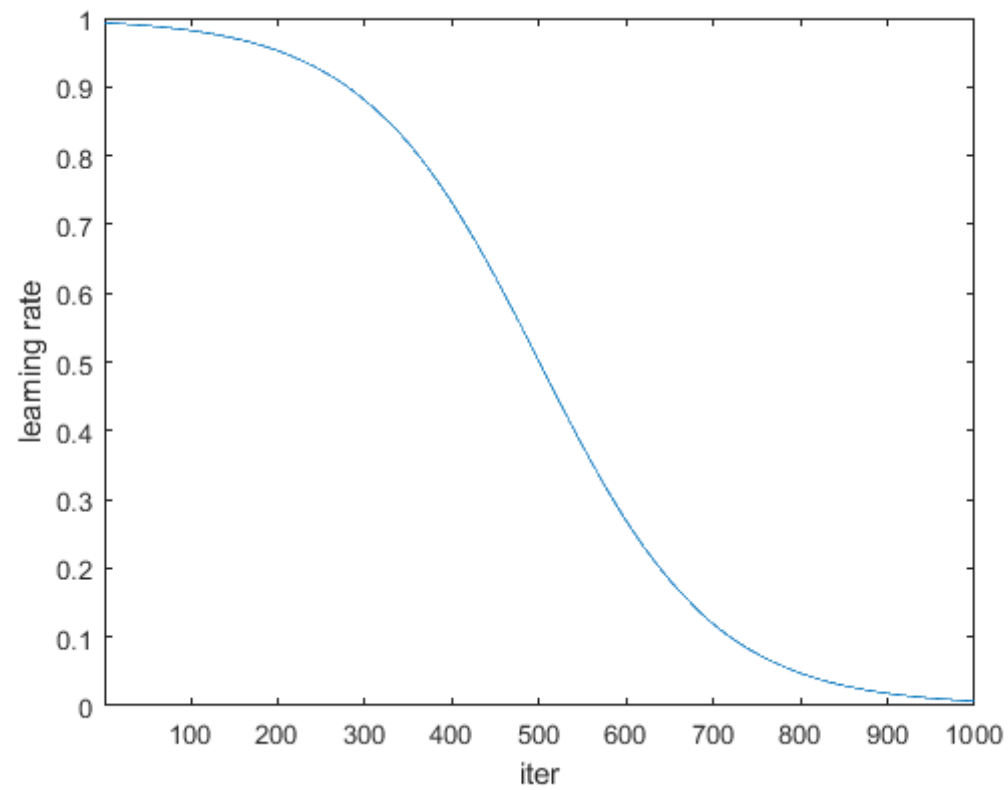
3) plot

$$\lambda(t) = \left(1 - \frac{t}{T}\right)^p$$



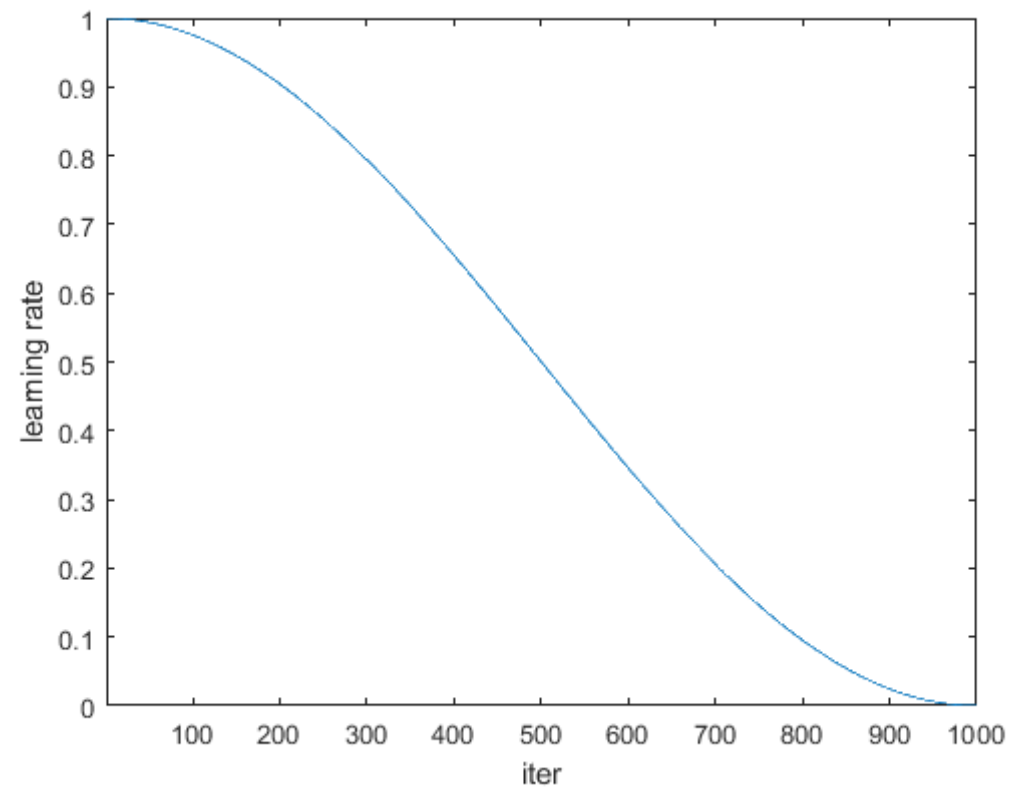
4) sigmoid

$$\lambda(t) = \frac{1}{1 + e^{-\gamma(t - \text{stepsize})}}$$



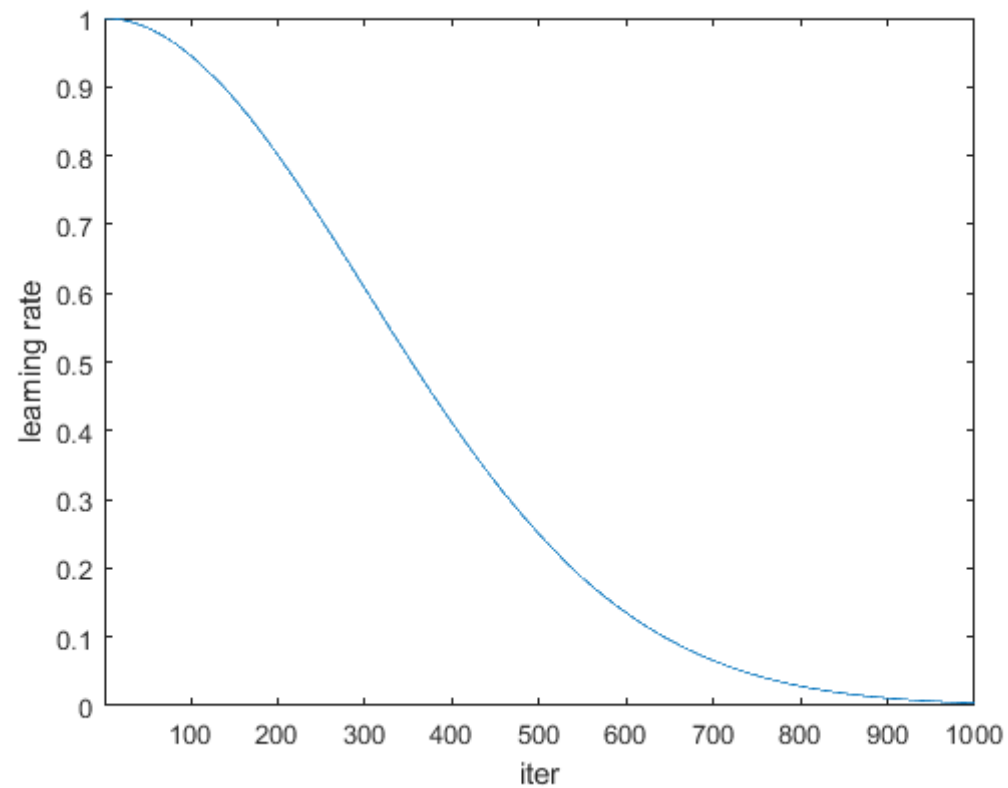
5) cosine\_decay

$$\lambda(t) = \frac{1}{2} \left( 1 + \cos\left(\frac{t\pi}{T}\right) \right)$$



## 6) Gaussian

$$\lambda(t) = e^{-\frac{t^2}{2\sigma^2}}$$



## 2. MATLAB程序

```
function learning_rate_decay(choose)
% Author: kailugaji 凯鲁嘎吉 - 博客园 http://www.cnblogs.com/kailugaji/
max_iter=1000;
y=zeros(1, max_iter);
t=1:max_iter;
if choose==1
    % cosine decay
    y(t)=0.5*(1+cos(pi*t/max_iter));
elseif choose==2
    % plot p>1:凹曲线, 0<p<1:凸曲线
    p=0.25;
    y(t)=(1-(t/max_iter)).^p;
elseif choose==3
    % inv
```

```

        gamma=0.99; p=0.25;
        y(t)=(1+gamma*t).^(-p);
elseif choose==4
    % exp
    gamma=0.99;
    y(t)=gamma.^t;
elseif choose==5
    % sigmoid
    gamma=-0.01; stepsize=max_iter/2;
    y(t)=1./(1+exp(-gamma*(t-stepsize)));
elseif choose==6
    % Gaussian
    sigma=300;
    y(t)=exp(-(t.^2)/(2*(sigma^2)));
else
    disp('input error!');
end
%%
plot(t, y);
axis([1, max_iter, 0, 1]);
xlabel('iter');
ylabel('learning rate');

```

### 3. 学习率衰减



从经验上看, 学习率在一开始要保持大些来保证收敛速度, 在收敛到最优  
点附近时要小些以避免来回震荡。比较简单的学习率调整可以通过**学习率衰减**  
(Learning Rate Decay)的方式来实现, 也称为**学习率退火**(Learning Rate An-  
nealing)。

假设初始化学学习率为  $\alpha_0$ , 在第  $t$  次迭代时的学习率  $\alpha_t$ 。常用的衰减方式为可  
以设置为按迭代次数进行衰减。常见的衰减方法有以下几种:

**分段常数衰减**(Piecewise Constant Decay): 即每经过  $T_1, T_2, \dots, T_m$  次迭  
代将学习率衰减为原来的  $\beta_1, \beta_2, \dots, \beta_m$  倍, 其中  $T_m$  和  $\beta_m < 1$  为根据经验设置  
的超参数。分段常数衰减也称为**步衰减**(Step Decay)。

**逆时衰减**(Inverse Time Decay):

$$\alpha_t = \alpha_0 \frac{1}{1 + \beta \times t}, \quad (7.5)$$

其中  $\beta$  为衰减率。

**指数衰减**(Exponential Decay):

$$\alpha_t = \alpha_0 \beta^t, \quad (7.6)$$

其中  $\beta < 1$  为衰减率。

自然指数衰减(Natural Exponential Decay):

$$\alpha_t = \alpha_0 \exp(-\beta \times t), \quad (7.7)$$

其中  $\beta$  为衰减率。

余弦衰减(Cosine Decay):

$$\alpha_t = \frac{1}{2} \alpha_0 \left( 1 + \cos \left( \frac{t\pi}{T} \right) \right), \quad (7.8)$$

其中  $T$  为总的迭代次数。

图7.4给出了不同衰减方法的示例(假设初始学习率为1)。

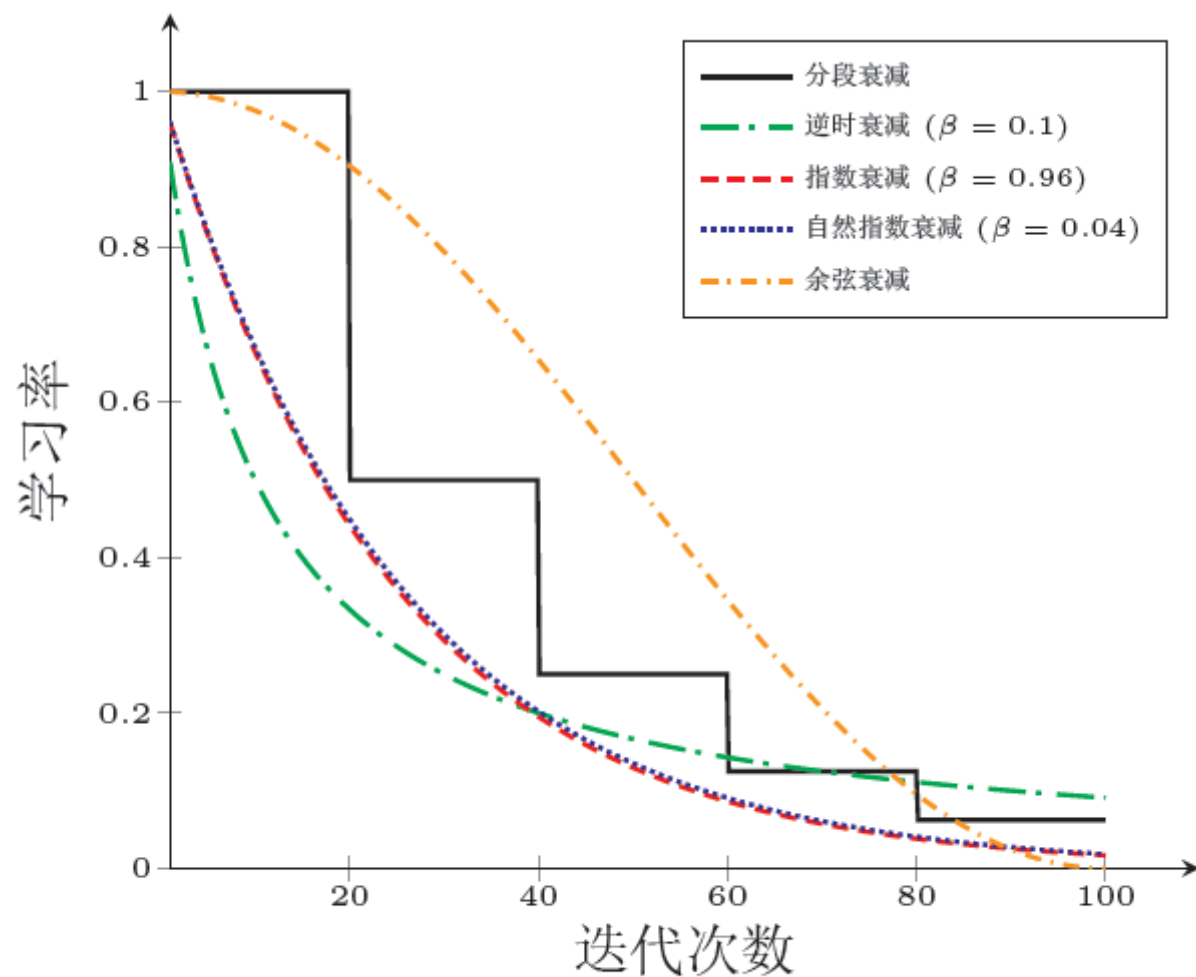


图 7.4 不同学习率衰减方法的比较

## 4. 参考文献

[梯度下降学习率的设定策略](#)

[Tensorflow中learning\\_rate\\_decay的奇技淫巧](#)

[TensorFlow学习 - - 学习率衰减/learning rate decay](#)

[图像分类训练技巧集锦 \(论文笔记\)](#)

邱锡鹏, [神经网络与深度学习](#)[M]. 2019.