RMSprop 法

公式:

$$v_t = \beta v_{t-1} + (1-\beta)\nabla f(x)^2$$

$$x = x - \alpha \frac{1}{\sqrt{v_t} + \epsilon} \nabla f(x)$$

RMSprop 法強調梯度的平方，通過對過去梯度平方進行指數移動平均，調整學習率，從而適應不同參數的變化情況。它讓變化較大的參數的學習率變小，從而減少了震盪和不穩定情況。

Adam 法

公式:

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t$$

$$v_t = \beta_2 m_{t-1} + (1-\beta_2) {g_t}^2$$

Adam 法就像是 RMSprop 和 Momentum 的結合，可視為一顆有摩擦力的球沿著斜坡運動。根據梯度的一階矩估計（均值）和二階矩估計（無中心的方差）來調整學習率。

梯度的一階矩估計（梯度平均）：這是對過去梯度的平均值的估計。一階矩估計告訴我們梯度的大致方向和變化趨勢，從而幫助我們了解參數更新的方向。

梯度的二階矩估計（梯度平方的平均）：這是對過去梯度平方的平均值的估計，也就是梯度的變異性。二階矩估計能夠反映梯度的變化大小，幫助我們判斷學習率是否應該增大或減小。

另外當β_1和β_2接近 1 時，m_t和v_t會接近 0 所以做出以下修正

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t}$$

$$\hat{v}_t = \frac{v_t}{1-\beta_2^t}$$

在上面兩式的基礎更新 x 得:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \varepsilon} \hat{m}_t$$

梯度驗證

為了避免出現梯度計算錯誤，除了調整學習率，也應該檢查梯度的計算的正確。

數值梯度:利用定義求導:$\frac{\partial f(x)}{\partial x} = \lim_{\epsilon \to 0} \frac{f(x+\epsilon)-f(x-\epsilon)}{2\epsilon}$

分析梯度:直接利用公式每個參數求導。
若兩者相差過大，應該檢查是否出錯。

通用的數值梯度

在機器學習中的假設函數參數通場很多，所以可以編寫一個通用的數值梯度計算函數，程式碼請見附檔。

optimizer 和能夠更新的梯度下降法

optimizer 裡除了參數更新方式不同，其他梯度下降的框架是雷同的，以下會時做一個 optimizer 和能夠更新參數的 optimizer 詳細程式碼請見附檔。