➢ trainingOptions 設置訓練選項

```
options = trainingOptions('sgdm',... %優化器為具有動量的隨機梯度下降
    'MaxEpochs',5,... %最大訓練回合數
    'Verbose',true,... %是否顯示訓練資訊
     'VerboseFrequency',60,...
     'MiniBatchSize',256,... %每次迭代使用的數據量
     'Shuffle',every-epoch,...
    'Plots','training-progress',... %是否畫出訓練過程
    "ExecutionEnvironment","cpu");
```

訓練網路

```
net = trainNetwork(XTrain,YTrain,layers,options);
```

➢ 訓練網絡的求解器，指定為以下之一：

'sgdm' - 使用帶動量的隨機梯度下降 (SGDM) 優化器。 您可以使用 'Momentum' 名稱-值對組參數指定動量值。

'rmsprop'— 使用 RMSProp 優化器。 您可以使用 'SquaredGradientDecayFactor' 名稱-值對組參數指定平方梯度移動平均值的衰減率。

'adam'——使用 Adam 優化器。 您可以分別使用 'GradientDecayFactor' 和 'SquaredGradientDecayFactor' 名稱-值對組參數指定梯度和平方梯度移動平均值的衰減率。

➢ 常用的設定參數很多，下面進行詳解

- solverName：優化函數，可選'sgdm'，'rmsprop'，'adam'
- Momentum：動量，[0,1]之間
- MaxEpochs：最大訓練回合數，正整數，默認為 20
- MiniBatchSize：就是 batchsize，每次迭代使用的數據量，正整數
- Verbose：是否在命令行窗口顯示實時訓練進程，0 或 1，若為 1，則在命令行顯示當前在幹啥了，默認為 true
- Shuffle：數據打亂策略，可選'once'，'never'，'every-epoch'
  'once'：在訓練前打亂
  'never'：不打亂
  'every-epoch'：每個 epoch 打亂一次
  默認為'once'，建議選擇'every-epoch'，因為 MATLAB 訓練網絡的時候，如果數據不夠一個 batchsize 會直接丟棄，'every-epoch'可以避免丟棄同一批

數據

- ExecutionEnvironment：硬件環境，用 GPU 還是 CPU，可選'auto'，'cpu'，'gpu'，'multi-gpu'，'auto'為有 gpu 則用，沒有就用 cpu

➢ 學習率設定

```
options = trainingOptions('sgdm', ...
    'LearnRateSchedule','piecewise', ...
   'InitialLearnRate',0.0001, ...
    'LearnRateDropFactor',0.2, ...
    'LearnRateDropPeriod',5, ...
    'MaxEpochs',20, ...
    'MiniBatchSize',64, ...
    'Plots','training-progress')
```

- InitialLearnRate：初始學習率
- LearnRateSchedule：學習率策略，'none'或者'piecewise'，'none'表示學習率不變，'piecewise'為分段學習率
- LearnRateDropFactor：學習率下降因子，[0,1]之間，降低之後學習率為：當前學習率*下降因子
- LearnRateDropPeriod：學習率下降週期，即幾個 epoch 下降一次學習率

➢ 早停條件

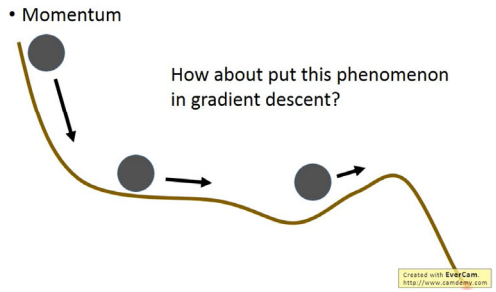
```
options = trainingOptions('sgdm',... %優化器為具有動量的隨機梯度下降
    'MaxEpochs',50,... %最大訓練回合數
     'MiniBatchSize',256,... %每次迭代使用的數據量
     'ValidationFrequency',128,...
     'ValidationPatience',5,...
    'Plots','training-progress',... %是否畫出訓練過程
```

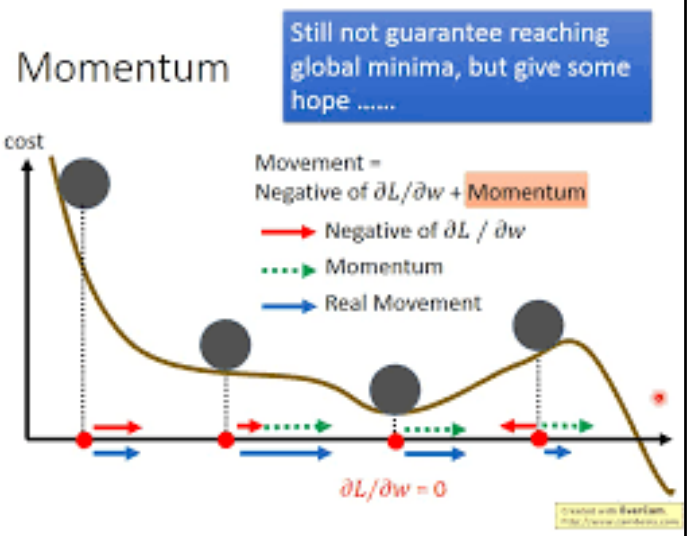
- VerboseFrequency：Verbose 在命令行打印的頻率，默認為 100
- ValidationData：驗證集數據，是一個 ImageDatastore 對象
- ValidationFrequency：驗證頻率，幾個 batchsize 後驗證一次，不是 epoch
- ValidationPatience：早停條件，Validation 上 loss 大於或等於最小 loss 多少 epoch 後停止訓練，比如，當前 loss 為最小值且為 0.01，再經過幾個回合都沒有低於 0.01，就停止訓練
- CheckpointPath：網絡保存路徑，默認為''，即默認不保存，如果設置的有路徑，則每個 epoch 後會非覆蓋的保存一次

- **Momentum**

它的原理是，在真實世界中，如果下圖是一個山坡，當你把一個球丟下去，我們知道它滾到plateau的地方，因為有慣性，所以它不會停下來，而就算是走到上坡的地方，只要這個坡沒有很陡，因為慣性它可能還是可以翻過這個山坡，那它就可以找到比這個local minimum還要好的地方。

In physical world .....





Adagrad

$$g^t = \frac{\partial C(\theta^t)}{\partial w} \qquad \eta^t = \frac{\eta}{\sqrt{t+1}}$$

- 將每個參數的學習率除以其先前導數的均方根

Vanilla Gradient descent

$$w^{t+1} \leftarrow w^t - \eta^t g^t$$

w is one parameters

Adagrad

$$w^{t+1} \leftarrow w^t - \frac{\eta^t}{\sigma^t} g^t$$

σ^t: ***root mean square*** of the previous derivatives of parameter w

Parameter dependent

Adagrad

σ^t: ***root mean square*** of the previous derivatives of parameter w

$$w^1 \leftarrow w^0 - \frac{\eta^0}{\sigma^0} g^0 \qquad \sigma^0 = g^0$$

$$w^2 \leftarrow w^1 - \frac{\eta^1}{\sigma^1} g^1 \qquad \sigma^1 = \sqrt{\frac{1}{2}\left[(g^0)^2 + (g^1)^2\right]}$$

$$w^3 \leftarrow w^2 - \frac{\eta^2}{\sigma^2} g^2 \qquad \sigma^2 = \sqrt{\frac{1}{3}\left[(g^0)^2 + (g^1)^2 + (g^2)^2\right]}$$

$$\vdots$$

$$w^{t+1} \leftarrow w^t - \frac{\eta^t}{\sigma^t} g^t \qquad \sigma^t = \sqrt{\frac{1}{t+1}\sum_{i=0}^{t} (g^i)^2}$$