ML_HW2 0751231 曾揚

(a)

ML - HW2 0751231 曾揚

$$f(x) = f(x+T) = \frac{A_0}{2} + \sum_{n=1}^{N} \left(a_n \cos \frac{2\pi n}{T} x + b_n \sin \frac{2\pi n}{T} x\right)$$

$$\hat{f}(x) = a_0 + \sum_{n=1}^{N} \left(\hat{a}_n \cos \frac{2\pi n}{T} x(k) + \hat{b}_n \sin \frac{2\pi n}{T} x(k)\right)$$

$$E(k) = \frac{1}{2}\left(f(x(k)) - \hat{f}(x(k))\right)^2 \cdots \text{pattern learning}$$

$$E = \frac{1}{2}\sum_{k=1}^{M}\left(f(x(k)) - \hat{f}(x(k))\right)^2 \cdots \text{batch learning}$$

$$w \leftarrow w - \alpha \frac{\partial E(\cdot)}{\partial w}$$

(a) pattern learning.

$$\frac{\partial E(\cdot)}{\partial a_n} = \left(f(x(k)) - \hat{f}(x(k))\right) \frac{\partial}{\partial a_n}\left(f(x(k)) - \hat{f}(x(k))\right) = \left(f(x(k)) - \hat{f}(x(k))\right) \cdot \left(-\cos \frac{2\pi n}{T} x(k)\right)$$

$$\frac{\partial E(\cdot)}{\partial b_n} = \left(f(x(k)) - \hat{f}(x(k))\right) \cdot \left(-\sin \frac{2\pi n}{T} x(k)\right)$$

$$\Rightarrow \begin{cases} \hat{a}_n \leftarrow \hat{a}_n + \alpha\left(f(x(k)) - \hat{f}(x(k))\right)\left(\cos \frac{2\pi n}{T} x(k)\right) \\ \hat{b}_n \leftarrow \hat{b}_n + \alpha\left(f(x(k)) - \hat{f}(x(k))\right)\left(\sin \frac{2\pi n}{T} x(k)\right) \end{cases}$$

batch learning.

$$\Rightarrow \begin{cases} \hat{a}_n \leftarrow \hat{a}_n + \alpha \sum_{k=1}^{M}\left(f(x(k)) - \hat{f}(x(k))\right)\left(\cos \frac{2\pi n}{T} x(k)\right) \\ \hat{b}_n \leftarrow \hat{b}_n + \alpha \sum_{k=1}^{M}\left(f(x(k)) - \hat{f}(x(k))\right)\left(\sin \frac{2\pi n}{T} x(k)\right) \end{cases}$$

(b)

使用之函數為 $f(x) = f(x+2\pi)$

$$f(x) = \begin{cases} 1 & 0 < x < \pi \\ -1 & \pi < x < 2\pi \end{cases}$$

(b)

選擇使用之目標函數如上圖所示,

模型參數 ： T = 2pi , N = 10 , alpha = 0.1 , 訓練次數 ：100000

結果如下 ：

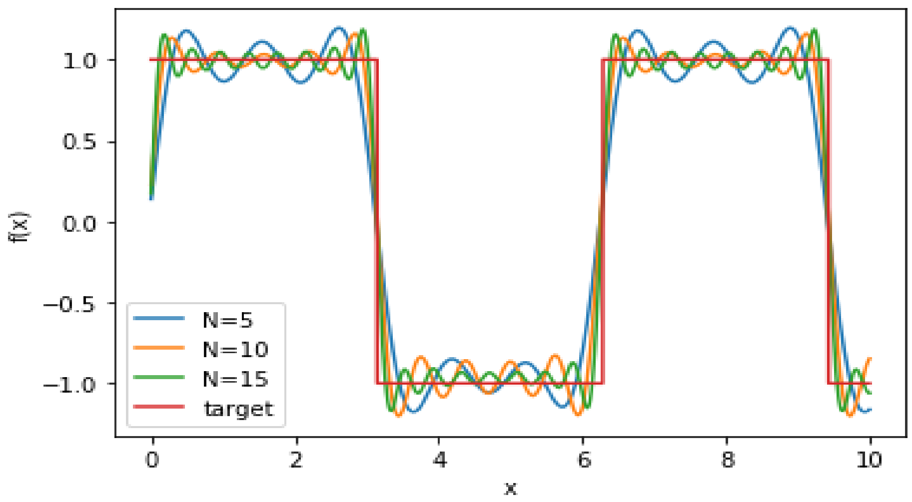
	an_true	bn_true	an_approx	bn_approx
n = 0	0	0	0.0173107	0
n = 1	0	1.2732395	0.0190419	1.2499669
n = 2	0	0	0.009509	-0.003824
n = 3	0	0.4244132	0.0173198	0.4199772
n = 4	0	0	0.0126895	-0.003738

n = 5	0	0.2546479	0.0166944	0.2517077
n = 6	0	0	0.0145895	-0.003912
n = 7	0	0.1818914	0.0181632	0.178649
n = 8	0	0	0.0154206	-0.005572
n = 9	0	0.1414711	0.0200166	0.1439192
n = 10	0	0	0.0344834	-0.014282

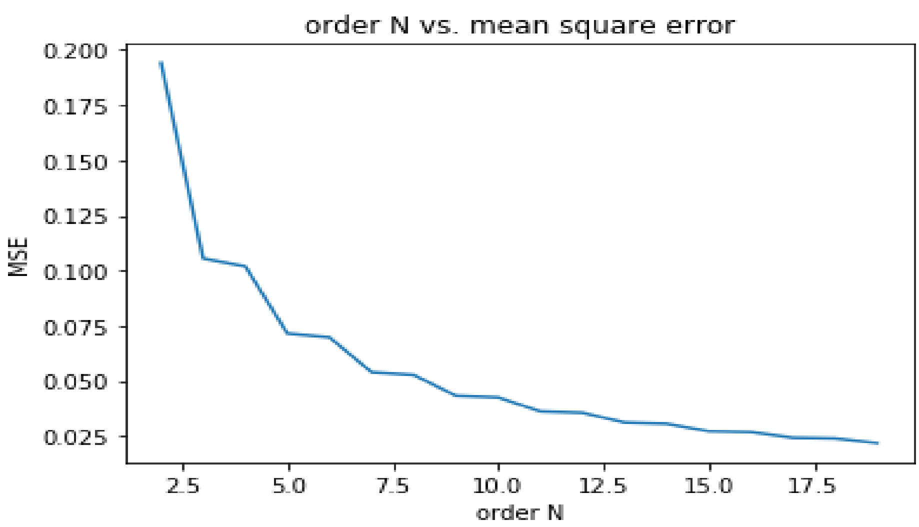
MSE = 0.042491362269754474

(c)

根據程式跑出來的結果，當 N 逐漸變大時，傅立葉級數能夠更加貼近目標函數，使得 MSE 逐漸縮小。下圖為當使用不同的 N 作為模型參數時，x 與 f(x)的變化。



下圖為當 order N 逐漸變大時，Mean Square Error 的變化。



(d)

在使用 N = 10 且 訓練次數維持在 100000 次的情況下我們可以大略得觀察到，當 learning rate 越小時，可以得到越小得 MSE，因其在 MSE curve 接近 Local min 時，較不容易因為過大的更新值而回彈至較高得 MSE，導致較崎嶇的變化，如上圖 leaning rate>0.06 時。但當 learning rate 足夠小時，則沒有 learning rate 對於 MSE 就沒有過大的影響。

(e)

就執行結果而言，batch learning 在訓練時需要以較 pattern learning 小的 learning rate 做訓練，否則也易出現如(d)所說之 MSE 不容易靠近 local minmum 的現象。單就用來更新參數的數學式而言，亦是如此在 batch learning 更新參數時有個 sigma 項，該 sigma 的值會隨著 batch size 提升而跟著提升，故 batch learning 若要與 pattern learning 有差不多的收斂速度，則需要使用較小的 learning rate 做訓練。

(f)

在 gradient descent method 中加入 momentum，相較於沒有加入 momentum 得方法，能使得收斂到 Local min 附近的速度更快速。