ศุภเดช บริสุทธนารักษ์ 600610783

HOMEWORK 1

Multilayer Perceptron (MLP)

ข้อ 1 Neural Network สำหรับการทำนายระดับน้ำที่สะพานนวรัตน์ในอีก 7 ชั่วโมงข้างหน้า

วิธีการทำงาน

1. นำเข้าข้อมูลระดับน้ำที่สะพานนวรัตน์ของแต่ละสถานี และทำ 10% Cross validation โดยแบ่งชุดข้อมูลออกเป็น 10 blocks และให้ 1 block เป็น Test dataset และ 9 blocks ที่เหลือเป็น Train dataset และทำการ train ทั้งหมด 10 รอบ โดยสลับให้ block ถัดไปเป็น Test dataset
2. นำ Train Dataset ไปเข้าสู่กระบวนการ Train ใน Multilayer Perceptron โดยมีขั้นตอนต่อไปนี้
3. กำหนดโครงสร้างของ Neural Network โดยกำหนดจำนวน node ภายใน hidden layer จากนั้นทำการ Initialize Weight and Bias และทำ Feed Forward ดังนี้
    - *Input layer:* กำหนดเวกเตอร์ X ใช้สำหรับเก็บ feature ที่เก็บค่าระดับน้ำที่วัดได้ในชั่วโมงปัจจุบัน 1 ชั่วโมงที่แล้ว 2 ชั่วโมงที่แล้ว และ 3 ชั่วโมงที่แล้ว ของทั้งสองสถานีตามลำดับ ดังนั้น input layer จะมี node ทั้งหมด 8 nodes
    - *Hidden layer:*
        - กำหนดเวกเตอร์ V_i ทำหน้าที่เป็น hidden nodes ของ layer ที่ i รับค่า weight และ output ที่เข้ามาจาก layer ก่อนหน้า
        - กำหนดเวกเตอร์ W_i เป็นน้ำหนักของเส้นเชื่อมที่เชื่อมกับ node ใน layer ก่อนหน้า จากนั้นสุ่มค่า weight ขึ้นมาให้แต่ละเส้นเชื่อม
        - กำหนดเวกเตอร์ b_i เป็น offset โหนดใน layer ที่ i แล้วกำหนดให้เป็นเวกเตอร์ศูนย์ที่มีมิติเท่ากับ (จำนวนโหนดใน layer i, จำนวนโหนดใน layer i+1)
        - กำหนดเวกเตอร์ A_i เป็น output ที่ออกจากแต่ละ node ใน hidden layer และใช้ฟังก์ชัน sigmoid เป็น activation function จะได้ว่า $A_i = \sigma(V_i)$
        - ดังนั้น $V_i = W_i^T \cdot A_{i-1} + b_i$ และหาก i = 1 จะได้ว่า $V_1 = W_1^T \cdot X + b_1$
    - *Output layer:*
        - V_L ทำหน้าที่เป็น output nodes รับค่า weight และ output ที่เข้ามาจาก hidden layer ชั้นท้ายสุด จะได้ว่า $V_L = W_L^T \cdot A_{L-1} + b_L$
        - A_L ใช้ sigmoid เป็น activation function จะได้ว่า $A_L = \sigma(V_L)$
4. คำนวณหาค่า error (E) ที่ได้จากกระบวนการ Feed Forward โดยนำ desired output (D) ลบกับผลลัพธ์ที่ได้ (A_L) แล้วนำ error ที่ได้มาหา sum square error = $\frac{1}{2} \cdot Sum(E^2)$

5. ทำ Back Propagation เพื่อคำนวณหาค่า Local Gradient (LG) และ Local Gradient Bias (LGB) ของทุกๆ node ใน hidden และ output layer
    - สำหรับ node ใดๆ ใน output layer หาได้จาก
        - $LG_L = \left(E * \frac{\partial A_L}{\partial V_L}\right) * X$ (ให้คูณกันแบบ element-wise คือนำค่าในสมาชิกแต่ละตัวมาคูณกัน)
        - $LGB_L = E * \frac{\partial A_L}{\partial V_L}$
        - และ $\frac{\partial A_L}{\partial V_L} = \sigma(V_L)(1 - \sigma(V_L))$
    - สำหรับ node ใดๆ ใน hidden layer หาได้จาก
        - $LG_l = (LG_L^T \cdot W_{l+1})^T * \frac{\partial A_l}{\partial V_l}$ (ให้คูณกันแบบ element-wise)
        - $LGB_l = (LGB_L^T \cdot W_{l+1})^T * \frac{\partial A_l}{\partial V_l}$
        - และ $\frac{\partial A_l}{\partial V_l} = \sigma(V_l)(1 - \sigma(V_l))$
6. กำหนด learning rate (η) และ momentum rate (α) จากนั้น update weight และ bias ของทุกๆ ตัว ซึ่งหาได้จากสมการ $W(n+1) = W(n) + \Delta W(n)$ และ $b(n+1) = b(n) + \Delta b(n)$
    - ΔW และ Δb ที่ output layer หาได้จาก
        - $\Delta W_L(n) = \alpha \Delta W_L(n-1) + \eta(LG_L \cdot A_{L-1}^T)$
        - $\Delta b_L(n) = \alpha \Delta b_L(n-1) + \eta LGB_L$
    - ΔW และ Δb ที่ hidden layer หาได้จาก
        - $\Delta W_l(n) = \alpha \Delta W_l(n-1) + \eta\left(LG_l \cdot A_{l-1}^T\right)$
        - หาก l = 1 จะได้ $\Delta W_1(n) = \alpha \Delta W_1(n-1) + \eta(LG_1 \cdot X^T)$
        - $\Delta b_l(n) = \alpha \Delta b_l(n-1) + \eta LGB_l$
7. เมื่อ train ครบทุก examples แล้วจะนับเป็น 1 epoch ทำซ้ำเรื่อย ๆ จนกว่าจะถึง max epoch ที่กำหนดไว้ หรือ sum square error ที่ได้ใน epoch ปัจจุบันมีค่ามากกว่า sum square error ใน epoch ที่แล้ว
8. นำ Test dataset มาทดสอบกับ neural network นี้ ทำ Feed Forward เพื่อทำนายระดับน้ำที่ได้ แล้วหา Mean Square Error

ผลการทดลอง

การทดลองที่ 1 กำหนดให้ neural network นี้เป็นแบบ 8-4-1 ที่ initialize weight อยู่ในช่วง [-1, 1] มีไฮเพอร์พารามิเตอร์ ได้แก่ learning rate = 0.12 กับ momentum rate = 0.15 และทำการ train ทั้งสิ้น 3,000 epochs

Round	1	2	3	4	5	6	7	8	9	10
MSE	69.701	18.879	28.780	24.217	7.017	3.428	20.612	22.435	25.232	16.026
Average MSE		23.6332								

การทดลองที่ 2 กำหนดให้ neural network นี้เป็นแบบ 8-3-1 ที่ initialize weight อยู่ในช่วง [-1, 1] มีไฮเพอร์พารามิเตอร์ ได้แก่ learning rate = 0.12 กับ momentum rate = 0.15 และทำการ train ทั้งสิ้น 3,000 epochs

Round	1	2	3	4	5	6	7	8	9	10
MSE	66.408	20.243	28.878	24.200	6.993	3.384	20.230	22.017	25.232	16.302
Average MSE		23.3890								

การทดลองที่ 3 กำหนดให้ neural network นี้เป็นแบบ 8-3-1 ที่ initialize weight อยู่ในช่วง [-1, 1] มีไฮเพอร์พารามิเตอร์ ได้แก่ learning rate = 0.06 กับ momentum rate = 0.09 และทำการ train ทั้งสิ้น 3,000 epochs

Round	1	2	3	4	5	6	7	8	9	10
MSE	66.944	21.961	29.974	24.197	7.054	3.169	21.099	21.547	25.257	17.713
Average MSE		23.8920								

การทดลองที่ 4 กำหนดให้ neural network นี้เป็นแบบ 8-3-1 ที่ initialize weight อยู่ในช่วง [-3, 3] มีไฮเพอร์พารามิเตอร์ ได้แก่ learning rate = 0.12 กับ momentum rate = 0.15 และทำการ train ทั้งสิ้น 3,000 epochs

Round	1	2	3	4	5	6	7	8	9	10
MSE	66.849	21.225	28.817	24.210	6.714	3.342	20.170	22.382	26.030	16.660
Average MSE		23.6403								

สรุปผลการทดลอง

การผลการทดลองทั้ง 4 การทดลอง สรุปได้ว่าการลดจำนวน hidden nodes ให้เหมาะสมดังการทดลองที่ 2 จะให้ค่า error ที่ได้จาก Test Data Set ที่น้อยที่สุด ถ้าหากกำหนดจำนวน hidden nodes เยอะเกินไปจะส่งผลให้ประสิทธิภาพในการคำนวณช้าลง และถ้าน้อยเกินไปจะส่งผลให้มีค่า error มาก การปรับค่า hyperparameter ได้แก่ learning rate และ momentum rate ถ้าหากปรับให้มีค่ามากเกินไปค่า error จะกวัดแกว่งทำให้ algorithm นี้ไม่สามารถหาค่า error ที่น้อยที่สุดได้ ถ้าหากปรับให้มีค่าน้อยเกินไปค่า error ที่ได้เมื่อครบ 3,000 epoch จะยังไม่ใช่ค่า error ที่น้อยที่สุด การปรับช่วง weight เริ่มต้นเมื่อเทียบการทดลองที่ 2 และการทดลองที่ 4 จะได้ว่าเมื่อปรับช่วงจนเหมาะสมกับค่า weight ที่ดีที่สุดแล้ว ช่วงการกระจายตัวที่แคบกว่าจะให้ค่า error น้อยกว่าช่วงการกระจายตัวที่กว้างกว่า เนื่องจากช่วงที่แคบกว่าทำให้ขั้นตอน backpropagation สามารถปรับค่า weight ให้เป็นค่าที่ดีที่สุดได้เร็วขึ้น

ข้อ 2 ทดสอบกับ cross.pat

1. นำเข้าข้อมูลจากไฟล์ cross.pat แล้วทำ 10% Cross Validation แบ่งข้อมูลเป็น Train dataset และ Test dataset ด้วยวิธีการเดียวกันกับโจทย์ข้อที่ 1
2. นำ Train dataset มาเข้าสู่กระบวนการ train ใน multilayer perceptron ด้วยวิธีการเช่นเดียวกันกับโจทย์ข้อ 1
3. นำ Test dataset มาทดสอบแล้วนำผลลัพธ์ที่ได้จาก Feed Forward มาแบ่งคลาสด้วยวิธีการดังต่อไปนี้
    - ให้ j เป็นเมทริกซ์ที่บอกตำแหน่งที่มีค่ามากที่สุดในเวกเตอร์ prediction ที่ได้จากการทำ Feed forward
    - วนลูปใส่ค่าในเวกเตอร์ Y ทีละตัว
        - ถ้า j = 0 แล้ว Y[0][j] = 1, Y[1][j] = 0
        - ถ้า j = 1 แล้ว Y[0][j] = 0, Y[1][j] = 1
4. สร้างตาราง Confusion Matrix วนลูปเช็ค Y (Prediction) และ Desired output ทีละคู่
    - เพิ่มค่า True Positive (TP) เมื่อ Y = [1 0] และ Desired output = [1 0]
    - เพิ่มค่า True Negative (TN) เมื่อ Y = [0 1] และ Desired output = [0 1]
    - เพิ่มค่า False Positive (FP) เมื่อ Y = [1 0] และ Desired output = [0 1]
    - เพิ่มค่า False Negative (FN) เมื่อ Y = [0 1] และ Desired output = [1 0]
    - คำนวณหา $accuracy\ (\%) = \frac{(TP+TN)}{(TP+TN+FP+FN)} \times 100$

ผลการทดลอง

การทดลองที่ 1 กำหนดให้ neural network นี้เป็นแบบ 8-4-1 ที่ initialize weight อยู่ในช่วง [-1, 1] มีไฮเพอร์พารามิเตอร์ ได้แก่ learning rate = 0.12 กับ momentum rate = 0.15 และทำการ train ทั้งสิ้น 3,000 epochs

Round	1	2	3	4	5	6	7	8	9	10
MSE	0.00	0.15	0.25	0.55	0.25	0.25	0.30	0.10	0.45	0.00
Average MSE		**0.23**								

Confusion Matrix		Predicted		
		[1 0]	**[0 1]**	
Actual	**[1 0]**	TP = 90	FN = 10	TP + FN = 100
	[0 1]	FP = 36	TN = 64	FP + TN = 100
		TP + FP = 126	FN + TN = 74	Total = 200
Accuracy (%)		**77.00**		

การทดลองที่ 2 กำหนดให้ neural network นี้เป็นแบบ 8-3-1 ที่ initialize weight อยู่ในช่วง [-1, 1] มีไฮเพอร์พารามิเตอร์ ได้แก่ learning rate = 0.12 กับ momentum rate = 0.15 และทำการ train ทั้งสิ้น 3,000 epochs

Round	1	2	3	4	5	6	7	8	9	10
MSE	0.00	0.00	0.25	0.55	0.25	0.20	0.30	0.60	0.40	0.10
Average MSE		**0.265**								

Confusion Matrix		Predicted		
		[1 0]	**[0 1]**	
Actual	**[1 0]**	TP = 83	FN = 17	TP + FN = 100
	[0 1]	FP = 36	TN = 64	FP + TN = 100
		TP + FP = 119	FN + TN = 81	Total = 200
Accuracy (%)		**73.50**		

การทดลองที่ 3 กำหนดให้ neural network นี้เป็นแบบ 8-4-1 ที่ initialize weight อยู่ในช่วง [-1, 1] มีไฮเพอร์พารามิเตอร์ ได้แก่ learning rate = 0.06 กับ momentum rate = 0.09 และทำการ train ทั้งสิ้น 3,000 epochs

Round	1	2	3	4	5	6	7	8	9	10
MSE	0.00	0.15	0.25	0.55	0.25	0.20	0.25	0.10	0.45	0.05
Average MSE		**0.225**								

Confusion Matrix		Predicted		
		[1 0]	[0 1]	
Actual	[1 0]	TP = 91	FN = 9	TP + FN = 100
	[0 1]	FP = 36	TN = 64	FP + TN = 100
		TP + FP = 127	FN + TN = 73	Total = 200
Accuracy (%)		**77.50**		

การทดลองที่ 4 กำหนดให้ neural network นี้เป็นแบบ 8-4-1 ที่ initialize weight อยู่ในช่วง [-3, 3] มีไฮเพอร์พารามิเตอร์ ได้แก่ learning rate = 0.06 กับ momentum rate = 0.09 และทำการ train ทั้งสิ้น 3,000 epochs

Round	1	2	3	4	5	6	7	8	9	10
MSE	0.05	0.45	0.05	0.55	0.20	0.20	0.30	0.10	0.45	0.10
Average MSE		**0.245**								

Confusion Matrix		Predicted		
		[1 0]	[0 1]	
Actual	[1 0]	TP = 87	FN = 13	TP + FN = 100
	[0 1]	FP = 36	TN = 64	FP + TN = 100
		TP + FP = 123	FN + TN = 77	Total = 200
Accuracy (%)		**75.50**		

สรุปผลการทดลอง

จากผลการทดลองทั้ง 4 การทดลอง การทดลองที่ 3 จะให้ค่าความแม่นยำเมื่อทดสอบกับ Test Data Set มากที่สุด คือ 77.50% ผลกระทบจากการปรับจำนวน hidden nodes ค่า hyperparameters และช่วงของ weight เริ่มต้น ที่เกิดขึ้นเมื่อทดสอบกับ cross.pat เป็นเช่นเดียวกับข้อที่ 1 นั่นคือ mean square error มีค่าน้อยลง จะได้ค่า accuracy มากขึ้น

ภาคผนวก

Source code สำหรับการบ้านที่ 1

ข้อที่ 1 https://github.com/B2BeeBosz/IntroToCI/blob/master/hw1.py

ข้อที่ 2 https://github.com/B2BeeBosz/IntroToCI/blob/master/hw1_2.py