COS4312 Deep Learning

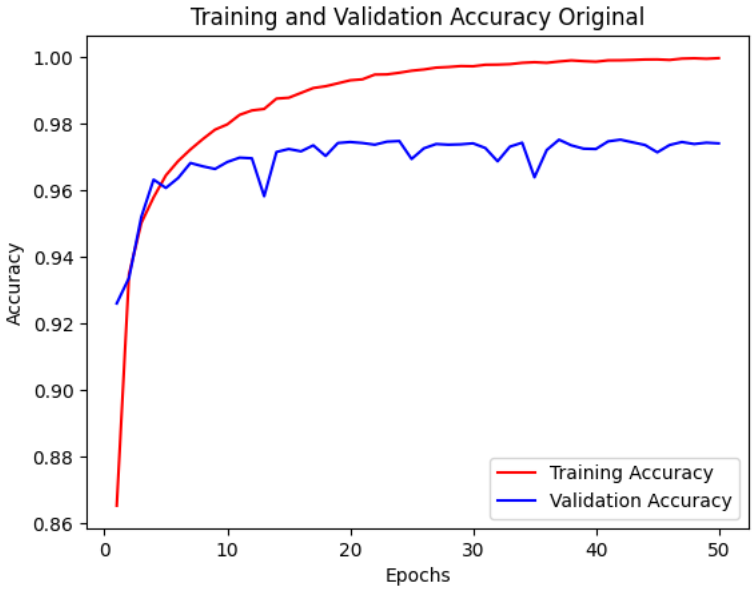
นายพลวัฒน์ ช้างโต รหัสนักศึกษา 6601001123

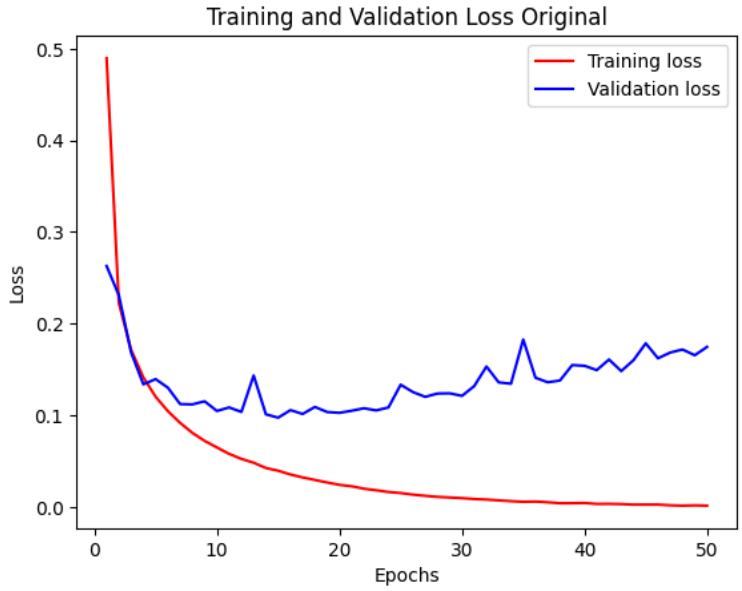
---

1. อธิบายเหตุผลว่าทำไมต้องแยก validation set ออกจาก training set

<u>ตอบ</u> เนื่องจาก model ที่เทรนด้วยข้อมูลไปเรื่อย ๆ นั้น อาจมีการเรียนรู้ที่จดจำกับ Training set มากจนเกินไป ในลักษณะที่ถ้า Model ไปเจอข้อมูลแปลกใหม่ จะไม่สามารถ Predict คำตอบที่ถูกต้องได้เลย หรือค่าที่ Predict ออกมานั้นเบี่ยงเบนออกไปจากคำตอบที่แท้จริงอย่างมาก เรียกสถานะของ Model นี้ว่า “Overfitting” ด้วยเหตุนี้ การที่เราแยก Validation set ออกมาจาก Training set นั้น ก็เพื่อทดสอบว่า Model มีความสามารถในการทำนายข้อมูลที่ไม่เคยเห็นมาก่อนได้หรือไม่ หรือ Model มีความ Generalize หรือไม่

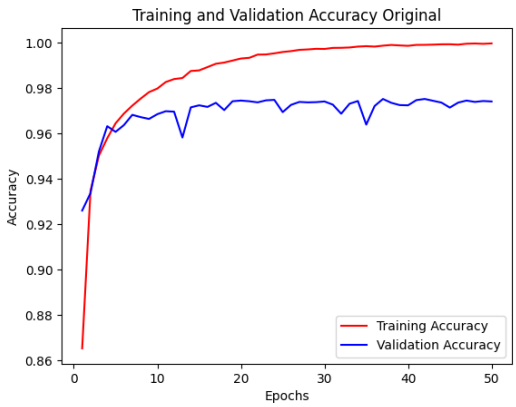
2. จากกราฟในข้อ 2. งานที่ต้องทำ เกิด underfitting overfitting หรือไหม ที่ไหน เลือก epoch ไหนดี

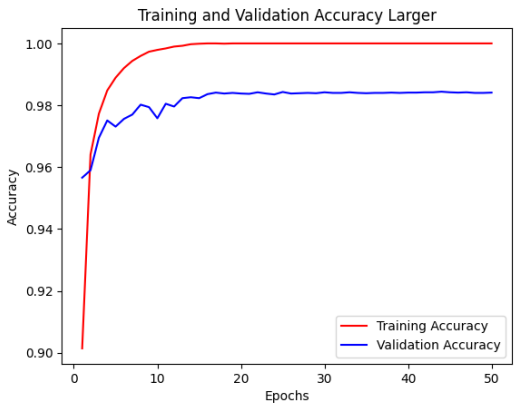


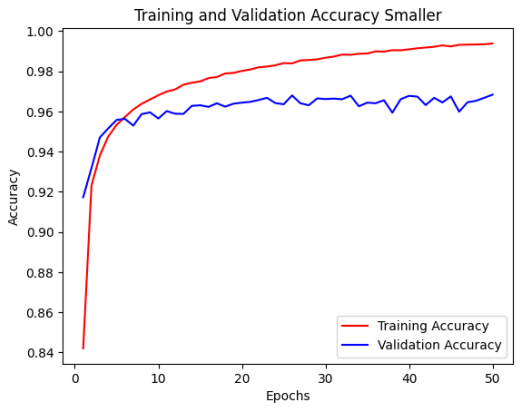


<u>ตอบ</u> การพิจารณาว่า Model เกิดสถานะ “Overfitting” หรือไม่นั้น พิจารณาจาก Validation Loss หรือ ค่า Loss ที่เกิดจากทดสอบด้วยข้อมูล Validation set โดยพิจารณาจากเส้นกราฟ หากแนวโน้มของเส้นกราฟ (ความชันของเส้นกราฟ) มีแนวโน้มยกตัวสูงขึ้นใน Epochs ใด ให้สันนิษฐานว่า Model มีแนวโน้มเริ่ม Overfitting แล้ว ด้วยเหตุนี้จึงควรเลือก Epochs ที่เส้นกราฟ Loss มีแนวโน้มความชันเข้าใกล้ 0 และต่ำมากที่สุด หากอ้างอิงจากหลักการข้างต้นแล้ว จึงควรเลือก Model Epochs ที่ 15

3. จากกราฟในข้อ 3. งานที่ต้องทำ จำนวน neuron เท่าใดเหมาะสมเทียบจาก acc







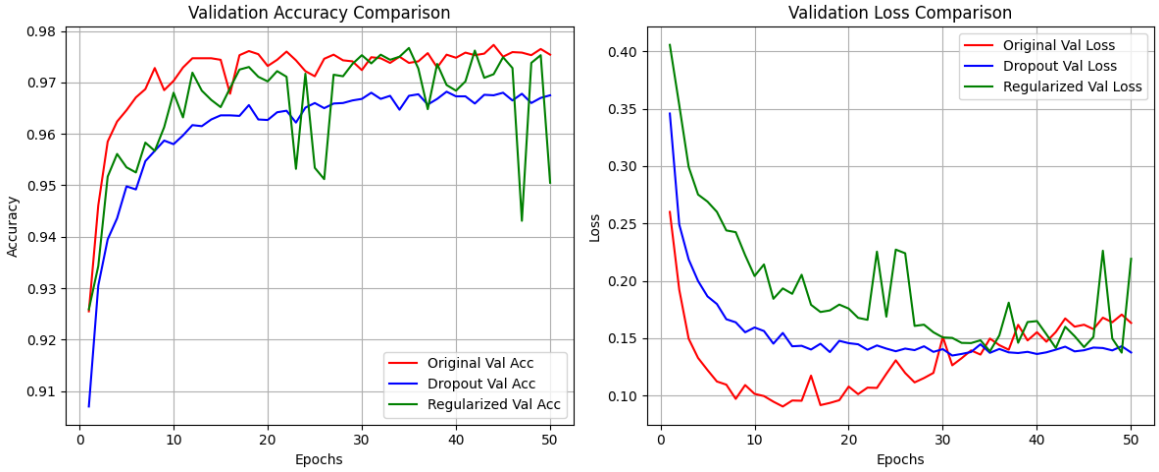
<u>ตอบ</u> จากกราฟ การเทรน Model Original ใช้ neuron 64

การเทรน Model Larger ใช้ neuron 512

การเทรน Model Smaller ใช้ neuron 32

อ้างอิงผลลัพธ์การเทรนจากกราฟข้างต้น สังเกตได้ว่า จากการเทรนจำนวน 50 Epochs นั้น Model Original ที่ใช้ neuron จำนวน 64 จะมี Accuracy ที่เป็นไปได้ดีที่สุด อยู่ในช่วง 0.96 – 0.975 เท่านั้น เช่นเดียวกันกับ Model Smaller ที่ใช้ neuron จำนวน 32 มี Accuracy ที่เป็นไปได้ดีที่สุดอยู่ในช่วง 0.95 – 0.96 กลับกัน Model Larger ที่ใช้ Neuron จำนวน 512 มีความเป็นไปได้ของ Accuracy ที่นิ่ง อยู่ที่ 0.98 ดีที่สุดใน 3 Model ที่เกิดขึ้น จึงเลือก Model Larger ที่ใช้ Neuron จำนวน 512 ในการเทรน

4. จากกราฟในข้อ 5. งานที่ต้องทำ Dropout, L2 regularization, ไม่มี regularization ตัวใดให้ผลดีกว่า



<u>ตอบ</u> อ้างอิงข้อมูลที่ปรากฏตามกราฟข้างต้น จากข้อมูลการเทรน Model โดยใช้ Model Original เป็น Model หลัก เทียบกับ Model Original + Dropout 0.5 และ Model Original + Regularized L2 0.001 นั้น พบว่า Model Original นั้นเริ่มเข้าสู่สถานะ Overfitting ในช่วง Epochs ที่ 17 – 19 (สังเกตจากเส้นสีแดง ในฝั่งกราฟ Validation Loss Comparison) และ Model ที่เทรนด้วย Regularized นั้น ถึงแม้จะมีแนวโน้มพัฒนาขึ้นเรื่อย ๆ และยังไม่เห็นถึงการเริ่ม Overfitting ในตลอด 50 Epochs แต่กราฟมีการแกว่งมาก (สังเกตจากเส้นสีเขียวในกราฟ Validation Accuracy Comparison และ Validation Loss Comparison) อาจพิจารณาได้ว่า การเทรนโมเดลลักษณะนี้ อาจทำให้ Model มีอัตราการเติบโตที่เสถียรต่ำ ต่อมาพิจารณาที่การเทรน Model แบบใช้ Dropout เพื่อลดการ Overfitting ของ Model พบว่า Model ที่เทรนในลักษณะนี้มีอัตราการ Overfitting ที่ต่ำมาก เนื่องจากเส้นกราฟสีน้ำเงินในกราฟ. Validation Loss Comparison ที่ไม่พบแนวโน้มที่เส้นกราฟจะพุ่งขึ้นเลย แต่ Model อาจมีประสิทธิภาพไม่เทียบเท่า Model Original พิจารณาได้จากการเส้นสีแดงของทั้งสองกราฟ ที่คะแนนทั้งจากฝั่ง Accuracy , Loss ก็ทำได้ดีกว่า Model with Dropout ในช่วงแรกเท่านั้น

จึงตอบได้ว่า Model Original นั้นอาจมีประสิทธิภาพที่ดีกว่า Model ที่ใช้ Regularized และ Dropout แต่ต้องเลือกใช้ Model ใน Epoch 13 – 17

5. Retrain เลือกโมเดลที่เหมาะสมที่สุดจากการทดลองทั้งหมด แล้วรายงาน accuracy ของ test set พร้อมเหตุผล

<u>ตอบ</u>

```
loss_ori , acc_ori = history_original.model.evaluate(test_x,test_y)
print(f"Loss : {loss_ori} , Acc : {acc_ori}")
✓ 0.1s

313/313 ━━━━━━━━━━━━━━━━━━━━ 0s 313us/step - accuracy: 0.9781 - loss: 0.1344
Loss : 0.13435569405555725 , Acc : 0.9781000018119812
```

```
loss_Large , acc_Large = history_larger.model.evaluate(test_x,test_y)
print(f"Loss : {loss_Large} , Acc : {acc_Large}")
✓ 0.2s

313/313 ━━━━━━━━━━━━━━━━━━━━ 0s 569us/step - accuracy: 0.9831 - loss: 0.0921
Loss : 0.0920991599559784 , Acc : 0.9830999970436096
```

```
loss_small , acc_small = history_smaller.model.evaluate(test_x,test_y)
print(f"Loss : {loss_small} , Acc : {acc_small}")
✓ 0.1s

313/313 ━━━━━━━━━━━━━━━━━━━━ 0s 301us/step - accuracy: 0.9681 - loss: 0.1346
Loss : 0.13463518023490906 , Acc : 0.9681000113487244
```

```
loss_dropout , acc_dropout = history_Dropout.model.evaluate(test_x,test_y)
print(f"Loss : {loss_dropout} , Acc : {acc_dropout}")
✓ 0.1s

313/313 ━━━━━━━━━━━━━━━━━━━━ 0s 397us/step - accuracy: 0.9638 - loss: 0.1512
Loss : 0.151175394654274 , Acc : 0.9638000130653381
```

```
loss_Regularized , acc_Regularized = history_Regularized.model.evaluate(test_x,test_y)
print(f"Loss : {loss_Regularized} , Acc : {acc_Regularized}")
✓ 0.1s

313/313 ━━━━━━━━━━━━━━━━━━━━ 0s 332us/step - accuracy: 0.9432 - loss: 0.2352
Loss : 0.23522987961769104 , Acc : 0.9431999921798706
```

จากการ Test นั้น Model Larger ทำแบบทดสอบออกมาได้ดีที่สุด โดยมีรายละเอียดการเทรน ดังนี้

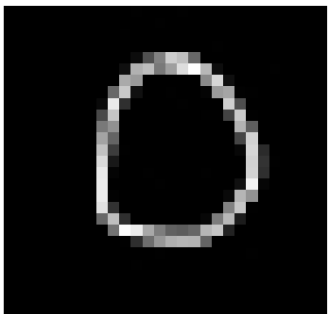
จำนวน Neuron : 512

Regularization : none

Epoch : 50

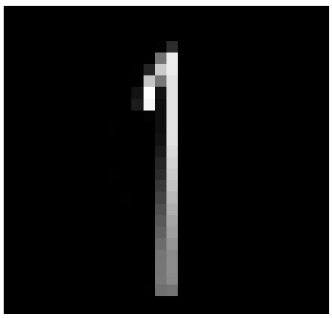
Acc of test : 0.983

6. วาดรูปคนละรูปรวม 3-4 รูป ตามจำนวนคนในกลุ่มแล้วทดสอบโมเดลที่ได้ทำนายถูกต้องไหม



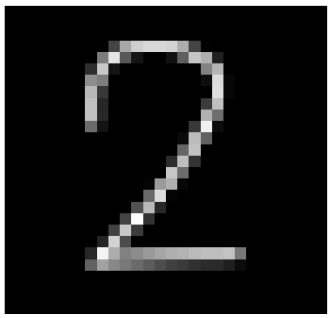
ผลการทำนาย: เลข 0

ความมั่นใจ: 74.57%



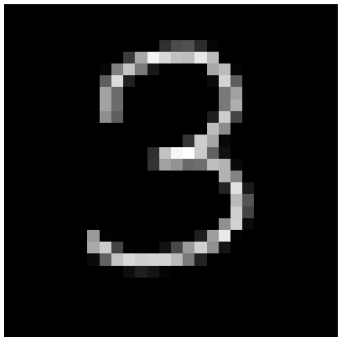
ผลการทำนาย: เลข 1

ความมั่นใจ: 99.94%



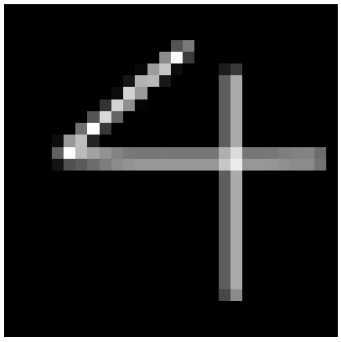
ผลการทำนาย: เลข 2

ความมั่นใจ: 87.31%



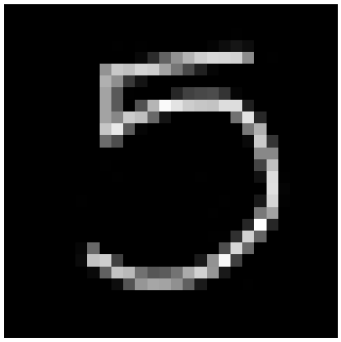
ผลการทำนาย: เลข 3

ความมั่นใจ: 100.00%



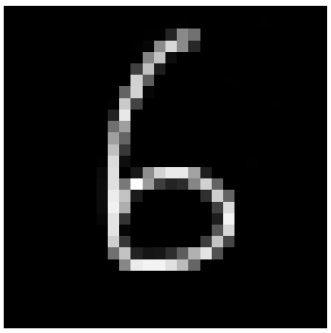
ผลการทำนาย: เลข 4

ความมั่นใจ: 75.92%



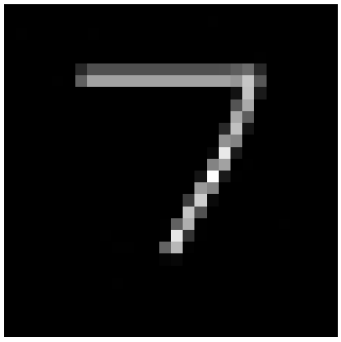
ผลการทำนาย: เลข 5

ความมั่นใจ: 99.78%



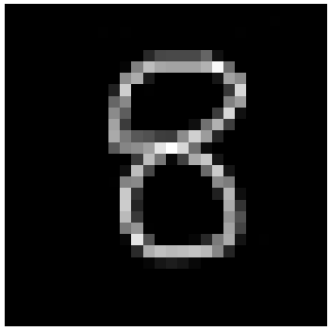
ผลการทำนาย: เลข 6

ความมั่นใจ: 99.99%



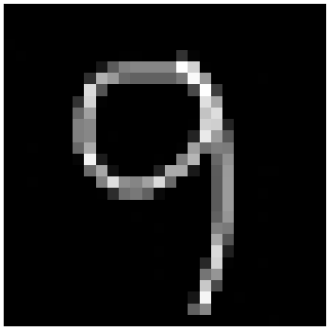
ผลการทำนาย: เลข 7

ความมั่นใจ: 75.48%



ผลการทำนาย: เลข 7

ความมั่นใจ: 75.48%



ผลการทำนาย: เลข 9

ความมั่นใจ: 99.99%