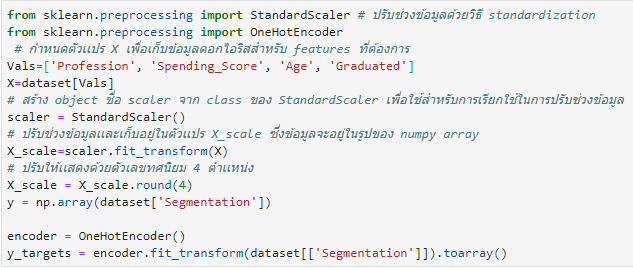
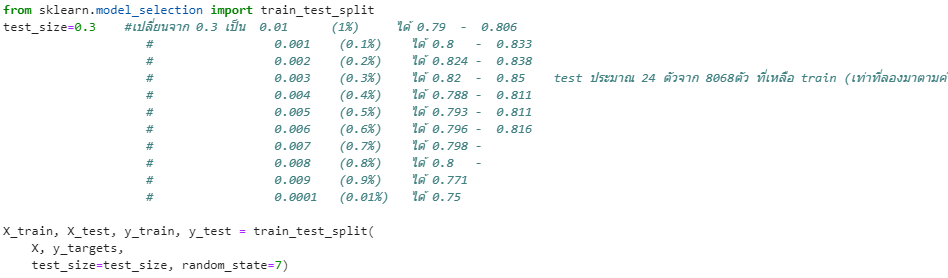
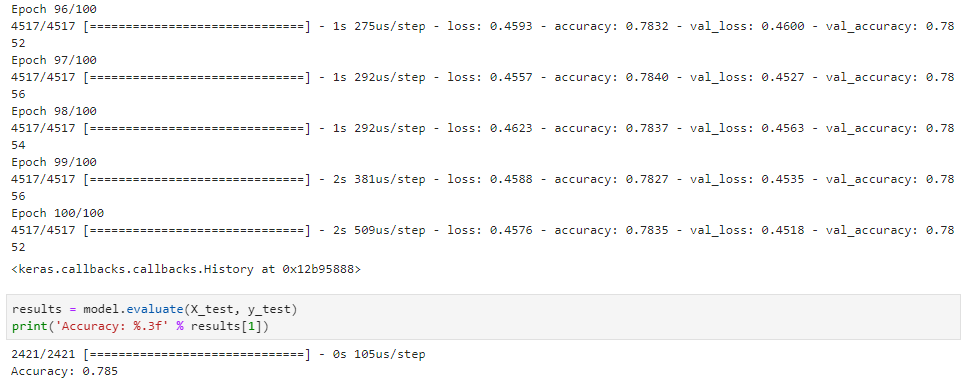
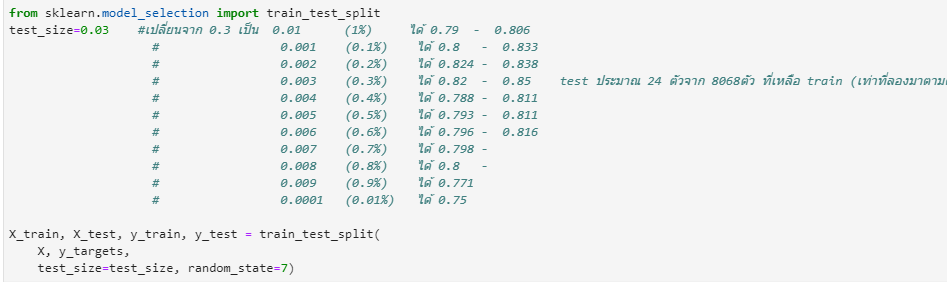
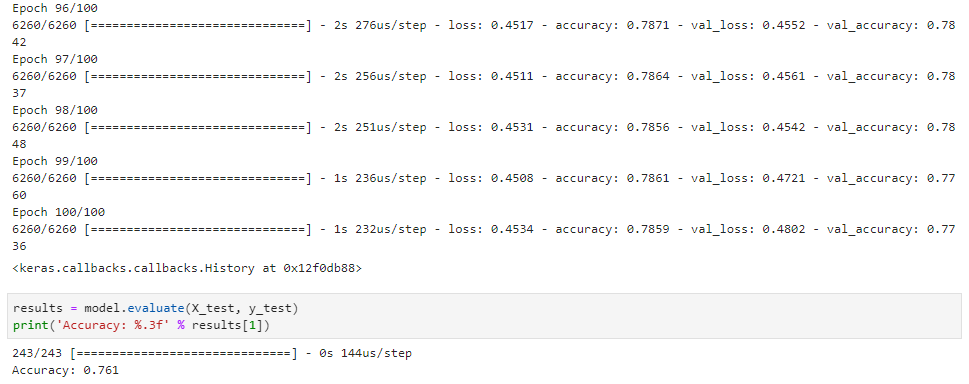
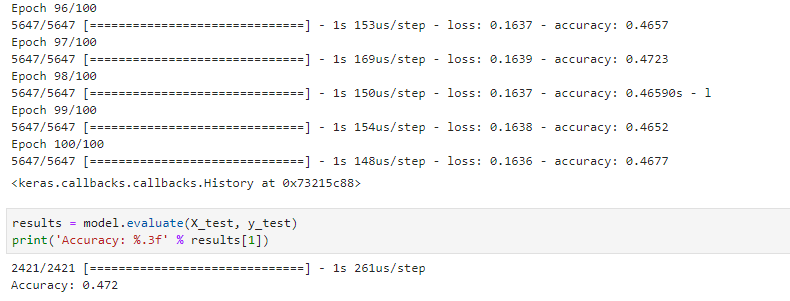
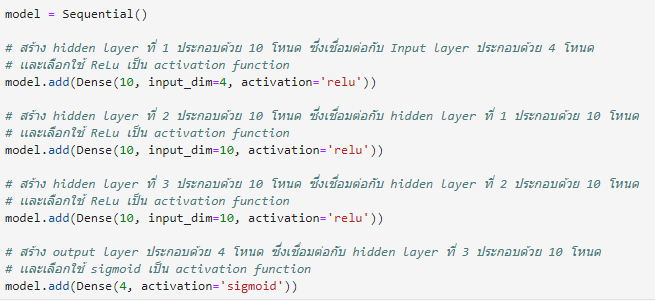
Feature ที่จะนำมาใช้มี 4 Feature ได้แก่ Profession, Spending\_Score, Age, Graduated ใช้ Feature ตัวเดียวกันกับในวิธี K-Nearest Neighbors ก่อนหน้า เพราะต้องการเปรียบเทียบความแม่นยำ (Accuracy ) ระหว่าง 2 วิธี

ในกรณีที่ใช้ Feature เหมือนกันในการพิจารณา ว่าวิธีไหนมีประสิทธิภาพในการให้ความแม่นยำได้มากกว่ากัน



เนื่องจากใช้ Feature 4 Feature นำมาพิจารณา จึงกำหนดให้ input\_dim=4

Class ของข้อมูลในที่นี้ถูกแบบด้วย Segmentation มีทั้งหมด 4 แบบ จึงกำหนดให้ output\_dim=4



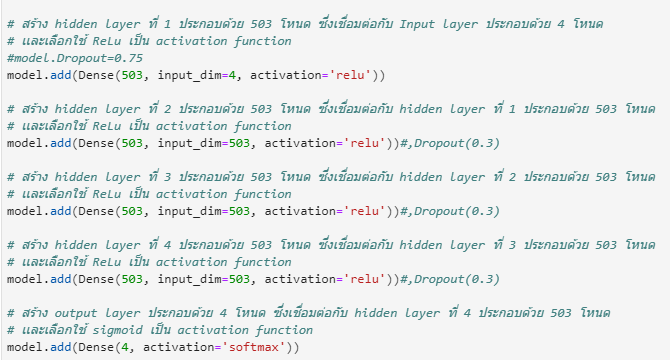
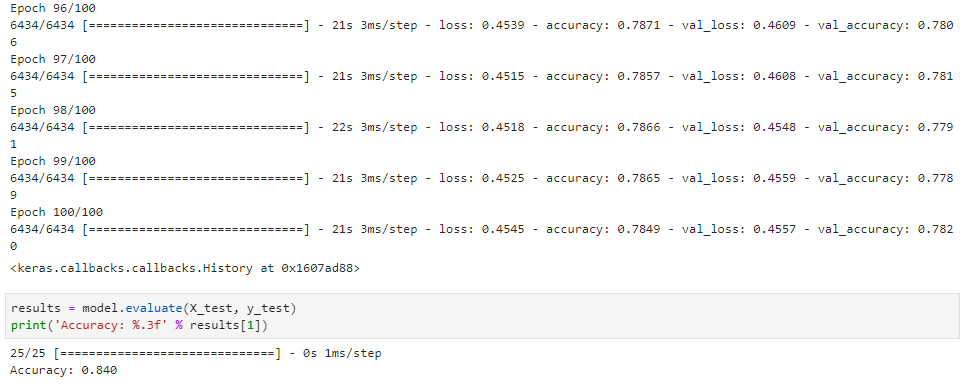
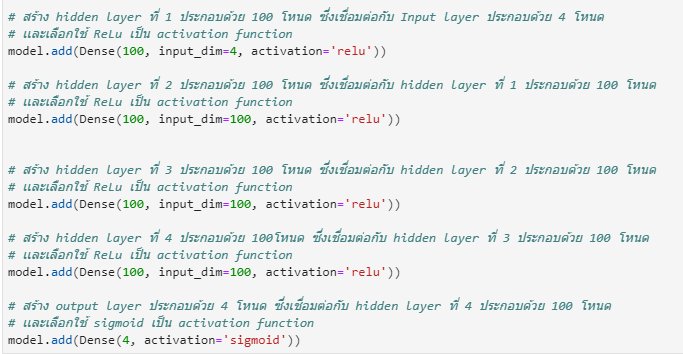
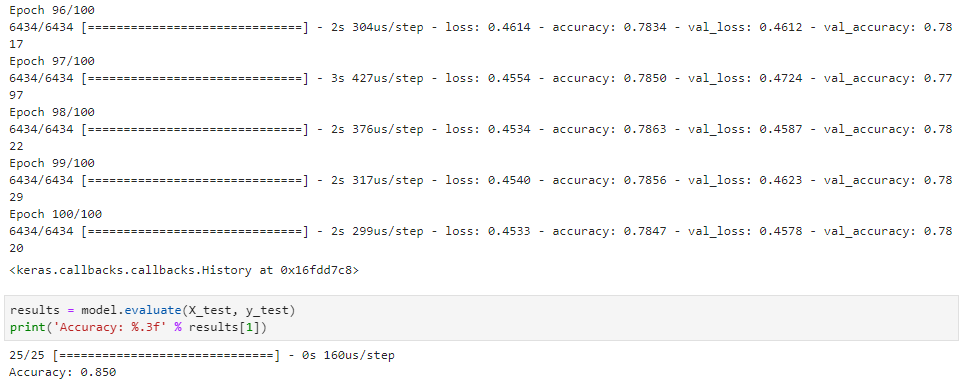
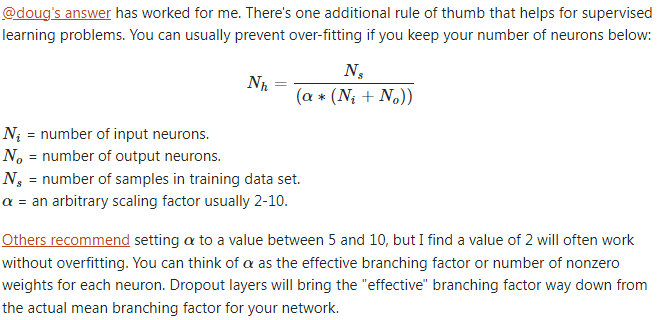
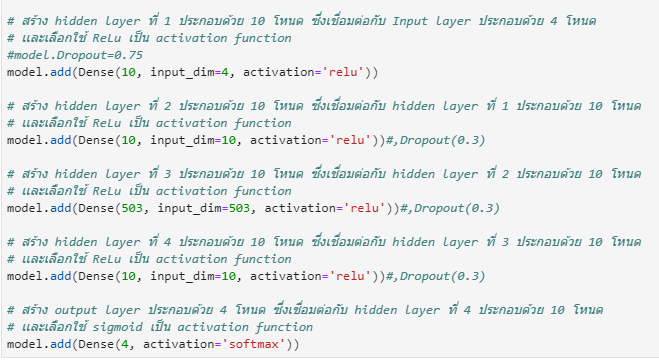
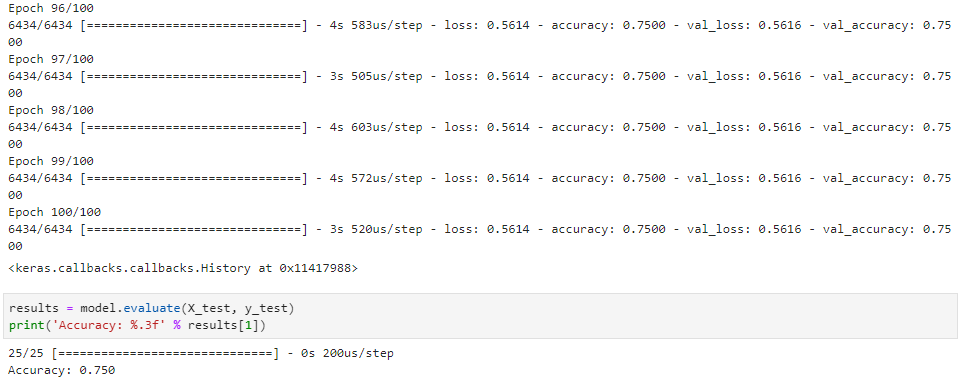
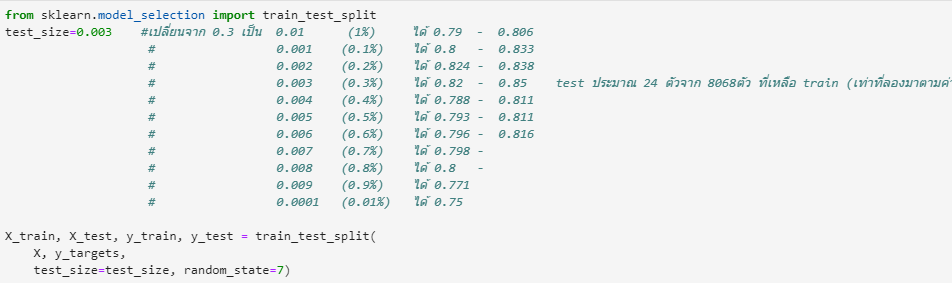
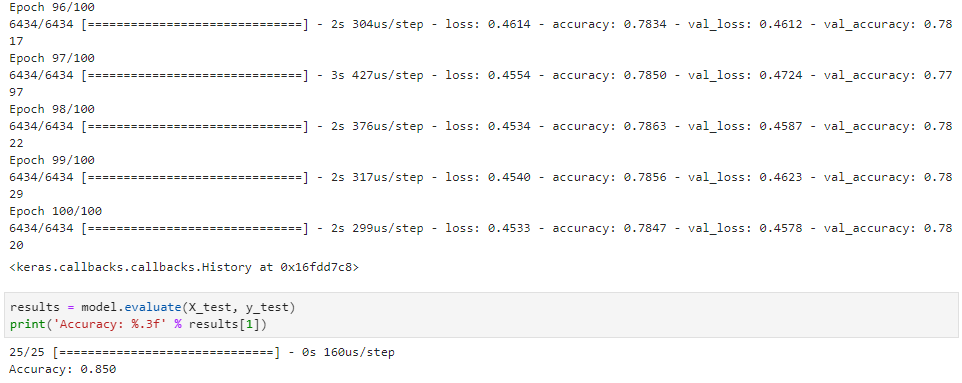
ค่าความแม่นยำที่ได้

test size เป็น

**Test size = 0.03**

**Test size = 0.3**

**ปรับเปลี่ยนขนาดค่า test size ที่ค่าต่างๆ**



ดังนั้นจะใช้ test size = 0.003 เพราะว่าให้ความแม่นยำที่สูง

**Test size = 0.003**

แหล่งข้อมูล :

<https://machinelearningmastery.com/choose-an-activation-function-for-deep-learning/> <https://sanparithmarukatat.medium.com/%E0%B8%AA%E0%B8%99%E0%B8%B8%E0%B8%81%E0%B8%81%E0%B8%B1%E0%B8%9A-neural-network-657fa293c4d1>

**Hidden neurons = 10**

**เลือกจำนวน hidden neurons**

**Hidden neurons = 100**

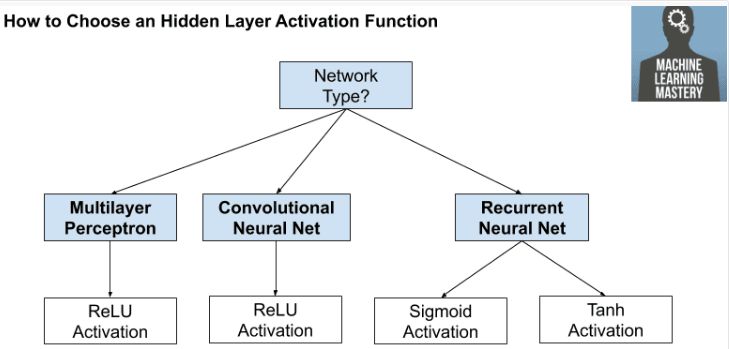
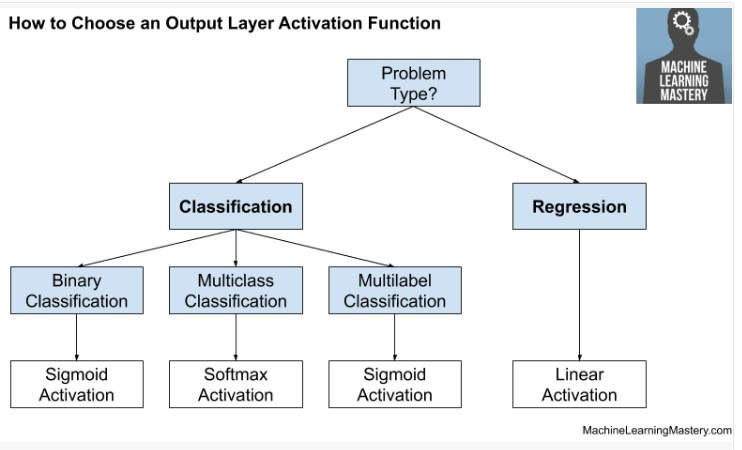
แหล่งข้อมูล : <https://stats.stackexchange.com/questions/181/how-to-choose-the-number-of-hidden-layers-and-nodes-in-a-feedforward-neural-netw>

**Hidden neurons = 503**

Activation mode จะขึ้นอยู่กับประเภทของ Network ที่ได้ออกแบบว่าเป็นชนิดไหน

โดยที่ activation mode จะกำหนดที่ชั้น Hidden layer และ Output layer

**เลือก Activation mode ให้แต่ละ Layer**

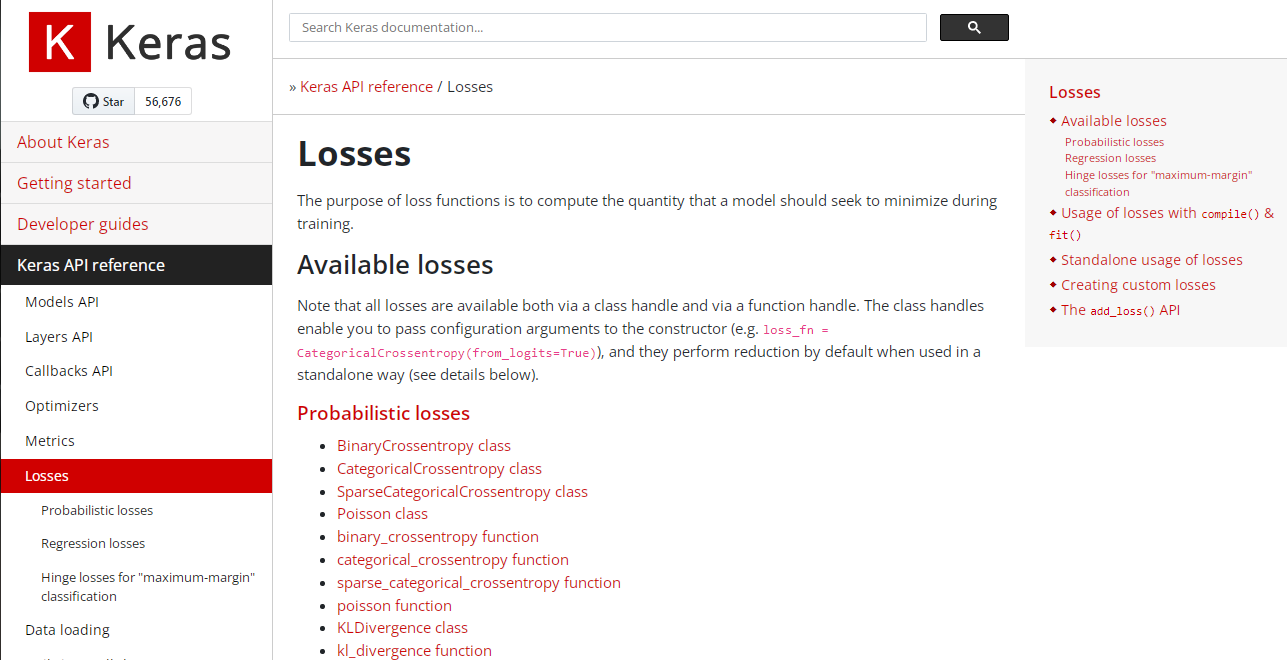
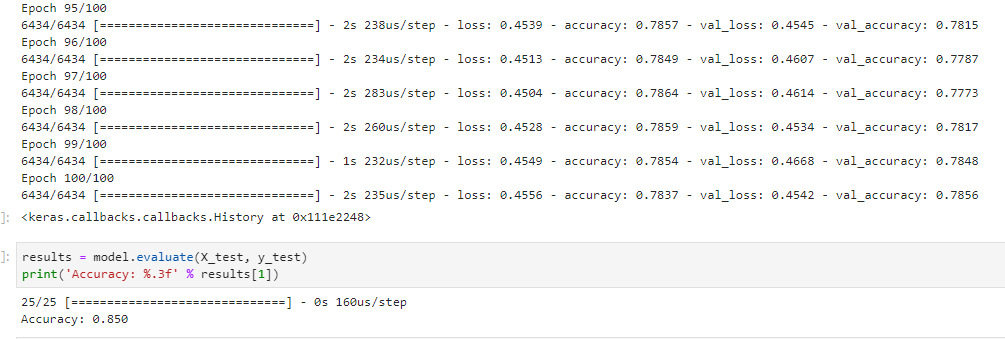
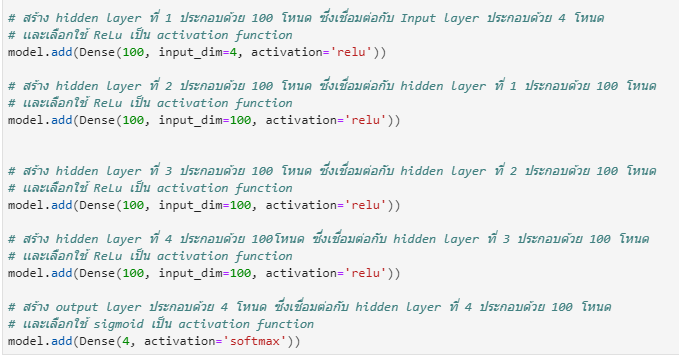
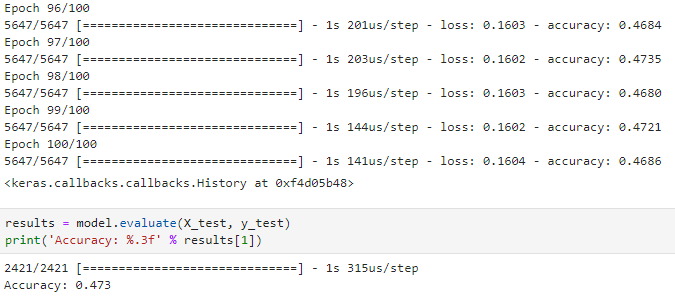


เนื่องจากระบบออกแบบเพื่อทำการ Classification Segmentation ซึ่งเป็น Multiclass จึงใช้ Softmax Activation

Activation mode สำหรับ Hidden layer และ Output layer

Activation mode สำหรับ Output layer

เนื่องจากออกแบบให้มีหลาย Hidden layer จึงใช้ Relu Activation



ค่าความแม่นยำที่ได้

แหล่งข้อมูล :

<https://machinelearningmastery.com/choose-an-activation-function-for-deep-learning/> <https://sanparithmarukatat.medium.com/%E0%B8%AA%E0%B8%99%E0%B8%B8%E0%B8%81%E0%B8%81%E0%B8%B1%E0%B8%9A-neural-network-657fa293c4d1>

Binary Crossentropy class คือ ค่าเฉลี่ยของ Cross-Entropy ที่เกิดจากการแจกแจงความน่าจะเป็น 2 แบบ คือ การแจกแจงความน่าจะเป็นที่เราอยากได้ (Actual) กับการแจกแจงความน่าจะเป็นที่ถูกประมาณโดย Model (Predicted)

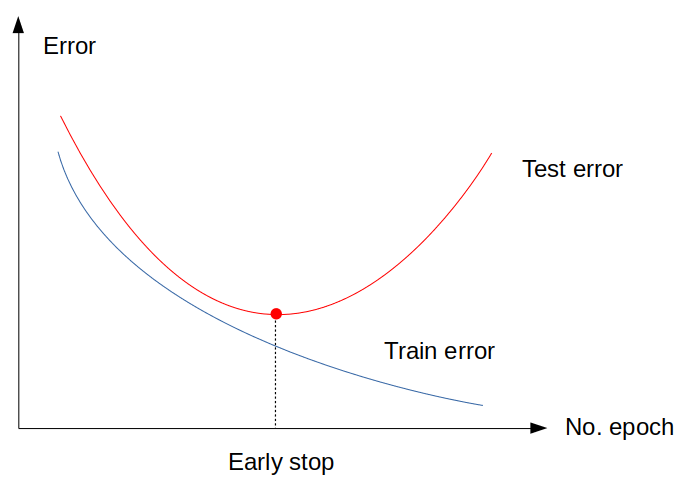
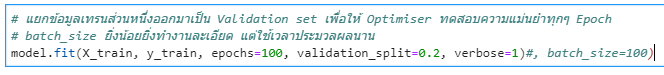
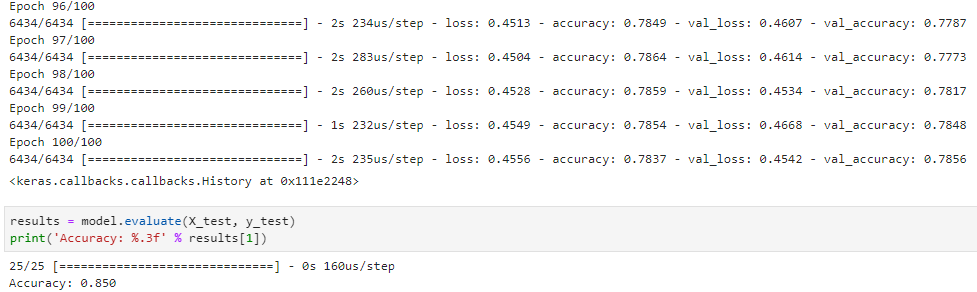
ใช้ Binary Crossentropy class

ทำการเปลี่ยน loss function เพื่อหา loss function ที่เหมาะสม และ ให้ความแม่นยำออกมาได้มากที่สุด

ในที่นี้จะใช้ Probabilistic Loss เพื่อหาความแม่นยำที่ได้จากการพิจารณาความน่าจะเป็น

**เปลี่ยน loss function**

**ลองใช้ class อื่นๆ**



แหล่งข้อมูล :

<https://keras.io/api/losses/>

<https://blog.pjjop.org/loss-functions-for-training-deep-learning-model-part2/>

ค่าความแม่นยำที่ได้

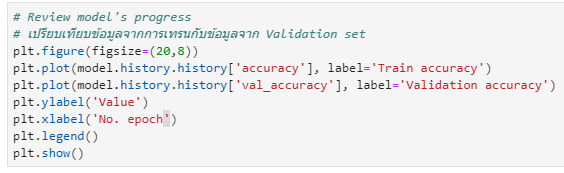
ในที่นี้จะลองกำหนดให้ขนาด Validation set มีขนาดเท่ากับ 0.2(20%) ของข้อมูลที่ถูกเทรน

ตามกราฟ ปกติ Optimizer จะเทรนเพื่อให้ได้ Train error ต่ำที่สุด แต่บางครั้ง Test error จะเริ่มวกกลับไปมีค่าสูงขึ้นก่อนที่จะพบจุดต่ำสุดของ Cost function ดังนั้น เราควรจะหยุดการเทรนที่จุดที่ Test error ต่ำที่สุด ไม่ใช่ Train error ต่ำที่สุด

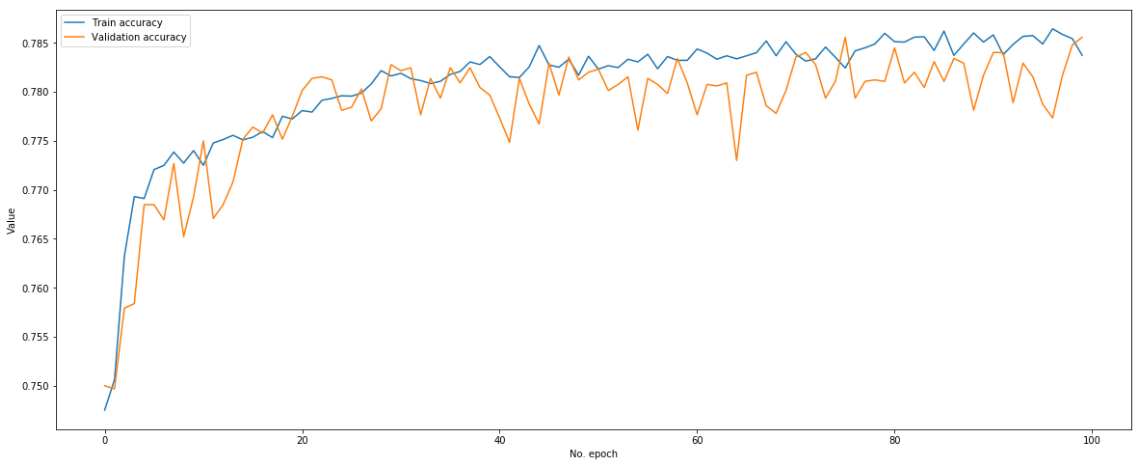
วิธีการในทางปฏิบัติ คือเมื่อเราเทรนโมเดลด้วยการเรียก Method .fit ให้เราแยกข้อมูลเทรนส่วนหนึ่งออกมาเป็น Validation set เพื่อให้ Optimiser ทดสอบความแม่นยำทุกๆ Epoch ดังนี้

Early Stopping เพื่อหยุดการเทรน ก่อนที่ Optimizer จะ Converge หา Loss ที่ต่ำที่สุด โดยมีสมมุติฐานว่ายิ่งเทรนไป โมเดลจะยิ่งซับซ้อนขึ้น

**ทำการ Early stopping**

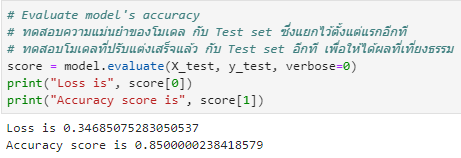


นำค่าความแม่นยำที่ได้มาพล็อตกราฟ



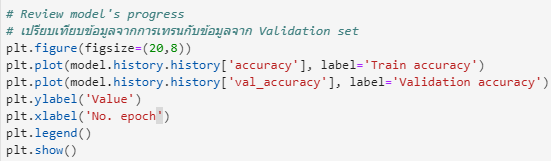
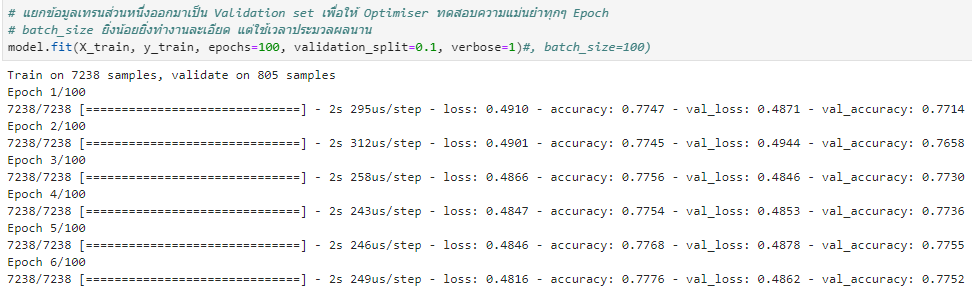
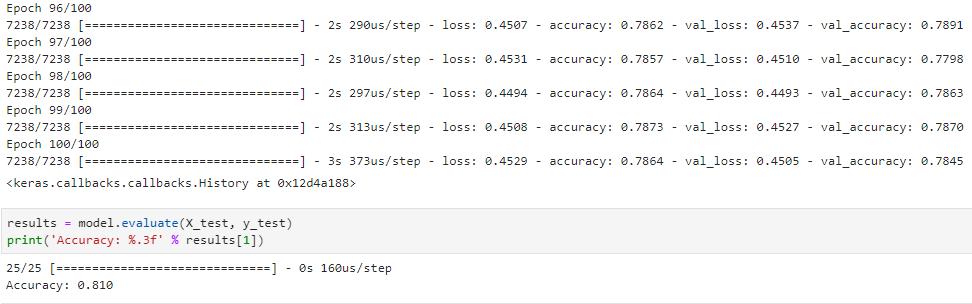
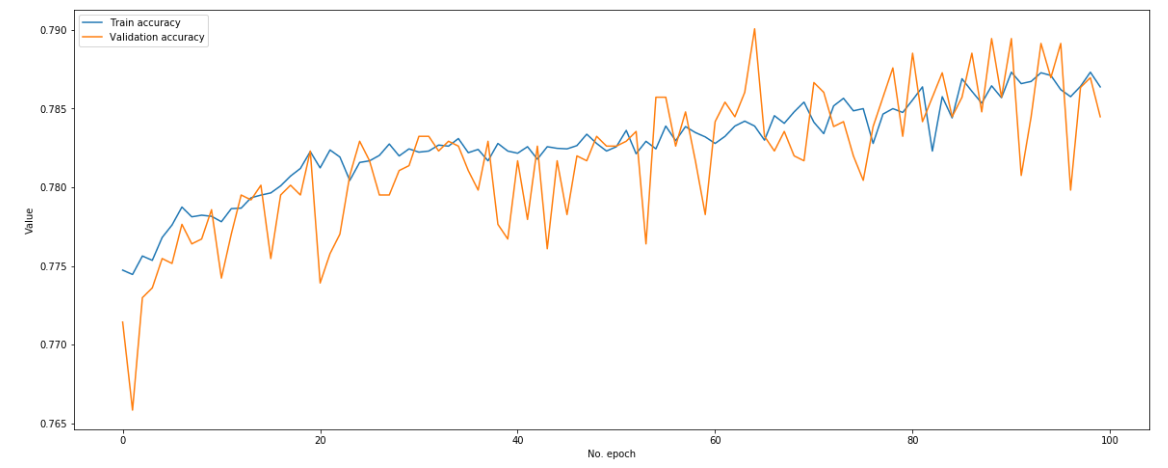
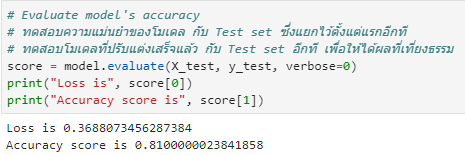
จะเห็นว่าโมเดลของเราสามารถเพิ่ม Train accuracy ได้อย่างต่อเนื่อง อย่างไรก็ตาม ดูเหมือน Validation accuracy จะไม่ยอมเพิ่มจากจุดสูงสุด คือประมาณ 0.93 ตั้งแต่ Epoch ที่ 8 จากข้อมูลนี้เราสามารถวิเคราะห์ได้ว่าโมเดลมีปัญหา Variance ซึ่งก็คือการที่ไม่สามารถ Generalise ให้พยากรณ์ข้อมูลที่ไม่เคยเห็นได้ดีเท่าที่ควร ดังนั้นข้อมูลนี้จึงบอกเราว่าเราควรลองมาตรการต่างๆ ที่จะช่วยลดปัญหา Variance เช่น การเพิ่มขนาด Validation set การเพิ่มหรือปรับเทคนิคการ Regularisation เป็นต้น

ทดสอบความแม่นยำของโมเดล กับ Test set ซึ่งแยกไว้ตั้งแต่แรกอีกที โดยการใช้ Method .evaluate



แหล่งข้อมูล : <https://guopai.github.io/ml-blog15.html>

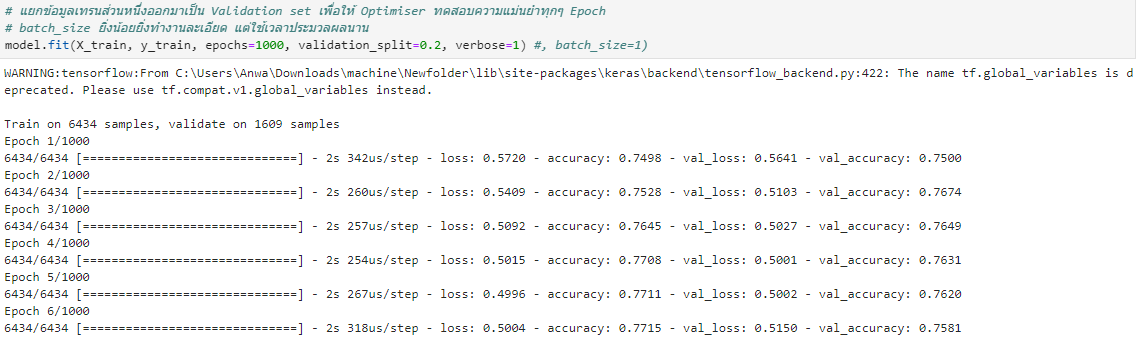
ต้องทดสอบกับ Test set อีก ทั้งๆ ที่ได้ทดสอบกับ Validation set แล้ว คำตอบคือ Validation set มีหน้าที่เป็นชุดทดสอบให้เราปรับแต่งค่าต่างๆ ของโมเดล ซึ่งทำให้เกิดความเสี่ยงที่เราจะปรับค่าเพื่อ "เอาใจ" Validation set ส่งผลให้เรา Overfit validation set ทำให้เราไม่รู้ว่าโมเดลของเราจะมีความสามารถในการพยากรณ์ข้อมูลจริงที่ไม่เคยมองเห็นเพียงใด ดังนั้นเราจึงต้องทดสอบโมเดลที่ปรับแต่งเสร็จแล้ว กับ Test set อีกที เพื่อให้ได้ผลที่เที่ยงธรรม ไม่หลอกตัวเอง



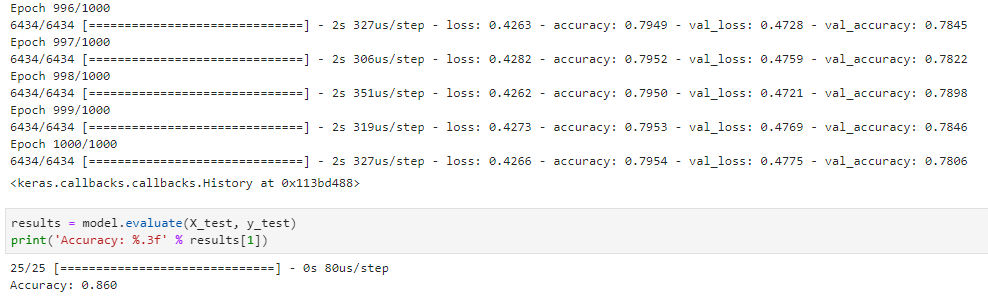
จำนวน epoch เป็นจำนวนรอบของการเทรน โดยทำให้แต่ละ Epoch จะทำให้ Loss ลดลง ในขณะที่ Accuracy เพิ่ม

**ที่ epochs = 100**

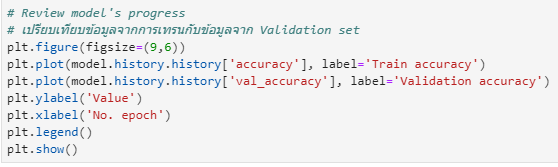
**ปรับเปลี่ยนจำนวน epoch**



เพิ่ม epochs จาก 100 เป็น 1000

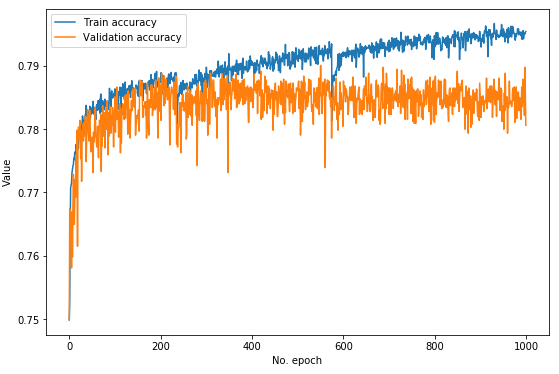


จะเห็นได้ว่าค่าความแม่นยำเพิ่มขึ้นหลังจากทำการเพิ่มจำนวน epochs



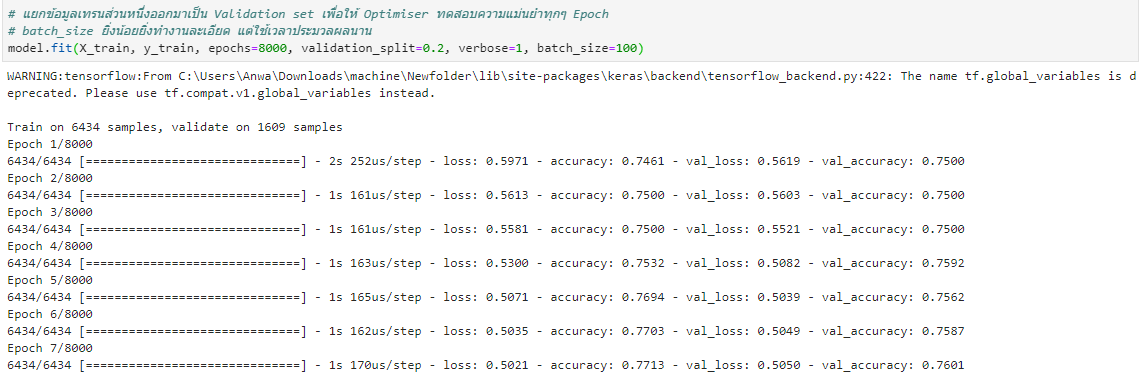
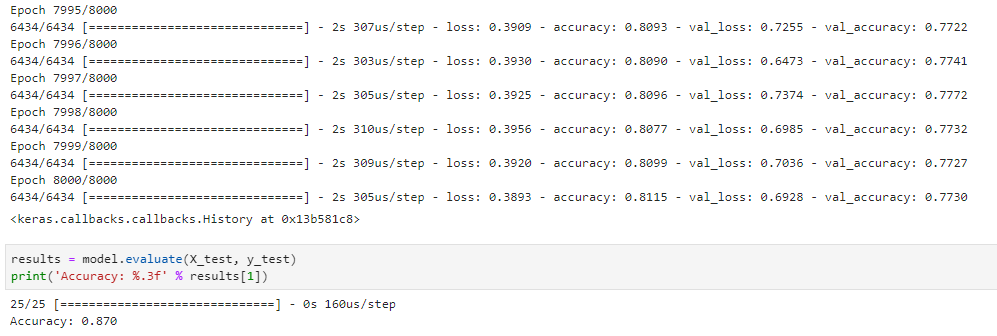
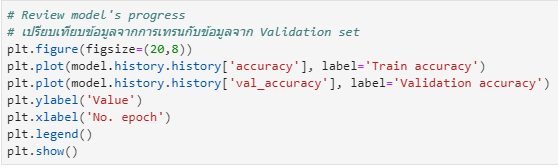
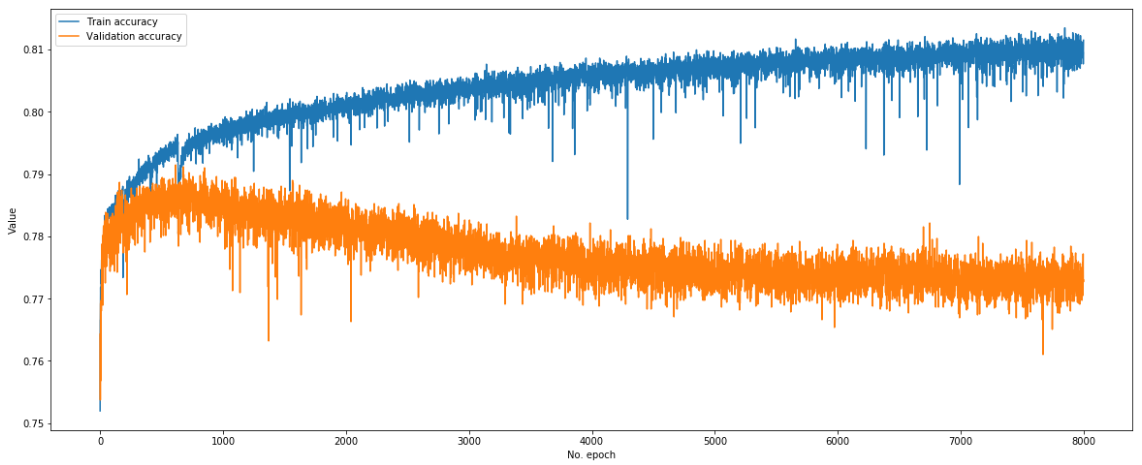
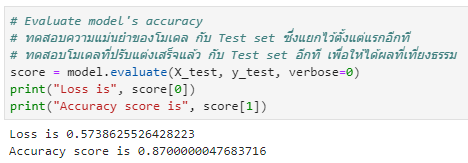
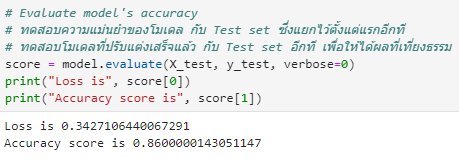
ทำการ plot กราฟระหว่างค่าความแม่นยำที่ได้จากการเทรน

(Train accuracy) กับ ค่าความแม่นยำที่ได้จากการแบ่งข้อมูล



จะเห็นได้ว่าเส้นกราฟความแม่นยำของข้อมูลที่ถูกเทรน

(Train accuracy) มีแนวโน้มที่เพิ่มขึ้นไปตามจำนวน epochs ที่เพิ่มขึ้น แต่ความแม่นยำที่ได้จะเห็นได้ว่ายังมีความไม่แน่นอน ระหว่าง epoch ความแม่นยำจะขึ้นๆลงๆ แต่เมื่อมองเป็นภาพรวมของepochs จะเห็นได้ว่าความแม่นยำมีแนวโน้มเพิ่มขึ้นเพิ่มขึ้น



จำนวน epochs สามารถปรับเพิ่มขึ้นได้ แต่ในที่นี้ขอทำไว้แค่ 8000 epochs เพราะประสิทธิภาพของเครื่องที่ทำการทดสอบนั้น มีประสิทธิภาพต่ำ ประมวลผลได้ช้า จากการลองเพิ่มจำนวน epochs พบว่า ยิ่งเพิ่มจำนวน epochs เครื่องที่ทำการทดสอบ จะต้องใช้เวลาประมวลผลมากขึ้นตามจำนวน epochs ที่เพิ่มขึ้น แต่จำนวน epoch มากขึ้นอาจจะให้ความแม่นยำ

เพิ่มขึ้น ถ้าแนวโน้มของกราฟยังคงไปในทิศทางเดิมเมื่อเพิ่ม epoch

แหล่งข้อมูล : <https://guopai.github.io/ml-blog15.html>

จะเห็นได้ว่าค่าความแม่นยำยังคงเพิ่มขึ้นหลังจากทำการเพิ่มจำนวน epochs จากเดิมที่เพิ่มไปแล้วก่อนหน้า

เพิ่ม epochs จาก 1000 เป็น 8000