

## Individual Study Report

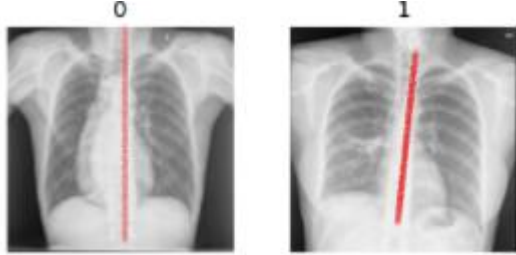
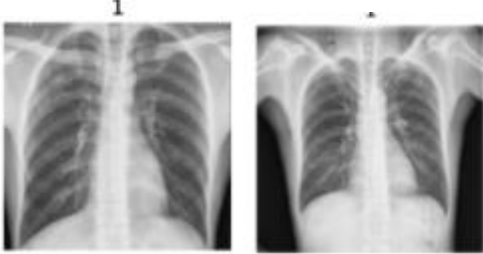
### ที่มา

เนื่องจากการทำ Model Image Classification ครั้งแรก อาจารย์เลยให้ไปเรียนคอร์ส Fast AI แล้วนำมาใช้กับ โจทย์ปัญหาการทำนายวัณโรคจากภาพ X-ray ปอด

### เตรียมข้อมูล

Dataset มีข้อมูลทั้งหมด 662 รูป ประกอบไปด้วยรูปที่เป็นปอดปกติ 326 รูป และปอดที่ไม่ปกติ 336 รูป โดยแบ่งเป็น Train-set 528 รูป, Valid-set 67 รูป และ Test-set 67 รูป ในสัดส่วน 80-10-10

หลังจากนั้นนำโมเดลมาทำ Image Augmentation โดยใช้วิธีดังนี้

| รายการ | การตั้งค่า                              | เหตุผล  |
|--------|---|---|
| Rotate | (max_deg=5, p=0.5)                      | เนื่องจากในโมเดลจะมีบางภาพที่ถ่ายไม่ตรงบ้าง หรือตัวคนนอนไม่ตรง เลยทำการหมุนรูปภาพเพียงเล็กน้อย เพื่อให้โมเดลไม่เรียนเฉพาะภาพตรงอย่างเดียว<br> |
| Zoom   | (draw=1.1, draw_x=0.5, draw_y=1, p=0.5) | บางภาพมีลักษณะการ zoom ภาพเข้าไป เลยทำให้ปอดใหญ่กว่าปกติ หรือปอดอยู่ตำแหน่งที่ไม่เหมือนกัน เลยทำการเพิ่มข้อมูลโดยการ zoom<br>                 |
| Flip   | (p=0.5)                                 | ตัวนี้มาจากสมมติฐานที่ว่า อาจจะมีเกิดที่ปอดข้างใดข้างหนึ่งเพียงข้างเดียว เลยทำให้การสลับข้างกัน อาจทำให้เรียนรู้ข้อมูลที่หลากหลายขึ้น   |

## โมเดล

เนื่องจากการทำนาย Image เลยเลือกใช้ pre-trained model แล้วนำมา fine-tune อีกที โดยตอนแรกได้ลอง Resnet 50, 101 แต่ด้วยความสงสัยเลยได้ทำการลองทุกโมเดล ที่ลองด้วยจำนวน epoch ที่ 10 epoch เพื่อดู แนวโน้มของ train-loss และ valid-loss โดยตารางที่นำมาให้ดูคือ loss ณ epoch ที่ 10

| Model         | Train_loss | Train_loss |
|---------------|------------|------------|
| Squeezenet1_0 | 0.216178   | 0.557256   |
| squeezenet1_1 | 0.221542   | 0.636527   |
| densenet121   | 0.133444   | 0.604451   |
| densenet201   | 0.103078   | 0.509935   |
| Resenet18     | 0.14931    | 0.68364    |
| Resnet34      | 0.12266    | 0.54941    |
| Resnet50      | 0.12989    | 0.76821    |
| Resnet101     | 0.13693    | 0.65718    |
| Alexnet       | 0.26517    | 0.48681    |
| vgg11_bn      | 0.15768    | 0.53938    |
| vgg13_bn      | 0.12463    | 0.57773    |
| vgg16_bn      | 0.11085    | 0.48646    |
| vgg19_bn      | 0.1121     | 0.67435    |

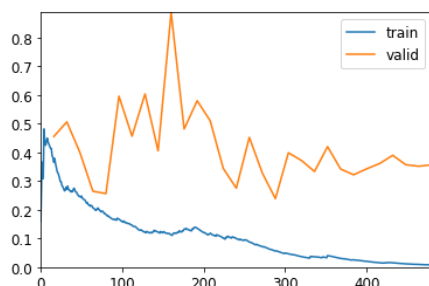
## เทรน

เกณฑ์ที่ใช้ในการวัดผลคือ Accuracy ได้มาจากการนับจำนวนผลลัพธ์ที่ทำนายว่าตรงกับค่าจริง เป็นร้อยละเท่าไร

ในการเทรนจะเทรนเป็นจำนวน 30 epoch แล้วดูลักษณะของ loss ว่าเป็นอย่างไร ถ้าไม่ overfit ก็จะเทรนต่อไป โดยได้ลองแต่ละโมเดล ดังนี้

### Densenet201

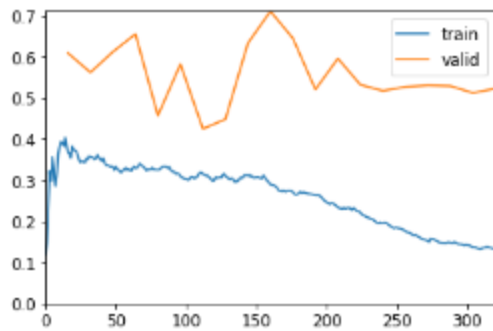
จากการที่ลองเทรนจะเห็นตัว valid-loss ที่ลงไปต่ำ และ accuracy ของ valid-set สูงถึง 92% เลยเลือกมาเทรน



จากกราฟจะเห็นว่า valid loss เติบโตมาแต่ก็พยายามลงเข้าหา train loss ตัว model ที่ดีที่สุดได้ Valid\_loss ที่ 0.275327 และ Valid\_accuracy ที่ 0.924

## Squeezenet1\_0

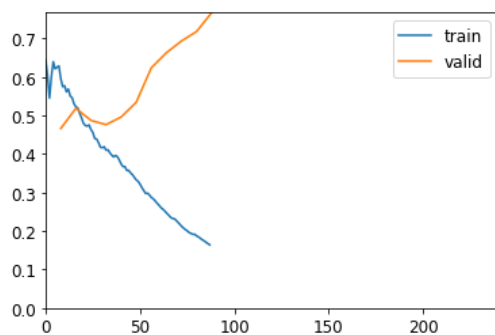
ตัวนี้ลองเพราะว่า เห็นว่าตัว train\_loss และ valid\_loss ไม่ต่างกันมาก และไม่ห่างกันเรื่อย เลยคิดว่าถ้าเทรนจำนวน epoch ที่เยอะขึ้น มันจะยังคงลงอย่างช้าๆ เรื่อยๆ ไม่ overfit



ผล loss ของทั้ง train และ valid ไม่ต่างกันมาก และเริ่มคงที่ไม่ลดลง ตัว model ที่ดีที่สุดได้ Valid\_loss ที่ 0.044645 และ Valid\_accuracy ที่ 0.8060

## Vgg16\_bn

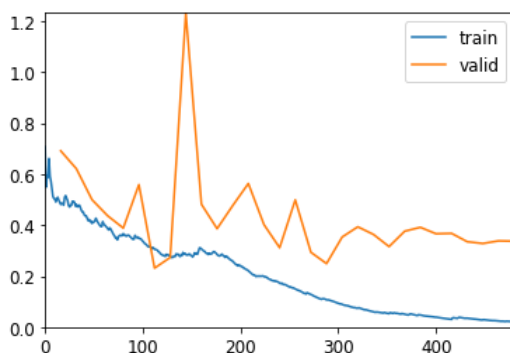
ค่า valid loss และ accuracy ตอนเทสเคยได้ถึง 0.39456 และ 0.92424 ตามลำดับ เลยทำให้สนใจลองเทรน



เมื่อลองเทรนไประดับหนึ่ง ตัว valid loss และ train loss กลับห่างกันมากๆ เลยทำให้หยุดเทรน

## Resnet34 ,50 , 101

ทั้ง 3 ตัวนี้ได้ลองแล้วพบว่าไม่ต่างกันมาก แต่ลักษณะของกราฟ Resnet 101 จะลงสวยกว่า คือ train-loss และ valid-loss ลงด้วยกันไปเรื่อยๆ



ผล โมเดลที่ดีที่สุด ตัว valid loss จะอยู่ที่ 0.231 และ valid accuracy อยู่ที่ 0.939

นำโมเดลที่ให้ valid loss และ valid accuracy ที่น่าสนใจมาทดสอบกับ Test set

| Model       | Valid loss | Valid accuracy | Test accuracy |
|-------------|------------|----------------|---------------|
| Densenet201 | 0.2781     | 0.9242         | 0.8507        |
| Resnet101   | 0.231      | 0.939          | 0.8507        |

### สิ่งที่ได้เรียนรู้

- ได้มีการลองใช้ fatsia library ครั้งแรก และทำหลายๆโมเดล (resnet34, resnet50, resnet101, vgg, densnet etc.)
- การลองผิดลองถูกในการทำ Image Augmentation