

## Report: neural networks ทั้ง 3 แบบ

ตัวแปรต้น คือข้อมูลที่ใช้ในการทำนาย ในกรณีนี้คือรูปภาพแมลง

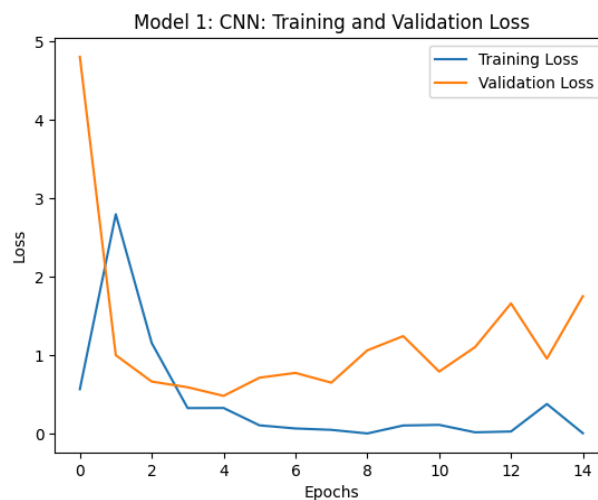
ตัวแปรตาม คือ ประเภทยของแมลง ซึ่งในที่นี้มี 3 ประเภท ได้แก่ หอยทาก ไล่ด้วง ผีเสื้อ

มีโมเดลที่ใช้ทั้งหมด 3 โมเดลดังนี้

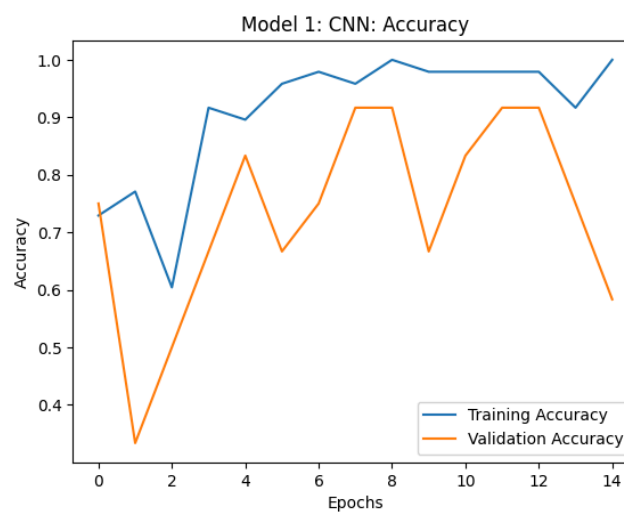
Model 1: CNN

```
model1 = tf.keras.models.Sequential([
    tf.keras.layers.Conv2D(16,(3,3),activation = 'relu',input_shape=(200,200,3)),
    tf.keras.layers.MaxPool2D(2,2),
    tf.keras.layers.Conv2D(32,(3,3),activation = 'relu'),
    tf.keras.layers.MaxPool2D(2,2),
    tf.keras.layers.Conv2D(64,(3,3),activation = 'relu'),
    tf.keras.layers.MaxPool2D(2,2),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(512,activation = 'relu'),
    tf.keras.layers.Dense(64,activation = 'relu'),
    tf.keras.layers.Dropout(0.5),
    tf.keras.layers.Dense(3,activation = 'softmax')
])
```

training loss



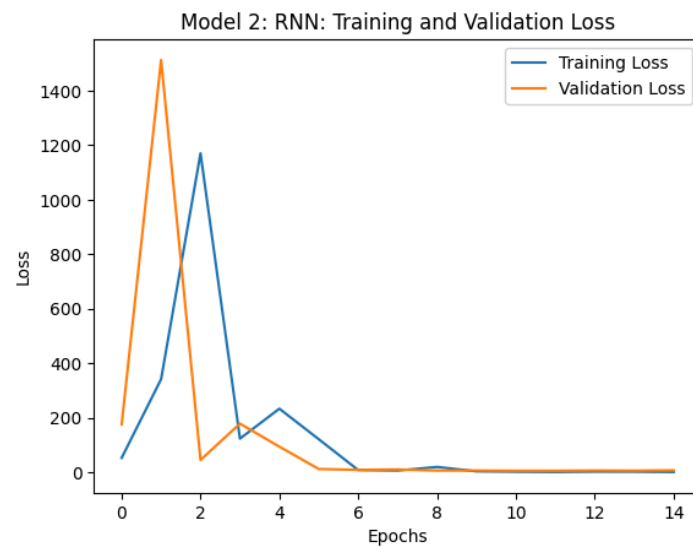
validation loss



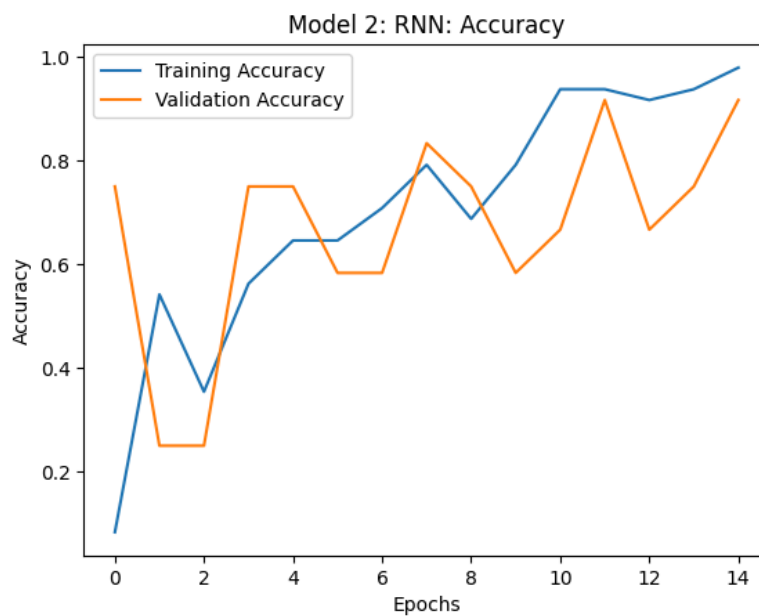
## Model 2: RNN

```
model2 = models.Sequential([
    layers.Conv2D(32, (3, 3), activation='relu', input_shape=(200, 200, 3)),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, (3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(128, (3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Flatten(),
    layers.Dense(512, activation='relu'),
    layers.Dropout(0.5),
    layers.Dense(3, activation='softmax')
])
```

training loss



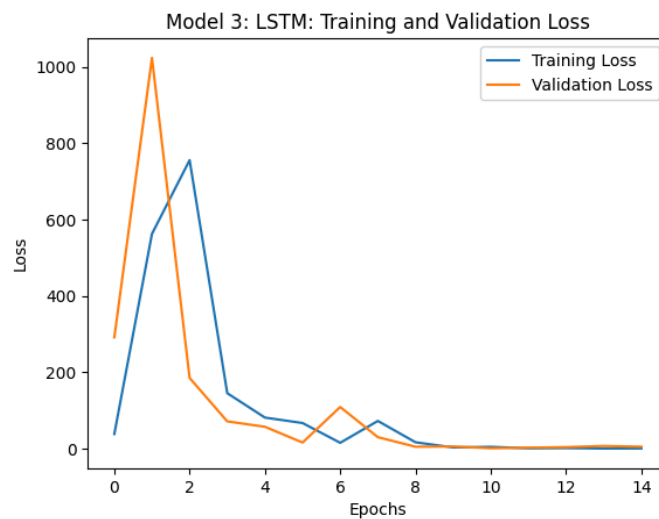
validation loss



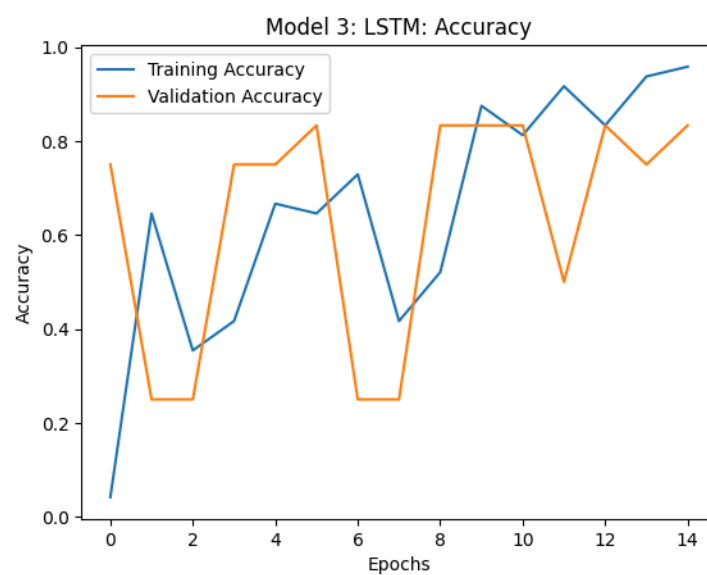
### Model 3: Deeper CNN

```
model3 = tf.keras.models.Sequential([
    tf.keras.layers.Conv2D(32,(3,3),activation = 'relu',input_shape=(200,200,3)),
    tf.keras.layers.MaxPool2D(2,2),
    tf.keras.layers.Conv2D(64,(3,3),activation = 'relu'),
    tf.keras.layers.MaxPool2D(2,2),
    tf.keras.layers.Conv2D(128,(3,3),activation = 'relu'),
    tf.keras.layers.MaxPool2D(2,2),
    tf.keras.layers.Conv2D(256,(3,3),activation = 'relu'),
    tf.keras.layers.MaxPool2D(2,2),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(512,activation = 'relu'),
    tf.keras.layers.Dense(64,activation = 'relu'),
    tf.keras.layers.Dropout(0.5),
    tf.keras.layers.Dense(3,activation = 'softmax')
])
```

training loss



validation loss



## สรุป

### CNN

#### 1. ความสัมพันธ์ระหว่าง (training loss, validation loss) vs epochs

- Training loss: ลดลงอย่างต่อเนื่องเมื่อ epochs เพิ่มขึ้น ซึ่งเป็นสัญญาณว่าโมเดลเรียนรู้จากข้อมูลฝึกได้ดี
- Validation loss: ในช่วงแรกลดลง แต่เริ่มมีความผันผวนและเพิ่มขึ้นเล็กน้อยหลังจาก epoch ที่ 20 บ่งบอกถึงการเกิด overfitting ซึ่งหมายความว่าโมเดลเริ่มจดจำข้อมูลฝึกมากเกินไปจนไม่สามารถ generalize กับข้อมูลใหม่ได้ดี

#### 2. ความสัมพันธ์ระหว่าง (training loss, validation loss) vs จำนวนตัวอย่างของ training set

- Training loss และ Validation loss ลดลงเมื่อจำนวนตัวอย่างของ training set เพิ่มขึ้น ซึ่งแสดงให้เห็นว่าเมื่อมีข้อมูลฝึกมากขึ้น โมเดลสามารถเรียนรู้ได้ดีขึ้นและลดการเกิด overfitting
- ช่องว่างระหว่าง training loss และ validation loss แคบลงเมื่อตัวอย่างเพิ่มขึ้น แสดงถึงการ generalize ที่ดีขึ้นของโมเดลต่อข้อมูลที่ไม่เคยเห็นมาก่อน

### การแก้ไข Overfitting

- การใช้เทคนิค early stopping สามารถช่วยป้องกันการ overfit ได้ โดยหยุดการเทรนเมื่อค่า validation loss เริ่มเพิ่มขึ้น (ประมาณ epoch ที่ 20-25) เพื่อให้โมเดลมีประสิทธิภาพมากขึ้นและไม่จดจำข้อมูลฝึกมากเกินไป

### RNN

#### 1. ความสัมพันธ์ระหว่าง (training loss, validation loss) vs epochs

- Training loss: ลดลงอย่างรวดเร็วในช่วงแรกและค่อยๆ ลดลงเมื่อเข้าสู่ epochs หลังๆ แสดงถึงการเรียนรู้ที่ดีในช่วงแรก
- Validation loss: ลดลงในช่วงแรก แต่เริ่มมีความผันผวนและมีแนวโน้มเพิ่มขึ้นหลังจาก epoch ที่ 10 นี่เป็นสัญญาณของการเริ่ม overfit โมเดลอาจเริ่มจดจำข้อมูลฝึกมากเกินไป แทนที่จะ generalize ไปยังข้อมูลใหม่

#### 2. ความสัมพันธ์ระหว่าง (training loss, validation loss) vs จำนวนตัวอย่างของ training set

- Training loss และ Validation loss: ลดลงเมื่อเพิ่มจำนวนตัวอย่าง แต่อาจสังเกตได้ว่า ช่องว่างระหว่าง training loss และ validation loss ยังคงกว้างอยู่ ซึ่งบ่งชี้ว่าโมเดลยังไม่สามารถ generalize ได้ดี
- Validation accuracy: มีความผันผวนและไม่ได้เพิ่มขึ้นอย่างชัดเจนหลังจาก epoch ที่ 10 บ่งบอกถึงการ overfit อีกครั้ง

### แนวทางการปรับปรุง

- Early stopping: เป็นเทคนิคที่แนะนำให้ใช้ในการหยุดการฝึกโมเดลเมื่อเริ่มเห็นการ overfitting ซึ่งควรหยุดประมาณ epoch ที่ 10-15

- Regularization techniques: ควรพิจารณาใช้การปรับปรุงโมเดลเพิ่มเติม เช่นการเพิ่ม L2 regularization, Dropout, หรือ Data augmentation เพื่อช่วยป้องกันการ overfit และเพิ่มความสามารถในการ generalize ของโมเดล

## LSTM

### 1. ความสัมพันธ์ระหว่าง (training loss, validation loss) vs epochs

- การที่ training loss ลดลงอย่างต่อเนื่องแสดงว่าโมเดลกำลังเรียนรู้จากข้อมูลการฝึกฝน
- Validation loss ที่ลดลงในช่วงแรกและคงที่หลังจากนั้นอาจบ่งชี้ว่าโมเดลเริ่มเรียนรู้ได้ดีขึ้นแต่มีการ overfitting อาจเริ่มต้นในบางช่วง

### 2. ความสัมพันธ์ระหว่าง (training loss, validation loss) vs จำนวนตัวอย่างของ training set

- การที่ทั้ง training loss และ validation loss ลดลงเมื่อมีจำนวนตัวอย่างเพิ่มขึ้น เป็นสัญญาณที่ดีว่าโมเดลมีความสามารถในการ generalize ที่ดีขึ้น
- ช่องว่างที่แคบลงระหว่าง training loss และ validation loss เป็นตัวบ่งชี้ว่าการเพิ่มข้อมูลช่วยให้โมเดลลดปัญหา overfitting

การที่ CNNs และ LSTMs มีประสิทธิภาพดีกว่า RNNs ในการ generalize และการที่ LSTM มีความสมดุลที่ดีที่สุดระหว่างการเรียนรู้และการ generalize เป็นข้อมูลที่สำคัญในการเลือกใช้โมเดลในโปรเจกต์ของคุณ

การใช้เทคนิค early stopping เป็นวิธีที่ดีเพื่อป้องกันการ overfitting โดยเฉพาะสำหรับ RNNs ที่แสดงสัญญาณของ overfitting ชัดเจน การเพิ่มข้อมูลในการฝึกก็มีผลดีต่อการปรับปรุงประสิทธิภาพของโมเดลด้วย