**เอกสารอธิบายการทดลองที่ 3 ตอนที่ 2**

**การทดลองสร้าง สอน และ ทดสอบโมเดล เพื่อดูค่าความแม่นยำเบื้องต้นของแต่ละโมเดล**

**(CNN, LSTM)**

**ตอนที่ 2 :** การทดลองสร้าง สอน และ ทดสอบโมเดล เพื่อดูค่าความแม่นยำเบื้องต้นของแต่ละโมเดล

(CNN, LSTM)

* 1. กำหนดโครงสร้างโมเดล (model architecture) ที่ต้องการนำมาทดสอบ

- กำหนดโครงสร้างโมเดล CNN โดยกำหนดโครงสร้างเป็น conv2d(32), MaxPooling2D(2, 2), conv2d(64), conv2d(128) , Flatten, Dense(128) และ Dense(5) ซึ่ง activation function เป็น ReLU ยกเว้น output layer เป็น sigmoid

model = Sequential()

model.add(Conv2D(32, kernel\_size=(3, 3), activation='relu', input\_shape=(X\_train.shape[1:]), padding='same'))

model.add(MaxPooling2D(pool\_size=(2, 2)))

model.add(BatchNormalization())

model.add(Dropout(0.25))

model.add(Conv2D(64, kernel\_size=(3, 3), activation='relu', padding='same'))

model.add(BatchNormalization())

model.add(Conv2D(128, kernel\_size=(3, 3), activation='relu', padding='same'))

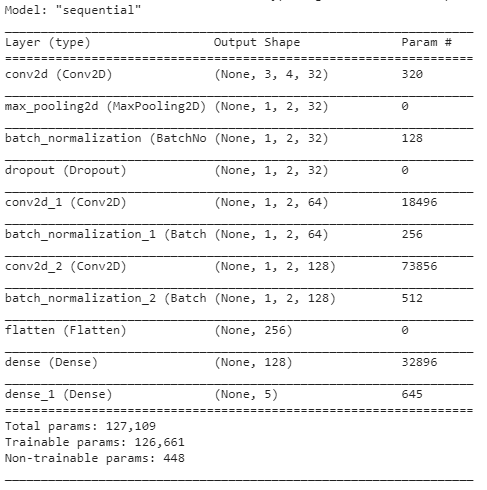
model.add(BatchNormalization())

model.add(Flatten())

model.add(Dense(128 , activation='relu' ))

model.add(Dense(y\_train.shape[1], activation='sigmoid'))

model.summary()



รูปโครงสร้างโมเดล CNN

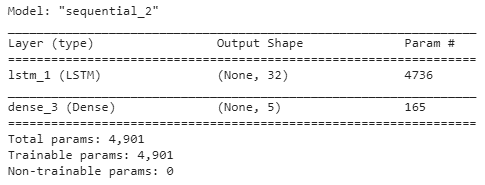
- กำหนดโครงสร้างโมเดล LSTM โดยกำหนดโครงสร้างเป็น LSTM(32) และ Dense(5) โดย output layer ใช้ activation function เป็น sigmoid

model = Sequential()

model.add(LSTM(32, input\_shape=(X\_2d\_train.shape[1:]) ))

model.add(Dense(y\_train.shape[1], activation='sigmoid'))

model.summary()



รูปโครงสร้างโมเดล LSTM

* 1. กำหนด Optimizer Parameters

- โมเดล CNN ใช้ optimizer เป็น Adam มี learning rate เท่ากับ 0.001 loss เป็น categorical\_crossentropy

optimizer = Adam(learning\_rate=0.001, beta\_1=0.9, beta\_2=0.999, amsgrad=False)

model.compile( loss='categorical\_crossentropy', optimizer=optimizer, metrics=['accuracy'])

- โมเดล LSTM ใช้ optimizer เป็น Adam มี learning rate เท่ากับ 0.003 loss เป็น categorical\_crossentropy

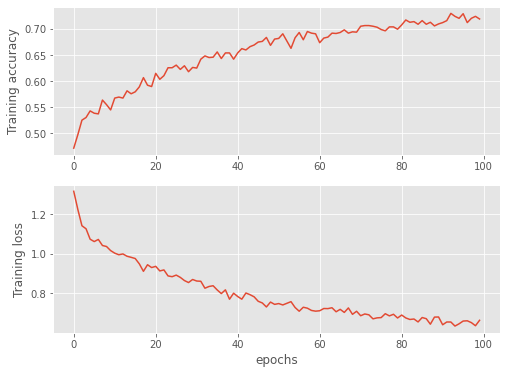
optimizer = Adam(learning\_rate=0.003, beta\_1=0.9, beta\_2=0.999, amsgrad=False)

model.compile( loss='categorical\_crossentropy', optimizer=optimizer, metrics=['accuracy'])

* 1. ทำการสอนโมเดล (Training model)

- สอนโมเดล CNN ด้วยข้อมูล X\_train และ y\_train ใช้ batch size เท่ากับ 16 epochs เท่ากับ 100 โดยมีชุดข้อมูล X\_valid และ y\_valid ในการตรวจสอบ

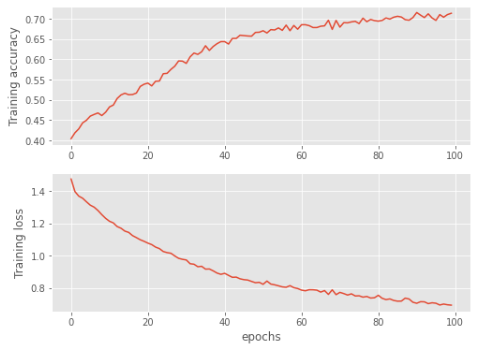
history = model.fit(X\_train, y\_train, batch\_size=16, validation\_data=(X\_valid, y\_valid), epochs=100, verbose=1)



รูปการณ์สอนโมเดล CNN

- สอนโมเดล LSTM ด้วยข้อมูล X\_2d\_train และ y\_2d\_train ใช้ batch size เท่ากับ 16 epochs เท่ากับ 100 โดยมีชุดข้อมูล X\_2d\_valid และ y\_2d\_valid ในการตรวจสอบ

history = model.fit( X\_2d\_train, y\_train, batch\_size=16, validation\_data=(X\_2d\_valid, y\_valid), epochs=100)



รูปการณ์สอนโมเดล LSTM

* 1. ทำการ predict ค่าจากโมเดลทั้ง 2 แบบ (CNN, LSTM) ด้วยข้อมูล X\_test, y\_test ที่แบ่งไว้

- ทำนายค่าจากโมเดล CNN ด้วยข้อมูล X\_test, y\_test

y\_prediction = model.predict(X\_test)

y\_pred\_single = [np.argmax(p) for p in y\_prediction]

y\_test\_single=[np.argmax(p) for p in y\_test]

- ทำนายค่าจากโมเดล LSTM ด้วยข้อมูล X\_2d\_test, y\_2d\_test

y\_prediction = model.predict(X\_2d\_test)

y\_pred\_single = [np.argmax(p) for p in y\_prediction]

y\_test\_single = [np.argmax(p) for p in y\_test]

* 1. และ 2.6 คำนวณค่าตัววัดประสิทธิภาพของการทำนายจากโมเดล CNN และ LSTM โดยวัดค่า

classification report/ confusion matrix และแสดงรูปภาพของ classification\_report และ confusion\_matrix

- คำนวณค่าตัววัดประสิทธิภาพของการทำนายจากโมเดล CNN ด้วยฟังชัน classification\_report จาก sklearn เพื่อดู precision, recall, f1-score, accuracy เป็นต้น และแสดง confusion matrix โดยพบว่าโมเดลมีความแม่นยำร้อยละ 70 จากการทำนายได้ดังนี้

print(classification\_report(y\_test\_single, y\_pred\_single))

conf\_mat = confusion\_matrix(y\_test\_single, y\_pred\_single)

plt.figure(figsize = (10, 7))

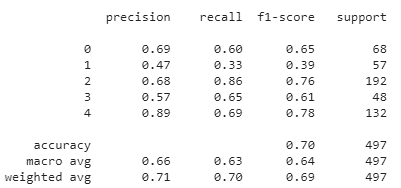
ax = sns.heatmap(conf\_mat, annot=True, fmt="d", xticklabels='0 1 2 3 4'.split(), yticklabels='0 1 2 3 4'.split(), cmap="Blues")

bottom, top = ax.get\_ylim()

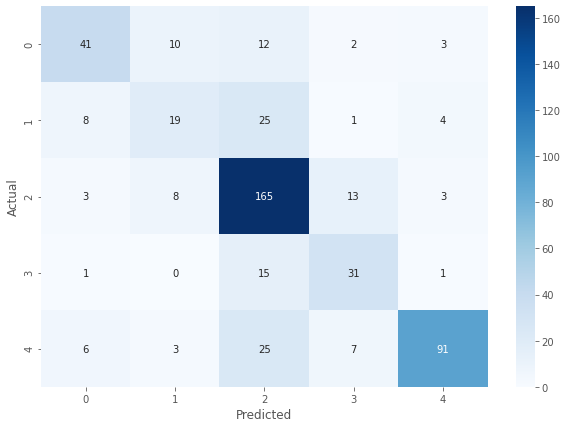
plt.ylabel("Actual")

plt.xlabel("Predicted")

plt.show()



รูปการวัดประสิทธิโมเดล CNN ด้วย classification\_report function



รูปการวัดประสิทธิโมเดล CNN ด้วย confusion matrix

- คำนวณค่าตัววัดประสิทธิภาพของการทำนายจากโมเดล LSTM ด้วยฟังชัน classification\_report จาก sklearn เพื่อดู precision, recall, f1-score, accuracy เป็นต้น และแสดง confusion matrix โดยพบว่าโมเดลมีความแม่นยำร้อยละ 74 จากการทำนายได้ดังนี้

print(classification\_report(y\_test\_single, y\_pred\_single))

conf\_mat = confusion\_matrix(y\_test\_single, y\_pred\_single)

plt.figure(figsize = (10, 7))

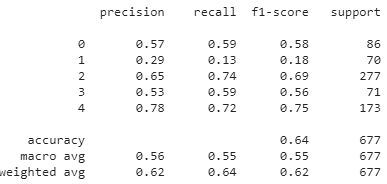
ax = sns.heatmap(conf\_mat, annot=True, fmt="d", xticklabels='0 1 2 3 4'.split(), yticklabels='0 1 2 3 4'.split(), cmap="Blues")

bottom, top = ax.get\_ylim()

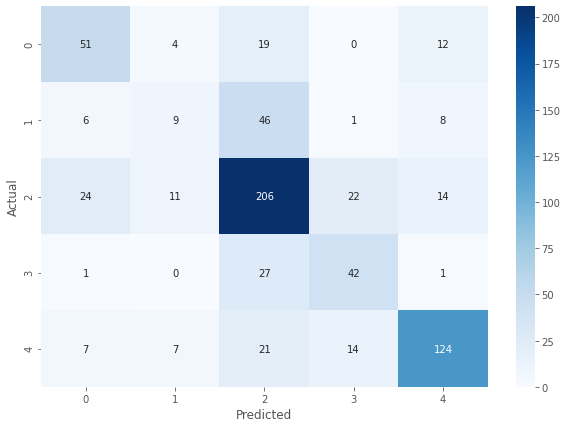
plt.ylabel("Actual")

plt.xlabel("Predicted")

plt.show()



รูปการวัดประสิทธิโมเดล LSTM ด้วย classification\_report function



รูปการวัดประสิทธิโมเดล LSTM ด้วย confusion matrix