# การบ้านการเขียนโปรแกรม 2:
# Logistic Regression

Thanks Andrew Ng for this beautiful programming exercise

ในการบ้านนี้เราจะทดลองเขียนโปรแกรมเพื่อทำตามขั้นตอนของ logistic regression โดยใช้ matrix operation ด้วย Octave/Matlab

ในpackage ประกอบด้วยไฟล์

- ex2.m - สคริปต์เพื่อรันโปรแกรมส่วนแรก
- ex2_reg.m - สคริปต์เพื่อรันโปรแกรมส่วนหลัง
- ex2data1.txt - ชุดข้อมูลสำหรับสอนระบบส่วนแรก
- ex2data2.txt - ชุดข้อมูลสำหรับสอนระบบส่วนหลัง
- mapFeature.m - ฟังก์ชันเพื่อสร้าง polynomial feature
- plotDecisionBoundary.m - ฟังก์ชันเพื่อสร้างกราฟแสดง decision boundary
- plotData.m - ฟังก์ชันเพื่อสร้าง 2D
- sigmoid.m* - sigmoid function
- costFunction.m* - Logistic Regression Cost Function
- predict.m* - Logistic Regression Prediction Function
- costFunctionReg.m* - Regularized Logistic Regression Cost

* คือ ไฟล์ที่ต้องแก้ไขและส่ง

# 1. Logistic Regression

ในส่วนนี้ เราจะต้องสร้างแบบจำลองด้วย logistic regression เพื่อทำนายว่านักเรียนจะได้รับเข้าเรียนในมหาวิทยาลัยหรือไม่

สมมติว่าคุณคือแอดมินที่ทำหน้าที่พิจารณารับเด็กเข้ามหาวิทยาลัย และมีข้อมูลเก่าในการรับนักเรียนประกอบด้วย คะแนนของนักเรียน 2 ครั้ง และผลการตัดสินรับเข้า คุณต้องสร้างระบบเพื่อทำนายความน่าจะเป็นที่นักเรียนได้รับการตอบรับเข้าเรียน

## 1.1 Sigmoid function

เพื่อสร้าง logistic regression model ผลลัพธ์ที่ได้ต้องมีค่า 0-1 ดังนั้นจึงต้องใช้ sigmoid function เพื่อแปลงค่า $h_\theta(x)$

ให้ใช้ไฟล์ sigmoid.m เพื่อเขียน sigmoid function โดย hypotheis function มีหน้าตาคือ

$$h_\theta(x) = g(\theta^T x),$$

โดยที่ function g คือ sigmoid function หาได้จาก $$g(z) = \frac{1}{1+e^{-z}}.$$

คุณควรเขียนโปรแกรมเป็น matrix operation ไม่ใช่วนลูปทำทีละค่า การทดสอบโปรแกรมว่าเขียนได้ถูกต้องหรือไม่ทำด้วยการเรียก sigmoid(x) ถ้าเรียก sigmoid(0) ควรได้เท่ากับ 0.5

## 1.2 Cost function และ gradient

ในส่วนนี้ เราต้องสร้าง cost function และ gradient สำหรับ logistic regression

### 1.2.1 Cost function

ให้แก้ไข costFunction.m เพื่อสร้าง cost function และ gradient ในไฟล์นี้
ค่า cost ให้คำนวณโดยใช้ฟังก์ชัน J( $\theta$ ) ดังสมการ

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\left[-y^{(i)}\log(h_\theta(x^{(i)})) - (1-y^{(i)})\log(1-h_\theta(x^{(i)}))\right],$$

ส่วน gradient หรืออนุพันธ์ของค่า cost เทียบกับ $\theta$ ดังนั้น gradient จึงเป็นเวกเตอร์ที่มีความยาวเท่ากับ $\theta$ โดย gradient ตัวที่ j หาได้จากสมการ

$$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}$$

ถึงแม้ว่าสมการ gradient จะมีหน้าตาเหมือน linear regression แต่จริงแล้วสำหรับ logistic regression มี $h_\theta$ (x) แตกต่างกัน

ในการเขียนโค้ดลงใน costFunction.m สามารถตรวจสอบว่าเขียนโปรแกรมได้ถูกต้องด้วยการรัน ex2.m ให้สังเกตว่าค่าที่คำนวณได้จากโปรแกรมที่คุณเขียนต้องเท่ากับ 0.693

### 1.2.2 Learning parameters ด้วย fminunc

fminunc() เป็นฟังก์ชันที่มีมาให้กับ Octave/Matlab สามารถใช้เพื่อหาค่าที่ดีที่สุด (optimization solver) ในที่นี้คือการหาค่าต่ำสุดของฟังก์ชันที่กำหนดให้

สำหรับ logistic regression เราต้อง optimized ค่า cost การเรียกใช้ fminunc() ได้เขียนโค้ดมาให้แล้วใน ex2.m ดังภาพด้านล่าง

```
%  Set options for fminunc
options = optimset('GradObj', 'on', 'MaxIter', 400);

%  Run fminunc to obtain the optimal theta
%  This function will return theta and the cost
[theta, cost] = ...
    fminunc(@(t)(costFunction(t, X, y)), initial_theta, options);
```

จะเห็นว่า fminunc() เรียกใช้ costFunction() ดังนั้นถ้าคุณเขียน costFunction() ได้ถูกต้องจะพบว่าค่าที่ได้คือ 0.203

### 1.2.3 การประเมินความถูกต้อง

เมื่อได้ค่า $h_\theta$(x) ที่มีค่าระหว่าง 0-1 แล้ว เราต้องสรุปเอาท์พุตสำหรับแต่ละอินพุต ในส่วนนี้ให้แก้ไขไฟล์ predict.m เพื่อทำนายค่าผลลัพธ์จาก $h_\theta$(x) ที่คำนวณได้ โดยกำหนดให้ $h_\theta$(x) >=0.5 ให้ผลลัพธ์เป็น 1 ถ้าไม่ใช่ให้ทำนายเป็น 0 เมื่อเขียนโปรแกรมส่วนนี้เสร็จ เมื่อรันสคริปต์ ex2.m จะพบค่าที่ได้จากการรัน คือ 89.00

# 2. Regularized Logistic Regression

ในส่วนนี้ คุณจะได้ฝึกเขียนโปรแกรม regularization โดยใช้ตัวอย่างข้อมูลเรื่องการทดสอบไมโครชิพ

การทดสอบประกอบด้วย 2 ขั้นตอน เมื่อพิจารณา 2 ขั้นตอนจะได้ผลว่าชิพผ่านการทดสอบหรือไม่ เรากำลังจะสร้าง logistic regression model เพื่อทำนายผล ระบบจะถูกสอนด้วยข้อมูลการทดสอบเก่าๆ

ในข้อ 2 นี้ให้สั่งรันด้วยสคริปต์ ex2_reg.m

## 2.1 Feature mapping

ในส่วนนี้ ใช้สำหรับขยาย feature ที่มีออกไปเป็นหลายๆ feature โดยทำ polynomial ของแต่ละ feature ในฟังก์ชันนี้ได้ขยายออกไปถึง 28 features คุณสามารถดูตัวอย่างการเขียน feature mapping ได้ในไฟล์ mapFeature.m คุณไม่ต้องแก้โค้ดในข้อนี้แต่อย่างใด

การที่เราขยาย feature มีข้อดีคือได้คุณลักษณะสำคัญเพิ่มขึ้น แต่ก็เกิดผลเสียคือ overfitting ด้วยเช่นกัน ดังนั้นเรา

ต้องทำ regularization เพื่อแก้ปัญหา overfitting ที่อาจเกิดขึ้นได้

## 2.2 Cost function และ gradient

ในส่วนนี้ คุณต้องแก้ไขไฟล์ costFunctionReg.m เพื่อทำให้เกิดการคำนวณแบบ regularized logistic regression ไฟล์นี้ทำหน้าที่คล้ายๆกับ costFunction.m ในข้อ 1.2 แต่เราต้องเพิ่มส่วน regularization ได้ตามสมการ

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\left[-y^{(i)}\log(h_\theta(x^{(i)})) - (1-y^{(i)})\log(1-h_\theta(x^{(i)}))\right] + \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2.$$

อย่าลืมว่า เราไม่ทำ regularization กับ $\theta_0$ ใน Octave/Matlab ใช้การอ้างอิง index เริ่มที่ 1 ดังนั้น เราจะต้องไม่ทำ regularization กับ theta(1)

ส่วน gradient ก็คล้ายกับ 1.2 เช่นกัน เราจะใช้ fminunc() ในการหาค่าที่ดีที่สุด แต่เราต้องทำหน้าที่เขียนการคำนวณ gradient ให้กับ fminunc() ให้ถูกต้อง โดยสมการ gradient เป็นดังนี้

$$\frac{\partial J(\theta)}{\partial \theta_0} = \frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)} \qquad \text{for } j = 0$$

$$\frac{\partial J(\theta)}{\partial \theta_j} = \left(\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}\right) + \frac{\lambda}{m}\theta_j \quad \text{for } j \geq 1$$

เมื่อเราสั่งรัน ex2_reg.m มันจะไปเรียก costFunctionReg.m เพื่อคำนวณ cost และ gradient โค้ดที่คุณเขียนควรให้ผลลัพธ์ค่า cost เท่ากับ 0.693

นอกจากนี้ตลอดการรัน สคริปต์ทั้งสอง คุณจะพบกับกราฟที่ถูกสร้างขึ้นเพื่อแสดงตัวอย่างข้อมูล นอกจากนี้ยังมีส่วนอื่นๆ ประกอบไปใน zip ไฟล์ชุดนี้ด้วย คุณสามารถศึกษาเพิ่มเติมได้

# ตารางคะแนน

| ที่ | งานที่ต้องทำ | ไฟล์ที่ต้องแก้ไขและส่ง | คะแนน |
|---|---|---|---|
| 1 | sigmoid function | sigmoid.m | 5 |
| 2 | คำนวณ cost สำหรับ logistic regression | costFunction.m | 30 |
| 3 | คำนวณ gradient สำหรับ logistic regression | costFunction.m | 30 |
| 4 | predict function | predict.m | 5 |
| 5 | คำนวณ cost สำหรับ regularized LR | costFunctionReg.m | 15 |
| 6 | คำนวณ gradient สำหรับ regularized LR | costFunctionReg.m | 15 |
| | รวม | | 100 |