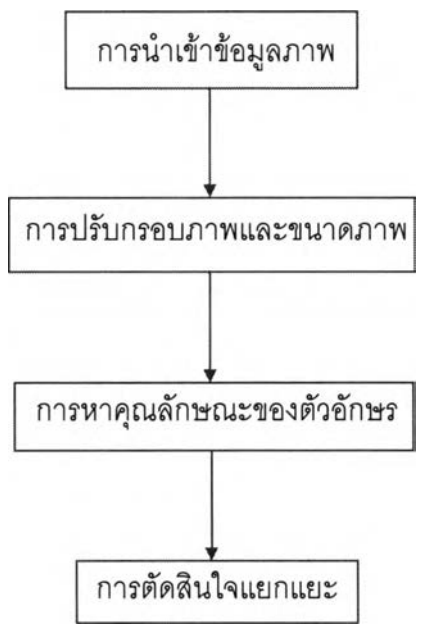
บทที่ 3
การออกแบบและพัฒนา

โปรแกรมสำหรับรู้จำตัวอักษรภาษาไทยในงานวิจัยนี้จะทำการรู้จำตัวอักษรภาษาไทยจากข้อมูลภาพที่ได้จากเครื่องสแกนเนอร์แล้วทำการรู้จำและแสดงผลลัพธ์ ในงานวิจัยนี้จะสามารถแสดงผังการทำงานได้ดังรูปที่ 3.1

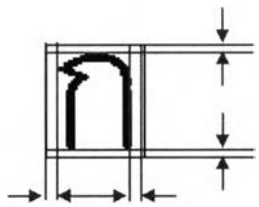


รูปที่ 3.1 ผังการทำงานของการรู้จำตัวอักษร

ซึ่งสามารถอธิบายแต่ละส่วนสำคัญได้ดังนี้

การนำเข้าข้อมูลภาพ

ข้อมูลภาพจะได้จากเครื่องสแกนเนอร์ชนิดที่รับข้อมูลเข้าทีละหน้ากระดาษซึ่งสามารถใช้ความละเอียดได้ตั้งแต่ 200 จุดต่อนิ้ว (DPI) ขึ้นไป โดยจะอ่านภาพในลักษณะขาว-ดำ (bi level) และจัดเก็บภาพของตัวอักษรหนึ่งตัวต่อหนึ่งแฟ้มข้อมูล โดยข้อมูลภาพภายในแฟ้มข้อมูลจัดเก็บในรูปแบบบิตแมพไฟล์ของไมโครซอฟต์วินโดวส์ (Microsoft Windows BMP) โดยจุดภาพจะต้องมีจุดขาวล้อมรอบกรอบของรูปอย่างน้อย 1 จุดดังรูปที่ 3.2



รูปที่ 3.2 กรอบจุดภาพของตัวอักษร

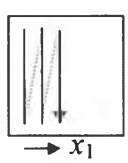
การปรับกรอบภาพและการปรับขนาดภาพ

ก่อนทำการแปลงแบบเค-แอล (K-L Transform) ข้อมูลภาพจะต้องถูกปรับกรอบ (กรณีที่มีจุดขาวล้อมรอบกรอบมากกว่า 1 จุด) และทำการปรับขนาดให้มีขนาด 32x32

การปรับกรอบภาพ

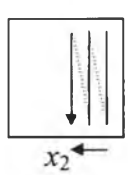
การปรับกรอบภาพทำได้โดย

1. หาจุดดำจุดแรกด้านซ้ายโดยการเริ่มหาจากจุดบนสุดทางซ้ายสุดไปยังจุดล่างสุดทางซ้ายสุด เมื่อพบจุดแรกให้บันทึกจำนวนสดมภ์ที่ทำการหา x_1 ดังรูปที่ 3.3



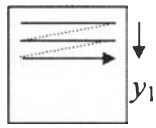
รูปที่ 3.3 การหาจุดดำจุดแรกด้านซ้าย

2. หาจุดดำจุดแรกด้านขวาโดยการเริ่มหาเริ่มหาจากจุดบนสุดทางขวาสุดไปยังจุดล่างสุดทางขวาสุด เมื่อพบจุดแรกให้บันทึกจำนวนสดมภ์ที่ทำการหา x_2 ดังรูปที่ 3.4



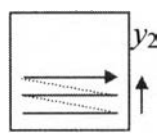
รูปที่ 3.4 การหาจุดดำจุดแรกด้านขวา

3. หาจุดดำจุดแรกด้านบน โดยการเริ่มหาจากจุดบนด้านซ้ายสุดไปยังจุดบนด้านขวาสุด เมื่อพบจุดแรกให้บันทึกจำนวนแถวที่ทำการหา y_1 ดังรูปที่ 3.5



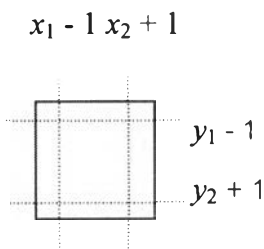
รูปที่ 3.5 การหาจุดดำจุดแรกด้านบน

4. หาจุดดำจุดแรกด้านล่าง โดยการเริ่มหาจากจุดล่างด้านซ้ายสุดไปยังจุดล่างด้านขวาสุด เมื่อพบจุดแรกให้บันทึกจำนวนแถวที่ทำการหา y_2 ดังรูปที่ 3.6



รูปที่ 3.6 การหาจุดดำจุดแรกด้านล่าง

5. กรอบภาพใหม่คือกรอบภาพจากระยะ x_1 - 1 ถึง x_2 + 1 ตามแนวตั้งและระยะ y_1 - 1 ถึง y_2 + 1 ตามแนวนอนดังรูปที่ 3.7



รูปที่ 3.7 กรอบภาพใหม่ของตัวอักษรซึ่งมีจุดขาวล้อม 1 จุด

การเปลี่ยนขนาดภาพ

ภาพที่ถูกปรับกรอบแล้วจะถูกคำนวนค่าอัตราการย่อขยาย (zoom factor) เพื่อทำการเปลี่ยนขนาดภาพ อัตราการย่อขยาย (zoom factor) สามารถหาได้จาก

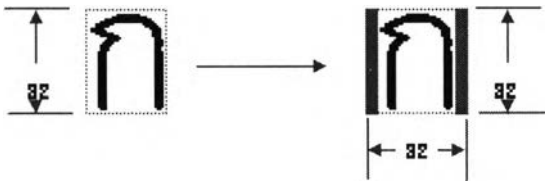
$$zoom = \frac{32}{x_2 - x_1 + 2} \quad \text{เมื่อ } x_2 - x_1 > y_2 - y_1 \tag{15}$$

$$zoom = \frac{32}{y_2 - y_1 + 2} \quad \text{เมื่อ } y_2 - y_1 > x_2 - x_1 \tag{16}$$

เมื่อได้ค่าอัตราการย่อขยาย (zoom factor) แล้ว ก็นำข้อมูลจุดภาพที่ทำการปรับกรอบแล้ว ทำการเปลี่ยนขนาดตามวิธีการ 2D-DDA (Digital Differential Analyzer) [14] ซึ่งมีขั้นตอนการทำงานดังนี้

1. คำนวนค่า izoom จาก izoom = (int) (1/zoom * 1000)
2. ตั้งค่า y = 0, dday = 0
3. for แต่ละแถว i ในภาพดั้งเดิม ทำข้อ 4 ถึงข้อ 16
    4. อ่านข้อมูลภาพดั้งเดิมในแถว i
    5. dday = dday - 1000, x = 0, ddax = 0
    6. for แต่ละจุด j ของข้อมูลภาพดั้งเดิมในแถว i ทำข้อ 7 ถึงข้อ 11
        7. ddax = ddax - 1000
        8. while ddax < 0 ทำข้อ 9 ถึงข้อ 11
            9. ข้อมูลภาพที่ย่อหรือขยายแถวที่ y สดมภ์ที่ x = ข้อมูลภาพดั้งเดิมแถวที่ i สดมภ์ที่ j
            10. x = x + 1
            11. ddax = ddax + izoom
    12. while dday < 0 ทำข้อ 13 ถึงข้อ 16
        13. dday = dday + izoom
        14. for แต่ละ k ที่น้อยกว่า x ทำข้อ 15
            15. ข้อมูลภาพที่ย่อหรือขยายแถวที่ y สดมภ์ที่ k = ข้อมูลภาพที่ย่อหรือขยายแถวที่ y - 1 สดมภ์ที่ k
        16. y = y + 1

หลังจากทำการย่อขยายแล้วภาพที่ได้จะยังไม่ได้ขนาด 32x32 เพียงแค่เป็นภาพที่มีขนาดในแนวตั้งหรือขนาดในแนวนอนเป็น 32 จุดเท่านั้น จึงต้องผ่านขั้นตอนการเพิ่มจุดขาวในด้านซ้ายและด้านขวาของภาพจำนวนเท่า ๆ กัน (หรือต่างกันไม่เกิน 1 สดมภ์) สำหรับภาพที่มีขนาดในแนวตั้งเป็น 32 จุด (เนื่องจากภาพก่อนทำการย่อขยายมีขนาด $y_2 - y_1 > x_2 - x_1$) ดังรูปที่ 3.8 หรือผ่านขั้นตอนการเพิ่มจุดขาวในด้านบนและด้านล่างของภาพจำนวนเท่า ๆ กัน (หรือต่างกันไม่เกิน 1 แถว) สำหรับภาพที่มีขนาดในแนวนอนเป็น 32 จุด (เนื่องจากภาพก่อนทำการย่อขยายมีขนาด $x_2 - x_1 > y_2 - y_1$) ดังรูปที่ 3.9



รูปที่ 3.8 การเพิ่มจุดขาวในด้านซ้ายและด้านขวาเพื่อให้ภาพมีขนาด 32x32

รูปที่ 3.9 การเพิ่มจุดขาวในด้านบนและด้านล่างเพื่อให้ภาพมีขนาด 32x32

การหาคุณลักษณะของภาพของตัวอักษร (Feature Extraction)

การหาคุณลักษณะของภาพในงานวิจัยนี้จะใช้หลักการของการวิเคราะห์ตัวประกอบสำคัญ (Principal Component Analysis) หรือการแปลงแบบเค-แอล (K-L Transform) โดยป้อนค่าข้อมูลภาพที่มีขนาด 32x32 ผ่านการแปลงแบบเค-แอล (โดยอาศัยเมตริกซ์ของการแปลง $\mathbf{A}_{128}$ หรือใช้ไอเกนเวกเตอร์จากค่าของไอเกนจำนวน 128 ตัว และ เมตริกซ์ของค่าเฉลี่ย - $\mathbf{m}_x$) จะได้ค่าเวกเตอร์รูปแบบเป็นคุณลักษณะ (Feature) ของตัวอักษรในทางสถิติ ซึ่งเวกเตอร์รูปแบบนี้มีจำนวนจริงเป็นสมาชิกและมีจำนวนสมาชิก 128 ตัว ซึ่งการใช้ค่าจำนวนสมาชิกของเมตริกซ์ของการแปลงจำนวน 128 ตัวนั้นได้จากการพิจารณาค่าของไอเกน ซึ่งถ้าเลือกใช้จำนวนสมาชิก 128 ตัวนั้น ค่าผิดพลาดกำลังสองเฉลี่ยจะมีค่าเท่ากับผลรวมของค่าของไอเกนตัวที่ 129 ถึงค่าของไอเกนตัวที่ 1024 ซึ่งมีค่า 0.029645 หรือค่าผิดพลาดกำลังสองเฉลี่ยของแต่ละภาพเมื่อเทียบกับภาพตัวอย่างจะมีค่า 0.008 %

ค่าของไอเกน 128 ตัวแรกที่ได้จากการเรียนรู้ตัวอักษรต้นแบบเรียงลำดับจากซ้ายไปขวาแสดงดังรูปที่ 3.10

หอสมุดกลาง
จุฬาลงกรณ์มหาวิทยาลัย

-356.87097168 0.01735689 0.01688636 0.00903188 0.01045159 0.00685551
0.00601884 0.00469873 0.00491467 0.00440400 0.00402912 0.00337862
0.00335495 0.00319503 0.00288317 0.00285533 0.00264186 0.00252409
0.00211901 0.00204232 0.00187430 0.00172579 0.00171791 0.00161079
0.00147081 0.00139779 0.00140822 0.00132775 0.00136504 0.00124086
0.00124917 0.00117222 0.00111388 0.00114137 0.00107043 0.00103792
0.00100244 0.00099172 0.00088708 0.00091082 0.00085372 0.00082574
0.00079393 0.00077895 0.00074494 0.00075587 0.00071370 0.00073206
0.00069172 0.00068174 0.00065691 0.00064103 0.00060068 0.00061241
0.00058066 0.00057124 0.00055933 0.00056797 0.00053370 0.00054810
0.00051851 0.00051563 0.00049265 0.00048588 0.00046940 0.00046466
0.00045380 0.00044977 0.00044082 0.00043368 0.00043068 0.00041776
0.00040841 0.00041418 0.00039073 0.00039488 0.00038461 0.00037377
0.00036757 0.00036816 0.00035680 0.00035598 0.00034776 0.00034532
0.00033879 0.00032940 0.00033310 0.00032167 0.00032073 0.00031720
0.00031184 0.00030534 0.00029720 0.00030195 0.00028819 0.00029298
0.00028130 0.00028558 0.00027738 0.00027410 0.00026977 0.00026809
0.00025821 0.00025212 0.00025342 0.00024381 0.00024788 0.00023991
0.00024147 0.00023503 0.00023638 0.00022840 0.00022736 0.00022191
0.00021685 0.00021718 0.00021526 0.00021206 0.00020912 0.00020749
0.00020396 0.00020080 0.00019824 0.00019438 0.00019598 0.00019317
0.00019009 0.00018660 ...

รูปที่ 3.10 ค่าของไอเกน (eigenvalue) 128 ตัวแรกที่ได้จากการเรียนรู้ตัวอักษรต้นแบบ

ตัวอย่างเวกเตอร์รูปแบบที่ได้จากการแปลงแบบเค-แอลของภาพของตัวอักษร "อ" แบบ AngsanaUPC ขนาด 18 จุดแสดงดังรูปที่ 3.11

[	-3.699393	8.296499	-4.891393	-0.620662	1.524827	0.096978	0.017958	-1.791362
	0.382805	-0.960314	5.893456	2.468534	-1.531986	3.247048	1.119271	3.677793
	-0.209297	-1.905469	-0.855790	-0.428803	0.852810	1.317816	3.073161	-0.921573
	-1.451192	-1.206515	-2.781378	-2.565777	2.875234	0.591285	1.666931	1.357944
	0.874958	1.738651	-4.051826	-2.629217	0.669532	3.032220	-2.633668	1.392023
	1.270272	-1.633380	0.298109	-2.805218	0.881618	2.725453	-0.383987	1.485536
	-3.138905	-2.970975	-0.385779	2.049857	0.646807	0.795004	0.227419	-0.403857
	1.471575	0.256897	-1.116753	-0.923488	0.924196	-0.363500	2.918891	-3.111311
	1.062739	1.804268	0.164240	-1.844158	0.290373	-0.454820	0.725363	-1.191349
	-0.954824	1.893200	-1.205526	0.144799	0.508617	-0.495954	-1.009125	0.487022
	-1.606899	-0.347653	0.678400	-1.108924	-0.827494	2.181400	0.245098	-0.187766
	2.210433	-1.027383	0.565491	-1.152517	1.725313	0.186326	-1.380497	-2.039173
	-0.393573	-0.678224	1.156076	2.250832	1.006601	0.679295	0.977033	-0.672213
	-0.788908	-0.189242	1.549033	1.692105	-0.192571	0.562044	-1.088539	0.584427
	-0.482797	-1.014668	-0.166762	1.536467	1.190907	-0.043104	1.094190	-0.254196
	-0.330953	0.778149	-1.487227	-0.316855	0.160531	-0.748823	1.362948	-1.333852]

รูปที่ 3.11 ตัวอย่างเวกเตอร์รูปแบบที่ได้จากการแปลงเค-แอลของตัวอักษร "อ" แบบ AngsanaUPC ขนาด 18 จุด

ขณะทำการเรียนรู้ตัวอักษรต้นแบบ โปรแกรมจะทำการคำนวนค่าเมตริกซ์ของค่าเฉลี่ย ($\mathbf{m}_x$) และเมตริกซ์ของการแปลง ($\mathbf{A}_{128}$) เก็บลงในแฟ้มข้อมูลซึ่งจะถูกอ่านมาใช้ขณะทำการหาคุณลักษณะของภาพ (ทำการแปลงแบบเค-แอล)

การตัดสินใจแยกแยะ (Classification)

ข้อมูลเวกเตอร์รูปแบบจะถูกป้อนเข้าสู่นิวรอลเน็ตเวิร์กเพื่อตัดสินใจแยกแยะตัวอักษร โดยนิวรอลเน็ตเวิร์กจะมี

1. จำนวนบัพ (node) ในชั้นทางเข้าจำนวน 128 บัพ

เนื่องจากเวกเตอร์รูปแบบมีจำนวนสมาชิก 128 ตัว ดังนั้นจำนวนบัพในชั้นทางเข้าจึงมี 128 บัพ

2. จำนวนบัพในชั้นที่ถูกซ่อนจำนวน 128 บัพ

จำนวนบัพในชั้นที่ถูกซ่อนได้จากการทดลอง จำนวนบัพในชั้นที่ถูกซ่อนที่มากเกินไปโดยมีจำนวนตัวอย่างที่ใช้เรียนรู้ไม่เพียงพอจะทำให้เกิดความพอดีเกินไป (overfitting) ซึ่งทำให้นอกจากนิวรอลเน็ตเวิร์กเรียนรู้ตัวอย่างแล้ว นิวรอลเน็ตเวิร์กยังเรียนรู้สัญญาณรบกวนด้วย มีผลให้การประมาณค่าของข้อมูลเมื่อข้อมูลไม่เหมือนตัวอย่างให้ผลลัพธ์ที่มีค่าผิดพลาดไปมาก[15][16]

จากการทดลองจำนวนบัพในชั้นที่ถูกซ่อนที่จำนวน 73 บัพ 100 บัพ 128 บัพ และ 200 บัพ พบว่าจำนวนบัพในชั้นที่ถูกซ่อนจำนวน 128 บัพให้ผลความถูกต้องของการรู้จำมากกว่าจำนวนบัพในชั้นที่ถูกซ่อน 200 บัพ 100 และ 73 บัพ ดังแสดงในตารางที่ 3.1 อย่างไรก็ตามอัตรารู้จำที่ได้ไม่แตกต่างกันมากนัก จำนวนรอบ (Iteration) ที่ใช้ในการสอนนิวรอลเน็ตเวิร์กก็เป็นตัวแปรสำคัญตัวแปรหนึ่ง เนื่องจากการสอนนิวรอลเน็ตเวิร์กใช้เวลาค่อนข้างนาน

ตารางที่ 3.1 เปรียบเทียบผลของการรู้จำเมื่อนิวรอลเน็ตเวิร์กมีจำนวนบัพในชั้นที่ถูกซ่อน 73 บัพ 100 บัพ 128 บัพ และ 200 บัพ

	73 บัพ	100 บัพ	128 บัพ	200 บัพ
จำนวนรอบ (Iteration) ที่ใช้ในการสอนให้นิวรอลเน็ตเวิร์กคอนเวอร์เจนส์	32765	51778	42721	31925
อัตราการรู้จำเมื่อทดสอบกับข้อมูลชุดที่ 2	95.07 %	94.09 %	97.15 %	96.29 %
อัตราการรู้จำเมื่อทดสอบกับข้อมูลชุดที่ 3	95.37 %	95.22 %	96.54 %	96.23 %

3. จำนวนบัพ (node) ในชั้นทางออกจำนวน 68 บัพ

เนื่องจากตัวอักษรที่ทำการรู้จำประกอบด้วยพยัญชนะ 44 ตัว สระ วรรณยุกต์ 24 ตัว รวมเป็น 68 ตัวอักษร จำนวนบัพในชั้นทางออกจึงมี 68 บัพ

4. ค่าอัตราการเรียนรู้ 0.08

ค่าอัตราการเรียนรู้เลือกค่าใด ๆ ระหว่าง 0 - 1 ถ้าทำการสอนแล้วไม่สำเร็จคือ นิวรอลเน็ตเวิร์กไม่คอนเวอร์เจนส์ (convergence) ให้เลือกค่าใหม่โดยเลือกค่าที่ต่ำลง ถ้าค่าอัตราการเรียนรู้สูงเกินไปจะทำให้นิวรอลเน็ตเวิร์กไม่คอนเวอร์เจนส์เนื่องจากเกิดการออสซิลเลชัน สำหรับค่าอัตราการเรียนรู้ที่ต่ำเกินไปจะทำให้นิวรอลเน็ตเวิร์กใช้เวลานานมากก่อนที่จะคอนเวอร์เจนส์[17]

ในการวิจัยนี้ใช้ค่า 0.08 ซึ่งนิวรอลเน็ตเวิร์กคอนเวอร์เจนส์

5. ค่าอินเนอร์เชีย (Inertia) 0.95

ค่าอินเนอร์เชียเป็นค่าใด ๆ ระหว่า 0 - 1 ใช้สำหรับกำหนดสัดส่วนของค่าโมเมนตัมที่ใช้ในขณะทำการเรียนรู้ การลดค่าอินเนอร์เชียอาจจะทำให้นิวรอลเน็ตเวิร์กเกิดอาการออสซิลเลชัน ระหว่างการเรียนรู้ ทำให้นิวรอลเน็ตเวิร์กไม่คอนเวอร์เจนส์ การใช้ค่าอินเนอร์เชียทำให้สามารถใช้ค่าอัตราการเรียนรู้ได้สูงขึ้น

6. ฟังก์ชันกระตุ้นแบบซิกมอยด์ซึ่งให้ค่าของฟังก์ชันอยู่ในช่วง (0,1)

ฟังก์ชันกระตุ้นแบบซิกมอยด์นี้เพื่อความรวดเร็วจะใช้การคำนวนแบบค้นหาในตาราง 1024 ค่า และไม่มีการอินเตอร์โพเลชัน (Interpolation)

เนื่องจากขณะที่ค่าของฟังก์ชันกระตุ้นแบบซิกมอยด์มีค่าใกล้เคียง 0 หรือใกล้เคียง 1 ค่าของอนุพันธ์ของฟังก์ชันกระตุ้นแบบซิกมอยด์ $f'(x)$ ซึ่งมีค่าเท่ากับ $f(x)\ (1 - f(x))$ จะมีค่าเข้าสู่ 0 ซึ่งทำให้ไม่เกิดการเรียนรู้หรือมีการเรียนรู้น้อยมาก (ไม่เกิดการเปลี่ยนแปลงค่า

ของน้ำหนักและค่าไบแอสในแต่ละชั้น) ดังนั้นจะใช้การไบแอสค่าอนุพันธ์ของฟังก์ชันกระตุ้นแบบซิกมอยด์เพื่อให้มีการเรียนรู้ขณะที่ฟังก์ชันซิกมอยด์มีค่าใกล้เคียง 0 หรือใกล้เคียง 1 [18][19]ตามสมการ

$$f'(x) = f(x)(1 - f(x)) + 0.05 \tag{17}$$

7. เงื่อนไขของการคอนเวอร์เจนส์

ในการเรียนรู้ตัวอักษรต้นแบบในการวิจัยนี้ได้เลือกเงื่อนไขของการคอนเวอร์เจนส์คือ ต้องเรียนรู้ทุกตัวอย่าง (3264 ตัวอย่าง) โดยมีค่าผิดพลาดไม่เกิน 0.25

8. ค่าเริ่มต้นของเมตริกซ์ของค่าน้ำหนักและค่าไบแอส

ในการวิจัยนี้ค่าเริ่มต้นของเมตริกซ์ของค่าน้ำหนักและค่าไบแอสได้จากการสุ่มค่าระหว่าง [-0.1,0.1]

ในแต่ละเวกเตอร์รูปแบบของแต่ละตัวอักษรจะทำให้บัพในชั้นทางออกเพียงบัพใดบัพหนึ่งเท่านั้นที่มีค่าสูง (เป็น 1 หรือมากกว่า 0.5) และบัพที่เหลือในชั้นทางออกจะมีค่าต่ำ (เป็น 0 หรือต่ำกว่า 0.5) สำหรับบางตัวอักษร (ขณะทำการทดสอบการรู้จำ) อาจจะไม่มีบัพใดที่ชั้นทางออกมีค่ามากกว่า 0.5 หรือมีค่าเป็นหนึ่งเลยจะถือว่าบัพที่ให้ค่ามากที่สุดเป็นค่าสูง และบางตัวอักษร (ขณะทำการทดสอบการรู้จำ) อาจมีบัพมากกว่า 1 บัพที่มีค่าสูง (เป็น 1 หรือมากกว่า 0.5) ซึ่งจะใช้บัพที่มีค่ามากที่สุดเป็นค่าสูง ส่วนบัพที่เหลือถือว่าเป็นค่าต่ำ

ในการตัดสินใจแยกแยะ นิวรอลเน็ตเวิร์กจะอ่านค่าน้ำหนักที่ได้ทำการเรียนรู้ในขั้นการเรียนรู้ซึ่งเก็บไว้ในแฟ้มข้อมูลมาใช้ และผลลัพธ์จะแสดงค่าผลลัพธ์ของแต่ละบัพในชั้นทางออกพร้อมกับแยกแยะว่าเป็นตัวอักษรค่าแอสกีใด

รูปที่ 3.12 แสดงตัวอย่างผลลัพธ์ที่ได้จากนิวรอลเน็ตเวิร์กเมื่อให้ทำการรู้จำเวกเตอร์รูปแบบที่ได้จากการ แปลงเค-แอลของตัวอักษร "อ" แบบ AngsanaUPC ขนาด 18 จุด

Iteration: 1

black box recog:

Outputs: 0 000001 0.000001 0.000001 0.000001 0.000001 0.000001 0.000001 0.000001
0.000001 0.000001 0.000001 0.000001 0.000001 0.000001 0.000001 0.000001 0.000001 0.000001
0.000001 0.000001 0.000001 0.000001 0.000001 0.000001 0.000001 0.000001 0.000001 0.000001
0.000001 0.000001 0.000001 0.000001 0.000001 0.021125 0.000001 0.000001 0.000001 0.000001
0.000001 0.000001 0.000001 0.000001 0.000001 0.000001 0.999999 0.210766 0.000001 0.000001
0.000001 0.000001 0.000001 0.000001 0.000001 0.000001 0.000001 0.000001 0.000001 0.000001
0.000001 0.000001 0.000001 0.000001 0.000001 0.000001 0.000001 0.000001 0.000001 0.000001

Output is 205 (อ)

Targets: 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 1.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000

Target is 205 (อ)

Iteration: 1 - pass

รูปที่ 3.12 ผลลัพธ์ที่ได้จากนิวรอลเน็ตเวิร์กเมื่อให้ทำการรู้จำเวกเตอร์รูปแบบที่ได้จากการแปลงเค-แอลของตัวอักษร "อ" แบบ AngsanaUPC ขนาด 18 จุด