COMP9444 Project 1 Report

“Japanese Character Recognition”

Alex Lee (z5207126)

Part 1 – Q3.

Final Accuracy

Test set: Average loss: 0.2481, Accuracy: 9387/10000 (94%)

Confusion Matrix

			Predicted									
			o	**ki**	**su**	**tsu**	**na**	**ha**	**ma**	**ya**	**re**	**wo**
			0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Target	**o**	**0**	949	4	1	1	33	1	1	5	3	2
	ki	**1**	0	923	6	0	10	1	41	4	6	9
	su	**2**	13	9	889	27	7	6	26	6	7	10
	tsu	**3**	0	3	14	960	3	4	6	3	3	4
	na	**4**	19	12	3	4	931	6	12	5	3	5
	ha	**5**	4	10	43	4	2	903	24	2	4	4
	ma	**6**	4	6	11	1	6	3	964	2	0	3
	ya	**7**	4	4	3	0	5	0	7	958	5	14
	re	**8**	4	13	6	3	13	2	8	3	943	5
	wo	**9**	7	8	3	2	3	1	2	3	4	967

Part 1: Japanese Character Recognition

Part 1 – Q1.

Final Accuracy

Test set: Average loss: 1.0102, Accuracy: 6967/10000 (70%)

Confusion Matrix

			Predicted									
			o	**ki**	**su**	**tsu**	**na**	**ha**	**ma**	**ya**	**re**	**wo**
			0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Target	**o**	**0**	772	5	9	11	29	61	2	63	30	18
	ki	**1**	7	672	107	18	29	21	58	12	25	51
	su	**2**	7	60	694	27	26	20	45	35	47	39
	tsu	**3**	5	37	63	753	14	60	14	18	25	11
	na	**4**	58	55	77	22	626	18	33	35	21	55
	ha	**5**	8	27	124	17	19	723	29	8	34	11
	ma	**6**	4	26	145	10	24	24	723	21	10	13
	ya	**7**	16	31	28	13	83	14	53	624	89	49
	re	**8**	11	37	97	42	5	31	44	7	704	22
	wo	**9**	9	50	87	3	53	31	18	29	41	679

Part 1 – Q2.

Final Accuracy

Test set: Average loss: 0.4974, Accuracy: 8492/10000 (85%)

Confusion Matrix

			Predicted									
			o	**ki**	**su**	**tsu**	**na**	**ha**	**ma**	**ya**	**re**	**wo**
			0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Target	**o**	**0**	847	4	3	6	25	37	4	38	30	6
	ki	**1**	4	818	33	2	16	12	62	5	15	33
	su	**2**	8	12	832	49	8	19	23	16	17	16
	tsu	**3**	3	11	22	924	1	16	5	1	7	10
	na	**4**	37	24	21	5	822	8	27	19	21	16
	ha	**5**	8	13	83	10	12	834	19	2	14	5
	ma	**6**	3	13	44	10	15	6	893	8	1	7
	ya	**7**	19	11	17	3	21	9	32	839	22	27
	re	**8**	10	23	29	49	4	8	30	3	838	6
	wo	**9**	3	15	44	7	33	5	21	16	11	845

Part 1 – Q3.

<u>Final Accuracy</u>

Test set: Average loss: 0.2481, Accuracy: 9387/10000 (94%)

<u>Confusion Matrix</u>

			Predicted									
			o	**ki**	**su**	**tsu**	**na**	**ha**	**ma**	**ya**	**re**	**wo**
			0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Target	**o**	**0**	949	4	1	1	33	1	1	5	3	2
	ki	**1**	0	923	6	0	10	1	41	4	6	9
	su	**2**	13	9	889	27	7	6	26	6	7	10
	tsu	**3**	0	3	14	960	3	4	6	3	3	4
	na	**4**	19	12	3	4	931	6	12	5	3	5
	ha	**5**	4	10	43	4	2	903	24	2	4	4
	ma	**6**	4	6	11	1	6	3	964	2	0	3
	ya	**7**	4	4	3	0	5	0	7	958	5	14
	re	**8**	4	13	6	3	13	2	8	3	943	5
	wo	**9**	7	8	3	2	3	1	2	3	4	967

Part 1 – Q4.

It is clear from the results above that the accuracy improves as the complexity of the model increases. We can see that 'NetLin' which was the simplest of the three models had the lowest accuracy of around 70% whereas 'NetFull', which employs the use of a hidden layer with 'tanh' activation performed better with 85% accuracy. Furthermore, NetConv utilised convolutional layers as well as a fully connected layer with 'relu' activations which gave us an accuracy of 94% which was the highest out of the three models.

Fig 1.4 – Most frequent misclassifications (red = most frequent, orange = 2nd most frequent, yellow = 3rd most frequent)

Target	**Predicted**		
	NetLin	**NetFull**	**NetConv**
お (o)	や (ya), は (ha)	や (ya), は (ha)	な (na)
き (ki)	ま (ma)	ま (ma)	ま (ma)
す (su)	き (ki)	つ (tsu)	つ (tsu), ま (ma)
つ (tsu)	す (su)	す (su)	す (su)
な (na)	す (su)	お (o)	お (o)
は (ha)	す (su)	す (su)	す (su)
ま (ma)	す (su)	す (su)	す (su)
や (ya)	れ (re)	ま (ma)	を (wo)
れ (re)	す (su)	つ (tsu)	な (na), き (ki)
を (wo)	す (su)	す (su)	き (ki)

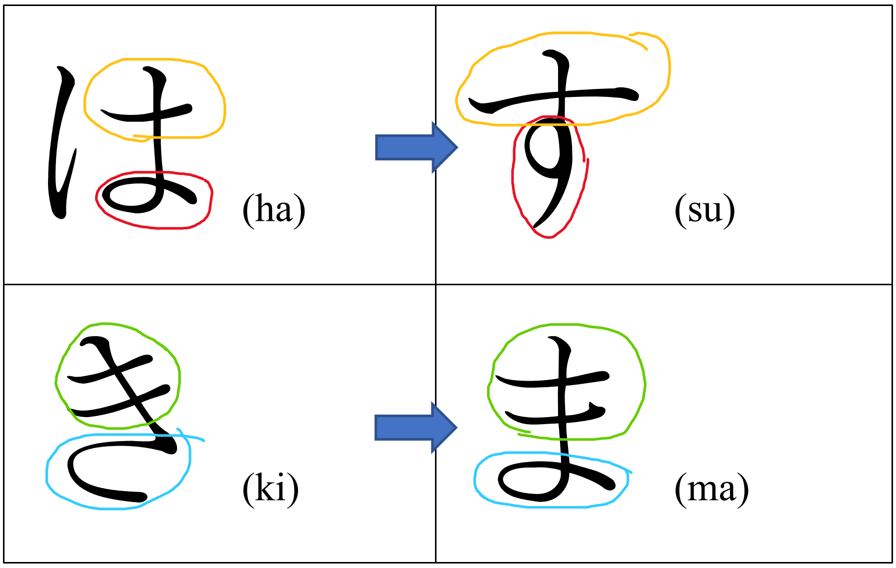
If we look at 'NetConv' and its three most frequent misclassifications in descending order from *fig1.4* above, we can see that this model is most likely to mistake:

1. は (ha) for す (su)
2. き (ki) for ま (ma)
3. お (o) for な (na)

All three models seem to misclassify 'は (ha) for す (su)' and 'き (ki) for ま (ma)'. However, 'NetLin' and 'NetFull' often mistakes 'お (o)' for 'や (ya) and は (ha)' rather than 'な (na)' which is how 'NetConv' behaves.

To understand why some characters may be mistaken for others, we look at three comparisons below.

We can see that the character ‘は (ha)’ is often mistaken for ‘す (su)’ and ‘き (ki)’ is mistaken for ‘ま (ma)’ in all three models. From looking at the comparisons below we can see two very similar features in both characters circled below for both pairs.



For the ‘NetConv’ model, ‘お (o)’ is often mistaken for ‘な (na)’, however the other two models ‘NetLin’ and ‘NetFull’ mistake ‘お (o)’ for either ‘や (ya)’ or ‘は (ha)’. By inspection we can see that the misclassification from ‘NetConv’ is more convincing compared to the misclassification from ‘NetFull’ and ‘NetFull’ for this example.