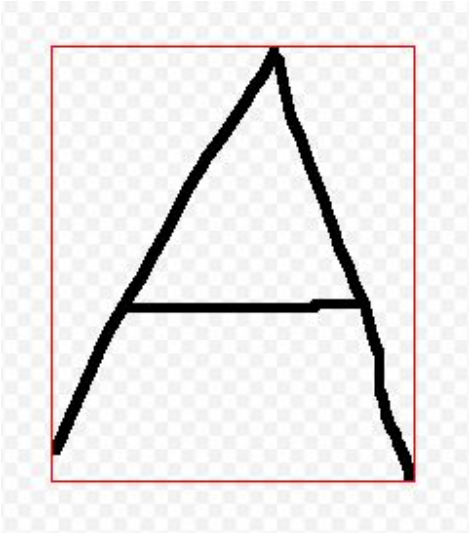
DrawBoard

Akshath Jain
Period 5/6

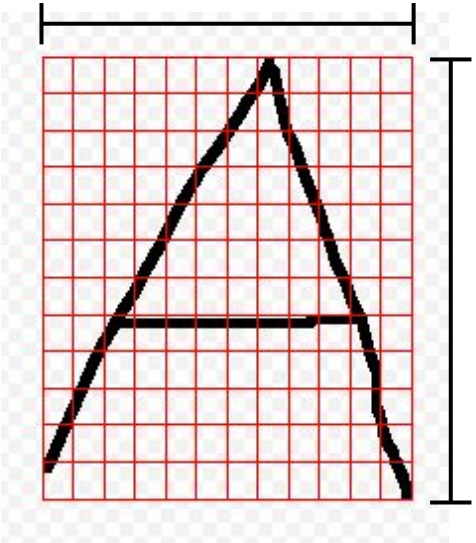
Handwriting Recognition

- Handwriting recognition (HWR) takes what someone writes and converts it into text on a computer
- HWR algorithms are very similar to Optical Character Recognition (OCR) algorithms in that both HWR and OCR use machine learning to correlate hundreds of features to a specific output
- HWR is used widely in commercial applications such as form recognition and USPS mail sorting

Feature Extraction Algorithm



1) Edges are identified

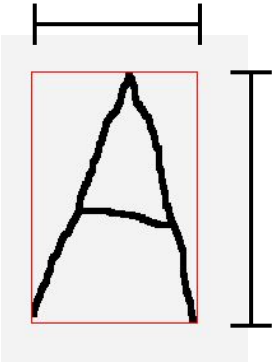


2) Aspect ratio is recorded and the image is broken into a 12 x 12 grid

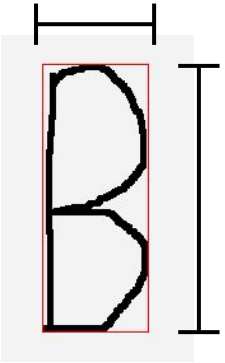
	A	B	C	D	E	F	G	H	I	J	K	L	M
1	0	0	0	0	0	0			0	0	0	0	
2	0	0	0	0	0	0.1			0.4	0	0	0	
3	0	0	0	0	0		0.5	0.5		0	0	0	
4	0	0	0	0			0	0		0.1	0	0	
5	0	0	0	0.9			0	0	0		0	0	
6	0	0	0		0.1	0	0	0	0		0	0	
7	0	0			0	0	0	0	0.2			0	
8	0											0	
9	0		0	0	0	0	0	0	0	0		0	
10			0	0	0	0	0	0	0	0		0.8	
11		0	0	0	0	0	0	0	0	0	0.6		
12	0.8	0	0	0	0	0	0	0	0	0	0		
13													

3) A 12 x 12 “rasterization” of the image is developed based on average pixel count

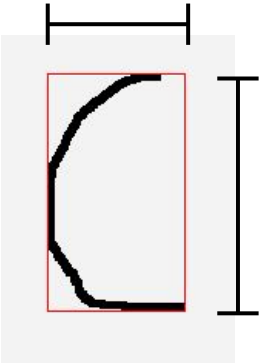
Examples



Aspect Ratio: 0.8



Aspect Ratio: 0.6



Aspect Ratio: 0.7

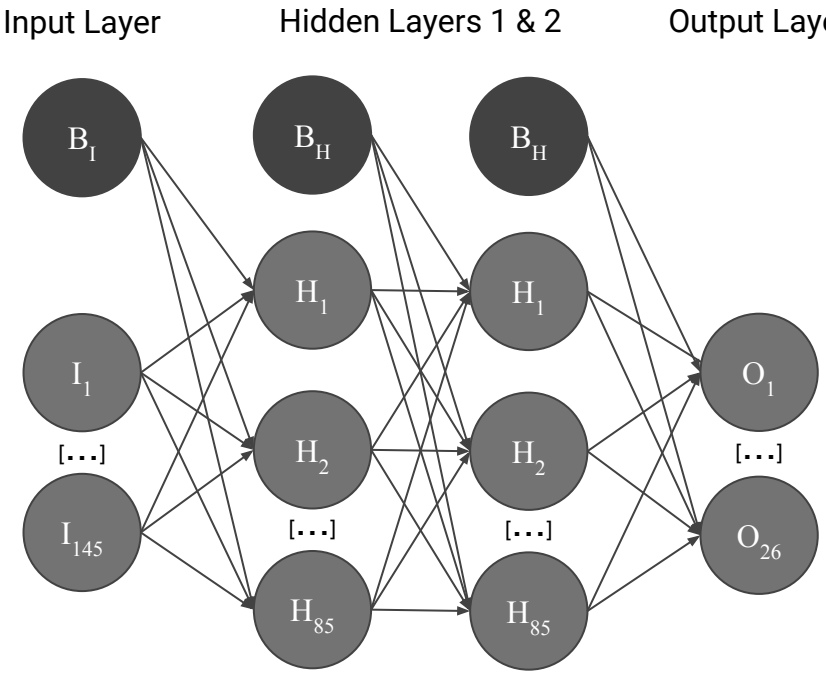
	A	B	C	D	E	F	G	H	I	J	K	L	M
1	0	0	0	0	0	0.6		0	0	0	0	0	
2	0	0	0	0	0		0.4	0	0	0	0	0	
3	0	0	0	0			0.9	0	0	0	0	0	
4	0	0	0			0	0		0	0	0	0	
5	0	0	0		0.2	0	0		0	0	0	0	
6	0	0			0	0	0	0			0	0	
7	0	0.1			0	0	0	0	0		0.2	0	
8	0											0	
9	0.1		0	0	0	0	0	0	0	0		0.2	
10			0	0	0	0	0	0	0	0			
11		0	0	0	0	0	0	0	0	0	0		
12		0	0	0	0	0	0	0	0	0	0		

	A	B	C	D	E	F	G	H	I	J	K	L
1										0	0	0
2		0	0	0	0	0	0	0		0.3	0	0
3		0	0	0	0	0	0	0			0	0
4		0	0	0	0	0	0			0.3	0	0
5		0	0	0	0	0.4				0	0	0
6							0.1	0	0	0	0	0
7											0.3	0
8		0	0	0	0	0	0	0	0	0.1		0.7
9		0	0	0	0	0	0	0	0	0	0.1	
10		0	0	0	0	0	0	0	0	0	0.1	
11		0	0	0	0	0.1						0.4
12								0.9	0	0	0	0

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	0	0	0	0.3						0.3	0	0	
2	0	0		0.9	0	0	0	0	0	0	0	0	
3	0.1			0	0	0	0	0	0	0	0	0	
4				0	0	0	0	0	0	0	0	0	
5				0	0	0	0	0	0	0	0	0	
6				0	0	0	0	0	0	0	0	0	
7				0	0	0	0	0	0	0	0	0	
8				0	0	0	0	0	0	0	0	0	
9	0			0	0	0	0	0	0	0	0	0	
10	0	0.2		0.4	0	0	0	0	0	0	0	0	
11	0	0	0.6			0	0	0	0	0	0	0.4	
12	0	0	0	0									
13													

Classification Algorithm

Neural Network Diagram

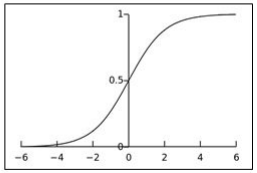


Regular Usage

$$H_j = \sigma\left(w_{B_I H_j} + \sum_{i=1}^{85} I_n w_{I_i H_j}\right)$$

$$\sigma(s) = \frac{1}{1+e^{-s}}$$

$$O_k = \sigma\left(w_{B_H O_k} + \sum_{j=1}^{26} H_j w_{H_j O_k}\right)$$



Network Training

$$\delta O_k = O_k(E)\big(1 - O_k(E)\big)\big(T_k(E) - O_k(E)\big)$$

$$\delta H_j = H_j(E)\left(1 - H_j(E)\right)\sum_{k=1}^{26} w_{H_j O_k}\delta O_k$$

$$\Delta w_{I_i H_j} = \eta I_i(E)\delta H_j$$

$$\Delta w_{H_j O_k} = \eta I_i(E)\delta H_j$$

Training Routine

Trying to encode 145 independent variables to 26 dependent variables by hand is impossible, so an automatic learning routine is used

1. Every weight in the network is randomly assigned a value between -0.5 and 0.5.
2. Training examples are run, and the error, δOk, is calculated for the output layer.
3. The error, δHj , is calculated for the hidden layer using δOk.
4. Weights are adjusted accordingly based on δHj and δOk
5. Repeat steps 2 – 4 until a termination condition is met.

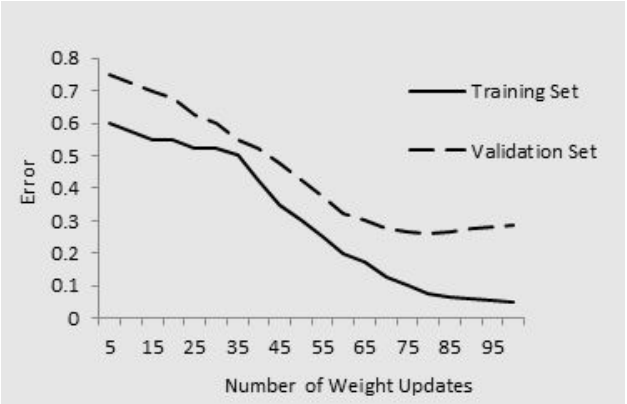
Network Training Equations

$$\delta O_k = O_k(E)\big(1 - O_k(E)\big)\big(T_k(E) - O_k(E)\big)$$

$$\delta H_j = H_j(E)\left(1 - H_j(E)\right)\sum_{k=1}^{26} w_{H_j O_k}\delta O_k$$

$$\Delta w_{I_i H_j} = \eta I_i(E)\delta H_j$$

$$\Delta w_{H_j O_k} = \eta I_i(E)\delta H_j$$



Overfitting considerations are taken into account

Demo