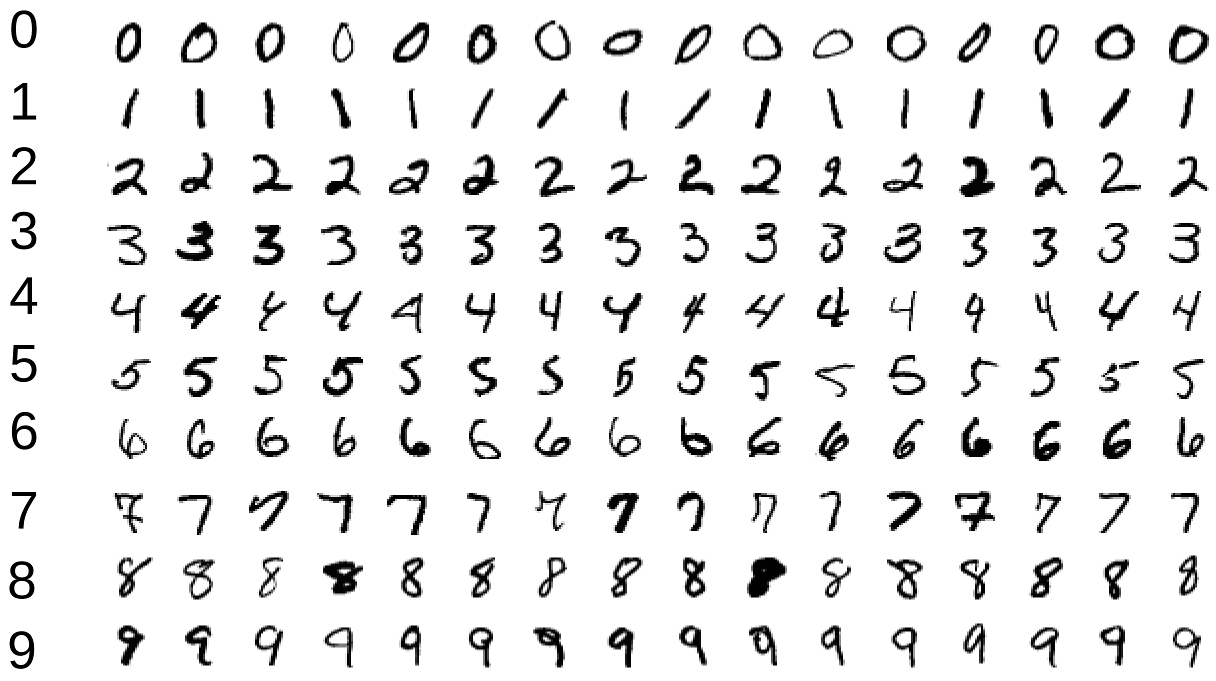
Parametric Approach

Parametric Approach: MNIST



- **10** labels
- **60,000** training images
- **10,000** test images
- Each image is an array of size **28 x 28 = 784** numbers total

Parametric Approach: Linear Classifier

Image

[28x28]
array of numbers
0...255
(784 numbers total)

10x1 **10x784** **784x1**

$$\boxed{f(x, W)} = \boxed{W}\boxed{x}$$

10x1

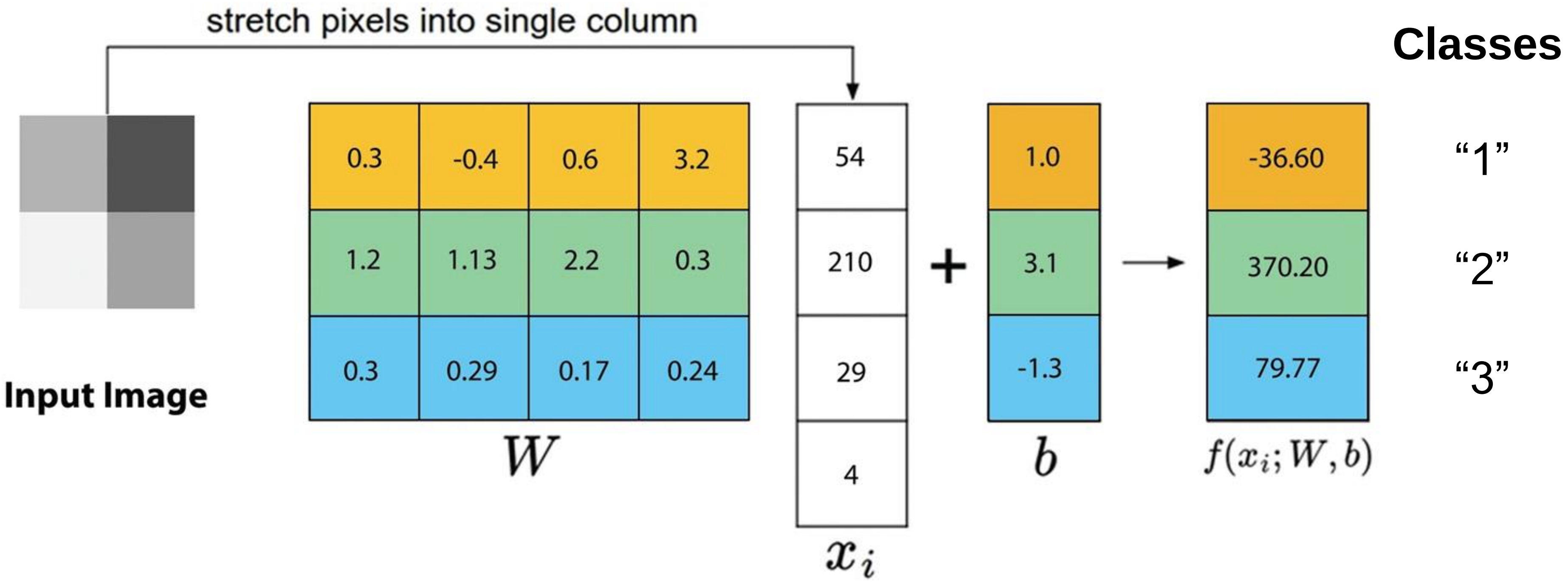
$\boxed{(+b)}$

10 numbers, indicating class scores

Input image

weights or parameters

Example with an Image with 4 Pixels, and 3 Classes (1/2/3)

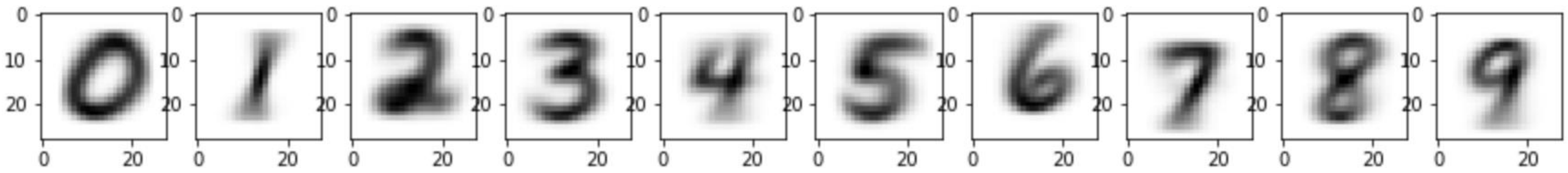


Interpreting a Linear Classifier

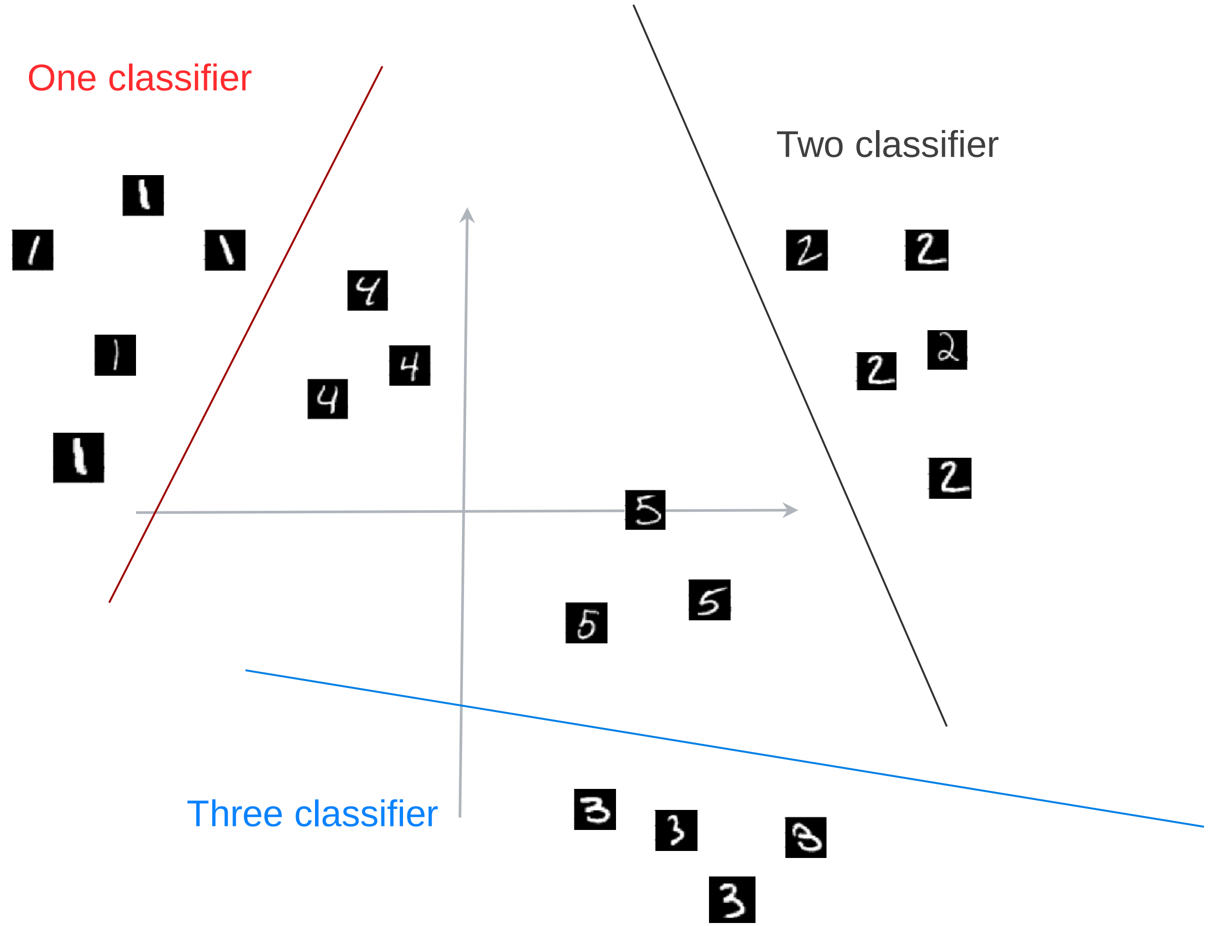
10x784

$$f(x_i, \boxed{W}, b) = Wx_i + b$$

Example trained weights of a linear classifier trained on MNIST:



Interpreting a Linear Classifier



$$f(x_i, W, b) = Wx_i + b$$

[28x28]
array of numbers 0…255
(784 numbers total)

We Defined a (Linear) Scoring Function:

$$f(x_i, W, b) = Wx_i + b$$



Example class scores for 3 images, with a random W:

0	-3.45	-0.51	3.42
1	**3.15**	1.1	2.3
2	5.3	**4.6**	1.9
3	-2.1	2.0	**-3.1**
4	4.48	-4.19	2.64
5	8.02	3.58	5.55
6	3.78	4.49	-4.34
7	1.06	-4.37	-1.5
8	-0.36	-2.09	-4.79
9	-0.72	-2.93	6.14

Going forward: Loss function / Optimization

Example class scores for 3 images, with a random W:

0	-3.45	-0.51	3.42
1	**3.15**	1.1	2.3
2	5.3	**4.6**	1.9
3	-2.1	2.0	**-3.1**
4	4.48	-4.19	2.64
5	8.02	3.58	5.55
6	3.78	4.49	-4.34
7	1.06	-4.37	-1.5
8	-0.36	-2.09	-4.79
9	-0.72	-2.93	6.14

1. Define a loss function that quantifies our unhappiness with the scores across the training data.

2. Come up with a way of efficiently finding the parameters (**W**, **b**) that minimize the loss function. (optimization)

3 training examples, 3 classes:
For some W the scores of $f(x, W) = Wx$ are:

			
one	**3.15**	1.1	2.3
two	5.3	**4.6**	1.9
three	-2.1	2.0	**-3.1**

Softmax Classifier (Multinomial Logistic Regression)

one	**3.15**	*scores = unnormalized log probabilities of the classes*
two	5.3	
three	-2.1	

Softmax Classifier (Multinomial Logistic Regression)

$$P(Y=k|X=x_i)=\boxed{\frac{e^{s_k}}{\sum_j e^{s_j}}} \quad \text{where} \quad s=f(x_i;W)$$

Softmax function

one	**3.15**
two	5.3
three	-2.1

scores = unnormalized log probabilities of the classes

Softmax Classifier (Multinomial Logistic Regression)

one	**3.15**
two	5.3
three	-2.1

scores = unnormalized log probabilities of the classes

$$P(Y=k|X=x_i) = \frac{e^{s_k}}{\sum_j e^{s_j}}$$ where $$s = f(x_i; W)$$

(is 1 (and 0 otherwise) if and only if sample belongs to class

We would like to maximize the log likelihood of correct class, i.e. decrease the negative log likelihood of the correct class :

$$L_i = -\log P(Y = y_i | X = x_i)$$

In summary: $L_i = -y_i \cdot log\left(\frac{e^{s_i}}{\sum_j e^{s_j}}\right)$ $L = \sum_i L_i$

Softmax Classifier (Multinomial Logistic Regression)

one	**3.15**
two	5.3
three	-2.1

unnormalized log probabilities

$$s = f(x_i; W)$$

$$L_i = -y_i \cdot log\left(\frac{e^{s_i}}{\sum_j e^{s_j}}\right) \quad , y = [1, 0, 0]^T$$

Softmax Classifier (Multinomial Logistic Regression)

$$\boxed{s = f(x_i; W)}$$

$$L_i = -y_i \cdot log\left(\frac{\boxed{e^{s_i}}}{\sum_j e^{s_j}}\right) \quad , y = [1, 0, 0]^T$$

		exp →	
one	**3.15**		**23.33**
two	5.3		200.3
three	-2.1		12
	unnormalized log probabilities		*unnormalized probabilities*

Softmax Classifier (Multinomial Logistic Regression)

, y=[1, 0, 0] ↑T

$$L_i = -y_i \cdot log\left(\boxed{\frac{e^{s_i}}{\sum_j e^{s_j}}}\right), y = [1,0,0]^T$$

	unnormalized log probabilities	exp →	unnormalized probabilities	normalize →	probabilities
one	**3.15**		**23.33**		**0.104**
two	5.3		200.3		0.895
three	-2.1		12		0.00

Softmax Classifier (Multinomial Logistic Regression)

, y=[1, 0, 0]↑T

$$L_i = \boxed{-y_i \cdot log\left(\frac{e^{s_i}}{\sum_j e^{s_j}}\right)}, y = [1, 0, 0]^T$$

	unnormalized log probabilities	$\xrightarrow{\text{exp}}$	unnormalized probabilities	$\xrightarrow{\text{normalize}}$	probabilities
one	**3.15**		**23.33**		**0.104**
two	5.3		200.3		0.895
three	-2.1		12		0.00

L_i = -log(0.104)
= **0.982**

Softmax Classifier (Multinomial Logistic Regression)

, y=[1, 0, 0]↑T

$$L_i = \boxed{-y_i \cdot log\left(\frac{e^{s_i}}{\sum_j e^{s_j}}\right)}, y = [1,0,0]^T$$

	unnormalized log probabilities		unnormalized probabilities		probabilities
one	**3.15**	exp →	**23.33**	normalize →	**0.104**
two	5.3		200.3		0.895
three	-2.1		12		0.00

L_i = -log(0.104)
= **0.982**

Question:
What can be the Minimum/Maximum possible loss (L_i)?

Softmax Classifier (Multinomial Logistic Regression)

, y=[1, 0, 0]↑T

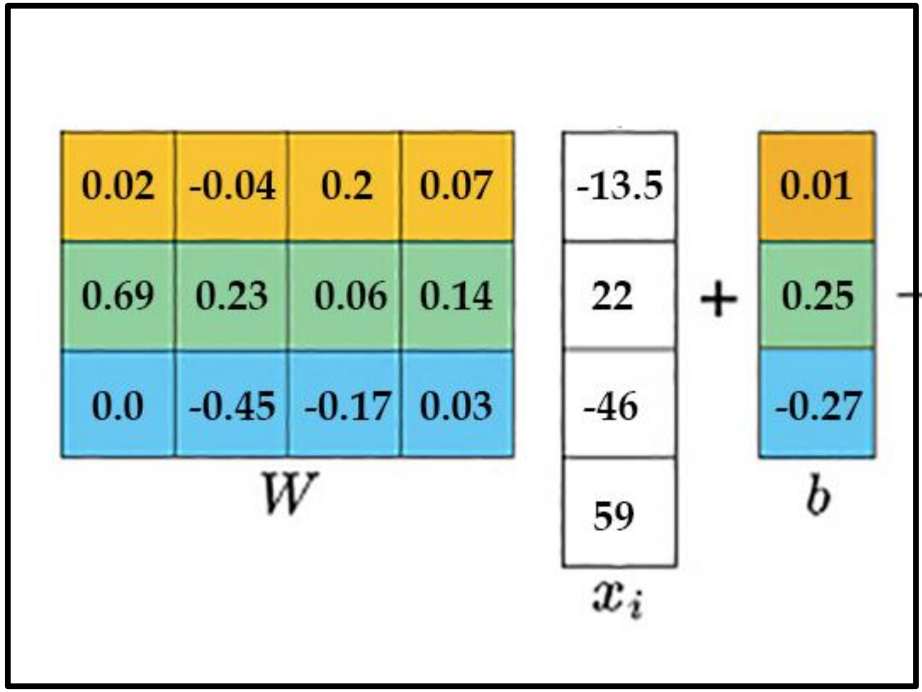
$$L_i = \boxed{-y_i \cdot log\left(\frac{e^{s_i}}{\sum_j e^{s_j}}\right)}, y = [1, 0, 0]^T$$

	unnormalized log probabilities	$\xrightarrow{\text{exp}}$	unnormalized probabilities	$\xrightarrow{\text{normalize}}$	probabilities
one	**3.15**		**23.33**		**0.104**
two	5.3		200.3		0.895
three	-2.1		12		0.00

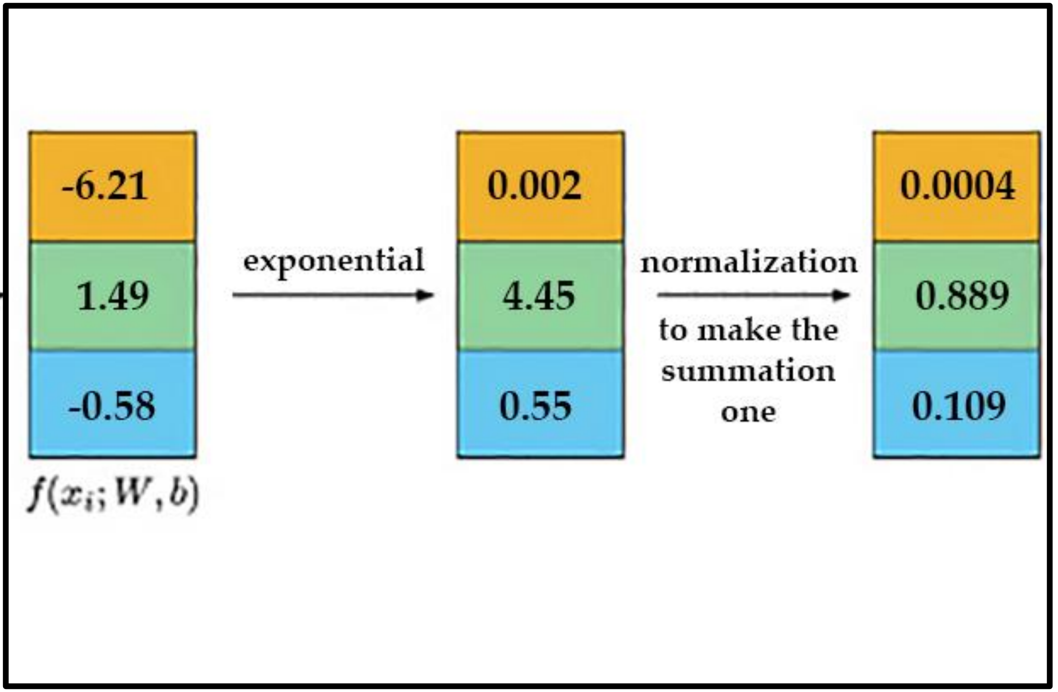
L_i = -log(0.104)
= **0.982**

Softmax Classifier (Multinomial Logistic Regression)

Matrix multiply plus bias offset



Cross-Entropy Loss



L_i = -log(0.104) = **0.982**

greatlearning
Learning for Life

Thank you!