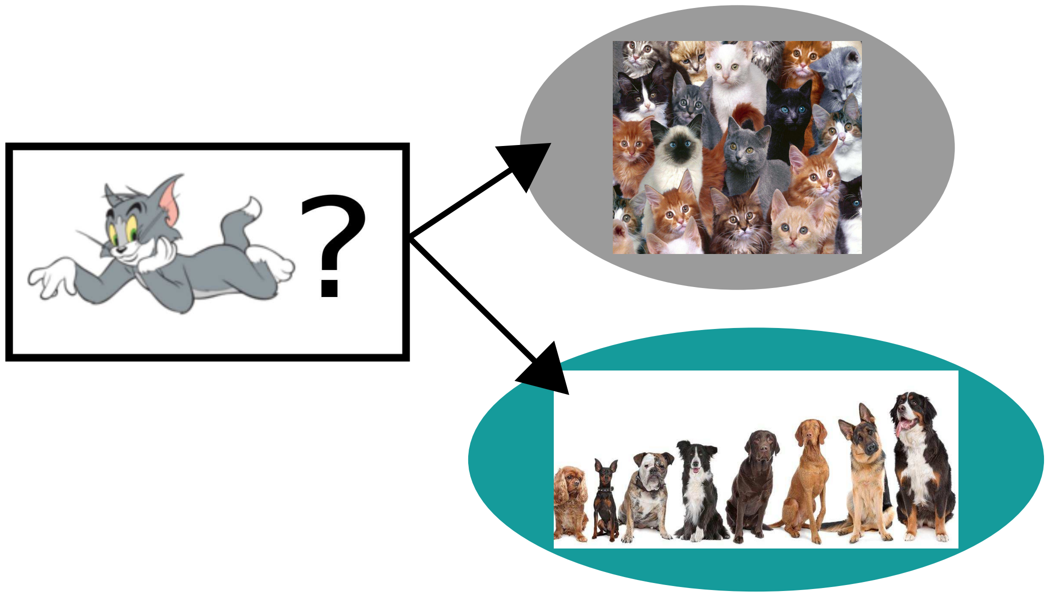
Perceptron lab
Computational Neuroscience

2021

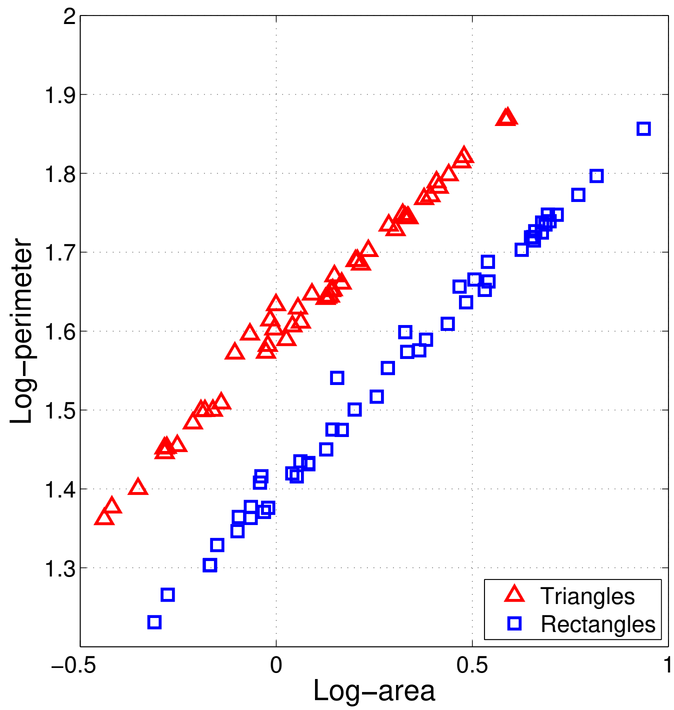
Instructor: Odelia Schwartz

Classification problem



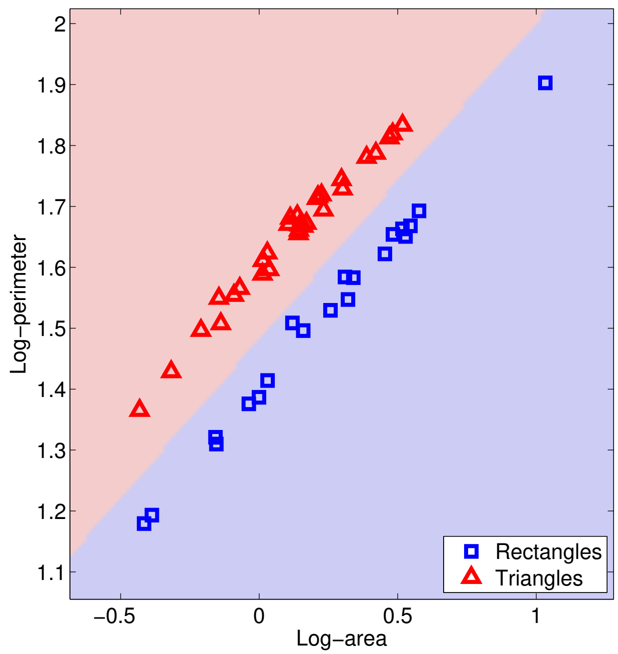
Classification problem

- Example: classify triangles versus rectangles



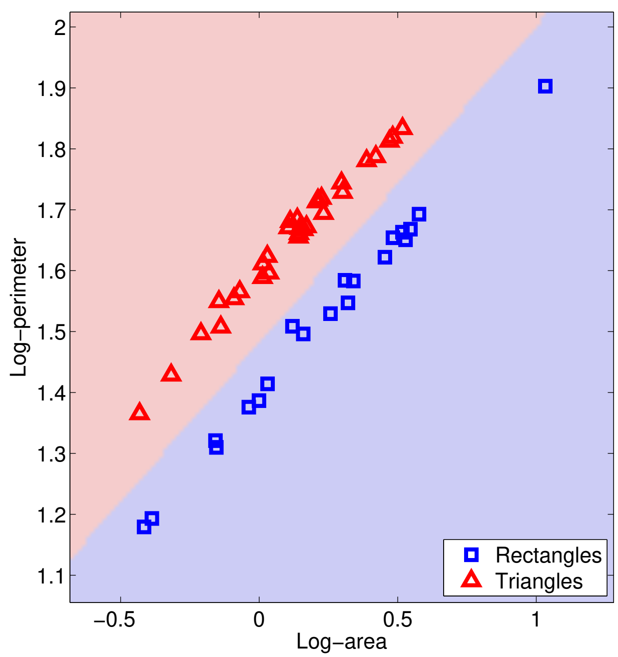
Classification problem

- Example: classify triangles versus rectangles
- They are linearly separable! Separate traingles from rectangles by dividing plane into two regions



Classification problem

- Example: classify triangles versus rectangles
- Main idea: Objects that belong together in the same class (e.g., triangle or rectangle) will have similar features (e.g., perimeter and area)



Perceptron Rosenblatt

THE NEW YORK TIMES, TUESDAY, JULY 8, 1958.

NEW NAVY DEVICE LEARNS BY DOING

Psychologist Shows Embryo of Computer Designed to Read and Grow Wiser

WASHINGTON, July 7 (UPI)—The Navy revealed the embryo of an electronic computer today that it expects will be able to walk, talk, see, write, reproduce itself and be conscious of its existence.

The embryo—the Weather Bureau's $2,000,000 "704" computer—learned to differentiate between right and left after fifty attempts in the Navy's demonstration for newsmen.

The service said it would use this principle to build the first of its Perceptron thinking machines that will be able to read and write. It is expected to be finished in about a year at a cost of $100,000.

Dr. Frank Rosenblatt, designer of the Perceptron, conducted the demonstration. He said the machine would be the first device to think as the human brain. As do human beings, Perceptron will make mistakes at first, but will grow wiser as it gains experience, he said.

Dr. Rosenblatt, a research psychologist at the Cornell Aeronautical Laboratory, Buffalo, said Perceptrons might be fired to the planets as mechanical space explorers.

Without Human Controls

The Navy said the perceptron would be the first non-living mechanism "capable of receiving, recognizing and identifying its surroundings without any human training or control."

The "brain" is designed to remember images and information it has perceived itself. Ordi-

Books of The Times

By CHARLES POORE

"IF this were an entirely accurate account of my life in Cork," the author of "Mrs. O' "* tells us, "I should probably be writing it behind bars. So I should say that it is impressionistically true when not always factually so."

Fair enough. However, when you have finished her entertaining book, you may want to go back to that preface and wonder whether the bit about *behind bars* is a pun or an Irish bull.

Why? Because she ran a pub in Cork. The idea of doing so came to her in London one afternoon when she found herself rather rich and completely free. "My decree absolute came through on the same day as my Great Aunt's legacy—not a fortune, but such a sum as I had never dreamed of owning or saving." The fact that she happened to choose for refreshment a place called Mooney's, in London, gave the notion a proper touch of predestination.

Once in Ireland she made forays around the country. It did not take her very long to find the pub she wanted in Cork and buy it from a maiden lady who did not appreciate its seedy elegance. What names she signed to the deed we do not know, although this book is copyrighted by C. M. Forde. As author of it she calls herself, with royal simplicity, Claude, just Claude.

Named by Irish Friends

It was her Irish friends and customers who gave her the name of Mrs. O'. A reference to herself, near the end of the book, as one who holds in reserve "the resignation to the inevitable that lingers in the blood of those born in fatalistic East," marks the beginning of a cosmopolitan outlook.

A beau sabreur named Sean soon spotted her as French in spite of a quickly acquired talent for Gaelic. And Claude tells us she has "drunk rye with Americans, schnapps with Dutchmen, beer with Germans, wine with Frenchmen, liqueurs with duchesses and gin with charladies." The charladies and the duchesses, presumably, carry interna-

Claude, author of "Mrs. O'."

witz with any attention, you would think of the long arid stretches with no hostelries."

Early on, as proprietor of her own pub, she had learned to tap a mighty keg of high stout, after just one lesson from a friendly rival called Foxy:

"Wrapping the barrel end of the tap in three thicknesses of newspaper as I had seen Foxy do, I placed the tap against the bung, raised the mallet, and thinking briefly that I should probably be the first foreigner ever to be killed by Guinness, I hit the tap two fairly light, quick blows."

Mallet's Force Augmented

It worked fine. The third whack was delivered at full strength. The tap went into place, the newspaper sealed the crack around it. One thing she was too shy to mention when congratulations, offered in awe, saluted her, was that she had, shall we say, augmented the force of the mallet with a huge horseshoe she had discovered under the bar.

A rather formally informal romance flowers in the book. It concerns Sean, whose past is a subject for gossip, and whose present is a matter of mystery. He is destined

Perceptron Rosenblatt (1957)

THE NEW YORK TIMES, TUESDAY, JULY 8, 1958.

NEW NAVY DEVICE LEARNS BY DOING

Psychologist Shows Embryo of Computer Designed to Read and Grow Wiser

WASHINGTON, July 7 (UPI)—The Navy revealed the embryo of an electronic computer today that it expects will be able to walk, talk, see, write, reproduce itself and be conscious of its existence.

The embryo—the Weather Bureau's $2,000,000 "704" computer—learned to differentiate between right and left after fifty attempts in the Navy's demonstration for newsmen.

The service said it would use this principle to build the first of its Perceptron thinking machines that will be able to read and write. It is expected to be finished in about a year at a cost of $100,000.

Dr. Frank Rosenblatt, designer of the Perceptron, conducted the demonstration. He said the machine would be the first device to think as the human brain. As do human beings, Perceptron will make mistakes at first, but will grow wiser as it gains experience, he said.

Dr. Rosenblatt, a research psychologist at the Cornell Aeronautical Laboratory, Buffalo, said Perceptrons might be fired to the planets as mechanical space explorers.

Without Human Controls

The Navy said the perceptron would be the first non-living mechanism "capable of receiving, recognizing and identifying its surroundings without any human training or control."

The "brain" is designed to remember images and information it has perceived itself. Ordi-

Books of The Times

By CHARLES POORE

"IF this were an entirely accurate account of my life in Cork," the author of "Mrs. O'"* tells us, "I should probably be writing it behind bars. So I should say that it is impressionistically true when not always factually so."

Fair enough. However, when you have finished her entertaining book, you may want to go back to that preface and wonder whether the bit about *behind bars* is a pun or an Irish bull.

Why? Because she ran a pub in Cork. The idea of doing so came to her in London one afternoon when she found herself rather rich and completely free. "My decree absolute came through on the same day as my Great Aunt's legacy—not a fortune, but such a sum as I had never dreamed of owning or saving." The fact that she happened to choose for refreshment a place called Mooney's, in London, gave the notion a proper touch of predestination.

Once in Ireland she made forays around the country. It did not take her very long to find the pub she wanted in Cork and buy it from a maiden lady who did not appreciate its seedy elegance. What names she signed to the deed we do not know, although this book is copyrighted by C. M. Forde. As author of it she calls herself, with royal simplicity, Claude, just Claude.

Named by Irish Friends

It was her Irish friends and customers who gave her the name of Mrs. O'. A reference to herself, near the end of the book, as one who holds in reserve "the resignation to the inevitable that lingers in the blood of those born in fatalistic East," marks the beginning of a cosmopolitan outlook.

A beau sabreur named Sean soon spotted her as French in spite of a quickly acquired talent for Gaelic. And Claude tells us she has "drunk rye with Americans, schnapps with Dutchmen, beer with Germans, wine with Frenchmen, liqueurs with duchesses and gin with charladies." The charladies and the duchesses, presumably, carry interna-

Claude, author of "Mrs. O'."

witz with any attention, you would think of the long arid stretches with no hostelries."

Early on, as proprietor of her own pub, she had learned to tap a mighty keg of high stout, after just one lesson from a friendly rival called Foxy:

"Wrapping the barrel end of the tap in three thicknesses of newspaper as I had seen Foxy do, I placed the tap against the bung, raised the mallet, and thinking briefly that I should probably be the first foreigner ever to be killed by Guinness, I hit the tap two fairly light, quick blows."

Mallet's Force Augmented

It worked fine. The third whack was delivered at full strength. The tap went into place, the newspaper sealed the crack around it. One thing she was too shy to mention when congratulations, offered in awe, saluted her, was that she had, shall we say, augmented the force of the mallet with a huge horseshoe she had discovered under the bar.

A rather formally informal romance flowers in the book. It concerns Sean, whose past is a subject for gossip, and whose present is a matter of mystery. He is destined

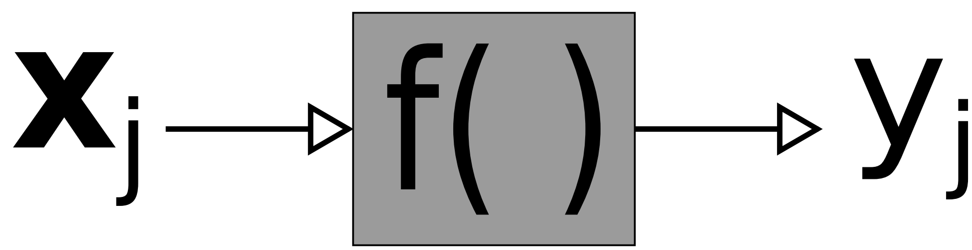
Books

- New York times 1958: "The navy reavaled the embryo of an electronic computer that it expects will be able to walk, talk, write, reproduce itself and be conscious of its existence

Perceptron Rosenblatt (1957)

- One of the earliest models for learning with supervision

Perceptron Rosenblatt



Perceptron Rosenblatt

$$\mathbf{x}_j \longrightarrow \boxed{f(\ )} \longrightarrow y_j$$

$$f(\mathbf{x}_j) = \operatorname{sign}\left(\sum_{i=1}^{d} w_i x_{ij} + b\right)$$

Feature i
Sample j

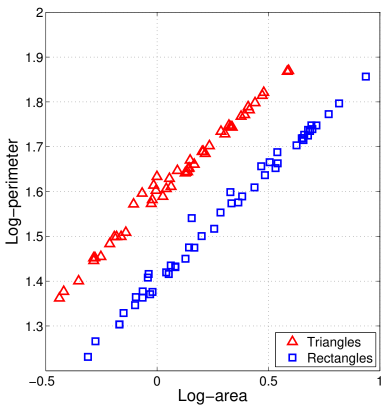
Perceptron Rosenblatt

$$\mathbf{x}_j \longrightarrow \boxed{f(\ )} \longrightarrow y_j$$

$$f(\mathbf{x}_j) = \mathrm{sign}\left(\sum_{i=1}^{d} w_i x_{ij} + b\right)$$

Feature i
Sample j

Perceptron Rosenblatt

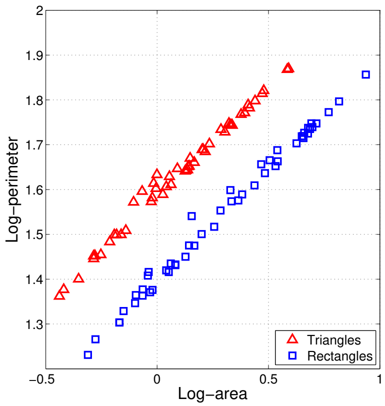


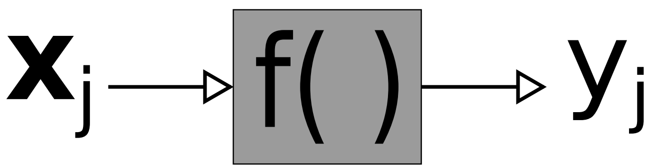
$$\mathbf{x}_j \longrightarrow \boxed{f(\;)} \longrightarrow y_j$$

$$f(\mathbf{x}_j) = \operatorname{sign}\left(\sum_{i=1}^{d} w_i x_{ij} + b\right)$$

Feature i. How many features in our example?
Sample j.

Perceptron Rosenblatt

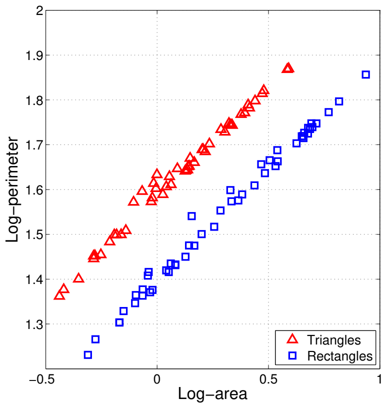


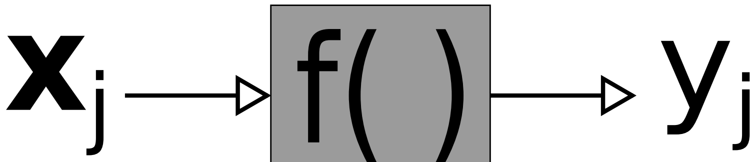


$$f(\mathbf{x}_j) = \operatorname{sign}\left(\sum_{i=1}^{d} w_i x_{ij} + b\right)$$

Feature i. How many features in our example?
Two features (triangles and squares). So d=2

Perceptron Rosenblatt





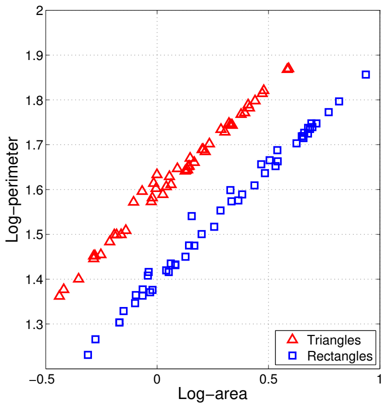
- In our example two features; i=2:

$$y = sign(w_1 x_1 + w_2 x_2 + b)$$

- We dropped the sample j for simplicity

Feature i. How many features in our example?
In our example i=2 features

Perceptron Rosenblatt

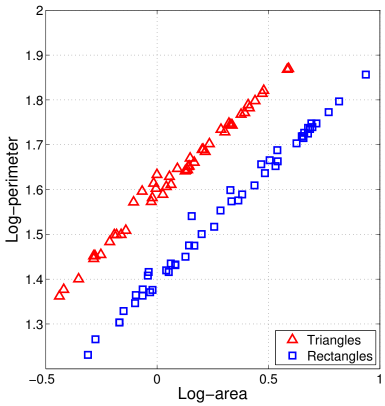


$$\mathbf{x}_j \longrightarrow f(\ ) \longrightarrow y_j$$

$$f(\mathbf{x}_j) = \operatorname{sign}\left(\sum_{i=1}^{d} w_i x_{ij} + b\right)$$

Sample j. How many samples in our example?

Perceptron Rosenblatt

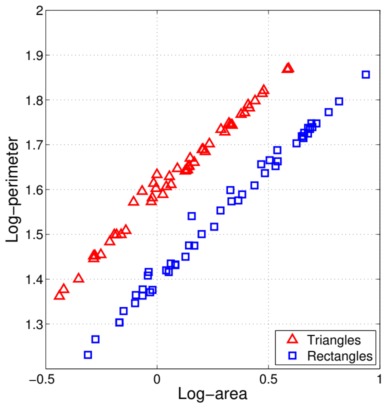


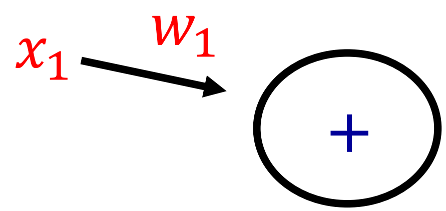
$$\mathbf{x}_j \longrightarrow \boxed{f(\ )} \longrightarrow y_j$$

$$f(\mathbf{x}_j) = \operatorname{sign}\left(\sum_{i=1}^{d} w_i x_{ij} + b\right)$$

Sample j. How many samples in our example?
The number of triangles or squares

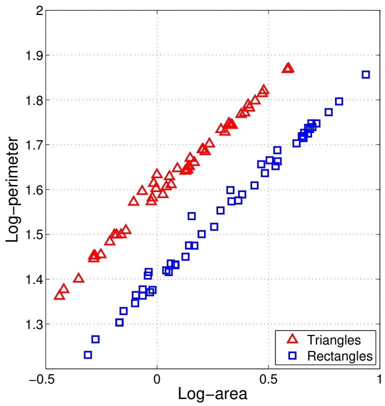
Perceptron Rosenblatt

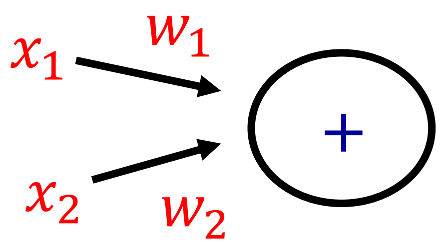




$$y = sign(w_1 \mathrm{x}_1 + \mathrm{w}_2 \mathrm{x}_2 + \mathrm{b})$$

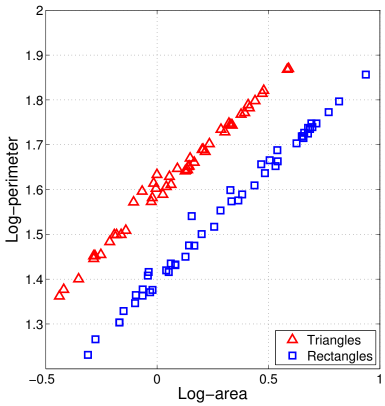
Perceptron Rosenblatt

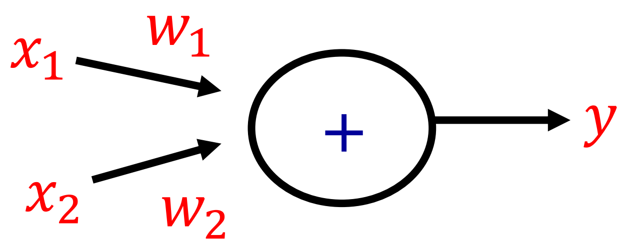




$$y = sign(w_1 \mathrm{x}_1 + \mathrm{w}_2 \mathrm{x}_2 + \mathrm{b})$$

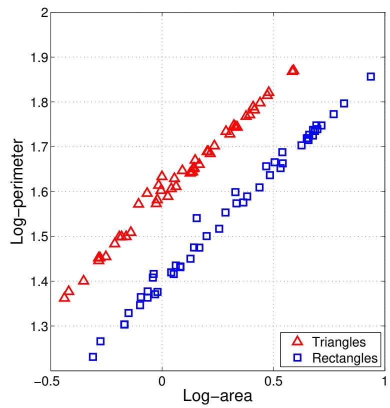
Perceptron Rosenblatt

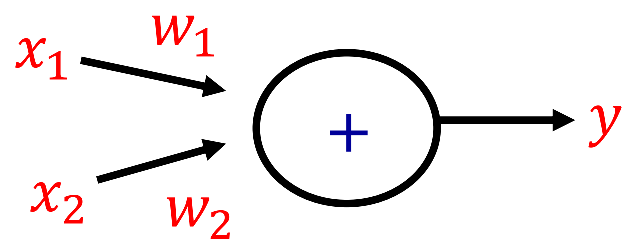




$$y = sign(w_1 \mathrm{x}_1 + \mathrm{w}_2 \mathrm{x}_2 + \mathrm{b})$$

Perceptron Rosenblatt

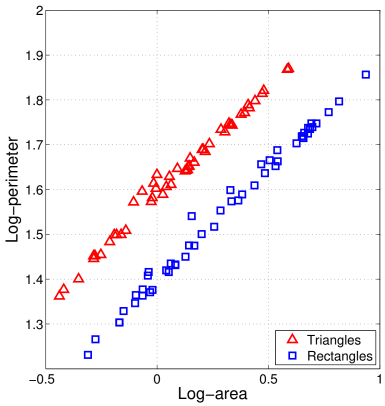


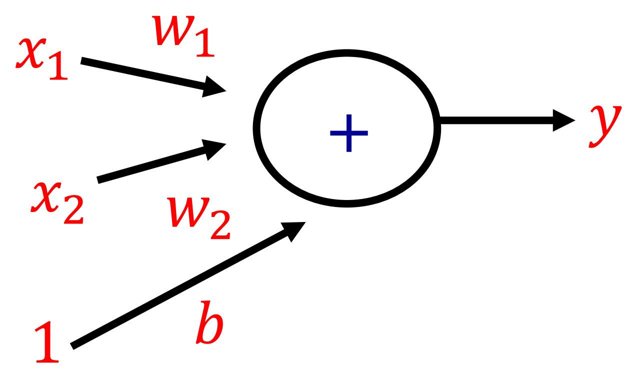


What about the b? Why do we need it?

$$y = sign(w_1 \mathrm{x}_1 + \mathrm{w}_2 \mathrm{x}_2 + \mathrm{b})$$

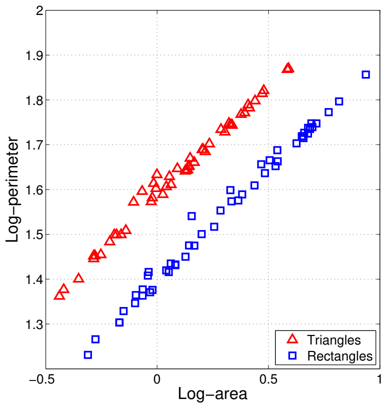
Perceptron Rosenblatt

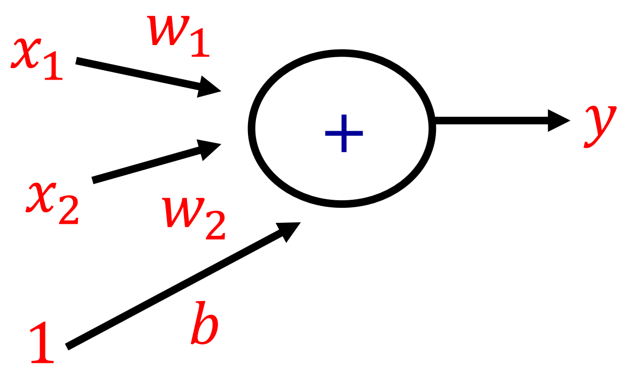




$$y = sign(w_1 x_1 + w_2 x_2 + b)$$

Perceptron Rosenblatt

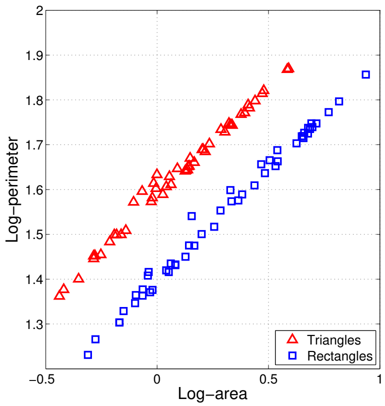


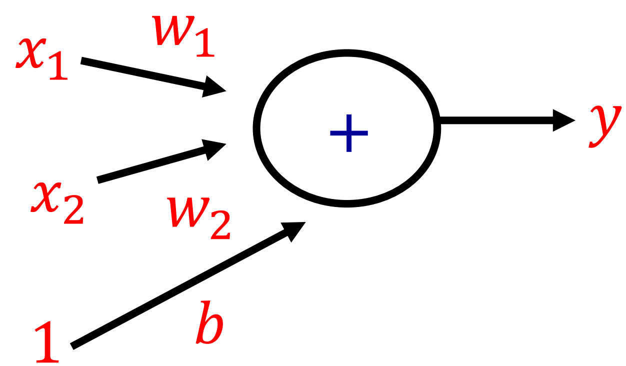


$$y = sign(w_1 x_1 + w_2 x_2 + b)$$

- Why the sign?

Perceptron Rosenblatt

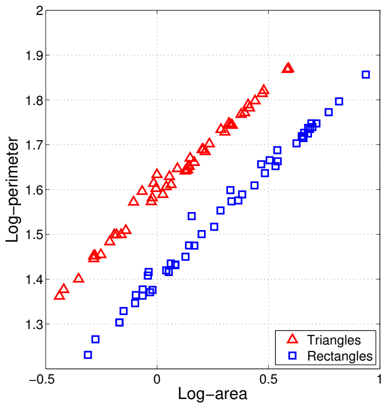




$$y = sign(w_1 \mathrm{x}_1 + \mathrm{w}_2 \mathrm{x}_2 + \mathrm{b})$$

- Why the sign? Class is set as either +1 or -1 (positive or negative)
- Corresponds here to triangles versus squares

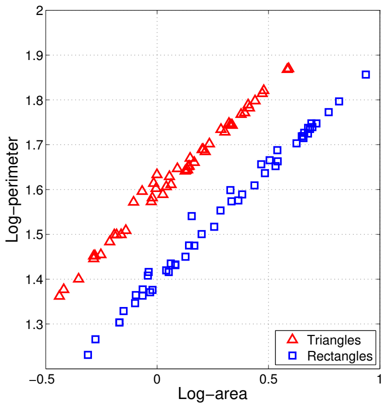
Perceptron Rosenblatt



$$y = sign(w_1 x_1 + w_2 x_2 + b)$$

- Changing the value of the wi and b give us different functions

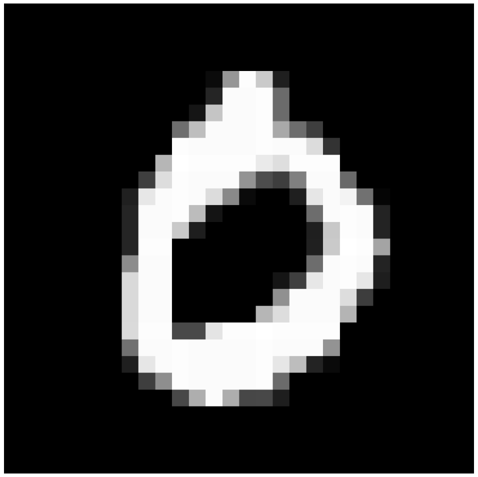
Perceptron Rosenblatt

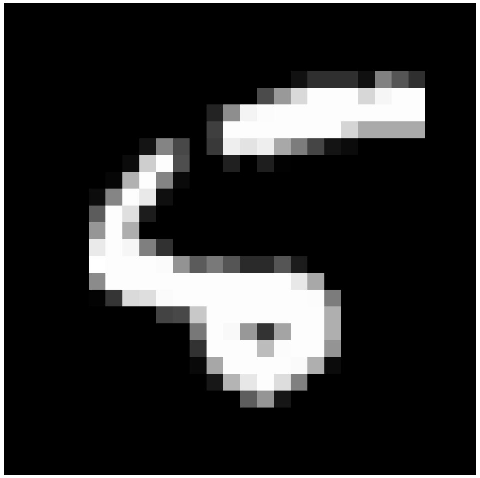


$$y = sign(w_1 x_1 + w_2 x_2 + b)$$

- Changing the value of the wi and b give us different functions
- Learning amounts to finding the values of wi and b that "best capture" the input output relationship (i.e., that best separates the two classes)

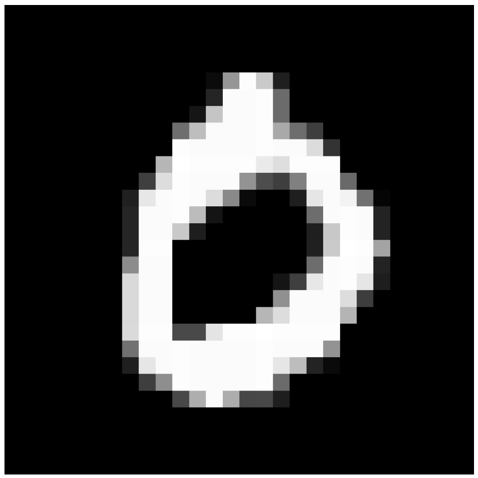
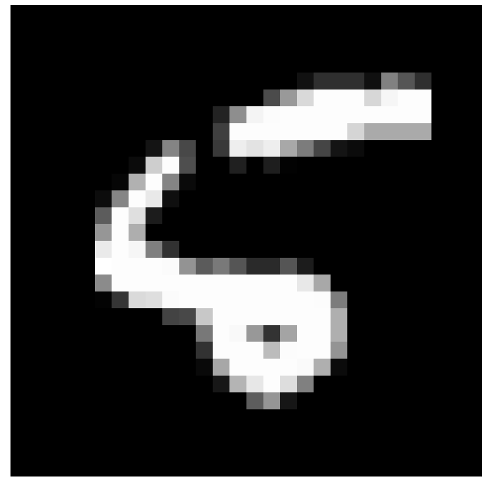
MNIST database





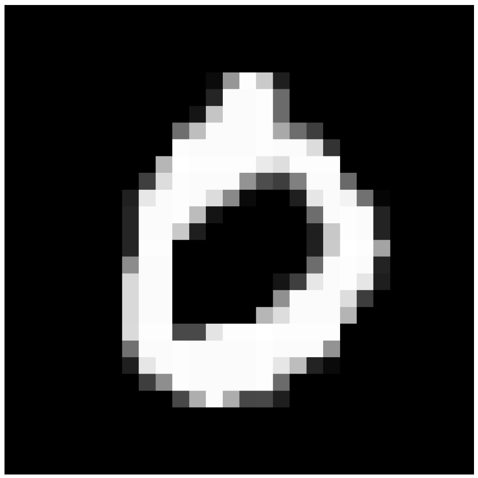
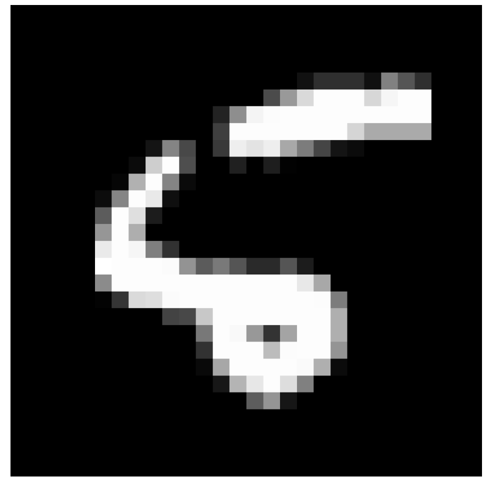
- We’ll look at the famous mnist database

MNIST database

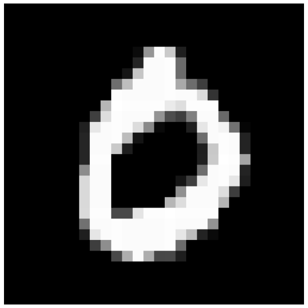
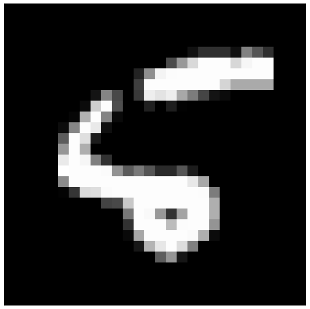
- We'll look at the famous mnist database
- Each digit is size 28 x 28 pixels = 784

MNIST database

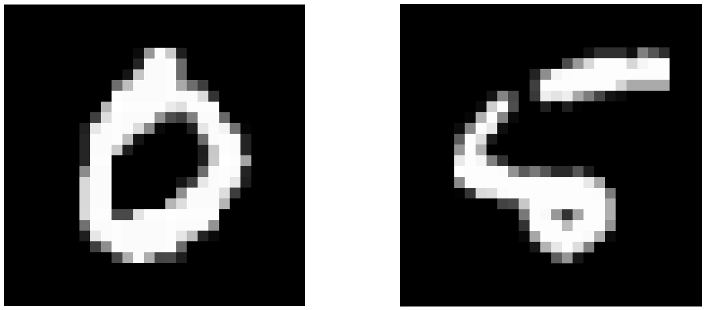
- Each digit is size 28 x 28 pixels = 784
- Set 0 to positive sign label (+1)
  Set 5 to negative sign label (-1)

MNIST database

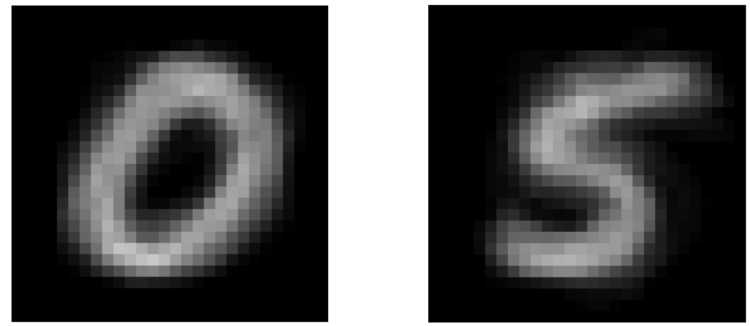
 

- We need “features” …

MNIST database



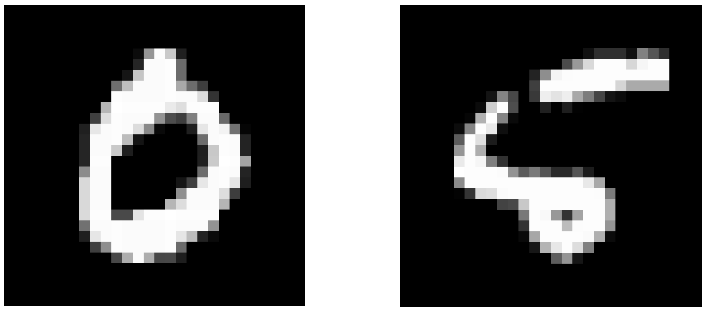
Example digit samples



Mean of each of the digits

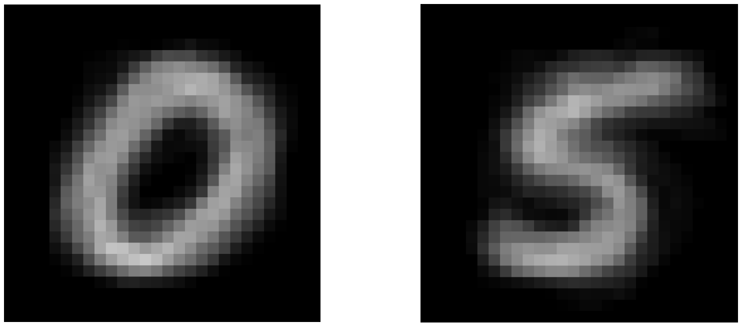
- We need “features” …
- We’ll project (dot product) each input sample onto the mean of the two classes (how similar is each input to the mean)

MNIST database



Example digit samples

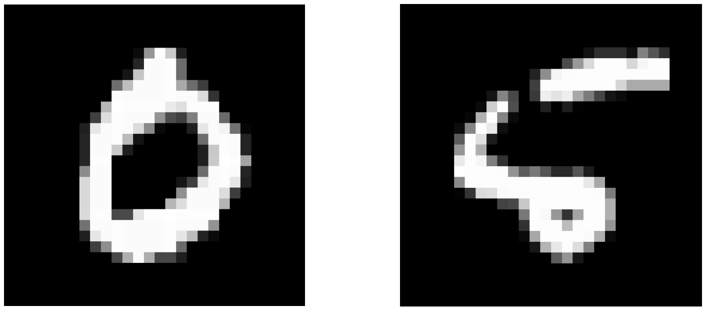
$v(1,:)$



Mean of each of the digits

$v(2,:)$

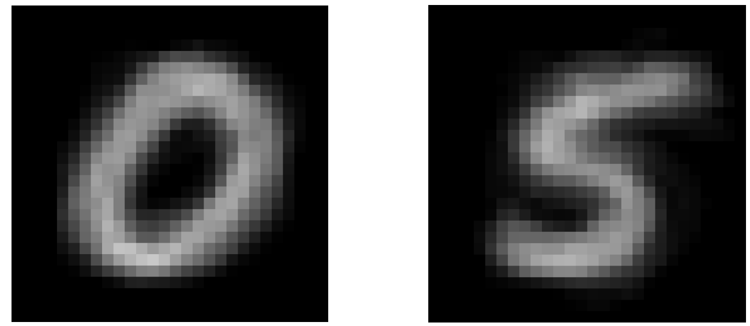
MNIST database



Example digit samples

$v(1,:)$

Dimensionality 1 x 784

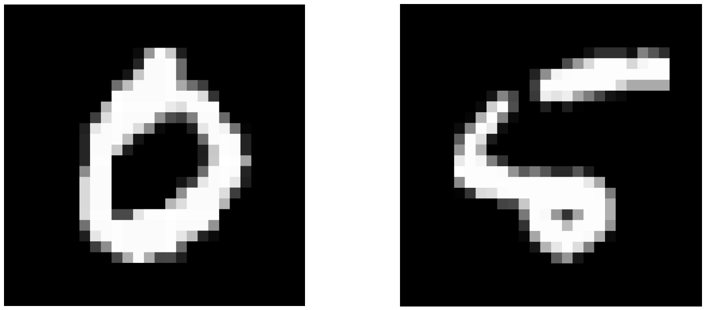


Mean of each of the digits

$v(2,:)$

Dimensionality 1 x 784

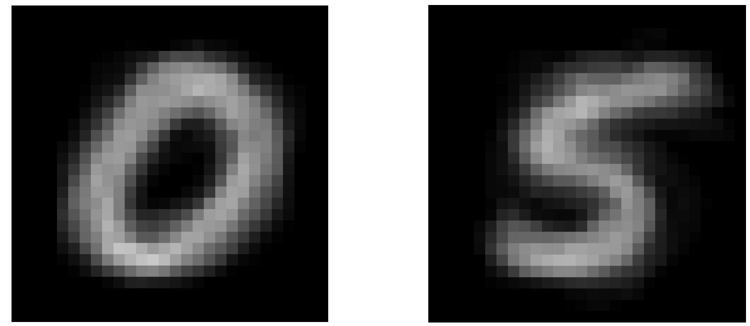
MNIST database



Example digit samples

$v(1,:)$

Dimensionality 1 x 784



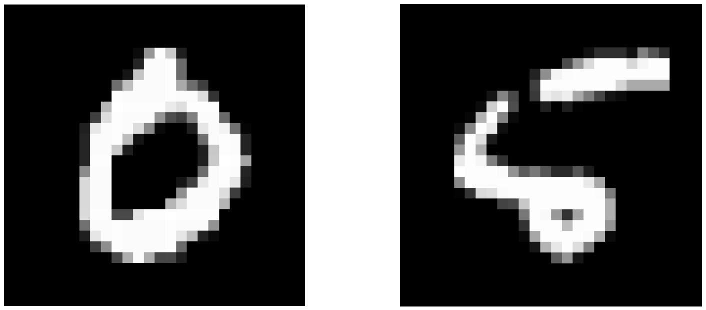
Mean of each of the digits

$v(2,:)$

Dimensionality 1 x 784

v: 2 x 784 (we now have two features)

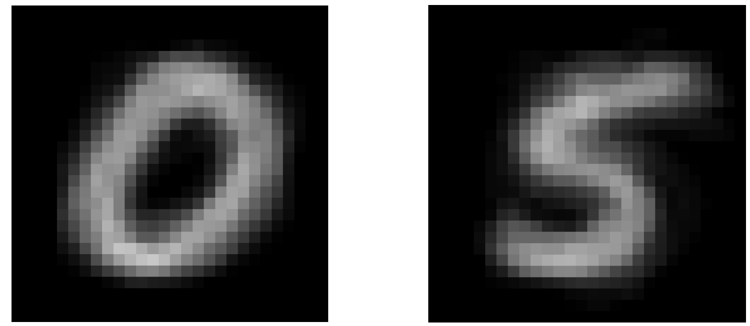
MNIST database



Example digit samples

$v(1,:)$

Dimensionality 1 x 784



Mean of each of the digits

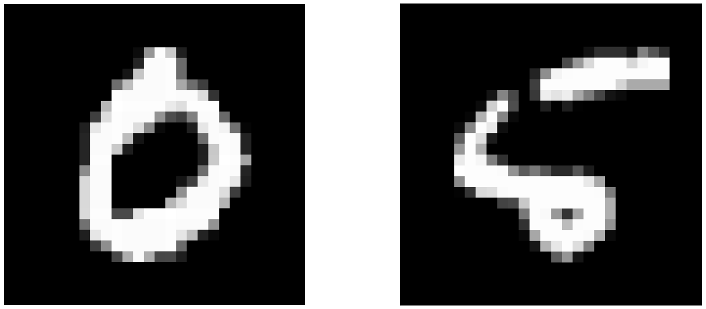
$v(2,:)$

Dimensionality 1 x 784

Input x: 60 samples x 784

v: 2 x 784

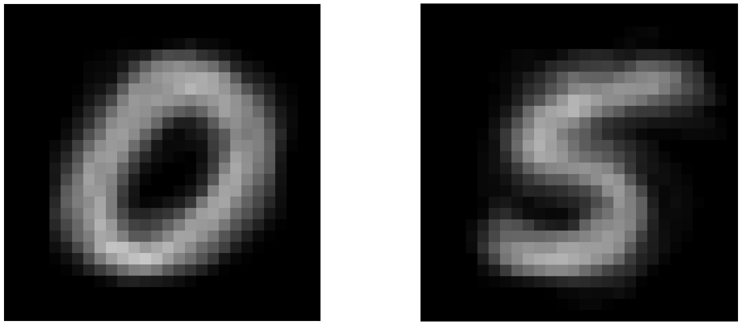
MNIST database



Example digit samples

$v(1,:)$

Dimensionality 1 x 784



Mean of each of the digits

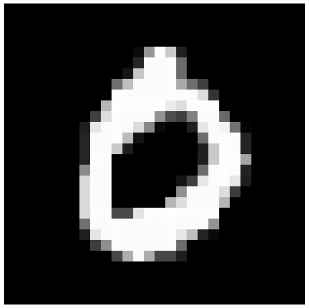
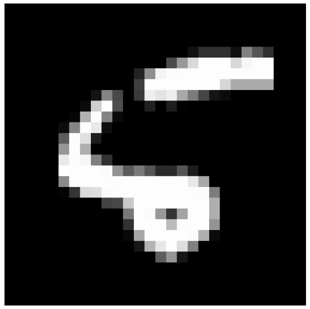
$v(2,:)$

Dimensionality 1 x 784

Input x: 60 samples x 784

v: 2 x 784

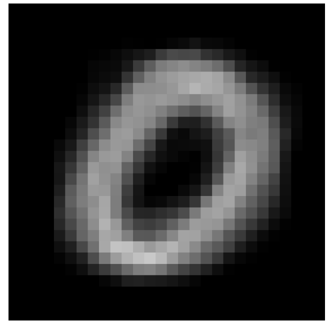
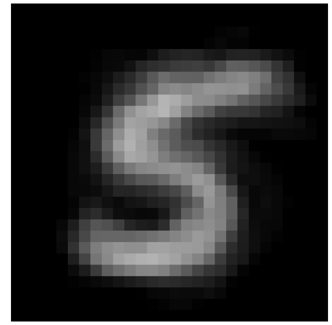
MNIST database

Example digit samples

$v(1,:)$

Dimensionality 1 x 784

Mean of each of the digits

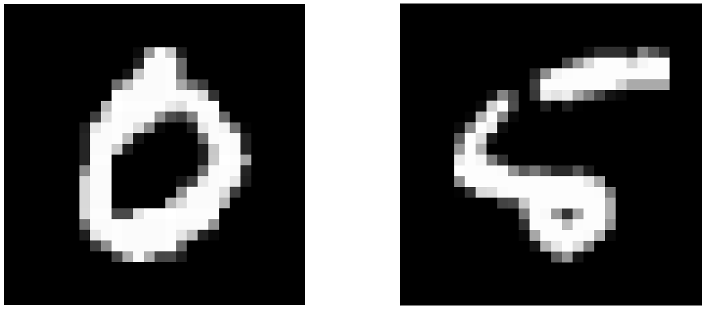
$v(2,:)$

Dimensionality 1 x 784

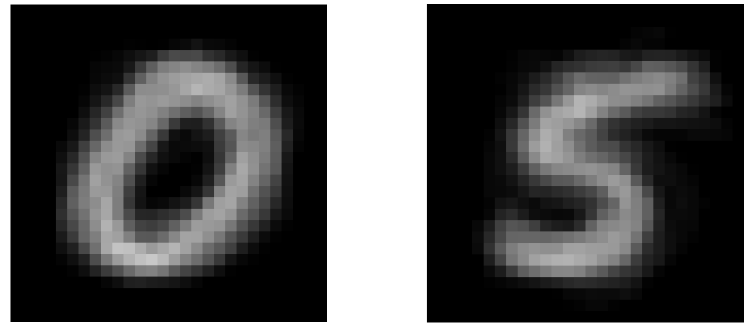
x: 60 samples x 784
v: 2 x 784

$z = x * v' =$ 60 x 784 * 784 x 2 = 60 x 2

MNIST database



Example digit samples



Mean of each of the digits

$v(1,:)$

Dimensionality 1 x 784

$v(2,:)$

Dimensionality 1 x 784

x: 60 samples x 784
v: 2 x 784

$z = x * v' =$ 60 x 784 * 784 x 2 = 60 x 2

So input now will be 60 samples by 2 features

Perceptron learning

$$f(\mathbf{x}_j) = \mathrm{sign}\left(\sum_{i=1}^{d} w_i x_{ij} + b\right).$$

- Need to learn w and b

Perceptron learning

Consist of the following steps:

1. Let $j = 1$, and initialize $\{w_i\}_{i=1}^{d}$ and b ($\mathbf{w}$ and b can be initialized with zeros).
2. Compute
$$f(\mathbf{x}_j) = \mathrm{sign}\left(\sum_{i=1}^{d} w_i x_{ij} + b\right).$$

Perceptron learning

Consist of the following steps:

1. Let $j = 1$, and initialize $\{w_i\}_{i=1}^{d}$ and b ($\mathbf{w}$ and b can be initialized with zeros).
2. Compute
$$f(\mathbf{x}_j) = \mathrm{sign}\left(\sum_{i=1}^{d} w_i x_{ij} + b\right).$$
3. If $f(\mathbf{x}_j)$ is NOT equal to y_j, update $\mathbf{w}$ and b as follows:
$$\begin{aligned} w_i &\leftarrow w_i + y_j x_{ij} \\ b &\leftarrow b + y_j \end{aligned}$$

Perceptron learning

Consist of the following steps:

1. Let $j = 1$, and initialize $\{w_i\}_{i=1}^{d}$ and b ($\mathbf{w}$ and b can be initialized with zeros).

2. Compute

$$f(\mathbf{x}_j) = \operatorname{sign}\left(\sum_{i=1}^{d} w_i x_{ij} + b\right). \tag{2}$$

3. If $f(\mathbf{x}_j)$ is NOT equal to y_j, update $\mathbf{w}$ and b as follows:

$$w_i \leftarrow w_i + y_j x_{ij} \tag{3}$$

$$b \leftarrow b + y_j \tag{4}$$

4. Repeat 2 for next value of j, that is, $j = j + 1$. When $j = N$, restart the counter to $j = 1$. If for all $j = 1, \ldots, N$, $f(\mathbf{x}_j)$ is equal to y_j, stop iterating.

We repeat for 100 epochs

Perceptron learning

- Why does it work?

→ Suppose we updates w and b using (x_J, y_J)

$\hat{w}, \hat{b}$ before update

w, b after update

Perceptron learning

- Why does it work?

→ Suppose we updates w and b using (x_J, y_J)

$\hat{w}, \hat{b}$ before update

w, b after update

$$\sum_{i=1}^{d} w_i x_{iJ} + b = \sum_{i=1}^{d} \big(\overbrace{\hat{w}_i + y_J x_{iJ}}^{\text{new weight}}\big) x_{iJ} + \overbrace{\hat{b} + y_J}^{\text{new } b}$$

Perceptron learning

- Why does it work?

→ Suppose we updates w and b using (x_J, y_J)

$\hat{w}, \hat{b}$ before update

w, b after update

$$\sum_{i=1}^{d} w_i x_{iJ} + b = \sum_{i=1}^{d} \Big(\overbrace{\hat{w}_i + y_J x_{iJ}}^{\text{new weight}}\Big) x_{iJ} + \overbrace{\hat{b} + y_J}^{\text{new } b}$$

Perceptron learning

- Why does it work?

→ Suppose we updated w and b using (x_J, y_J)

$\hat{w}, \hat{b}$ before update

w, b after update

$$\sum_{i=1}^{d} w_i x_{iJ} + b = \sum_{i=1}^{d} \big(\underbrace{\hat{w}_i + y_J x_{iJ}}_{\text{new weight}}\big) x_{iJ} + \underbrace{\hat{b} + y_J}_{\text{new } b}$$

$$= \sum_{i=1}^{d} \underline{\hat{w}_i x_{iJ}} + y_J x_{iJ} x_{iJ} + \underline{\hat{b}} + y_J$$

Perceptron learning

→ Suppose we updates w and b using (x_J, y_J)

$\hat{w}, \hat{b}$ before update

w, b after update

$$\sum_{i=1}^{d} w_i x_{iJ} + b = \sum_{i=1}^{d} \Big(\overbrace{\hat{w}_i + y_J x_{iJ}}^{\text{new weight}}\Big) x_{iJ} + \overbrace{\hat{b} + y_J}^{\text{new } b}$$

$$= \sum_{i=1}^{d} \underline{\hat{w}_i x_{iJ}} + y_J x_{iJ} x_{iJ} + \underline{\hat{b}} + y_J$$

$$= \sum_{i=1}^{d} \hat{w}_i x_{iJ} + \hat{b} + y_J \Big(\sum_{i=1}^{d} \underbrace{x_{iJ} x_{iJ}}_{\text{positive}} + 1 \Big)$$

Perceptron learning

- Why does it work?

→ Suppose we update w and b using (x_J, y_J)

$\hat{w}, \hat{b}$ before update

w, b after update

$$\sum_{i=1}^{d} w_i x_{iJ} + b = \sum_{i=1}^{d} (\overbrace{\hat{w}_i + y_J x_{iJ}}^{\text{new weight}}) x_{iJ} + \overbrace{\hat{b} + y_J}^{\text{new b}}$$

$$= \sum_{i=1}^{d} \underline{\hat{w}_i x_{iJ}} + y_J x_{iJ} x_{iJ} + \underline{\hat{b}} + y_J$$

$$= \sum_{i=1}^{d} \hat{w}_i x_{iJ} + \hat{b} + y_J \Big(\sum_{i=1}^{d} \underbrace{x_{iJ} x_{iJ}}_{\text{positive}} + 1 \Big)$$

applies positive or negative correction to current function to enforce agreement with actual class label y_J

Perceptron learning

- Run the tutorial perceptron_demo.m
- Try separating different digit classes…

Perceptron learning

- Extra: So far we have hand crafted features
- More modern versions: features are learned!

Perceptron learning

- Extra: So far we have hand crafted features
- More modern versions: features are learned!

Consider the following generalizations of the function in the perceptron

1. $$f(\mathbf{x}_j) = h\left(\sum_{i=1}^{d} w_i x_{ij} + b\right), \tag{6}$$

   where h is a nonlinear function. In this case the values of the outputs of f do not need to be restricted to -1 and $+1$ as with sign function.

2. To compare the output of $f(\mathbf{x}_i)$ with the actual label values y_j, we define a loss function:

   $$\mathcal{L}(f(\mathbf{x}_j), y_j) \tag{7}$$

The loss function can be used to guide the learning since it penalizes errors made by f.

Perceptron learning

- Extra: So far we have hand crafted features
- More modern versions: features are learned!

Common examples of loss functions are:

- Zero-One loss

$$\begin{cases} 1 & \text{if } \mathrm{sign}[f(\mathbf{x}_j)] \neq y_j, \\ 0 & \text{if } \mathrm{sign}[f(\mathbf{x}_j)] = y_j. \end{cases} \tag{8}$$

- Squared loss

$$(y_j - f(\mathbf{x}_j))^2 \,. \tag{9}$$

- Hinge loss

$$\max\{0,\ 1 - y_j f(\mathbf{x}_j)\}\,. \tag{10}$$

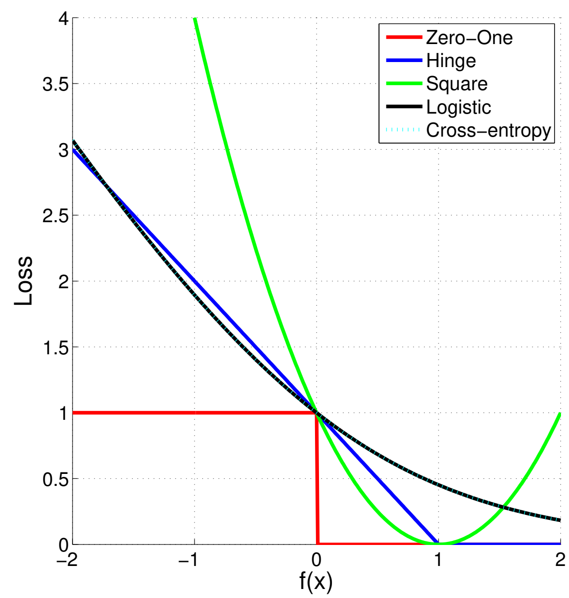
- Cross-entropy loss. In this case $f(\mathbf{x}_j)$ is transformed to a function $h(f(\mathbf{x}_j))$ with outputs in the range $[0, 1]$

$$-\frac{y_j + 1}{2}\log(f(\mathbf{x}_j)) - \frac{1 - y_j}{2}\log(1 - f(\mathbf{x}_j)). \tag{11}$$

Perceptron learning

- Extra: So far we have hand crafted features
- More modern versions: features are learned!

Values of the loss functions for $y_j = 1$ vs $f(\mathbf{x}_j)$



Perceptron learning

- Extra: So far we have hand crafted features
- More modern versions: features are learned!

Given a set $\{(\mathbf{x}_j, y_j)\}_{j=1}^{N}$ of feature-class pairs, learning can be accomplished by minimizing the average loss on this set. Having differentiable loss and activation functions provide a mathematically simple framework that allows the use gradient-based minimization techniques for learning. For instance, using the logistic function for h in (6) and the cross-entropy loss (11) yields an iterative procedure.

Perceptron learning

- Extra: So far we have hand crafted features
- More modern versions: features are learned!

Gradient descent for cross-entropy loss with logistic sigmoid activation.

1. Initialize $\{w_i\}_{i=1}^{d}$ and b ($\mathbf{w}$ and b can be initialized with zeros).
2. Let
$$f(\mathbf{x}_j) = \frac{1}{1+\exp(-z_i)}, \text{ where } z_j = \left(\sum_{i=1}^{d} w_i x_{ij} + b\right), \quad (12)$$
3. and
$$\Delta w_i = \frac{\partial}{\partial w_i}\mathcal{L}(f(\mathbf{x}_j), y_j) = \left(f(\mathbf{x}_j) - \frac{y_j+1}{2}\right) x_{ij}, \quad (13)$$
$$\Delta b = \frac{\partial}{\partial b}\mathcal{L}(f(\mathbf{x}_i), y_i) = \left(f(\mathbf{x}_j) - \frac{y_j+1}{2}\right). \quad (14)$$
4. Update parameters. Similarly to the perceptron
$$w_i \leftarrow w_i - \mu\Delta w_i \text{ and } b \leftarrow b - \mu\Delta b \quad (15)$$