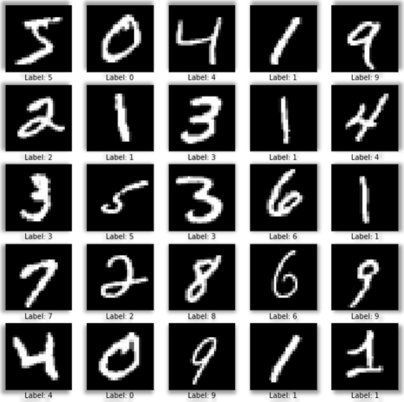
Perceptrons

Erhard Hinrichs

Seminar für Sprachwissenschaft
Eberhard-Karls Universität Tübingen

Classification Task for Handwritten Digits



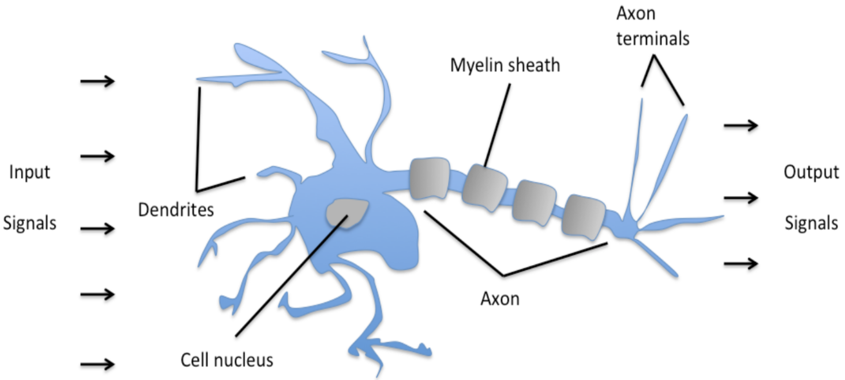
Representation of Handwritten Digits

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	3	18	18	18	126	136	175	26	166	255	247	127	0	0	0	0
6	0	0	0	0	0	0	0	0	30	36	94	154	170	253	253	253	253	253	225	172	253	242	195	64	0	0	0	0
7	0	0	0	0	0	0	0	49	238	253	253	253	253	253	253	253	253	251	93	82	82	56	39	0	0	0	0	0
8	0	0	0	0	0	0	0	18	219	253	253	253	253	253	198	182	247	241	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	80	156	107	253	253	205	11	0	43	154	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	14	1	154	253	90	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	139	253	190	2	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	11	190	253	70	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	35	241	225	160	108	1	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	81	240	253	253	119	25	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	45	186	253	253	150	27	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16	93	252	253	187	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	249	253	249	64	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	46	130	183	253	253	207	2	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	39	148	229	253	253	253	250	182	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	24	114	221	253	253	253	253	201	78	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	23	66	213	253	253	253	253	198	81	2	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	18	171	219	253	253	253	253	195	80	9	0	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	55	172	226	253	253	253	253	244	133	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	136	253	253	253	212	135	132	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

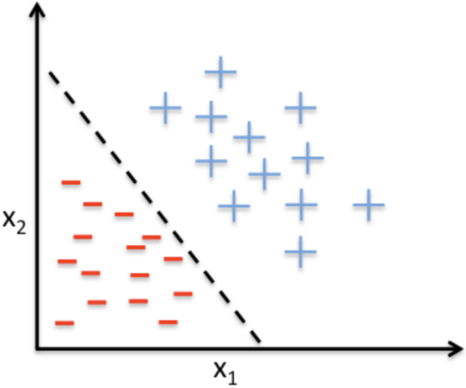
The MNIST Dataset

- MNIST short for: Modified National Institute of Standards and technology database
- A collection of handwritten digits (0-9) that are normalized in size and centered
- contains 60,000 training images and 10,000 testing images
- Representation of each digit as a 28x28 pixel grayscale image
- MNIST dataset is widely used for image recognition in machine vision
- a good use case for perceptron and regressions models in machine learning

Perceptron: Underlying Idea due to McCulloch-Pitts (1943)

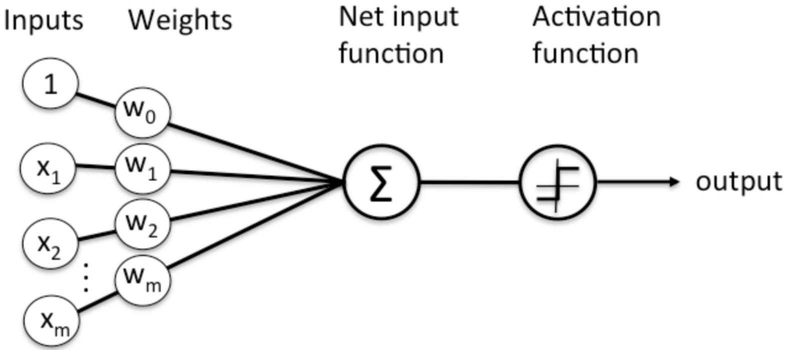


Binary Classification with Linear Decision Boundary



Example of a linear decision boundary for binary classification.

Perceptron Architecture



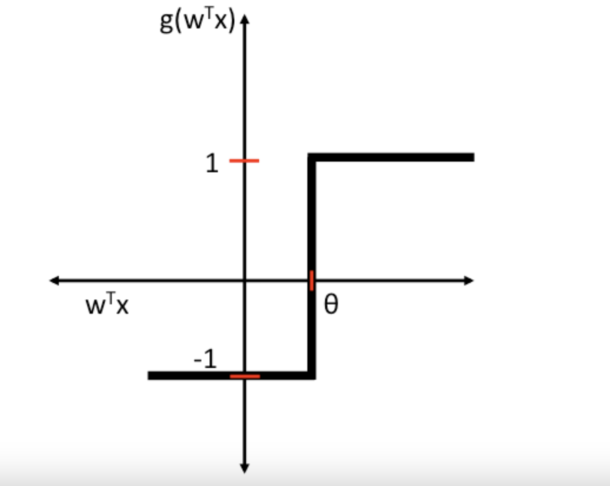
Schematic of Rosenblatt's perceptron.

Net Input Function and Activation Function

$$\mathbf{z} = \mathbf{w} \cdot \mathbf{x} + b \tag{1}$$

$$g(\mathbf{z}) = \begin{cases} 1 & \texttt{if z} \geq \theta \\ -1 & \texttt{otherwise} \end{cases} \tag{2}$$

Unit Step Function



Rosenblatt's Perceptron Rule

1. Initialize the weights w to 0 or small random numbers
2. For each training sample $x^{(i)}$:
   2.1 calculate the output value: $g(\mathbf{z})$
   2.2 update the weights: $w_j = w_j + \Delta w_j$

   where $\Delta w_j = \alpha(\texttt{targetlabel}^i - \texttt{predictedlabel}^i)x_j^{(i)}$
   and α is the learning rate.

Perceptron Update Rule

For each training sample x_i:

$$w := w + \Delta\, w$$

$$\Delta\, w := \alpha * (y_i - \hat{y}_i) * x_i$$

Vectors and Matrices

- Vectors as a special case of matrices:
  - matrices have m by n "shape": m rows and n columns.
  - vectors have m by 1 "shape": m rows and exactly one column.
  - vectors can be transposed into a 1 by m column matrix by the transpose operation.
    - Notation: $\mathbf{v}^T$
    - Example:
- Two types of multiplication
  - for vectors: dot product (aka: scalar product)
  - for matrices: matrix multiplication
- You will encounter both types of multiplication in feature weighting

Transpose of a Matrix

Given a matrix $A \in \mathbb{R}^{m \times n}$:

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}$$

the transpose of A, denoted as $A^T \in \mathbb{R}^{n \times m}$:

$$A^T = \begin{bmatrix} a_{11} & a_{21} & \dots & a_{m1} \\ a_{12} & a_{22} & \dots & a_{m2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1n} & a_{2n} & \dots & a_{mn} \end{bmatrix}$$

Input and Output Representations

- Input: A widely used formula for input specification of a training or test instance : $\mathbf{wx} + b$, or written as a function: $f(w, b) = \mathbf{w}^T\mathbf{x} + b$

  where the bias b is a scalar and where $\mathbf{w}^T$ is transpose of vector $\mathbf{w}$ of weights and $\mathbf{x}$ is a vector of feature values and where $\mathbf{w}$ and $\mathbf{x}$ have the same number of dimensions.

- Output: a scalar y or $\hat{y}$

  representing the predicted ($\hat{y}$) or gold class (y) membership of a training or test instance x^i

Matrix multiplication

Given:

$$u = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

$$v = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}$$

$$u \times v = \begin{bmatrix} 1 * 5 + 2 * 7 & 1 * 6 + 2 * 8 \\ 3 * 5 + 4 * 7 & 3 * 6 + 4 * 8 \end{bmatrix}$$

Main Keywords

- Supervised learning algorithm
- Input and output representations
  - Features, weights, and bias
  - Class assignment: binary or multi-class
- Training phase (also known as: learning phase) and test phase
- Weight and bias optimization
- Loss function (also known as: objective function, cost function)
- Decision boundary
- Activation function with threshold