Language Technologies Institute

Carnegie Mellon University

Multimodal Machine Learning

Lecture 2.1: Basic Concepts – Neural Networks

Louis-Philippe Morency

* Original version co-developed with Tadas Baltrusaitis

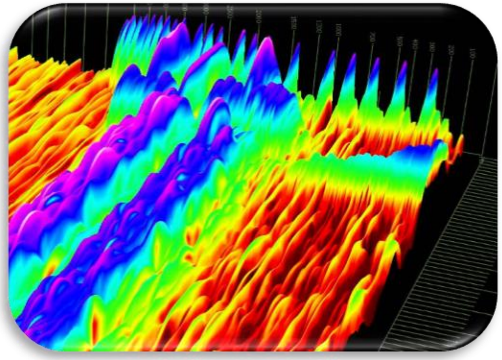
Lecture Objectives

- Unimodal basic representations
  - Visual, language and acoustic modalities
- Data-driven machine learning
  - Training, validation and testing
  - Example: K-nearest neighbor
- Linear Classification
  - Score function
  - Two loss functions (cross-entropy and hinge loss)
- Neural networks
- Course project team formation

Multimodal Machine Learning

Verbal

Vocal



Visual

Core Technical Challenges:

Representation   Translation   Alignment

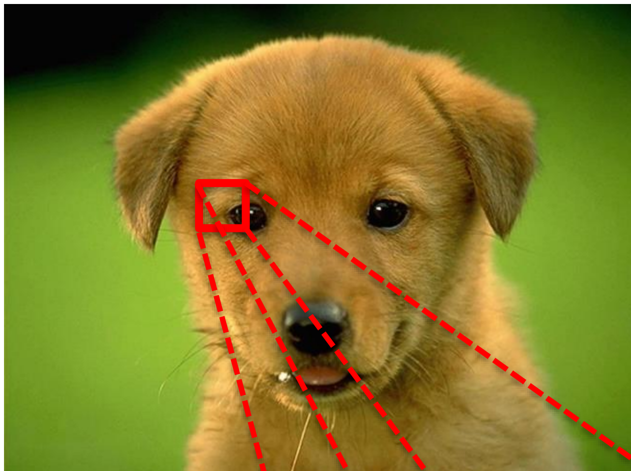
Fusion   Co-Learning

These challenges are non-exclusive.

Unimodal Basic Representations

Unimodal Classification – Visual Modality

Color image



Each pixel is represented in $\mathcal{R}^d$, d is the number of colors (d=3 for RGB)

88	82	84	88	85	83	80	93	102
88	80	78	80	80	78	73	94	100
85	79	80	78	77	74	65	91	99
38	35	40	35	39	74	77	70	65
20	25	23	28	37	69	64	60	57
22	26	22	28	40	65	64	59	34
24	28	24	30	37	60	58	56	66
21	22	23	27	38	60	67	65	67
23	22	22	25	38	59	64	67	66

Input observation x_i

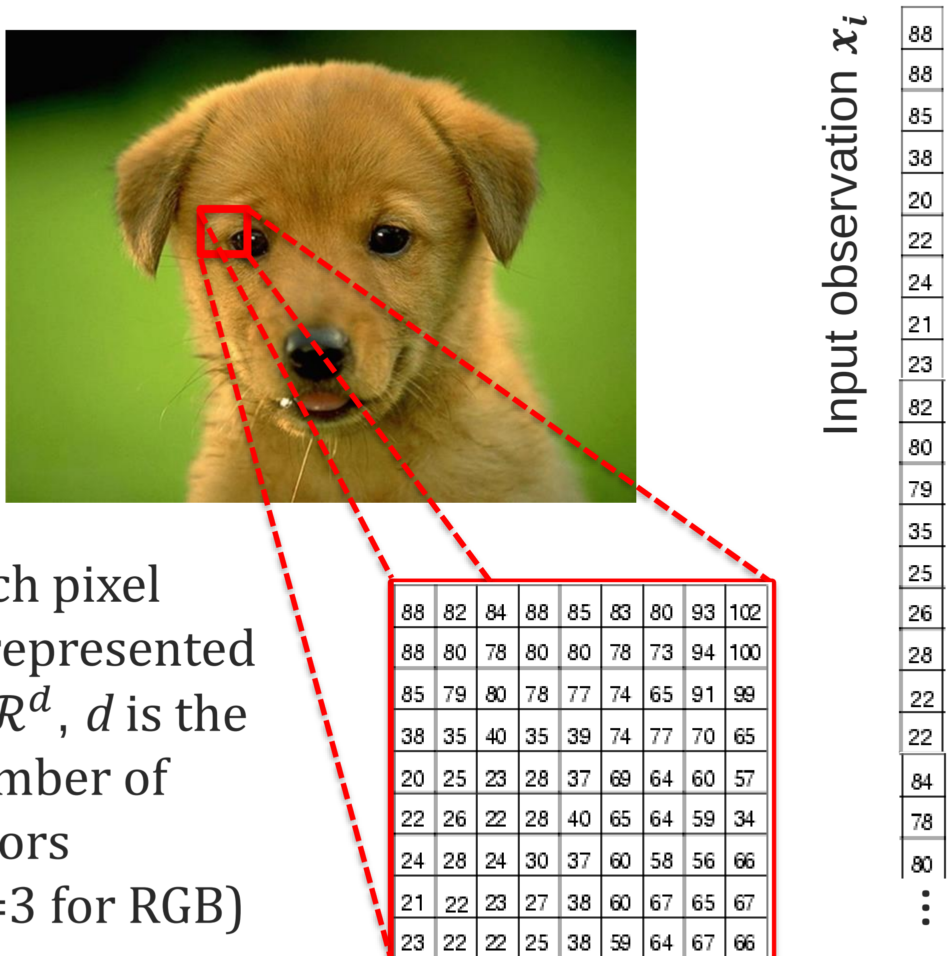
88
88
85
38
20
22
24
21
23
82
80
79
35
25
26
28
22
22
84
78
80
⋮

Binary classification problem

Dog ?

label $y_i \in \mathcal{Y} = \{0,1\}$

Unimodal Classification – Visual Modality



88	82	84	88	85	83	80	93	102
88	80	78	80	80	78	73	94	100
85	79	80	78	77	74	65	91	99
38	35	40	35	39	74	77	70	65
20	25	23	28	37	69	64	60	57
22	26	22	28	40	65	64	59	34
24	28	24	30	37	60	58	56	66
21	22	23	27	38	60	67	65	67
23	22	22	25	38	59	64	67	66

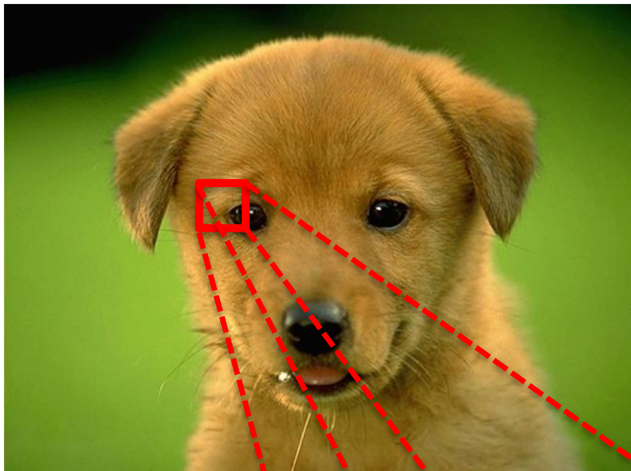
Each pixel is represented in $\mathcal{R}^d$, d is the number of colors (d=3 for RGB)

Multi-class classification problem

Duck
-or-
Cat
-or-
Dog
-or-
Pig
-or-
Bird ?

label $y_i \in \mathcal{Y} = \{0,1,2,3, \dots\}$

Unimodal Classification – Visual Modality



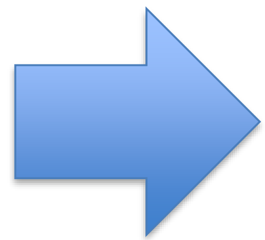
Each pixel is represented in $\mathcal{R}^d$, d is the number of colors (d=3 for RGB)

88	82	84	88	85	83	80	93	102
88	80	78	80	80	78	73	94	100
85	79	80	78	77	74	65	91	99
38	35	40	35	39	74	77	70	65
20	25	23	28	37	69	64	60	57
22	26	22	28	40	65	64	59	34
24	28	24	30	37	60	58	56	66
21	22	23	27	38	60	67	65	67
23	22	22	25	38	59	64	67	66

Input observation $\boldsymbol{x}_i$

88
88
85
38
20
22
24
21
23
82
80
79
35
25
26
28
22
22
84
78
80
⋮

Multi-label (or multi-task) classification problem



Duck?

Cat ?

Dog ?

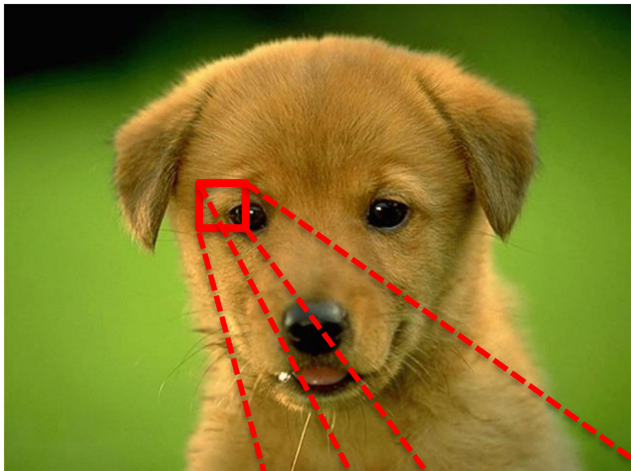
Pig ?

Bird ?

Puppy ?

label vector $\boldsymbol{y}_i \in \mathcal{Y}^m = \{0,1\}^m$

Unimodal Classification – Visual Modality



Each pixel is represented in $\mathbb{R}^d$, d is the number of colors (d=3 for RGB)

88	82	84	88	85	83	80	93	102
88	80	78	80	80	78	73	94	100
85	79	80	78	77	74	65	91	99
38	35	40	35	39	74	77	70	65
20	25	23	28	37	69	64	60	57
22	26	22	28	40	65	64	59	34
24	28	24	30	37	60	58	56	66
21	22	23	27	38	60	67	65	67
23	22	22	25	38	59	64	67	66

Input observation $\boldsymbol{x_i}$

88
88
85
38
20
22
24
21
23
82
80
79
35
25
26
28
22
22
84
78
80
⋮

Multi-label (or multi-task) regression problem

Age ?

Height ?

Weight ?

Distance ?

Happy ?

label vector $\boldsymbol{y_i} \in \mathcal{Y}^m = \mathbb{R}^m$

Unimodal Classification – Language Modality

Written language

★★★★★ **Masterful!**

By Antony Witheyman - January 12, 2006

Ideal for anyone with an interest in disguises who likes to see the subject tackled in a $\boxed{\text{humourous}}$ manner.

0 of 4 people found this review helpful

Spoken language

MARTHA(CON'T)
Look around you. Look at all the
great things you've done and the
people you've helped.

CLARK
But you've only put up the good
things they say about me.

MARTHA
Clark, honey. If I were to use the
bad things they say I could cover
the barn, the house and the
outhouse.

Input observation x_i

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \vdots \end{bmatrix}$$

"one-hot" vector

$|x_i|$ = number of words in dictionary

Word-level classification

Part-of-speech ?
(noun, verb,…)

Sentiment ?
(positive or negative)

Named entity ?
(names of person,…)

Unimodal Classification – Language Modality

Written language

★★★★★ **Masterful!**

By Antony Witheyman - January 12, 2006

Ideal for anyone with an interest in disguises who likes to see the subject tackled in a humourous manner.

0 of 4 people found this review helpful

Spoken language

```
                MARTHA(CON'T)
Look around you. Look at all the
great things you've done and the
people you've helped.

                CLARK
But you've only put up the good
things they say about me.

                MARTHA
Clark, honey. If I were to use the
bad things they say I could cover
the barn, the house and the
outhouse.
```

Input observation x_i

$$\begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ \vdots \end{bmatrix}$$

"bag-of-word" vector

$|x_i|$ = number of words in dictionary

Document-level classification

Sentiment ?

(positive or negative)

Unimodal Classification – Language Modality

Written language

★★★★★ **Masterful!**

By Antony Witheyman - January 12, 2006

Ideal for anyone with an interest in disguises who likes to see the subject tackled in a humourous manner.

0 of 4 people found this review helpful

Spoken language

```
              MARTHA(CON'T)
Look around you. Look at all the
great things you've done and the
people you've helped.

              CLARK
But you've only put up the good
things they say about me.

              MARTHA
Clark, honey. If I were to use the
bad things they say I could cover
the barn, the house and the
outhouse.
```

Input observation x_i

0
1
0
0
1
0
1
0
0
0
0
1
0
0
0
0
1
0
0
0
⋮

"bag-of-word" vector

$|x_i|$ = number of words in dictionary

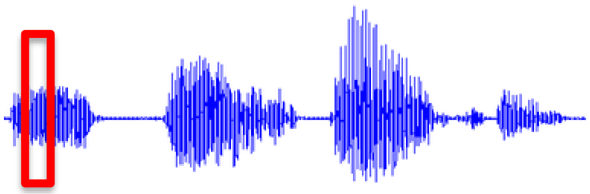
Utterance-level classification

Sentiment ?

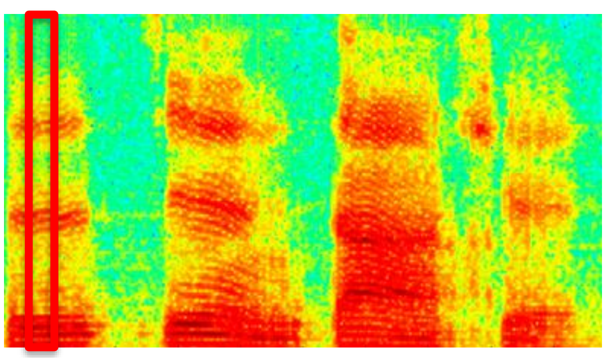
(positive or negative)

Unimodal Classification – Acoustic Modality

Digitalized acoustic signal



- Sampling rates: 8~96kHz
- Bit depth: 8, 16 or 24 bits
- Time window size: 20ms
  - Offset: 10ms



Spectogram

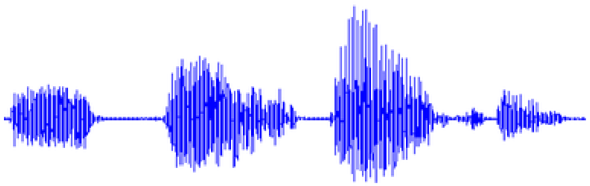
Input observation x_i

0.21
0.14
0.56
0.45
0.9
0.98
0.75
0.34
0.24
0.11
0.02

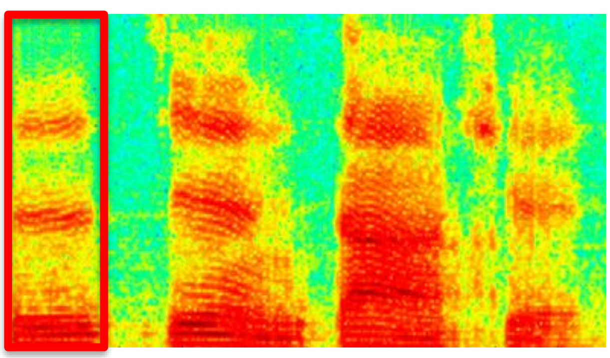
Spoken word ?

Unimodal Classification – Acoustic Modality

Digitalized acoustic signal



- Sampling rates: 8~96kHz
- Bit depth: 8, 16 or 24 bits
- Time window size: 20ms
  - Offset: 10ms



Spectogram

Input observation x_i

0.21
0.14
0.56
0.45
0.9
0.98
0.75
0.34
0.24
0.11
0.02
0.24
0.26
0.58
0.9
0.99
0.79
0.45
0.34
0.24
⋮

Emotion ?

Spoken word ?

Voice quality ?

Data-Driven Machine Learning

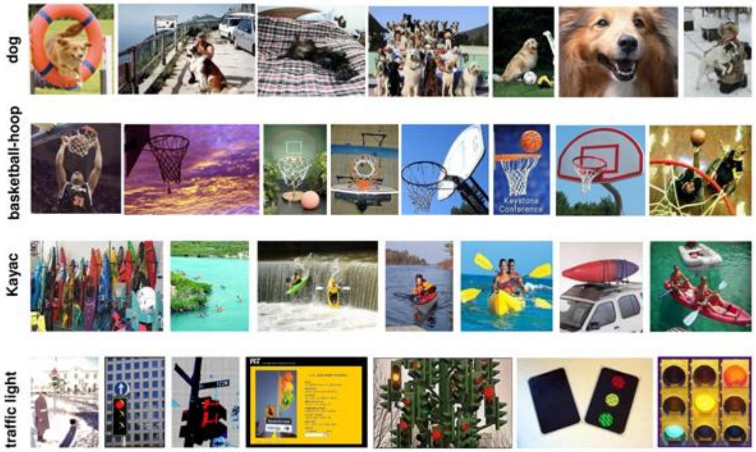
Data-Driven Machine Learning

1. **Dataset:** Collection of labeled samples $\mathrm{D}: \{x_i, y_i\}$
2. **Training:** Learn classifier on training set
3. **Testing:** Evaluate classifier on hold-out test set

Dataset

Simple Classifier ?

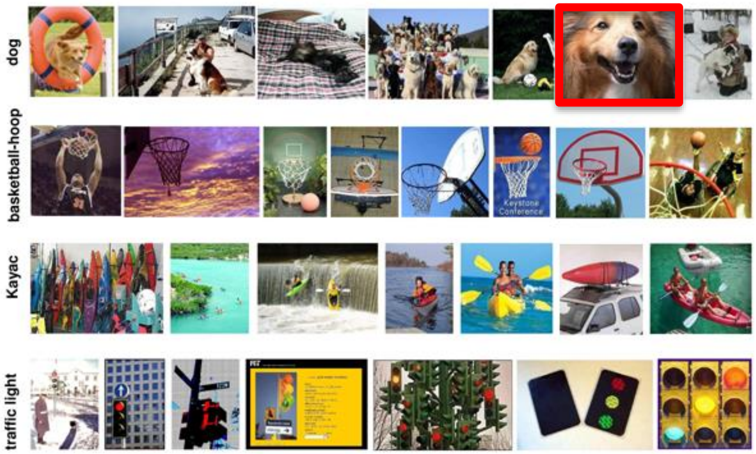
Traffic light
-or-
Dog
-or-
Basket
-or-
Kayak ?



Dataset

Simple Classifier: Nearest Neighbor

Traffic light
-or-
Dog
-or-
Basket
-or-
Kayak ?



Training

Nearest Neighbor Classifier

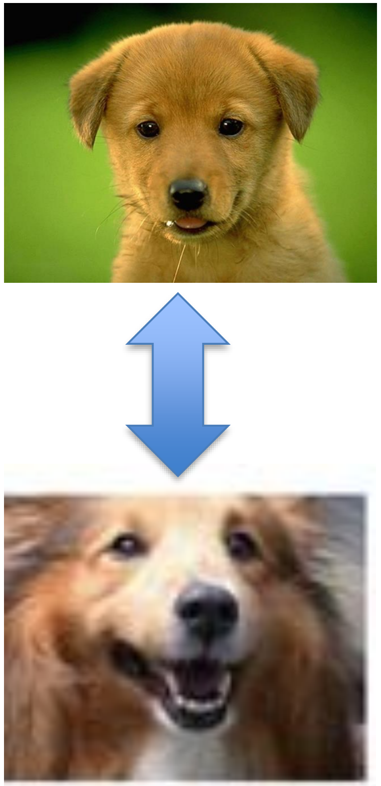
- Non-parametric approaches—key ideas:
  - *“Let the data speak for themselves”*
  - *“Predict new cases based on similar cases”*
  - *“Use multiple local models instead of a single global model”*

- What is the complexity of the NN classifier w.r.t training set of N images and test set of M images?
  - at training time?

    O(1)
  - At test time?

    O(N)

Simple Classifier: Nearest Neighbor



Distance metrics

L1 (Manhattan) distance:

$$d_1(x_1, x_2) = \sum_j \left| x_1^j - x_2^j \right|$$

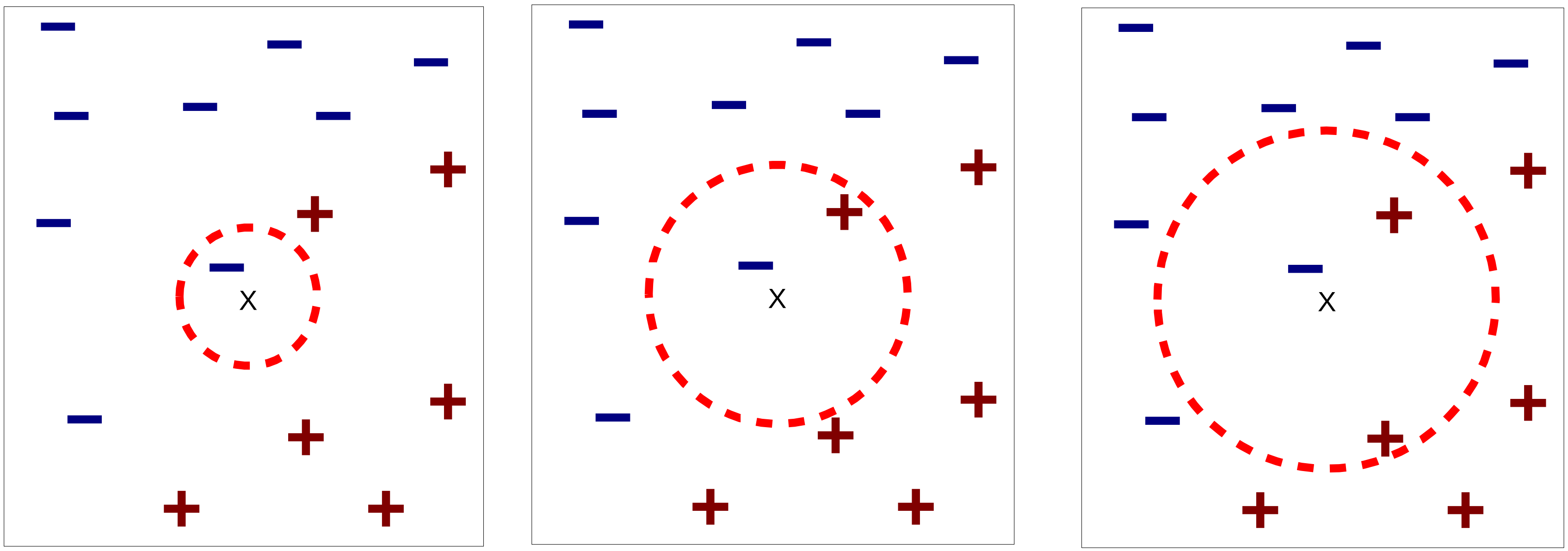
L2 (Eucledian) distance:

$$d_2(x_1, x_2) = \sqrt{\sum_j \left(x_1^j - x_2^j \right)^2}$$

Which distance metric to use?

First hyper-parameter!

Definition of K-Nearest Neighbor



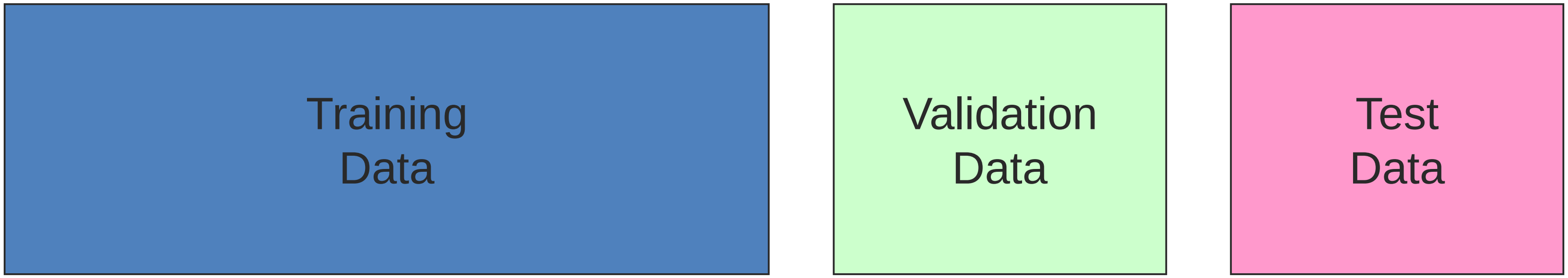
(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

What value should we set K?

Second hyper-parameter!

Data-Driven Approach

1. **Dataset:** Collection of labeled samples $\mathrm{D}: \{x_i, y_i\}$
2. **Training:** Learn classifier on training set
3. **Validation:** Select optimal hyper-parameters
4. **Testing:** Evaluate classifier on hold-out test set



Evaluation methods (for validation and testing)

- Holdout set: The available data set D is divided into two disjoint subsets,
  - the *training set* D_{train} (for learning a model)
  - the *test set* D_{test} (for testing the model)
- Important: training set should not be used in testing and the test set should not be used in learning.
  - Unseen test set provides a unbiased estimate of accuracy.
- The test set is also called the holdout set. (the examples in the original data set D are all labeled with classes.)
- This method is mainly used when the data set D is large.
- Holdout methods can also be used for validation

Evaluation methods (for validation and testing)

- n-fold cross-validation: The available data is partitioned into *n* equal-size disjoint subsets.
- Use each subset as the test set and combine the rest *n*-1 subsets as the training set to learn a classifier.
- The procedure is run *n* times, which give *n* accuracies.
- The final estimated accuracy of learning is the average of the *n* accuracies.
- 10-fold and 5-fold cross-validations are commonly used.
- This method is used when the available data is not large.

Evaluation methods (for validation and testing)

- Leave-one-out cross-validation: This method is used when the data set is very small.
- Each fold of the cross validation has only a single test example and all the rest of the data is used in training.
- If the original data has *m* examples, this is *m*-fold cross-validation
- It is a special case of cross-validation

Linear Classification: Scores and Loss

Linear Classification (e.g., neural network)

Image

(Size: 32*32*3)

?

1. **Define a (linear) score function**
2. **Define the loss function (possibly nonlinear)**
3. **Optimization**

1) Score Function

Image

(Size: 32*32*3)

Duck ?

Cat ?

Dog ?

Pig ?

Bird ?

What should be the prediction score for each label class?

For linear classifier:

Input observation *(ith element of the dataset)* [3072x1]

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

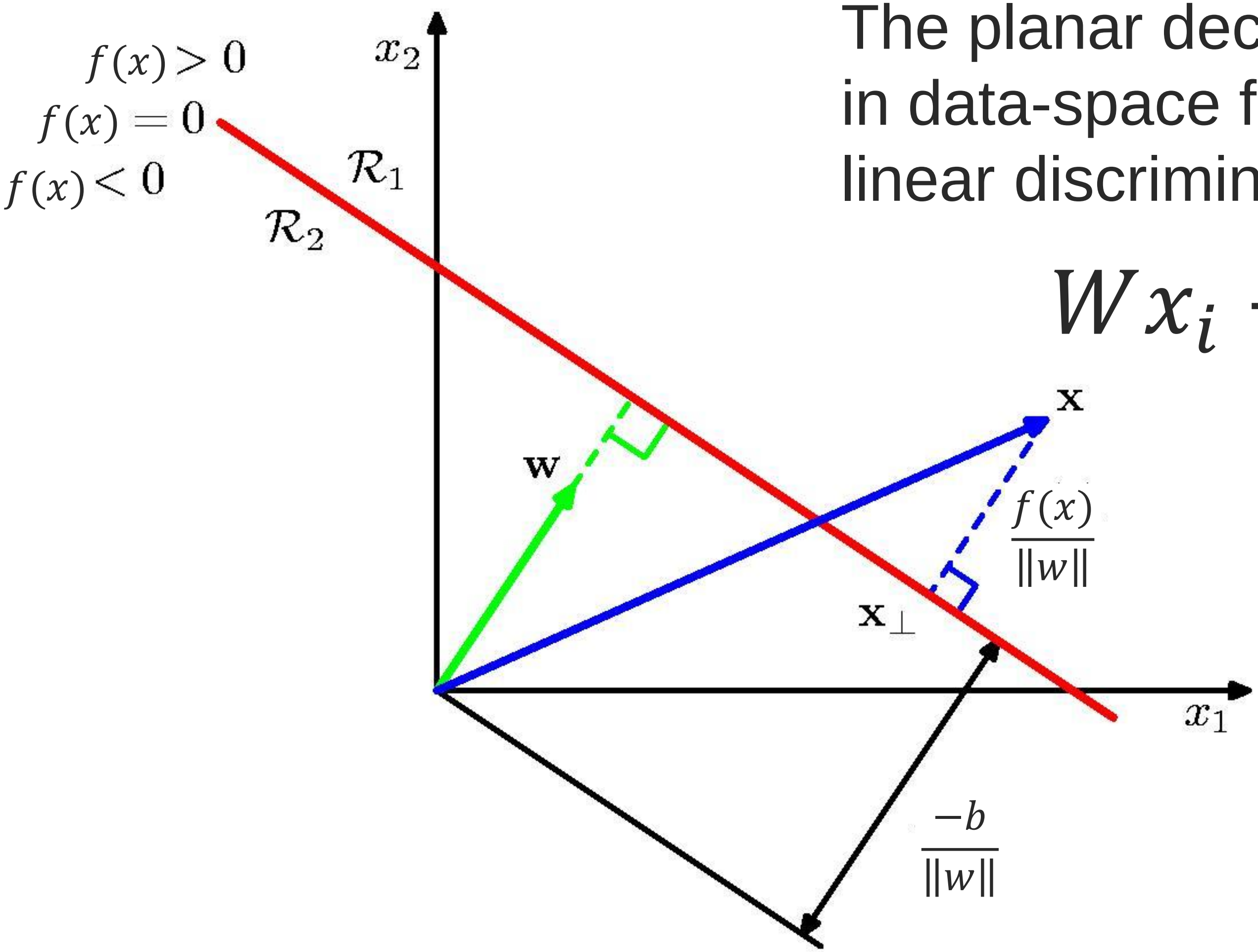
Weights [10x3072]

Bias vector [10x1]

Parameters [10x3073]

Class score [10x1]

Interpreting a Linear Classifier



The planar decision surface in data-space for the simple linear discriminant function:

$$W x_i + b > 0$$

Some Notation Tricks – Multi-Label Classification

$$W = [W_1 \quad W_2 \quad \dots \quad W_N]$$

$$f(x_i; W, b) = Wx_i + b \quad \longrightarrow \quad f(x_i; W) = Wx_i$$

Weights x Input + Bias

[10x3072] [3072x1] [10x1]

Weights x Input

[10x3073] [3073x1]

The bias vector will be the last column of the weight matrix

Add a “1” at the end of the input observation vector

Some Notation Tricks

General formulation of linear classifier: $f(x_i; W, b)$

"dog" linear classifier:

$$f\left(x_i; W_{dog}, b_{dog}\right) \quad \text{or}$$

$$f(x_i; W, b)_{dog} \quad \text{or} \quad f_{dog}$$

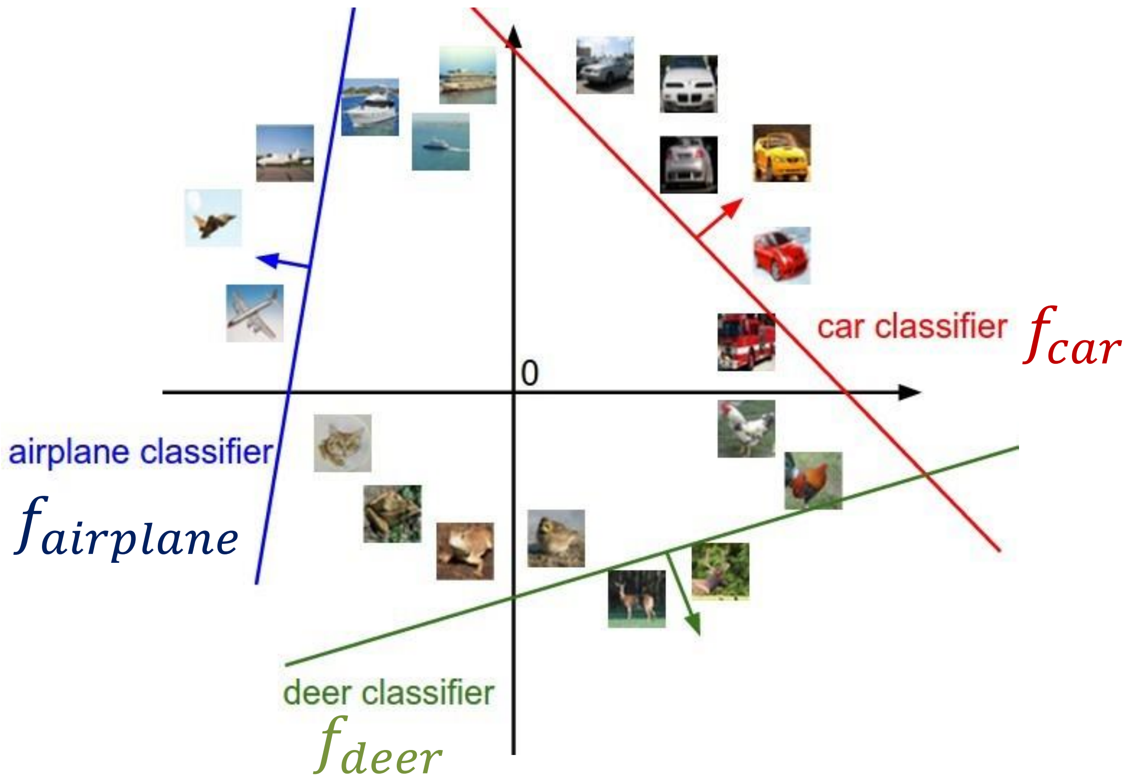
Linear classifier for label *j*:

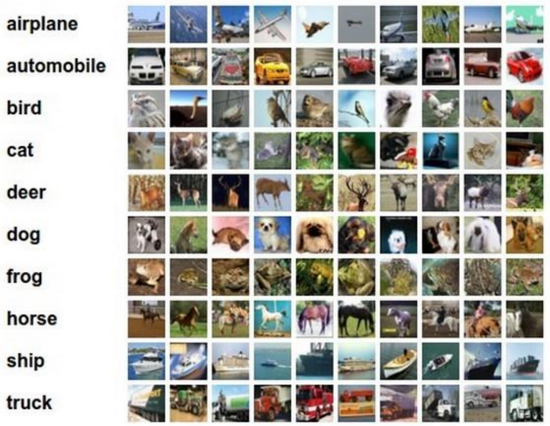
$$f\left(x_i; W_j, b_j\right) \quad \text{or}$$

$$f(x_i; W, b)_j \quad \text{or} \quad f_j$$

Interpreting Multiple Linear Classifiers

$$f(x_i; W_j, b_j) = W_j x_i + b_j$$





CIFAR-10 object recognition dataset

Linear Classification: 2) Loss Function

(or cost function or objective)

Scores $f(x_i; W)$ **Label** $y_i = 2\ (dog)$ → **Loss** $L_i = ?$

Image x_i

(Size: 32*32*3)

0 (duck) ?	-12.3
1 (cat) ?	45.6
2 (dog) ?	98.7
3 (pig) ?	12.2
4 (bird) ?	-45.3

Multi-class problem

How to assign only one number representing how “unhappy” we are about these scores?

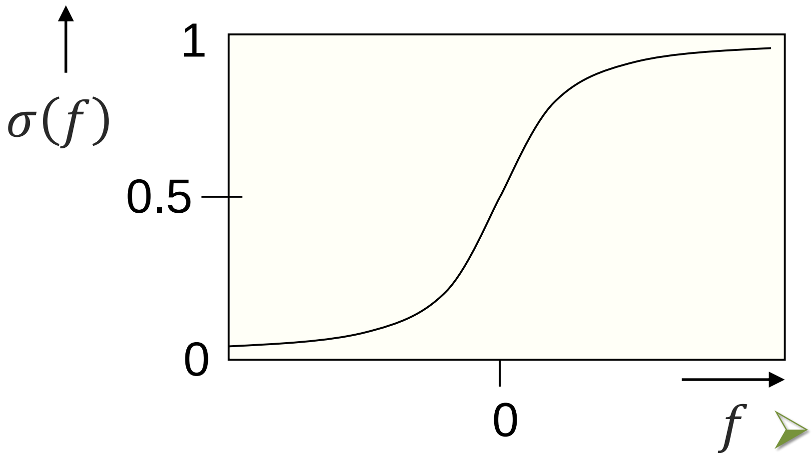
The loss function quantifies the amount by which the prediction scores deviate from the actual values.

A first challenge: how to normalize the scores?

First Loss Function: Cross-Entropy Loss

(or logistic loss)

Logistic function: $$\sigma(f) = \frac{1}{1 + e^{-f}}$$



➢ **Score function**

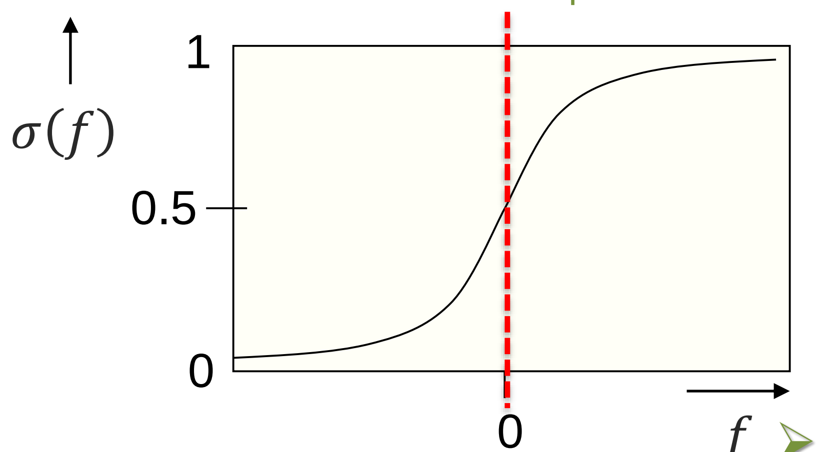
First Loss Function: Cross-Entropy Loss

(or logistic loss)

Logistic function: $$\sigma(f) = \frac{1}{1+e^{-f}}$$

Logistic regression: (two classes) $$p(y_i = \text{"}dog\text{"} | x_i; w) \quad = \sigma(w^T x_i)$$

= true for two-class problem



➢ **Score function**

First Loss Function: Cross-Entropy Loss

(or logistic loss)

Logistic function: $$\sigma(f) = \frac{1}{1 + e^{-f}}$$

Logistic regression:
(two classes)

$$p(y_i = "dog" | x_i; w) \quad = \sigma(w^T x_i)$$

= true
for two-class problem

Softmax function:
(multiple classes)

$$p(y_i | x_i; W) = \frac{e^{f_{y_i}}}{\sum_j e^{f_j}}$$

First Loss Function: Cross-Entropy Loss

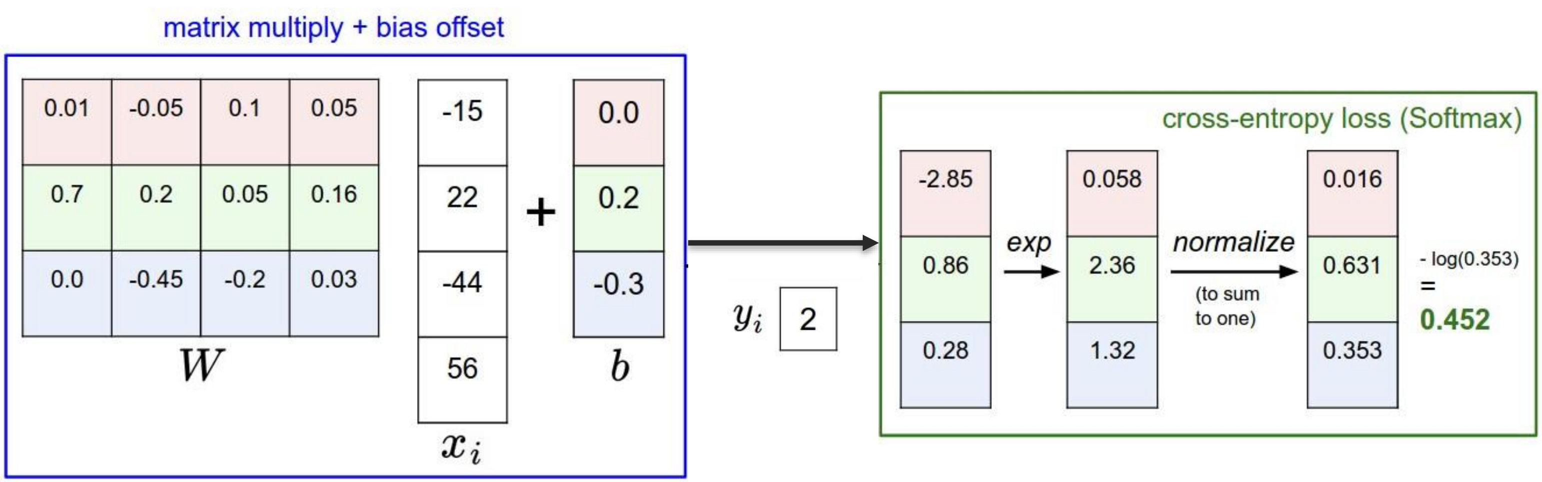
(or logistic loss)

Cross-entropy loss:

$$L_i = -\log\left(\frac{e^{f_{y_i}}}{\sum_j e^{f_j}}\right)$$

Softmax function

Minimizing the negative log likelihood.



Second Loss Function: Hinge Loss

(or max-margin loss or Multi-class SVM loss)

$$\boxed{L_i} = \boxed{\sum_{j \neq y_i}} \max(0, \boxed{f(x_i, W)_j - f(x_i, W)_{y_i}} + \Delta)$$

loss due to example i

sum over all incorrect labels

difference between the correct class score and incorrect class score

Second Loss Function: Hinge Loss

(or max-margin loss or Multi-class SVM loss)

$$L_i = \sum_{j \neq y_i} \max(0, f(x_i, W)_j - f(x_i, W)_{y_i} + \Delta)$$

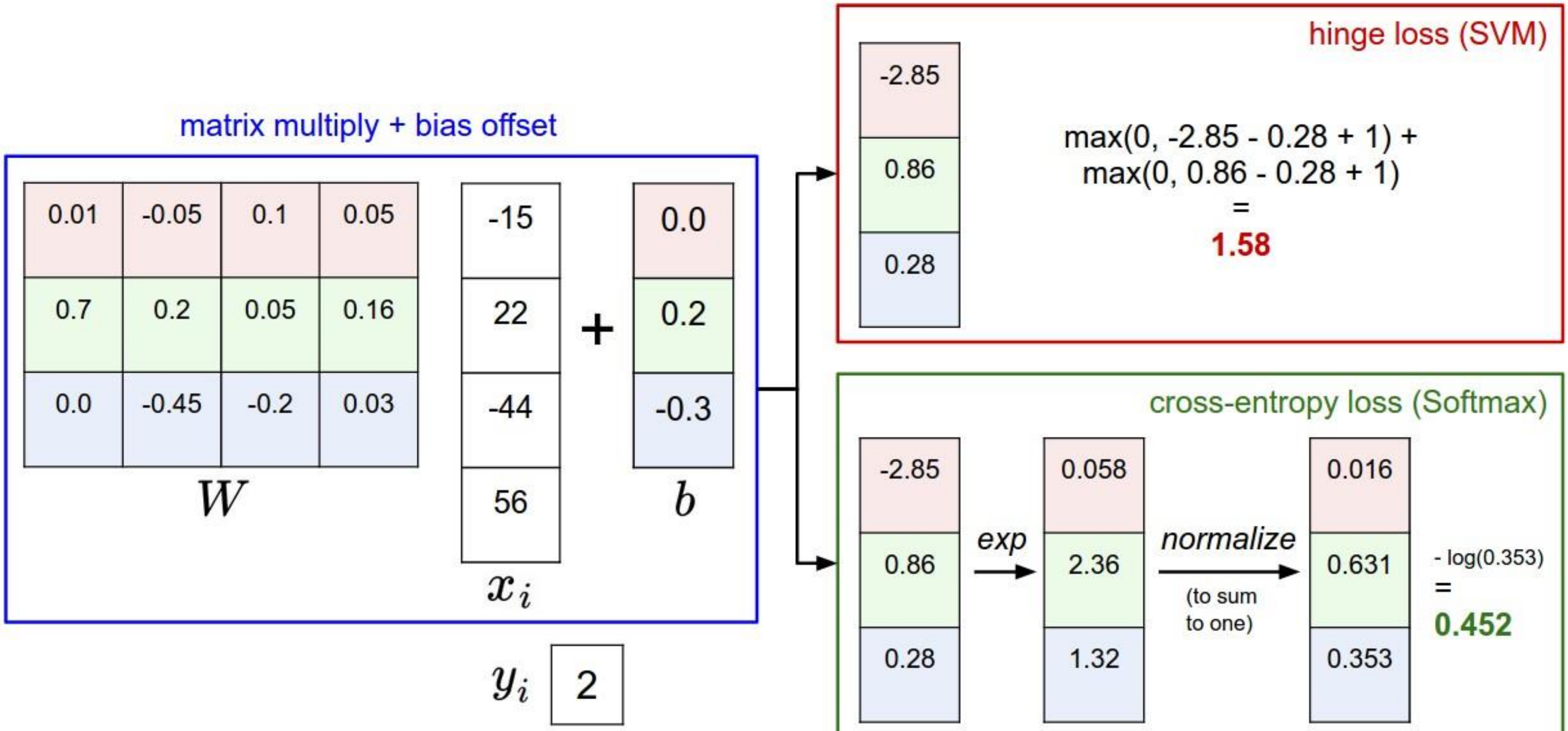
e.g. 10

Example: $f(x_i, W) = [13, -7, 11]$

$y_i = 0$

$$L_i = \max(0, -7 - 13 + 10) + \max(0, 11 - 13 + 10)$$

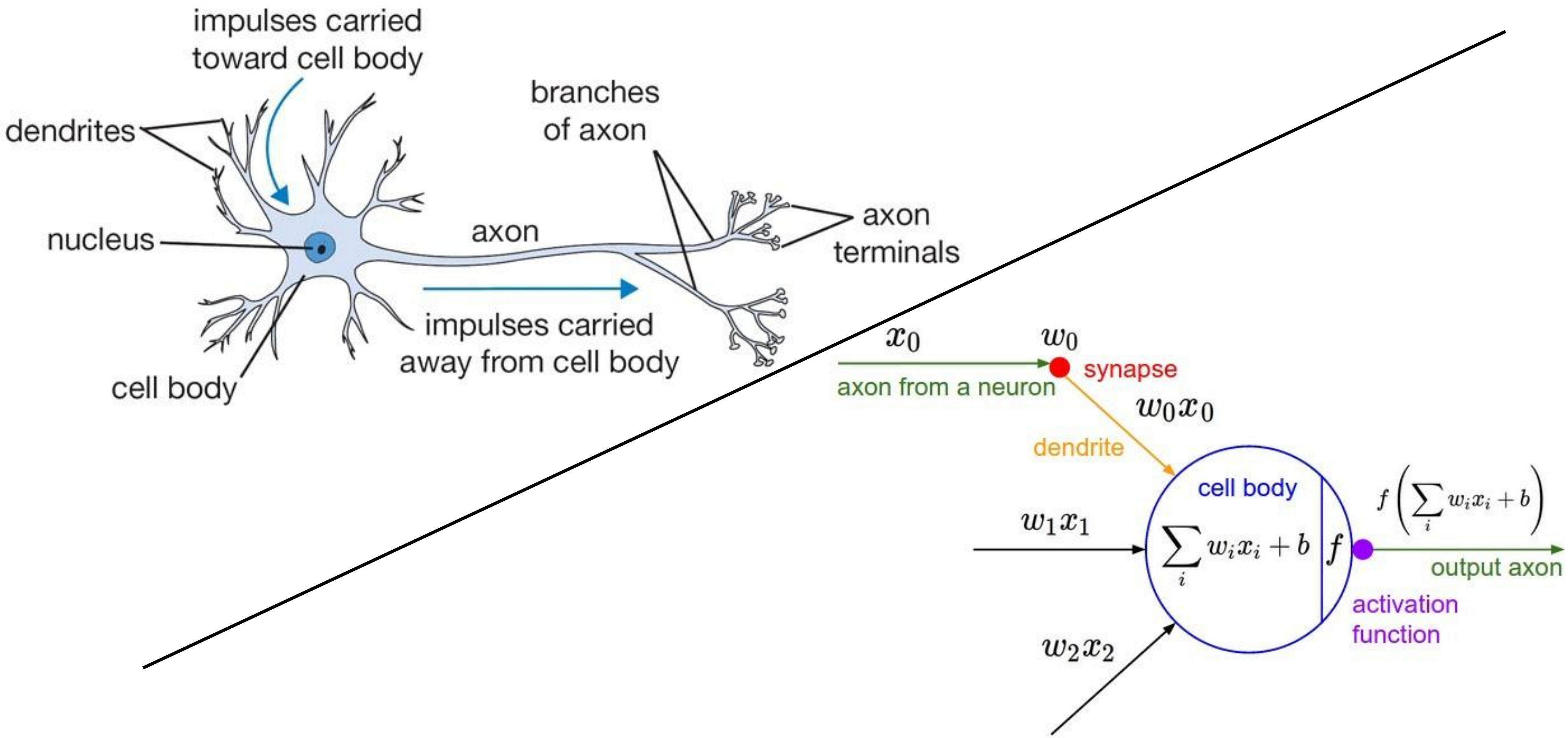
Two Loss Functions



Basic Concepts: Neural Networks

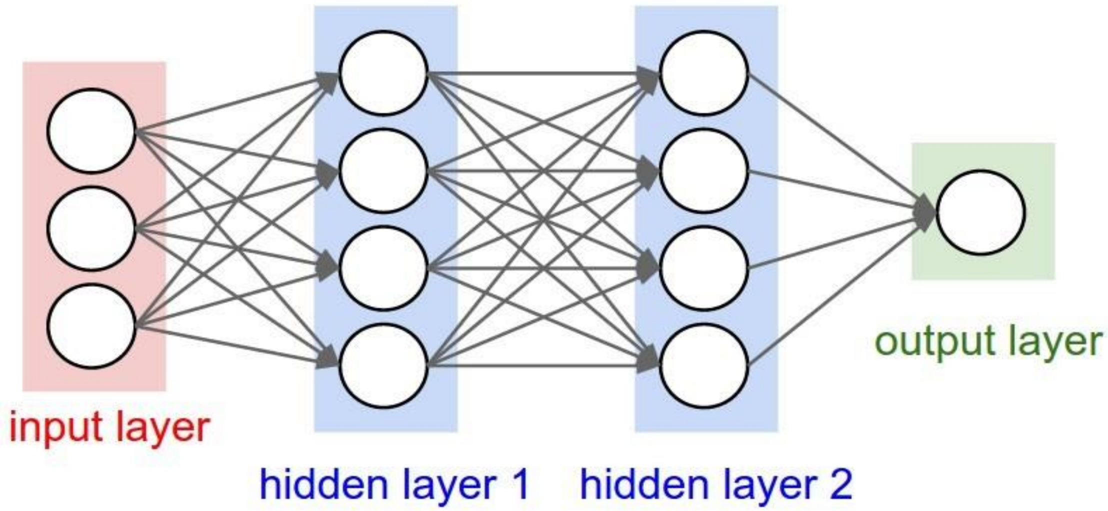
Neural Networks – inspiration

- Made up of artificial neurons



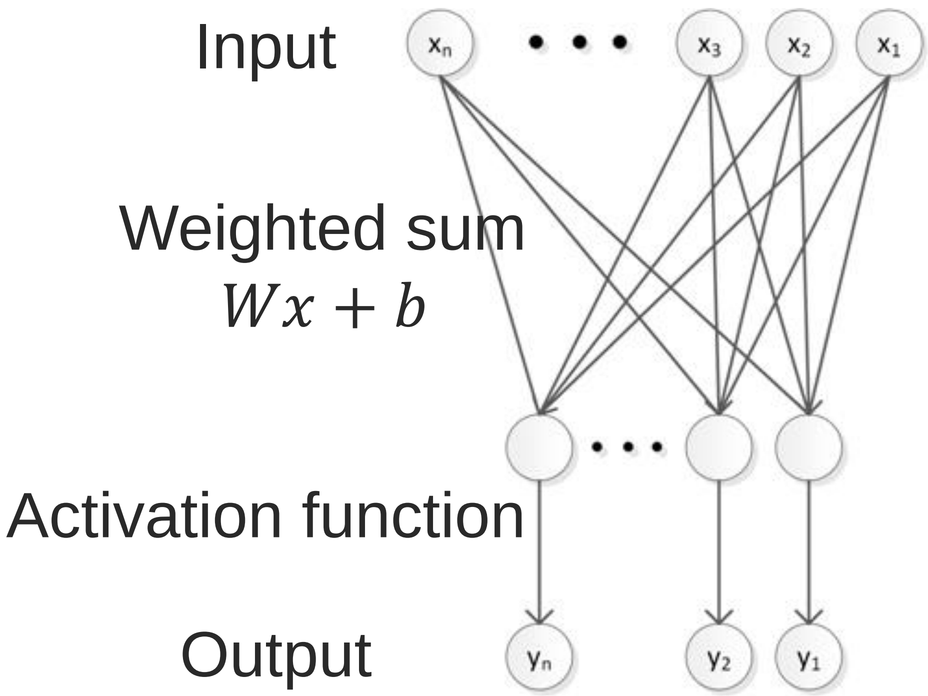
Neural Networks – score function

- Made up of artificial neurons
  - Linear function (dot product) followed by a nonlinear activation function
- Example a Multi Layer Perceptron



Basic NN building block

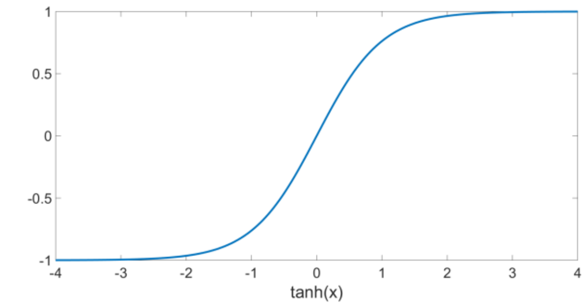
- Weighted sum followed by an activation function

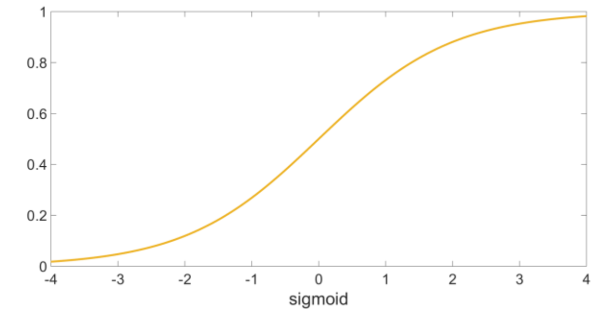


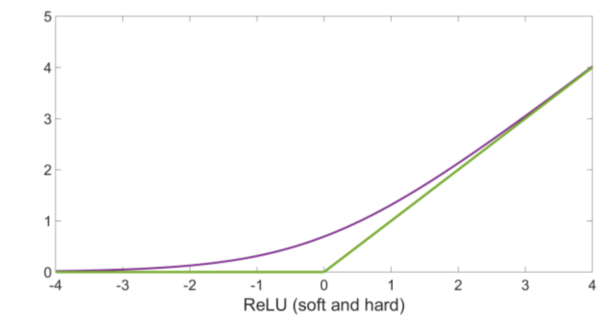
$$y = f(Wx + b)$$

Neural Networks – activation function

- $f(x) = \tanh(x)$
- Sigmoid - $f(x) = (1 + e^{-x})^{-1}$
- Linear – $f(x) = ax + b$
- ReLU $f(x) = \max(0, x) \sim \log(1 + \exp(x))$
  - Rectifier Linear Units
  - Faster training - no gradient vanishing
  - Induces sparsity







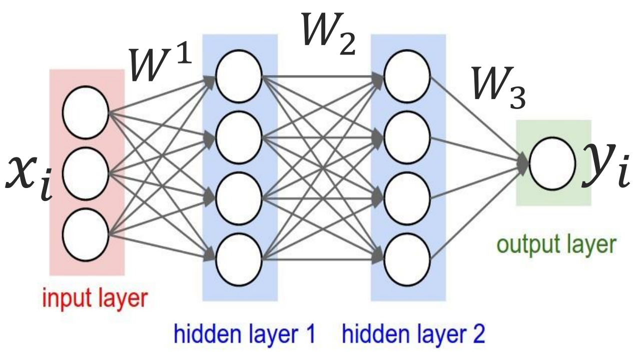
Multi-Layer Feedforward Network

Activation functions (individual layers)

$$f_{1;W_1}(x) = \sigma(W_1 x + b_1)$$

$$f_{2;W_2}(x) = \sigma(W_2 x + b_2)$$

$$f_{3;W_3}(x) = \sigma(W_3 x + b_3)$$



Score function

$$y_i = f(x_i) = f_{3;W_3}(f_{2;W_2}(f_{1;W_1}(x_i)))$$

Loss function (e.g., Euclidean loss)

$$L_i = (f(x_i) - y_i)^2 = (f_{3;W_3}(f_{2;W_2}(f_{1;W_1}(x_i))))^2$$

Neural Networks inference and learning

- Inference (Testing)
  - Use the score function ($y = f(\boldsymbol{x}; W)$)
  - Have a trained model (parameters W)
- Learning model parameters (Training)
  - Loss function (L)
  - Gradient
  - Optimization