Lecture 2 Linear Classification

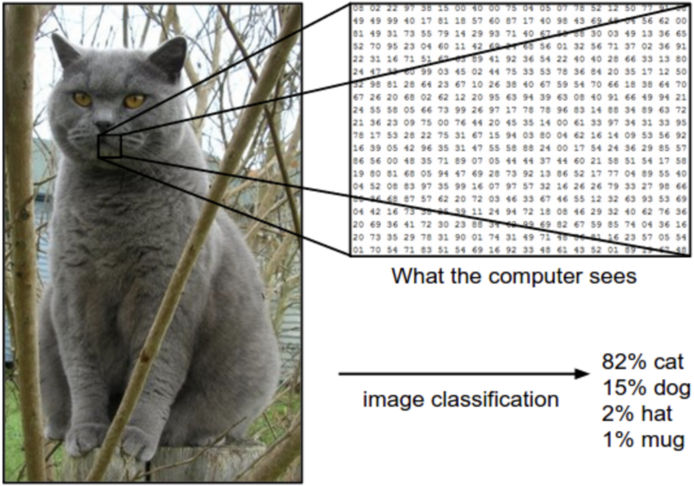
Fei Tan

Department of Economics
Chaifetz School of Business
Saint Louis University

E6930 Introduction to Artificial Neural Networks

June 16, 2024

Image Classification



The Road Ahead...

- Score function: parameterization, interpretation
- Loss function: data loss, regularization/prior
- SVM vs. Softmax classifiers

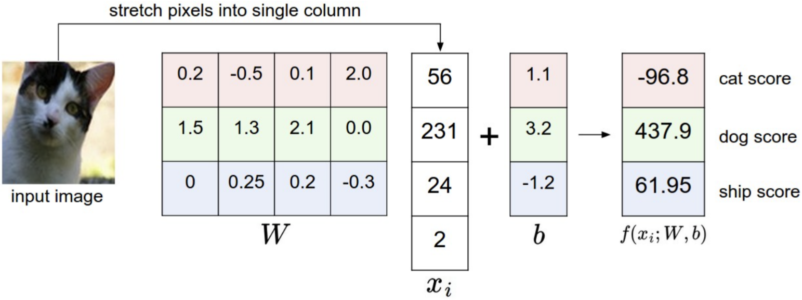
Score Function

Linear score

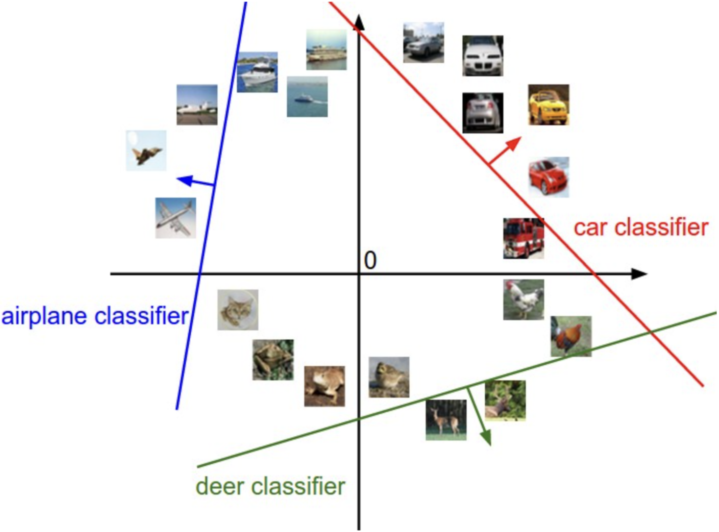
$$f(x_i, W, b) = Wx_i + b$$

- Notation
  - dataset of N examples $\{(x_i, y_i)\}_{i=1}^{N}$, D (normalized) features $x_i \in \mathbb{R}^D$, K categories $y_i \in 1, \ldots, K$
  - $K \times D$ weights W, $K \times 1$ bias b
- Pipeline
  - split data into training, validation, and test sets
  - train parameters and (cross) validate hyperparameters
  - evaluate on test test only once at the end

Algebraic Interpretation



Geometric Interpretation



Loss Function

Regularized loss

$$L = \frac{1}{N}\sum_i L_i + \lambda R(W), \quad \lambda > 0$$

- Examples of data loss
  - multiclass support vector machine (SVM) classifier uses hinge loss: $L_i = \sum_{j \neq y_i} \max(0, f_j - f_{y_i} + \Delta)$, $\Delta > 0$
  - Softmax classifier uses cross-entropy loss: $L_i = -\log\left(\frac{e^{f_{y_i}}}{\sum_j e^{f_j}}\right)$
- Examples of regularization/prior
  - L^1 regularization: $R(W) = \sum_{i,j} |W_{ij}|$
  - L^2 regularization: $R(W) = \sum_{i,j} W_{ij}^2$

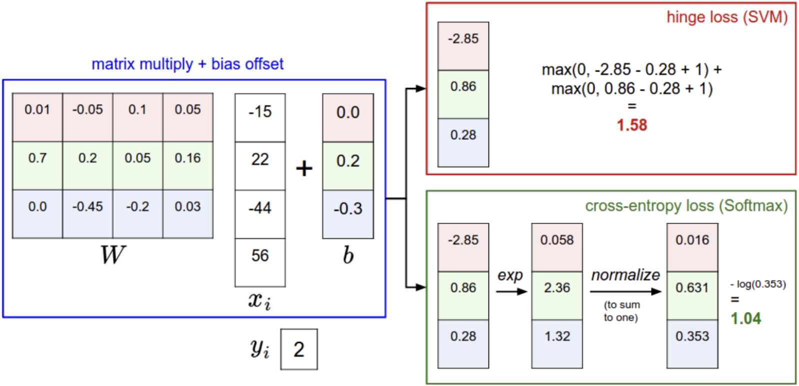
Loss Function (Cont'd)

```python
import numpy as np

# SVM classifier
def L1_i(x_i, y_i, W, b):
    delta = 1.0
    f = W.dot(x_i) + b
    margins = np.maximum(0, f - f[y_i] + delta)
    margins[y_i] = 0  # ignore true class
    return np.sum(margins)

# Softmax classifier
def L2_i(x_i, y_i, W, b):
    f = W.dot(x_i) + b
    f -= np.max(f)    # avoid potential blowup
    p = np.exp(f) / np.sum(np.exp(f))
    return -np.log(p[y_i])
```

SVM vs. Softmax



References

- cs231n.stanford.edu – CS231n: Deep Learning for Computer Vision
- Tang (2013), "Deep Learning using Linear Support Vector Machines", *arXiv:1306.0239*