3

Loss function and Optimization

⇒ Todo:

1. Determine a loss function

{quantifies our unhappiness with the scores across the training data}

2. Come up with a way of efficiently finding the parameters that minimize the loss function.

{Optimization}

⇒ Given a dataset of examples

$$\{(x_i, y_i)\}_{i=1}^{N}$$

where, x_i is image &

y_i is (integer) label

$$L = \frac{1}{N} \sum_i L_i(f(x_i, W), y_i)$$

★ Multiclass SVM loss

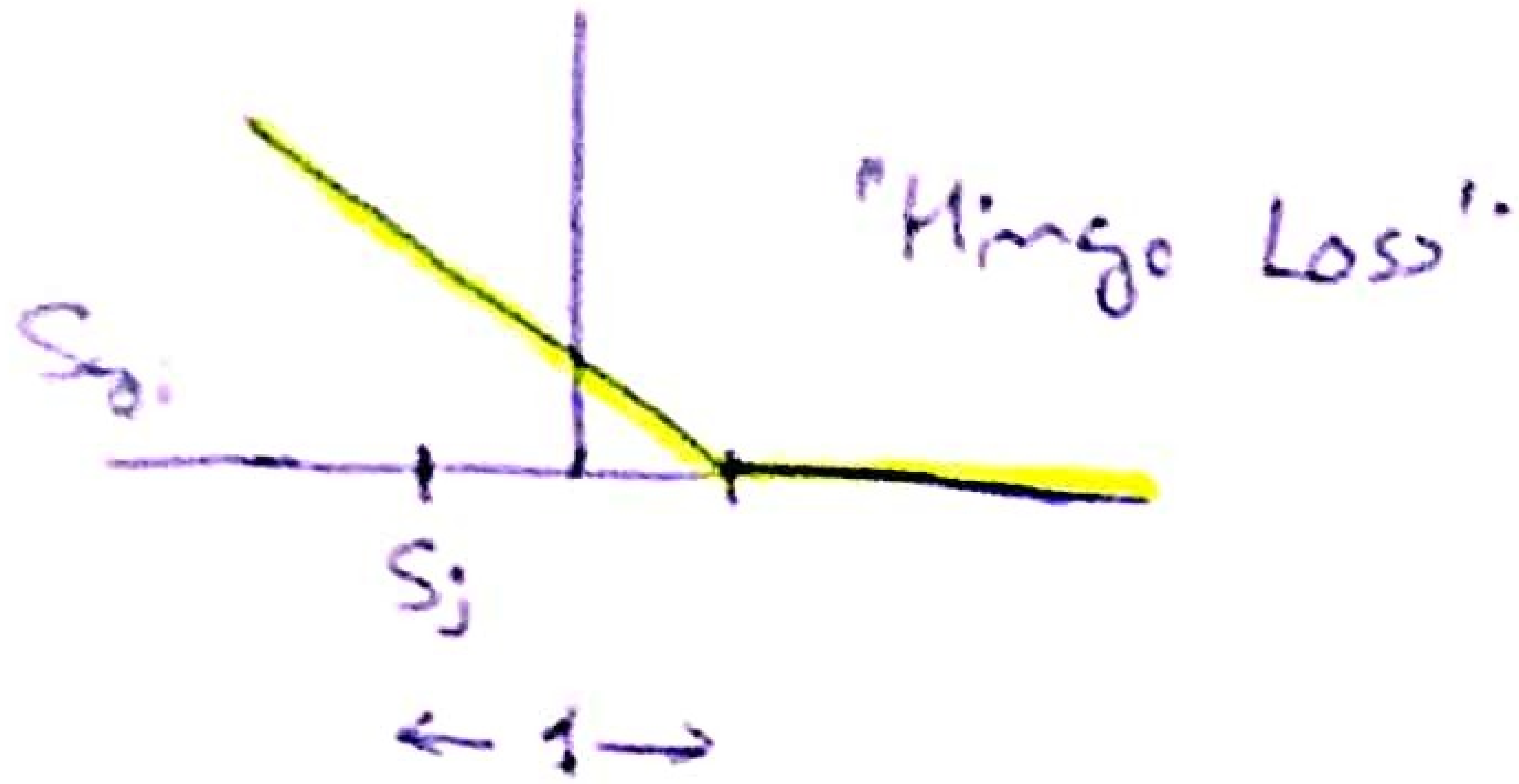
⇒ Let $S_i = f(x_i, W)$ be the shorthand for the score vector.

⇒ The SVM loss has the form:

$$L_i = \sum_{j \neq y_i} \begin{cases} 0 & \text{if } S_{y_i} \geq S_j + 1 \\ S_j - S_{y_i} + 1 & \text{otherwise} \end{cases}$$

$$= \sum_{j \neq y_i} \max(0, S_j - S_{y_i} + 1)$$

37:00 ②



$$L(W) = \frac{1}{N}\sum_{i=1}^{N} L_i(f(x_i, W), y_i) + \lambda R(W)$$

Regularization

Model should be "Simple", So it works On test data.

L2 regularization is most common

Occam's Razor

Among competing hypotheses the Simplest is the best

* Softmax Classifier (Multinomial Logistic Regression)

↳ Also called $\boxed{\text{Cross-entropy loss}}$

Score = unnormalized log probability of the classes

$$\boxed{P(Y=k \mid X=x_i) = \frac{e^{s_k}}{\sum_j e^{s_j}}}$$

where $S = f(x_i; W)$

⇒ We want to maximize the log likelihood or to minimize negative log likelihood of the correct class:

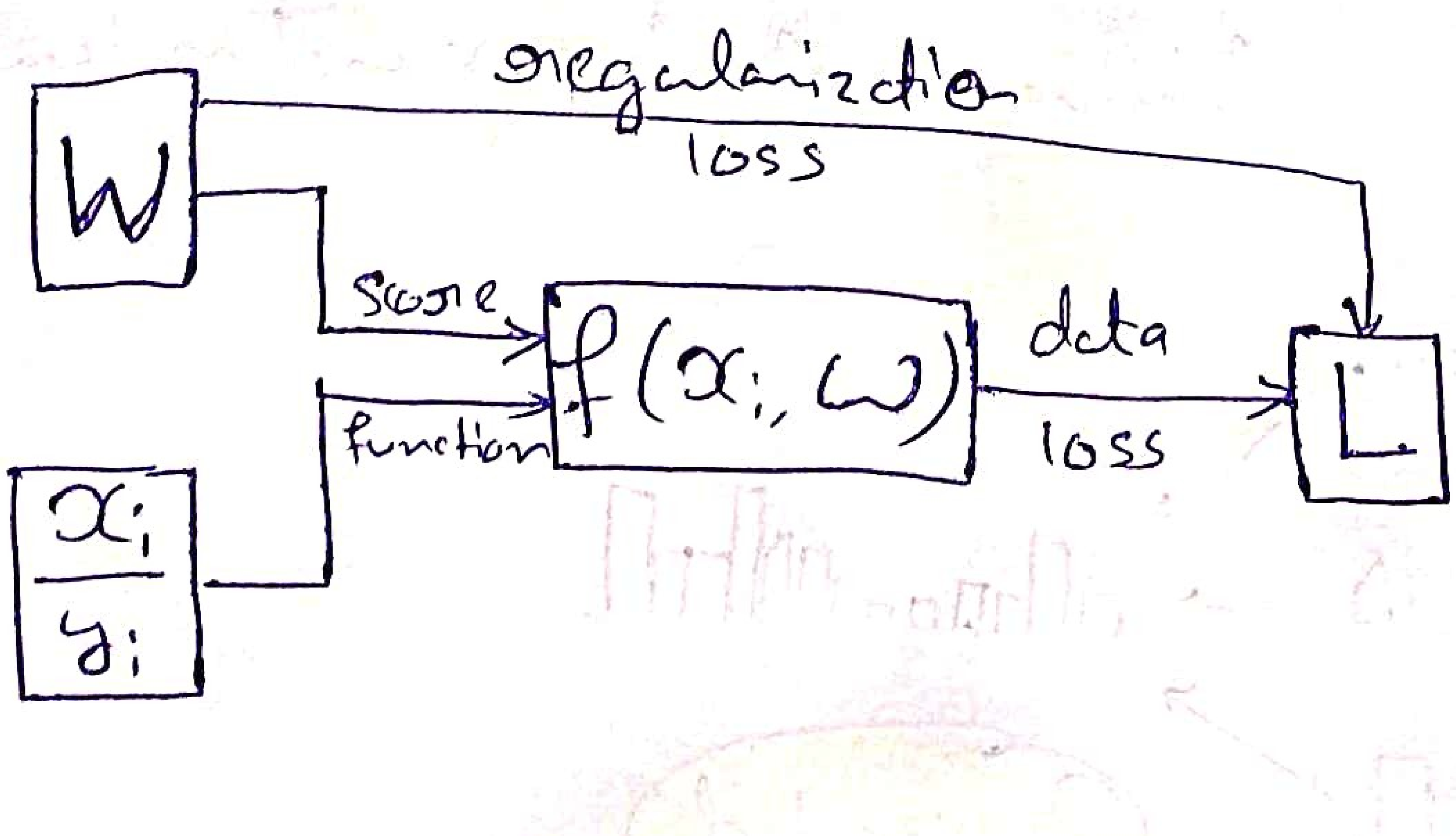
$$\boxed{L_i = -\log P(Y=y_i \mid X=x_i)}$$

↓

$$\boxed{L_i = -\log\left(\frac{e^{s_k}}{\sum_j e^{s_j}}\right)}$$

Minimum loss = 0

Maximum loss = ∞



★ Optimization

Strategy #1: Random Search

- Very bad Solution,
- Sample W randomly and evaluate all of these with loss function, & select the One with minimum loss.

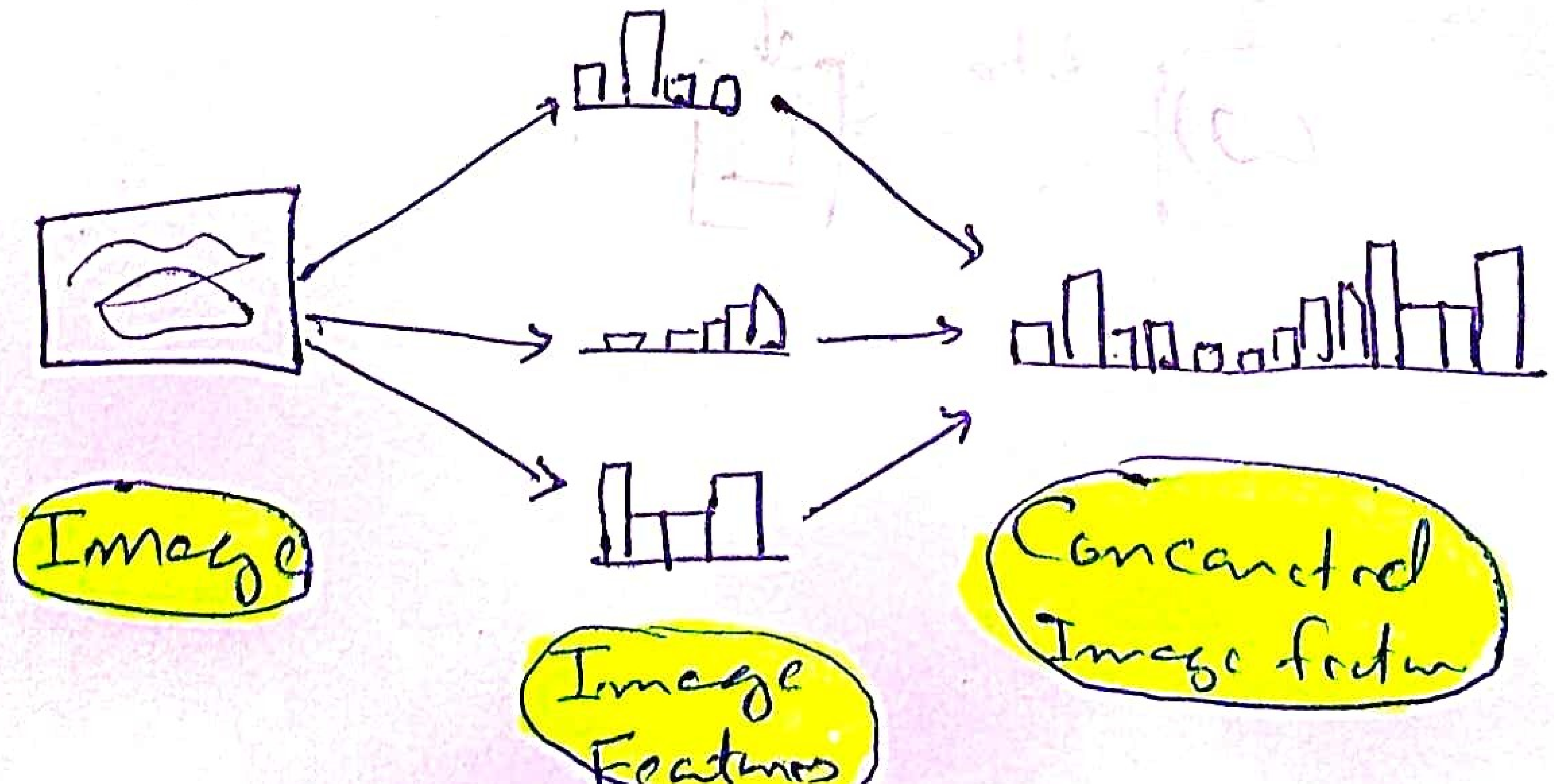
Strategy #2: Follow the Slope (Gradient decent)

- One way to calculate gradient is by finite difference method.
  (Not a good Idea, it is Very Slow!)
- Best is to use Claculus to compute an analytic gradient.

⇒ In practice always use analytic gradient, but check implementation with numerical gradient.
- This is called a gradient check

★ Image Features

⇒ Common before the dominance of deep neural network.



⇒ Color Histogram can be a feature vector.

⇒ Histogram of Oriented Gradient (HoG) can be a feature vector.

- Divide image into 8×8 pixel regions.
- Within each region quantize edge direction into 9 bins.

⇒ Bag of Words feature vector.

- Takes inspiration from Natural Language Processing.

⇒ In CNN, instead of writing down the features ahead of ~~the~~ time, we will learn the features.
