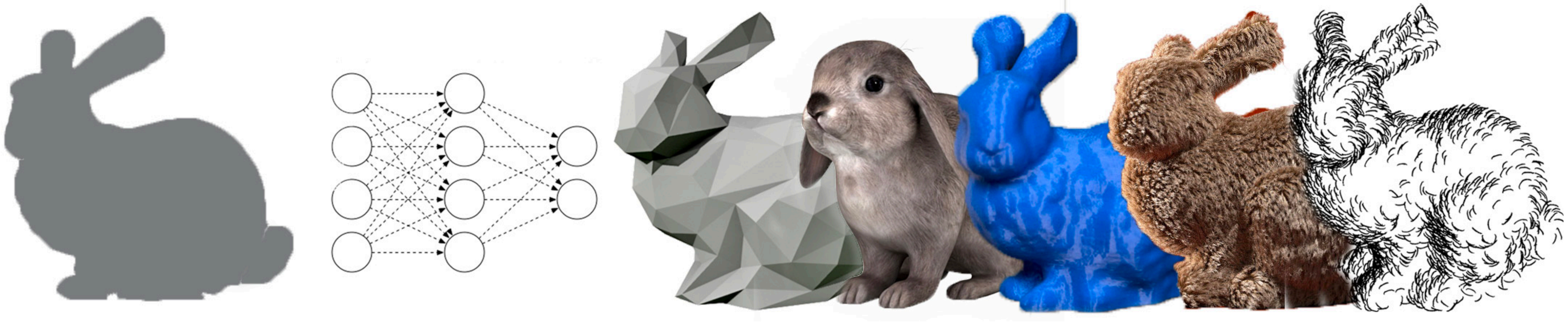
COMP0169: Machine Learning for Visual Computing

Linear and Nonlinear Models (Optimization)



Lectures will be Recorded

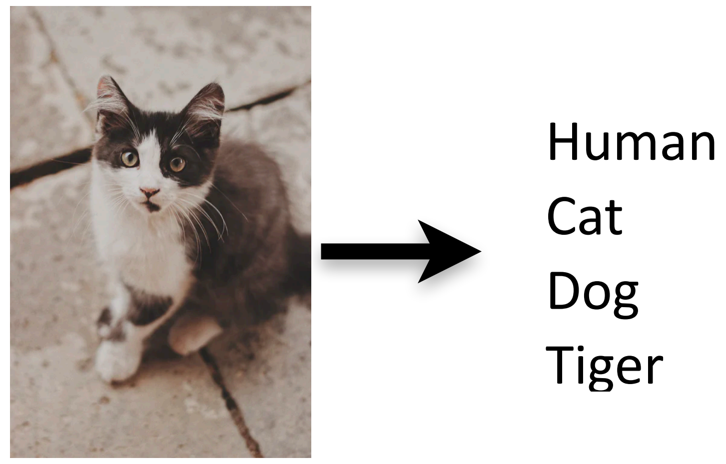
Last Week

Last Week

- Introduction and applications

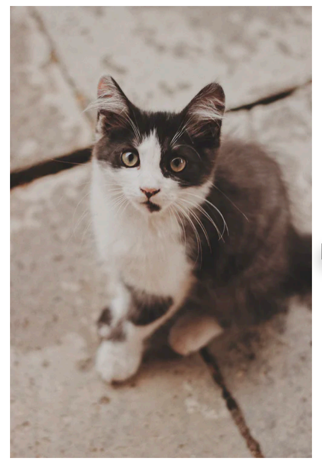
Last Week

- Introduction and applications
- Image classification

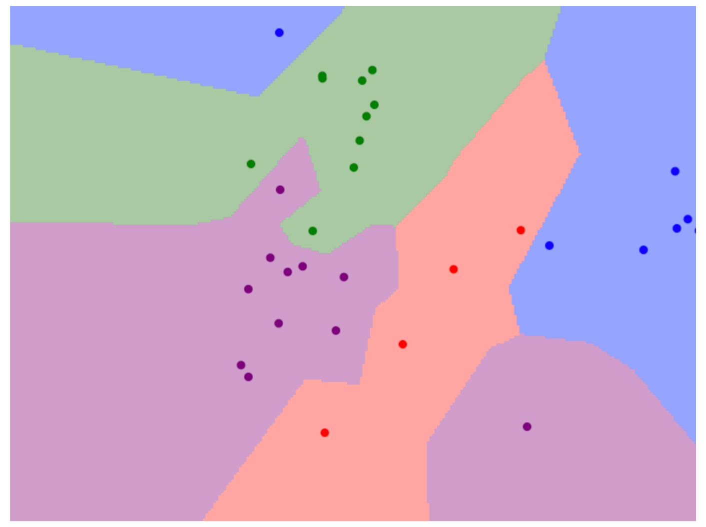


Last Week

- Introduction and applications
- Image classification
- kNN classifier

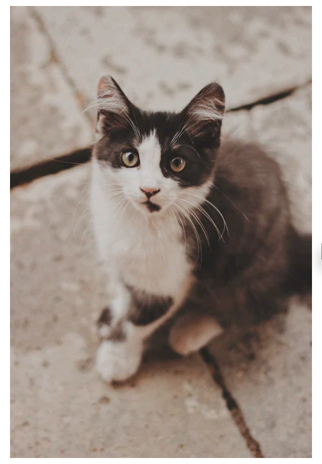


Human
Cat
Dog
Tiger

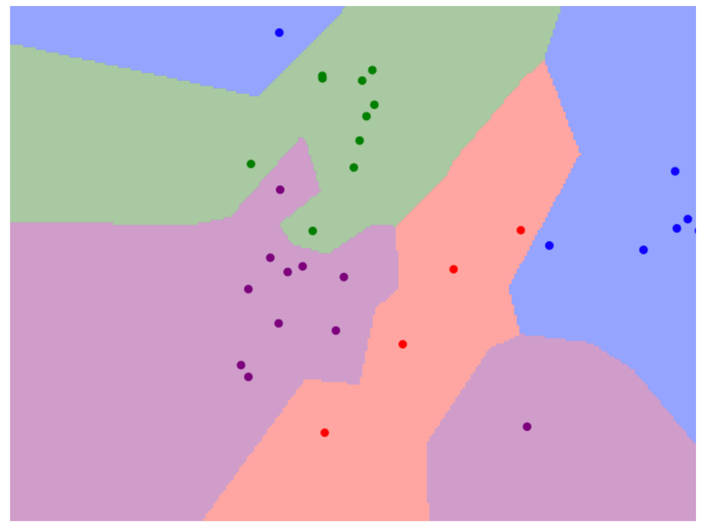


Last Week

- Introduction and applications
- Image classification
- kNN classifier
- LS Fitting

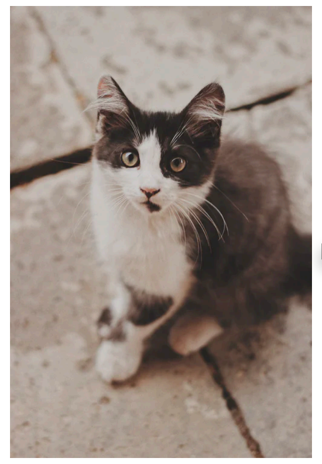


Human
Cat
Dog
Tiger

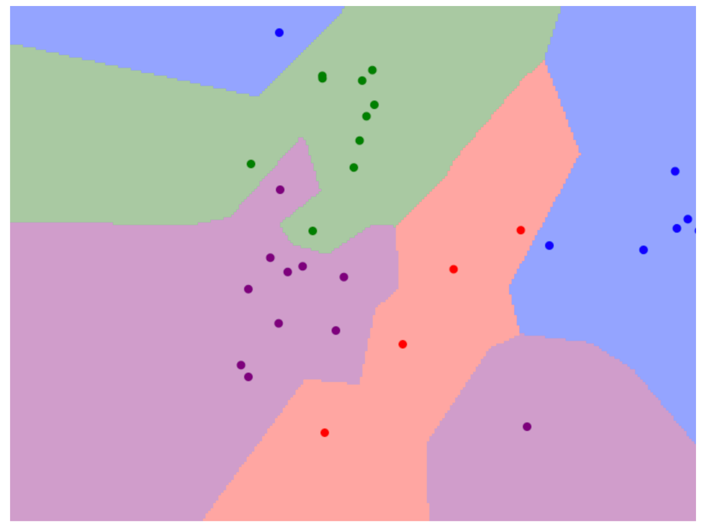


Last Week

- Introduction and applications
- Image classification



- kNN classifier

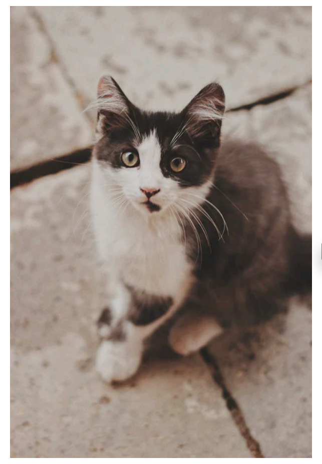


- LS Fitting
- Normal equation

$$\mathbf{w} = (X^T X)^{-1} X^T y$$

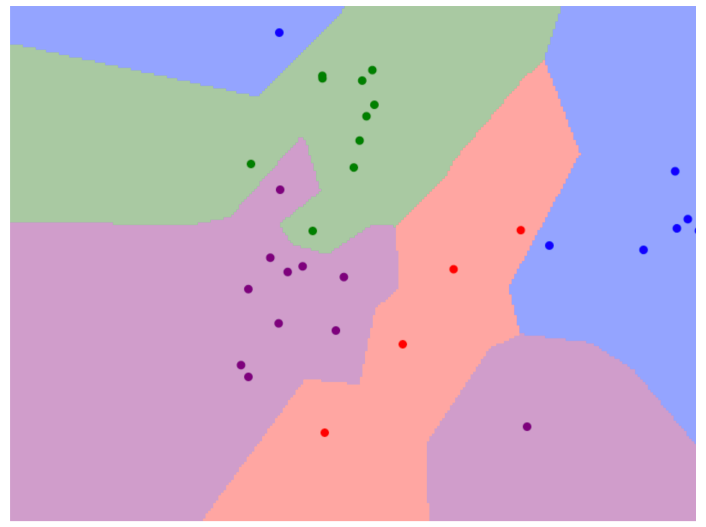
Last Week

- Introduction and applications



Human
Cat
Dog
Tiger

- Image classification



- kNN classifier
- LS Fitting
- Normal equation

$$\mathbf{w} = (X^T X)^{-1} X^T y$$

- Hyperparameter search (k-fold validation)

Image Classification

Image Classification

I

Image Classification

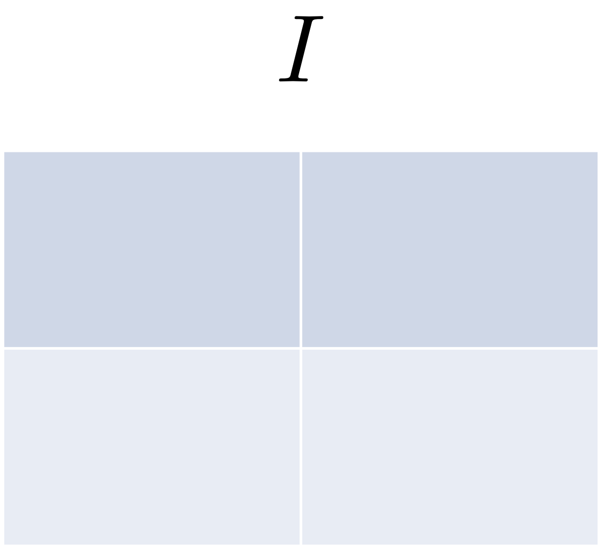


Image Classification

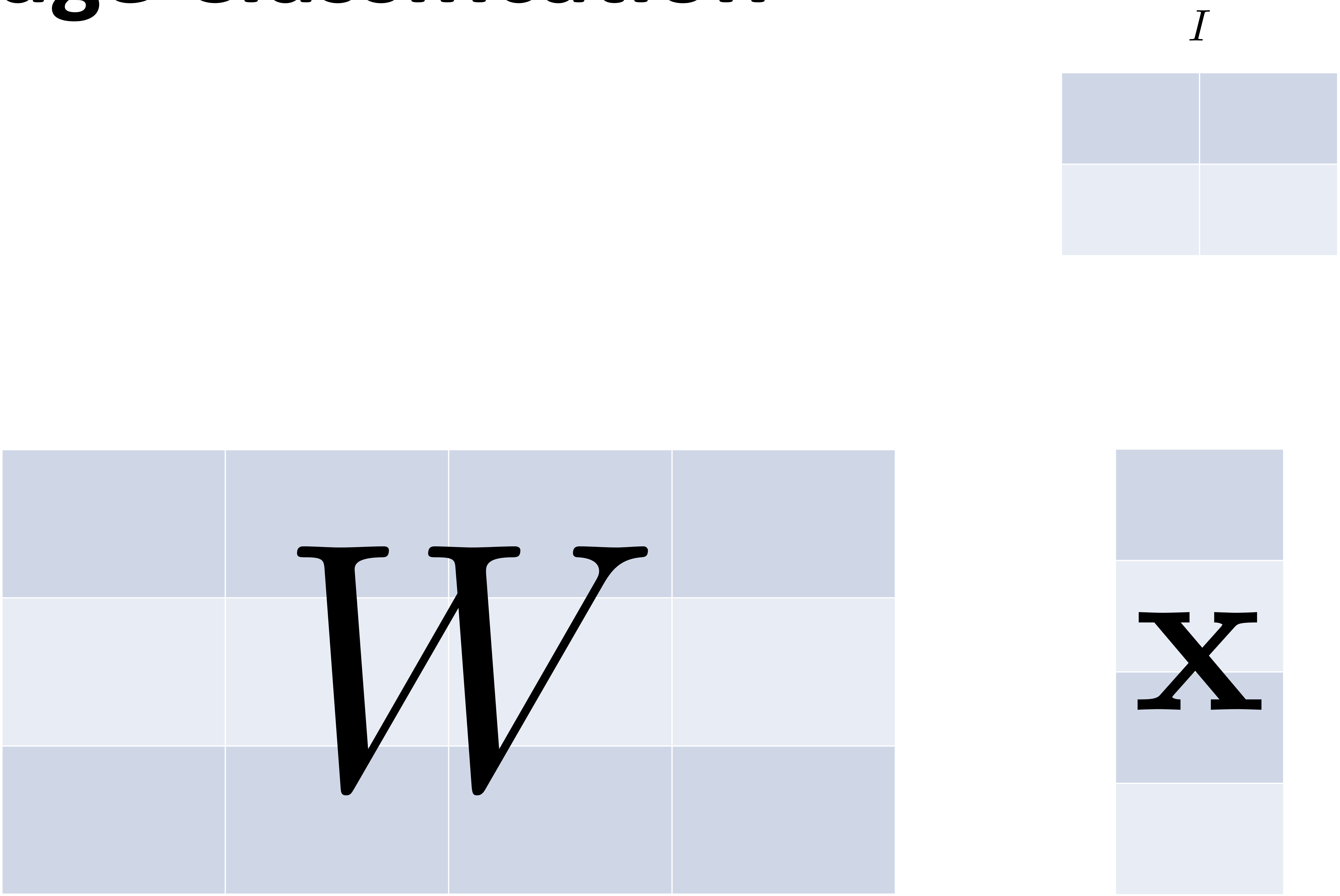


Image Classification

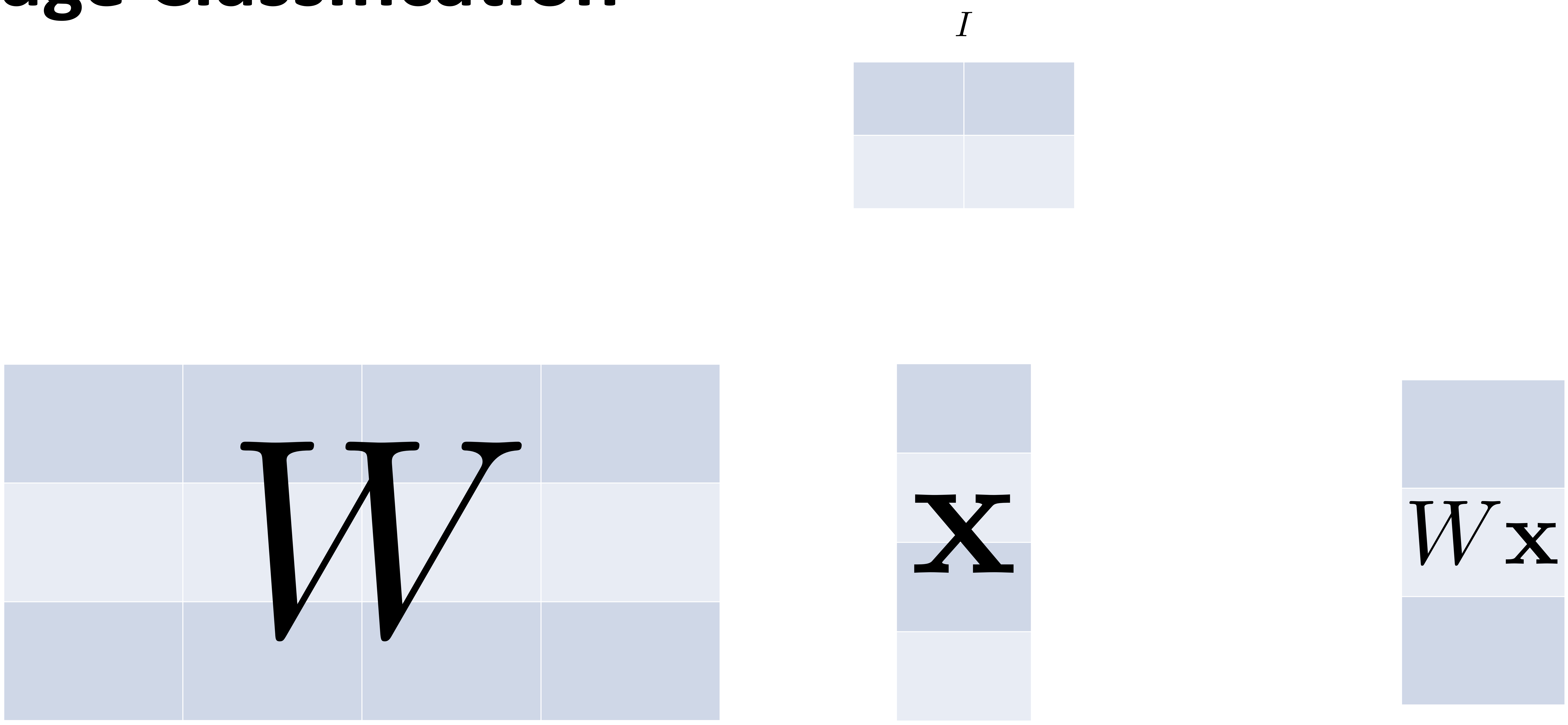


Image Classifier

Image Classifier

- Data (training/validation/test) $\{\mathbf{x}_i, y_i\}$

Image Classifier

- Data (training/validation/test) $\{\mathbf{x}_i, y_i\}$

- Scoring function
  (probability of belonging to class k) $f(\mathbf{x}_i, \mathbf{W})$

Image Classifier

- Data (training/validation/test) $\{\mathbf{x}_i, y_i\}$

- Scoring function
  (probability of belonging to class k) $f(\mathbf{x}_i, \mathbf{W})$

- Loss function $L_i(W) := h(f(\mathbf{x}_i, W), y_i) \quad L(W) := \frac{1}{N}\sum_i L_i(W)$

Image Classifier

- Data (training/validation/test) $\{\mathbf{x}_i, y_i\}$

- Scoring function
  (probability of belonging to class k) $f(\mathbf{x}_i, \mathbf{W})$

- Loss function $L_i(W) := h(f(\mathbf{x}_i, W), y_i) \quad L(W) := \frac{1}{N}\sum_i L_i(W)$

- Optimization $\mathbf{W}^* := \arg\min L(W) = \arg\min L_i(W)/N$

Linear versus Nonlinear Models

$$s = f(\mathbf{x}_i, W) = W\mathbf{x}_i$$

Linear versus Nonlinear Models

$$s = f(\mathbf{x}_i, W) = W\mathbf{x}_i$$

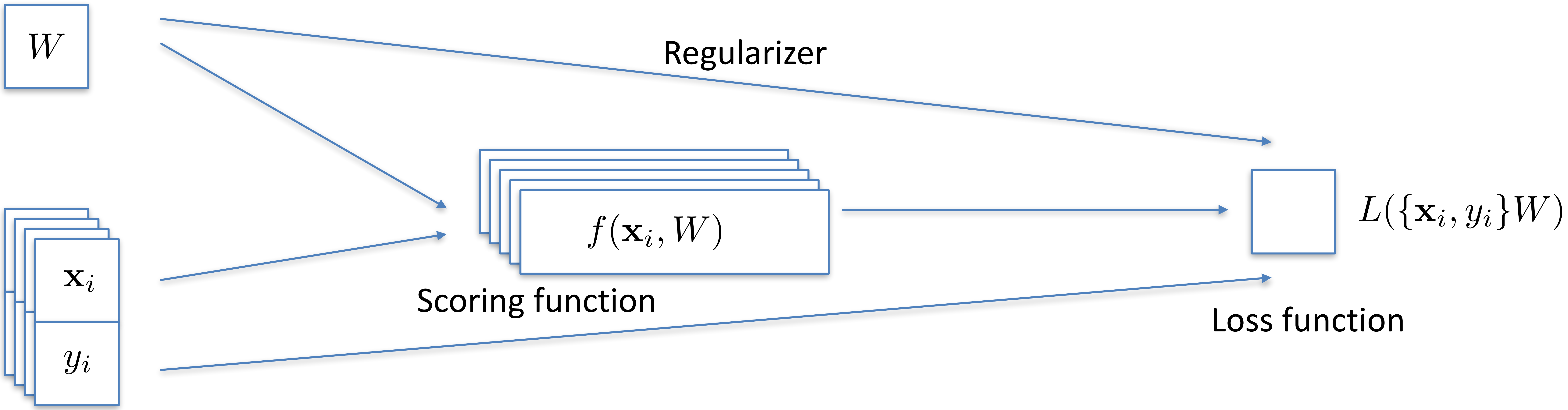
- Objective function #1 $\quad L_i(W) := (Wx_i - y_i)^2$

Linear versus Nonlinear Models

$$s = f(\mathbf{x}_i, W) = W\mathbf{x}_i$$

- Objective function #1 $$L_i(W) := (Wx_i - y_i)^2$$
- Objective function #2 $$L_i(W) := -\log\left(\frac{e^{Wx_iy_i}}{\sum_j e^{Wx_jy_j}}\right) = -\log\left(\frac{e^{sy_i}}{\sum_j e^{sy_j}}\right)$$

Image Classifier



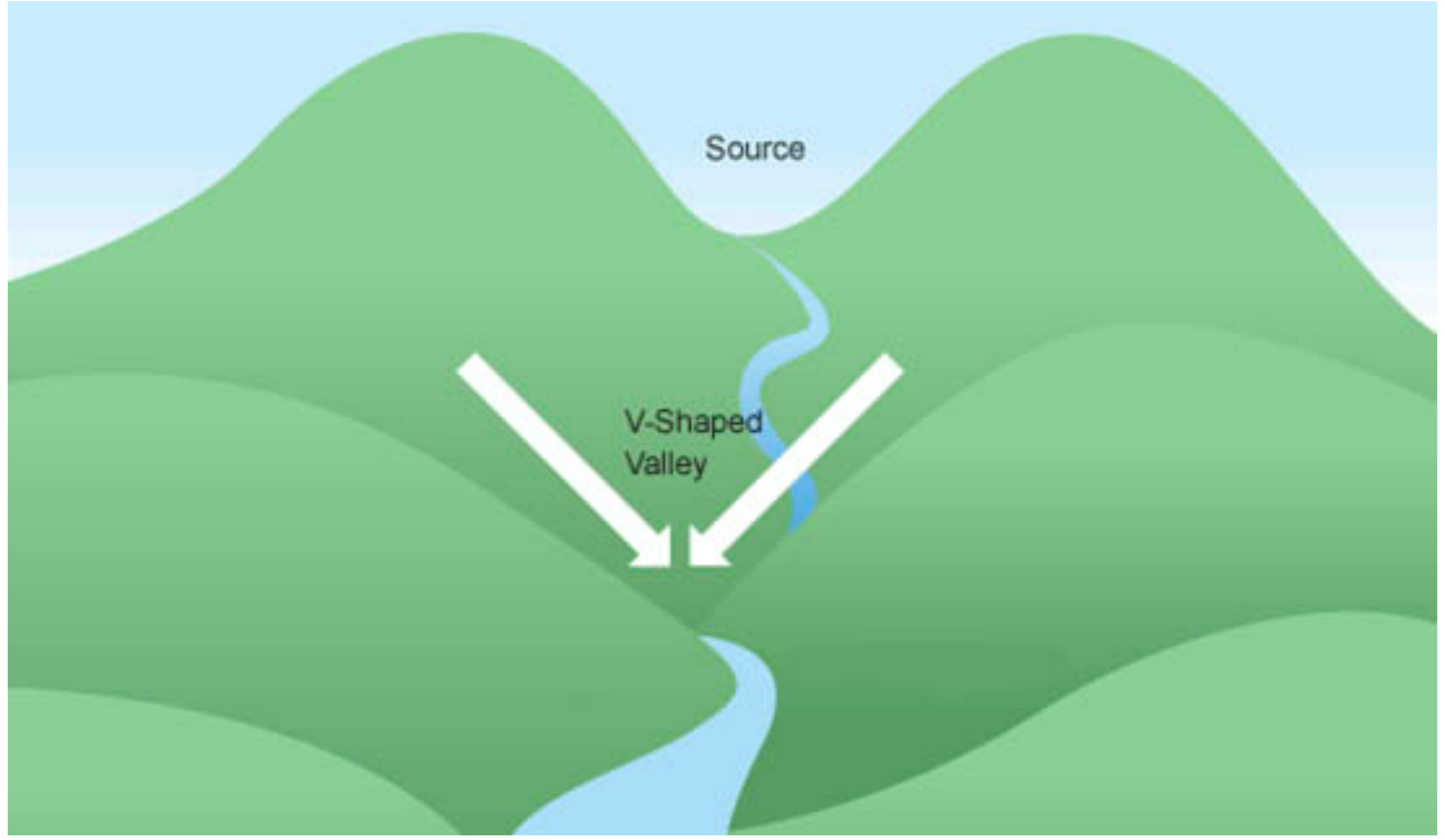
Optimize

$$\mathbf{W}^* = \arg\min g(\mathbf{x}_i, W)$$

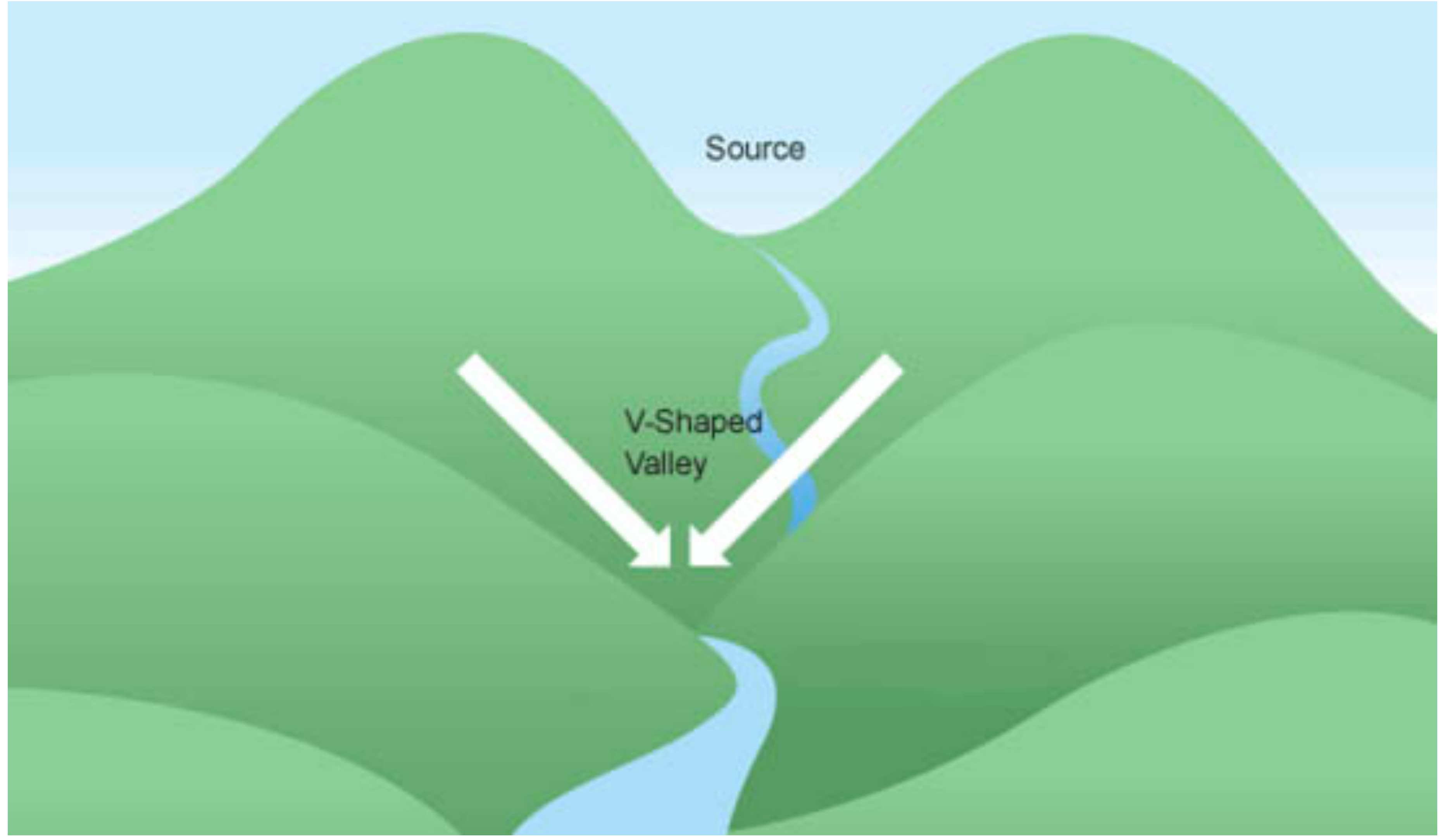
$$\nabla g(\mathbf{x}_i, W) = \left[\frac{\partial g(\mathbf{x}_i, W)}{w_i} = 0 \right]_{D \times 1}$$

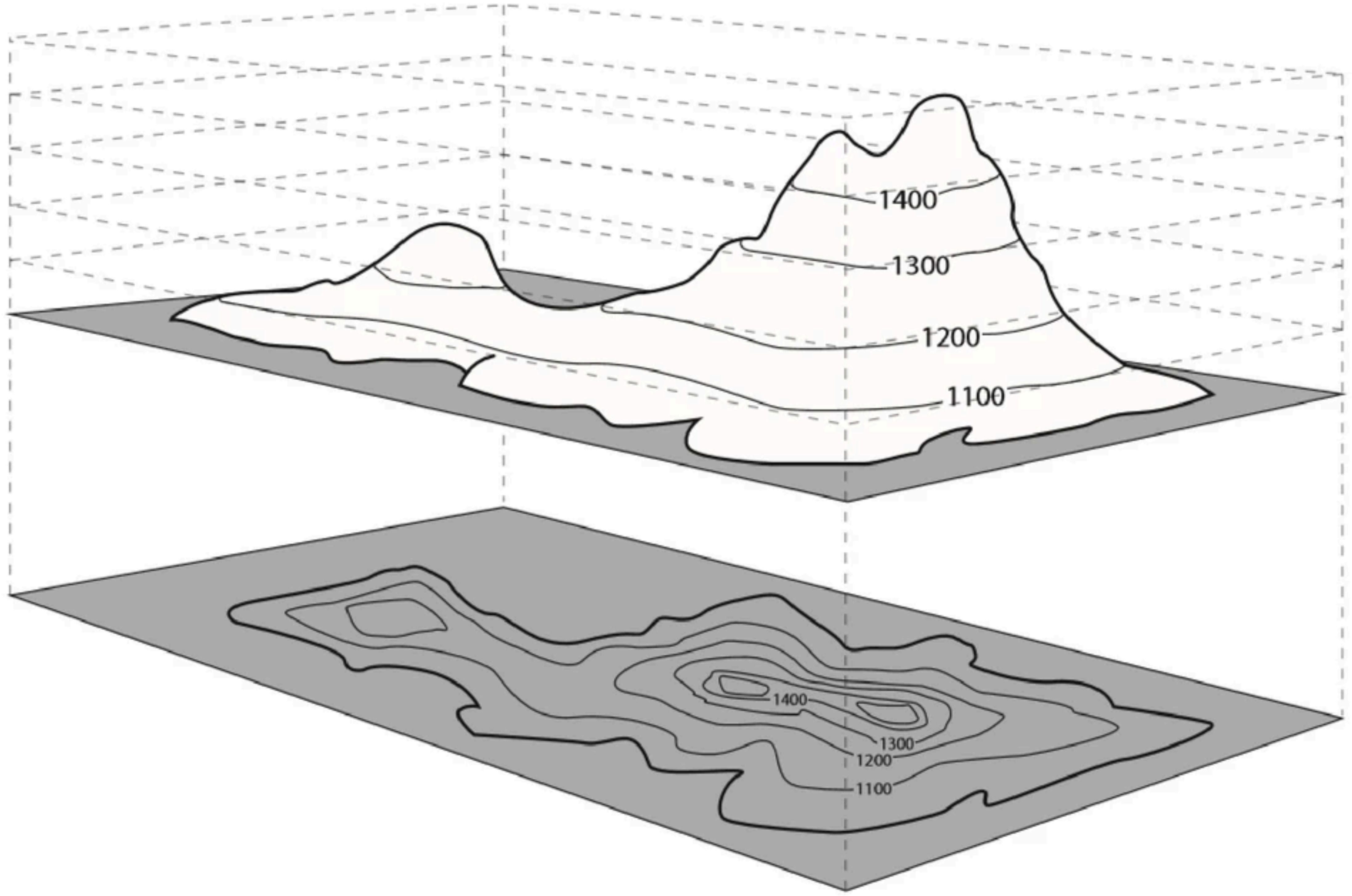
Error Landscape

Error Landscape



Error Landscape

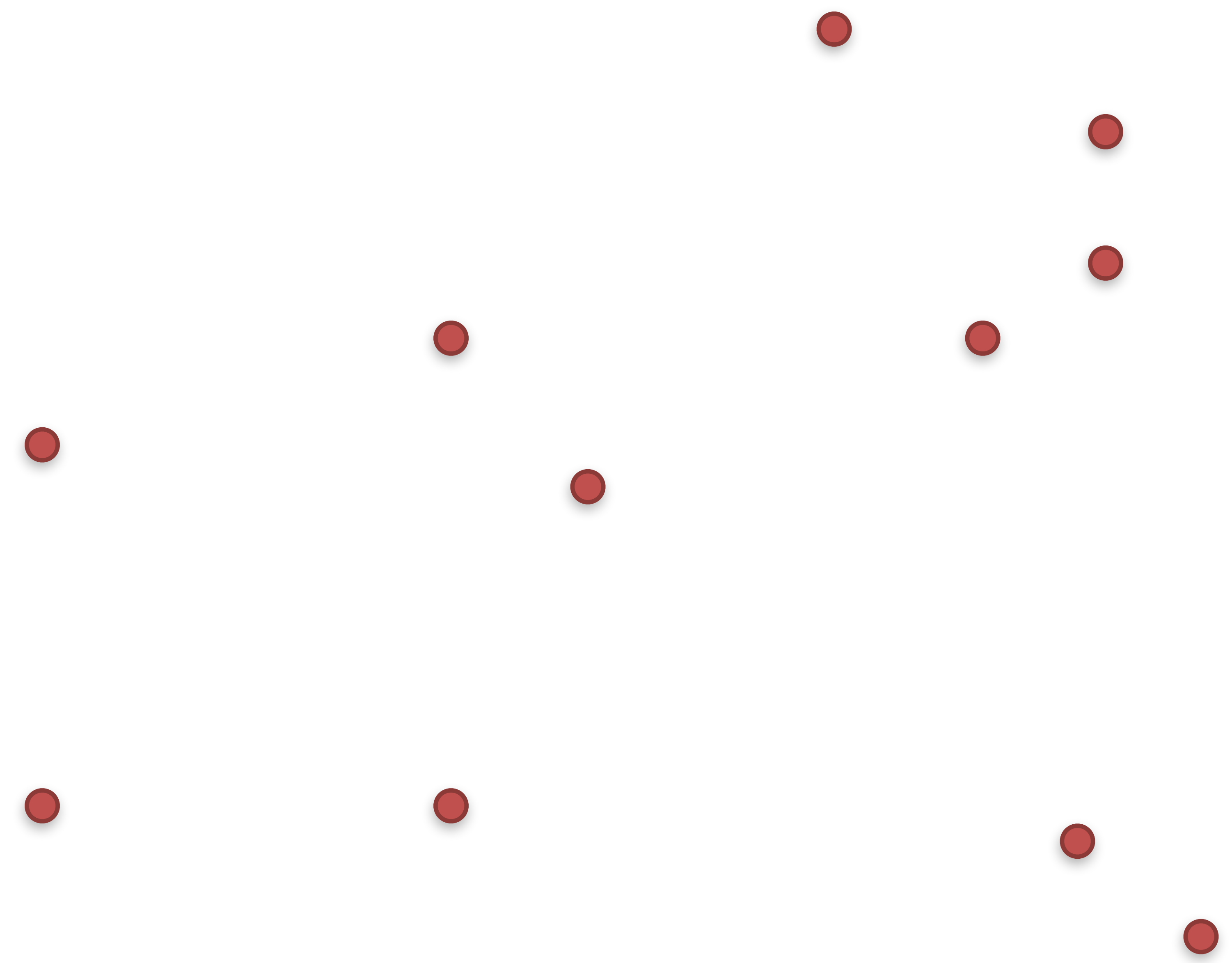




Method 1a: Random search

- bestLoss = inf
- Loop

```
    Pick random W
    Compute loss
    if loss<bestLoss
        bestLoss = loss
        currW = W
     endif
endLoop
```



Method 1a: Random search

- bestLoss = inf
- Loop

  Pick random W

  Compute loss

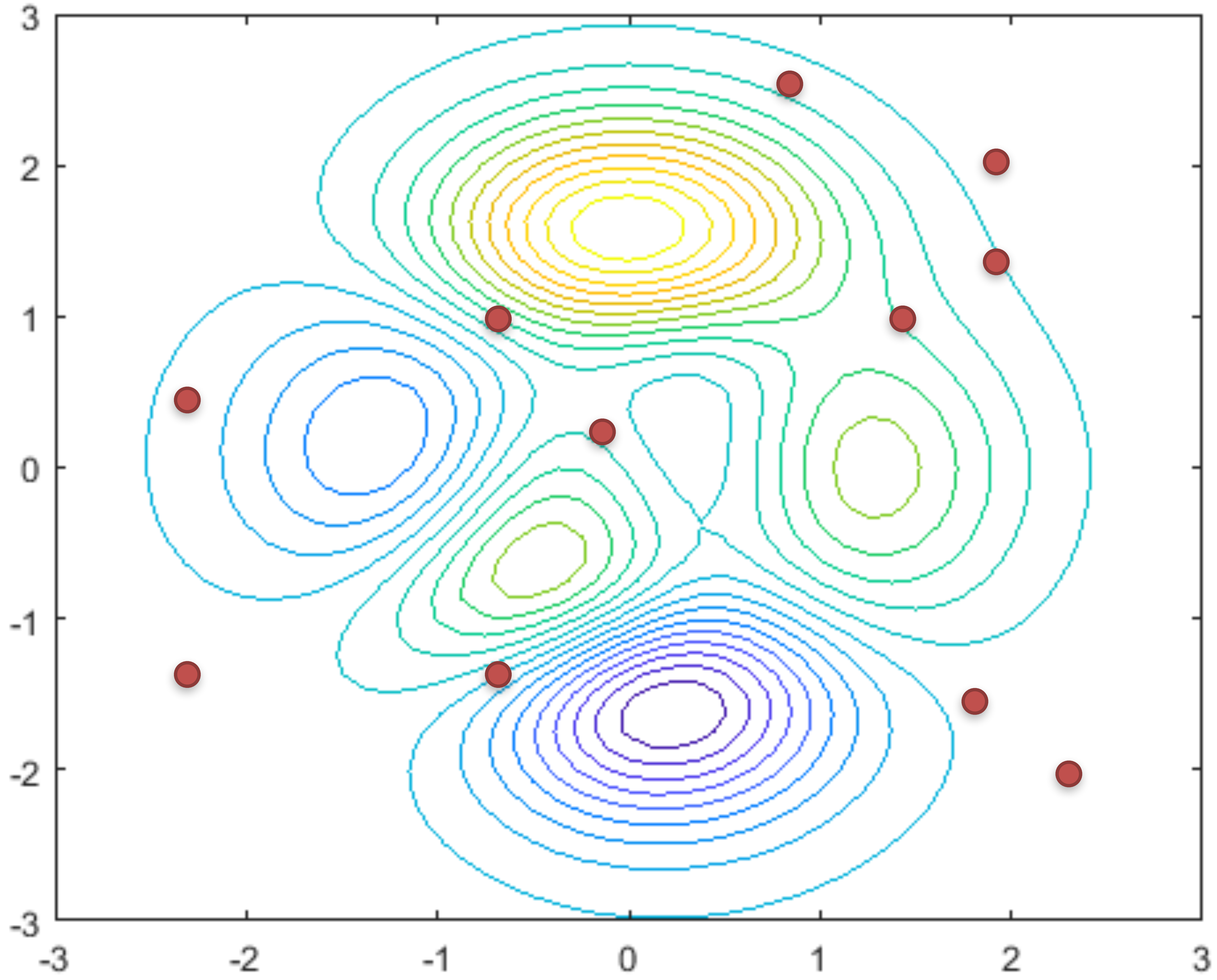
  if loss<bestLoss

  bestLoss = loss

  currW = W

  endif

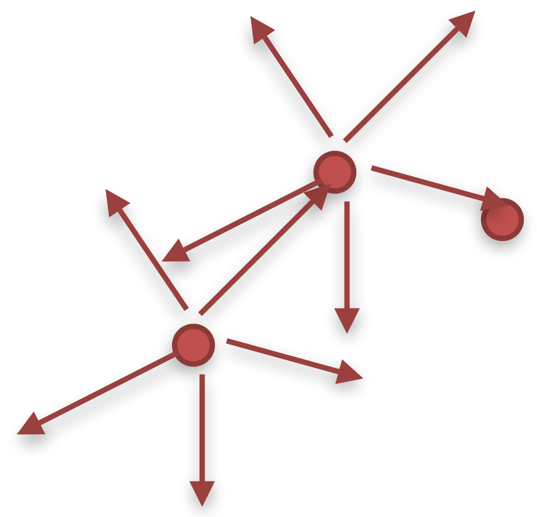
  endLoop



Method 1b: Random search++

- bestLoss = inf
- W = setRand
- stepsize
- Loop

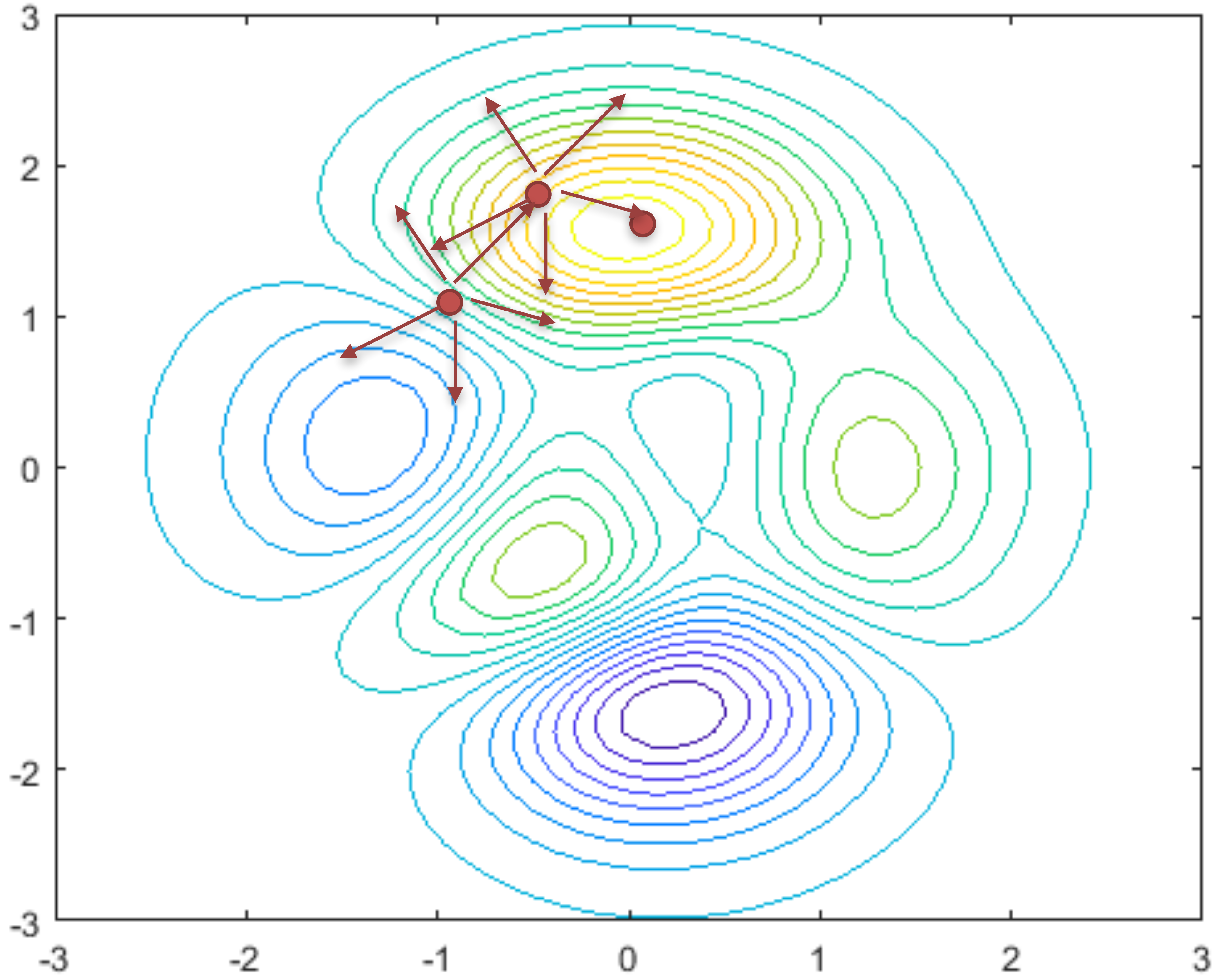
```
    W_local = W + stepsize*W_random
    Compute loss
    if loss<bestLoss
        bestLoss = loss
        W = W_local
    endif
endLoop
```



Method 1b: Random search++

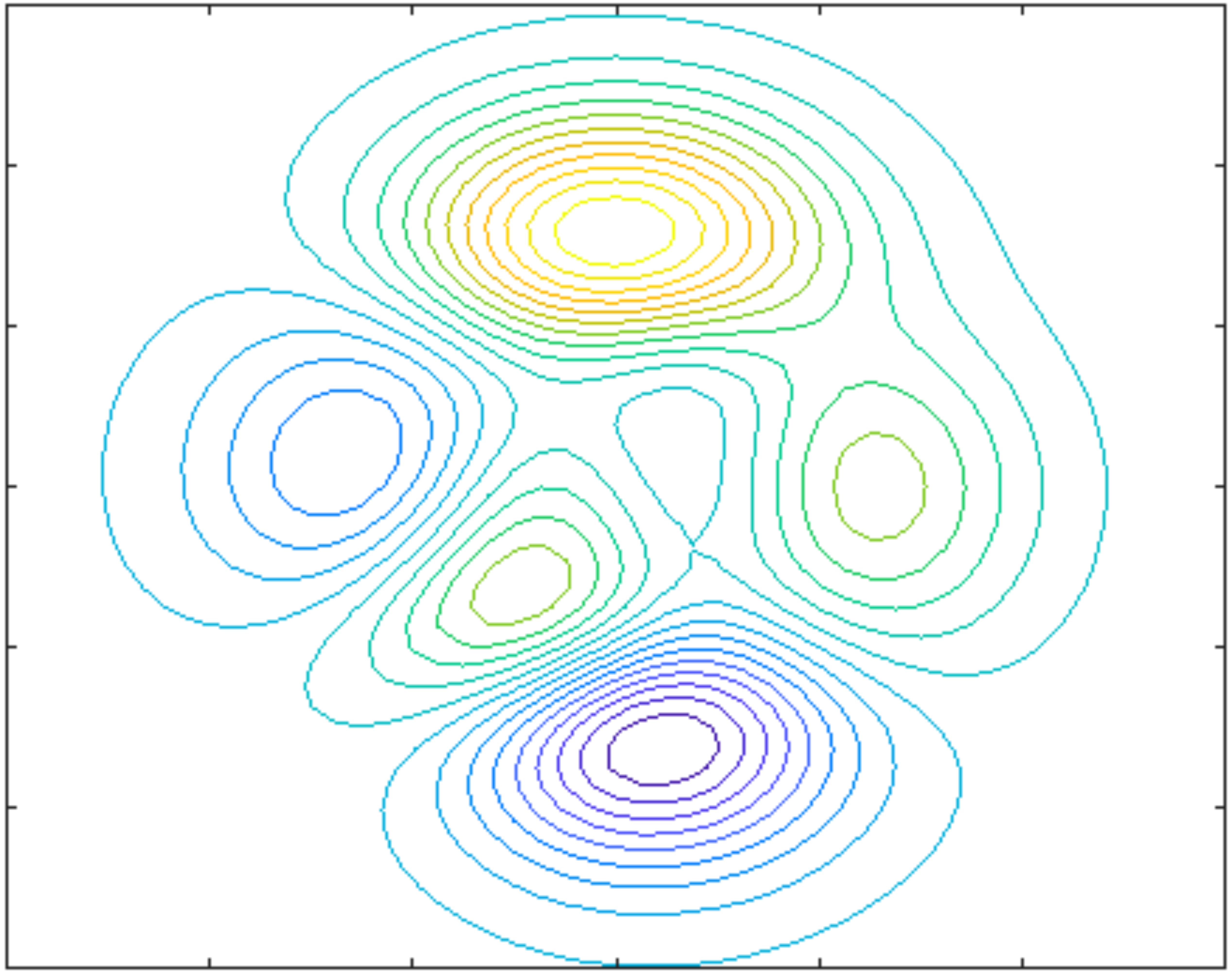
- bestLoss = inf
- W = setRand
- stepsize
- Loop

```
    W_local = W + stepsize*W_random
    Compute loss
    if loss<bestLoss
        bestLoss = loss
        W = W_local
     endif
endLoop
```

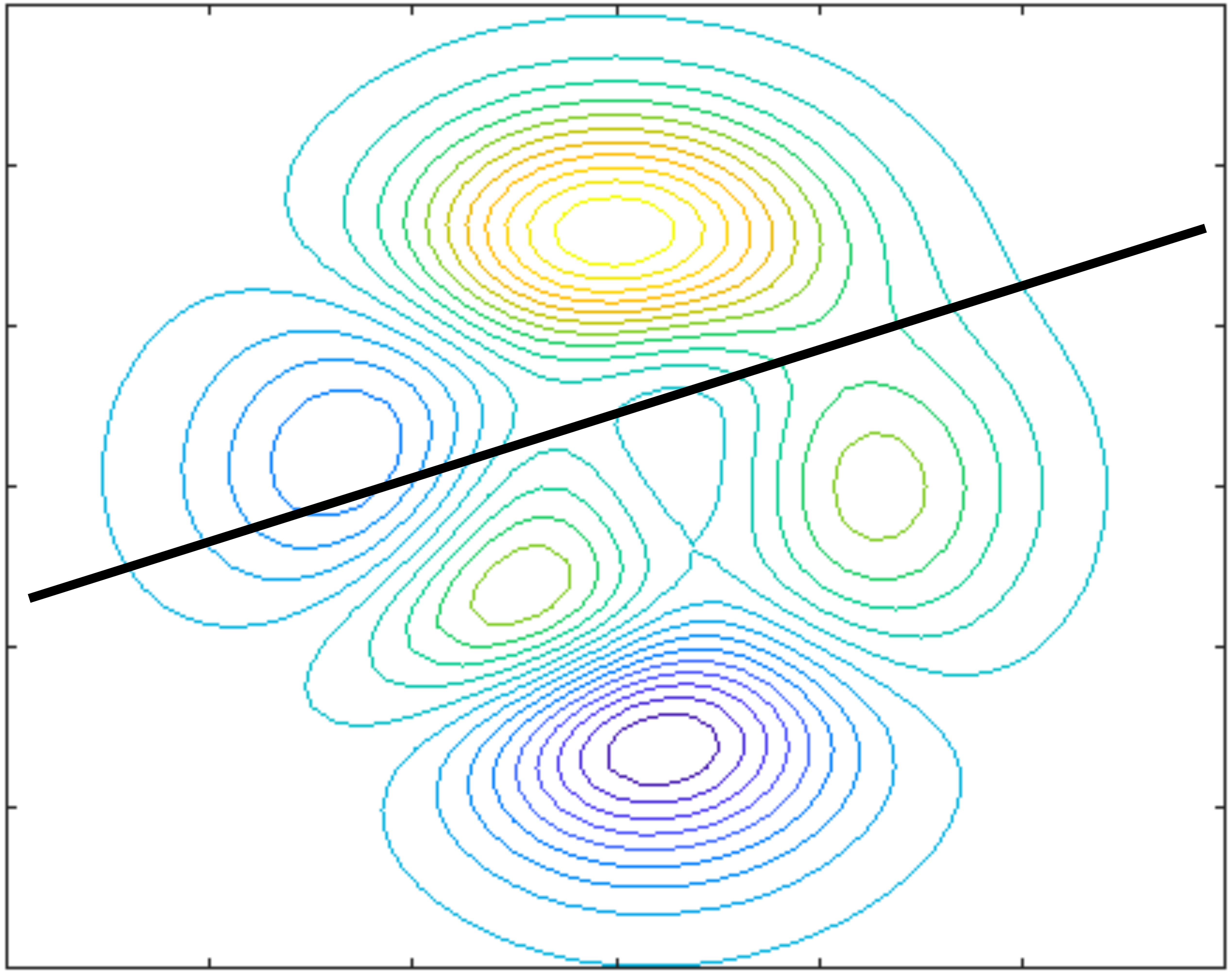


Method 2: Follow Negative of Gradient

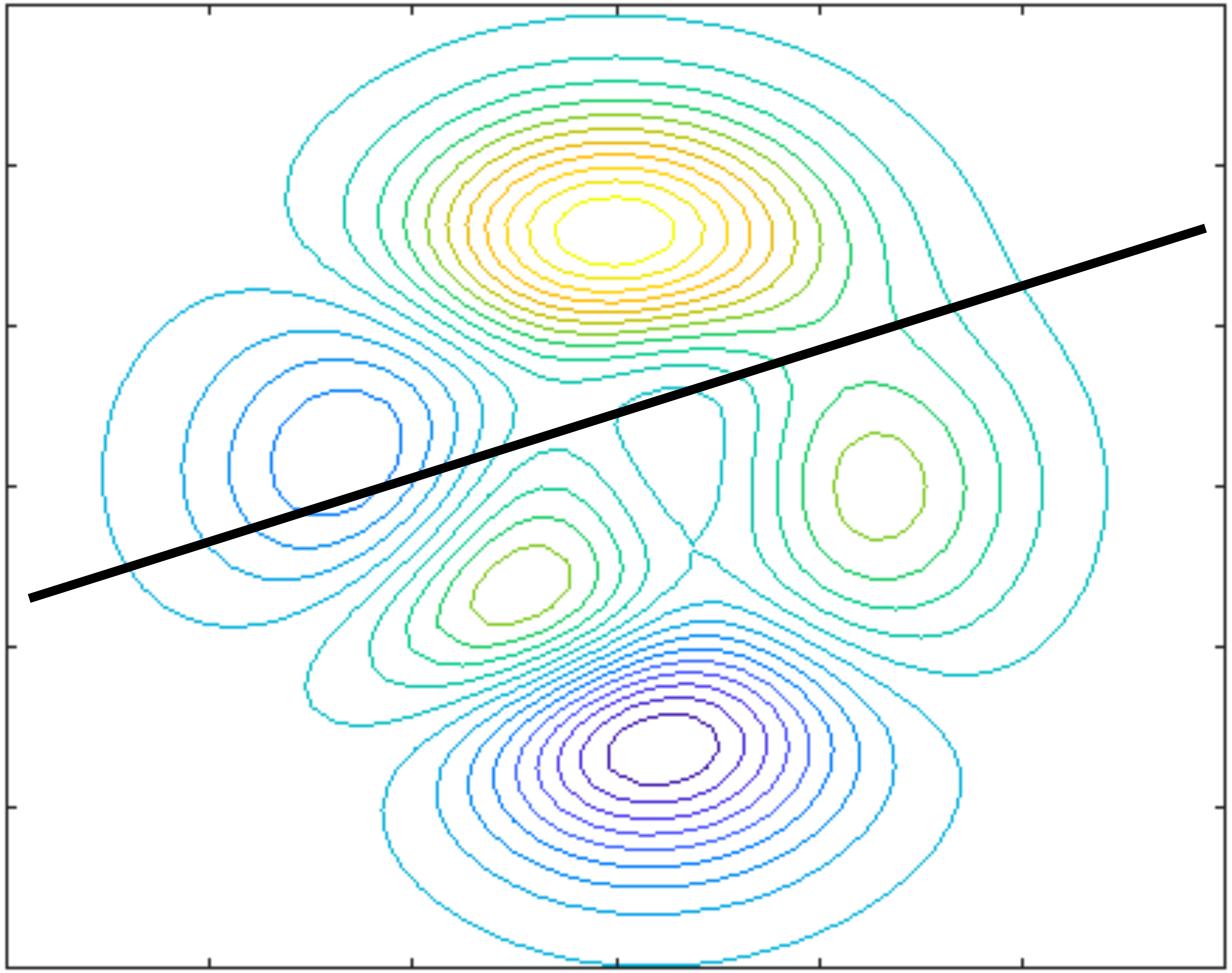
Method 2: Follow Negative of Gradient

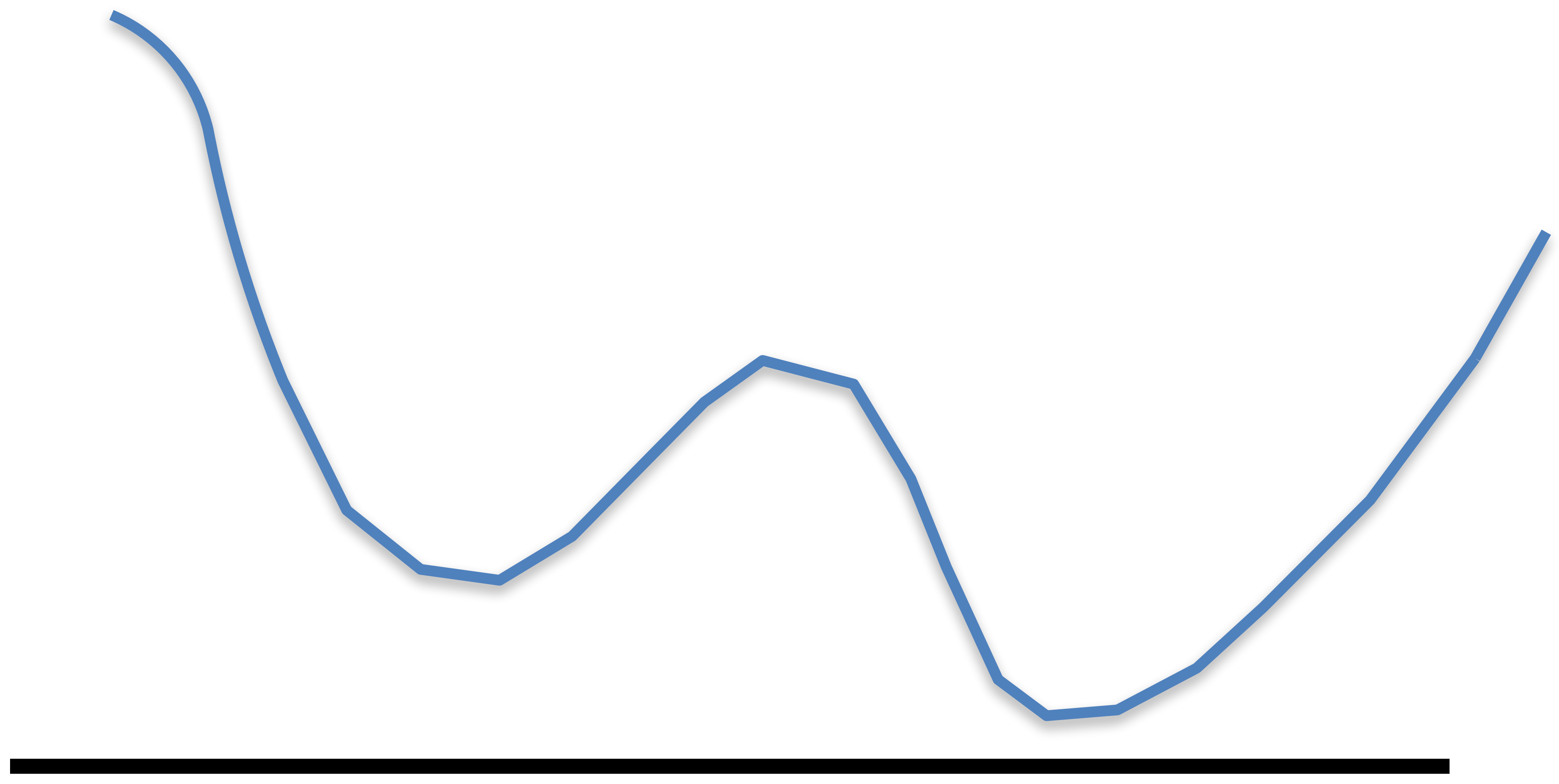


Method 2: Follow Negative of Gradient

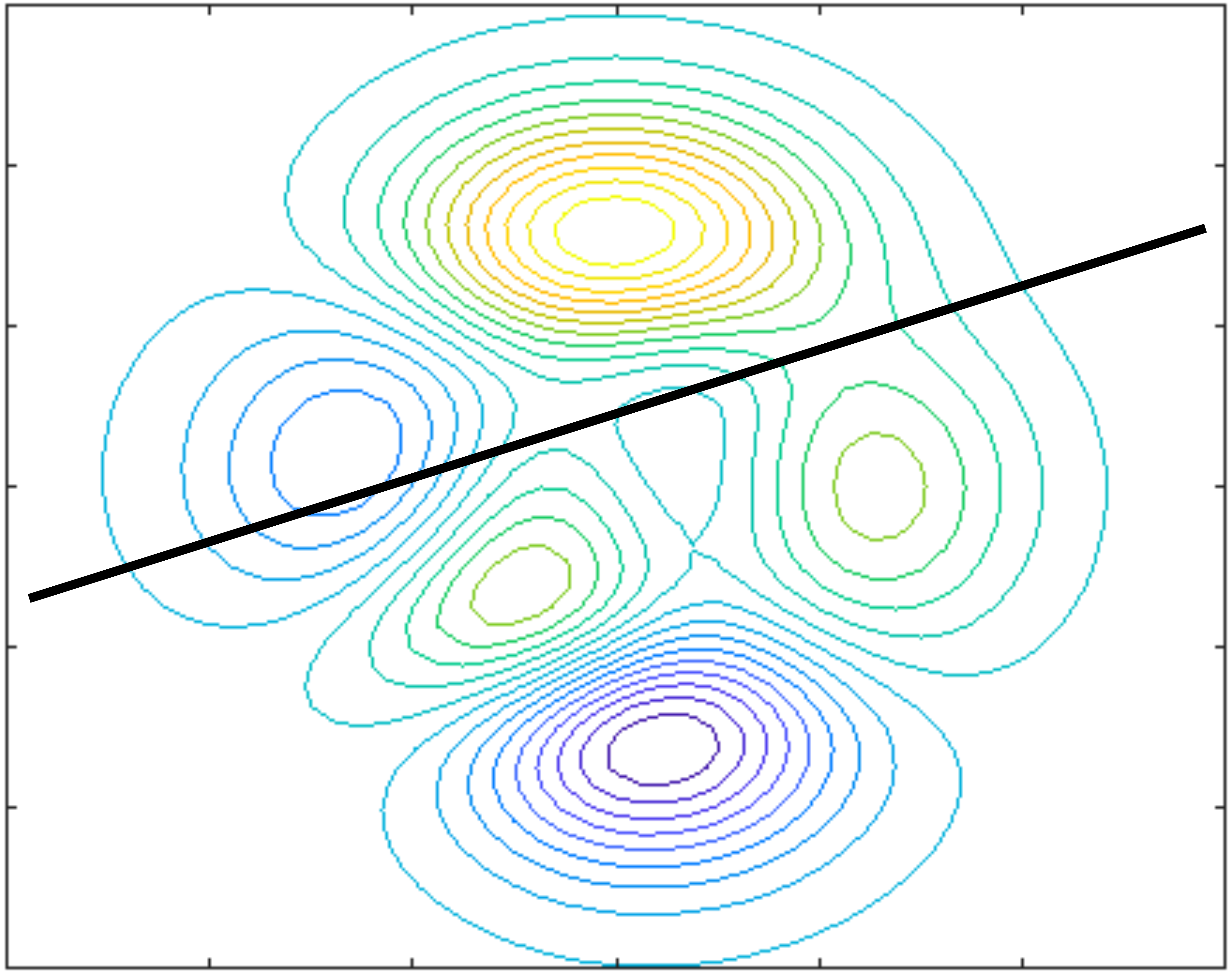


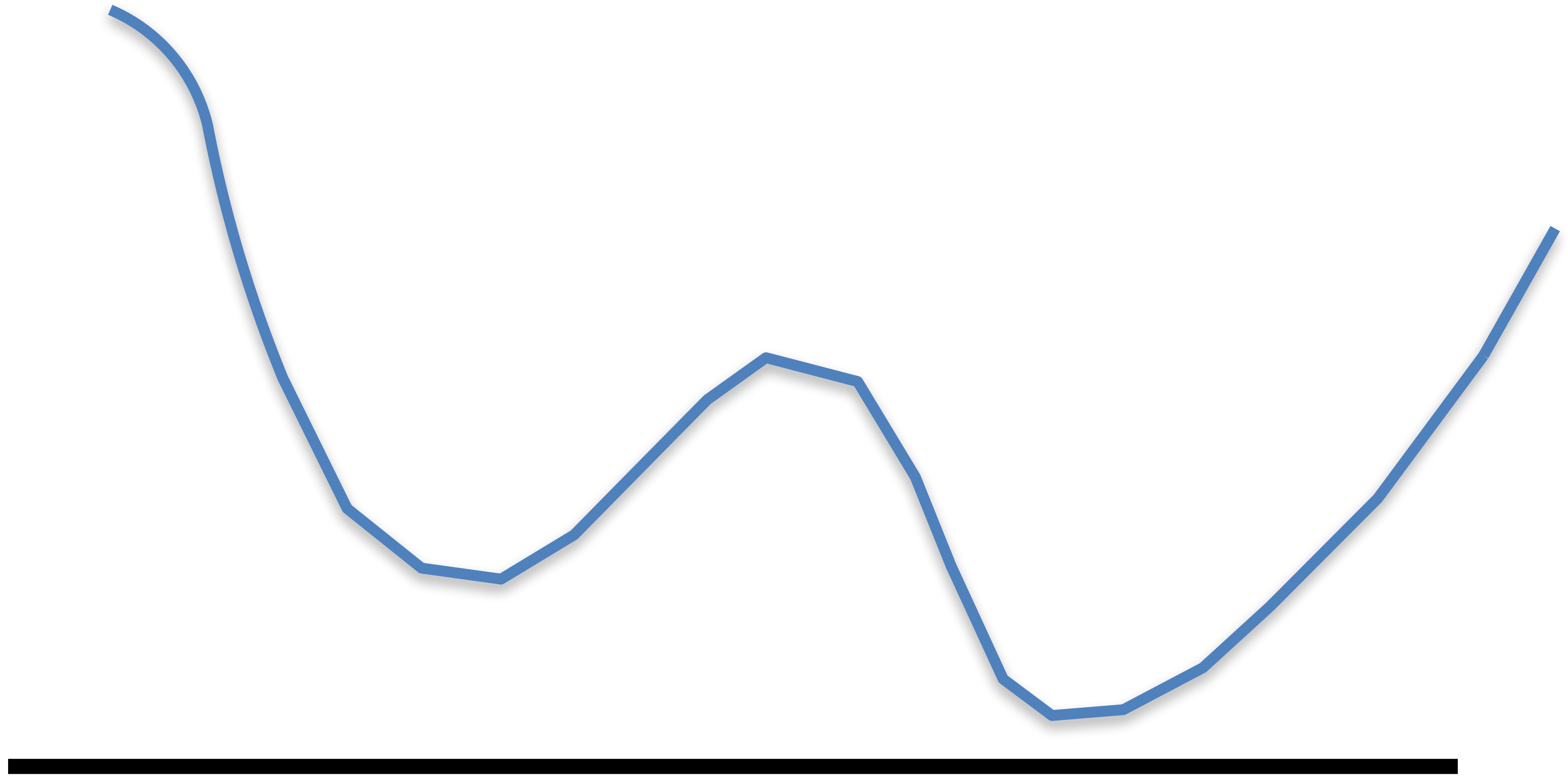
Method 2: Follow Negative of Gradient





Method 2: Follow Negative of Gradient





$$\frac{df(x)}{dx}$$