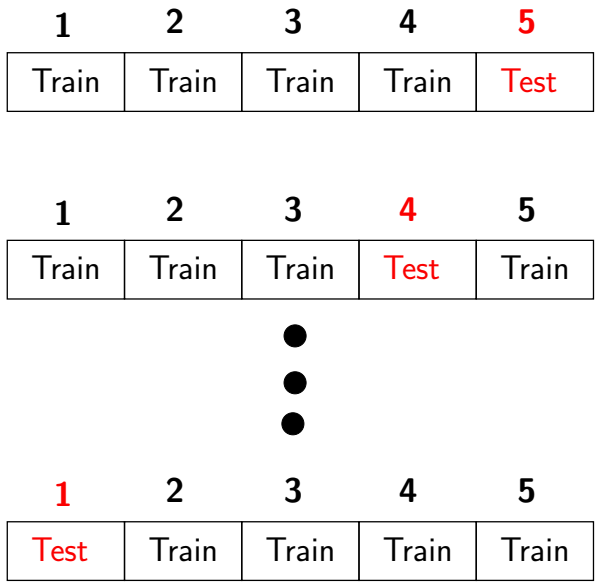
Cross-Validation & Prediction Error

Most commonly used method for estimating the prediction error (generalization error) on new data is *cross-validation*.



Split training data into K roughly equal-sized parts (e.g. $K = 5$). Learn neural network on $K - 1$ training parts and predict the not seen testing part. Perform this for $k = 1, 2, \ldots, K$ and combine the K estimates of prediction error (test error).

Cross-Validation & Prediction Error (cont.)

Typical choices of K are 5 or 10. If training data set consists of N data points and $K = N$, then one obtains the *leave-one-out* cross-validation method.

```
# number of folds
folds <- 10
samps <- sample(rep(1:folds, length=n), n, replace=FALSE)

for (fold.iter in 1:folds) {

  train <- dataset[samps!=fold.iter,] # fit the model
  test  <- dataset[samps==fold.iter,] # predict

  # perform here neural network training on set train
  # perform here neural network prediction on set test
}
```

Bias-Variance Dilemma

- Idea: decompose generalization error into the two terms (bias,variance).
- Let us consider our curve fitting problem again. Let $h(\mathbf{x})$ be the true (but unknown) function with continuous valued output with noise.
- We want to estimate $h(\cdot)$ based on training data sets $\mathcal{D}$ each of size N.
- The natural way to measure the effectiveness of the estimator is to use the mean-square deviation from the desired optimal.
- Averaging over all training data sets $\mathcal{D}$ one gets the decomposition:

$$\mathbb{E}_{\mathcal{D}}\left[\{y(\mathbf{x},\mathcal{D}) - h(\mathbf{x})\}^2\right]$$
$$= \underbrace{\{\mathbb{E}_{\mathcal{D}}[y(\mathbf{x},\mathcal{D})] - h(\mathbf{x})\}^2}_{\text{bias}^2} + \underbrace{\mathbb{E}_{\mathcal{D}}\left[\{y(\mathbf{x},\mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x},\mathcal{D})]\}^2\right]}_{\text{variance}}$$

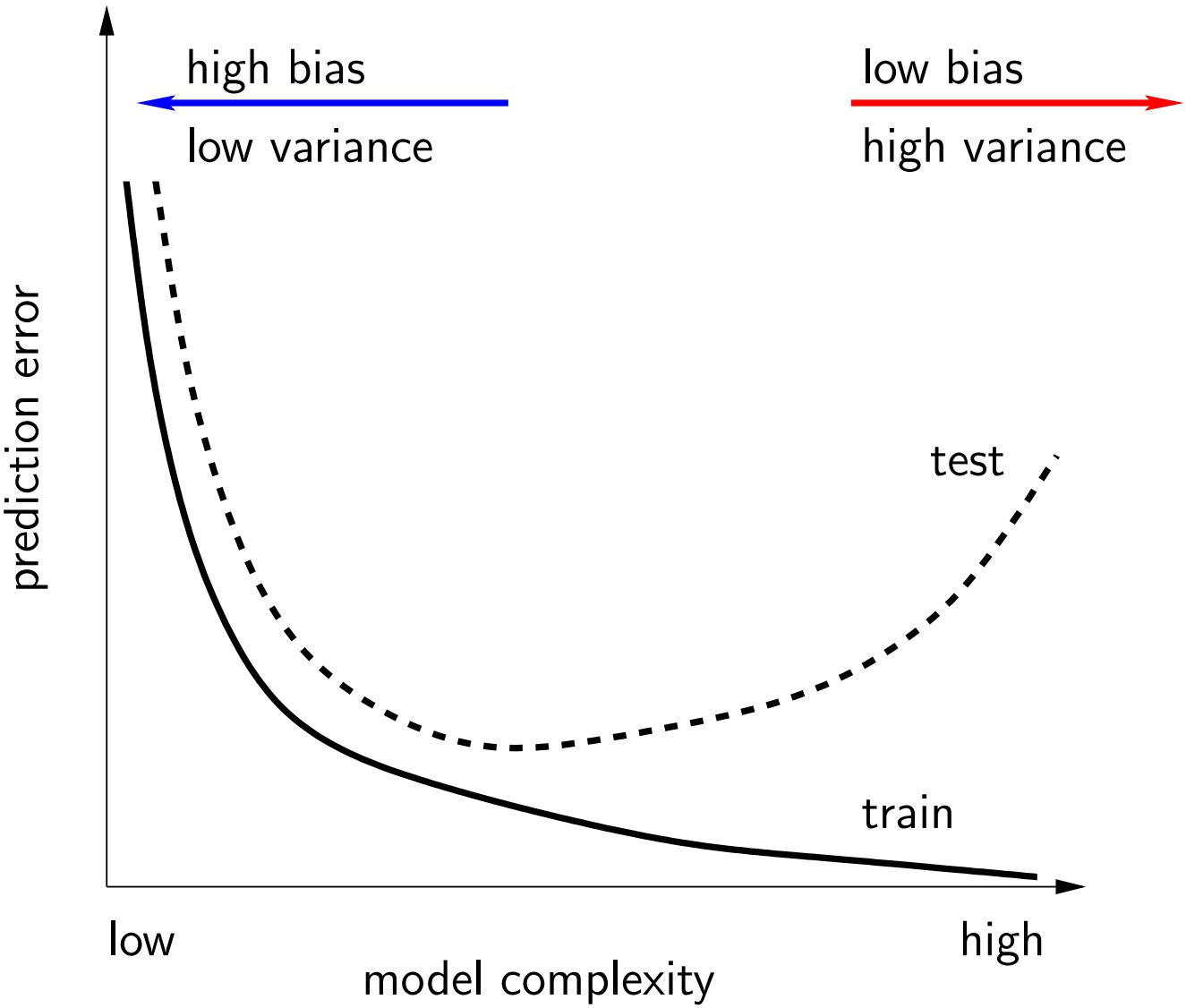
Bias-Variance Dilemma (cont.)

The bias-variance dilemma is general phenomenon and also occurs as the under/overfitting problem in neural networks. In the context of neural networks:

- Bias is a measure of how much the network output, averaged over all possible data sets differs from the desired function.
- Variance is a measure of how much the network output varies between data sets.

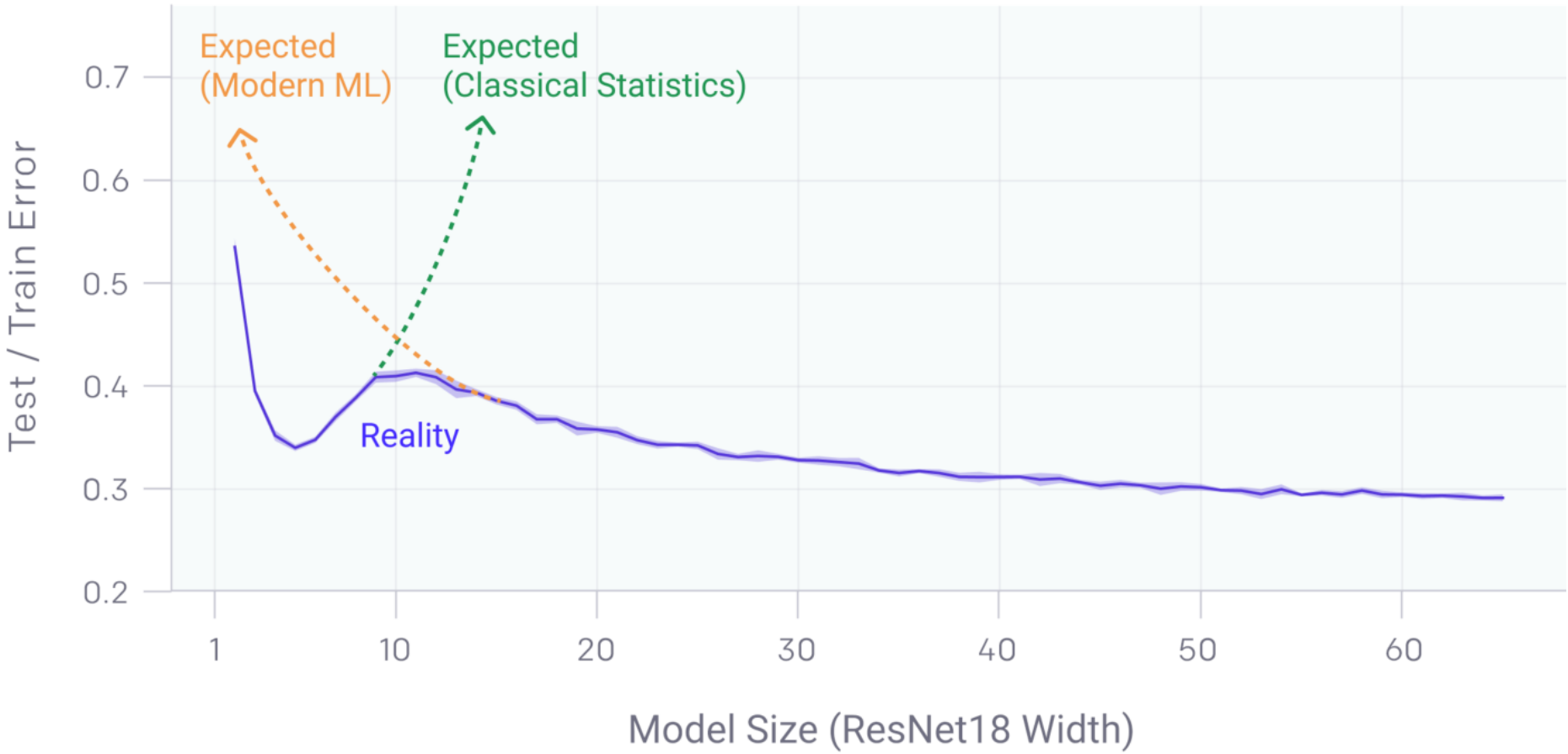
In early stage of training, the bias is large because the network is far from the desired function. If the network is trained too long, then the network will also have learned the noise specific in the data set.

Bias-Variance Dilemma (cont.)

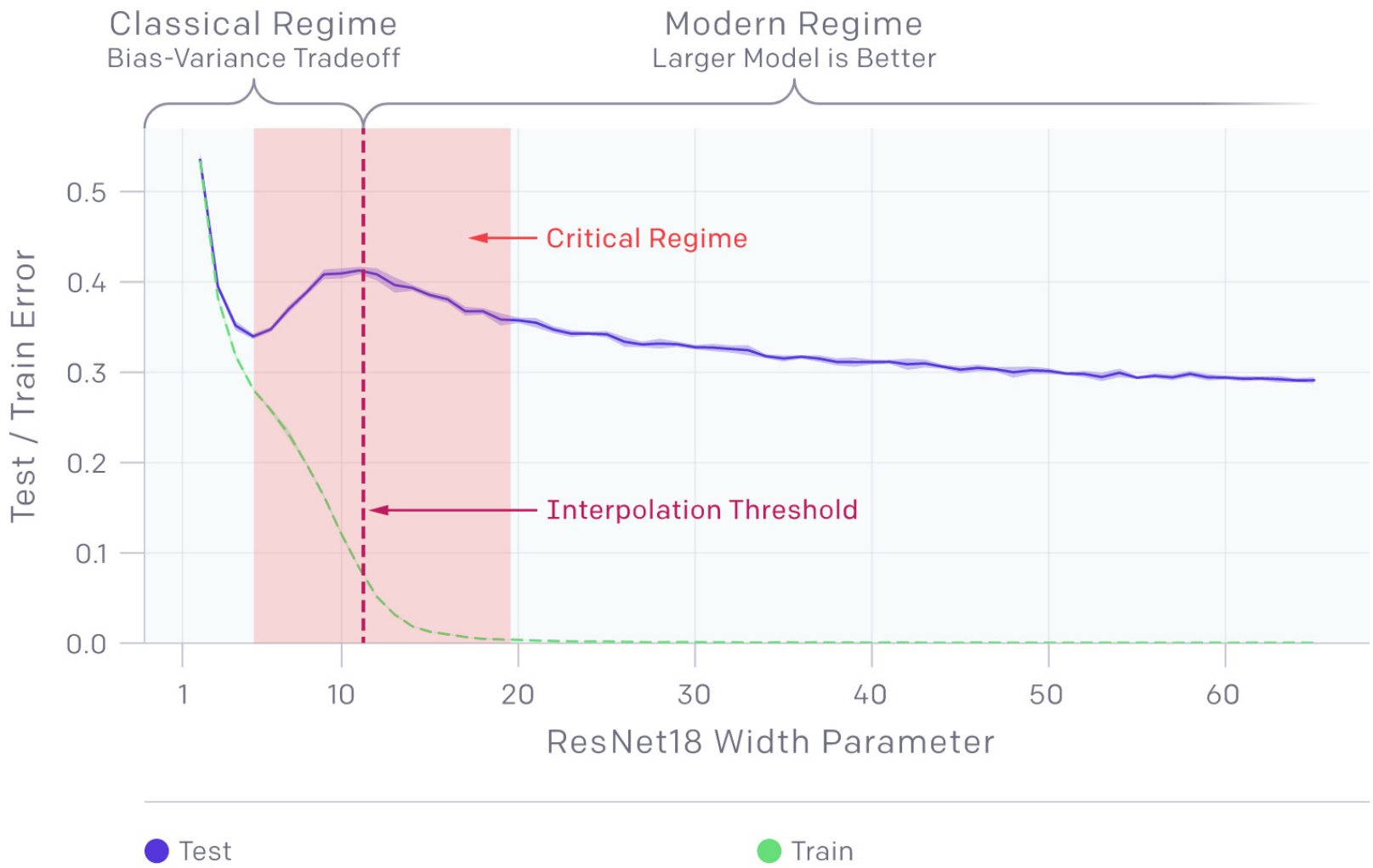


Deep Double Descent

- Conventional wisdom in classical statistics: Once we pass a certain threshold "large models are worse".
- Conventional wisdom among practitioners "larger models are better".

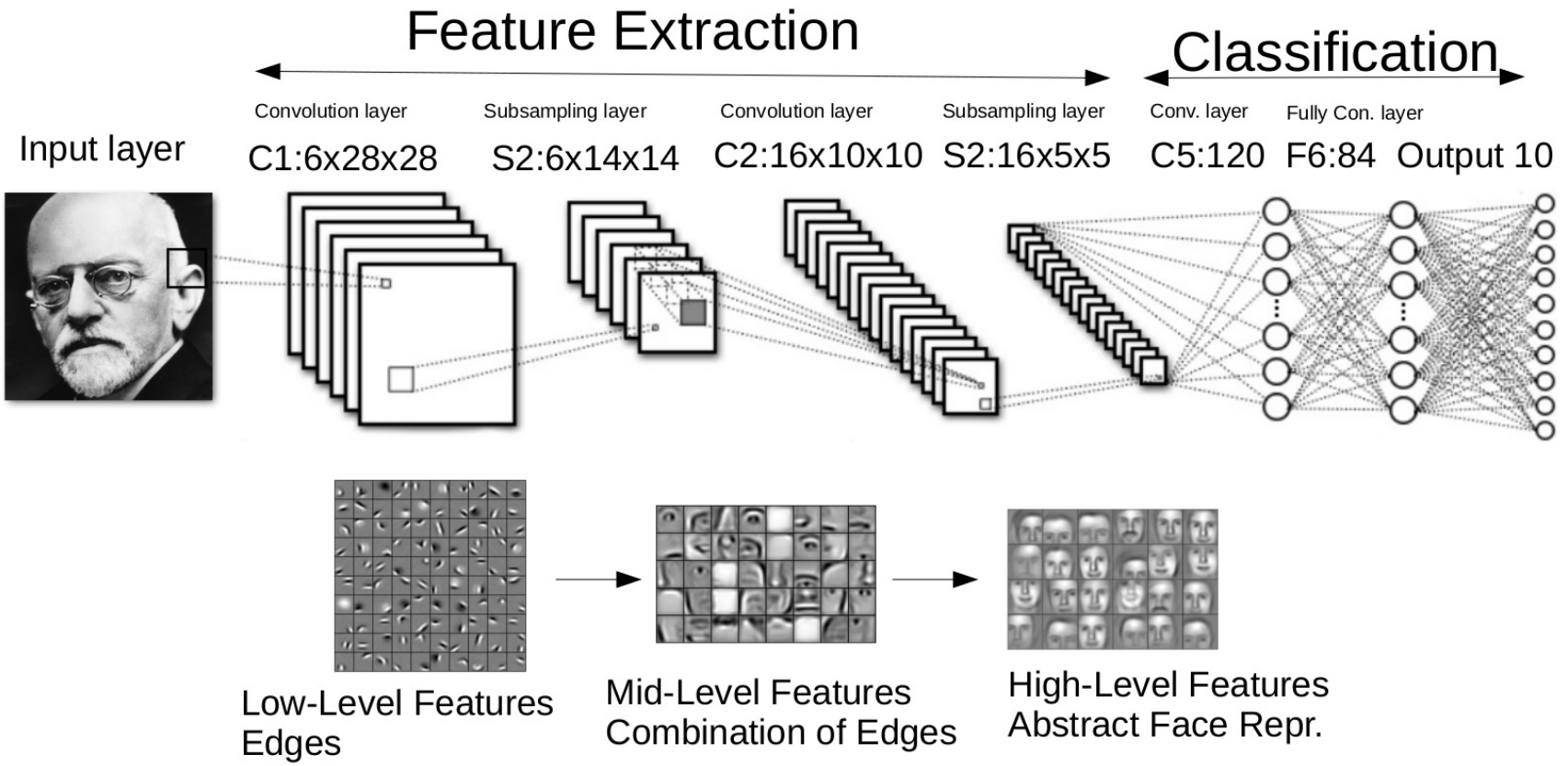


Deep Double Descent (cont.)



See Deep Double Descent: Where Bigger Models and More Data Hurt

Brief Overview of CNN



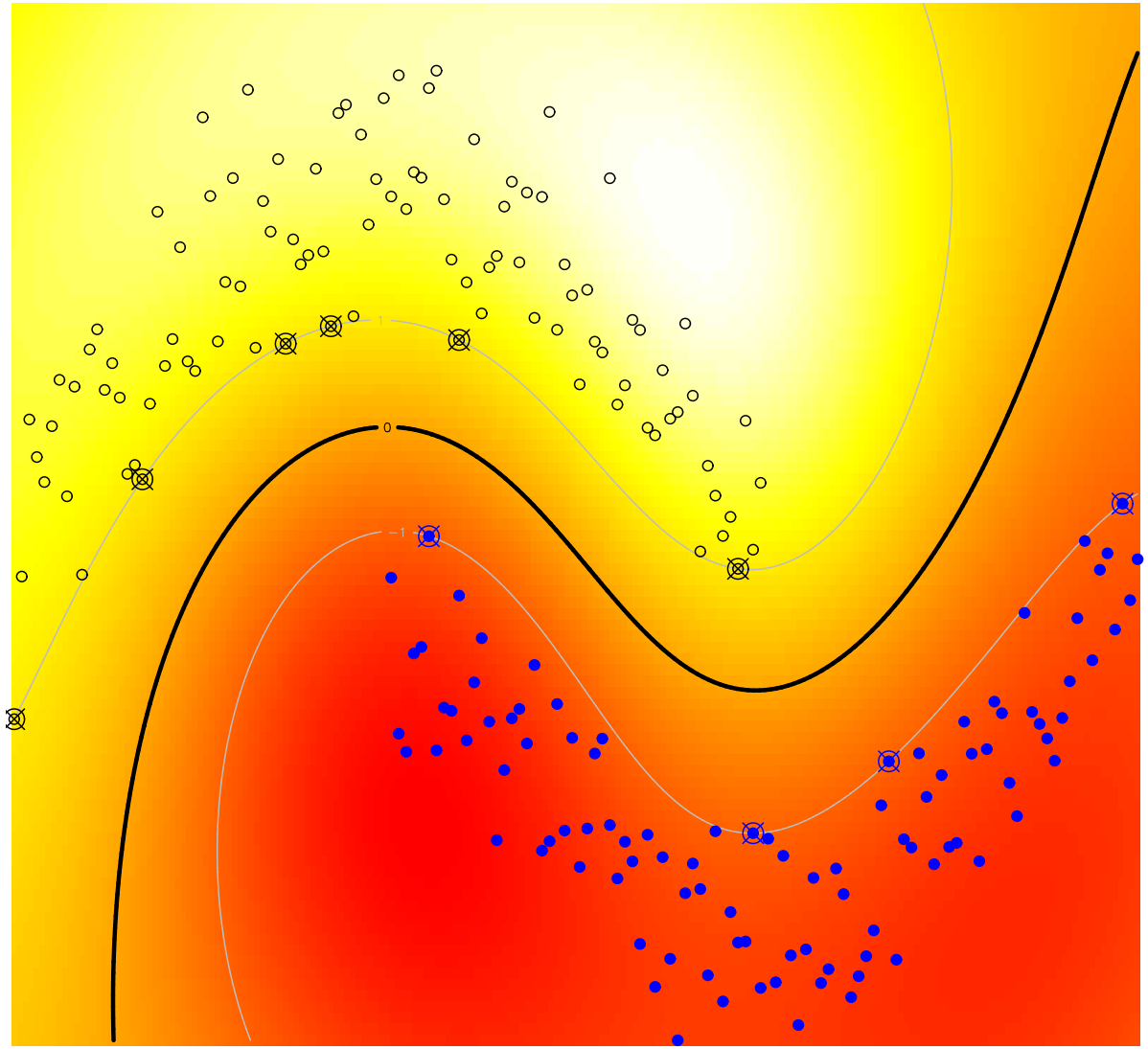
Note: This is the "famous" LeNet5 architecture originally proposed for digit recognition.
See:Visualizing and Understanding Convolutional Networks and
Video: Deep Visualization Toolbox

Recurrent Neural Networks, LSTM & Summary

Note: Our so far investigated feed-forward neural networks worked on independent and identically distributed (short i.i.d) data. For sequential data or time series data, recurrent neural network or LSTM are proper architectures.
Summary:

- Deep neural networks scale with data (traditional ML methods not) and currently need a lot of data.
- Pick proper neural network architecture (CNN, LSTM, GNN, etc...) for your problem.
- Don't give while training neural network.

Support Vector Machine (SVM)

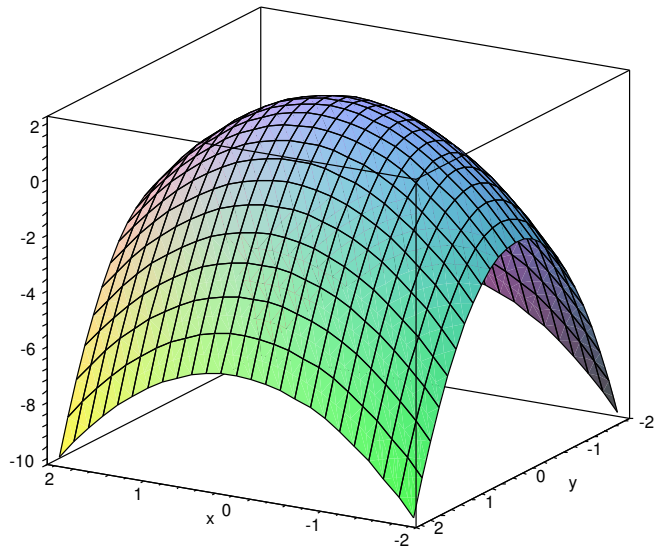


Quick Recap. on Lagrange Multipliers

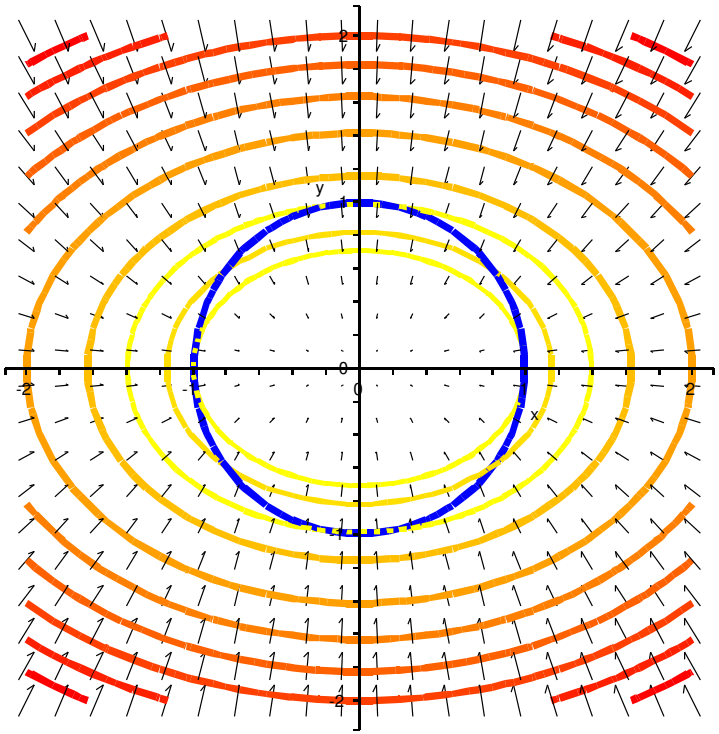
Given the following optimization problem:

$$\begin{aligned} \text{maximize} \quad & f(x,y) = 2 - x^2 - 2y^2 \\ \text{subject to} \quad & g(x,y) = x^2 + y^2 - 1 = 0. \end{aligned}$$

With *Lagrange multipliers* we can find the extrema of a function of several variables subject to one or more constraints.



Lagrange Multipliers (cont.)



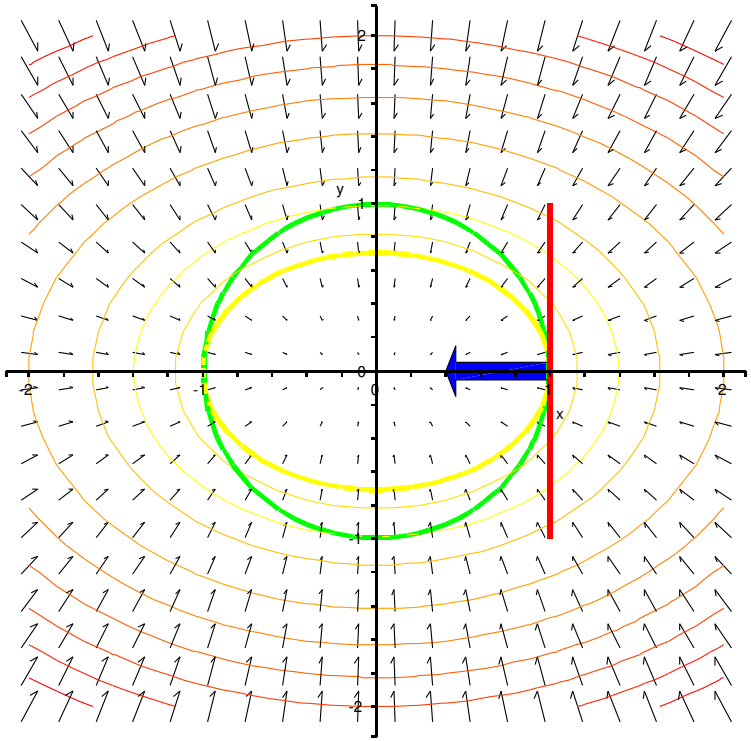
The gradient of f,

$$\begin{aligned} \nabla f &= \text{grad } f(\mathbf{x}) \\ &= \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_n} \right) \end{aligned}$$

is a vector field, where the vectors point in the directions of the greatest increase of f.

The direction of greatest increase is always perpendicular to the level curves. The circle (blue curve) is the feasible region satisfying the constraint $x^2 + y^2 - 1 = 0$

Lagrange Multipliers (cont.)



At extreme points (x, y) the gradients of f and g are parallel vectors, that is

$$\nabla f(x, y) = \lambda \nabla g(x, y)$$

To find the (x, y) we have to solve

$$\nabla f(x, y) - \lambda \nabla g(x, y) = 0$$

Lagrange Multipliers Example

Back to our optimization problem:

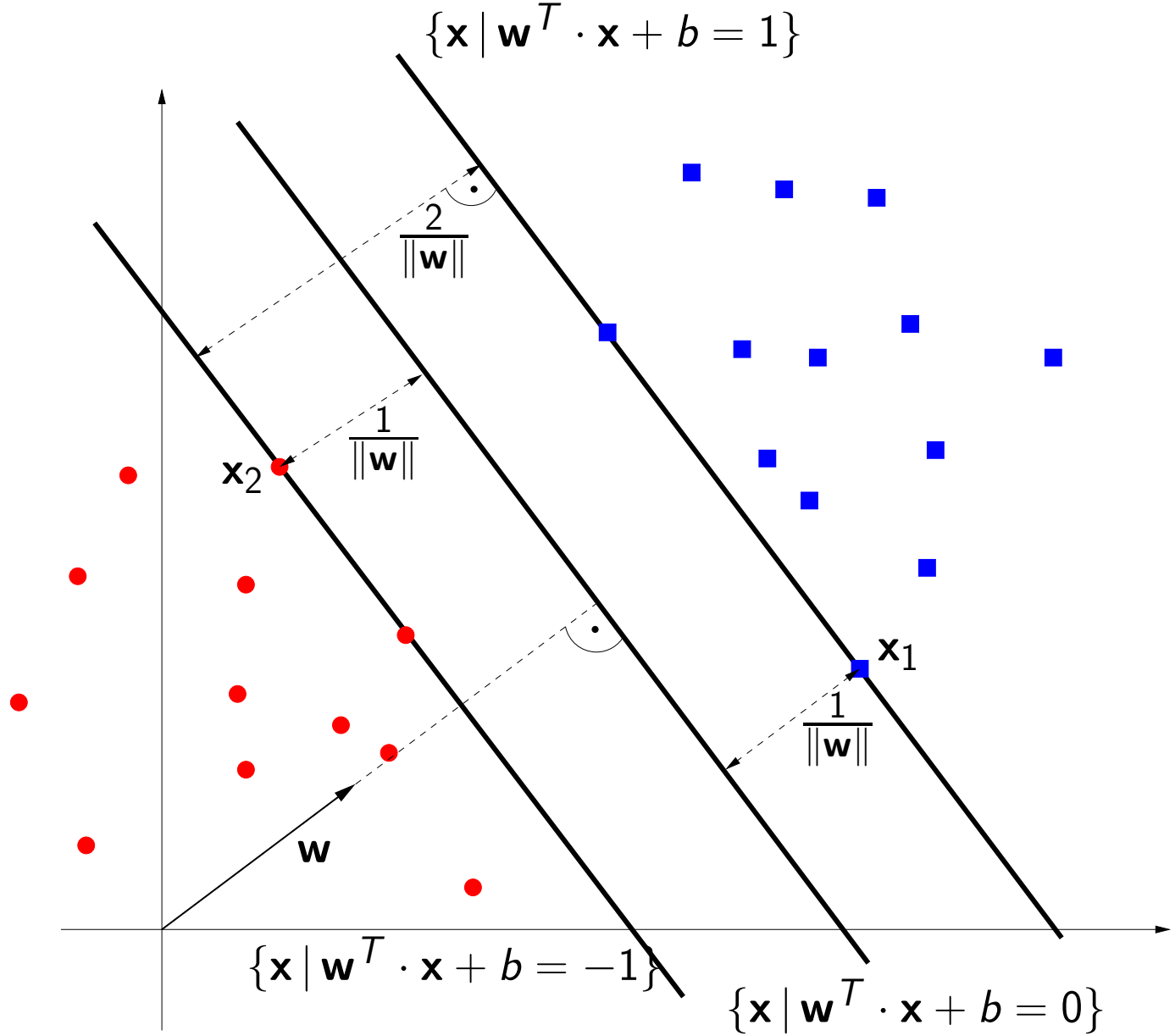
$$\begin{aligned} \text{maximize} \quad & f(x,y) = 2 - x^2 - 2y^2 \\ \text{subject to} \quad & g(x,y) = x^2 + y^2 - 1 = 0. \end{aligned}$$

$$L(x,y,\lambda) = f(x,y) - \lambda g(x,y) = 2 - x^2 - 2y^2 - \lambda(x^2 + y^2 - 1)$$

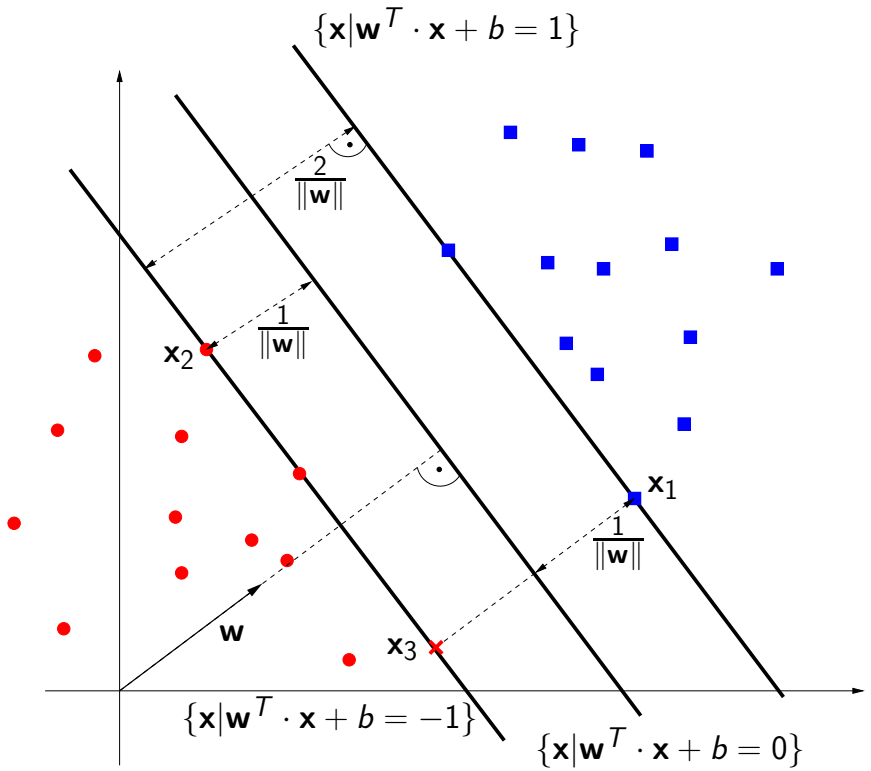
$$\begin{aligned} \frac{\partial L(x,y,\lambda)}{\partial x} &= -2x - 2\lambda x = 0 \\ \frac{\partial L(x,y,\lambda)}{\partial y} &= -4y - 2\lambda y = 0 \\ \frac{\partial L(x,y,\lambda)}{\partial \lambda} &= -x^2 - y^2 + 1 = 0 \end{aligned}$$

Solving the equation system gives: $x = \pm 1$ and $y = 0$ $(\lambda = -1)$ and $x = 0$ and $y = \pm 1$ $(\lambda = -2)$.

Plus/Minus and Zero Hyperplane



Optimal Separating Hyperplane



Let $\mathbf{x}_3$ be any point on the "minus" hyperplane and let $\mathbf{x}_1$ be the closest point to $\mathbf{x}_3$.

$$\mathbf{w}^T \cdot \mathbf{x}_1 + b = +1$$
$$\mathbf{w}^T \cdot \mathbf{x}_3 + b = -1$$
$$\mathbf{x}_1 = \mathbf{x}_3 + \lambda \mathbf{w}$$
$$\mathbf{w}^T \cdot \underbrace{(\mathbf{x}_3 + \lambda \mathbf{w})}_{\mathbf{x}_1} + b = 1$$

$\underbrace{\mathbf{w}^T \cdot \mathbf{x}_3 + b}_{-1} + \lambda \|\mathbf{w}\|^2 = 1 \Rightarrow \lambda = \frac{2}{\|\mathbf{w}\|^2}$. We are interested in margin, i.e.

$$\|\mathbf{x}_1 - \mathbf{x}_3\| = \|\lambda \mathbf{w}\| = \lambda \|\mathbf{w}\| = \frac{2}{\|\mathbf{w}\|^2} \|\mathbf{w}\| = \frac{2}{\|\mathbf{w}\|}$$

Formulation as an Optimization Problem

Hyperplane with maximum margin (the smaller the norm of the weight vector $\mathbf{w}$, the larger the margin):

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\|\mathbf{w}\|^2 \\ \text{subject to} \quad & y_i\left(\langle \mathbf{w}, \mathbf{x}_i \rangle + b\right) \geq 1, \quad i = 1, \ldots, m \end{aligned}$$

Introduce Lagrange multipliers $\alpha_i \geq 0$ and a Lagrangian

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{m} \alpha_i (y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) - 1)$$

At the extrema, we have

$$\frac{\partial}{\partial b} L(\mathbf{w}, b, \boldsymbol{\alpha}) = 0, \quad \frac{\partial}{\partial \mathbf{w}} L(\mathbf{w}, b, \boldsymbol{\alpha}) = 0$$

Optimization and Kuhn-Tucker Theorem

leads to solution

$$\sum_{i=1}^{m} \alpha_i y_i = 0, \qquad \mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i.$$

The extreme point solution obtained has an important property that results from optimization known as the Kuhn-Tucker theorem. The theorem says:

> Lagrange multiplier can be non-zero only if the corresponding inequality constraint is an equality at the solution.

This implies that *only* the training examples $\mathbf{x}_i$ that lie on the plus and minus hyperplane have their corresponding α_i non-zero.

Relevant/Irrelevant Support Vector

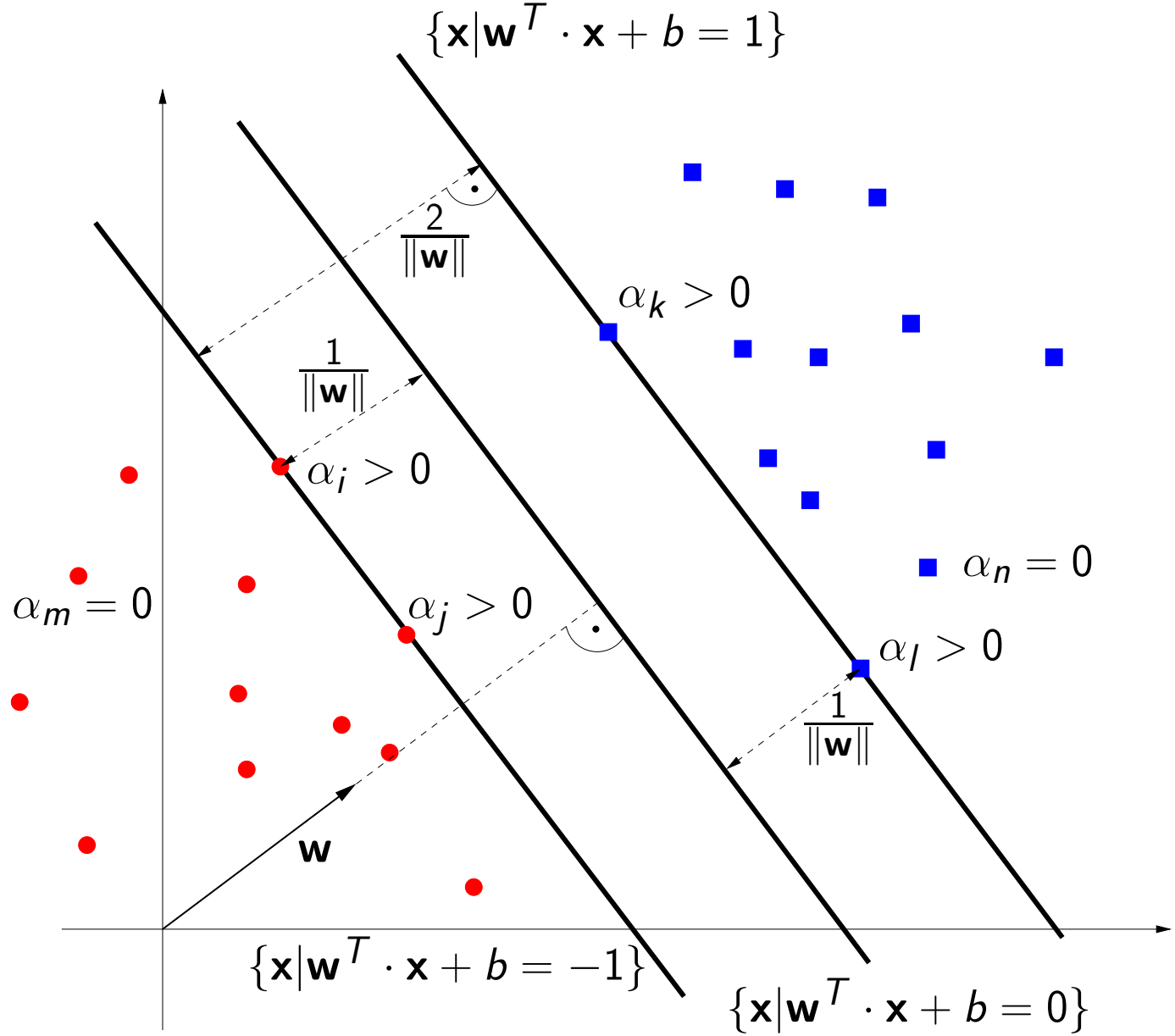
More formally, the Karush-Kuhn-Tucker complementarity conditions say:

$$\alpha_i \left[y_i (\langle \mathbf{w}, \mathbf{x}_i \rangle + b) - 1 \right] = 0, \quad i = 1, \ldots, m$$

the Support Vectors lie on the margin. That means for all training points

$$\begin{aligned} [y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b)] > 1 \quad & \Rightarrow \alpha_i = 0 \quad && \rightarrow \mathbf{x}_i \text{ irrelevant} \\ [y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b)] = 1 \quad & \text{(on margin)} \quad && \rightarrow \mathbf{x}_i \text{ Support Vector} \end{aligned}$$

Relevant/Irrelevant Support Vector



Dual Form

The dual form has many advantages

- Formulate optimization problem without **w** (mapping **w** in high-dimensional spaces).
- Formulate optimization problem by means of α, y_i and dot product $\langle \mathbf{x}_i, \mathbf{x}_j \rangle$.
- Quadratic Programming Solver.

$$\text{maximize} \qquad W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle$$

$$\text{subject to} \qquad \alpha_i \geq 0, \quad i = 1, \ldots, m \text{ and } \sum_{i=1}^{m} \alpha_i y_i = 0$$

Hyperplane Decision Function

The solution is determined by the examples on the margin. Thus

$$\begin{aligned} f(\mathbf{x}) &= \operatorname{sgn}\left(\langle \mathbf{w}, \mathbf{x} \rangle + b\right) \\ &= \operatorname{sgn}\left(\sum_{i=1}^{m} y_i \alpha_i \langle \mathbf{x}, \mathbf{x}_i \rangle + b\right) \end{aligned}$$

where

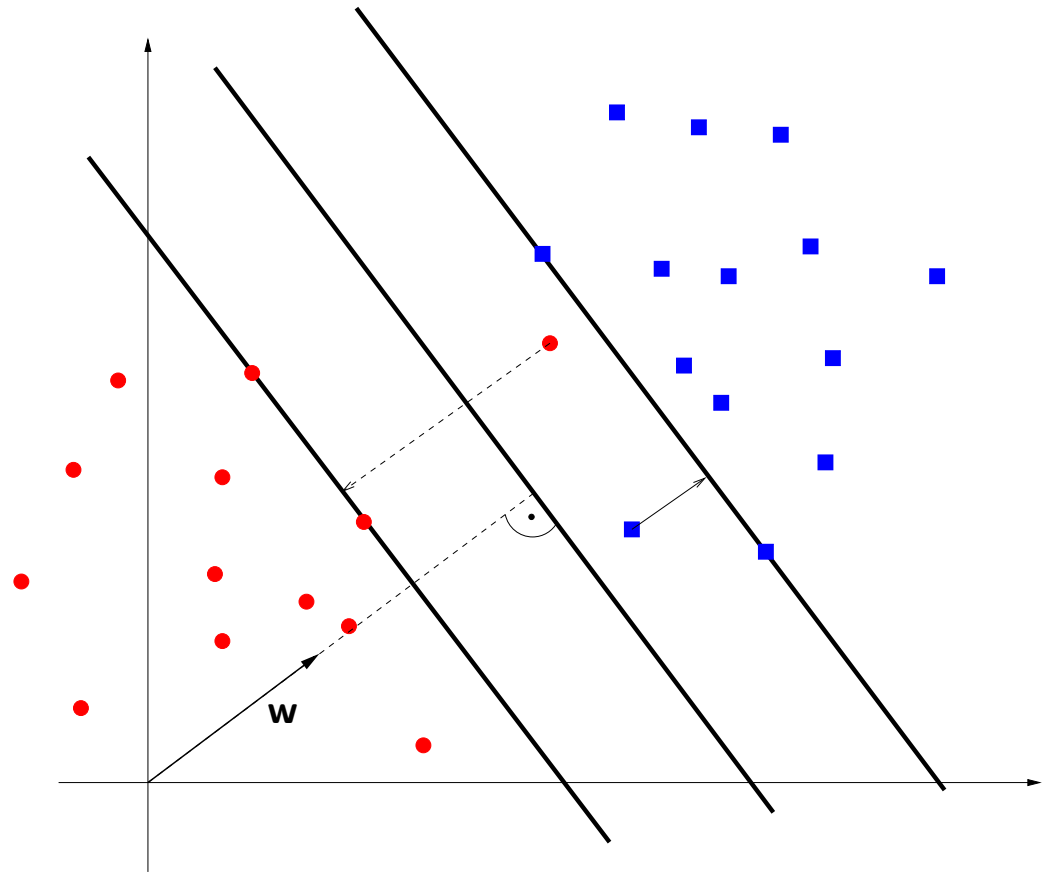
$$\alpha_i \left[y_i (\langle \mathbf{w}, \mathbf{x}_i \rangle + b) - 1 \right] = 0, \quad i = 1, \ldots, m$$

and

$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i$$

and choose i such that $\alpha_i > 0$ and $b = y_i - \sum_{j=1}^{m} \alpha_j y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle$

Non-separable Case



Case where the constraints $y_i \langle \mathbf{w}, \mathbf{x}_i \rangle + b \geq 1$ cannot be satisfied, i.e. $\alpha_i \to \infty$

Relax Constraints (Soft Margin)

Modify the constraints to

$$y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 - \xi_i, \text{ with } \xi_i \geq 0$$

and add

$$C \cdot \sum_{i=1}^{m} \xi_i$$

i.e. distance of error points to their correct place in the objective function

$$\text{minimize} \qquad \frac{1}{2}\|\mathbf{w}\|^2 + C \cdot \sum_{i=1}^{m} \xi_i$$

Same dual, with additional constraints $0 \leq \alpha_i \leq C$