Stat 577 "Statistical Learning Theory"

T5: Support vector machine, Neural networks

Xiongzhi Chen

Washington State University

Overview

- Training set $\mathcal{T}$: N observations $\{(x_i, y_i)\}_{i=1}^{N}$, each $x_i \in \mathbb{R}^p$ with class label $y_i \in \{-1, 1\}$
- Target: given a new $x \in \mathbb{R}^p$, estimate its associated class label y
- A "support vector machine (SVM)" employs a "hyperplane" to classify observations into their classes

We will discuss two scenarios for a SVM:

1. The two classes can be separated by a hyperplane
2. The two classes overlap and cannot be separated by a hyperplane

SVM under separability

Assume the two classes are separable, and define the hyperplane

$$\Pi = \left\{ x : x^T \beta + \beta_0 = 0 \right\} \text{ with } \|\beta\| = 1$$

- Π induces the classification rule $G(x) = \text{sgn}\left(x^T \beta + \beta_0\right)$, where sgn is the sign function
- Find the hyperplane with the biggest margin (if existent), i.e., solve for β and β_0 in

$$\max_{\beta_0, \beta, \|\beta\|=1} M \quad \text{subject to} \quad \min_{1 \le i \le N} y_i \left(x_i^T \beta + \beta_0 \right) \ge M \tag{1}$$

or equivalently

$$\min_{\beta, \beta_0} 2^{-1} \|\beta\|^2 \quad \text{subject to} \quad \min_{1 \le i \le N} y_i \left(x_i^T \beta + \beta_0 \right) \ge 1 \tag{2}$$

Proof of the equivalence

Proof of the equivalence between (1) and (2):

- Recall (1) as $\max_{\beta_0,\beta,\|\beta\|=1} M$ subject to

$$\min_{1\le i\le N} y_i\left(x_i^T\beta+\beta_0\right)\ge M$$

- Plug in (1) the mapping $\beta\mapsto\beta'=\beta/M$, $\beta_0\mapsto\beta_0'=\beta_0/M$ gives

$$\max_{\beta_0',\beta',\|\beta'\|=M^{-1}} M=\min_{\beta',\beta_0'}\left\|\beta'\right\|$$

  subject to

$$\min_{1\le i\le N} y_i\left(x_i^T\beta'+\beta_0'\right)\ge 1$$

- However, $\min_{\beta,\beta_0}\|\beta\|=\min_{\beta,\beta_0} c_0\|\beta\|^2$ for any constant $c_0>0$. So, (1) and (2) are equivalent.

SVM: illustration from Chapter 12 of Text

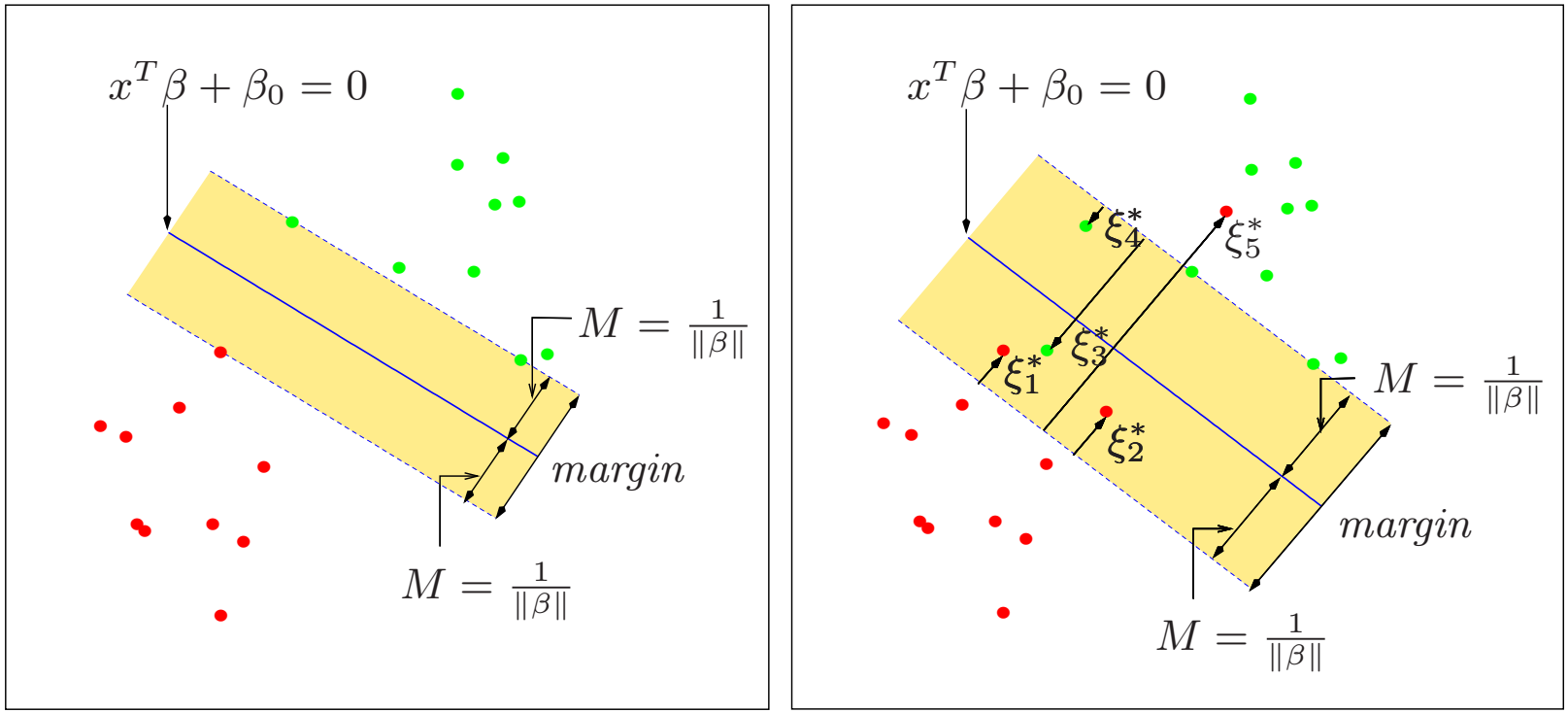


FIGURE 12.1. *Support vector classifiers. The left panel shows the separable case. The decision boundary is the solid line, while broken lines bound the shaded maximal margin of width* $2M = 2/\|\beta\|$. *The right panel shows the nonseparable (overlap) case. The points labeled* ξ_j^* *are on the wrong side of their margin by an amount* $\xi_j^* = M\xi_j$; *points on the correct side have* $\xi_j^* = 0$. *The margin is maximized subject to a total budget* $\sum \xi_i \leq$ *constant. Hence* $\sum \xi_j^*$ *is the total distance of points on the wrong side of their margin.*

SVM: general formulation

Assume the two classes overlap in feature space and are nonseparable. Then we have the general formulation

$$\max_{\beta_0,\beta,\|\beta\|=1} M \quad \text{subject to} \quad \begin{cases} y_i\left(x_i^T\beta+\beta_0\right) \geq M\left(1-\xi_i\right) \\ \xi_i \geq 0, \sum \xi_i \leq C, \forall i \end{cases} \tag{3}$$

where ξ_i is the "proportional amount" by which the prediction (i.e., the predicted class label) is on the wrong side of its margin, C is a constant and referred to as the "cost parameter", and $\{\xi_i\}_{i=1}^N$ are slack variables. The formulation (3) is equivalent to

$$\min \|\beta\| \quad \text{subject to} \quad \begin{cases} y_i\left(x_i^T\beta+\beta_0\right) \geq 1-\xi_i, \forall i \\ \xi_i \geq 0, \sum \xi_i \leq C \end{cases}$$

SVM: general formulation (recap)

Assume the two classes overlap in feature space. We implement

$$\min \|\beta\| \text{ subject to } \begin{cases} y_i \left(x_i^T \beta + \beta_0\right) \geq 1 - \xi_i, \forall i \\ \xi_i \geq 0, \sum \xi_i \leq C \end{cases}$$

- Bounding $\sum \xi_i$ restricts the total proportional amount of wrong predictions
- For SVM in this formulation, points well inside their class boundary do not play a significant role in shaping the boundary. In LDA, the decision boundary is determined by the covariance of the class distributions and the positions of the class centroids.
- The above formulation is equivalent to

$$\min 2^{-1} \|\beta\|^2 \text{ subject to } \begin{cases} y_i \left(x_i^T \beta + \beta_0\right) \geq 1 - \xi_i, \forall i \\ \xi_i \geq 0, \sum \xi_i \leq C \end{cases}$$

Solving the general formulation

Solving

$$\min 2^{-1} \|\beta\|^2 \quad \text{subject to} \quad \begin{cases} y_i \left(x_i^T \beta + \beta_0\right) \geq 1 - \xi_i, \forall i \\ \xi_i \geq 0, \sum \xi_i \leq C \end{cases}$$

via Lagrange multipliers

$$L = \frac{1}{2} \|\beta\|^2 + C \sum \xi_i - \sum \alpha_i \left[y_i \left(x_i^T \beta + \beta_0 \right) - (1 - \xi_i) \right] - \sum \mu_i \xi_i$$

gives the first order condition

$$\hat{\beta} = \sum_{i=1}^{N} \hat{\alpha}_i y_i x_i; \quad \sum \hat{\alpha}_i y_i = 0; \quad \hat{\alpha}_i = C - \hat{\mu}_i \tag{4}$$

as well as the positivity constraints $\hat{\alpha}_i, \hat{\mu}_i, \hat{\xi}_i \geq 0$.

The optimizer

The "Karush–Kuhn–Tucker (KKT)" condition for the dual problem requires

$$\begin{cases} \alpha_i \left[y_i \left(x_i^T \beta + \beta_0 \right) - (1 - \xi_i) \right] = 0 \\ \mu_i \xi_i = 0 \\ y_i \left(x_i^T \beta + \beta_0 \right) \geq 1 - \xi_i \end{cases} \tag{5}$$

In (4), $\hat{\alpha}_i \neq 0$ for observations i for which the constraints

$$y_i \left(x_i^T \beta + \beta_0 \right) \geq 1 - \xi_i$$

are met. Since $\hat{\beta}$ is represented only by such observations, they are called "support vectors". The decision function is

$$\hat{G}(x) = \operatorname{sgn} \left(x^T \hat{\beta} + \hat{\beta}_0 \right)$$

A Summary under general formulation

The problem

$$\min_{\beta_0,\beta_1} \frac{1}{2}\|\beta\|^2 + C\sum_{i=1}^{N}\xi_i$$

subject to

$$\xi_i \geq 0, y_i\left(x_i^T\beta + \beta_0\right) \geq 1 - \xi_i, \forall i$$

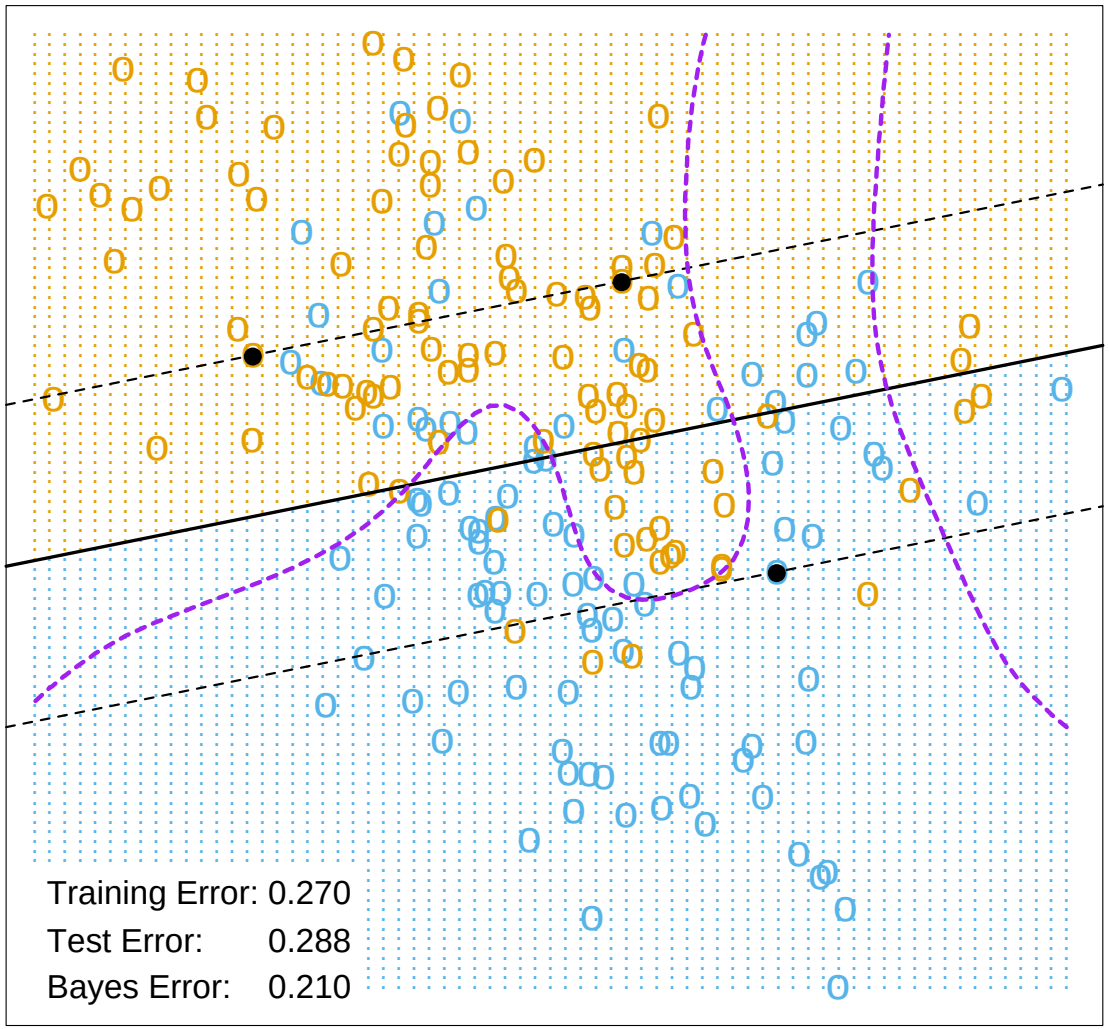
is solved via minimizing

$$L = \frac{1}{2}\|\beta\|^2 + C\sum\xi_i - \sum\alpha_i\left[y_i\left(x_i^T\beta + \beta_0\right) - (1-\xi_i)\right] - \sum\mu_i\xi_i$$

to obtain the "support vectors" and decision function

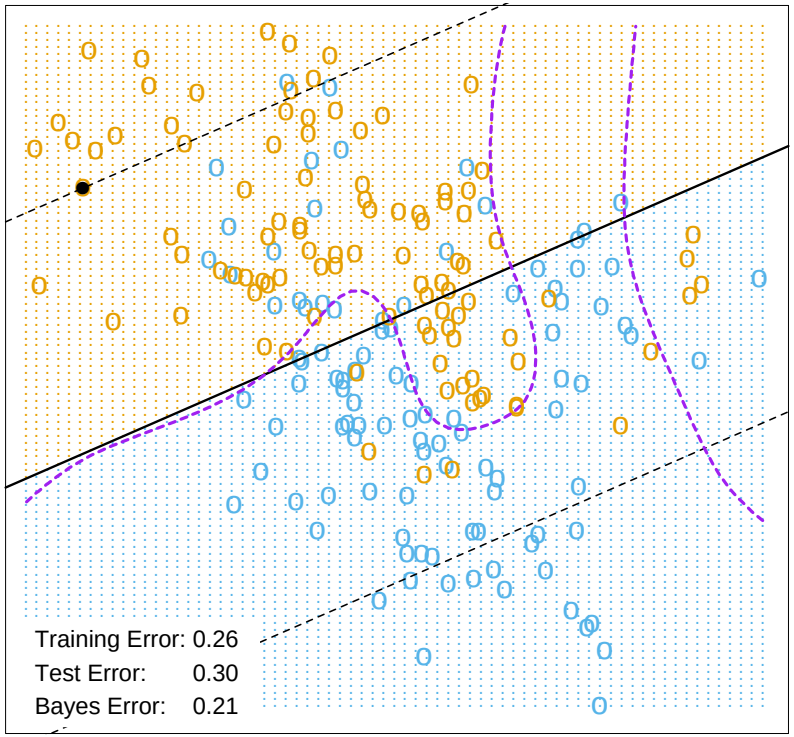
$$\hat{G}(x) = \text{sgn}\left(x^T\hat{\beta} + \hat{\beta}_0\right)$$

SVM: illustration (Figure 12.2 of Chapter 12 of Text)



$C = 10000$

SVM: illustration (Figure 12.2 of Chapter 12 of Text)



$C = 0.01$

FIGURE 12.2. *The linear support vector boundary for the mixture data example with two overlapping classes, for two different values of C. The broken lines indicate the margins, where $f(x) = \pm 1$. The support points ($\alpha_i > 0$) are all the points on the wrong side of their margin. The black solid dots are those support points falling exactly on the margin ($\xi_i = 0$, $\alpha_i > 0$). In the upper panel 62% of the observations are support points, while in the lower panel 85% are. The broken purple curve in the background is the Bayes decision boundary.*

Remarks

- In the illustration, the SVM is not very sensitive to the choices of the cost parameter C because the decision boundary of the SVM is a linear function of the feature variable.
- The parameter C can be determined by cross-validation
- There various extensions of SVM whose decision boundaries are not necessarily hyperplanes

SVM via the "hinge loss"

Assume the intended boundary is a hyperplane

- The optimization problem for SVM is

$$\min_{\beta_0,\beta_1} \frac{1}{2}\|\beta\|^2 + C\sum_{i=1}^{N}\xi_i \text{ subject to } \begin{cases} y_i\left(x_i^T\beta+\beta_0\right) \geq 1-\xi_i, \forall i \\ \xi_i \geq 0 \end{cases}$$

- It is equivalent to

$$\min_{\beta_0,\beta_1} \sum_{i=1}^{N} \max\left\{0, 1-y_i\left(x^T\beta+\beta_0\right)\right\} + \frac{\lambda}{2}\|\beta\|^2$$

with $\lambda = 1/C$, where $\max\{0,x\}$ is called the "hinge loss'

The optimization

Recall the problem

$$\min_{\beta_0,\beta_1} \frac{1}{2}\|\beta\| + C\sum_{i=1}^{N}\xi_i \text{ subject to } \begin{cases} y_i\left(x_i^T\beta+\beta_0\right) \geq 1-\xi_i, \forall i \\ \xi_i \geq 0 \end{cases}$$

is solved via minimizing

$$L = \frac{1}{2}\|\beta\|^2 + C\sum\xi_i - \sum\alpha_i\left[y_i\left(x_i^T\beta+\beta_0\right)-(1-\xi_i)\right] - \sum\mu_i\xi_i$$

to obtain the "support vectors" and decision function

$$\hat{G}(x) = \text{sgn}\left(x^T\hat{\beta}+\hat{\beta}_0\right)$$

The optimization

Setting the gradient $\nabla_{(\beta_0,\beta,\xi)} L = 0$ for

$$
\begin{aligned}
& L\left(x;\beta_0,\beta,\xi\right) \\
&= \frac{1}{2}\left\|\beta\right\|^2 + C\sum \xi_i - \sum \alpha_i \left[y_i\left(x_i^T\beta+\beta_0\right) - \left(1-\xi_i\right)\right] - \sum \mu_i\xi_i
\end{aligned}
$$

gives critical points

$$
\beta = \sum \alpha_i y_i x_i^T; 0 = \sum a_i y_i; \alpha_i = C - \mu_i, \forall i
$$

that (locally) minimizes L and evaluates L into its dual

$$
L_D = \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_i \sum_{i'} \alpha_i \alpha_{i'} y_i y_{i'} x_i^T x_{i'}
$$

The optimization

Maximizing

$$L_D = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_i \sum_{i'} \alpha_i \alpha_{i'} y_i y_{i'} x_i^T x_{i'}$$

subject to

$$0 = \sum a_i y_i \text{ and } 0 \leq \alpha_i \leq C, \forall i$$

via KKT condition

$$\begin{cases} \alpha_i \left[y_i \left(x_i^T \beta + \beta_0 \right) - (1 - \xi_i) \right] = 0 \\ \mu_i \xi_i = 0 \\ y_i \left(x_i^T \beta + \beta_0 \right) \geq 1 - \xi_i \end{cases}$$

Note: the first 2 are complementary slackness conditions.

Overview and settings for neural networks

Neural networks (NNs) are a class of nonlinear statistical models and are applicable to both regression and classification. We will only talk about the "vanilla" NN. A vanilla NN for K-class classification can described as below:

- There are K units at the top, with the kth unit modeling the probability of class k. There are K target measurements $\{Y_k\}_{k=1}^K$ such that each $Y_k \in \{0,1\}$ for the kth class
- Derived features Z_m and Y_k are defined by

$$Z_m = \sigma\left(\alpha_{0m} + \alpha_m^T X\right), m = 1, ..., M$$
$$T_k = \beta_{0k} + \beta_k^T Z, k = 1, ..., K$$
$$f_k(X) = g_k(T), k = 1, ..., K$$

where $X \in \mathbb{R}^p$, $Z = (Z_1, ..., Z_M) \in \mathbb{R}^M$ and $T = (T_1, ..., T_K) \in \mathbb{R}^K$

Overview and settings for neural networks

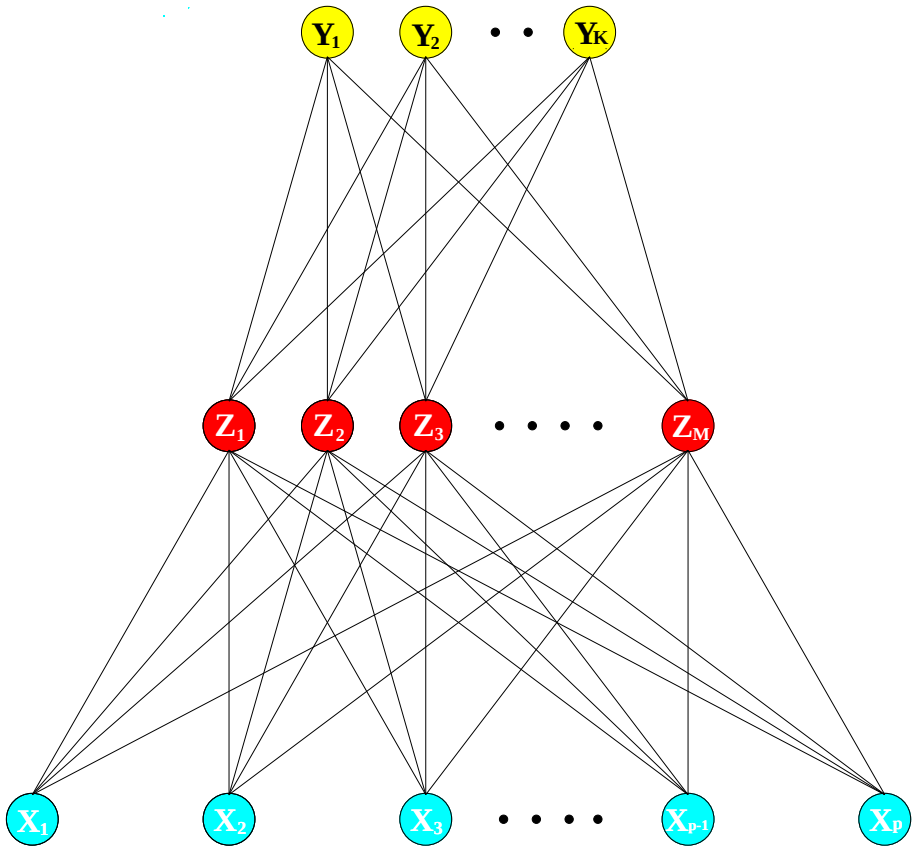


FIGURE 11.2. *Schematic of a single hidden layer, feed-forward neural network.*

Overview and settings for neural networks

- σ, referred to as the "activation function", is chosen to be the "sigmoid" function $\sigma(v) = 1/(1+e^{-v})$ for K-class classification
- For a regression task, $g_k(T) = T_k$ is set usually, whereas for a classification task, $g_k(T) = e^{T_k}\left(\sum_{l=1}^{K} e^{T_l}\right)^{-1}$
- The units Z_m are called "hidden units" because the values of Z_m are not directly observable. However, there can be more than one hidden layer in an NN

Fitting neural networks

The unknown parameters of an NN are called "weights". The complete set of weights is denoted by θ, which consists of

$$\{\alpha_{0m}, \alpha_m; m = 1, ..., M\} \ M(p+1) \text{ weights}$$
$$\{\beta_{0k}, \beta_k; k = 1, ..., K\} \quad K(M+1) \text{ weights}$$

and we seek values of θ that make the model fit the data well based on a criterion.
Given N observations $\{x_i\}_{i=1}^N$ for X and N observations $\{y_{ik}\}_{i=1}^N$ for each Y_k,

- For regression, a commonly used criterion is

$$R(\theta) = \sum_{k=1}^{K} \sum_{i=1}^{N} (y_{ik} - f_k(x_i))$$

Fitting neural networks

- For classification, a commonly used criterion is

$$R(\theta) = -\sum_{k=1}^{K}\sum_{i=1}^{N} y_{ik} \log f_k(x_i)$$

  and the corresponding classifier is $G(x) = \text{argmax}_k f_k(x)$ for $x \in \mathbb{R}^p$

Minimizing either criterion, usually done via gradient descent and if a minimizer $\hat{\theta}$ exists, gives a choice of θ. However, an NN is often over-parametrized, and we do not seek for a global minimizer $\hat{\theta}$ of $R(\theta)$ to potentially avoid overfitting

Some issues with training neural networks

- Starting values for weights: usually these values are chosen to be random values near zero
- Overfitting: since an NN is over-parametrized, seeking a global minimum of $R(\theta)$ often leads to an overfitted model. So, regularization on the weights θ is recommended to mitigate overfitting, and/or a validation dataset is used to check for overfitting.
- Number of hidden units and layers: this directly relates to the number of weights and the level of complexity of an NN
- Multiple minima: $R(\theta)$ is often nonconvex and posses many local minima. So, a minimizer $\hat{\theta}$ may heavily depend on the initial values for the weights. One recommendation is to try a number of random starting configurations for the NN.