Review V2

Tianhang Zheng
https://tianzheng4.github.io

Adjusted R2

$$\text{Adjusted } R^2 = 1 - \frac{\text{RSS}/(n-d-1)}{\text{TSS}/(n-1)}$$

Maximizing the adjusted R^2 is equivalent to minimizing $\frac{RSS}{n-p-1}$

Ridge regression

$$\mathrm{RSS} = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2$$

Ridge regression objective

$$\mathrm{RSS} + \lambda \sum_{j=1}^{p} \beta_j^2$$

λ is a hyperparameter

Lasso

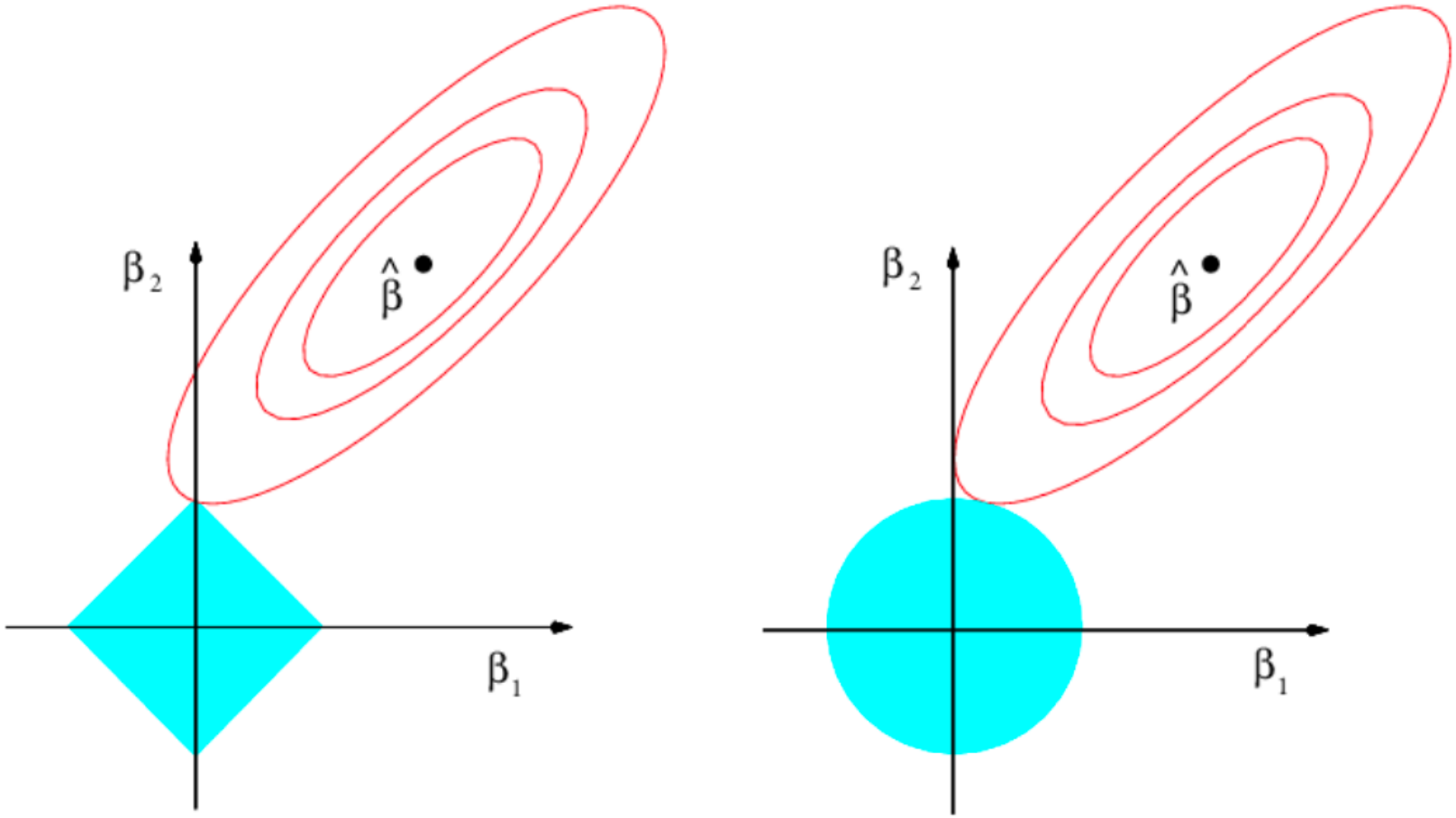
Ridge regression will include all p predictors in the final model
(Disadvantage: No predictor selection)

Objective of Lasso

$$\mathrm{RSS} + \lambda \sum_{j=1}^{p} |\beta_j|$$

If λ is sufficiently large, then Lasso will force some β to exactly zero (equivalent to predictor selection)

Lasso vs Ridge Regression



Why Lasso can force some β to exactly zero?

Principal Components Regression

Dimension reduction by Principal Components Analysis (PCA), and conduct linear regression on new predictors

The first principal component is that (normalized) linear combination of the variables with the largest variance.

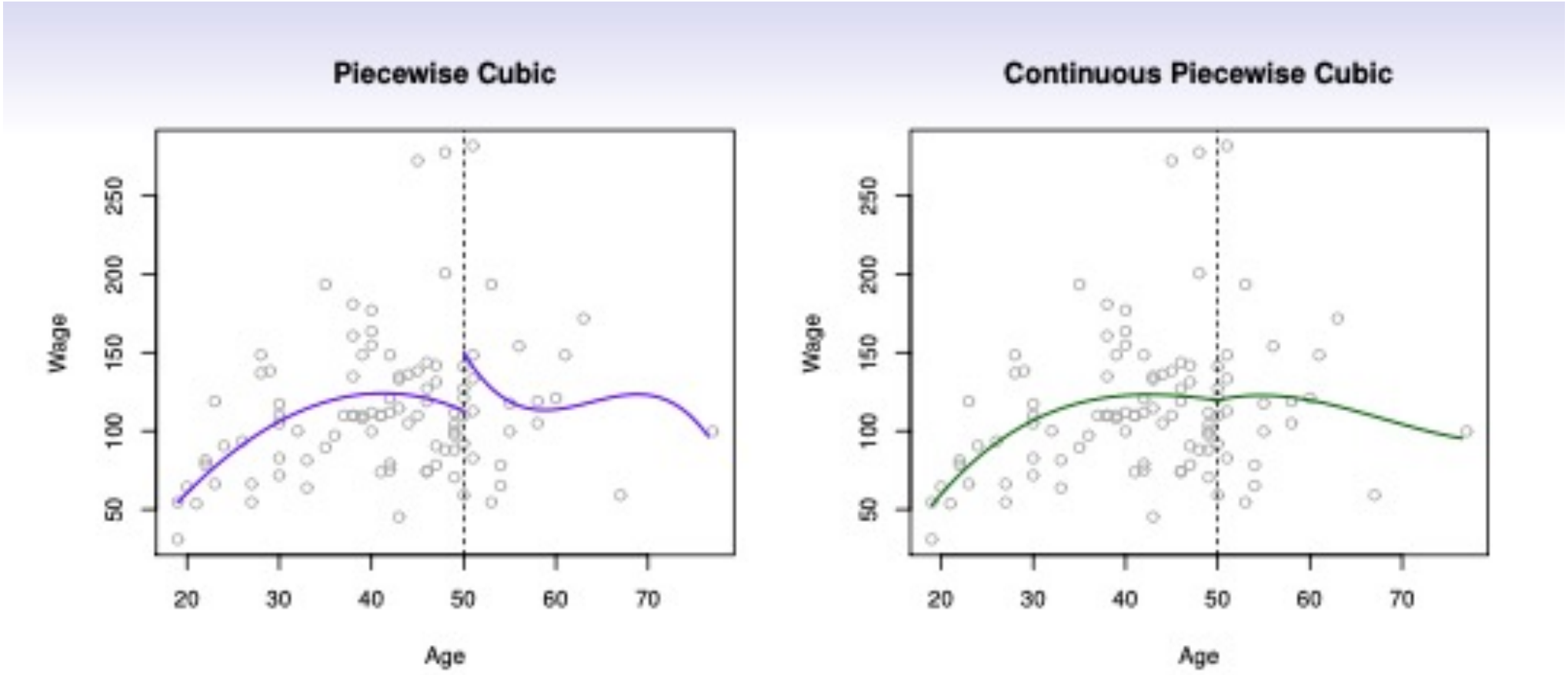
The second principal component has largest variance, subject to being uncorrelated with the first.

Piecewise Polynomials

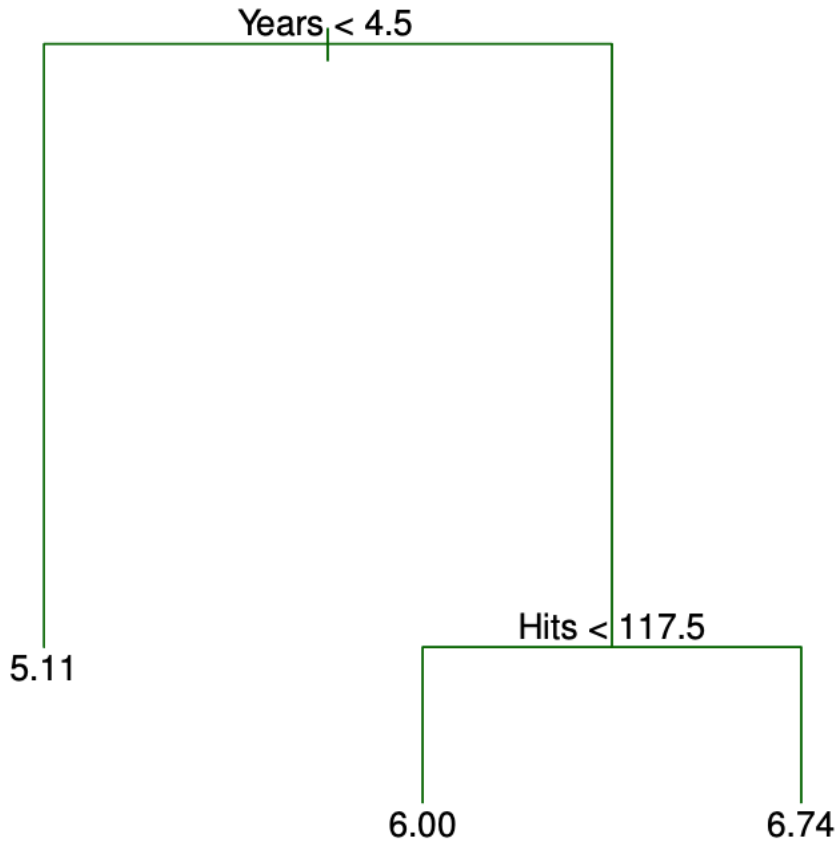
Instead of a single polynomial in X over its whole domain, we can rather use different polynomials in regions defined by knots.

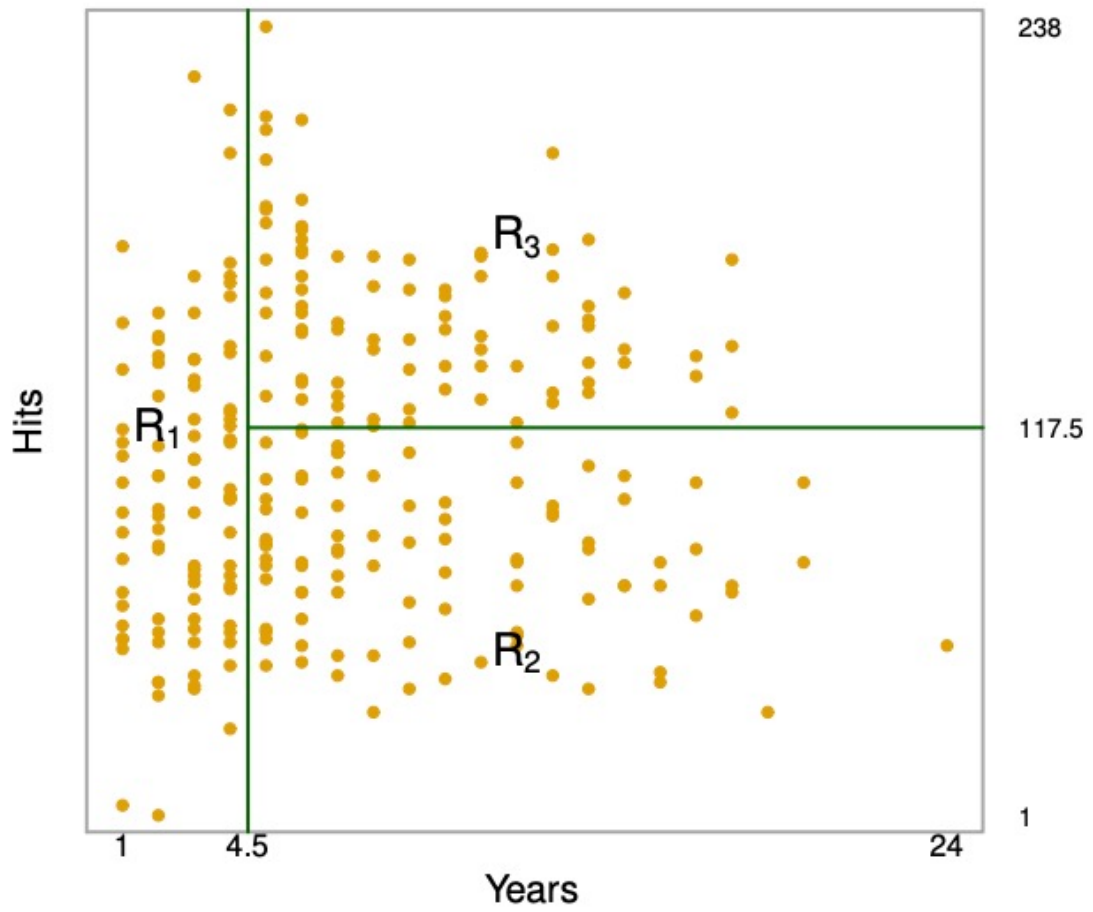
$$y_i = \begin{cases} \beta_{01} + \beta_{11} x_i + \beta_{21} x_i^2 + \beta_{31} x_i^3 + \epsilon_i & \text{if } x_i < c; \\ \beta_{02} + \beta_{12} x_i + \beta_{22} x_i^2 + \beta_{32} x_i^3 + \epsilon_i & \text{if } x_i \geq c. \end{cases}$$

Piecewise Polynomials



Decision Tree





Regression Tree-Building Process

The goal is to find boxes $R_1, \ldots, R_J$ that minimize the RSS, given by

$$\sum_{j=1}^{J} \sum_{i \in R_j} (y_i - \hat{y}_{R_j})^2,$$

where $\hat{y}_{R_j}$ is the mean response for the training observations within the jth box.

Greedy Method

We first select the predictor X_j and the cutpoint s such that splitting the predictor space into the regions $\{X|X_j < s\}$ and $\{X|X_j \geq s\}$ leads to the greatest possible reduction in RSS.

Greedy Method

Next, we repeat the process, looking for the best predictor and best cutpoint in order to split the data further so as to minimize the RSS within each of the resulting regions.

However, this time, instead of splitting the entire predictor space, we split one of the two previously identified regions. We now have three regions.

Bagging

Averaging a set of observations reduces variance. But this is not practical because we do not have access to multiple training sets.

Instead, we can bootstrap, by taking repeated samples from the (single) training data set.

Bootstrap aggregation, or bagging, is a general-purpose procedure for reducing the variance of a statistical learning method.

Random Forests

When building these decision trees, each time a split in a tree is considered, a random selection of m predictors is chosen as split candidates from the full set of p predictors. The split is allowed to use only one of those m predictors.

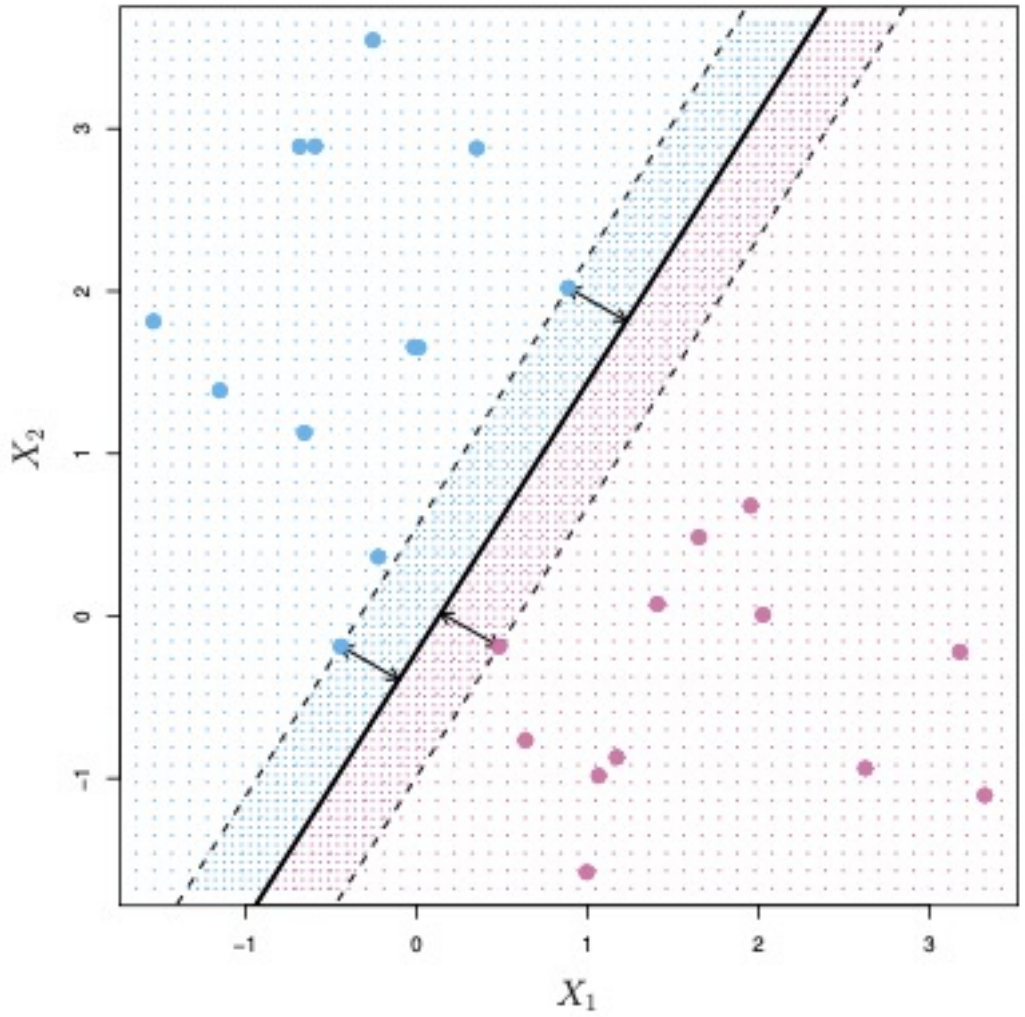
The reason: if one or a few features are very strong predictors for the target output, these features will be selected in many of the B trees, causing them (bagging trees) to become correlated.

Boosting Trees

Boosting works in a similar way as bagging, except that the trees are grown sequentially: each tree is grown using information from previously grown trees.

Unlike fitting a single large decision tree to the data, which amounts to fitting the data hard and potentially overfitting, the boosting approach instead learns slowly.

Support Vector Machine: Maximal Margin Classifier



Constrained optimization problem

$$\underset{\beta_0,\beta_1,\ldots,\beta_p}{\text{maximize}}\, M$$

$$\text{subject to } \sum_{j=1}^{p} \beta_j^2 = 1,$$

$$y_i(\beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip}) \geq M$$

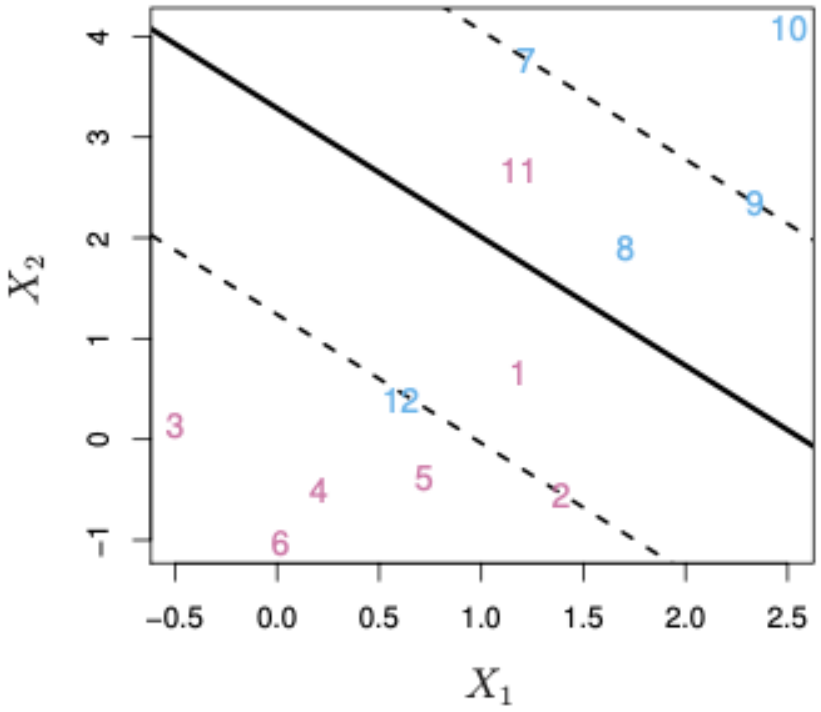
$$\text{for all } \; i = 1, \ldots, N.$$

Soft Margin: Support Vector Classifier

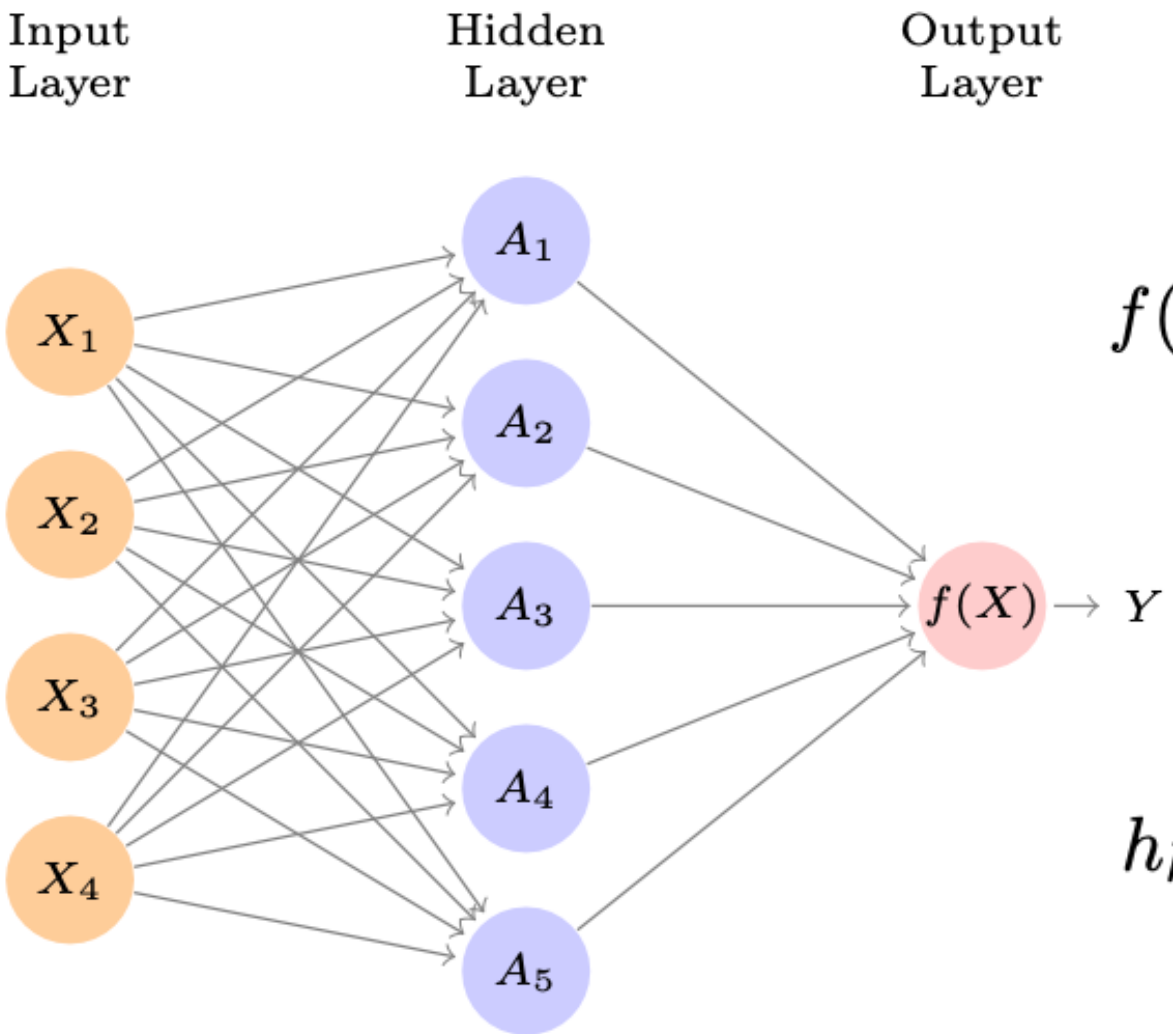
$$\underset{\beta_0,\beta_1,\ldots,\beta_p,\epsilon_1,\ldots,\epsilon_n}{\text{maximize}} \quad M \quad \text{subject to} \quad \sum_{j=1}^{p} \beta_j^2 = 1,$$

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip}) \geq M(1 - \epsilon_i),$$

$$\epsilon_i \geq 0, \quad \sum_{i=1}^{n} \epsilon_i \leq C,$$



Single Layer Neural Network



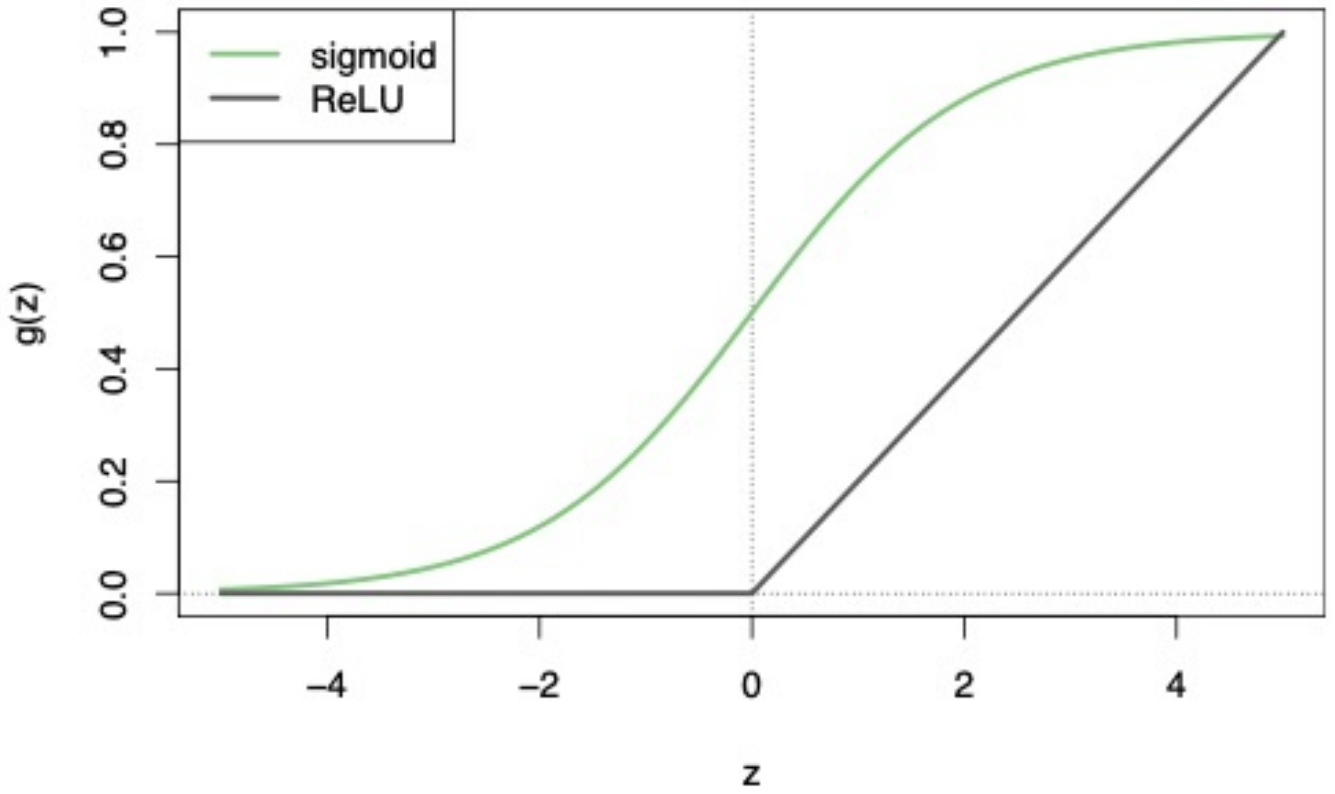
$$f(X) = \beta_0 + \sum_{k=1}^{K} \beta_k h_k(X)$$

$$h_k(X) = g(w_{k0} + \sum_{j=1}^{p} w_{kj} X_j)$$

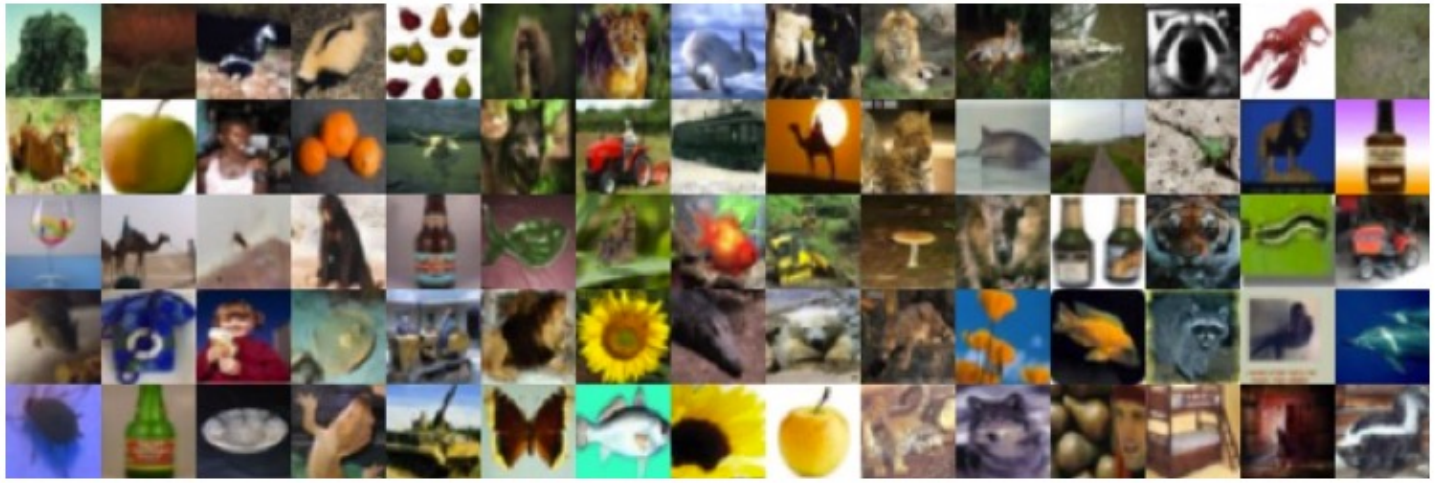
Activation Function

$$g(w_{k0} + \sum_{j=1}^{p} w_{kj} X_j)$$

Activation functions in hidden layers are typically nonlinear, otherwise the model collapses to a linear model.

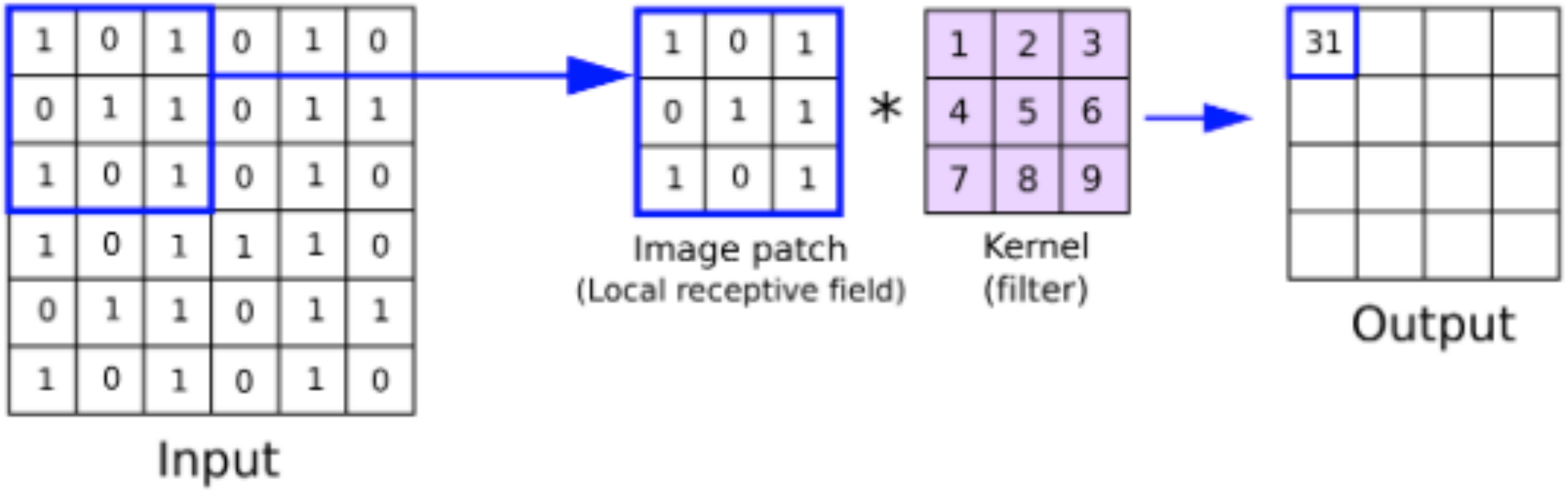


Convolutional Neural Network



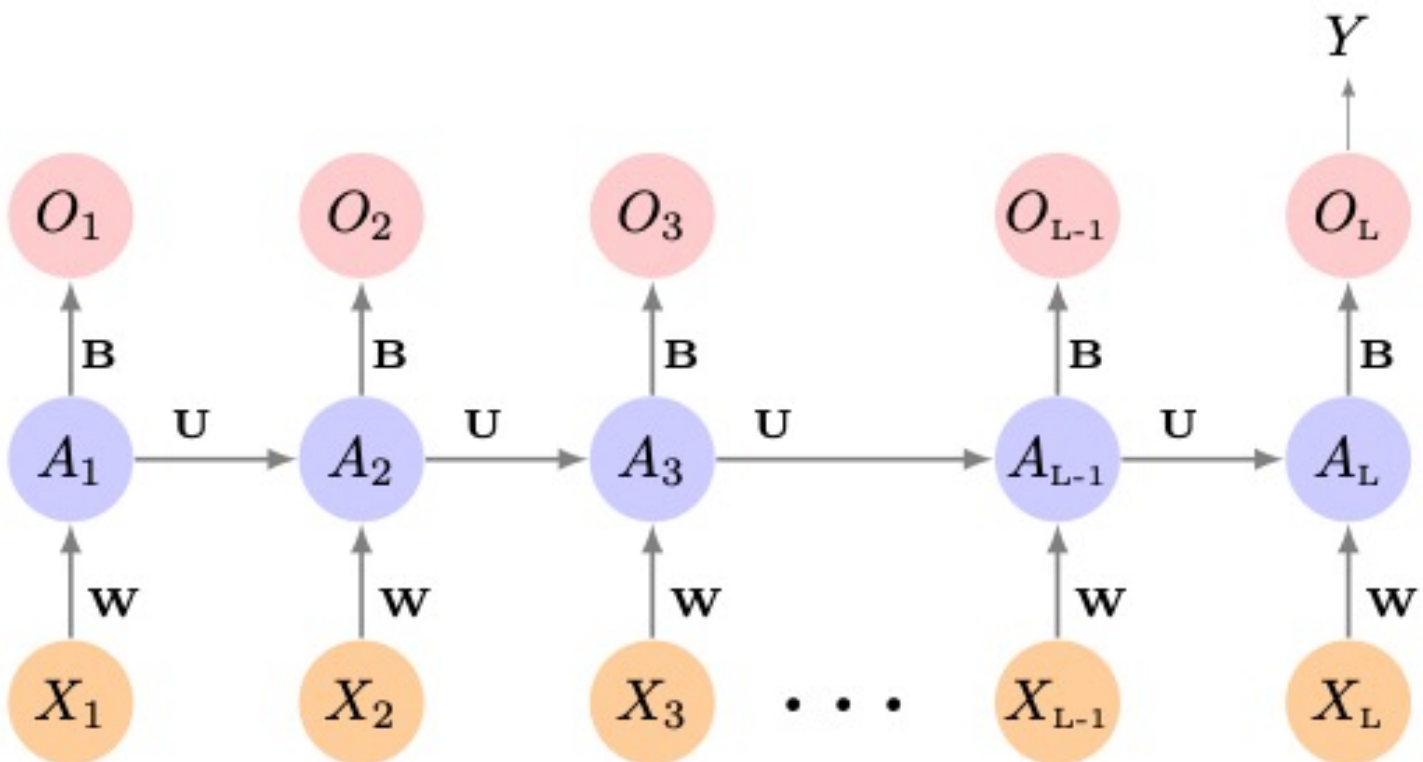
A commonly used network architecture for classifying images

Convolutional Kernel



Recurrent Neural Networks

The hidden layer is a sequence of vectors A_ℓ, receiving as input X_ℓ as well as $A_{\ell-1}$. A_ℓ produces an output O_ℓ.



Unsupervised Learning vs Supervised Learning

Unsupervised learning is more subjective than supervised learning, as there is no simple goal for the analysis, such as prediction of a response

It is often easier to obtain unlabeled data — from a lab instrument or a computer — than labeled data, which can require human intervention.

Principal Components Analysis (PCA)

PCA produces a low-dimensional representation of a dataset. It finds a sequence of linear combinations of the variables that have maximal variance, and are mutually uncorrelated.

Apart from producing derived variables for use in supervised learning problems, PCA also serves as a tool for data visualization.

Principal Components Analysis (PCA)

The first principal component of a set of features is the normalized linear combination of the features

$$Z_1 = \phi_{11} X_1 + \phi_{21} X_2 + \ldots + \phi_{p1} X_p$$

that has the largest variance. By normalized, we mean that

$$\sum_{j=1}^{p} \phi_{j1}^2 = 1$$

Principal Components Analysis (PCA)

$$\text{var}(x) = \frac{\sum_1^n (x_i - \bar{x})^2}{n-1}$$

$$\text{cov}(x, y) = \frac{\sum_1^n (x_i - \bar{x})(y_i - \bar{y})}{n-1}$$

$$\begin{bmatrix} \text{Var}(x_1) & \cdots & \text{Cov}(x_1, x_n) \\ \vdots & \ddots & \vdots \\ \text{Cov}(x_n, x_1) & \cdots & \text{Var}(x_n) \end{bmatrix}$$

Φ_1 is the eigenvector corresponding to the largest eigenvalue

Do not forget to normalize Φ_1

K-Means

Randomly assign a number, from 1 to K, to each of the observations. These serve as initial cluster assignments for the observations.

Iterate until the cluster assignments stop changing:

2.1 For each of the K clusters, compute the cluster *centroid*. The kth cluster centroid is the vector of the p feature means for the observations in the kth cluster.

2.2 Assign each observation to the cluster whose centroid is closest (where *closest* is defined using Euclidean distance).

Process of Hypothesis Testing

Hypothesis testing proceeds as follows:

1. Define the null and alternative hypotheses

2. Construct the test statistic (t-statistics, F-statistics)

3. Compute the p-value

4. Decide whether to reject the null hypothesis