Tutorial 9: Review of Supervised Learning

Rick Lin

The Chinese University of Hongkong, Shenzhen

November 14, 2022

Contents

Overview

Overview about supervised learning

Definition: Supervised learning (SL) is a machine learning paradigm for problems where the available data consists of labelled example.

The first half of the semester was all about the supervised learning. We have learnt several supervised learning models so far, namely linear model, logistic regression, SVM, tree-based model, and neural network.

Logistic Regression

Important concepts about logistic regression

- Hypothesis function: $f_{\mathrm{w},b}(\mathrm{x}) = g\left(\mathrm{w}^\top \mathrm{x} + b\right) = \frac{1}{1+\exp(-\mathrm{w}^\top \mathrm{x} - b)}$
- Cost function (cross-entropy and concave):

$$J(\mathrm{w}) = -\frac{1}{m}\sum_{i=1}^{m}\left[y_i \log\left(f_{\mathrm{w},b}\left(\mathrm{x}_i\right)\right) + \left(1-y_i\right)\log\left(1 - f_{\mathrm{w},b}\left(\mathrm{x}_i\right)\right)\right]$$

- One useful property about the sigmoid function:

$$f'(x) = f(x)(1 - f(x))$$

Exercises

Q1: Consider the sigmoid function $f(x) = 1/\left(1 + e^{-x}\right)$. The derivative $f'(x)$ is

a) $f(x)(1 - f(x))$

b) $f(x) \ln f(x) + (1 - f(x)) \ln(1 - f(x))$

c) $f(x) \ln(1 - f(x))$

d) $f(x)(1 + f(x))$

Q2: Logistic Regression

a) Minimizes cross-entropy loss

b) Models the log-odds as a linear function

c) Has a simple, closed form analytical solution

d) Is a classification method to estimate class posterior probabilities

Tree-based Model

Decision Tree, Bagging and Random Regression

- **Decision Trees** (DTs) are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. A tree can be seen as a piecewise constant approximation.
- **Bagging classifier** is an ensemble meta-estimator that fits base classifiers each on random subsets of the original dataset and then aggregate their individual predictions (either by voting or by averaging) to form a final prediction. (Data-level Randomness)
- **Random forest** is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. (Data-level randomness and attribute randomness)

Impurity

- **Entropy**:

$$\phi(p) = -\sum_{i=1}^{K} p_i \log_2 p_i$$

- **Gini Impurity or Gini Index**:

$$\phi(p) = \sum_{i=1}^{K} p_i \left(1 - p_i\right) = 1 - \sum_{i=1}^{K} p_i^2$$

- **Misclassification Rate or Misclassification Error**

$$\phi(p) = 1 - \max_i p_i$$

Exercises

Q1: An infinite depth binary Decision Tree can always achieve 100% training accuracy, provided that no point is mislabeled in the training set.

a) True

b) False

Q2: What strategies can help reduce overfitting in decision trees?

a) Pruning

b) Enforce a minimum number of samples in leaf nodes

c) Make sure each leaf node is one pure class

d) Enforce a maximum depth for the tree

Exercises

Q3 Which of the following are true about bagging?

a) In bagging, we choose random subsamples of the input points with replacement

b) The main purpose of bagging is to decrease the the bias of learning algorithms.

c) Bagging is ineffective with logistic, because all of the learners learn exactly the same decision boundary.

d) If we use decision trees that have one sample point per leaf, bagging never gives lower training error than one ordinary decision tree

Exercises

Q4: Compute the Gini impurity, entropy, misclassification rate for nodes A, B and C, as well as the overall metrics (Gini impurity, entropy, misclassification error) at depth 1 of the decision tree shown below.

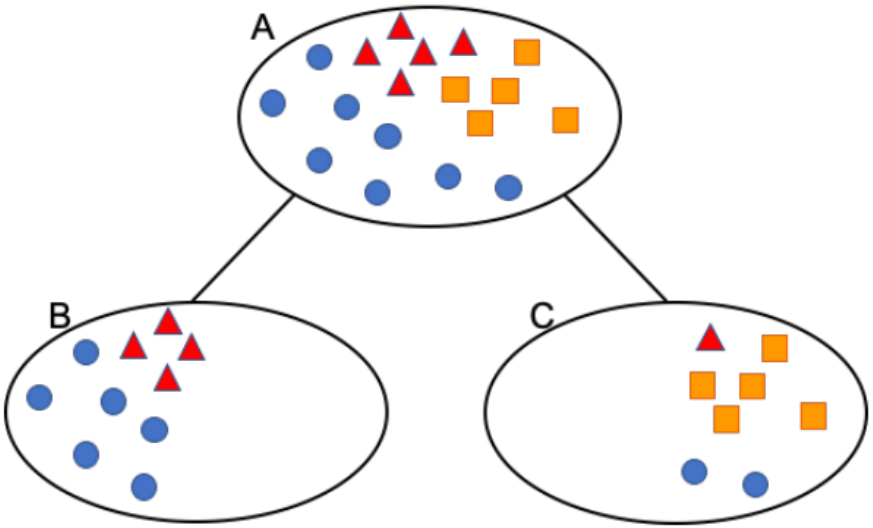


Figure: Sample decision tree problem

Neural Network and CNN

Important Concepts about neural network

- **Perceptron model**

  (1) One input layer to receive input signals.
  (2) One output layer including one M-P neuron
  (3) Cannot solve XOR problem.

- **Multi-layer feedforward neural networks**

  (1) Input layer, hidden layer(s), output layer.
  (2) Only on direction from the input layer to the output layer.
  (3) Fully connection between two layers No connections among neurons in the same layer, no connections among neurons in non-adjacent layers.

- **Backpropagation** is a widely used algorithm for training feedforward neural networks. See the lecture slides for detailed information.

Important formula about CNN

- Formula of the output size:

$$\text{output size } = \lfloor \frac{\text{input size } + 2 \times \text{padding} - \text{ kernel size}}{\text{stride}} \rfloor + 1$$

- Number of parameters:

$$\text{filter size }^2 \times \text{channel in} \times \text{channel out} + \text{channel out}$$

Exercises

Q1: It is possible to represent a XOR function with a neural network without a hidden layer.

a) True

b) False

Q2: Consider one layer of weights (edges) in a convolutional neural network (CNN) for grayscale images, connecting one layer of units to the next layer of units. Which type of layer has the fewest parameters to be learned during training? (Select one.)

a) A convolutional layer with 10 3×3 filters.

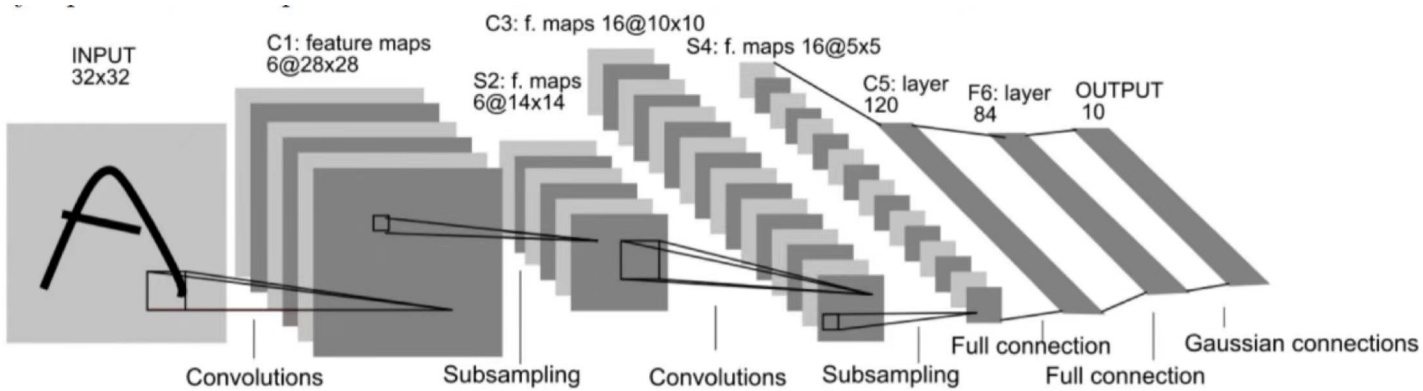
b) A convolutional layer with 8 5×5 filters.

c) A max-pooling layer that reduces a 10×10 image to 5×5.

d) A fully-connected layer from 20 hidden units image to 4 output units.

Exercises

Q3: Neural Networks Here is the historical LeNet Convolutional Neural Network architecture of Yann LeCun et al. for digit classification that we've discussed in class. Here, the INPUT layer takes in a 32×32 image, and the OUTPUT layer produces 10 outputs. The notation $6@28 \times 28$ means 6 matrices of size 28×28.



If the parameters of a given layer are the weights that connect to its inputs,
(1) Given that the input size is 32×32, and the Layer 1 size is 28×28, what's the size of the convolutional filter in the first layer (i.e. how many inputs is each neuron connected to)?
(2) How many independent parameters (weight and bias) are in C1, C3 and F6?

Variance-Bias tradeoff

Underfitting and Overfitting

- **Underfitting**: A model or a ML algorithm is said to have underfitting when it cannot capture the underlying trend of the data.
- **Overfitting**: A model or a ML algorithm is said to have overfitting when it captures the underlying trend of the data very accurately.
- **Best fit**: A model or a ML algorithm is said to have best fit when it captures the underlying trend of the data moderately and ignores the irrelevant details.

Variance and Bias

- **Bias**: The bias error is an error from erroneous assumptions in the learning algorithm. High bias can cause an algorithm to miss the relevant relations between features and target outputs (underfitting).
- **Variance**: The variance is an error from sensitivity to small fluctuations in the training set. High variance may result from an algorithm modeling the random noise in the training data (overfitting).

Underfitting	Overfitting	Best fit
High bias	Very low bias	Low bias
High variance	High variance	Low variance

Table: Relationship between Preceding Concepts

Exercises

Q1: You trained a binary classifier model which gives very high accuracy on the training data, but much lower accuracy on validation data. The following may be true:

a) This is an instance of overfitting.

b) This is an instance of underfitting.

c) The training was not well regularized.

d) The training and testing examples are sampled from different distributions.

Exercises

Q2: Which of the following are reasons why you might adjust your model in ways that increase the bias?

a) You observe high training error and high validation

b) You observe low training error and high validation error

c) You have few data points

d) Your data are not linearly separable

Exercises

Q3: Which of the following quantities affect the bias-variance tradeoff?

a) λ, the regularization coefficient in ridge regression

b) ϵ, the learning rate in gradient descent

c) d, the polynomial degree in least-squares regression

d) C, the slack parameter in soft-margin SVM

Thank you!