Final Review

Junyang Deng

January 27, 2023

1. For a hyperplane, $g(x) = w^T x + w_0$, please describe its meanings in regression, artificial neural networks, and support vector machine.
   SOLUTION

   (a) In regression, a hyperplane $g(x)$ minimizes the difference between predicted y and true y (measured by MSE). Most of the time, the optimal hyperplane can be solved analytically by $\boldsymbol{w} = \left(\mathbf{D}^T\mathbf{D}\right)^{-1}\mathbf{D}^T\boldsymbol{r}$ where $\boldsymbol{w}$ is the hyperplane, $\mathbf{D}$ is a matrix of $x_1, ...x_n$, and $\boldsymbol{r}$ is the class label. When an analytical solution is unavailable, we can use gradient descent to search for the best plane. With the hyperplane, people can predict new entry with known variables.

   (b) In ANN, each node (or called perceptron) includes a hyperplane. Each perceptron has inputs either from the environment or outputs of other perceptrons. For each input, the node can calculate a weighted sum of inputs by $y = \sum_{j=1}^{d} w_j x_j + w_0$, w_0 is the intercept value to make the model more general. This hyperplane divides the input space into two, which allows the perceptron to separate classes.

   (c) In support vector machine, the hyperplane is solved by maximizing the margin between classes. To be specific, the task is to find $\min \frac{1}{2}\|\boldsymbol{w}\|^2$ subject to $r^t\left(\boldsymbol{w}^T\boldsymbol{x}^t + w_0\right) \geq +1, \forall t$. The resulting $\boldsymbol{w}$ can be used to construct hyperplane.

2. What is the "gradient descent" optimization algorithm? Describe its usages in machine learning approaches.
   SOLUTION
   The "gradient descent" optimization algorithm is first construct a loss function $E(w)$, which is a differentiable function of a vector of variables. Then, for each parameter w,

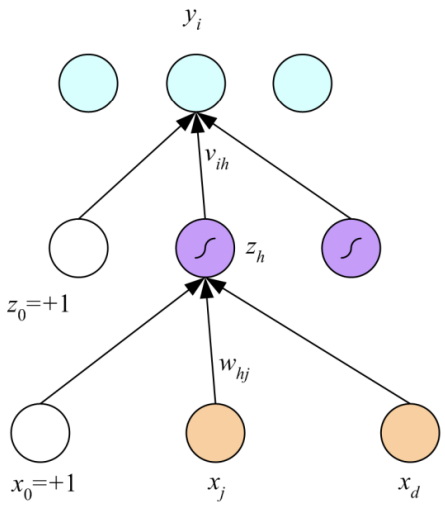
$$\Delta w_i = -\eta \frac{\partial E}{\partial w_i}, \forall i$$
$$w_i = w_i + \Delta w_i$$

   where η is called the learning factor, and determines how much to move in that direction. The use of a good value for η is critical; if it is too small, the convergence may be too slow, and a large value may cause oscillations and even divergence.

   Gradient descent can be applied in many machine learning approaches, including linear regression, logistic regression and neural network.

3. Describe the backpropagation algorithms in multilayer perceptrons.
   SOLUTION
   When we have a set of parameters, we can make predictions about classes or conduct regression based

on the given information. The predicted value may be different from true value, and such difference is measured by error function. In order to use this error to update parameters, we use the backpropagation algorithm. For a nonlinear regression, For example, with $\frac{\partial E}{\partial w_{hj}} = \frac{\partial E}{\partial y_i}\frac{\partial y_i}{\partial z_h}\frac{\partial z_h}{\partial w_{hj}}$, error E propagates back to w. The general update rule is Update = LearningFactor × (DesiredOutput - ActualOutput) × Input.



$$\frac{\partial E}{\partial w_{hj}} = \frac{\partial E}{\partial y_i}\frac{\partial y_i}{\partial z_h}\frac{\partial z_h}{\partial w_{hj}}$$

In the concept of gradient- descent

$$\begin{aligned}
\Delta w_{hj} &= -\eta \frac{\partial E}{\partial w_{hj}} \\
&= -\eta \sum_t \frac{\partial E^t}{\partial y^t}\frac{\partial y^t}{\partial z_h^t}\frac{\partial z_h^t}{\partial w_{hj}} \\
&= -\eta \sum_t \underbrace{-\left(r^t - y^t\right)}_{\partial E^t/\partial y^t} \underbrace{v_h}_{\partial y^t/\partial z_h^t} \underbrace{z_h^t\left(1-z_h^t\right)x_j^t}_{\partial z_h^t/\partial w_{hj}} \\
&= \eta \sum_t \left(r^t - y^t\right) v_h z_h^t \left(1-z_h^t\right) x_j^t
\end{aligned}$$

4. Describe "Kernel Trick" in support vector machine and present at least three different kernel functions.

SOLUTION

Kernel trick allows people to solve nonlinear problems without transformation. The idea is to replace the inner product of basis functions, $\boldsymbol{\phi}\left(\boldsymbol{x}^t\right)^T \boldsymbol{\phi}\left(\boldsymbol{x}^s\right)$, by a kernel function, $K\left(\boldsymbol{x}^t, \boldsymbol{x}^s\right)$, between instances in the original input space. Instead of mapping two instances $\boldsymbol{x}^t$ and $\boldsymbol{x}^s$ to the $\boldsymbol{z}$-space and doing a dot product there, we directly apply the kernel function in the original space.

There are many different kernel functions:

- Dot product kernel: $K\left(\boldsymbol{x}^t, \boldsymbol{x}\right) = \boldsymbol{x}^T \boldsymbol{x}^t$
- Polynomial (with degree q): $K\left(\boldsymbol{x}^t, \boldsymbol{x}\right) = \left(\boldsymbol{x}^T \boldsymbol{x}^t\right)^q$
- Radial-basis function: $K\left(\boldsymbol{x}^t, \boldsymbol{x}\right) = \exp\left[-\frac{\left\|\boldsymbol{x}^t - \boldsymbol{x}\right\|^2}{2s^2}\right]$
- Mahalanobis kernel: $K\left(\boldsymbol{x}^t, \boldsymbol{x}\right) = \exp\left[-\frac{1}{2}\left(\boldsymbol{x}^t - \boldsymbol{x}\right)^T \mathbf{S}^{-1}\left(\boldsymbol{x}^t - \boldsymbol{x}\right)\right]$
- Sigmoid function: $K\left(\boldsymbol{x}^t, \boldsymbol{x}\right) = \tanh\left(2\boldsymbol{x}^T \boldsymbol{x}^t + 1\right)$

5. Classes are in non-linear space; how to make them linear in a new space. Propose methods and apply them when using linear discrimination and SVM.
   SOLUTION
   Transformation can map non-linear separable classes into linear separable ones. Specifically, we can define a set of nonlinear basis functions Φ_{ij} to map the original data to a new space where the function can be written in a linear form.

   After transformation, linear discrimination and SVM can be applied directly in the new space.

6. Describe "sigmoid function, sigmoid(x)", which is applied to logistic discrimination using a hyperplane. (Hint: using the posteriors of $C1$ and $C2$, given x).
   SOLUTION