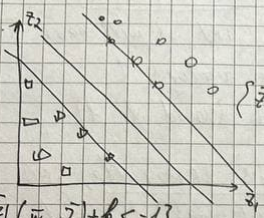
Group	Name	60	61	62	63	64	65	66	$\sum$
	Svetlana Seliunina								
	Aleksei Zhuravlev								

Technical Neural Networks Assignment Sheet 10

January 2023

Assignment 60



$\{\bar{z} \mid (\bar{w} \cdot \bar{z}) + b > 1\}$

$\{\bar{z} \mid (\bar{w} \cdot \bar{z}) + b < -1\}$

The hyperplane is designed in a way that the corridor is empty and all training points are outside the corridor. The scalar product for the delimiting hyperplanes is yielding -1 or $+1$, and for training data ≤ -1 or $\geq +1$.

For every data point from the 0 data set X_0, the scalar product will yield -1 or below:

$(\bar{w} \cdot \bar{X}_0) + b \leq -1{,}0$

We know that the teacher y_0 must be -1, so the product is $y_i\,((\bar{w} \cdot \bar{X}_0) + b) \geq +1$

Assignment 61

$f(x,y,z) = x + y - z$, $g(x,y,z) = x^2 + y^2 + z^2 = q = 3/100$

$F(x,y,z,\lambda) = f(x,y,z) - \lambda(g(x,y,z) - q)$

$x + y - z - \lambda x^2 - \lambda y^2 - \lambda z^2 + \lambda \cdot 3/100$

Partial derivatives of F:

$F_x = 1 - 2\lambda x = 0$, $x = 1/2\lambda$

$F_y = 1 - 2\lambda y = 0$; $y = 1/2\lambda$

$F_z = -1 - 2\lambda z = 0$, $z = -1/2\lambda$

$F_\lambda = -x^2 - y^2 - z^2 + 3/100 = 0$

Put x, y, z to F_λ:

$$-\frac{1}{4\lambda^2} - \frac{1}{4\lambda^2} - \frac{1}{4\lambda^2} + \frac{3}{100} = 0 \Rightarrow \lambda = \pm 5$$

$\lambda = 5$, $x = 1/10$, $y = 1/10$, $z = -1/10$

$\lambda = -5$, $x = -1/10$, $y = -1/10$, $z = 1/10$

$f(x,y,z) = x + y - z$.

$\lambda = 5$, $f(x,y,z) = \frac{3}{10}$ ← maximum $\lambda = -5$, $f(x,y,z) = -\frac{3}{10}$ ← minimum

Assignment 64

Give

Assignment 62

There is a second derivative test for constrained extrema. Here's the statement for the case of one constraint $g(x) = c$. Suppose a is a constrained critical point of f on the constraint set $g(x) = c$. Then we have $\nabla f(a) = \lambda \nabla g(a)$ for some scalar λ. Consider the Hessian matrix H (matrix of second partial derivatives) of $f(x) - \lambda g(x)$, evaluated at a. Then if $v^\top H v > 0$ for all v tangent to the constraint hypersurface at a, we conclude that a is a constrained local minimum; if $\boldsymbol{v}^\top \boldsymbol{H} \boldsymbol{v} < \mathbf{0}$ for all v tangent to the constraint hypersurface at a, we conclude that a is a constrained local maximum.

Assignment 63

All possible feature space points:

$$F(-1,-1) = (-1,-1,4)$$

$$F(-1,1) = (-1,1,0)$$
$$F(1,-1) = (1,-1,0)$$
$$F(1,1) = (1,1,4)$$

Formula for decision function: $f(Z) = sgn(z_3 - 3)$.

Assignment 64

Using the training sample $\{(x_i, d_i)\}_{i=1}^N$, find the optimum values of the weight vector w and bias b such that they satisfy the constraints $d_i\left(w^T x_i + b\right) \geq 1$ for $i = 1, 2, \dots, N$ and the weight vector w minimizes the cost function $\varphi(w) = \frac{1}{2} w^T w$. The scaling factor $1/2$ is for convenience. The primal form:

$$J(\vec{w}, b, \vec{\alpha}) = \frac{1}{2} w^T w - \sum_{i=1}^{N} \alpha_i \left[d_i \left(w^T x_i + b \right) - 1 \right]$$

We calculate the derivatives and after rearrangement we get:

$$\frac{\partial J(\vec{w}, b, \vec{\alpha})}{\partial \vec{w}} = 0 \leftrightarrow w = \sum_{i=1}^{N} \alpha_i d_i x_i$$
$$\frac{\partial J(\vec{w}, b, \vec{\alpha})}{\partial b} = 0 \leftrightarrow \sum_{i=1}^{N} \alpha_i d_i = 0$$

To construct the dual problem we expand the initial equation and get:

$$J(\vec{w}, b, \vec{\alpha}) = \frac{1}{2} w^T w - \sum_{i=1}^{N} \alpha_i d_i w^T x_i - b \sum_{i=1}^{N} \alpha_i d_i + \sum_{i=1}^{N} \alpha_i$$

Using the derivatives we see that some terms are zero using the derivatives calculated above, we get:

$$w^T w = \sum_{i=1}^{N} \alpha_i d_i w^T x_i = \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j d_i d_j x_i^T x_j$$

Setting the objective function $J(\vec{w}, b, \vec{\alpha}) = Q(\vec{\alpha})$ we may reformulate the equation:

$$Q(\vec{a}) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j d_i d_j x_i^T x_j$$

Where α_i are nonnegative and we have now obtained the dual form.

Assignment 65

Cross-validation (CV) is a technique for adjusting hyperparameters of predictive models. In K-fold CV, the available data S is partitioned into K subsets $S_1, \ldots, S_K$. Each data point in S is randomly assigned to one of the subsets such that these are of almost equal size (i.e., $\lfloor |S \mid /K \rfloor \leq |Si| \leq \lceil |S|/K \mid \rceil$).

Further, we define:

$$S_{\setminus i} = \bigcup S_{i\, j=1,\ldots,K \wedge j \neq i}$$

as the union of all data points except those in S_i. For each $i = 1, \ldots, K$, an individual model is built by applying the algorithm to the training data $S_{\setminus i}$. This model is then evaluated by means of a cost function using the test data in S_i. The average of the K outcomes of the model evaluations is called cross-validation (test) performance or cross-validation (test) error and is used a predictor of the performance of the algorithm when applied to S. Typical values for K are 5 and 10 .

T. Hastie, R. Tibshirani and J. Friedman. The Elements of Statistical Learning, section 4.3. SpringerVerlag, 2008.

Reqularization parameter:

The C parameter tells the SVM optimization how much you want to avoid misclassifying each training example. For large values of C, the optimization will choose a smaller-margin hyperplane if that hyperplane does a better job of getting all the training points classified correctly. Conversely, a very small value of C will cause the optimizer to look for a larger-margin separating hyperplane, even if that hyperplane misclassifies more points. For very tiny values of C, we should get misclassified examples, often even if your training data is linearly separable.

Reqularization parameter values:

- small C allows constraints to be easily ignored → large margin

- large C makes constraints hard to ignore → narrow margin

- C = ∞ enforces all constraints: hard margin

A. Zisserman, Machine Learning Lecture, The SVM classifier, Hilary 2013, PDF format.

Grid-search

The de facto standard way of performing hyperparameter optimization is grid search, which is simply an exhaustive searching through a manually specified subset of the hyperparameter space of a learning algorithm. A grid search algorithm must be guided by some performance metric, typically measured by cross-validation on the training set or evaluation on a held-out validation set.

Since the parameter space of a machine learner may include real-valued or unbounded value spaces for certain parameters, manually set bounds and discretization may be necessary before applying grid search.

For example, a typical soft-margin SVM classifier equipped with an RBF kernel has at least two hyperparameters that need to be tuned for good performance on unseen data: a regularization constant C and a kernel hyperparameter ψ. Both parameters are continuous, so to perform grid search, one selects a finite set of "reasonable" values for each, say

$$C \in \{10, 100, 1000\}$$
$$\gamma \in \{0.1, 0.2, 0.5, 1.0\}$$

Grid search then trains an SVM with each pair (C, γ) in the cross-product of these two sets and evaluates their performance on a held-out validation set (or by internal cross-validation on the training set, in which case multiple SVMs are trained per pair). Finally, the grid search algorithm outputs the settings that achieved the highest score in the validation procedure.

Bergstra, James; Bengio, Yoshua (2012). "Random Search for Hyper-Parameter Optimization", Machine Learning Research

Chin-Wei Hsu, Chih-Chung Chang and Chih-Jen Lin (2010). A practical guide to support vector classification, Technical Report, National Taiwan University