CS 189 Introduction to Machine Learning
Fall 2018 DIS9

1 Generalization of SVM: piecewise linear boundaries with ReLUs

In this problem we explore a generalization of the support vector machine for binary classification. In particular, we will see how a neural net with one hidden layer and ReLU activations can express a piecewise linear decision boundary, which can help find a good decision boundary for data sets that are not linearly separable. To be more precise, in this problem we consider two dimensional data points $\mathbf{x} \in \mathbb{R}^2$ which have labels $y \in \{-1, 1\}$. Given a training set of such points we are interested to find a model, of the form

$$\hat{y}_{\mathbf{w},b,\alpha}(\mathbf{x}) = \sum_{j=1}^{k} \alpha_j \max\{\langle \mathbf{w}_j, \mathbf{x} \rangle + b_j, 0\}, \tag{1}$$

which fits the data, where the coefficients α_j and b_j are real valued scalars, the coefficients $\mathbf{w}_j$ are two dimensional, and k denotes the number of hidden units of the neural network. The coefficient λ quantifies the amount of regularization on $\mathbf{w}_j$ and α_j. The subscript of $\hat{y}_{\mathbf{w},b,\alpha}$ is a reminder that the prediction depends on the coefficients of the neural network.

From now on we use the shorthand notation $(\gamma)_+ = \max\{\gamma, 0\}$, for any real value γ. During training we are interested in minimizing the following hinge loss objective

$$F(\mathbf{w}_1, \ldots, \mathbf{w}_k, b_1, \ldots, b_k, \alpha_1, \ldots \alpha_k) = \frac{1}{n} \sum_{i=1}^{n} \left[(1 - y_i \cdot \hat{y}_{w,b,\alpha}(\mathbf{x}_i))_+ + \frac{\lambda}{2} \sum_{j=1}^{k} (\|\mathbf{w}_j\|_2^2 + \alpha_j^2) \right]. \tag{2}$$

We denote

$$f_i(\mathbf{w}, b, \alpha) = (1 - y_i \cdot \hat{y}_{w,b,\alpha}(\mathbf{x}_i))_+ + \frac{\lambda}{2} \sum_{j=1}^{k} (\|\mathbf{w}_j\|_2^2 + \alpha_j^2).$$

(a) As a warmup, we discuss problem 3c) from the midterm exam. Consider classifying two data points $\mathbf{x}_1 = (a, b)$, $\mathbf{x}_2 = (-a, -b)$, with labels $y_1 = +1$ and $y_2 = -1$, respectively. For this data, calculate the form of the maximum margin separating hyperplane which goes through the origin. Make sure you justify your answer mathematically. Recall that for linear classifiers, the maximum margin is defined as:

$$\max_{\mathbf{w} \in \mathbb{R}^d} \min_{1 \le i \le n} \left(\frac{\mathbf{w}^\top \mathbf{x}_i}{\|\mathbf{w}\|_2} y_i \right)$$

Solution: There are only two data points, so the margin is

$$\begin{aligned}
&\max_{w} \min_{i=1,2} \left(\frac{\mathbf{w}^\top \mathbf{x}_i}{\|\mathbf{w}\|_2} y_i \right) \\
&= \max_{w:\|\mathbf{w}\|_2=1} \min \left\{ \mathbf{w}^\top \begin{pmatrix} a \\ b \end{pmatrix} (1), \mathbf{w}^\top \begin{pmatrix} -a \\ -b \end{pmatrix} (-1) \right\} \\
&= \max_{w:\|\mathbf{w}\|_2=1} \min \left\{ \mathbf{w}^\top \begin{pmatrix} a \\ b \end{pmatrix}, \mathbf{w}^\top \begin{pmatrix} a \\ b \end{pmatrix} \right\} \\
&= \max_{w:\|\mathbf{w}\|_2=1} \mathbf{w}^\top \begin{pmatrix} a \\ b \end{pmatrix}
\end{aligned}$$

By Cauchy-Schwarz we know that for any unit vector $\mathbf{w}$ we have $\mathbf{w}^\top \mathbf{x} \leq \|\mathbf{x}\|_2$. Since $\mathbf{w} = \mathbf{x}/\|\mathbf{x}\|_2$ saturates this upper bound, we know that the maximizing $\mathbf{w}$ is the unit vector in the direction $(a, b)^\top$. Therefore, we have that the maximizing hyperplane is defined by

$$\{\mathbf{x} : \mathbf{x}^\top \mathbf{w} = 0\}$$

where $\mathbf{w} = \begin{pmatrix} a \\ b \end{pmatrix}$.

(b) Now, let us consider a data set in $\mathbb{R}^2$ consisting of the 9 data points with integer coefficients (a, b), where $0 \leq a, b \leq 2$. A data point (a, b) is labelled $+1$ if $\max\{a, b\} \geq 2$ and is labelled -1 otherwise. Draw a picture of this data set and draw a maximum margin piecewise linear decision boundary for it. You do not need to be formal about proving it is maximum margin, but provide a discussion about why your choice maximizes the margin.

Solution:

Figure 1 shows the data set and the piecewise lienar maximum margin decision boundary. To see that this decision boundary maximes the margin we note that if the last column of data points were not present, then the horizontal dotten line would be the max margin decision boundary. Similarly, if the top row of the data set were not present then the vertical dotted line would be the max margin decision boundary. To be more precise, we note that the dotted lines interesect the coordinate axes at 1.5 and the margin is equal to $1/2$.

(c) Find a choice of k and a choice of coefficients $\mathbf{w}_1, \ldots, \mathbf{w}_k, b_1, \ldots, b_k, \alpha_1, \ldots \alpha_k$ such that the ReLU NN introduced in equation (1) achieves zero cost in terms of the loss function

$$\frac{1}{n} \sum_{i=1}^{n} \max\{1 - y_i \cdot \hat{y}_{w,b,\alpha}(\mathbf{x}_i), 0\}. \tag{3}$$

Discuss how the decision boundary of the ReLU NN you chose approximates the decision boundary found in the previous part (here, the decision boundary of the ReLU NN means the set $\{\mathbf{x} | \hat{y}_{\mathbf{w},b,\alpha}(\mathbf{x}) = 0\}$).

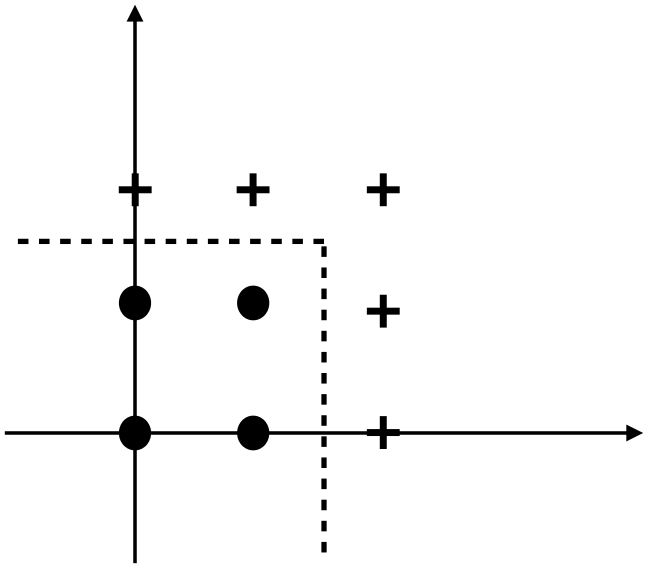

Figure 1: The piecewise lienar maximum margin decision boundary for the given data is depicted by dotted lines.

Solution: We see that we can take $k = 4$ and choose

$$\mathbf{w}_1 = (1, 0) \quad \mathbf{w}_2 = (-1, 0) \quad \mathbf{w}_3 = (0, 1) \quad \mathbf{w}_4 = (0, -1).$$

Then, we can choose $b_1 = b_3 = -1.5$ and $b_2 = b_4 = 1.5$ and $\alpha_1 = \alpha_3 = 5$ and $\alpha_2 = \alpha_4 = -1$. These choices ensure that the hinge loss is zero for all the data points.

We see that the sets

$$\{\mathbf{x} \colon \langle \mathbf{w}_1, \mathbf{x} \rangle + b_1 = 0 \text{ and } \langle \mathbf{w}_2, \mathbf{x} \rangle + b_2 = 0\}$$
$$\{\mathbf{x} \colon \langle \mathbf{w}_3, \mathbf{x} \rangle + b_3 = 0 \text{ and } \langle \mathbf{w}_4, \mathbf{x} \rangle + b_4 = 0\}$$

correspond to the vertical and horizontal decision boundaries found in the previous part. However, the decision boundary of the ReLU NN can be seen to look like the dotted lines in Figure 2. The larger we make α_1 and α_3 the better will the decision boundary of the ReLU NN approximate the maximum margin boundary found in the previous part.

(d) Compute the update rule for stochastic gradient descent on the objective F, defined in equation (2), for optimizing the coefficients $\mathbf{w}_j$, b_j and α_j.

Solution: Let's compute a gradient with respect to an arbitrary data point $(\mathbf{x}_i, y_i)$. We have that

$$\frac{\partial f_i}{\partial \alpha_j} = \begin{cases} -y_i(\langle \mathbf{w}_j, \mathbf{x}_i \rangle + b_j)_+ + \lambda \alpha_j & \text{if } y_i \cdot \hat{y}_{\mathbf{w},b,\alpha}(\mathbf{x}_i) < 1, \\ \lambda \alpha_j & \text{otherwise,} \end{cases}$$

because $\hat{y}_{\mathbf{w},b,\alpha}$ depends linearly on α_j (we abused notation and defined the derivative at zero to be zero although it is not defined).

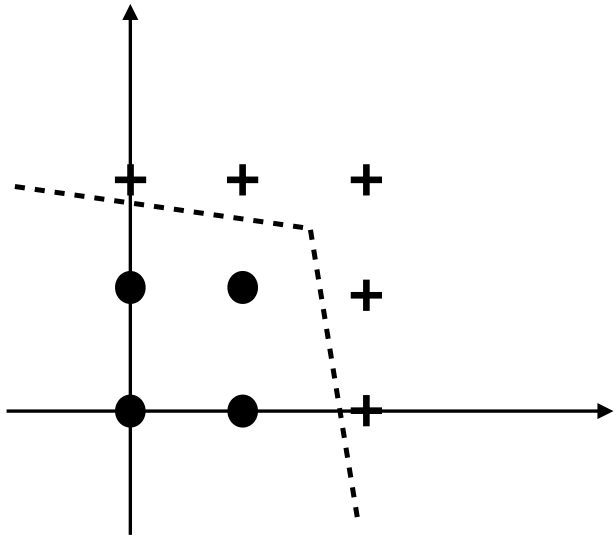

Figure 2: The piecewise lienar decision boundary for the ReLU NN. This drawing is approximate.

We also have that

$$\nabla_{\mathbf{w}_j} f_i = \begin{cases} -y_i \alpha_j \mathbf{x}_i + \lambda \mathbf{w}_j & \text{if } y_i \cdot \hat{y}_{\mathbf{w},b,\alpha}(\mathbf{x}_i) < 1 \text{ and } \langle \mathbf{w}_j, \mathbf{x} \rangle + b_j > 0, \\ \lambda \mathbf{w}_j & \text{otherwise,} \end{cases}$$

and

$$\frac{\partial f_i}{\partial b_j} = \begin{cases} -y_i \alpha_j & \text{if } y_i \cdot \hat{y}_{\mathbf{w},b,\alpha}(\mathbf{x}_i) < 1 \text{ and } \langle \mathbf{w}_j, \mathbf{x} \rangle + b_j > 0, \\ 0 & \text{otherwise,} \end{cases}$$

Now, if we denote by $\mathbf{w}_j^{(t)}$, $b_j^{(t)}$, and $\alpha_j^{(t)}$ the iterates at iteration t of SGD, we have

$$\mathbf{w}_j^{(t+1)} = \mathbf{w}_j^{(t)} - \gamma_t \nabla_{\mathbf{w}_j} f_i(\mathbf{w}^{(t)}, b^{(t)}, \alpha^{(t)}),$$

and the analogous for b_j and α_j.

(e) Given an arbitrary training set in $\mathbb{R}^2$ of distinct data points, intuitively discuss why a ReLU NN can perfectly fit the data if the number of hidden units k is large enough. Does it immediately follow that running SGD during training would find such a ReLU NN?

Solution:

With a large enough collection of linear decision boundaries we can construct any decision boundary for a finite collection of data points. To intuitively see this we can draw a decision boundary formed by three lines that isolates one given data point, i.e. the data point is in the center of the triangle formed with the three linear decision boundaries and no other data point is in this triangle.

We should not immediately expect SGD to find a decision boundary which perfectly classifies the training data because the optimization might get stuck in a local minimum. However, in practice, usually one can perfectly fit random data points when training with SGD if the NN has sufficiently many hidden units (many more hidden units than data points).