Solution 1: Perceptron Learning Algorithm

Given a dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$, where $\mathbf{x}_i \in \mathbb{R}^{d+1}$ is an augmented datavector $\begin{bmatrix}1 & x_{i,1} & \dots & x_{i,d}\end{bmatrix}^\top$ with label $y_i \in \{-1, 1\}$, the perceptron learning algorithm performs linear classification that can be written as

$$\hat{y}_i = h(\mathbf{x}_i) = \text{sign}(\mathbf{w}^\top \mathbf{x}_i), \tag{1}$$

where the model parameter $\mathbf{w} \in \mathbb{R}^{d+1}$ is a weightvector and $\hat{y}_n$ is the predicted label. Given an incorrectly classified training point $(\mathbf{x}_n, y_n) \in \mathcal{D}$, the weight update rule is

$$\mathbf{w}' = \mathbf{w} + y_n \mathbf{x}_n. \tag{2}$$

a) For a misclassified training point, let $\text{sign}(\mathbf{w}^\top \mathbf{x}_n) = \hat{y}_n$, for which we necessarily have $\hat{y}_n = -y_n \neq y_n$, and thus $y_n \hat{y}_n = -1$. Therefore, in general, without the sign$(\cdot)$ function that strips the magnitude of the dot product, we will have $y_n \mathbf{w}^\top \mathbf{x}_n < 0$. □

b) From the weight update step in equation (2), we have

$$y_n \mathbf{w}'^\top \mathbf{x}_n = y_n(\mathbf{w}^\top + y_i \mathbf{x}_n^\top)\mathbf{x}_n = y_n \mathbf{w}^\top \mathbf{x}_n + y_n^2 ||\mathbf{x}_n||^2.$$

Since $y_n^2 ||\mathbf{x}_n||^2 > 0$, we always have $y_n \mathbf{w}'^\top \mathbf{x}_n > y_n \mathbf{w}^\top \mathbf{x}_n$. □

c) From the above two proofs, notice that a misclassification always results in $y_n \mathbf{w}^\top \mathbf{x}_n < 0$, whereas a correct classification yields $y_n \mathbf{w}^\top \mathbf{x}_n > 0$. Therefore, it can be seen that the update step introduces a net addition towards making the inner product positive (hence a correct classification) for a misclassified point. This can also be viewed geometrically, where the weightvector $\mathbf{w}$, being normal to the decision boundary (a hyperplane), changes its alignment to the misclassified datavector $\mathbf{x}_n$. In the case where $y_n = 1 = -\hat{y}_n$, we have an obtuse angle between the weight and data vectors (since $\mathbf{w}^\top \mathbf{x}_n < 0$), for which $\mathbf{w}' = \mathbf{w} + \mathbf{x}_n$ aligns more closely with $\mathbf{x}_n$. In the other case where $y_n = -1 = -\hat{y}_n$, we have an acute angle between the two vectors, and the updated weight $\mathbf{w}' = \mathbf{w} - \mathbf{x}_n$ aligns less with $\mathbf{x}_n$. In both cases, the update magnitude is proportional to the norm of the datavector and is made in the correct direction, though it may misclassify other datavectors or not completely resolve the current error in one go. But since there is no selection preference or position constraint for a misclassified datavector, the algorithm will continue to explore freely in each update. □

Thus, with the above reasoning, we can conclude that we will have a definite and correct termination of the perceptron algorithm for a linearly separable dataset, since we will eventually come across an update step that resolves the only remaining misclassification when the decision boundary places itself in the label-separation region. Note that this algorithm does not produce unique (or deterministic) results and will not terminate for non-linearly separable datasets.

Solution 2: Linear Regression

Given a dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$, where $\mathbf{x}_i \in \mathbb{R}^{d+1}$ is an augmented datavector $\begin{bmatrix}1 & x_{i,1} & \dots & x_{i,d}\end{bmatrix}^\top$ with the corresponding output $y_i \in \mathbb{R}$, the linear regression model can be written as

$$\hat{y}_i = h(\mathbf{x}_i) = \mathbf{w}^\top \mathbf{x}_i, \tag{3}$$

where the model parameter $\mathbf{w} \in \mathbb{R}^{d+1}$ is a weightvector that is associated with the in-sample error, or loss function, being the mean squared error over the training dataset

$$E_{\text{in}}(\mathbf{w}) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 = \frac{1}{N}||\mathbf{y} - \hat{\mathbf{y}}||^2. \tag{4}$$

In equation (4), the vectors $\mathbf{y} = \begin{bmatrix}y_1 & \dots & y_N\end{bmatrix}^\top$ and $\hat{\mathbf{y}} = \begin{bmatrix}\hat{y}_1 & \dots & \hat{y}_N\end{bmatrix}$ are the true and predicted outputs over the training dataset, respectively. Note that we can write $\hat{\mathbf{y}} = \mathbf{X}\mathbf{w}$, for $\mathbf{X} = \begin{bmatrix}\mathbf{x}_1^\top & \dots & \mathbf{x}_N^\top\end{bmatrix}^\top \in \mathbb{R}^{N \times d+1}$ since the inner product is commutative, i.e., $\mathbf{w}^\top \mathbf{x} = \mathbf{x}^\top \mathbf{w}$.

a) In general, the squared norm for a vector is given by the inner product with itself. Therefore, we can write $\frac{1}{N}||\mathbf{y}-\mathbf{X}\mathbf{w}||^2 = \frac{1}{N}(\mathbf{y}-\mathbf{X}\mathbf{w})^\top(\mathbf{y}-\mathbf{X}\mathbf{w})$, which upon expanding gives the scalar loss

$$\begin{aligned} E_{\text{in}}(\mathbf{w}) &= \frac{1}{N}\Big(\mathbf{y}^\top\mathbf{y} - (\mathbf{X}\mathbf{w})^\top\mathbf{y} - \mathbf{y}^\top\mathbf{X}\mathbf{w} + (\mathbf{X}\mathbf{w})^\top(\mathbf{X}\mathbf{w})\Big) \\ &= \frac{1}{N}\big(\mathbf{y}^\top\mathbf{y} - 2\mathbf{y}^\top\mathbf{X}\mathbf{w} + \mathbf{w}^\top\mathbf{X}^\top\mathbf{X}\mathbf{w}\big) \\ &= \frac{1}{N}\big(||\mathbf{y}||^2 - 2\mathbf{y}^\top\hat{\mathbf{y}} + ||\hat{\mathbf{y}}||^2\big) \end{aligned} \tag{5}$$

Note that $\mathbf{y}^\top\mathbf{X}\mathbf{w}$ and $(\mathbf{X}\mathbf{w})^\top\mathbf{y}$ are equal and were grouped together since they are scalar quantities. □

b) To verify the gradient $\nabla_{\mathbf{w}}E_{\text{in}}(\mathbf{w})$ for both the sum and matrix expressions shown above, we first consider the gradient of error summation in equation (4)

$$\nabla_{\mathbf{w}}\left(\frac{1}{N}\sum_{i=1}^{N}(y_i - \underbrace{\mathbf{w}^\top\mathbf{x}_i}_{\hat{\mathbf{y}}_i})^2\right) = \frac{1}{N}\begin{bmatrix}\frac{\partial}{\partial w_0}\sum_{i=1}^{N}\big(y_i-(w_0x_{i,0}+\ldots+w_dx_{i,d})\big)^2 \\ \vdots \\ \frac{\partial}{\partial w_d}\sum_{i=1}^{N}\big(y_i-(w_0x_{i,0}+\ldots+w_dx_{i,d})\big)^2\end{bmatrix} = \frac{2}{N}\begin{bmatrix}\sum_{i=1}^{N}(\mathbf{w}^\top\mathbf{x}_i - y_i)x_{i,0} \\ \vdots \\ \sum_{i=1}^{N}(\mathbf{w}^\top\mathbf{x}_i - y_i)x_{i,d}\end{bmatrix}$$

$$= \frac{2}{N}\begin{bmatrix}\sum_{i=1}^{N}x_{i,0}(\mathbf{x}_i^\top\mathbf{w}) - \sum_{i=1}^{N}x_{i,0}y_i \\ \vdots \\ \sum_{i=1}^{N}x_{i,d}(\mathbf{x}_i^\top\mathbf{w}) - \sum_{i=1}^{N}x_{i,d}y_i\end{bmatrix} = \frac{2}{N}\left(\begin{bmatrix}\sum_{i=1}^{N}x_{i,0}\mathbf{x}_i^\top \\ \vdots \\ \sum_{i=1}^{N}x_{i,d}\mathbf{x}_i^\top\end{bmatrix}\mathbf{w} - \sum_{i=1}^{N}y_i\mathbf{x}_i\right) = \frac{2}{N}\big(\mathbf{X}^\top\mathbf{X}\mathbf{w} - \mathbf{X}^\top\mathbf{y}\big),$$

and then consider the gradient of the matrix expression in equation (5),

$$\nabla_{\mathbf{w}}\left(\frac{1}{N}\big(\mathbf{y}^\top\mathbf{y} - 2\underbrace{\mathbf{y}^\top\mathbf{X}}_{\mathbf{v}^\top}\mathbf{w} + \mathbf{w}^\top\underbrace{\mathbf{X}^\top\mathbf{X}}_{\mathbf{A}}\mathbf{w}\big)\right) = \frac{1}{N}\big(-2\mathbf{v} + 2\mathbf{A}\mathbf{w}\big) = \frac{2}{N}\big(\underbrace{\mathbf{X}^\top\mathbf{X}}_{\mathbf{A}=\mathbf{A}^\top}\mathbf{w} - \mathbf{X}^\top\mathbf{y}\big).$$

Hence, we have shown that both expressions are equivalent. □

c) Assuming $\mathbf{A} = \mathbf{X}^\top\mathbf{X}$ to be invertible, we let $\mathbf{w}^\star = \mathbf{A}^{-1}\mathbf{X}^\top\mathbf{y}$ and define $\hat{\mathbf{y}}_{\text{ls}} = \mathbf{X}\mathbf{w}^\star$. Based on these quantities, we can rewrite the loss function as

$$E_{\text{in}}(\mathbf{w}) = \frac{1}{N}||\underbrace{(\hat{\mathbf{y}}_{\text{ls}}-\hat{\mathbf{y}}) + (\mathbf{y}-\hat{\mathbf{y}}_{\text{ls}})}_{=\mathbf{y}-\hat{\mathbf{y}}}||^2 = \frac{1}{N}\big((\hat{\mathbf{y}}_{\text{ls}}-\hat{\mathbf{y}})^\top + (\mathbf{y}-\hat{\mathbf{y}}_{\text{ls}})^\top\big)\big((\hat{\mathbf{y}}_{\text{ls}}-\hat{\mathbf{y}}) + (\mathbf{y}-\hat{\mathbf{y}}_{\text{ls}})\big)$$

$$= \frac{1}{N}\big(||\hat{\mathbf{y}}_{\text{ls}}-\hat{\mathbf{y}}||^2 + 2(\hat{\mathbf{y}}_{\text{ls}}-\hat{\mathbf{y}})^\top(\mathbf{y}-\hat{\mathbf{y}}_{\text{ls}}) + ||\mathbf{y}-\hat{\mathbf{y}}_{\text{ls}}||^2\big), \text{ where } \hat{\mathbf{y}} = \mathbf{X}\mathbf{w}.$$

Further evaluating the inner product, we get

$$\begin{aligned}(\hat{\mathbf{y}}_{\text{ls}}-\hat{\mathbf{y}})^\top(\mathbf{y}-\hat{\mathbf{y}}_{\text{ls}}) &= (\mathbf{X}\mathbf{w}^\star - \mathbf{X}\mathbf{w})^\top(\mathbf{y}-\mathbf{X}\mathbf{w}^\star) = (\mathbf{w}^\star-\mathbf{w})^\top(\mathbf{X}^\top\mathbf{y} - \mathbf{X}^\top\mathbf{X}\mathbf{w}^\star) \\ &= (\mathbf{w}^\star-\mathbf{w})^\top(\mathbf{X}^\top\mathbf{y} - \mathbf{A}\mathbf{A}^{-1}\mathbf{X}^\top\mathbf{y}) = 0.\end{aligned}$$

Therefore, we have shown that

$$E_{\text{in}}(\mathbf{w}) = \frac{1}{N}\big(||\hat{\mathbf{y}}_{\text{ls}} - \mathbf{X}\mathbf{w}||^2 + ||\mathbf{y}-\hat{\mathbf{y}}_{\text{ls}}||^2\big). \tag{6}$$

Note that, this result follows from setting the gradient in part b) to zero, from which we can infer $\hat{\mathbf{y}}_{\text{ls}}$ to be the least squares prediction for the training dataset outputs $\mathbf{y}$. □

d) From equation (6), it can be seen that the loss is minimum when the predicted output $\hat{\mathbf{y}}$ equals least squares prediction $\hat{\mathbf{y}}_{\text{ls}}$, i.e.,

$$\min_{\mathbf{w}} E_{\text{in}}(\mathbf{w}) = \min_{\mathbf{w}}\frac{1}{N}\big(\underbrace{||\hat{\mathbf{y}}_{\text{ls}} - \mathbf{X}\mathbf{w}||^2}_{\text{variable}} + \underbrace{||\mathbf{y}-\hat{\mathbf{y}}_{\text{ls}}||^2}_{\text{constant}}\big) = \frac{1}{N}||\mathbf{y}-\hat{\mathbf{y}}_{\text{ls}}||^2,$$

since the variable term goes to zero (minimum norm). It follows from the definition of $\hat{\mathbf{y}}_{\text{ls}}$ in the previous proof, that the minimizer for the loss function is $\mathbf{w}^\star = \mathbf{A}^{-1}\mathbf{X}^\top\mathbf{y}$. Geometrically, $\hat{\mathbf{y}}_{\text{ls}} = \mathbf{X}\mathbf{w}^\star = \mathbf{P}\mathbf{y}$ corresponds to the projection of $\mathbf{y}$ onto the hyperplane $\mathcal{P} = \text{Im}(X)$ since it is given by a linear combination of the column vectors of $\mathbf{X}$. This also means that the residual error $(\mathbf{y} - \hat{\mathbf{y}}_{\text{ls}})$ is orthogonal to $\mathcal{P}$ which was supported by the previous proof when we showed that $\mathbf{X}^\top(\mathbf{y} - \mathbf{X}\mathbf{w}^\star) = 0$. Therefore, we have shown that any prediction by the linear regression model $\hat{\mathbf{y}} = \mathbf{X}\mathbf{w} \in \mathcal{P}$, from which it follows that $(\hat{\mathbf{y}} - \hat{\mathbf{y}}_{\text{ls}}) \perp (\mathbf{y} - \hat{\mathbf{y}}_{\text{ls}})$ $\square$