Machine Learning

Sanchit Chiplunkar

October 18, 2019

1 The Perceptron

Exercise 1

Since $\underline{x} \in \mathbb{R}^{d*k}$ and $\underline{W} \in \mathbb{R}^k$, the output vector $\overline{y}$ would be the resultant dimension of the matrix multiplication $\underline{x} * \underline{W}$, or $\overline{y} \in \mathbb{R}^d$.

Exercise 2

For $x \in \mathbb{R}^{d*k}, \overline{1_d}$ the d-dimensional vector of ones $\in \mathbb{R}^d$, and b as the scalar bias term, we have:

$\overline{y} = \underline{xW} + \overline{1_d}b$.

As discussed before, $\underline{xW} \in \mathbb{R}^d$. The operation $\overline{1_d}b \in \mathbb{R}^d, since$ b is a scalar.

Therefore, $\overline{y} \in \mathbb{R}^d$ following from linearity of matrix addition.

Exercise 3

Let $\overline{\hat{y}}$ be our training data outputs, and $\overline{y}$ our predicted outputs.Then the mean squared error in algebraic form would be:

$$\frac{1}{2d}\sum_{i=1}^{d}(\overline{y_i} - \overline{\hat{y}_i})$$

The equivalent expression in **vectorized** form is:

$$\frac{1}{2d}(\overline{y} - \overline{\hat{y}})^T(\overline{y} - \overline{\hat{y}})$$

Exercise 4

From before, we have MSE $= \frac{1}{2d}(\overline{y} - \overline{\hat{y}})^T(\overline{y} - \overline{\hat{y}})$.

Then, $\frac{dMSE}{dW} = \frac{d}{dW}[\frac{1}{2d}(\overline{y} - \overline{\hat{y}})^T(\overline{y} - \overline{\hat{y}})] = \frac{1}{2d}\frac{d}{dW}[(\overline{y} - \overline{\hat{y}})^T(\overline{y} - \overline{\hat{y}})]\frac{d}{dW}(\overline{y} - \overline{\hat{y}})$ by the chain rule.

We know, $\overline{y} = \underline{xW} + \overline{1_d}b$. So $\frac{d}{dW}\overline{y} = \underline{x}$

And given,$\frac{d}{dx}(x^T x) = 2\overline{x}$, we finally have $\frac{dMSE}{dW} = \frac{1}{2d}2(\overline{y} - \overline{\hat{y})}\underline{x} = \frac{1}{d}(\overline{y} - \overline{\hat{y}})\underline{x}$.

To accommodate for the rules of matrix multiplication, we note that $(\overline{y} - \overline{\hat{y}}) \in \mathbb{R}^d$ and $\underline{x} \in \mathbb{R}^{d*k}$.

The proper vectorized expression is then: $\frac{dMSE}{dW} = \underline{x}^T(\overline{y} - \overline{\hat{y}})$ □

Exercise 5

Now that we have our gradients, the weight update formula by gradient descent:
$w_{k+1} = w_k - \eta\frac{dMSE}{dW}$.
In vectorized form, the equivalent expression:
$\underline{W_{k+1}} = \underline{W_k} - \frac{\eta}{d}\underline{x}^T(\overline{y} - \overline{\hat{y}})$ □

Exercise 6

Given the data in Table 1 [$d = 10, k = 3$], learning rate of 0.02, $bias = 2$, and the initial weight matrix: $\begin{bmatrix} -0.1 \\ -0.3 \\ 0.2 \end{bmatrix}$

The updated weight matrix after one step of gradient descent [correct to two decimal points]:

$$\begin{bmatrix} 0.17 \\ -0.03 \\ 0.03 \end{bmatrix}$$

Exercise 7

Backpropagation is the process by which we propagate errors backwards [using the chain rule in calculus] through all the weight connections of the previous layers, and determine how much each weight of the network contributes to the overall error, and update them accordingly to minimise the error. To accomplish this, we use an iterative algorithm called **gradient descent**, which provides us with a rule of updating all the weights after each backward pass.