Neural Networks and Deep Learning: Exercises

Somnath Sikdar

January 21, 2020

Contents

Chapter 1

Using Neural Networks to Recognize Handwritten Digits

Exercise 1. Consider a network of perceptrons. Suppose that we multiply all weights and biases by a positive constant $c > 0$. Show that the behaviour of the network does not change.

Solution. First consider a single perceptron. Assume that weights and bias are $w_1, \ldots, w_n$ and b, respectively. Then $\sum_i w_i \cdot x_i + b$ and $c \cdot (\sum_i w_i \cdot x_i + b)$ have exactly the same sign and hence multiplying the weights and the bias by c will not change the behaviour of this single perceptron. Now if all perceptrons in a network have their weights and biases multiplied by $c > 0$, then each individual perceptron behaves as before and hence the network behaves as before. ■

Exercise 2. Suppose that we have network of perceptrons with a chosen input value $\boldsymbol{x}$. We won't need the actual input value, we just need the input to have been fixed. Suppose the weights and biases are such that all $\boldsymbol{w} \cdot \boldsymbol{x} + b \neq 0$ for the input $\boldsymbol{x}$ to any particular perceptron in the network. Now replace all the perceptrons in the network by sigmoid neurons, and multiply the weights and biases of the network by a positive constant $c > 0$. Show that in the limit as $c \to \infty$, the behaviour of this network of sigmoid neurons is exactly the same as the network of perceptrons. How can this fail when $\boldsymbol{w} \cdot \boldsymbol{x} + b = 0$ for one of the perceptrons?

Solution. As in the previous exercise, first consider a single perceptron in the network. When this is replaced by a sigmoid neuron, and we let $c \to \infty$, $c \cdot (\boldsymbol{w} \cdot \boldsymbol{x} + b)$ tends to either $+\infty$ or $-\infty$ depending on whether $\boldsymbol{w} \cdot \boldsymbol{x} + b$ is positive or negative. The upshot is that the output of the sigmoid neuron matches that of the perceptron it replaced. Thus when every sigmoid neuron behaves as the perceptron it replaced, the network as a whole behaves similarly.

This works as long as $\boldsymbol{w} \cdot \boldsymbol{x} + b \neq 0$. If this is zero, the output of the sigmoid neuron is "stuck" at $1/2$ irrespective of the value of c, while the perceptron outputs a 0. The outputs do not match and the behaviour of the sigmoid network may be different. ■

Exercise 3. There is a way of determining the bitwise representation of a digit by adding an extra layer to the three-layer network given in the book. The extra layer converts the output of the previous layer in binary representation. Find a set of weights and biases for

the new output layer. Assume that the first three layers of neurons are such that the correct output in the third layer (i.e., the old output layer) has activation at least 0.99, and incorrect outputs have activation less than 0.01.

Solution. Label the neurons of the third layer (the old output layer) as $0, 1, \ldots, 9$ and the neurons from the new output layer as $0', 1', 2', 3'$ with the interpretation that neuron $0'$ is the least significant bit and $3'$ is the most significant bit of the number represented by the output layer. The weight of the connection between the ith neuron from the third layer and the jth neuron of the output layer is w_{ij}, where $i \in \{0, \ldots, 9\}$ and $j \in \{0', 1', 2', 3'\}$. The bias of the jth output neuron is b_j. Denote the output of the ith neuron from the third layer as x_i. Then the input to the final layer may be represented as:

$$\begin{pmatrix} z_0 \\ z_1 \\ z_2 \\ z_3 \end{pmatrix} = \begin{pmatrix} w_{00} & w_{10} & \ldots & w_{90} & b_0 \\ w_{01} & w_{11} & \ldots & w_{91} & b_1 \\ w_{02} & w_{12} & \ldots & w_{92} & b_2 \\ w_{03} & w_{13} & \ldots & w_{93} & b_3 \end{pmatrix} \begin{pmatrix} x_0 \\ \vdots \\ x_9 \\ 1 \end{pmatrix}$$

Now we would like z_0 to be 1 when the number is $1, 3, 5, 7, 9$ and 0 otherwise. To be able to do this, first set

$$w_{10} = w_{30} = w_{50} = w_{70} = w_{90} = +1$$

and the remaining weights of the inputs to $0'$ to -1. Set $b_0 = 0$. Now if the third layer represents $k \in \{1, 3, 5, 7, 9\}$, we would have $w_{k0} > 0.99$ and $w_{j0} < 0.01$ for all $j \neq k$. With these weights, we would have $z_0 > 0.99 - 9 \times 0.01 = 0.90$. If the third layer represents a number $k \notin \{1, 3, 5, 7, 9\}$, then $z_0 < -0.99 + 9 \times 0.01 = -0.90$. We can amplify this phenomenon by multiplying all these weights by a large positive constant. This would lead the sigmoid neuron $0'$ to output a 1 for the digits $1, 3, 5, 7, 9$ and a 0 for the remaining digits.

We can use a similar strategy for the remaining neurons of the fourth layer. For example, the second most significant bit $1'$ must be a 1 for the digits $2, 3, 6, 7, 9$ and a 0 for the remaining digits. We would then set

$$w_{21} = w_{31} = w_{61} = w_{71} = w_{91} = +1$$

and the remaining weights to -1. The bias b_1 is set to 0. ■

Exercise 4. Let $C(v_1, \ldots, v_m)\colon \mathbf{R}^m \to \mathbf{R}$ be a differentiable function. Then $\Delta C \approx \nabla C \cdot \Delta \boldsymbol{v}$. Constrain $\|\Delta \boldsymbol{v}\| = \epsilon$, where $\epsilon > 0$ is a small fixed real. Show that the choice of $\Delta \boldsymbol{v}$ that minimizes $\nabla C \cdot \Delta \boldsymbol{v}$ is $\Delta \boldsymbol{v} = -\eta \nabla C$, where $\eta = \epsilon / \|\nabla C\|$.

Solution. The Cauchy-Schwarz inequality tells us that

$$\begin{aligned} |C^{(1)}_{v_1} \Delta v_1 + \cdots + C^{(1)}_{v_m} \Delta v_m| &\leq \left((C^{(1)}_{v_1})^2 + \cdots + (C^{(1)}_{v_m})^2\right)^{1/2} \left((\Delta v_1)^2 + \cdots + (\Delta v_m)^2\right)^{1/2} \\ &= \|\nabla C\| \cdot \epsilon, \end{aligned}$$

where $C^{(1)}_{v_i} = \frac{\partial C}{\partial v_i}$. Since the right-hand side is a positive number no matter what the values of the partial derivatives $C^{(1)}_{v_i}$ and the changes Δv_i in the values of the variables, the smallest

possible value of the left-hand side is $-\|\nabla C\| \cdot \epsilon$. Since we are trying to minimize ΔC which is approximated by the left-hand side, the goal is to find values for the Δv_i such that minimizes the left-hand side. Observe that when we set $\Delta v_i := -\epsilon \cdot \frac{\nabla C}{\|\nabla C\|}$ for all $1 \leq i \leq m$, then the left-hand side indeed equals the said minimum value. Hence it must be that this setting of the Δv_is is the optimum. ■

Chapter 2

The Backpropagation Algorithm

2.1 The Backpropagation Equations

Before we describe anything, we briefly recap notation. We let C denote the cost function and σ the activation function of the neurons.

1. w^l_{jk} is the weight of the link between the jth neuron in layer l and the kth neuron in layer $l-1$.
2. b^l_j is the bias of neuron j in layer l.
3. z^l_j is the weighted input to neuron j in layer l.
4. $a^l_j = \sigma(z^l_j)$ is the activation of neuron j in layer l.
5. $\delta^l_j := \partial C/\partial z^l_j$ is the "error" of neuron j in layer l.

Using this notation, we may write the weighted output to neuron j in the lth layer as:

$$z^l_j = \sum_k w^l_{jk} a^{l-1}_k + b^l_j = \sum_k w^l_{jk} \sigma(z^{l-1}_k) + b^l_j,$$

where the index k runs over all neurons in layer $l-1$ and $2 \leq l \leq L$. Symbols such as w^l, b^l, a^l without subscripts refer to either matrices or vectors as the case may be. For example, w^l refers to the matrix whose (j,k)th element is w^l_{jk}. This matrix has as many rows as there are neurons in the lth layer and as many columns as there are neurons in layer $l-1$. The symbol b^l refers to the vector of biases b^l_j of the neurons in layer l; similarly, a^l refers to the vector of activations a^l_j of the neurons in layer l.

With this notation in hand, we may write the backpropagation equations as:

$$\delta^L = \nabla_{a^L} C \odot \sigma'(z^L) \tag{2.1}$$

$$\delta^l = ((w^{l+1})^\intercal \delta^{l+1}) \odot \sigma'(z^l) \tag{2.2}$$

$$\frac{\partial C}{\partial b^l_j} = \delta^l_j \tag{2.3}$$

$$\frac{\partial C}{\partial w^l_{jk}} = a^{l-1}_k \delta^l_j \tag{2.4}$$