Chapter 4: Deep Neural Networks

Problem 4.1

Consider composing two neural networks in figure 4.8. Draw a plot of the relationship between the input x and output y' for $x \in [-1, 1]$.

Problem 4.2

Identify the four hyperparameters in figure 4.6.

1. Activation function $\mathbf{a}$
2. Number of hidden layers $\mathbf{h}_K$
3. Weights $\boldsymbol{\Omega}_K$
4. Biases $\boldsymbol{\beta}_K$

Problem 4.3

Using the non-negative homogeneity property of the ReLU function, show that:

$$\text{ReLU}\left[\boldsymbol{\beta}_1 + \lambda_1 \cdot \boldsymbol{\Omega}_1 \text{ReLU}\left[\boldsymbol{\beta}_0 + \lambda_0 \cdot \boldsymbol{\Omega}_0 \mathbf{x}\right]\right] = \lambda_0\lambda_1 \cdot \text{ReLU}\left[\frac{1}{\lambda_0\lambda_1}\boldsymbol{\beta}_1 + \boldsymbol{\Omega}_1 \text{ReLU}\left[\frac{1}{\lambda_0}\boldsymbol{\beta}_0 + \boldsymbol{\Omega}_0 \mathbf{x}\right]\right]$$

Where λ_0 and λ_1 are non-negative scalars. From this we see that the weight matrices can be rescaled by any magnitude as long as the biases are also adjusted, and the scale factors can be re-applied at the end of the network

We know that for $\alpha \in \mathbb{R}^+$:

$$\text{ReLU}[\alpha \cdot z] = \alpha \cdot \text{ReLU}[z]$$

Therefore, dividing the LHS of the above equation by λ_1:

$$\text{ReLU}\left[\boldsymbol{\beta}_1 + \lambda_1 \cdot \boldsymbol{\Omega}_1 \text{ReLU}\left[\boldsymbol{\beta}_0 + \lambda_0 \cdot \boldsymbol{\Omega}_0 \mathbf{x}\right]\right] = \lambda_1 \cdot \text{ReLU}\left[\frac{1}{\lambda_1}\boldsymbol{\beta}_1 + \boldsymbol{\Omega}_1 \text{ReLU}\left[\boldsymbol{\beta}_0 + \lambda_0 \cdot \boldsymbol{\Omega}_0 \mathbf{x}\right]\right]$$

And dividing the inner ReLU by λ_0:

$$\lambda_1 \cdot \text{ReLU}\left[\frac{1}{\lambda_1}\boldsymbol{\beta}_1 + \boldsymbol{\Omega}_1 \text{ReLU}\left[\boldsymbol{\beta}_0 + \lambda_0 \cdot \boldsymbol{\Omega}_0 \mathbf{x}\right]\right] = \lambda_0\lambda_1 \cdot \text{ReLU}\left[\frac{1}{\lambda_0\lambda_1}\boldsymbol{\beta}_1 + \boldsymbol{\Omega}_1 \text{ReLU}\left[\frac{1}{\lambda_0}\boldsymbol{\beta}_0 + \boldsymbol{\Omega}_0 \mathbf{x}\right]\right]$$

Problem 4.4

Write out the equations for a deep neural network that takes $D_i = 5$ inputs, $D_o = 4$ outputs and has three hidden layers of sizes $D_1 = 20, D_2 = 10,$ and$D_3 = 7$, respectively, in both the forms of equations 4.15 and 4.16. What are the sizes of each weight matrix $\boldsymbol{\Omega}_\bullet$ and bias vector $\boldsymbol{\beta}_\bullet$?

The network has three layers, therefore:

$$\begin{aligned}\mathbf{h}_3 &= \mathbf{a}[\boldsymbol{\beta}_2 + \boldsymbol{\Omega}_2\mathbf{h}_2] \\ \mathbf{y} &= \boldsymbol{\beta}_3 + \boldsymbol{\Omega}_3\mathbf{h}_3\end{aligned}$$

Where $\boldsymbol{\beta}_3$ has size 4×1 $(= D_o)$, $\boldsymbol{\Omega}_3$ has size 4×7 $(= D_o \times D_3)$, and $\mathbf{h}_3$ has size 7×1 $(D_3$. Equivalently, the network can be written as a single function:

$$\mathbf{y} = \boldsymbol{\beta}_3 + \boldsymbol{\Omega}_3\mathbf{a}[\boldsymbol{\beta}_2 + \boldsymbol{\Omega}_2\mathbf{a}[\boldsymbol{\beta}_1 + \boldsymbol{\Omega}_1\mathbf{a}[\boldsymbol{\beta}_0 + \boldsymbol{\Omega}_0\mathbf{a}]]]$$

$\boldsymbol{\beta}_2$ has size 7×1 $(= D_3)$, $\boldsymbol{\Omega}_2$ has size 7×10 $(= D_3 \times D_2)$.

$\boldsymbol{\beta}_1$ has size 10×1 $(= D_2)$, $\boldsymbol{\Omega}_1$ has size 10×20 $(= D_2 \times D_1)$.

$\boldsymbol{\beta}_0$ has size 20×1 $(= D_1)$, $\boldsymbol{\Omega}_0$ has size 20×5 $(= D_1 \times D_i)$.

(*Note: If the k^{th} layer has D_k hidden units, then the bias vector $\boldsymbol{\beta}_{k-1}$ will be of size D_k.*

Problem 4.5

Consider a deep neural network with $D_i = 5$ inputs, $D_o = 1$ output, and $K = 20$ hidden layers containing $D = 30$ hidden units each. What is the depth of this network? What is the width?

The number of hidden units in each layer is referred to as the *width* of the network, and the number of hidden layers as the *depth*. The total number of hidden units is a measure of the network's *capacity*. Therefore, the depth of this neural network is 20 and the width is 30.

Problem 4.6

Consider a network with $D_i = 1$ input, $D_o = 1$ output, and $K = 10$ layers, with $D = 10$ hidden units in each. Would the number of weights increase more if we increased the depth by one or the width by one? Provide your reasoning

The total number of weights for a network with 1 input, 1 output, and 10 hidden layers with 10 hidden units in each layer is calculated as follows:

- **Input to first hidden layer**: $1 \times 10 = 10$ weights
- **Between hidden layers**: Each of the 9 connections between the successive layers has $10 \times 10 = 100$ weights, resulting in a total of $9 \times 100 = 900$ weights
- **Last hidden layer to output**: $10 \times 1 = 10$

Resulting in a total of 920 weights. If the depth increased by one, then there would now be 10 connections between the hidden layers resulting in $10 \times 100 = 1000$ weights among the hidden layers, giving a total of 1020 weights. If the width is increased by one, each of the 9 connections between the hidden layers has $11 \times 11 = 121$ weights, resulting in a total of $9 \times 121 = 1089$ weights in the hidden layers, or $1089 + 22 = 1111$ total weights. Therefore, increasing the width by one results in a larger total number of weights.

Problem 4.7

Choose values for the parameters $\phi = \{\phi_0, \phi_1, \phi_2, \phi_3, \theta_{10}, \theta_{11}, \theta_{20}, \theta_{21}, \theta_{30}, \theta_{31}, \}$ for the shallow neural network in equation 3.1 that will define an identity function over a finite range $x \in [a, b]$.

$$y = \phi_0 + \phi_1 \mathrm{a}[\theta_{10} + \theta_{11}x] + \phi_2 \mathrm{a}[\theta_{20} + \theta_{21}x] + +\phi_3 \mathrm{a}[\theta_{30} + \theta_{31}x] \tag{3.1}$$

To create an identity function over a finite range $x \in [a, b]$, the parameters must be chosen such that $y = x$. Therefore we can set ϕ_0, ϕ_2, andϕ_3 to 0, and $\phi_1 = 1$ such that:

$$y = \mathrm{a}[\theta_{10} + \theta_{11}x]$$

And then set $\theta_{10} = 0$ and $\theta_{11} = 1$ such that $y = x$ as required.

Problem 4.8

Figure 4.9 shows the activations in the three hidden units of a shallow network (as in figure 3.3). The slopes in the hidden units are 1.0, 1.0, and -1.0, respectively, and the "joints" in the hidden units are at positions 1/6, 2/6, and 4/6. Find values of ϕ_0, ϕ_1, ϕ_2 and ϕ_3 that will combine the hidden unit activations as $\phi_0 + \phi_1 h_1 + \phi_2 h_2 + \phi_3 h_3$ to create a function with four linear regions that oscillate between output values of zero and one. The slope of the leftmost region should be positive, the next one negative, and so on. How many linear regions will we create if we compose this network with itself? How many will we create if we compose it with itself K times?

Points $x = 0, x = \frac{1}{6}, x = \frac{2}{6}$ and $x = \frac{4}{6}$ determine where the slope of the function changes. In region one, only hidden unit 3 is active. Therefore, since h_3 is only activated after $x = \frac{4}{6}$ and has a slope of -1, for the first section to start from $x = 0$ and have a positive slope, $\phi_0 = 4$ and $\phi_3 = -6$. The second region has slope $\phi_1 - \phi_3$ and requires a slope of -6 to bring the function down to zero at $x = \frac{2}{6}$, therefore $-6 = \phi_1 + 6$ so $\phi_1 = -12$. In the third region, the slope is $\phi_1 + \phi_2 - \phi_3$ which requires a slope of 3 to bring the function back to one at $x = \frac{4}{6}$. Therefore $3 = -12 + \phi_2 + 6$ so $\phi_2 = 9$.

If the network is composed with itself, there will be $4 \times 4 = 16$ linear regions, as each of the four regions in the second network will be replicated four times. If the network is composed with itself K times, there will be 4^K regions.

Problem 4.9

Following from problem 4.8, is it possible to create a function with three linear regions that oscillates back and forth between output values of zero and one using a shallow network with two hidden units? Is it possible to create a function with five linear regions that oscillates in the same way using a shallow network with four hidden units?

It is not possible to create a function with three linear regions that oscillates back and forth between output values of zero and one using a shallow network with two hidden units. This is because two hidden units introduce two breakpoints only, which results in three linear regions although oscillation between zero and one is not possible.

However, it is possible to create a function with five linear regions that oscillates in the same way using a shallow network with four hidden units, and generally, for $N \geq 3$ hidden units it's possible to make a function that oscillates back and forth N + 1 times.

Problem 4.10

Consider a deep neural network with a single input, a single output, and K hidden layers, each of which contains D hidden units. Show that this network will have a total of $3D + 1 + (K-1)D(D+1)$ parameters.

For a deep neural network with D hidden units, each hidden unit has the form:

$$D_n = \mathrm{a}[\theta_{0n} + \theta_{1n}x]$$

Meaning that within the hidden layers, where there are $K - 1$ connections, and each hidden unit is connected to every hidden unit in the previous layer, resulting in D^2 weights and an additional D biases, or $(K-1)D^2 + (K-1)D$ weights and biases across the hidden layers. This simplifies to $(K-1)D(D+1)$ parameters.

Additionally, each of the D units in the first layer receives 2 parameters from the single input, adding $2D$ parameters. Finally, the output receives D weights from the final layer plus a bias

term, $D + 1$.

In total this gives $2D + (K - 1)D(D + 1) + D + 1 = 3D + 1 + (K - 1)D(D + 1)$ parameters.

Problem 4.11

Consider two neural networks that map a scalar input x to a scalar output y. The first network is shallow and has $D = 95$ hidden units. The second is deep and has $K = 10$ layers, each containing $D = 5$ hidden units. How many parameters does each network have? How many linear regions can each network make? Which would run faster?

The shallow network has $D + D + D + 1$ (input, hidden layer, output) parameters, or $3D + 1$ total parameters. Therefore with $D = 95$ there are 286 parameters. This network is capable of creating $D + 1$ or 96 linear regions.

For the deep network, as shown above, there are $3D + 1 + (K - 1)D(D + 1)$ total parameters. For $K = 10$ and $D = 5$ this yields a total of 286 parameters. This network is capable of creating $(D + 1)^K$ linear regions, or $(5 + 1)^{10} = 60,466,176$ linear regions.

In principle, the shallow network will be faster to run on modern hardware as the computation is more parallel.

$\mathbb{F}\mathrm{reya}$