神经网络预测

优化问题

神经网络可视为一个黑箱，即知道输入和输出，而中间隐藏层的操作全部封装在黑箱中。用数学表达就是如下形式的映射

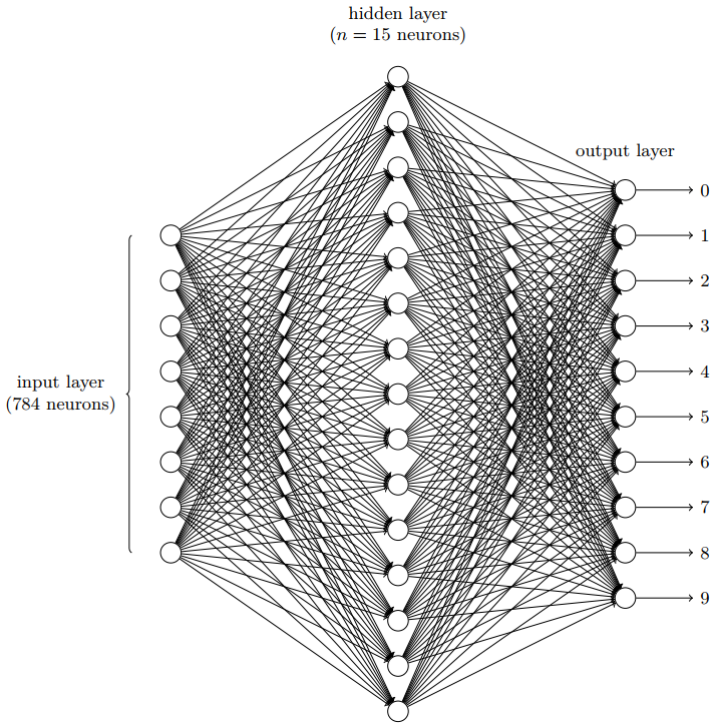
$$y = y(x;\theta).$$

这里，x 是输入，y 是输出，而 θ 是神经网络的参数，即 $wx+b$ 中的权重 w 和偏置 b。

- 当给定参数 θ , 神经网络函数就是确定的，从而任意给定输入，就可得到一个输出。
- 神经网络的训练就是通过一组已知的输入和输出 $\{(x,y)\}$ 确定网络的参数 θ ，它可视为数学中的拟合问题。

优化问题

- 在计算机中，许多数学问题都要转化为优化问题求解，神经网络的训练也是如此。
- 正因为如此，在用神经网络求解问题时，你要清楚地告诉别人对应的优化问题是什么。
- 神经网络优化问题的目标函数常称为损失函数。



现在给出手写数字分类对应的优化问题。

- 给定一个手写数字的输入 x，即 784 维的向量，设其**真实输出为** $y = y(x)$。注意输出是 10 维向量，例如 $y(x) = (0,0,1,0,0,0,0,0,0,0)^T$ 表示数字 2。
- 对训练数据 x，设神经网络给出的输出为 $a = a(x)$。
- 为了确定神经网络中的权重 w 和偏置 b，我们可以最小化如下的损失函数：

$$C(w,b) = \frac{1}{2n}\sum_x \|a(x) - y(x)\|^2, \tag{1}$$

其中的 n 是训练时的输入数据个数。注意，$a(x) := a^L(x)$ 实际上应该为 $a(x; w, b)$，它是神经网络函数。

优化问题的求解

- 神经网络的参数确定归结为一个优化问题，即找到合适的参数，使得损失函数最小。
- 神经网络的优化问题是高维优化问题，常用梯度下降法求解。

在考虑确定神经网络参数之前，我们先随机给定这些参数，然后按照神经网络箭头的顺序给出输出 $a(x)$。这种计算过程称为前向传播。

前向传播

我们先随机初始化神经网络的参数

```
sizes = [784, 15, 10];  % number of neurons on three layers
%% Initialize weights and biases
w2 = randn(sizes([2,1])); % weights from 1-layer to 2-layer
w3 = randn(sizes([3,2]));
b2 = randn(sizes(2),1);   % biases on 2-layer
b3 = randn(sizes(3),1);
```

为了更加清楚，我们假设隐藏层只有一层，从而权重矩阵只有从第 1 层到第 2 层的 w^2 和第 2 层到第 3 层的 w^3. 类似偏置只有 b^2 和 b^3. 要特别注意权重矩阵的阶。

假设 `mini_batch_x` 是输入数据，前向传播如下计算

```
% feedforward
a1 = mini_batch_x;
z2 = w2*a1 + b2;
a2 = sigmoid(z2);
z3 = w3*a2 + b3;
a3 = sigmoid(z3);
```

上面计算出的是样本 `mini_batch_x` 中所有 x 的结果。feedforward 的这几行语句就是进行网络预测（假设权重和偏置已经训练好）。激活函数取为 sigmoid 函数，它及其导数的定义如下

```
%% Define activation functions
sigmoid = @(z) 1./(1+exp(-z));
sigmoid_prime = @(z) sigmoid(z).*(1-sigmoid(z));
```