Deep Learning

Theoretical Exercises – Week 5 – Chapter 5

Exercises on the book "Deep Learning" written by Ian Goodfellow, Yoshua Bengio, and Aaron Courville.
Exercises and solutions by T. Méndez and G. Schuster

FS 2024

1 Exercises on Machine Learning Basics

1. Given is a set of samples $\{x^{(1)}..., x^{(m)}\}$ that are independently and identically distributed according to a uniform distribution on the interval $[-0.8, 1.2]$. In the last series of exercises it was shown that the sample mean

$$\hat{\mu}_m = \frac{1}{m}\sum_{i=1}^{m} x^{(i)}$$

is an unbiased estimator and has an expected value of

$$\mathbb{E}[\hat{\mu}_m] = \mu = 0.2.$$

Now check if the estimator also is consistent (variance goes to zero as the number of samples goes to infinity.).

Solution:
The samples are distributed according to the distribution

$$p(x^{(i)}) = \begin{cases} \frac{1}{2}, & -0.8 \leq x^{(i)} \leq 1.2 \\ 0, & \text{otherwise.} \end{cases}$$

The mean of the samples is

$$\mu = \frac{a+b}{2} = \frac{(-0.8)+1.2}{2} = 0.2$$

and the variance is

$$\mathrm{Var}\left(x^{(i)}\right) = \frac{(b-a)^2}{12} = \frac{(1.2-(-0.8))^2}{12} = \frac{1}{3}.$$

To determine whether the sample mean is a consistent estimator, the variance of $\hat{\mu}_m$ has to be calculated:

$$\begin{aligned}
\mathrm{Var}(\hat{\mu}_m) &= \mathrm{Var}\left(\frac{1}{m}\sum_{i=1}^{m} x^{(i)}\right) \\
&= \frac{1}{m^2}\sum_{i=1}^{m} \mathrm{Var}\left(x^{(i)}\right) \\
&= \frac{1}{m^2}\sum_{i=1}^{m} \frac{1}{3} \\
&= \frac{1}{m^2}\cdot\frac{m}{3} \\
&= \frac{1}{3m}.
\end{aligned}$$

Since the variance goes to zero as the number of samples m goes to infinity, it is a consistent estimator.

2. Given is one samples $\{x^{(1)}\}$ of a Gaussian distribution with a variance of $\sigma^2 = 1$ and a unknown mean value.

   (a) Determine the maximum likelihood estimation of the mean.

   (b) Repeat the exercise with a set of two samples $\{x^{(1)}, x^{(2)}\}$.

Solution:

To make a maximum likelihood estimation, a parameterized model distribution is required. The model distribution used here is a Gaussian distribution

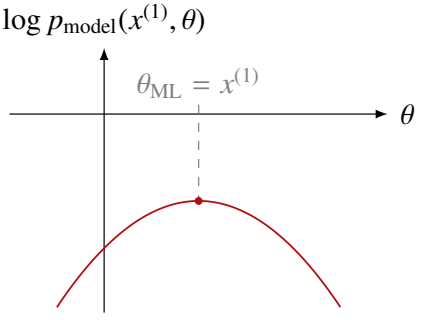
$$p_{\text{model}}(x^{(i)}, \theta) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\theta)^2}{2\sigma^2}\right),$$

with the mean value being the parameter θ to be determined.

(a) If there is only one sample, the function that has to be maximized is

$$\log\left(p_{\text{model}}(x^{(1)}, \theta)\right) = -\frac{1}{2}\log\left(2\pi\sigma^2\right) - \frac{(x^{(1)}-\theta)^2}{2\sigma^2}. \tag{1.1}$$

Graphically the solution can already be found as $\theta_{\text{ML}} = x^{(1)}$.



In order to find the solution mathematically, function (1.1) has to be derived and set to zero

$$\frac{\partial}{\partial\theta}\log\left(p_{\text{model}}(x^{(1)},\theta)\right)=\frac{x^{(1)}-\theta}{\sigma^2}=0,$$

which as well results in the maximum likelihood estimation of $\theta_{\text{ML}}=x^{(1)}$.

(b) If there are two samples, the function that has to be maximized is

$$\sum_{i=1}^{2}\log\left(p_{\text{model}}(x^{(i)},\theta)\right)=\sum_{i=1}^{2}-\frac{1}{2}\log\left(2\pi\sigma^2\right)-\frac{(x^{(i)}-\theta)^2}{2\sigma^2}. \tag{1.2}$$

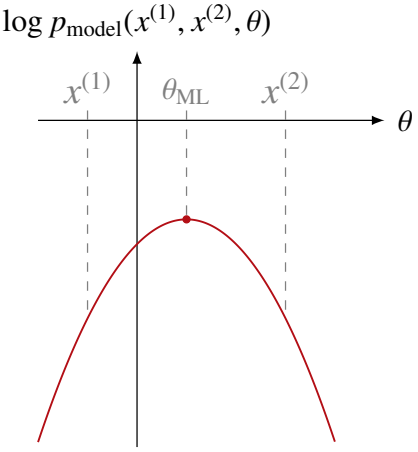
Again, equation (1.2) has to be derived and set to zero

$$\frac{\partial}{\partial\theta}\sum_{i=1}^{2}\log\left(p_{\text{model}}(x^{(i)},\theta)\right)=\sum_{i=1}^{2}\frac{x^{(i)}-\theta}{\sigma^2}=0,$$

which results in the maximum likelihood estimation of

$$\theta_{\text{ML}}=\frac{x^{(1)}+x^{(2)}}{2}.$$

This is nothing else than the sample mean. The solution could also be found graphically.



3. Why is the cross-entropy minimized when training a neural network?

Solution:
Minimizing the cross-entropy is equivalent to minimizing the Kullback–Leibler divergence which again is equivalent to maximizing the likelihood estimation. Thus, by minimizing the cross-entropy, the weights (the parameters) are set so that the model (the neural network) most likely generated the training data. That is why it makes sense to minimize the cross-entropy.

4. What is the difference between gradient descent and stochastic gradient descent and why is stochastic gradient descent used for deep learning?

Solution:
The gradient descent algorithm uses the entire data set to estimate the gradient, whereas

the stochastic gradient descent algorithm only uses a part of the data set (a minibatch of at most a few hundred examples) for the estimation of the gradient.

Since the training set size can grow to billions of examples in deep learning, the time required for a single gradient step would be prohibitively long, and thus the learning process would be massively slowed down.

5. Why is it ”allowed” to use stochastic gradient descent instead of gradient descent (why does it still work)?

   Solution:
   The insight of stochastic gradient descent is that the gradient is an expectation. The expectation also can be approximated using a smaller set of samples. The estimate will be less accurate (have more variance), but on average it is correct and roughly indicates the correct direction in which the loss function becomes smaller. Overall, many inaccurate steps lead to a better result than one exact step.