Neural Networks – Exercise Sheet 4

Anna Krasilnikova 2562668
Mirza Misbah Mubeen Baig 2571567
Shahzain Mehboob 2571564

Exercise 4.1 – *Gradient Descent and Newton Method*

In the optimization setting, Gradient Descent and Newton's Method are two commonly used iterative methods for finding a solution. Let's say we want to minimize a function $f: \mathbb{R}^2 \longrightarrow \mathbb{R}$ which is defined as $f(x) = x_1^2 - 3x_1 + x_2^2 - x_1x_2$. The starting point $x^{(0)} = [1, 1]^T$.

a) *Use Gradient Descent with step size (also known as learning rate) $\varepsilon = 0.5$ to minimize the function f. The iteration should stop if the L2-norm of the gradient at the current point is less than 0.2. Show your intermediate steps.*

To apply the gradient descent algorithm let's start with computing the gradient:

$$\nabla f(x_1, x_2) = \begin{bmatrix} \dfrac{\partial f(x_1, x_2)}{\partial x_1} \\ \dfrac{\partial f(x_1, x_2)}{\partial x_2} \end{bmatrix} = \begin{bmatrix} 2x_1 - 3 - x_2 \\ 2x_2 - x_1 \end{bmatrix}$$

Since the starting point for x is given as $x^{(0)} = [1, 1]^T$, we can compute

$$(x_1^{(1)}, x_2^{(1)} = \left(x^{(0)}, y^{(0)}\right) \in \nabla f\left(x^{(0)}, y^{(0)}\right)$$

Step 1:

$$= \begin{bmatrix} 1 \\ 1 \end{bmatrix} - 0.5 \begin{bmatrix} -2 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ 0.5 \end{bmatrix} = \|2 - 0.5\|^2 = 2.25$$

Iteration 2:

$$= \begin{bmatrix} 2 \\ 0.5 \end{bmatrix} - 0.5 \begin{bmatrix} 0.5 \\ -1 \end{bmatrix} = \begin{bmatrix} 1.75 \\ 0 \end{bmatrix} = \|1.75 - 0\|^2 = 3.0625$$

Iteration 3:

$$= \begin{bmatrix} 1.75 \\ 0 \end{bmatrix} - 0.5 \begin{bmatrix} 0.5 \\ -1.75 \end{bmatrix} = \begin{bmatrix} 1.5 \\ 0.875 \end{bmatrix} = \|1.5 - 0.875\|^2 = 0.39$$

Iteration 4:

$$= \begin{bmatrix} 1.5 \\ 0.875 \end{bmatrix} - 0.5 \begin{bmatrix} -0.875 \\ 0.25 \end{bmatrix} = \begin{bmatrix} 1.06 \\ 0.75 \end{bmatrix} = \|1.06 - 0.75\|^2 = \mathbf{0.09}$$

b) *Starting from the same point x, use Newton's Method to find the solution. Is the solution you get the global minimum? (Why/Why not)*

To apply Newton Method, we first compute the gradient and the Hessian.

$$\nabla f(x_1, x_2) == \begin{bmatrix} 2x_1 - 3 - x_2 \\ 2x_2 - x_1 \end{bmatrix}$$

$$Hf(x_1, x_2) = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

The starting point for x is given as:

$$x^{(0)} = [1, 1]^T$$

$$Hf(x_1, x_2) = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \Rightarrow Hf(x_1, x_2) = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix}$$

Iteration 1:

$$Hf(x_1, x_2)^{-1} \nabla f\left(x^{(0)}\right) = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix} \begin{bmatrix} 2x_1 - 3 - x_2 \\ 2x_2 - x_1 \end{bmatrix} = \begin{bmatrix} 0.5 \cdot (2x_1 - 3 - x_2) + 0 \\ 0 + 0.5 \cdot (2x_2 - x_1) \end{bmatrix}$$

$$x_1 = \begin{bmatrix} 0.5 \cdot (-2) \\ 0.5 \cdot (1) \end{bmatrix} = \begin{bmatrix} -1 \\ 0.5 \end{bmatrix}$$

Iteration 2:

$$\begin{bmatrix} 0.5 \cdot (2 \cdot (-1) - 3 - 0.5) \\ 0.5 \cdot (2 \cdot (0.5) - (-1)) \end{bmatrix} = \begin{bmatrix} -2.75 \\ 1 \end{bmatrix}$$

Iteration 3:

$$\begin{bmatrix} 0.5 \cdot (2 \cdot (-2.75) - 3 - 1) \\ 0.5 \cdot (2 \cdot (1) - (-2.75)) \end{bmatrix} = \begin{bmatrix} -4.75 \\ 2.375 \end{bmatrix}$$

Iteration 4:

$$\begin{bmatrix} 0.5 \cdot (2 \cdot (-4.75) - 3 - 2.375) \\ 0.5 \cdot (2 \cdot (2.375) - (-4.75)) \end{bmatrix} = \begin{bmatrix} -7.4375 \\ 4.75 \end{bmatrix}$$

No, the solution is not the global minimum because with every iteration we are going further away from zero.

c) *We learned from the previous Exercise 3.3 that a bad step size can lead to non-local minimum for Gradient Descent. But are we guaranteed to converge to a local minimum when using Newton's Method since it does not have an explicit step size? What is the implicit step size used in Newton's Method?*

Yes, Newton Method guarantees to converge to a local minimum. Implicit step size used in Newton Method simply means to start close to the root, where "close" is relative to the function we are dealing with.

d) *Is Newton's Method always applicable if the function f is twice continuously differentiable? Argue with the help of the function $f(x) = 2x^3 - 5x$ at $x = 0$.*

Yes, Newton Method is always applicable if the function is twice continuously differentiable in order to find the root of the derivative.

$$f(x) = 2x^3 - 5x$$
$$f'(x) = 6x - 5$$
$$f''(x) = 6$$

Therefore,

$$x = x^{(0)} - \frac{f'(x^{(0)})}{f''(x^{(0)})} = 0 - \left(\frac{-5}{6}\right) = \frac{5}{6} = 0.83$$

Exercise 4.2 – Overfitting

Overfitting happens when the model capacity is too high and there is no proper regularization applied. Figure 1 and Figure 2 show two different classification boundaries for a binary classification problem, where the blue points and the red points represent the training data consisting out of two classes. Please answer the following questions.

a) *Which classification boundary correspond to the overfitting and the underfitting, respectively?*

The boundaries for the classification in this case are Bias and Variance (for under- and overfitting resp.). When the variance is too large – we have overfitting of the training data, but decreasing the variance, generalization error is increasing, thus bias becomes too large.

b) *Explain the terms overfitting, underfitting, and model capacity with the help of Figure 1 and Figure 2.*

Overfitting – occurs when a model tries to predict a trend in data that is too noisy. In general, it's just a result of overly complex model with too many parameters. Such model will be inaccurate because the trend does not reflect the reality of data. The main problem that model is memorizing existing data points, instead of trying to predict how unseen data points would be. Essentially, overfitting comes from an excessive number of training points.

Underfitting – the opposite to the overfitting. Occurs when the model cannot capture the underlying structure of data. That means that such model is missing some terms, or parameters which should appear in the correctly specified model, e.g. fitting a linear model to non-linear data. Essentially, would lead to the poor prediction.

Model capacity of the model – roughly describes the complexity of the pattern that neural network is able to learn. One of easiest ways to estimate it – by counting the number of model's parameters – increasing the number of parameters would increase the capacity – meaning it could fit more complex functions. However, increasing capacity could potentially lead to the overfitting.
So, the black line on the figure 2 represents overfitted model. This black line follows the training data to precisely, it's too depending on that data, and thus the probability of having a higher error rate on new unseen data, compared to the regularized model.
On the other hand, figure 1 represents underfitting – it's clearly seen that the line doesn't cover all point shown in the graph, so basically the attempt to fit non-linear data to linear function (too simple to explain the variance).

c) *What happens to the training error and validation error when a model overfits? Explain.*

In overfitting model validation error is high, while training error is low. It means, that classifier's error measured on the training data is small, but the classification error measured on new instances, unseen during training, is high. This happens because of high variety of parameters in the overfitting model.

Exercise 4.3 – *Regularization*

The solution of the linear regression problem given on slide 14 of chap5 is also known as normal equation. However, the inverse of $x^T x$, where x is the design matrix, may not exist. Therefore, you may have to solve a linear system rather than applying the closed form solution directly. We saw in the lecture a modified linear regression problem, where we minimize a loss function $J(w) = MSE_{train} + \lambda w^T w$ for $w, \lambda > 0$. This modified version is known as *ridge regression*. One of the purposes of ridge regression is to obtain a unique solution for the normal equation when $x^T x$ is not invertible.

a) *Given the same setting of linear regression from the lecture (see slide 14, chap5), derive the closed form solution for the ridge regression.*

We can start by setting the gradient to zero[1]. Now we can rewrite the gradient of the total cost as following:

$\nabla cost(w) = -2X^T(y - Xw) + 2\lambda w$, which is essentially equivalent to the one below with the respect to the properties of the Identity Matrix:

$$\nabla cost(w) = -2X^T(y - Xw) + 2\lambda I w$$

As it's stated above, we set to the zero this equivalent form of the gradient with the respect to $\widehat{w}$:

$$\nabla cost(\widehat{w}) = -2X^T(y - X\widehat{w}) + 2\lambda I\widehat{w} = 0$$
$$-X^T y + X^T X\widehat{w} + \lambda I\widehat{w} = 0$$
$$X^T X\widehat{w} + \lambda I\widehat{w} = X^T y$$
$$(X^T X + \lambda I)\widehat{w} = X^T y$$

Again, by using the property of the Identity Matrix we can obtain the closed-form solution:

$$\widehat{w} = (X^T X + \lambda I)^{-1} X^T y$$

b) *Argue for the uniqueness of the closed form solution for the ridge regression, i.e. argue why the linear system has unique solution.*

The maximum likelihood estimator is not always unique, e.g. if X doesn't have the full rank, than $X^T X$ is not invertible number of β values maximize the likelihood. However, this problem doesn't occur in ridge regression. Overall, we can state following:

[1] The following derivation is based on materials from https://ru.coursera.org/

The quantity $n^{-1}X^TX + \lambda I$ is always invertible provided that $\lambda > 0$ for any design matrix X; thus there is always a unique solution $\widehat{w}$.

c) Argue that the solution you got from a) is indeed a global minimizer.

Ridge regression $\widehat{w} = (X^TX + \lambda I)^{-1}X^Ty$ adds some term to the objective function, usually after standardizing all variables in order to put them on a common footing. Hence, asking to minimize for some non-negative constant λ. It is the whole of squares of the residuals in addition to a various of the total of squares of the coefficients themselves, making it clear that it has a global minimum. Because $\lambda \geq 0$, it has a positive square root $v^2 = \lambda$.