Exercise 3 - **Assignment 1**

Deep Learning Lab
Due: Thursday 28 October 2021, 10:00 pm (time in Lugano)

October 6, 2021

1 Polynominal Regression [94/100 points]

Consider the polynomial p given by

$$p(x) = x^3 + 2x^2 - 4x - 8 = \sum_{i=0}^{3} \mathbf{w}_i^* x^i,$$

where $\mathbf{w}^* = [-8, -4, 2, 1]^T$.

Consider also an i.i.d. dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$, where $y_i = p(x_i) + \epsilon_i$, and each ϵ_i is drawn from a normal distribution with mean zero and standard deviation $\sigma = 0.5$.

Now if we assume that the vector $\mathbf{w}^*$ is unknown, **linear regression** (see slides Sec. 2.1) could estimate it given the dataset $\mathcal{D}$. For that, we need to represent the original dataset $\mathcal{D}$ as $\mathcal{D}' = \{(\mathbf{X}_i, y_i)\}_{i=1}^N$, where $\mathbf{X}_i = [1, x_i, x_i^2, x_i^3]^T$. Such data generation is partially illustrated in the code presented below.

Listing 1: Polynomial regression dataset generation (incomplete).

```
def create_dataset(w_star, x_range, sample_size, sigma, seed=None):
    """Incomplete documentation."""
    random_state = np.random.RandomState(seed)

    x = random_state.uniform(x_range[0], x_range[1], (sample_size))
    X = np.zeros((sample_size, w_star.shape[0]))
    for i in range(sample_size):
        X[i, 0] = 1.
        for j in range(1, w_star.shape[0]):
            X[i, j] = ? # Incomplete

    y = X.dot(w_star)
    if sigma > 0:
        y += random_state.normal(0.0, sigma, sample_size)

    return X, y
```

1. (10 pts) Complete the code above.

2. (10 pts) Use the completed code and the following parameters to generate training and validation data points:

   - Each x_i should be in the interval $[-3, 2]$.
   - $\mathbf{w}^* = [-8, -4, 2, 1]^T$.
   - Use $\sigma = 0.5$.

- Use a sample of size 100 created with a seed of 0 for training.
- Use sample of size 100 created with a seed of 1 for validation.

3. (5 pts) Create a 2D scatterplot (using x and y) of the generated training and validation dataset.
4. (4 pts) Search for the documentation of `torch.nn.Linear`. Notice the flag `bias`. Explain what it does using equations. Should that flag set to `True` or `False` for this problem? Explain.
5. (20 pts) Adapt the linear regression in the 1D case presented in the lecture (at the end of Sec. 2.1) to perform polynomial regression using the generated training dataset $\mathcal{D}'$. More specifically, find and report an estimate of $\mathbf{w}^* = [-8, -4, 2, 1]^T$ supposing that such a vector is unknown.
6. (10 pts) Find and report a suitable **learning rate** and **number of iterations** for gradient descent.
7. (10 pts) Plot the training and validation losses as a function of the gradient descent iterations.
8. (5 pts) Plot the polynomial defined by $\mathbf{w}^*$ and the polynomial defined by your estimate $\hat{\mathbf{w}}$.
9. (10 pts) Report and explain what happens when the training dataset is reduced to 50, 10, and 5 observations (while keeping the number of data points for the validation set **unchanged**). Illustrate your observations with some plots of your choice.
10. (10 pts) Report and explain what happens when σ is increased to 2, 4, and 8. Illustrate your observations with some plots of your choice.
11. (10 pts, extra) **Bonus**: Reduce your training dataset to 10 observations (keep the validation data size unchanged), and compare fitting a polynomial of degree three (as before) with fitting a polynomial of degree four (which does not match the underlying polynomial p). Plot the resulting polynomials and document the validation loss. What do you observe?

2 Questions [6/100 points]

Provide a **brief and concise** answer (**max. 4 sentences**) to the following questions **in your own words** and add them to your report. You can find more information in the relevant subsections of chapter 5,8 of the Deep Learning Book.

1. (2 pts) What is the difference between a local and a global minimum?
2. (2 pts) What does it mean if your model is overfitting/underfitting?
3. (2 pts) How can you mitigate underfitting/overfitting? (one example each is enough)

NB: We are expecting concise answers. Long answers including reformulations ("in other words...") or long examples will not be graded.

Submission

You should deliver the following by the deadline stipulated on iCorsi3:

- **Report**: a single *pdf* file that clearly and concisely provides evidence that you have accomplished each of the tasks listed above. The report should not contain source code (not even snippets). Instead, if absolutely necessary, briefly mention which functions were used to accomplish a task.
- **Source code**: a single Python script that could be easily adapted to accomplish each of the tasks listed above. The source code will be read superficially and checked for plagiarism. Therefore, if a task is accomplished but not documented in the report, it will be considered missing. Note: Jupyter notebook files are not accepted.

Please carefully read the instructions above to prepare your submission. Failure to stick to these rules may result in reduction of points.