Neural Networks Homework 4

Sayeh Jarollahi (7073520, saja00006@stud.uni-saarland.de)
Mahsa Amani (7064006, maam00002@stud.uni-saarland.de)

November 18, 2024

Exercise 4.1

Proof. a)

Bias refers to the extent to which the predicted values (averaged across all data sets) deviate from the actual desired outputs. Essentially, it highlights the error introduced when a simpler model is used for approximation. A higher bias indicates a larger gap between the estimated values and the true labels, leading to increased error. This situation often arises when the model lacks sufficient capacity to adequately represent the complexity of the training data—for instance, employing linear models to analyze quadratic relationships—resulting in underfitting, characterized by poor performance on both training and test datasets.

Variance quantifies the sensitivity of predictions and indicates the extent to which the predictions for individual data sets deviate from their mean. A higher variance suggests that the model is capturing more noise from the data, which hampers its ability to generalize effectively to new, unseen data. This situation typically arises when the model has excessive capacity, resulting in overfitting—manifested by low training error but high test error (for instance, employing a high-degree polynomial to fit a simple linear relationship).

There is a trade-off between bias and variance; as we increase model's complexity, bias decreases (a better fit to data) and variance increases (fit varies more with data). In most cases, you can only decrease one of them at the expense of the other.

b)

	Bias	Variance	Justification
Increase the width of hidden layers in a neural network	$\downarrow$	$\uparrow$	Wider layers increase model capacity, reducing bias but allowing more flexibility to fit the noise in training data
Increase the polynomial degree n in estimator $\hat{y} = b + \sum_{i=1}^{n} w_i x^i$	$\downarrow$	$\uparrow$	Higher degree polynomials can fit more complex patterns, reducing bias but increasing sensitivity to data fluctuations
Train a classification model on less data	NEI	$\uparrow$	Less data for sure increases variance as model becomes more sensitive to each training data, but Bias effect depends on data quality and model
Add regularization to the training objective of a classifier	$\uparrow$	$\downarrow$	Regularization constrains model complexity to limit overfitting, resulting in bias increase and variance decrease
Remove only some non-support vectors in a SVM	$-$	$-$	Non-support vectors have no affect on SVM decision boundary, so removing them doesn't effect bias or variance
Decrease hypothesis space of the model	$\uparrow$	$\downarrow$	Smaller hypothesis space means having a simpler model with less flexibility, which will increase bias but reduce variance

c)

For calculating mean squared error (MSE) on test data we have:

$$\text{MSE}(y_{\text{test}}, \hat{f}_n) = \frac{1}{n} \sum_{i=1}^{n} (y_{\text{test}}^i - \hat{f}_n(x_{\text{test}}^i))^2 \tag{1}$$

We can replace the $\frac{1}{n} \sum_{i=1}^{n}$ with $\mathbb{E}$, as they infer the same meaning:

$$\text{MSE}(y_{\text{test}}, \hat{f}_n) = \mathbb{E}[(y_{\text{test}} - \hat{f}_n(x_{\text{test}}))^2] \tag{2}$$

By substituting y_{test} with $f(x_{\text{test}}) + \epsilon_{\text{test}}$:

$$\text{MSE}(y_{\text{test}}, \hat{f}_n) = \mathbb{E}[(f(x_{\text{test}}) + \epsilon_{\text{test}} - \hat{f}_n(x_{\text{test}}))^2] \tag{3}$$

By expanding the squared term:

$$\text{MSE}(y_{\text{test}}, \hat{f}_n) = \mathbb{E}[(f(x_{\text{test}}) - \hat{f}_n(x_{\text{test}}))^2] + 2\mathbb{E}[(f(x_{\text{test}}) - \hat{f}_n(x_{\text{test}}))\epsilon_{\text{test}}] + \mathbb{E}[\epsilon_{\text{test}}^2] \tag{4}$$

1. The first term $\mathbb{E}[(f(x_{\text{test}}) - \hat{f}_n(x_{\text{test}}))^2]$ represents the MSE between the true function $f(x_{\text{test}})$ and the estimator $\hat{f}_n(x_{\text{test}})$.
2. The second term $2\mathbb{E}[(f(x_{\text{test}}) - \hat{f}_n(x_{\text{test}}))\epsilon_{\text{test}}]$, since ϵ_{test} is independent of $\hat{f}_n(x_{\text{test}})$ and $\mathbb{E}[\epsilon_{\text{test}}] = 0$, becomes 0.
3. The third term $\mathbb{E}[\epsilon_{\text{test}}^2]$ based on the definition of ϵ_{test}, is equal to its variance, τ^2.

Thus:

$$\text{MSE}(y_{\text{test}}, \hat{f}_n) = \mathbb{E}[(f(x_{\text{test}}) - \hat{f}_n(x_{\text{test}}))^2] + \tau^2 \tag{5}$$

Using bias and variance definitions:

$$\text{Bias}(\hat{f}_n(x_{\text{test}})) = f(x_{\text{test}}) - \mathbb{E}[\hat{f}_n(x_{\text{test}})] \tag{6}$$

$$\text{Var}(\hat{f}_n(x_{\text{test}})) = \mathbb{E}[(\hat{f}_n(x_{\text{test}}) - \mathbb{E}[\hat{f}_n(x_{\text{test}})])^2] \tag{7}$$

We can rewrite $\mathbb{E}[(f(x_{\text{test}}) - \hat{f}_n(x_{\text{test}}))^2]$:

$$\mathbb{E}[(f(x_{\text{test}}) - \hat{f}_n(x_{\text{test}}))^2] = (f(x_{\text{test}}) - \mathbb{E}[(\hat{f}_n(x_{\text{test}})])^2 + \mathbb{E}[(f(x_{\text{test}}) - \mathbb{E}[(\hat{f}_n(x_{\text{test}})])^2] \tag{8}$$

$$\mathbb{E}[(f(x_{\text{test}}) - \hat{f}_n(x_{\text{test}}))^2] = \text{Bias}^2(\hat{f}_n(x_{\text{test}})) + \text{Var}(\hat{f}_n(x_{\text{test}})) \tag{9}$$

Finally, we have:

$$\text{MSE}(y_{\text{test}}, \hat{f}_n) = \text{Bias}^2(\hat{f}_n(x_{\text{test}})) + \text{Var}(\hat{f}_n(x_{\text{test}})) + \tau^2 \tag{10}$$

□

Exercise 4.2

Proof. a)

0.0.1 i

A potential issue with the product of equation 1 in the homework is that the product is numerically unstable. As we know, all the values of $p(x(i);\theta)$ are between zero and one. The product of these values can be too small to be calculated and stored.

We can convert this product to a sum by knowing the fact that $log(ab) = log(a) + log(b)$:

$$\prod_{i=1}^{m} p_{\text{model}}(x^{(i)};\theta) \quad \xrightarrow{\log} \quad \sum_{i=1}^{m} \log p_{\text{model}}(x^{(i)};\theta).$$

This transformation preserves the argmax because logarithm is a monotonic function. it means that if $a > b$ then $log(a) > log(b)$ for sure. Hence, by maximizing the value of log of sums we can reach the argmax of products.

0.0.2 ii

We know that MLE of θ is defined as:

$$\theta_{\text{ML}} = \arg\max_{\theta} \mathbb{E}_{x\sim\hat{p}} \log p_{\text{model}}(x;\theta)$$

Now we start opening the equation for D_{KL} :

$$D_{KL}(\hat{p}_{data}||p_{model}) = \sum_{i=1}^{n} \hat{p}_{data} log(\hat{p}_{data}) - \hat{p}_{data} log(p_{model}).$$

For minimizing the equation above, the first term in the summation is useless since it does not have θ. So, we only have to minimize the second part of the summation which can be written as:

$$-\mathbb{E}_{x\sim p_{\hat{data}}}[log(p_{model})(x)]$$

Minimizing the above equation means maximizing $\mathbb{E}_{x\sim p_{\hat{data}}}[log(p_{model})(x)]$ which is the exact goal of MLE.
b) We define(based on the book mentioned in the question):

$$p(y \mid x) = \mathcal{N}(y; \hat{y}(x;w), \sigma^2)$$

where σ is set by the user and $\hat{y}(x;w)$ is the function that gives the prediction of the gausian. Now we apply the log-likelihood which will be:

$$\sum_{i=1}^{m} \log p\left(y^{(i)} \mid x^{(i)};\theta\right) = \sum_{i=1}^{m} \log[\frac{1}{\sqrt{2\pi\sigma^2}} exp^{\frac{-||\hat{y}-y||^2}{2\sigma^2}}]$$

By opening the log we get:

$$\sum_{i=1}^{m} -log(\sigma\sqrt{2\pi}) - \frac{||\hat{y}^{(i)} - y^{(i)}||^2}{2\sigma^2} = -mlog(\sigma\sqrt{2\pi}) - \frac{||\hat{y}^{(i)} - y^{(i)}||^2}{2\sigma^2}$$

We have to maximize the expression above, which means maximizing only the second term, because the first term is a constant.
Now we can compare the final maximization problem with MSE problem which is as follows:

$$\text{MSE}_{\text{train}} = \frac{1}{m}\sum_{i=1}^{m} \|\hat{y}^{(i)} - y^{(i)}\|^2$$

c)
Small training data set: When training dataset is small, the model can face overfiting to the existing data. Since the model does not see various data, it overfits to the ones it see. The MLE can become highly sensative to the fluctuations in data and fit itself with those fluctuations.
In this situation the estimator can be biased. This happens because it cannot fully capture the underlying true distribution of the data. The model can depend on some specific samples alot which causes bias. Also, this model can have high variance. The model is fitting in every noise that exists in the training sample and therefore, small changes to the input can cause large change in the output.
Large training data set: In this case, overfitting to the data is not an issue. As m increases, MLE will converge towards the true parameters. Therefore there is no concern for underfitting or overfitting. Additionally, with large amount of data, MLE is unbiased. Since it will converge to the true values. This can be proved by the law of large numbers. Variance will also decrease as m increases. With more data, the estimator becomes more stable, and its value will change less with small changes in the training data. □

Exercise 4.3

Proof. We know that MAP estimator is:

$$\theta_{MAP} = argmaxp(\theta|D) = argmaxp(D|\theta)p(\theta)$$

Since elements of dataset are independent from each other, we can write:

$$p(D|\theta) = \prod_{i=1}^{n} p(y_i \mid x_i, \theta)$$

Also, we know that weights come from a normal distribution and each y is coming from a normal distribution with mean $\theta^T x$ and variance σ^2.
Therefore we can write :

$$y_i \mid x_i, \theta \sim \mathcal{N}(\theta^\top x_i, \sigma^2)$$

Consequently, for the first part of our probabilities we can say:

$$p(D \mid \theta) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y_i - \theta^\top x_i)^2}{2\sigma^2}\right)$$

Also, for the prior we know that:

$$p(\theta) = \frac{1}{(2\pi)^{1/2}} \exp\left(-\frac{1}{2}\|\theta\|^2\right)$$

We get log from the MAP right-side equation because we know log does not change the argmax value. For that, we have:

$$\log p(D \mid \theta) = -\frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - \theta^\top x_i)^2$$

Since we are finding the argmax regarded to θ, we do not need the first term in the above equation. So we can say:

$$\log p(D \mid \theta) \propto -\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - \theta^\top x_i)^2$$

Also, we have:

$$\log p(\theta) \propto -\frac{1}{2}\|\theta\|^2$$

We can now say that:

$$\log p(\theta \mid D) \propto -\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - \theta^\top x_i)^2 - \frac{1}{2}\|\theta\|^2$$

By multiplying the equation by $-\sigma^2$ we have:

$$-\log p(\theta \mid D) \propto \sum_{i=1}^{n}(y_i - \theta^\top x_i)^2 + \lambda\|\theta\|^2$$

We can see that minimizing $L(\theta)$ can be written as a MAP procedure. □