

1 Machine Learning Theory

Throughout this section the theory of machine learning that will be used in this thesis is described and explained.

1.1 Motivation

It can be difficult for humans to recognize certain patterns and trends in data. This becomes more difficult the greater the quantity of the data is, which is becoming more and more common with the rapidly growing topic of *Big Data*. For this reason, computers are often used instead of humans to recognize patterns and trends in the data by analyzing the data, which is what is called *Machine Learning*. In this thesis, we will use machine learning in section **MANGLER REFERENCE** to develop a model to estimate the 2D pose of a single human in an image. Later, in section **MANGLER REFERENCE**, we will use machine learning to improve our understanding of the model.

1.2 Machine Learning Paradigms

Machine learning is usually split into the following three paradigms

- *Supervised learning* where the data consists of features and labels. By analyzing the data the algorithm learns to predict the labels given the features [5]. Supervised learning is further split into *classification* and *regression*. If the value of each label is limited, then the task is a classification task. If the value of each label is not limited, then the task is a regression task.
- *Unsupervised learning* where the data only consists of features. The algorithm then learns properties of the data, without any provided labels [5].
- *Reinforcement learning* where the algorithm learns to perform the action in a given environment that yields the highest reward [1].

In this thesis we will make use of supervised learning when developing our model for pose estimation. Later, unsupervised learning is used when we explore our developed model.

1.3 Evaluation of Machine Learning Models

When developing a machine learning model it is important to know how trustworthy the developed model is. This is usually done by testing how good the model is at generalizing unseen data, which is done by making use of *evaluation metrics*.

1.3.1 Splitting the dataset

When developing a machine learning model, the data needs to both create the model, but also to evaluate the model. For the evaluation of the model, one of the two following techniques is usually used

1. *Cross validation* where the data is split into K random non-overlapping chunks of equal size. The model is then trained for K rounds on $K - 1$ of the chunks, where the last chunk is used for evaluating the model. After each round the parameters of the model is reset to ensure one round does not affect another round. After the K rounds the average loss of the K rounds is the loss of the model [4].

2. *Train-validation-test* where the data is split into 3 random non-overlapping chunks. The training dataset is then used for training the model and the validation dataset is used for evaluating the model as it is being developed - this often means, that the *hyperparameters*, the parameters that are not possible to fit from the data, are being tweaked to yield the best validation loss. Lastly, the testing dataset is used as a final evaluation of the model to yield an unbiased evaluation of the model. Once the testing dataset has been used it can no longer be used for evaluating the data, as this ensure an unbiased evaluation [3].

Throughout this thesis the train-validation-test technique will be used over cross validation for evaluating the developed models, as cross validation is better suited for smaller datasets, as the runtime is much greater than the runtime of the train-validation-test technique.

1.3.2 Evaluation Metrics for Supervised Machine Learning (Loss Functions)

When we have trained a model, we need to somehow evaluate how well the model performs on unseen data. This is usually done by making use of evaluation metrics or *loss functions*. There are many different loss functions, each with their own advantages and disadvantages. One of the most common loss functions for regression is the *Mean Squared Error (MSE)*, defined as

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

where y_i is the true value of the i th observation and \hat{y}_i is the estimated value of the i th observation. Thus, *MSE* measures the average squared difference between the true observation and the estimated observation. The aim of a model is thus to make the *MSE* as small as possible [2].

1.4 Neural Networks

1.4.1 Termonology

Epoch

Mini-batch

Activation function

Optimizer

1.4.2 Convolutional Neural Networks

1.4.3 Stacked Hourglass

The Residual Modules

The Hourglass

The Stacked Hourglass

1.5 Principal Components Analysis and K-means Clustering

1.5.1 Principal Components Analysis (PCA)

1.5.2 K-means Clustering