Segment 1 Neural networks

- **NLP introduction (along with different types)**

Introduction to natural language processing

Natural language processing (NLP) is a method for a machine to process or, with some vague definition, understand written language. The purpose of extracting the meaning, is to create a computational representations handler which can take actions based on the output of the NLP. To better identify which properties is needed for the methods used, it is a good idea to first point out the complexity of natural language.

Introducing the ambiguity problem of natural language

In this thesis, spelling errors are not handled. The biggest problem is therefore its ambiguity. An example could be the sentence,

"She was very impressed by the duck."

This is a typical example, of a sentence which could mean two very different things depending on, if "duck" is a verb or a noun. In one case, it could be a woman being impressed by a physical downwards maneuver but in another, it's a woman being impressed by a special duck. In this case the context of the story is vital, as it's impossible to derive a non-interpretable meaning. This is a case of one word having two different meanings. Another ambiguity source for natural language, is the use of pronouns ("substitute nouns" make footnote in finished thesis) in a sentence, where multiple candidate nouns potentially could be its substitute. An example of this sentence type could be,

"The pizza was taken out of the oven and then it was put on the table."

In this case, it is both grammatically correct if the pizza was put on the table, or if the oven was put on the table. There are probably more ambiguous points which could be explained in detail. But the core idea is that contextual understanding of language is essential for a language model. A direct way of inserting contextual understanding into a mathematical formular, is by using large databases filled with sentences which can be used to train a language model. This approach is what neural networks use.

Introduction to general neural networks

The core idea behind a neural network, is to pass information into a high dimensional function, which can identify a specific categorisation. Its output is all the possible categorizations with weighted probabilities.

There are non-neural network related methods for Natural language processing, but these will not be used or described for this thesis.

- **Token classification job (specifying task: (whats input, whats output)**

The idea of different categorization tasks given NLP

Since NLP using neural networks, is usually a categorisation task, then the processing and the output for NLP neural networks are usually defined in that way. In this thesis the idea of the robot writing back, is not explored, therefore it is only considered using text analysis methods. The text analyses methods typically consist of three categories. It's either a sentence categorization task, a word categorization task, or a letter categorization task.

- Sentence categorization tasks involve sentiment analysis, question-answering.. I think?, and some more I don't know.
- Word categorization consists of named entity recognition, Part Of Speech tagging (POS tagging) and zero shot classification.
- Letter categorization I don't know any solely per-letter categorization techniques, but I do know that it can be used for the same as word categorizations
-

Flesing out the categorizations (filtering out the sentence classifier methods)

Sentence categorization is fundamentally a different class than the word and letter categorization, while letter categorization is a niche method for word categorization.

Sentence categorization usually involves problems where the solution can be described as a single statement, based on the whole sentence. Sentiment analysis is classifying a sentence into either having a positive or negative connotative meaning, as this is not going to be relevant for this thesis, it will not be explored any further.

Exploring word and letter classifiers – Shouldn't write this until I get a better understanding of letter classifiers

Word categorization is the main method for text analysis when complex meaning must be derived from the sentence.

Classifier techniques that will be used for this thesis.

For this thesis the natural language processing will be used, to extract robot command actions from sentences. Therefore, it is assumed, that good techniques for such problems, would POS-tagging, named entity recognition, zero shot classification maybe more, I stated all word classifiers I know as of right now.

As more than one technique was stated, then it is also given that several techniques will be used, such to maximize information extracted from the sentences.

- **NLP using NN**

Might need an introduction too how neural networks

Explaining in greater detail why neural networks are a good matchup for text processing

The idea behind using neural networks for natural language processing, lies in its ability to discern and categorize seemingly ambiguous tasks. Such as deciding whether the given picture is a pink cloud, or candy floss. The core idea, and the reason to its success, is its ability to handle problems in a multi-dimensional space beyond human understanding, in which it becomes capable of identifying and extracting important information from input, which it uses to create its output. Using a digital image as an example again for simplicity, then each pixel consisting of an RGB value, could possibly represent an input dimension which is fed into the neural network. Using fully connected hidden layers with their arbitrary input dimensionality, then the pixels are processed with the equivalence of a very large mathematical function such to extrapolate important key information which is used to analyse the objects within the digital image in some way. Here it is assumed that a normal feed forward layer is used to process an image, which is usually not the case, as it is too computationally slow. Normally something called a convolutional neural network is used for images, as it can achieve greater results with less computational power.

The connection between text processing in relation to image processing, is the use of the input, which is either words or pixels, which is translated into a high dimensional space where the problem is processed using trained hidden layers. For text, each word is first passed through a word embedding, which is an independent high dimensional space, which is a trained numerical representation of the word itself. This word embedding space is trained, such that words that mean the same will have a short Euclidean distance from each other. The words pink and blue, will be in the same cluster for example, and the words car and bicycle will also be close but presumably not equally as close as the other words. Using this word embedding, each word is then assigned a vector of numeric values, which can be processed by a neural network.

- **Transformers**

Introducing what transformers are

Transformers is a pioneering neural network design, specifically designed to handle language in text format. Its success can be measured, by the dominating number of transformer-based models in the top tiering list of language models in the GLUE benchmarking tests. There are several ideas behind the transformer model which makes it superior to the previously dominated design which is a recurrent neural network. Some of these properties are that transformers are highly parallelizable enabling faster computational efficiency, and another property, is that transformers have a constant big-O time complexity regarding self-attention. In comparison to recurrent neural networks where words values must “travel” through the sentence until it arrives at its connected word.

Going into what transformers do (not sure how deep I should go into this subject)

Explaining the preprocessing (word embedding and positional encoding)

Walking through the multiheaded attention layers

Going through the encoder decoder structure

Introducing pretraining

- **BERT**
- **Token classification finetuning for BERT**
- **Model evaluation? (own trained model and pretrained model)**