# The Attention Mechanism – Review

D. Gueorguiev 1/7/2023

## Introductory Notes

In many problems that involve the processing of natural language, the elements composing the source text are characterized by having each a different relevance to the task at hand. For instance, in aspect-based sentiment analysis, cue words, such as “good” or “bad”, could be relevant to some aspects under consideration but not to others. In machine translation, some words in the source text could be irrelevant in the translation of the next word. In a visual question-answering task, background pixels could be irrelevant in answering a question regarding an object in the foreground but relevant to questions regarding the scenery.

The attention mechanism is a part of a neural architecture that enables dynamically to select relevant features in the input data which in NLP is typically a sequence of textual elements. The idea behind attention is to compute a weight distribution on the input sequence, assigning higher values to more relevant elements.

//TODO: finish this paragraph which corresponds to Section II of [1]

## Appendix

### Probability Density Estimation

We consider the problem of modelling a probability density function , given a finite number of data points drawn from that density function. The methods for probability density estimation are used to build classifier systems by considering each of the classes in turn and estimating the corresponding class-conditional densities by making use of the fact that each data point is labelled according to its class. These densities can then be used with Bayes theorem to find the posterior probabilities corresponding to a new measurement of , which can in turn be used to make a classification of **.**

We consider three alternative approaches to density estimation. The first of these involves *parametric* methods in which a specific functional form for the density model is assumed. This contains a number of parameters which are then optimized by fitting the model to the data set.

#### Parametric Methods

//TODO: finish this paragraph and the corresponding Appendix section which corresponds to Chapter 2 of [3]

### Bi-directional Neural Networks

*Problem Statement*

Consider a (time) sequence of input data vectors

and a sequence of corresponding output data vectors

with neighboring data-pairs in time being statistically independent. Given time sequences and as training data, the aim is to learn the rules to predict output data given the input data. Inputs and outputs, can, in general, be continuous and/or categorical variables. When the outputs are continuous, we have *regression problem* at hand and when they are categorical (class labels), the problem is known as a *classification problem*. In general we talk about *prediction problem* which includes regression and classification.

1. *Unimodal Regression*

With unimodal regression or function approximation, the components of the output vectors are continuous variables. The network parameters are estimated to maximize some predefined objective criterion e.g. maximize the likelihood of the output data. When the distribution of the errors between the desired and the estimated output vectors is assumed to be Gaussian with zero mean and a fixed global data-dependent variance, the likelihood criterion reduces to the Euclidean distance measure between the desired and the estimated output vectors or the *mean-squared-error criterion*, which has to be minimized during training.

//TODO: finish this paragraph and the corresponding Appendix section which corresponds to Section B of [2]

## Bibliography

[1] [Attention in Natural Language Processing, Andrea Galassi, Marco Lippi, Paolo Torroni, 2019](https://github.com/dimitarpg13/transformers_intro/blob/main/articles_and_books/AttentionInNaturalLanguageProcessing.pdf)

[2] [Bidirectional Recurrent Neural Networks, Mike Schuster, Kuldip Paliwal, IEEE Transactions on Signal Processing, 1997](https://github.com/dimitarpg13/transformers_intro/blob/main/articles_and_books/Bidirectional_Recurrent_Neural_Networks_Schuster_Paliwal_1997.pdf)

[3] [Neural Networks for Pattern Recognition, Christopher M. Bishop, 1995](https://github.com/dimitarpg13/transformers_intro/blob/main/articles_and_books/Neural_Networks_for_Pattern_Recognition-Bishop_1995.pdf)

[4] [Neural Machine Translation by Jointly Learning To Align and Translate, Dzmitry Bahdanau, K. Cho, Yoshua Bengio, 2016](https://github.com/dimitarpg13/transformers_intro/blob/main/articles_and_books/NeuralMachineTranslationByJointlyLearningToAlignAndTranslateBahdanau2015.pdf)

[5] [Recurrent Continuous Translation Models, Nal Kalchbrenner, Phil Blunsom, Oxford U., 2013](https://github.com/dimitarpg13/transformers_intro/blob/main/articles_and_books/Recurrent_continuous_translation_models_Kalchbrenner_Blunsson_OxfordU_2013.pdf)

[6] [On The Properties of Neural Machine Translation: Encoder-Decoder Approaches, K. Cho, B. van Merrienboer, Dzmitry Bahdanau, 2014](https://github.com/dimitarpg13/transformers_intro/blob/main/articles_and_books/OnthePropertiesOfNeuralMachineTranslationEncoderDecoderApproaches.pdf)

[7] [Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation, K. Cho, B. van Merrienboer, C. Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, Yoshua Bengio, 2014](https://github.com/dimitarpg13/transformers_intro/blob/main/articles_and_books/LearningPhraseRepresentationsUsingRNNEncoderDecoderForStatisticalMachineTranslation.pdf)

[8] [Autoencoders, Dor Bank, Noam Koenigstein, Raja Giryes, 2021](https://github.com/dimitarpg13/transformers_intro/blob/main/articles_and_books/Autoencoders.pdf)

[9] [Attention and Augmented Recurrent Neural Networks, Chris Olah, Shan Carter, Google Brain blog, 2016](https://github.com/dimitarpg13/transformers_intro/blob/main/articles_and_books/AttentionandAugmentedRecurrentNeuralNetworks.pdf)

[10] [Attention is All You Need, Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit et al, NIPS, 2017](https://github.com/dimitarpg13/transformers_intro/blob/main/articles_and_books/Attention-is-all-you-need-NIPS-2017.pdf)

[11] [Transforming Auto-encoders, G.E. Hinton, A. Krizhevsky, S.D. Wang, U of Toronto, 2010](https://github.com/dimitarpg13/transformers_intro/blob/main/articles_and_books/TransformingAutoencodersHinton.pdf)

[12] [Sequence to Sequence Learning with Neural Networks, Ilya Sutskever, Oriol Vinyals, Quoc V. Le, Google Research, 2014](https://github.com/dimitarpg13/transformers_intro/blob/main/articles_and_books/SequencetoSequenceLearningwithNeuralNetworksSutsekver2014.pdf)

[13] [Generating Sequences With Recurrent Neural Networks, Alex Graves, U of Toronto, 2014](https://github.com/dimitarpg13/transformers_intro/blob/main/articles_and_books/Generating_Sequences_With_Recurrent_Neural_Networks_Graves_2014.pdf)

[14] [A Tutorial on Training Recurrent Neural Networks, covering BPPT, RTRL, EKF and the “echo state network” approach, Herbert Jaeger, Fraunhofer Institute for Autonomous Intelligent Systems, 2003](https://github.com/dimitarpg13/transformers_intro/blob/main/articles_and_books/TutorialOnRNNAndBPTTJaeger2002.pdf)

[15] [Understanding LSTM: a Tutorial into Long Short-Term Memory Recurrent Neural Networks, Ralf C. Staudemeyer, Eric Rothstein Morris, 2019](https://github.com/dimitarpg13/transformers_intro/blob/main/articles_and_books/TutorialOnLongShortTermMemory2019.pdf)

[16] [Long Short-Term Memory, Sepp Hochreiter, Juergen Schmidhuber, Neural Computation 9(8), 1997](https://github.com/dimitarpg13/transformers_intro/blob/main/articles_and_books/LongShortTermMemory.pdf)

[17] [Sequence Modeling With Neural Networks (Part 1): Language & Seq2Seq, Nathan Lintz’s blog, 2016](https://github.com/dimitarpg13/transformers_intro/blob/main/articles_and_books/SequenceModelingWithNeuralNetworksPart1_Seq2Seq.pdf)

[18] [Sequence Modeling With Neural Networks (Part 2): Attention Models, Nathan Lintz’s blog, 2016](https://github.com/dimitarpg13/transformers_intro/blob/main/articles_and_books/SequenceModelingwithNeuralNetworksPart2_AttentionModels.pdf)