1 Deep Learning Theory

The following section covers the most important background theory for the experiments in Section **??**. This includes an introduction to various types of neural networks, as well as an introduction to the optimization of such networks.

1.1 Feedforward Neural Networks

Feedforward neural networks are the most basic type of neural networks. The aim of a feedforward neural network is to approximate some function f^*, by defining a mapping $\boldsymbol{y} = f(\boldsymbol{x}; \boldsymbol{\theta})$ and learning the parameters $\boldsymbol{\theta}$, that results in the best approximation of f^*. These models are called **feedforward** because there are no **feedback** connections in which the outputs of the model are fed back into itself. Instead, information flows through the function being evaluated from $\boldsymbol{x}$, through the intermediate computations used to define f, and finally to the output $\boldsymbol{y}$. Feedforward neural networks generally consists of multiple **layers**, arranged in a chain structure, with each layer being a function of the layer that preceded it [2].

1.2 Fully-connected Layers

The most simple type of layer found in a feeforward neural network is the **fully-connected layer**. The fully-connected layer usually consists of some learnable parameter matrix $\boldsymbol{W}$ and learnable parameter vector $\boldsymbol{b}$, as well as a non-linear **activation function** g (which will be covered further in Section 1.6.1). In this case, the i'th layer is defined as [2]

$$\boldsymbol{h}^{(i)} = \begin{cases} g^{(i)}\left(\boldsymbol{W}^{(i)\top}\boldsymbol{h}^{(i)} + \boldsymbol{b}^{(i)}\right) & \text{if } i > 1 \\ g^{(1)}\left(\boldsymbol{W}^{(1)\top}\boldsymbol{x} + \boldsymbol{b}^{(1)}\right) & \text{if } i = 1 \end{cases}. \tag{1}$$

1.3 Convolutional Layer

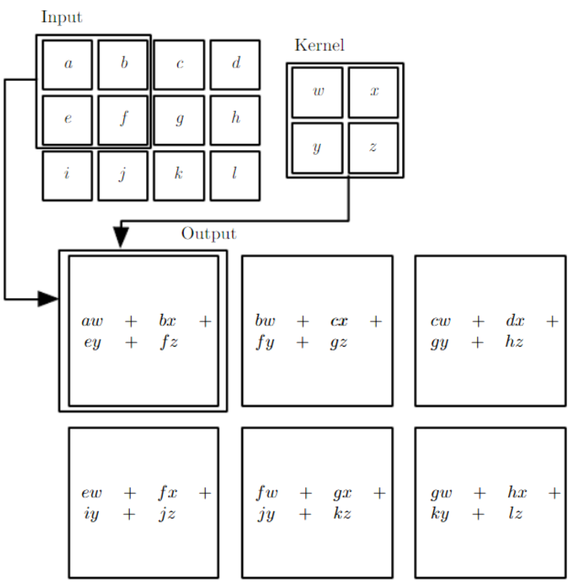

Figure 1: An example of applying a 2d kernel on an input [2].

A **convolutional layer** is a specialized kind of feedforward layer, usually used in analysis of time-series or image data [2]. If a network has at least one convolutional layer, it is called a **Convolutional neural network (CNN)**.

The convolutional layer consists of a set of **kernels**, each to be applied to the entire input vector, where each kernel is a learnable parameter matrix $k \times k$ [3]. Each kernel is applied on the input to produce a **feature map**. The kernels are applied to the input by "sliding" over the input (where the step size is called **stride**). Each $k \times k$ grid of the input is then used to compute the dot-product between the grid and each kernel, which is then placed in the corresponding feature map of each kernel, as visualized in Figure 1. [1]. To control the dimensions of the output, one might **pad** the sides with a constant value. Commonly, zero is used as the padding-value.

As seen in Figure 1, each kernel produces a linear combination of all pixel values in a neighbourhood defined by the size of the kernel. Thus, unlike a fully-connected layer, a convolutional layer captures the high correlation between a pixel and its neighbours. Further, by limiting the size of the kernel, the network will use much fewer parameters, than if a fully-connected layer would be used instead [2].

1.4 Recurrent Neural Networks

1.4.1 Long Short-Term Memory Unit

- Convolutional LSTM

1.4.2 Gated Recurrent Unit

- Convolutional GRU

1.5 Transformer

1.6 Training a Neural Network

1.6.1 Activation function