is the fact that they actually try to identify statistical patterns in the data. Thus the aim is to find a meaningful representation of the given data by projections, translations, rotations, nonlinear transformations or any other method and to apply it then to new input data.

- Deep Learning: Deep learning, as a sub field of machine learning, focuses on the progressive learning of several levels of abstraction which are increasingly meaningful representations of the data. One can think of the method as a multistage information distillation process where in every single process step a more meaningful representation of the data is obtained. The term deep in deep learning is a reference to the multiple layers which store the abstract representations of the input data in such an algorithm. The number of layers that contribute to a meaningful representation of the data is called depth. Although there is a considerable amount of learning involved, models with several hundred layers are quite common, depending on the type of problem at hand. The concept of the neural network is a reference to the field of neurobiology and the neurons that are connected in different ways in the human brain. Although neuronal networks are not models of the human brain, it is surprising what amazing results can be achieved with such a simple idea and a sufficiently large amount of data.

After a brief classification of the terms, the next section describes the basic structure and functionality of neural networks.

5.1. Fundamentals of Neural Networks

In the previous section the functionality of neuronal networks was already described in a rather abstract way as a multi-stage information distillation process. After this high level explanation, the focus is now on pointing out which individual components make up a simple neural network and how they interact. Figure (5.2) shows a simplified representation of the individual components required to build one of the most basic neural networks possible. Starting point for explaining the learning process of a neural network is a data set consisting of both the input and the associated output data. If the cash flows of insurance policies are to be simulated, the individual policy parameters can be used as input data and the associated cash flows as output data. These two exogenous quantities are then processed in several steps.

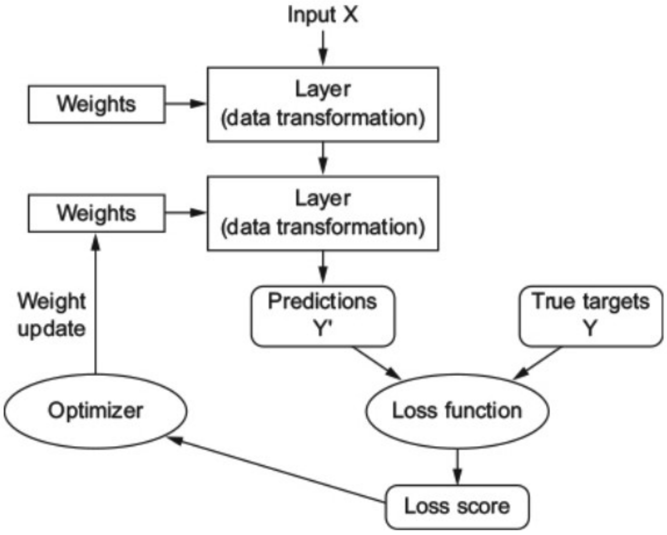

Figure 5.2.: Simplified representation of a neural network from [**Allaire2018**]

Step 1: The process is initiated by providing the input data `Input X` in a suitably format to the first layer.

Step 2: This first layer then performs a data transformation based on some weights associated with that specific layer. In the first cycle those weights are randomly initialized.

Step 3: The transformed output from the first layer is then passed to the next layer and serves as its input. This layer also performs a data transformation based on weights which are again initialized randomly for the first cycle.

Step 4: After the data has passed through the last data transformation layer of the model the transformed output values are used as `Predictions Y'`.

Step 5: A predefined function, called the `Loss function`, measures the quality of the network's output by comparing the `Predictions Y'` and the `True targets Y`. The result is the `loss score`, which is used as a feedback signal to adjust the `weights` associated with the layers.

Step 6: The task of the `Optimizer` is to take the `loss score` as an input and update the `weights` in such a way that the `loss score` is lowered.

Remark 5.1. *The choice of the correct loss function is of great importance. If the loss function does not correctly reflect an improvement of the model, the neural net will inevitably drift in the wrong direction.*

Remark 5.2. *As already explained in the previous chapters, in the specific case of cash flow matching the sum of the squared residuals would be suitable as a loss score.*

The above steps describe a single learning cycle for a neural network. Of course, passing the input data through the net once is not sufficient to find combinations of weights that would lead to good prediction results at all. So what learning means, is to find a set of values for all weights such that the input is mapped correctly to the output for as many different input combinations as possible. This learning is achieved by grouping the input data into so-called batches and repeatedly passing those batches through the network. Associated with a batch is the batch size which defines the number of training samples sent through the neural network before the weights are adjusted by the optimizer.

Remark 5.3. *Three different batch size approaches can be used to learn the weights in a network.*

- *Batch Gradient Descent: Batch size is equal to the size of the training samples.*
- *Stochastic Gradient Descent: Batch size is equal to one.*
- *Mini-Batch Gradient Descent: Batch size is bigger than one but less than the size of the training samples.*

Depending on the design of the network there can be millions of weights that need to be learned. Adjusting those millions of weights based on the feedback signal is by far the most computing intensive part of training a network. Due to the fact that the learning process is based on a large number of similar matrix operations it is possible to use highly parallelized algorithms. The availability of relatively affordable hardware that can efficiently handle such parallelizable tasks is besides the existence of large amount of data another major reason why neural networks took off in recent years.

Remark 5.4. *Compared to a pure CPU system, graphics cards with their massively parallel architecture are predestined for handling such training tasks and can therefore massively reduce the time needed to train a network.*

After providing an overview of all components used in a neural network and presenting the idea how a network learns, the next section focuses on how a single layer is organized.

5.1.1. One neuron