Even BCI users who previously could not gain control with classic offline calibration were successful within a single session. A similar online adaptation approach was described in [26]. It was tested with 12 novice users of which 10 were successful in controlling the BCI. Notably, the users stressed their preference for early feedback in the motor imagery BCI training, which had not been available in the control setting with conventional offline calibration. BCIs based on auditory or music imagery that combine feedback with an online adaptation strategy have not yet been described in the literature.

1.3 Deep Learning

Deep learning is a sub-field of machine learning which in turn *"is concerned with the question of how to construct computer programs that automatically improve with experience"* [67]. According to [12], the term can refer to either training artificial neural networks (ANNs) or probabilistic graphical models with high complexity – usually with multiple hidden layers although "deep" can also describe the model complexity in a more general sense like the "credit assignment path" as introduced in the comprehensive historical review of deep learning provided in [81]. A high-level introduction to deep learning is given in [57]. Typical machine learning tasks comprise *supervised learning* such as classification or regression where, given labeled training data, models are learned that can predict labels for unseen data, and *unsupervised learning* such as clustering where models are learned that capture the structure of the data.

Traditionally, machine learning techniques are applied on data represented by a predetermined and often laboriously hand-crafted set of features. Deep learning, in contrast, allows to also learn suitable feature representations along with the actual learning task using a general-purpose learning procedure such as backprogation. Thanks to this key advantage, deep learning *"will have many more successes in the near future because it requires very little engineering by hand, so it can easily take advantage of increases in the amount of available computation and data"* [57]. According to a popular perspective, deep networks learn representations of their (raw) input whereby each layer adds another level of complexity – much like structures of real neurons in the brain. A recent review of representation learning is provided in [12].

Individual layers can be pre-trained using unlabeled data before being stacked to form a complex network that can finally be fine-tuned. For supervised learning tasks, this final training step may involve a much smaller set of labeled data. The unsupervised layer-wise pre-training improves performance on test data [11] – especially for small datasets where the risk of overfitting to the training set is high [12]. Unsupervised pre-training also allows to make use of additional unlabeled data in supervised learning tasks where labeled data is scarce or costly to obtain. Popular deep learning techniques (most relevant to this project) comprise:

1. **Auto-Encoders** have become a popular technique in the field of deep learning over the course of the last decade. The general idea is to train a neural network to reconstruct its inputs and limit the internal hidden representation to make this a non-trivial task – for instance, through a structural bottleneck or regularization of weights or activations. Additionally, the inputs can be corrupted by adding random noise which can result in more robust features [100]. This approach has been successfully applied for learning compressed feature representations – usually during an unsupervised pre-training phase – in many domains such as for learning high-level image features [56], coding speech spectrograms [21] or sentiment analysis [87]. Auto-encoding can be combined with the ideas described in the following yielding variants such as Convolutional Auto-Encoders (CAEs) [63] or recursive auto-encoders [87].
2. **Convolutional Neural Networks (CNNs)** contain one or more convolutional layers. In such layers, the input is convolved with multiple kernels. Each kernel is a weight matrix that is moved along the input dimensions with some step size (called stride) and for each position, an output value is computed. This computation also involves the usual non-linear transform (activation function) common to ANNs. Afterwards, neighboring output values are usually pooled which commonly means an aggregation to the maximum or mean value. Depending on the stride parameter of the pooling window, the output can optionally be sub-sampled. Conceptually, the convolution operation implements local connectivity and weight sharing as inspired by the visual cortex [27]. Each neuron has a limited receptive field and all neurons share the same weights of their incoming connections for each kernel. The weights are trained with backpropagation like in conventional layers. The pooling step can also be considered as a separate network layer after the convolution although it does not have trainable parameters. An alternative trainable pooling implementations based on a Gaussian has been proposed in [106]. Deep CNNs have been especially successful in the field of computer vision (e.g., [55]). Using convolutional layers in auto-encoders requires to reverse the convolution and pooling operation which results in deconvolution and unpooling respectively [105]. Convolutional layers can also be used in Recurrent Neural Networks (RNNs) as, e.g., described in [75].