OXFORD

Briefings in Bioinformatics, 00(0), 2020, 1–15

doi: 10.1093/bib/bbaa355
Method Review

Deep learning approaches for neural decoding across architectures and recording modalities

Jesse A. Livezey and Joshua I. Glaser

Corresponding authors: Jesse Livezey, Biological Systems and Engineering Division, Lawrence Berkeley National Laboratory, Berkeley, California, United States; Redwood Center for Theoretical Neuroscience, University of California, Berkeley, Berkeley, California, United States. E-mail: jlivezey@lbl.gov; Joshua Glaser, Department of Statistics, Columbia University, New York, United States; Zuckerman Mind Brain Behavior Institute, Columbia University, New York, United States; Center for Theoretical Neuroscience, Columbia University, New York, United States; E-mail: j.glaser@columbia.edu

Abstract

Decoding behavior, perception or cognitive state directly from neural signals is critical for brain–computer interface research and an important tool for systems neuroscience. In the last decade, deep learning has become the state-of-the-art method in many machine learning tasks ranging from speech recognition to image segmentation. The success of deep networks in other domains has led to a new wave of applications in neuroscience. In this article, we review deep learning approaches to neural decoding. We describe the architectures used for extracting useful features from neural recording modalities ranging from spikes to functional magnetic resonance imaging. Furthermore, we explore how deep learning has been leveraged to predict common outputs including movement, speech and vision, with a focus on how pretrained deep networks can be incorporated as priors for complex decoding targets like acoustic speech or images. Deep learning has been shown to be a useful tool for improving the accuracy and flexibility of neural decoding across a wide range of tasks, and we point out areas for future scientific development.

Introduction

Using signals from the brain to make predictions about behavior, perception or cognitive state, i.e. 'neural decoding', is becoming increasingly important within neuroscience and engineering. One common goal of neural decoding is to create brain computer interfaces, where neural signals are used to control an output in real time [1, 2]. This could allow patients with neurological or motor diseases or injuries to, for example, control a robotic arm or cursor on a screen, or produce speech through a synthesizer. Another common goal of neural decoding is to gain a better scientific understanding of the link between neural activity and the outside world. To provide insight, decoding accuracy can be compared across brain regions, cell types, different types of subjects (e.g. with different diseases or genetics) and different experimental conditions [3–11]. In addition, the representations learned by neural decoders can be probed to better understand the structure of neural computation [12–16]. These uses of neural decoding span many different neural recording modalities and span a wide range of behavioral outputs (Figure 1A).

Within the last decade, many researchers have begun to successfully use deep learning approaches for neural decoding. A decoder can be thought of as a function approximator, doing either regression or classification depending on whether the output is a continuous or categorical variable. Given the great successes of deep learning at learning complex functions across many domains [17–26], it is unsurprising that deep learning has become a popular approach in neuroscience. Here, we review the many uses of deep learning for neural decoding. We emphasize how different deep learning architectures can induce biases that can be beneficial when decoding from different neural recording modalities and when decoding different behavioral outputs. We aim to provide a review that is both useful to deep learning researchers looking to understand current neural decoding problems and to neuroscience researchers looking to understand the state-of-the-art in neural decoding.

Jesse A. Livezey is a postdoctoral research scientist in the Neural Systems and Data Science Laboratory at the Lawrence Berkeley National Laboratory. He obtained his PhD in Physics from the University of California, Berkeley. His research interests include applications of machine learning and information theory to neuroscience datasets.

Joshua I. Glaser is a postdoctoral research scientist in the Center for Theoretical Neuroscience and Department of Statistics at Columbia University. He obtained his PhD in Neuroscience from Northwestern University. His research interests include neuroscience, machine learning and statistics.

Submitted: 18 May 2020; **Received (in revised form):** 31 October 2020

Deep learning architectures

At their core, deep learning models share a common structure across architectures: (1) simple components formed from linear operations (typically addition, matrix multiplication or convolution) plus a nonlinear operation (for example, rectification or a sigmoid nonlinearity) and (2) composition of these simple components to form complex, layered architectures [27]. The simplest fully connected neural networks combine matrix multiplication and nonlinearities. These fully connected networks, along with recurrent and convolutional neural networks (described below) are most frequently used in neuroscience. Although more complex deep network layer types, e.g. graph neural networks [28] or networks that use attention mechanisms [29], have been developed, they have not seen much use in neuroscience. In addition, given that datasets in neuroscience typically have limited numbers of trials, shallower neural networks are often used for neural decoding compared with the networks used in common machine learning tasks.

Recurrent neural networks (RNNs) act on a sequence of inputs of potentially varying length, which occurs in neuroscience data (e.g. trials of differing duration). This is unlike a fully connected network, which requires a fixed dimensionality input. In an RNN, the inputs, X_t, are then projected (with weights w_X) into a hidden layer, H_t, which recurrently connects to itself (with weights w_H) across time (Figure 1B)

$$\begin{aligned} \mathrm{H}_{t+1} &= f(w_H \cdot \mathrm{H}_t + w_X \cdot \mathrm{X}_t) \\ \mathrm{Y}_t &= g(w_\mathrm{Y} \cdot H_t) \end{aligned} \quad (1)$$

where $f(\cdot)$ and $g(\cdot)$ are nonlinearities and Y_t is the RNN output. Finally, the hidden layer projects to an output, Y_t, which can itself be a sequence (Figure 1B), or just a single data point. Commonly used RNN architectures like Long short-term memory networks (LSTMs) and Gated recurrent unit networks [22, 27, 30] have multiplicative 'gating' operations in addition to element-wise nonlinearities. Recurrent networks are commonly used for neural decoding as they can flexibly incorporate information across time.

Convolutional neural networks (CNNs) can be trained on input and output data in many different formats. For example, convolutional architectures can take in structured data (one-dimensional time series, two-dimensional images, three-dimensional volumes) of arbitrary size [23, 27, 31, 32]. Input neural data, X, may have one or more channels indexed by c [which may be combined in a filter as in Equation (2) or operated on individually] and temporal or spatial dimensions indexed by $(t, \ldots)$. A convolutional layer has weights with multiple filters (f), which combine across channels and have temporal (or spatial) extent $(T, \ldots)$ followed by a nonlinearity, $g(\cdot)$. For example, for a one-dimensional convolution, the activations in a layer are calculated as

$$h_{t,f} = g\left(\sum_{\tau=0,c}^{T-1} w_{\tau,f,c} \mathrm{X}_{t+\tau,c}\right). \quad (2)$$

The two-dimensional and three-dimensional extensions have more output indices in addition to t for the additional dimensions, and more dimensions are summed over for each filter. The convolutional layers will then learn filters of the corresponding dimensions, in order to extract meaningful local structure (Figure 1C). The convolutional layers are commonly used if there are important features that are translation invariant, as in images. This is done hierarchically, in order to learn filters of varying scales (i.e. varying temporal or spatial frequency content), which is a useful prior for multiscale data, such as images. Next, depending on the output that is being predicted, the convolutional layers are fed into other types of layers to produce the final output (e.g. into fully connected layers to classify an image).

Weight sharing, where the values of some parameters are constrained to be the same, is often used for neural decoding. For instance, the parameters of a convolutional (in time) layer can be made the same for differing input channels or neurons, so that these different inputs are filtered in the same way. For neural decoding, this can be beneficial for learning a shared set of data-driven features for different recording channels (e.g. a relevant frequency pattern for electrocorticography (ECoG) datasets) as an alternative to human-engineered features.

Training a neural decoder uses supervised learning, where the network's parameters are trained to predict target outputs based on the inputs. Recent work has combined supervised deep networks with unsupervised learning techniques, which learn lower dimensional representations that reproduce one data source. One common unsupervised method, generative adversarial networks (GANs) [33, 34], generate an output, e.g. an image, given a vector of noise as input. GANs are trained to produce images that fool a classifier deep network about whether they are real versus generated images. Another method is convolutional autoencoders, which are trained to encode an image into a latent state, and then reconstruct a high fidelity version [35]. These unsupervised methods can produce representations of the decoding input or output that are sometimes more conducive for decoding and can potentially leverage larger datasets for training that are available for neural decoding.

Stages of neural decoding

In order to go from the raw neural signal to the final predicted output (e.g. speech), the neural decoding pipeline can be conceptually broken down into a few components, each of which can incorporate deep learning.

1. *'Preprocessing / Feature Engineering'*. First, the raw neural signals are processed to create features that are beneficial for neural decoding. Sometimes, these features are hand engineered based on previous knowledge, traditionally with the goal of creating features that are most compatible with linear decoders. More recently, supervised feature engineering has been incorporated into deep learning architectures. That is, a more raw form of the input is provided into the neural decoder, and a first stage of the deep network decoder will automatically learn to extract relevant features. Specific neural network architectures can be beneficial for this automatic feature engineering (Figure 2). It is also possible to generate features from the neural data with deep learning [42, 43] in an unsupervised manner and then use those features with simple linear decoders.

2. 'Mapping from features to final (or intermediate) output'. This central part transforms the features to an output representation, and deep learning tools allow this mapping to be a flexible nonlinear function.

3. 'Mapping from intermediate to final output (optional)'. Neural decoding is used to predict many outputs, including movement, speech, vision and more. Sometimes, the output variable will be directly predicted from the neural inputs, e.g. when predicting movement velocities (and thus this stage is

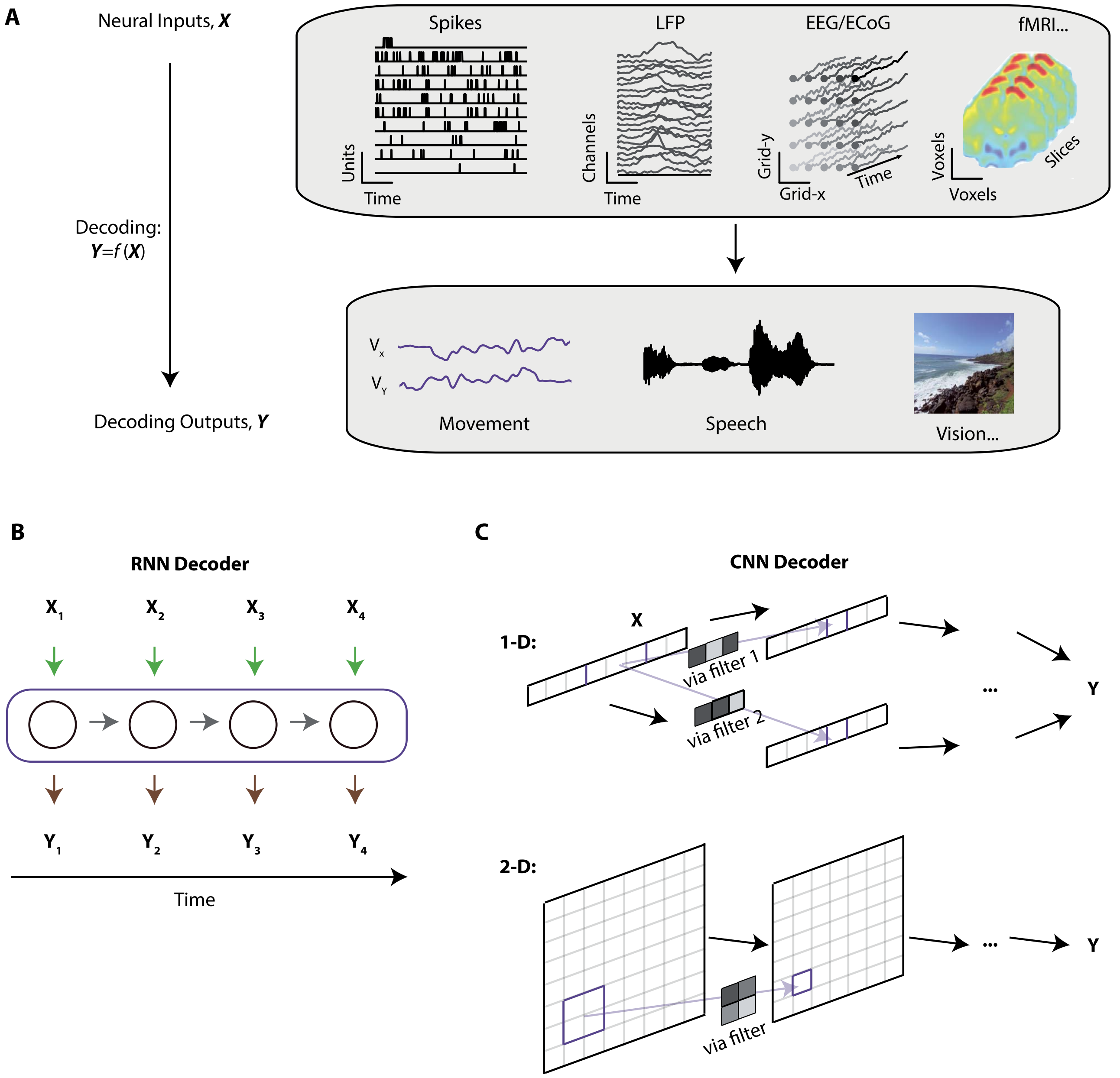

Figure 1. Schematics. **A:** Schematics of neural decoding, which can use many different neural modalities as input (top) and can predict many different outputs (bottom), such as movement velocities (left) [36], a waveform of speech (center) [37] or visual images (right) [38]. Embedded figures are adapted from [39–41]. **B:** A schematic of a standard recurrent neural network (RNN). Each arrow represents a linear transformation followed by a nonlinearity. Arrows of the same color represent the same transformations occurring. The circles representing the hidden layer typically contain many hidden units. More sophisticated versions of RNNs, which include gates that control information flow through various parts of the network, are commonly used. For example, see [27] for a schematic of an LSTM. **C:** A schematic of convolutional neural networks (CNNs). A convolutional transformation takes a learned filter and convolves it with the input, and then passes this through a nonlinearity. As an example of a one-dimensional convolutional transformation (top), as may be the case a single time series, a filter of length 3 (for example) is multiplied element wise with all input segments of length 3 to get the values of the next network layer. In CNNs, typically multiple filters (here, filter 1 and filter 2) are learned within each layer, and the outputs of all filters are combined in a subsequent layer. In our example of a two-dimensional convolutional transformation (bottom), as may be the case for spatial data, a 2×2 filter is multiplied pixel wise with all 2×2 blocks to get the values of the next layer in the network. Convolutions can also occur in three dimensions for neural decoding.

not relevant). Other times, the neural decoder may be trained to predict some intermediate representation, which has a predetermined mapping to the output (Figure 3). For example, a GAN can be trained to generate an image using a small number of latent variables. This mapping from the low-dimensional variables to images can be learned without having to simultaneously record neural activity. Then, to decode an image from neural activity, one can train the neural decoder to predict the latent variables to be fed into the GAN, rather than the entire high-dimensional image. This two-step approach can be especially beneficial when the output data is complex and high dimensional, as is often the case in vision or speech. In effect, the generative model can act as a prior on the underconstrained decoding solution.

We expand on these stages below. First, focusing on neural recordings and how they are transformed into features, and then

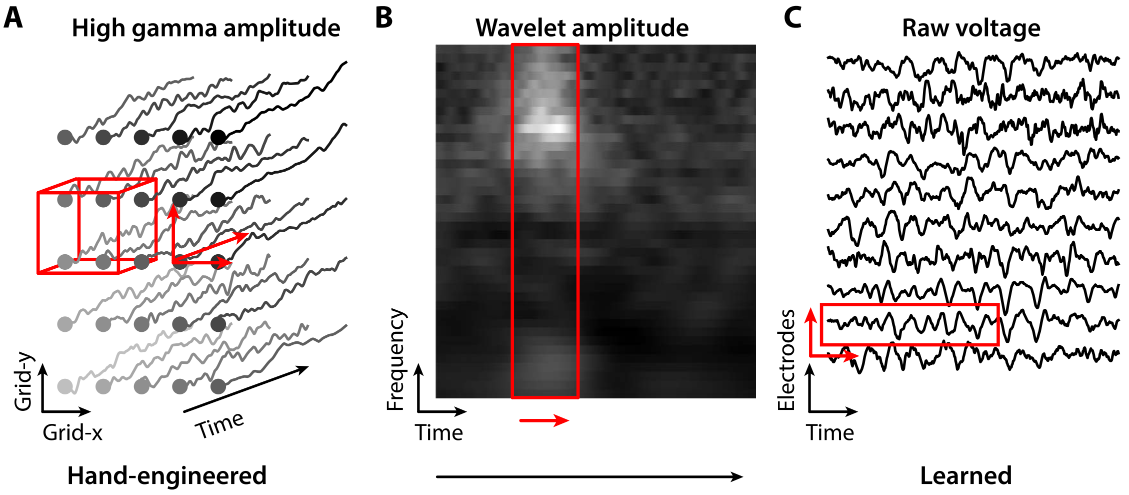

Figure 2. Feature engineering for neural decoding. Relevant features of neural data can be engineered completely by hand (left), automatically learned within a deep neural network (right) or somewhere in between. For all plots, the red box indicates a set of features across time, space or frequency, which will be filtered together by the first layer's convolutional or recurrent window. The red arrows indicate axes along which convolution or recurrence are performed. Sample data from [40]. **A:** High gamma amplitude, which is selected from a large filterbank of features from **B**, is shown spatially laid out in the ECoG grid locations. Deep network filters combine hand-engineered high gamma features across space and time. **B:** Spectrotemporal wavelet decomposition of one channel of the raw data, from **C**, may be used as the input to a deep network. The deep network filter shown combines features across frequency and time and can be shared across channels. **C:** Raw electrical potential recorded using ECoG across channels. The deep network filter shown combines features across time and can be shared across channels.

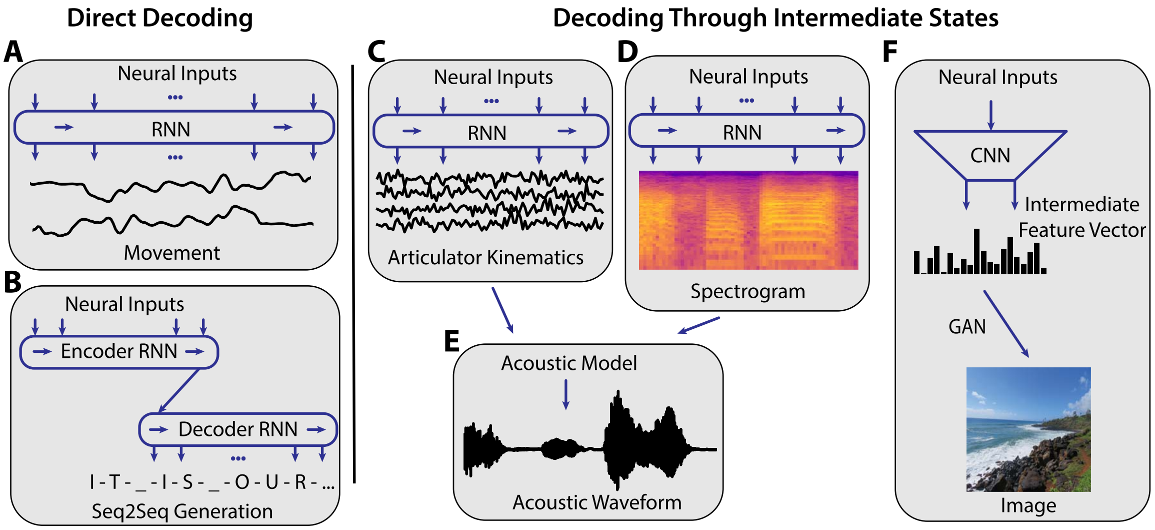

Figure 3. Architectures and outputs of neural decoding. We contrast examples of 'direct decoding', in which the neural network decoder outputs the final desired variable, versus 'decoding through intermediate states', in which the neural network predicts an intermediate variable that subsequently predicts the final desired variable. **A:** Sequential neural data is processed by RNNs that use past context to generate their output (or past and future in bidirectional RNNs). RNN outputs at each timestep can be mapped to behaviors, e.g. movements, measured concurrently, e.g. [36]. **B:** In a Seq2Seq-style RNN, as in [76], the final output of an encoder RNN is used as the input to a decoding RNN, which produces a second sequence of potentially different length, such as the text representation of speech. **C:** RNNs can produce an intermediate state to be used in a second decoding step, such as articulator kinematics (movement of lips, tongue, etc.) [95]. **D:** Intermediate states can often be structured, such as a spectrogram [37]. **E:** Intermediate states can then be fed into an acoustic model such as WaveNet [122] or a speech synthesizer, which produces acoustic waveforms [37, 95]. **F:** Temporal snapshot neural data, e.g. fMRI, can be processed by fully connected networks or CNNs to produce intermediate feature vectors [38]. These feature vectors can be fed into generative image models, e.g. a GAN, to produce a more realistic looking image [38].

focusing on the deep learning methods used for predicting the final outputs of neural decoding.

The inputs of decoding: neural recording modalities and feature engineering

To understand how varying neural network architectures can be preferable for processing different neural signals, it is important to understand the basics of neural recording modalities. These modalities differ in their invasiveness, and their spatial and temporal precision.

Spikes

The most invasive recordings involve inserting electrodes into the brain to record voltages. This allows experimentalists to record spikes, or action potentials, the basic unit of neural signaling. Action potentials are the fast electrical transients that individual neurons use to signal and are triggered when a neuron's membrane potential depolarizes past its threshold. To get binary spiking events, the recorded signals are high-pass filtered and thresholded. They are then often sorted into waveforms attributed to individual neurons, sometimes using deep learning tools [44, 45]. Datasets with spikes are thus binary time courses from all of the recording channels or neurons (Figure 1A). Spikes are more commonly recorded from animal models than humans because of their invasive nature.

For use in neural decoding, spikes are typically first converted into firing rates by determining the number of spikes in time bins, sometimes with additional temporal smoothing. Then, these firing rates are fed into the neural decoder. Commonly, these firing rates are considered the relevant features, and thus additional neural network architectures are not used to extract unknown features from the input.

One form of feature engineering that is used for spike trains, especially when many neurons are recorded, is dimensionality reduction. That is, a lower-dimensional representation of the firing rates is used to predict the outputs. This dimensionality reduction can use a variety of methods, from classical linear methods, e.g. principal component analysis, to deep learning approaches, e.g. autoencoders [43]. This dimensionality reduction step is usually done prior to decoding the output, but it is also possible to incorporate this step into a single neural network decoder [46], so that the learned lower-dimensional representations are particularly relevant for predicting the output. We note that dimensionality reduction is not specific to decoding with spiking activity, but can also be applied to the neural recording modalities described below [46, 47].

Finally, in future research, it might be advantageous to provide a more raw form of spiking as input, rather than binned spike counts. Then, one could use deep learning architectures to do feature engineering. For example, with binary spiking events as input, the best size and temporal placement of time bins could be automatically determined, or even features related to the precise timing of spikes could be learned. It was also recently shown that using the envelope of spiking activity, a continuous signal, followed by feature extraction within a neural network, was able to improve decoding performance [48].

Calcium imaging

Another invasive technique for recording individual neurons' activities is calcium imaging, which uses microscopy to capture images of fluorescent calcium indicators that are sensitive to neurons' spiking activity [49]. These calcium indicators are genetically encoded within neurons in animal models, often within specific neuron types. The raw outputs of calcium imaging are videos: pixels measure fluorescence at the times when, and locations where, neurons are active. Calcium imaging is only used with animal models.

When analyzing calcium imaging data, the videos are typically preprocessed to extract time traces of fluorescences over time for each neuron [50]. Sometimes, additional processing will be done to estimate spiking events from the calcium traces [51]. Deep learning tools exist for both of these processing steps [52, 53]. For decoding, either the fluorescences or the estimated firing rates (via the estimated spike trains) are then commonly used as input. Although it could be possible to develop an end-to-end neural decoder that works with the videos as input, this may prove challenging, given the potential for overfitting with high-dimensional input.

Wideband, local field potentials (LFPs), electroencephalography (EEG) and electrocorticography (ECoG)

The electrode recordings for spikes simultaneously record LFPs, which are the low-pass filtered version (typically below ∼200Hz) of the same recorded voltage. LFPs are thought to be the sum of input activity of local neurons [54]. When all voltage is included across frequency bands, the voltage is generally referred to as wideband activity. Datasets with LFP and wideband are continuous time courses of voltages from all the recording channels (Figure 1A). Note that traditionally, due to the distance between recording electrodes being greater than the spatial precision of recording, spatial relationships between electrodes are not utilized for neural decoding.

Electrical potentials measured from outside of the brain, that is ECoG and EEG, are common neural recording modalities used in humans. ECoG recordings are from grids that record electrical potentials from the surface of the cortex, require surgical implantation and often cover large functional areas of the cortex. EEG is a noninvasive method that records from the surface of the scalp from up to hundreds of spatially distributed channels. Like LFPs, datasets from ECoG and EEG recordings are continuous time courses of electrical potentials across recording channels (Figure 1A), but here the spatial layout of the channels is also sometimes used in decoding. Note that as these electrical recording methods get less invasive, spatial precision decreases (from spikes to LFP to ECoG to EEG), which can lead to inferior decoding performance [55, 56]. Still, all these electrical signals can be recorded at high temporal resolution (100–1000s of Hz), which make them good candidates for fast timescale decoding.

When decoding from wideband, LFP, EEG and ECoG data, it is common to first extract spectrotemporal features from the data, for example the signals in specific frequency bands. Sometimes, only 'task-relevant' frequencies will be used for decoding—for instance, using high gamma frequencies in ECoG to decode speech [57, 58] (Figure 2A). More frequently, many frequencies will be included to better understand which are contributing to decoding [15, 59]. In general, these extracted features can then be put into almost any type of neural decoder, such as linear (or logistic) regression or a deep neural network [60].

It is also possible to let a deep learning architecture do more of the feature extraction. One approach is to first convert each electrode's signal into a frequency domain representation over time (i.e. a spectrogram), often via a wavelet transform. Then, this two-dimensional representation (like an image) is provided as an input to a CNN [56, 61–63] (Figure 2B). If multiple electrode channels are being used for decoding, each channel can be fed into an independent CNN, or alternatively, the CNN weights for each channel can be shared [56]. The CNN will then learn the relevant frequency domain representation for the decoding.

Another approach is to provide the raw input signals into a deep learning architecture (Figure 2C). To learn temporal features, typically the signal is fed into a one-dimensional CNN, where the convolutions occur in the time domain. This has been done with a standard CNN [64], in addition to variant architectures. Ahmadi *et al.* [65] used a temporal convolutional network, which is a more complex version of a one-dimensional CNN that (among other things) allows for multiple timescales of inputs to affect the output. Li *et al.* [66] used parameterized versions of temporal filters that target synchrony between electrodes. These convolutional approaches will automatically learn temporal filters (like frequency bands) that are relevant for decoding.

In addition to temporal structure, there is often spatial structure of the electrode channels that can also be leveraged for neural decoding (Figure 2A). Convolutional filters can be used in the spatial domain to learn spatial representations that are relevant for decoding, for example local functional correlation structure. It is common for the temporal filters and spatial filters to be learned in successive layers of the network, either temporal followed by spatial [67, 68] or vice-versa [69, 70]. In addition, three-dimensional convolutional filters can be learned that simultaneously incorporate both temporal and (two dimensional) spatial dimensions [37] or three spatial dimensions [71]. Including spatial filters, which is most common in EEG and ECoG, can help learn spatial motifs that are most relevant for the task. Moreover, from a practical perspective, convolutional networks are an efficient way of processing high-dimensional spatial data.

Functional magnetic resonance imaging (fMRI) and other noninvasive modalities

Magnetoencephalography (MEG), functional near infrared spectroscopy (fNIRS) and fMRI are also noninvasive recording modalities, which are most often used in human decoding experiments. In this paper, among these noninvasive modalities, we primarily consider examples of decoding from fMRI. fMRI measures blood oxygenation (a proxy for neural activity) through resonance imagining, and its temporal resolution is limited by its dynamics. fMRI datasets contain activity signals in different 'voxels' (locations) of the brain over time. Due to the limited temporal resolution, sometimes the temporal continuity of this data is not used for decoding purposes (Figure 1A).

In fMRI, feature engineering is often done by hand. Commonly, the fMRI voxels that are used for decoding are subselected by hand or with statistical tests. In addition, other hand-engineered metrics like functional connectivity are sometimes used as decoder inputs [72, 73]. As in EEG and ECoG, CNNs can be used to automatically extract features. For instance, spatial features can be learned by inputting the entire set of voxels into a three-dimensional CNN [71, 74].

The outputs of decoding: behavior and perception

Movement

Some of the earliest uses of neural decoding were in the motor system [110]. Researchers have used neural activity from motor cortex to predict many different motor outputs, such as movement kinematics (e.g. position and velocity), muscle activity (EMG) and broad type of movement. Traditionally, this decoding has used methods (e.g. Kalman filter or Wiener filter) that assumed a linear mapping from neural activity to the motor output, which has led to many successes [111–115]. To improve the decoders, these methods were extended to allow specific nonlinearities (e.g. unscented Kalman filter, point process filter and Wiener cascade [116–120]). Within the last decade, deep learning methods have become more common, frequently outperforming linear methods and their direct nonlinear extensions when compared [39, 69, 81, 121]. Deep learning has shown to be a flexible tool for movement decoding, having been used to predict a wide range of movement variables from several different neural recording modalities (as catalogued in Table 1).

RNNs are by far the most common deep learning architecture for movement decoding. When predicting a continuous movement variable, there is generally a linear mapping from the RNN's output to the movement variable. When classifying movements, there is an additional softmax nonlinearity that determines the movement with the highest probability. From a deep learning perspective, given that this is a problem of converting one sequence (a temporal trace of neural activities) into another sequence (motor outputs), it would be expected that an RNN would be an appropriate architecture. Recurrent architectures also make sense from a scientific perspective: motor cortical activity has dynamics that are important for producing movements [123], plus movements themselves have dynamics.

LSTMs have generally been the most common and successful type of RNN for movement decoding [39, 60, 69, 77, 79, 81, 82, 87–89], although other standard types of RNN architectures (e.g. GRUs [80] and echostate networks [36]) have also proven successful. In addition, researchers have found that stacking multiple layers of LSTMs [81, 89] can improve performance beyond a single LSTM [81]. LSTMs are likely successful because they are able to learn long-term dependencies better than a standard 'vanilla' RNN [27].

A common goal of neural decoding of movement is to be able to create a usable brain computer interface for patients. Although the majority of deep learning uses have been in offline scenarios (decoding after the neural recording), there are several successful examples of real-time uses of deep learning for movement decoding [36, 84, 85, 121]. The first use of deep learning for real-time movement decoding was in Sussillo *et al.* [36]. Monkeys with implanted electrode arrays were able to control the velocity of a cursor on a screen in real time via the use of an echostate network, which outperformed a Kalman filter. In a more recent example, in human patients with tetraplegia who had implanted electrode arrays, Schwemmer *et al.* [84] were able to classify planned movements of wrist extension, wrist flexion, index extension and index flexion. This was done by inputting wideband activity into an LSTM, followed by a CNN, followed by a fully connected layer for classification. After classifying the movement type, the authors then applied functional electrical stimulation to activate muscles according to this neural decoder, so that the patient was able to make these movements in real time.

Although there has been great initial success, there are several challenges associated with using deep learning for real time decoding for brain computer interfaces. One challenge is that the source of the recorded neural activity can change across days, for example due to slight movement of implanted electrodes. One approach that has dealt with this is the multiplicative RNN, an architecture that allows mappings from the neural input to the motor output to partially change across days [121]. Another approach that helps to utilize data across multiple days is that of Pandarinath *et al.* [43], which uses recurrent autoencoders to find a consistent low-dimensional dynamical model of the data that is shared across days. Incorporating this dynamical model leads to more accurate low-dimensional representations that are more predictive of movement kinematics. One other approach is that of [83], which uses adversarial domain adaptation networks in order to align neural recordings across days.

Another challenge of using deep learning for real-time neural decoding is computation time, as there is the need to make predictions through the deep learning architecture at very high temporal resolution. When using a less complicated echostate network, Sussillo *et al.* [36] were able to decode with less than 25 ms temporal resolution. However, when using a more complex architecture of LSTMs followed by CNNs, Schwemmer *et al.* [84]

Table 1. Neural datasets and deep learning

Paper	Decoding objective	Neural modality (subject)	Architecture	Methods compared against	Intermediate var.	Real-time
Movement						
Sussillo *et al.* [36]	Predict cursor movement on screen	Spikes (NHP)	Echostate network	KF	No	Yes
Sussillo *et al.* [75]	Predict cursor movement on screen, stitch across days	Spikes (NHP)	Multiplicative RNN	KF	No	Yes
Pandarinath *et al.* [43]	Predict reach kinematics, stitch across days	Spikes (NHP, Human)	Recurrent autoencoder + Lin. Reg.	Lin. Reg., GPFA + Lin. Reg.	No	No
Glaser *et al.* [39]	Predict reach kinematics	Spikes (NHP)	LSTM	WF, WC, KF, Naive Bayes, SVR, XGBoost, FC, RNN, Ensemble	No	No
Makin *et al.* [76]	Predict reach kinematics	Spikes (NHP)	Restricted Boltzmann machine variant	WF, KF, Unscented KF	No	No
Ahmadi *et al.* [60]	Predict reach kinematics	LFP, Spikes (NHP)	LSTM	KF	No	No
Ahmadi *et al.* [65]	Predict reach kinematics	LFP (NHP)	TCN	LSTM	No	No
Ahmadi *et al.* [48]	Predict reach kinematics	Spikes (NHP)	QuasiRNN	WF, WC, KF, Unscented KF, RNN, GRU, LSTM	No	No
Park and Kim [77]	Predict reach kinematics	Spikes (NHP)	LSTM	KF	Speed, direction	No
Li *et al.* [78]	Predict forelimb reach location	Calcium imaging (Mouse)	CNN	None	No	No
Wang *et al.* [79]	Predict hindlimb kinematics	Spikes (NHP)	LSTM	WF, PLDS+WF, XGBoost, RNN,	No	No
Nakagome *et al.* [80]	Predict hindlimb kinematics	EEG (Human)	GRU	WF, Ridge Reg., Unscented KF, TCN, LSTM, Quasi RNN, CatBoost	No	No
Tseng *et al.* [81]	Predict reaching and hindlimb kinematics	Spikes (NHP)	Multilayer LSTM	WF, KF, Unscented KF, LSTM	No	No
Xie *et al.* [69]	Predict finger kinematics	ECoG (Human)	CNN+LSTM	Lin. Reg., Least Angle Reg., Random Forest, LSTM	No	No
Petrosuan *et al.* [70]	Predict finger kinematics	ECoG (Human)	CNN	WF	No	No
Naufel *et al.* [82]	Predict wrist EMG	Spikes (NHP)	LSTM	WF, WC	No	No
Farshchian *et al.* [83]	Predict wrist EMG, stitch across days	Spikes (NHP)	Recurrent autoencoder + adversarial domain adaptation network for alignment	CCA, KL Divergence minimization for alignment	No	No
Schwemmer *et al.* [84]	Classify wrist, index movements	Wideband (Human)	LSTM+CNN	SVM	No	Yes
Skomrock *et al.* [85]	Classify hand, wrist, index movements	Wideband (Human)	LSTM+CNN	SVM	No	Yes
Nurse *et al.* [86]	Classify hand squeeze	EEG (Human)	CNN	None	No	No
Pan *et al.* [87]	Classify hand gestures	ECoG (Human)	LSTM	Log. Reg., SVM, FC	No	No
Elango *et al.* [88]	Classify finger movements	ECoG (Human)	LSTM	LDA, HMM	No	No
Du *et al.* [89]	Classify finger movements	ECoG (Human)	FC + Multilayer LSTM	SVM	No	No
Speech						
Livezey *et al.* [15]	Classify produced speech syllable	ECoG (Human)	FC	Log. Reg., Lin. SVM	Yes	No
Sereshkeh *et al.* [90]	Classify yes/no/rest	EEG (Human)	FC	LDA, Lin. SVM, Poly. SVM, Naive Bayes, kNN	No	No

Continued

Table 1. Continued

Paper	Decoding objective	Neural modality (subject)	Architecture	Methods compared against	Intermediate var.	Real-time
Wang *et al.* [91]	Classify produced phrase	MEG (Human)	FC	GMM	No	No
Dash *et al.* [92]	Classify imagined and produced phrase	MEG (Human)	CNN	FC	No	No
Wilson *et al.* [93]	Classify produced phonemes	LFP (Human), Unsorted spikes	BiGRU	Log. Reg.	No	No
Yang *et al.* [58]	Reconstruct perceived speech spectrogram	ECoG (Human)	FC	Lin. Reg.	No	No
Heelan *et al.* [94]	Reconstruct perceived speech/call spectrogram	Spikes (NHP)	LSTM	FC, RNN, GRU, KF, Wiener filter, Wiener cascade	No	No
Angrick *et al.* [37]	Speech reconstruction	ECoG (Human)	3D CNN + WaveNet	None	Spectrogram	No
Anumanchipalli *et al.* [95]	Produced speech synthesis	ECoG (Human)	BiLSTM + BiLSTM + Synthesizer	Ablation	Articulator kinematics	No
Sun *et al.* [96]	Speech recognition	ECoG (Human)	BiLSTM + CNN	LSTM + ASR, Ablation	No	No
Makin *et al.* [97]	Speech recognition	ECoG (Human)	CNN + BiLSTM	HMM, Ablation	No	No
Krishna *et al.* [98]	Speech recognition, Speech reconstruction	EEG (Human)	RNN, GAN	Ablation	No	No
Willett *et al.* [99]	Reconstruct text	Spikes (Human)	GRU	KF+HMM on moving cursor to letters	Imagined handwriting	Yes
Vision						
Qiao *et al.* [100]	Classify visual stimuli	fMRI (Human)	CNN feature selection + BiLSTM	Decision Tree, RF, AdaBoost, Lin. SVM, Kernel SVM, FC	No	No
Ellis and Michaelides [101]	Classify visual stimuli	Calcium imaging (Mouse)	CNN	Lin. SVM, FC	No	No
Güçlütürk *et al.*[38]	Reconstruct perceived faces	fMRI (Human)	Bayesian CNN + GAN	Bayesian Linear + GAN	No	No
Parthasarathy *et al.* [35]	Reconstruct images	Spikes (NHP)	Lin. Reg. + CNN autoencoder	Low-fidelity image	No	No
St-Yves and Naselaris [102]	Reconstruct images	fMRI (Human)	CNN+dAE+GAN	None	No	No
Wen *et al.* [103]	Reconstruct and classify images	fMRI (Human)	Lin. Reg. + CNN	None	CNN activations	No
Seeliger *et al.* [104]	Reconstruct images	fMRI (Human)	Lin. Reg. + GAN + CNN	None	GAN inputs	No
Shen *et al.* [105]	Reconstruct images	fMRI (Human)	Lin. Reg. + CNN	Ablation	CNN activations	No
Shen *et al.* [106]	Reconstruct images	fMRI (Human)	GAN + CNN	Ablation	None	No
VanRullen and Reddy [107]	Reconstructing perceived faces	fMRI (Human)	Lin. Reg. + VAE + GAN	Lin. Reg. + PCA	VAE inputs	No
Dado *et al.* [108]	Reconstruct perceived faces	fMRI (Human)	GAN	VAE + GAN, Eigenfaces	GAN inputs	No
Kim *et al.* [109]	Reconstruct images	Spikes (NHP)	Nonlin. Reg. + CNN autoencoder	Low-fidelity image	No	No

Notes: Across the outputs of movement, speech and vision, we overview a nonexhaustive list of deep learning applications to neural decoding. The column 'Intermediate var.' refers to whether an intermediate variable was decoded, which was then used to predict the output. In papers where the goal was to compare many methods rather than focusing on a single method, we put a high-performing method in the 'Architecture' column. Lin. Reg., linear regression; Log. Reg., logistic regression; PCA, principal components analysis; WF, Wiener filter; WC, Wiener cascade; FC, fully connected network; RNN, standard recurrent neural network; GRU, gated recurrent unit network; LSTM, Long short-term memory network; SVR, support vector regression; SVM, support vector machine; PLDS, Poisson linear dynamical system; TCN, temporal convolutional network; LDA, linear discriminant analysis; HMM, hidden Markov model; (Bi)LSTM/GRU, (bidirectional) long short-term memory/gated recurrent unit; ASR, automatic speech recognition; NHP, nonhuman primate.

decoded at 100 ms resolution, slower than our perception. Relatedly, for linear methods that can be fit rapidly, researchers are able to adapt the neural decoder in real time to better match the subject's intention (trying to get to a target) to improve performance [112, 115, 117, 120]. Developing similar approaches for deep learning-based decoders is an exciting, unexplored area.

Speech

Vocal articulation of speech is a complex behavior that engages a large functional area of the brain to produce movements that have a high degree of articulatory temporal and spatial precision [124]. Its production is also a uniquely human ability, which limits the recording modalities and neuroscientific interventions that can be used to study it. Due to the functional and temporal requirements of decoding speech, cortical surface electrical potentials recorded using ECoG is the typical recording modality used, although penetrating electrodes, MEG, EEG and fNIRS are also used [90, 91, 125, 126]. When decoding from ECoG or EEG, researchers commonly use the signals' high gamma amplitude [57], although some use more broad spectrotemporal features as well [57, 59, 127].

Many approaches to decoding speech from neural signals have used some combination of linear methods and shallow probabilistic models. Clustering, SVMs, LDA, linear regression and probabilistic models have been used with spectrotemporal features of electrical potentials to decode vowel acoustics, speech articulator movements, phonemes, whole words and semantic categories [57, 59, 125, 128–131].

Deep learning approaches to decoding speech from neural signals have emerged that can potentially learn nonlinear mappings (Table 1). Some of these approaches have operated on temporally segmented neural data and have thus used fully connected neural network architectures. For example, spectrotemporal features derived from ECoG or EEG have been used to reconstruct perceived spectrograms, classify words or syllables or classify entire phrases [15, 58, 90–93, 127]. These examples with temporally segmented neural data are useful for increasing understanding about neural representations, and as a step towards decoding natural speech.

Mapping directly from continuous, time-varying neural signals to speech is the goal of speech brain–computer interfaces [1, 132]. Both convolutional and recurrent networks are able to flexibly decode timeseries data and are often used for decoding naturalistic speech. Heelan *et al.* [94] reconstructed perceived speech audio from multiunit spike counts from a nonhuman primate and found that LSTM-based networks outperformed other traditional and deep models. Speech represented as text does not have a simple one-to-one temporal alignment to regularly sampled neural signals. For this reason, speech-to-text decoding networks often use architectures and methods like sequence-to-sequence models or the connectionist temporal classification loss [24, 133], which are commonly used in machine translation or automated speech recognition applications. As such, several groups have decoded directly from neural signals to text during speech production or imagined handwriting using recurrent networks such as sequence-to-sequence models [96–99] (Figure 3C).

For decoding intelligible acoustic speech, it is also common to split neural decoding into a more constrained neural-to-intermediate mapping, followed by a second stage that maps this intermediate format into an acoustic waveform using acoustic priors for speech based on deep learning or hand-engineered methods. For instance, high gamma features recorded using ECoG have been used to decode spectrograms, which were fed into a WaveNet [122] deep network to produce an acoustic waveform [37]. As a specific example of a split decoding setup, Anumanchipalli *et al.* [95] trained a bidirectional LSTM to decode articulator kinematic features (movement of lips, tongue, etc.) from a combination of high gamma amplitude and a low-frequency component. The second stage was a separate bidirectional LSTM, which decoded acoustic features (mel-frequency cepstral coefficients, voicing, etc.) from the decoded articulatory features. Finally, these acoustic features were passed into a speech synthesizer. Compared with an RNN that skips the intermediate articulator kinematics stage, their two-stage method decoded perceptually improved speech acoustics. The second stages do not require invasive neural data for training and were trained on a larger second corpus.

Deep learning models have improved the accuracy of primarily offline speech decoding tasks. Many of the preprocessing and decoding methods reviewed here are done offline using acausal or high-latency deep learning models. Developing deep learning methods, software and hardware for real-time speech decoding is important for clinical applications of brain computer interfaces [131, 134].

Vision

Similar to decoding acoustic speech, decoding visual stimuli from neural signals requires strong image priors due to the large variability of natural scenes and the relatively small bit-rate of neural recordings. Early attempts to reconstruct the full visual experience restricted decoding to simple images [135] or relied on a filterbank encoding model and a large set of natural images as a sampled prior [136]. Qiao *et al.* [100] solved the simpler task of classifying perceived object category using one CNN to select a small set of fMRI voxels, which were fed into a second RNN for classification. Similarly, Ellis and Michaelides [101] classified among many visual scenes from calcium imaging data using feedforward or convolutional neural networks (CNNs).

As mentioned in Section 2, deep generative image models, such as GANs, can produce realistic images. In addition, convolutional neural networks trained to classify large naturalistic image databases [137] (discriminative models) have been shown to encode a large amount of textural and semantic meaning in their activations [138], which can be used as an image prior. Due to the variety of ways that natural image priors can be created with deep networks, there exist neural decoding methods that combine different aspects of both generative and discriminative networks.

Given a deep generative model of images, a simpler neural decoder can be trained to map from neural data to the latent space of the model [38, 104, 107, 108], and the generative model can be used for image reconstruction. As an example, Seeliger *et al.* [104] trained a convolutional GAN [33, 34] to generate grayscale images of objects given a noise vector as input. The parameters of the network were then frozen. Then, a linear regression model was trained to generate GAN input vectors from fMRI to optimize both the reconstruction of an image's individual pixels and higher order image features. Similarly, a linear stage or combined linear and deep learning reconstruction, followed by a deep network that cleans-up the image, has been used with retinal ganglion cell output [35, 109]. Generative models can also be trained to reconstruct images directly from fMRI responses on real data with data augmentation from a simulated encoding model [102].

Alternatively, generative and discriminative models can be used together. By leveraging a pretrained CNN, a simple neural decoder can be trained to map neural data to CNN activations that can then be passed into a convolutional image reconstruction model [103]. In addition, the input image in a pretrained CNN can be optimized so that the CNN activations match predictions given by the fMRI responses [105]. Researchers have also used an end-to-end approach in which they train the generative part directly on neural data with both an adversarial loss and a pretrained CNN feature loss [106]. Along with acoustic speech, decoding naturalistic visual stimuli presents one of the best cases to study the use of data-driven priors derived from deep networks.

Other

Although we have chosen to focus on a few decoding outputs that are prevalent in the literature, deep learning has been used for a myriad of neural decoding applications. For instance, RNNs such as LSTMs have been used to decode an animal's location [39, 56, 139, 140] and direction [141] from spiking activity in the hippocampus and head-direction cells, respectively. Deep networks have been used to decode what is being remembered in a working memory task from [142] and to predict illness [71–74, 143, 144] from human fMRI. Researchers have used LSTMs [145] and feedforward neural networks [146] to classify different classes of behaviors, using spiking activity in animals [146] and fNIRS or fMRI measurements in humans [16, 145]. LSTMs [147, 148] and CNNs [149] have been used to classify emotions from EEG signals. Feedforward neural networks have been used to determine the source of a subject's attention, using EEG in humans [150, 151] and spiking activity in monkeys [152]. CNNs [62–64], along with LSTMs [64], have been used to predict a subject's stage of sleep from their EEG. For almost any behavioral signal that can be decoded, examples exists of applying deep learning.

Discussion

Deep learning is an attractive method for use in neural decoding because of its ability to learn complex, nonlinear transformations from data. In many of the examples above, deep networks can outperform linear or shallow methods even on relatively small datasets; however, examples exist where this is not the case, especially when using fMRI [153, 154] or fNIRS data [155]. Relatedly, there are many times in which using hand-engineered features can outperform an end-to-end neural network that will learn the features. This is more likely with limited amounts of data, and also when there is strong prior knowledge about the relevant features. One general machine learning approach to efficiently use limited data is transfer learning, in which a neural network trained in one scenario (typically with more data) is used in a separate scenario. This has been used in neural decoding to more effectively train decoders for new subjects [88, 97] and for new predicted outputs [84]. As the capability to generate ever larger datasets develops with automated, long-term experimental setups for single animals [156] and large-scale recordings across multiple animals [157], deep learning is well poised to take advantage of this flood of data. As dataset sizes increase, this will also allow more features to be learned through data-driven network training rather than being selected by hand.

Although deep learning will inevitably improve decoding accuracy as neuroscientists collect larger datasets, extracting scientific knowledge from trained networks is still an area of active research. That is, can we understand the transformations deep networks are learning? In computer vision, layers that include spatial attention [158] and methods for performing feature attribution [159] have been developed to understand what parts of the input are important for prediction, although the latter are an active area of research [160]. These methods could be used to attribute what channels, neurons (e.g. of different genetically defined cell types) or time-points are most salient for neural decoding [159]. In addition, there are methods for understanding deep network representations in computer vision that examine the representations networks have learned across layers [161, 162]. Using these methods may help to understand the transformations that occur within neural decoders; however, results may be sensitive to the decoder's architecture and not purely the data's structure. Although deep learning interpretability methods are not commonly used on decoders trained on neural data, there are a few examples of networks that were built with interpretability in mind or were investigated after training [15, 16, 66, 67, 70, 142].

When interpreting neural decoders, it is often assumed that the decoder reveals the information contained in the brain about the decoded variable. It is important to note that this is only partially true when priors are being used for decoding [163], which is often the case when decoding a full image or acoustic speech. In these scenarios, the decoded outputs will be a function of both neural activity and the prior, so one cannot simply determine what information the brain has about the output.

The software used to create, train and evaluate deep networks has been steadily developed and is now almost as easy to use as other standard machine learning methods. A wide range of cost functions, layer types and parameter optimization algorithms are implemented and accessible in deep learning libraries such as PyTorch or TensorFlow [164, 165] and libraries in other programming languages. Like other machine learning methods, care must be taken to carefully cross-validate results as deep networks can easily overfit to the training data.

In addition to their use in neural decoding, deep learning has other prominent uses within neuroscience [166, 167]. Neural networks have a long history in neuroscience as models of neural processing [168, 169]. More recently, there has also been a surge of papers using deep networks as encoding models [12, 14, 75]. There has been a specific focus on using the representations learned by deep networks trained to perform behavioral tasks (e.g. image recognition) to predict neural responses in corresponding brain areas (e.g. across the visual hierarchy [170]). Combining these multiple complementary approaches is one promising approach to understanding neural computation.

Future directions

The use of deep learning for neural decoding has greatly increased within the last few years. Here, we highlight several open challenges and potential future directions for research.

- Increased use of deep learning for online decoding. This will be benefited by speed-ups in computing performance and reduced latency hardware [171, 172], online adaptation of neural network decoders to subjects' intentions [112, 115, 117] and robustness of decoders to signal changes across days [43, 75, 83].
- Keeping pace with the state-of-the-art in deep learning methods. Machine learning datasets are typically much larger than neural datasets. Do the architectural improvements in deep learning that are beneficial in other domains translate to neural datasets?
- Standardized benchmark datasets for better comparisons with strong baseline models and clear cross validation and evaluation standards [173–175].
- Increased interpretability of the inner workings of neural decoders, either by specifically creating more interpretable architectures [66, 70] or by *post hoc* analyses of the neural networks [15, 56].

Key Points

- We review many deep learning approaches, which have been used to create more accurate and flexible neural decoders.
- Traditionally, many decoders have used hand-engineered features as inputs; deep learning tools can help to automatically learn relevant features from neural inputs.
- Pretrained deep learning models can be used as priors for complex decoding targets such as images or acoustic speech.
- We discuss directions for future research.

Acknowledgments

The authors would like to thank Ella Batty and Charles Frye for very helpful comments on this manuscript.

Funding

This work was supported by National Science Foundation NeuroNex Award DBI-1707398 [J.I.G.]; The Gatsby Foundation AT3708 [J.I.G.]; and LBNL Laboratory Directed Research and Development [J.A.L.].

References

1. Wolpaw JR, Birbaumer N, McFarland DJ, *et al*. Brain–computer interfaces for communication and control. *Clin Neurophysiol* 2002; **113**(6): 767–91.
2. Zhang X, Yao L, Wang X, *et al*. A survey on deep learning based brain computer interface: recent advances and new frontiers. *arXiv preprint arXiv:190504149* 2019.
3. Quiroga RQ, Snyder LH, Batista AP, *et al*. Movement intention is better predicted than attention in the posterior parietal cortex. *J Neurosci* 2006; **26**(13): 3615–20.
4. Harrison SA, Tong F. Decoding reveals the contents of visual working memory in early visual areas. *Nature* 2009; **458**(7238): 632–5.
5. Acharya S, Fifer MS, Benz HL, *et al*. Electrocorticographic amplitude predicts finger positions during slow grasping motions of the hand. *J Neural Eng* 2010; **7**(4):046002.
6. Weygandt M, Blecker CR, Schäfer A, *et al*. fMRI pattern recognition in obsessive–compulsive disorder. *Neuroimage* 2012; **60**(2): 1186–93.
7. Rich EL, Wallis JD. Decoding subjective decisions from orbitofrontal cortex. *Nat Neurosci* 2016; **19**(7): 973.
8. Glaser JI, Perich MG, Ramkumar P, *et al*. Population coding of conditional probability distributions in dorsal premotor cortex. *Nat Commun* 2018; **9**(1): 1–14.
9. Hamilton LS, Edwards E, Chang EF. A spatial map of onset and sustained responses to speech in the human superior temporal gyrus. *Curr Biol* 2018; **28**(12): 1860–71.
10. Brackbill N, Rhoades C, Kling A, *et al*. Reconstruction of natural images from responses of primate retinal ganglion cells. *bioRxiv* 2020.
11. Rashid B, Calhoun V. Towards a brain-based predictome of mental illness. *Hum Brain Mapp* 2020; **41**(12): 3468–35.
12. McIntosh L, Maheswaranathan N, Nayebi A, *et al*. Deep learning models of the retinal response to natural scenes. In: *Advances in Neural Information Processing Systems*, 2016;1369–77.
13. Nagamine T, Mesgarani N. Understanding the representation and computation of multilayer perceptrons: a case study in speech recognition. In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, 2564–73.
14. Kell AJE, Yamins DLK, Shook EN, *et al*. A task-optimized neural network replicates human auditory behavior, predicts brain responses, and reveals a cortical processing hierarchy. *Neuron* 2018; **98**(3): 630–44.
15. Livezey JA, Bouchard KE, Chang EF. Deep learning as a tool for neural data analysis: speech classification and cross-frequency coupling in human sensorimotor cortex. *PLoS Comput Biol* 2019; **15**(9):e1007091.
16. Jang H, Plis SM, Calhoun VD, *et al*. Task-specific feature extraction and classification of fMRI volumes using a deep neural network initialized with a deep belief network: evaluation using sensorimotor tasks. *Neuroimage* 2017; **145**: 314–28.
17. Alipanahi B, Delong A, Weirauch MT, Frey BJ. Predicting the sequence specificities of DNA-and RNA-binding proteins by deep learning. *Nat Biotechnol* 2015; **33**(8): 831–8.
18. Piech C, Bassen J, Huang J, *et al*. Deep knowledge tracing.In: *Advances in Neural Information Processing Systems* 2015; 505–13.
19. Paganini M, de Oliveira L, Nachman B. CaloGAN: simulating 3d high energy particle showers in multilayer electromagnetic calorimeters with generative adversarial networks. *Physical Review D* 2018; **97**(1):014021.
20. Kurth T, Treichler S, Romero J, *et al*. Exascale deep learning for climate analytics. In: *SC18: International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE, 2018, 649–60.
21. Schütt KT, Sauceda HE, Kindermans P-J, *et al*. Schnet–a deep learning architecture for molecules and materials. *J Chem Phys* 2018; **148**(24): 241722.
22. Hochreiter S, Schmidhuber J. Long short-term memory. *Neural Comput* 1997; **9**(8): 1735–80.
23. Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. In: *Advances in Neural Information Processing Systems*, 2012, 1097–105.
24. Sutskever I, Vinyals O, Le QV. Sequence to sequence learning with neural networks. In: *Advances in Neural Information Processing Systems*, 2014, 3104–12.
25. He K, Zhang X, Ren S, *et al*. Deep residual learning for image recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* 2016;770–8.
26. Amodei D, Ananthanarayanan S, Anubhai R, *et al*. Deep speech 2: end-to-end speech recognition in english and mandarin. In: *International Conference on Machine Learning* 2016;173–82.
27. Goodfellow I, Bengio Y, Courville A. *Deep Learning*. Cambridge, MA: MIT Press, 2016.
28. Wu Z, Pan S, Chen F, *et al*. A comprehensive survey on graph neural networks. *IEEE Trans Neural Netw Learn Syst* 2020;1–21.
29. Vaswani A, Shazeer N, Parmar N, *et al*. Attention is all you need. In: *Advances in Neural Information Processing Systems*, 2017, 5998–6008.
30. Cho K, Van Merriënboer B, Bahdanau D, *et al*. On the properties of neural machine translation: encoder-decoder approaches. arXiv preprint arXiv:1409.1259. 2014.

31. LeCun Y, Bengio Y. Convolutional networks for images, speech, and time series. In: Michael A. Arbib (ed). *The Handbook of Brain Theory and Neural Networks*. Cambridge, MA: MIT Press, 1995; **3361**(10): 1995.
32. Ciresan DC, Meier U, Masci J, *et al.* Flexible, high performance convolutional neural networks for image classification. In: *Twenty-second International Joint Conference on Artificial Intelligence*, 2011.
33. Goodfellow I, Pouget-Abadie J, Mirza M, *et al.* Generative adversarial nets. *In: Advances in Neural Information Processing Systems*, 2014,2672–80.
34. Radford A, Metz L, Chintala S. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:151106434*, 2015.
35. Parthasarathy N, Batty E, Falcon W, *et al.* Neural networks for efficient Bayesian decoding of natural images from retinal neurons. *In: Advances in Neural Information Processing Systems*, 2017, 6434–45.
36. Sussillo D, Nuyujukian P, Fan JM, *et al.* A recurrent neural network for closed-loop intracortical brain–machine interface decoders. *J Neural Eng* 2012; **9**(2):026027.
37. Angrick M, Herff C, Mugler E, *et al.* Speech synthesis from ECoG using densely connected 3D convolutional neural networks. *J Neural Eng* 2019; **16**(3):036019.
38. Güçlütürk Y, Güçlü U, Seeliger K, *et al.* Reconstructing perceived faces from brain activations with deep adversarial neural decoding. In: *Advances in Neural Information Processing Systems*, 2017, 4246–57.
39. Glaser JI, Benjamin AS, Chowdhury RH, *et al.* Machine learning for neural decoding. *Eneuro* 2020; **7**(4).
40. Bouchard KE and Chang EF. *Human ECoG speaking consonant-vowel syllables*. Figshare 2019. doi: doi.org/10.6084/m9.figshare.c.4617263.v4.
41. Kazemifar S, Manning KY, Rajakumar N, *et al.* Spontaneous low frequency bold signal variations from resting-state fMRI are decreased in Alzheimer disease. *PloS One* 2017; **12**(6).
42. Sussillo D, Jozefowicz R, Abbott LF, *et al.* LFADS-latent factor analysis via dynamical systems arXiv preprint arXiv:1608.06315. 2016a.
43. Pandarinath C, O'Shea DJ, Collins J, *et al.* Inferring single-trial neural population dynamics using sequential auto-encoders. *Nat Methods* 2018; **15**(10): 805–15.
44. Lee JH, Mitelut C, Shokri H, *et al.* Yass: yet another spike sorter applied to large-scale multi-electrode array recordings in primate retina. *bioRxiv* 2020.
45. Rácz M, Liber C, Németh E, *et al.* Spike detection and sorting with deep learning. *J Neural Eng* 2020; **17**(1):016038.
46. Emami M, Sahraee-Ardakan M, Pandit P, *et al.* Low-rank nonlinear decoding of μ-ECoG from the primary auditory cortex. *arXiv preprint arXiv:*200505053 2020.
47. Prince LY, Richards BA. Inferring hierarchies of latent features in calcium imaging data. *NeurIPS 2019 Workshop Neuro AI*, 2019.
48. Ahmadi N, Constandinou T., Bouganis C-S. Robust and accurate decoding of hand kinematics from entire spiking activity using deep learning *bioRxiv* 2020.
49. Chen T-W, Wardill TJ, Sun Y, *et al.* Ultrasensitive fluorescent proteins for imaging neuronal activity. *Nature* 2013; **499**(7458): 295–300.
50. Giovannucci A, Friedrich J, Gunn P, *et al.* CaImAn an open source tool for scalable calcium imaging data analysis. *Elife* 2019; **8**:e38173.
51. Vogelstein JT, Packer AM, Machado TA, *et al.* Fast non-negative deconvolution for spike train inference from population calcium imaging. *J Neurophysiol* 2010; **104**(6): 3691–704.
52. Soltanian-Zadeh S, Sahingur K, Blau S, *et al.* Fast and robust active neuron segmentation in two-photon calcium imaging using spatiotemporal deep learning. *Proc Natl Acad Sci* 2019; **116**(17): 8554–63.
53. Speiser A, Yan J, Archer EW, *et al.* Fast amortized inference of neural activity from calcium imaging data with variational autoencoders. In: *Advances in Neural Information Processing Systems*, 2017, 4024–34.
54. Buzsáki G, Anastassiou C., Koch C.. The origin of extracellular fields and currents-EEG, ECoG, LFP and spikes. *Nat Rev Neurosci* 2012; **13**(6): 407–20.
55. Flint RD, Ethier C, Oby ER, *et al.* Local field potentials allow accurate decoding of muscle activity. *J Neurophysiol* 2012; **108**(1): 18–24.
56. Frey M, Tanni S, Perrodin C, *et al.* Deepinsight: a general framework for interpreting wide-band neural activity. *bioRxiv* 2019;871848.
57. Bouchard KE, Chang EF. Neural decoding of spoken vowels from human sensory-motor cortex with high-density electrocorticography. In: *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 2014, 6782–5.
58. Yang M, Sheth SA, Schevon CA, *et al.* Speech reconstruction from human auditory cortex with deep neural networks. In: *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.
59. Mugler EM, Patton JL, Flint RD, *et al.* Direct classification of all American English phonemes using signals from functional speech motor cortex. *J Neural Eng* 2014; **11**(3): 035015.
60. Ahmadi N, Constandinou TG, Bouganis C-S. Decoding hand kinematics from local field potentials using long short-term memory (LSTM) network. In: *2019 9th International IEEE/EMBS Conference on Neural Engineering (NER)*. IEEE, 2019a, 415–9.
61. Golshan HM, Hebb AO, Mahoor MH. LFP-Net: a deep learning framework to recognize human behavioral activities using brain STN-LFP signals. *J Neurosci Methods* 2020; **335**: 108621.
62. Wang J, Zhang Y, Ma Q, *et al.* Deep learning for single-channel EEG signals sleep stage scoring based on frequency domain representation. In: *International Conference on Health Information Science*. Springer, 2019, 121–33.
63. Barger Z, Frye CG, Liu D, *et al.* Robust, automated sleep scoring by a compact neural network with distributional shift correction. *PloS One* 2019; **14**(12).
64. Supratak A, Dong H, Wu C, *et al.* DeepSleepNet: a model for automatic sleep stage scoring based on raw single-channel EEG. *IEEE Trans Neural Syst Rehabil Eng* 2017; **25**(11): 1998–2008.
65. Ahmadi N, Constandinou TG, Bouganis C-S. End-to-end hand kinematic decoding from LFPS using temporal convolutional network. In: *2019 IEEE Biomedical Circuits and Systems Conference (BioCAS)*. IEEE, 2019b, 1–4.
66. Li Y, Dzirasa K, Carin L, *et al.* Targeting EEG/LFP synchrony with neural nets. In: *Advances in Neural Information Processing Systems*, 2017, 4620–30.
67. Schirrmeister RT, Springenberg JT, Fiederer LDJ, *et al.* Deep learning with convolutional neural networks for EEG

decoding and visualization. *Hum Brain Mapp* 2017; **38**(11): 5391–420.

68. Lawhern VJ, Solon AJ, Waytowich NR, *et al*. EEGNeT: a compact convolutional neural network for EEG-based brain–computer interfaces. *J Neural Eng* 2018; **15**(5):056013.
69. Xie Z, Schwartz O, Prasad A. Decoding of finger trajectory from ECoG using deep learning. *J Neural Eng* 2018; **15**(3):036009.
70. Petrosuan A, Lebedev M, Ossadtchi A. Decoding neural signals with a compact and interpretable convolutional neural network. *bioRxiv* 2020.
71. Liang Z, Zheng J, Miao C, *et al*. 3D CNN based automatic diagnosis of attention deficit hyperactivity disorder using functional and structural MRI. *IEEE Access* 2017; **5**:23626–36.
72. Kim J, Calhoun VD, Shim E, Lee J-H. Deep neural network with weight sparsity control and pre-training extracts hierarchical features and enhances classification performance: evidence from whole-brain resting-state functional connectivity patterns of schizophrenia. *Neuroimage* 2016; **124**:127–46.
73. Iidaka T. Resting state functional magnetic resonance imaging and neural network classified autism and control. *Cortex* 2015; **63**:55–67.
74. Dong N, Zhang H, Adeli E, *et al*. 3D deep learning for multi-modal imaging-guided survival time prediction of brain tumor patients. In: *International Conference on Medical Image Computing and Computer-assisted Intervention*. Springer, 2016, 212–20.
75. Sussillo D, Churchland MM, Kaufman MT, Shenoy KV. A neural network that finds a naturalistic solution for the production of muscle activity. *Nat Neurosci* 2015; **18**(7): 1025–33.
76. Makin JG, O'Doherty JE, Cardoso MMB, Sabes PN. Superior arm-movement decoding from cortex with a new, unsupervised-learning algorithm. *J Neural Eng* 2018; **15**(2): 026010.
77. Park J, Kim S-P. Estimation of speed and direction of arm movements from M1 activity using a nonlinear neural decoder. In: *2019 7th International Winter Conference on Brain-Computer Interface (BCI)*. IEEE, 2019, 1–4.
78. Li C, Chan DCW, Yang X, et al Prediction of forelimb reach results from motor cortex activities based on calcium imaging and deep learning. *Front Cell Neurosci* 2019; **13**:88.
79. Wang Y, Truccolo W, Borton DA. Decoding hindlimb kinematics from primate motor cortex using long short-term memory recurrent neural networks. In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE. 2018, 1944–47.
80. Nakagome S, Luu TP, He Y, *et al*. An empirical comparison of neural networks and machine learning algorithms for EEG gait decoding. *Sci Rep* 2020; **10**(1): 1–17.
81. Tseng P-H, Urpi NA, Lebedev M, Nicolelis M. Decoding movements from cortical ensemble activity using a long short-term memory recurrent network. *Neural Comput* 2019; **31**(6): 1085–113.
82. Naufel S, Glaser JI, Kording KP, et al. A muscle-activity-dependent gain between motor cortex andEMG. *J Neurophysiol* 2019; **121**(1): 61–73.
83. Farshchian A, Gallego JA, Cohen JP, *et al*. Adversarial domain adaptation for stable brain-machine interfaces. *arXiv preprint arXiv:181000045*, 2018.
84. Schwemmer MA, Skomrock ND, Sederberg PB, *et al*. Meeting brain–computer interface user performance expectations using a deep neural network decoding framework. *Nat Med* 2018; **24**(11): 1669–76.
85. Skomrock ND, Schwemmer MA, Ting JE, *et al*. A characterization of brain-computer interface performance trade-offs using support vector machines and deep neural networks to decode movement intent. *Front Neurosci* 2018; **12**:763.
86. Nurse E, Mashford BS, Yepes AJ, *et al*. Decoding EEG and LFP signals using deep learning: heading T0rueNorth. In: *Proceedings of the ACM International Conference on Computing Frontiers*, 2016, 259–66.
87. Pan G, Li J-J, Yu Q, *et al*. Rapid decoding of hand gestures in electrocorticography using recurrent neural networks. *Front Neurosci* 2018; **12**:555.
88. Elango V, Patel AN, Miller KJ, *et al*. Sequence transfer learning for neural decoding. *bioRxiv* 2017;210732.
89. Anming D, Yang S, Liu W, *et al*. Decoding ECoG signal with deep learning model based on LSTM. In: *TENCON 2018–2018 IEEE Region 10 Conference*. IEEE, 2018, 0430–5.
90. Sereshkeh AR, Trott R, Bricout A, *et al*. EEG classification of covert speech using regularized neural networks. *IEEE/ACM Trans Audio, Speech, Language Process* 2017; **25**(12): 2292–300.
91. Wang J, Kim M, Hernandez-Mulero AW, *et al*. Towards decoding speech production from single-trial magnetoencephalography MEG) signals. In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, 3036–40.
92. Dash D, Ferrari P, Wang J. Decoding imagined and spoken phrases from non-invasive neural (MEG) signals. *Front Neurosci* 2020; **14**: 290.
93. Wilson GH, Stavisky SD, Willett FR, *et al*. Decoding spoken english phonemes from intracortical electrode arrays in dorsal precentral gyrus. *bioRxiv* 2020.
94. Heelan C, Lee J, O'Shea R, *et al*. Decoding speech from spike-based neural population recordings in secondary auditory cortex of non-human primates. *Commun Biol* 2019; **2**(1): 1–12.
95. Anumanchipalli GK, Chartier J, Chang EF. Speech synthesis from neural decoding of spoken sentences. *Nature* 2019; **568**(7753): 493.
96. Sun P, Anumanchipalli GK, Chang EF. Brain2Char: a deep architecture for decoding text from brain recordings. *arXiv preprint arXiv:190901401* 2019.
97. Makin JG, Moses DA, Chang EF. Machine translation of cortical activity to text with an encoder–decoder framework. *Nat Neurosci*, 2020; **23**: 575–82.
98. Krishna G, Han Y, Tran C, *et al*. State-of-the-art speech recognition using EEG and towards decoding of speech spectrum from EEG. *arXiv preprint arXiv:190805743* 2019.
99. Willett FR, Avansino DT, Hochberg LR, *et al*. High-performance brain-to-text communication via imagined handwriting. *bioRxiv* 2020.
100. Qiao K, Chen J, Wang L, *et al*. Category decoding of visual stimuli from human brain activity using a bidirectional recurrent neural network to simulate bidirectional information flows in human visual cortices. *Front Neurosci* 2019; **13**.
101. Ellis RJ, Michaelides M. High-accuracy decoding of complex visual scenes from neuronal calcium responses. *bioRxiv* 2018:271296.
102. St-Yves G, Naselaris T. Generative adversarial networks conditioned on brain activity reconstruct seen images. In: *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2018, 1054–61.

103. Wen H, Shi J, Zhang Y, *et al*. Neural encoding and decoding with deep learning for dynamic natural vision. *Cereb Cortex* 2018; **28**(12): 4136–60.

104. Seeliger K, Güçlü U, Ambrogioni L, *et al*. Generative adversarial networks for reconstructing natural images from brain activity. *Neuroimage* 2018; **181**:775–85.

105. Shen G, Horikawa T, Majima K, *et al*. Deep image reconstruction from human brain activity. *PLoS Comput Biol* 2019a; **15**(1):e1006633.

106. Shen G, Dwivedi K, Majima K, *et al*. End-to-end deep image reconstruction from human brain activity. *Front Comput Neurosci* 2019b; **13**.

107. VanRullen R, Reddy L. Reconstructing faces from fMRI patterns using deep generative neural networks. *Commun Biol* 2019; **2**(1): 1–10.

108. Dado T, Gucluturk Y, Ambrogioni L, *et al*. Hyperrealistic neural decoding: linear reconstruction of face stimuli from fMRI measurements via the Gan latent space. *bioRxiv* 2020.

109. Kim YJ, Brackbill N, Batty E, *et al*. Nonlinear decoding of natural images from large-scale primate retinal ganglion recordings. *bioRxiv* 2020.

110. Georgopoulos AP, Caminiti R, Kalaska JF, Massey JT. Spatial coding of movement: a hypothesis concerning the coding of movement direction by motor cortical populations. *Exp Brain Res* 1983; **49**(Suppl. 7): 327–36.

111. Wu W, Black MJ, Gao Y, *et al*. Neural decoding of cursor motion using a Kalman filter. In: *Advances in Neural Information Processing Systems*, 2003, 133–40.

112. Gilja V, Nuyujukian P, Chestek CA, *et al*. A high-performance neural prosthesis enabled by control algorithm design. *Nat Neurosci* 2012; **15**(12): 1752.

113. Serruya MD, Hatsopoulos NG, Paninski L, *et al*. Instant neural control of a movement signal. *Nature* 2002; **416**(6877): 141–2.

114. Carmena JM, Lebedev MA, Crist RE, *et al*. Learning to control a brain–machine interface for reaching and grasping by primates. *PLoS Biol* 2003; **1**(2).

115. Orsborn AL, Moorman HG, Overduin SA, *et al*. Closed-loop decoder adaptation shapes neural plasticity for skillful neuroprosthetic control. *Neuron* 2014; **82**(6): 1380–93.

116. Zheng L, O'Doherty JE, Hanson TL, *et al*. Unscented Kalman filter for brain-machine interfaces. *PloS One* 2009; **4**(7).

117. Luu TP, He Y, Brown S, *et al*. Gait adaptation to visual kinematic perturbations using a real-time closed-loop brain–computer interface to a virtual reality avatar. *J Neural Eng* 2016; **13**(3): 036006.

118. Pohlmeyer EA, Solla SA, Perreault EJ, *et al*. Prediction of upper limb muscle activity from motor cortical discharge during reaching. *J Neural Eng* 2007; **4**(4): 369.

119. Ethier C, Oby ER, Bauman MJ, *et al*. Restoration of grasp following paralysis through brain-controlled stimulation of muscles. *Nature* 2012; **485**(7398): 368–71.

120. Shanechi MM, Orsborn AL, Carmena JM. Robust brain-machine interface design using optimal feedback control modeling and adaptive point process filtering. *PLoS Comput Biol* 2016; **12**(4):e1004730.

121. Sussillo D, Stavisky SD, Kao JC, *et al*. Making brain–machine interfaces robust to future neural variability. *Nat Commun* 2016b; **7**:13749.

122. van den Oord A, Dieleman S, Zen H, *et al*. Wavenet: a generative model for raw audio. *arXiv preprint arXiv:160903499* 2016.

123. Shenoy KV, Sahani M, Churchland MM. Cortical control of arm movements: a dynamical systems perspective. *Annu Rev Neurosci* 2013; **36**:337–59.

124. Bouchard KE, Mesgarani N, Johnson K, *et al*. Functional organization of human sensorimotor cortex for speech articulation. *Nature* 2013; **495**(7441): 327.

125. Chan AM, Halgren E, Marinkovic K, *et al*. Decoding word and category-specific spatiotemporal representations from MEG and EEG. *Neuroimage* 2011; **54**(4): 3028–39.

126. Herff C, Schultz T. Automatic speech recognition from neural signals: a focused review. *Front Neurosci* 2016; **10**:429.

127. Akbari H, Khalighinejad B, Herrero JL, *et al*. Towards reconstructing intelligible speech from the human auditory cortex. *Sci Rep* 2019; **9**(1): 1–12.

128. Conant DF, Bouchard KE, Leonard MK, *et al*. Human sensorimotor cortex control of directly measured vocal tract movements during vowel production. *J Neurosci* 2018; **38**(12): 2955–66.

129. Kellis S, Miller K, Thomson K, *et al*. Decoding spoken words using local field potentials recorded from the cortical surface. *J Neural Eng* 2010; **7**(5):056007.

130. Herff C, Heger D, De Pesters A, *et al*. Brain-to-text: decoding spoken phrases from phone representations in the brain. *Front Neurosci* 2015; **9**:217.

131. Guenther FH, Brumberg JS, Joseph Wright E, *et al*. A wireless brain-machine interface for real-time speech synthesis. *PloS One* 2009; **4**(12).

132. Schultz T, Wand M, Hueber T, *et al*. Biosignal-based spoken communication: a survey. *IEEE/ACM Trans Audio, Speech, Language Process* 2017; **25**(12): 2257–71.

133. Graves A, Fernández S, Gomez F, *et al*. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In: *Proceedings of the 23rd International Conference on Machine learning*, 2006, 369–76.

134. Moses DA, Leonard MK, Makin JG, *et al*. Real-time decoding of question-and-answer speech dialogue using human cortical activity. *Nat Commun* 2019; **10**(1): 1–14.

135. Miyawaki Y, Uchida H, Yamashita O, *et al*. Visual image reconstruction from human brain activity using a combination of multiscale local image decoders. *Neuron* 2008; **60**(5): 915–29.

136. Nishimoto S, Vu AT, Naselaris T, *et al*. Reconstructing visual experiences from brain activity evoked by natural movies. *Curr Biol* 2011; **21**(19): 1641–6.

137. Deng J, Dong W, Socher R, *et al*. ImageNet: a large-scale hierarchical image database. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*, Vol. **2009**. IEEE, 248–55.

138. Gatys LA, Ecker AS, Bethge M. Image style transfer using convolutional neural networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, 2414–23.

139. Tampuu A, Matiisen T, Ólafsdóttir HF, *et al*. Efficient neural decoding of self-location with a deep recurrent network. *PLoS Comput Biol* 2019; **15**(2):e1006822.

140. Rezaei MR, Gillespie AK, Guidera JA, *et al*. A comparison study of point-process filter and deep learning performance in estimating rat position using an ensemble of place cells. In: *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2018, 4732–5.

141. Xu Z, Wu W, Winter SS, *et al*. A comparison of neural decoding methods and population coding across

thalamo-cortical head direction cells. *Front Neural Circuits* 2019; **13**.

142. Li H, Fan Y. Interpretable, highly accurate brain decoding of subtly distinct brain states from functional MRI using intrinsic functional networks and long short-term memory recurrent neural networks. *Neuroimage* 2019; **202**: 116059.
143. Plis SM, Hjelm DR, Salakhutdinov R, *et al*. Deep learning for neuroimaging: a validation study. *Front Neurosci* 2014; **8**:229.
144. Han S, Huang W, Zhang Y, *et al*. Recognition of early-onset schizophrenia using deep-learning method. In: *Applied Informatics*, Vol. **4**. SpringerOpen, 2017, 1–6.
145. Yoo S-H, Woo S-W, Amad Z. Classification of three categories from prefrontal cortex using LSTM networks: fNIRS study. In: *2018 18th International Conference on Control, Automation and Systems (ICCAS)*. IEEE, 2018, 1141–6.
146. Batty E, Whiteway M, Saxena S, *et al*. BehaveNet: nonlinear embedding and Bayesian neural decoding of behavioral videos. In: *Advances in Neural Information Processing Systems* 2019, 15680–91.
147. Hofmann SM, Klotzsche F, Mariola A, *et al*. Decoding subjective emotional arousal during a naturalistic VR experience from EEG using LSTMs. In: *2018 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)*. IEEE, 2018, 128–31.
148. Garg A, Kapoor A, Bedi AK, *et al*. Merged LSTM model for emotion classification using EEG signals. In: *2019 International Conference on Data Science and Engineering (ICDSE)*. IEEE, 2019, 139–43.
149. Tripathi S, Acharya S, Sharma RD, *et al*. Using deep and convolutional neural networks for accurate emotion classification on DEAP dataset. In: *Twenty-Ninth IAAI Conference*, 2017.
150. Ciccarelli G, Nolan M, Perricone J, *et al*. Comparison of two-talker attention decoding from EEG with nonlinear neural networks and linear methods. *Sci Rep* 2019; **9**(1): 1–10.
151. de Taillez T, Kollmeier B, Meyer BT. Machine learning for decoding listeners' attention from electroencephalography evoked by continuous speech. *Eur J Neurosci* 2017; **51**(5): 1234–41.
152. Astrand E, Enel P, Ibos G, *et al*. Comparison of classifiers for decoding sensory and cognitive information from prefrontal neuronal populations. *PloS One* 2014; **9**(1).
153. Schulz M-A, Yeo T, Vogelstein J, *et al*. Deep learning for brains?: different linear and nonlinear scaling in UK biobank brain images vs. machine-learning datasets. *bioRxiv* 2019;757054.
154. Thomas RM, Gallo S, Cerliani L, *et al*. Classifying autism spectrum disorder using the temporal statistics of resting-state functional MRI data with 3D convolutional neural networks. *Front Psych* 2020; **11**:440.
155. Hennrich J, Herff C, Heger D, *et al*. Investigating deep learning for fNIRS based BCI. In: *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2015, 2844–7.
156. Dhawale AK, Poddar R, Wolff SBE, *et al*. Automated long-term recording and analysis of neural activity in behaving animals. *Elife* 2017; **6**:e27702.
157. Allen Institute for Brain Science. Allen Brain Observatory. http://observatory.brain-map.org/visualcoding, 2016 19 May 2020.
158. Xu K, Ba J, Kiros R, *et al*. Show, attend and tell: neural image caption generation with visual attention. *In: International Conference on Machine Learning* 2015;2048–57.
159. Sundararajan M, Taly A, Yan Q. Axiomatic attribution for deep networks. In: *Proceedings of the 34th International Conference on Machine Learning, Vol. 70*. JMLR. org, 2017, 3319–28.
160. Adebayo J, Gilmer J, Muelly M, *et al*. Sanity checks for saliency maps. *In: Advances in Neural Information Processing Systems* 2018;9505–15.
161. Olah C, Satyanarayan A, Johnson I, *et al*. The building blocks of interpretability. *Distill* 2018; **3**(3): e10.
162. The OpenAI Microscope. microscope.openai.com/models, (12 May 2020, date last accessed), 2020.
163. Kriegeskorte N, Douglas PK. Interpreting encoding and decoding models. *Curr Opin Neurobiol* 2019; **55**:167–79.
164. Paszke A, Gross S, Massa F, *et al*. PyTorch: an imperative style, high-performance deep learning library. *In: Advances in Neural Information Processing Systems* 2019, 8024–35.
165. Abadi M, Barham P, Chen J, *et al*. Tensorflow: a system for large-scale machine learning. In: *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, 2016, 265–83.
166. Kietzmann TC, McClure P, Kriegeskorte N. Deep neural networks in computational neuroscience. *bioRxiv* 2018: 133504.
167. Richards BA, Lillicrap TP, Beaudoin P, *et al*. A deep learning framework for neuroscience. *Nat Neurosci* 2019; **22**(11): 1761–70.
168. Hopfield JJ. Neural networks and physical systems with emergent collective computational abilities. *Proc Natl Acad Sci* 1982; **79**(8): 2554–8.
169. Zipser D, Andersen RA. A back-propagation programmed network that simulates response properties of a subset of posterior parietal neurons. *Nature* 1988; **331**(6158): 679–84.
170. Yamins DLK, DiCarlo JJ. Using goal-driven deep learning models to understand sensory cortex. *Nat Neurosci* 2016; **19**(3): 356.
171. Duarte J, Han S, Harris P, *et al*. Fast inference of deep neural networks in FPGAs for particle physics. *J Instrum* 2018; **13**(07):P07027.
172. Venieris SI, Bouganis C-S. Latency-driven design for FPGA-based convolutional neural networks. In: *2017 27th International Conference on Field Programmable Logic and Applications (FPL)*. IEEE, 2017, 1–8.
173. Teeters JL, Godfrey K, Young R, *et al*. Neurodata without borders: creating a common data format for neurophysiology. *Neuron* 2015; **88**(4): 629–34.
174. Walker B, Kording K. The database for reaching experiments and models. *PloS One* 2013; **8**(11):e78747.
175. Poldrack RA, Barch DM, Mitchell J, *et al*. Toward open sharing of task-based fMRI data: the OpenfMRI project. *Front Neuroinform* 2013; **7**:12.