Matthew J. Hall

Supervised by: Emilio Salazar-Gatzimas, Joana Soldado-Magraner, Matthew A. Smith

A convolutional neural network for generalized and efficient spike classification

Abstract

In neuronal recordings, analysis can classify electrode waveforms into spikes and noise. However, many automated sorting algorithms are highly variable in classification across different recordings and different implant areas. Here we trained a Convolutional Neural Network (CNN) on prelabeled waveforms collected from *in vivo* cortical recordings. The network, once trained, outputs a likelihood value that an input waveform should be a spike. To compare our network, we used a previous design from our lab using only fully connected layers, making a case for the benefit of convolutional layers for spike classification. We also compared classification across multiple cortical areas, showing improvement in classification accuracy and sensitivity to threshold parameters. Our classifier serves as a robust preprocessing step that can be applied to a diverse array of waveforms with predictions similar to that of a human sorter.

Introduction

Within the field of neuroscience, there is a need to understand the patterns of neural activity we observe in the brain in response to sensory stimuli, and how they encode information about the external world. However, before this analysis can be done, a researcher must identify and isolate neuronal activity itself, evident in changes in the neuron's membrane potential, known as an action potential or a spike. Spikes can be recorded from implanted electrodes, where the electrical potential or voltage signal can be attributed to the responses of neurons located in proximity to the electrode tip. The first step in processing the continuous voltage signal from an electrode is to identify which waveforms are neuronal spikes from random voltage fluctuations due to noise.

Traditionally, a researcher would have to hand-sort an entire neural recording to identify the spike waveforms of individual neurons and separate them from the inherent noise of electrophysiology. This process unfortunately relies on the spike classifier's proficiency in identifying complex features of data under different transforms. Additionally, waveform labels are biased to the particular sorter as different individuals may be more conservative in their sorting, leaving out some waveforms that may have been spikes, or may have been overly generous, including noise in the spike category. Numerous research groups have sought to facilitate and automate this process [8-9,11-13,17,20-21,23,27].

The use of machine learning algorithms involving artificial neural networks simplifies the process of spike/noise classification. A neural network can be adapted to take waveforms as inputs and classify them as either spikes or noise. To achieve this, a researcher can train the network with their own hand-labeled data or use a pre-trained version optimized with millions of examples. Our laboratory's previous effort created such an algorithm (named "Not A Sorter", or NAS), a single layer network able to separate spikes from noise within two different brain areas [12]. However, this simple network was not flexible in dealing with heterogeneous data types, performing inconsistently on waveforms from different brain areas and recording platforms.

To improve upon our classifier and make it more general, we implement an extension based on a convolutional neural network (CNN). Unlike the original NAS, the structure of a convolutional neural network includes trained filters that can find the temporal structure of the data. This allows the CNN to generalize across different spike shapes, with high classification accuracy on untrained spike data types, a known pitfall of our previous algorithm. Our classifier works as an important tool for preprocessing waveform data; our ultimate goal is to generalize our network to any neural recording.

Background

To understand how information is represented in the brain, analysis of the electrophysiological signals from neurons can give valuable information. These electrical signals, measured by implanted electrodes recording a local voltage potential, can be used to find the action potential of nearby neurons. By understanding the times at which a neuron fires an action potential, also known as a "spike," preceding stimuli can then be related to these spikes. After identifying the spikes within the electrical recordings, one has crucial information to understand how a specific event is encoded within the cortex. This identification of spikes from neural recordings is known as spike-sorting. The correlation of events to neuronal data has allowed for many practical applications of neuroscience such as the understanding of specific cortical areas' function, how information is passed between multiple areas, and the use of neural signals to control brain computer interfaces (BCIs) [4,19,22].

In classical applications, spike-sorting is done by human supervision of the data acquired from each electrode. Triggered by a release of neurotransmitters at the synaptic cleft, the action potential results in a measurable fluctuation in the membrane potential, where changes in voltage are recorded by nearby electrodes. Depending on the recording configuration, such as the density of the electrode array or sampling frequency, different methods are preferred to identify spike waveforms [15,18]. Some common traits analyzed for spike sorting are: spatial data of the electrodes where a change in potential is seen across many different electrodes, the temporal spacing of spike data which accounts for the inhibition of further spikes after a neuron has recently fired, the voltage of neurons crossing a threshold of spike activation, and dimensionality reduction of the waveform into specific components allowing for clustering of waveform types[8,13,20,27]. Through a researcher's previous experience and the application of a spike sorting method, one can determine either the single-unit data, identifying different spikes of specific neurons and noise within the recording, or multi-unit data, identifying spikes from noise, with spikes originating from multiple neurons.

However, by the nature of *in vivo* recordings, action potentials can vary greatly. The spatial placement of electrodes, or variation between subjects can cause vast differences in the shape of the spike waveform, making spikes difficult to identify from noise [10,26]. Issues such as differences in recording software, degradation of the implant signal with time, and electrode density contribute to the addition of noise and the distortion of spikes. This causes difficulty for even the most trained researchers in correctly identifying spikes. Although classic sorting methods, seen in Table 1 of the Appendix, have addressed many of these issues, their approaches vary in accuracy between sorters on the same datasets [15,18,26]. This causes the quality of the spikes identified to be dependent on both the skill of the researcher and the application of the spike-sorting algorithm.

To simplify the problem, recent advances in automated spike-sorting have used previously sorted data to improve decoding accuracy. By providing the algorithm with prelabeled data, one can tune the classifier toward their specific dataset. This prelabeled data, collected through previous spike sorting methods using human input, is seen as the ground truth for training the sorting algorithm. However, these methods are still inconsistent in their classification, often due to drifting of the electrode signal from movement, or spikes overlapping at the same time interval being misclassified [8-9,27]. As automated sorting algorithms use a ground truth which can vary from human input, these algorithms must be improved by using more complex classifiers and learning with better generalization to the dataset.

Most recently, neural networks have been trained on this ground truth data, allowing for supervised learning of waveform classification. Machine learning algorithms have been able to classify at or above the accuracy of classical sorters, without the need for manual intervention once trained [9,11-12,17,21,23]. These networks train weights and biases of specific nodes for a waveform input. Through gradient descent, the network's loss function is then optimized through training on hand-labeled spikes and comparing against a validation set. Outputs are placed through an activation function, giving classification of the waveform into a specified number of classes, with a binary classification for single-unit classification, or as many classes desired for multi-unit classification.

The most successful neural networks for spike classification have used convolutional layers to recognize features throughout the waveform [17,21,23]. Instead of only training specific node weights, these convolutional networks train filters to recognize the presence of shapes within the waveform, then use activations of each feature as the input into the fully connected layers for classification. CNNs have had success in other areas of signal processing such as arrhythmia recognition and environmental sound detection, proving their worth in noisy signal classification [2-3,14,17,28].

By applying a convolutional network design, we were able to achieve an automated method for spike-sorting with higher decoding accuracy on heterogeneous waveform data. This network design was able to classify spikes from various cortical areas and has been shown to generalize past common differences in spike waveform shape and threshold crossing. Our system is shown to classify a variety of waveforms, with better overall accuracy than our previous network and efficient classification speed.

Methods

In our study, we used recordings from multiple brain areas to train our network. With this diverse data set, we used hand-labeled spike waveforms to train a deep CNN network design, outlined in Figure 1.

Neural Recordings. The datasets used in this study were collected across two lab groups at Carnegie Mellon University and the University of Pittsburgh: namely the Smith and Batista laboratories. Data was collected from electrodes implanted in the cortex of non-human primates (NHP). We analyzed data from five male rhesus macaques (Macaca mulatta) that had previously been spike-sorted for ongoing experiments in each laboratory. Smith lab recordings from two96-electrode “Utah” arrays (Blackrock Microsystems, Salt Lake City, UT) and a 16-channel linear microelectrode arrays (U-Probe; Plexon, Dallas, TX) were band-pass filtered from 0.3 to 7,500 Hz, digitized at 30 kHz, and amplified by a Grapevine system (Ripple, Salt Lake City, UT). The interelectrode distance on the Utah arrays was 400 µm whereas the linear array was 150 or 200 µm. For each recording session, a voltage threshold (VT) was defined for each channel independently based on the root-mean-squared voltage (VRMS) of the waveforms recorded on that channel at the beginning of the session. Each time the signal crossed that threshold, a 52-sample waveform was captured. M1 data from the Batista lab data was also captured with the same 96-channel arrays (Blackrock Microsystems) recorded by 30-sample waveforms at 24414 Hz (Tucker Davis Technologies, Alachua, FL).

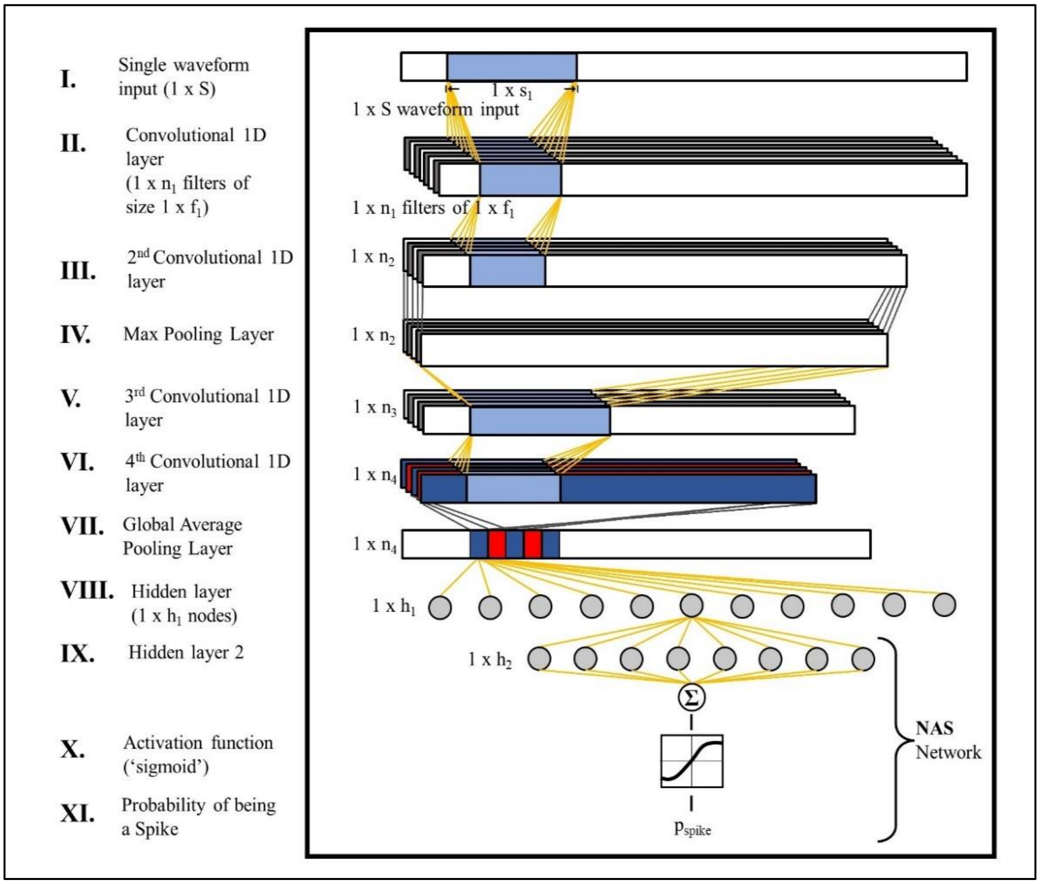

I. Single waveform input (1 x S)
1 x s_1
1 x S waveform input
II. Convolutional 1D layer (1 x n_1 filters of size 1 x f_1)
1 x n_1 filters of 1 x f_1
III. 2nd Convolutional 1D layer
1 x n_2
IV. Max Pooling Layer
1 x n_2
V. 3rd Convolutional 1D layer
1 x n_3
VI. 4th Convolutional 1D layer
1 x n_4
VII. Global Average Pooling Layer
1 x n_4
VIII. Hidden layer (1 x h_1 nodes)
1 x h_1
IX. Hidden layer 2
1 x h_2
Σ
X. Activation function (‘sigmoid’)
XI. Probability of being a Spike
p_{spike}
NAS Network

Figure 1. A simplified graphic of our deep-CNN network design. Our network takes a specified waveform input of size 1 x S and returns the probability that the waveform is a spike. Trained filters (convolutional layers) or trained weights and biases (hidden layers) are represented with yellow lines, while untrained pooling functions are represented by grey lines. The features are processed in two stages, with one max pooling layer (IV) and one global average pooling layer (VII). The average-pooled matrix serves as the input to a fully connected classifier network (VIII-XI). The last three stages (IX-XI) are equivalent to the previous NAS network classifier, where a waveform was classified using a fully connected network to return spike probabilities. Hence, our CNN can be seen as an extension of our previous NAS network.

Hand Labeled Spike Sorting. All data in this work was offline spike-sorted to identify well-defined single units for training. For Smith lab data, waveform segments were initially sorted into spike units and noise with a custom, off-line MATLAB spike-sorting algorithm that used an automated competitive mixture decomposition method [24]. These automated classifications for each recording session were subsequently refined manually by a researcher using custom MATLAB software (available at https://github.com/smithlabvision/spikesort). M1 data from the Batista lab was sorted by a researcher using the Plexon suite (Plexon, Dallas, TX). The researchers selected the classifications on each channel based on visualization of the overlaid waveform clusters, projections of the waveforms in principal component analysis (PCA) space, the interspike interval distribution of any potential single unit, and whether each potential single unit was present throughout the recording session. Although only a single researcher spike-sorted the waveforms from any particular session, the data in the training set, outlined in Table 2 of the appendix with 120 sessions total, collectively consisted of data spike-sorted by four different researchers. If multiple unique spike waveform shapes were present on a channel, the sorter would label those as different units. As some files were initially sorted into multi-unit spikes, all spikes were given the same label (1) and all noise waveforms were labeled (0) to train with single-unit classification.

Four distinct cortical areas were analyzed, motor area 1 (M1), visual area 4 (V4), prefrontal cortex (PFC), and the frontal eye fields (FEF). As M1 was recorded at a separate frequency and time period from other areas, waveforms were additionally processed by upsampling the signal from 24414 Hz to 30 kHz and removing a section at the end of the waveform to match the captured time segment to the rest of the data.

Network Implementation. For consistency across training sets, each area was given the same number of waveforms (500,000 waveforms with 52 samples) in µV units. Cortical area data was randomly selected from available recordings, ensuring a distribution of 70% spikes and 30% noise across all areas. This proportion was selected due to higher diversity within spike waveforms over noise. CNN classifiers of similar approach have also used higher percentages for non-noise examples, shown to increase the overall accuracy of the network and reduce the training time needed for convergence [2-3,17]. Training data was used to optimize the network hyperparameters through Keras [7], while validation data was used to confirm the validity of the hyperparameters selected during training. Testing data consisted of waveforms unused in training and validation sets, used to calculate the post-training accuracy of each network. Waveforms were split into each set at 60% training, 20% validation, and 20% testing with equal contributions of waveforms from each cortical area.

The network structure was initially compared across a variety of depths, finding the best trade off by comparing overfitting (seen by lower accuracy on the testing dataset) and loss function optimization (seen by the training accuracy on the validation dataset). Two networks were trained using TensorFlow and Keras within Python 3.8.10 [1,7]. Each network accepted a waveform segment of s samples long (s = 52) at 30 kHz (1.73 ms). Network training was cut off when the increase in validation accuracy was less than 0.001% between epochs. When excluding waveforms from a given area to train the network, such as in Figure 2B, the training and validation sets from that area were removed from training. The general network design of the NAS network was unchanged from Issar et al. 2020, which had a single hidden layer of 50 nodes. The CNN network design, outlined in Figure 1, consisted of four interconnected 1D CNN layers separated by a Max pooling layer (taking the max of each 1x3 matrix within each feature 1 x n2). A stride, the spacing between successive convolutions, of 1 was used on all layers, sliding filters and the pooling matrix across the entire array without skipping values. The output of the final CNN was then globally averaged (taking the average across each 1 x n4 feature), then placed through a two-layer hidden node classifier. Due to the complexity of the globally averaged CNN output, two hidden layers were needed for classification. However, it was found that a version of NAS with two hidden layers did not significantly improve generalized accuracy. The second hidden layer and activation function were the same size as the original NAS network, with 50 nodes fully summated into a sigmoid activation function. Hence, our CNN design can be seen as an extension of our previous NAS network. The final output of both networks, the probability of an initial waveform input being a spike (p-spike), was then used for network analysis. An initial gamma threshold, seen as the minimum probability needed for spike classification, was chosen as γ=0.2 for consistency with *Issar et al. 2020*. In cases where an optimal gamma value was chosen, possible gamma values were swept from $\gamma = 0.01$ to 0.99, shown in Figure 3, selecting the best value for average test set accuracy of the NAS network. Therefore, the CNN can be shown to improve upon NAS despite optimal tuning to the NAS network.

It is known that the most successful versions of neural networks have used CNNs in combination with human intervention, where the network serves as a selection set of the best channels for human sorting by PCA and K-means clustering [17,21,23]. However, as the goal of this paper was to fully automate spike classification and remove noise without the need for human intervention, our method placed an emphasis on the initial neural network classifier and implementation. We aimed to show the benefits of CNN classifiers over simple classifier networks like NAS, information which can be used to improve more complex CNN classifiers.

Hyperparameter selection for CNN. To decide on the hyperparameters for optimal training, different CNN sizes were compared by validation accuracy. Hyperparameter sweeps were compared using "Weights&Biases" for experiment tracking and dataset visualization [5]. Parameters tested included the number of features of each 1D CNN, trained filter size, max pool size, and training batch size. A visualized sweep of accuracy by network can be found in Supplementary Figure 1. The average for each hyperparameter from the best networks during the sweep was used for the final network trained, with consideration toward a network which converges in a reasonable timeframe (less than 24 hours on the given hardware) and an optimal parameter count to reduce overfitting.

Results

We trained a multi-layer Convolutional Neural Network (CNN) to evaluate *in vivo* neural recordings, outputting the likelihood each individual waveform was a spike. We used a diverse set of waveforms collected from four different cortical areas across six different implanted arrays with subjects performing a variety of tasks related to the recording region's function. Additionally, differences in recording equipment and human sorters provided a variety of classification for ground truth labels. All waveforms were assigned a binary label of spike (1) or noise (0) and split into designated training, validation, and testing sets. Our CNN was compared to our lab's previous NAS network trained with the same multi-area dataset. These networks were then compared using a variety of metrics to determine the benefit of convolutional layers in comparison to simple, hidden-layer classifier designs for waveform decoding. Both networks were trained for a sufficient number of epochs, denoted by a parameter search through the randomized training data, such that the validation accuracy was able to converge within a 0.001% change from the previous epoch.

Our network was considered a deep CNN due its complex network structure. Despite the increase in computational requirements of our CNN network from the simple NAS design by the addition of multiple layers requiring convolutions, multiplications, and summations, our CNN network was able to classify waveforms within real-time computing constraints on a modern CPU processor running at 4.2 GHz. Both networks were timed by classification speed over batches of 400 waveforms, then

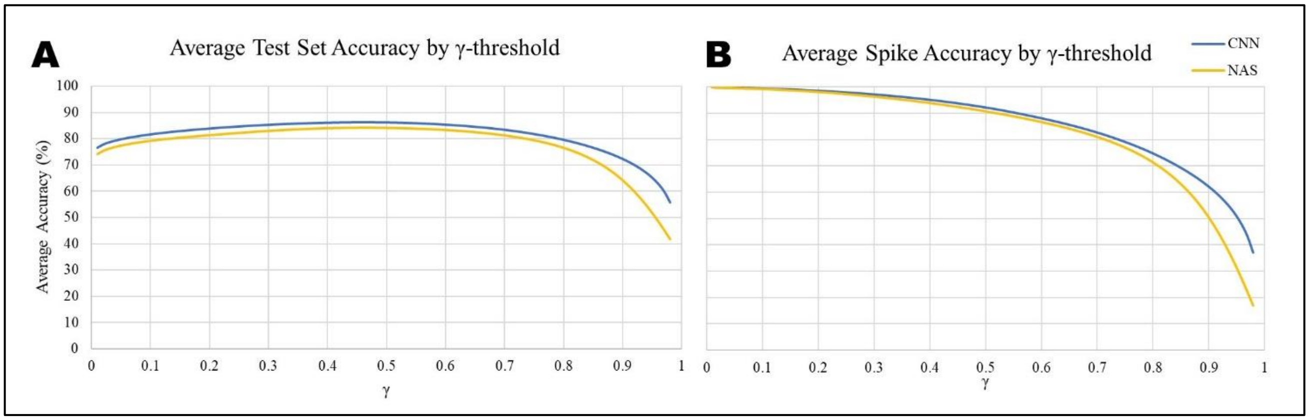


Figure 2. Gamma (γ) threshold values from 0.01 to 0.99 were swept and applied to each cortical area for classification. Accuracy was determined by the percentage of correct labels (at the given γ) when compared to the hand-labeled waveforms. (A) The accuracy on the entire test set of each area at each tested γ was averaged and plotted. We found no γ-threshold where the CNN had a lower accuracy than NAS. (B) The multi-area average accuracy of each network was also tested across only hand-labeled spikes, showing no γ-threshold where CNN dropped below the spike accuracy of NAS. Unaveraged accuracies across each area can be found in Supplement Figure 2. An optimal γ was selected by taking the γ-threshold where accuracy was the highest for each area, then averaging across all four areas. The γ-threshold for CNN lies in a range from 0.44-0.53 with an average of γ = 0.47. The γ-threshold for NAS lies on a wider range from 0.41-0.56 with an average of γ = 0.48, used as the best possible γ-threshold of NAS in figures 3 and 5.

averaged over 100 trials. Our CNN classified at an average speed of 3918 waveforms/second. Although this metric was nearly four times slower than the NAS classifier, it remained well under our simulation target with an expected recording rate of 100 waveforms/s [12].

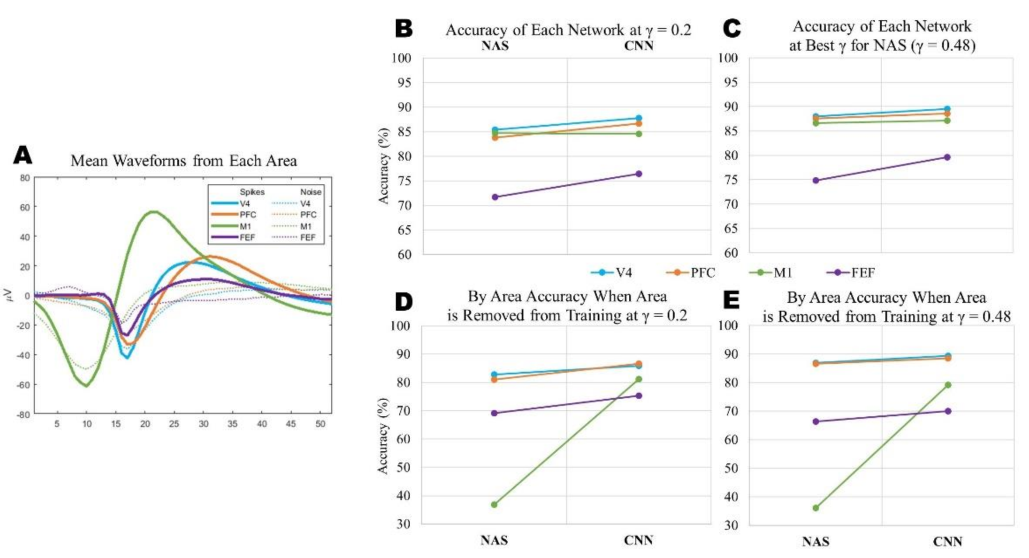


Figure 3. Changes in accuracy for each network across each cortical area tested. (A) Mean waveforms from identified spikes and noise from each area. (B-C) Accuracy of fully-trained networks at an appropriate γ-threshold, γ = 0.2 (B), and the best γ-threshold for NAS, γ = 0.48 (C). (D-E) Accuracy on the area removed from each networks' training set an appropriate γ-threshold, γ = 0.2 (D), and the best γ-threshold for NAS, γ = 0.48 (E). In all cases (every area B-E) except M1 at γ = 0.2, our CNN classified at higher accuracy than NAS. This effect is best demonstrated in D and E, where the classified area was completely removed from the training set.

Network Sensitivity to γ-threshold Selection. To test classification accuracy of either network, an appropriate gamma (γ) value must be selected. The γ-threshold was defined as the minimum spike probability (P_{spike}) required for a waveform to be considered a spike waveform. As shown in the previous NAS paper, low γ-thresholds such as γ = 0.2 assign most waveforms to classes a human sorter would deem appropriate. However, when training on a more diverse dataset, we noticed this γ-threshold could improve accuracy by choosing γ values closer to the midpoint of spike probability at 0.5. In Figure 2, we swept both classifier networks over γ-threshold values from 0.01 to 0.99. Accuracy was measured on the testing set from each cortical area by comparing the predicted labels of our CNN and NAS to hand-labeled classification, and averaging across all areas for each network. For accuracies separated by cortical area, see Supplemental Figure 2. Test set accuracy is summated in Figure 2B. As all waveforms are labeled as spikes at γ = 0.01, all spike waveforms are properly reported as spikes, with all noise waveforms being incorrectly labeled as spikes. The γ-threshold where each classifier achieved maximum accuracy was then averaged across all cortical areas, with NAS having a broader range of optimal γ over each area (γ range of 0.15 with an average γ of 0.48) than the CNN (γ range of 0.09 with an average γ of 0.47). This shows that the optimal γ is less sensitive to cortical area in our CNN over NAS. Values of γ = 0.2 and γ = 0.48 were chosen for further testing to compare networks with a previously published value [12] and the best possible value for NAS with this dataset.

By a 10-fold cross validation on each area, CNN was shown to be less variant across a handful of trials in either gamma. Variance for both networks was smallest at $\gamma = 0.2$, however the overall increases in accuracy caused selection of $\gamma = 0.48$ as the optimal gamma value for this dataset.

Comparison of Classification on Single Cortical Areas. Once our two γ-threshold values were chosen, we applied both thresholds to the P_{spike} output from each network. By comparing hand-labeled waveforms as a ground truth, we compared accuracy between each network. Our testing set consisted of waveforms from four different cortical areas (V4, PFC, M1, and FEF). Each cortical area had distinct, representative waveform shapes, as given by the mean waveform shape across all hand-labeled spike or noise waveforms from each area (see Fig 3A). All regions except M1 were threshold aligned at sample 16, set by the recording software on initial data collection. M1 could not be threshold aligned due to the recording software saving less data before the threshold crossing than the other areas.

V4 data had the closest mean waveform to all other areas, with a sharp depolarization (samples 13 to 16), repolarization (samples 16 to 27), and short recovery period (samples 27 to 40). In contrast, PFC spikes displayed a longer waveform past the threshold, with slower repolarization and recovery periods in comparison to V4. M1 data was collected with a different recording software, resulting in a different threshold alignment. Due to the large amplitude of M1 recordings and large oscillations distinct from noise, its waveform shape was considered as the most distinct for binary classification. In contrast, FEF data displayed the lowest signal (seen as the spike waveforms) to noise ratio, where labeled spikes are most difficult to classify from noise waveforms.

At best, our CNN had an average accuracy of 86.23%, ranging from 79.62% to 89.55% across all areas. By comparing the accuracy of each network (Fig 3B-C), our CNN improved upon accuracy in every cortical area with all γ-thresholds except M1 data at low spike selectivity ($\gamma = 0.2$). On average, our CNN had a 2.46% increase in accuracy for the previously established γ ($\gamma =0.2$, Fig 3B) and a 1.99% increase in accuracy at the best γ for NAS ($\gamma =0.48$, Fig 3C). Increases in accuracy were smallest on distinct waveforms, notably M1 and V4, while greater increases were seen in noisy FEF data.

To test our CNN's ability to generalize, we deliberately removed the waveforms for an entire area from the training and validation sets for each network. By hiding cortical areas, we sought to prove if our network was able to classify waveforms it had not previously seen by learning from other waveforms. We report this ability by showing the test accuracy on the area that was removed from the training set. Seen Figure 3D-E, differences in accuracy on the specific area removed were more drastic between networks. In all areas across both γ-thresholds, accuracy was higher for the CNN network. Although maximum accuracy was slightly below the fully trained networks in M1 and FEF, this was expected since these networks had not seen any data from the tested area. When compared to networks trained on all areas (henceforth "fully trained"), the drop in accuracy to this test was smaller for the CNN, highlighting its ability to generalize across data types. Between NAS and the CNN, the accuracy more than doubled for M1 data, showing the ability of the CNN to generalize past threshold alignment of waveforms. Additionally, maximum accuracy of our CNN at the best γ reached within 0.2% of fully trained networks for V4 and PFC, showing high accuracy in classification across similar areas in the training set (as mean waveforms are similar in shape and amplitude between V4 and PFC).

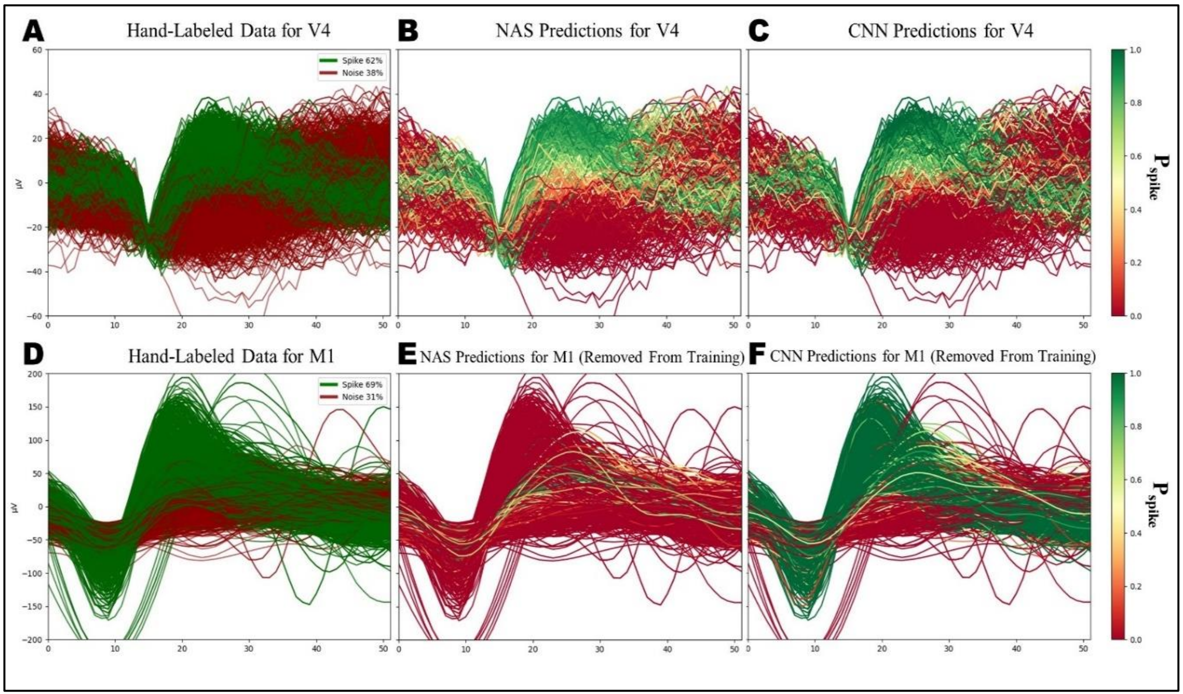


Figure 4. Comparison of hand-labeled waveforms to network classification by NAS and CNN. Predictions of spike probability from NAS and CNN were placed on a color map, where a value of 1 (seen as most green) is 100% likely to be a spike and a value of 0 (seen as most red) is 100% likely to be noise. (A-C) Each sorter classified V4 data from a single channel. (D-F) M1 data was classified from a random subset of the testing data by each sorter. Both networks, NAS and CNN, had M1 data removed during training and were trained using the other three areas (see Figure 3C-D).

Intensity of Classification in Comparison to Ground Truth Labels. As each network's output is the probability of a waveform being a spike, comparisons can be made to the degree of confidence in each prediction between each network. In Figure 4, waveforms from a selected channel in V4 (A-C) and a random subsection of M1 (D-F) were compared to best demonstrate differences in spike probability (P_{spike}) outputs. In Figure 4A and 4D, hand-labels for the waveforms are displayed with spikes in green and noise in red. Network spike prediction for NAS and CNN were then displayed as a bifurcating colormap between green (denoting 100% spike probability) and red (denoting 0% spike probability).

To demonstrate the ability of our CNN to classify waveforms with greater confidence than NAS, V4 waveforms were classified using networks trained on the entire training set. By qualitative inspection of Figure 4B and C, our CNN showed a greater confidence in spike waveforms by a darker green in the hand-labeled spike region than NAS. Additionally, a more bimodal gradient is seen between spike and noise waveforms, noted by a reduction of yellow (P_{spike} = 0.5) waveforms, showing the CNN's ability to classify at similar sensitivities to a human sorter. This effect is shown further in Supplemental Figure 3, where the distribution of P_{spike} is plotted for all waveforms. In this figure, the number of waveforms closest to 0 and 1 is much higher for CNN than NAS. Therefore, our CNN predicts waveforms with a better bimodal separation than NAS.

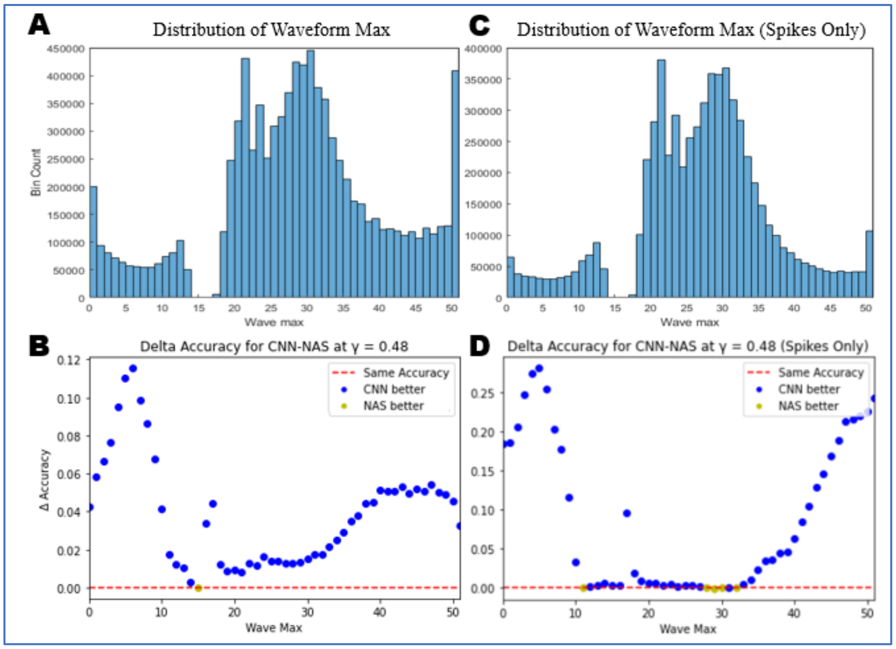


Figure 5. Differences in accuracy between CNN and NAS binned by the index of the max amplitude for all waveforms (A and B) or hand-labeled spike waveforms (C and D). (A) A histogram with each bin showing the count of all test waveforms with a peak max at one of the 52 samples. (B) Marker points showing the difference in accuracy for a given bin between CNN and NAS where a positive value shows a higher accuracy for the CNN in that bin (marked in blue) and a negative value shows a higher accuracy for NAS in that bin (marked in yellow). The best possible γ-threshold for NAS (γ = 0.48) was used for classification. Out of 52 possible max locations, the CNN performed better at 51, with greater differences seen at edge values. (C) A histogram showing the count of spike waveforms with a peak amplitude in each max location. (D) The CNN performed at a higher accuracy at 47 of 52 possible peak locations for spike only waveforms. Maximum differences in accuracy were more than twice that of B, with a similar effect seen of greater differences toward edge values.

To demonstrate the ability of our CNN to generalize to unseen waveforms, M1 data was classified with networks trained without M1 data. By qualitative inspection of Figure 4E and F, our CNN showed high confidence in spikes within the hand-labeled spike region. In contrast, the predictions of NAS showed high confidence on all spike waveforms, but erroneously assigned them a high noise probability. This further illustrates the CNN's ability to classify waveforms regardless of their presence in the training dataset.

Changes in Classification by Waveform Peak Location. Waveforms can vary in many ways, with one metric of comparison being their peak location. The peak of each waveform, defined as the index of the max amplitude, was calculated across all waveforms in the testing dataset. Waveform peaks were binned into each of 52 possible peak positions, then counted based on the number of waveforms in each bin (Fig 5A), or the number of spikes in each bin (Fig 5C). Binned accuracy for each network was then calculated using the best γ for NAS (γ = 0.48). The difference in accuracy between CNN and NAS was then calculated for each bin and plotted for all waveforms (Fig 5B) or spike waveforms (Fig 5D). Positive values, plotted in blue, denoted a higher accuracy for CNN for each bin, while negative values, plotted in yellow, denoted a higher accuracy for NAS.

For all waveforms (Fig 5B), all but one bin denoted a higher accuracy for the CNN. Similarly, for spike waveforms (Fig 5D), all but five bins denoted a higher accuracy for the CNN. Differences in accuracy increased with waveform peaks at the edges of the array, seen at peak maxes 0 to 10 and 33 to 51. The highest differences in accuracy were seen on spike only bins, showing this effect at greater degrees on spike waveforms.

Overall, this shows our CNN's ability to generalize across waveforms of varying peak maxes, when compared to NAS.

Discussion

By training a neural network classifier on a variety of waveform data, we assessed the benefits of a multi-layer convolutional neural network (CNN) in comparison to a previous one-layer network (NAS). To demonstrate differences in classification, our CNN and NAS predicted testing data from four distinct cortical areas, outputting the probability of each waveform being a neuronal spike. This spike probability output was then divided by reasonably justified γ-thresholds to assign binary labels of spike (1) and noise (0) to each waveform. By comparing the accuracy of each network's predictions to hand-labels of human sorters, we showed that our CNN was able to better generalize across our dataset and classify waveforms with higher accuracy than our NAS network.

Changes in Network Accuracy with Network Design. To test changes in accuracy across different networks, we chose a γ-threshold giving the best accuracy for the NAS network design, as well as a γ-threshold which predicts waveforms to classes a human sorter would deem appropriate. For both γ values selected, our CNN performed, on average, at higher classification accuracy than the NAS network. This effect was only counteracted by the high amplitude M1 data at a low γ-threshold, possibly explained by NAS's proved sensitivity to high SNR data, where spike waveforms are more distinguished from noise [12]. However, even when fitting γ to the best accuracy for NAS, our CNN performed at higher accuracy on every area, withan average increase in accuracy of 1.99%. Our CNN had an average accuracy of 86.23%, ranging from 79.62% to 89.55% across all areas. By showing the improvements our CNN has for binary classification, we concluded that convolutional network designs improve accuracy on diverse datasets of neuronal waveforms.

Furthermore, increases in accuracy were most significant when the area tested was not used to train each network. Data recorded on different hardware, seen as a shift in wave position for M1 data, experienced an over 200% increase in accuracy from NAS to our CNN. This effect could be explained by the convolutional filters in CNN layers moving across the entire waveform, extracting features for hidden layer classification. However, improvements on low-SNR FEF data, where spike waveforms overlap with noise from other areas, were minimal, likely due to a lack of training examples to differentiate FEF spikes from the noise of other areas. Additionally, accuracy of the CNN trained without an area were similar to fully trained networks for areas PFC and V4, where mean waveforms displayed similar shapes, and hereby similar training examples, between each region.

Confidence of Network Classification. To further investigate the improvements our CNN had over the previous NAS design, we compared the prediction of spike probabilities for each network to hand-labeled waveforms. When predicting subsets of our testing dataset, our CNN showed a more bimodal separation between spike and noise waveforms, where the spike probability output was closer to hand-labeled data than NAS. Separation was highest on waveforms with more distinctive peaks, where the most defined spike waveforms were given a near 100% spike prediction. As researchers must select an appropriate γ-threshold to assign binary labels, our CNN would be less sensitive to this value selection than NAS.

Generalization capability of networks. Testing data was also classified using networks trained without waveforms from the specified area. When M1 data was removed from training, our CNN was still able to classify with reasonable accuracy as opposed to NAS. This can be further seen in spike probability, where our CNN was still confident in the high amplitude M1 waveforms, showing similar separation to hand-labeled data. In contrast, NAS misclassified these waveforms as highly likely to be noise. This shows our CNN can generalize to waveforms despite differences in threshold alignment, whereas NAS would not recognize these spikes without trained examples. Additionally, distinctive waveforms were classified with high confidence, showing consistency in prediction of our CNN to hand-labels, even on untrained waveform shapes. Therefore, our CNN can differentiate waveforms with selectivity closer to that of human sorters.

Classification Stability Across Variable Waveform Peaks. As each network had a higher accuracy on spikes with greater peak amplitudes, we considered the changes in classification at different waveform peak locations. By comparing the accuracy of each network on the group of waveforms at each possible max amplitude, we found that accuracy changed with peak location. Across our entire testing set, the highest increases classification accuracy of our CNN over NAS was seen when waveform peaks fell within the first and last 20% of possible max locations. Additionally, our CNN had a higher accuracy than NAS at 98% of peak locations across all waveforms. Neuronal action potentials have been shown to vary in time, with most variability happening at the waveform peak [10]. As waveforms are prone to vary in peak location, the NAS network would misclassify more waveforms than our CNN in the presence of this variability. As our CNN uses features extracted from filters applied across the entire waveform, shifts in the location of these features are not as prone to affect the network's classification accuracy. In contrast, hidden-layer networks like NAS are trained to give the most characteristic amplitude values a higher weight. If those values vary, such as the shift of the waveform by recording setup or longer depolarization periods shifting the peak, the network would be unable to recognize waveforms. We showed that our CNN is more resistant to variations in waveforms, proving the value of CNN networks for classifying diverse neuronal data.

Limitations of our CNN Classifier. Although our CNN design was shown to be an improvement over the NAS network, limitations of this study may reduce its worth when compared to other classification methods. Due to findings from the previous NAS paper, training data was limited to channels with well-isolated units (SNR >2.2), and the effect of reduced noise in the training data was not further investigated. As this was shown to improve the classification accuracy of NAS [12], this may have also affected the accuracy of our CNN. Although the number of waveforms was balanced between each region, data variability in the number of subjects, recordings, and implant time differed greatly between regions, limited by the amount of data available.

Due to the deeper network structure and additional convolutional layers, the number of parameters for our CNN to train was much higher than NAS. Additionally, through hyperparameter sweeps seen in supplemental Figure 1, layers with a greater number of features were shown to improve accuracy. The addition of extra parameters may have improved accuracy; however, hidden layer networks with higher node counts did not substantially change decoding accuracy [12]. Our CNN was also limited in size due to computational hardware, as larger networks classified at slower speeds and were exponentially longer to train. Our network was kept at a size similar to other classifiers [14,17,23], with our final CNN implementation training at much slower rates than the NAS (from 10 minutes to 4 hours on the given hardware). However, as networks can be pre-trained for application, this was not considered as a significant limitation. Given our goal to show the benefits of CNN designs over simple, hidden-layer classifiers like NAS, we considered our network implementation sufficient.

Comparison to other CNN classifiers. In other studies, using CNNs for neural spike classification, training data often consisted of more selective examples across a larger pool of recordings [21,23], or simulated waveforms of varying complexity and background activity [11,17]. Given limitations in data availability across all cortical areas, we chose to limit the size of our data set to balance the amount of data from each region. Although other approaches allow for maximum possible accuracy of their final network by combining their network with other sorting methods, our study only aimed to identify the benefits of CNNs over simple network designs, despite the decoding accuracy of simple networks being comparable to human sorters. Given the high variability of classification accuracy across *in vivo* cortical recordings in both our results and other studies [21,23], understanding how to generalize classification to any spike waveform is key to creating a fully comprehensive spike sorter with reliable accuracy. Additionally, most neural network classifiers are used in conjunction with other offline spike-sorting methods, such as PCA, K-means, and other clustering methods, with network predictions serving as a pre-processing step for analysis [9,14,21,23]. As CNNs can be pre-trained and applied at speeds faster than real-time recording, automated classification has already been used in low latency decoding, required for real-time applications such as BCIs [22]. By investigating the benefits of more complex network classifiers, we hope to influence future network designs toward the best possible classification systems.

Extensions and Future Directions. Our CNN may provide better classification on more diverse datasets, however variation between regions could be improved. Each cortical area tested showed a different optimal γ-threshold which, although our CNN was less sensitive to γ than NAS, would still vary the accuracy of classification between areas. Additionally, regions with less-isolated spikes and noise were consistently classified at lower accuracies. This decrease is likely the result of the SNR threshold chosen for the training data, where training examples did not include spikes and noise of similar shape. However, including examples with poor isolation in the training data has been shown to reduce accuracy of the converged network [2-3,17].

Another possible direction yet unexplored is the use of preprocessing on the input waveforms. As it has been shown that the introduction of Gaussian noise greatly reduces accuracy [2-3,17], filtering inputs could provide better separation for frequencies present at both classes. For this study, we upsampled M1 data to match the frequency and recording time of the other areas. Although our CNN had inputs of the same frequency and time window, creating a network that accepts any input size or frequency could reduce computational time and support more recording setups. Features may also be found at frequencies lost due to the Nyquist limit when recording a discrete, 1D input. One classifier has addressed this by classifying estimated continuous waveforms with a 2D (waveform as an image) and 3D (images across channels) CNN network [21].

Overall, our CNN design classified a diverse selection of waveforms at appropriate levels when compared to a human sorter. We showed the value of a CNN over a simple, hidden-layer network at reducing output variability across changes in waveform shape and location. Furthermore, our CNN was shown to be less variable to threshold selection and can generalize classification to untrained waveforms. Given the ease of pre-training a CNN for classification, their use in automated and offline spike-sorting allows for generalized and efficient waveform classification.

Acknowledgements

We thank Samantha Smith for assistance with data collection and Deepa Issar for valuable insights on this project. We also thank the Smith and Batista laboratories for supplying the data. We also thank Dr. Steven Chase for reviewing this manuscript.

Disclosures

No conflicts of interest, financial or otherwise, are declared by the authors.

Author Contributions

M.J.H., E.S-G., J.S-M., and M.A.S. conceived and designed research; M.J.H. analyzed data; M.J.H., E.S-G., J.S-M., and M.A.S. interpreted results of experiments; M.J.H. prepared figures; M.J.H. drafted manuscript; M.J.H., E.S-G., J.S-M., and M.A.S. edited and revised manuscript; M.J.H.,E.S-G., J.S-M., and M.A.S. approved the final version of this manuscript.

Endnote

At the request of the authors, readers are herein alerted to the fact that additional materials related to this manuscript may be found at https://doi.org/10.5281/zenodo.5770652

Appendix

Table 1 - A comparison of Popular Sorting algorithms.

*Classical Sorters: Compared across SpikeForest's SYNTH_JANELIA dataset, used for accuracy figures in their paper

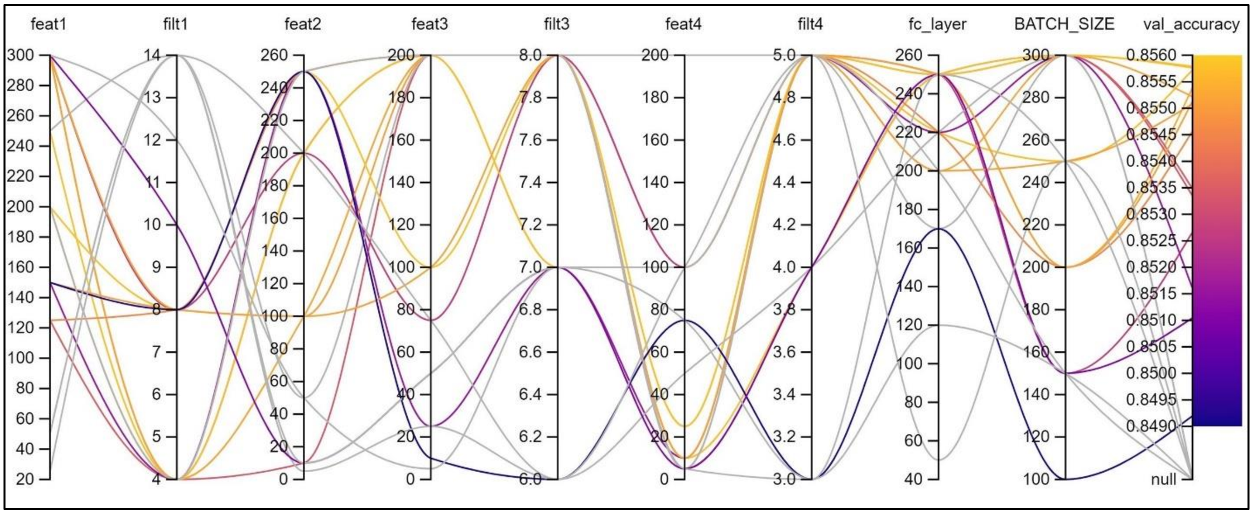
**ML Classifiers: Machine Learning Algorithms, using some form of supervised learning in all cases but J. Eom's autoencoder; Accuracies reported on either Wave_Clust or *in vivo* data

Sorting Algorithm	Main Contributor	Unique Appeal	Accuracy Reported
Classical Sorters*			
JRClust [13]	J.J. Jun	Using covariance of extracted features from density-based detection, best for high-density probes.	87%
KiloSort [20]	M. Pachitariu	Use of template matching at both spike detection and during spike clustering. Use of raw data allows fast processing.	84%
MountainSort4 [8]	J.E. Chung	Density-based clustering across a variety of electrode geometries with comparison through PCA space.	68%
SpyKingCircus [27]	P. Yger	A combination of template matching and density-based clustering, scalable across thousands of electrodes.	87%
ML Classifiers**			
1D-CNN [17]	Z. Li	A multi-unit classifier using 1D-CNNs to predict simulated data of varying noise levels.	95.16%-99.82%
Auto-Encoder Ensemble [9]	J. Eom	Use of three different auto-encoders to extract features of multi-unit data, then clustered. Tested with in-vivo and simulated data.	90.49%-100%
CNN-RNN [21]	M. Rácz	Use of a recurrent neural network (RNN) to extract spatiotemporal features from 2D and 3D CNN classifier outputs.	81.8%-97.7%
NAS (Not A Sorter) [12]	D. Issar	A simple, one-layer network to efficiently predict spike probability with γ-threshold selection. Initially designed for in-vivo data from two areas. Results reported from this study.	74.82%-87.9% at best γ
SpikeDeep [23]	M. Saif-ur-Rehman	A combination of supervised learning for Channel selection, a deep-CNN for background activity rejection, and unsupervised clustering for analysis.	86.95%-88.03%
WMsorting [11]	L. Huang	A semi-supervised solution using wavelet decomposition into frequency resolution, then classified using a muli-layer network. Minimal disturbance from noise, tested with simulated data.	86.45%-99.76%
Our CNN Model	M. Hall	A deep CNN network using a single probability output with γ-threshold sensitivity selection. Trained with in-vivo data from four areas. Results reported from this study.	79.62%-89.55% at best γ

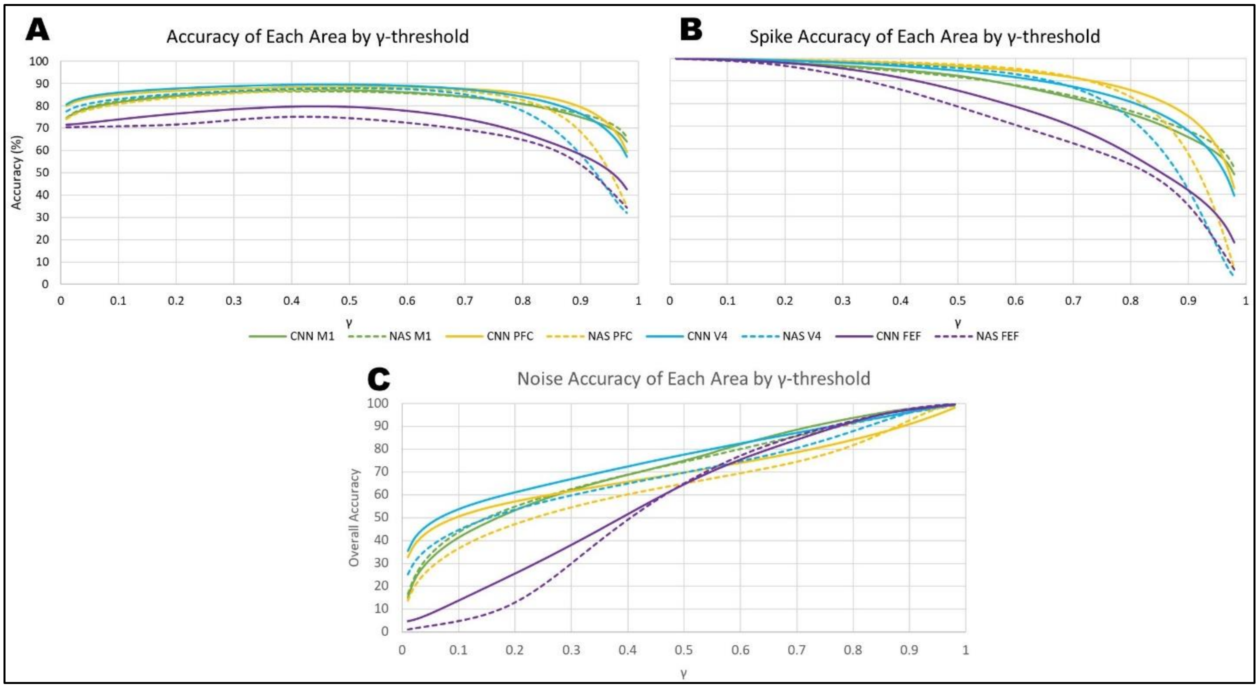
Table 2 - Dataset used to create training and testing sets. Training Data was filtered at SNR >2.2, found to increase accuracy of NAS, while testing data was unfiltered [12]. As found in *Li et al. 2020*, increases in 1D-CNN accuracy were found with increases in the number examples of spikes in the dataset [17]. Both training and testing data was selected to contain 70% spikes and 30% noise. Training data contained an equal number of waveforms from each area.

Cortical Area	***Recording Device***	***No. of Subjects***	***Time Range of Recordings***	***Total No. of Sessions***
M1	*Blackrock; 96 channel*	*1*	*3 days*	*2*
PFC	*Blackrock; 96 channel*	*1*	*8 days, 5 days*	*23*
V4	*Blackrock; 96 channel*	*2*	*12 days, 8 days*	*26*
FEF	*Plexon; 16 channel*	*3*	*30 days, 14 days, 341 days*	*69*

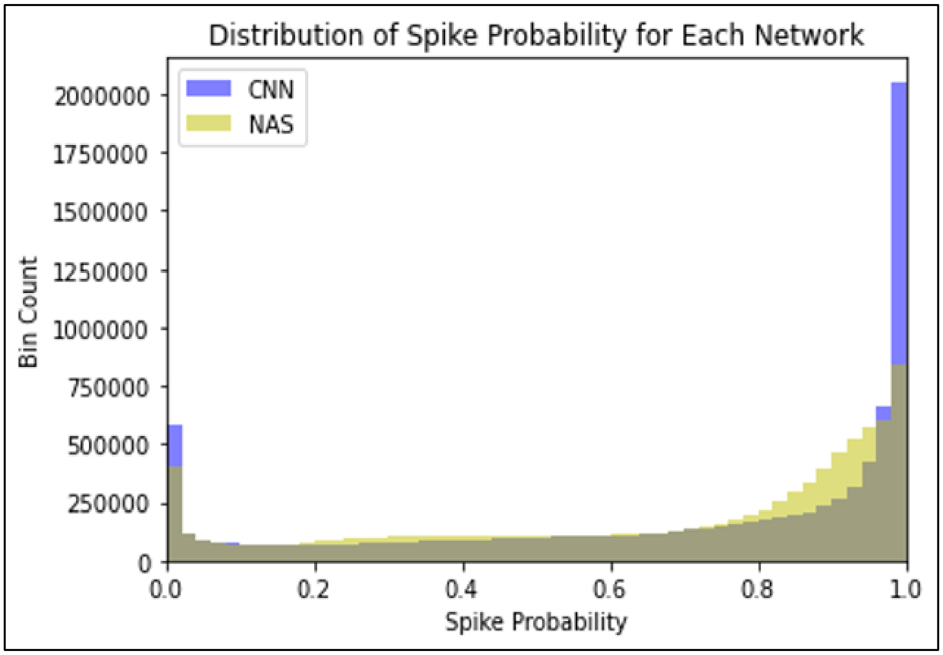
Supplementary Figures



Supplement Figure 1. Hyperparameter sweep to determine best values for the CNN network using Weights & Biases software. Each line denotes a single network trained on a subsection of the training set, where each network was given two epochs to converge. As our design used the same filter size for the first two convolutional layers, filt1 denotes the 1 x f1 size, filt3 and filt4 denoting the filter size for the last two convolutional layers. Feature sizes were tuned across every convolutional layer, as well as the first fully connected layer. The second fully connected layer was kept the same size as NAS for comparison, as well as the type of activation function. From the parameter sweep, similar values were chosen for the final network, however the number of features and fc_layer size were reduced to prevent overfitting to a large number of parameters. Hyperparameters were compared using their correlation with validation accuracy, showing the most importance of feature 3, batch_size, and filter size of 3 and 4. Our final CNN network reflected the average of most optimal networks (seen as most yellow) with the following parameters: feat1 = 150, filt1 = 4, feat2 =200, feat3 = 100, filt3 = 8, feat4 = 25, filt4 = 5, fc_layer (h1) = 200 giving a total of 222576 trainable parameters.



Supplement Figure 2. Accuracies of each individual area by sweeping γ-thresholds from 0.01 to 0.99. Both the NAS (plotted by dashed lines) and CNN networks (plotted by solid lines) were trained on hand-labeled data from all available areas as denoted in the Hand Labeled Spike Sorting section. Accuracy was averaged across all areas by network to obtain Figure 2. (A) In all cortical regions but M1, the CNN performed at higher overall accuracy than NAS. Within M1 accuracy sweeps, the CNN was able to perform at a higher max accuracy than NAS, while also having a higher accuracy at the best possible γ-threshold for NAS. (B) Spike-only waveform accuracies by area. (C) Noise-only waveform accuracies by area.



Supplement Figure 3. A histogram of predicted probabilities from each network. CNN (blue) was more confident in its predictions, shown by higher bin counts closest to 0 and 1. As hand labeled data was labeled with binary values, this bimodal separation in CNN spike predictions is closer to a human sorter than NAS.

References

(1)

Abadi, M.; Chu, A.; Goodfellow, I.; McMahan, H. B.; Mironov, I.; Talwar, K.; Zhang, L. Deep Learning with Differential Privacy. Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security 2016, 308–318. https://doi.org/10.1145/2976749.2978318.

(2)

Abdoli, S.; Cardinal, P.; Lameiras Koerich, A. End-to-End Environmental Sound Classification Using a 1D Convolutional Neural Network. Expert Systems with Applications 2019, 136, 252–263. https://doi.org/10.1016/j.eswa.2019.06.040.

(3)

Audhkhasi, K.; Osoba, O.; Kosko, B. Noise-Enhanced Convolutional Neural Networks. Neural Networks 2016, 78, 15–23. https://doi.org/10.1016/j.neunet.2015.09.014

(4)

Averbeck, B. B.; Latham, P. E.; Pouget, A. Neural Correlations, Population Coding and Computation. Nat Rev Neurosci 2006, 7 (5), 358–366. https://doi.org/10.1038/nrn1888.

(5)

Biewald, L. Experiment Tracking with Weights and Biases; Weights & Biases. Available at: http://wandb.com/.

(6)

Buccino, A. P.; Hagen, E.; Einevoll, G. T.; Häfliger, P. D.; Cauwenberghs, G. Independent Component Analysis for Fully Automated Multi-Electrode Array Spike Sorting. In 2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC); 2018; pp 2627–2630. https://doi.org/10.1109/EMBC.2018.8512788.

(7)

Chollet, F. Keras; 2015. Available at: https://keras.io.

(8)

Chung, J. E.; Magland, J. F.; Barnett, A. H.; Tolosa, V. M.; Tooker, A. C.; Lee, K. Y.; Shah, K. G.; Felix, S. H.; Frank, L. M.; Greengard, L. F. A Fully Automated Approach to Spike Sorting. Neuron 2017, 95 (6), 1381-1394.e6. https://doi.org/10.1016/j.neuron.2017.08.030.

(9)

Eom, J.; Park, I. Y.; Kim, S.; Jang, H.; Park, S.; Huh, Y.; Hwang, D. Deep-Learned Spike Representations and Sorting via an Ensemble of Auto-Encoders. Neural Networks 2021, 134, 131–142. https://doi.org/10.1016/j.neunet.2020.11.009.

(10)

Fee, M. S.; Mitra, P. P.; Kleinfeld, D. Variability of Extracellular Spike Waveforms of Cortical Neurons. Journal of Neurophysiology 1996, 76 (6), 3823–3833. https://doi.org/10.1152/jn.1996.76.6.3823.

(11)

Huang, L.; Ling, B. W.-K.; Cai, R.; Zeng, Y.; He, J.; Chen, Y. WMsorting: Wavelet Packets' Decomposition and Mutual Information-Based Spike Sorting Method. IEEE Transactions on NanoBioscience

2019, 18 (3), 283–295. https://doi.org/10.1109/TNB.2019.2909010.

(12)

Issar, D.; Williamson, R. C.; Khanna, S. B.; Smith, M. A. A Neural Network for Online Spike Classification That Improves Decoding Accuracy. Journal of Neurophysiology 2020, 123 (4), 1472–1485. https://doi.org/10.1152/jn.00641.2019.

(13)

Jun, J. J.; Mitelut, C.; Lai, C.; Gratiy, S. L.; Anastassiou, C. A.; Harris, T. D. Real-Time Spike Sorting Platform for High-Density Extracellular Probes with Ground-Truth Validation and Drift Correction; 2017; p 101030. https://doi.org/10.1101/101030.

(14)

Kiranyaz, S.; Avci, O.; Abdeljaber, O.; Ince, T.; Gabbouj, M.; Inman, D. J. 1D Convolutional Neural Networks and Applications: A Survey. Mechanical Systems and Signal Processing 2021, 151, 107398. https://doi.org/10.1016/j.ymssp.2020.107398.

(15)

Lefebvre, B.; Yger, P.; Marre, O. Recent Progress in Multi-Electrode Spike Sorting Methods. Journal of Physiology-Paris 2016, 110 (4, Part A), 327–335. https://doi.org/10.1016/j.jphysparis.2017.02.005.

(16)

Li, F.; Liu, M.; Zhao, Y.; Kong, L.; Dong, L.; Liu, X.; Hui, M. Feature Extraction and Classification of Heart Sound Using 1D Convolutional Neural Networks. EURASIP J. Adv. Signal Process. 2019, 2019 (1), 59. https://doi.org/10.1186/s13634-019-0651-3.

(17)

Li, Z.; Wang, Y.; Zhang, N.; Li, X. An Accurate and Robust Method for Spike Sorting Based on Convolutional Neural Networks. Brain Sciences 2020, 10 (11), 835. https://doi.org/10.3390/brainsci10110835.

(18)

Magland, J.; Jun, J. J.; Lovero, E.; Morley, A. J.; Hurwitz, C. L.; Buccino, A. P.; Garcia, S.; Barnett, A. H. SpikeForest, Reproducible Web-Facing Ground-Truth Validation of Automated Neural Spike Sorters. eLife 2020, 9, e55167. https://doi.org/10.7554/eLife.55167.

(19)

Murray, J. D.; Bernacchia, A.; Freedman, D. J.; Romo, R.; Wallis, J. D.; Cai, X.; Padoa-Schioppa, C.; Pasternak, T.; Seo, H.; Lee, D.; Wang, X.-J. A Hierarchy of Intrinsic Timescales across Primate Cortex. Nat Neurosci 2014, 17 (12), 1661–1663. https://doi.org/10.1038/nn.3862.

(20)

Pachitariu, M.; Steinmetz, N.; Kadir, S.; Carandini, M.; D, H. K. Kilosort: Realtime Spike-Sorting for Extracellular Electrophysiology with Hundreds of Channels; 2016; p 061481. https://doi.org/10.1101/061481.

(21)

Rácz, M.; Liber, C.; Németh, E.; Fiáth, R.; Rokai, J.; Harmati, I.; Ulbert, I.; Márton, G. Spike Detection and Sorting with Deep Learning. J. Neural Eng. 2020, 17 (1), 016038. https://doi.org/10.1088/1741-2552/ab4896.

(22)

Saad Zaghloul, Z.; Bayoumi, M. Adaptive Neural Matching Online Spike Sorting VLSI Chip Design for Wireless BCI Implants. In 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP); 2015; pp 977–981. https://doi.org/10.1109/ICASSP.2015.7178115.

(23)

Saif-ur-Rehman, M.; Ali, O.; Dyck, S.; Lienkämper, R.; Metzler, M.; Parpaley, Y.; Wellmer, J.; Liu, C.; Lee, B.; Kellis, S.; Andersen, R. A.; Iossifidis, I.; Glasmachers, T.; Klaes, C. SpikeDeep-Classifier: A Deep-Learning Based Fully Automatic Offline Spike Sorting Algorithm. J. Neural Eng. 2020. https://doi.org/10.1088/1741-2552/abc8d4.

(24)

Shoham, S.; Fellows, M. R.; Normann, R. A. Robust, Automatic Spike Sorting Using Mixtures of Multivariate t-Distributions. J Neurosci Methods 2003, 127 (2), 111–122. https://doi.org/10.1016/s0165-0270(03)00120-1.

(25)

Vargas-Irwin, C.; Donoghue, J. P. Automated Spike Sorting Using Density Grid Contour Clustering and Subtractive Waveform Decomposition. Journal of Neuroscience Methods 2007, 164 (1), 1–18. https://doi.org/10.1016/j.jneumeth.2007.03.025.

(26)

Wood, F.; Black, M. J.; Vargas-Irwin, C.; Fellows, M.; Donoghue, J. P. On the Variability of Manual Spike Sorting. IEEE Transactions on Biomedical Engineering 2004, 51 (6), 912–918. https://doi.org/10.1109/TBME.2004.826677.

(27)

Yger, P.; Spampinato, G. L.; Esposito, E.; Lefebvre, B.; Deny, S.; Gardella, C.; Stimberg, M.; Jetter, F.; Zeck, G.; Picaud, S.; Duebel, J.; Marre, O. A Spike Sorting Toolbox for up to Thousands of Electrodes Validated with Ground Truth Recordings in Vitro and *in vivo*. eLife 2018, 7, e34518. https://doi.org/10.7554/eLife.34518.

(28)

Yıldırım, Ö.; Pławiak, P.; Tan, R.-S.; Acharya, U. R. Arrhythmia Detection Using Deep Convolutional Neural Network with Long Duration ECG Signals. Computers in Biology and Medicine 2018, 102, 411–420. https://doi.org/10.1016/j.compbiomed.2018.09.009.