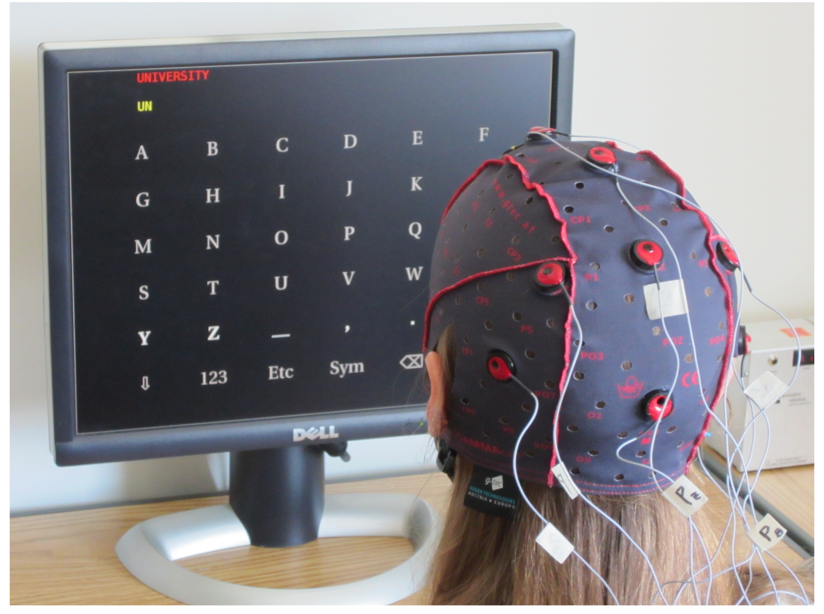
Detecting P300 ERPs with Convolutional Networks

Colorado State University
Brain-Computer Interfaces Lab.
CEBL3

Elliott Forney, Charles Anderson, Patricia Davies, William Gavin,
Marla Roll, Jewel Crasta, Brittany Taylor

Brainwaves Research Lab.
NSF Grant Number 1065513.

P300-Based Brain-Computer Interfaces 1



P300 Event-Related Potentials (ERPs) are waveforms that appear in time-locked Electroencephalography (EEG) signals following rare-but-expected stimuli.

This principal can be leveraged to construct Brain-Computer Interfaces (BCIs) by associating a number of stimuli with various instructions.

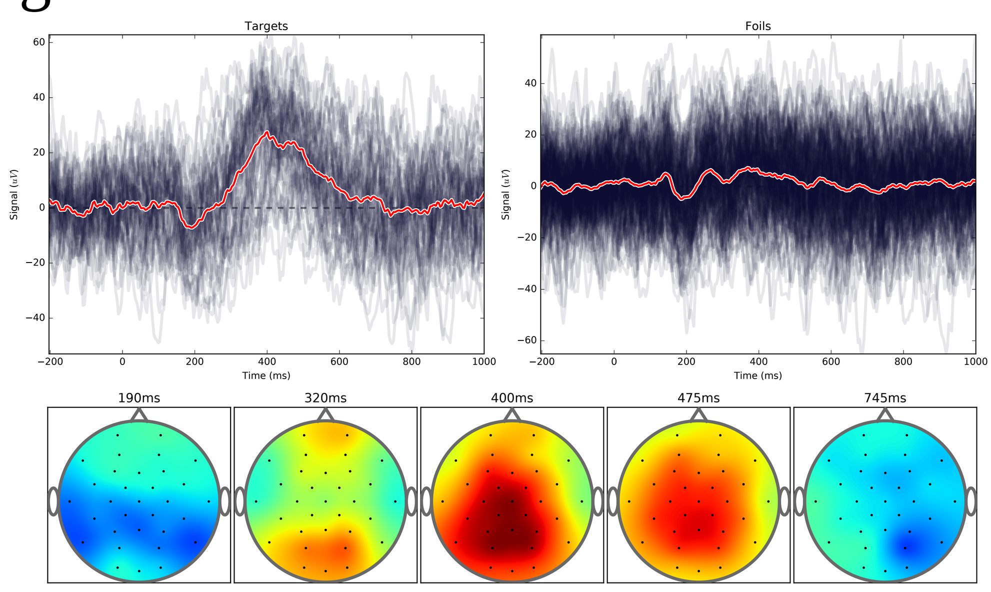
The stimuli are presented in a random order while the user attends to the desired stimuli. The BCI then attempts to identify the user's intent by classifying the signal as P300 (Target) or non-P300 (Foil).

For example, the P300-Speller type of BCI highlights the rows and columns of a virtual keyboard while the user attends to the letter they wish to type.

P300-based approaches can be very robust and useful for some people. However, current approaches typically require many repeated stimulus presentations.

Improvements in signal processing and machine learning methods may lead to increased communication rates and less user-fatigue by increasing the accuracy of P300 detection and reducing the number of stimulus presentations required.

Challenges in P300 Classification 2



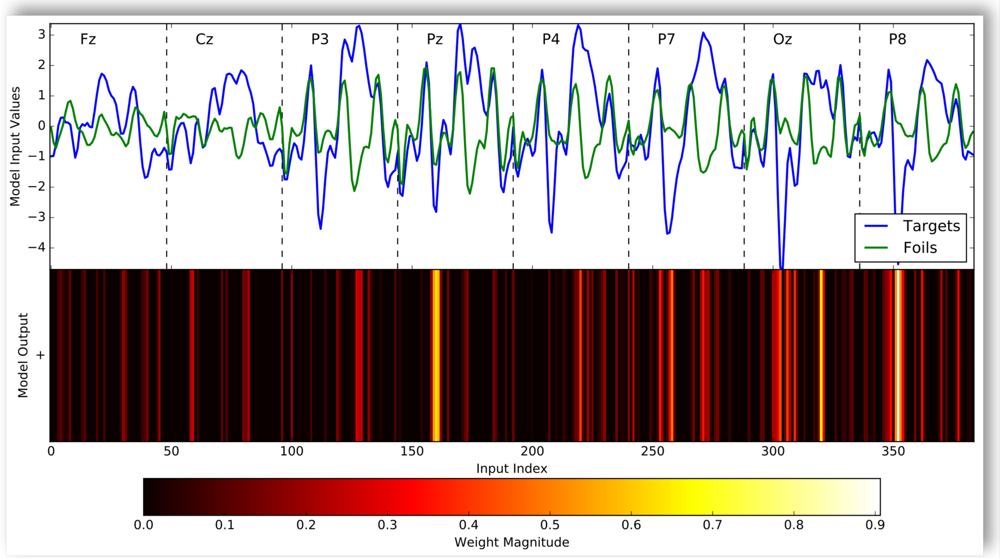
P300 ERPs contain sophisticated patterns that are both spatial (across the scalp) and temporal (over the course of time).

These patterns contain large amounts of noise and artifacts and can vary considerably across trials and participants.

A P300 trial also typically exists in a 2,000 - 30,000 dimensional space (sample rate x no. channels), yet only a few hundred trials can be recorded during a reasonable calibration session.

In other words, P300 classification is challenging due to noise, variability, under-sampling and high-dimensionality.

Current Approaches 3



Current approaches for classifying P300 ERPs typically concatenate the EEG channels across time.

This feature vector is then passed to any number of classification algorithms to assign a probability that the ERP is a target or foil.

Problems with noise, under-sampling and high-dimensionality are usually handled through aggressive dimensionality reduction (typically decimation) in combination with various techniques for model regularization.

Linear classifiers, such as regularized Linear Discriminant Analysis (LDA), have been particularly successful. This may, in part, be due to the fact that linear models are generally less susceptible to the types of problems encountered in P300 classification.

Motivation for using Deep Networks 4

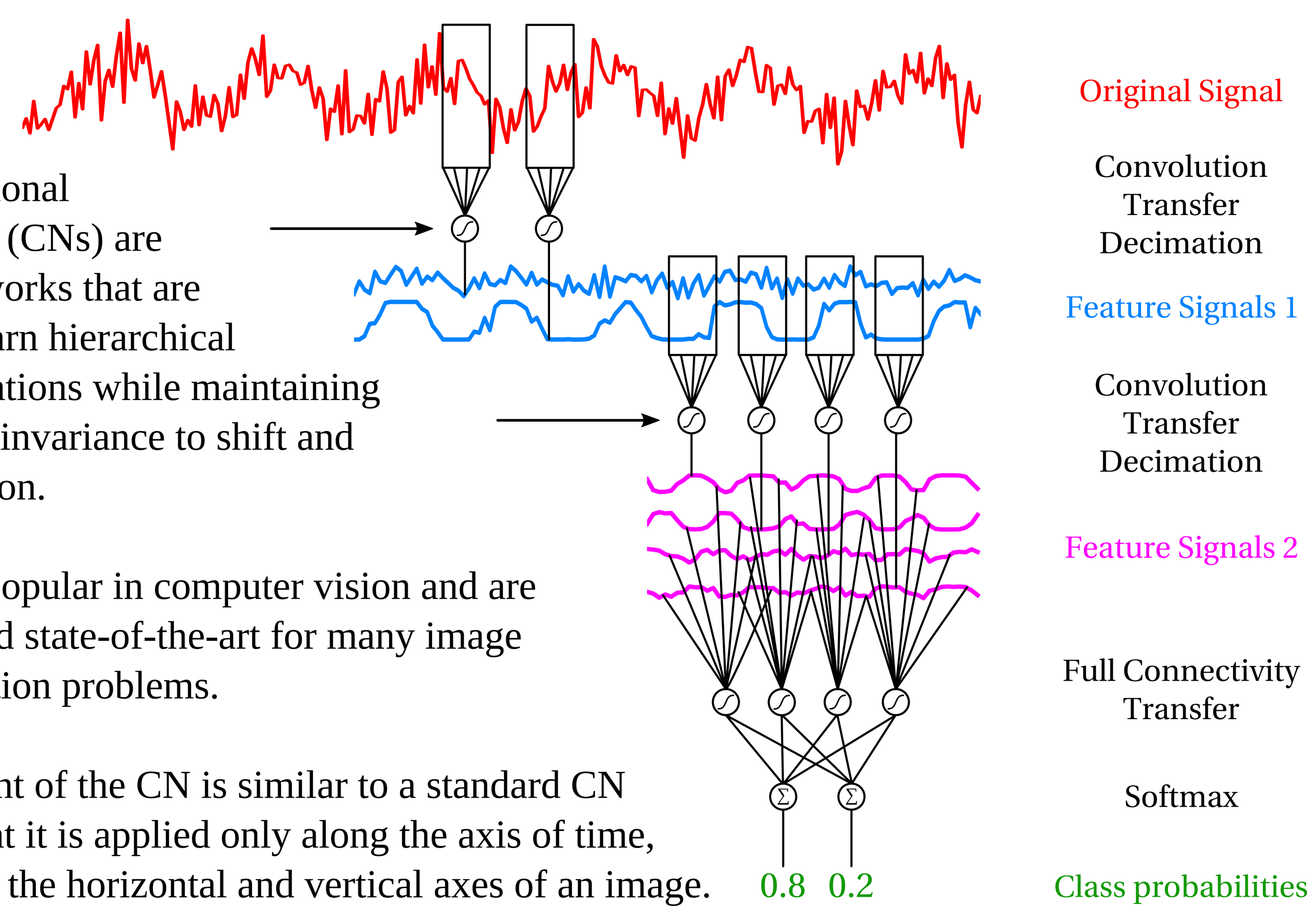
The current trend in machine learning is to move away from preprocessing, manual tuning and feature selection in favor of models that are able to automatically learn hierarchical, multi-scale representations.

These approaches typically involve multi-layer "deep" networks that rely on few prior assumptions about the data.

Avoiding prior assumptions may allow the model to exploit patterns that have not yet been identified or are not currently well-understood.

In addition to improved performance, analysis of deep networks may lead to new insights into the types and structure of patterns found in P300 ERPs.

Convolutional Networks for EEG Classification 5



Convolutional Networks (CNs) are deep networks that are able to learn hierarchical representations while maintaining a level of invariance to shift and deformation.

CNs are popular in computer vision and are considered state-of-the-art for many image classification problems.

Our variant of the CN is similar to a standard CN except that it is applied only along the axis of time, instead of the horizontal and vertical axes of an image.

Each layer of a CN consists of a number of computational units that perform a weighted convolution with the output of the previous layer followed by a hyperbolic tangent transfer function and downsampling.

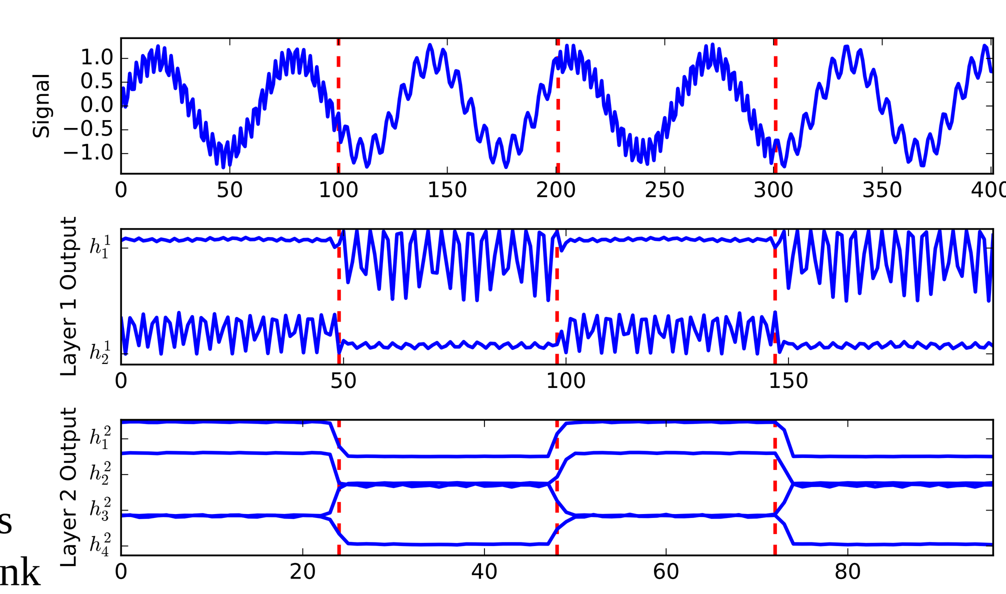
The output of the final convolutional layer is passed to a fully connected network with a non-linear layer followed by softmax readouts that indicate class membership probabilities.

Model weights are optimized using backpropagation, we use Scaled Conjugate Gradients.

Since the weights in each convolutional layer are applied to a sliding window over the output of the previous layer, these weights are considered to be shared. This results in fewer parameters to optimize relative to a fully-connected network.

This also allows the network to identify features, or events, that occur in the signal, regardless of the time at which they occurred, i.e., time-invariance.

Insights into the CN Architecture 6



Convolutional units have been compared to the receptive fields observed in biological neurons.

From this viewpoint, CN are designed to form a hierarchy of highly-specialized neurons with overlapping receptive fields.

An alternate interpretation views each convolutional layer as a bank of multi-variate, non-linear, finite impulse-response decimation filters.

From this viewpoint, each convolutional layer is viewed as a bank of filters that process the signal in a way that is useful for classification by the fully-connected layers.

Results 7

We tested our CN implementation on EEG data recorded during a serial oddball task.

Nine participants had no disabilities and recording took place in our EEG laboratory. Seven participants had severe motor impairments or progressive neurodegenerative diseases and recording took place in their homes. 60 target and 180 foil trials were recorded for each participant.

EEG data were recorded using a 32-channel Biosemi ActiveTwo amplifier. Eight channels, CZ, FZ, OZ, P3, P4, P7, P8, PZ, were used in all classification experiments.

All parameters were selected using a 10-fold cross-validation over the first 2/3 of the data and test performance was evaluated over the final 1/3.

We performed **single-trial classification** using a two-layer CN with 10 hidden units in each layer, a window width of 10 and a fully-connected layer with 10 hidden units.

Our CN was compared to LDA with shrinkage and a fully-connected Neural Network (NN) with 30 hidden units and early stopping.

The mean Area Under the ROC Curve (AUC) is 4.4% higher for CN than for LDA.

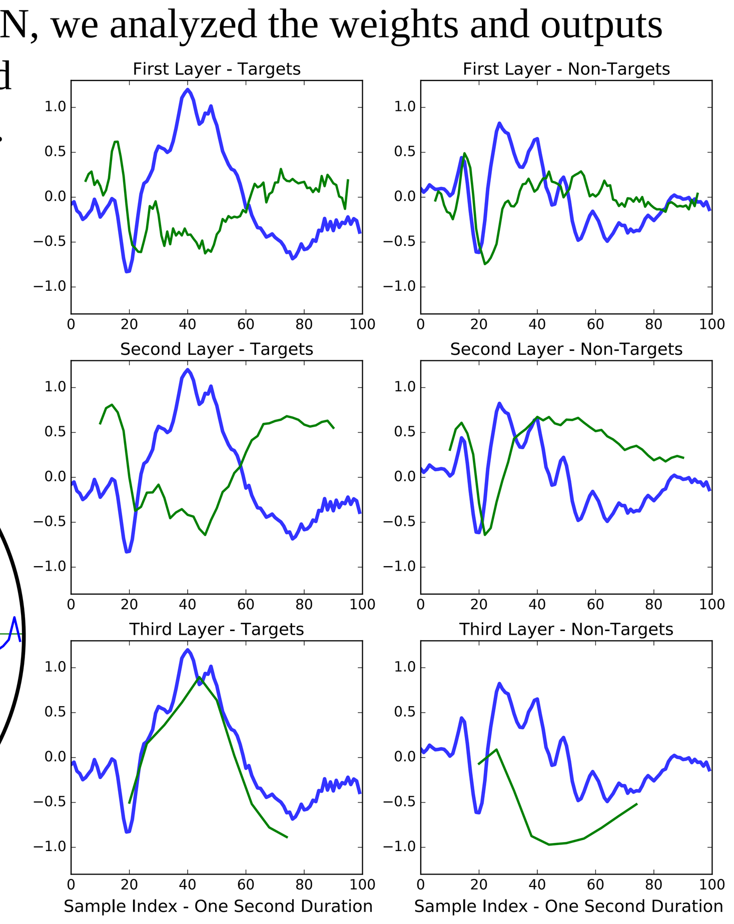
The mean Balanced Classification Accuracy was 3.7% higher for CN than for LDA.

	Subject	LDA AUC (BCA)	NN AUC (BCA)	CN AUC (BCA)
No Impairment	1	79.42 (68.33)	80.17 (70.00)	**89.33** (81.67)
	2	85.00 (80.83)	83.17 (71.67)	**87.17** (79.17)
	3	**88.83** (85.83)	88.17 (80.83)	88.58 (83.33)
	4	81.33 (70.83)	83.17 (72.50)	**93.00** (81.67)
	5	90.50 (77.50)	90.33 (78.33)	**91.33** (80.83)
	6	71.75 (65.83)	72.92 (67.50)	**78.00** (67.50)
	7	80.67 (76.67)	**82.42** (80.83)	80.50 (79.17)
	8	93.17 (83.33)	**95.25** (86.67)	91.75 (79.17)
	9	71.08 (63.33)	69.92 (65.83)	**79.08** (70.83)
Impairment	10	**68.36** (59.38)	65.49 (58.33)	65.49 (63.54)
	11	79.75 (71.67)	80.33 (70.00)	**81.25** (65.00)
	12	88.25 (80.83)	85.92 (77.50)	**92.58** (83.33)
	13	**60.50** (52.50)	55.83 (53.33)	58.75 (57.50)
	14	70.25 (64.17)	76.42 (65.83)	**82.33** (70.00)
	15	70.00 (63.33)	76.83 (69.17)	**87.67** (79.17)
	16	86.50 (80.00)	85.83 (80.00)	**88.58** (81.67)
	Mean	79.08 (71.52)	79.51 (71.77)	**83.46** (75.22)

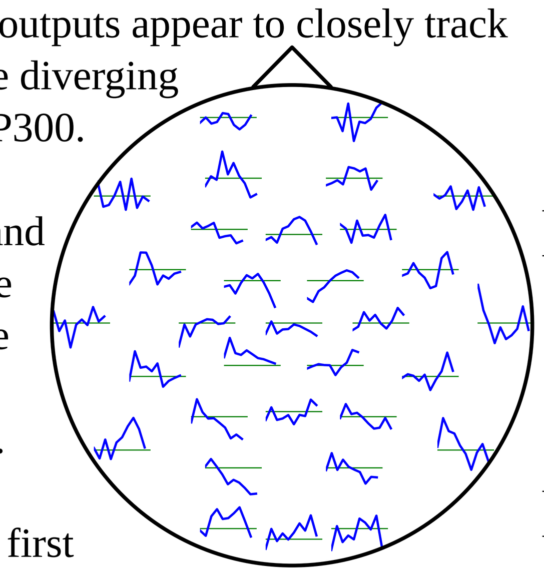
Mean AUC and BCA for the NN were less than one percent higher than LDA.

Interpretation and Analysis 8

In order to interpret the patterns learned by CN, we analyzed the weights and outputs of a second experiment using 32 channels and three convolutional layers with one unit each.



The outputs at each layer for a CN with three layers and a single hidden unit are shown to the right. The outputs appear to closely track the N200 while diverging slightly at the P300.



In the second and third layers, the outputs diverge considerably near the center.

Displaying the first layer weights on the scalp by channel reveals insights into the patterns that the CN uses to discriminate between target and foil.

Future Work 9

Fully exploring the hyper-parameters of CNs and incorporating various forms of regularization may further improve performance.

Interpretation and analysis remains challenging. Future experiments involve learning optimal inputs as well as time and frequency-domain analyses of the layer outputs.

Testing CNs in real-time implementations will ultimately determine their usefulness.