EC:447 Pattern Recognition and Machine Learning

Vowel-only versus consonant sound classification using EEG data corresponding to speech prompts

Submitted by:

Group K

Aditya Nishtala
Akarsh P
D R V Ramesh
Keerthan Shagrithaya
Sandeep Kotta

Department of Electronics and Communication Engineering
National Institute of Technology Karnataka
Surathkal,Karnataka, India

Contents

1 Features

Our data matrix size is 869 x 2790. Earlier, we were using the features extracted from EEG data during all the four stages (clearing, stimuli, thinking, speaking). Here, we consider only the features corresponding to the thinking phase only. That is, we have 11160 / 4 = 2790 features. There are 347 data points corresponding to vowel-only class and 522 corresponding to consonant class. We have divided the entire data into 80% : 20% for training and testing respectively.

We have used one one type of features.

- Features of importance : For the entire train data, we selected 100 features of importance from 2790, and divided by their corresponding standard deviations. The importance score was calculated as follows

$$I(k) = \frac{|\mu_k^{(1)} - \mu_k^{(2)}|}{|\sigma_k^{(1)} - \sigma_k^{(2)}|} \tag{1}$$

  where,
  $\mu_k^{(i)}$ is the mean of all the values corresponding to the k^{th} feature of points belonging to i^{th} class
  $\sigma_k^{(i)}$ is the standard deviation of all the values corresponding to the k^{th} feature of points belonging to i^{th} class

2 Results and Inferences

We use a Sequential model with SGD as optimizer, 128 as batch size, binary cross entropy as cost function, ReLu as activation unit for hidden layers and sigmoid for last layer unless specified otherwise, and run for 5000 epochs. We experiment with the number of hidden layers, neurons per layer, learning rate, activation unit type and regularization parameter.

Number of layers	Number of neurons	Regularization	Learning rate	Training accuracy	Validation accuracy	Test accuracy
2	200	0	0.01	100	50	50
1	200	0	0.01	100	50	50
2	5	0	0.01	83	49	49
1	5	0	0.01	80	61	61

Table 1: Effect of number of hidden layers

In the first case of 200 neurons per layer, we find that there is no difference between 1 and 2 hidden layers. Whereas, in the case of 5 neurons per layer, we clearly find that the validation/test accuracy drops with increase in the number of hidden layers. Therefore, one layer is better than two.

Number of layers	Number of neurons	Regularization	Learning rate	Training accuracy	Validation accuracy	Test accuracy
1	200	0	0.01	100	50	50
1	20	0	0.01	99.8	55	55
1	5	0	0.01	80	61	61
1	1	0	0.01	63	50	50

Table 2: Effect of number of neurons per layer

Fixing the number of hidden layers to be 1, we vary the number of neurons to twice the input dimension to 1 neuron. As we decrease the number of neurons, we find that the overfitness decreases as the training accuracy becomes closer to validation/test accuracy. But 1 neuron, produces very poor training and validation accuracies, so 5 neurons would be ideal.

Number of layers	Number of neurons	Regularization	Learning rate	Training accuracy	Validation accuracy	Test accuracy
1	200	0	0.01	100	50	50
1	200	0.1	0.01	71	56	56
1	200	1	0.01	60	60	60
1	5	0	0.01	80	61	61
1	5	0.1	0.01	66	51	51
1	5	1	0.01	60	60	60

Table 3: Effect of regularization

We experiment with three different regularization parameter. Without regularization, the classifier is overfit to a large extent in the case of 200 neurons. With 0.1 regularization parameter, the classifier is better with training and validation/test accuracies being comparable. But with value of 1, all accuracies are 60%, this classifier is properly fit but the accuracy is not great.

Number of layers	Number of neurons	Regularization	Learning rate	Training accuracy	Validation accuracy	Test accuracy
1	200	0.1	0.01	71	56	56
1	200	0.1	0.1	60	60	60
1	5	0.1	0.01	66	51	51
1	5	0.1	0.1	60	60	60

Table 4: Effect of learning rate

Bigger learning rates are making the classifier more fit but the accuracy is still only 60%.

Number of layers	Cost function	Number of neurons	Regularization	Learning rate	Training accuracy	Validation accuracy	Test accuracy
1	Binary cross entropy	5	0	0.01	80	61	61
1	MSE	5	0	0.01	83	53	53

Table 5: Effect of choosing cost function

The results are not very different in both the cases.

Number of layers	Activation unit	Number of neurons	Regularization	Learning rate	Training accuracy	Validation accuracy	Test accuracy
1	ReLu	5	0	0.01	80	61	61
1	Sigmoid	5	0	0.01	81	57	57

Table 6: Effect of choosing activation unit

The activation unit of the output layer is always kept as sigmoid since this is a binary classification problem. The results are not very different in both the cases.

3 Best result

The best result we obtained was for Sequential model with SGD as optimizer, 128 as batch size, binary cross entropy as cost function, ReLu as activation unit, 1 hidden layer with 5 neurons, no regularization, and 0.01 learning rate with 80% for training and 61% for validation/test accuracy.

References

[1] Duda, Richard O., Peter E. Hart, and David G. Stork. Pattern classification. Vol. 2. New York: Wiley, 1973.

[2] Walt, Stéfan van der, S. Chris Colbert, and Gael Varoquaux. ”The NumPy array: a structure for efficient numerical computation.” Computing in Science & Engineering 13.2 (2011): 22-30.

[3] Jones, Eric, Travis Oliphant, and Pearu Peterson. ”SciPy: open source scientific tools for Python.” (2014).

[4] Pedregosa, Fabian, et al. ”Scikit-learn: Machine learning in Python.” Journal of Machine Learning Research 12.Oct (2011): 2825-2830.

[5] Chollet, François. ”Keras”. Github (2015). `https://github.com/fchollet/keras`