CNN Sign Language Classification README

Nick Felix
nf2573

December 2024

1. One-Hot Encoding

One-hot encoding is a method of representing categorical labels as binary vectors. Each label is represented as a vector of zeros with a single one at the index corresponding to the label. This ensures that the labels are in a format the model can process. In Keras, one-hot encoding is implemented using the `to_categorical()` function.

2. Dropout and Overfitting

Dropout is a regularization technique used to prevent overfitting. It randomly disables (sets to zero) a fraction of neurons during training, forcing the model to learn more robust features and not rely on specific neurons. In Keras, dropout is implemented using the `Dropout()` layer.

3. ReLU vs. Sigmoid Activation Functions

ReLU (Rectified Linear Unit) outputs the input if it's positive and zero otherwise. It avoids the vanishing gradient problem often seen in sigmoid, which squashes inputs to a range between 0 and 1. ReLU is computationally simpler and works better for deeper networks.

4. Softmax in the Output Layer

The softmax function is necessary in the output layer of a classification model to convert raw logits into probabilities. It ensures that the output values sum to 1, making them interpretable as probabilities for each class.

5. Practical Calculation

(a) **Input image dimensions:** $100 \times 100 \times 1$

(b) **Convolution layer:** The output dimensions after applying a convolutional layer with 16 filters and a kernel size of 5×5:

$$\text{Output height/width} = \text{Input size} - \text{Kernel size} + 1 = 100 - 5 + 1 = 96$$

Therefore, the output dimensions are $96 \times 96 \times 16$.

(c) **MaxPooling layer:** After applying a max-pooling layer with pool size 2×2, the height and width are halved:

$$\text{Output height/width} = \frac{\text{Input size}}{\text{Pool size}} = \frac{96}{2} = 48$$

Therefore, the output dimensions are $48 \times 48 \times 16$.

6. Additional Information

Architecture Choices

My model uses two convolution layers followed by max-pooling layers to extract spatial features from the input images. I implemented a dense layer and a softmax output layer for classification. Dropout is applied to prevent overfitting.

Workflow

1. Preprocessed data by normalizing and reshaping the input images into the format required by the CNN.
2. Trained the CNN model using the training set while monitoring performance on a separate validation set.
3. Predictions were made on unseen data to evaluate the model's final performance and confirm its ability to generalize effectively.
4. Initially, I did not include a dropout layer, which resulted in a model that was fairly accurate but struggled to surpass a 90% accuracy threshold. After analyzing the results, I identified overfitting as a potential issue and added a dropout layer. This change improved the model's generalization and pushed the accuracy above 90%.