

20233508_DoHyoung Lee_term2s

20233508 DoHyoung Lee

1. What is the role and function of a single neuron in the predictive coding?

Single neuron receives input and operates prediction error computation. In single neuron, the error (epsilon) is defined as the gap of neural state and weighted sum of components from the post layer. Then, the gradient of each input of neuron is calculated. Furthermore, learning (parameter updating) occurs by assuming the error of the neuron and error from the post layer is same.

2. Why predictive coding is biologically more plausible?

Predictive coding model was suggested by mimicking sensory perception of organism. In sensory perception, lower-level sensory prediction is occurred by maximizing the probability when the lower-level sensory cortex receives signal from higher sensory cortex. In addition, high level cognition area calculated the gap of expectation and actual input from low level sensory cortex and controls it. In the predict coding model, inference occurs by summation of neural outputs of post layer, and weight update occurs by receiving error of previous layer. We can say that inference corresponds to feedback connection and parameter update corresponds to feedforward connection, and this trait can make predictive coding model is biologically plausible.

3. What is happening in the inference step?

The main goal of inference step is maximizing the probability, minimizing the loss, by modifying the expectation, neural state. By assuming prediction error is zero, "how neural state should be changed" can be calculated by (backward propagation of error from the post layer – error of the layer).

In the inference step, the gap of (neural state of specific layer "l") and (activation-functioned neural state of post layer + bias) is saved in "error[l]". Next, (delta entropy / delta neural state) can be represented with the "error term of the layer", error[l], and "multiplication of error and derivative of forward propagated signal of the previous layer", error[l+1], self.Weight and Neuronal_derivate_array[l], respectively, and it becomes "how neural state should be changed", where the learning rate is "inference_beta_parameter". This happens on every iteration. If the entropy decreases compared with the initial entropy, this means learning rate is too large to make the model fall into the global minima, so inference_beta_parameter is reduced by half.

4. What is happening in the parameter update step?

After the prediction is updated into the `Nerual_state_array` during the inference step, the gradients of weight and bias of every layer are calculated. $(\text{delta entropy} / \text{delta weight})$ is calculated as the multiplication of error of previous neuron and action-functioned post neuron around the weight. Similarly, $(\text{delta entropy} / \text{delta weight})$ corresponds to error of previous neuron. These values be gradient of gradient descent method, especially ADAM method, with the learning rate "self.lr".

5. Result and discussion

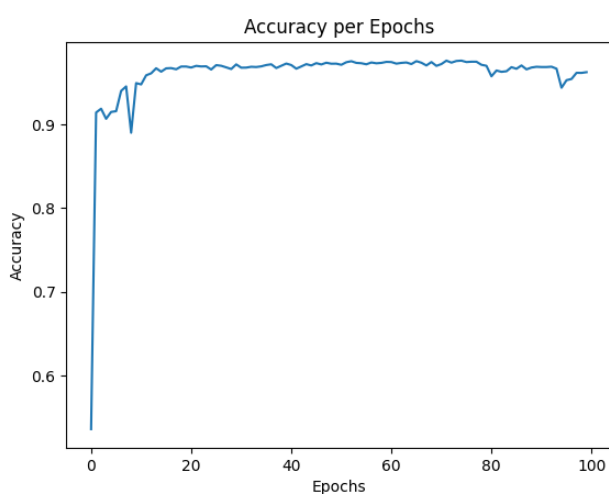


Figure 1 Learning Curve

After the training, the test accuracy peaked at 97.6 percent at almost 70~75th epoch, but it gradually decreased after 80th epoch. This is the effect of overfitting. In other words, 80th epoch is the borderline that model starts to fit the training dataset exclusively. To focus on the maximum accuracy, showing test accuracy over 97 percent means that the predictive coding model be successfully trained by the trained dataset.

There are several candidates to improve the model performance: early stopping, learning rate modification and neural layer modification etc. However, in my opinion, early stopping will be the best way. As figure 1 shows, model performance approaches to over 97% without any modification from the skeleton code, and we can see the learning curve successfully find the global minima with the learning rate 0.001. However, as mentioned above, the model starts to overfit after 80th epoch, so early stopping may be mandatory.