Lectures

Ilia Kamyshev

January 5, 2021

Contents

1 The Learning Problem

1.1 Introduction

It is assumed that training data is randomly generated. It is also assumed that there is a probability distribution P defined on Z. P is fixed and unknown for given problem. A sequence of labelled examples of the form (x, y) is presented to the neural network during training. For some positive integer m there is training sample:

$$z = ((x_1, y_1), \dots, (x_m, y_m)) = (z_1, \dots, z_m) \in Z^m$$

Random training sample of length m is an element of Z^m distributed according to the product probability distribution P^m. Let's denote the set of all functions the network can approximate as H. Then, an *error* of $h \in H$ will be:

$$error_P(h) = P\{(x, y) \in Z : h(x) \neq y\}$$

The sample error (*observed error*) is defined as:

$$error_z(h) = \frac{1}{m}|\{i : 1 \leq i \leq m \text{ and } h(x_i) \neq y_i\}|$$

Given h after the training is *hypothesis* (the error of this function should have minimum value), so:

$$opt_P(H) = \inf_{g \in H} error_P(g)$$

We may say that h is $\epsilon-$ good if for $\epsilon \in (0,1)$:

$$error_P(h) < opt_P(H) + \epsilon$$

1.2 Formal definition of learning

Let's denote δ as a *confidence parameter* to ensure that the learning algrotihm will be $\epsilon-$ good with probability at least $1-\delta$. Suppose that H maps from a set X to $\{0,1\}$. A learning algorithm L for H is a *function*:

$$L : \bigcup_{m=1}^{\infty} Z^m \rightarrow H$$

So if z is a training sample drawn randomly from distribution P^m, then the hypothesis $L(z)$ is such that:

$$error_P(L(z)) < opt_P(H) + \epsilon$$

In other words:

$$P^m\{error_P(L(z)) < opt_P(H) + \epsilon\} \geq 1 - \delta$$