Combating overfitting

Regularization

The idea behind regularization is to make sure that weight values are not too big. Let $E(W)$ be the error function to be minimized, where W is the weights vector. Let $r(W)$ be a function that becomes large if the weights are large. The regularized error function is formed as follows:

$$E_r(W) = E(W) + \lambda r(W)$$

the parameter λ is called **the regularization parameter**. Thus, minimizing $E_r(W)$ instead of $E(W)$ will have the effect of reducing the size of the weights. Observe the following:

$$\nabla E_r(W) = \nabla E(W) + \lambda \nabla r(W)$$

This shows how to calculate the gradient of the regularized error. The most common choice for $r(W)$ is:

$$r(W) = \frac{1}{2}\|W\|_2^2 = \frac{1}{2}\sum_i w_i^2, \quad \nabla r(W) = W = \begin{pmatrix} w_1, \\ \vdots \\ w_n \end{pmatrix}$$

Another (less common) choice is:

$$r(W) = \|W\|_1 = \sum_i |w_i|, \quad \nabla r(W) = \begin{pmatrix} \text{sgn}(w_1) \\ \vdots \\ \text{sgn}(w_n) \end{pmatrix}$$

The second choice has the advantage of producing sparse weights.

Dropouts

The algorithm

Consider a single node V. It is connected to nodes below and to nodes above. The dropout technique with probability parameter $0 < p \leq 1$ works as follows

Training: During training with a stochastic batch t flip a coin to determine drop or retain with probability p. With probability of $1 - p$ drop the node. This is the same as temporarily replacing all the weights of connections going into or coming out of the node with 0. (Going into is relevant for forward propagation, and coming out is relevant for back propagation.)

Testing: Replace each weight w on a connection from V to another node with the weight pw.

Why it works

View training as if multiple models are being trained separately to produce the same output. The final output is produced as an average of these models.

Why is the average better than a single model? Random models that overfit are expected to overfit in a different way. Therefore, averaging them is expected to reduce the overfitting.

Increasing (artificially) the amount of training data