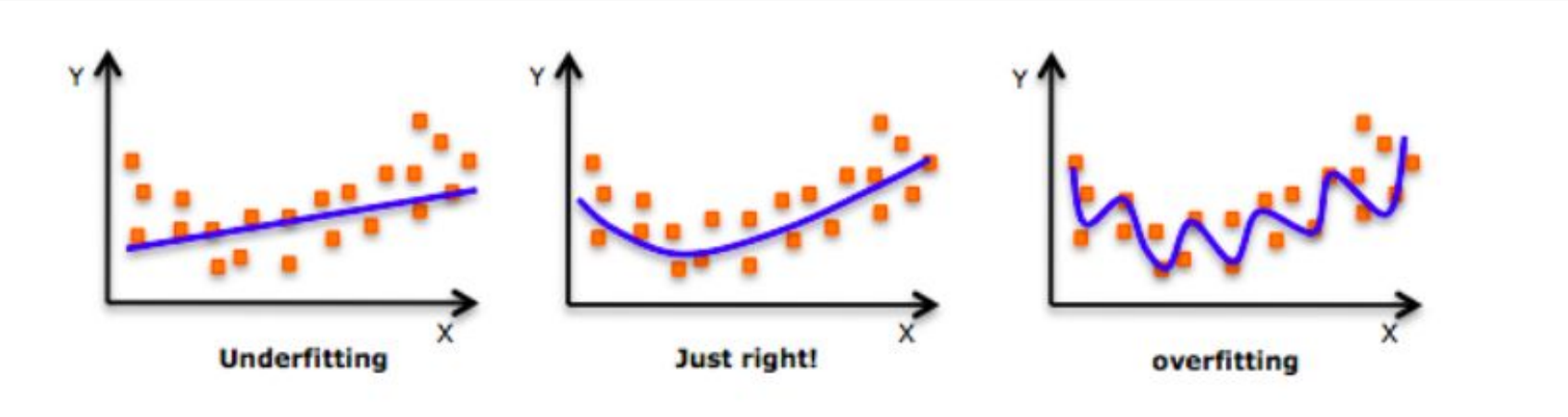
Complexity Regularization and Network Pruning

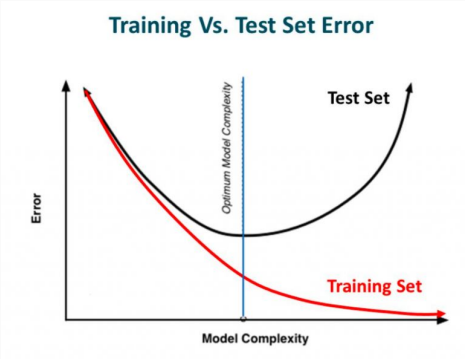
Bala Guga Gopal
I Mtech, Department of Electronics,
Cochin University of Science and Technology

As we move towards the right in this image, our model tries to learn too well the details and the noise from the training data, which ultimately results in poor performance on the unseen data.



Regularization

Regularization is a technique which makes slight modifications to the learning algorithm such that the model generalizes better. This in turn improves the model's performance on the unseen data as well.



the complexity of the model increases such that the training error reduces but the testing error doesn't

In the context of back-propagation learning, or any other supervised learning procedure for that matter, we may realize this tradeoff by minimizing the total risk, expressed as a function of the parameter vector w, as follows:

$$R(\mathbf{w}) = \mathscr{E}_{\text{av}}(\mathbf{w}) + \lambda \mathscr{E}_{c}(\mathbf{w})$$

The first term, $e_{av}(w)$, is the standard performance metric, which depends on both the network (model) and the input data. In back-propagation learning, it is typically defined as a mean-square error

The second term, $e_{c}(w)$, is the complexity penalty, where the notion of complexity is measured in terms of the network (weights) alone;

λ as a regularization parameter, which represents the relative importance of the complexity-penalty term with respect to the performance metric term.

Weight Decay procedure

THE COMPLEXITY PENALTY TERM IS DEFINED AS THE SQUARED NORM OF THE WEIGHT VECTOR W (I.E., ALL THE FREE PARAMETERS) IN THE NETWORK, AS SHOWN BY

$$\begin{aligned}\mathscr{E}_c(\mathbf{w}) &= \|\mathbf{w}\|^2 \\ &= \sum_{i \in \mathscr{C}_{\text{total}}} w_i^2\end{aligned}$$

where etotal the set refers to all the synaptic weights in the network. This procedure operates by forcing some of the synaptic weights in the network to take values close to zero, while permitting other weights to retain their relatively large values

The weights of the network are grouped roughly into two categories:
(i) weights that have a significant influence on the network's performance;
(ii) weights that have practically little or no influence on the network's performance.

- The weights in the latter category are referred to as excess weights.
- In the absence of complexity regularization, these weights result in poor generalization by virtue of their high likelihood of taking on completely arbitrary values or causing the network to overfit the data in order to produce a slight reduction in the training error
- The use of complexity regularization encourages the excess weights to assume values close to zero and thereby improve generalization

Hessian-Based Network Pruning: Optimal Brain Surgeon

The basic idea of an analytic approach to network pruning is to use information on second-order derivatives of the error surface in order to make a trade-off between network complexity and training-error performance.

The starting point in the construction of such a model is the local, approximation of the cost function $\mathscr{E}_{av}$ by using a Taylor series about the operating point, described as

$$\mathscr{E}_{av}(\mathbf{w} + \Delta\mathbf{w}) = \mathscr{E}_{av}(\mathbf{w}) + \mathbf{g}^T(\mathbf{w})\Delta\mathbf{w} + \frac{1}{2}\Delta\mathbf{w}^T\mathbf{H}\Delta\mathbf{w} + O(\|\Delta\mathbf{w}\|^3)$$

where $\Delta\mathbf{w}$ is a perturbation applied to the operating point $\mathbf{w}$ and $\mathbf{g}(\mathbf{w})$ is the gradient vector evaluated at $\mathbf{w}$. The Hessian is also evaluated at the point $\mathbf{w}$, and therefore, to be correct, we should denote it by $\mathbf{H}(\mathbf{w})$. We have not done so in Eq. (4.97) merely to simplify the notation.

The requirement is to identify a set of parameters whose deletion from the multilayer perceptron will cause the least increase in the value of the cost function .

Extremal Approximation - We assume that parameters are deleted from the network only after the training process has converged (i.e., the network is fully trained). The implication of this assumption is that the parameters have a set of values corresponding to a local minimum or global minimum of the error surface. In such a case, the gradient vector g may be set equal to zero, and the term $g^T!w$ on the right-hand side of equation may therefore be ignored; otherwise, the saliency measures (defined later) will be invalid for the problem at hand.

Quadratic Approximation - We assume that the error surface around a local minimum or global minimum is “nearly quadratic.” Hence, the higher-order terms in, may also be neglected.

Under these two assumptions, equation

$$\mathscr{E}_{av}(\mathbf{w} + \Delta\mathbf{w}) = \mathscr{E}_{av}(\mathbf{w}) + \mathbf{g}^T(\mathbf{w})\Delta\mathbf{w} + \frac{1}{2}\Delta\mathbf{w}^T\mathbf{H}\Delta\mathbf{w} + O(\|\Delta\mathbf{w}\|^3)$$

is simplified as

$$\begin{aligned}\Delta\mathscr{E}_{av} &= \mathscr{E}(\mathbf{w} + \Delta\mathbf{w}) - \mathscr{E}(\mathbf{w}) \\ &= \frac{1}{2}\Delta\mathbf{w}^T\mathbf{H}\Delta\mathbf{w}\end{aligned}$$

This provides the basis for the pruning procedure called optimal brain surgeon (OBS). The goal of OBS is to set one of the synaptic weights to zero in order to minimize the incremental increase in given in Eq. Let $w_i(n)$ denote this particular synaptic weight. The elimination of this weight is equivalent to the condition

$$\mathbf{1}_i^T\Delta\mathbf{w} + w_i = 0$$

where $\mathbf{1}_i$ is the *unit vector* whose elements are all zero, except for the ith element, which is equal to unity. We may now restate the goal of OBS as follows:

> *Minimize the quadratic form* $\frac{1}{2}\Delta\mathbf{w}^T\mathbf{H}\Delta\mathbf{w}$ *with respect to the incremental change in the weight vector,* $\Delta\mathbf{w}$*, subject to the constraint that* $1_i^T\Delta\mathbf{w} + w_i$ *is zero, and then minimize the result with respect to the index* i*.*

Thanks!