6.4 Error Surfaces

Li Yu

Hongda Mao

Joan Wang

Error Surfaces

- Backpropagation is based on gradient descent in a criterion function, we can gain understanding and intuition about the algorithm by studying error surfaces------the function J(***w***)
- Some general properties of error surfaces

➢ Local minima

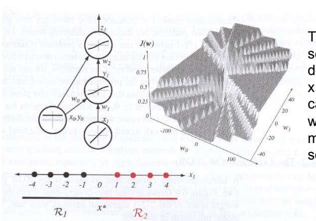
if there are many local minima plague the error landscape, then it is unlikely that the network will find the global minimum.

➢ Presence of plateaus

Regions where the error varies only slightly as a function of weights.

- We can explore these issues in some illustrative systems

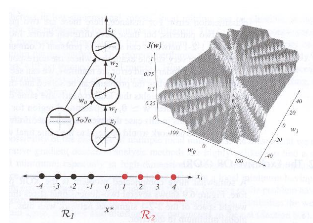
Some small networks (1)



The data shown are linearly separable, and the optimal decision boundary, a point near x_1=0, separates the two categories. During learning, the weights descend to the global minimum, and the problem is solved.

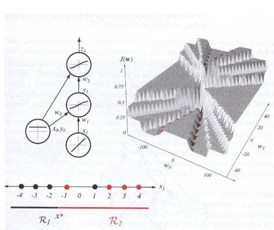
The simplest three-layer nonlinear network, here solving a two-category problem in one dimension.

Some small networks (1) cont'd



Here the error surface has a single minimum, which yields the decision point separating the patterns of the two categories. Different plateaus in the surface correspond roughly to different numbers of patterns properly classified; the maximum number of such misclassified pattern is four in this example.

Some small networks (2)



Note that overall the error surface is slightly higher than before because even the best solution attainable with this network leads to one pattern being misclassified.

The patterns are not linearly separable; there are two forms of minimum error solution; these correspond to -2<x*<-1 and 1<x*<2, in which one pattern is misclassified.

Conclusions

- From these very simple examples, where the correspondences among weight values, decision boundary, and error are manifest, we can see how the error of the global minimum is lower when the problem can be solved.
- The surface near w≈0 , the traditional region for starting learning, has high error and happens in this case to have a large slope

+ The Exclusive-OR(XOR)

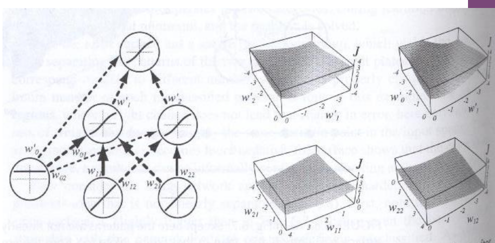


FIGURE 6.9. Two-dimensional slices through the nine-dimensional error surface after extensive training for a 2-2-1 network solving the XOR problem.

+ The Exclusive-OR(XOR) cont'd

- The error varies a bit more gradually as a function of a single weight than does the error in the networks solving the problem in the last two examples. This is because in a large network any single weight has on average a smaller relative contribution to the output.
- The error surface is invariant with respect to certain discrete permutations. For instance, if the labels on the two hidden units are exchanged, and the weight values changed appropriately, the shape of the error surface is unaffected.

+ Larger Networks

- For a network with many weights solving a complicated high-dimensional classification problem, the error varies quite gradually as a single weight is changed.
- Whereas in low-dimensional spaces, local minima can be plentiful, in high dimension, the problem of local minima is different: The high-dimensional space many afford more ways for the system to "get around" a barrier or local maximum during learning. The more superfluous the weights, the less likely it is a network will get trapped in local minima.
- However, networks with an unnecessarily large number of weights are undesirable because of the dangers of overfitting.

+ How Important are Multiple Minima

- The possibility of the presence of multiple local minima is one reason that we resort to iterative gradient descent (analytic methods are highly unlikely to find a single global minimum), especially in high-dimensional weight spaces. In computational practice, we do not want our network to be caught in a local minimum having high training error because this usually indicates that key features of the problem have not been learned by the network. In such cases it is traditional to reinitialize the weights and train again.
- In many problems, convergence to a nonglobal minimum is acceptable, if the error is nevertheless fairly low. Furthermore, common stopping criteria demand that training terminate even before the minimum if reached, and thus it is not essential that the network be converging toward the global minimum for acceptable performance.
- In short, the presence of multiple minima does not necessarily present difficulties in training nets.

+

6.5 Back propagation as feature mapping

+ The X-OR Problem

- Training neural network (without backpropagation) for X-OR problem…
  … Solution unreachable!

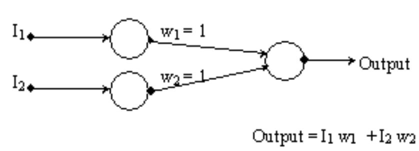


Figure from http://gseacademic.harvard.edu/

+ From Pattern Classification Point of View

- The input patterns are linearly inseparable

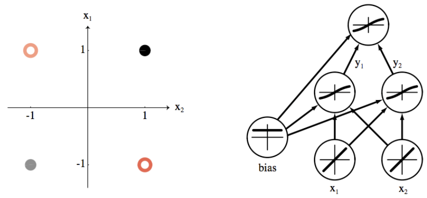


Figure from R. O. Duda, P. E. Hart, *Pattern classification*, 2001.

+ Solving the Problem with Backpropagation

- Add hidden layers with weight-adjustable nodes
- Weights are adjusted with backpaopagation of errors
- Discrete thresholding function is replaced with a continuous (sigmoid) one

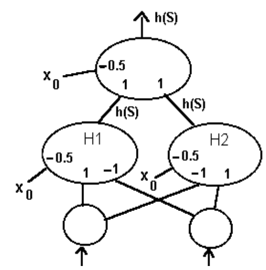


Figure from http://www.hpcc.org/

+ From Pattern Classification Point of View

- The hidden units contribute to nonlinear warping of input patterns to order to make them linearly separable

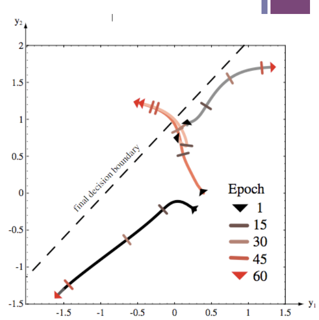
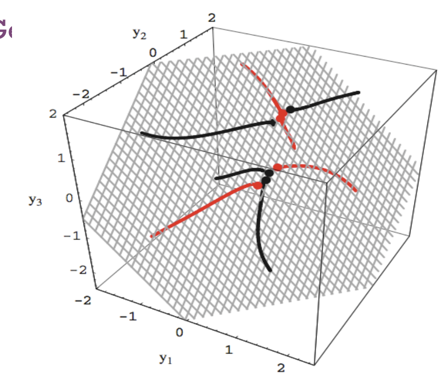


Figure from R. O. Duda, P. E. Hart, *Pattern classification*, 2001.

+ G



using neural networks", *Neural Networks*, 1999.

+ Weights in Hidden Layer

- Hidden-to-output weights leads to linear discriminant
- Input-to-hidden weights are most instructive

  - "finding features" (not exact but convenient)

+ 64-2-3 network for classifying three characters

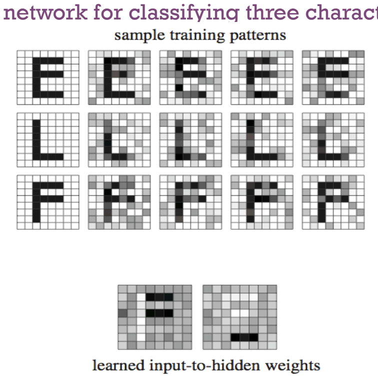
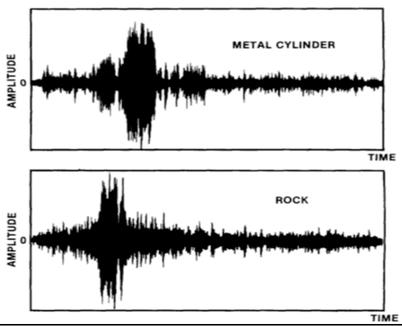


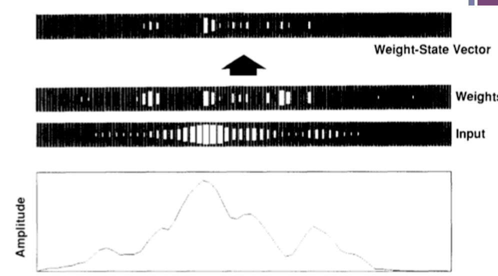
Figure from R. O. Duda, P. E. Hart, *Pattern classification*, 2001.

+ 60-3-2 Network for Classifying Sonar Signals [2]



[2] R. P. Gorman, T. J. Sejnowski, “Analysis of hidden units in a layered network trained to classify sonar targets”, *Neural Networks*, 1988.

+ Weights of One Hidden Node



+

6.6 Backpropagation, Bayes theory and probability

+ Backpropagation, Bayes theory and probability

- While multilayer neural networks may appear to be somewhat ad hoc, we now show that when trained via back propagation on a sum squared error criterion they form a least squares fit to the Bayes discriminant functions.

- In chapter 5, the LMS algorithm computed the approximation to the Bayes discriminant function for two-layer nets. We now generalize this result in two ways: to multiple categories and to nonlinear functions implemented by three layer neural networks.

+ Bayes discriminants and neural networks

- Recall first Bayes' formula:

$$P(w_k \mid x) = \frac{P(x \mid w_k)P(w_k)}{\sum_{i=1}^{c} P(x \mid w_i)P(w_i)} = \frac{P(x, w_k)}{P(x)}$$

$$g_k(x) = P(w_k \mid x) \qquad w_k$$

- Bayes decision for any pattern x: choose the category *having the largest* discriminant function:

+ Bayes discriminants and neural networks

- Suppose we train a network having *c output units with a target signal according*

$$t_k(x) = 0, x \in w_k$$
$$t_k(x) = 1, x \notin w_k$$

$$J(w) = \sum_x [g_k(x; w) - t_k]^2$$

- The contribution to the criterion function based on a single output unit *k for finite number of training* samples x is:

Bayes discriminants and neural networks

$$= \sum_{x\in w_k}[g_k(x;w)-1]^2 + \sum_{x\notin w_k}[g_k(x;w)-0]^2$$

$$= n\left\{\frac{n_k}{n}\frac{1}{n_k}\sum_{x\in w_k}[g_k(x;w)-1]^2 + \frac{n-n_k}{n}\frac{1}{n-n_k}\sum_{x\notin w_k}[g_k(x;w)-0]^2\right\}$$

- Where n *is the total number of training patterns,* n_k *of which are in* w_k

Bayes discriminants and neural networks

- In the limit of infinite data we can use Bayes' formula to express the equation above [3].

$$\int [g_k(x;w)-P(w_k \mid x)]^2 p(x)dx + \int P(w_k \mid x)P(w_{i\neq k} \mid x)p(x)dx$$

- The backpropagation rule changes weights to minimize the left hand side of the equation above.

$$\int [g_k(x;w)-P(w_k \mid x)]^2 p(x)dx$$

[3] Ruck, D.W., Rogers, S.K., Kabrisky, M., Oxley, M.E., Suter, B.W. "The multilayer perceptron as an approximation to a Bayes optimaldiscriminant function". *IEEE Transactions on Neural Networks*. Volume 1, P:296-298, 1990.

Bayes discriminants and neural networks

- For each category w_k *(k = 1, 2, …, c), backpropagation minimizes the* sum:

$$\sum_{k=1}^{c}\int [g_k(x;w)-P(w_k \mid x)]^2 p(x)dx$$

- Thus in the limit of infinite data the outputs of the trained network will approximate (in a least-squares sense) the true a posterior probabilities, that is, the output units represent the a posterior probabilities.

$$g_k(x;w) \approx P(w_k \mid x)$$

Outputs as probabilities

- In the previous subsection we saw one way to make the *c output units of a trained net* represent probabilities by training with 0–1 target values.
- While indeed given infinite amounts of training data (and assuming the net can express the discriminants, does not fall into an undesirable local minimum, etc.), then the outputs will represent probabilities.
- If these conditions do not hold — in particular we have only a *finite amount of training data — then the outputs will not represent probabilities;* for instance there is no guarantee that they will sum to 1.0. In fact, if the sum of the network outputs differs significantly from 1.0 within some range of the input space, it is an indication that the network is not accurately modeling the posteriors.

Outputs as probabilities

- Softmax method — a smoothed or softmax continuous version of a winner-take-all nonlinearity in which the maximum output is winnertake-all transformed to 1.0, and all others reduced to 0.0.

$$z_k = \frac{e^{net_k}}{\sum_{m=1}^{c} e^{net_m}}$$

- The softmax output finds theoretical justification if for each category Wk the hidden unit representations y can be assumed to come from an exponential distribution

Conclusion

- A neural network classifier trained in this manner approximates the posterior probabilities $P(w_i \mid x)$, *whether or not the data was sampled from unequal priors* $P(w_i)$. If such a trained network is to be used on problems in which the priors have been changed, it is a simple matter to rescale each network output, $g_i(x) = P(w_i \mid x)$ *by the* ratio of such priors.

+ Questions?

Thank you ☺