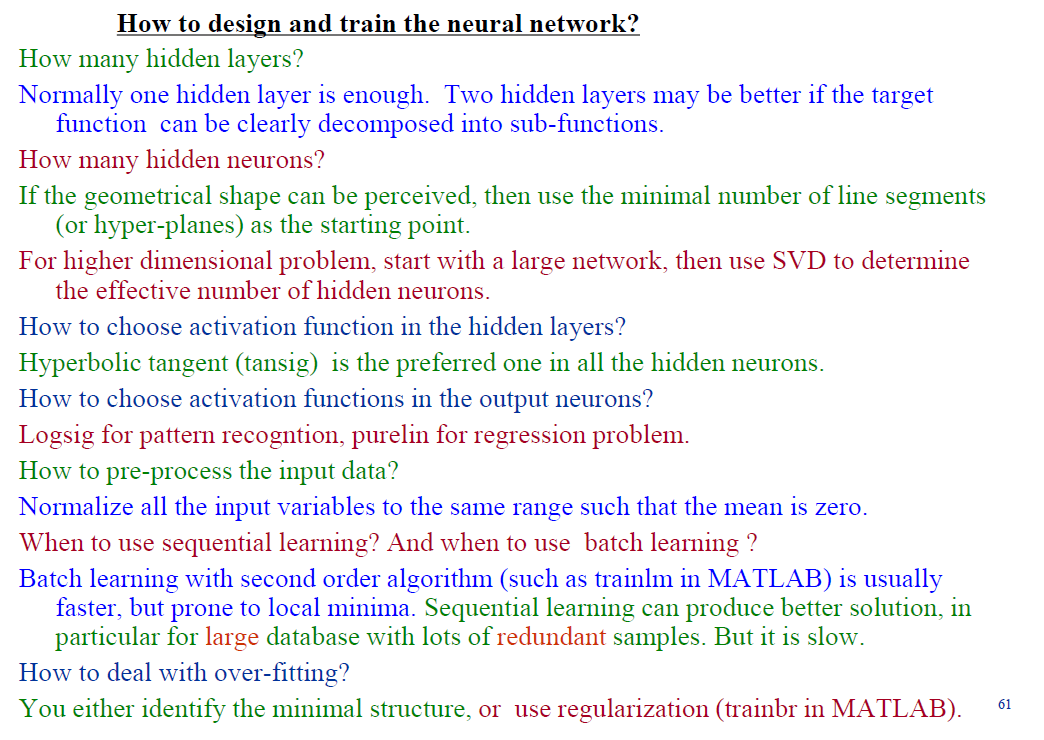


Newton’s method is one of the second order method used in batch training. It can converge to the minimum in one step if the cost function is quadratic. However, this comes at a high cost of computing the hessian matrix used in the newton’s method.



**Problems with back propagation algorithm and possible remedies.**

Problems:

1. For single layer perceptron, the least mean square algorithm does not converge. Similarly, back propagation algorithm does not converge. The learning process continues on an epoch-by-epoch basis until certain stopping criterion is met.
2. Learning in backpropagation algorithm often takes a long time to converge.
3. Gradient descent approach in back propagation algorithm only guarantees to reduce the total error to a local minimum, and not global minimum.
4. Back propagation algorithm is essentially a black box. The back propagation algorithm can provide a desired mapping between input and output vectors (x, y) but does not have the information of why a particular x is mapped to a particular y. This is because the hidden units and the learned weights do not have semantics. Thus, back propagation algorithm cannot provide an intuitive explanation for the computed results.

Remedies:

1. Declare stopping criterion. For instance,
2. When mean squared error over the entire training set is less than some threshold value
3. When the total number of epochs reaches a threshold
4. When absolute rate of change in mean squared error per epoch is sufficiently small
5. When synaptic weights and bias level stabilized
6. Use batch learning instead of sequential learning in back propagation algorithm.
7. (a) Try nets with different number of hidden layers and hidden units. They may lead to different error surfaces; some might be better than others.

(b) Try different initial weights. This will result in different starting points on the surface.

(c) Forced escape from local minima by random perturbation using simulated annealing.

(d) Use sequential learning instead of batch learning. In sequential learning, the estimate of the true gradient is noisy, the weights may not move precisely down the gradient at each iteration. This “noise” can be advantageous because it may help the network to jump out of the local minimum, and move into a deeper (therefore better) local minimum.

**Can MLP be used to solve 2 class classification problems that are not linearly separable.**

MLP can solve the nonlinearly separable problem. The nonlinearly separable problem can be transformed into linearly separable problem in the feature space produced by the hidden layer of the MLP.

**How learning rate and bias affect perceptron convergence.**

If learning rate is chosen to be very large and applied to example x(n), the learning is excellent as far as the present example is concerned, but at the cost of spoiling the learning that has taken place earlier with respect to other examples. Thus, although the perceptron converges fast, a large value of learning rate is not necessarily good as the results obtained when the perceptron converges is undesirable.

If learning rate is extremely small, this leads to slow learning. Thus, resulting in slower perceptron convergence.

Some intermediate value of learning rate is the best. Usually, the choice of learning rate is problem dependent.

The bias shifts the decision boundary away from the origin depending on the value of the bias. The use of a predicted bias that is close to the desired bias of the perceptron will result in faster convergence of the perceptron.

**Differences and similarities between (i) single layer linear network, (ii) single hidden layer and (iii) multi-layer perceptron in function approximation.**

|  |  |  |
| --- | --- | --- |
| **single layer linear network** | **single hidden layer** | **multi-layer perceptron** |
| Can only be used in linear regression problem to approximate linear function. | Can be used to approximate linear and non-linear function. | Can be used to approximate linear and non-linear function. |
| The function approximation can be carried out using offline calculation such as drawing the best fit line in the plot. | The minimum number of hidden neurons required in the hidden layer depends on the minimum number of line segments to construct the geometrical shape of the function. | More hidden layers are used in the function approximation if the function can be easily decomposed into simple sub-functions. |
| Learning method is least mean square algorithm | Learning method is back propagation algorithm | Learning method is back propagation algorithm |

**Discuss on how back-propagation (BP) learning in a multi-layer perceptron can be accelerated.**

