1.A neurons behaviour is made up of two sections:

a) a weighted sum of inputs

b) an activation function

The weighted sum of inputs is used by the activation function to produce the output of the neuron.

As the weighted sum of inputs is a linear operation, whether or not the neuron is linear or non-linear is ****determined by the activation function****.

Therefore there is no difference between a non-linear neuron and a non-linear activation function. The same is true for linear neuron and linear activation function.

1. Why do we use the bias value:

<https://stackoverflow.com/questions/2480650/role-of-bias-in-neural-networks>

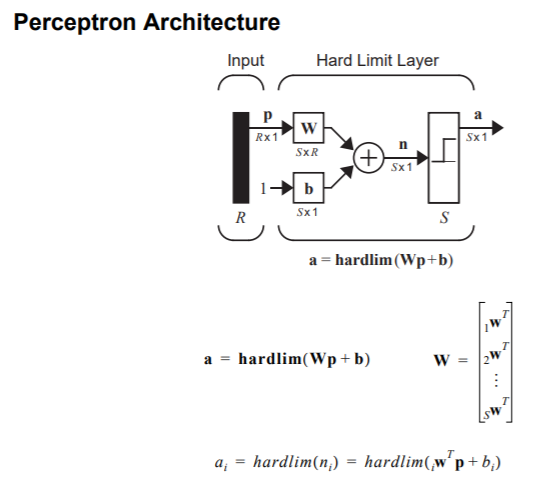
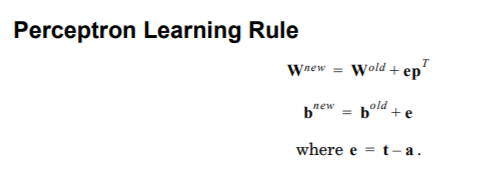
1. Yes, I can.It is very easy.
2. The activation functions as follows:

<https://en.wikipedia.org/wiki/Activation_function> The matlab name should be searched.

The sigmoid squashing function is the same as the [sigmoid function](http://en.wikipedia.org/wiki/Sigmoid_function).

The term sigmoid squashing function is favored in the neural net community. The [logistic function](http://en.wikipedia.org/wiki/Logistic_function) is the classical squashing function. Other sigmoid functions include: arctangent, the hyperbolic tangent, the Gudermannian function, and the error function.

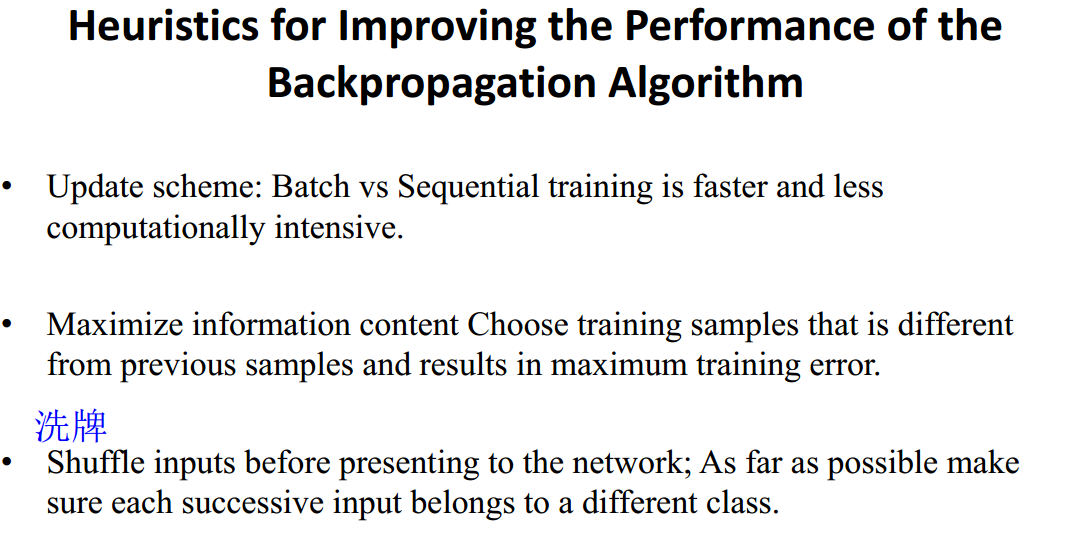
It is called the squashing function because it is tightly bounded to the pair of horizontal asymptotes. Thus, useful in compressing, or squashing, outputs.

1. Yes, I can.
2. 

7.8.

1. http://www.vias.org/encyclopedia/error\_surface.html

Find the best weight for the lowest error.

1. -16.
2. 02-13的slide.
3. a. It is not good to accelerate the convergence of the algorithms.

b. C,d:

Normalization: Let us say your inputs are 2-dimensional (x,y) where x ranges from 0-100 and y ranges from 0-10. While training the neural network with a learning rate, in each iteration the network has to make small step in y as compared to x because of different scales. The huge steps make the model to consume extra time to converge. if u normalize the data, every dimension will be in the same scale like 0-1 and therefore the size of steps is almost same which makes the model to converge faster.

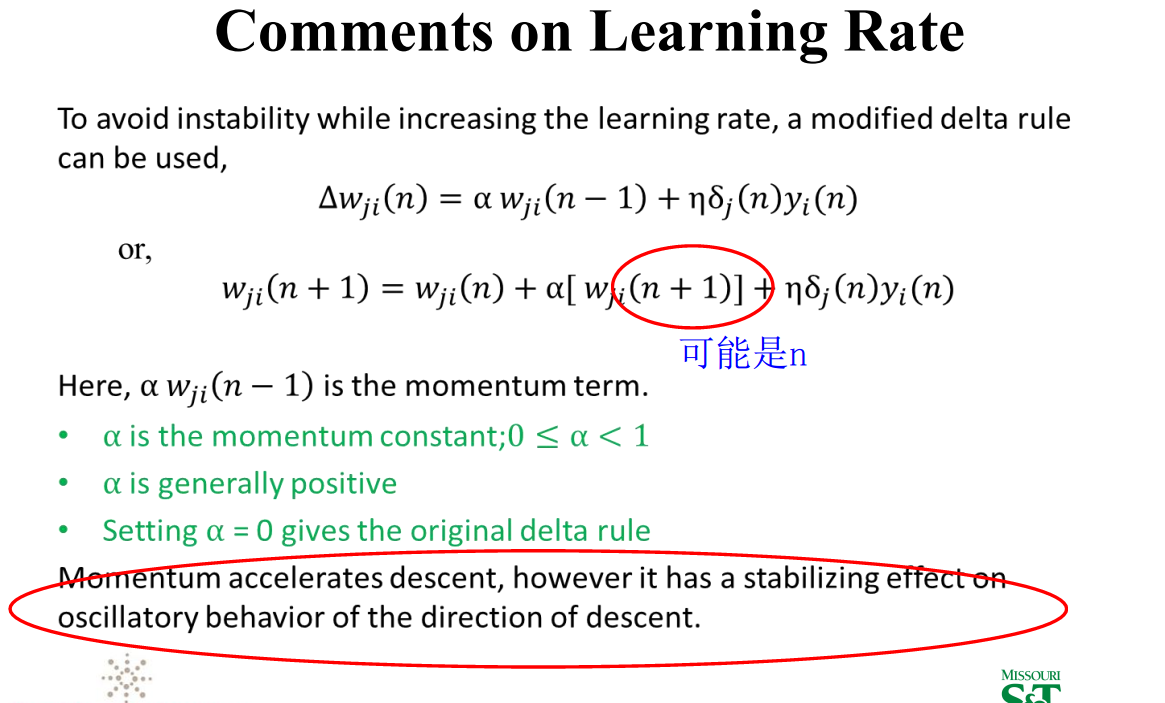
Randomize: let us say I have 100 pts for 2 classes. the first 50 belong to class 1 and last 50 to class 2. If I train my model without shuffling, and use 70 pts for train and 30 for test, then model will be partial towards class 1 as I have more samples for class 1 than class 2. So during validation (last 30 pts), even though all my pts belong to class 2, my model makes errors because it is poorly trained for identifying values of class 2. By randomizing, we maintain the balance between class 1 and class 2 so its accuracy will be better.

linear output is mainly used of regression problems (say predicting stocks). non-linear output is mainly used for classification when we have constraints on the values of output (like if output can only be between 0 and 1). Note that non-linear can also be used for regression if the data you work with requires it.

e.smaller learning rate,smoother trajectory,slower rate of convergence:

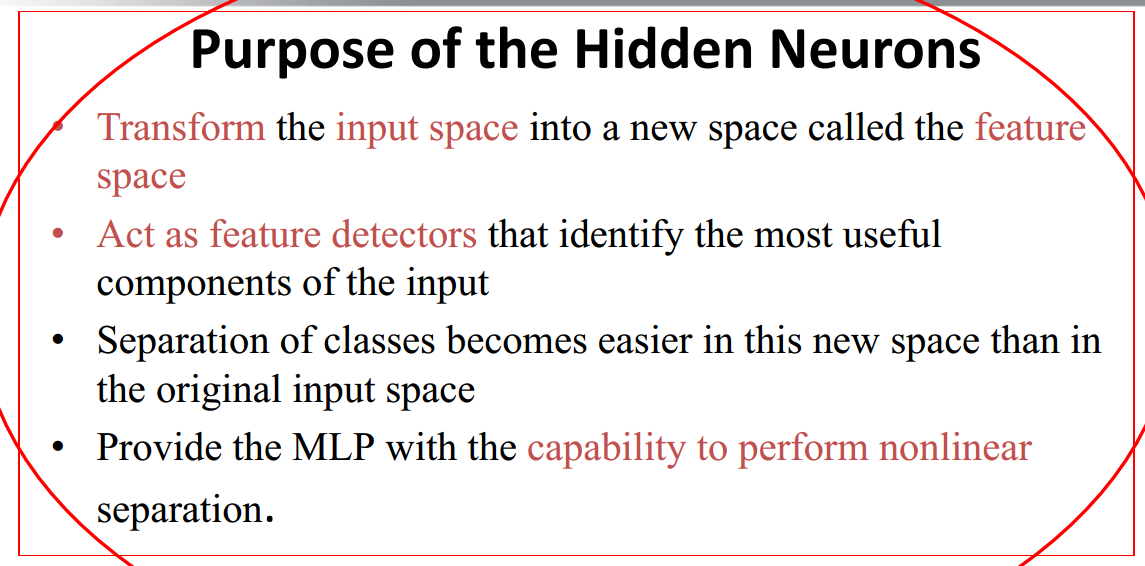
Larger learning rate,zigzagging trajectory,faster rate of learning;possible unstable behavior.

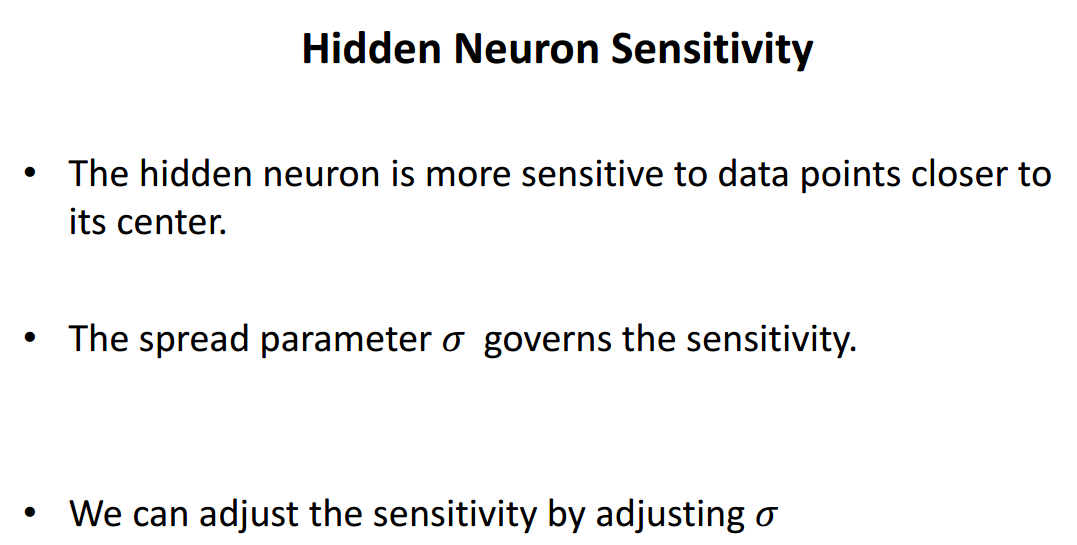
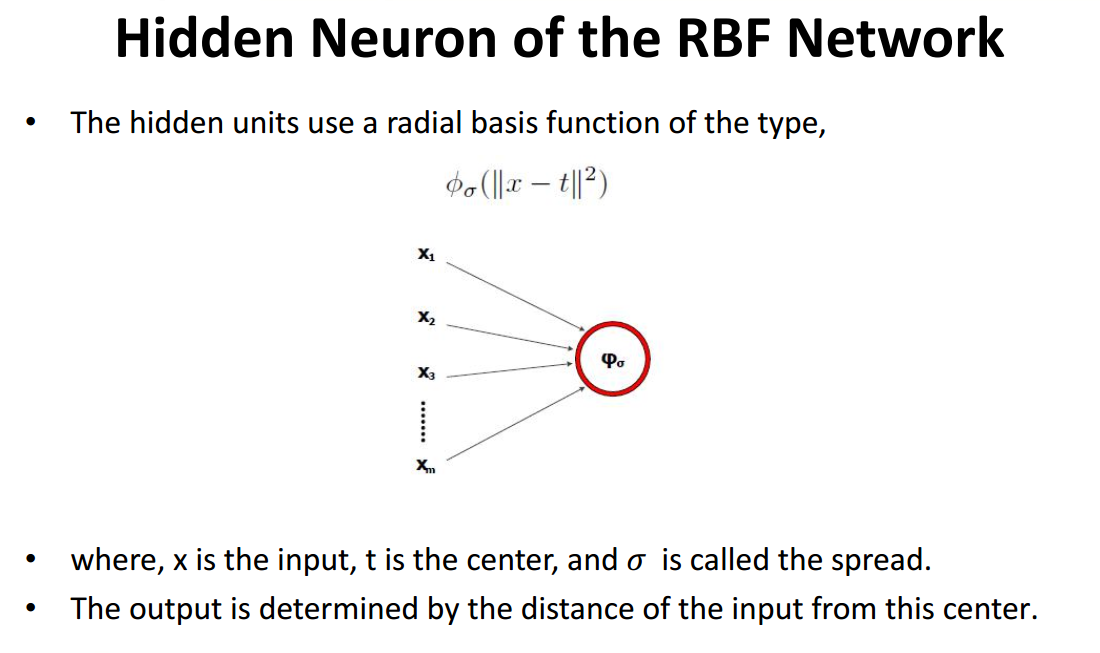
f. 02-13 slide:



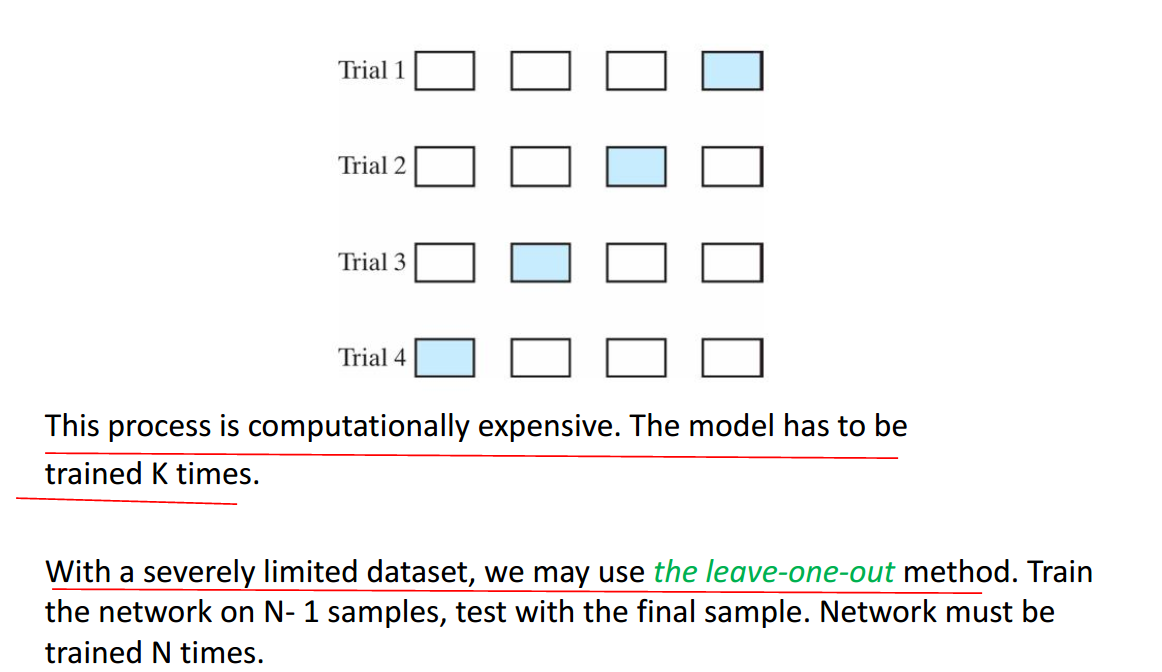
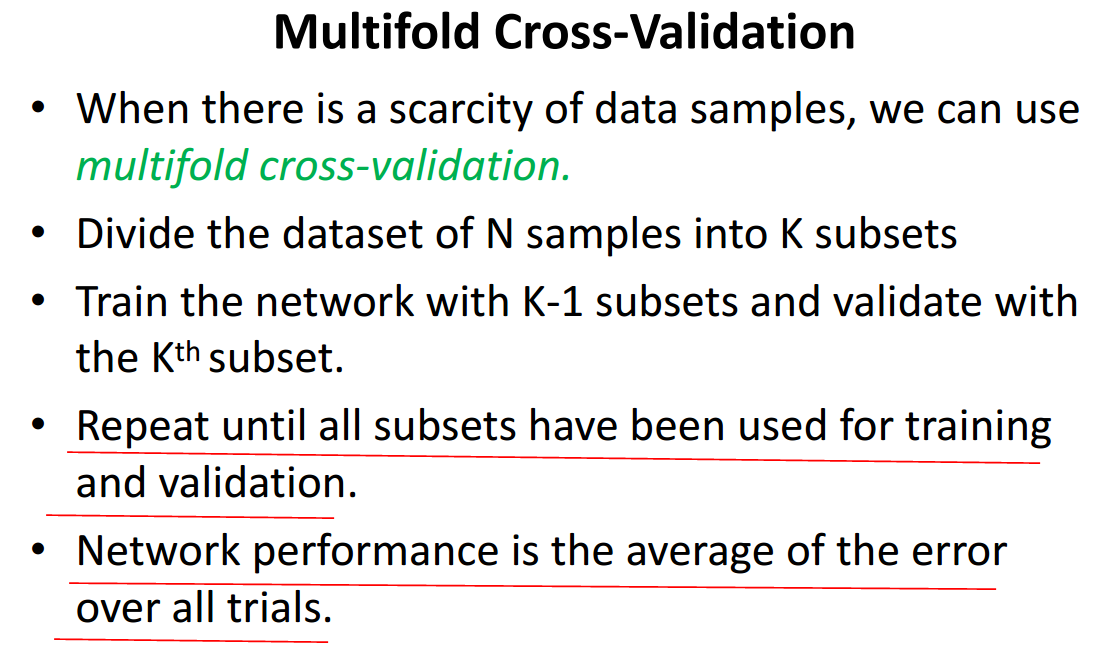
1. Not a continuous and differentiable.
2. <https://www.quora.com/What-is-the-difference-between-batch-learning-and-incremental-learning>
3. <https://elitedatascience.com/overfitting-in-machine-learning>

19-21.省略。

22.02-13slide



1. K-fold:



1. 