Week 7: Activation Functions

Thursday, March 31, 2022 4:21 PM

Practical Stream in Multilayer Networks

1. Load data
2. Define the model
3. Compile model
4. Adjust - train
5. validate
6. Predict / Use

Validation techniques

a) Hold out: separate the data before train and test.
   - 70% for training
   - 30% for test

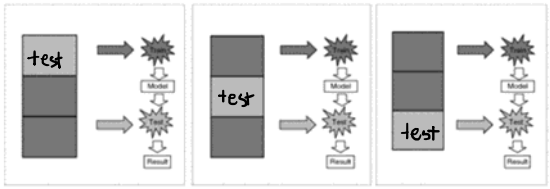
b) k-fold

c) First Round
   - 1st quartile for test
   - The rest for training

d) Second Round
   - 2nd quartile for test
   - The rest for training

e) Leave alone
   - Particular case of K-fold
   - Separate a set "K" and you will change it every epoch

K-fold idea



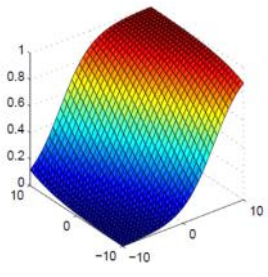
Remarks on cross validation

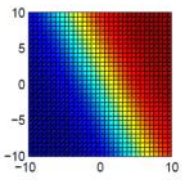
→ There are 3 things to consider
   → γ the portion of data saved for validation
   → s the complexity of function
   → m the amount of data
→ When s is small with respect to m, practically the γ value is not significant.
→ When s is large, the performance is sensitive to γ, for practice $\gamma = 0.75$
→ The best value of γ is inversely proportional to s.
→ As error decay in power law form, the bounding is difficult.

Neural Transfer Function

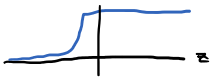
→ The decisions' contour flexibility is related to the number of neurons in the network
→ The most popular activation functions are: Sigmoid and Gaussian.
→ A neuron can be approximated to a random variable transformation

1. Logistic Function

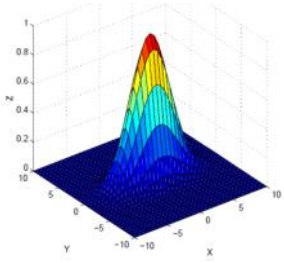
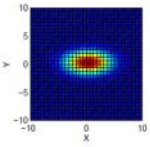




→ Parameters: input + 1
→ Asymmetric
→ Activation: internal point
→ Sigmoid is almost the same: $\frac{1}{1+e^{-z}}$



2. Gaussian Function

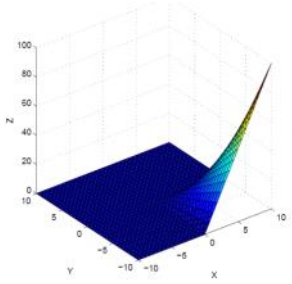
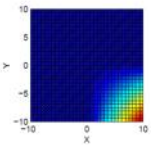



→ parameters: input +1
→ Asymmetric
→ Activation: distance } the distance to a point will define the output
(mean)

range: (0, 1]

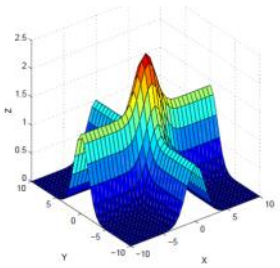
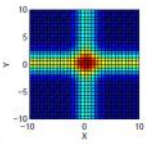
e^{-x^2}

3. Tensor product

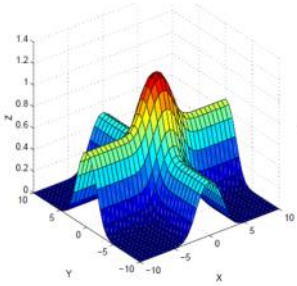
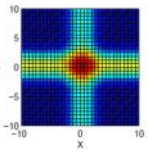
→ params: 2 · input + 1
→ Asymmetric
→ Activation: inner product (dot product?)

4. Gaussian Bar

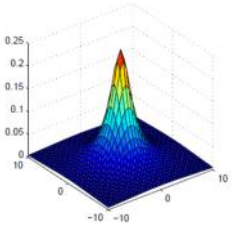
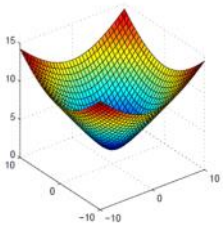
→ parameters: 3 · inputs
→ Symmetric
→ Activation: distance

5. Sigmoidal Bar

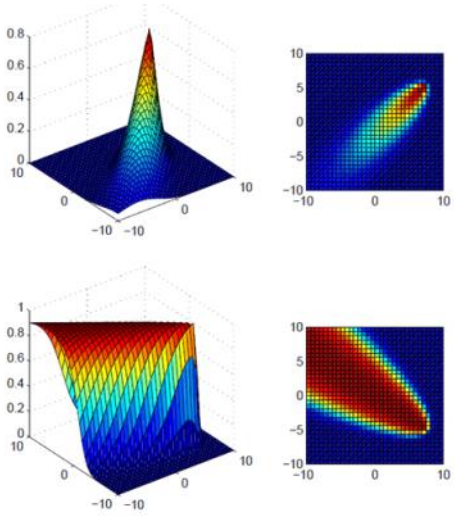
→ parameters: 3 · input
→ Symmetric
→ Activation: distance

6. Multicuadrics function

→ params: inputs +2
→ Symmetric
→ activation: distance

7. Conical Functions



→ Parameters: 2·inputs+2
→ Symmetric/Asymmetric
→ Activation: distance + product

Activation Function Remarks

→ Approaching problems from a graphic point of view can be a problem with multiple variables.
→ The more params, the more precision
→ The Activation Functions that lower the network size are:
  - circulars
  - conicals
  - bycentrals
  - gaussians
→ The one with less complexity is the Sigmoid.

Why we use activation functions in NN?

→ Used to determine the output of neural network like yes/no.
→ It maps the resulting values between 0 or 1, etc (depends on the function)