Deep Learning for the Real World

Annuli Detection in ultrasound images

Marcel Ruegenberg Anupama Jha
Sandip Mukherjee

July 24, 2013

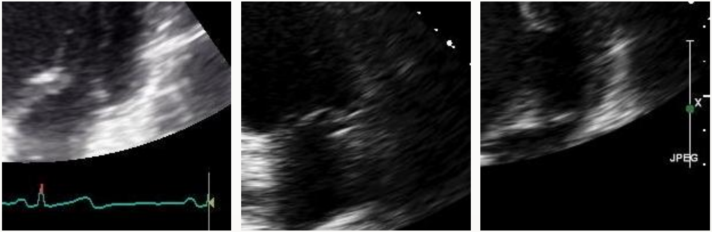
Dataset & Task

- In ultrasound of the heart, detect annuli (rings around the heart valves)
- Examples:

- Annulus roughly at the center of each sample

Preprocessing

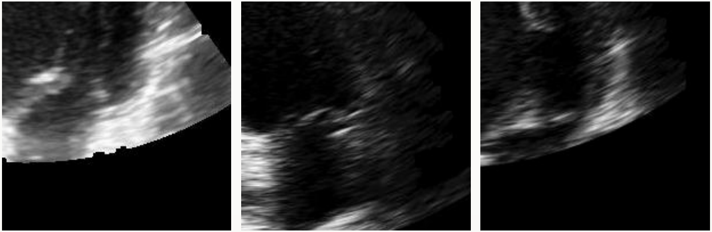
- Images contained artefacts:
  - Scales
  - Lines from ultrasound device



  - Fixed by masking colored/purely white pixels; flood fill; image morphology operations (dilation / erosion)

Preprocessing

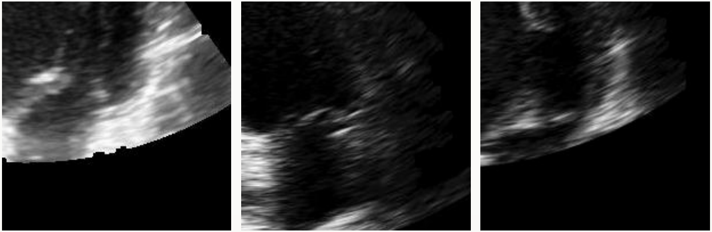
- Images contained artefacts:
  - Scales
  - Lines from ultrasound device



  - Fixed by masking colored/purely white pixels; flood fill; image morphology operations (dilation / erosion)

Preprocessing

- Images contained artefacts:
  - Scales
  - Lines from ultrasound device



  - Fixed by masking colored/purely white pixels; flood fill; image morphology operations (dilation / erosion)

- Standard data normalization: $\tilde{X} = \frac{X-\mu}{\sigma}$

Building the dataset

- Dataset only has 2 classes: *left* and *right* annulus
- Training requires 3rd class: *no annulus*

Building the dataset

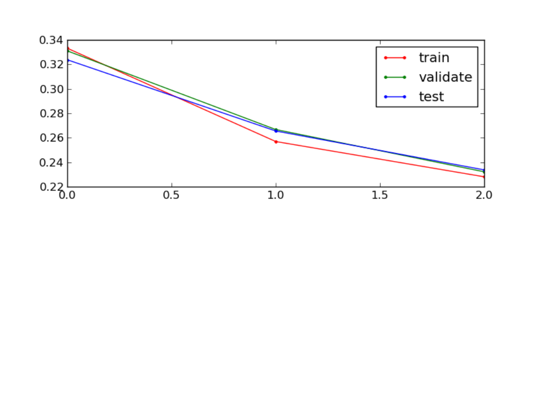
- Dataset only has 2 classes: *left* and *right* annulus
- Training requires 3rd class: *no annulus*
  - Build the dataset from smaller patches in the image (20×20px or 60×60px).
  - Take center patches of images as annulus, take surrounding patches as "no annulus".

Building the dataset

- Dataset only has 2 classes: *left* and *right* annulus
- Training requires 3rd class: *no annulus*
  - Build the dataset from smaller patches in the image (20×20px or 60×60px).
  - Take center patches of images as annulus, take surrounding patches as "no annulus".
  - Shuffle found patches, then split into training/set/validation set (60/20/20).

First approaches

- Standard Feed forward Neural Network
    - 1 Hidden layer, 0.1 learning rate and rmsprop
    - Best result obtained : 17%
- Plots for 2ch 60x60 patch with 2 class



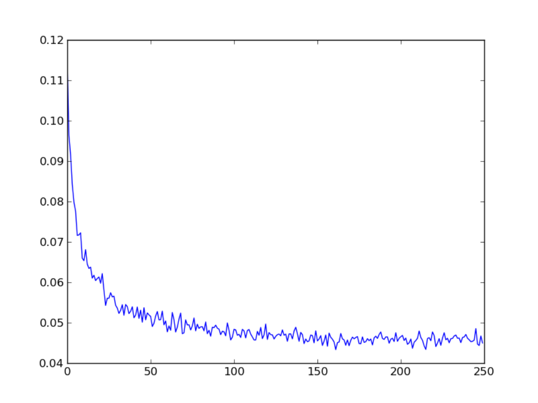
Standard NN with Dropout

- Implementation Details
  - 1 Hidden layer, with 1200 neurons size
  - 50% dropout on hidden layers,20% dropout on input layer
  - Used theano.binomial to create the mask with given probability
  - Momentum and weight updation is done according to Hinton's paper
  - decay of learning rate after each epoch
- Challenges
  - Determination of maximum squared length limit for incoming weights
  - Number and dimension of hidden layers
  - High dimension for hidden layers like 4200 didn't work so well

Result with NN Dropout

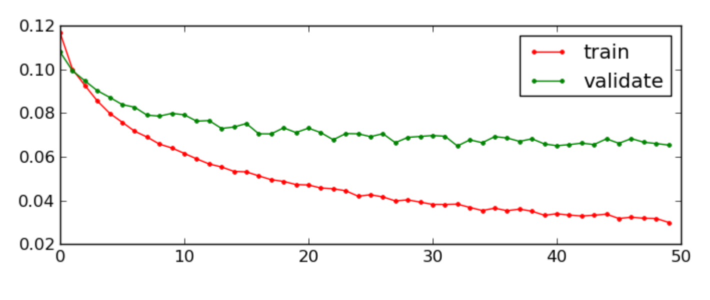
- Performance improved from standard NN
  - 4% test error for 2class unbalanced set
  - 6% error for multiclass unbalanced test set

- Plots for 2ch 60x60 patch with 2 class



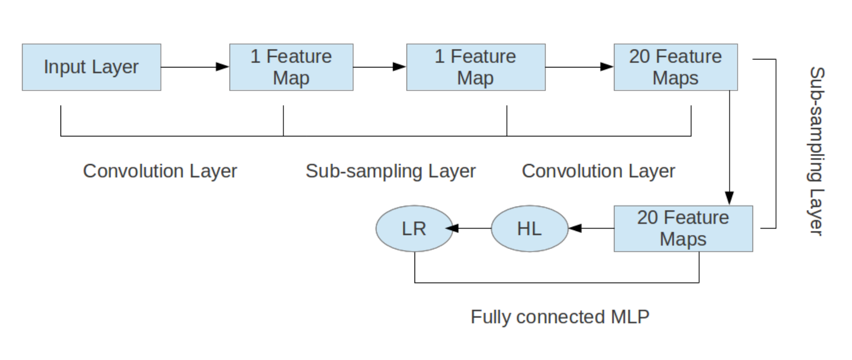
Dropout Result(Contiued)

- Plots of 2ch 60x60 patch multiclass



- Future Scope
  - Pre-trained DBN with dropout
  - Dropout on last hidden layer of Convolutional Neural Network

Convolutional Neural Network



CNN: Details

- Convolutional Layer 1
  - 20 Filters of 5 x 5
  - Maxpooling by the factor (2, 2)
- Convolutional Layer 2
  - 50 Filters of 5 x 5
  - Maxpooling by the factor (2, 2)
- Hidden Layer
  - Input: (number of Filters of CL2) * (CL2 output size)^2
  - Output: 500

CNN: Details

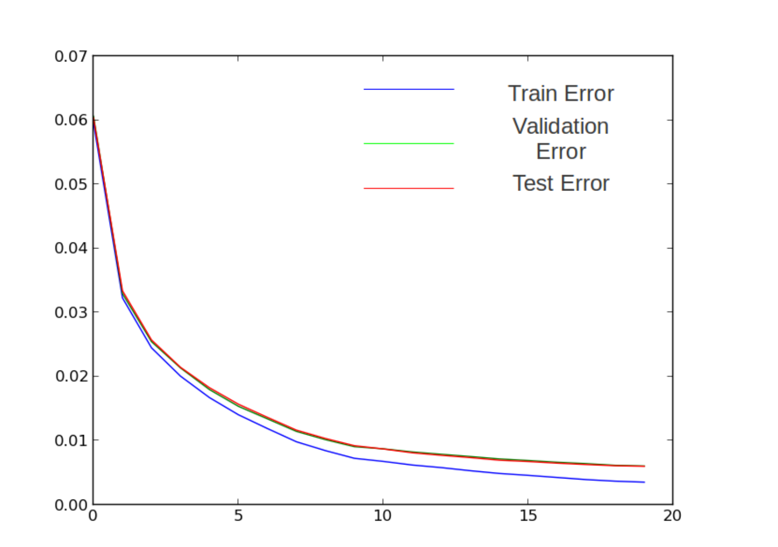
- Logistic Regression Layer
  - Input: 500
  - Output:
    2 (Annuli or not) or 3 (left, right or no Annuli) classes
- Hyperparameters: Minibatch Gradient Descent
  - Learning Rate: 0.1
  - Epochs: 20

Additional tricks

- Convolutional Neural Network with rmsprop and Momentum
  Hyperparameters: RMSPROP
  - Learning Rate: 0.0001
  - Momentum: 0.7
- Neural Network with dropout

Results: CNN-MSGD

2 chamber Images, 20 x 20 patches

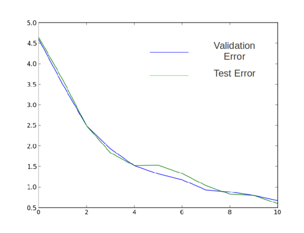


Results: CNN-MSGD

Best validation Error: 0.606816 %
Best Test Error: 0.602769 %
Train, validation and test data was balanced, but results were not much different then unbalanced test set
True Positive(Annuli) 49.9218 %
True Negative(Non Annuli) 49.4754 %
False Positive 0.5798 %
False Negative 0.0230 %

Result and Classifier

- Plots of 2ch 60x60 patch with CNN rmsprop
  - Validation Error: 0.676190 %
  - Test Error: 0.600000 %



- Classifier
  - It takes directory of images
  - Do all the preprocessing and create patches from images
  - Use previosuly saved weights
  - And finally predict whether the image has annulli or not

Questions?

Thank you