Practical Methodology

[almost] Dr. Kaitlin Palmer
San Diego State University

J&M: 9.4.1

Practical Methodology

• Project Design
• Performance Metrics
• Pipeline
  – Baseline Algorithm Selection
  – Optimization
• Data collection
• Hyperparameter Tunining
• Debugging
Performance Metrics

• Consider- training data will never be perfect
• How good does the system need to be?
  – Accurate, precise?
  – Better than competitors?

Error Types

• Accuracy
  – Rare events, set model equal to 0
• Precision/Recall
  – Fraction of correctly predicted observations
  – Fraction of true events that were detected
• Type I and II error

<table>
<thead>
<tr>
<th></th>
<th>( H_0 ) True</th>
<th>( H_0 ) False</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reject ( H_0 )</td>
<td>Type I Error</td>
<td>Correct Rejection</td>
</tr>
<tr>
<td>Fail to Reject ( H_0 )</td>
<td>Correct Decision</td>
<td>Type II Error</td>
</tr>
</tbody>
</table>
Performance Metrics

- Precision/Recall
- F-scores (manager’s like single numbers)

\[ F = \frac{2pr}{p + r} \]

- Maybe we shouldn’t leave everything to the machines…
- AI only useful if they can reduce the payload for human analysts
- Coverage
  - Proportion of which the machine is able to produce a response
  - Perfect accuracy - refuse to do anything
Other Performance Metrics

- Categorizing emotion?
  - User satisfaction
- Adverting
  - Sales outcomes
  - Click-through rates
- Numerous project specific metrics
  - Your turn!

Next Steps

- We have a performance metric
- Establish “end-to-end” system (ASAP)
  - Prototype
  - Check predictions, debug
  - Add complexity
  - Check predictions, debug
Pipeline

- Data In
- AI Algorithm Optimization
- Measure Error
- Tweak
  - Collect Data
  - Modify Model
  - Adjust Hyperparameters

Pipeline

- Data In
- AI Algorithm Optimization
- Measure Error
- Tweak
  - Collect Data
  - Modify Model
  - Adjust Hyperparameters
Deep Learning Algorithms

• Deep learning first? Why? Why not?
• Statistics models (there are lots!)
  – GMM, PCA
  – Random forest
  – CART
  – GAMs, GLM, linear regression
• Read with a skeptical eye

Deep Learning Algorithms

• No Data Structure
  – Lots of measurements, sequence of features
  – Feed forward network (2-3 hidden layer, RELU)
Deep Learning Algorithms

• Structured Data (images)
  – Convolutional network
    • ReLU
    • Leaky ReLU
    • PreLU
    • Maxout

• Sequences (in/out)
  – Gated recurrent net
    • LTSM
      – Train with momentum or SGD
      – Clip the gradients to avoid instability in back propagation
    • Set forget gate high (remember everything)
    • GRU
Deep Learning Algorithms

• Is your problem well understood?
  – Copy a trained model (architecture)
  – Convolutional networks trained on ImageNet

Optimization Algorithm

• Stochastics Gradient Descent
  – Decaying learning rate
    • Linear
    • Exponential
    • Minimum learning rate
  – Adam (adaptive moments)
  – Introduce batch normalization quickly if optimization is problematic
Additional Setup Considerations

- How much training data do you really have?
  - If less than millions see below
- Probably need to use regularization
  - Early stopping (almost always!)
  - Dropout
  - Batch normalization (reduces generalization error), skip dropout

Round 1

- Data In
- AI Algorithm Optimization
- Measure Error
- Tweak
  - Collect Data
  - Modify Model
  - Adjust Hyperparameters

17

18
Measure Error

• How well did the model perform wrt desired metric
  – On the training data (poor)
  – On the test data (poor)
• Poor performance on training set will not benefit from more data
  – Add more layers
  – Add more hidden units to each layer

When to Collect More Data

• Poor performance on training set
  – Try adjusting the learning algorithm
  – Learning rate (hyperparameter)
• Still no good? Humm..
  – Might be the data
Data Problems

• Are the features rich enough for the intended outputs?
• Are the data too noisy?
• Are the data correctly labeled?
  – Unsupervised learner error
  – Human error

When To Collect More Data

• How well did the model perform?
  – On the training data (good)
  – On the validation data (poor)
• Poor on the test set
  – Features in test set non-representative or incomplete
  – Collect more data!
Collecting More Data

- Dataset Augmentation
- How Much?
  - How much is available?
  - No more available
    - shrink the model
- Know your system
  - Plot generalization error vs training samples
  - Usually will need orders of magnitude (logarithmic scale)

Goodfellow et al. 2016
Getting the Most Out of Your Data and Algorithm

• Understand your data
• Understand your algorithms (especially limitations)
• Monitor output and adjust experimental system

Round 1

Data In

AI Algorithm Optimization

Measure Error

Tweak
• Collect Data
• Modify Model
• Adjust Hyperparameters

Debugging
Hyperparameter Tuning

- Tradeoff between run-time and memory
- Model quality
- Can be chosen manually (extensive knowledge or saint-like patience) or programmatically (computationally costly)
- Lowest generalization error for a given memory/time allowance

Manual Tuning

- Need to match the capacity of the model with the complexity of the task
- Model capacity determined by:
  - Representational capacity of the model
  - Ability of the learning algorithm(s) to minimize cost function used to train the model
  - Degree of regularization produced by the cost and training procedure
Model Capacity

• Overfitting
  – Value of hyperparameters too large
    • Too many hidden units per layer
  – Value of hyperparameters too small
    • Weight decay coefficient near 0 greatest effective capacity of the algorithm
  – Some hyperparameters discrete
    • Units in a layer, switches
Learning Rate

- Principle hyperparameter to tune
  - Model capacity highest when LR is correct for the problem
- Too High
  - Gradient decent increases rather than decrease the training error
  - Too small and may get stuck (also super slow)

Goodfellow et al. 2016

Automatic Hyperparameter Optimization

- Preferred when
  - No prior knowledge of the filed
  - Large (>40) number of parameters available
- Develop hyperparameter optimization algorithms
  - Optimize our objective function (training error)
  - Wrap around learning algorithm
Hyperparameter Optimization

• Grid Search
  – Three or fewer hyperparameters
    • Computational cost grows exponentially
  – “Bin” hyperparameters
  – Parameter selection on log scale
  – Monitor/update selected hyperparameters (e.g. edge of value range, scale)

Goodfellow et al. 2016
Hyperparameter Optimization

• Random Search
  – Less computationally expensive (independent exploration of the hyperparameters)
  – Define marginal distribution for each hyperparameter (Bern, binary, uniform, etc.)
  – Sample log number of hidden units form the distribution
  – Monitor/update selected hyperparameters (e.g. edge of value range, scale)
Round 1

Data In

AI Algorithm Optimization

Measure Error

Tweak
- Collect Data
- Modify Model
- Adjust Hyperparameters

Debugging

Debugging
Debugging

• Known unknowns
  – Programmatic errors

• Unknown unknowns
  – If we didn’t know what to expect, how can we tell if the result is good? Is 10% test error good?
  – Adaptive (that’s why we use deep learning)

Debugging

• Visualize / check output
  – Superimpose images for classification
  – Listen to generative

• Visualize mistakes-
  – Identify training errors in the data consistent across all datasets
Debugging

• High training error
  – Underfitting or training error
  – Small samples can be fit by small models
  – If not….

• Compare backpropagation to numerical derivatives
  – Some systems require building gradient descent
  – Implementation errors common
  – Use finite differences (calculus)
Debugging

- Visualize!
  - Monitor histograms of activations and gradients
  - Are units saturated
  - How often are rectifiers off?
  - Magnitude of parameter gradients to the magnitude of parameters themselves: have some feature parameters stalled?

Summary

- Know your question
  - Key error metrics
  - Coverage
- Know your data
  - Where might there be errors or inconsistencies?
- Know your algorithm
- Visualize, VISUALIZE, VISUALIZE