* General ML strategy
* Error Strategy
* Training setup from literature
  + Bert
  + Transformer
  + GPT
  + Rec sys
* References
  + [Advice for applying Machine Learning](https://cs229.stanford.edu/materials/ML-advice.pdf) [old ML]
  + <https://cs230.stanford.edu/files/C3M2.pdf> [DL}
  + [Andrew-NG-Notes/andrewng-p-3-structuring-ml-projects.md at master](https://github.com/ashishpatel26/Andrew-NG-Notes/blob/master/andrewng-p-3-structuring-ml-projects.md#ml-strategy-1)
  + [Troubleshooting Deep Neural Networks](http://josh-tobin.com/assets/pdf/troubleshooting-deep-neural-networks-01-19.pdf)

Diagnostics 101

1. Suppose you suspect the problem is either: –
   1. Overfitting (high variance)
      1. Variance: Training error will be much lower than test error.
   2. Too few features to classify spam (high bias).
      1. Bias: Training error will also be high.
   3. Fixes to try:
      1. – Try getting more training examples. ***Fixes high variance.***
      2. – Try a smaller set of features. ***Fixes high variance.***
      3. – Try a larger set of features. ***Fixes high bias.***
      4. – Try email header features. ***Fixes high bias.***
      5. – Run gradient descent for more iterations. Fixes optimization algorithm.
      6. – Try Newton’s method. Fixes optimization algorithm.
      7. – Use a different value for λ. Fixes optimization objective.
      8. – Try using an SVM. Fixes optimization objective.

Diagnostics 101 – DL

1. Incorrect label
2. Addressing data mismatch
   1. Carry out manual error analysis to try to understand difference between training and dev/test sets
   2. Make training data more similar; or collect more data similar to dev/test sets
3. [Structuring Machine Learning Projects](https://github.com/ashishpatel26/Andrew-NG-Notes/blob/master/andrewng-p-3-structuring-ml-projects.md#structuring-machine-learning-projects)
   1. [Table of contents](https://github.com/ashishpatel26/Andrew-NG-Notes/blob/master/andrewng-p-3-structuring-ml-projects.md#table-of-contents)
   2. [Course summary](https://github.com/ashishpatel26/Andrew-NG-Notes/blob/master/andrewng-p-3-structuring-ml-projects.md#course-summary)
   3. [ML Strategy 1](https://github.com/ashishpatel26/Andrew-NG-Notes/blob/master/andrewng-p-3-structuring-ml-projects.md#ml-strategy-1)
      1. [Why ML Strategy](https://github.com/ashishpatel26/Andrew-NG-Notes/blob/master/andrewng-p-3-structuring-ml-projects.md#why-ml-strategy)
      2. [Orthogonalization](https://github.com/ashishpatel26/Andrew-NG-Notes/blob/master/andrewng-p-3-structuring-ml-projects.md#orthogonalization)
      3. [Single number evaluation metric](https://github.com/ashishpatel26/Andrew-NG-Notes/blob/master/andrewng-p-3-structuring-ml-projects.md#single-number-evaluation-metric)
      4. [Satisfying and Optimizing metric](https://github.com/ashishpatel26/Andrew-NG-Notes/blob/master/andrewng-p-3-structuring-ml-projects.md#satisfying-and-optimizing-metric)
      5. [Train/dev/test distributions](https://github.com/ashishpatel26/Andrew-NG-Notes/blob/master/andrewng-p-3-structuring-ml-projects.md#traindevtest-distributions)
      6. [Size of the dev and test sets](https://github.com/ashishpatel26/Andrew-NG-Notes/blob/master/andrewng-p-3-structuring-ml-projects.md#size-of-the-dev-and-test-sets)
      7. [When to change dev/test sets and metrics](https://github.com/ashishpatel26/Andrew-NG-Notes/blob/master/andrewng-p-3-structuring-ml-projects.md#when-to-change-devtest-sets-and-metrics)
      8. [Why human-level performance?](https://github.com/ashishpatel26/Andrew-NG-Notes/blob/master/andrewng-p-3-structuring-ml-projects.md#why-human-level-performance)
      9. [Avoidable bias](https://github.com/ashishpatel26/Andrew-NG-Notes/blob/master/andrewng-p-3-structuring-ml-projects.md#avoidable-bias)
      10. [Understanding human-level performance](https://github.com/ashishpatel26/Andrew-NG-Notes/blob/master/andrewng-p-3-structuring-ml-projects.md#understanding-human-level-performance)
      11. [Surpassing human-level performance](https://github.com/ashishpatel26/Andrew-NG-Notes/blob/master/andrewng-p-3-structuring-ml-projects.md#surpassing-human-level-performance)
      12. [Improving your model performance](https://github.com/ashishpatel26/Andrew-NG-Notes/blob/master/andrewng-p-3-structuring-ml-projects.md#improving-your-model-performance)
   4. [ML Strategy 2](https://github.com/ashishpatel26/Andrew-NG-Notes/blob/master/andrewng-p-3-structuring-ml-projects.md#ml-strategy-2)
      1. [Carrying out error analysis](https://github.com/ashishpatel26/Andrew-NG-Notes/blob/master/andrewng-p-3-structuring-ml-projects.md#carrying-out-error-analysis)
      2. [Cleaning up incorrectly labeled data](https://github.com/ashishpatel26/Andrew-NG-Notes/blob/master/andrewng-p-3-structuring-ml-projects.md#cleaning-up-incorrectly-labeled-data)
      3. [Build your first system quickly, then iterate](https://github.com/ashishpatel26/Andrew-NG-Notes/blob/master/andrewng-p-3-structuring-ml-projects.md#build-your-first-system-quickly-then-iterate)
      4. [Training and testing on different distributions](https://github.com/ashishpatel26/Andrew-NG-Notes/blob/master/andrewng-p-3-structuring-ml-projects.md#training-and-testing-on-different-distributions)
      5. [Bias and Variance with mismatched data distributions](https://github.com/ashishpatel26/Andrew-NG-Notes/blob/master/andrewng-p-3-structuring-ml-projects.md#bias-and-variance-with-mismatched-data-distributions)
      6. [Addressing data mismatch](https://github.com/ashishpatel26/Andrew-NG-Notes/blob/master/andrewng-p-3-structuring-ml-projects.md#addressing-data-mismatch)
      7. [Transfer learning](https://github.com/ashishpatel26/Andrew-NG-Notes/blob/master/andrewng-p-3-structuring-ml-projects.md#transfer-learning)
      8. [Multi-task learning](https://github.com/ashishpatel26/Andrew-NG-Notes/blob/master/andrewng-p-3-structuring-ml-projects.md#multi-task-learning)
      9. [What is end-to-end deep learning?](https://github.com/ashishpatel26/Andrew-NG-Notes/blob/master/andrewng-p-3-structuring-ml-projects.md#what-is-end-to-end-deep-learning)
      10. [Whether to use end-to-end deep learning](https://github.com/ashishpatel26/Andrew-NG-Notes/blob/master/andrewng-p-3-structuring-ml-projects.md#whether-to-use-end-to-end-deep-learning)

Orthogonalization

1. Some deep learning developers know exactly what hyperparameter to tune in order to try to achieve one effect. This is a process we call orthogonalization. In orthogonalization, you have some controls, but each control does a specific task and doesn't affect other controls.
2. For a supervised learning system to do well, you usually need to tune the knobs of your system to make sure that four things hold true - chain of assumptions in machine learning
3. You'll have to fit training set well on cost function (near human level performance if possible). : If it's not achieved you could try bigger network, another optimization algorithm (like Adam)...
4. Fit dev set well on cost function. : If its not achieved you could try regularization, bigger training set…
5. Fit test set well on cost function. : If its not achieved you could try bigger dev. Set…
6. Performs well in real world. : If its not achieved you could try change dev. set, change cost function...

Evaluation

Single number, say F1 over precision/recall

Train/dev/test distributions

Size of the dev and test sets

An old way of splitting the data was 70% training, 30% test or 60% training, 20% dev, 20% test.

The old way was valid for a number of examples ~ <100000

In the modern deep learning if you have a million or more examples a reasonable split would be 98% training, 1% dev, 1% test.