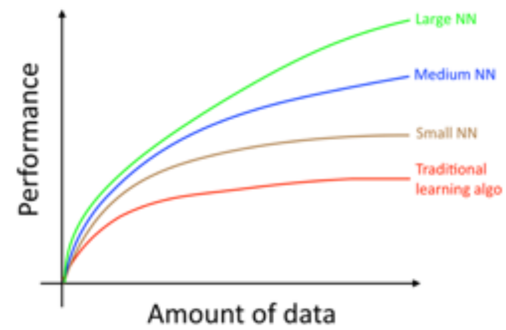
Nuts and bolts of building AI applications using Deep Learning

Andrew Ng

Trend #1: Scale driving Deep Learning progress



Trend #2: The rise of end-to-end learning

Learning with integer or real-valued outputs:

Problem	X	Y
Spam classification	Email	Spam/Not spam (0/1)
Image recognition	Image	Integer label
Housing price prediction	Features of house	Price in dollars
Product recommendation	Product & user features	Chance of purchase

Learning with complex (e.g., string valued) outputs:

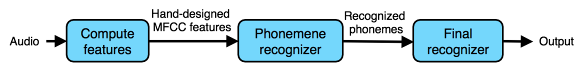
Problem	X	Y	Example
Image captioning	Image	Text	Mao et al., 2014
Machine translation	English text	French text	Suskever et al., 2014
Question answering	(Text,Question) pair	Answer text	Bordes et al., 2015
Speech recognition	Audio	Transcription	Hannun et al., 2015
TTS	Text features	Audio	van der Oord et al., 2016

Major categories of DL models

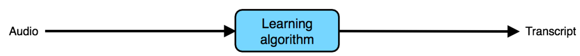
1. General neural networks
2. Sequence models (1D sequences)
   - RNN, GRU, LSTM, CTC, attention models, ….
3. Image models
   - 2D and 3D convolutional networks
4. Advanced/future tech:
   - Unsupervised learning (sparse coding, ICA, SFA, …), Reinforcement learning, ….

End-to-end learning: Speech recognition

Traditional model



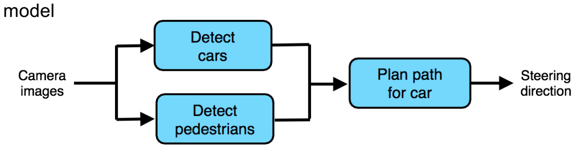
End-to-end learning



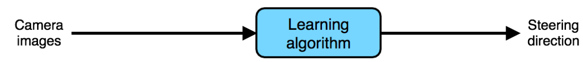
This works well given enough labeled (audio, transcript) data.

End-to-end learning: Autonomous driving

Traditional model



End-to-end learning



Given the safety-critical requirement of autonomous driving and thus the need for extremely high levels of accuracy, a pure end-to-end approach is still challenging to get to work. End-to-end works only when you have enough (x,y) data to learn function of needed level of complexity.

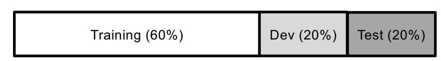
Machine Learning Strategy

Often you will have a lot of ideas for how to improve an AI system, what do you do?

Good strategy will help avoid months of wasted effort.

Traditional train/dev/test and bias/variance

Say you want to build a human-level speech recognition system. You split your data into train/dev/test:



Human level error 1%

"Avoidable bias"

Training set error 5%

"Variance"

Dev set error 8%

Compared to earlier eras, we still talk about bias and variance, but somewhat less about the "tradeoff" between them.

Basic recipe for machine learning

Training error high? — Yes → Bigger model / Train longer / New model architecture (Bias)

↓ No

Dev error high? — Yes → More data / Regularization / New model architecture (Variance)

↓ No

Done!

Automatic data synthesis examples

- OCR
  - Text against random backgrounds
- Speech recognition
  - Synthesize clean audio against different background noise
- NLP: Grammar correction
  - Synthesize random grammatical errors

Sometimes synthesized data that appears great to human eyes is actually very impoverished in the eyes of ML algorithms, and covers only a minuscule fraction of the actual distribution of data. E.g., images of cars extracted from video games.

Different training and test set distributions

Say you want to build a speech recognition system for a new in-car rearview mirror product. You have 50,000 hours of general speech data, and 10 hours of in-car data. How do you split your data? This is a **bad** way to do it:

Training | Dev — General speech data (50,000 hours)

Test — In-car data (10 hours)

Having mismatched dev and test distributions is not a good idea. Your team may spend months optimizing for dev set performance only to find it doesn't work well on the test set.

Different training and test set distributions

Better way: Make the dev and test sets come from the same distribution.

Training (~50,000h) | Training-Dev (20h) — General speech data

Dev (5h) | Test (5h) — In-car data

Human level error ……….. 1%

"Avoidable bias"

Training error ………………. 1.1%

Overfitting of training set

Training-Dev error ……….. 1.5%

Data mismatch

Dev set error ………………. 8%

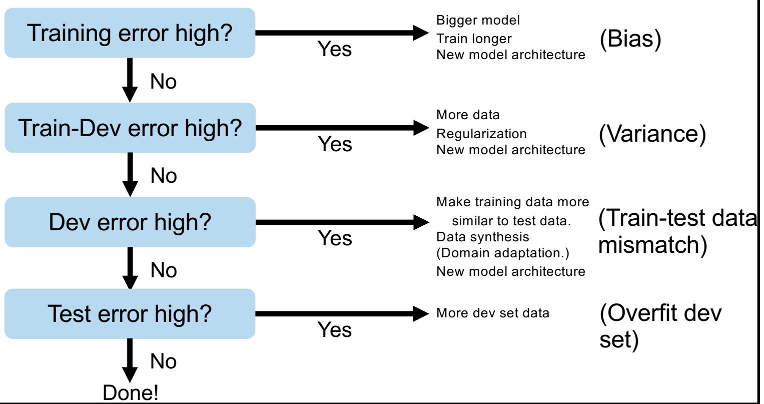
Overfitting of dev set

Test set error ………………. 8.5%

Baidu Research

Andrew Ng

New recipe for machine learning



General Human/Bias/Variance analysis

	General speech data (50,000 hours)	**In-car speech data (10 hours)**
Performance of humans	Human-level error	(Carry out human evaluation to measure.)
Performance on examples you've trained on	Training error	(Insert some in-car data into training set to measure.)
Performance on examples you haven't trained on	Training-Dev error	Dev/Test error

"Avoidable bias"

"Variance"/degree of overfitting

Data mismatch

Human level performance

You'll often see the fastest performance improvements on a task while the ML is performing worse than humans.

- Human-level performance is a proxy for Bayes optimal error, which we can never surpass.
- Can rely on human intuition: (i) Have humans provide labeled data. (ii) Error analysis to understand how humans got examples right. (iii) Estimate bias/variance. E.g., On an image recognition task, training error = 8%, dev error = 10%. What do you do? Two cases:

Human level error 1%
"Avoidable bias"
Training set error 8%
"Variance"
Dev set error 10%

Focus on bias.

Human level error 7.5%
"Avoidable bias"
Training set error 8%
"Variance"
Dev set error 10%

Focus on variance.

Quiz: Medical imaging

Suppose that on an image labeling task:

Typical human ……………….…. 3% error
Typical doctor …………………... 1% error
Experienced doctor ……………. 0.7% error
Team of experienced doctors …. 0.5% error

What is "human-level error"?

Answer: For purpose of driving ML progress, 0.5% is best answer since it's closest to Bayes error.

AI Product Management

The availability of new supervised DL algorithms means we're rethinking the workflow of how to have teams collaborate to build applications using DL. A Product Manager (PM) can help an AI team prioritize the most fruitful ML tasks. E.g., should you improve speech performance with car noise, café noise, for low-bandwidth audio, for accented speech, or improve latency, reduce binary size, or something else?

What can AI do today? Some heuristics for PMs:
- If a typical person can do a mental task with less than one second of thought, we can probably automate it using AI either now or in the near future.
- For any concrete, repeated event that we observe (e.g., whether user clicks on ad; how long it takes to deliver a package; ….), we can reasonably try to predict the outcome of the next event (whether user clicks on next ad).

AI Product Management

How should PMs and AI teams work together? Here's one default split of responsibilities:

Product Manager (PM) responsibility

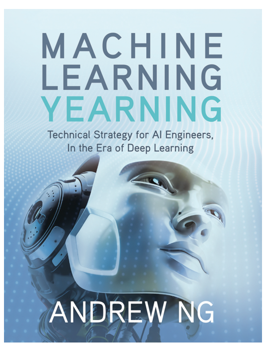
- Provide dev/test sets, ideally drawn from same distribution.
- Provide evaluation metric for learning algorithm (accuracy, F1, etc.)

This is a way for the PM to express what ML task they think will make the biggest difference to users.

AI Scientist/Engineer responsibility

- Acquire training data
- Develop system that does well according to the provided metric on the dev/test data.

Machine Learning Yearning



Book on AI/ML technical strategy.

Sign up at http://mlyearning.org

Thank you for coming to this tutorial!