Deep Learning for Natural Language Processing

Practicals Overview

Chris Dyer

Chris Dyer

DeepMind Carnegie Mellon

Where do I come from?

PhD Linguistics (U of Maryland, USA, 2010)
Postdoc CS (Carnegie Mellon, 2010–2012)
Faculty CS (Carnegie Mellon, 2012–)
DeepMind (2016–)

Chris Dyer

DeepMind Carnegie Mellon

Where do I come from?

PhD Linguistics (U of Maryland, USA, 2010)
Postdoc CS (Carnegie Mellon, 2010–2012)
Faculty CS (Carnegie Mellon, 2012–)
DeepMind (2016–)

What do I work on?

Linguistic structure in statistical/deep models

- better generalization, better sample complexity, more consistent in more languages

Discovering linguistic structure/units in data

- what kinds of linguistic knowledge can be inferred from the data?

Machine translation, (syntactic|semantic) parsing, representation learning, morphology, user-generated content, generation …

Practicals

Ground rules

- One **introductory practical** (more in a minute)
    - Turn in at beginning of week 3 or at the end of term
- Then, a **big multi-part practical** that you can do over the course of the term
    - “Choose your own adventure”
        - we provide a list of projects that you can choose from based on your interests (subject to some constraints)
        - You work at your own pace throughout the course
    - One report submitted at the end of term
    - Each student responsible for their own work

Practical topics I

Deep Learning for NLP

- **Perceiving and representing text** (and speech): “percepts” vs. “features”

Perceiving text

Deep Learning for NLP

- **Will you have the same "perceptual units" at test time as you have at training time?**
  - How do you handle *unknown words*?
  - Do you tokenize (segment) text into tokens? What are tokens? Is "New York City" one token? Or three?
  - Do you lowercase? Attempt to correct spelling?
  - Do you just treat text as a stream of characters? Bytes?

Perceiving text

Deep Learning for NLP

- **Will you have the same "perceptual units" at test time as you have at training time?**
  - How do you handle *unknown words*?
  - Do you tokenize (segment) text into tokens? What are tokens? Is "New York City" one token? Or three?
  - Do you lowercase? Attempt to correct spelling?
  - Do you just treat text as a stream of characters? Bytes?
- **Trade offs**
  - Smaller perceptual units need more complex models, but have fewer OOVs
  - Larger perceptual units can get away with simpler models, but OOVs are a problem

out of vocabulary

Companies must share benefits of globalisation, Theresa May tells Davos

PM says world's biggest firms must pay taxes and treat workers fairly, and market forces alone will not deliver for people

Theresa May has told the world's biggest companies they need to start paying their taxes and treat their workers more fairly in order to address the concerns of those who feel left behind by globalisation.

In a keynote speech to the World Economic Forum, the prime minister said governments could not rely on international market forces to deliver prosperity for everyone and action was needed to address the "deeply felt sense of economic inequality that has emerged in recent years".

May's warning that businesses needed to address the issue of executive pay and that market forces alone would not ensure the spread of prosperity to all ensured her first appearance in Davos was met with only lukewarm applause from her well-heeled audience.

The prime minister said that "across Europe, parties of the far left and the far right" were seeking to exploit the sense among people on modest incomes that globalisation was not working for them.

twitter

Taylor
@pbkingtaylor

Follow

@AtlantaFalcons finna make the highlight real on @YouTube before y'all claim the "hype" that's repeated every year.

9:04 PM - 17 Jan 2017

67 73

stephen.
@stephensh_arp

Follow

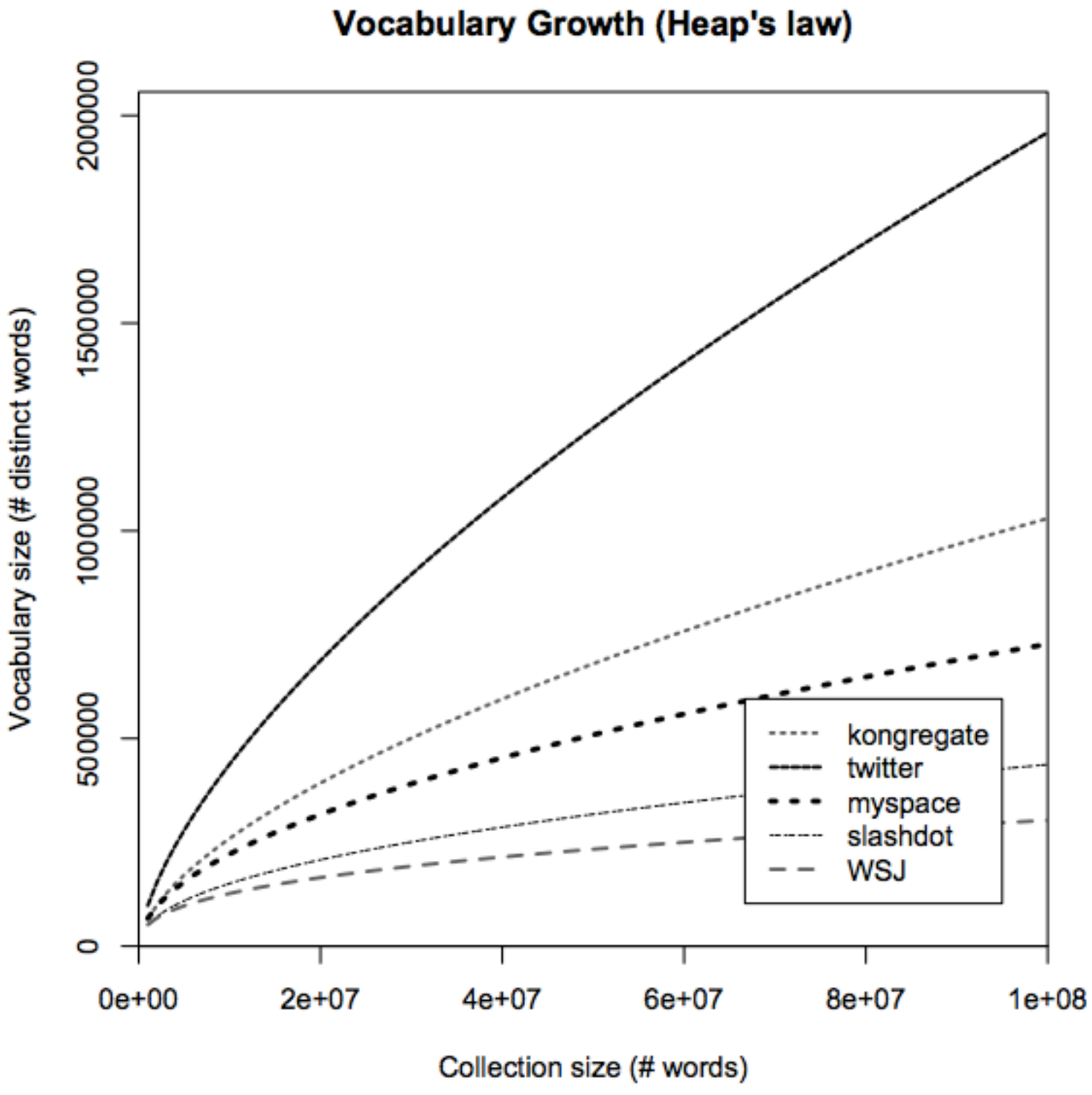
talking about Croatia and my mums just like "aye that's that auld Yugoslavia innit" don't think I've ever heard someone refer to it as that

4:25 PM - 9 Jan 2017

	Singletons	α
WSJ	34.3%	2.7
Twitter	70.0%	1.5

$$\log f(w) = \log C - \alpha \log(r(w) - b)$$

单词量和语料库大小存在这样的关系



Representing text

- Once we've engineered some percepts we can turn to…
- **Representation learning**
  - We want to compute representations of text by using it to predict something
  - As we saw in Part I of today's lecture, context is a pretty good representation of the lexical semantics of words
- **In practical 1**
  - You will explore representation learning based on predicting contexts
  - To do so, you will need to establish the vocabulary you operate on

Practical topics I

Deep Learning for NLP

practical 1

- **Perceiving and representing text** (and speech): “percepts” vs. “features”

Practical topics I

Deep Learning for NLP

practical 1

- **Perceiving and representing text** (and speech): “percepts” vs. “features”

- **Text categorisation** (“text cat”)

remaining practical

Practical topics I

Deep Learning for NLP

practical 1

- **Perceiving and representing text** (and speech): “percepts” vs. “features”

remaining practical

- **Text categorisation** (“text cat”)
- **Natural language generation**
  - language modelling
  - conditional language modelling
    - Conditional on a representation of context, generate appropriate text
    - Examples: speech recognition, caption generation

Practical topics II

Deep Learning for NLP

- **Natural language understanding**
  - Conditional language modeling applications (+NLG)
    - Translation, summarisation, conversational agents
  - Following instructions
  - Question answering, structured knowledge-base population
  - Dialogue

remaining practical

Practical topics II

Deep Learning for NLP

- **Natural language understanding**
  - Conditional language modeling applications (+NLG)
    - Translation, summarisation, conversational agents
    - Following instructions
    - Question answering, structured knowledge-base population
    - Dialogue
- **Analytic applications**
  - Topic modeling
  - Linguistic analysis (discourse, semantics, syntax, morphology)

remaining practical

Practicals

Deep Learning for NLP

- We are going to work with a single dataset that lets us explore many of these high level themes

- You will have the opportunity to transform a single dataset into a lot of different problems

- This is an important skill in real-world ML

    - Data has lots of value if you can imagine interesting things to do with it

- Outside of intro ML courses

Dataset

Deep Learning for NLP

Dataset

Deep Learning for NLP

- Appealing properties
  - Talks on many different topics but with a similar style
  - Enough data to learn interesting things, small enough to work on limited computational resources
- Interesting data of data associated with each talk
  - Topic labels, titles, summaries, topic labels, video, video alignments, translations into
- ***Who doesn't want to hear a computer-generated TED talk?***

Dataset

Deep Learning for NLP

```
  <head>
    <url>http://www.ted.com/talks/rajiv_maheswaran_the_math_behind_basketball_s_wildest_moves</url>
    <keywords>talks, math, sports, technology, visualizations</keywords>
    <speaker>Rajiv Maheswaran</speaker>
    <videourl>http://download.ted.com/talks/RajivMaheswaran_2015.mp4</videourl>
    <date>2015/03/17</date>
    <title>Rajiv Maheswaran: The math behind basketball's wildest moves</title>
    <description>TED Talk Subtitles and Transcript: Basketball is a fast-moving game of improvisation,
contact and, ahem, spatio-temporal pattern recognition. Rajiv Maheswaran and his colleagues are analyzing
the movements behind the key plays of the game, to help coaches and players combine intuition with new
data. Bonus: What they're learning could help us understand how humans move everywhere.</description>
    <transcription>
      <seekvideo id="954">My colleagues and I are fascinated by the science of moving dots.</seekvideo>
      <seekvideo id="4927">So what are these dots?</seekvideo>
      <seekvideo id="6101">Well, it's all of us.</seekvideo>
      <seekvideo id="7412">And we're moving in our homes, in our offices, as we shop and travel</seekvideo>
      . . .
```

Dataset

Deep Learning for NLP

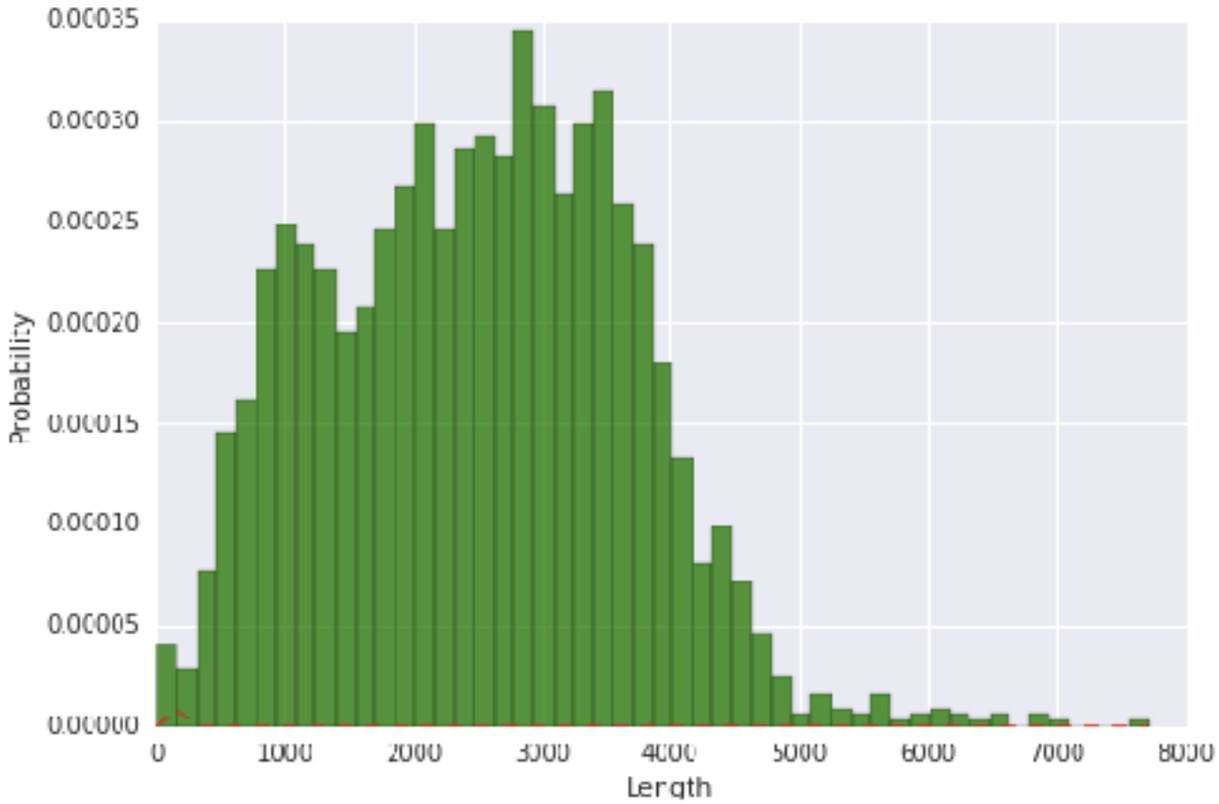
- The TED dataset contains **2,085 talks**, each containing:
    - Multiple **topic labels** (e.g., “talks, business, creativity, curiosity, goal-setting, innovation, motivation, potential, success, work”)
    - A **title** (12,766 words in all titles, excluding author names, which are always included in them)
    - A brief **content summary** (109,880 words in all summaries)
    - A content transcript with **alignments to frames** of the video recording (574,794 aligned segments)
        - Differences between subsequent video segments video frames give durations
    - Total content length is **5,201,252 tokens** when tokenized
    - Annotations for **applause** and **laughter**
    - **Translations** into over **100 languages** (although only a few languages have all talks)

Dataset

Deep Learning for NLP

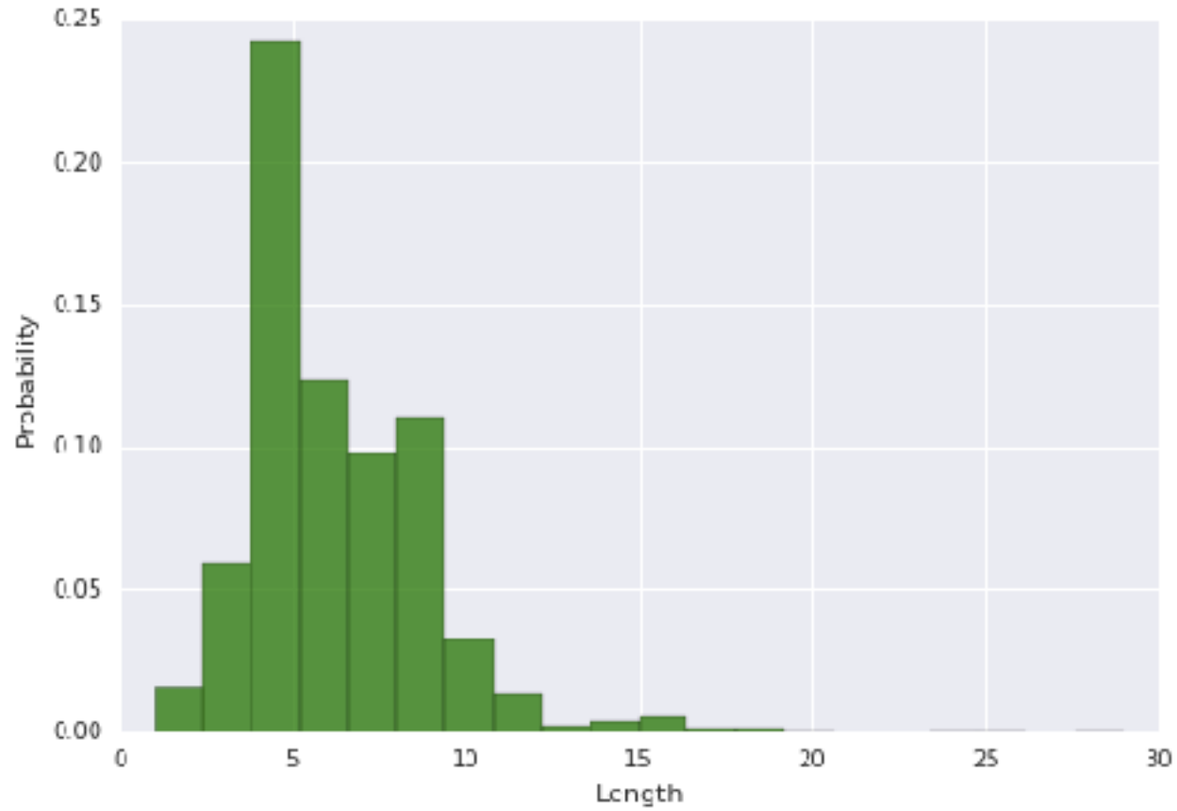
TED

Talk lengths (words)



(most talks are 2k-3k words)

of topic labels



(most talks have 4 keywords)

Practical topics I

Deep Learning for NLP

practical 1

- **Perceiving and representing text** (and speech): “percepts” vs. “features”

remaining practical

- **Text categorisation** (“text cat”)
- **Natural language generation**
  - language modelling
  - conditional language modelling
    - Conditional on a representation of context, generate appropriate text
    - Examples: speech recognition, caption generation

Practical topics I

Deep Learning for NLP

practical 1

- **Perceiving and representing text** (and speech): "percepts" vs. "features"

build word embeddings from TED talks

remaining practical

- **Text categorisation** ("text cat")
- **Natural language generation**
  - language modelling
  - conditional language modelling
    - Conditional on a representation of context, generate appropriate text
    - Examples: speech recognition, caption generation

Practical topics I

Deep Learning for NLP

practical 1

- **Perceiving and representing text** (and speech): "percepts" vs. "features"

build word embeddings from TED talks

- **Text categorisation** ("text cat")

predict talk labels

- **Natural language generation**
  - language modelling
  - conditional language modelling
    - Conditional on a representation of context, generate appropriate text
    - Examples: speech recognition, caption generation

remaining practical

Practical topics I

Deep Learning for NLP

practical 1

- **Perceiving and representing text** (and speech): "percepts" vs. "features"

build word embeddings from TED talks

- **Text categorisation** ("text cat")

predict talk labels

- **Natural language generation**
  - language modelling

generate random TED talks

  - conditional language modelling
    - Conditional on a representation of context, generate appropriate text
    - Examples: speech recognition, caption generation

remaining practical

Practical topics I

Deep Learning for NLP

practical 1

- **Perceiving and representing text** (and speech): "percepts" vs. "features"

build word embeddings from TED talks

- **Text categorisation** ("text cat")

predict talk labels

- **Natural language generation**
  - language modelling

generate random TED talks

  - conditional language modelling
    - Conditional on a representation of context, generate appropriate text

generate a TED talks about a topic

    - Examples: speech recognition, caption generation

remaining practical

Practical topics II

Deep Learning for NLP

- **Natural language understanding**
  - Conditional language modeling applications (+NLG)
    - Translation, summarisation, conversational agents
    - Following instructions
    - Question answering, structured knowledge-base population
    - Dialogue
- **Analytic applications**
  - Topic modeling
  - Linguistic analysis (discourse, semantics, syntax, morphology)

remaining practical

Practical topics II

Deep Learning for NLP

- **Natural language understanding**
  - Conditional language modeling applications (+NLG)
    - Translation, summarisation, conversational agents

build a TED talk translator

  - Following instructions
  - Question answering, structured knowledge-base population
  - Dialogue
- **Analytic applications**
  - Topic modeling
  - Linguistic analysis (discourse, semantics, syntax, morphology)

remaining practical

Practical topics II

Deep Learning for NLP

remaining practical

- **Natural language understanding**
  - Conditional language modeling applications (+NLG)
    - Translation, summarisation, conversational agents

build a TED talk translator

generate summaries from TED talks

  - Following instructions
  - Question answering, structured knowledge-base population
  - Dialogue
- **Analytic applications**
  - Topic modeling
  - Linguistic analysis (discourse, semantics, syntax, morphology)

Practical topics II

Deep Learning for NLP

remaining practical

- **Natural language understanding**
  - Conditional language modeling applications (+NLG)
    - Translation, summarisation, conversational agents

build a TED talk translator

generate summaries from TED talks

  - Following instructions
  - Question answering, structured knowledge-base population
  - Dialogue
- **Analytic applications**
  - Topic modeling
  - Linguistic analysis (discourse, semantics, syntax, morphology)

predict speaking for sentences

Practical topics II

Deep Learning for NLP

remaining practical

- **Natural language understanding**
  - Conditional language modeling applications (+NLG)
    - Translation, summarisation, conversational agents

build a TED talk translator

generate summaries from TED talks

  - Following instructions
  - Question answering, structured knowledge-base population
  - Dialogue
- **Analytic applications**
  - Topic modeling
  - Linguistic analysis (discourse, semantics, syntax, morphology)

predict speaking for sentences

when will the audience laugh?

Software in Practicals

Software Toolkits

- Deep learning operate on basic features with complex models that consist of common components stacked together
- Work flow
  - design model → implement → test → analyse → repeat

Software Toolkits

- Deep learning operate on basic features with complex models that consist of common components stacked together
- Work flow
  - design model → implement → test → analyse → repeat
- Implementation of models is non-trivial
  - Computations must be **fast**
  - Derivative calculations are **easy to get wrong**

Software Toolkits

- Deep learning operate on basic features with complex models that consist of common components stacked together
- Work flow
  - design model → implement → test → analyse → repeat
- Implementation of models is non-trivial
  - Computations must be **fast**
  - Derivative calculations are **easy to get wrong**
- Solution: toolkits that simplify implementation of models
  - standard component building blocks (lin alg, nonlinearities, convolutions, etc.)
  - facilities for **automatic differentiation**

Software Toolkits

- This course: **you are free to use your own toolkit**
  - However, demonstrators know
    - Torch (and interested in pytorch)
    - TensorFlow
- Many other options
  - Theano
  - DyNet (similar to pytorch, designed for language, fastest toolkit on CPU), written by me

Understanding Toolkits

- How do you declare computation graphs?
  - **Static** (e.g. TensorFlow, Theano)
    - Write down a symbolic expression representing all calculations you will carry out for different training instances
    - Toolkit optimizes it and gives you training/prediction code

Understanding Toolkits

- How do you declare computation graphs?
  - **Static** (e.g. TensorFlow, Theano)
    - Write down a symbolic expression representing all calculations you will carry out for different training instances
    - Toolkit optimizes it and gives you training/prediction code
- **Dynamic** (e.g., DyNet, pytorch)
  - Write code that computes predictions
  - Symbolic representation of computation is written down implicitly (based on operator overloading) by toolkit

Static vs. Dynamic

- **Static**
  - Pros: toolkits can optimize the computation graph (think: compilers)
  - Cons: you write code to write a symbolic program that the toolkit executes. The symbolic language is sometimes impoverished

Static vs. Dynamic

- **Static**
  - Pros: toolkits can optimize the computation graph (think: compilers)
  - Cons: you write code to write a symbolic program that the toolkit executes. The symbolic language is sometimes impoverished
- **Dynamic**
  - Pros: You write code in your favorite language (as long as your favorite language is C++ or Python)
  - Cons: Toolkit has fewer opportunities to optimize

Static vs. Dynamic

Model	Metric	Dynamic				Static	
		DyC++	DyPy	Chainer	DyC++ Seq	Theano	TF
RNNLM (MB=1)	words/sec	190	190	114	494	189	298
RNNLM (MB=4)	words/sec	830	825	295	1510	567	473
RNNLM (MB=16)	words/sec	1820	1880	794	2400	1100	606
RNNLM (MB=64)	words/sec	2440	2470	1340	2820	1260	636
BiLSTM Tag	words/sec	427	428	22.7	-	102	143
BiLSTM Tag+Char	words/sec	419	413	22.0	-	94.3	-
TreeLSTM	sents/sec	91.6	88.1	7.21	-	-	-

Enjoy the Practicals ;)