Improving Supervised Deep Learning with Unsupervised Learning

Eric Steinberger

MIT Europe Conference 2019
Single Deep Counterfactual Regret Minimization (SD-CFR)

• Find Nash Equilibria in humongous imperfect information games

• Use Deep Neural Networks to approximate Counterfactual Regret Minimization (CFR) to sample only a part of the game tree.

• Performs better and trains faster with less memory than Deep CFR (Brown et al.)
smART

- See: github.com/TinkeringCode/smart
- Team:
  - **Eric Steinberger** - Implementation of Style Transfer, ISS, stroke GA, and the RAPID translator
  - **Patrick Pelzmann** (TU Vienna) - Colour management & brush cleaning machines, hacking the ABB robot
  - **Benjamin Mörzinger** (TU Vienna) - Physical setup, management, supervision
  - **Manuel Stadler** (TU Vienna) - Helped setting up the ABB robot arm
  - **Alexander Raschendorfer** (TU Vienna) - Brush swapping mechanism
  - **Ralph Oswald** (TU Vienna) - Visualization of robot monitoring logs
Industry work

• Employee scheduling algorithm
• Predictive systems for various industries
• Domain-specific language prediction
• Concept work on matching algorithms
• Talks/workshops for companies
Motivation

**Your data:**
- Input: Some text or images
- Model should make some classification
- You have only 1000 examples :C

**But wait...**
- All books ever written
- Wikipedia
- Huge open-source images datasets
- Do we need more? There is more.
Humans vs. Supervised ML

*How humans are trained:*
- Years of seeing, reading, speaking
- Supervised Training
  - Parents, schools, books, ...
- Millions of years of evolution

*How AI is conventionally trained:*
- The target dataset
- Some hyperparameter tuning

© Eric Steinberger
Supervised Learning

\[ \text{Loss: } (164 - 35)^2 = 16641 \]

\[ \frac{\partial \text{Loss}}{\partial \text{Weight}} \]

Label: 35°C

© Eric Steinberger
Unsupervised Learning (this is just one kind of it)
Convolutional Neural Networks

Conv Layer 1 ➔ Conv Layer 2 ➔ Conv Layer 3 ➔ Conv Layer 4 ➔ Dense

Softmax

- P(Cat) = 0.03
- P(Dog) = 0.96
- P(Bird) = 0.01

ReLU

Conv gif: http://deeplearning.net/software/theano_versions/dev/tutorial/conv_arithmetic.html
Visualization: Unsupervised Learning of Hierarchical Representations with Convolutional DBNs [Honglak Lee et al. 2011]
New visualization (more convenient)
Basic Process: Train on your own Dataset

Task-Dataset → Conv Module → Weights → 1 class → Training
Supervised Pre-Training

ImageNet ▶️ Conv Module ▶️ Weights ▶️ 1 class ▶️ Training
Re-Training last few layers.

Fine Tuning.

© Eric Steinberger
Language Models. Let’s play a game!

Machine
Language Models. Let’s play a game!

Machine learning algorithms build a mathematical
Language Models. Let’s play a game!

Machine learning algorithms build a mathematical model of sample
Language Models. Let’s play a game!

Machine learning algorithms build a mathematical model of sample data, known as "training data", in
Language Models. Let’s play a game!

Machine learning algorithms build a mathematical model of sample data, known as “training data" , in order to make
Language Models. Let’s play a game!

Machine learning algorithms build a mathematical model of sample data, known as “training data”, in order to make predictions or decisions without being explicitly
Language Models. Let’s play a game!

Machine learning algorithms build a mathematical model of sample data, known as “training data”, in order to make predictions or decisions without being explicitly programmed.
Language Models. Let’s play a game!

Machine learning algorithms build a mathematical model of sample data, known as “training data", in order to make predictions or decisions without being explicitly programmed to perform the
Language Models. Let’s play a game!

Machine learning algorithms build a mathematical model of sample data, known as “training data”, in order to make predictions or decisions without being explicitly programmed to perform the task.
Language Models. Let’s play a game!

Machine learning algorithms build a mathematical model of sample data, known as “training data", in order to make predictions or decisions without being explicitly programmed to perform the task.

Source: https://en.wikipedia.org/wiki/Machine_learning

© Eric Steinberger
Natural Language Processing (NLP) tasks

- **Classification (e.g. sentiment):** “I love cats” -> Positive.
- **Language Models:** “I want to eat” -> “cake”
- **Translation:** “I like cats” -> “Ich mag Katzen”
- **Q&A:** “How gives this talk?” -> “Eric”
- **Similarity:** “I like cats” & “I like dogs” -> Yes.
- **Summarization:** long text -> summary

...
Recurrent Neural Networks (RNNs, LSTMs, GRUs, ...)

© Eric Steinberger
Recurrent Neural Networks

© Eric Steinberger
Recurrent Neural Networks: Seq2Seq Translation

Sequence to Sequence Learning with Neural Networks (Sutskever et al. 2014)
Transformers to replace RNNs

- Use Position- and Word/Character Embedding
- Masked Multi Attention
- Layer Norm
- Feedforward

Attention Is All You Need (Vaswani et al.; 2017)

© Eric Steinberger
Language Models (LM)

“Machine learning algorithms build a mathematical model of sample data, ...”
Language Models (LM)

“Machine learning algorithms build a mathematical model of sample data, ...”
"Machine learning algorithms build a mathematical model of sample data, ..."
Language Models (LM)

“Machine learning algorithms build a mathematical model of sample data, ...”

© Eric Steinberger
Language Models (LM)

“Machine learning algorithms build a mathematical model of sample data, ...”
Language Models (LM)

“Machine learning algorithms build a mathematical model of sample data, ...”
Transformer Decoder

Generating Wikipedia by summarizing long sequences (Liu et al.; 2018)

Transformer

Attention Is All You Need (Vaswani et al.; 2017)
Word Embeddings

• Say we know 20,000 words. Each word is a vector like \([0,0,0, ..., 0, 1, 0, ..., 0,0,0]\)
• Can we do better?
Word Embeddings: Word2Vec (Mikolov et al. 2013)

• E.g. “dog” = [..., 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, ...]
• We want: [0.23, 1.48, 0.62, 0.52, 1.14]
• Use the weights from input as embedding
Word Embeddings: Word2Vec (Mikolov et al. 2013)

Image: https://www.tensorflow.org/tutorials/representation/word2vec

© Eric Steinberger
Better Word Embeddings

- Word2Vec look-up ignores context of word in given sentence, e.g. spring:

- **ELMo (Embeddings from Language Models)** (Peters et al. Feb 2018)
  - Is contextual
  - Deep recurrent bi-directional language model
  - Learns linear combination of all internal layer representations for each downstream task
ULMFiT (Howard & Ruder; Jan 2018)

• Universal Language Model Fine-tuning (**ULMFiT**)
  • Successful transfer learning of task net in NLP
  • Until ULMFiT, this didn’t work well (Mou et al. 2016)

1. Pre-train LM on large data corpus
2. Fine-tune LM on target data
   • Fine-tunes later layers with higher learning rates
   • Slanted triangular learning rate schedules
3. Train classifier on target task
   • Concat pooling
   • Gradual unfreezing

© Eric Steinberger
Transformers?

- **GPT** *(Generative Pre-training of Transformers)* (Radford et al.; Jun 2018)
  - Why aren’t we using Transformers?
- **BERT** *(Bi-directional Encoder Representations from Transformers)* (Devlin et al.; Oct 2018)
  - Uses bi-directional Transformer Encoder
  - Masked bi-directional LM (e.g. Books can <PREDICT ME> you a lot about the world)
  - Next Sentence Prediction
- **GPT v2** (Radford et al.; Feb 2019)
  - Ditch all the fancy BERT stuff
  - bigger = better
  - Zero-shot LM evaluation, new SoTA.
  - Can perform tasks without ever being shown (although not SoTA)
Fine-tuning the GPT pre-trained Transformer

Figure 1: (left) Transformer architecture and training objectives used in this work. (right) Input transformations for fine-tuning on different tasks. We convert all structured inputs into token sequences to be processed by our pre-trained model, followed by a linear+softmax layer.

Improving Language Understanding by Generative Pre-Training [Radford et al. 2018]
Task embedding for GPT v2: e.g. Translation

\[\text{English sentence 1} = \text{French sentence 1} <X> \text{English sentence 2} = \text{French sentence 2} \ldots \text{English sentence} = \]\n
--> The network then continues “language modelling” this text with a French translation.
GPT v2

- #Parameters: 117M -> 1.5B
- Data: 40gb of diverse internet text
- Minor neural architecture modifications

Language Models are Unsupervised Multitask Learners [Radford et al. 2019]
GPT v2 results

Without any *dataset-specific* data!

<table>
<thead>
<tr>
<th>DATASET</th>
<th>METRIC</th>
<th>OUR RESULT</th>
<th>PREVIOUS RECORD</th>
<th>HUMAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Winograd Schema Challenge</td>
<td>accuracy (+)</td>
<td>70.70%</td>
<td>63.7%</td>
<td>92%+</td>
</tr>
<tr>
<td>LAMBADA</td>
<td>accuracy (+)</td>
<td>63.24%</td>
<td>56.25%</td>
<td>95%+</td>
</tr>
<tr>
<td>LAMBADA</td>
<td>perplexity (-)</td>
<td>8.6</td>
<td>99</td>
<td>-1-2</td>
</tr>
<tr>
<td>Children's Book Test Common Nouns (validation acc.)</td>
<td>accuracy (+)</td>
<td>93.30%</td>
<td>85.7%</td>
<td>96%</td>
</tr>
<tr>
<td>Children's Book Test Named Entities (validation acc.)</td>
<td>accuracy (+)</td>
<td>89.05%</td>
<td>82.3%</td>
<td>92%</td>
</tr>
<tr>
<td>Penn Tree Bank</td>
<td>perplexity (-)</td>
<td>35.76</td>
<td>46.54</td>
<td>unknown</td>
</tr>
<tr>
<td>WikiText-2</td>
<td>perplexity (-)</td>
<td>18.34</td>
<td>3914</td>
<td>unknown</td>
</tr>
<tr>
<td>enwik8</td>
<td>bits per character (-)</td>
<td>0.93</td>
<td>0.99</td>
<td>unknown</td>
</tr>
<tr>
<td>text8</td>
<td>bits per character (-)</td>
<td>0.98</td>
<td>1.08</td>
<td>unknown</td>
</tr>
<tr>
<td>WikiText-103</td>
<td>perplexity (-)</td>
<td>17.48</td>
<td>18.3</td>
<td>unknown</td>
</tr>
</tbody>
</table>

Language Models are Unsupervised Multitask Learners (Radford et al. @OpenAI)
GPT v2 zero-shot task transfer performance

Language Models are Unsupervised Multitask Learners (Radford et al. @OpenAI)
Takeaways

• Large networks can learn generally applicable knowledge from huge, diverse but labeled or unlabeled datasets

• These learnings can transfer to other domains and tasks

• In Deep Learning, bigger = better
Improving Supervised Deep Learning with Unsupervised Deep Learning

**Eric Steinberger**

E-mail: eric@steinberger-ai.com
GitHub: TinkeringCode
Twitter: @EricSteinb
LinkedIn: Eric Steinberger

*Learn more at*

Transformers & BERT: jalammar.github.io
GPT & GPT-2: blog.OpenAI.com
Deep Unsupervised Learning: Berkeley CS294-158 (YouTube)
Papers referenced on my slides are all on Arxiv.org

Q&A