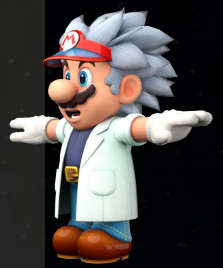
Fun-AI-Talk

- 10+ “Classic” & “Fun” NLP Papers



hululu.zhu@gmail.com

04/2022

How are you today?

Expected takeaways

- Understand some building blocks of NLP models
- Remember a few names/anecdotes related to NLP
- Roughly know something related to
  - train, pretrain, finetune and prompt-tuning
- Maybe ideas on how to apply to your current projects
- Or even ideas on how to improve NLP SOTA!
- Probably interest to join me to read and share!

About me, 10+ years at Google

My ML work experience starts at 2016 @ Google Fiber

intern

SWE

Just last week, my V1 WIP slides looked like below

- **technical**, but maybe quite **boring**

2013 Srivastava et al

- Dropout: A Simple Way to Prevent Neural Networks from Overfitting

How is code implemented?

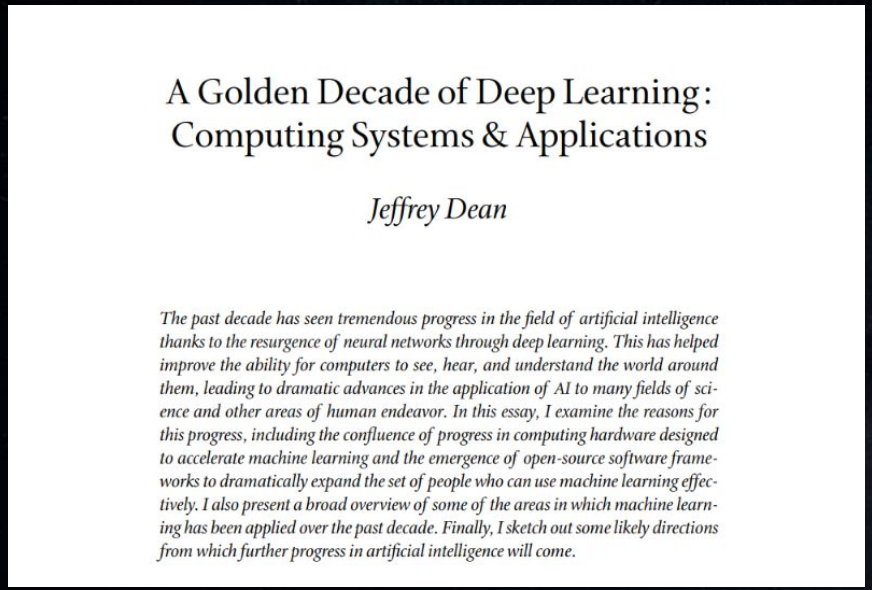
- Often “Inverted Dropout” used
  - Training time, weight for change of prob (e.g 50%) triggered neurons times 1/prob (e.g. 2x, basically increased weights in training)
  - Testing time, use all neuron with exact 1x weights

```
class Dropout():
    def __init__(self,prob=0.5):
        self.prob = prob
        self.params = []

    def forward(self, X, is_training=True):
        if not is_training:
            return X  # use all weights X if for evaluation
        # random binomial distribution to find triggered neurons
        self.mask = np.random.binomial(1,self.prob,X.shape)
        # divide by p in training as Inverted Dropout
        self.mask /= self.prob
        out = X * self.mask
        return out.reshape(X.shape)

    def backward(self,dout):
        # backprop by adjusting by Inverted Dropout p too
        dX = dout * self.mask
        return dX
```

I completely changed slides to a “fun” V2, after reading this Jeff Dean’s article published recently



A Golden Decade of Deep Learning: Computing Systems & Applications

Jeffrey Dean

The past decade has seen tremendous progress in the field of artificial intelligence thanks to the resurgence of neural networks through deep learning. This has helped improve the ability for computers to see, hear, and understand the world around them, leading to dramatic advances in the application of AI to many fields of science and other areas of human endeavor. In this essay, I examine the reasons for this progress, including the confluence of progress in computing hardware designed to accelerate machine learning and the emergence of open-source software frameworks to dramatically expand the set of people who can use machine learning effectively. I also present a broad overview of some of the areas in which machine learning has been applied over the past decade. Finally, I sketch out some likely directions from which further progress in artificial intelligence will come.

- Jeff Dean’s article is not technical, but very fun and encouraging to read!
- So I decided to mimic it
  - Not to focus on technical details
  - Instead, let me try to explain the “fun” aspects if any

My sources of DL/NLP paper recommendations

- High-quality articles
  - medium.com ($5 per month)
- Public Paper websites
  - paperswithcode.com
- Social networks
  - Chinese only: zhihu, wechat
  - Reddit
- Internal
  - go/cool-papers

All “classic” papers at a glance, let’s start!

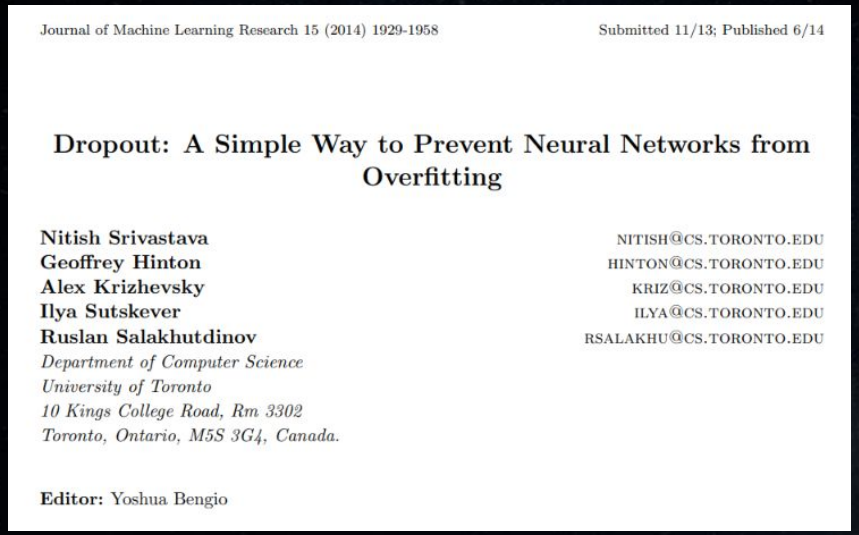
Deep Learning General

Year	Keyword/Links	Citations by 04/20/2022
2014	Dropout	*35176*
2014	Adam	*104486*
2015	Batch Normalization	*35996*
2015	Distillation	*9421*
2015	PReLU & Kaiming Initi	*15299*
2016	YT Recommendation	*2012*

NLP related

Year	Keyword/Links	Citations by 04/20/2022
2013	Word2Vec	*28263*
2014	Seq2Seq	*18404*
2017	Transformer/Attention	*40310*
2018	GLUE	*2723*
2018	BERT	*37413*
2019	T5	*3054*
2020	GPT3	*3533*

2013 Srivastava et al: Dropout



Journal of Machine Learning Research 15 (2014) 1929-1958 Submitted 11/13; Published 6/14

Dropout: A Simple Way to Prevent Neural Networks from Overfitting

Nitish Srivastava NITISH@CS.TORONTO.EDU
Geoffrey Hinton HINTON@CS.TORONTO.EDU
Alex Krizhevsky KRIZ@CS.TORONTO.EDU
Ilya Sutskever ILYA@CS.TORONTO.EDU
Ruslan Salakhutdinov RSALAKHU@CS.TORONTO.EDU
Department of Computer Science
University of Toronto
10 Kings College Road, Rm 3302
Toronto, Ontario, M5S 3G4, Canada.

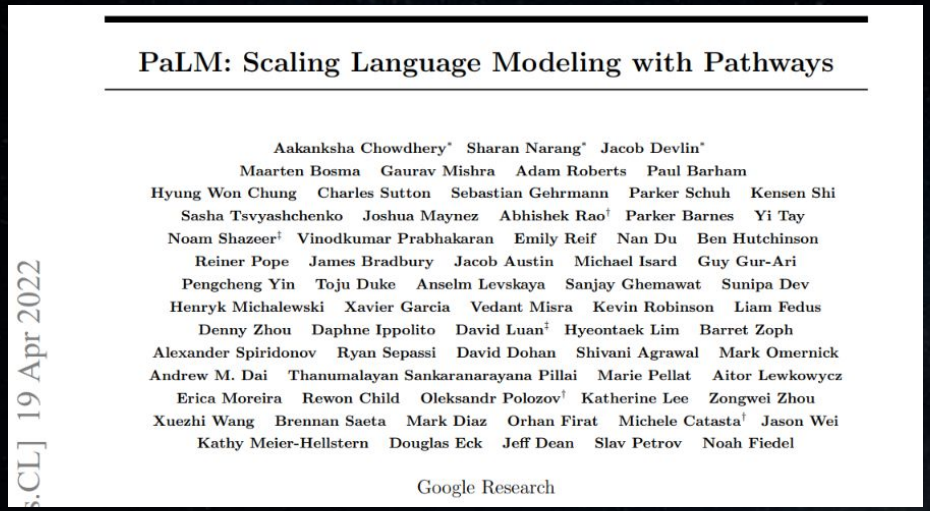
Editor: Yoshua Bengio

- Indeed “elegantly” simple
- *Background*: Overfit because of some “dominant” neurons
- *Solution*: Certain ratio to “deactivate” some neurons

Fun

- Training vs Test
- As if training an ensemble model
- 2nd author is Googler, & “Godfather of AI”
- 3rd author is famous for AlexNet with more than 110k citations!!!
- 4th author used to be Googler, now chief scientist of OpenAI
- Even the editor is also a Turing Award winner!!

2013 Srivastava et al: Dropout
- but 2022 PaLM has some unique recommendations



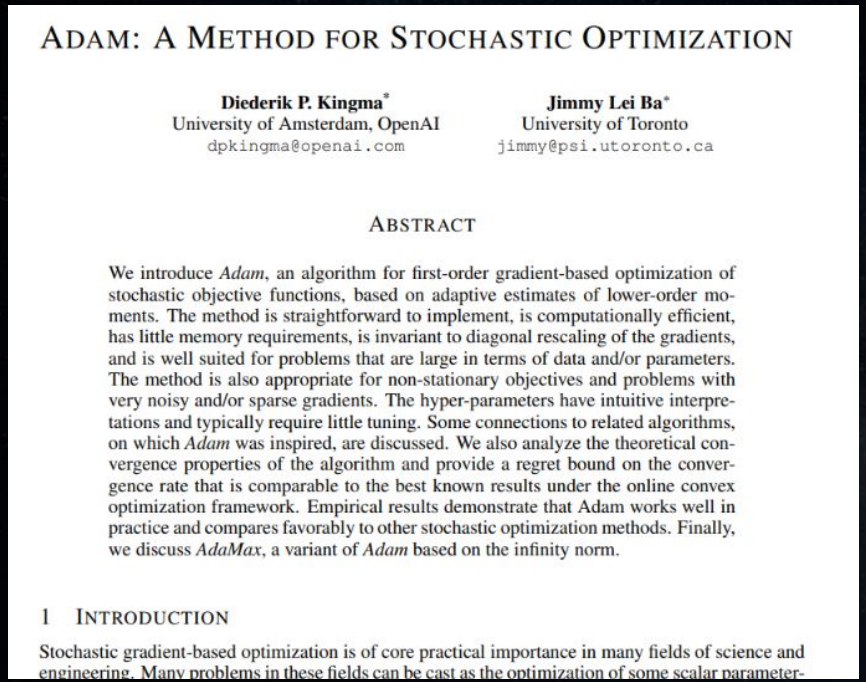
PaLM: Scaling Language Modeling with Pathways

Aakanksha Chowdhery* Sharan Narang* Jacob Devlin*
Maarten Bosma Gaurav Mishra Adam Roberts Paul Barham
Hyung Won Chung Charles Sutton Sebastian Gehrmann Parker Schuh Kensen Shi
Sasha Tsvyashchenko Joshua Maynez Abhishek Rao† Parker Barnes Yi Tay
Noam Shazeer‡ Vinodkumar Prabhakaran Emily Reif Nan Du Ben Hutchinson
Reiner Pope James Bradbury Jacob Austin Michael Isard Guy Gur-Ari
Pengcheng Yin Toju Duke Anselm Levskaya Sanjay Ghemawat Sunipa Dev
Henryk Michalewski Xavier Garcia Vedant Misra Kevin Robinson Liam Fedus
Denny Zhou Daphne Ippolito David Luan‡ Hyeontaek Lim Barret Zoph
Alexander Spiridonov Ryan Sepassi David Dohan Shivani Agrawal Mark Omernick
Andrew M. Dai Thanumalayan Sankaranarayana Pillai Marie Pellat Aitor Lewkowycz
Erica Moreira Rewon Child Oleksandr Polozov† Katherine Lee Zongwei Zhou
Xuezhi Wang Brennan Saeta Mark Diaz Orhan Firat Michele Catasta† Jason Wei
Kathy Meier-Hellstern Douglas Eck Jeff Dean Slav Petrov Noah Fiedel

Google Research

- **Dropout** – The model was trained without dropout, although dropout of 0.1 is used for finetuning in most cases.

2014 Kingma et al: Adam



ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION

Diederik P. Kingma*
University of Amsterdam, OpenAI
dpkingma@openai.com

Jimmy Lei Ba*
University of Toronto
jimmy@psi.utoronto.ca

ABSTRACT

We introduce *Adam*, an algorithm for first-order gradient-based optimization of stochastic objective functions, based on adaptive estimates of lower-order moments. The method is straightforward to implement, is computationally efficient, has little memory requirements, is invariant to diagonal rescaling of the gradients, and is well suited for problems that are large in terms of data and/or parameters. The method is also appropriate for non-stationary objectives and problems with very noisy and/or sparse gradients. The hyper-parameters have intuitive interpretations and typically require little tuning. Some connections to related algorithms, on which *Adam* was inspired, are discussed. We also analyze the theoretical convergence properties of the algorithm and provide a regret bound on the convergence rate that is comparable to the best known results under the online convex optimization framework. Empirical results demonstrate that Adam works well in practice and compares favorably to other stochastic optimization methods. Finally, we discuss *AdaMax*, a variant of *Adam* based on the infinity norm.

1 INTRODUCTION

Stochastic gradient-based optimization is of core practical importance in many fields of science and engineering. Many problems in these fields can be cast as the optimization of some scalar parameter-

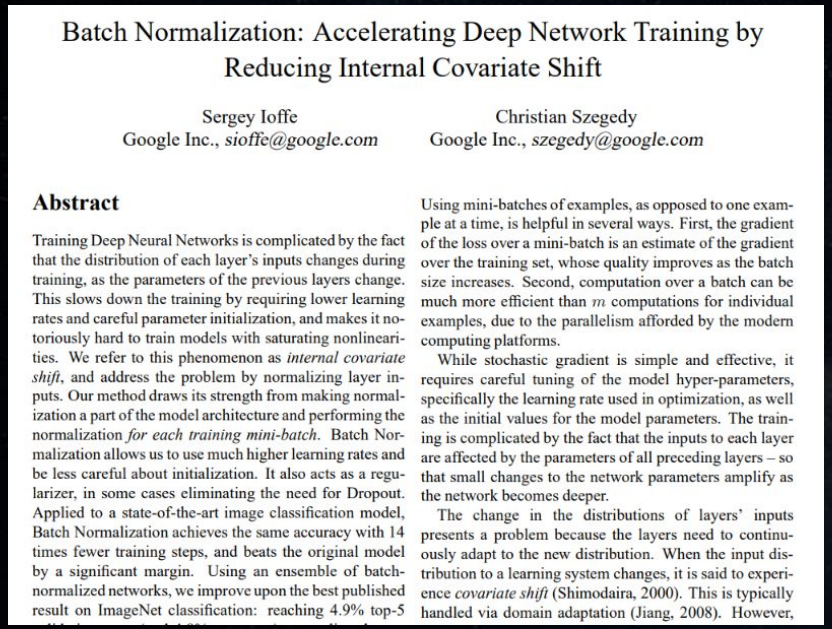
Optimizer Ancestor Tree IMO

- Grandparent: SGD
  - Parent: AdaGrad/RMSProp
  - Parent: Momentum
    - Adam
    - AdamW (**better** sibling)
      - Child: AdaFactor
      - Child: LAMB

Fun

- What is wrong with Adam that leads to AdamW?
- What is special for AdaFactor or LAMB?

2015 Ioffe et al: Batch Normalization



Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift

Sergey Ioffe
Google Inc., *sioffe@google.com*

Christian Szegedy
Google Inc., *szegedy@google.com*

Abstract

Training Deep Neural Networks is complicated by the fact that the distribution of each layer's inputs changes during training, as the parameters of the previous layers change. This slows down the training by requiring lower learning rates and careful parameter initialization, and makes it notoriously hard to train models with saturating nonlinearities. We refer to this phenomenon as *internal covariate shift*, and address the problem by normalizing layer inputs. Our method draws its strength from making normalization a part of the model architecture and performing the normalization *for each training mini-batch*. Batch Normalization allows us to use much higher learning rates and be less careful about initialization. It also acts as a regularizer, in some cases eliminating the need for Dropout. Applied to a state-of-the-art image classification model, Batch Normalization achieves the same accuracy with 14 times fewer training steps, and beats the original model by a significant margin. Using an ensemble of batch-normalized networks, we improve upon the best published result on ImageNet classification: reaching 4.9% top-5

Using mini-batches of examples, as opposed to one example at a time, is helpful in several ways. First, the gradient of the loss over a mini-batch is an estimate of the gradient over the training set, whose quality improves as the batch size increases. Second, computation over a batch can be much more efficient than m computations for individual examples, due to the parallelism afforded by the modern computing platforms.

While stochastic gradient is simple and effective, it requires careful tuning of the model hyper-parameters, specifically the learning rate used in optimization, as well as the initial values for the model parameters. The training is complicated by the fact that the inputs to each layer are affected by the parameters of all preceding layers – so that small changes to the network parameters amplify as the network becomes deeper.

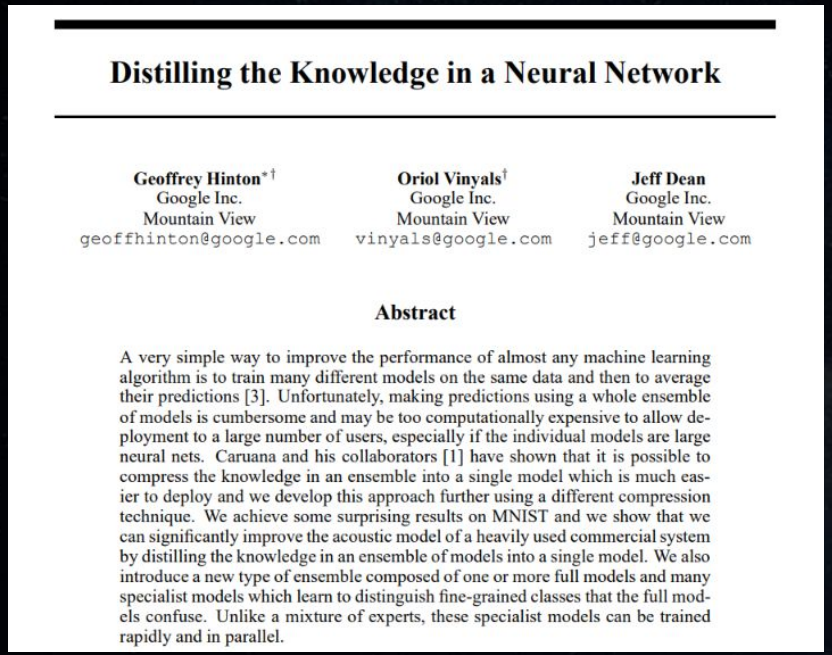
The change in the distributions of layers' inputs presents a problem because the layers need to continuously adapt to the new distribution. When the input distribution to a learning system changes, it is said to experience *covariate shift* (Shimodaira, 2000). This is typically handled via domain adaptation (Jiang, 2008). However,

- "Accelerating" in the title
  - Proper normalization speed up the convergence
- Why do we need BN?
  - Gradient vanishing or explosion after many layers
- Other normalization?
  - Layer normalization
    - RMS_norm
  - Group normalization
  - Weight normalization

Fun

- Is BN equal to get mean/std?
  - No, we also need to learn a linear transformation to NOT center by 0
  - "Scale and shift" after "normalization"
- Inference one single example?
  - Running average of mean/variance in training and applied in inference

2015 Hinton et al: Model Distillation



Distilling the Knowledge in a Neural Network

Geoffrey Hinton[*†] Google Inc. Mountain View geoffhinton@google.com

Oriol Vinyals[†] Google Inc. Mountain View vinyals@google.com

Jeff Dean Google Inc. Mountain View jeff@google.com

Abstract

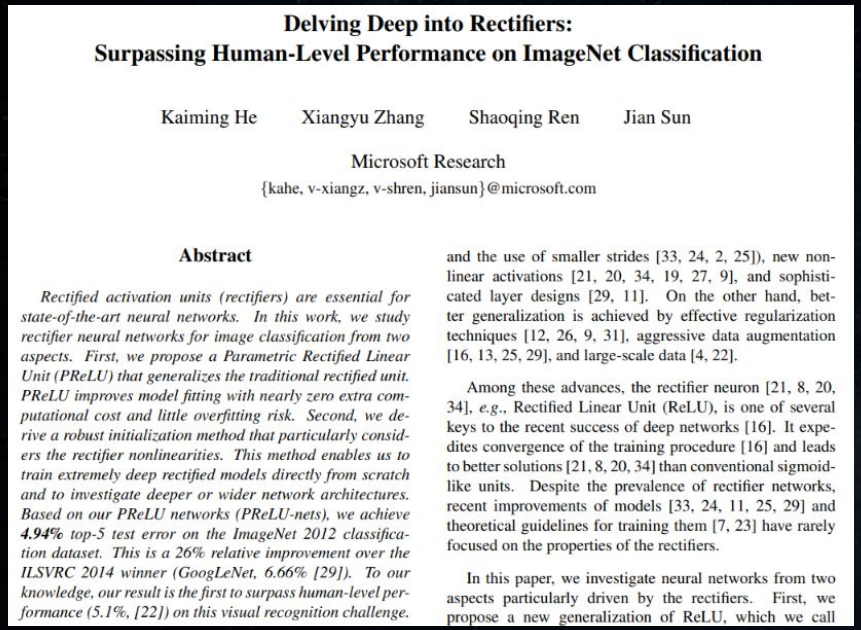
A very simple way to improve the performance of almost any machine learning algorithm is to train many different models on the same data and then to average their predictions [3]. Unfortunately, making predictions using a whole ensemble of models is cumbersome and may be too computationally expensive to allow deployment to a large number of users, especially if the individual models are large neural nets. Caruana and his collaborators [1] have shown that it is possible to compress the knowledge in an ensemble into a single model which is much easier to deploy and we develop this approach further using a different compression technique. We achieve some surprising results on MNIST and we show that we can significantly improve the acoustic model of a heavily used commercial system by distilling the knowledge in an ensemble of models into a single model. We also introduce a new type of ensemble composed of one or more full models and many specialist models which learn to distinguish fine-grained classes that the full models confuse. Unlike a mixture of experts, these specialist models can be trained rapidly and in parallel.

- Models are getting larger and larger!
  - Inference is slower and slower
- We need smaller model
  - A smaller model after training is much worse than large models
- A Teacher/Student paradigm
  - Larger teacher model trained first
  - Student model to “mimic” teacher
  - Student model can preserve much of teacher model power!

Fun

- ‘Godfather of AI’ collaborates with ‘Father of MapReduce, bigtable, tensorflow...’
- BERT serving at Google
- Roblox Blog: How We Scaled Bert To Serve 1+ Billion Daily Requests on CPUs

2015 He et al: PReLU & Kaiming Initialization



Delving Deep into Rectifiers:
Surpassing Human-Level Performance on ImageNet Classification

Kaiming He Xiangyu Zhang Shaoqing Ren Jian Sun

Microsoft Research

{kahe, v-xiangz, v-shren, jiansun}@microsoft.com

Abstract

Rectified activation units (rectifiers) are essential for state-of-the-art neural networks. In this work, we study rectifier neural networks for image classification from two aspects. First, we propose a Parametric Rectified Linear Unit (PReLU) that generalizes the traditional rectified unit. PReLU improves model fitting with nearly zero extra computational cost and little overfitting risk. Second, we derive a robust initialization method that particularly considers the rectifier nonlinearities. This method enables us to train extremely deep rectified models directly from scratch and to investigate deeper or wider network architectures. Based on our PReLU networks (PReLU-nets), we achieve ***4.94%*** *top-5 test error on the ImageNet 2012 classification dataset. This is a 26% relative improvement over the ILSVRC 2014 winner (GoogLeNet, 6.66% [29]). To our knowledge, our result is the first to surpass human-level performance (5.1%, [22]) on this visual recognition challenge.*

and the use of smaller strides [33, 24, 2, 25]), new nonlinear activations [21, 20, 34, 19, 27, 9], and sophisticated layer designs [29, 11]. On the other hand, better generalization is achieved by effective regularization techniques [12, 26, 9, 31], aggressive data augmentation [16, 13, 25, 29], and large-scale data [4, 22].

Among these advances, the rectifier neuron [21, 8, 20, 34], *e.g.*, Rectified Linear Unit (ReLU), is one of several keys to the recent success of deep networks [16]. It expedites convergence of the training procedure [16] and leads to better solutions [21, 8, 20, 34] than conventional sigmoid-like units. Despite the prevalence of rectifier networks, recent improvements of models [33, 24, 11, 25, 29] and theoretical guidelines for training them [7, 23] have rarely focused on the properties of the rectifiers.

In this paper, we investigate neural networks from two aspects particularly driven by the rectifiers. First, we propose a new generalization of ReLU, which we call

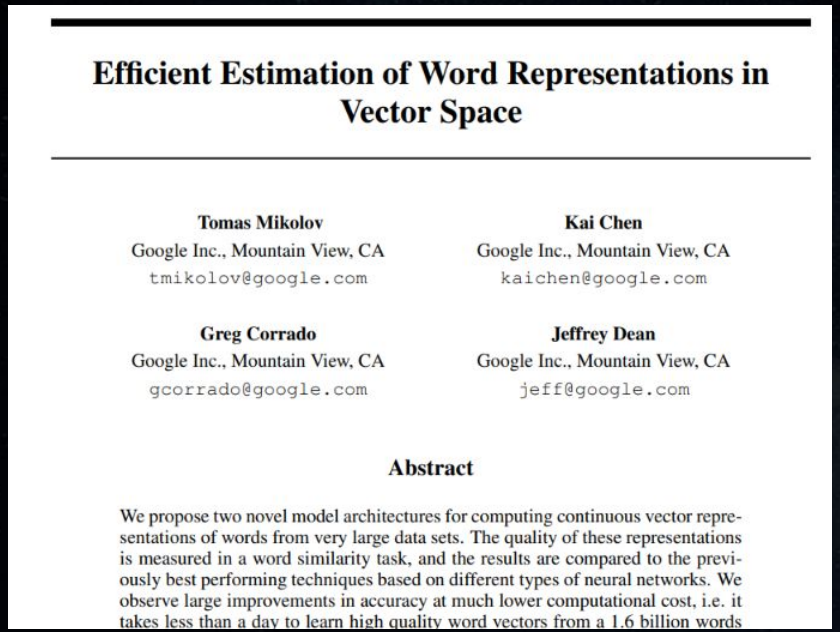
2 contributions

- A new Activation Function
  - Parametric ReLU
- Weight Initialization
  - Kaiming He Initialization
  - vs Glorot Initialization

Fun

- The first author has a more famous work ResNet with 110k+ citations!!!!
- Rumor said the first author got x million $ package when he joined FB in 2016!

2013 Mikolov et al: Word2Vec



Efficient Estimation of Word Representations in Vector Space

Tomas Mikolov
Google Inc., Mountain View, CA
tmikolov@google.com

Kai Chen
Google Inc., Mountain View, CA
kaichen@google.com

Greg Corrado
Google Inc., Mountain View, CA
gcorrado@google.com

Jeffrey Dean
Google Inc., Mountain View, CA
jeff@google.com

Abstract

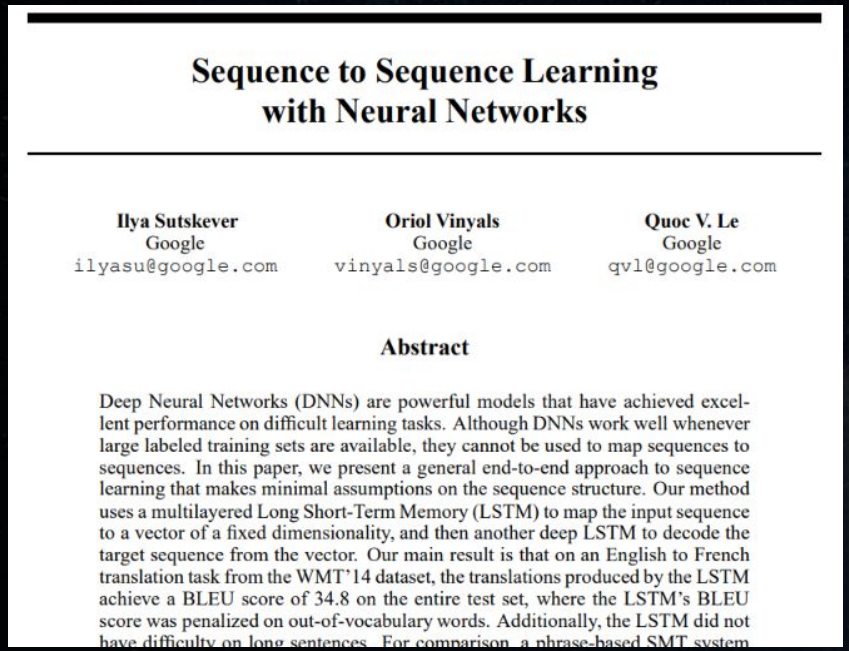
We propose two novel model architectures for computing continuous vector representations of words from very large data sets. The quality of these representations is measured in a word similarity task, and the results are compared to the previously best performing techniques based on different types of neural networks. We observe large improvements in accuracy at much lower computational cost, i.e. it takes less than a day to learn high quality word vectors from a 1.6 billion words

- Word2Vec (cbow vs skip-gram)
  - A sibling Glove to focus on co-occurrence
- Using
  - Mean(work_embeddings) often used as phrase embedding

Fun

- Input: “Woman”: ”Queen”, “Man”? Expect “King” if word2vec is properly trained
- Skip-gram is often preferred for less-popular words
- Jeff Dean stopped by a sec, pointed a direction, and bang! Huge success happens, as expected, because he is Jeff Dean...

2014 Sutskever et al: Seq2Seq



Sequence to Sequence Learning with Neural Networks

Ilya Sutskever
Google
ilyasu@google.com

Oriol Vinyals
Google
vinyals@google.com

Quoc V. Le
Google
qvl@google.com

Abstract

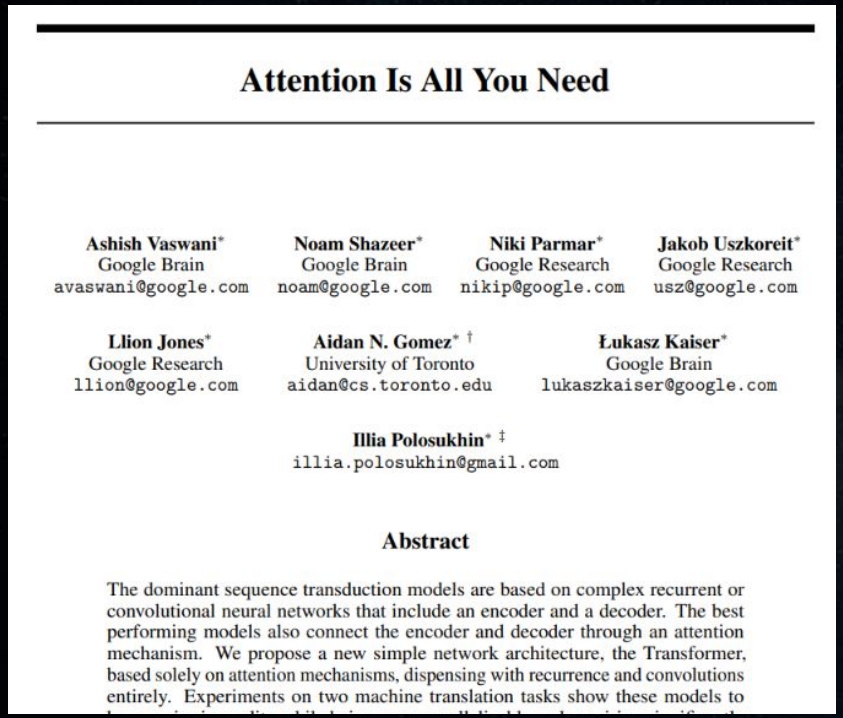
Deep Neural Networks (DNNs) are powerful models that have achieved excellent performance on difficult learning tasks. Although DNNs work well whenever large labeled training sets are available, they cannot be used to map sequences to sequences. In this paper, we present a general end-to-end approach to sequence learning that makes minimal assumptions on the sequence structure. Our method uses a multilayered Long Short-Term Memory (LSTM) to map the input sequence to a vector of a fixed dimensionality, and then another deep LSTM to decode the target sequence from the vector. Our main result is that on an English to French translation task from the WMT'14 dataset, the translations produced by the LSTM achieve a BLEU score of 34.8 on the entire test set, where the LSTM's BLEU score was penalized on out-of-vocabulary words. Additionally, the LSTM did not have difficulty on long sentences. For comparison, a phrase-based SMT system

- Seq2Seq, aka encoder-decoder
- Start with RNN/LSTM
- Pretty popular for neural translation
- Pretty cool for some toy applications
  - Write paper or write code

Fun

- First author now is the Chief Scientist of OpenAI, latest work is AI for Solving Math Olympiad problems
- 3rd author is also famous for his AutoML and Neural-arch-search work
  - Rumor says he was promoted almost every year in a consecutive 5 years ^_^

2013 Vaswani et al: Transformer/Attention



Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com

Noam Shazeer* Google Brain noam@google.com

Niki Parmar* Google Research nikip@google.com

Jakob Uszkoreit* Google Research usz@google.com

Llion Jones* Google Research llion@google.com

Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu

Łukasz Kaiser* Google Brain lukaszkaiser@google.com

Illia Polosukhin* ‡ illia.polosukhin@gmail.com

Abstract

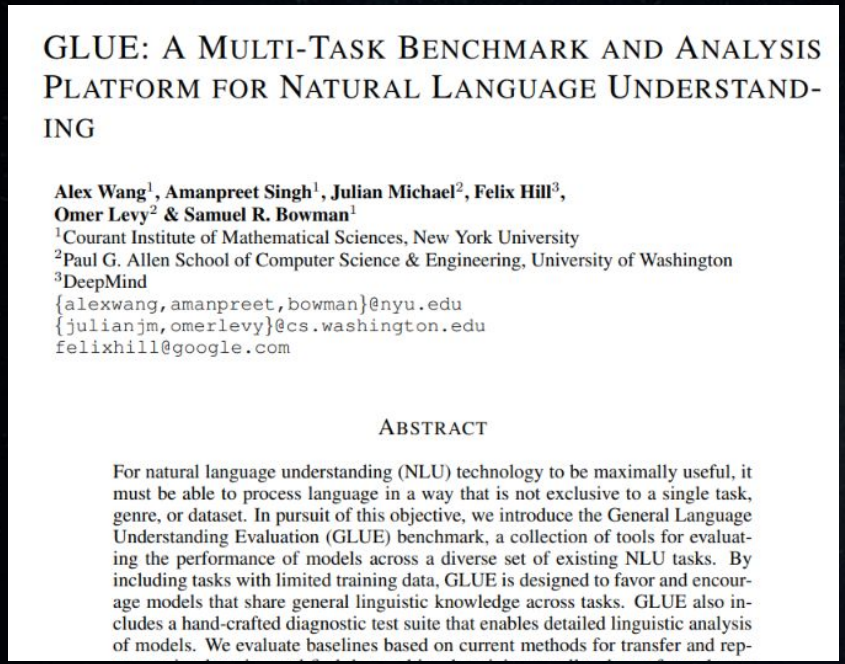
The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to

- The famous transformer paper
- The famous KQV formula with trig func based position encoding
- Visually complex arch, but more “parallel” efficient than RNN regarding computation
- Has huuuuuge impact! Self-attention transformer from this is the foundation for all large language/multi-modal models as of 04/2022!!!

Fun

- The first of later abused “xFormer” work (ref survey paper)
- The first of later abused “xx is all your need” paper title template

2018 Wang et al: GLUE



GLUE: A MULTI-TASK BENCHMARK AND ANALYSIS PLATFORM FOR NATURAL LANGUAGE UNDERSTANDING

Alex Wang[1], **Amanpreet Singh**[1], **Julian Michael**[2], **Felix Hill**[3], **Omer Levy**[2] **& Samuel R. Bowman**[1]
[1]Courant Institute of Mathematical Sciences, New York University
[2]Paul G. Allen School of Computer Science & Engineering, University of Washington
[3]DeepMind
{alexwang,amanpreet,bowman}@nyu.edu
{julianjm,omerlevy}@cs.washington.edu
felixhill@google.com

ABSTRACT

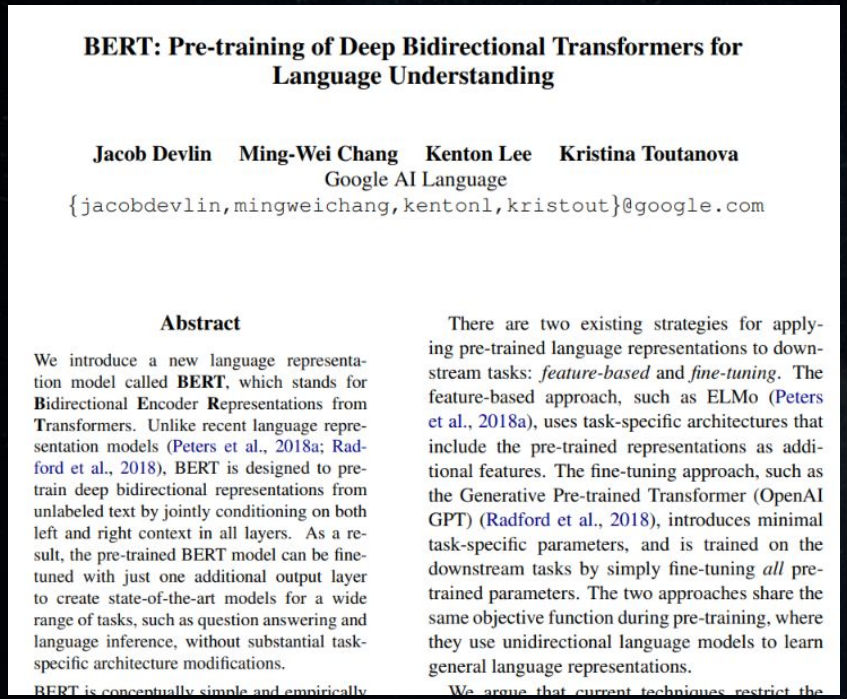
For natural language understanding (NLU) technology to be maximally useful, it must be able to process language in a way that is not exclusive to a single task, genre, or dataset. In pursuit of this objective, we introduce the General Language Understanding Evaluation (GLUE) benchmark, a collection of tools for evaluating the performance of models across a diverse set of existing NLU tasks. By including tasks with limited training data, GLUE is designed to favor and encourage models that share general linguistic knowledge across tasks. GLUE also includes a hand-crafted diagnostic test suite that enables detailed linguistic analysis of models. We evaluate baselines based on current methods for transfer and rep-

- NLP Benchmark
- So called “SOTA”
- “ImageNet” for NLP
- 9 sentence or sentence-pair language understanding tasks, see leaderboard

Fun

- Soon GLUE becomes too easy after “BERT”, so it was replaced by
  - SuperGLUE
  - Big-Bench
    - “Beyond the Imitation Game”
  - And more

2018 Devlin et al: BERT



BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

Jacob Devlin **Ming-Wei Chang** **Kenton Lee** **Kristina Toutanova**
Google AI Language
{jacobdevlin,mingweichang,kentonl,kristout}@google.com

Abstract

We introduce a new language representation model called **BERT**, which stands for **B**idirectional **E**ncoder **R**epresentations from **T**ransformers. Unlike recent language representation models (Peters et al., 2018a; Radford et al., 2018), BERT is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. As a result, the pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and language inference, without substantial task-specific architecture modifications.

BERT is conceptually simple and empirically

There are two existing strategies for applying pre-trained language representations to downstream tasks: *feature-based* and *fine-tuning*. The feature-based approach, such as ELMo (Peters et al., 2018a), uses task-specific architectures that include the pre-trained representations as additional features. The fine-tuning approach, such as the Generative Pre-trained Transformer (OpenAI GPT) (Radford et al., 2018), introduces minimal task-specific parameters, and is trained on the downstream tasks by simply fine-tuning *all* pre-trained parameters. The two approaches share the same objective function during pre-training, where they use unidirectional language models to learn general language representations.

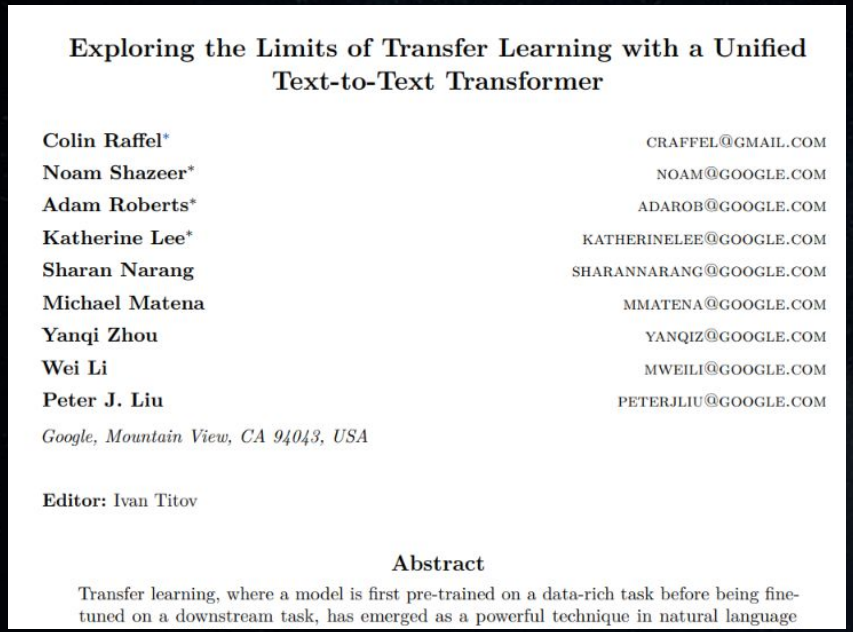
We argue that current techniques restrict the

- Encoder only transformers
- 15% masked prediction training
- Next sentence prediction
- Special CLS/SEP tokens
- New paradigm of “Pretrain and Finetune”

Fun

- Why called BERT, because of Sesame Street!
  - ELMo by AllenNLP
  - Ernie by Baidu
  - Big bird by Google
- Mask rate: 2022 research says 40%, MAE paper (by author of ResNet) says ViT mask prefer 75%
- 10% corrupted masked prediction training!
- How to use BERT embedding or finetune?
- FB RoBERTa said “Next sentence prediction” not necessary

2019 Raffel et al: T5



Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer

Colin Raffel* CRAFFEL@GMAIL.COM
Noam Shazeer* NOAM@GOOGLE.COM
Adam Roberts* ADAROB@GOOGLE.COM
Katherine Lee* KATHERINELEE@GOOGLE.COM
Sharan Narang SHARANNARANG@GOOGLE.COM
Michael Matena MMATENA@GOOGLE.COM
Yanqi Zhou YANQIZ@GOOGLE.COM
Wei Li MWEILI@GOOGLE.COM
Peter J. Liu PETERJLIU@GOOGLE.COM

Google, Mountain View, CA 94043, USA

Editor: Ivan Titov

Abstract

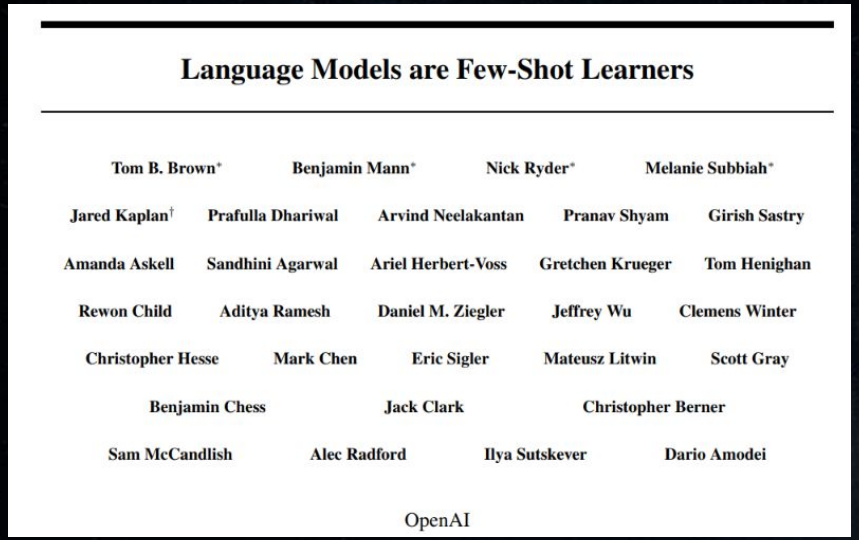
Transfer learning, where a model is first pre-trained on a data-rich task before being fine-tuned on a downstream task, has emerged as a powerful technique in natural language

- Transform every task into text-to-text!
  - Potential candidate for multi-task
  - Consistent pretraining/finetune objectives
- Same 15% mask & 10% corruption rates as BERT
- Mixing multiple objective in finetune
- Greedy decoding instead of more sophisticated decoding (e.g. beam search)

Fun

- T5 does not use standard transformer
  - Relative embedding other than positional encoding
  - Other changes to activation function and normalization
- Pretrain only read partial data of C4, less computation than BERT, but better results
- T5 is more a framework, see T5X
- Multi-language: MT5, ByT5

2020 Brown et al: GPT3



Language Models are Few-Shot Learners

Tom B. Brown* Benjamin Mann* Nick Ryder* Melanie Subbiah*

Jared Kaplan† Prafulla Dhariwal Arvind Neelakantan Pranav Shyam Girish Sastry

Amanda Askell Sandhini Agarwal Ariel Herbert-Voss Gretchen Krueger Tom Henighan

Rewon Child Aditya Ramesh Daniel M. Ziegler Jeffrey Wu Clemens Winter

Christopher Hesse Mark Chen Eric Sigler Mateusz Litwin Scott Gray

Benjamin Chess Jack Clark Christopher Berner

Sam McCandlish Alec Radford Ilya Sutskever Dario Amodei

OpenAI

- GPT is a family of "autoregressive" models
  - GPT, GPT2, GPT3, Codex (github copilot)
  - Decoder-only, FYI some Google decoder-only models: LaMDA, PaLM
  - With small modifications to Transformer
- The awesome "zero-shot" or "few-shot" prompt tuning paradigm!
  - As compared to "finetuning"

Fun

- Can "performing 3-digit arithmetic"
- The so called "dangerous" and "cannot share" model from OpenAI, who sold its Soul for $1bn
  - Alternative GPT-Neo, GPT-J by EleutherAI
- Cost 10-20 million USD to train!!
- And 😂

Unfortunately, a bug in the filtering caused us to ignore some overlaps, and due to the cost of training it was not feasible to retrain the model.

?? 2016 Covington et al: YT Recommendation

Deep Neural Networks for YouTube Recommendations

Paul Covington, Jay Adams, Emre Sargin
Google
Mountain View, CA
{pcovington, jka, msargin}@google.com

ABSTRACT

YouTube represents one of the largest scale and most sophisticated industrial recommendation systems in existence. In this paper, we describe the system at a high level and focus on the dramatic performance improvements brought by deep learning. The paper is split according to the classic two-stage information retrieval dichotomy: first, we detail a deep candidate generation model and then describe a separate deep ranking model. We also provide practical lessons and insights derived from designing, iterating and maintaining a massive recommendation system with enormous user-facing impact.

Keywords

recommender system; deep learning; scalability

1. INTRODUCTION

YouTube is the world's largest platform for creating, sharing and discovering video content. YouTube recommendations are responsible for helping more than a billion users

- A deep dive into a large-scale deep recommendation model
- A deep neural network to predict prob of watch given

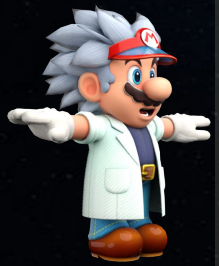
Fun

- Feature engineering
  - E.g. deal with fresh/viral features
- Training vs Inference (retrieval)
  - Sampling softmax instead of hierarchy softmax to speed up cross-entropy computation
  - ANN for fast retrieval

Expected takeaways

- Understand some building blocks of NLP models
- Remember a few names/anecdotes related to NLP
- Roughly know something related to
  - train, pretrain, finetune and prompt-tuning
- Maybe ideas on how to apply to your current projects
- Or even ideas on how to improve NLP SOTA!
- Probably interest to join me to read and share!

Thank you! Questions?



Fun-AI-Talk

- 10+ “Classic” & “Fun” NLP Papers

hululu.zhu@gmail.com

04/2022