Good Enough ML
with Google Cloud Platform

by Yasen Kiprov

About me

- 13 years of experience, dev, entrepreneur, PhD candidate, freelancer, datathon winner
- R&D team lead in AI @ SiteGround
- Interests: Sentiment analysis, named entity recognition, question answering, dialogue systems, recently: marketing, image processing

I believe in ML there are no perfect models, but some are good enough.

Quick Ad

- Regular meetups on Data Science and Python
- Usually **free beer**
- Coming up 2.08.2021 LIVE
- Looking for speakers

PyData
SOFIA

https://www.meetup.com/PyData-Sofia/

Agenda

- Current State of Machine Learning

- Google Cloud platform

- Vertex AI

Transformer

Attention-only based encoder-decoder network

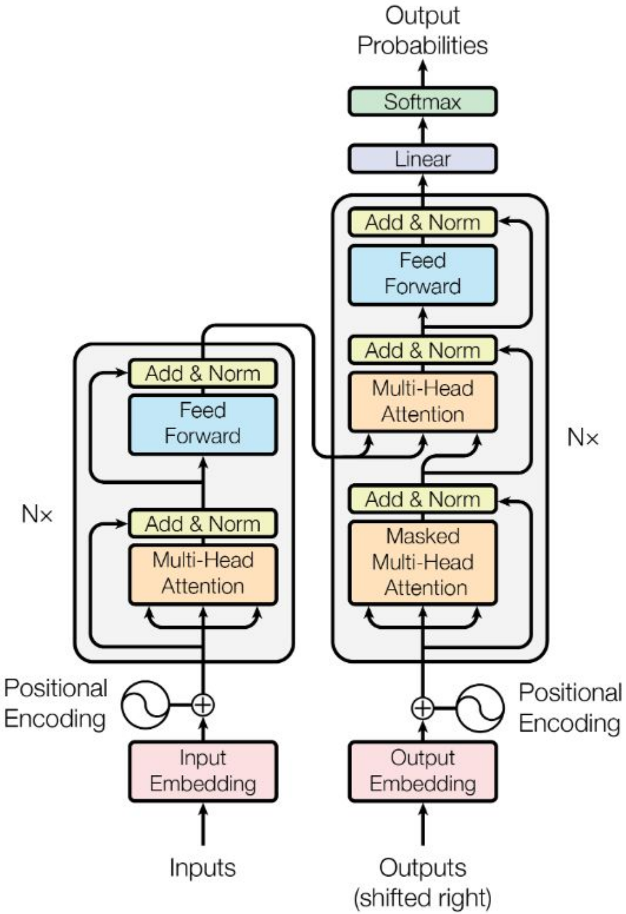


Figure 1: The Transformer - model architecture.

Bert

- Ground-breaking: pre-training on huge unlabeled datasets
  - Masked language model
  - Next sentence prediction
- Huge amounts of unlabeled data
- Fine-tuning on specific tasks
- State of art / game changer in 2019

Pre-training of Deep Bidirectional Transformers for Language Understanding (Devlin et. al, 2019)

The Bert Family

- RoBerta
- Distilbert, Albert, StructBert
- Ernie
- Electra
- GPT, T5, Pegasus
- Reformer

Result: near-human performance

TODO: show Glue benchmark

Size and cost of training (NLP)

- **Bert**: 350M parameters, three days on 16 TPUv3 chips
- “It costs $245,000 to train the **XLNet model** … **512 TPU v3** chips * 2.5 days”
- **T5**: **11B** parameters: “[we] train models on “slices” of Cloud TPU Pods.TPU pods are multi-rack ML supercomputers that contain 1,024TPU v3 chips”
- **GPT-3: 175B** parameters: “... memory requirement exceeding 350GB and training costs exceeding $12 million”

- **Wu Dao: 1.75 trillion** parameters, **trained on 4.9TB** of text and image data

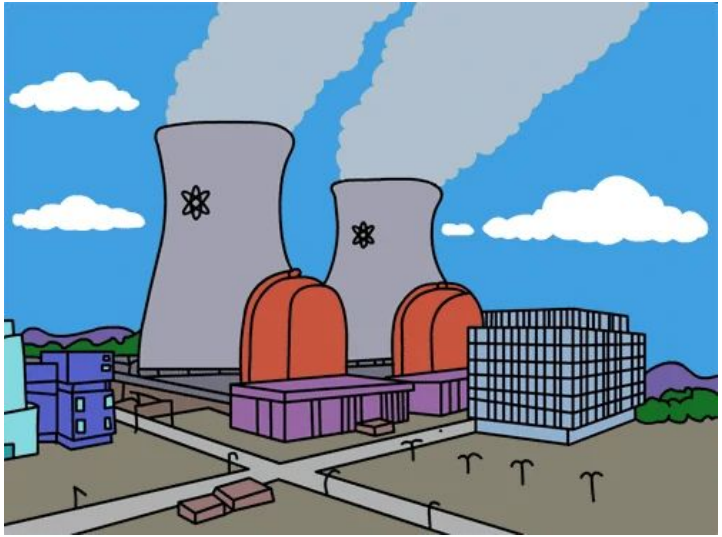
And in Image Classification

“Scaling Vision Transformers” - current SOTA

Models ranging from 5 million to **2 billion parameters**

Datasets ranging from 30 million to
3 billion training images

Compute budgets go
beyond 10 000 core-days on TPUv3



https://arxiv.org/pdf/2106.04560v1.pdf

Conclusion

My (biased) observation for the past 2 years of NN research:

- Optimizations in training speed and memory usage to allow **bigger models**
- Smart ideas how to **use more data**
- **Optimizations** to utilize data better

Massive pre-training + Optimisations + Huge Models =

State of art models
Models capable of few-shot learning

HuggingFace (last year)

Transformers: a community-driven OS library for NLP

Provides a **simple interface** and **great docs**

Pre-trained models for many state-of-art NLP systems

Good for out-of-domain researchers / developers

(also see https://paperswithcode.com/ and github)

Google Cloud Platform

A suite of **cloud computing services** that runs on Google infrastructure

- File storage
- Computing (Virtual Machines)
- DB / Streaming / Big Data / BigQuery
- Management / Monitoring / Ops
- Machine Learning

Google Cloud Platform provides infrastructure as a service, platform as a service, and serverless computing environments. (Wikipedia)

Google AI

- A lot of pretrained ready to use ML solutions
- AutoML provides ways to fine-tune many of them
  - Regularly updated with new models and features

Supported Project Stages

- Data Upload
- Data Observation & Split
- Training models
- Evaluation
- Automated deployment & scaling

Why Google AI

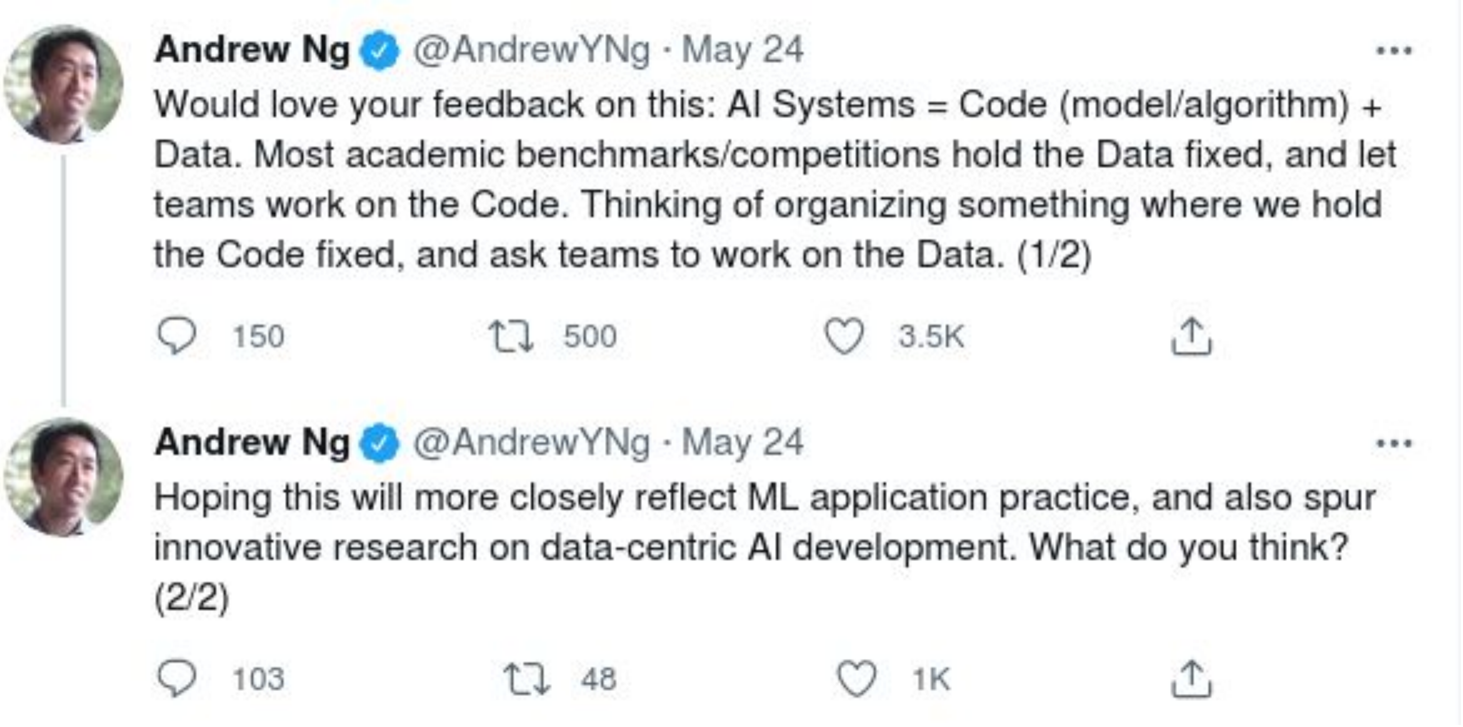
Pros:

- Easy to create proof-of-concepts
  - We can’t be experts in everything
- A “good enough” baseline
  - Fine tuning works well
  - Sometimes very hard to beat (translate)
- Production-ready
  - Managed service
  - Auto deployed & scaled
  - MLOps

Cons:

- Limited set of tasks
  - Some customizations may be hard
- Limited to their models
- Requires $$

Reality is data-centric



Andrew Ng @AndrewYNg · May 24

Would love your feedback on this: AI Systems = Code (model/algorithm) + Data. Most academic benchmarks/competitions hold the Data fixed, and let teams work on the Code. Thinking of organizing something where we hold the Code fixed, and ask teams to work on the Data. (1/2)

Andrew Ng @AndrewYNg · May 24

Hoping this will more closely reflect ML application practice, and also spur innovative research on data-centric AI development. What do you think? (2/2)

AI Tasks

Cloud Vision

- Image classification
- Face detection
- OCR
- Explicit content, etc.

Video Intelligence

- Object detection
- Classification

AutoML Vision

- Image Classification
- Object detection

AutoML Video

- Object detection
- Classification of shots and segments

AI Tasks

Cloud NLP

- Classification
- Entity Extraction
- Sentiment Analysis
- Entity Sentiment Analysis
- Medical Entity Extraction

AutoML NLP

- Classification
- Entity Extraction
- Sentiment analysis

AI Tasks

Other default models

- Translation
- Document parsing
    - Form parsing
    - OCR
    - Human in the loop

Other Trainable Models

- AutoML Translation
    - Import sentence pairs
    - Tune google translate
- Recommendations
    - Import products
    - Log user transactions
    - Train models
- Tables (beta)

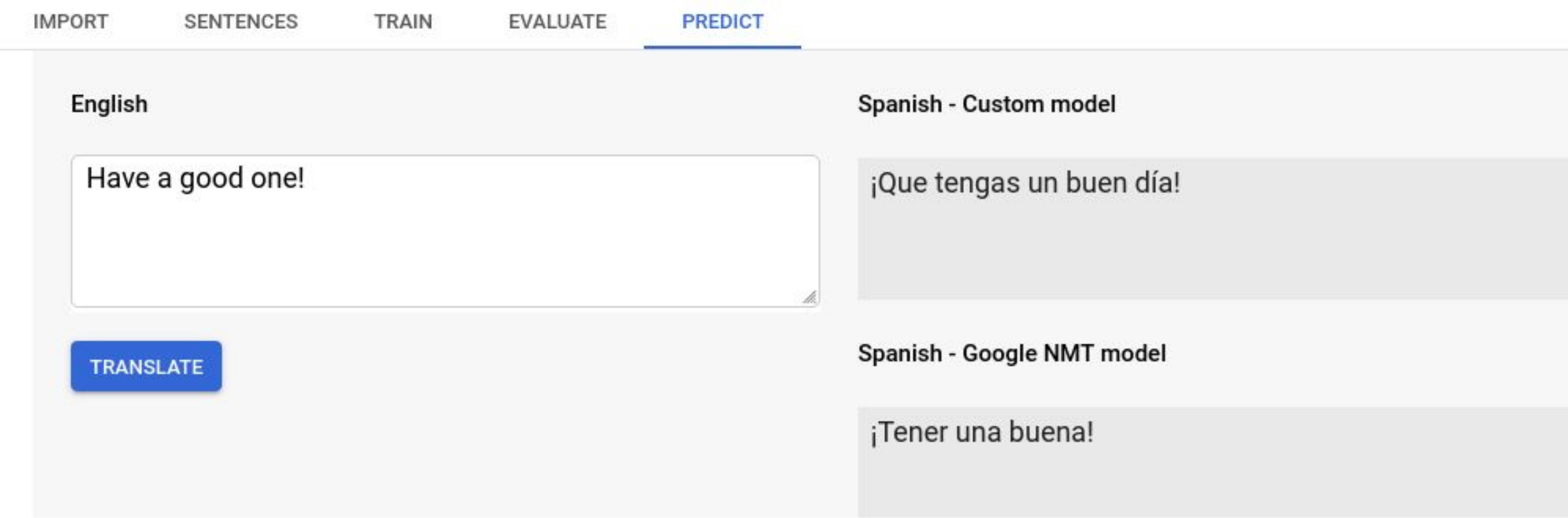
AutoML Demo

Translation

Tables

Image classification

Translate Prediction



English

Have a good one!

Spanish - Custom model

¡Que tengas un buen día!

Spanish - Google NMT model

¡Tener una buena!

Translate Use in Production

```
from google.cloud import translate
client = translate.TranslationServiceClient()

project_id = 'XXX'
text = 'YOUR_SOURCE_CONTENT'
location = 'us-central1'
model = 'projects/XXX/locations/us-central1/models/XXX'

parent = client.location_path(project_id, location)

response = client.translate_text(
   parent=parent,
   contents=[text],
   model=model,
   mime_type='text/plain',  # mime types: text/plain, text/html
   source_language_code='en',
   target_language_code='es')

for translation in response.translations:
 print('Translated Text: {}'.format(unicode(translation).encode('utf8')))
```

Vertex AI - the new home of AutoML

A toolkit to solve MLops on the google cloud

- Prepare data
  - Use cloud storage, big query, upload
  - Human labeling
- Train Models
  - AutoML
  - Custom: **Literally anything in a docker**
- Evaluate
- Deploy
  - Auto scale
  - **Anything in a docker**

Vertex AI Overview

Alternative Platforms

- Microsoft Azure Cognitive Services
  - We used it for image captioning

- Amazon AWS Sage Maker

- IBM Watson Cloud

Thank you!

Questions?

https://console.cloud.google.com/