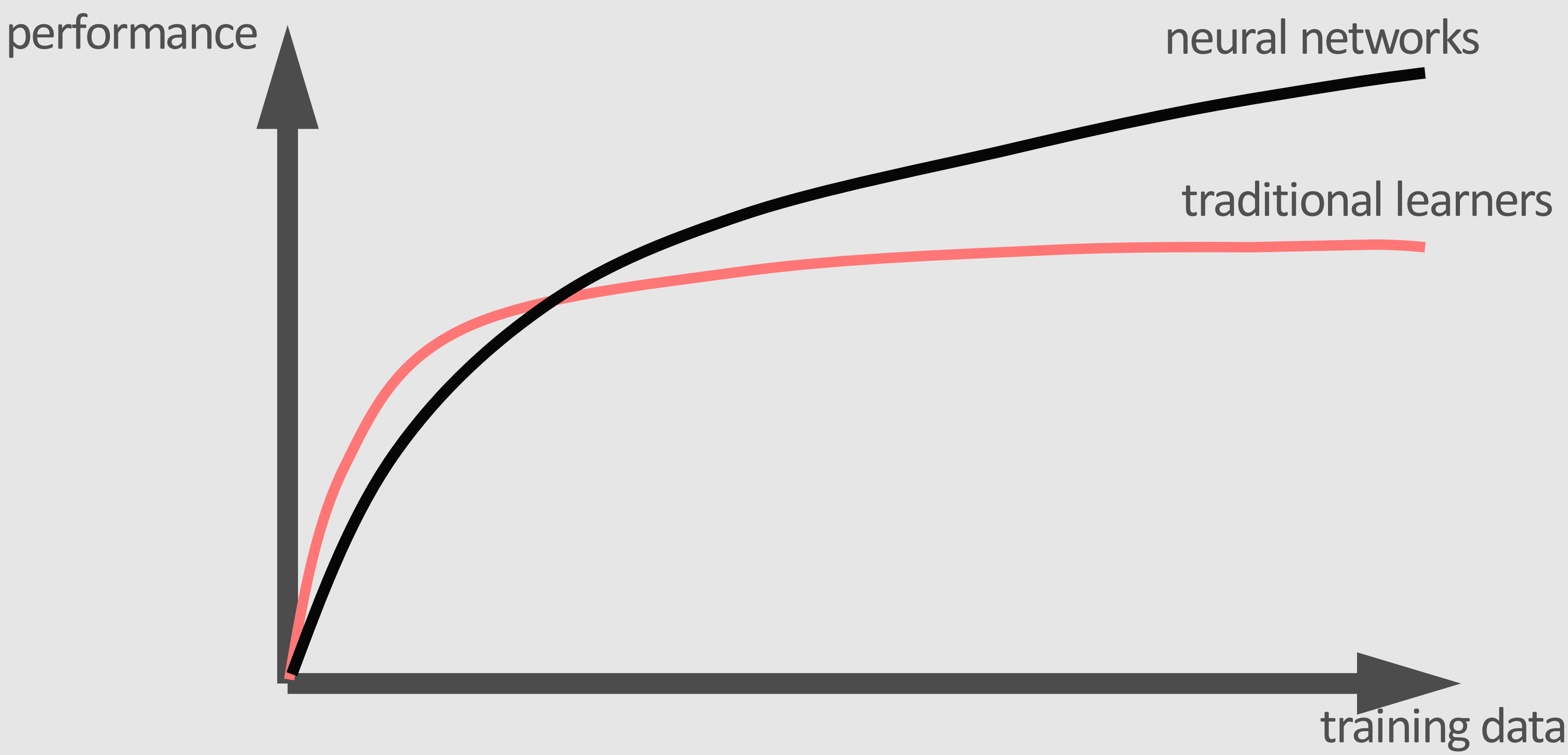
Distributed Tools for the Statistician
Experimentation, Exploration and Inference

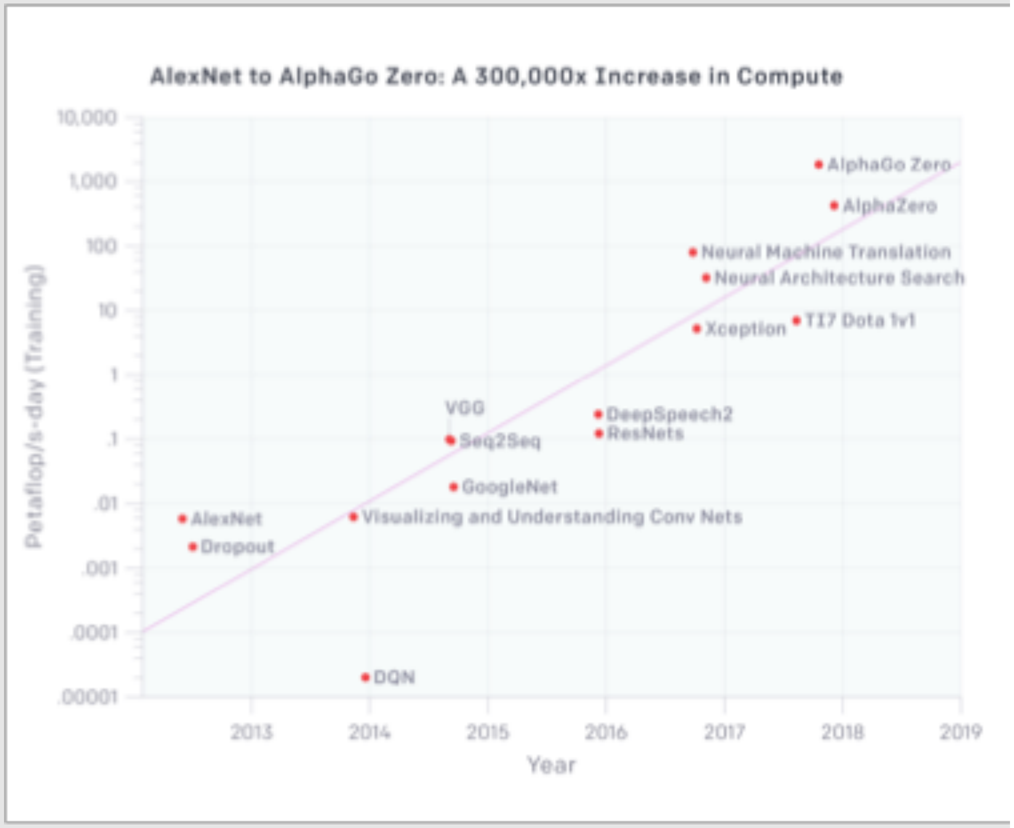
Ali Zaidi
STATS285 F2018

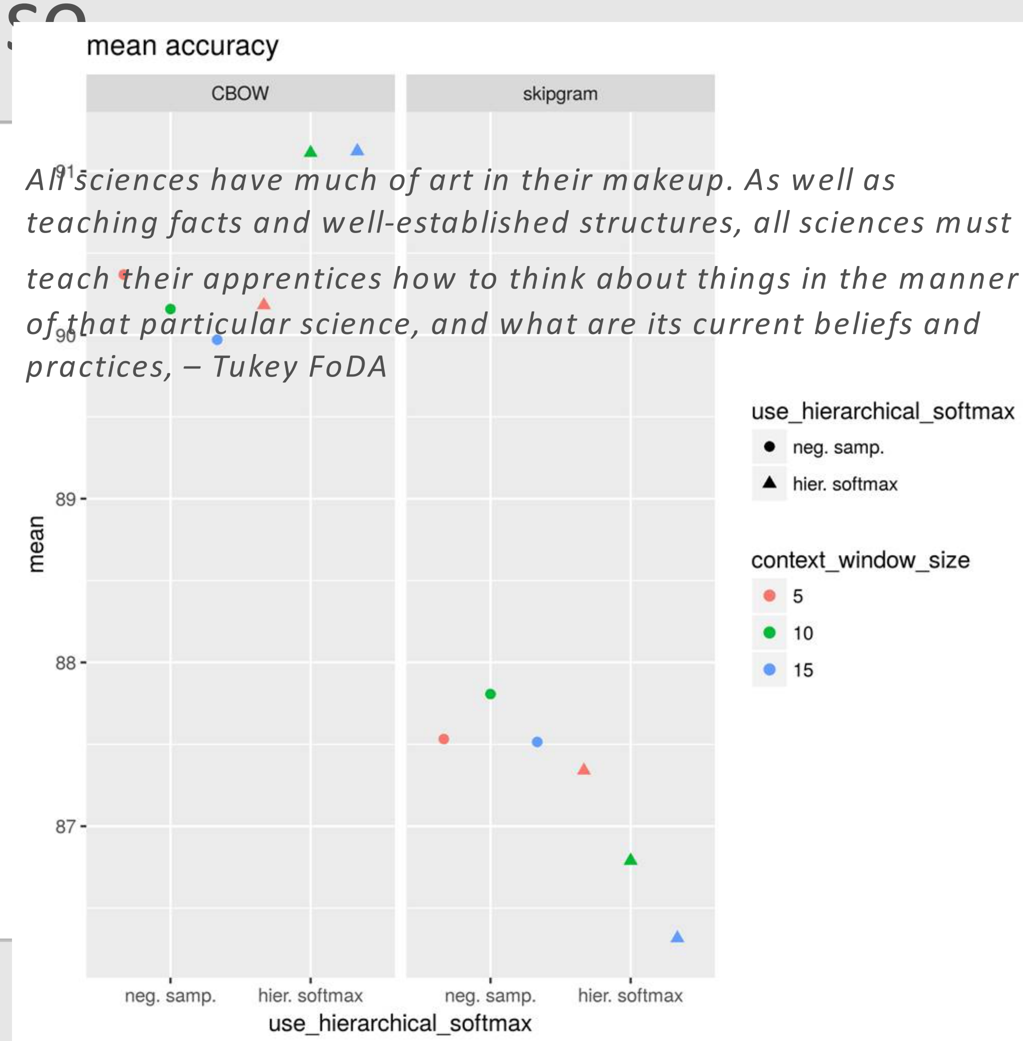
Capacity and Inductive Biases
Learning Curves: Learning from Experience



[Revisiting Unreasonable Effectiveness of Data in Deep Learning Era, Google](#)

AI: Compute and Expertise





All sciences have much of art in their makeup. As well as teaching facts and well-established structures, all sciences must teach their apprentices how to think about things in the manner of that particular science, and what are its current beliefs and practices, – Tukey FoDA

Regularizing and Parameter Sweeping

Improving Distributional Similarity with Lessons Learned from Word Embeddings

Omer Levy Yoav Goldberg Ido Dagan
Computer Science Department
Bar-Ilan University
Ramat-Gan, Israel
{omerlevy,yogo,dagan}@cs.biu.ac.il

- Most of performance gains in word embeddings are due to specific choices of hyperparameters
- Differences are mostly local and domain-specific

Regularizing and Optimizing LSTM Language Models

Stephen Merity [1] Nitish Shirish Keskar [1] Richard Socher [1]

On the State of the Art of Evaluation in Neural Language Models

Gábor Melis[†] Chris Dyer[†] Phil Blunsom[†‡]
[†]DeepMind [‡]University of Oxford
{melisgl,cdyer,pblunsom}@google.com

- Properly regularizing and optimizing LSTM-based models provides SOTA results

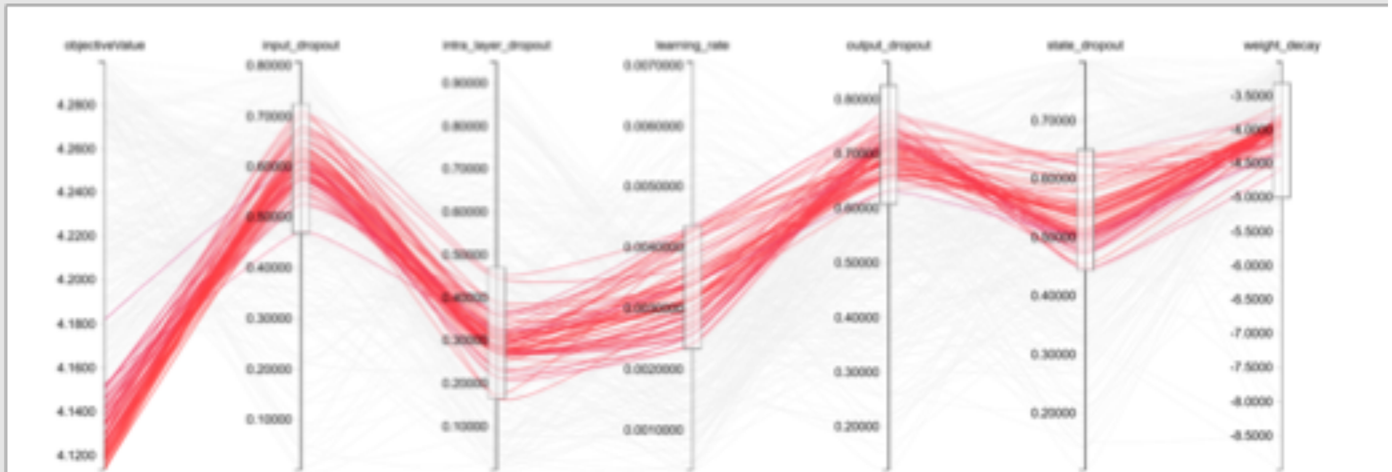


Figure 2: Negative log-likehoods of hyperparameter combinations in the neighbourhood of the best solution for a 4-layer LSTM with 24M weights on the Penn Treebank dataset.

Deep Learning Success Based on Data Hungry Systems

- Image Recognition
  - ImageNet: 14 million examples
- Machine Translation
  - WMT: Millions of sentence pairs
- Game Playing
  - AlphaGo: tens of millions of frames for Atari
  - AlphaGo Zero: hundreds of millions of self-play games

Training Large Networks

ResNet50
ImageNet
~7.02% error rate
Training time: 5 days

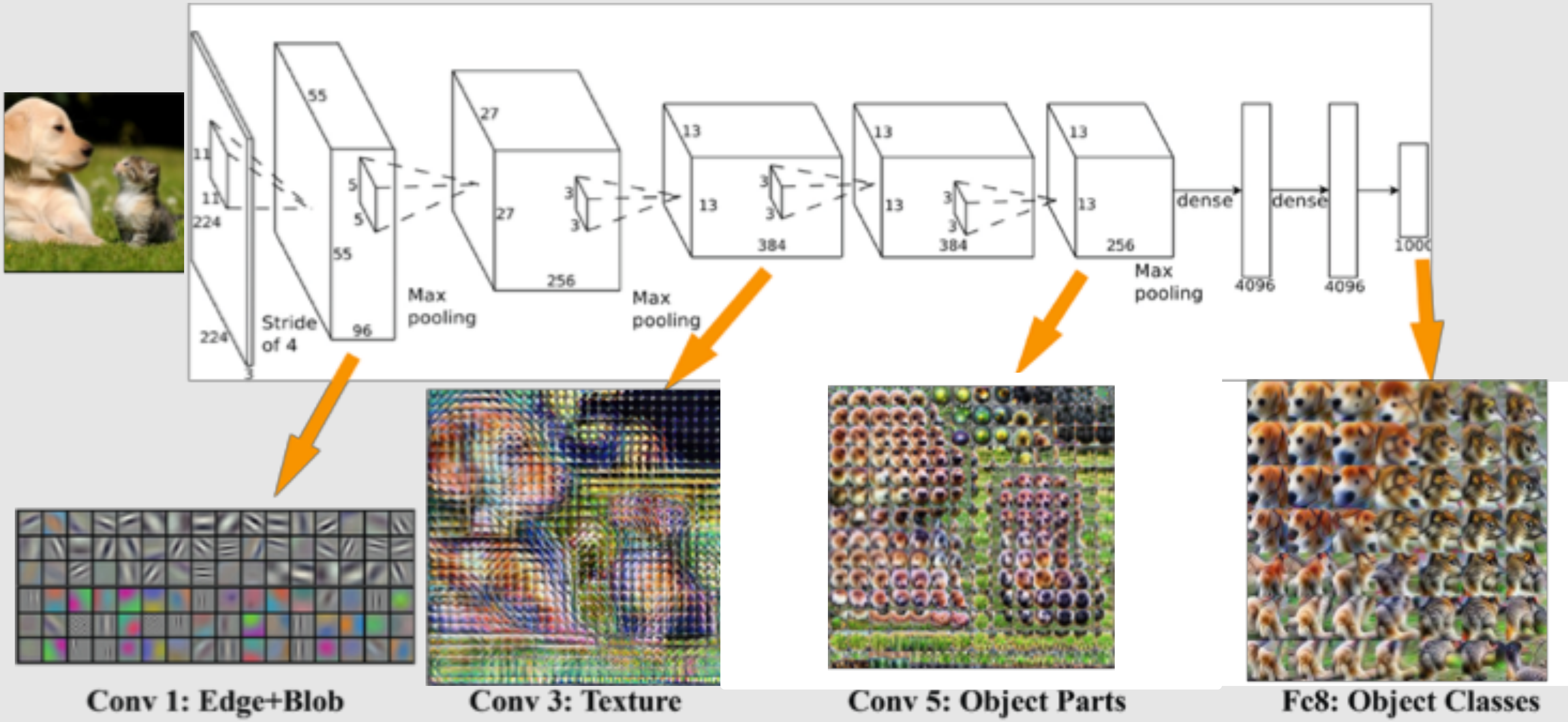
ResNet152
ImageNet
~3.02% error rate
Training time: 1.5 weeks

NMT System
WMT
~11 BLEU
> 2 weeks

https://research.fb.com/wp-content/uploads/2017/06/imagenet1kin1h5.pdf, 1 hour, 256 P100

Learning from Data

- CTF mindset in machine learning
  - Led to Kaggle mentality: optimization of empirical performance
  - Perhaps some overzealous claims about our ability to model complex systems and understand their behavior
  - Inferences from the particular to the general still far away in many pressing applications
- Division between data models and algorithmic models
  - Breiman defined two outlooks for extracting value from data:
    - Prediction: To be able to predict what the responses are going to be to future input variables [algorithmic models]; validation by predictive accuracy
    - Inference: To infer how nature is associating the response variables to the input variables [data models / generative?, Breiman actually labeled inference as information], validation using goodness-of-fit, information criteria, etc.



CV: large datasets (ImageNet), good inductive biases (Convolutions + Classification), pre-training / generative modeling + transfer learning has become the norm

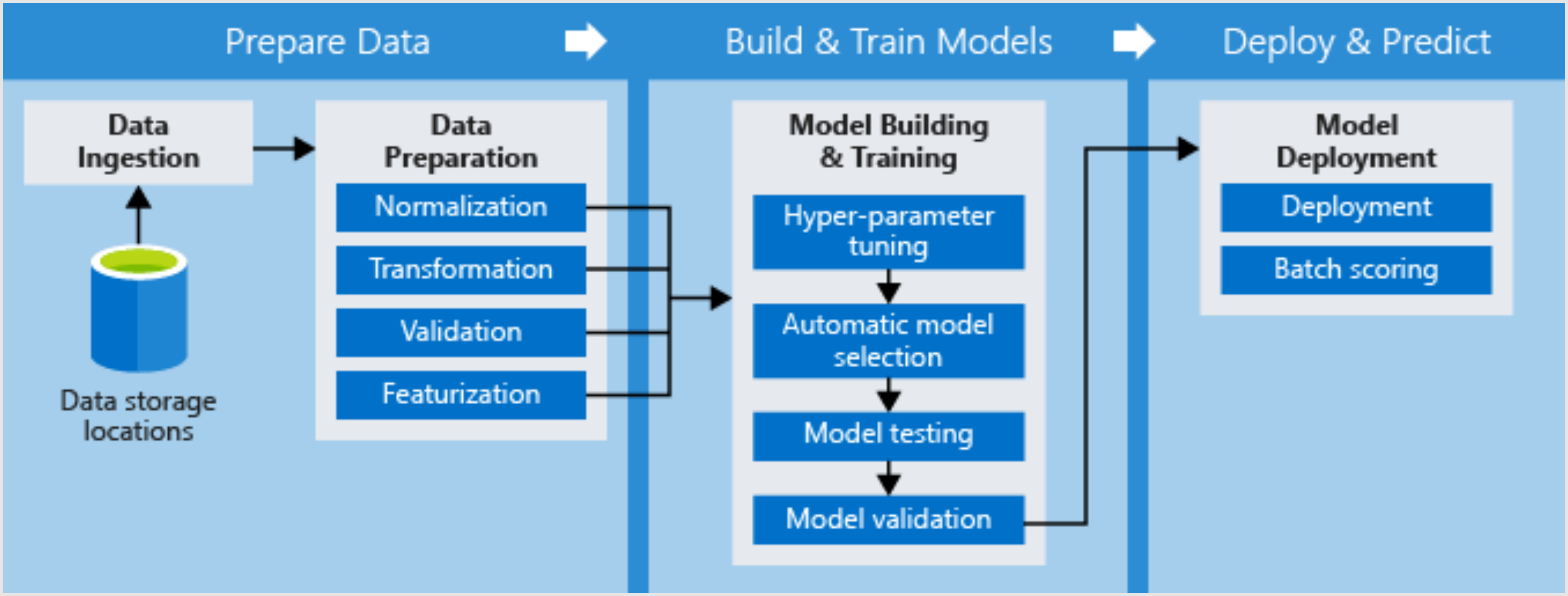
NLP: smaller datasets, unsure about the canonical task, stuck at word embeddings, end-to-end models for each task

`https://github.com/tensorflow/lucid`

Not so Great Data Science Divisions and Goals

- Data Gathering, Preparation, and Exploration
  - Acquisition and labeling [AML DataPrep SDK]
- Processing, Labeling, and Representation Learning
  - Active learning for strategic labeling of data
  - Generative models + feature stores for transfer learning
  - The promise of automl / neural model search [hyperdrive]
- Productionalization / Operationalization
  - Containers and services
- Monitoring and Debugging
  - Error analysis
  - Bias and interpretability

Azure ML Pipelines



Data Preparation

- Data Prep SDK:
  - `pip install --upgrade azureml-dataprep`
- Python based library for defining data processing steps similar to dplyr / pandas [Dataflow]
  - Define entire data processing workflow into a package
  - Package can be served efficiently, scaled across using Spark, Databricks, AKS
  - Pre-defined “intelligent” transforms: example-based transforms, imputation, fuzzy groups and joins, etc.
- Transform stores: once you define a transform once, can re-use it anywhere else

Data Prep SDK Example

```
import pandas as pd
import azureml.dataprep as dprep
dataset_root = "https://dprepdata.blob.core.windows.net/demo"
package_path = path.join(mkdtemp(), "new_york_taxi.dprep")
green_path = "/".join([dataset_root, "green-small/*"])
yellow_path = "/".join([dataset_root, "yellow-small/*"])
# read functions similar to pandas / readr
green_df = dprep.read_csv(path=green_path,
      header=dprep.PromoteHeadersMode.GROUPED)
yellow_df = dprep.smart_read_file(path=yellow_path)
```

Example-based Transforms

```
combined_df = (combined_df
      .derive_column_by_example(source_columns="pickup_date",
                                new_column_name="pickup_weekday",
                                example_data=[("2009-01-04", "Sunday"),
                                              ("2013-08-22", "Thursday")]
)
      .derive_column_by_example(source_columns="dropoff_date",
                                new_column_name="dropoff_weekday",
                                example_data=[("2013-08-22", "Thursday"),
                                              ("2013-11-03", "Sunday")])
```

Save Dataflow to Package

```
combined_df = combined_df.set_name(name="nyc_taxi")
package = dprep.Package(arg=combined_df)
package = package.save(file_path=package_path)
```

Serve on Spark cluster with data on HDFS / blob:

```
package = dprep.Package.open(package_path)
df = package.dataflows[0]
yel_step = df.get_steps()[7].
                  arguments['otherActivities'][0]['anonymousSteps'][0]
yel_step['arguments']['path']['resourceDetails'][0]['path'] = yellow_blob
green_dsource = dprep.BlobDataSource(green_blob)
df = df.replace_datasource(green_dsource)
```

AutoML for Model Search with DataPrep Steps

- Can compose DataPrep steps as part of a model tuning pipeline:

```
from azureml.train.automl import AutoMLConfig
simple_example_data_root =
'https://dprepdata.blob.core.windows.net/automl-notebook-data/'
X = dprep.auto_read_file(simple_example_data_root + 'X.csv').skip(1)
y = dprep.read_csv(simple_example_data_root + 'y.csv’).
         .to_long(dprep.ColumnSelector(term='.*', use_regex=True))



automl_settings = { "iteration_timeout_minutes" : 10, "iterations" : 2,
"primary_metric" : 'AUC_weighted', "preprocess" : False,
"n_cross_validations": 3 }
```

AutoML for Model Search with DataPrep Steps

- Pass Data as DataFlow Objects

```
automl_config = AutoMLConfig(task = 'classification', debug_log =
'automl_errors.log', X = X, y = y, **automl_settings)
```

- Run model selection locally:

```
local_run = experiment.submit(automl_config, show_output = True)
```

- Or run remotely:

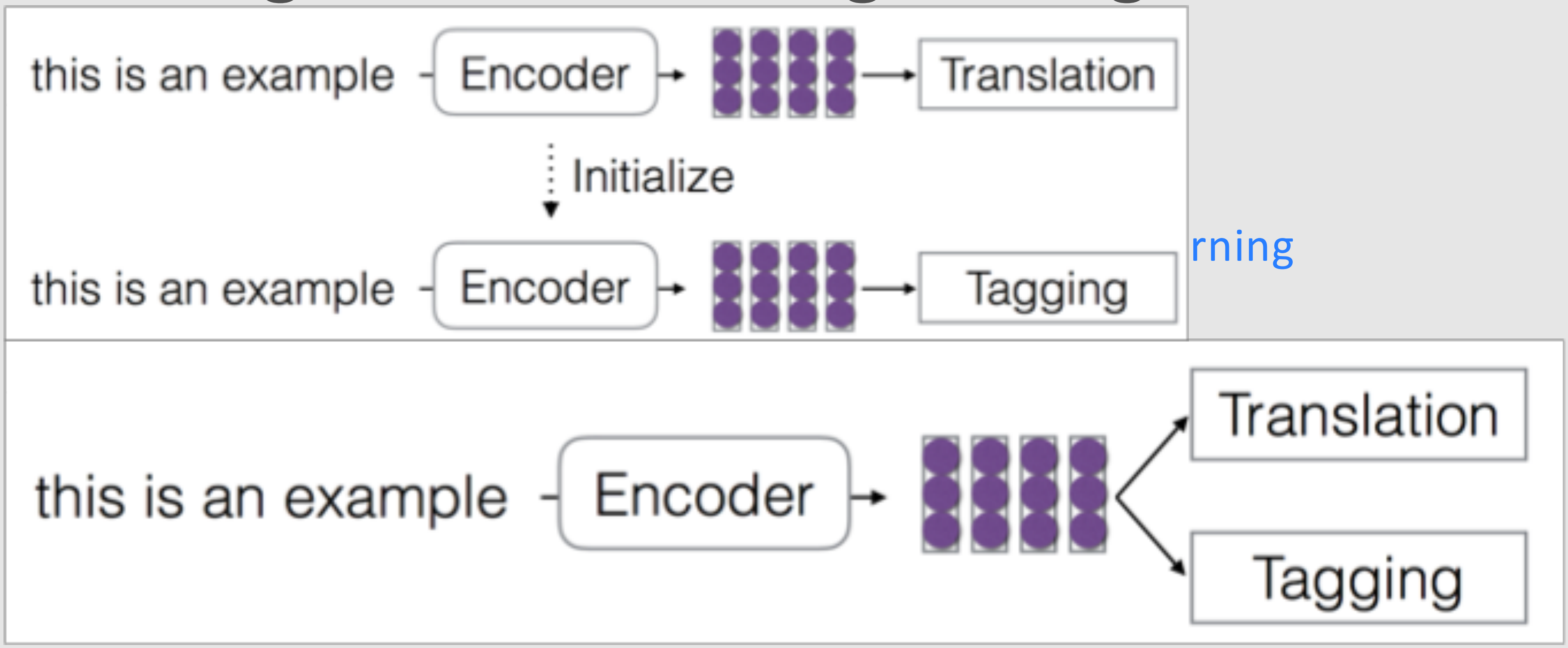
```
dsvm_config = DsvmCompute.provisioning_configuration()

dsvm_compute = DsvmCompute.create(ws, name = dsvm_name,
provisioning_configuration = dsvm_config)

conda_run_config.target = dsvm_compute
automl_config = AutoMLConfig('classification', debug_log =
'automl_errors.log', X = X, y = y, run_configuration=conda_run_config,
**automl_settings )
```

AzureML Notebook Examples

Convergence of Modeling Paradigms



rning

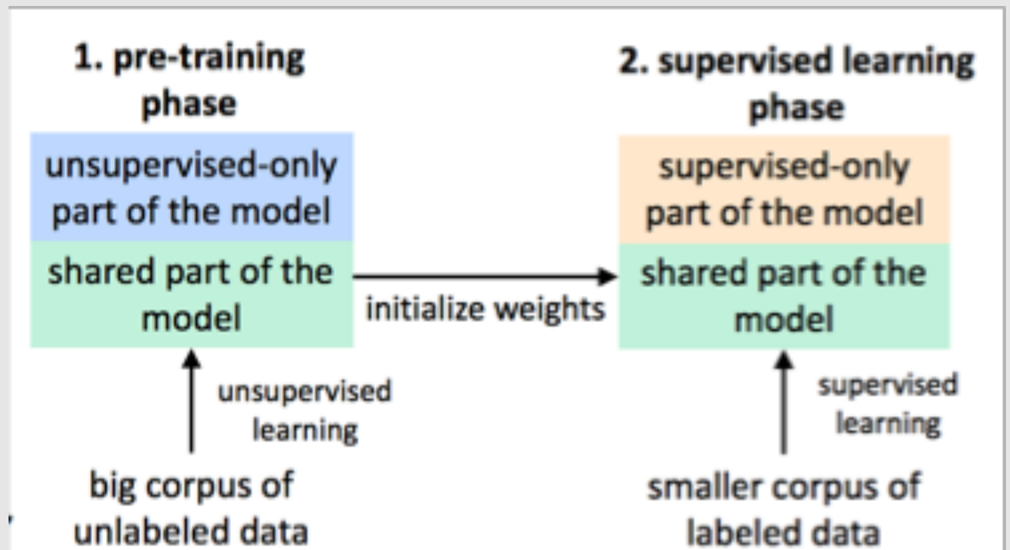
- Hyperparameter sweeping on Batch / Batch AI
  - Data Zoo, Model Zoo, shared experiments (including failures!)

Semi-Supervised Learning

- Pre-training
  - So far, only pre-training initial word vectors (word2vec, GloVe, fasttext)
- Self-training
  - Use unlabeled data
  - Label the confident ones, repeat
- Consistency Regularization
  - Recent idea to adversarially learn shared representations
  - Highly effective for multi-lingual learning

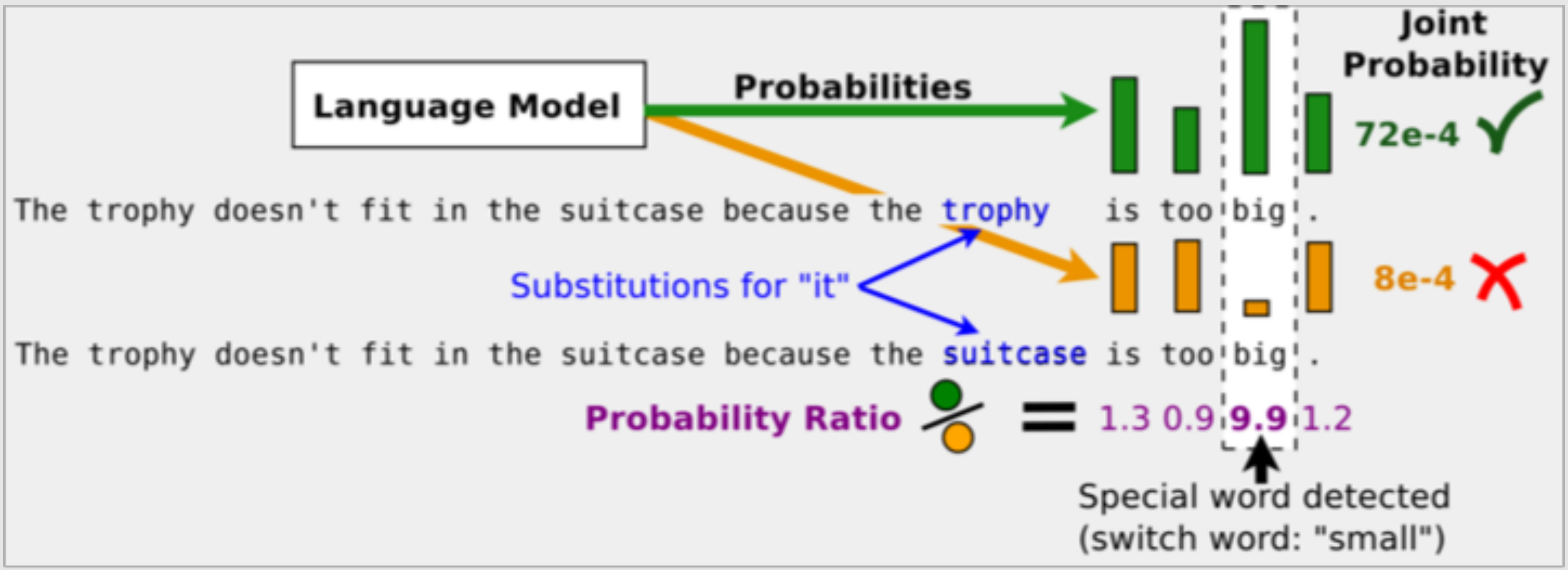
Pre-Training

- First train an unsupervised model on unlabeled data
- Then incorporate the model's learned weights into a supervised model and train it on the labeled data
  - Fixed, or fine-tune
  - Good initialization
  - More meaningful representations



Generative Pre-Training for NLP

- [OpenAI: Improving Language Understanding by Generative Pre-Training]
- [fastAI: Universal Language Model Fine-tuning for Text Classification]
- [Trieu H. Trinh & Quoc Le: A Simple Method for Commonsense Reasoning]



Thanks!