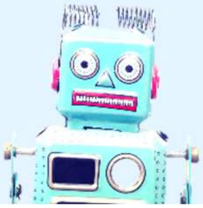


Dive Into

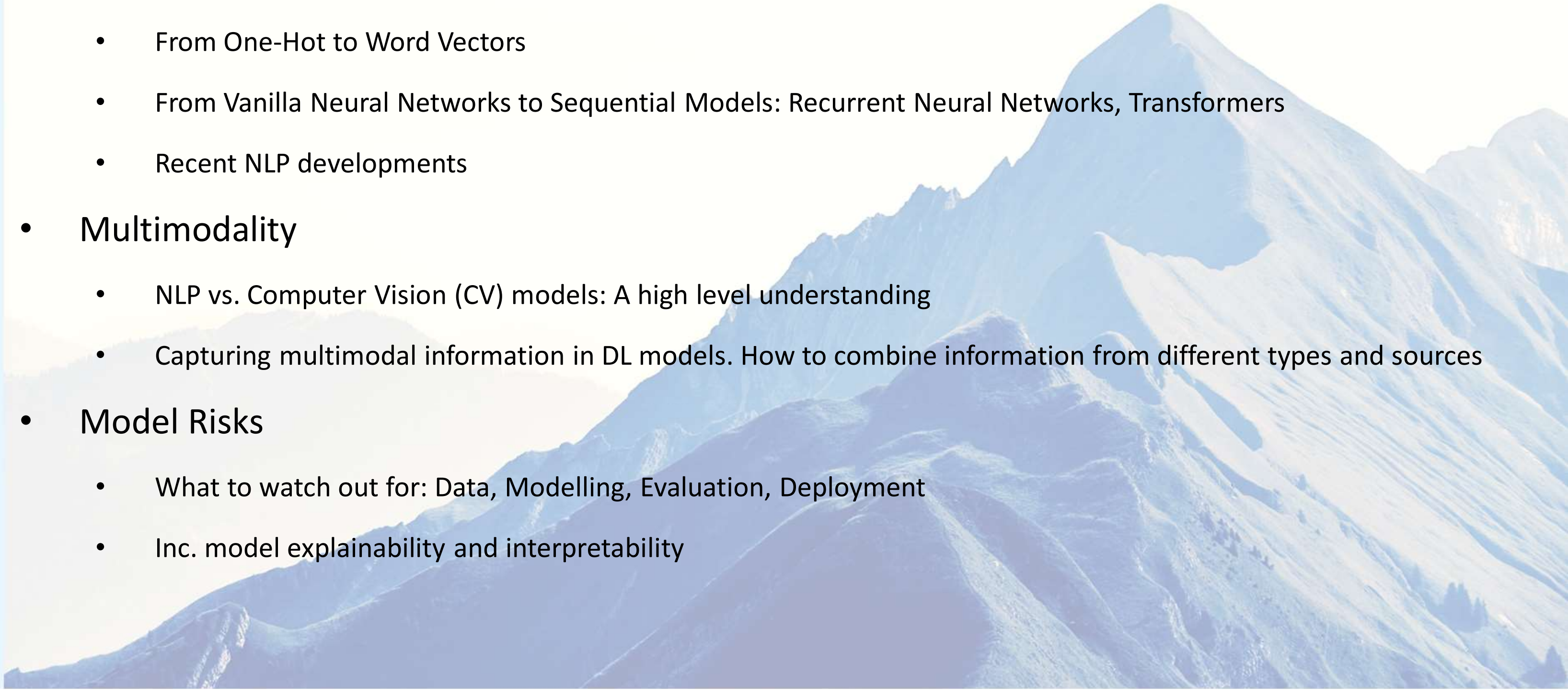
Machine and Deep Learning II

For Multimodal and Unstructured Data + Model Risks

Gary Ang

Overview

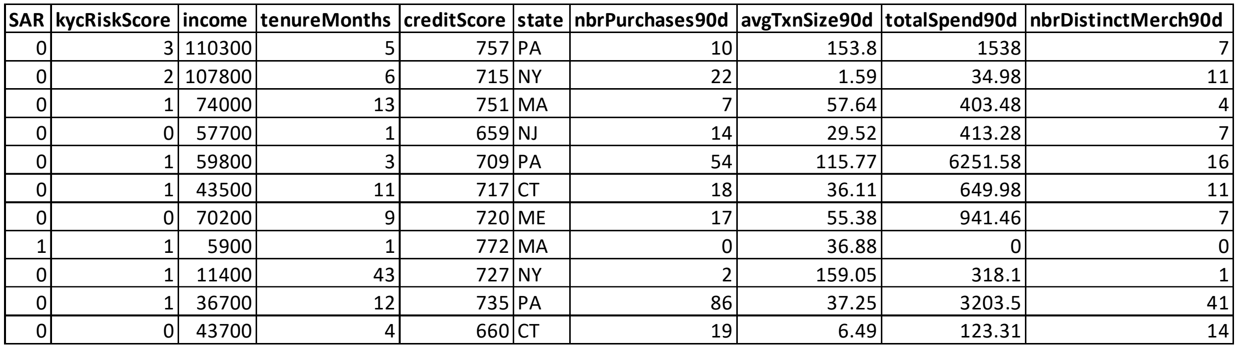
- From tabular and network data to text data
  - Recap
  - Overview of approaches for different types of data
- Introduction to Natural Language Processing (NLP)
  - Text Pre-Processing
  - What are ‘latent embeddings’ again?
  - From One-Hot to Word Vectors
  - From Vanilla Neural Networks to Sequential Models: Recurrent Neural Networks, Transformers
  - Recent NLP developments
- Multimodality
  - NLP vs. Computer Vision (CV) models: A high level understanding
  - Capturing multimodal information in DL models. How to combine information from different types and sources
- Model Risks
  - What to watch out for: Data, Modelling, Evaluation, Deployment
  - Inc. model explainability and interpretability



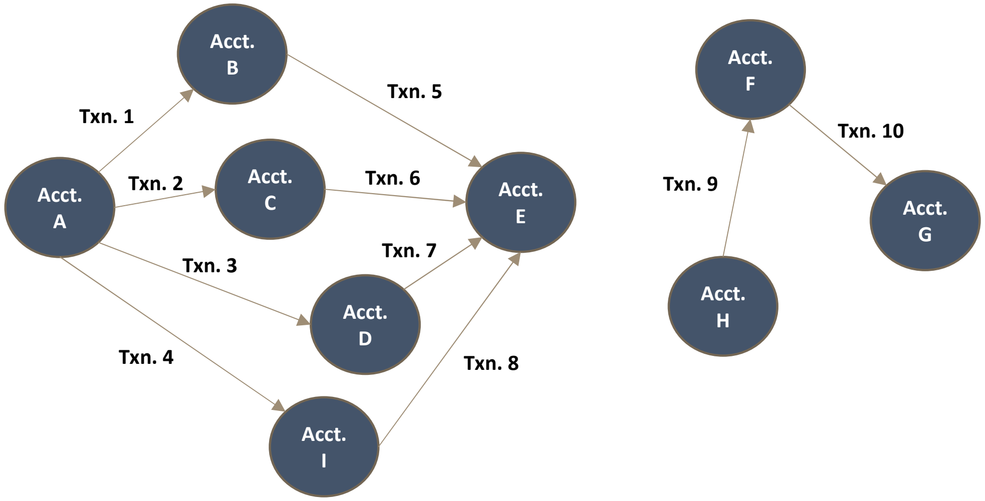
What we will focus on

- Intuition
- Mental models
- Patterns
- Concepts

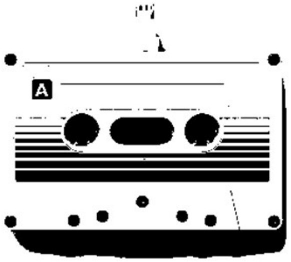
How did we combine tabular and network data?



SAR	kycRiskScore	income	tenureMonths	creditScore	state	nbrPurchases90d	avgTxnSize90d	totalSpend90d	nbrDistinctMerch90d
0	3	110300	5	757	PA	10	153.8	1538	7
0	2	107800	6	715	NY	22	1.59	34.98	11
0	1	74000	13	751	MA	7	57.64	403.48	4
0	0	57700	1	659	NJ	14	29.52	413.28	7
0	1	59800	3	709	PA	54	115.77	6251.58	16
0	1	43500	11	717	CT	18	36.11	649.98	11
0	0	70200	9	720	ME	17	55.38	941.46	7
1	1	5900	1	772	MA	0	36.88	0	0
0	1	11400	43	727	NY	2	159.05	318.1	1
0	1	36700	12	735	PA	86	37.25	3203.5	41
0	0	43700	4	660	CT	19	6.49	123.31	14



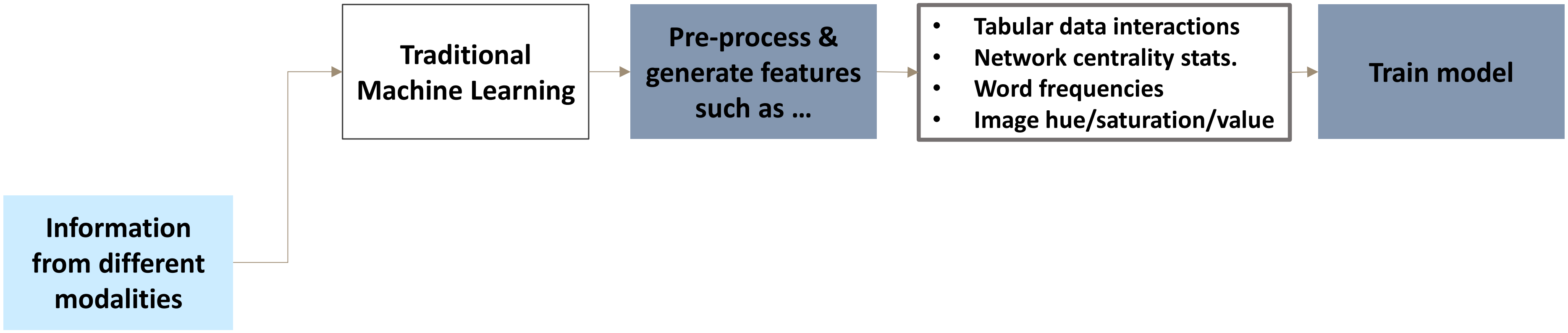
What if we have visual, text, audio data?



We refer to different types of information as information of different **modalities**

More than 1 type of information → multimodality

Two general approaches



Information from different modalities

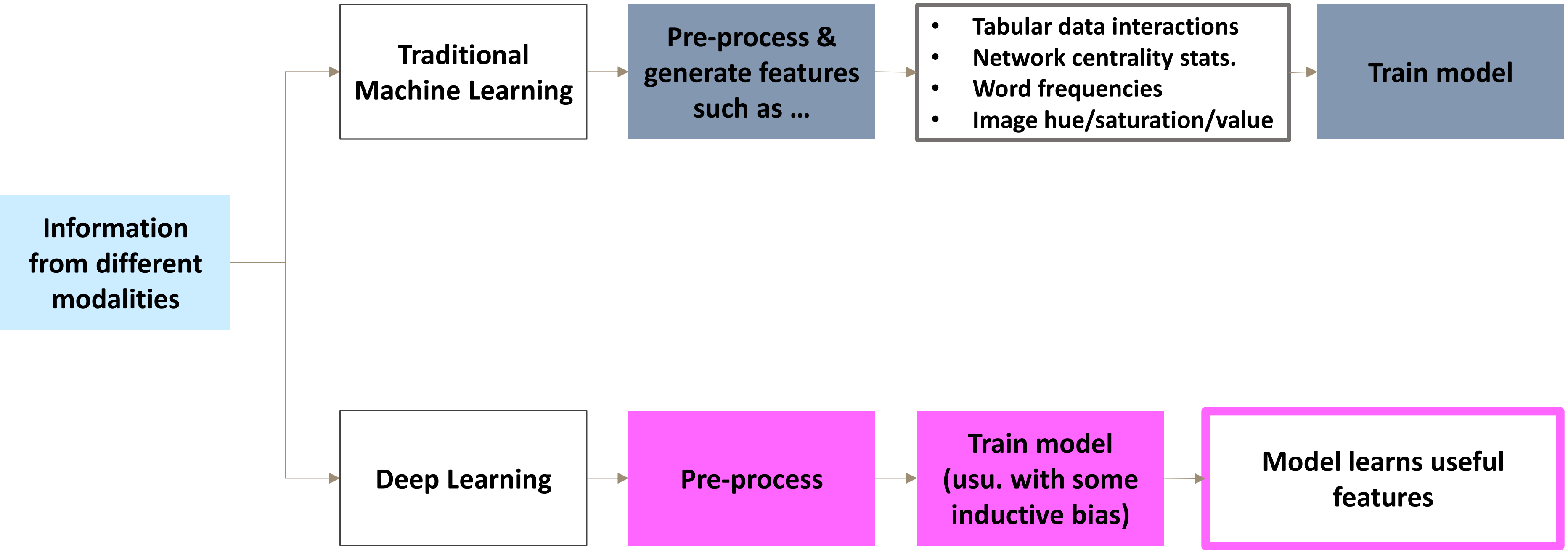
Traditional Machine Learning

Pre-process & generate features such as …

- Tabular data interactions
- Network centrality stats.
- Word frequencies
- Image hue/saturation/value

Train model

Two general approaches



Information from different modalities

Traditional Machine Learning

Pre-process & generate features such as …

- Tabular data interactions
- Network centrality stats.
- Word frequencies
- Image hue/saturation/value

Train model

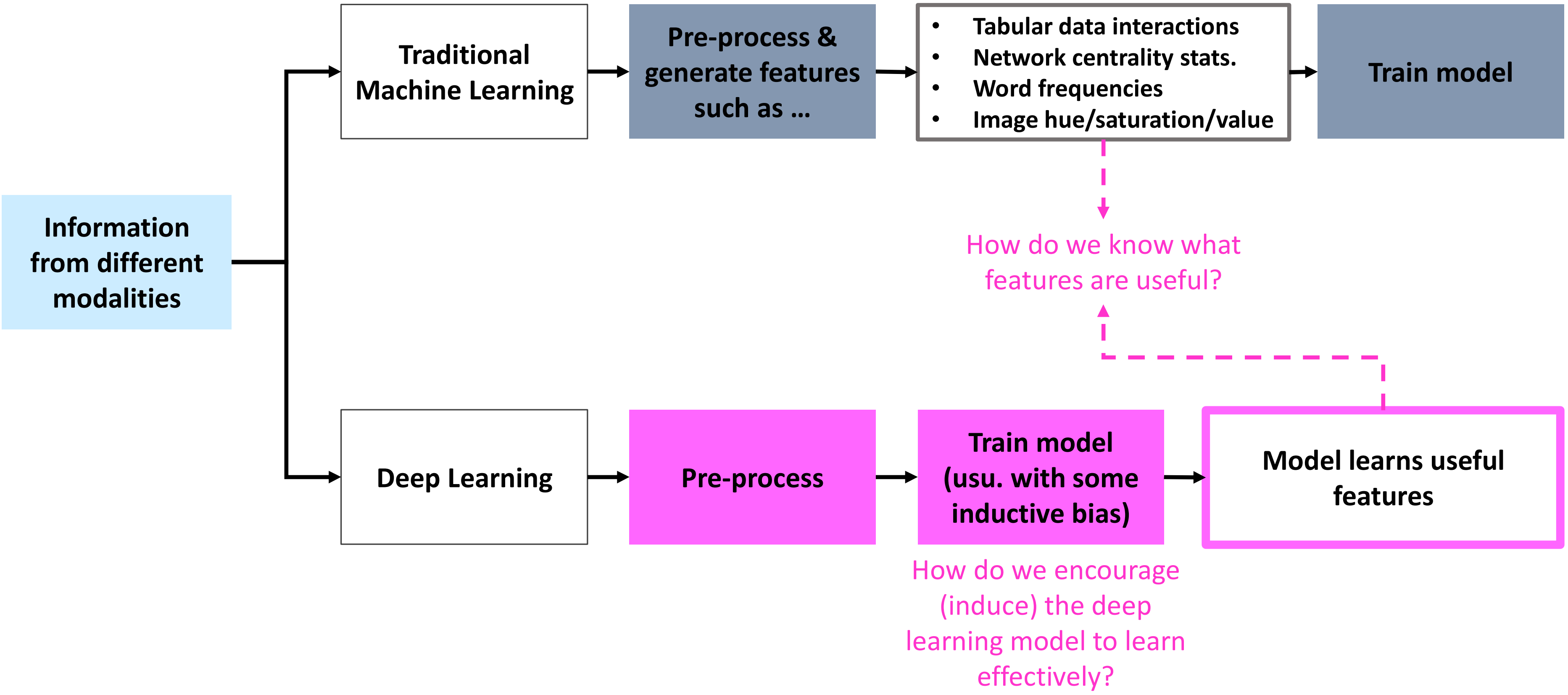
Deep Learning

Pre-process

Train model (usu. with some inductive bias)

Model learns useful features

Let’s discuss!



Information from different modalities

Traditional Machine Learning

Pre-process & generate features such as …

- Tabular data interactions
- Network centrality stats.
- Word frequencies
- Image hue/saturation/value

Train model

How do we know what features are useful?

Deep Learning

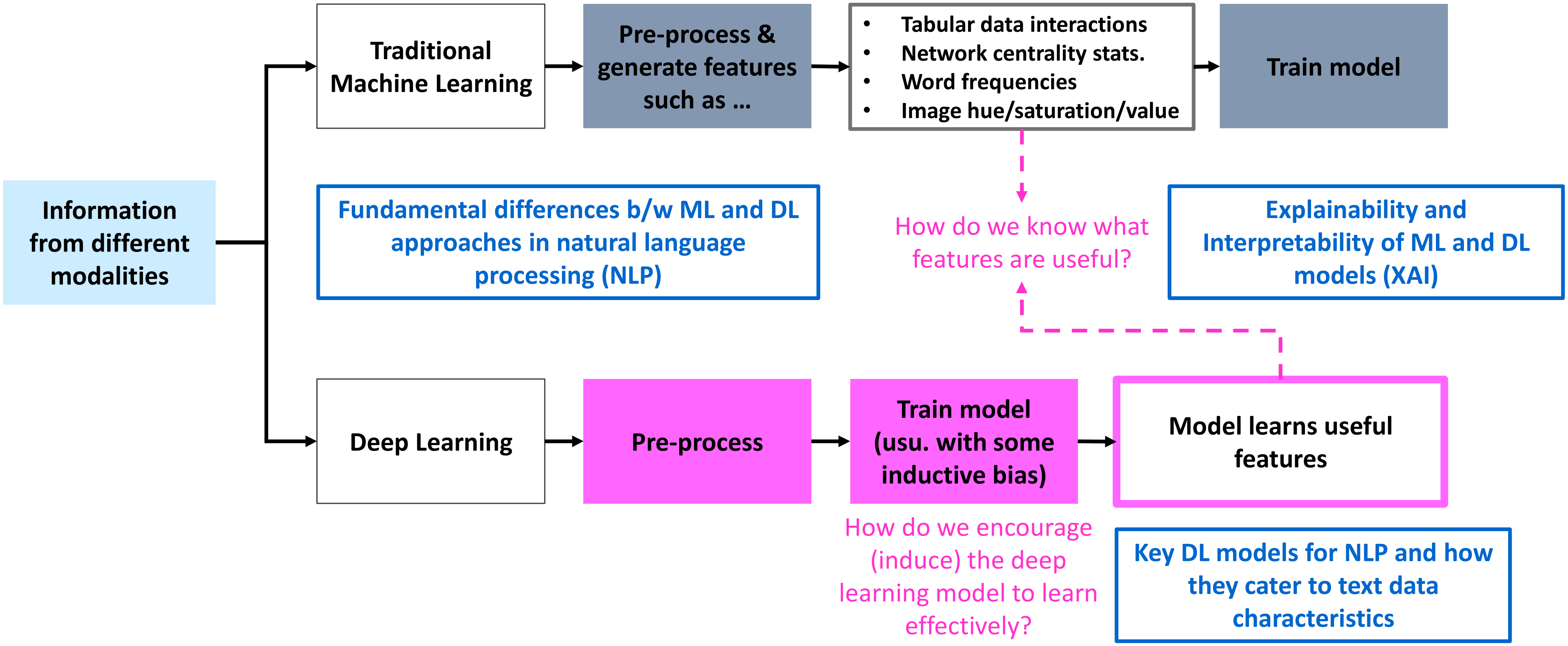
Pre-process

Train model (usu. with some inductive bias)

Model learns useful features

How do we encourage (induce) the deep learning model to learn effectively?

What we will cover today …



Information from different modalities

Traditional Machine Learning

Pre-process & generate features such as …

- Tabular data interactions
- Network centrality stats.
- Word frequencies
- Image hue/saturation/value

Train model

Fundamental differences b/w ML and DL approaches in natural language processing (NLP)

How do we know what features are useful?

Explainability and Interpretability of ML and DL models (XAI)

Deep Learning

Pre-process

Train model (usu. with some inductive bias)

Model learns useful features

How do we encourage (induce) the deep learning model to learn effectively?

Key DL models for NLP and how they cater to text data characteristics

NLP: Key text processing concepts

- Corpus → Document → Sentence → **Token**
- **What are tokens?**
- **Are they words?**

NLP: Key text processing concepts

- **How do you get unique tokens?**
  - Lemmatization: improving → improve
  - Stemming: improving → improv
- **Why do you want unique tokens?**
  - Vocabulary
- **N-grams**
  - ‘*star*’, vs ‘*star wars*’ vs ‘*star power*’ vs. ‘*dancing with the stars*’

What is the simplest way to represent text numerically?

- Say you have these 2 simple sentences (which we treat as 2 documents for simplicity)
  1. *The banks in the island of Singapore were flooded*
  2. *The banks of the river in Singapore were flooded*

Doc	the	banks	in	island	of	Singapore	were	river	flooded
1									
2									

- What's an issue with this simple approach?
  - Are repeated words/tokens more or less important?

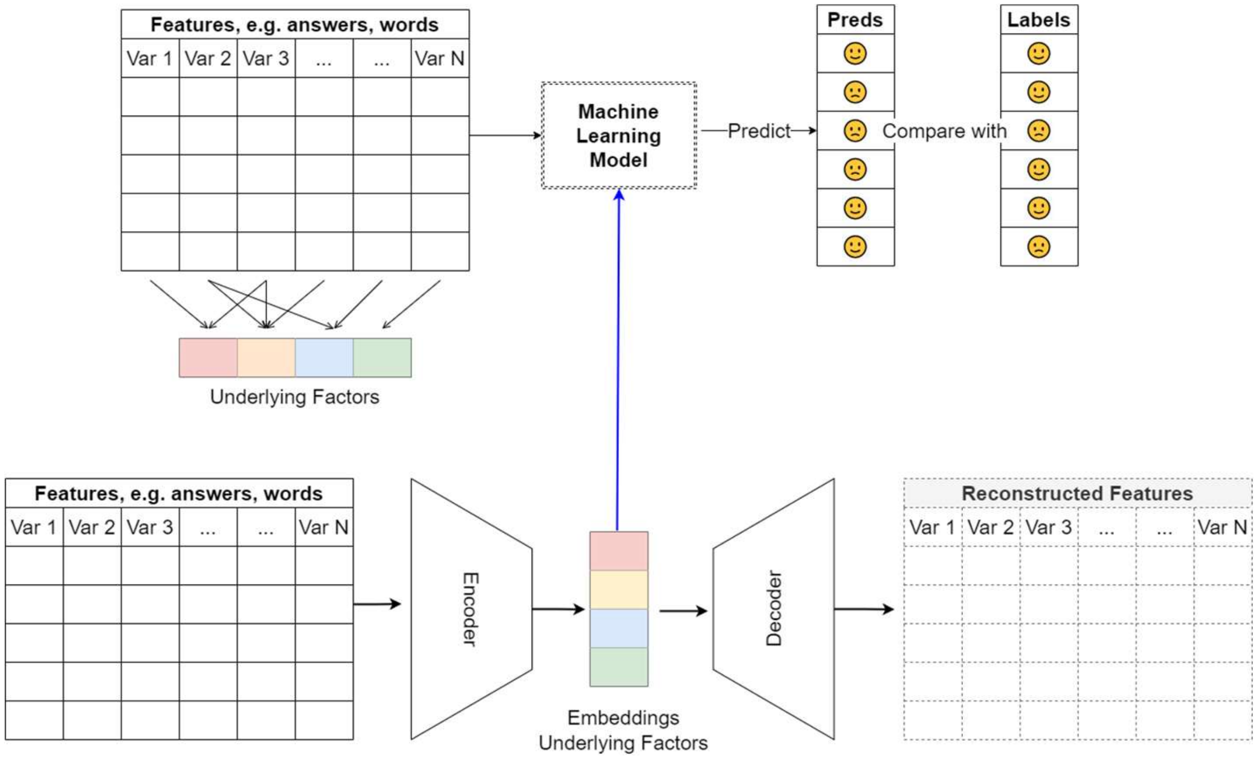
What is slightly smarter way to represent text numerically?

- **Term frequency (TF)** – # of times token in document vs. # tokens in document
- **Inverse document frequency (IDF)** – informativeness of a token
  - Log. of # of documents (N) in corpus / # of times token appears in a document (N_t)
    - Which is larger, rarer? $Log_{10}\ (N/2)$ or $Log_{10}\ (N/100)$
- **TFIDF** – TF × IDF
  - Word that appears many times in a specific document (TF) and is very informative (IDF) → Higher TFIDF score

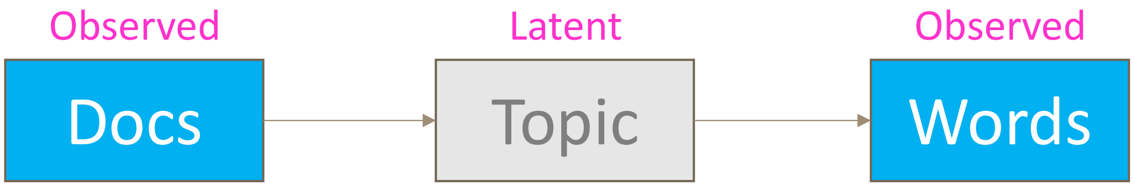
- This still does not solve another issue – think about the token 'bank'

Latent, latent, latent

- We use the word **latent** or **embeddings** a lot in deep learning
  - Can treat as unobserved ***underlying*** factors



Latent Dirichlet Allocation (LDA)



Observed

Docs

Latent

Topic

Observed

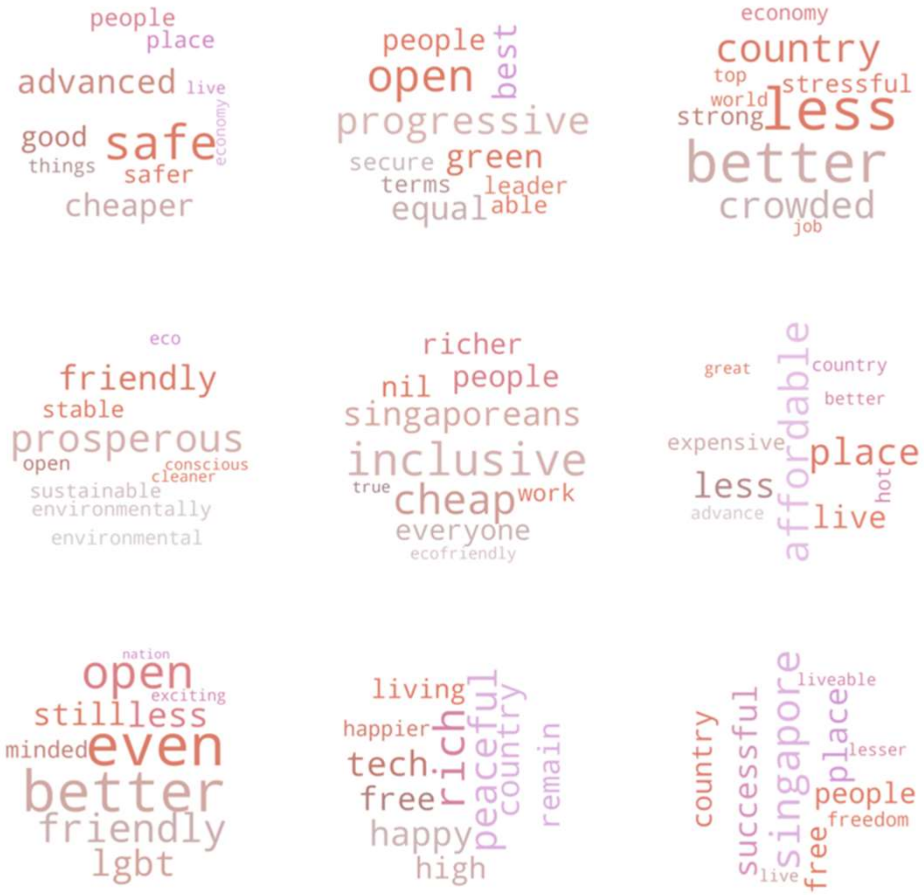
Words

We want to find K topics in the documents

- **Document is mixture of topics**
- **Topics are mixture of words**
- We can learn a model (with some parameters) to find topics
- Not easy to understand, requires some understanding of probabilistic modelling
- For the curious, good primers available here:
  - https://medium.com/@lettier/how-does-lda-work-ill-explain-using-emoji-108abf40fa7d
  - https://towardsdatascience.com/topic-modeling-with-lsa-plsa-lda-nmf-bertopic-top2vec-a-comparison-5e6ce4b1e4a5

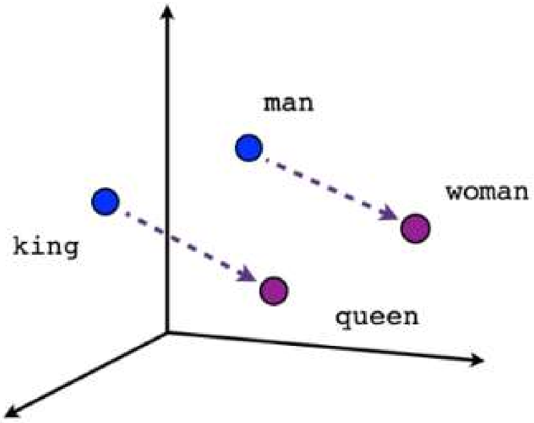
Latent Dirichlet Allocation (LDA)

- An example
  - **What is the K here?**
- See if you can find logical groupings of words
- Unfortunately, number of K (# of latent/underlying topics) requires experimentation
  - What does this remind you of from prev. classes?
- We'll try later

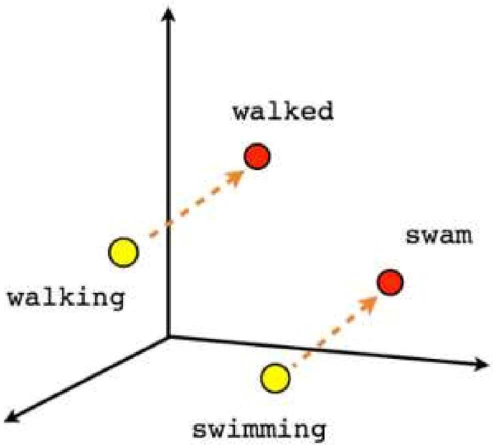


What if we could map words to a set of numbers that is meaningful?

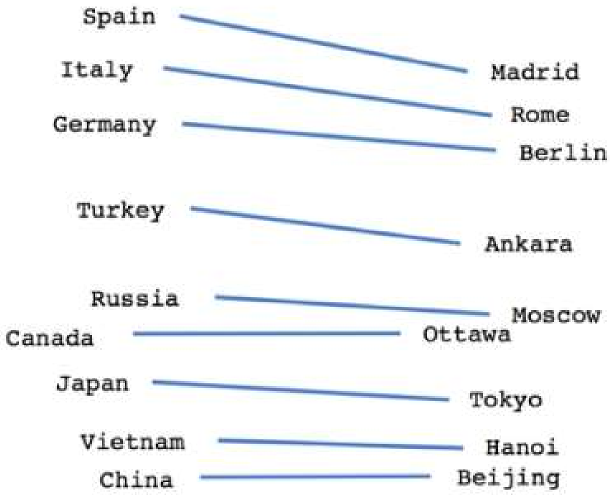
We call these latent or embedding spaces



Male-Female



Verb tense



Country-Capital

https://towardsdatascience.com/creating-word-embeddings-coding-the-word2vec-algorithm-in-python-using-deep-learning-b337d0ba17a8

Word2Vec

- Learns word vectors. How?
  - Basic flow (much more details than this, but simplified here)

Initialize an embedding – think of it as a super shallow neural network where each token has say 3 weights

Token	0	1	2
King	0.126	0.278	0.767
Queen	0.237	0.367	0.234
Man	0.443	0.475	0.453
Woman	0.222	0.376	0.899
…	…	…	…

For more details, can refer to https://jalammar.github.io/illustrated-word2vec/

Word2Vec

- Learns word vectors. How?
  - Basic flow (much more details than this, but simplified here)

Initialize an embedding – think of it as a super shallow neural network where each token has say 3 weights

Token	0	1	2
King	0.126	0.278	0.767
Queen	0.237	0.367	0.234
Man	0.443	0.475	0.453
Woman	0.222	0.376	0.899
…	…	…	…

Get training samples

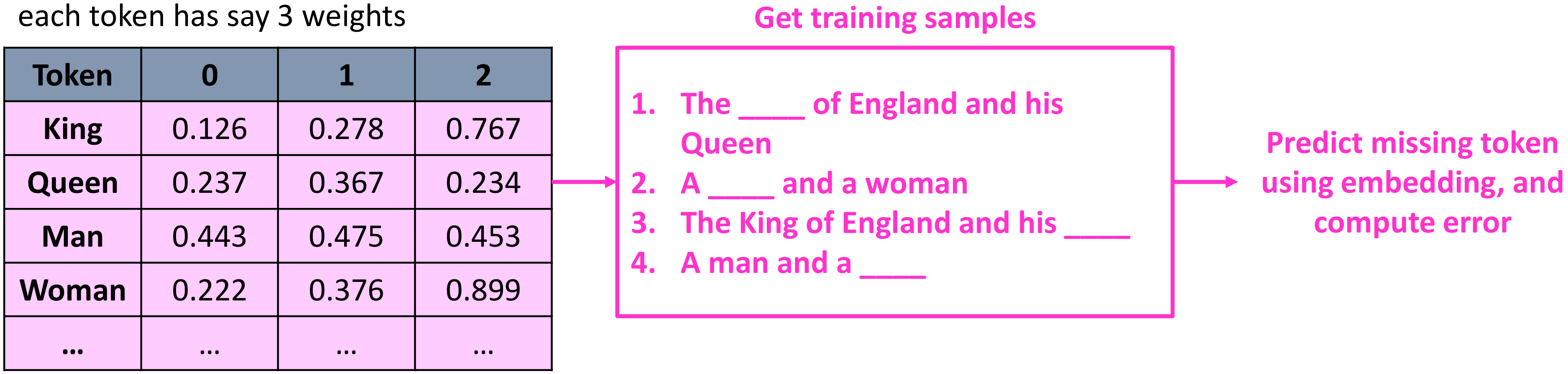
1. The ____ of England and his Queen
2. A ____ and a woman
3. The King of England and his ____
4. A man and a ____

For more details, can refer to https://jalammar.github.io/illustrated-word2vec/

Word2Vec

- Learns word vectors. How?
  - Basic flow (much more details than this, but simplified here)

Initialize an embedding – think of it as a super shallow neural network where each token has say 3 weights



Token	0	1	2
King	0.126	0.278	0.767
Queen	0.237	0.367	0.234
Man	0.443	0.475	0.453
Woman	0.222	0.376	0.899
…	…	…	…

Get training samples

1. The ____ of England and his Queen
2. A ____ and a woman
3. The King of England and his ____
4. A man and a ____

Predict missing token using embedding, and compute error

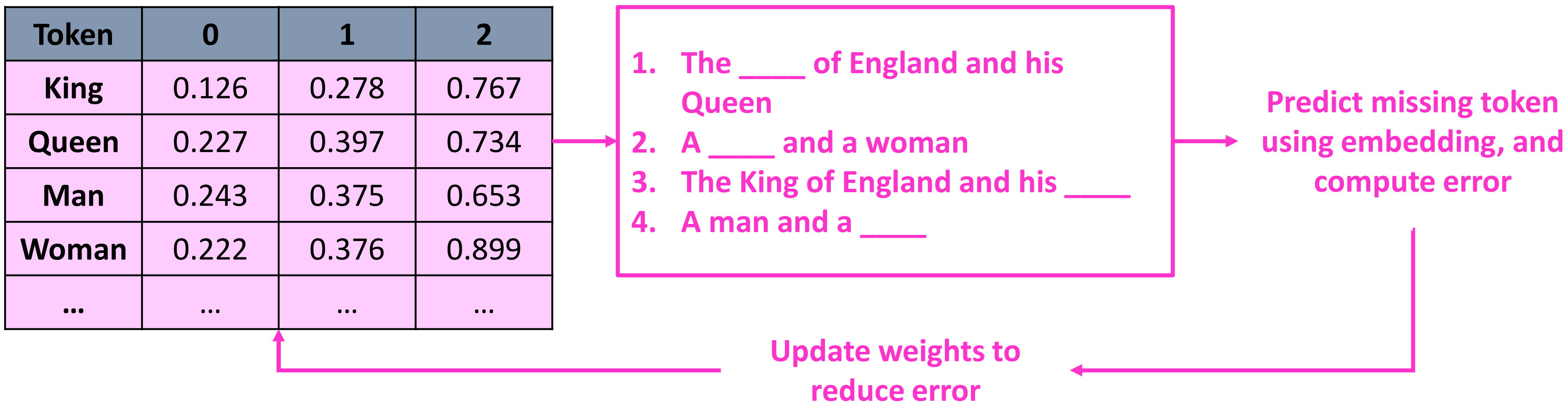
For more details, can refer to https://jalammar.github.io/illustrated-word2vec/

Word2Vec

- Learns word vectors. How?
  - Basic flow (much more details than this, but simplified here)

Initialize an embedding – think of it as a super shallow neural network where each token has say 3 weights

Get training samples



Token	0	1	2
King	0.126	0.278	0.767
Queen	0.227	0.397	0.734
Man	0.243	0.375	0.653
Woman	0.222	0.376	0.899
…	…	…	…

1. The ____ of England and his Queen
2. A ____ and a woman
3. The King of England and his ____
4. A man and a ____

Predict missing token using embedding, and compute error

Update weights to reduce error

For more details, can refer to https://jalammar.github.io/illustrated-word2vec/

Word2Vec

- When training is completed, each word is represented by a vector
- Why is this useful?
  - E.g., for sentiment analysis, i.e., predict whether a sentence is positive or negative in sentiment?

Think about the dimensions of Word2Vec representation vs. a one-hot of TFIDF representation

Token	0	1	2
King	0.146	0.278	0.757
Queen	0.227	0.377	0.834
Man	0.143	0.275	0.753
Woman	0.222	0.376	0.899
…	…	…	…

Think about the semantics (meaning) of a Word2Vec representation vs. a one-hot of TFIDF representation

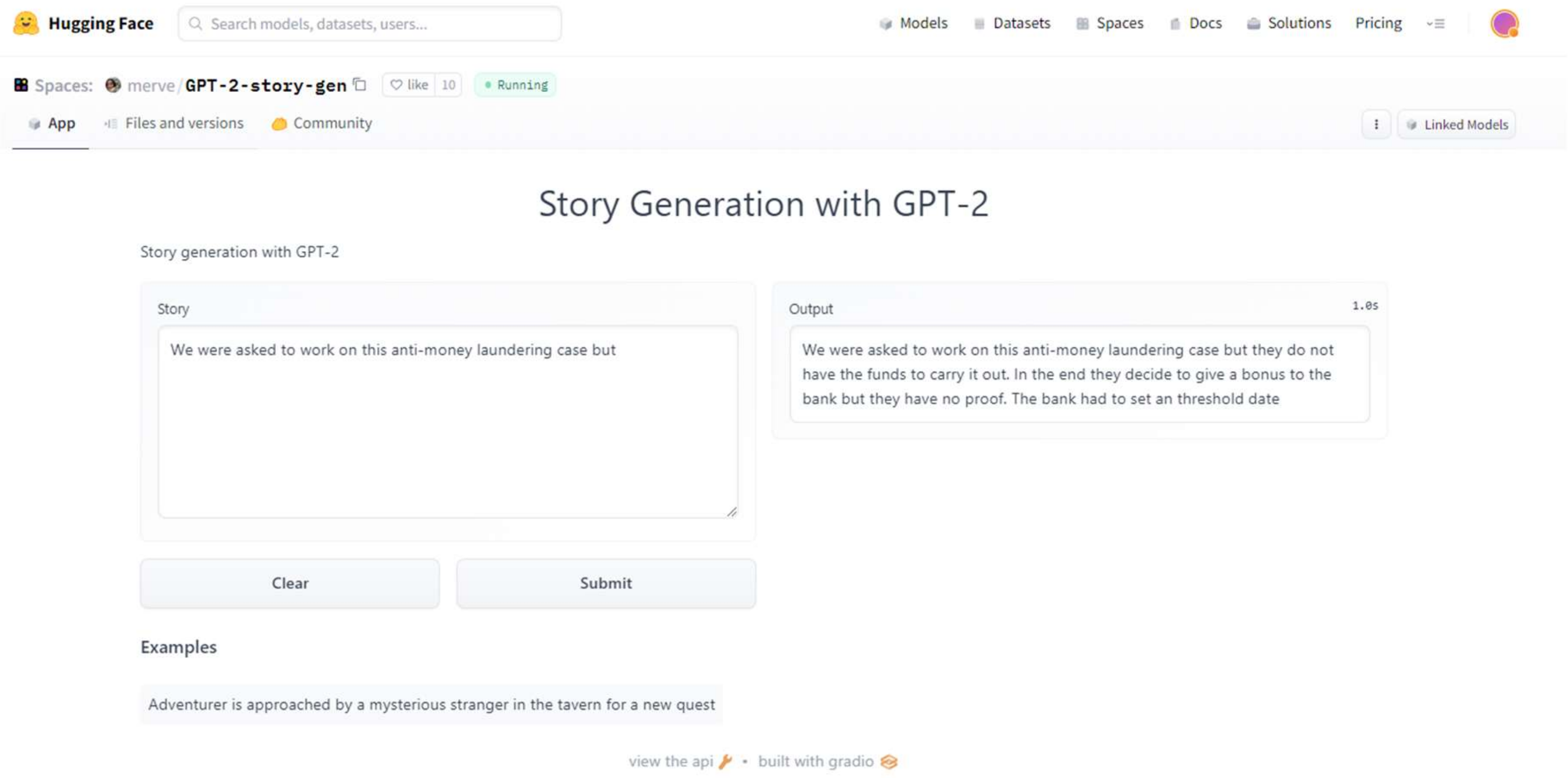
NLP task, e.g., sentiment analysis

From word vectors to neural networks for text

- Word2Vec is **shallow**.
  - → Recall why deeper neural networks can sometimes be better.
- Word2Vec is **context-independent**
  - → Recall the word '*bank*'
- We still need some model to use the embeddings for **different tasks**, e.g., sentiment analysis
  - → Machine learning – Any of the models we learnt can be used – Logistic regression, decision trees, random forest, XGBoost …
    - But basic ML models generally do not consider a very important aspect of text and language. Can you guess one characteristic that ML models usually do not address?
- **More recent deep learning models can usually learn richer semantics**

Demonstration -

- https://huggingface.co/spaces/merve/GPT-2-story-gen



Story Generation with GPT-2

Story generation with GPT-2

Story

We were asked to work on this anti-money laundering case but

Output

1.0s

We were asked to work on this anti-money laundering case but they do not have the funds to carry it out. In the end they decide to give a bonus to the bank but they have no proof. The bank had to set an threshold date

Examples

Adventurer is approached by a mysterious stranger in the tavern for a new quest

Key deep learning models for NLP

- **Recurrent Neural Networks** (RNN)
  - Gated Recurrent Units (GRU)
  - Long Short-Term Memory (LSTM)
    - Been around for a long time, more than 30 years, GRU and LSTM are more modern variants (that are also more than 10 years old)
    - Can be used for any sequences, whether text or time-series

- *More details in a while*

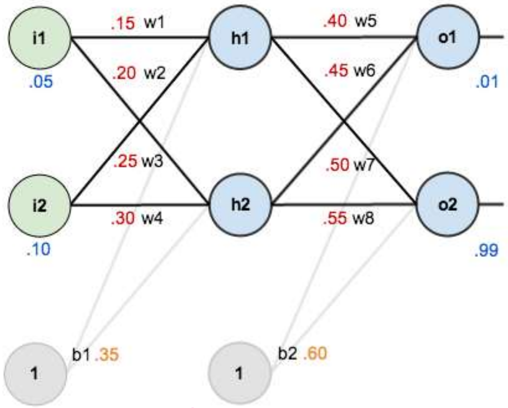
Key deep learning models for NLP

- **Transformers**
  - Based on basic idea of self-attention, proposed in 2017
  - Key driver of many of the interesting NLP models you see today
    - Not restricted to sequences, can be used in other domains such as computer vision and networks/graphs
- *More details in a while*

Form	Loss
Train	Evaluate

First, … Deep Learning 101

- Neural networks are basically layers and layers of neurons (weights)
- Get predictions by passing input data through weights (forward prop.)
- Get error and update weights (backward prop.)



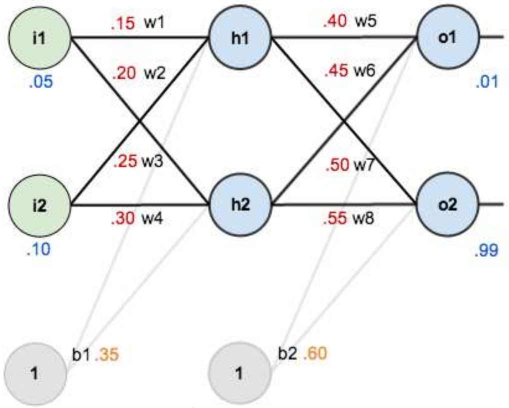
Forward Propagation

For a simple example, see https://mattmazur.com/2015/03/17/a-step-by-step-backpropagation-example/

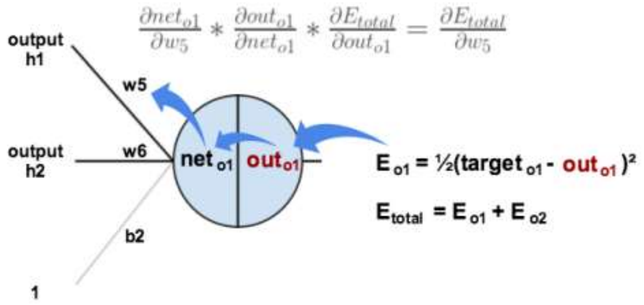
First, … Deep Learning 101

Form	Loss
Train	Evaluate

- Neural networks are basically layers and layers of neurons (weights)
- Get predictions by passing input data through weights (forward prop.)
- Get error and update weights (backward prop.)



Forward Propagation



$$\frac{\partial net_{o1}}{\partial w_5} * \frac{\partial out_{o1}}{\partial net_{o1}} * \frac{\partial E_{total}}{\partial out_{o1}} = \frac{\partial E_{total}}{\partial w_5}$$

$$E_{o1} = \tfrac{1}{2}(target_{o1} - out_{o1})^2$$

$$E_{total} = E_{o1} + E_{o2}$$

$$w_5^+ = w_5 - \eta * \frac{\partial E_{total}}{\partial w_5}$$

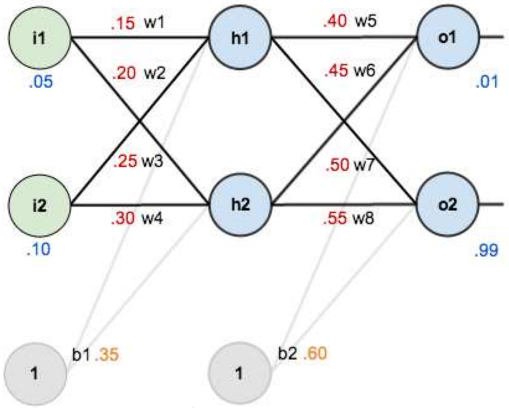
Backward Propagation

For a simple example, see https://mattmazur.com/2015/03/17/a-step-by-step-backpropagation-example/

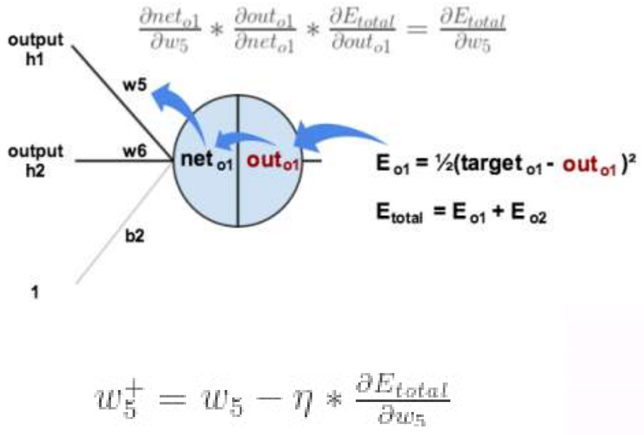
First, … Deep Learning 101

Form	Loss
Train	Evaluate

- Neural networks are basically layers and layers of neurons (weights)
- Get predictions by passing input data through weights (forward prop.)
- Get error and update weights (backward prop.)



Forward Propagation



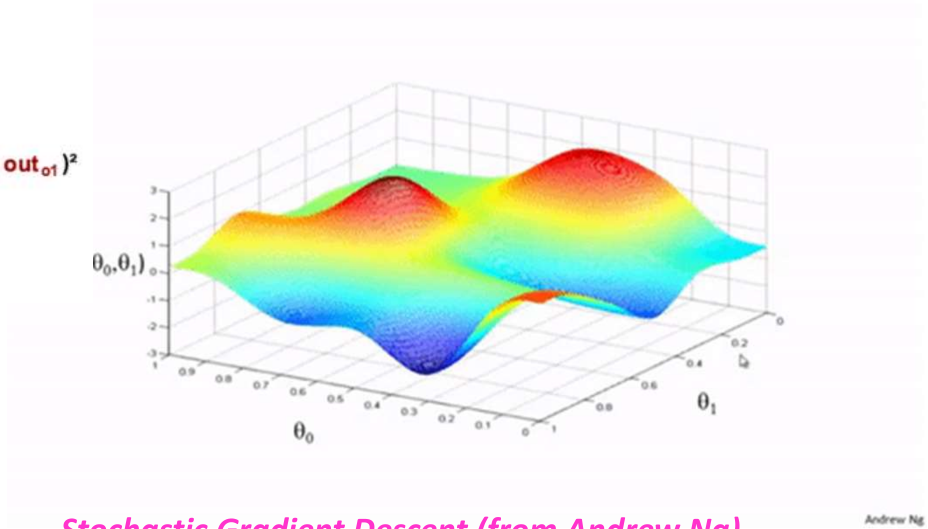
$$\frac{\partial net_{o1}}{\partial w_5} * \frac{\partial out_{o1}}{\partial net_{o1}} * \frac{\partial E_{total}}{\partial out_{o1}} = \frac{\partial E_{total}}{\partial w_5}$$

$$E_{o1} = ½(target_{o1} - out_{o1})^2$$

$$E_{total} = E_{o1} + E_{o2}$$

$$w_5^+ = w_5 - \eta * \frac{\partial E_{total}}{\partial w_5}$$

Backward Propagation

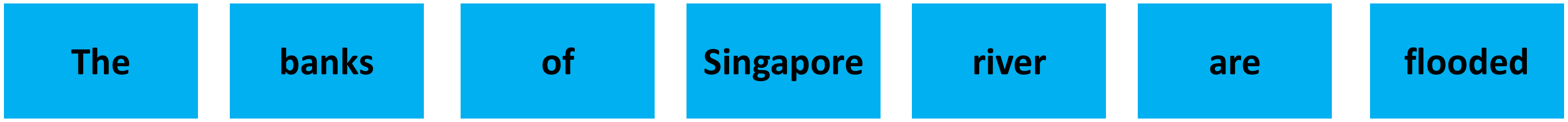


Stochastic Gradient Descent (from Andrew Ng)

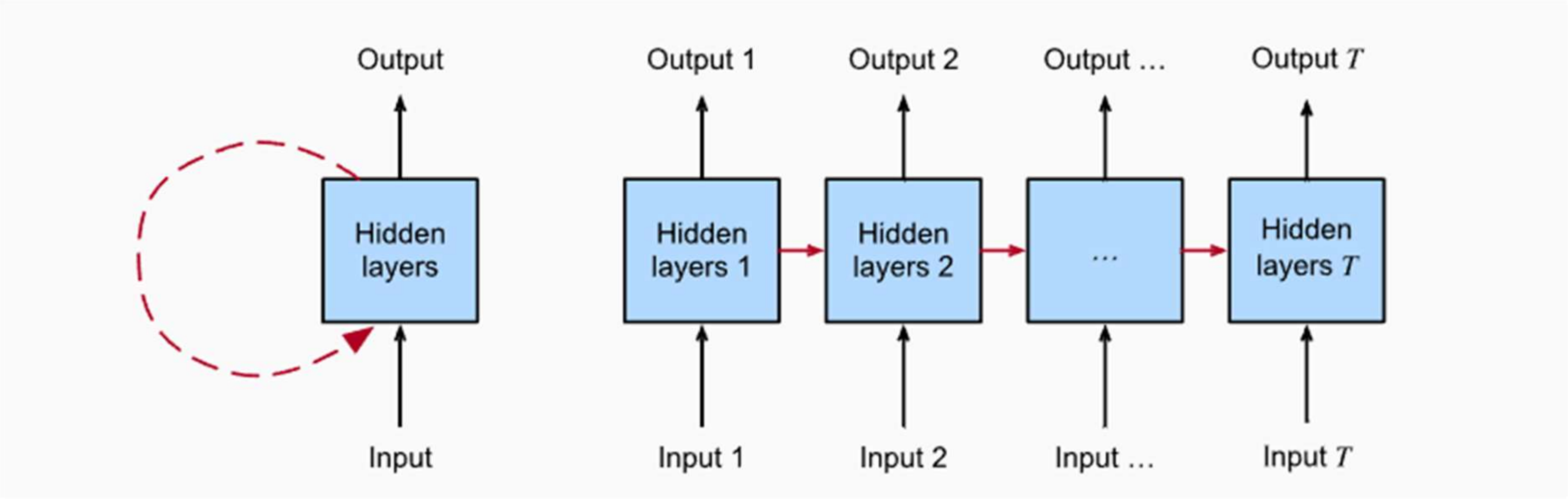
For a simple example, see https://mattmazur.com/2015/03/17/a-step-by-step-backpropagation-example/

Deep Learning for Language

- Recall
  - What are the specific characteristics of text and language data?
    - It is a s _ _ _ _ _ _ _
    - Meaning depends on c _ _ _ _ _ _



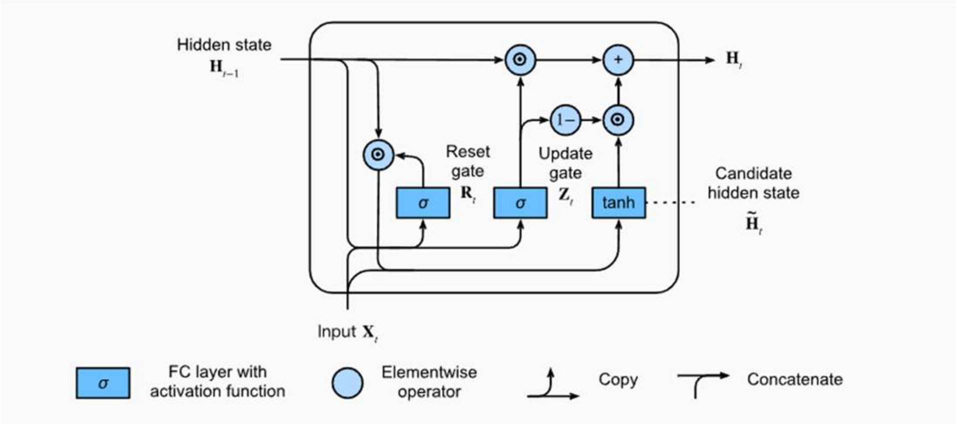
Recurrent Neural Networks

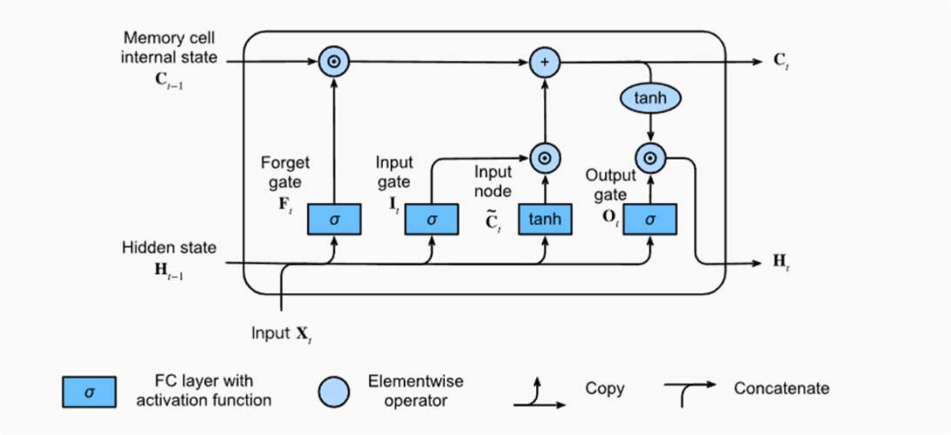


Figures from https://d2l.ai/

RNN Variants - GRU and LSTM

- I know figures look complex, but no need to worry about details
- Basic idea is that we need someway to determine how much of prev. memory to retain and use

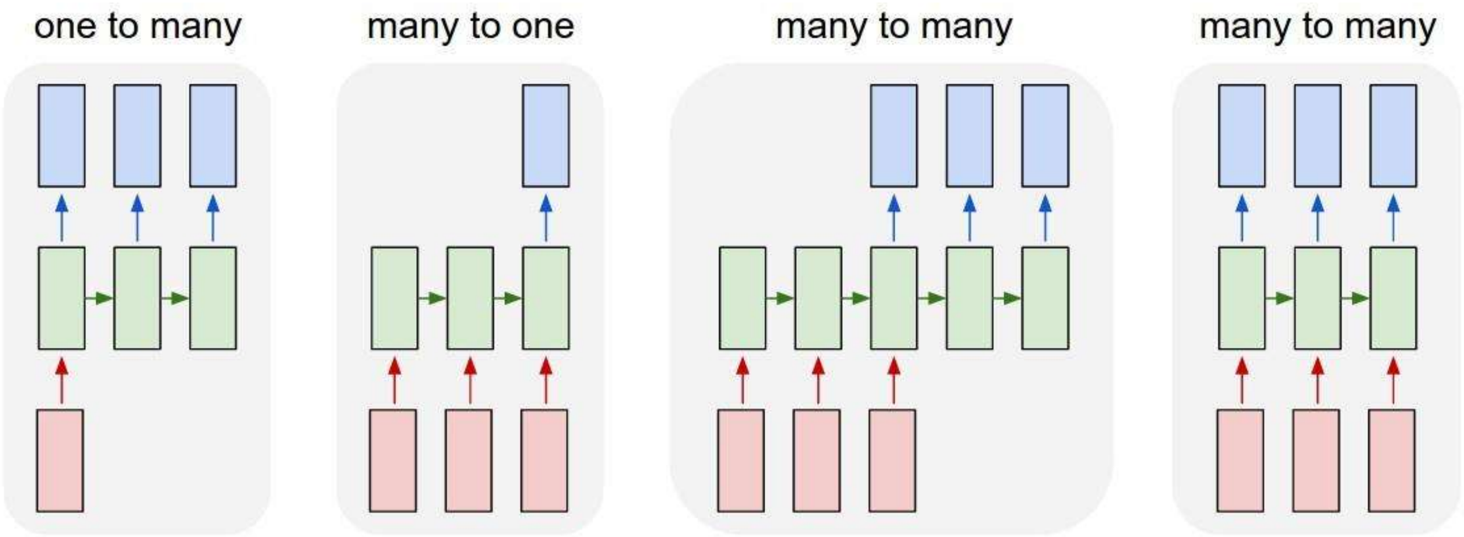




Figures from https://d2l.ai/

Recurrent Neural Networks in Use

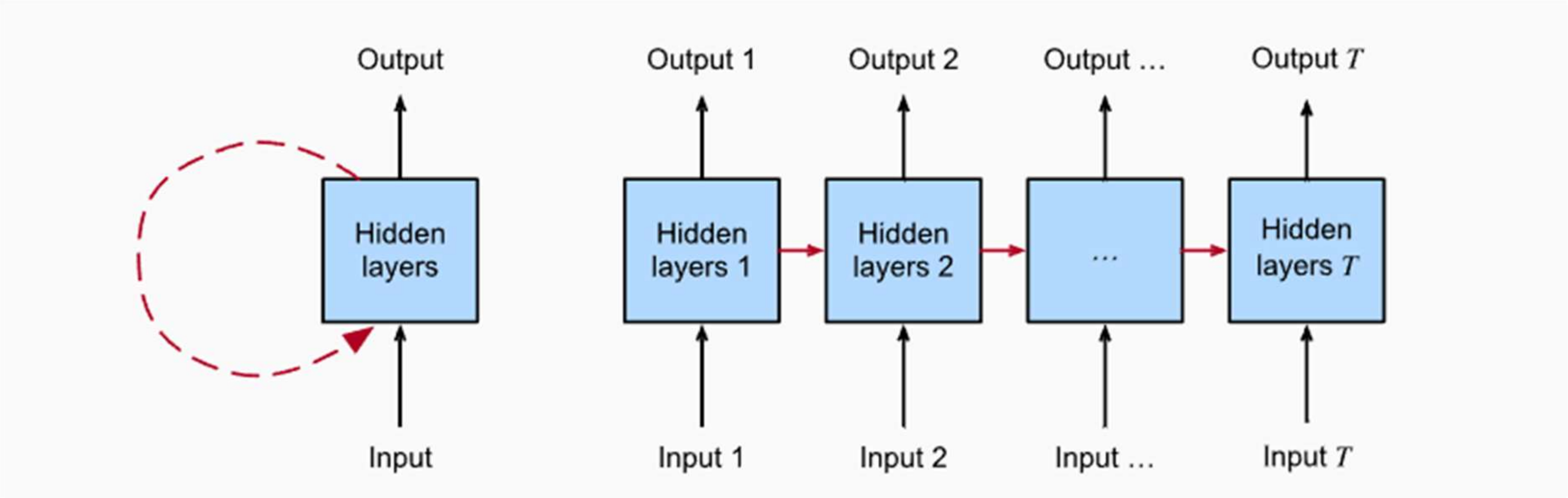
- What are the tasks that a RNN can be used for?



Figures from http://karpathy.github.io/2015/05/21/rnn-effectiveness/

Key Issues re. Recurrent Neural Networks

Memory

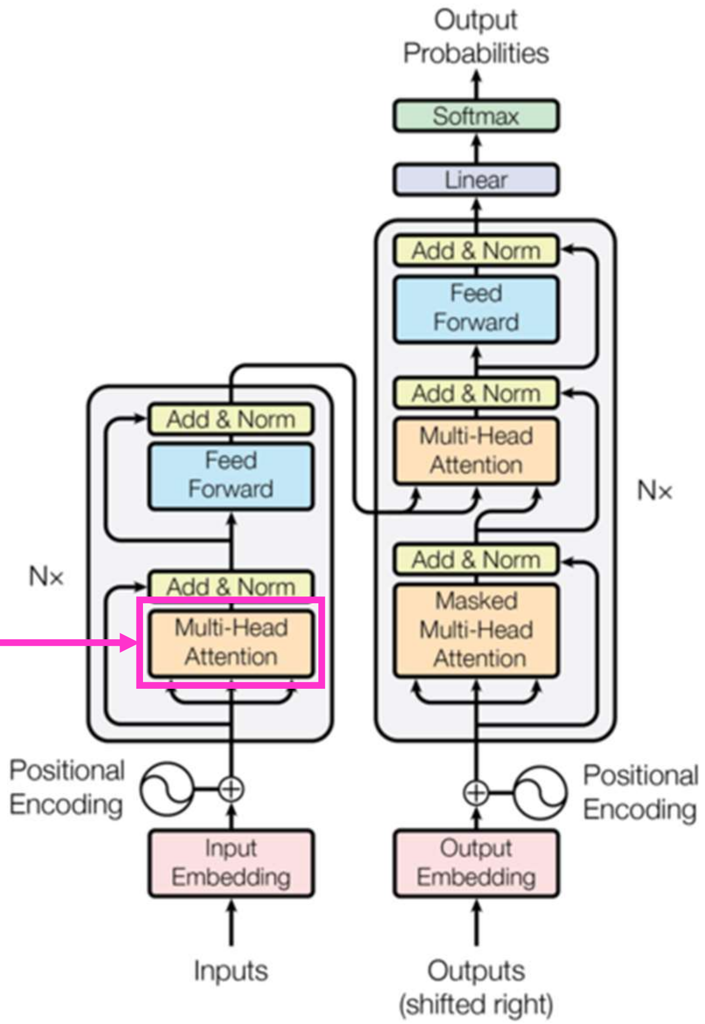


Training

Figures from https://d2l.ai

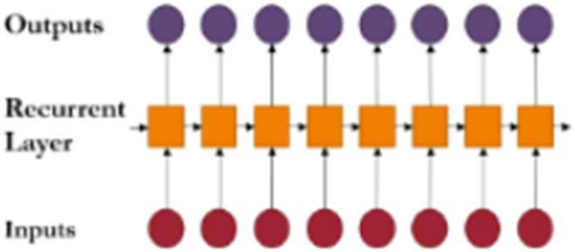
Transformers

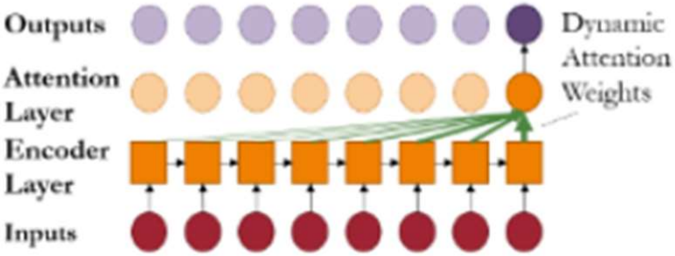
- Arguably one of the most important models proposed for NLP (and maybe other fields) in the last decade
- Key step in transformer is called **attention**



- Technical description: https://nlp.seas.harvard.edu/2018/04/03/attention.html
- Visual description: http://jalammar.github.io/illustrated-transformer/

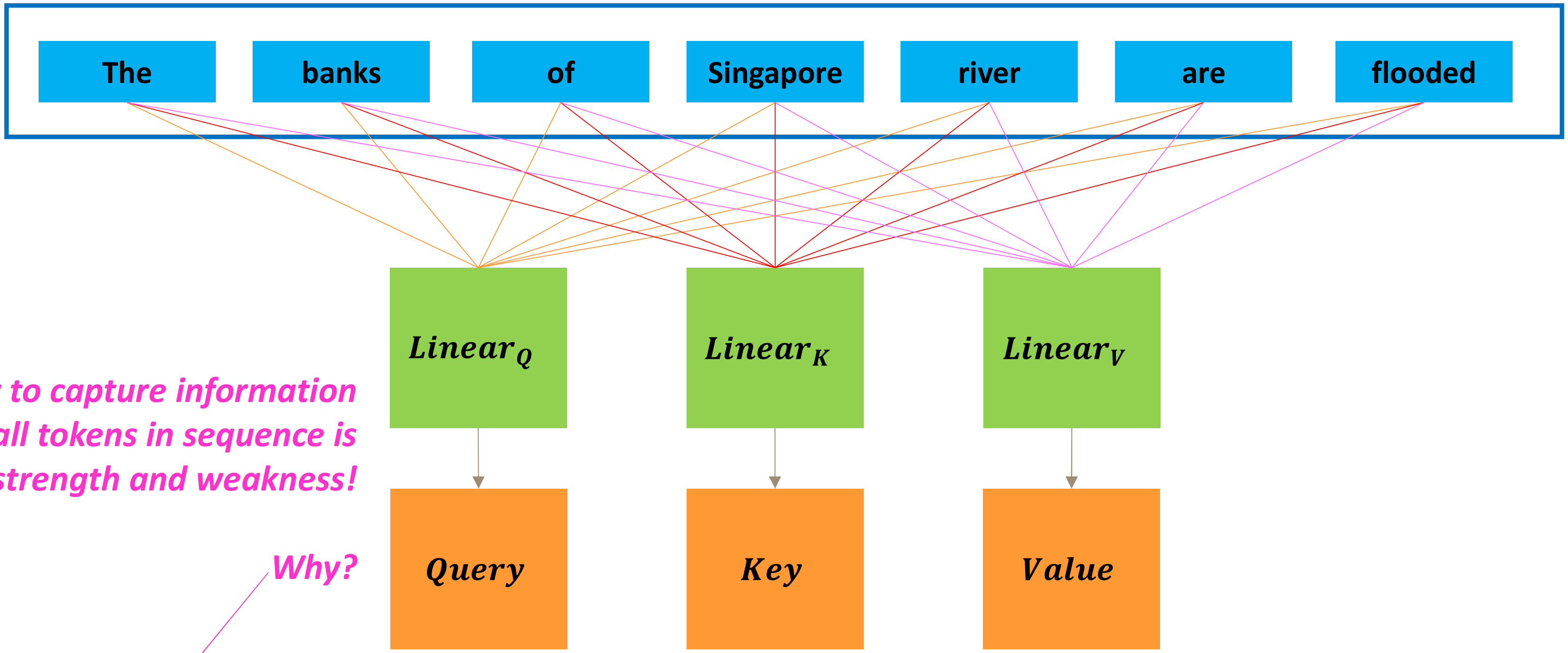
Comparing attention with RNN





Figures from Time Series Forecasting With Deep Learning: A Survey, Lim et al., 2020

A single attention step

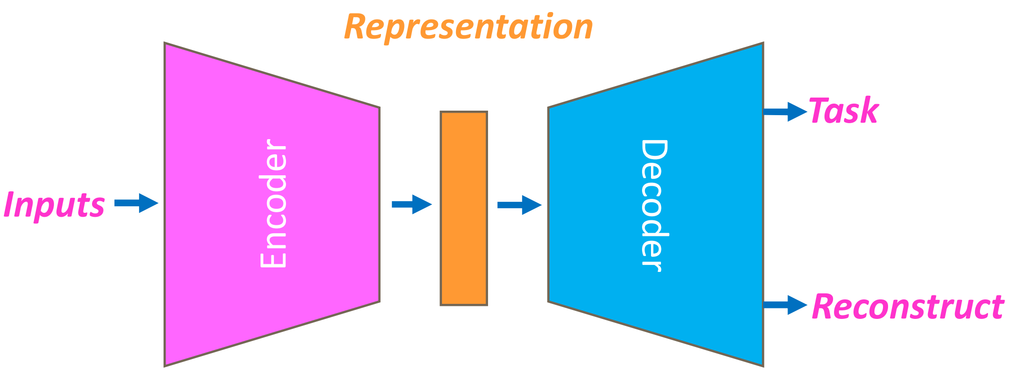


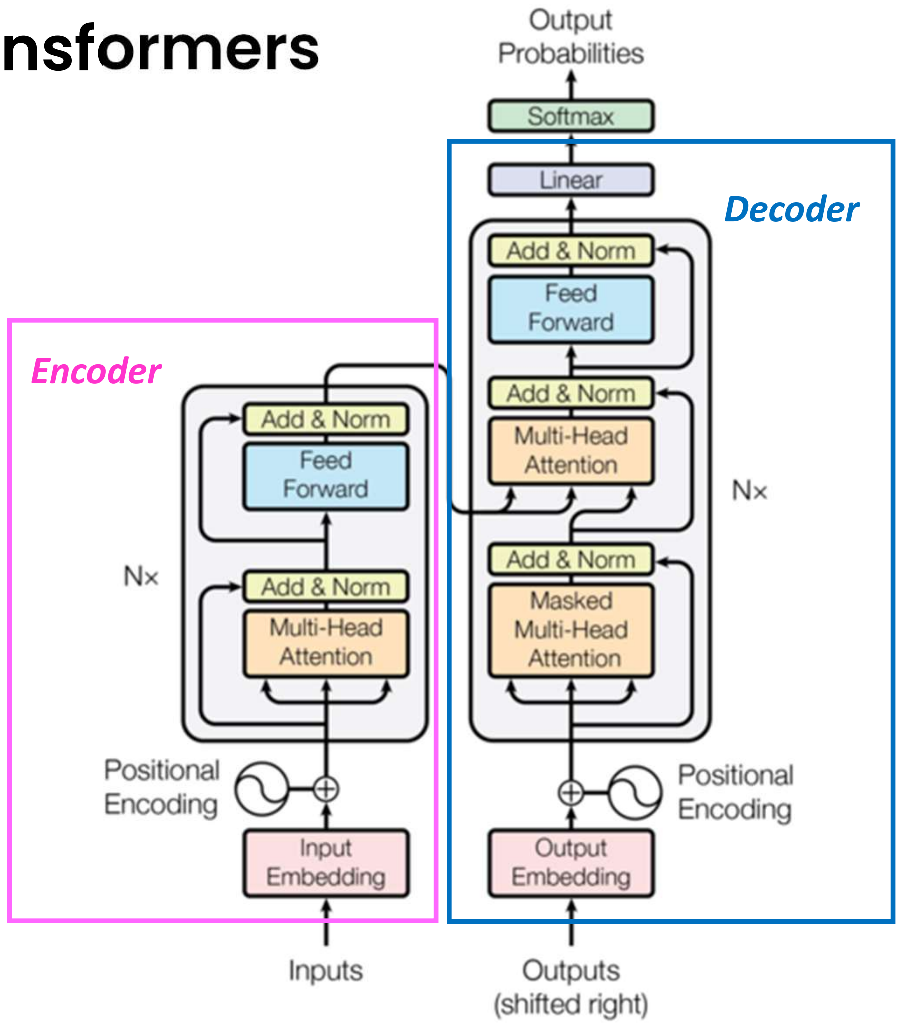
Ability to capture information from all tokens in sequence is both a strength and weakness!

Why?

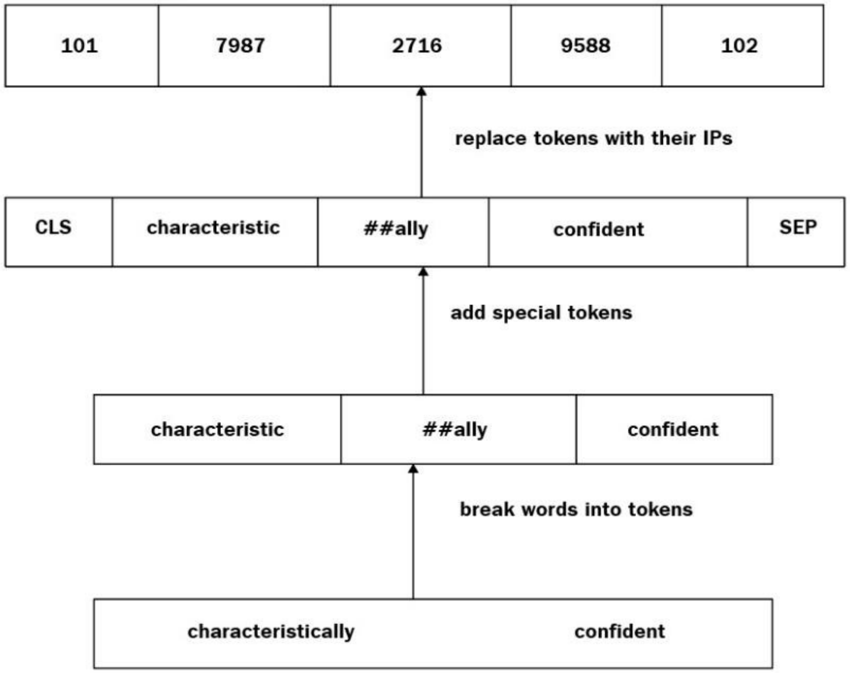
Attention Scores = **Query** · **Key** → Updated Token Representation = **Attention Scores** · **Value**

Encoders and Decoders in Transformers





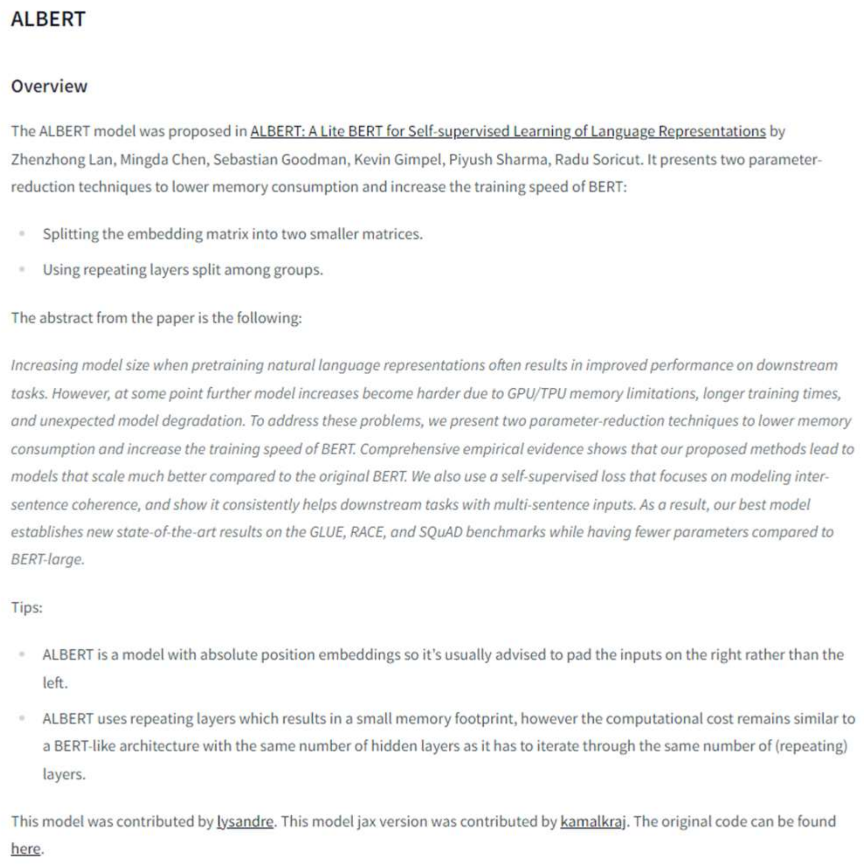
Tokenization in transformers – slightly different from prev.



WordPiece tokenization → subwords.

From http://jalammar.github.io/illustrated-transformer/

The many variants of transformers



ALBERT

Overview

The ALBERT model was proposed in ALBERT: A Lite BERT for Self-supervised Learning of Language Representations by Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, Radu Soricut. It presents two parameter-reduction techniques to lower memory consumption and increase the training speed of BERT:

- Splitting the embedding matrix into two smaller matrices.
- Using repeating layers split among groups.

The abstract from the paper is the following:

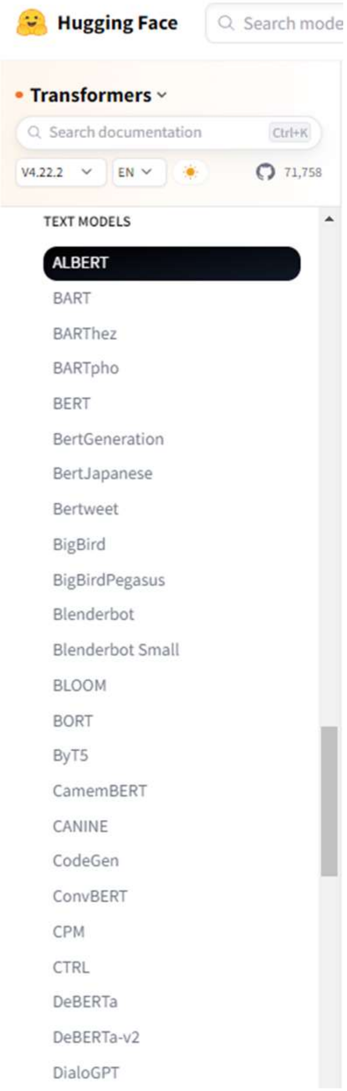
Increasing model size when pretraining natural language representations often results in improved performance on downstream tasks. However, at some point further model increases become harder due to GPU/TPU memory limitations, longer training times, and unexpected model degradation. To address these problems, we present two parameter-reduction techniques to lower memory consumption and increase the training speed of BERT. Comprehensive empirical evidence shows that our proposed methods lead to models that scale much better compared to the original BERT. We also use a self-supervised loss that focuses on modeling inter-sentence coherence, and show it consistently helps downstream tasks with multi-sentence inputs. As a result, our best model establishes new state-of-the-art results on the GLUE, RACE, and SQuAD benchmarks while having fewer parameters compared to BERT-large.

Tips:

- ALBERT is a model with absolute position embeddings so it's usually advised to pad the inputs on the right rather than the left.
- ALBERT uses repeating layers which results in a small memory footprint, however the computational cost remains similar to a BERT-like architecture with the same number of hidden layers as it has to iterate through the same number of (repeating) layers.

This model was contributed by lysandre. This model jax version was contributed by kamalkraj. The original code can be found here.

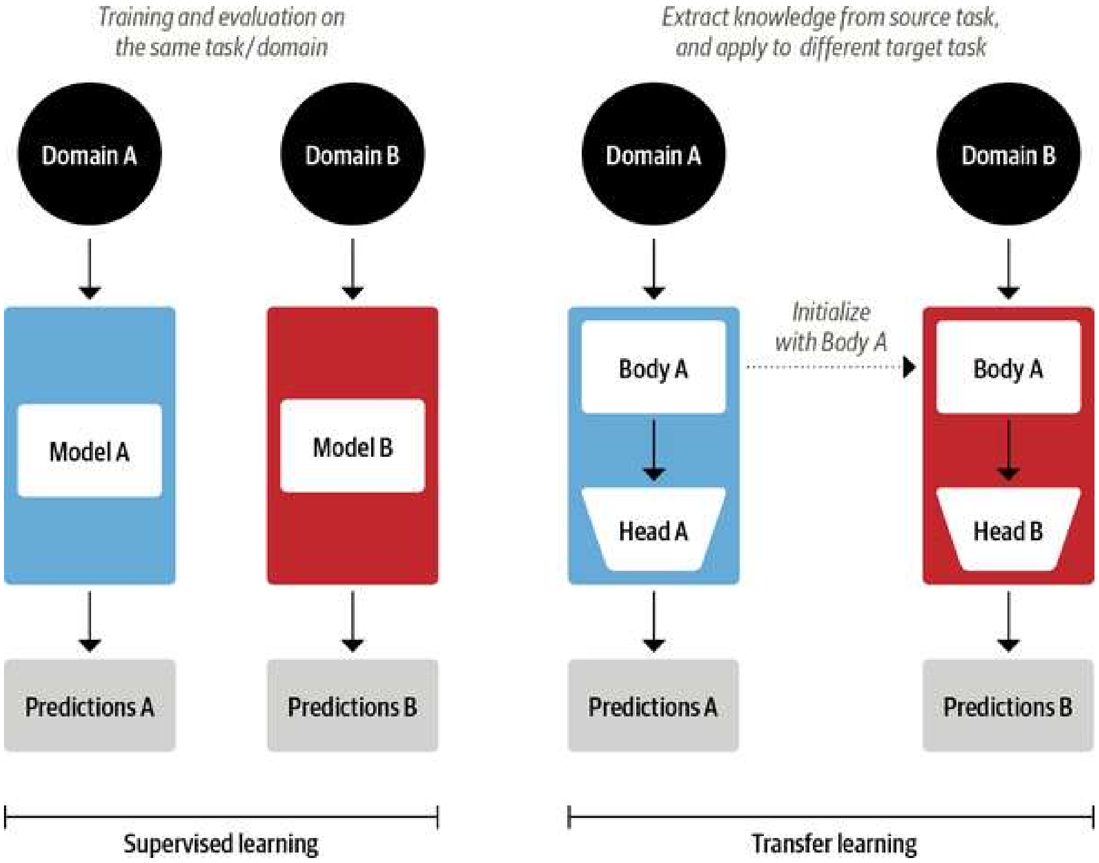
https://huggingface.co/docs/transformers/model_doc/albert



Key concept:

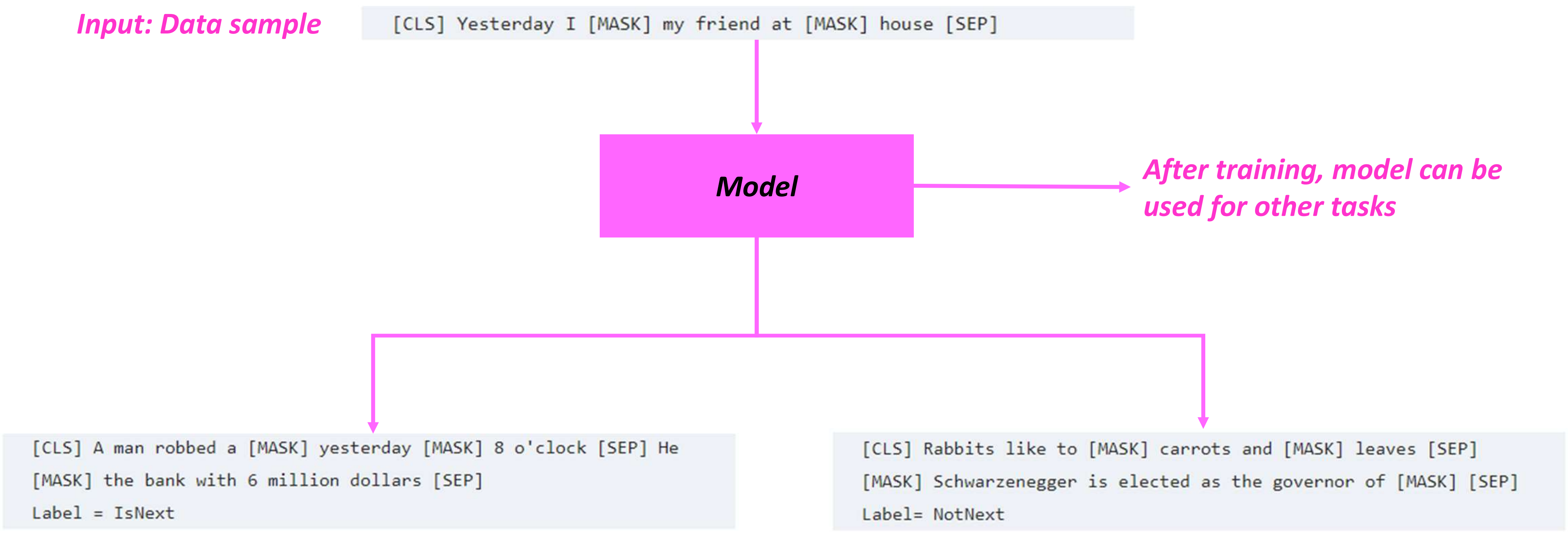
Pre-trained Models

Pre-trained models and transfer learning



From Natural Language Processing with Transformers

Language modelling



Input: Data sample

```
[CLS] Yesterday I [MASK] my friend at [MASK] house [SEP]
```

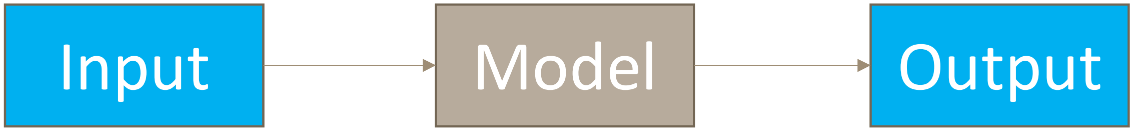
After training, model can be used for other tasks

```
[CLS] A man robbed a [MASK] yesterday [MASK] 8 o'clock [SEP] He
[MASK] the bank with 6 million dollars [SEP]
Label = IsNext
```

```
[CLS] Rabbits like to [MASK] carrots and [MASK] leaves [SEP]
[MASK] Schwarzenegger is elected as the governor of [MASK] [SEP]
Label= NotNext
```

From Natural Language Processing with Transformers

Different NLP Tasks


Input

Model

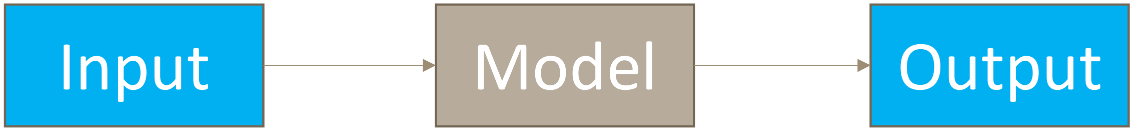
Output

- What is the task?
- What are the inputs?
- What are the outputs/labels?

A datapoint

{'document': 'Recent reports have linked some France-based players with returns to Wales.\n"I\'ve always felt - and this is with my rugby hat on now; this is not region or WRU - I\'d rather spend that money on keeping players in Wales," said Davies.\nThe WRU provides £2m to the fund and £1.3m comes from the regions.\nFormer Wales and British and Irish Lions fly-half Davies became WRU chairman on Tuesday 21 October, succeeding deposed David Pickering following governing body elections.\nHe is now serving a notice period to leave his role as Newport Gwent Dragons chief executive after being voted on to the WRU board in September.\nDavies was among the leading figures among Dragons, Ospreys, Scarlets and Cardiff Blues officials who were embroiled in a protracted dispute with the WRU that ended in a £60m deal in August this year.\nIn the wake of that deal being done, Davies said the £3.3m should be spent on ensuring current Wales-based stars remain there.\nIn recent weeks, Racing Metro flanker Dan Lydiate was linked with returning to Wales.\nLikewise the Paris club\'s scrum-half Mike Phillips and centre Jamie Roberts were also touted for possible returns.\nWales coach Warren Gatland has said: "We haven\'t instigated contact with the players.\n"But we are aware that one or two of them are keen to return to Wales sooner rather than later."\nSpeaking to Scrum V on BBC Radio Wales, Davies re-iterated his stance, saying keeping players such as Scarlets full-back Liam Williams and Ospreys flanker Justin Tipuric in Wales should take precedence.\n"It\'s obviously a limited amount of money [available]. The union are contributing 60% of that contract and the regions are putting £1.3m in.\n"So it\'s a total pot of just over £3m and if you look at the sorts of salaries that the... guys... have been tempted to go overseas for [are] significant amounts of money.\n"So if we were to bring the players back, we\'d probably get five or six players.\n"And I\'ve always felt - and this is with my rugby hat on now; this is not region or WRU - I\'d rather spend that money on keeping players in Wales.\n"There are players coming out of contract, perhaps in the next year or so… you\'re looking at your Liam Williams\' of the world; Justin Tipuric for example - we need to keep these guys in Wales.\n"We actually want them there. They are the ones who are going to impress the young kids, for example.\n"They are the sort of heroes that our young kids want to emulate.\n"So I would start off [by saying] with the limited pot of money, we have to retain players in Wales.\n"Now, if that can be done and there\'s some spare monies available at the end, yes, let\'s look to bring players back.\n"But it\'s a cruel world, isn\'t it?\n"It\'s fine to take the buck and go, but great if you can get them back as well, provided there\'s enough money."\nBritish and Irish Lions centre Roberts has insisted he will see out his Racing Metro contract.\nHe and Phillips also earlier dismissed the idea of leaving Paris.\nRoberts also admitted being hurt by comments in French Newspaper L\'Equipe attributed to Racing Coach Laurent Labit questioning their effectiveness.\nCentre Roberts and flanker Lydiate joined Racing ahead of the 2013-14 season while scrum-half Phillips moved there in December 2013 after being dismissed for disciplinary reasons by former club Bayonne.', 'id': '29750031', 'summary': 'New Welsh Rugby Union chairman Gareth Davies believes a joint £3.3m WRU-regions fund should be used to retain home-based talent such as Liam Williams, not bring back exiled stars.'}

Different NLP Tasks

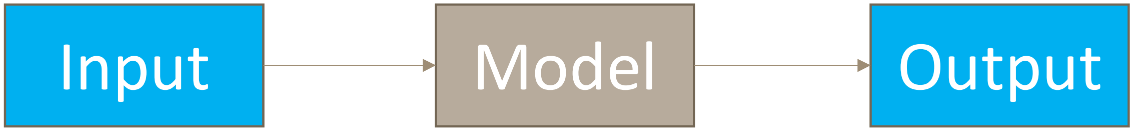


- What is the task?
- What are the inputs?
- What are the outputs/labels?

A datapoint

```
{
    "label": 0,
    "text": "I love sci-fi and am willing to put up with a lot. Sci-fi movies/TV are usually underfunded, under-appreciated and misunderstood. I tried to like this, I really did, but it is to good TV sci-fi as
Babylon 5 is to Star Trek (the original). Silly prosthetics, cheap cardboard sets, stilted dialogues, CG that doesn't match the background, and painfully one-dimensional characters cannot be
overcome with a 'sci-fi' setting. (I'm sure there are those of you out there who think Babylon 5 is good sci-fi TV. It's not. It's clichéd and uninspiring.) While US viewers might like emotion and
character development, sci-fi is a genre that does not take itself seriously (cf. Star Trek). It may treat important issues, yet not as a serious philosophy. It's really difficult to care about the characters
here as they are not simply foolish, just missing a spark of life. Their actions and reactions are wooden and predictable, often painful to watch. The makers of Earth KNOW it's rubbish as they have to
always say \"Gene Roddenberry's Earth...\" otherwise people would not continue watching. Roddenberry's ashes must be turning in their orbit as this dull, cheap, poorly edited (watching it without
advert breaks really brings this home) trudging Trabant of a show lumbers into space. Spoiler. So, kill off a main character. And then bring him back as another actor. Jeeez! Dallas all over again.",
}
```

Different NLP Tasks



Input

Model

Output

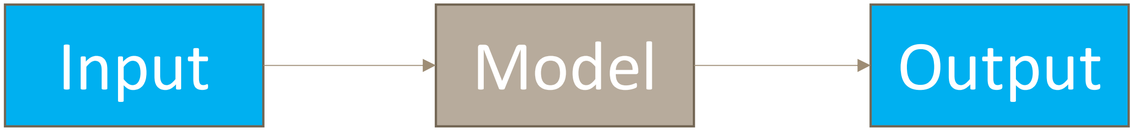
- What is the task?
- What are the inputs?
- What are the outputs/labels?

A datapoint

```
{'chunk_tags': [11, 21, 11, 12, 21, 22, 11, 12, 0],
 'id': '0',
 'ner_tags': [3, 0, 7, 0, 0, 0, 7, 0, 0],
 'pos_tags': [22, 42, 16, 21, 35, 37, 16, 21, 7],
 'tokens': ['EU',
  'rejects',
  'German',
  'call',
  'to',
  'boycott',
  'British',
  'lamb',
  '.']}

['O', 'B-PER', 'I-PER', 'B-ORG', 'I-ORG', 'B-LOC', 'I-LOC', 'B-MISC', 'I-MISC']
```

Different NLP Tasks

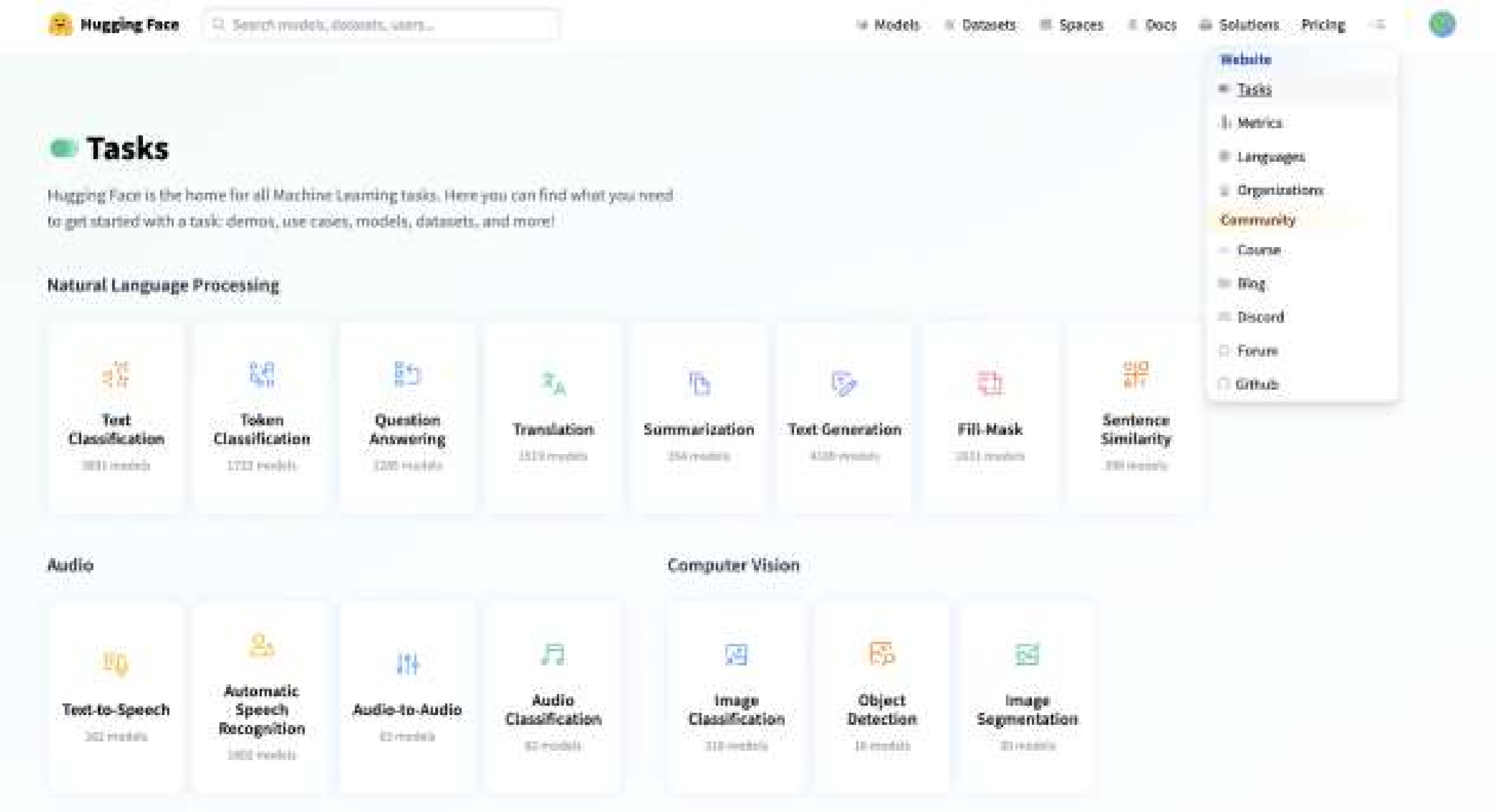


- What is the task?
- What are the inputs?
- What are the outputs/labels?

A datapoint

{'en': 'Measuring instruments for cold water meters for non-clean water, alcohol meters, certain weights, tyre pressure gauges and equipment to measure the standard mass of grain or the size of ship tanks have been replaced, in practice, by more modern digital equipment.', 'ro': 'Instrumentele de măsură pentru contoarele de apă rece pentru apa murdară, alcoolmetrele, anumite greutăți, manometrele pentru presiunea din pneuri și echipamentele de măsură pentru masa standard de cereale sau pentru dimensiunea rezervoarelor de nave au fost înlocuite, în practică, de echipamente digitale mai moderne.'}

Different NLP Tasks



Tasks

Hugging Face is the home for all Machine Learning tasks. Here you can find what you need to get started with a task: demos, use cases, models, datasets, and more!

Natural Language Processing

- Text Classification 3831 models
- Token Classification 1722 models
- Question Answering 1285 models
- Translation 1519 models
- Summarization 354 models
- Text Generation 4108 models
- Fill-Mask 2811 models
- Sentence Similarity 398 models

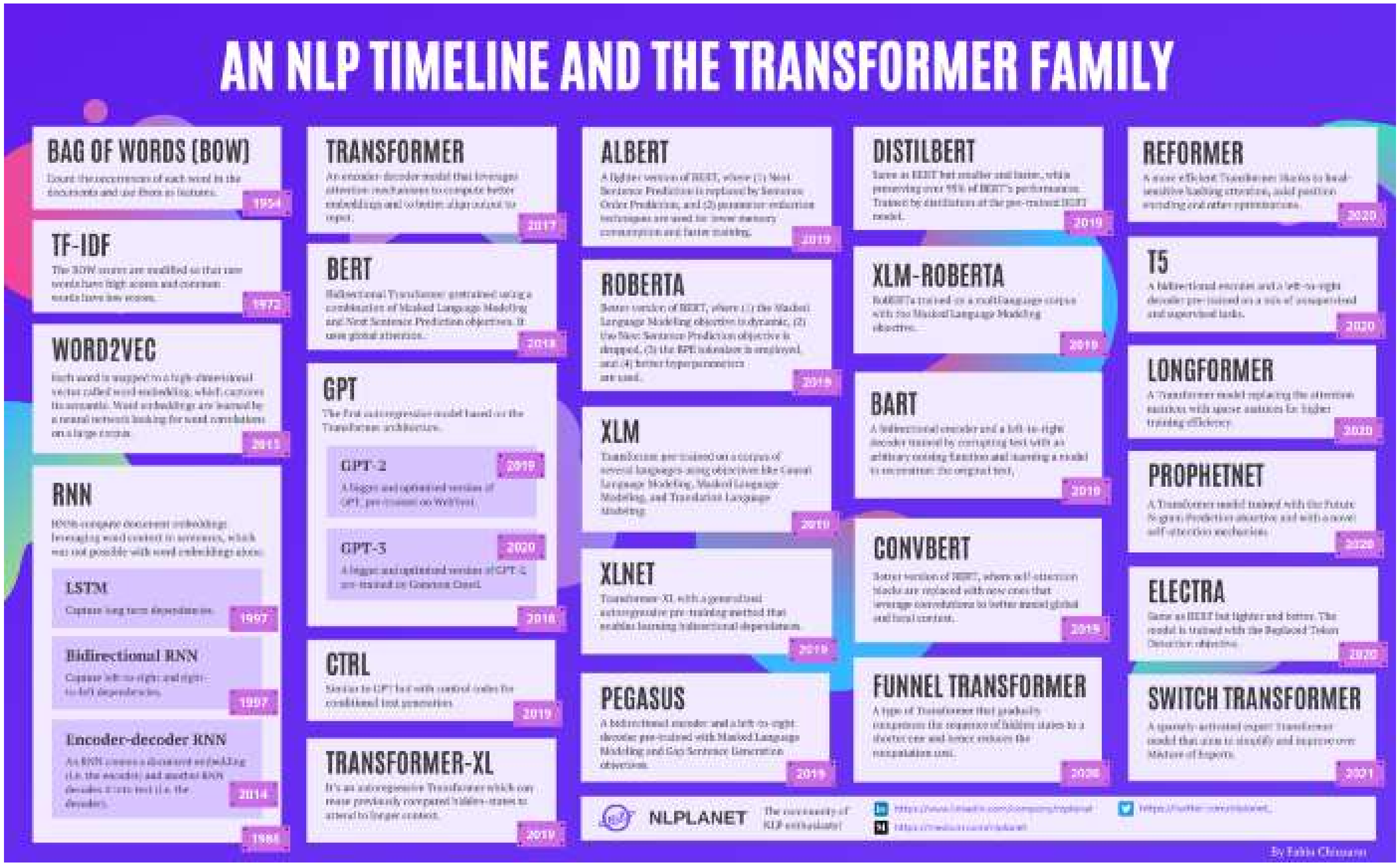
Audio

- Text-to-Speech 302 models
- Automatic Speech Recognition 1802 models
- Audio-to-Audio 63 models
- Audio Classification 82 models

Computer Vision

- Image Classification 318 models
- Object Detection 16 models
- Image Segmentation 33 models

Recent NLP Developments



AN NLP TIMELINE AND THE TRANSFORMER FAMILY

BAG OF WORDS (BOW)
Count the occurrences of each word in the documents and use them as features.
1954

TF-IDF
The BOW scores are modified so that rare words have high scores and common words have low scores.
1972

WORD2VEC
Each word is mapped to a high-dimensional vector called word embedding, which captures its semantic. Word embeddings are learned by a neural network looking for word correlations on a large corpus.
2013

RNN
RNNs compute document embeddings leveraging word context in sentences, which was not possible with word embeddings alone.

LSTM
Capture long term dependencies.
1997

Bidirectional RNN
Capture left-to-right and right-to-left dependencies.
1997

Encoder-decoder RNN
An RNN creates a document embedding (i.e. the encoder) and another RNN decodes it into text (i.e. the decoder).
2014

1986

TRANSFORMER
An encoder-decoder model that leverages attention mechanisms to compute better embeddings and to better align output to input.
2017

BERT
Bidirectional Transformer pretrained using a combination of Masked Language Modeling and Next Sentence Prediction objectives. It uses global attention.
2018

GPT
The first autoregressive model based on the Transformer architecture.

GPT-2
2019
A bigger and optimized version of GPT, pre-trained on WebText.

GPT-3
2020
A bigger and optimized version of GPT-2, pre-trained on Common Crawl.

2018

CTRL
Similar to GPT but with control codes for conditional text generation.
2019

TRANSFORMER-XL
It's an autoregressive Transformer which can reuse previously computed hidden-states to attend to longer context.
2019

ALBERT
A lighter version of BERT, where (1) Next Sentence Prediction is replaced by Sentence Order Prediction, and (2) parameter-reduction techniques are used for lower memory consumption and faster training.
2019

ROBERTA
Better version of BERT, where (1) the Masked Language Modeling objective is dynamic, (2) the Next Sentence Prediction objective is dropped, (3) the BPE tokenizer is employed, and (4) better hyperparameters are used.
2019

XLM
Transformer pre-trained on a corpus of several languages using objectives like Causal Language Modeling, Masked Language Modeling, and Translation Language Modeling.
2019

XLNET
Transformer-XL with a generalized autoregressive pre-training method that enables learning bidirectional dependencies.
2019

PEGASUS
A bidirectional encoder and a left-to-right decoder pre-trained with Masked Language Modeling and Gap Sentence Generation objectives.
2019

DISTILBERT
Same as BERT but smaller and faster, while preserving over 95% of BERT's performances. Trained by distillation of the pre-trained BERT model.
2019

XLM-ROBERTA
RoBERTa trained on a multilanguage corpus with the Masked Language Modeling objective.
2019

BART
A bidirectional encoder and a left-to-right decoder trained by corrupting text with an arbitrary noising function and learning a model to reconstruct the original text.
2019

CONVBERT
Better version of BERT, where self-attention blocks are replaced with new ones that leverage convolutions to better model global and local context.
2019

FUNNEL TRANSFORMER
A type of Transformer that gradually compresses the sequence of hidden states to a shorter one and hence reduces the computation cost.
2020

REFORMER
A more efficient Transformer thanks to local-sensitive hashing attention, axial position encoding and other optimizations.
2020

T5
A bidirectional encoder and a left-to-right decoder pre-trained on a mix of unsupervised and supervised tasks.
2020

LONGFORMER
A Transformer model replacing the attention matrices with sparse matrices for higher training efficiency.
2020

PROPHETNET
A Transformer model trained with the Future N-gram Prediction objective and with a novel self-attention mechanism.
2020

ELECTRA
Same as BERT but lighter and better. The model is trained with the Replaced Token Detection objective.
2020

SWITCH TRANSFORMER
A sparsely-activated expert Transformer model that aims to simplify and improve over Mixture of Experts.
2021

NLPLANET The community of NLP enthusiasts!
https://www.linkedin.com/company/nlplanet/
https://medium.com/nlplanet
https://twitter.com/nlplanet_

By Fabio Chiusano

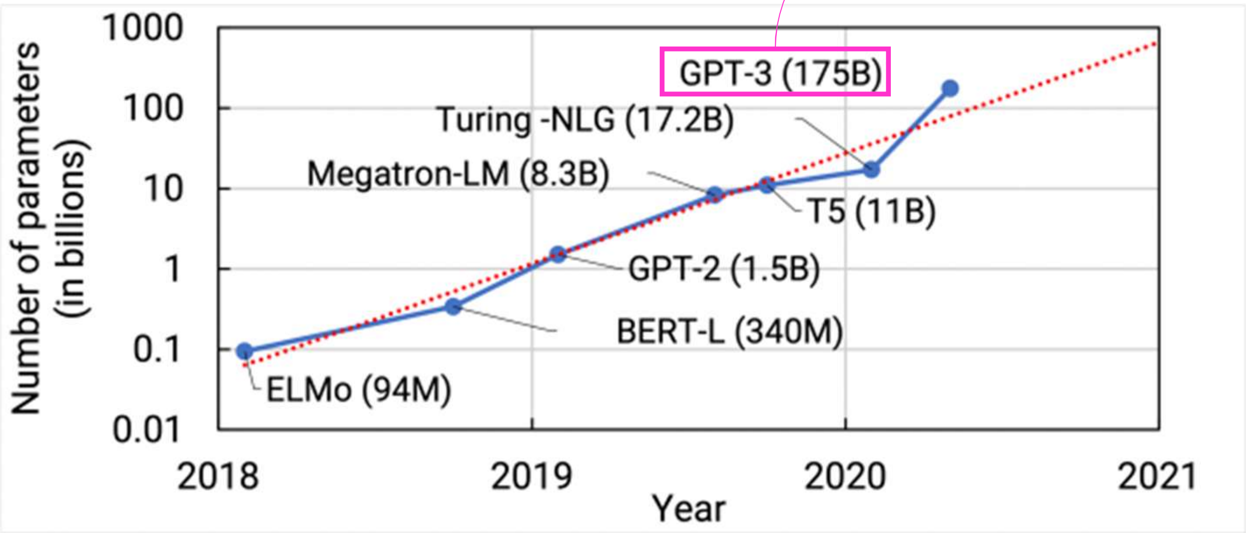
From https://medium.com/nlplanet/a-brief-timeline-of-nlp-from-bag-of-words-to-the-transformer-family-7caad8bbba56

Key differences between the (overwhelming) # of transformers

- What is the dataset that the model is (pre)-trained on?
- How do you tokenize and generate training samples?
- How do you mask?
- How do you train?
- How do you encode and/or decode?
- How do you address limitations of the attention score?

- **Why do we use pre-trained models?**
- **What are large language models?**

Est. cost to train GPT-3 - $14m in compute



- **What is model fine-tuning (recall transfer learning)?**

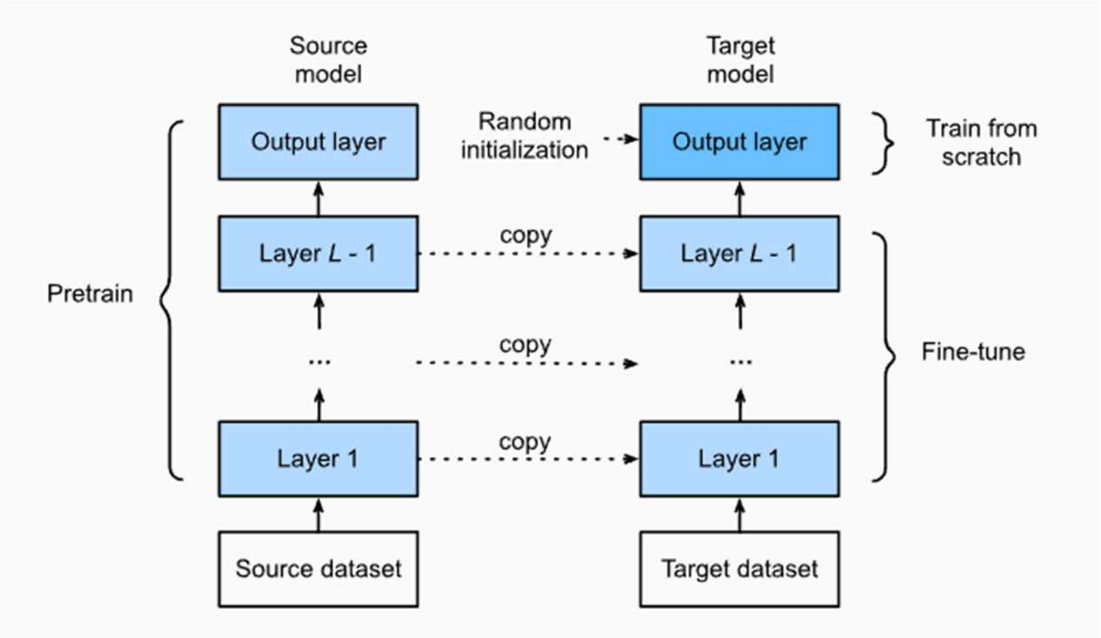
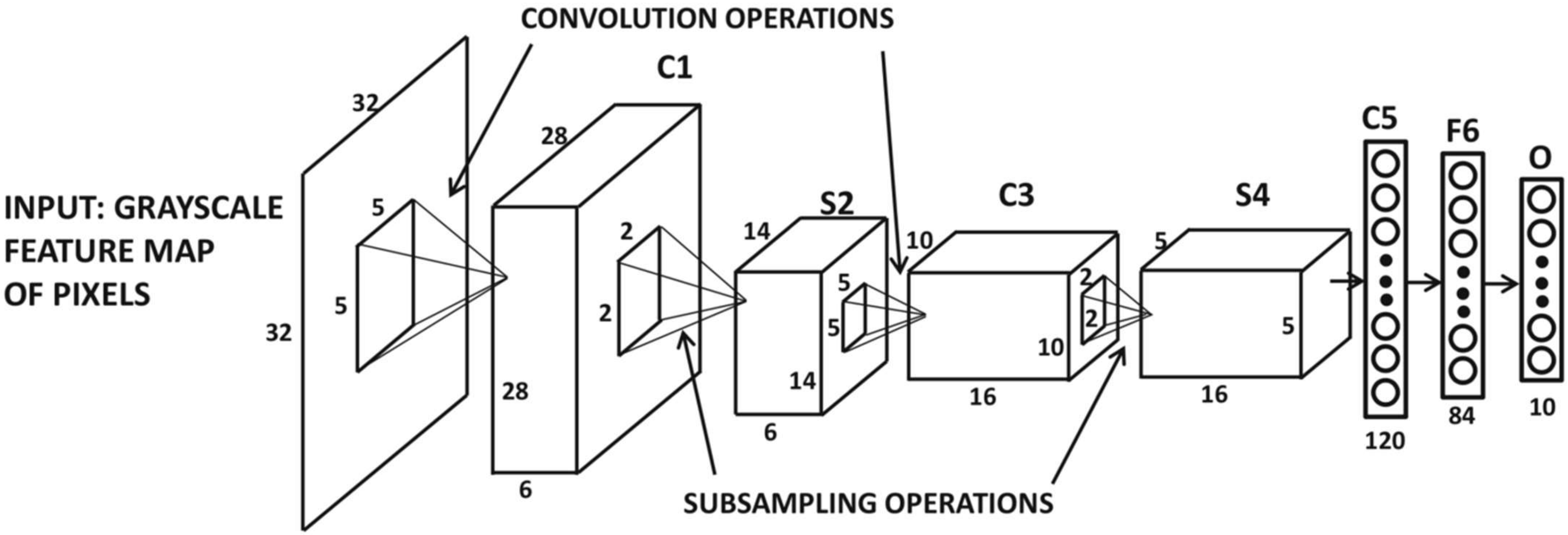


Fig. 14.2.1 Fine tuning.

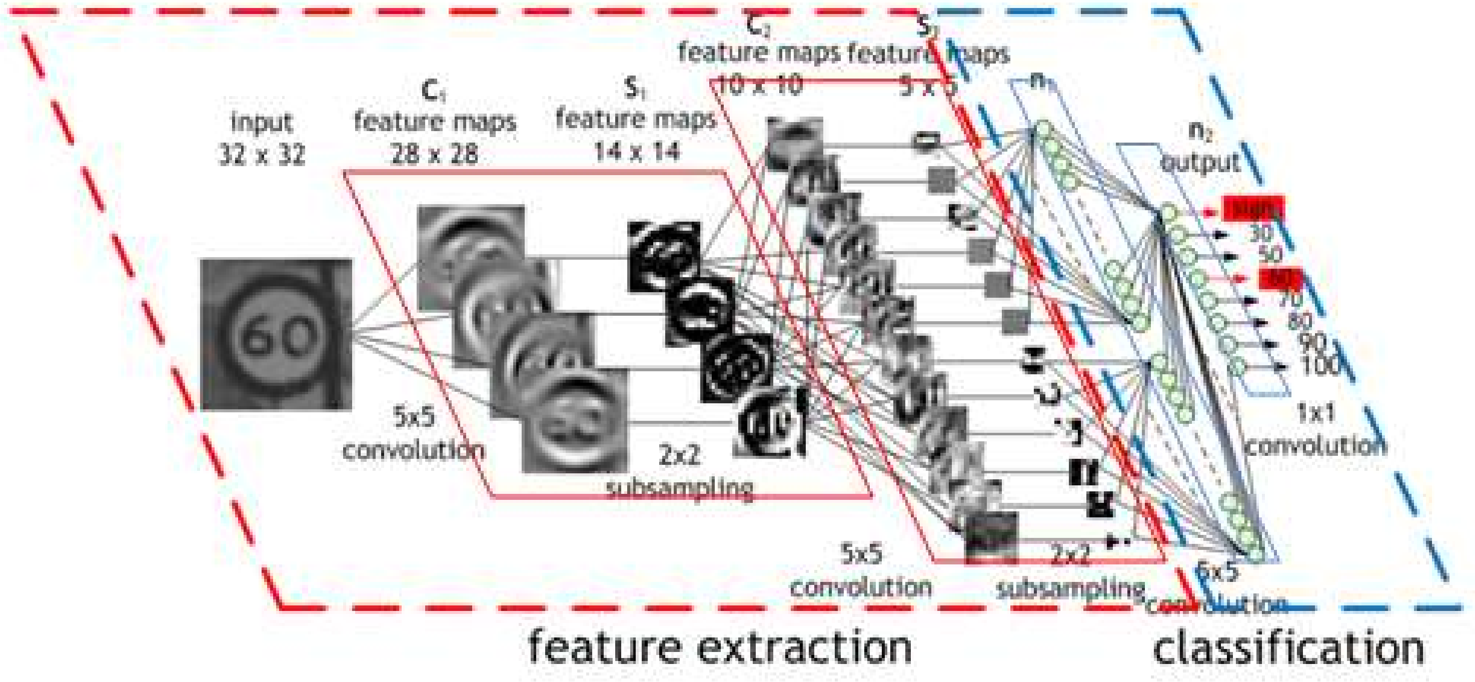
Figures from https://d2l.ai

CNN vs NLP Models



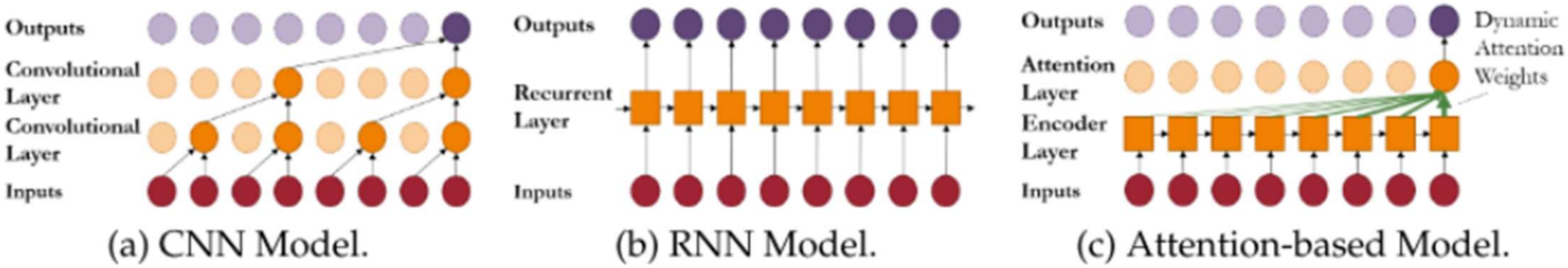
http://yann.lecun.com/exdb/publis/pdf/lecun-01a.pdf

CNN vs NLP Models



From https://developer.nvidia.com/discover/convolutional-neural-network

Comparing CNN with RNN & Attention-based Models



(a) CNN Model. (b) RNN Model. (c) Attention-based Model.

Figure 1: Incorporating temporal information using different encoder architectures.

Figures from Time Series Forecasting With Deep Learning: A Survey, Lim et al., 2020

Capturing multimodal information

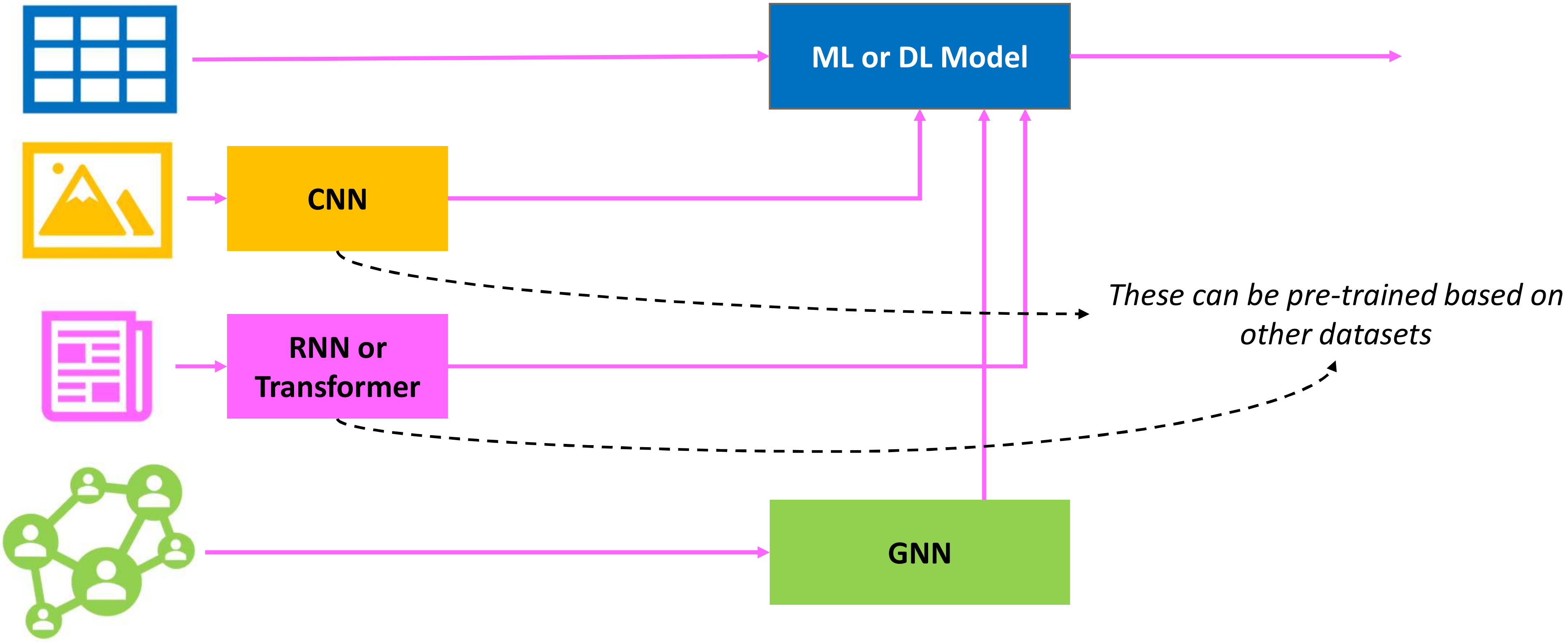
ML or DL Model

CNN

RNN or Transformer

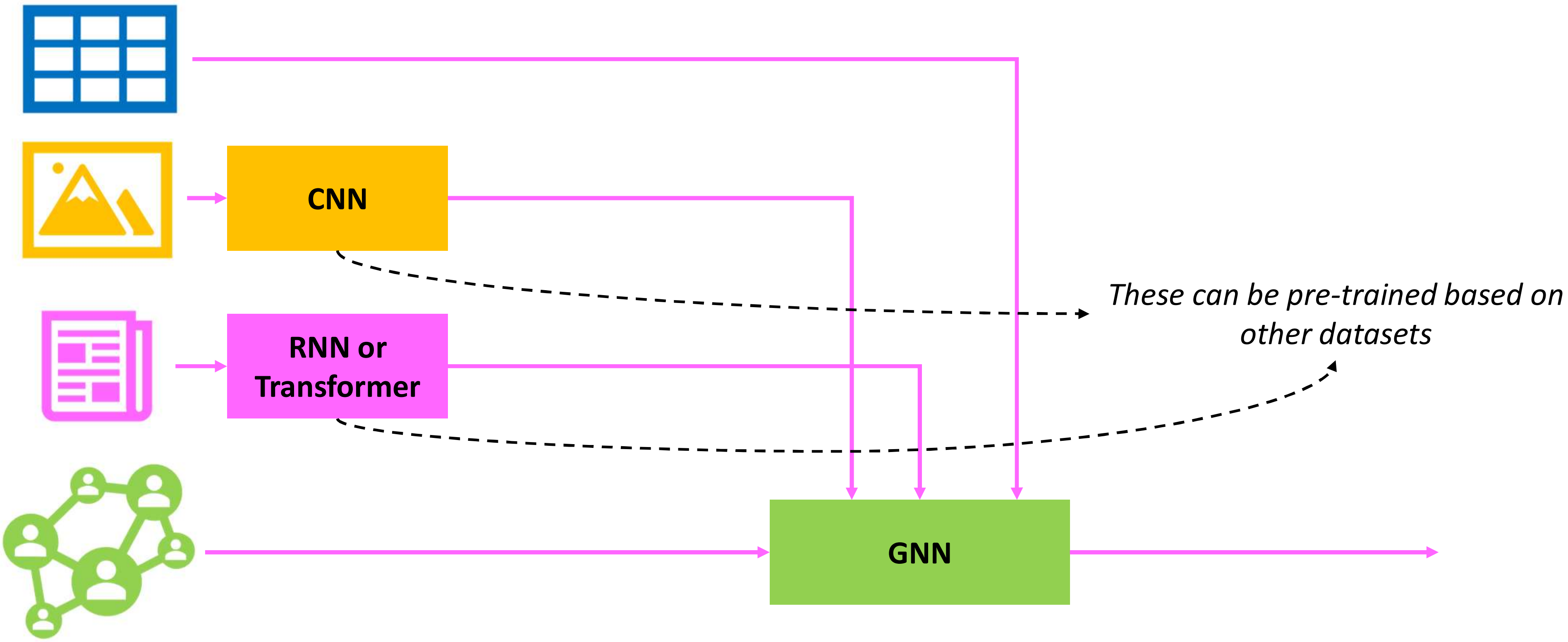
GNN

Capturing multimodal information



These can be pre-trained based on other datasets

Capturing multimodal information



Capturing multimodal information

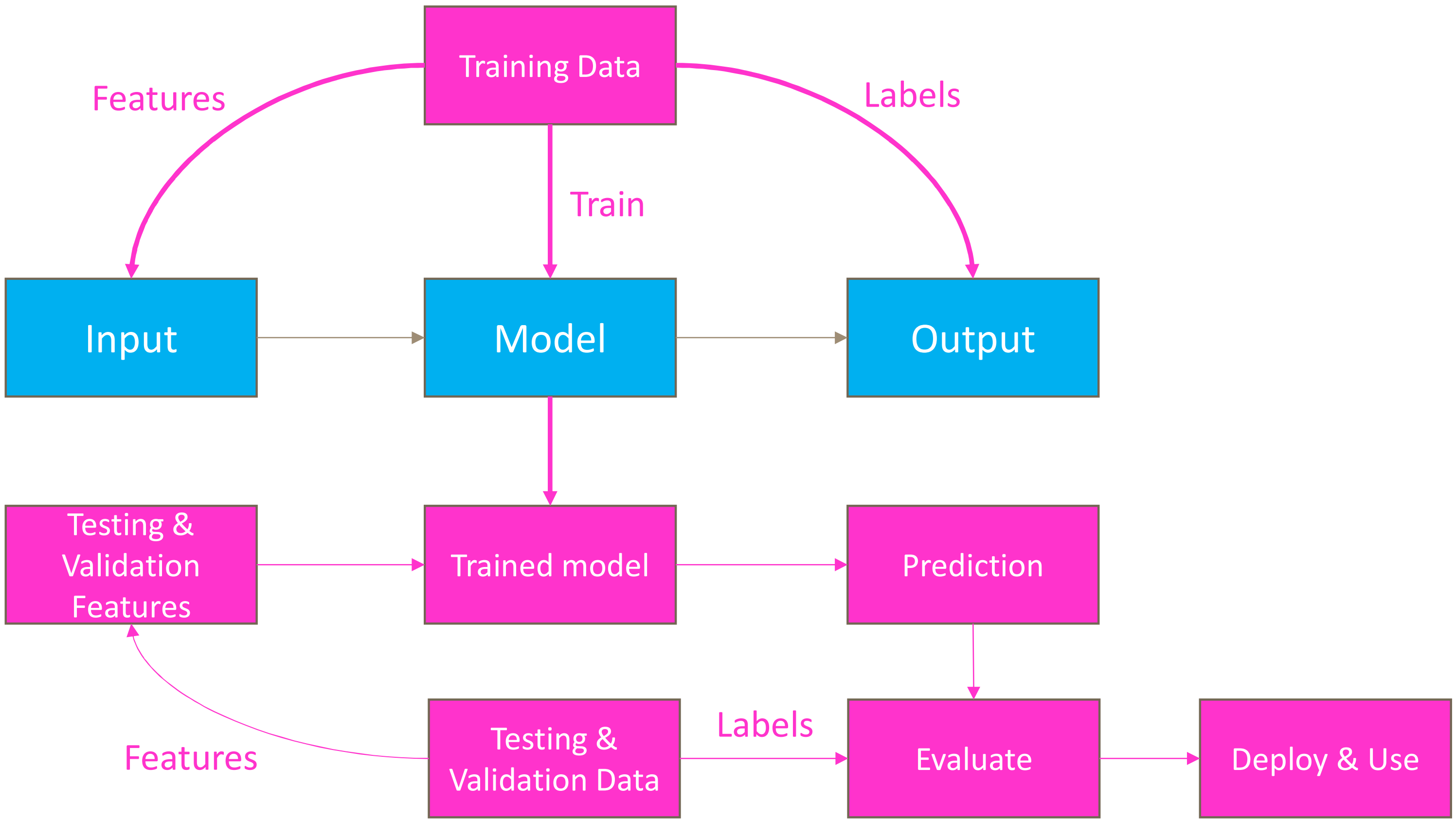
Still an active area of research!

- How to most effectively combine information from different modalities (multimodal fusion)?
  - How to align?
  - How to balance?

- How to interpret and explain a multimodal model?

Model Risks

General Flow



What to watch out for

Data	Data quality, bias, leakage
Modelling	Model Selection, Under vs. Overfitting, Explainability and Interpretability
Evaluation	Appropriateness of metrics
Deploy	Drifts

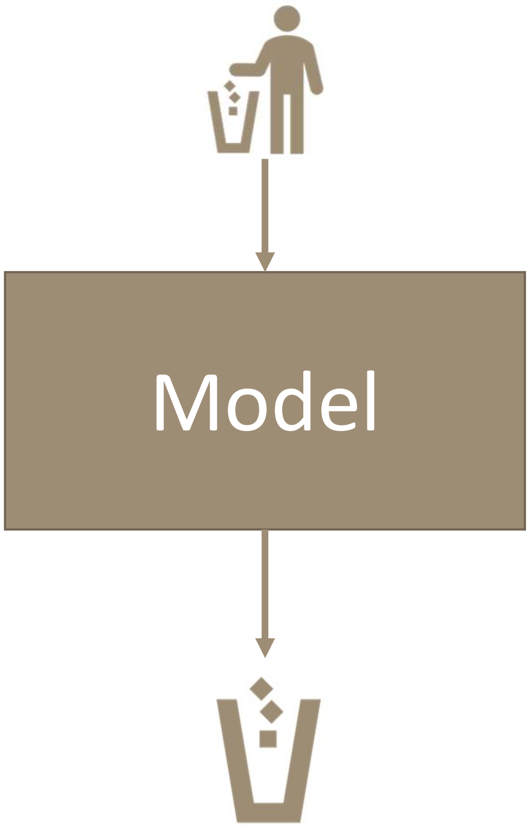
Data quality

- **Accuracy**
  - E.g., mis-labelled illicit transactions
- **Completeness**
  - E.g., omission of transactions stored in another banking system
- **Consistency**
  - E.g., unclear instructions when designating a loan as defaulted
- **Currency**
  - E.g., characteristics/distribution of fraudulent transactions changing over time due to change in tech. and consumer behaviour

Addressing all of these is a pipe-dream.

But important to know if these exist in the data.

Data Bias

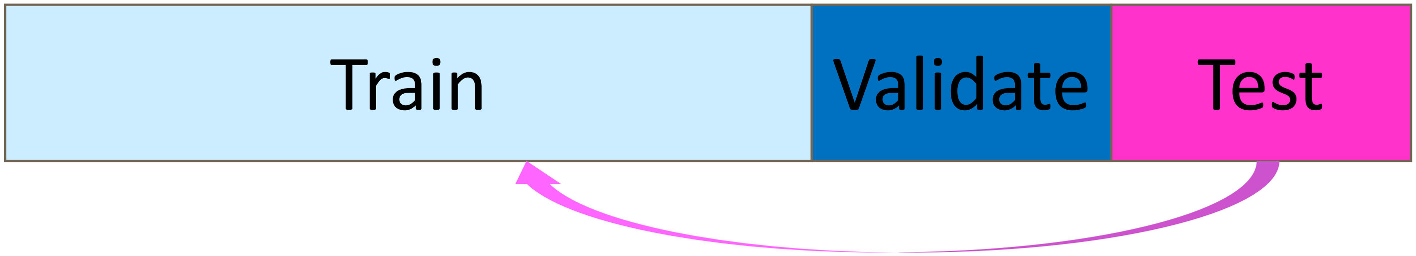


Model

Examples of harmful data bias

- **Distribution** bias
  - Personal attributes
- **Representation** bias
  - Match general population but under-represent certain segments
- **Implicit** bias
  - Not all bias are obvious, e.g., gender vis-à-vis income vis-à-vis location vis-à-vis race
- **Labelling** bias
  - Even experts label things differently, e.g., 2nd opinions?

Data leakage



- Very common even for random train, validation test splits
  - Using total time customer spent in bank to predict customer purchase intent so as to act on it while customer still in bank
  - Use data before t to predict prob. of default at $t + 1$, data before t includes post-default adjustments
  - Predicting illicit transactions using complete incident reports
  - Using mean and standard deviation of entire dataset to scale/normalize data

There is a whole paper on this at https://reproducible.cs.princeton.edu/

What is a good indication of data issues?

- **Extreme results**
  - Especially when compared to a naïve model

- **Both ways**
  - **Too good** – data leakage, evaluation errors ….
  - **Too bad** – dirty data, evaluation errors …

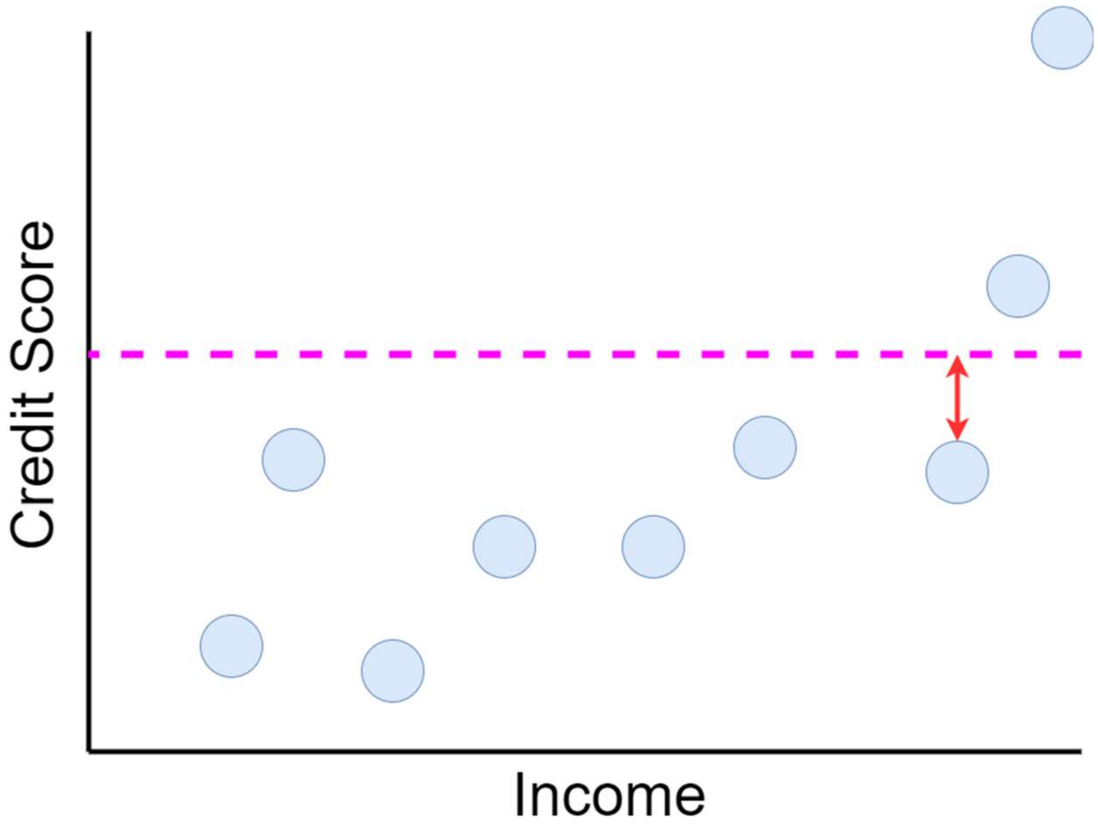
What to watch out for

Data	Data quality, bias, leakage
Modelling	Model Selection, Under vs. Overfitting, Explainability and Interpretability *(leave this to the end)*
Evaluation	Appropriateness of metrics
Deploy	Drifts

Appropriateness of Metrics

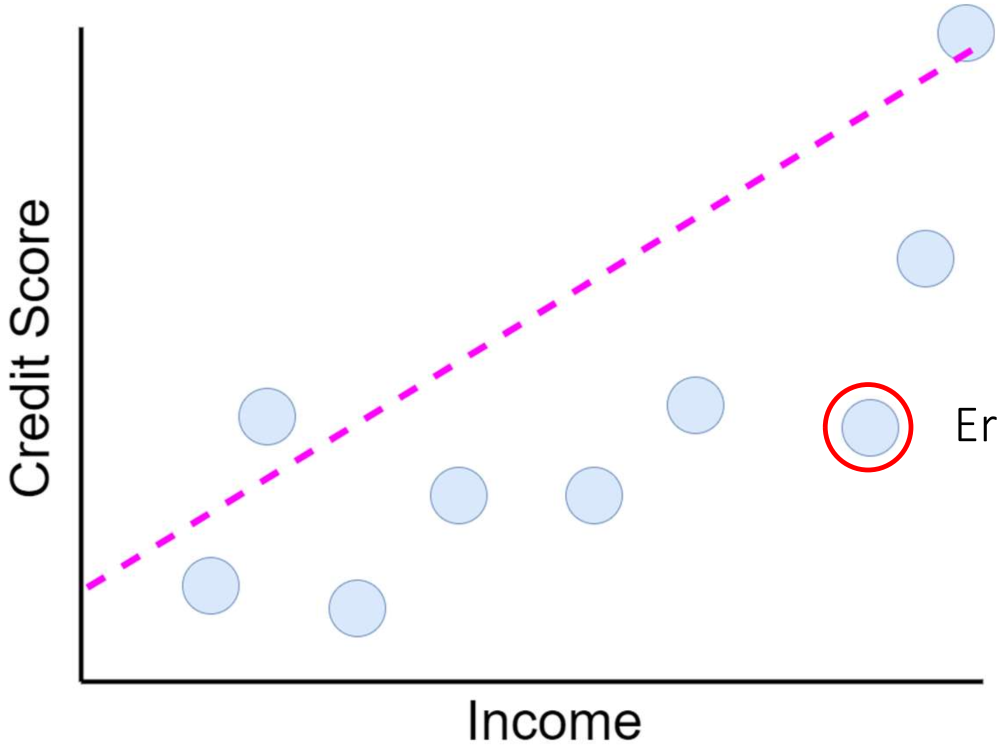
- **Is the metric fit for purpose?**
  - Remember recall vs. precision for single class
    - Multiple classes?
  - What if the task requires a list of predictions?
    - Ranking?
- **Many metrics – important to understand how it relates to task**

What is a good sanity check?



Naïve or simple rule-based model

What is a good sanity check?



Error analysis

What to watch out for

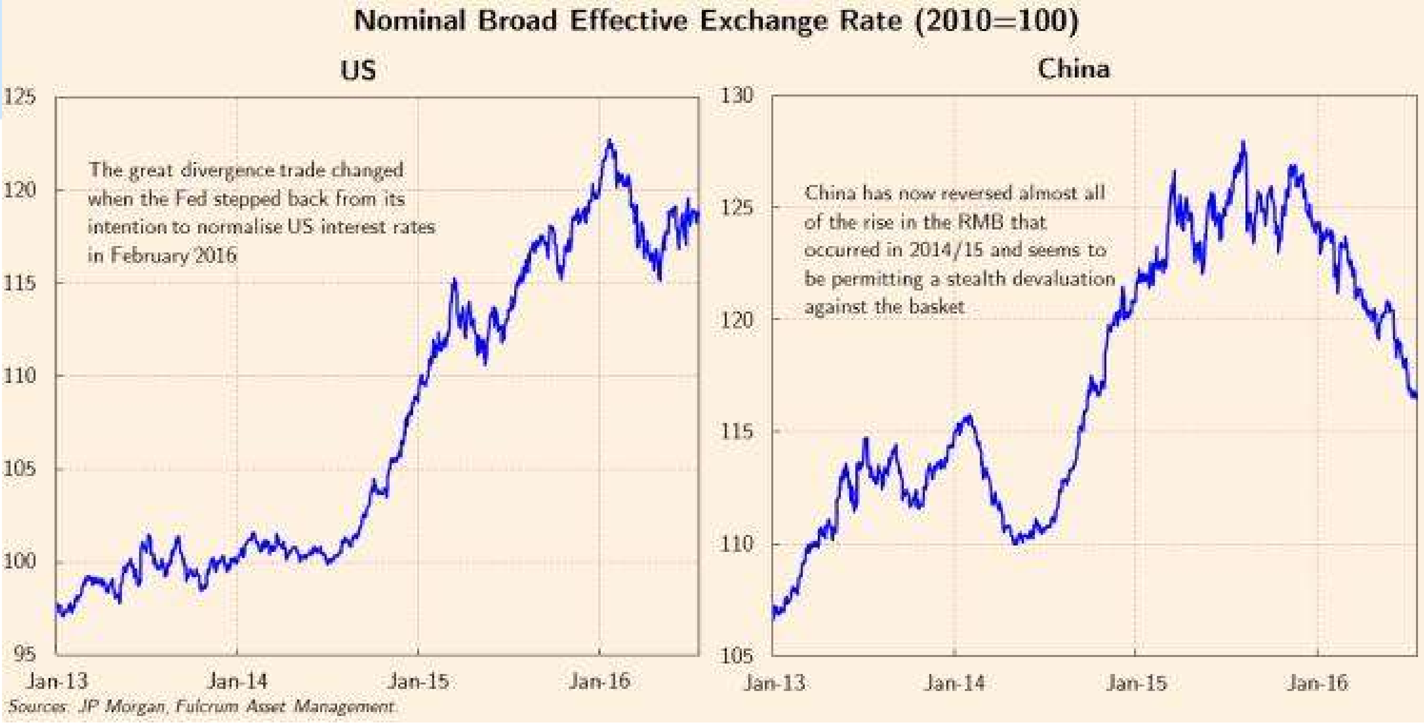
Data	Data quality, bias, leakage
Modelling	Model Selection, Under vs. Overfitting, Explainability and Interpretability
Evaluation	Appropriateness of metrics
Deploy	Drifts

Signal-to Noise & Non-Stationarity

Opinion Gavyn Davies' blog

Regime changes in the financial markets

GAVYN DAVIES + Add to myFT



Nominal Broad Effective Exchange Rate (2010=100)

The great divergence trade changed when the Fed stepped back from its intention to normalise US interest rates in February 2016

China has now reversed almost all of the rise in the RMB that occurred in 2014/15 and seems to be permitting a stealth devaluation against the basket

Sources: JP Morgan, Fulcrum Asset Management.

https://on.ft.com/3z3LU8J

Drift

- **Data drift**: Using credit transaction data before *chip and pin* to train a model for data after *chip and pin*

- **Concept drift**: Using the same model to detect fraud after it becomes known that your model depends on a specific network centrality measure to detect fraud

Recap – Text Processing

- **How do you break down text into units?**
  - **Why?**
- **What needs to be done before a ML or DL model can learn from text data?**
- **What is a simple way to represent text data numerically?**
- **Advantages and disadvantages?**

- *1a. Text processing and visualization*

Recap – Latent Stuff

- **What does ‘latent’ mean in ML or DL?**
- **Why do we learn ‘latent’ information?**
- **What does learning a ‘latent’ representation of text or a word lead to?**
- **How can we use these ‘latent’ representations?**

- *1b. Topic modelling*
- *2. Word vectors*

Recap – NLP Models

- **What are 2 key features of text data we need to recognize?**
- **How does a recurrent neural network deal with such features of text data?**
- **How does a transformer deal with such features of text data?**

- *3. Transformers & RNNs (warning, may be a little complex, just try to get the gist)*

Recap – Pre-Training

- **People frequently use the term 'LLM' when talking about NLP models these days**
  - **What is the first 'L'?**
  - **What is 'LM'?**
- **What is an advantage of using a pre-trained model?**
- **What are 2 ways of using pre-trained models?**
- **Can we use pre-trained models when we have multimodal information?**

- *4. Pre-trained NLP models for different tasks*

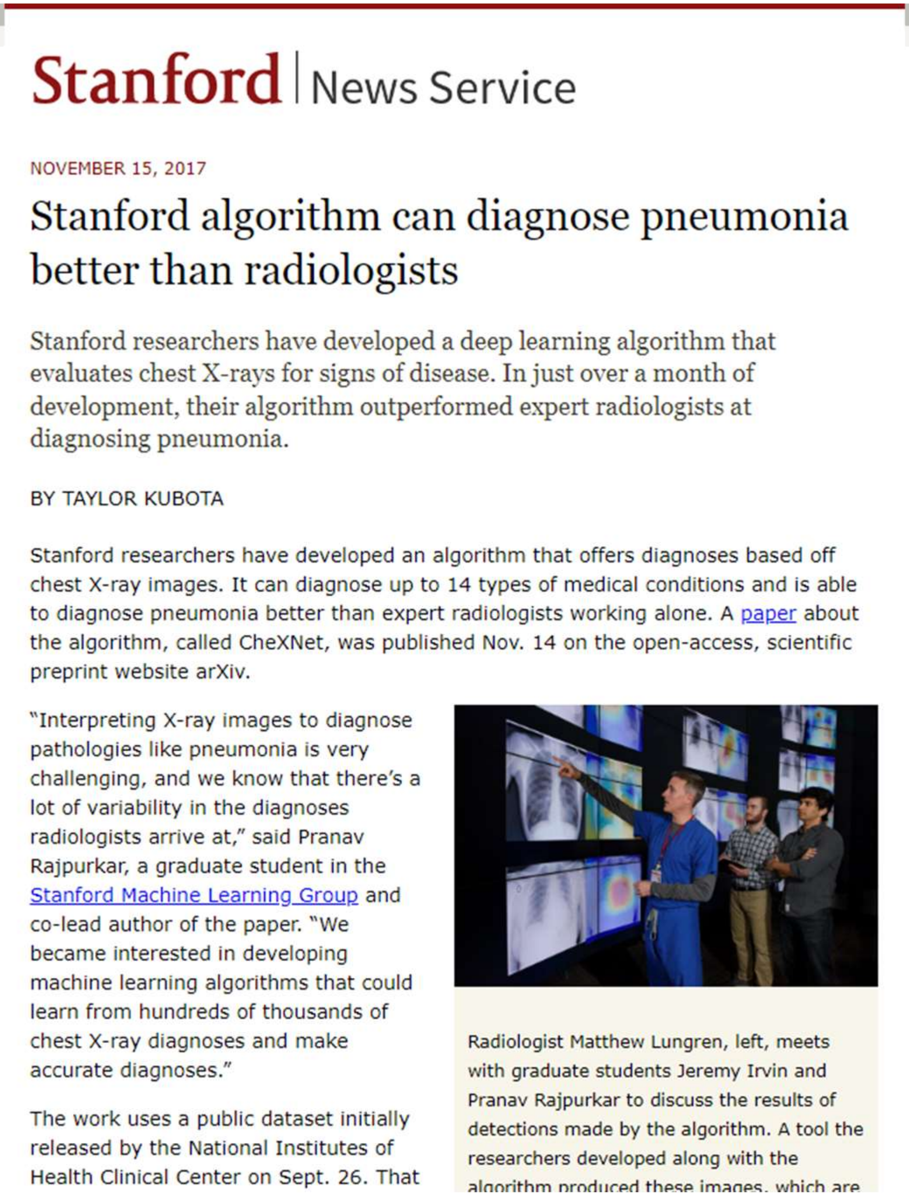
Recap – Model Risks

Data	**Three things to watch out for re. data?**
Modelling	Model Selection, Under vs. Overfitting, Explainability and Interpretability
Evaluation	**Should we just accept it when results look really really good?**
Deploy	**What can cause a model to be less effective over time?**

What to watch out for

Data	Data quality, bias, leakage
Modelling	Model Selection, Under vs. Overfitting, Explainability and Interpretability
Evaluation	Appropriateness of metrics
Deploy	Drifts

Case Study



Stanford | News Service

NOVEMBER 15, 2017

Stanford algorithm can diagnose pneumonia better than radiologists

Stanford researchers have developed a deep learning algorithm that evaluates chest X-rays for signs of disease. In just over a month of development, their algorithm outperformed expert radiologists at diagnosing pneumonia.

BY TAYLOR KUBOTA

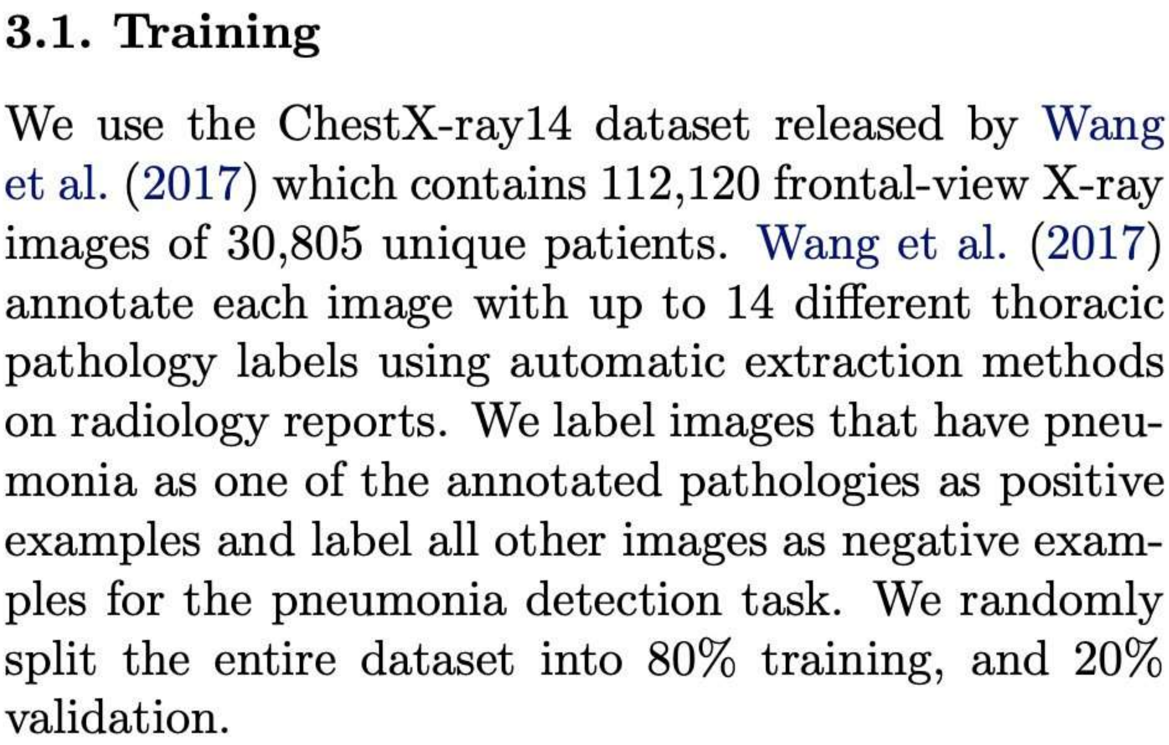
Stanford researchers have developed an algorithm that offers diagnoses based off chest X-ray images. It can diagnose up to 14 types of medical conditions and is able to diagnose pneumonia better than expert radiologists working alone. A paper about the algorithm, called CheXNet, was published Nov. 14 on the open-access, scientific preprint website arXiv.

"Interpreting X-ray images to diagnose pathologies like pneumonia is very challenging, and we know that there's a lot of variability in the diagnoses radiologists arrive at," said Pranav Rajpurkar, a graduate student in the Stanford Machine Learning Group and co-lead author of the paper. "We became interested in developing machine learning algorithms that could learn from hundreds of thousands of chest X-ray diagnoses and make accurate diagnoses."

The work uses a public dataset initially released by the National Institutes of Health Clinical Center on Sept. 26. That

Radiologist Matthew Lungren, left, meets with graduate students Jeremy Irvin and Pranav Rajpurkar to discuss the results of detections made by the algorithm. A tool the researchers developed along with the algorithm produced these images, which are

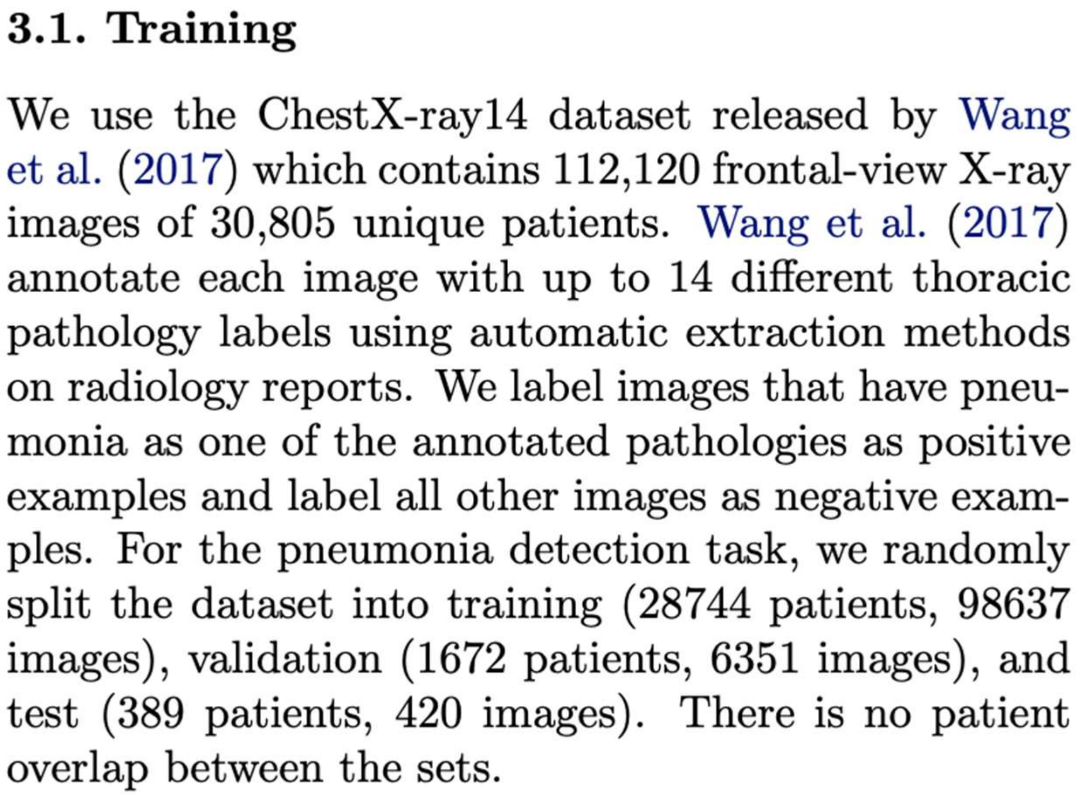
Spot the issue ….



3.1. Training

We use the ChestX-ray14 dataset released by Wang et al. (2017) which contains 112,120 frontal-view X-ray images of 30,805 unique patients. Wang et al. (2017) annotate each image with up to 14 different thoracic pathology labels using automatic extraction methods on radiology reports. We label images that have pneumonia as one of the annotated pathologies as positive examples and label all other images as negative examples for the pneumonia detection task. We randomly split the entire dataset into 80% training, and 20% validation.

Corrected



3.1. Training

We use the ChestX-ray14 dataset released by Wang et al. (2017) which contains 112,120 frontal-view X-ray images of 30,805 unique patients. Wang et al. (2017) annotate each image with up to 14 different thoracic pathology labels using automatic extraction methods on radiology reports. We label images that have pneumonia as one of the annotated pathologies as positive examples and label all other images as negative examples. For the pneumonia detection task, we randomly split the dataset into training (28744 patients, 98637 images), validation (1672 patients, 6351 images), and test (389 patients, 420 images). There is no patient overlap between the sets.

What to watch out for

Data	Data quality, bias, leakage
Modelling	Model Selection, Under vs. Overfitting, Explainability and Interpretability
Evaluation	Appropriateness of metrics
Deploy	Drifts

So far, you have encountered:

Supervised Learning

- Decision trees
- Random Forest
- XGBoost
- K-nearest neighbors
- Linear discriminant analysis
- Linear regression, Logistic regression
- Support vector machines

Unsupervised Learning

- Clustering - K-means
- Isolation Forest
- Dimensionality reduction - Principal component analysis
- Latent Dirichlet Allocation – Topic Modelling

Neural Networks

- Multilayer Perceptron/Dense Neural Network
- Convolutional Neural Network
- Transformer
- Recurrent Neural Network
- Graph Neural Network

Can we say that a model is not suitable for a specific task?

Supervised Learning

- Decision trees
- Random Forest
- XGBoost
- K-nearest neighbors
- Linear discriminant analysis
- Linear regression, Logistic regression
- Support vector machines

Unsupervised Learning

- Clustering - K-means
- Isolation Forest
- Dimensionality reduction - Principal component analysis
- Latent Dirichlet Allocation – Topic Modelling

Neural Networks

- Multilayer Perceptron/Dense Neural Network
- Convolutional Neural Network
- Transformer
- Recurrent Neural Network
- Graph Neural Network

It depends.

- **Structured, i.e. tabular data**: Ensembles of tree-based models (XGBoost, LightGBM, Random Forest) generally work well
    - But lots of ongoing research on neural networks for tabular data, some look promising but early days
    - Feature engineering and hyper-parameter tuning are key for now

It depends.

- **Time series:** Classical time series methods (ARIMA, GARCH) still work well, and trusted
  - But cannot deal with unstructured information
  - Some promising methods in machine/deep learning, but use with care:
    - Zillow's Zestimate led to deep losses probably cause they over-tuned it and trusted it too much
  - Non-stationary distributions means that drift is a key issue for time series models

It depends.

- **Unstructured data, images and text**: Deep learning models dominate
  - RNNs for text, CNNs for images work well
  - But transformers taking centre stage for both modalities in last 2-3 years
  - If you have enough data and a known task, results usually pretty good
- **Multimodal**
  - Still a very open research area
- **Caution: But what I say now may be outdated tomorrow**
  - Field is moving a breakneck pace, pace is many times faster than most other modelling-related fields
- **How then?**

Model Selection: Basic Considerations and Questions

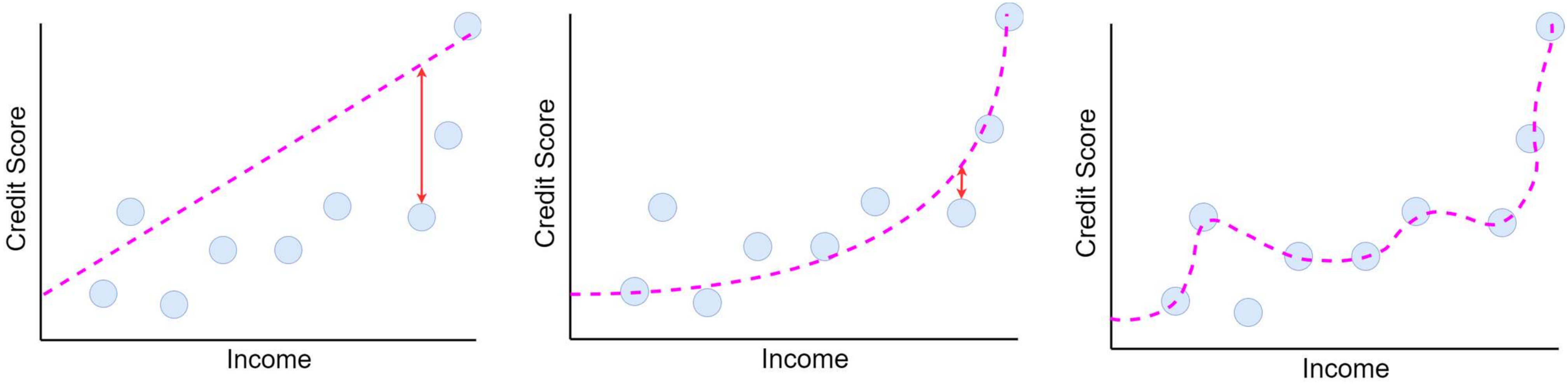
- What are the **unique characteristics** of your data? **(understand your data intimately, if not, rest of steps are futile)**
- What has **worked** for the problem or task? **(frame your task properly)**
    - If problem or task is unique, what is a **similar** problem or task?
- Get a **naïve baseline** **(don't bluff yourself)**
    - Choose a simple model, e.g., simple mean, linear or logistic regression
- Consider a few **challenger models** **(backup plans)**
- **Train and evaluate properly (choose metrics properly, avoid data leakage, error analysis)**
- **Monitor on an ongoing basis (keep a close eye on drifts)**

Balancing between Underfitting vs Overfitting

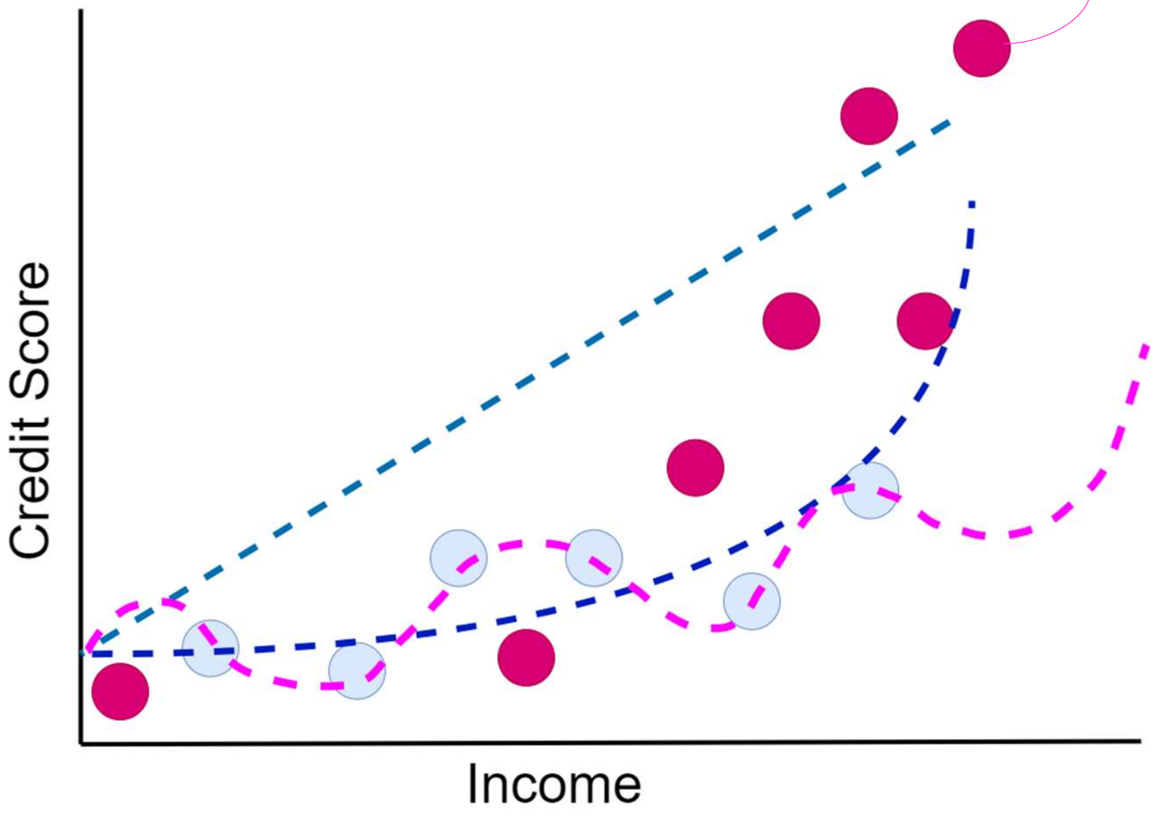
- What happens when you underfit?
- What happens when you overfit?

Model variance and bias

What is the issue with this model?



Model variance and bias



Data not used when training or fitting the model

Model variance and bias

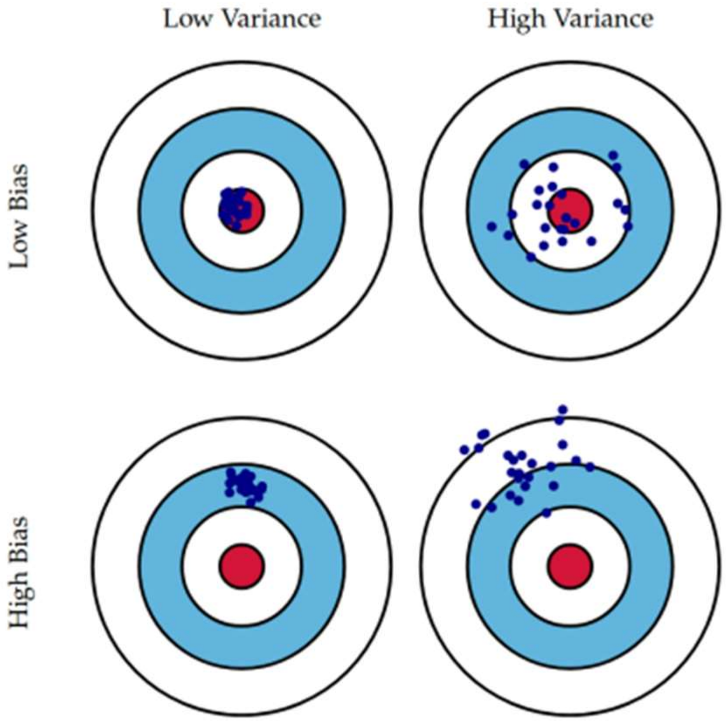
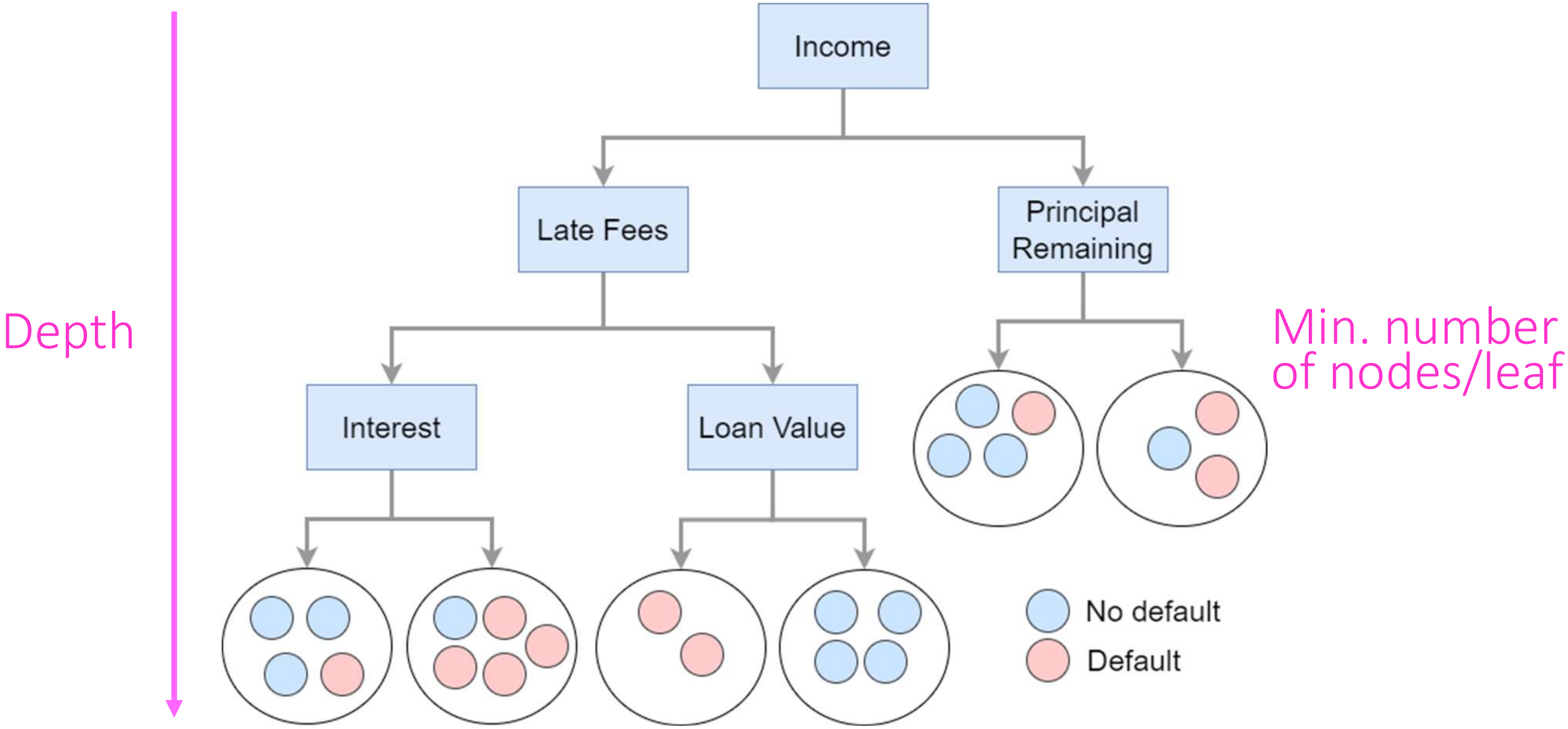
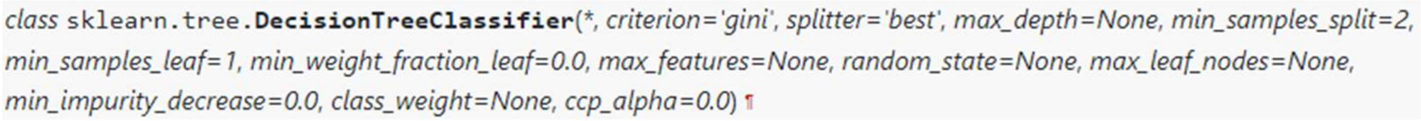


Fig. 1 Graphical illustration of bias and variance.

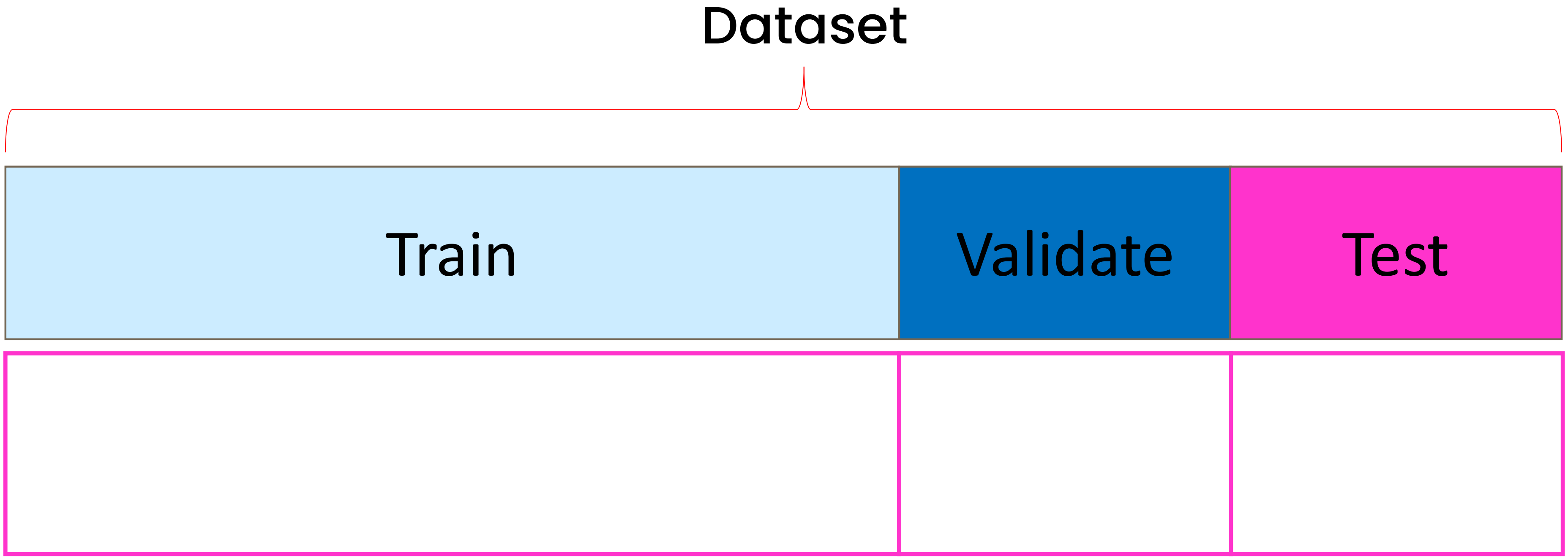
http://scott.fortmann-roe.com/docs/BiasVariance.html

Recall Hyperparameters




class sklearn.tree.**DecisionTreeClassifier**(**, criterion='gini', splitter='best', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features=None, random_state=None, max_leaf_nodes=None, min_impurity_decrease=0.0, class_weight=None, ccp_alpha=0.0*) ¶

Where should you tune hyper-parameters? Why?



Dataset

Train

Validate

Test

Explainability and Interpretability

- An area of active research, even definition of *explainability* vs. *interpretability* is not fixed
  - Why vs. how?

- No easy solutions
  - Performance vs. interpretability
  - Technical user vs lay-man

- Good for trust, fairness, checks on model robustness
  - But …
    - Is it really needed, e.g., impact, well studied areas?
    - Could it have unintended effects, e.g., adversarial attacks?

Explainability and Interpretability

- Even interpreting linear regression is not straightforward
- Why?

$$Y = \boldsymbol{A}X_1X_2X_3 + \boldsymbol{B}X_1X_3 + \boldsymbol{C}X_2X_3 + \boldsymbol{D}X_1^3 + \ \ldots$$

Explainability and Interpretability

- Intrinsic or post-hoc
  - Logistic regression vs. **LIME**
- Model-specific or agnostic
  - Decision trees/random forest vs. **SHAP**
- Local or global
  - **Instance or class**

Explainability and Interpretability

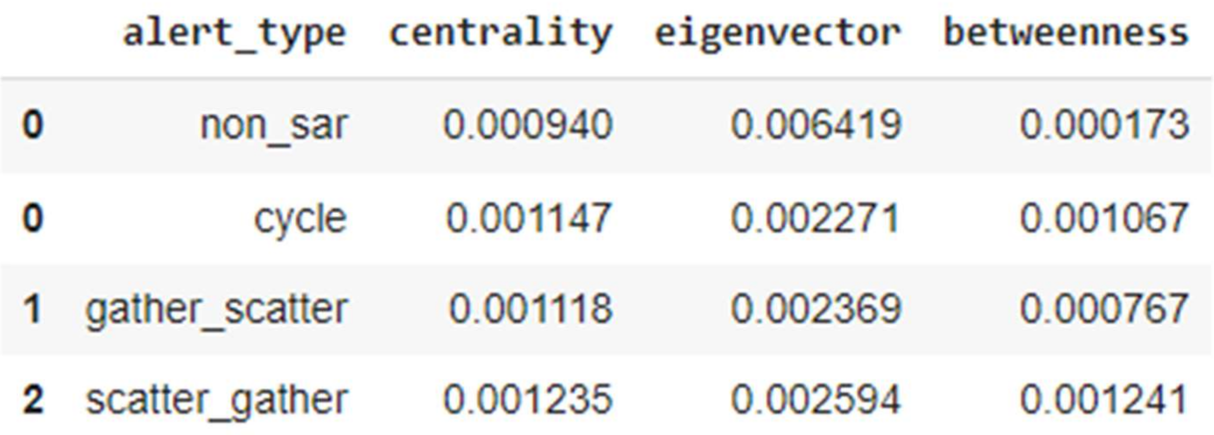
- Prior world views (prior knowledge)
  - Humans typically do not trust predictions far from what we expect, and require more convincing explanations
- Fidelity vs. abstraction
  - Humans cannot deal with a lot of details
- Contrastive and abnormal
  - We like contrasting examples, and things that are very different

Explainability and Interpretability

- Unfortunately, no one ideal method(s) as of now
  - **Exploratory data analysis (EDA)**
  - **Model-specific (use what comes from the model)**
    - Linear, logistic regression, decision trees, random forests
    - Neural networks with attention scores
  - **Model agnostic (treat model as black box and apply another method on black box model)**
    - LIME (Local)
    - SHAP (Global)
  - Others – Counterfactuals (what-if)

EDA

- Recall this from prev. class?
- Why do you think EDA can provide explanations, even without a model?

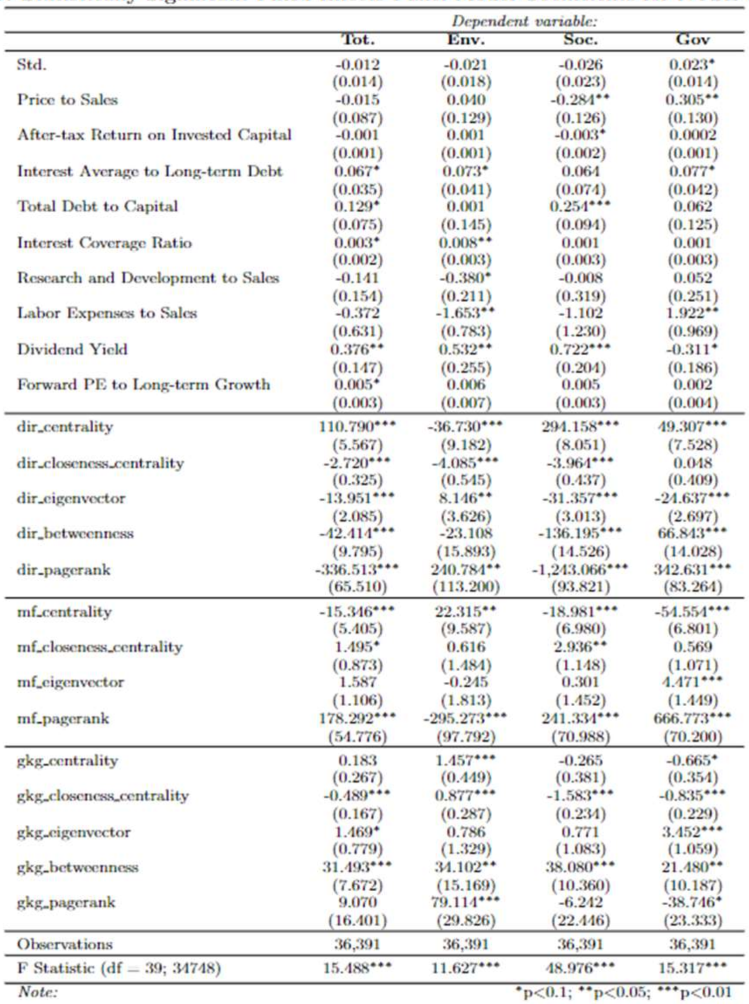


	alert_type	centrality	eigenvector	betweenness
0	non_sar	0.000940	0.006419	0.000173
0	cycle	0.001147	0.002271	0.001067
1	gather_scatter	0.001118	0.002369	0.000767
2	scatter_gather	0.001235	0.002594	0.001241

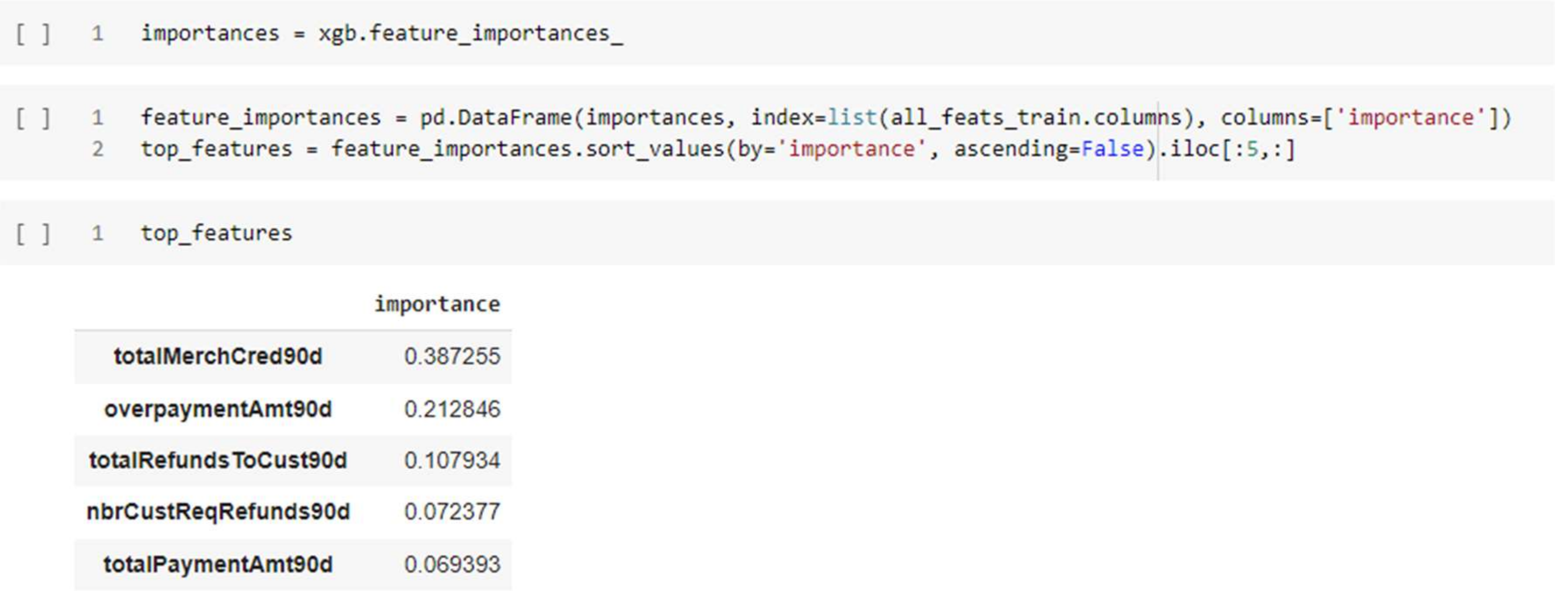
Model-specific

- Stats. from linear regression models
- Feature importances from trees and forests
- Extract attention scores from DL models using attention mechanisms

Table 7: Statistically Significant Fixed Effects Panel Model Coefficients for NYSE dataset



	Dependent variable:			
	Tot.	**Env.**	**Soc.**	**Gov**
Std.	-0.012	-0.021	-0.026	0.023*
	(0.014)	(0.018)	(0.023)	(0.014)
Price to Sales	-0.015	0.040	-0.284**	0.305**
	(0.087)	(0.129)	(0.126)	(0.130)
After-tax Return on Invested Capital	-0.001	0.001	-0.003*	0.0002
	(0.001)	(0.001)	(0.002)	(0.001)
Interest Average to Long-term Debt	0.067*	0.073*	0.064	0.077*
	(0.035)	(0.041)	(0.074)	(0.042)
Total Debt to Capital	0.129*	0.001	0.254***	0.062
	(0.075)	(0.145)	(0.094)	(0.125)
Interest Coverage Ratio	0.003*	0.008**	0.001	0.001
	(0.002)	(0.003)	(0.003)	(0.003)
Research and Development to Sales	-0.141	-0.380*	-0.008	0.052
	(0.154)	(0.211)	(0.319)	(0.251)
Labor Expenses to Sales	-0.372	-1.653**	-1.102	1.922**
	(0.631)	(0.783)	(1.230)	(0.969)
Dividend Yield	0.376**	0.532**	0.722***	-0.311*
	(0.147)	(0.255)	(0.204)	(0.186)
Forward PE to Long-term Growth	0.005*	0.006	0.005	0.002
	(0.003)	(0.007)	(0.003)	(0.004)
dir_centrality	110.790***	-36.730***	294.158***	49.307***
	(5.567)	(9.182)	(8.051)	(7.528)
dir_closeness_centrality	-2.720***	-4.085***	-3.964***	0.048
	(0.325)	(0.545)	(0.437)	(0.409)
dir_eigenvector	-13.951***	8.146**	-31.357***	-24.637***
	(2.085)	(3.626)	(3.013)	(2.697)
dir_betweenness	-42.414***	-23.108	-136.195***	66.843***
	(9.795)	(15.893)	(14.526)	(14.028)
dir_pagerank	-336.513***	240.784**	-1,243.066***	342.631***
	(65.510)	(113.200)	(93.821)	(83.264)
mf_centrality	-15.346***	22.315**	-18.981***	-54.554***
	(5.405)	(9.587)	(6.980)	(6.801)
mf_closeness_centrality	1.495*	0.616	2.936**	0.569
	(0.873)	(1.484)	(1.148)	(1.071)
mf_eigenvector	1.587	-0.245	0.301	4.471***
	(1.106)	(1.813)	(1.452)	(1.449)
mf_pagerank	178.292***	-295.273***	241.334***	666.773***
	(54.776)	(97.792)	(70.988)	(70.200)
gkg_centrality	0.183	1.457***	-0.265	-0.665*
	(0.267)	(0.449)	(0.381)	(0.354)
gkg_closeness_centrality	-0.489***	0.877***	-1.583***	-0.835***
	(0.167)	(0.287)	(0.234)	(0.229)
gkg_eigenvector	1.469*	0.786	0.771	3.452***
	(0.779)	(1.329)	(1.083)	(1.059)
gkg_betweenness	31.493***	34.102**	38.080***	21.480**
	(7.672)	(15.169)	(10.360)	(10.187)
gkg_pagerank	9.070	79.114***	-6.242	-38.746*
	(16.401)	(29.826)	(22.446)	(23.333)
Observations	36,391	36,391	36,391	36,391
F Statistic (df = 39; 34748)	15.488***	11.627***	48.976***	15.317***
Note:		*p<0.1; **p<0.05; ***p<0.01		



```python
importances = xgb.feature_importances_
```

```python
feature_importances = pd.DataFrame(importances, index=list(all_feats_train.columns), columns=['importance'])
top_features = feature_importances.sort_values(by='importance', ascending=False).iloc[:5,:]
```

```python
top_features
```

	importance
totalMerchCred90d	0.387255
overpaymentAmt90d	0.212846
totalRefundsToCust90d	0.107934
nbrCustReqRefunds90d	0.072377
totalPaymentAmt90d	0.069393

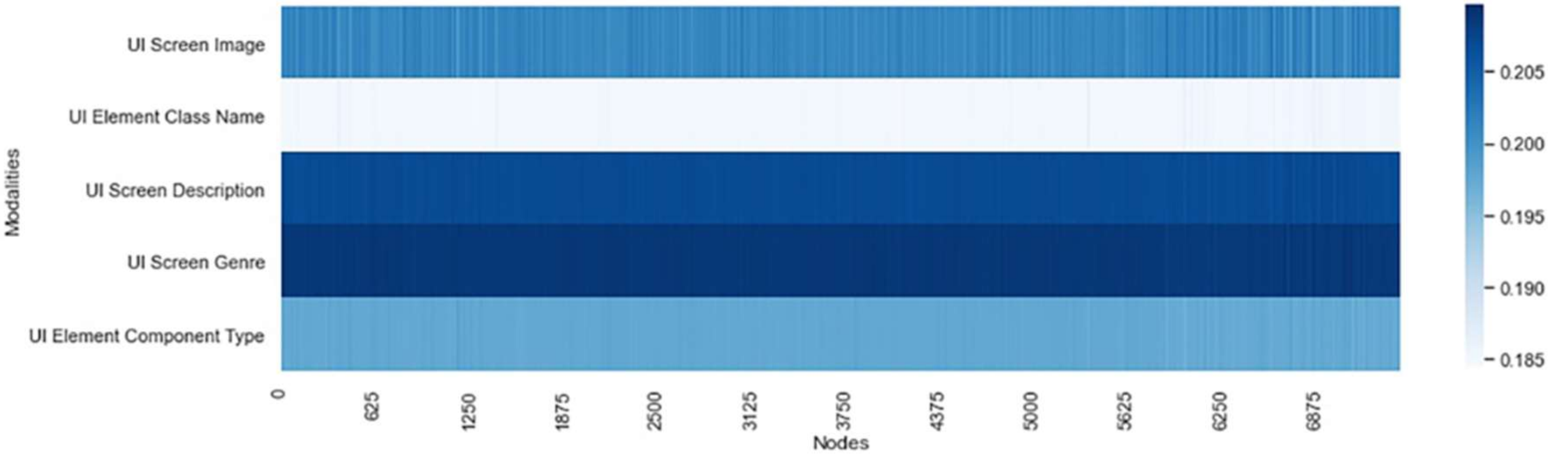
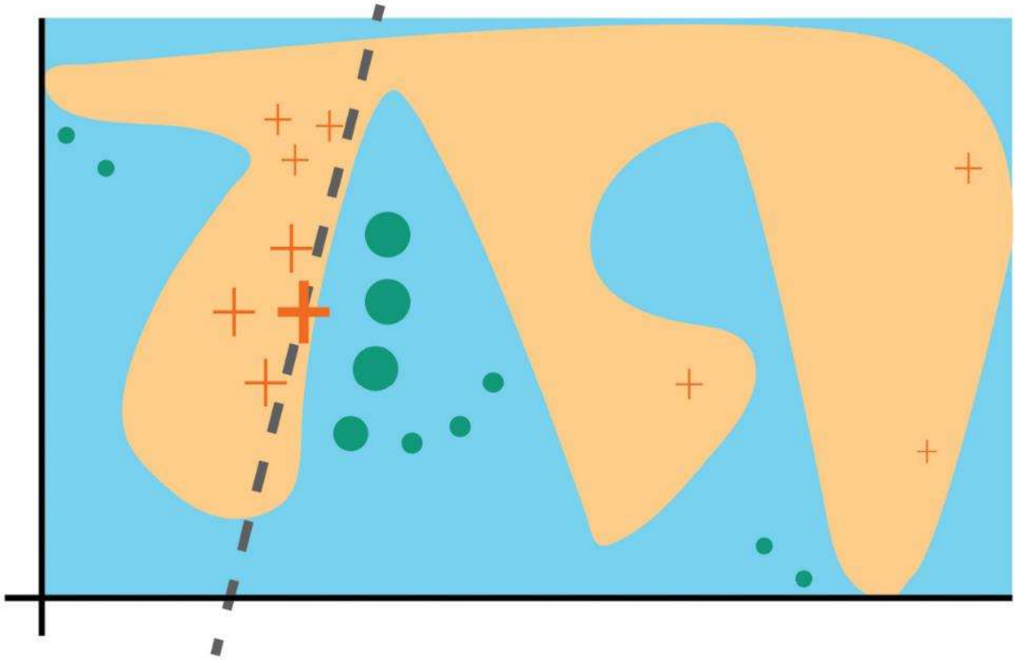


Fig. 11. Visualization of attention weights for EMAAN-A (W-RICO-N).

Model Agnostic – LIME

- Local Interpretable Model-Agnostic Explanations (LIME)



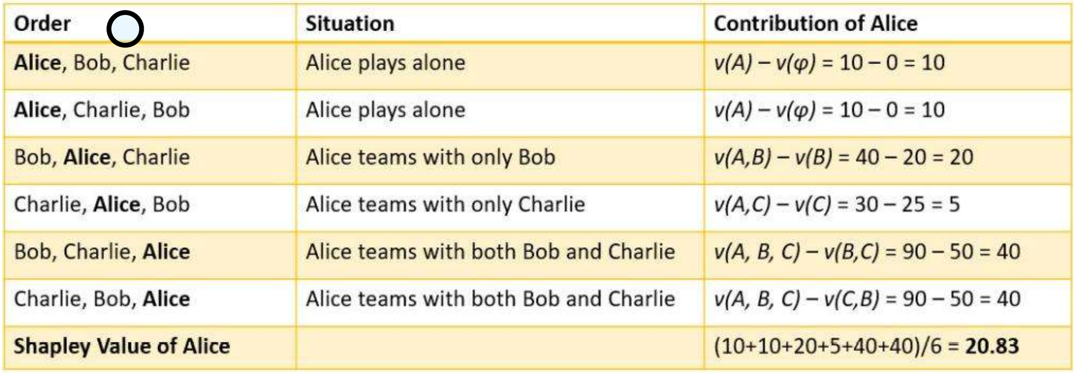
From [Applied Machine Learning Explainability Techniques]

Model Agnostic – SHAP

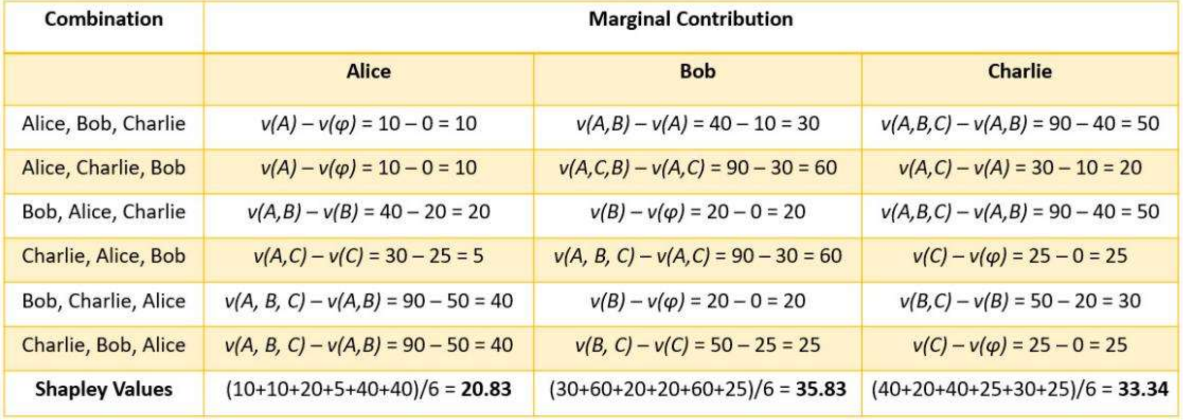
How do I distribute the commission?

- SHapley Additive exPlanation (SHAP)

Completed Sale



Order	Situation	Contribution of Alice
Alice, Bob, Charlie	Alice plays alone	*v(A)* – *v(φ)* = 10 – 0 = 10
Alice, Charlie, Bob	Alice plays alone	*v(A)* – *v(φ)* = 10 – 0 = 10
Bob, **Alice**, Charlie	Alice teams with only Bob	*v(A,B)* – *v(B)* = 40 – 20 = 20
Charlie, **Alice**, Bob	Alice teams with only Charlie	*v(A,C)* – *v(C)* = 30 – 25 = 5
Bob, Charlie, **Alice**	Alice teams with both Bob and Charlie	*v(A, B, C)* – *v(B,C)* = 90 – 50 = 40
Charlie, Bob, **Alice**	Alice teams with both Bob and Charlie	*v(A, B, C)* – *v(C,B)* = 90 – 50 = 40
Shapley Value of Alice		(10+10+20+5+40+40)/6 = **20.83**



Combination	**Marginal Contribution**		
	Alice	**Bob**	**Charlie**
Alice, Bob, Charlie	*v(A)* – *v(φ)* = 10 – 0 = 10	*v(A,B)* – *v(A)* = 40 – 10 = 30	*v(A,B,C)* – *v(A,B)* = 90 – 40 = 50
Alice, Charlie, Bob	*v(A)* – *v(φ)* = 10 – 0 = 10	*v(A,C,B)* – *v(A,C)* = 90 – 30 = 60	*v(A,C)* – *v(A)* = 30 – 10 = 20
Bob, Alice, Charlie	*v(A,B)* – *v(B)* = 40 – 20 = 20	*v(B)* – *v(φ)* = 20 – 0 = 20	*v(A,B,C)* – *v(A,B)* = 90 – 40 = 50
Charlie, Alice, Bob	*v(A,C)* – *v(C)* = 30 – 25 = 5	*v(A, B, C)* – *v(A,C)* = 90 – 30 = 60	*v(C)* – *v(φ)* = 25 – 0 = 25
Bob, Charlie, Alice	*v(A, B, C)* – *v(A,B)* = 90 – 50 = 40	*v(B)* – *v(φ)* = 20 – 0 = 20	*v(B,C)* – *v(B)* = 50 – 20 = 30
Charlie, Bob, Alice	*v(A, B, C)* – *v(A,B)* = 90 – 50 = 40	*v(B, C)* – *v(C)* = 50 – 25 = 25	*v(C)* – *v(φ)* = 25 – 0 = 25
Shapley Values	(10+10+20+5+40+40)/6 = **20.83**	(30+60+20+20+60+25)/6 = **35.83**	(40+20+40+25+30+25)/6 = **33.34**

From Applied Machine Learning Explainability Techniques

Explainability and Interpretability

- Still open research area!
- But even simple feature permutation importance can be very useful in understanding any model

- ***Notebook 5. Explainability and Interpretability (XAI)***
  - *Feature importance*
  - *LIME*
  - *SHAP*

Resources

- Good book on deep learning - https://d2l.ai
- Lots of pre-trained models (NLP, computer vision etc) to play with - https://huggingface.co
- Good book on interpretability (but in R) - https://christophm.github.io/interpretable-ml-book/
- Good book on interpretability in Python - Applied Machine Learning Explainability Techniques