Language Models

A *Language Model* is an instance of the "predict the next" paradigm where

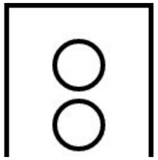
- given a sequence of words
- we try to predict the next word

Recall the architecture to solve "predict the next word" and data preparation

Language Modeling task

Architecture	Data preparation

$\mathbf{y}_{(t)}$



$\mathbf{s} = \mathbf{s}_{(1)}, \ldots, \mathbf{s}_{(T)}$

The raw data

- e.g., the sequence of words $\mathbf{s} = \mathbf{s}_{(1)}, \ldots \mathbf{s}_{(\bar{T})}$

is not naturally labeled.

We need a Data Preparation step to create examples

$$\langle \mathbf{x^{(i)}}, \mathbf{y^{(i)}} \rangle = \langle \mathbf{s}_{(1)}, \ldots \mathbf{s}_{(i)}, \mathbf{s}_{(i+1)} \rangle$$

to create labelled examples.

We have called this method of turning unlabeled data into labeled examples: *Semi-Supervised* Learning.

In the NLP literature, it is called *Unsupervised Learning*.

There are abundant sources of raw text data

- news, books, blogs, Wikipedia
- not all of the same quality

The large number of examples that can be generated facilitates the training of models with very large number of weights.

This is extremely expensive but, fortunately, the results can be re-used.

- Someone with abundant resources trains a Language Model on a broad domain
- Publishes the architecture and weights
- Others re-use

Models that have been trained with the intent that they be re-used are called *Pre-Trained* models.

The process of creating such a Language Models from unlabeled raw text is referred to as

> *Unsupervised Pre-Training: Train a model on a* ***very large*** *number of examples from a* ***broad*** *domain*

Using a Pre-Trained Language Model

Feature based

Consider the behavior of a Language Model as it processes a word sequence (either an RNN or Encoder Transformer).

It produces an output (or latent state) $\bar{\mathbf{h}}_{(t)}$ for each position t of the sequence.

This is a *context sensitive* representation specific to input word $\mathbf{s}_t p$ at position t.

- context sensitive because it depends on
  - prefix $\mathbf{s}_{(1)}, \ldots, \mathbf{s}_{(t-1)}$
  - entire sequence $\mathbf{s}_{(1)}, \ldots, \mathbf{s}_{(\bar{T})}$

These Context Sensitive Representations of words may be useful representations for down-stream tasks

- Better than Word Embeddings, which have no context
- See the [ELMo paper (https://arxiv.org/abs/1802.05365)](https://arxiv.org/abs/1802.05365)

Fine-Tuning

Logically, we use the process that we described as Transfer Learning

- where we use the output of some layer of the Pre-Trained model
    - default: all layers, excluding the Classification Head
- as a "meaningful" **fixed length** representation of input sequence $\mathbf{x}^{(\mathbf{i})}_{(1)}, \ldots, \mathbf{x}^{(\mathbf{i})}_{(m)}$
- which is then fed to a Classification head with the object of matching the target $\mathbf{y}^{(\mathbf{i})}$

Recall the diagram from our module on [Transfer Learning (Transfer_Learning.ipynb)](Transfer_Learning.ipynb)

The process is

- Import the Pre-Trained model (which was trained on a large number c
  from a broad domain)
- Fine-Tune the weights using a **small** number of examples for a **specific**
  narrow domain.

Often, the specific task is Supervised (e.g., sentiment analysis).

Example: Using a Pre-trained Language Model to analyze sentimer

This is a straight-forward application of Transfer Learning

- Replace the Classification Head used for Language Modeling
    - e.g., a head that generated a probability distribution over wo
      vocabulary
- By an un-trained Binary Classification head (Positive/Negative sentim
- Train on examples. Pairs of
    - sentence

Language Models: the future (present ?) of N

Pre-trained Language Models (especially the *Large Language Models* that ha
massive amounts of data) seem to transfer well to other tasks via Supervis

We call this paradigm "Unsupervised Pre-Trained Model + Supervised Fine

This paradigm means that we might not need to create a new model for a r

Instead: we transform our task into one amenable to the "Unsupervised Pr
Model + Supervised Fine-Tuning" paradigm

- using a Language Model as our Pre-Trained Model

Input Transformations

One impediment to using the paradigm is that

- the task-specific input
- is not the simple, unstructured sequence of words that characterize
  Language Modeling.

We need to apply *input transformations*

- to transform structured task-specific input
- to the unstructured sequence of words used in the Language Mode

Here are common examples of tasks with structured inputs:

- Entailment

    - Input is a *pair* of text sequences $[\text{Premise}, \text{Hypothesis}]$

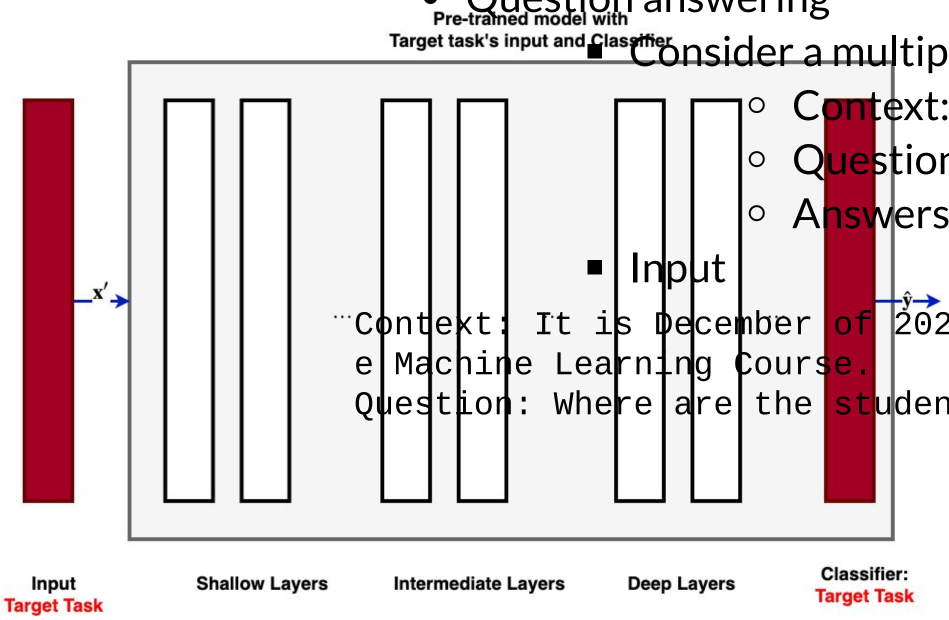
    - Binary classification: Does the Hypothesis Logically follow
    Premise ?

```
Premise: The students are attending a lecture on Mac
g
```

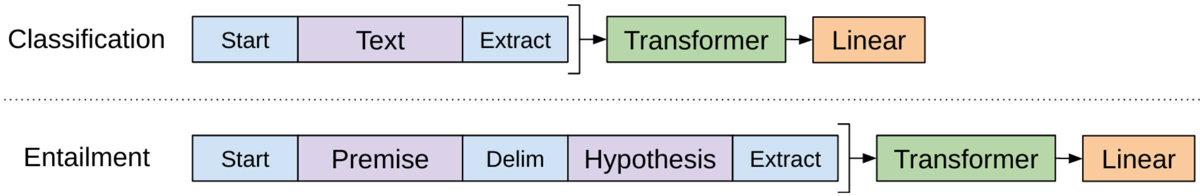
Transfer Learning: replace the head of the pre-trained model

- Question answering
  - Consider a multiple choice questions consisting of
    - Context: a sentence or paragraph stating facts
    - Question
    - Answers: a set of possible answer sentences
  - Input

```
Context: It is December of 2022.  Prof. Perry is teaching the secon
e Machine Learning Course.
Question: Where are the students ?
```



GPT: Task encoding



For example: for multiple choice questions answering

- Create a triple for each answer,
  - [Context, Question, Answer 1],
  - [Context, Question, Answer 2], ...
- Obtain a representation of each triple using a LM
  - using Delimiter tokens to separate elements of the triple
- Fine-tune using a new Multinomial classifier head
  - to obtain probability distribution over answers

- Similarity
  - Input is a *pair* (or more) of text sequences $[\text{First}, \text{Second}, \text{Third}]$
  - Binary/Multinomial classification: Probability that other ser similar to First ?

```
First: Machine Learning is easy not hard
Second: Machine Learning is not difficult
```

Conclusion

Language Models are the basis for the paradigm of Unsupervised Pre-training + Supervised Fine-Tuning.

This has become the dominant paradigm in NLP.

The ability to train Large Language models stem is due, in part, to the advantages of the Transformer

- Execution Parallelism: can run larger models than an RNN for the same amount of elapsed time
- This also facilitates the use of extremely large training datasets.

To use the Pre-Trained LM + Fine-Tuning approach

- we need to convert the structured input into simple sequences.

See [this paper (https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf)](https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf) for some transformations