Language Models

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

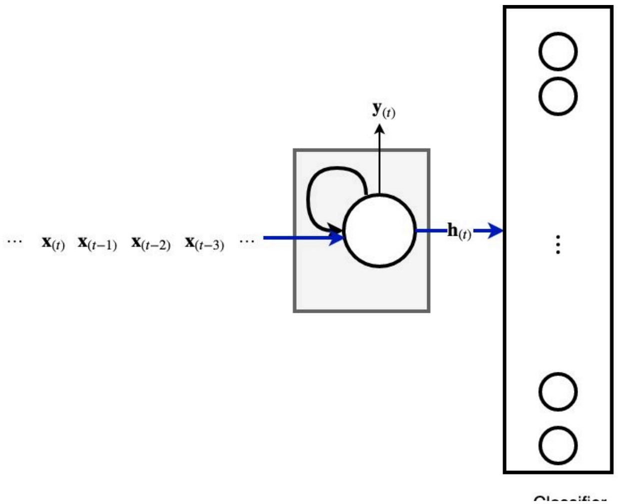
- given a sequence of tokens
- we try to predict the next token

$$p(\mathbf{y}_{(t)} \,|\, \mathbf{y}_{(1:t-1)})$$

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

Language Modeling task

Architecture



Data preparation

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

i	$\mathbf{x}^{(\mathbf{i})}$	$\mathbf{y}^{(\mathbf{i})}$
1	$\mathbf{y}_{(0)}$	$\mathbf{y}_{(1)}$
2	$\mathbf{y}_{([0:1])}$	$\mathbf{y}_{(2)}$
$\vdots$		
t	$\mathbf{y}_{([1:t-1])}$	$\mathbf{y}_{(t)}$
$\vdots$		
T	$\mathbf{y}_{([1:T-1])}$	$\mathbf{y}_{(T)}$

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 labeled example i

$$\begin{aligned}
\mathbf{x^{(i)}} &= \mathbf{s}_{(1)}, \ldots \mathbf{s}_{(i)} \\
\mathbf{y^{(i)}} &= \mathbf{s}_{(i+1)}
\end{aligned}$$

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

Predict the next ? Really: predict the *distribution* of next

We have casually defined the Language Modeling objective as predicting the next token.

As you can see: the head layer is a Classifier

- produces a probability for *each token* in the vocabulary as being the next
$$p(\mathbf{y}_{(t)} \mid \mathbf{y}_{(1:t-1)})$$
- We choose one token by sampling from this probability distribution
    - Greedy sampling: always chose the token with highest probability
    - Non-greedy sampling

The Masked Language Modeling objective

There is a variation on the Language Modeling objective called the *Masked Language Modeling* objective.

- Language Modeling objective: given $s[1:t-1]$, predict $s[t]$
- Masked Language Modeling objective
    - Given $s[1:t]$
    - Randomly chose an index $1 \leq j \leq t$
    - "Mask" token j by replacing it with <MASK> so that the input becomes
    $$s_{(0)}, \ldots s_{(j-1)}, <\mathrm{MASK}>, s_{(j+1)}, \ldots s_{(t)}$$
    - Predict the value behing the mask, e.g., $s_{(j)}$
- The Language Modeling objective is the special case where $j = t$

Unsupervised Pre-Training + Supervised Fine-Tuning (Transfer Learning)

How do we adapt a Language Model to solve other Target tasks ?

The obvious answer is via Transfer Learning

- The Language Model has learned a lot about the nature of language
    - perhaps the language-knowledge can be transfered to a new task
- Replace the "head" that predicts the next token
- With a new task-specific head
- Train the new model on labeled examples from the Target task
    - the task-specific head **must** be trained
    - the language-model weights **can** (but don't have to) be adapted

This paradigm is called *Unsupervised Pre-Traininng + Supervised Fine-Tuning*.

Example: Fine-Tuning a Pre-trained Language Model sentiment

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 v vocabulary
- By an untrained Binary Classification head (Positive/Negative senti
- Train on examples. Pairs of
    - sentence
    - label: Positive/Negative

Other uses of a Language Model: Feature ba
Transfer Learning

We can generalize the procedure of "replacing the head": re-use the featu
by the Source model.

Let $f_\Theta(\mathbf{x})$ denote the function computed by the Source Language Model
sequence $\mathbf{x}$

- the output of the layer **before the final Classification layer** that tra
into the token Vocabulary
- the Source model is parameterized by Θ

Feature based Transfer Learning computes

$$g_\Phi(f_\Theta(\mathbf{x}))$$

for the function g (parameterized by Φ) computed by a new NN.

That is

Using the final representation

The final representation of some models may be useful in surprising ways.

For example

- consider the final representation created by an Encoder Transform
  Language Modeling task
- it is a *sequence* (of length $\bar{T}$, where the input is also a sequence of le
    - a "context sensitive" representation of each element in the
      sequence

Many tasks that use an Encoder modify the original input sequence

- by bracketing it with special tokens `<START>, <END>`

The context sensitive representation of the `<START>, <END>` tokens

Two interesting uses of this fixed-length summary

- Classification of sequences: e.g., Sentiment Analysis
- Semantic Search
    - "Google"-like search
    - encode your query as a fixed length vector
    - encode each document in a collection as a fixed length vect
    - -query-retrieval: return the document whose vector is clos the query

Multi-task learning

One area of recent interest is *multi-task learning*

- Training a model to implement multiple tasks

A model that implements a single task computes

$$p(\text{output} \mid \text{input})$$

A model that implements several tasks computes

$$p(\text{output} \mid \text{input, task-id })$$