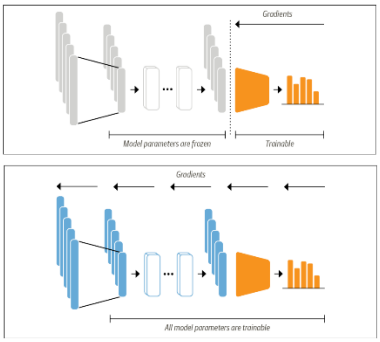
Principais contribuições do artigo “BERT: pre-training of deep bidirectional transformers for language understanding”, de Jacob Devlin et al

Aluno: Leandro Carísio Fernandes

- Há duas estratégias para aplicar modelos de linguagem pré-treinados para tarefas específicas: *feature-based* e *fine-tuning*. No primeiro caso, o modelo de linguagem fica travado e o estado da rede (os pesos da última camada associado ao token [CLS] por exemplo) serve de entrada para uma outra rede (específica para o problema que será resolvido), que é treinada. No segundo caso, os pesos do próprio modelo de linguagem também são ajustados. A fig. abaixo foi retirada do livro “NLP with Transformers”.



- Os autores comentam que a maior limitação dos modelos de linguagem é que são unidirecionais (da esquerda para a direita) e que essa restrição é sub ótima para tarefas que envolvem toda a sentença.
- Propõem o BERT, que remove essa restrição unidirecional usando um “masked language model”. No treinamento, mascaram aleatoriamente uma palavra da sentença. O objetivo do treinamento é adivinhar a palavra escondida (o token id). Tanto o contexto da esquerda como o da direita são usados (bidirecional). Além disso, consideram a tarefa de predição da próxima frase. => essa tarefa de mascaramento também é chamada, na literatura, de *Cloze task*.
- No treinamento do BERT, há duas fases: pré-treino e fine-tuning. No primeiro caso, o modelo é treinado com dados não rotulados. Na fase de fine-tuning, o modelo é inicializado com os parâmetros da fase de pré-treino e todos os parâmetros são treinados com dados rotulados para executar a tarefa específica (downstream task – por exemplos, question answering, classificação, reconhecimento de entidades etc).
- A arquitetura básica do modelo é a mesma independentemente da tarefa. O que muda de acordo com a tarefa é só o final do modelo (cabeça).
- Layers (transformer blocks) = L, Hidden size = H, número de cabeças de auto-atenção = A.
    - $BERT_{BASE}$ => L = 12; H = 768; A = 12; Total de parâmetros = 100 milhões
    - $BERT_{LARGE}$ => L = 24; H = 1024; A = 16; Total de parâmetros = 340 milhões
- Usam o WordPiece embeddings (vocabulário: 30.000 palavras). Cada sequência começa com um token especial [CLS]. O estado da camada oculta final associado a este token é usado para tarefas de classificação.
- A entrada para cada token é feita somando o token com um token de segmento e um token de posição.
- Dados de pré-treino: BooksCorpus (800M palavras) e English Wikipedia (2.500M palavras). É necessário usar corpus de documentos, para ter contextos maiores.