Networks

Networks are combinations of `tf.keras` layers (and possibly other networks). They are `tf.keras` models that would not be trained alone. It encapsulates common network structures like a transformer encoder into an easily handled object with a standardized configuration.

- `BertEncoder` implements a bi-directional Transformer-based encoder as described in "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". It includes the embedding lookups, transformer layers and pooling layer.

- `AlbertEncoder` implements a Transformer-encoder described in the paper ["ALBERT: A Lite BERT for Self-supervised Learning of Language Representations"] (https://arxiv.org/abs/1909.11942). Compared with BERT, ALBERT refactorizes embedding parameters into two smaller matrices and shares parameters across layers.

- `MobileBERTEncoder` implements the MobileBERT network described in the paper "MobileBERT: a Compact Task-Agnostic BERT for Resource-Limited Devices".

- `Classification` contains a single hidden layer, and is intended for use as a classification or regression (if number of classes is set to 1) head.

- `PackedSequenceEmbedding` implements an embedding network that supports packed sequences and position ids.

- `SpanLabeling` implements a single-span labeler (that is, a prediction head that can predict one start and end index per batch item) based on a single dense hidden layer. It can be used in the SQuAD task.

- `XLNetBase` implements the base network used in "XLNet: Generalized Autoregressive Pretraining for Language Understanding" (https://arxiv.org/abs/1906.08237). It includes embedding lookups, relative position encodings, mask computations, segment matrix computations and Transformer XL layers using one or two stream relative self-attention.