Layers

Layers are the fundamental building blocks for NLP models. They can be used to assemble new `tf.keras` layers or models.

- MultiHeadAttention implements an optionally masked attention between query, key, value tensors as described in "Attention Is All You Need". If `from_tensor` and `to_tensor` are the same, then this is self-attention.

- BigBirdAttention implements a sparse attention mechanism that reduces this quadratic dependency to linear described in "Big Bird: Transformers for Longer Sequences".

- CachedAttention implements an attention layer with cache used for auto-agressive decoding.

- KernelAttention implements a group of attention mechansim that express the self-attention as a linear dot-product of kernel feature maps and make use of the associativity property of matrix products to reduce the complexity from quadratic to linear. The implementation includes methods described in "Transformers are RNNs: Fast Autoregressive Transformers with Linear Attention", "Rethinking Attention with Performers", "Random Feature Attention".

- MatMulWithMargin implements a matrix multiplication with margin layer used for training retrieval / ranking tasks, as described in "Improving Multilingual Sentence Embedding using Bi-directional Dual Encoder with Additive Margin Softmax".

- MultiChannelAttention implements an variant of multi-head attention which can be used to merge multiple streams for cross-attentions.

- TalkingHeadsAttention implements the talking heads attention, as decribed in "Talking-Heads Attention".

- Transformer implements an optionally masked transformer as described in "Attention Is All You Need".

- TransformerDecoderBlock TransformerDecoderBlock is made up of self multi-head attention, cross multi-head attention and feedforward network.

- RandomFeatureGaussianProcess implements random feature-based Gaussian process described in "Random Features for Large-Scale Kernel Machines".

- ReuseMultiHeadAttention supports passing attention scores to be reused and avoid recomputation described in "Leveraging redundancy in attention with Reuse Transformers".

- ReuseTransformer supports reusing attention scores from lower layers in higher layers to avoid recomputing attention scores described in "Leveraging redundancy in attention with Reuse Transformers".

- ReZeroTransformer implements Transformer with ReZero described in "ReZero is All You Need: Fast Convergence at Large Depth".

- OnDeviceEmbedding implements efficient embedding lookups designed for TPU-based models.

- PositionalEmbedding creates a positional embedding as described in "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding".

- SelfAttentionMask creates a 3D attention mask from a 2D tensor mask.

- SpectralNormalization implements a tf.Wrapper that applies spectral normalization regularization to a given layer. See Spectral Norm Regularization for Improving the Generalizability of Deep Learning

- MaskedSoftmax implements a softmax with an optional masking input. If no mask is provided to this layer, it performs a standard softmax; however, if a mask tensor is applied (which should be 1 in positions where the data should be allowed through, and 0 where the data should be masked), the output will have masked positions set to approximately zero.

- `MaskedLM` implements a masked language model. It assumes the embedding table variable is passed to it.

- ClassificationHead A pooling head over a sequence of embeddings, commonly used by classification tasks.

- GaussianProcessClassificationHead A spectral-normalized neural Gaussian process (SNGP)-based classification head as described in "Simple and Principled Uncertainty Estimation with Deterministic Deep Learning via Distance Awareness".

- GatedFeedforward implements the gated linear layer feedforward as described in "GLU Variants Improve Transformer".

- MultiHeadRelativeAttention implements a variant of multi-head attention with support for relative position encodings as described in "Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context". This also has extended support for segment-based attention, a re-parameterization introduced in "XLNet: Generalized Autoregressive Pretraining for Language Understanding".

- TwoStreamRelativeAttention implements a variant of multi-head relative attention as described in ["XLNet: Generalized Autoregressive Pretraining for Language Understanding"] (https://arxiv.org/abs/1906.08237). This takes in a query and content stream and applies self attention.

- TransformerXL implements Transformer XL introduced in ["Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context"] (https://arxiv.org/abs/1901.02860). This contains `TransformerXLBlock`, a block containing either one or two stream relative self-attention as well as subsequent feedforward networks. It also contains `TransformerXL`, which contains attention biases as well as multiple `TransformerXLBlocks`.

- MobileBertEmbedding and MobileBertTransformer implement the embedding layer and also transformer layer proposed in the MobileBERT paper.

- BertPackInputs and BertTokenizer and SentencepieceTokenizer implements the layer to tokenize raw text and pack them into the inputs for BERT models.

- TransformerEncoderBlock implements an optionally masked transformer as described in "Attention Is All You Need".