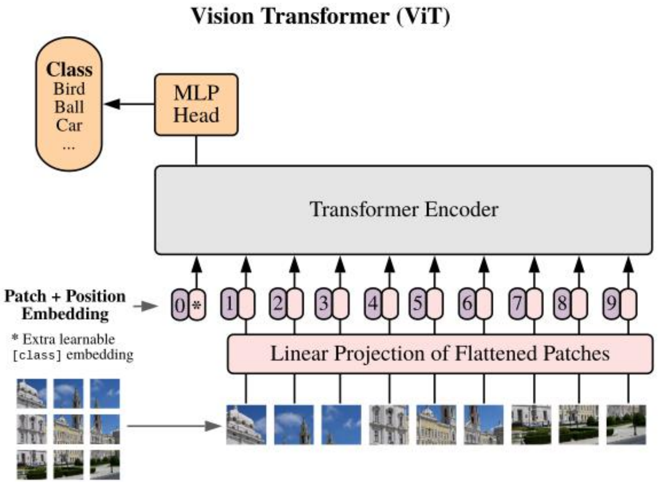
Vision Transformer

AN IMAGE IS WORTH 16X16 WORDS:
TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE

Alexey Dosovitskiy[*,†], Lucas Beyer[*], Alexander Kolesnikov[*], Dirk Weissenborn[*],
Xiaohua Zhai[*], Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer,
Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby[*,†]
[*]equal technical contribution, [†]equal advising
Google Research, Brain Team
{adosovitskiy, neilhoulsby}@google.com

2020 CVPR



原文链接：https://arxiv.org/abs/2010.11929
博文链接：https://blog.csdn.net/qq_37541097/article/details/118242600

Vision Transformer

	Ours-JFT (ViT-H/14)	Ours-JFT (ViT-L/16)	Ours-I21K (ViT-L/16)	BiT-L (ResNet152x4)	Noisy Student (EfficientNet-L2)
ImageNet	**88.55** ± 0.04	87.76 ± 0.03	85.30 ± 0.02	87.54 ± 0.02	88.4/88.5*
ImageNet ReaL	**90.72** ± 0.05	90.54 ± 0.03	88.62 ± 0.05	90.54	90.55
CIFAR-10	**99.50** ± 0.06	99.42 ± 0.03	99.15 ± 0.03	99.37 ± 0.06	–
CIFAR-100	**94.55** ± 0.04	93.90 ± 0.05	93.25 ± 0.05	93.51 ± 0.08	–
Oxford-IIIT Pets	**97.56** ± 0.03	97.32 ± 0.11	94.67 ± 0.15	96.62 ± 0.23	–
Oxford Flowers-102	99.68 ± 0.02	**99.74** ± 0.00	99.61 ± 0.02	99.63 ± 0.03	–
VTAB (19 tasks)	**77.63** ± 0.23	76.28 ± 0.46	72.72 ± 0.21	76.29 ± 1.70	–
TPUv3-core-days	2.5k	0.68k	0.23k	9.9k	12.3k

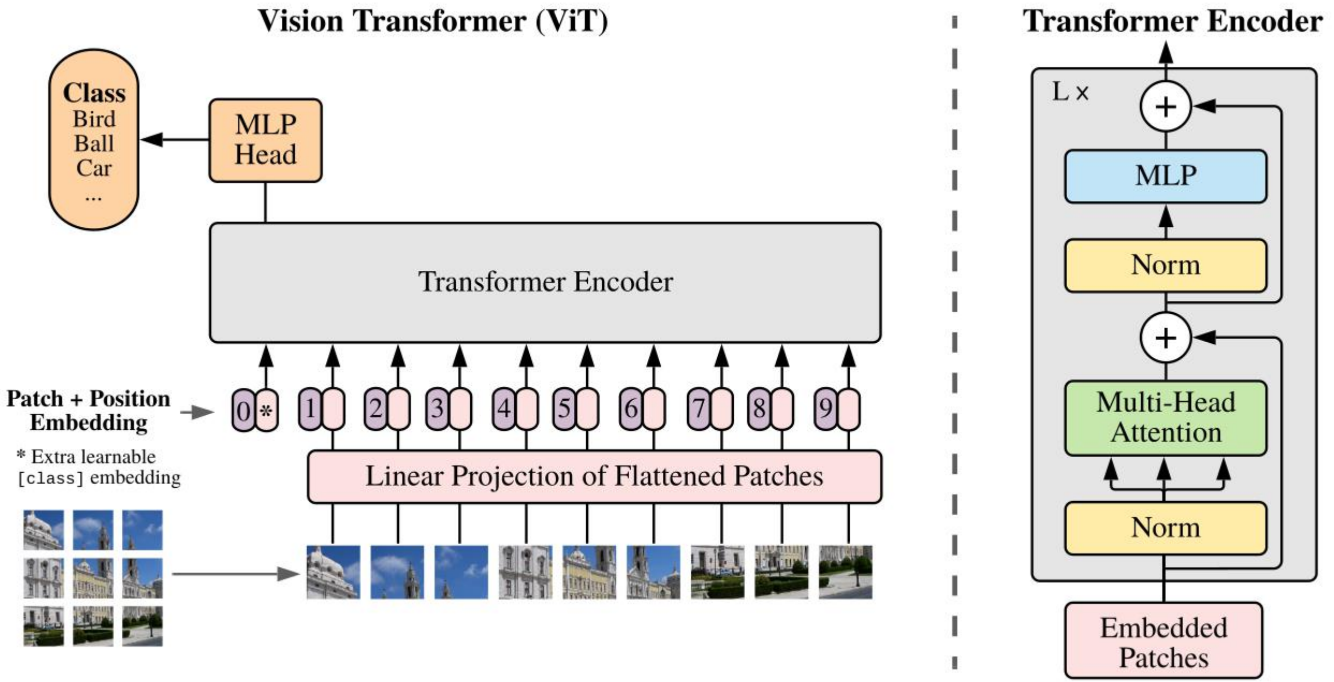
Table 2: Comparison with state of the art on popular image classification benchmarks. We report mean and standard deviation of the accuracies, averaged over three fine-tuning runs. Vision Transformer models pre-trained on the JFT-300M dataset outperform ResNet-based baselines on all datasets, while taking substantially less computational resources to pre-train. ViT pre-trained on the smaller public ImageNet-21k dataset performs well too. *Slightly improved 88.5% result reported in Touvron et al. (2020).

Vision Transformer

ViT(“纯”Transformer模型)

Hybrid(传统CNN和Transformer混合模型)

Vision Transformer

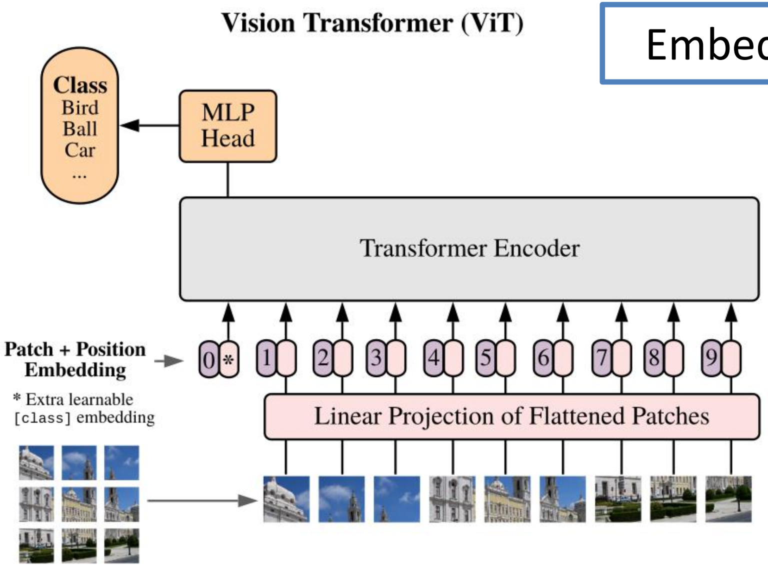


- ➢ Linear Projection of Flattened Patches(Embedding层)
- ➢ Transformer Encoder(图右侧有给出更加详细的结构)
- ➢ MLP Head（最终用于分类的层结构）

Vision Transformer

ViT推理过程

Vision Transformer



Embedding层

对于标准的Transformer模块，要求输入的是token (向量)序列，即二维矩阵[num_token, token_dim]

在代码实现中，直接通过一个卷积层来实现以ViT-B/16为例，使用卷积核大小为16x16，stride为16，卷积核个数为768

[224, 224, 3] -> [14, 14, 768] -> [196, 768]

在输入Transformer Encoder之前需要加上[class]token以及Position Embedding，都是可训练参数

拼接[class]token：Cat([1, 768], [196, 768]) -> [197, 768]

叠加Position Embedding：[197, 768] -> [197, 768]

Vision Transformer

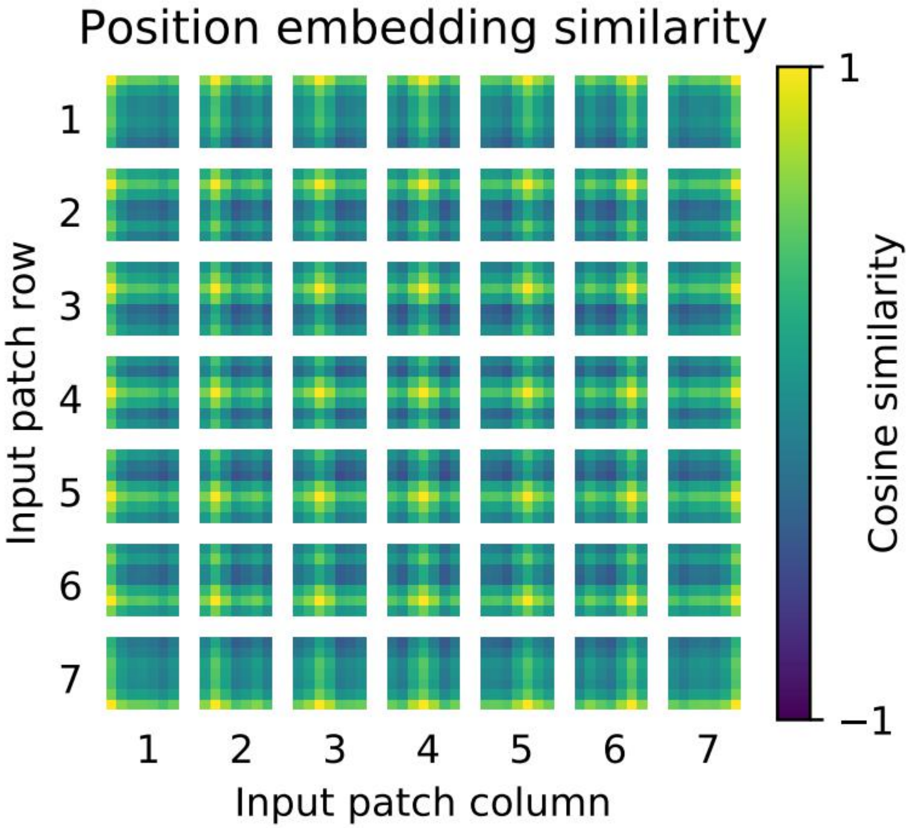
Position Embedding

Pos. Emb.	Default/Stem	Every Layer	Every Layer-Shared
No Pos. Emb.	0.61382	N/A	N/A
1-D Pos. Emb.	0.64206	0.63964	0.64292
2-D Pos. Emb.	0.64001	0.64046	0.64022
Rel. Pos. Emb.	0.64032	N/A	N/A

Table 8: Results of the ablation study on positional embeddings with ViT-B/16 model evaluated on ImageNet 5-shot linear.

the differences in how to encode spatial information is less important

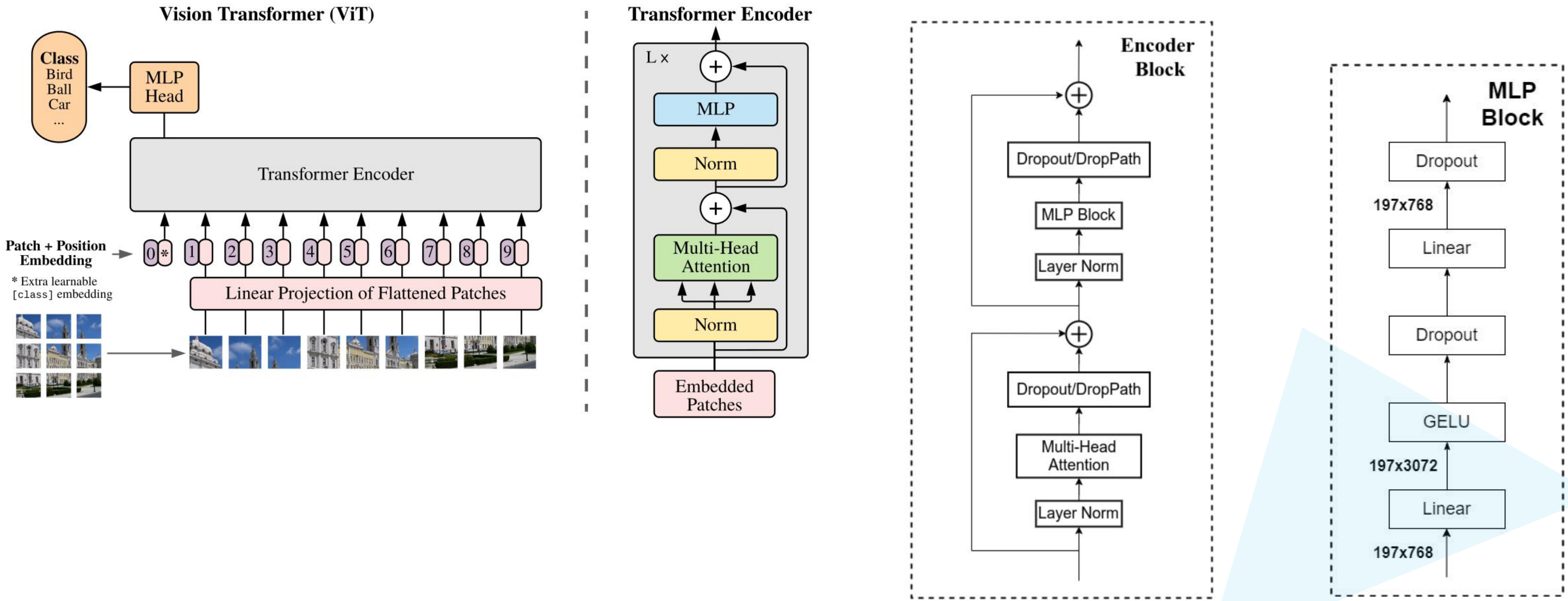
Vision Transformer



closer patches tend to have more similar position embeddings

Vision Transformer

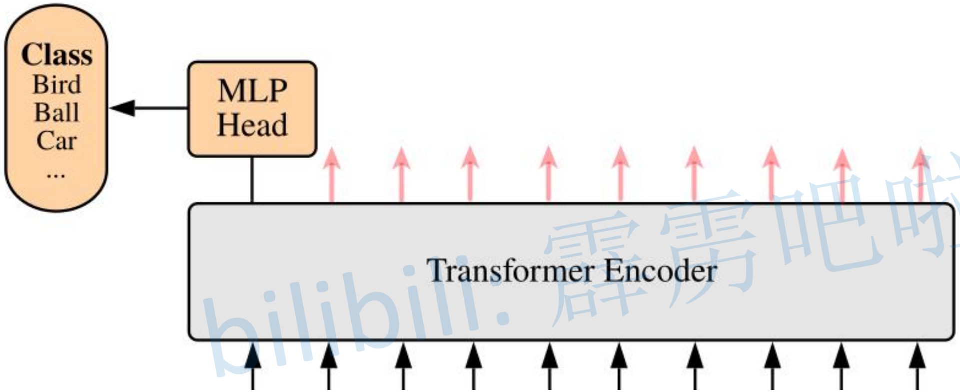
Transformer Encoder层



博文链接：https://blog.csdn.net/qq_37541097/article/details/117653177

Vision Transformer

MLP Head层



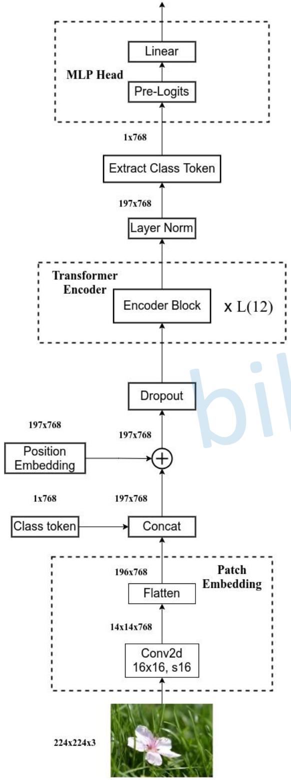
注意，在Transformer Encoder前有个Dropout层，后有一个Layer Norm

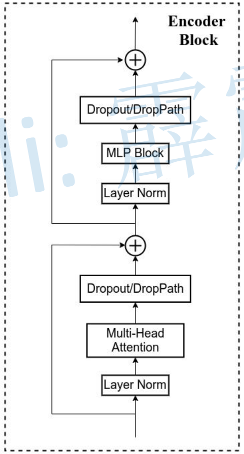
训练ImageNet21K时是由Linear+tanh激活函数+Linear

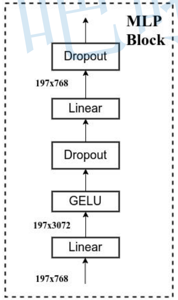
但是迁移到ImageNet1K上或者你自己的数据上时，只有一个Linear

Vision Transformer

ViT-B/16







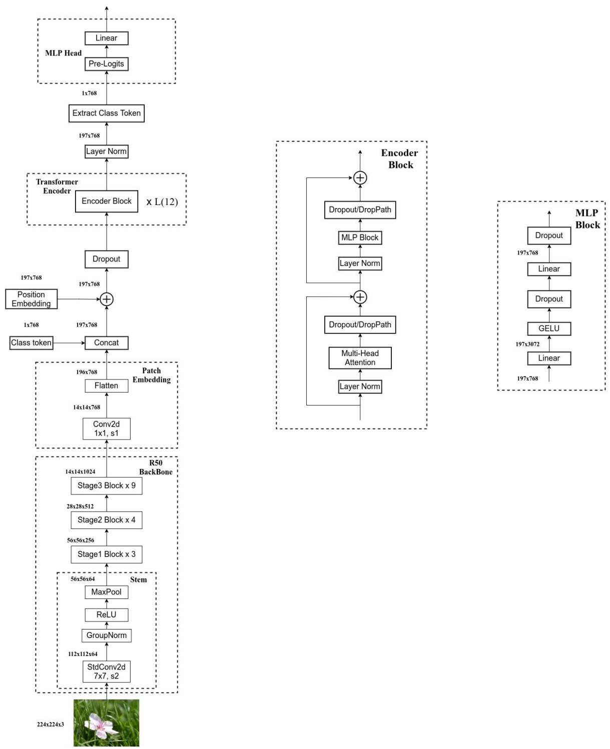
Vision Transformer

Model	Patch Size	Layers	Hidden Size D	MLP size	Heads	Params
ViT-Base	16x16	12	768	3072	12	86M
ViT-Large	16x16	24	1024	4096	16	307M
ViT-Huge	14x14	32	1280	5120	16	632M

- Layers是Transformer Encoder中重复堆叠Encoder Block的次数
- Hidden Size是通过Embedding层后每个token的dim（向量的长度）
- MLP size是Transformer Encoder中MLP Block第一个全连接的节点个数（是Hidden Size的四倍）
- Heads代表Transformer中Multi-Head Attention的heads数

Vision Transformer

R50+ViT-B/16 hybrid model



Hybrid混合模型

R50的卷积层采用的StdConv2d不是传统的Conv2d

将所有的BatchNorm层替换成GroupNorm层

把stage4中的3个Block移至stage3中

博文链接：https://blog.csdn.net/qq_37541097/article/details/118016048

Vision Transformer

Model	Epochs	ImageNet	ImageNet ReaL	CIFAR-10	CIFAR-100	Pets	Flowers	exaFLOPs
ViT-B/32	7	80.73	86.27	98.61	90.49	93.40	99.27	164
ViT-B/16	7	84.15	88.85	99.00	91.87	95.80	99.56	743
ViT-L/32	7	84.37	88.28	99.19	92.52	95.83	99.45	574
ViT-L/16	7	86.30	89.43	99.38	93.46	96.81	99.66	2586
ViT-L/16	14	87.12	89.99	99.38	94.04	97.11	99.56	5172
ViT-H/14	14	88.08	90.36	99.50	94.71	97.11	99.71	12826
ResNet50x1	7	77.54	84.56	97.67	86.07	91.11	94.26	150
ResNet50x2	7	82.12	87.94	98.29	89.20	93.43	97.02	592
ResNet101x1	7	80.67	87.07	98.48	89.17	94.08	95.95	285
ResNet152x1	7	81.88	87.96	98.82	90.22	94.17	96.94	427
ResNet152x2	7	84.97	89.69	99.06	92.05	95.37	98.62	1681
ResNet152x2	14	85.56	89.89	99.24	91.92	95.75	98.75	3362
ResNet200x3	14	87.22	90.15	99.34	93.53	96.32	99.04	10212
R50x1+ViT-B/32	7	84.90	89.15	99.01	92.24	95.75	99.46	315
R50x1+ViT-B/16	7	85.58	89.65	99.14	92.63	96.65	99.40	855
R50x1+ViT-L/32	7	85.68	89.04	99.24	92.93	96.97	99.43	725
R50x1+ViT-L/16	7	86.60	89.72	99.18	93.64	97.03	99.40	2704
R50x1+ViT-L/16	14	87.12	89.76	99.31	93.89	97.36	99.11	5165

沟通方式

1.github

https://github.com/WZMIAOMIAO/deep-learning-for-image-processing

2.bilibili

https://space.bilibili.com/18161609/channel/index

3.CSDN

https://blog.csdn.net/qq_37541097/article/details/103482003