Economics 2355: Transformers for Computer Vision

Harvard University

April 2021

Outline

Transformers for NLP

- Since its unveiling in 2017, the transformer architecture has radically changed the landscape of modern NLP
- NLP benchmarks are now almost exclusively being set by models with transformer architectures
- Conceptually, the value add of transformers is two-fold:
  - the ability to generate robust contextualized representations of inputs, i.e., self-attention, all while still being able to...
  - parallelize training across compute nodes, i.e., computational efficiency
- Moreover, transformer-based architectures like GPT-3 have shown that massive amounts of data can be fed to transformers without sharply diminishing returns to performance–in other words transformers are highly scalable too

Transformers for Computer Vision

- The ability to generate contextualized representations of inputs in parallel is not just of use to NLP, however!
- It stands to reason that computer vision tasks could benefit just as much from robust attention mechanisms that train quickly in parallel
- The promise of squeezing lots of available image data even further for computer vision tasks in a “pre-train and fine-tune” paradigm is likewise compelling

Transformers for Computer Vision

- A recent wave of models and methods have therefore been working on adapting the transformer architecture to computer vision settings
- The literature on transformers for computer vision is *extremely* recent, with many of the most important publications in this space having been released under a year ago
- Although the transformer architecture has fewer inductive biases–architectural predispositions to certain assumptions about the data–than some neural networks, transitioning the transformer to computer vision tasks has required a fair number of new architectural insights and training insights
- We will now discuss how the transformer has been modified and repurposed to accomplish image classification and object detection, the core computer vision tasks covered at the start of this course!

Outline

Transformers for Computer Vision

Transformers for Image Classification

Transformers for Object Detection

Transformers for Image Classification

- Image classification is the quintessential computer vision task, and progress in image classification in some sense proxies the evolution of image understanding in computers
- In October of 2020, the paper “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale” was released by researchers at Google (Dosovitskiy et al. 2020), which pioneered the high performance application of transformers to the task of image classification

The Vision Transformer (ViT)

- Dosovitskiy et al. (2020) developed a model called the Vision Transformer, or ViT, which, for the first time, showed that you can get a SOTA image classifier using only a transformer-based architecture, i.e., no CNNs are required to extract features from image inputs
  - Dosovitskiy et al. (2020) does introduce a "hybrid" model too, which incorporates CNNs into their novel architecture, although this hybrid model performs nearly equivalently to the entirely transformer-based model in most settings of interest

The Vision Transformer (ViT)

- Despite how quickly the field of computer vision evolves, as of this week, according to `paperswithcode.com`, ViT is still #6 for ImageNet, and #2 for CIFAR-10 and CIFAR-100, only being beaten out by CNN-based architectures that have benefitted from years of "architectural search," hyperparameter search, and other forms of development
- Delving into how and why ViT works will pave the way for a greater understanding of the emergent literature on transformers for computer vision

ViT: Inputs

- ViT aims at "applying a standard Transformer directly to images, with the fewest possible modifications"
- ViT therefore first finds a way to translate images into something analogous to, e.g., BERT's inputs: it splits images into fixed-size "patches" and embeds these patches into a high-dimensional space of choice using a linear transformation; this 1D sequence of patch-based, "linear embeddings" is analagous to a 1D sequence of token embeddings in BERT, RoBERTa, etc.
- Learned position embeddings are then added to each patch embedding to form input embeddings, again similarly to a classic transformer

ViT: Inputs

- A learned classification embedding, also analagously to BERT, is prepended to this sequence of patch embeddings
- Treating an image as a sequence of patches (and patch embeddings) may seem obvious, but, as with many developments in DL, this only appears to be so in retrospect: many early applications of transformers to computer vision naturally worked at the pixel level, but struggled because “With quadratic cost in the number of pixels, this [pixel-wise self-attention] does not scale to realistic input sizes,” which meant complicated fixes like sparser attention mechanisms, etc.

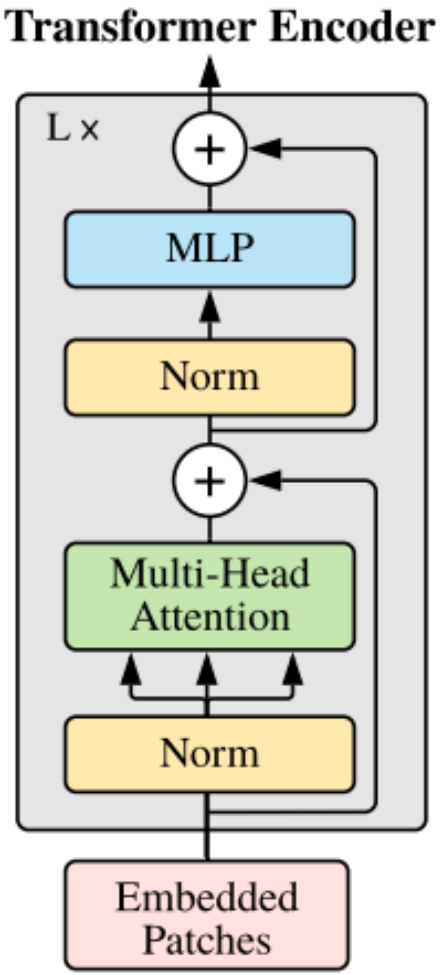
ViT: Inputs

- Patch-based embedding allowed for ViT to be the first transformer “with global self-attention to full-sized images”
- In the hybrid version of ViT, “the patch embedding projection ... is applied to patches extracted from a CNN feature map” instead of the actual image

ViT: Architecture

- With transformer-compatible inputs formed, the rest of the ViT architecture looks much like a standard transformer-encoder in the style of BERT
- The transformer-encoder for ViT consists of "alternating layers of multiheaded self-attention... and MLP blocks" for which "Layernorm (LN) is applied before every block, and residual connections after every block," and for which "The MLP contains two layers with a GELU non-linearity"

ViT: Architecture



A Short Digression on GELU

- GELU is another non-linear activation function introduced by Hendrycks and Gimpel (2016), frequently used by transformer-based architectures, including BERT and the GPT series
- GELU stands for "Gaussian Error Linear Unit", and mathematically it is just $GELU(x) = x\Phi(x)$, where $\Phi(x)$ is the standard Gaussian cumulative distribution function

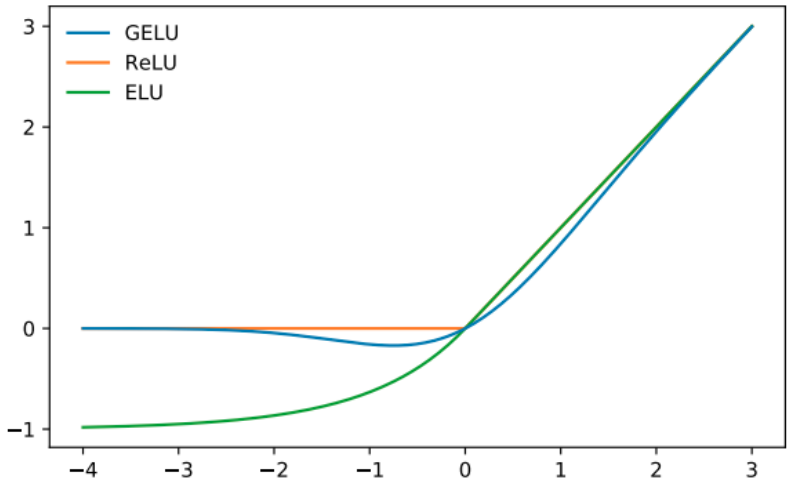


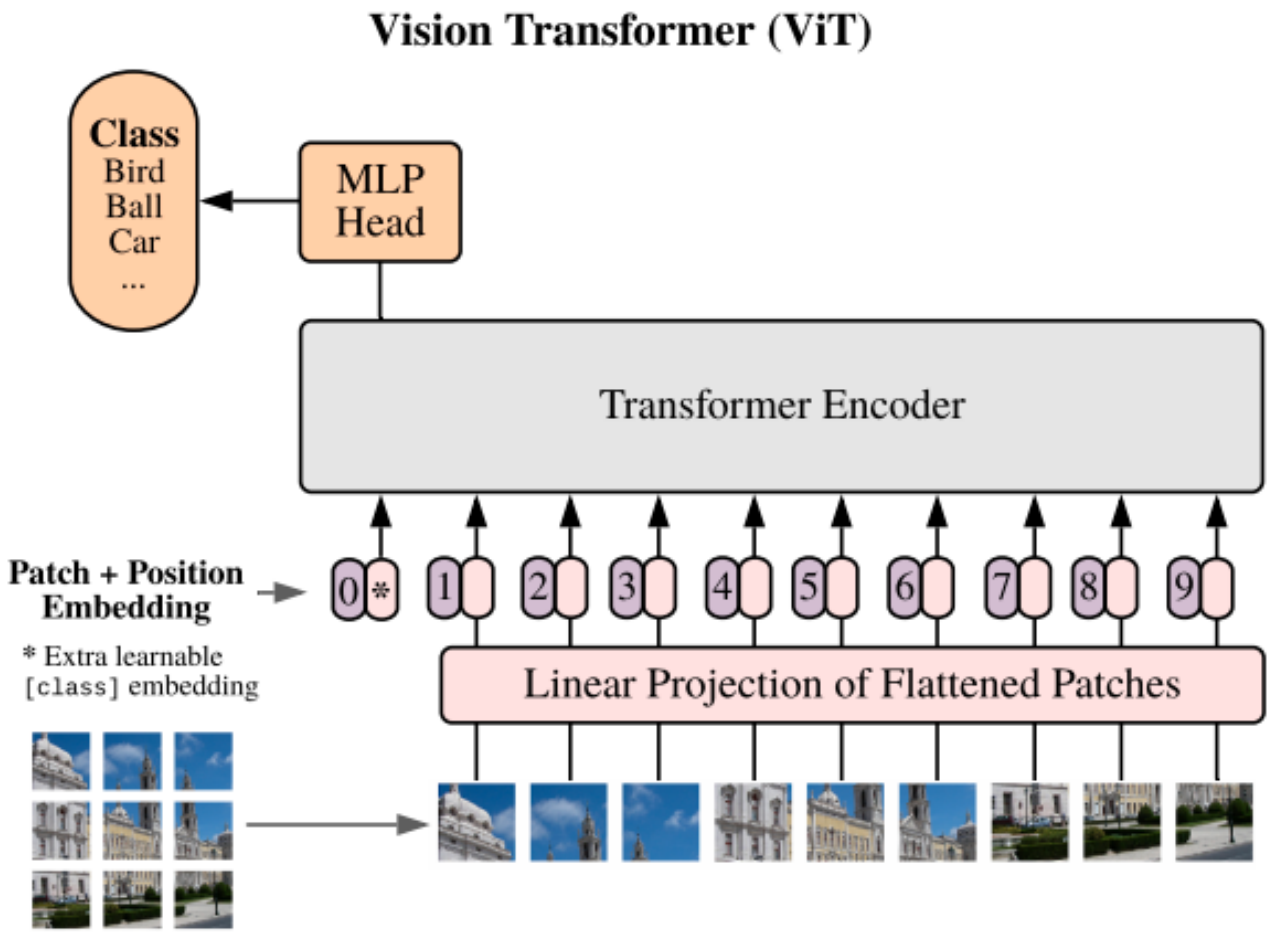
Figure 1: The GELU ($\mu = 0, \sigma = 1$), ReLU, and ELU ($\alpha = 1$).

ViT: Architecture

- The “tokenized” (patched) and then embedded input sequence passes through the transformer encoder, and the output “[CLASS]” token is attached to a classification head that “is implemented by a MLP with one hidden layer at pre-training time and by a single linear layer at fine-tuning time”

ViT: Architecture

All together this looks like...



ViT: Architecture

- Multiple sizes of ViT are created, with the largest having over 600M parameters

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

Table 1: Details of Vision Transformer model variants.

ViT: Training

- How and on what ViT was trained is also crucial to its success as an image classifier
- As the authors indicate: “When trained on mid-sized datasets such as ImageNet, such [transformer-based] models yield modest accuracies of a few percentage points below ResNets of comparable size”
- As they go on to state: “This seemingly discouraging outcome may be expected: Transformers lack some of the inductive biases inherent to CNNs, such as translation equivariance and locality, and therefore do not generalize well when trained on insufficient amounts of data. However, the picture changes if the models are trained on larger datasets (14M-300M images). We find that large scale training trumps inductive bias”

ViT: Training

- Practically, what this means is that ViT needs massive amounts of training data to achieve SOTA performance on image classification tasks
- This, too, echoes the NLP transformer paradigm of pre-training, and, in some sense, ViT can be thought of as needing to "pre-train" on a massive number of images to do well on downstream classification tasks
- In standard transfer learning fashion, the authors of ViT "pre-train" it on large image datasets like JFT-300M (an internal Google dataset), and then fine-tune ViT to downstream tasks with fewer labeled data points (using a new "zero-initialized D x K feedforward layer, where K is the number of downstream classes")

ViT: Training

- Few-shot performance after JFT-300M pre-training is shown to be quite strong with more and more pre-training data

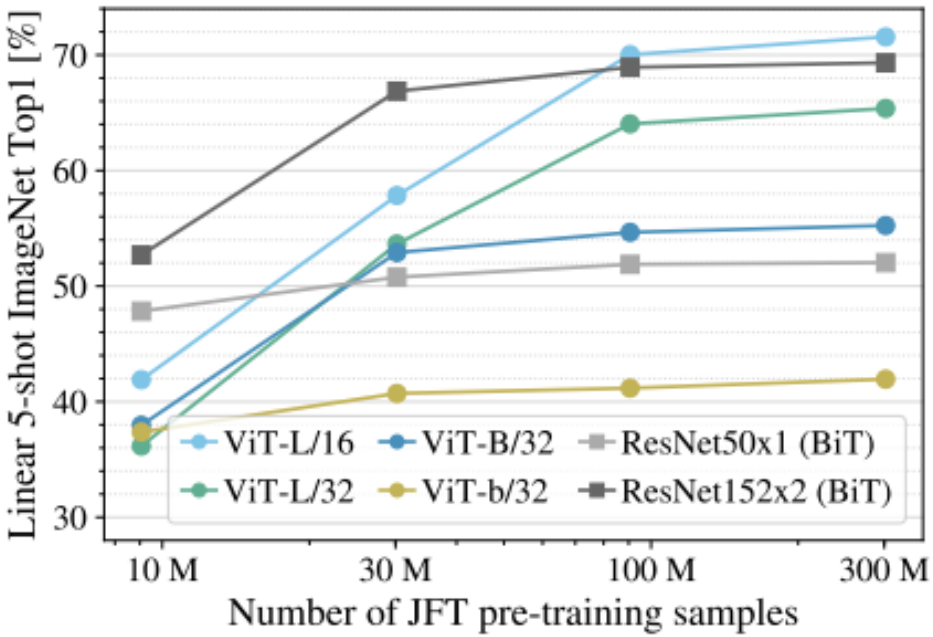


Figure 4: Linear few-shot evaluation on ImageNet versus pre-training size. ResNets perform better with smaller pre-training datasets but plateau sooner than ViT, which performs better with larger pre-training. ViT-b is ViT-B with all hidden dimensions halved.

ViT: Training

- ViT's authors also tout its computational efficiency, holding compute fixed

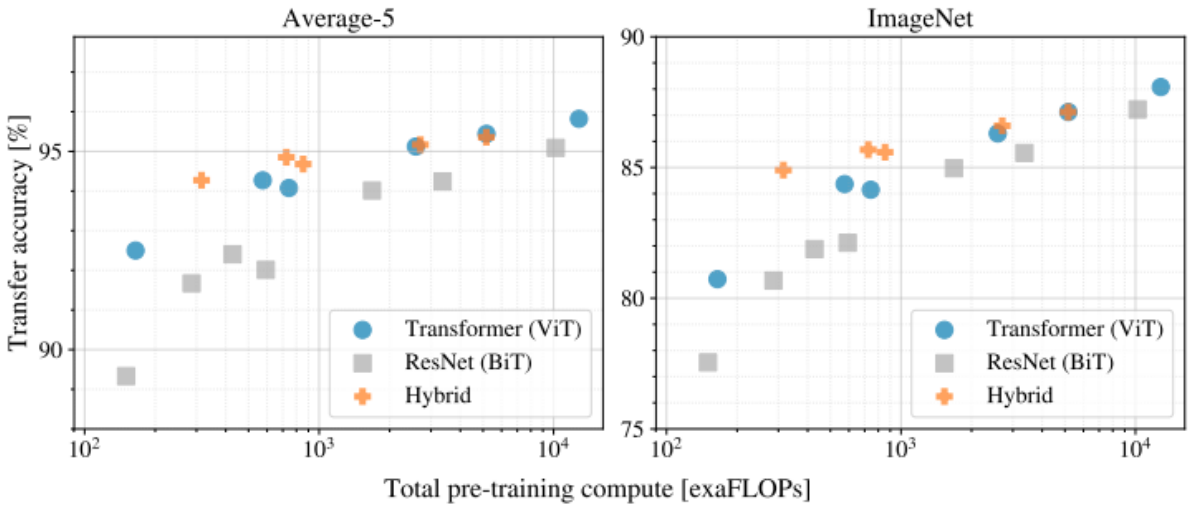


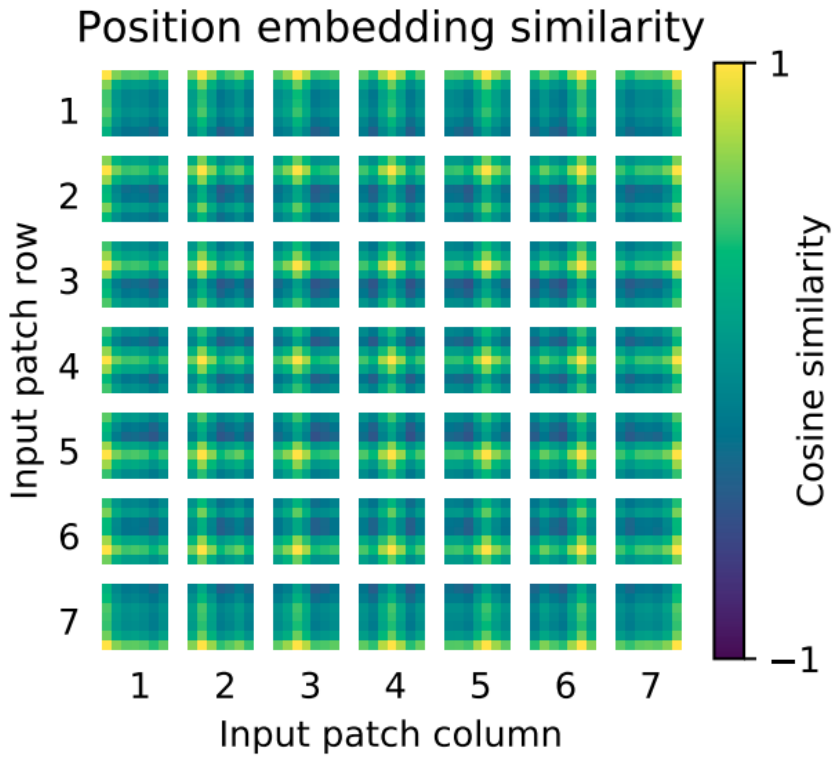
Figure 5: Performance versus cost for different architectures: Vision Transformers, ResNets, and hybrids. Vision Transformers generally outperform ResNets with the same computational budget. Hybrids improve upon pure Transformers for smaller model sizes, but the gap vanishes for larger models.

ViT: Training

- As introduced, ViT is trained end to end on classification loss for image recognition/classification tasks
- However, ViT authors also try out a self-supervised pre-training regime too, called masked patch prediction, which resembles BERT MLM pre-training
- In masked patch predition, 50% of patch embeddings are corrupted and some prediction about the corrupted patches is made, e.g., mean color, regression on the full patch
- The performance of the self-supervised model on ImageNet classification is slightly worse than the supervised model, but requires a dataset much smaller than JFT to get similar performance
- Advancements in self-supervision, much like in NLP, may pave the way for less data hungry vision transformers in the future

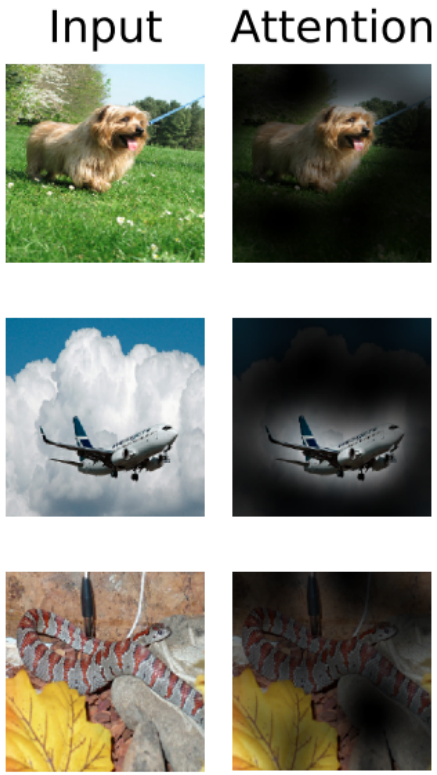
ViT: Dissection

- When dissecting a trained ViT to study its internal representations, it can be seen, for example, that ViT learns a grid-like structure for its learned position embeddings



ViT: Dissection

- Using a method called “Attention Rollout” (Abnar and Zuidema, 2020), authors of ViT also visualize the attention of the classification token in the input space, to sanity-check passing results



ViT: Takeaways

- That an entirely transformer-based image classifier has SOTA or near-SOTA performance on a variety of image classification tasks already in its introductory paper is remarkable given how much time, effort, and study has been put into getting CNNs to have similar performance since AlexNet
- A motif in DL appears again, however: to get SOTA performance, you need massive amounts of data...

The Data-Efficient Image Transformer (DeiT)

- Just a few months after the release of ViT, Touvron et al. (2020) released a model called DeiT, which starts to address this data hungriness of vision transformers
- DeiT has the same fundamental architecture as ViT, but differs substantially in its training

DeiT: Training

- There are two primary ideas behind DeiT:
  - Use a knowledge distillation procedure to train high-performing, small versions of ViT quickly
  - Invoke off-the-shelf data augmentation for the ImageNet dataset so no external datasets are required, as well as perform more intensive hyperparameter search for better, faster training
- While the data augmentation piece of creating DeiT is somewhat straightforward (e.g., using RandAugment to randomly rotate, crop, shear, translate, change contrast of, etc., an image), there is a novel knowledge distillation procedure introduced by DeiT, and unique to transformers

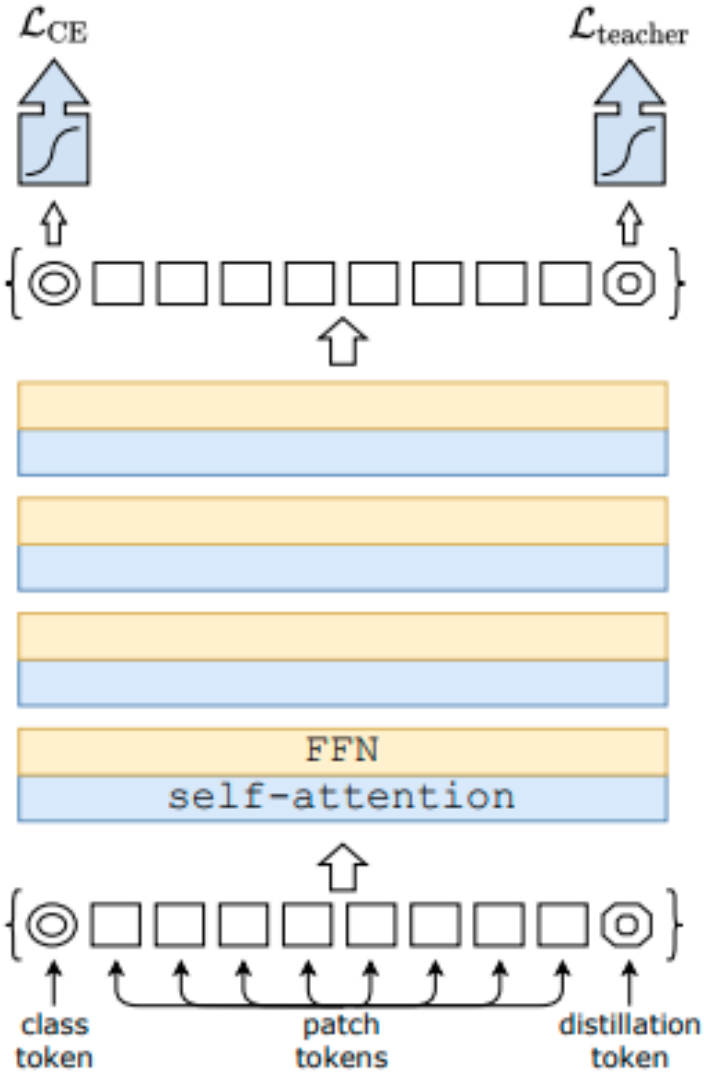
DeiT: Knowledge Distillation

- Knowledge distillation "refers to the training paradigm in which a student model leverages 'soft' labels coming from a strong teacher network," thereby allowing the typically smaller student model to learn to produce results of similar quality to the teacher network
- Typically, knowledge distillation has an objective that will minimize the Kullback-Leibler divergence between the softmax of a teacher model and the softmax of a student model, a.k.a., soft distillation; alternatively, "hard distillation" may be performed, whereby minimizing the cross-entropy between the teacher highest probability prediction and the student prediction is the objective

DeiT: Knowledge Distillation

- DeiT tests out hard and soft knowledge distillation, both using the classification results derived from the standard classification token and using their novel "distillation token"
- This distillation token is another learned embedding like the classification token
- During knowledge distillation-based training, it is the output derived from the distillation token–which communicates with the rest of the model inputs through self-attention–that is used for the knowledge distillation loss
- The DeiT authors find that the distillation token does not converge to the classification token, helps preserve the inductive biases of the teacher network when the teacher network is a CNN-based image classifier, and leads to less of a diminishing return to increasing training epochs

DeiT: Knowledge Distillation



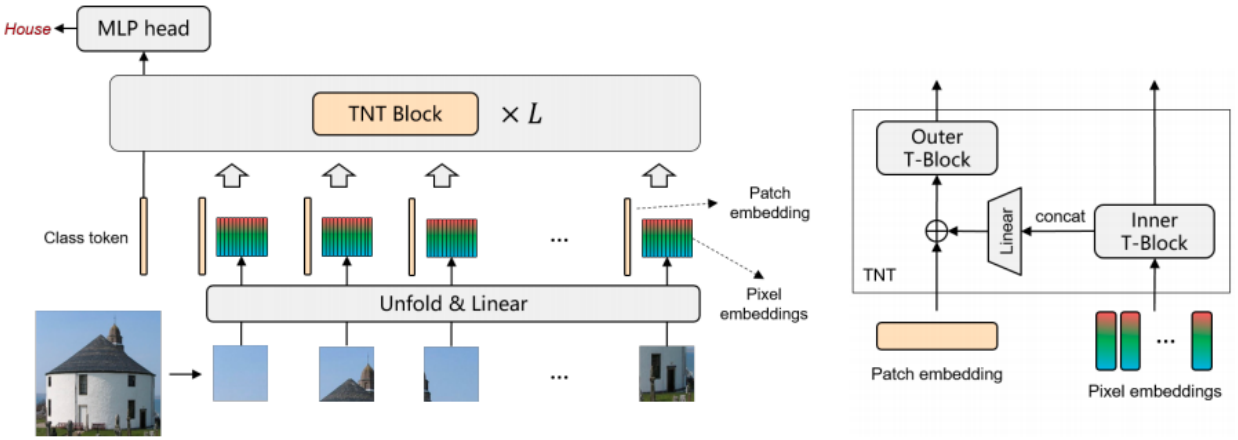
DeiT: Takeaways

- By training smaller models through knowledge distillation–enchanced by a novel distillation token, better training hyperparameters, and many more epochs of training, and only using augmented ImageNet data–DeiT achieves classification performance only slightly below the current SOTA image classifier EfficientNet
- As the DeiT authors state, this "shows that we have almost closed the gap between vision transformers and convnets when training with Imagenet only," despite the fact that EfficientNet is a CNN that "has benefited from years of research on convnets and was optimized by architecture search on the ImageNet validation set"

The Transformer-iN-Transformer (TNT)

- And the image transformer literature continues to grow...
- In February 2021, the model Transformer-iN-Transformer (TNT) was released (Han et al. 2021), which "achieves 81.3% top-1 accuracy on ImageNet which is 1.5% higher than that of DeiT with similar computational cost"
- It does so by using an "inner transformer" block to model pixel level relationships–within patches, through attention–as a form of local feature extraction that maintains spatial structure within patches, and by using an "outer transformer" block to extract features across patches through attention, as in ViT

The Transformer-iN-Transformer (TNT)



(a) TNT framework. (b) TNT block.

Figure 1: Illustration of the proposed Transformer-iN-Transformer (TNT) framework. The position embedding is not drawn for neatness. T-Block denotes transformer block.

Outline

Transformers for Object Detection

- Image transformers aren't just being used for image classification–they're also being used successfully for object detection
- Transformers for object detection actually predate ViT; DETR (Carion et al. 2020) used a transformer architecture to model object detection as a "set prediction" problem, and achieved near-SOTA results
- However, DETR made use of a CNN backbone, much like Mask or Faster R-CNN, and was therefore not a fully transformer-based approach to feature extraction in images (in some sense, DETR can be thought of as using transformers to replace everything *except* the CNN backbone for object detection)

Transformers for Object Detection

- Now, however, there is a high performing object detection model that entirely uses transformers for feature extraction for object detection: Swin-T
- As of today, April 14, 2021, Swin-T is the SOTA for object detection: it has the highest performance ever recorded on the COCO object detection dataset and task

The Swin Transformer (Swin-T)

- Where DETR was everything but the backbone for object detection, the Swin Transformer (Swin-T) *is* the backbone for object detection
- Swin-T is "a hierarchical Transformer whose representation is computed with **s**hifted **win**dows"
- Swin-T aims to solve the problem of generating multi-scale features from high resolution images using transformers, a key component of doing good object detection with no clear analog in the original transformer architecture
- Swin-T also aims to address the vision transformer's seeming incompatibility with doing dense prediction, e.g., for semantic segmentation, because of the computational complexity involved in self-attention at the pixel or near-pixel level in high resolution images (note: ViT doesn't focus on dense prediction, and therefore can rely solely on patch level features)

Swin-T: Architecture

- Swin-T creates a hierarchical, multi-scale representation of an image–analagous to a multi-scale feature pyramid–by first performing self-attention on small patches of an image only within local “windows” of the image, then merging neighboring patches and performing windowed self-attention on these new, larger patches, and continuing this pattern through the depth of the transformer layers

Swin-T: Architecture

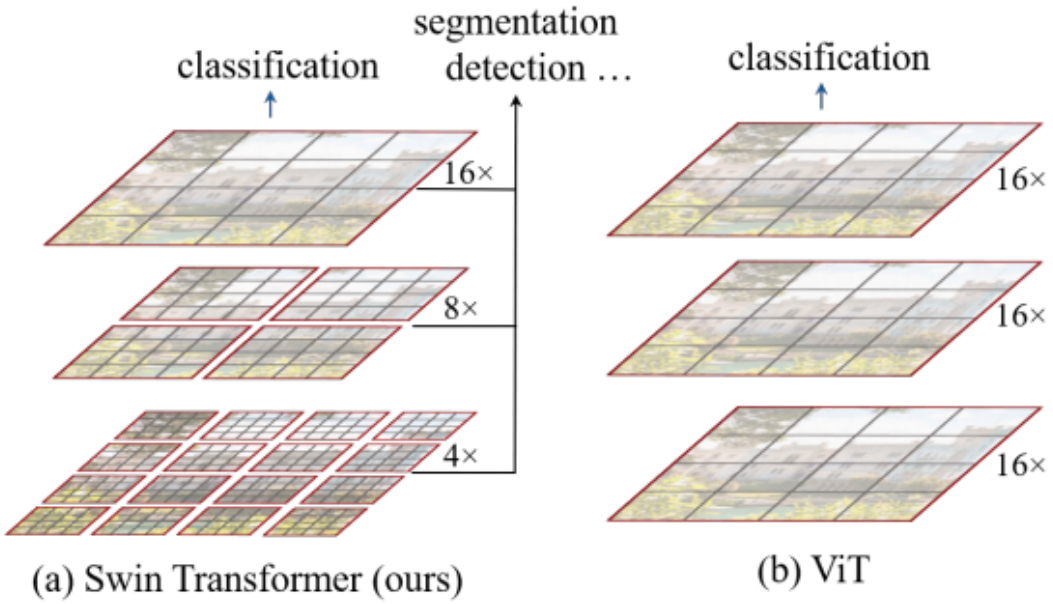


Figure 1. (a) The proposed Swin Transformer builds hierarchical feature maps by merging image patches (shown in gray) in deeper layers and has linear computation complexity to input image size due to computation of self-attention only within each local window (shown in red). It can thus serve as a general-purpose backbone for both image classification and dense recognition tasks. (b) In contrast, previous vision Transformers [19] produce feature maps of a single low resolution and have quadratic computation complexity to input image size due to computation of self-attention globally.

Swin-T: Architecture

- Each stage of the Swin-T, composed of multiple transformer blocks, produces something akin to a feature map, which can be used as input to create a FPN, and/or as a backbone for, e.g., a Mask R-CNN
- For Swin-T, the “number of patches in each window is fixed, and thus the complexity becomes linear to image size”

Swin-T: Architecture

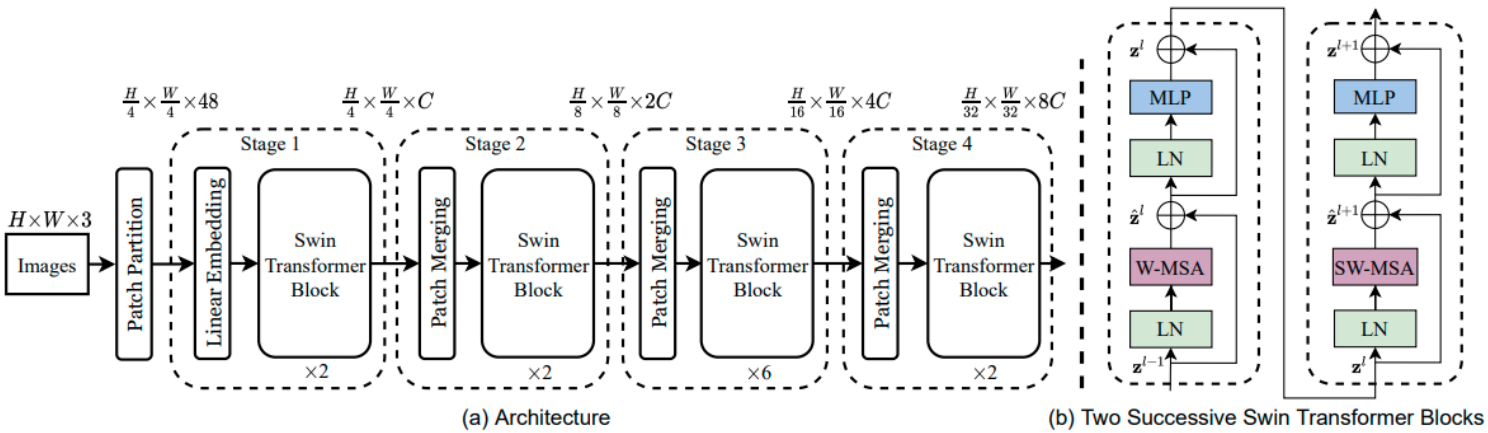


Figure 3. (a) The architecture of a Swin Transformer (Swin-T); (b) two successive Swin Transformer Blocks (notation presented with Eq. (3)). W-MSA and SW-MSA are multi-head self attention modules with regular and shifted windowing configurations, respectively.

Swin-T: Shifted Windows

- The authors of Swin-T show that one of its most important design choices is to perform windowed self-attention over windows that shift in spatial extent across alternating layers
- As the authors state, “The shifted window partitioning approach introduces connections between neighboring non-overlapping windows in the previous layer and is found to be effective in image classification, object detection, and semantic segmentation,” i.e., increasing the model’s downstream performance on all tasks
- Shifted windows, as opposed to sliding windows, are also lower latency on real world hardware in the context of transformers, as demonstrated in ablation studies, as all query patches/embeddings within a window share the same key patches/embeddings

Swin-T: Shifted Windows

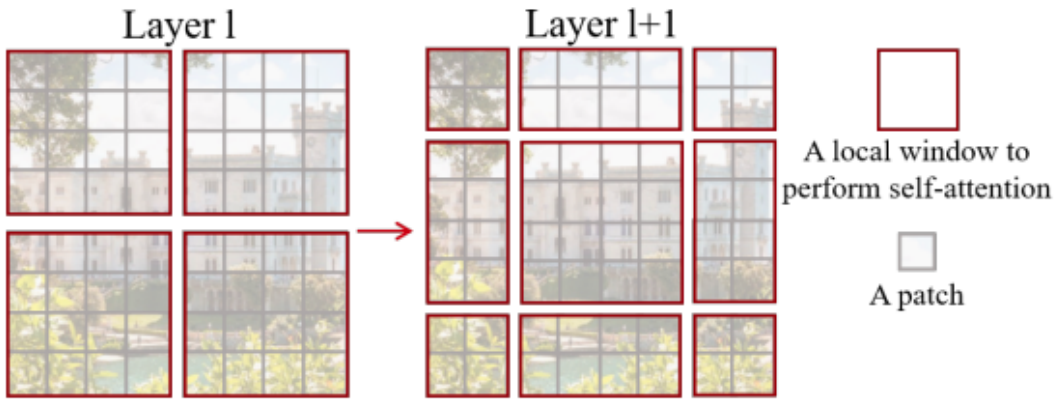


Figure 2. An illustration of the *shifted window* approach for computing self-attention in the proposed Swin Transformer architecture. In layer l (left), a regular window partitioning scheme is adopted, and self-attention is computed within each window. In the next layer $l + 1$ (right), the window partitioning is shifted, resulting in new windows. The self-attention computation in the new windows crosses the boundaries of the previous windows in layer l, providing connections among them.

Swin-T: Takeaways

- Essentially, by applying transformer self-attention at multiple scales, Swin-T is able to computationally efficiently extract rich features from even high resolution images
- Consequently, Swin-T's deepest, largest variant is able to get a box AP score of 58.7 on the COCO object detection task, the highest ever reported!

The Unification of Deep Learning Methods

- Swin-T's authors see their development as pushing computer vision methods and models ever closer to NLP methods and models, and hope that it encourages the "unified modeling of vision and language signals"
- From ViT to Swin-T, it's clear that concepts like representation learning, manifested as architectures with fewer and fewer inductive biases like the transformer, are making the entire deep learning space more unified!
- Models like VisualBERT (Liunian et al. 2019), ViLBERT (Lu et al. 2019), and CLIP (Radford et al. 2021) continue to push this frontier
- Under the hood, as computer vision and NLP models and methods become more similar, we expect to see amazing progress on benchmark and non-benchmark tasks over the coming years!