Transformers

Earl Wong

Introduction

- Transformers originated from research in natural language processing (NLP).
- The transformer architecture is currently SOTA for many NLP tasks [2020].
- Recently [2020 / 2021], transformers have been successfully applied to computer vision - specifically, image understanding.
- By stacking / layering transformer building blocks, transformer architectures (exceeding 100 billion parameters) have been created and trained.
- Although data and loss functions still play a crucial role in achieving high performance, the current star is the transformer, since the transformer “upper bound” has yet to be realized.

Introduction

- So what is a transformer?
- A transformer consists of many classical ML components.
- However, the “new component” is the attention mechanism.
- Hence, let’s start by motivating attention.

Attention

- In language, attention is associated with the words (tokens) in a sentence or paragraph that convey important information.

- The Giants beat the Dodgers. (Giants, Dodgers, beat)

- Once the keywords are identified, important associations between the keywords can be determined.

- i.e. Who beat who?

- By construction, attention is a very natural and intuitive construct for high dimensional data, since great benefits can be derived by focusing attention on the important words and their relationships.

Attention

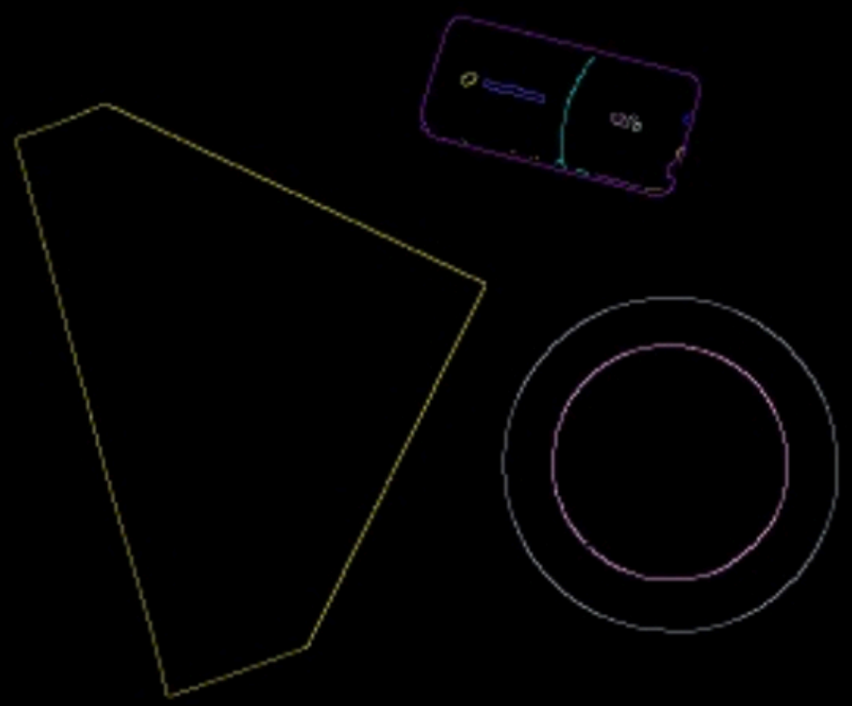
- Images typically contain millions of pixels.
- By “attending” to these pixels / patches, 1) meaningful semantic objects can be identified and 2) their underlying relationships can be determined.

A dog running in the grass with a frisbee.

Attention

- Attention also naturally operates in low level vision via edge detection.

- Edge detection is a fundamental construct and building block of the human visual system (HVS).



Attention

- In general, attention is exactly what the word implies: i.e. What do we attend to?

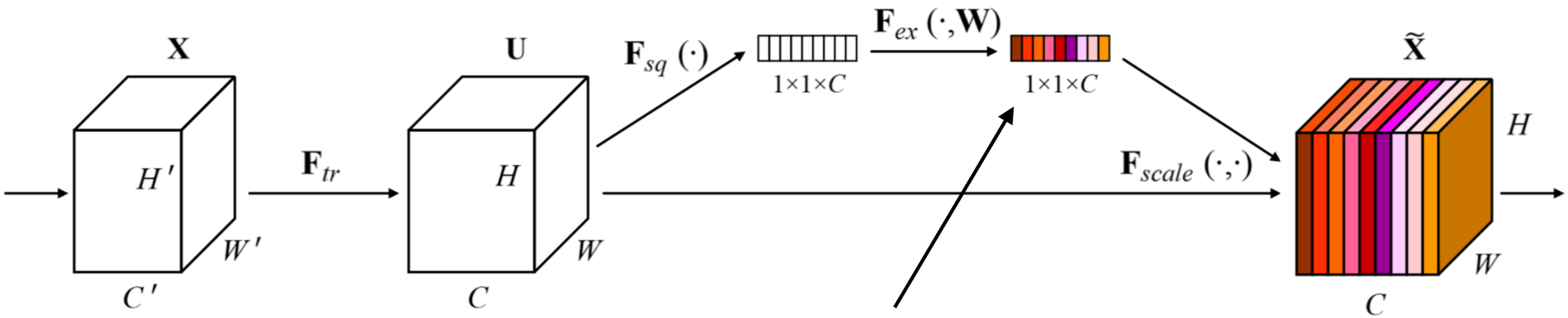
- In a long sentence or paragraph, we attend to the important words (tokens) and their relationships.

- In an image (high level), we attend to the high level semantic objects and their relationships.

- In an image (low level), we attend to the low level image descriptors used by the HVS.

- In deep learning backbone architecture, we attend to the important feature maps constructed by the backbone.

An Simple Deep Learning Based Example of Attention



Excite was a simple, low level attention mechanism.

In 2017, a squeeze and exite block was introduced to improve the performance of backbone CNN's.

Qualitatively, the squeeze mechanism compressed the HxWxU tensor into a scalar descriptor for each channel.

The excite mechanism then determined a weight multiplier for each channel.

Different channels / feature maps in the HxWxU tensor then received different levels of attention, based on their weight multiplier.

Attention

- Machine translation is a very challenging problem in NLP.
- i.e. I am a student -> Je suis etudiant.
- Avenues of active research in machine translation included RNN's, CNN's and hybrid approaches that employed an attention mechanism.
- Question: Can machine translation be performed using only attention? If so, how?

Attention Is All You Need

- In 2017, the answer was provided in a paper titled “Attention Is All You Need”.

- Here, the authors achieved SOTA performance in machine translation using ONLY attention.

- First, let’s define the machine translation problem (aka sequence to sequence problem).

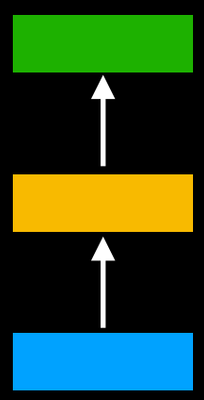
- Then, let’s look at the shortcomings of the previous approaches based on RNN’s and CNN’s.

Sequence Problems

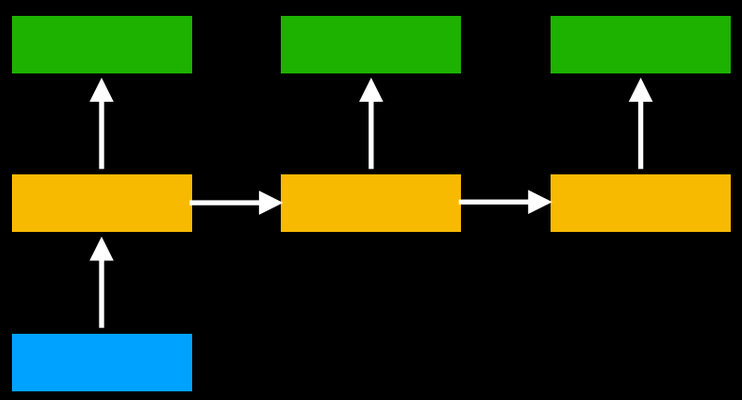
When processing sequences, the inputs and outputs can occur in different ways:

Input **Processing** **Output**

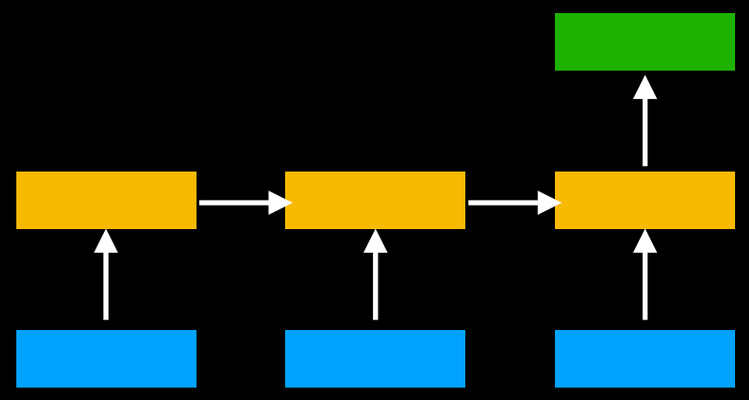
One to One



One to Many



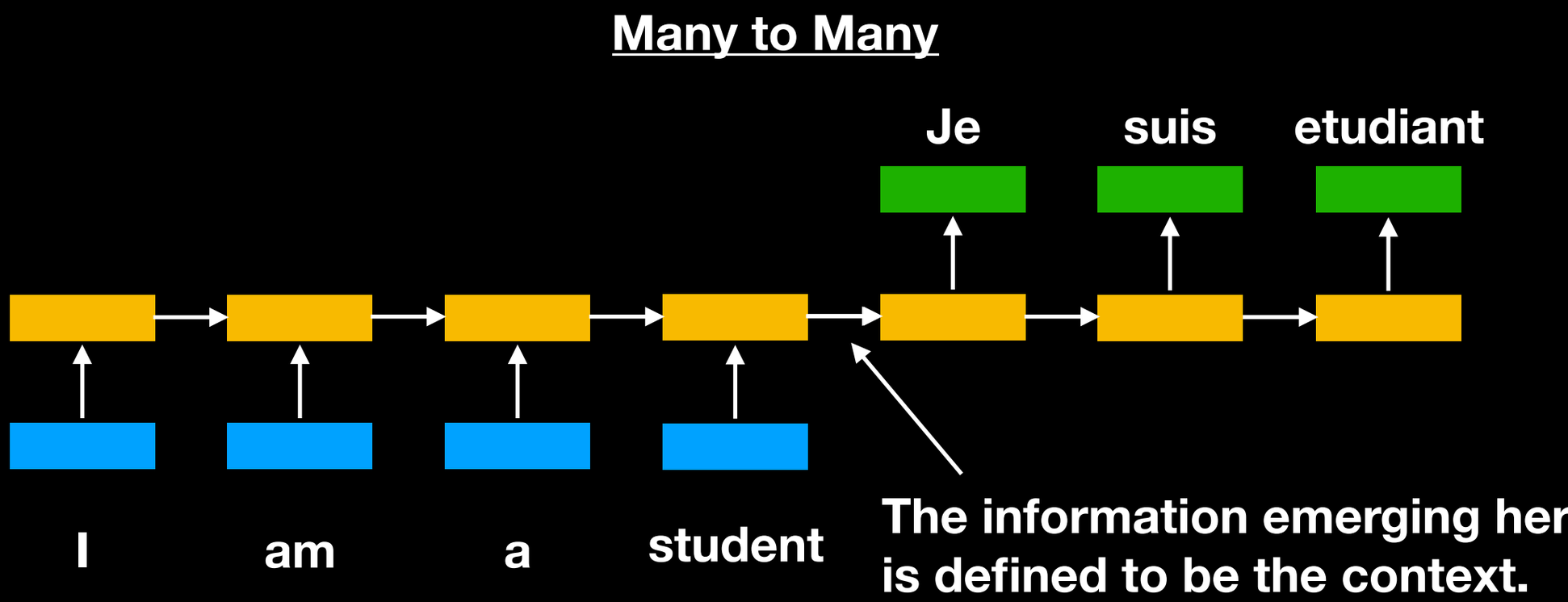
Many to One



Sequence to Sequence Problem

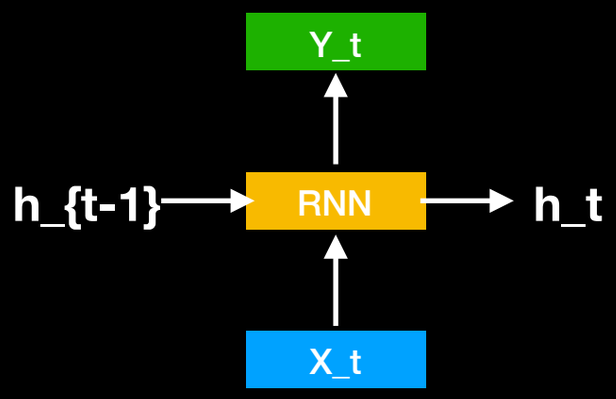
In machine translation, a “Many to Many” (aka sequence to sequence) architecture results.

Input **Processing** **Output**



RNN

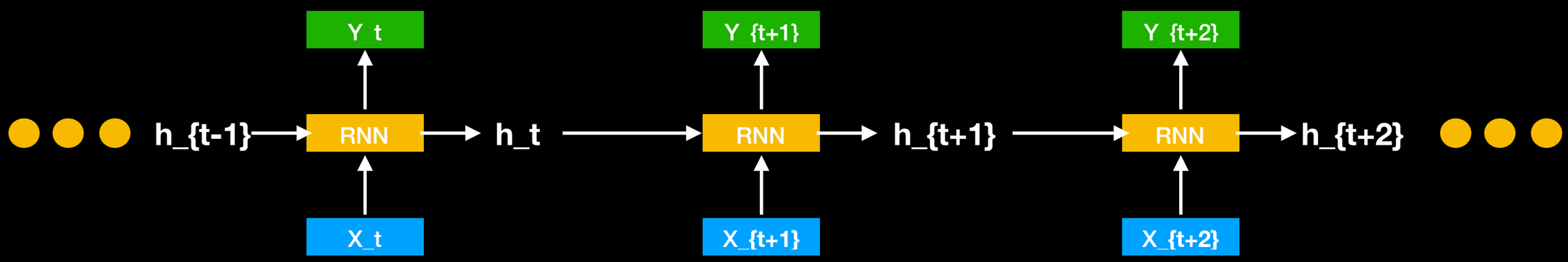
One construct to the sequence to sequence problem is the RNN.



Mathematically, an RNN is composed of 3 matrix operations: W_xh, W_hh and W_hy

This “black box” then yields:
h_t = tanh(W_hh * h_{t-1} + W_xh * X_t)
Y_t = W_hy * h_t

“Unrolling” the RNN, the following architecture results:

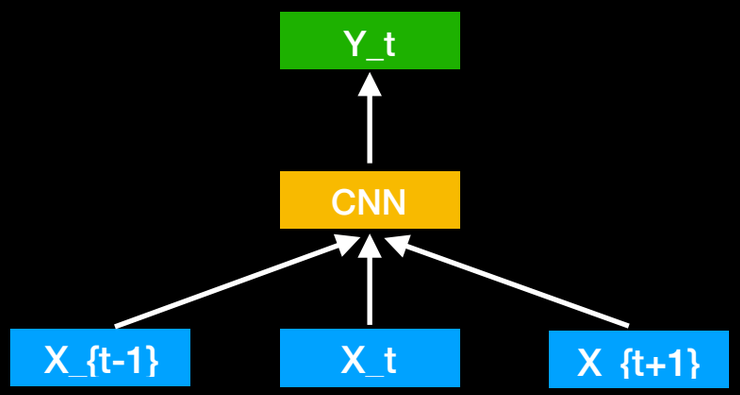


W_xh, W_hh and W_hy share the same learned parameters for each RNN block.

By construction, “future” states h_{t+n} capture / summarize past information.

CNN

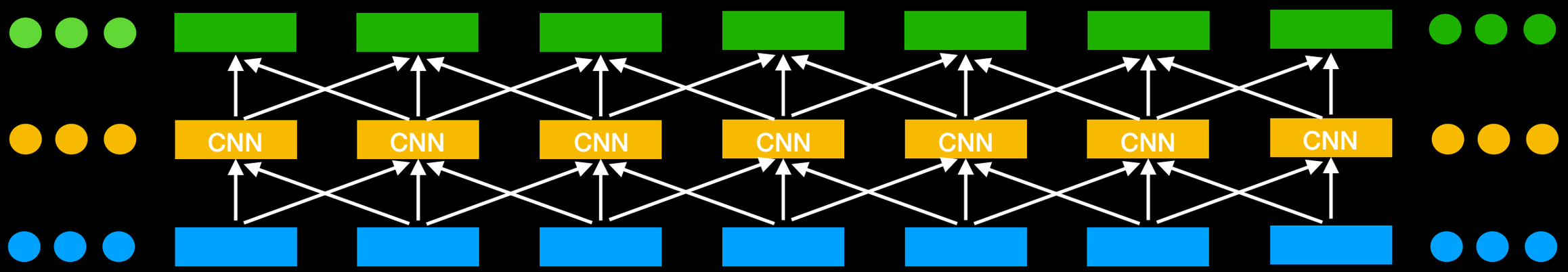
Another construct to the sequence to sequence problem is the CNN.



Mathematically, a 1D convolutional filter of width N=3 is given by Y_t = X_{t-1}W0 + X_t*W1 + X_{t+1}*W2.

Here, the receptive field is defined by the width of the 1D convolutional filter.

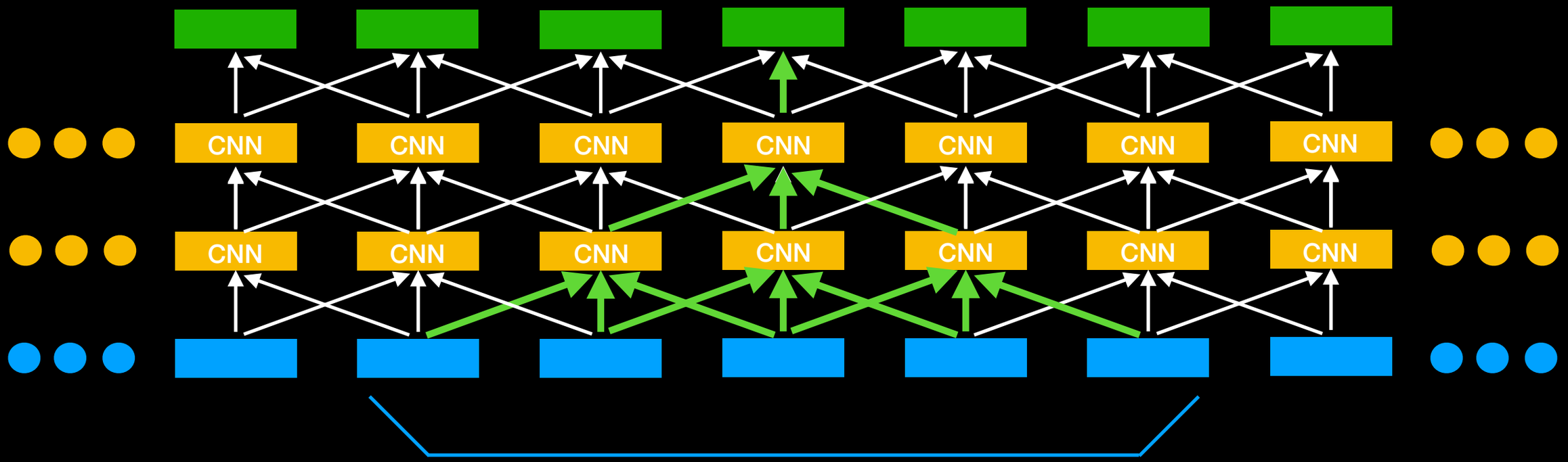
From this building block, the following architecture results:



CNN

In addition, the ”effective” receptive field can be increased by stacking 1D convolutional layers.

The green arrows show how the “effective” receptive field has now grown from three to five, by stacking an additional 1D convolutional layer.

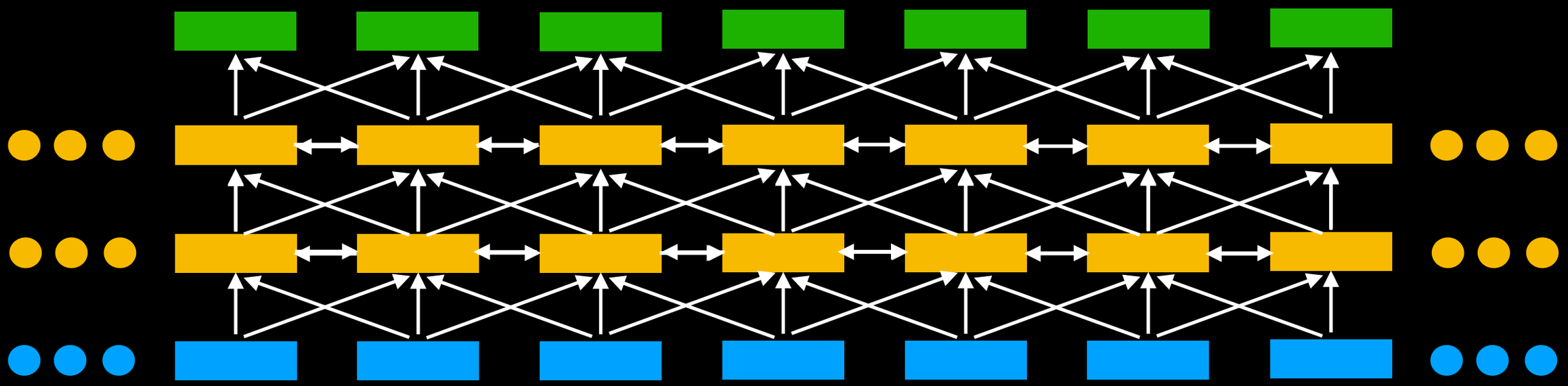


General Formulation

In general, we would like to train an architecture (using supervised learning) to predict a desired output sequence for a given input sequence.

We observe that RNN's (and it's advanced constructs - GRU, LSTM) process and acquire information by chaining additional RNN units.

We observe that CNN's process and acquire information by increasing it's receptive field using either wider convolutional filters or layer stacking.



However, both approaches have shortcomings.

Shortcomings of RNN's and CNN's

Suppose we had a very long input sequence (=high dimensional sequence):

- "Jane went running on Friday, up 10 hills. Some hills were muddy, other were rocky and others were slippery. In addition, the weather conditions were hot, dry and windy. All in all, the running conditions were not pleasant. She did not have a good run."

- Suppose we wanted to focus on Jane.

- We observe that Jane is mentioned twice in the paragraph.

- However, a large number of words separate the two "mentions".

Shortcomings of RNN's and CNN's

- In an RNN architecture, each word in the paragraph is processed sequentially.

- By construction, the process will be slow, since many words lie between Jane and she.

- In a CNN architecture, a large number of stacked convolutional layers will be required, so that the receptive field "covers" Jane and she.

- Clearly, both architectures are non-ideal, for long input sequences.

Improved

- Now, suppose we were able to look at / scan the entire paragraph and assign “attention” to the important pairs of words.

- Clearly, long sentences would benefit greatly from such an “attention” construct.

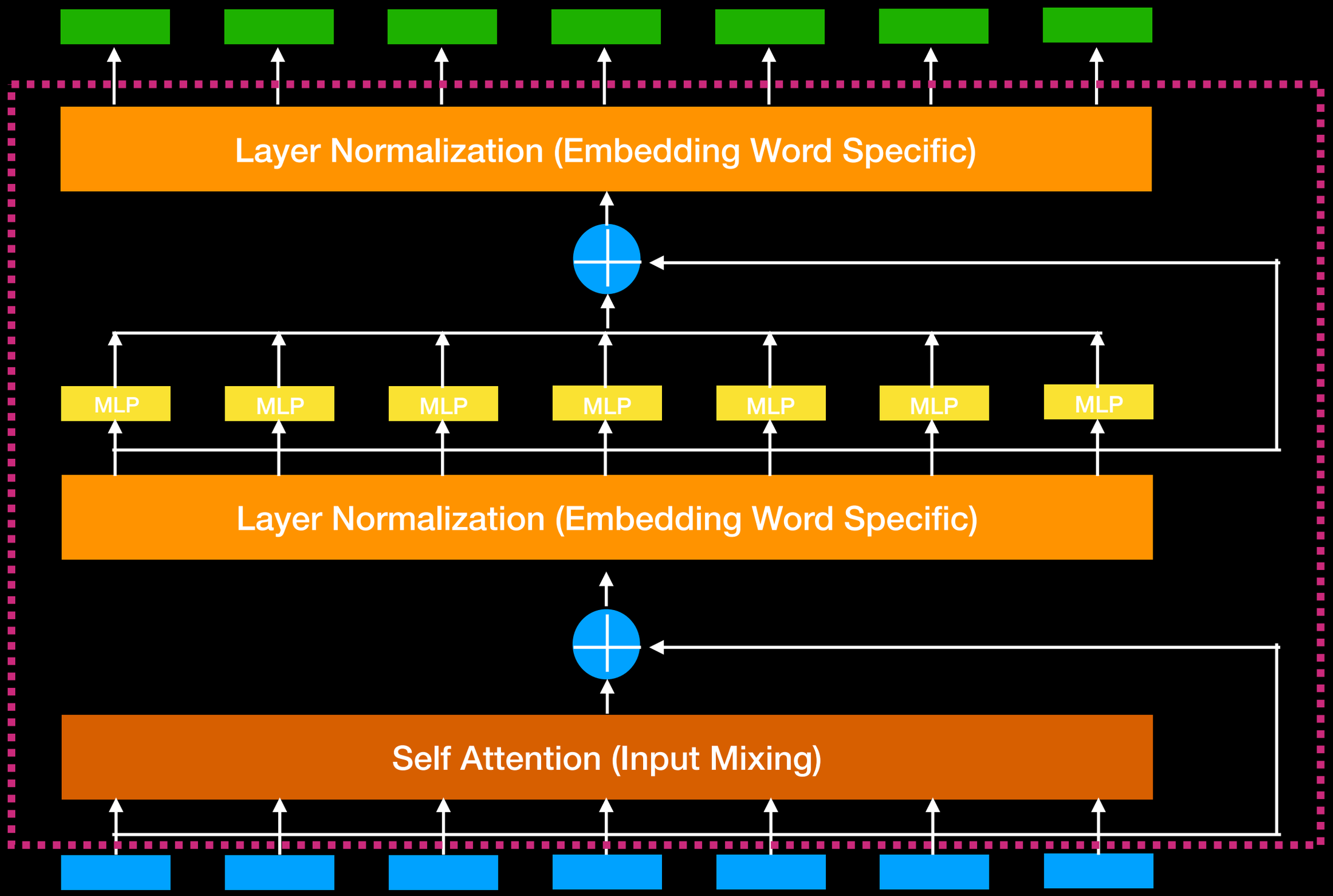
- The same is true for images, where words are now replaced by image patches.

- Transformers incorporate such an attention construct.

Transformer

- A transformer is created by layering transformer blocks.
- A transformer block consists of the following components: an attention mechanism, layer normalization and a feed forward network.
- The attention mechanism computes a “weighted” relationship between the various words (tokens).
- For pairwise attention, a 2D correlation matrix results.
- A transformer block is shown in the next slide.

Transformer Block



Self Attention
(The Creation of the 2D Correlation Matrix)

- Suppose a word (token) is represented by an embedding vector e.

- Suppose there are n words (=n embedding vectors).

- **Self attention performs an inner product between all possible word pair combinations.**

- The results are (then) normalized using a softmax.

- New embedding vectors are (then) created using a softmax weighted version of the original embedding vectors.

Self Attention

We represent the k dimensional embedding vector as:

$$e_{i,k} = [x_{i,1},\ x_{i,2}, \ldots, x_{i,k}]$$

$$E = \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{bmatrix} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,k} \\ x_{1,1} & x_{2,2} & \cdots & x_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n,1} & x_{n,2} & \cdots & x_{n,k} \end{bmatrix}$$

We compute the attention quantities, by performing inner products with all possible embedding vector combinations.

$$a_{i,j} = e_i \cdot e_j,\ \ \forall i, j \in \{1, 2, \ldots, n\}$$

$$A = EE^T = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,n} \end{bmatrix}$$

Self Attention

We normalize the results using a softmax:

$$softmax(a_i) = soft\max([a_{i,1}, a_{i,2}, \ldots, a_{i,n}]) = [\bar{a}_{i,1},\ \bar{a}_{i,2}, \ldots, \bar{a}_{i,n}]$$

$$\bar{A} = \begin{bmatrix} \bar{a}_{11} & \bar{a}_{12} & \ldots & \bar{a}_{1n} \\ \bar{a}_{21} & \bar{a}_{22} & \ldots & \bar{a}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \bar{a}_{n,1} & \bar{a}_{n,2} & \ldots & \bar{a}_{n,n} \end{bmatrix}$$

We compute new embedding vectors v of dim k, that are the the softmax weighted sum of the original embedding vectors.

$$V = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} = \bar{A}E = \begin{bmatrix} \bar{a}_{1,1} & \bar{a}_{1,2} & \ldots & \bar{a}_{1,n} \\ \bar{a}_{1,1} & \bar{a}_{2,2} & \ldots & \bar{a}_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ \bar{a}_{n,1} & \bar{a}_{n,2} & \ldots & \bar{a}_{n,n} \end{bmatrix} \begin{bmatrix} x_{1,1} & x_{1,2} & \ldots & x_{1,k} \\ x_{2,1} & x_{2,2} & \ldots & x_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n,1} & x_{n,2} & \ldots & x_{n,k} \end{bmatrix}$$

Self Attention

- In practice, the embedding vectors undergo three separate transformations, using three learned weight matrices Wq, Wk and Wv.
- For example, Wq, Wk and Wv can be of dimension m x k, where m is a hyperparameter.
- q = Wq * e
- k = Wk * e
- v = Wv * e
- The previous operations then follow.

Self Attention

$$q_i = W_q e_i$$

$$k_i = W_k e_i$$

$$z_i = W_v e_i$$

$$w^{'}_{i,j} = q_i^T k_j$$

$$w_{i,j} = softmax(w^{'}_{i,j})$$

$$v_i = \sum_j w_{i,j} z_j$$

Attention Heads

- However, the current attention mechanism only “partially” addressed an underlying problem in NLP.

- Consider the following two sentences:

  The Giants Beat The Dodgers

  The Dodgers Beat The Giants

- In the above example, the newly computed embedding vector for the word (token) Beat is the same in both sentences.

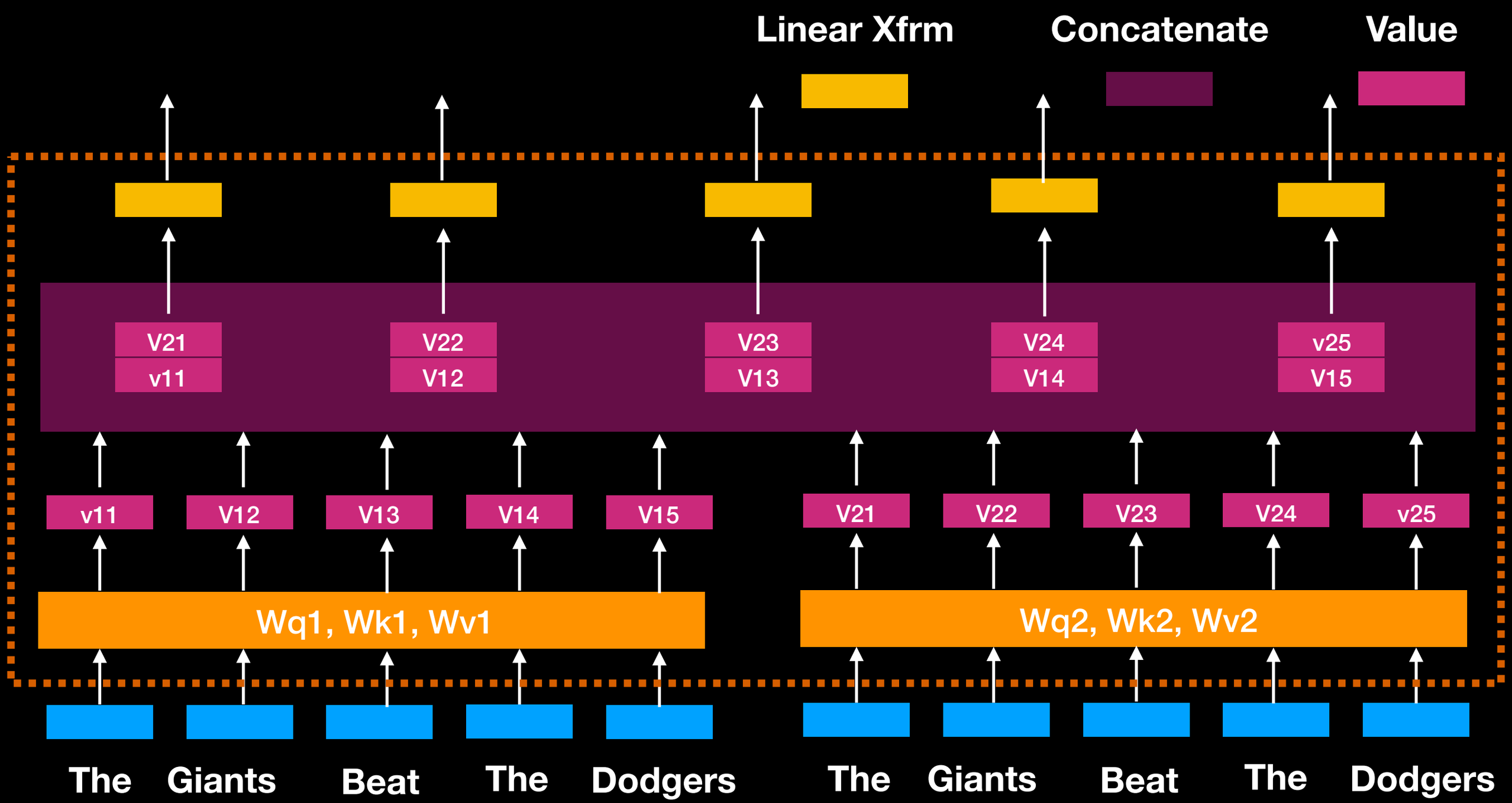
- Yet, the sentences conveyed very different information.

- To provide greater discrimination power (= to tease apart the nuances), multiple attention heads were added.

Attention Heads

- With each additional attention head, a new set of matrices (Wq, Wk and Wv) were required.
- With each additional attention head, a new embedding vector was produced for each input embedding (=word).
- In the end, the new embeddings need to be appropriately aggregated, producing ouput tensors with dimensions identical to the single attention head case.
- This was achieved using concatenation and a “dimension modifying” linear transformation.

Self Attention (MHSA)
2 Head Example



Self Attention

- In general, by adding multiple attention heads and stacked transformer blocks, superior results were achieved in machine translation.

- In general, the new architecture utilized a large number of weight parameters (Wq, Wk, Wv, linear transforms and MLP) that “captured” the information contained in the training examples.

- Although stacked transformer blocks increased performance, the resulting architecture size increased dramatically.

- In 2021, typical tranformer architectures ranged from several hundred million parameters to 140 billion parameters, with no upper bound in sight.

Other Details: Layer Normalization & MLP

Layer normalization is normalization with respect to the input vector / feature vector.

In general, normalization makes training via backpropagation easier and faster.

$$x_{i_{norm}} = \frac{x_i - \mu}{\sigma}$$

$$\mu = \frac{1}{N}\sum_i x_i$$

$$\sigma = \sqrt{\frac{1}{N}\sum_i (x_i - u)^2}$$

MLP's add valuable universal function approximation capabilities to an architecture. A two layer MLP is given by:

$$FFN(x) = max(0,\ xW_1 + b_1)W_2 + b_2$$

The Race Is On

- Thus far, 4 major players have emerged in the “transformer race”: Google, Facebook, Nvidia and OpenAI

- The networks trained include: BERT, RoBERTa, Megatron and GPT

- GPT-3 is currently the largest trained transformer architecture [2020], encompassing 175 billion parameters.

- The cost for training GPT-3 was estimated to be 4.6 million dollars.

The Application to Computer Vision

- Although transformers proved to be highly valuable for NLP tasks, more recent work has focused on vision based applications [2020 / 2021].

- The first major breakthough was Vision Transformer (ViT).

- Here, image recognition results exceeded current SOTA (ResNet-152x4) when training and tested using JFT-300M.

- The second major breakthrough followed shortly after, with Data Efficient Image Transformer (DeiT).

- Here, image recognition results comparable to and exceeding current SOTA (EfficientNet) was achieved when trained and tested using ONLY ImageNet21k. (Comparable => Add training tricks. Exceeding => Add distillation token.)

ViT

- ViT tried to mimick the transformer architecture in “Attention Is All You Need” as closely as possible.

- Here, unrolled image patches were used in place of input word embeddings.

- In addition, (unique) positional encoding vectors were added to each word embedding.

- Finally, a separate class embedding tensor (class token) was introduced (as input) into the transformer.

- The class token was a learned parameter that benefited from the attention mixing in the transformer blocks (MHSA).

ViT

- Note: In the current transformer instantiations, mixing ONLY occurs in the attention block.

- At inference, a class token (fixed) and input patches (variable) were fed into the transformer.

- The output class was then predicted using a MLP attached to the class token path of the transformer.

- ViT produced SOTA results for very large data sets (JFT-300M) and was highly competitive for Imagenet1k size data sets.

- The authors postulated that transformers trained on large datasets (JFT-300M) overcame the inductive biases present in CNN architectures (ResNet) trained on smaller dataset like ImageNet.

ViT

Predicted Class

Transformer with stacked transformer blocks (3 shown)

MLP

Class Token Path

Learned Class Token [1 x D]

Position Specific Vector [1 x D]

Linear Transform
[Input Dim: 1x (MN/9)]
[Output Dim: 1 x D]

Input [1 x (MN/9)]

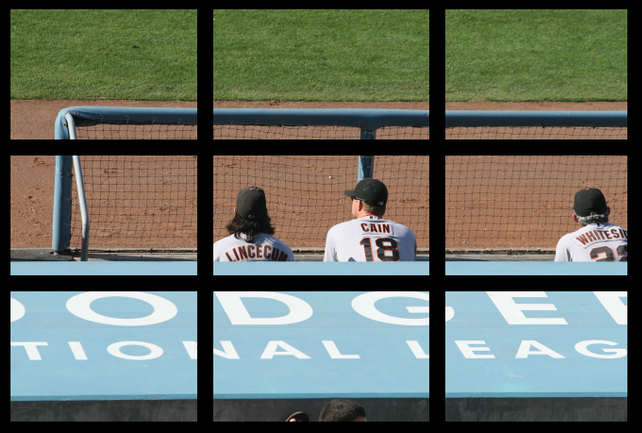
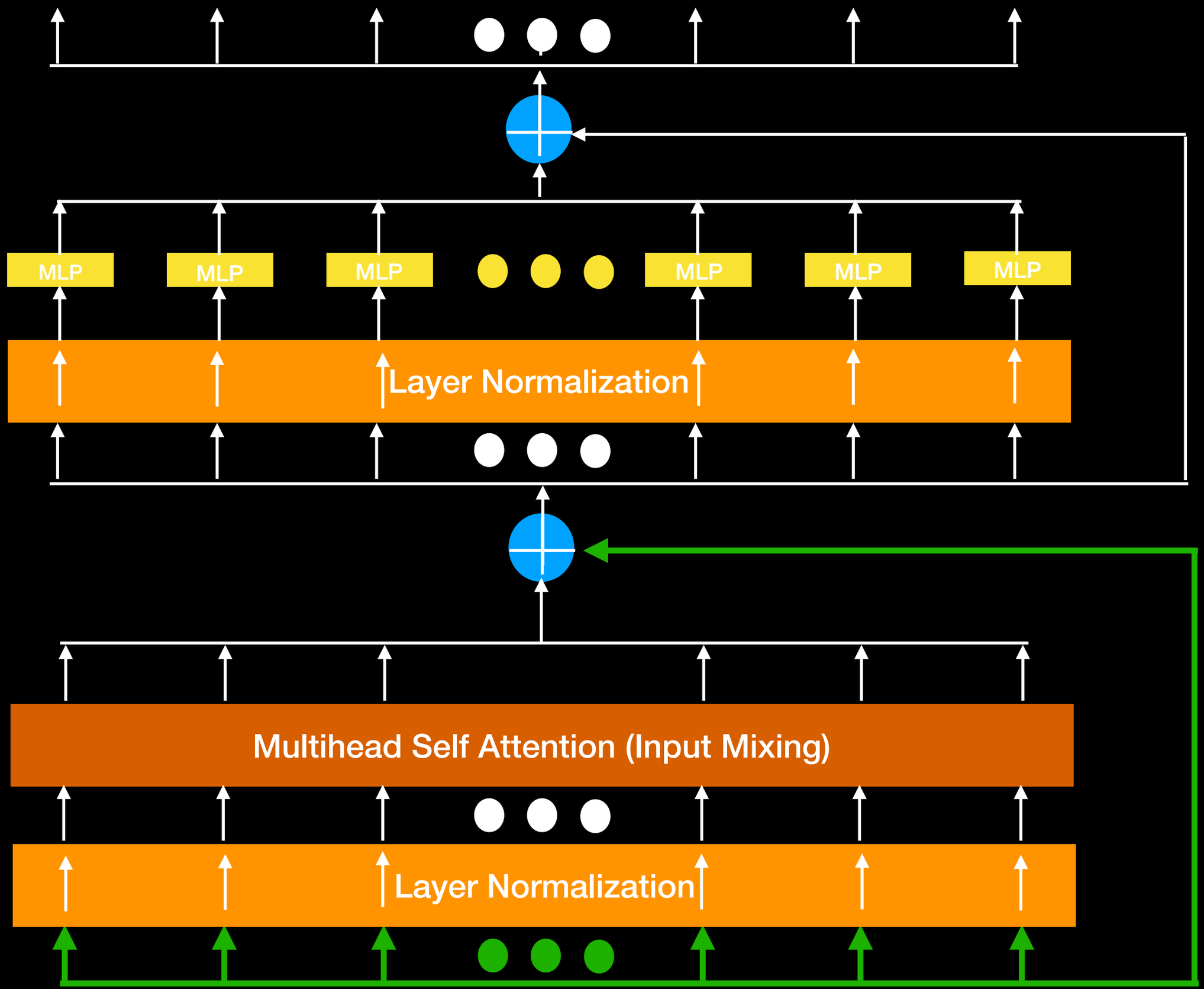


Image Dim: M x N
Patch Dim: M/3 x N/3
Patch Unrolled =
Input: 1 x (MN/9)

ViT (Transformer Block)



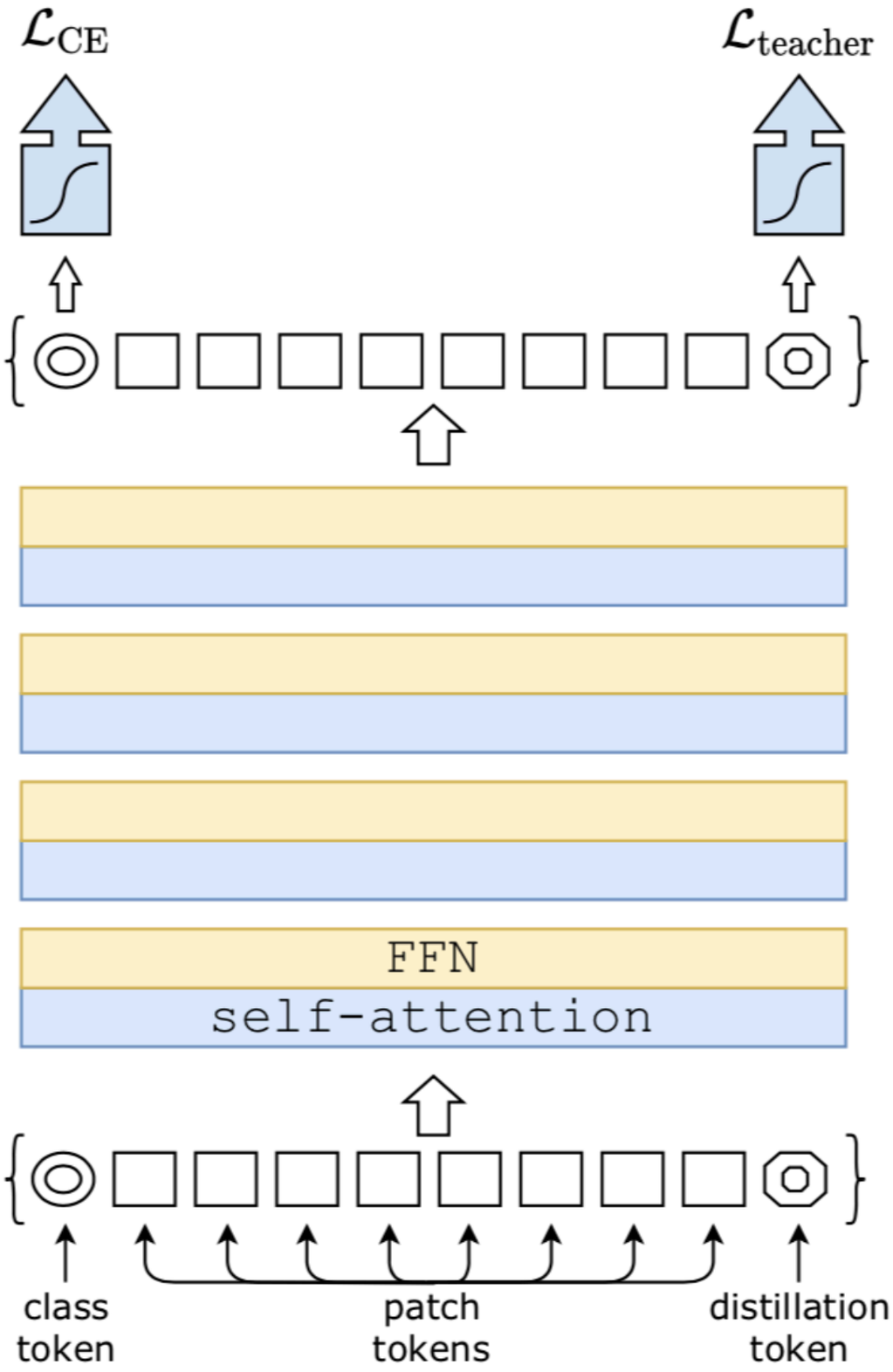
DeiT

- Two contributions were made in this paper.
- In the first contribution, the authors significantly improved the performance of ViT (86M+ parameters) using only ImageNet1k training data in a “data starved” scenario.
- In the second contribution, the authors further improved their result by adding a distillation token (in addition to the class token present in ViT) to the transformer.
- The authors then exploited hard teacher label distillation, to backpropagate into the tranformer.

Contribution1

- The authors changed the learning rate hyperparmeter from .003 to .001.
- The authors changed the weight decay hyperparameter from .3 to .05.
- The authors incorporated extensive data augmentation using AutoAugment data augmentation.
- The authors changed the optimizer from SGD to AdamW.
- The authors changed the weight initialization to a truncated normal distribution.
- The aggregation of these changes resulted in a 5% in Top 1% Accuracy improvement for ImageNet1k over the original ViT results.

Contribution2



Contribution2

A distillation token is added to the transformer input.

Using a CNN based teacher, the network tries to learn a distillation token that will reproduce the hard labels output by the CNN based teacher (instead of the true labels associated with a given input image).

The aim of the class and distillation tokens is to produce similar (but not identical) outputs for tokens with cosine similarity ~0.

Fine tuning is performed using teacher predictions with high resolution inputs.

A prediction is made by applying a softmax to the classifier outputs associated with the class and distillation tokens paths.

Contribution2

- The distillation token plays the same role as the class token in ViT.
- Like the class token in ViT, the distillation token interacts in the transformer through attention.
- The best teacher is the output from a trained convnet (and not the real labels themselves) since data augmentation can result in images that have label mismatches.
- Networks fine tuned using higher resolution images produced more accurate results when using the current scheme with employing extensive data augmentation.
- Contribution1 + Contribution2 result in SOTA Top 1% Accuracy versus EfficientNet for ImageNet1k training and test data.

Knowledge Distillation

- The authors goal was to use knowledge distillation (per Abnar), to improve the classifier outputs associated with the class and distillation tokens.

- This required replacing the true image labels with teacher based image labels, computed from an external CNN.

- With this action, the knowledge from the teacher was then distilled into the student / transformer.

Knowledge Distillation

- The loss function employed during training was:

$$\mathcal{L}_{\text{global}}^{\text{hardDistill}} = \frac{1}{2}\mathcal{L}_{\text{CE}}(\psi(Z_s), y) + \frac{1}{2}\mathcal{L}_{\text{CE}}(\psi(Z_s), y_{\text{t}}).$$

- Where CE denoted the cross entropy loss, psi denoted the softmax function, Z_s denoted the logits from the student model, y was the true label and y_t denoted the hard decision labels produced by the teacher (via argmax).

References

- Squeeze and Excitation Networks, Hu, et. al., arXiv September 2017
- Selective Kernel Networks, Li, et. al., CVPR 2019
- ResNeSt: Split Attention Networks, Zhang, et. al., arXiv 2020
- Global Self Attention Networks for Image Recognition, Shen, et. al., arXiv October 2020
- Neural Machine translation by Jointly Learning to Align and Translate, Bahdanau et. al., ICLR 2015
- Attention Is All You Need, Vaswani, et. al., arXiv June 2017
- BERT, Roberta, Megatron, GPT
- An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale, Dosovitskiy, et. al., October 2020
- Training Data Efficient Transformers and Distillation Through Attention, Touvron, et. al., arXiv January 2021
- Transferring Inductive Biases Through Knowledge Distillation, Abnar, arXiv October 2020
- End to End Object Detection with Transformers, Carion, et. al., ECCV 2020