MixA: A Mixed Attention approach with Stable Lightweight Linear Attention to enhance Efficiency of Vision Transformers at the Edge

Supplementary Material

A. Overview

The supplementary material is structured as follows:

- In Section B, we present the proofs for Theorem 5.1, Theorem 5.2 and Theorem 5.3.
- In Section C, we present implementation details for three CV tasks we evaluated.

B. Proofs

Theorem 5.1. Consider the query and key matrices $\mathbf{Q}, \mathbf{K} \in \mathbb{R}^{N\times d}$ and assume that the components of query $\mathbf{Q}_i \in \mathbb{R}^d$ and key $\mathbf{K}_j \in \mathbb{R}^d$ are independent random variables following standard normal distribution. Then the variance of dot-product between $\mathbf{Q}_i$ and $\mathbf{K}_j$ follows

$$\text{Var}(\mathbf{Q}_i^T\mathbf{K}_j) \in \mathcal{O}(d)$$

where d is the dimension of the attention head.

Proof of Theorem 5.1. The dot product $\mathbf{Q}_i^T\mathbf{K}_j$ can be written as:

$$\mathbf{Q}_i^T\mathbf{K}_j = \sum_{k=1}^{d} \mathbf{Q}_{ik}\mathbf{K}_{jk}.$$

Since each $\mathbf{Q}_{ik}\mathbf{K}_{jk}$ term is uncorrelated with $\mathbf{Q}_{il}\mathbf{K}_{jl}$ for $k \neq l$ (due to the independence of $\mathbf{Q}$ and $\mathbf{K}$ components), we can sum the variances of each term individually:

$$\text{Var}(\mathbf{Q}_i^T\mathbf{K}_j) = \sum_{k=1}^{d} \text{Var}(\mathbf{Q}_{ik}\mathbf{K}_{jk}).$$

Since $\mathbf{Q}_{ik}$ and $\mathbf{K}_{jk}$ are independent standard normal variables, variance of their product $\mathbf{Q}_{ik}\mathbf{K}_{jk}$ can be calculated as follows

$$\text{Var}(\mathbf{Q}_{ik}\mathbf{K}_{jk}) = \mathbb{E}[\mathbf{Q}_{ik}^2] \cdot \mathbb{E}[\mathbf{K}_{jk}^2] = 1 \cdot 1 = 1.$$

Since each $\text{Var}(\mathbf{Q}_{ik}\mathbf{K}_{jk}) = 1$, the total variance is:

$$\text{Var}(\mathbf{Q}_i^T\mathbf{K}_j) = \sum_{k=1}^{d} 1 = d.$$

This shows that the variance of the dot product scales as $\mathcal{O}(d)$. □

Theorem 5.2. Consider an attention matrix $\mathbf{A} \in \mathbb{R}^{N\times N}$ and value matrix $\mathbf{V} \in \mathbb{R}^{N\times d}$. Assume that the variance of product terms is bounded, i.e., $\text{Var}(\mathbf{A}_{ij}\mathbf{V}_{jk}) \leq C$ for some constant C. Then the variance of elements of the resultant matrix, i.e., $\mathbf{O}_{ik} = \sum_{j=1}^{N} \mathbf{A}_{ij}\mathbf{V}_{jk}$ has the following growth rate

$$\text{Var}(\mathbf{O}_{ik}) \in \mathcal{O}(N^2)$$

where N is the number of tokens in the sequence.

Proof of Theorem 5.2. By using triangle inequality and Cauchy-Schwarz we get the following

$$\begin{aligned}
\text{Var}(\mathbf{O}_{ik}) &= \text{Var}\left(\sum_{j=1}^{N} \mathbf{A}_{ij}\mathbf{V}_{jk}\right) \\
&= \sum_{j=1}^{N}\sum_{l=1}^{N} \text{Cov}(\mathbf{A}_{ij}\mathbf{V}_{jk}, \mathbf{A}_{il}\mathbf{V}_{lk}) \\
&\leq \sum_{j=1}^{N}\sum_{l=1}^{N} |\text{Cov}(\mathbf{A}_{ij}\mathbf{V}_{jk}, \mathbf{A}_{il}\mathbf{V}_{lk})| \\
&\leq \sum_{j=1}^{N}\sum_{l=1}^{N} \sqrt{\text{Var}(\mathbf{A}_{ij}\mathbf{V}_{jk})\text{Var}(\mathbf{A}_{il}\mathbf{V}_{lk})} \\
&\leq \sum_{j=1}^{N}\sum_{l=1}^{N} \sqrt{C \cdot C} \\
&= CN^2
\end{aligned}$$

This shows that the variance of $\mathbf{O}_{ik}$ scales as $\mathcal{O}(N^2)$. □

Theorem 5.3. Consider the query and key matrices $\mathbf{Q}, \mathbf{K} \in \mathbb{R}^{N\times d}$ and assume that the components of query $\mathbf{Q}_i \in \mathbb{R}^d$ and key $\mathbf{K}_j \in \mathbb{R}^d$ are independent random variables following standard normal distribution. Then the variance of dot-product between $\text{ReLU}(\mathbf{Q}_i)$ and $\text{ReLU}(\mathbf{K}_j)$ vectors follows

$$\text{Var}(\text{ReLU}(\mathbf{Q}_i)^T\text{ReLU}(\mathbf{K}_j)) \in \mathcal{O}(d)$$

where d is the dimension of the attention head.

Proof of Theorem 5.3. The dot product between $\text{ReLU}(\mathbf{Q}_i)$ and $\text{ReLU}(\mathbf{K}_j)$ can be written as:

$$\text{ReLU}(\mathbf{Q}_i)^T\text{ReLU}(\mathbf{K}_j) = \sum_{k=1}^{d} \text{ReLU}(\mathbf{Q}_{ik})\text{ReLU}(\mathbf{K}_{jk}),$$

where each component $\mathbf{Q}_{ik}$ and $\mathbf{K}_{jk}$ is an independent standard normal random variable. Since $\text{ReLU}(\mathbf{Q}_{ik})$ and

$\text{ReLU}(\mathbf{K}_{jk})$ are also independent, we can sum the variances of each term individually:

$$\begin{aligned}&\text{Var}(\text{ReLU}(\mathbf{Q}_i)^T\text{ReLU}(\mathbf{K}_j))\\&=\sum_{k=1}^{d}\text{Var}(\text{ReLU}(\mathbf{Q}_{ik})\text{ReLU}(\mathbf{K}_{jk})).\end{aligned}$$

Now, for a standard normal variable $X \sim \mathcal{N}(0,1)$, we have the following:

$$\begin{aligned}\mathbb{E}[\text{ReLU}(X)] &= \int_0^\infty x\frac{1}{\sqrt{2\pi}}e^{-\frac{x^2}{2}}\,dx = \frac{1}{\sqrt{2\pi}},\\ \mathbb{E}[\text{ReLU}(X)^2] &= \int_0^\infty x^2\frac{1}{\sqrt{2\pi}}e^{-\frac{x^2}{2}}\,dx = \frac{1}{2}.\end{aligned}$$

Using these results and the independence of $\text{ReLU}(\mathbf{Q}_{ik})$ and $\text{ReLU}(\mathbf{K}_{jk})$, we have the following:

$$\begin{aligned}&\text{Var}(\text{ReLU}(\mathbf{Q}_{ik})\text{ReLU}(\mathbf{K}_{jk}))\\&=\mathbb{E}[\text{ReLU}(\mathbf{Q}_{ik})^2]\cdot\mathbb{E}[\text{ReLU}(\mathbf{K}_{jk})^2]-\\&\quad(\mathbb{E}[\text{ReLU}(\mathbf{Q}_{ik})]\cdot\mathbb{E}[\text{ReLU}(\mathbf{K}_{jk})])^2\\&=\frac{1}{2}\cdot\frac{1}{2}-\left(\frac{1}{\sqrt{2\pi}}\right)^2=\frac{1}{4}-\frac{1}{2\pi}\end{aligned}$$

Therefore, the total variance is:

$$\begin{aligned}\text{Var}(\text{ReLU}(\mathbf{Q}_i)^T\text{ReLU}(\mathbf{K}_j)) &= \sum_{k=1}^{d}\left(\frac{1}{4}-\frac{1}{2\pi}\right)\\&= d\left(\frac{1}{4}-\frac{1}{2\pi}\right).\end{aligned}$$

This shows that the variance of the dot product scales as $\mathcal{O}(d)$. □

C. Additional Implementation Details

For classification task, we take pretrained ViT models [17, 28] and fine-tune them after applying MixA. For finetuning the models, we utilize the cross-entropy loss and standard knowledge distillation loss [12], using pretrained models as teachers. This fine-tuning process is conducted over 150 epochs with the AdamW optimizer [18] and a cosine learning rate schedule, including 10 warm-up epochs with a base learning rate of 1×10^{-4}. For DeiT models we use a batch size of 512 and for Swin-T and Swin-S models we use a batch size of 384 and 256 respectively. To avoid overfitting, we follow DeiT [28] and apply RandAugment [6] and and random erasing [39] for DeiT-T. And for DeiT-S, and Swin models we use the standard Mixup [37] and CutMix [36] augmentations as used in DeiT [28]. In addition, a weight decay of 0.05 is also used for all model training. For DeiT models [28], we use a patch size of 14×14 similar to [21] and for Swin models we use a window size of 14×14. For fair comparison, we carry out exact fine-tuning process and finetune the pretrained ViT models with softmax-based quadratic attention under the same settings and report their results. Similarly, we carry out the same fine-tuning process to report performance of existing linear attention mechanisms [13, 15, 23]. For CosFormer [23], we report performance of ReLU Linear attention without the cosine re-weighting similar to [15]. Similar to [15], we also found that CosFormer does not converge with the cosine reweighting part.

For object detection on the COCO dataset, we use Faster R-CNN [24] model with respective ViT backbones, initializing the ViT backbones by loading the corresponding classification checkpoints. For DeiT-T [28], we adopt the ViT-Adapter [4] configuration due to its superior performance. We train the models using the AdamW optimizer [18] with a base learning rate of 1×10^{-4} and a weight decay of 0.05. We apply a layer-wise decay rate of 0.6 across 12 transformer layers. The learning rate follows a step decay schedule, starting with a linear warmup for 3000 iterations and decaying at epochs 27 and 33. We train the models for 36 epochs with a batch size of 16 on a single GPU, using an input resolution of 448×448 for both training and evaluation. We evaluate the performance every epoch using the mean Average Precision (mAP) metric and report the best mAP score.

For semantic segmentation on the ADE20K dataset, we use the SemanticFPN model with ViT backbones, initializing the ViT backbones by loading the corresponding classification checkpoints. For DeiT-T [28], we adopt the ViT-Adapter [4] configuration due to its superior performance. We train the models using the AdamW optimizer [18] with a base learning rate of 2×10^{-4} and a weight decay of 1×10^{-4}. A polynomial learning rate decay schedule with a power of 0.9 is employed. We train the models for 40K iterations with a batch size of 16 on a single GPU, using an input resolution of 448×448 for both training and evaluation. We evaluate the models every 4000 iterations using the mean Intersection over Union (mIoU) metric and report the best mIoU score.

We set the the number of quadratic attention layers, i.e., $k = 6$ for MixA across all models and all tasks.