Transformer Learns Optimal Variable Selection in Group-Sparse Classification

Chenyang Zhang[1] Xuran Meng[2] Yuan Cao[1]

[1]Department of Statistics & Actuarial Science, The University of Hong Kong
[2]Department of Biostatistics, University of Michigan, Ann Arbor

Transformers are proficient at structure data

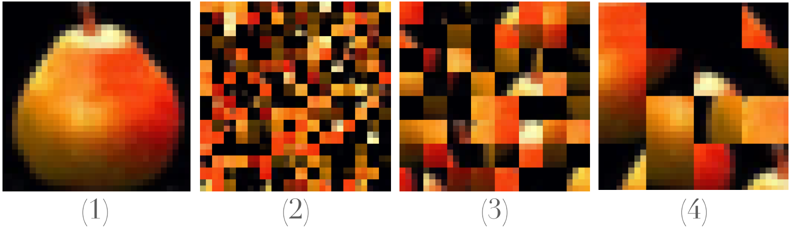
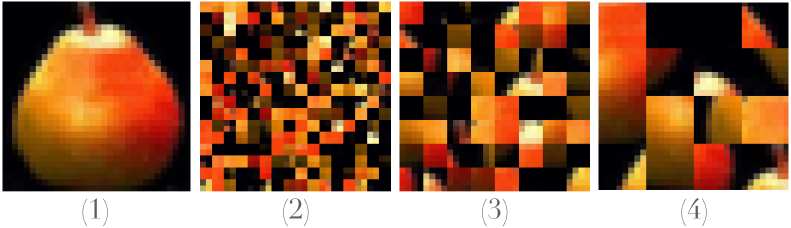


Figure: An example from Jelassi et al. (2022)

- Attention layer of **transformers** (Vaswani et al., 2017) can effectively extract structure information from data in multiple tasks, containing computer vision (Jelassi et al. 2022), language process (Vaswani et al., 2017), and token selection (Wang et al. 2024).

Transformers are proficient at structure data



(1) (2) (3) (4)

Figure: An example from Jelassi et al. (2022)

- Attention layer of **transformers** (Vaswani et al., 2017) can effectively extract structure information from data in multiple tasks, containing computer vision (Jelassi et al. 2022), language process (Vaswani et al., 2017), and token selection (Wang et al. 2024).

Our question: Are transformers capable of utilizing the attention mechanism to conduct variable selection, a classic statistical task?

Group sparse learning problem

- **Generalized linear group sparse model**
  - For each feature vector $\hat{\mathbf{x}} \in \mathbb{R}^p$, its label is determined as $y = \phi(\langle \hat{\mathbf{x}}, \boldsymbol{\beta}^* \rangle)$, where $\boldsymbol{\beta}^*$ is the ground-truth parameter vector and $\phi(\cdot)$ is the labeling function.
  - There exists a pre-defined D disjoint partitions of $[p]$ s.t. $[p] = \cup_{j=1}^{D} G_j$.
  - This learning problem is defined as "group sparse" if $\boldsymbol{\beta}^*$ satisfying that

$$\mathrm{supp}(\boldsymbol{\beta}^*) := \{k \in [p] : \beta_k^* \neq 0\} \subset G_{j^*},$$

where $j^* \in [D]$ is the index of label-relevant group.

Group sparse learning problem

- **Group sparse linear classification with Gaussian inputs**
  - Let $p = dD$ with d denoting the dimension of each group. Then, reshape the feature vector $\hat{\mathbf{x}}$ into $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_D]$, where each column $\mathbf{x}_j \overset{\text{i.i.d}}{\sim} \mathcal{N}(\mathbf{0}, \sigma_x^2 \mathbf{I}_d)$.
  - The label of $\mathbf{X}$ is determined as $y = \operatorname{sign}(\langle \mathbf{x}_{j^*}, \mathbf{v}^* \rangle)$. Here, $\mathbf{v}^* \in \mathbb{R}^d$ consists of the entries of $\boldsymbol{\beta}^*$ at the positions corresponding to the index set G_{j^*}.

  We denote the data model distribution above as $\mathcal{D}$.
- **Concatenating positional encodings to input matrix $\mathbf{X}$**

$$\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_D] = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \ldots & \mathbf{x}_D \\ \mathbf{p}_1 & \mathbf{p}_2 & \ldots & \mathbf{p}_D \end{bmatrix} \in \mathbb{R}^{(d+D)\times D},$$

where $\mathbf{p_j} \in \mathbb{R}^D$ is the orthogonal positional encoding vector.

One-layer self-attention and training algorithm

- **One-layer self-attention architecture**

$$f(\mathbf{Z}, \mathbf{v}, \mathbf{W}) = \sum_{j=1}^{D} \mathbf{v}^\top \mathbf{Z} \cdot \mathrm{softmax}(\mathbf{Z}^\top \mathbf{W} \mathbf{z}_j) = \mathbf{v}^\top \mathbf{Z} \mathbf{S} \mathbf{1}_D.$$

$\mathbf{S}$ is the attention score matrix obtained after softmax calculation. Compared to the classic one-layer attention, we make some mild re-parameterizations on the architecture:

  - Combine the query and key matrices into one trainable matrix $\mathbf{W} \in \mathbb{R}^{(d+D)\times(d+D)}$.
  - Replace the value matrix with one trainable value vector $\mathbf{v} \in \mathbb{R}^{d+D}$.

One-layer self-attention and training algorithm

- **Loss function and training algorithm**
  - Consider minimizing the population cross-entropy loss, which is defined as

$$\mathcal{L}(\mathbf{v}, \mathbf{W}) = \mathbb{E}_{(\mathbf{X},y)\sim\mathcal{D}}\big[\ell(y \cdot f(\mathbf{Z}, \mathbf{v}, \mathbf{W}))\big],$$

    where $\ell(a) = \log(1 + \exp(-a))$ is the cross-entropy loss function for binary classification.

  - Optimize the objective loss function using gradient descent as follows:

$$\mathbf{v}^{(t+1)} = \mathbf{v}^{(t)} - \eta\nabla_{\mathbf{v}}\mathcal{L}(\mathbf{v}^{(t)}, \mathbf{W}^{(t)});\ \ \mathbf{W}^{(t+1)} = \mathbf{W}^{(t)} - \eta\nabla_{\mathbf{W}}\mathcal{L}(\mathbf{v}^{(t)}, \mathbf{W}^{(t)}),$$

    where $\mathbf{v}^{(0)} = \mathbf{0}_{d+D}$ and $\mathbf{W}^{(0)} = \mathbf{0}_{(d+D)\times(d+D)}$.

Transformers learn "group sparse" data

Theorem

For any $\epsilon > 0$*, suppose that* $D \geq \omega(\log^2(1/\epsilon))$*,* $d \leq O(\mathrm{poly}(D))$*,* $\sigma_x, \eta = \Theta(1)$ *with* $\sigma_x \leq 1/3$ *and let* $T^* = \Theta\big(D^3 \vee \frac{1}{D^3\epsilon^3}\big)$*. Under these conditions, the following conclusions hold:*

1. *The self-attention extracts the variables from the label-relevant group: with probability at least* $1 - \exp(-\Theta(\sqrt{D}))$,
$$\mathbf{S}_{j^*,j}^{(T^*)} \geq 1 - \exp(-\Theta(D)), \ \forall j \in [D].$$
2. *Value vector* $\mathbf{v}$ *successfully learn the ground truth classifier* $\mathbf{v}^*$*:* $\mathbf{v}^{(T^*)} = [\mathbf{v}_1^{(T^*)}, \mathbf{0}_D]$*, and*
$$\left\|\mathrm{normalized}(\mathbf{v}_1^{(T^*)}) - \mathbf{v}^*\right\|_2 \leq \epsilon D \exp(-\Theta(\sqrt{D})).$$
3. *The loss is sufficiently minimized:*
$$\mathcal{L}(\mathbf{v}^{(T^*)}, \mathbf{W}^{(T^*)}) = \Theta(\epsilon \wedge D^{-2}).$$

Downstream task and training algorithm

- **Downstream group sparse data model**

  The downstream training data set $\{(\tilde{\mathbf{X}}^{(i)}, \tilde{y}^{(i)})\}_{i=1}^{n}$ is generated from the following downstream distribution $\tilde{\mathcal{D}}$:

  - Label $\tilde{y}$ is linear separable w.r.t the variable from the j^*-th group: $\max_{\tilde{\mathbf{v}}^*:\|\tilde{\mathbf{v}}^*\|_2\le 1} \tilde{y}^{(i)} \cdot \langle \tilde{\mathbf{v}}^*, \tilde{\mathbf{x}}_{j^*}^{(i)} \rangle \ge \gamma$ almost surely for all $i \in [n]$.
  - Each entry of $\tilde{\mathbf{X}}^{(i)}$ is independent sub-Gaussian random variable, with $\left\|\tilde{\mathbf{x}}_{j,k}^{(i)}\right\|_{\psi_2} \le \tilde{\sigma}_x$.

  Note that for the downstream distribution $\tilde{\mathcal{D}}$, we only require the label-relevant group index j^* to be the same as that in $\mathcal{D}$, while the ground-truth vector $\tilde{\mathbf{v}}^*$ can differ from $\mathbf{v}^*$.

Downstream task and training algorithm

- **Fine-tuning with online stochastic gradient descent**
  - For training data set $\{(\tilde{\mathbf{X}}^{(i)}, \tilde{y}^{(i)})\}_{i=1}^{n}$, online SGD conducts total n iterations.
  - At each iteration $i \in [n]$, online SGD updates the parameters as

$$\tilde{\mathbf{v}}^{(i+1)} = \tilde{\mathbf{v}}^{(i)} - \tilde{\eta} \nabla_{\tilde{\mathbf{v}}} \tilde{\mathcal{L}}_i(\tilde{\mathbf{v}}^{(i)}, \tilde{\mathbf{W}}^{(i)});\ \tilde{\mathbf{W}}^{(i+1)} = \tilde{\mathbf{W}}^{(i)} - \tilde{\eta} \nabla_{\tilde{\mathbf{W}}} \tilde{\mathcal{L}}_i(\tilde{\mathbf{v}}^{(i)}, \tilde{\mathbf{W}}^{(i)}),$$

    where $\tilde{\mathcal{L}}_i(\tilde{\mathbf{v}}^{(i)}, \tilde{\mathbf{W}}^{(i)}) = \ell(\tilde{y}^{(i)} \cdot f(\tilde{\mathbf{Z}}^{(i)}, \tilde{\mathbf{v}}^{(i)}, \tilde{\mathbf{W}}^{(i)}))$ is the cross-entropy loss calculated only by the i-th training data point. The initializations are set as $\tilde{\mathbf{v}}^{(0)} = \mathbf{0}_{d+D}$ and $\tilde{\mathbf{W}}^{(0)} = \mathbf{W}^{(T^*)}$, which is obtained from the pre-trained model.

Generalization error bound on downstream task

Theorem

Suppose that $D \geq \omega(\log^2(n))$, $d \leq O(\mathrm{poly}(D))$, $\tilde{\sigma}_x \leq O(1)$, *and* $\tilde{\eta} = \Theta(\frac{1}{(d \vee D)D^2})$. *Under these conditions, and for any* $\delta > 0$, *with probability at least* $1 - \delta - n\exp(-\Theta(\sqrt{D}))$ *over the randomness of* $\{(\tilde{\mathbf{X}}^{(i)}, \tilde{y}^{(i)})\}_{i=1}^{n}$, *it holds that:*

$$\frac{1}{n}\sum_{i=1}^{n}\mathbb{P}_{(\tilde{\mathbf{X}},\tilde{y})\sim\tilde{\mathcal{D}}}\Big(\tilde{y}\cdot f(\tilde{\mathbf{Z}},\tilde{\mathbf{v}}^{(i)},\tilde{\mathbf{W}}^{(i)}) \leq 0\Big) \leq O\left(\frac{(d+D)\log^2 n}{\gamma^2 n}\right) + O\left(\frac{\log(1/\delta)}{n}\right),$$

- This result establishes a sample complexity at the scale of $\tilde{\Omega}(\frac{d+D}{\epsilon} + \frac{1}{\epsilon}\log(\frac{1}{\delta}))$
- In comparison, the existing lower bound for the sample complexity of linear logistic regression on vectorized $\mathbf{X}$ is $\Omega(\frac{dD}{\epsilon} + \frac{1}{\epsilon}\log(\frac{1}{\delta}))$

Experiments results of pre-training task

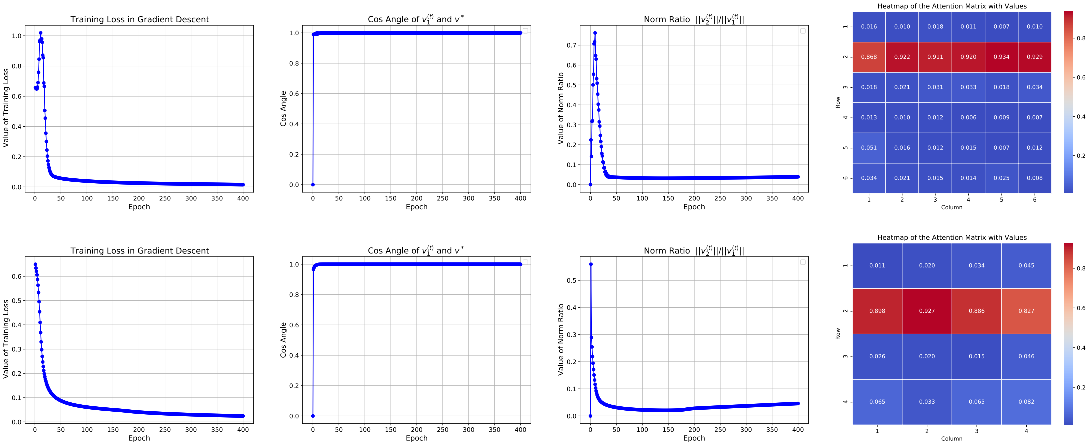
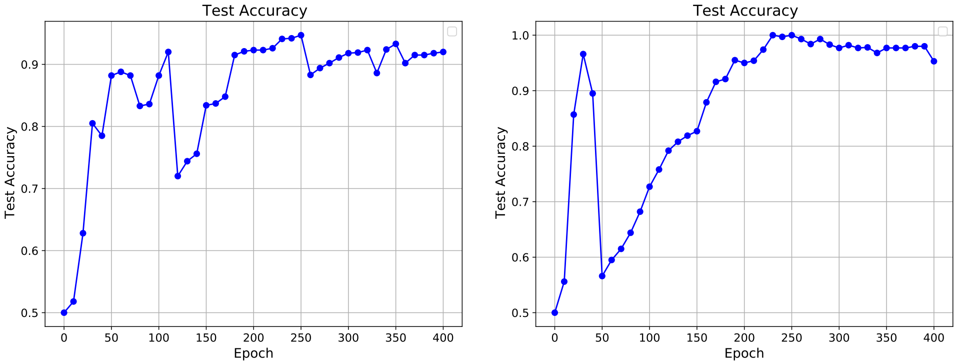


Figure: Training loss, cosine similarity, norm ratio, and attention score for $(n, d, D) = (500, 4, 6)$ and $(n, d, D) = (200, 2, 4)$ respectively when set $j^* = 2$.

Experiments results of downstream task



(a) (n,d,D)=(400,4,6)

(b) (n,d,D)=(400,2,4)

Figure: Test accuracy in the downstream task when utilizing the pre-trained $\mathbf{W}^{(T^*)}$ with the corresponding dimension as initialization.

Summary

- One-layer transformers can almost attend to the variables from the label-relevant group, and disregard other ones by leveraging the self-attention mechanism.
- Pre-trained one-layer transformers on a “group sparse” data can be effectively transferred to a downstream task with the same sparsity pattern. It achieves a sample complexity surpassing linear logistic regression applied to vectorized features.
- These findings align with the phenomenon observed in experiments when training one-layer transformers on “group sparse” data.

Summary

- One-layer transformers can almost attend to the variables from the label-relevant group, and disregard other ones by leveraging the self-attention mechanism.
- Pre-trained one-layer transformers on a "group sparse" data can be effectively transferred to a downstream task with the same sparsity pattern. It achieves a sample complexity surpassing linear logistic regression applied to vectorized features.
- These findings align with the phenomenon observed in experiments when training one-layer transformers on "group sparse" data.

Thank you!