Convex Neural Autoregressive Models: Towards Tractable, Expressive, and Theoretically-Backed Models for Sequential Forecasting and Prediction

Vikul Gupta, Burak Bartan, Tolga Ergen, Mert Pilanci

Department of Electrical Engineering, Stanford University

Poster ID: 5131

Preliminaries

We have a two-layer autoregressive network to model the sampling distribution of binary data $\boldsymbol{X} \in \{0,1\}^{N\times M}$. The task is to predict the d-th entry x_d given the previous r entries,

$$\boldsymbol{x}_{<d,(r)} = [x_{d-r-1}, x_{d-r}, \ldots, x_{d-1}] \in \mathbb{R}^r.$$

The model takes $\boldsymbol{x}_{<d,(r)}$ as input and outputs a prediction for the mean parameter of x_d's Bernoulli distribution:

$$\sigma(\boldsymbol{\alpha}^T\boldsymbol{h}) = \sigma(\boldsymbol{\alpha}^T(\boldsymbol{U}\boldsymbol{x}_{<d,(r)})_+),$$

where the first-layer weights are $\boldsymbol{U} \in \mathbb{R}^{m\times r}$, the second-layer weights are $\boldsymbol{\alpha} \in \mathbb{R}^m$, the hidden layer, $\boldsymbol{h} \in \mathbb{R}^m$, uses ReLU activation, and the output uses sigmoid activation. The training problem is then

$$\min_{\{\alpha_j,\boldsymbol{u}_j\}_{j=1}^m} \mathcal{L}\left(\boldsymbol{\alpha}^T(\boldsymbol{U}\boldsymbol{X}_{(r)}^T)_+, \boldsymbol{y}\right) + \frac{\beta}{2}\sum_{j=1}^m(||\boldsymbol{u}_j||_2^2 + \alpha_j^2),$$

where the loss $\mathcal{L}$ is binary cross-entropy (BCE).

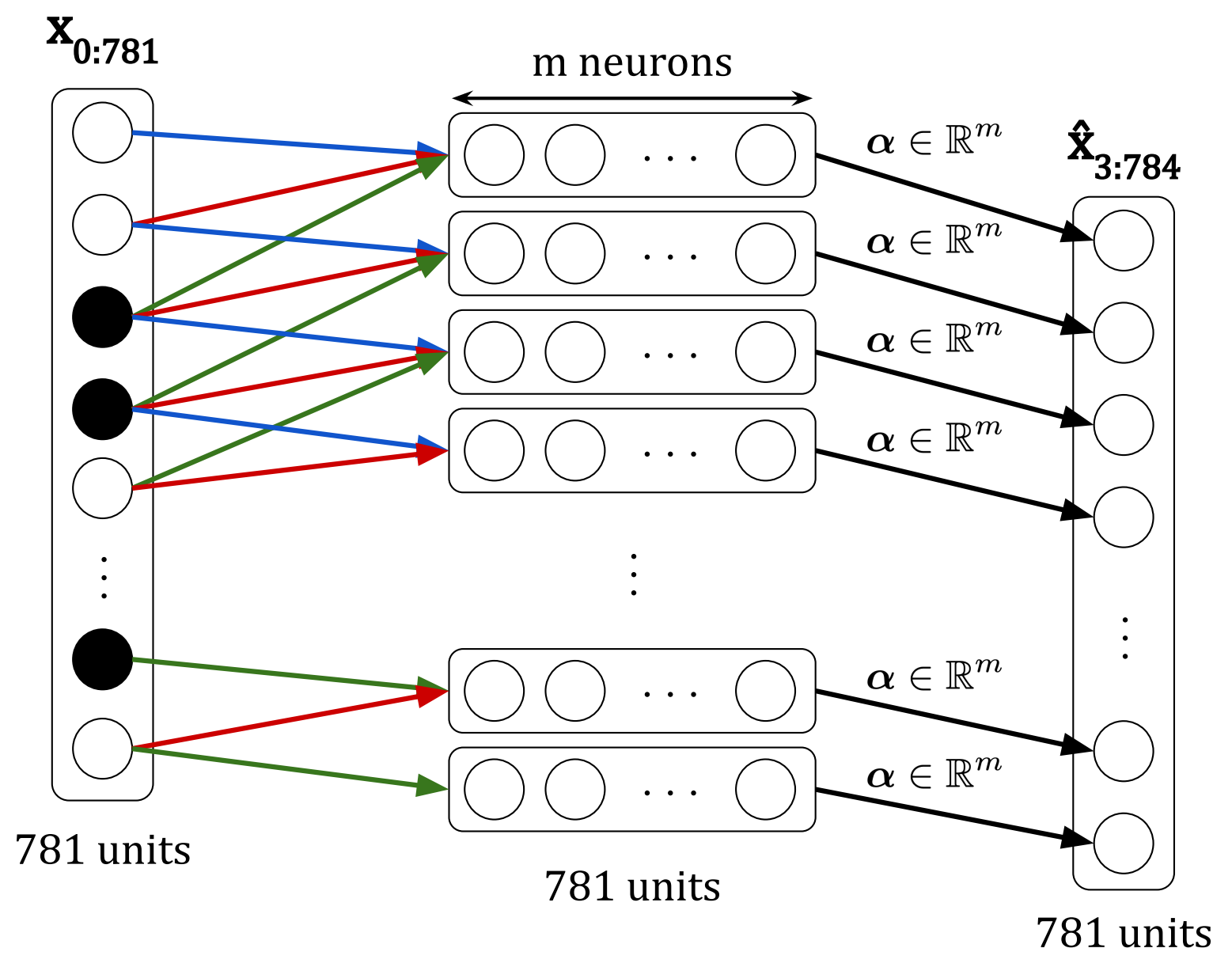
$\mathbf{x}_{0:781}$
m neurons
$\boldsymbol{\alpha} \in \mathbb{R}^m$
$\boldsymbol{\alpha} \in \mathbb{R}^m$
$\boldsymbol{\alpha} \in \mathbb{R}^m$
$\boldsymbol{\alpha} \in \mathbb{R}^m$
$\boldsymbol{\alpha} \in \mathbb{R}^m$
$\boldsymbol{\alpha} \in \mathbb{R}^m$
$\hat{\mathbf{x}}_{3:784}$
781 units
781 units
781 units

Figure: Autoregressive model for an image, with $r = 3$.

Exact Convex Program and Theorem

If $m \geq m^*$, the non-convex problem has the same optimal value as the convex problem

$$\min_{\{\boldsymbol{v}_i,\boldsymbol{w}_i\}_{i=1}^P} \mathcal{L}\Big(\sum_{i=1}^P \boldsymbol{D}_i\boldsymbol{X}_{(r)}(\boldsymbol{v}_i - \boldsymbol{w}_i), \boldsymbol{y}\Big) + \beta\sum_{i=1}^P(||\boldsymbol{v}_i||_2 + ||\boldsymbol{w}_i||_2)$$

subject to $\boldsymbol{G}_i\boldsymbol{v}_i \geq 0, \boldsymbol{G}_i\boldsymbol{w}_i \geq 0$ for $i = 1, \ldots, P$,

where $\boldsymbol{G}_i = (2\boldsymbol{D}_i - \boldsymbol{I}_n)\boldsymbol{X}_{(r)}$. Furthermore, given optimal solutions $\boldsymbol{v}_i^*, \boldsymbol{w}_i^*$ for this problem such that at most one of $\boldsymbol{v}_i^*$ or $\boldsymbol{w}_i^*$ is non-zero for all $i = 1, \ldots, P$, an optimal solution for the non-convex problem with m^* neurons is

$$(\boldsymbol{u}_{j_i}^*, \alpha_{j_i}^*) = \begin{cases} (\frac{\boldsymbol{v}_i^*}{\sqrt{||\boldsymbol{v}_i^*||_2}}, \sqrt{||\boldsymbol{v}_i^*||_2}) & \text{if } ||\boldsymbol{v}_i^*||_2 > 0 \\ (\frac{\boldsymbol{w}_i^*}{\sqrt{||\boldsymbol{w}_i^*||_2}}, -\sqrt{||\boldsymbol{w}_i^*||_2}) & \text{if } ||\boldsymbol{w}_i^*||_2 > 0 \end{cases}.$$

Interpretation of Exact Convex Program

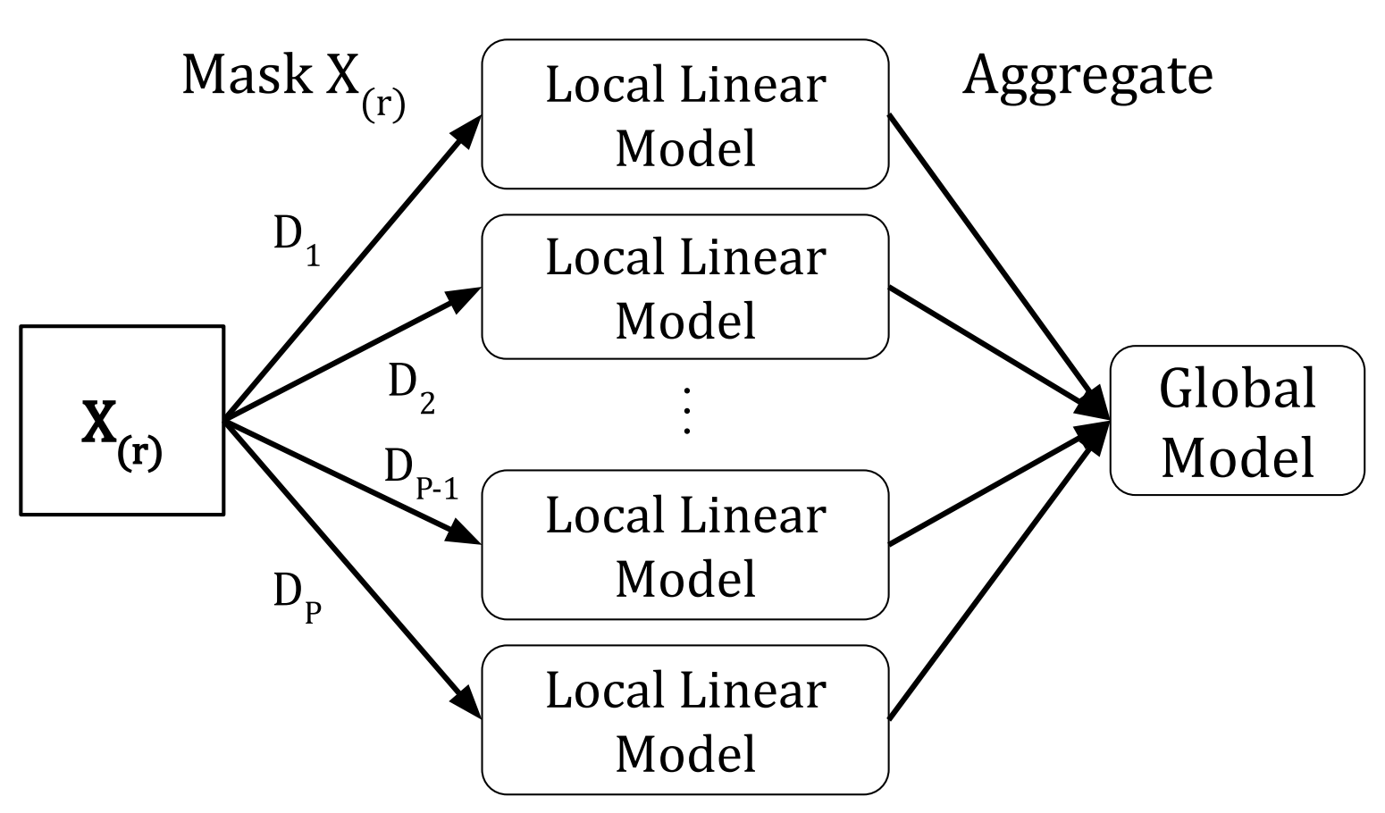

Figure: P local (constrained) models are aggregated to create the global model via summation and regularization.

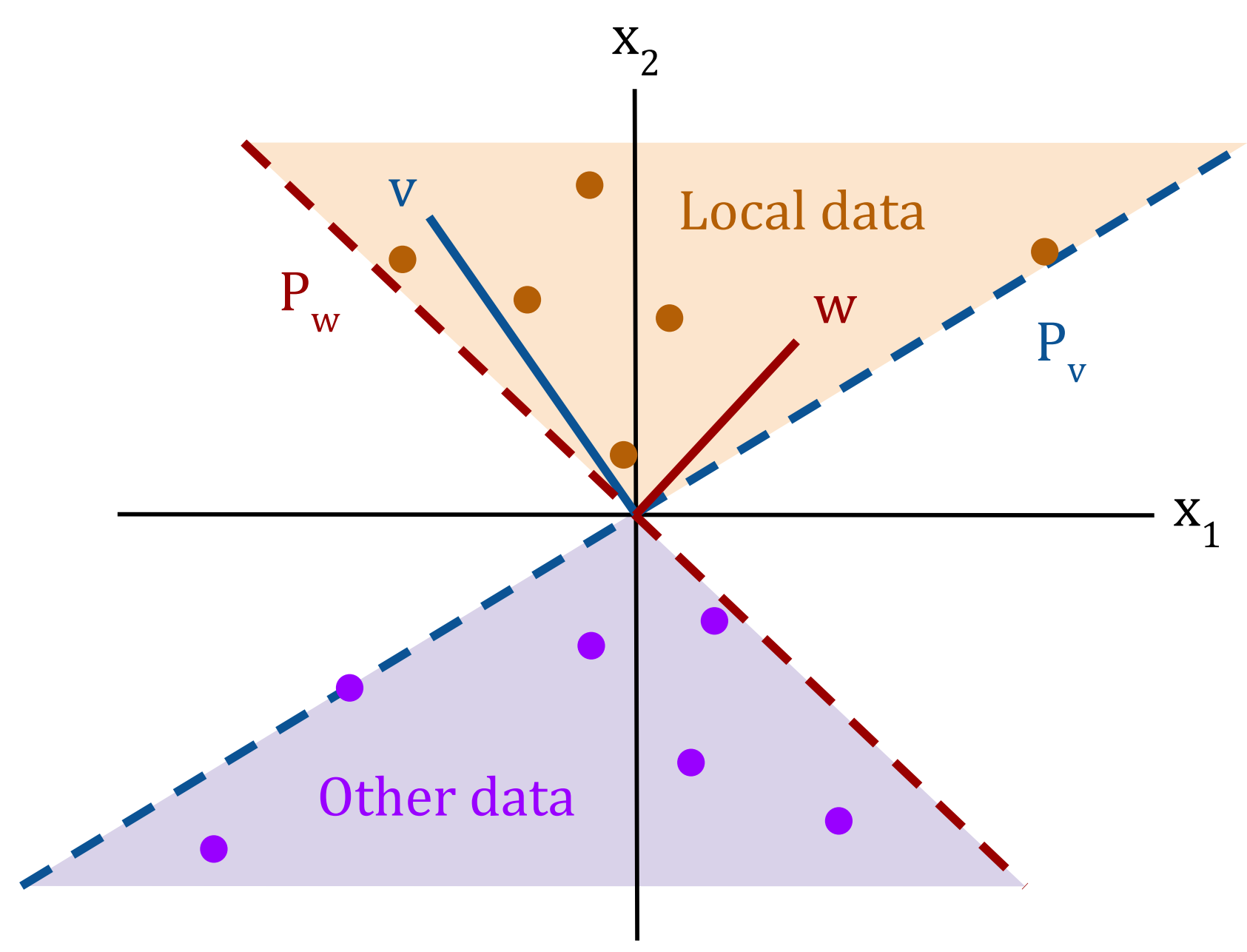

Figure: Local geometry of the constraints with optimization variables v and w.

Each term $\boldsymbol{D}_i\boldsymbol{X}_{(r)}$ is an arbitrary mask of $\boldsymbol{X}_{(r)}$, giving us the following:

- **First term:** aggregate BCE loss over P local linear models, where the local feature matrix is a randomly masked version of the global feature matrix.
- **Second term:** group lasso regularization to merge local models into global.
- **Constraints:** dot products of local variables with local data points are positive, while those with the other data points are negative. Geometrically, we construct two planes, P_v and P_w, perpendicular to the two local optimization variables. The local models use all of the data (and no other data) in the intersection of the two half-spaces containing the local variables to compute their contribution to the global loss.

Tractable Formulations

Relaxed Convex Program

$$\min_{\{\boldsymbol{v}_i,\boldsymbol{w}_i\}_{i=1}^P} \mathcal{L}\Big(\sum_{i=1}^P \boldsymbol{D}_i\boldsymbol{X}_{(r)}(\boldsymbol{v}_i - \boldsymbol{w}_i), \boldsymbol{y}\Big) + \beta\sum_{i=1}^P(||\boldsymbol{v}_i||_2 + ||\boldsymbol{w}_i||_2)$$

Hinge Loss Approximation

$$\min_{\{\boldsymbol{v}_i,\boldsymbol{w}_i\}_{i=1}^P} \mathcal{L}\Big(\sum_{i=1}^P \boldsymbol{D}_i\boldsymbol{X}_{(r)}(\boldsymbol{v}_i - \boldsymbol{w}_i), \boldsymbol{y}\Big) + \beta\sum_{i=1}^P(||\boldsymbol{v}_i||_2 + ||\boldsymbol{w}_i||_2)$$
$$+ \rho\sum_{i=1}^P \mathbf{1}^T\left((\boldsymbol{G}_i\boldsymbol{v}_i)_+ + (\boldsymbol{G}_i\boldsymbol{w}_i)_+\right)$$

Sampling Diagonal Matrices

The number of diagonal matrices P needed explodes quickly (Pilanci 2020): with $N = 1000$ sequences of dimension $D = 100$ and $r = 10$ (a small dataset), $P \leq O(10^{50})$. We sample $\tilde{P}$ diagonal matrices by generating $\tilde{P}$ uniformly random vectors $\boldsymbol{u}_i$ and picking $\boldsymbol{D}_i = 1[\boldsymbol{X}_{(r)}\boldsymbol{u}_i \geq 0]$ for $i = 1, \ldots, \tilde{P}$.

Numerical Experiments

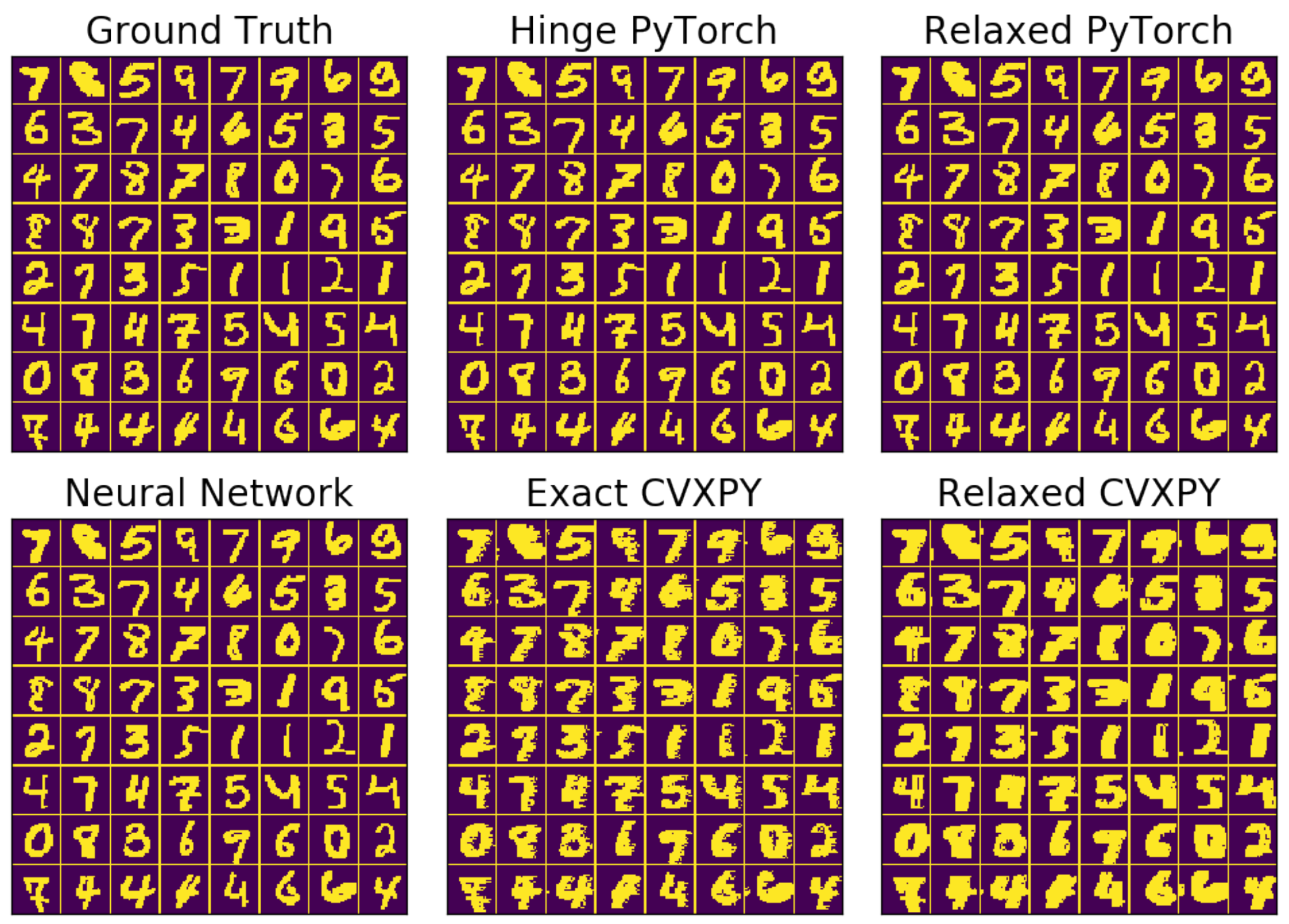

Figure: Ground truth and generated images from MNIST test data. CVXPY implementations trained on 10 vectors (less than an image). Images generated by predicting each pixel given previous 10 pixels in the ground truth image (not recursively).

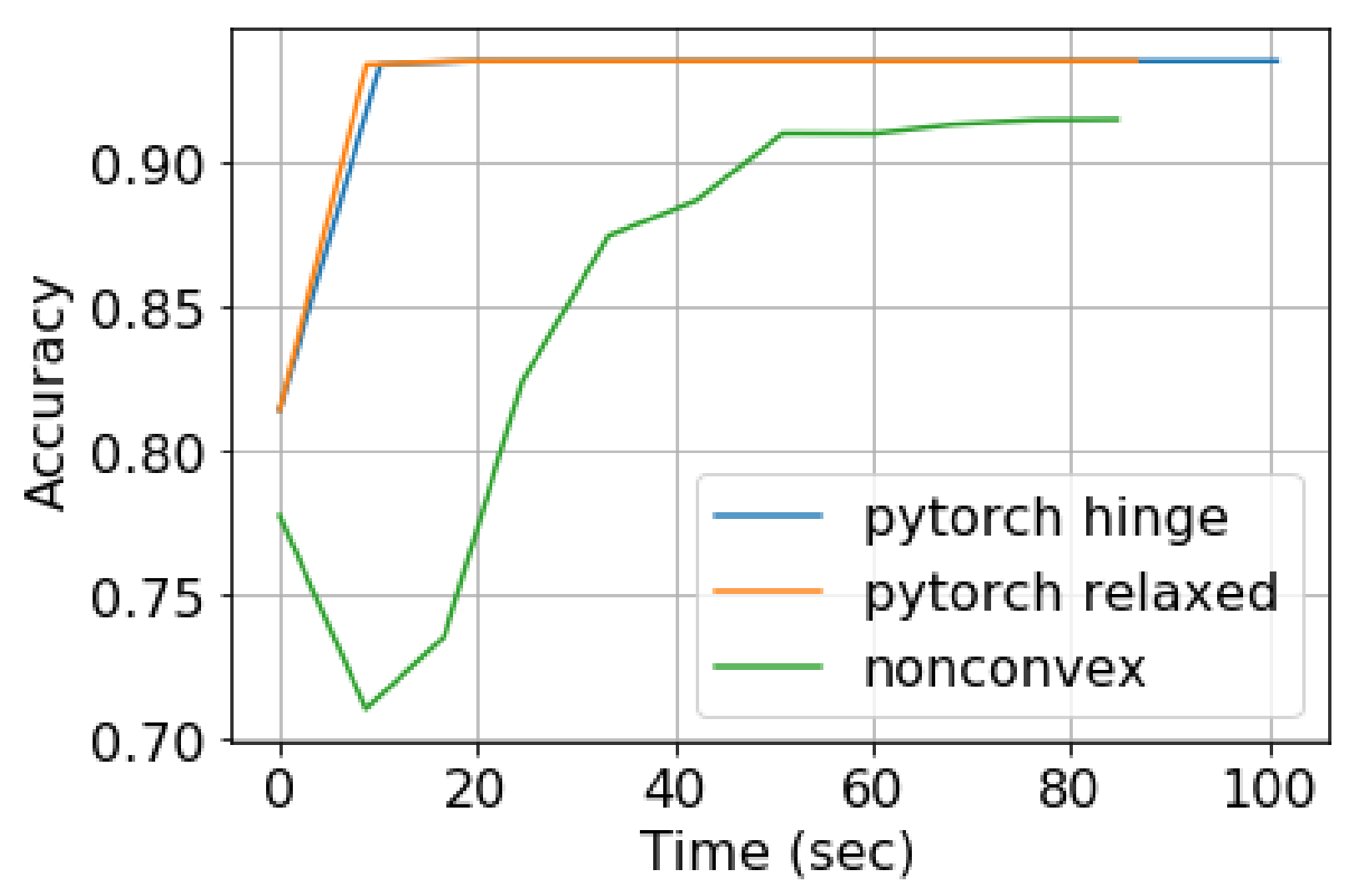

Figure: Accuracy on MNIST val data during training.

Concluding Remarks

- Two-layer autoregressive models are equivalent to constrained, regularized logistic regression.
- Relaxing the constraints allows us to use batched gradient descent. This combined with sampling the diagonal matrices results in faster training and better performance.
- Possible extensions: deeper autoregressive networks, including PixelRNN, WaveNet, Image-GPT.

References: M. Pilanci and T. Ergen, "Neural Networks are Convex Regularizers: Exact Polynomial-Time Convex Optimization Formulations for Two-Layer Networks," ICML 2020.