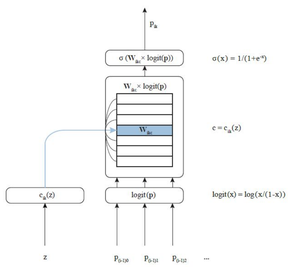
DeepMind

Gated Linear Networks

Joel Veness*, Tor Lattimore*, David Budden*, Avishkar Bhoopchand*, Christopher Mattern, Agnieszka Grabska-Barwinska, Eren Sezener, Jianan Wang, Peter Toth, Simon Schmitt, Marcus Hutter

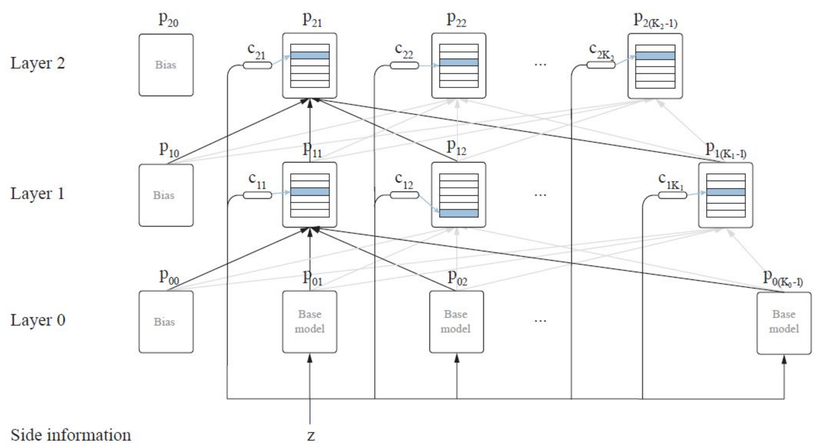
Introduction

- Gated Linear Networks (GLNs) are a general purpose family of neural networks, with an interesting and distinct take on credit assignment.
- Many possible practical uses, such as regression [1], contextual bandits [3], multiclass classification [4], transfer learning and non-stationary time series modelling [2]., density estimation [5] and data compression.



- Associated to each neuron is a fixed, deterministic context function which given an input selects a weight vector to use
- Log loss w.r.t. local weights is convex => can use OGD for optimisation.
- Input/output non-linearities cancel.

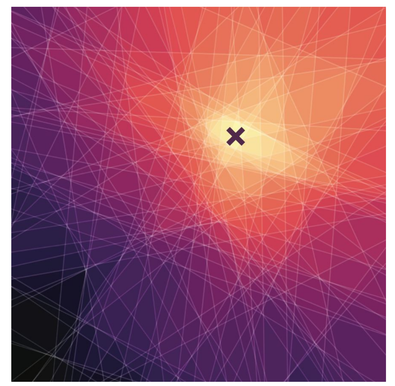
GLN Architecture



Key Properties

- Fast, online learners
- Robust to forgetting
- Universal
- Parameters adapted via convex optimisation
- Can think of each neuron acting as a kind of distributional "boosting" mechanism
- The loss functions are **convex** for exponential family PDFs.
- Targets and input are broadcast to each neuron, each neuron has own loss function.

- Interpretable
- Distributional
- Compositional



Gating + Local Learning => Universality

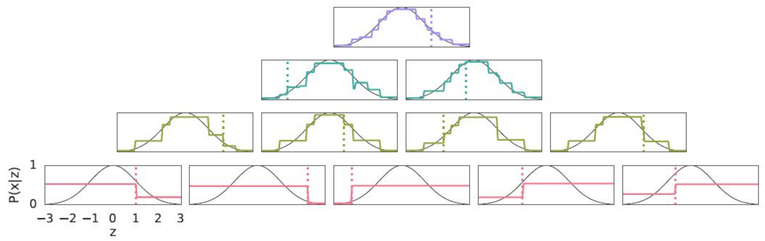
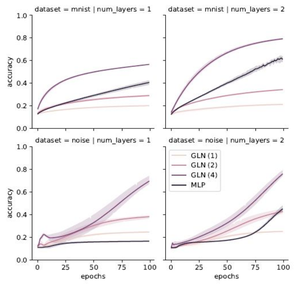


Figure 2. Output of a four layer network with random half-space contexts after training to convergence. Each box represents a non-bias neuron in the network, the function to fit is shown in black, and the output distribution learnt by each neuron is shown in colour (for example, red for the first layer and purple for the top-most neuron). All axes are identical, as labeled in the bottom left neuron. The dashed coloured lines represent the sampled hyperplane for each neuron.

- Can get a sense of model capacity by comparing ability to fit randomly shuffled or noisy labels.
- GLNs compare favourably with Deep ReLu networks.
- (Open Question) Can we characterize this formally?



Linear Interpretability

$$\sigma\Big(\underbrace{W_L(z)\,W_{L-1}(z)\,\ldots\,W_1(z)}_{\text{multilinear polynomial of degree L}}\,\text{logit}(p_0)\Big),$$

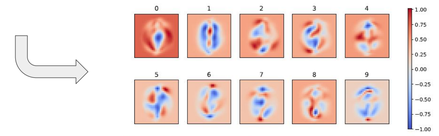


Figure 5. Saliency maps for constituent GLN binary classifiers of one-vs-all MNIST classifier after a single training epoch.

Resilience to Catastrophic Forgetting

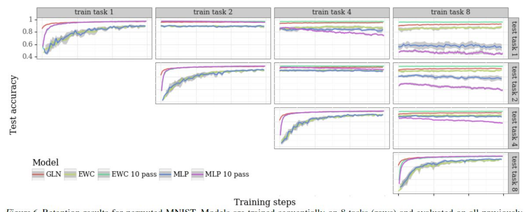


Figure 6. Retention results for permuted MNIST. Models are trained sequentially on 8 tasks (rows) and evaluated on all previously encountered tasks (columns). For example, the top-right plot indicates performance on Task 1 after being trained sequentially on Tasks 1 to 8 inclusive (not all tasks shown). Each model only trains for one epoch per task, with the exception of "EWC 10 pass" and "MLP 10 pass" (shrunken 10-fold on x axis). Error bars denote 95% confidence levels over 10 random seeds.

Inductive Bias and Results

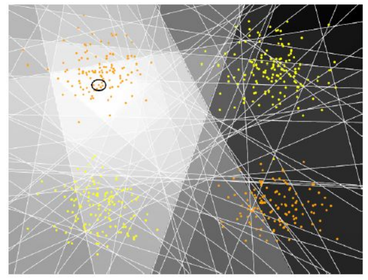


Figure 4. The effect of a single noisy XOR update (circled) on the decision boundaries of a halfspace gated GLN. Sampled hyperplanes for each gate are shown in white.

Inputs close in terms of cosine similarity will map to similar products of weight matrices!

- Single pass classification performance of GLNs matches general purpose batch techniques like SVMs, XGBoost, Deep Relu Networks on UCI datasets.
- 98% accuracy with a single online pass over MNIST
- Matches SOTA NATs-per-image for autoregressive MNIST density modelling using just 1 pass!

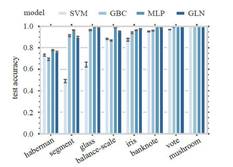
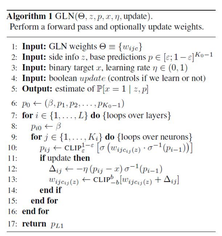


Figure 7. Online (single-pass) GLN classification accuracy on a selection UCI datasets, compared to three contemporary batch methods (Support Vector Machine, Gradient Boosting for Classification, Multi-Layer Perceptron) trained for 100 epochs.

Algorithm and Code

- Of course, the best way to build understanding and intuition is to see them in action:

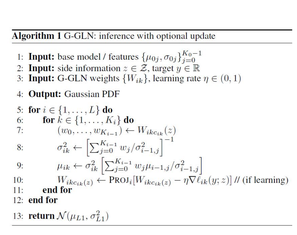
  github.com/deepmind/deepmind-research/tree/master/gated_linear_networks



Algorithm 1 GLN(Θ, z, p, x, η, update).
Perform a forward pass and optionally update weights.

Input: GLN weights $\Theta \equiv \{w_{ijk}\}$
Input: side info z, base predictions $p \in [\varepsilon, 1-\varepsilon]^{K_0-1}$
Input: binary target x, learning rate $\eta \in (0, 1)$
Input: boolean *update* (controls if we learn or not)
Output: estimate of $\mathbb{P}[x = 1 \mid z, p]$

$p_0 \leftarrow (\beta, p_1, p_2, \ldots, p_{K_0-1})$
for $i \in \{1, \ldots, L\}$ **do** {loops over layers}
  $p_{i0} \leftarrow \beta$
  for $j \in \{1, \ldots, K_i\}$ **do** {loops over neurons}
    $p_{ij} \leftarrow \text{CLIP}^{1-\varepsilon}_{\varepsilon}\left[\sigma\left(w_{ijc_{ij}(z)} \cdot \sigma^{-1}(p_{i-1})\right)\right]$
    if update **then**
      $\Delta_{ij} \leftarrow -\eta\,(p_{ij} - x)\,\sigma^{-1}(p_{i-1})$
      $w_{ijc_{ij}(z)} \leftarrow \text{CLIP}^{b}_{-b}[w_{ijc_{ij}(z)} + \Delta_{ij}]$
    end if
  end for
end for
return p_{L1}



Algorithm 1 G-GLN: inference with optional update

Input: base model / features $\{\mu_{0j}, \sigma_{0j}\}_{j=0}^{K_0-1}$
Input: side information $z \in \mathcal{Z}$, target $y \in \mathbb{R}$
Input: G-GLN weights $\{W_{ik}\}$, learning rate $\eta \in (0, 1)$
Output: Gaussian PDF
for $i \in \{1, \ldots, L\}$ **do**
  for $k \in \{1, \ldots, K_i\}$ **do**
    $(w_0, \ldots, w_{K_{i-1}}) \leftarrow W_{ikc_{ik}(z)}$
    $\sigma^2_{ik} \leftarrow \left[\sum_{j=0}^{K_{i-1}} w_j / \sigma^2_{i-1,j}\right]^{-1}$
    $\mu_{ik} \leftarrow \sigma^2_{ik}\left[\sum_{j=0}^{K_{i-1}} w_j \mu_{i-1,j} / \sigma^2_{i-1,j}\right]$
    $W_{ikc_{ik}(z)} \leftarrow \text{PROJ}_t[W_{ikc_{ik}(z)} - \eta \nabla \ell_{ik}(y; z)]$ // (if learning)
  end for
end for
return $\mathcal{N}(\mu_{L1}, \sigma^2_{L1})$

Additional GLN Resouces

- [1] Budden, David, et al. "Gaussian Gated Linear Networks", NeurIPS, 2020.
- [2] Wang, Jianan, et al. "A Combinatorial Perspective on Transfer Learning.", NeurIPS, 2020.
- [3] Sezener, Eren, et al. "Online Learning in Contextual Bandits using Gated Linear Networks", NeurIPS, 2020.
- [4] Sezener, Eren, et al. "Gated Linear Networks and Extensions, NeurIPS Beyond Backpropagation Workshop, 2020.
- [5] Veness, et al. "Online learning with gated linear networks" arXiv preprint arXiv:1712.01897, 2017.

* *Equal contributions. Contact author: aixi@google.com*