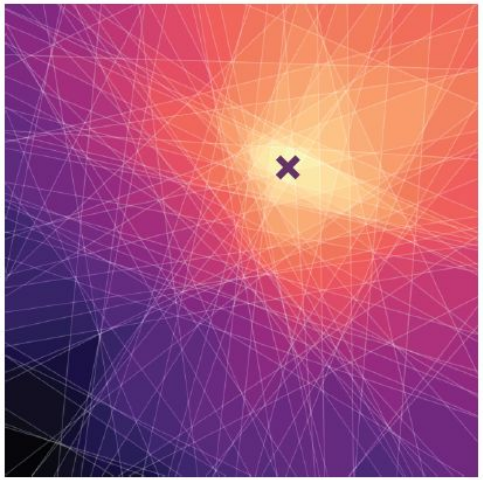
Gated Linear Networks



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

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 [3], contextual bandits [2], transfer learning and non-stationary time series modelling [4].

- Here we will focus on how and why they learn, what are the advantages, what are the current limitations, and discuss some open questions.

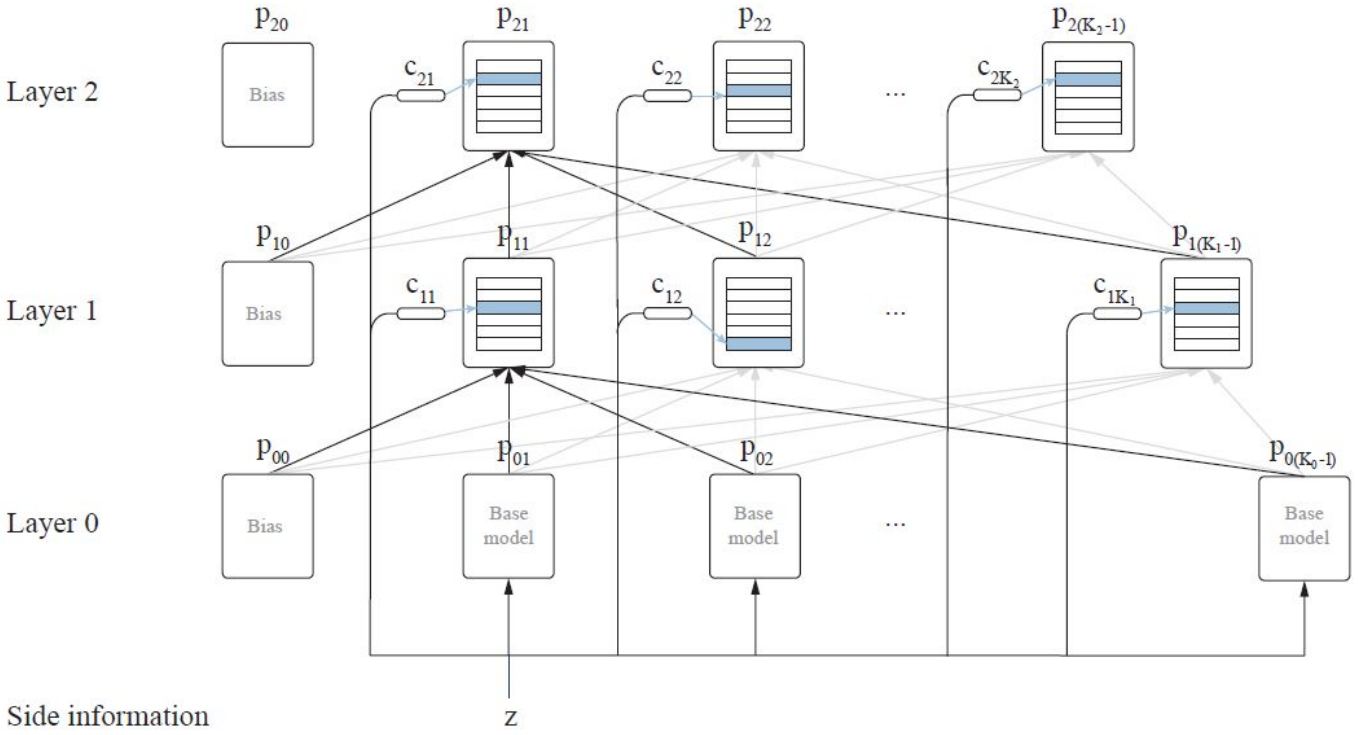
So what is a GLN?

- A scalable and universal (conditional target) density approximation technique. Origins in data compression [1] and can be thought as a generalisation of PAQ mixing networks.

- A feedforward neural network trained via the simultaneous optimization of many convex losses, one per neuron. Interestingly, each neuron attempts to predict the target directly.

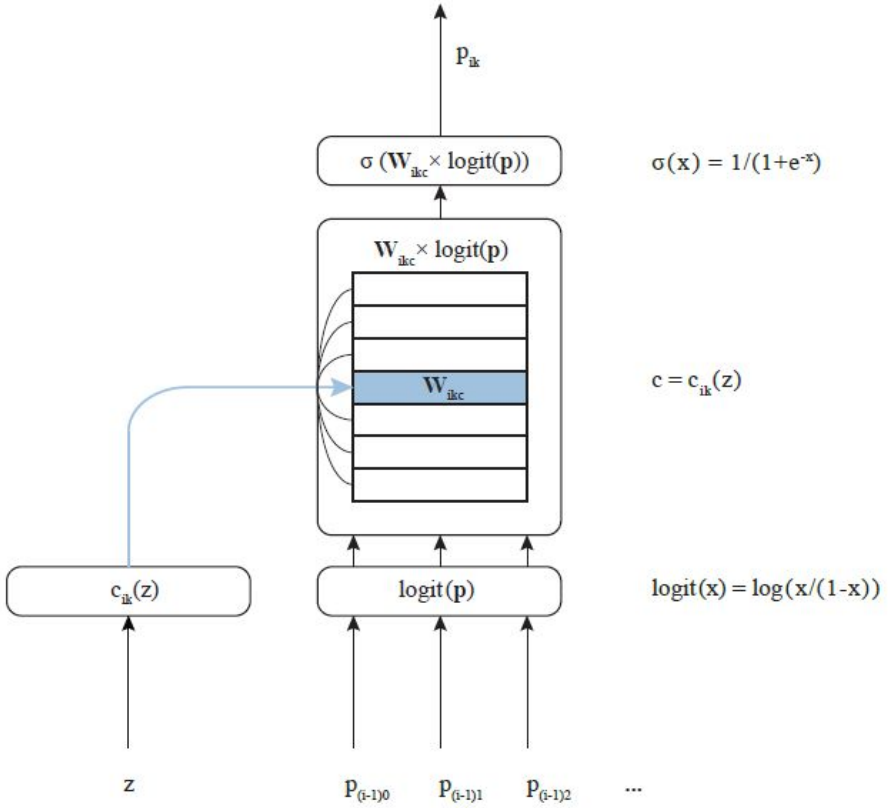
- Relies on the interaction between gating and local learning to gain (non-linear) representation power. *No backpropagation and no implicit feature construction!* GLNs are a smoothing technique with an inductive bias which is can be controlled by a choice of gating function.

GLN Architecture



GLN Neuron

- 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 Neuron with Halfspace Gating

- Given a choice of weights, a GLN computes a weighted product of experts:

$$\sigma\left(w^{\top}\sigma^{-1}(p)\right) = \frac{\prod_{i=1}^{d} p_i^{w_i}}{\prod_{i=1}^{d} p_i^{w_i} + \prod_{i=1}^{d}(1-p_i)^{w_i}}$$

- Context functions are generated randomly at initialisation, by uniformly sampling normal vectors from the surface of a unit sphere, and using them to define a pair of halfspaces, with a weight vector associated to each one. Given an example, the context function will determine which halfspace it lies in, and use corresponding weight vector.

Gating + Local Learning: is it all you need?

See [1] for theoretical analysis

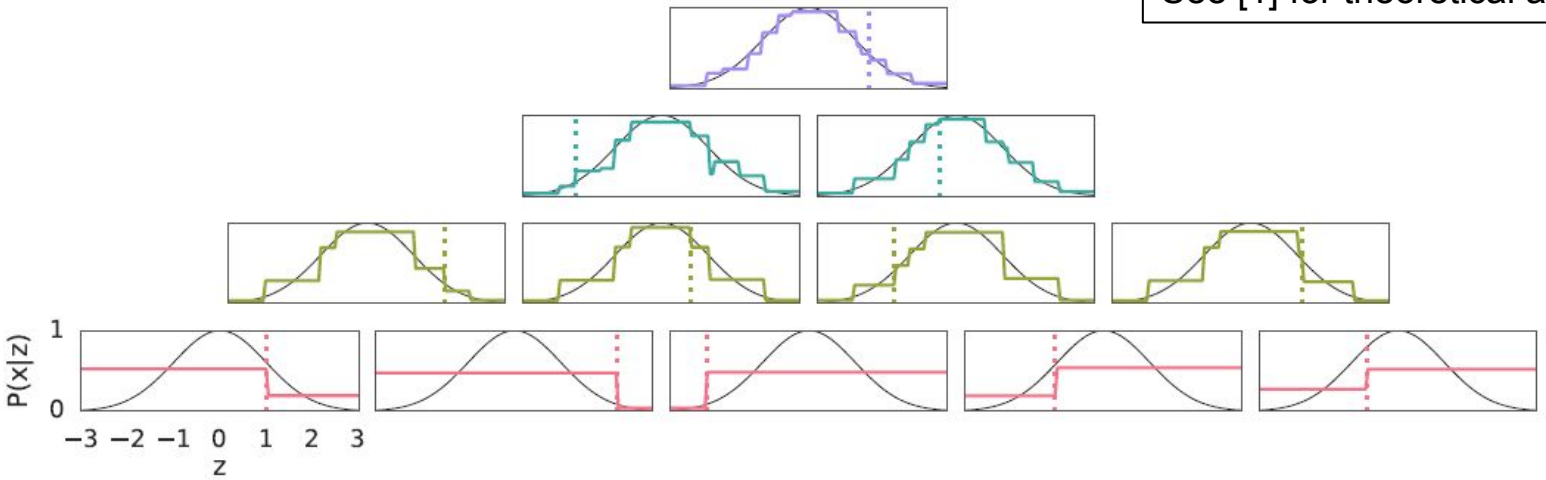
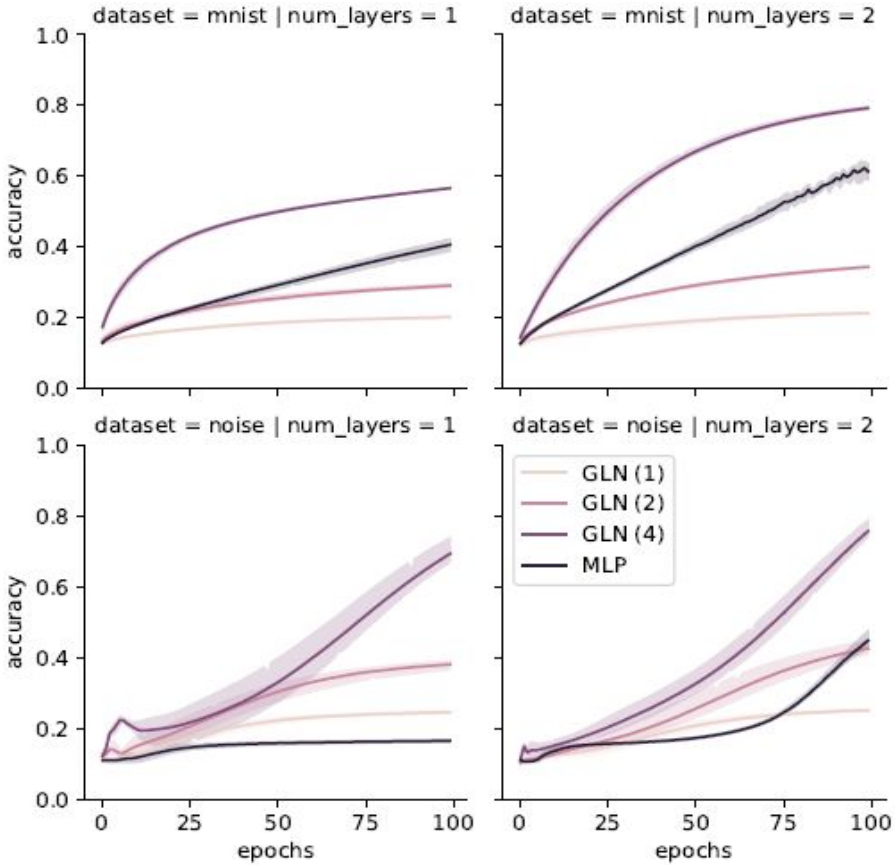


Figure 2. Output of a four layer network with random half-space contexts after training to convergence. Each box represents a non-bias neuron in the network, the function to fit is shown in black, and the output distribution learnt by each neuron is shown in colour (for example, red for the first layer 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.

Empirical Capacity

- 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?



Learning Dynamics

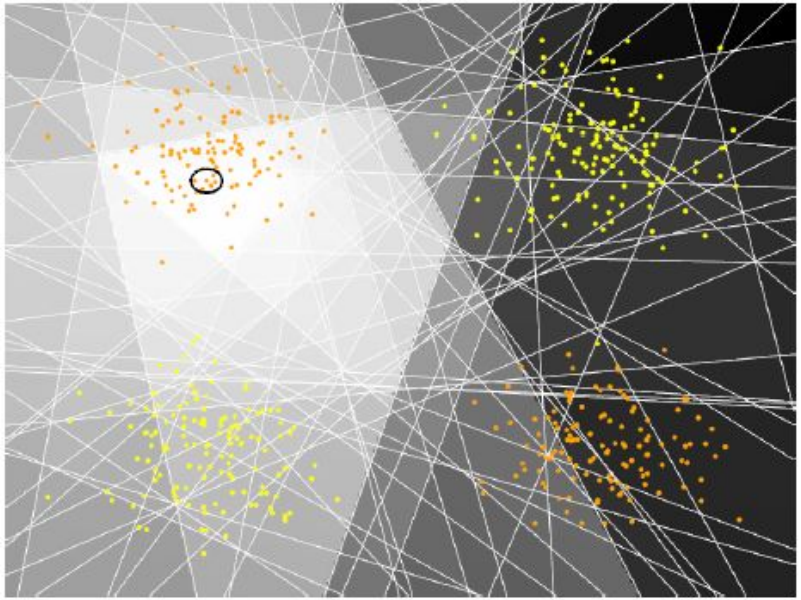


Figure 4. The effect of a single noisy XOR update (circled) on the decision boundaries of a halfspace gated GLN. Sampled hyperplanes for each gate are shown in white.

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

GLNs are data efficient neural networks suited for online learning

- 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!

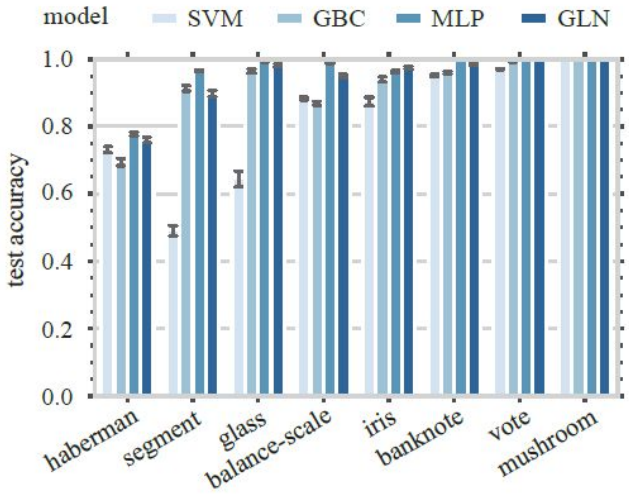


Figure 7. Online (single-pass) GLN classification accuracy on a selection UCI datasets, compared to three contemporary batch methods (Support Vector Machine, Gradient Boosting for Classification, Multi-Layer Perceptron) trained for 100 epochs.

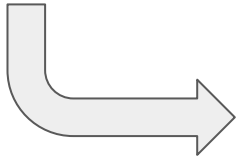
Additional results:

- SOTA performance in contextual bandits and regression:
  - Sezener, et al., *Online Learning in Contextual Bandits using Gated Linear Networks*, NeurIPS, 2020.
  - Budden, et al., *Gaussian Gated Linear Networks*, NeurIPS, 2020.

- Modular and decoupled learning opens up many avenues for building networks which can adapt to non-stationarity and transfer across tasks.
  - Wang, et al., *A Combinatorial Perspective on Transfer Learning*, NeurIPS, 2020.

Linear Interpretability

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



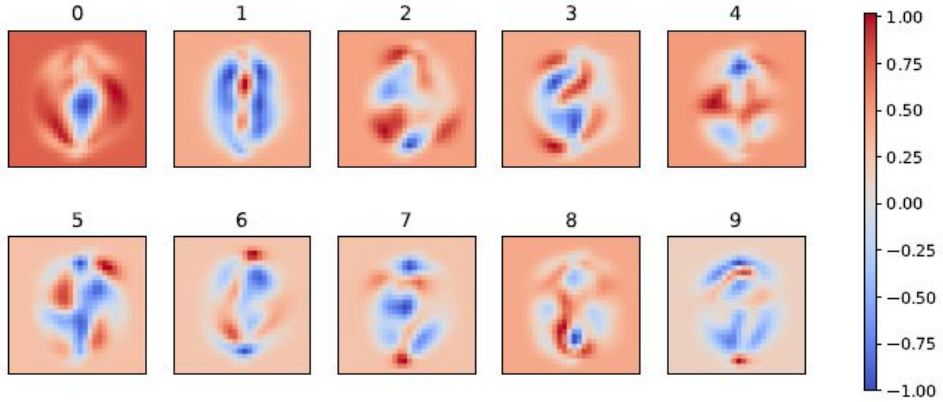


Figure 5. Saliency maps for constituent GLN binary classifiers of one-vs-all MNIST classifier after a single training epoch.

Resilience to Catastrophic Forgetting

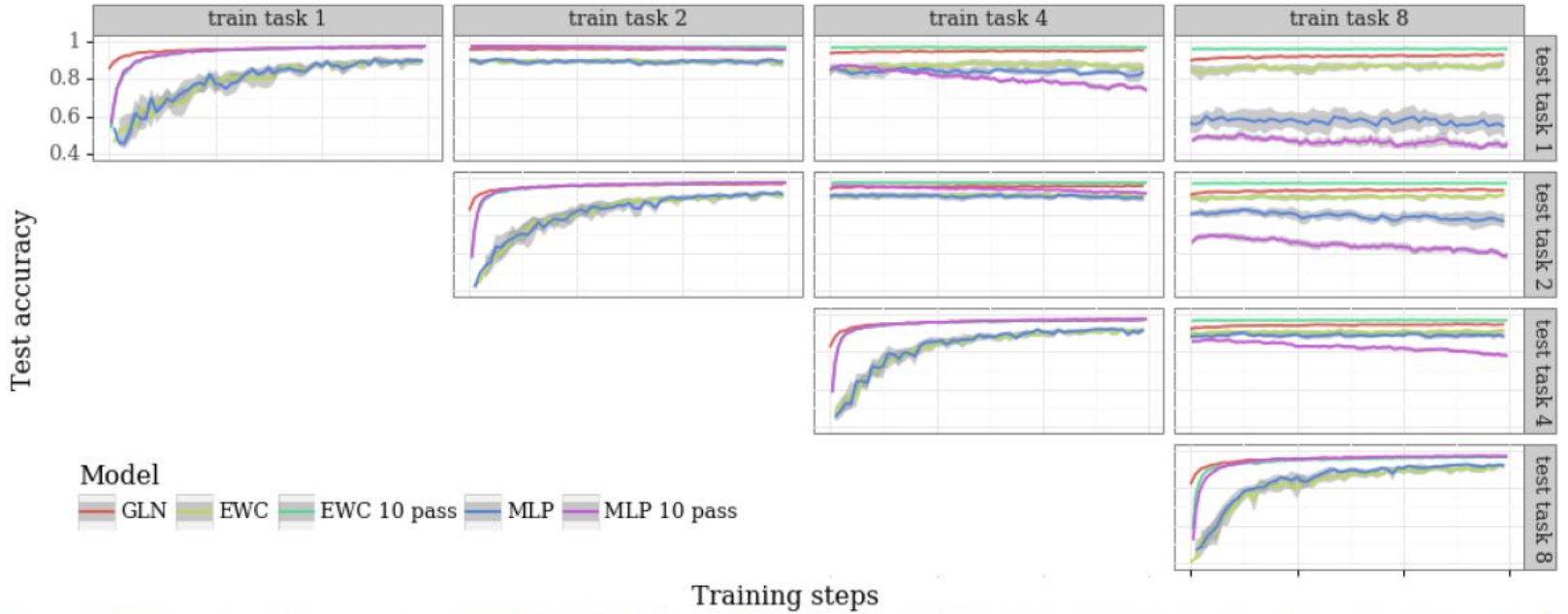


Figure 6. Retention results for permuted MNIST. Models are trained sequentially on 8 tasks (rows) and evaluated on all previously encountered tasks (columns). For example, the top-right plot indicates performance on Task 1 after being trained sequentially on Tasks 1 to 8 inclusive (not all tasks shown). 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.

GLN algorithms as propagation of sufficient statistics

Algorithm 1 GLN$(\Theta, z, p, x, \eta, \text{update})$.
Perform a forward pass and optionally update weights.

Input: GLN weights $\Theta \equiv \{w_{ijc}\}$
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}
    $p_{ij} \leftarrow \text{CLIP}_{\varepsilon}^{1-\varepsilon}\left[\sigma\left(w_{ijc_{ij}(z)} \cdot \sigma^{-1}(p_{i-1})\right)\right]$
    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**
    $(w_0, \ldots, w_{K_{i-1}}) \leftarrow W_{ikc_{ik}}(z)$
    $\sigma_{ik}^2 \leftarrow \left[\sum_{j=0}^{K_{i-1}} w_j/\sigma_{i-1,j}^2\right]^{-1}$
    $\mu_{ik} \leftarrow \sigma_{ik}^2 \left[\sum_{j=0}^{K_{i-1}} w_j \mu_{i-1,j}/\sigma_{i-1,j}^2\right]$
    $W_{ikc_{ik}(z)} \leftarrow \text{PROJ}_i[W_{ikc_{ik}(z)} - \eta \nabla \ell_{ik}(y; z)]$ // (if learning)
  end for
end for

return $\mathcal{N}(\mu_{L1}, \sigma_{L1}^2)$

Code Available!

- 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

Summary

- GLNs are a different take on neural networks, a combination of ideas from data compression, online learning, deep learning, which are well suited to online or data limited regimes.

- Many interesting questions remain: are there other general purpose classes of context functions with different inductive biases? How do we incorporate translation invariance or other image specific prior knowledge? Can we say something stronger than (asymptotic) universality in theory?

- Early days for this method, but initial results exciting and we are only just beginning to understand which problems they are best suited to...

References

[1] Veness et al., *Online learning with gated linear networks*, Technical Report, arXiv:1712.01897, 2017.

[2] Sezener, et al., *Online Learning in Contextual Bandits using Gated Linear Networks*, NeurIPS, 2020.

[3] Budden, et al., *Gaussian Gated Linear Networks*, NeurIPS, 2020.

[4] Wang, et al., *A Combinatorial Perspective on Transfer Learning*, NeurIPS, 2020.