
Deep Equilibrium Models as Estimators for Continuous Latent Variables

Russell Tsuchida
Data61, CSIRO, Australia

Cheng Soon Ong
Data61, CSIRO
and Australian National University

Abstract

Principal Component Analysis (PCA) and its exponential family extensions have three components: observations, latents and parameters of a linear transformation. We consider a generalised setting where the canonical parameters of the exponential family are a nonlinear transformation of the latents. We show explicit relationships between particular neural network architectures and the corresponding statistical models. We find that deep equilibrium models — a recently introduced class of implicit neural networks — solve maximum a-posteriori (MAP) estimates for the latents and parameters of the transformation. Our analysis provides a systematic way to relate activation functions, dropout, and layer structure, to statistical assumptions about the observations, thus providing foundational principles for unsupervised DEQs. For hierarchical latents, individual neurons can be interpreted as nodes in a deep graphical model. Our DEQ feature maps are end-to-end differentiable, enabling fine-tuning for downstream tasks.

1 INTRODUCTION

Deep learning provides a means of fitting highly flexible but theoretically opaque functions to data. Functions are defined as compositions of parameterised mappings called layers. Parameters of layers are typically adjusted by applying first-order optimisation methods to an objective involving data and the parameters. Layers may provide an explicit description of a mapping. For example, given an input $x \in \mathbb{R}^D$, the L -dimensional output of a layer with parameters $\theta \in \mathbb{R}^{L \times D}$ might be defined by $\sigma(\theta x)$, where σ is a nonlinear activation function.

Alternatively, layers may be defined implicitly as solutions to certain problems. For example, given an input $x \in \mathbb{R}^D$, a Deep Equilibrium Model (DEQ) (Bai et al., 2019) layer might output a solution to the fixed point equation $z = \sigma(\Gamma z + \theta x)$, where $\theta \in \mathbb{R}^{L \times D}$ and $\Gamma \in \mathbb{R}^{L \times L}$ are parameters and $z \in \mathbb{R}^L$ is the implicitly defined output. Implicit layers include DEQs which solve fixed point equations, Neural ODEs (Chen et al., 2018) which solve ODEs and Deep Declarative Networks (DDNs) (Gould et al., 2021) which solve optimisation problems. All implicit layers and in particular DEQs carry an issue of well-posedness, that is, whether there exists a unique solution to the problem that defines the layer (Winston and Kolter, 2020). Implicit networks’ parameters retain the ability of being easy to adjust using first-order optimisation methods. Under mild conditions, the gradient of the layer can be computed without back-propagating through the solver that computes the output of the implicit layer, via the implicit function theorem.

The use of DEQs, and deep learning models more generally, is usually motivated from a top-down view. One uses a deep learning model for some combination of predictive performance (measured empirically *a posteriori*), or representational capacity (proven mathematically *a priori*). In contrast, classical statistical models can often be motivated from a bottom-up data generating process and are therefore interpretable (Rudin, 2019), but might not necessarily achieve as good performance for a given task. Our paper closes this gap, by understanding DEQ models through data generating processes (Efron, 2020).

Contributions We present Principal Equilibrium Dimensions (PED), a DEQ that solves the problem of joint MAP estimation in a graphical model representing nonlinearly parameterised exponential family PCA. We stress that our contributions are mostly theoretical in nature, with the aim of closing the gap between black box predictors and data generating models. Our analysis provides a bottom-up justification for the use of particular DEQ layers in unsupervised learning. PED layers have similar components to explicit networks, such as activation functions and dropout. PED layers are end-to-end differentiable and may be fine-tuned in supervised settings. Our contributions are as follows.

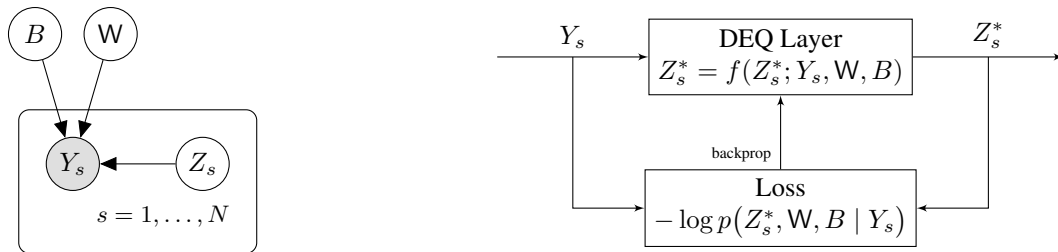


Figure 1: (Left) The graphical model for nonlinearly parameterised exponential family PCA with observations Y_s , latents Z_s , and parameters W, B . The likelihood of the observation Y_s is an exponential family with a canonical parameter $R(WZ_s + B)$, for some nonlinearity R . (Right) MAP estimates can be found using PED. Latents are estimated by the prediction of the DEQ layer given fixed parameters. Parameters of the DEQ layer are adjusted through backpropagation.

- We show that MAP estimates of the latents and parameters of nonlinearly parameterised exponential family PCA can be computed by a DEQ layer. See Figure 1.
- We relate statistical assumptions about the data to a choice of activation function, giving conceptual meaning to the use of activations such as tanh, logistic sigmoid, softmax, and ReLU, as well as a form of learned dropout. See Table 1.
- In the nicest setting (Theorem 1), PED layers compute solutions to strongly convex optimisation problems. Hence they are guaranteed to admit unique fixed points. In less nice settings (Theorem 2), we ensure that PED layers compute local minima.
- We derive a principled method for constructing a DEQ that solves a graphical model with hierarchical latent variables, called deep PED. See Figure 3.
- We provide an implementation for shallow and deep PED and compare it to PCA, tSNE and UMAP on illustrative synthetic datasets (Figure 7)¹.

2 BACKGROUND AND NOTATION

Exponential Families An exponential family is a class of probability distributions supported on $\mathbb{Y} \subseteq \mathbb{R}$ whose sufficient statistics admit a moment generating function defined on an open set $\mathbb{F} \subseteq \mathbb{R}$. See Appendix B.1 for more details. Such an exponential family possesses a base density (mass) $h : \mathbb{Y} \rightarrow \mathbb{R}_{\geq 0}$, sufficient statistic $T : \mathbb{Y} \rightarrow \mathbb{R}$ and *log partition function* $A : \mathbb{F} \rightarrow \mathbb{R}$. Members of the exponential family are characterised by their canonical parameter $v \in \mathbb{F}$. The probability density (mass) function is

$$p(y | v) = h(y) \exp(vT(y) - A(v)). \quad (1)$$

For such a family, A is infinitely differentiable and strictly convex (Wainwright and Jordan, 2008, Proposition 3.1).

A acts as a cumulant generating function; in particular the expected value under $p(y | v)$ is $\mu = A'(v)$. We write

¹PyTorch implementation and reproducible experiments are available at <https://github.com/RussellTsuchida/ped>.

$y \sim \mathcal{D}_{A,\chi}(v)$ to mean that a scalar-valued random variable y follows an exponential family with log partition function A , scalar canonical parameter v and additional parameter χ^2 . If $Y = (y_1, \dots, y_d)$ and $V = (v_1, \dots, v_d)$, we write $Y \sim \mathcal{D}_{A,\chi}(V)$ to mean $y_i \sim \mathcal{D}_{A,\chi}(v_i)$ for all $1 \leq i \leq d$, with each y_i mutually conditionally independent given V .

Matrices and Data We use I to denote a square identity matrix, with dimensions defined from context. Let $Y_s = (y_{s1}; \dots; y_{sd}) \sim \mathcal{D}_{A,\chi}(V_s)$, where for each s , $V_s \in \mathbb{R}^d$ is a vector consisting of canonical parameters. Let $Y \in \mathbb{R}^{d \times N}$ denote a matrix with N such vectors Y_s , $1 \leq s \leq N$, each sampled conditionally independently given the canonical parameters. Each of the Y_s are associated with a latent representation $Z_s \in \mathbb{R}^l$. Write $Z \in \mathbb{R}^{l \times N}$ for the matrix consisting of such latent representations. We sometimes drop subscripts for cleanliness when they are fixed. Functions are composed using \circ , and \odot represents elementwise product of vectors.

Implicit Neural Networks Our analysis will reveal that MAP estimates of nonlinearly parameterised exponential family PCA can be computed by training a DEQ in an unsupervised setting. Connections between DEQs and DDNs (which solve optimisation problems) have previously been explored by Tsuchida et al. (2022); Li et al. (2022). We work with a prototype problem to represent a DEQ,

$$\begin{aligned} \min_{\theta, \Gamma} \quad & \sum_{s=1}^N \mathcal{L}(\mathcal{F}_\Gamma(Z_s^*, Y_s), Y_s) \\ \text{subject to} \quad & Z_s^* = g(\theta, Y_s, Z_s^*), \quad s = 1, \dots, N. \end{aligned} \quad (2)$$

Here θ are parameters of the implicit layer, Z_s^* are outputs, \mathcal{F}_Γ is a neural network (that may also contain implicit layers) parameterised by Γ , and \mathcal{L} is a loss function. The constraint is satisfied at a fixed point of a function g , which depends on parameters θ and input data Y_s .

²Some exponential families involve a parameter that is not canonical. E.g. the univariate Gaussian with known variance has a scale parameter (Nielsen and Garcia, 2009, p. 16), and Laplace has a centering parameter. Both A and χ are fixed.

| $R(\eta)$ | $A \circ R(\eta)$ | Soft dropout $\rho(\eta) := R'(\eta)$ | Activation $\sigma(\eta) := (A \circ R)'(\eta)$ |
|---|--|--|---|
| Linear R, Quadratic A (Classical PCA) | | | |
| η | $\eta^2/2$ | 1 | η |
| Linear R, General A (Exponential family PCA) | | | |
| η | $\log(1 + \exp(\eta))$ | 1 | $(1 + e^{-\eta})^{-1}$ |
| η | $\log \cosh \eta$ | 1 | $\tanh(\eta)$ |
| η | $\log \frac{e^\eta - 1}{\eta}$ | 1 | $\frac{e^\eta(\eta - 1) + 1}{(e^\eta - 1)\eta}$ |
| Nonlinear R, Quadratic A (Non-linearly parameterised classical PCA) | | | |
| $\text{ReLU}_\tau(\eta)$ | $\text{ReLU}_\tau^2(\eta)/2$ | $\Phi_\tau(\eta)$ | $\Phi_\tau(\eta) \text{ReLU}_\tau(\eta)$ |
| $\text{ReLU}(\eta)$ | $\text{ReLU}^2(\eta)/2$ | $\Theta(\eta)$ | $\text{ReLU}(\eta)$ |
| $\eta + \log \cosh \eta$ | $(\eta + \log \cosh \eta)^2/2$ | $1 + \tanh(\eta)$ | $(\tanh(\eta) + 1)(\eta + \log \cosh \eta)$ |
| Nonlinear R, General A (Non-linearly parameterised exponential family PCA) | | | |
| $-\text{ReLU}_\tau(\eta)$ | $-\log \text{ReLU}_\tau(\eta)$ | $-\Phi_\tau(\eta)$ | $-\frac{\Phi_\tau(\eta)}{\text{ReLU}_\tau(\eta)}$ |
| $\text{ReLU}_\tau(\eta)$ | $t \log(1 + \exp(\text{ReLU}_\tau(\eta)))$ | $\Phi_\tau(\eta)$ | $t \frac{\exp(\text{ReLU}_\tau(\eta))}{1 + \exp(\text{ReLU}_\tau(\eta))} \Phi_\tau(\eta)$ |
| $\text{ReLU}(\eta)$ | $t \log(1 + \exp(\text{ReLU}(\eta)))$ | $\Theta(\eta)$ | $t \frac{\exp(\text{ReLU}(\eta))}{1 + \exp(\text{ReLU}(\eta))} \Theta(\eta)$ |
| $-\text{ReLU}_\tau$ | $\text{Li}_{j+1}(\exp(-\text{ReLU}_\tau(\eta)))$ | $-\Phi_\tau(\eta)$ | $-\text{Li}_j(\exp(-\text{ReLU}_\tau(\eta))) \Phi_\tau(\eta)$ |

Table 1: Different combinations of A and R induce different soft dropout and activation functions ρ and σ . The first four rows represent cases of exponential family PCA with Gaussian, Bernoulli, non-interacting Ising and continuous Bernoulli (Loaiza-Ganem and Cunningham, 2019) likelihoods respectively. Our setting allows for general A and R , all of which can be solved using single-layer PED. Of particular interest is the case where A is quadratic but R is general, where an extended setting involving a hierarchy of latent variables can be solved using deep PED. Further details of the probabilistic interpretation of each of these cases is described in Appendix B. The function ReLU_τ is discussed in Appendix C. The functions Φ_τ and Θ are CDFs of a Gaussian with standard deviation $|\tau|$ and the Heaviside step function respectively. The function Li_{j+1} is the polylogarithm of order $j + 1$. See also Nielsen and Garcia (2009) for the case $\rho = 1$.

The outer problem learns the parameters of the network, and the inner constraint outputs predictions of the implicit layer. The inner constraint is subject to questions of well-posedness, that is, whether solutions to the problem exist and are unique. The workhorse tools for analysing well-posedness are contraction principles for fixed point equations (Hasselblatt and Katok, 2003) and convexity for optimisation problems (Wright and Recht, 2022, Chapter 2). Following usual deep learning philosophy, we ignore such questions for the outer parameter learning problem, which is likely to be highly nonconvex. Our aim is to cast problems in the form (2) with well-posed inner constraints. A problem cast in the form of (2) can use the machinery of implicit deep learning — the implicit function theorem and GPU-optimised libraries — to solve the problem.

Variants of PCA. PCA admits linear algebraic, probabilistic (Bishop, 2006) and functional interpretations (Schölkopf et al., 1997). We work with the probabilistic view (Bishop, 1998; Collins et al., 2001; Bishop, 2006; Mohamed et al., 2008; Avalos et al., 2018), since it allows for extensions with intuitive statistical meaning. Other variants of PCA are more extensively reviewed by Smallman and Artemiou (2022).

3 DERIVATION OF PED

We propose to insert nonlinear mappings $R : \mathbb{R} \rightarrow \mathbb{F}$ called the *canonical nonlinearity* into the exponential family. In a sense to be made more precise, we work with a composition $A \circ R(\cdot)$. Various combinations of A and R are given in Table 1. The derivatives $\rho := R'$ and $\sigma := (A \circ R)'$ (where defined) play important roles, and we call them the *soft dropout* and *activation* functions respectively. We consider the graphical model in Figure 1, which mirrors various versions of PCA (Bishop, 2006; Collins et al., 2001; Mohamed et al., 2008) but allows non-identity R .

3.1 MAP Estimation Using DEQs

Setup Let $H = WZ + B\mathbf{1}_{1 \times N}$, where $W \in \mathbb{R}^{d \times l}$, $B \in \mathbb{R}^d$ and $Z \in \mathbb{R}^{l \times N}$. Equivalently $H_s = WZ_s + B$ where $Z_s \in \mathbb{R}^{l \times 1}$ are the latent variables for data $Y_s = (y_{s1}, \dots, y_{sd})$. Typically $l < d$, although this need not be the case. We introduce a (potentially nonlinear) canonical map $R(\cdot)$. The canonical parameters of the distribution of the observed variables Y_s are $R(H_s) \in \mathbb{F}$. In other words, we assume that the observed vector of data Y_s is drawn from a nonlinear parameterisation of the exponential fam-

ily $\mathcal{D}_{A,\chi}(R(H_s))$ with log partition function A , and additional parameter χ . This immediately implies that the distribution is properly normalised. The nonlinear parameterisation does not violate the Fisher-Neyman factorisation (see Deisenroth et al. (2020, Theorem 6.14)). That is, T is still a sufficient statistic for $WZ_s + B$. The expectation parameter satisfies $\mu = \sigma(\eta)/\rho(\eta)$ when the right hand side is defined (see Appendix B.8). Following (1), our proposed nonlinear exponential family is

$$p(Y | Z, W, B) = \prod_{s=1}^N \prod_{i=1}^d h(y_{si}) \quad (3)$$

$$\exp \left(R((WZ_s + B)_i) T(y_{si}) - A \circ R((WZ_s + B)_i) \right).$$

For each s , we place independent zero mean iid Gaussian priors over Z_s , $Z_s \sim \mathcal{N}(0, \lambda^{-1}I)$, $Z_1 \perp \dots \perp Z_N$.

There are two settings of interest. In the first, the parameters W and B are fixed and belong to a nonempty set $\mathbb{W} \subseteq \mathbb{R}^{d \times l + d}$. The posterior over Z satisfies

$$p(Z | W, B, Y) = p(Y | Z, W, B) p(Z) / p(Y | W, B). \quad (4)$$

In the second setting, we place a prior \mathcal{P} admitting a density supported on \mathbb{W} over W and B . The joint posterior is

$$p(Z, W, B | Y) = p(Y | Z, W, B) p(Z) p(W, B) / p(Y). \quad (5)$$

In Appendix A, we show the special case where the likelihood is Gaussian and R is the identity, i.e. probabilistic PCA. This special case recovers the usual MAP estimate for Z , as well as an auto-encoder style reprojection error objective for the parameters W and B .

Results We now turn to the more general case of nonlinearly parameterised exponential family likelihoods (3). Our conditions for well-posedness of PED layers can be understood from the perspective of optimisation or fixed points. Both views lead to a spectral constraint on W up to a constant κ which includes effects from the observation Y , log partition function A and canonical nonlinearity R .

Assumption 1. *The support \mathbb{W} satisfies*

$$\mathbb{W} \subseteq \left\{ W \in \mathbb{R}^{d \times l}, B \in \mathbb{R}^d \mid \kappa \|W^\top W\|_2 < 1 \right\},$$

where $\kappa = \lambda^{-1} \left(\sup_{\eta, s, i} T(y_{si}) \rho'(\eta) - \sigma'(\eta) \right)$.

Note that if R is the identity, \mathbb{W} may be any subset of $\mathbb{R}^{d \times l + d}$ since $\rho' = 0$ and σ' is nonnegative by convexity of A . Assumption 1 ensures well-posedness when R is the identity or its second derivative is still relatively well-behaved. However, cases of interest such as ReLU require a different analysis (see § 3.2). Under Assumption 1, the

first order optimality conditions are sufficient to characterise the unique global optima of the latents given a fixed set of parameters. This first order optimality condition can be rearranged to a fixed point equation, which yields the following DEQ (8) of the form (2).

Theorem 1. *Suppose Assumption 1 holds. Let*

$$f(Z; Y_s, W, B) = \frac{1}{\lambda} W^\top (T(Y_s) \odot \rho(WZ + B) - \sigma(WZ + B)). \quad (6)$$

Any MAP estimate of (4) satisfies

$$Z_s^* = f(Z_s^*; Y_s, W, B), \quad \forall 1 \leq s \leq N, \quad (7)$$

solutions of which are guaranteed to exist and be unique. Any joint MAP estimate of (5) satisfies

$$W^*, B^* \in \operatorname{argmin}_{W, B \in \mathbb{W}} -\log p(W, B) - \log p(Z^*) + \quad (8)$$

$$\sum_{s=1}^N \mathbf{1}_{d \times 1}^\top A \circ R(WZ_s^* + B) - T(Y_s)^\top R(WZ_s^* + B)$$

subject to $Z_s^* = f(Z_s^*; Y_s, W, B), \quad \forall 1 \leq s \leq N$.

The proof is given in Appendix D. Our analysis relies on the quadratic negative log Gaussian prior $\log p(z)$, which when differentiated yields λz . When this term is isolated, a fixed point condition is obtained. When we extend our analysis to hierarchical latent variables, the prior $p(z)$ will be replaced by a layer-dependent likelihood, for example $p(Z^{(1)} | Z^{(2)})$. This means that A for layers $l > 1$ must be quadratic, but crucially, R may be nonlinear.

The fixed point (7) of a fully connected or convolutional layer (6) represents a DEQ with activation σ , soft dropout ρ , shared weights or convolutional kernel (see Appendix F) W and biases B . We name ρ soft dropout because (a) as shown in Table 1, it is often close to a $\{0, 1\}$ -valued function, and (b) its role is to multiply inputs $T(Y_s)$ by a value that is close to zero or one. This resembles common usage of dropout in explicit networks but applied to an implicit DEQ setting.

An existing but different DEQ dropout mechanism exists (Bai et al., 2019, 2020) — adapted from Gal and Ghahramani (2016b)— where for each parameter update and input Y_s , a single **static mask** is sampled and re-applied in each step of the fixed point solver. We offer an alternative theoretically motivated dropout mechanism, where the mask is a **deterministic function** of the current solution in the fixed point solver. The **mask is dynamic** (recalculated at each step of the fixed point solver). The mask is elementwise multiplied by input $T(Y_s)$. In the case $\sigma \equiv \text{ReLU}$ and $\rho \equiv \Theta$, where Θ is the Heaviside step function, the mask $\rho(WZ_s + B)$ is $\{0, 1\}$ -valued, hence “dropping out” inputs. When R is the identity the mask is 1 (no

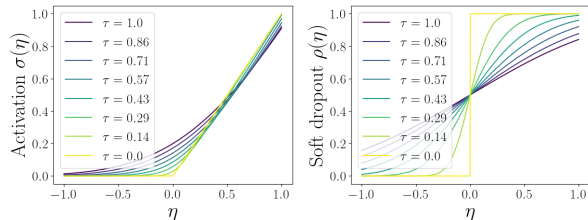


Figure 2: Choosing $A(r) = r^2/2$ and $R = \text{ReLU}_\tau$, (Left) Activation function $\sigma(\eta) = \Phi_\tau(\eta) \text{ReLU}_\tau(\eta)$ (Right) Soft dropout function $\rho(\eta) = \Phi_\tau(\eta)$ for different values of τ .

dropout). When using smooth ReLU functions ReLU_τ , the mask is the CDF of a Gaussian random variable with variance τ^2 . This is a kind of $(0, 1)$ -valued smooth Heaviside function, which becomes $\{0, 1\}$ -valued as $\tau \rightarrow 0$. Another setting is $\rho(\eta) = 1 + \tanh(\eta)$ as shown in Table 1, where the mask is $(0, 2)$ -valued which is also a smooth version of the Heaviside function, up to a factor of 2. Because these last two settings do not zero-out inputs exactly, but do multiply by the inputs similar to the mask in dropout, but in a smooth manner, we call ρ “soft dropout”.

Equations (6) and (7) resemble what practitioners might use as a generic black-box DEQ predictor, without any explicit intention to compute with an underlying statistical model. The outer optimisation is a non-Gaussian exponential family generalisation of a squared reprojection error, as detailed in Appendix A.

3.2 Rectified Activations and Hard Dropout

Smooth ReLU Define the ReLU by $\text{ReLU}(\eta) = \Theta(\eta)\eta$, where $\Theta(\cdot)$ is the Heaviside function. Let $\text{ReLU}_\tau(\eta) = \int_{\mathbb{R}} \text{ReLU}(\eta + \epsilon) p_\tau(\epsilon) d\epsilon$ denote the τ -smoothed ReLU. Here p_τ is the PDF of a zero-mean Gaussian random variable with standard deviation $|\tau|$. We have

$$\text{ReLU}_\tau(\eta) = \eta \Phi\left(\frac{\eta}{|\tau|}\right) + \frac{|\tau|}{2} \sqrt{\frac{2}{\pi}} \exp\left(-\eta^2/2\tau^2\right)$$

$$\frac{\partial}{\partial \eta} \text{ReLU}_\tau(\eta) = \Phi(\eta/|\tau|)$$

$$\frac{\partial^2}{\partial \eta^2} \text{ReLU}_\tau(\eta) = \frac{1}{\sqrt{2\pi\tau^2}} \exp\left(-\frac{\eta^2}{2\tau^2}\right) = p_\tau(\eta),$$

where Φ is the cdf of the standard Gaussian random variable. See Appendix C for details and Figure 2 for a visualisation of σ and ρ when $A(r) = r^2/2$.

The Need for a Separate Analysis We consider the special case of Theorem 1 when the exponential family is Gaussian and $R = \text{ReLU}_\tau$. We find $\rho(\eta) = \Phi_\tau(\eta)$ and $\sigma(\eta) = \Phi_\tau(\eta) \text{ReLU}_\tau(\eta)$. When τ is small, ρ and σ begin to look like the Heaviside step function and

the ReLU. This amounts to using a DEQ consisting of dropout and ReLU activations. Assumption 1 requires $\frac{\sqrt{\lambda^{(0)}} \max_{s,i} y_{si}}{\tau \lambda} \|\mathbf{W}^\top \mathbf{W}\|_2 < 1$, where $\sqrt{\lambda^{(0)}}$ is the standard deviation of the Gaussian likelihood. As $\tau \rightarrow 0$, this condition becomes impossible to satisfy for nontrivial \mathbf{W} .

Instead of requiring that the inner constraint of a DEQ finds a unique global minimum, we relax the constraint so that it finds a local minimum. This allows us to use a weaker assumption that may be satisfied by the ReLU.

Assumption 2. Suppose $R(\eta) = \text{ReLU}(\eta)$. Let $a < \infty$ be a Lipschitz constant of A' . The support \mathbb{W} satisfies

$$\mathbb{W} \subseteq \{\mathbf{W} \in \mathbb{R}^{d \times l}, b \in \mathbb{R}^d \mid \kappa \|\mathbf{W}^\top \mathbf{W}\|_2 < 1\},$$

where $\kappa = a\lambda^{-1}$.

An interesting property of the ReLU is that the dropout function (where it is defined) maps to a finite set of values. We may form and analyse a set of nicely-behaved fixed point equations for each dropout value instead of analysing one fixed point equation involving Θ , whose derivative is poorly behaved. This allows us to avoid dealing with the non-continuous nature of the dropout function Θ .

Theorem 2. Suppose $R(\eta) = \text{ReLU}(\eta)$ and fix some parameters (\mathbf{W}, B) and data index s . Any stationary point Z_s^* of objective (4) is a solution to

$$Z_s^* = \frac{1}{\lambda} \mathbf{W}^\top (T(Y_s) \odot \Theta(\mathbf{W}Z_s^* + B) - \text{ReLU}(\mathbf{W}Z_s^* + B)).$$

Under Assumption 2, there exists at least 1 and at most 2^d stationary points, all of which are local minima.

The proof is given in Appendix D. As expected, the fixed point equations in Theorems 1 and 2 are identical, up to a substitution and ignoring the nondifferentiable point. Dropout (where defined) maps to a finite set for other activations such as the Leaky ReLU and hard sigmoid, and Theorem 2 can likely be extended to these cases.

3.3 Deep PED

It is well-known that compositions of linear transformations are linear transformations. This means that stacking multiple PED layers when the likelihood is Gaussian and R is the identity is trivial. When a nonlinearity is introduced, the analysis becomes challenging, but might offer the chance of more expressive neural network models due to the deep nonlinear structure of the architecture (Poole et al., 2016). As we now show, one may stack PED layers in a principled manner to obtain a deep PED architecture when A is quadratic and R is nonlinear.

We consider a natural generalisation of Figure 1 in Figure 3. Instead of a single latent prior node, we have L latent nodes built as a hierarchical prior. Here every object is

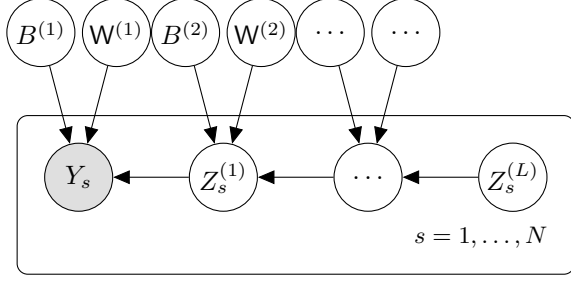


Figure 3: Restricting our attention to the Gaussian case $A(\eta) = \eta^2/2$, we consider a deep graphical model. Here each of the latent $Z^{(l)}$ variables are Gaussian conditioned on their parents, with variance $1/\lambda^{(l)}$ and canonical parameter $R^{(l+1)}(W^{(l+1)}Z^{(l+1)} + B^{(l+1)})$. Surprisingly, a DEQ can be used to find local minima in the latent variables.

given a superscript denoting the layer $1 \leq l \leq L$ to which it belongs. The conditional distribution of the latent node in layer l is Gaussian in exponential family form $\mathcal{D}_{A,\chi}$, having $A(r) = r^2/2$ and an additional standard deviation parameter $\nu = \sqrt{\lambda^{(l)}}^{-1}$. Variables carrying over from the shallow case in § 3 are renamed $Y_s = Z_s^{(0)}$, $\lambda = \lambda^{(1)}$ and $(W, B) = (W^{(1)}, B^{(1)})$. Let $\theta = \{W^{(l)}, B^{(l)}\}_{l=1}^L$. With s fixed, the joint posterior over $\theta, \{Z^{(l)}\}_{l=1}^L$ of the graphical model in Figure 3 satisfies

$$p(Z^{(1)}, \dots, Z^{(L)}, \theta | Y) \quad (9)$$

$$\propto p(Y | Z^{(1)}, \theta) \prod_{l=1}^{(L-1)} p(Z^{(l)} | Z^{(l+1)}, \theta) p(Z^{(L)}) p(\theta).$$

As in the shallow case, we find the fixed point condition implied by the stationary points of the posterior, leading to a DEQ layer. Let $\zeta = (Z^{(1)}; \dots; Z^{(L)}) \in \mathbb{R}^D$ be an augmented state and write

$$G^{[1:L]}(\zeta) = \left(G^{(1)}(Z^{(1)}; Z^{(0)}, Z^{(2)}); \dots; G^{(L)}(Z^{(L)}; Z^{(L-1)}, 0) \right) \quad (10)$$

where

$$G^{(l)}(Z^{(l)}; Z^{(l-1)}, Z^{(l+1)}) = \frac{1}{\sqrt{\lambda^{(l)}}} R^{(l+1)}(W^{(l+1)}Z^{(l+1)} + b^{(l+1)}) + \frac{1}{\lambda^{(l)}} W^{(l)\top} \left(\rho^{(l)}(W^{(l)}Z^{(l)} + b^{(l)}) \odot \sqrt{\lambda^{(l-1)}} Z^{(l-1)} - \sigma^{(l)}(W^{(l)}Z^{(l)} + b^{(l)}) \right).$$

In Appendix E, we show that fixed points of (10) are the stationary points of (9) under certain conditions. This augmentation of the state space shares an interesting connection to the construction used to argue that a single layer

DEQ is sufficient to represent multiple DEQ layers (Bai et al., 2019, Theorem 2). Theorem 2 of Bai et al. (2019) says that instead of stacking two DEQ layers with widths r and d , one can use a single DEQ layer with width $r + d$ to represent the same function. We find that instead of stacking L coupled DEQ layers for each latent $Z^{(l)}$, we may use a single DEQ layer with a width that is the sum of individual widths. Interestingly, due to the way we constructed the augmented state, we can identify different subsets of coordinates as corresponding with latent variables at different layers l . *To the best of our knowledge, this is the first such DEQ with this interpretable quality.* Bai et al. (2019)’s statement is about the representation or capacity that can be obtained by increasing layer widths, whereas our observation is one about modelling, bias and interpretability. A related augmentation and stacking of DEQ layers into a single wide layer is exploited in the context of diffusion models (Pokle et al., 2022, Equation 11).

Unfortunately, formally describing the nature of the stationary points which are the fixed points of $G^{[1:L]}$ is difficult — a limitation of our work. See Appendix E for a sketch.

3.4 Related Work and Other Considerations

Several theoretical issues of DEQs have been investigated. These include methods for ensuring certified robustness (Wei and Kolter, 2022), constraints and architectures that ensure or help well-posedness of the fixed point condition (Winston and Kolter, 2020; Revay et al., 2020; El Ghaoui et al., 2021) and connections to nonparametric statistical estimators (Tsuchida et al., 2022). Li et al. (2022) design DEQ layers from a (non-statistically motivated) optimisation perspective. Other works view DEQs in light of classical inverse problems (Gilton et al., 2021; Riccio et al., 2022; Guan et al., 2022). We note that Jacobian regularisation (Bai et al., 2021) may be used as a soft penalty on the weights to allow Assumptions 1, 2 hold, where required. Implicit networks offer the opportunity to inject a degree of interpretability into neural network models, if the problem that the implicit layer solves is itself interpretable. This is especially true for optimisation-based layers (Monga et al., 2021), but not necessarily for DEQ layers. Our work identifies problems which DEQ layers solve, and shows that such problems have meaningful statistical interpretations.

Linearly Parameterised Exponential Families Exponential families are widely used due to their flexibility and the strict convexity and smoothness properties of the log likelihood of (1) in r . Typically, the canonical parameters are set to be a parameterised linear transformation of an input (in supervised settings (McCullagh and Nelder, 1989; Loader, 2006)) or a latent variable (in unsupervised settings (Collins et al., 2001; Mohamed et al., 2008)). This extends when the linear transformation represents an element

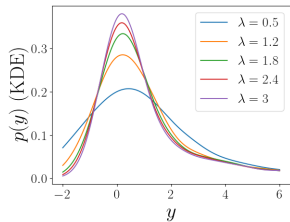


Figure 4: KDE of (11) in 1 dimension with $\tau = 0.5$ using 10000 samples of W , z and y .

of an RKHS (Canu and Smola, 2006). Linear transformations compose well with exponential families because they preserve convexity (in the parameters) and smoothness.

A is Not All You Need Using a non-identity R allows for a number of potential benefits. If R is chosen to be the identity, the activation function σ must be increasing by strict convexity of A . Therefore, activations such as the ReLU, GELU, (nonconstant) periodic functions, or Gaussian activations are not obtainable with linear R , each of which have been shown to possess useful properties (Nair and Hinton, 2010; Hendrycks and Gimpel, 2016; Meronen et al., 2021) (although perhaps in slightly different architectures). We further detail this lack of expressiveness for A for the softplus activation in Appendices B.5 and B.6.

Another advantage of nonlinear R is that it can map inputs into an acceptable range for the canonical parameter of the exponential family. For example, an exponential distribution requires a negative canonical parameter, which can be ensured by choosing $R(\eta) = -\text{ReLU}_\tau(\eta)$, which may be easier than constraining η to be negative.

Finally, R allows us to model situations that cannot be modelled using the linearly parameterised exponential family. For example, let $A(r) = r^2/2$ and $R(\eta) = \text{ReLU}_\tau(\eta)$. In order to visualise what this means, consider the case $d = l = 1$. With a $\mathcal{N}(0, \sqrt{2})$ prior over $W \in \mathbb{R}$ and $z \in \mathbb{R}$,

$$p(y) = \mathbb{E}_{W, z \sim \mathcal{N}(0,1)} [p(y | \text{ReLU}_\tau(Wz))] \quad (11)$$

is visualised by sampling 10000 W , z and y and plotting a KDE for y . This is shown in Figure 8, where the variance of the conditional distribution of y given Wz is λ^{-1} .

A Recipe for Mapping σ and ρ to Nonlinearly Parameterised Gaussians Using the various relations between A , R , σ and ρ , it is possible to understand soft dropout and activation functions in terms of nonlinearly parameterised exponential family likelihoods. We demonstrate this for quadratic $A(v) = v^2/2$ and nonlinear R . We have by definition that $\sigma(\eta) = (A \circ R)'(\eta) = \frac{1}{2} \frac{d}{d\eta} R^2(\eta)$, so that

$$|R(\eta)| = \sqrt{\int 2\sigma(\eta) d\eta}, \quad \rho(\eta) = \sigma(\eta)/R(\eta). \quad (12)$$

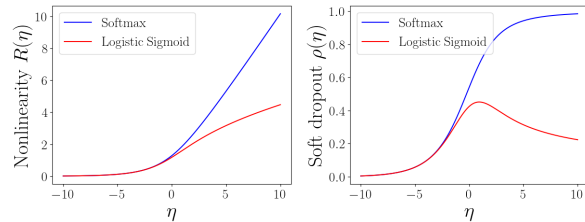


Figure 5: Choosing σ to be logistic sigmoid or softmax, the nonlinearity R and soft dropout function ρ , in (12).

As discussed above and in Appendices B.5 and B.6, if σ is the commonly used softplus activation, it does not have an interpretation as the derivative of a log-partition function. However, we may understand it through a nonlinear transformation $R(\eta) = \sqrt{\int 2 \log(1 + \exp(\eta)) d\eta} = \sqrt{-2\text{Li}_2(-e^\eta)}$ of the canonical parameter of a Gaussian distribution, allowing it to model real-valued data. Here Li_2 denotes the dilogarithm function. See Appendix B.6 for details. See Figure 5 for plots of R and ρ .

This construction can also be used to understand instances of σ that are derivatives log-partition functions of non-Gaussian exponential families alternatively as arising from nonlinear transformations of Gaussians. For example, as is well-known, the logistic sigmoid is the inverse-link function of a Bernoulli random variable, suggesting that it should be used to model binary-valued data. However, by (12), we may understand it through a nonlinear transformation $R(z) = \sqrt{\int 2(1 + \exp(-z))^{-1} dz} = \sqrt{2 \log(1 + e^z)}$ of the canonical parameter of a Gaussian distribution, allowing it to model real-valued data. See Appendix B.7 for details. See Figure 5 for plot of R and ρ .

Bregman Divergences The negative log-likelihood of an exponential family can always be written as a Bregman divergence $B_\phi(y, \mu)$ in terms of its expectation parameters μ (with an additional factor that is constant with respect to the expectation parameter) and convex conjugate ϕ of A (Banerjee et al., 2005, Theorem 4). For each exponential family, there is a unique Bregman divergence. For example, the Gaussian corresponds with the squared loss, the Poisson corresponds with relative entropy, and Exponential corresponds with the Itakura Saito divergence. Under our nonlinear parameterisation, the expectation parameter is $\mu = A' \circ R(\eta)$, leading to a space of nonlinearly parameterised Bregman divergences. See Appendix B.8.

Convolutional Layers Operators that perform neural network convolution may be represented as sparse matrices with repeated entries. Our results apply when \mathbb{W} is a space of convolutional layers. The transpose operator can be computed using transposed convolution (Zeiler et al., 2010). The constraints in Assumption 1 and 2

may be computed efficiently when the norm is the spectral norm (Sedghi et al., 2019). See Appendix F.

4 ILLUSTRATION ON A 2D LATENT SPACE



Figure 6:
ground truth, Z

In this section, we compare the performance and key features of PCA, tSNE (Van der Maaten and Hinton, 2008), UMAP (McInnes et al., 2018) and PED. It can be difficult to evaluate latent projections. Our evaluation focuses on two issues: visualising the latent space, and performance in a downstream classifier.

Ground Truth Latents We generate datasets through graphical model in Figure 1, except that Z is fixed as shown in Figure 6. Each latent Z_s is 2-dimensional. For each distribution shown in the left column of Figure 7, we generate 100 parameters W according to the graphical model and them sample the corresponding Y . Each observation Y_s is 50-dimensional. This results in 100 different datasets for each distribution, each with known ground truth Z .

Downstream Performance and Deep Learning Compatibility A key feature of PED is that it is compatible with any other neural network layer. This means that we can fine-tune backbone PED layers that are pretrained in an unsupervised manner on a supervised task with a head network. In contrast, since PCA is a linear mapping, composing it with a head network and optimising the result is equivalent to just using the head network. Similarly, other dimensionality reduction techniques that do not possess neural network parameters such as UMAP cannot benefit from fine-tuning. tSNE, which is better thought of as a visualisation tool than a dimensionality reduction tool, is even less suitable in this respect. The mapping produced by tSNE on training data cannot be used on testing data, so it is incompatible in a pipeline with downstream tasks.

We use a simple fully connected head with widths 2, 100, 1 and ReLU activations at width 100. Together with a PED or (untrainable) PCA/UMAP backbone, we train the full network to predict $g(Z_s) \in \mathbb{R}$ given an input $Y_s \in \mathbb{R}^{50}$ by minimising the sum of squared errors. This is done for each of the 100 randomly generated datasets. The corresponding $Z_s \in \mathbb{R}^2$ are not known to the network. This is a simple to visualise and deceptively difficult task that relies heavily on the quality of the latent embedding. Full hyperparameter details and further experiments are given in Appendix G.

A limitation of our evaluation is that we did not tune hyperparameters of any algorithm. However, if latents were to be used in a downstream task that was not known *a priori*, there would be no way of objectively tuning hyperparameters.

We have demonstrated the important benefit that PED has over all other techniques — it is end-to-end differentiable and can be fine-tuned. PED motivates use of DEQ layers for supervised tasks as using fine-tuned latents.

5 DISCUSSION AND CONCLUSION

We proposed a canonical nonlinear map for parameters of an exponential family, and considered the problem of MAP estimation in PCA. We derived shallow and deep DEQ architectures that solve this problem, and called these shallow and deep PED respectively. Our main contribution is the theoretical analysis. It grounds DEQ architectures in a probabilistic framework, and shows how certain architectural choices can be related back to statistical assumptions on the observations. More generally, use of DEQ layers admitting the same form as PED layers can be viewed as fine-tuning for a supervised task.

As noted by Bai et al. (2019), DEQs built from contraction mappings may be understood as infinitely deep neural networks with shared weights. In this sense, our analysis motivates the use of such infinitely deep networks. Machine learning has a history of taking algorithms that at first seem to require infinitely many samples or steps and running those algorithms for one or very few steps (Hinton, 2002; Sutskever and Tieleman, 2010; Kingma and Welling, 2013). In future, it might be possible that few steps of a DEQ forward iteration can be understood using the theory of infinitely many forward iterations, just as well-behaved wide neural networks can now be understood using the theory of infinitely wide neural networks (Jacot et al., 2018).

Another direction is to extend PED to a Bayesian setting by taking the Hessian of the negative log likelihood evaluated at the MAP. Expressions of second order approximations are isolated in Appendix H, and may be useful for future investigations, for example in Laplace approximation (Daxberger et al., 2021). Interestingly, in some cases the curvature at the MAP depends on a deterministic dropout, which offer another (Srivastava et al., 2014; Gal and Ghahramani, 2016a) perspective on dropout.

Empirical success stories of DEQs applied to language prediction and computer vision usually other deep learning elements such as activations that are not canonical inverse link functions (such as the ReLU), attention and/or multi-scale residual blocks (Bai et al., 2020). In future, we hope to cast these elements in light of MAP estimation, and further unpack the relationship between R and the statistician’s inverse link function. Although we implicitly focused on the setting where $l < d$ in the shallow case, and $d^{(0)} < d^{(1)} \dots < d^{(L)}$ in the deep case, none of our analysis actually requires this. This allows one to construct general DEQ layers for use in unsupervised tasks, which may be interpreted as fine-tuned nonlinearly parameterised exponential family PCA.

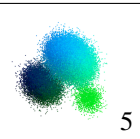
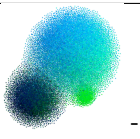
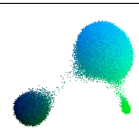
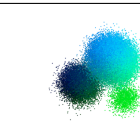
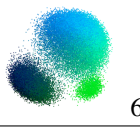
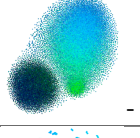
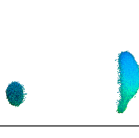
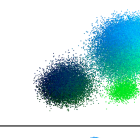
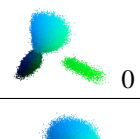
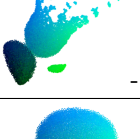
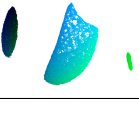
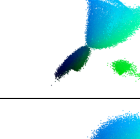
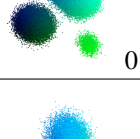
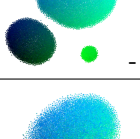
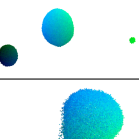
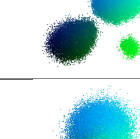
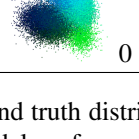
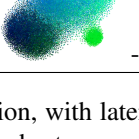
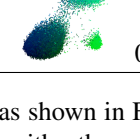
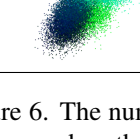
| | PCA (sklearn) | tSNE | UMAP | PED |
|---|---|---|--|--|
| Gaussian $A(r) = r^2/2$ $R(\eta) = \eta$ |  5 |  - |  0 |  95 |
| Bernoulli $A(r) = \log(1 + e^r)$ $R(\eta) = \eta$ |  6 |  - |  0 |  94 |
| Poisson $A(r) = \exp(r)$ $R(\eta) = \eta$ |  0 |  - |  16 |  84 |
| Rectified-mean Gaussian (shallow) $A(r) = r^2/2$ $R(\eta) = \text{ReLU}(\eta)$ $L = 1$ |  0 |  - |  11 |  89 |
| Rectified-mean Gaussian (deep) $A(r) = r^2/2$ $R(\eta) = \text{ReLU}(\eta)$ $L = 2, d^{(1)} = 30, d^{(2)} = 2$ |  0 |  - |  0 |  100 |

Figure 7: The left column describes the ground truth distribution, with latents as shown in Figure 6. The number in each cell represents the number of times that model performed the best compared with others when used as the input for a downstream task over 100 random trials. Interestingly, PCA and PED often produce visually quite similar results to each other and the ground truth. However, PED is able to obtain much better results on downstream tasks on account of its ability to be fine-tuned. While UMAP’s results are visually pleasing, its embeddings do not bear much similarity with the ground truth, except in the Gaussian case. For non-Gaussian cases, UMAP sometimes places latents in good clusters, but fails to globally position the clusters accurately relative to one another. tSNE is incompatible in a pipeline with a downstream task. tSNE produces globally accurate positioning, but sometimes adds artefacts into the visualisation. Embeddings are an equivalence class up to a rotation and scaling, since W and Z are non-identifiable; we rotated images by hand to align them. For PED, we visualise the latents in an orthogonal basis, i.e. shown are RZ , where $W = QR$ is a QR decomposition.

We hope that our link between neural network design choices (activation functions, dropout, and layer structure) and their corresponding statistical distribution counterparts (log partition function, canonical nonlinearity) enables other researchers to gain a better understanding of equilibrium networks.

Acknowledgments

Both authors gratefully acknowledge the support of CSIRO’s Machine Learning and Artificial Intelligence Future Science Platform. Ke Sun and Frank Nielsen provided helpful feedback on an earlier version of this paper. We thank the anonymous reviewers for their constructive comments.

References

- Avalos, M., Nock, R., Ong, C. S., Rouar, J., and Sun, K. (2018). Representation learning of compositional data. *Advances in Neural Information Processing Systems*, 31.
- Bai, S., Koltner, J. Z., and Koltun, V. (2019). Deep equilibrium models. *Advances in Neural Information Processing Systems*, 32.
- Bai, S., Koltun, V., and Koltner, J. Z. (2020). Multiscale deep equilibrium models. *Advances in Neural Information Processing Systems*, 33:5238–5250.
- Bai, S., Koltun, V., and Koltner, Z. (2021). Stabilizing equilibrium models by jacobian regularization. In *International Conference on Machine Learning*, pages 554–565. PMLR.
- Banerjee, A., Merugu, S., Dhillon, I. S., Ghosh, J., and Lafferty, J. (2005). Clustering with bregman divergences. *Journal of machine learning research*, 6(10).
- Bishop, C. (1998). Bayesian PCA. *Advances in Neural Information Processing Systems*, 11.

- Bishop, C. M. (2006). *Pattern recognition and machine learning*, volume 4. Springer.
- Canu, S. and Smola, A. (2006). Kernel methods and the exponential family. *Neurocomputing*, 69(7-9):714–720.
- Chen, R. T., Rubanova, Y., Bettencourt, J., and Duvenaud, D. K. (2018). Neural ordinary differential equations. *Advances in Neural Information Processing Systems*, 31.
- Collins, M., Dasgupta, S., and Schapire, R. E. (2001). A generalization of principal components analysis to the exponential family. *Advances in Neural Information Processing Systems*, 14.
- Daxberger, E., Kristiadi, A., Immer, A., Eschenhagen, R., Bauer, M., and Hennig, P. (2021). Laplace redux—effortless Bayesian deep learning. *Advances in Neural Information Processing Systems*, 34.
- Deisenroth, M. P., Faisal, A. A., and Ong, C. S. (2020). *Mathematics for Machine Learning*. Cambridge University Press.
- Dingle, R. (1957). The Bose-Einstein integrals. *Applied Scientific Research, Section A*, 6(1):240–244.
- Efron, B. (2020). Prediction, estimation, and attribution. *International Statistical Review*, 88:S28–S59.
- El Ghaoui, L., Gu, F., Travacca, B., Askari, A., and Tsai, A. (2021). Implicit deep learning. *SIAM Journal on Mathematics of Data Science*, 3(3):930–958.
- Gal, Y. and Ghahramani, Z. (2016a). Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1050–1059.
- Gal, Y. and Ghahramani, Z. (2016b). A theoretically grounded application of dropout in recurrent neural networks. *Advances in Neural Information Processing Systems*, 29.
- Gilton, D., Ongie, G., and Willett, R. (2021). Deep equilibrium architectures for inverse problems in imaging. *IEEE Transactions on Computational Imaging*, 7:1123–1133.
- Gould, S., Hartley, R., and Campbell, D. J. (2021). Deep declarative networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- Guan, P., Jin, J., Romberg, J., and Davenport, M. A. (2022). Loop unrolled shallow equilibrium regularizer (LUSER) - a memory-efficient inverse problem solver. In *NeurIPS 2022 AI for Science: Progress and Promises Workshop*.
- Hasselblatt, B. and Katok, A. (2003). *A first course in dynamics: with a panorama of recent developments*. Cambridge University Press.
- Hendrycks, D. and Gimpel, K. (2016). Gaussian error linear units (GELUs). *arXiv preprint arXiv:1606.08415*.
- Hinton, G. E. (2002). Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800.
- Hutchinson, J. E. (1981). Fractals and self similarity. *Indiana University Mathematics Journal*, 30(5):713–747.
- Jacot, A., Gabriel, F., and Hongler, C. (2018). Neural tangent kernel: Convergence and generalization in neural networks. *Advances in Neural Information Processing Systems*, 31.
- Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In *International Conference on Learning Representations*.
- Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.
- Li, M., Wang, Y., Xie, X., and Lin, Z. (2022). Optimization inspired multi-branch equilibrium models. In *International Conference on Learning Representations*.
- Loader, C. (2006). *Local regression and likelihood*. Springer Science & Business Media.
- Loaiza-Ganem, G. and Cunningham, J. P. (2019). The continuous bernoulli: fixing a pervasive error in variational autoencoders. *Advances in Neural Information Processing Systems*, 32.
- Loshchilov, I. and Hutter, F. (2019). Decoupled weight decay regularization. In *International Conference on Learning Representations*.
- McCullagh, P. and Nelder, J. (1989). *Generalized Linear Models, Second Edition*. Chapman & Hall/CRC Monographs on Statistics & Applied Probability. Taylor & Francis.
- McInnes, L., Healy, J., Saul, N., and Großberger, L. (2018). Umap: Uniform manifold approximation and projection. *Journal of Open Source Software*, 3(29).
- Meronen, L., Trapp, M., and Solin, A. (2021). Periodic activation functions induce stationarity. In *Advances in Neural Information Processing Systems*, volume 34, pages 1673–1685.
- Mohamed, S., Ghahramani, Z., and Heller, K. A. (2008). Bayesian exponential family PCA. *Advances in Neural Information Processing Systems*, 21.
- Monga, V., Li, Y., and Eldar, Y. C. (2021). Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing. *IEEE Signal Processing Magazine*, 38(2):18–44.
- Nair, V. and Hinton, G. E. (2010). Rectified linear units improve restricted Boltzmann machines. In *International Conference on Machine Learning*.
- Nielsen, F. and Garcia, V. (2009). Statistical exponential families: A digest with flash cards. *arXiv preprint arXiv:0911.4863*.

- Pokle, A., Geng, Z., and Kolter, J. Z. (2022). Deep equilibrium approaches to diffusion models. In *Advances in Neural Information Processing Systems*.
- Poole, B., Lahiri, S., Raghu, M., Sohl-Dickstein, J., and Ganguli, S. (2016). Exponential expressivity in deep neural networks through transient chaos. *Advances in Neural Information Processing Systems*, 29.
- Revay, M., Wang, R., and Manchester, I. R. (2020). Lipschitz bounded equilibrium networks. *arXiv preprint arXiv:2010.01732*.
- Riccio, D., Ehrhardt, M. J., and Benning, M. (2022). Regularization of inverse problems: Deep equilibrium models versus bilevel learning. *arXiv preprint arXiv:2206.13193*.
- Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215.
- Schölkopf, B., Smola, A., and Müller, K.-R. (1997). Kernel principal component analysis. In *International conference on artificial neural networks*, pages 583–588. Springer.
- Sedghi, H., Gupta, V., and Long, P. M. (2019). The singular values of convolutional layers. In *International Conference on Learning Representations*.
- Smallman, L. and Artemiou, A. (2022). A literature review of (sparse) exponential family PCA. *Journal of Statistical Theory and Practice*.
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.
- Sutskever, I. and Tieleman, T. (2010). On the convergence properties of contrastive divergence. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 789–795. JMLR Workshop and Conference Proceedings.
- Tsuchida, R., Yong, S. Y., Armin, M. A., Petersson, L., and Ong, C. S. (2022). Declarative nets that are equilibrium models. In *International Conference on Learning Representations*.
- Van der Maaten, L. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of machine learning research*, 9(11).
- Wainwright, M. and Jordan, M. (2008). Graphical Models, Exponential Families, and Variational Inference. *Foundations and Trends in Machine Learning*, 1(1-2):1–305.
- Wei, C. and Kolter, J. Z. (2022). Certified robustness for deep equilibrium models via interval bound propagation. In *International Conference on Learning Representations*.
- Winston, E. and Kolter, J. Z. (2020). Monotone operator equilibrium networks. *Advances in Neural Information Processing Systems*, 33:10718–10728.
- Wright, S. J. and Recht, B. (2022). *Foundations of Smooth Optimization*. Cambridge University Press.
- Zeiler, M. D., Krishnan, D., Taylor, G. W., and Fergus, R. (2010). Deconvolutional networks. In *2010 IEEE Computer Society Conference on computer vision and pattern recognition*, pages 2528–2535. IEEE.

A The special case of Gaussian distributions

In this section, we work under the assumption is that the exponential family model is a Gaussian distribution in order to illustrate a simple special case of Theorem 1. We follow (Deisenroth et al., 2020, §10.7) for probabilistic PCA, with some small differences. Firstly, our setup allows for a prior over joint W, B , and we consider point estimates of Z_s . Secondly, our prior over Z has variance λ^{-1} and the Gaussian likelihood has unit variance, whereas in (Deisenroth et al., 2020, §10.7) Z has a unit variance and the Gaussian likelihood has variance σ^2 . As we shall see, our resulting MAP estimates are the same, up to this scaling reparameterisation.

For the Gaussian, we have $T(y) = y/b$, where $b > 0$ is a fixed scale parameter, $A(v) = v^2/2$, and $R = 1$. Without loss of generality, we take $b = 1$, since input data Y may be preprocessed to account for different scaling. Let $\|\cdot\|_F$ denote the Frobenius matrix norm. Following (3), the negative log posterior of (4) is

$$-\log p(Z | W, B, Y) = -\log h(Y) + \frac{\lambda}{2} \|Z\|_F^2 + \sum_{s=1}^N \sum_{i=1}^d \frac{1}{2} (WZ_s + B)_i^2 - y_{si} (WZ_s + B)_i.$$

This is a strongly convex objective, whose derivative is zero at the unique minima Z^* . Differentiating with respect to Z_s , we have that the minimiser Z^* satisfies

$$\begin{aligned} 0 &= \lambda Z_s^* + W^\top (WZ_s^* + B - Y_s) \\ Z_s^* &= \frac{1}{\lambda} W^\top (Y_s - (WZ_s^* + B)). \end{aligned} \quad (13)$$

This is a DEQ with linear activation having evaluation $(WZ_s + B)$. The dropout function is 1. The reprojection error loss is obtained by considering the minimiser of the joint posterior

$$\begin{aligned} -\log p(Z, W, B | Y) &= -\log p(W, B) - \log p(Z | W, B, Y) \\ Z^*, W^*, B^* &= \operatorname{argmin}_{Z, W, B} -\log p(W, B) - \log p(Z | W, B, Y) \\ W^*, B^* &= \operatorname{argmin}_{W, B} -\log p(W, B) - \log p(Z^* | W, B, Y), \quad \text{subject to } Z^* = \operatorname{argmin}_Z -\log p(Z | W, B, Y). \end{aligned}$$

The constraint is equivalent to (13). Substituting for the outer minimisation problem, we have

$$\begin{aligned} W^*, B^* &= \operatorname{argmin}_{W, B} -\log p(W, B) + \sum_{s=1}^N \sum_{i=1}^d \frac{1}{2} (WZ_s + B)_i^2 - y_{si} (WZ_s + B)_i \\ \text{subject to } Z_s^* &= \frac{1}{\lambda} W^\top (Y_s - (WZ_s^* + B)) \quad \forall s = 1, \dots, N. \end{aligned}$$

Or, more explicitly as a squared reprojection error,

$$\begin{aligned} W^*, B^* &= \operatorname{argmin}_{W, B} -\log p(W, B) + \frac{1}{2} \sum_{s=1}^N \sum_{i=1}^d \left((WZ_s + B)_i - y_{si} \right)^2 \\ \text{subject to } Z_s^* &= \frac{1}{\lambda} W^\top (Y_s - (WZ_s^* + B)) \quad \forall s = 1, \dots, N. \end{aligned} \quad (14)$$

In this special case, we may rearrange the constraint to a more familiar form with the help of the push-through (Woodbury) matrix identity,

$$\begin{aligned} \lambda Z_s^* + W^\top WZ_s^* &= W^\top (Y_s - B) \\ Z_s^* &= (\lambda I + W^\top W)^{-1} W^\top (Y_s - B) \\ Z_s^* &= W^\top (\lambda I + WW^\top)^{-1} (Y_s - B), \end{aligned}$$

which is the same as Equation (10.74) of Deisenroth et al. (2020), up to a scaling reparameterisation.

Finally, we note a difference in approaches typically identified as ‘‘latent variable models’’ and ‘‘neural network autoencoder models’’. In the former, the marginal distribution $p(Y | W, B)$ or $p(W, B | Y)$ obtained by integrating out latent variables Z is used as an objective for point estimation of parameters W and B (see Deisenroth et al. (2020, Remark p. 342)). In the latter, a reprojection error like (14), which depends on the latent variable, is used as an objective for estimation the parameters W and B .

B Exponential family calculations

B.1 Construction of exponential families

Let $\mathbb{Y} \subseteq \mathbb{R}$ and form a measurable space $(\mathbb{Y}, \mathcal{B})$ for some sigma algebra \mathcal{B} . Let ω be some reference measure (nominally Lebesgue or counting) and let $h : \mathbb{Y} \rightarrow \mathbb{R}_{\geq 0}$ be some ω -integrable function, $\int_{\mathbb{Y}} h(y) d\omega(y) < \infty$. Let T be some measurable function and let $\mathbb{F} \subseteq \mathbb{R}$ be the set of all canonical parameters r such that $\int_{\mathbb{Y}} \exp(T(y)r) h(y) d\omega(y) < \infty$. Assume that \mathbb{F} is open. Define the *log partition function* $A : \mathbb{F} \rightarrow \mathbb{R}$ by $A(r) = \log \int_{\mathbb{Y}} \exp(T(y)r) h(y) d\omega(y)$. Call

$$p(y | r) = h(y) \exp(rT(y) - A(r)) \quad (15)$$

the PDF of a minimal regular exponential family. For such a family, A is both infinitely differentiable and strictly convex (Wainwright and Jordan, 2008, Proposition 3.1). Strictly convex and infinitely differentiable functions A are admissible if and only if $\exp(A(ir))$ is a positive definite function. See Lemma 1, Appendix B.2. A acts as a cumulant generating function, so that in particular the expected value under $p(y | r)$ is $\mu = A'(r)$. The parameter μ is called the expectation parameter. Following common convention, we will use $p(\cdot | \cdot)$ to denote all (conditional) probability density functions, with a meaning clear from the arguments. We write $y \sim \mathcal{D}_{A,\chi}(r)$ to mean that a scalar-valued random variable y follows an exponential family with log partition function A , scalar canonical parameter r and additional parameter ν^3 . If $Y = (y_1, \dots, y_d)$ and $V = (r_1, \dots, r_d)$, we write $Y \sim \mathcal{D}_{A,\chi}(V)$ to mean $y_i \sim \mathcal{D}_{A,\chi}(r_i)$ for all $1 \leq i \leq d$, with each y_i mutually conditionally independent given H .

B.2 How to check whether a log partition function is admissible

Lemma 1. *An infinitely differentiable and strictly convex A is an admissible log partition function for a minimal, regular, one-dimensional exponential family if and only if*

- *there exists some c such that $c \exp(A(r))$ is the moment generating function of some random variable evaluated at $r \in \mathbb{F} \subseteq \mathbb{R}$.*
- *$\exp(A(ir))$ is a positive definite function evaluated at $r \in \mathbb{R}$ and is continuous at $r = 0$.*

Proof. Define

$$q(y | r) = h(y) \exp(T(y)r - A(r))$$

for some nonnegative h and $A : \mathbb{F} \rightarrow \mathbb{R}$ with $\mathbb{F} \subseteq \mathbb{R}$. For a given A , we would like to determine whether there exist an h such that $q(\cdot | r)$ defines a valid probability density function (or mass function). This is true if and only if $ch(\cdot)$ for some $c > 0$ is the probability density function (or probability mass function) of a random variable y satisfying

$$\begin{aligned} \frac{1}{c} \mathbb{E}_{y \sim ch(\cdot)} [\exp(T(y)r - A(r))] &= 1 \\ \mathbb{E}_{y \sim ch(\cdot)} [\exp(T(y)r)] &= c \exp(A(r)), \end{aligned} \quad (16)$$

which is the moment generating function of a random variable $T(y)$. Alternatively, we may view $\phi(r) := c \exp(A(ir))$ as the characteristic function of a random variable $T(y)$ evaluated at real value r . Choose $c = \exp(-A(0))$. By Bochner's theorem, ϕ is a characteristic function if and only if it is continuous at 0 and ϕ is positive definite. \square

B.3 Antiderivative of hyperbolic tangent

Choose $A(r) = \int \tanh(r) dr = \log \cosh(r)$. One readily observes that A is admissible since $\exp(A(ir)) = \cosh(r)$, which is positive definite.

Since the characteristic function of the base pdf is 2π -periodic, it coincides with a discrete integer-valued random variable. In fact, we can construct an exponential family supported on $\{-1, 1\}$ with this choice of A , $T(y) = y$ and uniform $h(y) = 1/2$ by verifying that (16) holds with $c = 1$. This is a non-interacting Ising model.

³Some exponential families have an additional parameter that is not a canonical parameter. For example, Gaussian has a scale parameter and Laplace has a centering parameter. Both A and ν are fixed and not learned.

B.4 Rectified canonical parameters

Appendix C.1 and C.2 describe some special cases where R is ReLU or ReLU_τ .

B.5 The Bose-Einstein integral

Let

$$B_j(r) = \frac{1}{\Gamma(j+1)} \int_0^\infty \frac{y^j}{e^{y-r} - 1} dy, \quad j > -1, \quad r < 0$$

denote the complete Bose-Einstein integral of order j . It is known (Dingle, 1957) that $B_j(r) = \text{Li}_{j+1}(e^r)$, where Li_{j+1} denotes the polylogarithm of order $j+1$. The Bose-Einstein integral satisfies a recursive relationship

$$\frac{d}{dr} B_j(r) = B_{j-1}(r),$$

with a closed-form initial value $B_1(r) = -\log(1 - e^r)$.

Define

$$p(y | r) = h(y) \exp(yr - B_j(\beta r) \beta^{-(j+1)}),$$

where $h(y)$ is a nonnegative function and $\beta > 0$ is some fixed value. We would like to determine whether this choice of log partition function is valid.

Since the composition of the exponential function with a positive definite function is positive definite, it suffices to show that the Bose-Einstein integral $\text{Li}_{j+1}(e^{ir})$ is positive definite. This is observed by a comparison to the characteristic function of the Geometric distribution. We have that

$$\begin{aligned} \int_0^\infty \frac{y^j}{e^{y-ir} - 1} dy &= \int_0^\infty \frac{y^j e^{-y} e^{ir}}{1 - e^{-y} e^{ir}} dy \\ &= \int_2^1 \frac{(p-1)(-\log(p-1))^j e^{ir}}{1 - (p-1)e^{ir}} \frac{-1}{p-1} dp, \quad p = e^{-y} + 1 \\ &= \int_1^2 \left(\frac{(-1)^j \log(p-1)^j}{p} \right) \frac{pe^{ir}}{1 - (p-1)e^{ir}} dp. \end{aligned}$$

Now observe that $\left(\frac{(-1)^j \log(p-1)^j}{p} \right)$ is positive in $(1, 2)$, and $\frac{1}{1 - (p-1)e^{ir}}$ admits a Fourier series (via the geometric series) $\sum_{k=0}^\infty (p-1)^k e^{irk}$. Since the Fourier series coefficients are positive, $\frac{1}{1 - (p-1)e^{ir}}$ is positive definite, as is the product of positive definite functions $\frac{pe^{ir}}{1 - (p-1)e^{ir}}$. The integrand is therefore positive definite, and by Lévy's continuity theorem the integral is a positive definite function.

Since ϕ is 2π -periodic, it is the characteristic function of a discrete random variable taking integer values.

B.6 A non-example in the Fermi-Dirac integral

The Fermi-Dirac integral is closely related to the Bose-Einstein integral. Let

$$B_j(r) = \frac{1}{\Gamma(j+1)} \int_0^\infty \frac{y^j}{e^{y-r} + 1} dy, \quad j > -1, \quad r < 0$$

denote the complete Fermi-Einstein integral of order j . It is known (Dingle, 1957) that $B_j(r) = -\text{Li}_{j+1}(-e^r)$, where Li_{j+1} denotes the polylogarithm of order $j+1$. The Fermi-Dirac integral satisfies a recursive relationship

$$\frac{d}{dr} B_j(r) = B_{j-1}(r) \iff \frac{d}{dr} -\text{Li}_{j+1}(-e^r) = -\text{Li}_j(-e^r) \quad (17)$$

with a closed-form initial value $B_1(r) = \log(1 + e^r)$.

At first, this might appear to be an attractive way to construct softplus activations and their zero-temperature limits. If $B_2(r)$ were an admissible log partition function, then choosing R to be the identity, $\sigma(r) = A'(r) = \log(1 + e^r)$. However, the Fermi-Dirac integral is not necessarily positive definite. Following the same method as the Bose-Einstein integral, we have

$$\begin{aligned} \int_0^\infty \frac{y^j}{e^{y-ir} + 1} dy &= \int_0^\infty \frac{y^j e^{-y} e^{ir}}{1 + e^{-y} e^{ir}} dy \\ &= \int_2^1 \frac{(p-1)(-\log(p-1))^j e^{ir}}{1 + (p-1)e^{ir}} \frac{-1}{p-1} dp, \quad p = e^{-y} + 1 \\ &= \int_1^2 \left(\frac{(-1)^j \log(p-1)^j}{p} \right) \frac{pe^{ir}}{1 + (p-1)e^{ir}} dp. \end{aligned}$$

As before, observe that $\left(\frac{(-1)^j \log(p-1)^j}{p} \right)$ is positive in $(1, 2)$. However, $\frac{1}{1+(1-p)e^{ir}}$ admits a Fourier series (via the geometric series) $\sum_{k=0}^\infty (1-p)^k e^{irk}$. Since the Fourier series coefficients are **not always positive**, the Fermi-Dirac integral is indefinite. Hence we cannot conclude that its exponential is a positive definite function.

However, an alternative method of understanding softplus activations is given by (12). Choosing a Gaussian likelihood $A(v) = v^2/2$, a softplus activation σ implies that

$$\begin{aligned} R(\eta) &= \sqrt{\int 2 \log(1 + \exp(\eta)) d\eta} \\ &= \sqrt{-2\text{Li}_2(-e^\eta)}. \quad \text{by (17)} \end{aligned}$$

Roughly speaking, $\log(1 + \exp(\eta))$ looks like a smooth ReLU, so its integral $-\text{Li}_2(-e^\eta)$ looks like a squared smooth ReLU. The square root $R(\eta) = \sqrt{-2\text{Li}_2(-e^\eta)}$ looks like a smooth ReLU. The corresponding soft dropout function looks like a smooth step function, and is given by

$$\begin{aligned} \rho(\eta) &= \frac{d}{d\eta} \sqrt{-2\text{Li}_2(-e^\eta)} \\ &= \log(1 + e^\eta) \frac{1}{\sqrt{-2\text{Li}_2(-e^\eta)}}. \end{aligned}$$

B.7 Logistic sigmoid

It is well-known that the classical logistic sigmoid function $\sigma(\eta) = (1 + \exp(-\eta))^{-1}$ is the inverse-link function, or derivative of the log-partition function, of a Bernoulli exponential family with $A(r) = \log(1 + \exp r)$. Under such a model, the logistic sigmoid should be used to model binary-valued data, but it is commonly used in practice for real-valued data. We now provide an alternative method for constructing the logistic sigmoid via a non-linearly parameterised Gaussian, providing a principled derivation for its use in modelling real-valued data. Following the recipe (12), we have

$$\begin{aligned} R(\eta) &= \sqrt{\int 2(1 + \exp(-\eta))^{-1} d\eta} \\ &= \sqrt{2 \log(1 + \exp(\eta))} \quad \text{by (17)}. \end{aligned}$$

Roughly speaking, this looks like the square root of a smooth ReLU. The corresponding dropout function is small for negative inputs or large inputs,

$$\begin{aligned} \rho(\eta) &= \frac{d}{d\eta} \sqrt{2 \log(1 + \exp(\eta))^{-1}} \\ &= \sigma(\eta) \frac{1}{\sqrt{2 \log(1 + e^\eta)}}. \end{aligned}$$

B.8 Bregman divergences

In order to describe the effect that the nonlinearity R has on the Bregman divergence, we need to recall some machinery from Banerjee et al. (2005). In this subsection, without loss of generality, suppose T is the identity. We work with the

exponential family form

$$p(y | r) = h(y) \exp(yr - A(r)),$$

which is a probability density function for a scalar-valued random variable. Connections extend to the vector-valued case, but since the setting in our text is restricted to factorised (conditionally independent) scalar-valued densities, we write the connection in terms of scalar-valued objects.

Let ϕ be a strictly convex and differentiable function mapping from a convex subset of \mathbb{R} to \mathbb{R} . The Bregman divergence D_ϕ is defined as

$$D_\phi(v_1, v_2) = \phi(v_1) - \phi(v_2) - (v_1 - v_2) \frac{\partial}{\partial v_2} \phi(v_2).$$

Every Bregman divergence corresponds with exactly one exponential family through a convex conjugate. Concretely, the function ϕ that generates the Bregman divergence may be thought of as the convex conjugate of a log partition function A ,

$$\phi(\mu) = \sup_{r \in \mathbb{F}} \{\mu r - A(r)\}.$$

We have that (Banerjee et al., 2005, Theorem 4)

$$p(y | r) = \exp(-D_\phi(y, \mu)) b_\phi(y),$$

where b_ϕ is a uniquely determined function that does not depend on the expectation parameter μ (or the canonical parameter r).

The expectation parameter is related to the canonical parameter through $\mu = A'(r)$ and $r = \phi'(\mu)$. In our setting, since the canonical parameter is the result of a canonical map applied to a parameter η , we have $\mu = A' \circ R(\eta)$ and $R(\eta) = \phi'(\mu)$. If R is invertible, then $\eta = (R^{-1} \circ \phi')(\mu)$, otherwise there is more than one value of η such that $R(\eta) = \phi'(\mu)$. Finally, note that by definition, $\sigma(\eta) = (A \circ R)'(\eta) = A' \circ R(\eta) \times \rho(\eta)$, so that $\sigma(\eta)/\rho(\eta) = A' \circ R(\eta) = \mu$, whenever the operation $\sigma(\eta)/\rho(\eta)$ is defined.

As an example, we may obtain the setting of nonlinear least squares by choosing $\phi(v) = v^2$. In this case, A' is the identity, so that minimising the negative log likelihood is equivalent to minimising

$$D_\phi(y, A' \circ R(\eta)) = (y - R(\eta))^2.$$

More generally, our nonlinearly parameterised exponential family corresponds with a nonlinearly parameterised Bregman divergence $D_\phi(y, A' \circ R(\eta))$.

C The smooth ReLU

Generalised functions. Our calculus will involve limits of integrals against measures that converge to the singular Dirac delta measure. In particular, we make use of evaluation and certain derivative properties of the Dirac delta measure using the abuse of notation

$$\lim_{\tau \rightarrow 0} \int_{-\infty}^{\infty} g(x) \phi_{\tau}(x+a) dx = \int_{-\infty}^{\infty} g(x) \delta(x+a) dx = \frac{\partial}{\partial a} \int_{-\infty}^{\infty} g(x) \Theta(x+a) dx = g(-a)$$

for any continuous and compactly supported g , where ϕ_{τ} is any nascent delta function and Θ is the Heaviside step function.

This property will turn out to be useful because we may convert functions that are non-differentiable at a point into differentiable functions by introducing a well-motivated integral and limit. This enables to analyse activation functions such as the rectified linear unit which may be expressed as $\text{ReLU}(\eta) = \Theta(\eta)\eta$.

The ReLU_{τ} . Let $\text{ReLU}(\eta) = \Theta(\eta)\eta$ denote the ReLU, and $\text{ReLU}_{\tau}(\eta) = \int_{\mathbb{R}} \text{ReLU}(\eta + \epsilon) p_{\tau}(\epsilon) d\epsilon$ denote the τ -smoothed ReLU. Here p_{τ} is the PDF of a zero-mean Gaussian random variable with standard deviation $|\tau|$. We have an alternate representation using the expected value of the absolute value of a Gaussian random variable,

$$\begin{aligned} \text{ReLU}_{\tau}(\eta) &= \int \text{ReLU}(\eta + \epsilon) p_{\tau}(\epsilon) d\epsilon \\ &= \frac{1}{2} \left(\eta + \mathbb{E}[|\eta + \epsilon|] \right) \\ &= \eta \Phi\left(\frac{\eta}{|\tau|}\right) + \frac{|\tau|}{2} \sqrt{\frac{2}{\pi}} \exp(-\eta^2/2\tau^2), \end{aligned}$$

where Φ is the CDF of the standard normal distribution. The first term is also known as the GELU activation function, $\text{GELU}(\eta) = \eta \Phi\left(\frac{\eta}{|\tau|}\right)$. The GELU approaches the ReLU as $\tau \rightarrow 0$. The second term is a correction term that approaches 0 as $\tau \rightarrow 0$. Convergence is uniform, since by 1-Lipschitz of the ReLU,

$$\begin{aligned} |\text{ReLU}_{\tau}(\eta) - \text{ReLU}(\eta)| &= \left| \int (\text{ReLU}(\eta) - \text{ReLU}(\eta + \epsilon)) p_{\tau}(\epsilon) d\epsilon \right| \\ &\leq \int |\epsilon| p_{\tau}(\epsilon) d\epsilon \\ &= |\tau| \sqrt{\frac{2}{\pi}}, \end{aligned} \tag{18}$$

which decreases monotonically as $\tau \rightarrow 0$ for all η . As a convolution with a Gaussian, the ReLU_{τ} is infinitely differentiable. The first derivative is

$$\begin{aligned} \frac{\partial}{\partial \eta} \text{ReLU}_{\tau}(\eta) &= \frac{\partial}{\partial \eta} \int (\eta + \epsilon) \Theta(\eta + \epsilon) p_{\tau}(\epsilon) d\epsilon \\ &= \int \left(\Theta(\eta + \epsilon) + (\eta + \epsilon) \delta(\eta + \epsilon) \right) p_{\tau}(\epsilon) d\epsilon \\ &= \int \Theta(\eta + \epsilon) p_{\tau}(\epsilon) d\epsilon \\ &= \int_{-\eta}^{\infty} p_{\tau}(\epsilon) d\epsilon =: \Phi_{\tau}(\eta) \end{aligned}$$

which is the CDF of a zero-mean Gaussian random variable with standard deviation τ evaluated at η , i.e. $\Phi_{\tau}(\eta) = \mathbb{P}(\varepsilon \leq \eta)$, where $\varepsilon \sim \mathcal{N}(0, \tau^2)$. The second derivative is then the PDF of ε evaluated at η ,

$$\frac{\partial^2}{\partial \eta^2} \text{ReLU}_{\tau}(\eta) = \frac{1}{\sqrt{2\pi\tau^2}} \exp\left(-\frac{\eta^2}{2\tau^2}\right).$$

Leaky ReLU_{τ} . Let $m \in [0, 1]$ be some gradient parameter. We may define a Leaky ReLU_{τ} by

$$\text{LReLU}_{\tau}(\eta) = (1 - m) \text{ReLU}_{\tau}(\eta) + m\eta.$$

When $m = 0$, LReLU_{τ} is ReLU_{τ} and when $m = 1$, LReLU_{τ} is linear.

C.1 Rectified positive mean Gaussian

Let $A(r) = r^2/2$ and $R(\eta) = \text{ReLU}(\eta)$. This corresponds with an exponential family likelihood that always has a mean greater than or equal to zero. In order to visualise what this means, consider the case $d = l = 1$. With a $\mathcal{N}(0, \sqrt{2})$ prior over $W \in \mathbb{R}$ and $z \in \mathbb{R}$, the model evidence

$$p(y) = \mathbb{E}_{W, z \sim \mathcal{N}(0,1)} [p(y | \text{ReLU}(Wz))]]$$

is visualised by sampling 10000 W, z and plotting a KDE for y . This is shown in Figure 8, where the variance of the conditional distribution of y given Wz is λ^{-1} .

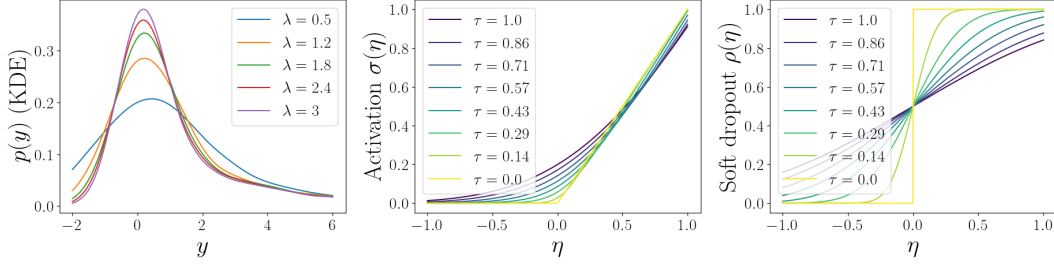


Figure 8: (Left) KDE of hard-rectified positive mean Gaussian marginal likelihood in 1 dimension with $\tau = 0.5$. (Middle) Activation function $\Phi_\tau(\eta)$ $\text{ReLU}_\tau(\eta)$ for different values of τ . (Right) Soft dropout function $\Phi_\tau(\eta)$ for different values of τ .

C.2 Rectified positive logit Bernoulli and binomial

Let $A(r) = t \log(1 + \exp(r))$ and $R(\eta) = \text{LReLU}_\tau(\eta)$. The parameter t is the number of trials for a binomial random variable, and $t = 1$ is the special case of a Bernoulli random variable. We may produce similar plots to Figure 8 by plotting histograms of the integrated binomial distribution. The case $t = 1$ is not interesting, since this is just a $\{0, 1\}$ -valued random variable and hence another Bernoulli. The case $t = 10$ is shown in Figure 9.

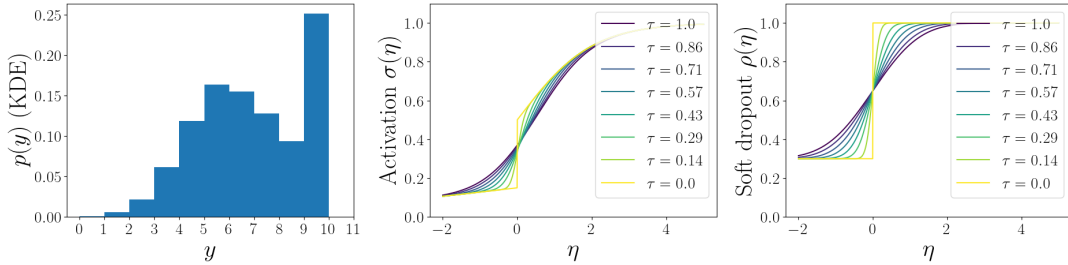


Figure 9: (Left) Histogram of samples from hard-rectified positive logit Binomial marginal likelihood in 1 dimension with $t = 10$ trials and $\tau = 0.5$. The leaky soft-rectified Bernoulli leads to interesting activation functions and dropout functions. Shown are the activation function (middle) and dropout function (right) when $m = 0.3$ for various values of τ . Recall that the activation function is $(A \circ R)'$, which is different to the expectation parameter $A' \circ R$.

In this case, the activation function is $\sigma(\eta) = t \frac{\exp(\text{LReLU}_\tau(\eta))}{1 + \exp(\text{LReLU}_\tau(\eta))} (\Phi_\tau(\eta)(1 - m) + m)$. The soft dropout function is $\rho(\eta) = \Phi_\tau(\eta)(1 - m) + m$

D Proofs

Theorem 1. *Suppose Assumption 1 holds. Let*

$$\begin{aligned} f(Z; Y_s, W, B) \\ = \frac{1}{\lambda} W^\top (T(Y_s) \odot \rho(WZ + B) - \sigma(WZ + B)). \end{aligned} \quad (6)$$

Any MAP estimate of (4) satisfies

$$Z_s^* = f(Z_s^*; Y_s, W, B), \quad \forall 1 \leq s \leq N, \quad (7)$$

solutions of which are guaranteed to exist and be unique. Any joint MAP estimate of (5) satisfies

$$\begin{aligned} W^*, B^* \in \operatorname{argmin}_{W, B \in \mathbb{W}} -\log p(W, B) - \log p(Z^*) + \\ \sum_{s=1}^N 1_{d \times 1}^\top A \circ R(WZ_s^* + B) - T(Y_s)^\top R(WZ_s^* + B) \\ \text{subject to } Z_s^* = f(Z_s^*; Y_s, W, B), \quad \forall 1 \leq s \leq N. \end{aligned} \quad (8)$$

Proof. The log posterior over Z and W, B is

$$\log p(Z, W, B | Y) = \log p(Y | W, Z, B) + \log p(Z) + \log p(W, B) - \log p(Y).$$

The joint MAP estimate (Z^*, W^*, B^*) solves the constrained optimisation problem

$$\begin{aligned} W^*, B^* = \operatorname{argmin}_{W, B \in \mathbb{W}} -\log p(Y | R(WZ^* + B)) - \log p(W, B) - \log p(Z^*) \\ \text{subject to } Z_s^* = \operatorname{argmin}_{Z_s} \frac{\lambda}{2} \|Z_s\|_2^2 - R(Z_s^\top W^\top + B^\top) T(Y_s) + 1_{d \times 1}^\top A \circ R(WZ_s + B). \end{aligned}$$

The outer problem is converted into a sum by replacing $\log p(Y | R(WZ^* + B))$ with the sum of likelihoods (1) evaluated at the datapoints $Y_s = (y_{si})_{i=1}^d$ and canonical parameters $R(H_s) = R(WZ_s^* + B)$.

We convert the inner DDN layer into a DEQ layer. We consider the inner problem for fixed s , which finds coefficients given W and b . The stationary points of the inner constraint satisfy

$$\begin{aligned} Z^* = \operatorname{argmin}_{z \in \mathbb{R}^l} \frac{\lambda}{2} \|z\|_2^2 - R(Z^\top W^\top + B^\top) T(Y) + 1_{d \times 1}^\top A \circ R(Wz + B) \\ 0 = \lambda Z^* - W^\top (T(Y) \odot \rho(Wz + B)) + W^\top \sigma(WZ^* + B) \end{aligned} \quad (19)$$

$$Z^* = \frac{1}{\lambda} W^\top (T(Y) \odot \rho(WZ^* + B) - \sigma(WZ^* + B)). \quad (20)$$

The Hessian H of the constraint is

$$H = \lambda I - W^\top \left(\operatorname{diag}(T(Y) \odot \rho'(WZ^* + B) - \sigma'(WZ^* + B)) \right) W$$

This is positive definite if the largest eigenvalue of $W^\top \left(\operatorname{diag}(T(Y) \odot \rho'(WZ^* + B) - \sigma'(WZ^* + B)) \right) W$ is less than λ . Let m denote the largest entry of $\operatorname{diag}(T(Y) \odot \rho'(WZ^* + B) - \sigma'(WZ^* + B))$.

Suppose m is nonnegative. The largest eigenvalue is less than λ if $\frac{m}{\lambda} \|W^\top W\|_2 < 1$. Since the objective is strongly convex and continuously differentiable, the solution to the fixed point equation is the unique global minimiser (Wright and Recht, 2022, Theorem 2.8). If m is negative, then the matrix clearly has only positive eigenvalues. \square

Theorem 2. *Suppose $R(\eta) = \operatorname{ReLU}(\eta)$ and fix some parameters (W, B) and data index s . Any stationary point Z_s^* of objective (4) is a solution to*

$$Z_s^* = \frac{1}{\lambda} W^\top (T(Y_s) \odot \Theta(WZ_s^* + B) - \operatorname{ReLU}(WZ_s^* + B)).$$

Under Assumption 2, there exists at least 1 and at most 2^d stationary points, all of which are local minima.

Proof. Any stationary point satisfies

$$\begin{aligned} 0 &= \frac{\partial}{\partial Z_s^*} \left(\frac{\lambda}{2} \|Z_s^*\|_2^2 - \text{ReLU}(Z_s^{*\top} W^\top + B^\top) T(Y_s) + 1_{d \times 1}^\top A \circ \text{ReLU}(WZ_s^* + B) \right) \\ 0 &= \lambda Z_s^* - W^\top T(Y_s) \odot \Theta(WZ_s^* + B) + W^\top \Theta(WZ_s^* + B) A' \circ \text{ReLU}(WZ_s^* + B) \\ Z_s^* &= \frac{1}{\lambda} W^\top (T(Y_s) \odot \Theta(WZ_s^* + B) - \sigma(WZ_s^* + B)), \end{aligned}$$

and also $WZ_s^* + B$ contains no elements that are 0. Since $\Theta(WZ_s^* + B) \in \{0, 1\}^d$, there are 2^d possible values of $\Theta(WZ_s^* + B)$. Noting that $\sigma(\eta) = \Theta(\eta) A' \circ \text{ReLU}(\eta)$, this implies that any Z_s^* must be a fixed point solution to one of the 2^d equations

$$Z_s^* = \frac{1}{\lambda} W^\top \left(P \odot (T(Y_s) - A' \circ R(WZ_s^* + B)) \right) \quad (21)$$

for some $P \in \{0, 1\}^d$. The right hand side is a contraction whenever $\frac{\alpha}{\lambda} \|W^\top W\| < 1$, since the composition of the two Lipschitz functions A' and R is also Lipschitz. This implies that there exists a unique solution Z_s^* to each of the 2^d possible fixed point equations. Some of these fixed point equations may not admit solutions such that $P = \Theta(WZ_s^* + B)$, so there are at most 2^d fixed points.

To see that there exists at least one fixed point, construct an iterated function system consisting of 2^d functions for each value of P . The attractor of such a system is nonempty. In particular for any starting point Z_s , we may apply a sequence of maps (21) with corresponding values of $P = \Theta(WZ_s + B)$, which converges to an element of the attractor (Hutchinson, 1981, Theorem §3.1).

Now observe that all stationary points are local minima. At the stationary point, where $WZ_s^* + B$ does not contain any zero elements, the Hessian is given by

$$\lambda I - W^\top \text{diag} \left(\Theta(WZ_s^* + B) A'' \circ R(WZ_s^* + B) \right) W.$$

This matrix is positive definite since $\frac{\alpha}{\lambda} \|W^\top W\|_2 < 1$. Therefore, the stationary points are local minima (Wright and Recht, 2022, Theorem 2.5). \square

E Deep PED

The joint maximum $(Z^{(1)*}, \dots, Z^{(L)*}, \theta^*)$ satisfies for each l

$$Z^{(l)*}, W^{(l)*}, B^{(l)} \in \operatorname{argmax}_{Z^{(l)} \in \mathbb{R}^{d_l}, W^{(l)}, B^{(l)} \in \mathbb{W}^{(l)}} p(Z^{(l-1)*} | Z^{(l)}, W^{(l)} B^{(l)}) p(Z^{(l)} | Z^{(l+1)*}, W^{(l+1)*}, B^{(l+1)*})$$

where we understand the second factor to mean $p(Z^{(L)})$ in the case $l = L$.

Stationary points. Here the conditional distribution of $Z^{(l)}$ given $Z^{(l+1)}$ takes the role of the prior from the shallow network case. It behaves like the unconditional Gaussian prior in the sense that its logarithm is quadratic, plus an extra term depending on $Z^{(l+1)*}$. At a stationary point, each $Z^{(l)*}$ must satisfy

$$\begin{aligned} 0 &= -W^{(l)\top} \left(\rho^{(l)}(W^{(l)} Z^{(l)} + B^{(l)}) \odot \sqrt{\lambda^{(l-1)}} Z^{(l-1)*} - \sigma^{(l)}(W^{(l)} Z^{(l)} + B^{(l)}) \right) + \\ &\quad \lambda^{(l)} Z^{(l)} - \sqrt{\lambda^{(l)}} R^{(l+1)} (W^{(l+1)} Z^{(l+1)*} + B^{(l+1)}) \\ Z^{(l)*} &= \frac{1}{\lambda^{(l)}} W^{(l)\top} \left(\rho^{(l)}(W^{(l)} Z^{(l)*} + B^{(l)}) \odot \sqrt{\lambda^{(l-1)}} Z^{(l-1)*} - \sigma^{(l)}(W^{(l)} Z^{(l)*} + B^{(l)}) \right) + \\ &\quad \frac{1}{\sqrt{\lambda^{(l)}}} R^{(l+1)} (W^{(l+1)} Z^{(l+1)*} + B^{(l+1)}), \end{aligned} \quad (22)$$

and additionally $W^{(l)} Z^{(l)*} + B^{(l)}$ must not contain any zero coordinates for any l .

Augmenting the state space. The fixed point solution for layer l depends on the solutions in layer $l-1$ and $l+1$, ignoring boundary cases. We may jointly compute the fixed points in a single DEQ layer by augmenting the $Z^{(l)}$ variables into a single state of size $D = \sum_{l=1}^L d^{(l)}$.

Let $\zeta = (Z^{(1)}; \dots; Z^{(L)}) \in \mathbb{R}^D$ and write

$$\begin{aligned} G^{(l)}(Z^{(l)}; Z^{(l-1)}, Z^{(l+1)}) &= \frac{1}{\sqrt{\lambda^{(l)}}} R^{(l+1)} (W^{(l+1)} Z^{(l+1)} + B^{(l+1)}) + \\ &\quad \frac{1}{\lambda^{(l)}} W^{(l)\top} \left(\rho^{(l)}(W^{(l)} Z^{(l)} + B^{(l)}) \odot \sqrt{\lambda^{(l-1)}} Z^{(l-1)} - \sigma^{(l)}(W^{(l)} Z^{(l)} + B^{(l)}) \right), \end{aligned}$$

and

$$G^{[1:L]}(\zeta) = \left(G^{(1)}(Z^{(1)}; Z^{(0)}, Z^{(2)}); \dots; G^{(L)}(Z^{(L)}; Z^{(L-1)}, 0) \right).$$

Counting the stationary points. If R is ReLU, following the same argument as the proof of Theorem 2, each $\rho^{(l)}$ maps to a finite set of size $2^{d^{(l)}}$. We may construct an iterated function system with $\prod_{l=1}^L 2^{d^{(l)}}$ elements, with components of the form

$$\begin{aligned} G_{P^{(l)}}^{(l)}(Z^{(l)}; Z^{(l-1)}, Z^{(l+1)}) &= \frac{1}{\sqrt{\lambda^{(l)}}} R^{(l+1)} (W^{(l+1)} Z^{(l+1)} + B^{(l+1)}) + \\ &\quad \frac{1}{\lambda^{(l)}} W^{(l)\top} \left(P^{(l)} \odot \sqrt{\lambda^{(l-1)}} Z^{(l-1)} - \sigma^{(l)}(W^{(l)} Z^{(l)} + B^{(l)}) \right), \end{aligned} \quad (23)$$

with a corresponding $G_{P^{[1:L]}}^{[1:L]}(\zeta)$. By the proof of Theorem 2, at least one 1 and at most $\prod_{l=1}^L 2^{d^{(l)}}$ stationary points of $G_{P^{[1:L]}}^{[1:L]}(\zeta)$ exist if all of the functions in the IFS are contractions. The Jacobian of $G_{P^{[1:L]}}^{[1:L]}(\zeta)$ is a tridiagonal block matrix. The diagonal block is

$$-\frac{1}{\lambda^{(l)}} W^{(l)\top} \operatorname{diag} \left(\sigma^{(l)'}(W^{(l)} Z^{(l)} + B^{(l)}) \right) W^{(l)}$$

and is always defined, since $W^{(l)}Z^{(l)*} + B^{(l)}$ must not contain any zero coordinates for any l . For $l < L$, the right off-diagonal block is

$$\frac{1}{\sqrt{\lambda^{(l)}}} \text{diag}\left(\rho^{(l+1)}(W^{(l+1)}Z^{(l+1)} + B^{(l+1)})\right)W^{(l+1)},$$

and is similarly always defined. For $l > 1$, the left off-diagonal block is

$$\frac{\sqrt{\lambda^{(l-1)}}}{\lambda^{(l)}} W^{(l)\top} \text{diag}\left(P^{(l)}\right),$$

and is similarly always defined. To show that this tridiagonal matrix is a contraction, split the matrix into a sum of a diagonal block matrix, a superdiagonal block matrix, and a subdiagonal matrix then use the subadditive property of matrix norms. The matrix norm of the diagonal matrix is bounded by $\max_l \frac{1}{\lambda^{(l)}} \|W^{(l)\top} W^{(l)}\|$. The matrix norm of the second and third matrices are respectively bounded by $\max_l \frac{1}{\sqrt{\lambda^{(l)}}} \|W^{(l)}\|$ and $\max_l \frac{\sqrt{\lambda^{(l-1)}}}{\lambda^{(l)}} \|W^{(l)\top}\|$.

If R is not ReLU, we may obtain a similar bound involving a factor analogous to Assumption 1.

Definiteness of Hessian. To reason about the nature of the stationary points, we must compute the eigenvalues of the Hessian. This Hessian is a block tridiagonal matrix. From the derivative of (23) with respect to $Z^{(l)}$, the diagonal block of the Hessian is

$$\lambda^{(l)} I - W^{(l)\top} \text{diag}\left(\sigma^{(l)'}(W^{(l)}Z^{(l)} + B^{(l)})\right)W^{(l)}$$

and is always defined, since $W^{(l)}Z^{(l)*} + B^{(l)}$ must not contain any zero coordinates for any l when R is the ReLU. For $l < L$, the right off-diagonal block is

$$\sqrt{\lambda^{(l)}} \text{diag}\left(\rho^{(l+1)}(W^{(l+1)}Z^{(l+1)} + B^{(l+1)})\right)W^{(l+1)},$$

and is similarly always defined. For $l > 1$, the left off-diagonal block is

$$\sqrt{\lambda^{(l-1)}} W^{(l)\top} \text{diag}\left(\rho^{(l)}(W^{(l)}Z^{(l)} + B^{(l)})\right),$$

and is similarly always defined.

Bounding the eigenvalues of this block tridiagonal matrix in a useful way remains an open challenge that we leave for future work.

F Convolutional layers

It is well-known that convolutional layers may be represented as sparse matrices with repeated entries. Our results apply when \mathbb{W} is a space of convolutional layers. Forming the equivalent sparse matrix for a convolutional layer is undesirable since it introduces an additional step and requires constraints. Therefore, it is of interest to discuss how to implement PED convolutional layers without forming the equivalent sparse fully connected layer.

The transpose operator can be computed using transposed convolution (Zeiler et al., 2010). In Pytorch, this operator may be implemented using `torch.nn.ConvTranspose2d()`.

The constraints in Assumption 1 and 2 may be computed efficiently when the norm is the spectral norm (Sedghi et al., 2019).

We leave practical investigation of convolutional layers for future work.

G Details of the illustrative example

PCA. We use the scikit learn implementation of PCA with default hyperparameters. We scale the input data into PCA with scikit learn’s StandardScaler.

UMAP. We use the publically available UMAP implementation (McInnes et al., 2018, BSD 3-Clause License) with default hyperparameters.

Data-generating process, shallow networks. We generate a 2D grid of size $10^5 \times 10^5$ with corners $(\pm 5, \pm 5)$. We then remove all points that are not inside the circle of radius 3 centered at $(2, 2)$, the circle of radius 2 centered at $(-3, -3)$ or the diamond of side length 1 centered at $(4, -4)$. These are the ground truth Z , consisting of 42453 points in the 2D plane. We choose an A and R , then repeat the following steps 100 times to obtain 100 random datasets. We generate $W^{(1)} \in \mathbb{R}^{50 \times 2}$ by drawing elements iid from a Gaussian with zero mean and variance $\frac{1}{2}$. We then sample Y from an exponential family with canonical parameter $R(WZ)$, log partition function A and sufficient statistic T .

Data-generating process, deep networks. The same as for shallow networks, but we generate $W^{(l)} \in \mathbb{R}^{d^{(l-1)} \times d^{(l)}}$ from $\mathcal{N}(0, 1/d^{(l)})$ for each l , with the widths $d^{(l)}$ as described in Figure 7 and $d^{(0)} = 50$. Each of the $Z^{(l)}$ are from an exponential family with canonical parameter $R(W^{(l+1)}Z^{(l+1)})$.

PED. We use PED with the same choice of A , T and R that matches the data generating process. This automatically defines the dropout functions (if any) and activation functions in the network. We use a value of $\lambda = 0.1$ whenever R is the identity and $\lambda = 1$ whenever R is ReLU. The reason for these different choices of λ is that PED layers admit a unique fixed point no matter the value of λ when R is the identity (see Theorem 1), so λ may be small so that the prior is weak. When R is ReLU, larger λ help to encourage PED layers to find local minima. We found it important to initialise weights with a small variance, especially for Poisson distributions, where the exponential inverse link function can either overflow or become very large. We use a zero mean iid Gaussian prior over W , B , which is implemented by applying weight decay to the neural network optimiser. We use Adam (Kingma and Ba, 2015) to optimise parameters W and B using default hyperparameters using a batch size of 500 and weight decay varying with layer (note this is not the same as using weight decay with AdamW, which would not be equivalent to L2 regularisation or Gaussian prior regularisation (Loshchilov and Hutter, 2019)). Weight decay is set to $10L \times d^{(l-1)}$ in layer l . For shallow models, we train the network for 30 epochs, whereas for deep models, we train the network for 10 epochs, freezing the parameters in layer l for the first $5l$ epochs. We use a publically available DEQ repository (Bai et al., 2019, MIT License) to implement our DEQs, with a Anderson acceleration method used as the solver with default hyperparameters.

Downstream task. We use a small head network Linear(100) - γ ReLU - γ Linear(1) using default initialisation in Pytorch. The headnet is appended to our pretrained backbone of each of PCA, PED and UMAP and fine-tuned using Adam with default hyperparameters. For PCA and UMAP, which do not contain learnable parameters, the backbone network is fixed. The full network is trained to minimise the sum of squared errors between the output of the network and the function $g(z) = Z_1 + Z_2$, where Z_1 and Z_2 are respectively the first and second coordinate of z . We use a batch size of 500 and train the network for 200 epochs. In order to evaluate our model on the downstream task, we turn dropout off. We randomly partition each of the datasets into 80% – 20% training-testing split and evaluate our network on the test set. We report the number of times the network performed the best out of the 100 random runs for each of PCA, UMAP and PED. In the case of Deep PED, to allow a fair comparison to other techniques, we only use the bottleneck neurons as input to the head network.

Hardware, software and computational cost. We parallelised runs over multiple nodes, each consisting of a single Tesla P100-SXM2-16GB GPU. We used CUDA 11.4 and Pytorch 1.11.0. Only PED is GPU-accelerated. We found that typical run times for the dimensionality reduction task were task dependent. Table 2 shows typical run times for each task.

Interpretable neurons. One interesting feature of deep PED in the case $L > 1$ is that certain collections of neurons have interpretations of latent variables at different layers. For example, if $L = 2$ and $d^{(1)} = 30$, $d^{(2)} = 2$, $D = d^{(1)} + d^{(2)} = 32$, we obtain deep PED which may be implemented as a single layer of the form (10). Even though this DEQ layer has 32 neurons, we may pick out the 2 neurons which represent the latent variables in last layer. For this reason, deep PED might be considered interpretable. We are not aware of any other DEQ layers that possess a similar quality.

| | PCA | tSNE | UMAP | PED |
|--------------|------|--------|--------|---------|
| Gaussian | 0.09 | 901.94 | 180.46 | 294.09 |
| Bernoulli | 0.11 | 923.85 | 174.71 | 267.81 |
| Poisson | 0.10 | 698.34 | 176.33 | 189.87 |
| ReLU $L = 1$ | 0.10 | 622.25 | 168.11 | 1101.38 |
| ReLU $L = 2$ | 0.10 | 813.25 | 159.71 | 356.40 |

Table 2: Measured run times (seconds) for each dataset and model.

H Approximation to the posterior

In shallow PED, the Hessian of the Laplace approximation is found by taking the derivative of the right hand side of (19) with respect to Z^* . We find

$$H(Z^*) := -\log p(Z^* | Y, W, B) = \lambda I - W^\top \text{diag}(T(Y) \odot \rho'(WZ^* + B) - \sigma'(WZ^* + B))W.$$

This leads to the Laplace approximation to the posterior,

$$q(z | Y, W, B) = \mathcal{N}\left(z | Z^*, H(Z^*)^{-1}\right).$$

Note that the Hessian is positive definite at the local minima that we found, so that inversion of the Hessian is well-defined.

In deep PED, one may work with the expressions found in Appendix E, which lead to a Hessian with sparse block structure. In many applications, only the square block at the final layer is of interest, which leads to

$$H(Z^{(L)*}) := \lambda I - W^{(L)\top} \text{diag}(T(Y) \odot \rho'(W^{(L)}Z^{(L)*} + B) - \sigma'(W^{(L)}Z^{(L)*}))W^{(L)}$$

$$q(Z^{(L)} | Y, \theta) = \mathcal{N}\left(z | Z^*, H(Z^*)^{-1}\right).$$

If the likelihood were linearly parameterised Gaussian, σ' would be 1, ρ' would be zero and we would recover the well-known Gaussian case,

$$H(Z^{(L)*}) = \lambda I + W^{(L)\top}W^{(L)},$$

which is independent of the MAP $Z^{(L)*}$.

If the likelihood were nonlinearly parameterised Gaussian with $R = \text{ReLU}$, we observe an interesting dropout effect. Since at the stationary point, ρ' and σ' are well-defined (by definition) and are 0 and Θ respectively, we find

$$H(Z^{(L)*}) = \lambda I + W^{(L)\top} \text{diag}(\Theta(W^{(L)}Z^{(L)*} + B))W^{(L)}.$$

Interestingly, this Hessian only depends on $Z^{(L)*}$ through $\Theta(W^{(L)}Z^{(L)*} + B)$.