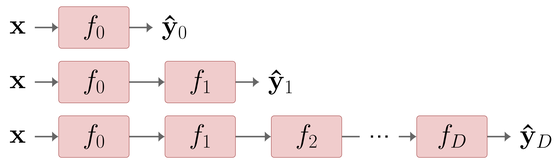
Variational Depth Search in ResNets

Javier Antorán, James Urquhart Allingham, José Miguel Hernández-Lobato
{ja666, jua23, jmh233}@cam.ac.uk

UNIVERSITY OF CAMBRIDGE

GitHub

How Deep Should Networks Be?



- In NNs, computational cost scales linearly in depth.
- Want to trade-off model flexibility and cost.

Learnt Depth Networks (LDNs)

We exploit sequential nature of NNs for efficient one-shot NAS using the architecture in Figure 1:

Residual Blocks f_i with Binary Gating:

$$\mathbf{a}_i = \mathbf{a}_{i-1} + b_i f_i(\mathbf{a}_{i-1})$$

Can Limit Model to Depth of d:

$$b_i = 1 \,\forall\, i \leq d; \quad b_i = 0 \,\forall\, i > d$$

Evaluate All Depths in Single Forward Pass:

$$\hat{\mathbf{y}}_i = f_{D+1}(\mathbf{a}_i)$$

Inference in LDNs

Define Likelihood and Categorical Prior:

$$p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x}, d); \quad p(d) = Cat(d)$$

Tractable Categorical Posterior:

$$p(d{=}j|\mathcal{D}) = \frac{p(d{=}j) \cdot \Pi_{n=1}^{N} p(\mathbf{y}^{(n)}|\mathbf{x}^{(n)}, d{=}j)}{\Sigma_{i=0}^{D} p(d{=}i) \cdot \Pi_{n=1}^{N} p(\mathbf{y}^{(n)}|\mathbf{x}^{(n)}, d{=}i)}$$

Repeatedly computing the posterior by iterating over $\mathcal{D}$ is expensive. We learn an approximate distribution over depth $q(d)$ and model weights $\boldsymbol{\theta}$ simultaneously using **Variational Inference**:

$$\text{ELBO}(q, \boldsymbol{\theta}) = \Sigma_{n=1}^{N} \mathbb{E}_{q(d)} \left[\log p_{\boldsymbol{\theta}}(\mathbf{y}^{(n)}|\mathbf{x}^{(n)}, d)\right] - \text{KL}(q(d) \,\|\, p(d))$$

Minibatch estimator of ELBO is evaluated in closed form with single forward pass. We **choose** $\mathbf{d}$ as:

$$d_{opt} = \min_i \{i : q(d{=}i) \geq 0.95 \max_j q(d{=}j)\}$$

Predictions are made through **Marginalisation**:

$$p(\mathbf{y}^*|\mathbf{x}^*) \approx \sum_{i=0}^{d_{\text{opt}}} p_{\boldsymbol{\theta}}(\mathbf{y}^*|\mathbf{x}^*, d{=}i) q(d{=}i)$$

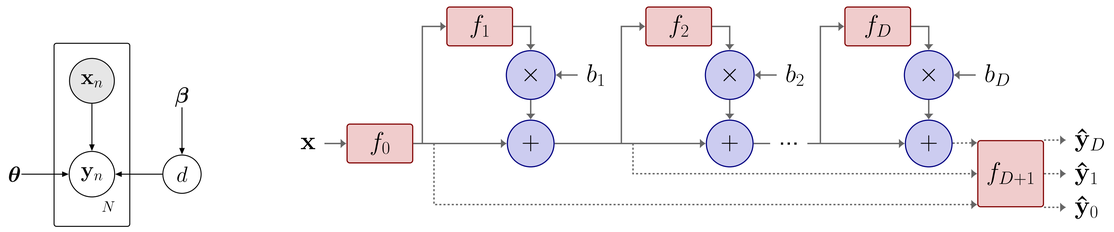


Figure 1: Left: graphical model. Right: computational model. Each block's activation is passed through the output block.

Learnt Depth Distributions

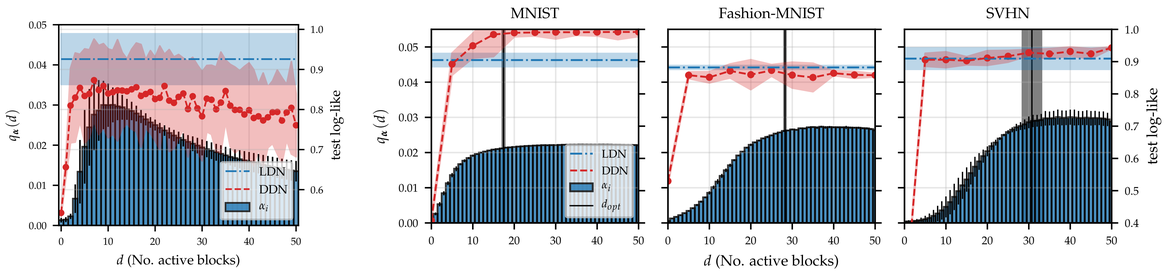


Figure 2: Left: Spirals. Right: Image datasets. Histograms show learnt distributions over depth. Vertical black lines indicate chosen depths. LDNs using up to d_{opt} layers obtain test set log-likelihoods (blue lines) similar to sufficiently deep regular NNs (red lines).

Making an Efficient use of Layers

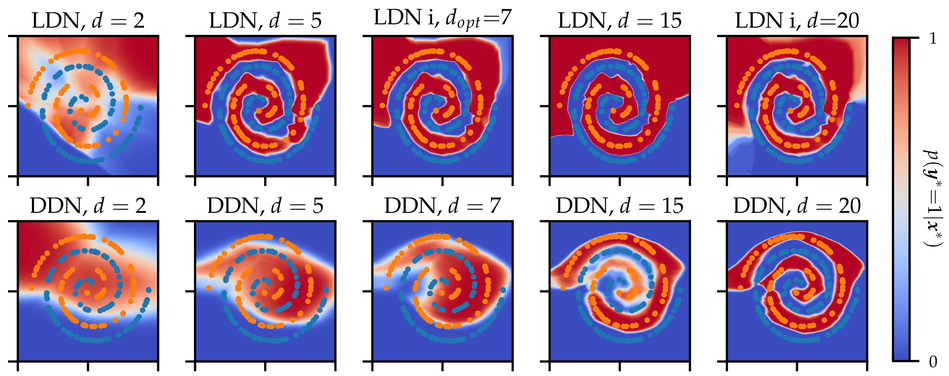


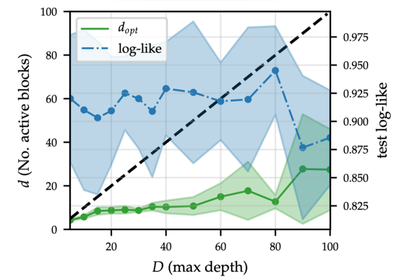
Figure 3: Top: spiral functions learnt at different depths of an LDN. Bottom: functions learnt at different depths of a regular network (DDN). In all cases the max possible depth is 20. LDNs require less layers for same problems, enabling pruning.

Better Calibrated Predictions, For Free

Table 1: Expected Calibration Errors obtained by regular NNs (DDNs), pruned LDNs and unpruned LDNs on image datasets.

	DDN	LDN, $d \in [0, d_{opt}]$	LDN, $d \in [0, D]$
SVHN	9.59 ± 0.065	$\mathbf{9.48 \pm 0.011}$	$\mathbf{9.47 \pm 0.002}$
Fashion-MNIST	10.17 ± 0.121	$\mathbf{9.73 \pm 0.021}$	$\mathbf{9.71 \pm 0.020}$
MNIST	$\mathbf{9.06 \pm 0.004}$	9.50 ± 0.230	9.49 ± 0.207

Consistent Depth Predictions



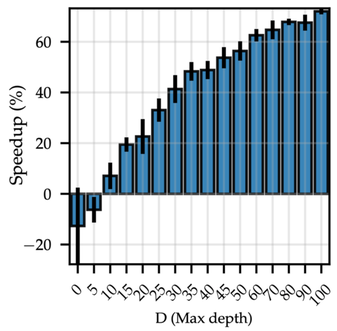


Figure 4: Top: The learnt depth and test log-likelihood remain mostly invariant to the maximum allowed depth on Spirals. Bottom: Inference time reduction provided by LDNs with respect to regular NNs of increasing depth on MNIST.

Discussion

We formulate a variational objective over ResNet depth which can be evaluated exactly. It allows for one-shot learning of both model weights and a distribution over depth. We leverage this distribution to prune our networks, making test-time inference cheaper, and to obtain model uncertainty estimates.

Our training procedure encourages an efficient use of model capacity, making models amenable to pruning. Pruned networks perform competitively with regular ones of any depth on a toy spiral dataset, MNIST, Fashion-MNIST and SVHN. They often provide better calibrated uncertainty estimates.

Acknowledgments

JA acknowledges support from Microsoft Research, through its PhD Scholarship Programme, and from the EPSRC. JUA acknowledges funding from the EPSRC and the Michael E. Fisher Studentship in Machine Learning.