# Adapting Branched Neural Networks to Realise Progressive Intelligence

MINDS·CDT

University of Southampton

Scan for paper


Jack Dymond*, Sebastian Stein, Steve R. Gunn

*jd5u19@soton.ac.uk

Electronics and Computer Science

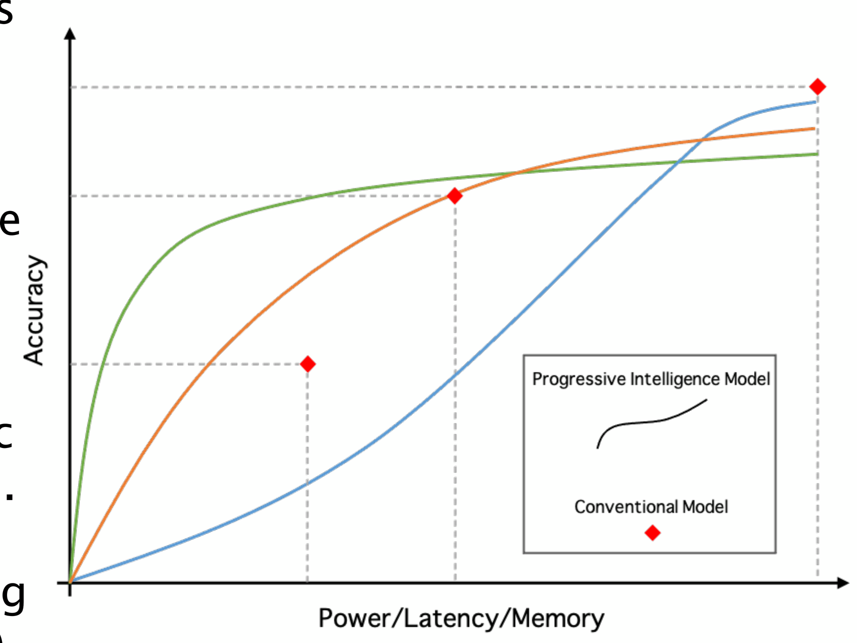
University of Southampton


UK Research and Innovation

## Overview

Progressive intelligence is a form of machine learning which approaches the inference process incrementally, obtaining low power, lower confidence results first, and then using more power to improve its prediction accordingly. We explore how branched networks can be used to achieve this.

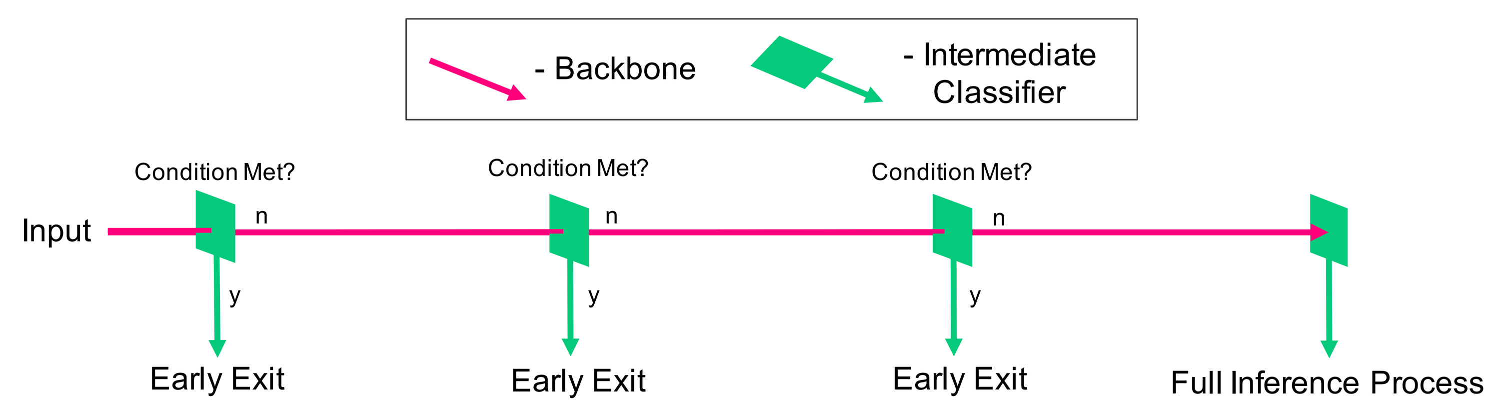
- It is represented as continuous distributions in performance—cost spaces.
- Conventional models are represented as discrete points.
- Better suited to dynamic operating environments. (eg. search and rescue drones, cloud computing servers, mobile devices)




## Realising Progressive Intelligence

To produce a progressive intelligence system we explore branched neural networks as a good starting point. These networks process the input incrementally implicitly in their design, and can exit inference early depending on their confidence.

### Branched Neural Networks




- Branched networks are made up of two main components:
  - The neural network backbone.
  - The early exit enabling intermediate classifiers.
- Entropy used as a measure of uncertainty. If this is below a certain value an early exit can occur.

$$e(x) = -\sum_{i=1}^{C} p(x_i)\log p(x_i)$$

- We introduce a new exit policy called mutual agreement:
  - Network will also exit if concurrent branches agree with one-another.
- Weighted loss is used to train network.

$$L_{total}(\hat{y}, y; \theta) = \sum_{n}^{N} w_n L_n(\hat{y_{exit}}, y; \theta)$$

### Experimental Procedure

For our experiments we:

- Vary the branch weighting during training to understand the representational changes of progressive intelligence.
- Change the confidence requirement of the exit to produce a cost—performance distribution, we refer to this as the operating range of the network.
- Experiment with this by changing the backbone of the branched network and its exit policy.

To quantify representational performance we use two metrics:

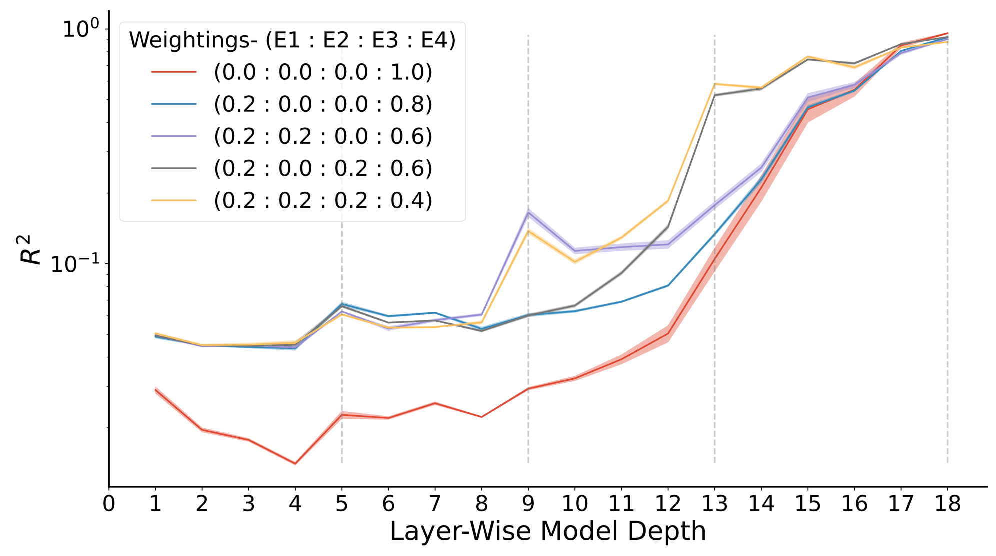
- Class separation using the metric: $R^2$

$$R^2 = 1 - \frac{\sum_{k=1}^{K}\sum_{m=1}^{N_k}\sum_{n=1}^{N_k}(1-\mathrm{sim}(\boldsymbol{X}_{k,m},\boldsymbol{X}_{k,n}))/(KN_k^2)}{\sum_{j=1}^{K}\sum_{k=1}^{K}\sum_{m=1}^{N_j}\sum_{n=1}^{N_k}(1-\mathrm{sim}(\boldsymbol{X}_{j,m},\boldsymbol{X}_{k,n}))/(KN_jN_k)}$$

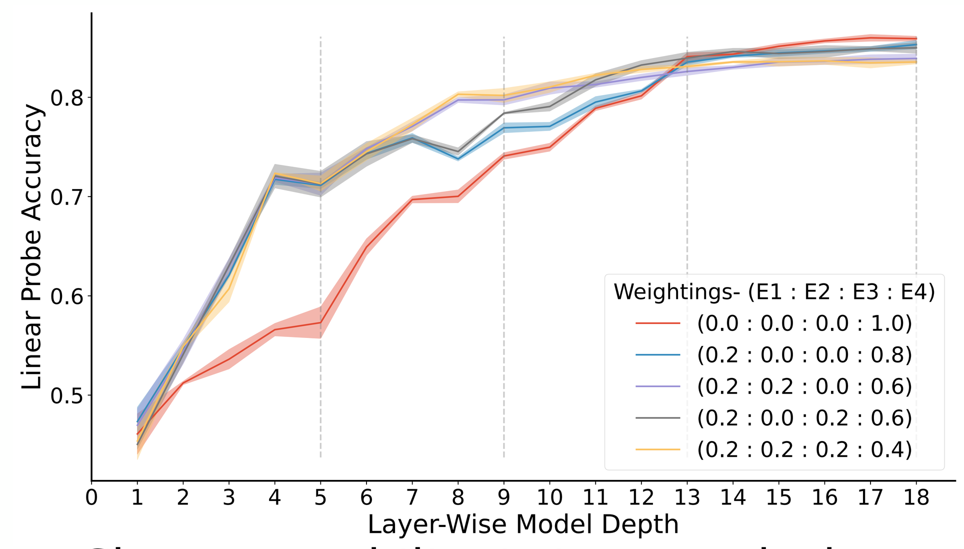
  - Where sim refers to the cosine similarity, $K$ the number of classes, and $X$ the representations of a particular input.
  - Hence, this measures the ratio of the similarity within a class to that across all classes.
- Linear probe accuracy
  - This is the accuracy a linear classifier can achieve on the representations $X$ produced by the model.
  - Measures the class separability of the model.

## Results

All results shown are taken from experiments on the CIFAR10 dataset unless otherwise stated. The figures below illustrate the effect of the loss function when training neural networks. They analyse the effect of branch positioning as well as branch weighting. Dashed vertical lines refer to the branch positions in the networks.



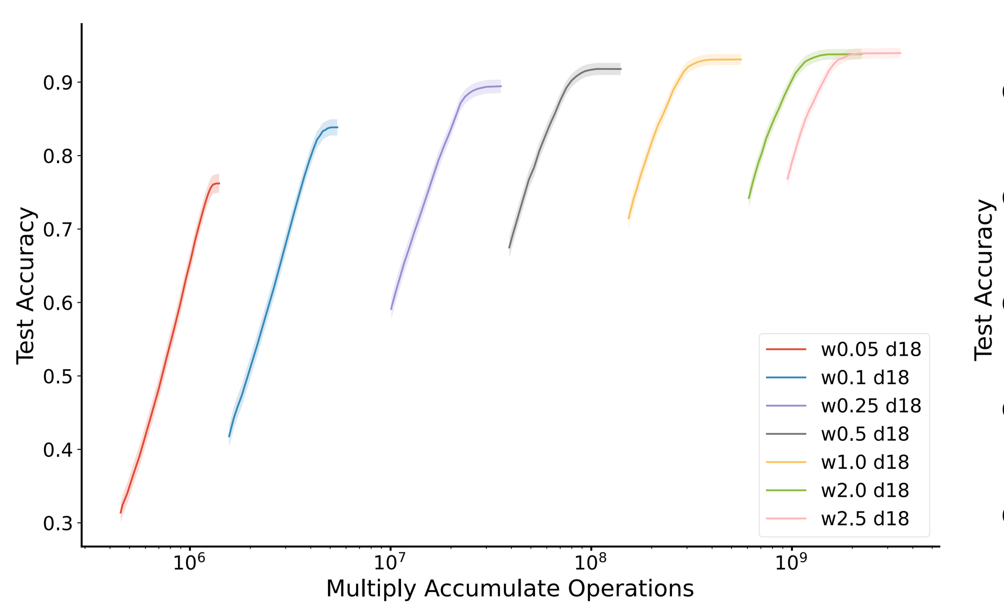

Weighting earlier branches increases class separation, $R^2$, at the point of the branch connection.



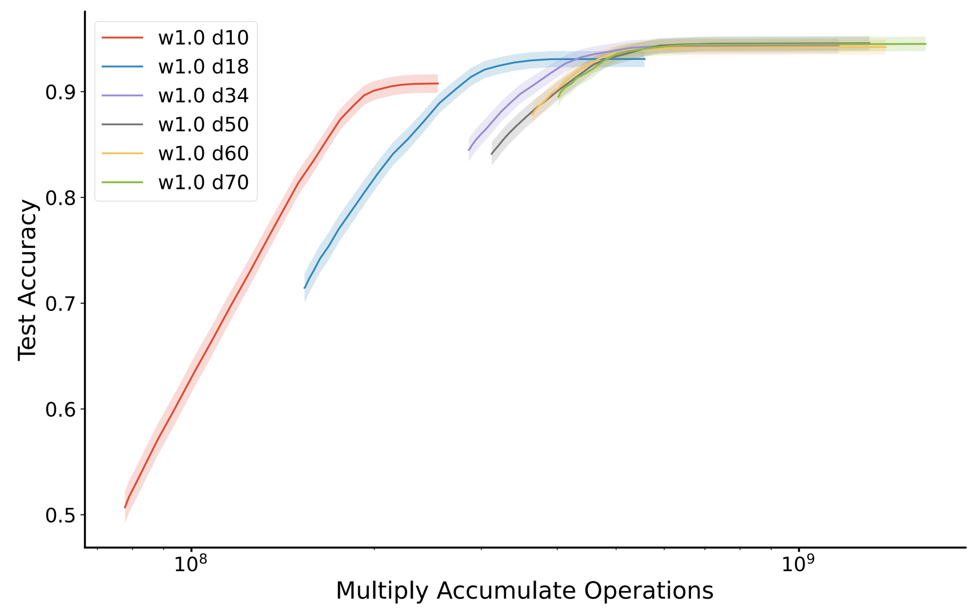

Class separability is increased when weighting earlier branches.

- There is a trade-off between class separation and linear probe accuracy, possibly due to the learning process restricting class-separation to improve later representations.
- Branches allow for the increase in class separation without drastically reducing classification accuracy.

The figures below denote the operating range of a network. This is the distribution in accuracy—MAC operations a network can operate at.




Varied width results show networks can be reduced to a quarter of the width without significantly degrading the accuracy range of a network.

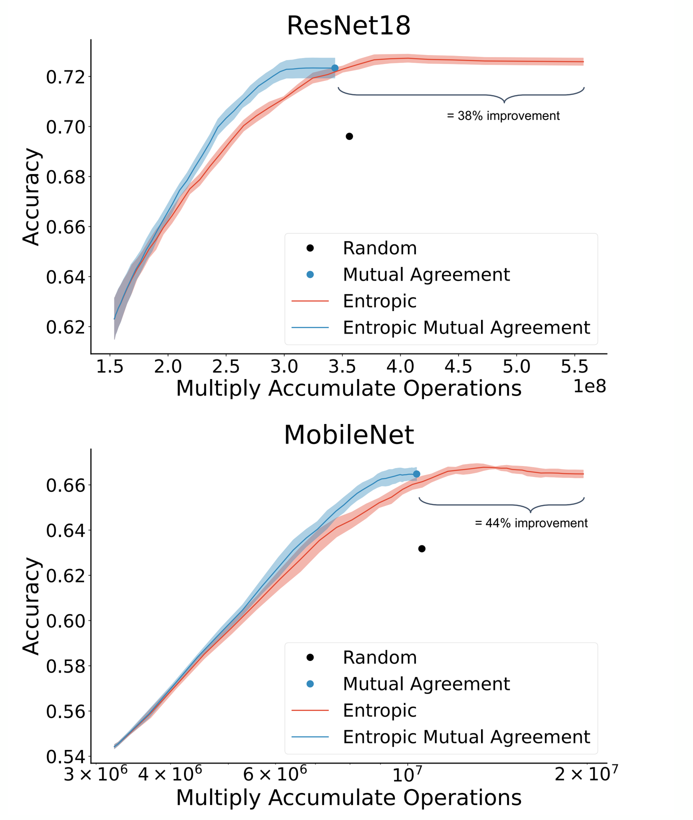



Varying the depth has less impact on the operating range.

- Varying the width of the branched neural network has a much greater effect in shifting the operating range of the system, more so than the early exiting itself.
- Neither strategy is able to improve the accuracy the model is capable of achieving on the dataset used.

| $w$ - (Exit 1 : Exit 2 : Exit 3 : Final Exit) | AUC | Raw ↑ (%) | Frac. ↑ (%) |
|---|---|---|---|
| Baseline: (0.0 : 0.0 : 0.0 : 1.0) | 0.6878 | - | - |
| **(0.2 : 0.0 : 0.0 : 0.8)** | **0.7276** | **3.98** | **5.78** |
| (0.4 : 0.0 : 0.0 : 0.6) | 0.7266 | 3.88 | 5.64 |
| (0.6 : 0.0 : 0.0 : 0.4) | 0.7217 | 3.39 | 4.93 |
| (0.8 : 0.0 : 0.0 : 0.2) | 0.7262 | 3.84 | 5.58 |
| (0.0 : 0.2 : 0.0 : 0.8) | 0.7096 | 2.18 | 3.16 |
| (0.0 : 0.4 : 0.0 : 0.6) | 0.7093 | 2.15 | 3.13 |
| (0.0 : 0.6 : 0.0 : 0.4) | 0.7099 | 2.21 | 3.21 |
| (0.0 : 0.8 : 0.0 : 0.2) | 0.7078 | 2.0 | 2.91 |
| (0.0 : 0.0 : 0.2 : 0.8) | 0.6926 | 0.48 | 0.69 |
| (0.0 : 0.0 : 0.4 : 0.6) | 0.6932 | 0.54 | 0.78 |
| (0.0 : 0.0 : 0.6 : 0.4) | 0.6919 | 0.41 | 0.6 |
| (0.0 : 0.0 : 0.8 : 0.2) | 0.6937 | 0.59 | 0.86 |
| (0.2 : 0.3 : 0.1 : 0.4) | 0.7308 | 4.3 | 6.26 |
| **(0.2 : 0.2 : 0.2 : 0.4)** | **0.7345** | **4.67** | **6.79** |
| (0.2 : 0.1 : 0.3 : 0.4) | 0.7317 | 4.39 | 6.39 |

Area under curve (AUC) measurements of linear probe plots are presented. Equally weighting the branches maximises AUC performance.




New exit policy improves upon conventional exit policies. Results taken from CIFAR100 experiments.

## Conclusions

- Branched networks can be adapted to show progressive intelligence.
- Weighted loss affects class separation and linear probe accuracy throughout model.
- Equal weighting maximises linear probe accuracy throughout model.
- Scaling confidence thresholds allows for a progressive inference range.
- Width shifts this range.
- Depth extends this range.
- New exit policy proposed: Mutual agreement.
- Improves upon conventional methods saving up to 44% of inference cost.


### ACKNOWLEDGEMENTS

This work was supported and funded by: The UK Research and Innovation (UKRI) Centre for Doctoral Training in Machine Intelligence for Nano-electronic Devices and Systems [EP/S024298/1]; the UKRI Turing AI Acceleration Fellowship on Citizen-Centric AI Systems [EP/V022067/1]; the Defence Science and Technology Laboratory (DSTL).


[dstl]


BMVC 2022

The 33rd British Machine Vision Conference

21st - 24th November 2022, London, UK