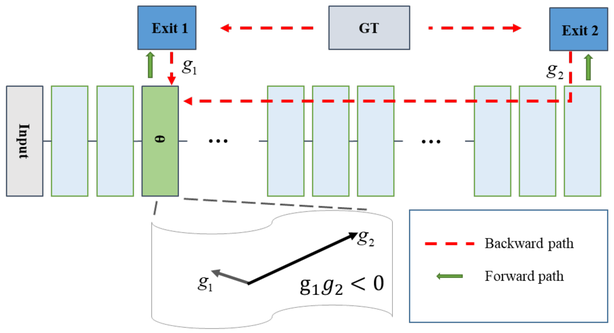
# Meta-GF: Training Dynamic-Depth Neural Networks Harmoniously

Yi Sun, Jian Li, and Xin Xu
The College of Intelligence Science and Technology, National University of Defense Technology, Changsha 410000, China
{sunyi13, lijian, xinxu}@nudt.edu.cn

## The gradient conflict when training multi-exit neural networks




As the frequently used dynamic-depth neural networks, the multi-exit networks achieve adaptive inference by attaching multiple output exits at different depths of the networks, and taking early-exit policy conditioning on the input. Unfortunately, these exits usually interfere with each other in the training stage. The interference would reduce performance of the models and cause negative influences on the convergence speed.

## Meta-GF: weighted gradient fusion policy by meta-learning

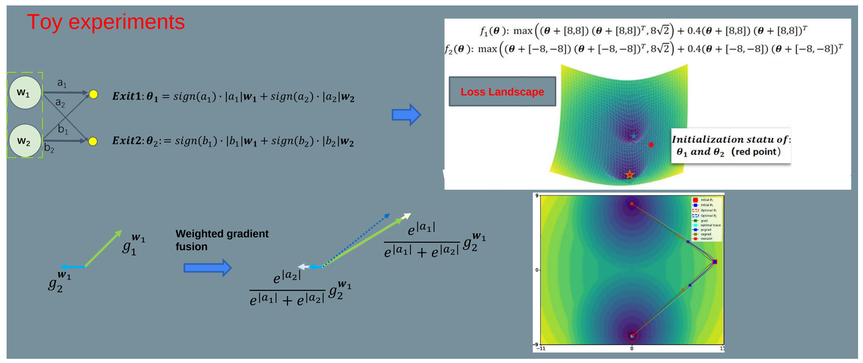
The proposed Meta-GF takes a meta-learned weighted fusion policy to combine the gradients of each exit, which takes account of the importance of the shared parameters for different exits:

$$g_f^{\mathbf{W}} = \frac{\sum_{i=1}^{n} e^{\eta_i^{\mathbf{W}}} g_i^{\mathbf{W}}}{\sum_{i=1}^{n} e^{\eta_i^{\mathbf{W}}}}.$$

Denoting the trainable importance values of $\boldsymbol{w}$ to each exit is:$\{\eta_i^{\boldsymbol{w}}, i \in [1,n]\}$ The final gradient $g_f^{\boldsymbol{w}}$ of parameter $\boldsymbol{w}$ is obtained by weighted sum.

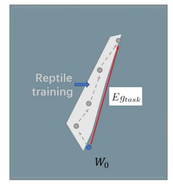
The $\{\eta_i^{\boldsymbol{w}}, i \in [1,n]\}$ is optimized by minimizing the joint task loss $F$(e.g. the cross-entropy loss)

$$\eta = \arg\min_{\eta} F(\mathbf{W} - \epsilon \frac{\sum_{i=1}^{n} e^{\eta_i} g_i}{\sum_{i=1}^{n} e^{\eta_i}}).$$




We use the expected gradient $\{Eg_i^{\boldsymbol{W}}\}$ of each task to replace the noisy mini-batch gradient $\{g_i^{\boldsymbol{W}}\}$, which is obtained by training the task independently for one epoch. These independent training process share the same initial model $\boldsymbol{w_0}$:

$$\mathbf{W}_{task} = \arg\min_{\mathbf{W}} F_{task}(\mathbf{W}; \mathbf{W}_0, D),$$
$$Eg_{task} = \mathbf{W}_{task} - \mathbf{W}_0,$$




**Algorithm 1** Meta Weighted Gradient Fusion:

**Input:** Initial parameters:$\mathbf{W}_0$, training dataset:$D$, learning rate:$\epsilon$, fusion weight:$\eta$. The number of exits is $n$. $F = \{F_1, ..., F_n\}$ is the objective function of the exits.
**Output:** W

```
while i < MaxIter do
    ▼ 1. Calculating expected gradients:
    for j=1,...,n do
        Ŵ = W_0
        W_j = arg min_W F_j(W; Ŵ, D)
        g_j = W_j − Ŵ,
    end for
    ▼ 2. Meta Weighted Gradient Fusion:
    η = arg min_η F(W_0 − ε Σ_{i=1}^n e^{η_i} g_i / Σ_{i=1}^n e^{η_i}; D).
    W = W_0 − ε Σ_{i=1}^n e^{η_i} g_i / Σ_{i=1}^n e^{η_i}
    W_0 = W
end while
return W
```


## Experiments

### ➢ Performance of each exit

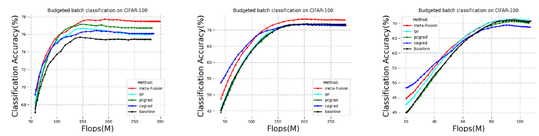
| | Params(M) | flops(M) | CIFAR100 | | | | | CIFAR10 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | MSDnet | GE | Cagrad | Pcgrad | ours | MSDnet | GE | Cagrad | Pcgrad | ours |
| Exit-1 | 0.90 | 56.43 | 66.41 | 67.74 | **68.78** | 67.06 | 67.97 | 91.13 | 92.02 | 92.19 | 91.66 | **92.38** |
| Exit-2 | 1.84 | 101.00 | 70.48 | 71.87 | **72.55** | 71.37 | 72.27 | 92.91 | 93.53 | 93.49 | 93.59 | **94.22** |
| Exit-3 | 2.80 | 155.31 | 73.25 | 73.81 | 74.23 | 74.86 | **75.06** | 93.98 | 94.14 | 94.47 | 94.32 | **94.49** |
| Exit-4 | 3.76 | 198.10 | 74.02 | 75.13 | 74.97 | **75.78** | 75.77 | 94.46 | 94.49 | 94.45 | 94.60 | **94.96** |
| Exit-5 | 4.92 | 249.53 | 74.87 | 75.86 | 75.35 | 76.25 | **76.38** | 94.68 | 94.73 | 94.48 | 94.81 | **94.82** |
| Exit-6 | 6.10 | 298.05 | 75.33 | 76.23 | 75.82 | 76.95 | **77.11** | 94.78 | 94.89 | 94.53 | 94.83 | **94.97** |
| Exit-7 | 7.36 | 340.64 | 75.42 | 75.98 | 76.08 | 76.71 | **77.47** | 94.64 | 94.96 | 94.48 | 94.82 | **94.97** |
| Average | - | - | 72.83 | 73.80 | 73.96 | 74.14 | **74.57** | 93.80 | 94.11 | 94.01 | 94.09 | **94.54** |

**Table 1.** Classification accuracy of individual classifiers in multi-exit MSDnet on CIFAR-100 and CIFAR10.

| | Params(M) | flops(M) | CIFAR100 | | | | | CIFAR10 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | SDN-vgg | GE | Cagrad | Pcgrad | ours | SDN-vgg | GE | Cagrad | Pcgrad | ours |
| Exit-1 | 0.05 | 39.76 | 44.42 | 44.46 | **53.08** | 43.59 | 49.91 | 69.03 | 68.97 | **76.27** | 67.41 | 74.92 |
| Exit-2 | 0.29 | 96.52 | 61.08 | 61.0 | 61.39 | **63.02** | 61.09 | 84.72 | 84.52 | 86.3 | 85.28 | **88.69** |
| Exit-3 | 1.22 | 153.25 | 69.8 | 69.54 | 70.9 | 70.04 | **71.38** | 92.15 | 92.02 | 92.4 | 91.8 | **92.75** |
| Exit-4 | 1.85 | 191.08 | 72.23 | 72.11 | 71.55 | 73.14 | **75.77** | 92.5 | 92.62 | 92.79 | 92.74 | **93.07** |
| Exit-5 | 5.47 | 247.81 | 72.48 | 72.32 | 72.41 | 72.59 | **74.12** | 92.46 | 92.78 | 92.99 | 92.75 | **93.13** |
| Exit-6 | 7.86 | 285.68 | 72.63 | 72.38 | 72.45 | 72.54 | **74.23** | **93.59** | 92.83 | 93.07 | 92.7 | 93.12 |
| Exit-7 | 15.47 | 314.45 | 71.76 | 71.58 | 71.43 | 71.39 | **73.1** | 92.61 | 92.85 | 93.0 | **93.69** | 93.07 |
| Average | - | - | 66.34 | 66.19 | 67.60 | 66.61 | **68.51** | 88.15 | 88.08 | 89.54 | 88.05 | **89.82** |

**Table 3.** Classification accuracy of individual classifiers in multi-exit Vgg-SDN[26] on CIFAR-100 and CIFAR10.

### ➢ Budgeted batch classification

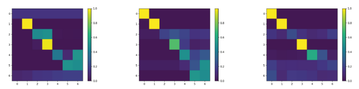



(a) MSDnet (b) Vgg-SDN (c) Resnet-SDN

**Fig. 2.** Performance comparison: classification accuracy of budgeted batch classification as a function of average computational budget per image on the CIFAR-100.

### ➢ The learned fusion weights

We select the important parameters of each exit according to the learned meta fusion weights, and iteratively prune them to find the correlations between these params and the performance degradation.



(a) Resnet-SDN (b) MSDnet (c) Vgg-SDN

**Fig. 4.** The accuracy degradations when pruning the important shared parameters of different exits.