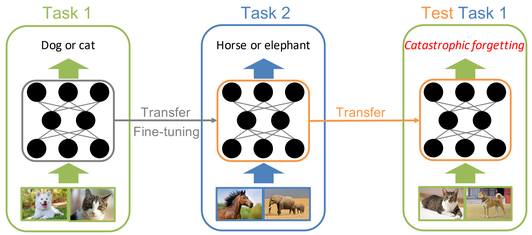
Uncertainty-based Continual Learning with Adaptive Regularization

Hongjoon Ahn* [1], Sungmin Cha* [2], Donggyu Lee [2], and Taesup Moon [1,2]
Department of Artificial Intelligence[1], Department of Electrical and Computer Engineering[2]
Sungkyunkwan University
{hong0805, csm9493, ldk308, tsmoon}@skku.edu

Catastrophic Forgetting



- **Algorithms for overcoming *Catastrophic forgetting***
  - Dynamic network architecture-based (ex. PNN, DEN)
  - Dual memory system-based (Ex. GEM, iCaRL)
  - **Regularization-based** (Ex. EWC, LWF, SI)
- **Contributions of our work**
  - Define **a notion of "Uncertainty" for each hidden node**
    - Much smaller number of additional parameters
  - Add two additional regularization terms
    - ***Freezing*** the weights that are identified to be important
    - The ***actively*** learning parameters for new tasks

Bayesian Online Learning

- **Notation and a review of Bayesian online learning**
  - In **Bayesian online learning**, they apply this standard variational inference method to the continual learning

$$\mathcal{F}(D_t, \boldsymbol{\theta}_t) = \mathbb{E}_{q(\boldsymbol{\mathcal{W}}|\boldsymbol{\theta}_t)}[-\log p(D_t|\boldsymbol{\mathcal{W}})] + D_{KL}(q(\boldsymbol{\mathcal{W}}|\boldsymbol{\theta}_t)||q(\boldsymbol{\mathcal{W}}|\boldsymbol{\theta}_{t-1})$$

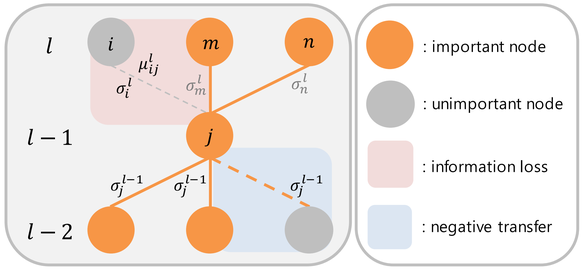
- **Interpreting KL-divergence and motivation of UCL**

$$\frac{1}{2}\sum_{l=1}^{L}\Bigg[\underbrace{\left\|\frac{\boldsymbol{\mu}_t^{(l)}-\boldsymbol{\mu}_{t-1}^{(l)}}{\boldsymbol{\sigma}_{t-1}^{(l)}}\right\|_2^2}_{(a)}+\underbrace{\mathbf{1}^\top\left\{\left(\frac{\boldsymbol{\sigma}_t^{(l)}}{\boldsymbol{\sigma}_{t-1}^{(l)}}\right)^2-\log\left(\frac{\boldsymbol{\sigma}_t^{(l)}}{\boldsymbol{\sigma}_{t-1}^{(l)}}\right)^2\right\}}_{(b)}\Bigg]$$

  - $\boldsymbol{\theta}_t^{(l)} = (\boldsymbol{\mu}_t^{(l)}, \boldsymbol{\sigma}_t^{(l)})$: the mean and standard deviation of the weight matrix for layer l at task t
  - Term (b) is minimized when $\boldsymbol{\sigma}_t^{(l)} = \boldsymbol{\sigma}_{t-1}^{(l)}$

Uncertainty-based Continual Learning(UCL)

- **Contribution 1: A notion of uncertainty for a *node***
  - We devise **a notion of uncertainty for each *node*** of the network



- **Contribution 2: Modifying term (a)**
  - ✓ **Regularization for the important connection**

$$\frac{1}{2}\Big(\sum_{l=1}^{L}\Big\|\boldsymbol{\Lambda}^{(l)}\odot(\boldsymbol{\mu}_t^{(l)}-\boldsymbol{\mu}_{t-1}^{(l)})\Big\|_2^2\Big),\ \boldsymbol{\Lambda}_{ij}^{(l)} \triangleq \max\Big(\frac{\boldsymbol{\sigma}_{\text{init}}^{(l)}}{\boldsymbol{\sigma}_{t-1,i}^{(l)}}, \frac{\boldsymbol{\sigma}_{\text{init}}^{(l-1)}}{\boldsymbol{\sigma}_{t-1,j}^{(l-1)}}\Big) \quad (4)$$

  - $\boldsymbol{\sigma}_{\text{init}}^{(l)}$: Pivot value for defining regularization strength
    - Constrain the incoming weights to the node to have the *same* variance parameters
    - Then, set that variance as the uncertainty of the node
  - ✓ **Regularization for the important weight**

$$\sum_{l=1}^{L}(\boldsymbol{\sigma}_{\text{init}}^{(l)})^2\left\|\left(\frac{\boldsymbol{\mu}_{t-1}^{(l)}}{\boldsymbol{\sigma}_{t-1}^{(l)}}\right)^2\odot(\boldsymbol{\mu}_t^{(l)}-\boldsymbol{\mu}_{t-1}^{(l)})\right\|_1 \quad (5)$$

  - A weight is important if the ratio μ/σ is high
    - The ℓ_1-norm will promote sparsity and $\mu_{t,ij}^{(l)}$ will tend to ***freeze*** to $\mu_{t-1,ij}^{(l)}$
- **Contribution 3: Modifying term (b)**
  - ✓ **Regularization for the actively learning**

$$\frac{1}{2}\mathbf{1}^\top\left((\boldsymbol{\sigma}_t^{(l)})^2-\log(\boldsymbol{\sigma}_t^{(l)})^2\right) \quad (6)$$

  - which forces $\boldsymbol{\sigma}_t^{(l)}$ to get close to $\sqrt{2}\boldsymbol{\sigma}_{t-1}^{(l)}$ when minimized with term (b)
    - Increase the number of **"*actively*"** learning nodes
    - Result in ***gracefully forgetting*** the past tasks

- **Final loss function for UCL**

$$-\log p(D_t|\boldsymbol{\mathcal{W}}) + \sum_{l=1}^{L}\Big[\frac{1}{2}\Big\|\boldsymbol{\Lambda}^{(l)}\odot(\boldsymbol{\mu}_t^{(l)}-\boldsymbol{\mu}_{t-1}^{(l)})\Big\|_2^2 + (\boldsymbol{\sigma}_{\text{init}}^{(l)})^2\Big\|\Big(\frac{\boldsymbol{\mu}_{t-1}^{(l)}}{\boldsymbol{\sigma}_{t-1}^{(l)}}\Big)^2\odot(\boldsymbol{\mu}_t^{(l)}-\boldsymbol{\mu}_{t-1}^{(l)})\Big\|_1\Big)$$
$$+\frac{\beta}{2}\mathbf{1}^\top\Big\{\Big(\frac{\boldsymbol{\sigma}_t^{(l)}}{\boldsymbol{\sigma}_{t-1}^{(l)}}\Big)^2-\log\Big(\frac{\boldsymbol{\sigma}_t^{(l)}}{\boldsymbol{\sigma}_{t-1}^{(l)}}\Big)^2+(\boldsymbol{\sigma}_t^{(l)})^2-\log(\boldsymbol{\sigma}_t^{(l)})^2\Big\}\Big], \quad (7)$$

  - β : the hyperparameter controls the increasing or decreasing speed of $\sigma_t^{(l)}$
  - The number of sampling is 1 for each iteration
- **Illustration of the regularization mechanism**

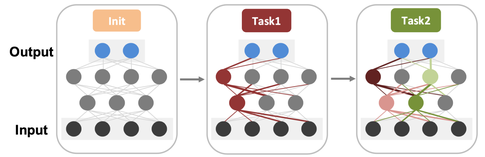


Figure 2: Colored hidden nodes and edges denote important nodes and highly regularized weights due to (4), respectively. The width of colored edge denotes the regularization strength of (5). Note as new task comes the uncertainty level of a node can vary due to (6), represented with color changes.

Experimental Results

- **Supervised learning**

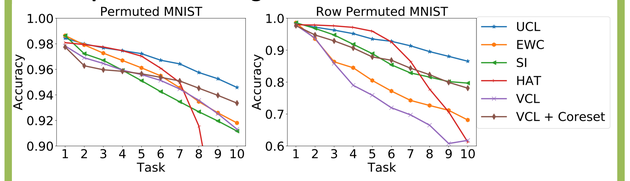


Figure 3: Experimental results on various Permuted MNIST with single head.

- ✓ **Standard deviation histogram in Permuted MNIST**

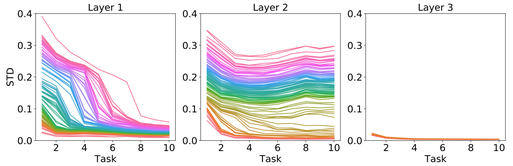


Figure 4: Standard deviation histogram in Permuted MNIST experiment. We randomly selected 100 standard deviations for layer 1 and 2. In layer 3, all 10 nodes are shown.

- ✓ **Ablation study**

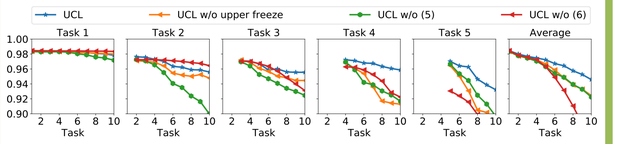
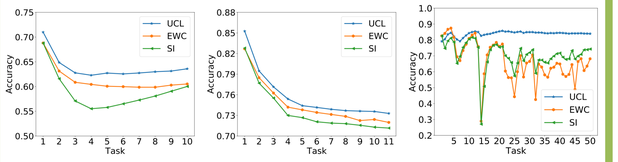


Figure 6: Ablation study in Permuted MNIST. Each line denote the test accuracy.

- ✓ **Convolutional neural network**



(a) Split CIFAR-100 (b) Split CIFAR-10/100 (c) Omniglot

Figure 7: Experiments on supervised learning using convolutional neural network

- ✓ **The number of parameters**

Table 1: The number of parameters used for each benchmark.

Dataset\Methods	Vanilla	UCL	EWC	SI	HAT	VCL
Permuted MNIST	478K	960K	1435K	1435K	486K	1914K
Split MNIST	270K	538K	808K	808K	272K	1077K
Split notMNIST	187K	375K	559K	559K	190K	749K
Split CIFAR10/100	839K	1655K	2467K	2467K	-	-
Omniglot	1773K	1884K	1995K	1995K	-	-

- **Reinforcement learning**
  - Environment: Roboschool {*Walker-Hflagrun-Hopper-Ant -IDPendulum-Cheetah-Humanoid-IPendulum*}
  - Algorithm: PPO(5m per task)
  - Baseline: EWC($\lambda = 10$), Fine-tuning

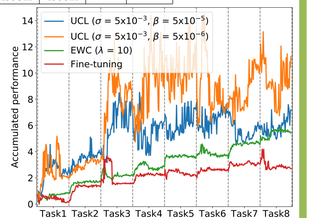


Figure 8: Accum. rewards.