**LEARNING TO MODEL THE TAIL**

Yu-Xiong Wang  
Deva Ramanan  
Martial Hebert

Email: lyuxiongw, dramanan, hebert@cs.cmu.edu

---

**Motivation**

- Intrinsic long-tailed distribution for recognition tasks in the wild
- Head-to-tail meta-knowledge transfer
  - Meta-level network: Operate on the space of model parameters
  - Model dynamics: Transformations from few-shot to many-shot models
- Progressive transfer
  - A single, chained MetaModelNet for models of different sample sizes
  - Recursive class splitting into head and tail
- An illustration: Learn both many-shot and few-shot living-room models, and train a regressor that maps between the two

---

**Regressing 1-Shot to Many-Shot Models**

- Base learner and meta-learner
  - Training set \(H_t\) of head classes: \((x, y)\) data-label pairs for classes with more than \(t\) training examples
  - Base learner \(q(x; \theta)\): Feedforward function with parameters \(\theta\)
  - Optimal model parameters \(\theta_i\): Tuning \(q\) on \(H_t\) with a standard loss function
  - Few-shot model parameters \(\theta_i\): Tuning \(q\) on random subsets of \(H_t\) with \(k\) examples per class
- MetaModelNet \(F(\theta; w)\): Meta-network regressing \(\theta\) to 0, with parameters \(w\)

**Loss function**

\[
\sum_{x \in \sum_{i \leq l, y \in H_t}} \left\{ \| F(\theta; w) - \theta_i \|^2 + \lambda \sum_{x \in \sum_{i \leq l}} \text{loss}(q(x,F(\theta;w),y)) \right\}
\]

**Parameters \(\theta\) and \(\theta_i\)**

- In principle: \(F()\) applies to model parameters from multiple CNN layers
- Shareable across classes: Parameters from the classifier module (last fully-connected layer) for a single class

---

**Regressing Different Sample-Size Few-Shot to Many-Shot Models**

- Key properties
  - Simple-sample dependency: Generate a sequence of different meta-learners \(F_i\) tuned for a specific \(k = 2^i\)
  - Identity regularization: \(F_i \rightarrow Z \rightarrow \infty\)
  - Compositionality: \(\forall j \in 2^i, F_{ij}(\theta) = F_{ij}(\theta_i)\), \(F_{ij}\) is the regressor that maps between \(k(i)\)-shot and \(k(j)\)-shot models
- Recursive residual network

\[
F_i(\theta) = F_{i-1}(\theta + f(\theta, w))
\]

**Training:** Back-to-front

**Head-tail split:** Log-linear scale

---

**Ablation Analysis**

- Importance of sample-size dependent transformation and identity regularization
- Curriculum learning in the way of recursive head-tail class splitting further improves performance
- Progressive learning through joint feature fine-tuning & model/classifier dynamics learning performs the best
- Fine-tuning the entire ResNet50

---

**Comparison with State-of-the-Art**

- Task: Scene classification on long-tailed SUN-397 with \(1 \leq 132 \) images per class
- Base learner: Fine-tuning the classifier module & Freezing the representation module of a pre-trained ResNet152

- MetaModelNet: 7 residual blocks with 1,000 1-shot, 500 2-shot, and 200 4-shot till 64-shot models as inputs
- Baselines: Over-Sampling / Under-Sampling / Cost-Sensitive
- Significant improvement for few-shot tail classes

---

**Understanding Model Dynamics**

- Structure in dual model (parameter) space: Models \(\theta \in \mathbb{R}^{256}\) as points for ResNet
- Model evolution with increasing sample sizes: Trajectories over the model space
- Class-specific data augmentation
- PCA: Approximately smooth, nonlinear warping of model space
- t-SNE: Similar semantic classes tend to be close and transform in similar ways

---

**Conclusions**

- Long-tail recognition: Meta-knowledge transfer from data-rich head to data-poor tail classes
- Representation of model dynamics: How model evolves when gradually encountering more training examples
- Learning to learn: Regressing model parameters
- Progressive learning: Back-to-front residual learning
- Curriculum learning: Learning from classes in the head to the body and then to the tail