Paper-Reading-Notes


```
### MainTitle

- [x] **MainTitle** (conf) [[paperswithcode]()]
    - Author et al. "Title"


| 核心在哪? | 精读? 代码? | 关键词? | 亮点? | 笔记时间? |
| --------- | ----------- | ------- | ----- | --------- |
|           |             |         |       |           |

---



+ **背景? 提出了什么问题?**
+ **为了解决此问题提出了什么具体的idea?**
+ **如何从该idea形式化地对问题建模、简化并解决的?**
+ **理论方面证明的定理与推导过程?**
+ **这个任务/解决方法有什么意义?**
+ **对论文的讨论/感想?**
```

D1

出发点：降低模型对样本的需要，常规思路：

- 引入正则化项，降低模型的复杂度.
- 考虑用其他方法简化模型.
- 考虑数据的先验，利用其他任务或者数据集本身的一些特性.

Meta-Learning，在学习过程是学得一个好的embedding的思路，从data学得一个 ϕ，数据点被映射到另一个特征空间.

1. Siamese Neural Network:

   训练方式为 生成pairs: (x_i, x_j, y_{ij})，两幅图片 像/不像，训练过程优化 $\|\phi(x_i) - \phi(x_j)\|_F^2$，转化成距离的远近.

   缺点：无法反映相似程度，有些图片之间更相似/更不相似无法度量.

2. 由上缺点，引入 triplet loss，(x_i, x_j, x_k) 其中x_i和x_j之间相似，x_i和x_k之间不相似，$\min_\phi \mathcal{D}_{ik} - \mathcal{D}_{ij}$.

Multi-Task learning 视角:

两个数据集:

- $(x_i^{(1)}, y_i^{(1)}) \sim \mathcal{D}_1$， N_1
- $(x_i^{(2)}, y_i^{(2)}) \sim \mathcal{D}_2$， N_2

有点像第一个数据集是动物实拍图像，第二个是动物标本图像，两个数据集有相似的地方，也有不相似的地方，设计模型(两种极端情况):

1. 混合 $\mathcal{D}_1, \mathcal{D}_2$，相当于参数全部共享.
2. 单独训练，相当于参数全部分开.

设计模型的时候要考虑共享的训练参数，也要考虑到不同的部分:

$$\min_w \sum_{i=1}^{N_i} \mathcal{L}_t \left(\Delta w_t (w^T x_i^{(t)}) \right)$$

如上式，w_t就是不同数据集(任务)上不同的参数，w就是共有的部分.

具体的训练过程，针对$\mathcal{T}$个Task就是:

$$\min_\phi \frac{1}{T} \sum_{t=1}^{T} \min_{w_t} \frac{1}{N_t} \mathcal{L} \left(w_t^T \phi(x_i^T) \right)$$

注意上式 ϕ 就相当于共享的参数，w_t 就相当于独立的参数，上式的训练过程就像Meta-Learning的训练过程(两层优化): 现在内层优化 w_t，然后出来优化 ϕ，如此迭代循环.

转化到Meta-Learning，就是要适应新的类，就是之前没有见过的类也要会分类，则为"省略共享的参数":

$$\min_\phi \frac{1}{T} \sum_{t=1}^{T} \frac{1}{N_Q^t} \sum_{i}^{N_Q^t} \mathcal{L} \left(\phi(x_i^{query}); \phi(x_i^{Support}) \right))$$

Basic Method

Batch Normalization

- [x] **Batch Normalization** (2015) [paperswithcode code snippet]
  - loffe et al. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift"

核心在哪?	精读? 代码?	关键词?	亮点?	笔记时间?

- 背景? 提出了什么问题?

- 为了解决此问题提出了什么具体的**idea?**
- 如何从该**idea**形式化地对问题建模、简化并解决的**?**
- 理论方面证明的定理与推导过程**?**
- 这个任务**/**解决方法有什么意义**?**
- 对论文的讨论**/**感想**?**

Basic CNN Architectures

AlexNet

- [x] **AlexNet** (NIPS 2012) [paper] [code (PyTorch)]
    - Alex Krizhevsky et al. "ImageNet Classification with Deep Convolutional Neural Networks"

核心在哪**?**	精读**?** 代码**?**	关键词**?**	亮点**?**	阅读时间**?**
Figure2(3.5节) & 防止过拟合技巧(4节)	精读, 代码	大CNN, 基本处理过拟合技巧		2020-10-2

- 提出了什么问题**?**

  ImageNet 太大, 其目标检测任务是具有 the immense complexity 的.
- 为了解决此问题提出了什么具体的**idea?**

  利用先验知识, compensate 没有的数据. CNN 模型.
- 如何从该**idea**形式化地对问题建模、简化并解决的**?**
    - current GPUs, paired with a highly-optimized implementation of 2D convolution: 在ImageNet, 大CNN模型上训练, 并且大数据集可以防止过拟合.

      > Current GPUs are particularly well-suited to cross-GPU parallelization, as they are able to read from and write to one another's memory directly, without going through host machine memory.

    - 输入:

      ImageNet 是 variable-resolution 的, 下采样到 `256 x 256`.

    - 网络结构:

      5层卷积, 3层全连接.

      激活函数: ReLU (non-saturating nonlinearity), 因为 tanh 之类的太慢.

        - Local Response Normalization (LRN):

          已被证明并无实际作用, 反而增加不少的计算量, 一般都不再使用.

          激活的神经元抑制相邻的. ReLU 不需要输入归一化, 但是局部归一化有助于泛化性能, 对局部神经元创建了竞争的机制, 提高精度.

          kernel i 在 (x, y) 的激活值 归一化, $\sum$ 作用于 相邻的 映射在相同空间位置的 kernel:

          $$b_{x,y}^i = a_{x,y}^i / \left(k + \alpha \sum_{j=\max(0, i-n/2)}^{\min(N-1, i+n/2)} \left(a_{x,y}^j \right)^2 \right)^\beta$$

        - Overlapping Pooling:

pooling unit 大小为 `z x z`，`stride < z, kernel_size=3, stride=2` 时为overlap的.

这可以防止过拟合，提升特征的丰富性.

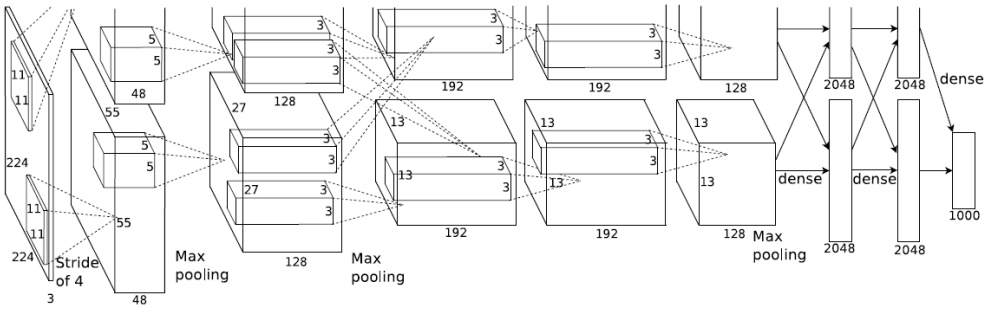


Figure 2: An illustration of the architecture of our CNN, explicitly showing the delineation of responsibilities between the two GPUs. One GPU runs the layer-parts at the top of the figure while the other runs the layer-parts at the bottom. The GPUs communicate only at certain layers. The network's input is 150,528-dimensional, and the number of neurons in the network's remaining layers is given by 253,440–186,624–64,896–64,896–43,264–4096–4096–1000.

- 减少过拟合:

  网络很大，很容易过拟合.

  Data Augmentation:

  1 label-preserving transformations:

  裁切，缩放，翻转.

  > We do this by extracting random 224 x 224 patches (and their horizontal reflections) from the 256 x 256 images and training our network on these extracted patches4. This increases the size of our training set by a factor of 2048, though the resulting training examples are, of course, highly interdependent.

  2 altering the intensities of the RGB channels:

  > Specifically, we perform PCA on the set of RGB pixel values throughout the ImageNet training set. To each training image, we add multiples of the found principal components, with magnitudes proportional to the corresponding eigenvalues times a random variable drawn from a Gaussian with mean zero and standard deviation 0.1.

  即在每个通道上(RGB三个)增加:

  $$[\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3]\,[\alpha_1\lambda_1, \alpha_2\lambda_2, \alpha_3\lambda_3]^T$$

  其中 $\mathbf{p}_i/\lambda_i$ 是特征向量/值，α_i 是高斯随机变量.

  3 Dropout:

  > setting to zero the output of each hidden neuron with probability 0.5.
  >
  > The neurons which are "dropped out" in this way **do not** contribute to the forward pass and do not participate in backpropagation.
  >
  > 所以不同时刻的输入是参与不同的 神经网络架构的. 但是这些不同架构共享参数. 一个神经元不能依赖于其他特定神经元的存在. 因此，它被迫学习更健壮的特征，这些特征在与其他神经元的许多不同随机子集结合时是有用的.

- **理论方面证明的定理与推导过程?**

- 这个任务/解决方法有什么意义?

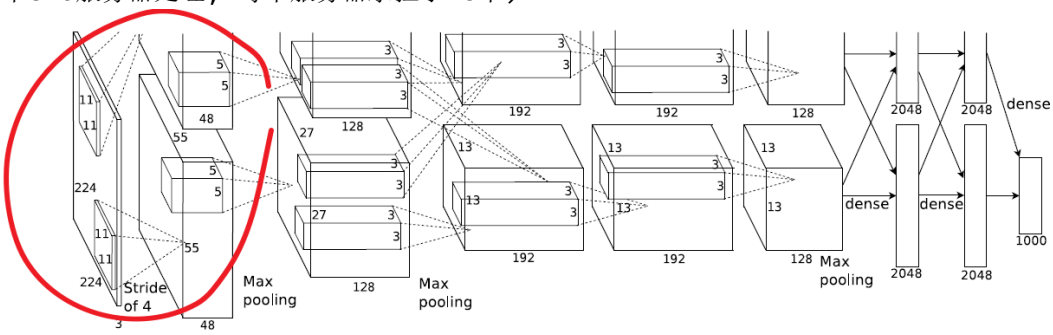
  历史意义远大于模型影响. 在此之后, 深度学习重新迎来春天.

  引爆了神经网络的应用热潮, 赢得了2012图像识别大赛的冠军. 使CNN成为在图像分类上的核心算法模型.

- 对论文的讨论/感想?

  要搞清楚 PyTorch 实现中各层参数维度. 卷积核个数/高/宽/深度.

  例子: `227 x 227 x 3` 的输入图像和卷积核 `11 x 11 x 3` 做卷积, 步长为4, 卷积后的尺寸是 `55 x 55 x 1`. 因为: (227 - 11 + 2 * 0) / 4 + 1 = 55.
  一共有96个卷积核, 最后的输出: 55 x 55 x 48 x 2. (图上之所以看起来是48个是由于2个GPU服务器处理, 每个服务器承担了48个):



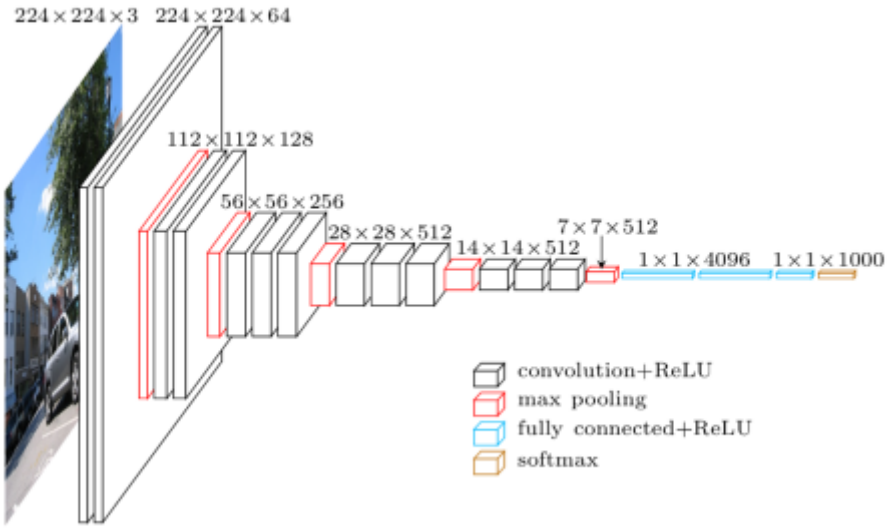
GoogLeNet, a.k.a **Inception v.1** (2014) [paper]

- Szegedy et al. "Going Deeper with Convolutions" [Google]
- Original LeNet page from Yann LeCun's homepage.
- **Inception v.2 and v.3** (2015) Szegedy et al. "Rethinking the Inception Architecture for Computer Vision" [paper]
- **Inception v.4 and InceptionResNet** (2016) Szegedy et al. "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning" [paper]
- "A Simple Guide to the Versions of the Inception Network" [blogpost]

VGG

- [ ] **VGG** (2014) [paperswithcode]
  - **Simonyan et al. "Very Deep Convolutional Networks for Large-Scale Image Recognition" (2014) [Google DeepMind & Oxford's Visual Geometry Group (VGG)] [paper]**
  - *VGG-16*: Zhang et al. "Accelerating Very Deep Convolutional Networks for Classification and Detection" [paper]

核心在哪?	精读? 代码?	关键词?	亮点?	笔记时间?



- 背景**?** 提出了什么问题**?**
- 为了解决此问题提出了什么具体的**idea?**
- 如何从该**idea**形式化地对问题建模、简化并解决的**?**
- 理论方面证明的定理与推导过程**?**
- 这个任务**/**解决方法有什么意义**?**
- 对论文的讨论**/**感想**?**

GoogLeNet

- [ ] **MainTitle** (CVPR 2015) [paperswithcode]
    - **Szegedy et al. "Going Deeper with Convolutions" [Google]**
    - Original LeNet page from Yann LeCun's homepage.
    - **Inception v.2 and v.3** (2015) Szegedy et al. "Rethinking the Inception Architecture for Computer Vision" [paper]
    - **Inception v.4 and InceptionResNet** (2016) Szegedy et al. "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning" [paper]
    - "A Simple Guide to the Versions of the Inception Network" [blogpost]

核心在哪**?**	精读**?** 代码**?**	关键词**?**	亮点**?**	笔记时间**?**

- 背景**?** 提出了什么问题**?**
- 为了解决此问题提出了什么具体的**idea?**
- 如何从该**idea**形式化地对问题建模、简化并解决的**?**
- 理论方面证明的定理与推导过程**?**
- 这个任务**/**解决方法有什么意义**?**
- 对论文的讨论**/**感想**?**

ResNet

- [ ] **ResNet** (CVPR 2016) [paperswithcode (ResNet Explained)]
    - He et al. "Deep Residual Learning for Image Recognition"

核心在哪?	精读? 代码?	关键词?	亮点?	笔记时间?
	精读，代码			

- 背景? 提出了什么问题?
- 为了解决此问题提出了什么具体的**idea?**
- 如何从该**idea**形式化地对问题建模、简化并解决的?
- 理论方面证明的定理与推导过程?
- 这个任务/解决方法有什么意义?
- 对论文的讨论/感想?

Meta Learning

如何才能让模型仅用少量的数据学习?
我们期望好的meta learning模型能够具备强大的适应能力和泛化能力. 在测试时，模型会先经过一个自适应环节（adaptation process），即根据少量样本学习任务. 自适应本质上来说就是一个短暂的学习过程.

最佳的meta learning模型参数可以表示为:

$$\theta^* = \arg\min_\theta \mathbb{E}_{\mathcal{D}\sim p(\mathcal{D})}\left[\mathcal{L}_\theta(\mathcal{D})\right]$$

上式中的每个*数据集*是一个*数据样本*. Few-shot Learning中数据集 $\mathcal{D}$ 被分为 support set/query set.

Training Set 是 Support Set, Test Set 是 Query Set.

- ⋆ N-ways K-shot classification:
  在 Training 和 Test Tasks 里，有 N 个类别，每个类别有 K 个用例.

Siamese Network

- [x] **Siamese Network** (ICML 2015) [paper] [code (keras)]
  - Gregory Koch et al. "Siamese Neural Networks for One-shot Image Recognition"

核心在哪?	精读? 代码?	关键词?	亮点?	阅读时间?
Model 3.1节	精读			2020-10-2

- 提出了什么问题?
- 为了解决此问题提出了什么具体的**idea?**

- 如何从该**idea**形式化地对问题建模、简化并解决的?
  - **Model:**

$$a_{1,m}^{(k)} = \text{max-pool}\left(\max\left(0, \mathbf{W}_{l-1,l}^{(k)} \star \mathbf{h}_{1,(l-1)} + \mathbf{b}_l\right), 2\right)$$
$$a_{2,m}^{(k)} = \text{max-pool}\left(\max\left(0, \mathbf{W}_{l-1,l}^{(k)} \star \mathbf{h}_{2,(l-1)} + \mathbf{b}_l\right), 2\right)$$

共享参数.

预测(输出层):

$$\mathbf{p}=\sigma\left(\sum_j \alpha_j\left|\mathbf{h}_{1,L-1}^{(j)}-\mathbf{h}_{2,L-1}^{(j)}\right|\right)$$

请注意最关键就在上述式子, x embedding之后比较距离, 接softmax转换成概率.

> This final layer induces a metric on the **learned** feature space of the (L - 1)th hidden layer and **scores** the similarity between the two feature vectors.

loss function:

$$\mathcal{L}\left(x_1^{(i)},x_2^{(i)}\right)=\mathbf{y}\left(x_1^{(i)},x_2^{(i)}\right)\log\mathbf{p}\left(x_1^{(i)},x_2^{(i)}\right)+$$
$$\left(1-\mathbf{y}\left(x_1^{(i)},x_2^{(i)}\right)\right)\log\left(1-\mathbf{p}\left(x_1^{(i)},x_2^{(i)}\right)\right)+\boldsymbol{\lambda}^T|\mathbf{w}|^2$$

正类: 两个同一个类, 相似, 即 $\mathbf{y}\left(x_1^{(i)},x_2^{(i)}\right)=1$, 反之等于0.

Optimization:

$$\mathbf{w}_{kj}^{(T)}\left(x_1^{(i)},x_2^{(i)}\right)=\mathbf{w}_{kj}^{(T)}+\Delta\mathbf{w}_{kj}^{(T)}\left(x_1^{(i)},x_2^{(i)}\right)+2\lambda_j\left|\mathbf{w}_{kj}\right|$$
$$\Delta\mathbf{w}_{kj}^{(T)}\left(x_1^{(i)},x_2^{(i)}\right)=-\eta_j\nabla w_{kj}^{(T)}+\mu_j\Delta\mathbf{w}_{kj}^{(T-1)}$$

> the gradient is additive **across the twin networks** due to the tied weights.

> **learning rates** were decayed uniformly across the network by 1 percent per epoch, so that $\eta_j^{(T)}=0.99\eta_j^{(T-1)}$.

> We used the beta version of Whetlab, a Bayesian optimization framework, to perform **hyperparameter selection**.

Weight initialization:

卷积层: $\mathcal{N}(0,10^{-2})$, bias: $\mathcal{N}(0.5,10^{-2})$.

Affine distortions:

$$\mathrm{x}_1'=T_1\left(\mathrm{x}_1\right),\mathrm{x}_2'=T_2\left(\mathrm{x}_2\right)$$

其中T_i是 multidimensional uniform distribution 中随机选取的.

- 理论方面证明的定理与推导过程?
- 这个任务/解决方法有什么意义?

  假设: 学到的 embedding 在未见过的分类上依然能很好的衡量图片间的距离, 这与迁移学习中是一致的.

  > a) are capable of **learning generic image features** useful for making predictions about unknown class distributions even when very few examples from these **new distributions are available**;
  >
  > b) are easily trained **using standard optimization techniques** on pairs sampled from the source data;
  >
  > c) provide a competitive approach that **does not rely** upon **domain-specific** knowledge by instead exploiting deep learning techniques.

- 对论文的讨论/感想?

关注Siamese的不足在哪，似乎没有考虑其他训练集数据的影响，就只是与单一对的0/1比较？
对比matching和Prototypical.

Matching Networks

- [x] **Matching Networks** (NIPS 2016) [paper] [code (PyTorch)]
    - Oriol Vinyals et al. "Matching Networks for One Shot Learning"

核心在哪？	精读？代码？	关键词？	亮点？	阅读时间？
2.1.1, 2.1.2, Appendix	精读			2020-10-2

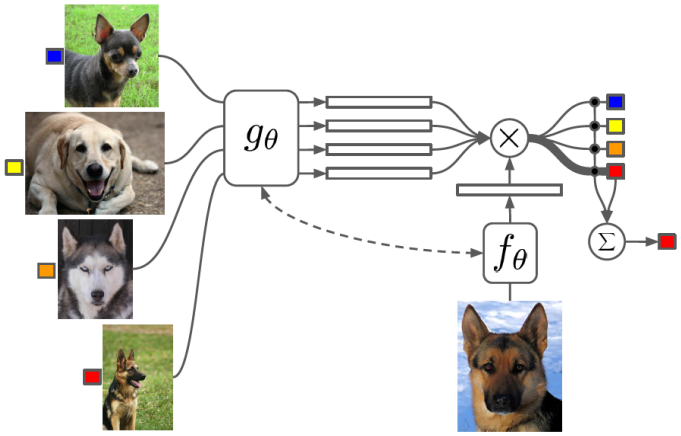


Figure 1: Matching Networks architecture

- 背景？提出了什么问题？

> a neural attention mechanism, often fully differentiable, is defined to access (or read) a memory matrix which stores useful information to solve the task at hand.

- 为了解决此问题提出了什么具体的**idea**？

> cast the problem of one-shot learning within the set-to-set framework

> reinterpreting a well studied framework (neural networks with external memories) to do one-shot learning.

分到某个类的权重由在 support set 中一系列label的加权和，权重由一个注意力核(attention kernel) $a(\mathbf{x}, \mathbf{x}_i)$ 决定.

权重应当与 $\mathbf{x}_i$ 和 $\mathbf{x}_i$ 间的相似度成正比.

- 如何从该**idea**形式化地对问题建模、简化并解决的？

$$\hat{y} = \sum_{i=1}^{k} a\left(\hat{x}, x_i\right) y_i$$

Attention kernel由两个embedding function f 和 g 决定.

分别用于encoding测试样例和支持集样本. 两个样本之间的注意力权重是经过softmax归一化后的，embedding vectors之间cosine距离 $\operatorname{cosine}(\cdot)$.

$$a\left(\mathbf{x}, \mathbf{x}_i\right) = \frac{\exp(\operatorname{cosine}(f(\mathbf{x}), g\left(\mathbf{x}_i\right))}{\sum_{j=1}^{k} \exp(\operatorname{cosine}(f(\mathbf{x}), g\left(\mathbf{x}_j\right))}$$

- **Full Context Embeddings**:

  只把一个数据样本作为**embedding function**的输入，会导致很难高效的估计出整个特征空间.

  通过把整个支持集 S 作为embedding function的额外输入来加强embedding的有效性，相当于给样本添加了语境，让embedding根据样本与支持集中样本的关系进行调整.

  以下看Appendix:

  $g_\theta(\mathbf{x}_i, S)$ 在整个支持集 S 的语境下用一个双向LSTM来编码 $\mathbf{x}_i$
  $f_\theta(\mathbf{x}, S)$ 在支持集 S 上使用read attention机制编码测试样本 $\mathbf{x}$

  1. 首先测试样本经过一个简单的神经网络，比如CNN以抽取基本特征 $f'(\mathbf{x})$.
  2. 然后，一个带有**read attention vector**的**LSTM**被训练用于生成部分hidden state:

$$\begin{aligned}\hat{\mathbf{h}}_t, \mathbf{c}_t &= \mathrm{LSTM}(f'(\mathbf{x}), [\mathbf{h}_{t-1}, \mathbf{r}_{t-1}], \mathbf{c}_{t-1}) \\ \mathbf{h}_t &= \hat{\mathbf{h}}_t + f'(\mathbf{x}) \\ \mathbf{r}_{t-1} &= \sum_{i=1}^{k} a\left(\mathbf{h}_{t-1}, g\left(\mathbf{x}_i\right)\right) g\left(\mathbf{x}_i\right)\end{aligned}$$

$$a\left(\mathbf{h}_{t-1}, g\left(\mathbf{x}_i\right)\right) = \mathrm{softmax}\left(\mathbf{h}_{t-1}^{\top} g\left(\mathbf{x}_i\right)\right) = \frac{\exp\left(\mathbf{h}_{t-1}^{\top} g\left(\mathbf{x}_i\right)\right)}{\sum_{j=1}^{k} \exp\left(\mathbf{h}_{t-1}^{\top} g\left(\mathbf{x}_j\right)\right)}$$

  3. 最终，如果我们做k步的读取 $f(\mathbf{x}, S) = \mathbf{h}_K$.

  这类方法对于困难的任务（few-shot classification on mini ImageNet）有所帮助.

- 理论方面证明的定理与推导过程?

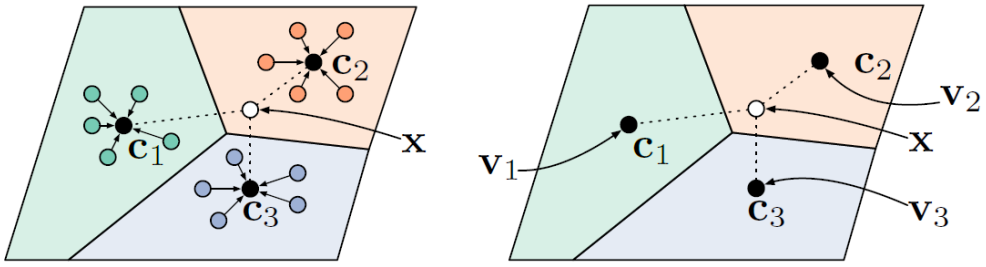
> We propose embedding the elements of the set through a function which takes as input the full set S in addition to x_i, i.e. g becomes $g(x_i, S)$. Thus, as a function of the whole support set S, g can modify how to embed x_i. This could be useful when some element x_j is very close to x_i, in which case it may be beneficial to change the function with which we embed x_i— some evidence of this is discussed **in Section 4** . We use a bidirectional Long-Short Term Memory (LSTM) [8] to encode x_i in the context of the support set S, considered as a sequence (**see appendix** for a more precise definition).

- 这个任务/解决方法有什么意义?
- 对论文的讨论/感想?

Prototypical Networks

- [x] **Prototypical Networks** (NIPS 2017) [paper] [code]
  - Snell et al. "Prototypical Networks for Few-shot Learning"

核心在哪?	精读? 代码?	关键词?	亮点?	笔记时间?
Model 2.2节	精读	Prototypical Networks	新的网络	2020-10-2



(a) Few-shot (b) Zero-shot

Figure 1: Prototypical networks in the few-shot and zero-shot scenarios. **Left**: Few-shot prototypes $\mathbf{c}_k$ are computed as the mean of embedded support examples for each class. **Right**: Zero-shot prototypes $\mathbf{c}_k$ are produced by embedding class meta-data $\mathbf{v}_k$. In either case, embedded query points are classified via a softmax over distances to class prototypes: $p_{\boldsymbol{\phi}}(y = k|\mathbf{x}) \propto \exp(-d(f_{\boldsymbol{\phi}}(\mathbf{x}), \mathbf{c}_k))$.

- 提出了什么问题?
- 为了解决此问题提出了什么具体的**idea?**

  对于meta-data, 学到它们的embedding (non-linear mapping), 在shared space构建每个类的prototype. 对于query set的, 先embedding, 再找最相邻的prototype.

- 如何从该**idea**形式化地对问题建模、简化并解决的?
  - **Model:**

    embedding function $f_\phi : \mathbb{R}^D \to \mathbb{R}^M$, 参数 ϕ. prototype $\mathbf{c}_k$ 是 support set 里面数据的embedding后均值:

    $$\mathbf{c}_k = \frac{1}{|S_k|} \sum_{(\mathbf{x}_i, y_i) \in S_k} f_{\boldsymbol{\phi}}(\mathbf{x}_i)$$

    对于每个query point x, 基于"距离"原型网络建立了该点到 每个原型$\mathbf{c}_k$的概率分布:

    $$p_{\boldsymbol{\phi}}(y = k \mid \mathbf{x}) = \frac{\exp(-d(f_{\boldsymbol{\phi}}(\mathbf{x}), \mathbf{c}_k))}{\sum_{k'} \exp(-d(f_{\boldsymbol{\phi}}(\mathbf{x}), \mathbf{c}_{k'}))}$$

  - 学习过程:

    通过SGD最小化负对数概率 (到每个原型的): $J(\boldsymbol{\phi}) = -\log p_{\boldsymbol{\phi}}(y = k \mid \mathbf{x})$

    training episode是training set的随机选取的子集, 在每个类中选择一些作为support set, 其余是query set. 结合episodes和mini-batch的思路模拟测试场景进行训练.

    伪代码:

Algorithm 1 Training episode loss computation for prototypical networks. N is the number of examples in the training set, K is the number of classes in the training set, $N_C \leq K$ is the number of classes per episode, N_S is the number of support examples per class, N_Q is the number of query examples per class. RANDOMSAMPLE(S, N) denotes a set of N elements chosen uniformly at random from set S, without replacement.

Input: Training set $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$, where each $y_i \in \{1, \ldots, K\}$. $\mathcal{D}_k$ denotes the subset of $\mathcal{D}$ containing all elements $(\mathbf{x}_i, y_i)$ such that $y_i = k$.
Output: The loss J for a randomly generated training episode.

$V \leftarrow$ RANDOMSAMPLE$(\{1, \ldots, K\}, N_C)$ ▷ Select class indices for episode
for k in $\{1, \ldots, N_C\}$ **do**
  $S_k \leftarrow$ RANDOMSAMPLE$(\mathcal{D}_{V_k}, N_S)$ ▷ Select support examples
  $Q_k \leftarrow$ RANDOMSAMPLE$(\mathcal{D}_{V_k} \setminus S_k, N_Q)$ ▷ Select query examples
  $\mathbf{c}_k \leftarrow \frac{1}{N_C} \sum_{(\mathbf{x}_i, y_i) \in S_k} f_{\boldsymbol{\phi}}(\mathbf{x}_i)$ ▷ Compute prototype from support examples
end for
$J \leftarrow 0$ ▷ Initialize loss
for k in $\{1, \ldots, N_C\}$ **do**
  for $(\mathbf{x}, y)$ in Q_k **do**
    $J \leftarrow J + \frac{1}{N_C N_Q} \left[d(f_{\boldsymbol{\phi}}(\mathbf{x}), \mathbf{c}_k)) + \log \sum_{k'} \exp(-d(f_{\boldsymbol{\phi}}(\mathbf{x}), \mathbf{c}_k)) \right]$ ▷ Update loss
  end for
end for

- 学习过程是 Mixture Density Estimation:

  使用[Bregman散度](), 相当于在support set上用指数族 混合密度估计. Bregman定义如下:

$$d_{\varphi}(\mathbf{z}, \mathbf{z}') = \varphi(\mathbf{z}) - \varphi(\mathbf{z}') - (\mathbf{z} - \mathbf{z}')^T \nabla \varphi(\mathbf{z}')$$

  where φ is a differentiable, strictly convex function of the Legendre type.

  注意Bregman散度 通过改变φ 可以生成不同的距离度量:

Table 1: Bregman divergences generated from some convex functions.

Domain	$\varphi(\mathbf{x})$	$d_{\varphi}(\mathbf{x},\mathbf{y})$	Divergence
$\mathbb{R}$	x^2	$(x-y)^2$	Squared loss
$\mathbb{R}_+$	$x \log x$	$x \log(\frac{x}{y}) - (x-y)$	
$[0,1]$	$x \log x + (1-x)\log(1-x)$	$x \log(\frac{x}{y}) + (1-x)\log(\frac{1-x}{1-y})$	Logistic loss [3]
$\mathbb{R}_{++}$	$-\log x$	$\frac{x}{y} - \log(\frac{x}{y}) - 1$	Itakura-Saito distance
$\mathbb{R}$	e^x	$e^x - e^y - (x-y)e^y$	
$\mathbb{R}^d$	$\|\mathbf{x}\|^2$	$\|\mathbf{x}-\mathbf{y}\|^2$	Squared Euclidean distance
$\mathbb{R}^d$	$\mathbf{x}^T A \mathbf{x}$	$(\mathbf{x}-\mathbf{y})^T A (\mathbf{x}-\mathbf{y})$	Mahalanobis distance [4]
d-Simplex	$\sum_{j=1}^d x_j \log_2 x_j$	$\sum_{j=1}^d x_j \log_2(\frac{x_j}{y_j})$	KL-divergence
$\mathbb{R}_+^d$	$\sum_{j=1}^d x_j \log x_j$	$\sum_{j=1}^d x_j \log(\frac{x_j}{y_j}) - \sum_{j=1}^d (x_j - y_j)$	Generalized I-divergence

  为什么选择均值作为 prototype:
  It has been shown [4] for Bregman divergences that the cluster representative achieving minimal distance to its assigned points is the cluster mean.

下面似乎是直接写出了混合密度估计中后验的表达式(指数族混合模型)，关于Bregman散度：

Moreover, any regular exponential family distribution $p_\psi(\mathbf{z}|\boldsymbol{\theta})$ with parameters $\boldsymbol{\theta}$ and cumulant function ψ can be written in terms of a uniquely determined regular Bregman divergence [4]:

$$p_\psi(\mathbf{z}|\boldsymbol{\theta}) = \exp\{\mathbf{z}^T\boldsymbol{\theta} - \psi(\boldsymbol{\theta}) - g_\psi(\mathbf{z})\} = \exp\{-d_\varphi(\mathbf{z}, \boldsymbol{\mu}(\boldsymbol{\theta})) - g_\varphi(\mathbf{z})\} \tag{4}$$

Consider now a regular exponential family mixture model with parameters $\boldsymbol{\Gamma} = \{\boldsymbol{\theta}_k, \pi_k\}_{k=1}^K$:

$$p(\mathbf{z}|\boldsymbol{\Gamma}) = \sum_{k=1}^K \pi_k p_\psi(\mathbf{z}|\boldsymbol{\theta}_k) = \sum_{k=1}^K \pi_k \exp(-d_\varphi(\mathbf{z}, \boldsymbol{\mu}(\boldsymbol{\theta}_k)) - g_\varphi(\mathbf{z})) \tag{5}$$

Given $\boldsymbol{\Gamma}$, inference of the cluster assignment y for an unlabeled point $\mathbf{z}$ becomes:

$$p(y = k|\mathbf{z}) = \frac{\pi_k \exp(-d_\varphi(\mathbf{z}, \boldsymbol{\mu}(\boldsymbol{\theta}_k)))}{\sum_{k'} \pi_{k'} \exp(-d_\varphi(\mathbf{z}, \boldsymbol{\mu}(\boldsymbol{\theta}_k)))} \tag{6}$$

For an equally-weighted mixture model with one cluster per class, cluster assignment inference (6) is equivalent to query class prediction (2) with $f_\phi(\mathbf{x}) = \mathbf{z}$ and $\mathbf{c}_k = \boldsymbol{\mu}(\boldsymbol{\theta}_k)$. In this case, prototypical networks are effectively performing mixture density estimation with an exponential family distribution determined by d_φ. The choice of distance therefore specifies modeling assumptions about the class-conditional data distribution in the embedding space.

- 作为线性模型的解释：

  $$-\|f_\phi(x) - c_k\|^2 = -f_\phi(x)^T f_\phi(x) + 2c_k^T f_\phi(x) - c_k^T c_k$$

  RHS 第一项不变，所以后面对应线性模型有：

  $$2c_k^T f_\phi(x) - c_k^T c_k = w_k^T f_\phi(x) + b_k, \text{其中 } w_k = 2c_k, b_k = -c_k^T c_k$$

  这个假设是基于：all of the required non-linearity can be learned within the embedding function.

- **理论方面证明的定理与推导过程？**

  [4] Arindam Banerjee, Srujana Merugu, Inderjit S Dhillon, and Joydeep Ghosh. Clustering with bregman divergences. Journal of machine learning research, 6(Oct):1705–1749, 2005.

- **这个任务/解决方法有什么意义？**
- **对论文的讨论/感想？**

  注意考虑模型和其他之间的关联(比如上面和混合密度估计/线性模型).

  学习过程的细节？ $\mathbf{c}_k$ 是不断变化的.

Relation Network

- [x] **Relation Network** (CVPR 2018) [paper] [code (PyTorch)]
  - Sung et al. "Learning to Compare: Relation Network for Few-Shot Learning"

核心在哪？	精读？代码？	关键词？	亮点？	阅读时间？
3.2 Model 3.4 Network Architecture	精读			2020-10-3

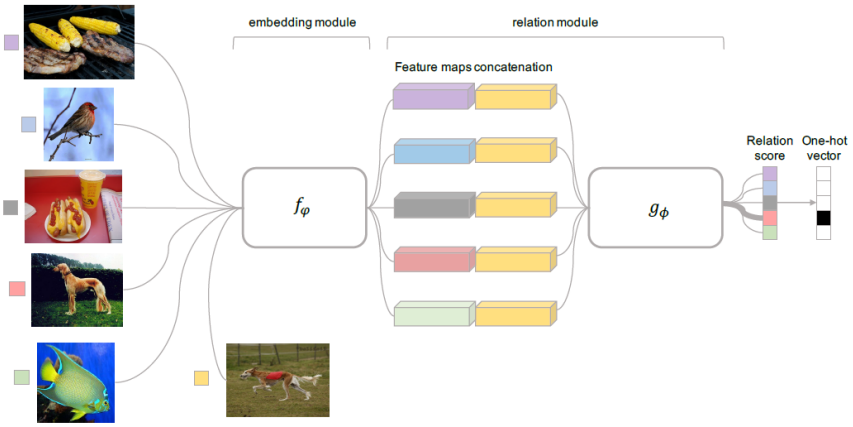


Figure 1: Relation Network architecture for a 5-way 1-shot problem with one query example.

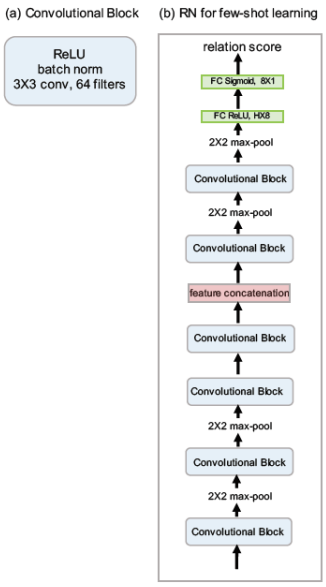


Figure 2: Relation Network architecture for few-shot learning (b) which is composed of elements including convolutional block (a).

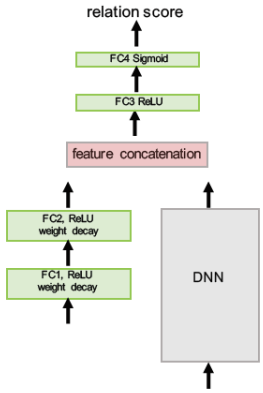


Figure 3: Relation Network architecture for zero-shot learning.

5-way 1-shot contains 15 query images, and the **5-way 5-shot** has 10 query images for each of the C sampled classes in each training episode. This means for example that there are $15\times5+1\times5=80$ images in one training episode/mini-batch for 5-way 1-shot experiments. We resize input images to 84×84. Our model is trained end-to-end from scratch, with random initialisation, and no additional training set.

- 背景**?** 提出了什么问题**?**

- 为了解决此问题提出了什么具体的**idea?**

> As most few-shot learning models utilize four convolutional blocks for embedding module [39, 36], we follow the same architecture setting for fair comparison, see Figure 2.

> More concretely, each convolutional block contains a 64-filter `3 x 3` convolution, a batch normalization and a ReLU nonlinearity layer respectively. The first two blocks also contain a `2 x 2` max-pooling layer while the latter two do not.

- 如何从该**idea**形式化地对问题建模、简化并解决的**?**
  - **Model:** (Figure 1)

    1 an embedding module f_φ:

    feature maps & combine(a concatenation):

    $$\mathcal{C}\left(f_\varphi\left(x_i\right), f_\varphi\left(x_j\right)\right)$$

    2 a relation module g_φ:

生成 relation score: $\in [0,1]$ **representing the similarity** between x_i and x_j:

$$r_{i,j} = g_\phi\left(\mathcal{C}\left(f_\varphi\left(x_i\right), f_\varphi\left(x_j\right)\right)\right), \quad i = 1, 2, \ldots, C$$

- 损失函数:

  目标优化函数是MSE损失，而不是cross-entropy，因为RN在预测时更倾向于把相似系数预测过程作为一个**regression**问题，而不是二分类问题.

$$\varphi, \phi \leftarrow \underset{\varphi,\phi}{\operatorname{argmin}} \sum_{i=1}^{m} \sum_{j=1}^{n} \left(r_{i,j} - 1\left(y_i == y_j\right)\right)^2$$

- 理论方面证明的定理与推导过程?
- 这个任务/解决方法有什么意义?
- 对论文的讨论/感想?

MAML

- [ ] **MAML** (ICML 2017) [paper] [code (原) code(PyTorch)]
  - Finn et al. "Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks"

核心在哪?	精读? 代码?	关键词?	亮点?	笔记时间?
	精读			

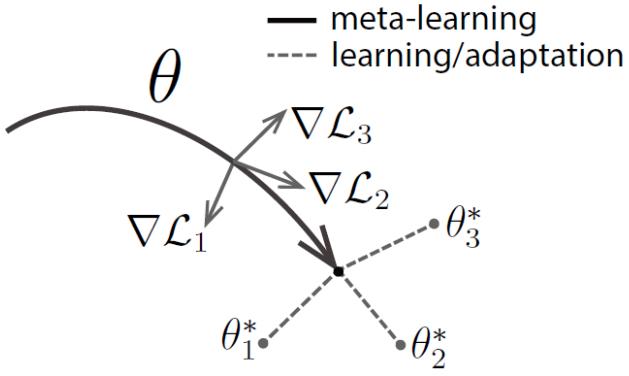


Figure 1. Diagram of our model-agnostic meta-learning algorithm (MAML), which optimizes for a representation θ that can quickly adapt to new tasks.

Algorithm 1 Model-Agnostic Meta-Learning

Require: $p(\mathcal{T})$: distribution over tasks
Require: α, β: step size hyperparameters
1: randomly initialize θ
2: **while** not done **do**
3: Sample batch of tasks $\mathcal{T}_i \sim p(\mathcal{T})$
4: **for all** $\mathcal{T}_i$ **do**
5: Evaluate $\nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$ with respect to K examples
6: Compute adapted parameters with gradient descent: $\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$
7: **end for**
8: Update $\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta_i'})$
9: **end while**

- 背景？ 提出了什么问题？
- 为了解决此问题提出了什么具体的**idea**？
- 如何从该**idea**形式化地对问题建模、简化并解决的？

假设我们的模型是 f_θ，参数为 θ 给定一个任务 τ_i 和其相应的数据集 $\left(\mathcal{D}_{\text{train}}^{(i)}, \mathcal{D}_{\text{test}}^{(i)}\right)$，我们可以对模型参数进行一次或多次梯度下降。(下式中只进行了一次迭代)：

$$\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{\tau_i}^{(0)}(f_\theta)$$

其中 $\mathcal{L}^{(0)}$ 是由编号为0的小数据batch算得的loss.

当然，上面这个式子只针对一个特定的任务进行了优化。而MAML为了能够更好地扩展到一系列任务上，我们想要寻找一个在给定任意任务后微调过程最高效的 θ^*.

现在假设我们采样了一个编号为1的数据batch用于更新元目标，对应的loss记为 $\mathcal{L}^{(1)}$. $\mathcal{L}^{(0)}$ 和 $\mathcal{L}^{(1)}$ 的上标只代表着数据batch不同，都是同一个目标方程计算得到的。那么：

$$\theta^* = \arg\min_\theta \sum_{\tau_i \sim p(\tau)} \mathcal{L}_{\tau_i}^{(1)}\left(f_{\theta_i'}\right) = \arg\min_\theta \sum_{\tau_i \sim p(\tau)} \mathcal{L}_{\tau_i}^{(1)}\left(f_{\theta - \alpha \nabla_\theta \mathcal{L}_{\tau_i}^{(0)}(f_\theta)}\right)$$

$$\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\tau_i \sim p(\tau)} \mathcal{L}_{\tau_i}^{(1)}\left(f_{\theta - \alpha \nabla_\theta \mathcal{L}_{\tau_i}^{(0)}(f_o)}\right)$$

https://lilianweng.github.io/lil-log/2018/11/30/meta-learning.html

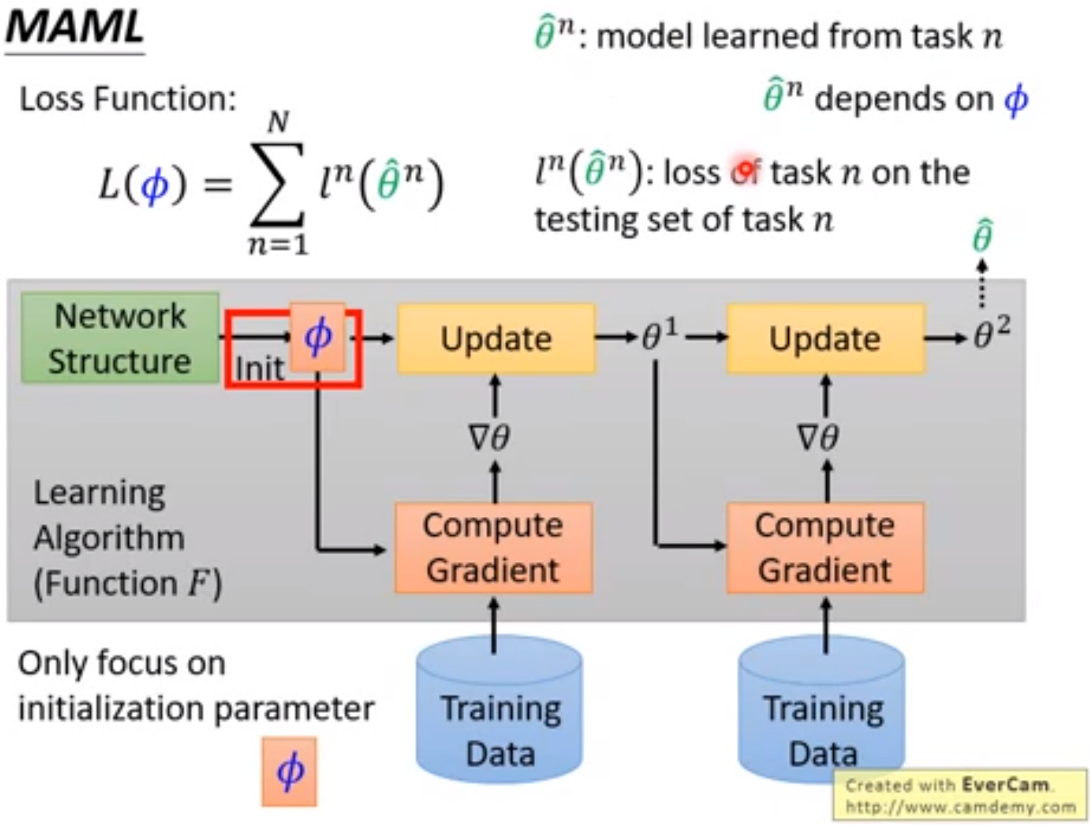
湾湾李老师网课：

Meta Learning 就是找到Learning Algorithm F，这个 F 的能力是可以找到最优预测函数 f^*.

$$F(D_{train}) = f^*$$

Meta Learning 就是找到最好的 Learning Algorithm，同样也可以有$loss$.

以前的Initialization是从一个分布中采样，现在就要学习一个Initialization.



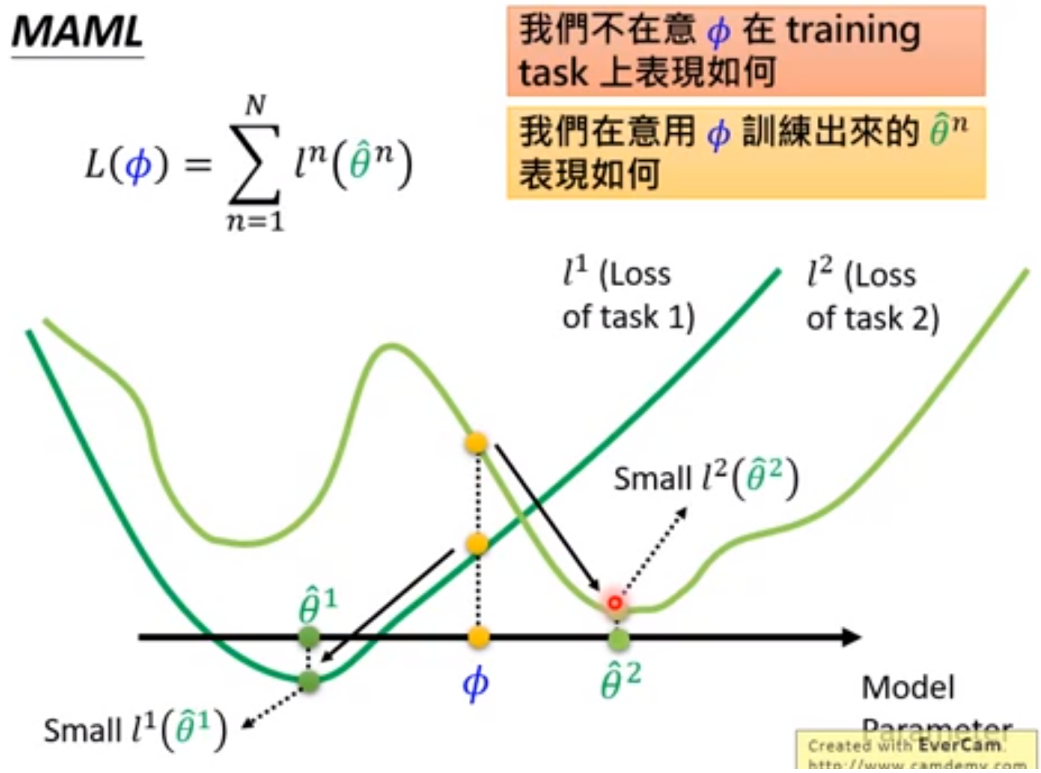
各变量解释看上图. 每一个小任务都是训练的过程.

- 与 Model Pre-training 的区别:

  Model Pre-training 损失函数的自变量是超参数(这里就是ϕ)，但是MAML的损失函数自变量是 经过 F 学习得到的 f 的最优参数在test set上的损失.

  Model Pre-training 关注的是即刻的 ϕ 给进去loss如何，而 MAML 关注的是经过训练后怎么样.

  MAML 并不在意Initialization: ϕ 在 training Task(set) 上的表现，而是经过 ϕ 训练出来的 $f^*/\hat{\theta}$ 的表现. | 即对于学习器的loss(上面有): $l^i(\phi)$ 可能不是很好，但是经过这个 ϕ 训练过后的 $\hat{\theta}$ 也就是 $l^i(\hat{\theta})$ 可能很好. 不同Task的$\hat{\theta}$当然不同:



- 训练细节：

  在训练的时候希望 Gradient Update 一次就到最优，但是测试时可以Update多次.

  Few-Shot Learning 就希望只Update一次.

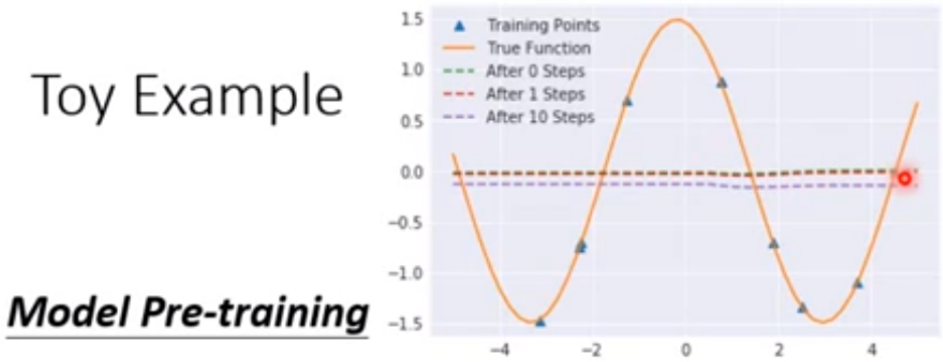
- Toy Example:

  有一个 y，一个Task就是从 y 采样出不同的 x，通过这些采样的来估测y的形式.

  $$y = a\sin(x + b)$$

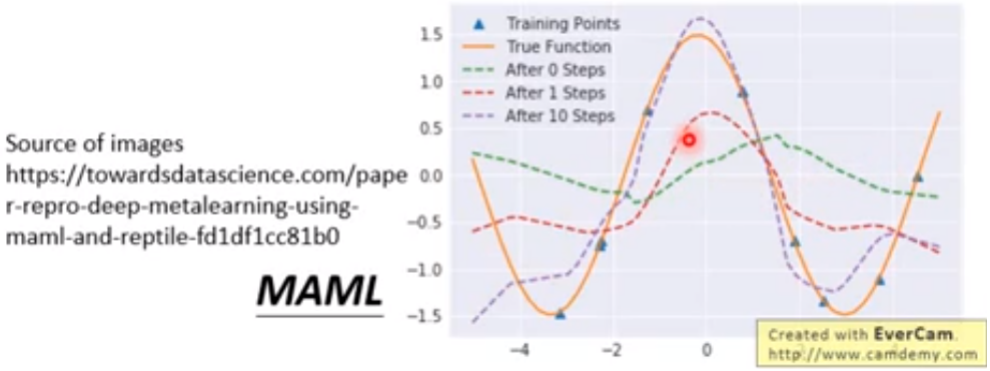
  不同的Task：选不同的 a, b 即可.

  - Model Pre-training 的结果：

    因为拿 ϕ 要在所有 y 上都表现好，所有 y 综合就是上图.

  -



- 理论方面证明的定理与推导过程**?**
- 这个任务**/**解决方法有什么意义**?**
- 对论文的讨论**/**感想**?**

- **One-shot Learning with Memory-Augmented Neural Networks**, (2016), *Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, Timothy Lillicrap*. [pdf] [code]

MANN

- [ ] **MANN** (2016) [paper] [code, code (tf)]
    - Santoro et al. "One-shot Learning with Memory-Augmented Neural Networks"

核心在哪**?**	精读**?** 代码**?**	关键词**?**	亮点**?**	阅读时间**?**

- 背景**?** 提出了什么问题**?**
- 为了解决此问题提出了什么具体的**idea?**
- 如何从该**idea**形式化地对问题建模、简化并解决的**?**
- 理论方面证明的定理与推导过程**?**
- 这个任务**/**解决方法有什么意义**?**
- 对论文的讨论**/**感想**?**

PFA

- [x] **PFA** (CVPR 2018) [paperswithcode]
    - Qiao et al. "Few-Shot Image Recognition by Predicting Parameters from Activations"

核心在哪**?**	精读**?** 代码**?**	关键词**?**	亮点**?**	笔记时间**?**
2. Model	精读	激活值 → 分类器权值的预测	采样，线性近似，混合策略	2020-10-3

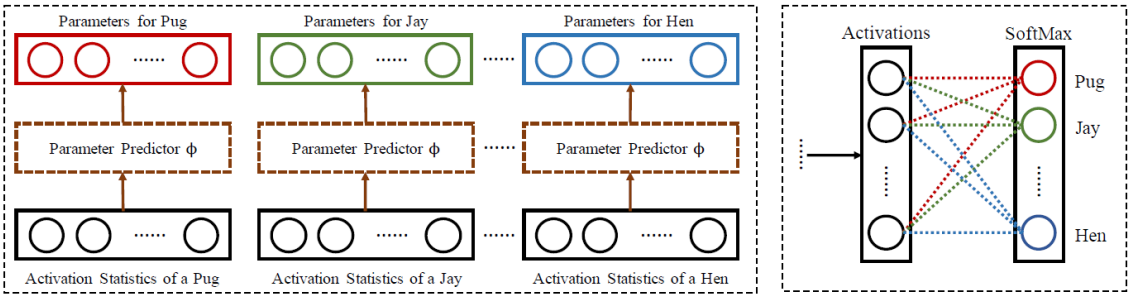


Figure 3: Building the fully connected layer by parameter prediction from activation statistics.

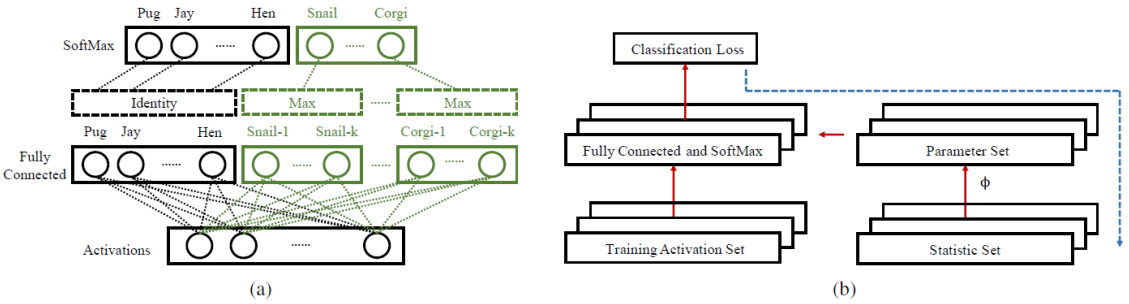


Figure 4: Illustration of the novel category adaption (a) and the training strategies for parameter predictor ϕ (b). (b): red and solid arrows show the feedforward data flow, while blue and dashed arrow shows the backward gradient flow.

- 背景? 提出了什么问题?

> We argue that a good classifier should **have the following properties:**
>
> **(1)** It achieves reasonable performance on $\mathcal{C}_{\text{few}}$.
> **(2)** Adapting to $\mathcal{C}_{\text{few}}$ does not degrade the performance on $\mathcal{C}_{\text{large}}$ significantly (if any).
> **(3)** It is fast in inference and adapts to few-shot categories with little or zero training, i.e., an efficient lifelong learning system.
>
> However, due to the limited number of samples in $\mathcal{D}_{\text{few}}$ and the **imbalance** between $\mathcal{D}_{\text{large}}$ and $\mathcal{D}_{\text{few}}$, parametric models usually **fail to learn well from the training samples.**
>
> On the other hand, many **non-parametric** approaches such as nearest neighbors can adapt to the novel concepts easily without severely forgetting the original classes. **But this requires careful designs of the distance metrics,** which can be difficult and sometimes empirical. To remedy this, some previous work instead adapts feature representation to the metrics by using Siamese networks.
>
> As we will show later through experiments, these methods do not fully satisfy the properties mentioned above.

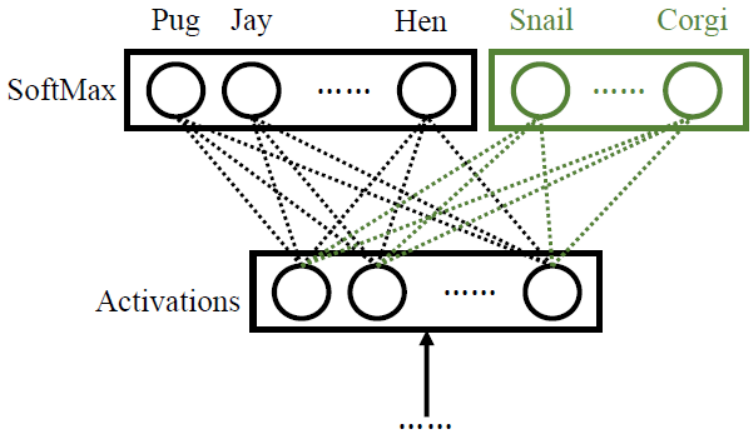
$\mathcal{D}_{\text{large}}$ 就是support set, 对应的 $\mathcal{D}_{\text{few}}$ 是query set.

  - how to **re-parameterize** the last fully connected layer to include the novel categories under the few-shot constraints, *i.e.*, for each category in $\mathcal{C}_{few}$ we have only a few examples.

we believe **the activations and the parameters** have similar **local and global structure** in their respective space.

$\mathcal{D}_{\text{large}}$ 上学到的, 泛化到 $\mathcal{C}_{\text{few}}$.

- 为了解决此问题提出了什么具体的**idea?**



上图是最后的全连接层和softmax层.

- $\mathbf{w}_y \in \mathbb{R}^n$ 是全连接层对应类别 y 的参数.
- $\mathbf{a}(x) \in \mathbb{R}^n$ 是图像 x 的 在全连接层之前的 激活层值.

> how to **re-parameterize** the last fully connected layer to include the novel categories under the few-shot constraints, *i.e.*, for each category in $\mathcal{C}_{few}$ we have only a few examples.

解决上述挑战就是:

使用激活值直接预测 $\mathbf{w}_y$, *i.e.*,
$\mathcal{A}_y = \{\mathbf{a}(x) \mid x \in \mathcal{D}_{\text{large}} \cup \mathcal{D}_{\text{few}}, Y(x) = y\}$, where $Y(\cdot)$ denotes the category of the image.

- $\overline{\mathbf{a}}_y \in \mathbb{R}^n$ 是 $\mathcal{A}_y$ 中所有元素的均值.

直觉上, 想让 $\mathbf{w}_y \cdot \mathbf{a}_y$ 尽可能大.

- 如何从该**idea**形式化地对问题建模、简化并解决的?

  直觉上, 想让 $\mathbf{w}_y \cdot \mathbf{a}_y$ 尽可能大.

> To verify this intuition, we use **t-SNE** to **visualize the neighbor embeddings** of the activation statistic $\overline{\mathbf{a}}_y$ and the parameters $\mathbf{w}_y$ for each category of a pre-trained deep neural network, as shown in Figure 2. Comparing them and we observe a high similarity **in both the local and the global structures.** More importantly, **the semantic structures** are also **preserved** in both activations and parameters, indicating **a promising generalizability to unseen categories.**

依据: 通过**t-SNE**将激活值可视化后, $\overline{\mathbf{a}}_y$ 与对应类别的分类权重 w_y 非常接近.

使用 the semantic structures. 学到一个好的 **category-agnostic mapping**: from the activations to the parameters given a good feature extractor $a(\cdot)$. 这个mapping由神经网络来学.

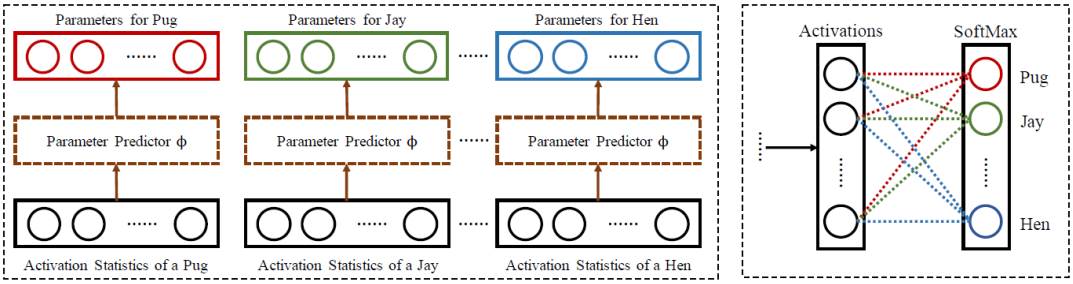


Figure 3: Building the fully connected layer by parameter prediction from activation statistics.

$$\text{parameter predictor } \phi : \overline{\mathbf{a}}_y \rightarrow \mathbf{w}_y$$

- 学习过程：ϕ 在 $y \in \mathcal{C}_{large}$ 就是 support set

$$\mathcal{L}(\phi) = \sum_{(y,x)\in\mathcal{D}_{\text{large}}} \left[-\phi\left(\bar{\mathbf{a}}_y\right)\mathbf{a}(x) + \log \sum_{y'\in\mathcal{C}_{\text{large}}} e^{\phi(\bar{\mathbf{a}}_{y'})\mathbf{a}(x)} \right] + \lambda\|\phi\|$$

但是，因为query set上可能每一类只有一个，这样 $\bar{\mathbf{a}}_y$ 可能不具有代表性了，解决方法就是采样：

one-shot，构建新的集合，并从中取一个 统计量：

$$\mathbf{s}_y \in \mathcal{A}_y \cup \bar{\mathbf{a}}_y$$

关键：以 p_{mean} 的概率取 $\bar{\mathbf{a}}_y$，反之取 $\mathcal{A}_y$ 中的元素.

$S_{large} = \left\{\mathbf{s}_1, \ldots, \mathbf{s}_{|\mathcal{C}_{\text{large}}|}\right\}$

新的损失函数：

$$\mathcal{L}(\phi) = \sum_{(y,x)\in\mathcal{D}_{\text{large}}} \mathbb{E}_{S_{\text{large}}} \left[-\phi\left(\mathbf{s}_y\right)\mathbf{a}(x) + \log \sum_{y'\in\mathcal{C}_{\text{large}}} e^{\phi(\mathbf{s}_{y'})\mathbf{a}(x)} \right] + \lambda\|\phi\|$$

以上，models the sampling from one-shot and mean activations.

- 后验：

$$P(y \mid x) = e^{\mathbb{E}_S[\phi(\mathbf{s}_y)\mathbf{a}(x)]} \Big/ \left(\sum_{y'\in\mathcal{C}} e^{\mathbb{E}_S[\phi(\mathbf{s}_{y'})\mathbf{a}(x)]} \right)$$

为了计算快速，简化 ϕ 为线性映射：

$$\begin{aligned} P(y \mid x) &= e^{\mathbf{a}(x)\cdot\phi(\mathbb{E}_S[\mathbf{s}_y])} \Big/ \left(\sum_{y'\in\mathcal{C}} e^{\mathbf{a}(x)\cdot\phi(\mathbb{E}_S[\mathbf{s}_{y'}])} \right) \\ &= e^{\mathbf{a}(x)\cdot\Phi\cdot\mathbb{E}_S[\mathbf{s}_y]} \Big/ \left(\sum_{y'\in\mathcal{C}} e^{\mathbf{a}(x)\cdot\Phi\cdot\mathbb{E}_S[\mathbf{s}_{y'}]} \right) \end{aligned}$$

上述简化(直接线性的方法) 和 采样的方法结合使用.

其中 $\mathbb{E}_S[s_y]$ 可以在训练时提前计算好并保存下来. 在小样本数据集上进行训练时，对于新的类别需要更新 $\mathbb{E}_S[s_y]$，但由于小样本数量较少，计算得到的$\mathbb{E}_S[s_y]$可靠性不高，采用了一种混合策略：

- 对于大规模数据集中的类别，直接使用训练得到的 $\mathbb{E}_S[s_y]$ 简化的线性近似.
- 对于小样本数据集中的类别，使用类别 y 对应的所有统计值 s_y 与 query set图像 x' 的激活值 $a(x')$ 内积的最大值，来作为分类器的输出. 该过程如下所示：

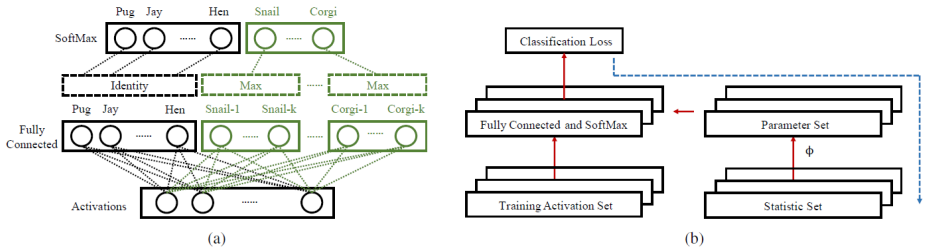


Figure 4: Illustration of the novel category adaption (a) and the training strategies for parameter predictor ϕ (b). (b): red and solid arrows show the feedforward data flow, while blue and dashed arrow shows the backward gradient flow.

> **(4a)** For each novel category in C_{few}, we compute the **maximal** response of the activation of the test image to **the parameter set predicted from each activation in the statistic set of the corresponding novel category in** C_{few}.
>
> We use them as the inputs to the SoftMax layer to compute the probabilities.

- 总结：

利用一个网络 ϕ，根据每类样本的激活向量平均值 $\bar{\mathbf{a}}_y$（对于 query set 中的元素有采样），来预测分类器中全连接层的权重.

- 理论方面证明的定理与推导过程?
- 这个任务/解决方法有什么意义?
- 对论文的讨论/感想?

  如果把线性变换矩阵 Φ 设为单位矩阵，那么:

  $$a(x) \cdot \Phi \cdot \mathbb{E}_S\left(s_y\right) \Rightarrow a(x) \cdot \mathbb{E}_S\left(s_y\right)$$

  相当于计算 $a(x)$ 和 $\mathbb{E}_S\left(s_y\right)$ 之间的余弦相似性，对于类别表征（或者说原型 prototype）的计算中增加了一个随机采样的过程和一个可学习的线性变换 ϕ，当然如果不把 ϕ 近似为线性变换，其性能可能会更高，但计算复杂度也会大幅上升.

TADAM (难)

- [x] **TADAM** (NIPS 2018) [paperswithcode]
  - Boris N. Oreshkin et al. "TADAM: Task dependent adaptive metric for improved few-shot learning"

核心在哪?	精读? 代码?	关键词?	亮点?	笔记时间?
	精读		1. scale metric, 2. TEN结构, 3. 辅助任务合作训练.	2020-10-3

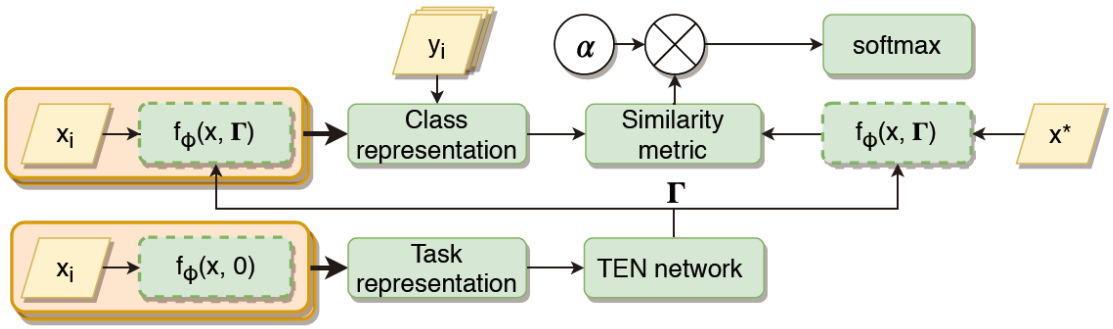
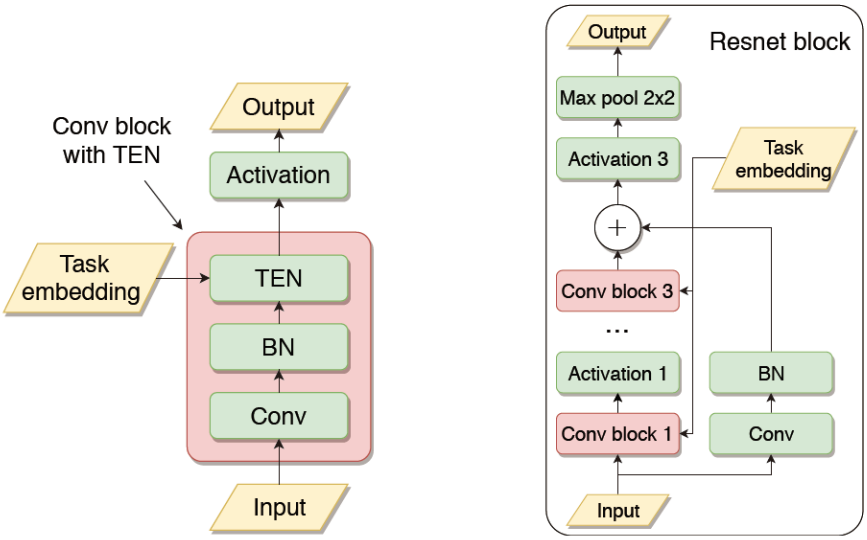


Figure 1: Proposed few-shot architecture. Blocks with shared parameters have dashed border.



(a) Convolutional block with TEN. (b) Resnet block with TEN.

Figure 1: Components of the ResNet-12 feature extractor.

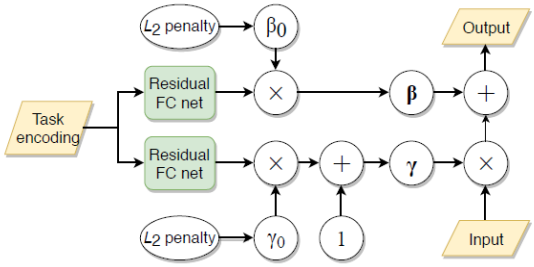


Figure 2: Architecture of the TEN block.

- **背景？提出了什么问题？**

  学到一个context的度量空间，即学习一个合适的相似性度量，提取输入数据的特征后，将其映射到similarity space.

  similarity metric 和 cost function 之间的 non-trivial interaction 可以通过 scaling 来提高性能.

- **为了解决此问题提出了什么具体的idea？**

  metric scaling and metric task conditioning

  - M-shot, K-way.
  - **feature extractor**:

    $$f_\phi : \mathbb{R}^{D_x} \rightarrow \mathbb{R}^{D_z}$$

    这里 $f_\phi(x)$ 可以被直接用来解决问题（比如matching, prototype就是这么做的）.
  - **similarity measure** $d : \mathbb{R}^{D_z \times D_z} \rightarrow \mathbb{R}$

    d 并不满足度量的基本性质（non-negativity, symmetry, subadditivity）.

1.2 Summary of contributions

Metric Scaling: To our knowledge, this is the first study to (i) propose metric scaling to improve performance of few-shot algorithms, (ii) mathematically analyze its effects on objective function updates and (iii) empirically demonstrate its positive effects on few-shot performance.

Task Conditioning: We use a task encoding network to extract a task representation based on the task's *sample* set. This is used to influence the behavior of the feature extractor through FILM [19].

Auxiliary task co-training: We show that co-training the feature extraction on a conventional supervised classification task reduces training complexity and provides better generalization.

- 采用 metric scaling，并数学上分析.
- task encoding network.
- co-training the feature extraction on a conventional supervised classification task.

- 如何从该**idea**形式化地对问题建模、简化并解决的**?**
  - **model**:

    1 scale the distance metric，乘一个 α，$\alpha\, d(\cdot,\cdot)$.

    在计算完距离度量后学习一个scaling factor.

    此时该metric的学习过程，the class-wise cross-entropy loss function:

$$J_k(\phi,\alpha)=\sum_{\mathbf{x}_i\in\mathcal{Q}_k}\left[\alpha d\left(f_\phi\left(\mathbf{x}_i\right),\mathbf{c}_k\right)+\log\sum_j\exp(-\alpha d\left(f_\phi\left(\mathbf{x}_i\right),\mathbf{c}_j\right))\right]$$

    其中 $\mathcal{Q}_k$ 是query set，第 k 类.

    对 ϕ 求导，(就是内函数外函数导数)有 注意 α 提到外面了:

$$\frac{\partial}{\partial\phi}J_k(\phi,\alpha)=\alpha\sum_{\mathbf{x}_i\in\mathcal{Q}_k}\left[\frac{\partial}{\partial\phi}d\left(f_\phi\left(\mathbf{x}_i\right),\mathbf{c}_k\right)-\frac{\sum_j\exp(-\alpha d\left(f_\phi\left(\mathbf{x}_i\right),\mathbf{c}_j\right))\frac{\partial}{\partial\phi}d\left(f_\phi\left(\mathbf{x}_i\right),\mathbf{c}_j\right)}{\sum_j\exp(-\alpha d\left(f_\phi\left(\mathbf{x}_i\right),\mathbf{c}_j\right))}\right]$$

    上式:

    直观上，α 有两个影响:

    1. 梯度的全局放缩.
    2. 增加 RHS 括号内第二项权重.

    Lemma 1 (Metric scaling). *If the following assumptions hold:*

    $\mathcal{A}_1: d(f_\phi(\mathbf{x}),\mathbf{c}_k)\neq d(f_\phi(\mathbf{x}'),\mathbf{c}_k),\forall k,\mathbf{x}\neq\mathbf{x}'\in\mathcal{Q}_k;\quad \mathcal{A}_2:\left|\frac{\partial}{\partial\phi}d(f_\phi(\mathbf{x}),\mathbf{c})\right|<\infty,\forall\mathbf{x},\mathbf{c},\phi,$

    then it is true that:

$$\lim_{\alpha\to0}\frac{1}{\alpha}\frac{\partial}{\partial\phi}J_k(\phi,\alpha)=\sum_{\mathbf{x}_i\in\mathcal{Q}_k}\left[\frac{K-1}{K}\frac{\partial}{\partial\phi}d(f_\phi(\mathbf{x}_i),\mathbf{c}_k)-\frac{1}{K}\sum_{j\neq k}\frac{\partial}{\partial\phi}d(f_\phi(\mathbf{x}_i),\mathbf{c}_j)\right],\tag{3}$$

$$\lim_{\alpha\to\infty}\frac{1}{\alpha}\frac{\partial}{\partial\phi}J_k(\phi,\alpha)=\sum_{\mathbf{x}_i\in\mathcal{Q}_k}\left[\frac{\partial}{\partial\phi}d(f_\phi(\mathbf{x}_i),\mathbf{c}_k)-\frac{\partial}{\partial\phi}d(f_\phi(\mathbf{x}_i),\mathbf{c}_{j_i^*})\right];\tag{4}$$

    where $j_i^*=\arg\min_j d(f_\phi(\mathbf{x}_i),\mathbf{c}_j)$.

    Proof. Please refer to Appendix A. □

    - $\alpha\to0$ 时，注意(3)式 RHS两项的正负号:
      - 第一项 $\frac{K-1}{K}\frac{\partial}{\partial\phi}d\left(f_\phi\left(\mathbf{x}_i\right),\mathbf{c}_k\right)$:

        最小化 和第 k 类 prototype 的距离.
      - 第二项负号，所以是最大化和其他类的prototype.
    - $\alpha\to\infty$ 时，注意RHS:
      - 第一项一样.
      - 第二项是最大化 那个最接近的错误类的prototype.

    2 Task conditioning

    特征提取 f_ϕ 应是task-independent. 通过任务的样例集来提取任务表示.

$$f_\phi(\mathbf{x},\Gamma)$$

其中task representation的参数表达. is related to **the FILM conditioning layer** [19] and **conditional batch normalization** [3,18] of the form $h_{\ell+1} = \gamma \odot h_\ell + \beta$, where γ and β are **scaling and shift** vectors applied to the layer h_ℓ.

使用the mean of the class prototypes作为任务上的prototype.

- **task embedding network (TEN):**

  predict layer-level element-wise scale and shift vectors γ, β for each convolutional layer in the feature extractor.

  Our implementation of the TEN (see Supplementary Materials, Section S1 for more details) uses two separate fully connected residual networks to generate vectors $\boldsymbol{\gamma}, \boldsymbol{\beta}$. Following the terminology in [18], the $\boldsymbol{\gamma}$ parameter is learned in the delta regime, *i.e.* predicting deviation from unity. The most critical component in being able to successfully train the TEN was the addition of the scalar L_2 penalized post-multipliers γ_0 and β_0. They limit the effect of $\boldsymbol{\gamma}$ (and $\boldsymbol{\beta}$) by encoding a prior belief that all components of $\boldsymbol{\gamma}$ (and $\boldsymbol{\beta}$) should be simultaneously close to zero for a given layer unless task conditioning provides a significant information gain for this layer. Mathematically, this can be expressed as $\boldsymbol{\beta} = \beta_0 g_\theta(\overline{\mathbf{c}})$ and $\boldsymbol{\gamma} = \gamma_0 h_\varphi(\overline{\mathbf{c}}) + 1$, where g_θ and h_φ are predictors of $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$.

  仔细看才看得懂.

1) 通过Auxiliary task co-training的训练方式训练feature extractor, 为support set和query set中的样例抽取特征得到class representation. 其中feature extractor中采用的是ResNet-12结构.

2) 借鉴Prototypical Network的思想, 将每类的样例得到的向量表示求平均得到每一类的原型, 随后使用类原型的平均值作为task representation, 将这个任务表示作为输入, 到TEN network中, 然后根据它的数据更新feature extractor提取的特征, 将任务特有的特征与样例提取的特征相结合, 使得support set和query set的class representation更具有泛化性.

3) 根据上一步的class representation计算similarity metric, 随后乘一个可学习的系数 α 来缩放距离度量, 增强模型的可适性. 最后将这步输出投入到softmax中得到图片的最终分类.

- 理论方面证明的定理与推导过程?
  - *Lemma 1*, Appendix A.
- 这个任务/解决方法有什么意义?
- 对论文的讨论/感想?