

# 极客大学机器学习训练营

## 神经网络训练方法

王然

众微科技 AI Lab 负责人

二〇二一年四月七日

1 基本训练方法

2 预训练方法

3 半监督学习

4 参考文献

1 基本训练方法

2 预训练方法

3 半监督学习

4 参考文献

- ▶ 在本讲中我们只关心有监督学习和相关内容；我们不讨论生成模型和强化学习模型的训练；
- ▶ 即使如此，深度学习的训练也比传统模型的训练要困难；
- ▶ 原因：
  - ▶ 训练时间长；
  - ▶ 训练稳定性差；
  - ▶ 训练方法繁多；
  - ▶ 训练透明性差；
  - ▶ ...

- ▶ 基本训练 Trick;
- ▶ 预训练相关主题;
- ▶ 半监督学习相关主题。

- ▶ 传统优化器：SGD (+Momentum) 或 Adam；
- ▶ 其他优化器：AdamW/RAadam/Lookahead/Lars/Lamb；
- ▶ PyTorch 实现样例

- ▶ 两种情况：有权重、无权重；
- ▶ 有权重：学习率多半需要很低，增加复杂上层网络的时候要注意对于已经训练好的权重和随机初始化的权重采用不同的学习率；后者实现时注意 train 和 eval 的正确选择；
- ▶ 无权重：Warmup；Reduce on Plateau；Cosine Annealing 和 Snapshot Ensemble；一些例子见 [PyTorch LR](#)；

- ▶ 一般来说，当模型快进入到收敛的阶段时候，需要减少模型的变化程度（进行微调）；
- ▶ 其他目的，减少算力消耗；
- ▶ 加大 Batch Size；
- ▶ 选用更慢的优化器（SGD+Momentum/Lookahead+SGD+Momentum）；



- ▶ 在一些情况下，自然的损失函数未必是最有效的；
- ▶ 其他损失函数可以在不同阶段进行引入，在一些时候，过早的引入损失函数只会使得训练更加困难；
- ▶ 一些有用的损失函数：Focal Loss；Label Smoothing；Temperature in Softmax
- ▶ 其他的函数我们会在后面提到。

- ▶ 不同的初始化 Ensemble 效果更好;
- ▶ 一些文章表明增加惩罚使得不同网络的权重不一样会更好。

- ▶ 一般应用于 CV 和 NLP 数据当中；
- ▶ 是比赛中最重要的手段；
- ▶ 两种数据扩充方式：增加合理的样本、增加噪声；
- ▶ 不宜过多、过多会实际导致模型记住观测；
- ▶ Prediction Time Augmentation；
- ▶ 不同数据扩充方法可以进行 ensemble。

- ▶ 一般应用与 CV 和 NLP 数据当中；
- ▶ 将不同 encoder 进行（两两）拼接并训练；
- ▶ 增加 ensemble 多样性；
- ▶ 一般来说 encoder 架构/训练方式越不同，带来的增益越大。

1 基本训练方法

2 预训练方法

3 半监督学习

4 参考文献

- ▶ 一些任务具备基本的通用性质：例如图像识别可以认为是（其他）CV 任务的基础，而语言模型的训练可以认为是 NLP 各项任务的基础；
- ▶ 我们关注的任务可能数据集并不够大；
- ▶ 使用在其他任务上训练好的模型，针对当前模型进行 finetune。

- ▶ 预训练；
- ▶ Encoding Refinement。



- ▶ 选择一个非常典型且观测量巨大的数据集进行训练；
- ▶ 典型例子：ImageNet；
- ▶ 对于 CV 来说，不同的网络架构对于细节的捕捉程度不一样，所以需要尝试不同的组合；
- ▶ 问题：难以找到合适的数据集、任务不具有代表性；



## 预训练方式 2：多任务学习（思考题）

如果单任务学习的任务不够代表性，是否可以采取多个任务呢？

- ▶ 多任务学习经常导致模型难以应付不同任务的需求 (Negative Transfer);
- ▶ 解决 Negative Transfer 的方法：
  - ▶ Soft Sharing;
  - ▶ Mixture Model (Fedus, Zoph, and Shazeer 2021)

- ▶ 采用无监督的方式进行训练;
- ▶ 思想: 使用网络将输入进行“还原”。

- ▶ 沙漏型的网络；
- ▶ 对输入增加随机噪声，要求网络恢复原始的输入；
- ▶ 应用时注意将中间的多层提出。

见Doersch (2016)。

- ▶ 主要应用于 CV 的生成任务；
- ▶ 虽然数学形式很优美，但是由于其比较复杂，实际效果不一定好；
- ▶ 重要思路及其衍生：
  - ▶ ELBO，例如 (Hafner et al. 2019)；
  - ▶ 随机输入的处理，例如 Jaegle et al. (2021)；

- ▶ 目前最火的预训练框架之一；
- ▶ 主要来源于预训练语言模型；
- ▶ 在多模态和 CV 当中也开始有应用。

- ▶ 核心思想：如果两个字/词周围的语境类似，则该字/词表示语义也接近；
- ▶ 数学上来讲  $P(o|c) = \frac{\exp(v_o^t v_c)}{\sum_{w \in V} \exp(v_w^t v_c)}$ ；
- ▶ Negative Sampling：由于可选文字很多，所以只抽取部分不属于上下文的字/词；数学形式  $\log \sigma(v_o^t v_c) - \sum_i \log \sigma(-v_i^t v_c)$ ，其中求和针对的是 Negative Sample；
- ▶ 对于结构化数据来说，在有时序的情况下可以考虑使用这种方法；
- ▶ 一系列来源于 DeepWalk(Perozzi, Al-Rfou, and Skiena 2014) 的方法将这种思路应用在了图网络当中。



- ▶ 使用 Masked Language Model 作为目标，使用 Transformer 作为网络结构；
- ▶ 加入 [CLS] token；
- ▶ 还有很多其他的提升，但是核心思路都类似。

- ▶ 一些研究引入了多任务的学习方法作为对原始模型的 refinement, 由于细节十分模糊, 并不清楚这些 paper 的实际训练过程;
- ▶ 比较有价值的应用为 T5(Raffel et al. 2019),
  - ▶ 核心思路: 将所有问题转换成 seq2seq 问题;
- ▶ 这些思路的一个重要应用是将结构化数据转换成文本数据后进行编码。

- ▶ BART(Lewis et al. 2019);
- ▶ Electra (Clark et al. 2020);
- ▶ 这些方法都十分有挑战性，所以更适合于 encoder refinement。

- ▶ 见 Visual Bert 及相关研究 (Li et al. 2019);
- ▶ 目前类似方法的效果非常不稳定, Ensemble 常常可以达到极好的效果;
- ▶ 其他可以见 Wu et al. (2020)。

- ▶ 在大部分时候，以上方法均可以应用在 Encoder Refinement 当中；
- ▶ 通过对 Encoder 进一步提升（而不是对上层模型进行控制），是深度学习训练中非常独特的方法，也是很多比赛/SOTA 方法的核心；
- ▶ 注意：在进行 Encoder Refinement 的过程中，如果我们进行过多轮数的训练，则 Encoder 会丢失之前的所有信息；所以建议只做轮数非常少的训练。

## 1 基本训练方法

## 2 预训练方法

## 3 半监督学习

■ Consistency Regularization 和 Adversarial Training ■ Triplet Mining 及 Contrast Learning ■ Pseudo Label

## 4 参考文献

- ▶ 在本节当中，我们将以半监督学习为提纲，介绍若干比较先进的方法；
- ▶ 当时这些方法可以进行单独使用；
- ▶ （部分）实践表明，一般这些方法在训练的后期使用可以有很好的效果；
- ▶ 建议：将这些方法写成一套方法论并进行实验（在训练最开始和训练结束后）。



- ▶ 数据分为两部分：
  - ▶ 包含解释变量和目标变量（有标签）；
  - ▶ 只包含解释变量不包含目标变量（无标签）；
- ▶ 核心问题：如何应用不包含目标变量的数据。



- ▶ 核心思想：对于等价的数据扩充（例如在训练过程中不改变数据的 Label），其得到的表征应该尽可能接近；
- ▶ 由于数据扩充本身并不要求知道**真实的标签**，所以可以利用无标签的样本；
- ▶ PyTorch 库可见[PyTorch Consistency Regularization](#)；
- ▶ UDA (Xie et al. [2019](#)) 和 FixMatch (Sohn et al. [2020](#))；

见Xie et al. (2019).

见Sohn et al. (2020).

## 一个额外的（奇怪的）例子：MixedMatch

- ▶ 大部分数据扩充的结果都是 Reasonable 的；
- ▶ MixedMatch (Berthelot et al. 2019) 将多个观测进行混合；
- ▶ 只能应用于 CV 数据。

## 如果无法进行数据扩充...?

- ▶ 在结构化数据当中，大部分时候无法进行数据扩充；
- ▶ 可能的解决方案：对部分输入增加噪声（常常在 Embedding 之后）；
- ▶ 问题：增加怎样的噪声是有效的？

- ▶ Adversarial Training 本身是一个非常大的领域；其核心研究方向是神经网络的鲁棒性问题；
- ▶ 我们关注的是 Adversarial Training 其中一个很小的领域，即利用梯度信息增加神经网络鲁棒性，并借此 (?) 增加其准确性；
- ▶ PyTorch 实现可见 [Adversarial Attacks](#)；
- ▶ 在大部分时候，这类方法的提升有限（但比较稳定）；我们将会对比较基础的方法进行介绍。

见Wong, Rice, and Kolter (2020)。

- ▶ 可以单独使用，也可以结合 Consistency Regulation 使用；
- ▶ 一般不建议在最开始的时候使用；
- ▶ 严重破坏模型结构；
- ▶ 默认 Fast FGSM；



- ▶ 在 Consistency Regularization 中，我们只要求标签一样的输入输出表征接近；
- ▶ 那么对于标签不一样的呢？
- ▶ Triplet Mining 和 Contrast Learning 基本思路：不一样的标签应该表现尽可能不一样。

- ▶ 经典的 Triplet 包含 Anchor, Positive 和 Negative 三个样本, 我们表示为  $(a, p, b)$ ;
- ▶ Positive 和 Anchor 标签一样, Negative 和 Anchor 标签不同;
- ▶ 要求: Positive 的和 Anchor 的 encoding 尽可能一致, 而 Negative 则应该和 Anchor 的 encoding 尽可能不一致;
- ▶ 经典的 Triplet Loss:  $\max(d(a, p) - d(a, n) + \text{margin}, 0)$ ;
- ▶ N-pair loss:  $-\log \frac{\exp(f(x_a)^t f(x_p))}{\exp(f(x_a)^t f(x_p)) + \sum_i \exp(f(x_a)^t f(x_{n_i}))}^\circ$

- ▶ Easy Triplets:  $d(a, p) + \text{margin} < d(a, n)$ ;
- ▶ Hard Triplets:  $d(a, n) < d(a, p)$ ;
- ▶ Semi-hard Triplets:  $d(a, p) < d(a, n) < d(a, p) + \text{margin}$ ;

- ▶ 用哪些 Sample (Triplet Selection);
- ▶ 用哪些 loss,
- ▶ Online vs Offline;
- ▶ 见Xuan, Stylianou, and Pless (2020) 和Sikaroudi et al. (2020);

- ▶ 基本思路： 对一个样本做不同扩充， 并比较和其他样本的距离区别；
- ▶ Contrast Learning 可以完全基于无标签样本；
- ▶ 距离常常是习得的；
- ▶ 细节见Chen et al. (2020)。

# 提升








1 基本训练方法

2 预训练方法

3 半监督学习

4 参考文献



-  Berthelot, David et al. (2019). “Mixmatch: A holistic approach to semi-supervised learning”. In: *arXiv preprint arXiv:1905.02249*.
-  Chen, Ting et al. (2020). “A simple framework for contrastive learning of visual representations”. In: *International conference on machine learning*. PMLR, pp. 1597–1607.
-  Clark, Kevin et al. (2020). “Electra: Pre-training text encoders as discriminators rather than generators”. In: *arXiv preprint arXiv:2003.10555*.
-  Doersch, Carl (2016). “Tutorial on variational autoencoders”. In: *arXiv preprint arXiv:1606.05908*.
-  Fedus, William, Barret Zoph, and Noam Shazeer (2021). “Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity”. In: *arXiv preprint arXiv:2101.03961*.
-  Hafner, Danijar et al. (2019). “Dream to control: Learning behaviors by latent imagination”. In: *arXiv preprint arXiv:1912.01603*.
-  Jaegle, Andrew et al. (2021). “Perceiver: General Perception with Iterative Attention”. In: *arXiv preprint arXiv:2103.03206*.



Lewis, Mike et al. (2019). “Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension”. In: *arXiv preprint arXiv:1910.13461*.



Li, Liunian Harold et al. (2019). “Visualbert: A simple and performant baseline for vision and language”. In: *arXiv preprint arXiv:1908.03557*.



Perozzi, Bryan, Rami Al-Rfou, and Steven Skiena (2014). “Deepwalk: Online learning of social representations”. In: *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 701–710.



Raffel, Colin et al. (2019). “Exploring the limits of transfer learning with a unified text-to-text transformer”. In: *arXiv preprint arXiv:1910.10683*.



Sikaroudi, Milad et al. (2020). “Offline versus online triplet mining based on extreme distances of histopathology patches”. In: *International Symposium on Visual Computing*. Springer, pp. 333–345.



Sohn, Kihyuk et al. (2020). “Fixmatch: Simplifying semi-supervised learning with consistency and confidence”. In: *arXiv preprint arXiv:2001.07685*.



Wong, Eric, Leslie Rice, and J Zico Kolter (2020). “Fast is better than free: Revisiting adversarial training”. In: *arXiv preprint arXiv:2001.03994*.



Wu, Bichen et al. (2020). “Visual transformers: Token-based image representation and processing for computer vision”. In: *arXiv preprint arXiv:2006.03677*.



Xie, Qizhe et al. (2019). “Unsupervised data augmentation for consistency training”. In: *arXiv preprint arXiv:1904.12848*.



Xuan, Hong, Abby Stylianou, and Robert Pless (2020). “Improved embeddings with easy positive triplet mining”. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 2474–2482.