Bag of tricks for Image Classification with CNN

Large-batch training

1. **Linear scaling learning rate**
   - e.g. ResNet-50 SGD 256 batch size 0.1 learning rate
   - init learning $rate = 0.1 * b/256$. where b is the batch size
2. **Learning rate warm up**
   - at the beginning, paras are far from the final solution
   - e.g. we use first m batch to warm up, the init learning rate is η, at the i batch where $1 \leq i \leq m$, set the learning rate to be $i\eta/m$
3. **Zero γ**
   - Batch Normalization: $\gamma \hat{x} + \beta$ **Normally**, both elements γ and β are initialized to 1s and 0s
   - Instead of setting them in a normal way, it set it as $\gamma = 0$ to all BN layers that **sit at the end of the residual block (最后一层residual block的BN层)**.
   - **easy to train at the initial stage**
4. **No bias decay**
   - Weight decay will apply to both weight and bias
   - it recommended that only apply to weight regularization to avoid overfitting. BN parameters are left unregularized

Low-precision training (降低位数)

1. Normal setting: 32-bit floating point (FP32) precision
2. Trick switching it to larger batch size (1024) with FP16 and get higher accuracy

Model Tweaks

ResNet Architecture

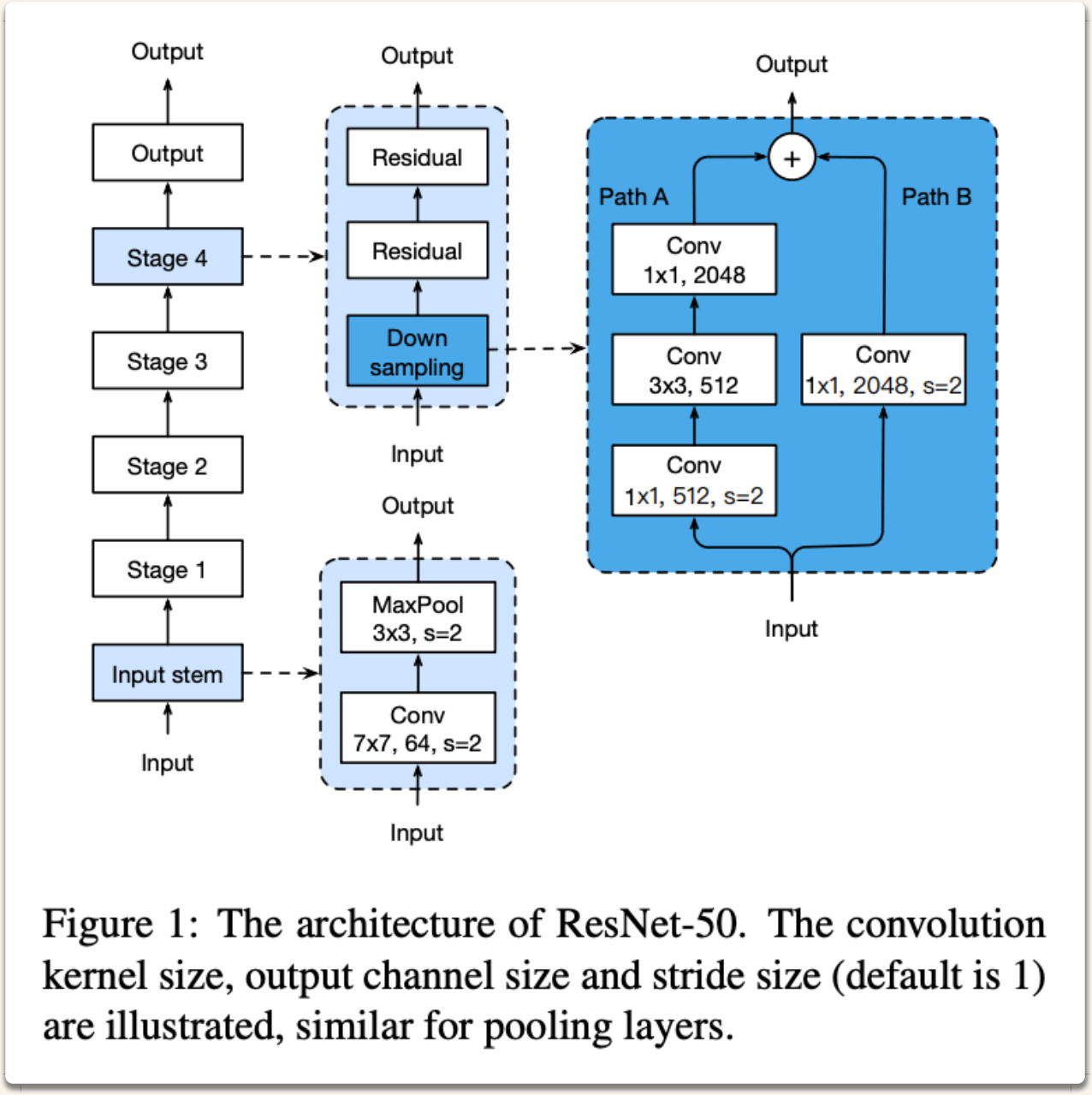
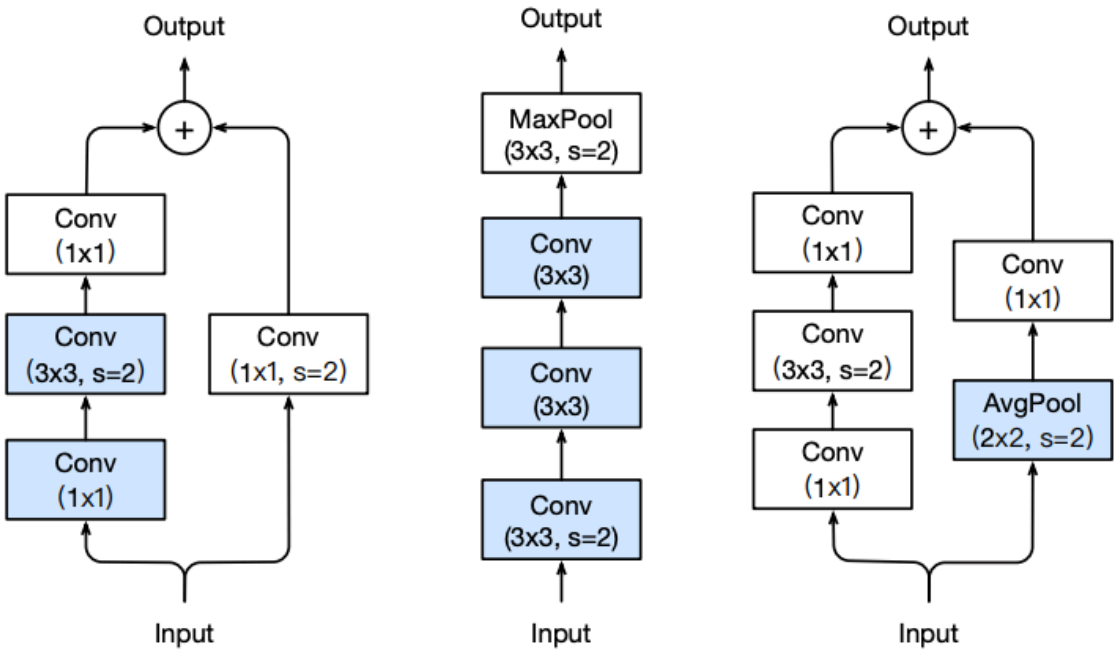


Figure 1: The architecture of ResNet-50. The convolution kernel size, output channel size and stride size (default is 1) are illustrated, similar for pooling layers.

1. ResNet-B
    - 为了避免 1x1 conv stride=2 带来的information loss
2. ResNet-C
    - 为了避免计算量，使用两个3x3 conv代替一个7x7 conv
3. ResNet-D
    - ResNet-B中path B中的1x1 conv stride=2还是会带来信息丢失，在之前加一个avgpool stride=2 能够有效避免信息丢失



(a) ResNet-B (b) ResNet-C (c) ResNet-D

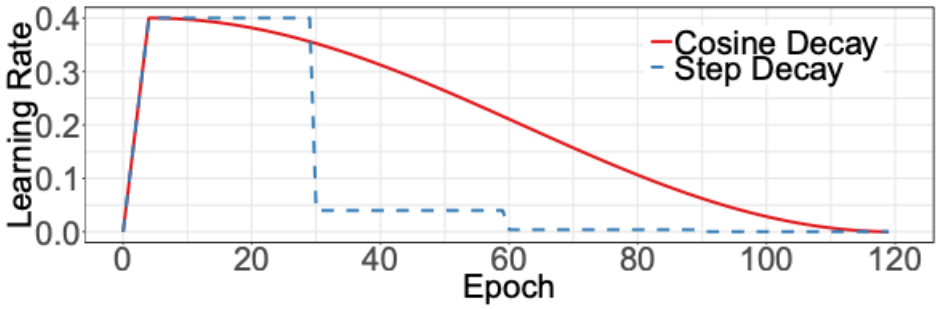
Model	#params	FLOPs	Top-1	Top-5
ResNet-50	25 M	**3.8 G**	76.21	92.97
ResNet-50-B	25 M	4.1 G	76.66	93.28
ResNet-50-C	25 M	4.3 G	76.87	93.48
ResNet-50-D	25 M	4.3 G	**77.16**	**93.52**

Table 5: Compare ResNet-50 with three model tweaks on model size, FLOPs and ImageNet validation accuracy.

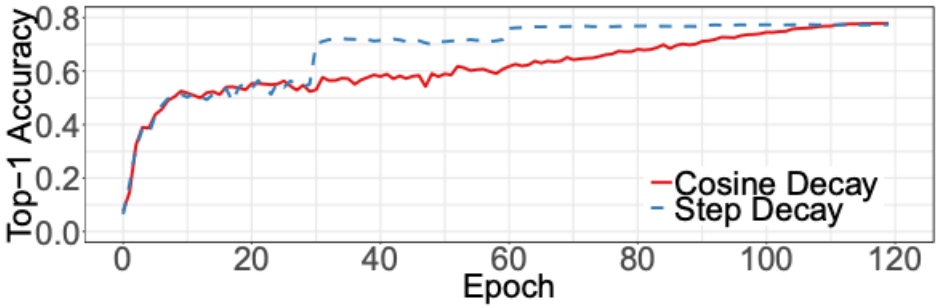
Training Refinement

1. **Cosine Learning Rate Decay**
   - $\eta_t = \frac{1}{2}(1 + cos(\frac{t\pi}{T}))\eta$
     - where T is the total number of batches (ignore warmup stage)
     - t is the current batch
     - η is the init learning rate

- potentially improve the training progress



(a) Learning Rate Schedule



(b) Validation Accuracy

Figure 3: Visualization of learning rate schedules with warm-up. Top: cosine and step schedules for batch size 1024. Bottom: Top-1 validation accuracy curve with regard to the two schedules.

2. Label Smoothing
   - 正则化方法，对于ground truth的分布进行混合。原始gt分布记为q_i，经过label smoothing之后

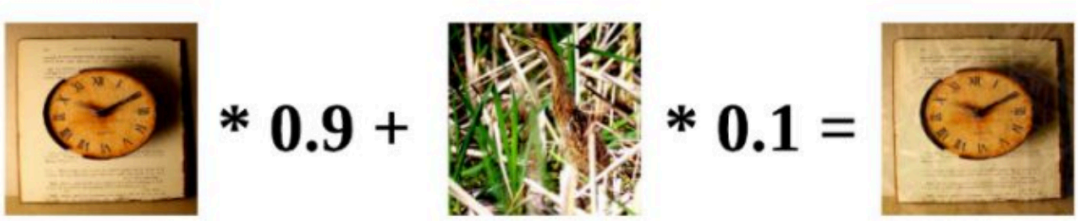
$$q_i^{`} = (1-\epsilon)q_i + \frac{\epsilon}{K}$$

   - ϵ 为常量，K为分类类别。可以减少模型对于标签的过度信赖，对于标签不够精准有较好的帮助。

3. Knowledge Distillation
   1. 训练一个复杂的网络（N1）
   2. 使用数据train N1网络并得到（M1）

3. 根据复杂网络设计一个简单网络 （N0）
4. 将M1 softmax 设T=20 预测数据得到 soft target
5. soft target 和 hard target加权得出Target （推荐0.1:0.9）
6. 使用 $label = Target$ 的数据集训练N0（T=20）得到 M0
7. 设T=1，M0 模型为我们得到的训练好的精简模型

4. **Mixup Training**



   - Data Augmentation,数据进行插值扩充
   - Weighted linear interpolation (双线性插值)
     - $x = \lambda x_i + (1-\lambda)x_j$
     - $y = \lambda y_i + (1-\lambda)y_j$
   - $\lambda \in [0,1]$ **In mixup training, we only use** (x, y)

5. **Result of Image Classification**

Refinements	ResNet-50-D		Inception-V3		MobileNet	
	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5
Efficient	77.16	93.52	77.50	93.60	71.90	90.53
+ cosine decay	77.91	93.81	78.19	94.06	72.83	91.00
+ label smoothing	78.31	94.09	78.40	94.13	72.93	91.14
+ distill w/o mixup	78.67	94.36	78.26	94.01	71.97	90.89
+ mixup w/o distill	79.15	94.58	**78.77**	**94.39**	**73.28**	**91.30**
+ distill w/ mixup	**79.29**	**94.63**	78.34	94.16	72.51	91.02

6. **Result of Object Detection**

Incremental Tricks	mAP	Δ	Cumu Δ
- data augmentation	64.26	-15.99	-15.99
baseline	80.25	0	0
+ synchronize BN	80.81	+0.56	+0.56
+ random training shapes	81.23	+0.42	+0.98
+ cosine lr schedule	81.69	+0.46	+1.44
+ class label smoothing	82.14	+0.45	+1.89
+ mixup	**83.68**	**+1.54**	**+3.43**

Table 3. Incremental trick validation results of YOLOv3, evaluated at 416×416 on Pascal VOC 2007 test set.

Incremental Tricks	mAP	Δ	Cumu Δ
- data augmentation	77.61	-0.16	-0.16
baseline	77.77	0	0
+ cosine lr schedule	79.59	+1.82	+1.82
+ class label smoothing	80.23	+0.64	+2.46
+ mixup	**81.32**	**+0.89**	**+3.55**

Table 4. Incremental trick validation results of Faster-RCNN, evaluated at 600×1000 on Pascal VOC 2007 test set.