S5E3

Publish | Edit | ⋮

Advanced Concepts

Pre-requisites are:

1. How many layers,
2. MaxPooling,
3. 1x1 Convolutions,
4. 3x3 Convolutions,
5. Receptive Field (old),
6. SoftMax,
7. Learning Rate,
8. Kernels and how do we decide the number of kernels?
9. Batch Normalization,
10. Image Normalization,
11. Position of MaxPooling,
12. Concept of Transition Layers,
13. Position of Transition Layer,
14. Number of Epochs and when to increase them,
15. DropOut
16. When do we introduce DropOut, or when do we know we have some overfitting
17. The distance of MaxPooling from Prediction,
18. The distance of Batch Normalization from Prediction,
19. When do we stop convolutions and go ahead with a larger kernel or some other alternative (which we have not yet covered)
20. How do we know our network is not going well, comparatively, very early
21. Batch Size, and effects of batch size
22. When to add validation checks
23. LR schedule and concept behind it
24. Adam vs SGD

BATCH NORMALIZATION

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1...m}\}$;
Parameters to be learned: γ, β
Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

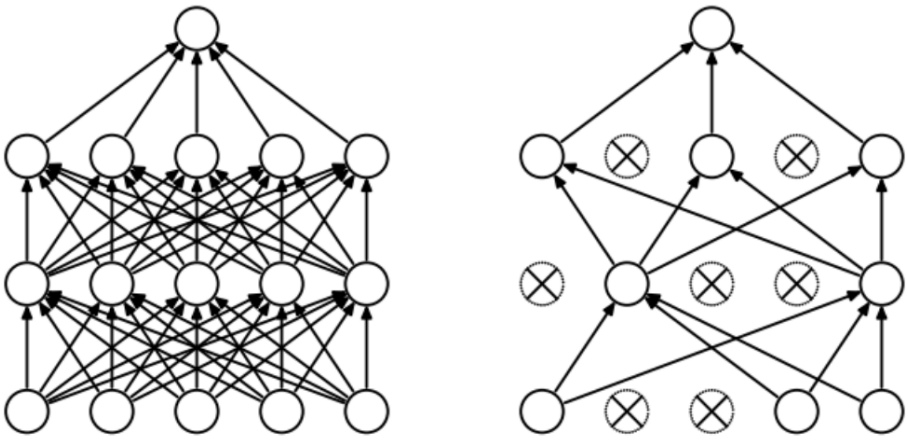
$$\sigma^2_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma^2_{\mathcal{B}} + \epsilon}} \qquad \text{// normalize}$$

$$y_i \leftarrow \gamma \widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

Algorithm 1: Batch Normalizing Transform, applied to activation x over a mini-batch.

DropOut



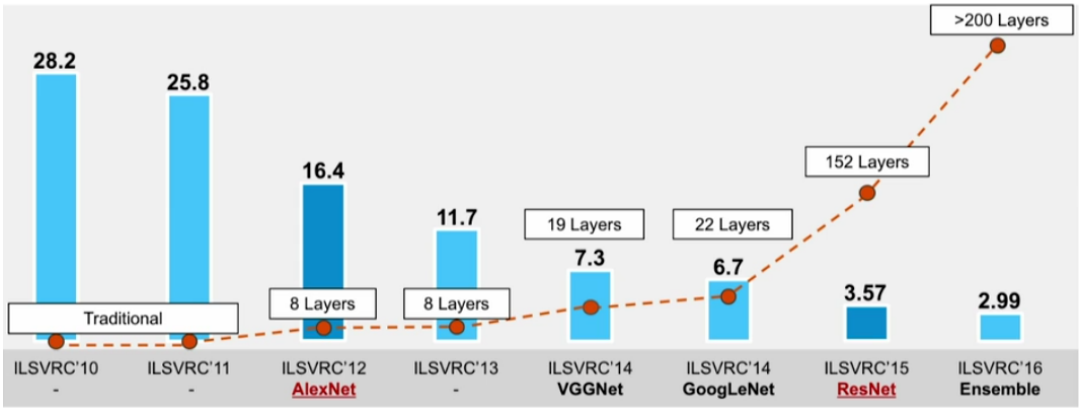
(a) Standard Neural Net (b) After applying dropout.

What is the target for the RF?

What is the size of the objects here?

Remember tiny-imagenet?

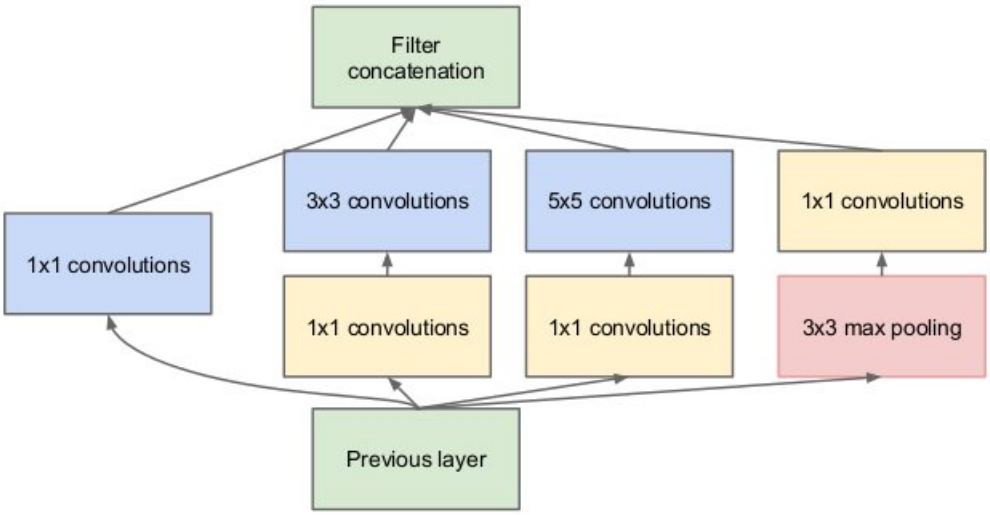
ILSVRC



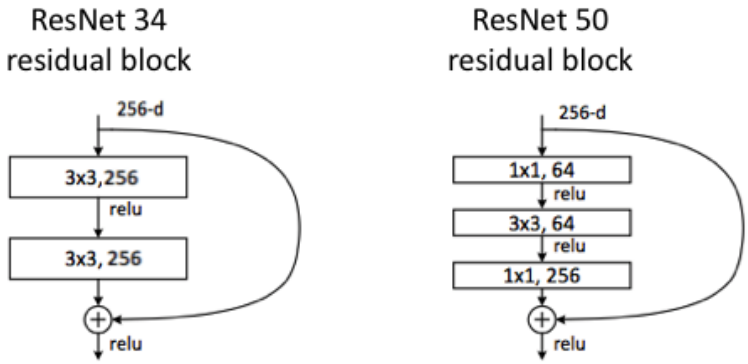
VGG

OUR Architecture

INCEPTION V1



RESNET



What is happening to our receptive fields here and why?

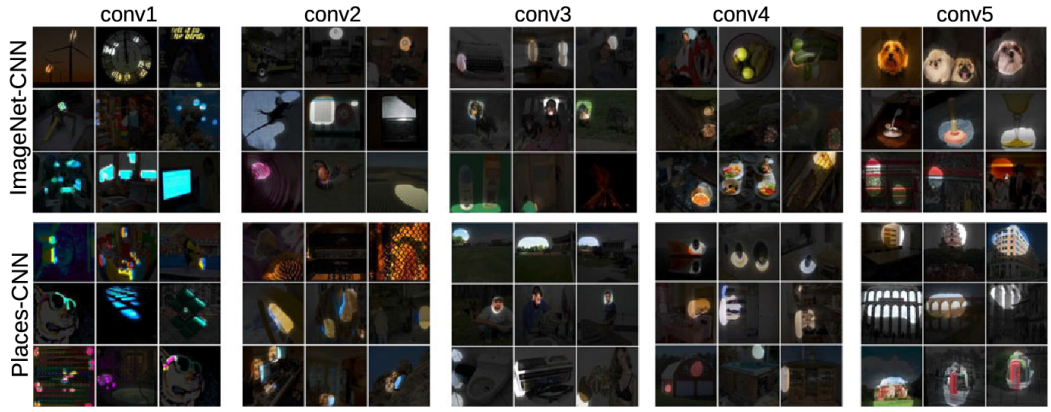


Fig. 15. a) Visualization of the units' receptive fields at different layers for the ImageNet-CNN and Places-CNN. Subsets of units at each layer ar

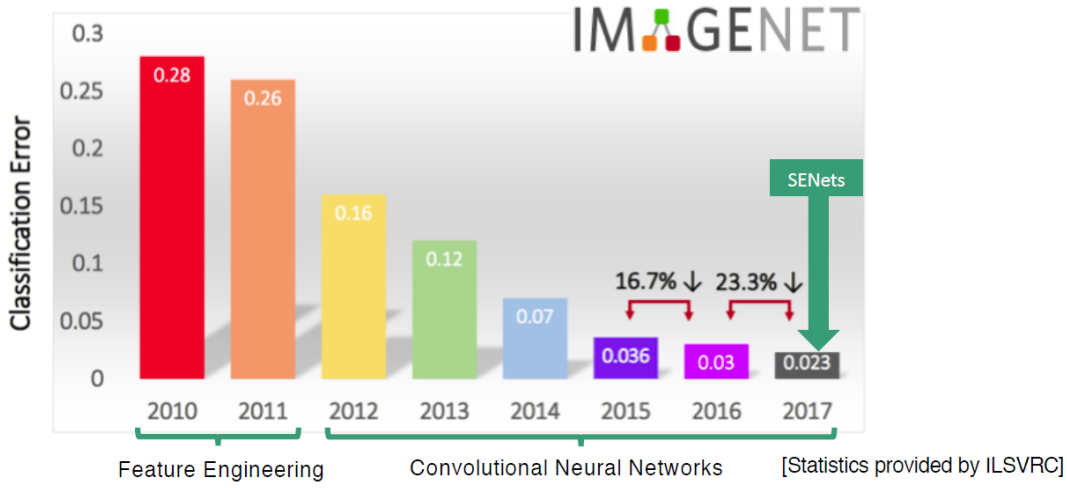
OUR NEW ARCHITECTURE

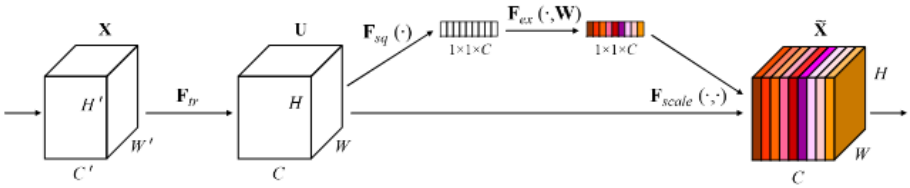
A PROBLEM IN OUR NEW ARCHITECTURE

Solutions:

1. SENET
2. eNAS

SENET





Explicitly model channel-interdependencies within modules
Feature Recalibration - selectively enhance useful features and suppress less useful ones

eNAS

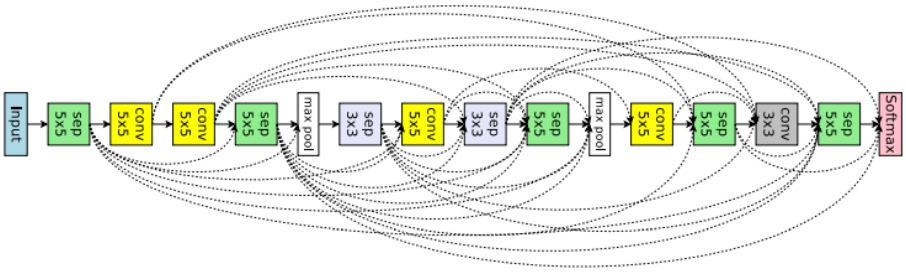


Figure 7. ENAS's discovered network from the macro search space for image classification.

Receptive Field Size	Top-1 Accuracy (single frame)
79×79	75.2%
151×151	76.4%
299×299	76.6%

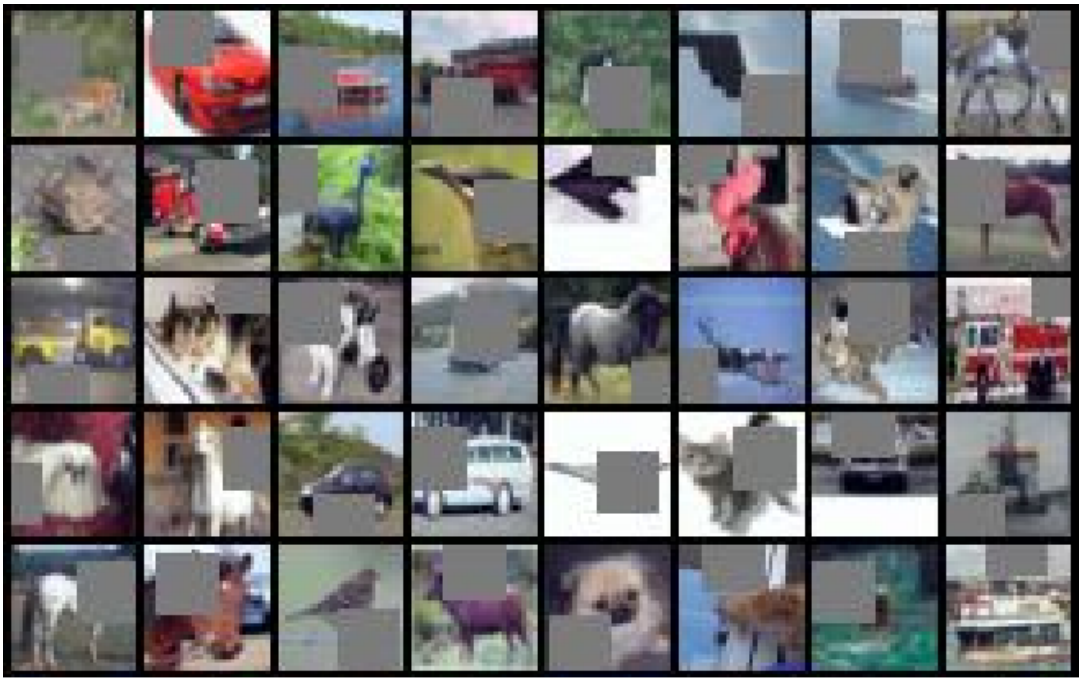
Table 2. Comparison of recognition performance when the size of the receptive field varies, but the computational cost is constant.

RECEPTIVE FIELD CALCULATIONS

$$n_{out} = \left\lfloor \frac{n_{in} + 2p - k}{s} \right\rfloor + 1$$
$$j_{out} = j_{in} * s$$
$$r_{out} = r_{in} + (k-1) * j_{in}$$
$$start_{out} = start_{in} + \left(\frac{\mathrm{k}-1}{2} - \mathrm{p}\right) * j_{in}$$

IMAGE AUGMENTATION

Cutout (https://arxiv.org/pdf/1708.04552.pdf) **:**



Method	C10	C10+	C100	C100+	SVHN
ResNet18 [5]	10.63 ± 0.26	4.72 ± 0.21	36.68 ± 0.57	22.46 ± 0.31	-
ResNet18 + cutout	9.31 ± 0.18	3.99 ± 0.13	34.98 ± 0.29	21.96 ± 0.24	-
WideResNet [22]	6.97 ± 0.22	3.87 ± 0.08	26.06 ± 0.22	18.8 ± 0.08	1.60 ± 0.05
WideResNet + cutout	$\mathbf{5.54 \pm 0.08}$	3.08 ± 0.16	$\mathbf{23.94 \pm 0.15}$	18.41 ± 0.27	$\mathbf{1.30 \pm 0.03}$
Shake-shake regularization [4]	-	2.86	-	15.85	-
Shake-shake regularization + cutout	-	$\mathbf{2.56 \pm 0.07}$	-	$\mathbf{15.20 \pm 0.21}$	-

Table 1: Test error rates (%) on CIFAR (C10, C100) and SVHN datasets. "+" indicates standard data augmentation (mirror + crop). Results averaged over five runs, with the exception of shake-shake regularization which only had three runs each. Baseline shake-shake regularization results taken from [4].

Example Code (https://github.com/yu4u/cutout-random-erasing)

Reinforcement learning or AutoAugment (https://arxiv.org/abs/1805.09501)

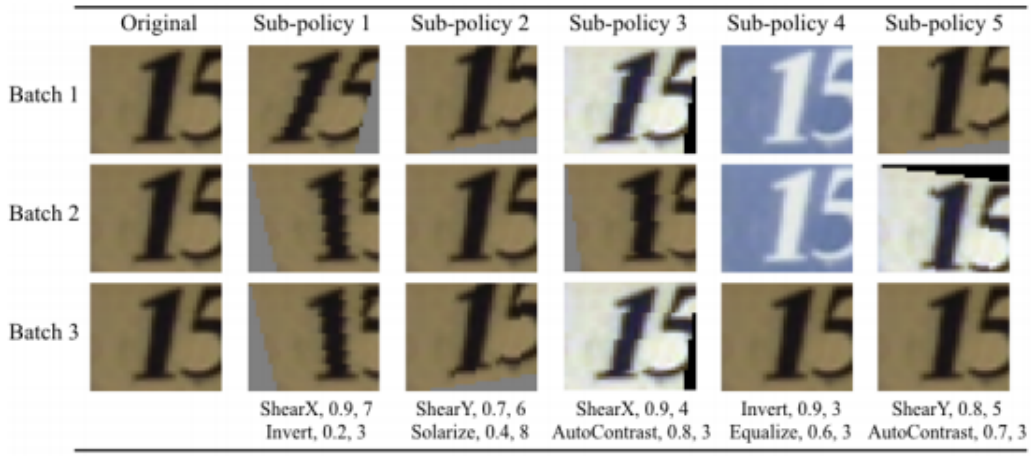
Pros:

1. Can be applied directly on the dataset
2. Learned policies can be transferred to a new dataset
3. Achieves State-of-art for ImageNet, CIFAR10, and CIFAR100
4. NAS took CIFAR10 error to 2% (cannot be beaten by humans), AutoAugment with NAS took this number to 1.5%

Cons:

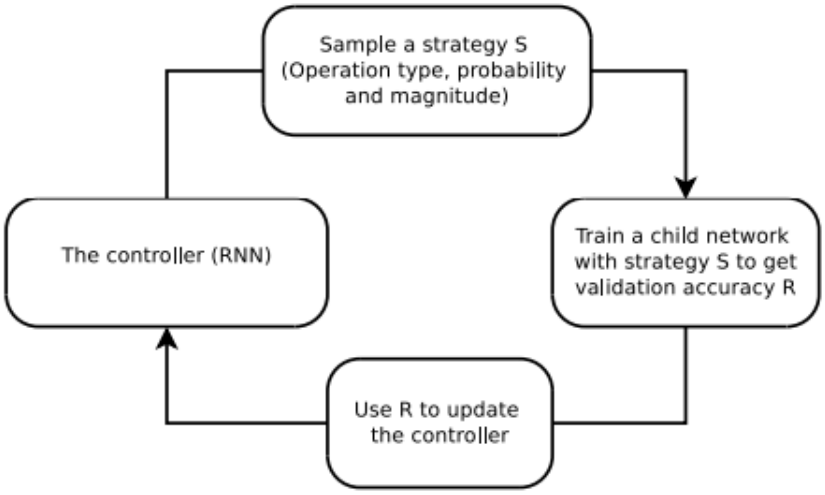
1. Takes 5000 P100 GPU hours for CIFAR and 15000 GPU hours for ImageNet

Search Space Details: A policy consists of 5 sub-policies with each sub-policy consisting of two image operations to be applied in sequence with a) the probability of applying the operation, and b) the magnitude of the operations. Few examples are below:



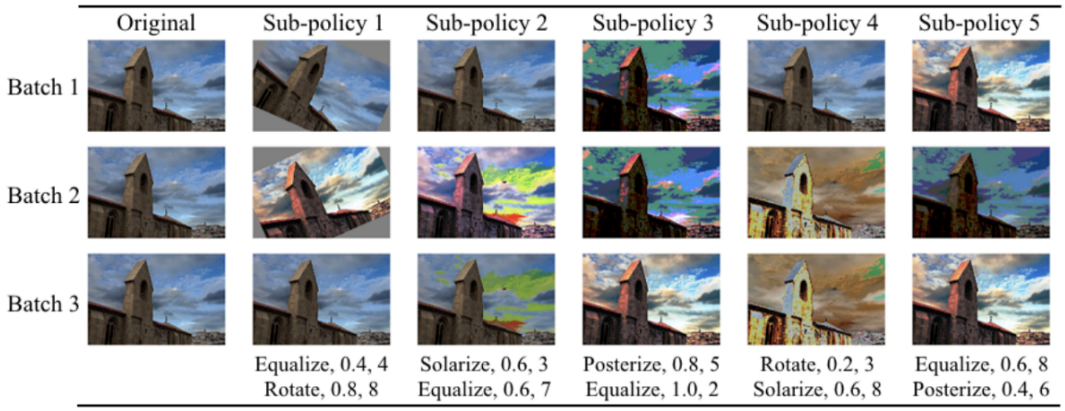
Operations considered are ShearX/Y, Translate X/Y, Rotate, AutoContrast, Invert, Equalize, Solarize, Posterize, Contrast, Color, Brightness, Sharpness, Cutout, Sample Pairing. Each operation also comes with a default 10 range of magnitudes. There are roughly 2.9×10^{32} possibilities.

Search Space Algorithm: RL



Results:

Dataset	Model	Baseline	Cutout [12]	AutoAugment
CIFAR-10	Wide-ResNet-28-10 [67]	3.9	3.1	2.6±0.1
	Shake-Shake (26 2x32d) [17]	3.6	3.0	2.5±0.1
	Shake-Shake (26 2x96d) [17]	2.9	2.6	2.0±0.1
	Shake-Shake (26 2x112d) [17]	2.8	2.6	1.9±0.1
	AmoebaNet-B (6,128) [48]	3.0	2.1	1.8±0.1
	PyramidNet+ShakeDrop [65]	2.7	2.3	**1.5 ± 0.1**
Reduced CIFAR-10	Wide-ResNet-28-10 [67]	18.8	16.5	14.1±0.3
	Shake-Shake (26 2x96d) [17]	17.1	13.4	**10.0 ± 0.2**
CIFAR-100	Wide-ResNet-28-10 [67]	18.8	18.4	17.1±0.3
	Shake-Shake (26 2x96d) [17]	17.1	16.0	14.3±0.2
	PyramidNet+ShakeDrop [65]	14.0	12.2	**10.7 ± 0.2**
SVHN	Wide-ResNet-28-10 [67]	1.5	1.3	1.1
	Shake-Shake (26 2x96d) [17]	1.4	1.2	**1.0**
Reduced SVHN	Wide-ResNet-28-10 [67]	13.2	32.5	8.2
	Shake-Shake (26 2x96d) [17]	12.3	24.2	**5.9**



Assignment:

1. Find 50 misclassified images from the pre-trained ResNet18 Model trained on CIFAR10. **https://keras.io/examples/cifar10_resnet/ (https://keras.io/examples/cifar10_resnet/)**
2. Run **GradCam (http://www.hackevolve.com/where-cnn-is-looking-grad-cam/)** on these images
3. Create a gallery of your GradCam results
4. Upload your Colab file to a public github repo, and
5. Upload your GitHub Link here: **https://tinyurl.com/yxt6x2qq (https://tinyurl.com/yxt6x2qq)**
6. You need to attempt this quiz before the next session starts: **https://tinyurl.com/y2t2ux8z (https://tinyurl.com/y2t2ux8z)**

Some Points to remember:

1. Top 50 percentile will move to Phase 3
2. Since your submissions are public, we recommend making new profiles, and not to share your links with others.
3. Even if the evaluation of assignments is late, we will see the data of change on GitHub to make sure you have followed submission deadlines
4. Each Assignment is 500 marks. Each Quiz is 250 marks (scaled up).

Points 500

Submitting a website url

Due	For	Available from	Until
-	Everyone	-	-

+ Rubric