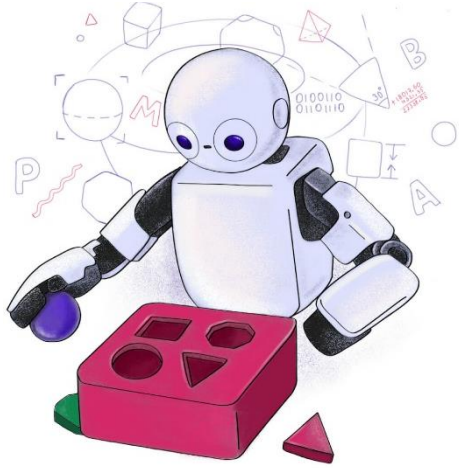
TP558 - Tópicos avançados em Machine Learning:

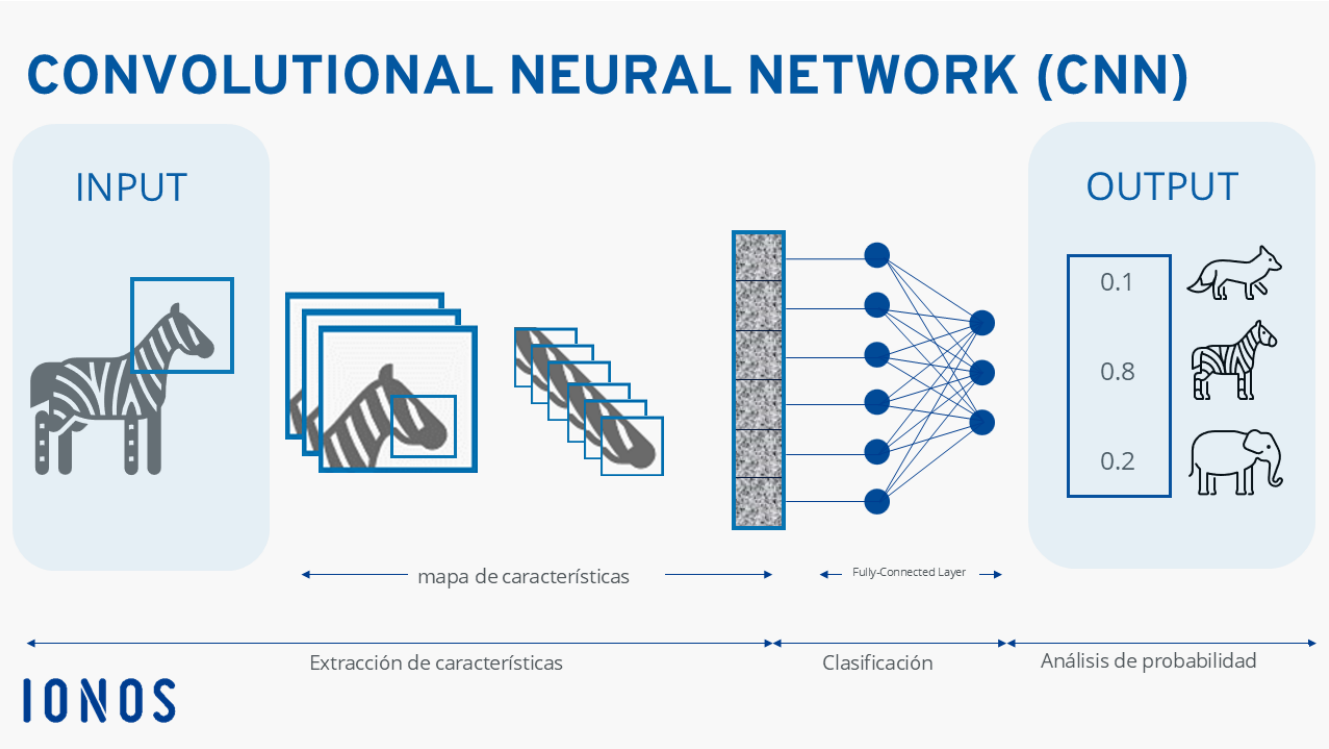
Squeeze and Excitation Networks



Inatel

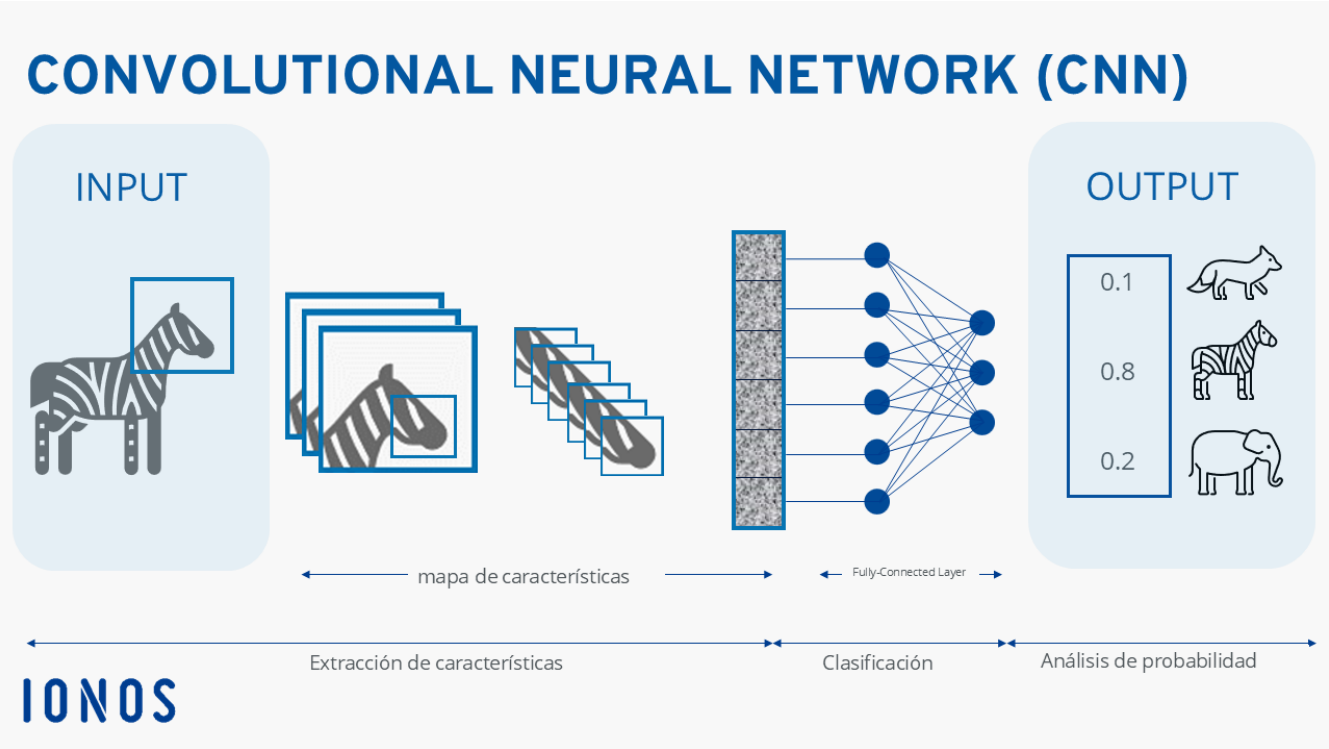
Brithany Michelle Oliva Chuquimia
brithany.oliva@mtel.inatel.br

Introdução



- **Avanços em visão computacional:** A visão computacional avançou significativamente graças às redes neurais convolucionais (CNNs).
- **Efetividade das CNNs:** São altamente eficazes em tarefas como classificação de imagens, detecção de objetos e segmentação semântica.

Introdução



- **Capacidade de aprendizado:** As CNNs aprendem representações de imagens diretamente a partir dos dados.
- **Operador fundamental:** O operador de convolução é o bloco de construção chave das CNNs.

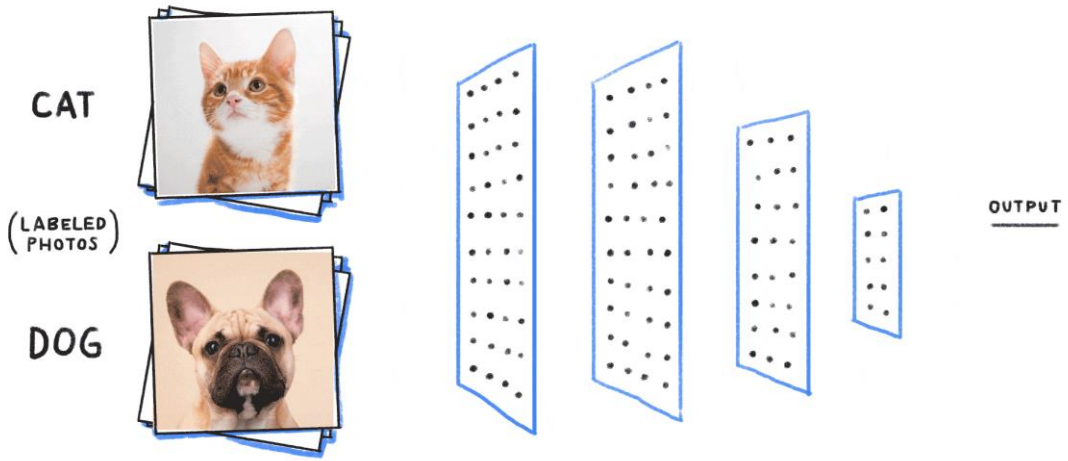
Introdução

- **Pesquisa em visão computacional:** Focou-se em melhorar a captura de relações espaciais por meio de:
  - Arquiteturas mais profundas: Exemplo, redes ResNet, que aumentam a capacidade da rede para processar informações complexas.
  - Processos de múltiplas escalas: Exemplo, modelos Inception, que analisam imagens em diferentes níveis de detalhe.

Introdução

- **Relações entre canais:** As relações espaciais são cruciais, mas as interações entre canais de características foram menos exploradas, segundo os autores do artigo.
- **Proposta do artigo:** Apresenta um novo bloco arquitetônico chamado **Squeeze-and-Excitation** (SE) para abordar a falta de exploração nas interdependências entre canais.
- **Função do bloco SE:** Permite que a rede **identifique e priorize características mais informativas**, recalibrando as respostas dos filtros para focar no relevante.

Fundamentação teórica



O que é uma Rede Neural Convolucional (CNN)?

- É um tipo de rede neural especializada em imagens.
- Ela não trata a imagem como uma longa lista de pixels, mas utiliza uma operação chamada convolução para escanear a imagem.
- Isso permite identificar padrões no espaço, como bordas, texturas ou formas, de maneira muito eficiente.

Fundamentação teórica

O Processo de Convolução:

- A transformação convolucional(F_{tr}) recebe uma entrada X e produz um conjunto de mapas de características U.
- O artigo descreve essa operação com a seguinte fórmula:

$$u_c = v_c * X = \sum_{s=1}^{C'} v_c^s * x^s$$

Fundamentação teórica

O Processo de Convolução:

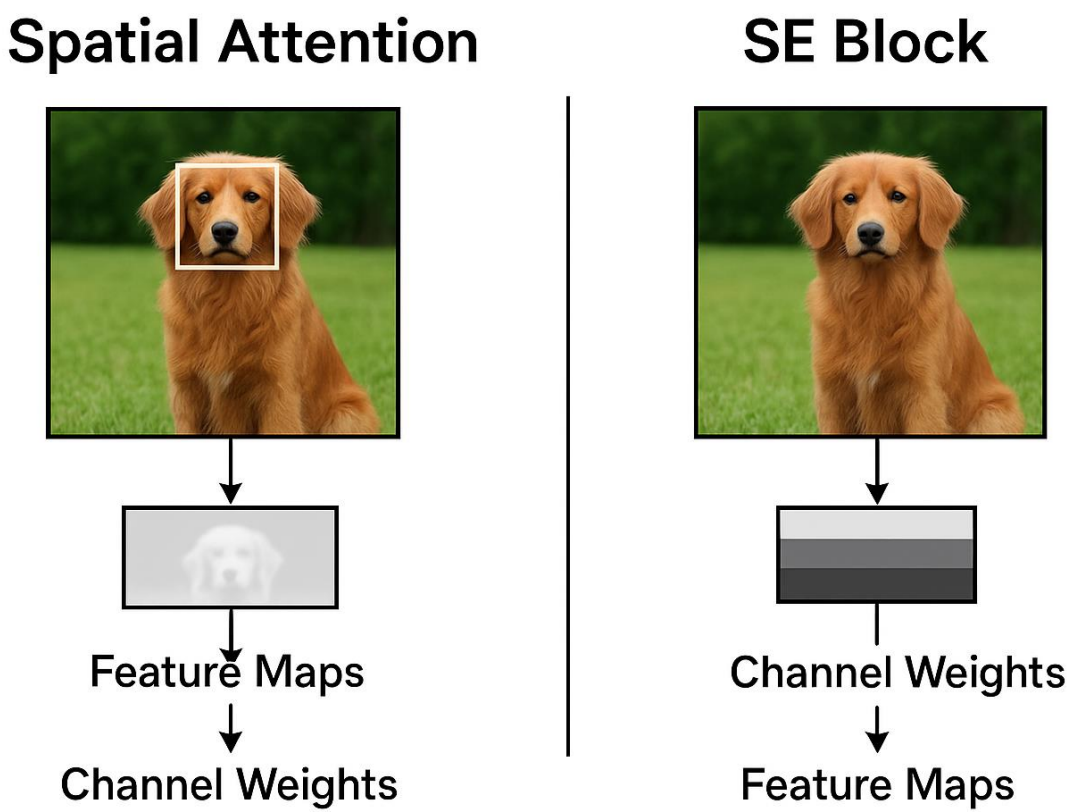
$$u_c = v_c * X = \sum_{s=1}^{C'} v_c^s * x^s$$

- O ponto-chave aqui é que a saída (u_c) é obtida como a soma das convoluções através de todos os canais de entrada.
- Isso significa que a relação entre os canais é implícita e está vinculada à correlação espacial local do filtro.

Fundamentação teórica

Mecanismos de Atenção (Attention Mechanisms):

- O SE Block é inspirado no conceito de **atenção**
- A atenção, em geral, **visa alocar recursos computacionais** para os componentes mais informativos de um sinal.
- Em vez de atenção espacial (onde olhar), o SE Block implementa atenção em canais.

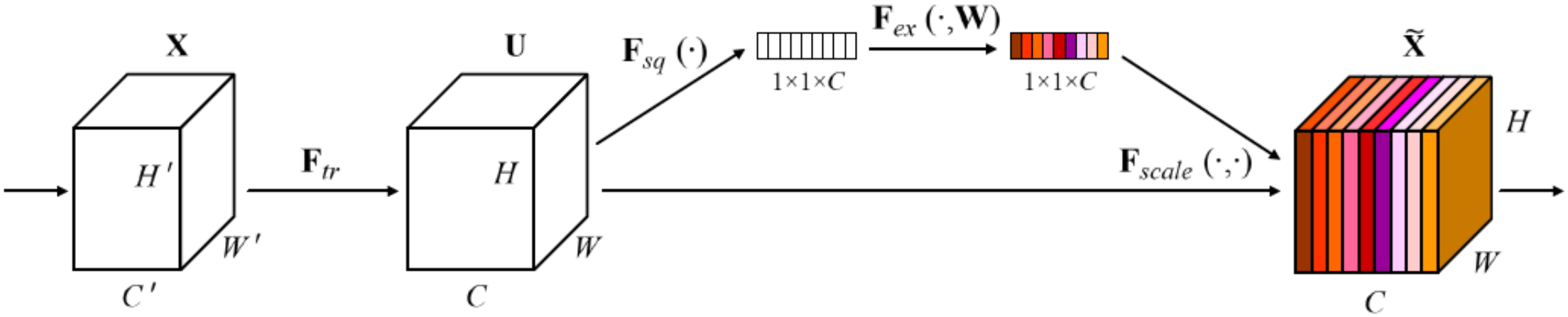


Fundamentação teórica

Recalibração de Características:

- A ideia central é que **a rede deve ser capaz de ajustar a importância de cada canal de características com base no contexto global da imagem**.
- Os autores argumentam que, ao fazer isso, **a rede pode dar mais atenção a informações úteis e suprimir as menos importantes**, melhorando sua capacidade de representação.

Fundamentação teórica

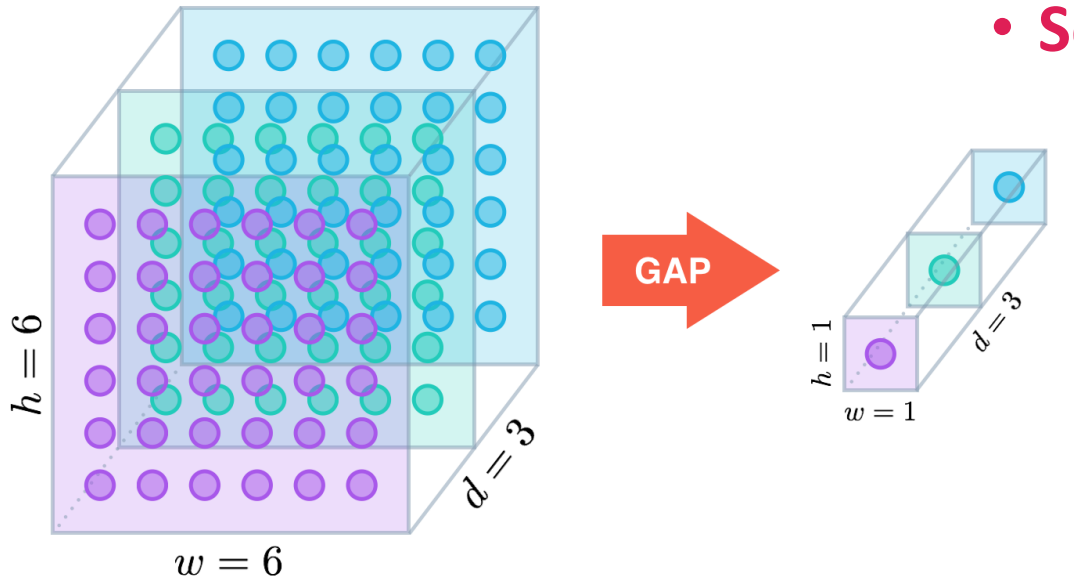


Fundamentação teórica

Squeeze and Excitation:

- **Squeeze (Compressão):**
  - O objetivo da operação Squeeze é **extrair uma representação global da informação contida em cada canal** de características, **agregando os valores espaciais de todo o mapa de características**.
  - Isso é feito através de um **Global Average Pooling** (pooling médio global) aplicado a cada canal individualmente.

Fundamentação teórica



Squeeze and Excitation:

- **Squeeze (Compressão):**
  - **Global Average Pooling (GAP)**: Resume cada canal de recurso calculando a média de todos os seus valores. Cada canal é representado por um único número que reflete suas informações gerais, independentemente da posição dos padrões na imagem.

Fundamentação teórica

- **Squeeze (Compressão):**

A Operação de Compressão(Squeeze):

$$z_c = F_{sq}(u_c) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} u_c(i, j)$$

- z_c: É $o\ c-$ésimo $elemento\ do\ vetor\ de\ saída\ da\ compressão, z.$
- u_c: É $o\ c-$ésimo $mapa\ de\ características\ da\ entrada.$
- $H \times W$: $São\ as\ dimensões\ espaciais\ do\ mapa\ de\ características.$
- F_{sq}: $Representa\ a\ operação\ de\ compressão.$

Fundamentação teórica

- **Squeeze (Compressão):**

  A Operação de Compressão(Squeeze):

$$z_c = F_{sq}(u_c) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} u_c(i,j)$$

- A operação "**aplasta**" a informação espacial (altura × largura) de cada canal em um único número.
- O resultado é **um vetor *z* de dimensão *C*** , onde cada elemento **representa a atividade média global do respectivo canal**.
- Este vetor serve como uma descrição compacta da resposta de cada canal ao input, capturando informações contextuais globais.

Fundamentação teórica

- **Squeeze (Compressão):**

  Por que isso é importante?

  - Permite que o modelo tenha acesso à informação global do campo receptivo, mesmo que as convoluções locais só vejam partes da imagem.
  - É como se cada canal fosse uma “lente” especializada (bordas, texturas, cores). O Squeeze pega todas as ativações desse canal espalhadas na imagem e faz um resumo em um único valor médio, representando a “força” ou presença global daquele padrão

Fundamentação teórica

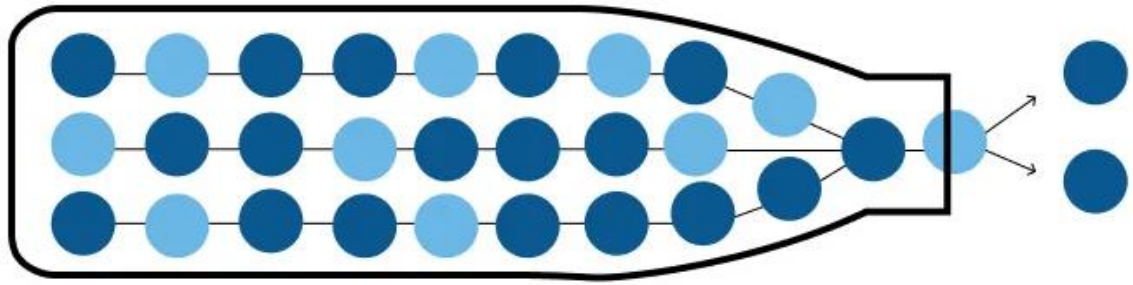
Squeeze and Excitation:

- **Excitation (excitação):**
  - Tem como objetivo usar a informação global extraída pela operação de *Squeeze* para **decidir quan importante é o canal y gera um peso numérico (entre 0 y 1) para esse canal**. Posteriormente, este peso é usado para ajustar a força das características desse canal. Essa recalibração é feita através de um **MLP (rede totalmente conectada) de duas camadas**, que inclui um **bottleneck** para reduzir temporariamente a dimensão do vetor.

Fundamentação teórica

Squeeze and Excitation:

- **Excitation (excitação):**
  - **Bottleneck**: técnica que reduz temporariamente a dimensão de C para C/r, tornando a operação mais eficiente e permitindo que a rede aprenda relações compactas entre canais.



Fundamentação teórica

- **Excitation (excitação):**

  A Operação de Excitação(Excitation) :

$$s = F_{ex}(z, W) = \sigma(W_2 \delta(W_1 z))$$

  - s: *É o vetor de pesos de ativação, onde* $s \in RC$.
  - z: É *o vetor de saída da operação de compressão*.
  - δ: É *a função de ativação* $ReLU$. *O fator* r é *um parâmetro de redução que controla o tamanho da camada intermediária*.
  - σ: É *a função de ativação Sigmoide, que garante que os pesos de saída estejam no intervalo* $[0, 1]$.
  - $W_1 \in \mathbb{R}^{\frac{C}{r} \times C}$ *y* W_2 $\mathbb{R}^{c \times \frac{C}{r}}$: *São os pesos das duas camadas totalmente conectadas* (*o gargalo*).

Fundamentação teórica

- **Excitation (excitação):**

  Por que isso é importante?

  - Permite que o modelo ajuste dinamicamente a importância de cada canal, focando nos mais relevantes e suprimindo os menos úteis, aumentando a precisão e eficiência da rede.

Fundamentação teórica

- **Scaling:**

  Finalmente, a saída do bloco SE é obtida multiplicando cada mapa de características u_c pelo seu peso de ativação correspondente s_c.

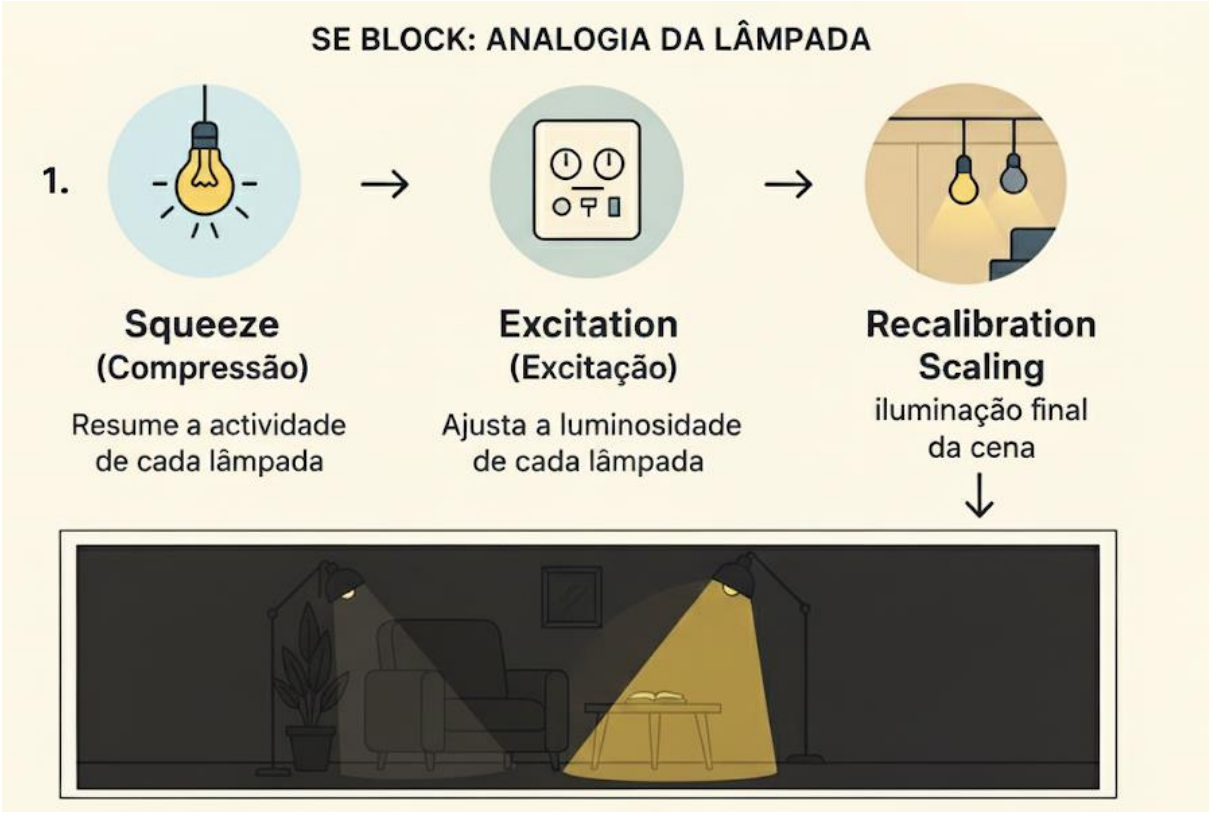
$$x_c = s_c \cdot u_c$$

  - O resultado (x_c) **é o mapa de características original, mas com seus valores "recalibrados" pela rede**. Este mapa de características recalibrado pode ser passado para a próxima camada da rede, melhorando o fluxo de informação e o poder de representação da rede.

Fundamentação teórica

Exemplo intuitivo: Bloco SE como iluminação de cena

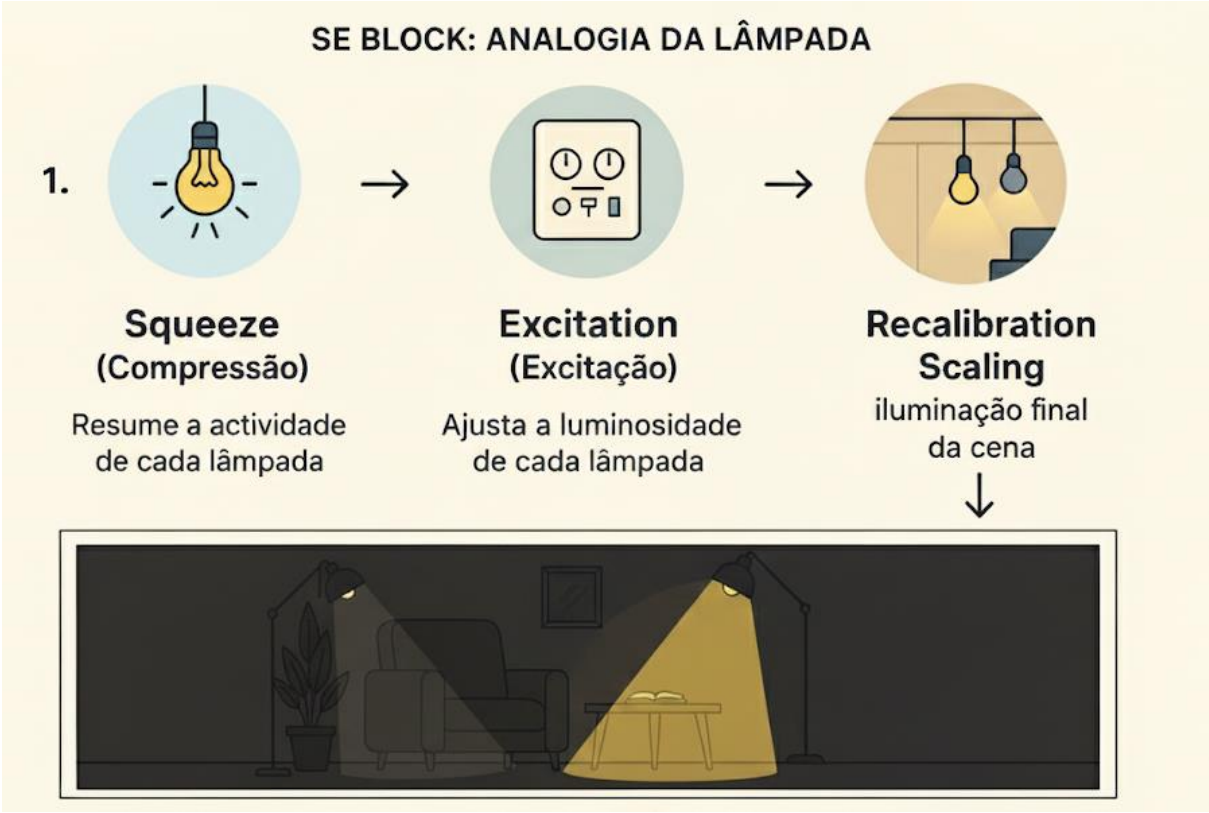
- Squeeze (Compressão)
  - Cada canal de destaque é como uma luminária em um ambiente que ilumina diferentes detalhes (bordas, texturas, cores).
  - O Squeeze mede a quantidade de luz que cada luminária contribui para todo o ambiente, resumindo sua intensidade total em um valor.



Fundamentação teórica

Exemplo intuitivo: Bloco SE como iluminação de cena

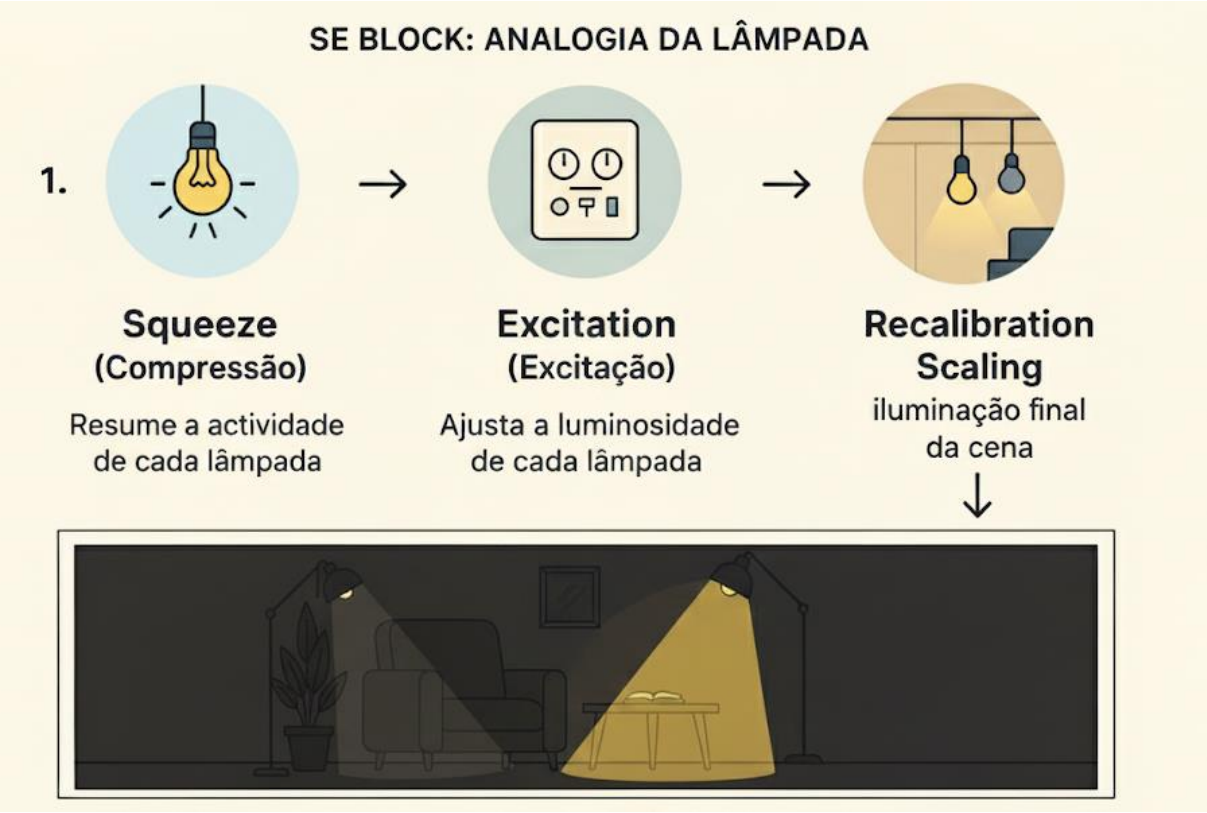
- Excitation (Excitação)
  - Usando esses valores, o Excitation decide quais lâmpadas são mais importantes para a cena atual.
  - Ele ajusta a intensidade de cada lâmpada: algumas ficam mais brilhantes, outras mais fracas.



Fundamentação teórica

Exemplo intuitivo: Bloco SE como iluminação de cena

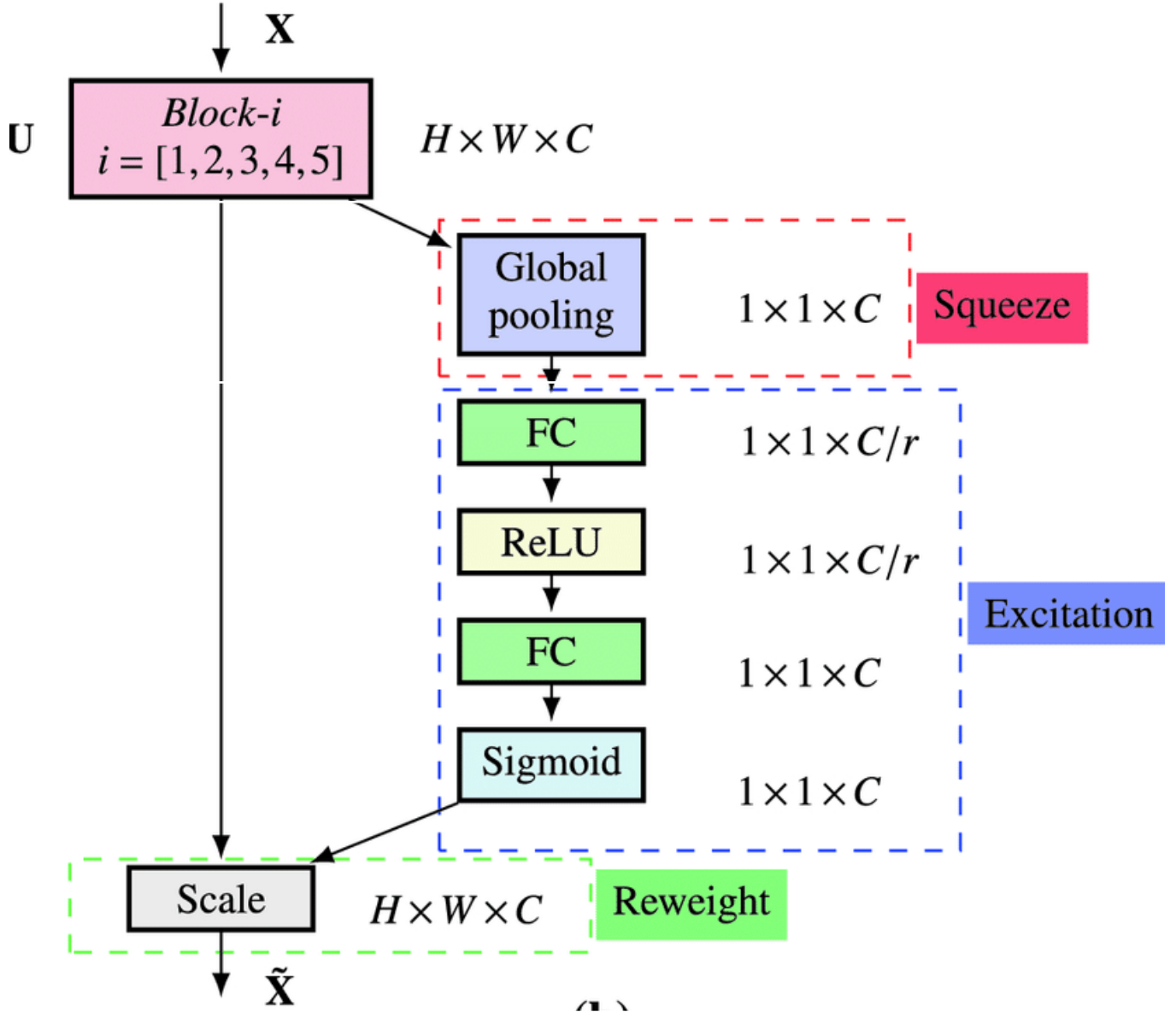
- Recalibration / Scaling (Recalibração)
  - Por fim, cada lâmpada ilumina de acordo com sua intensidade recalibrada, destacando os elementos mais importantes do ambiente.
  - A cena final fica mais nítida e equilibrada, com detalhes relevantes mais bem destacados.



Arquitetura e funcionamento

Contexto na rede

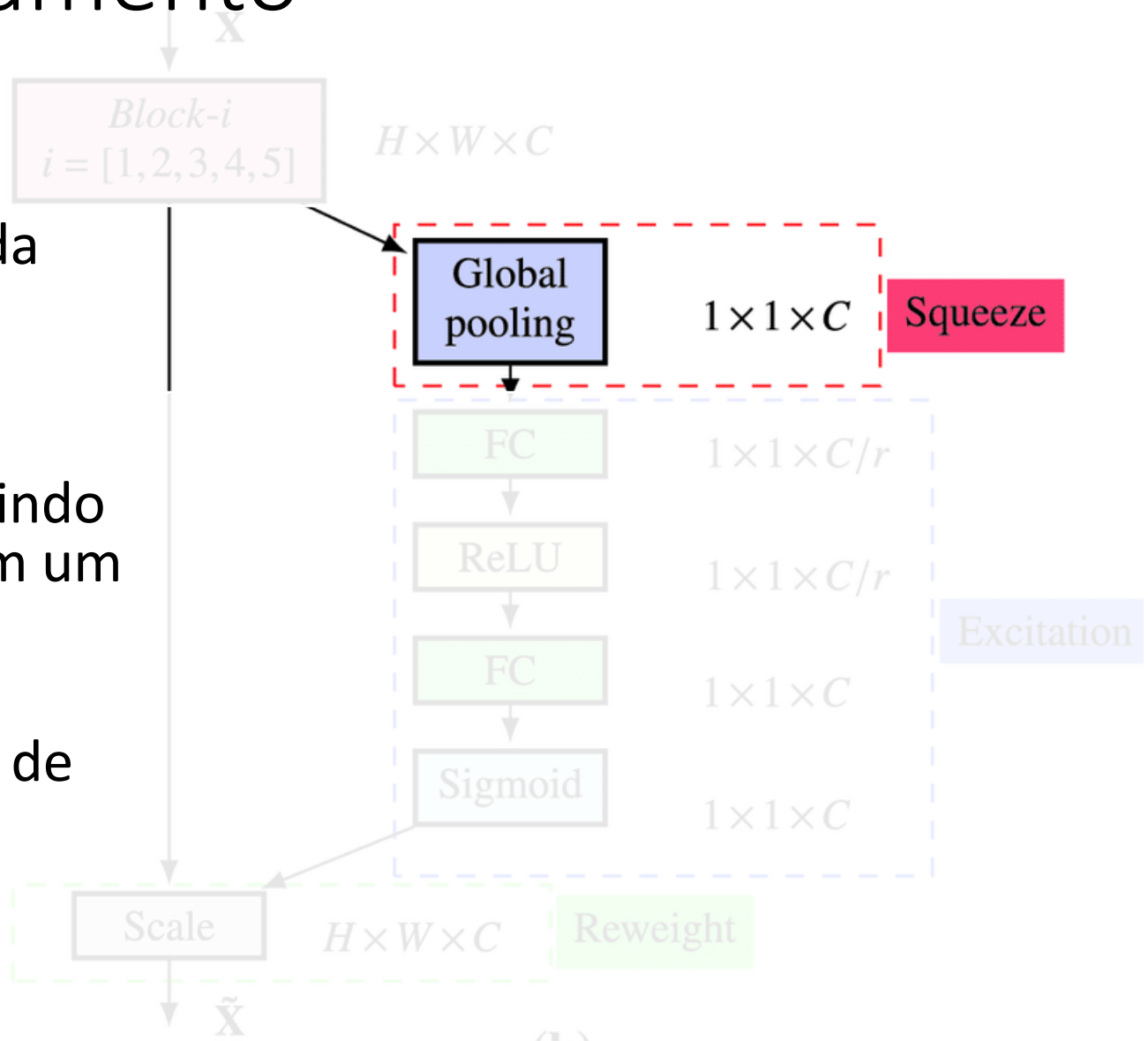
- O bloco SE é geralmente inserido após um conjunto de camadas convolucionais (por exemplo, depois de um bloco residual na ResNet).
- O objetivo é recalibrar dinamicamente a importância de cada canal de características antes de enviá-los para a próxima camada convolucional.



Arquitetura e funcionamento

Operação Squeeze (Compressão)

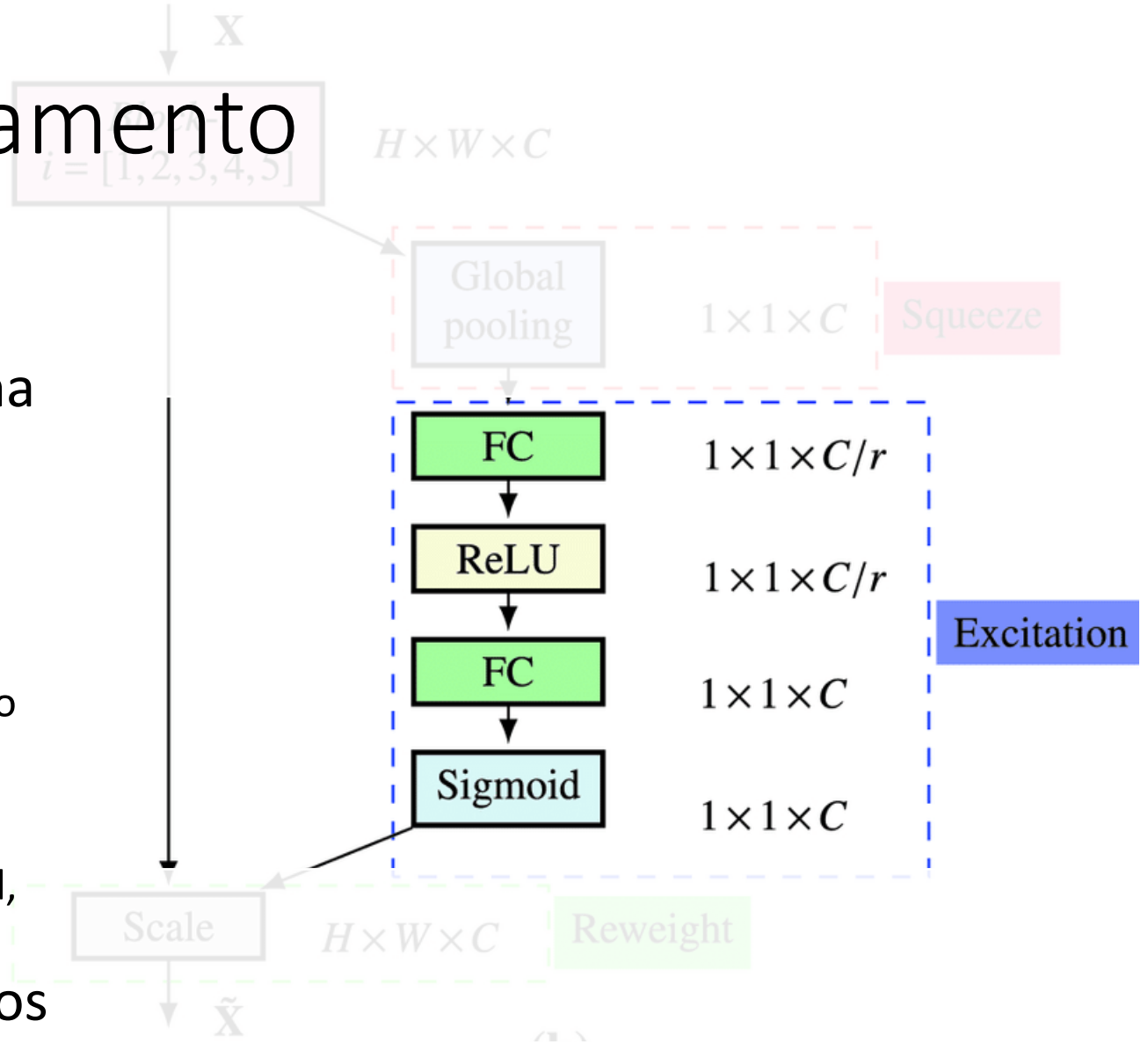
- Entrada: Mapas de características da camada convolucional anterior, de tamanho H×W×C.
- Processo: Aplica-se o **GAP** nas dimensões espaciais (H×W), resumindo toda a informação de cada canal em um único valor.
- Saída: Vetor z de tamanho 1×1×C, representando a “atividade global” de cada canal.



Arquitetura e funcionamento

Operação Excitation

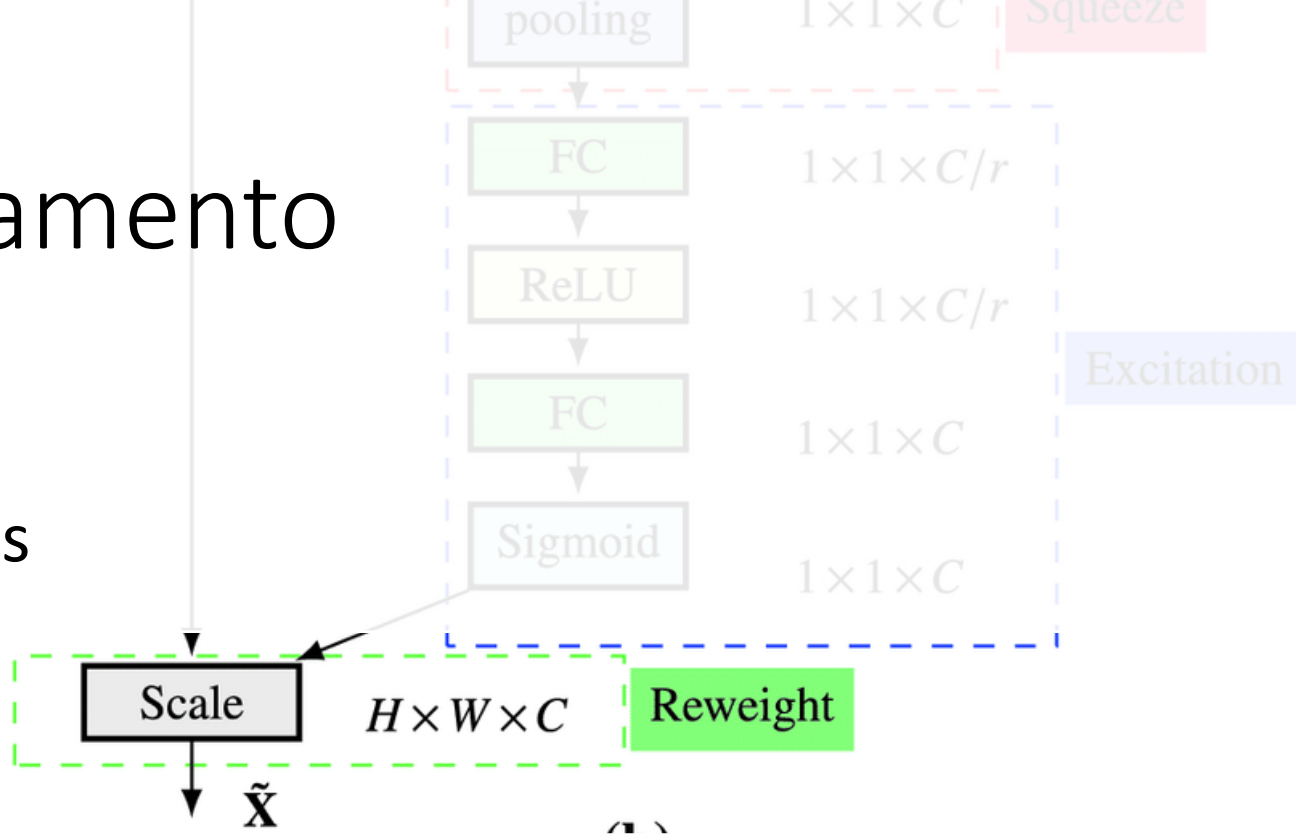
- Entrada: Vetor z de 1×1×C gerado na etapa Squeeze.
- Processo:
  - Passa por uma pequena rede de duas camadas (MLP):
    - Camada 1 (redução): Reduz a dimensão de C → C/r com ativação ReLU (bottleneck).
    - Camada 2 (expansão): Restaura a dimensão para C com ativação Sigmoid, gerando pesos entre 0 e 1.
- Saída: Vetor s de 1×1×C, contendo os pesos de atenção de cada canal.



Arquitetura e funcionamento

Operação Scale

- Entrada: Mapas de características originais e vetor de pesos s.
- Processo: Multiplicação elemento a elemento de cada canal pelo seu respectivo peso.
- Saída: Mapas de características recalibrados, prontos para serem processados pela próxima camada convolucional.



Treinamento e otimização

- **Treinamento conjunto**
  - Os blocos SE não são treinados isoladamente, ele aprende junto com toda a rede convolucional.
  - Eles são integrados às redes convolucionais (como ResNet, Inception, etc.) e aprendem junto com todos os outros parâmetros da rede.
- **Função de perda**
  - A mesma usada na tarefa principal.
  - O bloco SE não adiciona uma perda extra, apenas recalibra canais de forma adaptativa.

Treinamento e otimização

- **Integração com arquiteturas base**
  - O SE pode ser facilmente adicionado a redes já existentes sem alterar o método de treinamento.
  - O aumento de custo computacional é mínimo e não exige técnicas de otimização especiais.

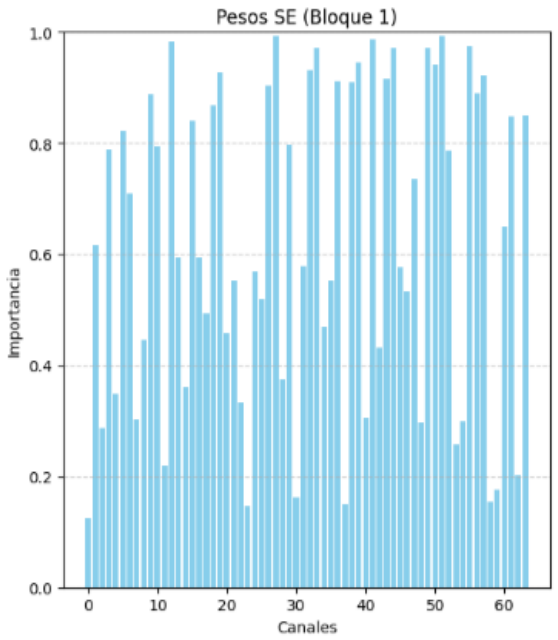
Vantagens e desvantagens

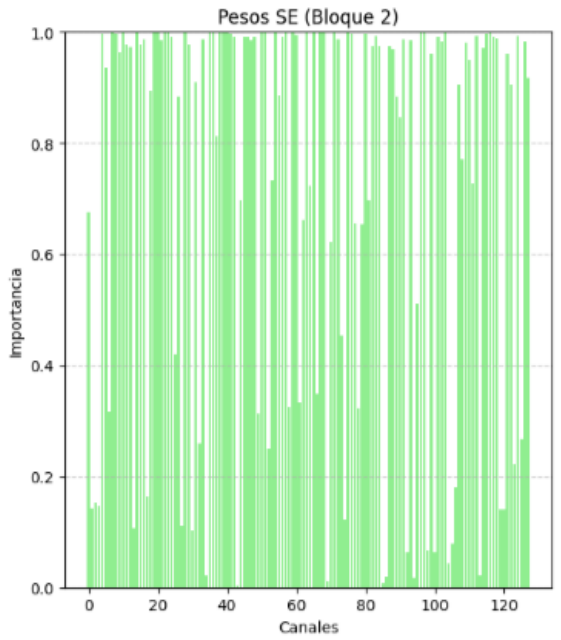
- **Vantagens:**
  - Melhora de desempenho: O bloco SE proporciona ganhos significativos em modelos de última geração com um custo computacional mínimo.
  - Eficiência: O aumento da complexidade do modelo é muito leve.
  - Plug-and-Play: É um bloco modular que pode ser facilmente integrado em arquiteturas existentes.
  - Versatilidade: Os benefícios do bloco SE não se limitam a redes grandes; também melhoram o desempenho de arquiteturas leves como MobileNet e ShuffleNet.

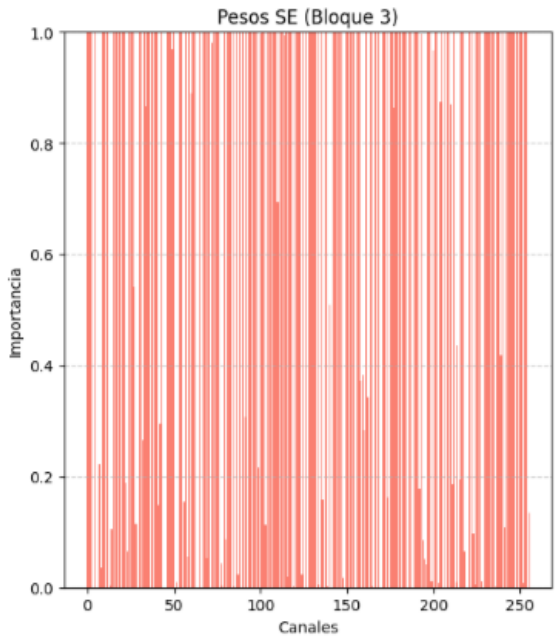
Vantagens e desvantagens

- **Desvantagens/Limitações:**
  - Aumento de complexidade: Embora pequeno, ainda adiciona parâmetros e cálculos extras, o que pode ser crítico em dispositivos com recursos muito limitados.
  - Foco apenas em canais: O SE considera a importância global de cada canal, mas ignora a dimensão espacial (não destaca regiões específicas da imagem).
  - Treinamento ligeiramente mais lento: A adição das camadas extras no MLP aumenta o tempo de treinamento em comparação com a rede base.

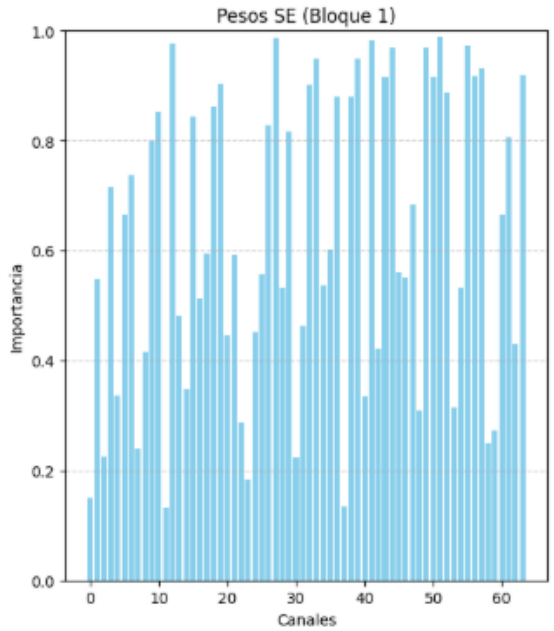
Exemplo(s) de aplicação

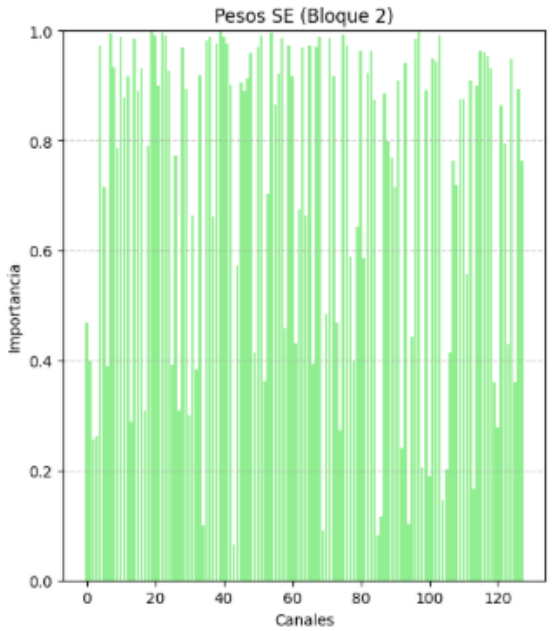


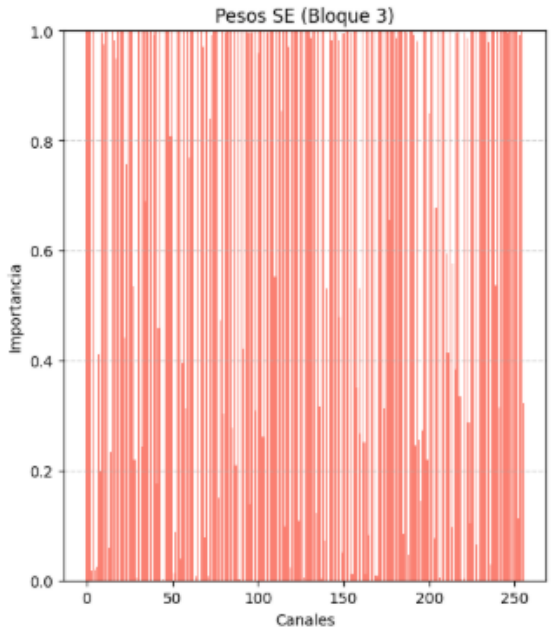




Exemplo(s) de aplicação







Comparação com outros algoritmos

- **Comparación del Módulo SE en Diversas Arquitecturas**

	original		re-implementation			SENet		
	top-1 err.	top-5 err.	top-1 err.	top-5 err.	GFLOPs	top-1 err.	top-5 err.	GFLOPs
ResNet-50 [13]	24.7	7.8	24.80	7.48	3.86	$23.29_{(1.51)}$	$6.62_{(0.86)}$	3.87
ResNet-101 [13]	23.6	7.1	23.17	6.52	7.58	$22.38_{(0.79)}$	$6.07_{(0.45)}$	7.60
ResNet-152 [13]	23.0	6.7	22.42	6.34	11.30	$21.57_{(0.85)}$	$5.73_{(0.61)}$	11.32
ResNeXt-50 [19]	22.2	-	22.11	5.90	4.24	$21.10_{(1.01)}$	$5.49_{(0.41)}$	4.25
ResNeXt-101 [19]	21.2	5.6	21.18	5.57	7.99	$20.70_{(0.48)}$	$5.01_{(0.56)}$	8.00
VGG-16 [11]	-	-	27.02	8.81	15.47	$25.22_{(1.80)}$	$7.70_{(1.11)}$	15.48
BN-Inception [6]	25.2	7.82	25.38	7.89	2.03	$24.23_{(1.15)}$	$7.14_{(0.75)}$	2.04
Inception-ResNet-v2 [21]	$19.9^{\dagger}$	$4.9^{\dagger}$	20.37	5.21	11.75	$19.80_{(0.57)}$	$4.79_{(0.42)}$	11.76

- Esta tabela apresenta uma avaliação detalhada do desempenho e da eficiência dos modelos ResNet, ResNeXt, VGG e Inception, com e sem a adição do módulo SE, no conjunto de dados ImageNet.
- Os resultados mostram melhorias consistentes, demonstrando que a inclusão do SENet aumenta o desempenho dos modelos com um custo computacional mínimo

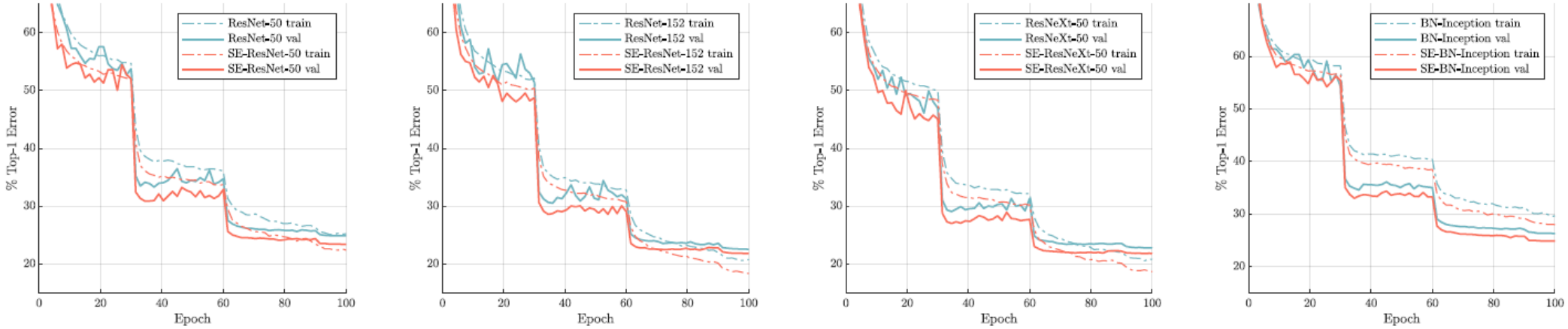
Comparação com outros algoritmos

- **Rendimiento y Complejidad en Redes Ligeras**

	original		re-implementation				SENet			
	top-1 err.	top-5 err.	top-1 err.	top-5 err.	MFLOPs	Params	top-1 err.	top-5 err.	MFLOPs	Params
MobileNet [64]	29.4	-	28.4	9.4	569	4.2M	$25.3_{(3.1)}$	$7.7_{(1.7)}$	572	4.7M
ShuffleNet [65]	32.6	-	32.6	12.5	140	1.8M	$31.0_{(1.6)}$	$11.1_{(1.4)}$	142	2.4M

- Esta tabela compara o desempenho e a eficiência computacional dos modelos MobileNet e ShuffleNet com a adição do módulo SE, demonstrando sua efetividade e baixo custo em arquiteturas projetadas para serem rápidas e leves.
- Os resultados mostram que a inclusão do SENet torna esses modelos mais precisos sem um aumento significativo na complexidade.

Comparação com outros algoritmos



- **Curvas de Trenamiento dos Modelos SE**
- Este conjunto de gráficos mostra o processo de treinamento de diferentes modelos, evidenciando que a adição dos blocos SE resulta em uma convergência mais rápida e estável, alcançando taxas de erro menores tanto nos dados de treinamento quanto nos de validação.
- Visualmente, observa-se que os modelos SENet treinam de forma mais consistente e finalizam com erro mais baixo em relação aos modelos originais.

Comparação com outros algoritmos

- **Efecto do Módulo SE em Diversos Conjuntos de Dados**
- Estas tabelas resumem a melhoria na taxa de erro em tarefas de classificação e detecção de objetos proporcionada pelo módulo SE em modelos aplicados a conjuntos de dados como CIFAR-10, CIFAR-100, Places365 e COCO.

- Elas demonstram a eficácia dos modelos SENet na classificação e detecção de objetos, evidenciando uma redução consistente do erro em diferentes conjuntos de dados

Classification error (%) on CIFAR-10.

	original	SENet
ResNet-110 [14]	6.37	5.21
ResNet-164 [14]	5.46	4.39
WRN-16-8 [67]	4.27	3.88
Shake-Shake 26 2x96d [68] + Cutout [69]	2.56	2.12

Classification error (%) on CIFAR-100.

	original	SENet
ResNet-110 [14]	26.88	23.85
ResNet-164 [14]	24.33	21.31
WRN-16-8 [67]	20.43	19.14
Shake-Even 29 2x4x64d [68] + Cutout [69]	15.85	15.41

Single-crop error rates (%) on Places365 validation set.

	top-1 err.	top-5 err.
Places-365-CNN [72]	41.07	11.48
ResNet-152 (ours)	41.15	11.61
SE-ResNet-152	**40.37**	**11.01**

Faster R-CNN object detection results (%) on COCO *minival* set.

	AP@IoU=0.5	AP
ResNet-50	57.9	38.0
SE-ResNet-50	61.0	40.4
ResNet-101	60.1	39.9
SE-ResNet-101	62.7	41.9

Comparação com outros algoritmos

- **Desempenho dos Modelos SENet em ImageNet**
- Estas tabelas apresentam as taxas de erro dos modelos SENet, destacando seu desempenho competitivo ou superior em comparação com outras arquiteturas de ponta, como Inception, DenseNet e NASNet, no conjunto de dados ImageNet.
- Elas mostram que os modelos SENet alcançam a menor taxa de erro no ImageNet, consolidando-os como líderes frente a outras arquiteturas avançadas.

Single-crop error rates (%) of state-of-the-art CNNs on ImageNet validation set with crop sizes 224×224 and 320×320 / 299×299.

	224×224		320×320 / 299×299	
	top-1 err.	top-5 err.	top-1 err.	top-5 err.
ResNet-152 [13]	23.0	6.7	21.3	5.5
ResNet-200 [14]	21.7	5.8	20.1	4.8
Inception-v3 [20]	-	-	21.2	5.6
Inception-v4 [21]	-	-	20.0	5.0
Inception-ResNet-v2 [21]	-	-	19.9	4.9
ResNeXt-101 (64×4d) [19]	20.4	5.3	19.1	4.4
DenseNet-264 [17]	22.15	6.12	-	-
Attention-92 [58]	-	-	19.5	4.8
PyramidNet-200 [77]	20.1	5.4	19.2	4.7
DPN-131 [16]	19.93	5.12	18.55	4.16
SENet-154	**18.68**	**4.47**	**17.28**	**3.79**

Comparison (%) with state-of-the-art CNNs on ImageNet validation set using larger crop sizes/additional training data. †This model was trained with a crop size of 320×320.

	extra data	crop size	top-1 err.	top-5 err.
Very Deep PolyNet [78]	-	331	18.71	4.25
NASNet-A (6 @ 4032) [42]	-	331	17.3	3.8
PNASNet-5 (N=4,F=216) [35]	-	331	17.1	3.8
SENet-154†	-	320	16.88	3.58
AmoebaNet-C [79]	-	331	16.5	3.5
ResNeXt-101 32×48d [80]	✓	224	14.6	2.4

Perguntas?

Referências

- Jie Hu, et al., “Squeeze-and-Excitation Networks”, https://arxiv.org/pdf/1709.01507.
- Repositório do GitHub: https://github.com/hujie-frank/SENet

Links

- GitHub: https://github.com/Mish0404/TP558/tree/main/Seminario_Squeeze_and_Excitation%20_Networks
- Quiz: Squeeze_and_Excitation_Networks

Obrigado!