Question 1. Key Innovations in Resnet

The main innovation in Resnet [1] is the introduction of Residual connection (or "Skip connection", written as RC hereafter) and the residual block (Figure 2A and mechanism explained later). They allowed inputs to forward propagate faster across layers, enabled deeper models to learn identity function thus ensure their performance to be not worse than their shallower versions[2]. It was hypothesized and proved later in [1] that this mapping is also easier to optimize than the traditional unreferenced mapping. With this, the authors were able to build deeper model (with 34,50,101 layers) at that time which gained best performance among existing solutions on ImageNet.

Explain what was the problem that this feature addressed and how ResNet addressed the problem

Before Resnet, leading results on ImageNet challenges are brought by deeper networks. Thus, one may expect that deeper networks are with better results. However, experimentally, He et al defied our beliefs by pointing out that: as network's depth increases, accuracy is saturated and then degrades rapidly (Figure 1). **This problem is known as degradation**. Which were addressed by features offered in Resnet.

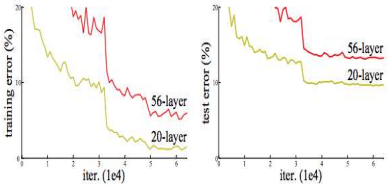

Figure 1: Training (left) and testing error (right) of a 20-layer network and a 56-layer network.

Degradation was not caused by overfitting as the deeper model has higher training error. One of the other causes for degrading may be **vanishing gradient problem (VGP)**. When training deep networks, during gradient descent, backproping from the final layer to the input layer leads to multiplying gradients of layers together, and the result gradient decreased exponentially quickly to 0 leading to extremely small changes to initial layers and making the model harder to train and longer to converge to optimal. However, in his original paper [1], He also noted that this difficulty was less like to be caused by VGP. But still, **this problem is also addressed by Resnet.**

Resnet addressed these problems by using skip connection. Suppose the input to a few stacked layers is X, instead of directly learn an underlying mapping H(x), the residual connection allows networks to learn an underlying residual mapping of H(x) – x, or equivalently, F(x) + x. By this mechanism, if the optimal mapping is the identity function (e.g. When the newly added layers added are not useful), it would be easy for the model to put the residual F(x) to zero instead of directly fitting an identity mapping. H(x) now becomes H(X) ~ x. Thus, this mechanism allows the learning of identity mapping, and skipping ineffective layers while retaining the useful information from previous layers. It thereby guaranteed that the performance of the model would not decrease but would increase thanks to increasing depth. So, the degradation problem is solved. Mathematically, if we express gradient of the loss function w.r.t x as $\frac{\partial L}{\partial x}$, we can see that $\frac{\partial L}{\partial x} = \frac{\partial L}{\partial H} * \frac{\partial H}{\partial x} = \frac{\partial L}{\partial x} * \left(\frac{\partial F}{\partial x} + 1\right)$ (1). As mentioned, multiplications of gradients lead to extreme small value, without the +x terms, the gradient in (1) would be ~0, while with adding x, some information is retained, since the local gradient $\frac{\partial H}{\partial x} = 1$. Thus, VGP is solved.

Compare and contrast the design of ResNet vs that of VGG.

VGG stands for Visual Geometry Group, a group of scientists invented this network. **VGG has 2 variants, VGG16 and VGG19**, the number denotes the number of layers**. A VGG network consists of VGG blocks, each block is composed of several convolutional layers (2 for VGG16, and 2 or 4 for VGG19) following by a Max pooling layer**. After stacking 5 VGG Blocks, **3 dense layers and a softmax activation is attached to output classification**. VGG gradually increased the depth of feature maps. In comparison to Alexnet, VGG employed different strategy: deeper network, but with smaller filter size: instead of using large kernels, they stacked multiple 3x3 filters to have same filter effect with large filters while having fewer parameters, VGG nets are deeper and introduced more non-linearity.

Inspiring by VGG, Resnet is also based on building blocks called residual blocks, containing 2 convolutional layers with 3x3 filter size. Input (x) are directly added to the output, which was obtained after passing several stacked convolutional layers, each was followed by a ReLU activation (Fig 2A). For bigger variants (Resnet50, or 101) the building block changed (see Figure 2: Architecture of Resnet 34(A), building block for Resnet 34(B) and for resnet50/101 (C). Each residual block has fixed number of channels (64,128,256,512). When there is a difference in dimension when adding x to F(x) (dotted line in Fig 2A) either zero-padding the input volume X or perform 1x1 convolution to match dimensions. **In comparison, VGG's architecture contains less layers than Resnet, and is shallower when it's come to depth. VGG could not solve Vanishing Gradient or Degradation problem while Resnet can.**

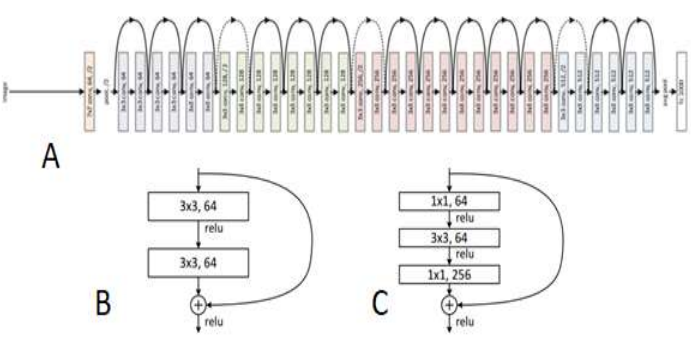

Figure 2: Architecture of Resnet 34(A), building block for Resnet 34(B) and for resnet50/101 (C)

Differences in number of parameters and reasons

Resnet 11 has around ~11.5M parameters (hereby written as params), Resnet 34 has around ~22M params while VGG16 has around ~138M params [3]. **The huge difference comes from the last fully connected layers (hereafter written as FCNs). In VGG, before the FCNs, the output of the above Conv2d + Pool part is 512 feature maps of size 7*7. Thus, connection to the first FCN in VGG with 4096 neurons requires 512*7*7*4096 around 102M params - over 70% of params in VGG are used for this layer, the second and third FCN in VGG requires 16M and 4M params while in Resnet, only 1 FCN with 1000 neurons is used and before that, global average pooling is applied on each map, thus only 512*1000=0.512M params required.**

• Importance of Batch Normalization (BN) in this network

While RC mostly tackled VGP, **BN accelerated convergence speed, solved Internal Covariance Shift (ICS).** Scientists believed that ICS is the source of slow training and convergence as when input's distribution changes, layers in network are forced to adapt that change respectively - This slows down the training by requiring lower learning rates. BN address the problem by normalizing layer inputs [4] and offered faster training. On the other hand, scientists in [5] discovered that instead of actually reducing ICS, BN helped in smoothing optimization landscape thereby allowing faster training and convergence. **BN also has regularizing effect and reduced overfitting** as it introduced noises to layers' activation, therefore forcing later layers not to rely much on one input and reduced the reliance on good weights initialization.

Question 3. Source's paper ["Deep Residual Learning for Image Recognition", He et al] **summary:**

[1] introduced degradation as a problem when training deep CNNs, and thereby **proposed an innovation to tackle it called residual connections and residual blocks** (for more details, revisit q1) with these things, the authors **built a deeper network and achieved the highest accuracy on ImageNet dataset** at the time it was published (19.38 error rate for single model and 3.57 error rate for ensembled version). It outperformed previous SOTA models and **won the 1st place in ILSVRC 2015**. Resnet contains same weakness as other CNNs: couldn't model dependencies between inputs & positional encoding of features, and specifically it brings complexity when dealing with different dimensions in skip connections. We later see Vision Transformer tackled the 2 first issues.

Name and briefly describe the key innovation(s) of the follow-on paper [**"**An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale" by Alexey Dosovitskiy et al[6]] **from the source paper: (Note: ViT from here stands for Vision Transformer model)**

The key innovation of the follow-up paper from the source paper **is the directly application of Transformer architecture** [7] **as a complete alternative to CNN-based architecture to classify images**. Since its invention, Attention and transformer is the first choice in Natural Language Processing but in computer vision, convolutional architectures remain dominant and Resnet variants are widely used as SOTA and baseline to build up models. Although attention is sometimes used, it is used as a subcomponent in CNN networks. Thus, the usage of Transformer to replace CNN is a key and breakthrough innovation**. Another innovation is that ViT architectures feature minimal inductive bias**. While almost CNN (includes Resnet of source paper) carry at least 2 biases: (i) nearby pixel are related (locality) or (ii) translation equivariance. Thus, not having these biases allow ViT freer to train on large scale dataset. However, it is worth-noting that since there is no inductive bias, on small datasets, ViT performance is outperformed by CNN-architecture. **Another innovation is that instead of feeding a whole image like in CNN and source paper, patches of a single image is fed, these patches are treated the same way as tokens in NLP. Finally, the input to the ViT can be patches of image or patches of feature map of a CNN.** Inside its study, ViT has largely used Resnet as a baseline to compare performances. When trained on large and sufficient dataset (JFT-300M with 300M images), the ViT outperformed Resnet Big Transfer (BiT, Resnet 152x4) on most of evaluation set by a distance from 0.08 – 2.89% accuracy.

How the follow-up paper used the key innovation and adapt them in the context of image classification task

Previously, Transformer are mainly used in the NLP field. Original transformer includes an encoder and a decoder part. The encoder encodes input vectors, maps those features to some extend and let the decoder decode relevant information from them. Both used skip connection to prevent VGP like Resnet and includes attention mechanism**. In ViT, only the encoder part of transformer is used. Instead of a decoder, a classification module called MLP head is attached, to convert output of the encoder to classification probabilities. ViT further leverages powerful natural language processing embeddings (BERT [7]) to embed inputs before feeding into encoder**. The ViT was first pretrained on very large-scale datasets, and then fine-tuned evaluated on different datasets, include ImageNet.

Since transformer is not initially designed for Computer Vision, small modifications should be made to make it adapt to the image input. Firstly, since original Transformer accepts a sequence of data as inputs, the original image is split into patches of PxP, forming a sequence of smaller images. Subsequently, every single patch is flattened into a vector, say X $\in R^{P^2C}$, and multiply with an embedding matrix E $\in R^{(P^2C)xD}$ to produce a same D-dimension vector. To retain information about positions of input just like in the original Transformer's architecture, the learnable positional vectors of size D is added to every embedded vectors. These vectors form a sequence of vector input to the Transformer encoder, which remains the same as in [7]. In addition, borrowing an idea from BERT [8] , an encoded 0th class token is added to the beginning of the input sequence vector. These steps made this

Transformer architecture applicable to image while still retaining important characteristics and features: positional encoding, embedding vectors, 0^{th} class token like in BERT.

$$z_0 = [x_{class}; x_p^1 E; x_p^2 E; \cdots; x_p^N E] + E_{pos}, \quad E \in \mathbb{R}^{(P^2 \cdot C) \times D}, E_{pos} \in \mathbb{R}^{(N+1) \times D}$$

Equation 2: Formula for forming the sequence of embedded vectors, which is the input to the encoder. Where X_{class} denotes the D-dimension class token, X^N_p denotes the p^{th} patch with size pxp, E_{pos} is the positional vector.

The input sequence is now passed to the standard Transformer encoder, which includes multihead self-attention[7], MLP blocks, LayerNorm [9] and residual connection. Output from the encoder is then passed to the classification head which is a perceptron with one hidden layer to output predictions.

Why they used the new innovations?

For years, CNN ruled the computer vision tasks. Thus, CNN models tend to be more and more complex by time, the existence of Transformer revolutionized the Computer Vision field, and show that it is possible to have another comparable replacement to complicated CNN. Secondly, as explained, CNN used conductive biases, while it allows better performance on a small dataset, it also constraint the hypothesis space and CNN only works well when most of their data conforms to this assumption. Thirdly, the positional encoding in Tranformer allows encoding and learning the relative position and distances between features, e.g., eyes are above nose, which is hard to do the same in CNN. In addition, Transformer allows modelling long-dependencies between inputs. Finally, Transformer's design is straightforward, so that different types of input (videos…) can be processed with similar blocks.

This paper has made several contributions: The first one is the extending of Transformer architecture's application to Computer Vision tasks. Secondly, the author proposed a standard and guidelines to modify/adapt this innovation to serve image input and classification tasks (e.g., split image into patches…). Thirdly, the ViT outperformed SOTA models when trained on large enough dataset. Finally, it shows that sometimes no inductive bias is advantageous.

Question 2: The key innovation of Squeeze and Excitation networks is the new Squeeze and Excitation (SE) block. Each SE Block includes a squeeze operation, which receives a feature map U ∈ R^{WxHxC} and produce a vector of length C by aggregating these maps along HxW dimensions. This vector is then followed by an excitation operation to produce per-channel weights, which are then applied to U as output and forward to next layers.

• How does this feature improve performance?

With such mechanism, this innovation improves model's performance by improve the quality of representation of the network. In details, SE blocks allows capturing channel-wise dependencies and allows feature calibration and filter, assigning different weight to each channel so that the model can learn to select informative features while suppress less useful features which can harm model's performance. In addition, within earlier layers, SE blocks has been shown to strengthening shared low-level feature, while in later layers, it emphasized class-specific features. Thus, thanks to these advantages, the model is improved in performance.

Mechanism: The squeeze operator is a global average pooling operator that output a single mean value for each channel. The vector output at length C is fed to the Excitation operator which includes two Dense layers producing vector of shape C/r where r is the reduction ratio. Between these dense layers is a ReLU, and after these layers is the Sigmoid activation.

• Squeeze and excitation networks introduce a new module. Explain how many parameters are used by these modules

There are two introduced SE module, that is the SE-Resnet module and the SE-inception module. The number of parameters for these modules can be calculated using $\frac{2}{r}\sum_{s=1}^{S} N_s(C_s)^2$, where r is the reduction ratio, C is the number of dimensions of output channels, Ns is the number of repeated blocks in stage s, and S is the number of stage where a stage means a list of blocks applied on feature map with same dimension in spatial.

The paper "Image Super-Resolution Using Very Deep Residual Channel Attention Networks" employed the idea from Squeeze and Excitation network to create residual channel attention block (RCAB) – which is a subblock to build up the final model. In particular, they employed Channel Attention on a set of feature maps (Xg,b in the figure) and multiply the weight with the output after passing pooling, convolution layer, ReLU. This enables the model to adaptively rescale channel-wise features.

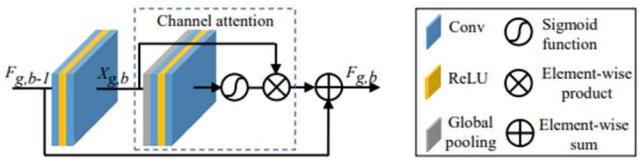

Figure 3: RCAB block's architecture

REFERENCES

[1] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” p. 9.

[2] W. B. Shen and S. Han, “Exploration of the Effect of Residual Connection on top of SqueezeNet A Combination study of Inception Model and Bypass Layers,” p. 7.

[3] “Frontiers | Automatic Facial Recognition of Williams-Beuren Syndrome Based on Deep Convolutional Neural Networks | Pediatrics.” https://www.frontiersin.org/articles/10.3389/fped.2021.648255/full (accessed Sep. 26, 2021).

[4] S. Ioffe and C. Szegedy, “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift,” p. 9.

[5] S. Santurkar, D. Tsipras, A. Ilyas, and A. Madry, “How Does Batch Normalization Help Optimization?,” in *Advances in Neural Information Processing Systems*, 2018, vol. 31. Accessed: Sep. 25, 2021. [Online]. Available: https://proceedings.neurips.cc/paper/2018/hash/905056c1ac1dad141560467e0a99e1cf-Abstract.html

[6] A. Dosovitskiy *et al.*, “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale,” *ArXiv201011929 Cs*, Jun. 2021, Accessed: Sep. 22, 2021. [Online]. Available: http://arxiv.org/abs/2010.11929

[7] A. Vaswani *et al.*, “Attention is All you Need,” in *Advances in Neural Information Processing Systems*, 2017, vol. 30. Accessed: Sep. 25, 2021. [Online]. Available: https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

[8] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” *ArXiv181004805 Cs*, May 2019, Accessed: Sep. 26, 2021. [Online]. Available: http://arxiv.org/abs/1810.04805

[9] J. L. Ba, J. R. Kiros, and G. E. Hinton, “Layer Normalization,” *ArXiv160706450 Cs Stat*, Jul. 2016, Accessed: Sep. 26, 2021. [Online]. Available: http://arxiv.org/abs/1607.06450