**Deep Learning Bird Classification: Cheat Sheet**

**1. Problem Overview**

* **Objective:** Classify bird species based on input images.
* **Challenges:** High variability in poses, lighting, backgrounds, and species appearances.
* **Approach:** Like all CNNs, Inception v3 uses convolutional layers, pooling layers, and fully connected layers to learn spatial hierarchies in images.While standard CNNs stack layers sequentially, Inception v3 uses a more complex design, processing features at multiple scales in parallel (using filters of different sizes) and improve computational efficiency.

**2. Model Selection: Why CNNs, ResNet and Inception v3?**

* **CNNs: Ideal for image data because they:**
  + Automatically learn spatial hierarchies of features.
  + Reduce parameters compared to fully connected networks by using shared weights (kernels).
* **ResNet (Residual Network):**
  + Solves the vanishing gradient problem by introducing skip connections.
  + Allows for very deep networks while maintaining efficient training.
  + Pre-trained ResNet models (e.g., ResNet-50 or ResNet-101) leverage features learned on large datasets like ImageNet, boosting performance on smaller datasets.
* **Inception v3:**
  + While ResNet is designed for very deep networks, Inception v3 finds a balance between depth and computational efficiency. This balance is often better suited to medium-sized datasets or tasks where extremely deep networks (like ResNet-101) might overfit or be unnecessarily complex.
  + Incorporates advanced regularization methods, such as batch normalization (for faster convergence and better generalization).
  + Inception modules analyze the input at multiple scales simultaneously by applying various filter sizes (e.g., 1x1, 3x3, 5x5) in parallel. This makes it particularly powerful for tasks with diverse patterns or textures in the image data.

**3. Workflow**

1. **Data Preparation:**
   * **Preprocessing**: Resize images, normalize pixel values, and apply data augmentation (e.g., rotation, flipping, cropping) to increase variability and reduce overfitting.
   * **Split to train** set and validation set: Use 20% of the training set to validate the model. The split was stratified, to have at least one representation of each class in the validation set.
2. **Model Architecture:**
   * Use a pre-trained Inception v3.
   * Replace the fully connected (FC) layer with a custom classifier for bird species (e.g., softmax layer with 200 outputs for 200 classes).
3. **Training:**
   * Fine-tune the model by unfreezing some layers (e.g., the last few ResNet blocks) while keeping others frozen to retain pre-trained features, but didn’t get the output we expected.
   * Use a learning rate scheduler to optimize training.
   * Use gridsearch cv to fine-tune different hyperparameters like learning rate, optimizer and weight decay.
4. **Evaluation:**
   * Use metrics like accuracy, F1-score, and confusion matrix to evaluate performance.
   * Evaluate based on the accuracy of the validation set.

**4. Common Questions and Answers**

**Why did you choose Inception over other architectures (e.g., VGG, Resnet)?**

* ResNet, especially deeper versions like ResNet-50 or ResNet-101, can be computationally intensive. Inception v3 provides a more balanced approach between depth and efficiency. This makes Inception suitable for setups with limited computational resources or when training time is a concern.
* Inception modules analyze data at multiple scales simultaneously by applying filters of different sizes (e.g., 1x1, 3x3, 5x5) in parallel. This allows it to capture both fine-grained and high-level features effectively, which can be crucial for complex image datasets. ResNet’s strength lies in its hierarchical feature learning and ability to train very deep networks via **skip connections**. However, it does not explicitly incorporate multi-scale processing like Inception does, which might make Inception more versatile for datasets requiring feature extraction at various levels of detail.
* Inception v3 comes with pre-trained weights on large datasets like ImageNet, making it highly effective for transfer learning. Its balance of depth, feature extraction, and computational efficiency often translates well to smaller or medium-sized datasets.

**Why is data augmentation important?**

* Augmentation increases the diversity of training samples, reducing overfitting by helping the model generalize better to unseen data.

**How does transfer learning help in this case?**

* Inception v3 comes with pre-trained weights on large datasets like ImageNet, making it highly effective for transfer learning. Its balance of depth, feature extraction, and computational efficiency often translates well to smaller or medium-sized datasets.
* Fine-tuning the model tailors these features to bird-specific patterns, reducing the amount of data and time required for training.

**What loss function and optimizer did you use?**

* **Loss function:** Cross-entropy loss is standard for multi-class classification tasks.
* **Optimizer:** Adam optimizer is often used for faster convergence, combined with a learning rate scheduler.

**How do you handle class imbalance?**

* **Use techniques like:**
  + Data augmentation for under-represented classes.
  + Weighted loss functions to assign higher penalties to misclassified minority class samples.

**How do skip connections in ResNet work – I am not sure if we need this now?**

* Skip connections allow the network to bypass certain layers, passing the input directly to later layers.
* This helps gradients flow through the network during backpropagation, avoiding vanishing gradients and ensuring effective training.

**How did you evaluate the model's performance?**

* **Metrics:**
  + **Accuracy:** Measures overall correctness but can be misleading if classes are imbalanced.
  + **F1-score:** Balances precision and recall, especially useful for imbalanced datasets.
  + **Confusion Matrix:** Identifies which species are most often misclassified.
  + ‘*not sure if we have used a confusion matrix or F1 score or not but we could mention the issue with the Kaggle submissions and if we had more time we would have used them for evaluation.*’
* **Validation Strategies:** Used k-fold cross-validation to ensure robust evaluation.

**What challenges did you face, and how did you address them?**

* **Challenge:** Overfitting due to limited data.
  + **Solution:** Applied data augmentation and used dropout regularization.
  + *‘Again we thought we were over fitting so we used these techniques, we didn’t have enough time to really know if that was our problem or not.’*
* **Challenge:** Class imbalance.
  + **Solution:** Adjusted loss function weights and used augmentation.

**How does Inception generalize better compared to other architectures?**

* Inception v3 integrates advanced regularization techniques, such as:
  + **Auxiliary classifiers:** To help combat overfitting.
  + **Label smoothing:** For better generalization.
* These features make Inception particularly robust for datasets prone to overfitting.
* ResNet doesn’t have auxiliary classifiers or similar regularization techniques, but its **skip connections** do help stabilize training in very deep networks. Still, Inception might edge out ResNet in scenarios where built-in regularization is needed.

**5. Key Hyperparameters**

* **Learning Rate:** Start with 0.001; use a scheduler to reduce it dynamically.(not sure if we used a scheduler or not)
* **Batch Size:** Experiment with sizes like 32 or 64, depending on GPU memory.(Beware of data leakage and compute crash XD)
* **Number of Epochs:** Monitor validation loss to decide when to stop (early stopping if necessary).
* **Augmentation:** Random rotations (0–30°), horizontal flips, random crops and colour jitter.

**6. Visualization Tips**

* **Feature Maps:** Use tools to visualize feature maps learned by convolutional layers. This helps explain what the model focuses on (e.g., bird beaks, wings, etc.).(**AKA LIME**)
* **Confusion Matrix:** Present a confusion matrix to highlight specific strengths and weaknesses (e.g., which species are most misclassified).

**7. Practical Advice for Handling Questions**

* **Stay confident in explaining your choices and methods.**
* **If unsure of a question, admit it and discuss how you’d approach finding the answer (e.g., further experiments or research).**
* **Bring visual aids (e.g., graphs, examples of bird misclassifications) to make your answers more compelling.**

**Pitch (~2 minutes)**

Hello all and good evening, I’m here to introduce you to our project, 'Feather in Focus'.

This project aims to tackle the challenge of identifying bird species using image classification with deep learning. We approached it by classifying 200 bird species from a dataset provided by Kaggle.

For dataset and preprocessing, we worked with 3,926 labeled training images and 4,000 test images, noting the dataset's significant class imbalance.

To address this, we applied preprocessing steps like resizing, normalizing, and augmenting images through transformations such as rotation and color jitter. A weighted random sampler was also used to mitigate class imbalance.

We split the training images into a train and validation set to 80% and 20% respectively to get validation accuracy scores before making predictions on the test set.

For our methodology, we started with a simple Convolutional Neural Network (CNN) and we went up to advanced pre-trained models like ResNet-18, ResNet50, and Inception-v3 to improve the accuracy that was very poor using a CNN model.

Also using a grid search, we found the best hyperparameters for optimizers, learning rates, weight decay values, and batch size.

The best model and hyperparameters we used was the pre-trained Inception-v3 paired with the Adam optimizer with a weight decay of 0.001 and a small learning rate of 0.0001. This delivered the best performance, achieving a test accuracy of 60.73%.

While this is promising, we observed that test accuracy lagged 6-8% behind training accuracy, which highlights overfitting challenges.

Additionally, we applied LIME, a model-agnostic interpretability tool, to explain the predictions of our models. It provided insights into classification accuracy at different levels and demonstrated how the model with Inception-v3 improved prediction precision.

In conclusion, we faced challenges like limited data and GPU constraints, our results show the potential of using transfer learning for bird species classification. Moving forward, we aim to expand the dataset for better generalization and explore more powerful architectures to further enhance accuracy.

Thank you for listening! We’d be happy to answer any questions that you may have.