Ayesha Fathima

Afathima20760@ucumberlands.edu

**NETWORK WITH CONVOLUTIONAL FILTERS**

Professor. David Ostrowski

**Title: Analyzing the Impact of Stride on Convolutional Neural Networks**

**1. Introduction**

This assignment explores the effect of different stride values on the performance of Convolutional Neural Networks (CNNs). We examine two models:

* The **Original Model** with a stride of 2 (as provided in the textbook).
* The **Modified Model** with a custom stride of 3.

We evaluate both models by training them on the CIFAR-10 dataset, comparing their training and validation accuracy, and interpreting the effects of stride on network performance.

**2. Model Architecture and Setup**

**Original Model**

The original model uses a standard stride of 2 for its convolutional layers. The architecture follows three convolutional layers, each with 3x3 filters, followed by max-pooling and dense layers.

**Modified Model**

In the modified model, we adjust the stride value to 3 in the first convolutional layer. We also added padding to ensure that the output size does not become negative during the convolution operations.

Both models were trained for 10 epochs using the Adam optimizer, and their performance was evaluated using the CIFAR-10 test dataset.

**3. Training and Evaluation Results**

**Original Model Performance:**

* **Training Accuracy (Epoch 10):** 80.37%
* **Validation Accuracy (Epoch 10):** 71.65%
* **Validation Loss (Epoch 10):** 0.8914

A screenshot of a computer

Description automatically generated

**Modified Model Performance:**

* **Training Accuracy (Epoch 10):** 59.12%
* **Validation Accuracy (Epoch 10):** 55.20%
* **Validation Loss (Epoch 10):** 1.2713

A screenshot of a computer

Description automatically generated

**4. Interpretation of Results**

**Original Model:**

* The original model performs significantly better, with training accuracy reaching over 80% and validation accuracy of 71.65%.
* The validation loss stays relatively low, indicating that the model generalizes well on unseen data.
* The smaller stride (2) means that the convolutional filters move more slowly across the image, allowing the model to capture more fine-grained details and learn features more effectively.

**Modified Model (Custom Stride = 3):**

* The modified model shows slower learning progress, with a final training accuracy of 59.12% and validation accuracy of 55.20%.
* The higher validation loss (1.2713) compared to the original model indicates that the model struggles more with generalization.
* A larger stride (3) results in the network down sampling the input more aggressively, leading to a loss of important spatial information in the feature maps. This can explain the lower accuracy and higher loss.

**5. Comparison and Discussion**

The most notable difference between the two models lies in their ability to learn and generalize:

* **Original Model**: The smaller stride allows the model to retain more spatial details from the input images, leading to better feature extraction and higher accuracy.
* **Modified Model**: A larger stride reduces the spatial resolution faster, resulting in a loss of critical information, which negatively affects accuracy and increases the loss.

While increasing the stride speeds up the down sampling process and reduces the number of parameters, it comes at the cost of reduced accuracy and generalization ability. For tasks where detail preservation is important (like object recognition), a smaller stride is more effective.

**6. Conclusion**

This experiment demonstrates that while altering the stride parameter in convolutional layers can make the network more computationally efficient, it can also lead to a significant reduction in performance. The original model with a stride of 2 outperformed the modified model with a stride of 3 in both training and validation accuracy. Future work could explore other hyperparameter adjustments, such as kernel size, padding, or deeper architectures, to optimize the balance between computational efficiency and accuracy.

Top of Form

Bottom of Form

**REFERENCES**

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.

Krizhevsky, A., Hinton, G., & Nair, V. (2009). *Learning multiple layers of features from tiny images*. CIFAR-10 Dataset.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. <https://doi.org/10.1038/nature14539>

Agarap, A. F. (2018). Deep learning using rectified linear units (ReLU). *arXiv preprint arXiv:1803.08375*. <https://arxiv.org/abs/1803.08375>