SEQUENTIAL EXPERIMENTATION COMPARING CNN AND DNN COMPLEXITY AND IMAGE CLASSIFICATION PERFORMANCE

Jamia Russell[1,†]

[1] Northwestern University School of Professional Studies
Master of Science in Data Science Program
633 Clark St. Evanston, IL 60208

Address to which correspondence should be addressed:
jamialashe@gmail.com

Abstract

Examining the impact of varying neural network hyperparameter settings and topologies on computer vision tasks such as image classification, this paper explores deep learning techniques and neural network architecture manipulation's role in accurate image processing. Using this CIFAR-10 data set allowed for evaluation across Dense Neural Networks (DNNs), and Convolutional Neural Networks (CNNs) comparing structure, computational efficiency, feature extraction, scalability, and generalization efforts.

Pooling layer operations, regularization, convolutional layers, activation functions, node variation are utilized to alter network structure to depict the most efficient network in classifying images. Specifically, dropout layers, ReLU, Softmax, Tanh, ELU, and SELU activation functions are used to measure network efficiency.

This paper examines the network performance through max pooling and batch normalization techniques, presenting the effects of changing network architecture during sequential experimentation of CIFAR-10 data classification.

The networks were created and analyzed using Python, TensorFlow, and Keras packages. Key Results uncovered that Convolutional Neural Networks outperform Dense Neural Networks in classification accuracy by substantial margins and meet most efficient model performance with deep networks including ReLU activation, normalization, and regularization.

.

Keywords: Dense Neural Networks (DNNs), Convolutional Neural Networks (CNNs), Hidden Layer Nodes, CIFAR-10, Facial Recognition

Table of Contents

I. Introduction & Problem Statement

As the capabilities of artificial intelligence increase industries benefit from deep learning functions to obtain meaningful information from digital images and videos. This can relate to facial recognition, self-driving cars, medical abnormality detection, and robotic automation (Rickson, 2022). Within the healthcare industry professionals across practice areas have seen artificial intelligence as a tool in creating general and condition specific technologies and strategies to aid medical condition diagnosis and treatment.

Specifically, artificial intelligent and deep learning techniques have a 97.66% success rate analyzing oral biopsy images classifying malignancy (Rawi et al., 2022). Convolutional Neural Network architectures were used to classify oral lesions and their stage in cancer development with models outperforming experienced practitioners. It is expected as confidence in artificial intelligence abilities increases, machine and deep learning techniques will be utilized across many industries daily.

Using the CIFAR-10 data set to develop a variation of Dense Neural Network and Convolutional Neural Network models to predict and classify with accuracy categorical computer images deep learning techniques max-pooling and batch normalization are key. Max pooling used in CNN architectures alter dimensionality and computational load whilst maintaining necessary features. Batch normalization, normalizes activations and improve network generalization and stability.

Obtaining optimal architecture and training in deep learning networks to achieve high performance is challenging. Real world applications such as facial recognition security authentication for mobile devices require accuracy and computational efficiency and robustness to reflect variations in human phenotype. There is a need for high dimensional image data and the ability to extract spatial features which may be difficult to achieve in networks and can lead to negative outcomes such as overfitting, slower processing, and reduced generalization. Despite this, there is feasibility in CNNs and DNNs adaptable frameworks providing accurate and efficient facial recognition.

II. Literature Review

The proficiency of deep learning techniques have led to revolutionary advancements in image and object recognition. Developments in object detection, facial recognition, and image detection have exceeded traditional machine learning techniques in providing accurate networks and computationally efficient performance. Sinh et al. (2018) dissected the different network frameworks that encompass deep learning techniques finding that Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Restricted Boltzmann Machines (RBMs), and Auto Encoders - despite the large scope for advancement surpass precision in human testers.

CNNs and DNNs and Image Classification

Due to the nature of deep learning techniques there is an acknowledgement of CNN and DNN capabilities as it relates to classification. Because of this, Klawonn at al., (2024) constructed a CNN-DNN architecture supporting parallel training built off established CNN structure. Combining the effective preprocessing and grid topology with DNN training time efficiency to create a network efficient in classifying two dimensional images. Results found that the combined architectures accelerated training time and improved classification accuracy.

Dutta et al., (2017) compared CNN and DNN medical image classification performance using Diabetic Retinopathy and Emphysema computed topology to assist in diagnosis of these conditions. Focusing on diabetic vision loss and emphysema data recognition, DNN hidden layer nodes were manipulated increasing accuracy, but left issues with vanishing gradients. CNNs were then used to increase classification of the convoluted data adapting to the robustness and complexity.

Deep Learning and Facial Recognition

Recent literature depicts the use of CNN models for image facial recognition and security authentication use. In 2018 Ghenescu et al., used the YOLO (You Only Look Once) model method encompassing CNN Darknet-19 network architecture to detect "different instances of the same class"(Ghenescu, 2018). A data set of 120,00 images across 10 subjects was used to test classification accuracy and model performance in interacting with differing input and facial resolution, image quality, lighting conditions, and rotation angle. 17 configurations of DarkNet-19 led to a classification accuracy of 89.65%, containing a 416x416 pixel input resolution, 16 filter first convolutional layer and 512 filters in the last layer.

Ling et al. (2020), examined the role and impact of activation functions on CNN models in facial recognition. Sigmoid, ReLU, tanh, leaky-ReLU, and softplus-ReLU were analyzed and compared to a new activation LS-ReLU. The small JAFFE data set was used to evaluate photos of individual expressions across 7 categories - anger, disgust, happiness, fear, normal, surprise, sad. Results showed that the LS-ReLU learning rate was 99.91% compared to the other activation functions spanning from 57.43% to 96.25%. The new activation was found to preserve

ReLU features, but uses an adjustable log function and softmax function to avoid overfitting and reduce oscillations.

III. Methods

Data Overview

To complete the evaluation of Convolutional Neural Network and Dense Neural Network architectures as it relates to image classification the CIFAR 10 dataset was used. The CIFAR 10 dataset contains 60,000, 32x32 pixel images across 10 categories including dog, cat, bird, deer, frog, airplane, automobile, horse, ship, and truck. The diversity of objects and image resolution allow for measures of accuracy, generalization, and advantages of different network architectures to be used as a guide for effective and efficient image classification.

The availability of computational feasible data is essential for deep learning research. Specifically, in image classification challenging datasets provide more insightful results that inform future network structure. Here, CIFAR-10 complexity will provide robust representation of deep learning techniques such as regularization and batch normalization.

To conduct preprocessing and use of the CIFAR-10 data for experimentation Python programming language was used. Preprocessing included many steps such as splitting of training and validation data, normalization, and one-hot encoding. The data was fit into a 50,000/10,000 split for training and validation respectively, and additionally split by image data and class label. Pixel images were normalized by dividing the existing pixel value by the maximum pixel value (255) re-scaling the pixel to a range between 0 and 1.From there, one hot encoding converts the existing class integer labels 1-10 into binary vectors. This preprocessing ensures proper formatting for interaction with the CNN and DNN network models.

Network Construction - DNN Experiments

Keras and TensorFlow Python libraries were used to create the deep feedforward DNN network used during the sequential experimentation. Constructed using Sequential API allowing for layer by layer structure definition this two layer network includes an initial flattening layer to promote compatibility with dense layers that converts the 32x32x3 images into a one dimensional vector of 3,072 elements. These 32x32x3 reflect image pixel value and 3 color channels. The output layer contains 10 neurons corresponding to CIFAR-10 class categories. An Adam optimizer trained on the network enables categorical cross-entropy and convergence. To determine the effectiveness of the network, experiments with varying hidden layers, activation function use and combination, and regularization were altered to examine impact on network performance.

In experiment 1 (DNN-2), the first hidden layer contains 512, the second 256 neurons. A Rectified Linear Unit (ReLU) activation introduced non-linearity and softmax activation computed probability distributions for multi-class identification.

Experiment 2 (DNN-3), uses three hidden layers containing 512,256, and 128 neurons, respectively all using ReLU activation to enable network learning capabilities, implement non-linearity, and prevent vanishing gradients. A decreasing number of neurons encourages extraction of hierarchical features to learn abstract patterns.

Experiment 3 (DNN-2-1), uses two hidden layers and combines ReLU and Tanh activation functions to train the network. Hidden layer one containing 512 nueroons and utilizing ReLU activation function increases computational efficiency and by only allowing positive values it avoids vanishing gradients. The second hidden layer with 256 neurons employs the Tanh activation to map input ranges between -1 and 1 to capture feature relationships and increase optimization. Here the output layer uses the softmax activation function to create a probability distribution. The combination of activation functions aim to increase pattern recognition while avoiding overfitting.

Experiment 4 (DNN-3-1), uses two hidden layers and ReLU and Tanh activation functions. The first dense layer containing 512 neurons uses ReLU to prevent vanishing gradients and limit to positive activations, The second, containing 256 implements Tanh activation to center data between -1 and 1 stabilizing weight. Lastly, the third layer contains 128 neurons and uses Sigmoid activation to limit the values to between 0 and 1.

Experiment 5 (DNN-2-2), uses two hidden layers and regularization techniques L2, batch normalization, and dropout to refine generalization and prevent overfitting. The first hidden layer similar to experiment 3 includes 512 neurons and the ReLU activation function, but also introduces L2 regularization adding a penalty term to the loss function avoiding large weight values. Following this, is a layer of batch normalization stabilizing training and accelerating convergence. The same structure and process continues with the second layer containing 256 neurons. After this, to prevent overfitting a dropout layer with a 50% rate disabling neurons in the layer during training forcing the network to interact with more feature representations.Lastly, the output layer uses the softmax activation for probability distribution.

Experiment 6 (DNN-3-2), uses three hidden layers, ReLU activation and L2 regularization and a dropout layer. Similar to the experiment 4 hidden layers one and two with 512 and 256 neurons respectively layer use ReLU and L2 followed by bach normalization. Within the new third hidden layer with 128 neurons ReLU, L2, and batch normalization take place. After this, a dropout layer is implemented to refine feature representation. The goal of this repetition is to ensure there is balance in depth, regularization, and generalization within the network.

Network Construction - CNN Experiments

Similar to the DNN architecture, Keras and TensorFlow Python libraries were used to create the Convolutional Neural Network (CNN) model used during the sequential experimentation. The network contains two convolutional and max-pooling layers with ReLU activation. The initial Conv2D layer extracts low level edge and texture features applying 32 filters and kernel size 3x3. A max-pooling 2D layer then reduces dimensionality and retains important features.Another Conv2D layer with 64 filters enhances feature extraction performance and a second max-pooling 2D layer downsamples the feature maps. After these convolutional steps the network now flattens the input data converting the 2D features into 1D vectors allowing for compatibility with other layers. The output layers contain 10 neurons reflective of CIFAR-10's existing class categories, and a softmax activation function. Lastly, an Adam optimizer is implemented for convergence and categorical cross-entropy loss. To determine the effectiveness of CNN architectures as it relates to image classification, varying activation functions and regularization techniques were used to examine network performance.

. Experiment 1 (CNN_2), in addition to the two max-pooling and two convolutional layers with ReLU activation, follows with a dense layer containing 128 neurons and ReLU activation encouraging the network to examine high level feature representations.

Experiment 2 (CNN-3), with the existing two max-pooling and two convolutional layers adds another of each. The third convolutional layer contains 128 filters representing more detailed abstract features. The third max-pooling 2D layer downsamples then inputs are flattened to 1D vectors from 2D features. After input and flattening the dense layer with 128 neurons, ReLU activation improves feature representation.

In experiment 3 (CNN-2-1), similar to the first two using the two max-pooling and two convolutional layers to address vanishing gradients. reduce spatial dimensions, and increase computational complexity. Experiment 3, incorporates ReLU, Tanh, Sigmoid, and Softmax activation functions at different layers of the network. Tanh activation is included in the second Conv2D layer applying 64 filters outputting feature representation values between -1 and 1. The 128 neuron dense layer with Sigmoid activation limits output between 0 and 1, and non-linearity unlike ReLU. At the output layer Softmax activation is used for class probability.After input and flattening the dense layer with 128 neurons, ReLU activation improves feature representation

Experiment 4 (CNN-3-1), uses three convolutional and three max-pooling layers for network structure. The first convolutional layer applying 32 filters and using ReLU activation zeros out negative values and diminished vanishing gradients The first max-pooling layer reduces spatial dimension improving network efficiency. The second Conv2D layer with Tanh activation 64 filters encourages weight balancing. The second max-pooling is added to continue downsampling. The third Conv2D layer with 128 neurons introduces Exponential Linear Unit (ELU) activation function which permits small negative values for outputs ensuring smoother gradient flow. The first dense layer made of 128 neurons integrates Scaled Exponential Linear Unit (SELU) activation function. SELU self-normalizes during training to maintain stable activations and prevents exploding gradients. After input and flattening the dense layer with 128 neurons, ReLU activation improves feature representation

Experiment 5 (CNN-2-2), contains two convolutional layers, two max-pooling layers, batch normalization, L2 regularization, and dropout. The network’s first Conv2d layer applies 32 filers with ReLU activation followed by batch normalization.The first max-pooling layer preserves key features and reduced dimension. The second Conv2D layer with 64 filters, ReLU and L2 regularization prevents overfitting. Next, another batch normalization layer and max-pooling layer downsamples feature maps and encourages activation distributions.After input and flattening the dense layer with 128 neurons, ReLU activation improves feature representation

Experiment 6 (CNN-3-2), uses three convolutional layers, three max-pooling layers, batch normalization, L2 regularization, and dropout. The initial Conv2D layer with 32 filters paired with ReLU activation, followed by batch normalization normalizes activations and increases convergence speed. Right after the max-pooling layer, the second Conv2D layer with 64 filters paired with ReLU activation and L2 regularization prevents large weights. A second batch normalization stabilizes activation and the second max-pooling layer reduces samples. The final Conv2D layer applies 128 filters with ReLU activation and L2 regularization,for more complex feature extraction. Next, batch normalization and another max-pooling layer ensure diverse feature representation. After input and flattening the dense layer with 128 neurons, ReLU activation improves feature representation.

IV. Results

DNN Experiments

DNN-2 with a run time of 4 ½ minutes showed improvement over the 10 training epochs showing an increase in accuracy from 32.79% to 51.27% at the last epoch. Validation accuracy ranges from 38.32% reaching a high of 50.66% before lowering to 48.53% on the tenth epoch. These results indicate that the network is overfitting values and is unable to generalize and learn from itself. Through loss decline the network shows there are inefficiencies where additional deep learning techniques can improve performance.

DNN-3 delivers an initial accuracy of 31.90% rising to 44.59% by the 10th epoch. Validation accuracy reaching a high of 41.23% shows the network addition of another dense layer with node variation increases model learning progression, but does not substantially improve performance. However, fluctuating validation accuracy proves there are overfitting or generalization issues within the network. Epoch 4 and epoch 9 validation loss suggests that the network struggles with unseen data. The model can benefit from additional complexity.

There is a consistent improvement in accuracy DNN-2-1's 10 epochs. Accuracy starts at 27.33% at the first epoch to 52.26% by the tenth. Validation accuracy also shows steady increase reaching 48.33% at the final epoch. Training loss declines and validation loss fluctuates downward. Here, Tanh activation in the second dense layer stabilizes the network. Despite this, validation accuracy does not perform well enough to show true impact.

DNN-3-1 training and validation accuracy depict an incremental increase across the 10 epochs. Network training accuracy and validation rose from 28.52% and 36.04% to 52.99% and 48.12%, respectively. Training loss depicts a consistent decrease indicating the network is learning despite validation loss fluctuation that shows there is room for improvement with unseen data. The training log depicts ReLU activation impact on training accuracy in the second epoch where there is an increase of 11%. Fluctuation of validation of loss may be from Sigmoid activation reducing model capacity to present diverse features.

DNN-2-2's use of ReLU activation, L2 regularization, and batch normalization led to consistent improvements in training and validation accuracy across 10 epochs from 31.90% to

44.59% . L2 regularization decreased model loss from 2.47 to 1.76 by preventing overfitting. There are limited fluctuations in training loss and increases in accuracy likely caused by batch normalization. Model accuracy improvement shows that dropout preventing overfitting caused improvement in network generalization, but slowed down convergence as all neurons were not utilized.

DNN-3-2 showed a steady increase in training accuracy from 32.34 to 46.78%, but limited increase in validation accuracy peaking at 43.72%. This indicates that regularization impacted generalization performance. Drops in validation accuracy suggest that a 50% dropout rate may have caused instability.

CNN Experiments

Performance of CNN-2 depicts convolutional networks effectiveness in image classification tasks. The network training log depicts substantial increases in both training and validation accuracy across all 10 epochs rising from 46.23% to 81.09% for training and 56.95% to 71.22% for validation. ReLU's introduction of non-linearity allowed the model to learn data patterns. Despite a plateau at 71% the network showed a balance in under and overfitting.

CNN-3, an improved version of CNN-2 shows substantial increases in training and validation accuracy. Specifically, an almost 40% increase in training accuracy 43.06% to 80.58% across 10 epochs. Validation accuracy jumps almost 52.38% to 71.91%. The third convolutional layer contributed to the model feature extraction and representation while ReLU increased model learning and prevented vanishing gradients. Despite this validation accuracy plateau's at 72% suggesting overfitting.

CNN-2-1's performance depicts notable increases in network training and validation accuracy. Jumping from 45.09% to 81.12% training accuracy, and 51.17% to 68.49% validation accuracy. This large incremental increase is due to ReLU activation performance in extracting strong feature representation, and Tanh activation encouraging the model to balance those representations. However, validation loss indicates that Sigmoid activation in the dense layer may have limited the network's ability to identify complex relationships in deeper layers. In addition, CNN-2-1's overall validation accuracy compared to CNN-2 and CNN-3 uncover overexertion due to activation functions. Implementing oher deep learning techniques such as dropout and normalization may improve network performance.

CNN-3-1 demonstrated a stronger initial training accuracy at 50.21% and test accuracy of 70.28% the network's additional convolutional layer improved feature extraction. Comparing network runtime CNN-3-1 does have a longer processing time due to the extra convolutional layer. Higher validation accuracy and validation loss indicates overfitting.

CNN-2-2's network training accuracy rises from 41.70% to 75.09% across the 10 epochs while validation accuracy reaches a high of 70.65% depicting a generalization performance better than previous models. Normalization contributed accuracy progression due to normalizing activations. The effect of L2 regularization is seen with the network's lower convergence rate and gradually increasing training accuracy. Validation accuracy instability seen in epoch 8 suggests that refining of dropout may improve generalization and increase model stability.

CNN-3-2's presents significant improvement in model learning, but suffers from challenges with overfitting.The network's three convolutional layers all using ReLU all contribute to faster gradient gradient propagation. This is seen with the dramatic increase in training and validation accuracy from epoch 1 at 43.69%, and 49.54% to epoch 3 at 63.56% and

65.81%, respectively. Higher training and validation accuracy indicate batch normalization has prevented excess activation drift. However, training and validation loss show hat L3 regularization has slowed down model convergence and has not done well to prevent overfitting as the model learns complex representations.

V. Conclusion

The sequential experiments on Dense Neural Network (DNN) and Convolutional Neural Networks (CNN) explore deep learning techniques that include activation functions, regularization, activation function selection, and normalization in image classification performance. Across experiments each network model presented varying advantages and disadvantages, displaying the effect of architecture design and technique implementation on computational efficiency and performance. Activation function performance of ReLU, Sigmoid, ELU and Tanh show ReLU outperformed the others in the convolutional layer across all models in encouraging convergence and preventing vanishing gradients. Sigmoid and Tanh activations brought model inefficiency in the dense layers slowing down weight updates negatively affecting weight updates. L2 regularization reduced overfitting across models, but slowed convergence and increased loss towards latter epochs. Dropout, despite its effectiveness in model fitting the rate of .5 across models proved to be aggressive and limited validation performance. Network's with batch normalization have consistent and incremental changes avoiding drastic fluctuations on model learning and thus accuracy. Albeit, deeper networks did show normalization would perform better with refined parameters. Overall CNN models outperformed DNN models showing that convolutional networks with multiple layers improve feature extraction and model accuracy. CNNs with deeper networks with normalization and regularization have efficiently improved network convergence rate, stabilized training, and controlled overfitting for image tasks.

Among all models there was room for improvement. There were instances where unnecessary complexity led to activation function and regularization technique mismatch in function leading to inefficiency and missed opportunities for model learning. For example, use of dropout and L2 regularization caused excess refinement limiting model learning capacity and feature representation. Network model CNN-3 outperformed all models reaching a 71.89% accuracy and practical training time.

Tuning CNN-3 architecture for facial recognition modifying the output layer and adjusting pre-processing to include normalization and adding a 128 dimension feature embedding to classify faces. Implementing a triplet loss function to ensure embeddings were clustered whilst also differentiating identity would be beneficial. Measures of performance through false acceptance and rejection rate would inform the network's effectiveness and room for improvement.

References

Al-Rawi, N., Sultan, A., Rajai, B., Shuaeeb, H., Alnajjar, M., Alketbi, M., Mohammad, Y., Shetty, S. R., & Mashrah, M. A. (2022). The Effectiveness of Artificial Intelligence in Detection of Oral Cancer. *International Dental Journal*, *72*(4), 436–447. https://doi.org/10.1016/j.identj.2022.03.001

Dutta, S., Manideep, B. C. S., Rai, S., & Vijayarajan, V. (2017). A comparative study of deep learning models for medical image classification. *IOP Conference Series: Materials Science and Engineering*, *263*, 042097. https://doi.org/10.1088/1757-899x/263/4/042097

Ghenescu, V., Roxana Elena Mihaescu, Serban-Vasile Carata, Marian Traian Ghenescu, Eduard Barnoviciu, & Mihai Chindea. (2018). *Face Detection and Recognition Based on General Purpose DNN Object Detector*. https://doi.org/10.1109/isetc.2018.8583861

Karthikeyan Elumalai. (2024). Improving oral cancer diagnosis and management with artificial intelligence: A promising future for patient care. *Oral Oncology Reports*, *11*, 100624–100624. https://doi.org/10.1016/j.oor.2024.100624

Klawonn, A., Lanser, M., & Weber, J. (2024). A Domain Decomposition–Based CNN-DNN Architecture for Model Parallel Training Applied to Image Recognition Problems. *SIAM Journal on Scientific Computing*, *46*(5), C557–C582. https://doi.org/10.1137/23m1562202

Krizhevsky, A. (2009). *CIFAR-10 and CIFAR-100 datasets*. Toronto.edu. https://www.cs.toronto.edu/~kriz/cifar.html

Sawtell-Rickson, J. (2022, December 21). *What Is Computer Vision? (Definition, Examples, Uses) | Built In*. Builtin.com. https://builtin.com/machine-learning/computer-vision

Sinha, R. K., Pandey, R., & Rohan Pattnaik. (2018). Deep Learning For Computer Vision Tasks: A review. *ArXiv (Cornell University)*. https://doi.org/10.48550/arxiv.1804.03928