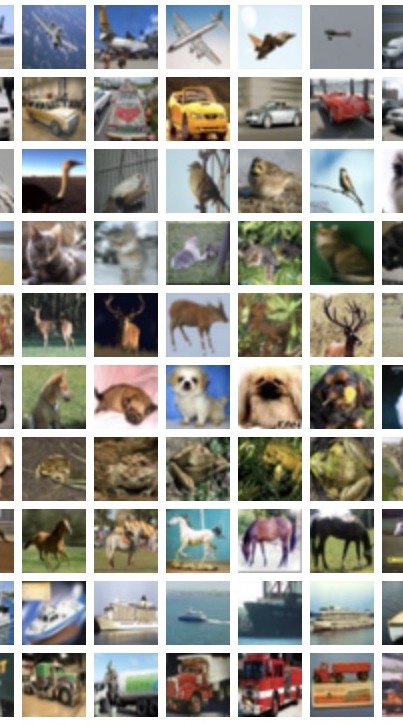
**Image classification by deep learning.**



Yohanan Kling: 316004555  
Tair Mazriv: 209188382

**Summary**

-Classification of images from the cifar10 data according to the categories using a model

- Comparison with logistic regression

- Building a model using a neural network with 10 epochs in part 2 with 69 percent accuracy

- Improving the model from part 2 and comparetion between the models

- Presentation of the updated model with 10 epochs and with 15 epochs

**Introduction.**

תמונה שמכילה יונק, חתלתול, שפם, חתול

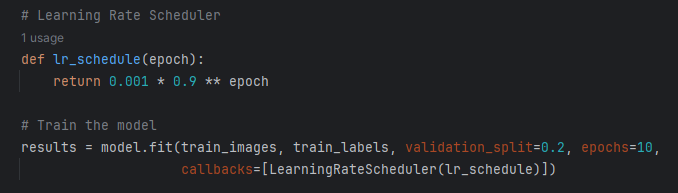
התיאור נוצר באופן אוטומטי**Image classification:**  
Image classification is the task of assigning images to predefined classes based on their visual content. It enables many critical applications such as medical imaging, self-driving vehicles, and facial recognition systems.  
  
**Data set:**  
The CIFAR-10 dataset used consists of 60,000 32x32 color images across 10 classes including airplanes, cars, birds, cats, deer etc. The images have 3 channels (RGB) and integer pixel values from 0-255. The training and test sets contain 50,000 and 10,000 images respectively.  
  
**method:**  
The dataset was split into 83.4% training, 16.6% test and 20% validation sets. Pixel values were normalized to [0,1] by dividing by 255 to speed up learning. Labels were one-hot encoded for categorical classification.  
The aim is to utilize the image pixels to train the model for classifying images into one of ten predefined categories.  
  
**Goal:**  
achieve a high rate of correct classiﬁcations while avoiding overﬁtting.  
Trying different models to decide which is the most suitable for the task.  
In each model, ﬁnding the right balance between the training, validation, and test datasets.  
  
**Load and normalize data:**  
The images are in 32X32 pixels and to normalize them, the model convert each value of each pixel to be in a range between [0,1] by dividing it by 255.   
In addition, the model converts the labels to one-hot encoding for both training and test data, representing each label as a vector with a 1 in its corresponding category and 0s everywhere else. For example, if the categories are {A, B, C, D, E, F}, then the label "D" would be represented as the vector [0, 0, 0, 1, 0, 0].

**Logistic Regression model.**

The Logistic Regression method consists of four main steps.  
  
1. **Flatten Images**: Reshape 3D images (like pixels, height, width) into 2D arrays for Logistic Regression.   
  
2. **Train Model:** Train a Logistic Regression model with MAX\_ITERATIONS (variable that can be changed) of optimization steps.   
  
3. **Predict & Evaluate:** Make predictions on unseen images result and calculate accuracy compared to true labels (converted to single class using argmax).   
  
4. **Loss:** Log loss (discrepancy) helps to predict probability of model accuracy and improvement potential.

**result**:  
After several improvements and a learning process, the logistic model gave an accuracy of 0.4054.

**Improve the CNN model.**

**Architecture:**  
The newer model has additional layers compared to the older one. Specifically, it has an extra convolutional layer with 128 filters and a dense layer with 128 units.   
  
**Batch Normalization:**  
The batch normalization standardizes inputs to each layer (by an average around 0 and a standard deviation close to 1), making them easier to learn from and improving overall network performance.  
  
**Dropout:**  
Randomly drops connections during training to prevent overfitting and encourage various of features.  
  
**Learning rate:**  
The learning rate scheduler function for decreasing the learning rate exponentially with each epoch by multiplying 0.001 by 0.9 raised to the power of the current epoch.  
  


In addition, there is the max pooling and flatten layers that were also in the old model.

**2 max-pooling layers**

follow the convolutional layers. Max- pooling reduces spatial dimensions, helping the network focus on the most important features.

**Flatten Layer**

to flatten the output of the last convolutional layer, preparing it for the fully connected (dense) layers.  
  
  
**Results of the new model in comparison to the results of the old model**:

תמונה שמכילה טקסט, צילום מסך, קו, תרשים

התיאור נוצר באופן אוטומטיתמונה שמכילה טקסט, צילום מסך, קו, עלילה

התיאור נוצר באופן אוטומטיתמונה שמכילה צילום מסך, טקסט, גופן, קו

התיאור נוצר באופן אוטומטיתמונה שמכילה טקסט, צילום מסך, גופן, תפריט

התיאור נוצר באופן אוטומטי

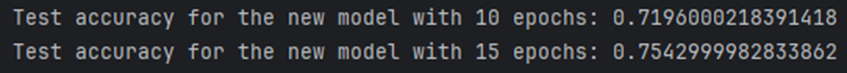
New model training epochs:

**An increase of almost 2.5 percent accuracy**

**An attempt to improve the accuracy using 15 epochs:**

תמונה שמכילה טקסט, צילום מסך, קו, עלילה

התיאור נוצר באופן אוטומטיתמונה שמכילה טקסט, צילום מסך, קו, עלילה

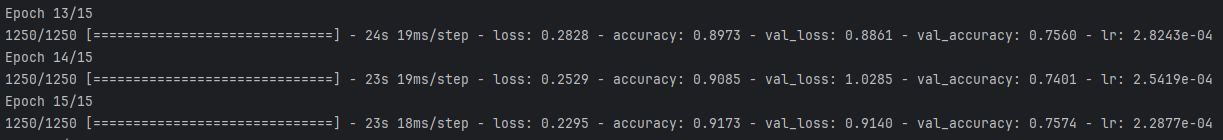
התיאור נוצר באופן אוטומטי

**An increase of 3.5 percent accuracy**

**The final model:**

תמונה שמכילה טקסט, צילום מסך, תפריט, גופן

התיאור נוצר באופן אוטומטי



תמונה שמכילה טקסט, תרשים, קו, עלילה

התיאור נוצר באופן אוטומטיתמונה שמכילה טקסט, תרשים, עלילה, קו

התיאור נוצר באופן אוטומטי

  
  
**Conclusion:**

**The most successful model with 75.4 percent accuracy**

The decision to utilize a Convolutional Neural Network (CNN) over a logistic model has proven to be advantageous, resulting in improved accuracy for the task. Leveraging CNNs allows the model to effectively capture spatial hierarchies in the data, particularly beneficial for image classification tasks like the one being addressed. Moreover, incorporating validation data during training enhances the model's adaptability and guards against overfitting, ensuring it generalizes well to unseen data.

Architectural enhancements such as batch normalization, dropout, and dynamic learning rate adjustments have significantly contributed to refining the model's performance. Batch normalization aids in stabilizing and accelerating the training process, while dropout introduces regularization, mitigating overfitting by randomly dropping units during training. Additionally, dynamically adjusting the learning rate schedule facilitates adaptive learning rates, further optimizing model training.

As a result of these improvements, the CNN model achieves a notable accuracy of approximately 75% across ten distinct classes. This achievement underscores the effectiveness of incorporating advanced techniques and optimizations in developing a robust and high-performing classification model.